Calcolatori Elettronici T
Ingegneria Informatica

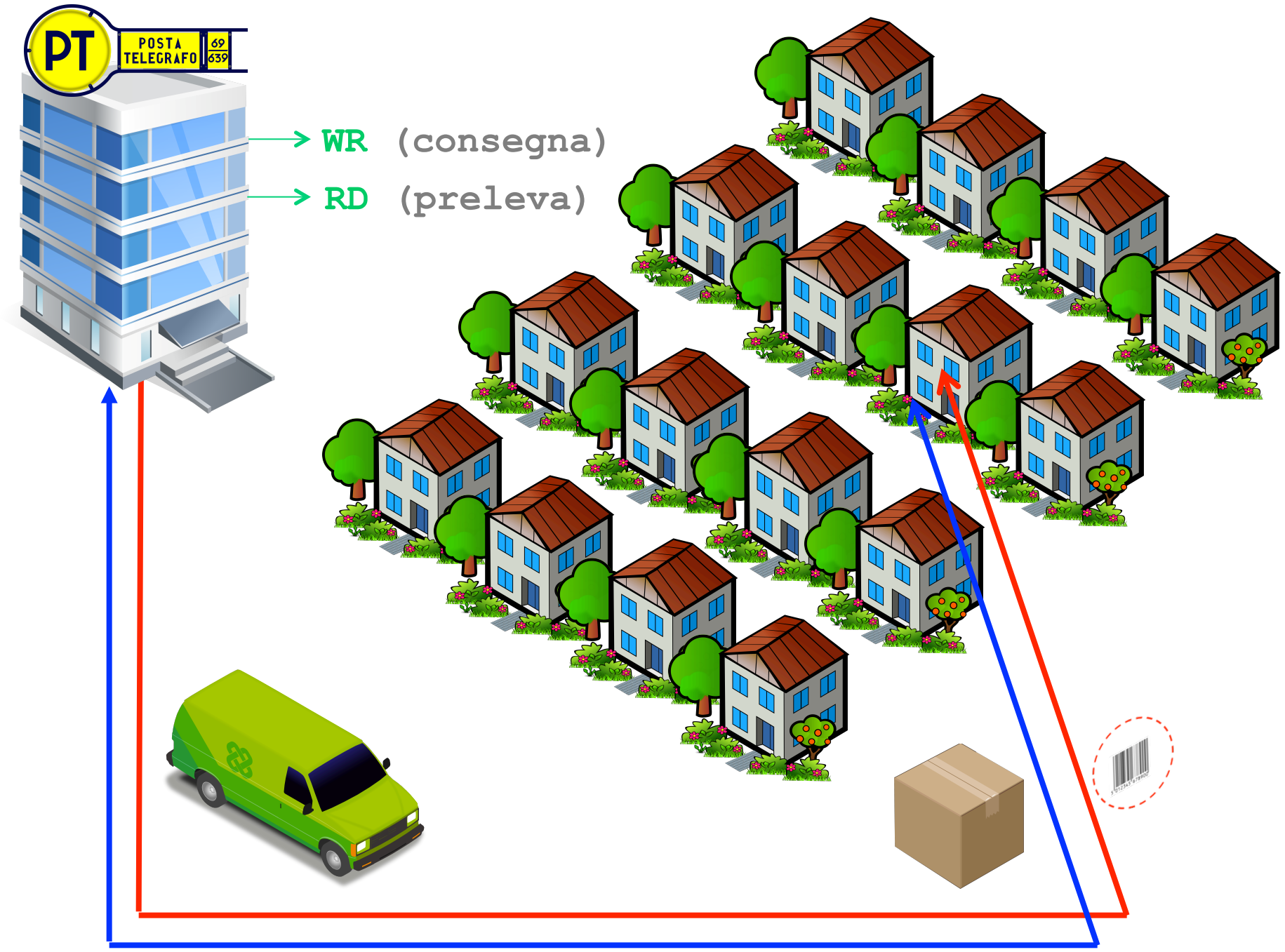
02 Mapping e decodifica

Stefano Mattoccia

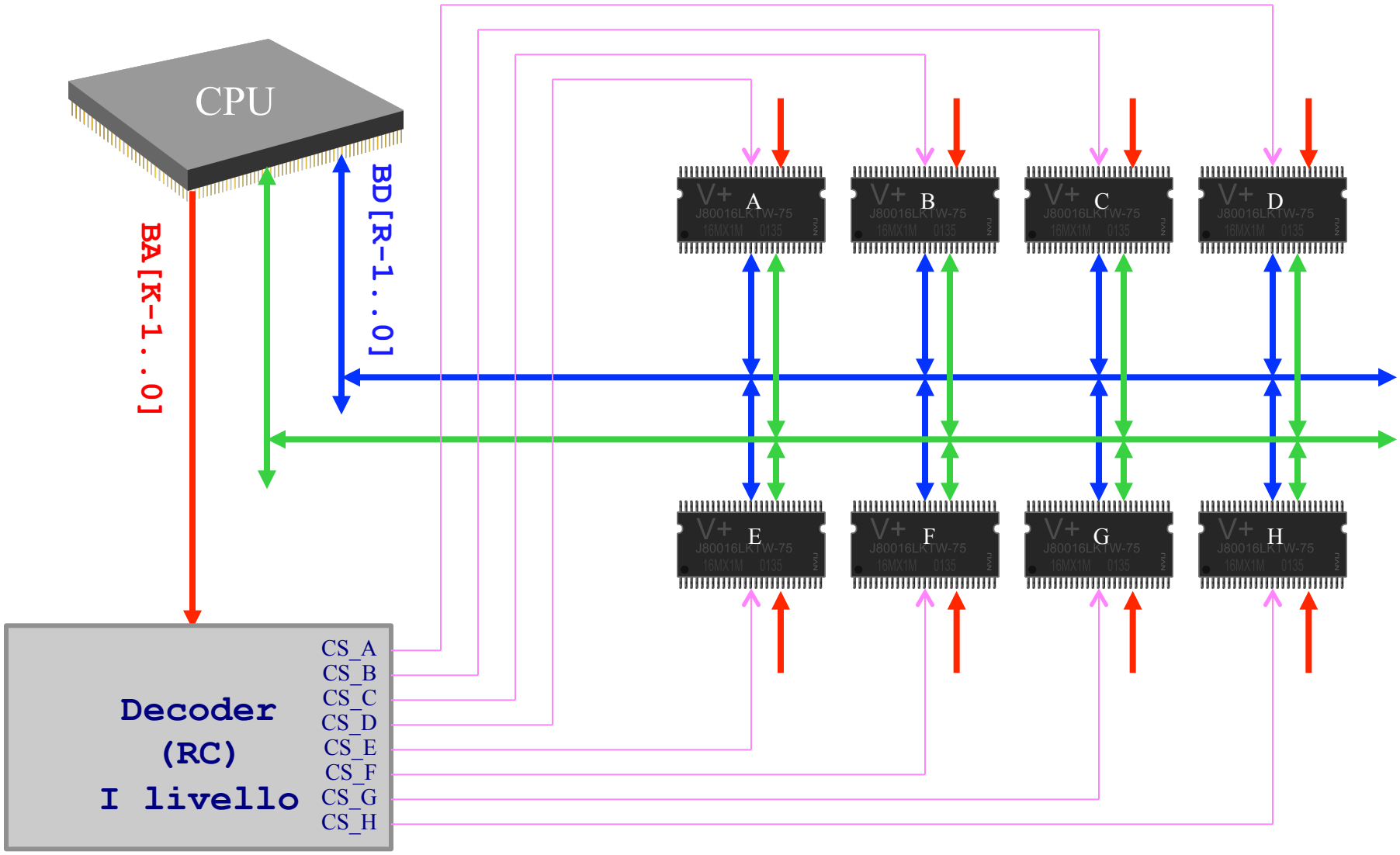
Spazio di indirizzamento

- Una CPU emette un certo numero di indirizzi e altri segnali sui bus di sistema per comunicare con altri moduli
- Il numero di diversi indirizzi emessi dalla CPU costituisce lo spazio di indirizzamento
- Una CPU che emette un indirizzo a 20 bit ha uno spazio di indirizzamento di 1 MB (2^20)
- Una CPU che emette un indirizzo a 32 bit ha uno spazio di indirizzamento di 4 GB (2^32)
- Le prime CPU avevano spazi di indirizzamento molto ridotto di alcuni KB (e.g., 64 KB o meno)
- Oggi è consuetudine avere almeno 32 bit di indirizzo

Un indirizzo per distribuire merci



Un indirizzo per distribuire dati (CPU)



Il decoder di II livello è all'interno di ciascun dispositivo (memoria, etc)

Condizione di visibilità di un dispositivo da parte del software

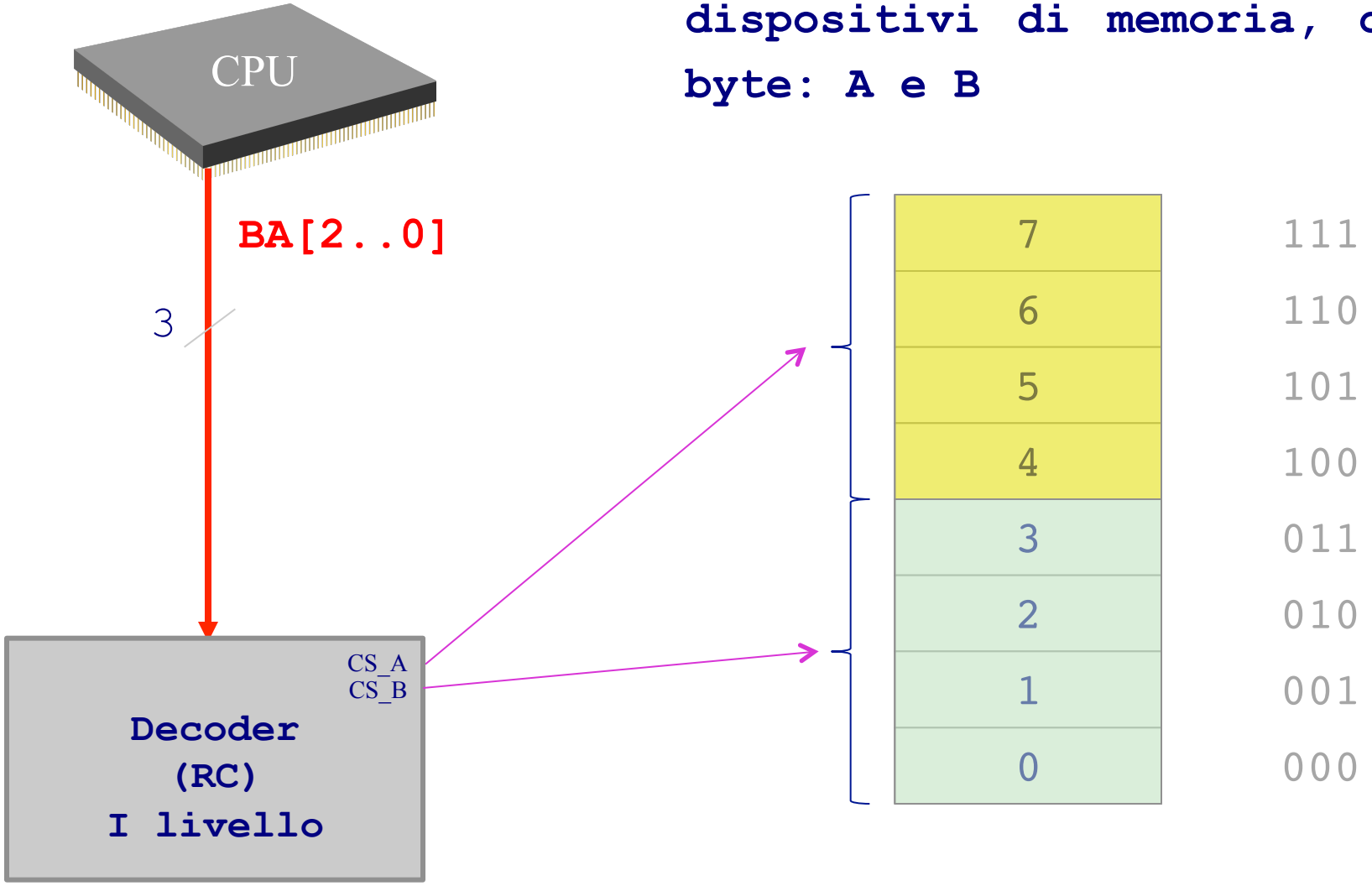
- **Condizione necessaria affinché un dispositivo fisico (memoria, interfaccia, o altra entità) sia accessibile al software è:**
  - ***il dispositivo deve essere mappato in uno spazio di indirizzamento***

- **Mappare in uno spazio di indirizzamento significa:**
  - ***associare al dispositivo una finestra di indirizzi di quello spazio di indirizzamento***

- **Si accede ai dispositivi mappati in uno spazio di indirizzamento con cicli di bus**

Esempio: una CPU con K=3 bit di indirizzo

- Lo spazio di indirizzamento sarebbe di solo 8 elementi
- Supponiamo di avere due dispositivi di memoria, da 4 byte: A e B



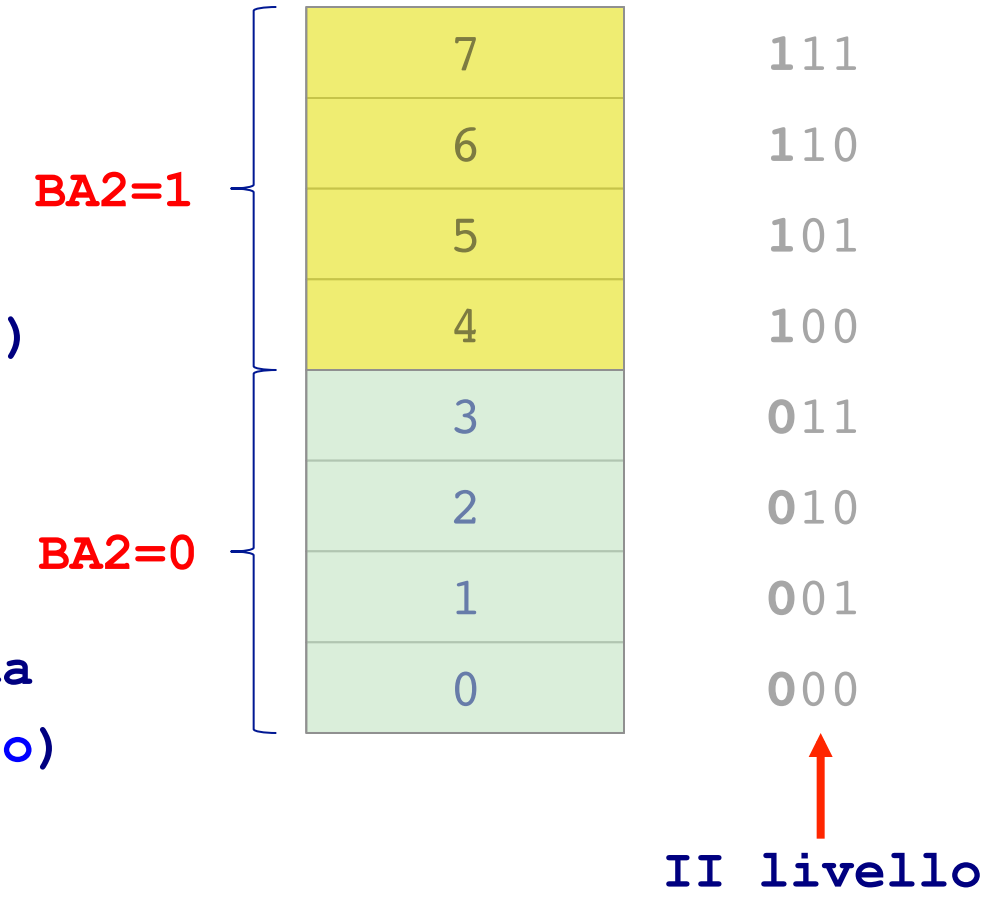
- Come facciamo ad attivare una delle due memorie in base all'indirizzo BA[2..0] emesso dalla CPU?

- Ovvero, come è fatta la rete di decodifica (I livello) che genera i due segnali CS_A e CS_B?

