OF THE
DEPARTMENT OF
INFORMATION ENGINEERING

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Digital Systems

Memory Basics

Marta Bagatin, marta.bagatin@unipd.it

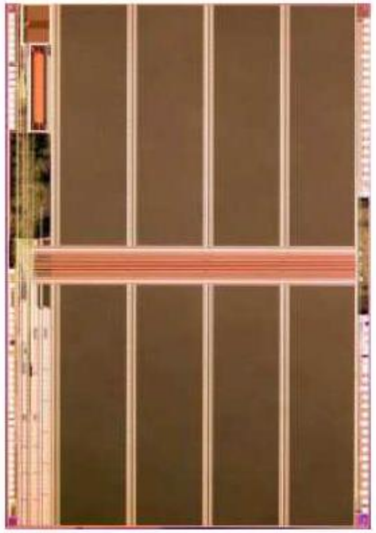
Degree Course in Information Engineering
Academic Year 2023-2024

Purpose of the Lesson

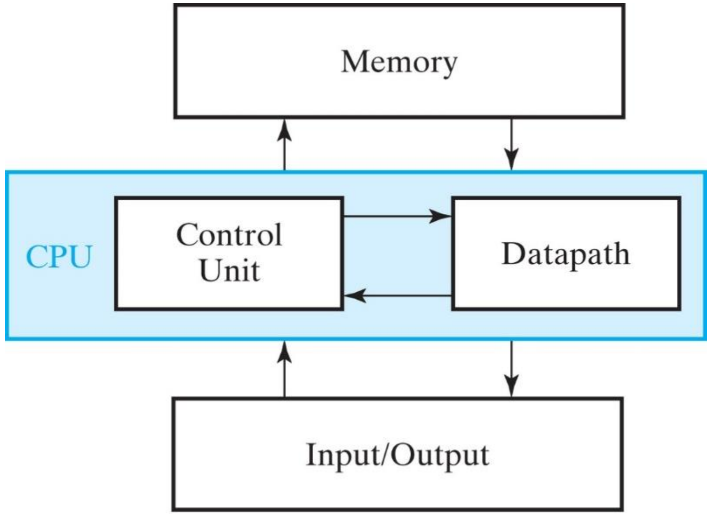
- Introduce the basic concepts of memories

Introduction

- Memory is a collection of binary storage cells together with associated circuits needed to transfer information into and out of the cells (**control circuitry**)
- In a calculator, memory is used to store instructions and data (input, output, or intermediate data of logic, arithmetic, shift operations, etc.)
- The operation of transferring new data to the memory is called **write**
- The operation of transferring the data stored in the memory to the output of the memory itself is called **read**
- Binary information is stored in memories in groups of bits. Each group of bits is called **word** (often consisting of a number of bits equal to multiples of 8), e.g. 16-bit words
- A memory can be **stand alone** or integrated in a chip (**embedded**)



