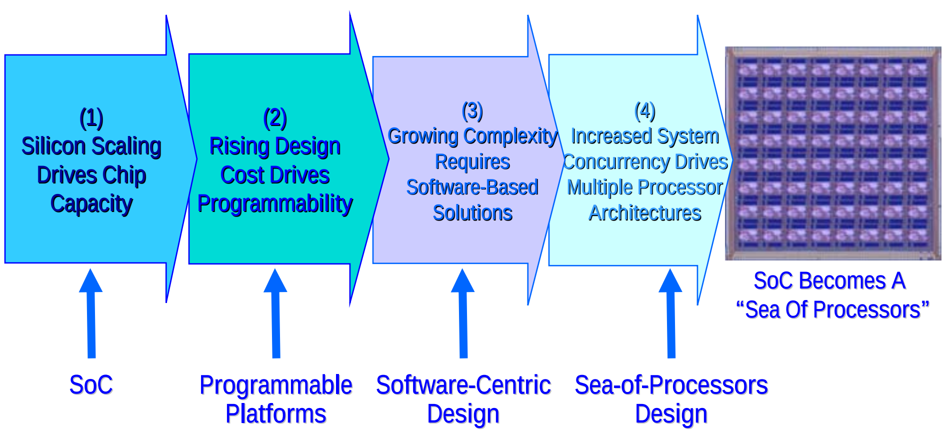
The Next Major Advance in Chip-Level Design Productivity

A. Richard Newton
University of California, Berkeley

Synopsys EDA Interoperability Developers' Forum
Santa Clara, CA
October 21st, 2004

Fundamental Drivers of Future Chip Designs

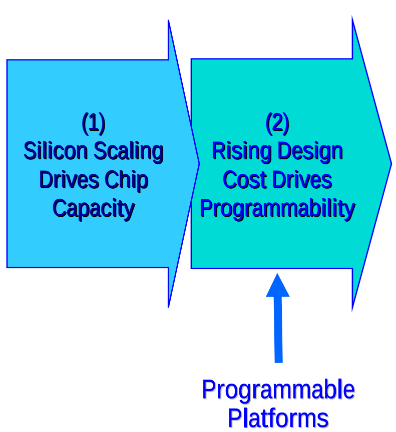


Source: Chris Rowen, Tensilica

Key Points

- The future mainstream building-block of electronic system-level design will present a (configurable) clocked synchronous Von Neumann programmer's model to the system-level application developer (ASIP or TSP)
- The majority of large silicon systems will consist of many such processors, connected in an asynchronous network
- These processors may be integrated on a single chip (CMP) and/or as a (possibly very large) collection of chips
- These conclusions lead to a number of critical design-technology research challenges and new business opportunities

