CSE 331 / EEE 332 / ETE 332/EEE 453 Microprocessor and Interfacing

DR. SHAIKH ASIF MAHMOOD
PROFESSOR, EEE, BUET

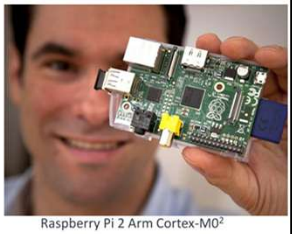
Introduction

- The modern computer is less than 100 years old.
- The first electromechanical and valve-based machines were produced in the 1930s and 1940s.
- Today's machines are many orders of magnitude faster, lower power, more reliable, and cheaper.

EDSAC was the **second electronic digital stored-program computer to go into regular service**

EDSAC replica (2018)[1]

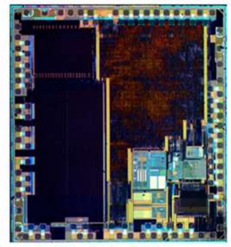
A single-board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer.



Raspberry Pi 2 Arm Cortex-M0[2]

Introduction

- The modern microprocessor is often the engine behind how we communicate and work.
- It has helped to create our digital world where communication, computation, and storage is almost free.
- It underpins many scientific breakthroughs and can help us make better use of the world's limited resources

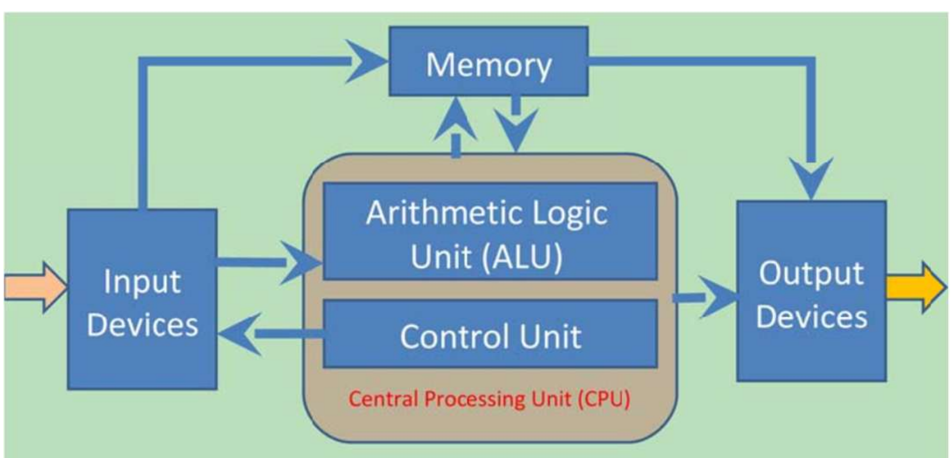


Die photo of Arm Cortex-M3 Microcontroller with 16KB flash memory and 4KB RAM[1]

The ARM Cortex-M is a group of 32-bit RISC ARM processor cores licensed by ARM. These cores are optimized for low-cost and energy-efficient integrated circuits, which have been embedded in tens of billions of consumer devices

Google Data Center[2]

