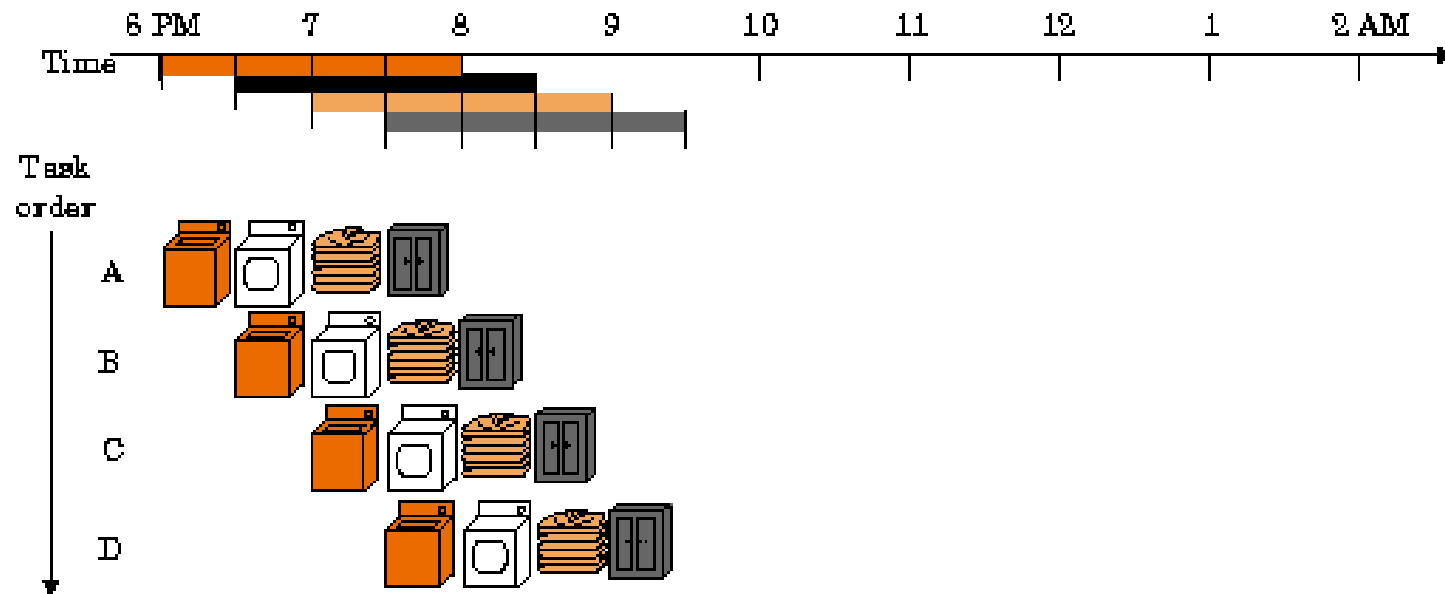
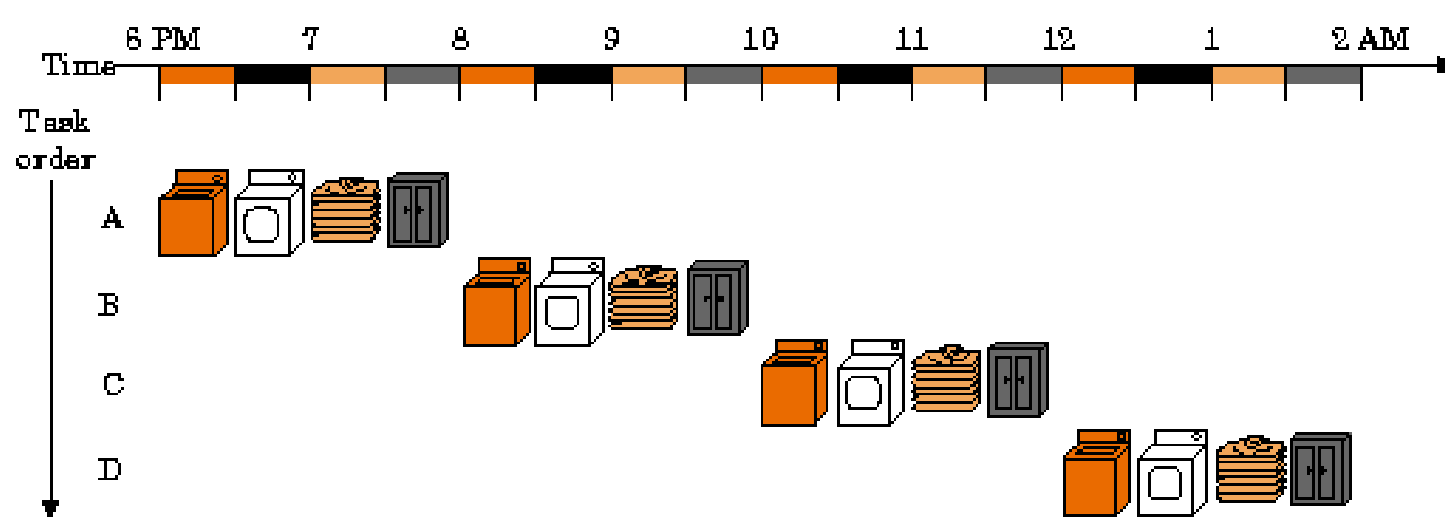


