

Proiectare cu Microprocesoare

Curs 11 – Interfațarea memoriei

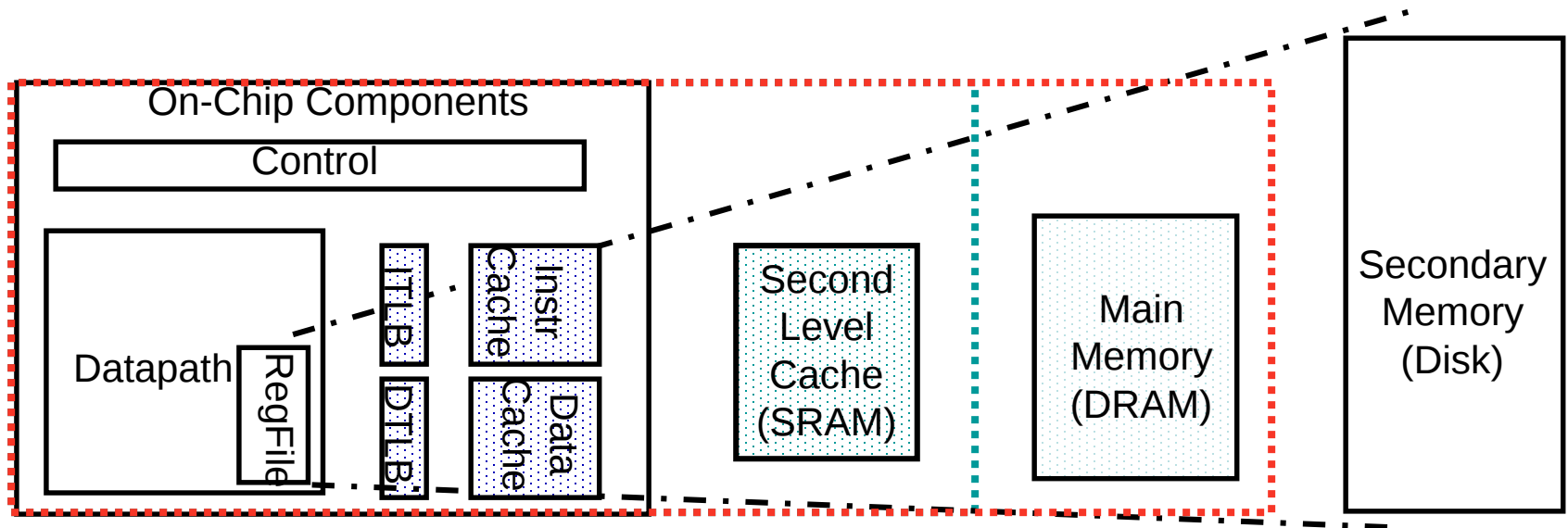
An 3 CTI

An universitar 2018/2019

Semestrul 1

Lector: Radu Danescu

Ierarhia tipică a memoriilor



Viteza (ns):	.1	1	10	100	1000
Dimensiune(bytes):	100	10K	100K ... M's	100M ... G	100G ... T
Cost/capacitate:	maxim				minim

Clasificarea memoriilor

Read Write (RWM)		NVRWM	ROM
Random Access	Non-Random Access	EPROM	Mask-prog. ROM
SRAM (cache, registri) DRAM (memoria principala)	FIFO, LIFO Registru de deplasare	EEPROM FLASH	Electrically- prog. PROM

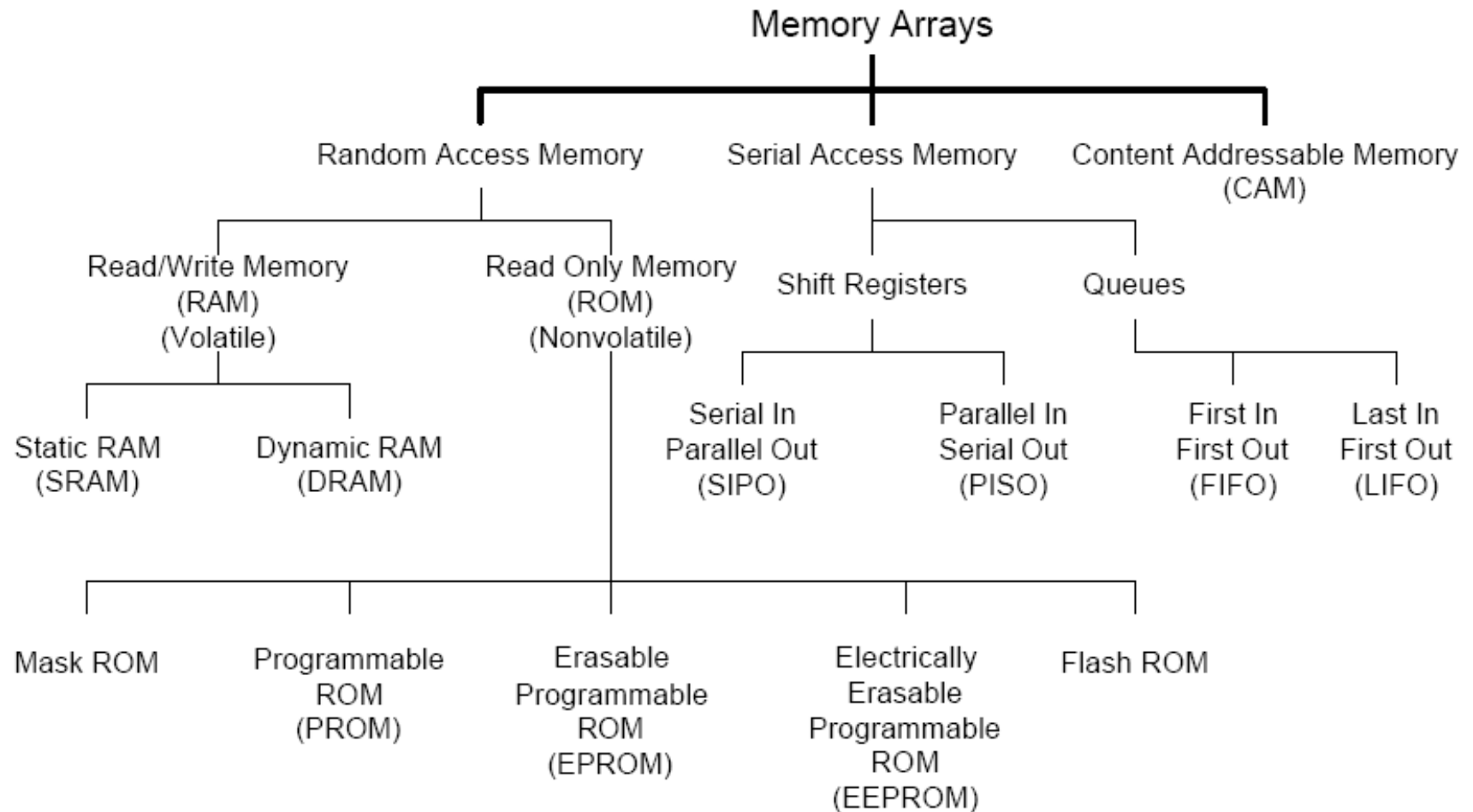
Clasificare după funcționalitate și mecanism de stocare

- Statică sau dinamică, volatilă sau nonvolatilă (NV); read only (ROM)

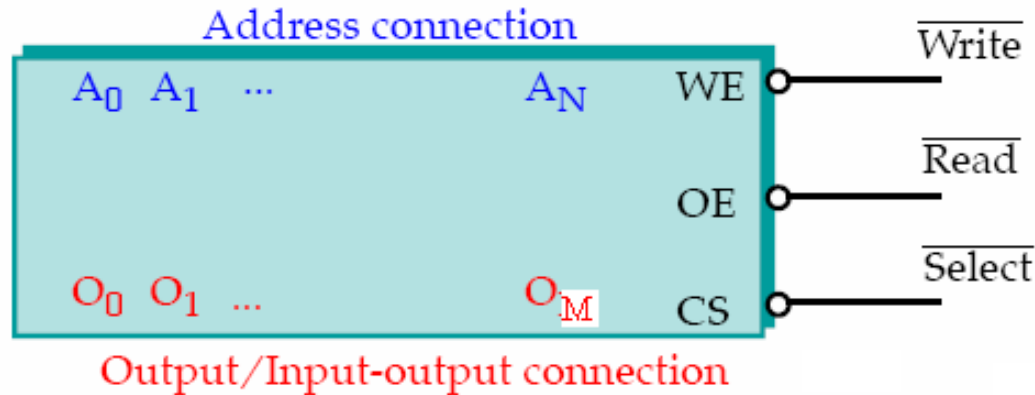
Clasificare după modul de acces

- aleator, serial, adresabil prin conținut

Clasificarea memoriilor



Configurația generală a pinilor



Pini de adresa

N pini de adresă $\Rightarrow$ numărul locațiilor de memorie = 2^N

Pini de date

M $\Rightarrow$ dimensiunea cuvântului din locația de memorie

Bidirecționali, ieșire 3-state (#OE)

