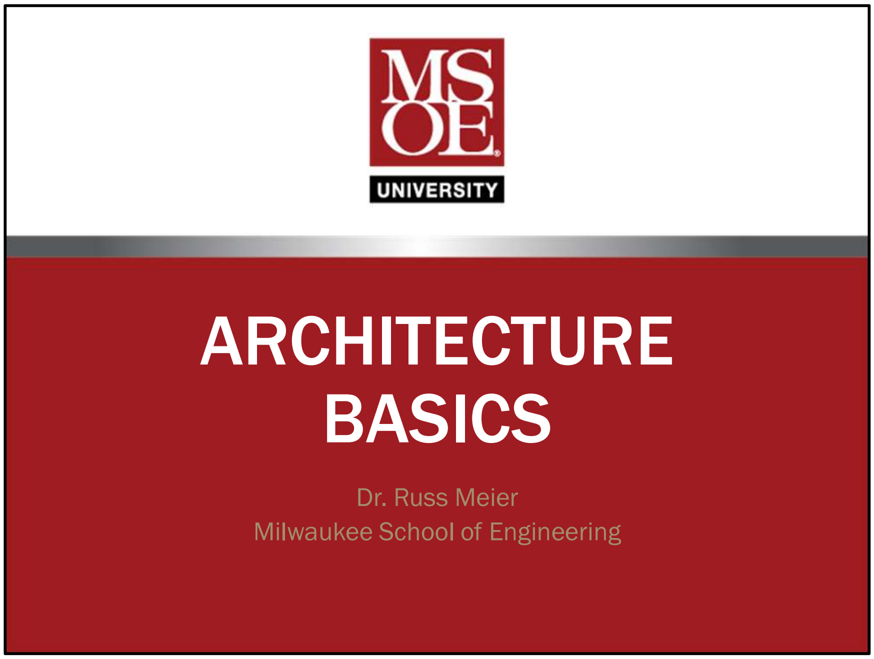
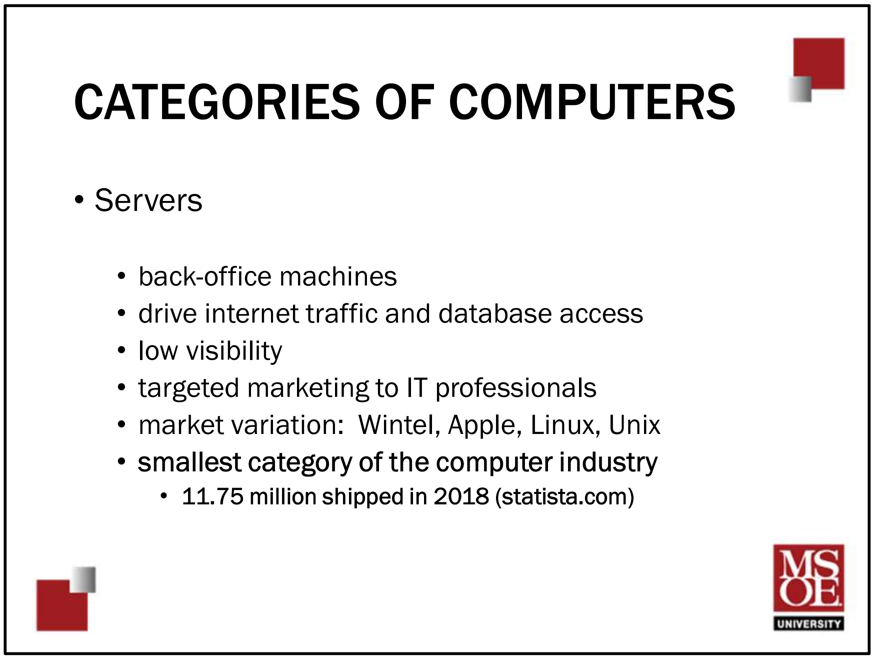


ARCHITECTURE BASICS

Dr. Russ Meier
Milwaukee School of Engineering

Welcome to Computer Architecture. I am excited to show you this fantastic field of hardware engineering.

- Architecture is the study of organizing components together to form useful things.
- Computer architects design and build computer hardware and computer systems.
- As hardware engineers, computer architects build the machines that software runs on – without the computer architects, there would be no computers.



CATEGORIES OF COMPUTERS

- Servers
  - back-office machines
  - drive internet traffic and database access
  - low visibility
  - targeted marketing to IT professionals
  - market variation: Wintel, Apple, Linux, Unix
  - smallest category of the computer industry
    - 11.75 million shipped in 2018 (statista.com)

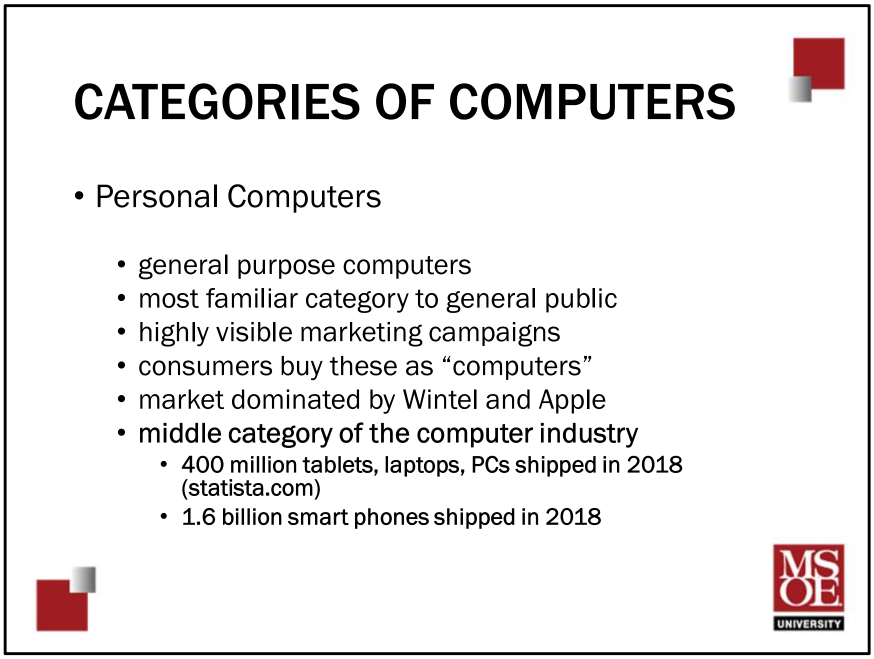
The modern computer industry can be categorized in various ways. One common categorization groups the industry based on three primary service sectors:

- Remote information and data services - **servers**
- Desktop and handheld general-purpose computers – **personal computers**
- Computers embedded in other smart devices – **embedded computers**

The server sector of the industry is the smallest sales segment. Data is still being collected for 2019 and thus the last full year of data is 2018. One respected data source for statistics reports that near twelve million servers shipped in 2018 (statista.com).

Servers are typically rack-mounted machines collected in clusters and located in server racks maintained by organizations at their primary site of business or at remote sites. Servers are typically headless machines – meaning they are not connected directly to monitors or keyboards. Instead they are accessed by system administrators through the internet using remote login applications like secure shell (ssh).

It may seem surprising that this segment is the smallest sales segment given the huge size of the internet and the prevalence of internet browsing, e-commerce, and streaming. It is important to remember, however, that these machines are **architecturally designed** to handle hundreds of clients simultaneously connected through the internet. Because they handle multi-users at the same time, the number required is relatively small.



CATEGORIES OF COMPUTERS

- Personal Computers
  - general purpose computers
  - most familiar category to general public
  - highly visible marketing campaigns
  - consumers buy these as “computers”
  - market dominated by Wintel and Apple
  - middle category of the computer industry
    - 400 million tablets, laptops, PCs shipped in 2018 (statista.com)
    - 1.6 billion smart phones shipped in 2018

Personal computers are general-purpose machines that are what most people typically call “a computer.”

- Historically, personal computers go on sale in the 1970s as the integrated circuit industry microminiaturizes architectural components into chip form.
- Controversy surrounds what is labeled as the “first personal computer.” Historians look at price, footprint, implementation technology, marketing terms, and general-purpose programmability.
- Traditional computer companies had introduced some small-sized calculators, terminals, and indeed computers that could be argued as historical firsts.
- Two respected computer history museums (the Computer History Museum and the American Computer Museum) consider the Kenbak-1 as the first true personal computer. It was released in early 1971. This machine **did not have a microprocessor;** instead, it implemented the central processing unit using logic gate chips and functional logic chips (multiplexers, adders, etc.).
- Others argue that the true personal computer industry didn’t develop until the first microprocessor-based designs.
- Some of the earliest example 1970s microprocessor-based personal computers are kit-form machines assembled by the consumer: the Altair 8800, the KIM-1, and the

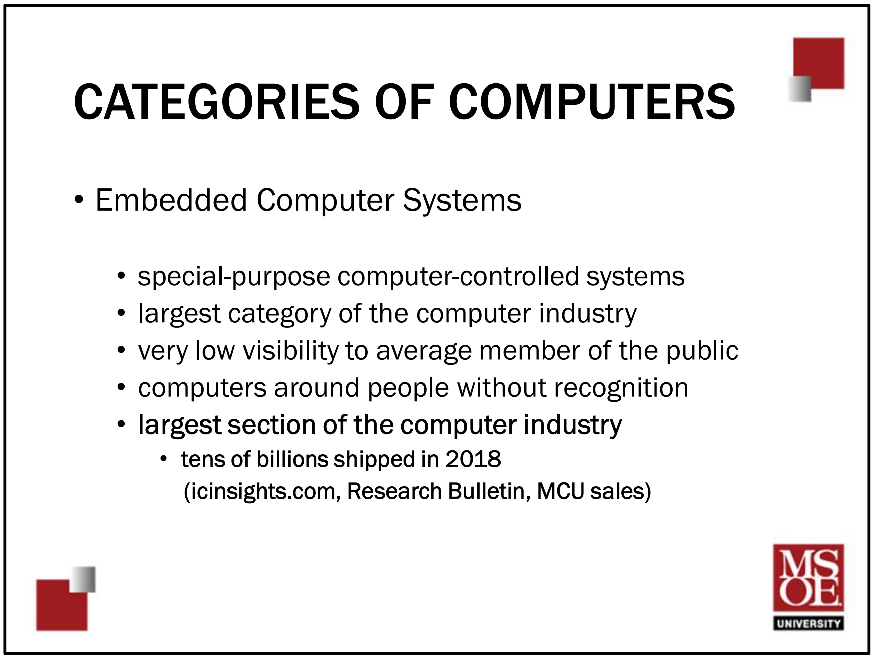
Apple I are examples. The MSOE library has an Altair 8800 on display along the wall of windows on the second floor.

- In 1977, three affordable pre-assembled microprocessor-based machines were released: the Apple II, the TRS-80, and the Commodore PET. These machines brought computing into more homes and businesses and truly helped spawn the modern personal computer era.
- As the 1970s ended and the 1980s began, some of the most loved home computers arrive: the TI99/4A, the Commodore 64, the Atari 800, the IBM PC, the BBC Micro, the Apple IIE and the Apple IIC are examples.
- The personal computers of the 1970s and 1980s were mostly accessed and programmed through text-based monochrome monitors, although some users could afford color monitors. The operating systems were controlled through **command-line** instructions. Programs were mostly entered in the **BASIC** programming language.
- By the mid-1980s, the graphical user interface arrived as the Apple Macintosh, the Commodore Amiga, and the first editions of Microsoft Windows made computers easier to use. Point-and-click control and what-you-see-is-what-you-get (WYSIWYG – an acronym pronounced “whas-eee-whig”) editing became common and the personal computer industry exploded because computers were now able to be used by everyone with relatively little training.
- In the 21st century, personal computers have become highly mobile. Laptops, tablets, mobile phones, and smart watches have become primary computing platforms. When the smart phone counts are included, the personal computer industry shipped around 2 billion units in 2018. This makes it the second largest computer industry sector.

