Ch.3 Input and Output

Topics

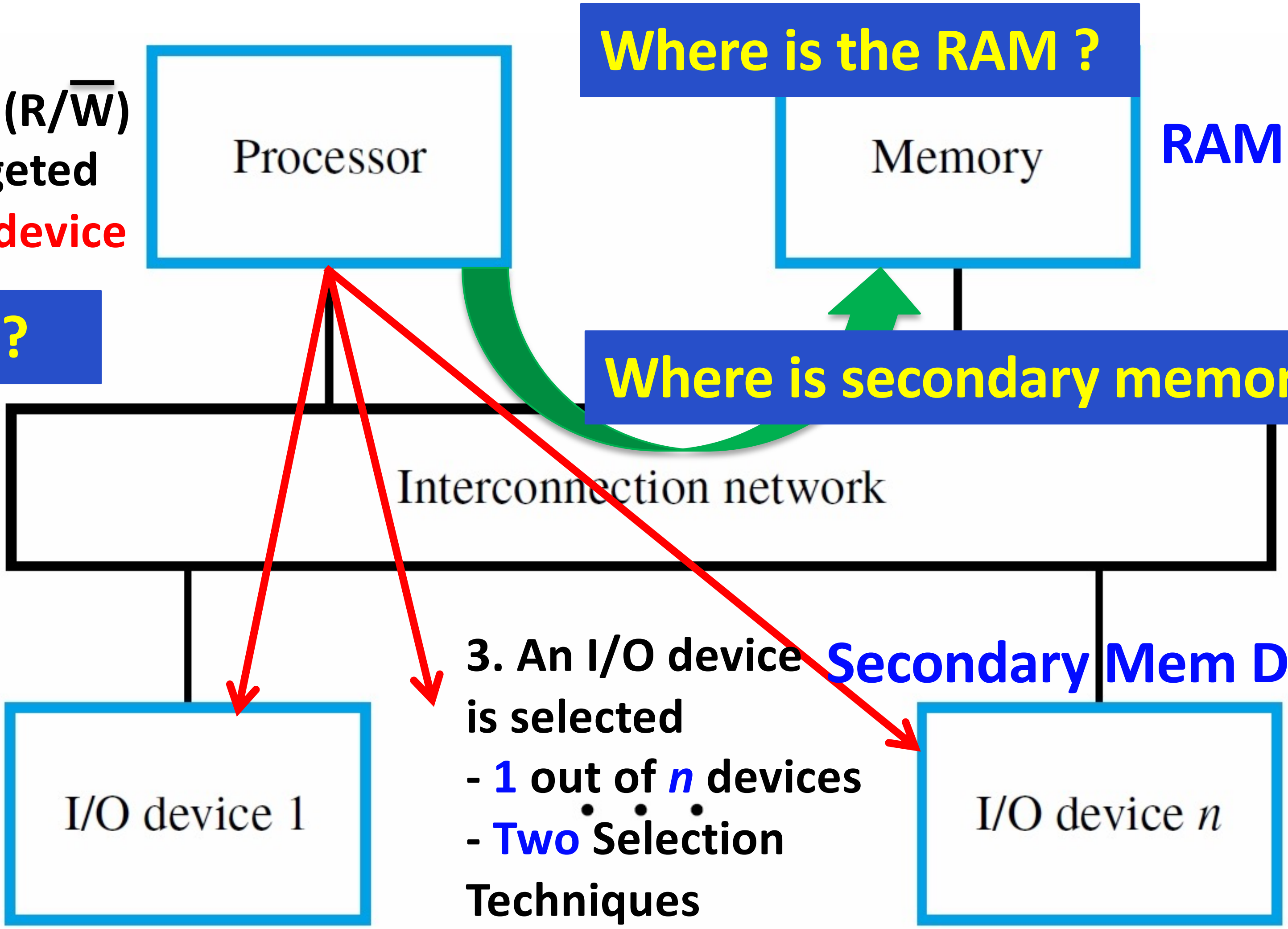
- Introduction: input and output (I/O)
  - Gaming industry devices
- **3.1.1: I/O device – system interface**
- 3.1.2: Program-controlled I/O transfer
- 3.2: Interrupt-based I/O
- 3.2.6: Exceptions

Processor to I/O Device: Transfer initiation

Processor Steps

1. Initiate transfer (R/$\overline{W}$)
2. Identify the targeted memory or I/O device

What is R / $\overline{W}$?



Where is the RAM ?

RAM

Where is secondary memory ?

