Modern Computer Architecture

Department of Electrical Engineering
College of Engineering
Salahaddin university-Erbil
Prepared By: Diary R. SULAIAMAN

MSc Course

First Semester

2020-2021

Modern Computer Architecture

MSc Course

First Semester

CH1

Review and Introduction

OBJECTIVES

□ What is Computer Architecture ?

□ Why do we study Computer Architecture ?

INTRODUCTION

□ Computer architecture, like other architecture, is the art of determining the needs of the user of a ***structure*** and then ***designing*** to meet those needs as effectively as possible within ***economic and technological constraints.***

□ Computer architecture involves ***instruction set architecture design***, ***microarchitecture design***, ***logic design , and implementation*** .

WHAT IS COMPUTER ARCHITECTURE ?

□ *Computer architecture is a set of rules and methods that describe the functionality, organization, and implementation of computer systems.*

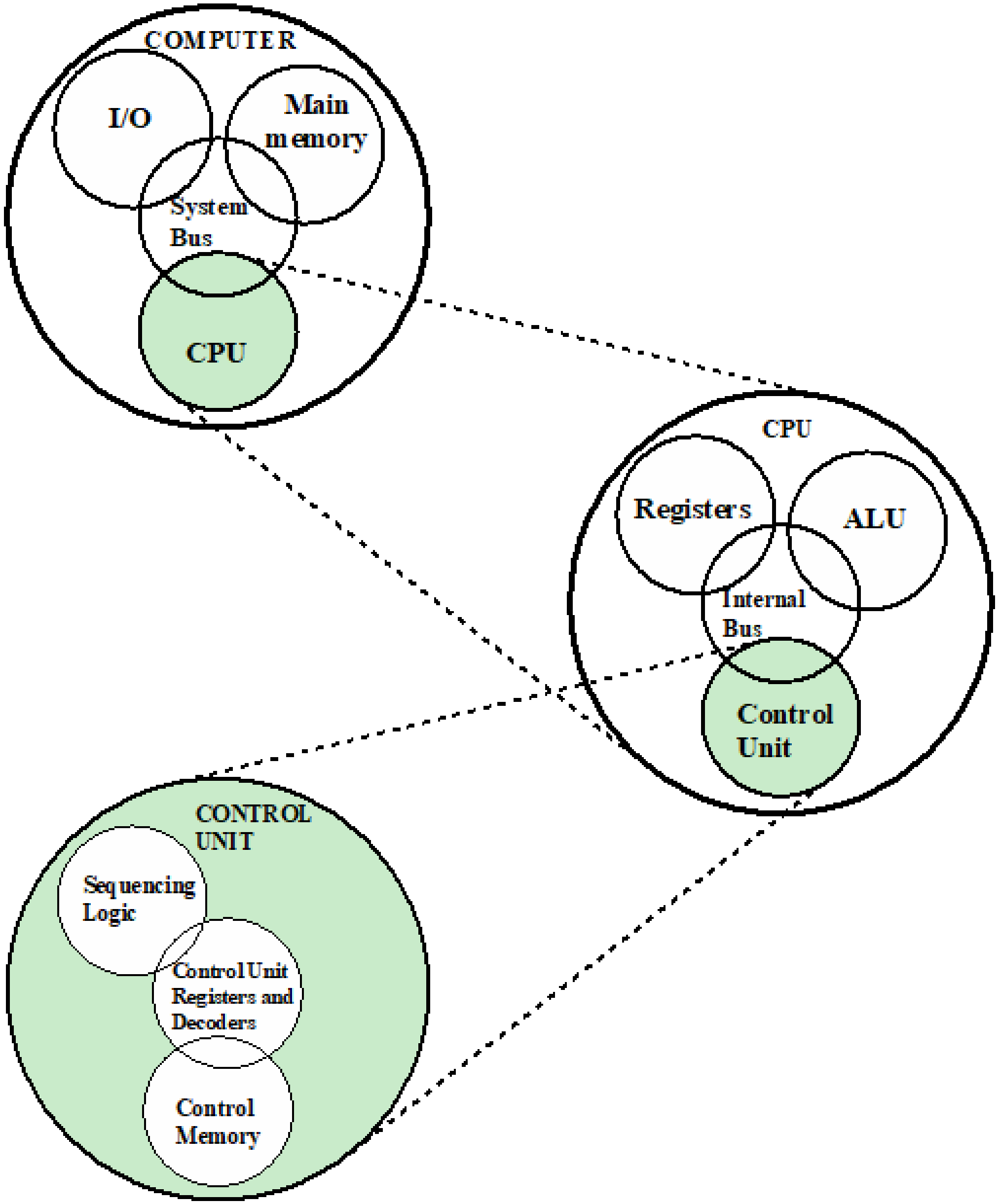
□ *Structure: static arrangement of the parts*

□ *Organization: dynamic interaction of the parts and their control*

□ *Implementation: design of specific building blocks*

□ *Performance: behavioral study of the system*

Structure


COMPUTER
I/O
Main memory
System Bus
CPU
CPU
Registers
ALU
Internal Bus
Control Unit
CONTROL UNIT
Sequencing Logic
Control Unit Registers and Decoders
Control Memory