The history of the personal computer industry is fascinating. It contains many great stories of innovation, hardship, competition, lawsuits, and rags-to-riches biographies. Here are some to consider reading about:

- Federico Faggin: Designer of Intel 4004, Intel 8080, and Zilog Z80. Co-founder and CEO of Zilog.
- Edward Roberts: Designer of Altair 8800 kit computer. Co-founder of MITS.
- Steve Jobs, Steve Wozniak, Ronald Wayne: Co-founders of Apple Computer.
- Bill Gates, Paul Allen: Co-founders of Microsoft.
- John Hennessy: Co-founder of MIPS
- David Patterson: Leader of the Berkeley RISC Project which strongly influences the SPARC machines commercialized by Sun Microsystems.
- Sophie Wilson: Co-designer of the BBC Micro, designer of the ARM Instruction Set Architecture
- Steve Furber: Lead designer of the ARM microprocessor – implementing the ARM ISA

Of course, this is by no means an exhaustive list. There are many pioneers in computer science and engineering that helped create the theories of computation, the engineering techniques, and the engineering tools that have all led to the modern computer era.

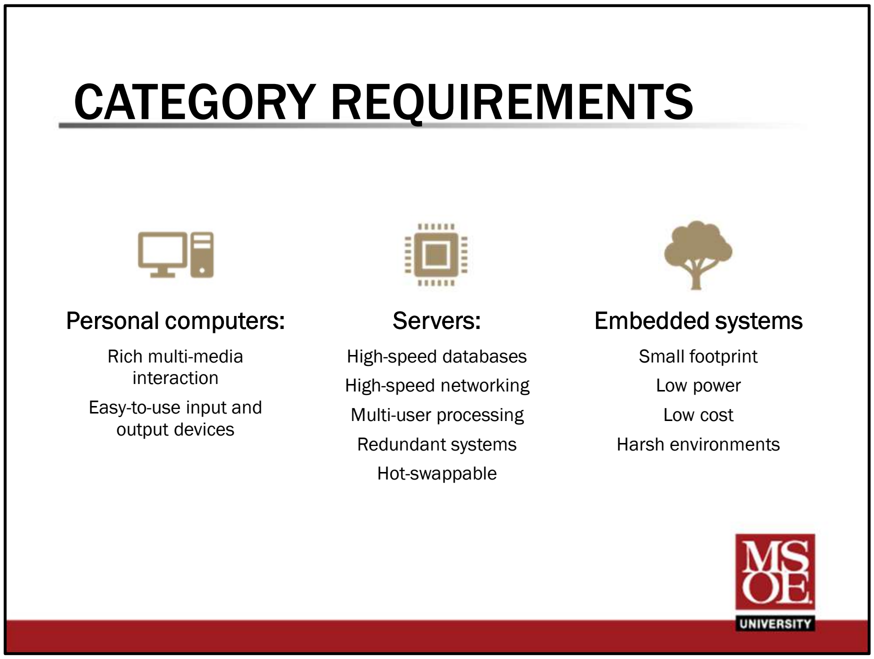


CATEGORIES OF COMPUTERS

- Embedded Computer Systems
  - special-purpose computer-controlled systems
  - largest category of the computer industry
  - very low visibility to average member of the public
  - computers around people without recognition
  - largest section of the computer industry
    - tens of billions shipped in 2018
      (icinsights.com, Research Bulletin, MCU sales)

Embedded systems are the largest category of the computer industry.

- This segment contains all the smart devices that run firmware on a computer embedded in some other product.
- These devices surround us but are not typically thought of as a computer by the general public.
- As the largest segment of the computer market this is the segment in which most computer engineers will have very rewarding careers.
- Products fall in large subcategories: transportation systems, appliances, audiovisual systems, entertainment systems, heating/ventilation/air-conditioning, biomedical, agricultural, and power and energy systems are examples.
- At one time, I considered smart phones as embedded systems. Today, I consider them more as personal computers because of how people use them. Thus, I have moved them to the previous category.



CATEGORY REQUIREMENTS

Personal computers:

Rich multi-media interaction

Easy-to-use input and output devices

Servers:

High-speed databases

High-speed networking

Multi-user processing

Redundant systems

Hot-swappable

Embedded systems

Small footprint

Low power

Low cost

Harsh environments

Each of the computer industry categories has different requirements that dictate how architects design and build systems.

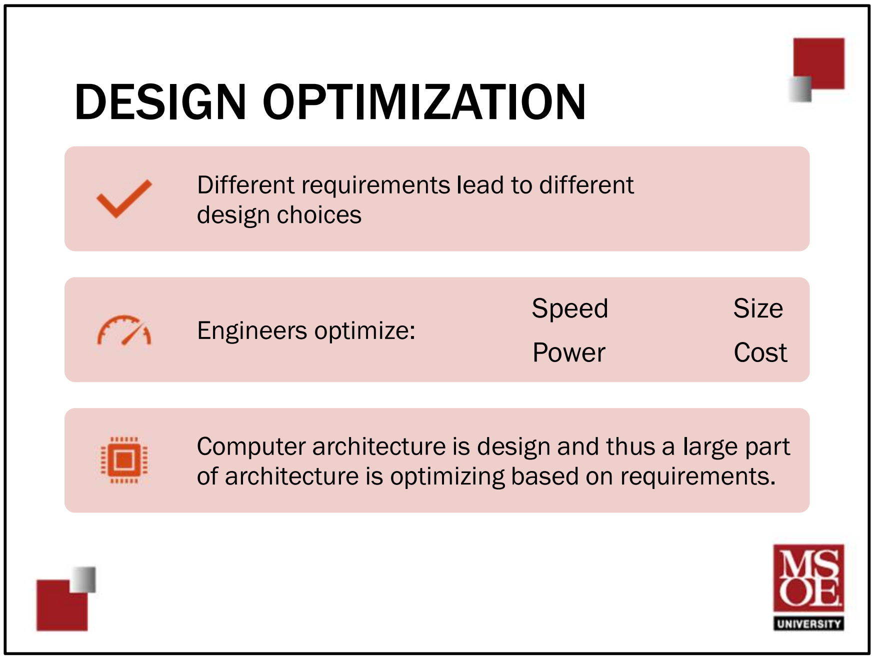
- Personal computer users generally maintain computer-literacy but most are not engineers or computer science experts. Today, these users demand mobile devices with easy to use interfaces. The industry is responding with touchscreens, audio-responsive virtual assistants (https://en.wikipedia.org/wiki/Virtual_assistant), and wireless I/O connected through personal area networks like Bluetooth (IEEE Standard 805.15.1: https://en.wikipedia.org/wiki/Bluetooth).

- Servers provide data retrieval to tens or hundreds of simultaneous users. This requires high-speed networking, multicore CPUs, redundant arrays of disks, redundant power-supplies, redundant memories, and hot-swappable components. Redundancy improves fault-tolerance so that an error in a component doesn’t move the server from up-time to down-time. Hot-swappable components allow system administrators to replace faulty components while the system remains operating through the redundant components.

- Embedded systems add weight and space to the systems they are placed within.

The embedded system footprint must be reduced to help constrain the size of the product. Many embedded systems also experience temperature, humidity, vibration, dirt, ionizing radiation, and other environmental variations through their lifetimes.

- The wide range of smart embedded systems means a wide range of power and cost points. In general, though, most are low-power and low-cost systems.



DESIGN OPTIMIZATION

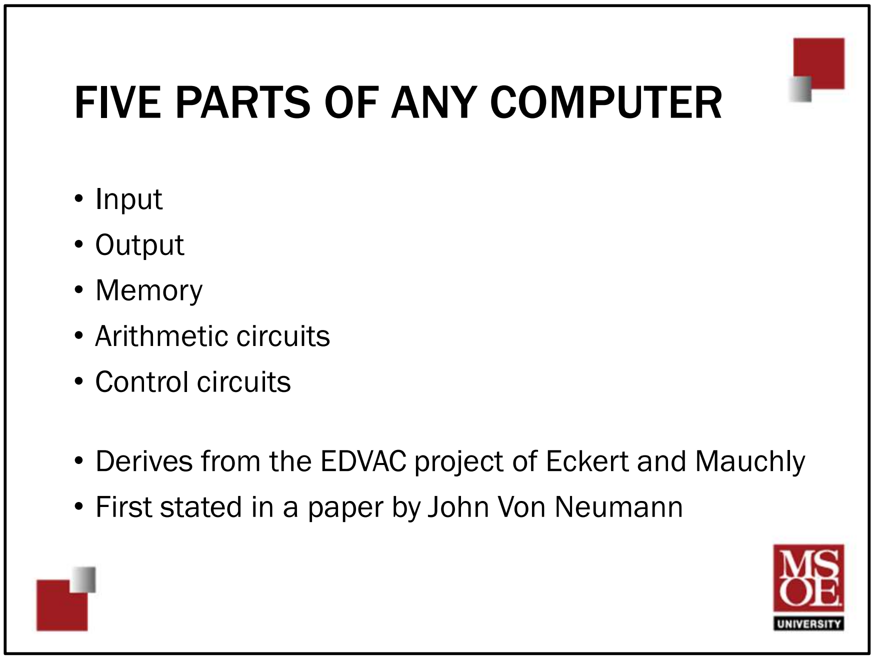
✓ Different requirements lead to different design choices

Engineers optimize: Speed Size Power Cost

Computer architecture is design and thus a large part of architecture is optimizing based on requirements.

All engineers optimize their systems.

- In our field, we tend to optimize speed, size, power, and cost.
- The design choices we make to meet system specifications impact all four of these areas.
- Thought is given to each area as we make choices.
- Sometimes we can strike a good balance optimizing all four areas.
- Other systems will demand sacrifice in some areas to achieve goals in others.



FIVE PARTS OF ANY COMPUTER

- Input
- Output
- Memory
- Arithmetic circuits
- Control circuits

- Derives from the EDVAC project of Eckert and Mauchly
- First stated in a paper by John Von Neumann

Computer architecture is an old field. Computers move through four primary eras:

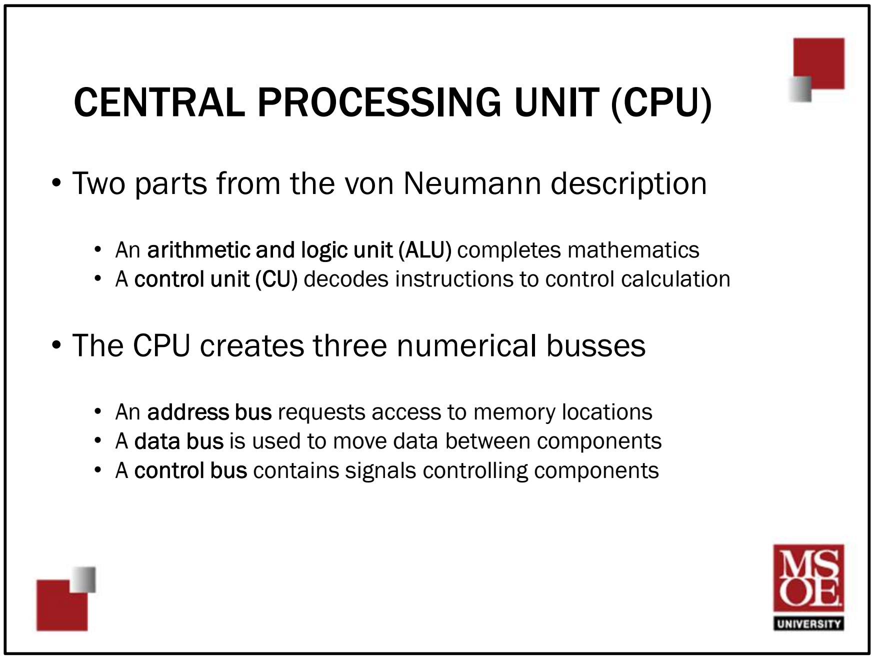
- **Mechanical computers** using levers, cams, and gears are the first machines built to aid calculation. Examples exist from before the common era (BCE) through the centuries of the modern era (CE). These fascinating mechanical machines include clocks, adders, cash registers, looms, and even machine to work calculus and differential equations.
- **Electro-mechanical computers** arrive in the early decades of the twentieth century. These machines used relays – mechanical switches controlled by electromagnets – to create logic gates. They are not purely electrical machines because of these mechanical switches. Important examples include the Z2 (Konrad Zuse, Germany, 1940), and the Harvard Mark 1 (Howard Aiken, USA, 1944).
- **Electronic computers without stored programs** build logic gates from electronic parts. The vacuum tube was invented in 1904 and vacuum-tube based switching circuits began slowly began to replace electro-mechanical computers through the 1940s and 1950s. Some earlier models included the Atanasoff-Berry Computer (Atanassoff/Berry, USA, 1942, non-programmable), the Colossus (Flowers, England, 1943, programmable), and the ENIAC (Mauchly/Eckert, USA, 1945). ENIAC is regarded as the first fully-programmable general-purpose digital computer.

