Uvod v računalništvo

Aleksander Sadikov

2015/2016

Computer organization studies computers in terms of their major functional units.

Virtually every computer in use today is based on a single design, be it \$1M supercomputer, \$1k laptop, or \$100 smartphone.

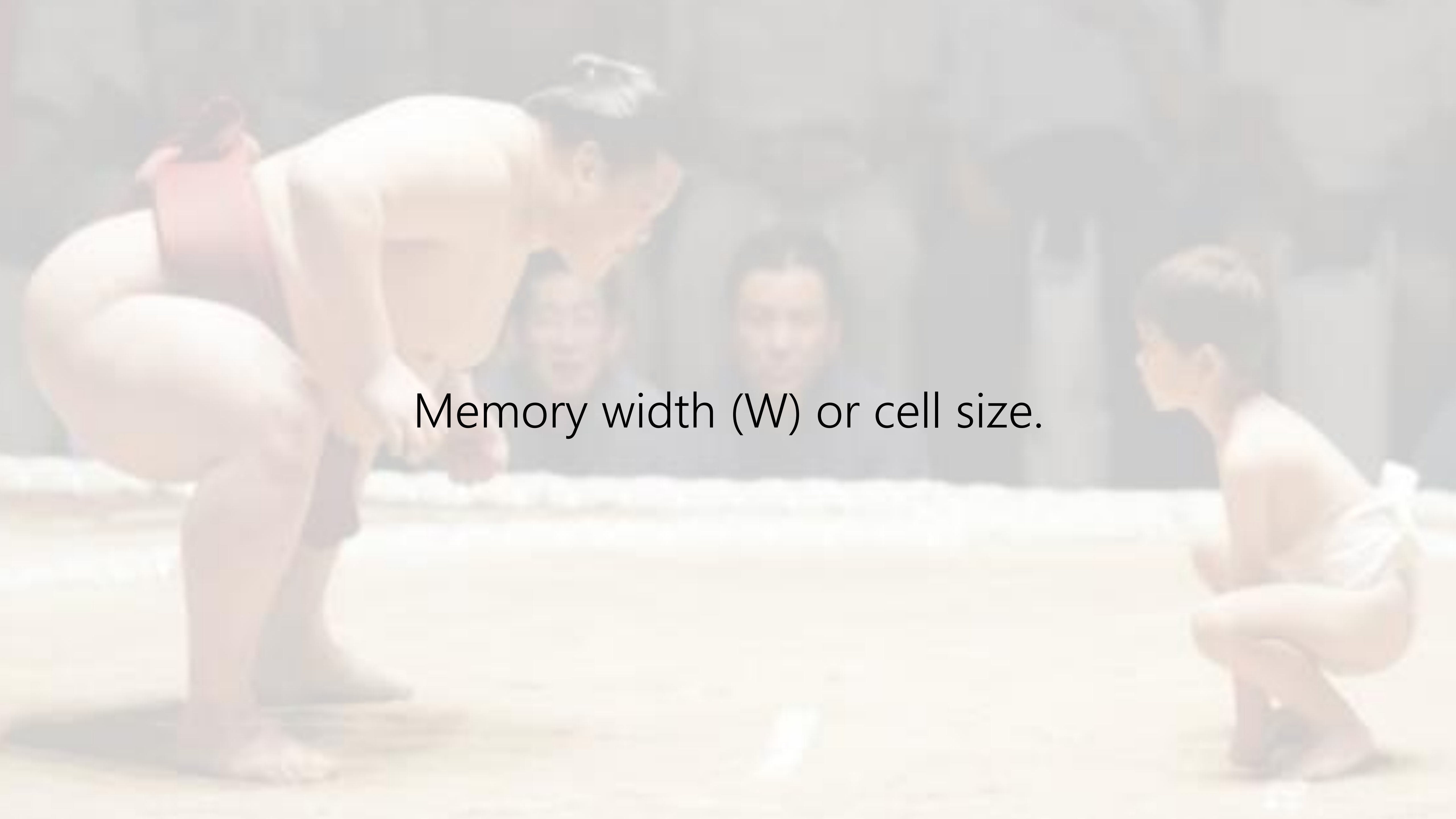
The Von Neumann architecture (1946).

1. Four major subsystems: memory, input/output, the arithmetic/logic unit, and the control unit.
2. The stored program concept
3. The sequential execution of instructions

Memory & Cache

1. Divided into fixed-size units (**cells**);
   each cell has its own unique **address**.
2. We must always fetch or store a complete cell.
3. The time to fetch or store is the same
   for all the cells in memory.

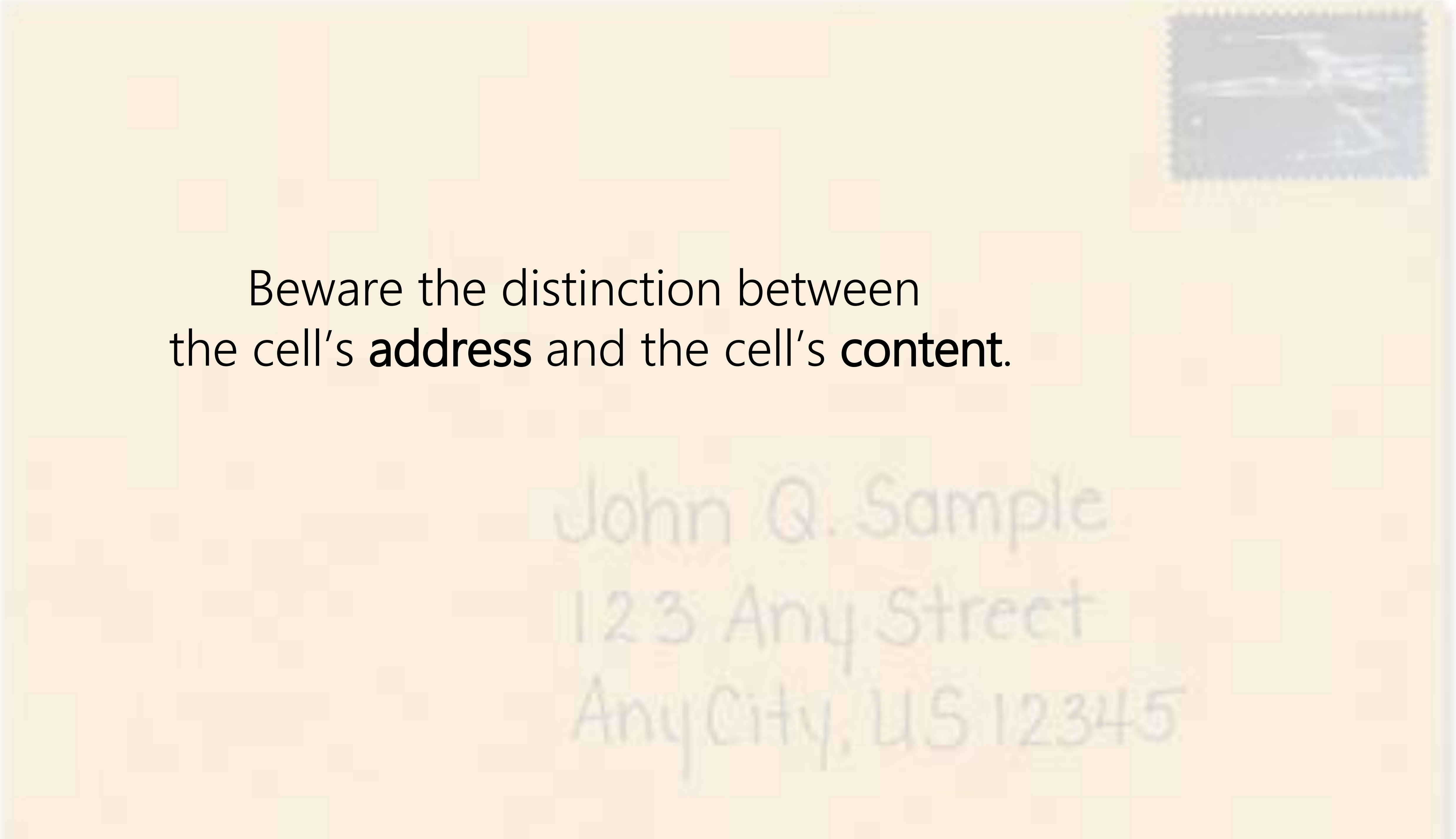
Memory width (W) or cell size.



Memory width (W) or cell size.

Address space and maximum memory size

Beware the distinction between
the cell's **address** and the cell's **content**.



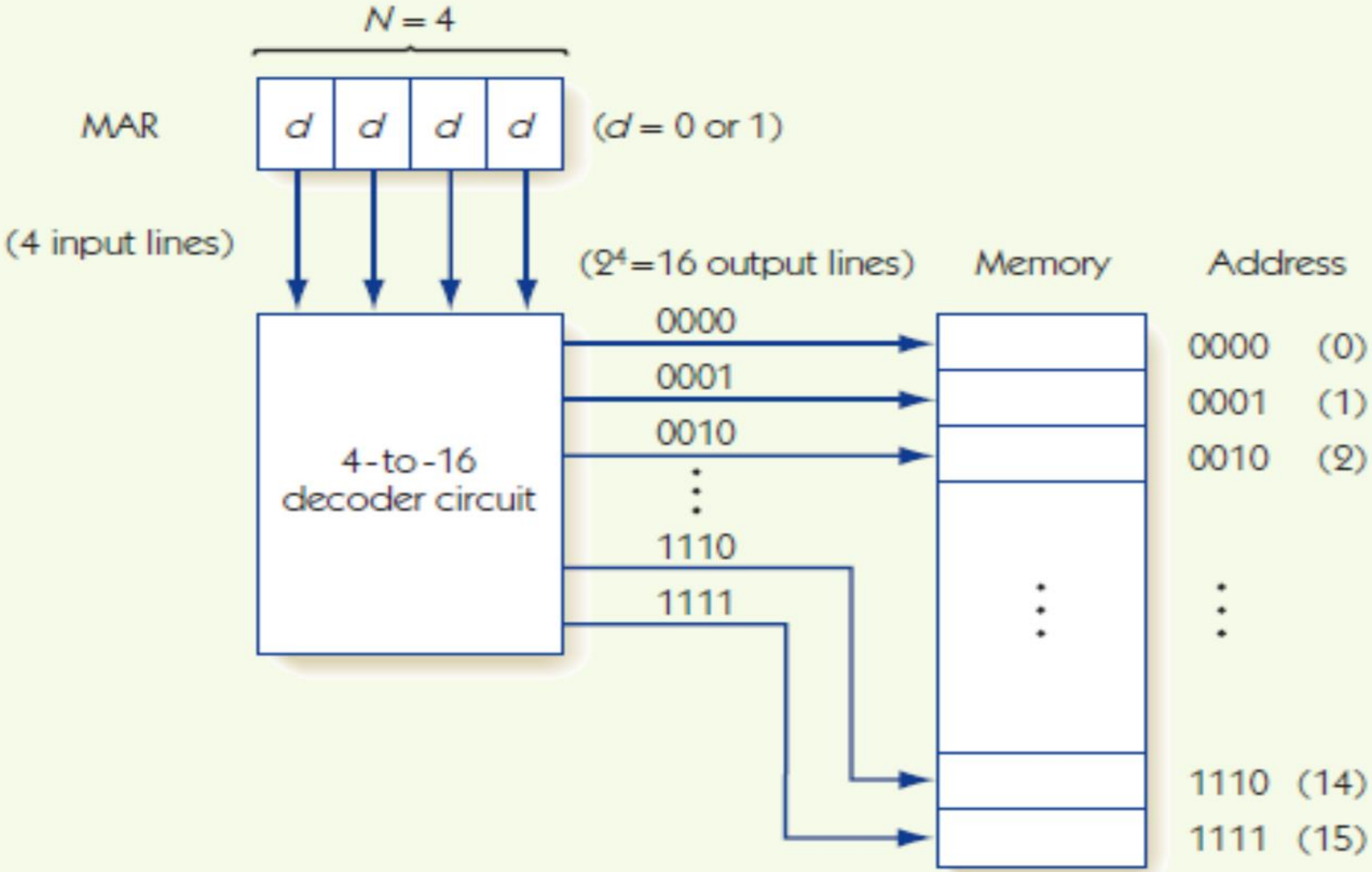
Memory access time.