Pini de control

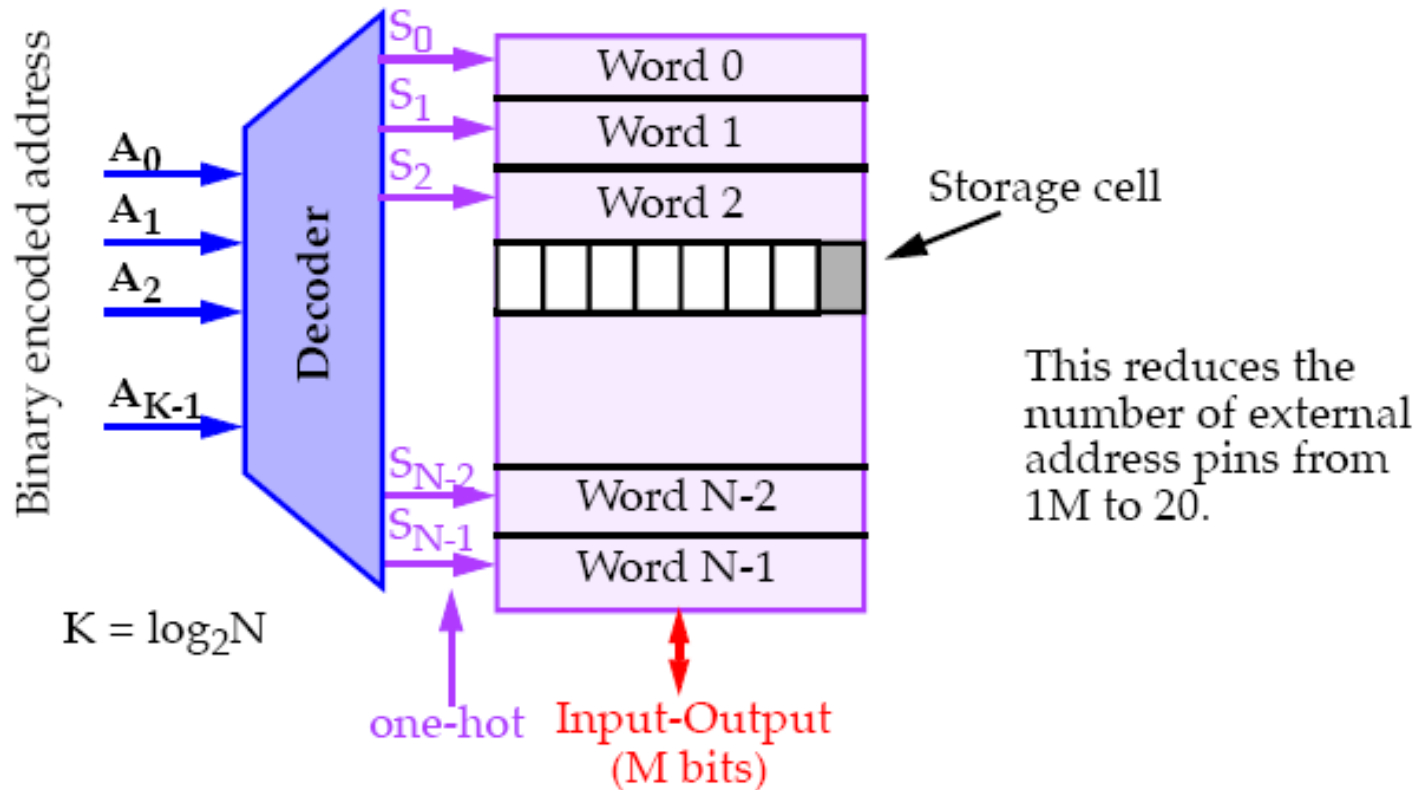
Chip select/enable $\Rightarrow$ activează dispozitivul

Read (#OE) /write (#WE) $\Rightarrow$ selecția operației

Arhitectura memoriei

Organizare liniară (1D)

Add a decoder to solve the package problem:



Nu se rezolvă problema raportului dintre lățimea și lungimea tabloului

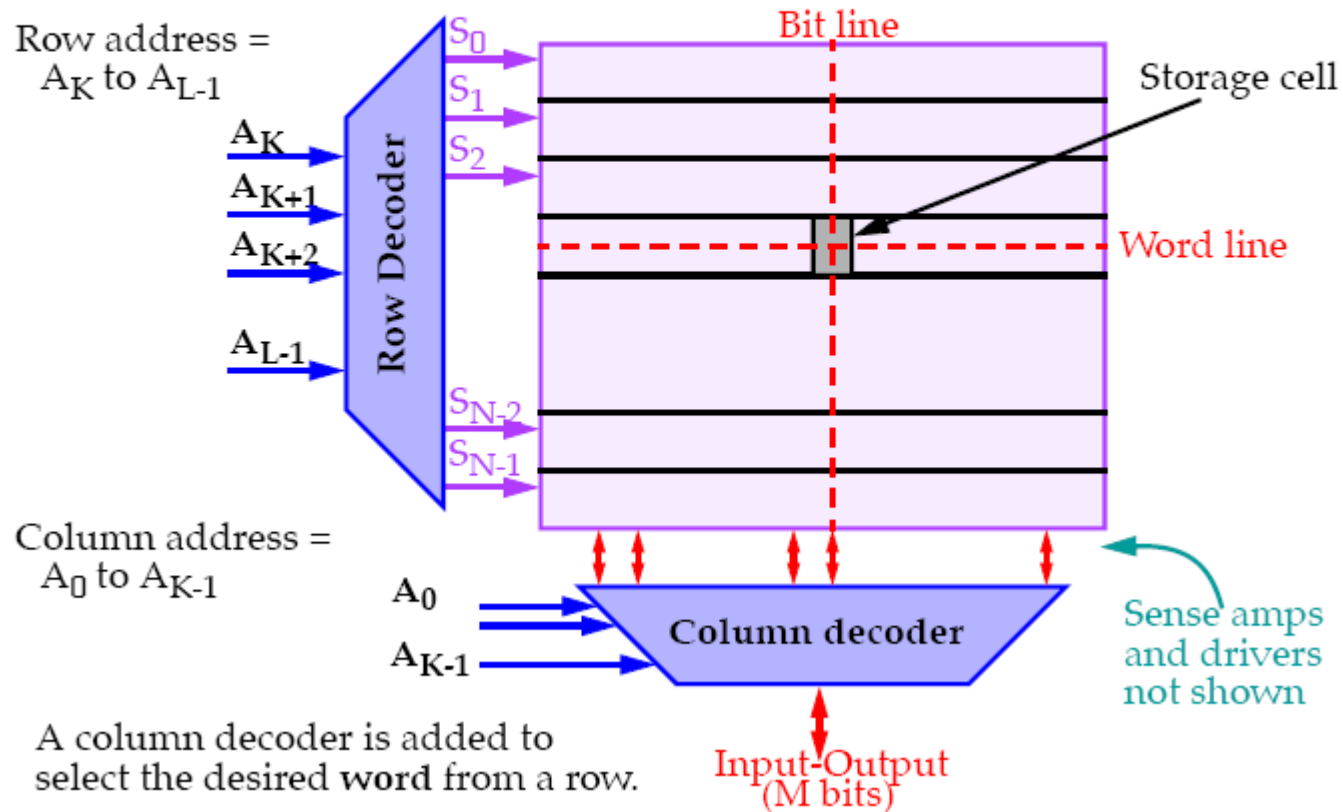
Această organizare este foarte lentă, deoarece firele verticale sunt foarte lungi

Arhitectura memoriei

Organizare 2-d

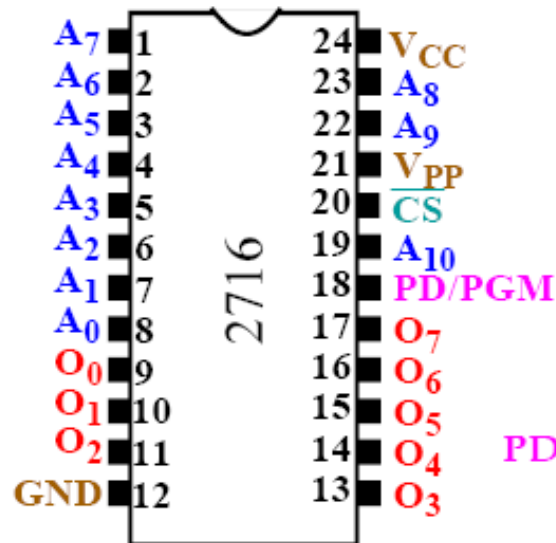
Dimensiunea verticală și cea orizontală sunt de obicei similare

Mai multe cuvinte sunt stocate pe aceeași linie



ROM/EPROM

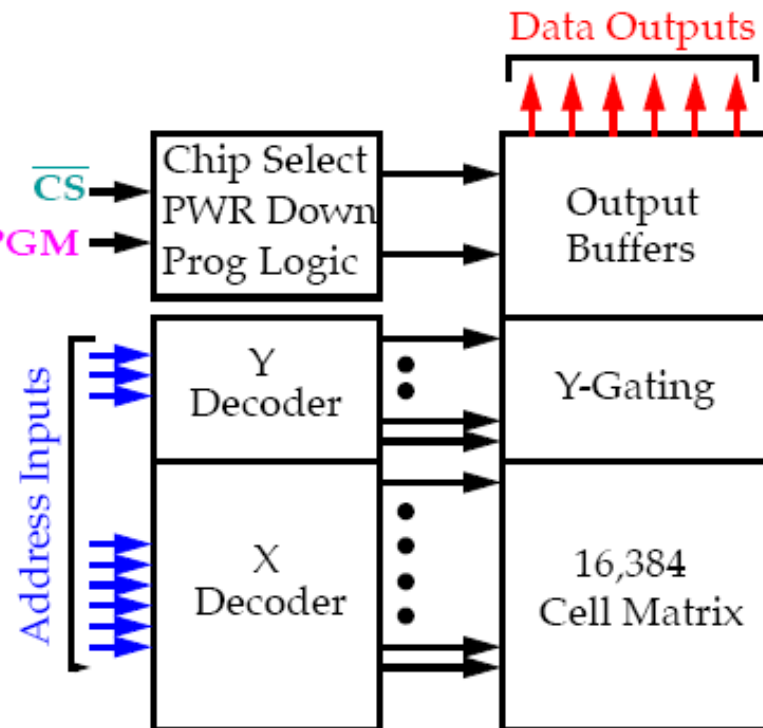
Intel 2716 EPROM (2K X 8):



2K x 8 EPROM

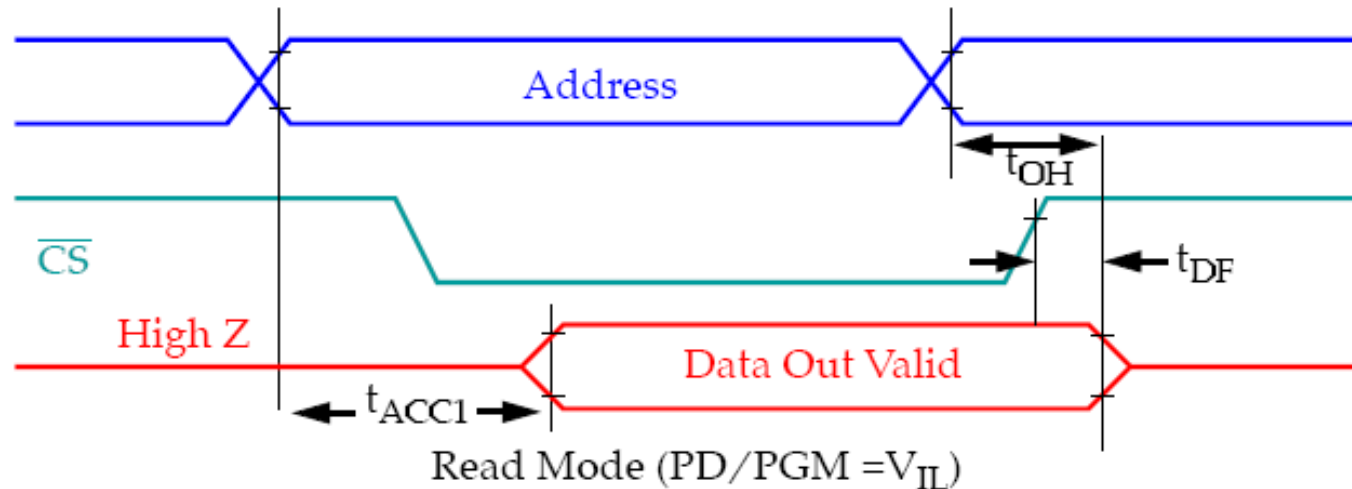
Pin(s)	Function
A_0-A_{10}	Address
PD/PGM	Power down/Program
$\overline{CS}$	Chip Select
O_0-O_7	Outputs

V_{PP} is used to program the device by applying 25V and pulsing PGM while holding $\overline{CS}$ high.