  - ENIAC used 20,000 vacuum tube – and lots more stuff – and cost around $500,000 in 1945. This would be about $6 million in 2020 dollars.
  - ENIAC was programmed by six women that rewired the machine for each new task. These women are often forgotten in computer history but played a very significant role.
  - ENIAC was designed to calculate artillery tables during World War II and was also used to do simulations during the development of the atomic bomb.

- **Electronic stored program computers** emerge at the end of the 1940s and spawn modern computer architecture.
  - The stored program concept states that a computer program should be stored in computer memory as binary voltage bits rather than through physical wired connections to power and ground.
  - The stored program concept freed the computer from constant rewiring. This concept encourages generality in machine design and improved fault-tolerance as it removed the rewiring task and the difficulty of identifying hardware bugs introduced by rewiring. Of course, software bugs still occur but are generally easier to find.
  - The stored program concept is first stated by Alan Turing in 1936 when describing what is now called the Universal Turing Machine. He noted that actions of the machine could be provided on the same "tape" that served as data memory.
  - John von Neumann collaborated with J. Prespert Eckert and John Mauchly as a consultant on the design of EDVAC. During this work, they planned a stored-program computer.
  - **John von Neumann wrote a seminal report that described the architecture of EDVAC** while crossing the country by train to Los Alamos in 1945. This report entitled "First Draft of a Report on the EDVAC" is **considered by historians as the first piece of written work that fully describes the architecture of an electronic digital stored-program computer.**
  - The report is controversial because the person that distributed it only put John von Neumann's name on it. Without the credit they deserved on the report, Eckert and Mauchly lost some of the credit for the project they were leading as the report was cited and passed around the world.
  - The "**five parts of any computer**" have stood the test of time. All modern computers still map to the model documented in the EDVAC report.
  - The report inspired computer designers around the world. The first stored-program computer was the Manchester Baby (Williams/Kilburn/Tootill, England, 1948). This project leads to the Manchester Mark 1 (Williams/Kilburn/Tootill/Edwards/Thomas, England, 1949) which is then

commercialized as the Ferranti Ltd. Mark 1, the first commercially available electronic digital stored-program computer.

- EDVAC became operational in 1949. Even though it inspired the stored-program computer through its architectural description, it wasn’t the first to operate.



CENTRAL PROCESSING UNIT (CPU)

- Two parts from the von Neumann description
  - An **arithmetic and logic unit** (ALU) completes mathematics
  - A **control unit** (CU) decodes instructions to control calculation
- The CPU creates three numerical busses
  - An **address bus** requests access to memory locations
  - A **data bus** is used to move data between components
  - A **control bus** contains signals controlling components

In Section 2 of the EDVAC report, von Neumann describes a **central arithmetic part** and a **central control part.** It is fascinating to read the simplicity of the sections of the report that describe these circuits. In very succinct sentences, he describes how optimization will be required when choosing the basic set of arithmetic operations in the central arithmetic part and how the central control part should be all-purpose – what we today call **general-purpose** – to allow the machine to be the most useful. Interestingly, he never joins the two parts together into a set of parts called a **central processing unit (CPU)**. This grouping occurs later as the computer industry matures in the 1950s and appears regularly in printed English language dictionaries by the early 1960s. This slide shows the standard terms that develop during the decades following the EDVAC report and now in standard use.

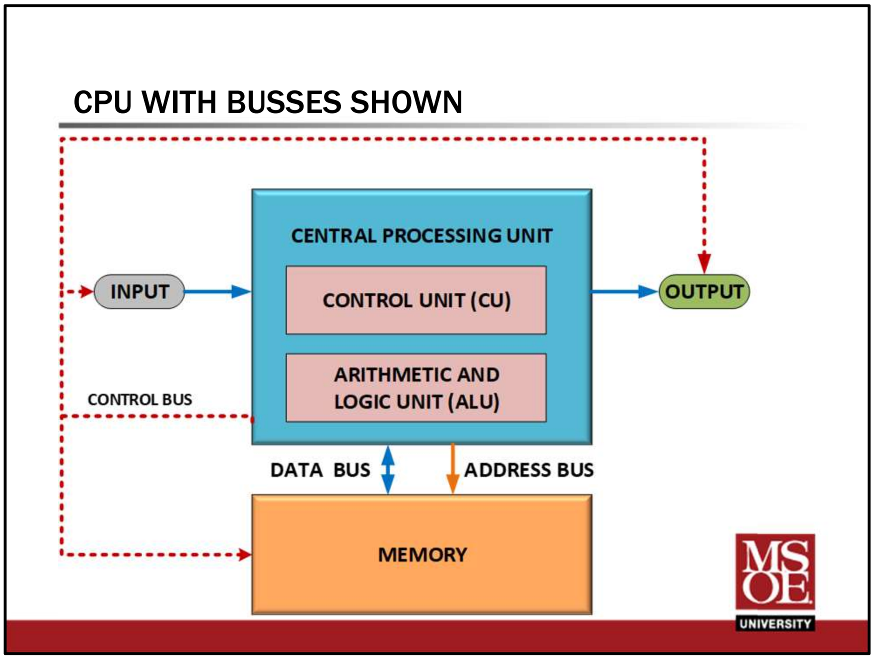
- The **arithmetic and logic unit** is the modern version of von Neumann's central arithmetic part. The ALU completes arithmetic and bitwise-logic operations on two n-bit wide numbers.
- The **control unit** is the modern version of von Neumann's central control part. This circuit is responsible for advancing through a stored program, decoding binary instructions, executing the operation by controlling the ALU, and storing the calculated value back to memory. These four responsibilities define unique time

unit required to complete any instruction.

  - **Fetch**: advance through the program memory to the next instruction
  - **Decode**: interpret the instruction binary number to determine the operation and required data
  - **Execute**: control the ALU through control signals so that the correct arithmetic or logic operation completes
  - **Memory**: store the calculated result back into memory

In order to implement control and execution of the required operation, the CPU circuit generates a minimum of three numbers as the instruction executes. Some CPU architectures, as we will see, will generate more numbers. Recall from prerequisite courses that a **bus** contains an n-bit wide number. Sometimes engineers use alternate words for a bus, such as **cable** and **vector**. In computer architecture, those terms are rarely used.

- The **address bus** is a number representing the location in memory that the CPU wants to access.
- The **data bus** is a number representing data flowing to the CPU from the memory or data flowing from the CPU to the memory.
- The **control bus** is a number holding the control voltage bits attached to the control inputs of the ALU, memory, input, and output devices.

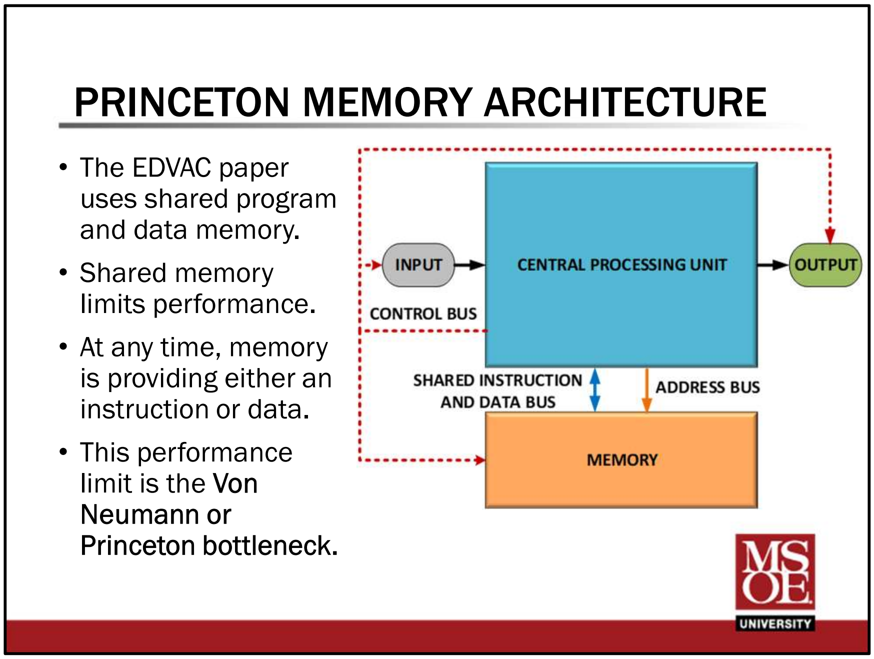


CPU WITH BUSSES SHOWN

This diagram shows the **central processing unit (CPU)** communicating with the other components of the computer using the address, data, and control busses.

- The **control bus** is shown in red as a dashed line. Directional arrows have been added to show these are **unidirectional** bus voltages created by the CPU circuitry.
- The **address bus** is shown as a unidirectional signal created by the CPU to provide the desired location to the memory circuit.
- The **data bus** is shown as a **bidirectional** bus moving data numbers between the CPU and memory.

This diagram represents the machine described by von Neumann in the EDVAC report using modern terms. This machine is known as the **Princeton** architecture because von Neumann was working at the Princeton Institute for Advanced Studies**.**

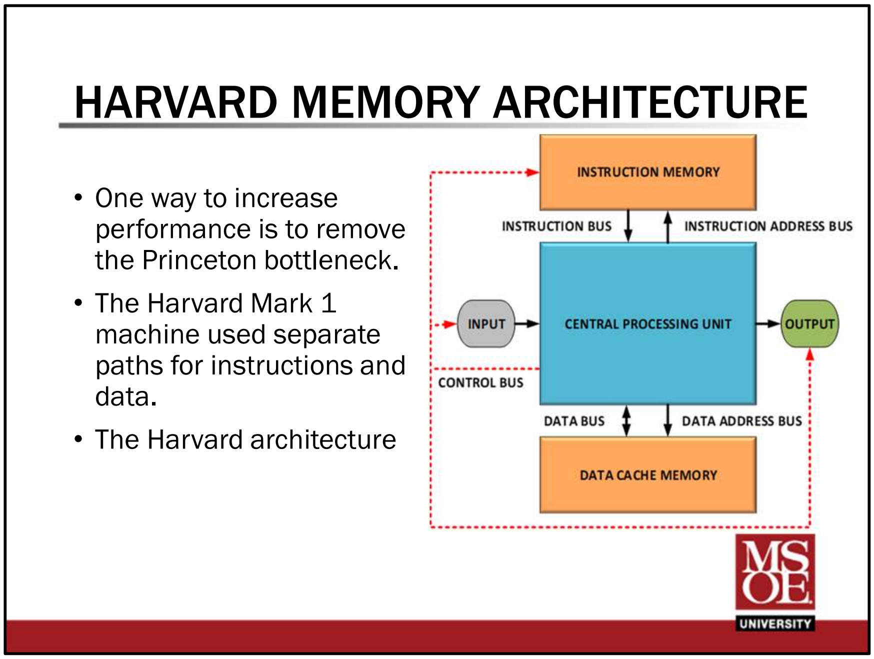


PRINCETON MEMORY ARCHITECTURE

- The EDVAC paper uses shared program and data memory.
- Shared memory limits performance.
- At any time, memory is providing either an instruction or data.
- This performance limit is the Von Neumann or Princeton bottleneck.

When the EDVAC paper was written, electronic memory was in its infancy and very expensive. The EDVAC report proposed a shared memory where both data and instructions would be held as binary numbers. The central processing unit would place a number on the address bus to specify a memory location to retrieve data from. The memory would energize the data result on the data bus.

- This communication handshake would occur one retrieved number at a time.
- This single output port memory throttled number flow through time.
- With a single output port, the memory could only retrieve **either** an instruction **or** data at any moment in time – but not both.
- Historically called the Von Neumann bottleneck, architects today call this the **Princeton bottleneck** because von Neumann was working at the Princeton Institute for Advanced Studies (IAS).

