Computer Organization and Architecture

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

Content

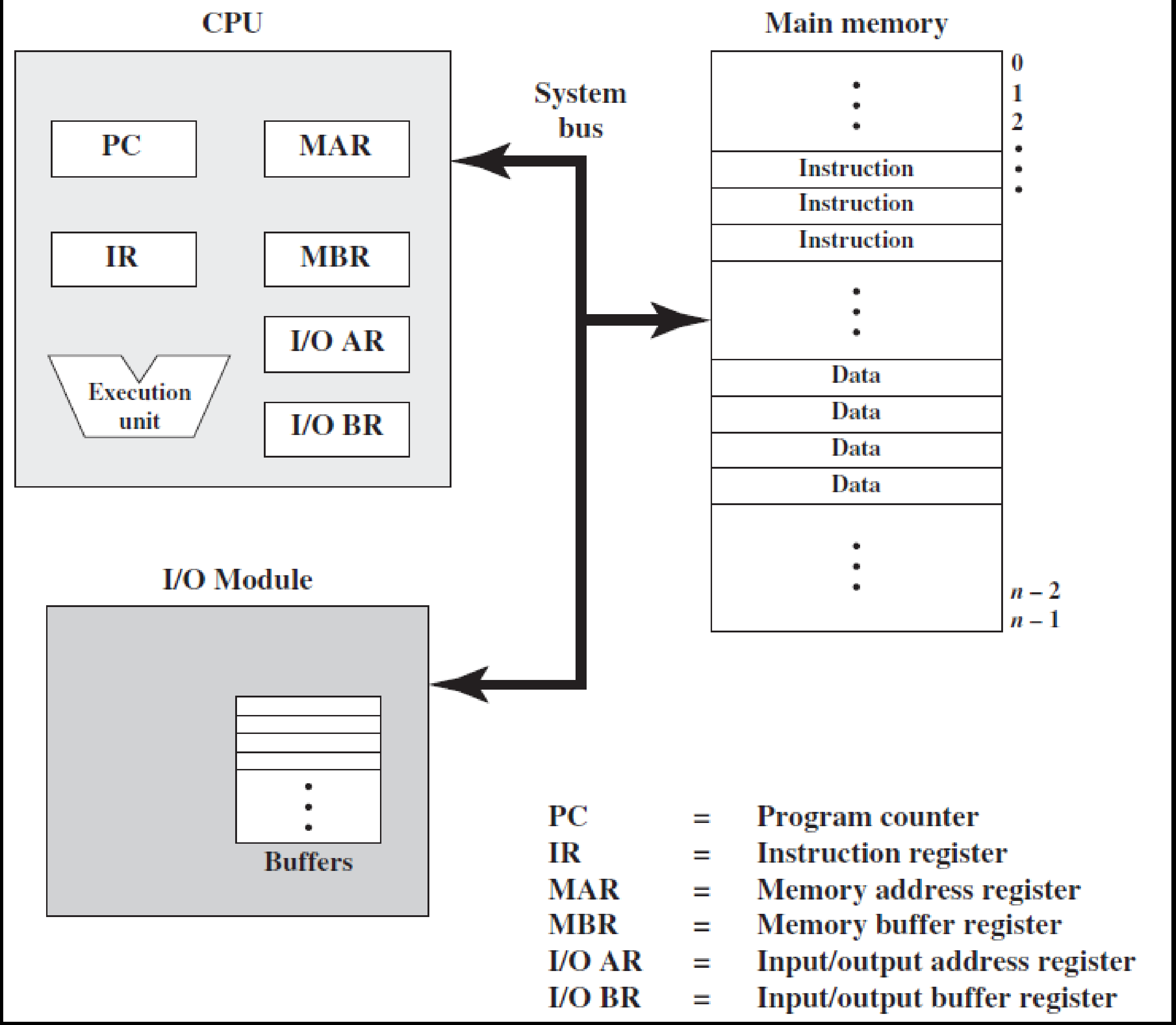
	CH 03
➡	Computer Components
	Interrupts
	Interconnection Structures
	Bus Interconnection
	Elements of Bus Design: Timing
	PCI

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

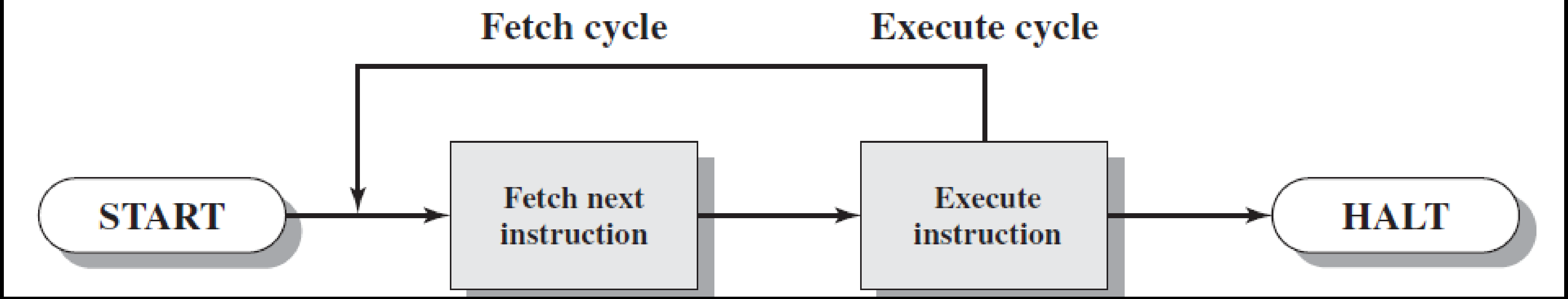
- Top level view of computer components



PC	=	Program counter
IR	=	Instruction register
MAR	=	Memory address register
MBR	=	Memory buffer register
I/O AR	=	Input/output address register
I/O BR	=	Input/output buffer register

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

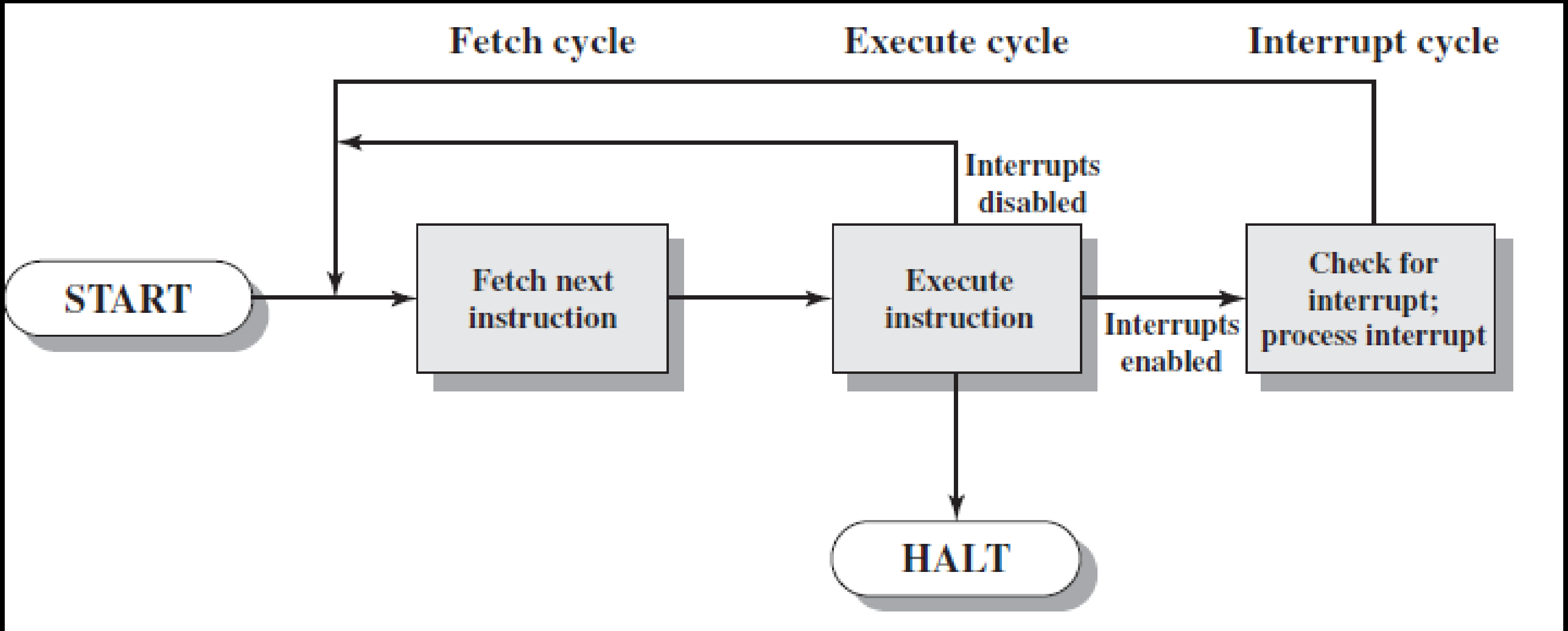
- Consider a computer in which each instruction occupies one 16-bit word of memory.
  1. The program counter is set to location 300. The processor will next fetch the instruction at location 300.
     - On succeeding instruction cycles, it will fetch instructions from locations 301, 302, 303, and so 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

