Computer Organization and Architecture

CH 03: A TOP-LEVEL VIEW OF COMPUTER FUNCTION AND INTERCONNECTION

Content

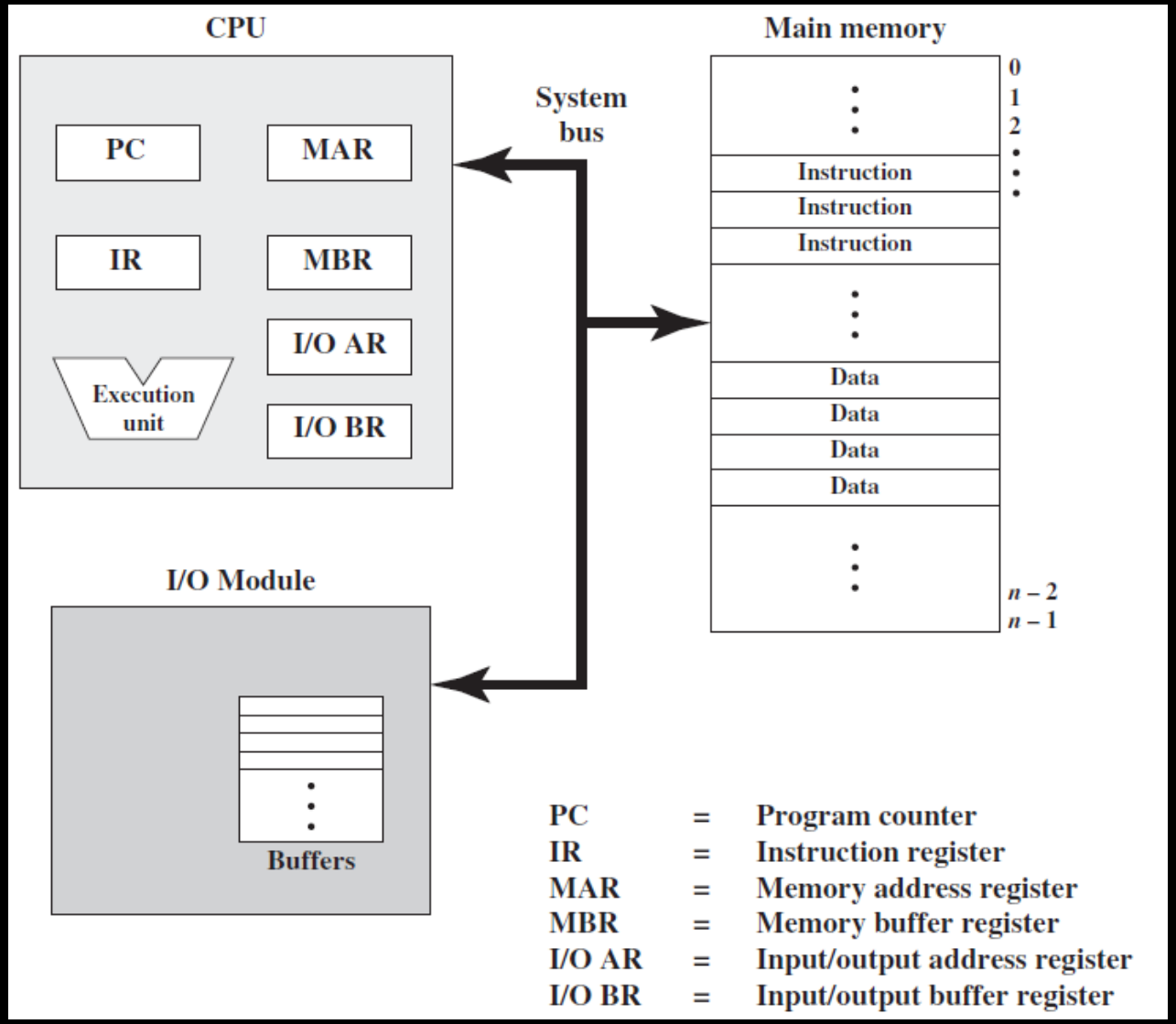
CH 03
Computer Components
Interrupts
Interconnection Structures
Bus Interconnection
Elements of Bus Design: Timing
PCI

Computer Components

- The von Neumann architecture is based on three key concepts:
  1. Data and instructions are stored in a single read–write memory.
  2. The contents of this memory are addressable by location, without regard to the type of data contained there.
  3. Execution occurs in a sequential fashion (unless explicitly modified) from one instruction to the next.

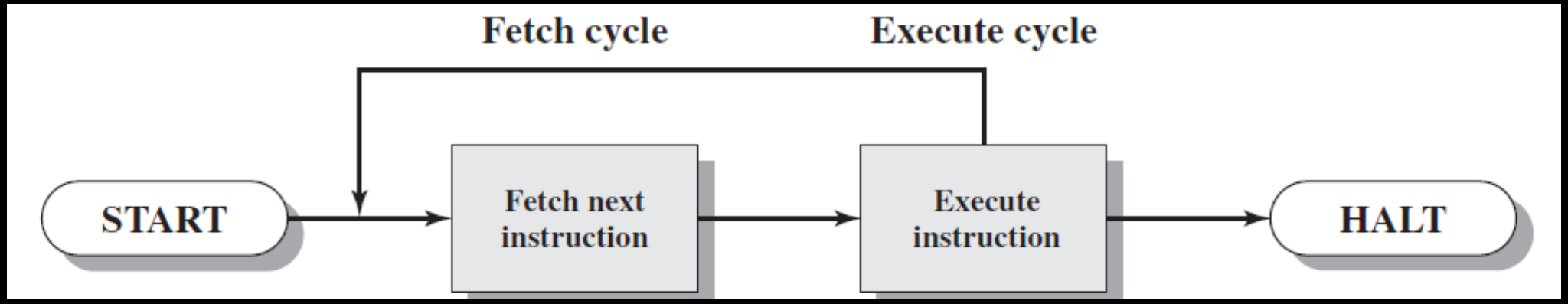
Computer Components

- Top level view of computer components



Computer Components

- Instruction processing steps:
  1. The processor reads (fetches) instructions from memory one at a time and executes each instruction.
  2. The instruction execution may involve several operations and depends on the nature of the instruction.



Computer Components

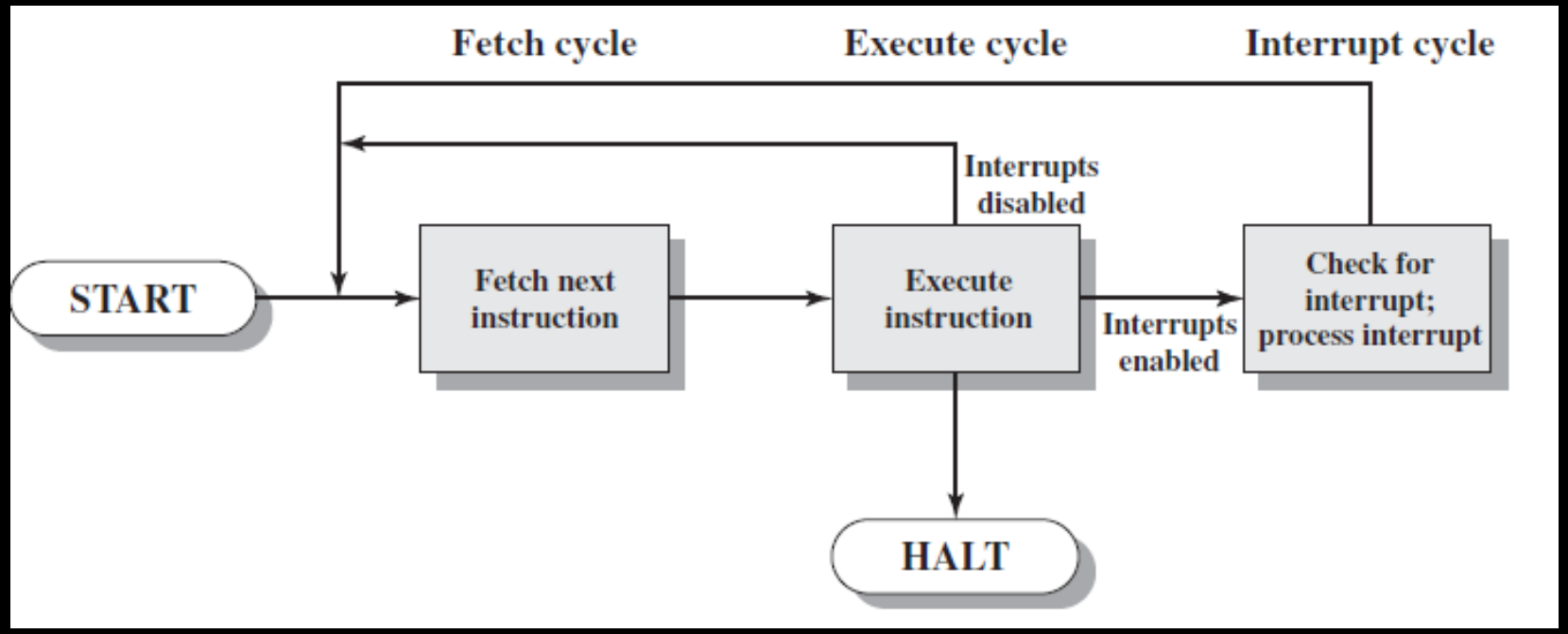
- Consider a computer in which each instruction occupies one 16-bit word of memory.
  1. The program counter is set to location 300. The processor will next fetch the instruction at location 300.
     - On succeeding instruction cycles, it will fetch instructions from locations 301, 302, 303, and so on.
  2. The fetched instruction is loaded in the processor into the instruction register (IR).
     - The instruction contains bits that specify the action the processor is to take.
  3. The processor interprets the instruction and performs the required action.

Content

CH 03
Computer Components
Interrupts
Interconnection Structures
Bus Interconnection
Elements of Bus Design: Timing
PCI

Interrupts

- With interrupts, the processor can be engaged in executing other instructions while an I/O operation is in progress.
  - Suppose that the processor is transferring data to a printer. After each write operation, the processor must pause and remain idle until the printer catches up.



Interrupts

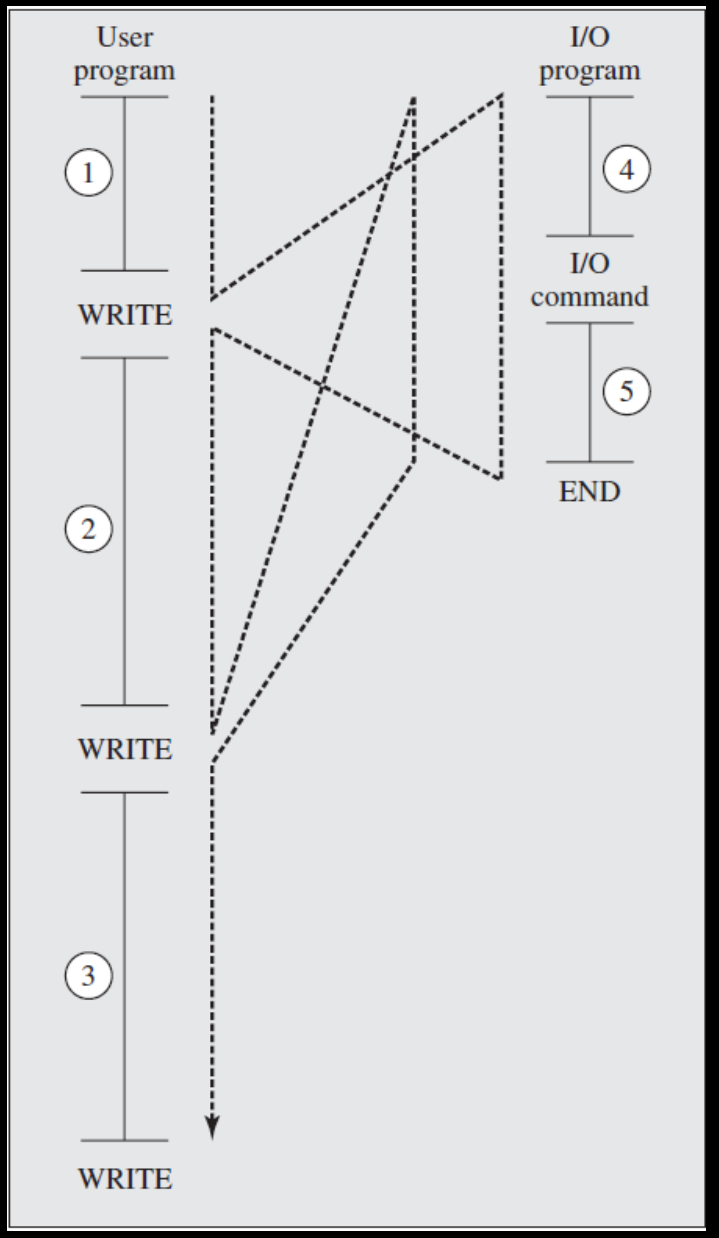
- Classes of interrupts

Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, or reference outside a user's allowed memory space.
Timer	Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis.
I/O	Generated by an I/O controller, to signal normal completion of an operation or to signal a variety of error conditions.
Hardware failure	Generated by a failure such as power failure or memory parity error.

Interrupts

Without interrupts

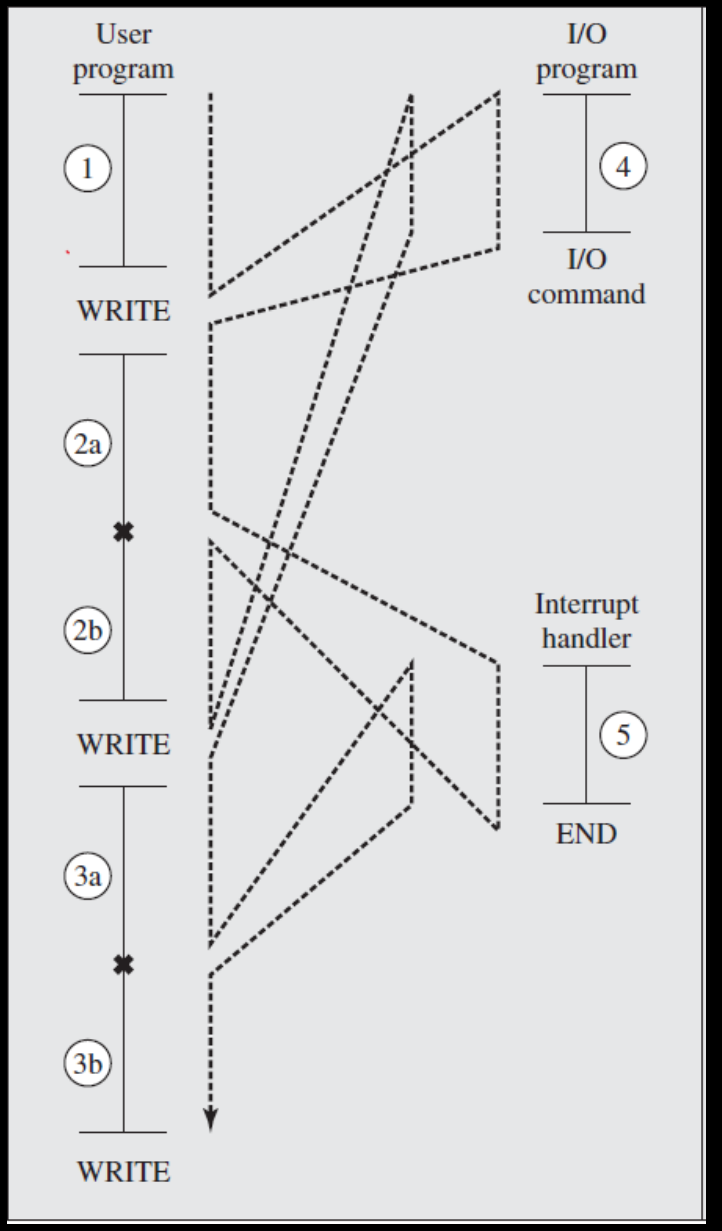
- The user program performs a series of WRITE calls interleaved with processing.
  - Code segments 1, 2, and 3 refer to sequences of instructions that do not involve I/O.
- The WRITE calls are to an I/O program that is a system utility and that will perform the actual I/O operation.
- Once the **I/O command** is issued, the program must wait for the I/O device to perform the requested function.



