Lecture 1:
Intro to Computer Architecture

Today's Agenda

- Course logistics
- A brief history of computer architecture

Course Resources

Textbooks (none required)

Patterson & Hennessy, *Computer Organization and Design,* **5th edition**

Reese & Thornton, *Intro to Logic Synthesis using Verilog HDL*

Digital Design: A Systems Approach by Dally and Harting

The System Verilog Labs

- "Build your own processor" (pipelined 32-bit RISC-V CPU)
- Use SystemVerilog HDL (hardware description language)
  Programming language compiles to gates/wires not insns
- Implement and test on real hardware
  FPGA (field-programmable gate array)

+ Instructive: learn by doing
+ Satisfying: "look, I built my own processor"

Lab Logistics

Tools

Quartus/Xilinx Vivado hardware compiler
DE-2/DE-10 FPGA boards

Logistics

Development and simulation can be done before final testing on the board

Labs

Labs done in groups of two or three

- start forming groups now, be mindful of Pass/Fail

Lab descriptions

- Lab 1: SystemVerilog debugging
- Lab 2: Arithmetic unit
- Lab 3: Single-cycle RISC-V & Register File
- Lab 4: Pipelined RISC-V
- Lab 5: Pipelined+ RISC-V

Labs are **cumulative** and increasingly complex

Each lab broken down into "milestone" deadlines

- Roughly one per week

Threshold grading: grades are [0%,90%] or 100%

In many ways this is a class about **debugging**

http://debuggingrules.com

Microprocessor vs Computer Architecture

In EE2039 you learned how a processor worked, in EE3043 we will tell you how to make it work **well**.

Microprocessor
EE2039

Computer Architecture
EE3043

What is EE3043 ?

- EE2013: Introduction to Computer Systems
  - "C" as the model of computation
  - interact with the computer hardware through OS
  - what about the details below the abstraction?

Somehow a program ends
up executing as digital logic

- EE1009: Fundamentals of Computer Engineering
  - digital logic as the model of computation
  - gates and wires as building blocks
  - what about the details below this abstraction?

EE3043: Fuzzy to Concrete

EE2009

- "Computer Architecture"
  - functional spec for software and programmers
  - design spec for the hardware people
- Computer Organization
  - take architecture to "micro"architecture
  - how to assemble/evaluate/tune
- Computation Structures
  - digital representations
  - processing, storage and I/O elements

EE1009

Abstraction, Layering, and Computers

Computers are complex, built in layers

Several **software** layers: assembler, compiler, OS, applications
Instruction set architecture (ISA)
Several **hardware** layers: transistors, gates, CPU/Memory/IO

99% of users don't know hardware layers implementation
90% of users don't know implementation of any layer

That's okay, world still works just fine
But sometimes it is helpful to understand what's "under the hood"

What is a Computer?

- **Computer, 2. a.** A calculating-machine; esp. an automatic electronic device for performing mathematical or logical operations; freq. with defining word prefixed, as *analogue*, *digital*, *electronic computer*.

--- Oxford English Dictionary, circa 2000

Some Familiar Computers

[images from Wikipedia]

Where is the computer?

[images from Wikipedia]

Modern computing is as much about enhancing capabilities as data processing!!

Keeping up with the times

- Computer, 3. An electronic device (or system of devices) which is used to store, manipulate, and communicate information, perform complex calculations, or control or regulate other devices or machines, and is capable of receiving information . . . and of processing it in accordance with variable procedural instructions . . . used esp. for handling text, images, music, and video, accessing and **using the Internet, communicating with other people (e.g. by means of email), and playing games**.

--- Oxford English Dictionary, circa 2018

What is A Computer?

- Three key components
- Computation
- Communication
- Storage/memory

Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.

Image source: https://lbsitbytes2010.wordpress.com/2013/03/29/john-von-neumann-roll-no-15/

What is A Computer?

- Three key components
- Computation
- Communication
- Storage/memory

Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.

Image source: https://lbsitbytes2010.wordpress.com/2013/03/29/john-von-neumann-roll-no-15/

So what makes a computer a computer?