Secondary Mem Disk

3. An I/O device is selected
- 1 out of n devices
- Two Selection Techniques

Selected I/O Device; last step

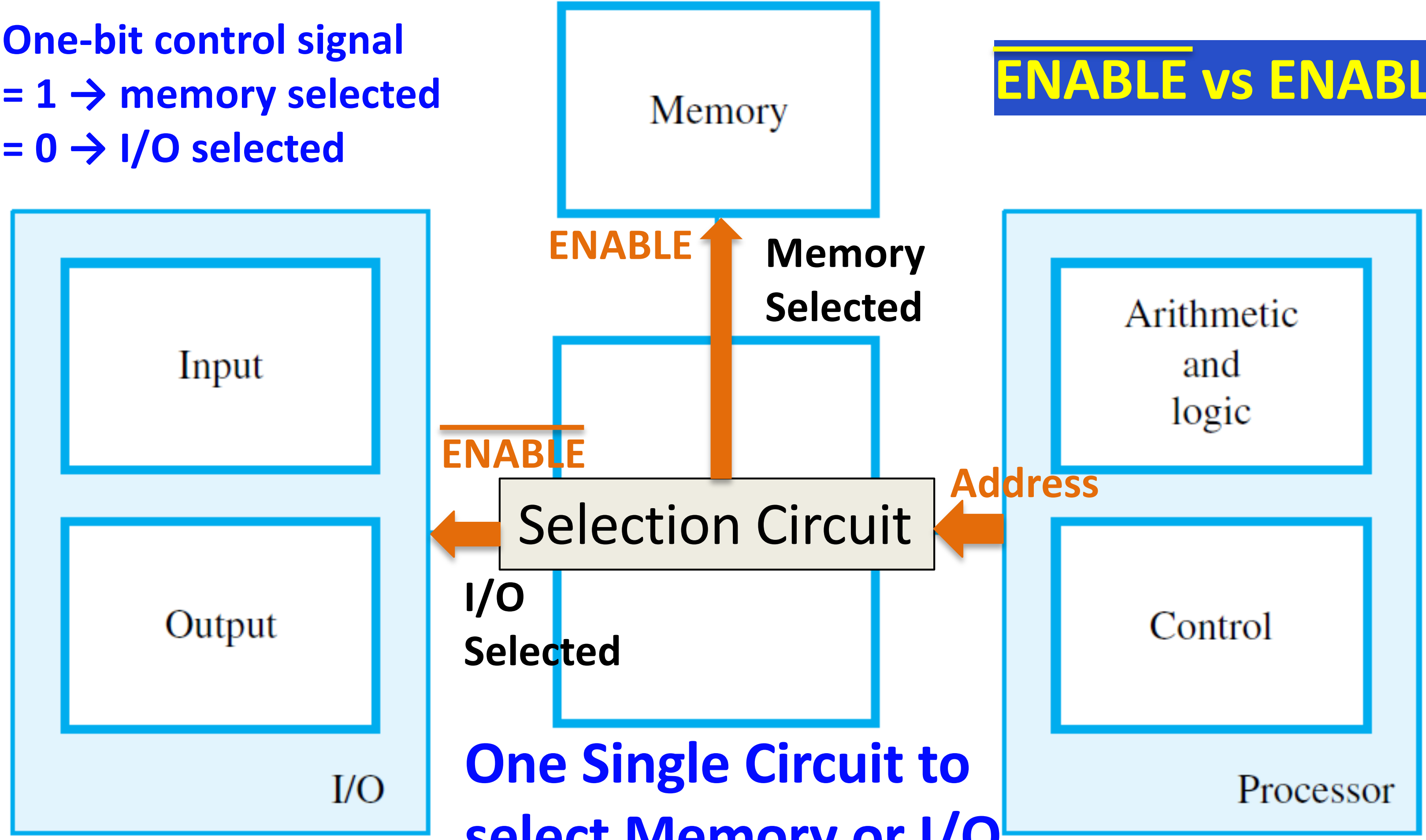
4. Respond to Master when selected

I/O Selection Technique 1: Memory-Mapped

ENABLE:

- One-bit control signal
- = 1 → memory selected
- = 0 → I/O selected

$\overline{\text{ENABLE}}$ vs ENABLE ?



One Single Circuit to select Memory or I/O

3.1 Memory-Mapped I/O Selection

- Locations of **I/O device's registers (control & data storage)** are in the same address space as memory (0 to 2^{k}–1)

Assume 3-bit address:
8 locations

Binary			Address Space	
A2	**A1**	**A0**	**Memory**	**I/O**
0	**0**	**0**	**0**	
0	**0**	**1**	**1**	
0	**1**	**0**	**2**	
0	**1**	**1**	**3**	
1	**0**	**0**	**4**	
1	**0**	**1**	**5**	
1	**1**	**0**		**6-Input-1**
1	**1**	**1**		**7-Output-1**