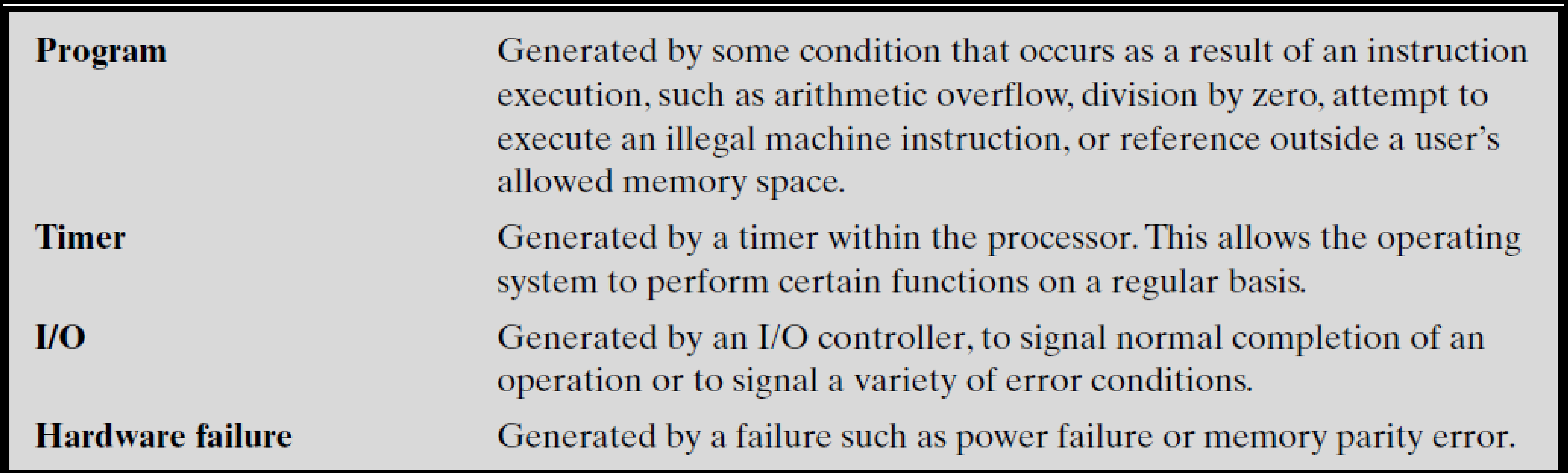
Interrupts

- 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

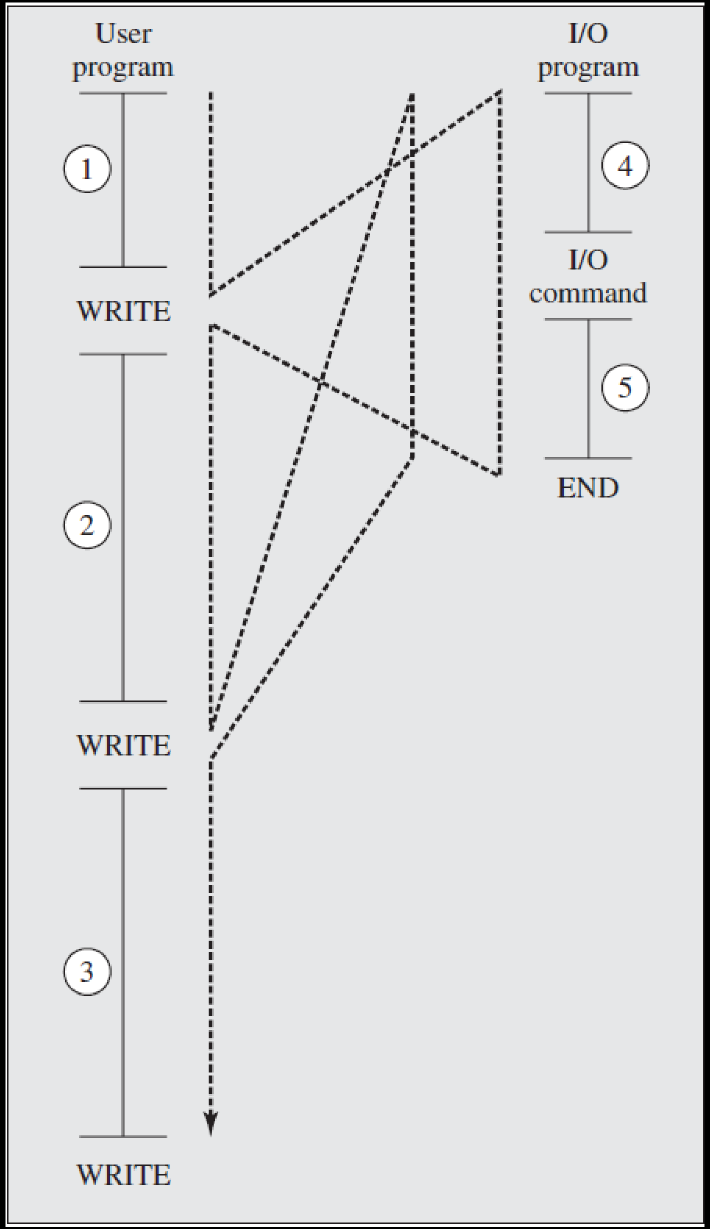


Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, or reference outside a user's allowed memory space.
Timer	Generated by a timer within the processor. This allows the 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

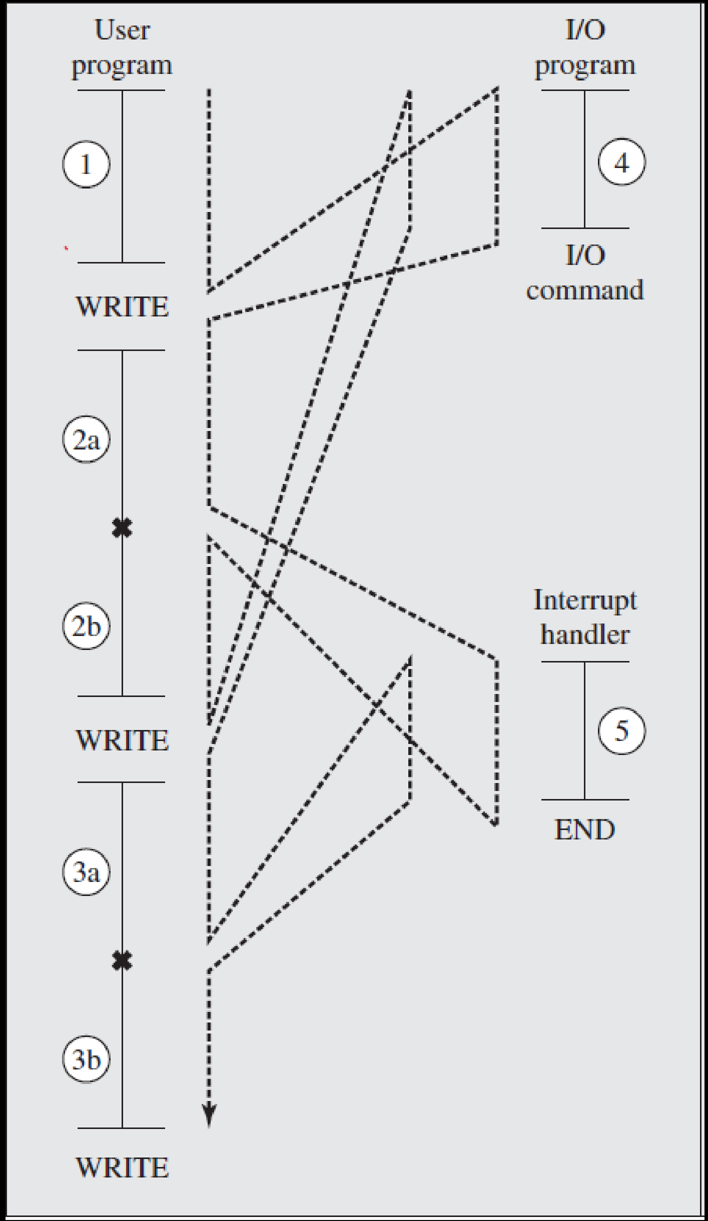
- The user program performs a series of WRITE calls interleaved with processing.
  - Code segments 1, 2, and 3 refer to sequences of instructions that do not involve I/O.
- The WRITE calls are to an I/O program that is a system utility and that will perform the actual I/O operation.
- Once the **I/O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



