Do Now

Comment in the chat

- Write down the internal components of a computer you know?

You have 1 minute

Isaac Computer Science

Systems Architecture

2024

Learning Outcomes

By the end of this 90-minute session, you will:

- Understand and differentiate between Harvard, Von Neumann, and contemporary CPU architectures.
- Identify and explain the functions of key internal components of a computer system
- Examine and explain how various hardware components and design features impact the performance of computer systems.

Confidence Check-In

- Please take time to anonymously share your confidence in today's topics

Isaac Computer Science

Confidence check-in form

CPU Architecture

Image generated by OpenAI's DALL-E

In the chat

What characteristics describe a computer?

You have 1 minute

The Antikythera Mechanism

The oldest known computer was submerged for over 2,000 years near the Greek Island of Antikythera.

The device could perform calculations (compute) to:

- predict lunar and solar eclipses.
- track the movements of the Sun, Moon, and known planets.
- calculate the dates of important events, such as Olympic Games.

Computer architecture

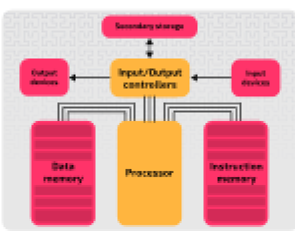
The Harvard Mark I was an early electromechanical computer. Designed by Howard Aiken and built by IBM for Harvard University in 1944, its first programs were written by John von Neumann ("*von NOY-mən*") for the Manhattan "atomic bomb" project.

The Mark I featured separate memories and pathways for data and instructions.

This separation allows the computer to fetch instructions and data simultaneously

Harvard architecture

The processor accesses main memory using dedicated connections called **buses**. The Harvard design has **separate** buses for instructions and data:



The Harvard architecture is inflexible hence not suited to general purpose computers.

It is used extensively in embedded systems.

Computer architecture

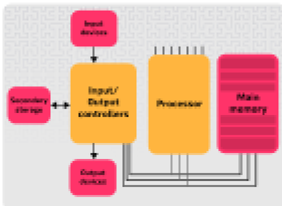
The EDVAC computer was designed by John von Neumann in 1945 and has influenced all subsequent computer designs. EDVAC pioneered the "stored-program" concept that we now call the **Von Neumann architecture**.

This design uses a single memory store for both instructions and data.

This architecture is much more versatile, can be implemented with simpler hardware and software, and allows programs to dynamically allocate memory at runtime.

Von Neumann architecture

The von Neumann architecture stores the program instructions and data in the same shared main memory. The instructions and data **share the same buses**



Von Neumann architecture is flexible and versatile so it is suitable for general purpose computers.

Handout 1 – Von Neumann vs Harvard architecture

Arrange the descriptions provided for each architecture

Feature	Harvard Architecture	Von Neumann Architecture
Memory Use		
Customisation		
Speed		
Flexibility		
Where It's Used		

You have 5 minutes

Handout 1 – Von Neumann vs Harvard architecture – answers

Feature	Harvard Architecture	Von Neumann Architecture
Memory Use	Has two sets of memory and buses: one for instructions and another for data.	Uses the same memory and buses for both instructions and data
Customisation	Each part of the memory can be tailored to the system's needs	Memory is more general-purpose, allowing for a wide range of programs.
Speed	Generally faster, especially for tasks that need to handle lots of data quickly,	Can be slower due to limitations in how it handles memory.
Flexibility	Instruction memory is often fixed and unchangeable, good for security.	Very flexible, as the same memory is used for everything.
Where It's Used	In gadgets designed for specific jobs, like smart thermostats or car brake systems.	In PCs, laptops, and servers that run different kinds of software for various tasks.

Contemporary architectures

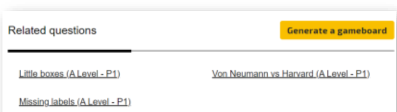
Both Von Neumann and Harvard architectures were introduced in the 1940's. Many modern architectures are known as 'Contemporary' architectures. They include elements of both to allow for better overall performance, such as:

- Multitasking
- Hardware Multithreading
- Hardware Acceleration
- Pipelining
- Energy Efficiency Features
- Integrated GPU

Activity – Isaac Questions

Open the Isaac Computer Science section here
Complete the **related** questions

bit.ly/iaarch1



You have 3 minutes!

