Semiconductor Memory Organization & Addressing

Organización del Computador 2025

Basel - Ferreyra - Laprovitta - Vélez - Vodanovic

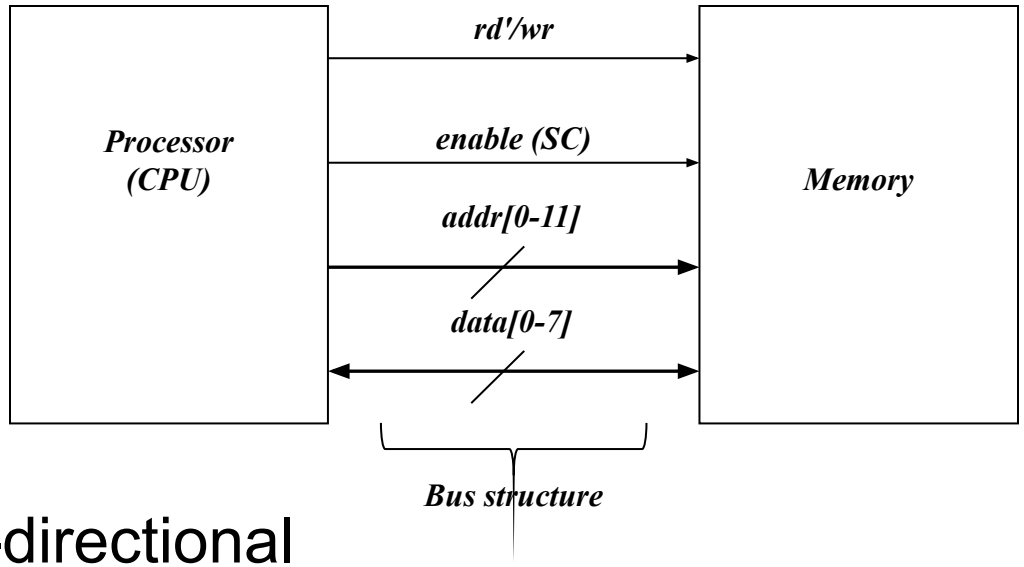
FAMAF Facultad de Matemática, Astronomía, Física y Computación

UNC Universidad Nacional de Córdoba

Introduction

- Computer systems functionality aspects
  - Processing
    - Transformation of data
    - Implemented using processors
  - Storage
    - Retention of data
    - Implemented using memory
  - Communication
    - Transfer of data between processors and memories
    - Implemented using buses
    - Called *interfacing*

A simple bus

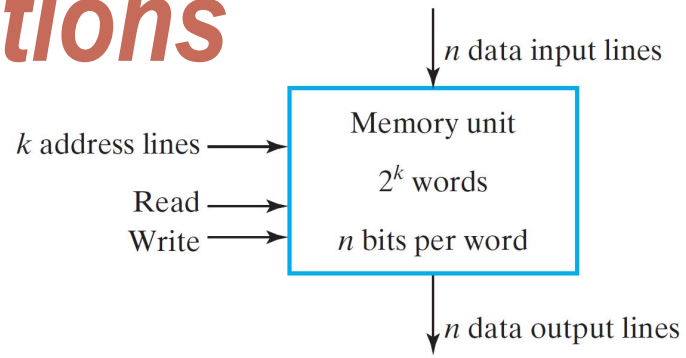


- Wires:
  - Uni-directional or bi-directional
  - One line may represent multiple wires
- Bus
  - Set of wires with a single function
    - Address bus, data bus
  - Or, entire collection of wires
    - Address, data and control
    - Associated protocol: rules for communication

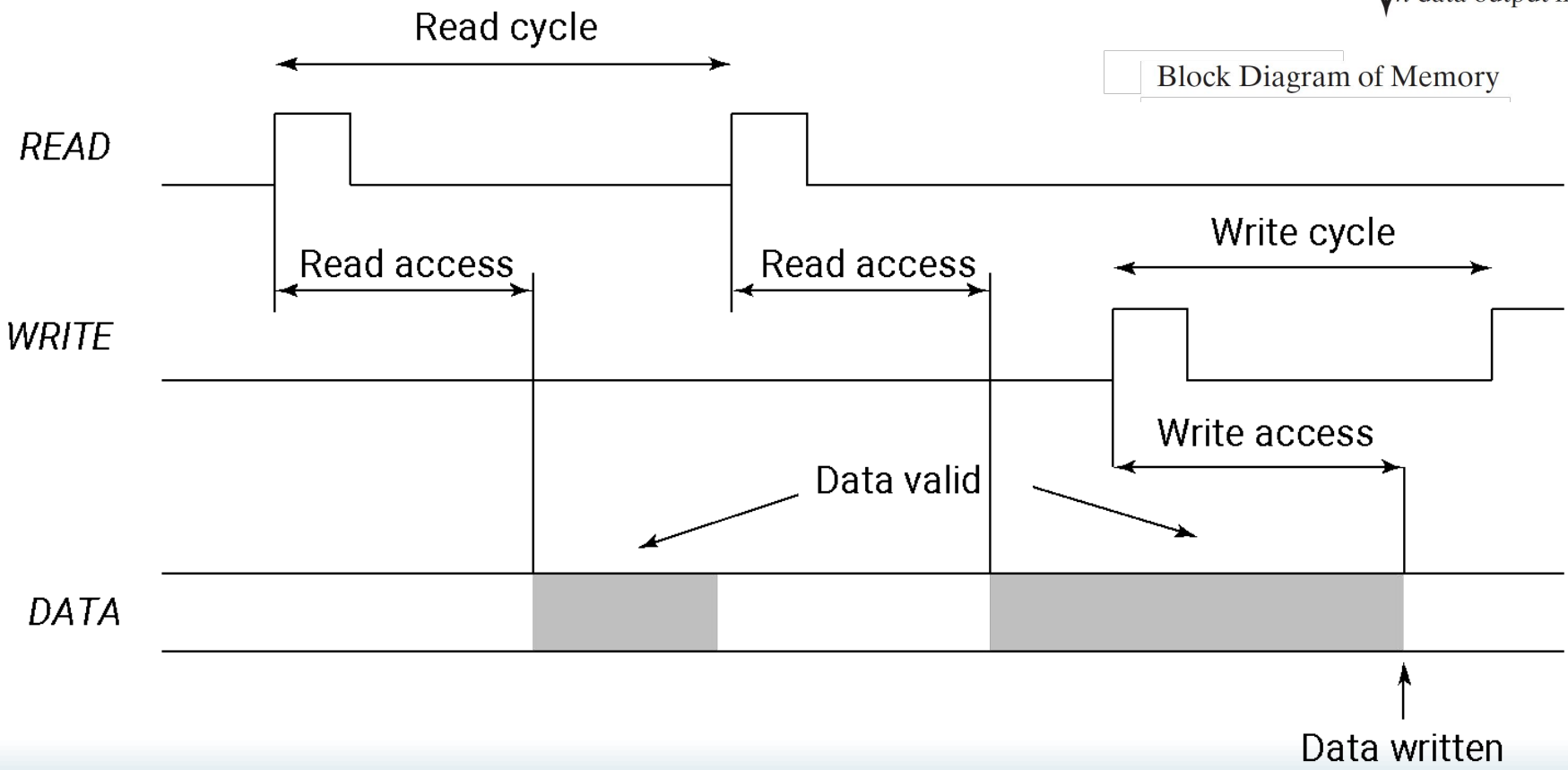
Semiconductor Memory Classification

Read-Write Memory		Non-Volatile Read-Write Memory	Read-Only Memory
Random Access	**Non-Random Access**	EPROM E^2PROM FLASH	Mask-Programmed Programmable (PROM)
SRAM DRAM	FIFO LIFO Shift Register CACHE		