Fundamental Drivers of Future Chip Designs

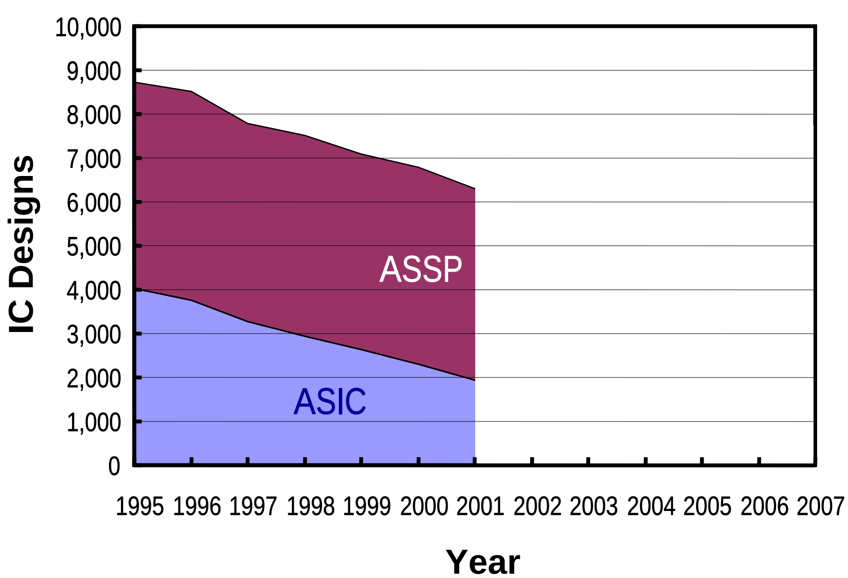


newton@coe.berkeley.edu

Conventional Arguments: The Changing Landscape of Design, Manufacture, and Test

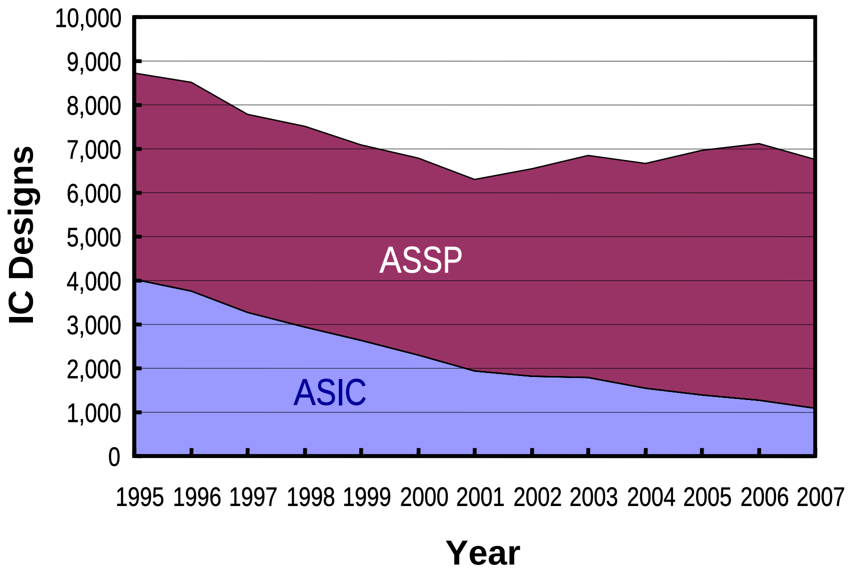
- The NRE cost of building a complex chip is O($20M) in 2004:
  - Fixed Costs (Masks, EDA Tools, IP Blocks, Diagnosis and Test)
  - Design Costs (Team Size, Verification, Timing Closure)
  - Opportunity Cost (Predictability Of Design Time, Chip Characteristics, and Manufacturing Reliability)
- Need either a single, huge market or ability to address multiple application variants and system product generations with same physical device
- Programmability brings adaptability to SoC. Two popular forms:
  - Field-programmable logic, based on low-level logic and interconnect hardware configuration, from hardware description languages (e.g. Verilog), and O(20-40) times slower/larger/more power than equivalent custom logic
  - Processors, based on sequential instruction programming from high-level languages (mostly C/C++ plus limited assembly code), and O(10-1,000) times slower/more power than equivalent custom logic

Total IC Designs



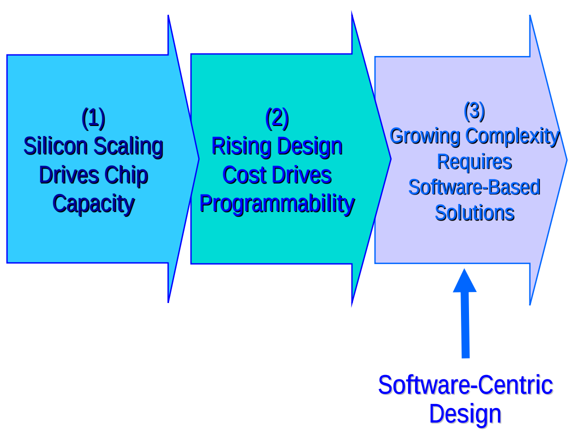
Source: Handel Jones, IBS, October 2002

Total IC Designs



Source: Handel Jones, IBS, October 2002

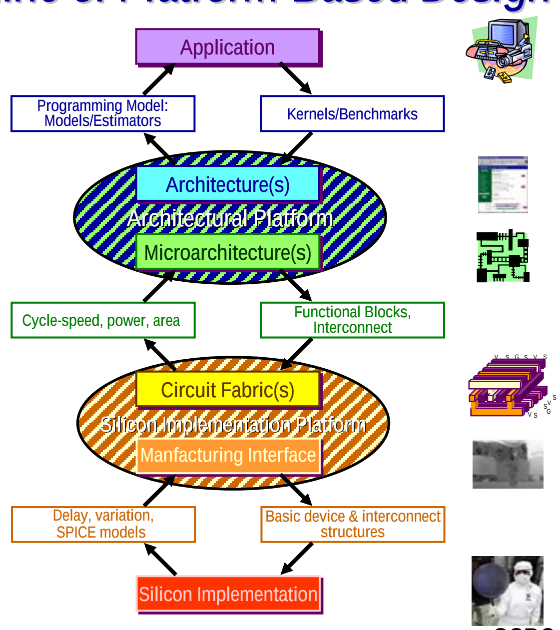
Fundamental Drivers of Future Chip Designs



Growing Complexity Drives Software-Centric Design

- Growing product complexity driven by both market competition in end products and growing capability of silicon
- Complexity of the external application domain makes accurate specification of application domain almost impossible
- Example: voice codec ITU document size
  - G.711 (1988): 190KB, G.726 (1990): 290KB, G.729 (1996): 2.1MB
- Growing complexity means:
  1. Greater design time
  2. Greater bug risk and bug fix effort
  3. Greater diversity of customer requirements
  4. Greater exposure to changing standards
- Software, today written in high-level languages (e.g. C/C++) is the best understood, most scalable means of developing and debugging complex functions.

