

COMS12200

Introduction to

Computer Architecture

Dr. Cian O'Donnell

cian.odonnell@bristol.ac.uk

Topic 3: The Execution Cycle

Topics

1. Data, Control and Instructions
2. Memory
3. Execution cycle
4. Processor control flow
5. State machines and decoding
6. Machine types
7. Memory paradigms

Memory intro

Recap

1. Memory as a big long list of addresses for storing info.
2. The memory hierarchy.
3. Strengths and weaknesses of various memory addressing modes.

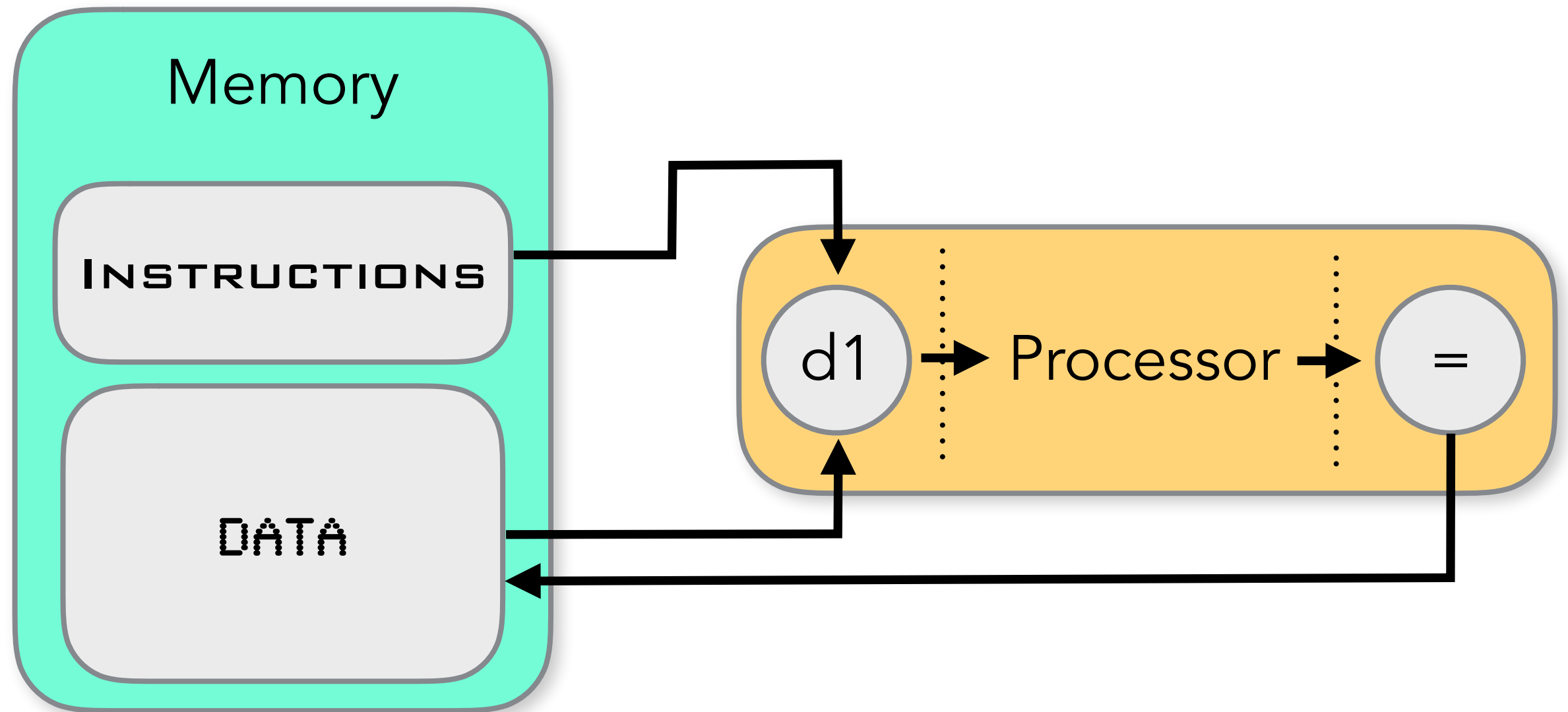
Topics

1. Data, Control and Instructions
2. Memory
3. Execution cycle
4. Processor control flow
5. State machines and decoding
6. Machine types
7. Memory paradigms