- CS_A = BA2
  CS_B = BA2*

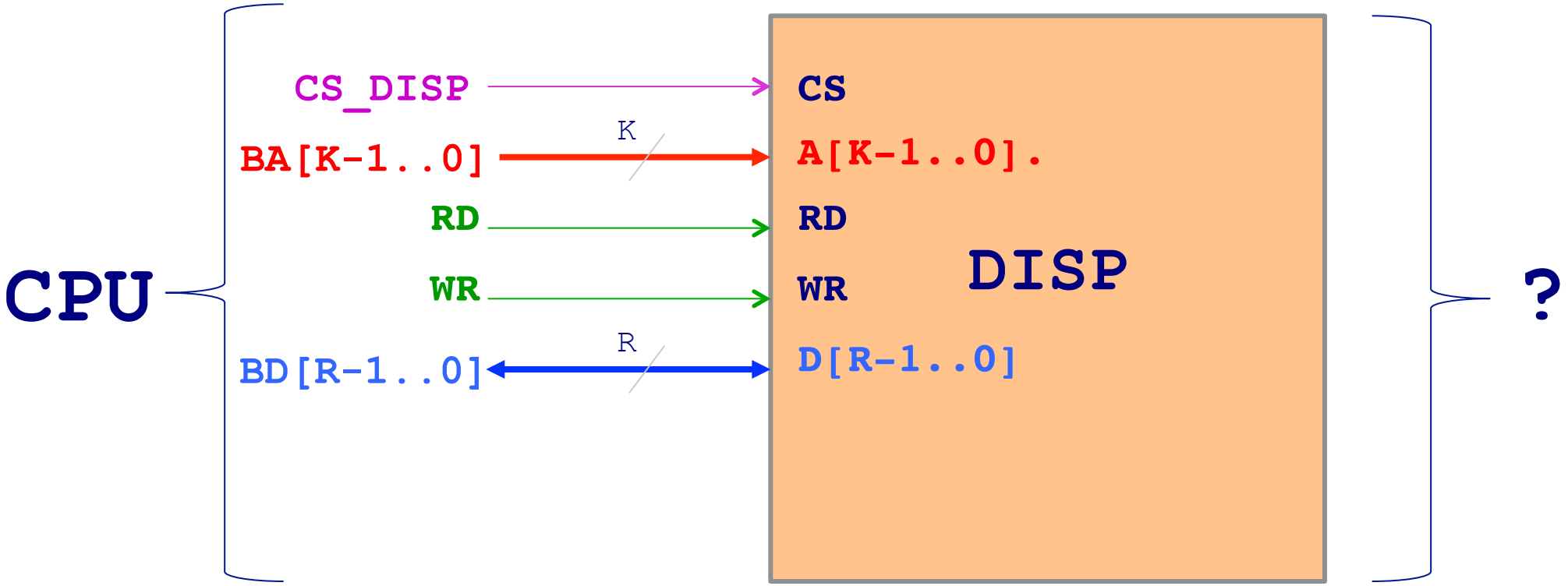
- Questi segnali saranno inviati a alle memorie (decodifica di I livello)

- Poi, sarà individuato l'elemento all'interno delle memoria selezionata (decodifica di II livello)



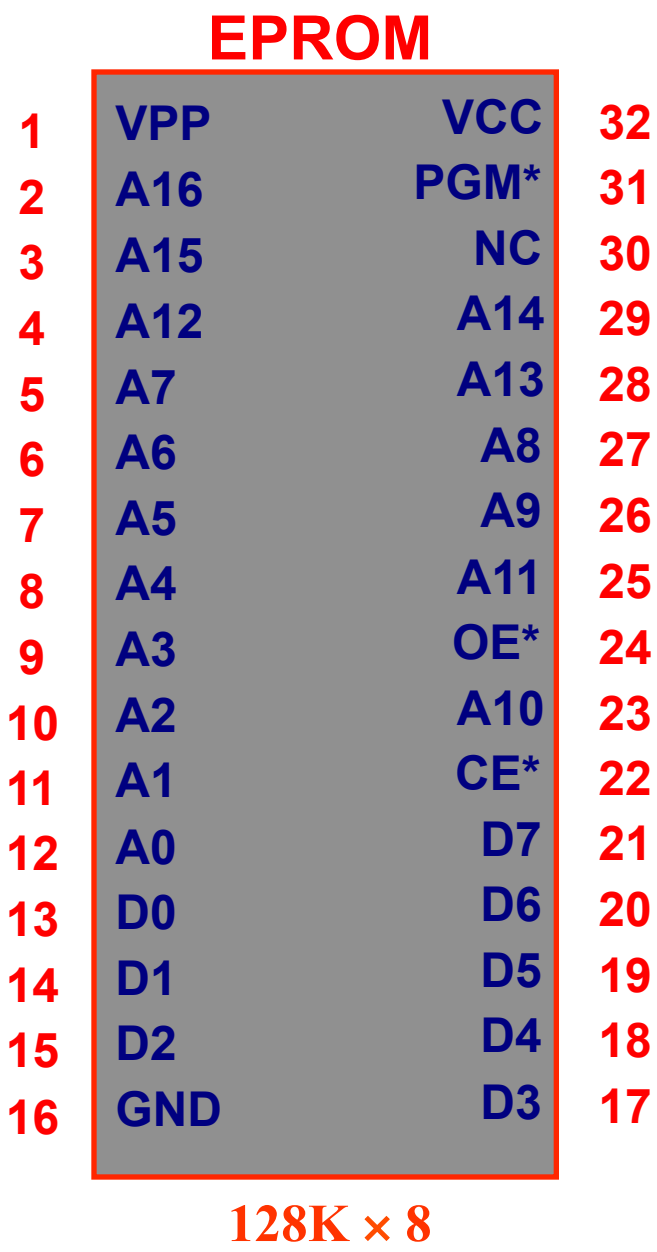
Come è fatto un generico dispositivo?

- Un qualsiasi dispositivo (memoria, periferica, etc), comunica con la CPU mediante una interfaccia standard a sx

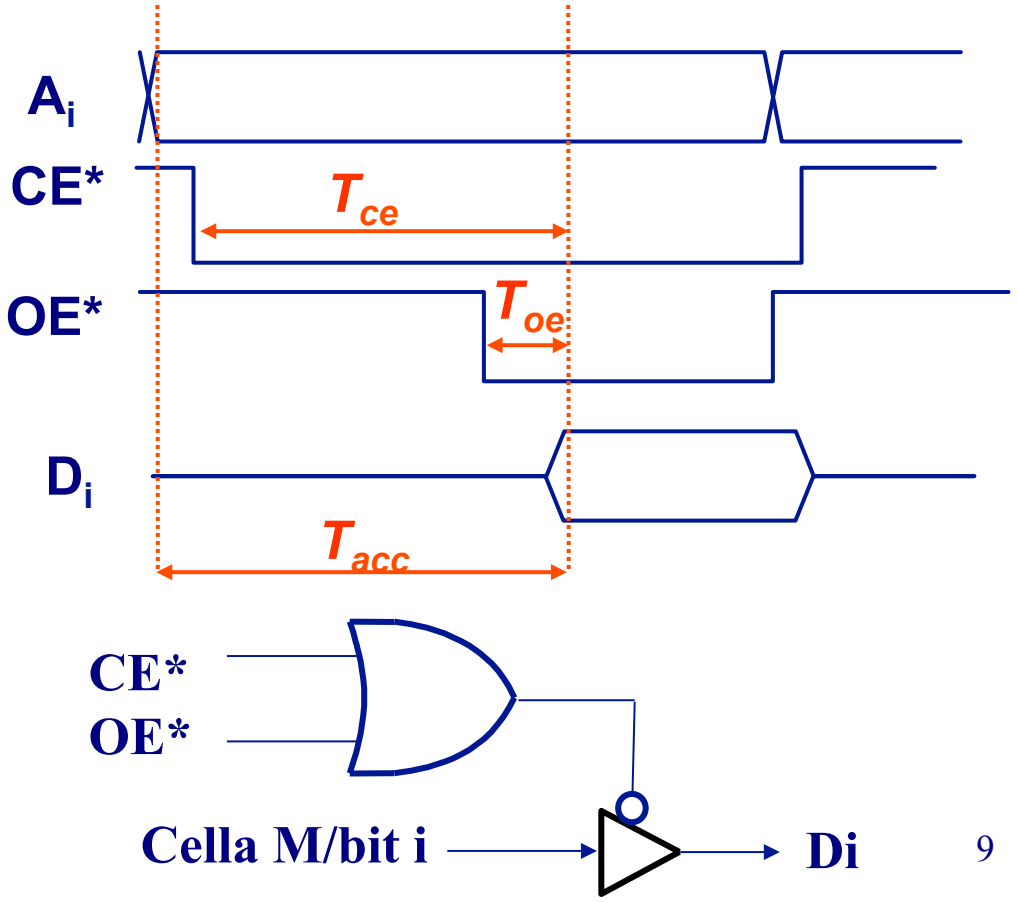


- La comunicazione con l'esterno avviene secondo modalità che sono specifiche del dispositivo e quindi non standard
- BA[K-1..0] utilizzati (internamente) per decodifica di II livello

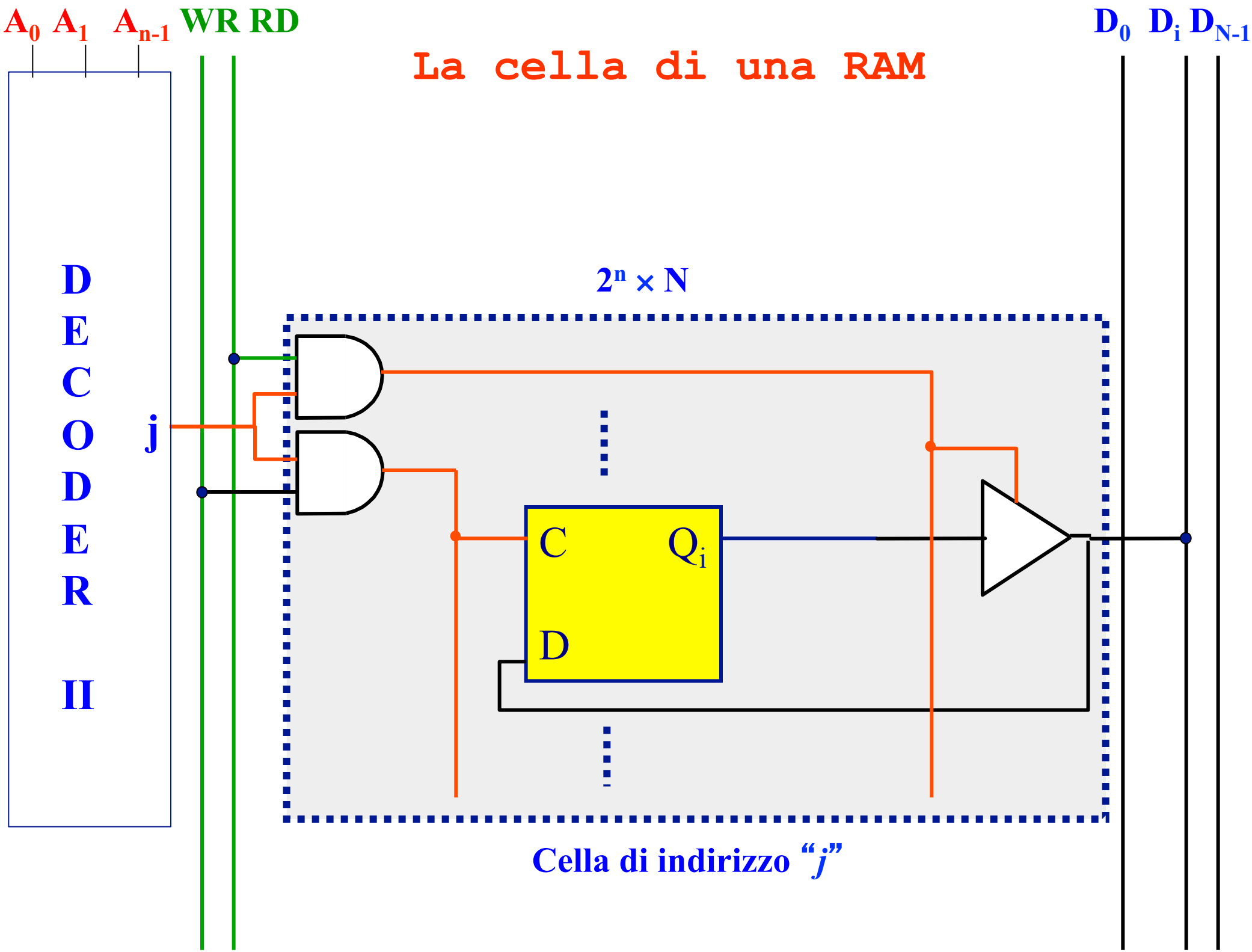
Memorie EPROM



- Memorie non volatili a sola lettura
- Capacità a multipli di 2:
  32K, 64K, 128K, 256K, etc



La cella di una RAM



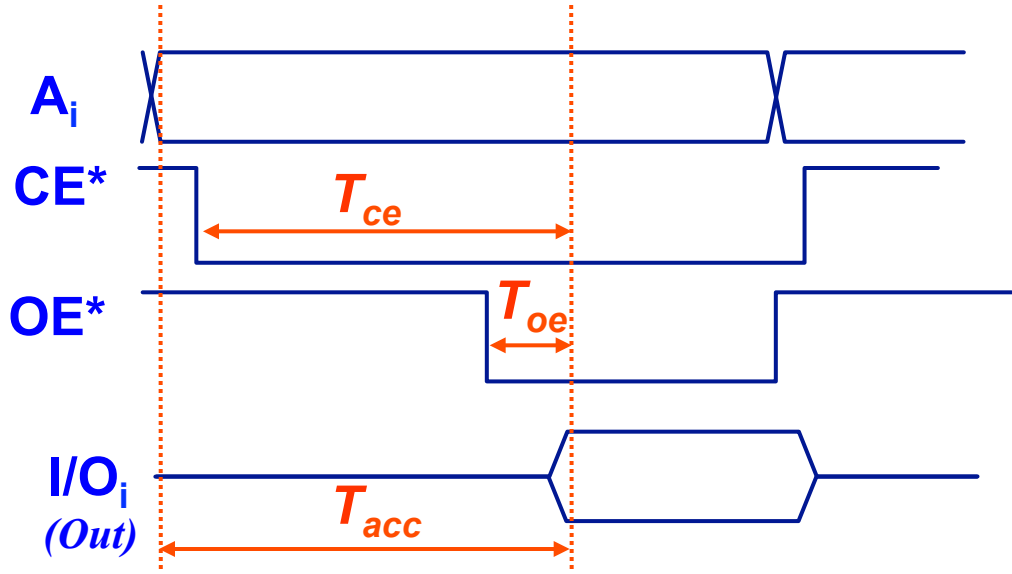
Memorie RAM (*SRAM*)

RAM

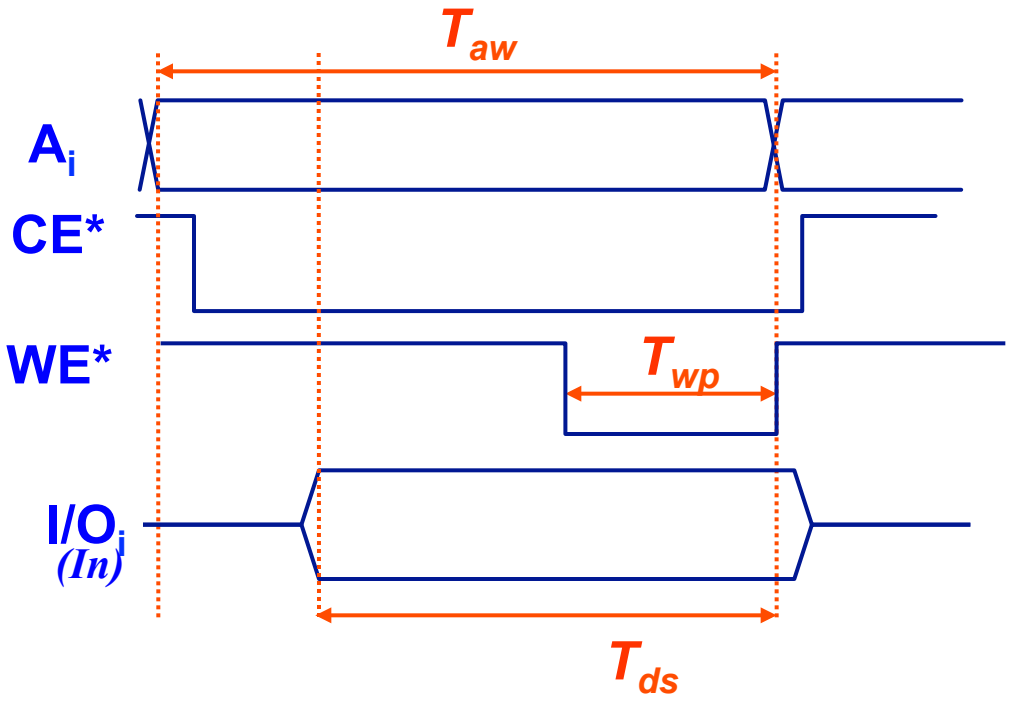
1	NC	VCC	32
2	A16	A15	31
3	A14	NC	30
4	A12	WE*	29
5	A7	A13	28
6	A6	A8	27
7	A5	A9	26
8	A4	A11	25
9	A3	OE*	24
10	A2	A10	23
11	A1	CE*	22
12	A0	I/O7	21
13	I/O0	I/O6	20
14	I/O1	I/O5	19
15	I/O2	I/O4	18
16	GND	I/O3	17

128K × 8

Read Cycle



Write Cycle



- Memorie volatili, leggibili e scrivibili
- Capacità a multipli di 4: 8K, 32K, 128K, 512K, etc
- *DRAM: 1 transistore per bit, maggiore capacità, più lente*

CYPRESS

CY7C199

32K x 8 Static RAM

Functional Description

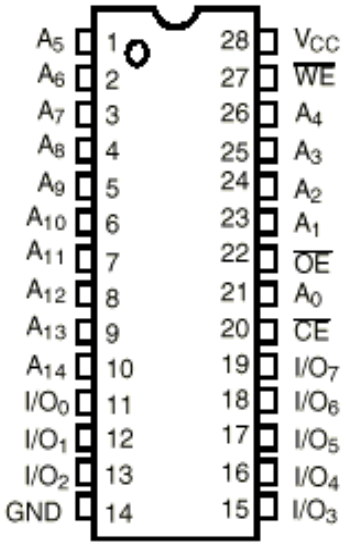
The CY7C199 is a high-performance CMOS static RAM organized as 32,768 words by 8 bits. Easy memory expansion is provided by an active LOW chip enable ($\overline{CE}$) and active LOW output enable ($\overline{OE}$) and three-state drivers. This device has an automatic power-down feature, reducing the power consumption by 81% when deselected. The CY7C199 is in the standard 300-mil-wide DIP, SOJ, and LCC packages.

An active LOW write enable signal ($\overline{WE}$) controls the writing/reading operation of the memory. When $\overline{CE}$ and $\overline{WE}$ inputs are both LOW, data on the eight data input/output pins (I/O_0 through I/O_7) is written into the memory location addressed by the address present on the address pins (A_0 through A_{14}). Reading the device is accomplished by selecting the device and enabling the outputs, $\overline{CE}$ and $\overline{OE}$ active LOW, while $\overline{WE}$ remains inactive or HIGH. Under these conditions, the contents of the location addressed by the information on address pins are present on the eight data input/output pins.

The input/output pins remain in a high-impedance state unless the chip is selected, outputs are enabled, and write enable ($\overline{WE}$) is HIGH. A die coat is used to improve alpha immunity.

