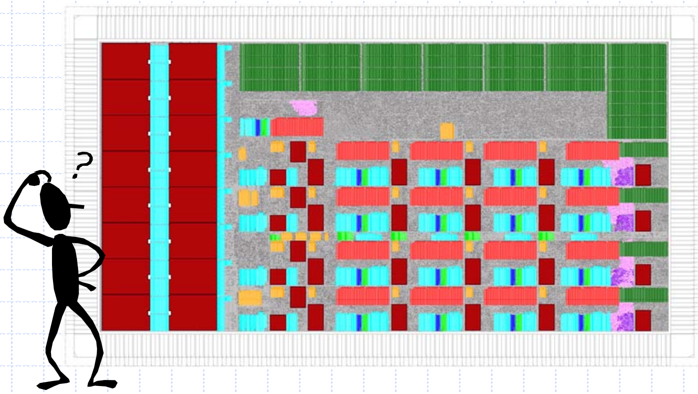
# 6.375: Complex Digital Systems



Lecturer: Arvind
TA: Richard S. Uhler
Administration: Sally Lee



# 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 …

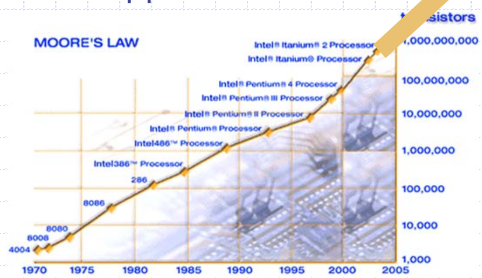


# Wide Variety of Products Rely on ASICs

ASIC = Application-Specific Integrated Circuit

# What's required?

ICs with dramatically higher performance, optimized for applications




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

*and* at a

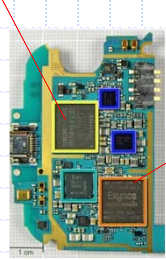
- size and power to deliver mobility
- cost to address mass consumer markets

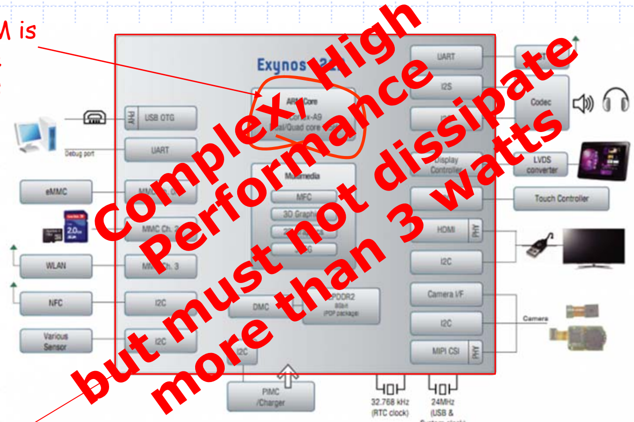


# Cell Phones: Samsung Galaxy S III April 2012

Quad core ARM is just one of the complex blocks

16GB NAND flash






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

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

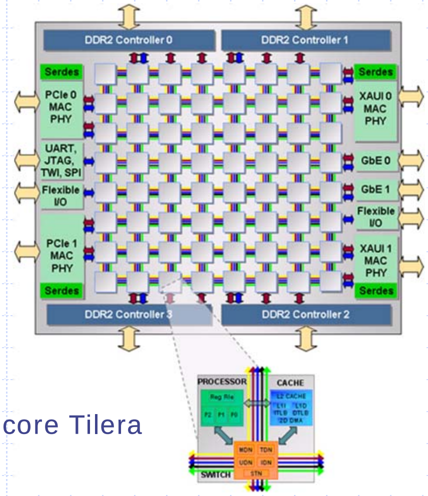
- H.264 video decoder implementations in software vs. hardware
  - the power/energy savings could be 100 to 1000 fold

*but our mind set is that hardware design is:*

- Difficult, risky
  - Increases time-to-ma
- Inflexible, brittle, er
  - Difficult to deal with changing standards, …