Topics

1. Data, Control and Instructions
2. Memory
3. Execution cycle
4. Processor control flow
5. State machines and decoding
6. Machine types
7. Memory paradigms

Recap of information flow

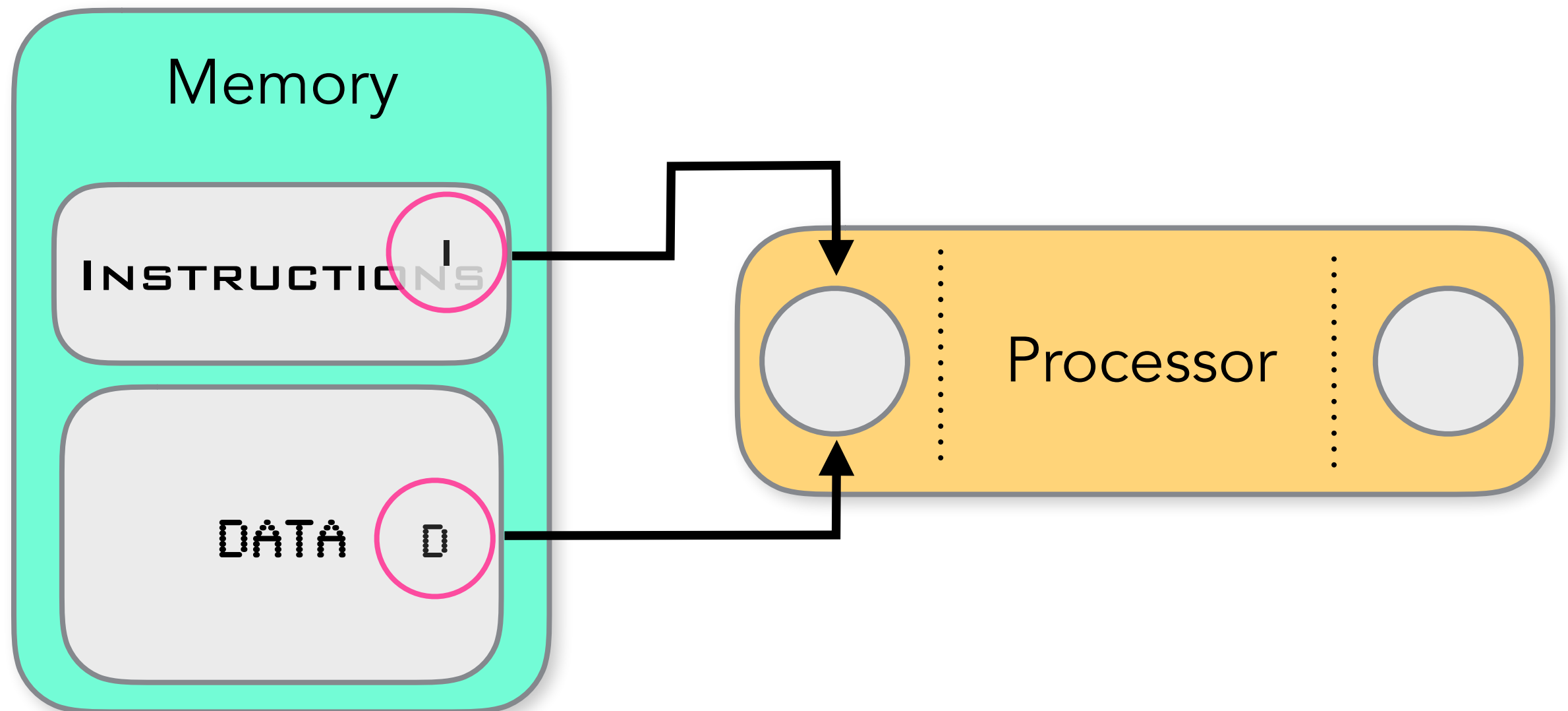


Fetch-execute

- In all modern processors, the necessary instructions and data to do work are gathered in what is called the "fetch-execute cycle".
- There are two basic stages to this cycle:
 1. Fetching of instructions and data.
 2. Execution of the instruction and creation of a result.

