Electronics Systems (938II)

Lecture 3.1

Semiconductor Memories – Basic principles

Memories

- They are another fundamental building block that is part of modern electronic systems
  - Processors need memory to work

- Most (if not almost all) memories in modern computers and electronic systems are **semiconductor memories**
  - They exist(ed) also magnetic and optical memories or storage devices, but they are actually outdated or used only in very few applications

(Semiconductor) Memories - Classification

- Access mode
  - Random Access Memory (RAM)
  - Sequential Access Memory (SAM)

- Supported operation(s)
  - Read-Only Memory (ROM)
  - Read and Write Memory (RWM)

(Semiconductor) Memories - Classification

- Data retention
  - Volatile
    - Non-permanent data
    - Data is lost after the memory is turned off
  - Non-volatile
    - Permanent data
    - Data is retained even when the memory is turned off

Semiconductor Memories - Rationale

- We have already seen an example of semiconductor memory
  - D flip-flop
    - 1x D flip-flop = 1 memory bit
    - It is an example of memory which is volatile and RWM

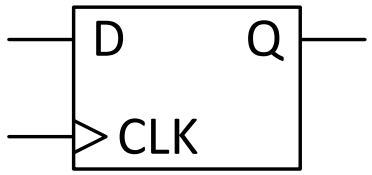
- D flip-flops are used as memory in computers, but they are expensive in terms of cost, cost per bit, …
  - There are memories based on semiconductor technologies (such as CMOS) that achieve lower cost per bit or, in other words, higher density per silicon area

Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**

Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of

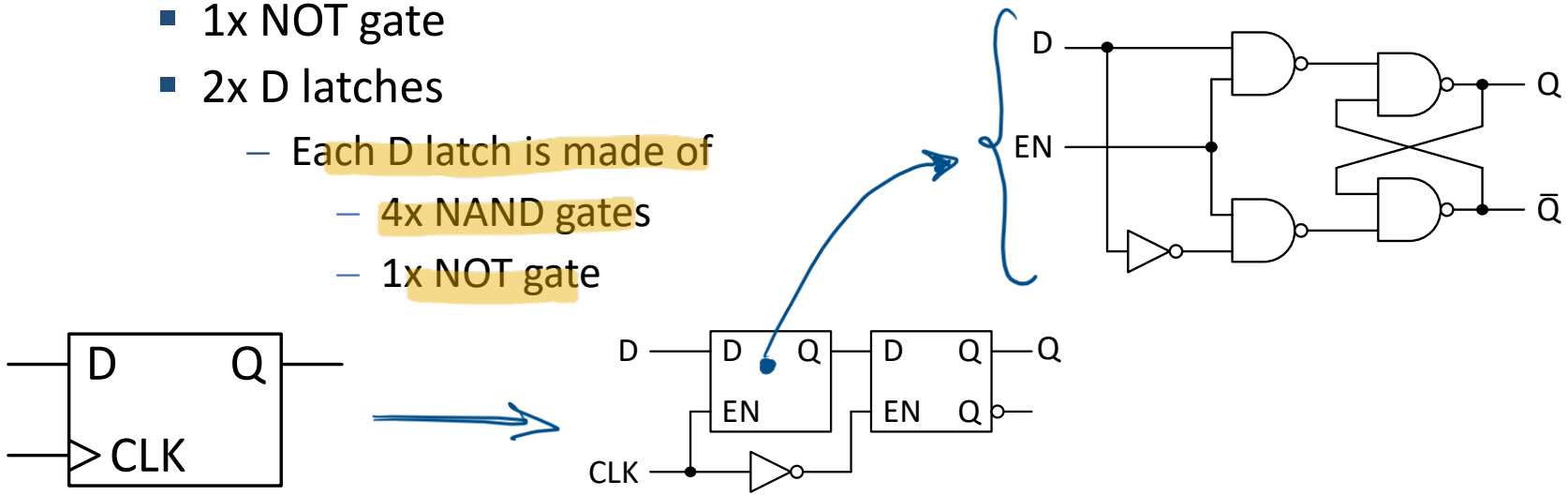


Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of
    - 1x NOT gate
    - 2x D latches

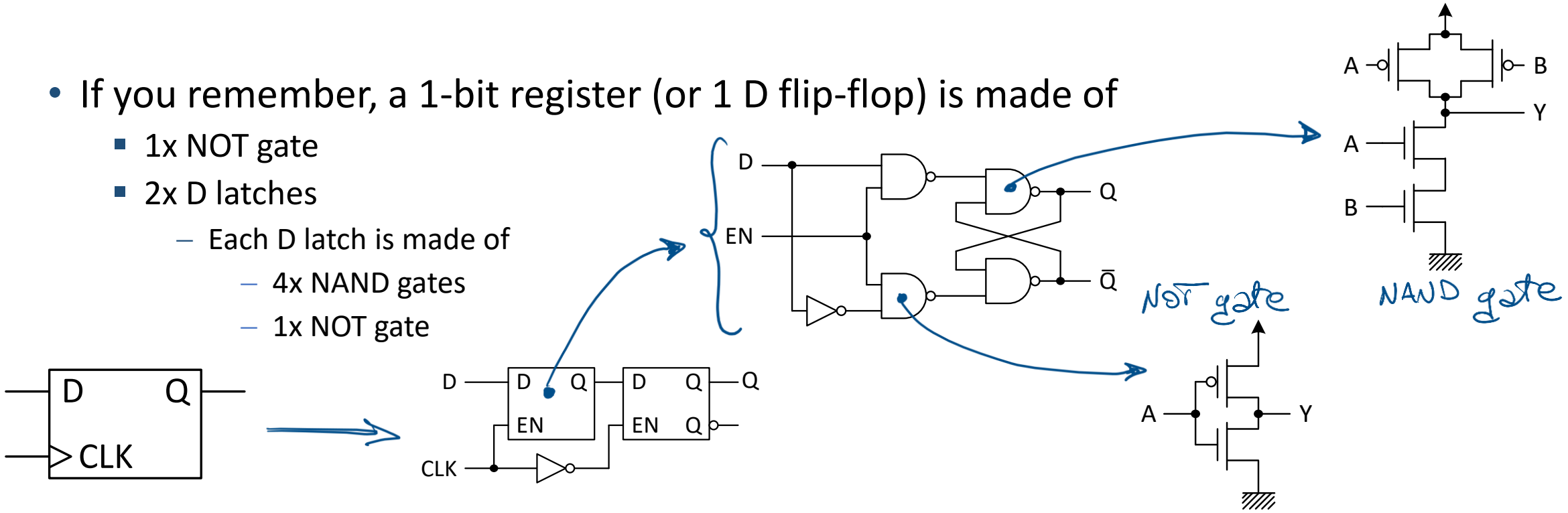
Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of
    - 1x NOT gate
    - 2x D latches
      - Each D latch is made of
        - 4x NAND gates
        - 1x NOT gate



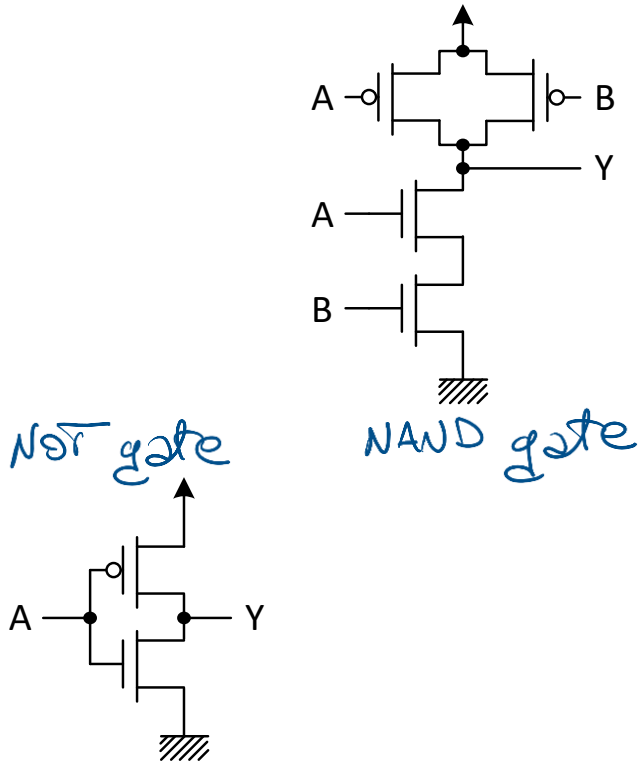
Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of
    - 1x NOT gate
    - 2x D latches
      - Each D latch is made of
        - 4x NAND gates
        - 1x NOT gate