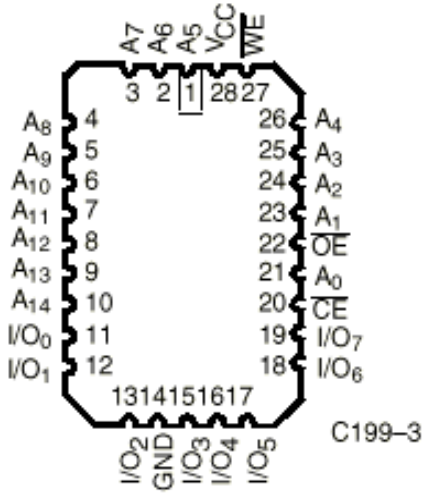
Pin Configurations

DIP / SOJ / SOIC
Top View

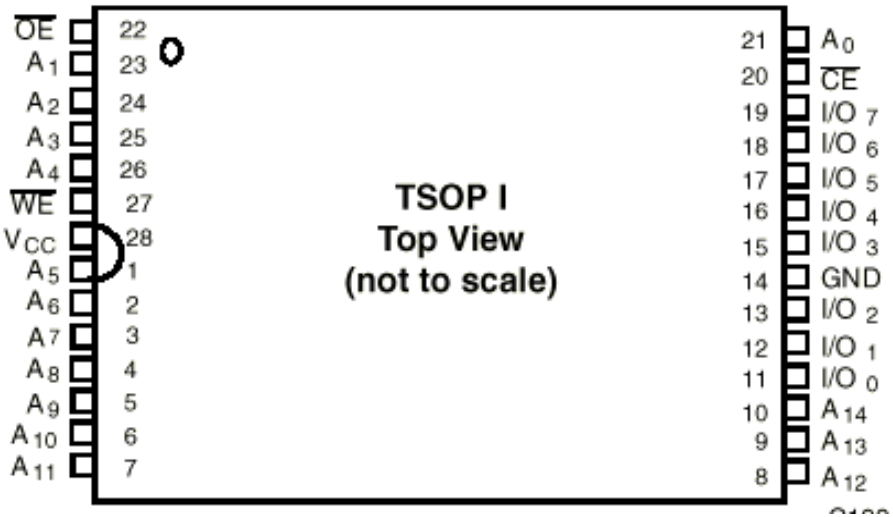


C199–2

LCC
Top View

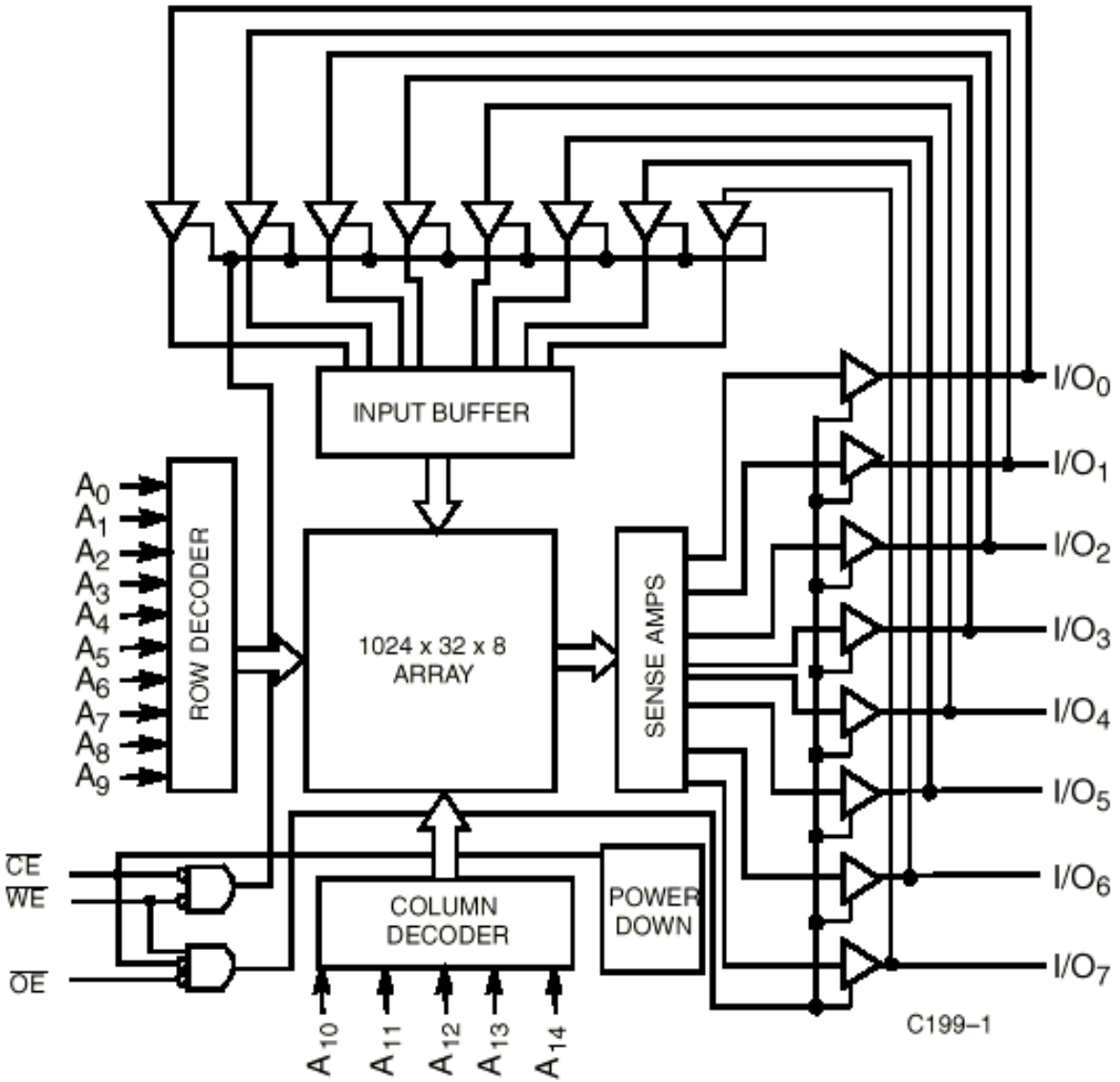


C199–3

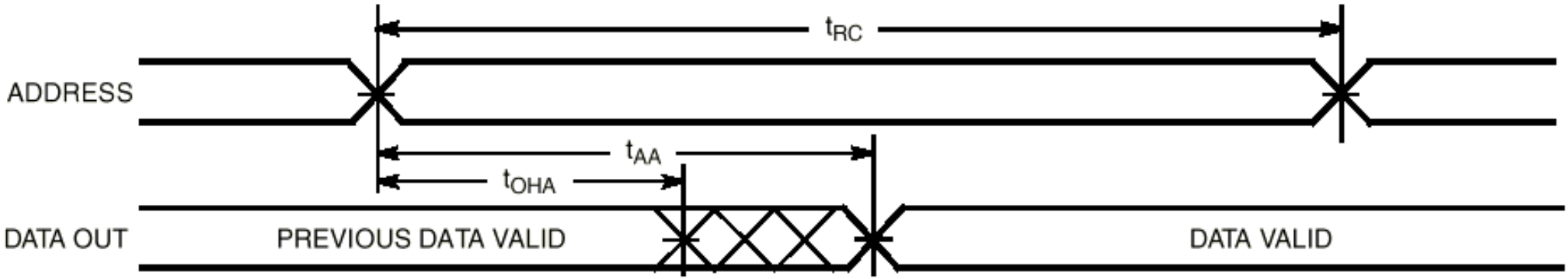


TSOP I
Top View
(not to scale)

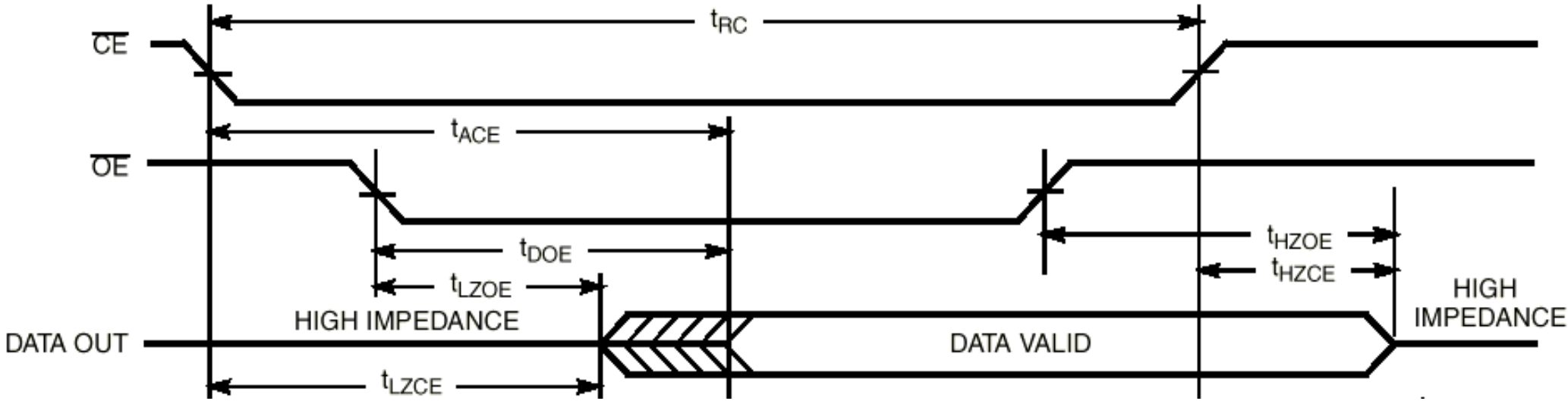
Logic Block Diagram



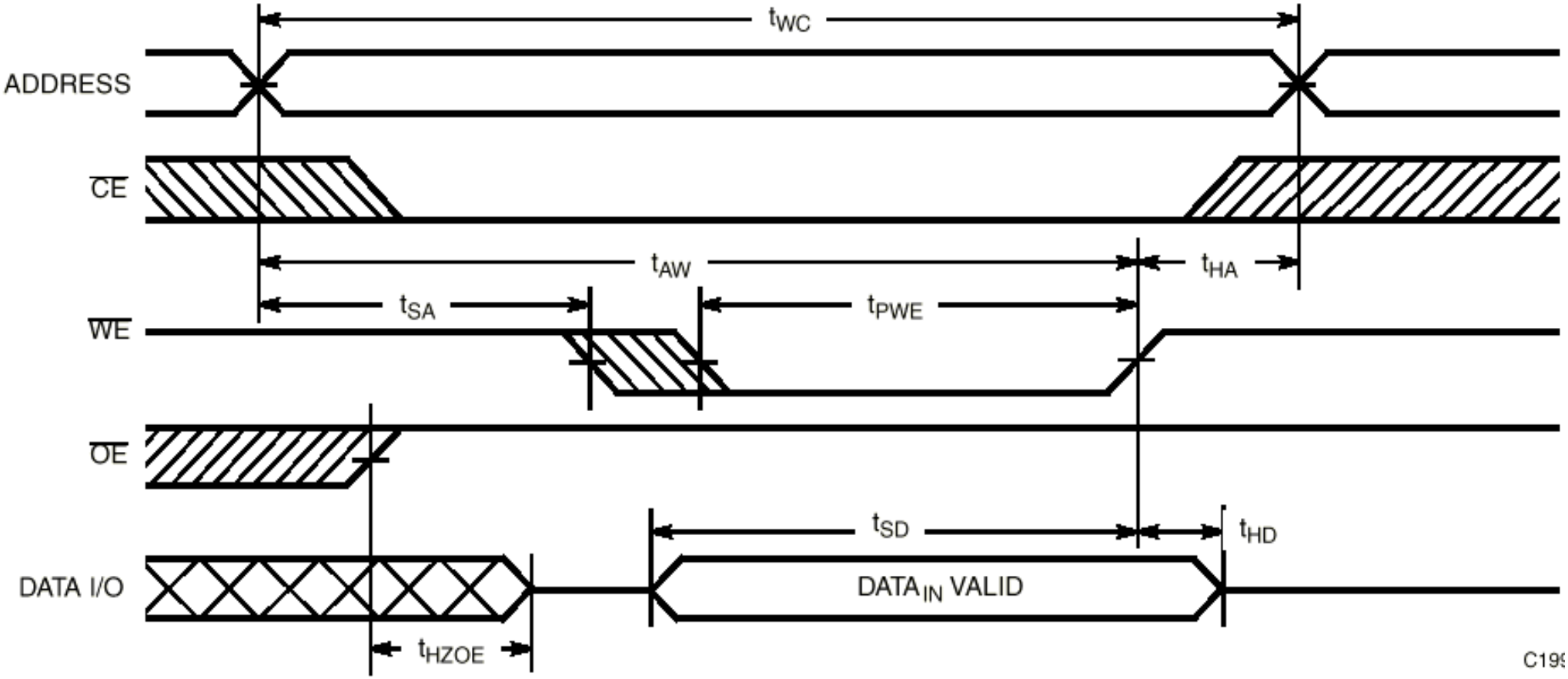
Read Cycle No. 1[12, 13]



Read Cycle No. 2 [13, 14]



Write Cycle No. 1 (WE Controlled)[10, 15, 16]



Switching Characteristics Over the Operating Range[3,7] (continued)

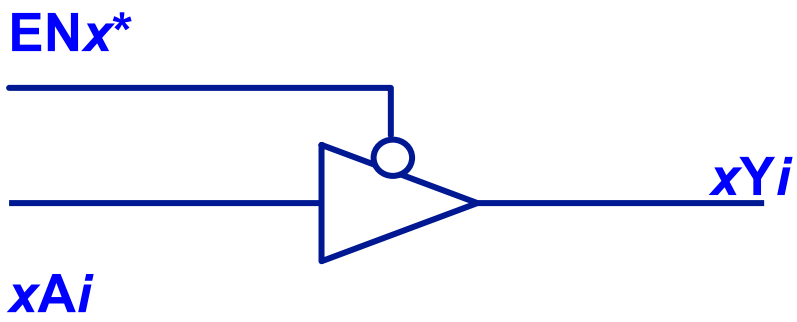
Parameter	Description	7C199-20		7C199-25		7C199-35		7C199-45		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
READ CYCLE										
t_{RC}	Read Cycle Time	20		25		35		45		ns
t_{AA}	Address to Data Valid		20		25		35		45	ns
t_{OHA}	Data Hold from Address Change	3		3		3		3		ns
t_{ACE}	$\overline{CE}$ LOW to Data Valid		20		25		35		45	ns
t_{DOE}	$\overline{OE}$ LOW to Data Valid		9		10		16		16	ns
t_{LZOE}	$\overline{OE}$ LOW to Low Z[8]	0		0		0		0		ns
t_{HZOE}	$\overline{OE}$ HIGH to High Z[8,9]		9		11		15		15	ns
t_{LZCE}	$\overline{CE}$ LOW to Low Z[8]	3		3		3		3		ns
t_{HZCE}	$\overline{CE}$ HIGH to High Z[8,9]		9		11		15		15	ns
t_{PU}	$\overline{CE}$ LOW to Power-Up	0		0		0		0		ns
t_{PD}	$\overline{CE}$ HIGH to Power-Down		20		20		20		25	ns
WRITE CYCLE[10,11]										
t_{WC}	Write Cycle Time	20		25		35		45		ns
t_{SCE}	$\overline{CE}$ LOW to Write End	15		18		22		22		ns
t_{AW}	Address Set-Up to Write End	15		20		30		40		ns
t_{HA}	Address Hold from Write End	0		0		0		0		ns
t_{SA}	Address Set-Up to Write Start	0		0		0		0		ns
t_{PWE}	$\overline{WE}$ Pulse Width	15		18		22		22		ns
t_{SD}	Data Set-Up to Write End	10		10		15		15		ns
t_{HD}	Data Hold from Write End	0		0		0		0		ns
t_{HZWE}	$\overline{WE}$ LOW to High Z[9]		10		11		15		15	ns
t_{LZWE}	$\overline{WE}$ HIGH to Low Z[8]	3		3		3		3		ns

Integrati Notevoli: “244”

74XX244

1A1	1Y1
1A2	1Y2
1A3	1Y3
1A4	1Y4
2A1	2Y1
2A2	2Y2
2A3	2Y3
2A4	2Y4
EN1*	EN2*

Driver 3-state ad 8-bit
(strutturato in 2 gruppi di 4 bit)



SN54AC244, SN74AC244
OCTAL BUFFERS/DRIVERS WITH 3-STATE OUTPUTS

SCAS514C – JUNE 1995 – REVISED SEPTEMBER 1996

description

These octal buffers and line drivers are designed specifically to improve the performance and density of 3-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters.

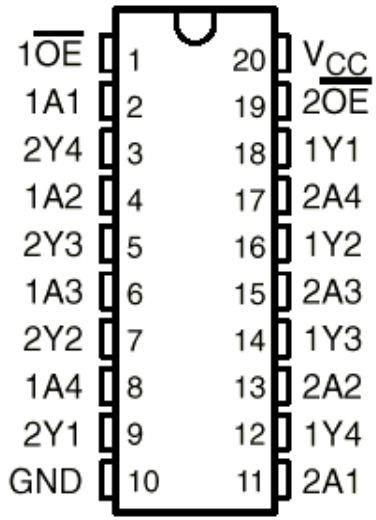
The 'AC244 are organized as two 4-bit buffers/drivers with separate output-enable ($\overline{OE}$) inputs. When $\overline{OE}$ is low, the device passes noninverted data from the A inputs to the Y outputs. When $\overline{OE}$ is high, the outputs are in the high-impedance state.

The SN54AC244 is characterized for operation over the full military temperature range of –55°C to 125°C. The SN74AC244 is characterized for operation from –40°C to 85°C.

