ANDIJAN STATE PEDAGOGICAL INSTITUTE

Lecture materials · 2026–2027 academic year

SUBJECT

Information Systems and Digital Technologies

Lecture 4 · Understanding Computer Technology

Part 2 of 2 · 2 hours

Lecturer: **Ismailov Alisher**

E-mail: alisherismailov1991@gmail.com

Contact: **99-499-73-64**

LECTURE 04 · 2 HOURS · PART 2/2

2

CHAPTER TWO

Fundamentals of Computers

Computer Organisation and Architecture

Inside the processor · how it talks to memory and devices · what an instruction really is

- The CPU and its units
- Memory & registers
- Machine cycle & buses
- CISC and RISC

LEARNING OBJECTIVES

By the end of this lesson you will be able to…

4 Sequence the four phases of the machine cycle and the steps of a memory read and a memory write.

6 Compare CISC and RISC processors on instruction count, format, execution and hardware design.

7 Explain how the CPU, memory and storage work together to process and store information.

8 Describe the von Neumann and Harvard designs and how early computers shaped today’s machines.

04

The Machine Cycle and the Bus

What happens for a single instruction — and the wires that carry everything

- Four phases
- Instruction cycle
- Execution cycle
- Data · address · control

FIG. 2.10

The machine cycle — one instruction, four phases

The cycle during which a machine language instruction is executed by the processor is known as the machine cycle. A program of 10 machine language instructions runs 10 separate machine cycles.

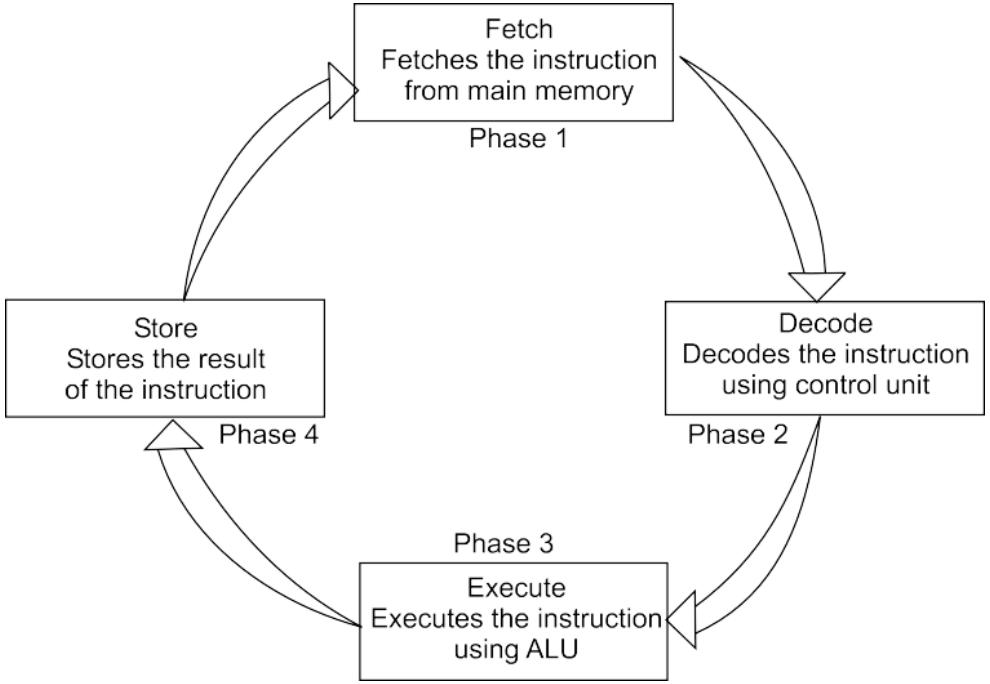


Fig. 2.10 The four phases of the machine cycle

The four phases group into two cycles

Instruction cycle

Fetching | **Decoding**

Getting the instruction and working out what it means.

Execution cycle

Executing | **Storing**

Carrying the instruction out and putting the result away.