There are four main structural components of the computer

- CPU – controls the operation of the computer and performs its data processing functions
- Main Memory – stores data
- I/O – moves data between the computer and its external environment
- System Interconnection – some mechanism that provides for communication among CPU, main memory, and I/O

CPU

- Major structural components:

- Control Unit
  - Controls the operation of the CPU and hence the computer
- Arithmetic and Logic Unit (ALU)
  - Performs the computer’s data processing function
- Registers
  - Provide storage internal to the CPU
- CPU Interconnection
  - Some mechanism that provides for communication among the control unit, ALU, and registers

Multicore Structure

- Central processing unit (CPU)
  - Portion of the computer that fetches and executes instructions
  - Consists of an ALU, a control unit, and registers
  - Referred to as a processor in a system with a single processing unit
- Core
  - An individual ***processing unit*** on a processor chip
  - May be equivalent in functionality to a CPU on a single-CPU system
  - Specialized processing units are also referred to as cores
- Processor
  - A physical piece of silicon containing one or more cores
  - Is the computer component that interprets and executes instructions
  - Referred to as a ***multicore processor*** if it contains multiple cores

MACRO-SCALE TOPICS

□ Macro-scale topics includes ***motherboard organization, bus architecture, L1 -L3 cache organization and their policies, SDRAM DDR1 -4 memory and DIMM cards, storage and networking devices and interconnection networks including switch architecture.***

MICRO-SCALE TOPICS

□ Micro-scale topics explicate the internal working of ***microprocessors*** including ***integer and floating point arithmetic, pipelining concepts with dynamic instruction scheduling, branch prediction with speculation, hardware multithreading, exploiting instruction level parallelism, data level parallelism and thread level parallelism.***

OBJECTIVES LARGE-SCALE TOPICS

□ Large-scale topics involve *many cores to many virtual/physical machines and their programming paradigms that power datacenters with adaptive elastic computing infrastructures or cloud computing. The technologies that power datacenters include virtualization of physical resources, virtual machine monitors, load balancing of virtual/physical machines with autonomous as well as scheduled virtual machine migration.*

HISTORY OF COMPUTER ARCHITECTURE

□ The first documented computer architecture was in the correspondence between *Charles Babbage and Ada Lovelace* , describing the analytical engine.

□ When building the computer Z1 in 1936, Konrad Zuse described in two patent applications for his future projects that machine instructions could be stored in the same storage used for data, i.e. the stored-program concept.

□ Two other early and important examples are:John von Neumann's 1945 paper, First Draft of a Report on the EDVAC, which described an organization of logical elements; and Alan Turing's more detailed
Proposed Electronic Calculator for the Automatic Computing Engine , also 1945 and which cited John von Neumann's paper.

THE TERM ARCHITECTURE

□ The term "architecture" in computer literature can be traced to the work of *Lyle R. Johnson, Frederick P. Brooks, Jr., and Mohammad Usman Khan*, all members of the Machine Organization department in *IBM's main research center in 1959*. Johnson had the opportunity to write a proprietary research communication about the Stretch , an IBM developed supercomputer for Los Alamos National Laboratory (at the time known as Los Alamos Scientific Laboratory). To describe the level of detail for discussing the luxuriously embellished computer, he noted that his description of formats,

instruction types, hardware parameters, and speed enhancements were at the level of "system"

EARLIEST COMPUTER ARCHITECTURE

□ The earliest computer architectures were designed on *paper* and then directly built into the *final hardware* form. Later, *computer architecture prototypes* were physically built in the form of a transistor–*transistor logic (TTL) computer*—such as the prototypes of the 6800 and the PA-RISC — tested, and tweaked, before committing to the final hardware form. As of the 1990s, new computer architectures are typically "built", tested, and tweaked—inside some other computer architecture in a computer architecture simulator ; or inside a FPGA as a soft microprocessor ; or both—before committing to the final hardware form.

THE DISCIPLINE OF COMPUTER ARCHITECTURE HAS THREE MAIN SUBCATEGORIES

1. Instruction Set Architecture , or ISA. The ISA defines the machine code that a processor reads and acts upon as well as the word size , memory address modes , processor registers , and data type .

2. Microarchitecture , or computer organization describes how a particular processor will implement the ISA.[14] The size of a computer's CPU cache for instance, is an issue that generally has nothing to do with the ISA.

3. System Design includes all of the other hardware components within a computing system. These include:

a. Data processing other than the CPU, such as direct memory access (DMA)

b. Other issues such as virtualization , multiprocessing, and ...

INSTRUCTION SET ARCHITECTURE

☐ An instruction set architecture (ISA) is the interface between the computer's software and hardware and also can be viewed as the programmer's view of the machine. Computers do not understand high-level programming languages such as Java, C++, or most programming languages used. A processor only understands instructions encoded in some numerical fashion, usually as binary numbers . Software tools, such as *compilers* , translate those high level languages into instructions that the processor can understand.