Stand-alone Flash memory
Courtesy of Micron



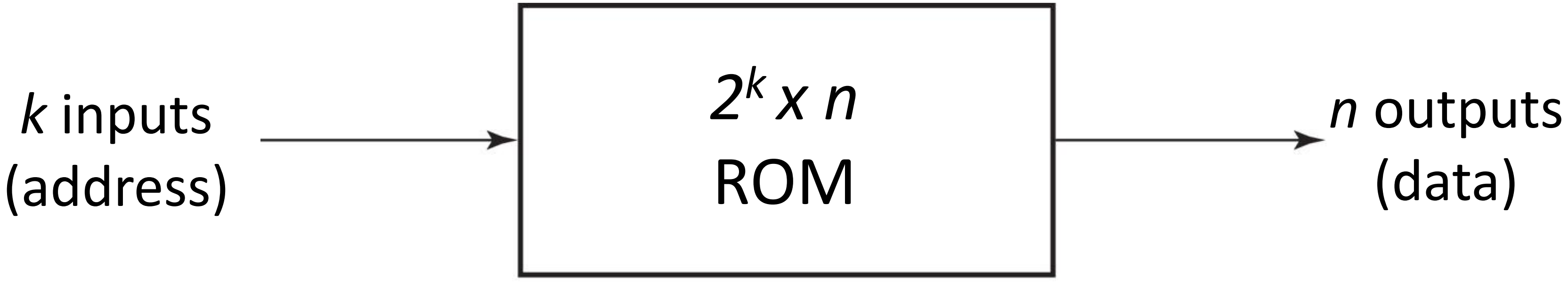
Classification of Memories

- Memories are classified according to the possibility of reading and writing data
  - **Read-only memories**: ROM (Read Only Memory). We have seen previously that there are also Programmable ROMs (EPROM, E^{2}PROM, Flash)
  - **Read and write memories**: memories in which read and write operations happen very frequently
- Memories are classified according to the time required to access the stored information
  - **RAM (Random Access Memory)**: each word in the memory can be accessed with the same effort (e.g. SRAM, DRAM)
  - **SAM (Serial Access Memory)**: access to a bit requires access to entire sequences of data (e.g. magnetic disk, CD), so the access time depends on the position of the bit in the memory
  - **Hybrid access Memories**: accessible at block/sector level (e.g. Flash)
- Memories are also classified according to their ability to retain information when power is removed
  - **Volatile memories** (e.g. SRAM, DRAM): lose information in the absence of power
  - **Non-volatile memories** (e.g. Flash): keep information even in the absence of power

Characteristics of Memories

- Memories are characterized by a series of parameters that determine their performance (and cost)
  - **Capacity** (density): total number of stored bytes (or bits)
  - **Speed**
    - Random access time: amount of time required to gain access to a cell [ns]
    - Throughput: rate at which sequential data can be accessed [Mbit/s]
  - **Power consumption**
  - **For non-volatile memories**
    - **Retention**: Ability to store information for a long time (e.g. 10 years)
    - **Endurance**: Ability to be erased and programmed several times without degrading (e.g. 10^5 cycles)

