

Processor Architecture
Past
Present
Future

Steve Wallach

swallach"at"conveycomputer.com

Discussion

- What has happened in the past
 - Instruction Set Architecture
 - Logical Address Space
 - Compilers
 - What technology survived
- What should happen in the future
 - Is it time for a transformation?
 - Is it time for heterogeneous computing?

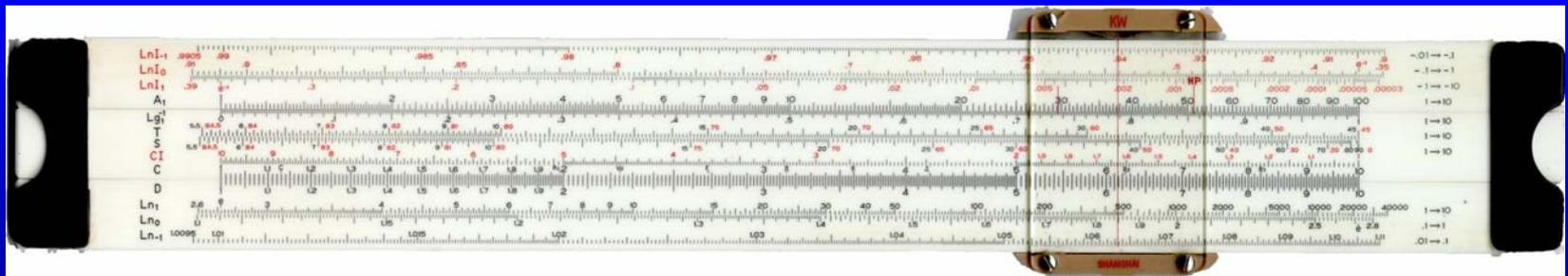


History

- 1960's, 1970's, 1980's, 1990's, 2000 & Today

“Those who can not remember the past are condemned to repeat it”

George Santayana, 1905

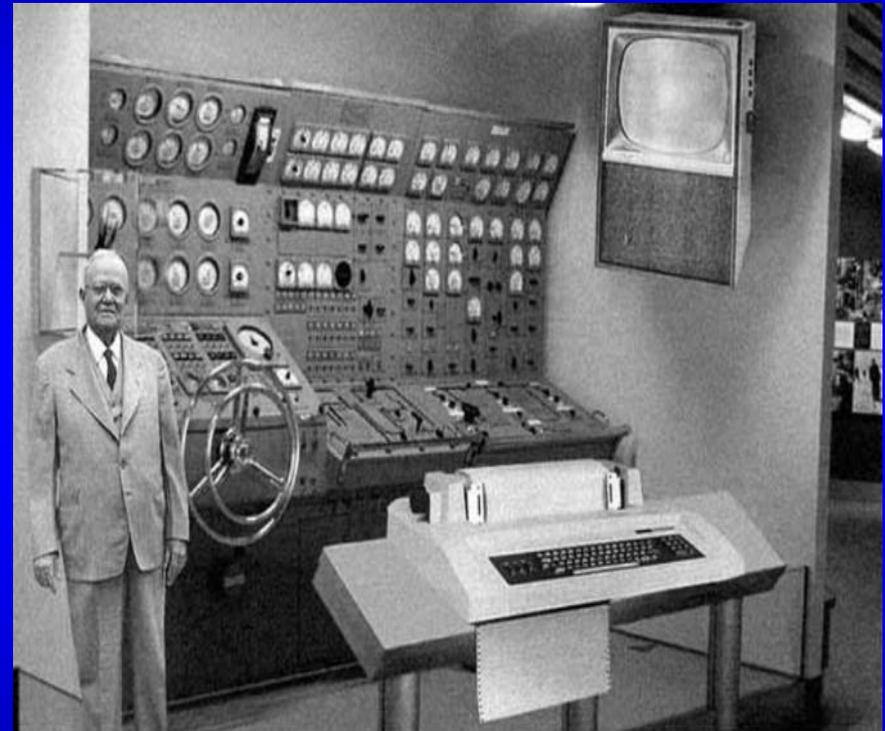


Way Back When – 1960's

- Commercial – IBM 1401 (1960's)
 - Character Oriented
- Technical – IBM 7040/7090 (1960's)
 - Technical
 - Word oriented
 - Floating Point (FAP)
- 1966 – IBM 360
 - One integrated commercial and technical instruction set
 - Byte addressability
 - Milestone architecture
 - Family of compatible systems
- 1966 – CDC – Technical Computing
 - Word Oriented

Address Space/Compilers - 1960

- Mapped Physical
 - 12 to 24 bits
- Project MAC (Multics)
 - Virtual Memory
 - Process Encapsulation
- Fortran Compilers begin appearing
 - Can you really write an application in a higher level language?