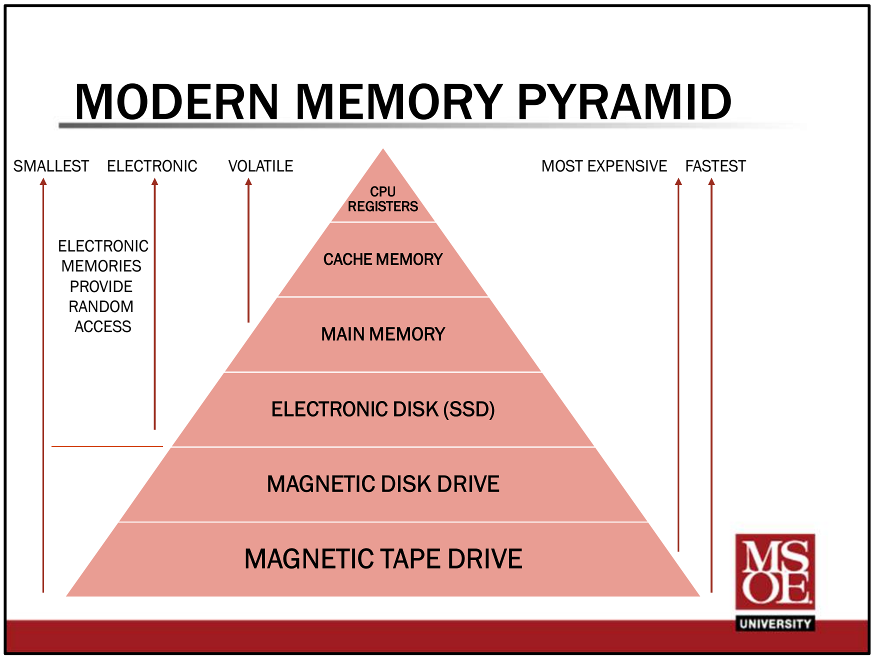


HARVARD MEMORY ARCHITECTURE

- One way to increase performance is to remove the Princeton bottleneck.
- The Harvard Mark 1 machine used separate paths for instructions and data.
- The Harvard architecture

When memory is large and cost-prohibitive, it may make sense to limit performance with the Princeton bottleneck because it helps control cost. But, when high performance is demanded, separating instructions and data into two separate memories drastically improves performance. This is an example of design tradeoffs when optimizing speed, size, power, and cost.

- The Harvard Mark 1 architecture described separate instruction and data paths. Remember that the Mark 1 was an electro-mechanical computer and pre-dates the EDVAC report. The Mark 1 read instructions from paper tape and thus was not an electronic stored-program machine.
- Image an address binary number broadcast on an **address bus**. This binary number is some value on a number line from 0 to the maximum number that can be stored on the bus. This number line is called the **address space**. Every number is a unique memory address within that address space.
- In a Princeton architecture, the instructions and data share the same address bus and thus the same address space.
- In a Harvard architecture, the instructions and data have distinct memories and distinct address busses and thus the CPU can access both at any moment in time.



MODERN MEMORY PYRAMID

SMALLEST

ELECTRONIC

VOLATILE

ELECTRONIC MEMORIES PROVIDE RANDOM ACCESS

CPU REGISTERS

CACHE MEMORY

MAIN MEMORY

ELECTRONIC DISK (SSD)

MAGNETIC DISK DRIVE

MAGNETIC TAPE DRIVE

MOST EXPENSIVE

FASTEST

MSOE UNIVERSITY

Through the decades, memory improved, and memory prices came down. Memory architects applied optimization in speed, size, power, and cost and provided different types of memories for use in various application domains. This led to the concept of layering memory as a pyramid based on speed and size.

PYRAMID BASICS

- **Electronic** memories store numeric bits as voltages while **magnetic** memories store a bit by altering the magnetic field of a magnetic material.
- **Volatile memories** forget stored values when power is removed. **Non-volatile memories** retain values through power cycles.
- **Electronic memories** are integrated electrical circuits providing **random access**.
- **Random access memories** retrieve any specified value in constant time.
- Magnetic memories provide **sequential access** because the hard disk motor or the tape motor must advance the hard disk platter or tape reel past potentially many locations before the desired byte arrives under the read/write head.
- Sequential access memories retrieve values in a **variable amount of time** that depends on data distance from the read/write head.

OPERATION

- Data **moves up** the memory pyramid from the slower, non-volatile storage devices to the volatile main memory to the volatile cache to the volatile CPU registers.
- This process of moving data up is called **paging** as **pages** of data are copied from the larger memory into the smaller memory.
- Paging is coordinated by various memory and I/O **controllers** layered in the memory system.

LAYERS

- The **CPU register file** is at top of the memory pyramid. This is a small collection of storage registers directly connected to the ALU.
    - The CPU register file today typically contains 16 or 32 numbers.
    - The CPU register file is very high speed because the address decode logic is simple and results in minimal delay.

- **Cache** memory is system memory not visible to the user program.
    - User programs access main memory.
    - Main memory is **paged** into the smaller system cache memory on demand.
    - The CPU can access the paged data more quickly from the smaller cache memory.
    - Cache memory is implemented as static random-access memory (SRAM) that does not require periodic refreshing when powered.

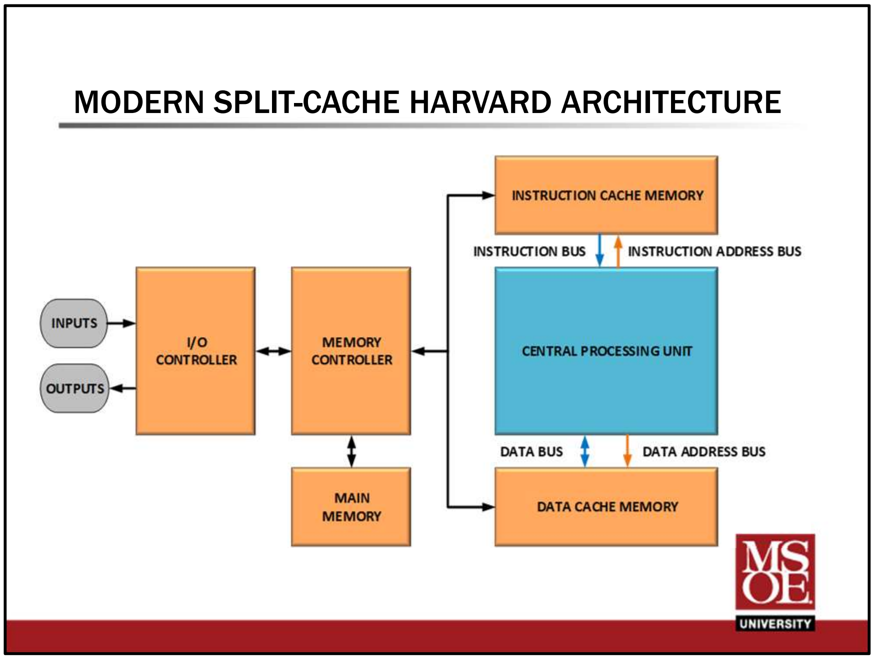
- **Main memory** is user memory accessed by software programs.
    - Of course, the CPU never truly modifies any location in main memory because it actually access the cache memory page.
    - The memory controller is responsible for writing page values back to main memory if a page is removed from the cache.
    - Main memory is implemented with dynamic random-access memory (DRAM) that does require periodic refreshing when powered.

- **Electronic disks** are solid-state drives built from integrated circuit non-volatile RAM memory chips.
    - Non-volatile RAM is slower than volatile RAM.
    - SSDs now overlap magnetic hard drives in terms of size. But the speed and cost-per-bit of SSD keeps it placed higher up the memory pyramid.

RULES OF THUMB

- A rule of thumb in computer architecture is **simple implies fast**.

- Another rule of thumb in computer architecture is that **speed implies higher cost**. It is certainly true in computer memory faster memory circuits are more expensive per bit.



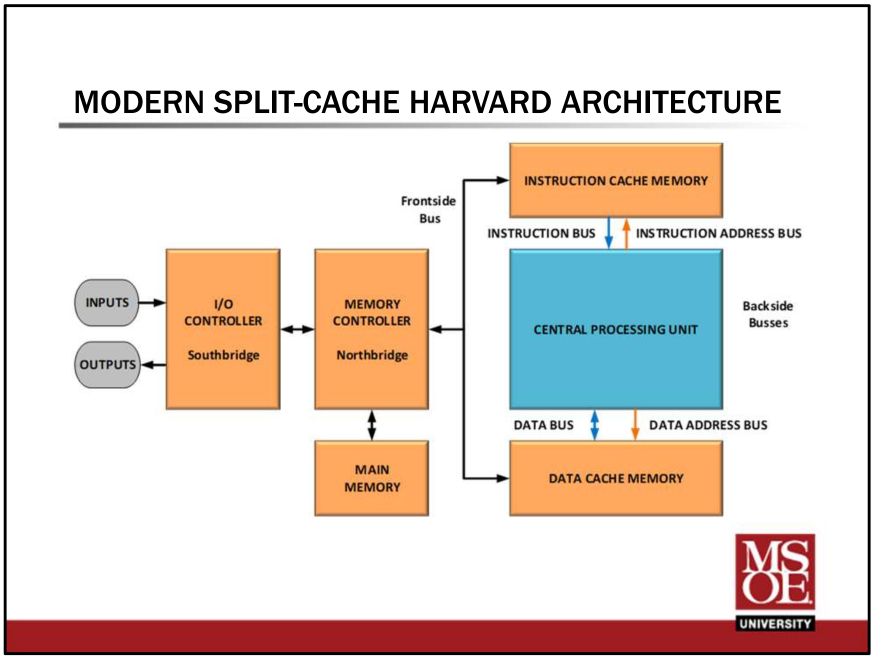
MODERN SPLIT-CACHE HARVARD ARCHITECTURE

The stored-program electronic computer industry is nearly a century old. It has matured through room-sized machines consuming massive amounts of electrical power to micro-miniaturized single-chip computers requiring minimal footprint and operating for days off rechargeable batteries. The computer concepts described by John von Neumann in his EDVAC report have been deeply studied and refined by computer scientists and engineers. Today, most computers are super-fast micro-miniaturized machines connected to the much slower human manipulated I/O devices like keyboards, mice, touchscreens, stylus-pens, and headsets. This micro-miniaturized machine still implements the basic principle of the Princeton memory architecture: programs and data sharing one main memory. However, the central processing unit uses instructions and data that have been paged into separate cache memories accessed through separate busses. A **modern split-cache Harvard memory architecture** is implemented at the upper levels of the memory pyramid.

- The CPU can access instructions and data in the same time unit. The Princeton bottleneck does not exist.
- High speed access is provided by the smaller cache memories.
- Both instructions and data are stored in the same main memory (stored-program in Princeton form).

- A **memory controller** controls on-demand paging from main memory into the cache memories.
- The memory controller also controls on-demand retrieval of programs from non-volatile I/O storage devices into main memory.
- The memory controller **owns** main memory.
- An I/O controller is used to move user input data from input devices through the memory controller and into main memory.
- An I/O controller is used to move calculated data from main memory through the memory controller and out to output devices.

Thus, modern machines are Princeton memory architectures with a **performance boost** provided by the split-cache Harvard memory architecture implemented at the upper levels.

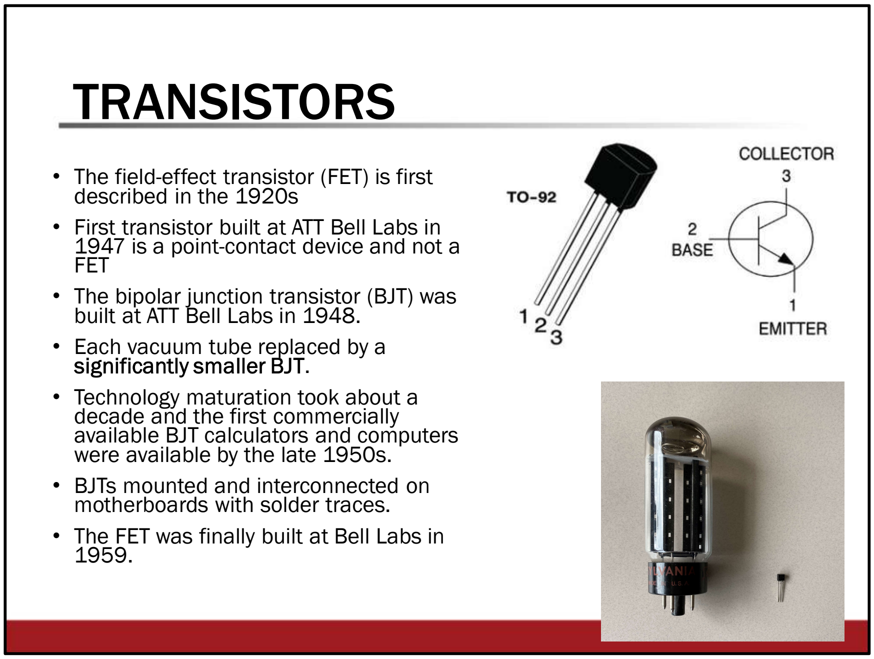


MODERN SPLIT-CACHE HARVARD ARCHITECTURE

Intel introduced a set of terminology that describes the busses and controllers it produced for manufacturers to use on motherboard when implementing the memory pyramid. Other companies also adopted these terms.