Interrupts

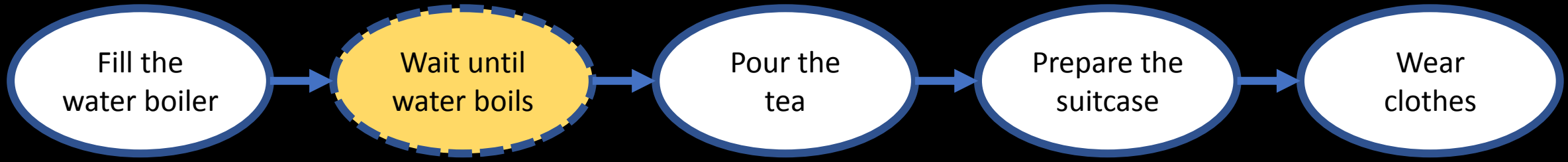
With interrupts

- The program reaches a point at which it makes a system call in the form of a WRITE call.
- The processor initiates the I/O program.
- Control returns to the user program. Meanwhile, the external device is busy accepting data from computer memory and printing it.
  - This I/O operation is conducted concurrently with the execution of instructions in the user program.
- When the I/O operation finishes, the I/O device sends an *interrupt request* signal to the processor.



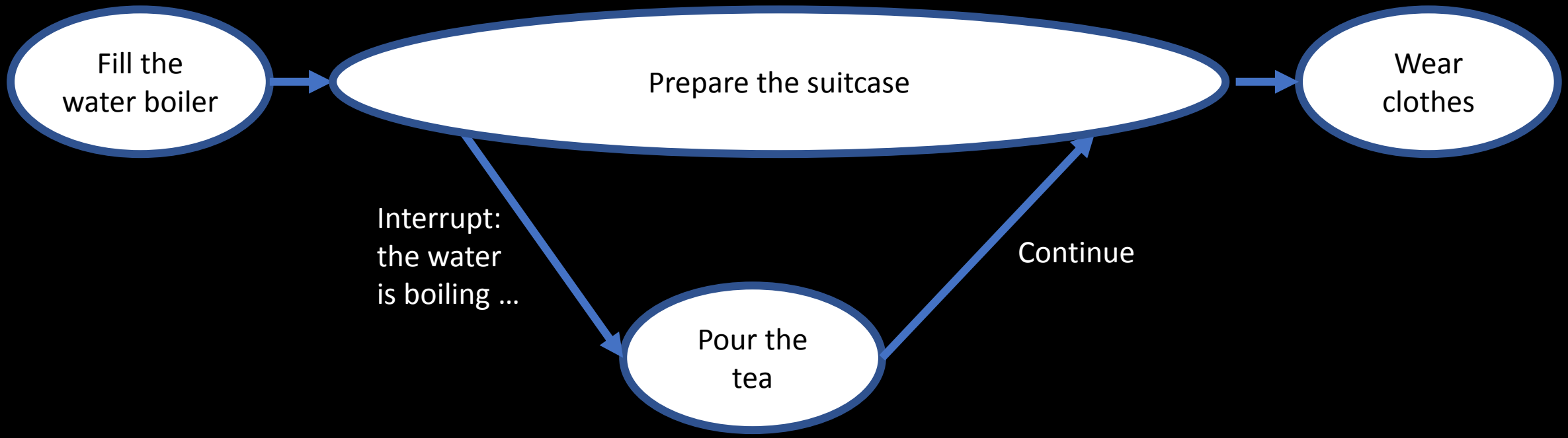
Interrupts

Travelling and prepare tea example. suppose you are preparing your suitcase and want to prepare a cup of tea (without interrupts).



Interrupts

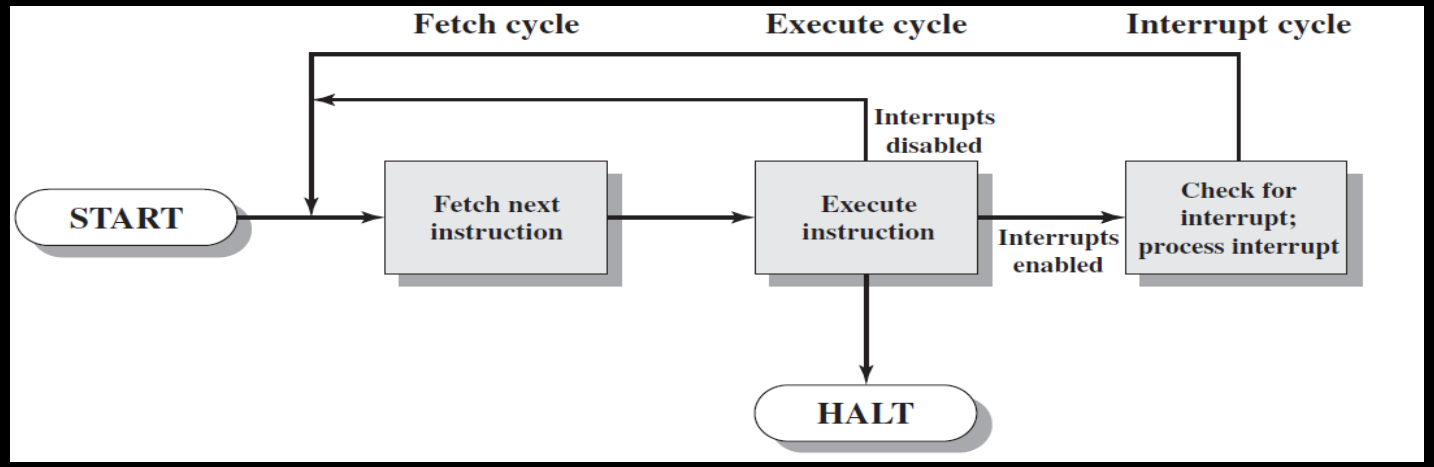
Travelling and prepare tea example. suppose you are preparing your suitcase and want to prepare a cup of tea (with interrupts).



Interrupts

If an interrupt is pending, the processor does the following:

- It suspends execution of the current program being executed and saves its context.
  - Saving the address of the next instruction to be executed and any other data relevant to the processor's current activity.
- It sets the program counter to the starting address of an interrupt handler routine.



Exercises

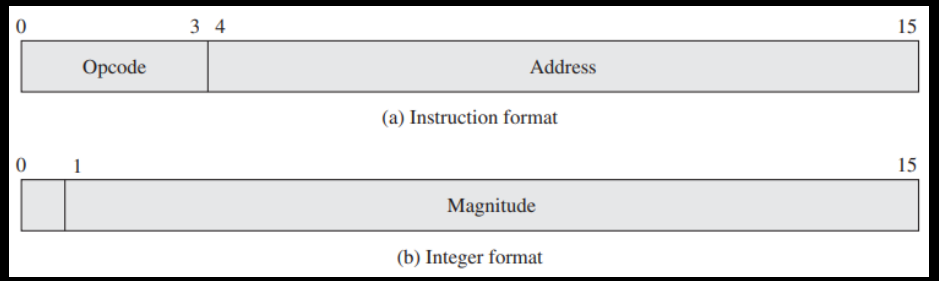
3.1 The hypothetical machine of Figure 3.4 also has two I/O instructions:

$$0011 = Load\ AC\ from\ I/O$$

$$0111 = Store\ AC\ to\ I/O$$

In these cases, the 12-bit address identifies a particular I/O device. Show the program execution (using the format of Figure 3.5) for the following program:

1. Load AC from device 5.
2. Add contents of memory location 940.
3. Store AC to device 6.



Assume that the next value retrieved from device 5 is 3 and that location 940 contains a value of 2.

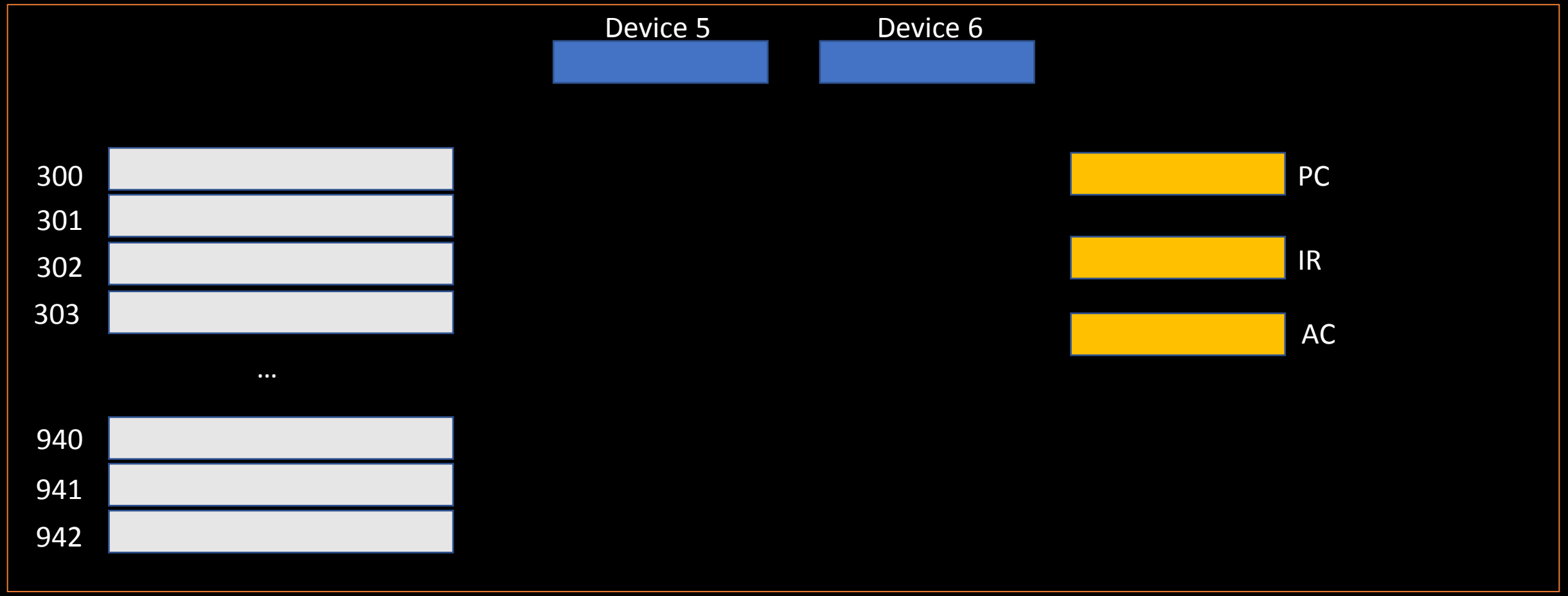
Exercises

Given the instruction set (opcodes):

- 0001 = Load AC from memory (1 in hex)
- 0010 = Store AC to memory (2 in hex)
- 0101 = Add to AC from memory (5 in hex)
- 0011 = Load AC from I/O (3 in hex)
- 0111 = Store AC to I/O (7 in hex)

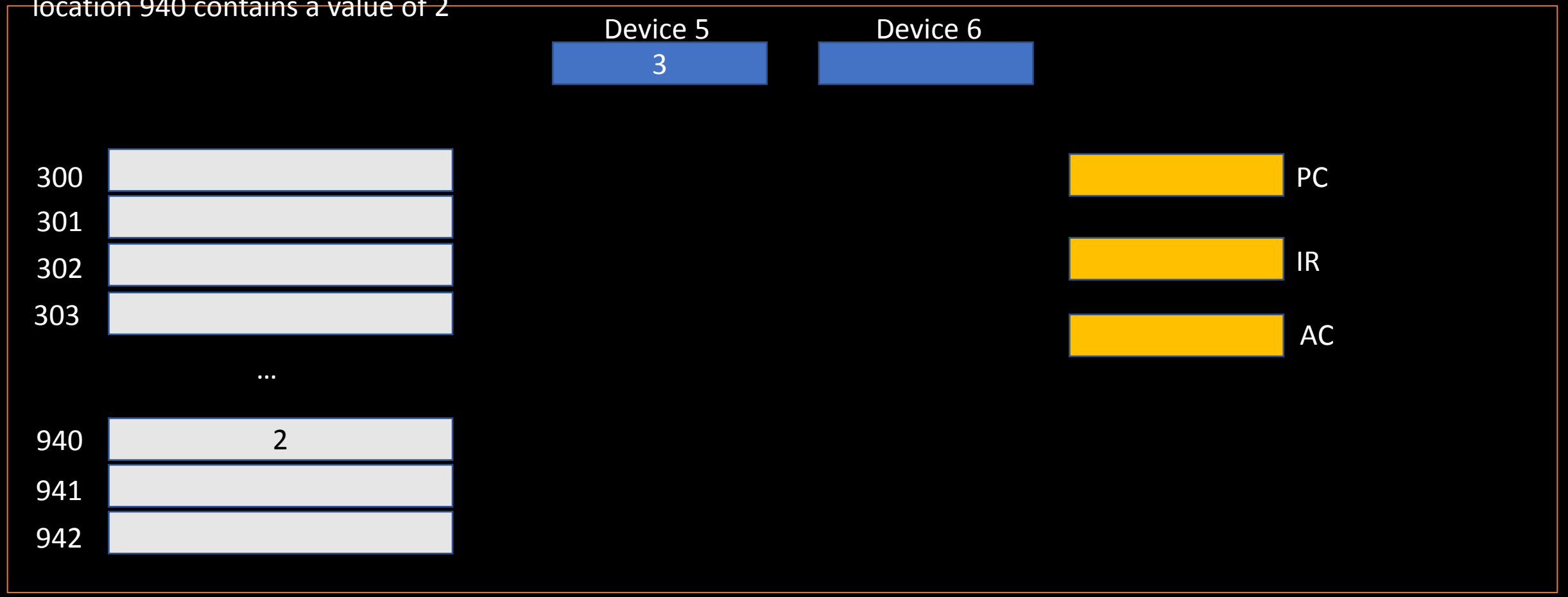
- Given that device 5 has the value 3
- Given location 940 has the value 2
- Output: memory and registers content at each step

Exercises



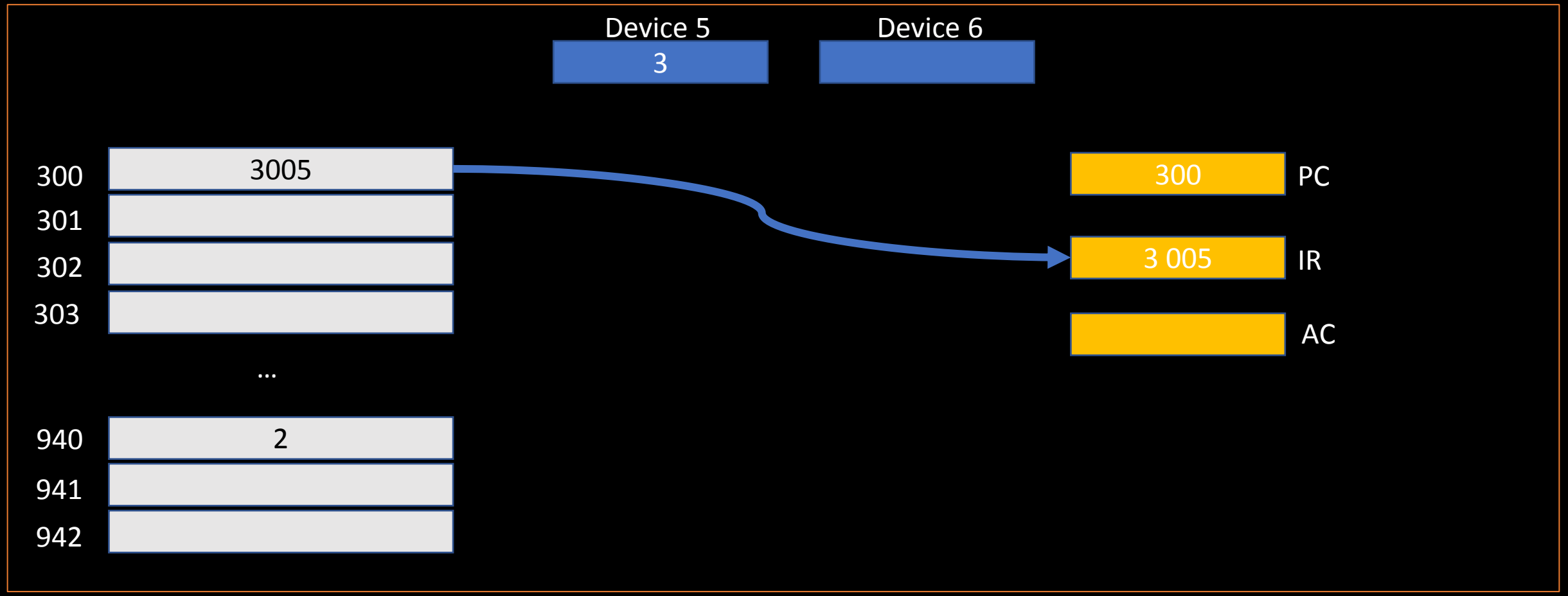
Exercises

Assume that the next value retrieved from device 5 is 3
location 940 contains a value of 2