Scientists from the RAND Corporation have created this model to illustrate how a "home computer" could look like in the year 2004. However the needed technology will not be economically feasible for the average home. Also the scientists readily admit that the computer will require not yet invented technology to actually work, but 50 years from now scientific progress is expected to solve these problems. With teletype interface and the Fortran language, the computer will be easy to use.

1970's

- The decade of the minicomputer & language directed design
 - APL Machines
 - ALGOL Machines (Burroughs 5500/6500)
 - Complex ISA (e.g., VAX) (Single Instruction per Language Statement)
- Co processor
 - Floating Point – (Data General and DEC)
 - Microcoded and Hardwired
 - String and Byte instructions
 - Writable Control store for special apps
- B1700
 - S-language instruction set
 - Different ISA for Fortran, Cobol, RPG, etc
- Cray – 1 – Vector Processing for Technical Market
 - TI ASC
 - CDC STAR
- Array Processors to accelerate minicomputers (primarily)
 - FPS 120b/264
 - IBM 3838
 - CDC MAP

Address Space/Compilers - 1970

- Movement from 16 to 32 bits
- Multics Trickles Down (Intellectually) to Massachusetts Companies
 - DEC (VAX)
 - DG (MV)
 - Prime
- Rethinking the Address Space Model
 - Object Based, System-Wide & Persistent Address Space
 - IBM Future System (FS)
 - Data General Fountainhead (FHP)
 - INTEL I432
- Compilers begin to perform optimizations
 - Local & Beginnings of Global
 - Beginnings of dependency analysis for Vector Machines
 - Hardware prompts compiler optimizations

1970's

- We begin to see specialized processors and Instruction sets tuned to particular applications
- Unix emerges
 - Singular MULTICS
- Array processors used for signal/image processing
 - 2 compilers needed
 - “vertical programming”
- System Definitions:
 - Mainframe - West of the Hudson River
 - Minicomputer - East of the Hudson River

1970's What we learnt

- Hardware makes user application software easier to develop
 - Virtual Memory
 - Large Physical Memory
 - Application accelerators were commercially viable
 - Single/image processing
 - Writable Control Store (Microprogramming)
- Compiler and OS Technology moving to take advantage of hardware technology
 - Dependency Analysis (vectors)
 - University of Illinois
 - Process Multiplexing and multi-user

1980's

- Vector and Parallel Processors for the masses
 - Vector and Parallel Instruction sets
 - Convex and Alliant
 - Virtual Memory
 - Integrated scalar and vector instructions
- Beginnings of the “killer micro” (RISC)
 - MIPS, SPARC, PA-RISC, PowerPC
- VLIW Instructions
 - Instruction Level Parallelism (superscalar)
 - MultiFlow
- Unique designs for unique apps
 - Systolic
 - Dataflow
 - Database
 - ADA Machine (from Rational)
 - LISP Machine from Symbolics
 - DSP

The advertisement is divided into two sections. The top section features a Dell System 320LX computer with a monitor displaying '386SX'. The bottom section features an ALR PowerFlex Model 1 Computer with a monitor displaying '286'. Both sections include detailed specifications and pricing.

DELL COMPUTER CORPORATION
Dell System® 320LX
- 386SX 20MHz
- 1MB RAM - 40MB HD
- 1.2MB 5¼" floppy drive
- VGA Color Plus monitor/card
- One year on-site service
\$2799
Option #2: With MS-DOS 3.3 Add: 99⁹⁵

ALR
PowerFlex Model 1 Computer
- 286 12.5MHz
- 1MB RAM
- 1.44MB 3½" floppy drive
- Upgradeable to 386/486SX
\$849
Option #2: Above w/40MB HD, mono monitor and card. **\$1249**
Option #3: Above w/40MB HD, VGA/color monitor and card. **\$1499**

Address Space/Compilers – 1980's

- Systems generally 32 bit virtual (or mapped)
 - More Physical Memory
 - Better TLB designs
 - What is the size of INT? (Unix issue)
 - Big or Little Endian
- Compilers perform global optimization for Fortran and C
 - Automatic Parallelization
 - University of Illinois & Rice