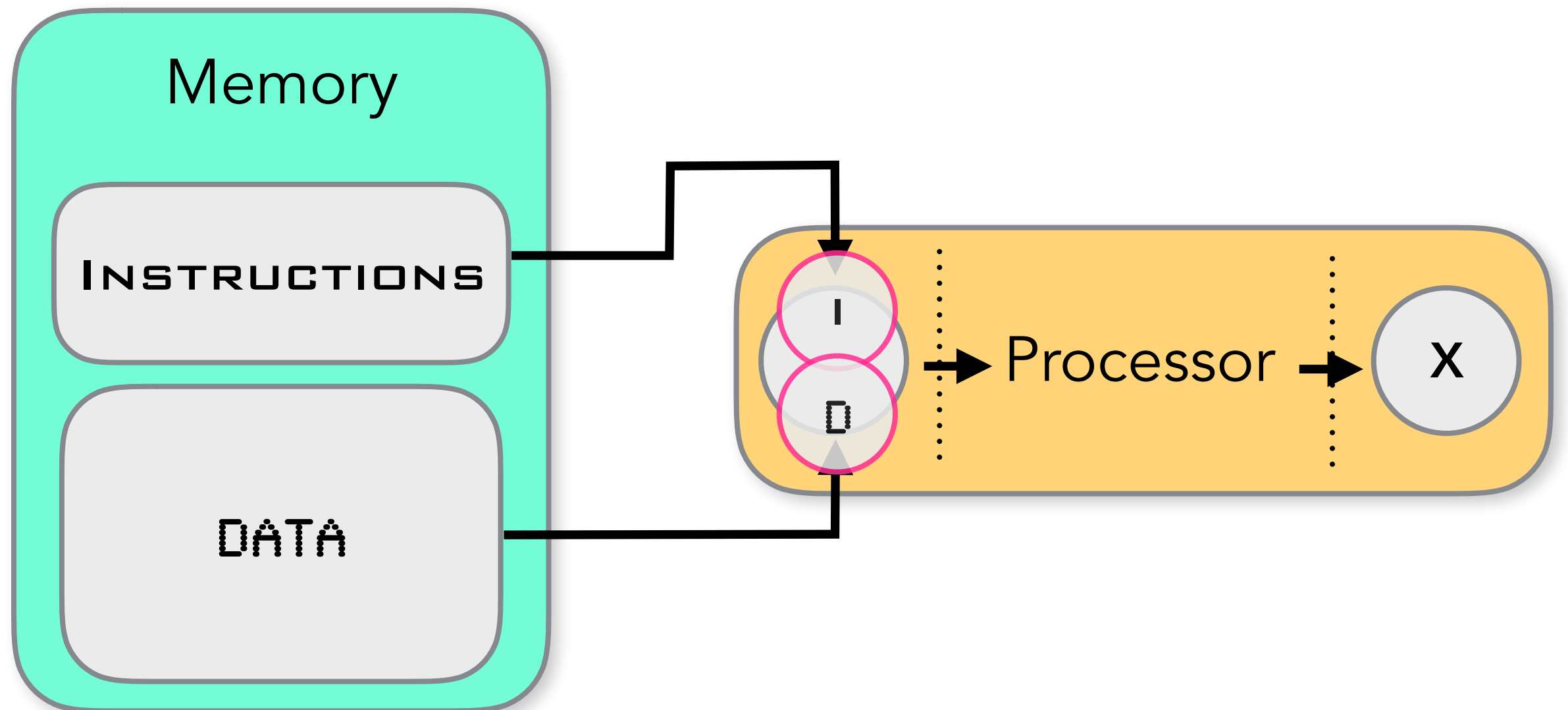
The “fetch”

- An instruction and data are loaded into the processor.