Fetching & storing.

Memory Address Register (MAR) and Memory Data Register (MDR).

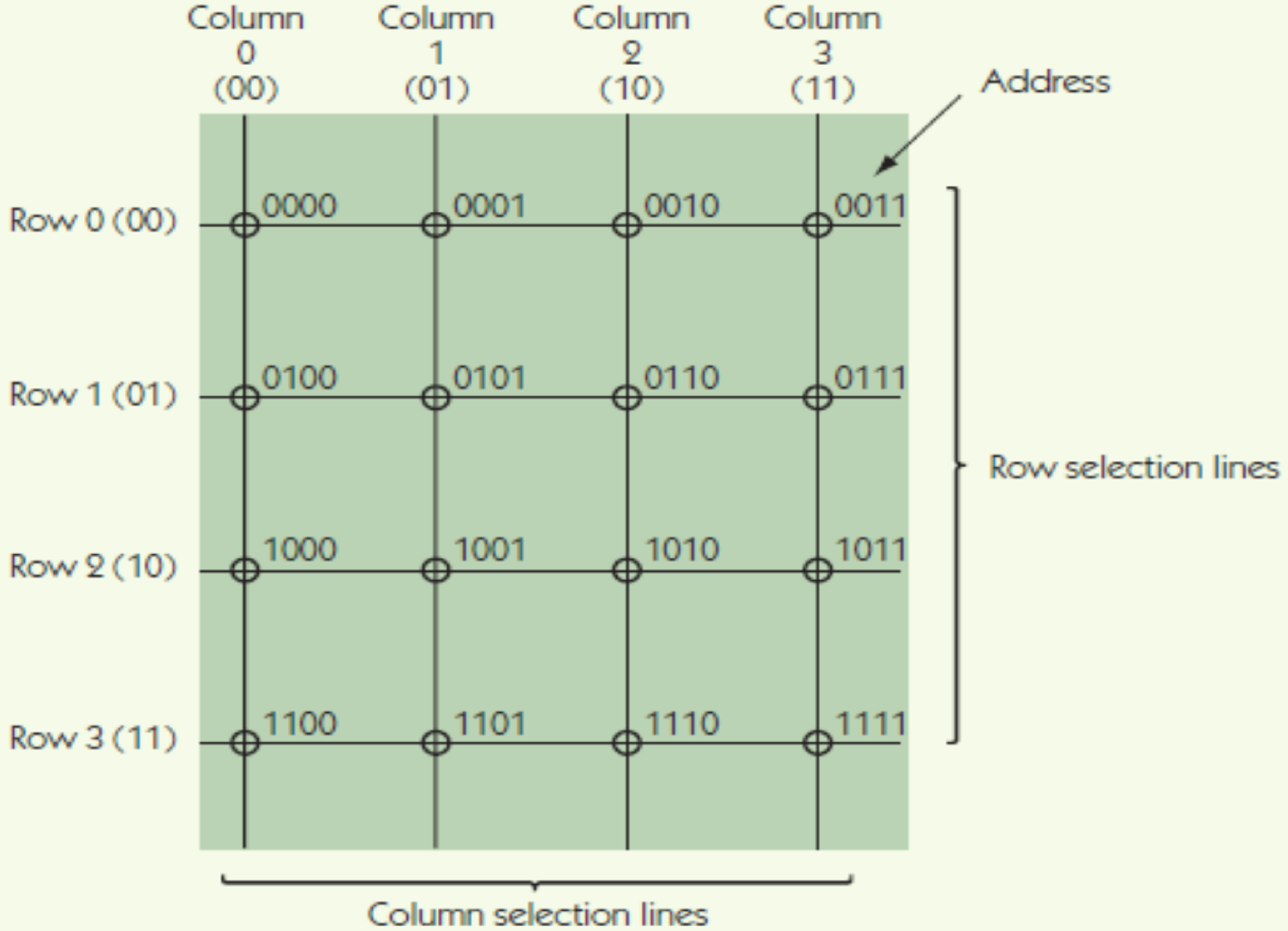
1. Load the address into the MAR.
2. Decode the address in the MAR.
3. Copy the contents of decoded location into the MDR.

1. Load the address into the MAR.
2. Load the value into the MDR.
3. Decode the address in the MAR.
4. Store the contents of the MDR into the decoded location.



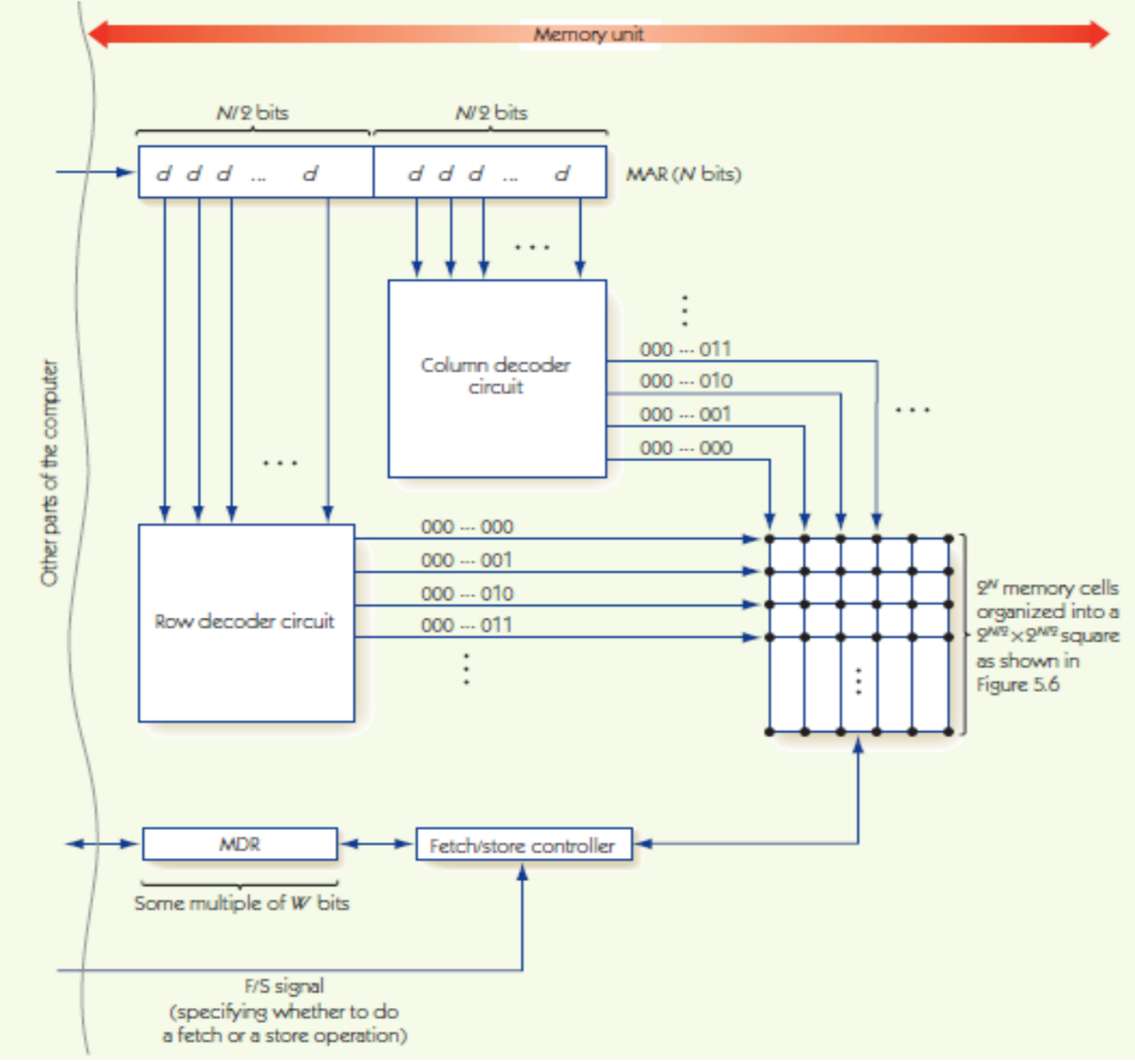
N = 4
MAR
d
d
d
d
(d = 0 or 1)
(4 input lines)
(2⁴=16 output lines)
Memory
Address
4-to-16
decoder circuit
0000
0001
0010
1110
1111
0000 (0)
0001 (1)
0010 (2)
1110 (14)
1111 (15)

The problem of scaling.



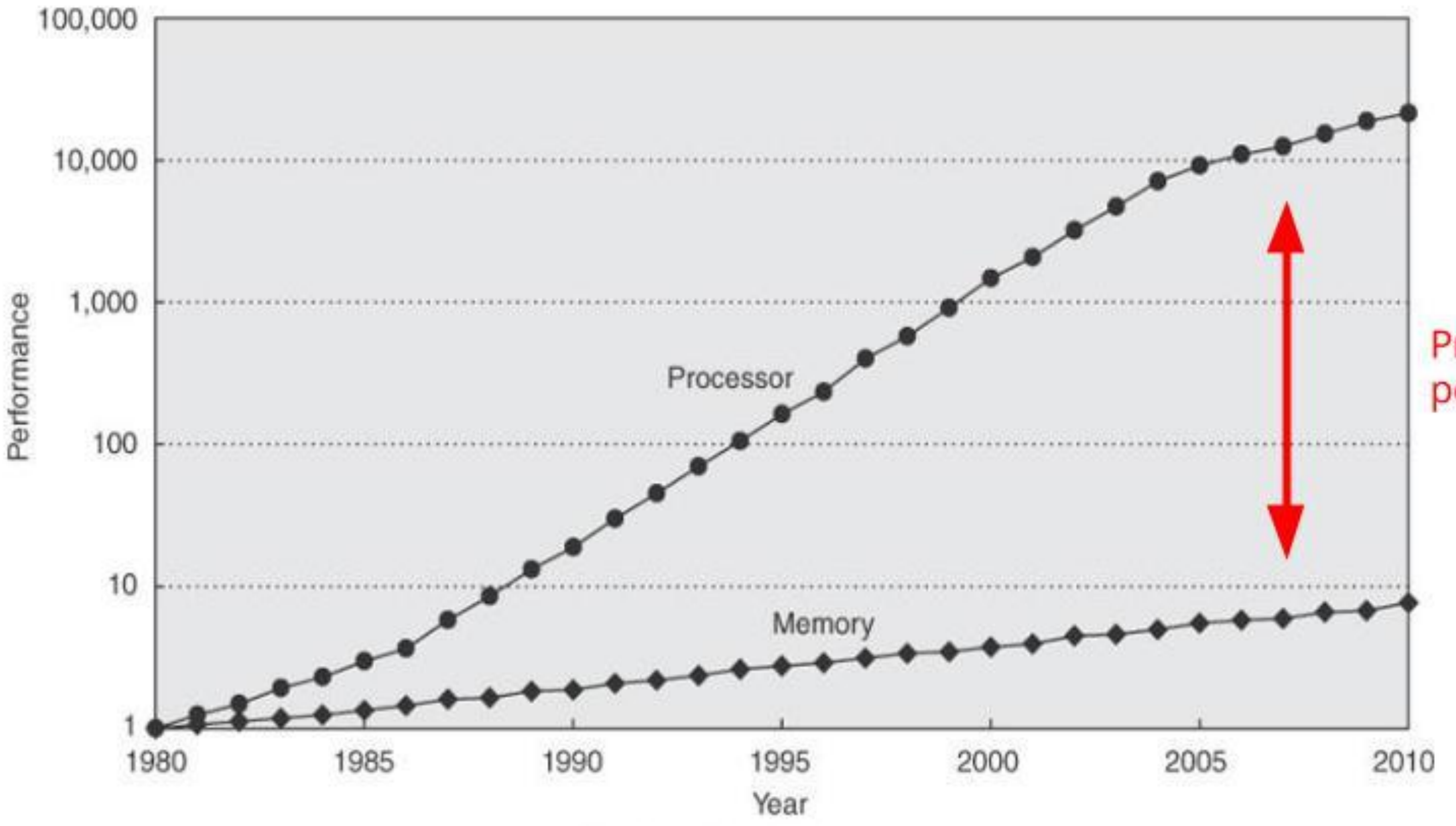
Column 0 (00)
Column 1 (01)
Column 2 (10)
Column 3 (11)
Address
Row 0 (00)
Row 1 (01)
Row 2 (10)
Row 3 (11)
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
Row selection lines
Column selection lines

Only the cell at the intersection is active.



