# Digital Electronics and Microprocessor Systems (ELEC211)

Dave McIntosh and **Valerio Selis**

dmc@liverpool.ac.uk
**V.Selis@liverpool.ac.uk**

UNIVERSITY OF LIVERPOOL

# Outline

- Control Processing Unit
- The ARM Cortex M0 core
  - Instruction register
  - Instruction decoder
  - Control Unit
  - Arithmetic and Logical Unit (ALU)
  - Bank register
  - Address register
- Fetch, decode and execute
- Processor core connections
  - Internal connections
  - External connections
- Instruction set and program counter

# The Central Processing Unit

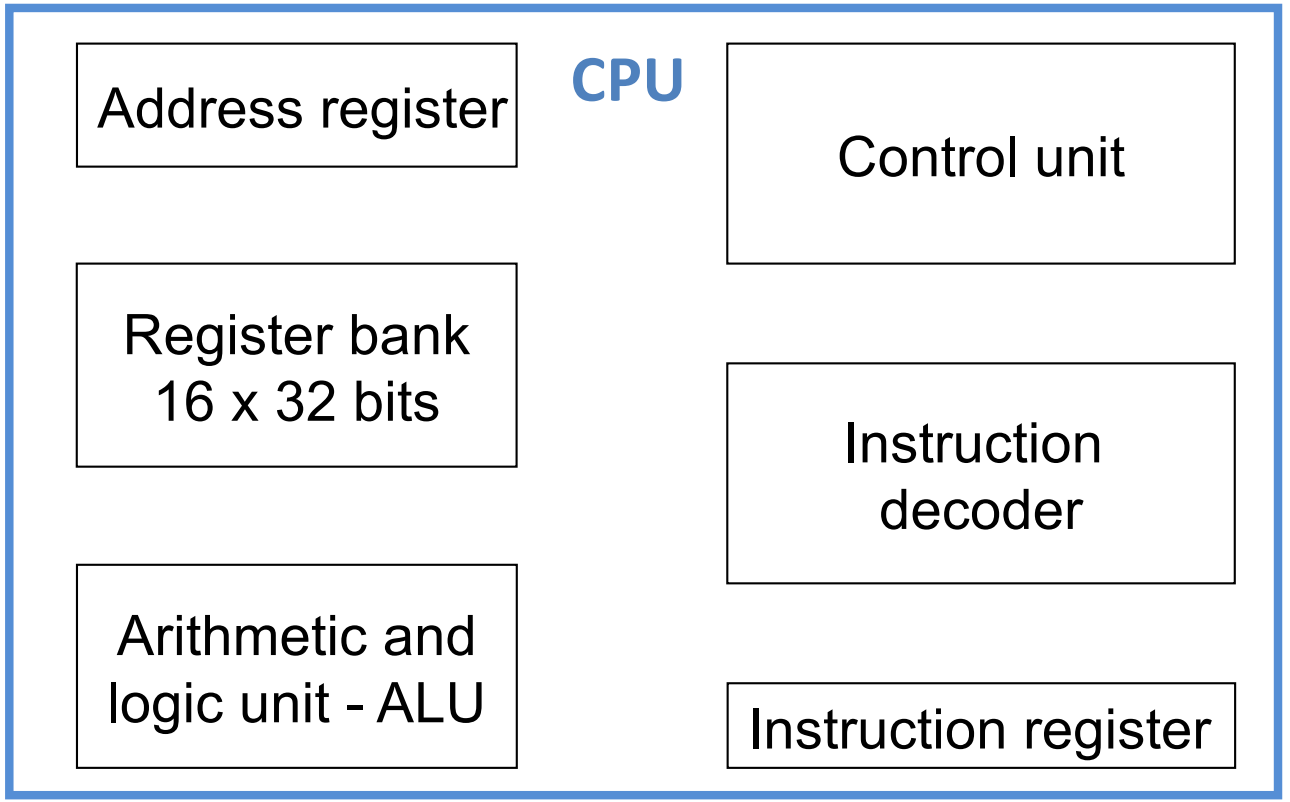
- The central processing unit or CPU is the part of a microprocessor system that does all the ‘work’.
- It interprets the instructions stored in memory.
- It performs the calculations.
- It controls the flow of data along the data bus.
- It determines which memory address to use.

# Inside the CPU

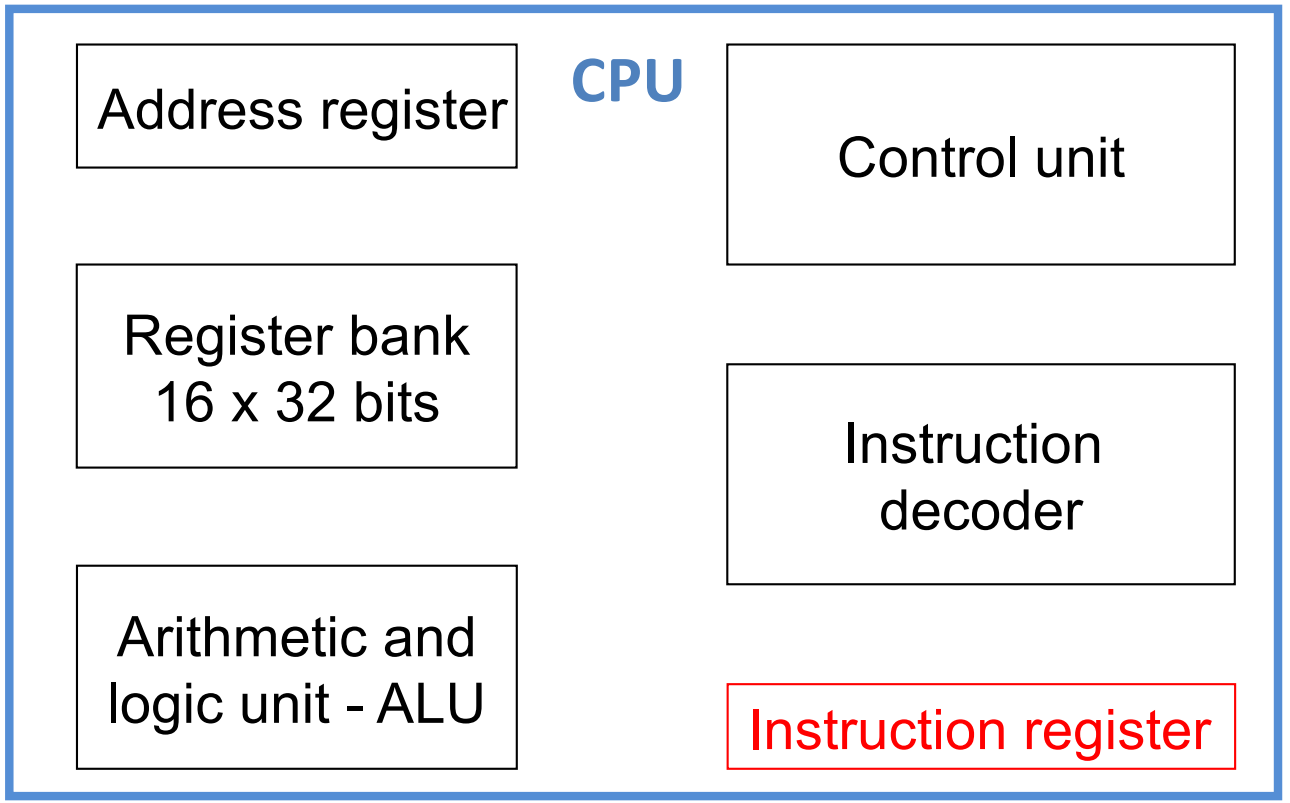
- The CPU is designed to perform all of these functions as efficiently as possible.
- It can be subdivided into a number of blocks; each with a distinct function.
- Every CPU is different and we will concentrate on the ARM Cortex M0 ‘core’ - the CPU for the Cortex M0 range of microprocessors.