☐ ISAs vary in quality and completeness. A good ISA compromises between programmer convenience (*how* easy the code is to understand), *size* of the code (how much code is required to do a specific action), *cost* of the computer to interpret the instructions (more complexity means more hardware needed to decode and execute the instructions), and *speed* of the computer (with more complex decoding hardware comes longer decode time). *Memory organization* defines how instructions interact with the memory, and how memory interacts with itself.

COMPUTER ORGANIZATION

□ Computer organization helps *optimize performance* based products. For example, software engineers need to know the *processing power* of processors. They may need to *optimize software* in order to gain the *most performance for the lowest price.* This can require quite detailed analysis of the computer's organization. For example, in a SD card, the designers might need to arrange the card so that the most data can be processed in the fastest possible way.

□ Computer organization also helps *plan the selection of a processor for a particular project.* Multimedia projects may need very rapid *data access*, while virtual machines may need *fast interrupts.* Sometimes certain tasks need *additional components* as well. For example, a computer capable of running a virtual machine needs virtual memory hardware ...

IMPLEMENTATION

Once an instruction set and micro-architecture have been designed, a practical machine must be developed. This design process is called the implementation. Implementation is usually not considered architectural design, but rather hardware design engineering. Implementation can be further broken down into several steps:

- ***Logic implementation*** designs the circuits required at a logic-gate level.
- ***Circuit implementation*** does transistor-level designs of basic elements (e.g., gates, multiplexers, latches) as well as of some larger blocks (ALUs, caches etc.) that may be implemented at the logic-gate level, or even at the physical level if the design calls for it.
- ***Physical implementation*** draws physical circuits. The different circuit components are placed in a chip floorplan or on a board and the wires connecting them are created.
- ***Design validation*** tests the computer as a whole to see if it works in all situations and all timings. Once the design validation process starts, the design at the logic level are tested using logic emulators. However, this is usually too slow to run a realistic test. So, after making corrections based on the first test, prototypes are constructed using Field-Programmable Gate-Arrays (FPGAs). Most hobby projects stop at this stage. The final step is to test prototype integrated circuits, which may require several redesigns.

For CPUs, the entire implementation process is organized differently and is often referred to as CPU design.

PERFORMANCE

- Modern computer performance is often described in *IPC (instructions per cycle).* This measures the efficiency of the architecture at any clock frequency. Since a faster rate can make a faster computer, this is a useful measurement. Older computers had IPC counts as low as 0.1 instructions per cycle. Simple modern processors easily reach near 1.

□ Performance is *affected* by a very wide range of *design choices* – for example, *pipelining a processor* usually makes *latency* worse, but makes *throughput* better. Computers that control machinery usually need *low interrupt latencies*. These computers operate in a real-time environment and fail if an operation is not completed in a specified amount of time. For example, computer-controlled anti-lock brakes must begin braking within a predictable, short time after the brake ......

POWER EFFICIENCY

□ Power efficiency is another important measurement in modern computers. A *higher power* efficiency can often be traded for *lower speed* or *higher cost*. The typical measurement when referring to power consumption in computer architecture is MIPS/W (millions of instructions per second per watt).

