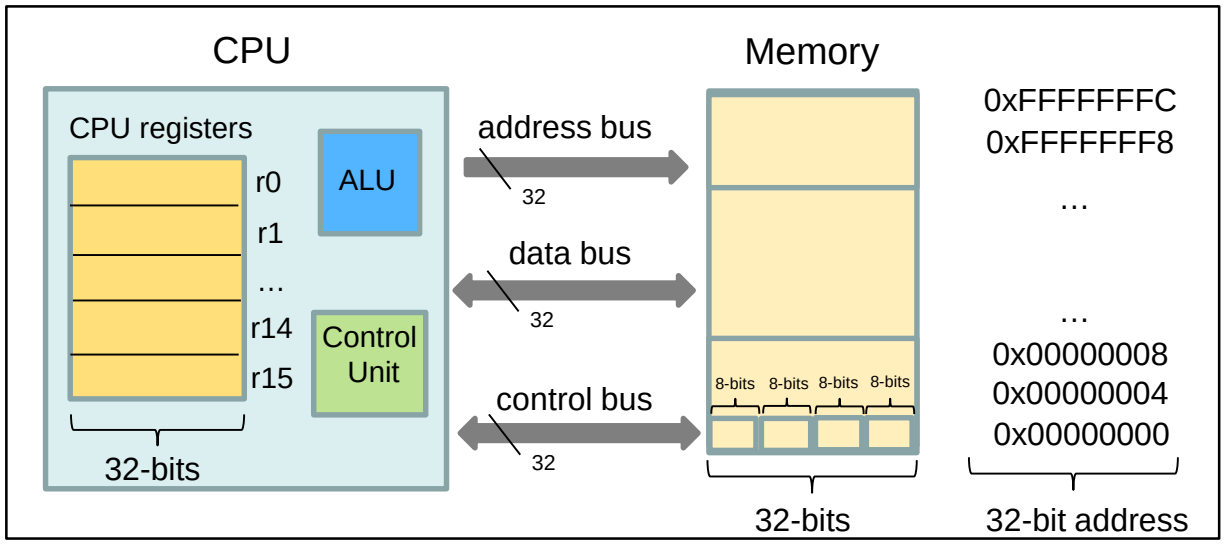
Modern Computer Architecture
Fundamental concepts 2

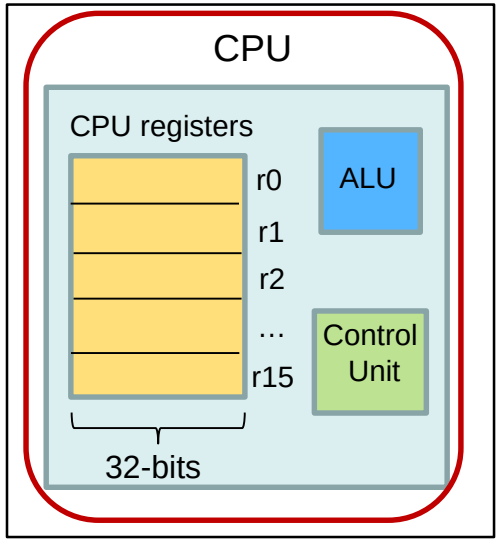


Simplified block diagram of a modern computer

RISC and CISC Architectures

Modern Computer Architecture
RISC vs CISC

- CPU architecture is classified into RISC and CISC
  - Reduced Instruction Set Computing (RISC)
  - Complex Instruction Set Computing (CISC)



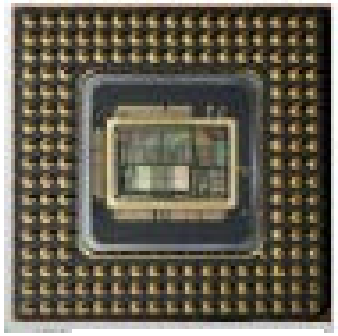
Simplified block diagram of a modern CPU

Modern Computer Architecture
RISC vs CISC

- CPU architecture is classified into RISC and CISC
  - Reduced Instruction Set Computing (RISC)
  - Complex Instruction Set Computing (CISC)

- Characteristics of CISC
  - Minimises the number of instructions per program. An entire program consists of few instructions and occupies less space in memory. Memory cost is reduced.
  - One CISC instruction performs many low level operations.
  - Instruction set has few operations

- Example of a CISC processors
  - Intel 80486: has 235 operations



Intel 80486

Modern Computer Architecture
RISC vs CISC

- CPU architecture is classified into RISC and CISC
  - Reduced Instruction Set Computing (RISC)
  - Complex Instruction Set Computing (CISC)

- Characteristics of RISC
  - Minimises the number of cycles for each operation or the time it takes each operation to complete
  - One RISC instruction, typically perform one low level operation
  - Instruction set has many operations
  - A program uses more memory space

Modern Computer Architecture
RISC vs CISC

- CPU architecture is classified into RISC and CISC
  - Reduced Instruction Set Computing (RISC)
  - Complex Instruction Set Computing (CISC)

- Characteristics of RISC
  - Minimises the number of cycles for each operation or the time it takes each operation to complete
  - One RISC instruction, typically perform one low level operation
  - Instruction set has many operations
  - A program uses more memory space

- Example of a RISC processors
  - Advanced RISC Machines (ARM)
  - Million Instructions Per Second (MIPS)
  - Intel Itanium
  - Scalable Processor ARChitecture (SPARC)

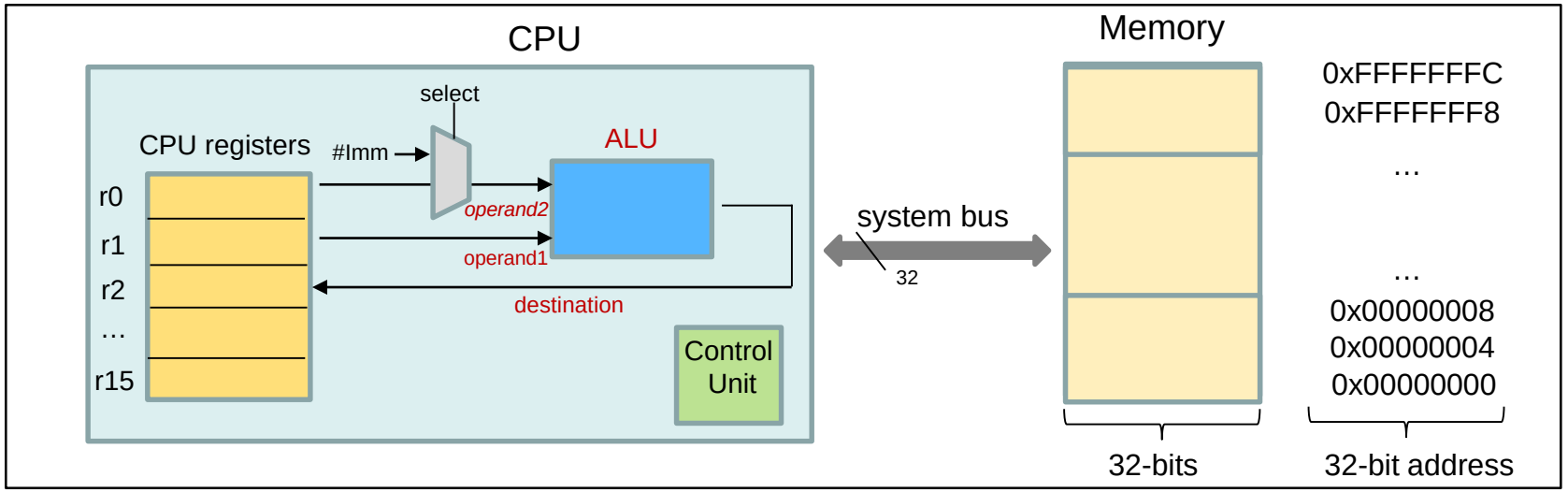
Modern Computer Architecture
RISC vs CISC: applications

- Where are CISC architectures used?
  - In systems where many floating-point Digital Signal Processing (DSP) operations need to be computed in real-time
  - Application areas: Radar, Communications, as well as Personal Computers so that software used in older PCs would continue to function.