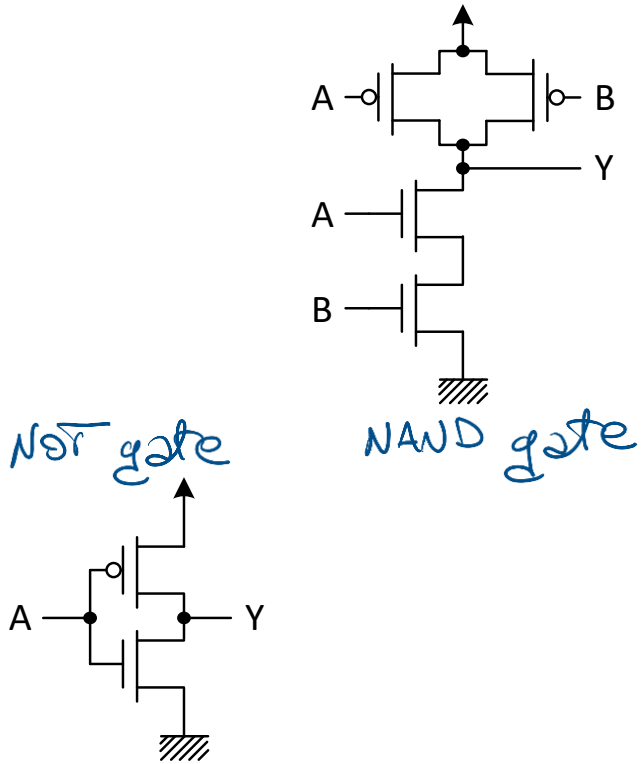
Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of
    - 1x NOT gate
    - 2x D latches
      - Each D latch is made of
        - 4x NAND gates → 4x (4 transistors) = 16 transistors
        - 1x NOT gate → 1x (2 transistors) = 2 transistors



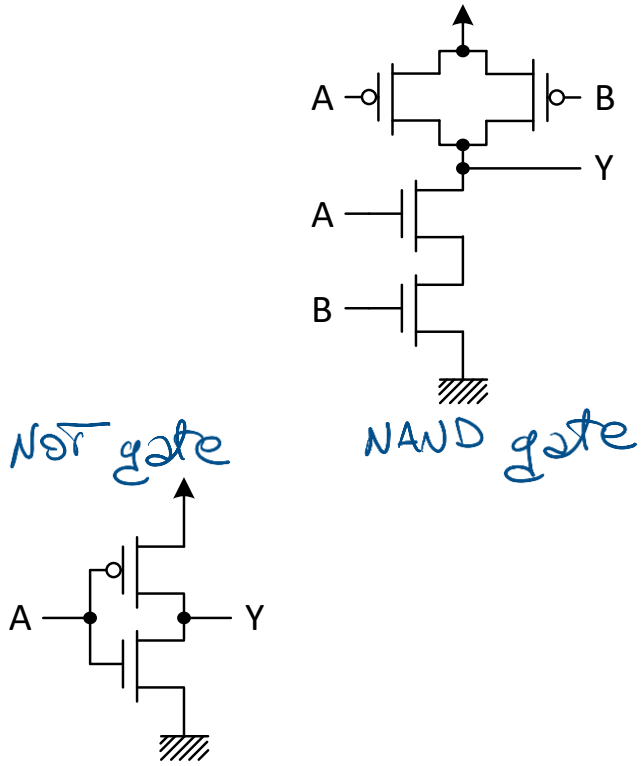
Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of
    - 1x NOT gate
    - 2x D latches
      - Each D latch is made of → 18 transistor
        - 4x NAND gates → 4x (4 transistors) = 16 transistors
        - 1x NOT gate → 1x (2 transistors) = 2 transistors



Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of
    - 1x NOT gate → 1x (2 transistors) = 2 transistors
    - 2x D latches → 2x (18 transistors) = 36 transistors
      - Each D latch is made of → 18 transistor
        - 4x NAND gates → 4x (4 transistors) = 16 transistors
        - 1x NOT gate → 1x (2 transistors) = 2 transistors



Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of **38 transistors**!
    - 1x NOT gate → 1x (2 transistors) = 2 transistors
    - 2x D latches → 2x (18 transistors) = 36 transistors
      - Each D latch is made of → 18 transistor
        - 4x NAND gates → 4x (4 transistors) = 16 transistors
        - 1x NOT gate → 1x (2 transistors) = 2 transistors

Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of **38 transistors**!
- This was an academic example
  - A real-world CMOS D flip-flop consists of fewer transistors
  - However, looking at a real-world CMOS technology (7 nm) …

Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of **38 transistors**!
- This was an academic example
  - A real-world CMOS D flip-flop consists of fewer transistors
  - However, looking at a real-world CMOS technology (7 nm) …
    - Area of a NAND gate $\cong 0.0768\ \mu^2$

Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of **38 transistors**!
- This was an academic example
  - A real-world CMOS D flip-flop consists of fewer transistors
  - However, looking at a real-world CMOS technology (7 nm) …
    - Area of a NAND gate $\cong 0.0768\ \mu^2$ → area of 1 transistor $\cong \frac{0.0768}{4}\ \mu^2 = 0.0192\ \mu^2$

Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of **38 transistors**!
- This was an academic example
  - A real-world CMOS D flip-flop consists of fewer transistors
  - However, looking at a real-world CMOS technology (7 nm) …
    - Area of a NAND gate $\cong 0.0768\ \mu^2$ → area of 1 transistor $\cong \frac{0.0768}{4}\ \mu^2 = 0.0192\ \mu^2$
    - Area of a DFF $\cong 0.4032\ \mu^2$

Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of **38 transistors**!
- This was an academic example
  - A real-world CMOS D flip-flop consists of fewer transistors
  - However, looking at a real-world CMOS technology (7 nm) …
    - Area of a NAND gate $\cong 0.0768\ \mu^2$ → area of 1 transistor $\cong \frac{0.0768}{4}\ \mu^2 = 0.0192\ \mu^2$
    - Area of a DFF $\cong 0.4032\ \mu^2 \rightarrow \frac{0.4032}{0.0192} = 21$

Semiconductor Memories - Rationale

- These specialized memory technologies enable a memory cell (1 bit) to be made at a cost of about **1 transistor**
  - If you remember, a 1-bit register (or 1 D flip-flop) is made of **38 transistors**!
- This was an academic example
  - A real-world CMOS D flip-flop consists of fewer transistors
  - However, looking at a real-world CMOS technology (7 nm) …
    - Area of a NAND gate $\cong 0.0768\ \mu^2$ → area of 1 transistor $\cong \frac{0.0768}{4}\ \mu^2 = 0.0192\ \mu^2$
    - Area of a DFF $\cong 0.4032\ \mu^2 \rightarrow \frac{0.4032}{0.0192} = 21 \rightarrow$ **One DFF is about 21 transistors**

Semiconductor Memories - Rationale

- We are going to see different types of semiconductor memories
  - All the main ones used in modern electronic systems

Semiconductor Memories - Rationale

- We are going to see different types of semiconductor memories
  - All the main ones used in modern electronic systems

- But first, let's take a look at the outline of the architecture of semiconductor memories
  - It is common to all of them (Read only / RW ... it's the same)

Semiconductor Memories – Architecture outline

- The architecture of semiconductor memories has a **matrix** shape
  - Two dimensions
  - **Bi-dimensional addressing**