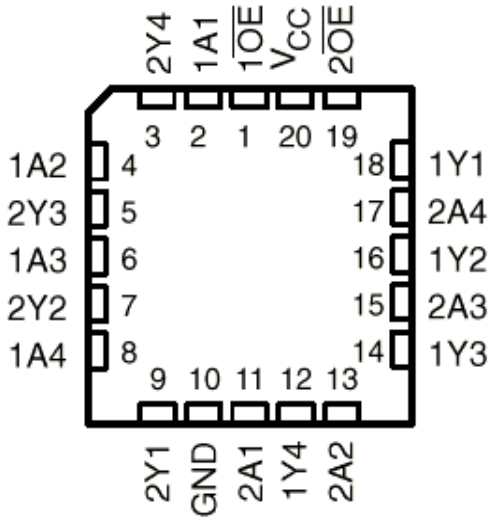
FUNCTION TABLE
(each buffer)

INPUTS		OUTPUT
$\overline{OE}$	A	Y
L	H	H
L	L	L
H	X	Z

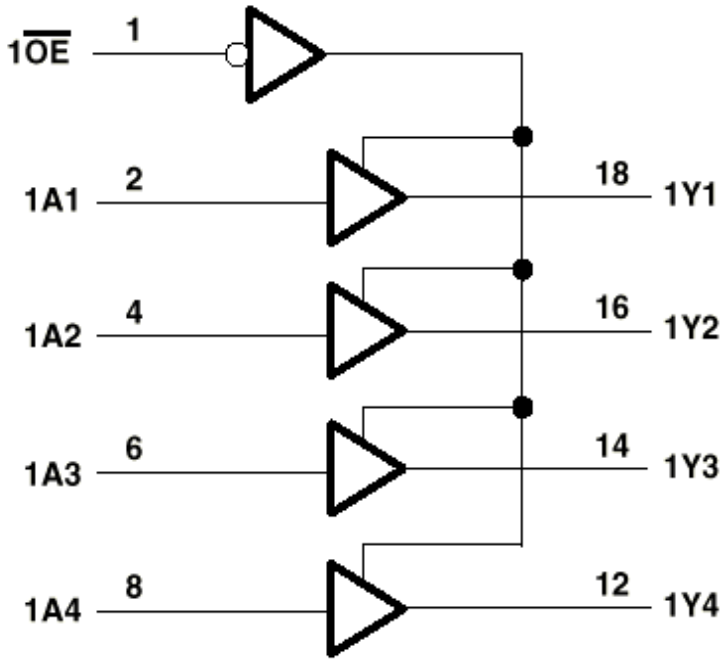
SN54AC244 . . . J OR W PACKAGE
SN74AC244 . . . DB, DW, N, OR PW PACKAGE
(TOP VIEW)

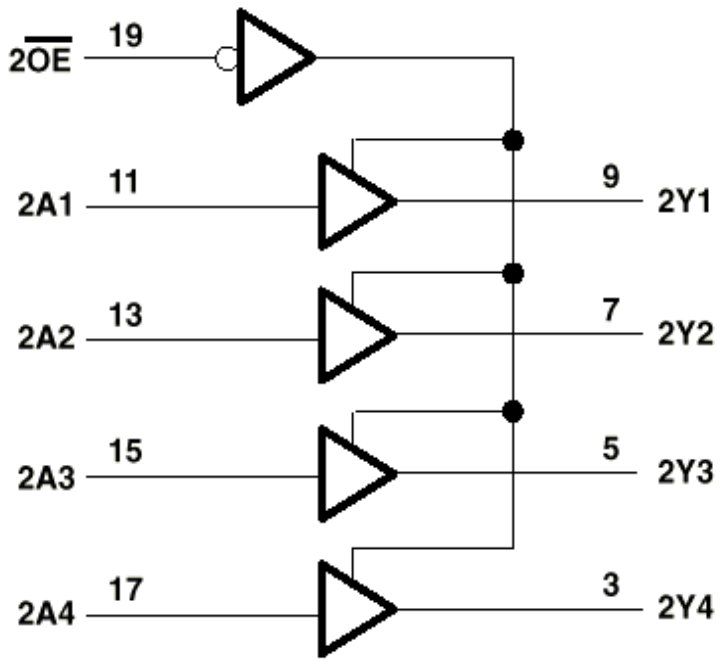


SN54AC244 . . . FK PACKAGE
(TOP VIEW)



logic diagram (positive Logic)





switching characteristics over recommended operating free-air temperature range, V_{CC} = 5 V ± 0.5 V (unless otherwise noted) (see Figure 1)

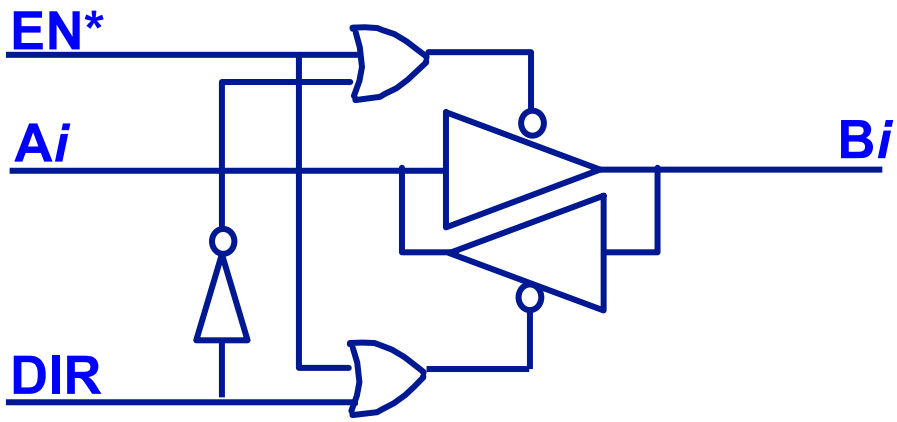
PARAMETER	FROM (INPUT)	TO (OUTPUT)	T_A = 25°C			SN54AC244		SN74AC244		UNIT
			MIN	TYP	MAX	MIN	MAX	MIN	MAX	
t_{PLH}	A	Y	1.5	5	7	1	9.5	1	7.5	ns
t_{PHL}			1.5	5	7	1	9	1	7.5	
t_{PZH}	$\overline{OE}$	Y	1.5	5	7	1	9	1.5	8	ns
t_{PZL}			1.5	5.5	8	1	10.5	1.5	8.5	
t_{PHZ}	$\overline{OE}$	Y	2.5	6.5	9	1	10.5	1	9.5	ns
t_{PLZ}			2	6.5	9	1	11	2	9.5	

Integrati Notevoli: “245”

74XX245

A1	B1
A2	B2
A3	B3
A4	B4
A5	B5
A6	B6
A7	B7
A8	B8
EN*	DIR

Driver bidirezionale (*transceiver*) ad 8-bit.



description

This octal bus transceiver is designed for asynchronous two-way communication between data buses. The control-function implementation minimizes external timing requirements.

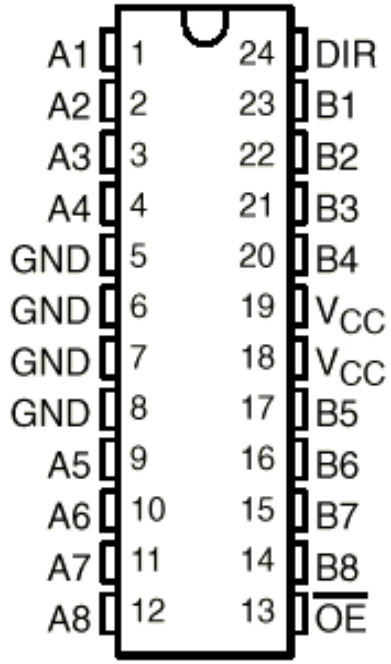
The device allows noninverted data transmission from the A bus to the B bus or from the B bus to the A bus, depending on the logic level at the direction-control (DIR) input. The output-enable ($\overline{OE}$) input can be used to disable the device so that the buses are effectively isolated.

The 74AC11245 is characterized for operation from –40°C to 85°C.

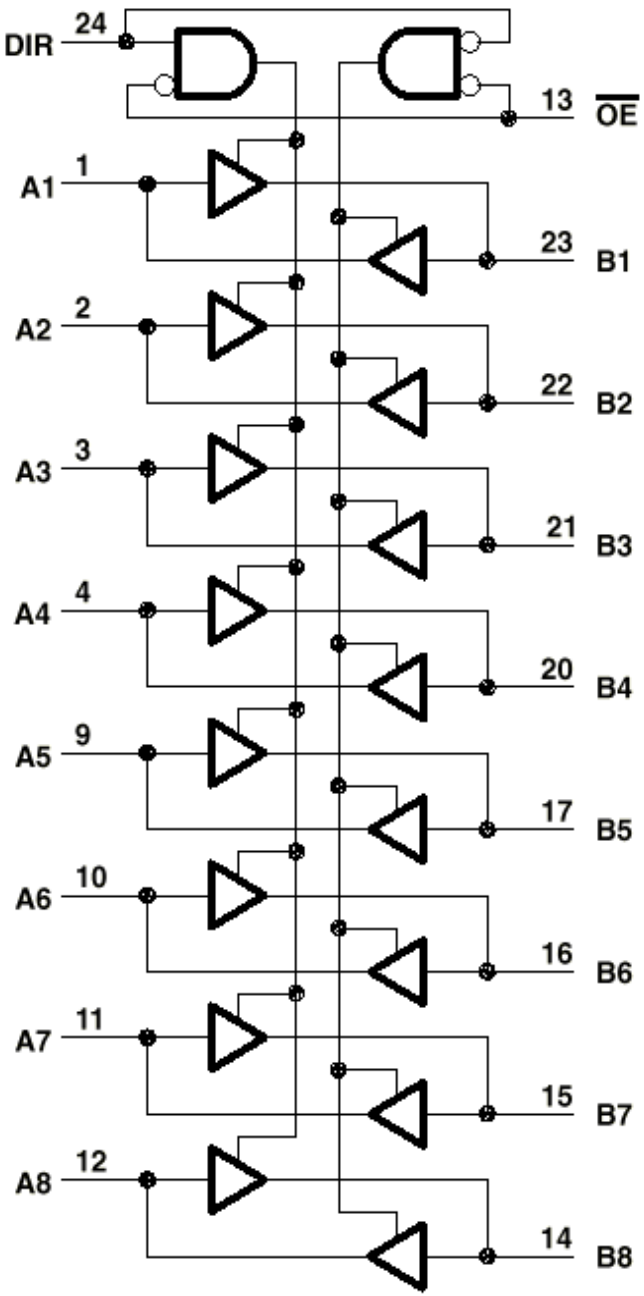
FUNCTION TABLE

OUTPUT ENABLE $\overline{OE}$	DIRECTION CONTROL DIR	OPERATION
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

DB, DW, NT, OR PW PACKAGE
(TOP VIEW)



logic diagram (positive logic)



switching characteristics over recommended operating free-air temperature range, V_{CC} = 5 V ± 0.5 V (unless otherwise noted) (see Figure 1)

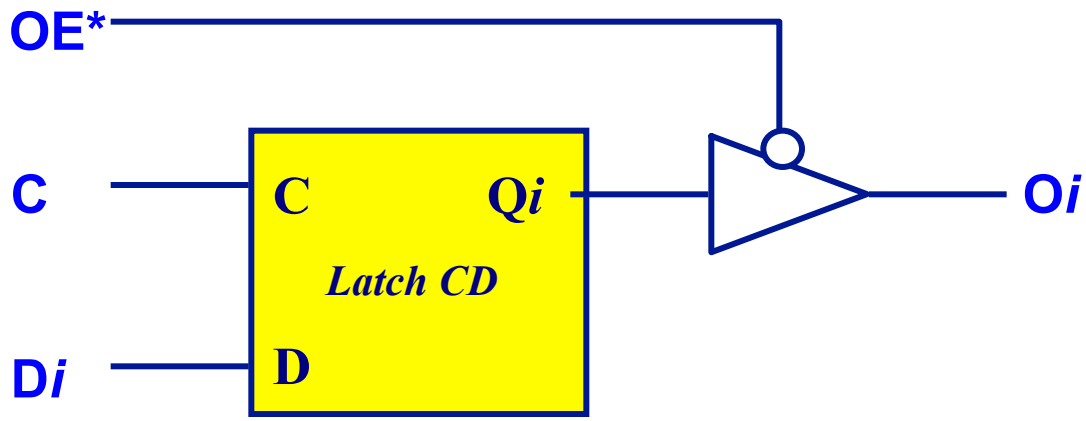
PARAMETER	FROM (INPUT)	TO (OUTPUT)	T_A = 25°C			MIN	MAX	UNIT
			MIN	TYP	MAX			
t_{PLH}	A or B	B or A	1.5	4.8	8.5	1.5	9.5	ns
t_{PHL}			1.5	4.1	6.3	1.5	6.9	
t_{PZH}	$\overline{OE}$	B or A	1.5	6.2	10.2	1.5	11.4	ns
t_{PZL}			1.5	5.9	8.6	1.5	9.5	
t_{PHZ}	$\overline{OE}$	B or A	1.5	6.4	8.8	1.5	9.5	ns
t_{PLZ}			1.5	7	9.6	1.5	10.4	

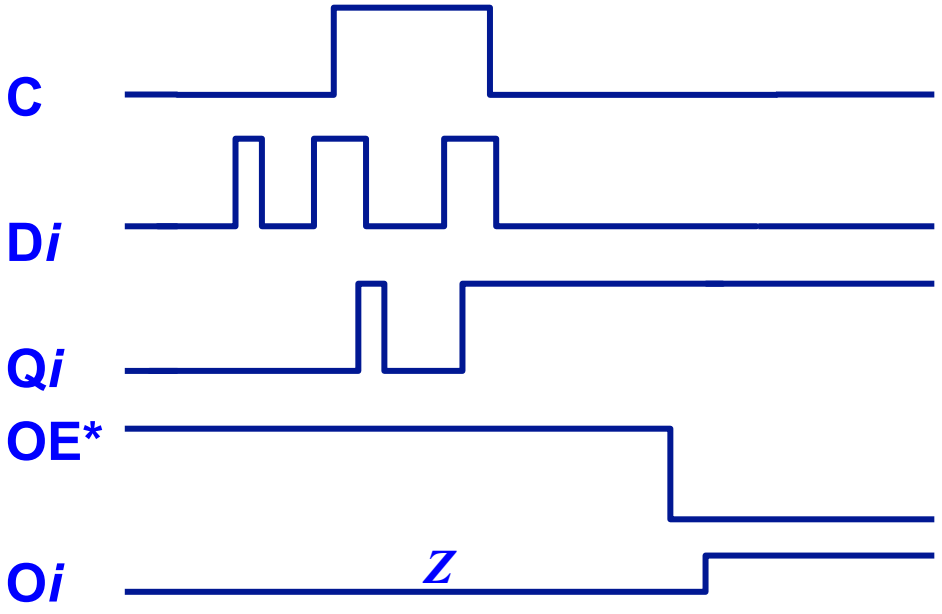
Integrati Notevoli: “373”

74XX373

D0	O0
D1	O1
D2	O2
D3	O3
D4	O4
D5	O5
D6	O6
D7	O7
C	OE*

Latch a 8-bit
con uscite 3-state



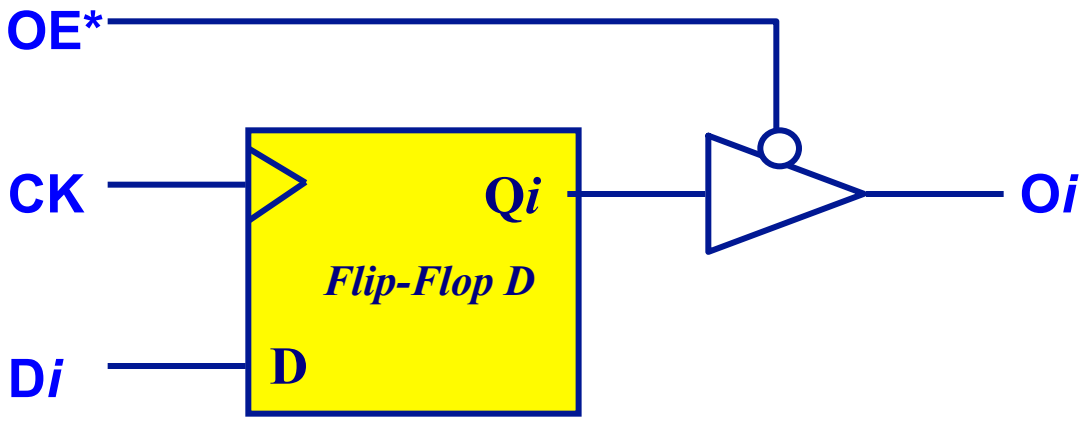


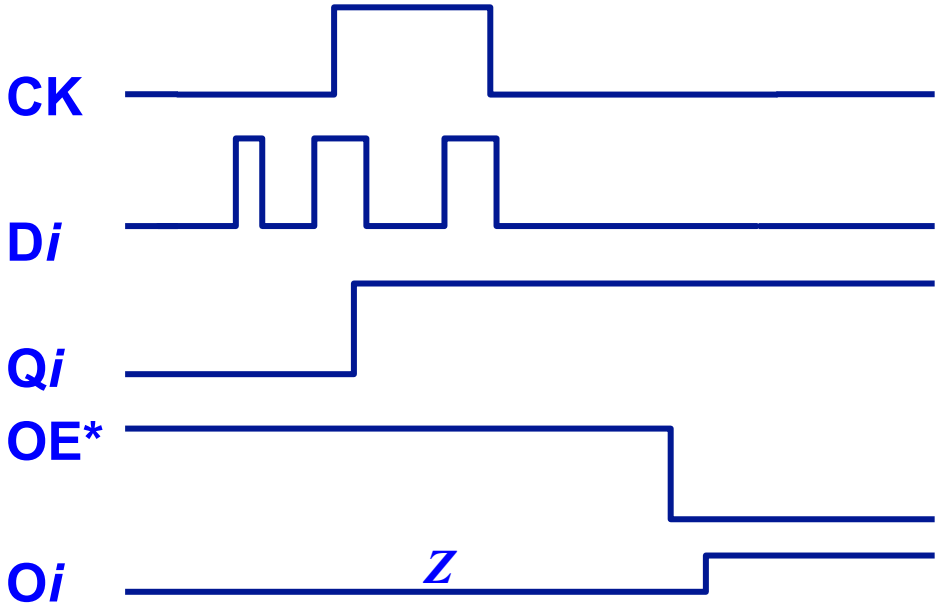
Integrati Notevoli: “374”

74XX374

D0	O0
D1	O1
D2	O2
D3	O3
D4	O4
D5	O5
D6	O6
D7	O7
CK	OE*

Registro *edge-triggered*
con uscite 3-state





SN54LS373, SN54LS374, SN54S373, SN54S374, SN74LS373, SN74LS374, SN74S373, SN74S374
OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

SDLS165 – OCTOBER 1975 – REVISED MARCH 1988

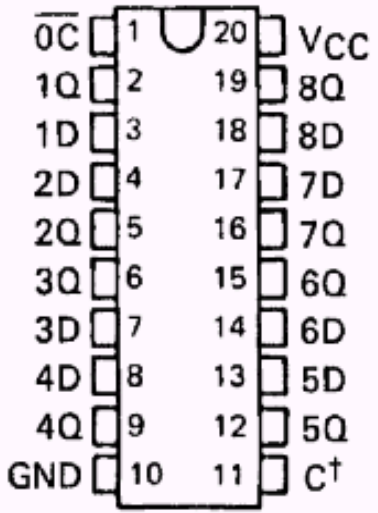
'LS373, 'S373
FUNCTION TABLE

OUTPUT ENABLE	ENABLE LATCH	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

'LS374, 'S374
FUNCTION TABLE

OUTPUT ENABLE	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

SN54LS373, SN54LS374, SN54S373, SN54S374 . . . J OR W PACKAGE
SN74LS373, SN74LS374, SN74S373, SN74S374 . . . DW OR N PACKAGE
(TOP VIEW)



description

The eight latches of the 'LS373 and 'S373 are transparent D-type latches meaning that while the enable (C) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was set up.