ROM/EPROM

2716 Timing diagram:

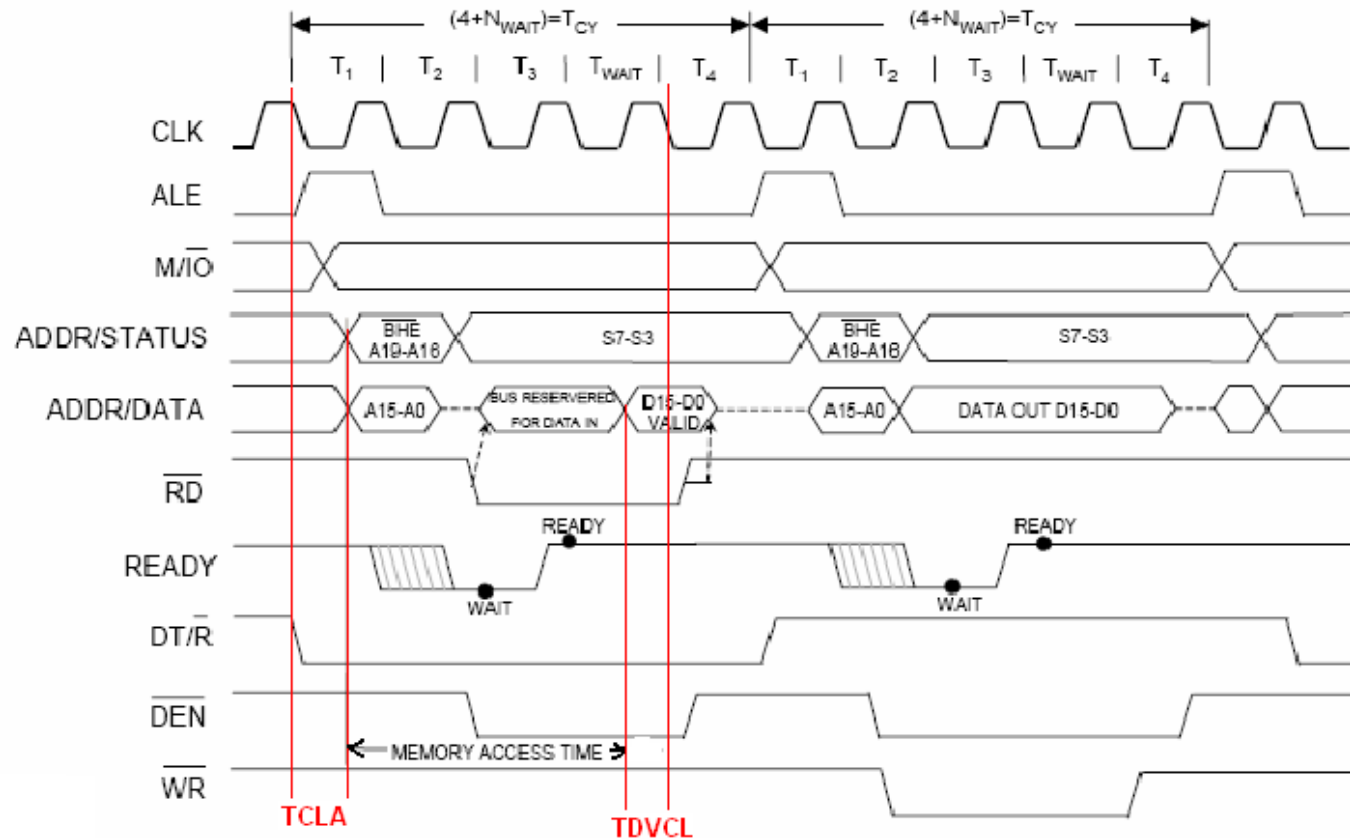


Sample of the data sheet for the 2716 A.C. Characteristics.

Symbol	Parameter	Limits			Unit	Test Condition
		Min	Typ.	Max		
t_{ACC1}	Addr. to Output Delay		250	450	ns	PD/PGM = $\overline{CS} = V_{IL}$
t_{OH}	Addr. to Output Hold	0			ns	PD/PGM = $CS = V_{IL}$
t_{DF}	Chip Deselect to Output Float	0		100	ns	PD/PGM = V_{IL}
...	...	...	...	...	...	...

This EPROM requires a wait state for use with the 8086 (460ns constraint).

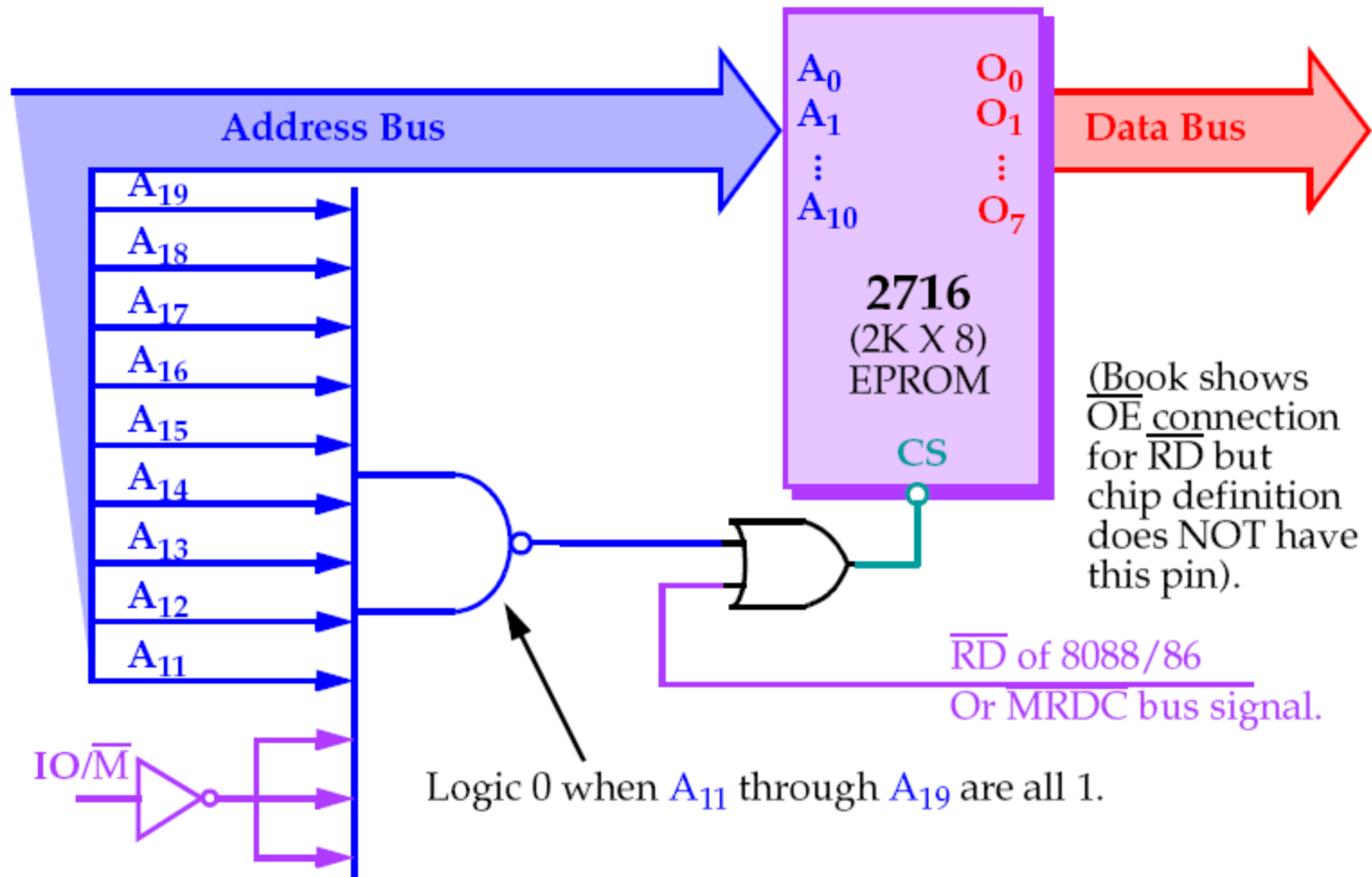
Temporizarea pe magistrală (8086)



Timpul maxim de acces la memorie = $3T - T_{CLAV} - T_{DVCL}$. De fapt timpul de acces trebuie să fie mai mic deoarece vor exista întârzieri de propagare