# The ARM Cortex M0 core

- The basic building blocks of the ARM Cortex M0 core are:



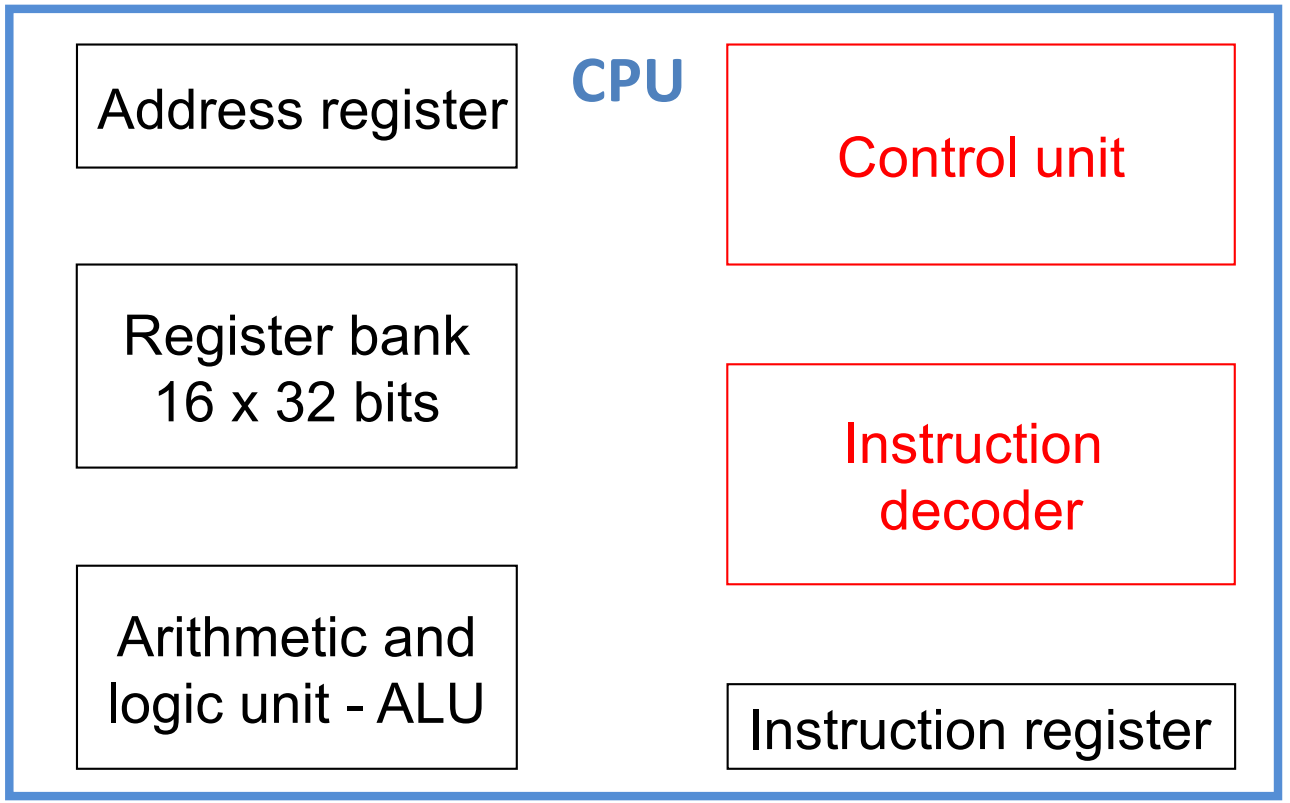
# The instruction register



# The instruction register

- The instructions stored in memory travel along the data bus to the CPU where they are loaded into the **Instruction Register (IR)**.
- Instructions are mostly 16 bits long but some are 32 bits long. They are stored in the instruction register - not part of the main memory.
- The process of loading the instruction register from memory is known as a ‘fetch’ - **FETCH Operation (IF)**.

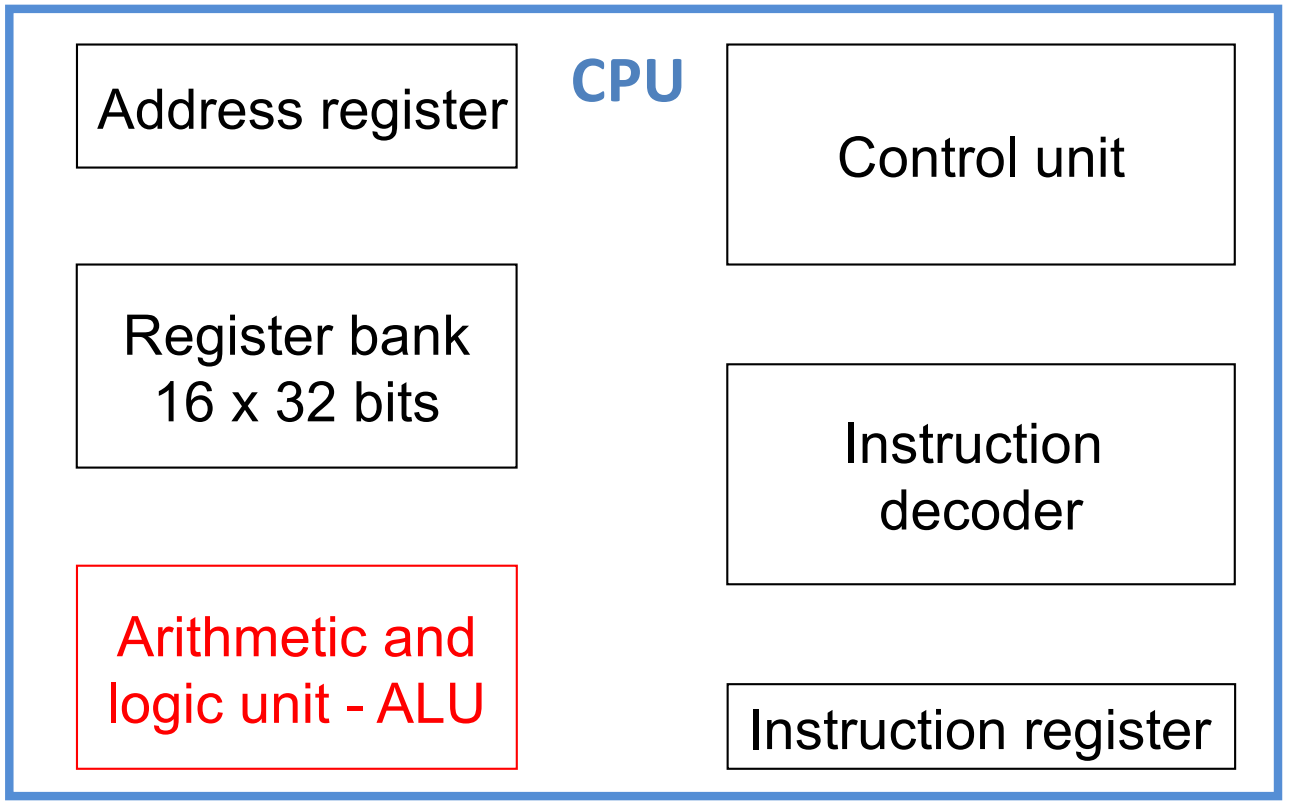
# Instruction decoder and control unit



# Instruction decoder and control unit

- The instructions are in ‘machine code’ and the instruction decoder determines the function of each instruction.
- The instruction decoder and control unit determine what the other parts of the CPU do.
- The control unit is also in charge of the control bus.
- The process of interpreting each instruction is known as the **‘decode’ cycle (ID)**.