"I have known a great many intelligent people in my life. I will say this: I have never met a man who was his equal. He was the cleverest man I have ever known."

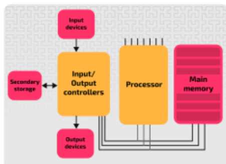
Albert Einstein on John von Neumann

Internal computer components

The internal components of a computer system consist of the hardware required to store and process data, and communicate with external, peripheral devices.

The main internal components of a computer system are:

- Processor
- Main memory
- Input/output controllers

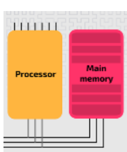


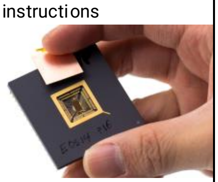
With a partner
How were their roles explained at GCSE level?

Internal computer components

The processor

The processor, sometimes referred to as the CPU (central processing unit), is the part of the computer that processes data by executing program instructions

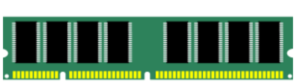




Program instructions and data are moved from secondary storage to main memory for the CPU to execute them.

Primary Memory

The 2 types of primary memory are **RAM** and **ROM**
Each location has a **physical address**, which is a number used by the CPU, to locate that memory location and access its contents.



RAM

Stored in RAM (Random Access Memory)

- currently used parts of the operating system
- currently used instructions and data

RAM is volatile
Comment in the chat, what does this term mean?

Primary Memory

The other type of main memory is **ROM** (Read Only Memory) A microchip (EEPROM) located on the motherboard stores the Basic Input Output System (BIOS), it is read only.

Image generated by OpenAI's DALL-E

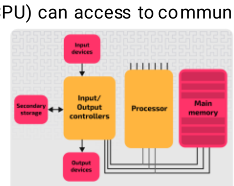
Stored in ROM
- BIOS
- bootloader program
- system firmware
- code to handle I/O operations

ROM is non-volatile
This is because the contents of the BIOS are not lost when the power is turned off

Input/output (I/O) controllers

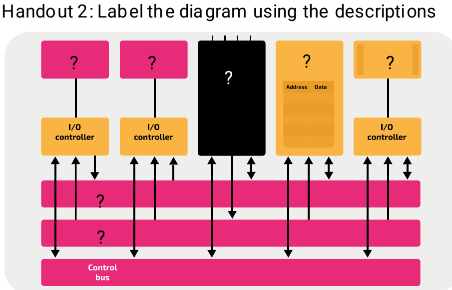
Peripherals like keyboards and printers connect to the computer's processor through. The **processor can't work directly with these**, so they are accessed through I/O controllers.

I/O controllers provide a set of addressable registers that the processor (CPU) can access to communicate with the I/O devices



A model of the system bus

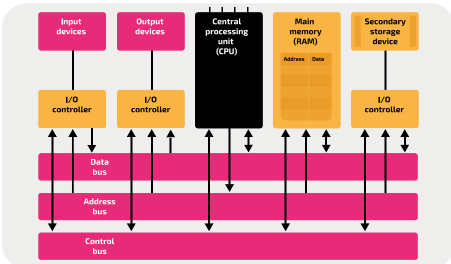
Handout 2: Label the diagram using the descriptions



CPU
RAM
Output devices
Input devices
Address bus
Data bus
Secondary Storage

A model of the system bus

Answers



A model of the system bus

Address bus

The address bus carries memory addresses, specifying where data should be read from or written to in the memory.

Data bus

The data bus transfers actual data between the processor, memory, and peripherals within a computer system.

Control bus

The control bus sends read/write signals to manage and synchronise operations among a computer's components.

Think back!

