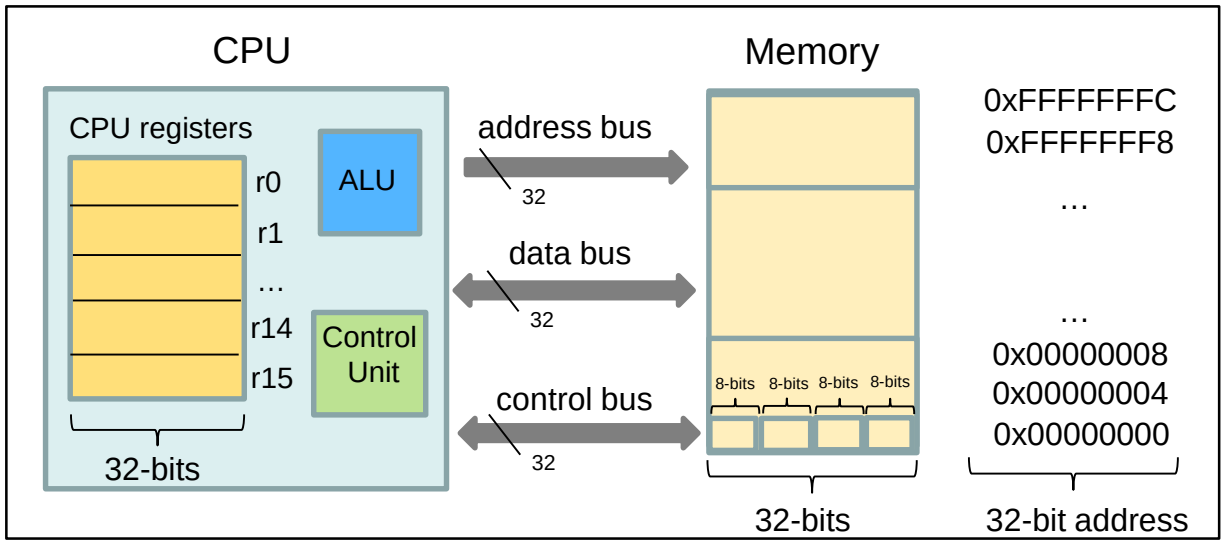
# Modern Computer Architecture
## Fundamental concepts 1

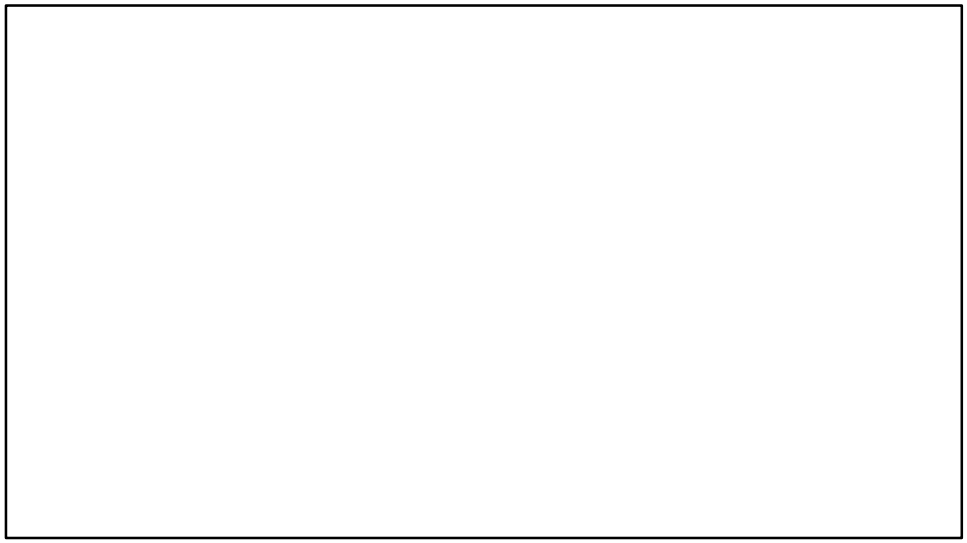



Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks

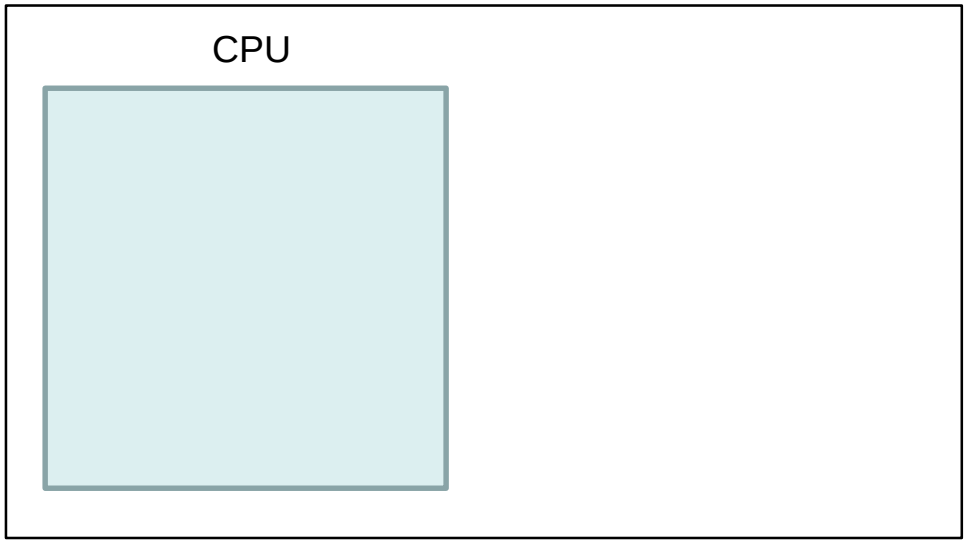


Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
    - Central Processing Unit (CPU): performs basic arithmetic, logical and control operations that are specified by the machine instruction



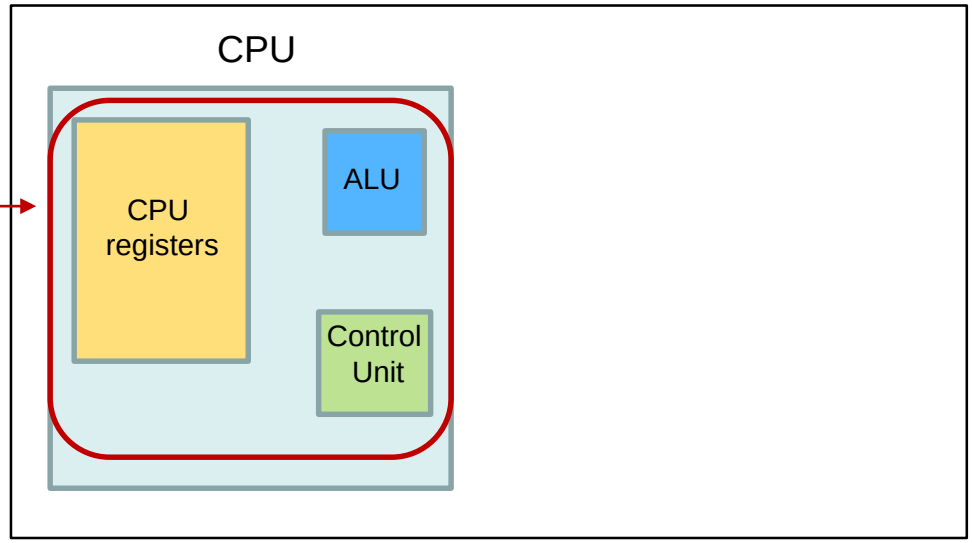

Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
    - Central Processing Unit (CPU): performs basic arithmetic, logical and control operations that are specified by the machine instruction

**CPU**: Identifying the major building blocks of a CPU and deliberately not showing the interconnections between these blocks as yet




Simplified block diagram of a modern computer

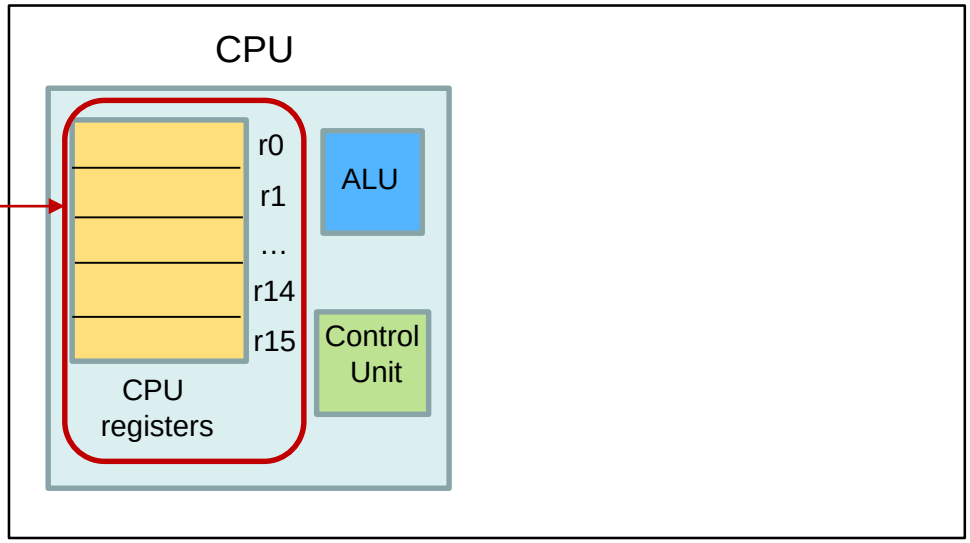
# Modern Computer Architecture

## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
    - Central Processing Unit (CPU): performs basic arithmetic, logical and control operations that are specified by the machine instruction

**CPU registers**: stores temporary data that can be passed quickly to the ALU

Example: CPU registers can be made up of 16 registers, r0 to r15, which are each 32-bits wide



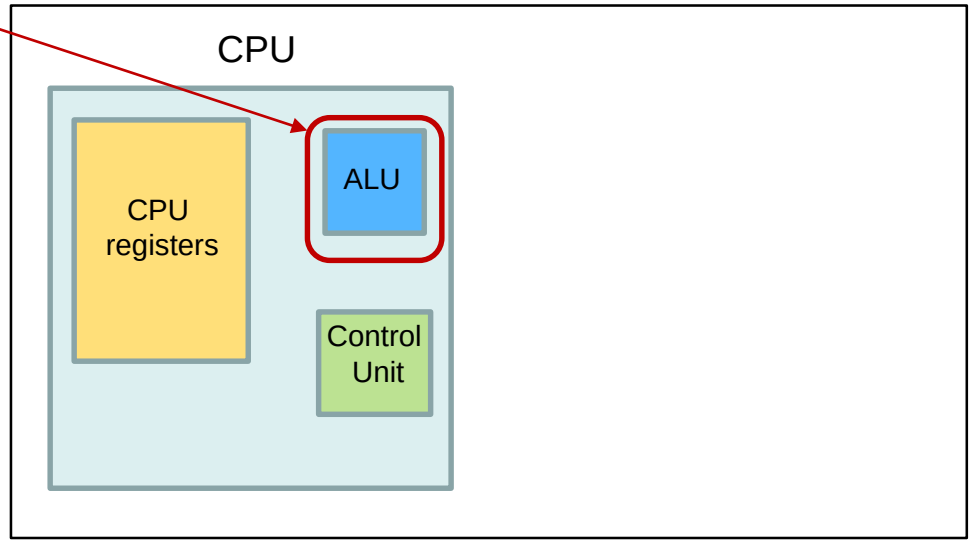

Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
    - Central Processing Unit (CPU): performs basic arithmetic, logical and control operations that are specified by the machine instruction

**ALU**: performs arithmetic, logical and control operations

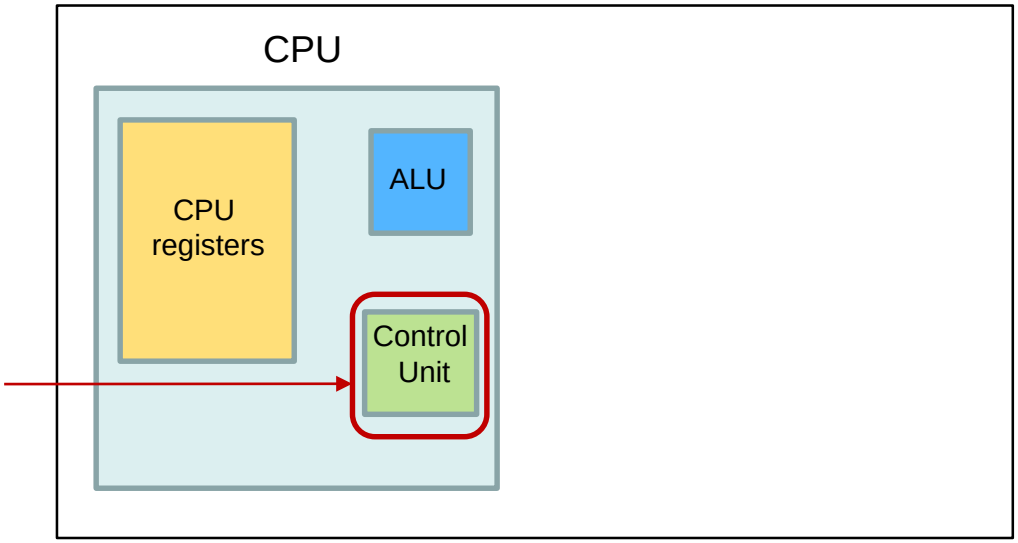



Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
    - Central Processing Unit (CPU): performs basic arithmetic, logical and control operations that are specified by the machine instruction



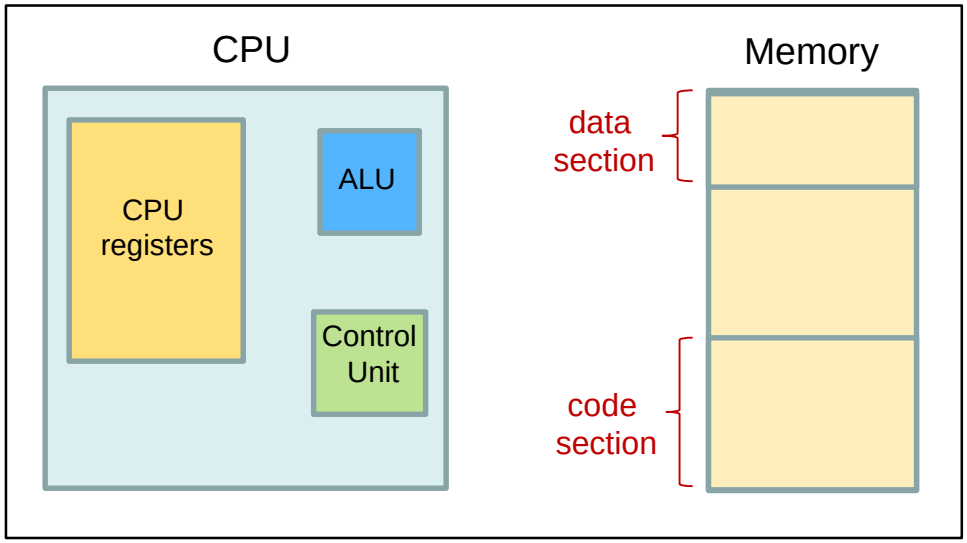

Simplified block diagram of a modern computer

**Control Unit**: controls the fetching of the machine instructions from memory and controls other blocks to ensure that the machine instructions execute accurately

# Modern Computer Architecture

## What does ‘computer architecture’ mean?

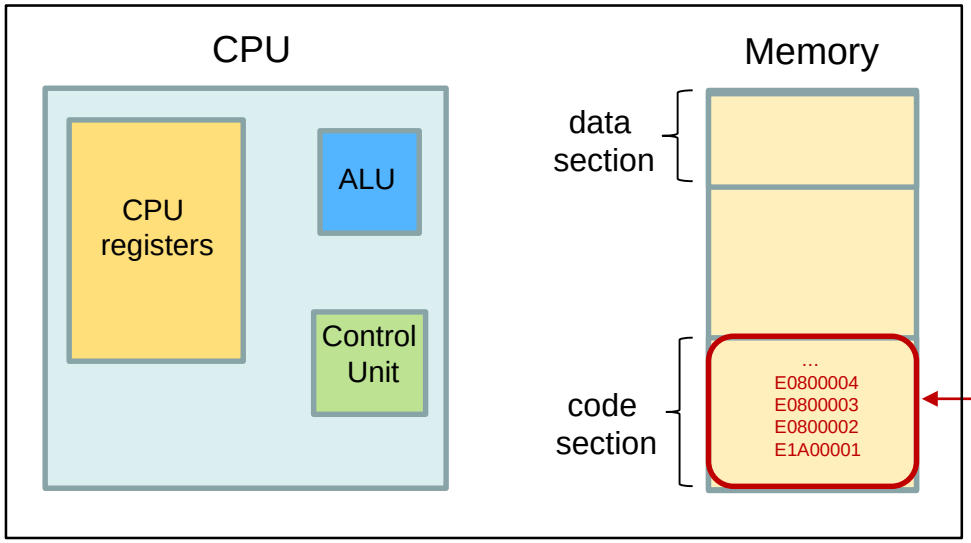
- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
    - Central Processing Unit (CPU): performs basic arithmetic, logical and control operations that are specified by the machine instruction
    - Memory: consists of a data section and a code section and other sections




Simplified block diagram of a modern computer

# Modern Computer Architecture
## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
    - Central Processing Unit (CPU): performs basic arithmetic, logical and control operations that are specified by the machine instruction
    - Memory: consists of a data section and a code section and other sections



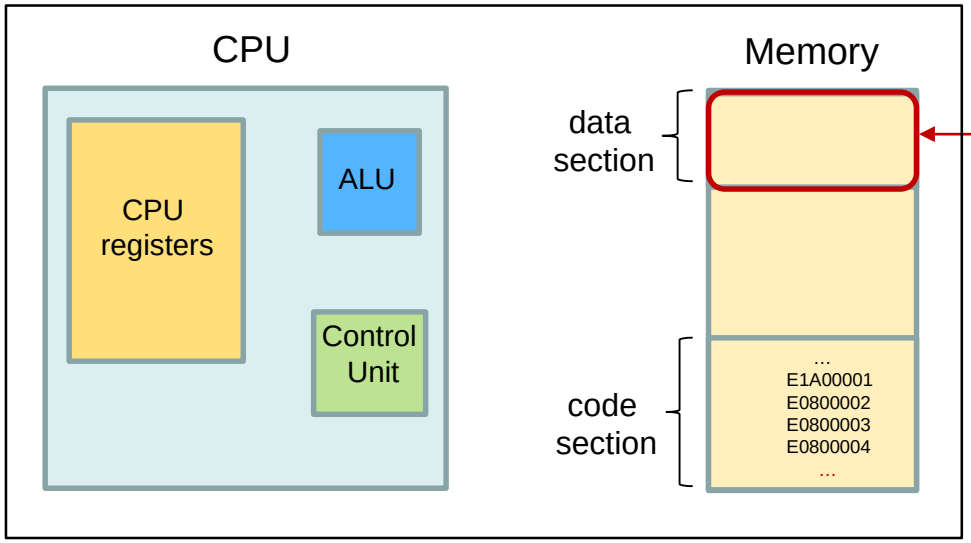

Example: machine instruction for a portion of the code, which was shown in previous slides would be stored in the ‘code section’

Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
    - Central Processing Unit (CPU): performs basic arithmetic, logical and control operations that are specified by the machine instruction
    - Memory: consists of a data section and a code section and other sections

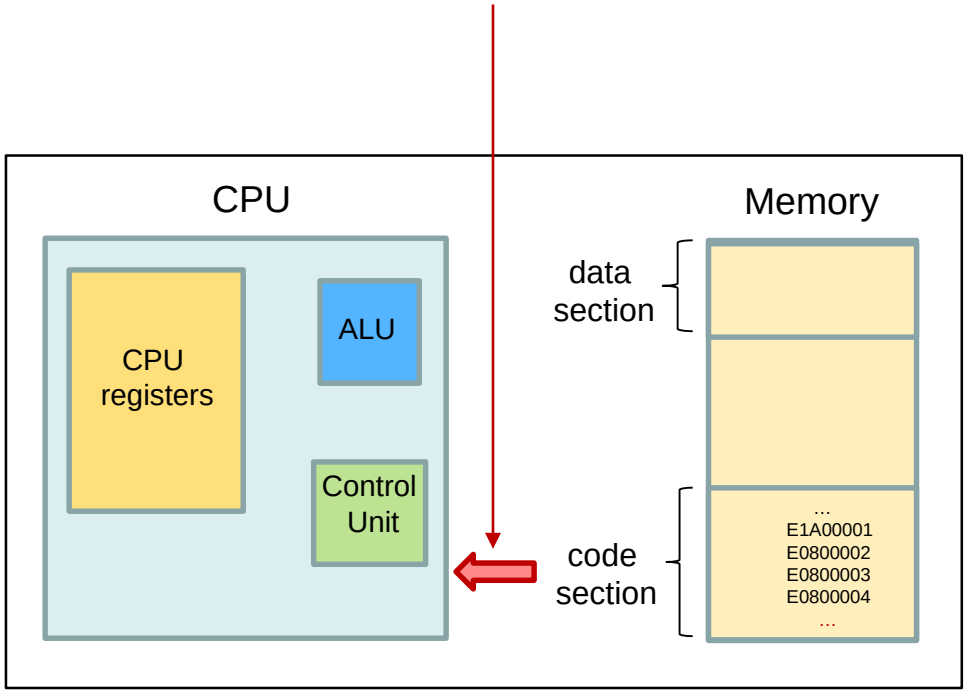



Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
  - Interaction: describing the interface between the internal building blocks
    - Machine instructions from the code section are transferred from memory to the CPU