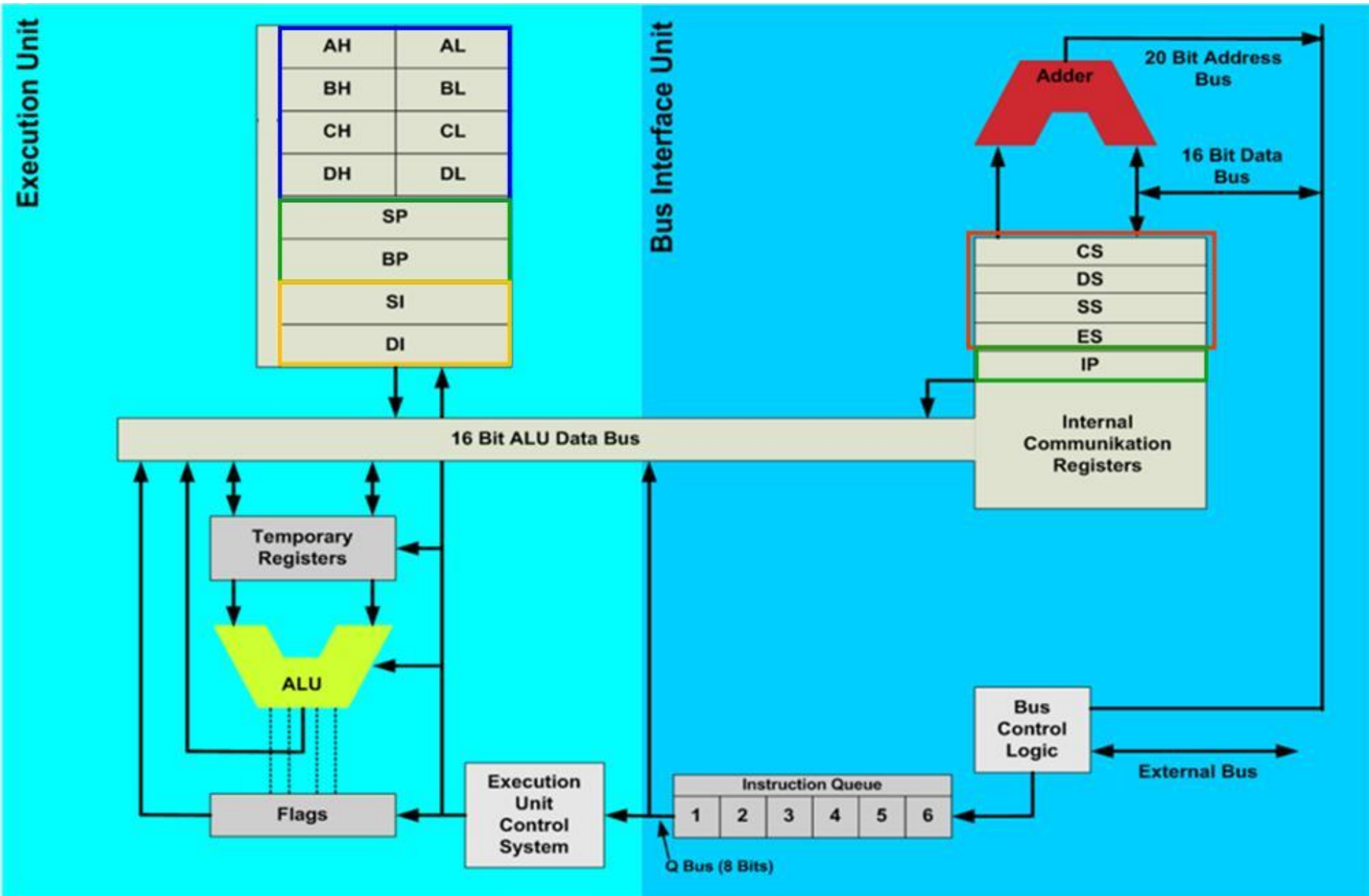
8086 Pipelining



Trivia

- For example, an x86 MOV instruction requires a two-byte opcode.
- 8086 instruction length is not fixed like MIPS. 8086 Instruction length depends on the instruction. It is a minimum of 1 byte and a maximum of 6 bytes long.

8086 Internal Architecture



8086 Internal Architecture

The **BIU** provides hardware functions, including generation of the memory and I/O addresses for the transfer of data between itself and the outside world.

The **EU** receives program instruction codes and data from the **BIU**, executes these instructions, and stores the results in the general registers. By passing the data back to the **BIU**, data can also be stored in a memory location or written to an output device.

- The **EU** has no connection to the system buses. It receives and outputs all of its data through the **BIU**. The execution unit, or **EU**, handles the arithmetic and logical operations on the data and has a 6-byte first-in-first-out (FIFO) instruction queue.
- The main linkage between the two functional blocks is the instruction queue, with the **BIU** looking ahead of the current instruction being executed in order to keep the queue filled with instructions for the **EU** to decode and operate on.

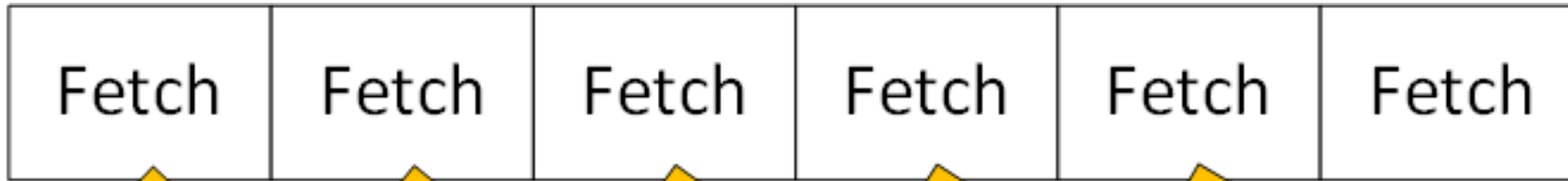
Pipelining

- In 8086, to speed up the execution of program, the instructions fetching and execution of instructions are overlapped each other. This technique is known as pipelining. In pipelining, when the n th instruction is executed, the $n+1$ th instruction is fetched and thus the processing speed is increased.
- The organization of the processor into a separate BIU and EU allows the **fetch** and **execute** cycles **to overlap**. In the beginning, the CS:IP is loaded with the required address from which the execution is to be started.

The Fetch and Execute Cycle of 8086 Processor

1. Register IP is incremented by one to prepare for the next instruction fetch.
2. Once inside the BIU, the instruction is passed to the **queue**: a first-in/first-out storage register sometimes likened to a **pipeline**. Assuming that the queue is initially empty.
3. The EU immediately draws this instruction from the queue and begins execution.
4. While the EU is executing this instruction, the BIU proceeds to fetch a new instruction. Depending on the execution time of the first instruction, the BIU may fill the queue with several new instructions before the EU is ready to draw its next instruction.
5. The cycle continues, with the BIU filling the queue with instructions and the EU executing these instructions.

