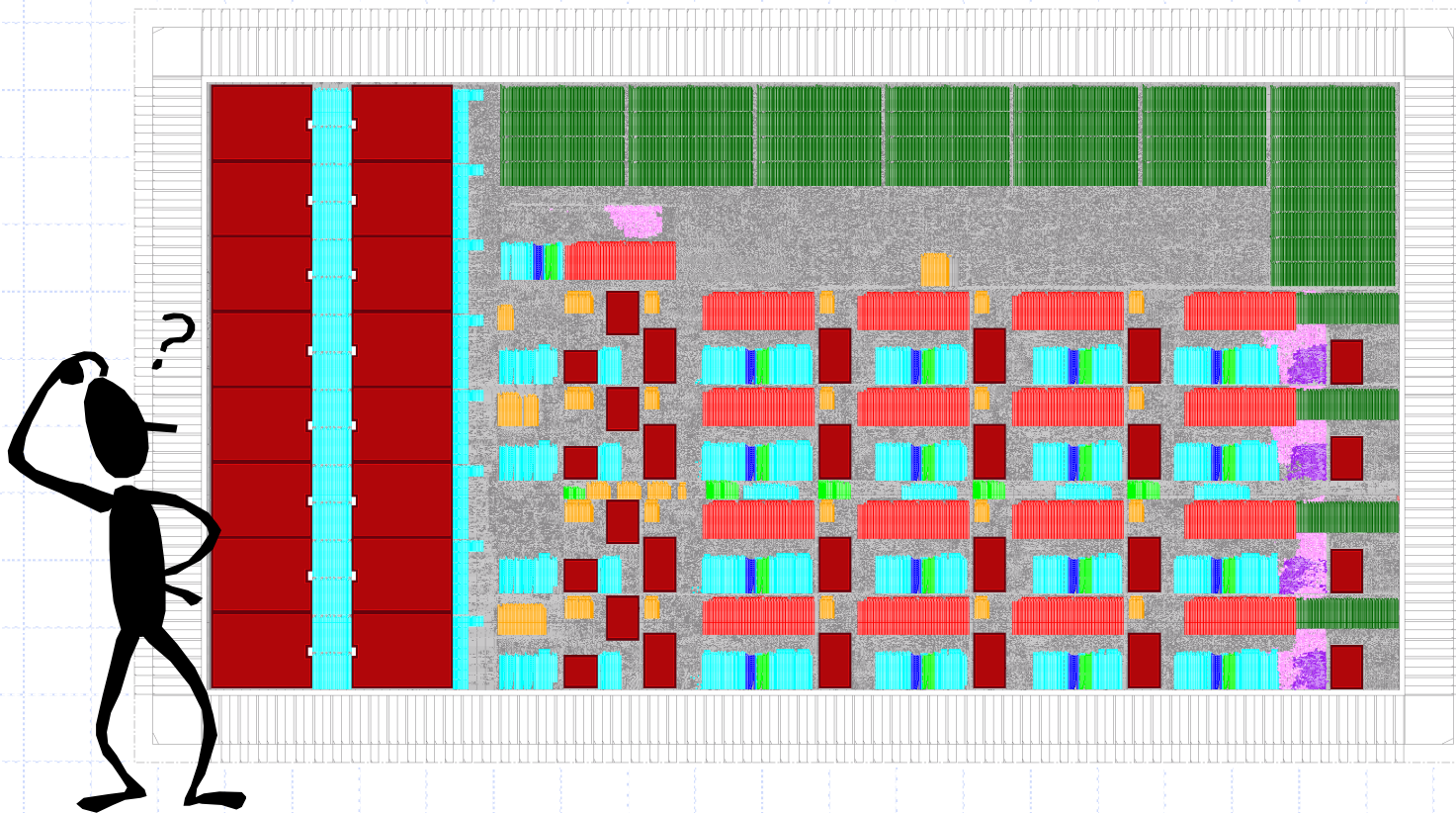


6.375: Complex Digital Systems



Arvind
Computer Science and AI Lab, MIT

Why take 6.375

- ◆ Something new and exciting as well as useful
- ◆ Fun: Design systems that you never thought you could design in a course
 - made possible by large FPGAs and Bluespec

You will also discover that is possible to design complex digital systems with little knowledge of circuits



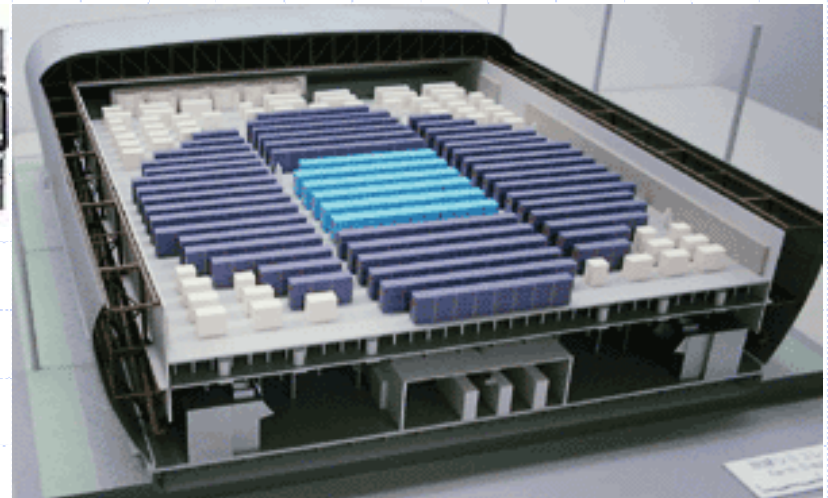
New, exciting and useful ...

Wide Variety of Products Rely on ASICs

ASIC = Application-Specific Integrated Circuit

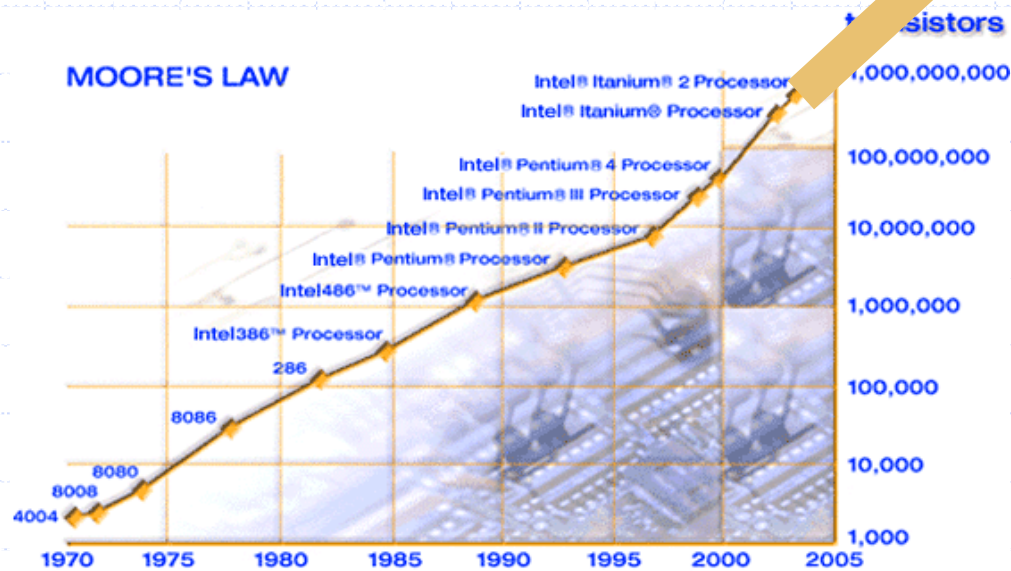
Wide Variety of Products Rely on ASICs

ASIC = Application-Specific Integrated Circuit



What's required?