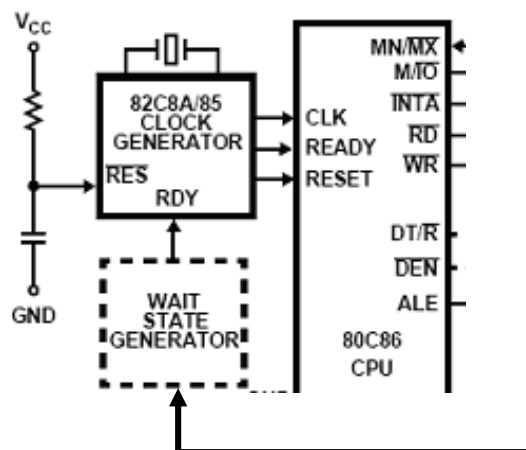
ROM/EPROM

Memory Address Decoding

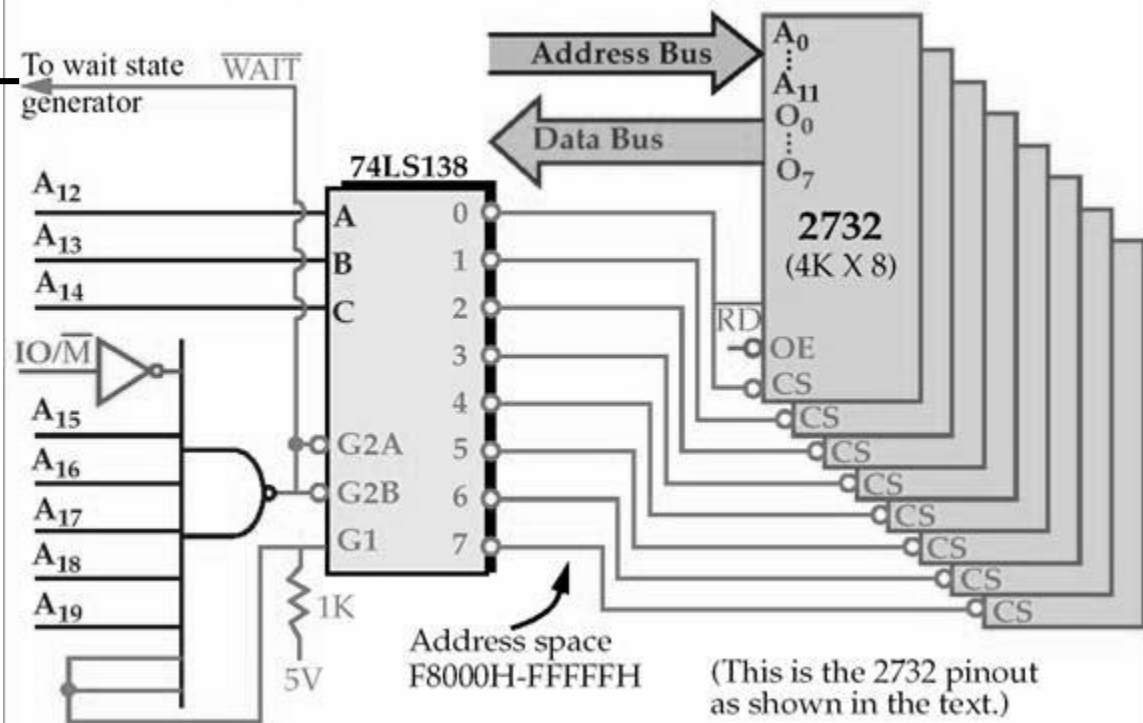


Domeniul de adrese: FF800h - FFFFFh

ROM/EPROM



8088 and 80188 (8-bit) EPROM Memory Interface



EPROM	Density (bits)	Capacity (bytes)
2716	16K	2K × 8
2732	32K	4K × 8
27C64	64K	8K × 8
27C128	128K	16K × 8
27C256	256K	32K × 8
27C512	512K	64K × 8
27C010	1M	128K × 8
27C020	2M	256K × 8
27C040	4M	512K × 8

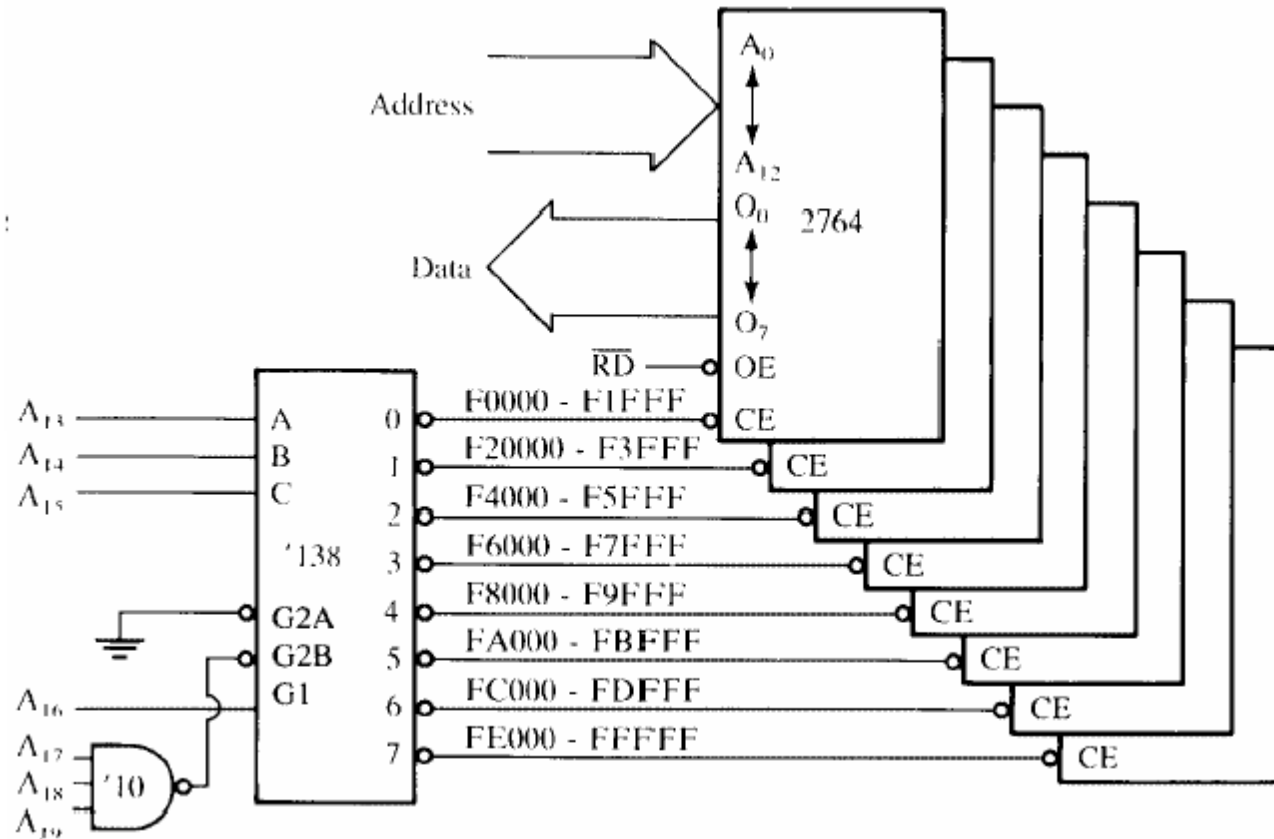
8088, min.mode, EPROM $8 \times 2732 = 8 \times 4\text{kB} = 32 \text{ KBytes}$

Organizarea adreselor: 0: F8000- F8FFF; 1: F9000- F9FFF; 7: FF00-FFFF

• acces EPROM (450ns) + intarziere decoder (12ns) > timp acces 8088

• se foloseste o stare de asteptare (1 clock = 200ns): timp total de acces 460+200=660 ns

ROM/EPROM



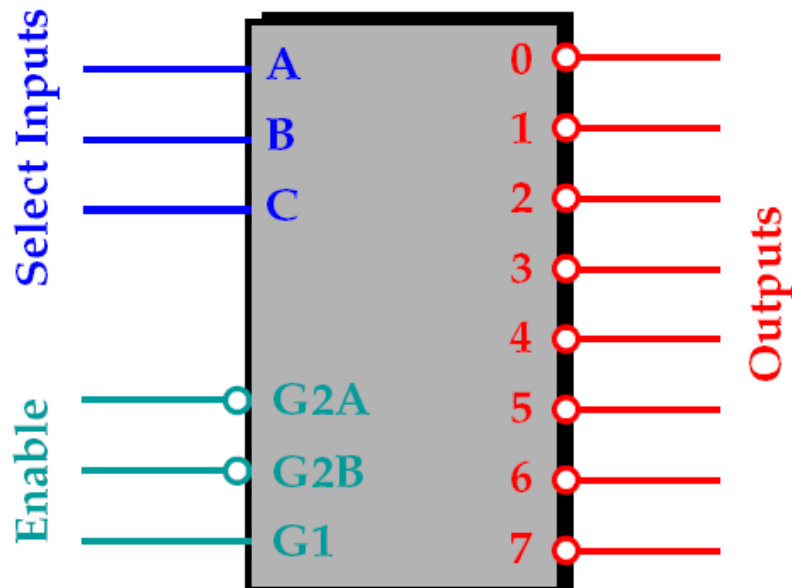
8088: $8 \times 2764 = 8 \times 8\text{KB} = 64 \text{ KB}$

Spatiul de adrese: F0000 – FFFFF

Decodificatoare de adrese

Memory Address Decoding

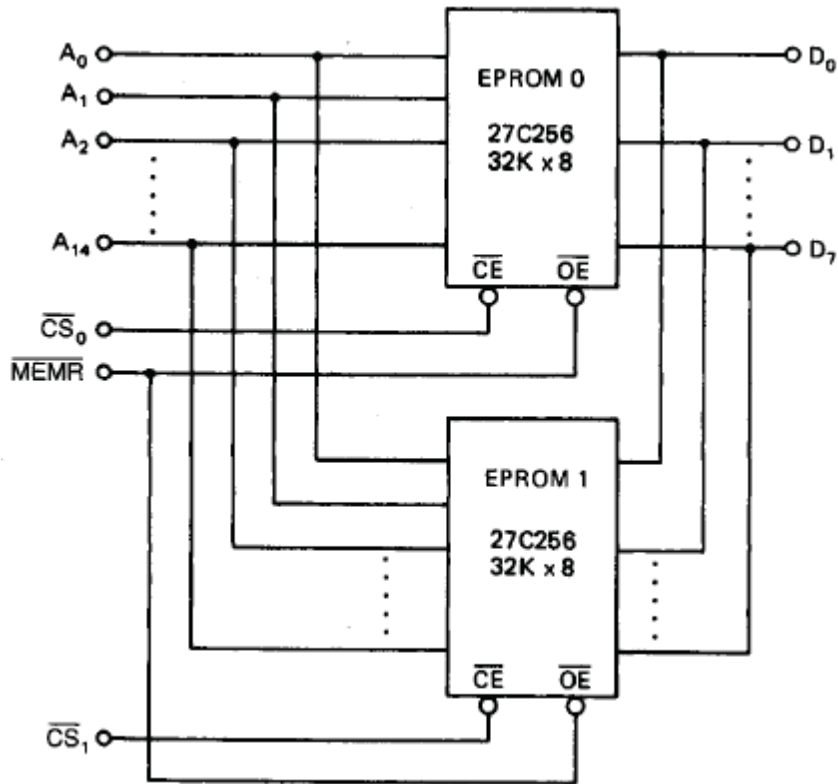
The 3-to-8 Line Decoder (74LS138)