Having program stored as data is an <u>extremely</u> important step in the evolution of computer architectures

"Canonical" Computer Organization

Atmel ATmega8

state

I/O

logic

[image from Wikipedia]

Page 9 Atmel 8-bit AVR ATmega8 Databook

Recall: The Transformation Hierarchy

Computer Architecture (expanded view)

Computer Architecture (narrow view)

What We Will Cover (I)

- Combinational Logic Design
- Hardware Description Languages (Verilog)
- Sequential Logic Design
- Timing and Verification
- ISA (MIPS and LC3b)
- MIPS Assembly Programming

What We Will Cover (II)

- Microarchitecture Fundamentals
- Single-cycle Microarchitectures
- Multi-cycle and Microprogrammed Microarchitectures
- Pipelining
- Issues in Pipelining: Control & Data Dependence Handling, State Maintenance and Recovery, …
- Branch Prediction
- Out-of-Order Execution
- Superscalar Execution
- Other Paradigms: Dataflow, VLIW, Systolic, SIMD/GPUs, …

What We Will Cover (II)

- Memory Technology and Organization
- Caches
- Prefetching
- Virtual Memory

Processing Paradigms We Will Cover

- Pipelining
- Out-of-order execution
- Dataflow (at the ISA level)
- Superscalar Execution
- VLIW
- Decoupled Access-Execute
- Systolic Arrays
- SIMD Processing (Vector & Array, GPUs)

Computer Architecture is Engineering

- An applied discipline of finding and optimizing solutions under the joint constraints of demand, technology, economics, and ethics
- Thus, instances of what we practice evolve continuously
- Need to learn the principles that govern how to develop solutions to meet constraints
- **Don't memorize instances; understand why it is that way**

Historical Perspectives:
prelude to modern computer architecture

You should read "Historical Perspectives" at end of P&H chapters.
For more, read "A History of Modern Computing" by Ceruzzi.

Why Study Hardware?

- **Understand where computers are going**
  - Future capabilities drive the (computing) world
  - Real world-impact: no computer architecture → no computers!
- **Understand high-level design concepts**
  - The best system designers understand all the levels
    - Hardware, compiler, operating system, applications
- **Understand computer performance**
  - Writing well-tuned (fast) software requires knowledge of hardware
- **Write better software**
  - The best software designers also understand hardware
  - Understand the underlying hardware and its limitations
- **Design hardware**
  - Intel, AMD, IBM, ARM, Qualcomm, Apple, Oracle, NVIDIA, Samsung, …

Penn Legacy

- **ENIAC**: electronic numerical integrator and calculator
  - First operational general-purpose stored-program computer
  - Designed and built here by Eckert and Mauchly
  - See it in Moore 100!

- **First seminars on computer design**
  - Moore School Lectures, 1946
  - "*Theory and Techniques for Design of Electronic Digital Computers*"

Computer Architecture

- **Computer architecture**
  - Definition of ISA to facilitate implementation of software layers
  - The hardware/software interface

- **Computer micro-architecture**
  - Design processor, memory, I/O to implement ISA
  - Efficiently implementing the interface

- EE3043 is mostly about processor micro-architecture
  - "architecture" is also a vacuous term for "the design of things"
    - software architect, network architect, …

Application Specific Designs

- This class is about **general-purpose CPUs**
  - Processor that can do anything, run a full OS, etc.
  - E.g., Intel Atom/Core/Xeon, AMD Ryzen/EPYC, ARM M/A series
- In contrast to **application-specific chips**
  - Or **ASICs** (Application specific integrated circuits)
    - Also application-domain specific processors
  - Implement critical domain-specific functionality in hardware
    - Examples: video encoding, 3D graphics, machine learning
  - General rules
    - \- Hardware is less flexible than software
    - \+ Hardware more effective (speed, power, cost) than software
    - \+ Domain specific more "parallel" than general purpose
      - But mainstream processors are quite parallel as well

Forces on Innovation

- Timely innovations are rarely unique or original
- Similar constraints lead to similar engineering solutions

Beginnings of Digital Computing

- Industrial Revolution era's "hi-tech" in mechanization
  - steam engines
  - mechanical calculators,
  - Jacquard's loom:

    gears, pulleys,

    chains and

    punch cards

[images from Wikipedia]