1980's

- Portability of Unix and Venture Capital
 - New Machine Architectures
 - Beginning of Open Source Movement
 - LAPACK
- Scalar Instructions form basis of all new architectures
- Moore's Law HELPS to create new architectures
- Array Processors disappear
 - Integrated Systems easier to program
 - Dual licenses for certain apps
 - Host and attached processor

1980's What we learnt

- Parallel machines are easy to build but harder to program
- Rethink applications
- New languages (i.e., C & C++) get used and accepted because users like to use them and NOT due to an edict (i.e., ADA)
- Compilers and OS move to parallel machines
- Startups provide the innovative technology
- Hardware makes user application software easier to develop

1990's

- Microprocessor microarchitecture evolves
 - Moores Law and Millions of Transistors drive increase in complexity
 - Multi-threading
 - SuperScalar
- ILP
 - Itanium (multiple RISC instructions in one WORD”
- ISA extensions for imaging
 - PA-RISC
 - x86 SSE1
- Beginning to use other technologies
 - GPU's
 - FPGA's
 - Game Chips

Address Space/Compilers - 1990

- Micro's move to a 64 bit Virtual Address
- System-Wide cache coherent interconnects
 - SCI
- Distributed Physical Memory
 - Shared Nothing
 - Shared Everything
- Compilers address
 - Distributed Memory
 - UPC
 - InterProcedural Analysis
 - Rice University

1990's

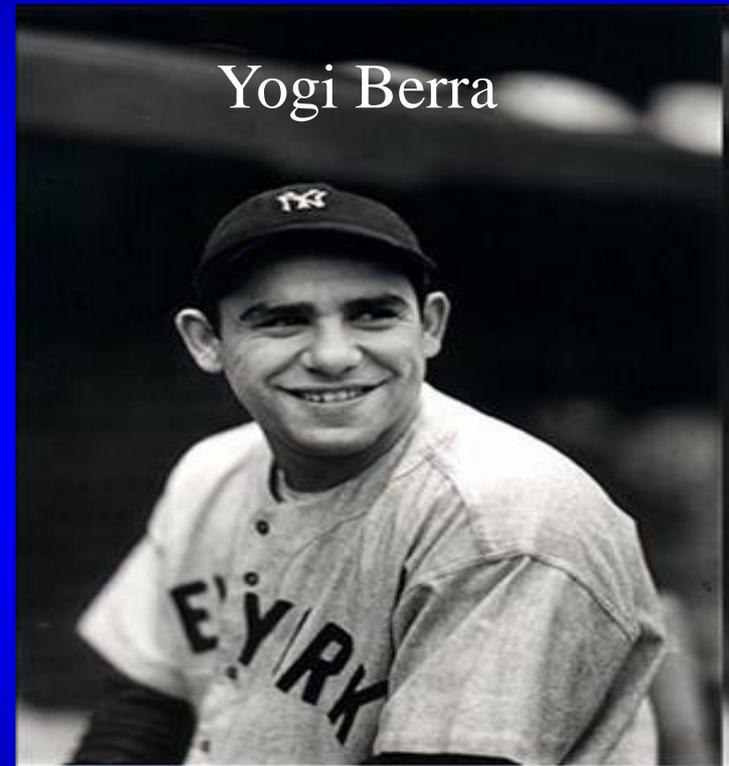
- Micro's Take Over
 - Cost of Fabs
 - Moore's Law INHIBITS new architectures
 - Cost of development escalates
 - Table stakes approach Billion Dollars
 - PC's begin to dominate desktop
 - ILP vs. Multi-Core
 - Will ILP help uniprocessor performance?
 - Cache blocking algorithms

1990's What we learnt

- Cost of semi-conductor Fabs and design of custom logic determine the dominant architectures
 - Need the volume to justify the cost of a Fab
 - Thus the beginning of the x86 Hegemony
- The most significant software technology is OPEN SOURCE
 - Linux begins to evolve
- There is no such thing as too much main memory or too much disk storage
- Compilers, with the proper machine state model, can produce optimized performance within a standard language structure