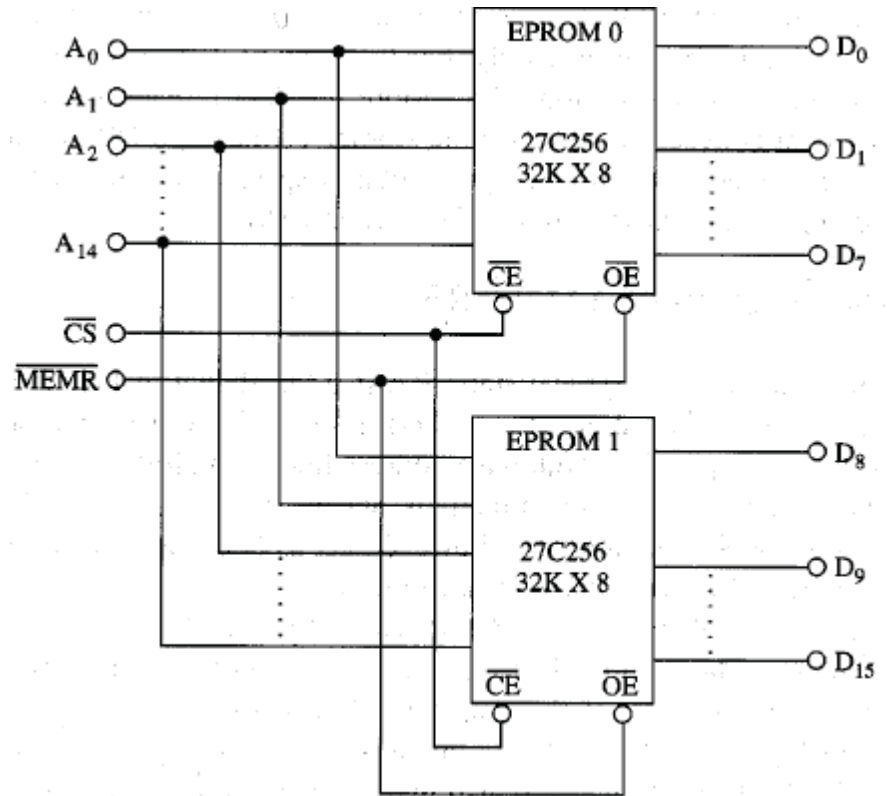
Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	0	1	2	3	4	5	6	7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Toate cele trei semnale Enable (G2A, G2B și G1) trebuie să fie active (0, 0 și 1)

Conectarea memoriilor

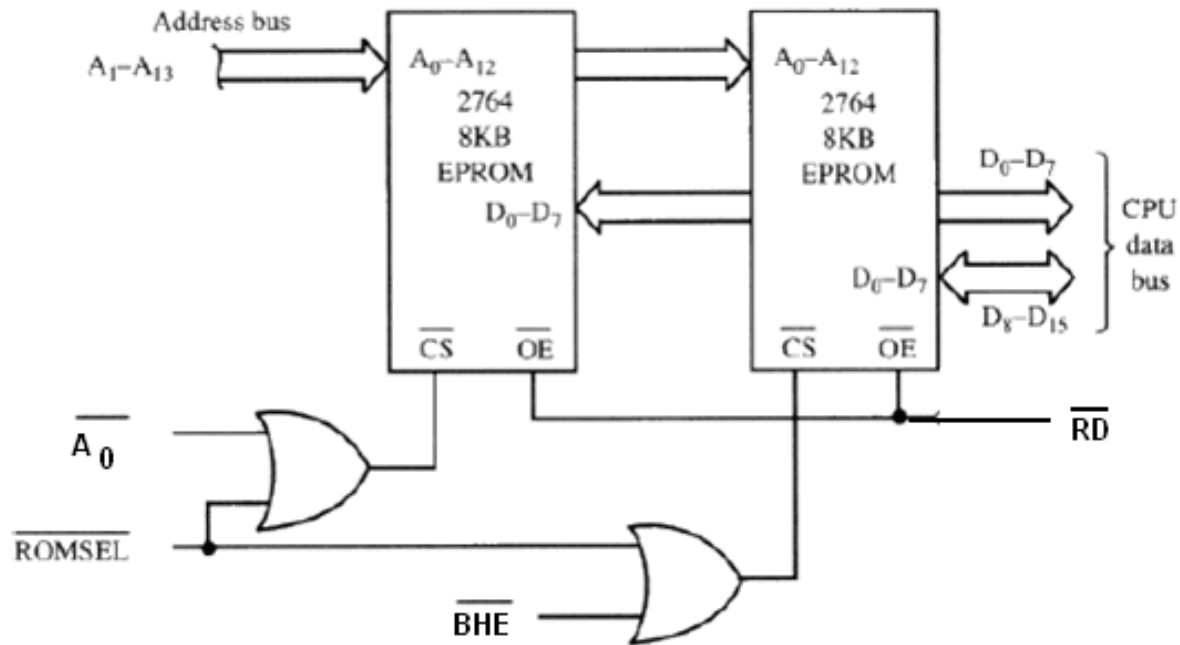


Conectarea paralela: extinde adâncimea memoriei



Conectarea seriala: extinde lățimea cuvântului

Conexiunea la o magistrala de 16 biti (8086)



$\overline{\text{BHE}}$	A_0	Characteristics
0	0	Whole word
0	1	Upper byte from/to odd address
1	0	Lower byte from/to even address
1	1	None

RAM

Static (SRAM)

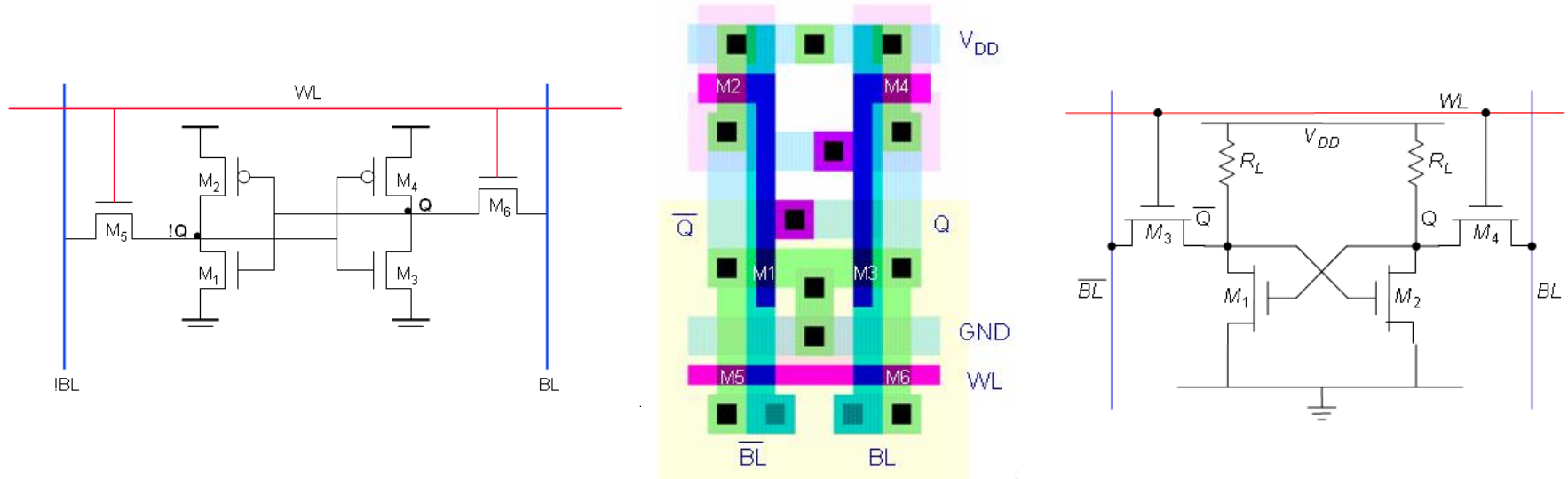


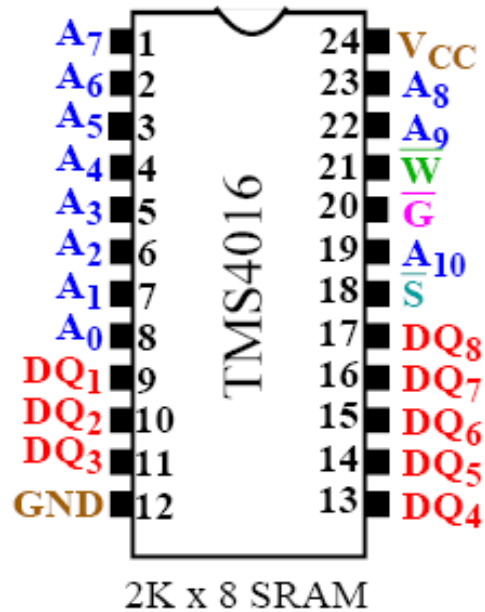
Table 12-2 Comparison of CMOS SRAM cells used in 1-Mbit memory (from [Takada91])

	Complementary CMOS	Resistive Load	TFT Cell
Number of transistors	6	4	4 (+2 TFT)
Cell size	58.2 μm^2 (0.7- μm rule)	40.8 μm^2 (0.7- μm rule)	41.1 μm^2 (0.8- μm rule)
Standby current (per cell)	10^{-15} A	10^{-12} A	10^{-13} A

SRAM

SRAMs

TI TMS 4016 SRAM (2K X 8):



Pin(s)	Function
A_0-A_{10}	Address
DQ_0-DQ_7	Data In/Data Out
$\overline{S}$ ($\overline{CS}$)	Chip Select
$\overline{G}$ ($\overline{OE}$)	Read Enable
$\overline{W}$ ($\overline{WE}$)	Write Enable

