

6.375 Complex Digital System Spring 2009



Lecturer: Arvind
TA: K. Elliott Fleming
Assistant: Sally Lee

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-1

Why take 6.375? Take 1

◆ We need a much greater variety of chips (ASICs)

◆ Why?

- Power savings: Specialized hardware for a video decoder (H.264) may consume $1/100^{\text{th}}$ to $1/1000^{\text{th}}$ the power of a software implementation
- Cost
- Performance
- Size ...

ASIC = Application-Specific Integrated Circuit

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-2

Wide Variety of Products Rely on ASICs



February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-3

What's required?

ICs with dramatically higher performance, optimized for applications



and at a

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

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-4

ASIC 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 require different design tools and have vastly different chip development cost

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-5

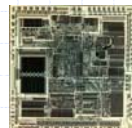
Exponential growth: Moore's Law



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



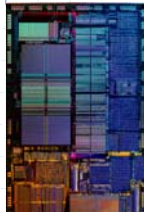
Intel 8086, 1978, 33mm²
10Mhz, 29K transistors, 3u



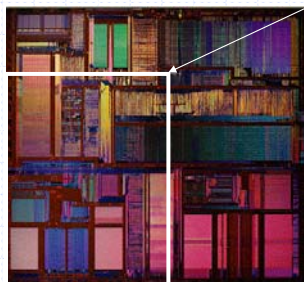
Intel 80286, 1982, 47mm²
12.5Mhz, 134K transistors, 1.5u



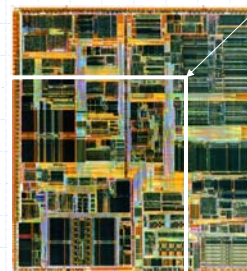
Intel 386DX, 1985, 43mm²
33Mhz, 275K transistors, 1u



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

February 4, 2009

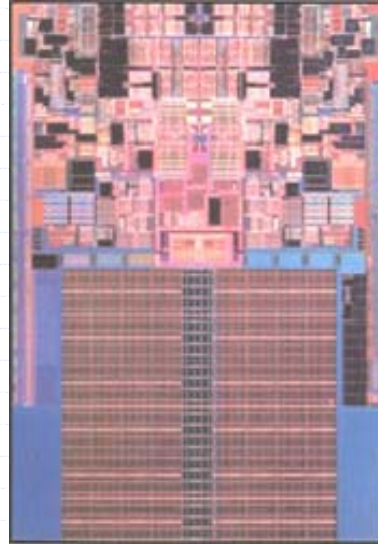
<http://csg.csail.mit.edu/6.375/>

L01-6

Intel Penryn (2007)

- ◆ Dual core
- ◆ Quad-issue out-of-order superscalar processors
- ◆ 6MB shared L2 cache
- ◆ 45nm technology
 - Metal gate transistors
 - High-K gate dielectric
- ◆ 410 Million transistors
- ◆ 3+? GHz clock frequency

Could fit over 500 486 processors on same size die.



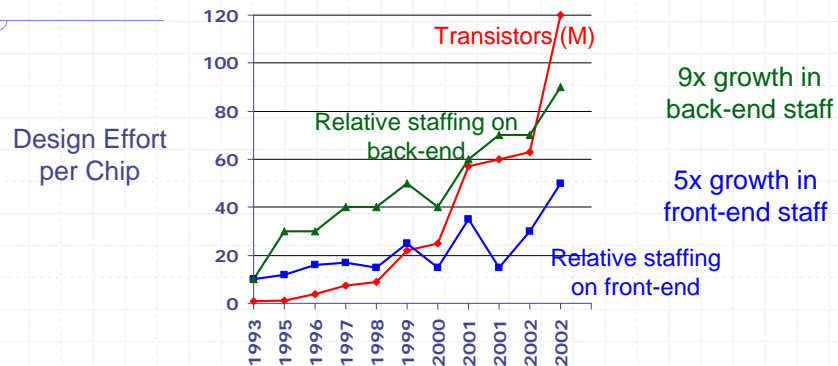
February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-7

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

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-8

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

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

Alternative: Use FPGAs

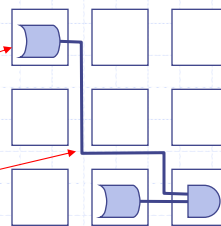
February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-9

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 but startup costs are low, so much cheaper than ASIC for small volumes



February 4, 2009

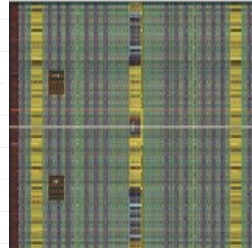
<http://csg.csail.mit.edu/6.375/>

L01-10

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

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-11

What is needed to make hardware design easier

- ◆ 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)
- ◆ Ability to do modular refinement
- ◆ Whole system simulation to enable concurrent hardware-software development