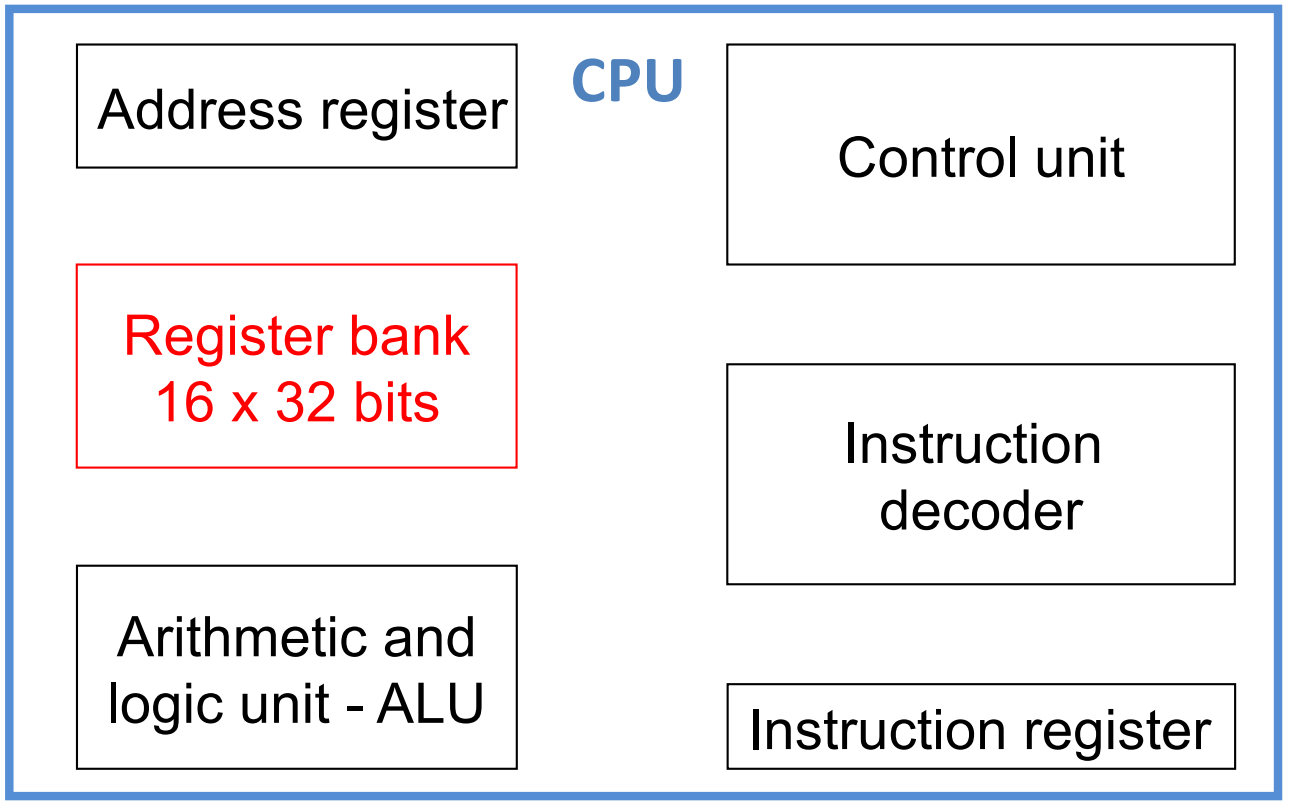
# Arithmetic and logic unit



# Arithmetic and logic unit

- The arithmetic and logic unit or ALU performs the mathematical functions as required.
  - Logical
    - AND, OR, XOR etc.
  - Arithmetic
    - Addition, subtraction, multiplication
- The process of performing each instruction is known as the **'execution' cycle (IE)**.

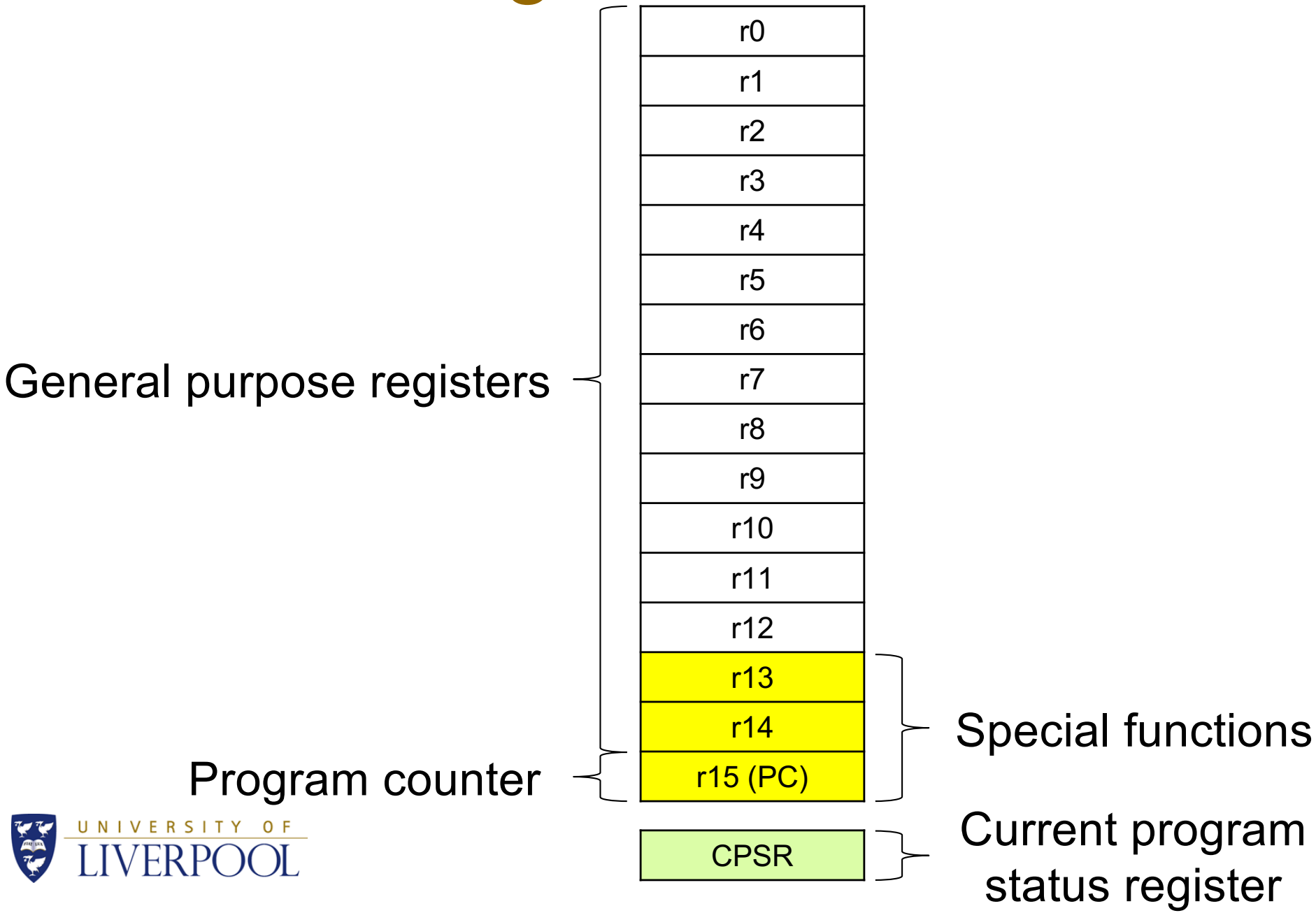
# Register bank



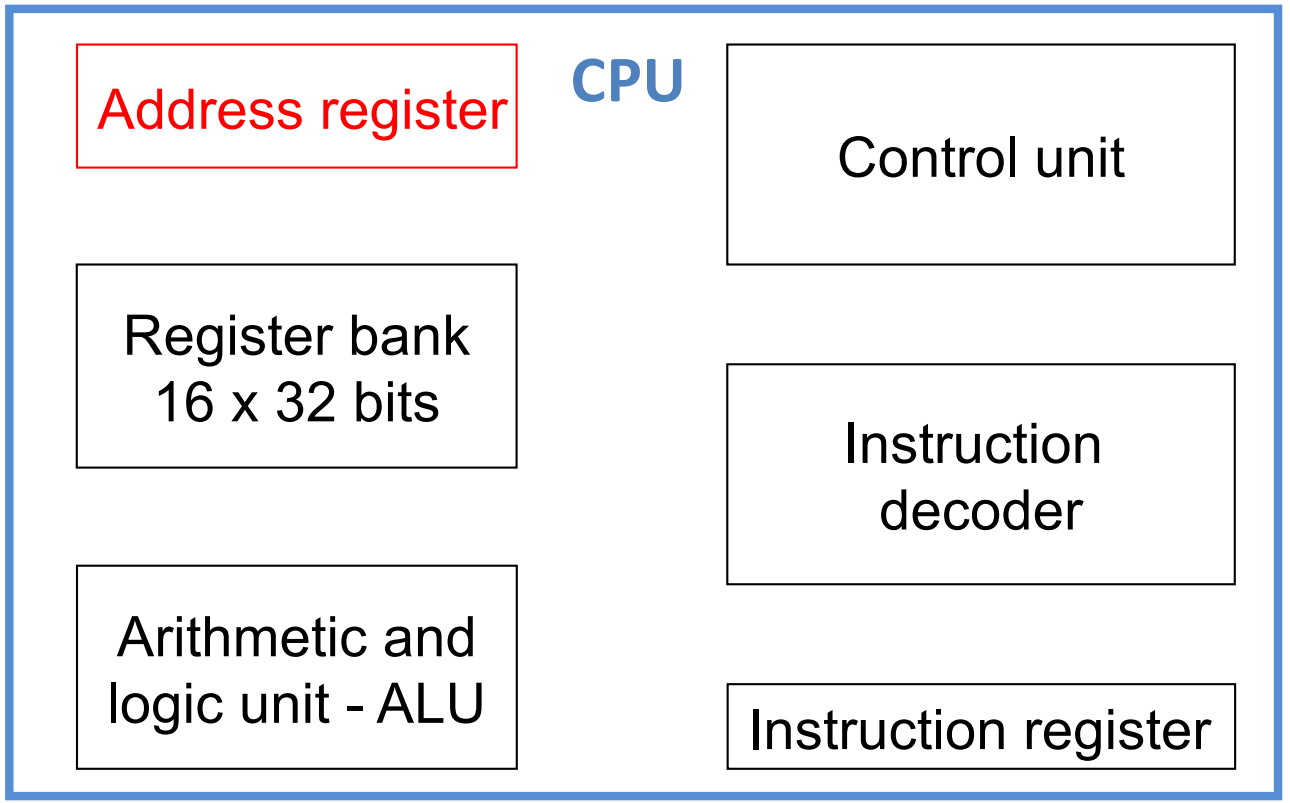
# Register bank

- The register bank is a local memory for the CPU.
- It has 16 locations - each location can hold 32 bits of data.
- The registers are named r0, r1, r2, r3, ... etc. up to r15.
- They are used to hold data which is processed by the ALU and also hold the results of any calculation.
- Registers r13, r14 and r15 have special functions which we will cover later.