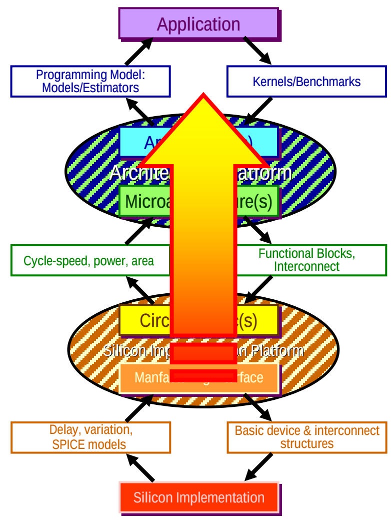
A Discipline of Platform-Based Design



GSRC Review, Sep. 2001

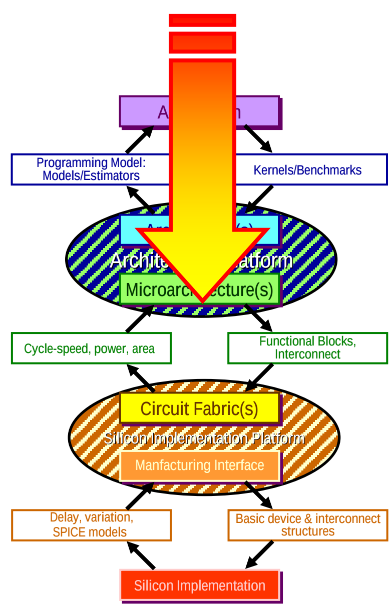
Today: "Given a Processor Chip (and it's Accelerators)..."

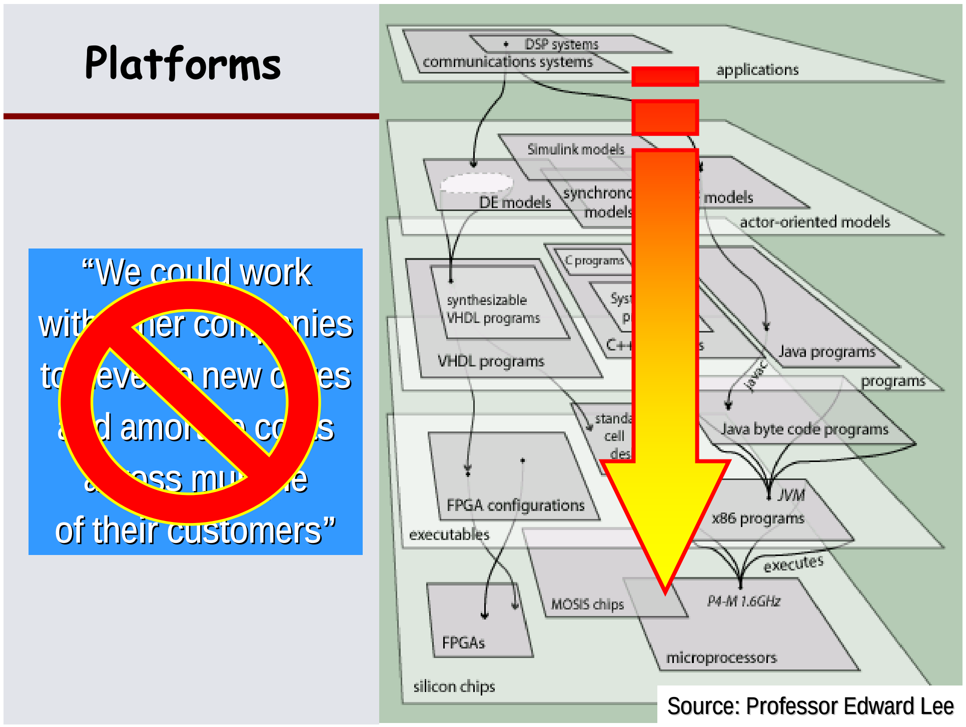
- I get to choose from existing hardware product offerings...
- Then I decide what software components I have or can find, for OS, for IO, for data conversion, etc., then I port what I must, and I plan to write the rest.
- A "Hardware-up" methodology



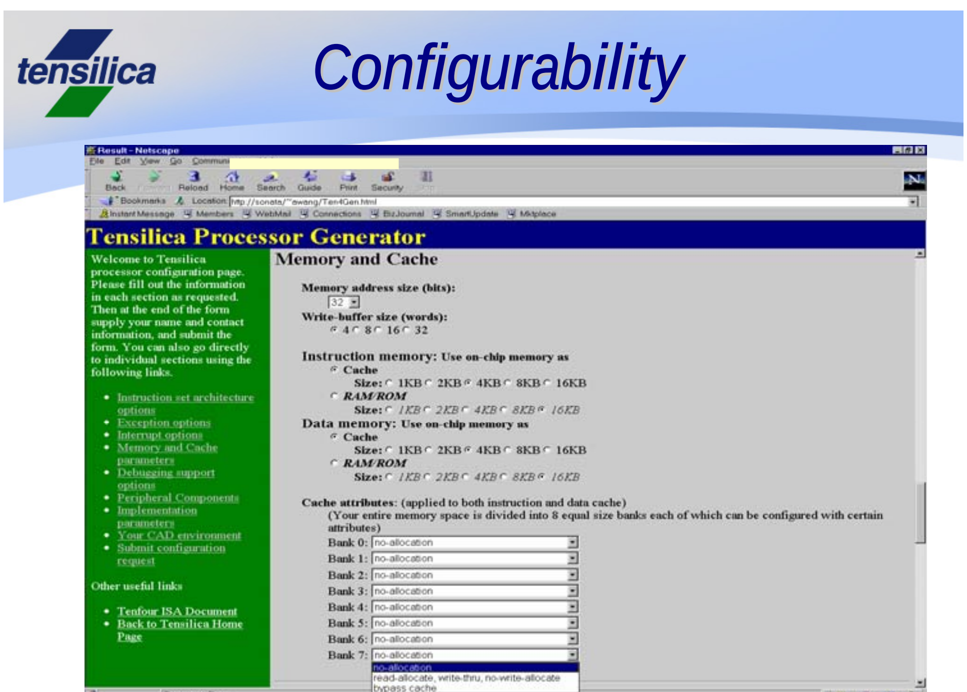
Tomorrow: "Given an Application, and a software development environment..."