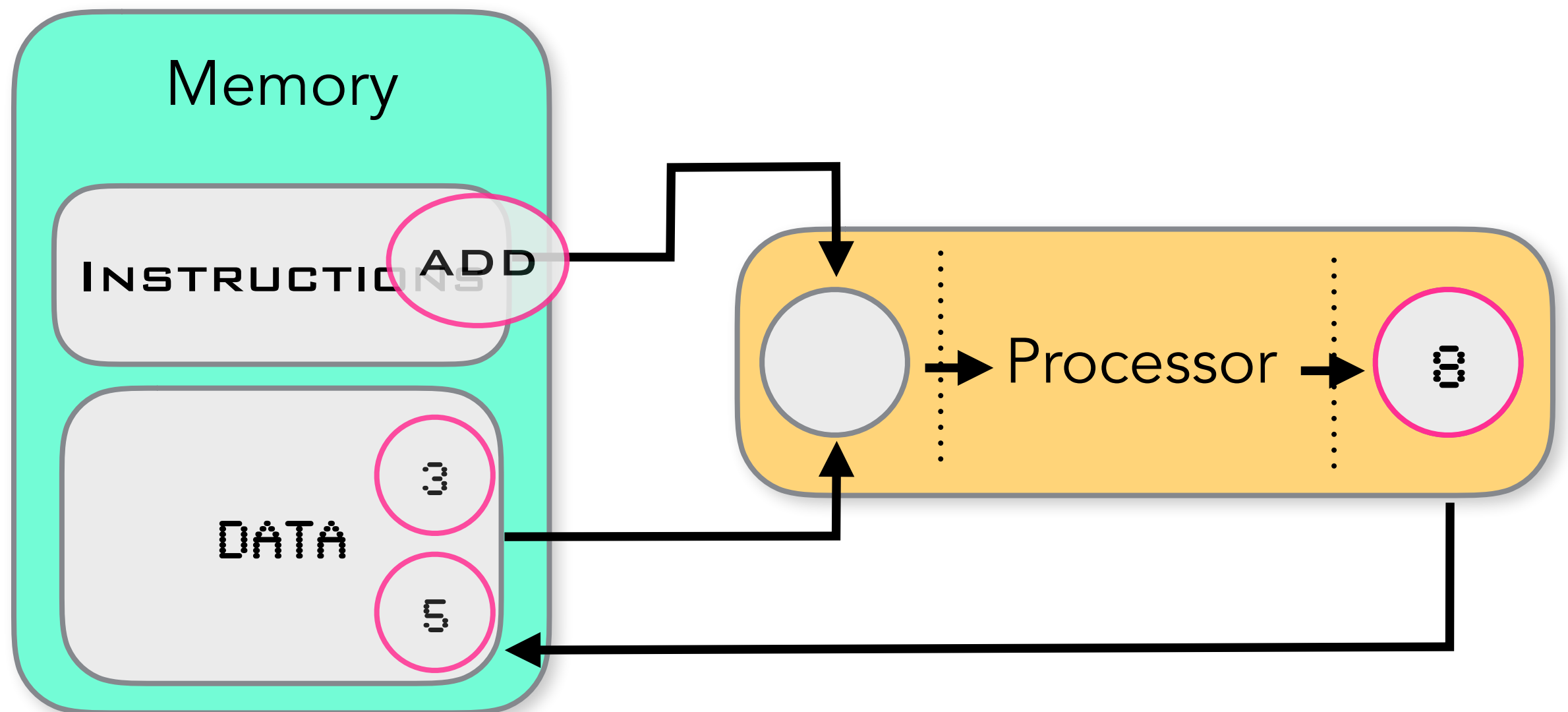


The "execute"

