EE 445 Computer Architecture I

Fall 2025

Administrative Details-Instructors

- Section 1: Ece Güran Schmidt
  - Email: eguran@metu.edu.tr
  - Web page: http://users.metu.edu.tr/eguran/
  - Office: A402
- Section 2: Ahmed Hareedy
  - Email: ahareedy@metu.edu.tr
  - Web page: http://users.metu.edu.tr/ahareedy/
  - Office: D124-2
- You can check out our web pages to see what we do in our research

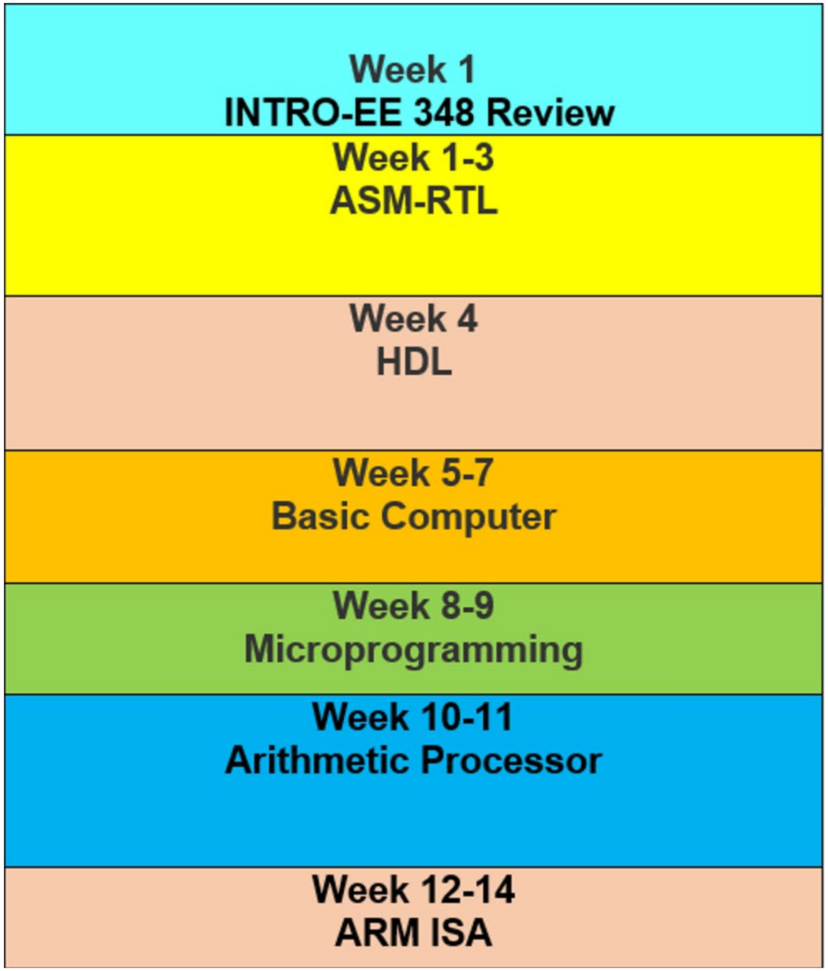
Administrative Details

- Schedule
  - Section 1: Tuesday 14:40-15:30 Thursday 10:40-12:30, @ A209
  - Section 2: Tuesday15:40-17:30 Thursday 16:40-17:30, @ A206
- Course Conduct
  - In-class lectures and problem solving sessions. Video lectures of the previous years (subject to change) and lecture notes are posted in odtuclass.
- Required
  - Prerequisite(s): EE348
  - Core course for Computer Option

Grading

- 4 Short Exams: 52% (13% each)
- Final exam: 36%
- HDL Homeworks: 12%
- 5% bonus for attendance >=80%

Schedule



Week 1
INTRO-EE 348 Review

Week 1-3
ASM-RTL

Week 4
HDL

Week 5-7
Basic Computer

Week 8-9
Microprogramming

Week 10-11
Arithmetic Processor

Week 12-14
ARM ISA

Administrative Details

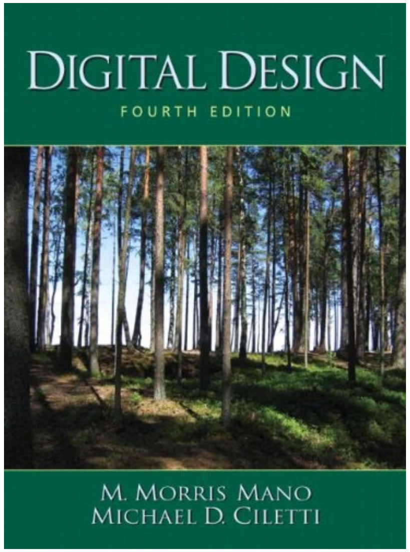
- **Follow**
  - https://odtuclass.metu.edu.tr/

  for lecture slides, all class material, recorded lecture videos and announcements
  - Your e123456@metu.edu.tr email
- **Communication**
  - Preferred communication mean: E-MAIL
  - Send with subject **including** EE445 (no guarantee of reply otherwise)

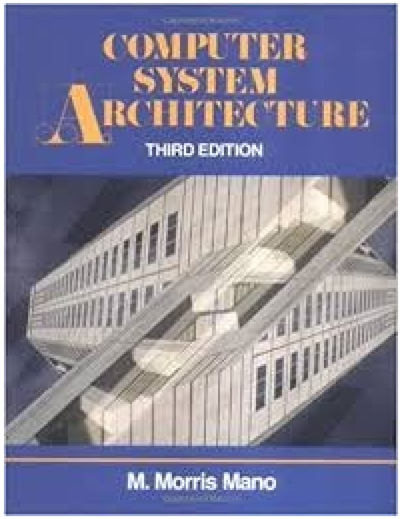
Text Books

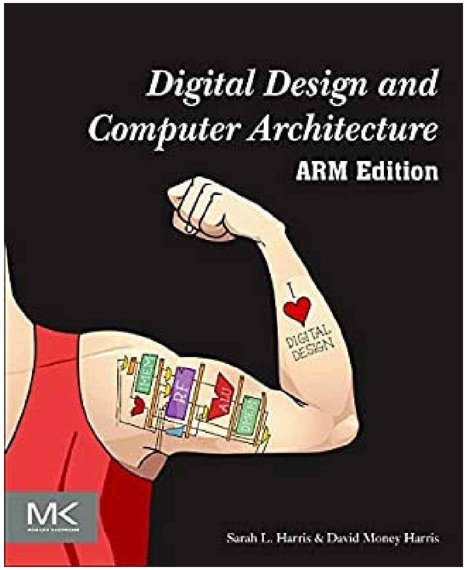
Copyright notice: Lecture Note Slides are compiled from the teaching material of these books, previous lecture notes of EE445 and additional resources. Part of the slides are entirely created by the instructors.

