



THE FETCH- DECODE- EXECUTE CYCLE

HOW THE CPU PROCESSES INSTRUCTIONS



The Fetch-Decode-Execute Cycle is the continuous process the CPU uses to run programs. It involves fetching an instruction from memory, decoding it to understand what action is needed, and then executing it. This cycle repeats for every instruction in a program.



VON NEUMANN ARCHITECTURE

The Von Neumann architecture, proposed by John von Neumann, uses a single memory unit for both instructions and data, connected to a single control unit via one address bus and one data bus. This design simplifies computer construction and forms the basis of many generalpurpose computers today, which follow the Fetch-Decode-Execute (FDE) cycle.

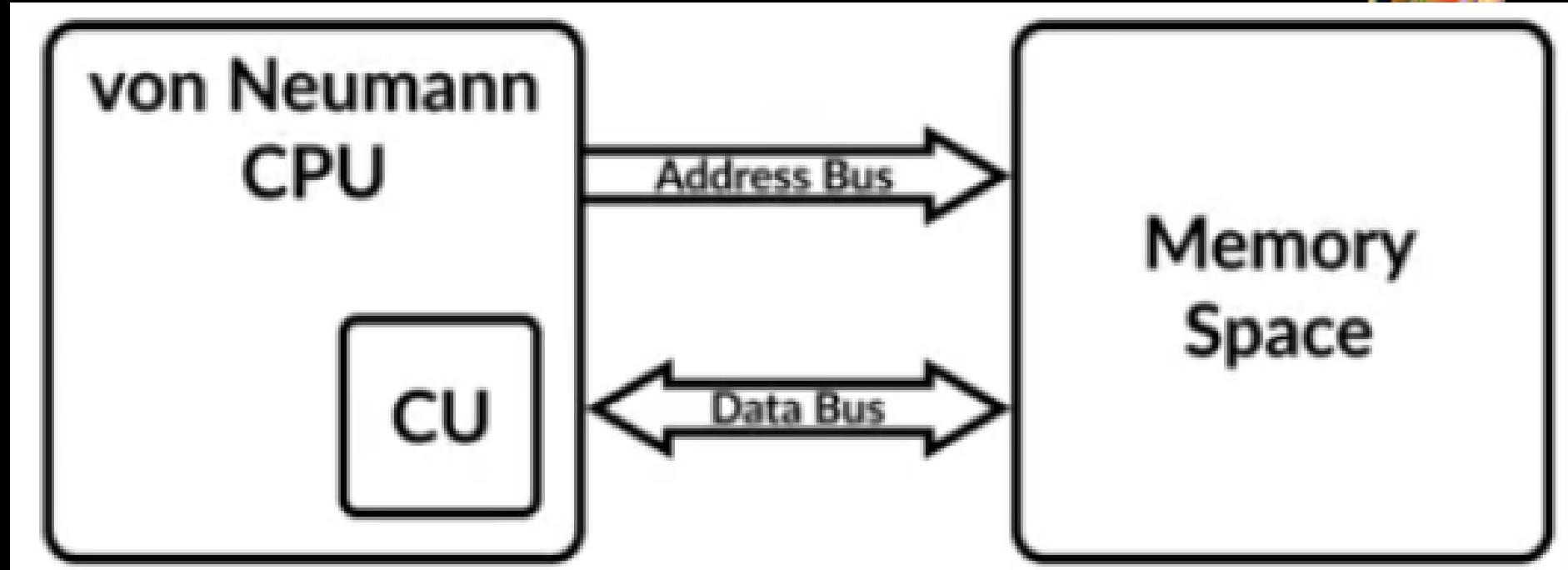
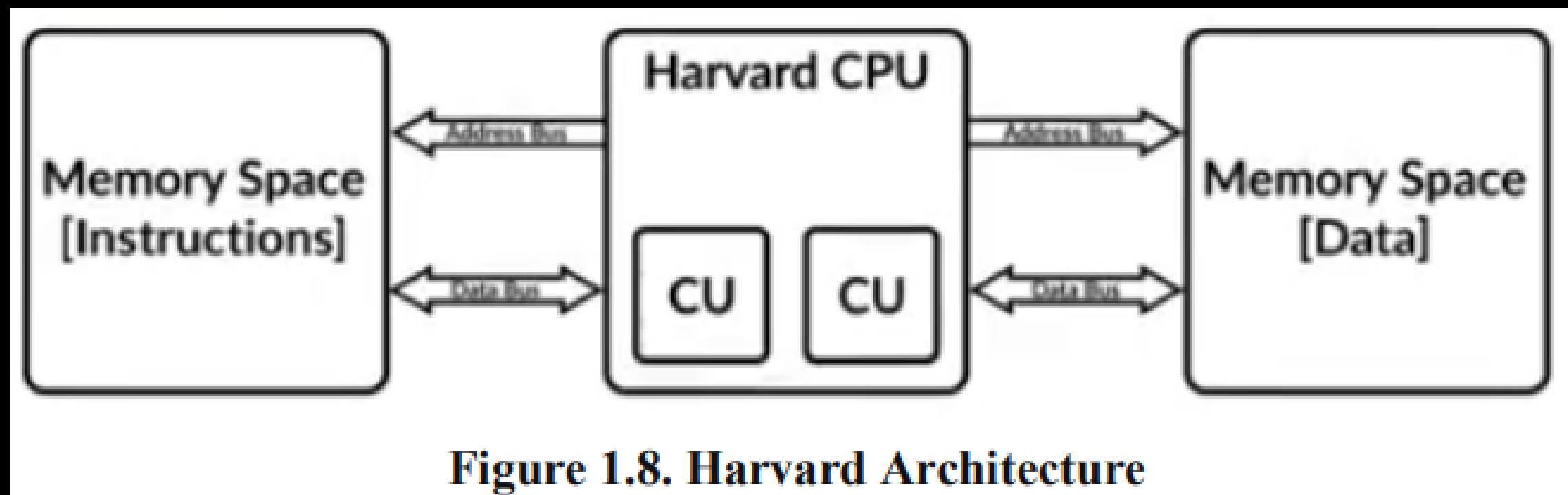