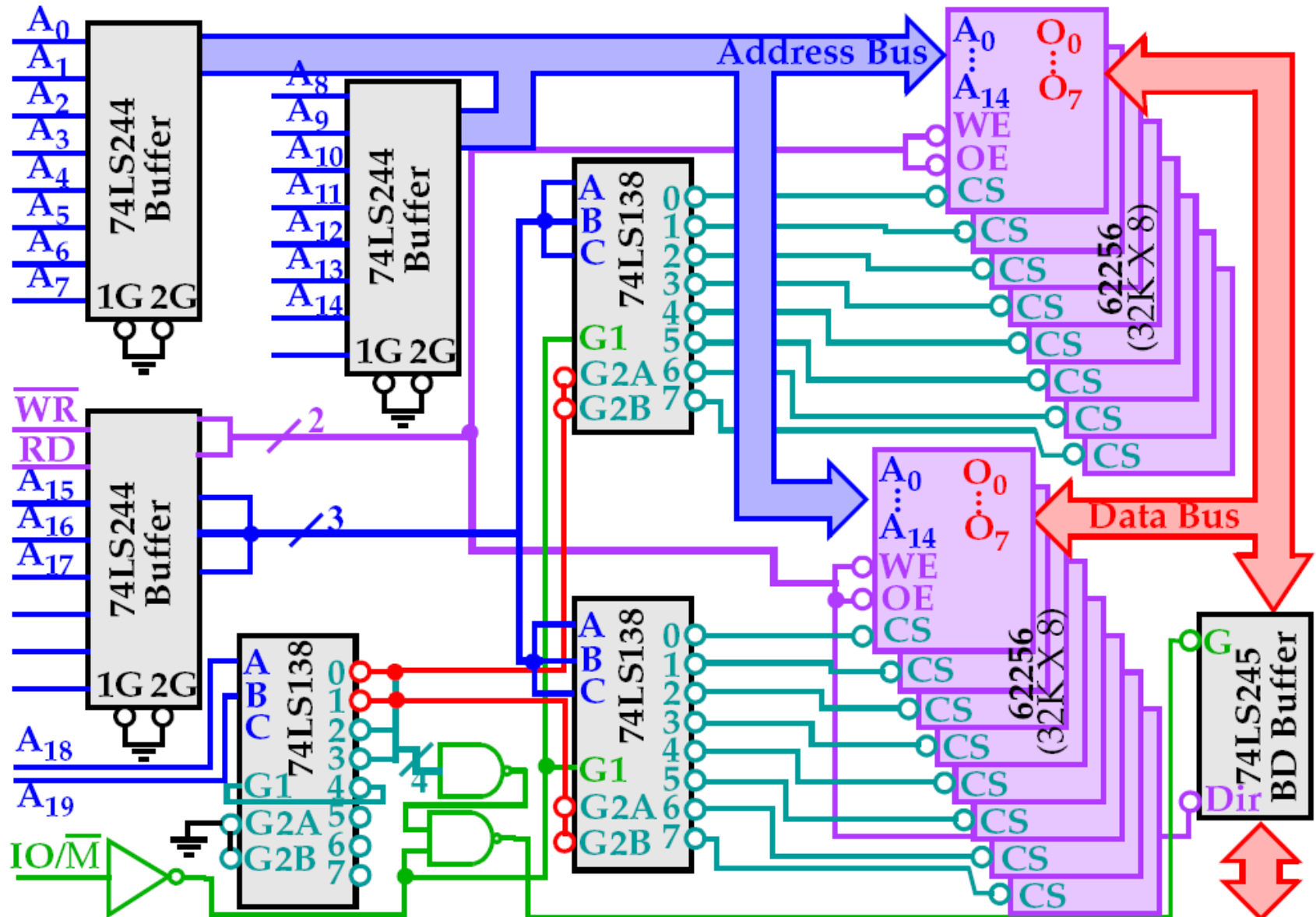
Pini similari cu cei ai EPROM-ului, cu excepția semnalului de scriere

Timp de acces mai rapid

Folosit pentru memorii Cache

SRAM

8088 and 80188 (8-bit) RAM Memory Interface



Conectare cu magistrale de 16, 32 biți

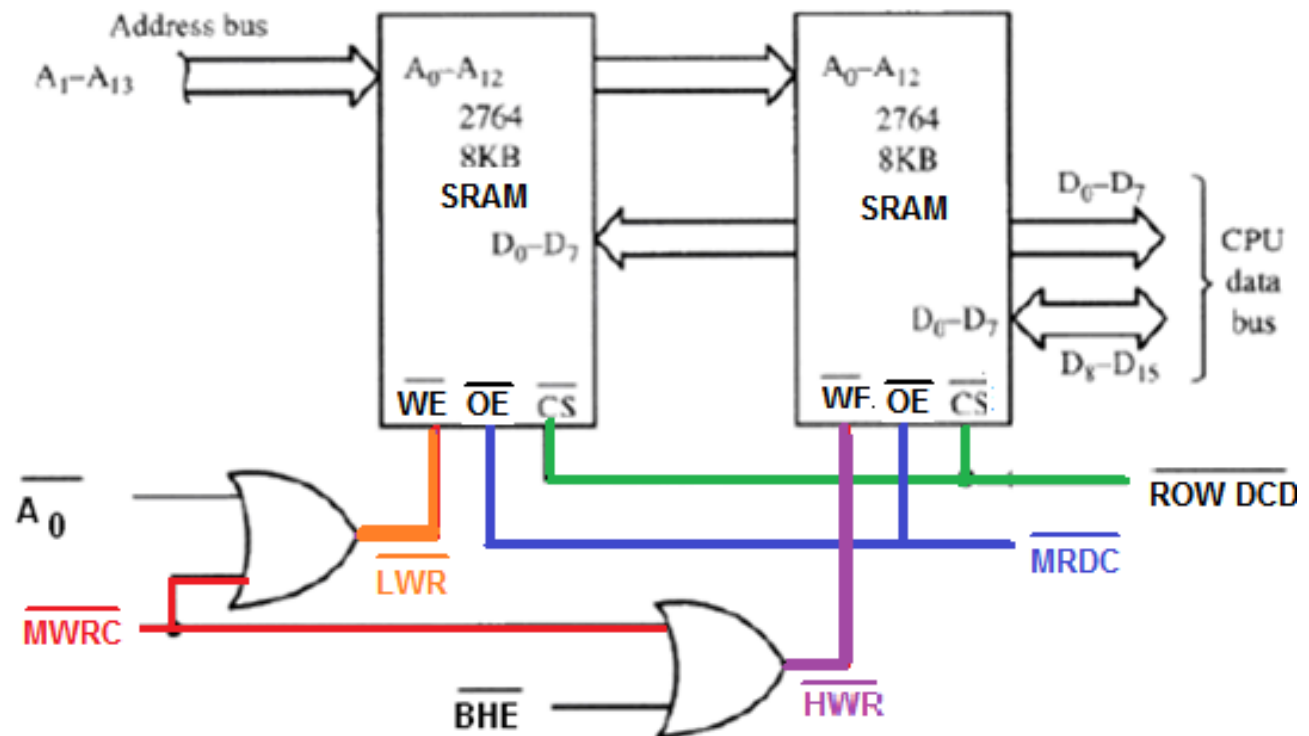
Nu există probleme la citire:

- Memoria va pune pe magistrală tot cuvântul (16 biți, 32 biți ...)
 - CPU citește byte-ul sau cuvântul dorit (BL, BH, BH+BL...)
- (Ex: MOV AL, mem ; MOV AH, mem; MOV AX, mem)

Scrierea trebuie tratată separat pentru fiecare byte:

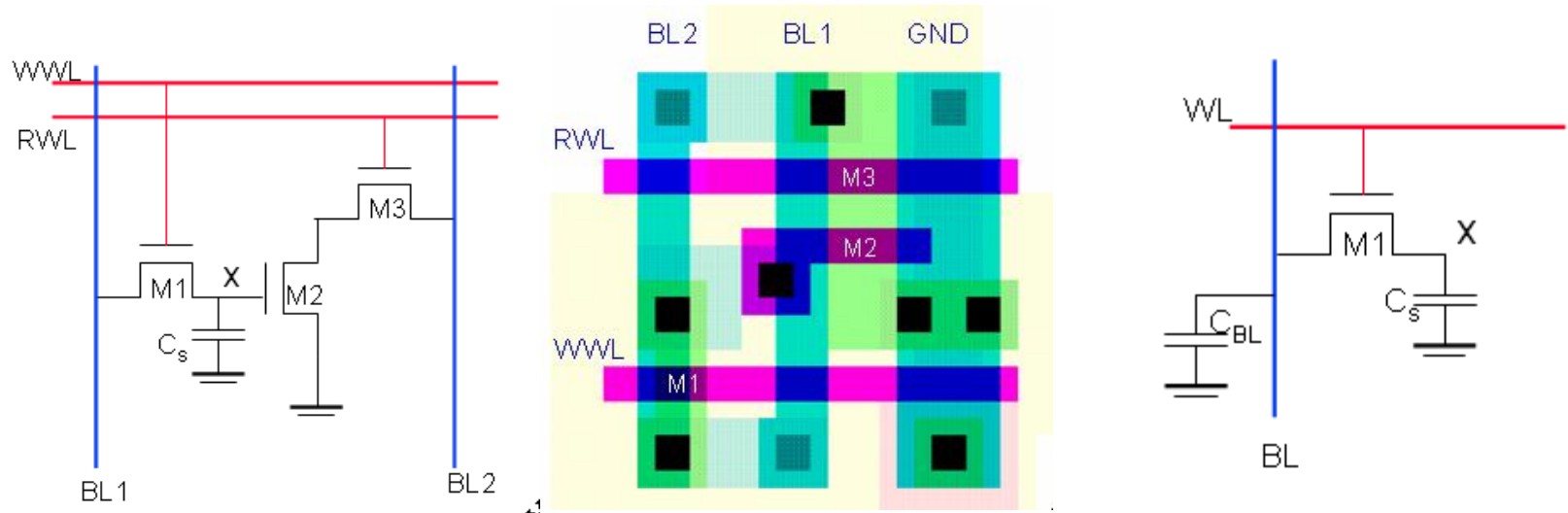
- LWR, HWR

Exemplu:



DRAM

Dynamic (DRAM)



1

Dimensiune: $\frac{1}{2}$ din dimensiunea celulei SRAM $\Rightarrow$ capacitate sporită $\Rightarrow$ pinii de adresă sunt multiplexați

Refresh (reîmprospătare): 1 .. 4 msec $\Rightarrow$ circuit dedicat

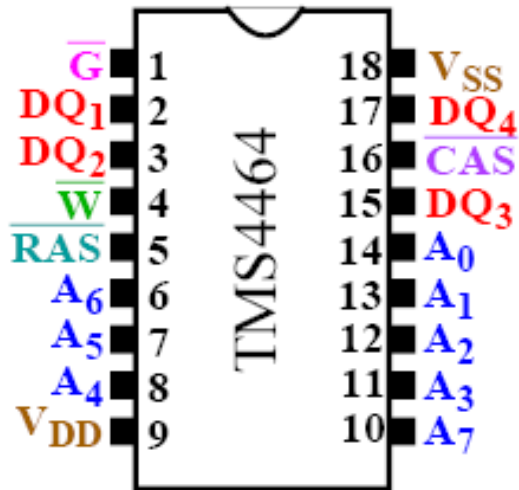
cicluri de citire, scriere, refresh

Variații: SDRAM, DDRAM, Rambus, ...