Digital Design (4th Edition)
M. Morris Mano, Michael D. Ciletti
Published by Prentice Hall, 2006



Computer System Architecture 3rd Ed., M. Morris Mano Prentice Hall, 1992
Computer System Architecture 2nd Ed., M. Morris Mano Prentice Hall, 1982




Digital Design and Computer Architecture
ARM Edition
I ♥ DIGITAL DESIGN
MK
Sarah L. Harris & David Money Harris

Harris & Harris, “Digital Design and Computer Architecture. ARM Edition”, 1st Ed., Kaufmann, 2015.
We go on with this text book for EE446

Course Objective (Why should you take this course?)

- A smooth extension of EE348
- Describes how a computer works at EE348 level of detail on a simple fictitious Basic Computer
- Preparation for the advanced topics: pipelining, memory and I/O organization that are covered in EE446

Course Outline

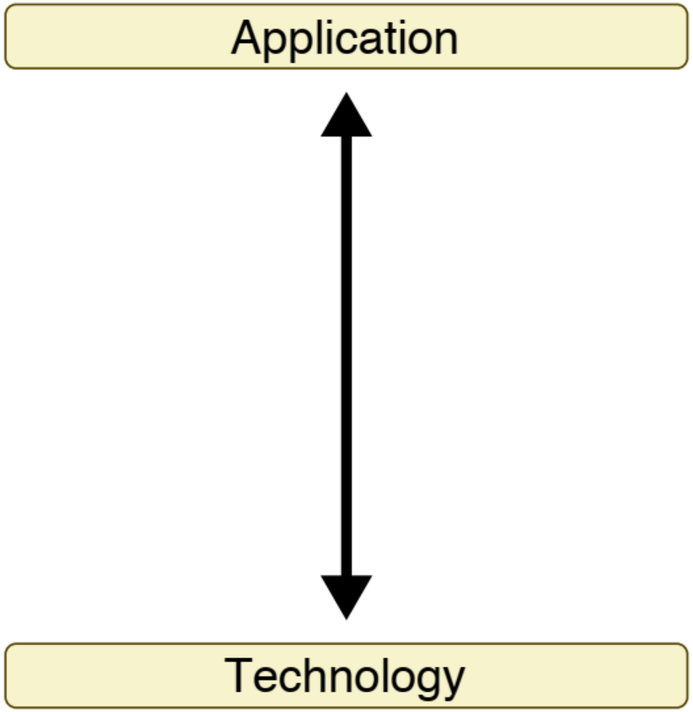
- Introduction to Computer Architecture
- EE348 review
- Algorithmic State Machine
- Register Transfer Language
- HDL
- Basic Computer Architecture
- Computer Organization and Microprogramming
- Arithmetic Processor Design
- ARM Instruction Set Architecture

Introduction to Computer Architecture

Resources:

http://www.csl.cornell.edu/courses/ece4750
https://safari.ethz.ch/digitaltechnik/spring2020/doku.php
Computer Architecture
A Quantitative Approach, Sixth Edition

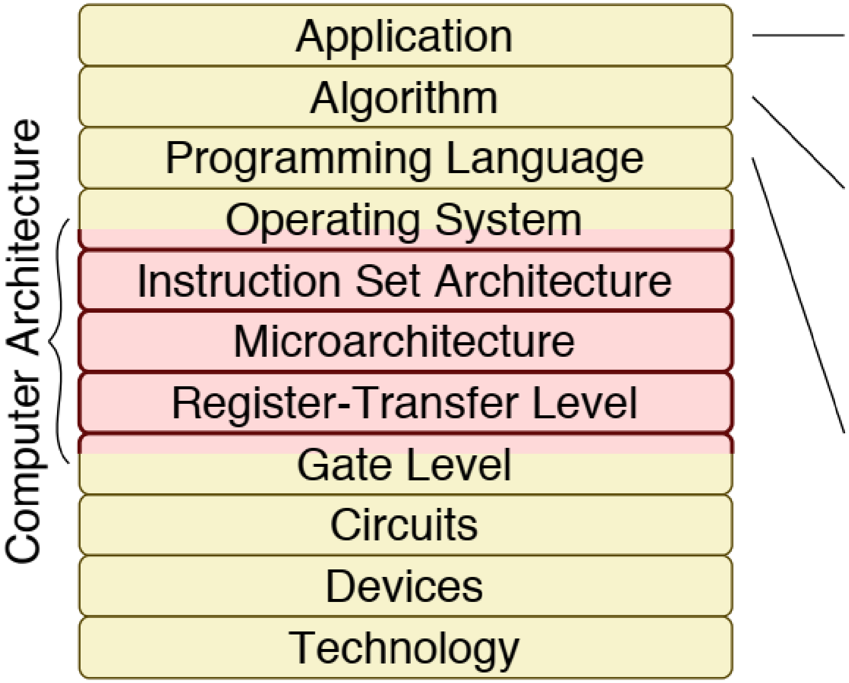
Computer Architecture



- abstraction/implementation layers
  - to execute information processing applications
  - efficiently using available manufacturing technologies