ICs with dramatically higher performance, optimized for applications

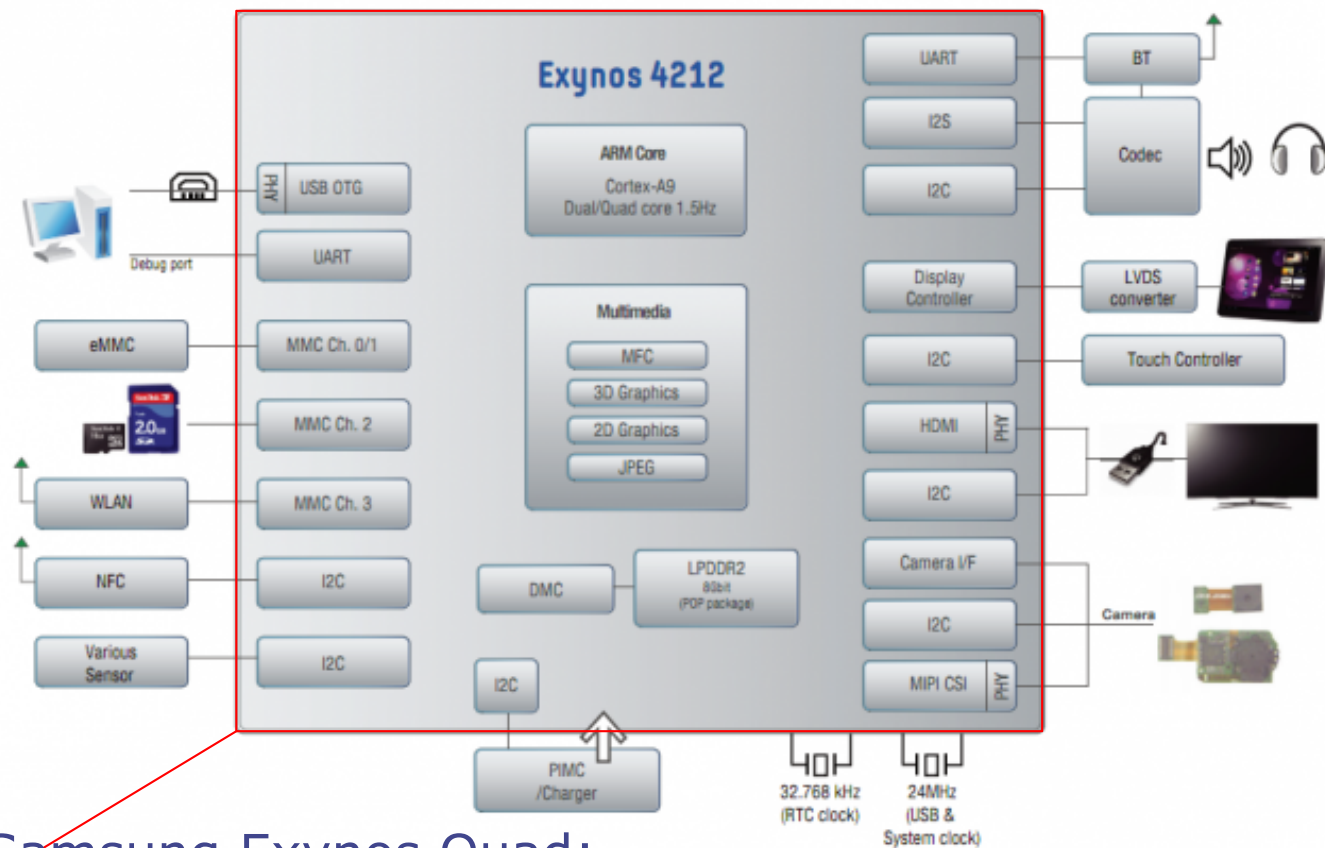


Source: <http://www.intel.com/technology/silicon/mooreslaw/index.htm>

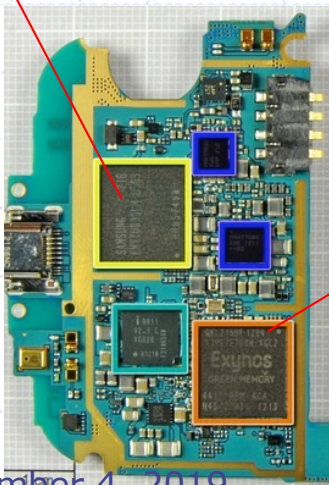
and at a

- ◆ size and power to deliver mobility
- ◆ cost to address mass consumer markets

Cell Phones: Samsung Galaxy S III April 2012



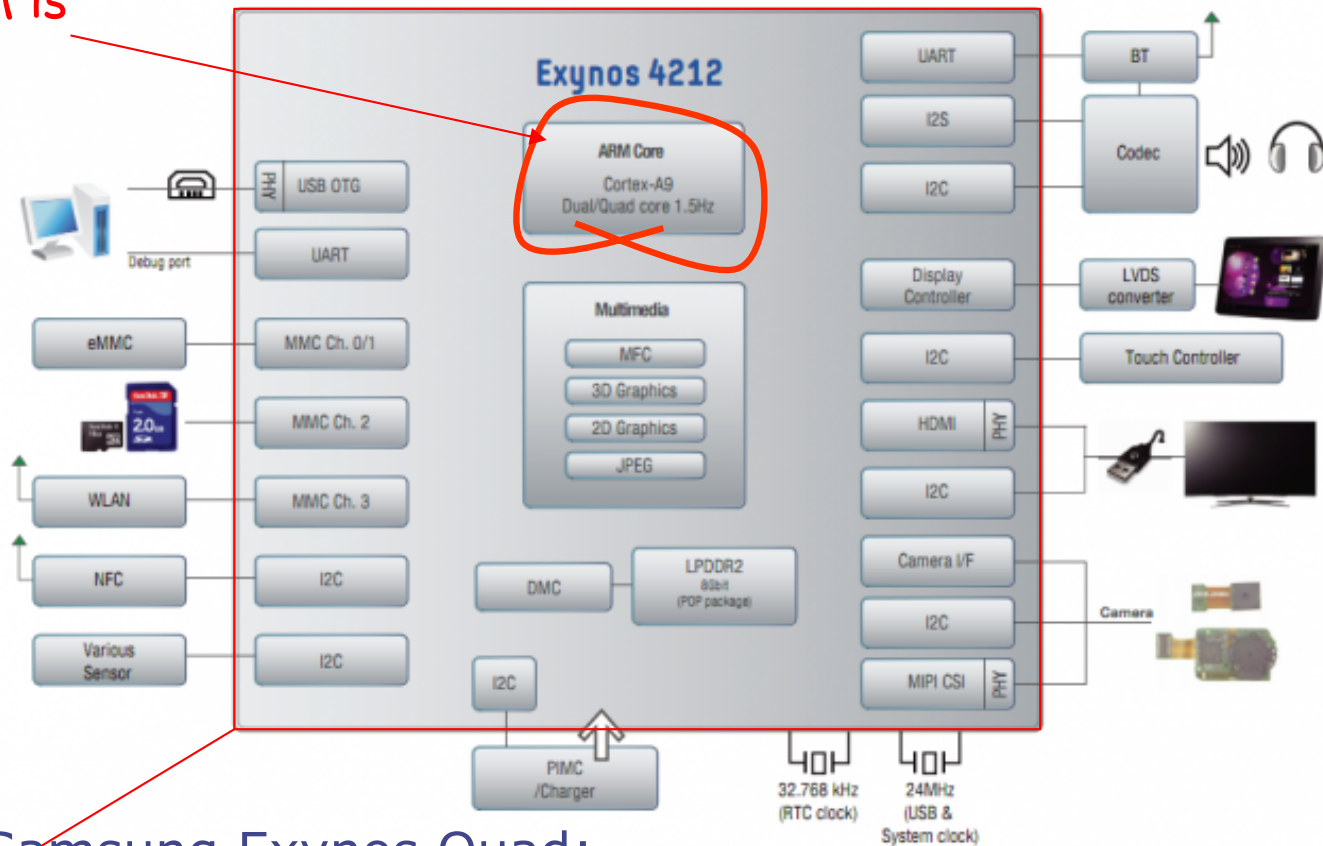
16GB
NAND flash



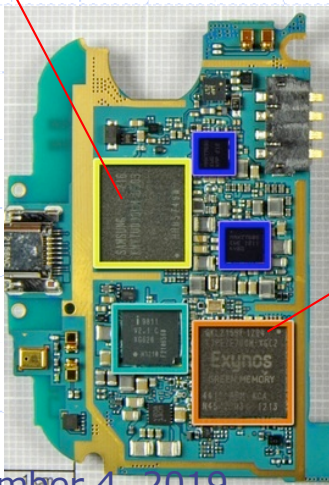
Samsung Exynos Quad:
 - quad-core A9
 - 1GB DDR2 (low power)
 - Multimedia processor