Need new methods and tools to raise the level of design

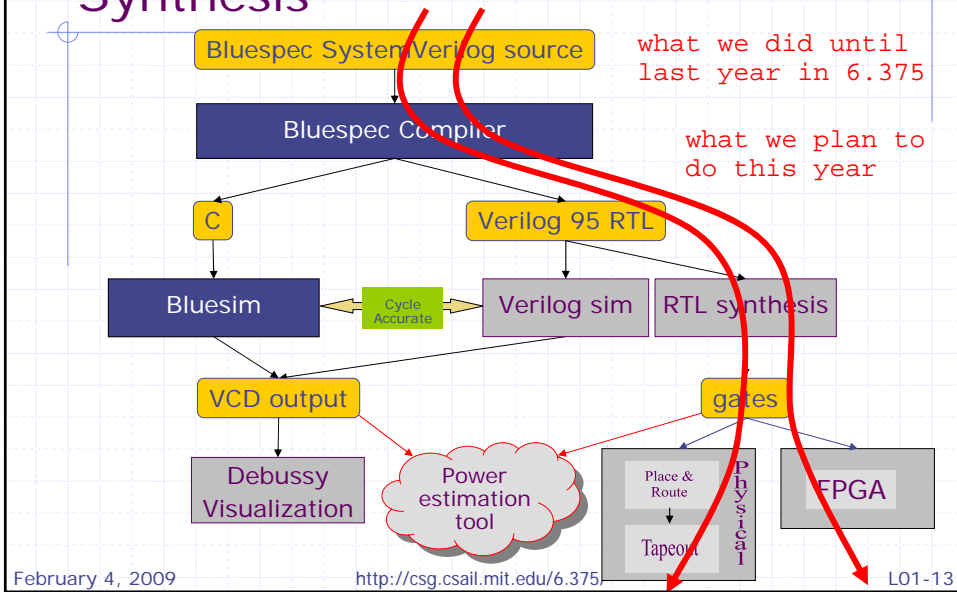
Bluespec

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-12

Bluespec: Enabling High-level Synthesis



Why take 6.375?

Take 2 - The 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

Some cool projects

- ◆ IBM PowerPC Prototype
- ◆ Intel's HAsim – Cycle-accurate performance models
- ◆ AirBlue – A new platform to experiment with wireless protocols
- ◆ Video decoder – H.264
- ◆ Hardware software co-generation

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-15

IBM: PowerPC Prototype

K. Ekanadham, Jessica Tseng (IBM)
Asif Khan, M. Vijayaraghavan (MIT)

- ◆ Goal: Implement a multithreaded, multicore, in-order PowerPC on an FPGA platform and boot Linux on it in 12 months
- ◆ Team:
 - 2(IBM) + 2(MIT) + Linux and FPGA help

The team accomplished the goal

- Bluespec PowerPC boots Linux on FPGAs in 10min;
- 100M instructions to reach "Hello World";
- 15K lines of Bluespec generated 90K lines of Verilog

IBM synthesized the generated Verilog using their tools in 40nm library

- ran at 500MHz in the first try!

February 4, 2009

Working on a public release...
<http://csg.csail.mit.edu/6.375/>

L01-16

HAsim: Performance modeling of CPUs

Joel Emer ... (Intel), M. Pellauer ... (MIT)

- ◆ Intel Asim:
 - Framework for execution-driven simulation
 - Performance: 10s to 100s of KIPS for high-detail models
 - Parallelizing the simulator could get 3x to 5x
- ◆ But want 1,000x or 10,000x speedup
- ◆ HAsim: Configure FPGAs into a simulator of the target design

Three different models of MIPS/Alpha have been developed over the last two years

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-17

AirBlue: A platform to experiment with wireless protocols

Hari Balakrishnan, R. Gummadi, A. Ng, E. Flemming

- ◆ SoftPHY: Expose signal quality to higher layers
 - Enables new protocols
 - ◆ MIXIT (wireless network coding)
 - ◆ PPR (Partial Packet Recovery)
 - ◆ Rate adaptation
- ◆ Allocate OFDM channels efficiently
 - Variable demands
 - Variable SNRs

Supported
by Nokia

Status: Several cross-layer experiments have already been conducted on a 24Mbps implementation of 802.11 implementation developed in the last six months

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-18

Parameterized modules

Example OFDM based protocols

WiFi: 64pt @ 0.25MHz

WiMAX: 256pt @ 0.03MHz

WUSB: 128pt 8MHz

Convolutional

Reed-Solomon

Turbo

- 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, ...