Basic Elements of a Computer System



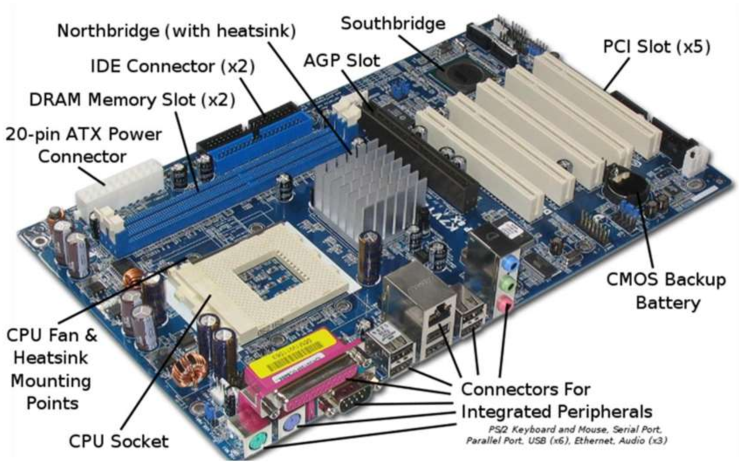
Computer System Components

- **Memory**
-Stores instructions and data
- **Input / Output**
-Called peripherals
- Used to input and output instructions and data
- **Arithmetic and Logic Unit**
-Performs arithmetic operations (addition, subtraction)
- Performs logical operations (AND, OR, XOR, SHIFT, ROTATE)

Computer System Components

- **Control Unit**
- Coordinates the operation of the computer
- **System Interconnection and Interaction**
- ***Bus*** — A group of lines used to transfer bits between the microprocessor and other components of the computer system. Bus is used to communicate between parts of the computer. There is only one transmitter at a time and only the addressed device can respond.
— ***Types***
• Address
• Data
• Control signals

Computer Motherboard



Microprocessor

- A microprocessor is a silicon chip that contains a CPU.
- In the world of personal computers, the terms ***microprocessor*** and CPU are used interchangeably.
- At the heart of all personal computers and most workstations sits a microprocessor.
- Microprocessors also control the logic of almost all digital devices, from clock radios to fuel injection systems for automobiles.

Microprocessor

- It is a semiconductor device consisting of electronic logic circuits capable of performing computing functions
- Capable of transporting data/information
- It is a programmable device
- The programmer selects instruction from the list and determines the sequence of execution for a given task.
- It can be divided into 3 segments:

- Arithmetic and Logic Unit
- Register Unit
- Control Unit

CPU Components (Control Unit)

- Generates control signals which are necessary for execution of an instruction.
- Connect registers to the bus.
- Controls the data flow between CPU and peripherals (including memory).
- Provides status, control & timing signals required for the operation of memory and I/O devices to the system.

CPU Components (Registers)

- Hold data, instructions, or other items
- Various sizes
- Program counter and memory address registers must be of same size/width as address bus
- Registers which hold data must be of same size/width as memory words

CPU Components (Arithmetic Logic Unit)

- Executes arithmetic and logical operations.
- Accumulator is a register associated with ALU.
- Source of one of the operands of an arithmetic or logical operation.
- Serves as one input to ALU.
- Final result of an arithmetic or logical operation is placed in accumulator.

ALU (Functions)

ALU performs the following arithmetic and logical operations:

Arithmetic Operations

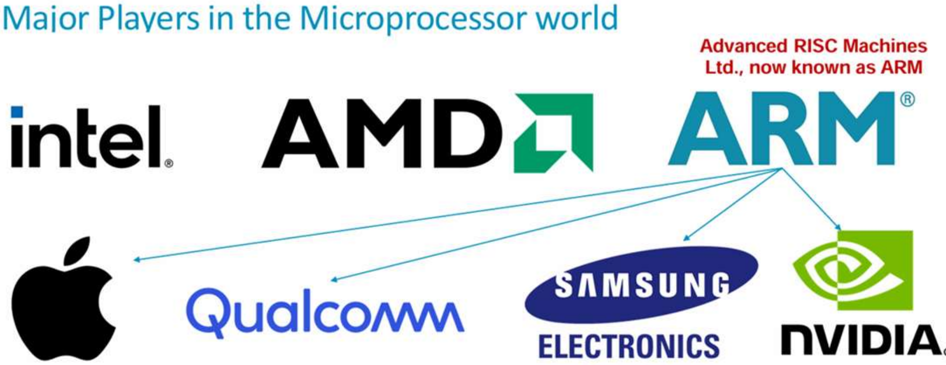
Addition
Subtraction
Increment (add 1)
Decrement (subtract 1)

Logical Operations

Logical AND, OR
Logical EXCLUSIVE OR
Complement(logical NOT)
Left shift, Rotate Left, Rotate right

Other Operations Clear etc.