# Register bank



# Address register



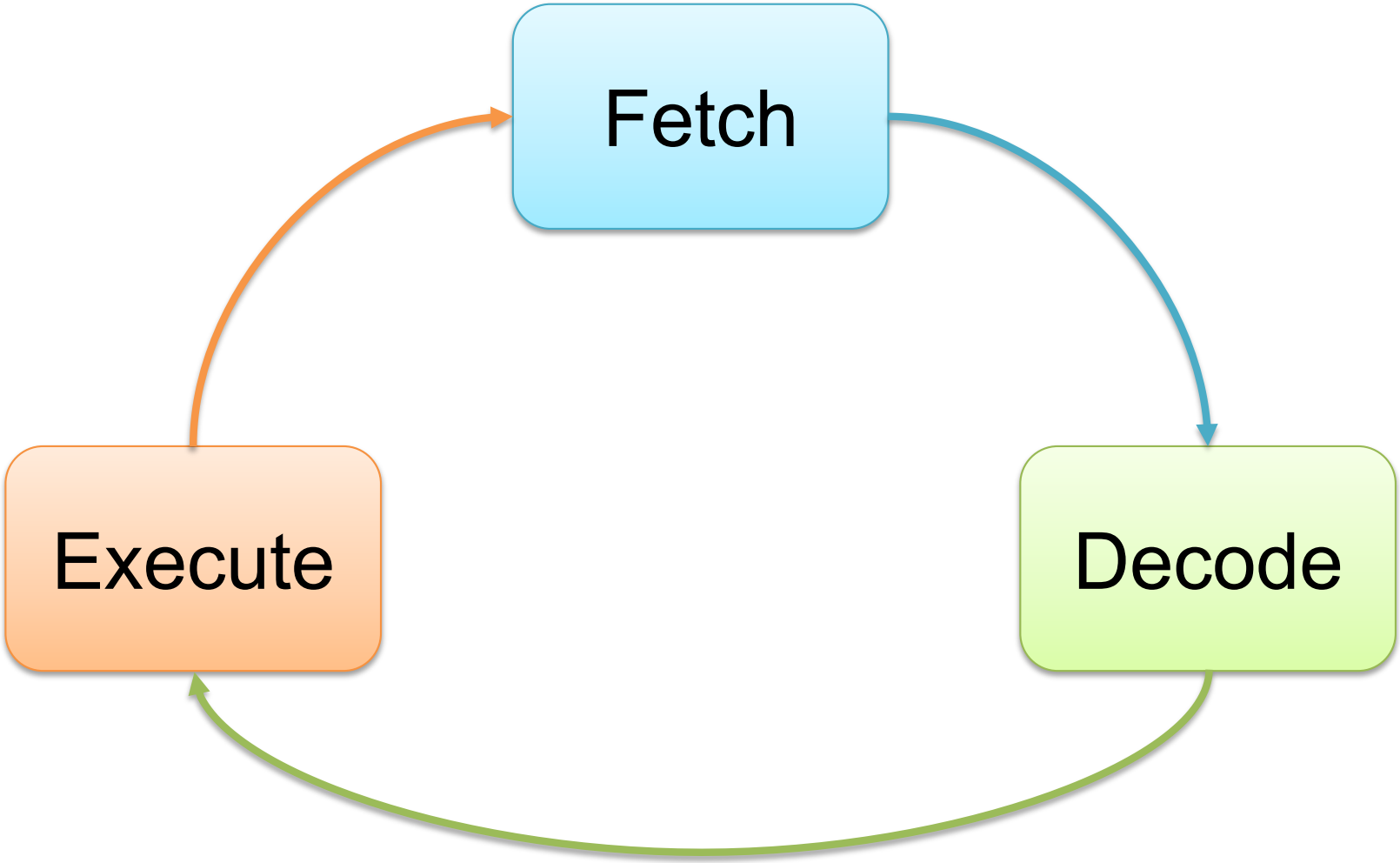
# Address register

- The address register is a 32 bit memory device which holds a memory address value.
- Either this address may be for the memory location of the next instruction during the **'fetch' cycle (IF)**.
- Or during the **'execute' cycle** the address is for a memory location either containing data to be loaded into a register or where data from a register is to be stored.

# Fetch, decode, execute

- The CPU performs three cycles sequentially.
- During the FETCH cycle an instruction in memory is loaded into the instruction register.
- During the DECODE cycle the instruction is interpreted by the instruction decoder.
- During the EXECUTE cycle either the ALU performs a calculation on values held in registers
- or a value in a register is stored into memory
- or a value in memory is loaded into a register.

# Fetch, decode, execute



# Question

When poll is active, respond at **PollEv.com/elec211**

Text **ELEC211** to **22333** once to join

**How many bits do we need to address the general purpose registers in the register bank?**

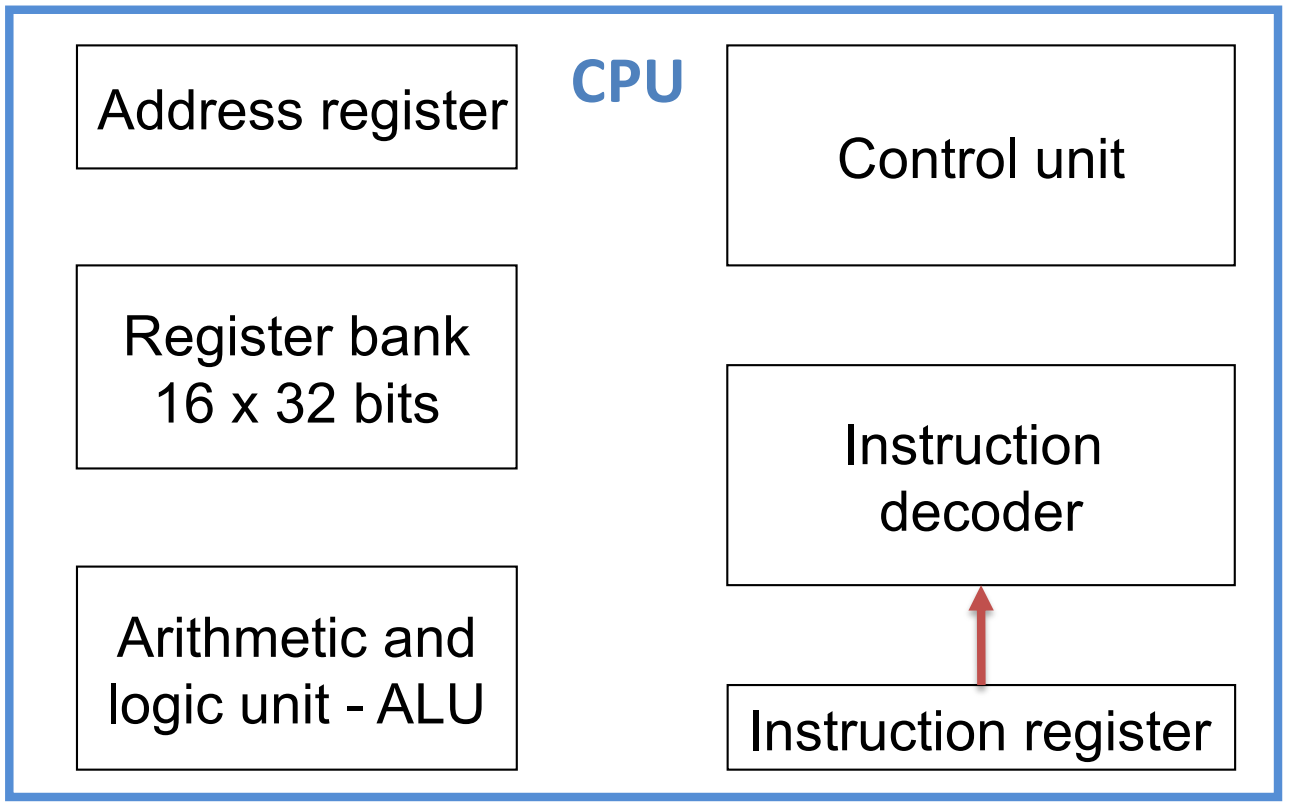
- 512
- 480
- 1024
- 320

Total Results: 0

Start the presentation to see live content. Still no live content? Install the app or get help at **PollEv.com/app**