- I get to specify the characteristics of a programmable hardware core or sea-of-cores...
- Then I decide what accelerators/additional instructions I might need, select IP from libraries, and use them to design a chip for this class of application
- A "Software-down" methodology

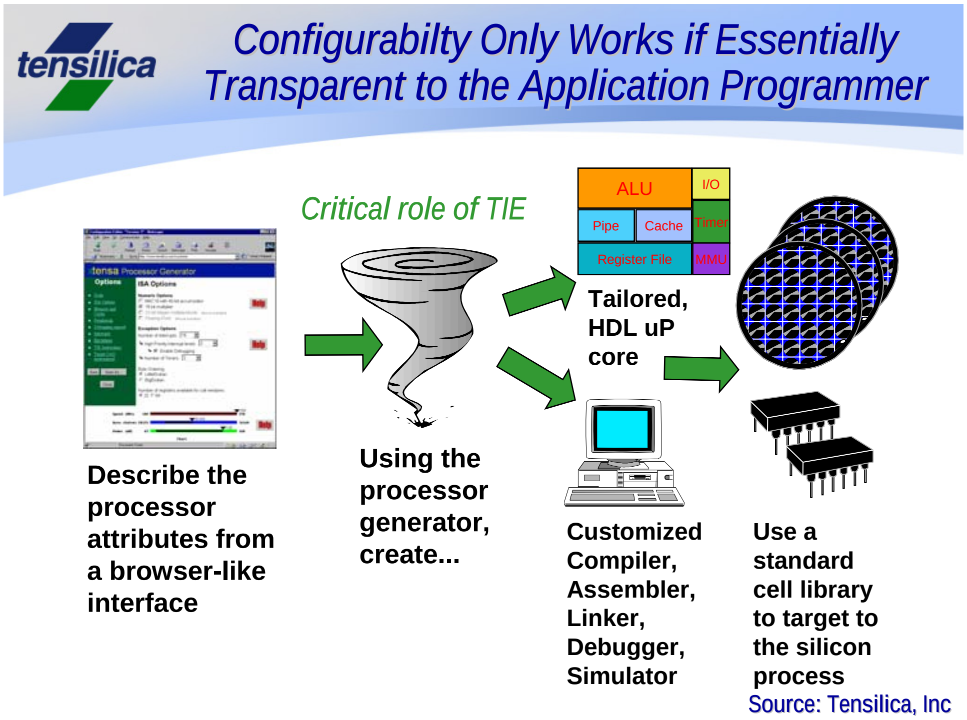


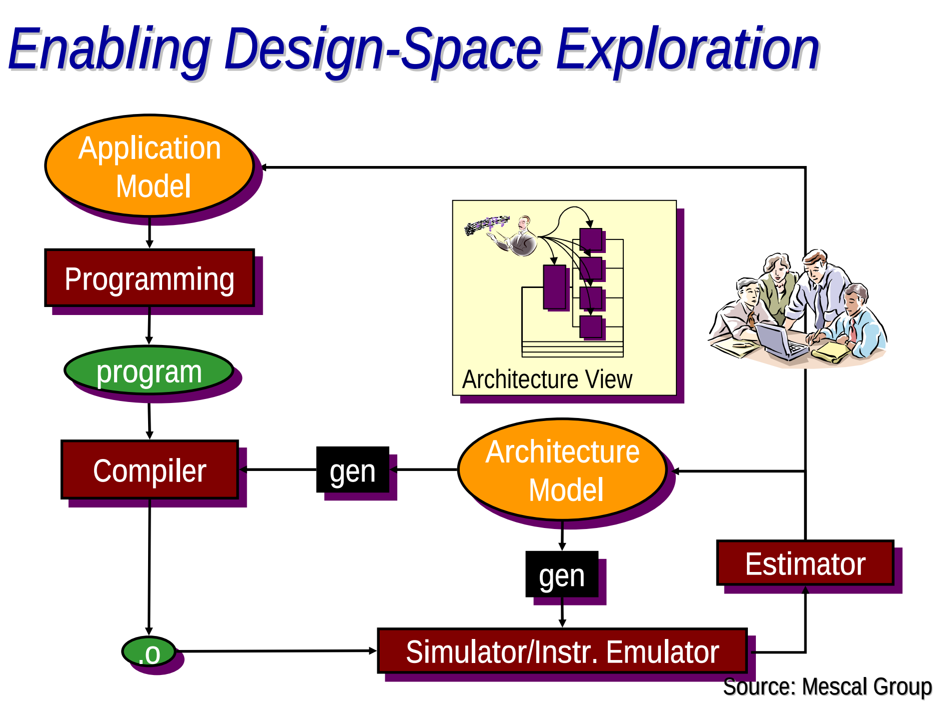


Source: Professor Edward Lee



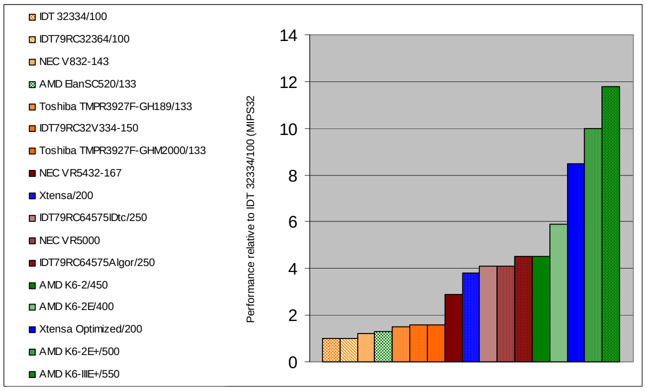
Source: Tensilica, Inc

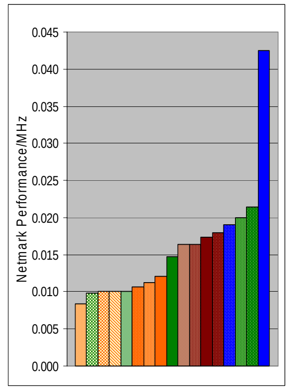

tensilica
Configurabilty Only Works if Essentially Transparent to the Application Programmer
Critical role of TIE
ALU
I/O
Pipe
Cache
Timer
Register File
MMU
Tailored, HDL uP core
Describe the processor attributes from a browser-like interface
Using the processor generator, create...
Customized Compiler, Assembler, Linker, Debugger, Simulator
Use a standard cell library to target to the silicon process
Source: Tensilica, Inc


Enabling Design-Space Exploration
Application Model
Programming
program
Architecture View