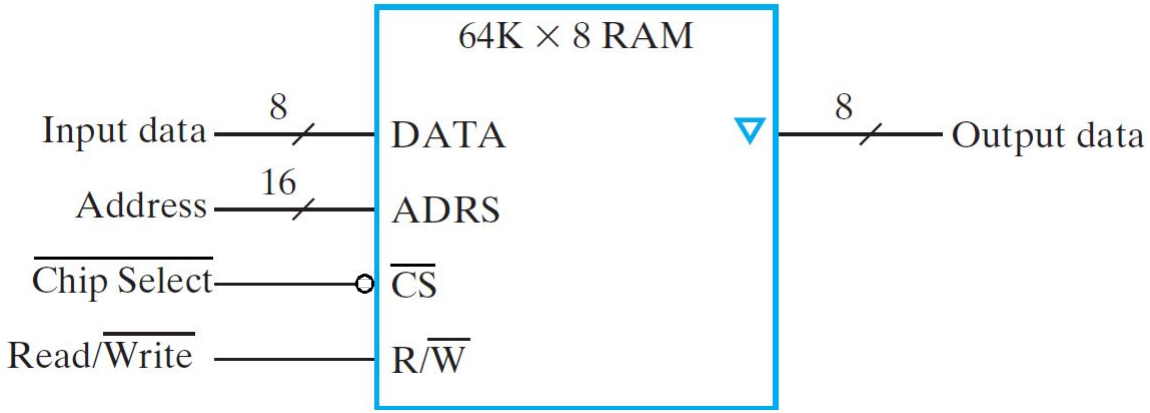
Memory Timing: Definitions



Block Diagram of Memory

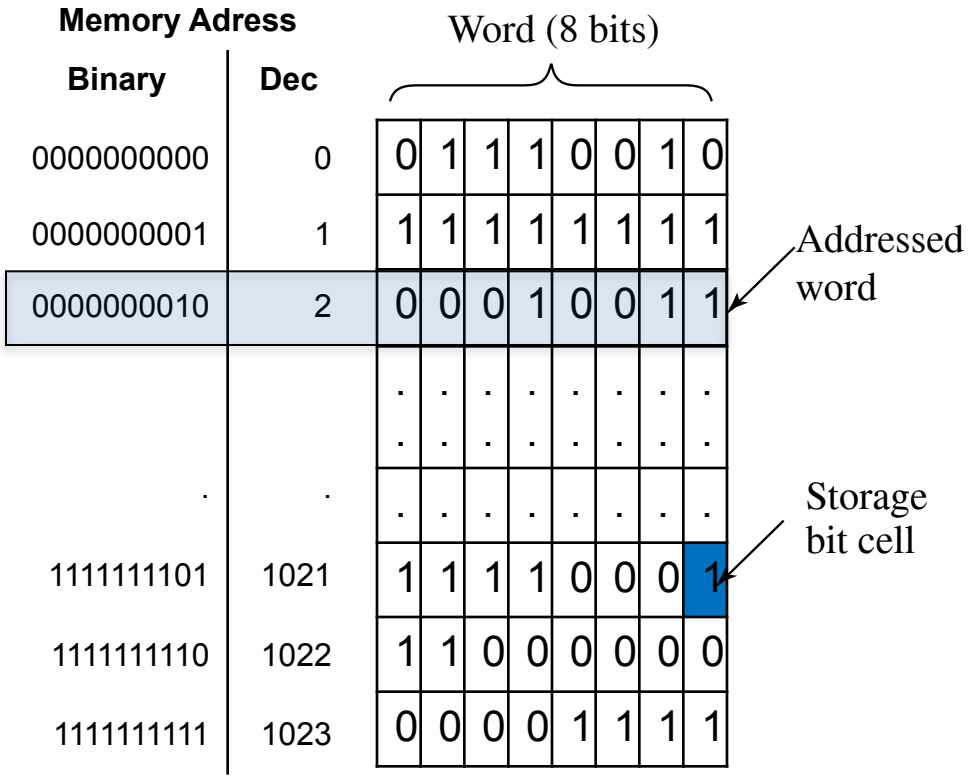


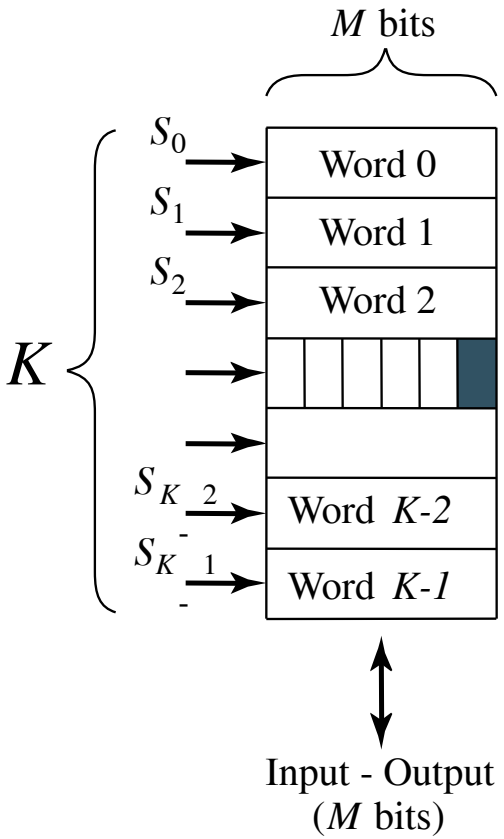
Control Inputs to a Memory Chip



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

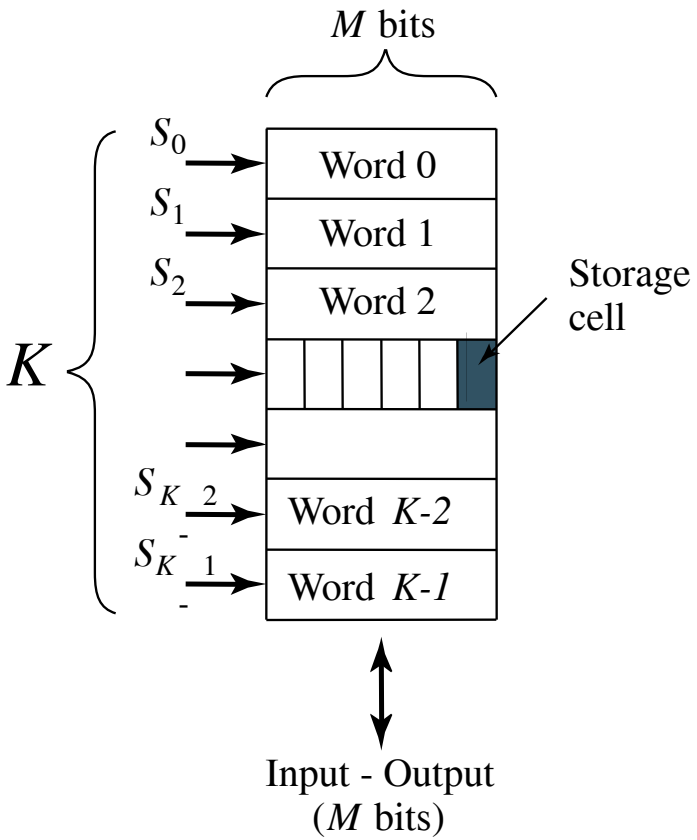
Memory Organization