New design flows and tools can change this mind set

# Will multicores reduce the need for new hardware?



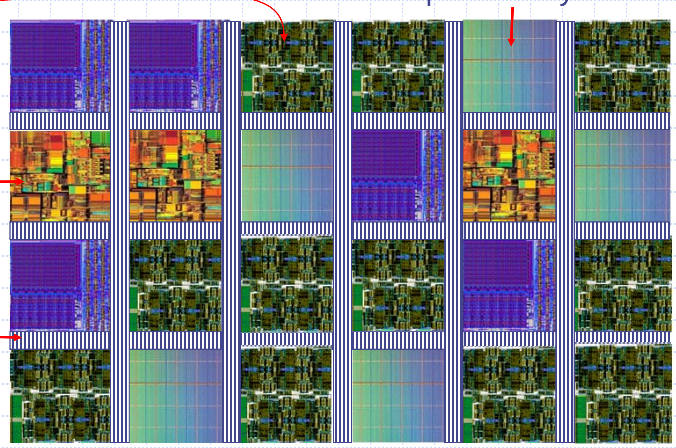

64-core Tilera

Unlikely –
because of
power and
performance



# 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)

1950s reaction

IBM 650 Instruction: 60 1234 1009

*distribution*, put it also into the *upper accumulator*, set *lower accumulator* to zero; and then go to location 1009 for the next instruction."

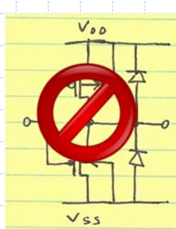
Can you program a computer without knowing , for example, how many registers it has?

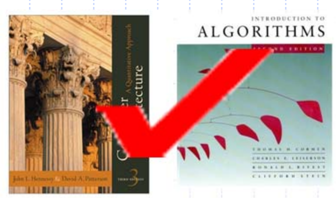
Fortran changed this mind set (1956)



# 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



## 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*



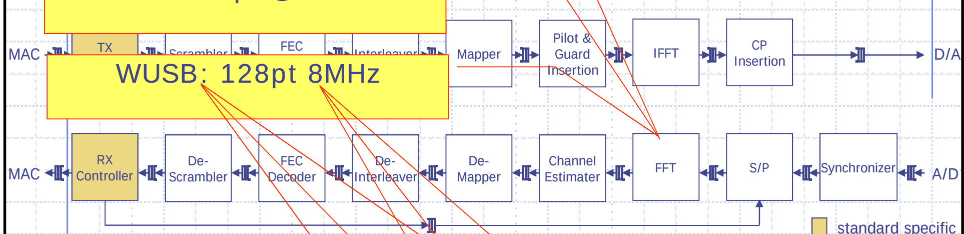
WiFi: 64pt @ 0.25MHz

## ...ameterized modules

*...ed protocols*

WiMAX: 256pt @ 0.03MHz

WUSB: 128pt 8MHz




Convolutional

Reed-Solomon

Turbo

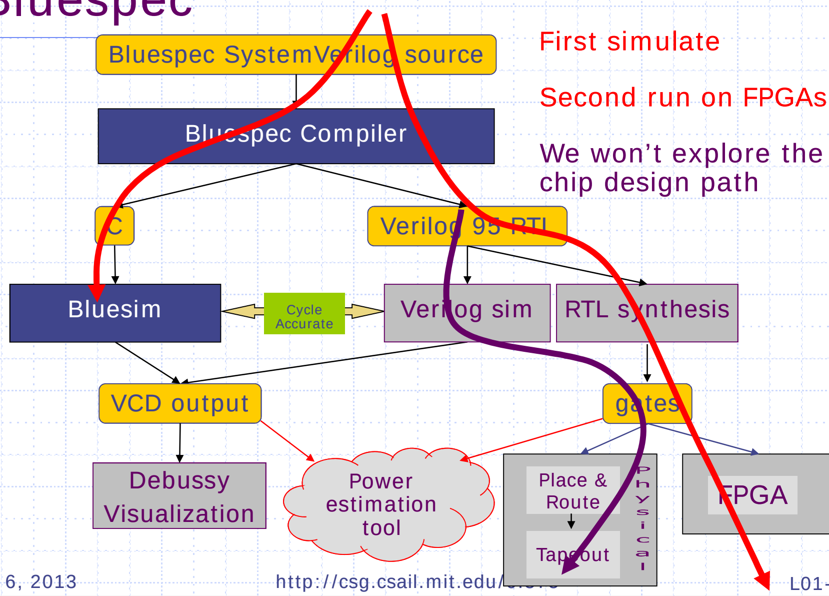
- Reusable algorithm with different parameter settings
- 
- Different algorithms

85% reusable code between WiFi an... WiMAX

From WiFi to WiMAX in 4 wee...

(Alfred) Man Cheuk Ng, ...