Compiler
gen
Architecture Model
gen
Estimator
.o
Simulator/Instr. Emulator
Source: Mescal Group

EEMBC Networking Benchmark

- Benchmarks: OSPF, Route Lookup, Packet Flow
- Xtensa with no optimization comparable to 64b RISCs
- Xtensa with optimization comparable to high-end desktop CPUs
- Xtensa has outstanding efficiency (performance per cycle, per watt, per mm^2)
- Xtensa optimizations: custom instructions for route lookup and packet flow



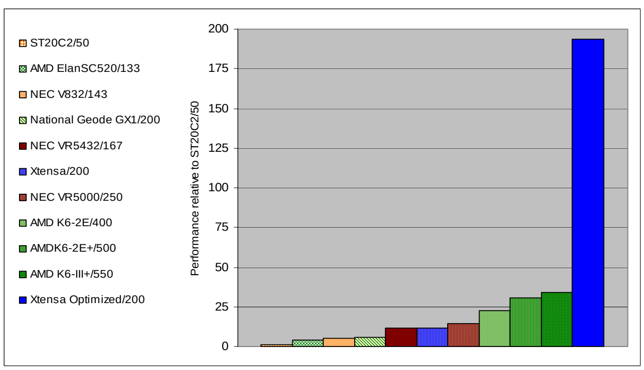


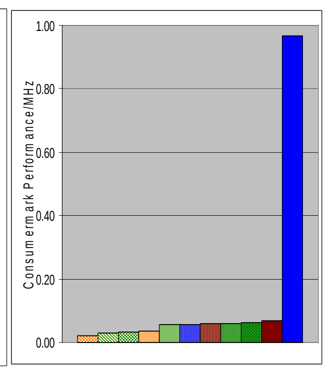
Colors: Blue-Xtensa, Green-Desktop x86s, Maroon-64b RISCs, Orange-32b RISCs

Source: Tensilica, Inc

EEMBC Consumer Benchmark

- Benchmarks: JPEG, Grey-scale filter, Color-space conversion
- Xtensa with no optimization comparable to 64b RISCs
- Xtensa with optimization beats all processors by 6x (no JPEG optimization)
- Xtensa has exceptional efficiency (performance per cycle, per watt, per mm^2)
- Xtensa optimizations:custom instructions for filters, RGB-YIQ, RGB-CMYK