ROM: Review

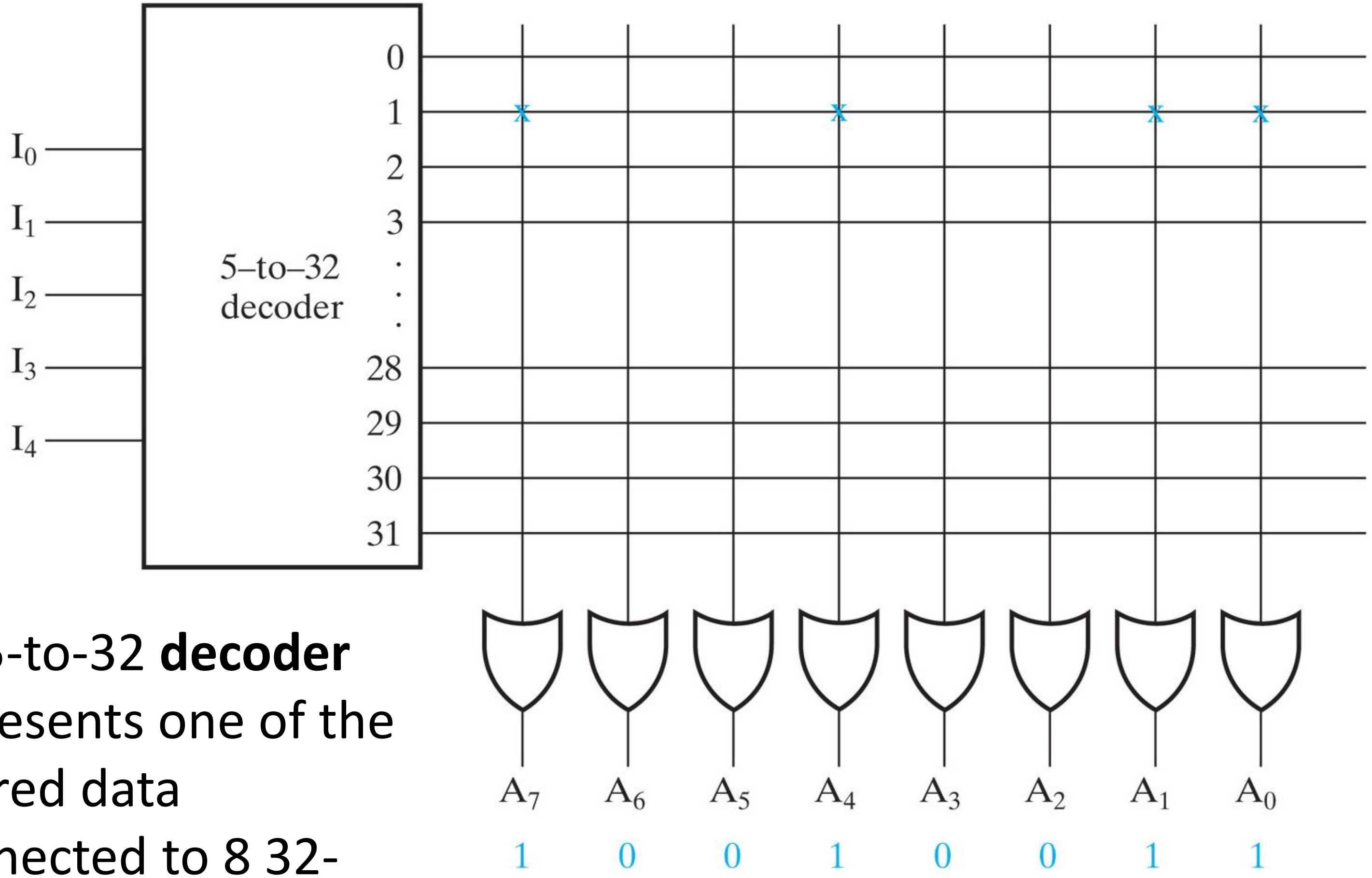


- The **input** contains the **address** of the data
- The **output** contains the **data** bit (word) selected by the address provided at the input
- A ROM can store ***2^k*** **words of *n* bits**
- Being a **read-only memory**, there is no possibility to write on it, i.e. a ROM does not have input data

32 x 8 ROM: Example (32 8-bit Words)

A **2^k x n ROM** consists of a k-to-2^k decoder and n OR gates

Fixed AND plane (decoder)