- Where are RISC architectures used?
  - In embedded systems that consume low power. The processor can typically be put into a ‘sleep mode’ or a low power mode to reduce power consumption
  - Widely used in smartphones, tablets and gaming consoles, such as the Nintendo Wii, Microsoft Xbox 360, Sony Playstation 3, Nintendo DS

Modern Computer Architecture
RISC Architecture

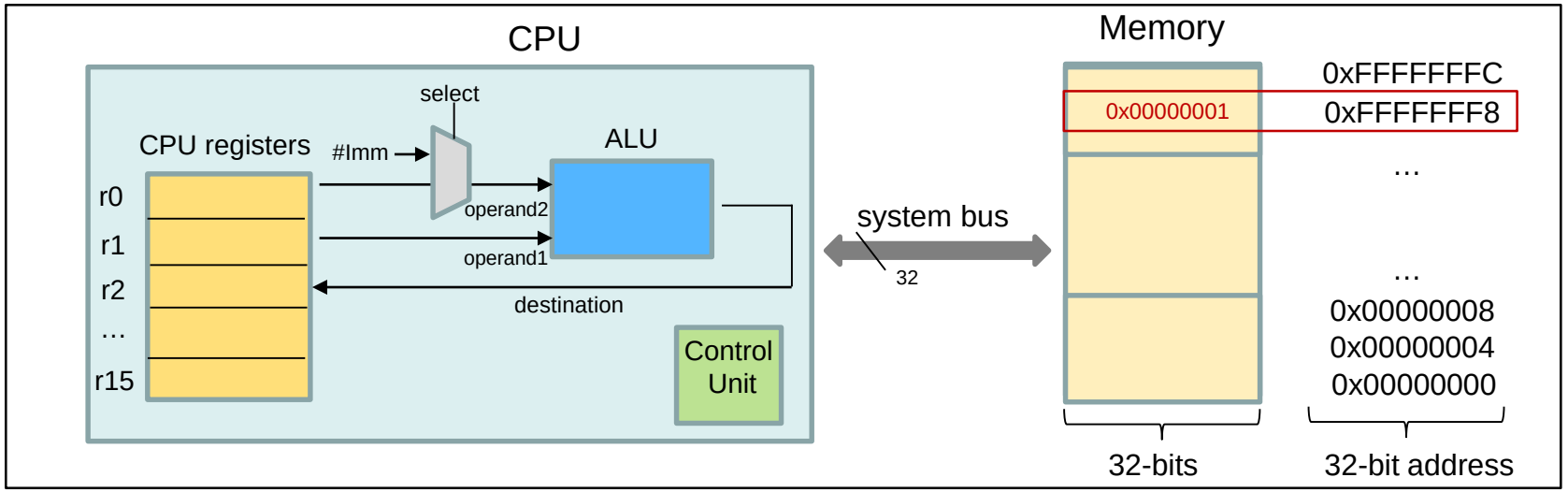
- Characteristics of RISC CPUs
  - The operands of the ALU can either be CPU registers or immediate values
  - The destination is a CPU register



Simplified, conceptual block diagram of a computer

Modern Computer Architecture
RISC Architecture

- What if operations need to be performed on values in memory?
  - Example: increment the value 0x00000001 in address 0xFFFFFFF8

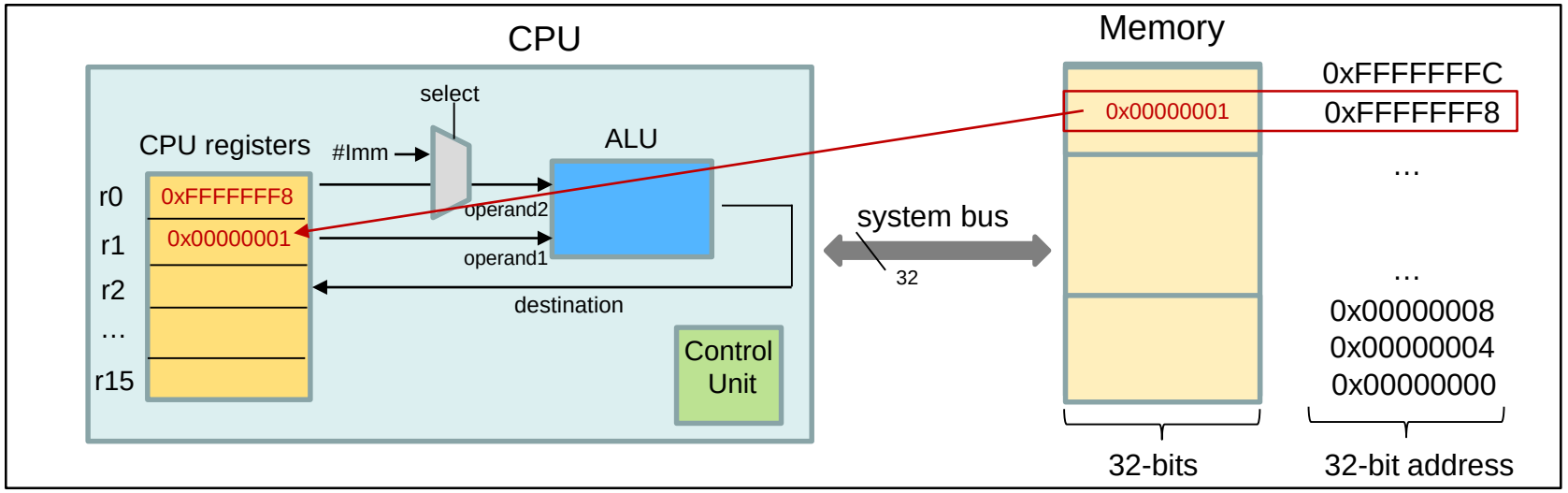


Simplified, conceptual block diagram of a computer

Modern Computer Architecture
RISC Architecture

- What if operations need to be performed on values in memory?
  - Example: increment the value 0x00000001 in address 0xFFFFFFF8
  - First, the contents of address 0xFFFFFFF8 should be transferred to a CPU register using a LoaD Register (LDR) operation

```
MOV  r0, #FFFFFFF8
LDR  r1, [r0]
```