Cell Phones: Samsung Galaxy S III April 2012

Quad core ARM is just one of the complex blocks



16GB NAND flash

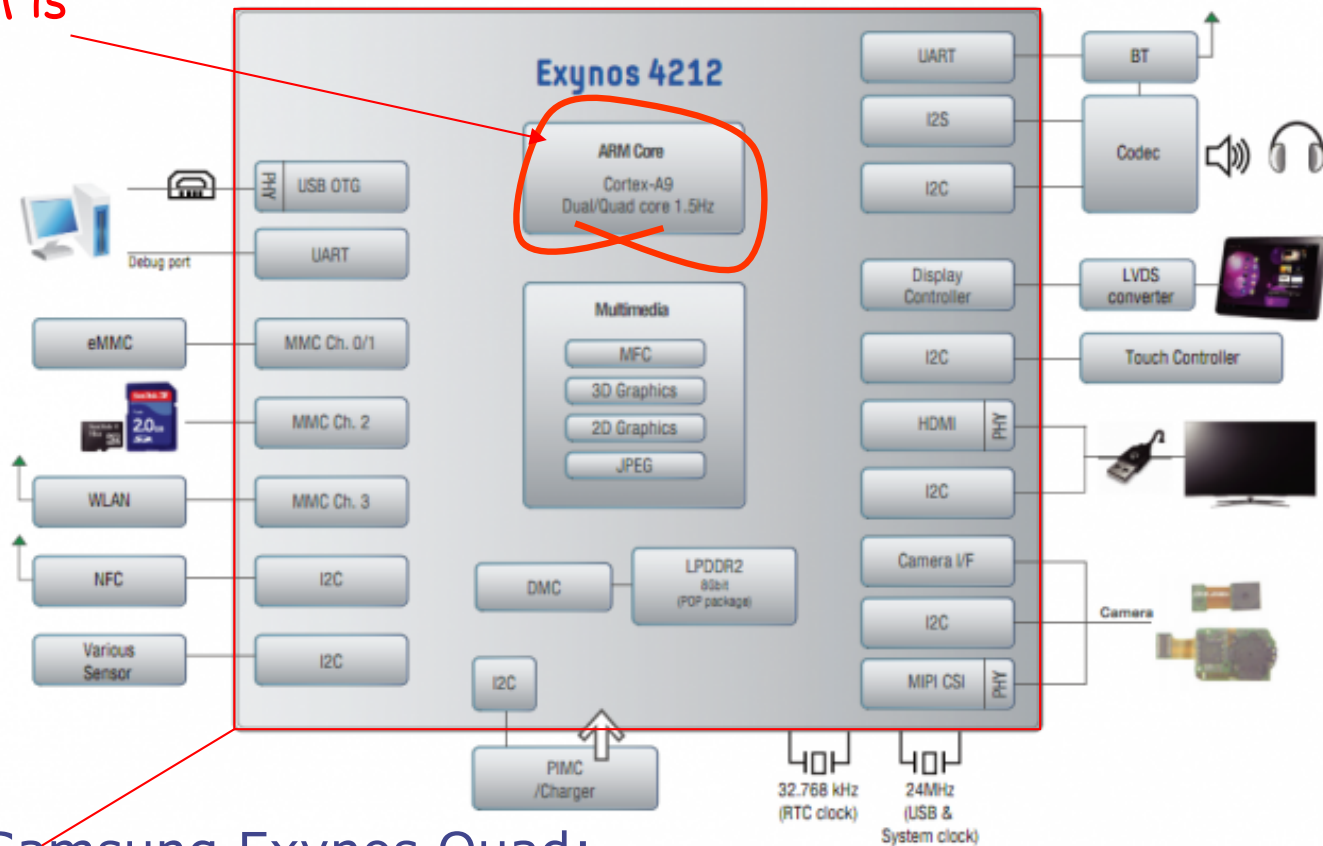


Samsung Exynos Quad:

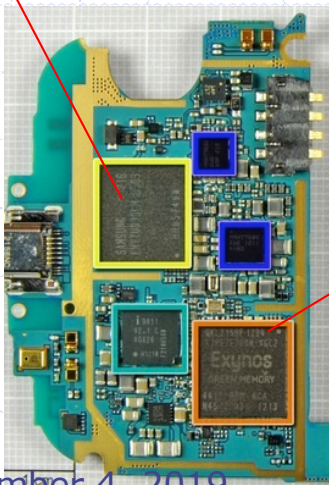
- quad-core A9
- 1GB DDR2 (low power)
- Multimedia processor

Cell Phones: Samsung Galaxy S III April 2012

Quad core ARM is just one of the complex blocks



16GB NAND flash



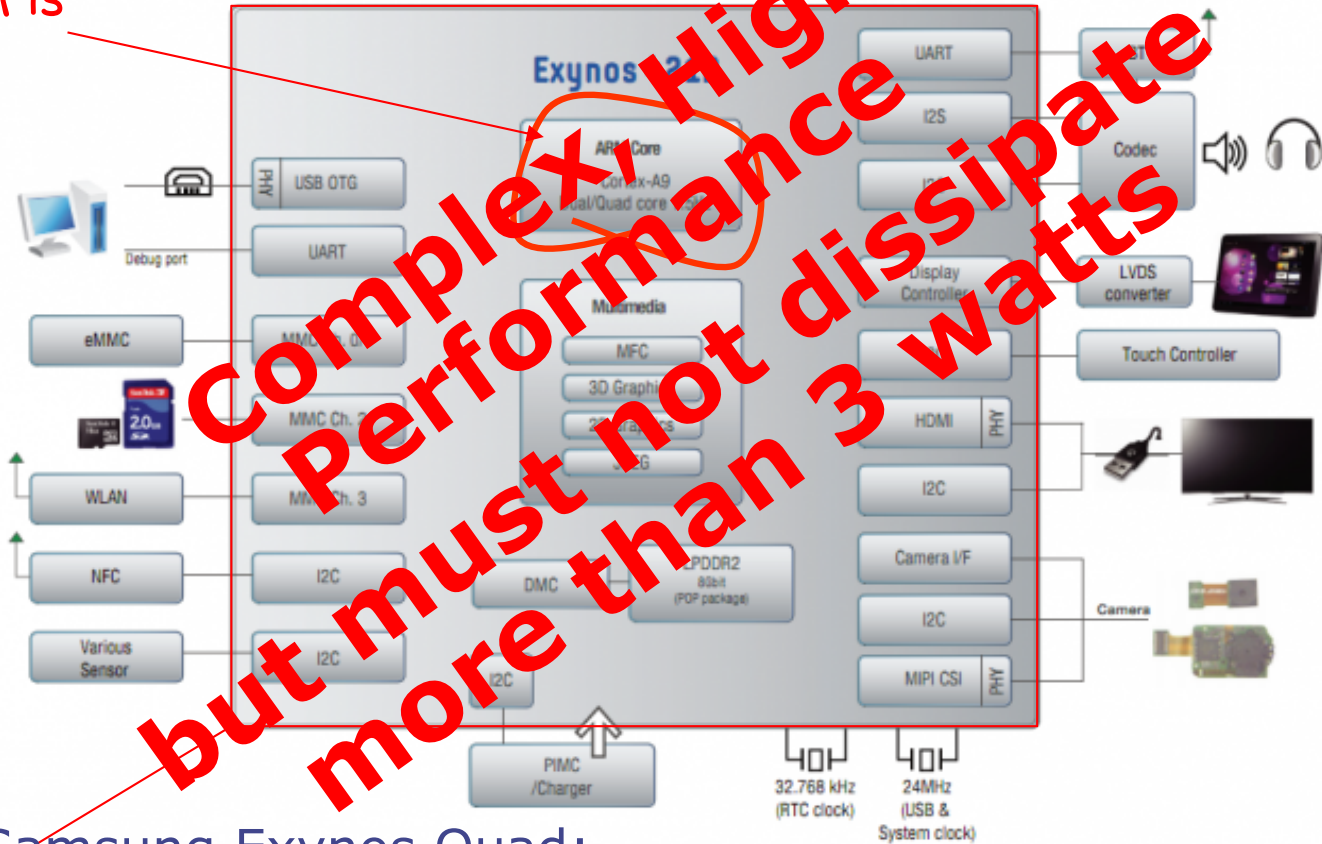
Samsung Exynos Quad:

- quad-core A9
- 1GB DDR2 (low power)
- Multimedia processor

power consumption <1W

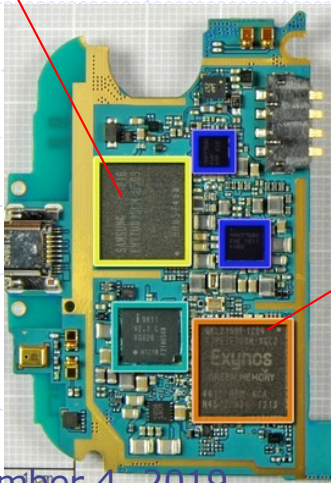
Cell Phones: Samsung Galaxy S III April 2012

Quad core ARM is just one of the complex blocks



Complex, High Performance but must not dissipate more than 3 watts

16GB NAND flash



Samsung Exynos Quad:

- quad-core A9
- 1GB DDR2 (low power)
- Multimedia processor
- ...

power consumption <1W

Server microprocessors also need specialized blocks

- ◆ compression/decompression
- ◆ encryption/decryption
- ◆ intrusion detection and other security related solutions
- ◆ Dealing with spam
- ◆ Self diagnosing errors and masking them
- ◆ ...