Memory:
Address 0 to 5

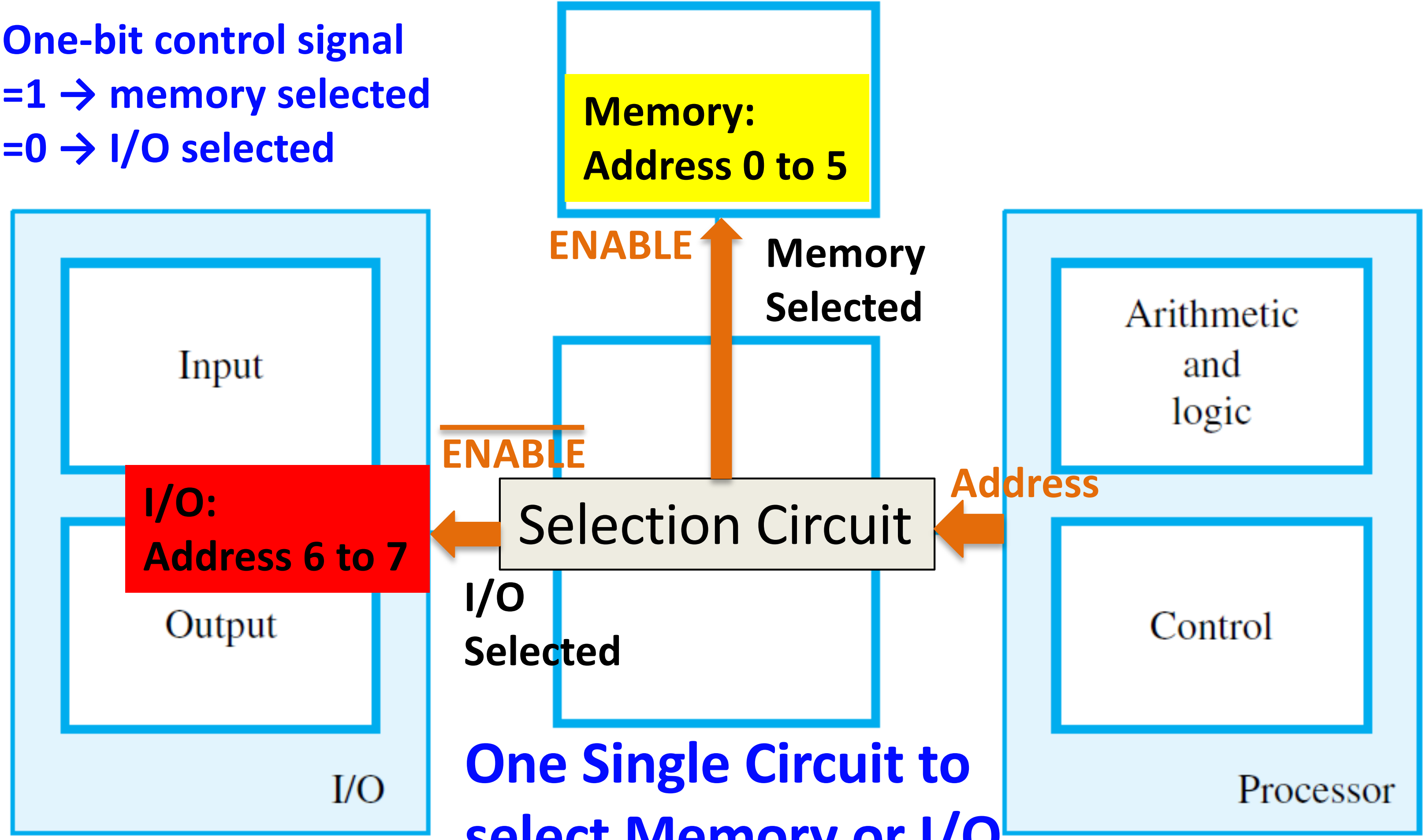
I/O:
Address 6 to 7

- Use Load/Store instructions to access input/output
- Output: **Store** R0, Output-1 (Output-1=%111=7)
- Input: **Load** Input-1, R1 (Input1=%110=6)
- Memory address: [0..5] (note: less space for programs)

Memory-Mapped I/O and Memory Selection

ENABLE:

- One-bit control signal
- =1 → memory selected
- =0 → I/O selected



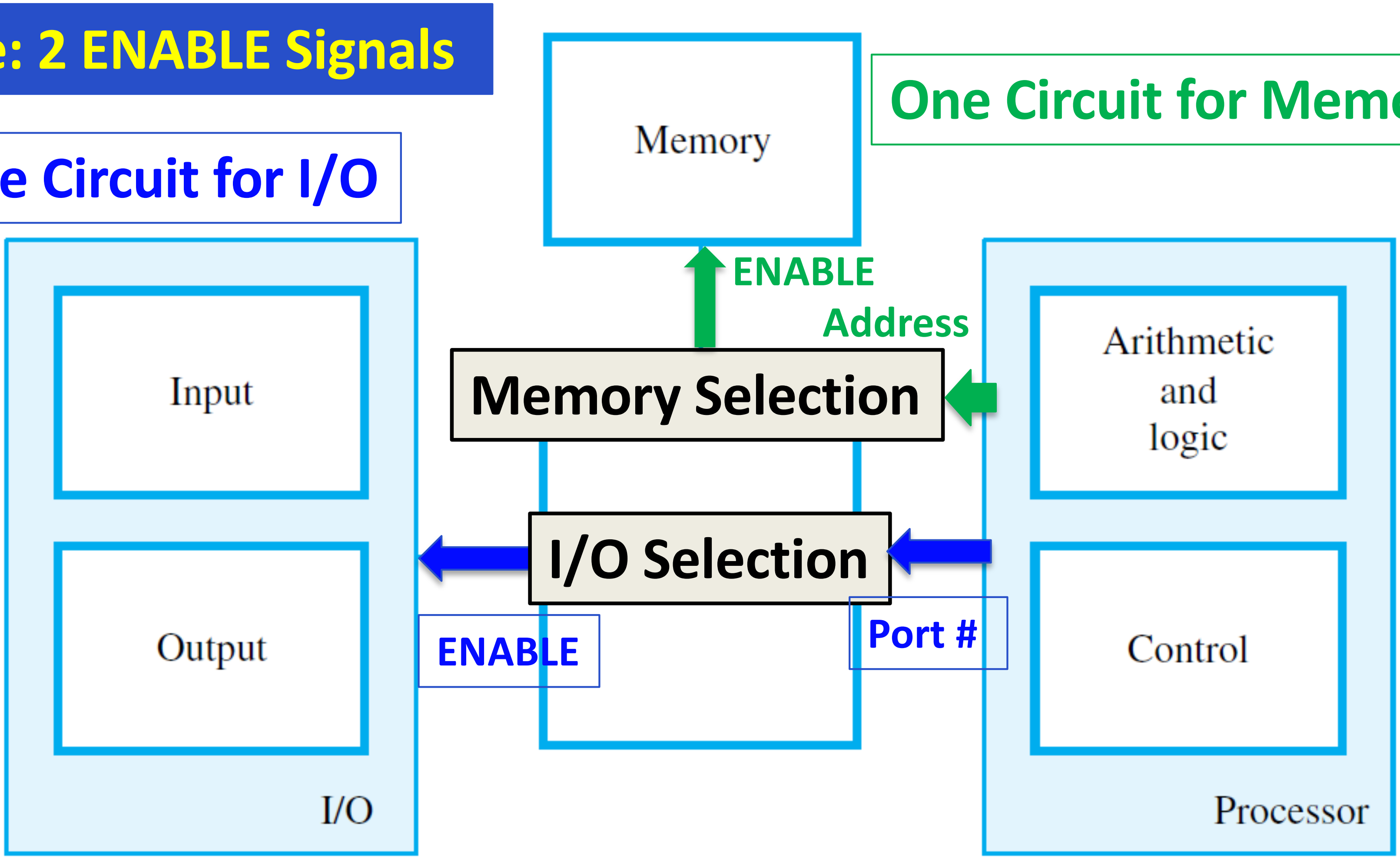
One Single Circuit to select Memory or I/O

I/O Selection Technique 2: I/O-Mapped

Note: 2 ENABLE Signals

One Circuit for I/O

One Circuit for Memory



I/O-Mapped I/O Selection

- Each I/O device and its registers are assigned a distinct "I/O address space" or "Ports"

More Instructions ?
Good or Bad

  - Use special I/O instructions (i.e., more instructions in ISA!)
  - Output: **Output** R6, OutPort1 ; OutPort1 ← R6
  - Input: **Input** InPort1, R5 ; R5 ← InPort1
  - Need additional circuit to select (i.e., more hardware!)
  - BUT Full memory space is available for programs (vs less space in Memory-mapped case)