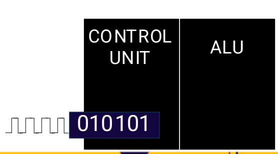
- Are these 1 way or 2 way?

The Processor

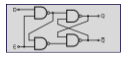
The **Arithmetic Logic Unit** (ALU) is responsible for performing arithmetic calculations and logical operations.

The **control unit** (or program control unit) controls the flow of data around the CPU. It does this using read and write signals.

The **system clock** generates regular clock pulses by emitting a signal that continuously oscillates between a low (or '0') and a high (or '1') state.



Registers



Registers are **small** locations of computer memory within the processor that provide extremely **fast temporary** access.

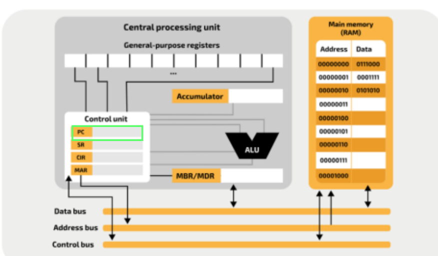
They are **part of the CPU's core architecture** and are used to hold data and addresses temporarily that are needed during instruction execution.

General-purpose registers

- temporarily store and access the results of operations
- help transfer data within the CPU and to external components.
- hold operands, so CPU can quickly fetch them when needed
- support Addressing Modes (immediate/direct)

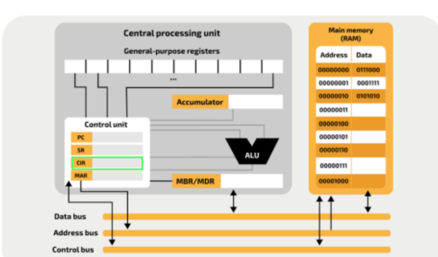
Special-purpose registers

Program Counter (PC):
Contains the address of the **next** instruction to be fetched from memory.



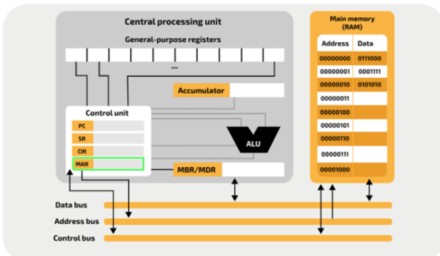
Special-purpose registers

Current Instruction Register (CIR):
Holds the instruction currently being executed.



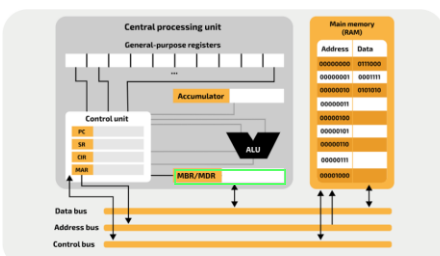
Special-purpose registers

Memory Address Register (MAR):
Holds the address of the instruction being fetched from memory **or** the address of data being read or written to memory



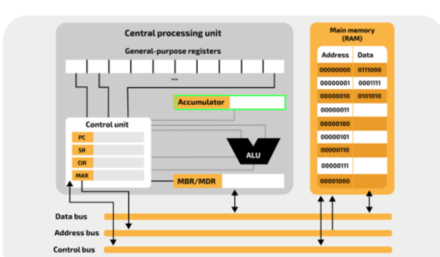
Special-purpose registers

Memory Data Register (MDR):
Holds the instruction fetched from memory
or the data being read or written to memory



Special-purpose registers

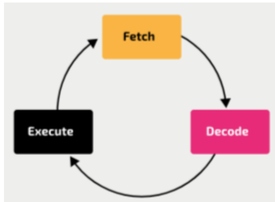
Accumulator:
Holds the results of calculations from the ALU



The fetch, decode, execute cycle

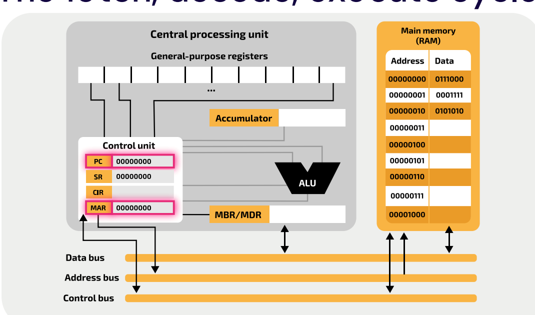
Every time the processor carries out an instruction, it goes through three main stages: it **fetches**, **decodes**, and **executes** the instruction.