- The term **hub** or **bridge** was used to describe the components responsible for memory and I/O devices.
- The **memory controller hub** was also called the **northbridge.**
- The **I/O controller hub** was also called the **southbridge**.
- Rotate this image in your head so that the CPU is at the top. Then, the memory controller is **north** of the I/O controller.
- The **frontside bus** connected the cache memories to the memory controller.
- The **backside busses** connected the CPU to its instruction and data cache memories.

As silicon density increased, Intel integrated first the northbridge and then the southbridge onto the same chip with the CPU. Other companies integrated as well. Thus, while the terms are still in use, newer terms are also emerging. For example, Intel now calls the circuitry that contains the northbridge functionality the **system agent**.



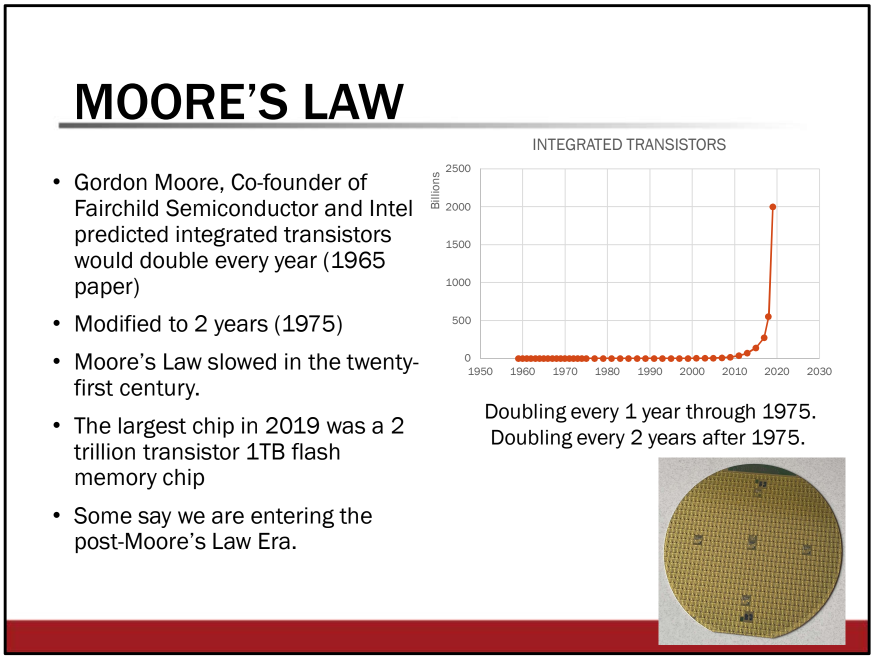
TRANSISTORS

- The field-effect transistor (FET) is first described in the 1920s
- First transistor built at ATT Bell Labs in 1947 is a point-contact device and not a FET
- The bipolar junction transistor (BJT) was built at ATT Bell Labs in 1948.
- Each vacuum tube replaced by a **significantly smaller BJT.**
- Technology maturation took about a decade and the first commercially available BJT calculators and computers were available by the late 1950s.
- BJTs mounted and interconnected on motherboards with solder traces.
- The FET was finally built at Bell Labs in 1959.

The invention of the transistor in 1947 at ATT Bell Labs resulted in the first miniaturization revolution and the **second generation** of computers in the late 1950s and the 1960s. Vacuum tubes were replaced by discrete transistors and machine drastically reduced in size. Computers that once filled rooms with thousands of vacuum tubes and relays could now be created with transistors on motherboards in cabinets that might be the size of one or two office desks. Transistors provided not only space savings, but also power savings as silicon transistors switched at consumer battery-level voltages rather than the hundreds or thousands of volts used to control vacuum tubes.

- The early twentieth century is one of the most remarkable times in physics. Quantum mechanics changed the way scientists viewed the atom and the structure of materials. By the 1940s the field of solid-state physics was active, and scientists were actively pursuing the **transistor**. A solid-state device has no moving parts. A transistor is a solid-matter electrical switch. It is engineered material that responds to electrical voltages or currents to form an electrical path between two terminals.
- Julius Lilienfeld had first described the transistor in the 1920s, but his **field-effect transistor** wasn't fabricated for three more decades.

- Scientists at ATT Bell Labs created a different type of transistor, the **point-contact transistor** in 1947. This was followed in 1948 by the **bipolar junction transistor (BJT)**. The junction transistor is the device used in the second generation of computers.
- Discrete transistors were packaged into metal, ceramic, or plastic cases. Each discrete transistor is one single switch. Gate and functional logic designers interconnected these discrete transistors on motherboards using solder traces.
- The field-effect transistor was finally created in 1959 in the form of a **metal-oxide-semiconductor** (MOSFET) device. When compared with BJT devices, MOSFETs consume less power and switch faster. And, as MOSFETs scale to smaller dimension their power-density stays constant. This means that their power usage gets smaller as the device gets smaller.



MOORE'S LAW

- Gordon Moore, Co-founder of Fairchild Semiconductor and Intel predicted integrated transistors would double every year (1965 paper)
- Modified to 2 years (1975)
- Moore's Law slowed in the twenty-first century.
- The largest chip in 2019 was a 2 trillion transistor 1TB flash memory chip
- Some say we are entering the post-Moore's Law Era.

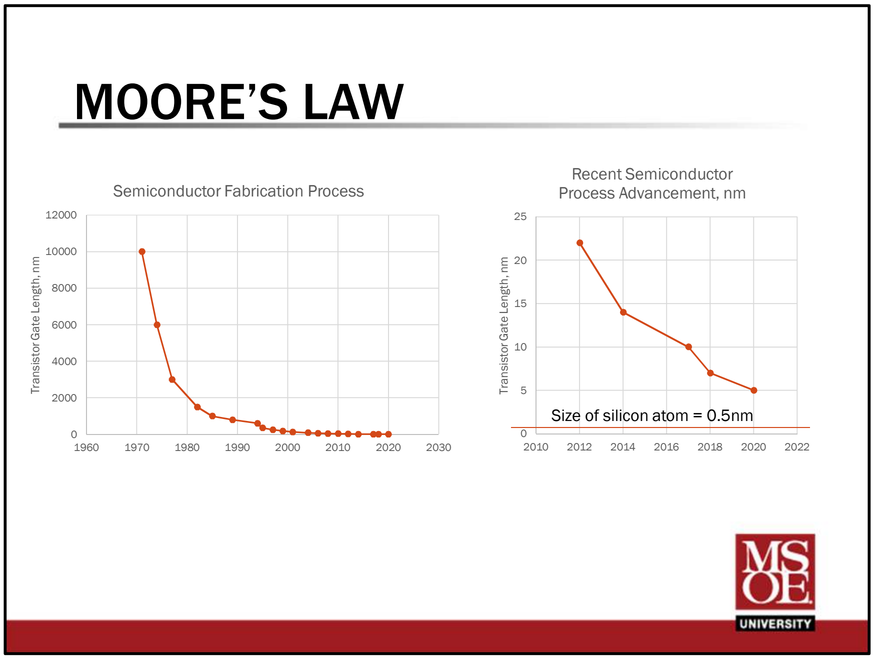
Doubling every 1 year through 1975.
Doubling every 2 years after 1975.

The development of engineered material that could be controlled like a switch led to the industry aggressively developing fabrication techniques and a new industry was born: the **semiconductor industry**. Process engineers work with chemists, physicists, metallurgists, optics specialists, and others to devise techniques to control the chemistry of silicon and other semiconductor materials. The results is **semiconductor wafer fabrication** creating **integrated circuits** containing many interconnected transistors built in microminature.

- An i**ntegrated circuit** is a micro-miniaturized circuit fully implemented on a piece of material. Silicon is the most-used semiconductor material because it is abundant on earth and has good electrical semiconductor properties. Each micro-miniaturized circuit is called a **die** and multiple dies are replicated on the surface area of a round **wafer**.
- The semiconductor industry begins its maturation curve in the 1960s.
- Gordon Moore predicted that transistors would double every year in the first decade of the industry. He changed his position to every two years by the mid-1970s. Carver Mead, a professor at Caltech, named this **Moore's Law**.
- Moore's prediction has been quite accurate through the decades. The plot shows the rapid growth in the 21st century as the doubling hit the billions and then the

trillions.

- Moore predicted that we would be fabricating transistor features that would be nearing the actual size of the silicon atom (rather than being bigger than it) by around 2025. Thus, the industry and media often say we are approaching the post-Moore's Law Era.
- To compensate for the transistor feature sizes approaching the size of the atom, the industry has responded with innovations that they hope will keep Moore's Law alive for many years to come. Examples include FinFETs and 3-D stacking.



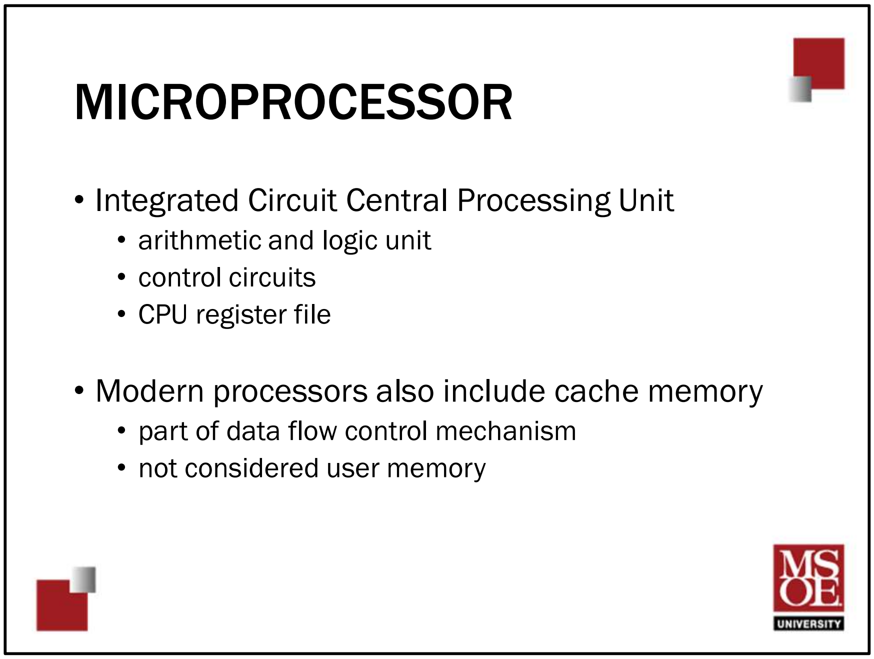
MOORE'S LAW

Semiconductor Fabrication Process

Recent Semiconductor Process Advancement, nm

These plots show device scaling approaching the size of the silicon atom. The first plot shows the length of transistor control gates shrinking from the 1970s to today. The plot is exponentially approaching an asymptote. The second plot is a zoomed in look at the approach from 2012 to 2020.

Moore's Law is an amazing industry trend. The semiconductor industry kept pace with the prediction and micro-miniaturized computers have revolutionized our word. A 2018 Computer History Museum report noted that more than a sextillion MOSFETs have been fabricated – making the MOSFETs the most built device in history.



