

CPTR-380

Course Introduction

Professor Larry Aamodt

Key topics this quarter

IC design \leftrightarrow Computer Architecture

Moore's law, Power wall

- Characterizing computer performance
- High level language \leftrightarrow Assembly language
 - Instruction set design
- Computer arithmetic
- Processor design
 - Instruction decoding, pipelining
- Memory hierarchy & design
- Parallel processing
- How to increase computer performance

What You Will Learn

- How programs are translated into the machine language
 - And how the hardware executes them
- The hardware/software interface
- What determines program performance
 - And how it can be improved
- How hardware designers improve performance
- What is parallel processing

Things to get from this class

- A working vocabulary that includes:

DRAM, pipelining, cache hierarchies, I/O busses, virtual memory, floating-point processing, etc.

A knowledge base that helps you make intelligent choices i.e.

Deciding which processor to buy:

4 GHz Core i7 or 3.5 GHz Athlon (performance vs. power vs portability).

A stepping stone toward the fields of:

IC design, compiler design, operating systems design.

Today's topics

- Introductory material
 - Classes of computers
 - The Post PC era
 - Great ideas in computer design
 - Components of a computer
-

The Computer Revolution

- Progress in computer technology
 - Underpinned by domain-specific accelerators
- Makes novel applications feasible
 - Computers in automobiles
 - Cell phones
 - Human genome project
 - World Wide Web
 - Search Engines
- Computers are pervasive