Registers play a crucial role

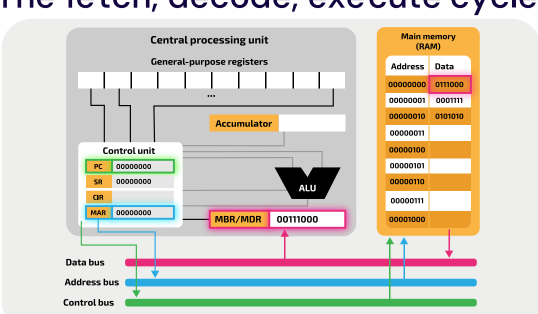


Let's see it happening!

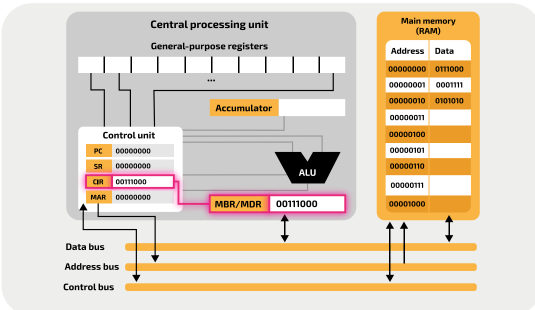
The fetch, decode, execute cycle



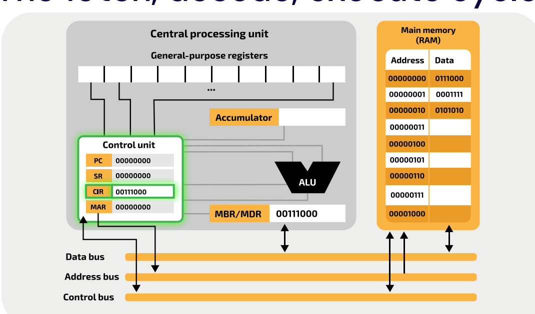
The fetch, decode, execute cycle



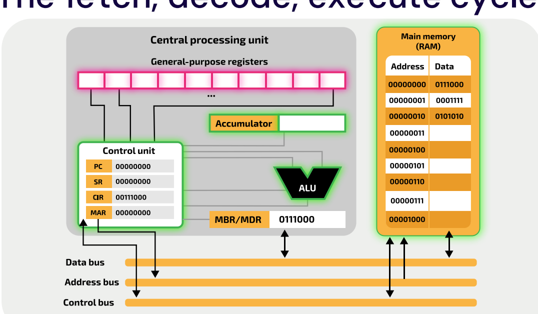
The fetch, decode, execute cycle



The fetch, decode, execute cycle



The fetch, decode, execute cycle



Systems Architecture

Factors affecting processor performance

- Cores
- Bus Width
- Pipelining
- Cache size
- Clock Speed
- Word length

Clock speed

- Processors have a clock speed. The clock produces a signal to synchronise the operation of the processor.
- This clock signal is either located on the motherboard's or integrated within the CPU itself.
- In general terms, the greater the clock speed, the faster the instructions are carried out.
- Clock speed is usually measured in GHz (gigahertz).
- 1Hz means the processor can process 1 instruction per second, so **1GHz** means **1 billion** instructions per second.

Cache memory

When running programs, a computer system is constantly swapping data in and out of cache and RAM.