Interrupts

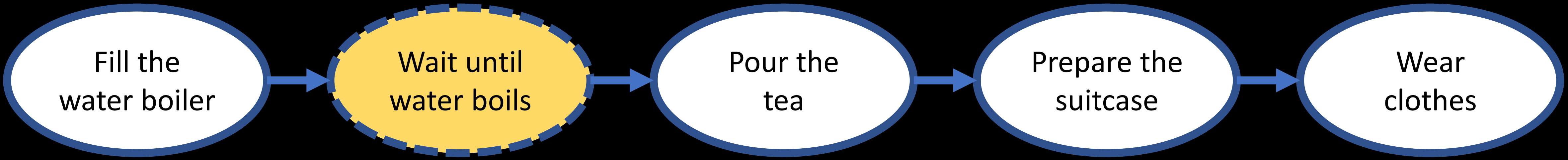
With interrupts

- 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.



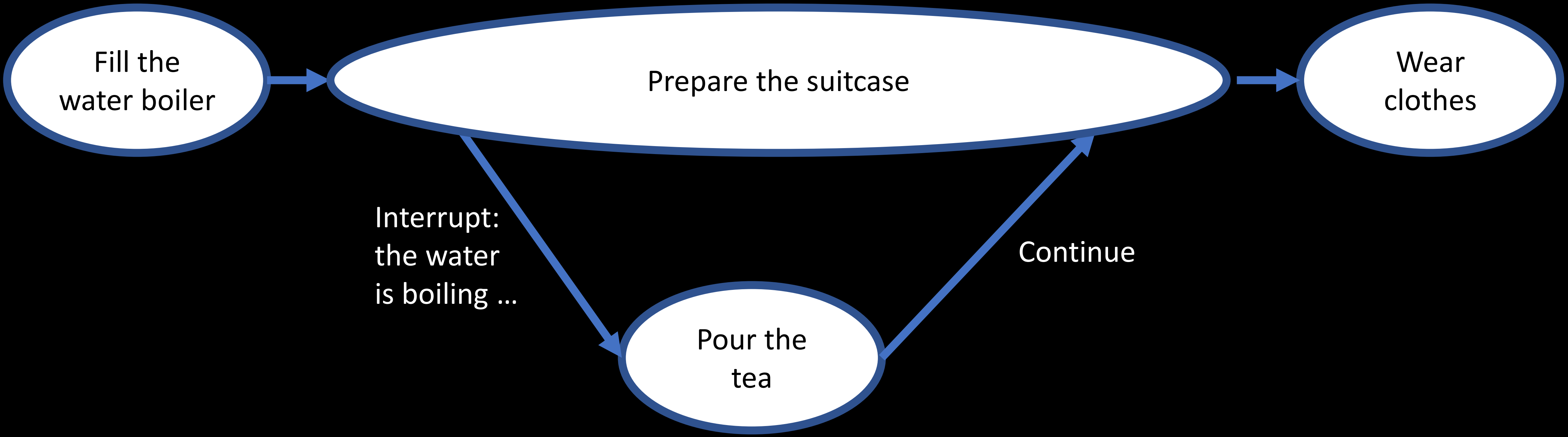
Interrupts

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



Interrupts

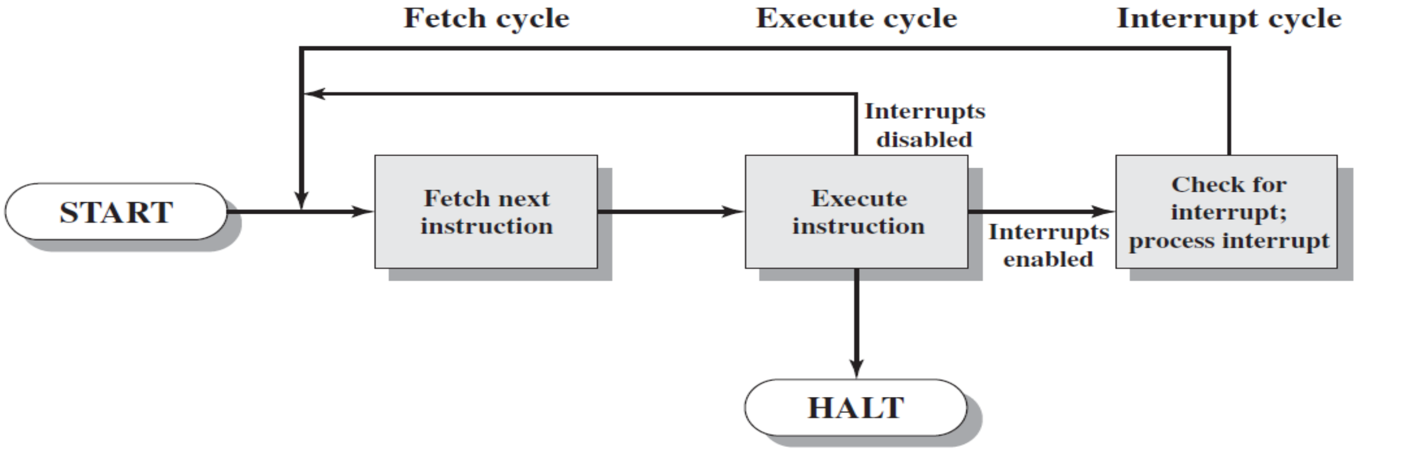
Travelling and prepare tea example. suppose you are preparing your suitcase and want to prepare a cup of tea (with interrupts).



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

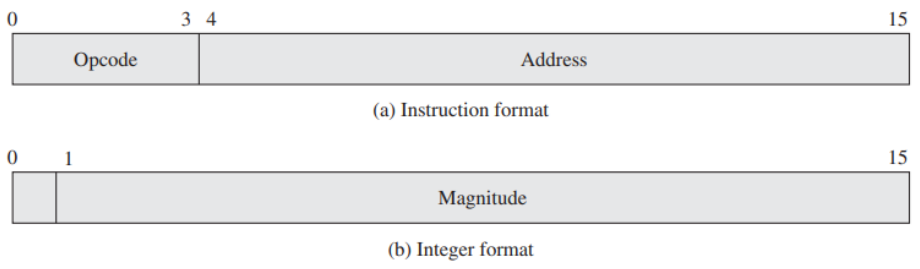
3.1 The hypothetical machine of Figure 3.4 also has two I/O instructions:

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

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

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.