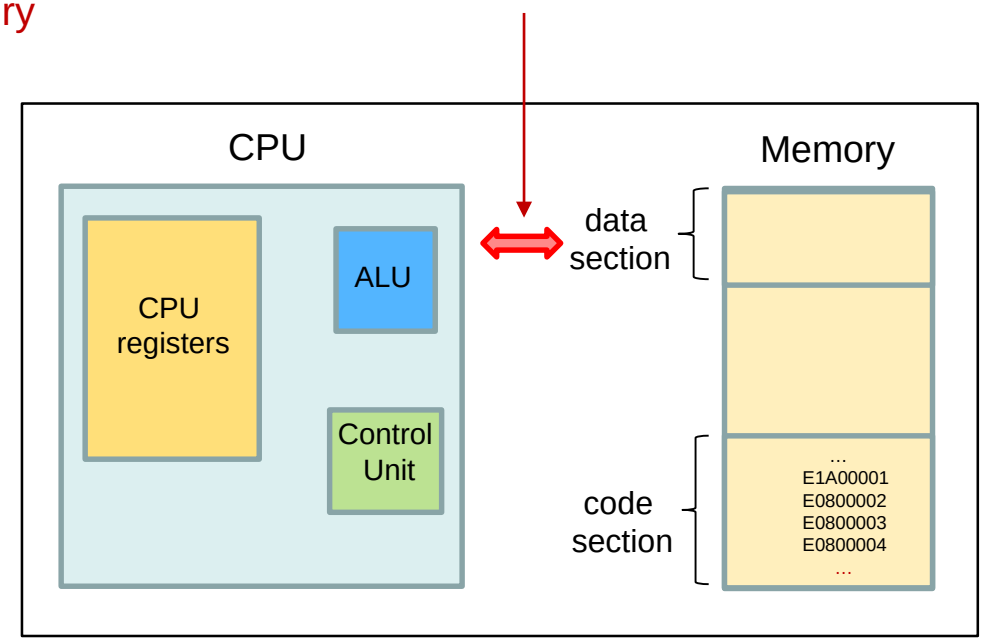



Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
  - Interaction: describing the interface between the internal building blocks
    - Machine instructions from the code section are transferred from memory to the CPU
    - Data from the data section are transferred from memory to the CPU and from the CPU to memory



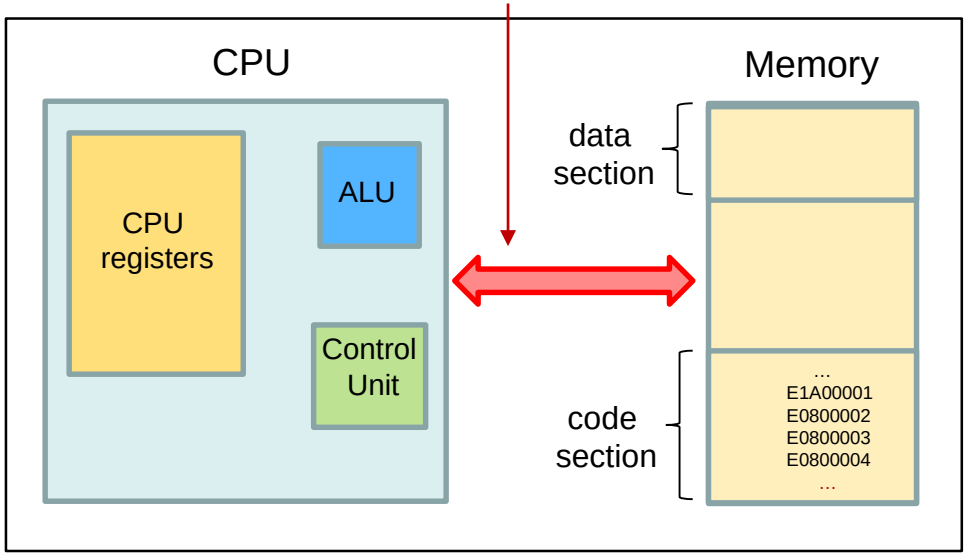

Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
  - Interaction: describing the interface between the internal building blocks
    - Machine instructions from the code section are transferred from memory to the CPU.
    - Data from the data section are transferred from memory to the CPU

Simply identifying that there is a two way communication interface between the CPU and memory. Deliberately not providing any more details of this interface, at this stage

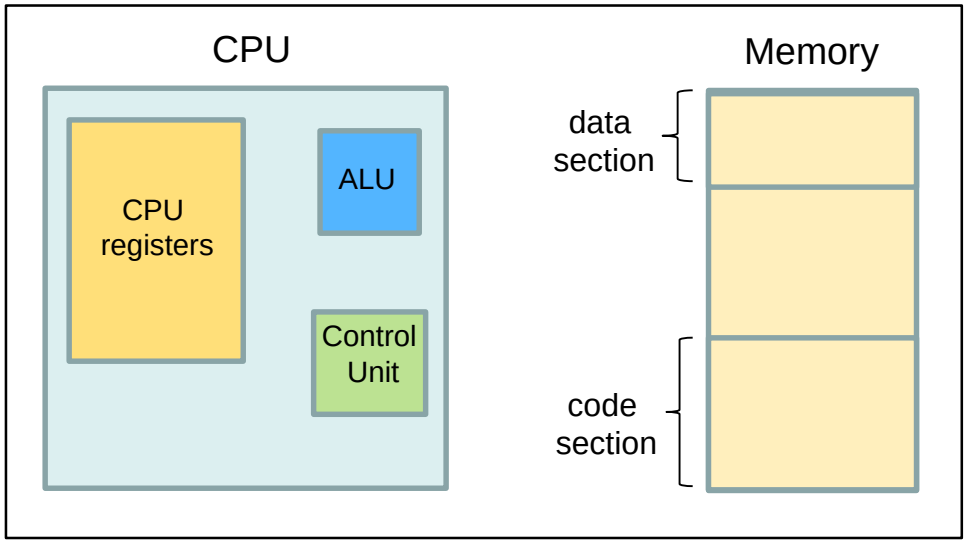



Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does ‘computer architecture’ mean?

- In simple terms, ‘modern computer architecture’ refers to the following elements of a modern computer
  - Internal structure: identifying the major building blocks
  - Interaction: describing the interface between the internal building blocks
  - Functionality: understanding how the building blocks work together to run a single instruction and a program, consisting of many instructions. This topic is discussed in more depth in later slides.




Simplified block diagram of a modern computer

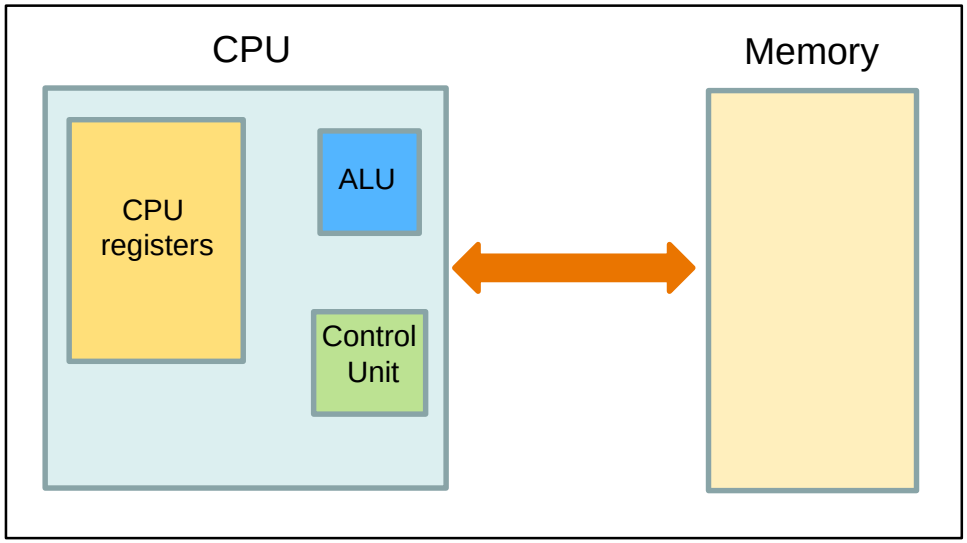
# Examples of Basic Computer Architectures



# Modern Computer Architecture

## Examples of basic computer architectures

- There are many types of computer architectures
- Computer architectures may be classified by the connection between the CPU and memory

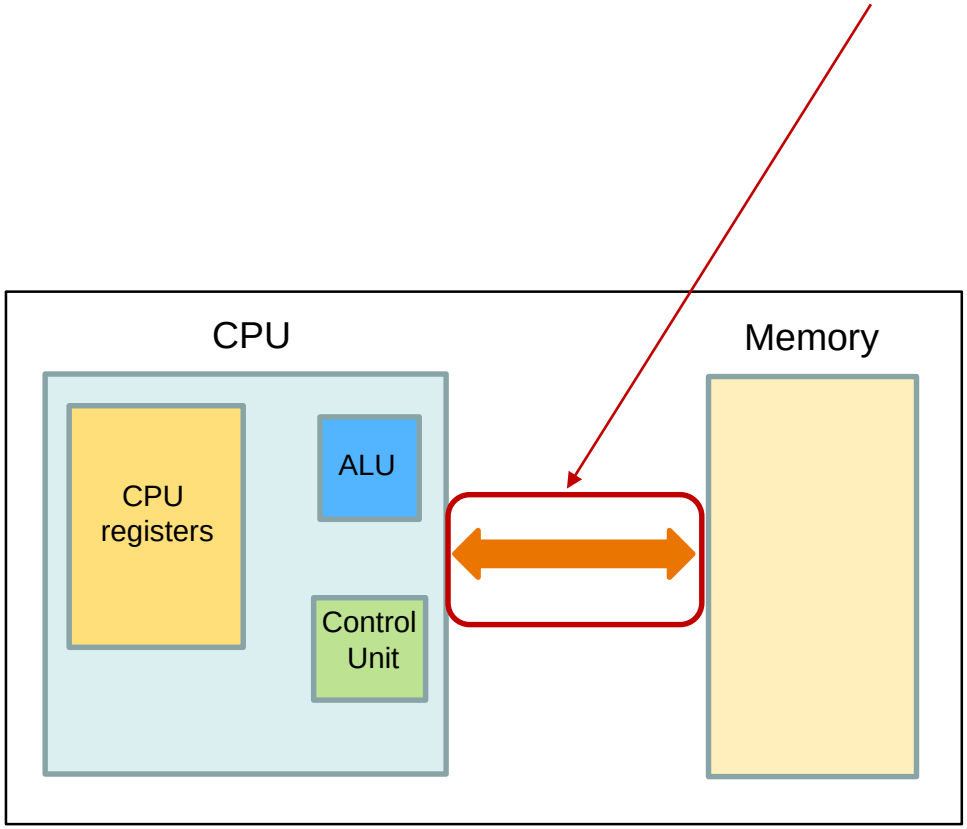



Simplified block diagram of a modern computer

# Modern Computer Architecture

## Examples of basic computer architectures

- There are many types of computer architectures
- Computer architectures may be classified by the connection between the CPU and memory

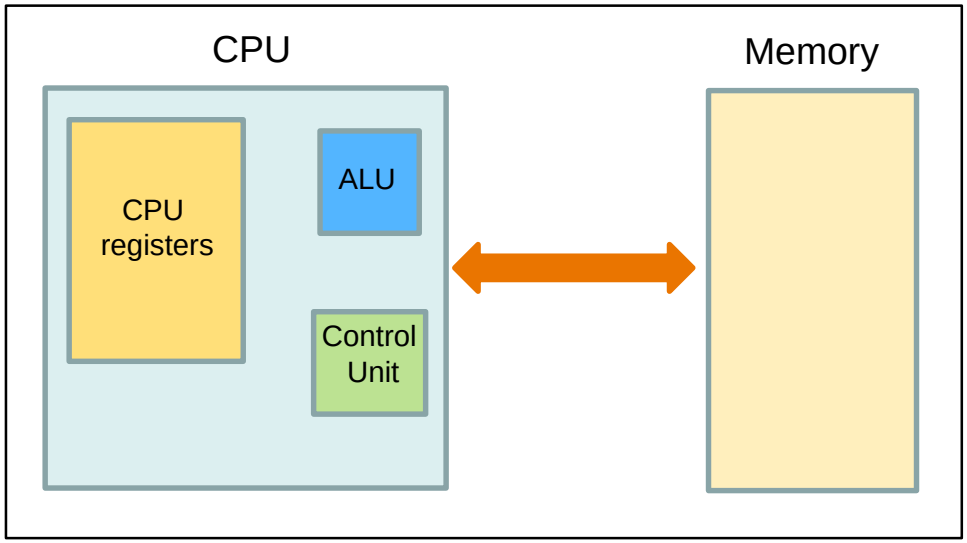



Simplified block diagram of a modern computer

# Modern Computer Architecture

## Examples of basic computer architectures

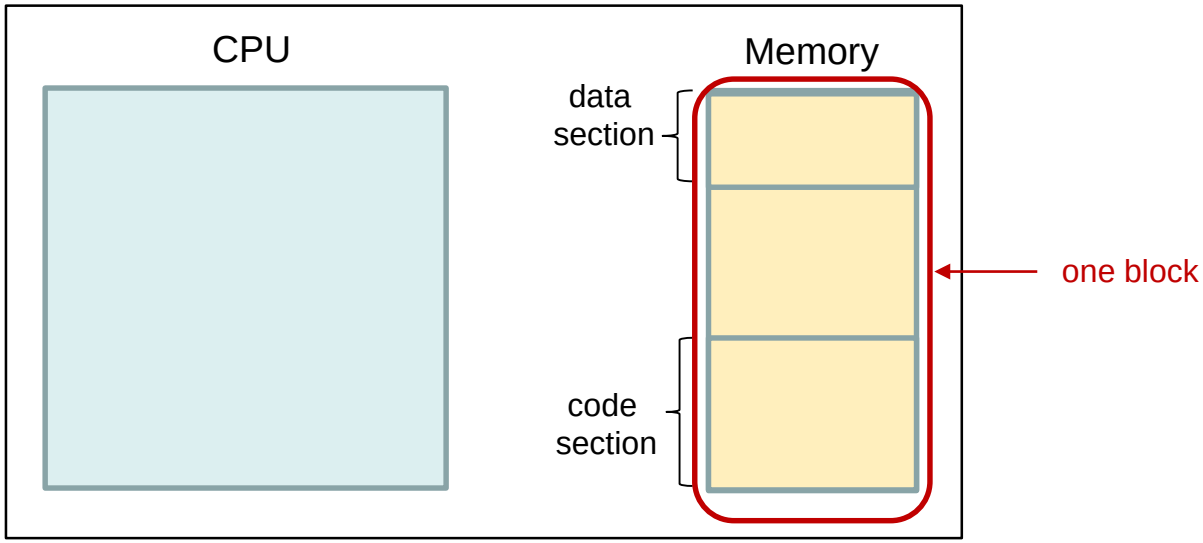
- There are many types of computer architectures
- Computer architectures may be classified by the connection between the CPU and memory
- The two types we will look at
  - Von Neumann
  - Harvard




Simplified block diagram of a modern computer

# Modern Computer Architecture
## Von Neumann architecture

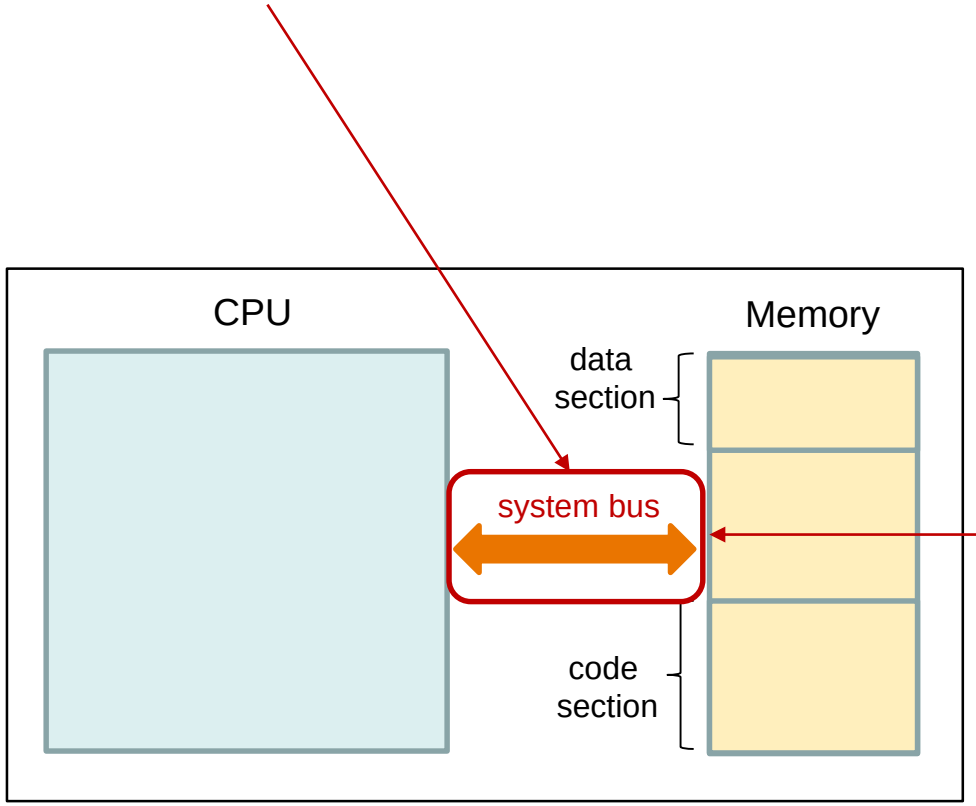
- A von Neumann architecture has the following elements
  - Unified memory that contains both the data section and the code section in one block




Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Von Neumann architecture

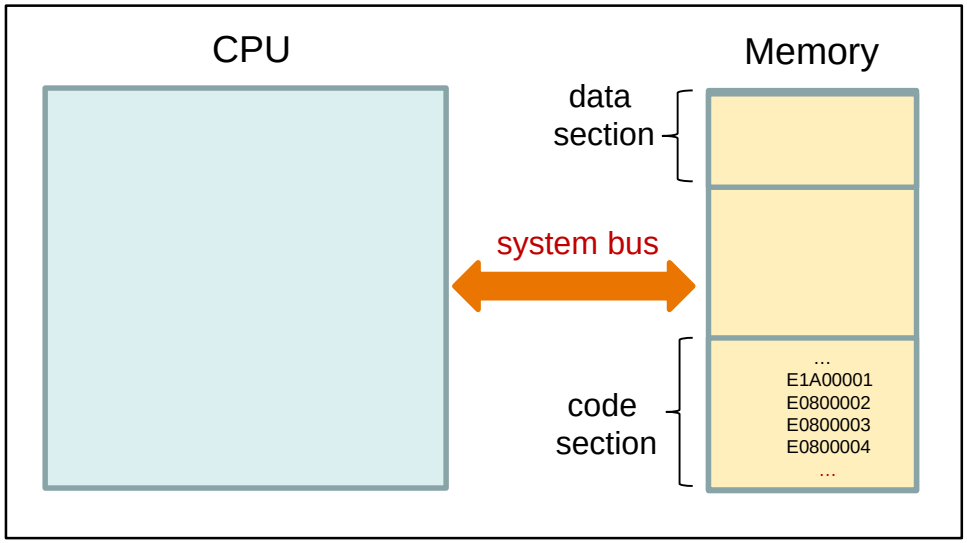
- A von Neumann architecture has the following elements
  - Unified memory that contains both the data section and the code section in one block
  - A single interface or bus between the CPU and memory




Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Von Neumann architecture

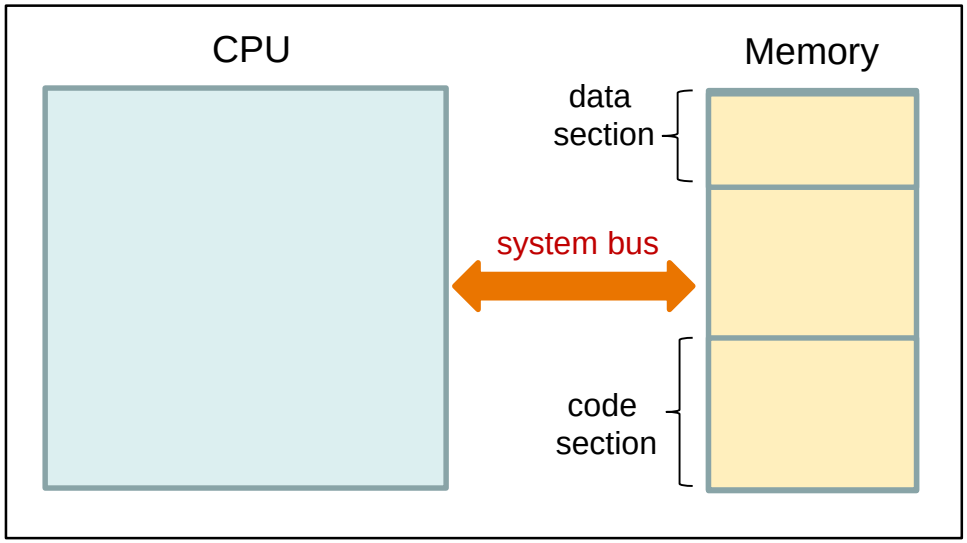
- A von Neumann architecture has the following elements
  - Unified memory that contains both the data section and the code section in one block
  - A single interface or bus between the CPU and memory
    - Thus, a machine instruction can be transferred from memory to the CPU




Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Von Neumann architecture

- A von Neumann architecture has the following elements
  - Unified memory that contains both the data section and the code section in one block
  - A single interface or bus between the CPU and memory
    - Thus, a machine instruction can be transferred from memory to the CPU
    - Or a data element from memory to the CPU

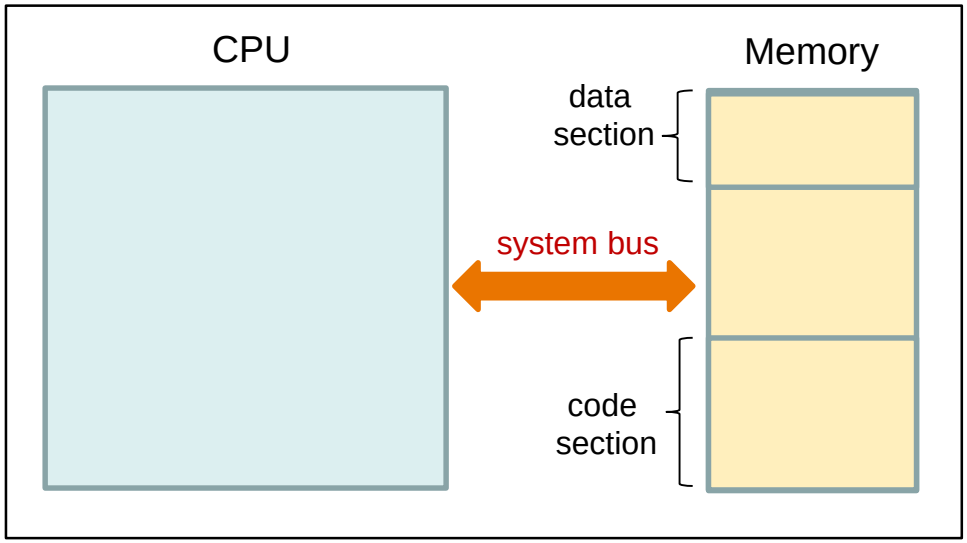



Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Von Neumann architecture

- A von Neumann architecture has the following elements
  - Unified memory that contains both the data section and the code section in one block
  - A single interface or bus between the CPU and memory
    - Thus, a machine instruction can be transferred from memory to the CPU
    - Or a data element from memory to the CPU

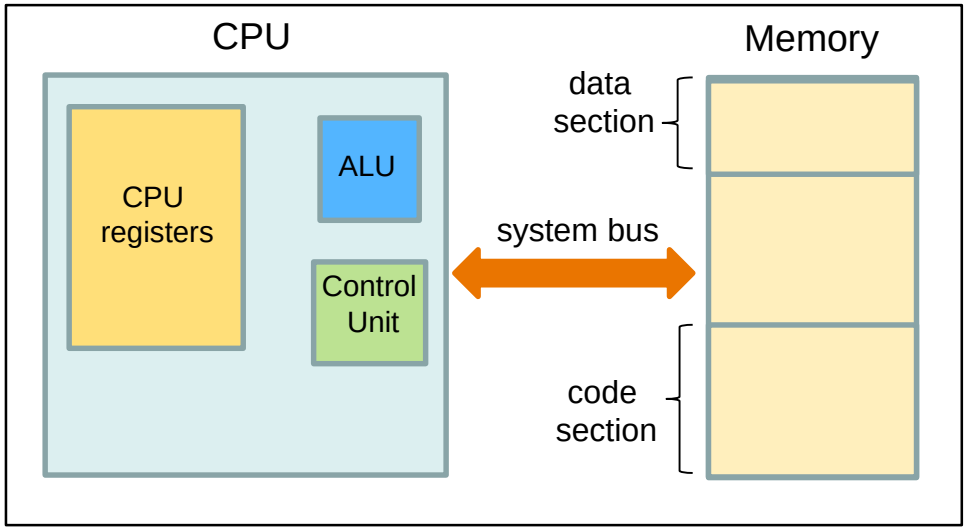
Due to a single system bus, an instruction from the ‘code section’ and a data element from the ‘data section’ cannot be both transferred at the same time




Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Von Neumann architecture

- Advantages of a von Neumann architecture
  - Simple design: the control unit in the CPU gets data and instructions in the same way from one memory block.
  - Since the ‘data section’ & ‘code section’ are accessed in the same way, machine instructions can be written to the ‘data section’ and executed as code.