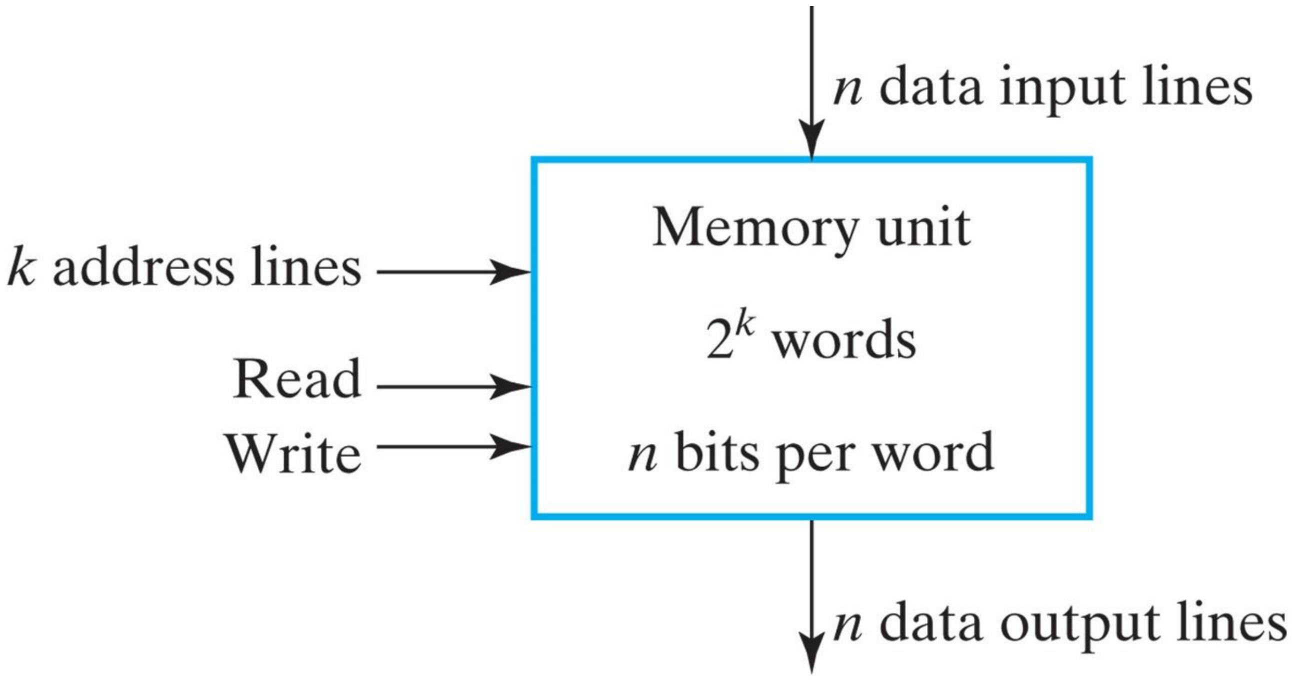


Programmable OR plane

- Input (5 address bits) goes to a 5-to-32 **decoder**
- Each output of the decoder represents one of the 32 possible addresses of the stored data
- The 32 decoder outputs are connected to 8 32-input **programmable OR gates** (x indicates the presence of a connection)
- In the example, address 1 of the memory contains the word "10010011"

Random Access Memory (RAM)

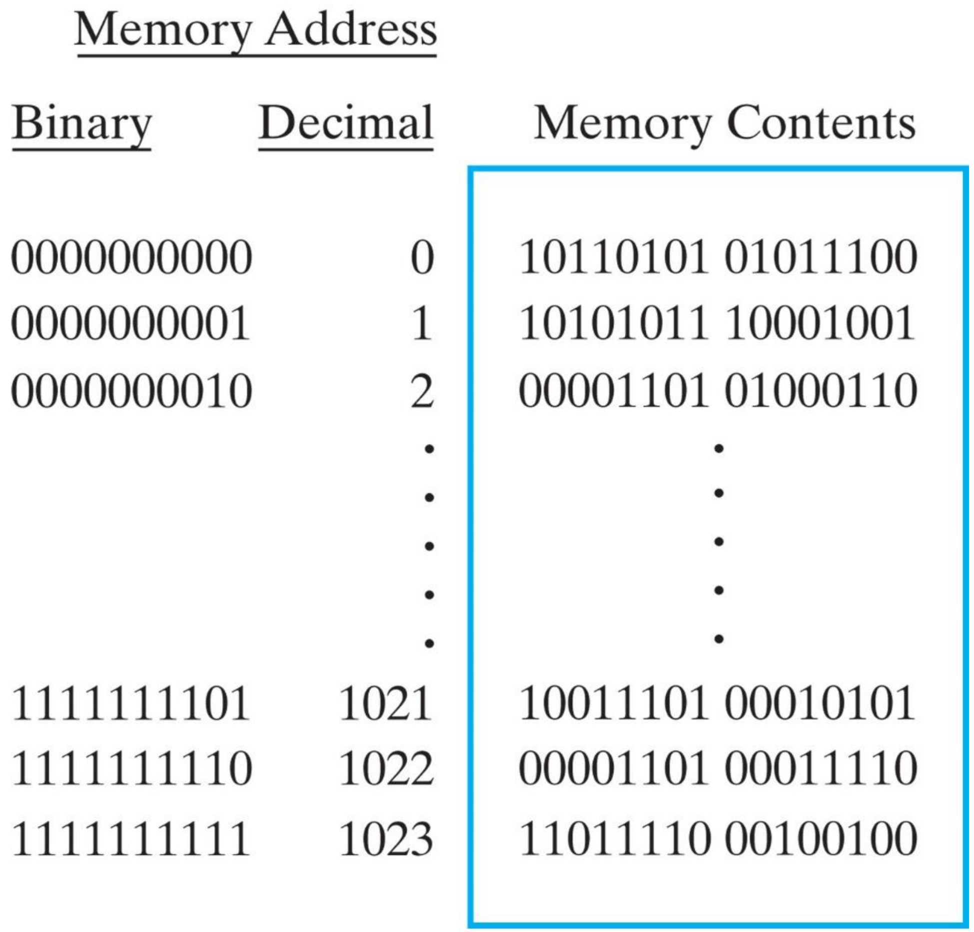
- The interface of RAM with the outside world is given by
  - Data input (n bits): information to be written in the memory
  - Data output (n bits): information coming out of the memory
  - Address (k bit): identifies a particular word within the memory
  - Control input (read, write): indicates the type of operation to be performed on the memory
- The capacity (size) of the RAM is given by the number of words (2^k) multiplied by the number of bits in each word