(a) Instruction format

(b) Integer format

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

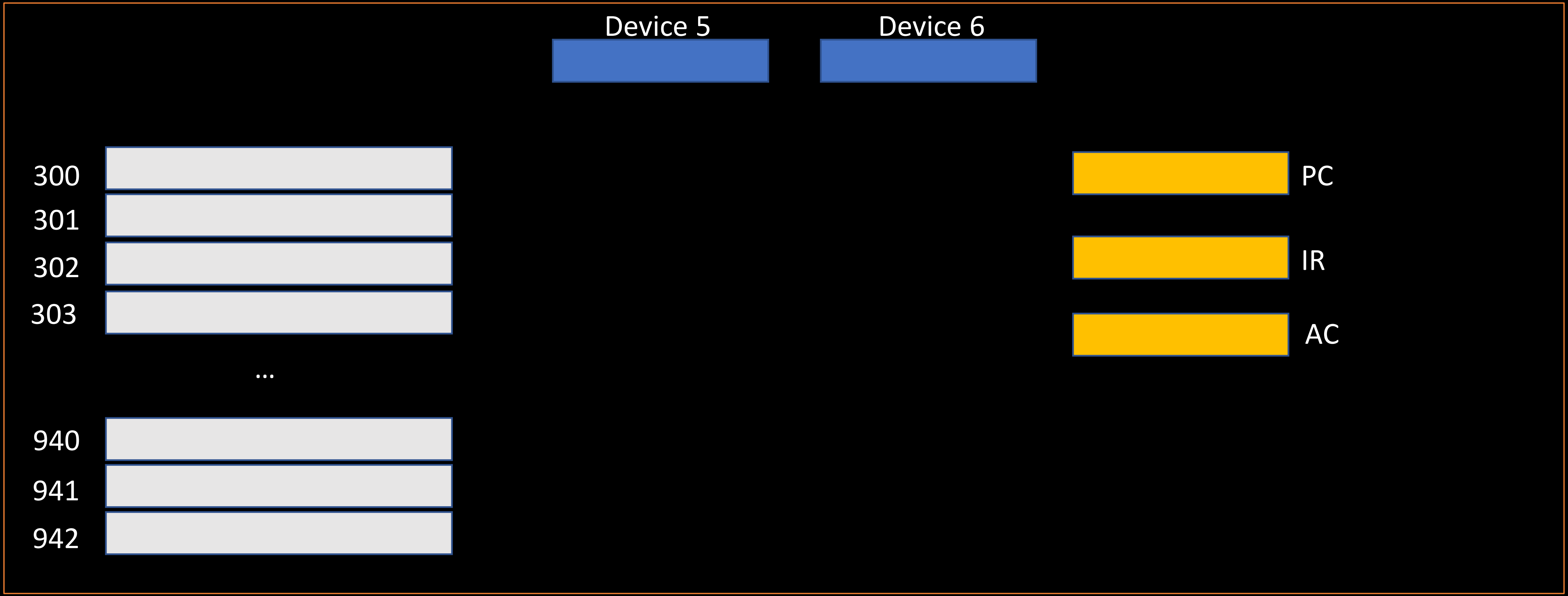
Exercises

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)

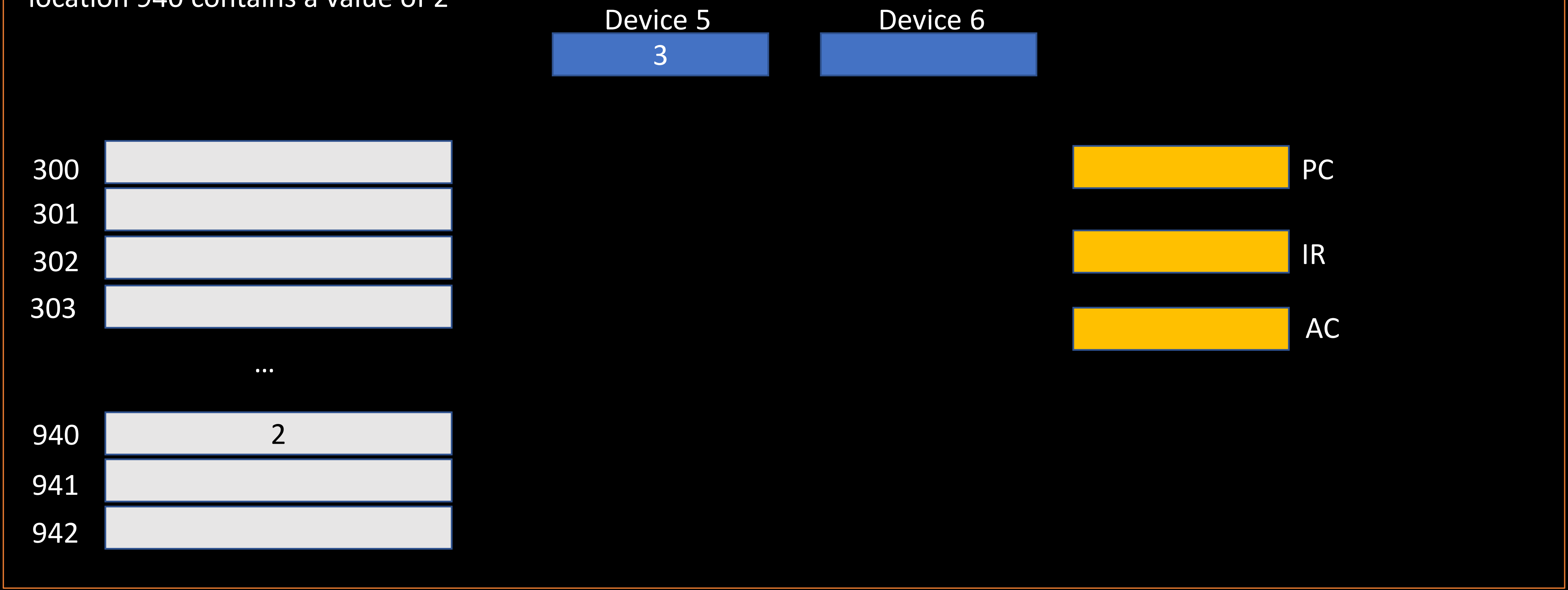
- Given that device 5 has the value 3
- Given location 940 has the value 2
- Output: memory and registers content at each step