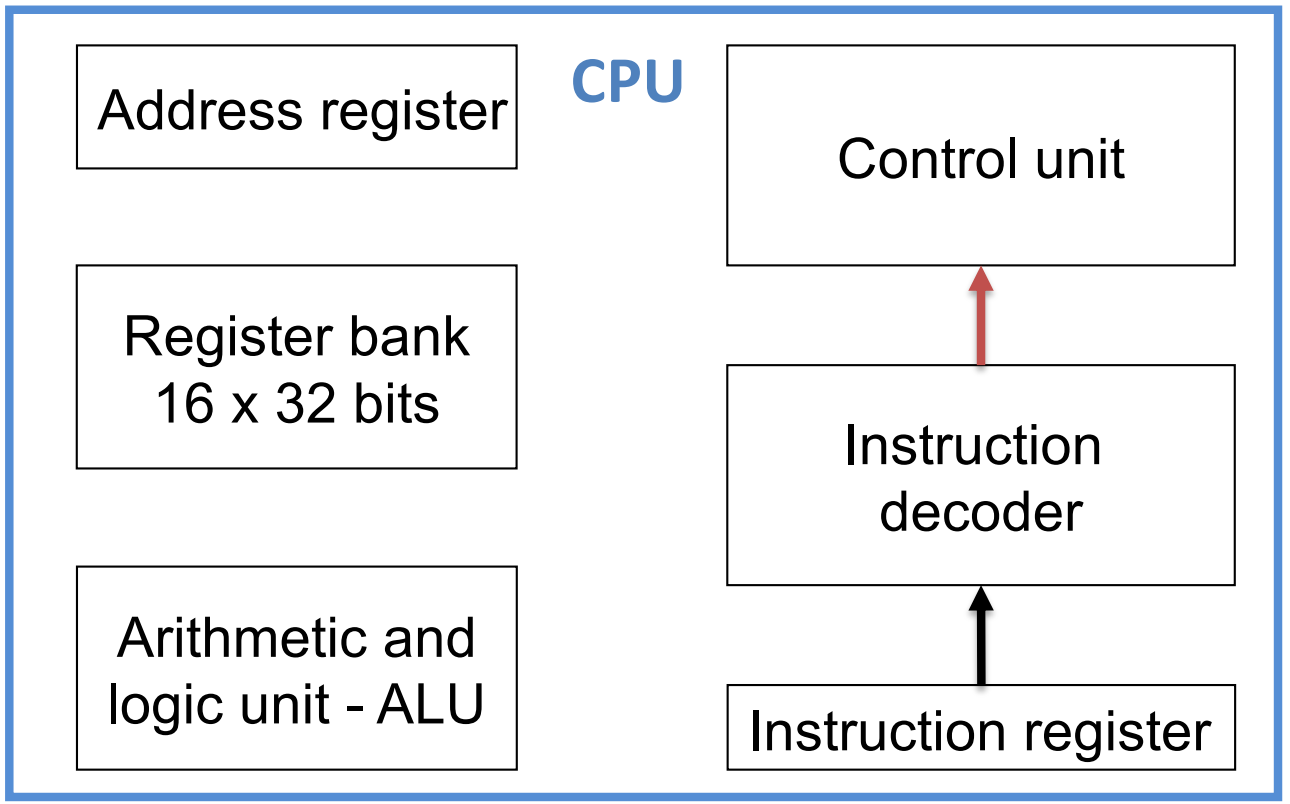
UNIVERSITY OF LIVERPOOL

# Internal connections



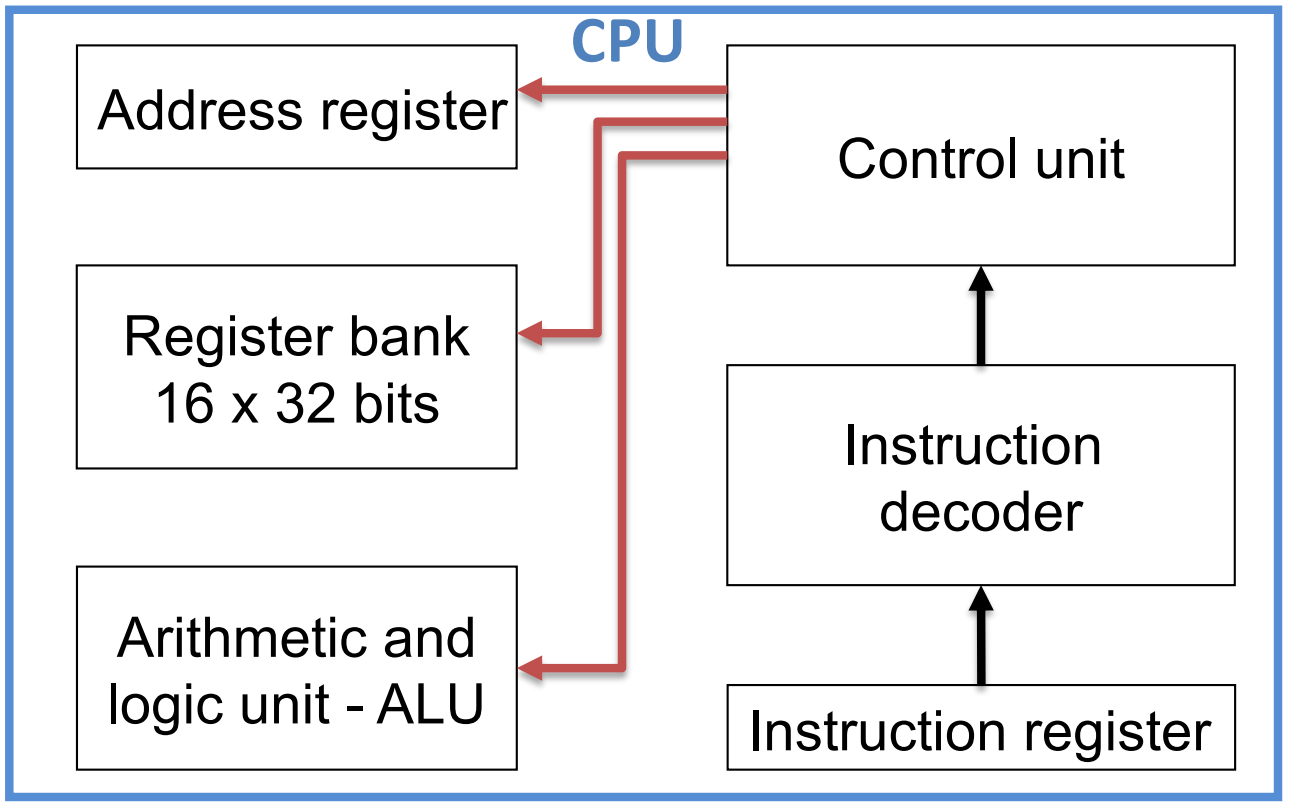

- The instruction register is connected to the instruction decoder.