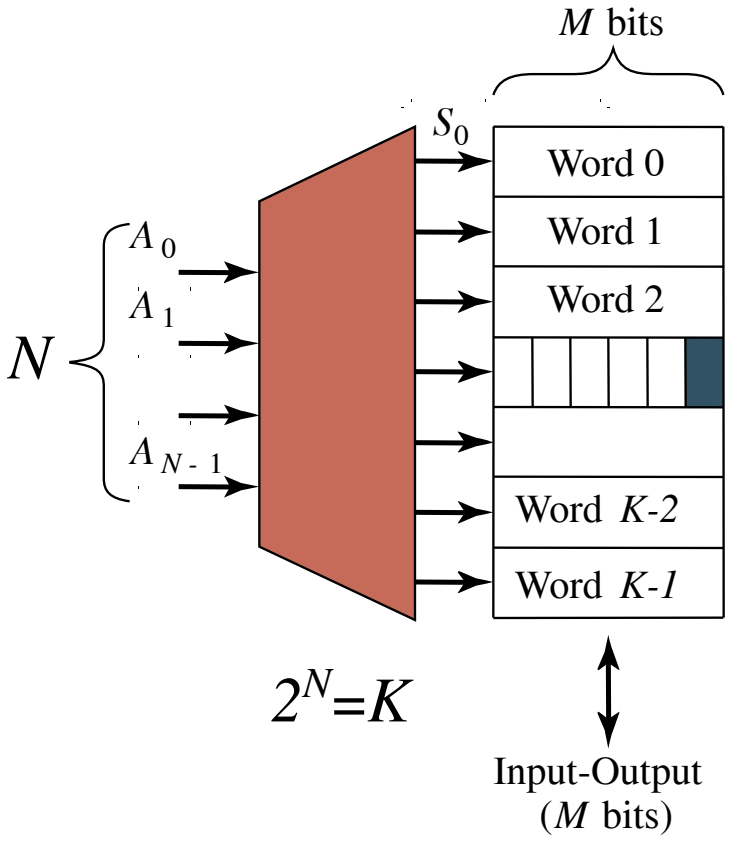




Example organization for
1Kword x 8 bits = 8K bits memory

Memory Architecture: Decoders



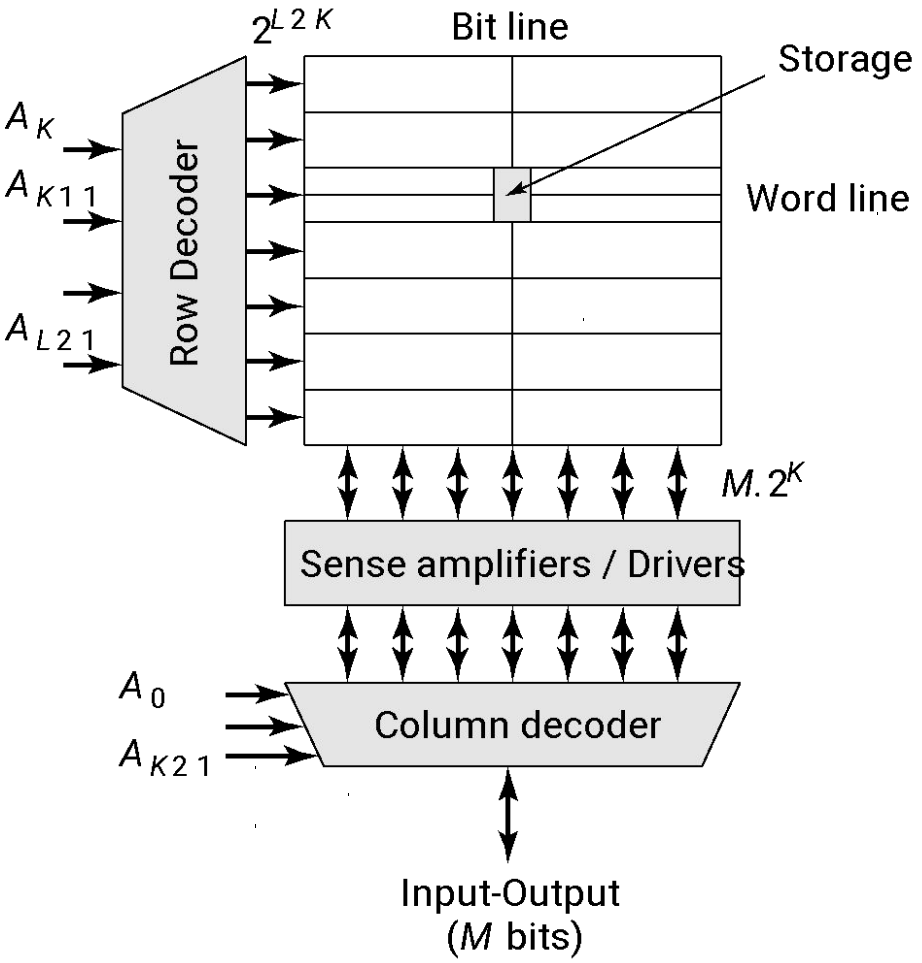


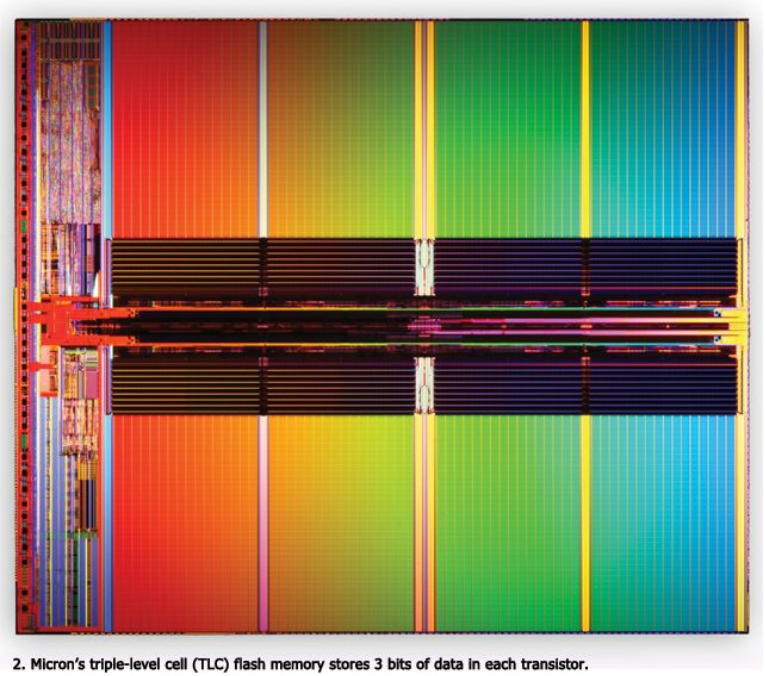
Intuitive architecture for K x M memory
Too many select signals:
K words == K select signals