Major Players in the Microprocessor world



Architecture

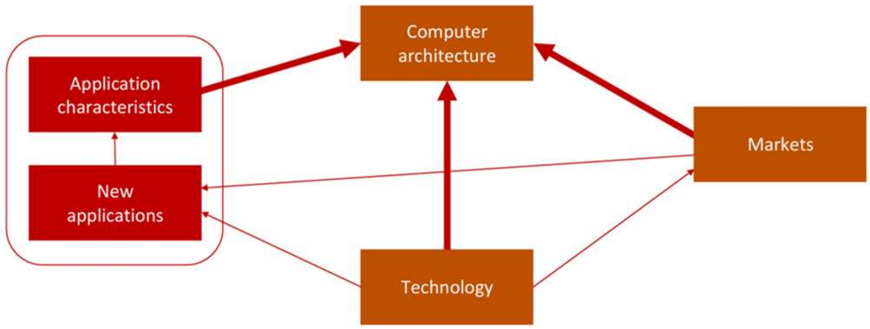
The Architecture is a contract between the hardware and the software.

- The hardware defines a set of operations, their semantics, and rules for their use.
- The software agrees to follow these rules.

Computer Architecture

- Computer architecture is concerned with how best to exploit fabrication technology to meet marketplace demands.
  - *e.g., how best might we use five billion transistors and a power budget of two watts to design the chip at the heart of a mobile phone?*
- Computer architecture builds on a few simple concepts, but is challenging as we must constantly seek new solutions.
- What constitutes the "best" design changes over time and depending on our use-case. It involves considering many different trade-offs.

Forces acting on Computer Architecture



Source: "Early 21st Century Processors," S. Vajapeyam and M. Valero, IEEE Computer, April 2004

Design Goals

- **Functional**– hard to correct (unlike software). **Verification** is perhaps the highest single cost in the design process. We also need to **test** our chips once they have been manufactured, again this can be a costly process and requires careful thought at the design stage
- **Performance**–what does this mean? No single best answer, e.g., sports car vs. off-road 4x4 vehicle performance will always depend on the "workload"; often attributed to **Speed**
- **Power** –a first-order design constraint for most designs today. Power limits the performance of most systems.
- **Security**– e.g., the ability to control access to sensitive data or prevent carefully crafted malicious inputs from hijacking control of the processor
- **Cost** – design cost (complexity), die costs (i.e., the size or area of our chip), packaging, etc.
- **Reliability**– do we need to try to detect and/or tolerate faults during operation?

Markets and Features

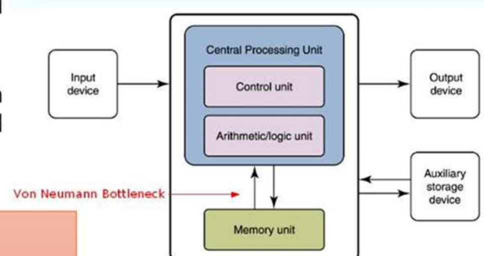
Each target market will require a different trade-off in terms of *power consumption, cost, area, performance, security, reliability,* etc.

Here are some example processor classes from Arm:

- **Cortex-A:** high-performance application processors, e.g., for mobile phones
- **Cortex-R**: deterministic real-time performance, fault detection, and tolerance.
- **Cortex-M**: energy-efficient embedded devices ("microcontroller" class cores)
- **Neoverse**: scalable networks of processors on a single chip
  - e.g., 8, 16, 64, or 128 cores. Used in datacenters, edge servers, and storage

Stored Program Computer/Concept
(Von Neumann Arch.)

- Instructions and data are stored together in memory (in binary form).
- The program is then fetched from memory an instruction at a time and executed



Questions...
- How are the program represented?
- How do we implement an algorithm in the computer?
- How does a computer interpret a program?

Representing Programs

- We need some basic building blocks -- call them "instructions"
- What does "execute a program" mean?
- What instructions do we need?
- What should instructions look like?
- Is it enough to just specify the instructions?
- How complex should an instruction be?