I/O-Mapped I/O and Memory Selection

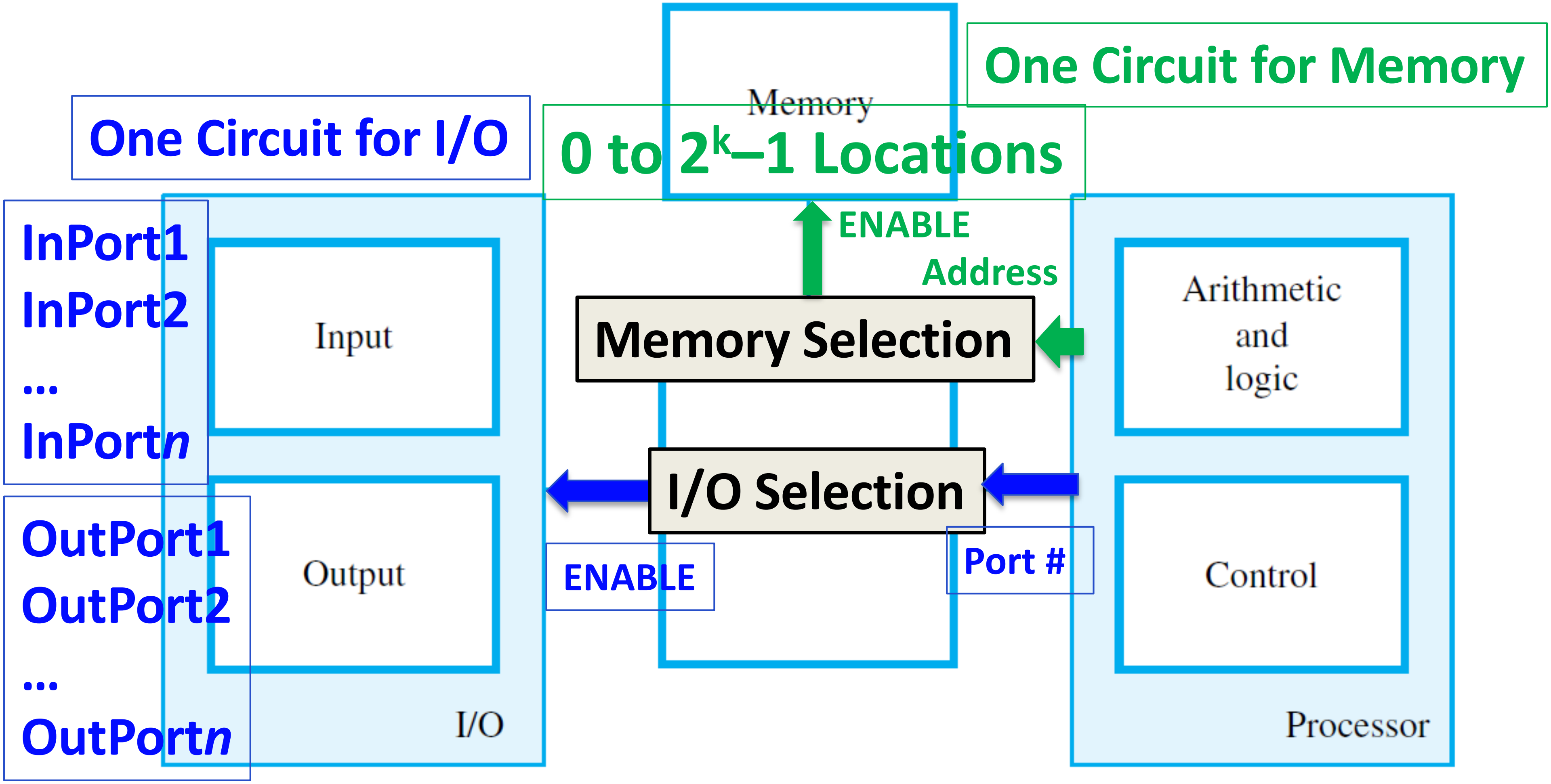
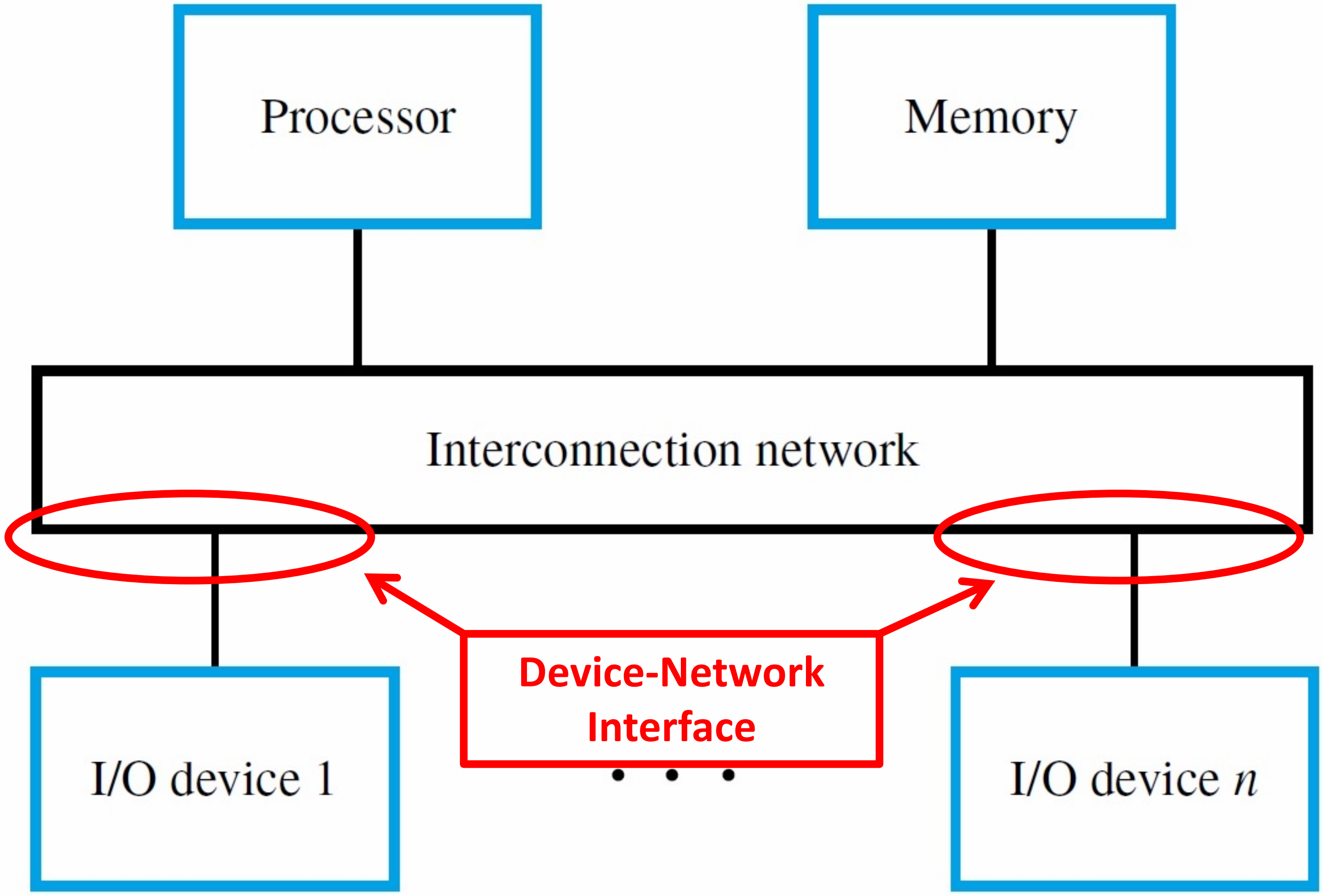