Figure 1.7. Von Neumann Architecture

HARVARD ARCHITECTURE

In contrast, the Harvard architecture has separate memory units and buses for instructions and data, allowing faster processing but with less flexibility. As a result, von Neumann architecture remains more common in general-purpose computing, while Harvard architecture is used where speed is critical.



FETCH-DECODE-EXECUTE CYCLE

The von Neumann fetch-decode-execute cycle starts, obviously, with the fetch part. The following are the step-by-step procedures involved in the FDE cycle, as shown in Figure 1.9

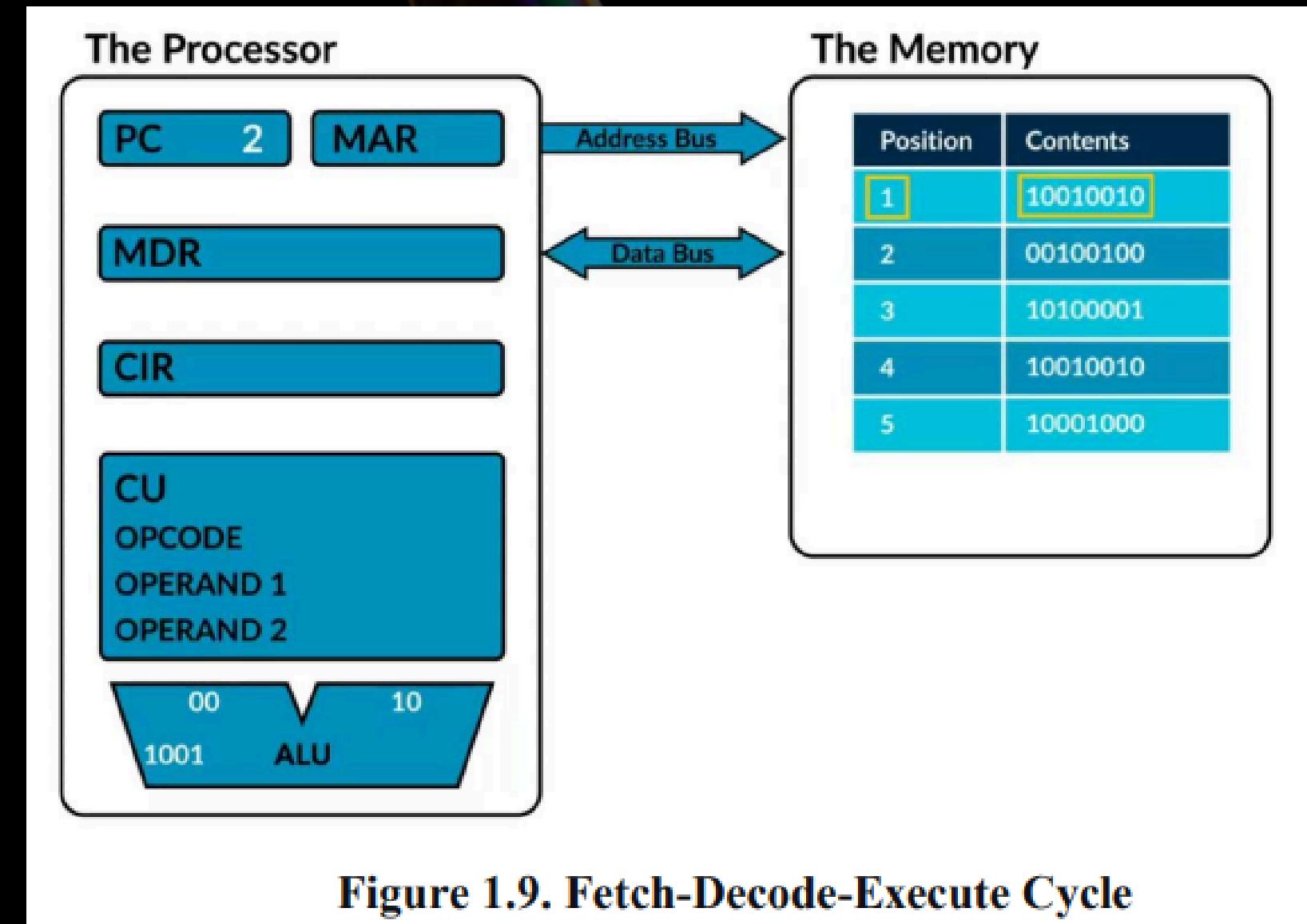


Figure 1.9. Fetch-Decomd-Execute Cycle

1. The fetch-decode-execute (FDE) cycle starts with the fetch stage:

- The program counter (PC) points to the first instruction.

- The address is copied to the memory address register (MAR).

- The program counter increments to the next instruction.

- The address is sent via the address bus, and the instruction is retrieved through the data bus into the memory data register (MDR).

- The instruction is then copied into the current instruction register (CIR).