2000 & now

- Multi-Core Evolves
 - Many Core
 - ILP fizzles
- x86 extended with sse2, sse3, and sse4
 - -application specific enhancements
- Basically performance enhancements by
 - On chip parallel
 - Instructions for specific application acceleration
- Déjà vu – all over again – 1980's
 - Need more performance than micro
 - GPU, CELL, and FPGA's
 - Different software environment



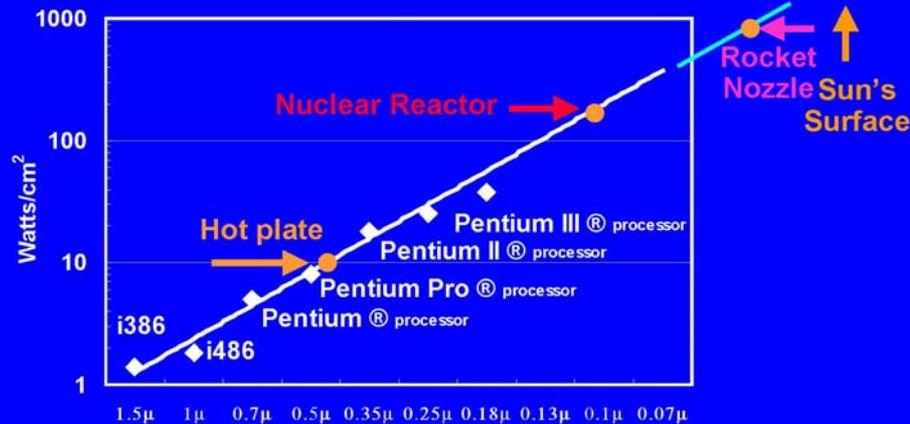
2000 Technology

- Moore's Law provides billions of transistors but clock speed static
 - Power $\sim C \cdot (V^{**2}) \cdot T +$ Leakage Power
- Main Memory technology not tracking cpu performance
 - Memory Wall
 - Cache Hierarchies
- Most significant software technology is the OPEN SOURCE movement
 - Easier to develop software using existing applications as a base.
 - OS and Compiler
 - Cluster aware frameworks



2000 Power Considerations

Power density continues to get worse



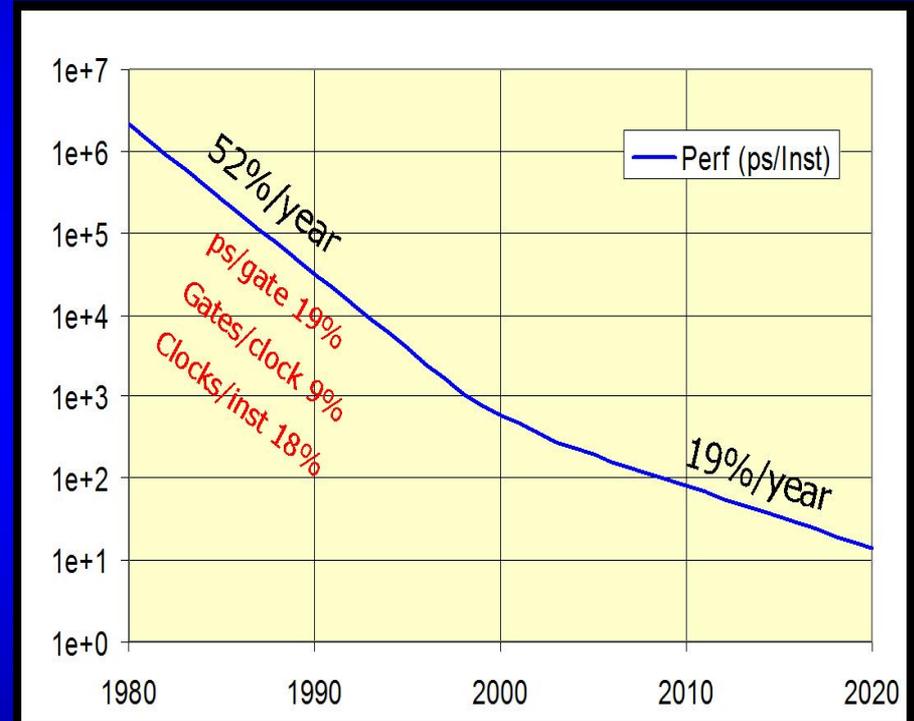
Surpassed hot-plate power density in 0.5µ
Not too long to reach nuclear reactor