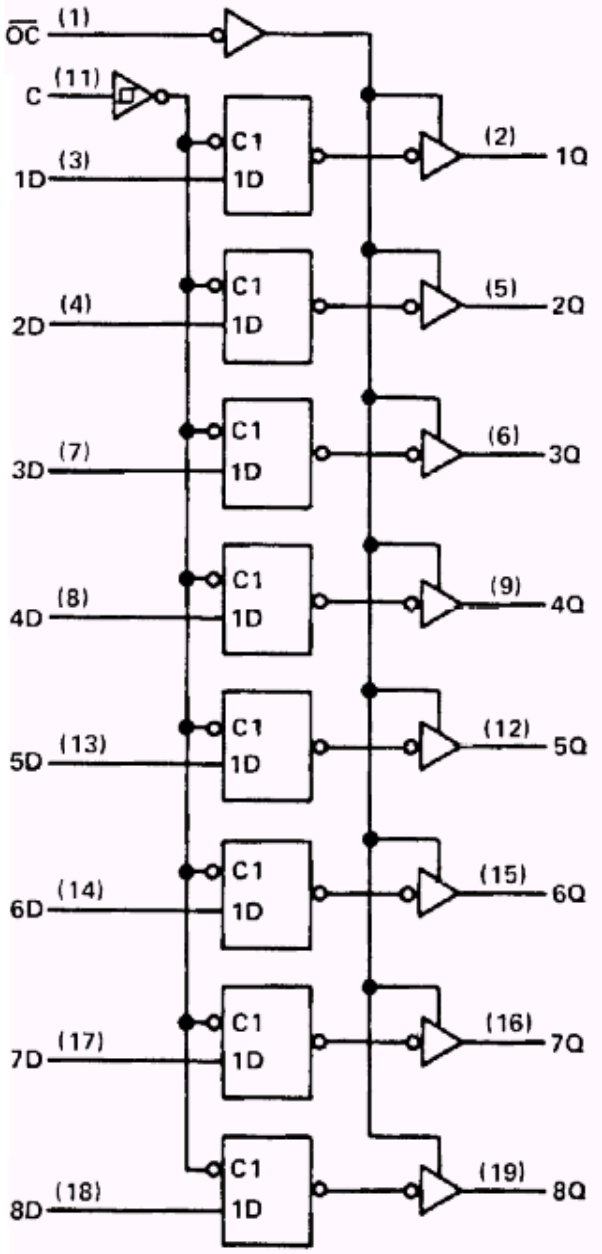
The eight flip-flops of the 'LS374 and 'S374 are edge-triggered D-type flip-flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were setup at the D inputs.

Schmitt-trigger buffered inputs at the enable/clock lines of the 'S373 and 'S374 devices, simplify system design as ac and dc noise rejection is improved by typically 400 mV due to the input hysteresis. A buffered output control input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or a high-impedance state. In the high-impedance state the outputs neither load nor drive the bus lines significantly.

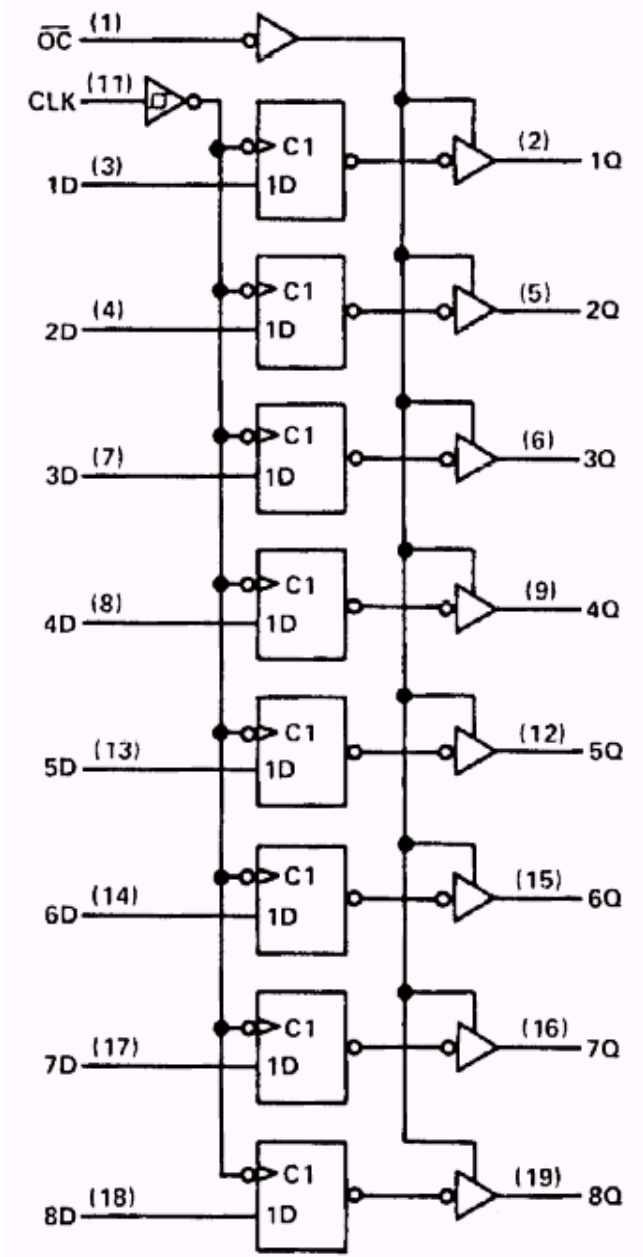
The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are off.

logic diagrams (positive logic)

'LS373, 'S373
TRANSPARENT LATCHES



'LS374, 'S374
POSITIVE-EDGE-TRIGGERED FLIP-FLOPS



recommended operating conditions

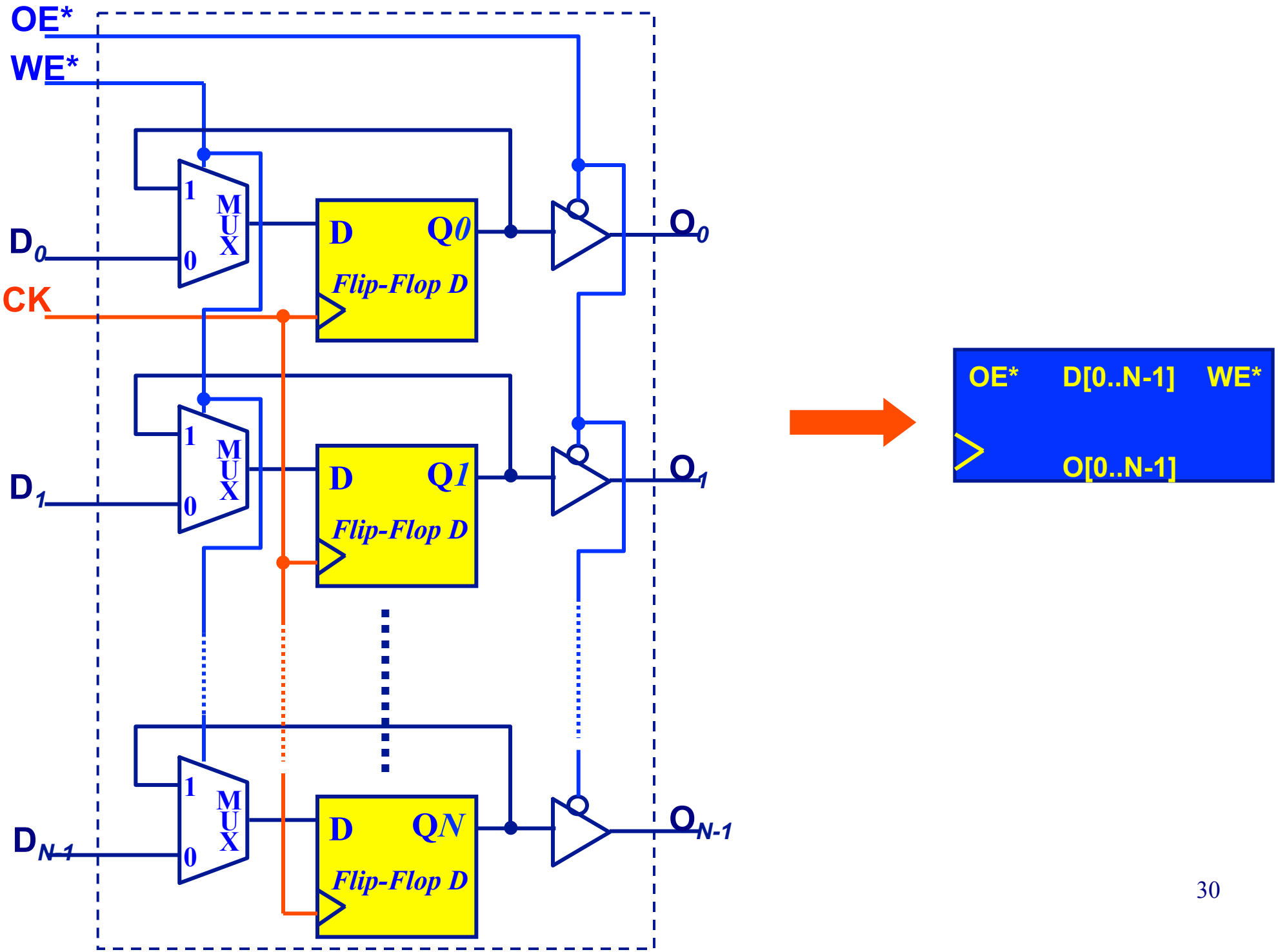
			SN54LS'			SN74LS'			UNIT
			MIN	NOM	MAX	MIN	NOM	MAX	
V_{CC}	Supply voltage		4.5	5	5.5	4.75	5	5.25	V
V_{OH}	High-level output voltage				5.5			5.5	V
I_{OH}	High-level output current				−1			−2.6	mA
I_{OL}	Low-level output current				12			24	mA
t_w	Pulse duration	CLK high	15			15			ns
		CLK low	15			15			
t_{su}	Data setup time	'LS373	5↓			5↓			ns
		'LS374	20↑			20↑			
t_h	Data hold time	'LS373	20↓			20↓			ns
		'LS374†	5↑			0↑			
T_A	Operating free-air temperature		−55		125	0		70	°C

†The t_h specification applies only for data frequency below 10 MHz. Designs above 10 MHz should use a minimum of 5 ns. (Commercial only)

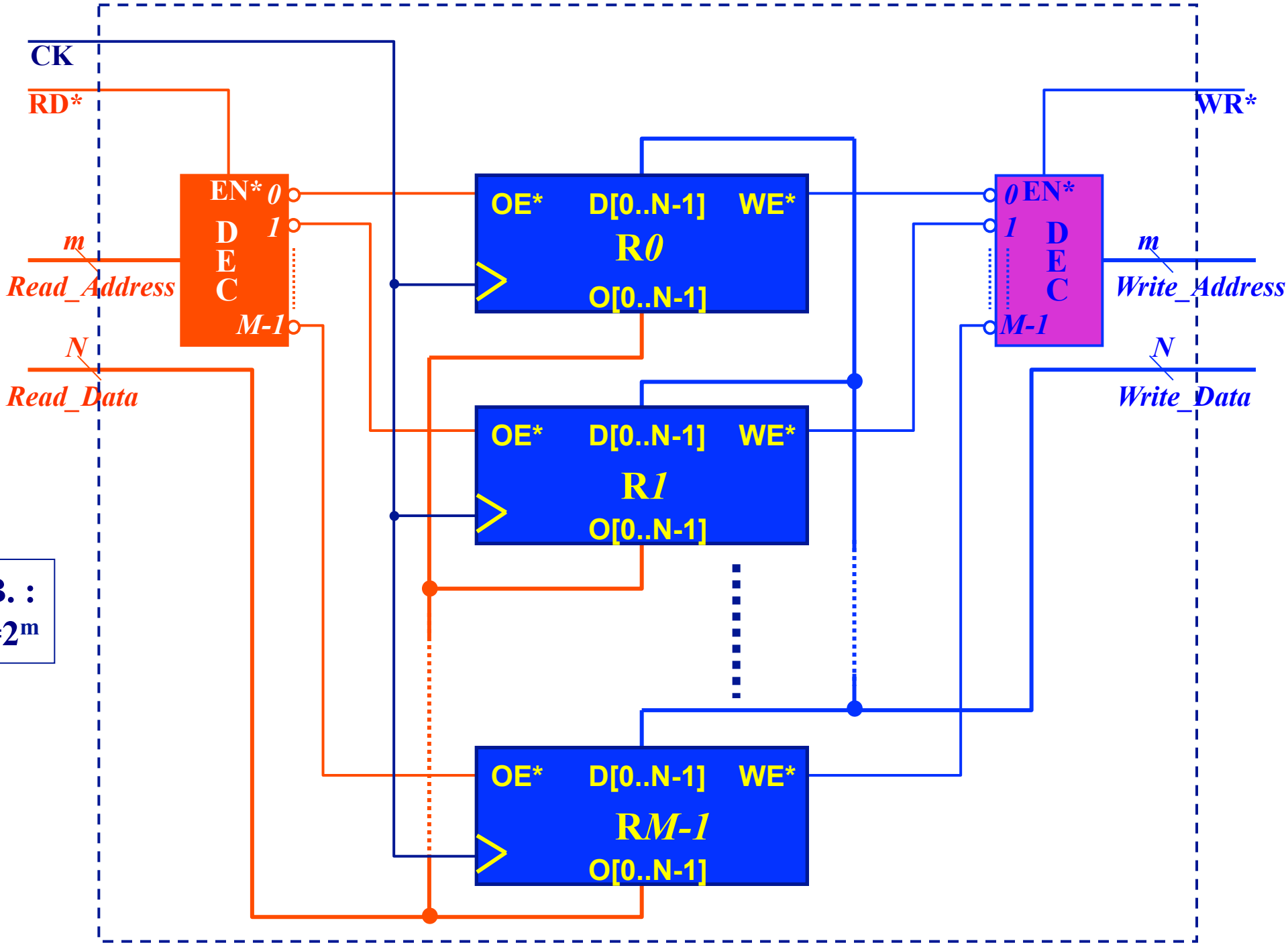
switching characteristics, V_{CC} = 5 V, T_A = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'LS373			'LS374			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
f_{max}			C_L = 45 pF, R_L = 667 Ω, See Notes 2 and 3				35	50		MHz
t_{PLH}	Data	Any Q			12	18				ns
t_{PHL}					12	18				
t_{PLH}	Clock or enable	Any Q			20	30		15	28	ns
t_{PHL}					18	30		19	28	
t_{PZH}	Output Control	Any Q			15	28		20	26	ns
t_{PZL}					25	36		21	28	
t_{PHZ}	Output Control	Any Q	C_L = 5 pF, R_L = 667 Ω, See Note 3		15	25		15	28	ns
t_{PLZ}	Output Control	Any Q			12	20		12	20	ns

Registro Edge-Triggered con WE*



Register File (1 read-port, 1 write-port)



Mapping di dispositivi da 8 bit in sistemi con bus dati da 8 bit

- **Consideriamo dispositivi con porta dati a 8 bit**
- **Imponiamo (temporaneamente) l'ulteriore condizione che il parallelismo del bus dati sia a 8 bit**
- **In queste ipotesi l'assegnamento a un dispositivo di una finestra di indirizzi in uno spazio di indirizzamento avverrà in generale nel rispetto delle due seguenti ulteriori condizioni restrittive:**

  - ***la dimensione della finestra di indirizzi associata a un dispositivo è una potenza di due***

  - ***la finestra è composta da indirizzi contigui***

Dimensione della finestra occupata da un dispositivo - esempi

- Un dispositivo accessibile attraverso il bus occupa in generale n = 2^K posizioni nello spazio di indirizzamento

- n rappresenta il numero di *oggetti* a 8 bit indirizzabili all'interno del dispositivo (es. numero di celle di memoria nelle RAM ed EPROM)

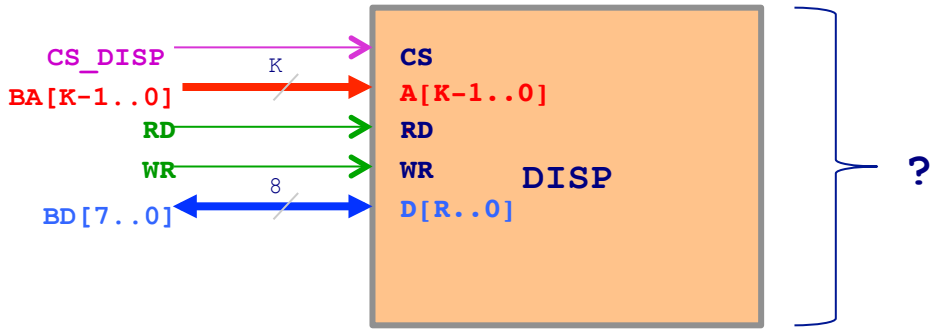
- K (numero di bit di indirizzo *interni* al dispositivo) è fortemente variabile al variare del dispositivo:
  - In generale nei dispositivi di input/output (i.e., le interfacce) K è piccolo (e.g., 2)
  - in generale nei dispositivi di memoria K è grande (e.g., per una RAM da 128 KB si ha K =17)

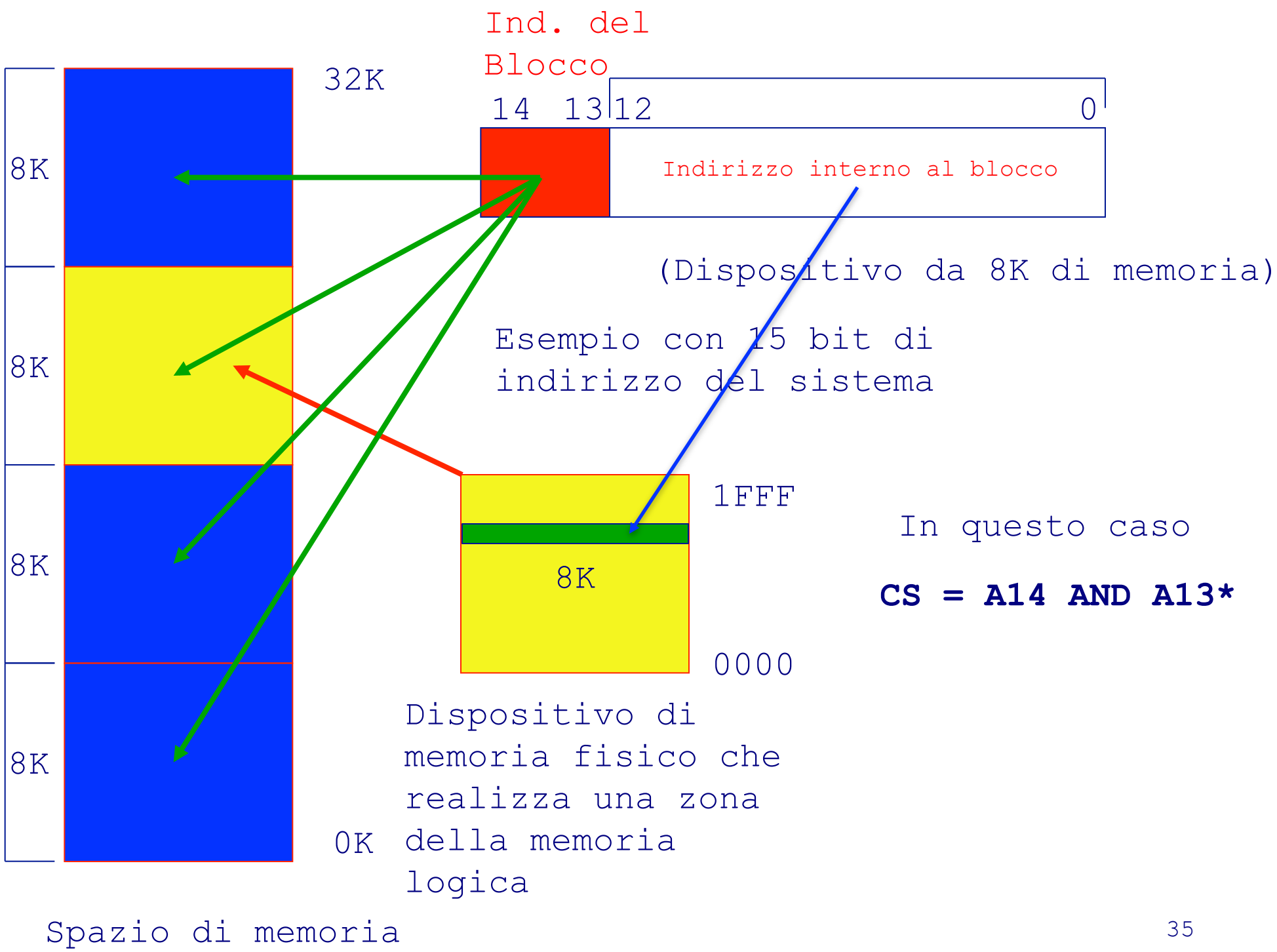
Caratteristiche ai morsetti di un dispositivo indirizzabile su una finestra di n = 2^K byte