Semiconductor Memories – Architecture outline

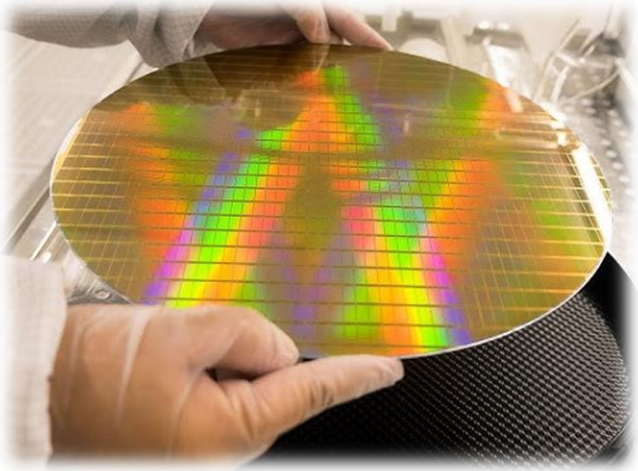
- The architecture of semiconductor memories has a **matrix** shape
  - Two dimensions
  - **Bi-dimensional addressing**

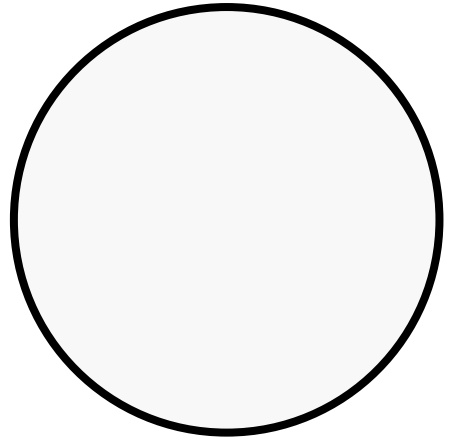
- Why not a vector shape? Lower costs mainly
  - For two main reason

Semiconductor Memories – Architecture outline

1. Compatibility with manufacturing process

   - Semiconductor systems/devices are fabricated by processing a circular die of silicon (typically called wafer)

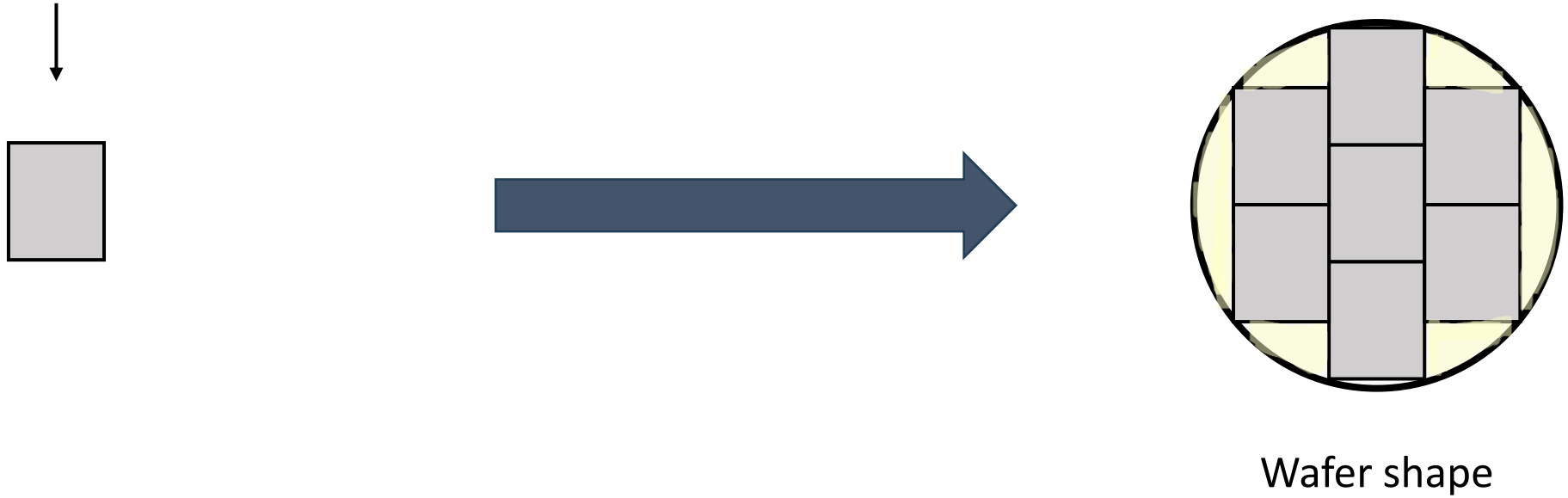




Wafer shape

Semiconductor Memories – Architecture outline

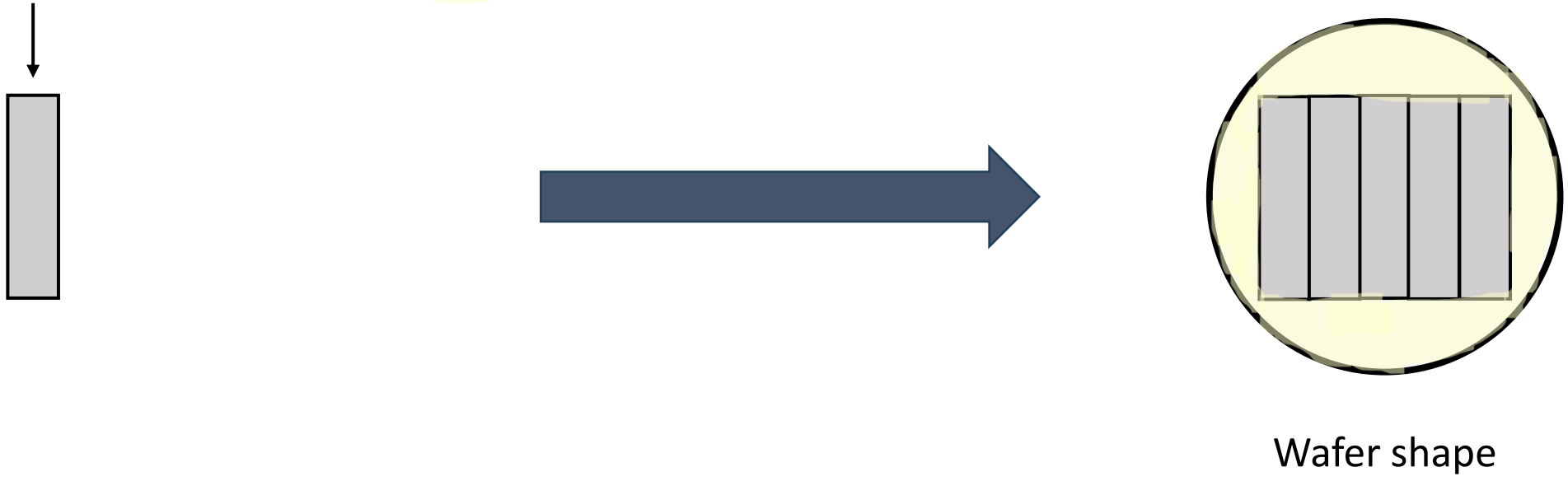
1. Compatibility with manufacturing process
   - Semiconductor systems/devices are fabricated by processing a circular die of silicon (typically called wafer)
   - Using a **matrix shape**, some parts of the wafer are wasted



Wafer shape

Semiconductor Memories – Architecture outline

1. Compatibility with manufacturing process
   - Semiconductor systems/devices are fabricated by processing a circular die of silicon (typically called wafer)
   - If a **vector shape** is used, more parts of the wafer are wasted



Wafer shape

Semiconductor Memories – Architecture outline

2. Resource cost Monodimensional addressing!
   - Suppose that L bits of address are required
     - Vector shape
       - Mono-dimensional addressing → 2^L addresses
     - Matrix shape
       - Bi-dimensional addressing
       - Assuming to split L as $L = N + M$
         - 2^N addresses for one dimension
         - 2^M addresses for the other dimension
       - Total: $2^N + 2^M$ addresses Less addresses.