Exercises



Exercises

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



Device 5

3

Device 6

300	
301	
302	
303	
…	
940	2
941	
942	

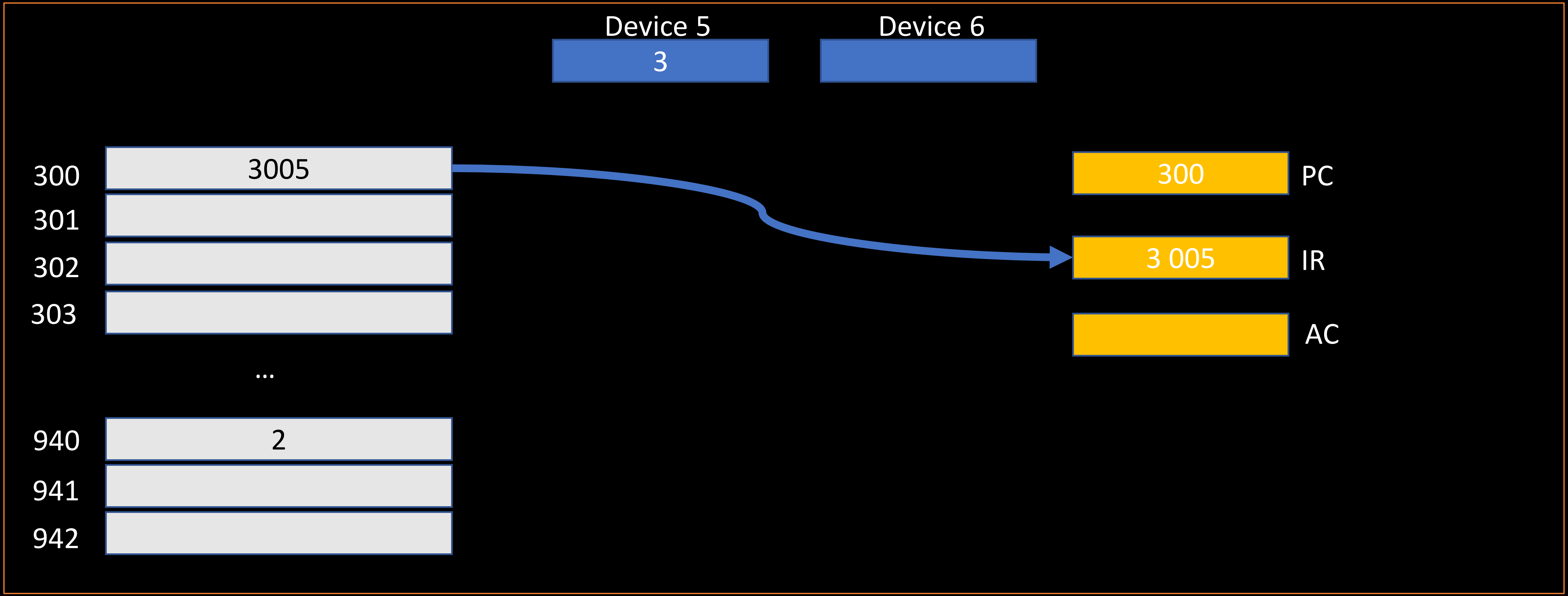
PC

IR

AC

Exercises

Load AC from device 5



Device 5

3

Device 6

300 3005

301

302

303

…

940 2

941

942

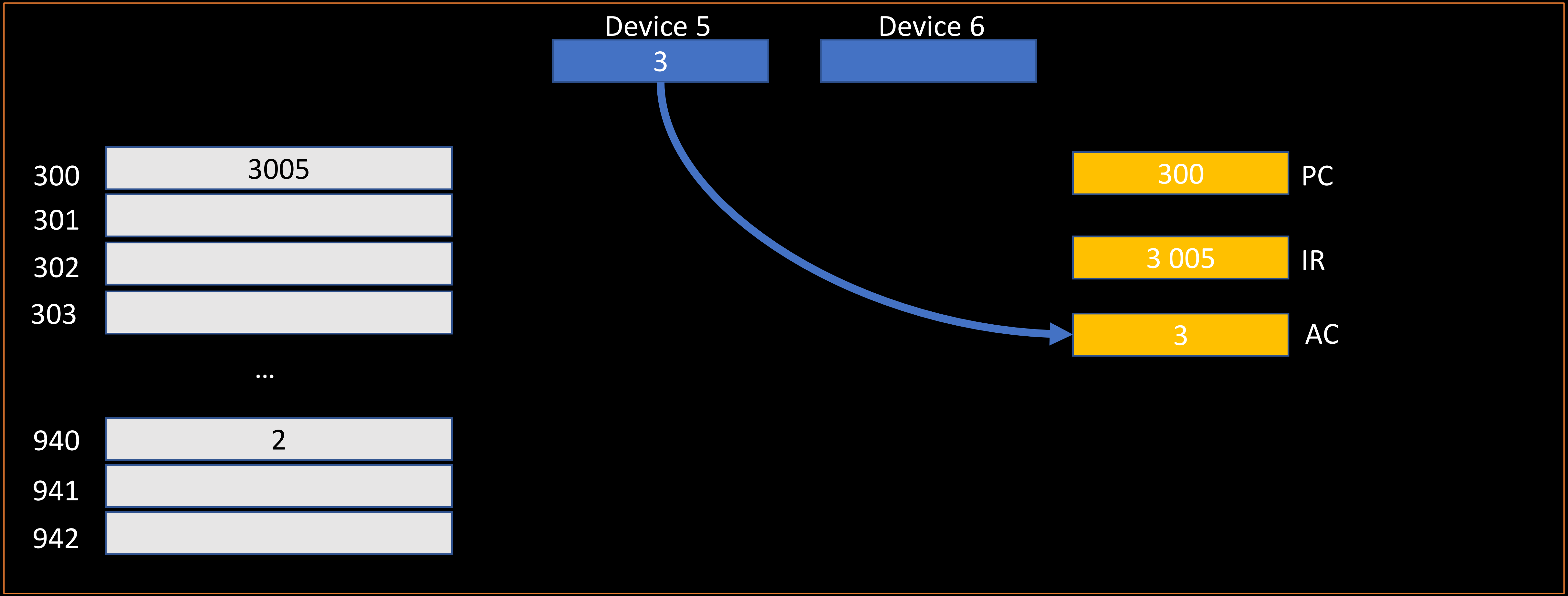
300 PC

3 005 IR

AC

Exercises

Load AC from device 5



Device 5

3

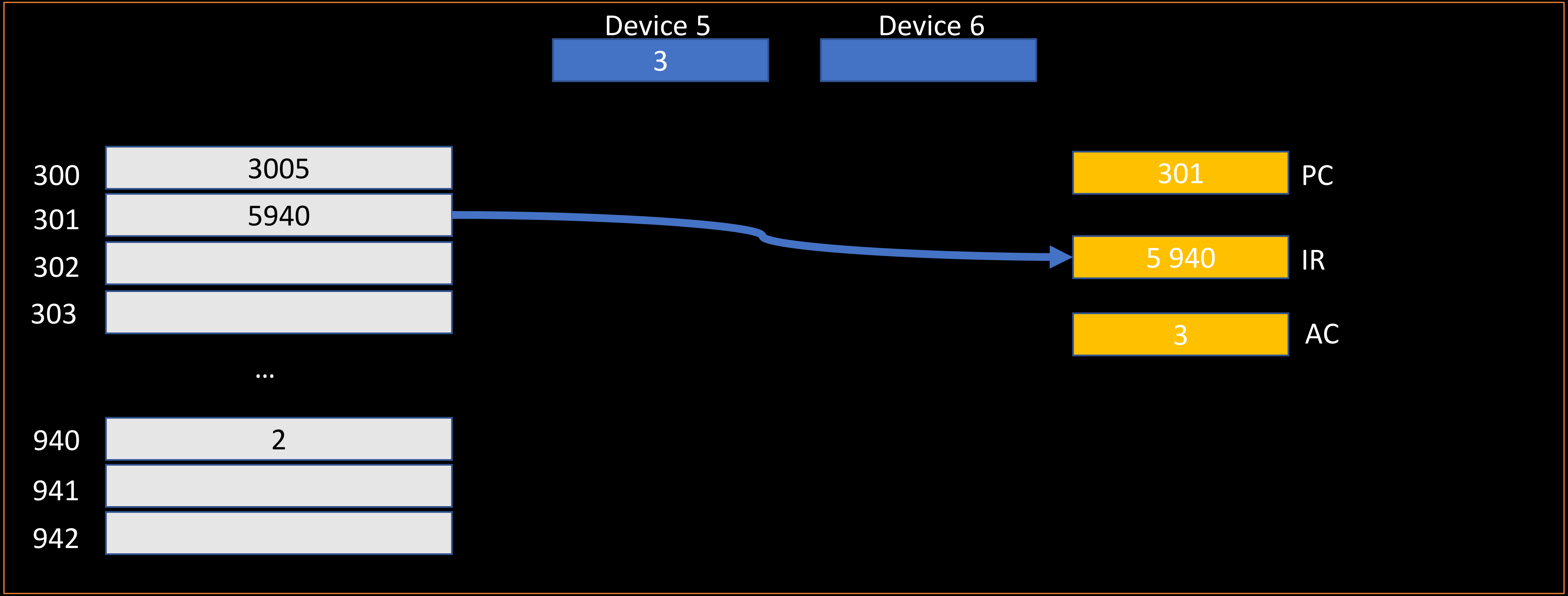
Device 6

300	3005
301	
302	
303	
	…
940	2
941	
942	

300	PC
3 005	IR