Decoder reduces the number of select signals $N = log_2 K$

Array-Structured Memory Architecture

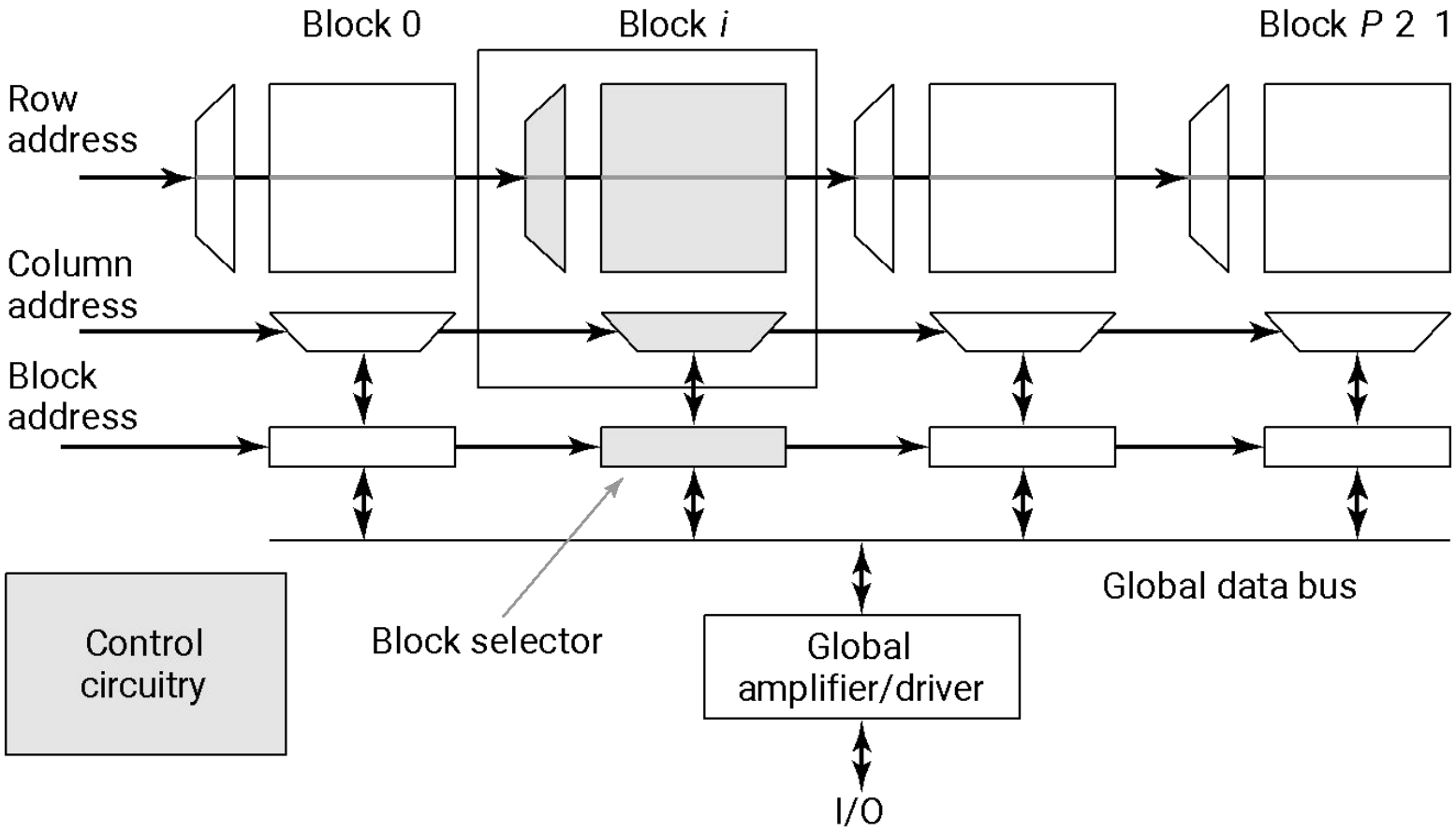
Problem: ASPECT RATIO or HEIGHT >> WIDTH