Semiconductor Memories – Architecture outline

2. Resource cost
   - Suppose that L bits of address are required
     - Vector shape
       - Resource cost = logic resources to decode $\mathbf{2^L}$ addresses
     - Matrix shape
       - Resource cost = logic resources to decode $\mathbf{2^N + 2^M}$ addresses
     - With $L = N + M \rightarrow 2^L = 2^{N+M} = 2^N \cdot 2^M$
     - $2^N \cdot 2^M > 2^N + 2^M$???

Semiconductor Memories – Architecture outline

2. Resource cost
   - Suppose that L bits of address are required: $L = N + M$
     - $2^N \cdot 2^M > 2^N + 2^M$

Semiconductor Memories – Architecture outline

2. Resource cost
   - Suppose that L bits of address are required: $L = N + M$
     - $2^N \cdot 2^M > 2^N + 2^M$

Semiconductor Memories – Architecture outline

2. Resource cost
   - Suppose that L bits of address are required: $L = N + M$
     - $2^N \cdot 2^M > 2^N + 2^M$
     - $2^N \cdot (2^M - 1) > 2^M$

Semiconductor Memories – Architecture outline

2. Resource cost
   - Suppose that L bits of address are required: $L = N + M$
     - $2^N \cdot 2^M > 2^N + 2^M$
     - $2^N \cdot (2^M - 1) > 2^M$

Semiconductor Memories – Architecture outline

2. Resource cost
   - Suppose that L bits of address are required: $L = N + M$
     - $2^N \cdot 2^M > 2^N + 2^M$
     - $2^N \cdot (2^M - 1) > 2^M$
     - $2^N > \frac{2^M}{(2^M - 1)}$

Semiconductor Memories – Architecture outline

2. Resource cost

- Suppose that L bits of address are required: $L = N + M$
  - $2^N > \frac{2^M}{(2^M-1)}$
  - Some numeric examples

Non realistic examples

– $N = M = 1$	$\rightarrow 2 > \frac{2}{2-1}$	$\rightarrow 2 > \frac{2}{1}$	$\rightarrow 2 = 2$	(NO)
– $N = 1, M = 2$	$\rightarrow 2 > \frac{4}{4-1}$	$\rightarrow 2 > \frac{4}{3}$	$\rightarrow 2 > 1.3333$	(YES)
– $N = 2, M = 1$	$\rightarrow 4 > \frac{2}{2-1}$	$\rightarrow 4 > \frac{2}{1}$	$\rightarrow 4 > 2$	(YES)
– $N = 3, M = 7$	$\rightarrow 8 > \frac{128}{128-1}$	$\rightarrow 8 > \frac{128}{127}$	$\rightarrow 8 > 1.008$	(YES)
– …				

In theory not always satisfied, in reality it is always.

Semiconductor Memories – Architecture outline

2. Resource cost

- Suppose that L bits of address are required: $L = N + M$
  - $2^N > \frac{2^M}{(2^M - 1)}$
  - A more realistic example:
    - 1 Mbit of memory $\cong 2^{20} \rightarrow L = 20$
    - Assume $N = 8$ and $M = 12$
    - $256 > \frac{4096}{4096-1}$ $\rightarrow$ $256 > \frac{4096}{4095}$ $\rightarrow$ $256 >> \sim 1$

Cost to decode addresses is much higher if we don't. You would need more silicon

Semiconductor Memories – Architecture outline

2. Resource cost

- Suppose that L bits of address are required: $L = N + M$
  - $2^N > \frac{2^M}{(2^M - 1)}$
  - A more realistic example:
    - 1 Mbit of memory $\cong 2^{20} \rightarrow L = 20$
    - Assume $N = 8$ and $M = 12$
    - $256 > \frac{4096}{4096-1} \quad \rightarrow \quad 256 > \frac{4096}{4095} \quad \rightarrow \quad 256 >> \sim 1$
  - In other words, assuming realistic examples, it happens that $\mathbf{2^L >> 2^N + 2^M}$!!!

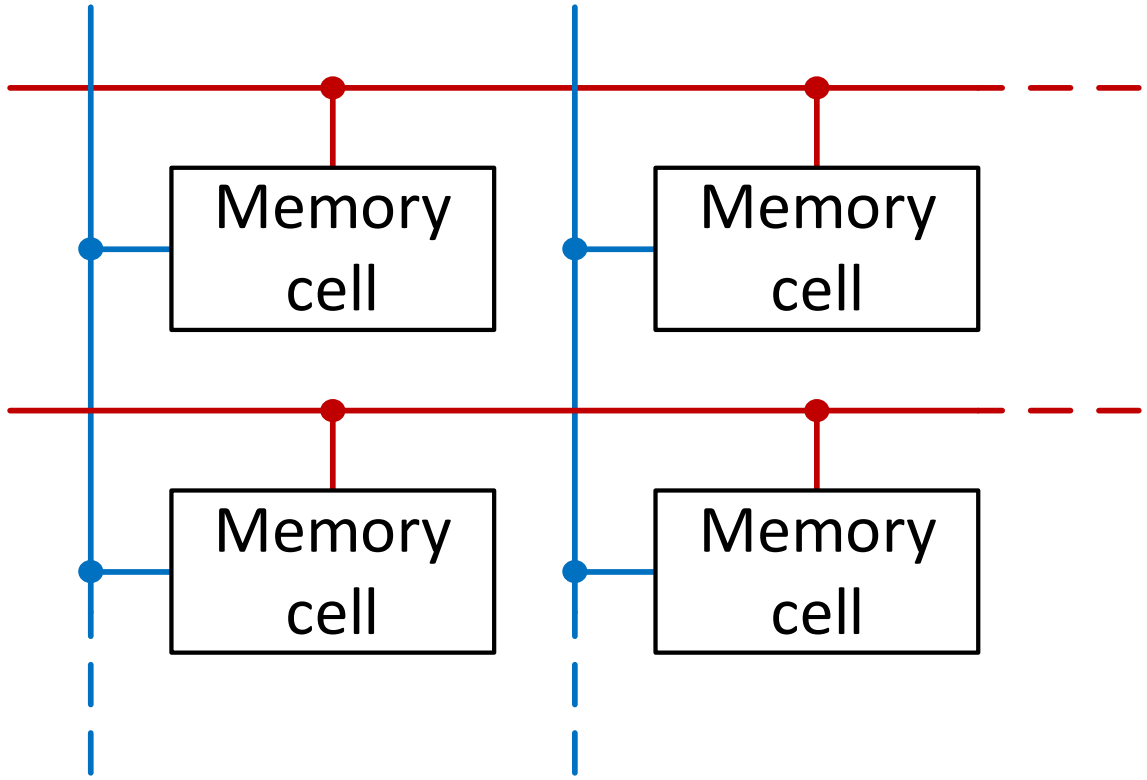
Semiconductor Memories – Architecture outline

2. Resource cost

- Suppose that L bits of address are required: $L = N + M$
  - $2^N > \frac{2^M}{(2^M-1)}$
  - A more realistic example:
    - 1 Mbit of memory $\cong 2^{20} \rightarrow L = 20$
    - Assume $N = 8$ and $M = 12$
    - $256 > \frac{4096}{4096-1} \quad \rightarrow \quad 256 > \frac{4096}{4095} \quad \rightarrow \quad 256 >> \sim 1$
  - In other words, assuming realistic examples, it happens that $\mathbf{2^L >> 2^N + 2^M}$!!!
  - **Bi-dimensional addressing requires far fewer resources for address decoding** !!!