3	AC

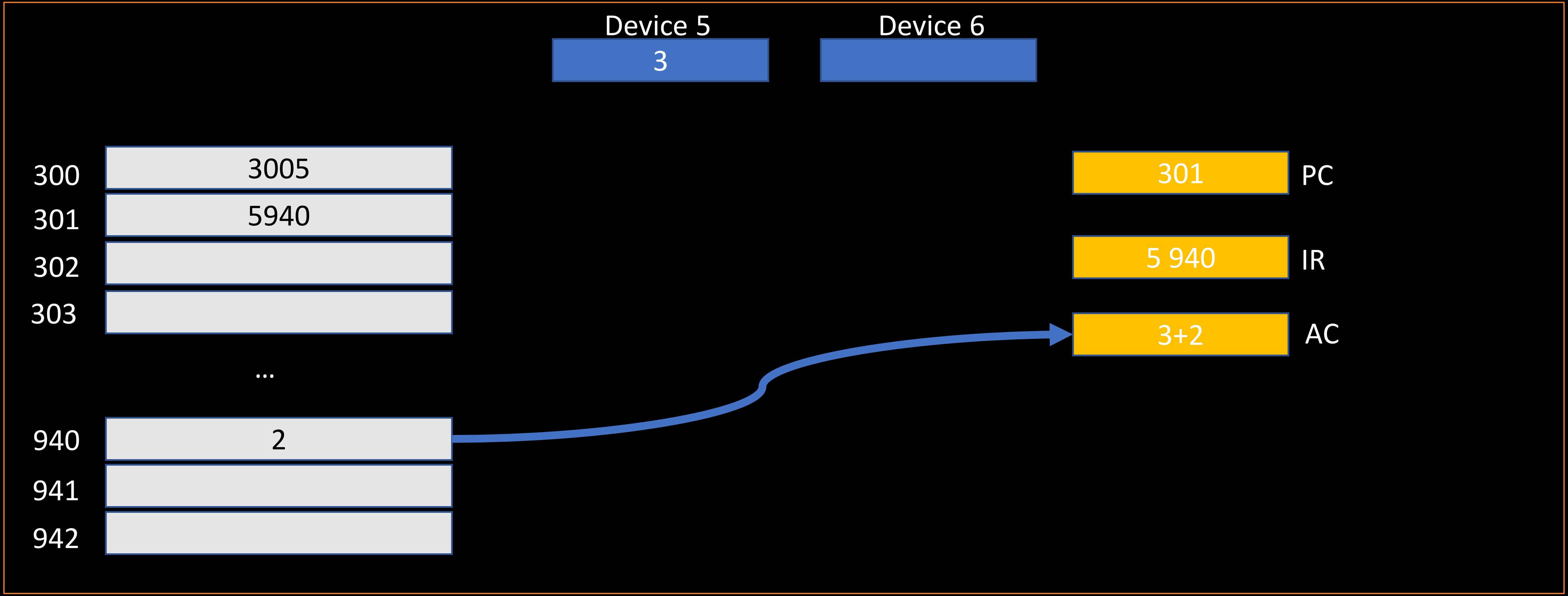
Exercises

Add contents of memory location 940



Exercises

Add contents of memory location 940

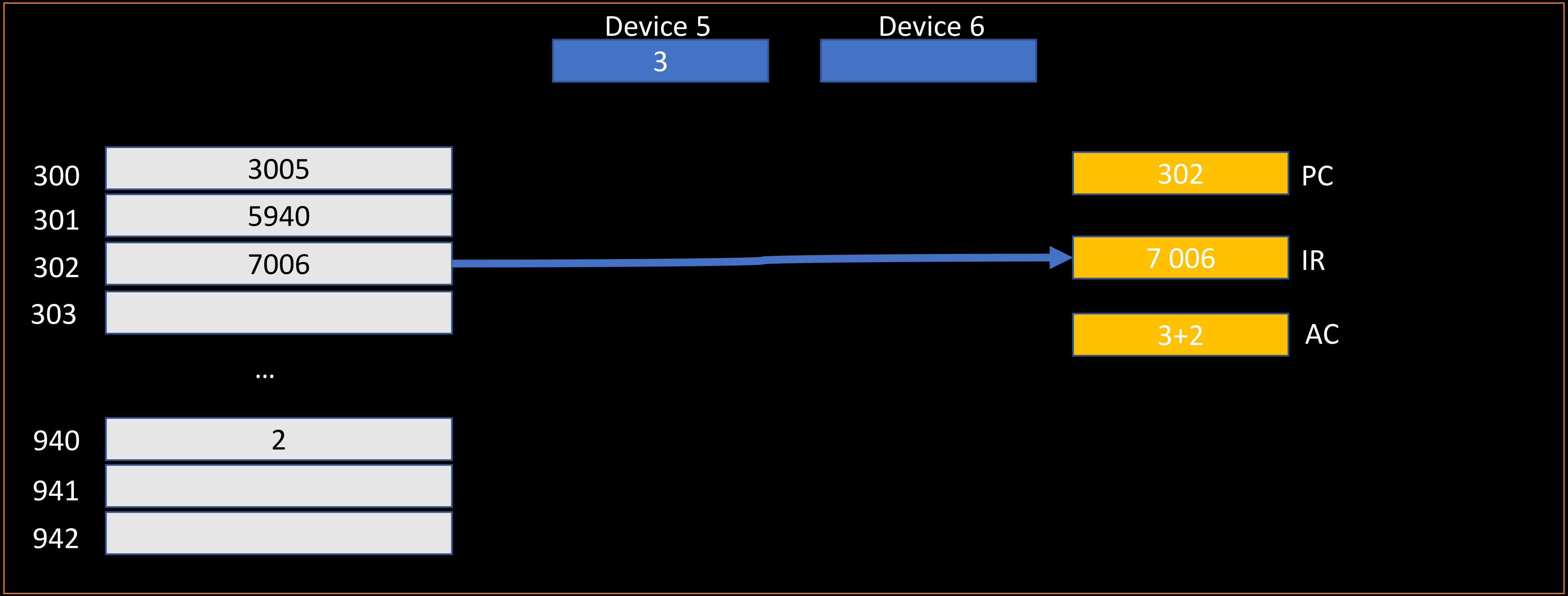


300	3005
301	5940
302	
303	
	…
940	2
941	
942	

301	PC
5 940	IR
3+2	AC

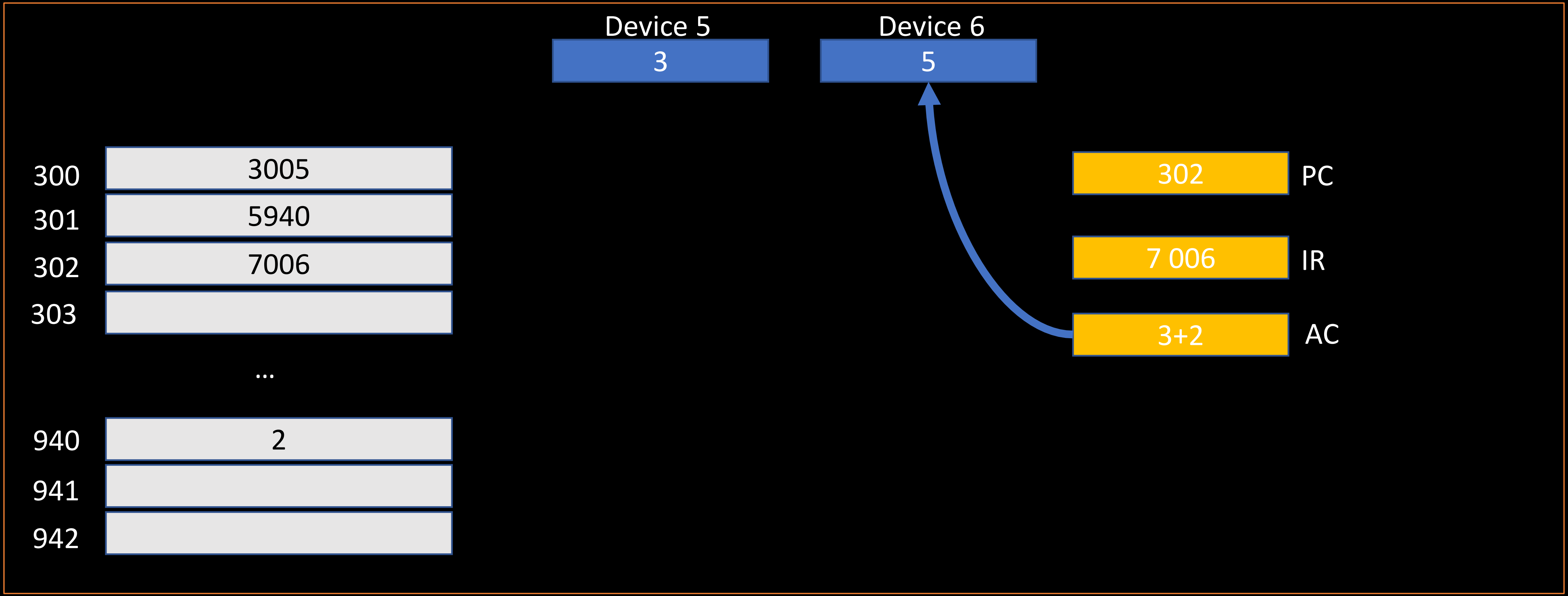
Exercises

Store AC to device 6



Exercises

Store AC to device 6



Device 5

3

Device 6

5

300	3005
301	5940
302	7006
303	
…	
940	2
941	
942	

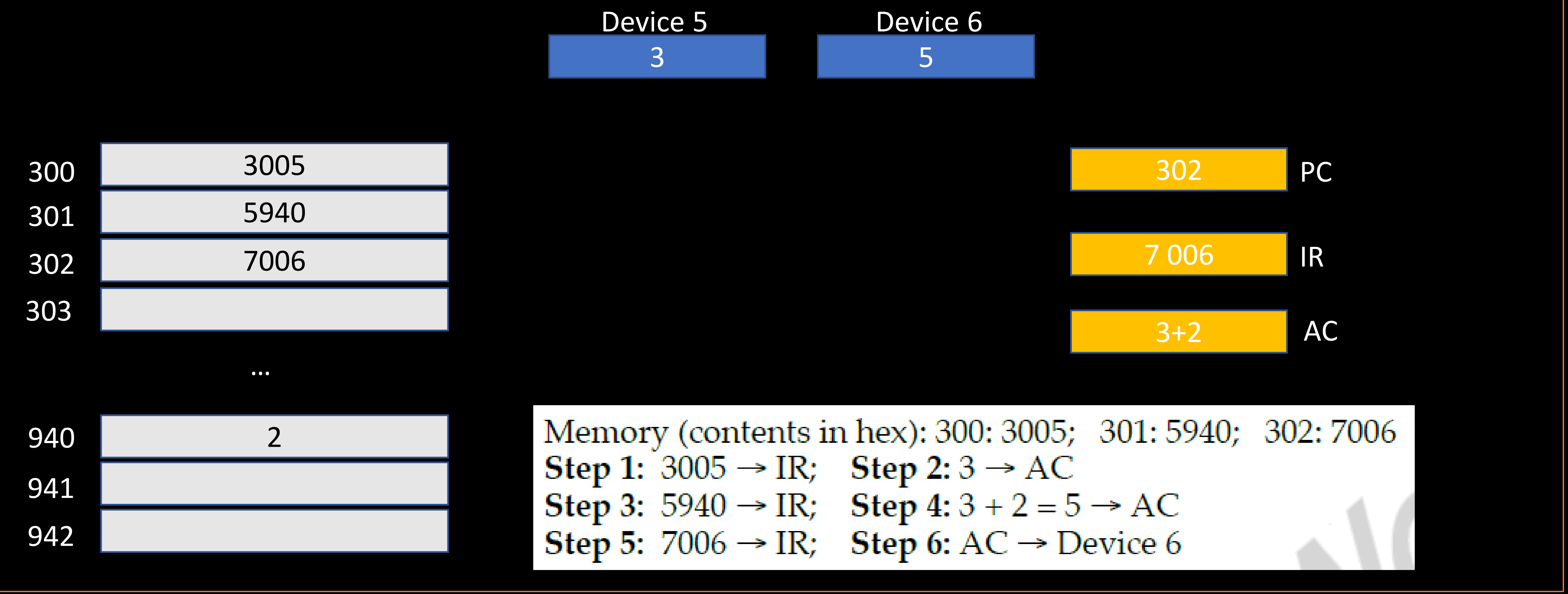
302 PC

7 006 IR

3+2 AC

Exercises