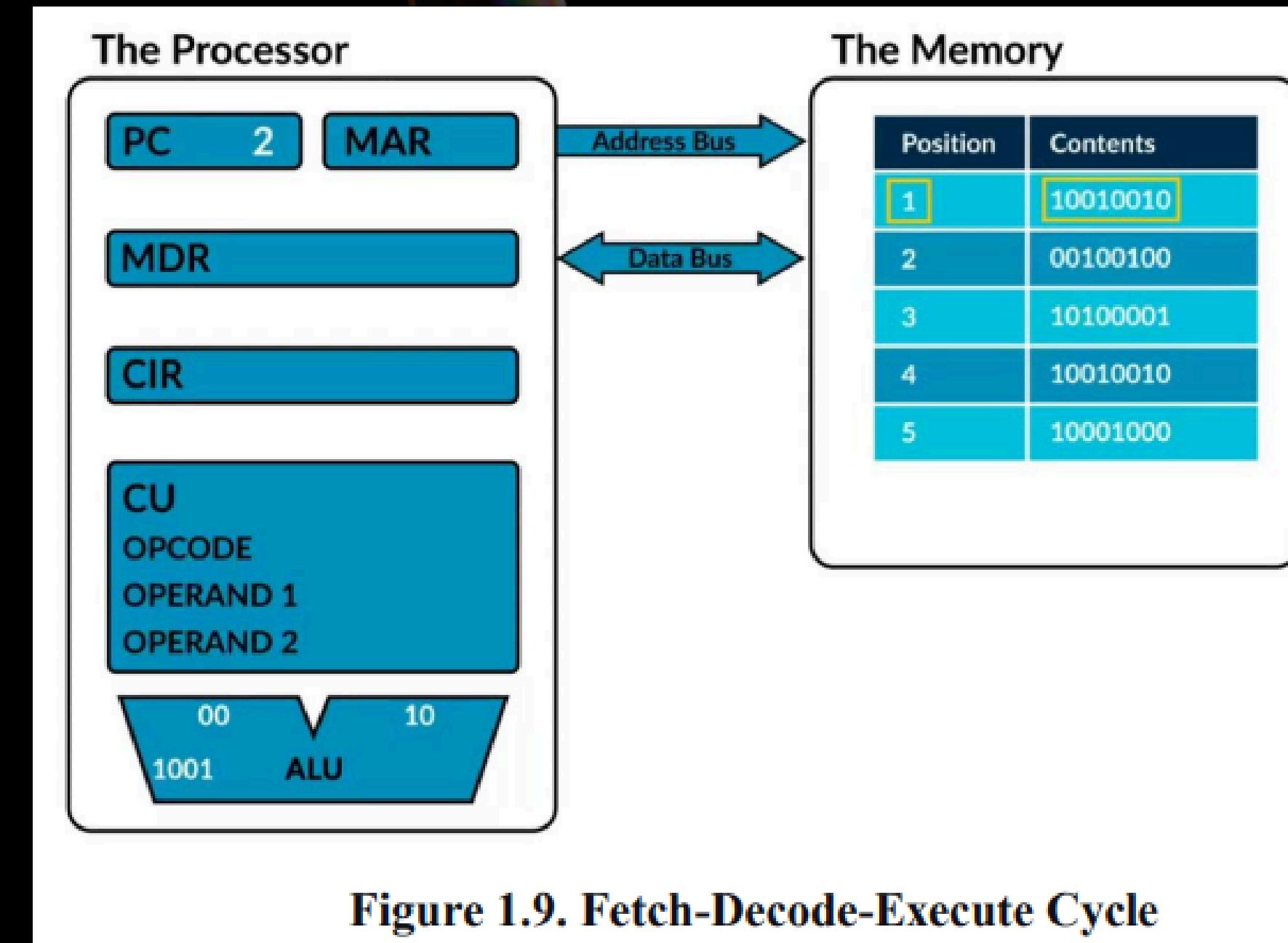


Figure 1.9. Fetch-Decomd-Execute Cycle

2. In the decode stage:

- The control unit (CU) decodes the instruction into its opcode and operands.

3. In the execute stage:

- The instruction is sent to the relevant parts of the processor, often the ALU, for processing.