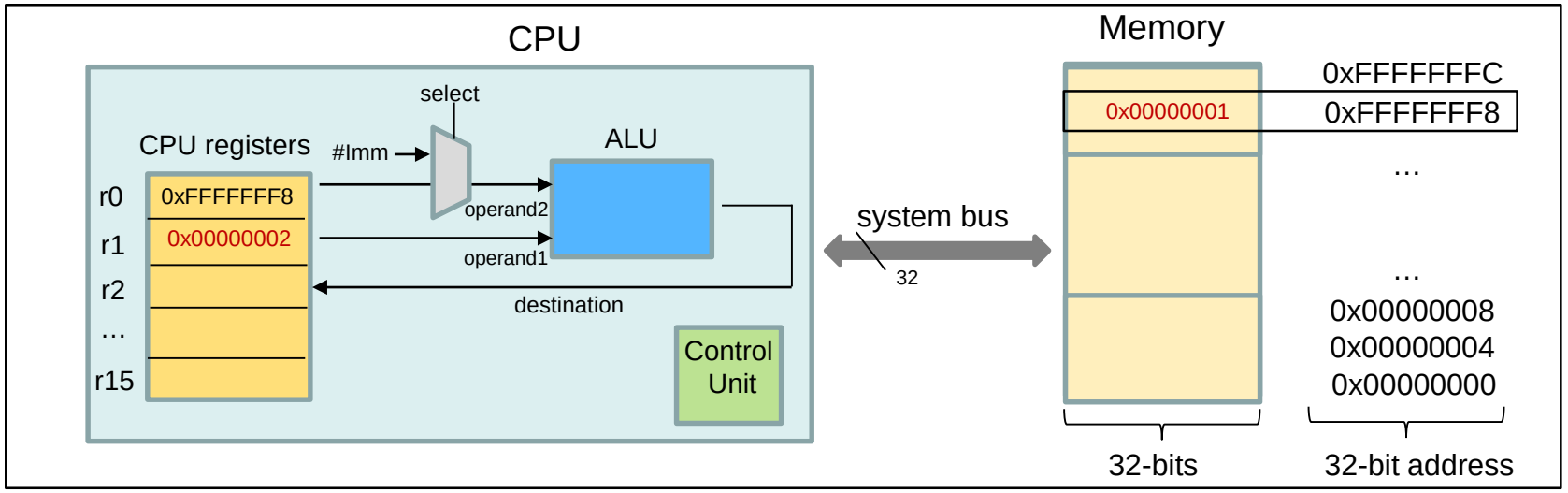


Simplified, conceptual block diagram of a computer

Modern Computer Architecture
RISC Architecture

- What if operations need to be performed on values in memory?
  - Example: increment the value 0x00000001 in address 0xFFFFFFF8
  - First, the contents of address 0xFFFFFFF8 should be transferred to a CPU register using a LoaD Register (LDR) operation
  - Then, the value of r1 should be incremented

```
MOV  r0, #FFFFFFF8
LDR  r1, [r0]
ADD  r1, r1, #1
```

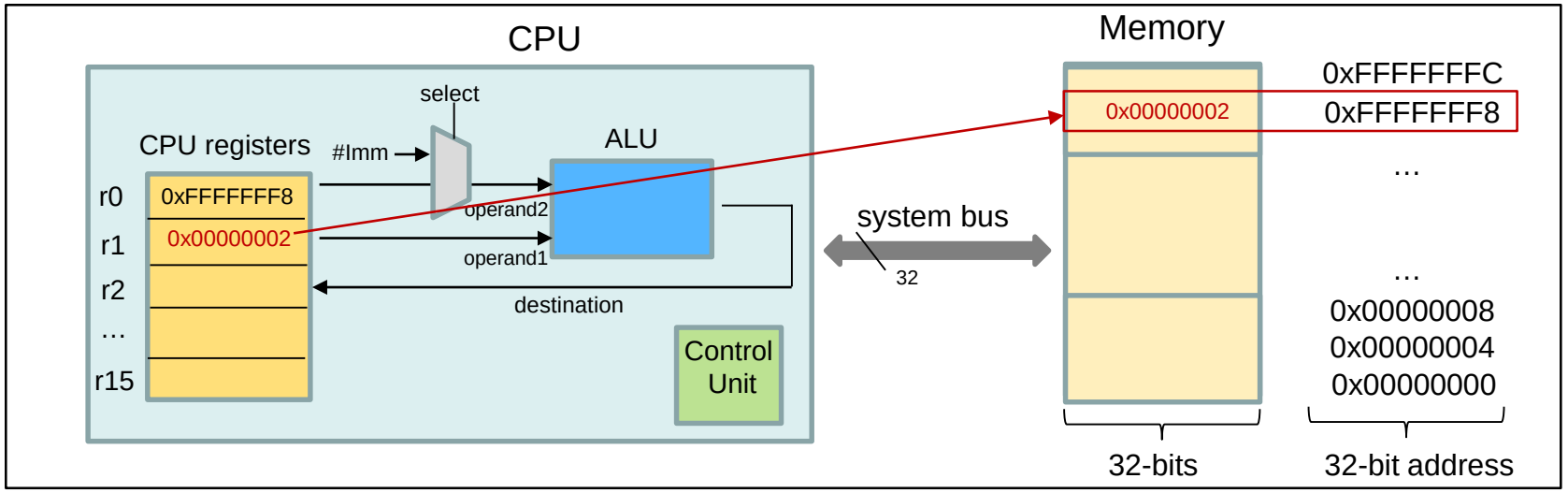


Simplified, conceptual block diagram of a computer

Modern Computer Architecture
RISC Architecture

- What if operations need to be performed on values in memory?
  - Example: increment the value 0x00000001 in address 0xFFFFFFF8
  - First, the contents of address 0xFFFFFFF8 should be transferred to a CPU register using a LoaD Register (LDR) operation
  - Then, the value of r1 should be incremented
  - Lastly, the computed value is transferred to memory using a STore Register (STR) operation

```
MOV  r0, #FFFFFFF8
LDR  r1, [r0]
ADD  r1, r1, #1
STR  r1, [r0]
```

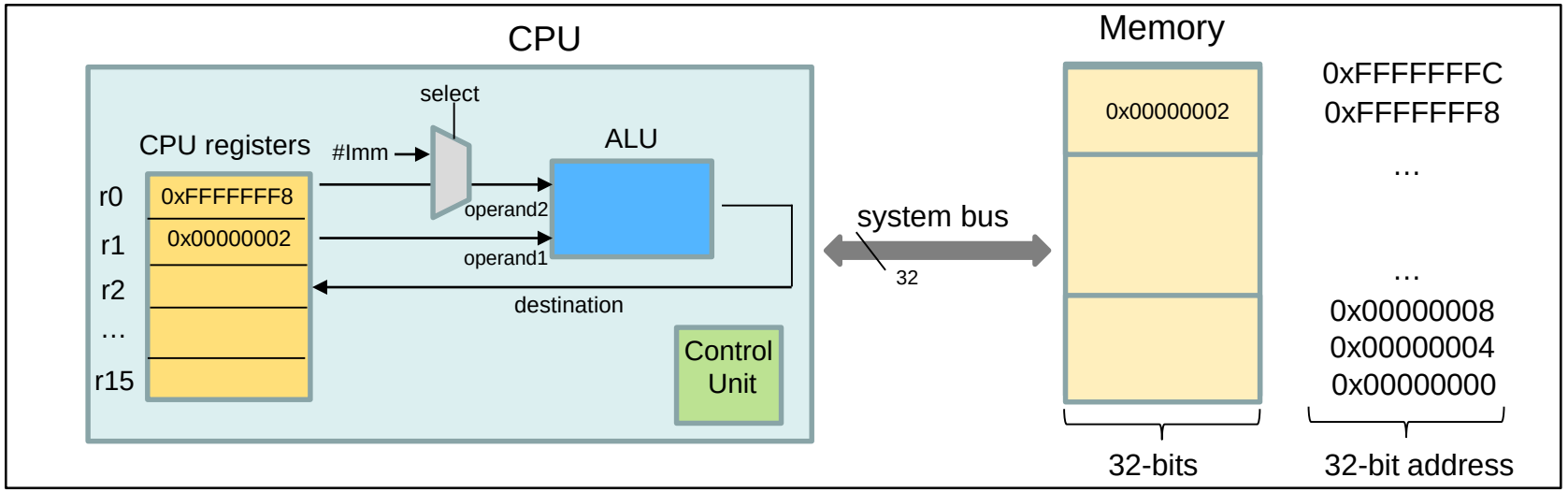


Simplified, conceptual block diagram of a computer

Modern Computer Architecture
RISC Architecture

- Fort this reason, RISC is referred to as a 'load-store' architecture
  - Data from memory must be loaded into the CPU registers in order for the ALU to operate on them

```
MOV  r0, #FFFFFFF8
LDR  r1, [r0]
ADD  r1, r1, #1
STR  r1, [r0]
```