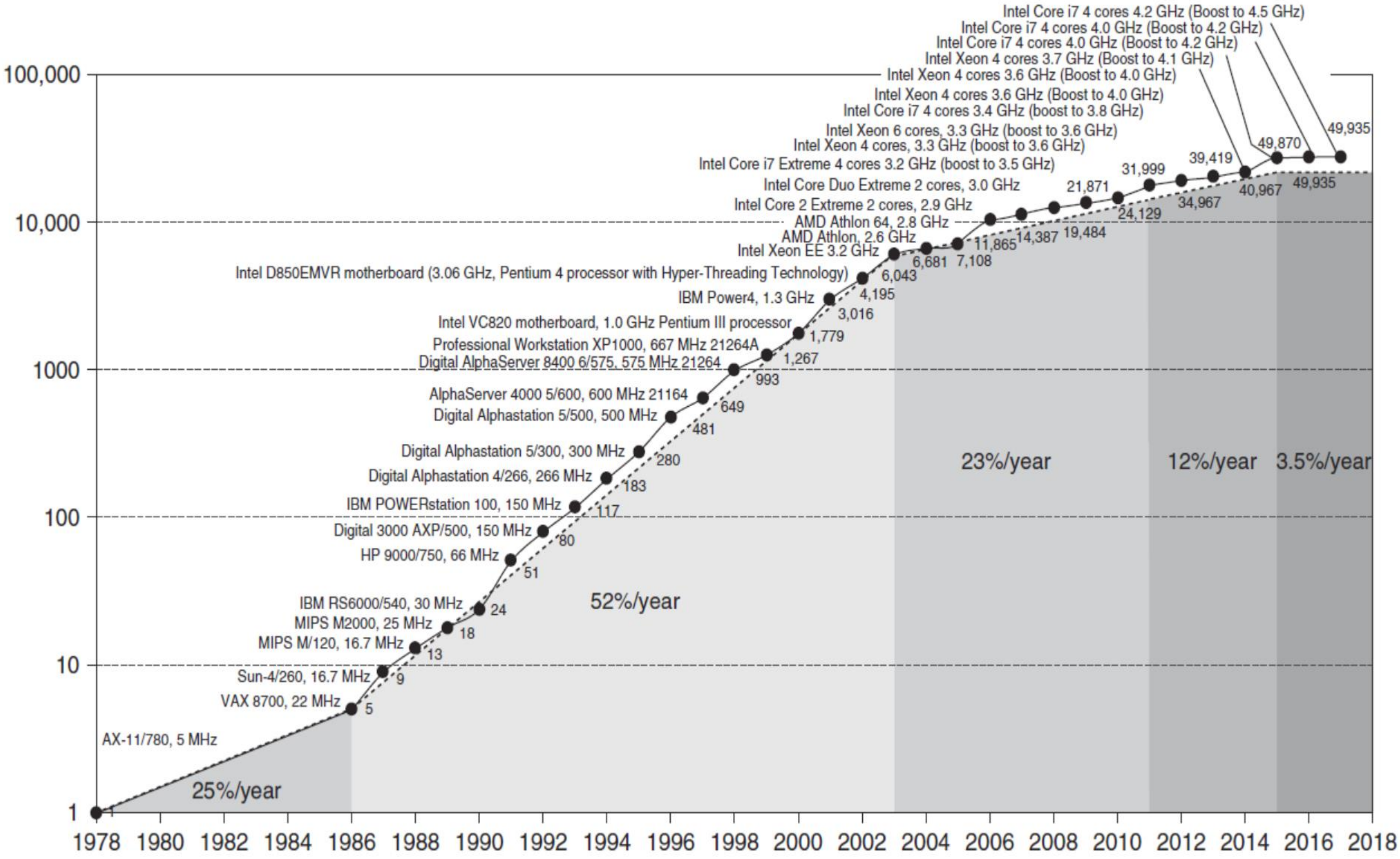
□ Modern circuits have *less power* required *per transistor* as the number of transistors per chip grows. [16] This is because each transistor that is put in a new chip requires its own power supply and requires new pathways to be built to power it. However the number of transistors per chip is starting to increase at a slower rate. Therefore, power efficiency is starting to become as important, if not more important than fitting more and more transistors into a single chip. Recent processor designs have shown this emphasis as they put more focus on power efficiency.....

Performance Improvement: from where?

- Semi-conductor Technology
  - More transistors per chip
  - Faster logic
- Micro Architecture: Machine Organization
  - Deeper pipelines
  - More instructions executed in parallel
- Architecture: ISA, etc.
  - Reduced Instruction Set Computers (RISC > 1985)
  - Multimedia extensions
  - Explicit parallelism
- Compiler technology
- Finding more parallelism in code
  - Greater levels of optimization

Computer
(Micro) Architecture

Performance trends of processors (for 1 core)



Trends in Computer Architecture

- Cannot continue to leverage Instruction-Level parallelism (ILP)
  - Single processor performance improvement ended in 2003

- New models for performance:
  - Data-level parallelism (DLP)
  - Thread-level parallelism (TLP)
  - Request-level parallelism (RLP)