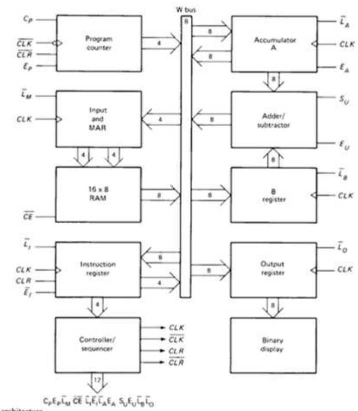
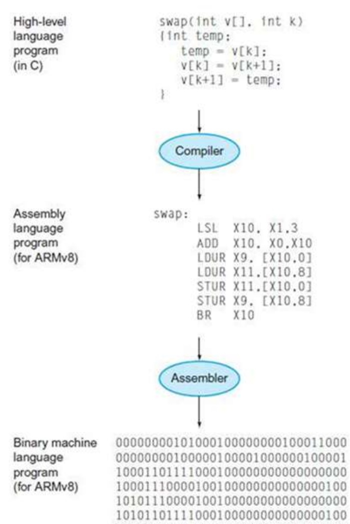


Fig. 10-1 SAP-1 architecture.

Levels of Program Code

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



Things to consider…

What instructions do we need?
- Depends on the application
    - Add, subtract, multiply, bitwise operations, branches, jumps, function calls, load, store, etc
- Trade-offs: Performance, power, efficiency, pipelining, etc

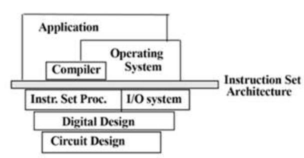
What instructions should look like?
- Binary codes (String of bits)
- All Same size or different sizes?
- What size? How many bits? 32 or 64 ?

The complexity of instructions
- Complex Instructions
    - Complex hardware
    - Difficult to pipeline
    - Different types of instructions required
    - Good code density
- Less complex instructions
    - Easier to pipeline
    - Poor code density
    - Less readable

"Instruction Set Architecture (ISA) or simply "Architecture"

- An ISA is "the agreed-upon interface between all the software that runs on the machine and the hardware that executes it."
- The "contract" between software and hardware
- Functional definition of operations, modes, and storage locations supported by hardware
- Precise description of how to invoke, and access them

- **Same** ISA or Architecture can be implemented by **different** microarchitecture, hardware designs



Instruction Set Architecture (ISA)	Type
x86	CISC
ARM	RISC
MIPS	RISC
RISC-V	RISC
PowerPC	RISC
IA-64 (Itanium)	VLIW
SPARC	RISC

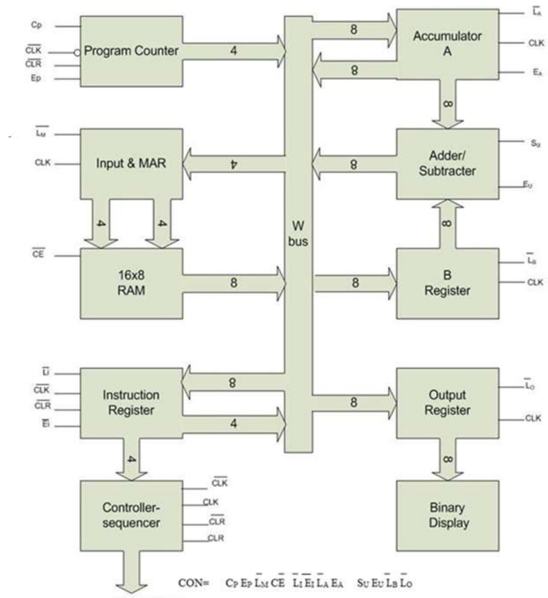
CISC: Complex instruction set computer
RISC: Reduced instruction set computer

SAP Generations

- There are *three* generations of Simple-As-Possible (SAP) computer.
- SAP-1 is the first stage towards the evolution of modern computers.
- Designed for the beginners.
- Main purpose is to introduce all the crucial ideas behind computer operation avoiding too much details.
- May seem primitive compared to modern computers but SAP-1 is a giant step for a beginner.