https://www.csl.cornell.edu/courses/ece4750/handouts/ece4750_overview.pdf

Computer Architecture



Application

Algorithm

Programming Language

Operating System

Instruction Set Architecture

Microarchitecture

Register-Transfer Level

Gate Level

Circuits

Devices

Technology

Computer Architecture

https://www.csl.cornell.edu/courses/ece4750/handouts/ece4750_overview.pdf

Sort an array of numbers

2,6,3,8,4,5 -> 2,3,4,5,6,8

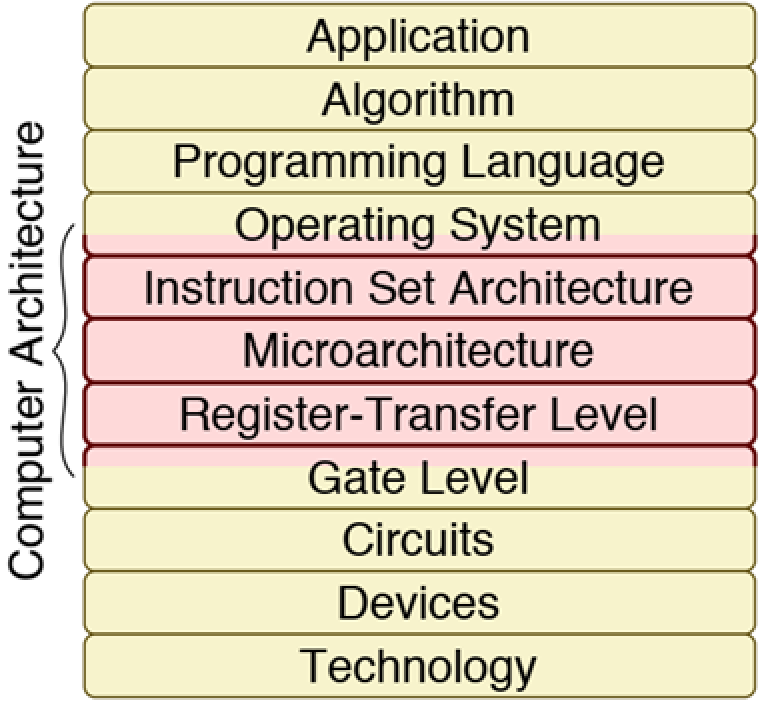
Out-of-place selection sort algorithm

1. Find minimum number in array
2. Move minimum number into output array
3. Repeat steps 1 and 2 until finished

C implementation of selection sort

```
void sort( int b[], int a[], int n ) {
  for ( int idx, k = 0; k < n; k++ ) {
    int min = 100;
    for ( int i = 0; i < n; i++ ) {
      if ( a[i] < min ) {
        min = a[i];
        idx = i;
      }
    }
    b[k]   = min;
    a[idx] = 100;
  }
}
```

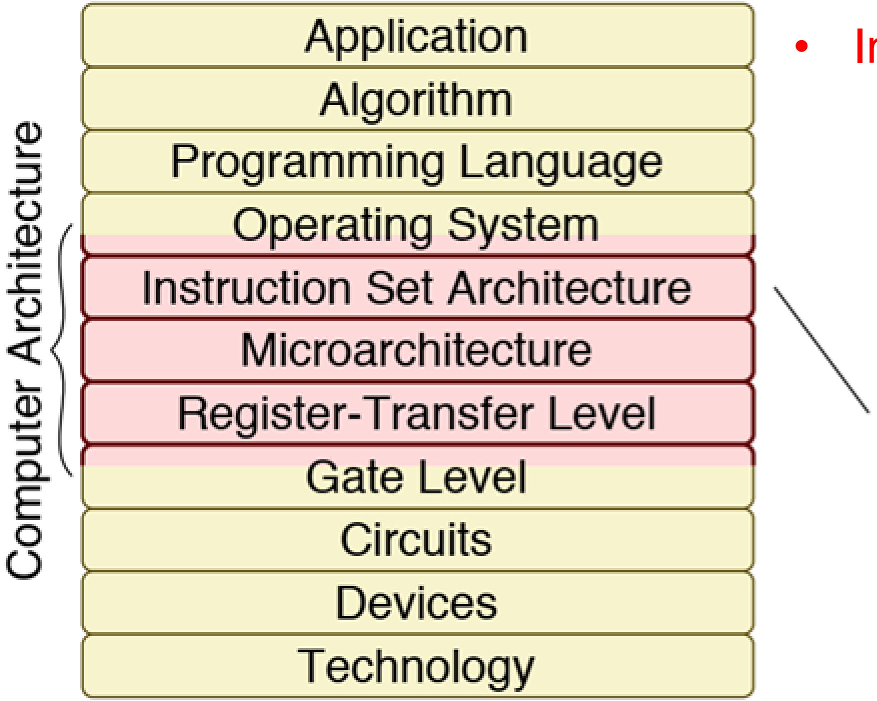
Computer Architecture



Mac OS X, Windows, Linux

Handles low-level hardware management

Computer Architecture



Application

Algorithm

Programming Language

Operating System

Instruction Set Architecture

Microarchitecture

Register-Transfer Level

Gate Level

Circuits

Devices

Technology

Computer Architecture

- Instruction Set Architecture (ISA):
  - Structure and behavior of the computer as seen by the programmer
  - There can be many implementations of the same ISA

MIPS32 Instruction Set

Instructions that machine executes

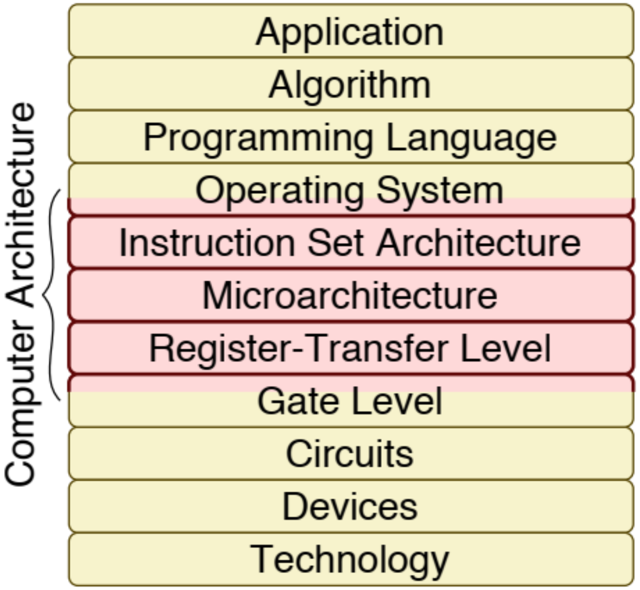
```
blez  $a2, done
move  $a7, $zero
li    $t4, 99
move  $a4, $a1
move  $v1, $zero
li    $a3, 99
lw    $a5, 0($a4)
addiu $a4, $a4, 4
slt   $a6, $a5, $a3
movn  $v0, $v1, $a6
addiu $v1, $v1, 1
movn  $a3, $a5, $a6
```

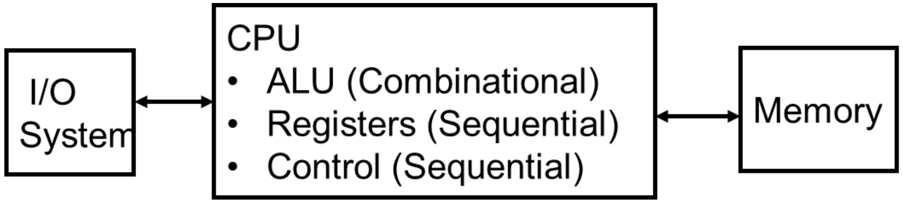
Instruction Set Architecture (ISA)

- Represents
  - all the information necessary to write a machine language program that will run correctly on the machine
  - the conceptual structure and functional behavior
- Abstracts away
  - the organization of the data flows and controls
  - the logic design
  - the physical implementation.
- Enables implementations of varying cost and performance to run identical software
- Includes
  - Addressing modes
  - Operand specifications
  - Operation specifications
  - Control flow instructions