2. Micron's triple-level cell (TLC) flash memory stores 3 bits of data in each transistor.

Hierarchical Memory Architecture

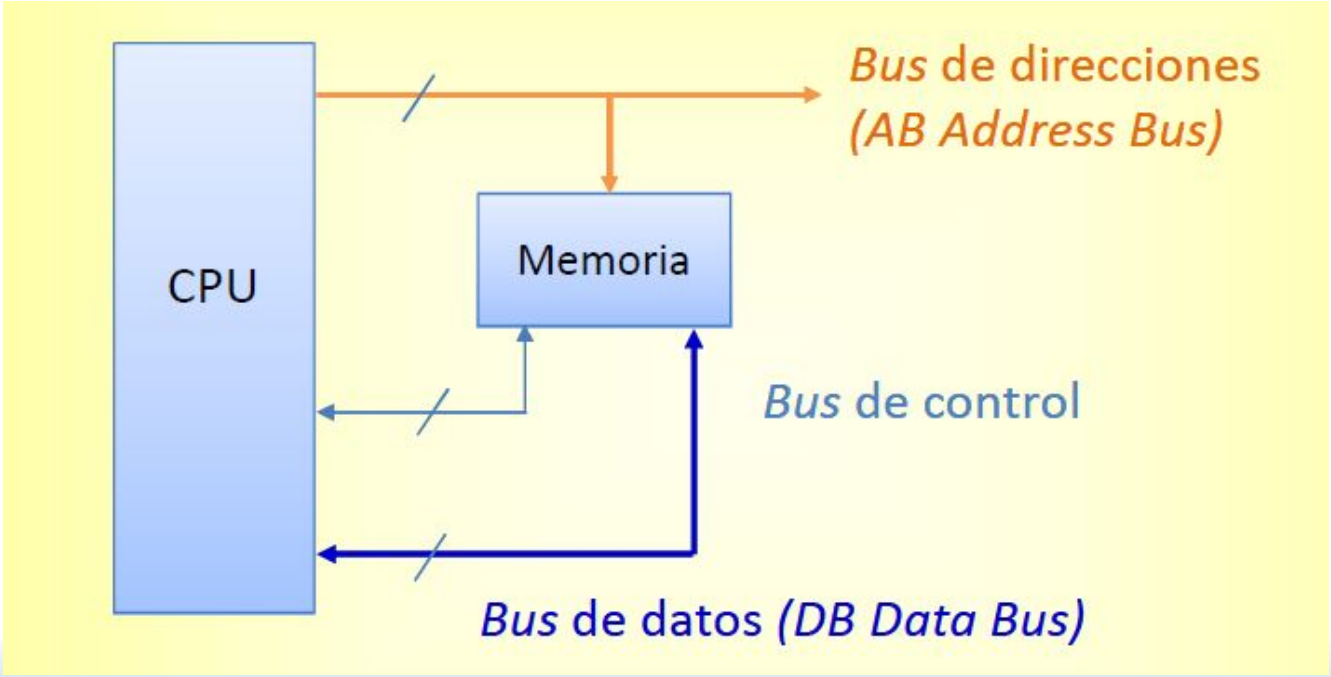


Advantages

1. Shorter wires within blocks
2. Block address activates only 1 block => power savings

Addressable Space

- It´s defined as the **total number** of addresses the CPU can access.
- It depends on the width (number of bits) of the address bits: n bits -> 2^n addresses



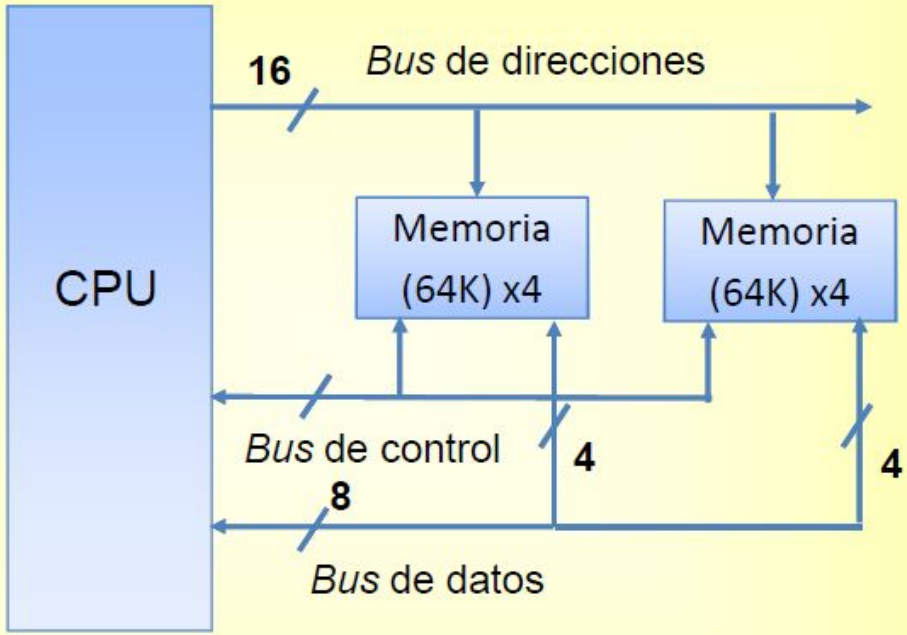
Size prefixes and symbols

N bits (2^N)		Capacidad (en words)
2^1	=	**2** words
2^2	=	**4** words
2^3	=	**8** words
2^4	=	**16** words
2^5	=	**32** words
2^6	=	**64** words
2^7	=	**128** words
2^8	=	**256** words
2^9	=	**512** words

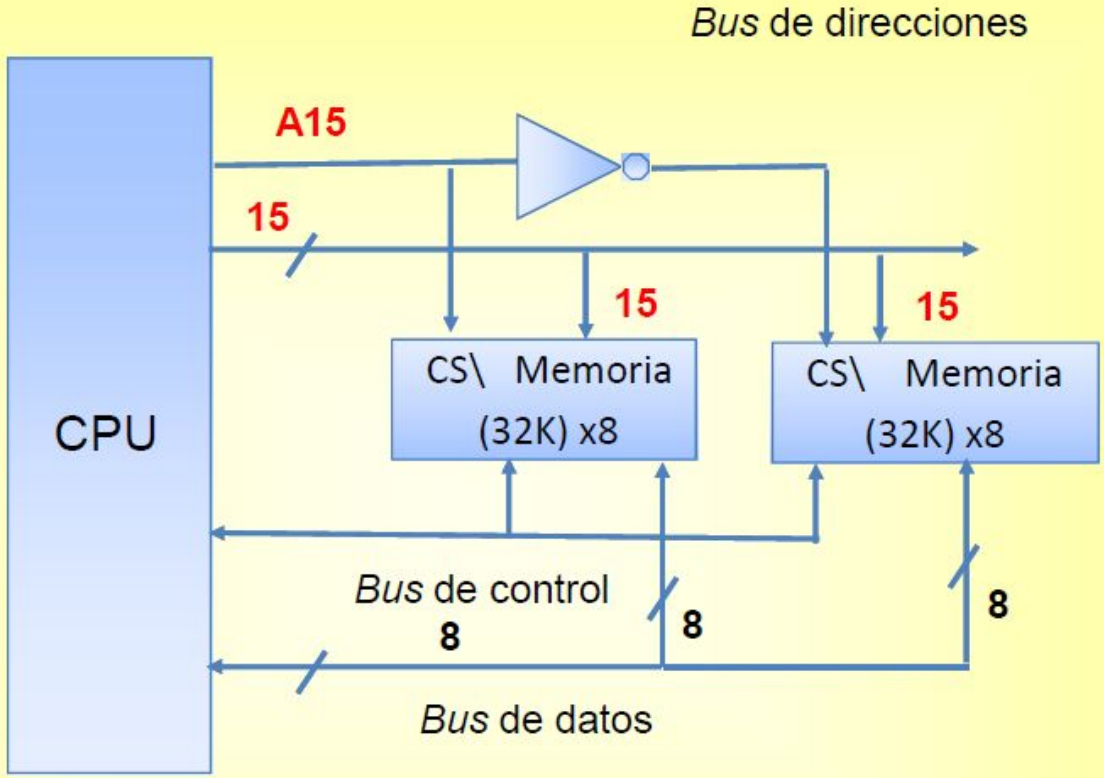
N bits (2^N)		Capacidad (en words)		Simbolo [Prefijo]
2^{10}	=	1024 words	=	**1Kw [Kilo]**
2^{20}	=	1024 Kw	=	**1Mw [Mega]**
2^{30}	=	1024 Mw	=	**1Gw [Giga]**
2^{40}	=	1024 Gw	=	**1Tw [Tera]**
2^{50}	=	1024 Tw	=	**1Pw [Peta]**