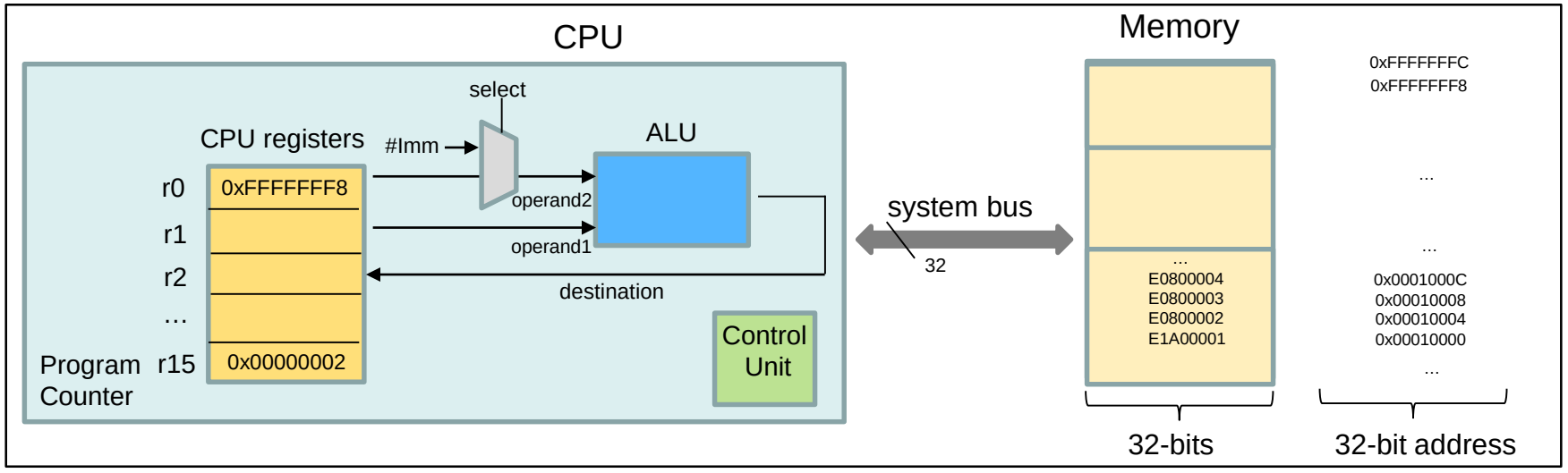


Simplified, conceptual block diagram of a computer

How are instructions processed?

Modern Computer Architecture
Processing of instructions of a program

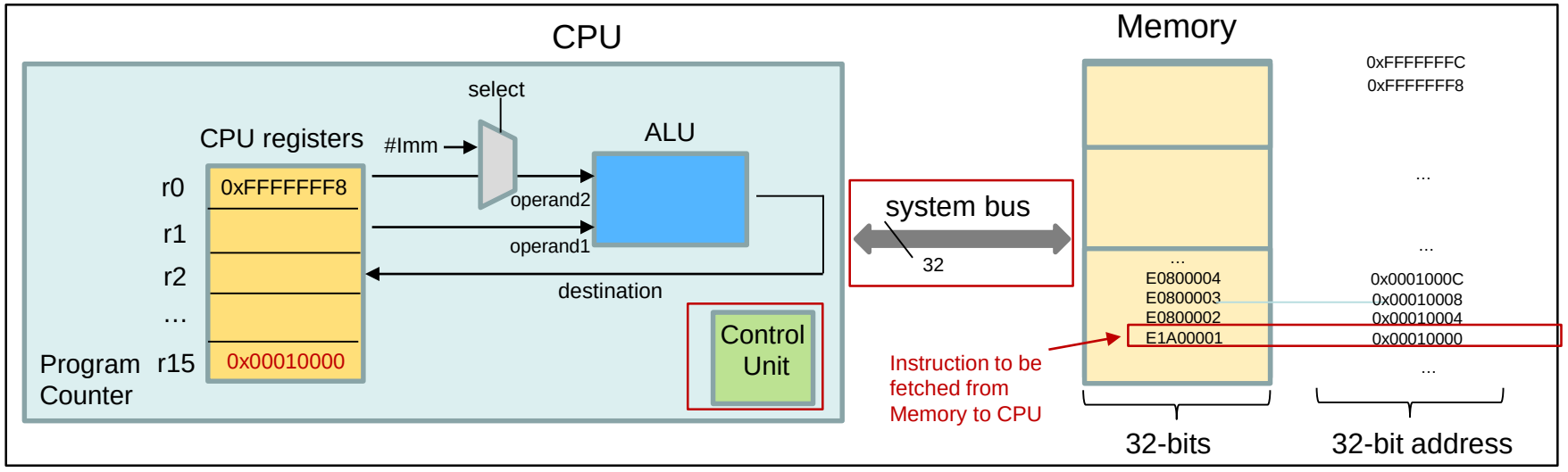
- A CPU implements a single machine instruction using three steps:
  1) Fetch:
  2) Decode:
  3) Execute:



Simplified, conceptual block diagram of a computer

Modern Computer Architecture
Processing of instructions of a program

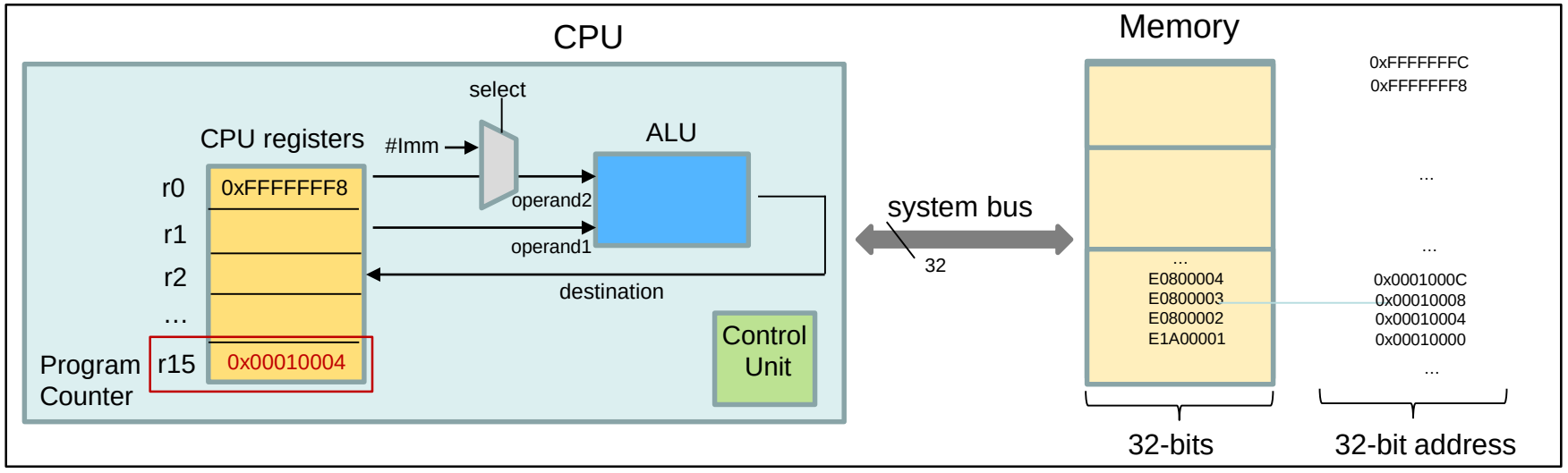
- A CPU implements a single machine instruction using three steps:
  1) Fetch: a machine instruction is fetched from Memory. CPU register r15, called the Program Counter (PC), contains the address of the instruction to be fetched.
  2) Decode:
  3) Execute:



Simplified, conceptual block diagram of a computer

Modern Computer Architecture
Processing of instructions of a program

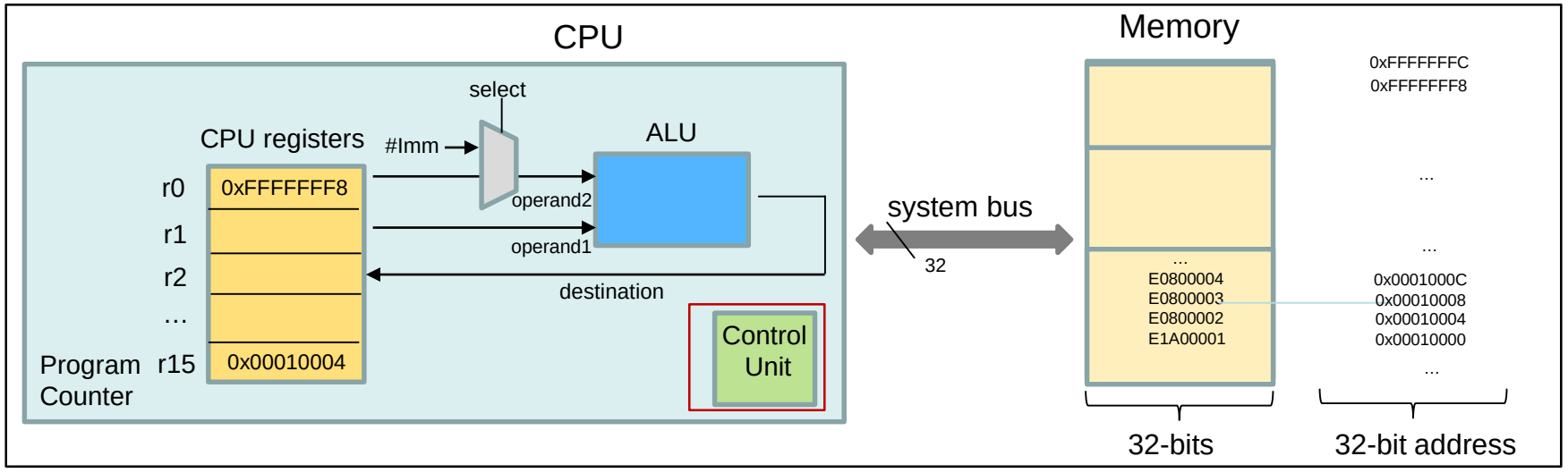
- A CPU implements a single machine instruction using three steps:
  1) Fetch: a machine instruction is fetched from Memory. CPU register r15, called the Program Counter (PC), contains the address of the instruction to be fetched. Thereafter, the PC is increased by four. The PC now contains the address of the next instruction to be fetched
  2) Decode:
  3) Execute:



Simplified, conceptual block diagram of a computer

Modern Computer Architecture
Processing of instructions of a program

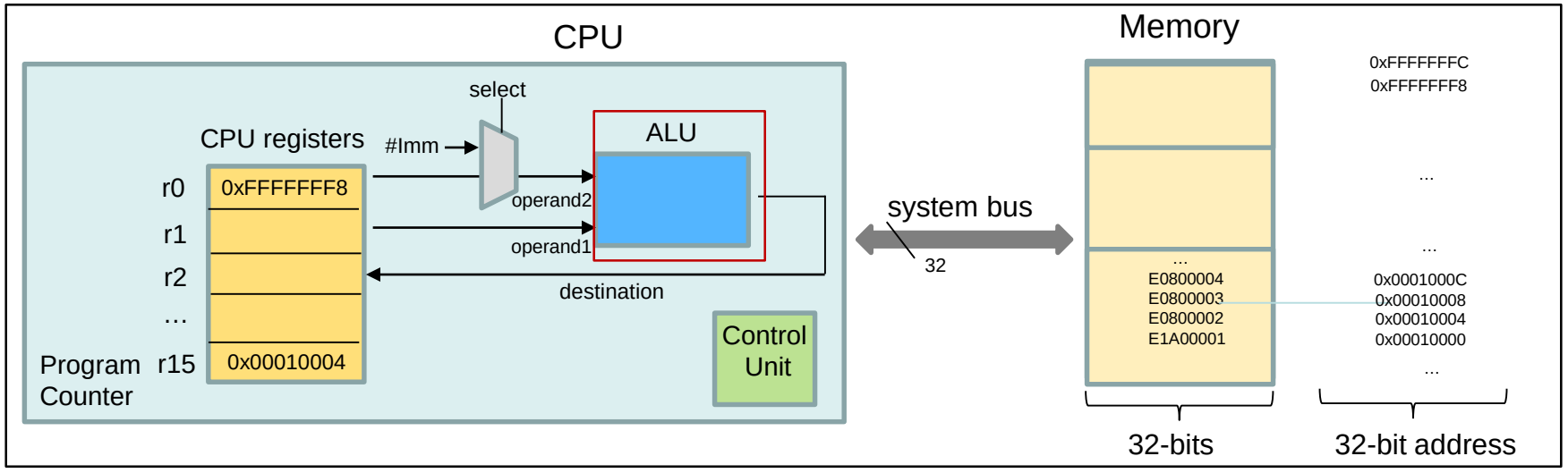
- A CPU implements a single machine instruction using three steps:
  1) Fetch: a machine instruction is fetched from Memory. CPU register r15, called the Program Counter (PC), contains the address of the instruction to be fetched. Thereafter, the PC is increased by four. The PC now contains the address of the next instruction to be fetched
  2) Decode: the machine instruction is broken down by the control unit to understand what operation needs to be done and the operands.
  3) Execute:



Simplified, conceptual block diagram of a computer

Modern Computer Architecture
Processing of instructions of a program

- A CPU implements a single machine instruction using three steps:
  1) Fetch: a machine instruction is fetched from Memory. CPU register r15, called the Program Counter (PC), contains the address of the instruction to be fetched. Thereafter, the PC is increased by four. The PC now contains the address of the next instruction to be fetched
  2) Decode: the machine instruction is broken down by the control unit to understand what operation needs to be done and the operands.
  3) Execute: the instruction is executed by the ALU

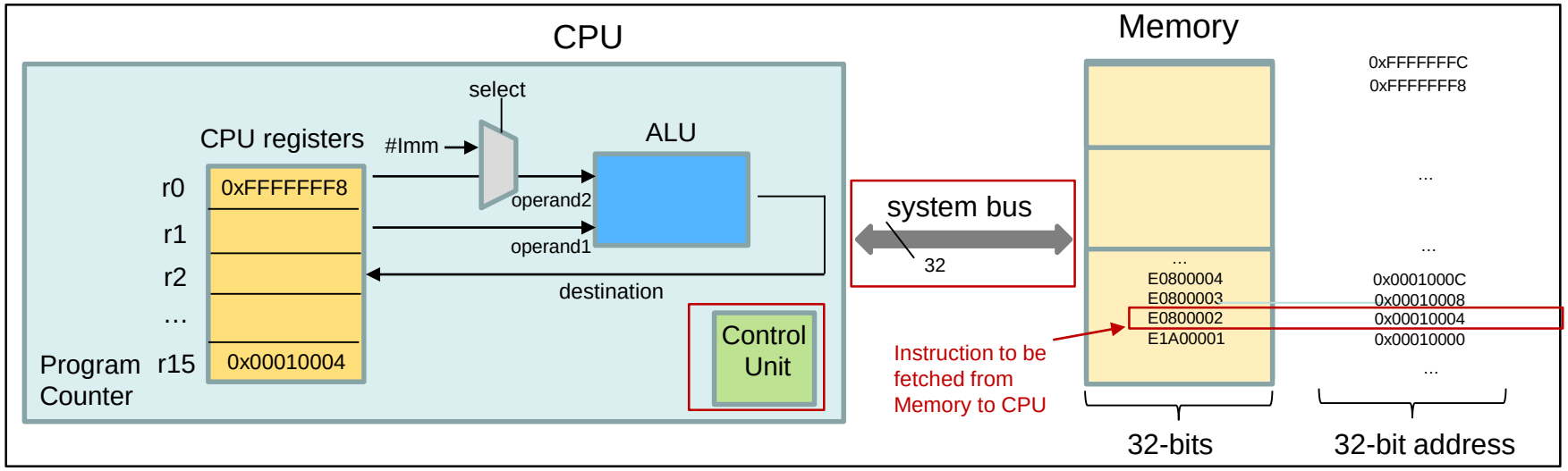


Simplified, conceptual block diagram of a computer

Modern Computer Architecture
Processing of instructions of a program

- A CPU implements a single machine instruction using three steps:
  1) Fetch: a machine instruction is fetched from Memory. CPU register r15, called the Program Counter (PC), contains the address of the instruction to be fetched. Thereafter, the PC is increased by four. The PC now contains the address of the next instruction to be fetched
  2) Decode: the machine instruction is broken down by the control unit to understand what operation needs to be done and the operands.
  3) Execute: the instruction is executed by the ALU