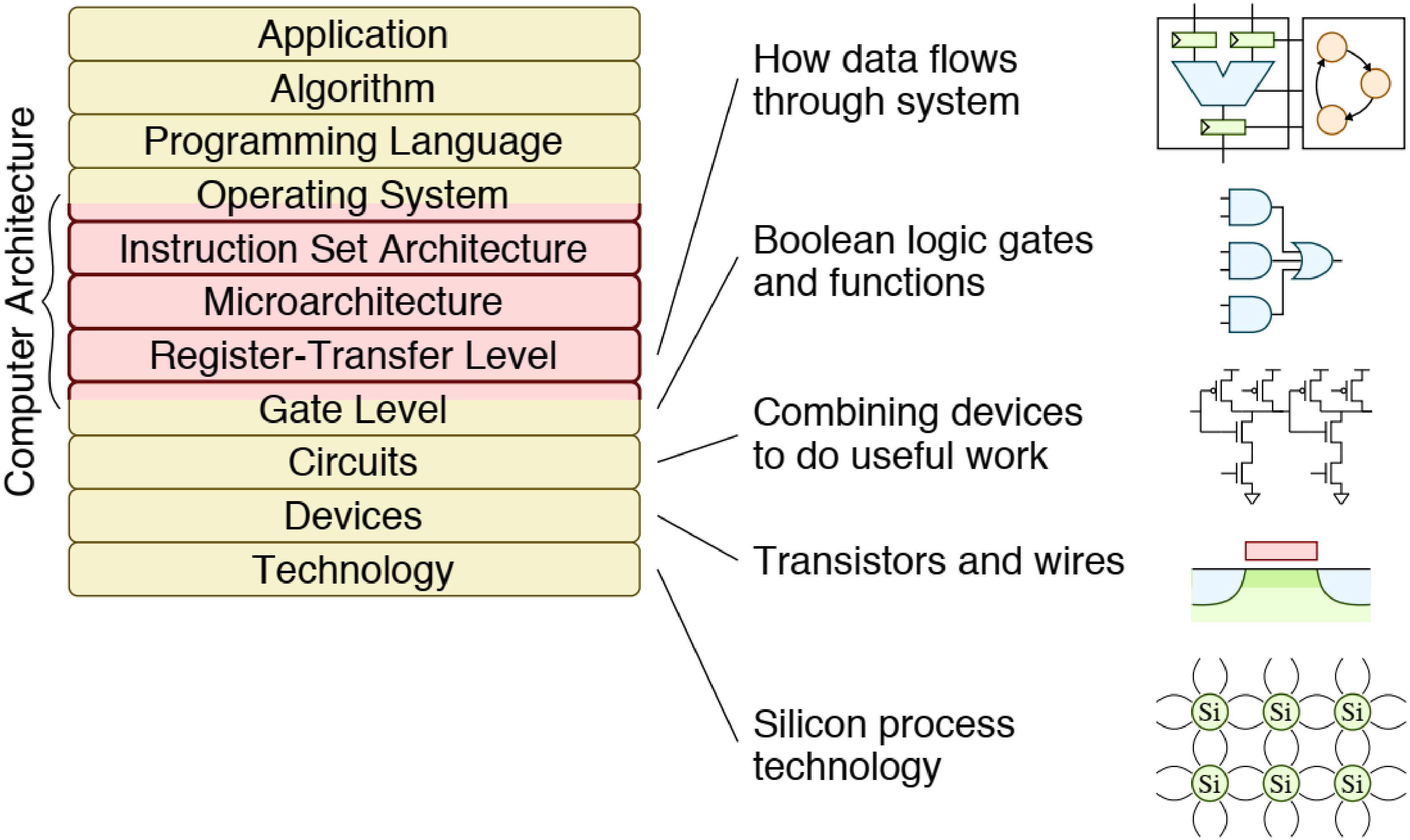
Microarchitecture

- Microarchitecture/Organization: The specific arrangement of registers, ALUs, finite state machines (FSMs), memories, and other logic building blocks needed to implement an ISA.
- Example: AMD Opteron and the Intel Core i7 implement the 80x86 instruction set with very different pipeline and cache organizations





Computer Architecture



Application
Algorithm
Programming Language
Operating System
Instruction Set Architecture
Microarchitecture
Register-Transfer Level
Gate Level
Circuits
Devices
Technology

Computer Architecture

How data flows through system

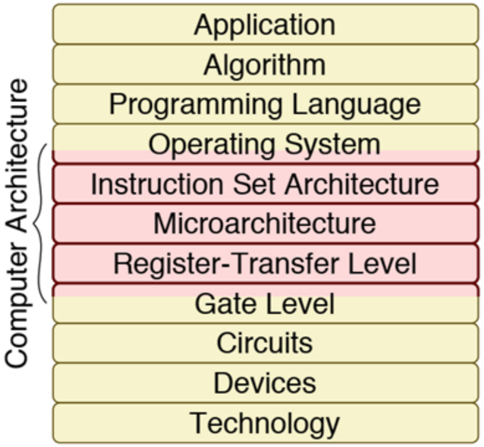
Boolean logic gates and functions

Combining devices to do useful work

Transistors and wires

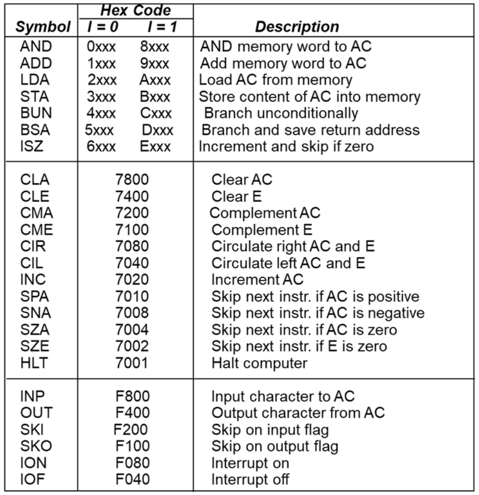
Silicon process technology

EE445 Coverage



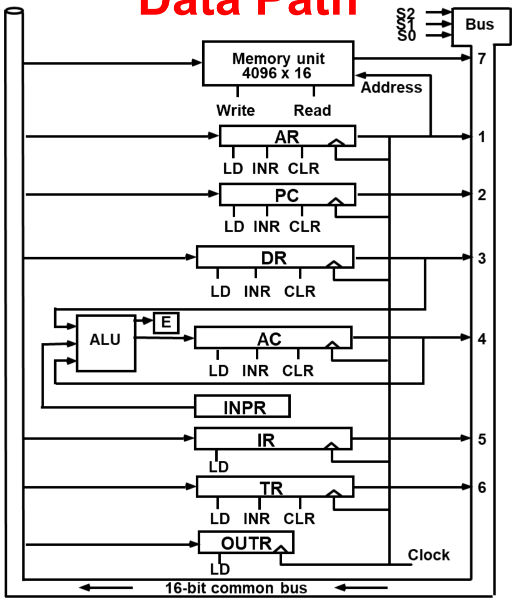
EE445 focuses on these layers using a fictitious Basic Computer