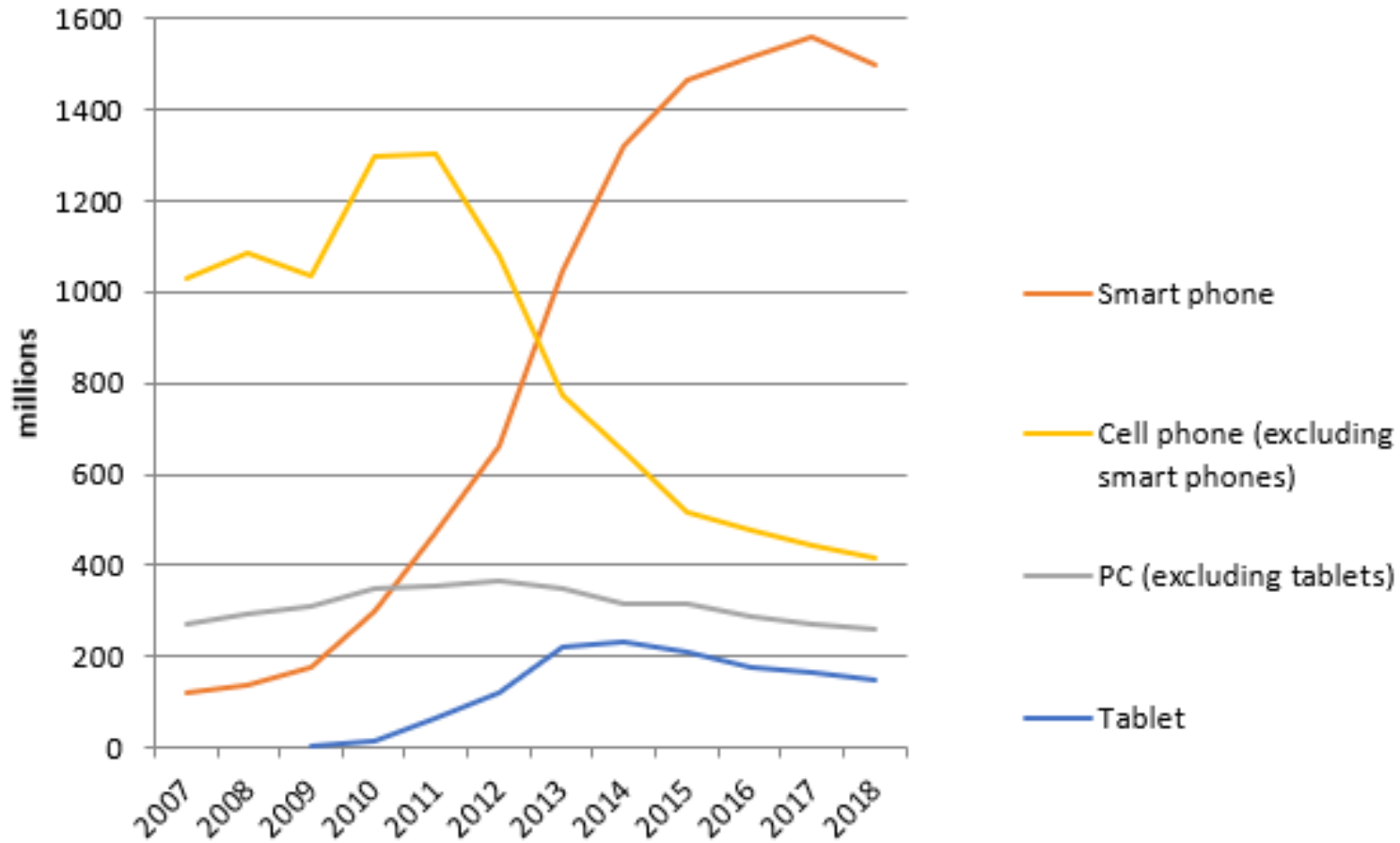
Classes of Computers

- Personal computers
 - General purpose, variety of software
 - Subject to cost/performance tradeoff
- Server computers
 - Network based
 - High capacity, performance, reliability
 - Range from small servers to building sized

Classes of Computers

- Supercomputers
 - Type of server
 - High-end scientific and engineering calculations
 - Highest capability but represent a small fraction of the overall computer market
- Embedded computers
 - Hidden as components of systems
 - Stringent power/performance/cost constraints

The PostPC Era



The PostPC Era

- Personal Mobile Device (PMD)
 - Battery operated
 - Connects to the Internet
 - Hundreds of dollars
 - Smart phones, tablets, electronic glasses
- Cloud computing
 - Warehouse Scale Computers (WSC)
 - Software as a Service (SaaS)
 - Portion of software run on a PMD and a portion run in the Cloud
 - Amazon and Google

Understanding Performance

- Algorithm
 - Determines number of operations executed
- Programming language, compiler, architecture
 - Determine number of machine instructions executed per operation
- Processor and memory system
 - Determine how fast instructions are executed
- I/O system (including OS)
 - Determines how fast I/O operations are executed

Seven Great Ideas

- Use *abstraction* to simplify design
- Make the *common case fast*
- Performance *via parallelism*
- Performance *via pipelining*
- Performance *via prediction*
- *Hierarchy* of memories
- *Dependability* *via* redundancy



COMMON CASE FAST



PARALLELISM



PIPELINING



PREDICTION

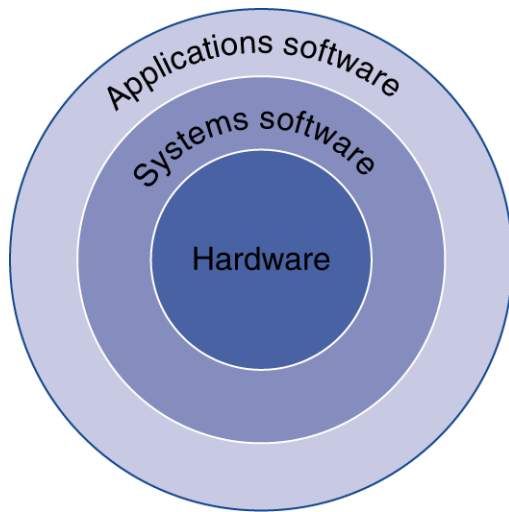


HIERARCHY



DEPENDABILITY

Below Your Program



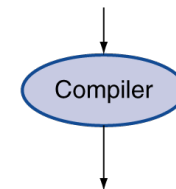
- Application software
 - Written in high-level language
- System software
 - Compiler: translates HLL code to machine code
 - Operating System: service code
 - Handling input/output
 - Managing memory and storage
 - Scheduling tasks & sharing resources
- Hardware
 - Processor, memory, I/O controllers

Levels of Program Code

- High-level language
 - Level of abstraction closer to problem domain
 - Provides for productivity and portability
- Assembly language
 - Textual representation of instructions
- Hardware representation
 - Binary digits (bits)
 - Encoded instructions and data

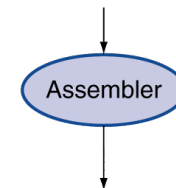
High-level
language
program
(in C)

```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```



Assembly
language
program
(for MIPS)