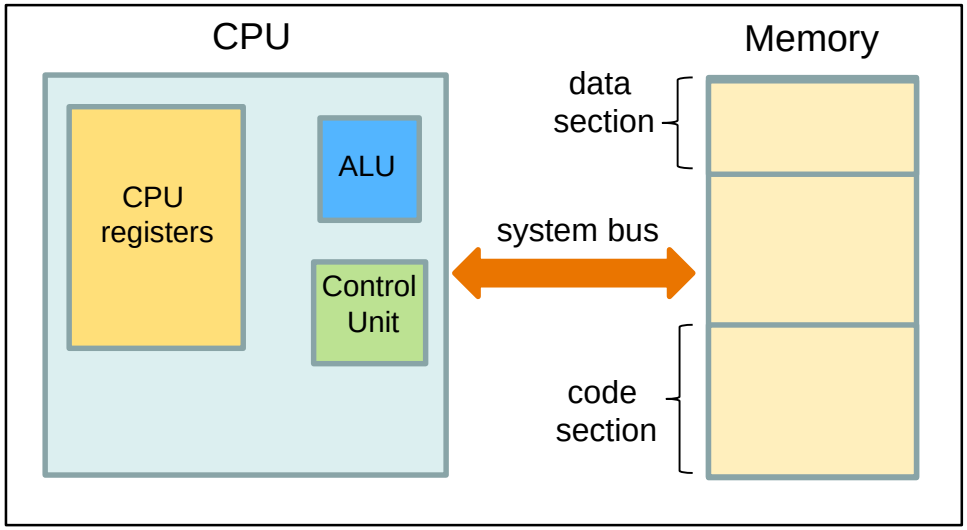



Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Von Neumann architecture

- Advantages of a von Neumann architecture
  - Simple design: the control unit in the CPU gets data and instructions in the same way from one memory block.
  - Since the ‘data section’ & ‘code section’ are accessed in the same way, machine instructions can be written to the ‘data section’ and executed as code.

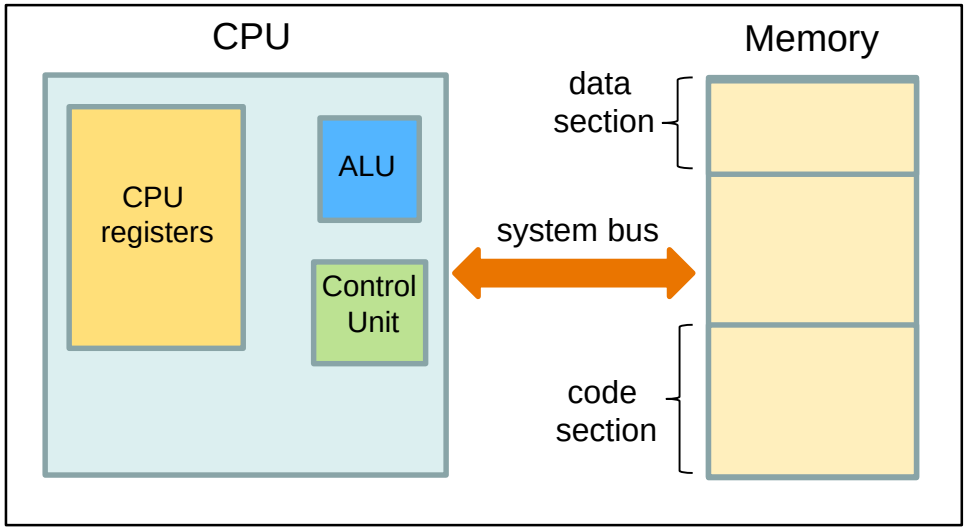
A program whose data is another program is simply an operating system!




Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Von Neumann architecture

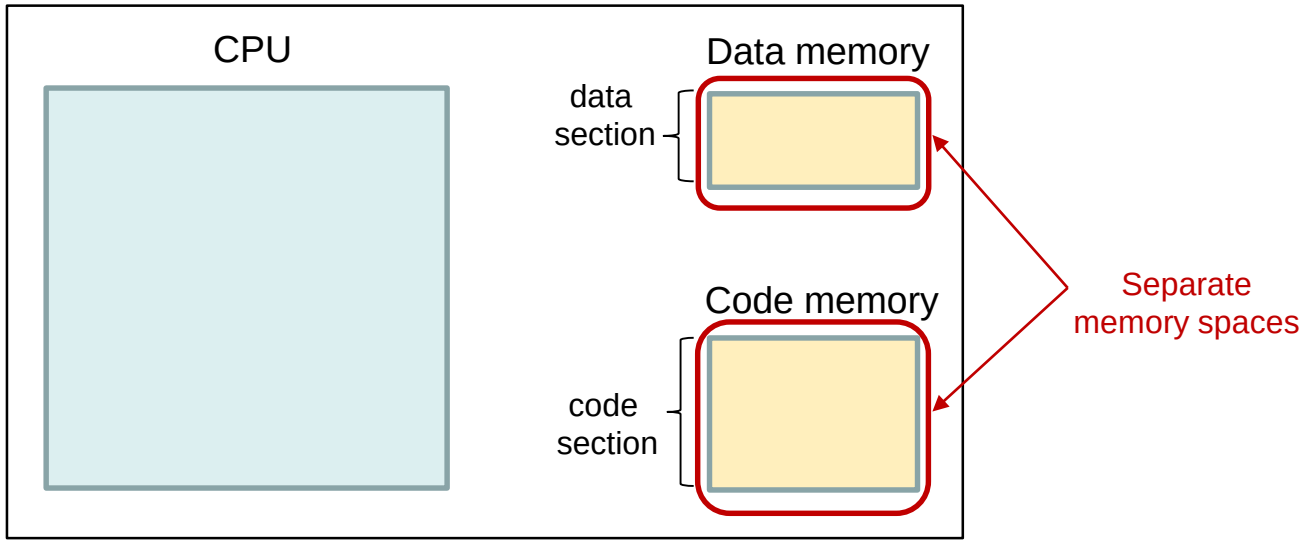
- Advantages of a von Neumann architecture
  - Simple design: the control unit in the CPU gets data and instructions in the same way from one memory block.
  - Since the ‘data section’ & ‘code section’ are accessed in the same way, machine instructions can be written to the ‘data section’ and executed as code.
- Disadvantages of a von Neumann architecture.
  - One bus bottleneck: data and machine instructions cannot be to be transferred to the CPU at the same time.




Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Harvard architecture

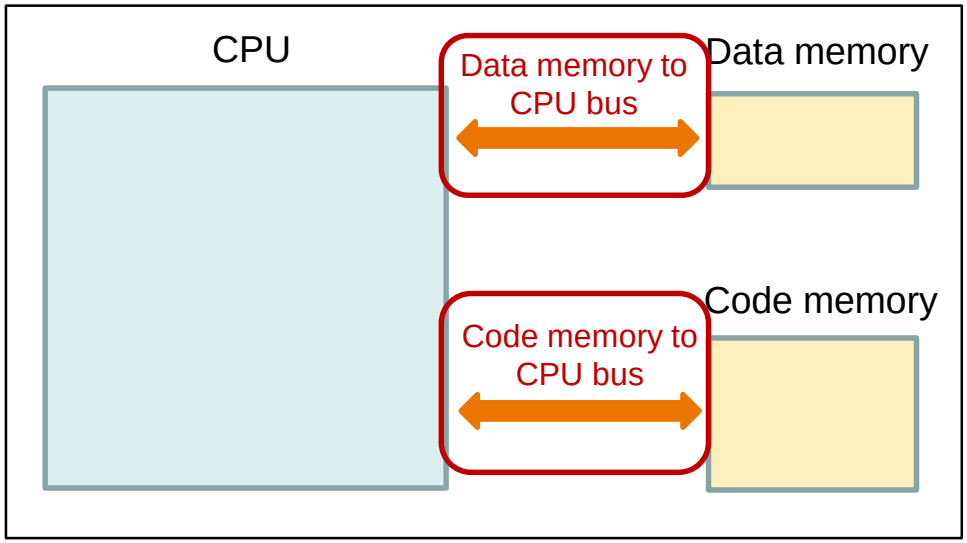
- The Harvard architecture has the following elements
  - Separate memory spaces for the data section and the code section




Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Harvard architecture

- The Harvard architecture has the following elements
  - Separate memory spaces for the data section and the code section
  - Separate buses between the data memory and the CPU, and the code memory and the CPU




Bus = data, address & control bus

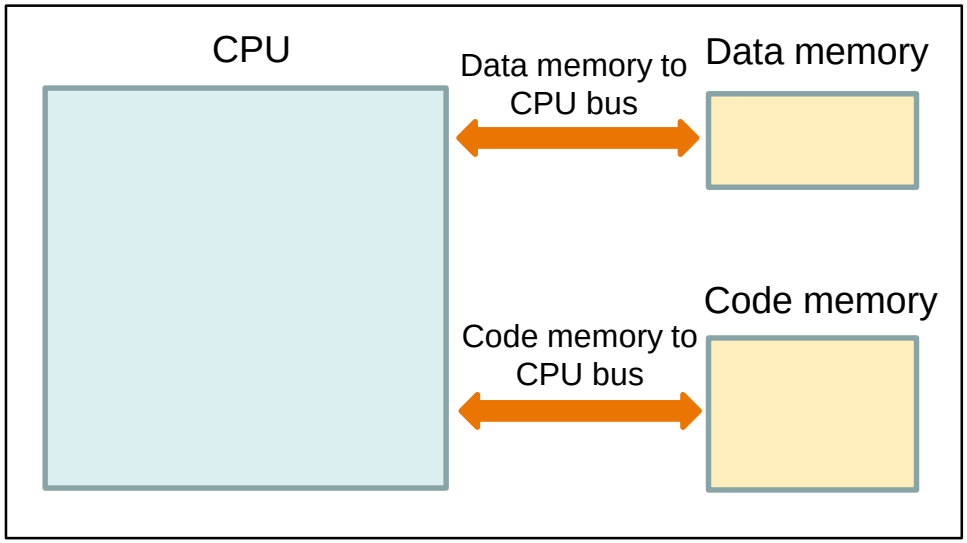
Data can pass through the bus in **half duplex** mode to and from the CPU

Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Harvard architecture

- The Harvard architecture has the following elements
  - Separate memory spaces for the data section and the code section
  - Separate buses between the data memory and the CPU, and the code memory and the CPU

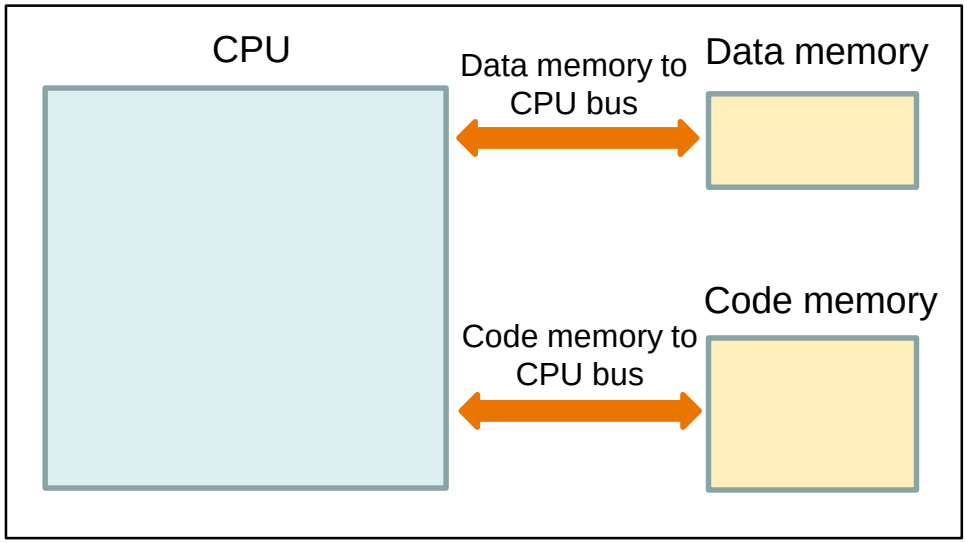
Due to separate buses, an instruction from ‘code memory’ and a data element from ‘data memory’ can both be transferred at the same time




Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Harvard architecture

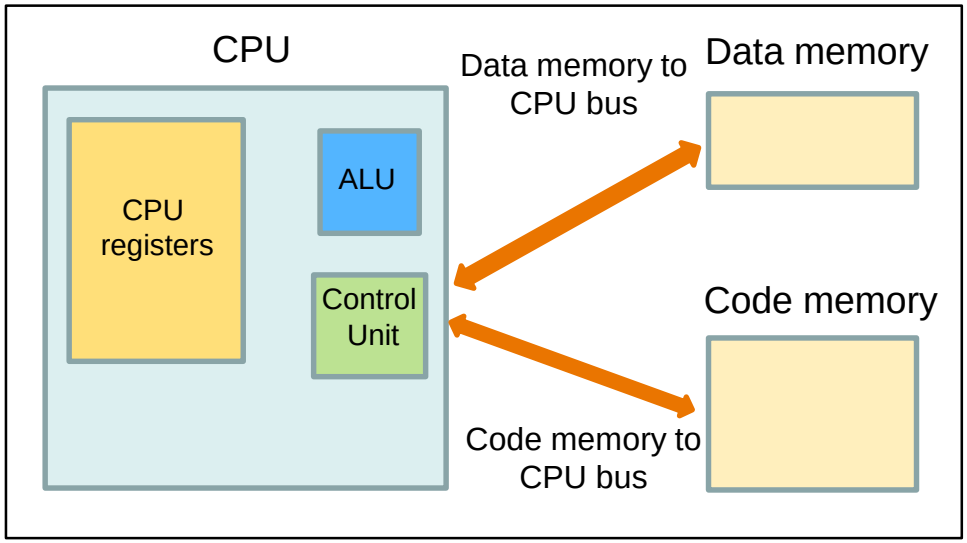
- Advantages of a Harvard architecture
  - Faster program execution: two buses allow parallel access to data and instructions. Execution can be 2x faster than a von Neumann architecture
  - Program only writes to the data memory. Program cannot mistakenly overwrite instructions in the code memory




Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## Harvard architecture

- Advantages of a Harvard architecture
  - Faster program execution: two buses allow parallel access to data and instructions. Execution can be 2x faster than a von Neumann architecture
  - Program only writes to the data memory. Program cannot mistakenly overwrite instructions in the code memory
- Disadvantages of a Harvard architecture
  - More complex design of the control unit that interfaces with the two buses.
  - Cannot interpret contents from 'data memory' as code.




Both buses interface with the control unit of the CPU

Simplified block diagram of a von Neumann architecture

# Modern Computer Architecture
## von Neumann vs Harvard architecture

- Which is better?
  - Both architectures have advantages and disadvantages: one architecture is not better than the other in all aspects
- Which one is used in modern computers?
  - Both architectures are used
- Where are these architectures used?
  - Von Neumann: used for desktop computers, laptops and high performance computers. Supports the use of an operating system, where machine instructions of a program can be written in the ‘data section’ and later executed as a program.
  - Harvard: used for small embedded computers and digital signal processing processors where speed of processing instructions is critically important.

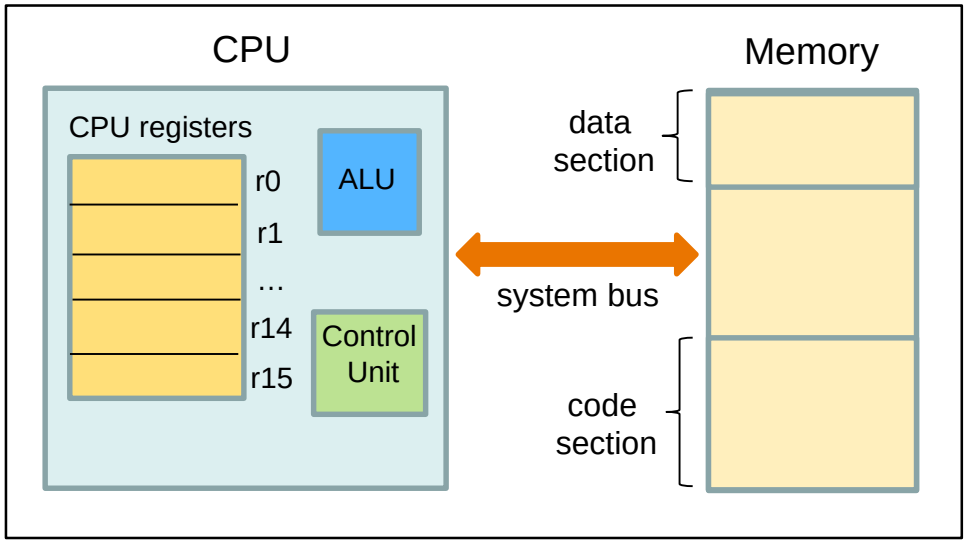
# Further concepts related to computer architecture



# Modern Computer Architecture

## What does a 32-bit processor mean?

- A 32-bit processor means:

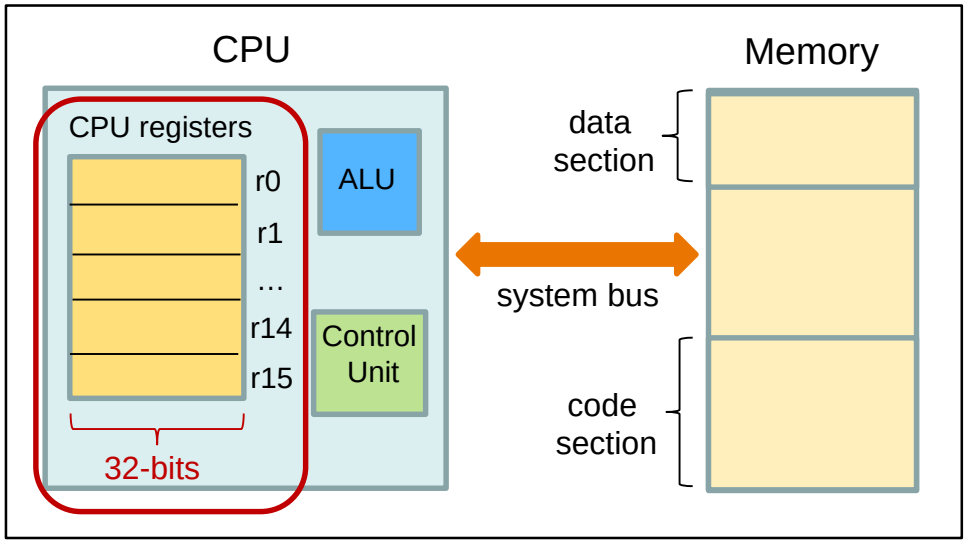



Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does a 32-bit processor mean?

- A 32-bit processor means:
  - CPU registers, r0 – r15, can each store 32-bits of temporary data

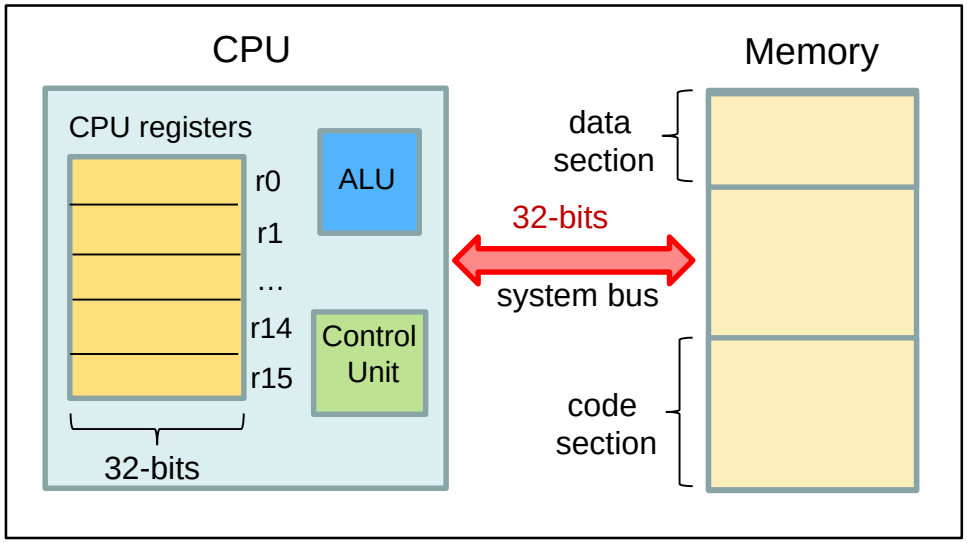



Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does a 32-bit processor mean?