Micro32
Fred Pollack

intel

2000 Design Technology

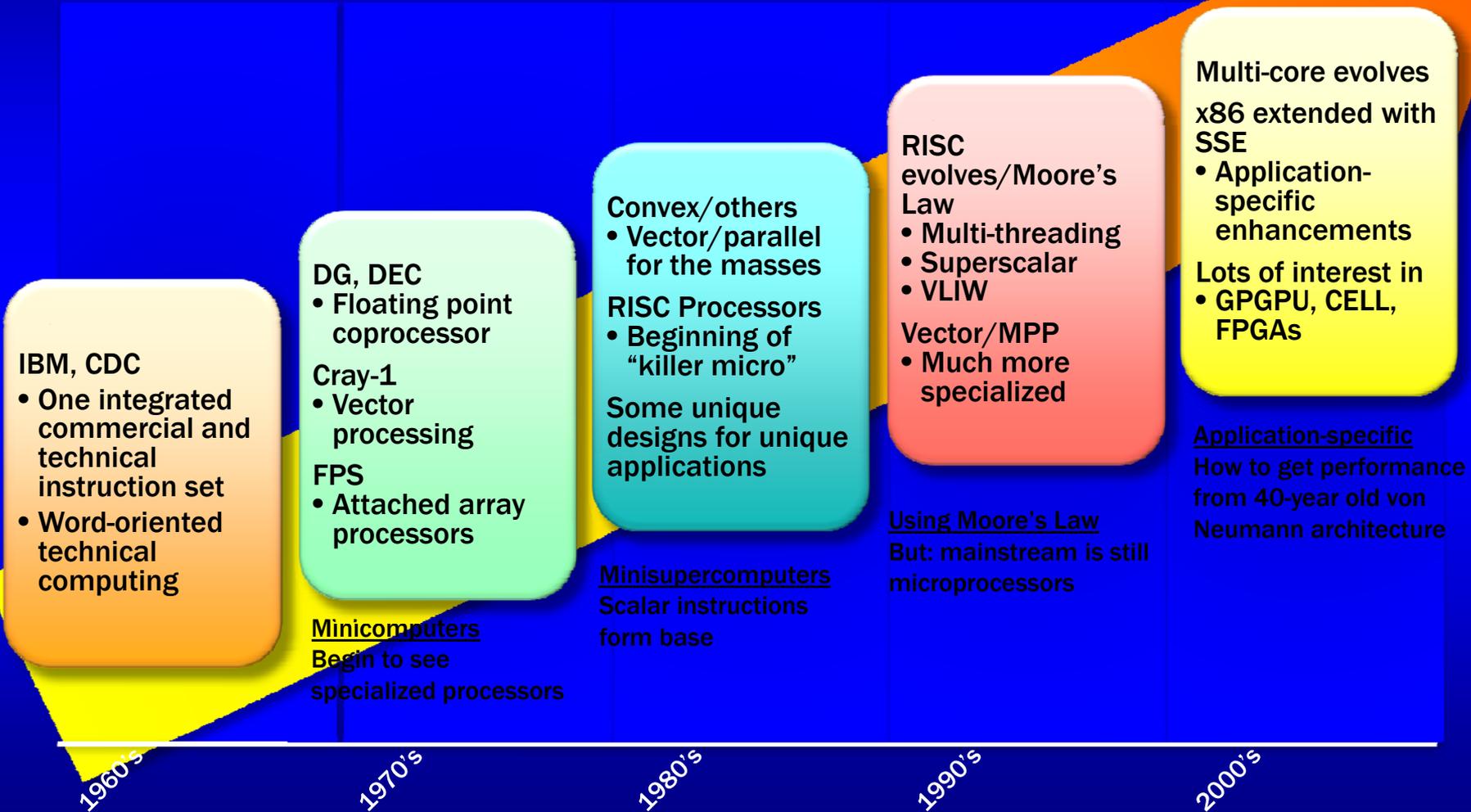
- New Arch ~ 2-3X die area of the last Arch but only Provides 1.5-1.7X integer performance of the last Arch
 - The Wrong Side of a Square Law
- Key Challenges for future Micro architectures
 - SIMD ISA extensions
 - Special Purpose Performance
 - Increased execution performance