- The instruction is interpreted and operated on the data, producing a result



Example: fetch/execute



About that result

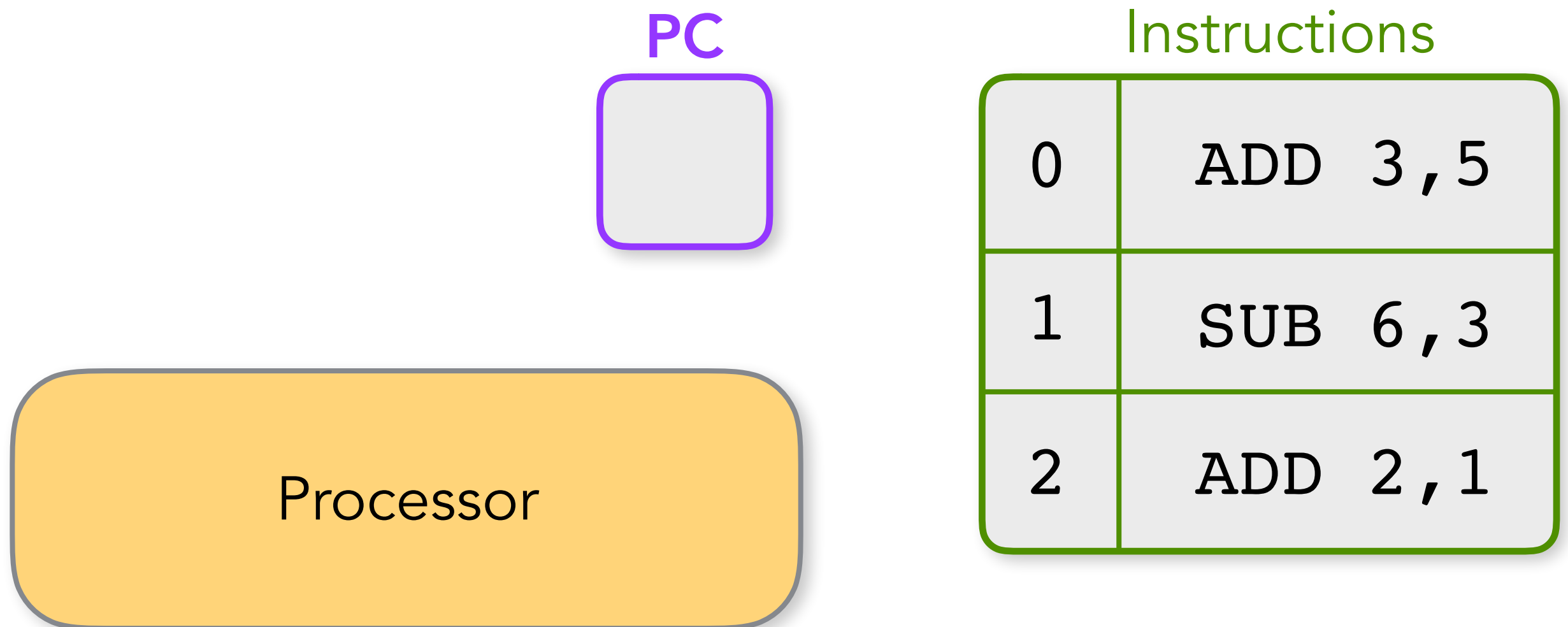
- The generated result went back into memory... this is now 'value added data'!
- The operation of saving the result is called '**write-back**'.
- In many machines, this is considered a separate operation.
- This leads to an extension of the paradigm:
"Fetch, execute, write-back"
- Or sometimes extended even further to:
"Fetch, decode, execute, write-back"
- This is the method adopted by most real processors.

The Program Counter

- How do I know what instruction to fetch?
- Most machines have a dedicated register called a **Program Counter** (PC).
- It contains a **number**, normally equal to the word-length of the processor (4-bit, 32-bit, etc).
- This number is interpreted as the **address of the next instruction** to be executed.
- Once an instruction has been executed, the PC is **incremented** and points to the following instruction.