# Internal connections



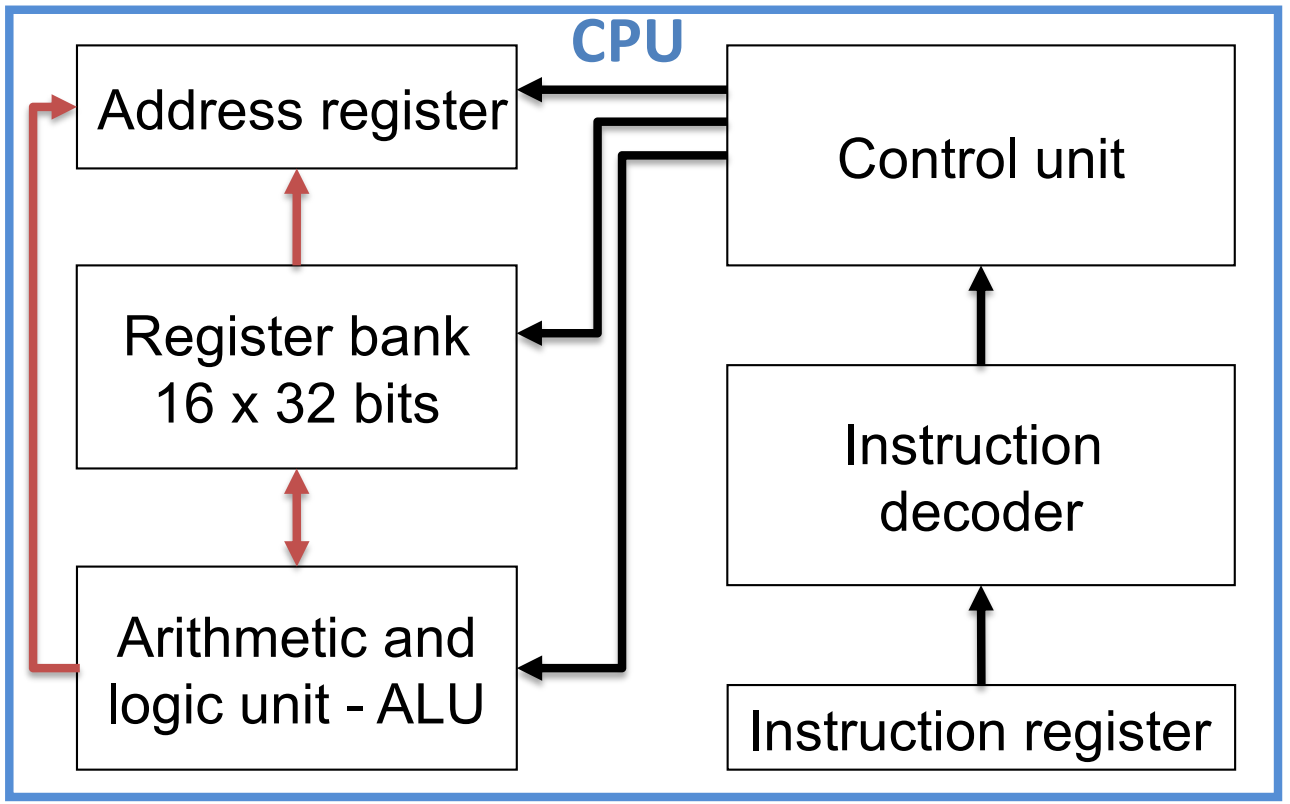

- The instruction decoder is connected to the control unit.

# Internal connections



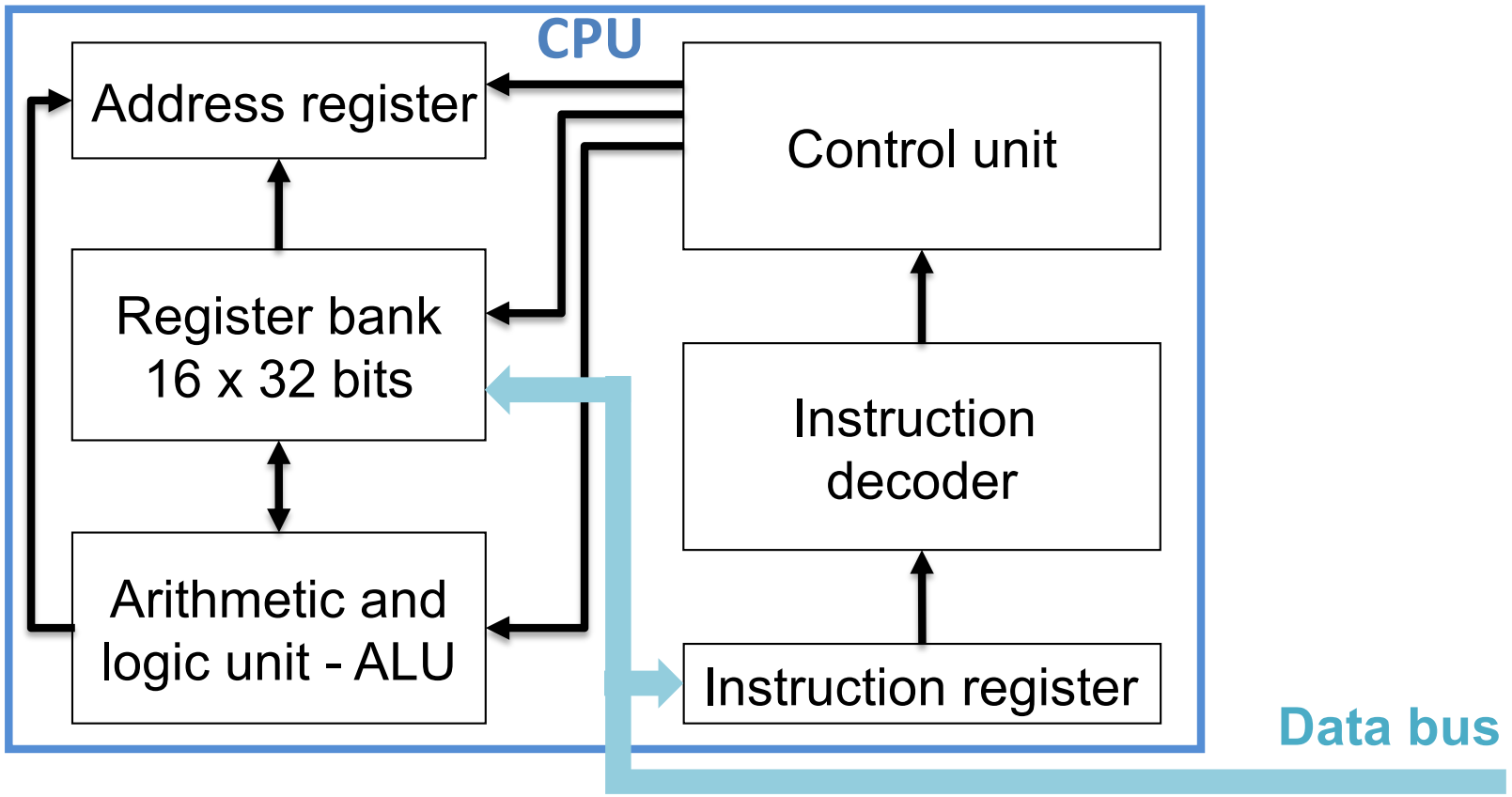

- The control unit is connected to the ALU, the register bank and the address register.