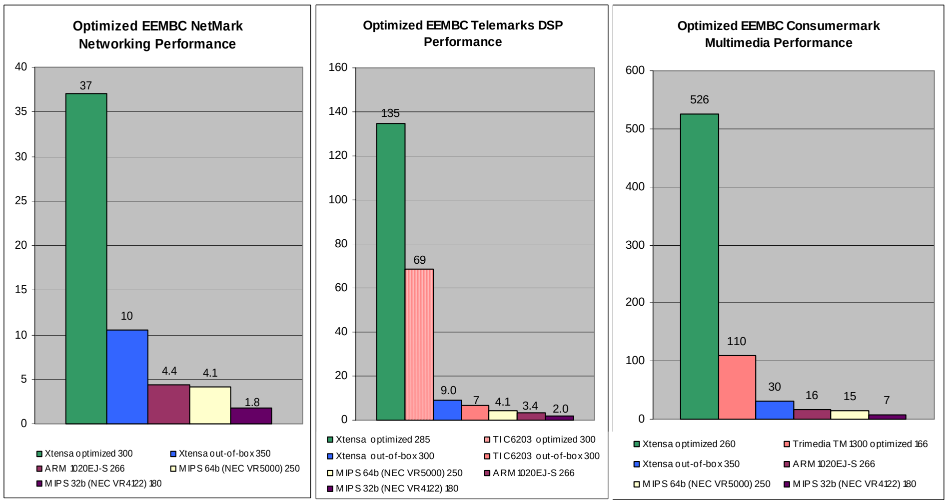


Colors: Blue-Xtensa, Green-Desktop x86s, Maroon-64b RISCs, Orange-32b RISCs

Source: Tensilica, Inc

newton@coe.berkeley.edu

Configurable Processors Lead Across Wide Application Range



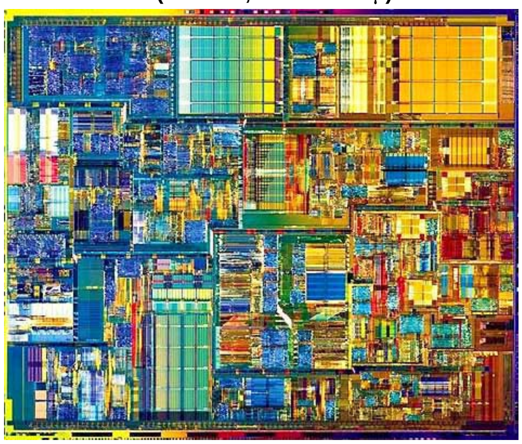
Source: Chris Rowen, Tensilica


2003: Radical Specialization
Around a Common Core
Architectural Platform


2003: What Can We Learn About a
Maturing Industry?
Radically Different
Core Architectures?
Traditional ASIC?
Modern DSP Architectures—
Jack of All Trades?

Size Determines Cost and Power

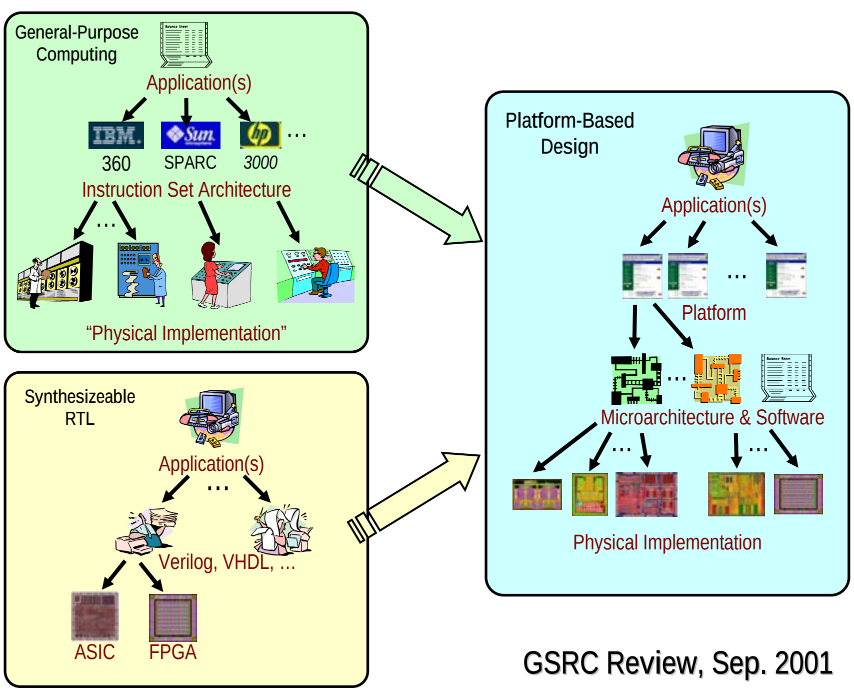
Intel
Pentium 4
(145mm², 50W in 0.13μ)



Tensilica
Xtensa processor
(1.5mm², 0.1W in 0.13μ)

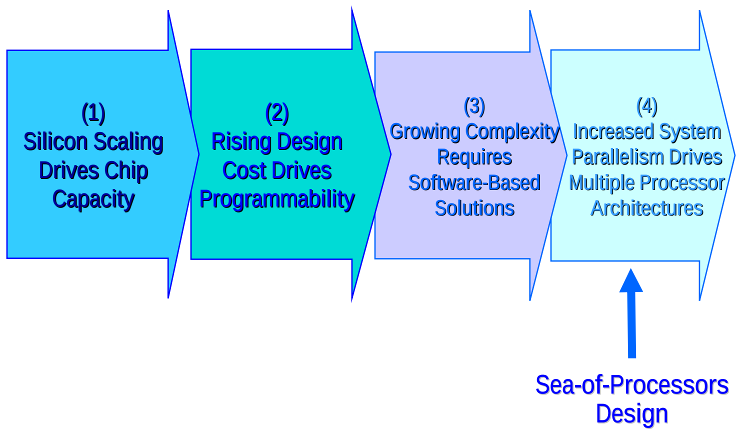
100x smaller
500x lower power
often faster