BIU



EU

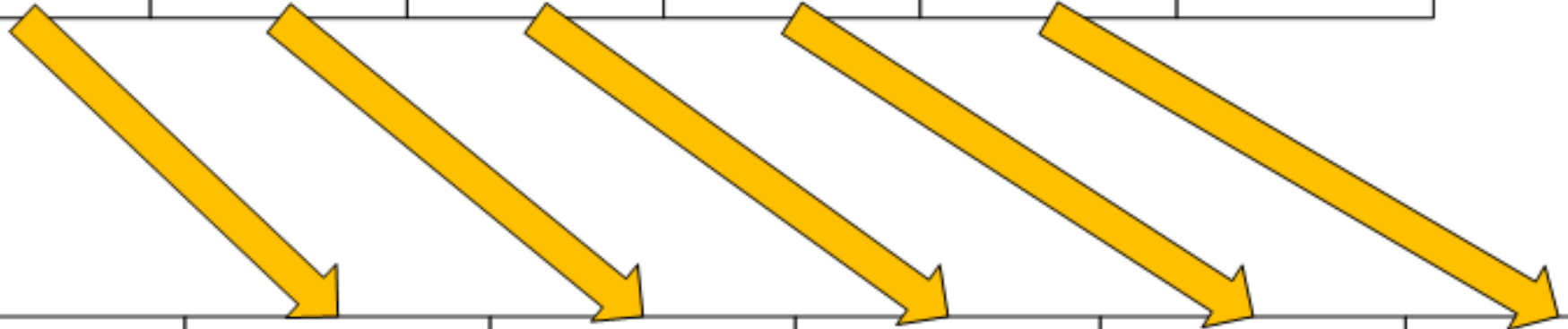
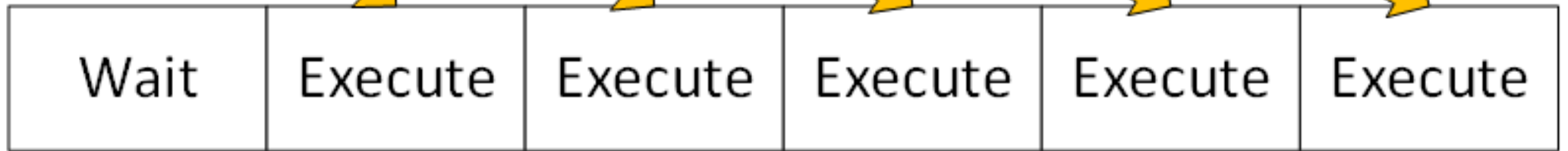


Figure: Block diagram showing Pipelining (Fetch and Execution cycle) in 8086 microprocessor

Instruction execution cycle

The instruction execution cycle can be clearly divided into three different steps.