SIZE	PREFIX	SYMBOL	SIZE	PREFIX	SYMBOL
10^3	Kilo-	K	2^{10}	Kibi-	Ki
10^6	Mega-	M	2^{20}	Mebi-	Mi
10^9	Giga-	G	2^{30}	Gibi-	Gi
10^{12}	Tera-	T	2^{40}	Tebi-	Ti
10^{15}	Peta-	P	2^{50}	Pebi-	Pi
10^{18}	Exa-	E	2^{60}	Exbi-	Ei
10^{21}	Zetta-	Z	2^{70}	Zebi-	Zi
10^{24}	Yotta-	Y	2^{80}	Yobi-	Yi
10^{-3}	milli-	m	10^{-15}	femto-	f
10^{-6}	micro-	μ	10^{-18}	atto-	a
10^{-9}	nano-	n	10^{-21}	zepto-	z
10^{-12}	pico-	p	10^{-24}	yocto-	y

Memory Addressing (data bus)

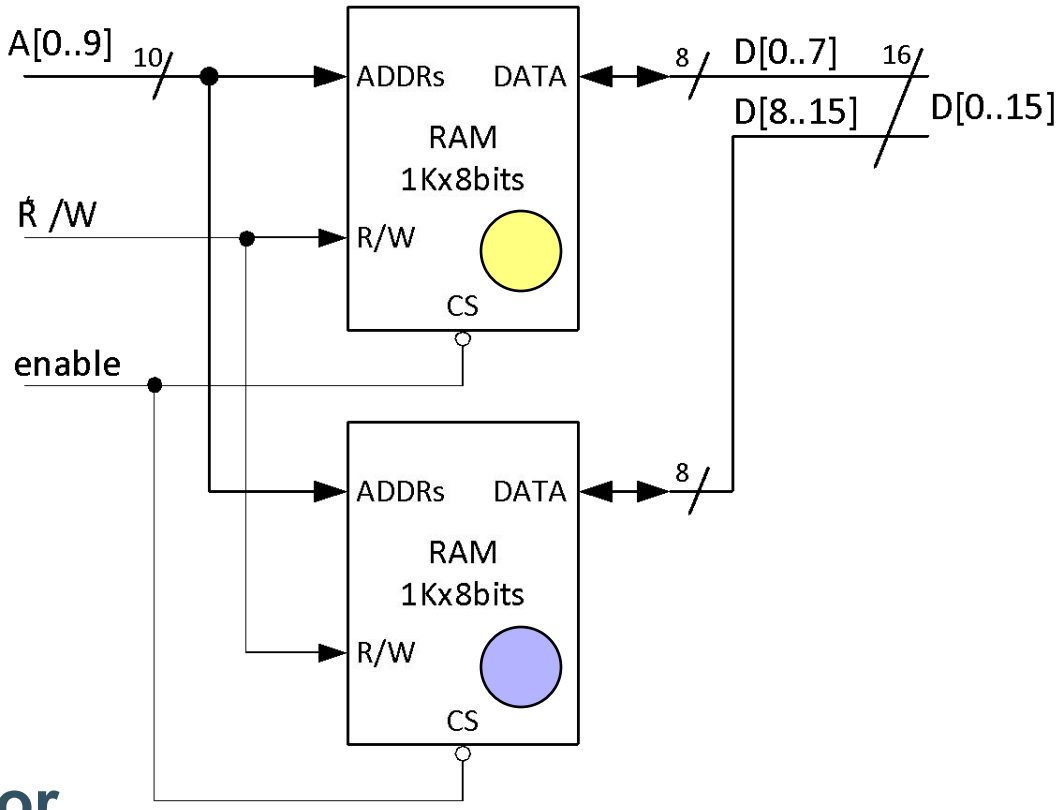


Memory Addressing (address bus)



Memory Addressing (parallel)

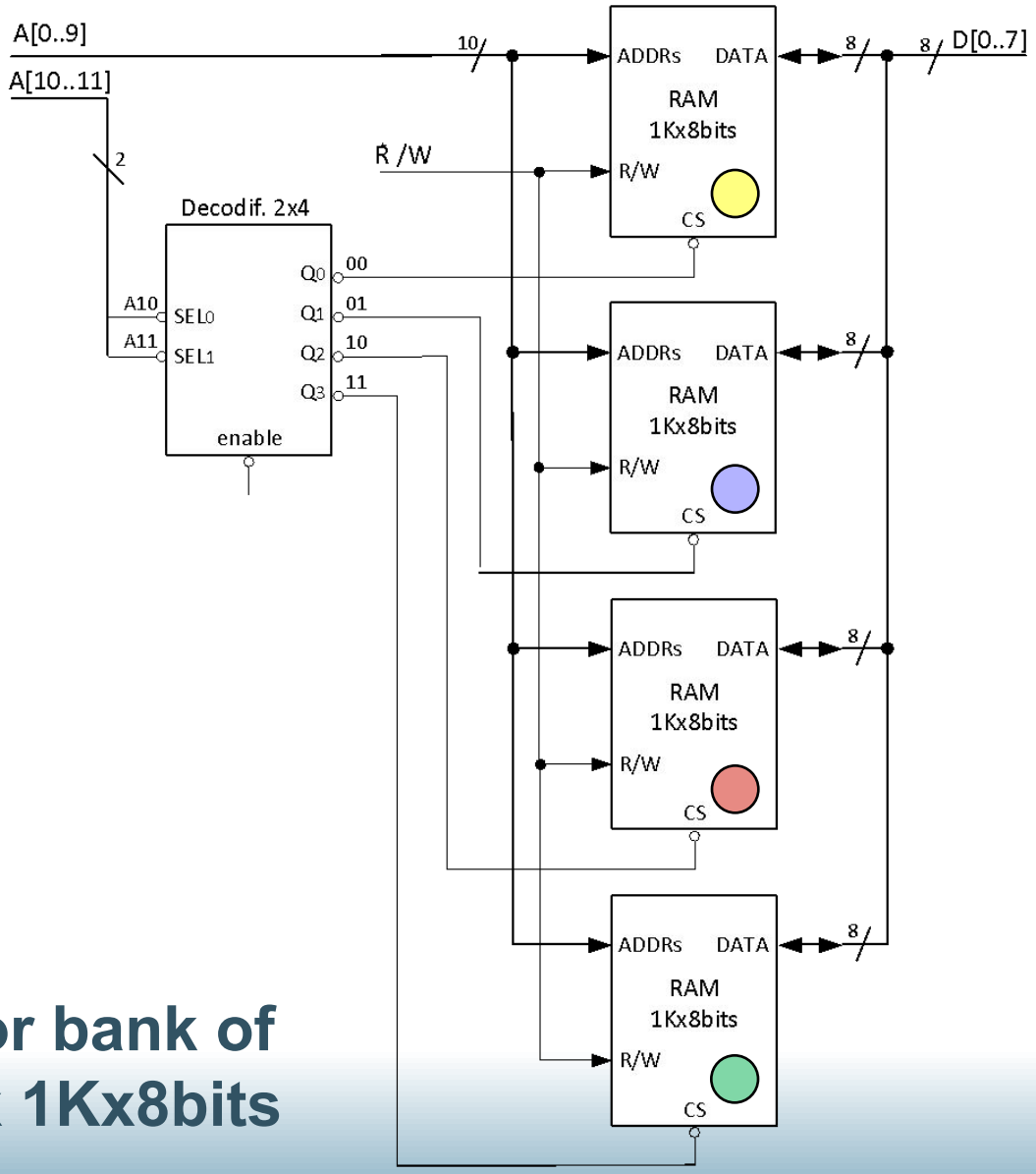
$A_9 \ldots A_0$	$D_{15} \ldots D_8$	$D_7 \ldots D_0$
0000000000	01.10.10	
0000000001	11.11.11	
0000000010	00.10.11	
	.....	.....
	.....	.....
1111111101	11.10.01	
1111111110	11.00.00	
1111111111	00.01.11	