- Cache stores frequently/commonly used instructions & data.
- The cache provides fast access compared to RAM

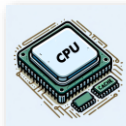


Image generated by OpenAI's DALL-E

Levels of Cache
Level 1: part of the circuitry of each core
Level 2: often shared by cores
Level 3: sits on the processor or nearby on the motherboard.
Level 4 : (newer systems) more cache on the motherboard

Number of cores

A core is an individual processor within the CPU that can work on a different task separately from other cores.

Components Common to Both a CPU and Its Cores:
- Arithmetic Logic Unit (ALU)
- Control Unit (CU)
- Registers
- L1 Cache

Other performance factors

Word Length 0 1 1 1 0 1 0 0 1 1 0 0 1 1 0 1

The number of bits that a CPU can work with is its **word length,** typically 32 or 64 bits. Word length describes the size of the **registers** and width of the **buses** and is an upper limit on the amount of data processed in one clock cycle.

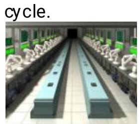


Image generated by OpenAI's DALL-E

Pipelining
Instruction pipelining divides execution into simultaneous steps. For instance, during the FDE cycle, one instruction is **fetched** while another is **decoded,** and another is **executed**.

Pipelining works with RISC due to its simpler instructions

The system bus

The width of the address bus refers to its number of parallel lines, which determines the number of bits that can be used to form an address of a memory location, which imposes a limit on the amount of addressable memory.

Bus width

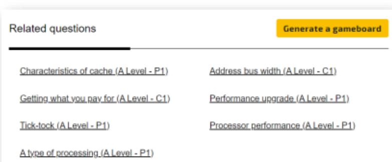
Data bus
Address bus
Control bus

Why do you think that increasing the width of these buses could improve CPU performance?

Activity – Isaac Questions

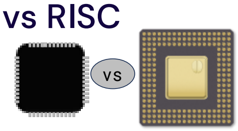
Open the Isaac Computer Science section here
Complete the **related** questions

bit.ly/iaarch2



You have 5 minutes!

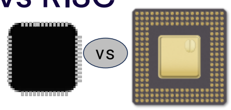
CISC vs RISC



Complex **I**nstruction **S**et **C**omputers

- Instruction set of many complex, variable-length instructions
- Instructions can take several clock cycles to be executed…
  - …reducing the effectiveness of pipelining.
- Compound addressing modes
- Emphasises hardware over software: the ISA does as much as possible using processor hardware circuitry
- Makes efficient use of RAM
- Used in desktop computers and servers

CISC vs RISC



Reduced **I**nstruction **S**et **C**omputers

- Instruction set of a small number of simple, fixed-length instructions
- One instruction is executed per clock cycle…
  - …can thus perform pipelining better than CISC
- Limited addressing modes
- Emphasises software over hardware: high-level compilers take on most of the burden of translating programs
- Uses a lot RAM, which can cause system bottlenecks
- Used in portable devices e.g. smartphones, tablets, wearables (some apple desktops use RISC)

Confidence Check-In

- Please take time to anonymously share your new confidence in today's topics

Isaac Computer Science

Confidence check-in form

Learning Outcomes

By the end of this 90-minute session, you will:

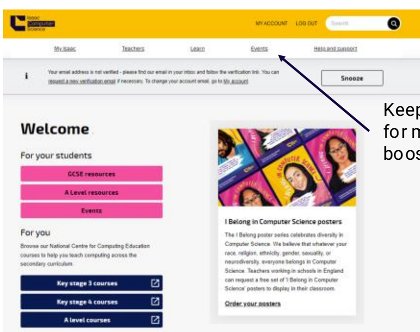
- Understand and differentiate between Harvard, Von Neumann, and contemporary CPU architectures.
- Identify and explain the functions of key internal components of a computer system
- Examine and explain how various hardware components and design features impact the performance of computer systems.

Isaac Gameboard practice

- If you want more systems architecture practice, then try this gameboard.
- You will need to sign in to **Isaac Computer Science** or register for a free account if not done already.

ncce.io/isc-Asysarch

Check for more ISAAC boosters



Keep an eye out for more student booster events

Thank you

Isaac Computer Science

STEM LEARNING