Qualunque dispositivo da 8 bit con all'interno n = 2^k elementi indirizzabili separatamente ha al suo interno un decoder (II livello) di K variabili con ingresso di enable che seleziona i singoli oggetti indirizzabili

- Read (RD), detto anche Output Enable (OE) è il comando di lettura.
  Quando RD e CS sono attivi, il dispositivo espone il su BD[7..0] il contenuto della cella indirizzata

- Write (WR), è il comando di scrittura. Quando CS asserito sul fronte di discese di WR è campionato il daato presente su BD[7..0]




Ind. del Blocco
32K
14 13 12 0
Indirizzo interno al blocco
8K
8K
8K
8K
(Dispositivo da 8K di memoria)
Esempio con 15 bit di indirizzo del sistema
1FFF
8K
0000
In questo caso
CS = A14 AND A13*
Dispositivo di memoria fisico che realizza una zona della memoria logica
0K
Spazio di memoria

Mapping allineato di dispositivi da 8 bit in sistemi con bus dati da 8 bit

Si consideri un dispositivo D di n=2^K byte indirizzabili

- Si dice che D è mappato all'indirizzo A se gli indirizzi dei byte di D sono compresi tra A e A+(n-1), cioè se A è l'indirizzo più basso tra tutti gli indirizzi associati a D
- Si dice che D è allineato se A è un multiplo di n (numero di bytes interni al dispositivo), cioè se:

  (indirizzo più basso di D) MOD n = 0 (condizione di allineamento)
- Se D è allineato allora i k bit meno significativi di A sono uguali a zero

Esempi:

- Un dispositivo da due byte è allineato se è mappato a un indirizzo pari
- Una dispositivo da 8 byte è allineato se è mappato a un indirizzo il cui valore codificato in binario termina con 3 zeri
- Un dispositivo da 16 byte è allineato se il suo indirizzo iniziale in codice esadecimale ha la cifra meno significativa uguale a zero
- Un dispositivo da 64 KB è allineato se il suo indirizzo in codice esadecimale ha le quattro cifre meno significative uguali a zero

Come individuare univocamente una finestra allineata di 2^K byte in uno spazio di indirizzamento

- Supponiamo di mappare un dispositivo D di 2^k bytes (k=4) a un indirizzo A allineato di uno spazio di indirizzamento di 1 MB (bus di indirizzi di 20 bit):
- Allora possiamo porre A = α ## $(0)^k$ (ex F8570) ove α è una configurazione binaria di *20 - K* bit e gli indirizzi associati a D saranno compresi tra

  Amin = A = α ## $(0)^k$ e Amax = Amin + 2^k -1 = α ## $(1)^k$

  (Amin = F8570 – Amax = F857F)

- Dunque, possiamo indicare l'indirizzo Ai dell'i-esimo byte di D come l'insieme di due campi concatenati:

  Ai = α ## i (Ai = F8573)

α individua tra le 2^(20-K) finestre allineate di 2^K byte presenti nello spazio di indirizzamento, quella su cui è mappato (a = F857)

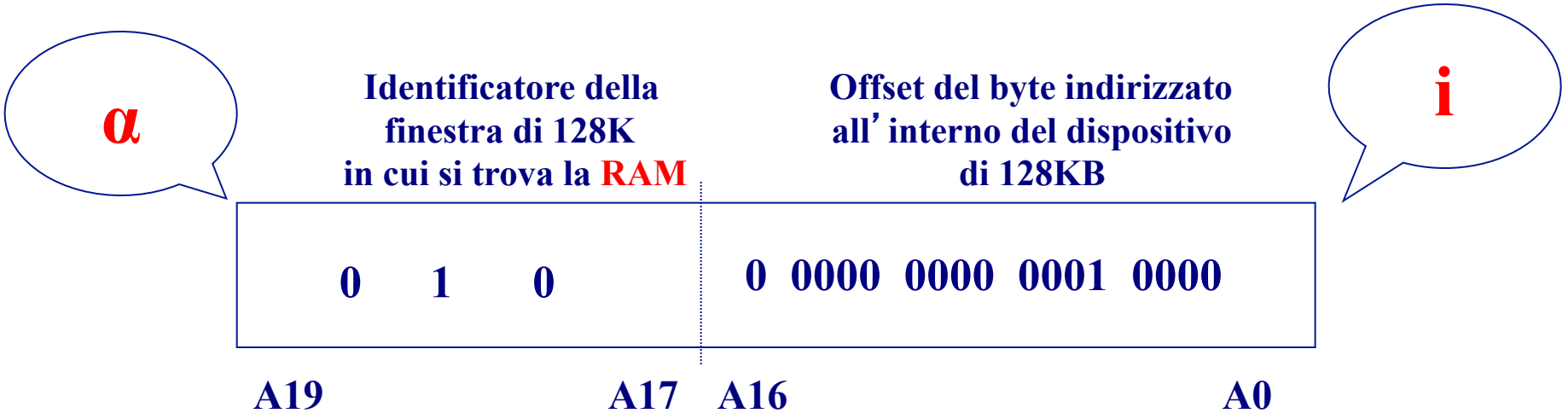
i individua l'offset nel chip del byte indirizzato (i = 3)

(*NB ## è l'operatore simbolico concatenazione*)

Campi in cui si suddivide l'indirizzo di dispositivi mappati in uno spazio di indirizzamento - esempio n. 1

Indirizzamento di un byte di una RAM all'indirizzo 40010H in uno spazio di indirizzamento di 1 MB nell'ipotesi di disporre di un chip da 128 KB mappato all'indirizzo 40000H:

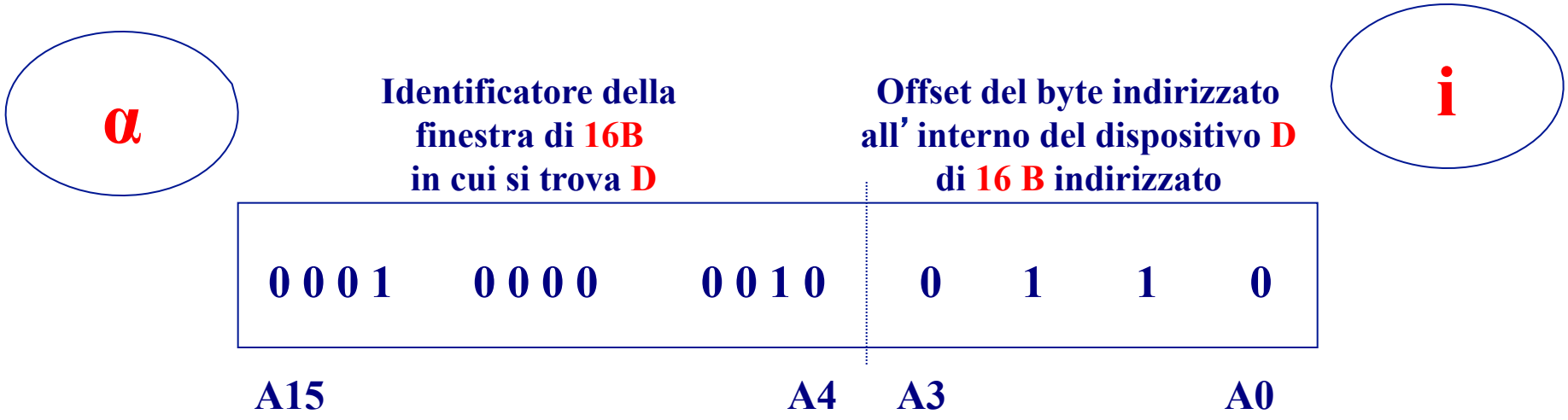
L'indirizzo viene suddiviso in due campi: il primo identifica la finestra di 128 KB in cui è mappata la RAM, il secondo identifica l'offset all'interno della RAM



Campi in cui si suddivide l'indirizzo di dispositivi mappati in uno spazio di indirizzamento - esempio n. 2

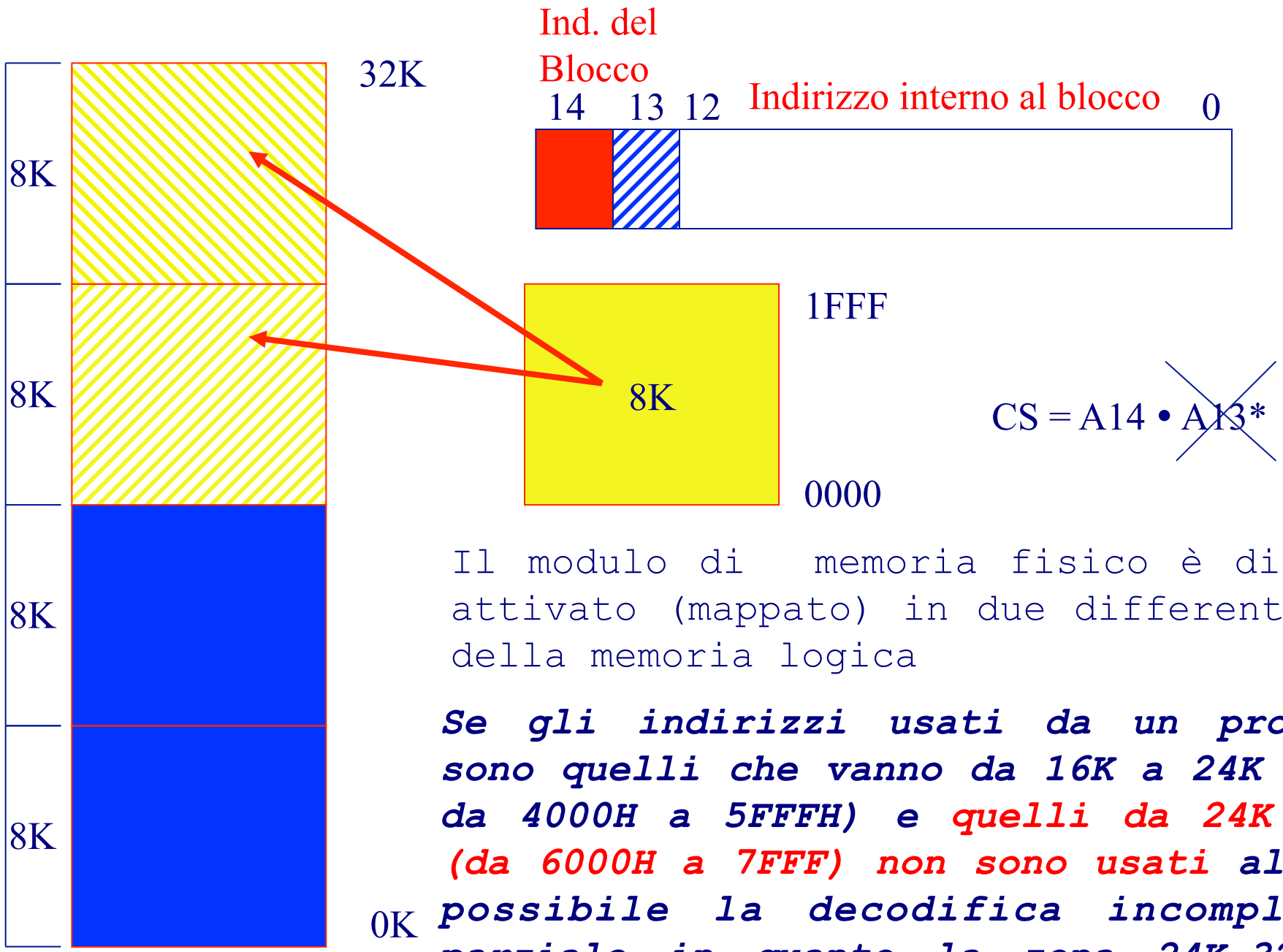
Indirizzamento di un byte all'indirizzo 1026H in un dispositivo di I/O di 16 byte mappato all'indirizzo 1020H di uno spazio di indirizzamento di 64 KB

L'indirizzo viene suddiviso in due campi: il primo identifica la finestra di 16 B in cui è mappato il dispositivo, il secondo identifica l'offset nel dispositivo



Decodifica degli indirizzi in caso di mapping allineato

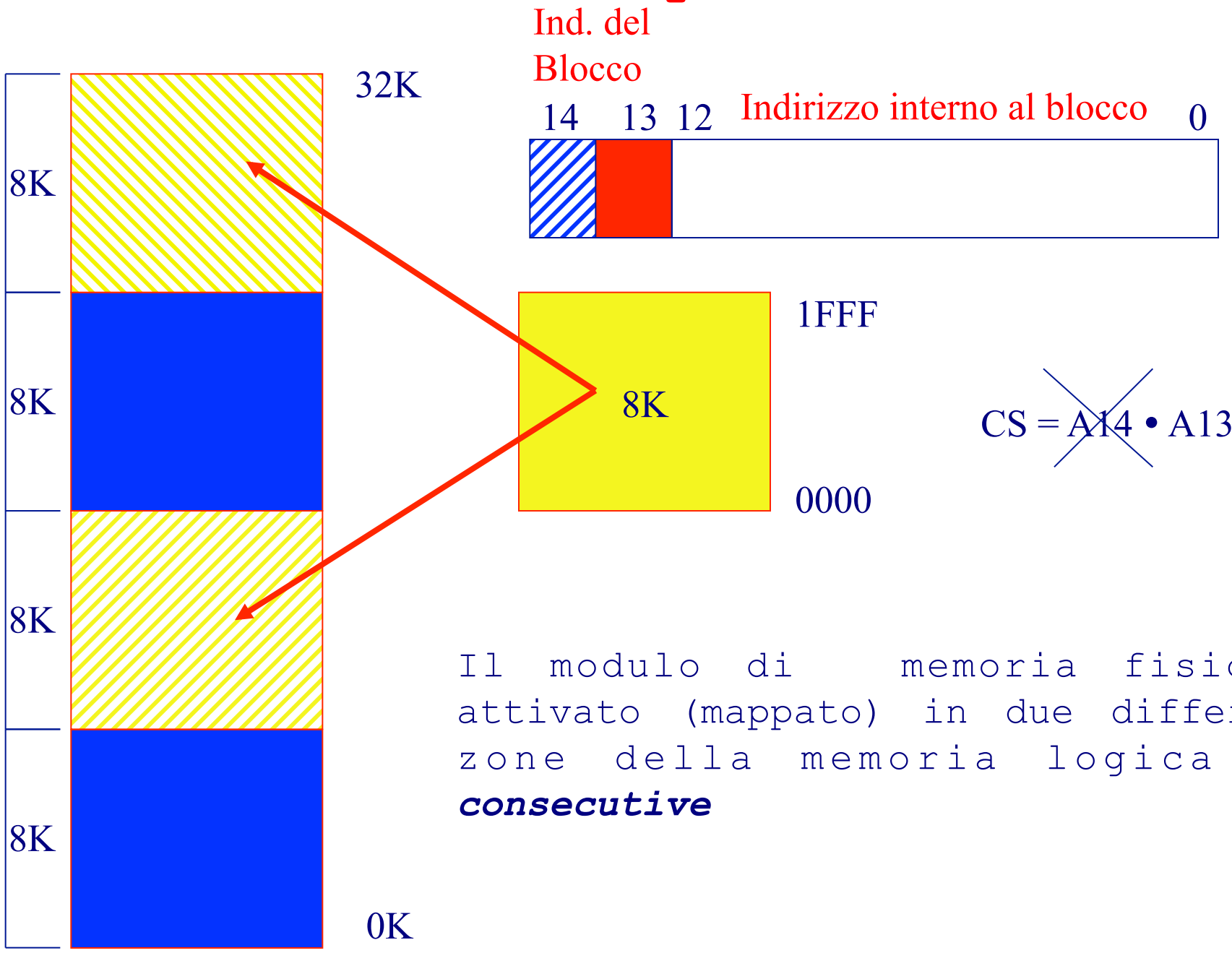
- Consideriamo uno spazio di indirizzamento di 1 MB in cui sia mappato un dispositivo di 2^K byte
- Per individuare una cella di indirizzo Ai = α ## i possiamo decodificare tutti i 20 bit che compongono Ai
- Questa decodifica è effettuata ricorrendo alla struttura dei decoder ad albero, con albero di due livelli:
  - Il I livello è usato per decodificare α (che identifica la posizione in cui il chip è mappato); per decodificare α dobbiamo decodificare 20-K variabili
  - il II livello viene utilizzato per decodificare i (che identifica il byte all'interno del chip, serve un decoder di k variabili)
- Il decoder di II livello si trova all'interno del chip mentre la decodifica di α è a carico del progettista del sistema che può utilizzare un decoder di 20-k variabili con cui si decodifica α
- La decodifica è completa se si utilizzano tutti i 20-K bit per decodificare α, semplificata se si utilizza solo un sottoinsieme (minimo) dei 20-K bit



Il modulo di memoria fisico è di fatto attivato (mappato) in due differenti zone della memoria logica

Se gli indirizzi usati da un programma sono quelli che vanno da 16K a 24K (overo da 4000H a 5FFFH) e quelli da 24K a 32K (da 6000H a 7FFF) non sono usati allora è possibile la decodifica incompleta o parziale in quanto la zona 24K-32K non viene mai indirizzata. Espressione CS più semplice

Decodifica parziale 2/2



Il modulo di memoria fisico è attivato (mappato) in due differenti zone della memoria logica ***non consecutive***

Spazio di memoria

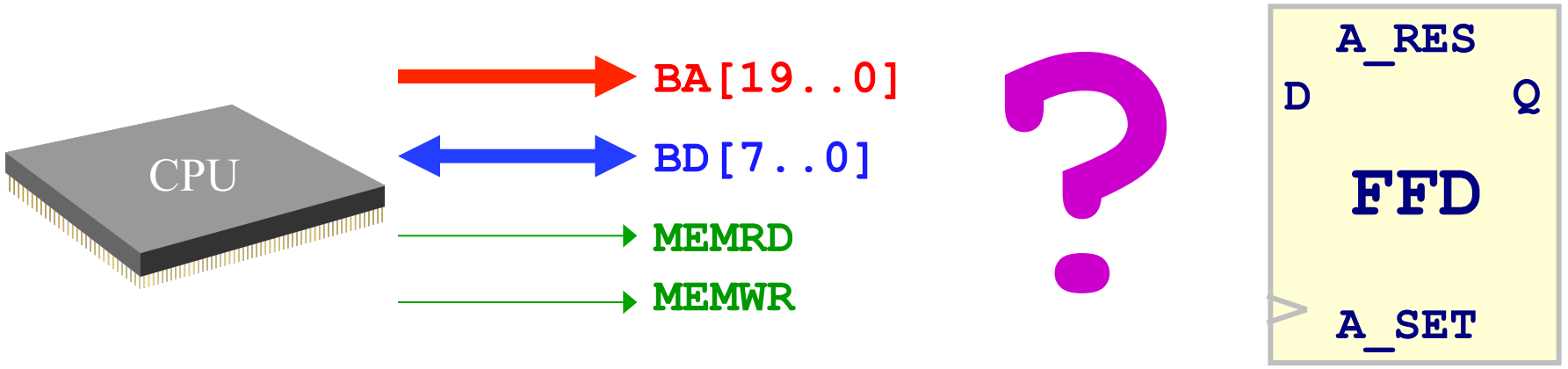
Esercizio

Si consideri un sistema con bus indirizzi a 16 bit e bus dati a 8 bit. Scrivere le espressioni di decodifica completa e semplificata (quella da usare all'esame) nei seguenti casi:

1) **Dispositivo di memoria da 16 KB mappato a 8000h**
2) **Dispositivo di memoria da 8 KB mappato a 0000h**
3) **Entrambi i dispositivi precedenti**

Se c'è un solo dispositivo (casi 1 e 2) il CS è molto particolare….