Size = $\underbrace{2^k}_{= 2^k \text{ word}}$ x n bit

Random Access Memory (RAM)

- Possible addresses range from 0 to 2^k-1 (k is the number of bits of the address)
- The **number of bits of the address (k) is independent of the number of bits (n) of the words** stored in the memory
  - Ex: the RAM in the figure has 16-bit (2-byte) data lines, 10-bit address, therefore it contains 1024 16-bit words. Its capacity is 1K x 16 bits (1K = 1024). In this example, the location of address 1 contains the data "10101011 10001001"
- NOTE: when a word is read or written, the memory operates in parallel on all the bits of the word, as if they were a single unit (word is minimum number of addressable bits)

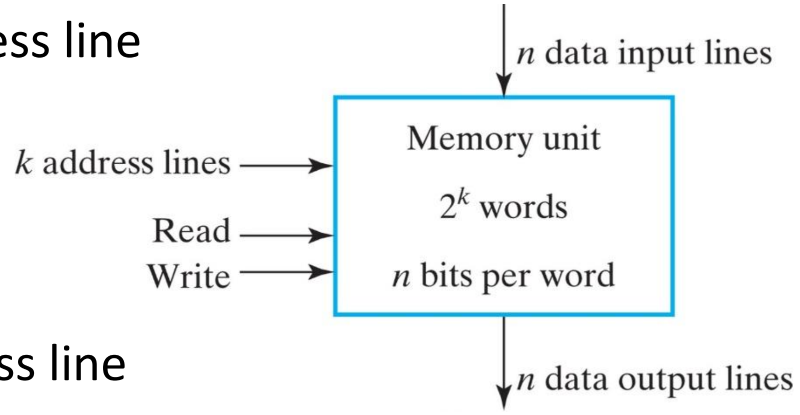


Memory Address		
Binary	Decimal	Memory Contents
0000000000	0	10110101 01011100
0000000001	1	10101011 10001001
0000000010	2	00001101 01000110
	·	·
	·	·
	·	·
	·	·
	·	·
1111111101	1021	10011101 00010101
1111111110	1022	00001101 00011110
1111111111	1023	11011110 00100100