MICROPROCESSOR

- Integrated Circuit Central Processing Unit
  - arithmetic and logic unit
  - control circuits
  - CPU register file
- Modern processors also include cache memory
  - part of data flow control mechanism
  - not considered user memory

MSOE
UNIVERSITY

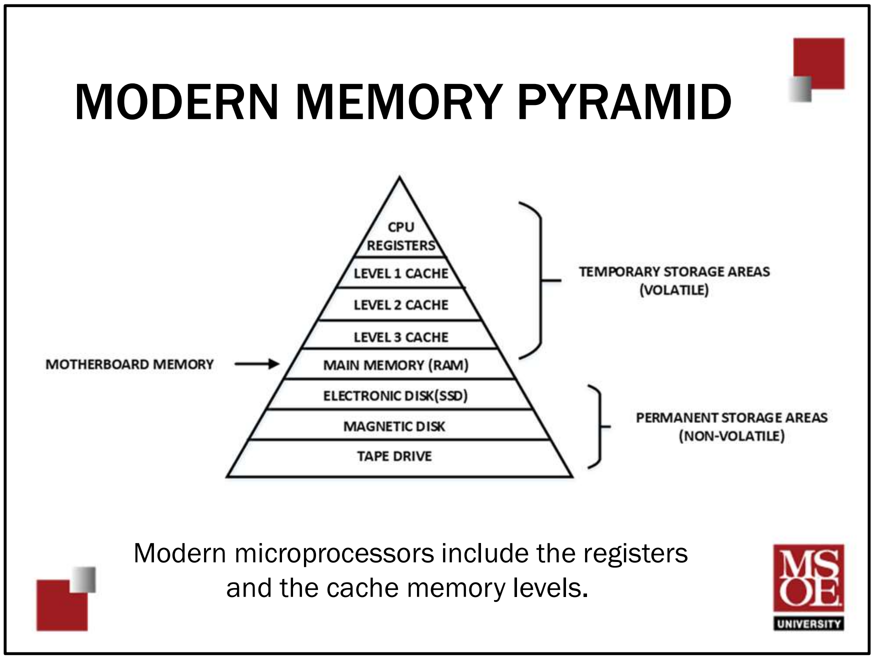
As the semiconductor industry began wafer fabrication, computer architecture moved from the macroscopic to the microscopic. Companies first began fabricating digital logic gates and functional logic parts as silicon density remained too low to fabricate an entire processor. Logic gates, adders, ALUs, decoders, multiplexers, flip-flops, registers, and encoders are common parts used in computer design that could each be purchased as individual chips and wired together to form a computer.

- Texas Instruments released the 7400-logic family in 1964. The TI 74181 was the first single-chip ALU.
- RCA released the CD4000 family in 1968. Fabricated from MOSFETs, these devices were low-power compared to the 7400 BJT based devices.

Silicon density progresses through the 1960s into the early 1970s following Moore's Law. Gordon Moore and Robert Noyce found Intel in 1968 as an **int**egrated **el**ectronics company. They founded the company to produce semiconductor memory to replace magnetic-core memory. Thus, Intel's history begins as a **memory company.** By 1971, Intel had also created the first commercially available integrated circuit processor – a **microprocessor.** The microprocessor further reduced the footprint required by desktop calculators and computers.

The central processing unit has historically been defined as the ALU, the control unit, and the registers that provide data to the ALU. The first microprocessor, the Intel 4004, implemented this historic model. Then, computer system integration basically follows what is allowed by Moore's Law.

- Chipsets were created to provide functionality when connected on a motherboard: microprocessor, memory controllers, I/O controllers, cache memory chips, and main memory chips would all be fabricated and sold individually or as a set.
- Integration progressed down the memory pyramid. As silicon density increased, architects began integrating instruction and data cache memories to form the modern split-cache Harvard architecture.
- As we enter the third full decade of the twenty-first century, architects continue to use silicon density to integrate more of the computer system onto one chip. Intel has moved the north and south bridges on-chip as well. This fully integrated chip becomes a **system-on-chip** (SOC) form-factor.

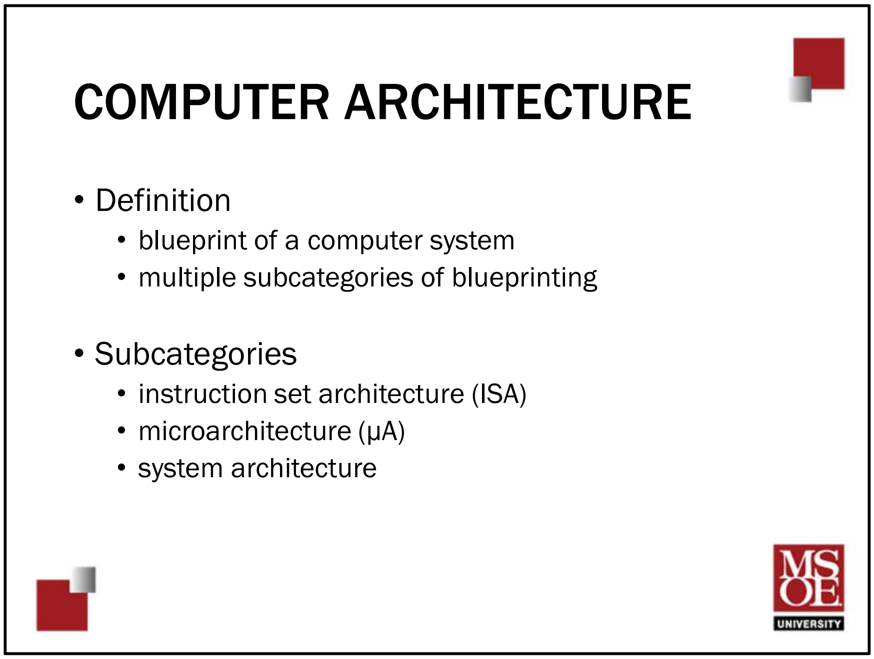


MODERN MEMORY PYRAMID

Modern microprocessors include the registers and the cache memory levels.

The ability to integrated billions of transistors means modern microprocessors contain up to three levels of cache memory. The split-cache model uses separate level 1 instruction and data caches, a shared level 2 cache, and a shared level 3 cache.

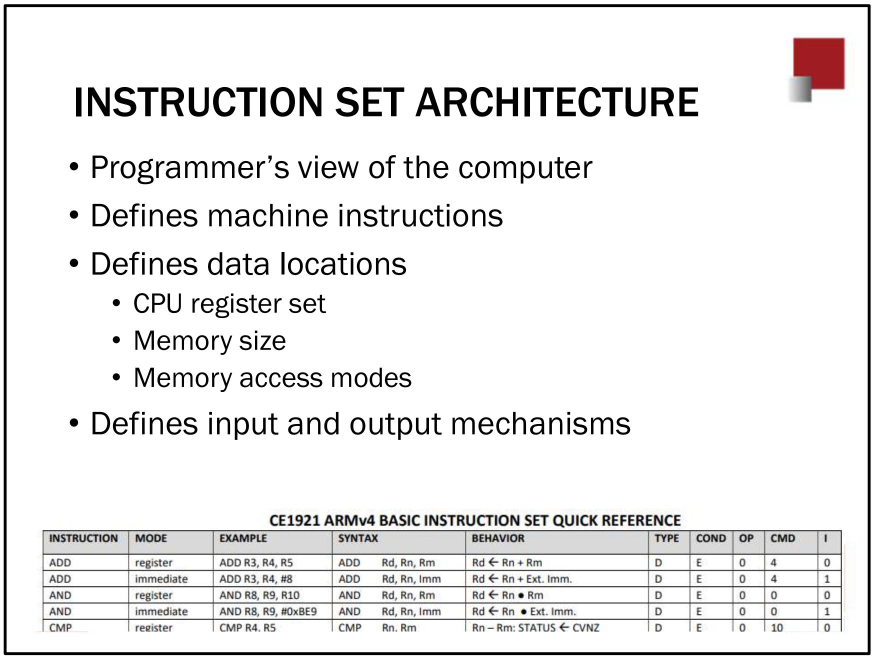
- Why are there three levels of cache? Remember, small implies fast. Segmenting cache memory into progressively smaller pieces gives a performance boost.
- The user RAM is the first level of motherboard memory. Large RAMs remain as separate motherboard items to allow for variability in memory size and user upgrades. (Of course, RAM could be integrated and some microprocessor SOCs do integrate an amount of user RAM).



COMPUTER ARCHITECTURE

- Definition
  - blueprint of a computer system
  - multiple subcategories of blueprinting
- Subcategories
  - instruction set architecture (ISA)
  - microarchitecture (μA)
  - system architecture

Computer architects blueprint computer systems. The computer must be created in the mind of the designers and then realized as circuitry. As complex systems, blueprinting is done in multiple ways. This results in subcategories of computer architecture. Each of these subcategories has existed since the first computers were built. The development of micro-miniaturization renamed one subcategory from circuit architecture to microarchitecture.



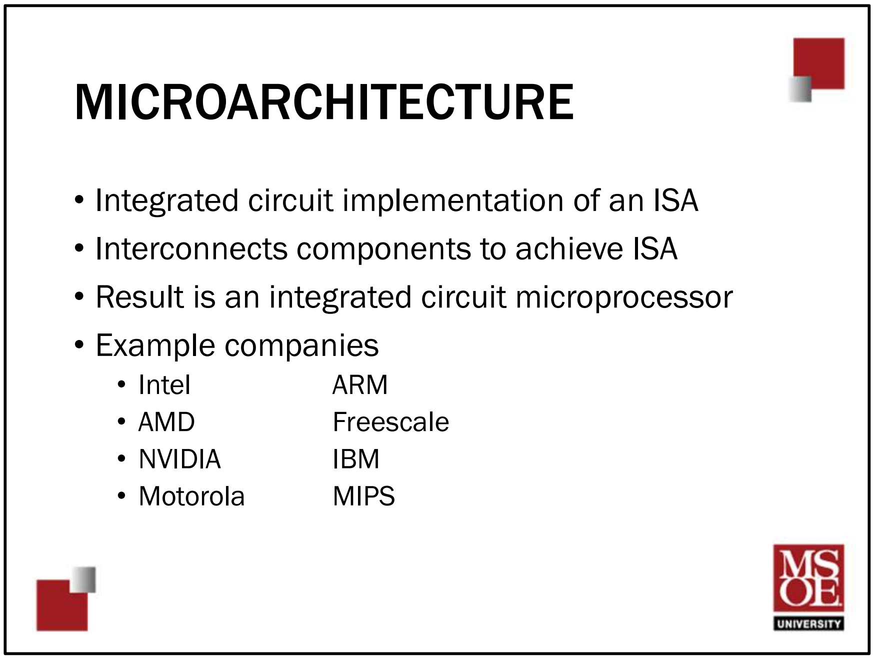
INSTRUCTION SET ARCHITECTURE

- Programmer's view of the computer
- Defines machine instructions
- Defines data locations
  - CPU register set
  - Memory size
  - Memory access modes
- Defines input and output mechanisms

CE1921 ARMv4 BASIC INSTRUCTION SET QUICK REFERENCE

INSTRUCTION	MODE	EXAMPLE	SYNTAX	BEHAVIOR	TYPE	COND	OP	CMD	I
ADD	register	ADD R3, R4, R5	ADD Rd, Rn, Rm	Rd ← Rn + Rm	D	E	0	4	0
ADD	immediate	ADD R3, R4, #8	ADD Rd, Rn, Imm	Rd ← Rn + Ext. Imm.	D	E	0	4	1
AND	register	AND R8, R9, R10	AND Rd, Rn, Rm	Rd ← Rn ● Rm	D	E	0	0	0
AND	immediate	AND R8, R9, #0xBE9	AND Rd, Rn, Imm	Rd ← Rn ● Ext. Imm.	D	E	0	0	1
CMP	register	CMP R4, R5	CMP Rn, Rm	Rn – Rm: STATUS ← CVNZ	D	E	0	10	0

The **instruction set architecture** is defined using words, charts, and tables. Instruction set architecture describes the machine as a programmer would view it. For this reason, it is often called the **programmer's model** of the machine. Programmers need to know what instructions they can use to command the computer to algorithmically calculate a result. They also need to know what memory, input devices, and output devices look like. This slide shows the start of a table of instructions for the ARMv4 ISA.