```
swap:
  muli $2, $5,4
  add $2, $4,$2
  lw $15, 0($2)
  lw $16, 4($2)
  sw $16, 0($2)
  sw $15, 4($2)
  jr $31
```

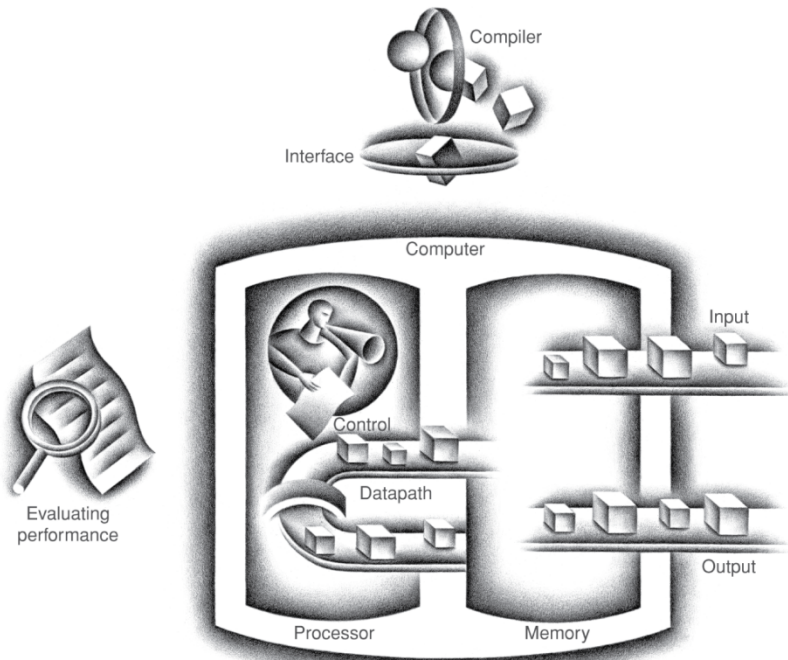


Binary machine
language
program
(for MIPS)