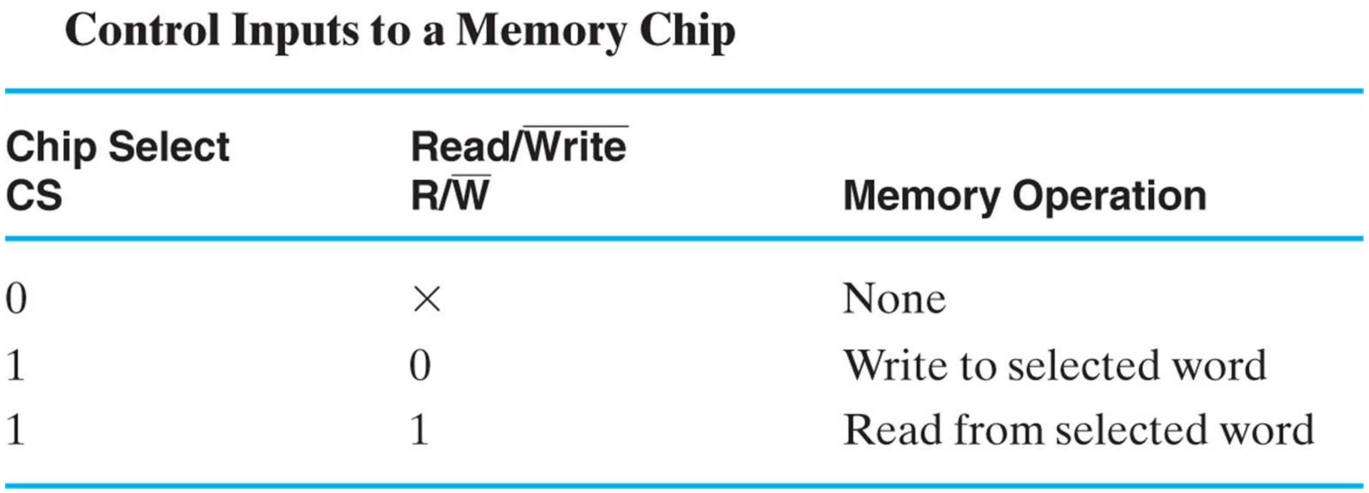
Size = 2^{10}x 16 bit

RAM: Read and Write Operations

- **Writing** the memory includes the following steps
  - Apply the binary address of the word you want to write to the address line
  - Apply the data to be written to the data input line
  - Activate the Write control signal
- **Reading** the memory includes the following steps
  - Apply the binary address of the word you want to read to the address line
  - Activate the Read control signal



- A control signal (**Chip Select**) selects the portion of the memory chip, enabling the ability to perform write and read
- The signals are provided by an external unit (e.g. CPU)