- These require explicit restructuring of the application

Trends in Technology

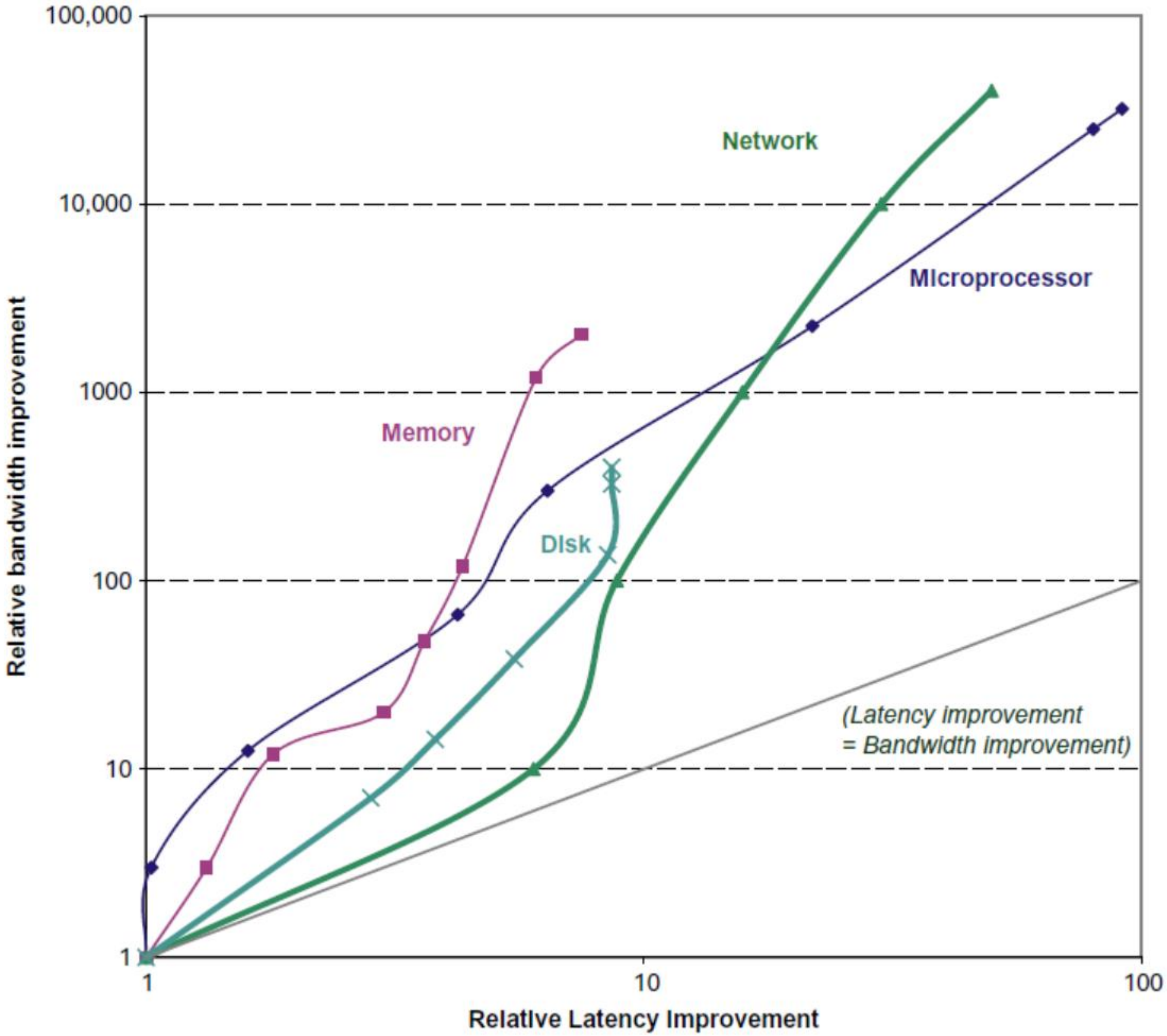
- Integrated circuit technology (Moore's Law)
  - Transistor density: 35%/year
  - Die size: 10-20%/year
  - Integration overall: 40-55%/year
- DRAM capacity: 25-40%/year (slowing)
  - 8 Gb (2014), 16 Gb (2019), possibly no 32 Gb
- Flash capacity: 50-60%/year
  - 8-10X cheaper/bit than DRAM
- Magnetic disk capacity: recently slowed to 5%/year
  - Density increases may no longer be possible, maybe increase from 7 to 9 platters
  - 8-10X cheaper/bit than Flash
  - 200-300X cheaper/bit than DRAM

Bandwith vs Latency

- Bandwidth or throughput
  - Total work done in a given time
  - 32,000-40,000X improvement for processors
  - 300-1200X improvement for memory and disks
- Latency or response time
  - Time between start and completion of an event
  - 50-90X improvement for processors
  - 6-8X improvement for memory and disks

(for more on above numbers see next slide)

Bandwith vs Latency



Log-log plot of bandwidth and latency milestones

Transistors and Wires

- Feature size
  - Minimum size of transistor or wire in x or y dimension
  - 10 microns in 1971 to .011 microns in 2017
  - Transistor performance scales linearly
    - Wire delay does not improve with feature size!
  - Integration density scales quadratically

Power / Energy

- Problem: Get power in, get power out

- Thermal Design Power (TDP)
  - Characterizes sustained power consumption
  - Used as target for power supply and cooling system
  - Lower than peak power (1.5X), higher than average power consumption