Mapping, read, write e set/reset di un FFD



- Il FFD è un elementare dispositivo di memoria
- Con una CPU, come possiamo:
  - **scrivere** nel FFD
  - **leggere** nel FFD
  - **settare o resettare** in modo asincrono il FFD

Consideriamo il caso di una **CPU con bus dati a 8 bit** con **20 bit di indirizzo. 64 K di RAM agli indirizzi alti** e **64 K di EPROM agli indirizzi bassi**

Nelle pagine seguenti assumiamo che i comandi del FFD e del latch siano mappati nei seguenti indirizzi:

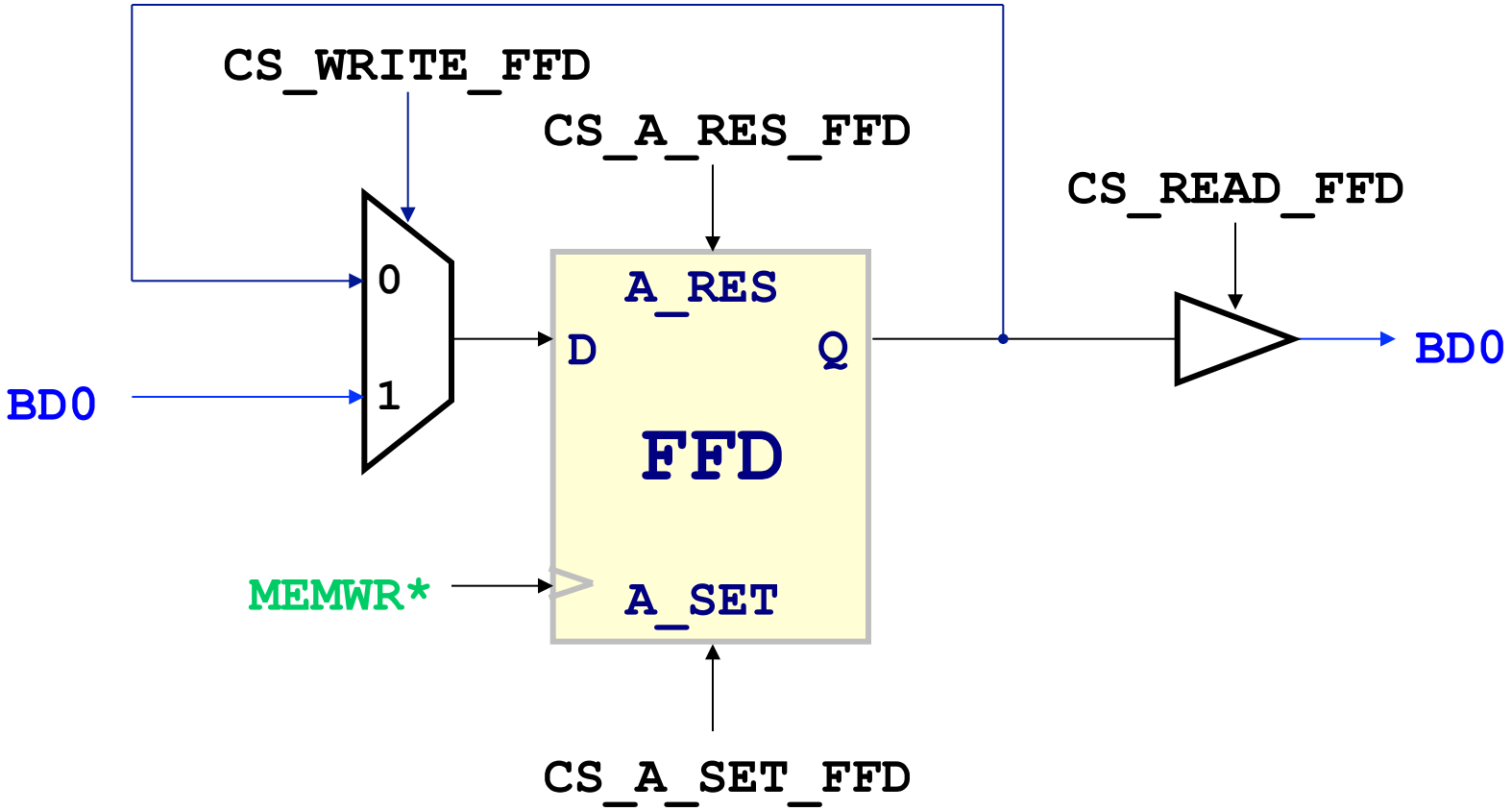
CS_READ_FFD -> 80003h
CS_WRITE_FFD -> 80002h
CS_A_RES_FFD -> 80001h
CS_A_SET_FFD -> 80000h

Assumiamo inoltre di utilizzare il segnale **BD0** del bus dati per leggere e scrivere il singolo bit di dato.

Ovviamente sarebe possibile utilizzare altri indirizzi non appartenenti alle memorie e anche altri segnali del bus dati (anche diversi per letture e scritture).

Se il testo dell'esame non specifica quali indirizzi usare la scelta è lasciata allo studente.

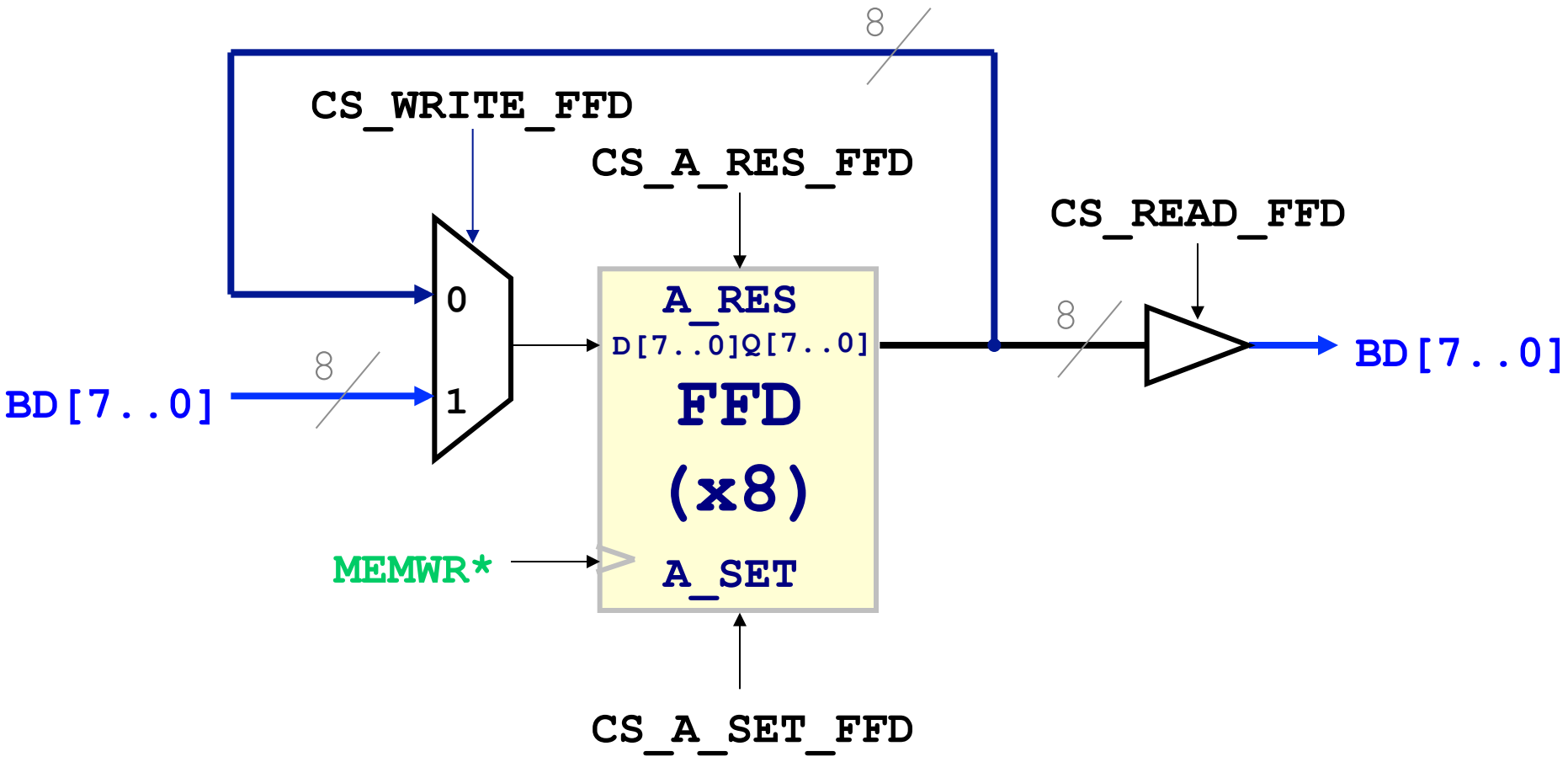
Spesso, la scelta degli indirizzi semplifica/complica i segnali di decodifica.




```
CS_RAM_H       = BA19·BA18·BA15
CS_RAM_H       = BA19·BA18·BA15*
CS_READ_FFD    = BA19·BA18*·BA1·BA0·MEMRD      (ist. lettura)
CS_WRITE_FFD   = BA19·BA18*·BA1·BA0*           (ist. scrittura)
CS_A_RES_FFD   = BA19·BA18*·BA1*·BA0·MEMWR     (ist. scrittura)
CS_A_SET_FFD   = BA19·BA18*·BA1*·BA0*·MEMWR    (ist. scrittura)
CS_EPROM       = BA19*
```

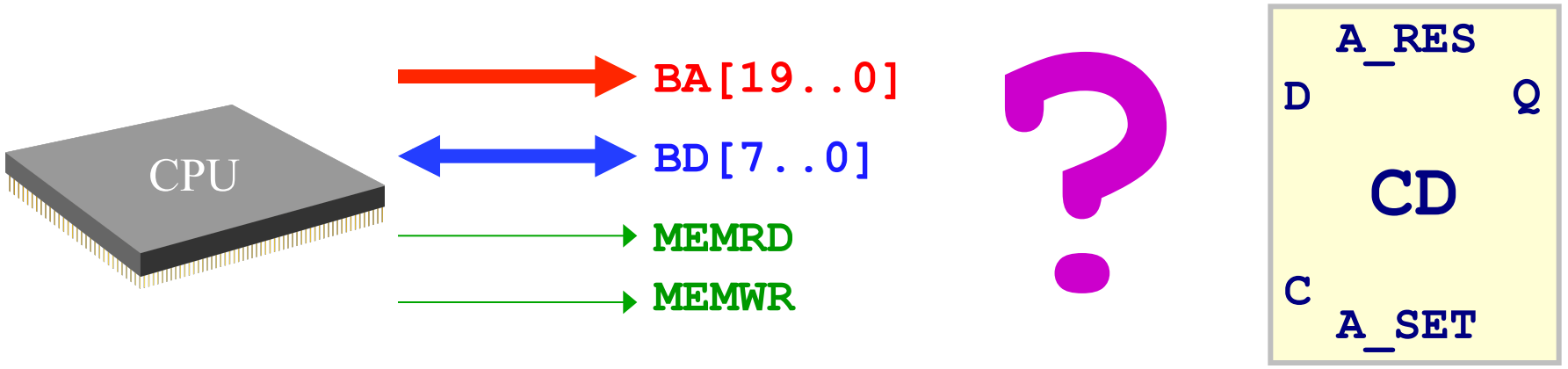
Vedremo che istruzioni di lettura e scrittura sono (risp.) *load byte* (LB) e *store byte* (SB).

Estensione a 8 bit



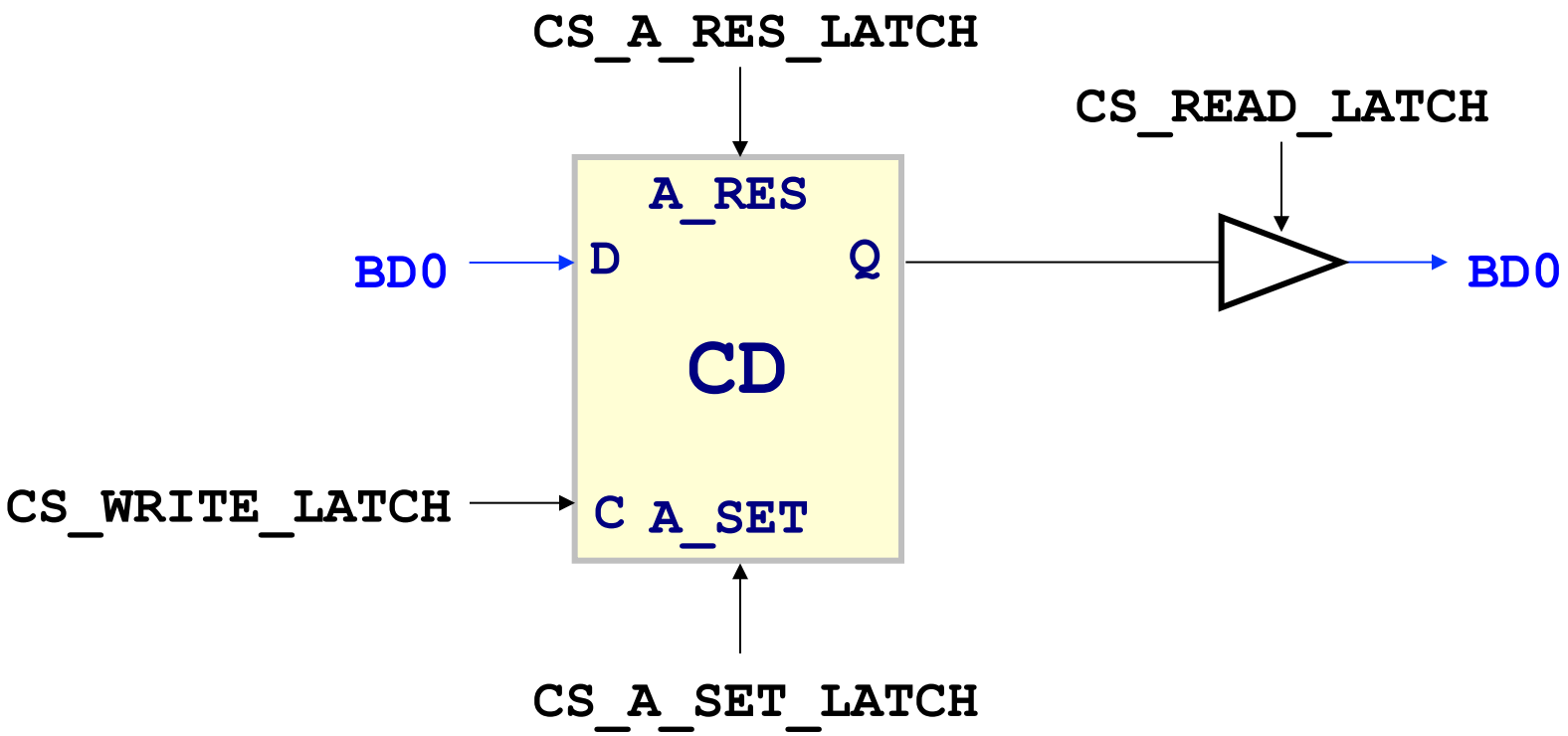
Stessi CS della pagina precedente, cambia solo il numero di bit di dato trasferiti.