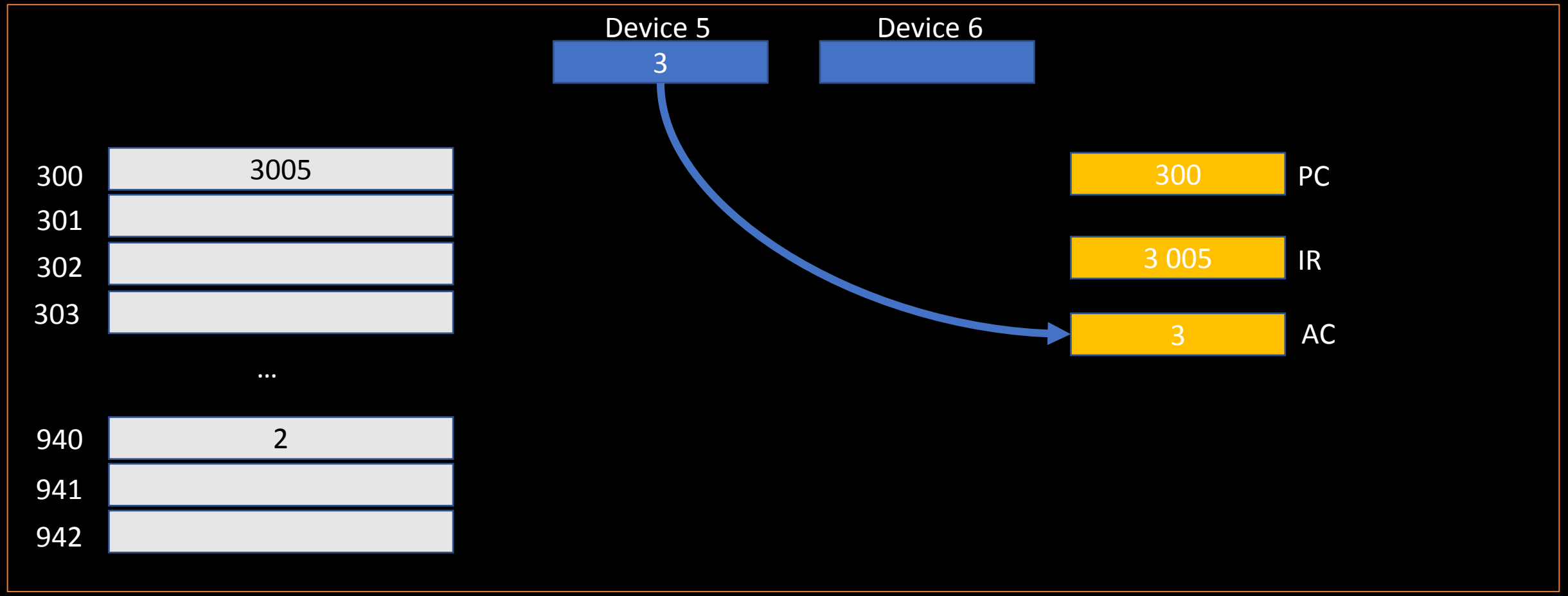
Exercises

Load AC from device 5



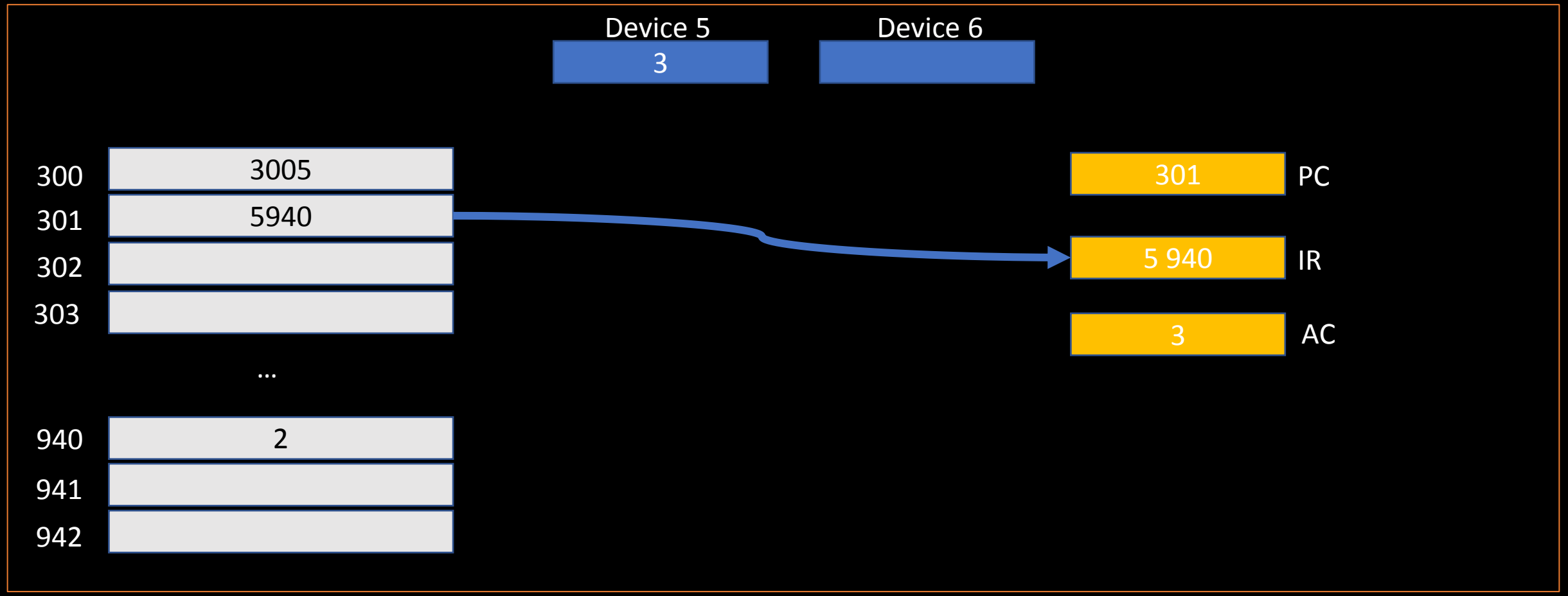
Exercises

Load AC from device 5



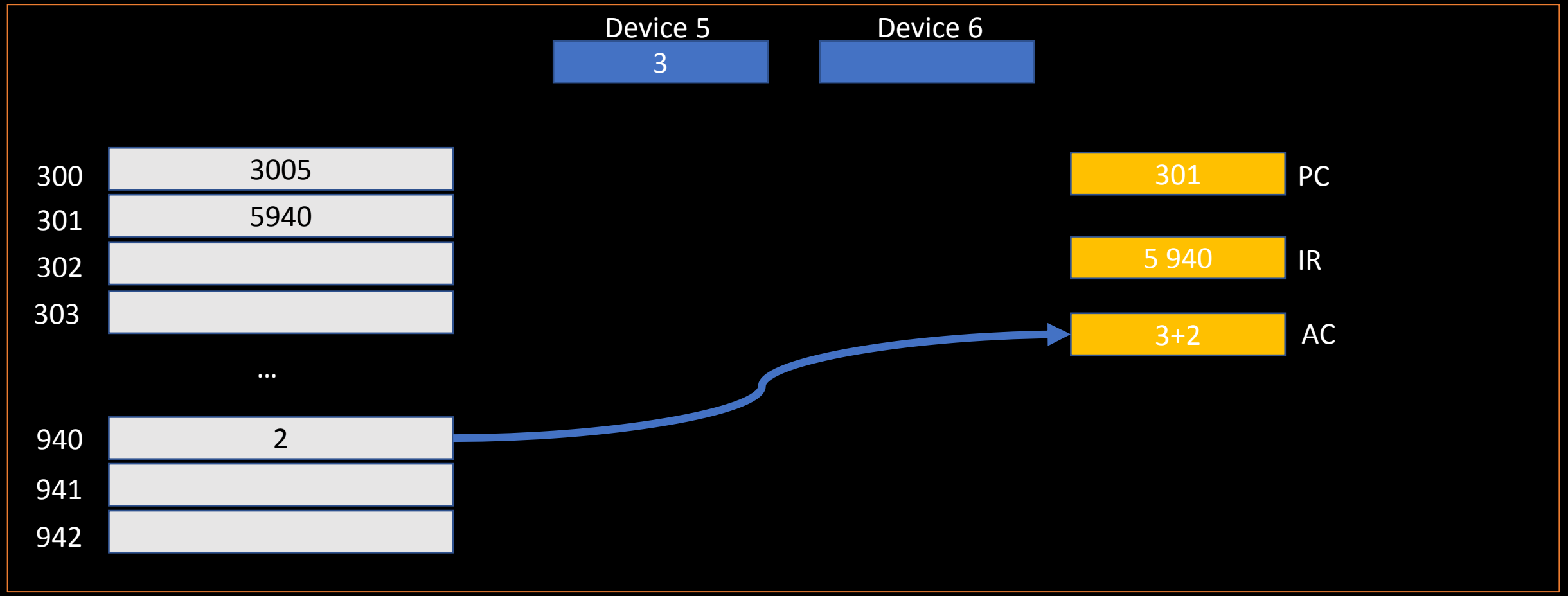
Exercises

Add contents of memory location 940



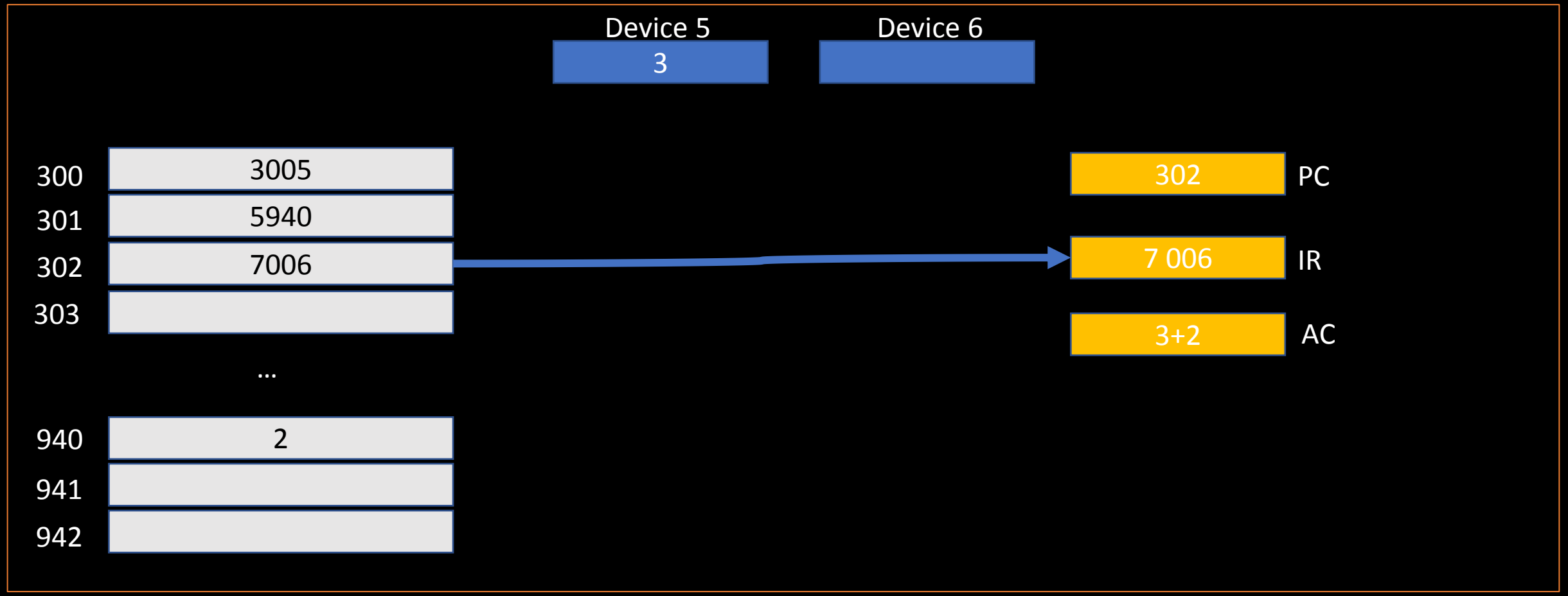
Exercises

Add contents of memory location 940



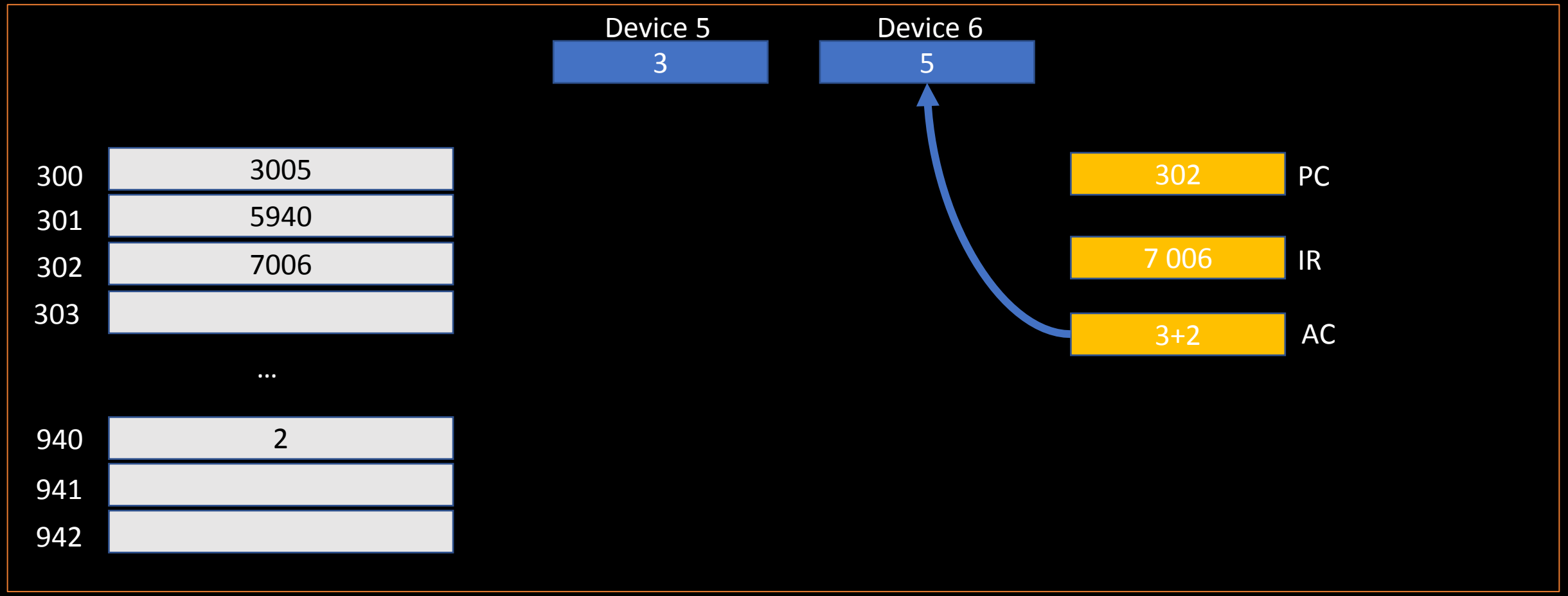
Exercises

Store AC to device 6



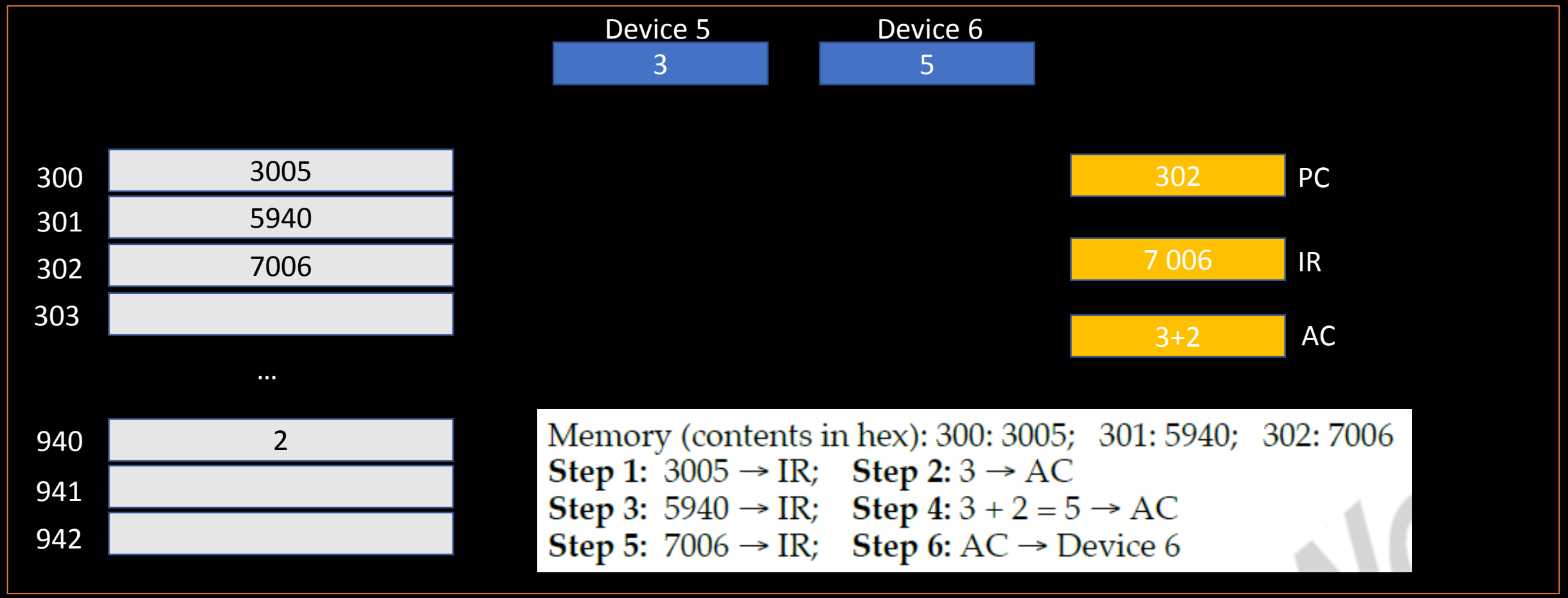
Exercises

Store AC to device 6



Exercises

Store AC to device 6



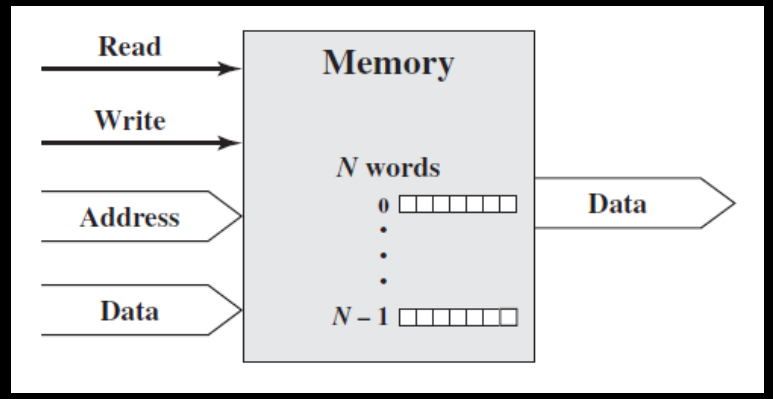
Memory (contents in hex): 300: 3005; 301: 5940; 302: 7006
Step 1: 3005 → IR; **Step 2:** 3 → AC
Step 3: 5940 → IR; **Step 4:** 3 + 2 = 5 → AC
Step 5: 7006 → IR; **Step 6:** AC → Device 6

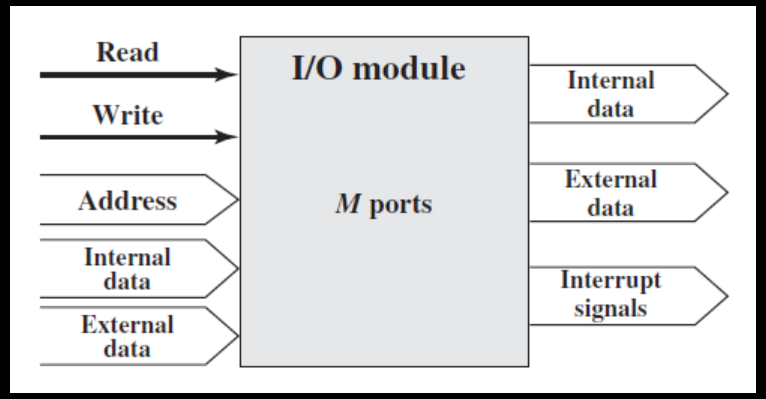
Content

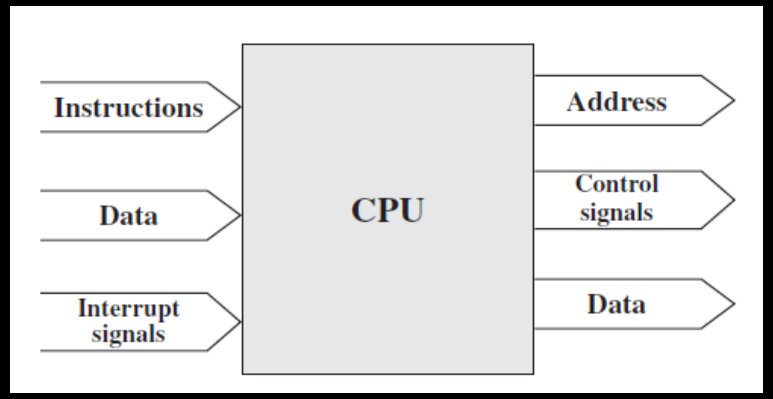
CH 03
Computer Components
Interrupts
Interconnection Structures
Bus Interconnection
Elements of Bus Design: Timing
PCI

Interconnection Structures

- A computer consists of a set of modules of three basic types (processor, memory, I/O) that communicate with each other.
  - The collection of paths connecting the various modules is called the *interconnection structure*.





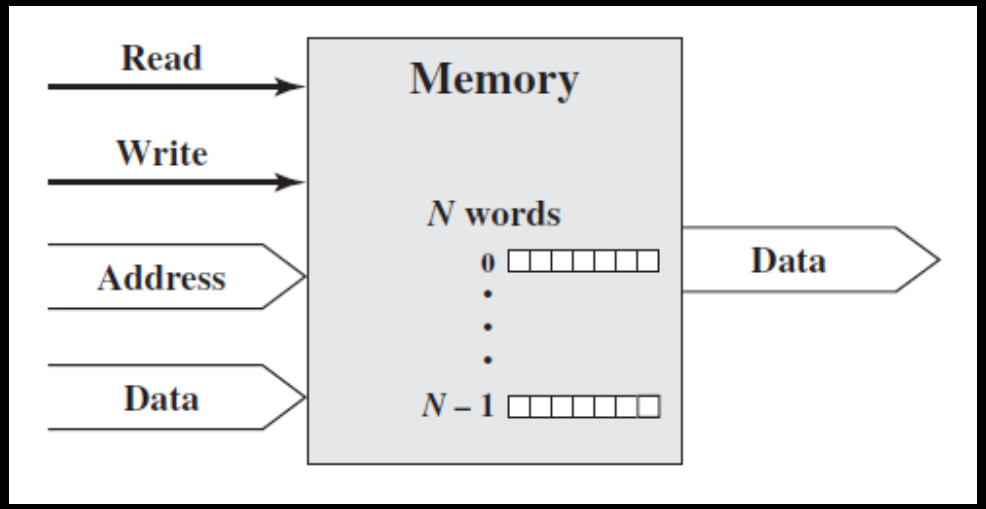


  - The wide arrows represent multiple signal lines carrying multiple bits of information in parallel. Each narrow arrows represents a single signal line.

Interconnection Structures

Memory:

- Memory module will consist of N words of equal length.
- Each word is assigned a unique numerical address (0, 1, . . . ,N – 1).
- A word of data can be read from or written into the memory.
- The nature of the operation is indicated by read and write control signals.

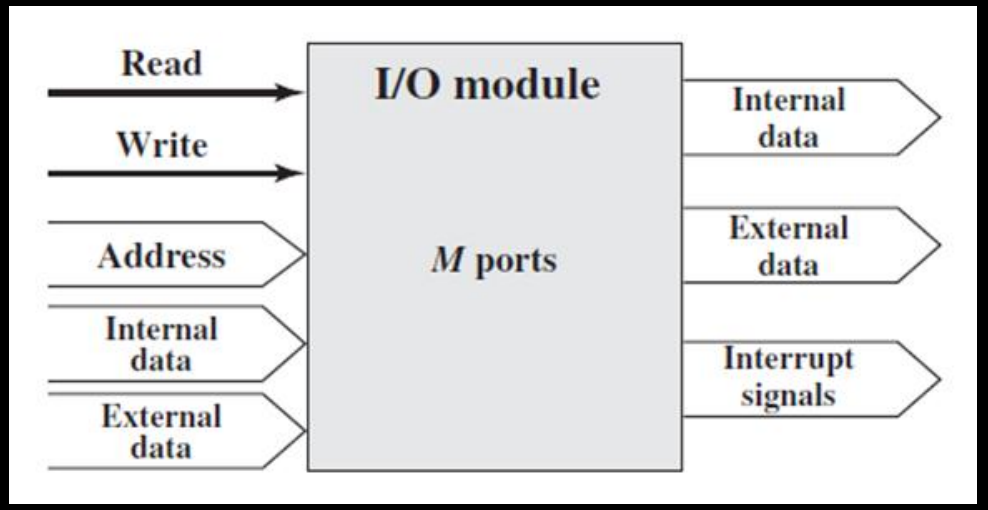


- The location for the operation is specified by an address.

Interconnection Structures

I/O module: computer system considers I/O functionally similar to memory.

- There are two operations, read and write.
- I/O module can send/receive data to/from external device (e.g., Keyboard).
- I/O module can send/receive data to/from internal components (e.g., CPU, memory).
- I/O module may be able to send interrupt signals to the processor.

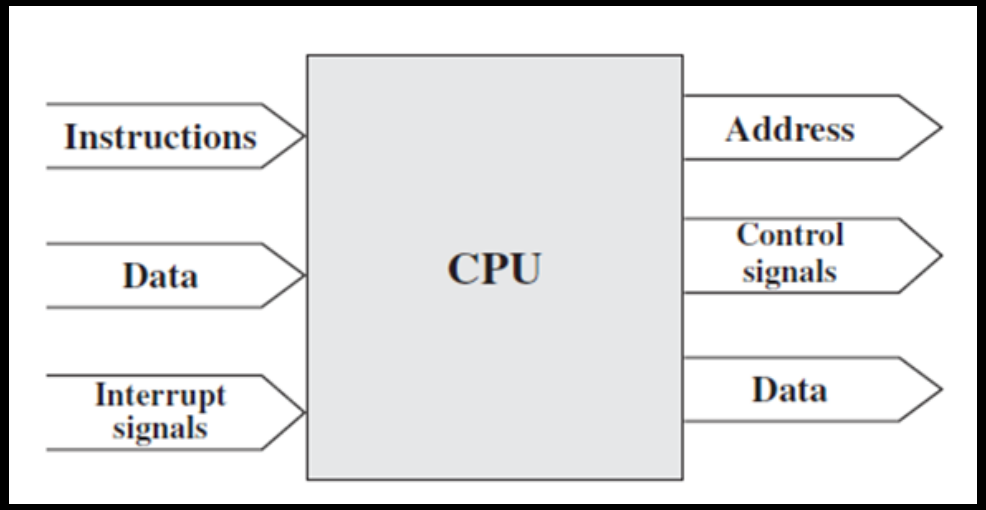


- Address used to identify the external device via *port* number.

Interconnection Structures

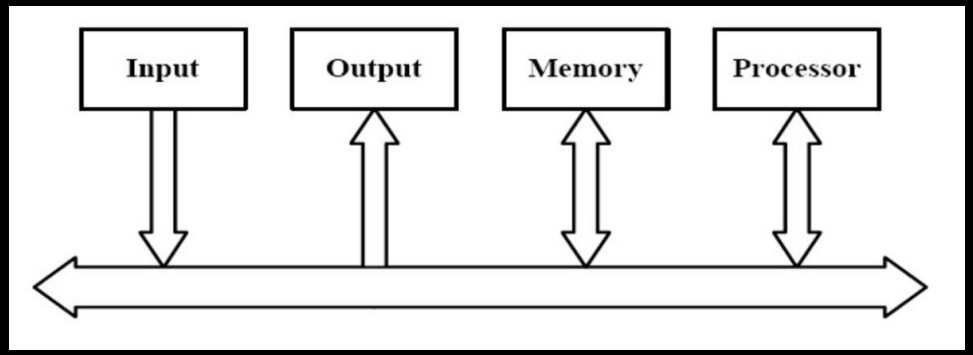
Processor:

- The processor reads in instructions and data, writes out data after processing.
- Uses control signals to control the overall operation of the system.
- It also receives interrupt signals.



Interconnection Structures

- The interconnection structure must support the following types of transfers:
  - Memory to processor
  - Processor to memory
  - I/O to processor
  - Processor to I/O
  - I/O to or from memory
- The most common interconnection structure is the bus and multiple-bus structures.