Dally, ISAT Study – Aug 2001

Pollack Keynote Micro-32

The road to performance



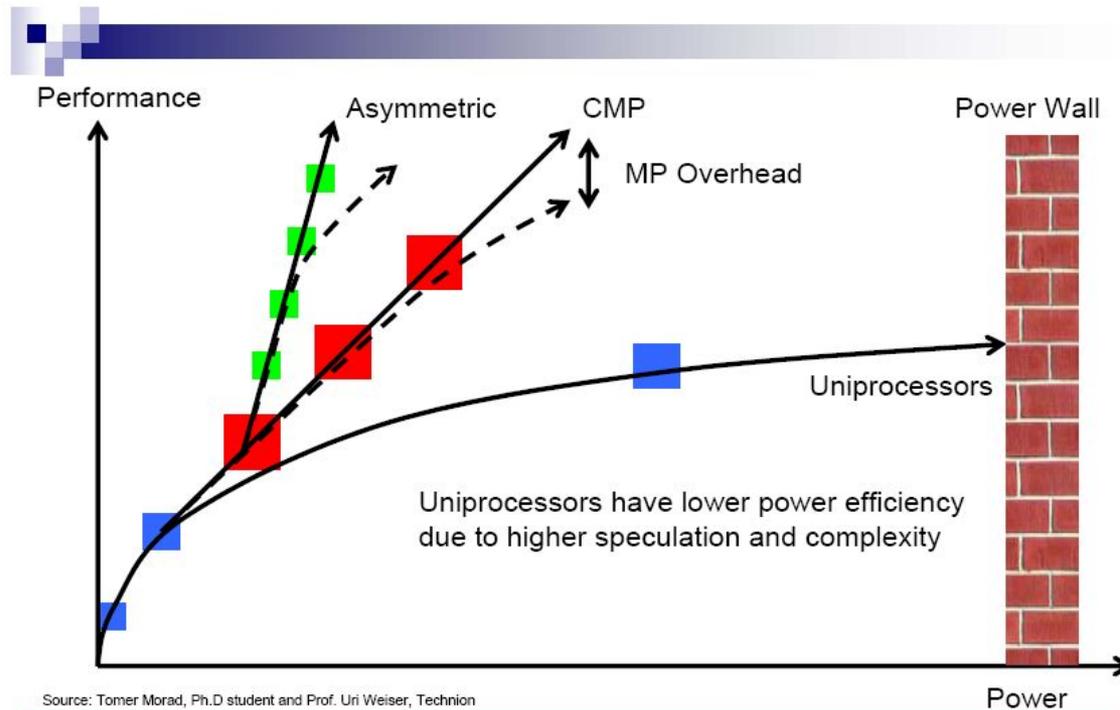
The standard desktop/server environment

- 64 bit virtual address space
- Multi-Core
- Cache coherent cores
- Gigabytes of ECC protected physical memory
- x86 Instruction Set
- Compilers
 - ANSI Fortran, C, and C++
 - Automatic Vectorizing and Parallelizing
 - One compiler used for application development
- One a.out (.exe) file
- I/O directly into application memory

What Next?

- Extend standard x86 architecture for application specific environments
 - Use the x86 as the canonical ISA (base level)
 - Implement cache coherency and share the same virtual and physical address space (QPI, HT)
 - Facilitates compiler global optimization
 - Permits more innovative physical memory design
- Provide compiler support and also provide time to market solutions
- Incremental hardware makes it easier to program
 - Consistent with the last 40 years

Basis of Discussion



BUT: The problem is still SW not hardware.
Programming asymmetric systems such as Cell is still a nightmare