```
000000001010000100000000000011000
000000000000110000001100000100001
100011000110001000000000000000000
100011001111001000000000000000100
101011001111001000000000000000000
101011000110001000000000000000100
00000011111000000000000000001000
```

Components of a Computer

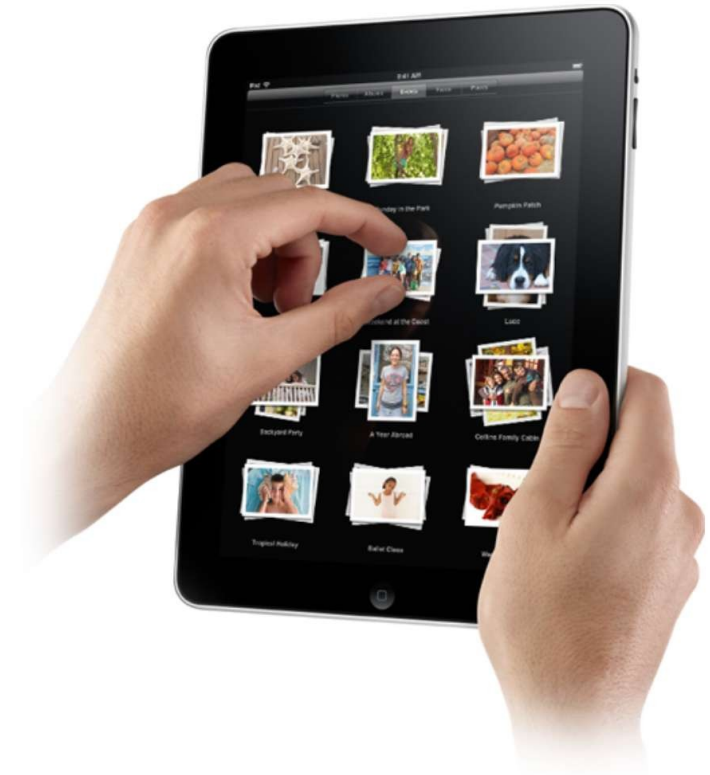
The BIG Picture



- Same components for all kinds of computer
 - Desktop, server, embedded
- Input/output includes
 - User-interface devices
 - Display, keyboard, mouse
 - Storage devices