February 4, 2009 <http://csg.csail.mit.edu/6.375/> L01-19

H.264 Video Decoder

Elliott Fleming, Chun Chieh Lin

Supported by Nokia

Different requirements for different environments

- QVGA 320x240p (30 fps)
- DVD 720x480p
- HD DVD 1280x720p (60-75 fps)

May be implemented in hardware or software depending upon ...

February 4, 2009 <http://csg.csail.mit.edu/6.375/> L01-20

H.264 in Bluespec

- ◆ Initial Design: Base profile
 - Eight man-months
 - 8K lines of Bluespec
 - ◆ in contrast to 80K lines of C standard
 - Decoded 720p @ 32FPS

- ◆ Major architectural explorations over 3 months to meet different performance or cost criteria
 - High performance designs (4.2 mm sq in 180nm)
 - ◆ 720p@75FPS, 1080p@ 65FPS,
 - Low cost designs
 - ◆ QCIF@15FPS (2.2mm sq), 720p@30FPS (2.4mm sq)
 - FPGA implementations for VGA output

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

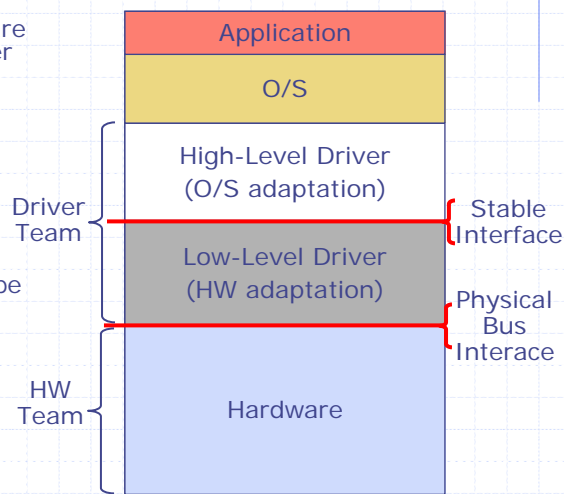
L01-21

Hw/Sw codesign in Bluespec: FEC Decoder

- ◆ Any changes in hardware affects the device driver
- ◆ Split the device-driver
- ◆ Make the low-level device driver the responsibility of the hardware team
- ◆ Use Bluespec to describe both the hardware and the low-level device driver

The compiler is still under development

Has implications for parallel programming



Supported by Nokia

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-22

6.375 Course 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 automatically ensure correctness of composition of existing 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

A unified view of language, design discipline and tools that supports rapid design space exploration to find best area, power, and performance point

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-23

6.375 Objectives

By end of term, you should be able to:

- ◆ Decompose system requirements into a hierarchy of sub-units that are easy to specify, implement, and verify, and which can be *reused*
- ◆ Select appropriate microarchitectures to meet performance and area goals
- ◆ Develop efficient verification and test plans
- ◆ Understand FPGA specific optimizations
- ◆ Learn how to integrate your design into a complex system
- ◆ Use industry-standard tool flows
- ◆ Complete a working FPGA implementation!

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-24

6.375 Prerequisites

- ◆ You must be familiar with undergraduate (6.004) logic design and basic programming:
 - Combinational and sequential logic design
 - Dynamic Discipline (clocking, setup and hold)
 - Finite State Machine design
 - Binary arithmetic and other encodings
 - Simple pipelining
 - ROMs/RAMs/register files
- ◆ Additional circuit knowledge may be useful but is not vital
- ◆ Architecture knowledge (6.823) is helpful for projects

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-25

6.375 Structure

- ◆ First half of term (before Spring Break)
 - Lecture or tutorial MWF, 2:30pm to 4:00pm in 32-124
 - Three labs (lab machines in 38-301, home computers)
 - Form project teams (3 students); prepare project proposal (watch website for project ideas)
- ◆ Second half of term (after Spring Break)
 - Weekly project milestones, with 1-2 page report
 - Weekly project meeting with the instructor, TA and a graduate student mentor
 - Final project presentations and demonstrations in the last week of classes
 - Final project report (~15-20 pages) due Thursday May 14 (no extensions)

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-26

6.375 Grade Breakdown

- ◆ Three labs 30%
- ◆ Five project milestones 20%
- ◆ Final project demonstration on FPGAs 25%
- ◆ Final project report (including presentation) 25%

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-27

6.375 Collaboration Policy

- ◆ We strongly encourage students to collaborate on understanding the course material, BUT:
 - Each student must turn in individual solutions to labs
 - If you ever borrow ideas, code, ... from anywhere, you must explicitly acknowledge the source

February 4, 2009

<http://csg.csail.mit.edu/6.375/>

L01-28