4. The cycle then repeats for the next instruction.

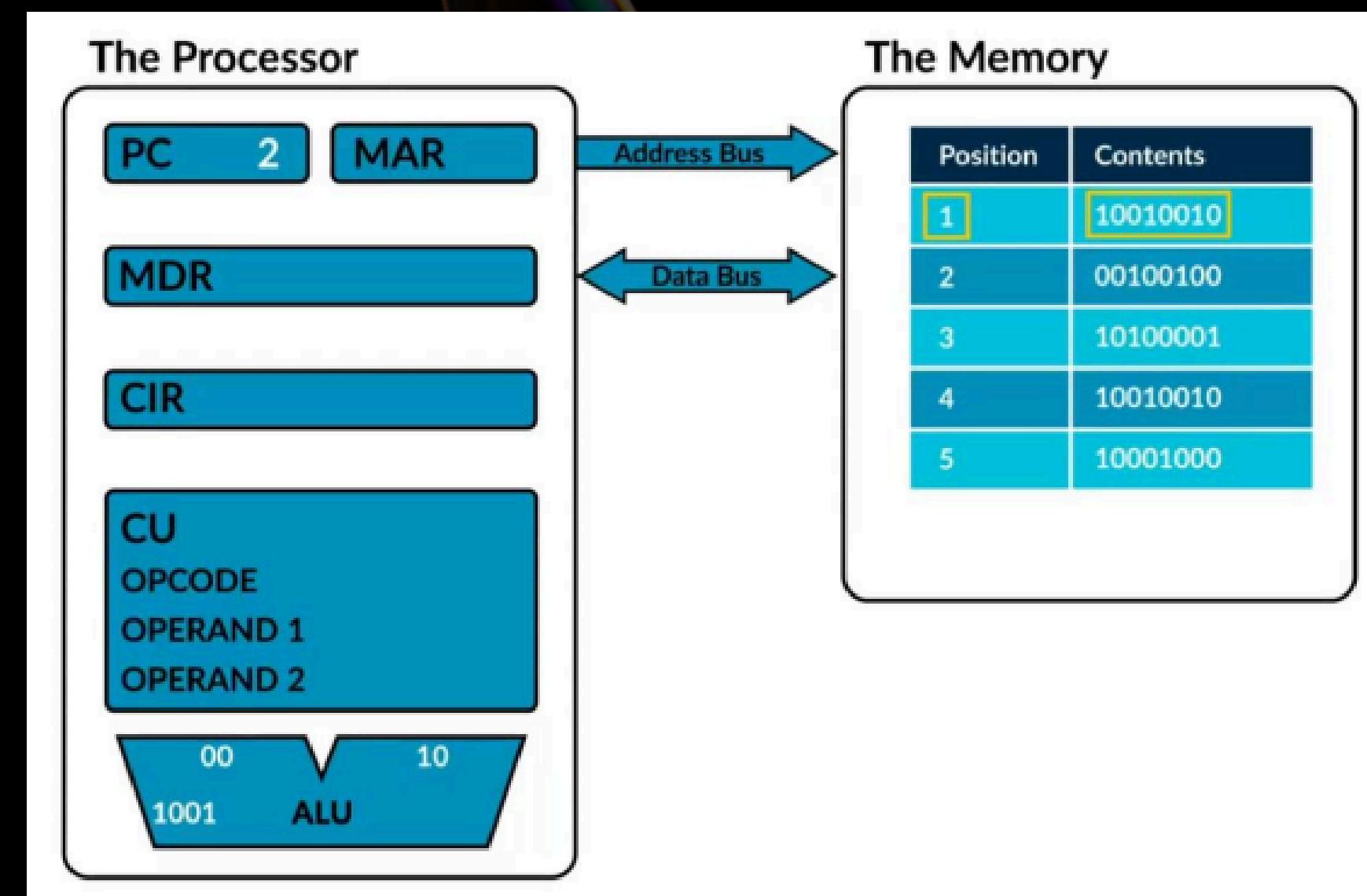
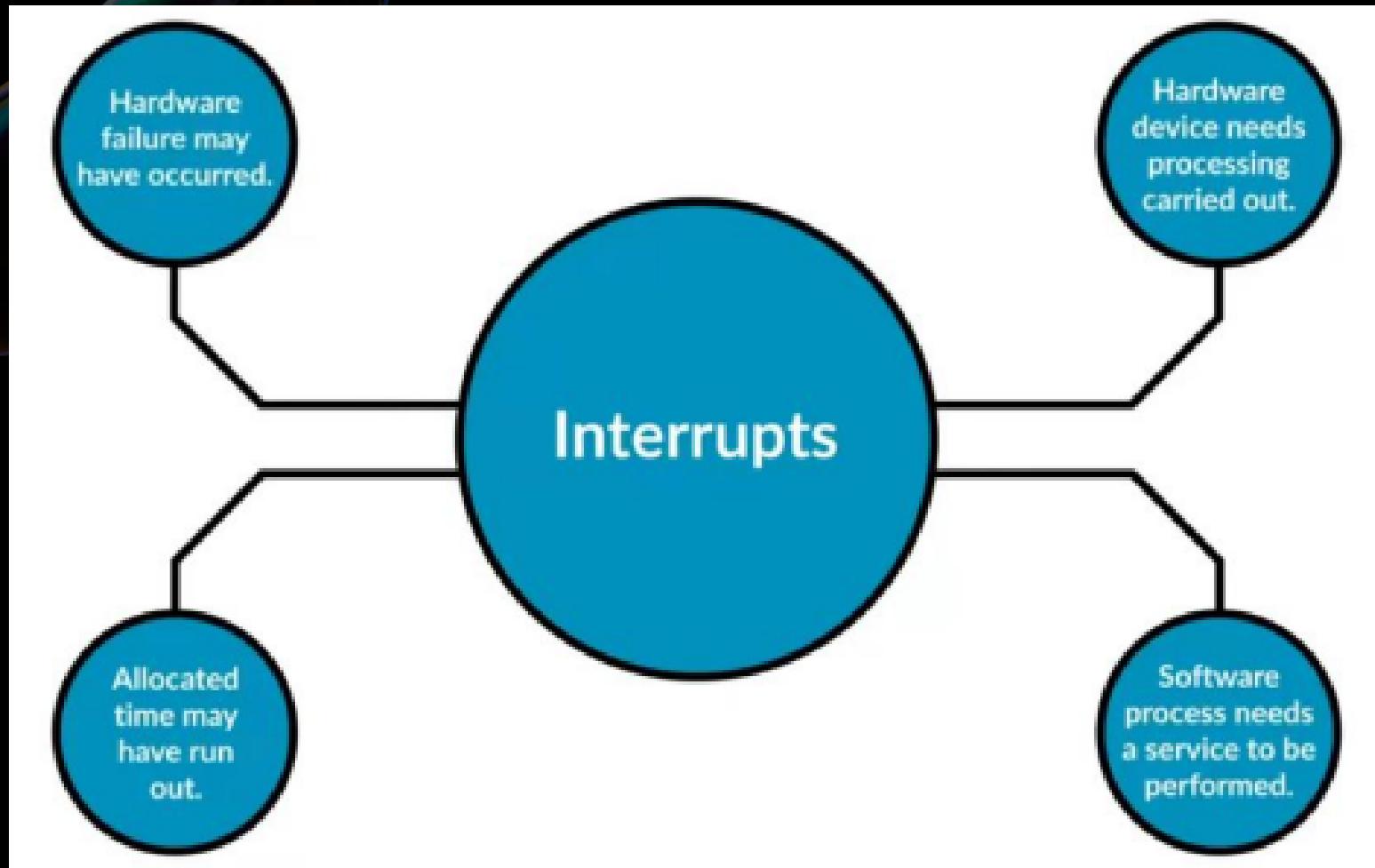