SAP-1 Architecture

- Bus organized architecture.
- All register outputs to the W bus are three-state which allows orderly transfer of data.
- All other register outputs are two-state.



SAP-1 Architecture

Program Counter

- A part of control unit.
- 4-bit up counter, counts from 0000 to 1111.
- The program for the computer is stored at the beginning of the memory with the first instruction at binary address 0000.
- The job of the program counter is to send to the memory address of the next instruction to be fetched and executed.

SAP-1 Architecture

Input and MAR

- Contains address and data switch registers.
- Address and data switch registers are part of input unit.
- These registers allow programmer to send 4-bit address and 8-bit data bits to RAM.
- MAR (Memory Address Register) is a part of memory unit.
- During a computer run, the addresses of program counter are latched into it.
- These latched addresses are fed to RAM when a read operation is performed.

SAP-1 Architecture

RAM

- The RAM is a 16 x 8 static TTL RAM.
- Can be programmed by means of address and data switches.
- Instruction and data words are written into the RAM before a computer run.
- During a computer run, the RAM receives 4-bit addresses from the MAR and a read operation is performed.
- The instruction or data word stored in the RAM is placed on the W bus for use in some other part of the computer

SAP-1 Architecture

Instruction Register

- Part of control unit.
- To fetch an instruction from the memory the computer does a memory read operation.
- This places the contents of the addressed memory location on to W bus.
- At the same time the IR is set up for load on the next positive clock edge.
- The contents of IR are divided into two nibbles.
- The upper nibble is two state and goes to controller/sequence.
- The lower nibble is three state output that is read onto the W bus when needed.

SAP-1 Architecture

Controller-Sequencer

- Before each computer run, C/S sends clear signal to reset program counter and IR.
- Sends clock signal to all buffer registers to synchronize the operation of the computer.
- The output of C/S is a 12-bit word known as Control Word (CON) contains signals those controls the rest of the computer.
- These 12-bit signal constitutes what is called control bus.

$$\mathbf{CON} = C_P E_P \overline{L}_M \overline{CE} \quad \overline{L}_I \overline{E}_I \overline{L}_A E_A \quad S_U E_U \overline{L}_B \overline{L}_O$$

- This word determines how the registers will react to the next positive CLK edge.

SAP-1 Architecture

Accumulator

- The accumulator (A) is a buffer register that stores intermediate answers during a computer run.
- The accumulator has two outputs.

✓The two-state output goes directly to the adder/subtracter.

✓The three-state output goes to the W bus.

- 8-bit accumulator word continuously drives adder/subtracter

SAP-1 Architecture

Adder-Subtracter

- 2's complement adder/subtracter.
- Performs

✓Addition when $S_u = 0$.

✓Subtraction when $S_U = 1$.

- Asynchronous - output changes as soon as the inputs change.
- Output appears on the W bus when E_U is high.

SAP-1 Architecture

B Register

- It is another buffer register.
- Used in arithmetic operations.
- A low L_B and positive clock edge load the word on the W bus into the B register.
- The two-state output of the B register drives the adder-subtracter, supplying the number to be added or subtracted from the contents of the accumulator.

SAP-1 Architecture

Output Register

- At the end of a computer run, accumulator contains the result.
- The output register is used to transfer the result from accumulator to the outside world.
- When E_A is high and L_0 is low, the next positive clock edge loads the accumulator word into the output register.
- Also called Output Port since the processed data leave the computer through this register.
- In microcomputers the output port is connected to interface circuits that drive peripheral devices like printers, cathode-ray tubes, teletypewriters and so on.

SAP-1 Architecture

Binary Display

- The binary display is a row of eight light-emitting diodes (LEDs).
- Because each LED connects to one flip-flop of the output port, the binary display shows us the contents of the output port.