After the execution cycle is completed, the next machine cycle begins.

Instruction cycle — fetching and decoding

1 Fetching

The CPU retrieves the instruction from the main memory of the computer system.

Address arrives
The address of the instruction is sent to the CPU through the address bus.

PC stores it
The CPU stores that address in its internal register, the Program Counter (PC).

IR holds the instruction
Once the address is confirmed, the instruction is retrieved from main memory and stored in the Instruction Register (IR).

2 Decoding

The instruction is broken down into different parts so that it can be easily understood before being processed by the CPU.

Who does it
The instruction decoder, a vital component of the CPU.

Another name
Decoding an instruction is also known as interpreting.

What it finds
Two key attributes — the opcode (the operation) and the operands (the data). The data goes to the Data Register (DR).

Execution cycle — executing and storing

The execution cycle executes the instruction and stores the result back to the main memory, or sends it to an output device of the computer system.

3 Executing

The decoded instruction is executed by the ALU of the CPU.

Timing varies
The execution time spent by the ALU may vary depending on the type of instruction.

And on the design
The execution time also depends on the processor architecture.

4 Storing

The result computed in the execution phase is sent onwards.

Where it goes
Either to the memory or to an output device of the computer system.

And one more thing
The PC of the CPU is updated in this phase to point to the next instruction to be executed.

Updating the PC in the storing phase is what makes the next machine cycle possible.

COMPARISON

Instruction cycle and execution cycle

	Instruction cycle	Execution cycle
Phases included	Fetching and decoding	Executing and storing
What happens	The instruction is retrieved from main memory and broken down into its parts	The instruction is carried out and the result is put away
Registers involved	Program Counter (PC), Instruction Register (IR), Data Register (DR)	The PC is updated to point to the next instruction
Which part of the CPU	The instruction decoder does the decoding	The ALU does the executing
Ends with	An instruction that the CPU can understand	A result in memory or at an output device

CHECK FOR UNDERSTANDING

Which phase? · say the name

Which phase of the machine cycle does each belong to?

1

The instruction is stored in the Instruction Register.

Event 1

2

The instruction is broken into opcode and operands.

Event 2

3

The ALU carries out the operation.

Event 3

4

The Program Counter is updated.

Event 4

Answers: 1 — fetching. 2 — decoding. 3 — executing. 4 — storing. Follow-up: which two of these make up the instruction cycle? (1 and 2.)

A bus is a set of wires

A set of wires used to connect the different internal components of the computer system, for the purpose of transferring data as well as addresses amongst them. There may be several buses in a computer system.

Serial bus

Only one bit of data is transferred at a time amongst the various hardware components.

1 bit at a time

Parallel bus

Several bits of data can be transferred at a time amongst the various hardware components.

several bits at a time

How bus speed is measured: the number of bits transferred per second between two components.

FIGS. 2.12 AND 2.13

The data bus and the address bus

Data bus

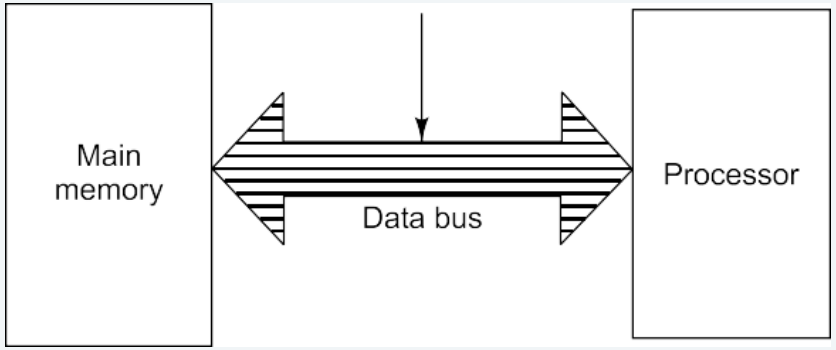


Fig. 2.12 The data bus between main memory and the processor

Carries Data amongst the different internal components.