Real power saving implies specialized hardware

- ◆ H.264 video decoder implementations in software vs. hardware
 - the power/energy savings could be 100 to 1000 fold

Real power saving implies specialized hardware

- ◆ H.264 video decoder implementations in software vs. hardware
 - the power/energy savings could be 100 to 1000 fold

but our mind set is that hardware design is:

Real power saving implies specialized hardware

- ◆ H.264 video decoder implementations in software vs. hardware
 - the power/energy savings could be 100 to 1000 fold

but our mind set is that hardware design is:

- Difficult, risky
 - ◆ Increases time-to-market

Real power saving implies specialized hardware

- ◆ H.264 video decoder implementations in software vs. hardware
 - the power/energy savings could be 100 to 1000 fold

but our mind set is that hardware design is:

- Difficult, risky
 - ◆ Increases time-to-market
- Inflexible, brittle, error prone, ...
 - ◆ Difficult to deal with changing standards, ...

Real power saving implies specialized hardware

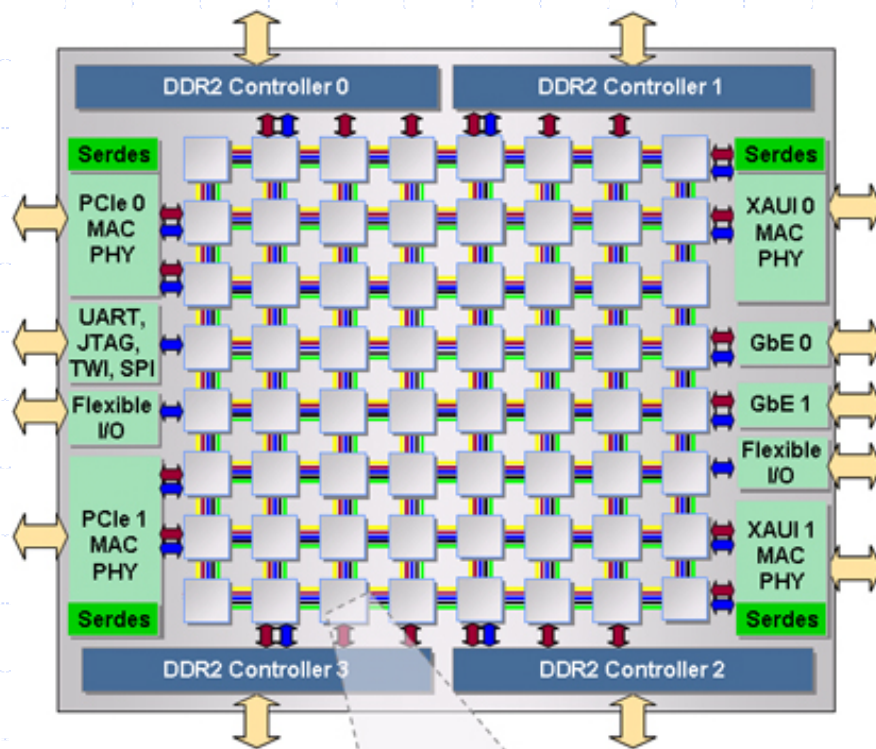
- ◆ H.264 video decoder implementations in software vs. hardware
 - the power/energy savings could be 100 to 1000 fold

but our mind set is that hardware design is:

- Difficult, risky
 - ◆ Increases time-to-market
- Inflexible, brittle, error prone, ...
 - ◆ Difficult to deal with changing standards, ...

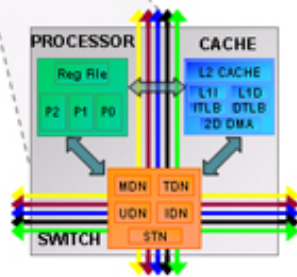
**New design flows and tools
can change this mind set**

Will multicores reduce the need for new hardware?

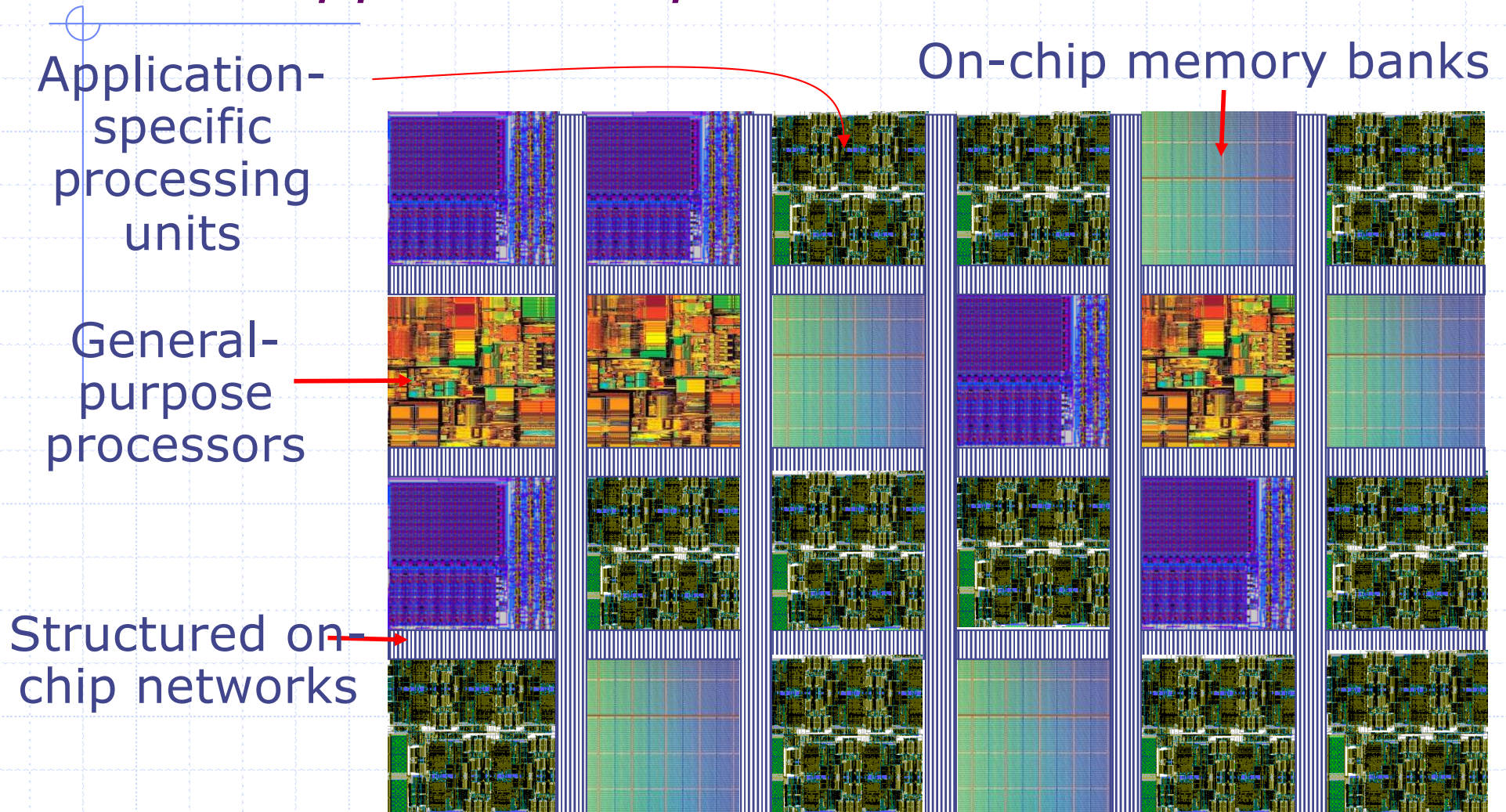


Unlikely –
because of
power and
performance

64-core Tiler



SoC & Multicore Convergence: *more application specific blocks*



To reduce the design cost of SoCs we need ...