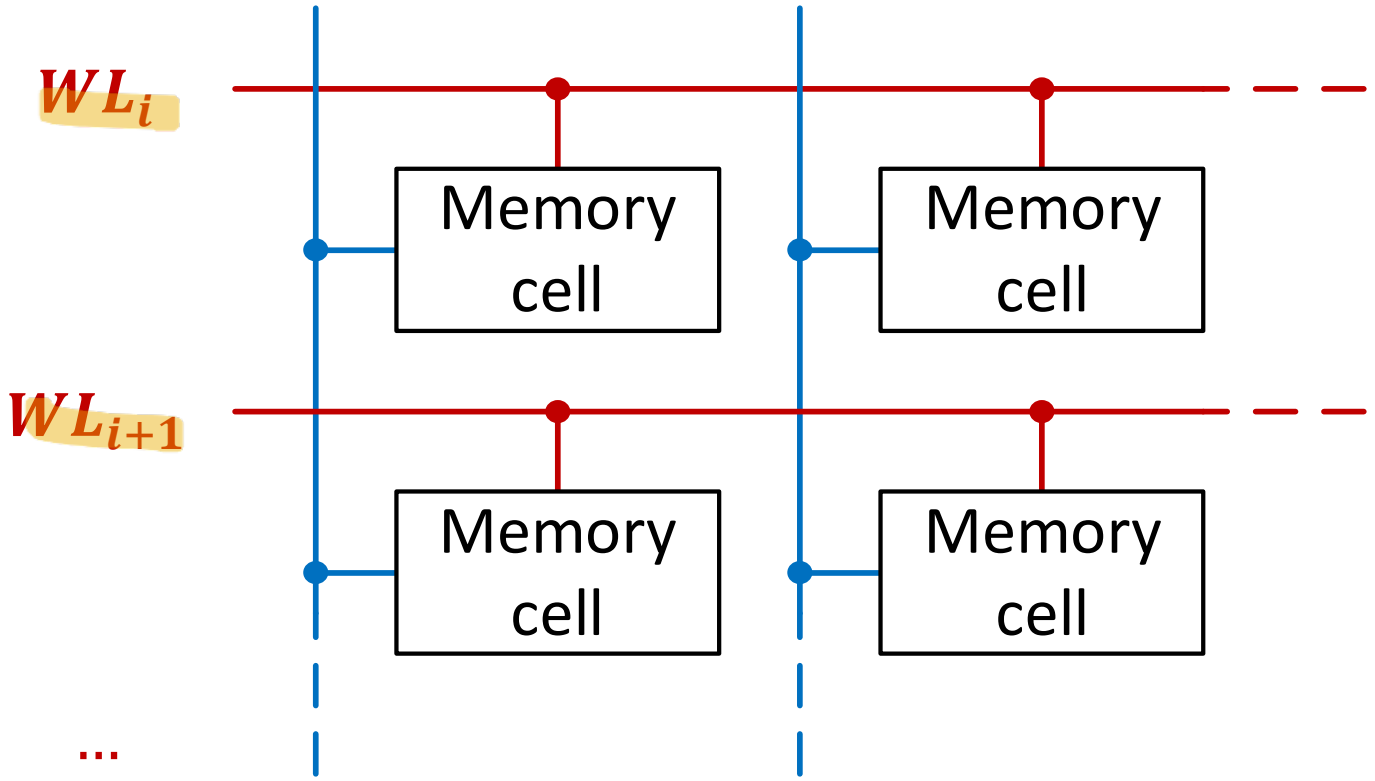
Semiconductor Memories – Architecture outline

- Draft



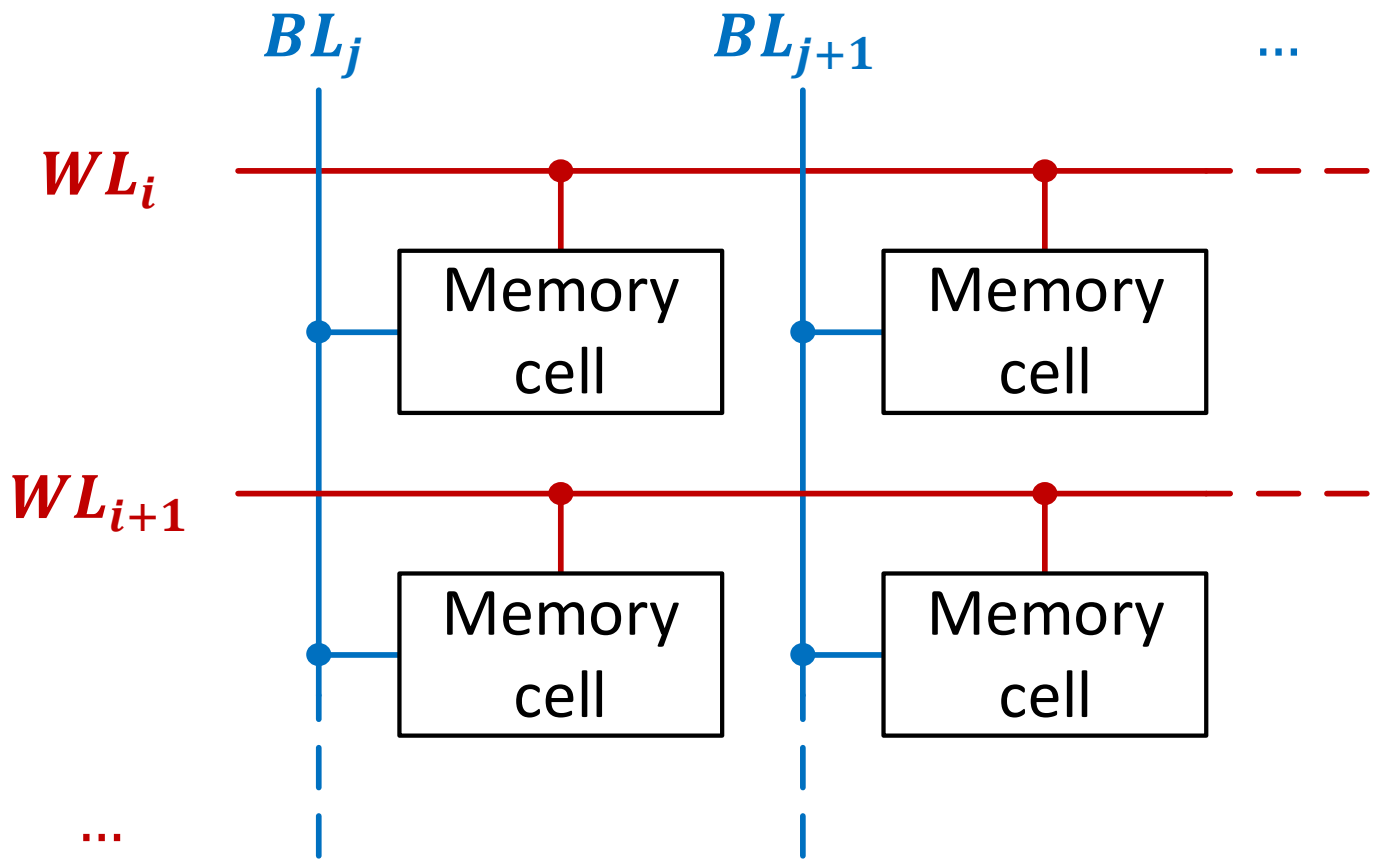
Semiconductor Memories – Architecture outline