Direction Bidirectional — it allows transfer in both directions, and is bidirectional in most computer systems.

Width Modern computer systems use 32-bit data buses, transferring 32 bits at a time. Its speed affects the overall processing power.

Address bus

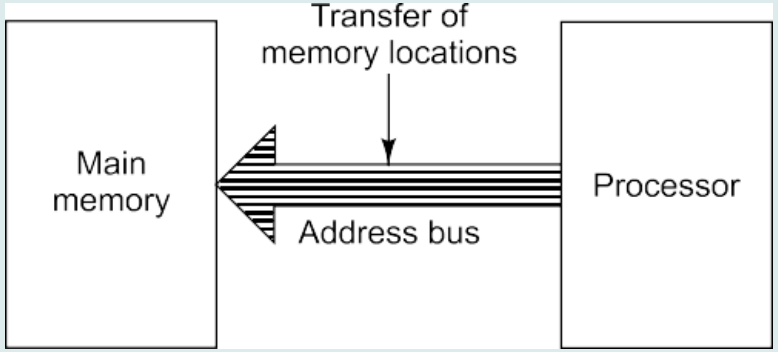


Fig. 2.13 The address bus between processor and memory

Also known as The memory bus.

Carries The memory addresses for read and write memory operations.

Address lines The number of address lines determines the range of addresses. A 32-bit address bus can reference 2^{32} memory locations.

Direction Unidirectional between main memory and processor — but bidirectional between the processor and the I/O system.

FIG. 2.11

Three buses hold the system together

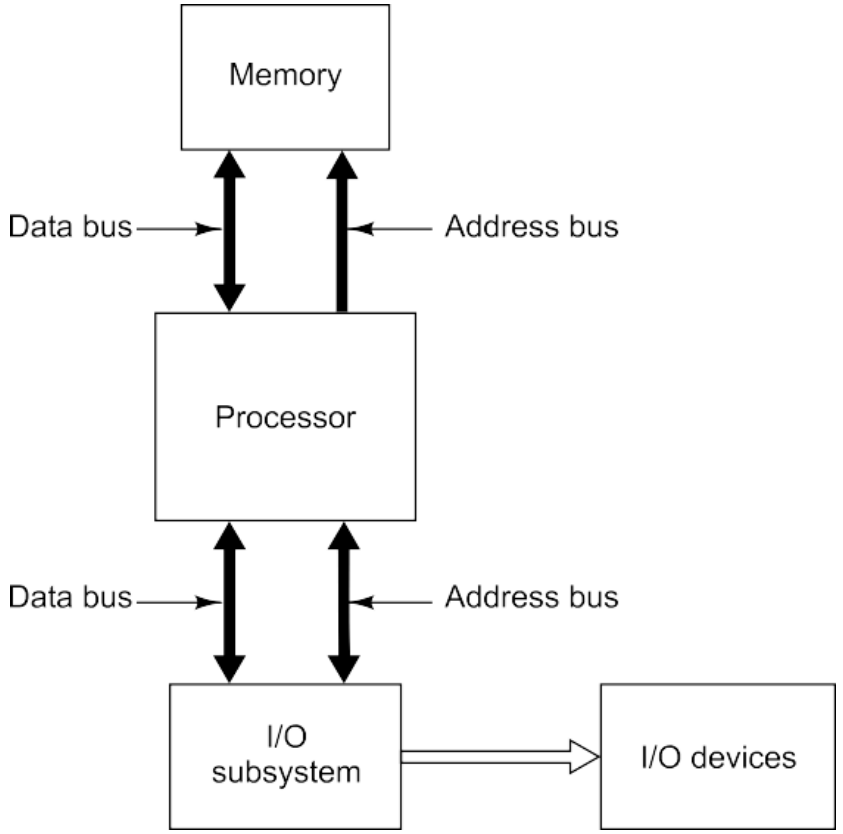


Fig. 2.11 Data and address buses between memory, processor and the I/O subsystem

Data bus

Transfers data amongst the different internal components. Bidirectional in most systems; 32 bits at a time in modern machines.