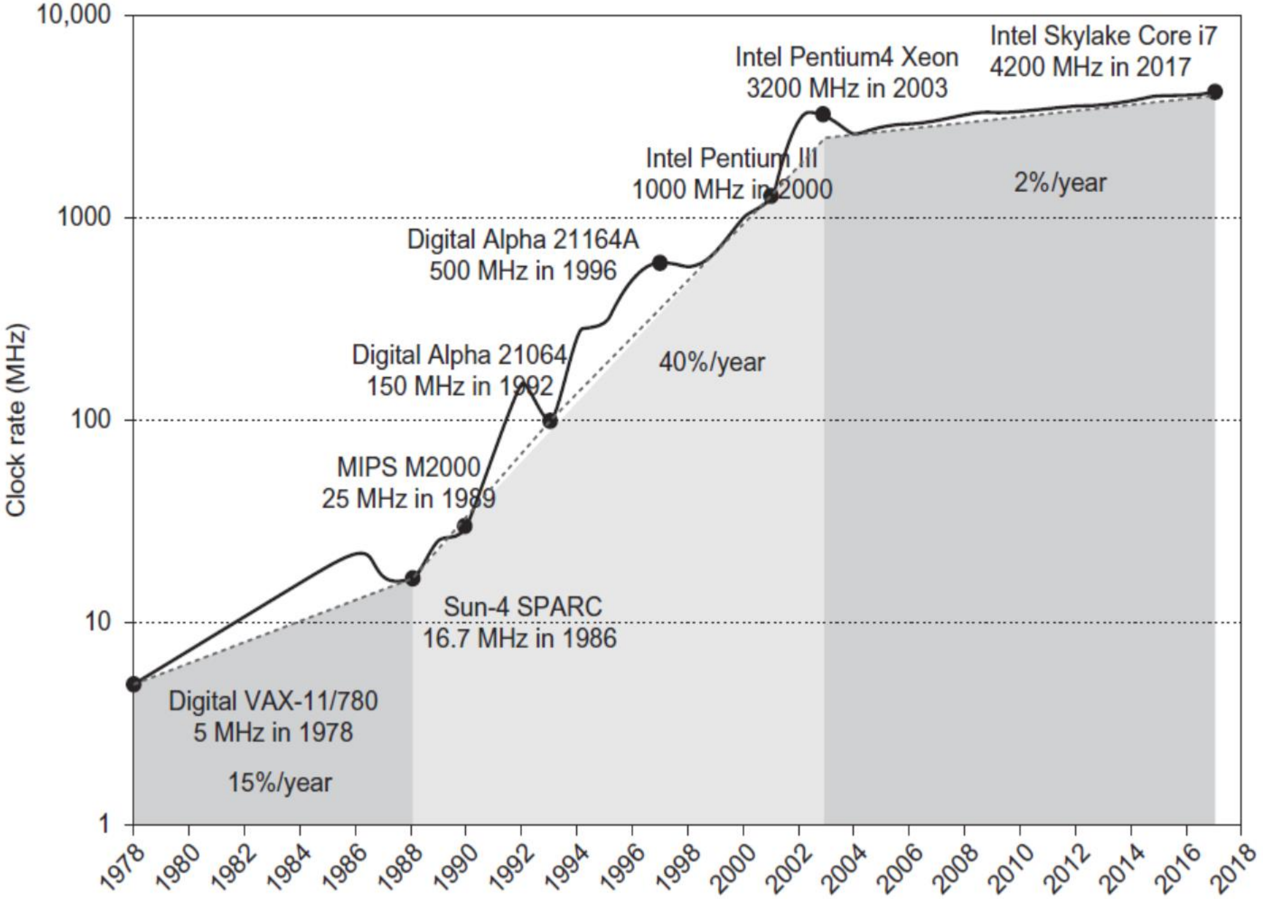
- Clock rate can be reduced dynamically to limit power consumption

- Energy per task is often a better measurement

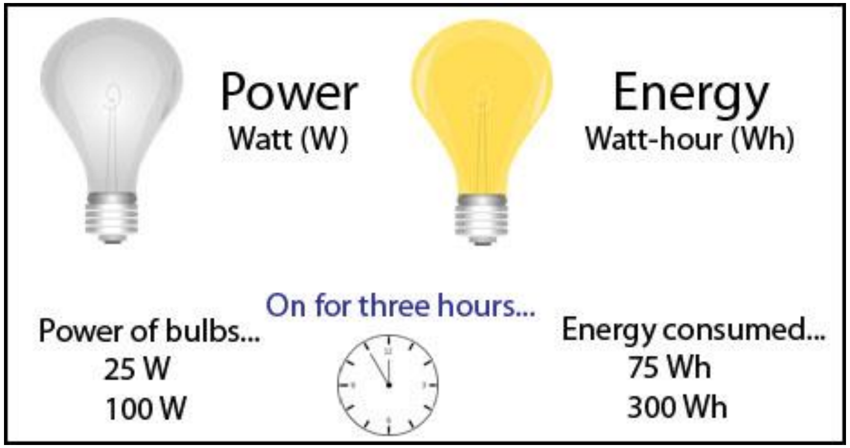
Frequency and Power trends (of processors)

- Intel 80386 consumed ~ 2 W
- 3.3 GHz Intel Core i7 consumes 130 W
- Heat must be dissipated from 1.5 x 1.5 cm chip

- This is about the limit of what can be cooled by air



Power / Energy



Power vs Energy

- Energy E = P*t

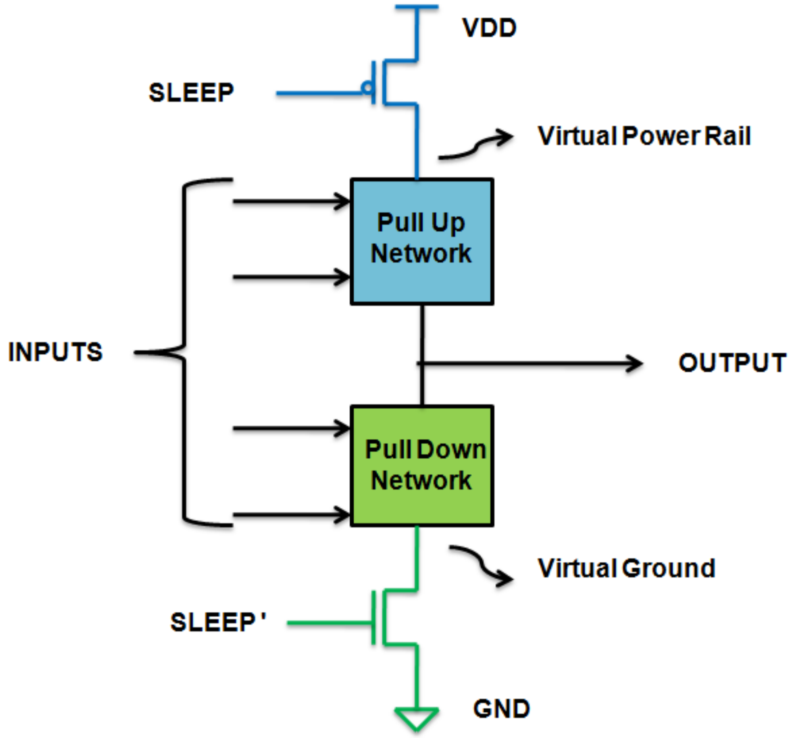
  (power P, execution time t)
- $P = P_{static} + P_{dynamic}$