- Draft
  - WL_i = Word Line



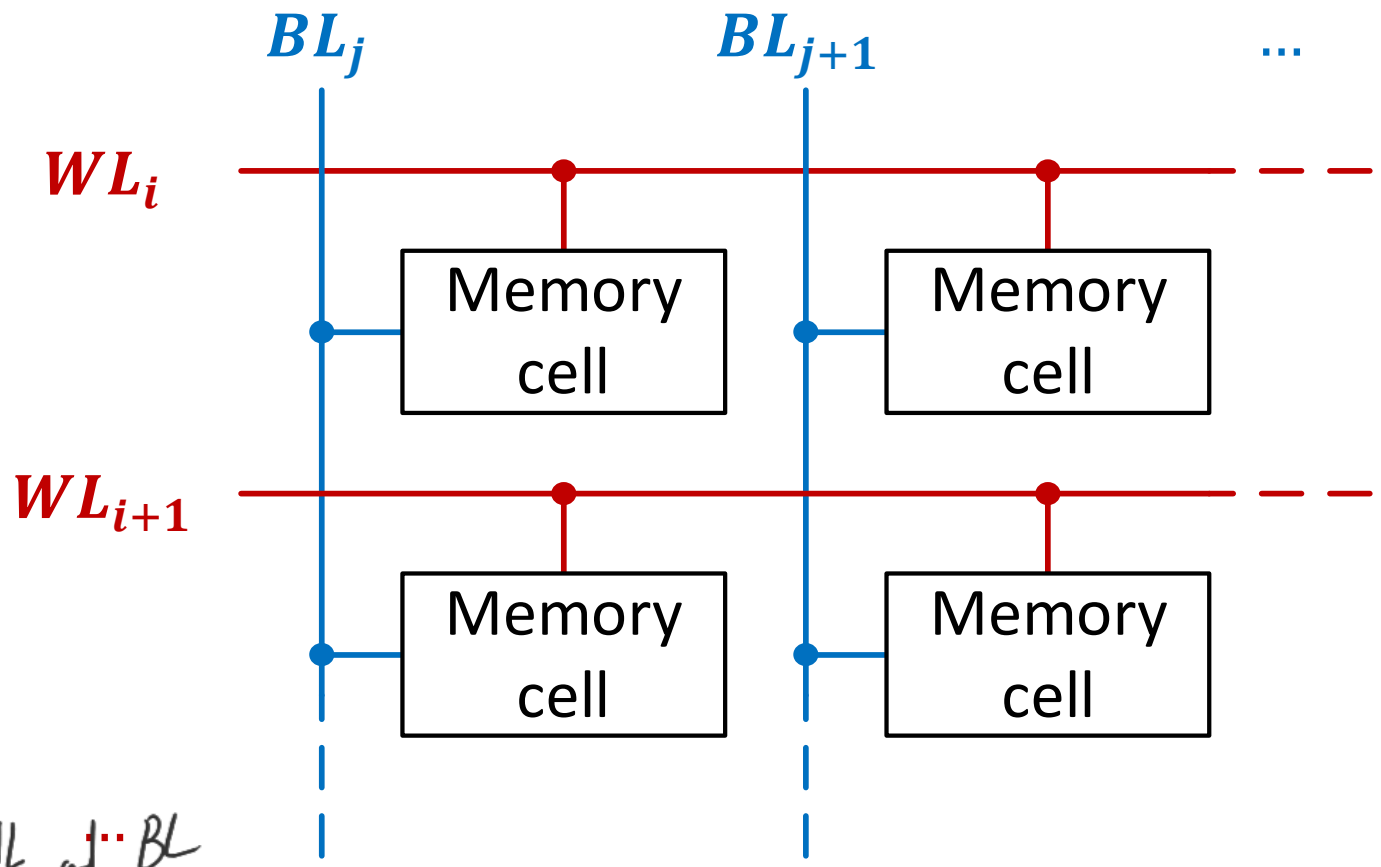
Semiconductor Memories – Architecture outline

- Draft
  - WL_i = Word Line
  - BL_j = Bit Line



Semiconductor Memories – Architecture outline

- Draft
  - WL_i = Word Line
  - BL_j = Bit Line
  - Memory cell
    - Typically contains 1 transistor
    - Or something different, but with "similar" cost in terms of silicon area



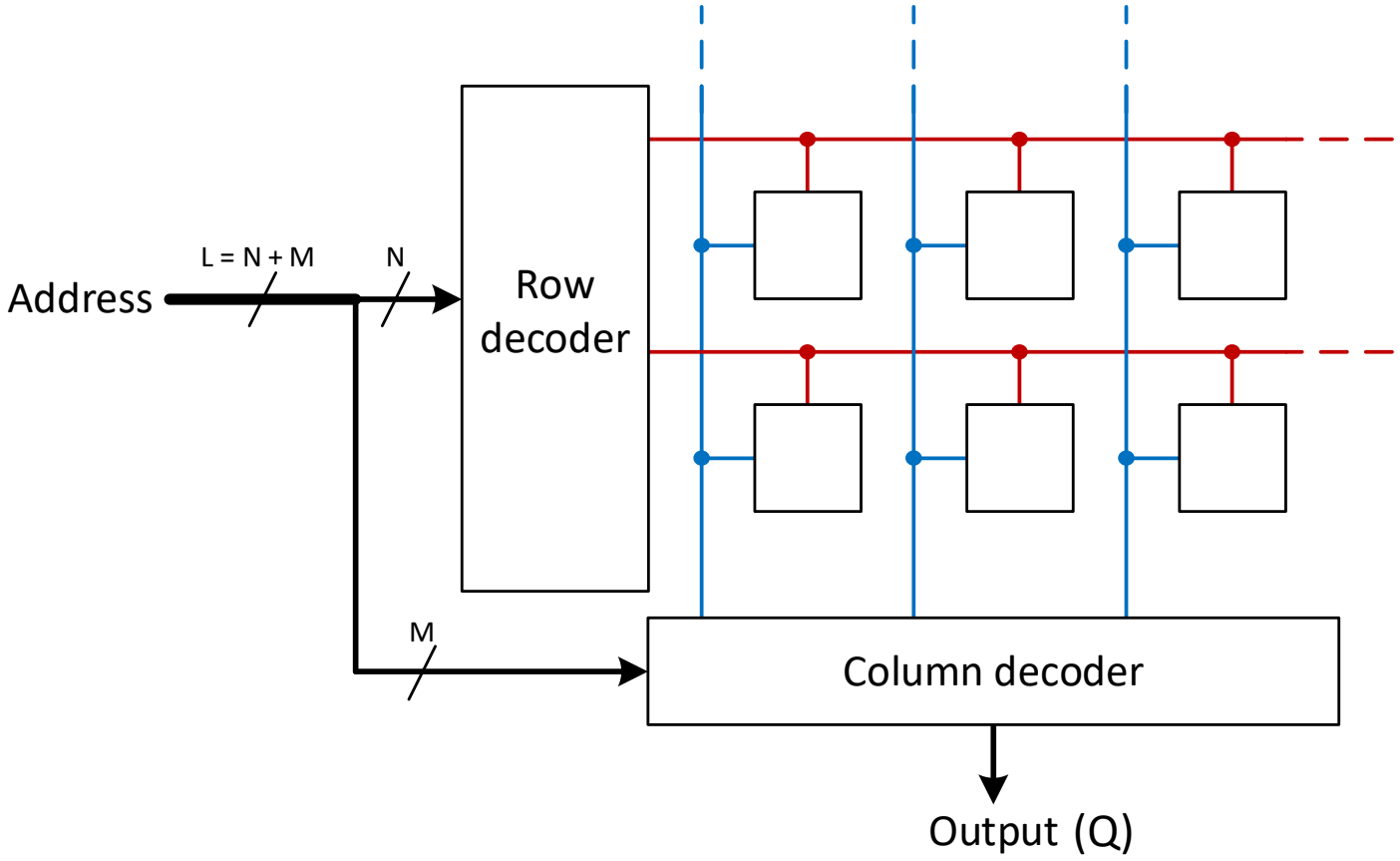
each cell is linked to just one WL and BL

Semiconductor Memories – Architecture outline

- Draft
  - To complete the architecture, they are required
    - A row decoder (or x-decoder)
      - For decoding the Word Line address
    - A column decoder (or y-decoder)
      - For decoding the Bit Line address