# High-level Synthesis from Bluespec



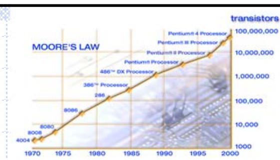



# Chip Design Styles

- 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

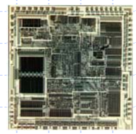
**Different design styles have vastly different costs**

# Exponential growth: Moore's Law

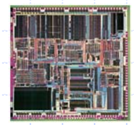



Intel 8080A, 1974
3Mhz, 6K transistors, 6u

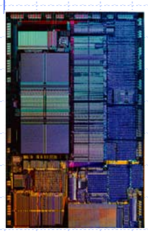
Intel 8086, 1978, 33mm$^2$
10Mhz, 29K transistors, 3u



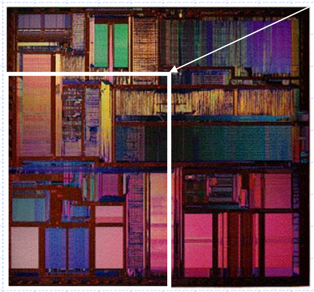
Intel 80286, 1982, 47mm$^2$
12.5Mhz, 134K transistors, 1.5u



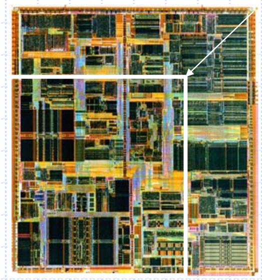
Intel 386DX, 1985, 43mm$^2$
33Mhz, 275K transistors, 1u



Intel 486, 1989, 81mm$^2$
50Mhz, 1.2M transistors, .8u



Intel Pentium, 1993/1994/1996, 295/147/90mm$^2$
66Mhz, 3.1M transistors, .8u/.6u/.35u



Intel Pentium II, 1997, 203mm$^2$/104mm$^2$
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



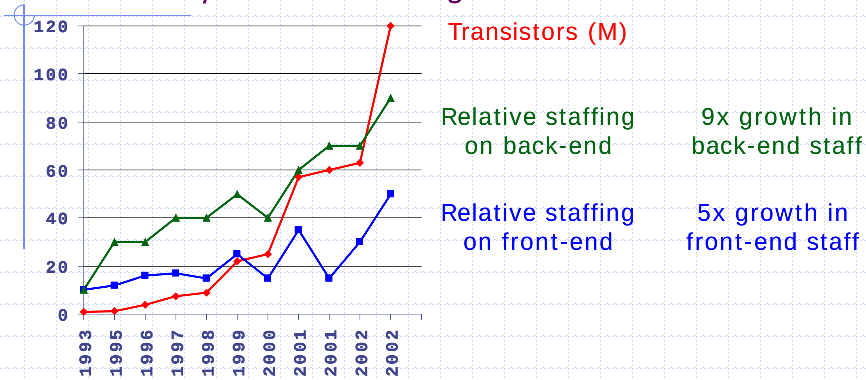
# Intel Ivy Bridge 2012

- 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*




- 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

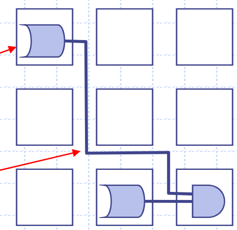
$30M/100K = $300 per chip!

*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)

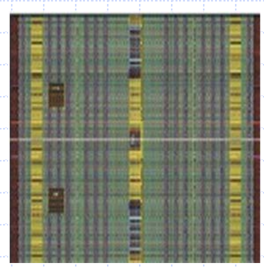




# FPGA Pros and Cons

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 ~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



# 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 largest impact on solution quality

We learn by doing actual designs

# 6.375 Complex Digital Systems: 2011 projects

- Optical flow in Harvard Robo Bee project
- Spinal Codes for Wireless Communication
- Data Movement Control Instruction and OS extension for multicore PPC
- H.265 Motion Estimation for video compression
  - A chip was fabricated soon afterwards
- Hard Viterbi Decoder

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

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



# Resources – beyond TA, mentors and classmates

- Lecture slides (with animation)
  - Make sure you sure you understand the lectures before exploring other materials
  - http://csg.csail.mit.edu/6.375/handouts.html
- BSV By Example, Rishiyur S. Nikhil and Kathy R. Czeck (2010)
- Computer Architecture: A Constructive Approach, Arvind, Rishiyur S. Nikhil, Joel S. Emer, and Murali Vijayaraghavan (2012)
  - Uses Executable and Synthesizable processor Specifications
- Bluespec System Verilog Reference manual
- Bluespec System Verilog Users guide
  - How to use all the tools for developing BSV programs