Charles Babbage (1791-1871)

- Difference Engine, 1823: a special-purpose computer
  - evaluated polynomial functions by Newton's method of successive differences (requiring only additions)
  - eventually built by Georg and Edvard Schuetz in 1855

- Analytical Engine, 1833: a general-purpose computer
  - programmed by punch cards, “assembly language” included loops and branches
  - 1000 word data store, punch card I/O
  - unfortunately never completed (would have been 10x30 meters,

    steam-engine powered)

[images from Wikipedia]

100 Years of Technology Advances

- Mechanical, 1800s
  - gears, chains, pulleys, and steam power
  - punch cards!!
- Electromechanical, early 1900s
  - switches, relays, “acoustic” delay line “memory”
  - e.g. Harvard/IBM Mark 1, Aiken 1939~1944, 50ft long, 5ton, 750K parts, 3~6 sec per addition

Used ideas from Analytical Engine

- Electrical, mid 1900s and on
  - plugboards, vacuum tubes, CRTs
  - and later DRUM, core, transistors and so on . . . . .

Changing demands and economics?

ENIAC, 1946

Eckert and Mauchly, U of Penn

[images from Wikipedia]

- the first programmable electronic digital computer
- 18,000 vacuum tubes
- 30 ton, 80 by 8.5 feet
- 1900 additions per second
- 20 10-decimal-digit words (100-word core by 1952)
- Programmed by 3000 switches in the function table and plug-cables (later became stored program for faster program loading)

Proliferation in 40s and 50s

- From “Moore School Lectures”
  - ENIAC, Eckert & Mauchly, 1946 (revealed)
  - EDVAC, von Neumann, 1944~1952
  - EDSAC, Wilkes, 1949 (first stored-program built)
  - IAS, Bigelow, 1952
  - ORDVAC, SEAC, MANIAC, JOHNIAC, ILLIAC ...
- They were not alone:
  - ABC, Atanasoff and Berry, 39~42
  - Z3, Z4, Konrad Zuse late 30’s early 40’s
  - Colossus, Alan Turing, 1943
- Don’t forget software advances---Fortran was first done in 1954

Commercialization in the 50s

- UNIVAC (1951) the first commercial computer

  contract price $400K, actual cost ~$1M, sold 48 copies
- IBM 701 (1952) "leased" 19 units, $12K per month

  (www-1.ibm.com/ibm/history/exhibits/701/701_customers.html)
- IBM 650 (1953) sold ~2000 units at $200K ~ 400K
- IBM System/360, 1964 Redefined Industry!!
  - a family of **binary compatible** computers (previously, IBM had 4 incompatible lines)
  - 19 combinations of varying speed and memory capacity from $200K ~ $2M
  - ISA still alive today in *z/Architecture* mainframes

Cheaper or Faster in 60s and 70s

- Minicomputers
  - DEC PDP-8, 1965, $20K, size of large refrigerators
  - less powerful than “mainframes”, 10x cheaper
  - departmental computers, timesharing---PDP-11 and VAXs enjoyed extreme popularity in the 70s and 80s
- Supercomputers
  - performance at all cost!! (ECL, liquid-cool, hand-built)
  - biggest customers: national security, nuclear weapons, cryptography, (also aerospace, petroleum, automotive, pharmaceutical, sciences)
  - see Seymour Cray (1925~1996) on Wikipedia

What happened to these computer lines?

Early Examples

DEC PDP 8, 1963
an early mini

Xerox Alto, 1973
an early “PC” with
mouse and GUI

[images from Wikipedia]

Cray 3, 1993

90KW: liquid cooled by “Fluorinert”
$30,000,000
15 GFLOPS (1 sec on Cray3 ≈ 67 years ENIAC)

[images from Wikipedia]

The "Killer Micros" from 70s and on

[from Molecular Expressions]

- **Intel 4004, first single chip CPU**
  - 4-bit processor for calculator
  - 2,300 transistors
  - 16-pin DIP package
  - 740kHz (eight clock cycles per CPU cycle of 10.8 μsec)
  - ~100K OPs per second

download the actual schematic from www.4004.com

Intel Itanium (Montecito) 2004

[from Best Servers of 2004, Microprocessor Report, January 2005.]