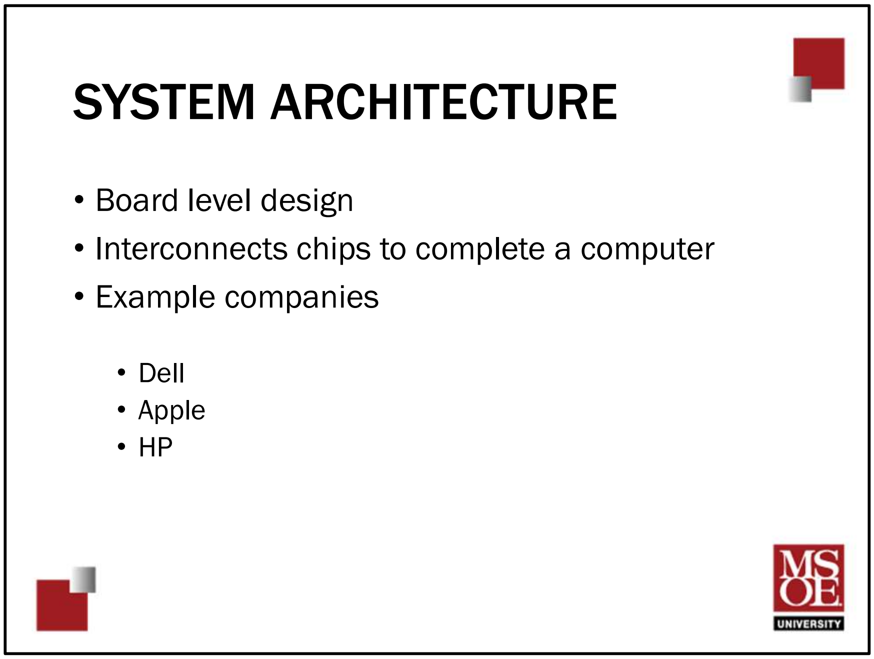
MICROARCHITECTURE

- Integrated circuit implementation of an ISA
- Interconnects components to achieve ISA
- Result is an integrated circuit microprocessor
- Example companies
  - Intel ARM
  - AMD Freescale
  - NVIDIA IBM
  - Motorola MIPS

Architects must implement the programmer’s model using integrated transistors interconnected to form digital logic such as gates, multiplexers, ALUs, decoders, encoders, and registers. The result is integrated circuit chips.

- The term **microarchitecture** implies the actual micro-miniaturized integrated circuit chip that implements an ISA.
- Different microarchitectures can implement the same ISA. Each results in a **microprocessor**. For example, Intel produces different microprocessors that implement its x86-64 ISA. Some are optimized for desktop machines, others for servers, and others for mobile devices.
- Companies give their microarchitectures great names. For example, Intel recently used CoffeeLake and Ice Lake. AMD will soon release a microprocessor built from its AMD Zen 3 microarchitecture that also implements the x86-64 ISA.

Many companies are well-known microarchitecture foundries.

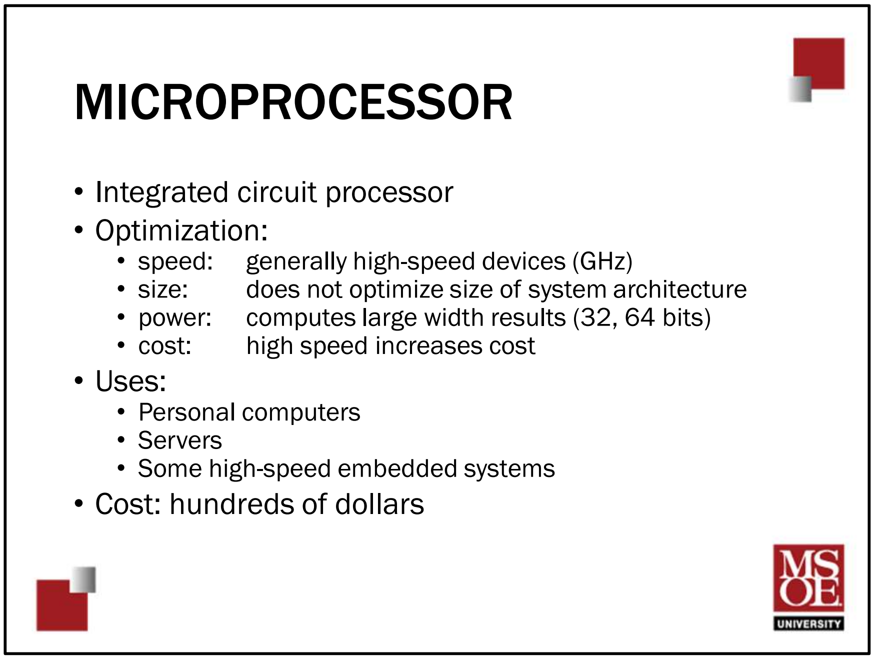


SYSTEM ARCHITECTURE

- Board level design
- Interconnects chips to complete a computer
- Example companies
  - Dell
  - Apple
  - HP

Once microarchitecture companies have created integrated circuits, these integrated circuits are combined, interconnected on motherboards, interfaced to I/O devices, placed in casing, branded and marketed to consumers.

- Sometimes microarchitecture companies also create and sell computer systems. Apple is an example of a company that has microarchitects working on chips such as the A13 Bionic system-on-chip that is used in the iPhone 11. Released in 2019, this chip integrates the 64-bit ARM ISA as a microprocessor, machine learning blocks, and a GPU.
- Other times, original equipment manufacturers purchase chips from a microarchitecture foundry like Intel or AMD and then complete system design, build, and test. Dell, HP, Lenovo, and Acer are some examples.

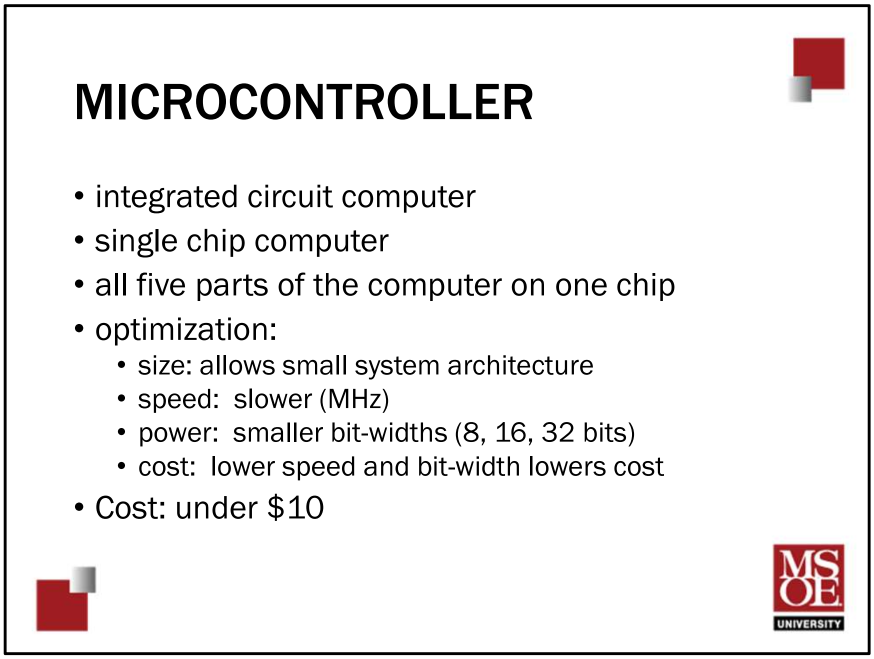


MICROPROCESSOR

- Integrated circuit processor
- Optimization:
  - speed: generally high-speed devices (GHz)
  - size: does not optimize size of system architecture
  - power: computes large width results (32, 64 bits)
  - cost: high speed increases cost
- Uses:
  - Personal computers
  - Servers
  - Some high-speed embedded systems
- Cost: hundreds of dollars

As the semiconductor industry ramped out large-scale and very-large scale integration, the microarchitecture industry split into two different branches on order to serve the needs of high-performance computing and embedded systems.

- Microprocessors are generally optimized to create wide number lines at high-speed. Cache memory is usually implemented on-chip to help with memory access speed. Microprocessors have traditionally been targeted to the personal computer and server markets and today processors create 64-bit numbers with GHz clocks.

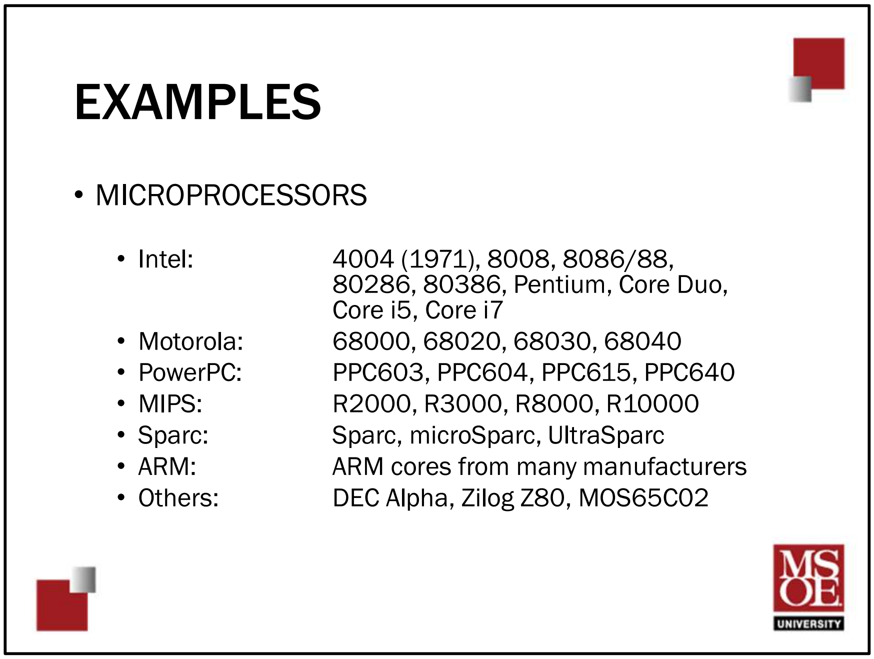


MICROCONTROLLER

- integrated circuit computer
- single chip computer
- all five parts of the computer on one chip
- optimization:
  - size: allows small system architecture
  - speed: slower (MHz)
  - power: smaller bit-widths (8, 16, 32 bits)
  - cost: lower speed and bit-width lowers cost
- Cost: under $10

The other branch of the microarchitecture industry used silicon space to integrate an entire system-on-chip. This resulted in a single-chip computer targeted to the embedded systems market. This **single-chip computer** has been generally known as a **microcontroller**.

- Embedded systems cannot support the higher price points that personal computers and servers can support. The public doesn’t want to spend $2000 on a microwave or a clock-radio.
- Embedded systems usually do not need high-speed calculation and most don’t need very large number lines.
- Remember fast implies expensive is an architecture rule-of-thumb.
- To meet the price-points, architects created slower-speed, smaller bit-width devices and integrated user RAM, program memory, and I/O devices as well.
- Microcontrollers exist that are one-time programmable and cost around $1.
- Most microcontrollers cost under $10.



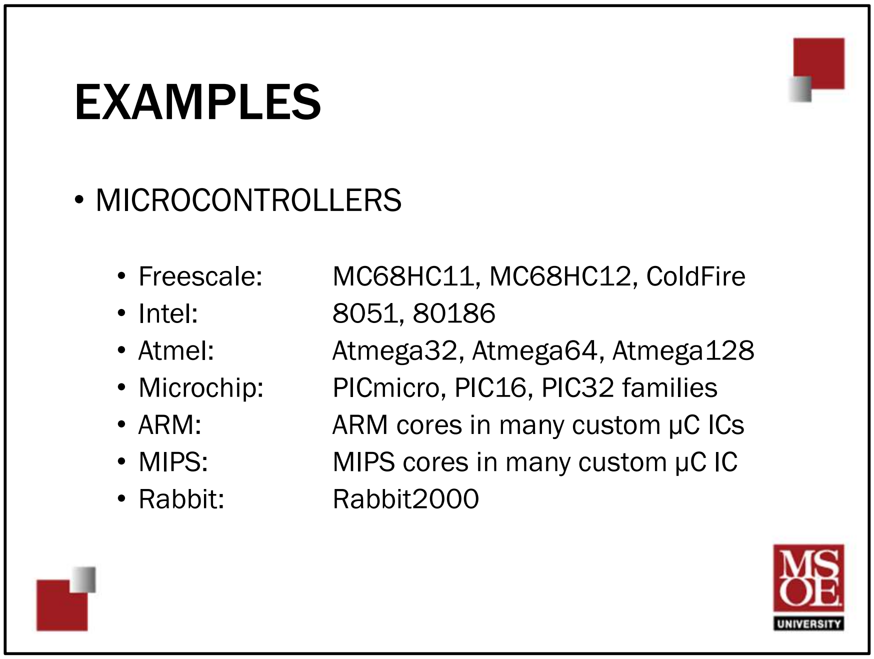
EXAMPLES

- MICROPROCESSORS

• Intel:	4004 (1971), 8008, 8086/88, 80286, 80386, Pentium, Core Duo, Core i5, Core i7
• Motorola:	68000, 68020, 68030, 68040
• PowerPC:	PPC603, PPC604, PPC615, PPC640
• MIPS:	R2000, R3000, R8000, R10000
• Sparc:	Sparc, microSparc, UltraSparc
• ARM:	ARM cores from many manufacturers
• Others:	DEC Alpha, Zilog Z80, MOS65C02

This list of microprocessors is, of course, not a complete set. Thousands of different microprocessors have been fabricated implementing many, many instruction set architectures. This list provides some historically important examples for your reference. Read about some of the ISAs or processor chips to broaden your historical perspective on the computer industry.