Content

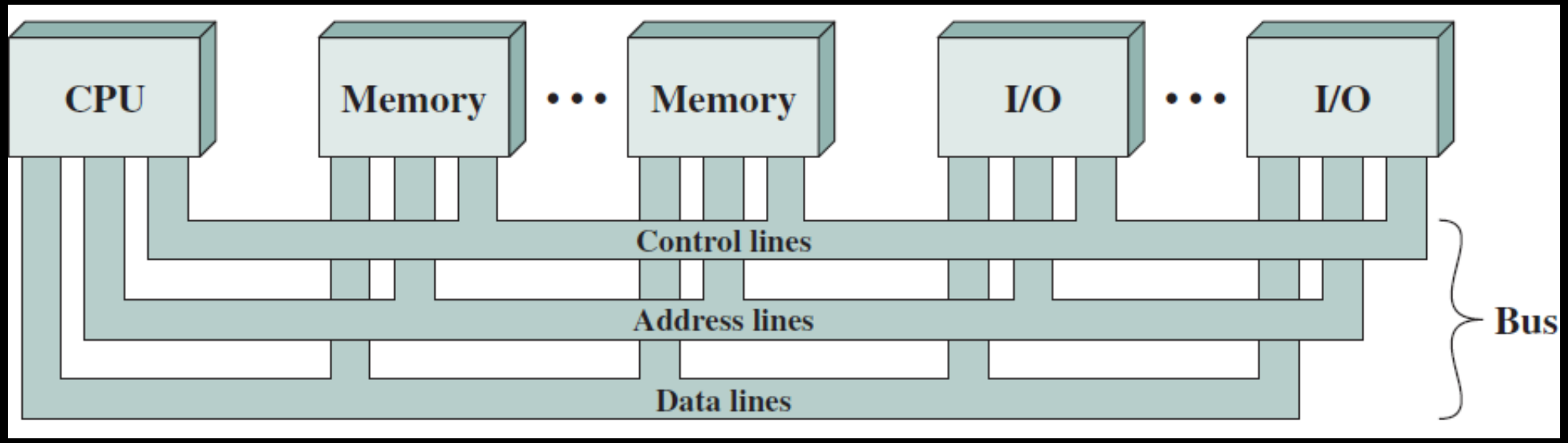
CH 03
Computer Components
Interrupts
Interconnection Structures
Bus Interconnection
Elements of Bus Design: Timing
PCI

Bus Interconnection

- A bus is a communication pathway connecting two or more devices.
  - it is a shared transmission medium.
- Multiple devices connect to the bus, and a signal transmitted by any one device is available for reception by all other devices attached to the bus.
  - If two devices transmit during the same time period, their signals will overlap and become garbled.
  - Thus, only one device at a time can successfully transmit.
- Several lines of a bus can be used to transmit binary digits in parallel.
  - For example, an 8-bit unit of data can be transmitted over eight bus lines.

Bus Interconnection

- The lines of a bus can be classified into three functional groups:
  - data, address, and control lines.
- Each bus is specified by a **width**.
  - **Width** is the number of lines that carry the 1 bit at a time.



Bus Interconnection

Data Bus

- The data bus provides a path for moving data (instructions and operands) among system modules.
- The width of the data bus is a key factor in determining overall system performance.
  - For example, if the data bus is 32 bits wide and each instruction is 64 bits long, then the processor must access the memory module twice during each instruction cycle.

Bus Interconnection

Address Bus

- The address bus is used to designate the source/destination of the data.
  - The address lines are also used to address I/O ports.
- The width of the address bus determines the maximum possible memory capacity of the system.
- The higher-order bits are used to select a particular module on the bus, and the lower-order bits select a memory location or I/O port within the module.
  - For example, on an 8-bit address bus, address 01111111 and below might reference locations in a memory module (module 0) with 128 words of memory, and address 10000000 and above refer to devices attached to an I/O module (module 1).

Bus Interconnection

Control Bus

- The control lines are used to control the access to and the use of the data and address lines.
  - Because the data and address lines are shared by all components, there must be a means of controlling their use.
- Control signals transmit both command and timing information among system modules.
  - Timing signals indicate the validity of data and address information.
  - Command signals specify operations to be performed.

Exercises

3.3 Consider a hypothetical 32-bit microprocessor having 32-bit instructions composed of two fields: the first byte contains the opcode and the remainder the immediate operand or an operand address.

a. What is the maximum directly addressable memory capacity (in bytes)?

b. Discuss the impact on the system speed if the microprocessor bus has

1. a 32-bit local address bus and a 16-bit local data bus, or
2. a 16-bit local address bus and a 16-bit local data bus.

c. How many bits are needed for the program counter and the instruction register?

Exercises

a. $2^{32-8} = 2^{24}$ = 16,777,216 bit / 8 = 2097152 bytes = 2 Mb

b. (1) if we have a 32-bit address bus, the processor can transfer the whole address at once, because the operand address of the instruction occupies 24 bits. However, because the data bus is only 16-bits, it will require 2 cycles to fetch a 32-bit instruction.
(2) for a 16-bit address bus, the processor cannot address the whole memory because the address occupies 24 bits. Thus, the processor will need 2 cycles to fetch an address. Also, because the data bus is 16 bits, the processor will need 2 cycles to fetch an instruction/operand.

c. The program counter needs at least 24 bits, because the memory has 2^{24} words. If the instruction register is to contain the whole instruction, it will have to be 32-bits; if the register will contain only the opcode, it will have to be 8 bits.

Content

CH 03
Computer Components
Interrupts
Interconnection Structures
Bus Interconnection
Elements of Bus Design: Timing
PCI

Elements of Bus Design: Timing

- Timing refers to the way in which events are coordinated on the bus.
  - Buses use either **synchronous** timing or **asynchronous** timing.

Elements of Bus Design: Timing

- Timing refers to the way in which events are coordinated on the bus.
  - Buses use either **synchronous** timing or **asynchronous** timing.
- In synchronous timing, the occurrence of events is determined by a clock.

Elements of Bus Design: Timing

- Timing refers to the way in which events are coordinated on the bus.
  - Buses use either **synchronous** timing or **asynchronous** timing.
- In synchronous timing, the occurrence of events is determined by a clock.
- In asynchronous timing, the occurrence of one event on a bus follows and depends on the occurrence of a previous event.

Elements of Bus Design: Timing

- Timing refers to the way in which events are coordinated on the bus.
  - Buses use either **synchronous** timing or **asynchronous** timing.
- In synchronous timing, the occurrence of events is determined by a clock.
- In asynchronous timing, the occurrence of one event on a bus follows and depends on the occurrence of a previous event.
- The bus includes a clock line upon which a clock transmits a regular sequence of alternating 1s and 0s of equal duration.
  - A single 1–0 transmission is referred to as a clock or bus cycle.

Exercises

3.5 Consider a 32-bit microprocessor, with a 16-bit external data bus, driven by an 8-MHz input clock. Assume that this microprocessor has a bus cycle whose minimum duration equals four input clock cycles. What is the maximum data transfer rate across the bus that this microprocessor can sustain, in bytes/s? To increase its performance, would it be better to make its external data bus 32 bits or to double the external clock frequency supplied to the microprocessor? State any other assumptions you make and explain. Hint: Determine the number of bytes that can be transferred per bus cycle.

Exercises

We have:

- Processor is 32 bit.
- Data bus width = 16 bits.
- Clock speed = 8 MHz (8,000,000 cycles per second).
- 1 bus cycle = 4 clock cycles.

Output required:

- Maximum data transfer rate of the processor’s bus.
- To increase its performance, …

Exercises

So, for the processor to transfer 1-bit data, it requires 4 clock cycles. The clock performs 8 Million cycles per second.

Then, maximum bus cycle rate = 8 M / 4 = 2 M/sec.

Data transfer rate per second = bus cycle rate * bus width =

$$2 * 16 = 32\ bit/\text{sec} = 4\ byte/second$$

Exercises

To increase its performance:

- By doubling the frequency, it may mean adopting a new chip manufacturing technology (assuming each instruction will have the same number of clock cycles);

  Therefore, the speed of the memory chips will need to double, not to slow down the microprocessor.

- By doubling the external data bus, that means wider on-chip data bus drivers and modifications to the bus control logic.

  Therefore, the word length of the memory will must double to be able to send/receive 32-bit quantities.

Exercises

3.7 Consider two microprocessors having 8- and 16-bit-wide external data buses, respectively. The two processors are identical otherwise and their bus cycles take just as long.

a. Suppose all instructions and operands are two bytes long. By what factor do the maximum data transfer rates differ?

b. Repeat assuming that half of the operands and instructions are one byte long.

Exercises

a. Through a single bus cycle, the 8-bit microprocessor transfers one byte while the 16-bit microprocessor transfers two bytes. The 16-bit microprocessor has twice the data transfer rate.

b. By assuming that we have to perform 100 transfers of operands and instructions which 50 are one byte long and 50 are two bytes long. Suppose that one transfer operation takes 1 cycle. The 8-bit microprocessor needs 50 + (2 x50) = 150 bus cycles for the transfer. The 16-bit microprocessor needs 50 + 50 = 100 bus cycles. Therefore, the data transfer rates differ by a factor of 150/100 = 1.5.

Exercises

3.9 The VAX SBI bus uses a distributed, synchronous arbitration scheme. Each SBI device (i.e., processor, memory, I/O module) has a unique priority and is assigned a unique transfer request (TR) line. The SBI has 16 such lines (TR0,TR1, . . ., TR15), with TR0 having the highest priority. When a device wants to use the bus, it places a reservation for a future time slot by asserting its TR line during the current time slot. At the end of the current time slot, each device with a pending reservation examines the TR lines; the highest-priority device with a reservation uses the next time slot.

A maximum of 17 devices can be attached to the bus. The device with priority

16 has no TR line. Why not?

Exercises

The lowest-priority device has priority 16. That device must wait for all other devices. Therefore, it may transmit in any slot not reserved by the other SBI devices. Thus, it does not need TR line.

Exercises

3.11 For a synchronous read operation (Figure 3.19), the memory module must place the data on the bus sufficiently ahead of the falling edge of the Read signal to allow for signal settling. Assume a microprocessor bus is clocked at 10 MHz and that the Read signal begins to fall in the middle of the second half of T3.

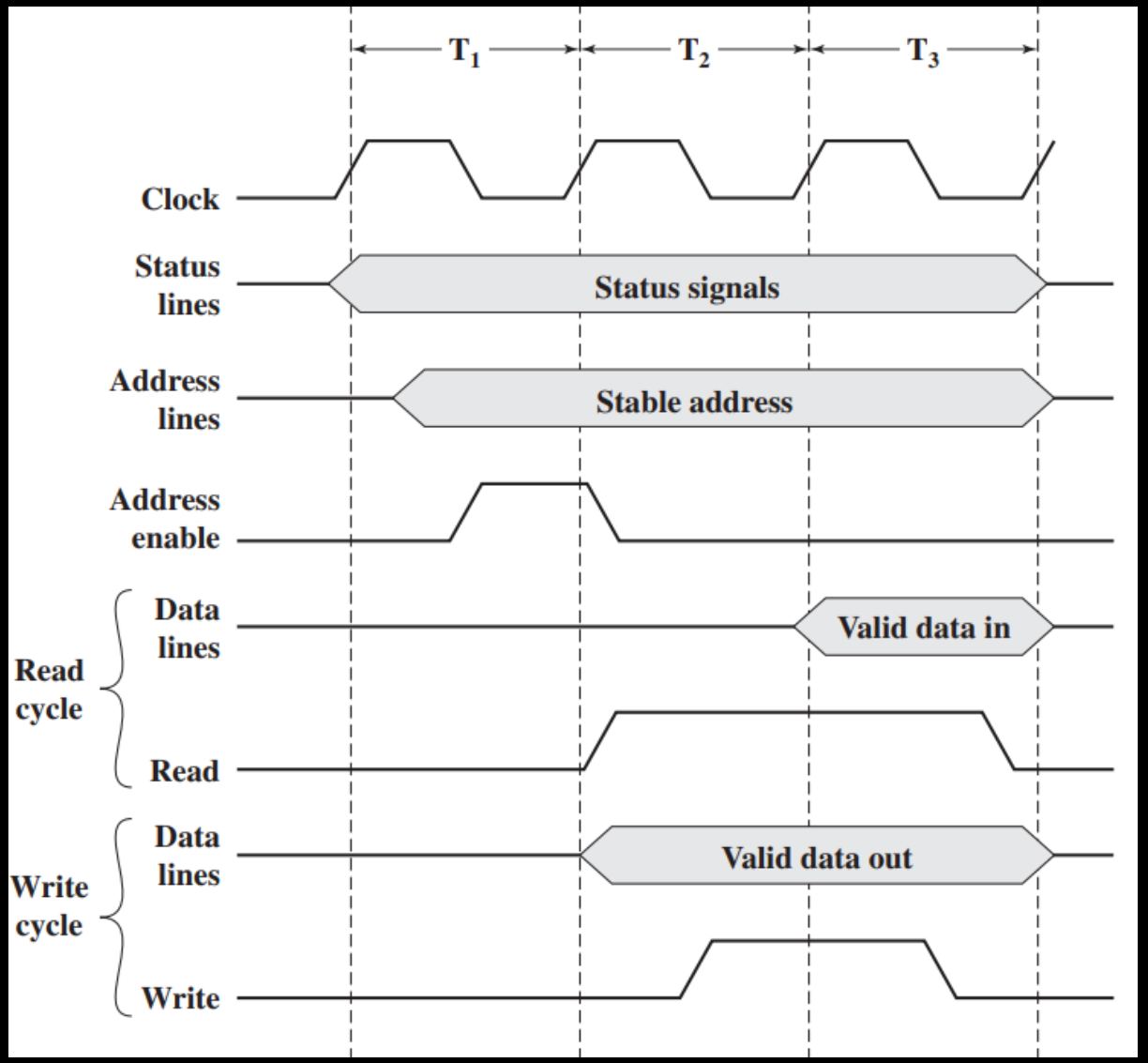
a. Determine the length of the memory read instruction cycle.

b. When, at the latest, should memory data be placed on the bus? Allow 20 ns for the settling of data lines.