Store AC to device 6



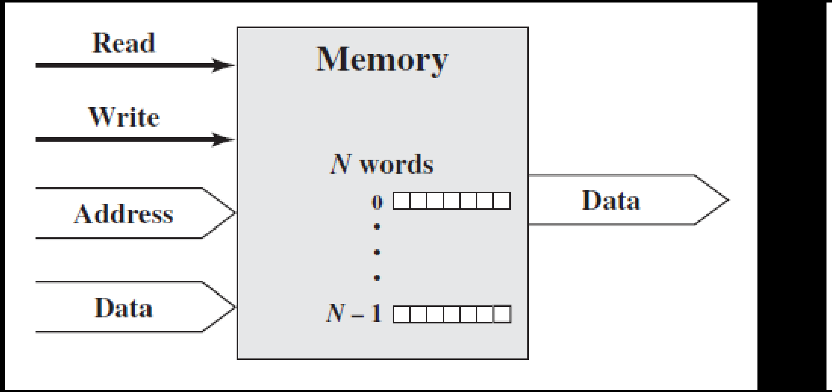
Memory (contents in hex): 300: 3005; 301: 5940; 302: 7006
Step 1: 3005 → IR; **Step 2:** 3 → AC
Step 3: 5940 → IR; **Step 4:** 3 + 2 = 5 → AC
Step 5: 7006 → IR; **Step 6:** AC → Device 6

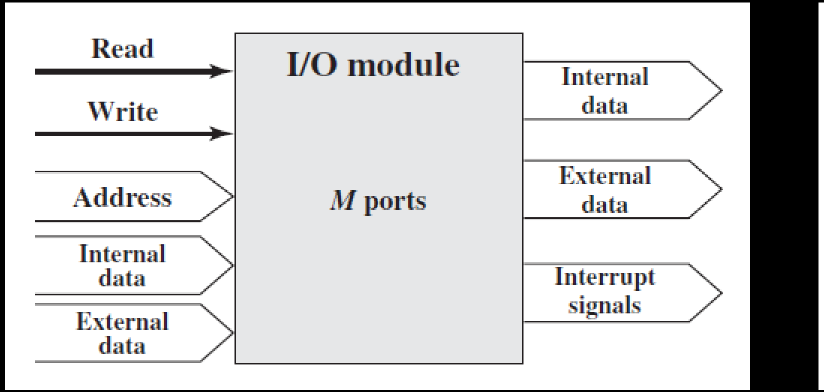
Content

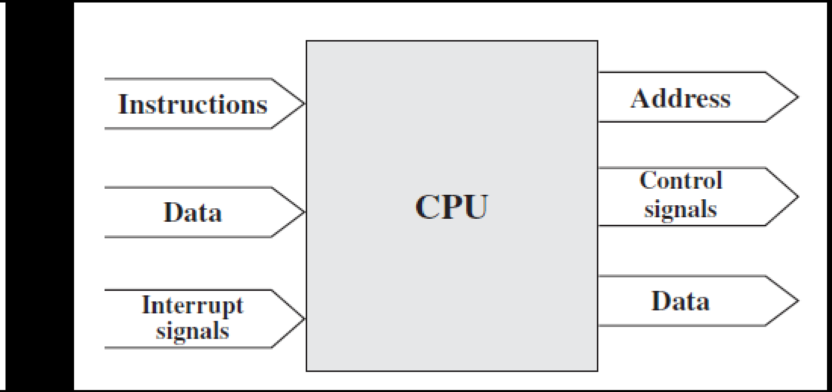
CH 03
Computer Components
Interrupts
Interconnection Structures
Bus Interconnection
Elements of Bus Design: Timing
PCI

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*.