So who is faster, the processor or the memory?

(And what’s the consequence?)



Processor vs. Memory performance gap

Memory speed lags behind CPU speed

Often, a faster processor just means that it will spend more time idle.

The principle of locality.

When a program fetches an instruction or a piece of data from memory, there is a high likelihood that:

1. The same instruction or piece of data will be accessed in the near future.
2. The instructions or data located near this piece of data will be accessed.

Cache is a refrigerator.

I/O

Direct Access
Storage Device

seek time, latency,
transfer time

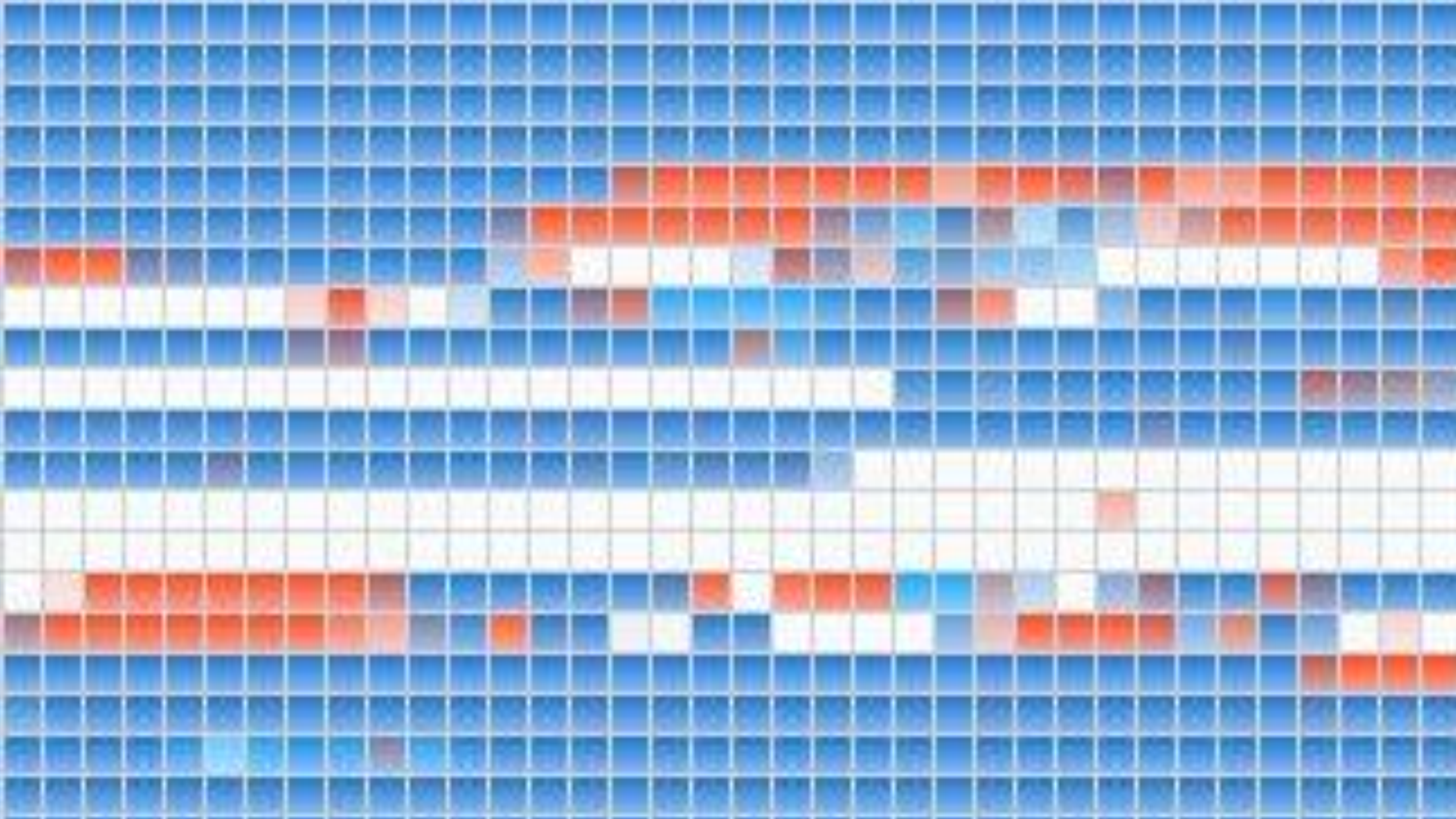
Rotation speed: 7,200 rpm

Arm movement time: 0.02 ms

Number of tracks: 1,000

Number of sectors/track: 64

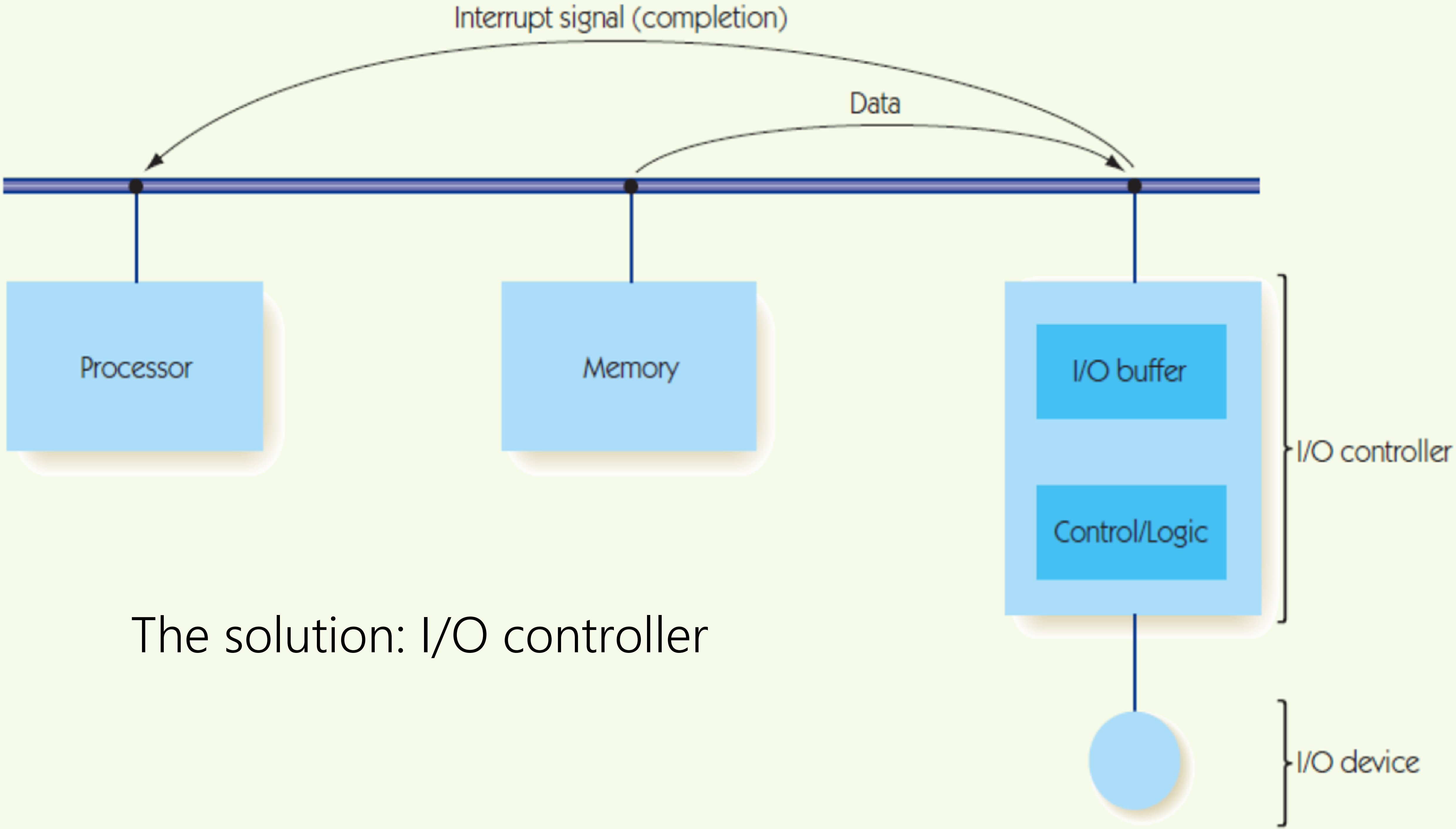
Number of bytes/sector: 1,024



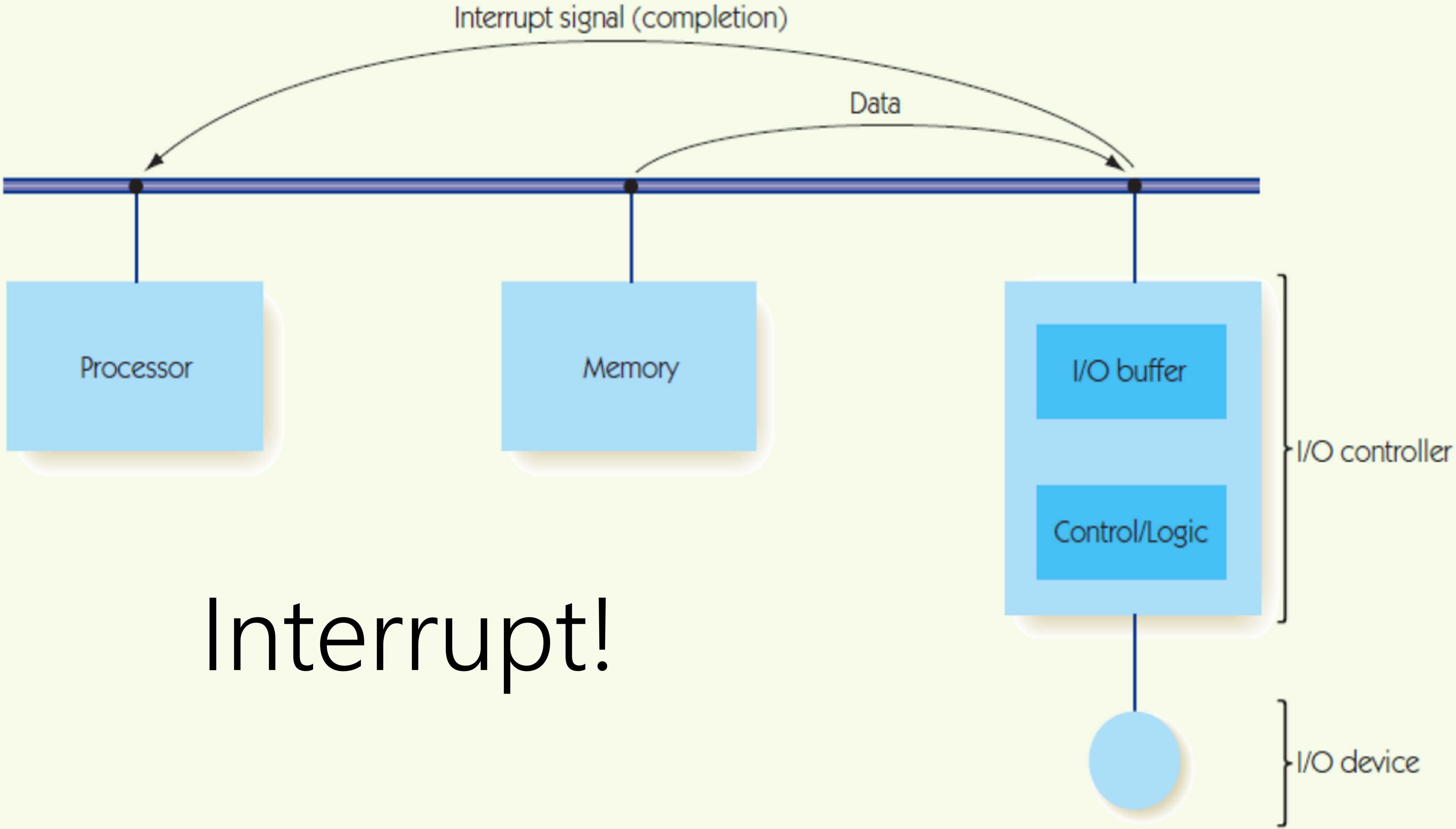
Sequential Access Storage Device

SLOOOOW

SLOOOOW



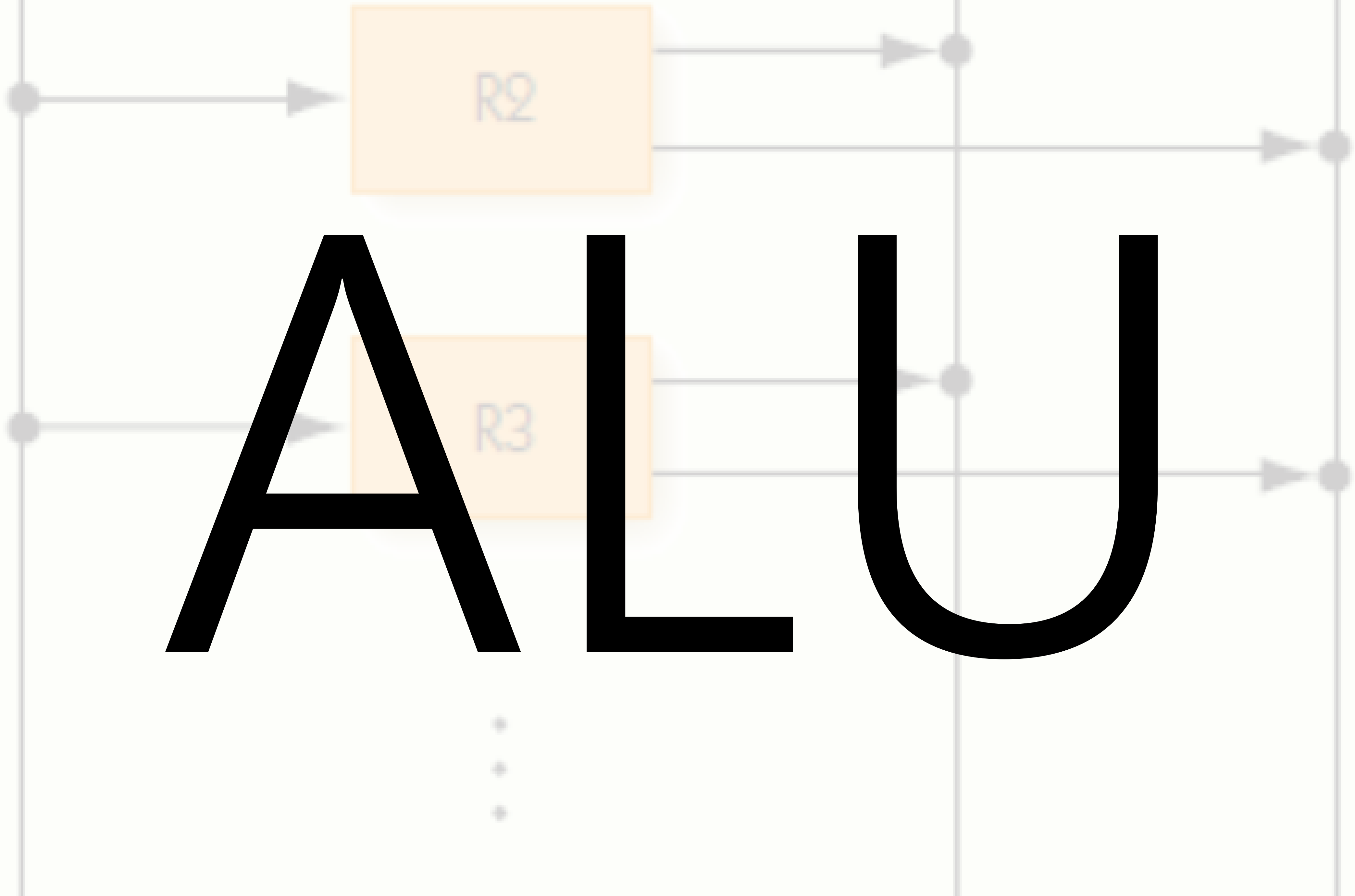
The solution: I/O controller



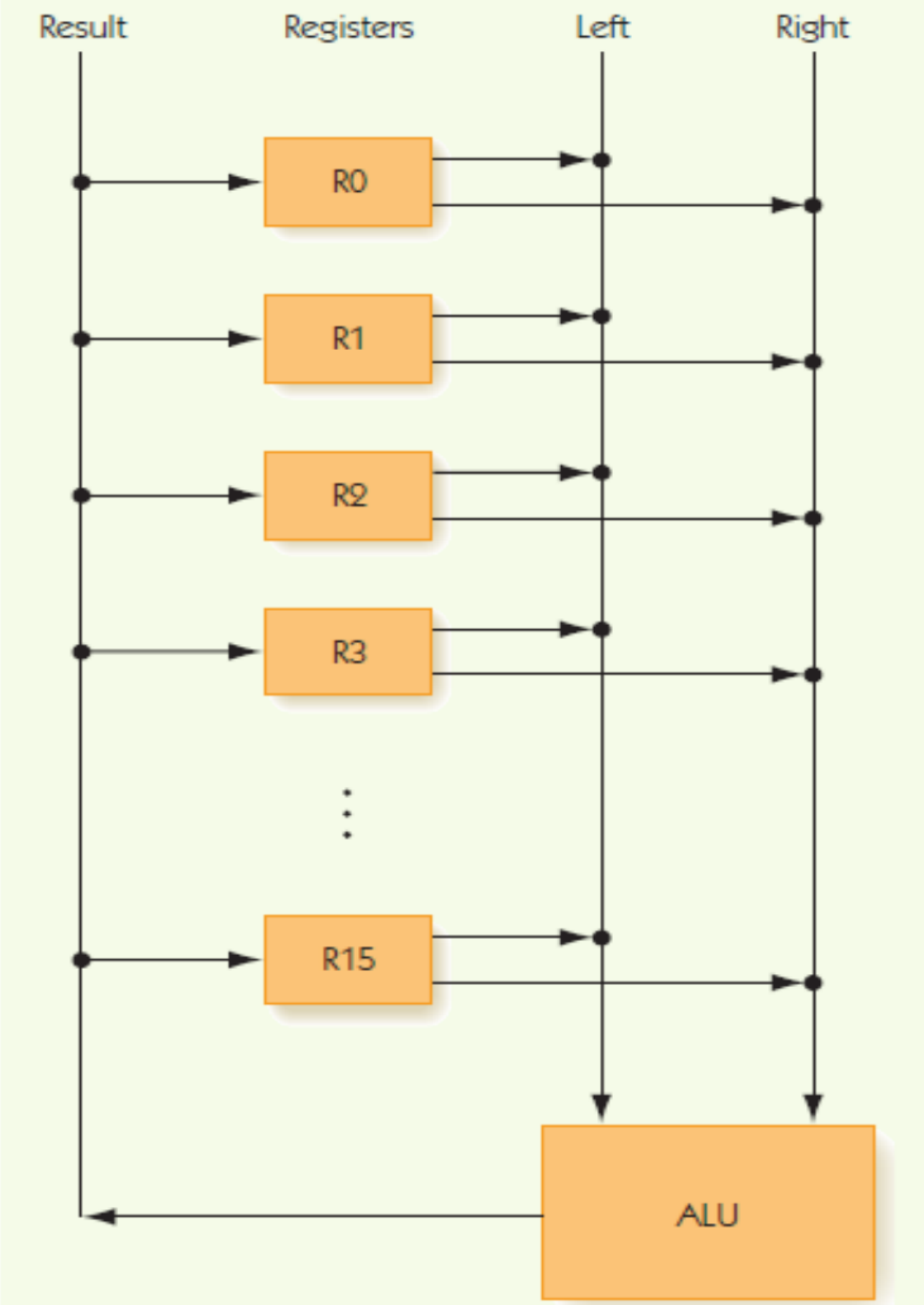
Interrupt signal (completion)
Data
Processor
Memory
I/O buffer
Control/Logic
I/O controller
I/O device