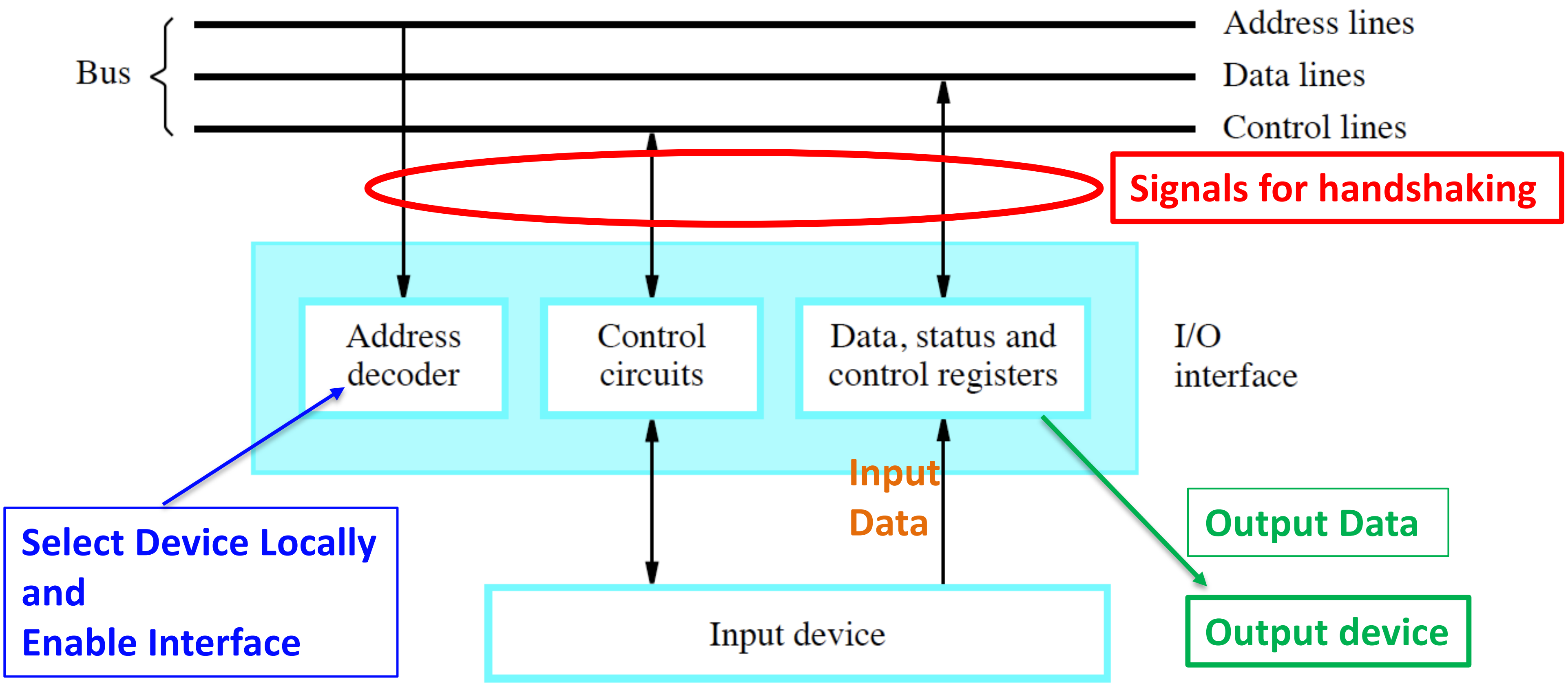
Fig.3.2: Interconnection Network



3.1.1. I/O Device-Network Interface

- Interconnection Network = System Bus
  - Data — Number of signals or lines = word size (16, 32, 64 etc.)
  - Address — Number of signals or lines often = word size
  - Control — Number of signals or lines depends on the CPU design
- Device-Network Interface:
  - An electronic circuit
  - Between a device and the network
  - Provides control, data transfer, and status information to/from the device

Fig. 7.2: I/O Interface for Input



Signals for handshaking

Select Device Locally and Enable Interface

Input Data

Output Data

Output device

Is this Memory-mapped or IO-mapped ?