Figure 1.9. Fetch-Decide-Execute Cycle

There are certain occasions when the cycle might need to be interrupted.



These all trigger interrupts (as shown in Figure 1.10); the processor puts the current status codes, register contents, and program counter value onto a stack, and deals with the new process that needs carrying out.

A stack can be visualized like a stack of plates; you put something on the top (a plate, or a register value) and then can't access what's underneath until you take off the top plate (or register value). This is known as a FILO—first in, last out—data structure. When done, the values are copied back off the stack and processing can carry on. The reason we use a stack instead of just copying to a "spare" set of registers is that this new process may itself be interrupted.

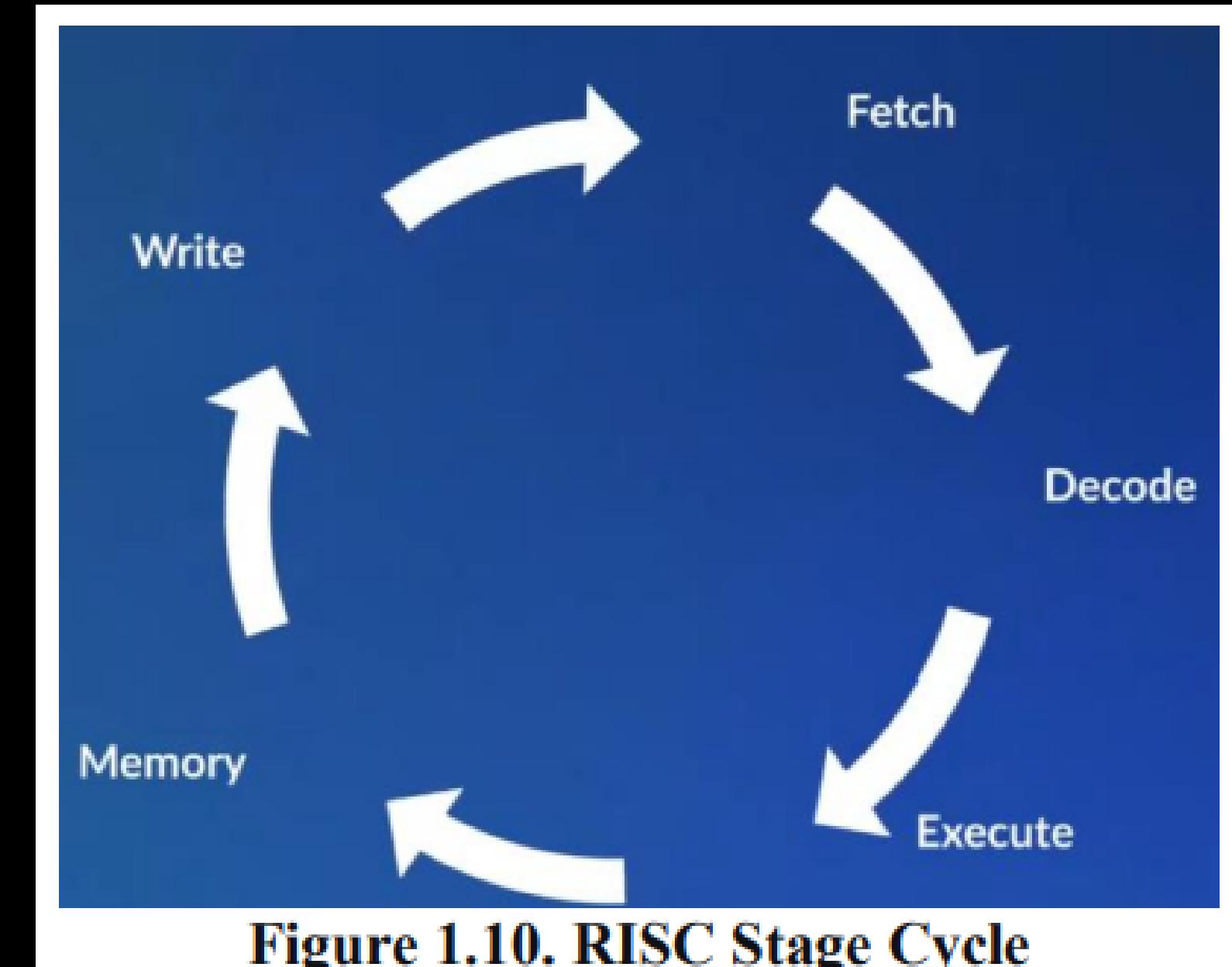


Figure 1.10. RISC Stage Cycle

Some processors—such as RISC (reduced instruction set computer processors) have a five stage cycle. The fetch, decode, and execute are the same as the von Neumann cycle. Rather than starting a new cycle to fetch any data needed from memory, this is done in the memory part of the cycle, with the writing of any results into a destination location being completed in the final write phase.



Microprocessor Systems

THANK YOU

for your time and attention