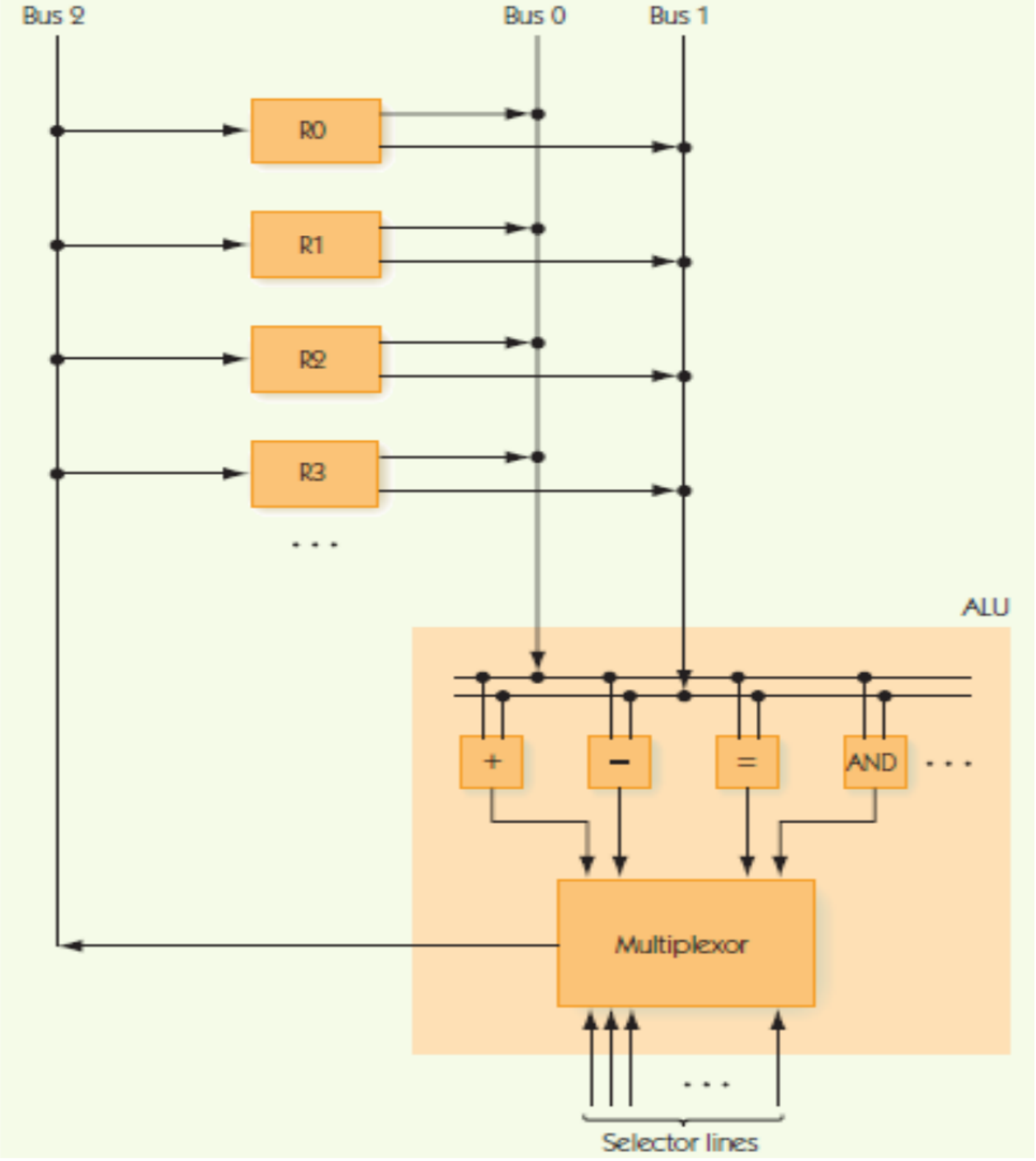
Interrupt!

ALU

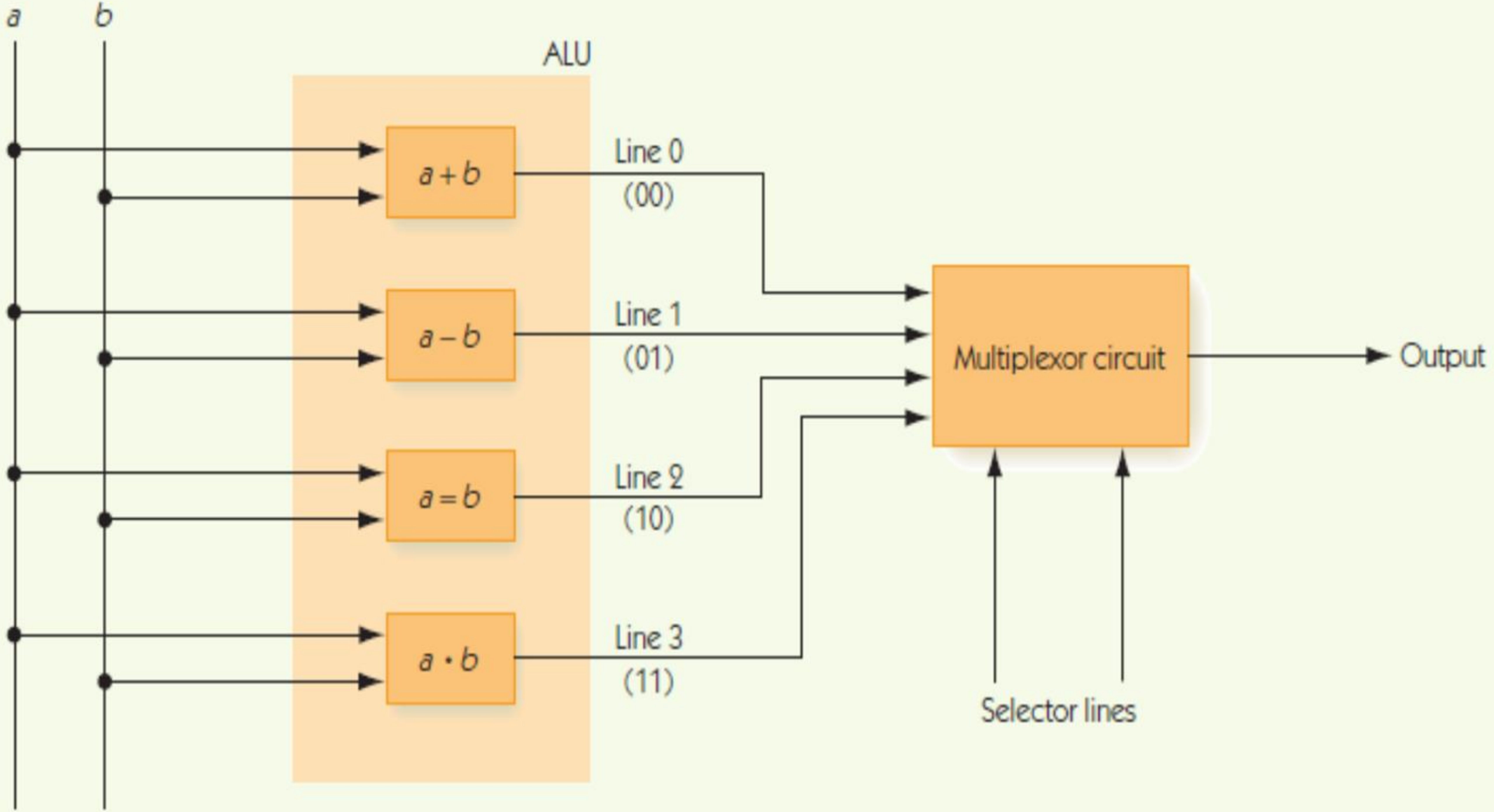


Data path



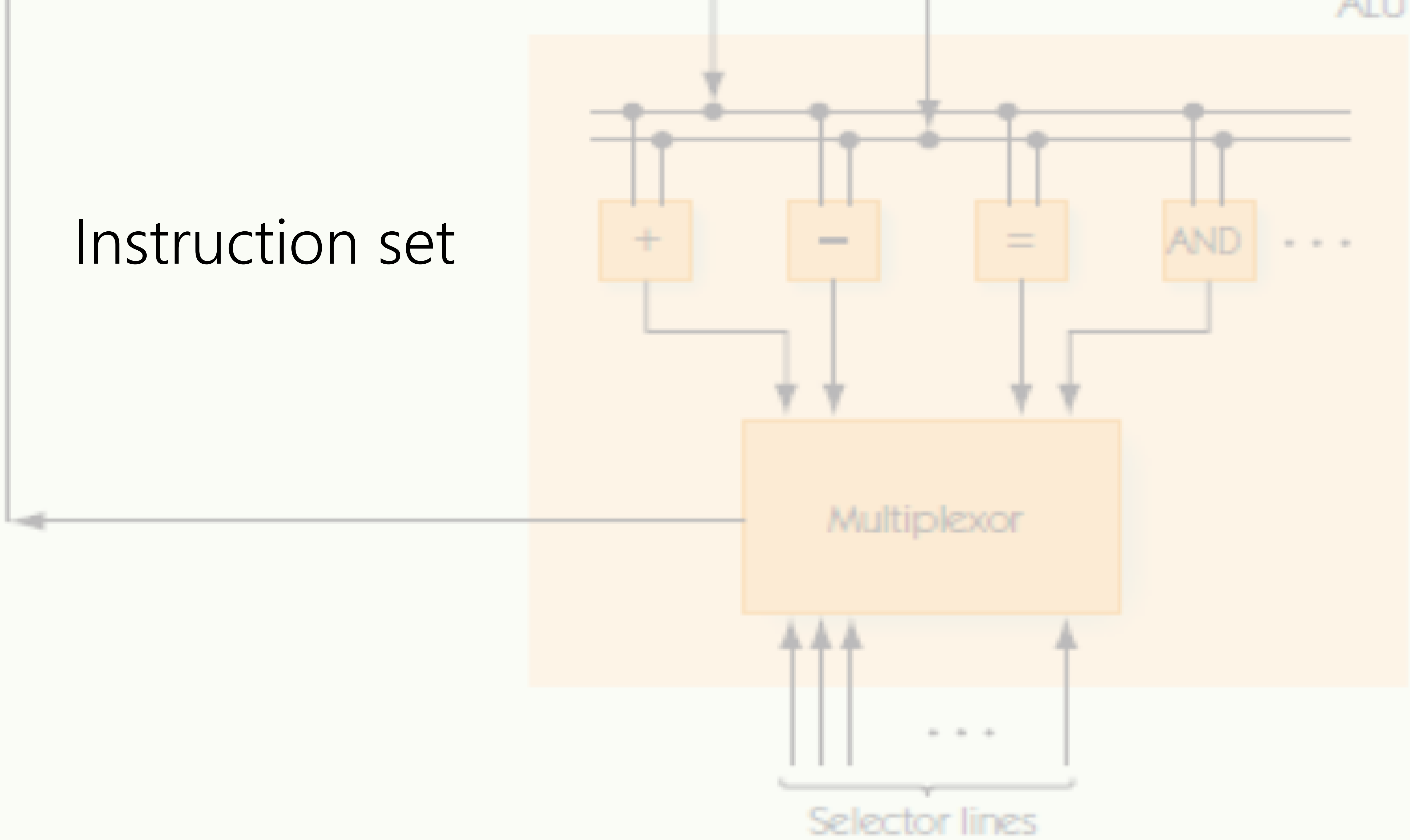


It’s a… calculator!


a
b
ALU
a + b
a − b
a = b
a · b
Line 0
(00)
Line 1
(01)
Line 2
(10)
Line 3
(11)
Multiplexor circuit
Output
Selector lines

01000101000010101010001010110

Instruction set



The power of the processor doesn’t come
from the sophistication of the instructions but rather
from the processor’s ability to execute
each instruction very very fast (in nsecs).

RISC versus CISC

So how many instructions are needed?

RISC versus CISC

So how many instructions are needed?