- One of my favorites is the 6502 – the 8-bit microprocessor that helped bring computers into the home. Apple, Commodore, Atari, and Nintendo all used this chip!
- Another of my favorites is the 68000 – a 32-bit ISA microprocessor used in the original Macintosh computers and the Commodore Amiga. It’s higher bit-width and speed helped enable the graphical user interface.

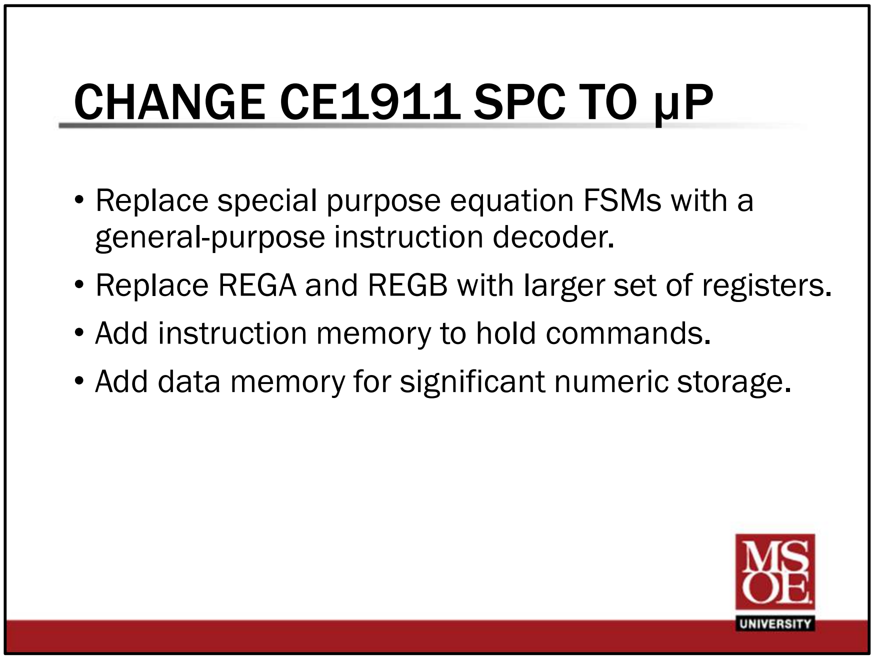


EXAMPLES

- MICROCONTROLLERS

• Freescale:	MC68HC11, MC68HC12, ColdFire
• Intel:	8051, 80186
• Atmel:	Atmega32, Atmega64, Atmega128
• Microchip:	PICmicro, PIC16, PIC32 families
• ARM:	ARM cores in many custom μC ICs
• MIPS:	MIPS cores in many custom μC IC
• Rabbit:	Rabbit2000

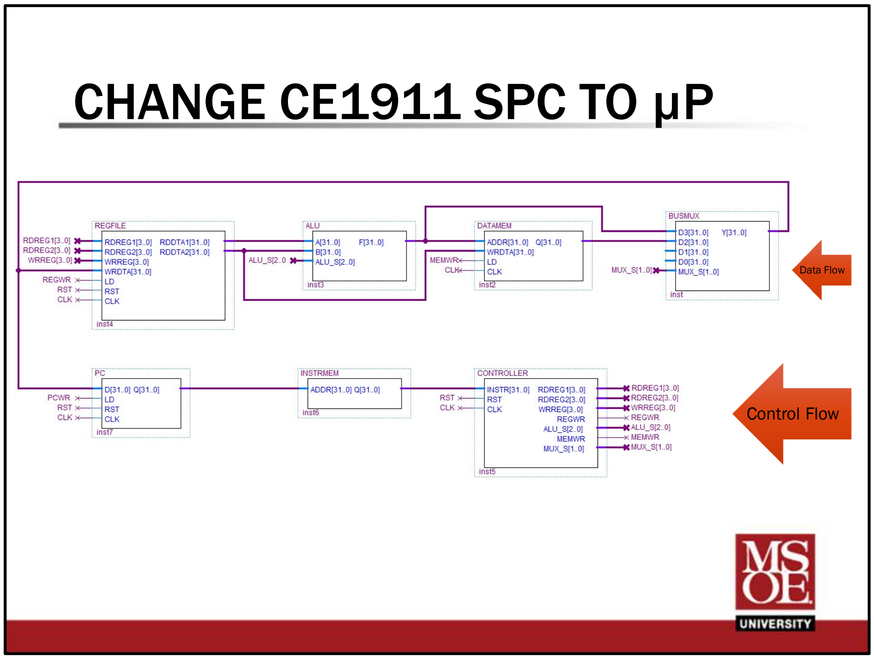
Like the list of microprocessors, this list of microcontrollers provides you with some historical companies and chips. While we will study the ARM ISA, the 6800 ISA from Motorola (now Freescale) and the MIPs ISAs are some of my personal favorites.



CHANGE CE1911 SPC TO μP

- Replace special purpose equation FSMs with a general-purpose instruction decoder.
- Replace REGA and REGB with larger set of registers.
- Add instruction memory to hold commands.
- Add data memory for significant numeric storage.

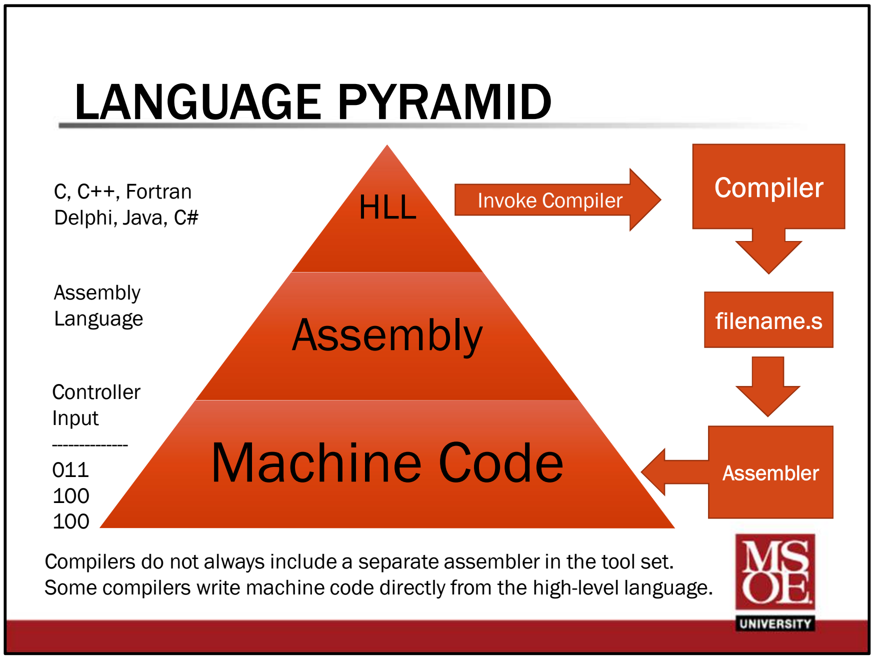
As we move from basic logic design in CE1901 and CE1911, we can apply what we have learned in this slide set about computer architecture basics to determine how we might change our CE1911 special-purpose computer into a general-purpose processor. These bullets hint at how we can apply the principles of the EDVAC report. The diagram on the next slide shows some of the changes in place.



CHANGE CE1911 SPC TO µP

After adding a larger set of registers as well as Harvard memory architecture, the CE1911 SPC now looks like a general-purpose device. Two paths of electrical flow exist.

- **Data flow** moves through the data memory pyramid and the ALU.
- Instructions flows from the instruction memory through the controller. This is known as **control flow**.



LANGUAGE PYRAMID

C, C++, Fortran
Delphi, Java, C#

Assembly
Language

Controller
Input

011
100
100

HLL

Assembly

Machine Code

Invoke Compiler

Compiler

filename.s

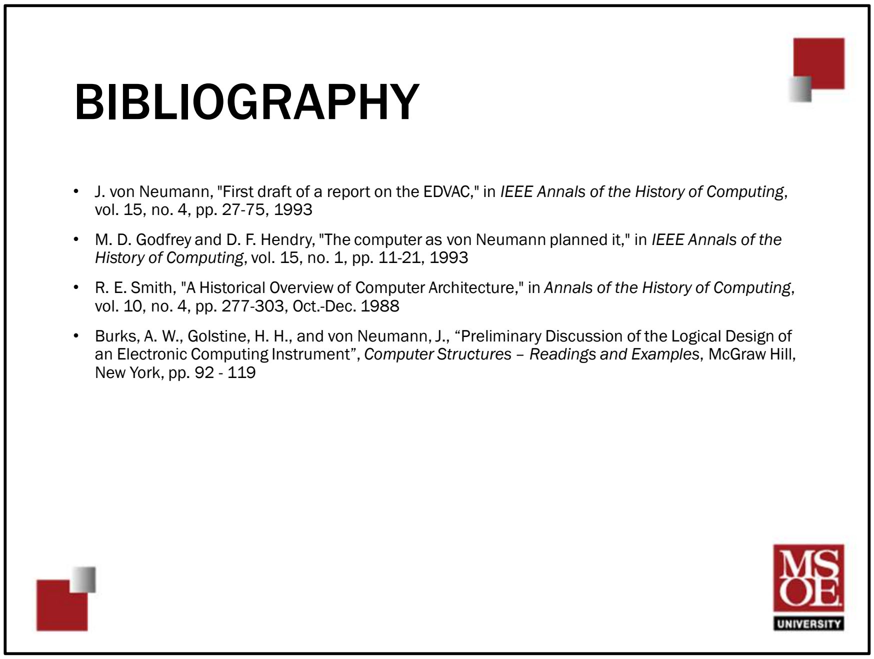
Assembler

Compilers do not always include a separate assembler in the tool set.
Some compilers write machine code directly from the high-level language.

MSOE UNIVERSITY

Finally, it is appropriate to abstractly think about how software becomes instructions stored in memory as binary voltages.

- Today, computer algorithms are written in high-level languages like C, C++, Fortran, etc.
- **Compilers** are programs that convert the high-level language into ISA form. The ISA form is known as **assembly language.**
- **Assemblers** convert ISA form programs into **machine code**.
- **Machine code** is a set of sequentially stored binary numbers representing the sequentially-ordered instructions of the program.



BIBLIOGRAPHY

- J. von Neumann, "First draft of a report on the EDVAC," in *IEEE Annals of the History of Computing*, vol. 15, no. 4, pp. 27-75, 1993
- M. D. Godfrey and D. F. Hendry, "The computer as von Neumann planned it," in *IEEE Annals of the History of Computing*, vol. 15, no. 1, pp. 11-21, 1993
- R. E. Smith, "A Historical Overview of Computer Architecture," in *Annals of the History of Computing*, vol. 10, no. 4, pp. 277-303, Oct.-Dec. 1988
- Burks, A. W., Golstine, H. H., and von Neumann, J., “Preliminary Discussion of the Logical Design of an Electronic Computing Instrument”, *Computer Structures – Readings and Examples*, McGraw Hill, New York, pp. 92 - 119

Here are some papers from the IEEE literature If you would like to read more about EDVAC and John von Neumann. You have access to these papers through the MSOE library website. Use **alphabetical list of databases → IEEE Xplore.**