Thereafter, the next instruction is fetched, decoded, executed. In this way, many instructions of a program are processed.

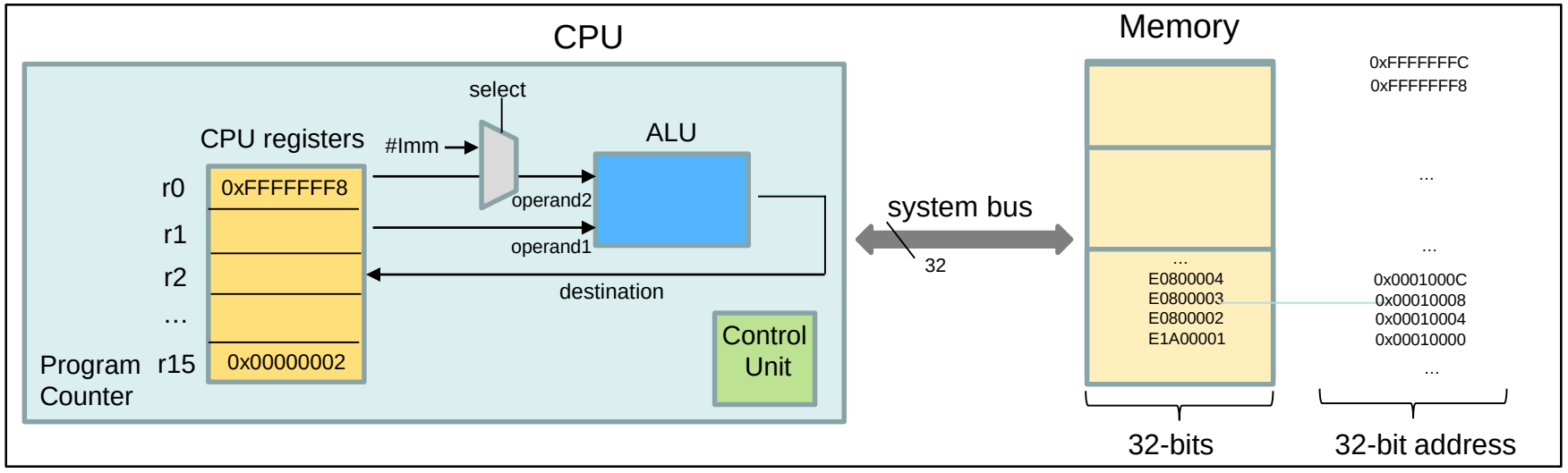


Simplified, conceptual block diagram of a computer

Modern Computer Architecture
Processing of instructions of a program

- A CPU implements a single machine instruction using three steps:
  1) Fetch: a machine instruction is fetched from Memory. CPU register r15, called the Program Counter (PC), contains the address of the instruction to be fetched. Thereafter, the PC is increased by four. The PC now contains the address of the next instruction to be fetched
  2) Decode: the machine instruction is broken down by the control unit to understand what operation needs to be done and the operands.
  3) Execute: the instruction is executed by the ALU

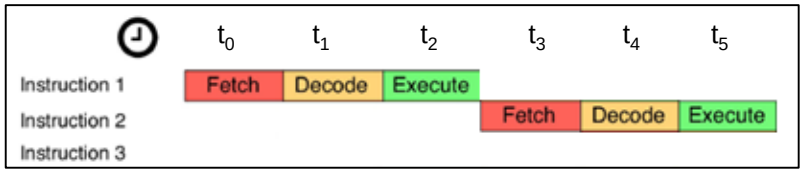
Thereafter, the next instruction is fetched, decoded, executed. In this way, many instructions of a program are processed.



Simplified, conceptual block diagram of a computer

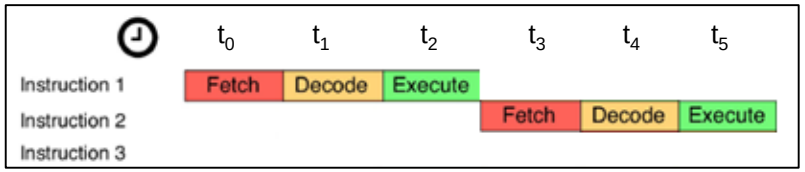
Modern Computer Architecture
Speeding up execution of a program

- A CPU uses three steps or time cycles to process a single instruction
  - Instruction 1 uses cycles t_0 – t_2 to process
  - Instruction 2 uses cycles t_3 – t_5 to process

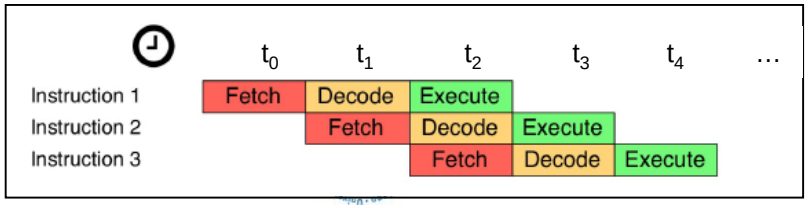


Modern Computer Architecture
Speeding up execution of a program

- A CPU uses three steps or time cycles to process a single instruction
  - Instruction 1 uses cycles t_0 – t_2 to process
  - Instruction 2 uses cycles t_3 – t_5 to process



- Since each step can be performed independently, modern CPUs use pipelining to speed up execution of a program
  - Cycle t_0: fetch instruction 1
  - Cycle t_1: decode instruction 1 and fetch instruction 2
  - Cycle t_2: execute instruction 1, decode instruction 2 and fetch instruction 3



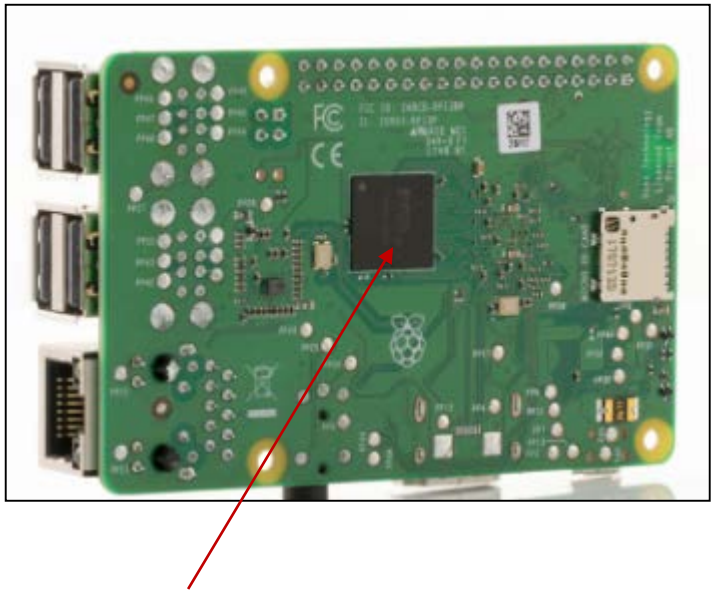
Computer architecture elements of the Raspberry Pi 3