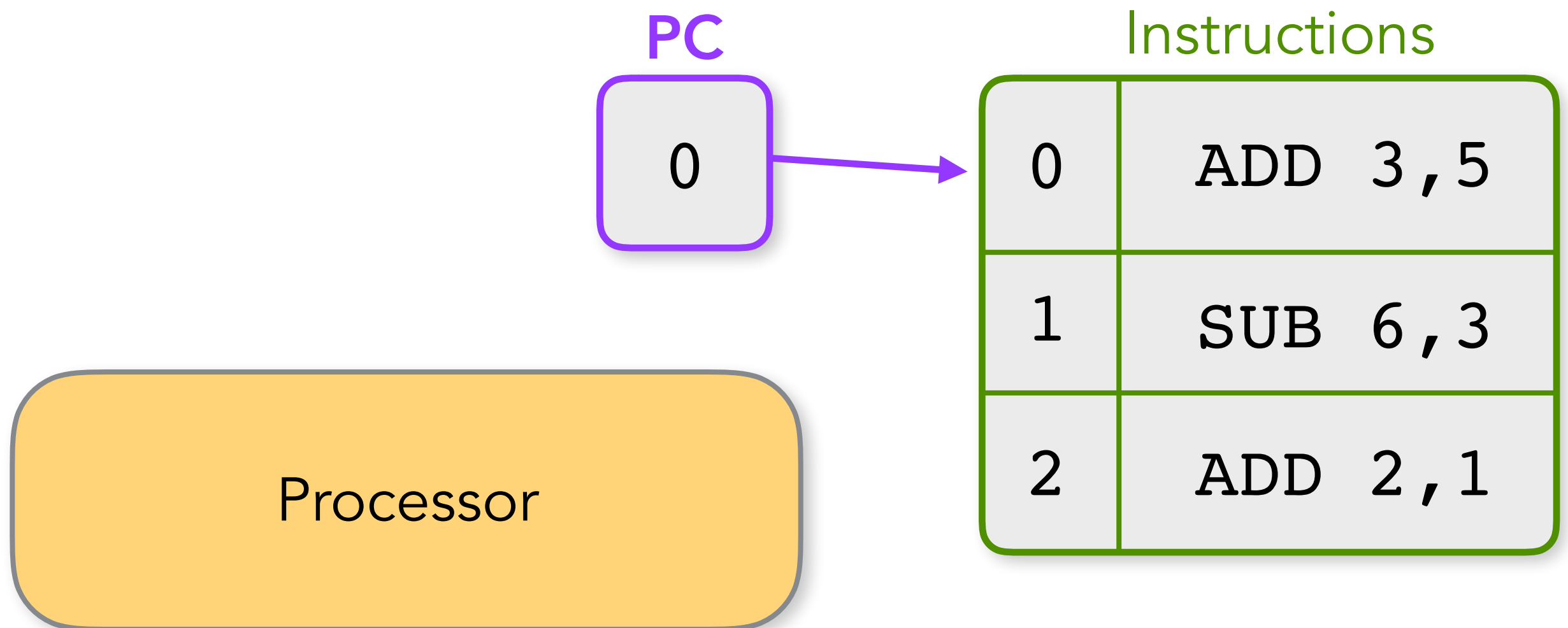
PC Operation

- Under normal program flow, the PC increments by one instruction address per instruction fetched.