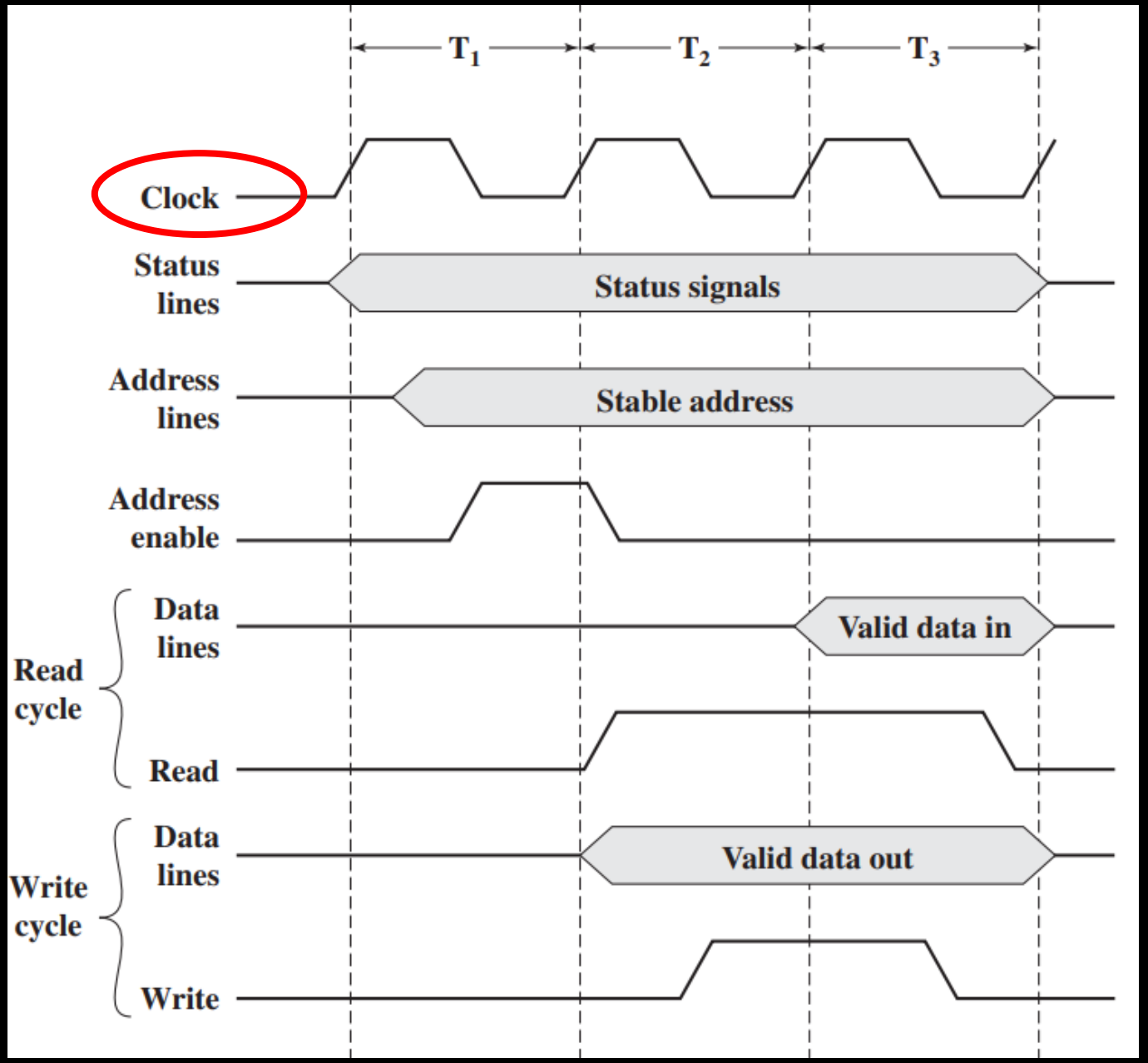
Exercises

Figure 3.19 shows a timing diagram for synchronous read and write operations

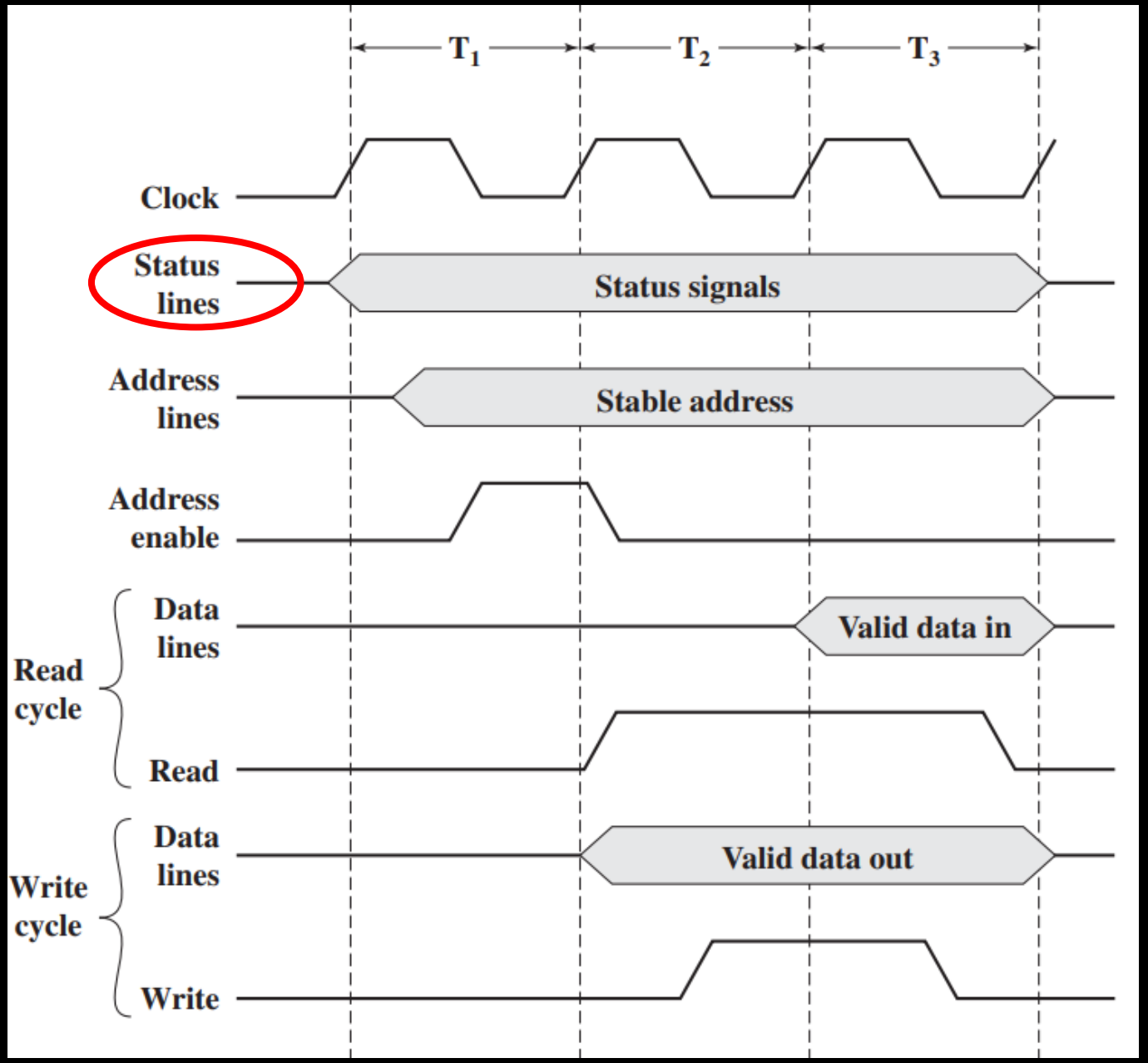
Before answering,
let's analyse the figure



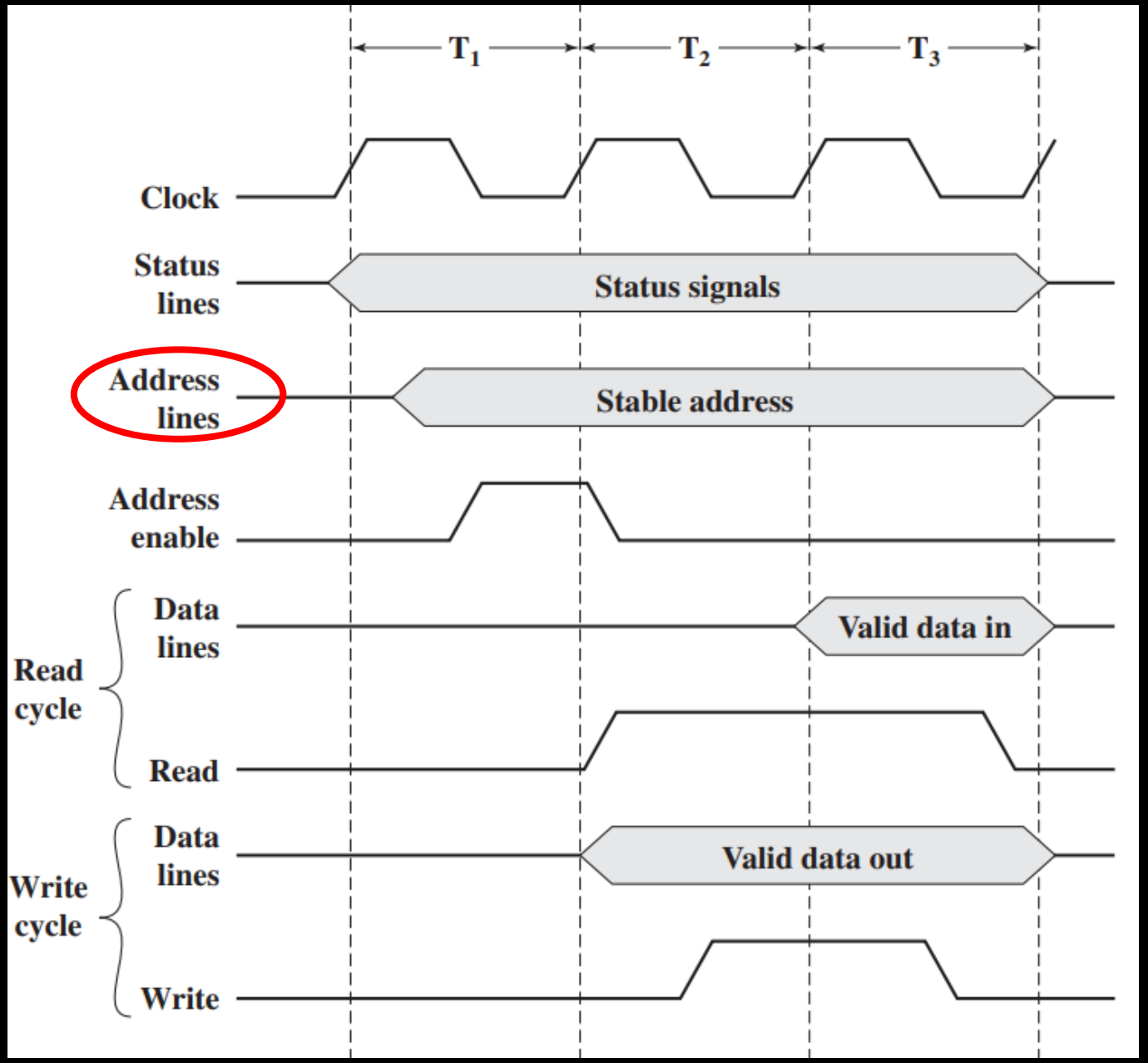
Shows the cycles of the clock. In this case, we have T1, T2, and T3



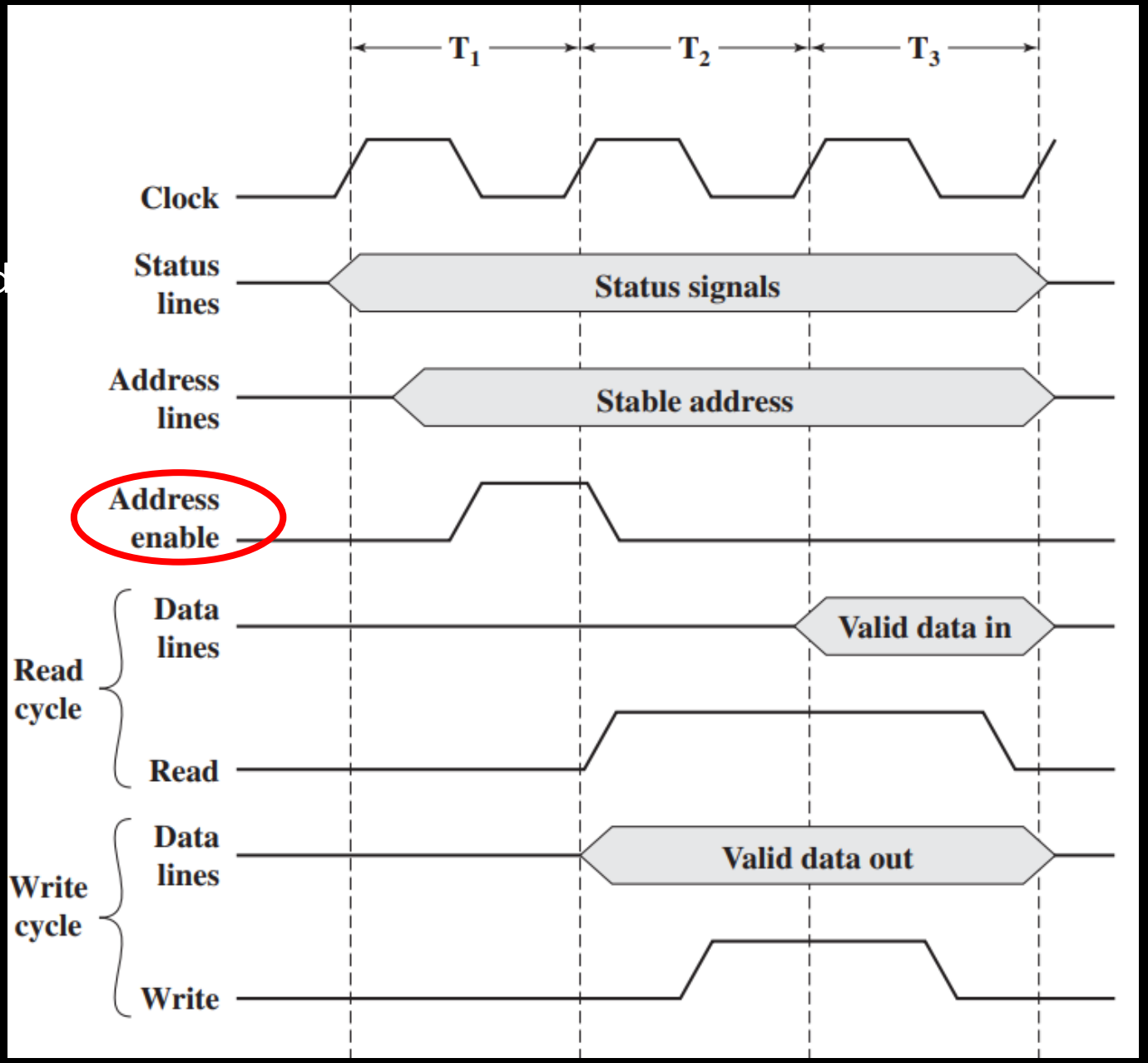
Multiple lines indicates the status of the operations



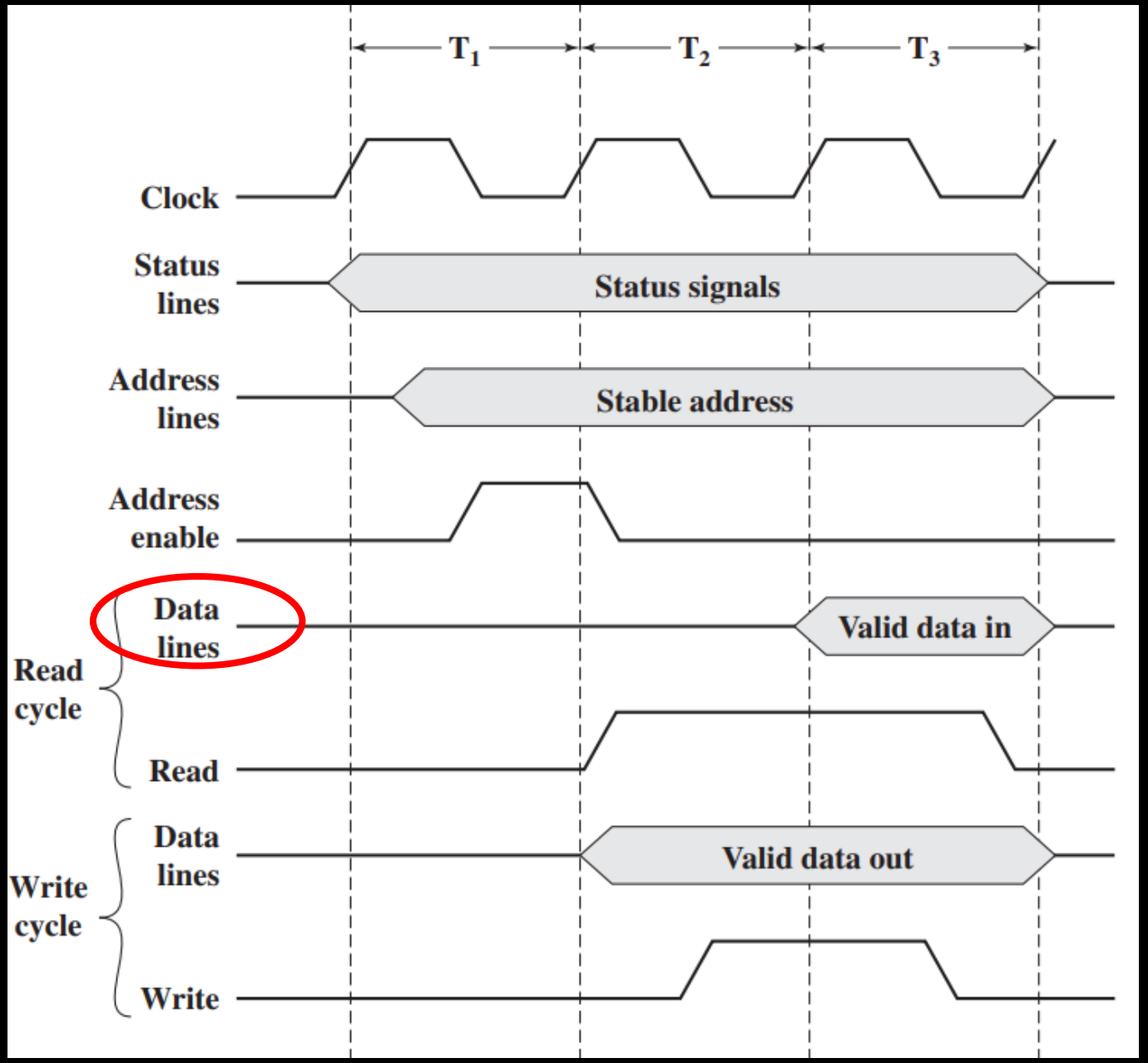
Multiple lines define the memory address for read/write