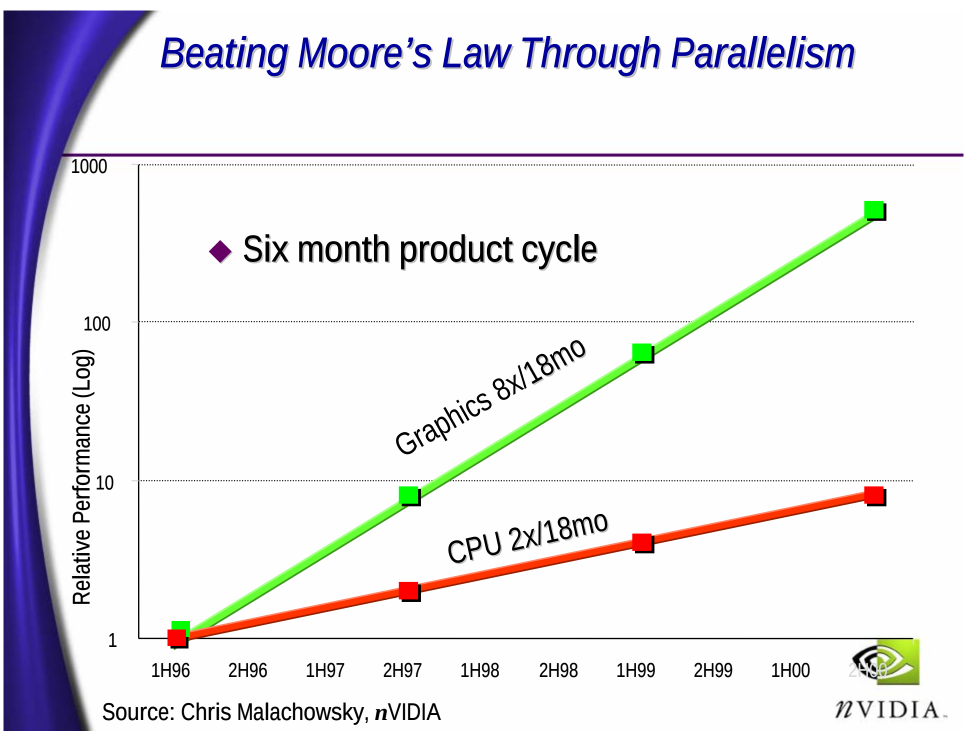
Source: Chris Rowen, Tensilica



GSRC Review, Sep. 2001

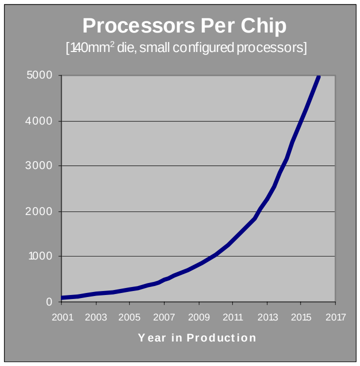
Fundamental Drivers of Future Chip Designs

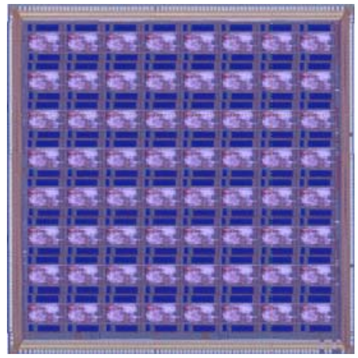




"The SOC Processor is the New Transistor"

Prof. David Patterson, UC Berkeley



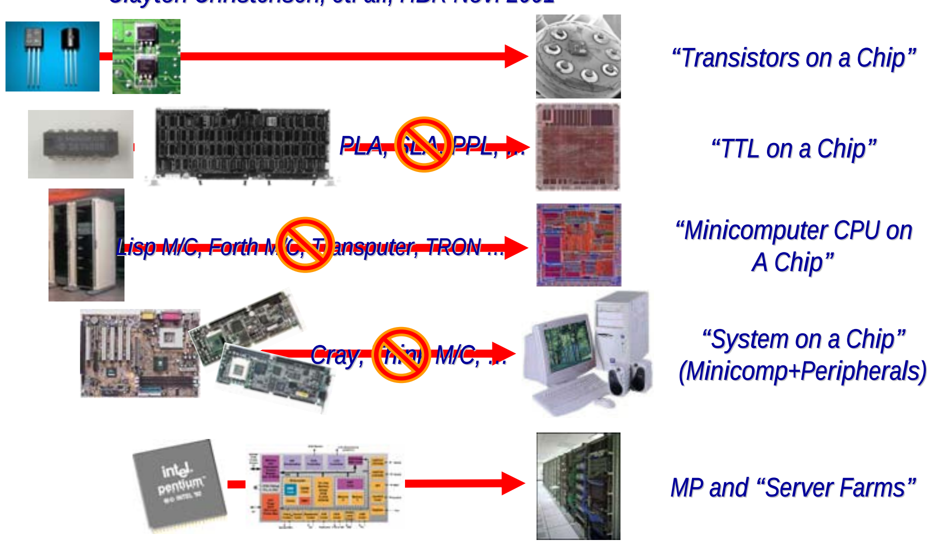


Trend: Pervasive use of application-specific processors as basic building block: *The Sea of Processors*

Observation: Data-intensive applications often have high parallelism, so large numbers of processors efficiently utilized

"Great Companies Take What We Do Today and Do it Better"