Modern Computer Architecture
Raspberry Pi 3

- The Rpi 3 has a System on Chip (SoC): Broadcom BCM2837
  - 4 x ARM Cortex A53, each running at 1.2 GHz

Modern Computer Architecture
Raspberry Pi 3

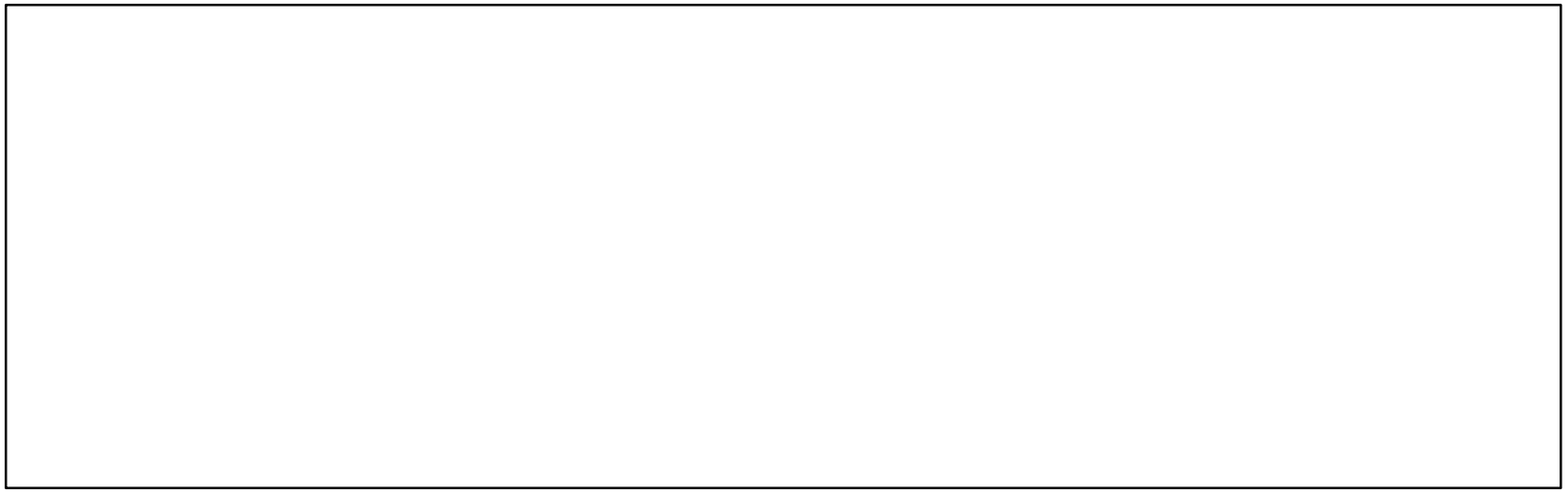
- The Rpi 3 has a System on Chip (SoC): Broadcom BCM2837
  - 4 x ARM Cortex A53, each running at 1.2 GHz



1GB RAM Memory

Modern Computer Architecture
Raspberry Pi 3

- The Rpi 3 has a System on Chip (SoC): Broadcom BCM2837
  - 4 x ARM Cortex A53, each running at 1.2 GHz
- What computer architecture is used in each ARM Cortex A53?

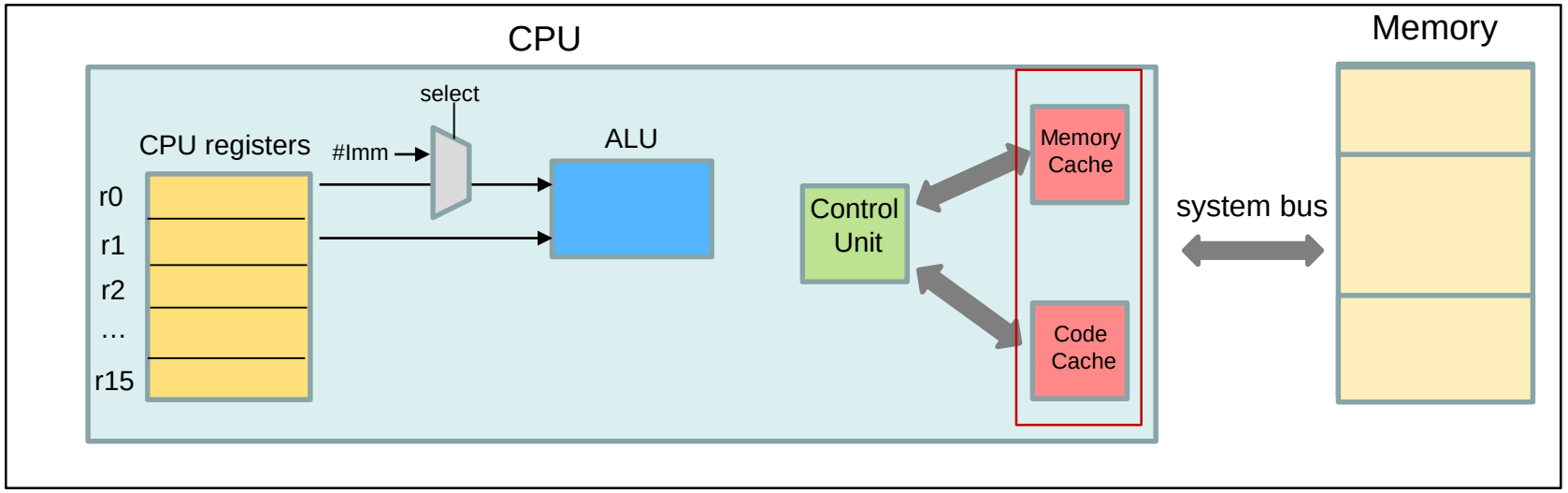


Simplified, conceptual block diagram of the ARM Cortex A53

Modern Computer Architecture
Raspberry Pi 3

- The Rpi 3 has a System on Chip (SoC): Broadcom BCM2837
  - 4 x ARM Cortex A53, each running at 1.2 GHz

- What computer architecture is used in each ARM Cortex A53?
  - Memory and Code cache are used to access information quicker than Memory. The CPU first searches for instructions & data in the cache, before searching in Memory. It is quicker to read information from cache than Memory!

