Introduction to Digital Design and Computer Architecture

1. Basics of Digital Design

Lilia Kirakosyan

Russian-Armenian University

2025

Course Materials and Tools

- **Textbook:**
  - *Digital Design and Computer Architecture* by David Harris and Sarah Harris
- **Software:**
  - **Quartus Prime:** A powerful FPGA design software used for programming and simulating digital circuits.
- **Development Board:**
  - **Terasic Cyclone V Development Kit:** A versatile FPGA board used for implementing and testing digital designs.

Game Plan

- Introduction to FPGA and ASIC
- Resistors, Transistors, Diodes, LEDs, and Semiconductors
- Logical Elements and Creation of Logic Circuits
- Logical Elements - D-Trigger and RS Trigger
- Quartus and Verilog Basics
- AND, OR, NOT, and XOR Gates Design
- Multiplexer (MUX)
- Decoder
- Seven-Segment Display
- VGA (Video Graphics Array)
- Shift Register
- Segment Word with Shift Register
- Counter
- Note Recognition
- Snail FSM Game

From Lamp to Computer

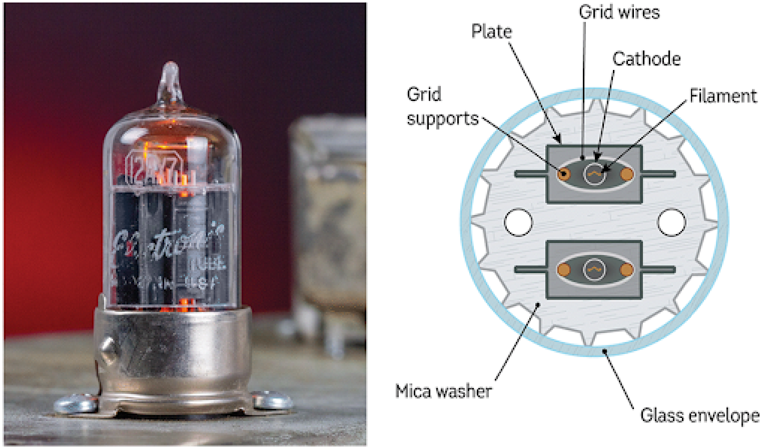


Figure: Vacuum Lamp

First Generation of Computers (1940-1956)

- **Vacuum Tubes:** The first generation of computers was based on vacuum tubes
- **Programming:** These computers used machine language.
  - Large size and heat generation
  - Limited processing power
  - High maintenance cost

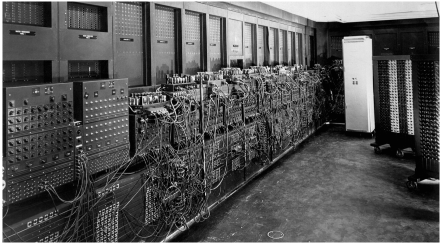


Figure: ENIAC: One of the first general-purpose computers.

Second Generation of Computers (1956-1963)

- **Transistors:** Replaced vacuum tubes, making computers smaller, faster, more reliable, and energy-efficient.
- **Programming:** High-level programming languages such as Cobol and Fortran.
    - Size Reduction
    - Improved Performance

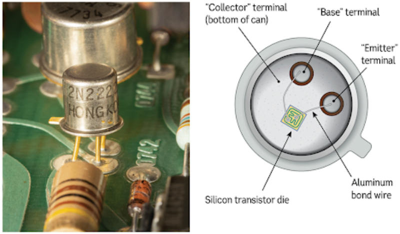


Figure: Transistor

Third Generation of Computers (1964-1971)

- **Integrated Circuits:** Allowed the integration of multiple transistors on a single chip, leading to smaller, more efficient, and more powerful computers.
- **Computing Evolution:**
  - Reduced Size
  - Increased Speed
  - Enhanced Reliability

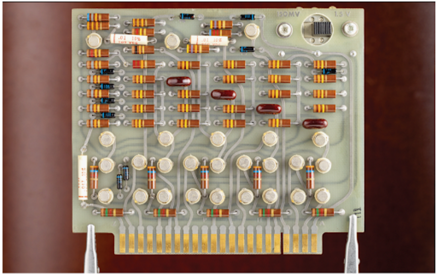


Figure: Integrated Circuit

Fourth Generation of Computers (1971-Present)

- **Microprocessors:** Integrated thousands of ICs onto a single silicon chip, marking the beginning of personal computers.
- **Key Features:**
  - Increased Processing Power
  - Miniaturization
  - Widespread Use of Personal Computers