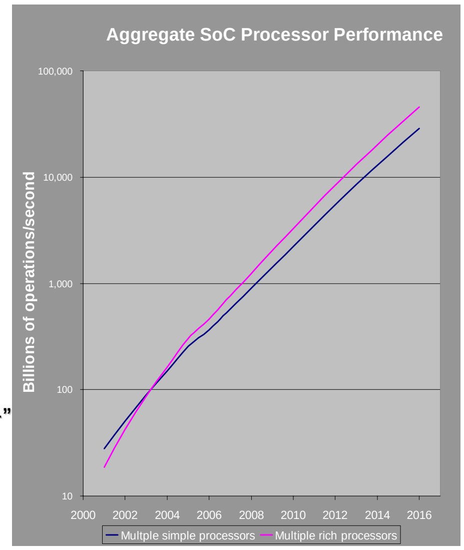
Clayton Christensen, et. al., HBR Nov. 2001



"Chip-Level Multiprocessors (CMP's)"

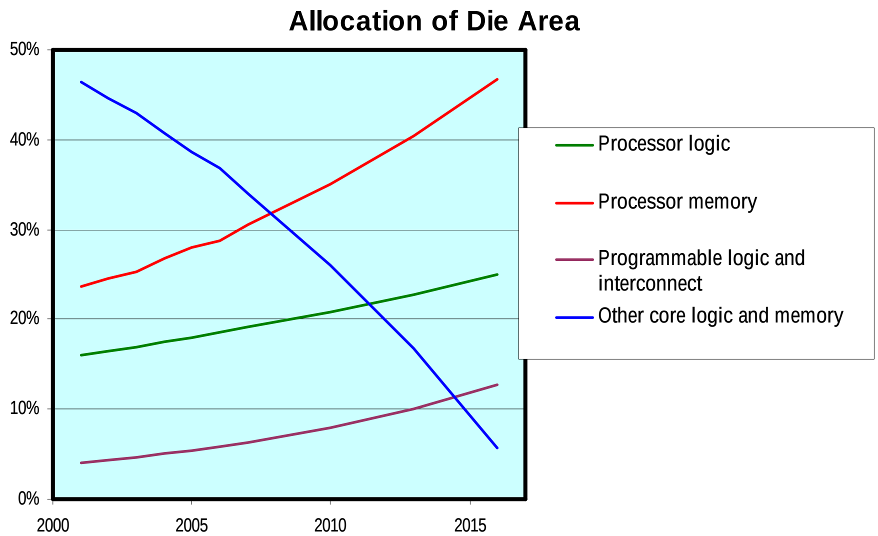
Rowen's Law of SoC Processor Scaling

- Part 1: Processors/chip:
  Up to >30% year growth
- Part 2: Programmable operations/sec:
  65% per year growth
- By 2010:
  >1000 processors/chip
  >> 10^{12} operations/sec
- *Key enablers:*
  - Automated processor creation from "C/C++" application
  - Automated multiple processor model and interconnect generation



Source: Chris Rowen, Tensilica

Implications of Rowen's Law



Implications of Rowen's Law

1. Automated processor design
   - Range of architectural styles from tiny to high ILP
   - Automatic instruction set generation from C/C++
2. Concurrent programming innovation
   - Distributed programming models
   - Novel communication networks (asynchrony, application-specific topologies, automated optimization of cost and bandwidth)
3. System design methodology
   - Rapid software-centric MP system architecture exploration
   - Complete hardware/software co-generation
   - Tight architecture ⇔ physical design tool coupling
4. Allocation of silicon area
   - Processor (and its memory) dominates
   - Programmable interface and interconnect
   - Non-processor logic shrinks
5. Cost of processors
   - Raw logic for base processor: millicents
   - Total cost with memory: cents

"It's All About Concurrency"

- A global, synchronous model no longer works: neither in hardware nor in software
- The majority of errors most difficult to detect and eliminate in modern software development are due to concurrency issues: from Windows XP to Wind River
- We are at the beginning of a revolution in embedded runtime support. e.g. Sun Jini, COM+, Universal Plug-and-Play, Ninja
- Should consider the verification issue up front, and use a verifiable underlying model for concurrency

Key Points

- The future mainstream building-block of electronic system-level design will present a (configurable) clocked synchronous Von Neumann programmer's model to the system-level application developer
- The majority of large silicon systems will consist of many such synchronous processors, connected in an asynchronous network
- These processors may be integrated on a single chip (CMP) and/or as a (possibly very large) collection of chips
- These conclusions lead to a number of critical design-technology research challenges and new business opportunities

Summary

- *No More Debate!* ... The future of system-level design is CMP/MCMP, not {SS, VLIW, XYZ...) so let's get on with it.
- The most successful systems will define a Programmer's Model that:
  - Supports one or more clocked sequential processors integrated (asynchronously) on a chip
  - Is natural for application developers
  - Supports task-level processor customization (mask level or field programmable)
  - Protects task/application software development investment as much as possible
- Such systems must subsume both hardware implementation/assembly and core software tasks in a single, integrated development environment that is viewed "from the top"
  - It is about methodology and tools, not SIP-centric
  - Will automatically support very high levels of design reuse
  - The biggest research challenge is how to implement concurrent computation on and among processors in a reliable and verifiable way, while preserving as much efficiency as possible (speed, power, cost, etc.)


Thank You!
CALIFORNIA
College of Engineering
GO BEARS!!!