1. Data Transfer
2. Arithmetic
3. Compare
4. Branch

Data transfer

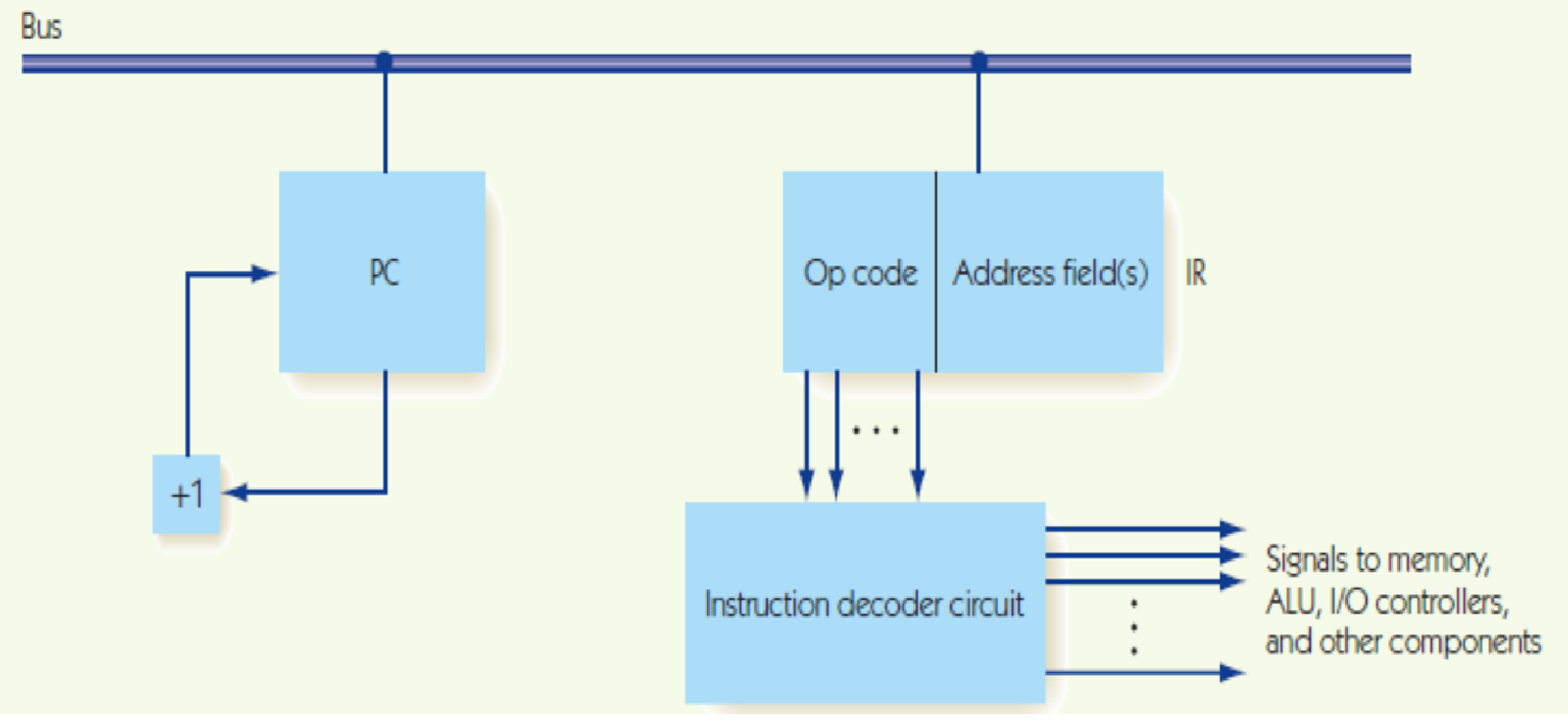
Arithmetic

Compare

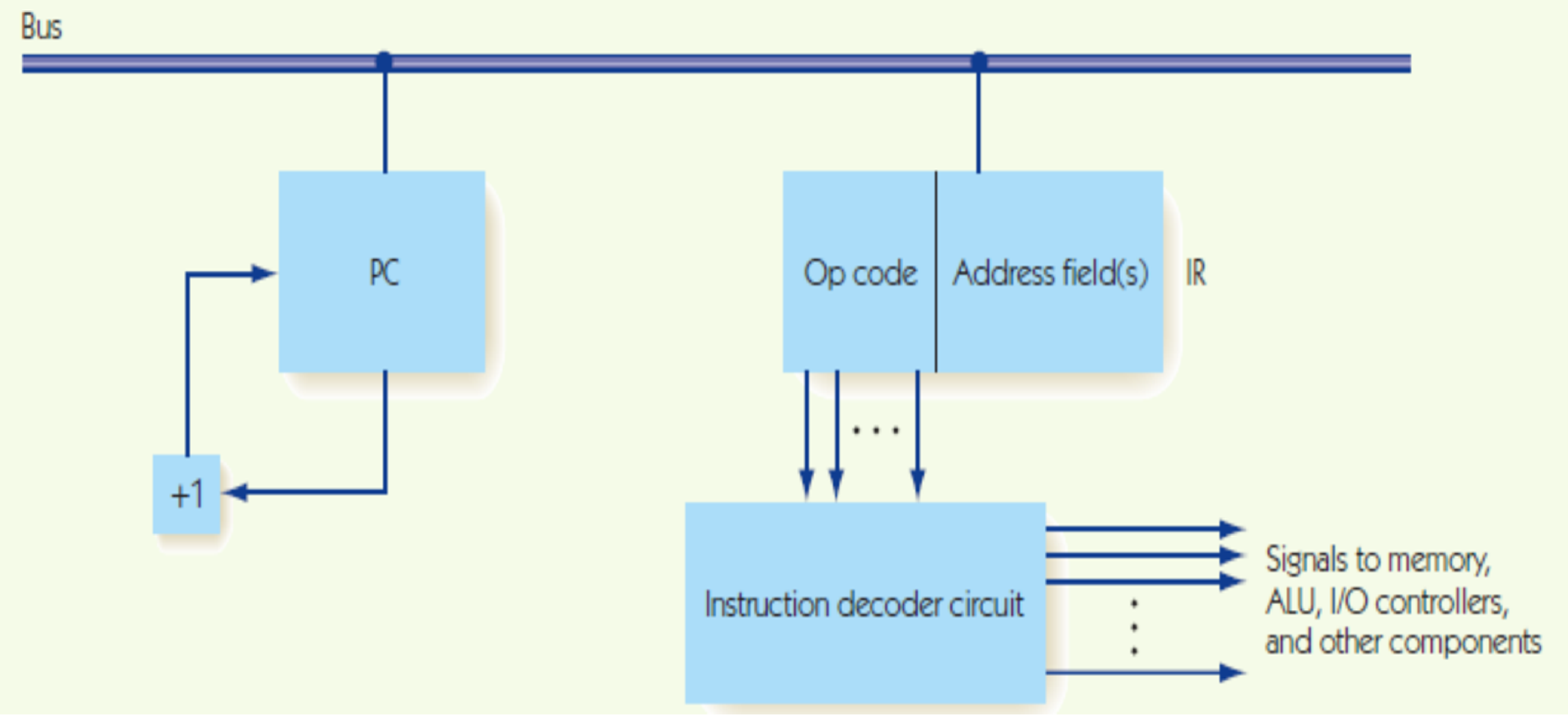
Branch

fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute
fetch, decode & execute, fetch, decode & execute, fetch, decode & execute