Instruction Set

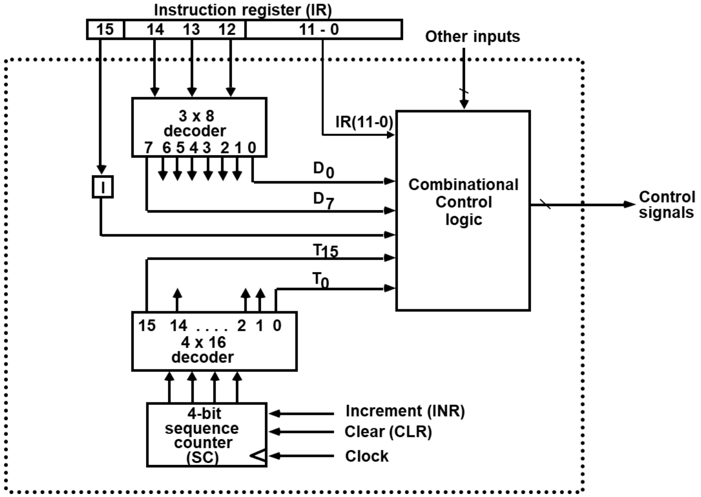


Symbol	Hex Code I = 0	 I = 1	Description
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load AC from memory
STA	3xxx	Bxxx	Store content of AC into memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear AC
CLE	7400		Clear E
CMA	7200		Complement AC
CME	7100		Complement E
CIR	7080		Circulate right AC and E
CIL	7040		Circulate left AC and E
INC	7020		Increment AC
SPA	7010		Skip next instr. if AC is positive
SNA	7008		Skip next instr. if AC is negative
SZA	7004		Skip next instr. if AC is zero
SZE	7002		Skip next instr. if E is zero
HLT	7001		Halt computer
INP	F800		Input character to AC
OUT	F400		Output character from AC
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

Data Path

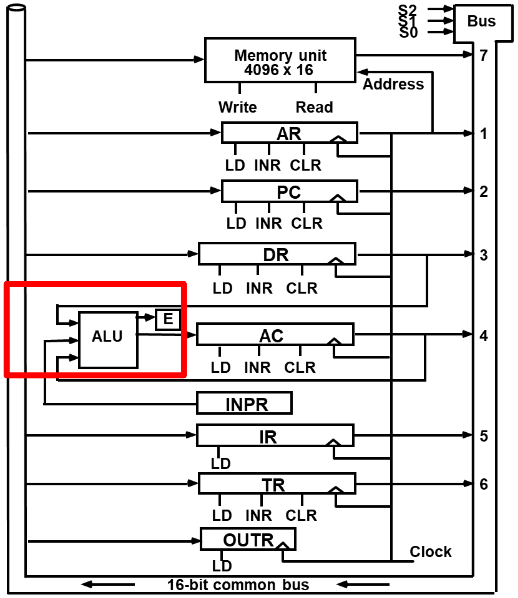


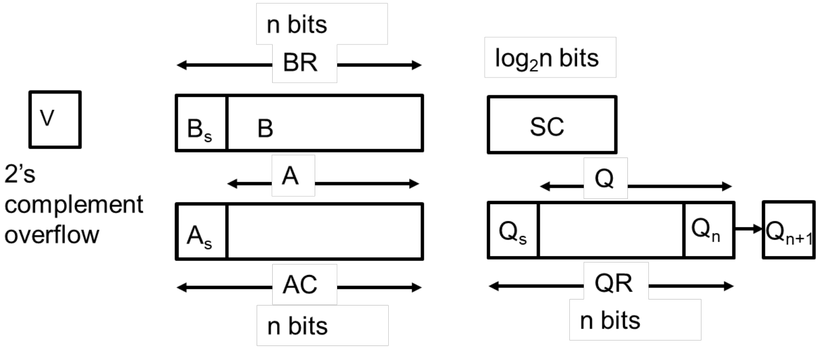
Controller



EE445 Coverage

Data Path

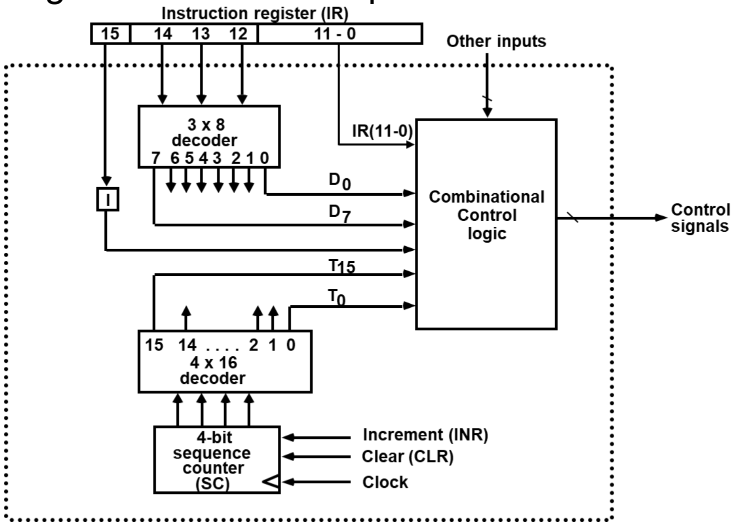




ALU implementation, hardware algorithms for multiplication and division➔ Needs EE348 refresher ☺