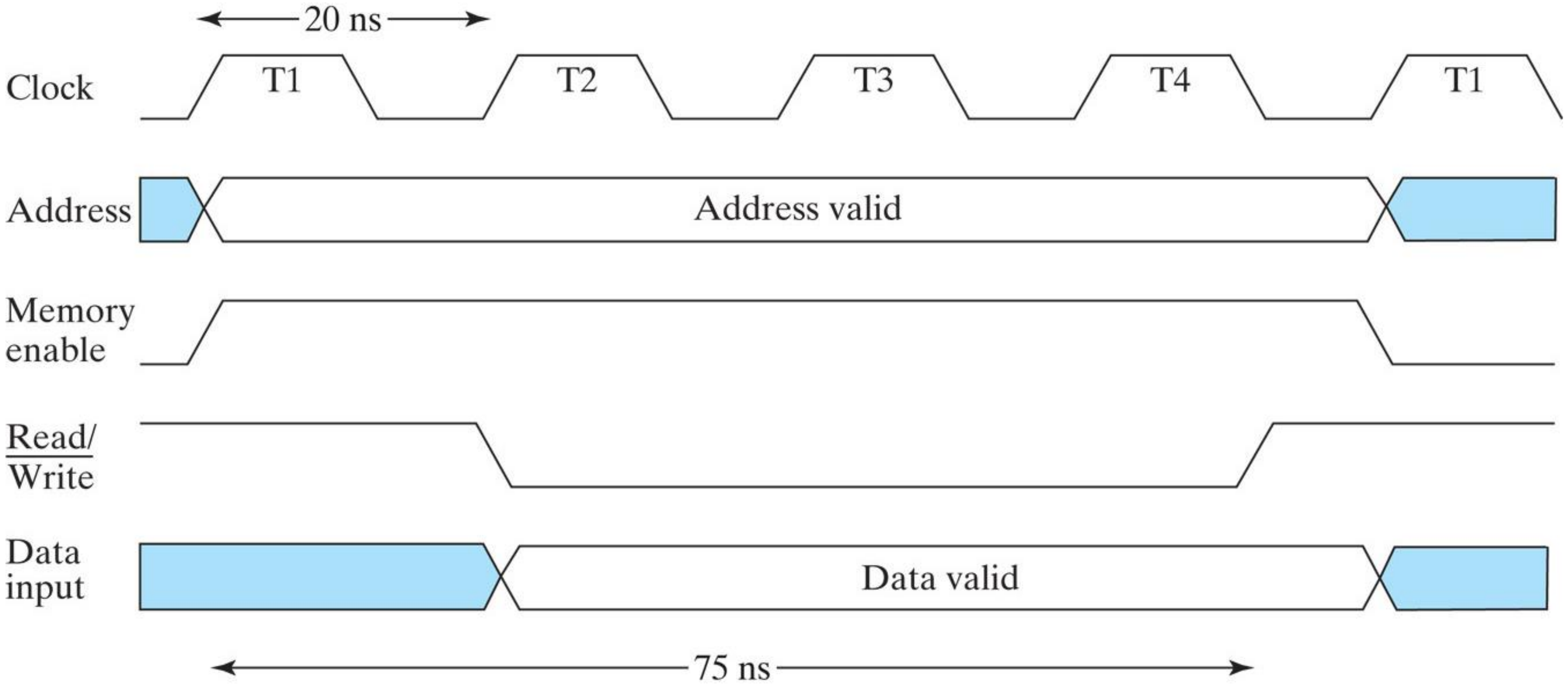


Control Inputs to a Memory Chip

Chip Select CS	Read/$\overline{\text{Write}}$ $R/\overline{W}$	Memory Operation
0	×	None
1	0	Write to selected word
1	1	Read from selected word

RAM Write: Example of Waveforms

- System clock (e.g. CPU) with period of 20 ns (50 MHz)
- Let's assume the time for a write cycle is 75 ns, so the CPU must allocate 4 clock cycles for the write operation