- ◆ Extreme IP reuse — “Intellectual Property”
 - Multiple instantiations of a block for different performance and application requirements
 - Packaging of IP so that the blocks can be assembled easily to build a large system (black box model)
- ◆ Architectural exploration to understand cost, power and performance tradeoffs
- ◆ Full system simulations for validation and verification

Hardware design today is like programming was in the fifties, i.e., before the invention of high-level languages

Programmers had to know many detail of their computer



IBM 650
(1954)

An IBM 650 Instruction: 60 1234 1009

“Load the contents of location 1234 into the *distribution*; put it also into the *upper accumulator*; set *lower accumulator* to zero; and then go to location 1009 for the next instruction.”

Programmers had to know many detail of their computer



IBM 650
(1954)



1950s
reaction

Can you program a computer without knowing , for example, how many registers it has?

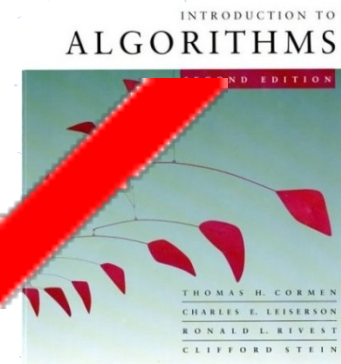
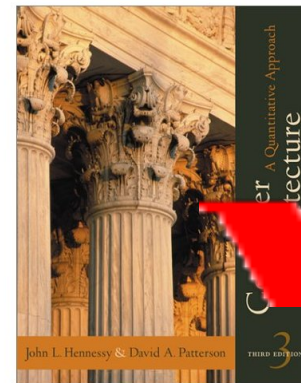
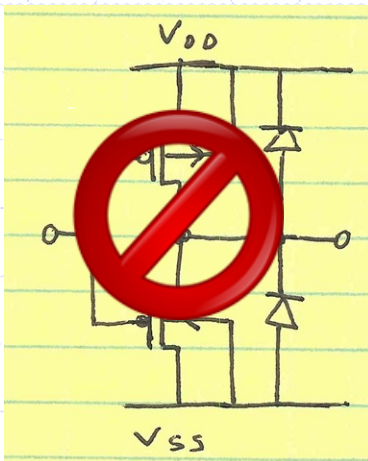
Fortran changed this mind set (1956)

For designing complex SoCs deep circuits knowledge is secondary

Using modern high-level hardware synthesis tools like Bluespec requires computer science training in programming and architecture rather than circuit design

For designing complex SoCs deep circuits knowledge is secondary

Using modern high-level hardware synthesis tools like Bluespec requires computer science training in programming and architecture rather than circuit design



Bluespec A new way of expressing behavior

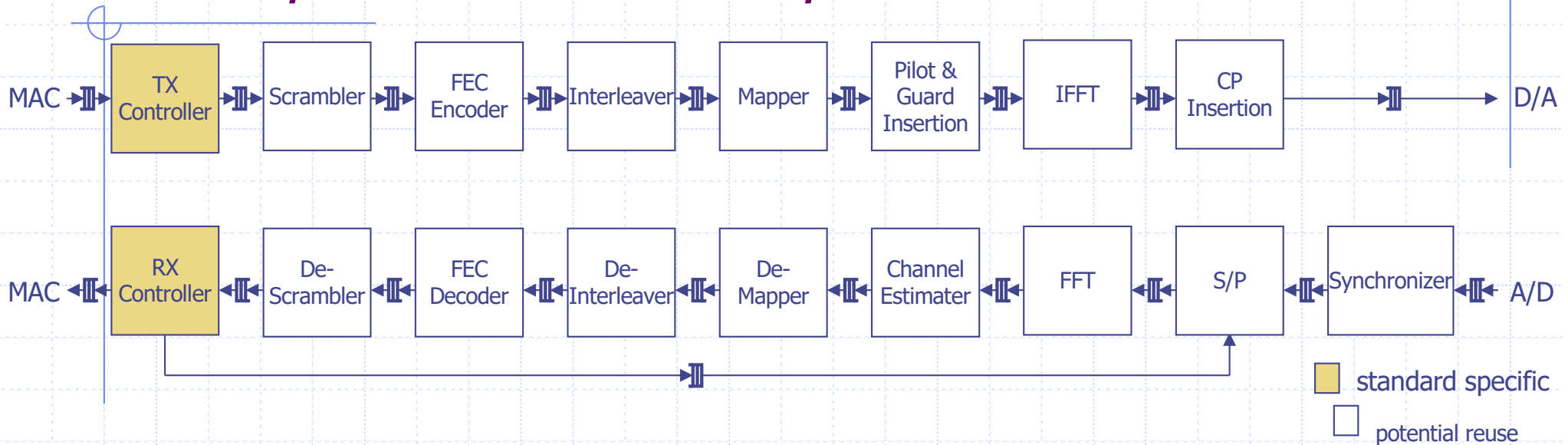
- ◆ A formal method of composing modules with parallel interfaces (ports)
 - Compiler manages muxing of ports and associated control
- ◆ Powerful and *zero-cost* parameterization of modules
 - ◆ Encapsulation of C and Verilog codes using Bluespec wrappers
 - Helps Transaction Level modeling

→ *Smaller, simpler, clearer, more correct code*

→ *not just simulation, synthesis as well*

IP Reuse via parameterized modules

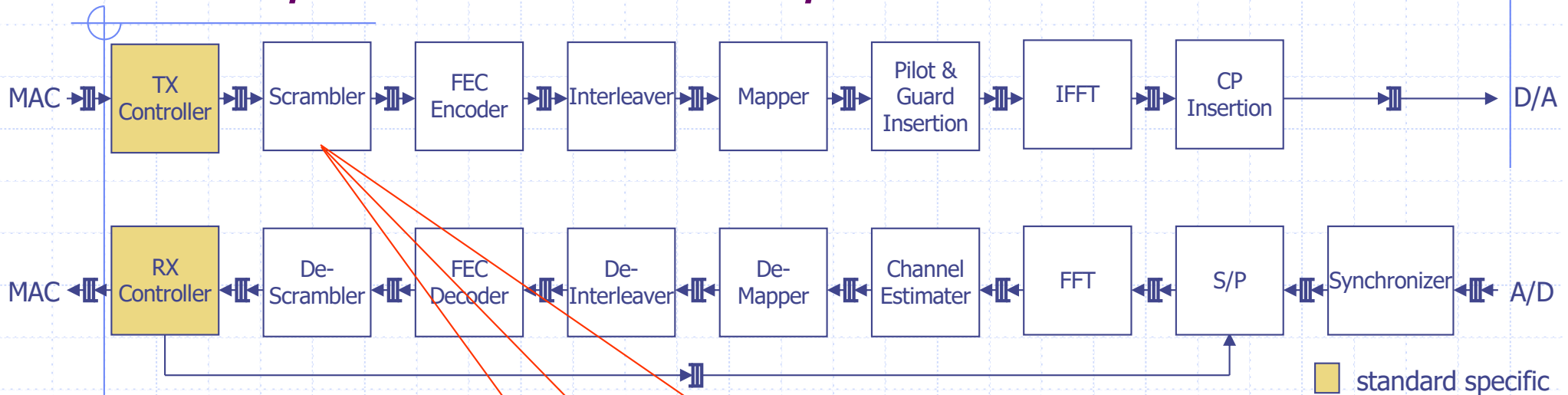
Example OFDM based protocols



(Alfred) Man Cheuk Ng, ...