# Internal connections



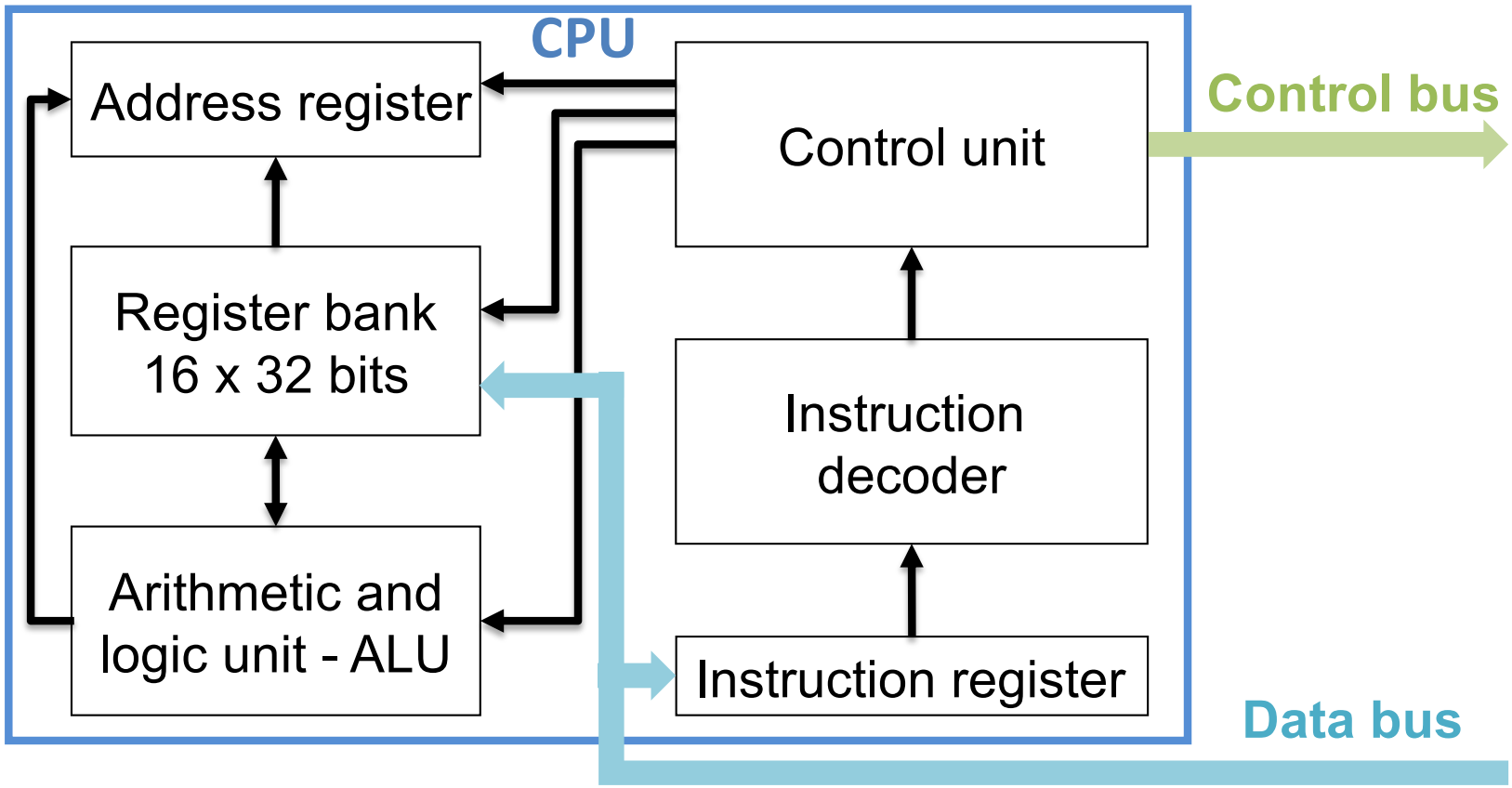

- The ALU and the register bank are connected to each other and the address register.

# External connections



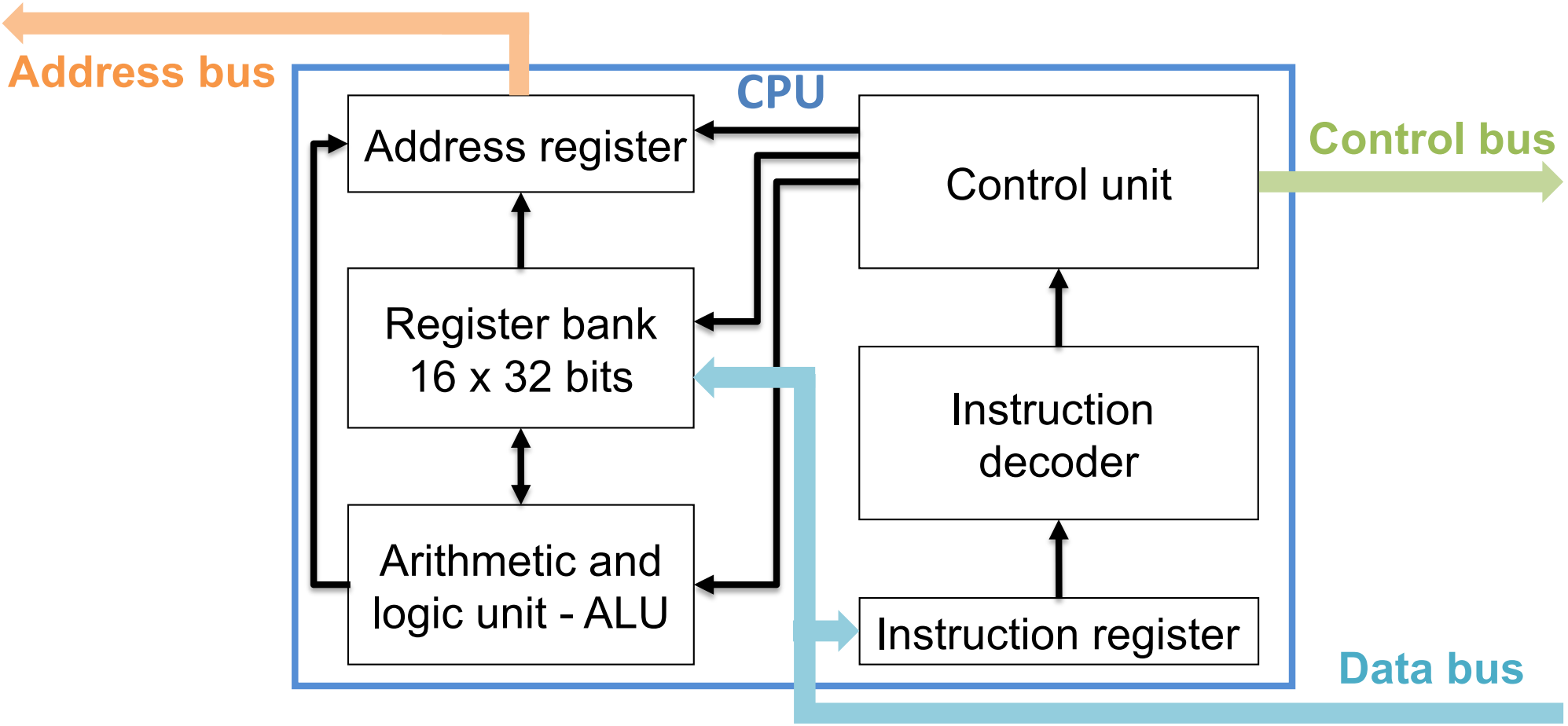

- The data bus is connected to both the instruction register and the register bank.

# External connections




- The control bus is connected to the control unit.

# External connections




- Address bus is connected to the address register.

# Program counter

- Local register to the CPU
- Instructions are stored in memory so:
  - What is the memory address for the next instruction?
  - Register r15 always holds the memory address of the next instruction to be executed.
  - An alternative name for register r15 is the ‘program counter’.

# Instruction stored in memory

- Instructions are either 16 bits (half-words) or 32 bits (words) long
- Memory locations are 8 bits long so one instruction occupies either two or four locations in memory
  - E.g. one 16 bit instruction would be stored in 2 memory locations with addresses 0x00008000 and 0x00008001.
  - The following 16 bit instruction would be stored at addresses 0x00008002 and 0x00008003.

# Question

When poll is active, respond at **PollEv.com/elec211**

Text **ELEC211** to **22333** once to join

**A 16 bit instruction is stored at addresses 0x00008002 and 0x00008003.**

**Where would the following 16 bit instruction be stored at?**

- 0x00008005 and 0x00008006
- 0x00008003 and 0x00008004
- 0x00008004 and 0x00008005
- 0x00008006 and 0x00008007

Total Results: 0

Start the presentation to see live content. Still no live content? Install the app or get help at **PollEv.com/app**

UNIVERSITY OF LIVERPOOL

# Question

When poll is active, respond at **PollEv.com/elec211**

Text **ELEC211** to **22333** once to join

**A 16 bit instruction is stored at addresses 0x00008004 and 0x00008005.**

**Where would the following 32 bit instruction be stored at?**

- 0x00008005, 0x00008006
- 0x00008006, 0x00008007
- 0x00008007, 0x00008008, 0x00008009, 0x00008010
- 0x00008006, 0x00008007, 0x00008008, 0x00008009

Total Results: 0

Answers to this poll are anonymous

Start the presentation to see live content. Still no live content? Install the app or get help at **PollEv.com/app**

UNIVERSITY OF LIVERPOOL

# Instruction stored in memory

| 16 bits instruction | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| | | 7 | 6 | 5 | 4 | 3 | 2 | 0x00008000 |
| 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 0x00008001 |
| 1 | 0 | | | | | | | 0x00008002 |
| | | | | | | | | |
| | | | | | | | | |

| 16 bits instruction | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00008000 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00008001 |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

UNIVERSITY OF LIVERPOOL

# Instruction stored in memory

| 16 bits instruction | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| | | 7 | 6 | 5 | 4 | 3 | 2 |
| 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 |
| 1 | 0 | | | | | | |
| | | | | | | | |
| | | | | | | | |

0x00008000
0x00008001
0x00008002

| 32 bits instruction | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 7 | 6 | 5 | 4 | 3 | 2 |
| 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 |
| 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 |
| 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 |
| 1 | 0 | | | | | | |
| | | | | | | | |