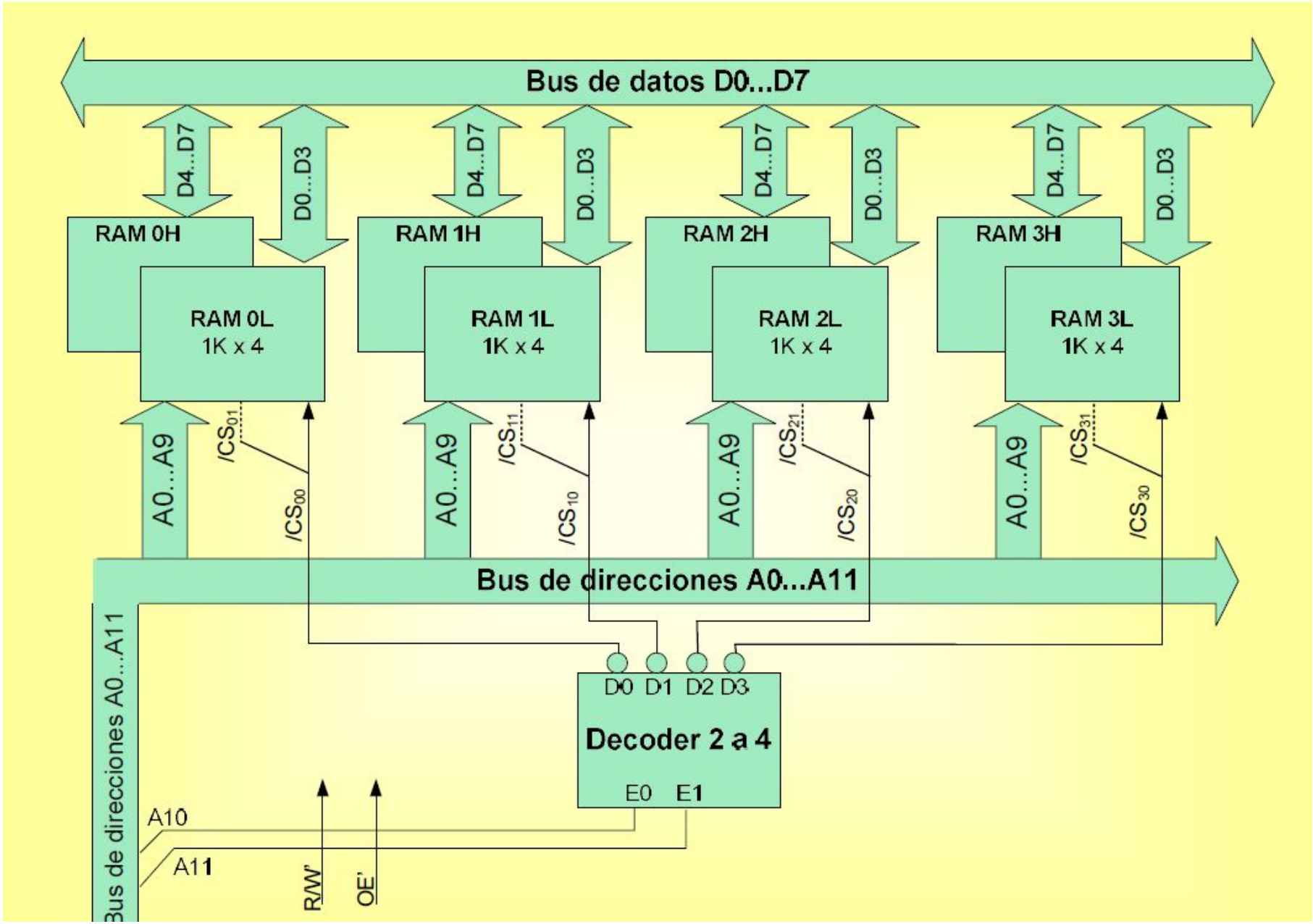
Example organization for bank of 1Kword x 16 bits from 2 x 1Kx8bits

Memory Addressing (serial)

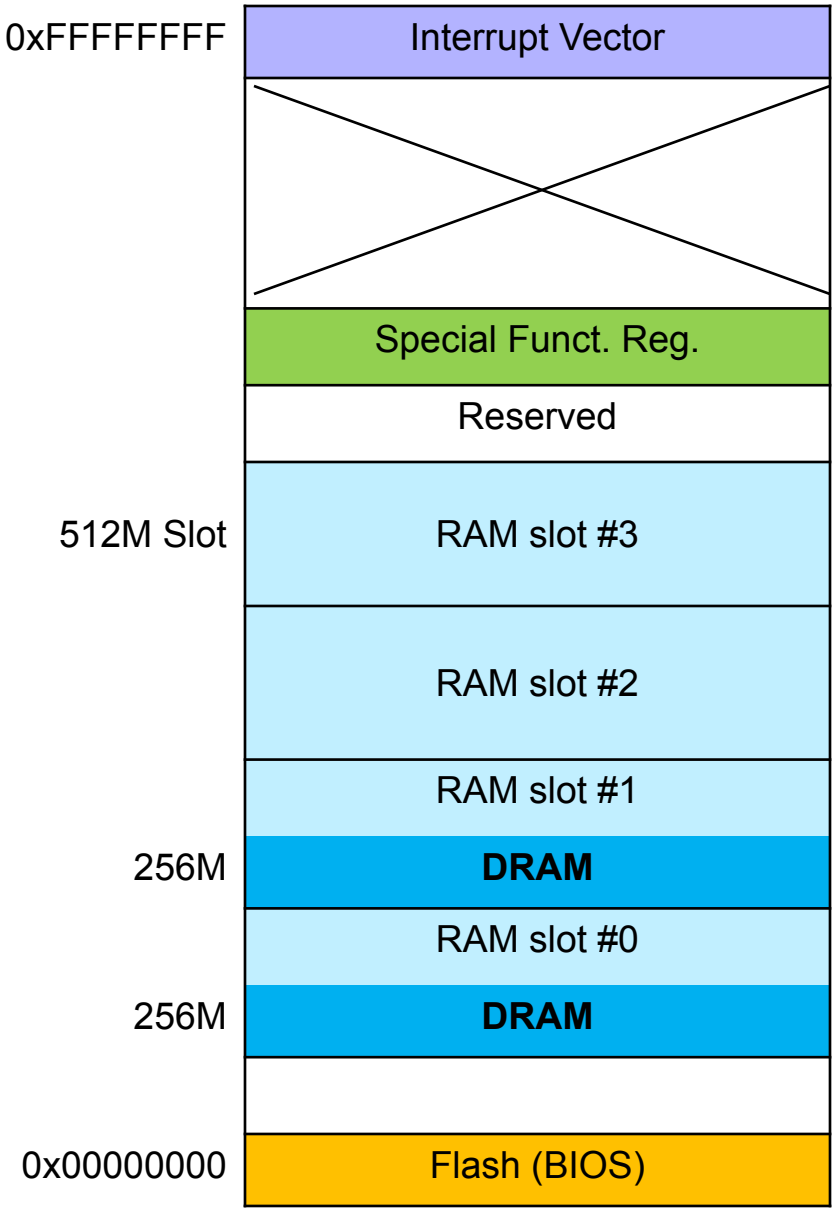
A_{11} A_{10}	A_9 . . . A_0	D_7 . . . D_0
0 0	0000000000	0 1 1 1 0 0 1 0
		. . . .
0 0	1111111111	0 0 0 1 0 1 1 1
0 1	0000000000	0 1 1 1 0 0 1 0
		. . . .
0 1	1111111111	0 0 0 1 0 1 1 1
1 0	0000000000	0 1 1 1 0 0 1 0
		. . . .
1 0	1111111111	0 0 0 1 0 1 1 1
1 1	0000000000	0 1 1 1 0 0 1 0
		. . . .
1 1	1111111111	0 0 0 1 0 1 1 1