Fig. 3.2: Interface Blocks

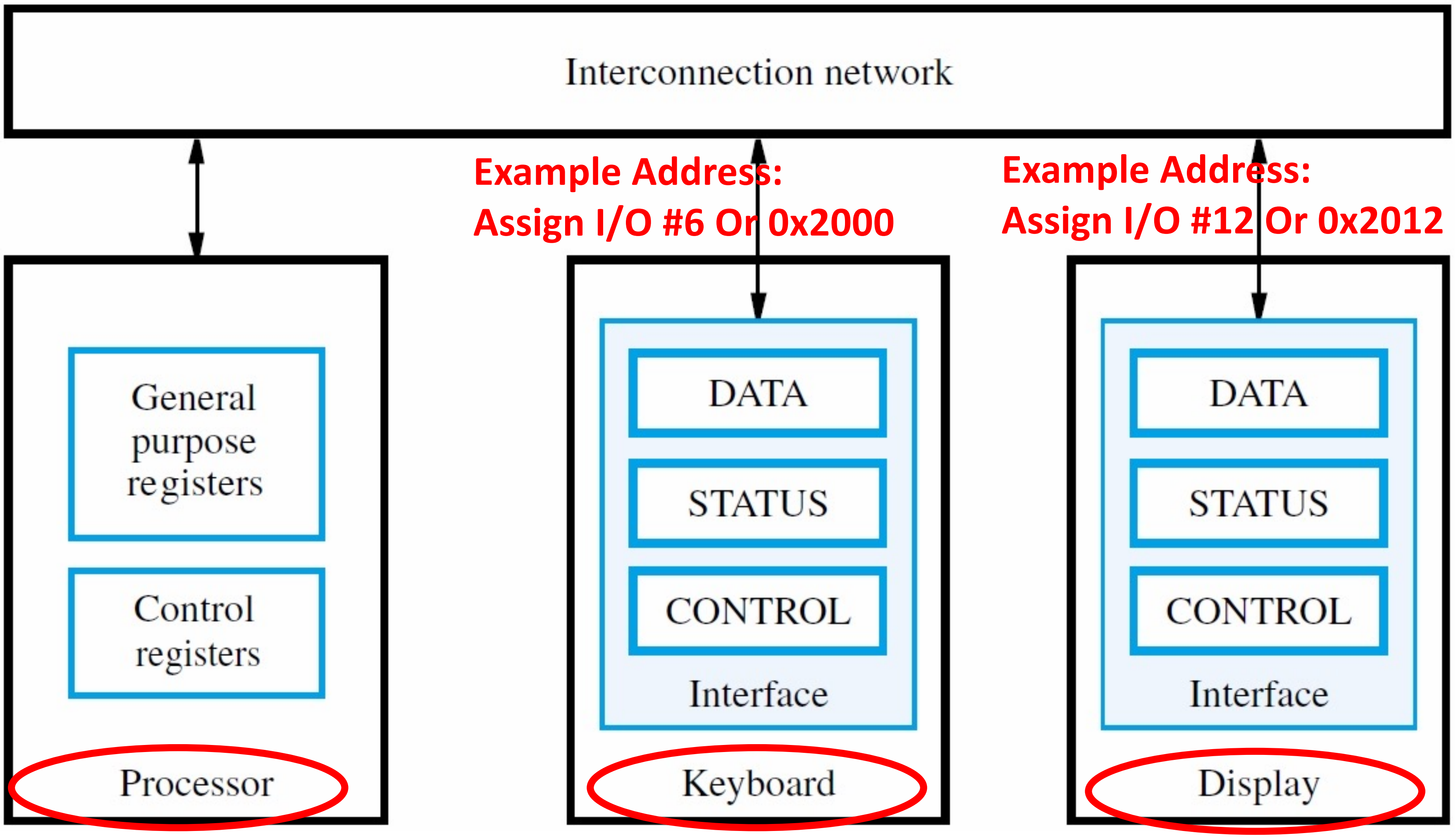
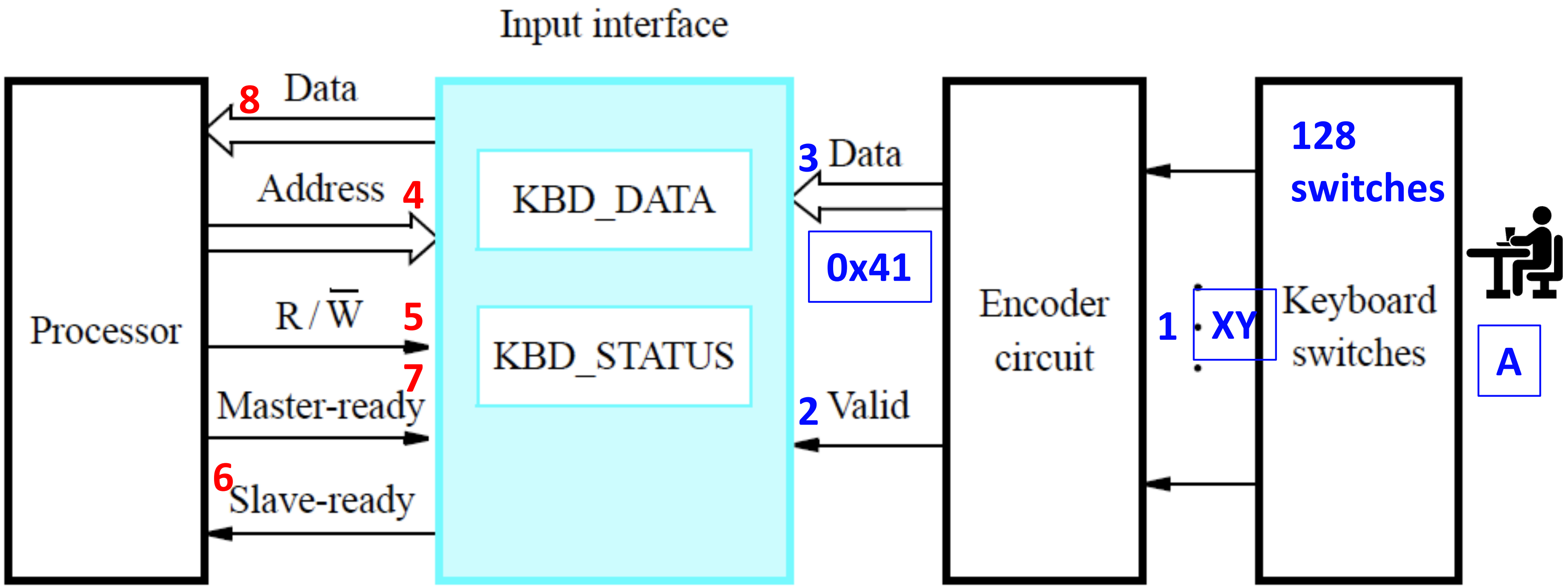