- 64-bit processor
- 1.7 billion transistors
- 1.7 GHz, issue up to 8 instructions per cycle
- 26 MByte of cache!!

In ~30 years, about 100,000 fold growth in transistor count and performance!

The Era of Moore's Law

[http://www.intel.com/research/silicon/mooreslaw.htm]

Original article at http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=658762

The "Other" Moore's Law

© 2007 Elsevier, Inc. All rights reserved.

The Actual Moore's Law

[Cramming More Components Onto Integrated Circuits, G. E. Moore, 1965]

The End of Moore's Law?

[Tri-gate FinFET, Intel Newsroom, 2015]

Distance between silicon atoms ~ 500 pm

Moore's Law without Dennard Scaling

Under fixed power ceiling, more ops/second only achievable if less Joules/op?

Why multicores everywhere?

Energy per Instruction Trends in Intel® Microprocessors, Grochowski et al., 2006

Moore's Law Scaling with Cores

Future is about
Performance/Watt and Ops/Joule

This is a sign of desperation . . . .

Technology Trends

Technology that Drives

- Basic element
  - Solid-state **transistor** (i.e., electrical switch)
  - Building block of **integrated circuits (ICs)**

- What's so great about ICs? Everything
  + High performance, high reliability, low cost, low power
  + Lever of mass production

- Several kinds of integrated circuit families
  - **SRAM/logic**: optimized for speed (used for processors)
  - **DRAM**: optimized for density, cost, power (used for memory)
  - **Flash**: optimized for density, cost (used for storage)
  - Increasing opportunities for integrating multiple technologies

- Non-transistor storage and inter-connection technologies
  - Magnetic disks, optical storage, ethernet, fiber optics, wireless

Moore's Law - 1965

Today:
2^{35} transistors

Moore's Law today

data c/o WikiChip, Wikipedia

Moore's Law today

data c/o WikiChip, Wikipedia

Revolution I: The Microprocessor

- **Microprocessor revolution**
  - One significant technology threshold was crossed in 1970s
  - Enough transistors (~25K) to put a 16-bit processor on one chip
  - Huge performance advantages: fewer slow chip-crossings
  - Even bigger cost advantages: one "stamped-out" component

- Microprocessors have allowed new market segments
  - Desktops, CD/DVD players, laptops, game consoles, set-top boxes, mobile phones, digital camera, mp3 players, GPS, automotive

- And replaced incumbents in existing segments
  - Microprocessor-based system replaced supercomputers, "mainframes", "minicomputers", "desktops", etc.

First Microprocessor

- Intel 4004 (1971)
  - Application: calculators
  - Technology: 10,000 nm

  - 2300 transistors
  - 13 mm^2
  - 108 KHz
  - 12 Volts

  - 4-bit data
  - Single-cycle datapath

Revolution II: Implicit Parallelism

- Then to **extract implicit instruction-level parallelism**
  - Hardware provides parallel resources, figures out how to use them
  - Software is oblivious

- Initially using pipelining …
  - Which also enabled increased clock frequency
- … caches …
  - Which became necessary as processor clock frequency increased
- … and integrated floating-point
- Then deeper pipelines and branch speculation
- Then multiple instructions per cycle (superscalar)
- Then dynamic scheduling (out-of-order execution)

- We will build many of these things!