- A 32-bit processor means:
  - CPU registers, r0 – r15, can each store 32-bits of temporary data
  - The system bus between the CPU and memory is also 32-bits

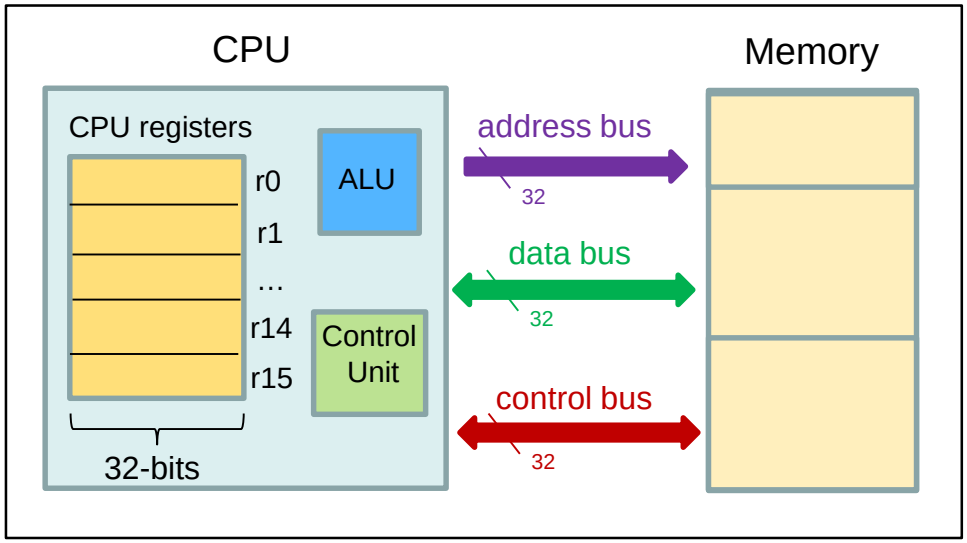



Simplified block diagram of a modern computer

# Modern Computer Architecture

## What does a 32-bit processor mean?

- A 32-bit processor means:
  - CPU registers, r0 – r15, can each store 32-bits of temporary data
  - The system bus between the CPU and memory is also 32-bits.
    - The address bus is 32-bits wide
    - The data bus is 32-bits wide
    - The control bus can be up to 32-bits wide

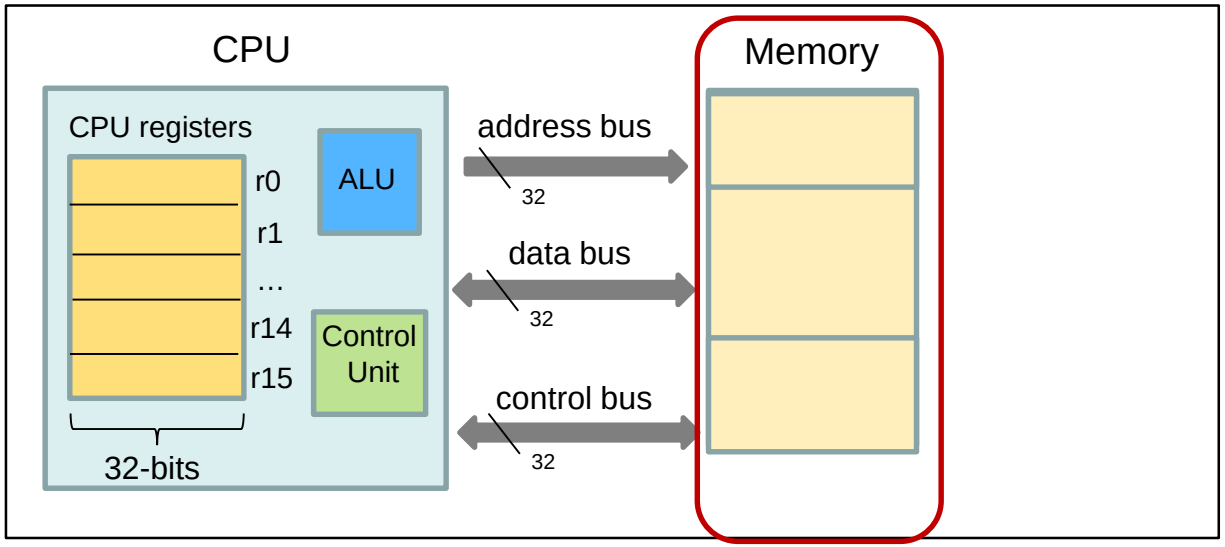



Simplified block diagram of a modern computer

# Modern Computer Architecture

## Memory: register width and address

- Consider a 32-bit processor, what is the register width in Memory?

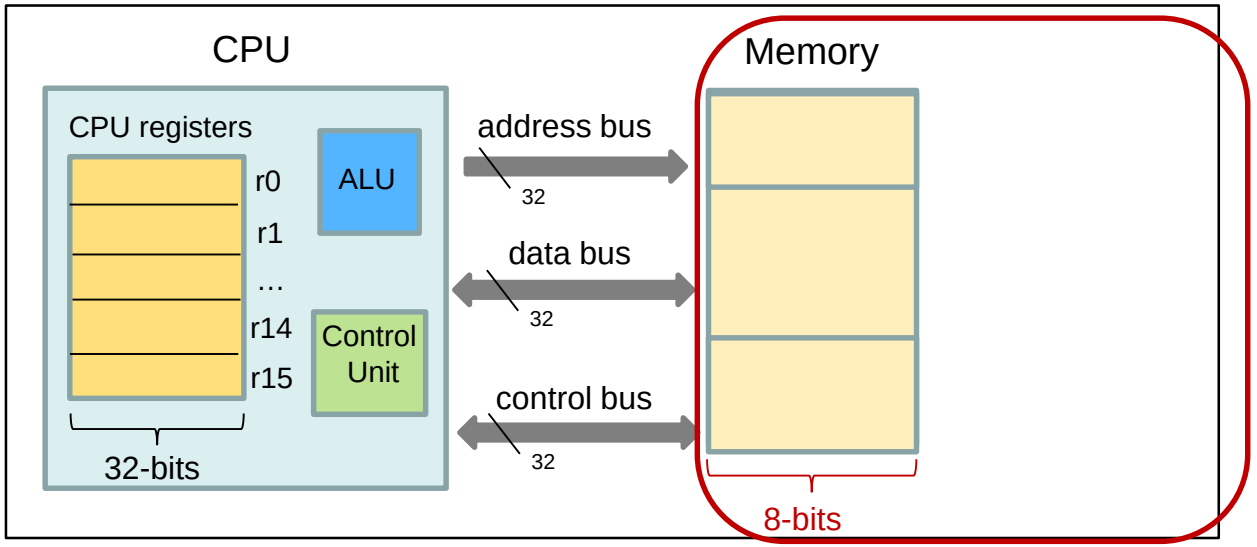



Simplified block diagram of a modern computer

# Modern Computer Architecture

## Memory: register width and address

- Consider a 32-bit processor, what is the register width in Memory?
  - Each register in Memory is 8-bits wide and each has a unique address

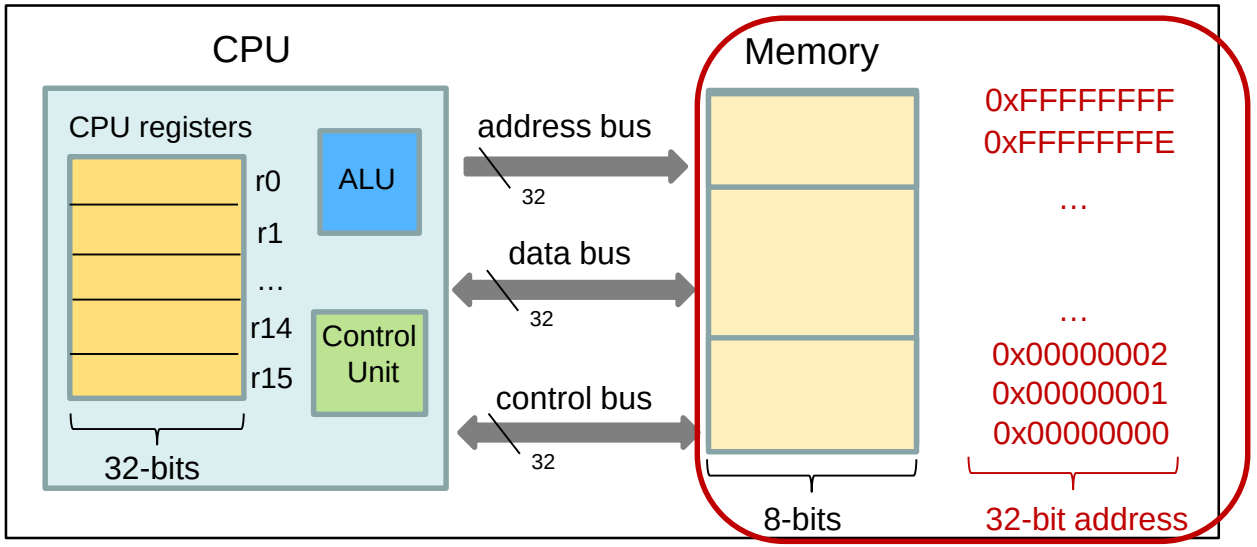



Simplified block diagram of a modern computer

# Modern Computer Architecture

## Memory: register width and address

- Consider a 32-bit processor, what is the register width in Memory?
  - Each register in Memory is 8-bits wide and each has a unique address
  - The address bus is 32-bits wide, so ($2^{32} - 1$) unique addresses can be accessed. [Note: $2^{32} - 1$ = 0xFFFFFFFF]

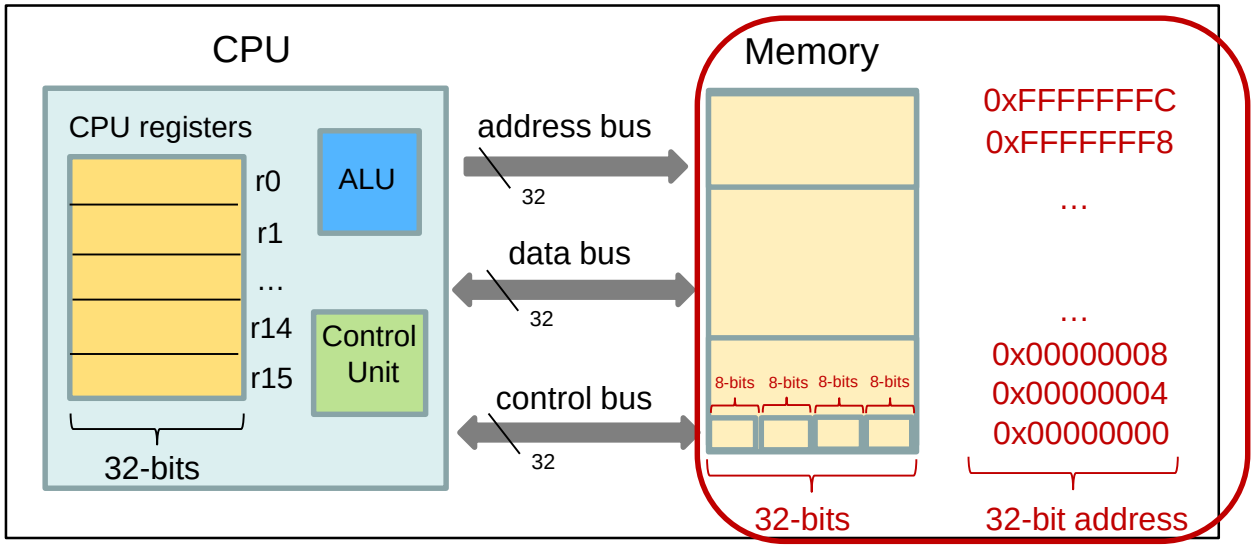



Simplified block diagram of a modern computer

# Modern Computer Architecture

## Memory: register width and address

- Consider a 32-bit processor, what is the register width in Memory?
  - Each register in Memory is 8-bits wide and each has a unique address
  - The address bus is 32-bits wide, so ($2^{32} - 1$) unique addresses can be accessed. [Note: $2^{32} - 1$ = 0xFFFFFFFF]
- The Memory can be redrawn to have a width of 32-bits, where the address of each 32-bit word increments by four. This representation is often used.

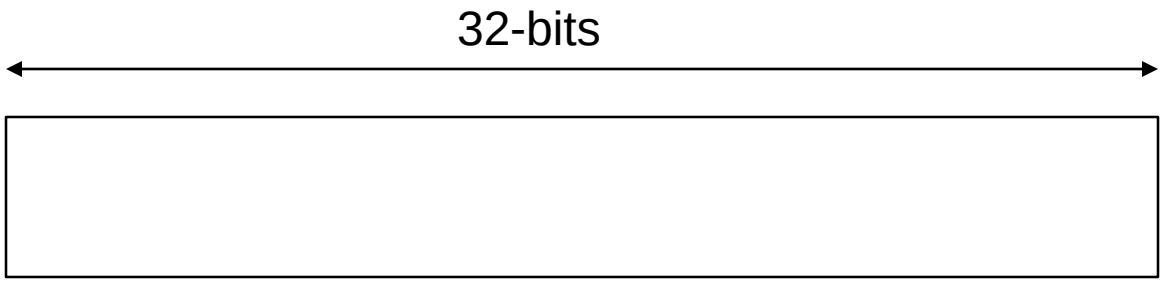



Simplified block diagram of a modern computer

# Modern Computer Architecture

## Storing a 32-bit number in memory

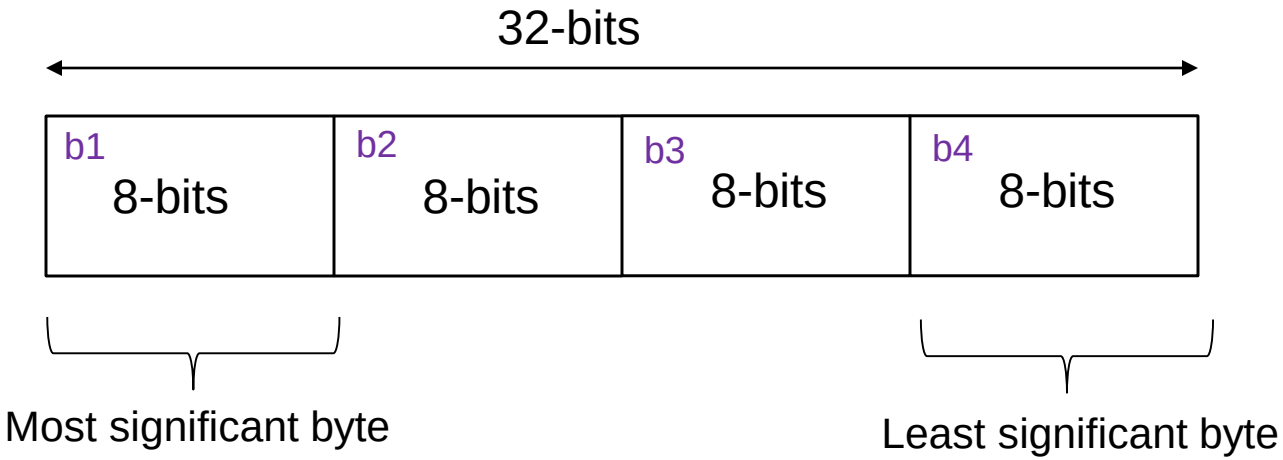
- The contents of a CPU register may be stored in Memory. A CPU register is 32-bits, whereas a register width in memory is 8-bits.
- How is a 32-bit number stored in memory?



# Modern Computer Architecture

## Storing a 32-bit number in memory

- The contents of a CPU register may be stored in Memory. A CPU register is 32-bits, whereas a register width in memory is 8-bits.
- How is a 32-bit number stored in memory?
  - Break up the 32-bit number into four bytes, denoted by b1, b2, b3, b4 where b1 is the most significant byte and b4 is the least significant byte

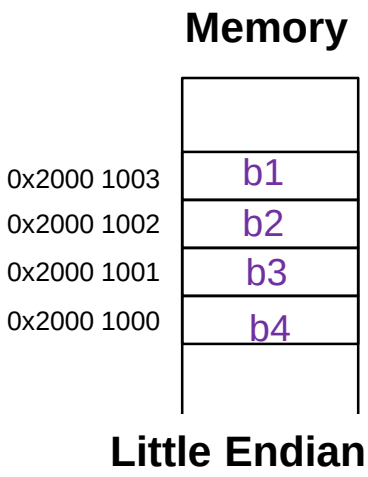


# Modern Computer Architecture

## Storing a 32-bit number in memory

- The contents of a CPU register may be stored in Memory. A CPU register is 32-bits, whereas a register width in memory is 8-bits.
- How is a 32-bit number stored in memory?
  - Break up the 32-bit number into four bytes, denoted by b1, b2, b3, b4 where b1 is the most significant byte and b4 is the least significant byte
  - The four bytes may be stored in two sequences

    Little Endian: the least significant byte (b4) is stored in the smallest memory address location



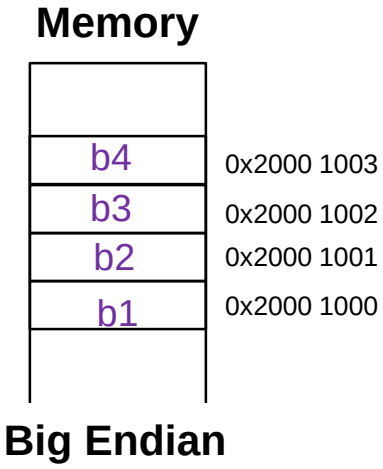
# Modern Computer Architecture

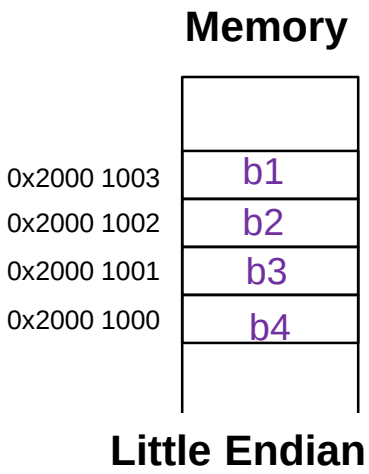
## Storing a 32-bit number in memory

- The contents of a CPU register may be stored in Memory. A CPU register is 32-bits, whereas a register width in memory is 8-bits.
- How is a 32-bit number stored in memory?
  - Break up the 32-bit number into four bytes, denoted by b1, b2, b3, b4 where b1 is the most significant byte and b4 is the least significant byte
  - The four bytes may be stored in two sequences

    Little Endian: the least significant byte (b4) is stored in the smallest memory address location

    Big Endian: the most significant byte (b1) is stored in the smallest memory address location

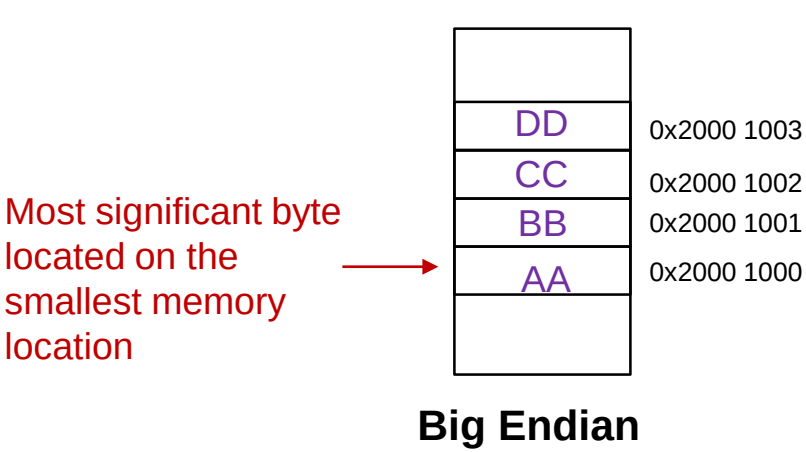


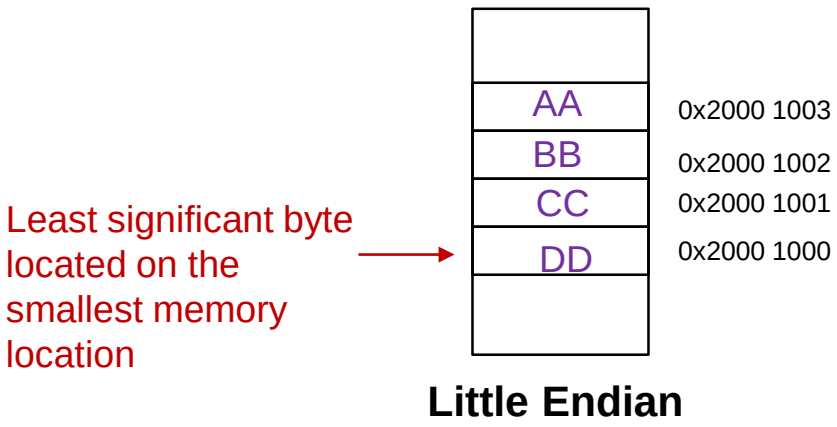



# Modern Computer Architecture

## Storing a 32-bit number in memory

- Example: store the word 0xAABBCCDD into memory according to both little endian and big endian format.