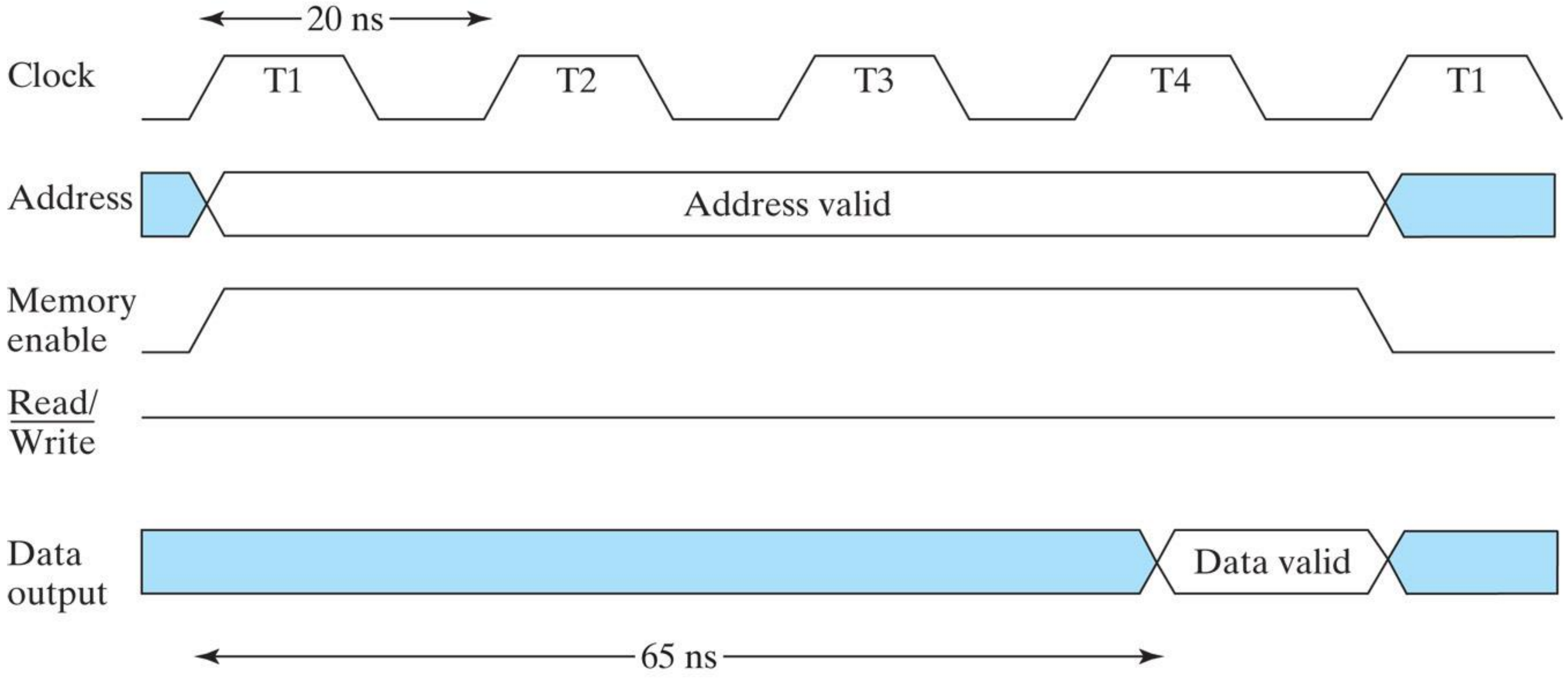
(a) Write cycle

- At the first rising edge of the clock, the address of the data to be written is provided to Address and Memory enable is asserted
- On the second rising edge $\overline{\text{Write}}$ is put at 0 (if it was done before the address is stable at the correct value, we would accidentally write other memory locations) and the data to be written is provided on Data input
- Address and data must remain stable for a time beyond the rise of $\overline{\text{Write}}$, to avoid writing other cells

Crossing lines indicate a possible change in the value of the line bits. Blue areas indicate unspecified data

RAM Read: Example of Waveforms

- System clock (e.g. CPU) with period of 20 ns (50 MHz)
- Let's assume the time for a read cycle is 65 ns, so the CPU must allocate 4 clock cycles for the read operation



(b) Read cycle

- At the first rising edge of the clock, the address of the data to be read is provided to Address and Memory enable is asserted
- After a time equal to the access time (read cycle: 65 ns in this example), the memory will provide the content of the addressed word at the Data output
- At the next clock edge, the CPU transfers the data read into one of its registers and the values present at the data address and control signals can change again

Static and Dynamic RAM

- RAMs can be static or dynamic, depending on how binary information is stored
- In **SRAM (Static Random Access Memory)** the bits are stored in bistable elements (latches). This keeps the information valid as long as the circuit is powered (without requiring external intervention, hence the name ‘static’)
- The **DRAM (Dynamic Random Access Memory)** stores information in the form of electrical charge stored in a capacitor (accessed via an n-channel MOSFET). The capacitor tends to discharge over time, so periodic operations are required (refresh= rewrite of the memory cell)
- SRAMs are simpler to use (do not require refresh) and are faster, while DRAMs occupy less area (therefore they cost less, since they offer a higher number of bits per unit area)
  - Example of SRAM use: cache (very fast but density is not important)
  - DRAM use example: main memory (basic requirement is density, lower cost per bit)

Disclaimer

Figures from*Logic and Computer Design Fundamentals*, Fifth Edition, GE Mano |Kime| Martin