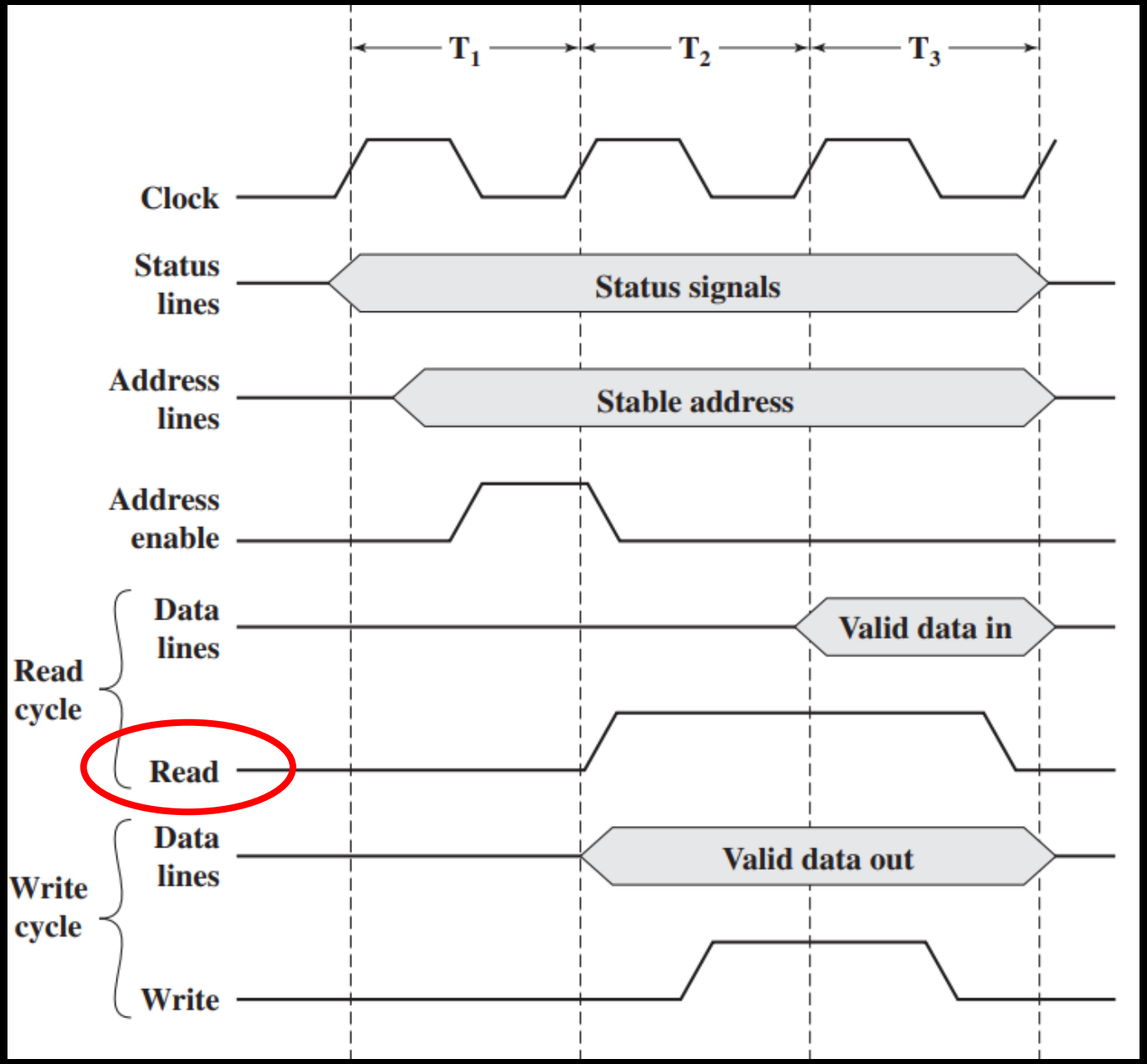
Indicates that there is address to be transferred through the address lines



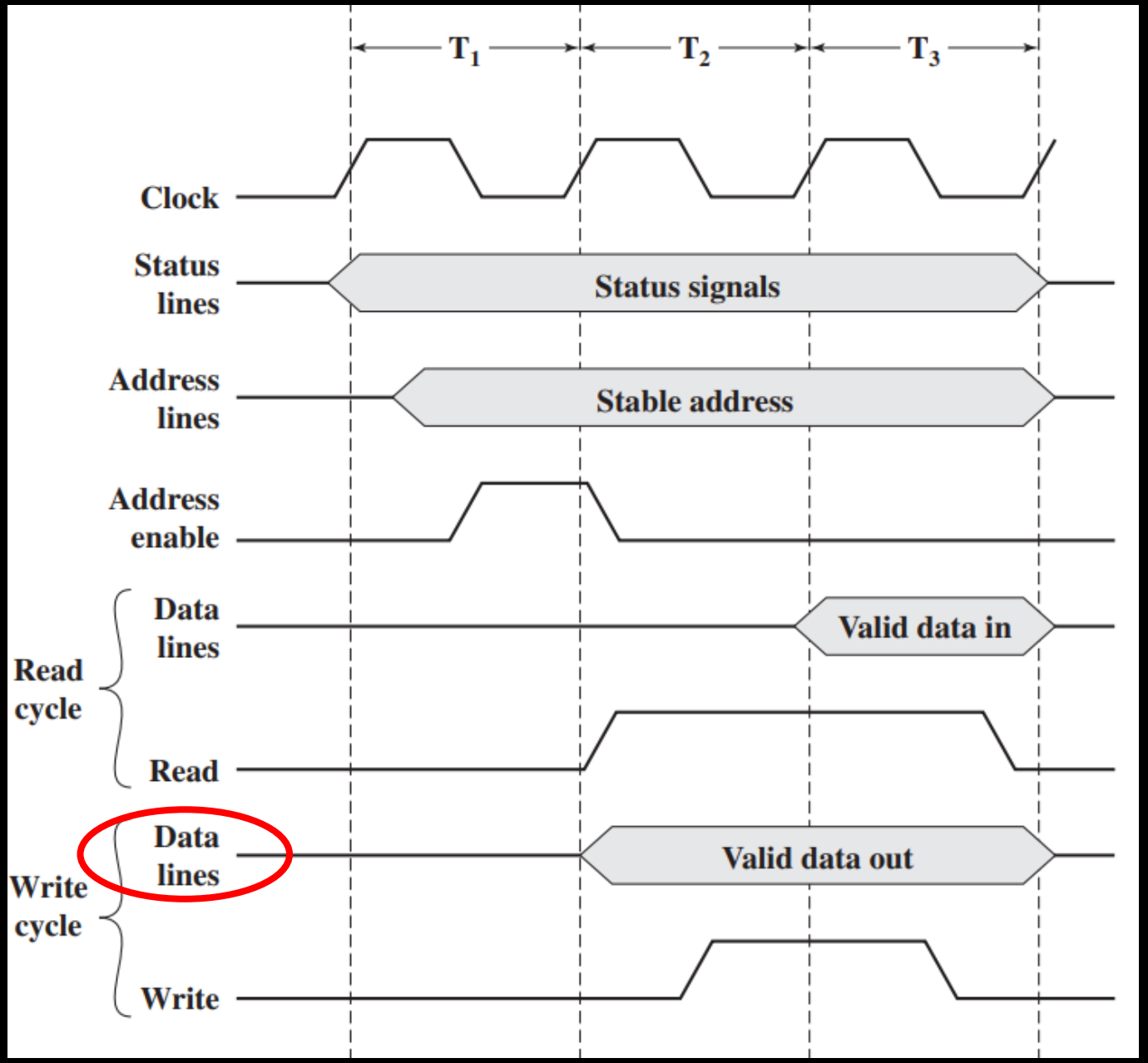
Data to be read



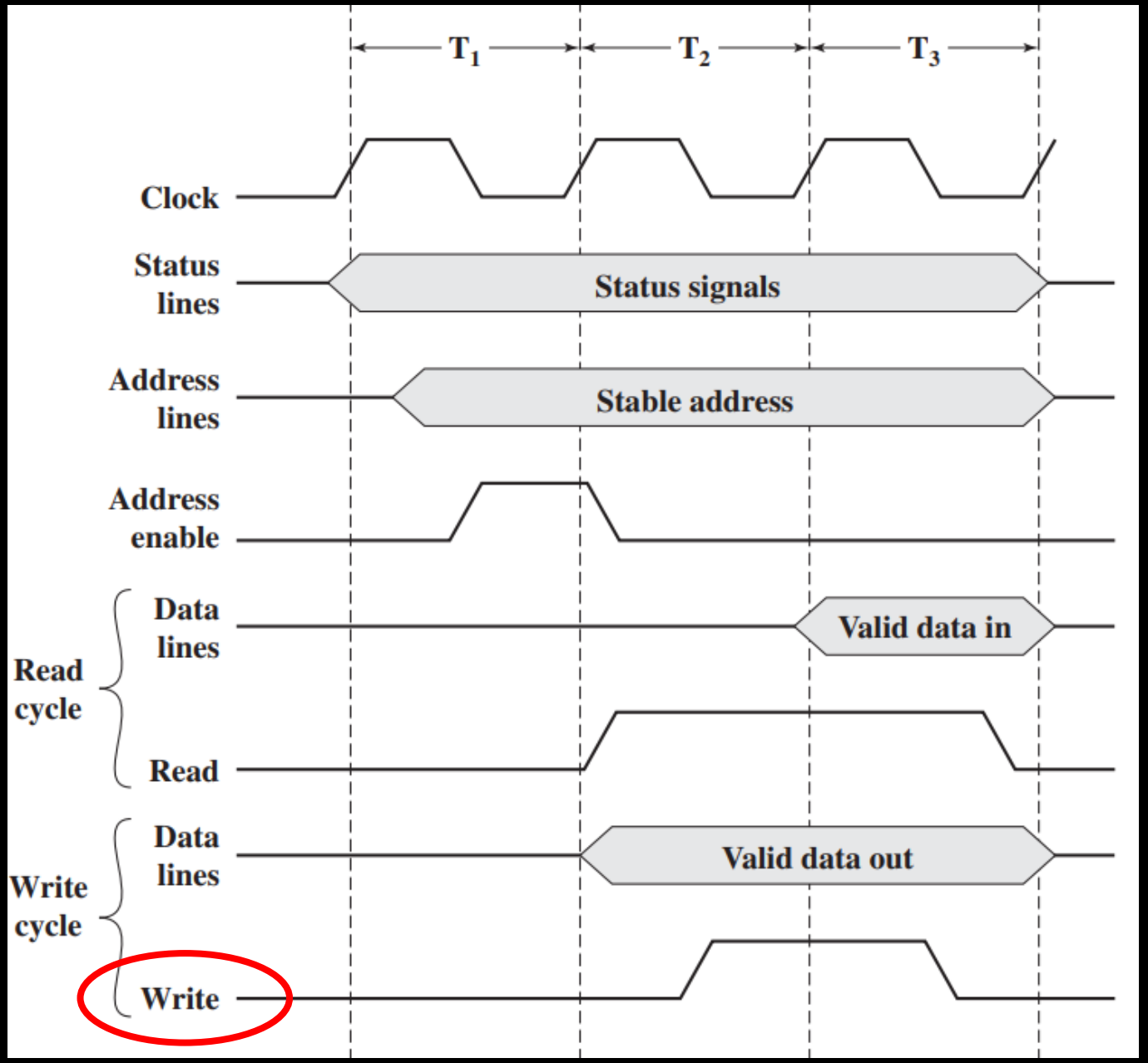
Read operation



Data to be written

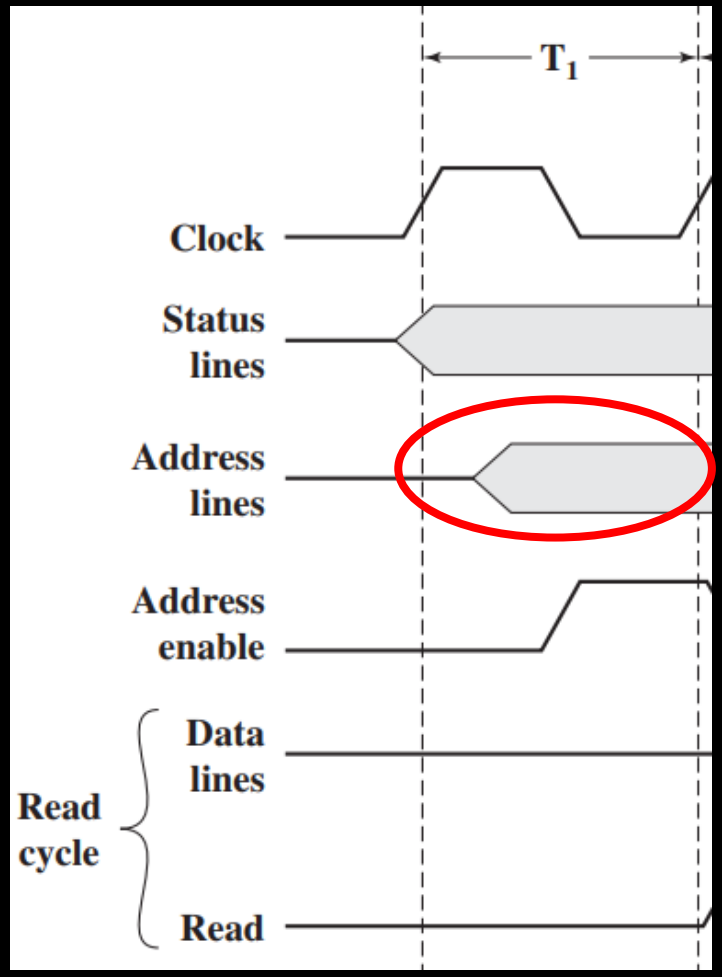


Write operation



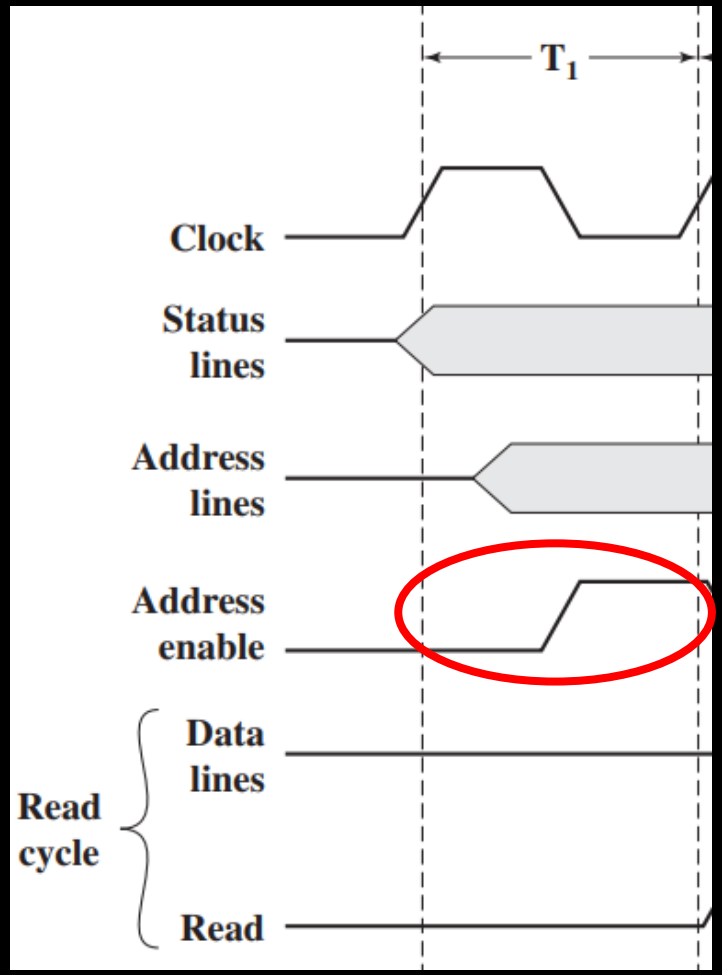
During first clock cycle:

1. The processor places a memory address on the address lines

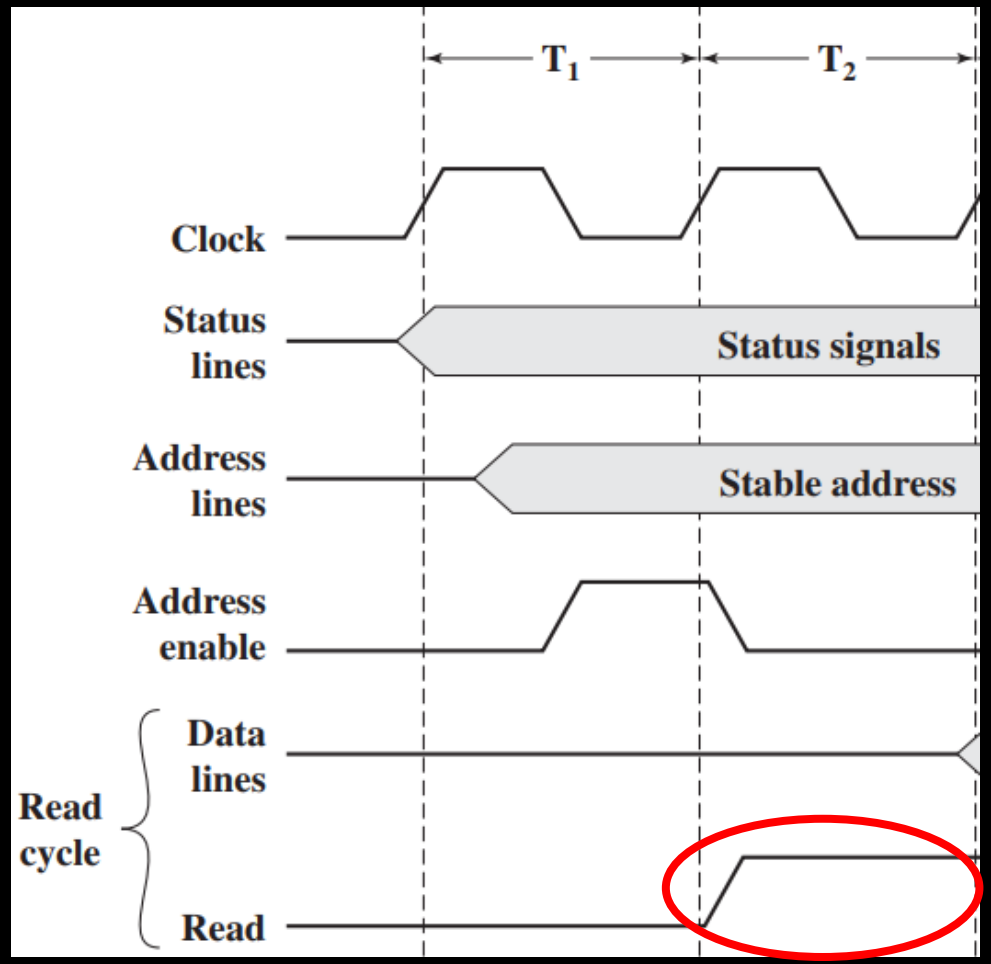


During first clock cycle:

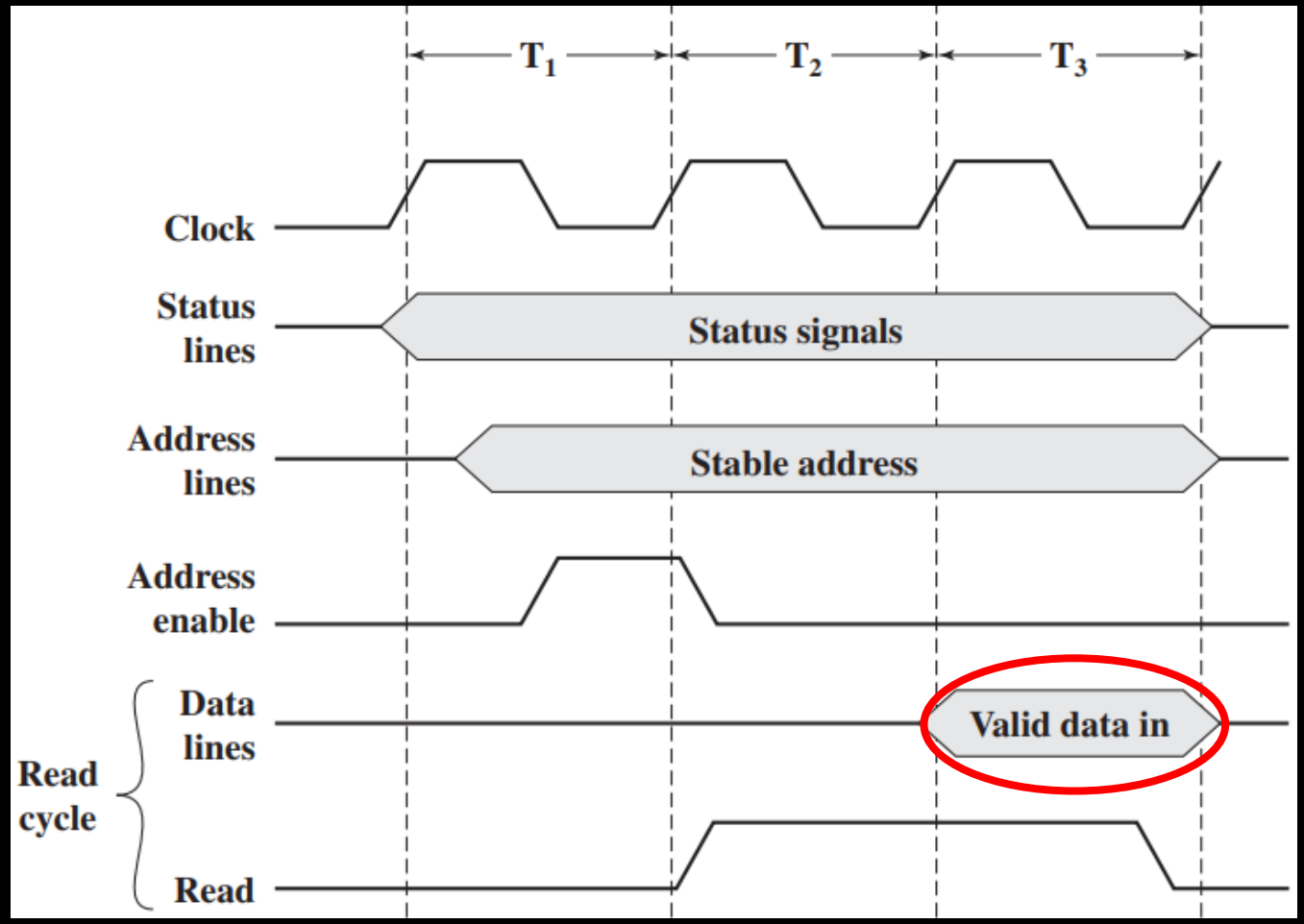
1. The processor places a memory address on the address lines
2. Once the address lines have stabilized, the processor issues an address enable signal.



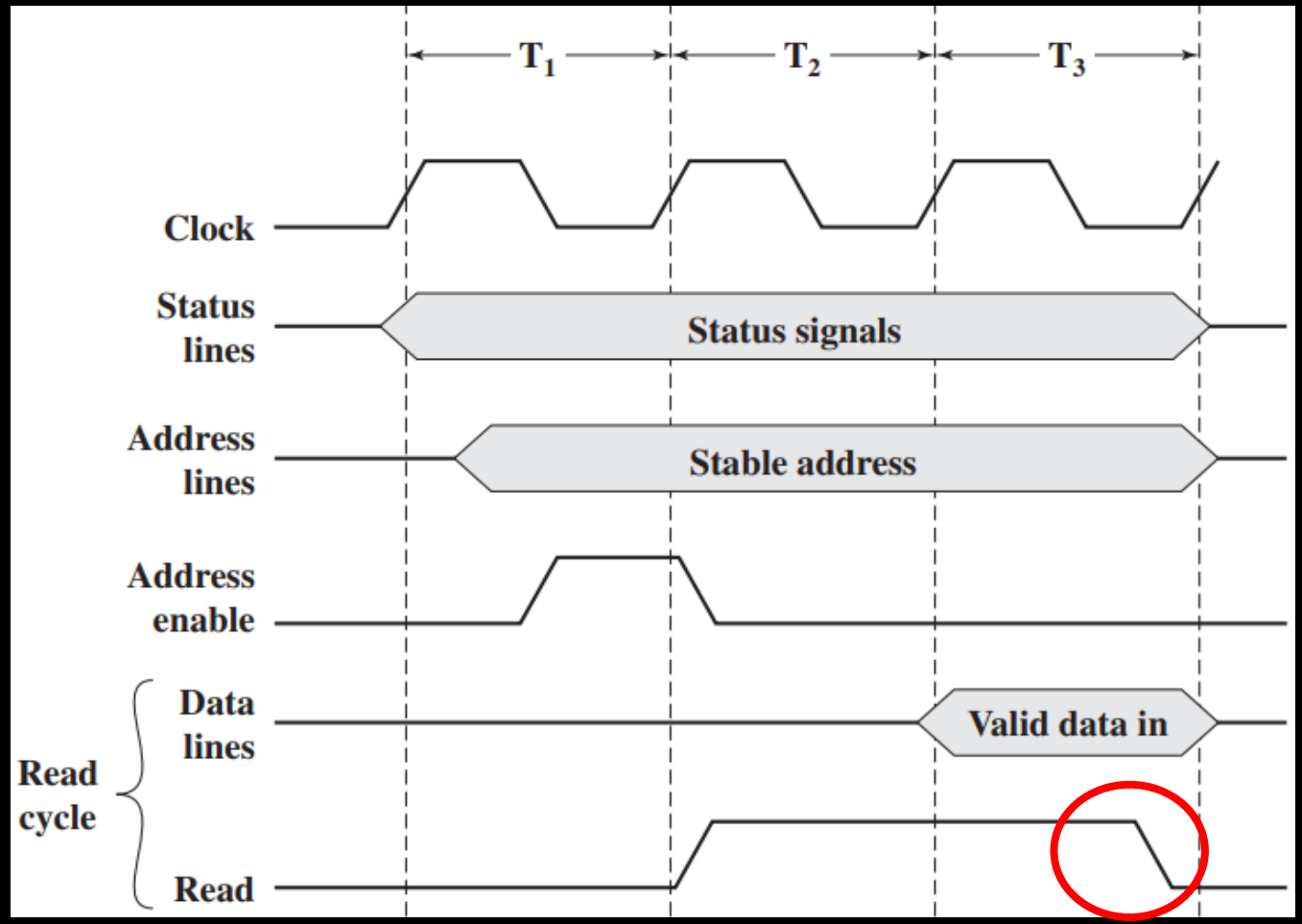
For a read operation, the processor issues a read command at the start of the second cycle.



A memory module recognizes the address and, after a delay of one cycle, places the data on the data lines.



The processor reads the data from the data lines and drops the read signal.



Exercises

We have

- Clock speed 10 Mhz
- 20 ns for the settling of data lines.

Output required:

- Length of the cycle required for memory read instruction.
- Time to place memory data on the bus.

Exercises

a. Since the clocking frequency is 10 MHz, the clock period is $\frac{1}{10{,}000{,}000} = 10^{-7}\ s = 100\ ns$. Therefore, the length of the memory read instruction cycle equals to $100 * 3 = 300\ ns$ since the Read signal begins to fall in the middle of the second half of T3.

b. Since the Read signal begins to fall at 75 ns from the beginning of T3 (in the middle of the second half of T3). Therefore, the memory have to copy or put the data on the bus before $75 - 20 = 55$ ns from the beginning of T3.

Exercises

3.14 A microprocessor has an increment memory direct instruction, which adds 1 to the value in a memory location. The instruction has five stages: fetch opcode (four bus clock cycles), fetch operand address (three cycles), fetch operand (three cycles), add 1 to operand (three cycles), and store operand (three cycles).

a. By what amount (in percent) will the duration of the instruction increase if we have to insert two bus wait states in each memory read and memory write operation?

b. Repeat assuming that the increment operation takes 13 cycles instead of 3 cycles.

Exercises

We have

- Fetch opcode → 4 clock cycles
- Fetch operand address → 3 clock cycles
- Fetch operand → 3 clock cycles
- Add 1 to operand → 3 cycles
- Store operand → 3 cycles
- Number of wait states = 2

Output required:

- The increasing amount of duration to execute the instruction.

Exercises

a) Before inserting wait bus, the instruction requires 4+3+3+3+3 = 16 clock cycles.

We have 4 operations require memory read/write: Fetch opcode, Fetch operand address, Fetch operand, Store operand. Each memory operation 2 wait states (cycles). Thus, the instruction requires
(2 wait states * 4 operations) + 16 = 24 clock cycle, which is increased to 8/16 = 50%

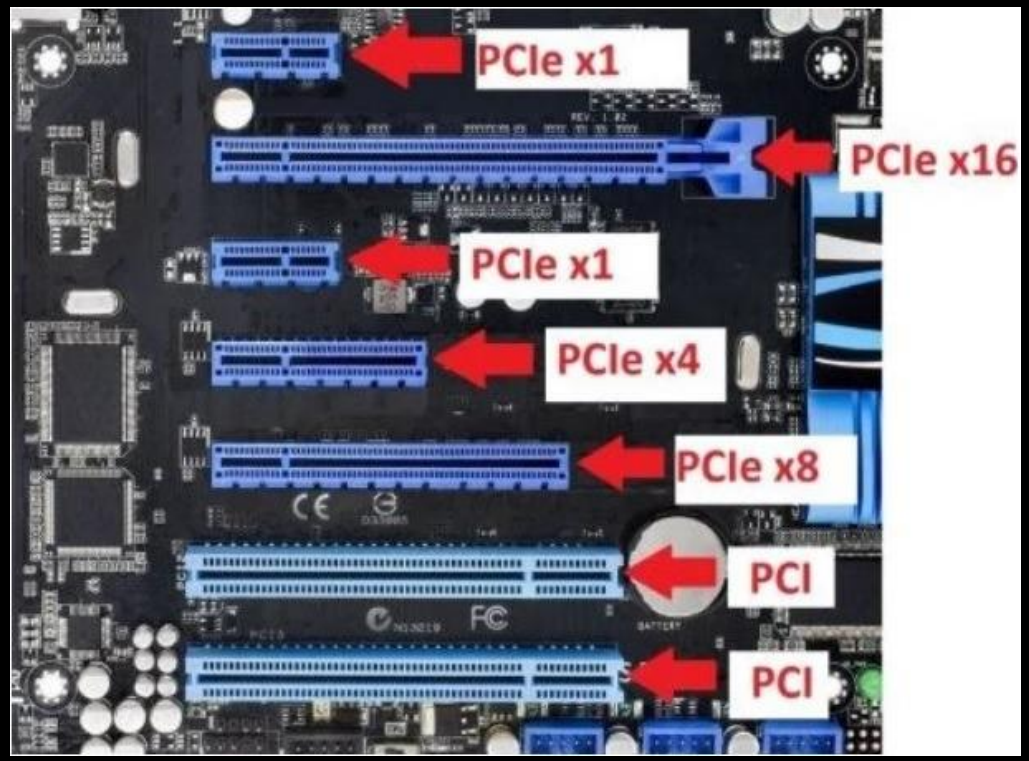
b) By assuming that (13 cycles to add 1 to operand) instead of (3 cycles to add 1 to operand), when there is no wait states, the instruction needs 4+3+3+13+3 = 26 cycles. When the case has wait states, the instruction needs 26 + 8 = 34 cycles which is increased to 8/26 = 31%.

Content

CH 03
Computer Components
Interrupts
Interconnection Structures
Bus Interconnection
Elements of Bus Design: Timing
PCI

PCI

- The peripheral component interconnect (PCI) is a bus that can deliver better system performance for high-speed I/O subsystems
  - graphic display adapters, network interface controllers, disk controllers, and so on



TASK

CH03
10
17