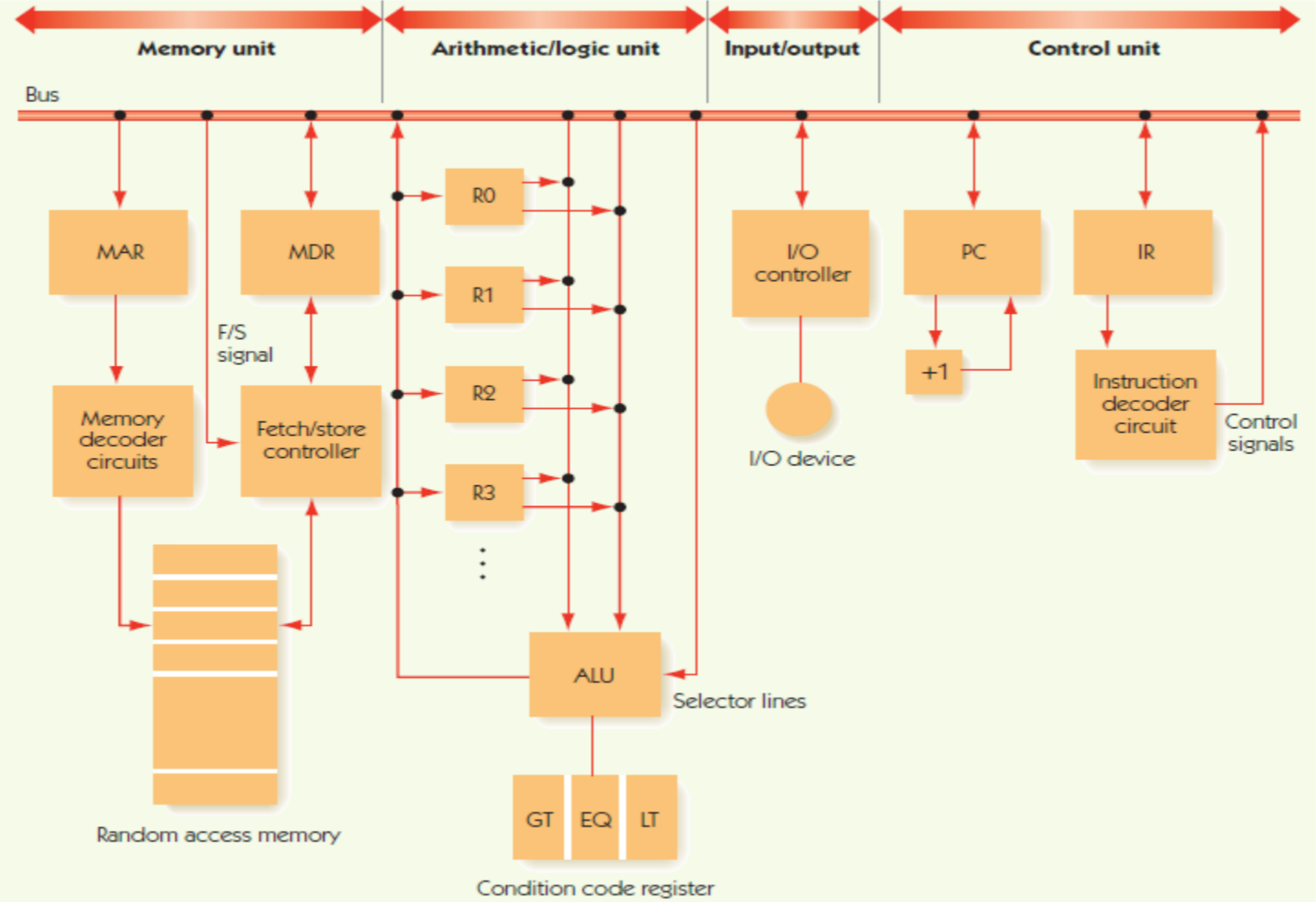
Control unit

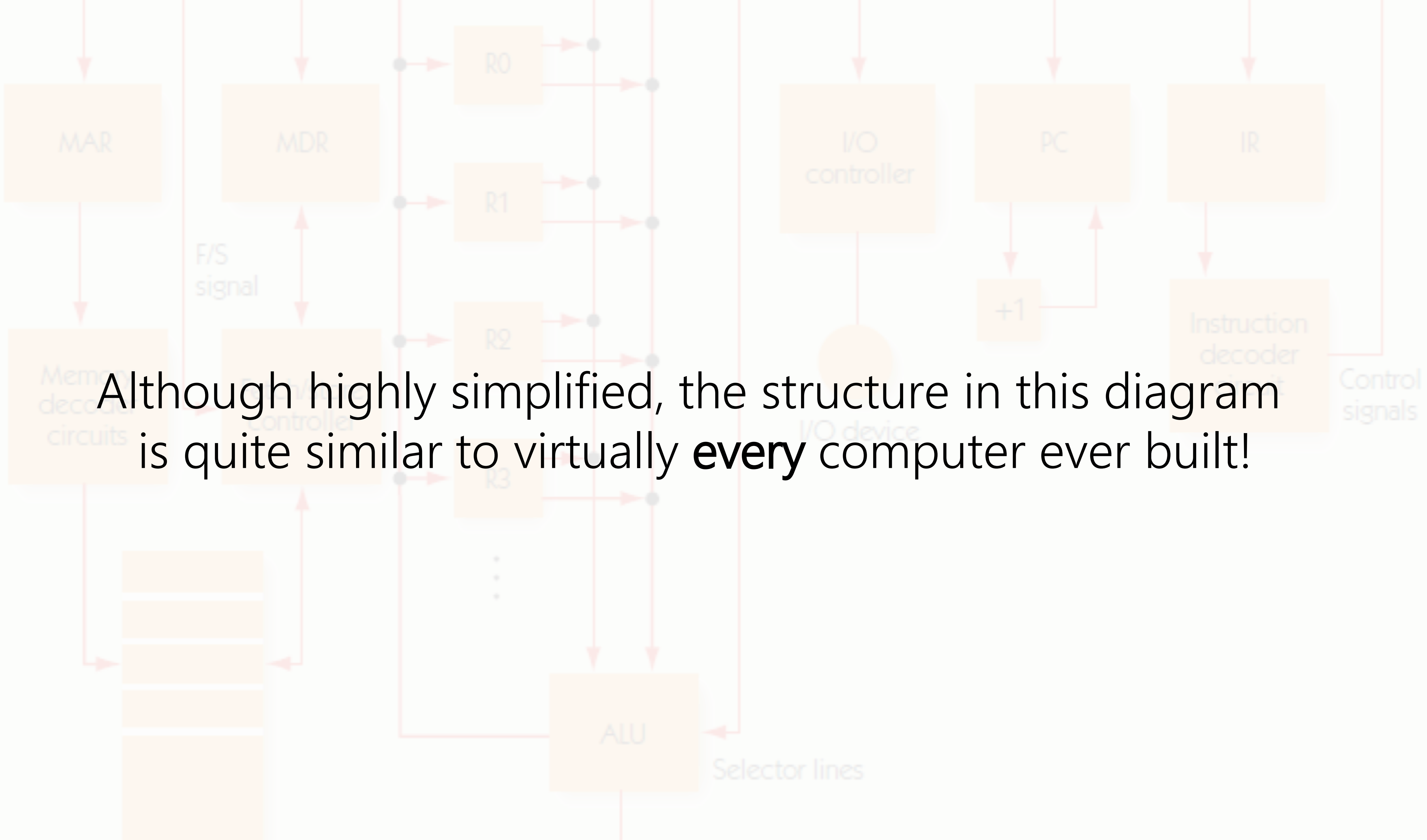


Control unit



And how does the computer look like?


Memory unit
Arithmetic/logic unit
Input/output
Control unit
Bus
MAR
MDR
F/S
signal
Memory
decoder
circuits
Fetch/store
controller
Random access memory
R0
R1
R2
R3
ALU
Selector lines
GT
EQ
LT
Condition code register
I/O
controller
I/O device
PC
+1
IR
Instruction
decoder
circuit
Control
signals



Although highly simplified, the structure in this diagram is quite similar to virtually **every** computer ever built!

And what does the computer do?

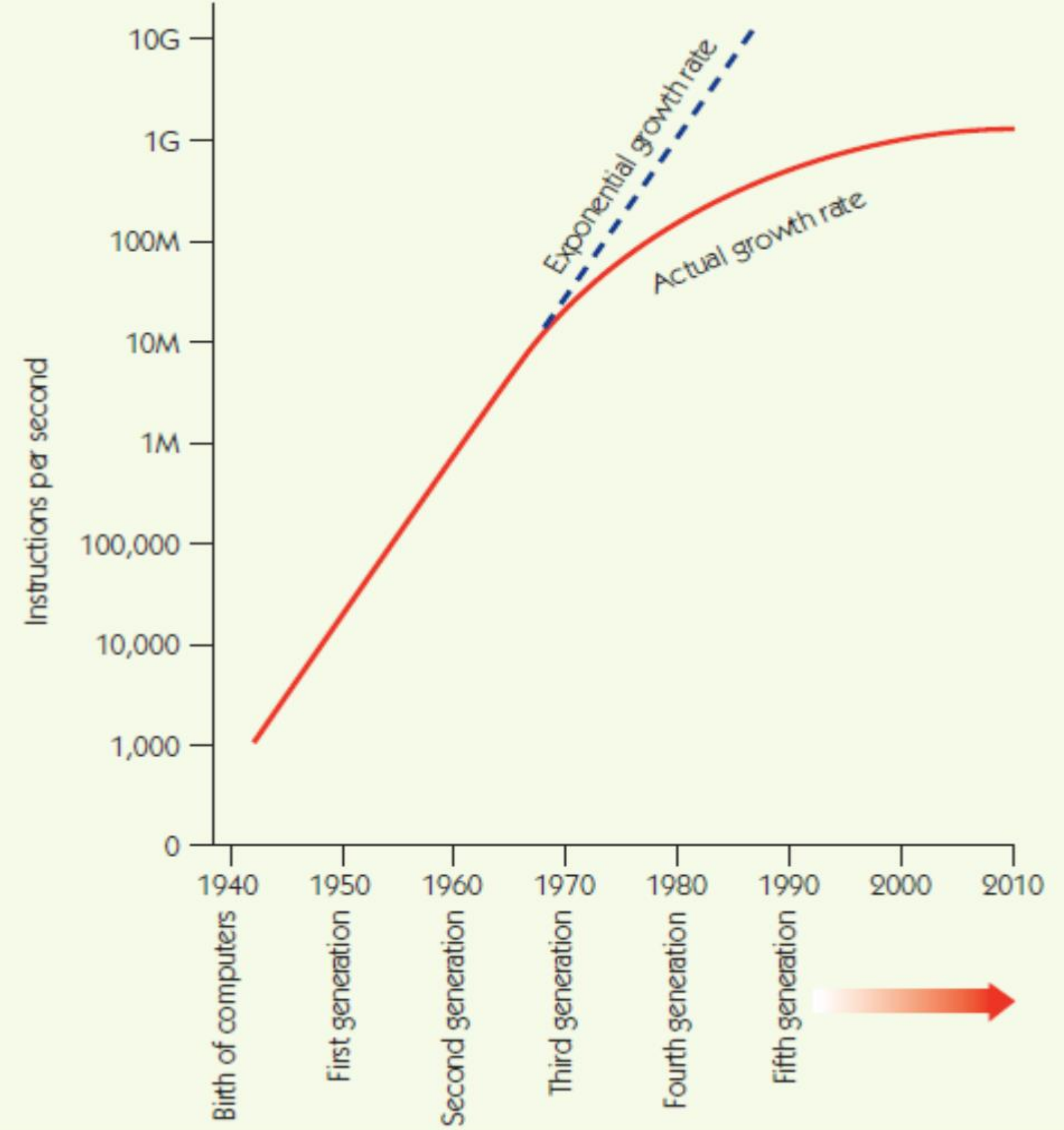
Fetch, decode & execute, fetch, decode & execute, fetch, …

Fetch phase

Decode phase

Execution phase

Remember that law of Moore's?