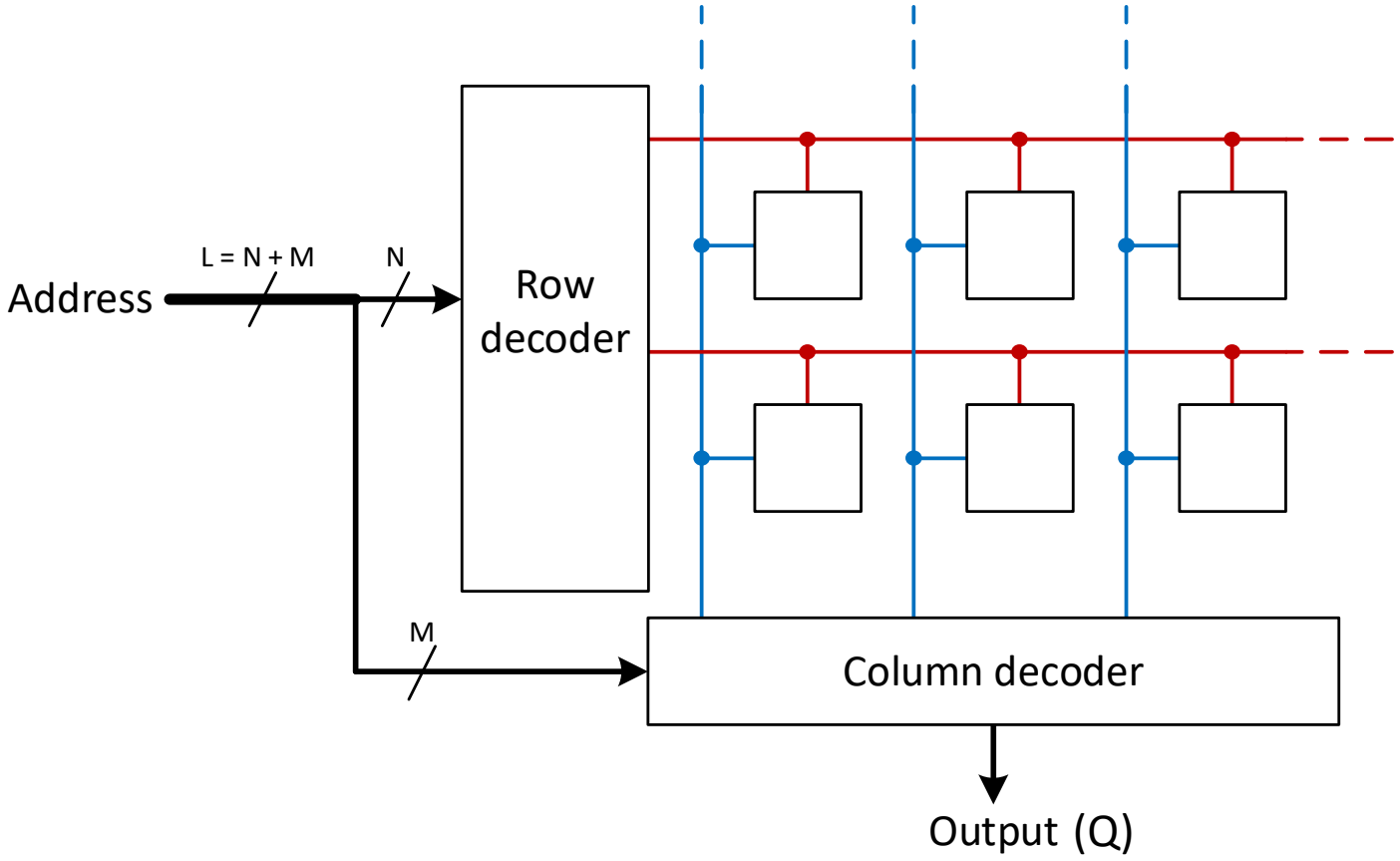
Semiconductor Memories – Architecture outline

- Final



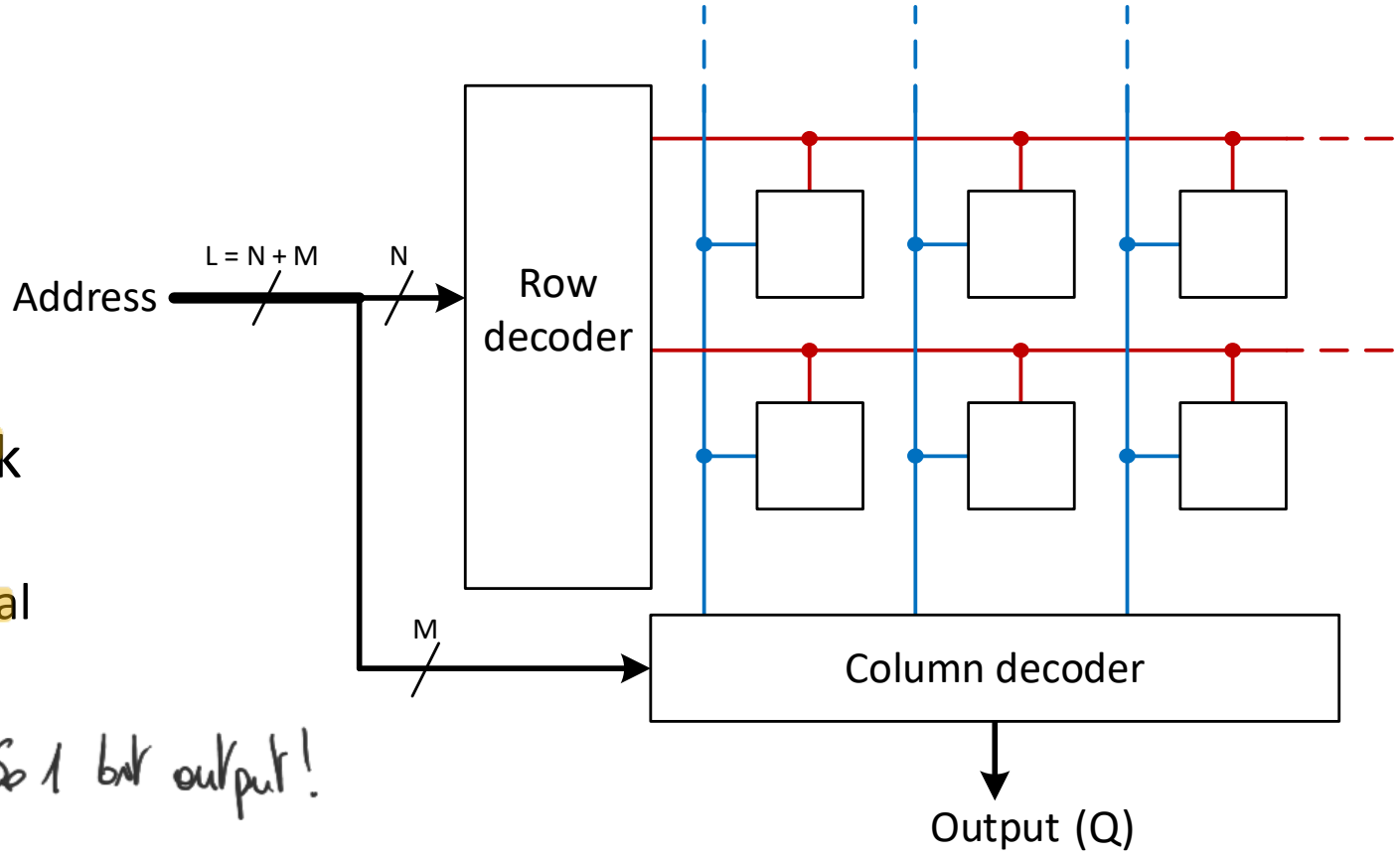
Semiconductor Memories – Architecture outline

- Final – Read-Only Memory
  - But only for ROM
    - Just output (Q)