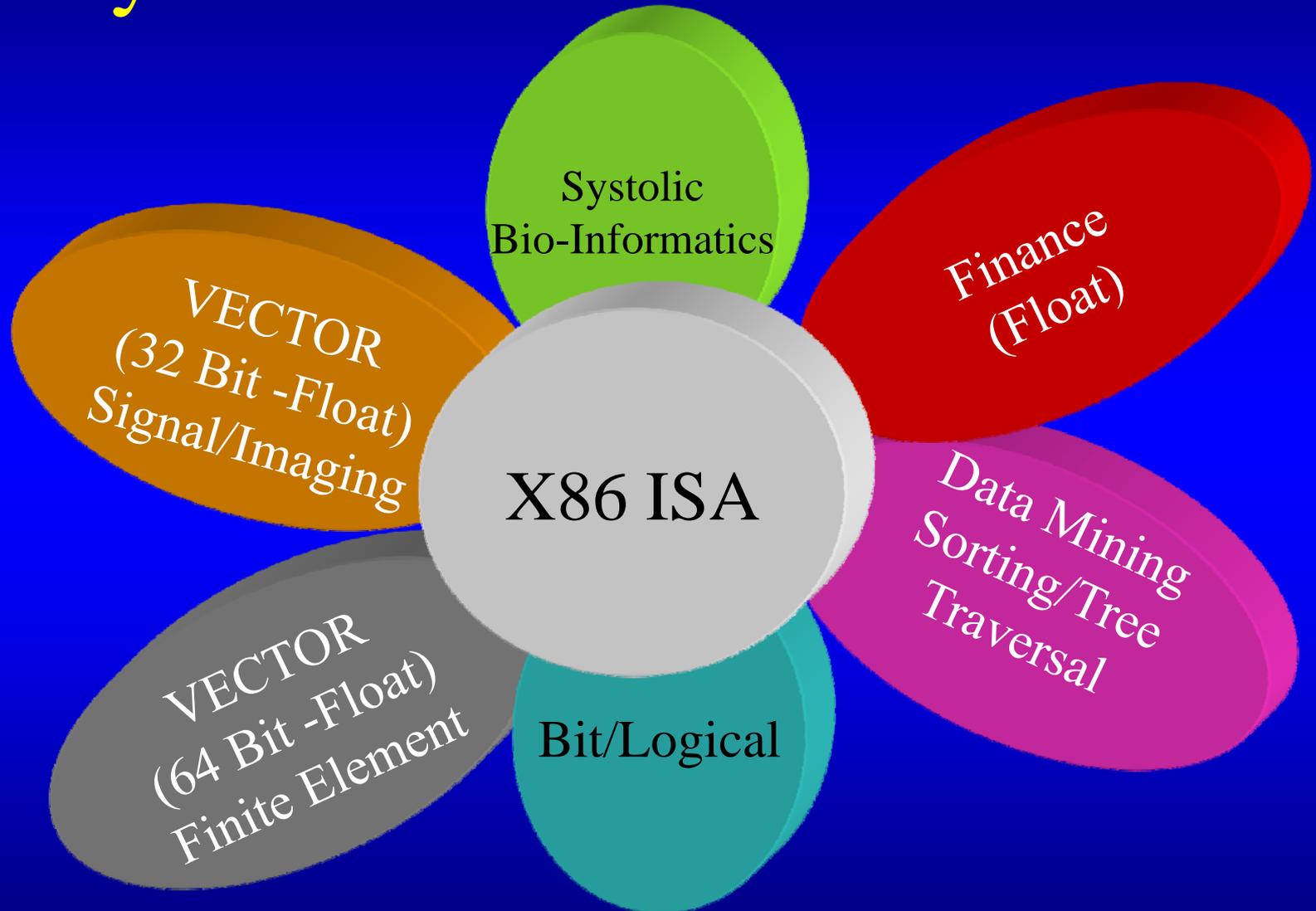
Dr. Avi Mendelson - Keynote speech, ICS 2007 ©

21

Asymmetric Processor

- Now is the time to refocus on uniprocessor performance
 - ILP does not deliver
 - Multi-Core does not help uniprocessor performance
- Serial Instruction sets and Cache Block Based Memory systems form the base level
 - Have to figure out how to deal with sparse datasets
- High Level Uniprocessor Semantics rather than ILP is needed
 - Use the transistors to build specific application functional units
 - Machine state appropriate to the computation
- One compiler generating both x86 and asymmetric instructions
- Highly interleaved Memory system optimized for:
 - Vector like memory access
 - Non-unity strides
 - Hashed Memory Lookups