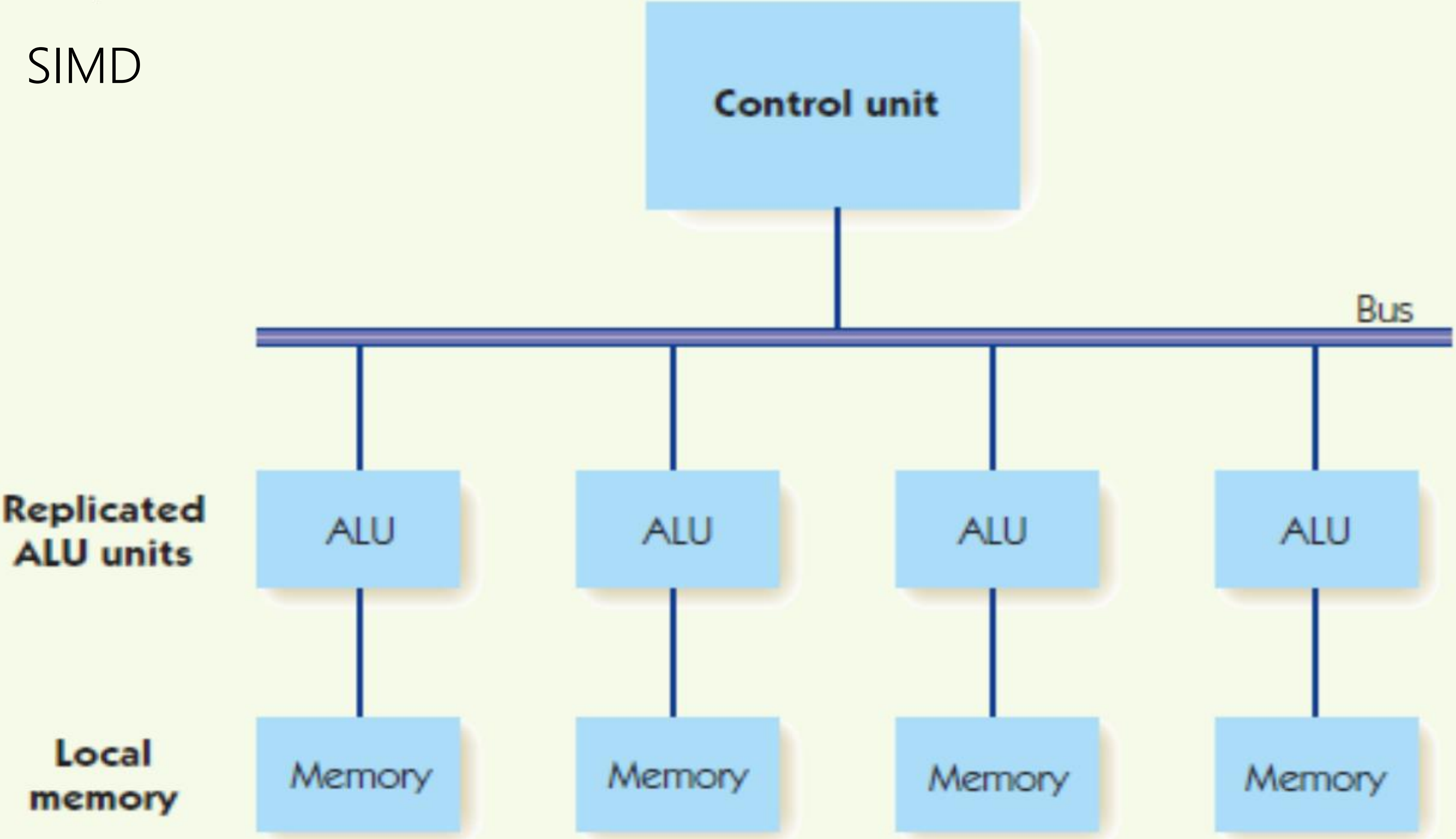


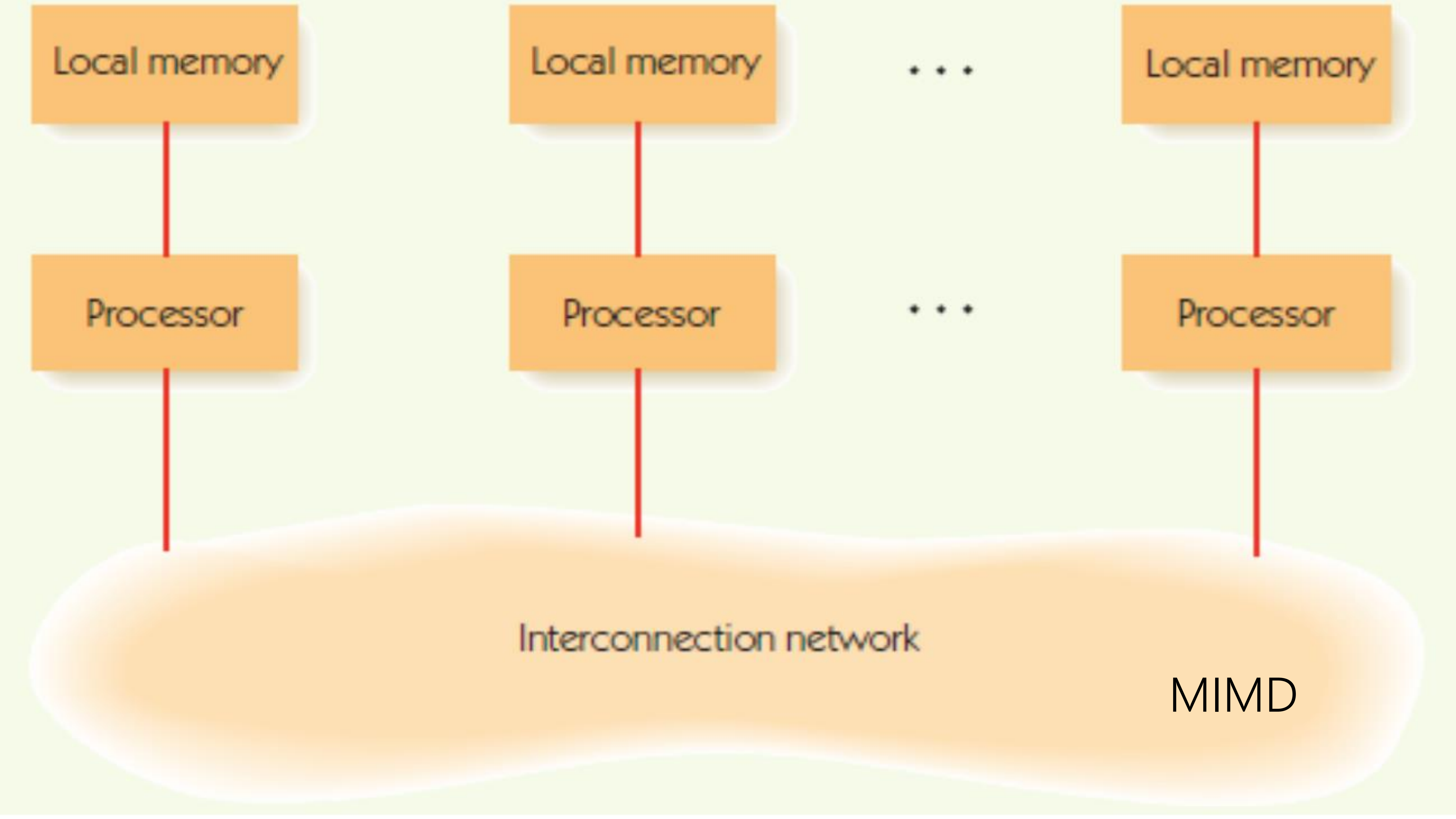
How about synchronization?

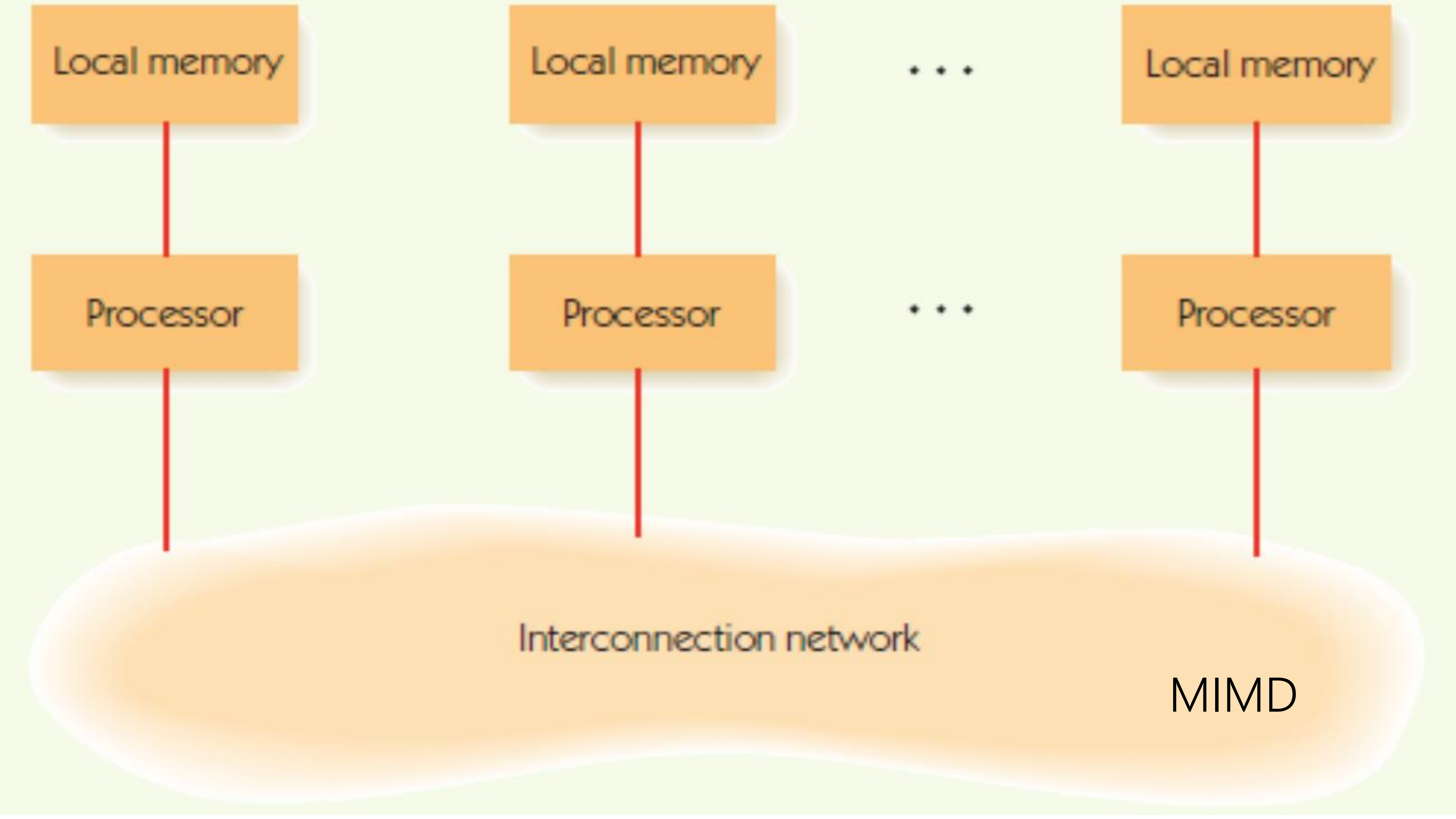
Let’s talk about speed some more…

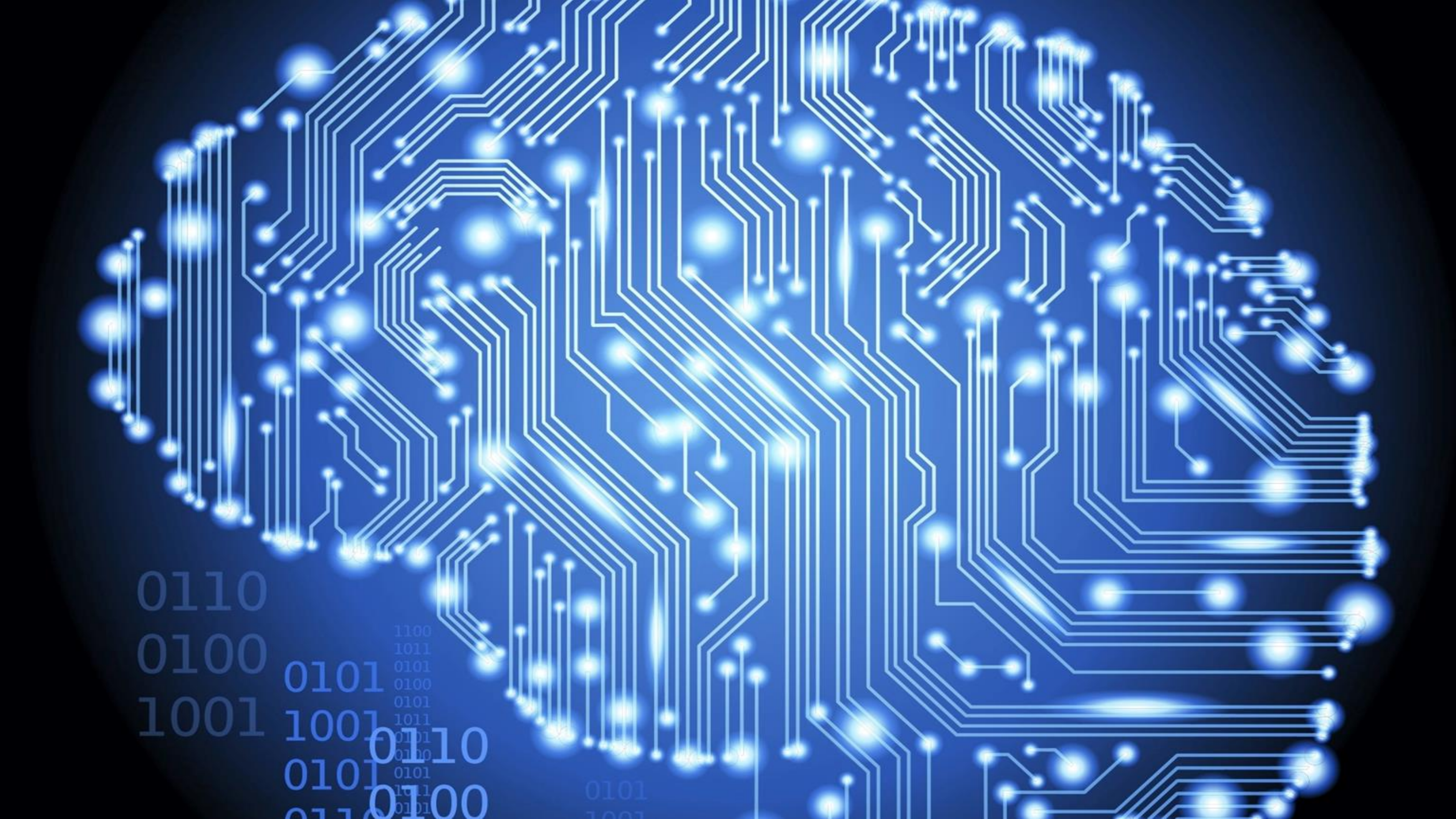
If you cannot build something to work twice as fast,
build it to do two things at once. The result will be identical.

SIMD


Control unit
Bus
Replicated ALU units
ALU
ALU
ALU
ALU
Local memory
Memory
Memory
Memory
Memory


Local memory
Local memory
Local memory
Processor
Processor
Processor
Interconnection network
MIMD


Local memory
Local memory
Local memory
Processor
Processor
Processor
Interconnection network
MIMD


0110
0100
1001
0101
1001
0101
0110
0100

Take away lessons #P6

Memory is slow.
I/O is slooooow.
Speed of light is… well, also slow.

LEGO. LEGO. LEGO.
(we now have all the HW building blocks)