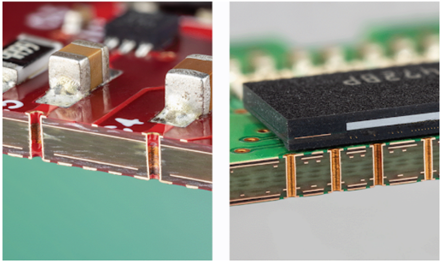
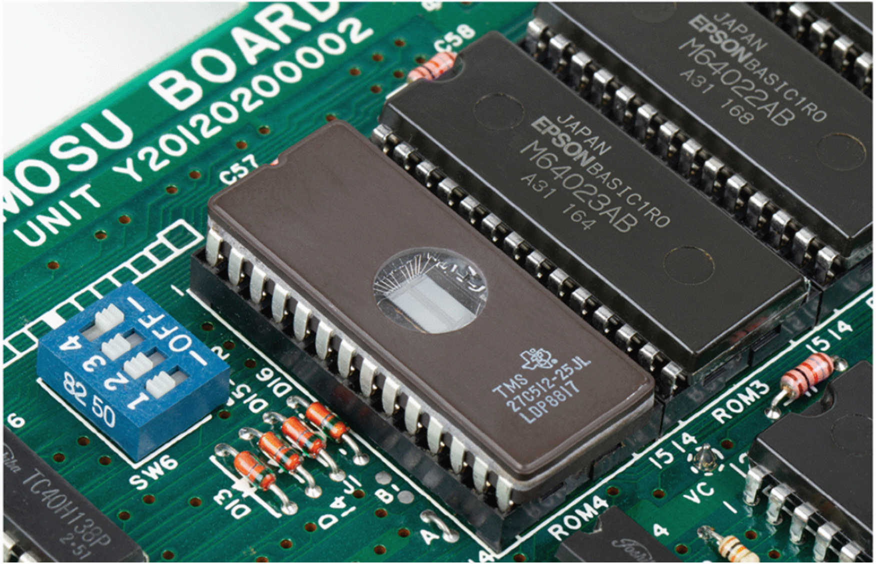


Figure: Microprocessor

Fourth Generation of Computers



Fourth Generation of Computers

Figure: IBM Personal Computer, 1981

What is an ASIC?

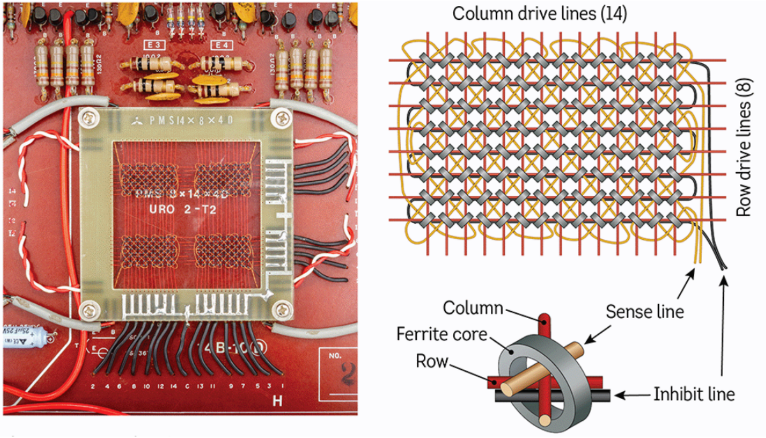
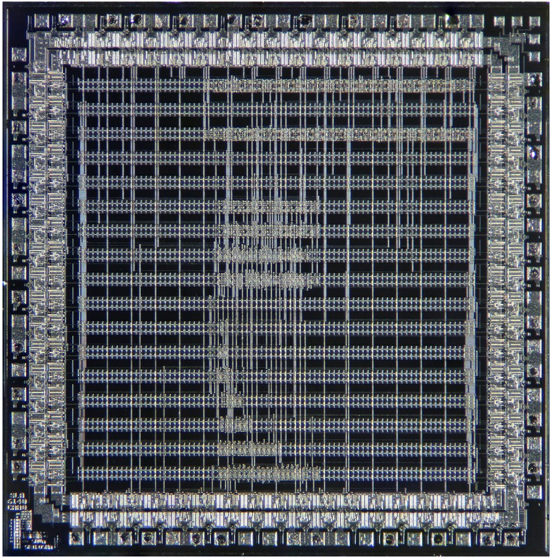


Figure: One of the first ASIC

Application-Specific Integrated Circuit (ASIC)

- **Definition:** An ASIC is a type of integrated circuit that is custom-designed for a specific application or purpose, rather than being intended for general-purpose use.
- **Characteristics:**
  - Tailored for particular tasks
  - High efficiency and optimized performance for specific functions
  - Reduced power consumption compared to multipurpose chips
- **Applications:**
  - Consumer electronics (e.g., smartphones, tablets)
  - Cryptography (e.g., Bitcoin mining)
  - Telecommunications (e.g., network routers)

Application-Specific Integrated Circuit (ASIC)



What is an FPGA?