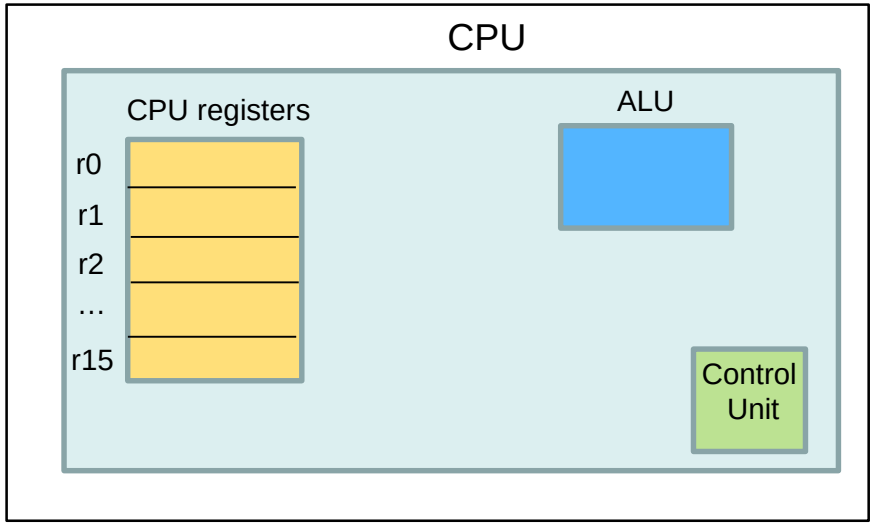



# Machine instructions and assembly instructions



# Modern Computer Architecture
## Machine instructions

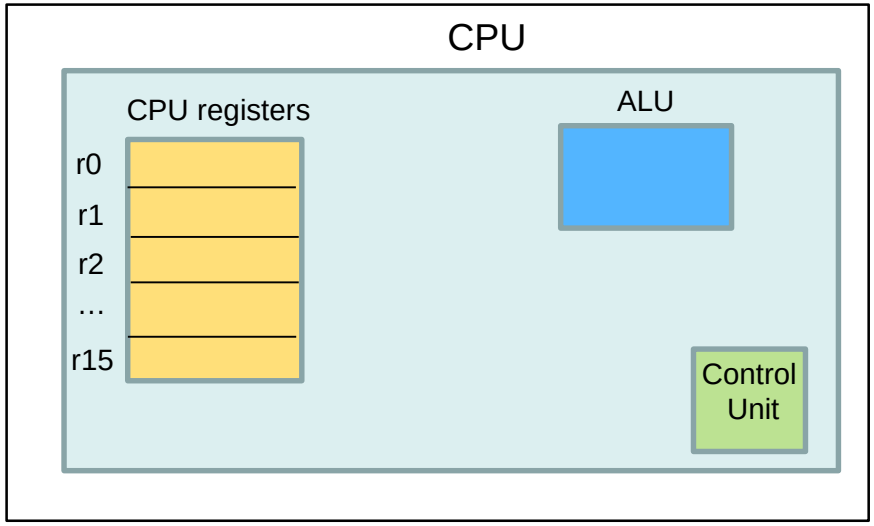
- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU




Simplified block diagram of a CPU

# Modern Computer Architecture
## Machine instructions

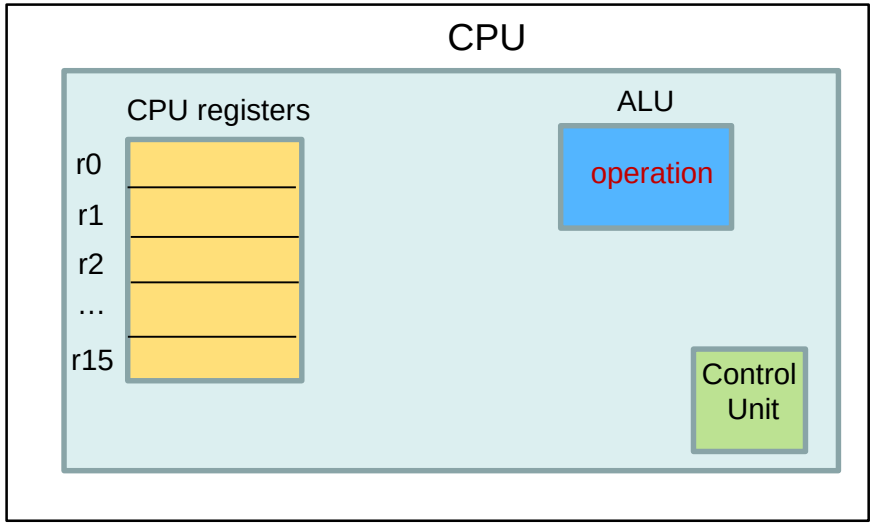
- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands




Simplified block diagram of a CPU

# Modern Computer Architecture
## Machine instructions

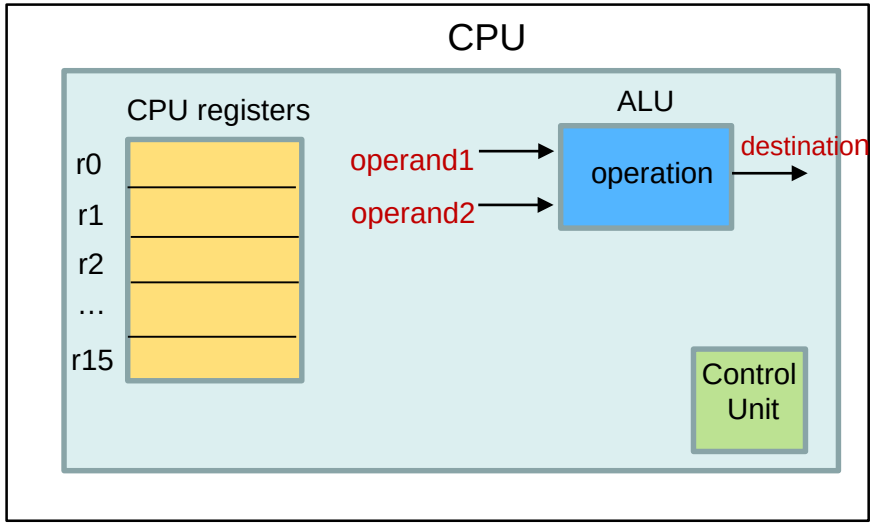
- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands
    - Opcode: specifies the operation that should be performed by the ALU. Examples of operations are addition, subtraction, moving data between CPU registers, …




Simplified block diagram of a CPU

# Modern Computer Architecture
## Machine instructions

- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands
    - Opcode: specifies the operation that should be performed by the ALU. Examples of operations are addition, subtraction, moving data between CPU registers, …
    - Operands: inputs to the ALU: operand1 and operand 2. Operands can either be CPU registers or values.

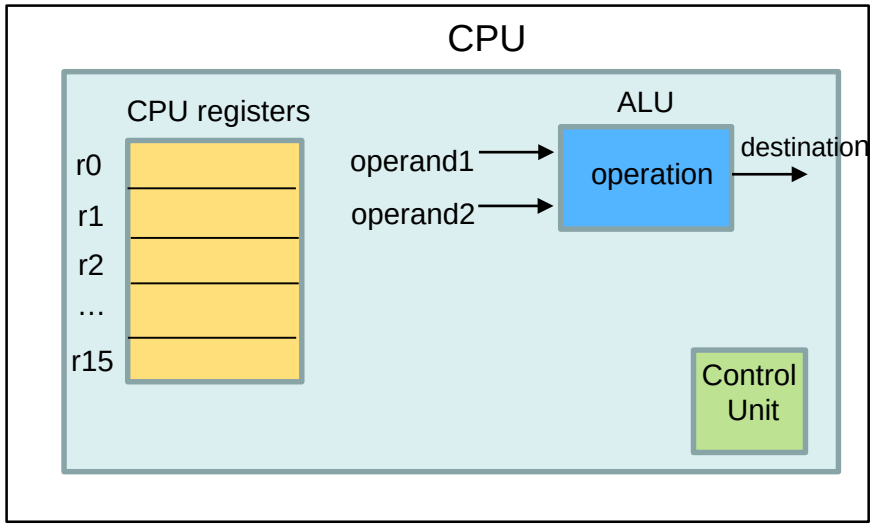



Simplified block diagram of a CPU

# Modern Computer Architecture
## Machine instructions

- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands
    - Opcode: specifies the operation that should be performed by the ALU. Examples of operations are addition, subtraction, moving data between CPU registers, …
    - Operands: inputs to the ALU: operand1 and operand 2. Operands can either be CPU registers or values.
- 32-bit machine instruction example

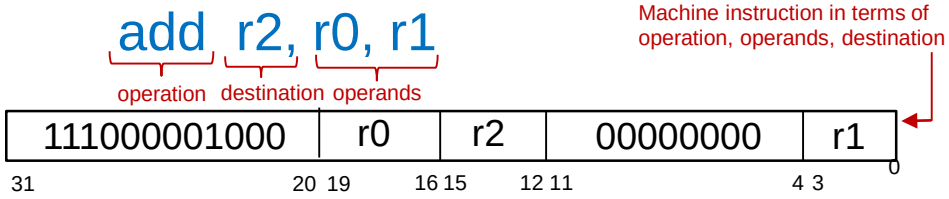
  add r2, r0, r1 | r2 = r0 + r1

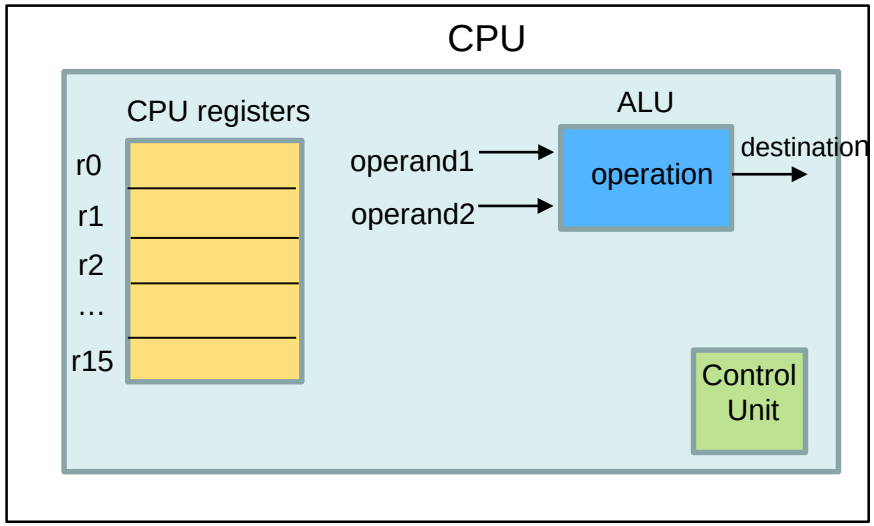



Simplified block diagram of a CPU

# Modern Computer Architecture
## Machine instructions

- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands
    - Opcode: specifies the operation that should be performed by the ALU. Examples of operations are addition, subtraction, moving data between CPU registers, …
    - Operands: inputs to the ALU: operand1 and operand 2. Operands can either be CPU registers or values.
- 32-bit machine instruction example

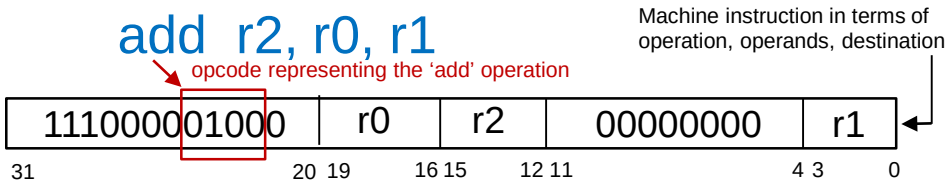


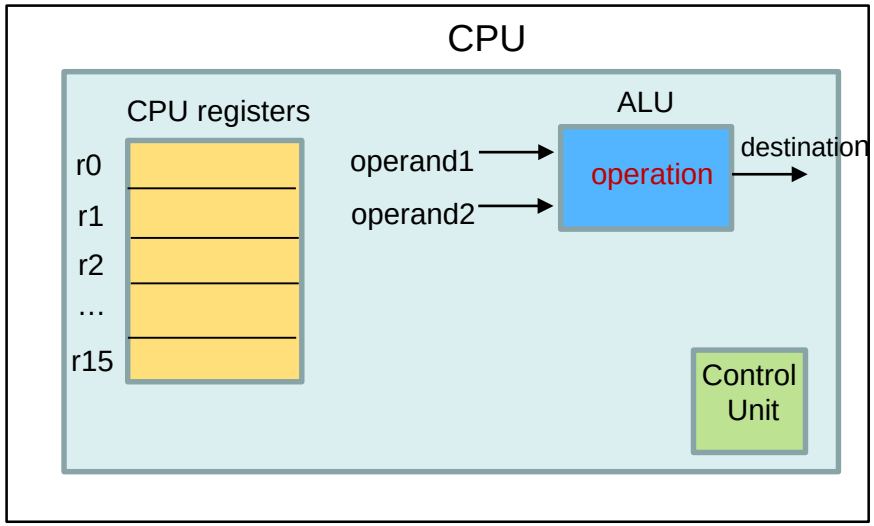




Simplified block diagram of a CPU

# Modern Computer Architecture
## Machine instructions

- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands
    - Opcode: specifies the operation that should be performed by the ALU. Examples of operations are addition, subtraction, moving data between CPU registers, …
    - Operands: inputs to the ALU: operand1 and operand 2. Operands can either be CPU registers or values.
- 32-bit machine instruction example

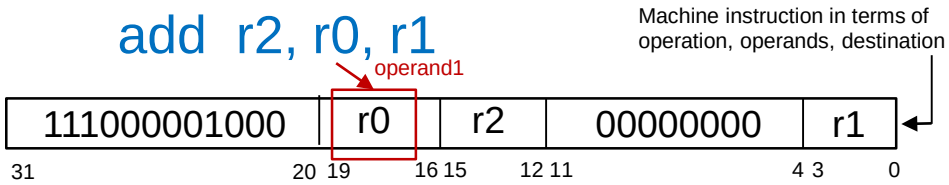


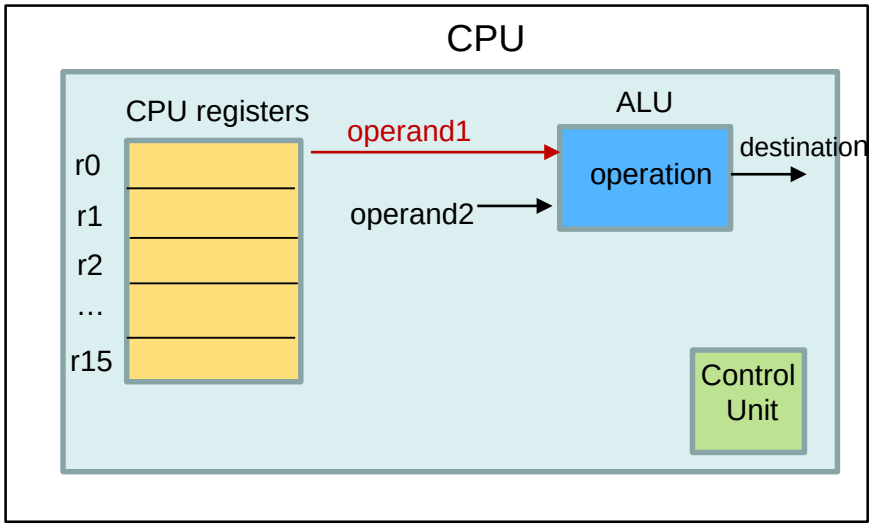




Simplified block diagram of a CPU

# Modern Computer Architecture
## Machine instructions

- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands
    - Opcode: specifies the operation that should be performed by the ALU. Examples of operations are addition, subtraction, moving data between CPU registers, …
    - Operands: inputs to the ALU: operand1 and operand 2. Operands can either be CPU registers or values.
- 32-bit machine instruction example

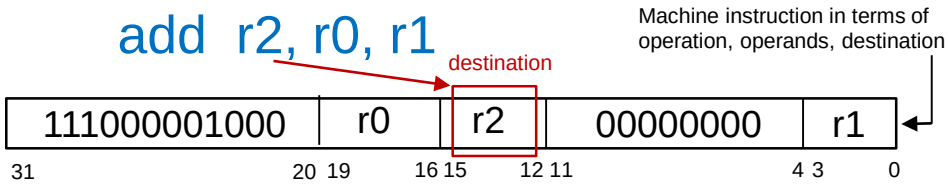


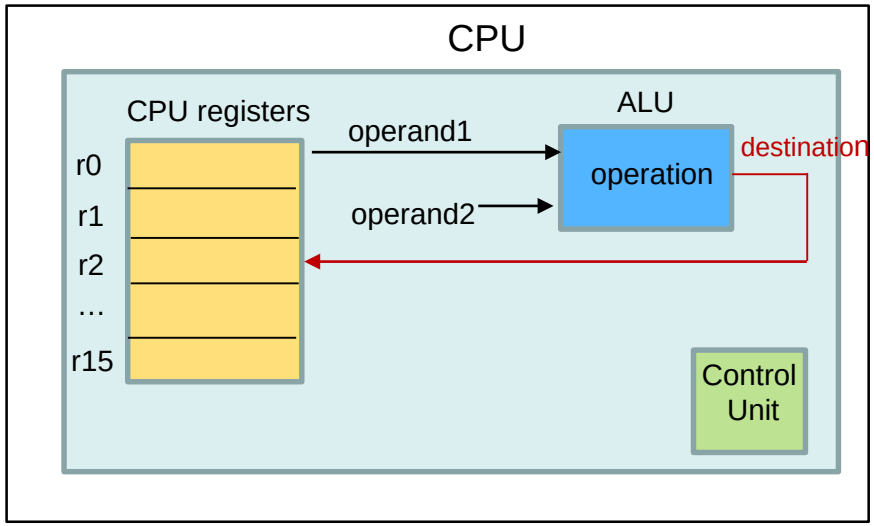




Simplified block diagram of a CPU

# Modern Computer Architecture
## Machine instructions

- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands
    - Opcode: specifies the operation that should be performed by the ALU. Examples of operations are addition, subtraction, moving data between CPU registers, …
    - Operands: inputs to the ALU: operand1 and operand 2. Operands can either be CPU registers or values.
- 32-bit machine instruction example



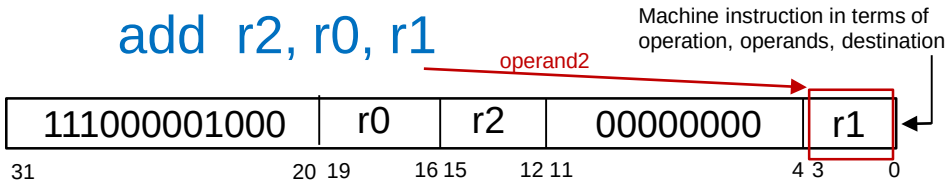



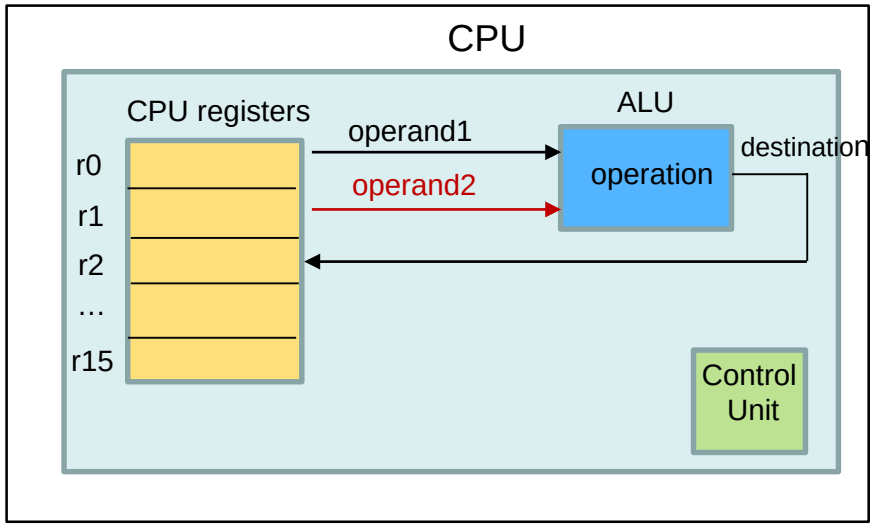

Simplified block diagram of a CPU

# Modern Computer Architecture
## Machine instructions

- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands
    - Opcode: specifies the operation that should be performed by the ALU. Examples of operations are addition, subtraction, moving data between CPU registers, …
    - Operands: inputs to the ALU: operand1 and operand 2. Operands can either be CPU registers or values.
- 32-bit machine instruction example

add r2, r0, r1



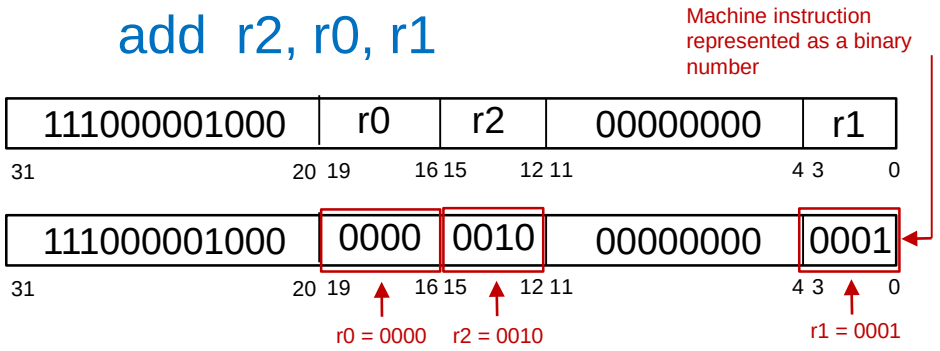



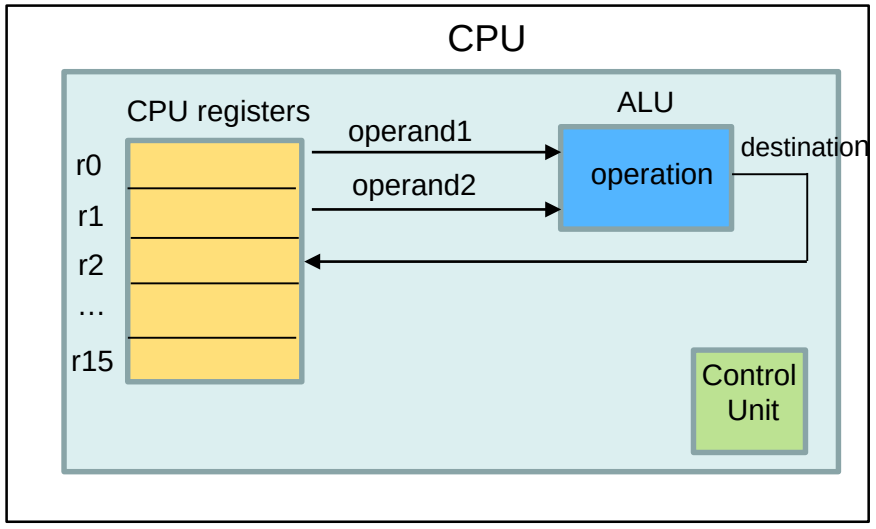

Simplified block diagram of a CPU

# Modern Computer Architecture
## Machine instructions

- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands
    - Opcode: specifies the operation that should be performed by the ALU. Examples of operations are addition, subtraction, moving data between CPU registers, …
    - Operands: inputs to the ALU: operand1 and operand 2. Operands can either be CPU registers or values.
- 32-bit machine instruction example