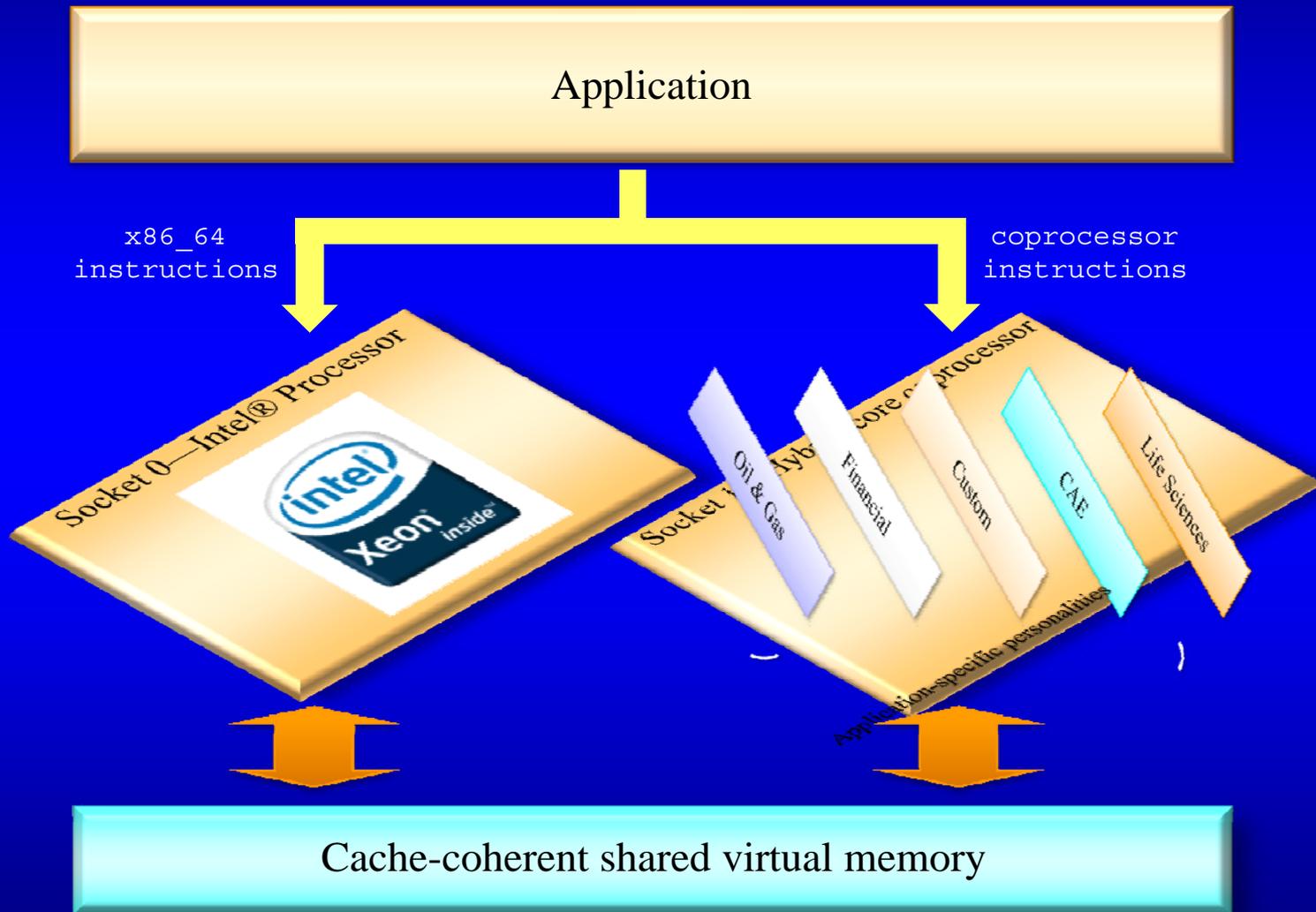
Asymmetric Processor - ISA



Asymmetric Processor - Compiler

- One Unified Compiler
 - x86 code generator
 - Multiple code generators for asymmetric processor ISA
 - Each extension presents a different machine state model
 - Benefits
 - Programmer Productivity Enhanced
 - Global Optimizations includes both the x86 core and asymmetric ISA
 - One compiler, as contrasted compiler for x86 and compiler for accelerator
- The past 40 years has taught us that ultimately the system that is easier to program will always win
 - Cost of ownership
 - Cost of development

Hybrid-Core Computing



The Convey Hybrid-Core Computer



- *Extends x86 ISA with performance of a hardware-based architecture*
- *Adapts to application workloads*
- *Programmed in ANSI standard C/C++ and Fortran*
- *Leverages x86 ecosystem*

What Next

- Is it time to go the next step in the address space?
 - 128 bit persistent
 - Network-Wide address space
 - IPv6
 - Use Moore's Law to make it easier to manage and access the world's data (not just local data)
 - TAKE SECURITY SERIOUSLY
 - 30 years ago workable security models were developed
- Compilers address hybrid distributed memory
 - PGAS
 - Cache coherent within SOCKET
 - Cache coherent (or not) external to socket
 - Augment/Replace MPI

And of Course Performance