Semiconductor Memories – Architecture outline

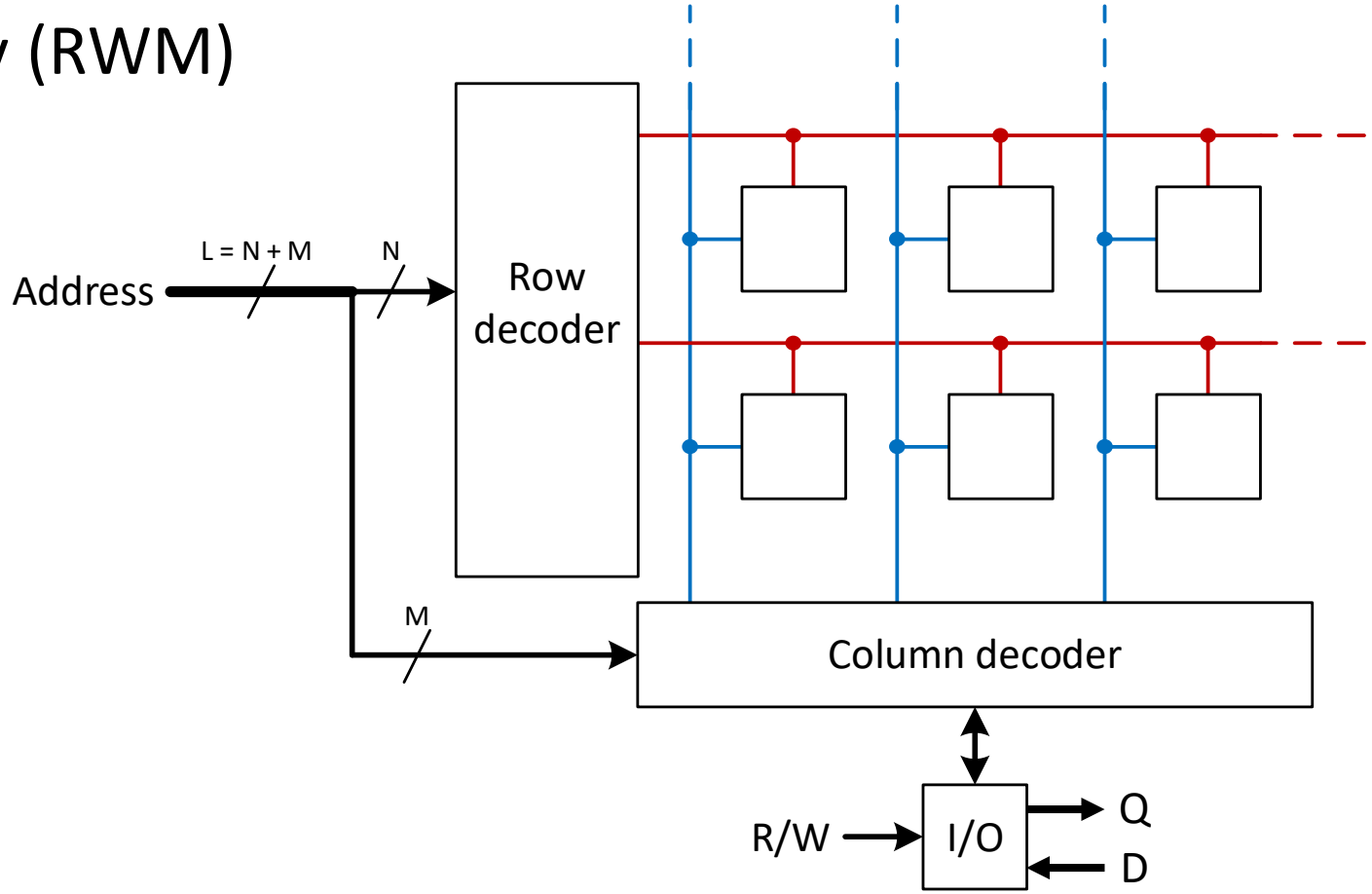
- Final – Read-Only Memory
  - But only for ROM
    - Just output (Q)
  - In case of RWM, an I/O block is required
    - And a Read/Write (R/W) signal



1 bit memory cells! So 1 bit output!

Semiconductor Memories – Architecture outline

- Final – Read-Write Memory (RWM)
  - D = input data (bit)
  - Q = output data (bit)



Semiconductor Memories – Architecture outline

- The architecture shown before is the one for single-bit memories
  - Each memory cell store 1 bit
  - Read or Write of 1 bit at a time

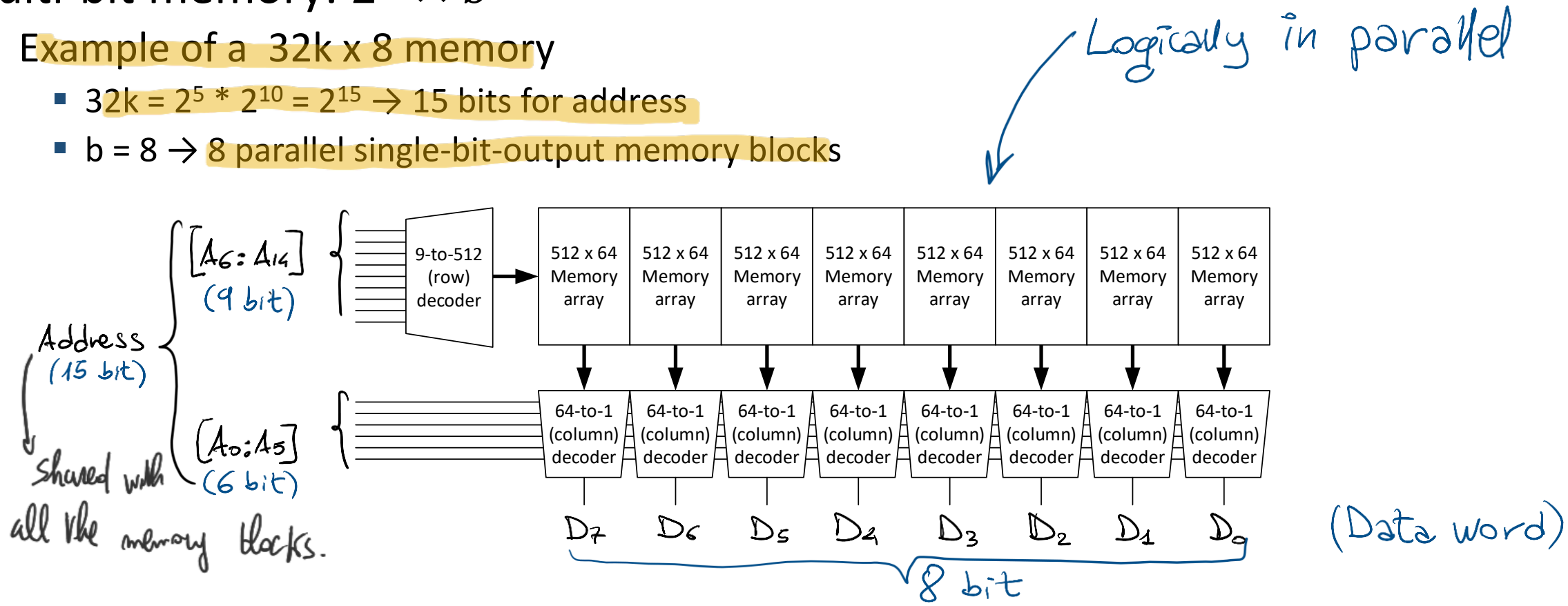
- Multi-bit memories are built by connecting in parallel single-bit memories
  - Assuming
    - Number of memory cells = 2^L → $L = N + M$ bits of address
    - Bit length = b (bits)
  - Memory is organized as $2^L \times b$
  - Data (of b bits) is called *word*

Semiconductor Memories – Architecture outline

- Multi-bit memory: $2^L \times b$
  - b parallel single-bit block
  - Each single bit block
    - Contains 2^L (single-bit) memory cells
    - That are arranged as a matrix of from $2^N \cdot 2^M$
  - Typically, L is an integer multiple of 10, so that …
    - $L = 10 \rightarrow 2^{10} \rightarrow$ k (kilo)
    - $L = 20 \rightarrow 2^{20} \rightarrow$ M (Mega)
    - $L = 30 \rightarrow 2^{30} \rightarrow$ G (Giga)

Semiconductor Memories – Architecture outline

- Multi-bit memory: $2^L \times b$
  - Example of a 32k x 8 memory
    - 32k = $2^5 * 2^{10} = 2^{15}$ → 15 bits for address
    - b = 8 → 8 parallel single-bit-output memory blocks



Thank you for your attention

Luca Crocetti
(luca.crocetti@unipi.it)