add r2, r0, r1



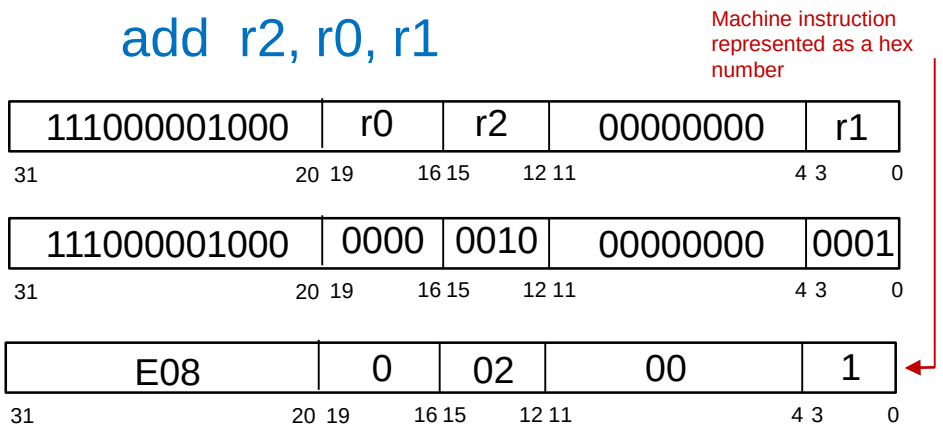



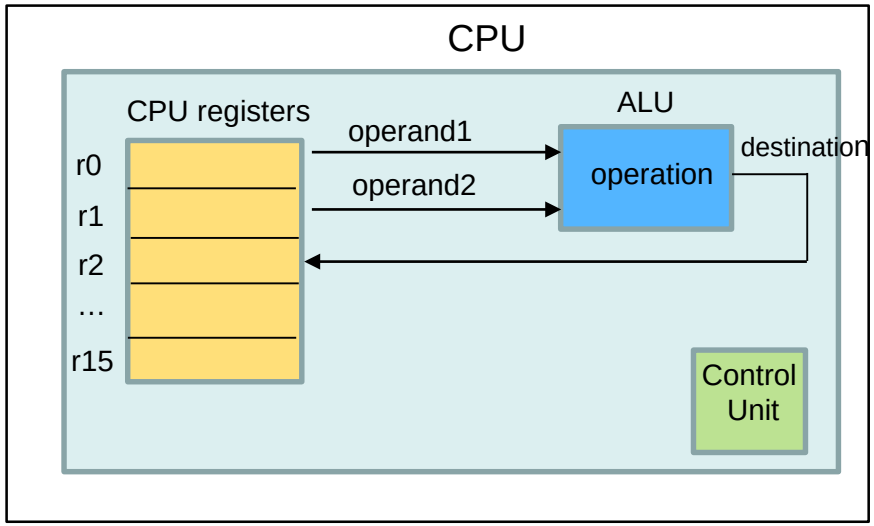

Simplified block diagram of a CPU

# Modern Computer Architecture
## Machine instructions

- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands
    - Opcode: specifies the operation that should be performed by the ALU. Examples of operations are addition, subtraction, moving data between CPU registers, …
    - Operands: inputs to the ALU: operand1 and operand 2. Operands can either be CPU registers or values.
- 32-bit machine instruction example

  add r2, r0, r1



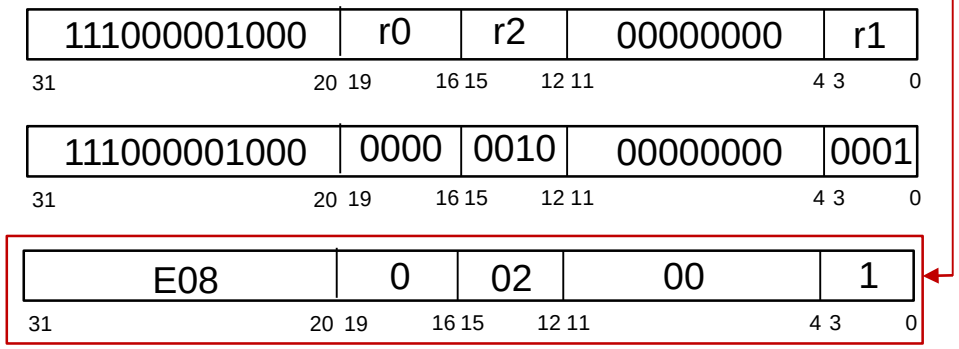

Simplified block diagram of a CPU

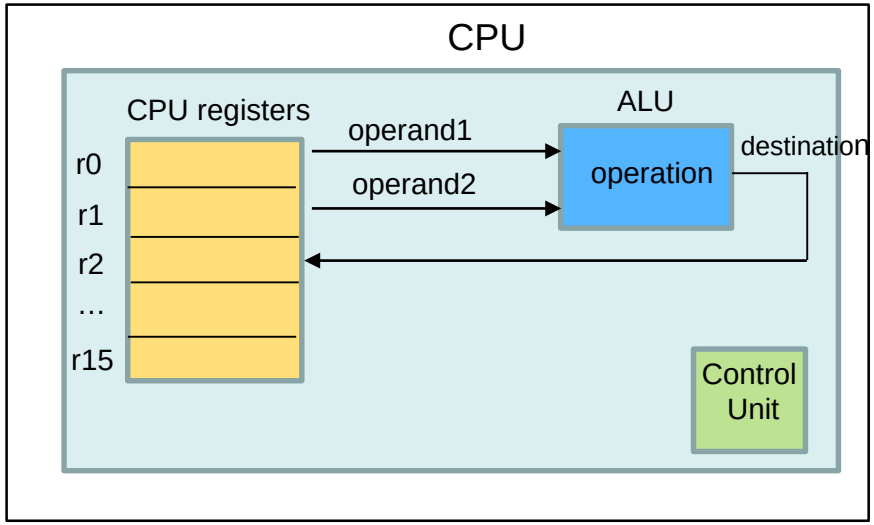
# Modern Computer Architecture
## Machine instructions

- What is a machine instruction?
  - A machine instruction is a specific task executed by a computer's CPU
- What are the components of a machine instruction?
  - A machine instruction is made up of an opcode and operands
    - Opcode: specifies the operation that should be performed by the ALU. Examples of operations are addition, subtraction, moving data between CPU registers, …
    - Operands: inputs to the ALU: operand1 and operand 2. Operands can either be CPU registers or values.
- 32-bit machine instruction example

add r2, r0, r1

The machine instruction 0xE0800201 informs the CPU to add r0 and r1 and put the result into r2

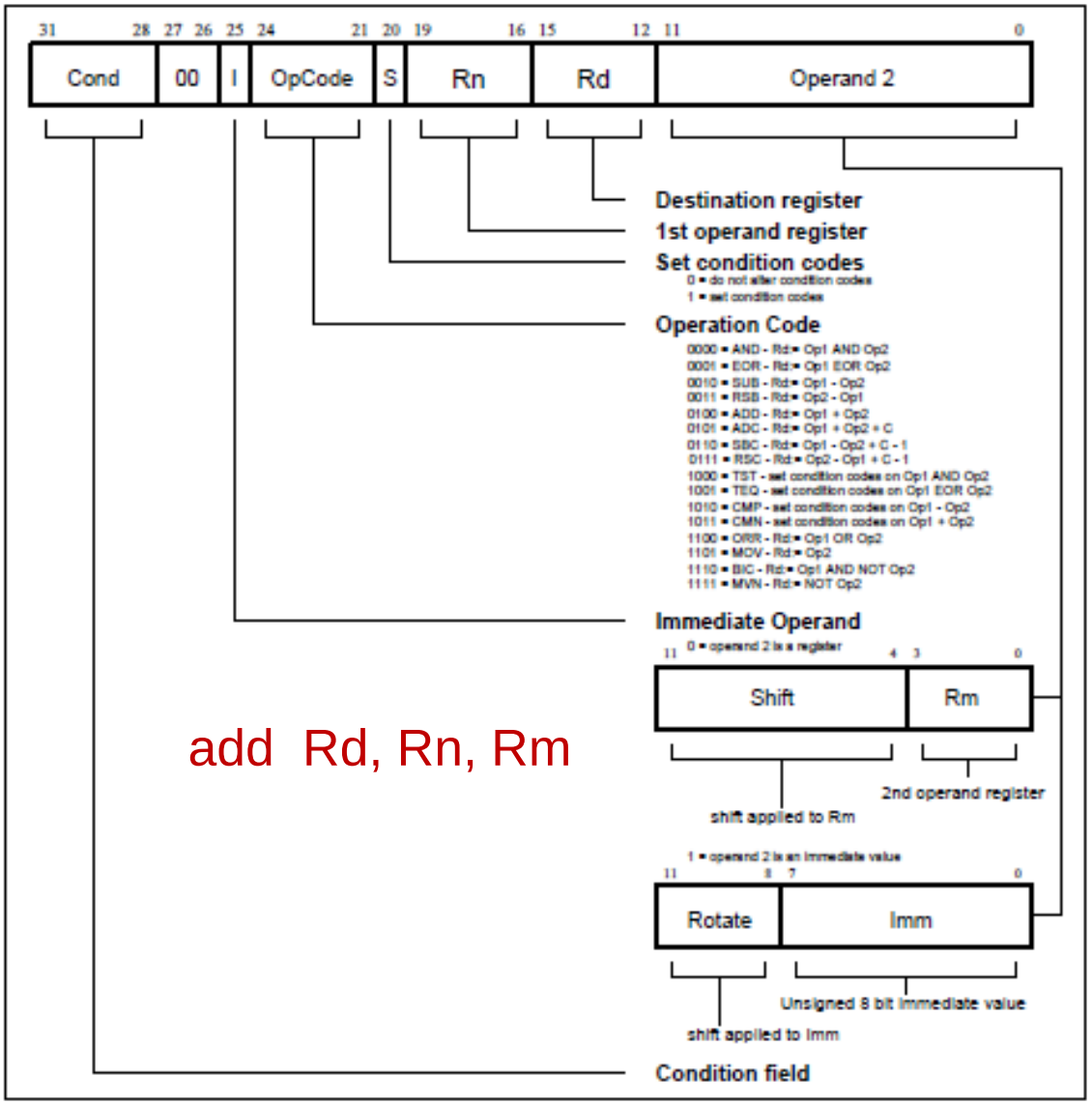



Simplified block diagram of a CPU

# Modern Computer Architecture

## Machine instructions



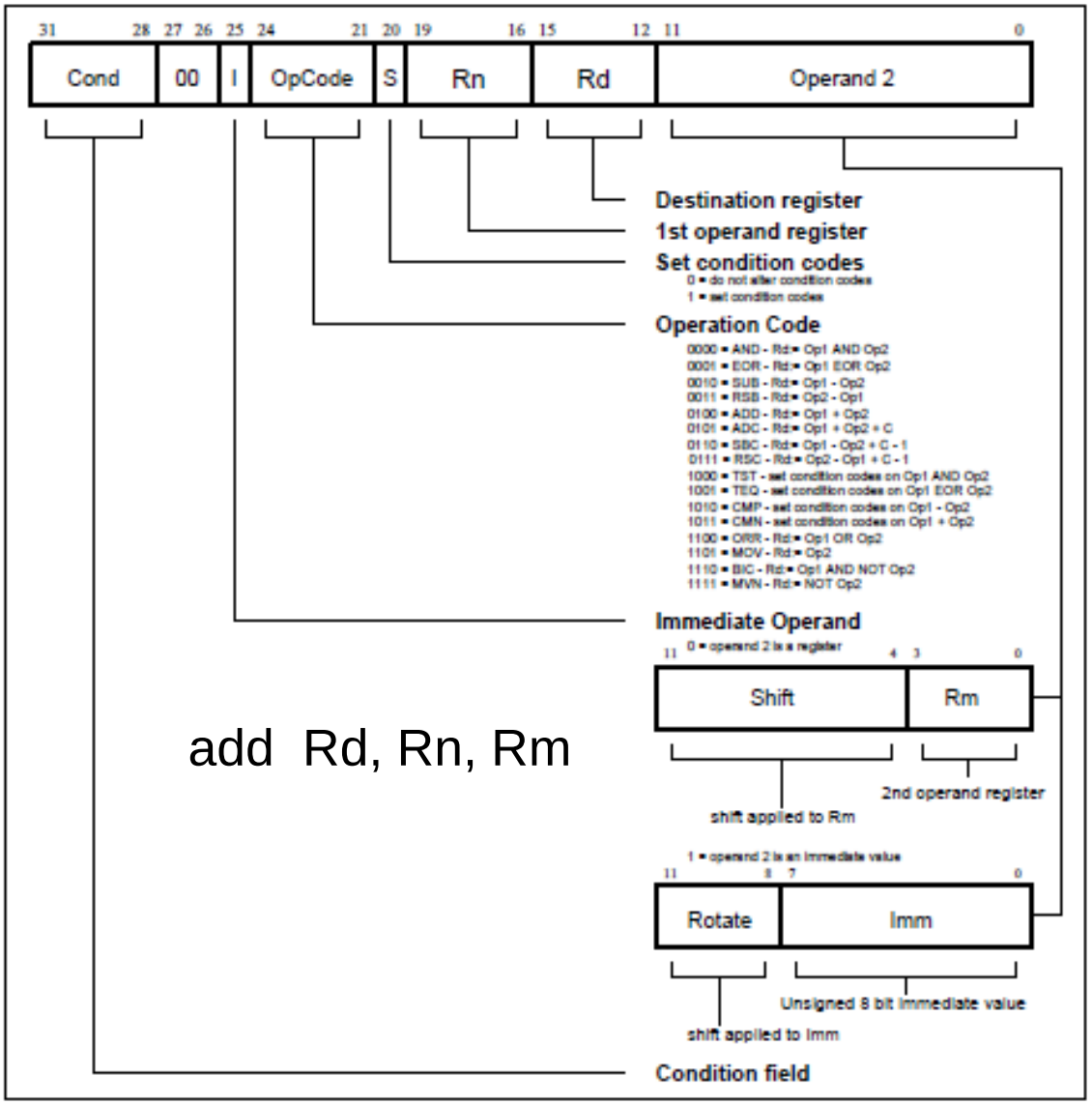

add Rd, Rn, Rm

General structure of the machine instructions for the ARM7 CPU for data processing operations.

Machine instructions for data processing operations – [Arm7 Data Sheet]

# Modern Computer Architecture

## Machine instructions




- 32-bit machine instruction example

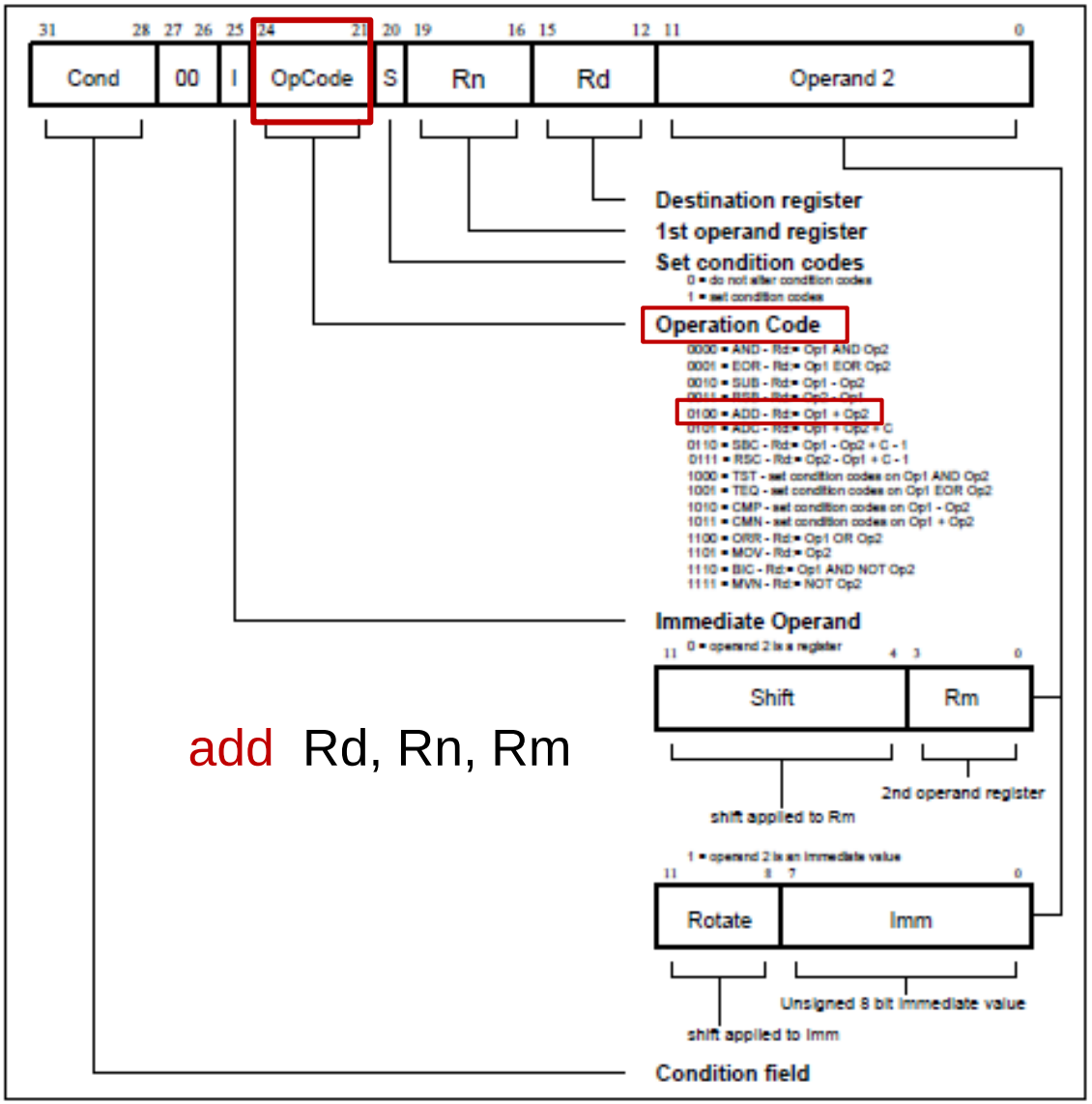
  add r2, r0, r1

General structure of the machine instructions for the ARM7 CPU for data processing operations.

Machine instructions for data processing operations – [Arm7 Data Sheet]

# Modern Computer Architecture

## Machine instructions




add Rd, Rn, Rm

- 32-bit machine instruction example

add r2, r0, r1

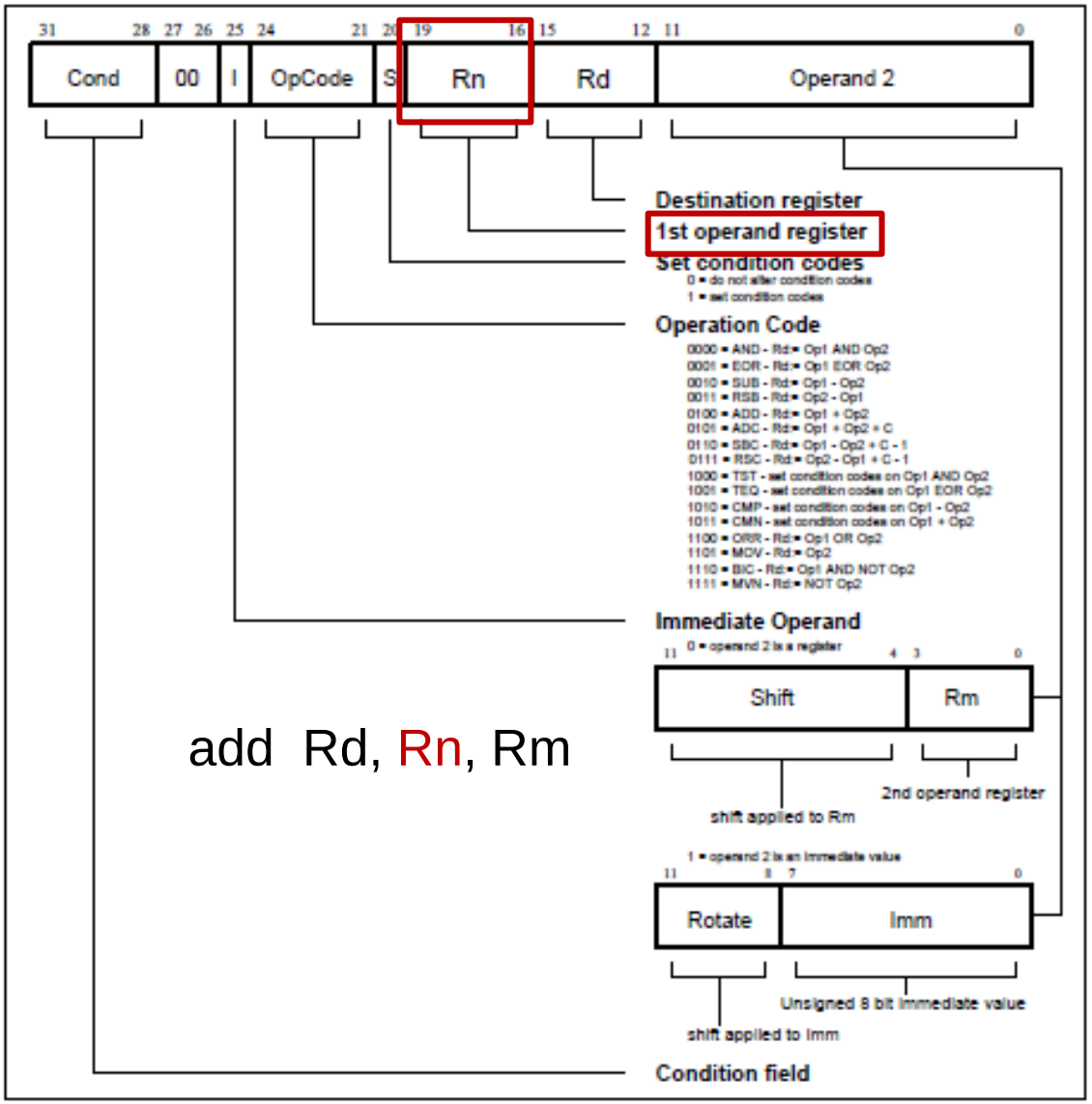
operation

General structure of the machine instructions for the ARM7 CPU for data processing operations.