Address bus

Transfers memory addresses for read and write operations. Also known as the memory bus.

Control bus

Manages the transfer of data and addresses among the various components by transferring the appropriate control signals.

You have already met all three at work — in the memory read, the memory write and both I/O operations.

CHECK FOR UNDERSTANDING

Which bus? · call it out

Which bus would carry each of these?

1

The location in memory that the processor wants to read from.

Traffic 1

2

The READ command issued by the processor.

Traffic 2

3

The value 42 travelling back from memory to the processor.

Traffic 3

4

A status signal generated by a device in response to a command.

Traffic 4

Answers: 1 — the address bus. 2 — the control bus. 3 — the data bus. 4 — the control bus, which carries both commands and status signals.

05

Instruction Sets

What a processor can be asked to do — and the two schools of thought about how many things that should be

- What a set is
- Complex — CISC
- Reduced — RISC
- Head to head

A group of instructions a processor can execute

An instruction set can be defined as a group of instructions that a processor can execute to perform different operations.

On the basis of complexity and the number of instructions used, the instruction set can be classified as:

Complex instruction set

100 – 250

instructions

CISC — Complex Instruction Set Computer

Reduced instruction set

0 – 100

instructions

RISC — Reduced Instruction Set Computer

COMPLEX INSTRUCTION SET

CISC — many instructions, complicated hardware

Many instructions

From 100 to 250 instructions in the set.

Memory-based

Mostly memory-based instructions, involving frequent references to the memory.

Many addressing modes

Because of frequent references to both registers and memory.

Variable length format

The instruction format is not limited to 32 bits.

Execution takes a lot of time, because the instructions are memory-based and accessing memory is slower than accessing registers. Each instruction type needs separate circuitry, which makes the CPU design more complicated.

ADVANTAGES

- No need to invent an instruction set for each new design — a new processor can use its predecessor's set
- A program written in CISC requires less memory space, as the code is confined to fewer instructions
- CISC makes the job of a compiler easier by helping to implement high-level language constructs

DISADVANTAGES

- Inheriting old instructions into new processors increases complexity
- Many CISC instructions are not frequently used
- CISC commands translate into a large number of lines of microcode, which makes processing slower
- CISC systems have complex hardware, so they require more time for designing

REDUCED INSTRUCTION SET

RISC — few instructions, simple hardware

Few instructions

From 0 to 100 — only those frequently used by the processor, and generally very simple to execute.

Register-based

Execution involves frequent references to the registers. Memory-based instructions are very few: only load and store.

Fixed length format

Every instruction has a fixed length format of 32 bits.

An instruction format divides the bits of an instruction into small groups called fields

1 Opcode field

Represents the operation to be performed by the instruction.

2 Operand field

Represents the data on which the operation is to be performed, or the memory location or register where the data is stored.

3 Mode field

Represents the method of fetching the operands stored at the specified memory location or registers.

Speed is measured in MIPS — Millions of Instructions Per Second.

The only disadvantage: the number of instructions required to perform an operation is comparatively large.

HEAD TO HEAD

RISC and CISC compared

	CISC	RISC
Number of instructions	100 to 250	0 to 100
Instructions mostly	Memory-based	Register-based; only load and store touch memory
Instruction format	Variable length, not limited to 32 bits	Fixed length of 32 bits
How instructions run	First translated into equivalent microcode instructions	Executed by decoding
Clock cycles per instruction	Multiple clock cycles	A single clock cycle
Hardware	Complex, difficult to design and large in size	Very simple and easily designed

The chapter's verdict: RISC processors are always preferred over CISC processors, because of their compact size and small instruction set.

CHECK FOR UNDERSTANDING

CISC or RISC? · write C or R

Which kind of processor is being described?

1

Instructions have a fixed length format of 32 bits.

Clue 1

2

The instruction set contains between 100 and 250 instructions.

Clue 2

3

Instructions are executed in a single clock cycle.

Clue 3

4

Commands are translated into a large number of lines of microcode.