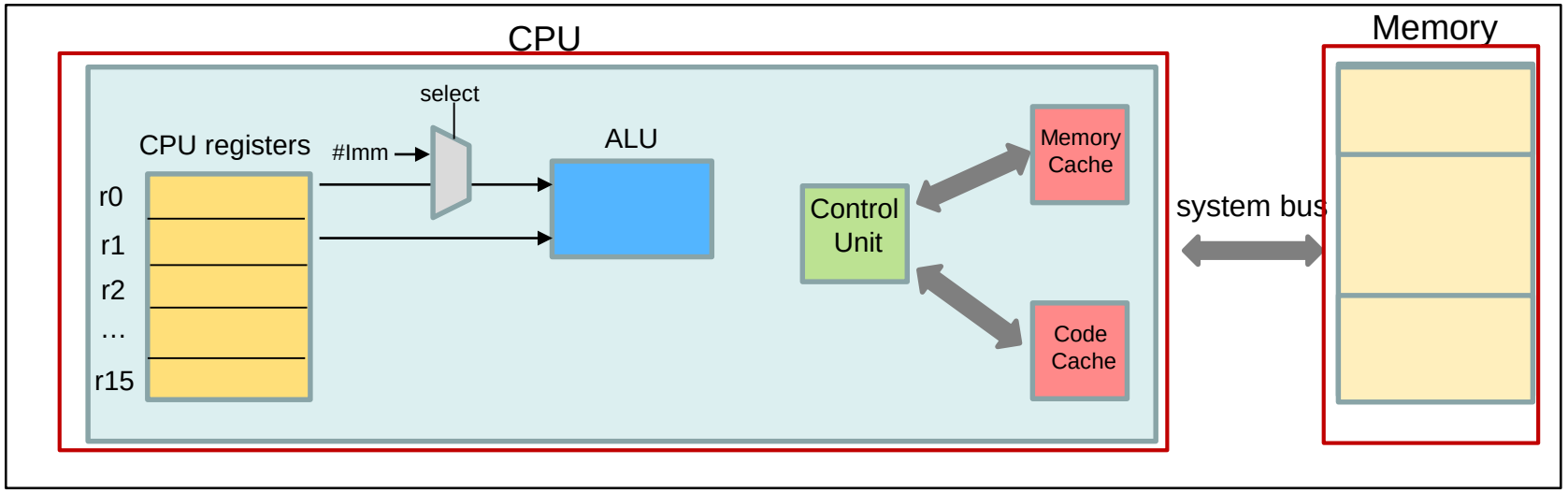


Simplified, conceptual block diagram of the ARM Cortex A53

Modern Computer Architecture
Raspberry Pi 3

- The Rpi 3 has a System on Chip (SoC): Broadcom BCM2837
  - 4 x ARM Cortex A53, each running at 1.2 GHz

- What computer architecture is used in each ARM Cortex A53?
  - Memory and Code cache are used to access information quicker than Memory
  - On the large scale, the architecture is 'von Neumann'

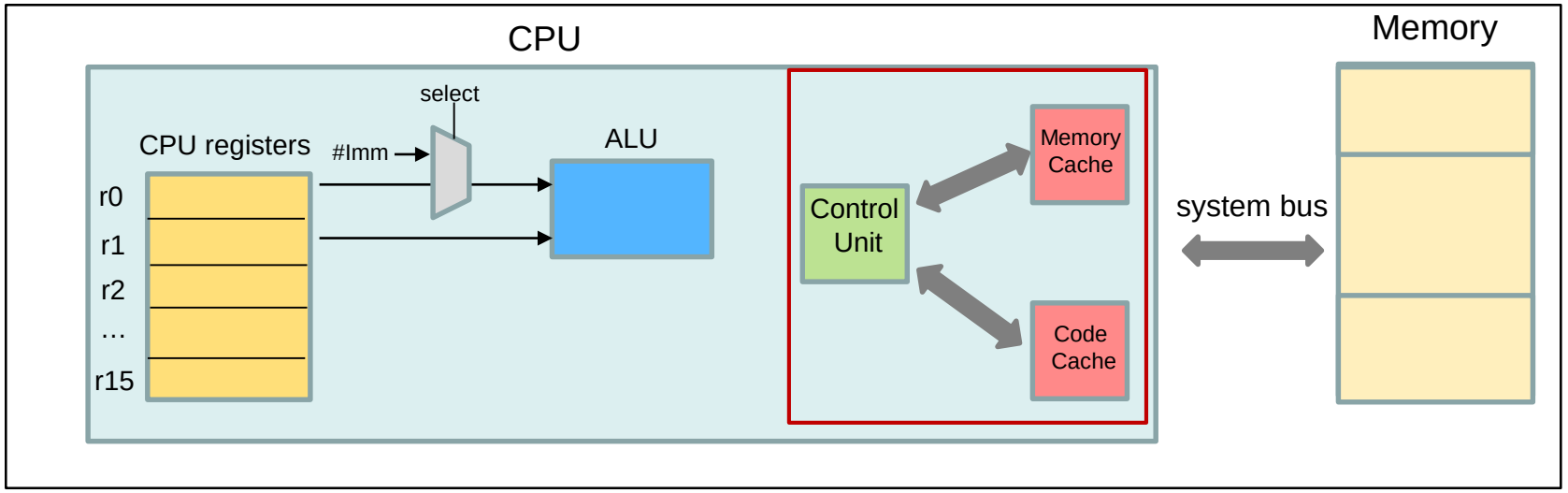


Simplified, conceptual block diagram of the ARM Cortex A53

Modern Computer Architecture
Raspberry Pi 3

- The Rpi 3 has a System on Chip (SoC): Broadcom BCM2837
  - 4 x ARM Cortex A53, each running at 1.2 GHz

- What computer architecture is used in each ARM Cortex A53?
  - Memory and Code cache are used to access information quicker than Memory
  - On the large scale, the architecture is ‘von Neumann’
  - On a small scale, the architecture is ‘Harvard’

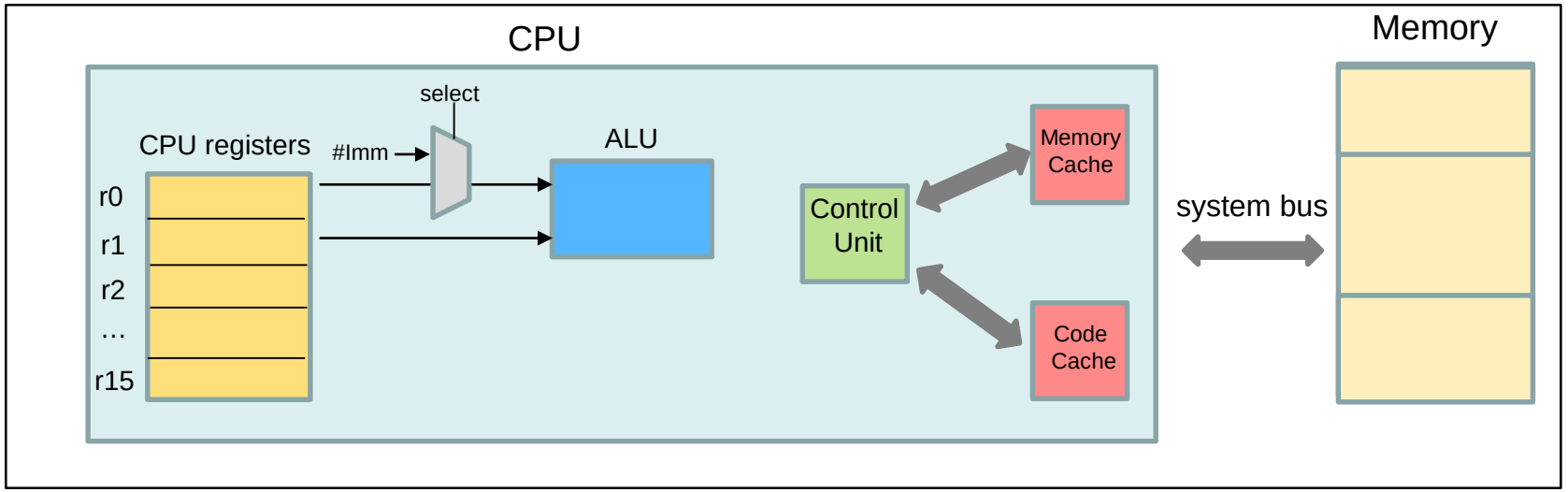


Simplified, conceptual block diagram of the ARM Cortex A53

Modern Computer Architecture
Raspberry Pi 3

- The Rpi 3 has a System on Chip (SoC): Broadcom BCM2837
  - 4 x ARM Cortex A53, each running at 1.2 GHz

- What computer architecture is used in each ARM Cortex A53?
  - Memory and Code cache are used to access information quicker than Memory
  - On the large scale, the architecture is ‘von Neumann’
  - On a small scale, the architecture is ‘Harvard’
  - Architecture referred as ‘modified Harvard’



Simplified, conceptual block diagram of the ARM Cortex A53