Machine instructions for data processing operations – [Arm7 Data Sheet]

# Modern Computer Architecture

## Machine instructions




add Rd, Rn, Rm

- 32-bit machine instruction example

add r2, r0, r1

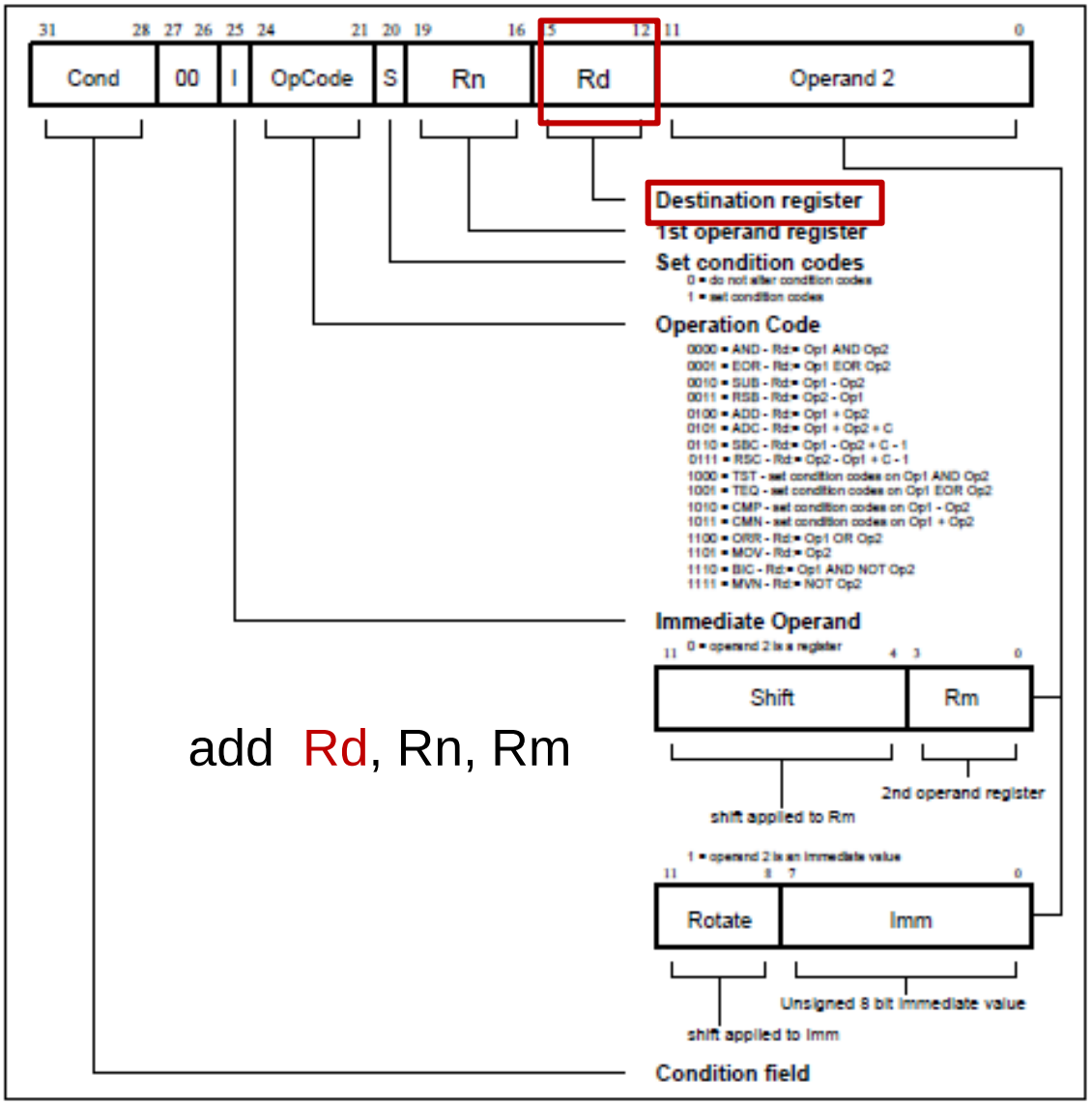
Rn = r0

General structure of the machine instructions for the ARM7 CPU for data processing operations.

Machine instructions for data processing operations – [Arm7 Data Sheet]

# Modern Computer Architecture

## Machine instructions




add Rd, Rn, Rm

- 32-bit machine instruction example

add r2, r0, r1

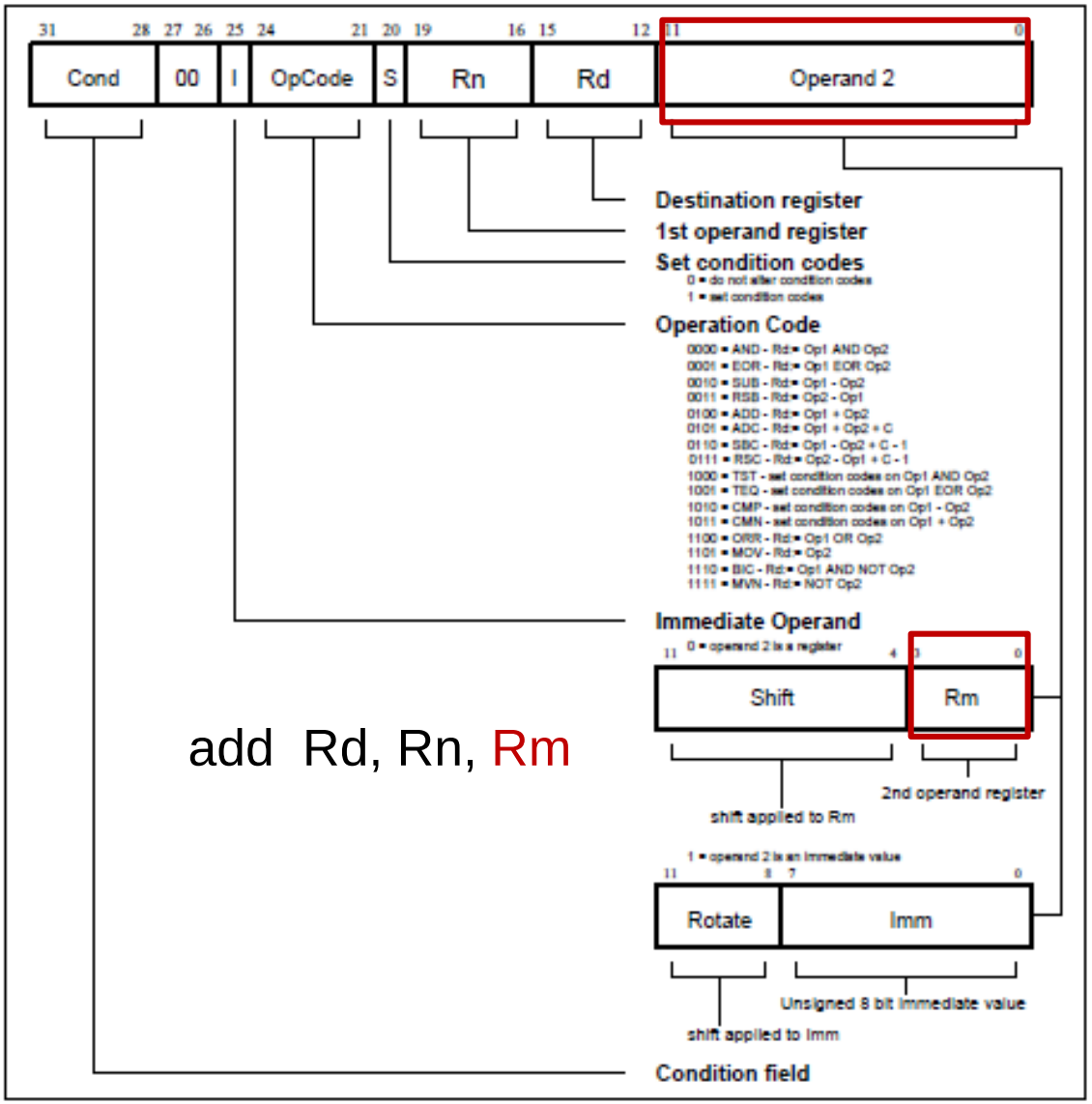
Rd = r2

General structure of the machine instructions for the ARM7 CPU for data processing operations.

Machine instructions for data processing operations – [Arm7 Data Sheet]

# Modern Computer Architecture

## Machine instructions




- 32-bit machine instruction example

  add r2, r0, r1

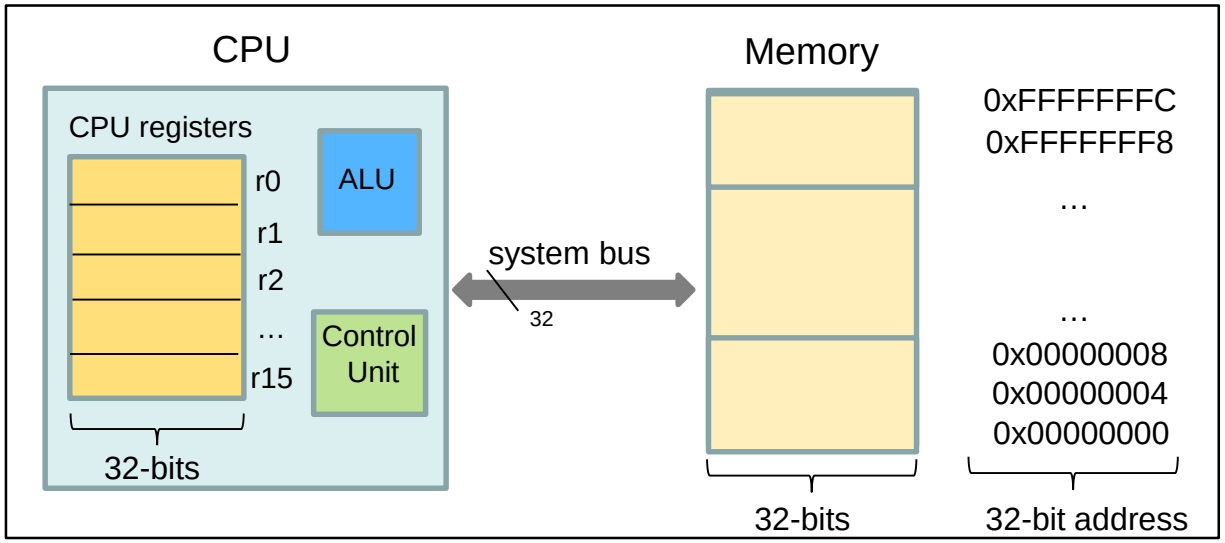
  Rm = r1

General structure of the machine instructions for the ARM7 CPU for data processing operations.

Machine instructions for data processing operations – [Arm7 Data Sheet]

# Modern Computer Architecture
## Assembly instructions

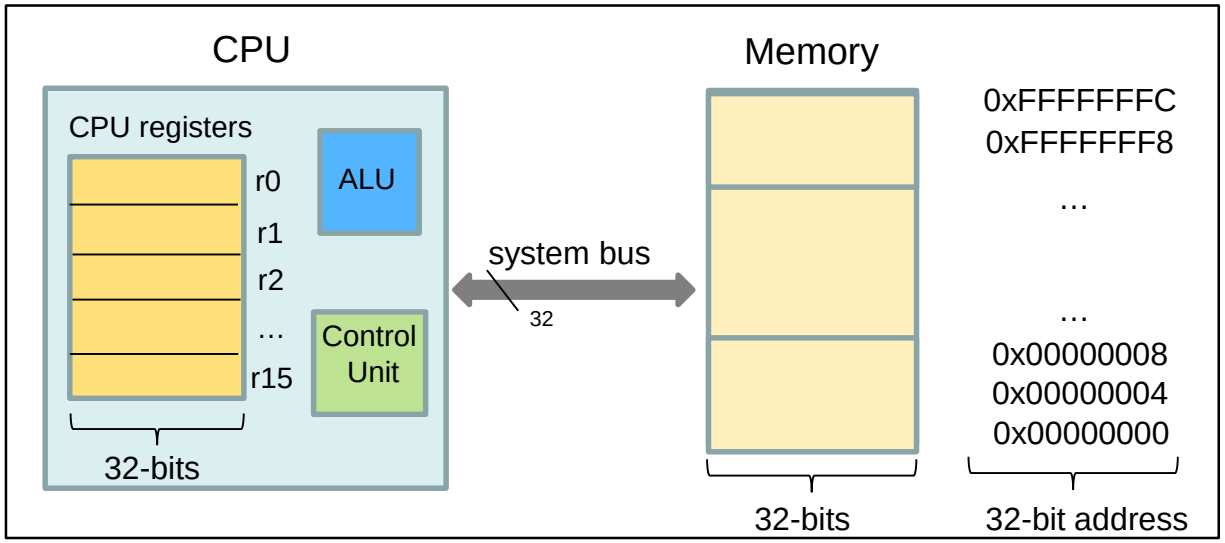
- It is tedious for programmers to code using machine instructions
- By assigning mnemonics to each opcode, and defining a convention to specify operands, code becomes easier to write and understand




Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- It is tedious for programmers to code using machine instructions
- By assigning mnemonics to each opcode, and defining a convention to specify operands, code becomes easier to write and understand
- Example: a machine instruction that moves immediate data into a CPU register uses “MOV” as the mnemonic for its opcode
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]

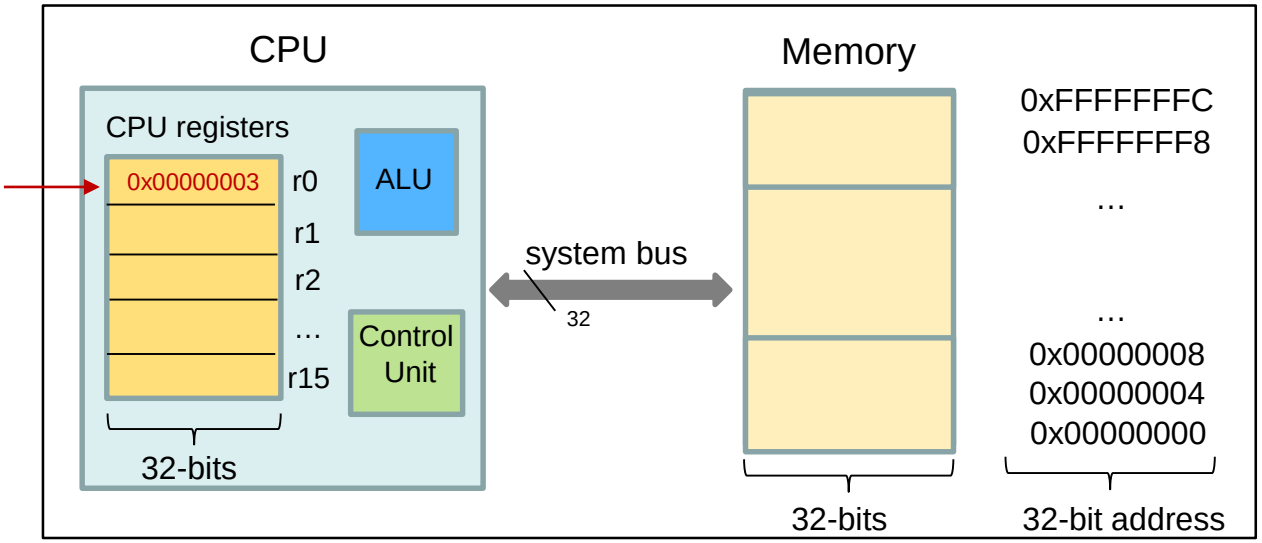



Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- It is tedious for programmers to code using machine instructions
- By assigning mnemonics to each opcode, and defining a convention to specify operands, code becomes easier to write and understand
- Example: a machine instruction that moves immediate data into a CPU register uses “MOV” as the mnemonic for its opcode
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]

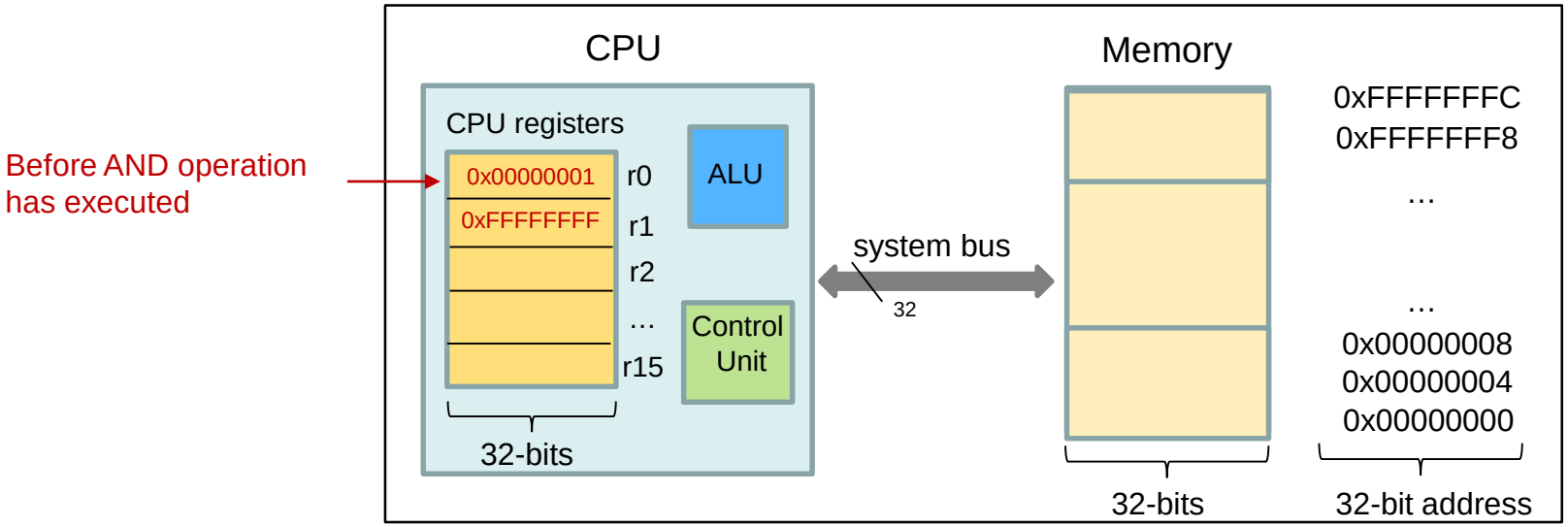
After the MOV operation has executed, the value 0x00000003 gets moved into r0




Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- Examples of other assembly instructions
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]
  - AND Rd, Rn, Rm  AND r2, r0, r1  [ r2 = r0 AND r1 ]

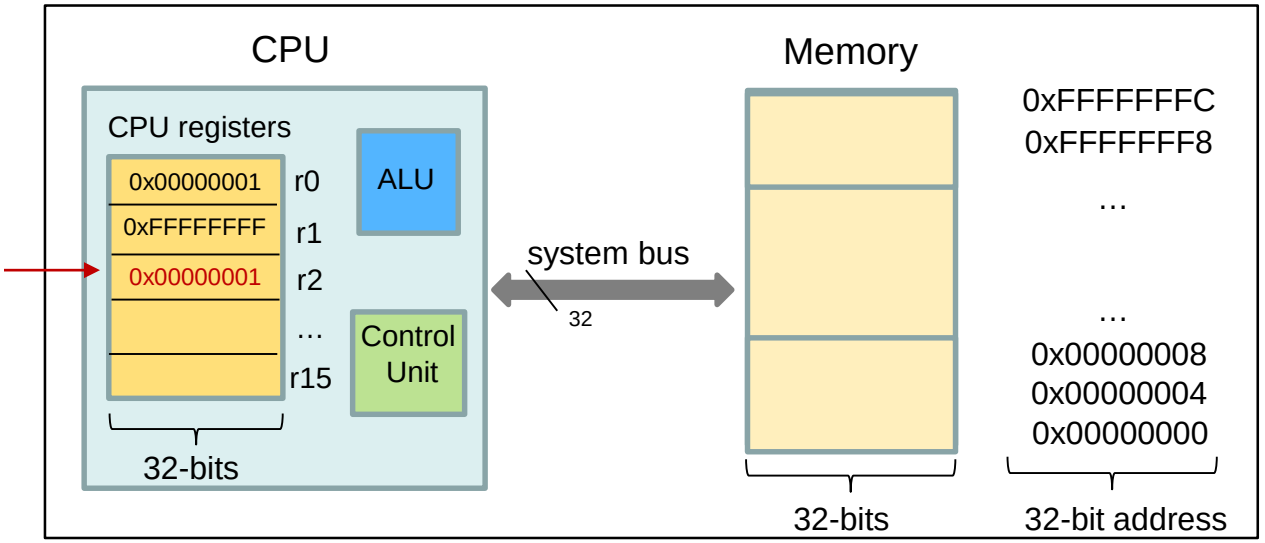



Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- Examples of other assembly instructions
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]
  - AND Rd, Rn, Rm  AND r2, r0, r1  [ r2 = r0 AND r1 ]

After AND operation has executed, the result is transferred to r2

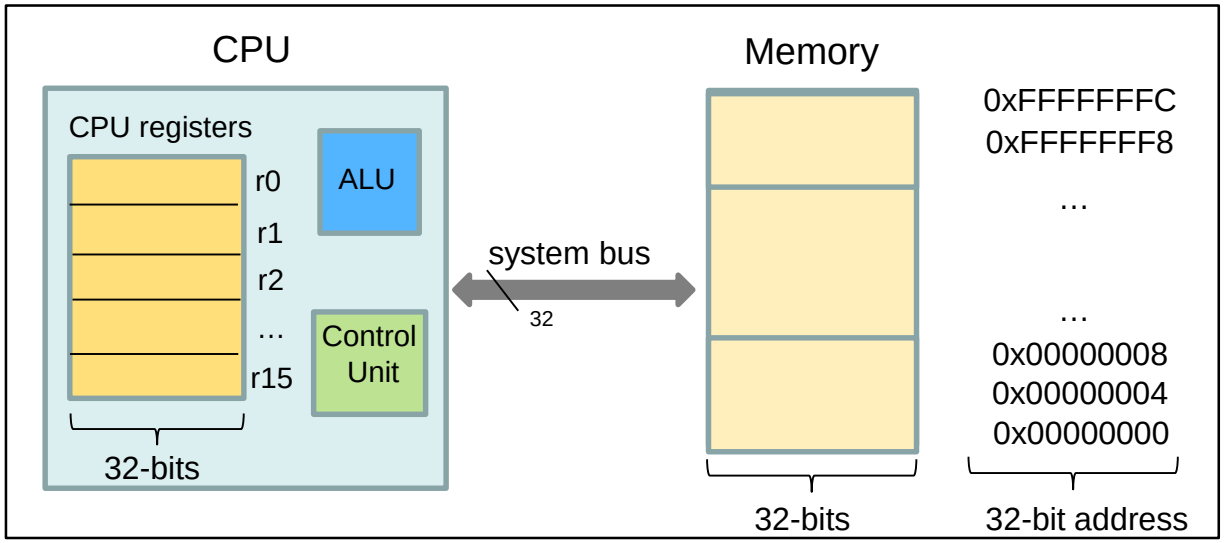



Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- Examples of other assembly instructions
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]
  - AND Rd, Rn, Rm  AND r2, r0, r1  [ r2 = r0 AND r1 ]
  - LDR Rd, [Rn]  LDR r1, [r0]  [ r1 = *r0: r0 contains a memory address. Load the value inside this memory address into r1 ]