0x00008000
0x00008001
0x00008002
0x00008003
0x00008004

| 16 bits instruction | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

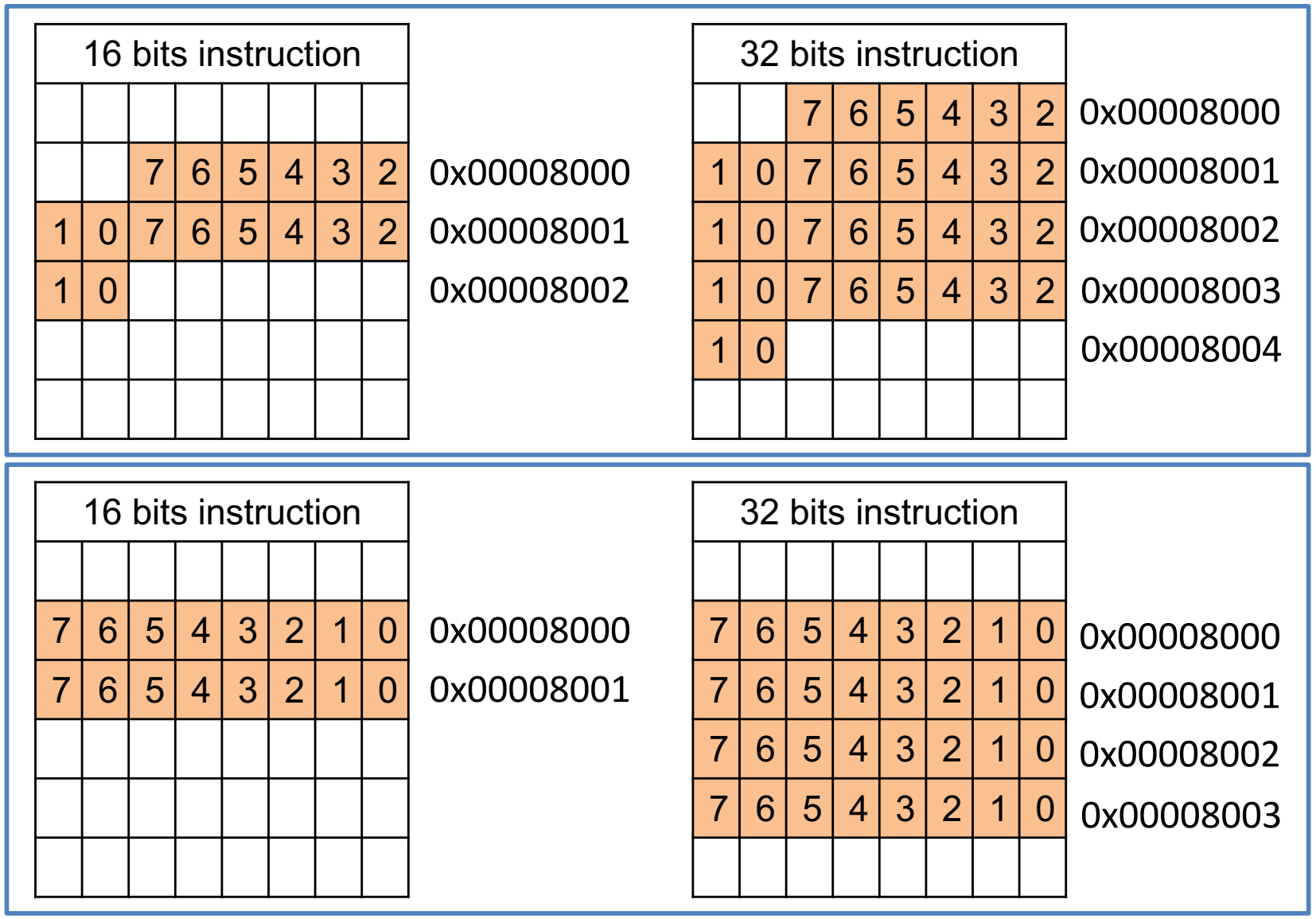
0x00008000
0x00008001

| 32 bits instruction | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | |

0x00008000
0x00008001
0x00008002
0x00008003

UNIVERSITY OF LIVERPOOL

# Instruction stored in memory




UNIVERSITY OF LIVERPOOL

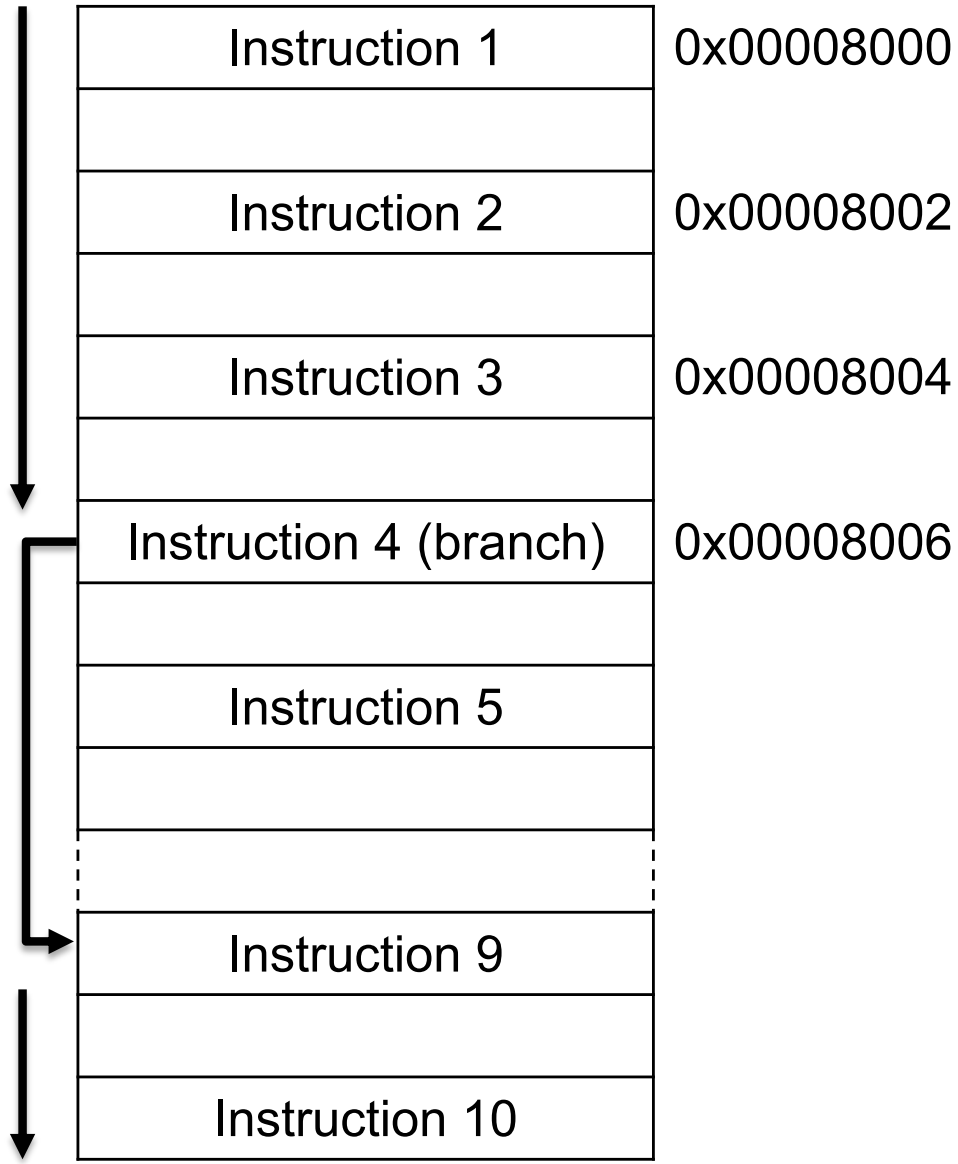
# Instruction stored in memory

- In general instructions are in consecutive locations in memory so that they are executed in the same order as they appear in memory.
- Assuming instructions are 16 bits long:
- If the instruction being executed is stored at address 0x00008000, the next instruction to be executed will be stored at address 0x00008002, the next at 0x00008004, the next at 0x00008006 and so on – except when…...

# Program counter

- When an instruction is executing, the program counter, r15, increments by 2 (or 4) so that it holds the memory address of the next instruction.
- EXCEPT when a ‘branch’ instruction is executed when the computer program ‘branches’ to another part of memory and the program counter holds a completely new memory address.

# Instruction stored in memory




UNIVERSITY OF LIVERPOOL

# Summary

- Control Processing Unit
- The ARM Cortex M0
- Fetch, decode and execute
- Processor core connections
- Instruction set and program counter

# Next class?

Monday at 4 p.m. in the
Building 502,
Lecture Theatre 2
(502-LT2)