IP Reuse via parameterized modules

Example OFDM based protocols



- Reusable algorithm with different parameter settings

WiFi: $x^7 + x^4 + 1$

WiMAX: $x^{15} + x^{14} + 1$

WUSB: $x^{15} + x^{14} + 1$

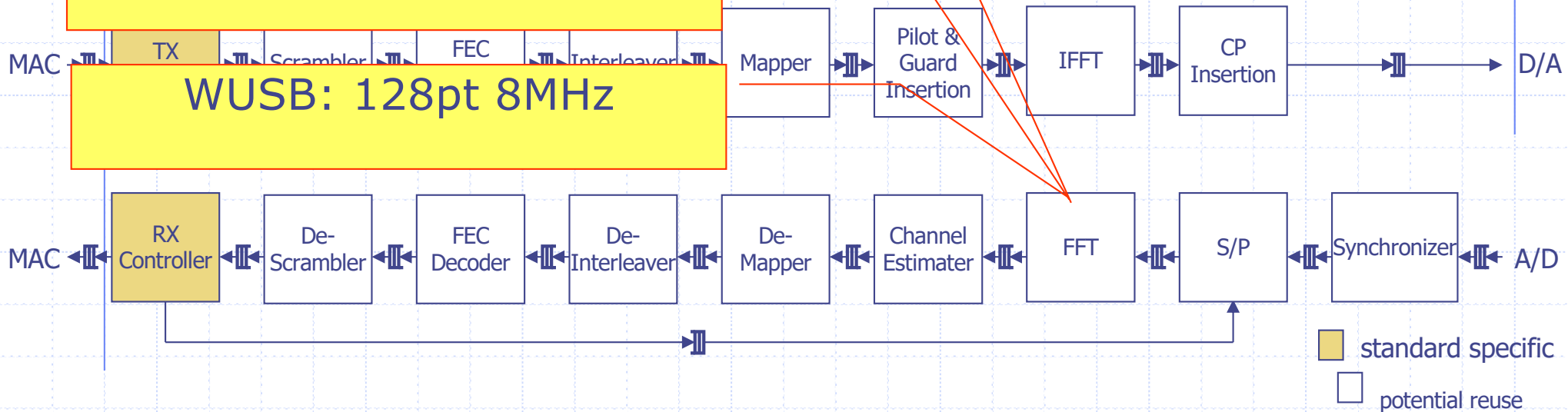
(Alfred) Man Cheuk Ng, ...

WiFi: 64pt @ 0.25MHz

WiMAX: 256pt @ 0.03MHz

WUSB: 128pt 8MHz

Parameterized modules Example OFDM-based protocols

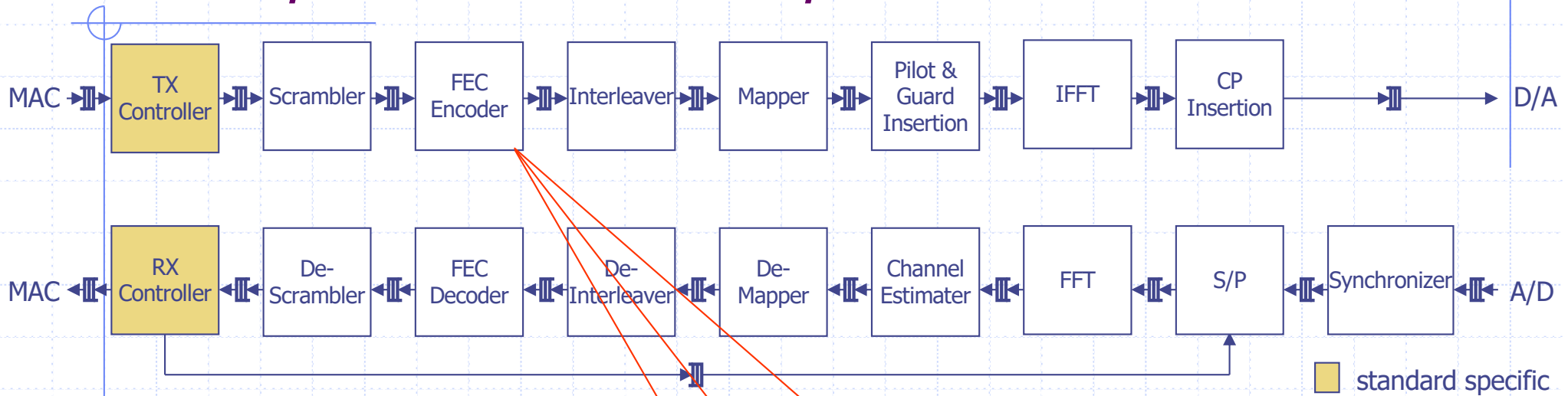


- Reusable algorithm with different parameter settings
- Different throughput requirements

(Alfred) Man Cheuk Ng, ...

IP Reuse via parameterized modules

Example OFDM based protocols



- Reusable algorithm with different parameter settings
- Different throughput requirements
- Different algorithms

Convolutional

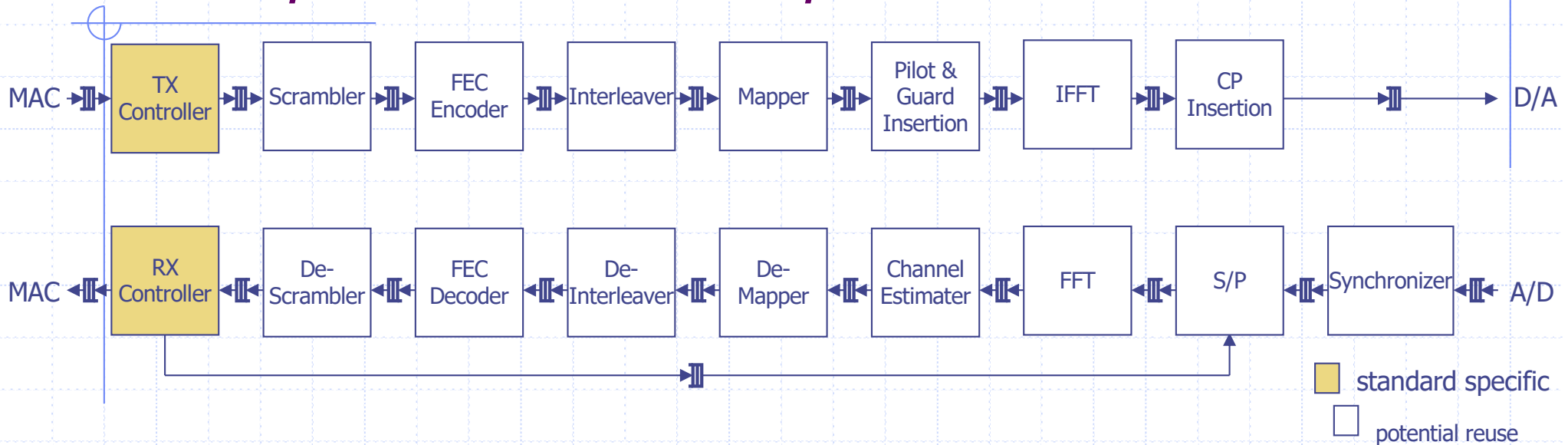
Reed-Solomon

Turbo

(Alfred) Man Cheuk Ng, ...