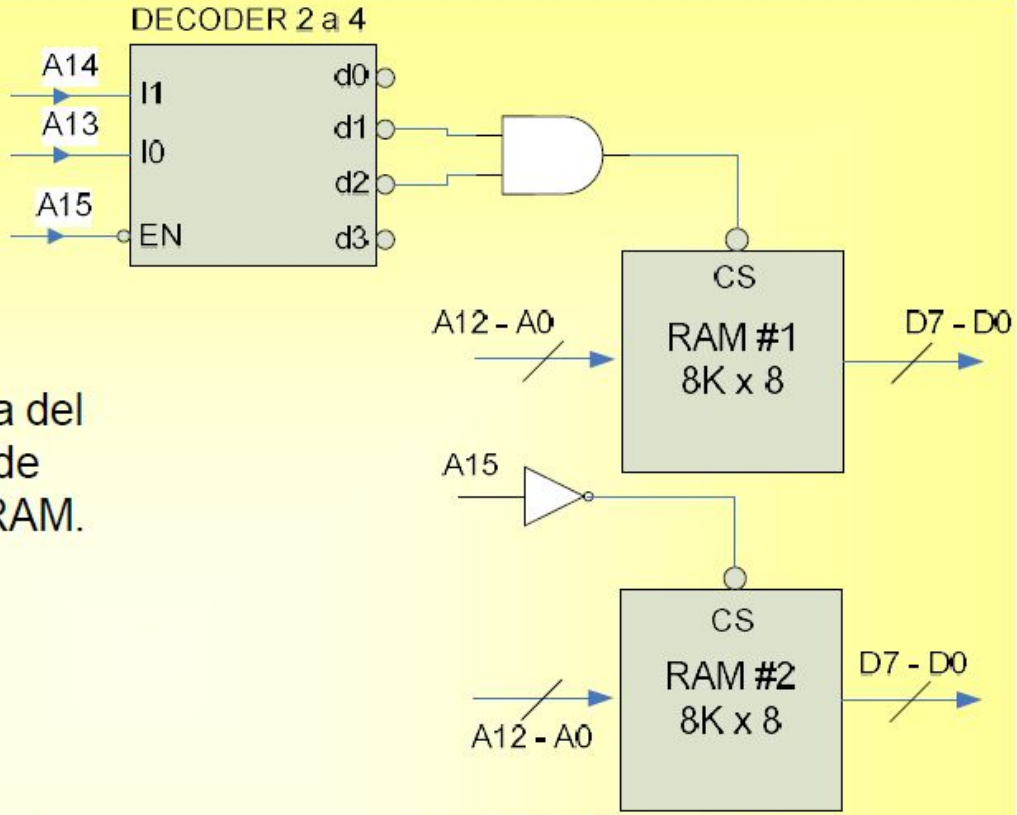
Example organization for bank of 4Kword x 8 bits from 4 x 1Kx8bits


Bus de datos D0...D7
D4...D7
D0...D3
RAM 0H
RAM 0L
1K x 4
RAM 1H
RAM 1L
1K x 4
RAM 2H
RAM 2L
1K x 4
RAM 3H
RAM 3L
1K x 4
A0...A9
/CS01
/CS00
/CS11
/CS10
/CS21
/CS20
/CS31
/CS30
Bus de direcciones A0...A11
Bus de direcciones A0...A11
D0 D1 D2 D3
Decoder 2 a 4
E0 E1
A10
A11
R/W'
OE'

Example: Conventional PC Memory map



‘Espejos’ en el mapa de memoria

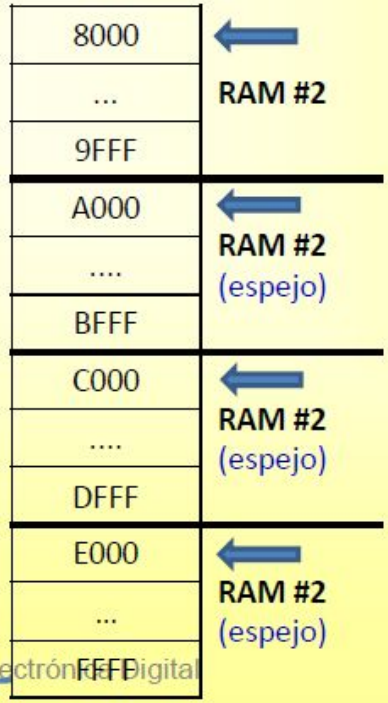


Caso: Hallar el mapa de memoria del circuito indicando las posiciones de memoria ocupadas por cada CI RAM.

A15	A14	A13	Decoder	Mem	Posición en el mapa	Bytes
0	0	0	d0	-	H0000 – H1FFF	8K
	0	1	d1	RAM1	H2000 – H3FFF	8K
	1	0	d2	RAM1	H4000 – H5FFF	8K
	1	1	d3	-	H6000 – H7FFF	8K
1	X	X		RAM2	H8000 - HFFFF	32K

Por ejemplo, a la primer posición de memoria de RAM #2 se accede indistintamente con las direcciones:

	A_{15} ...A_{12}	A_{11} ..... A_0
8000	1 0 0 **0**	**0000 0000 0000**
A000	1 0 1 **0**	**0000 0000 0000**
C000	1 1 0 **0**	**0000 0000 0000**
E000	1 1 1 **0**	**0000 0000 0000**



Bibliografía: M. Morris Mano, Charles R. Kime, and Tom Martin, "Logic and computer design fundamentals” 5th Edition (2015).