- What are the key differences between an FPGA and an ASIC?
- How is an FPGA programmed or configured?
- What are some common use cases for FPGAs in modern technology?

A Field-Programmable Gate Array (FPGA)

- **Definition:** An FPGA is a type of integrated circuit that can be configured by the user after manufacturing to execute specific tasks, allowing flexibility and adaptability in hardware design.

A Field-Programmable Gate Array (FPGA)

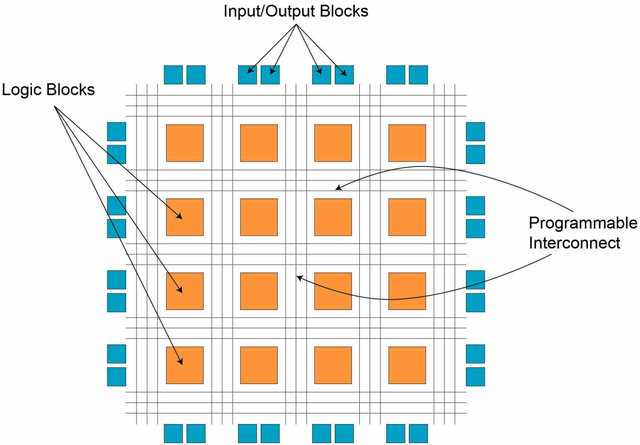


Figure: FPGA schematic representation

A Field-Programmable Gate Array (FPGA)

- **Definition:** An FPGA is a type of integrated circuit that can be configured by the user after manufacturing to execute specific tasks, allowing flexibility and adaptability in hardware design.
- **Applications:**
  - Digital signal processing (DSP)
  - Hardware acceleration for algorithms
  - Cryptography and secure communications
  - Embedded systems
  - Prototyping of digital circuits
- **Advantages:**
  - Reconfigurable hardware
  - High parallelism for performance improvement
  - Reduced time-to-market for hardware design

FPGA Programming Languages: VHDL and Verilog

- **VHDL (VHSIC Hardware Description Language):**
  - Used to describe the behavior and structure of digital systems.
  - Known for its strong typing and verbosity.
- **Verilog:**
  - A more compact HDL used for modeling hardware.
  - Often preferred for its similarity to C programming language.
- Both languages are used for:
  - FPGA design and simulation
  - Creating custom hardware circuits
  - Developing test benches

FPGA Design Flow

- **1. Design Specification:** Define functional requirements and performance goals.
- **2. HDL Coding:** Write code in VHDL or Verilog to describe the hardware behavior.
- **3. Synthesis:** Convert HDL code into a netlist (hardware logic gates).
- **4. Simulation:** Verify design functionality using simulation tools (e.g., ModelSim).
- **5. Implementation:** Perform place-and-route to map the design onto FPGA resources.
- **6. Programming:** Program the FPGA with the generated bitstream.

FPGA vs ASIC: Comparison

	FPGA	**ASIC**
Flexibility	Reconfigurable	Fixed for a specific task
Cost	High upfront cost	High NRE (Non-Recurring Engineering)
Performance	Lower than ASIC	Higher performance
Time to Market	Fast design changes	Longer design cycle
Power Consumption	Higher power consumption	Lower power consumption

Real-World Applications of FPGAs

- **Digital Signal Processing (DSP):**
  - Audio and video processing
  - Image enhancement and filtering
- **Machine Learning and AI Acceleration:**
  - FPGA-based neural network accelerators
- **Cryptography:**
  - Bitcoin mining
  - Secure communications
- **Telecommunications:**
  - 5G base station processing
  - Network packet filtering
- **Automotive Systems:**
  - Autonomous vehicle processing
  - Real-time sensor fusion

Challenges in FPGA Design

- **Complexity:** FPGA design involves managing large projects with numerous components.
- **Resource Limitations:** Limited logic elements, memory, and power budget.
- **Debugging:** Difficult to debug hardware issues compared to software.
- **Timing Issues:** Ensuring that all timing constraints are met is challenging.
- **Design Verification:** Verifying a design can be time-consuming, requiring extensive simulation and testing.

FPGA and Machine Learning Acceleration

- **Use Cases:**
  - FPGA-based acceleration of neural networks and deep learning models.
  - Real-time inference for edge devices.
- **Advantages:**
  - Parallel processing of AI workloads
  - Lower latency compared to general-purpose processors
- **Example:** Xilinx's Deep Learning Processing Unit (DPU) for accelerating AI workloads on FPGAs.

FPGA Development Tools

- **Quartus Prime (Intel FPGA):**
  - Comprehensive FPGA design software suite from Intel (formerly Altera).
- **Vivado (Xilinx):**
  - Design and verification tools for Xilinx FPGAs.
- **ModelSim:**
  - Simulation tool for VHDL/Verilog designs, used for functional verification.

Thank You!

Any Questions?