IP Reuse via parameterized modules

Example OFDM based protocols



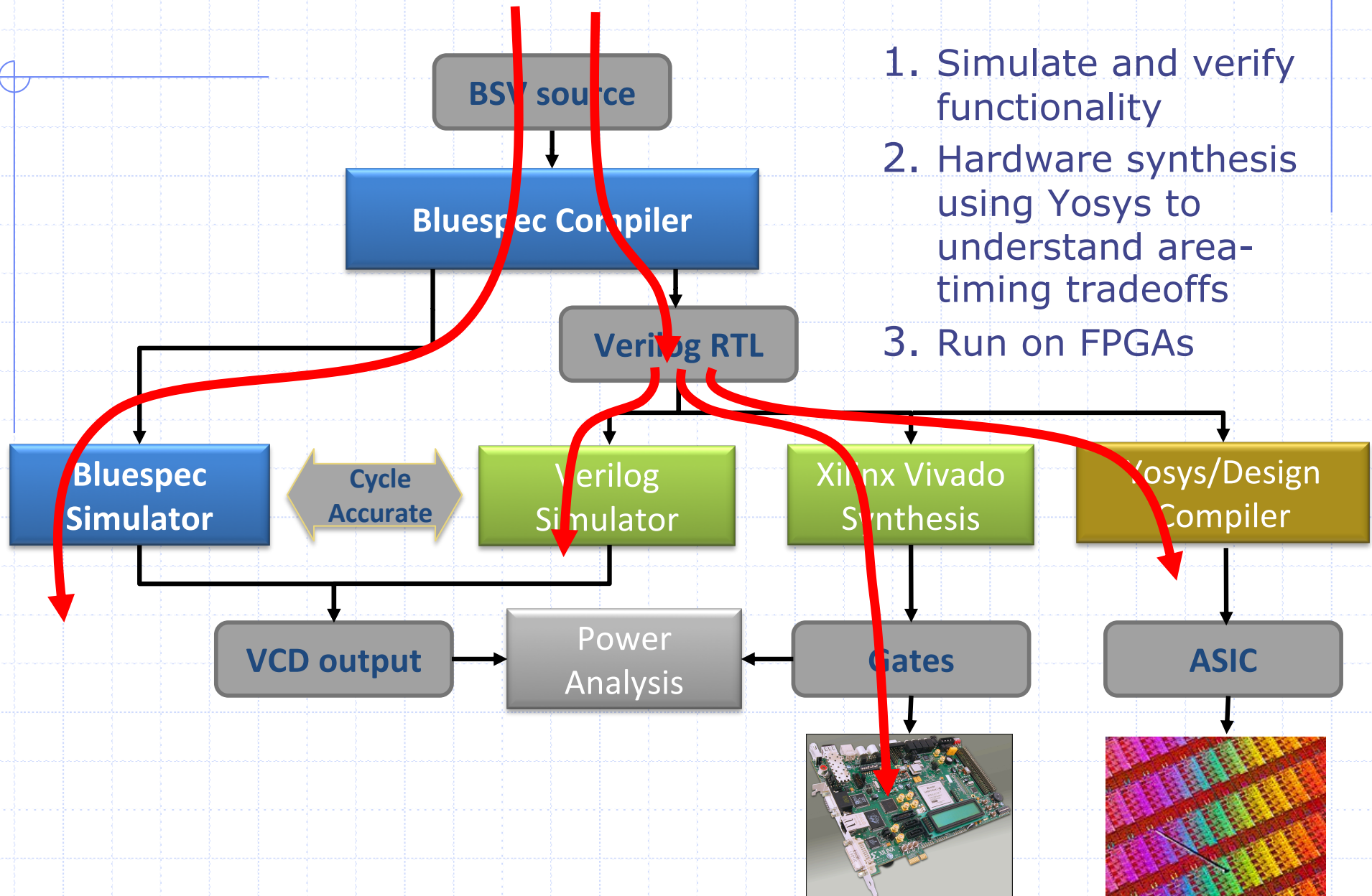
- Reusable algorithm with different parameter settings

- **85% reusable code between WiFi and WiMAX**
From WiFi to WiMAX in 4 weeks
- Different algorithms

(Alfred) Man Cheuk Ng, ...

BSV Design Flow

1. Simulate and verify functionality
2. Hardware synthesis using Yosys to understand area-timing tradeoffs
3. Run on FPGAs



Chip Design Styles

◆ Custom and Semi-Custom

- Hand-drawn transistors (+ some standard cells)
- High volume, best possible performance: used for most advanced microprocessors

◆ Standard-Cell-Based ASICs

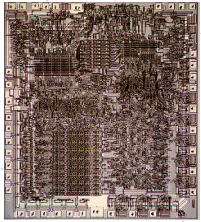
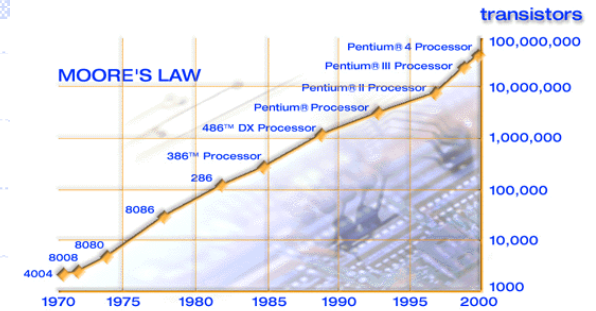
- High volume, moderate performance: Graphics chips, network chips, cell-phone chips

◆ Field-Programmable Gate Arrays

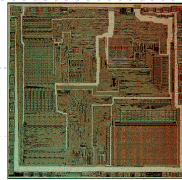
- Prototyping
- Low volume, low-moderate performance applications

Different design styles have vastly different costs

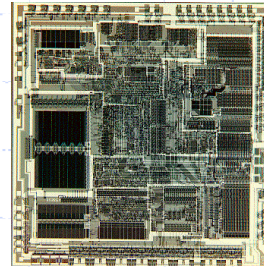
Exponential growth: Moore's Law



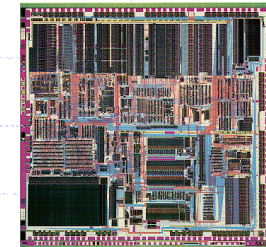
Intel 8080A, 1974
3Mhz, 6K transistors, 6u



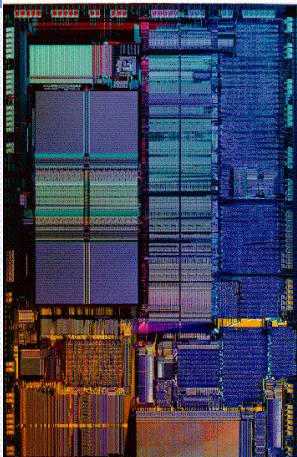
Intel 8086, 1978, 33mm²
10Mhz, 29K transistors, 3u



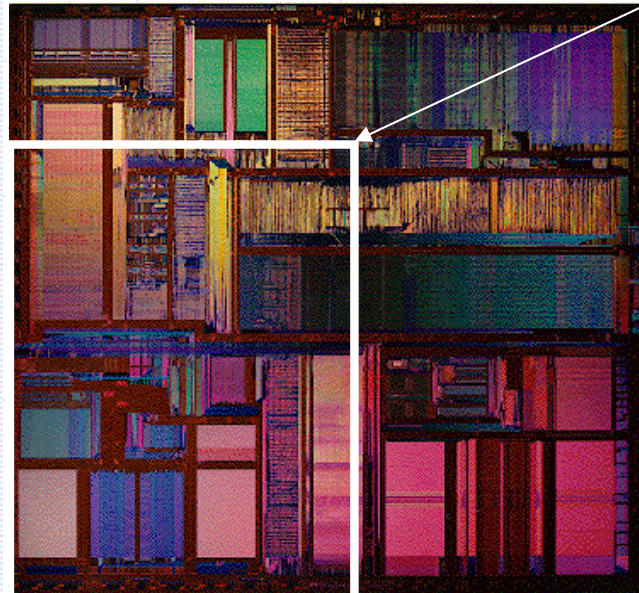
Intel 80286, 1982, 47mm²
12.5Mhz, 134K transistors, 1.5u



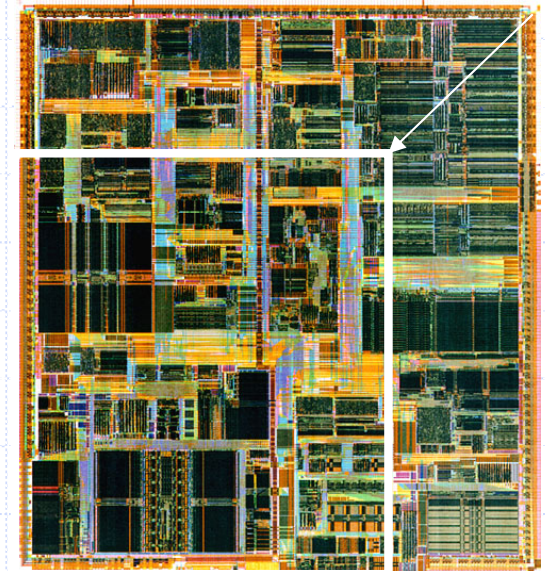
Intel 386DX, 1985, 43mm²
33Mhz, 275K transistors, 1u



Intel 486, 1989, 81mm²
50Mhz, 1.2M transistors, .8u



Intel Pentium, 1993/1994/1996, 295/147/90mm²
66Mhz, 3.1M transistors, .8u/.6u/.35u