DRAM

DRAMs

TI TMS4464 DRAM (64K X 4):



64K x 4 DRAM

Pin(s)	Function
A_0-A_7	Address
DQ_0-DQ_4	Data In/Data Out
$\overline{RAS}$	Row Address Strobe
$\overline{CAS}$	Column Address Strobe
$\overline{G}$	Output Enable
$\overline{W}$	Write Enable

TMS4464 poate stoca 256K biți de date

64 locații adresabile – necesită 16 linii de adresă, dar are numai 8

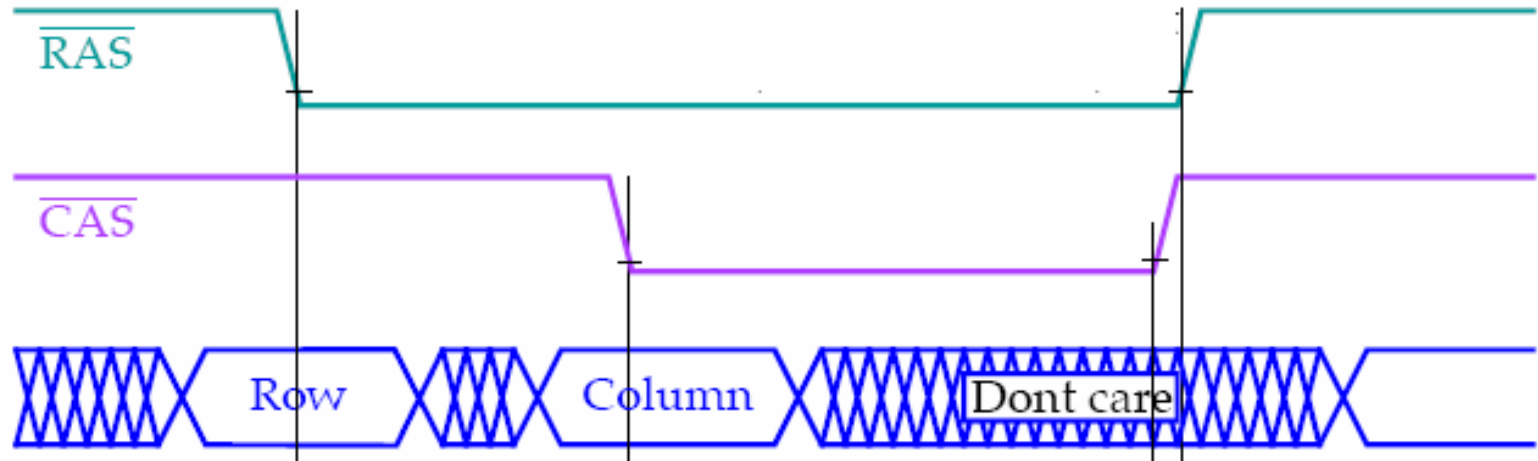
Adresa rândului ($A_8:A_{15}$) este plasată pe pinii de adresă, și memorată în latch-uri interne, folosind semnalul RAS

Adresa coloanei ($A_0:A_7$) este memorată ulterior, folosind semnalul CAS

DRAM

DRAMs

TI TMS4464 DRAM (64K X 4) Timing Diagram:



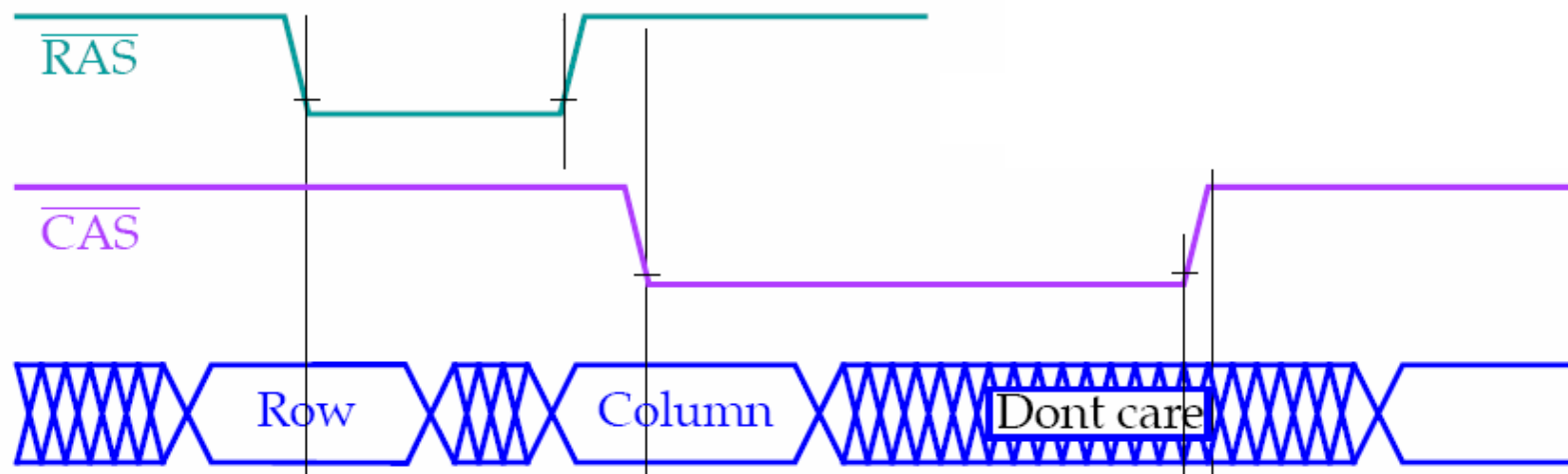
CAS also performs the function of the chip select input.

#RAS & #CAS trebuie transmise de un controller DRAM

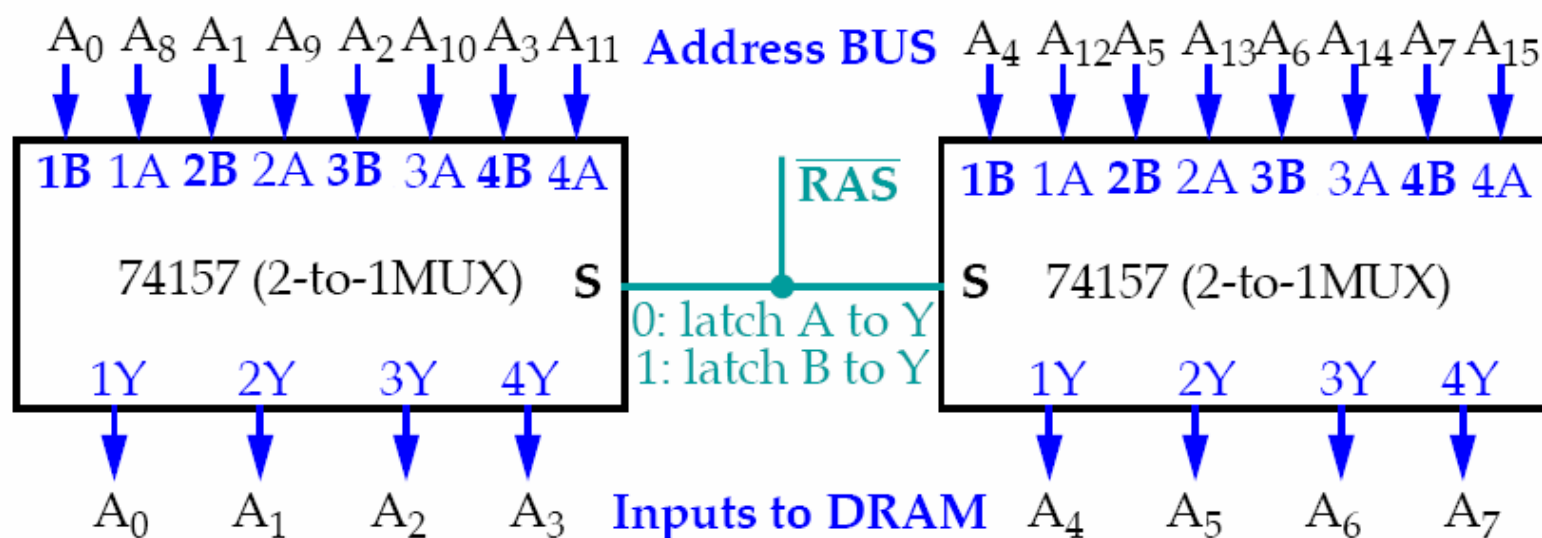
Controllerul DRAM trebuie să multiplexeze (în timp) liniile de adresă, ca:

- Adresa rândului (ex. A8-15)
- Adresa coloanei (ex. A0-7)

DRAM

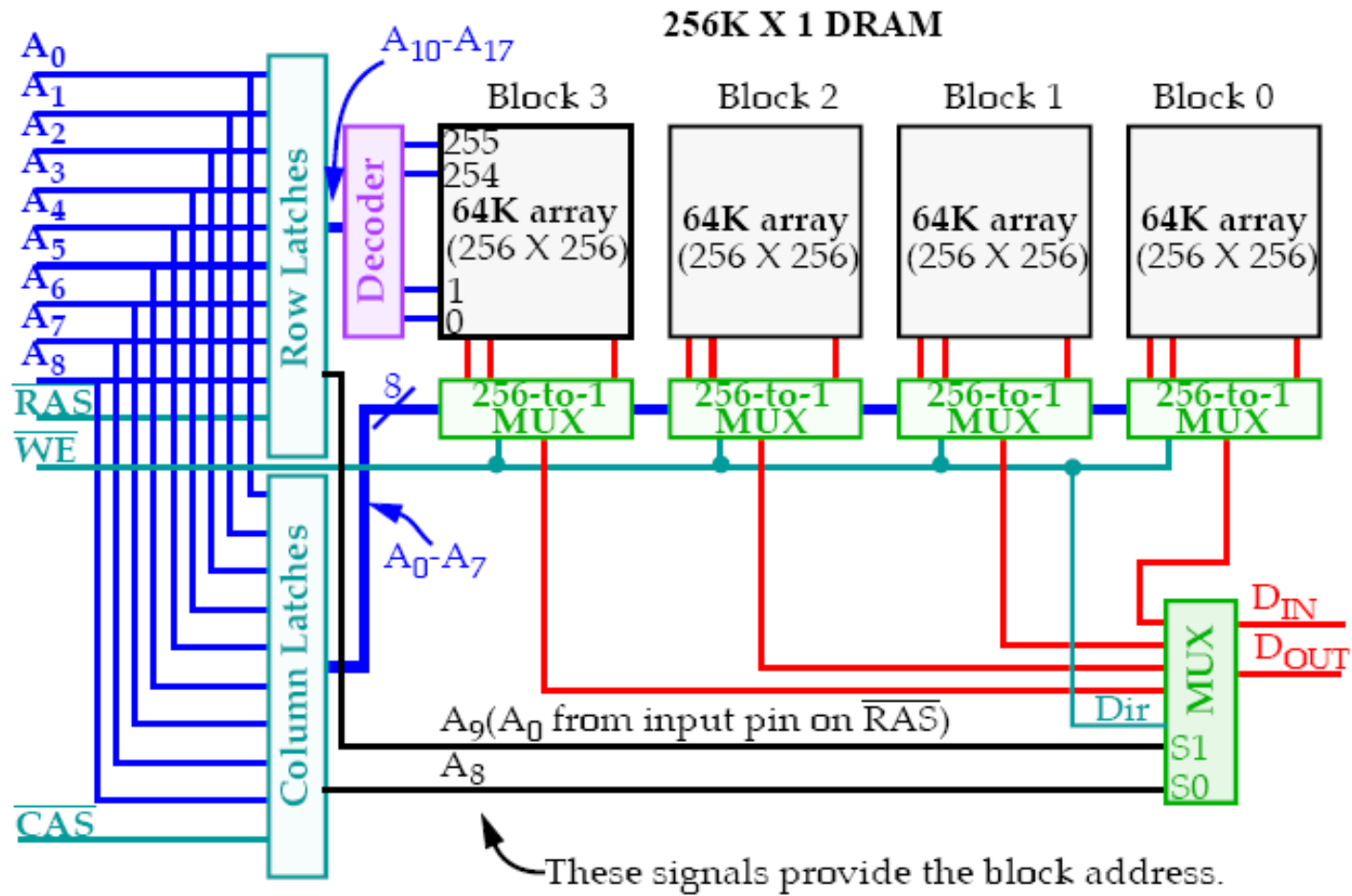


$\overline{\text{CAS}}$ also performs the function of the chip select input.