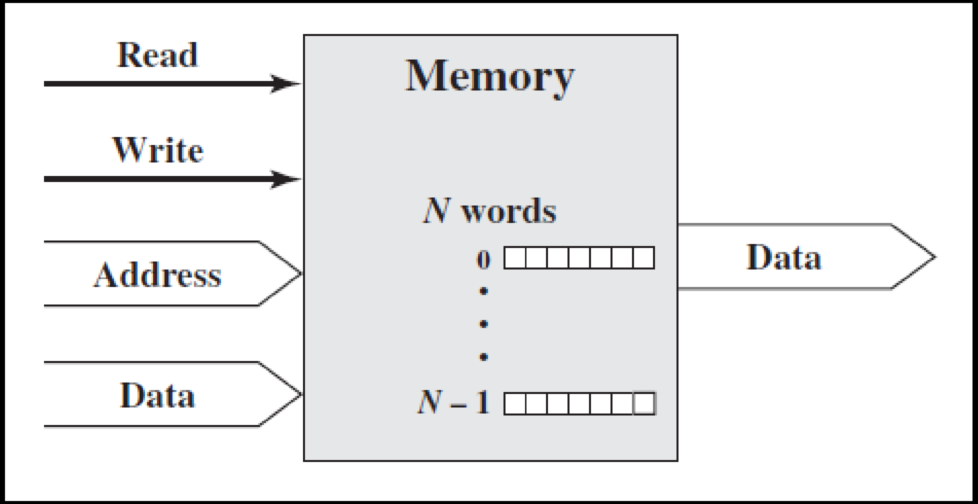


  - The wide arrows represent multiple signal lines carrying multiple bits of information in parallel. Each narrow arrows represents a single signal line.

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.

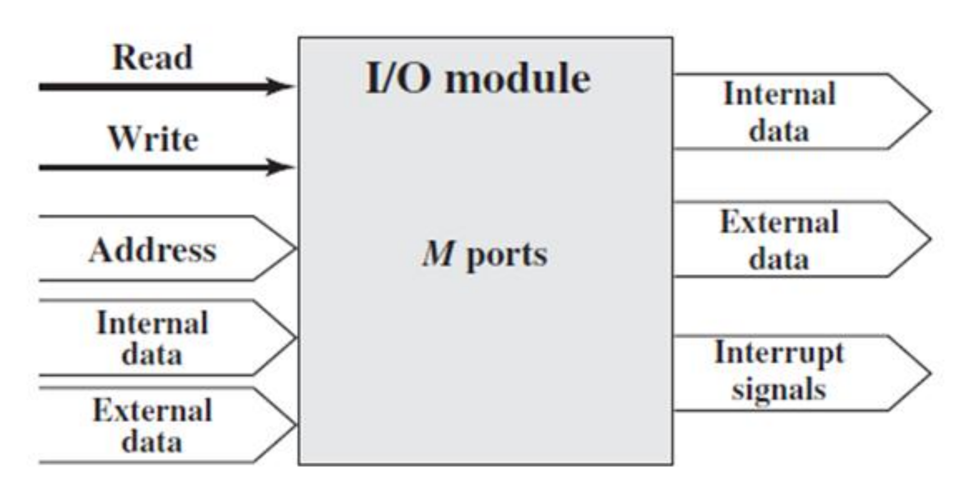


- The location for the operation is specified by an address.

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.

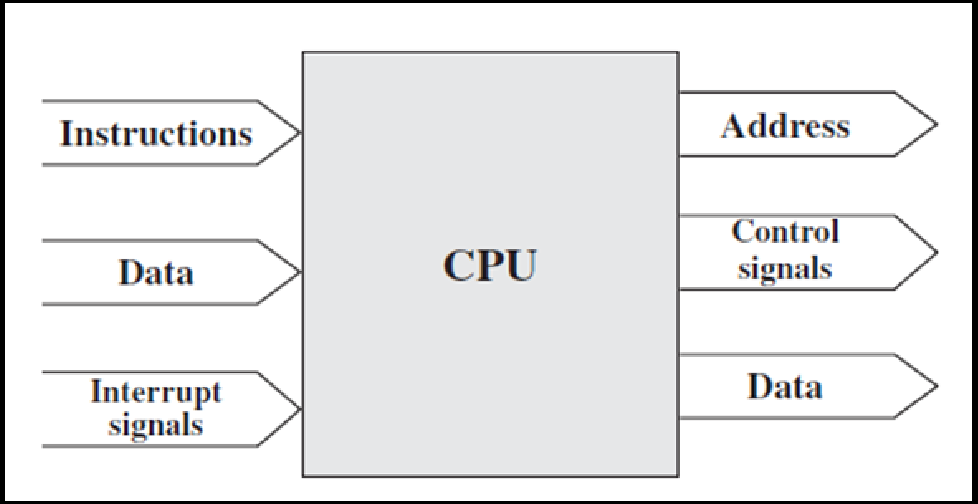


- Address used to identify the external device via *port* number.

Interconnection Structures

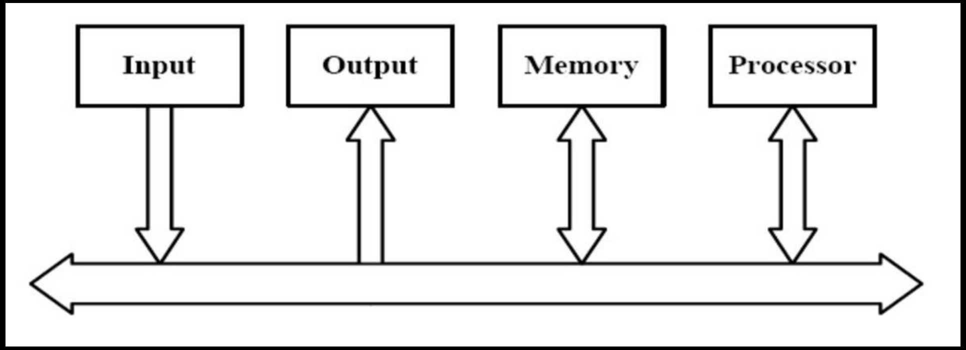
Processor:

- 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:
  - o Memory to processor
  - o Processor to memory
  - o I/O to processor
  - o Processor to I/O
  - o I/O to or from memory
- The most common interconnection structure is the bus and multiple-bus structures.



Content

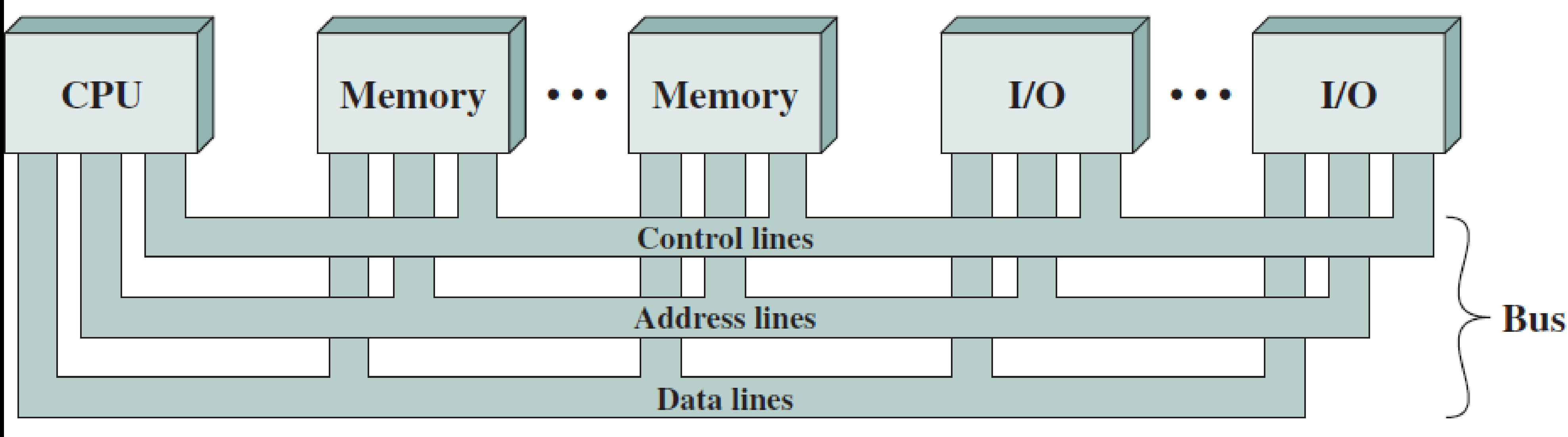
	CH 03
	Computer Components
	Interrupts
	Interconnection Structures
➡	Bus Interconnection
	Elements of Bus Design: Timing
	PCI

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?