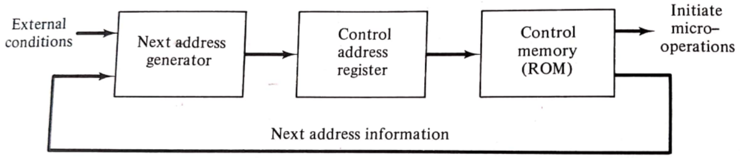
Hardwired control

Control signals are circuit outputs



Microprogrammed Control

Control signals are the control memory word contents

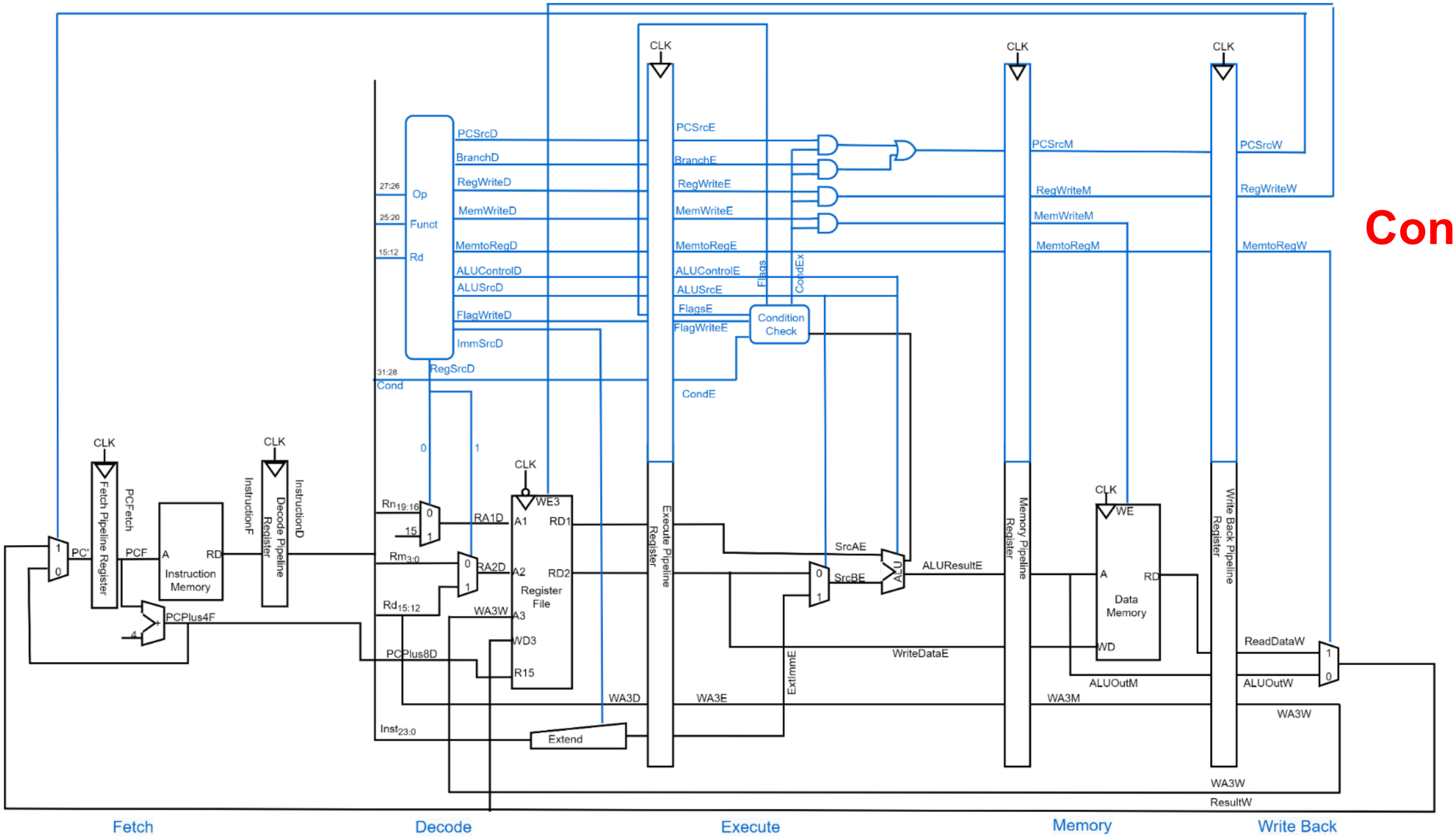


A Sneak Peek into EE446 ☺

A Pipelined Microarchitecture for a *representative* subset of ARM ISA

```
LDR Rd, [Rn, imm12]
STR Rd, [Rn, imm12]
ADD Rd, Rn, imm8
B BTA
```