Fig. 7.10: Keyboard and Processor Interaction

Signal Sequence: e.g. 1,2…8 in order (a handshaking protocol example)



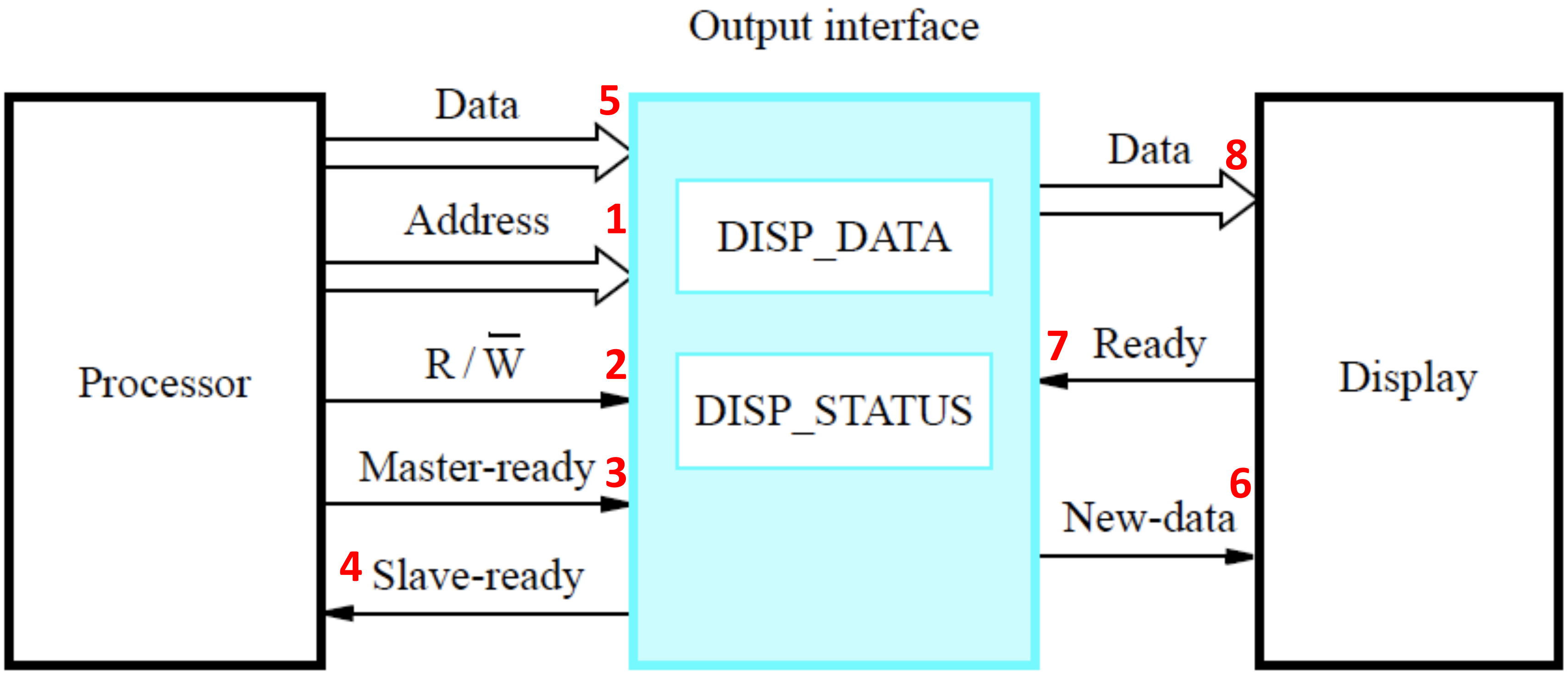
4. Address to select device for Input
5. R/W to select direction
6. I am the selected Slave and I am ready; or I am not ready, Processor to try later, go to 4
7. I am the Master and I am reading now
8. Supply Data

2. Data is valid
3. Supply Data

1. Key position

Fig. 7.13: Display and Processor Interaction

Signal Sequence: e.g. 1,2…8 in order (a handshaking protocol example)



1. Address to select device for Output
2. R/$\overline{W}$ to select direction
3. I am the Master and I am ready now
4. I am the selected Slave and I am ready; or not ready and go to 1
5. Supply Data

6. I have new data to display
7. I can accept new data
8. Supply Data