- **Fetch**

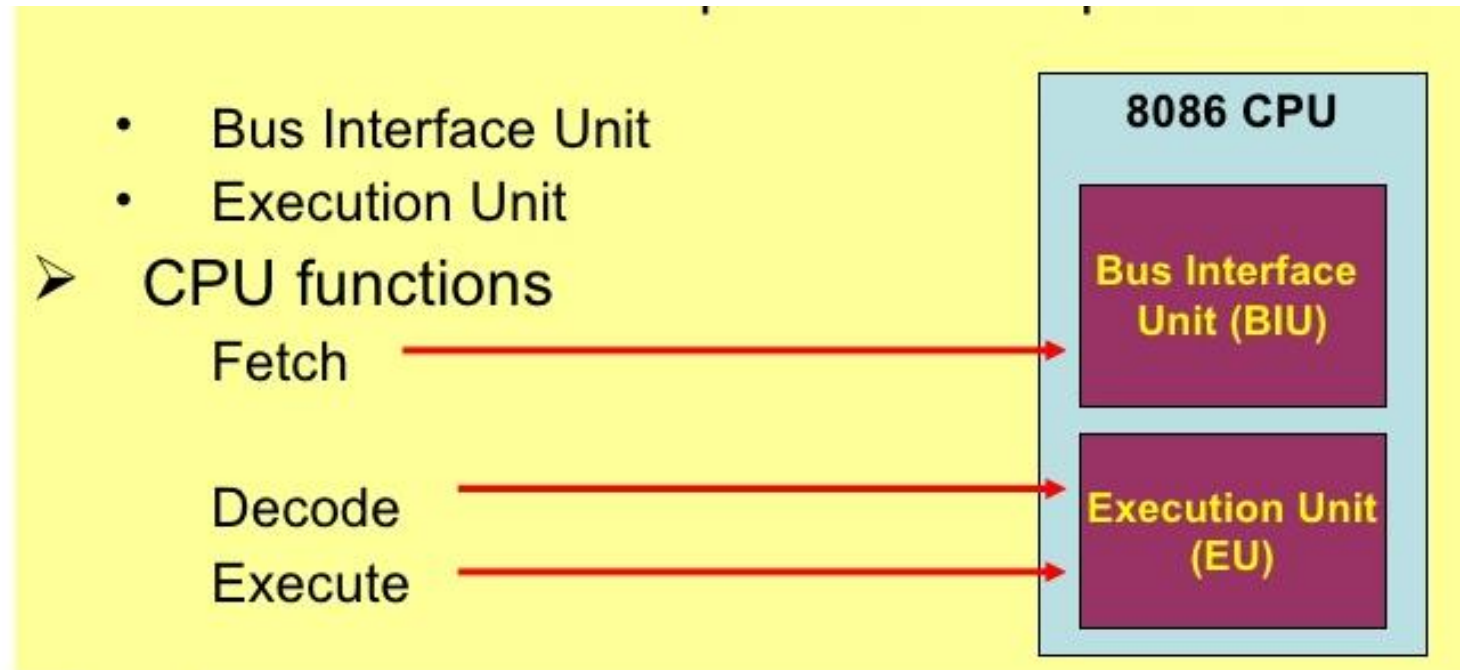
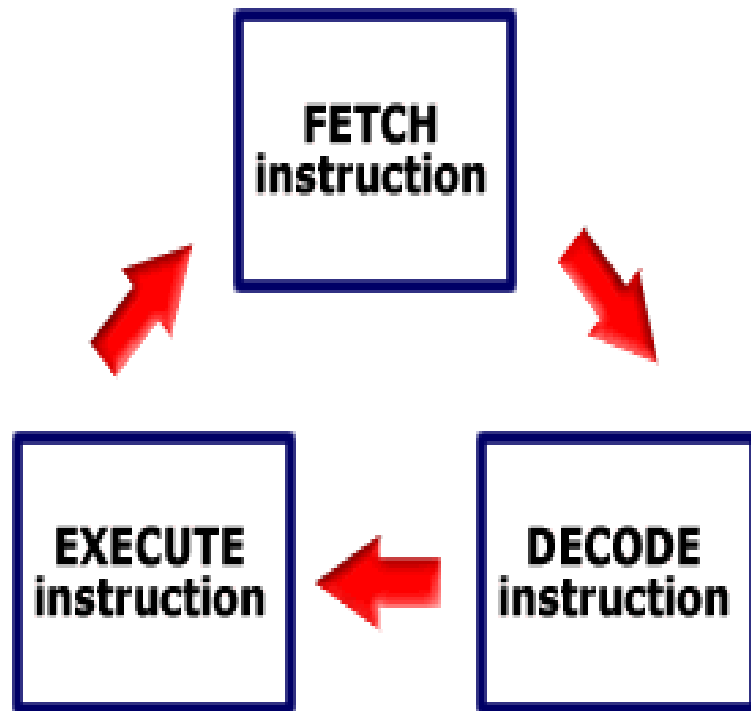
In the fetch step, an instruction is copied from the memory of the computer into the queue.

- **Decode**

In decode step, the microprocessor figures out what actions need to be carried out in order to execute the instruction.

- **Execute**

In execution step, the instruction is executed.