LDR : instruction to LoaD a Register

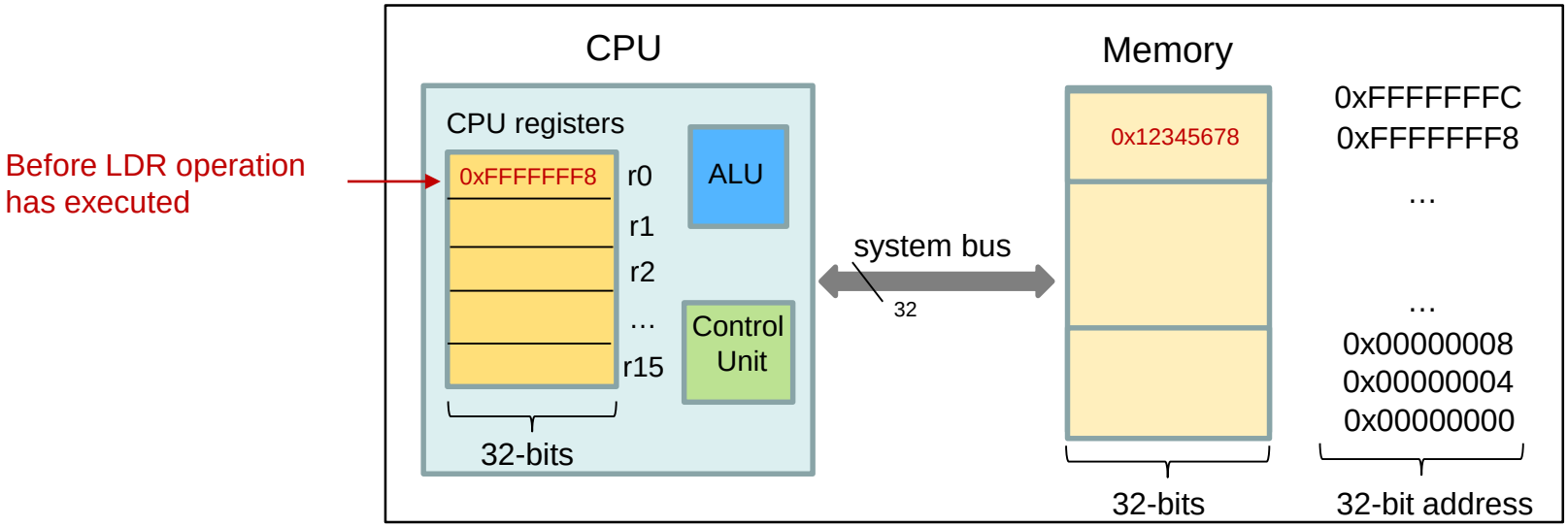



Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- Examples of other assembly instructions
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]
  - AND Rd, Rn, Rm  AND r2, r0, r1  [ r2 = r0 AND r1 ]
  - LDR Rd, [Rn]  LDR r1, [r0]  [ r1 = *r0: r0 contains a memory address. Load the value inside this memory address into r1 ]

LDR : instruction to LoaD a Register



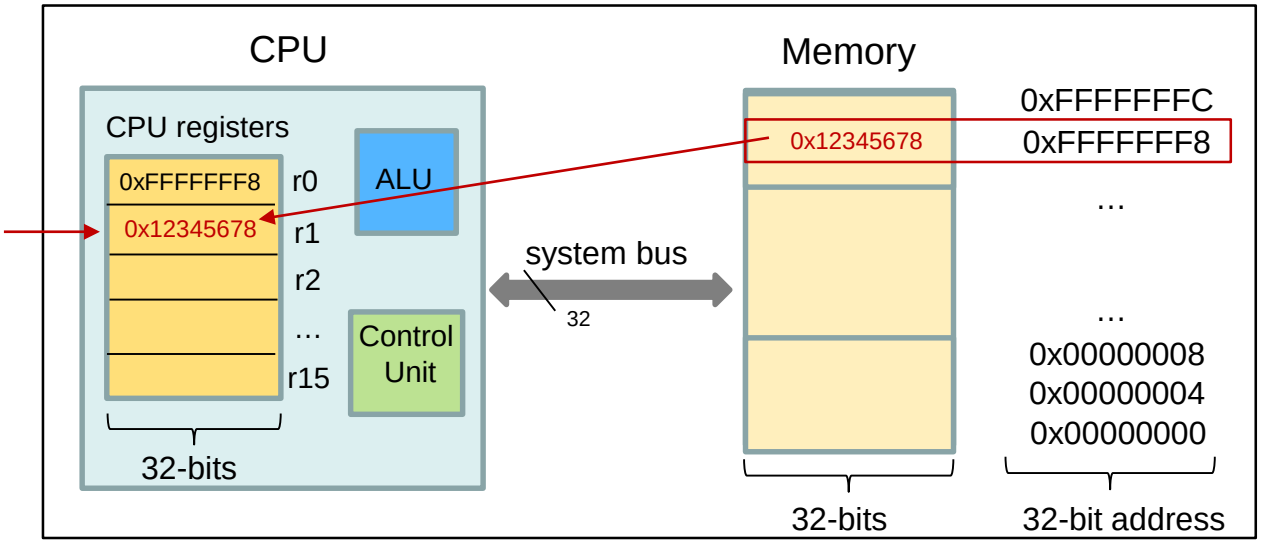

Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- Examples of other assembly instructions
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]
  - AND Rd, Rn, Rm  AND r2, r0, r1  [ r2 = r0 AND r1 ]
  - LDR Rd, [Rn]  LDR r1, [r0]  [ r1 = *r0: r0 contains a memory address. Load the value inside this memory address into r1 ]

LDR : instruction to LoaD a Register

After the LDR operation, the contents inside the address specified by r0 is copied to r1



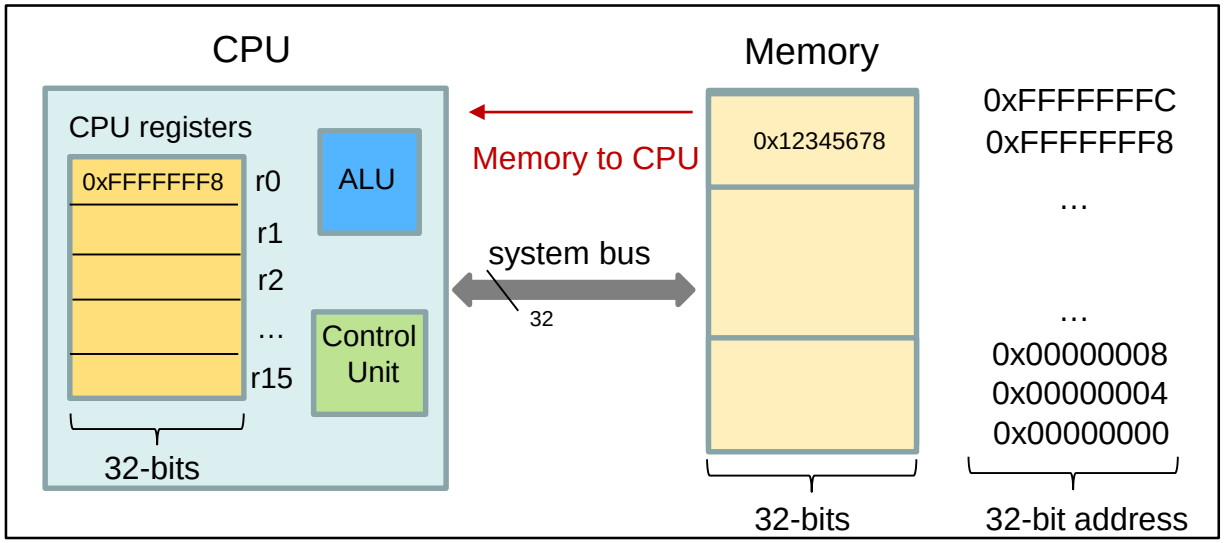

Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- Examples of other assembly instructions
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]
  - AND Rd, Rn, Rm  AND r2, r0, r1  [ r2 = r0 AND r1 ]
  - LDR Rd, [Rn]  LDR r1, [r0]  [ r1 = *r0: r0 contains a memory address. Load the value inside this memory address into r1 ]

Copy data from Memory to the CPU

LDR : instruction to LoaD a Register

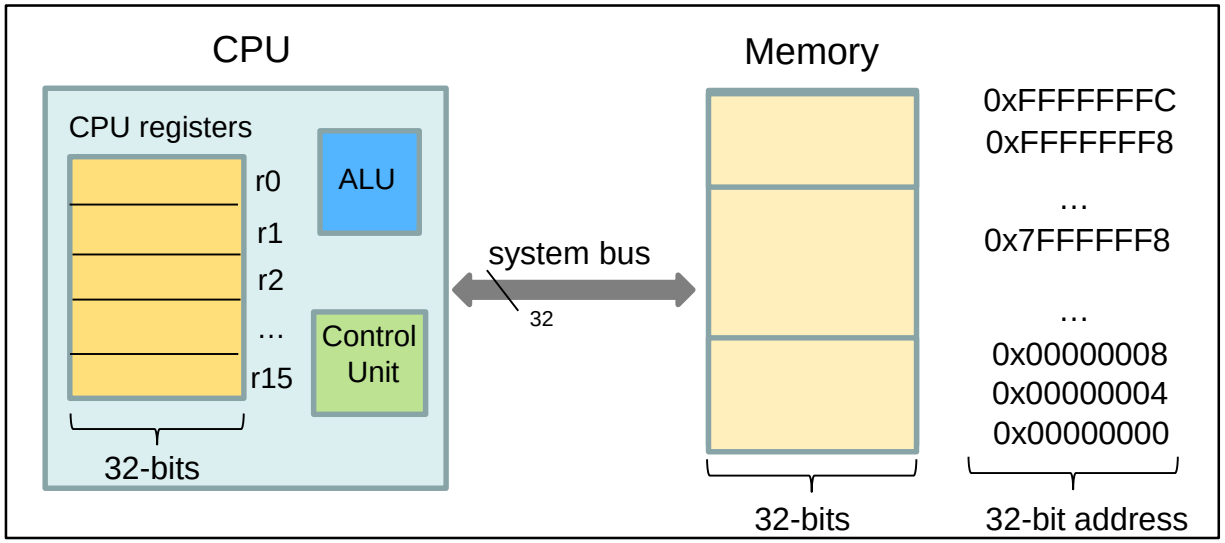



Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- Examples of other assembly instructions
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]
  - AND Rd, Rn, Rm  AND r2, r0, r1  [ r2 = r0 AND r1 ]
  - LDR Rd, [Rn]  LDR r1, [r0]  [ r1 = *r0: r0 contains a memory address. Load the value inside this memory address into r1 ]
  - STR Rn, [Rm]  STR r1, [r2]  [ *r2 = r1 : r2 contains a memory address. Store the contents of r1 into the memory address specified by r2]

STR : instruction to STore a Register

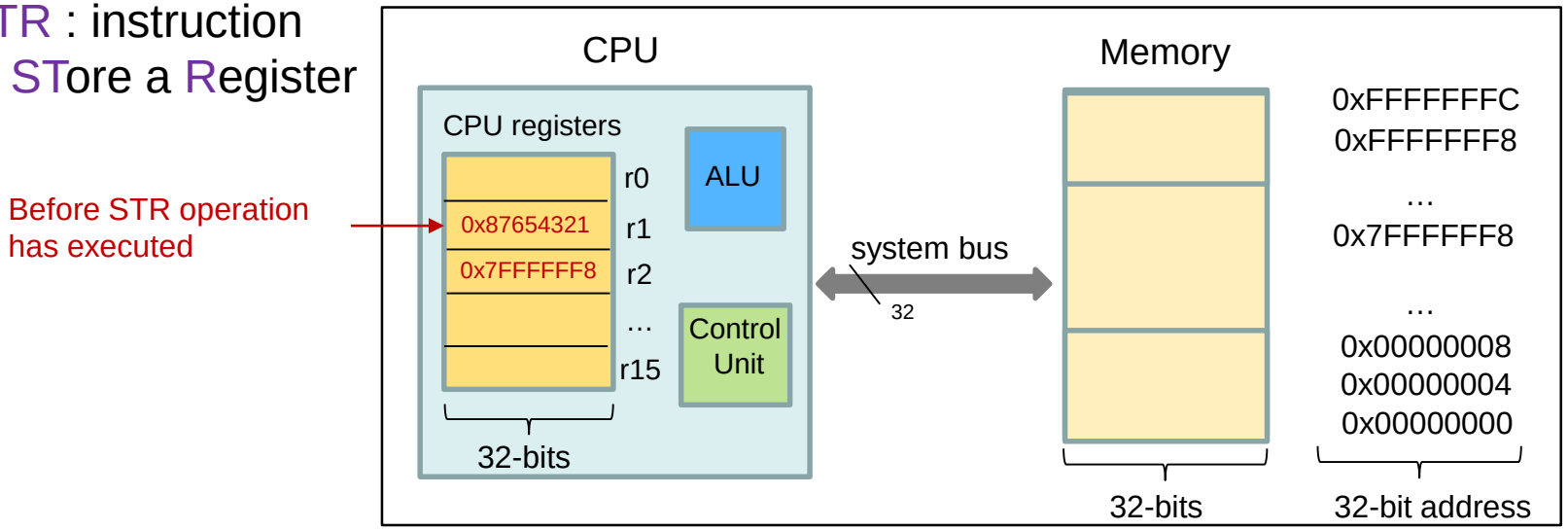



Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- Examples of other assembly instructions
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]
  - AND Rd, Rn, Rm  AND r2, r0, r1  [ r2 = r0 AND r1 ]
  - LDR Rd, [Rn]  LDR r1, [r0]  [ r1 = *r0: r0 contains a memory address. Load the value inside this memory address into r1 ]
  - STR Rn, [Rm]  STR r1, [r2]  [ *r2 = r1 : r2 contains a memory address. Store the contents of r1 into the memory address specified by r2]

STR : instruction to STore a Register



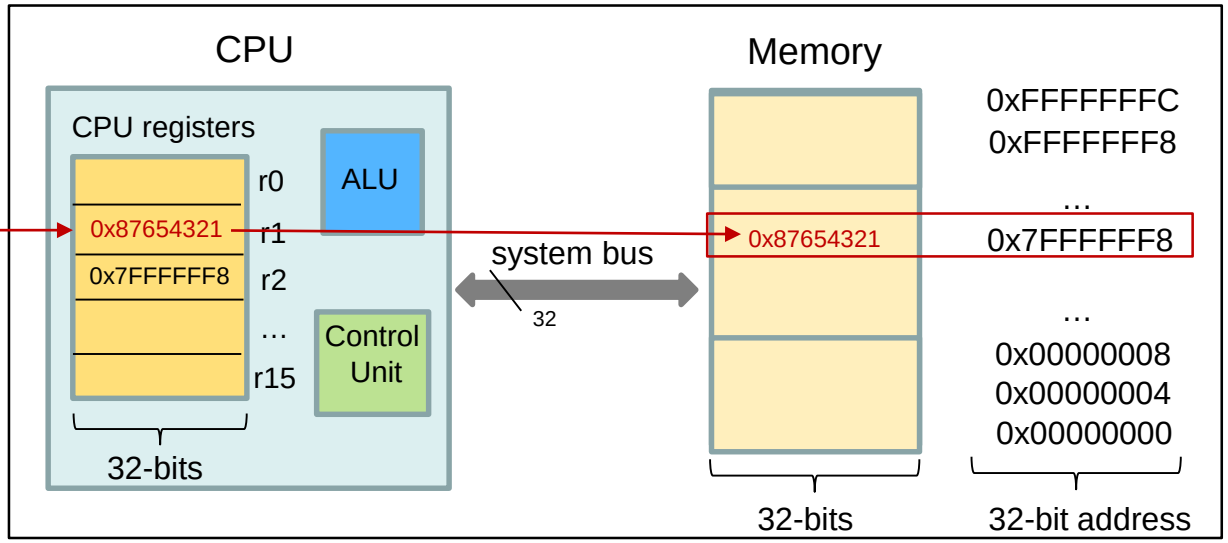

Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- Examples of other assembly instructions
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]
  - AND Rd, Rn, Rm  AND r2, r0, r1  [ r2 = r0 AND r1 ]
  - LDR Rd, [Rn]  LDR r1, [r0]  [ r1 = *r0: r0 contains a memory address. Load the value inside this memory address into r1 ]
  - STR Rn, [Rm]  STR r1, [r2]  [ *r2 = r1 : r2 contains a memory address. Store the contents of r1 into the memory address specified by r2]

STR : instruction to STore a Register

After STR operation has executed, the contents of r1 is copied into the memory address specified by r2



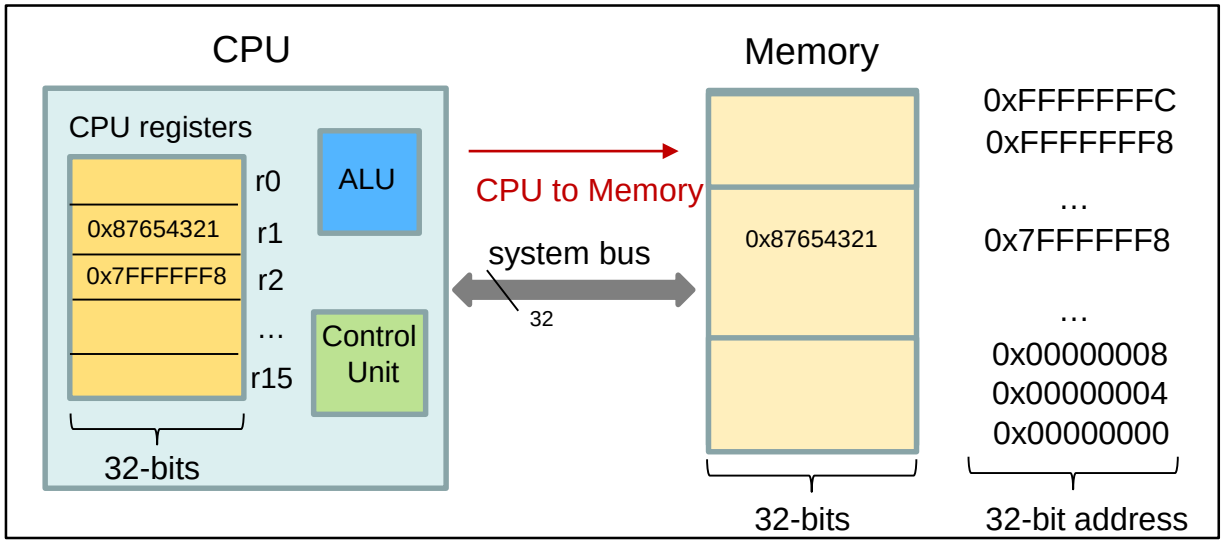

Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- Examples of other assembly instructions
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]
  - AND Rd, Rn, Rm  AND r2, r0, r1  [ r2 = r0 AND r1 ]
  - LDR Rd, [Rn]  LDR r1, [r0]  [ r1 = *r0: r0 contains a memory address. Load the value inside this memory address into r1 ]
  - STR Rn, [Rm]  STR r1, [r2]  [ *r2 = r1 : r2 contains a memory address. Store the contents of r1 into the memory address specified by r2]

Copy data from CPU to Memory

STR : instruction to STore a Register



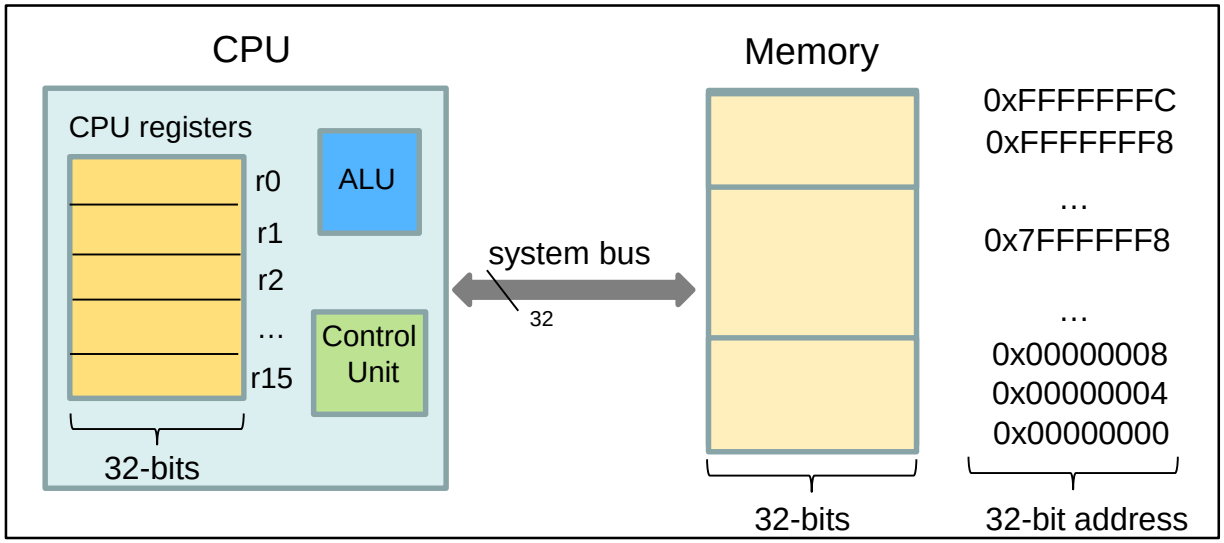

Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- Examples of other assembly instructions
  - MOV Rd, #Imm  MOV r0, #3  [ r0 = 3: Move the value 3 into r0 ]
  - AND Rd, Rn, Rm  AND r2, r0, r1  [ r2 = r0 AND r1 ]
  - LDR Rd, [Rn]  LDR r1, [r0]  [ r1 = *r0: r0 contains a memory address. Load the value inside this memory address into r1 ]
  - STR Rn, [Rm]  STR r1, [r2]  [ *r2 = r1 : r2 contains a memory address. Store the contents of r1 into the memory address specified by r2]
  - …

All the instructions of a CPU is called an **instruction set**

**Generally, machine instructions from one family of CPUs will not execute on another family of CPUs. Example: ARM and Intel**




Simplified block diagram of a modern computer

# Modern Computer Architecture
## Assembly instructions

- More ARM assembly instructions
  - SUB Rd, Rn, Rm — Subtraction — Rd = Rn – Rm
  - RSB Rd, Rn, Rm — Reverse Subtract — Rd = Rm - Rn
  - EOR Rd, Rn, Rm — Exclusive OR — Rd = Rn EOR Rm
  - ORR Rd, Rn, Rm — Logical OR — Rd = Rn OR Rm