Pinnacle of Single-Core Microprocessors

- Intel Pentium4 (2003)
  - Application: desktop/server
  - Technology: 90nm

  - 55M transistors
  - 101 mm^2
  - 3.4 GHz
  - 1.2 Volts

  - 32/64-bit data (16x)
  - 22-stage pipelined datapath
  - 3 instructions per cycle (superscalar)
  - Two levels of on-chip cache
  - data-parallel vector (SIMD) instructions, hyperthreading

Revolution III: Explicit Parallelism

- Then to support **explicit data & thread level parallelism**
  - Hardware provides parallel resources, software specifies usage
  - Why? diminishing returns on instruction-level-parallelism
- First using (subword) vector instructions…, Intel's SSE
  - One instruction does four parallel multiplies
- … and general support for multi-threaded programs
  - Coherent caches, hardware synchronization primitives
- Then using support for multiple concurrent threads on chip
  - First with single-core multi-threading, now with multi-core
- Graphics processing units (GPUs) are highly parallel

Modern Multicore Processor

- AMD EPYC 7H12
  - Application: server
  - Technology: 7nm

  - 39.5B transistors
  - 1008 mm^2
  - 2.6 to 3.3 Ghz
  - 256-bit data (2x)
  - 19-stage pipelined datapath
  - 4 instructions per cycle
  - 292MB of on-chip cache
  - data-parallel vector (SIMD) instructions, hyperthreading
  - **64-core multicore**

Historical Microprocessor Evolution

Feature	Intel 4004	Intel Pentium 4 Prescott	AMD EPYC Rome
release date	1971	2004	2019
transistor size	10,000 nm	90 nm	7 nm, 14 nm
transistor count	2,300	125M	39.5B
area	13 mm^2	112 mm^2	1008 mm^2
frequency	740 KHz	3.8 GHz	2.6-3.3 GHz
data width	4-bit	64-bit	256-bit
pipeline stages	n/a	31	19
pipeline width	n/a	3	4
core count	1	1	64
on-chip cache	n/a	1MB	292MB

Revolution IV: Accelerators

- Combining **multiple** kinds of compute engines in one die
  - not just homogenous collection of cores
  - System-on-Chip (SoC) is one common example in mobile space
- Lots of stuff on the chip beyond just CPUs
  - Graphics Processing Units (GPUs)
    - throughput-oriented specialized multicore processors
    - good for gaming, machine learning, computer vision, …
  - Special-purpose logic
    - media codecs, radios, encryption, compression, machine learning
- Excellent energy efficiency and performance
  - extremely complicated to program!

SNAPDRAGON 805 PROCESSOR

Faster performance and more multitasking with Krait 450 CPU at up to 2.7 GHz

Console quality gaming with new generation Adreno 420 GPU

More power-efficient apps and system processing with the Hexagon™ QDSP6

Capture and play back Ultra HD video and enjoy 7.1 surround sound on the go or at home with advanced video and audio engines

Get more use and greater accuracy from sensor-intensive apps with the dedicated Snapdragon Sensor Engine

Stay connected and stream large files fast with industry leading connectivity, including the world's most advanced 4G LTE and VIVE™ 2-stream 802.11ac Wi-Fi

Capture sharper photos, even in low light, with the mobile industry's first dual ISP

Enjoy Ultra HD resolution content on Ultra HD-capable mobile devices and Ultra HD TVs with the Snapdragon Display Processor

Find your way outdoors and indoors with IZat GNSS with support for GPS, Glonass and BeiDou constellations

c/o Qualcomm

Zedboard SoC

Cerebras Wafer-Scale Engine

- giant 8.5” square chip!
- full of deep learning accelerators
- 18GB on-chip memory
- 9 PB/sec on-chip memory bandwidth
- TSMC 16nm transistors
- ~$2.5M each?

Technology Disruptions

- Classic examples:
  - transistor
  - microprocessor
- More recent examples:
  - flash-based solid-state storage
  - shift to accelerators
- Nascent disruptive technologies:
  - non-volatile memory ("disks" as fast as DRAM)
  - Chip stacking (also called "3D die stacking")
- The end of Moore's Law
  - "If something can't go on forever, it must stop eventually"
  - Transistor speed/energy efficiency not improving like before

"Golden Age of Computer Architecture"

- Hennessy & Patterson, 2018 Turing Laureates
- the end of Dennard Scaling & Moore's Law means no more free performance
  - "The next decade will see a Cambrian explosion of novel computer architectures"

Themes

- Parallelism
  - enhance system performance by doing multiple things at once
  - instruction-level parallelism, multicore, GPUs, accelerators
- Caching
  - exploiting locality of reference: storage hierarchies
  - try to provide the illusion of a single large, fast memory