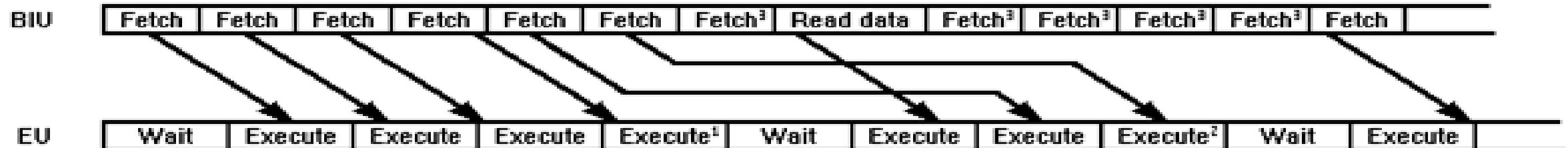
Conditions under which EU enter into the “Wait” state

- When an instruction requires access to a **memory location**. The BIU must **suspend** fetching instructions and access the address of this memory location. After waiting for the memory access, the EU can resume executing instruction codes from the **queue**, and the BIU can resume filling the queue.
- When the instruction to be executed is a **jump** instruction. In this case, control is to be transferred to a new address. The **EU must wait** while the instruction at the jump address is fetched. Any bytes presently in the queue must be discarded (they are overwritten).

Fetch and Execute Cycle

| | | | | | | | | |
|-------|---------|-------|---------|-------|---------|-------|---------|--|
| Fetch | Execute | Fetch | Execute | Fetch | Execute | Fetch | Execute | |
|-------|---------|-------|---------|-------|---------|-------|---------|--|

FETCH AND EXECUTE CYCLE

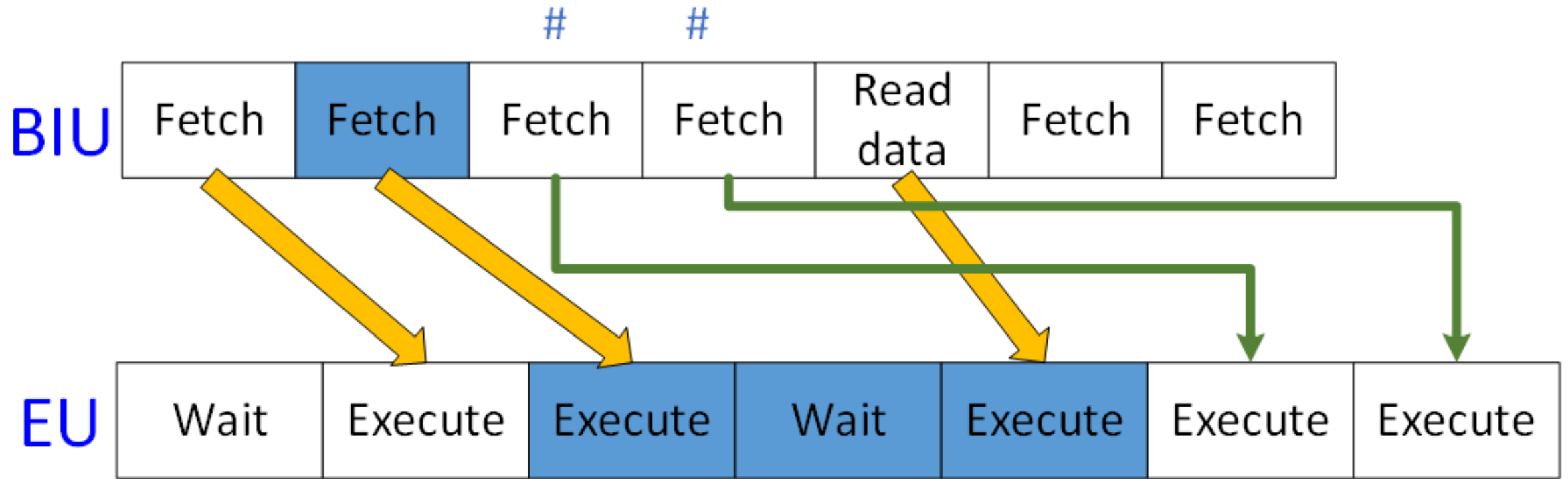


¹This instruction requires a request for data not in the queue

²Jump instruction occurs

³These bytes are discarded

INSTRUCTION PIPELINE



This instruction requires a request not in the queue

BIU



EU



* : 'Jump' instruction

: Instruction presently in the queue will be discarded