 - Hard disk, CD/DVD, flash
 - Network adapters
 - For communicating with other computers

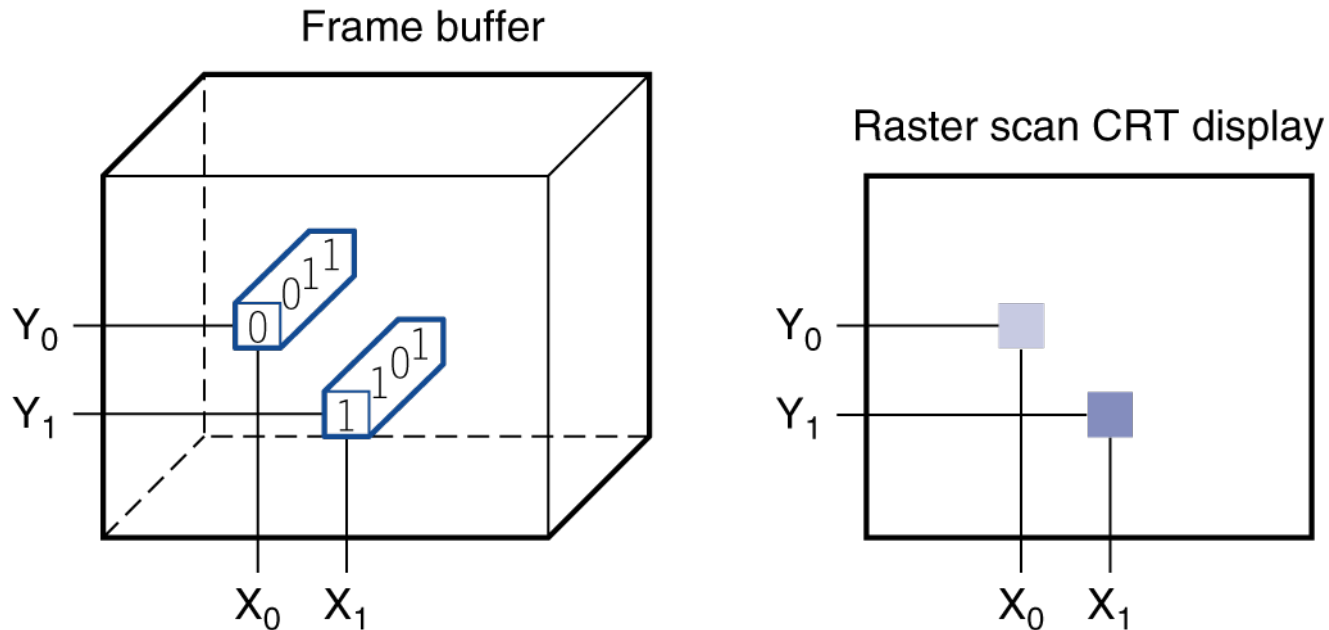
Touchscreen

- PostPC device
- Supersedes keyboard and mouse
- Resistive and Capacitive types
 - Most tablets, smart phones use capacitive
 - Capacitive allows multiple touches simultaneously

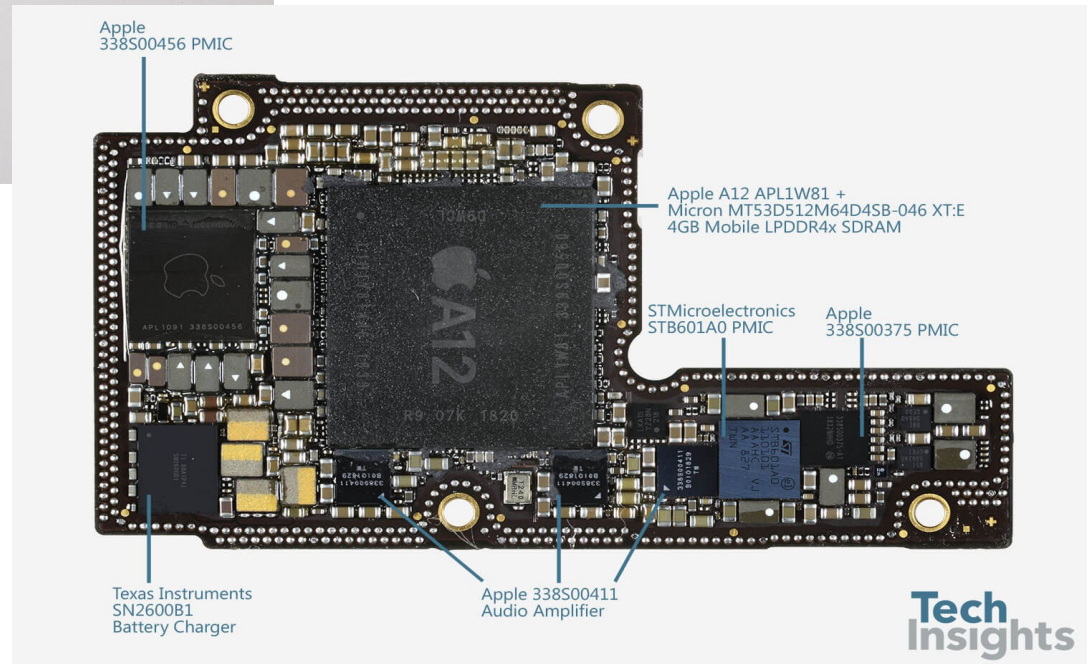
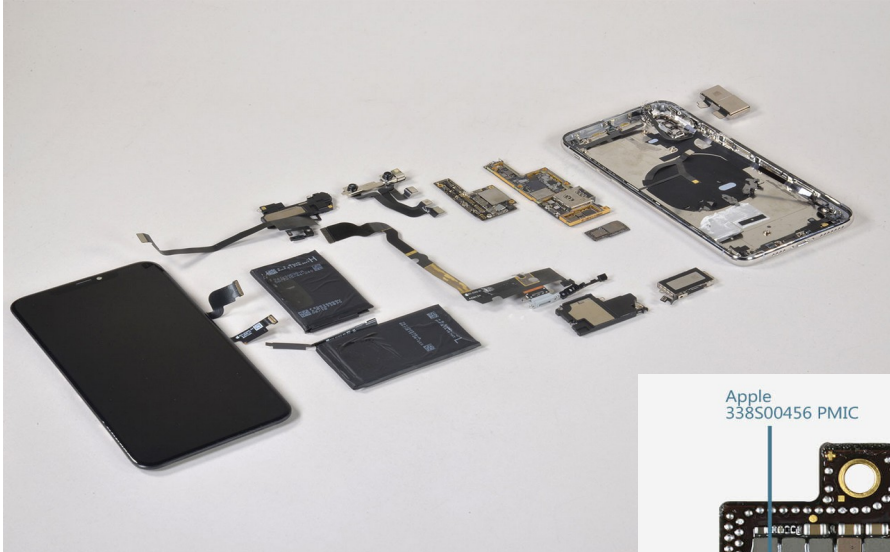


Through the Looking Glass

- LCD screen: picture elements (pixels)