Instruction Set

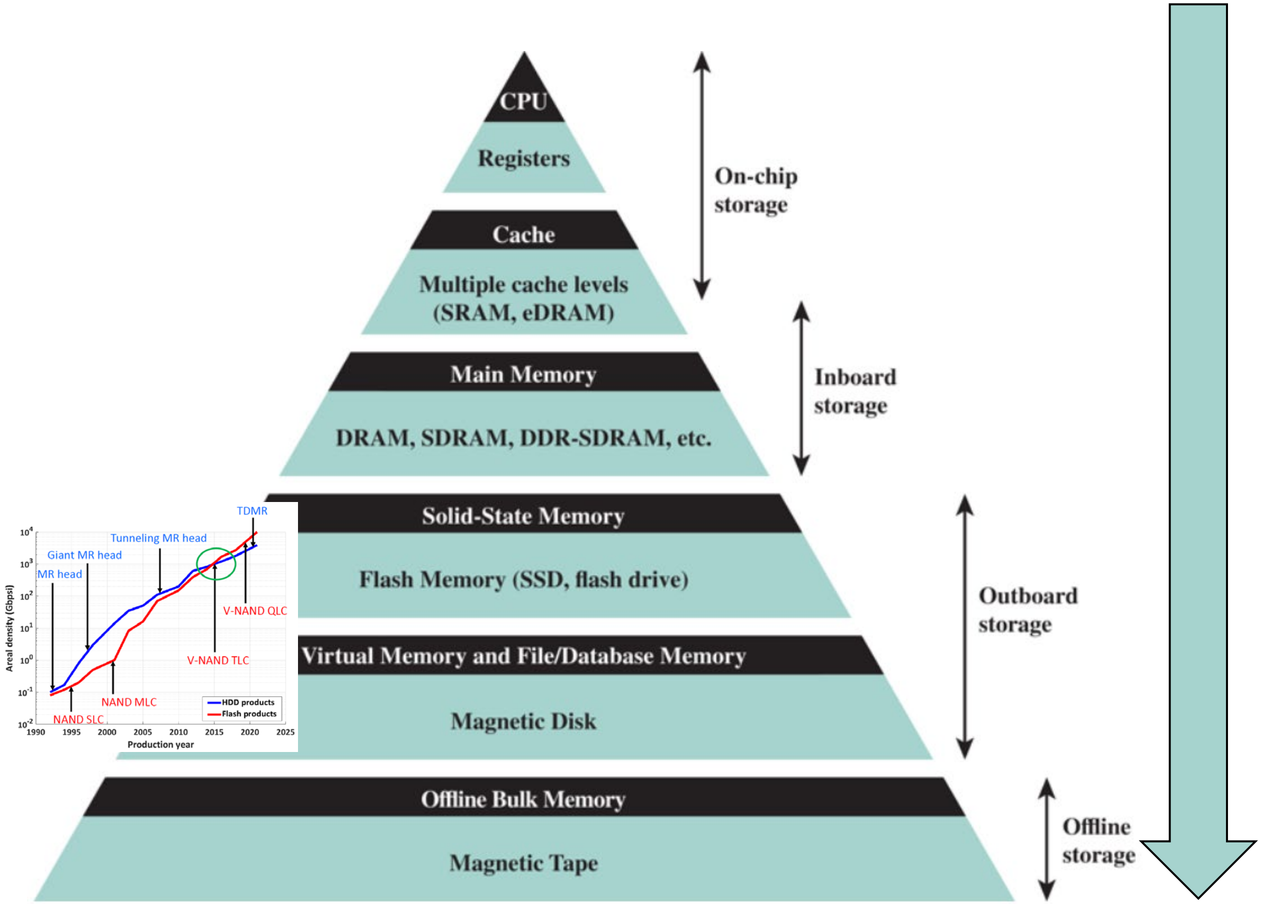


Controller

Data Path

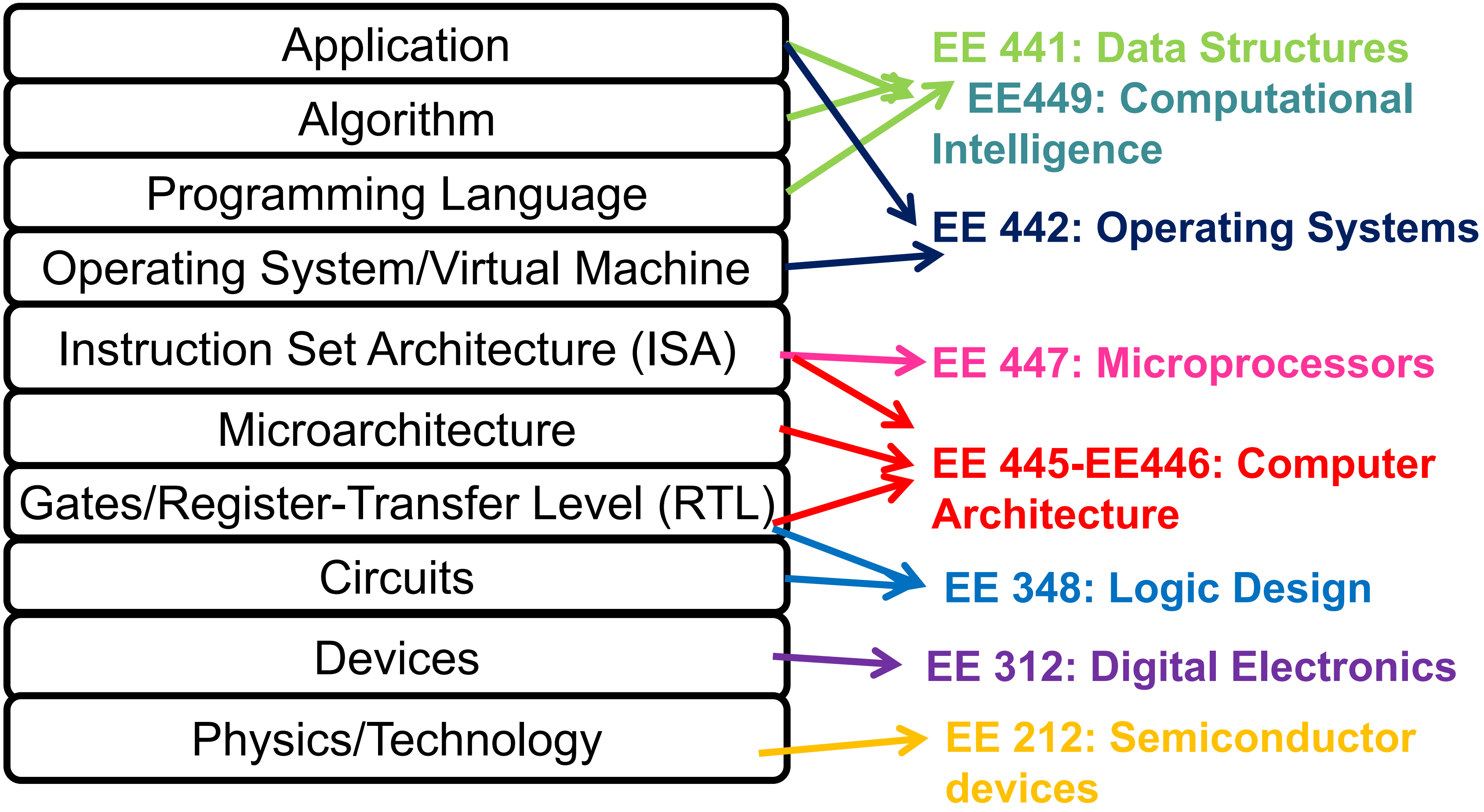
A Sneak Peek into EE446 ☺

Memory Hierarchy



- Cost per byte decreases
- Average access time increases
- Average data transfer rate decreases
- Total memory size increases
- Frequency of access decreases ➔Principle of locality
- Data contained in a lower level are a superset of the next higher level ➔ Inclusion property

Computer Architecture in METU EE



Application

Algorithm

Programming Language

Operating System/Virtual Machine

Instruction Set Architecture (ISA)

Microarchitecture

Gates/Register-Transfer Level (RTL)