- static: leakage ~ chip area
- dynamic: switching (0->1 or 1->0
  - $P_{dynamic} = ½\, α\, f\, C\, V_{dd}^2$
  - $E_{dynamic} = ½\, α\, C\, V_{dd}^2$
- Reducing clock rate reduces power, not energy

Static Power Consumption

- 25-50% of total power
- Current$_{static}$ x Voltage
- Scales with number of transistor

- To reduce: power gating:

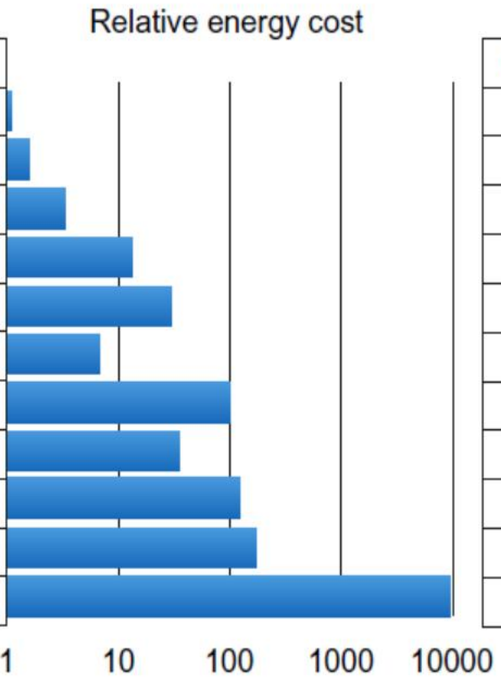


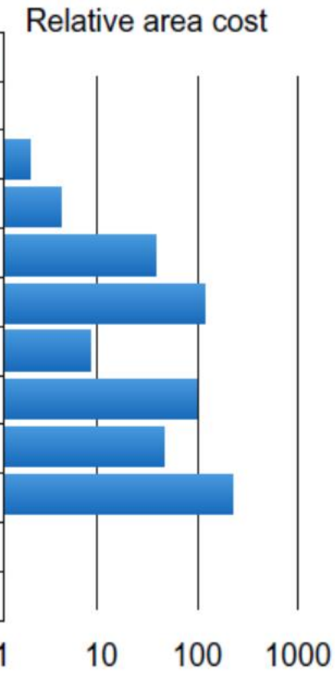
Where is the energy / power going?

Dynamic power ~ Switching energy

Static power ~ Area

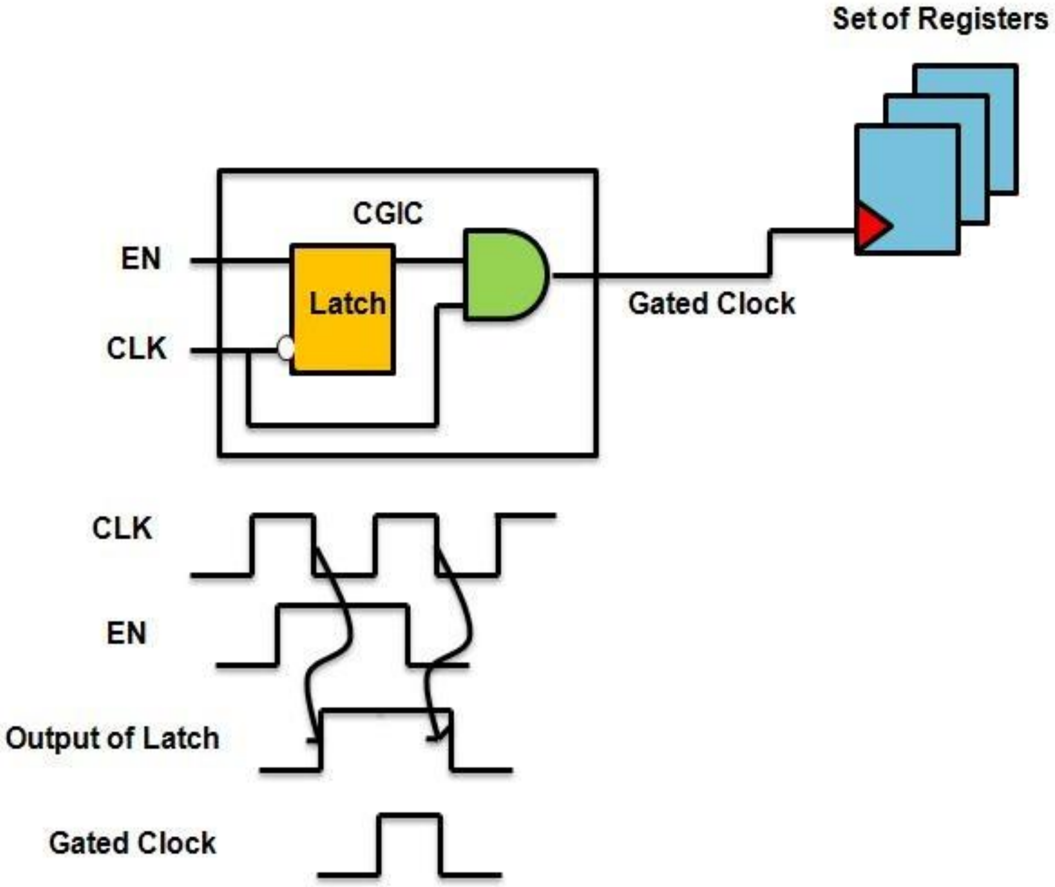
Operation:	Energy (pJ)	Area (µm^2)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FB Add	0.4	1360
32b FB Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FB Mult	1.1	1640
32b FB Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A