256K X 1 DRAM – structura internă

Dynamic RAM



DRAM refresh

Ciclu special de refresh

- Se petrece transparent când sunt folosite alte componente –*transparent refresh* sau *cycle stealing*.
- Un ciclu ce folosește doar RAS încarcă o adresă a rândului în DRAM
- Condensatorii rândului selectat sunt reîncărcați prin citirea internă a biților și scrierea lor înapoi.

Exemplu:

256K X 1 DRAM (256 rânduri x 256 coloane x 4 blocuri)

⇒ refresh este necesar la fiecare $15.6\mu s$ ($4ms/256$).

Pentru 8086, o citire sau o scriere se petrece la fiecare $800ns$ ($4 \times 200 = 4 \times T_{clk}$).

⇒ **19** citiri/scrieri per refresh ($15.6\mu s / 0.8\mu s = 19.5$)

⇒ Ciclurile de citire /scriere iau **5%** din timpul de refresh

Controllere DRAM

⇒ Multiplexarea adreselor si generarea semnalelor de control pentru DRAM.

Intel 82C08, poate controla doua bancuri de memorie DRAM 256K X 16 , pentru un total de 1 MB.

Biții A1 - A18 (18 biți) sunt conectați la intrările (AL - coloană) si (AH - linie) ale 82C08.

În funcție de adresă, se activează **RAS0/CAS0** sau **RAS1/CAS1**.

WE, **BHE** si **A0** sunt folosiți pentru determinarea scrierii, și unde anume se va scrie.

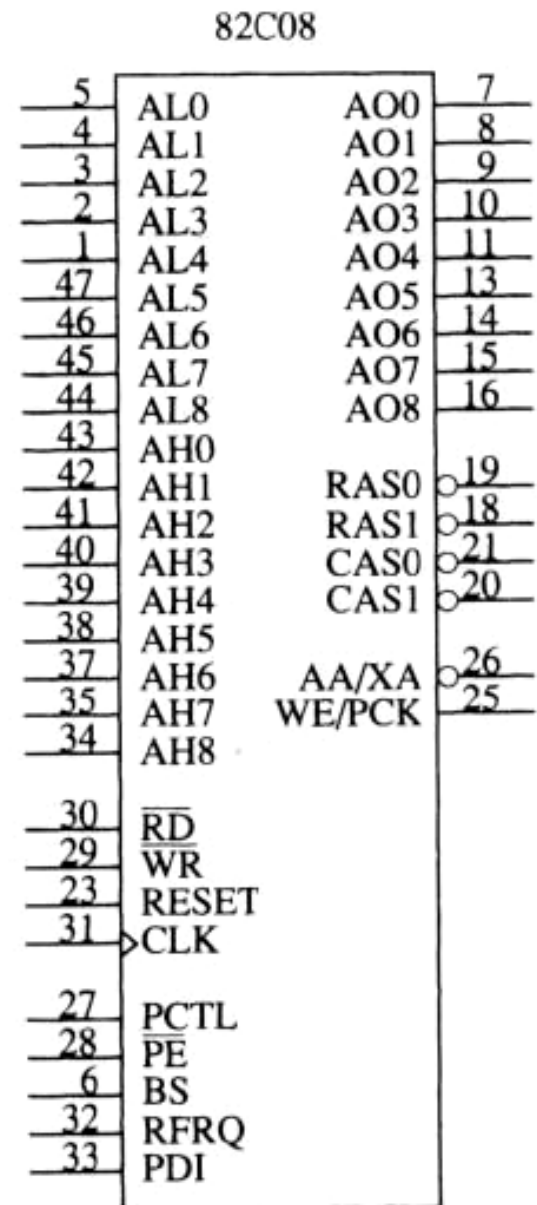
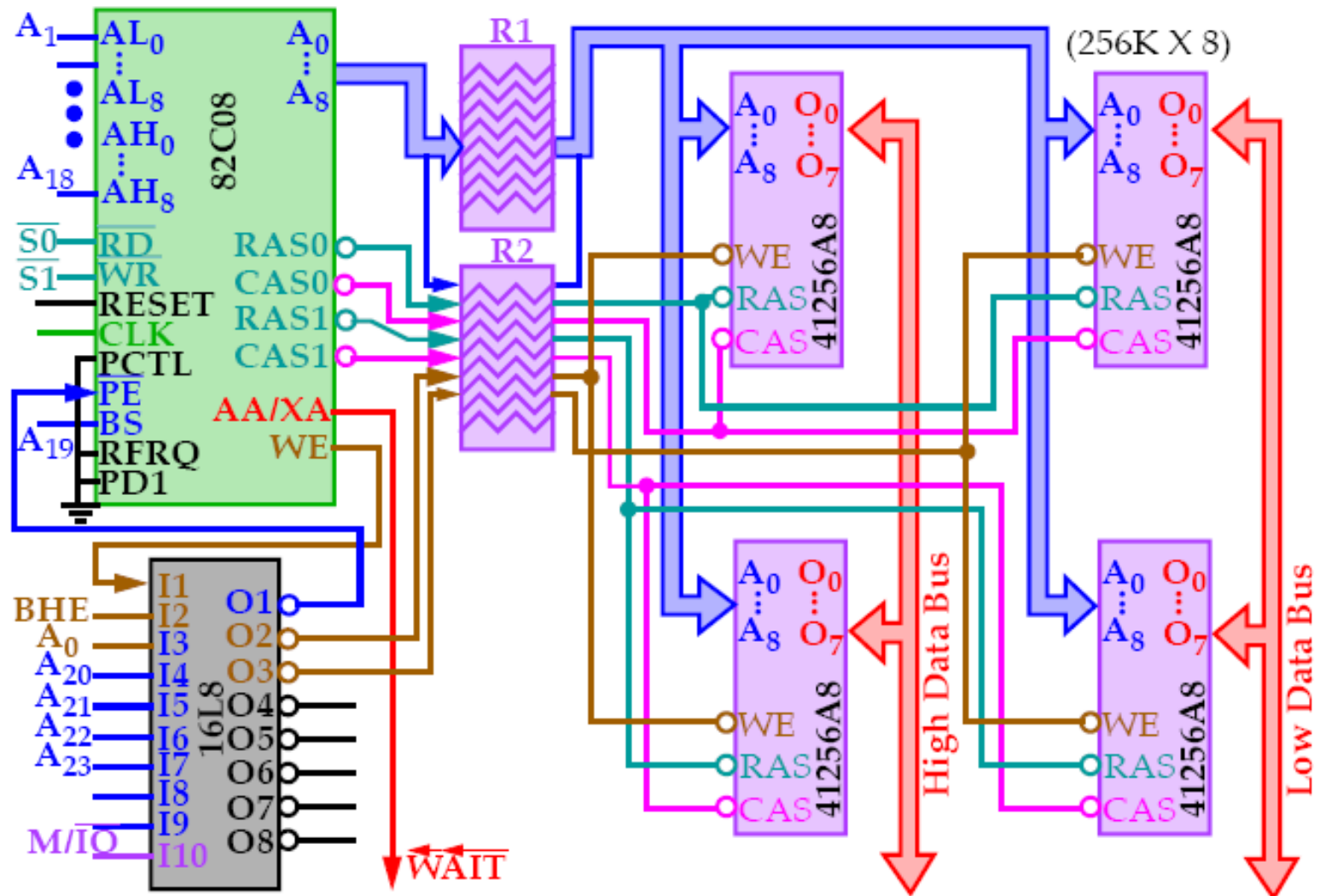


FIGURE 10-41 The 82C08 DRAM controller that controls two banks of memory.

Controllere DRAM

DRAM Controllers



Folosirea PAL pentru logica interfațării memoriei

Decodificarea adresei:

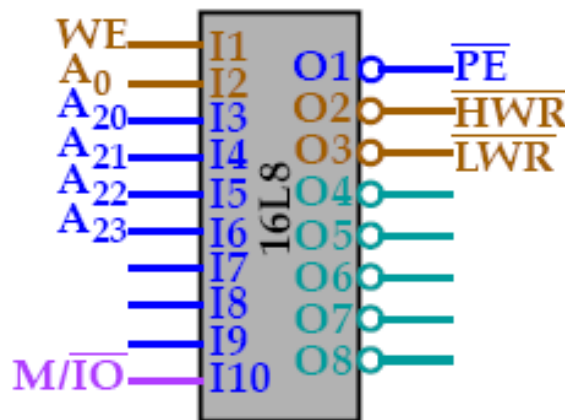
- Controllerul DRAM este activat atunci când liniile superioare de adresă indică spațiul DRAM (de obicei începe de la 0)
- PE (port enable) va fi activat de logica de decodificare a adresei

Comandă scriere octet superior sau inferior:

- Low write (LWR), High write (HWR)

Un singur chip 16L8 poate implementa toate aceste funcții:

16L8 Programming:



pins	1	2	3	4	5	6	7	8	9	10
	WE	BHE	A0	A20	A21	A22	A23	NC	NC	GND
pins	11	12	13	14	15	16	17	18	19	20
	MIO	CE	NC	NC	NC	NC	LWR	HWR	PE	VCC

Equations:

$$/LWR = /A0 * /WE$$

$$/HWR = /BHE * /WE$$

$$/PE = /A20 * /A21 * /A22 * /A23 * MIO$$

Controllere DRAM