Clue 4

Answers: 1 — RISC. 2 — CISC. 3 — RISC. 4 — CISC. Follow-up: name the one disadvantage of RISC compared with CISC.

06

Working Together and Early Designs

CPU, memory and storage as one system · the stored-program idea · von Neumann and Harvard designs

- Memory vs storage
- Working together
- Von Neumann
- Harvard and today

CPU, memory and storage: three different jobs

	CPU	Main memory (RAM)	Storage (SSD / HDD)
Job	Processes data: calculates and decides	Holds the programs and data in use now	Keeps files and programs for later
Speed	Fastest (registers, cache)	Fast	Slower
Typical size	Bytes in registers; MB of cache	8 – 32 GB	256 GB – several TB
Power off	Contents lost	Contents lost (volatile)	Contents kept (non-volatile)
Kitchen analogy	The cook	The worktop	The cupboard

The CPU can work only on what is in memory — anything on storage must first be copied into RAM.

Opening, editing and saving a document

1 — Request

You double-click a file; the mouse sends input and the OS takes the request.

→

2 — Load

The program and the file are copied from storage into RAM.

→

3 — Fetch

The CPU fetches instructions from RAM over the buses, through the cache.

→

4 — Execute

The control unit and ALU do the work; registers hold the values; results go to RAM.

→

5 — Output

The screen shows the document and every change you make.

→

6 — Save

The file is written from RAM back to storage — now it is permanent.

Try it: if the power fails before step 6, your changes are lost — they were only in RAM. That is why programs auto-save.

The stored-program idea: von Neumann architecture

The idea (1945)

1. **Before:** ENIAC was programmed by re-plugging cables and setting switches — days of work.
2. **Stored program:** instructions are kept in memory, just like data.
3. **Five parts:** input, output, memory, ALU and control unit.
4. **One at a time:** fetch, decode, execute — the machine cycle.

The von Neumann bottleneck

1. **One path:** instructions and data share one memory and one bus.
2. **The problem:** the fast CPU often waits for the slower memory.
3. **Today's fixes:** cache, wider and faster buses, pipelining, many cores.
4. **Still used:** almost every PC, laptop and phone follows this model.

EARLY DESIGNS

Von Neumann vs Harvard architecture

	Von Neumann	Harvard
Origin	EDVAC report, 1945	Harvard Mark I, 1944 (program on punched tape)
Memory	One memory for instructions and data	Separate memories for instructions and data
Buses	One shared set	Separate sets — both fetched at the same time
Strength	Simple and flexible	Faster: no bottleneck between code and data
Used today in	PCs, laptops, phones, servers	Microcontrollers, signal processors; split L1 cache in CPUs

Modern CPUs mix both: von Neumann main memory, but separate L1 caches for instructions and data (“modified Harvard”).

How early designs shaped today’s computers

1944 — Harvard Mark I

Program on tape, data separate → the Harvard design in microcontrollers and CPU caches.

1945 — ENIAC

Electronic speed, but programmed by rewiring → showed that programs must be stored.

1945 — EDVAC report

Program and data in one memory → the model of almost every computer today.

1949 — EDSAC

One of the first stored-program computers in regular use → the start of software libraries.

1971 — Intel 4004

The whole CPU on one chip → microprocessors in PCs, phones and cars.

NOW — Today

Multicore CPUs, caches, SSDs — the same five parts, billions of times faster.

Why it matters: the ideas of the 1940s still explain how your phone works — and why memory speed limits it.

CHECK FOR UNDERSTANDING

Name it · write your answers down

Which part or design is being described?

1

It holds the program you are using now, but forgets everything when the power goes off.

Clue 1

2

Instructions and data share one memory and one bus.

Clue 2

3

Separate memories and buses for instructions and data.

Clue 3

4

The fast CPU keeps waiting because memory cannot deliver data quickly enough.

Clue 4

Answers: 1 — main memory (RAM). 2 — von Neumann architecture. 3 — Harvard architecture. 4 — the von Neumann bottleneck.