Intel Pentium II, 1997, 203mm²/104mm²
300/333Mhz, 7.5M transistors, .35u/.25u

Shown with approximate relative sizes

http://www.intel.com/intel/intelis/museum/exhibit/hist_micro/hof/hof_main.htm

Intel Ivy Bridge 2012

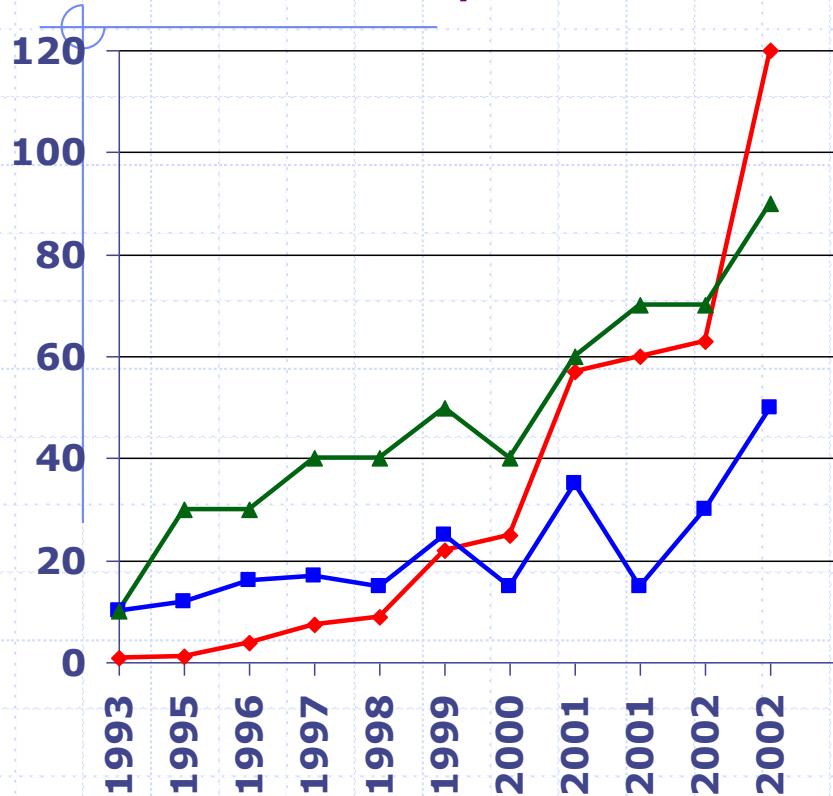
- ◆ Quad core
- ◆ Quad-issue out-of-order superscalar processors
- ◆ Caches:
 - L1 64 KB/core
 - L2 256 KB/core
 - L3 6 MB shared
- ◆ 22nm technology
- ◆ 1.4 Billion transistors
- ◆ 3.4 GHz clock frequency
- ◆ Power > 17 Watts (under clocked)



*Could fit over 1200 486 processors
on same size die.*

But Design Effort is Growing

Nvidia Graphics Processing Units



Transistors (M)

Relative staffing
on back-end

9x growth in
back-end staff

Relative staffing
on front-end

5x growth in
front-end staff

- ◆ Front-end is designing the logic (RTL)
- ◆ Back-end is fitting all the gates and wires on the chip; meeting timing specifications; wiring up power, ground, and clock

Design Cost Impacts Chip Cost

An Altera study

- ◆ Non-Recurring Engineering (NRE) costs for a 90nm ASIC is ~ \$30M
 - 59% chip design (architecture, logic & I/O design, product & test engineering)
 - 30% software and applications development
 - 11% prototyping (masks, wafers, boards)
- ◆ If we sell 100,000 units, NRE costs add

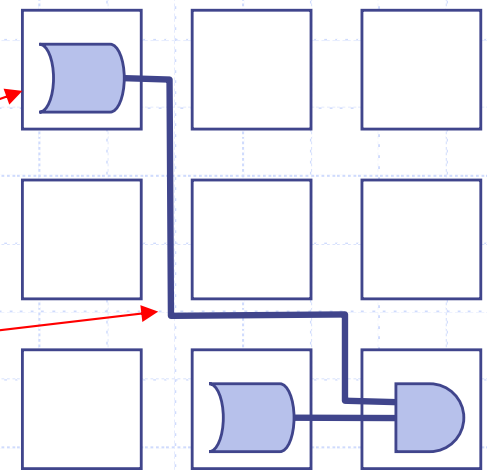
$$\$30\text{M}/100\text{K} = \$300 \text{ per chip!}$$

Hand-crafted IBM-Sony-Toshiba Cell microprocessor achieves 4GHz in 90nm, but at the development cost of >\$400M

Alternative: Use FPGAs

Field-Programmable Gate Arrays (FPGAs)

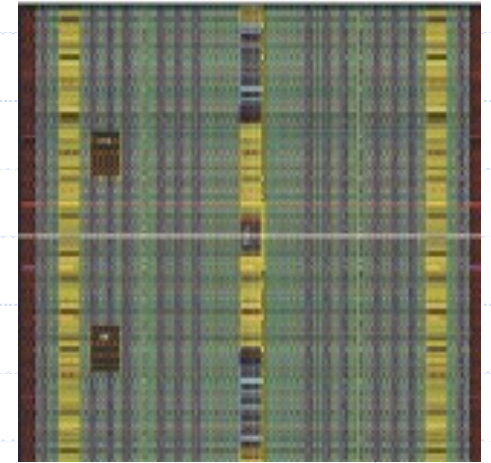
- ◆ Arrays mass-produced but programmed by customer after fabrication
 - Can be programmed by loading SRAM bits, or loading FLASH memory
- ◆ Each cell in array contains a programmable logic function
- ◆ Array has programmable interconnect between logic functions
- ◆ Overhead of programmability makes arrays expensive and slow as compared to ASICs
- ◆ However, much cheaper than an ASIC for small volumes because NRE costs do not include chip development costs (only include programming)



FPGA Pros and Cons

Advantages

- Dramatically reduce the cost of errors
- Little physical design work
- Remove the reticle costs from each design



Disadvantages (as compared to an ASIC)

[Kuon & Rose, FPGA2006]

- Switching power around $\sim 12X$ worse
- Performance up 3-4X worse
- Area 20-40X greater

Still requires
tremendous design
effort at RTL level

FPGAs: a new opportunity

- ◆ “Big” FPGAs have become widely available
 - A multicore can be emulated on one FPGA
 - but the programming model is RTL and not too many people design hardware
- ◆ Enable the use of FPGAs via Bluespec or other High-Level Synthesis (HLS) tools

6.375 Philosophy

- ◆ Effective abstractions to reduce design effort
 - High-level design language rather than logic gates
 - Control specified with Guarded Atomic Actions rather than with finite state machines
 - Guarded module interfaces to systematically build larger modules by the composition of smaller modules
- ◆ Design discipline to avoid bad design points
 - Decoupled units rather than tightly coupled state machines
- ◆ Design space exploration to find good designs
 - Architecture choice has the largest impact on solution quality

We learn by doing actual designs

6.375 Complex Digital Systems: past projects

- ◆ Optical flow in Harvard Robo Bee project
- ◆ Spinal Codes for Wireless Communication
- ◆ Beat tracker
- ◆ H.265 Motion Estimation for video compression
 - A chip was fabricated later
- ◆ Hard Viterbi Decoder
- ◆ Video motion magnification
- ◆ RSA
- ◆ Programmable packet filter for 1Gbps stream

6 weeks of individual lab work
+ 6-week group projects

Fun: Design systems that you thought you would never design in a course

Resources, in addition to classmates and mentors

- ◆ Lecture slides (with animation)
 - <http://csg.csail.mit.edu/6.375/> > Handouts
- ◆ “Computer Architecture: Introduction to Digital Design as Cooperating Sequential machines”, Arvind, Rishiyur S. Nikhil, James E. Hoe, and Silvina Hanono Wachman
- ◆ Bluespec System Verilog Reference manual
 - You may need to refer to this if you use advanced features of BSV

- ◆ The following books may also be useful
 - BSV By Example, Rishiyur S. Nikhil and Kathy R. Czeck (2010)
 - Bluespec System Verilog Users guide
 - ◆ How to use all the tools for developing BSV programs