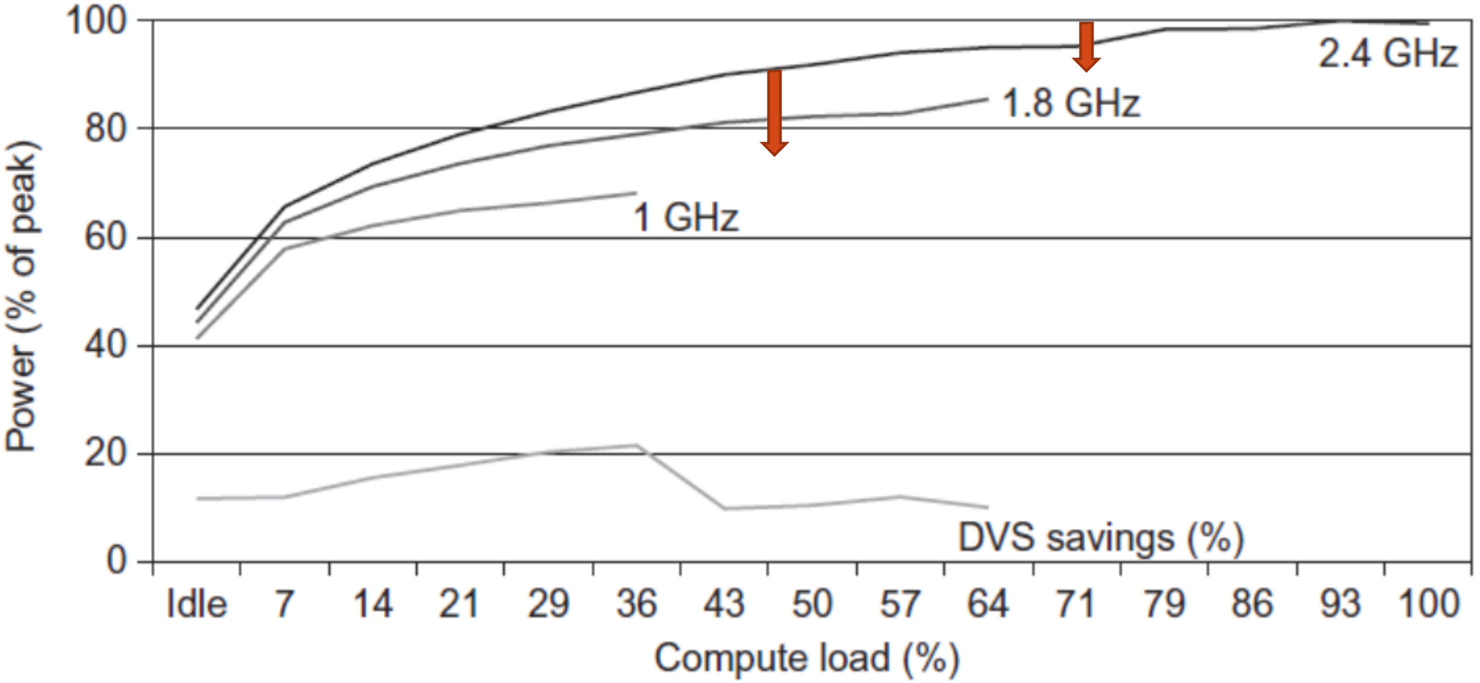
Dynamic power reduction

- Reduce switching (dynamic) energy by **clock gating**:



Dynamic power reduction

- Dynamic Voltage-Frequency Scaling (DVFS)



Modern Computer Architecture

MSc Course

First Semester

2020-2021

END of CH1

Review and Introduction