REAL-WORLD CONNECTIONS

The chapter's idea, and where you meet it today

Idea from this chapter	What the chapter says	Where you meet it today
Cache memory	Small, fast, expensive; checked before the main memory	*The “L1 / L2 cache” line in a laptop's specifications*
Main memory (RAM)	Temporary, limited, erased when the power goes off	*Unsaved work lost when a phone battery dies*
Swapping	Instructions copied from secondary storage into RAM before execution	*An app pausing while it loads from storage*
Machine cycle	Fetch, decode, execute, store — once per instruction	*A processor's clock speed: how many cycles per second*
Bus width	32-bit data buses transfer 32 bits at a time	*“64-bit” processors described in device specifications*
RISC	Few, simple, register-based instructions; single clock cycle	*The processor architecture used in most phones and tablets*
Instruction set	The group of instructions a processor can execute	*Why an app built for one processor family will not run on another*

OPTIONAL CLARIFICATION — NOT EXAMINABLE

Where the textbook and modern sources differ

For all tests and exercises, use the chapter as written. The notes below are background only, so that nobody is confused by another source.

MBR and MDR

The chapter introduces processor-to-memory communication as using MAR and MBR, then uses MDR in the figure and in all the steps. Most modern sources treat MBR and MDR as two names for the same register.

→ ***Use whichever name the question uses.***

“32-bit data buses”

The chapter says modern computer systems use 32-bit data buses. Written before 64-bit machines became the norm for desktops and phones.

→ ***Answer 32 bits, as the chapter does.***

RISC “always preferred”

The chapter concludes RISC processors are always preferred over CISC. In practice the two approaches have converged, and CISC-descended processors remain common in desktops and servers.

→ ***Answer as the chapter does.***

Levels of cache

The chapter describes two levels, L1 and L2. Many processors today also have an L3 cache, and the chapter's own multiple choice lists L3 and L4 only as wrong options.

→ ***Two levels, for this chapter.***

“16 logical operations”

The chapter states the Logic Unit performs 16 different types of logical operation and gives six examples.

→ ***Learn the number and the examples given.***

ALU, AU and LU

The chapter names the Arithmetic Unit and Logic Unit separately, then refers to “the ALU” in the executing phase. They are the same hardware described at two levels of detail.

→ ***Both terms are correct.***

REVIEW

Eight questions before we finish

1. Which unit of the CPU performs all the arithmetic operations?
2. Which register keeps track of the next instruction to be executed?
3. Which memory does the CPU check first when searching for data?
4. Which cache is also known as the internal cache, and where is it located?
5. Name the four phases of the machine cycle, in order.
6. Which two registers implement direct processor-to-memory communication?
7. What is the fixed instruction length used by RISC processors?
8. Name the three types of bus found in a computer system.

REVIEW

Answers

1 **The Arithmetic Unit**

Addition, subtraction, multiplication and division.

2 **The Program Counter (PC)**

The Control Unit uses it to retrieve the next instruction.

3 **Cache memory**

It is checked before reading from or writing to the main memory.

4 **Primary cache — L1**

Located inside the CPU. Smaller but the fastest type of cache.

5 **Fetch, decode, execute, store**

Fetch and decode form the instruction cycle; execute and store the execution cycle.

6 **MAR and MBR**

MAR holds the address; the data register — MBR, shown as MDR in the figures — holds the data.

7 **32 bits**

RISC instructions have a fixed length format of 32 bits.

8 **Data bus, address bus, control bus**

The control bus carries commands and status signals.

EXIT TICKET

Answer these three before you leave

Write your answers on a slip of paper with your name and hand it in at the door.

1

In one sentence each, say what computer architecture is and what computer organisation is.

Tests objective 1

2

List the four phases of the machine cycle in order. Which two form the instruction cycle?

Tests objective 4

3

Give two differences between a RISC processor and a CISC processor.

Tests objective 6

Next: five interactive activities — CPU components, the machine cycle sequencer, registers and memory, identify the component, and the final knowledge challenge.