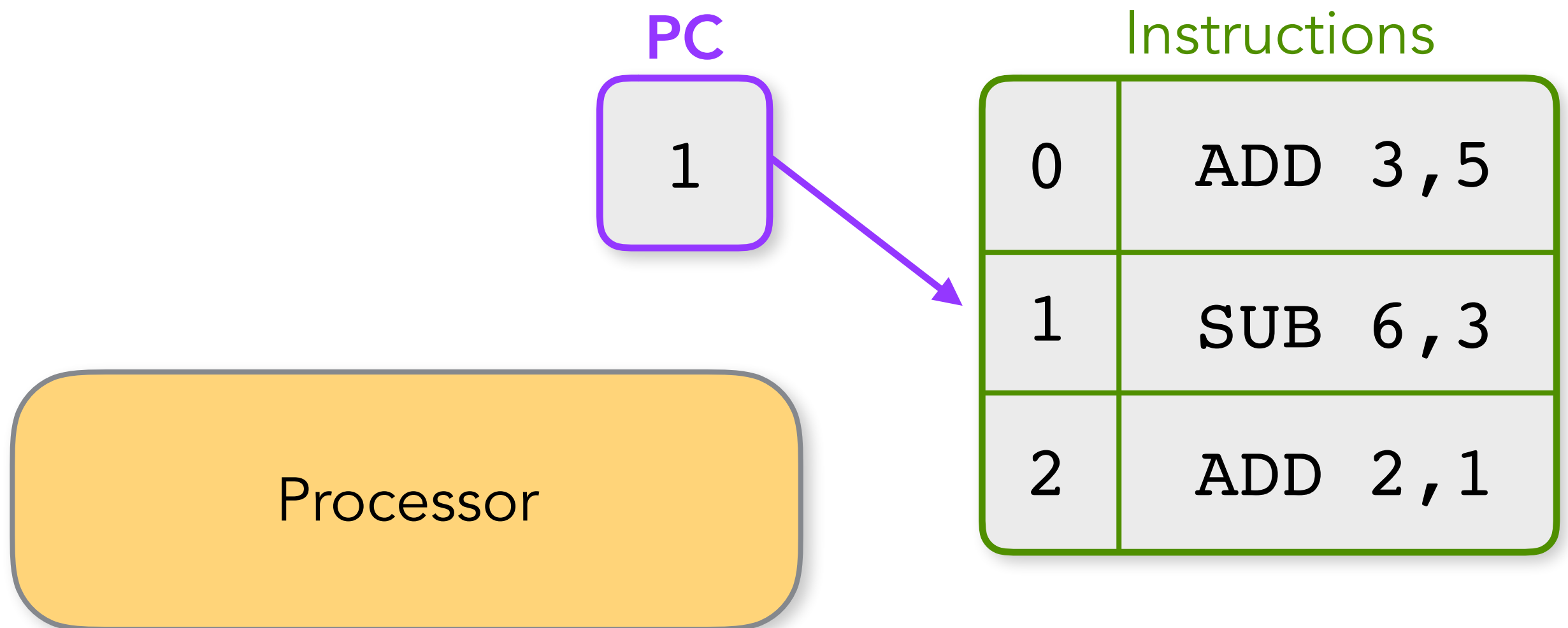
PC Operation

- Under normal program flow, the PC increments by one instruction address per instruction fetched.



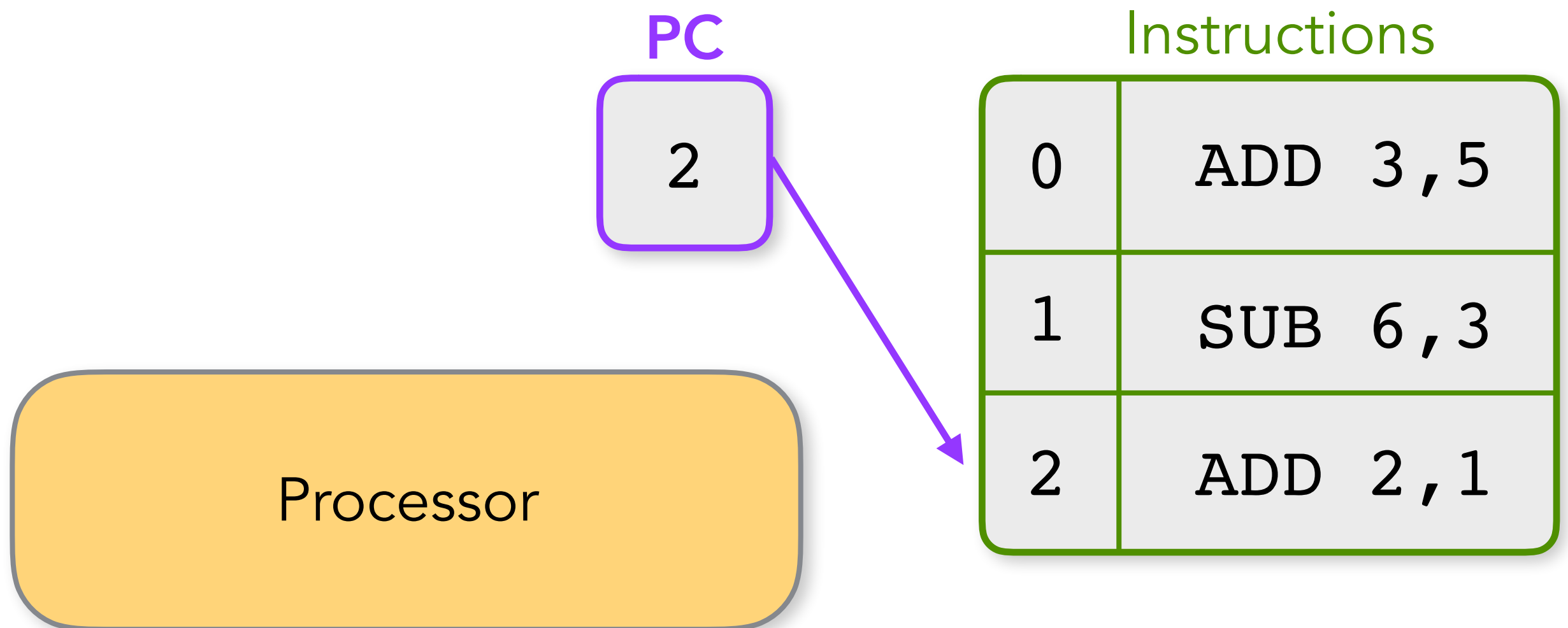
PC Operation

- Under normal program flow, the PC increments by one instruction address per instruction fetched.



PC Operation

- Under normal program flow, the PC increments by one instruction address per instruction fetched.



Controlling control

- The main source for control flow in a modern processor is the Program Counter (PC).
- The PC makes things happen.
It initiates all instruction executions.
- It does this by providing the address for instruction fetches.