Circuits

Devices

Physics/Technology

EE 441: Data Structures

EE449: Computational Intelligence

EE 442: Operating Systems

EE 447: Microprocessors

EE 445-EE446: Computer Architecture

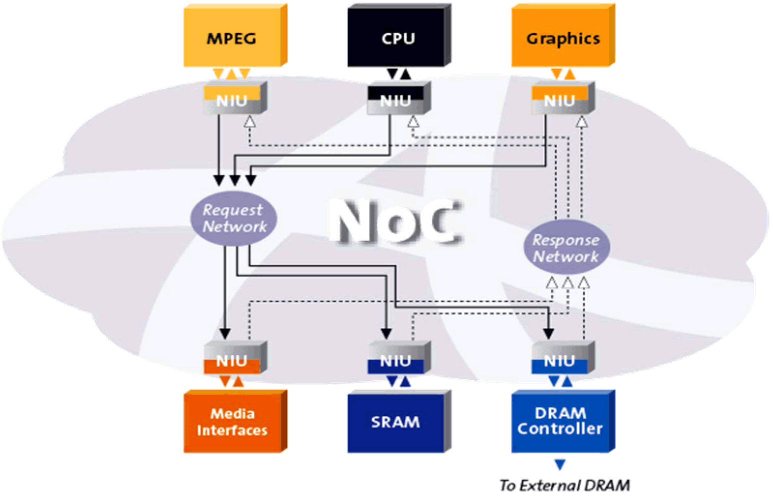
EE 348: Logic Design

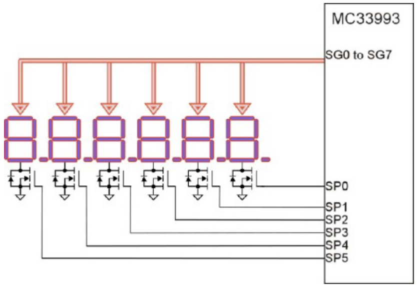
EE 312: Digital Electronics

EE 212: Semiconductor devices

Computers/Computer Architecture in METU EE

EE 444: Computer Networks





EE 447: Microprocessors: I/O device interfacing

Many computing Devices

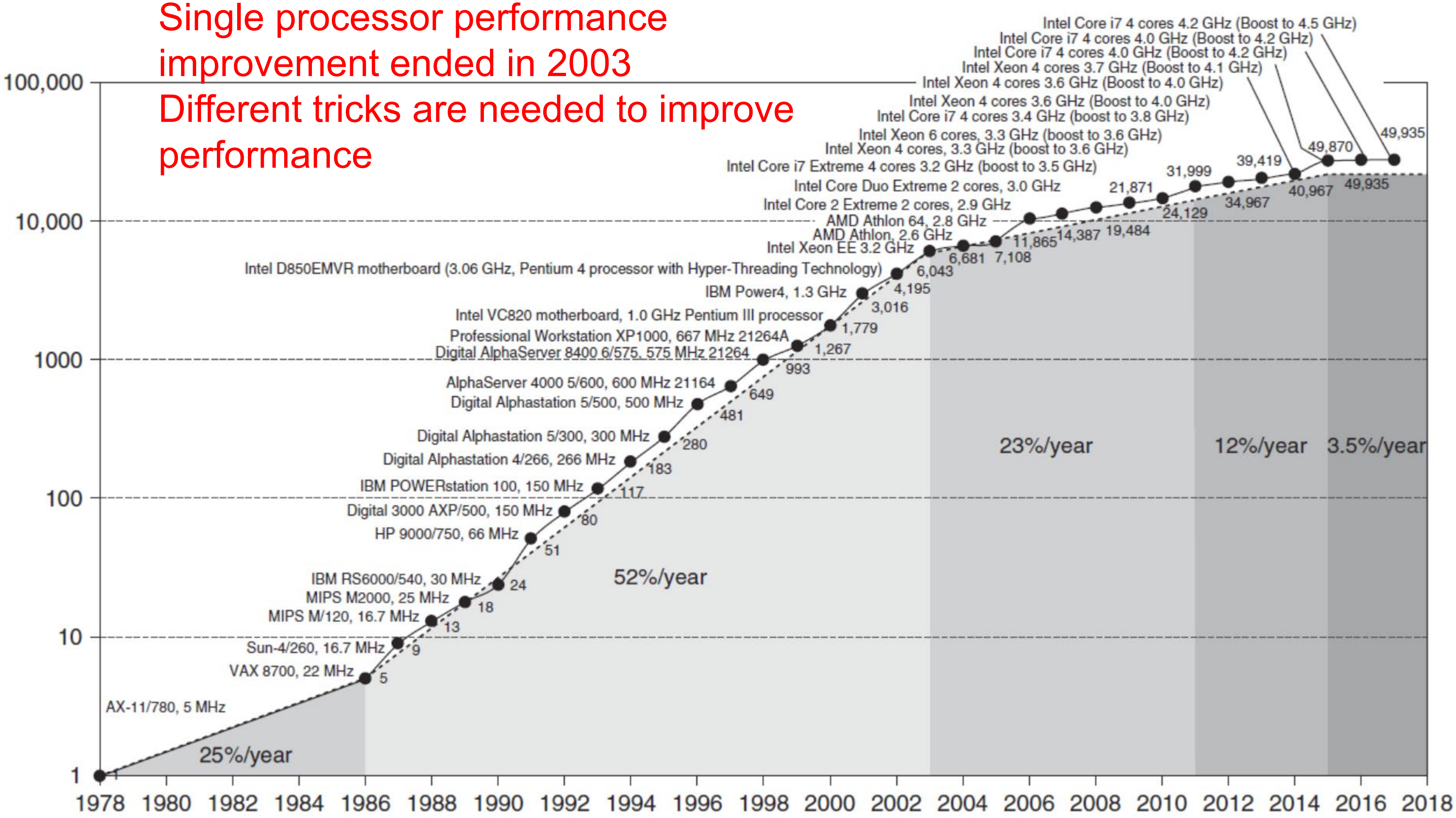
Classes of Computers

- Personal Mobile Device (PMD)
  - e.g. smart phones, tablet computers
  - Emphasis on energy efficiency and real-time
- Desktop Computing
  - Emphasis on price-performance
- Servers
  - Emphasis on availability, scalability, throughput
- Clusters / Warehouse Scale Computers
  - Used for “Software as a Service (SaaS)”
  - Emphasis on availability and price-performance
  - Sub-class: Supercomputers, emphasis: floating-point performance and fast internal networks
- Internet of Things/Embedded Computers
  - Emphasis: price

Trends and Performance

Single processor performance improvement ended in 2003

Different tricks are needed to improve performance



Trends and Performance

- Integrated circuit technology (Moore's Law)
  - Transistor density: 35%/year
  - Die size: 10-20%/year
  - Integration overall: 40-55%/year
- DRAM capacity: 25-40%/year (slowing)
  - 8 Gb (2014), 16 Gb (2019), possibly no 32 Gb
- Flash capacity: 50-60%/year
  - 8-10X cheaper/bit than DRAM
- Magnetic disk capacity: recently slowed to 20%/year
  - Density increases may no longer be possible (TDMR is an exception), maybe increase from 7 to 9 platters
  - 8-10X cheaper/bit then Flash
  - 200-300X cheaper/bit than DRAM

Performance

- Legacy view of computer architecture:
  - Instruction Set Architecture (ISA) design
  - i.e. decisions regarding:
    - registers, memory addressing, addressing modes, instruction operands, available operations, control flow instructions, instruction encoding
- Now computer architecture also focuses on:
  - Specific requirements of the target machine
  - Design to maximize performance within constraints: cost, power, and availability
  - ISA, microarchitecture, hardware

Performance Metrics

- Typical performance metrics:
  - Response time: Time between start and completion of an event
  - Throughput: Total work done in a given time
- Speedup of X relative to Y
  - Execution time$_Y$ / Execution time$_X$
- Execution time
  - Wall clock time: includes all system overheads
  - CPU time: only computation time
- Benchmarks
  - Kernels (e.g. matrix multiply)
  - Toy programs (e.g. sorting)
  - Synthetic benchmarks (e.g. Dhrystone)
  - Benchmark suites (e.g. SPEC06fp, TPC-C)

Introduction to Computer Architecture