 - Mirrors content of frame buffer memory



Opening the Box

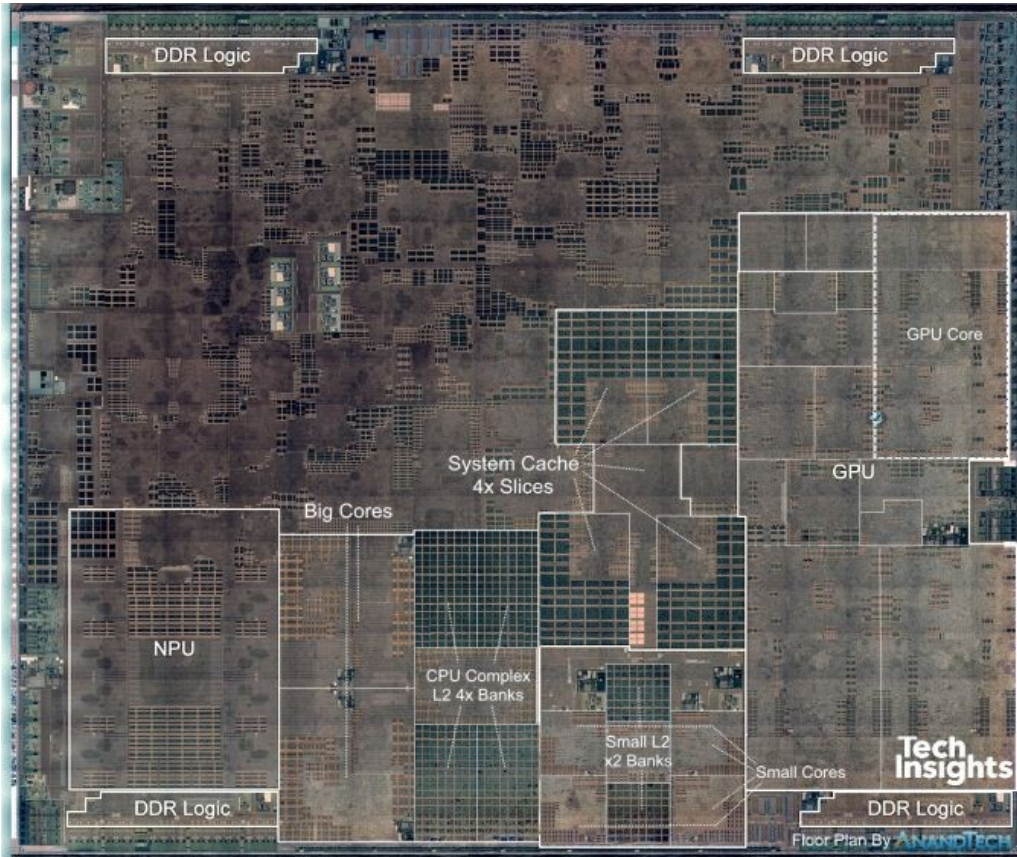


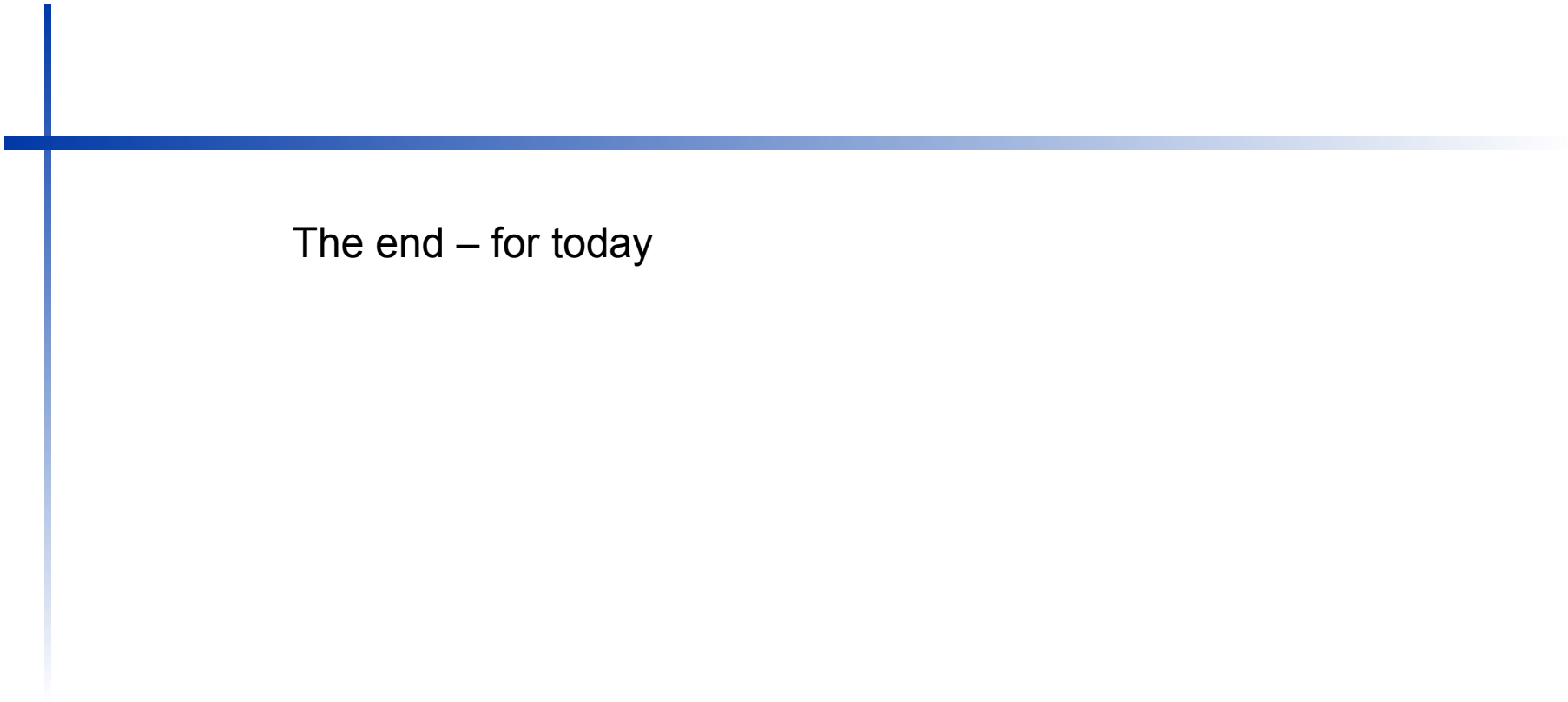
Inside the Processor (CPU)

- Datapath: performs operations on data
- Control: sequences datapath, memory, ...
- Cache memory
 - Small fast SRAM memory for immediate access to data

Inside the Processor

- A12 processor





The end – for today

Slides in this lecture are a mix of slides provided by the textbook author and material produced by this instructor.