Mapping, read, write e set/reset di un latch



- Anche il latch CD è un elementare dispositivo di memoria
- Con una CPU, come possiamo:
  - **scrivere** nel latch
  - **leggere** nel latch
  - **settare o resettare** in modo asincrono il latch

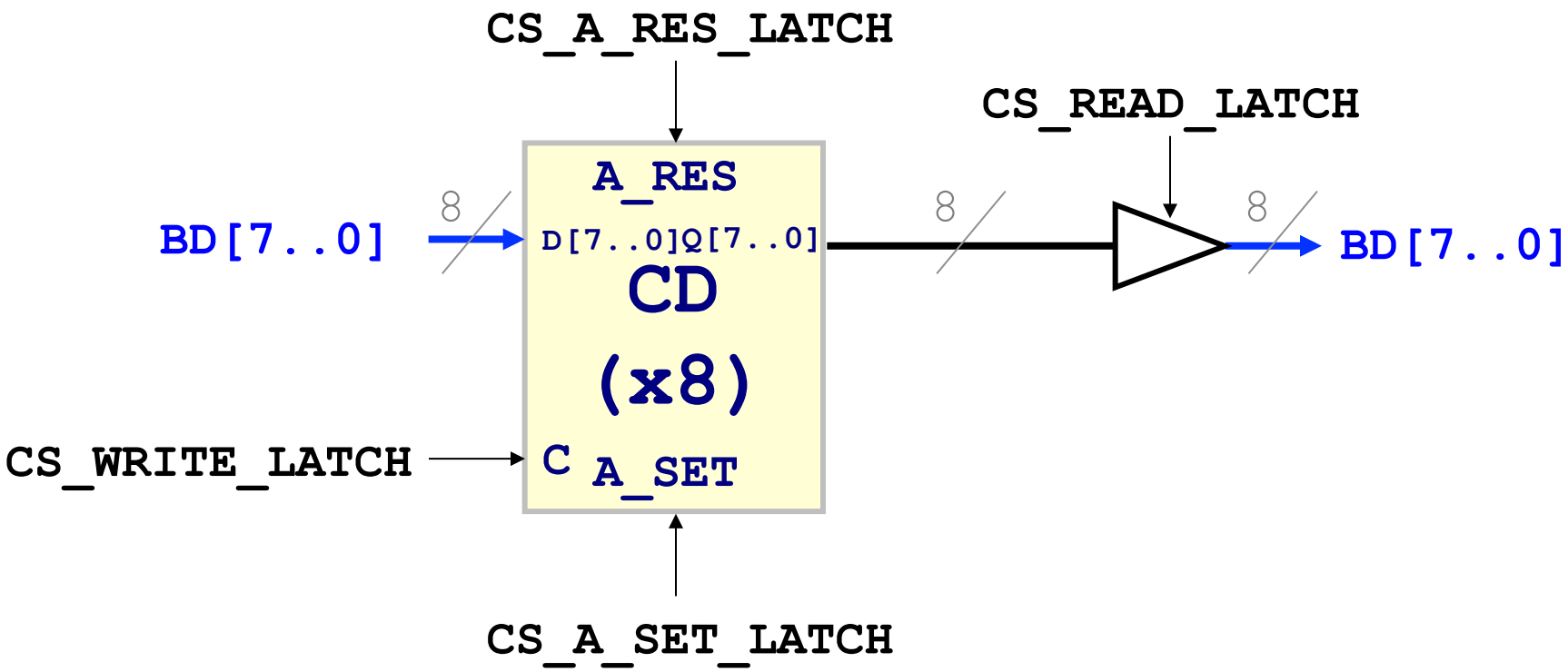
Consideriamo il caso di una **CPU con bus dati a 8 bit** con **20 bit di indirizzo. 64 K di RAM agli indirizzi alti** e **64 K di EPROM agli indirizzi bassi**




```
CS_RAM_H        = BA19·BA18·BA15
CS_RAM_H        = BA19·BA18·BA15*
CS_READ_LATCH   = BA19·BA18*·BA1·BA0·MEMRD  (ist. lettura)
CS_WRITE_LATCH  = BA19·BA18*·BA1·BA0*·MEMWR  (ist. scrittura)
CS_A_RES_LATCH  = BA19·BA18*·BA1*·BA0·MEMWR  (ist. scrittura)
CS_A_SET_LATCH  = BA19·BA18*·BA1*·BA0*·MEMWR  (ist. scrittura)
CS_EPROM        = BA19*
```

Vedremo che istruzioni di lettura e scrittura sono (risp.) *load byte* (LB) e *store byte* (SB).

Estensione a 8 bit

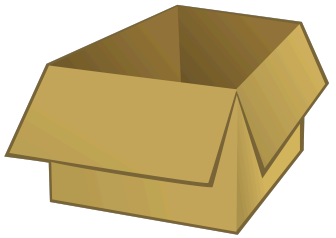


Stessi CS della pagina precedente, cambia solo il numero di bit di dato trasferiti.

Incrementare il parallelismo dei dati

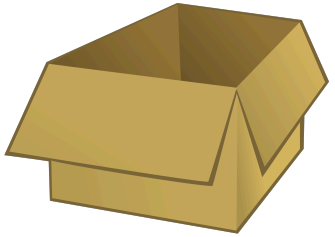
- Abbiamo considerato fino a ora sistemi con un parallelismo (bus dati) a 8 bit
- Ogni trasferimento richiede un ciclo di bus
- N elementi (byte) -> N cicli di bus
- Sappiamo che le memorie (e non solo) sono lente (vs CPU)

i)



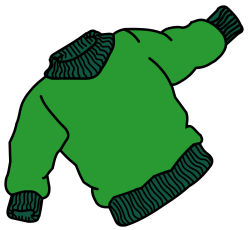
1

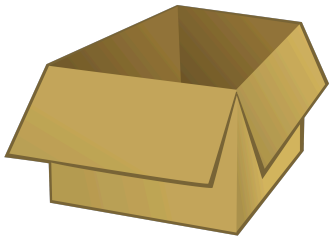
ii)



1

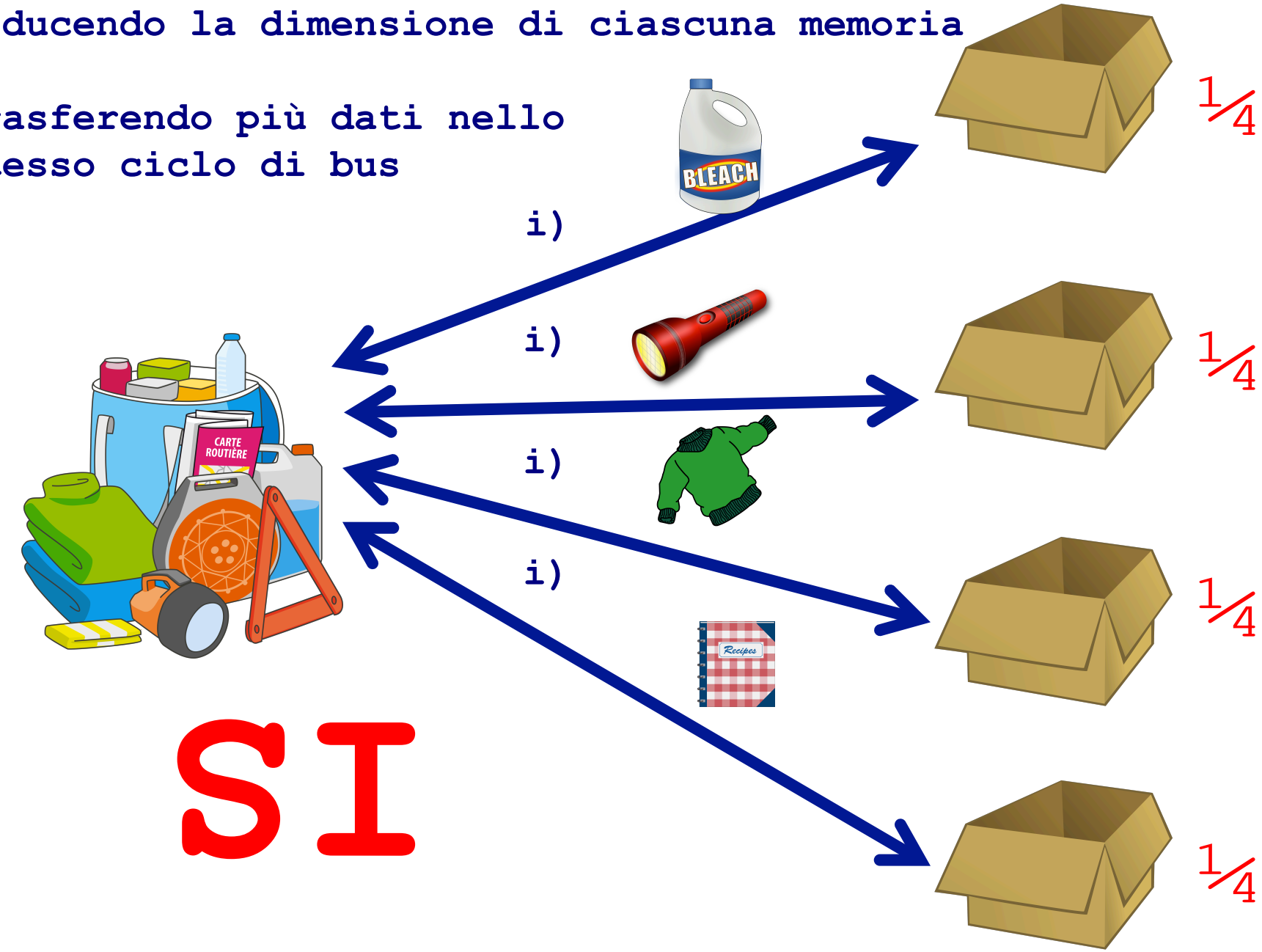
iii)



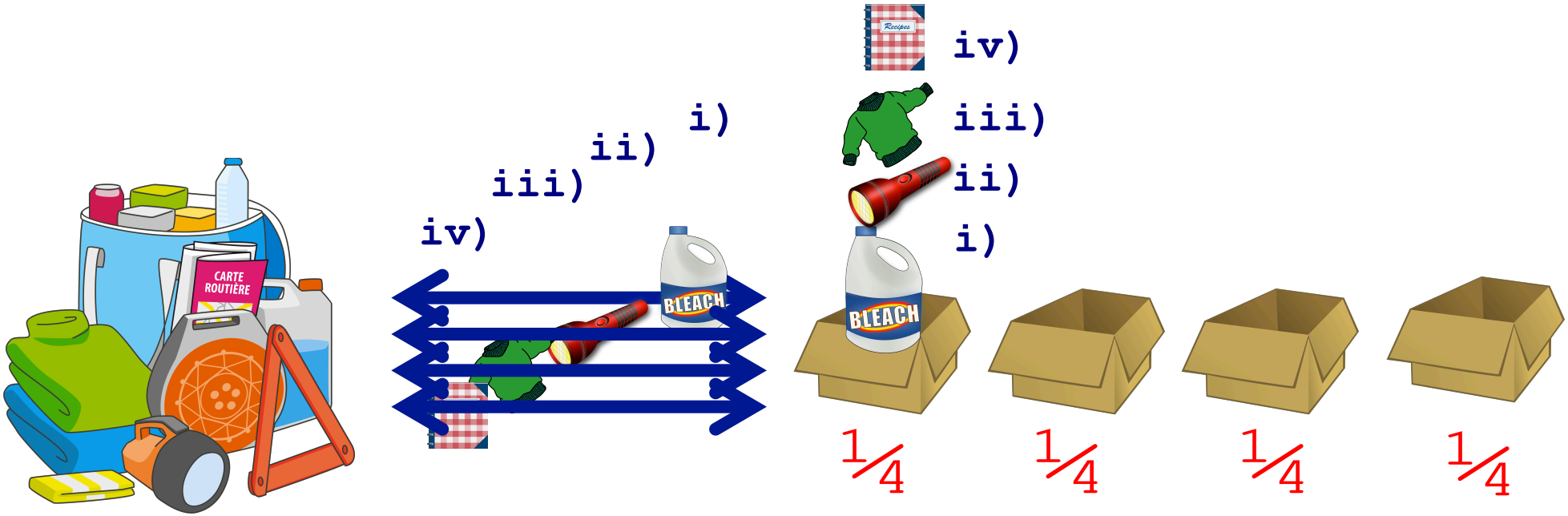


1

- Possiamo fare meglio?
- Si, aumentando il parallelismo dei dati
- Riducendo la dimensione di ciascuna memoria
- Trasferendo più dati nello stesso ciclo di bus



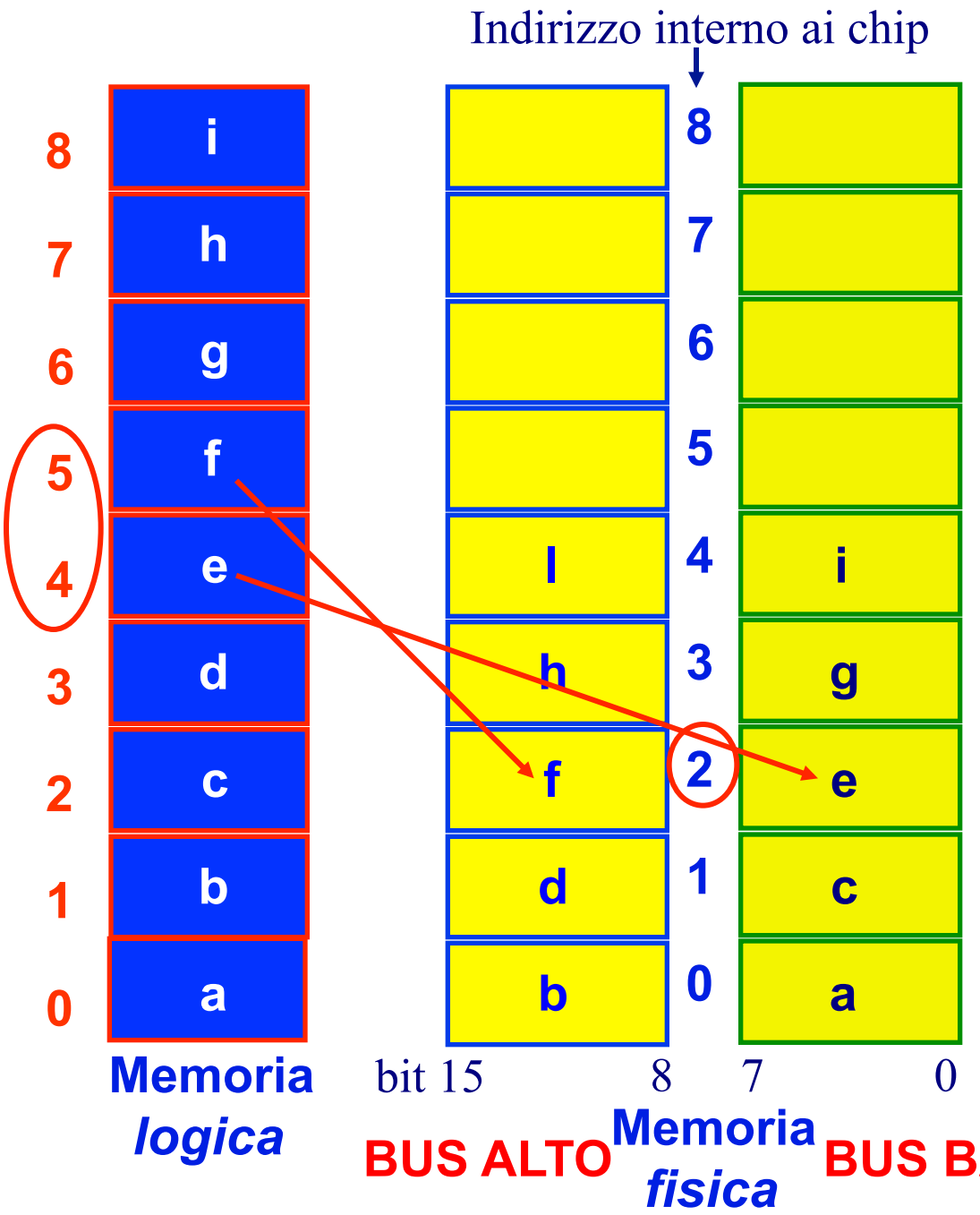
- Cosa NON fare?
- Trasferire gli elementi sequenzialmente in memorie più piccole
- Elementi contigui vanno su memorie diverse



NO

il parallelismo di ciascuna memoria è sempre 8 bit!

Memoria con processori a parallelismo > 8
Il caso dei 16 bit



Indirizzo fisico memorie = Indirizzo logico / 2

Sul piedino A0 della memoria -> BA1 bus
A1 della memoria -> BA2 bus

.....

Word(3) -> Byte$_h$(1) e Byte$_b$(2) -> 2 letture

Logico	Fisico	Fisico
(d ,e)	(d)	(e)

(Lettura bytes 3 e 4 che però stanno a indirizzi fisici interni delle memorie differenti)

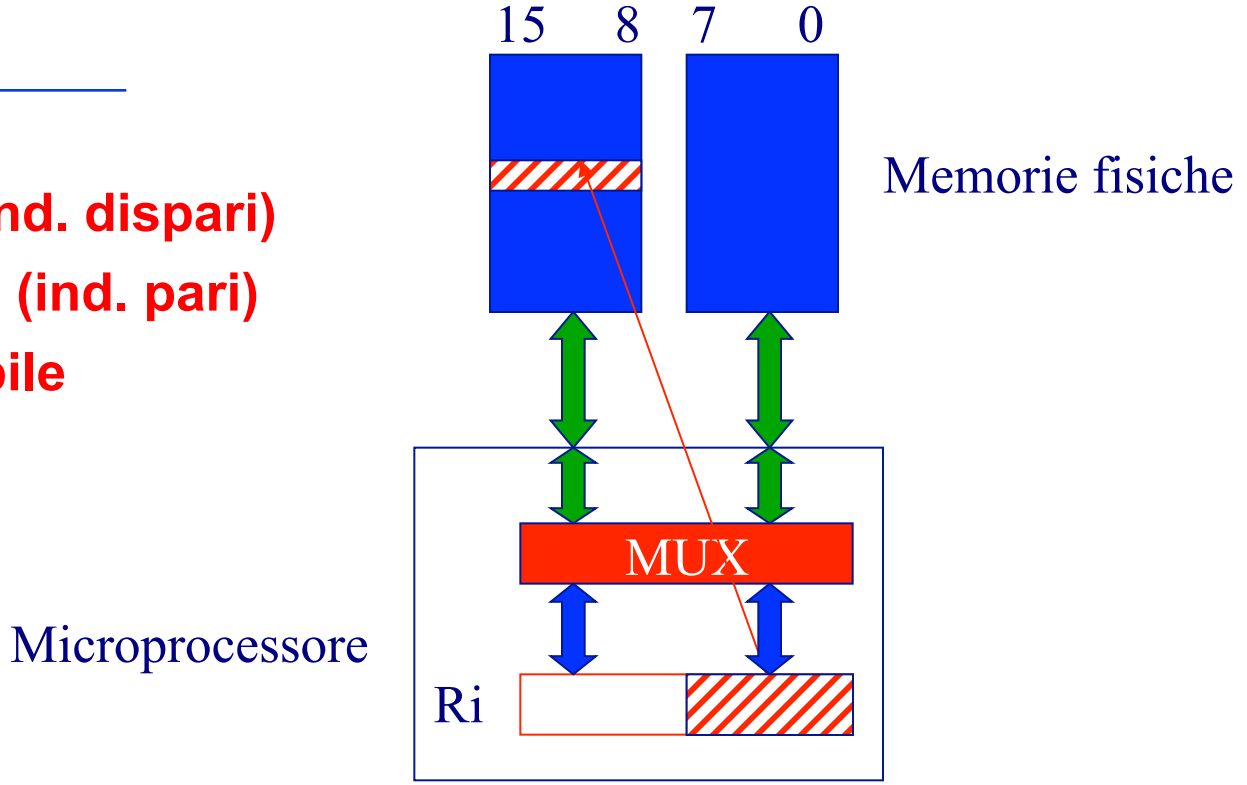
Le memorie fisiche vanno sempre in coppia
Per ogni "banco" ci deve essere un ByteEnable
BE0 per banco 7-0 e BE1 per banco 15-8

Memorie sempre in coppia
Ad esempio 2 x 8K = 16 K

Memorie con bus a 16 bit

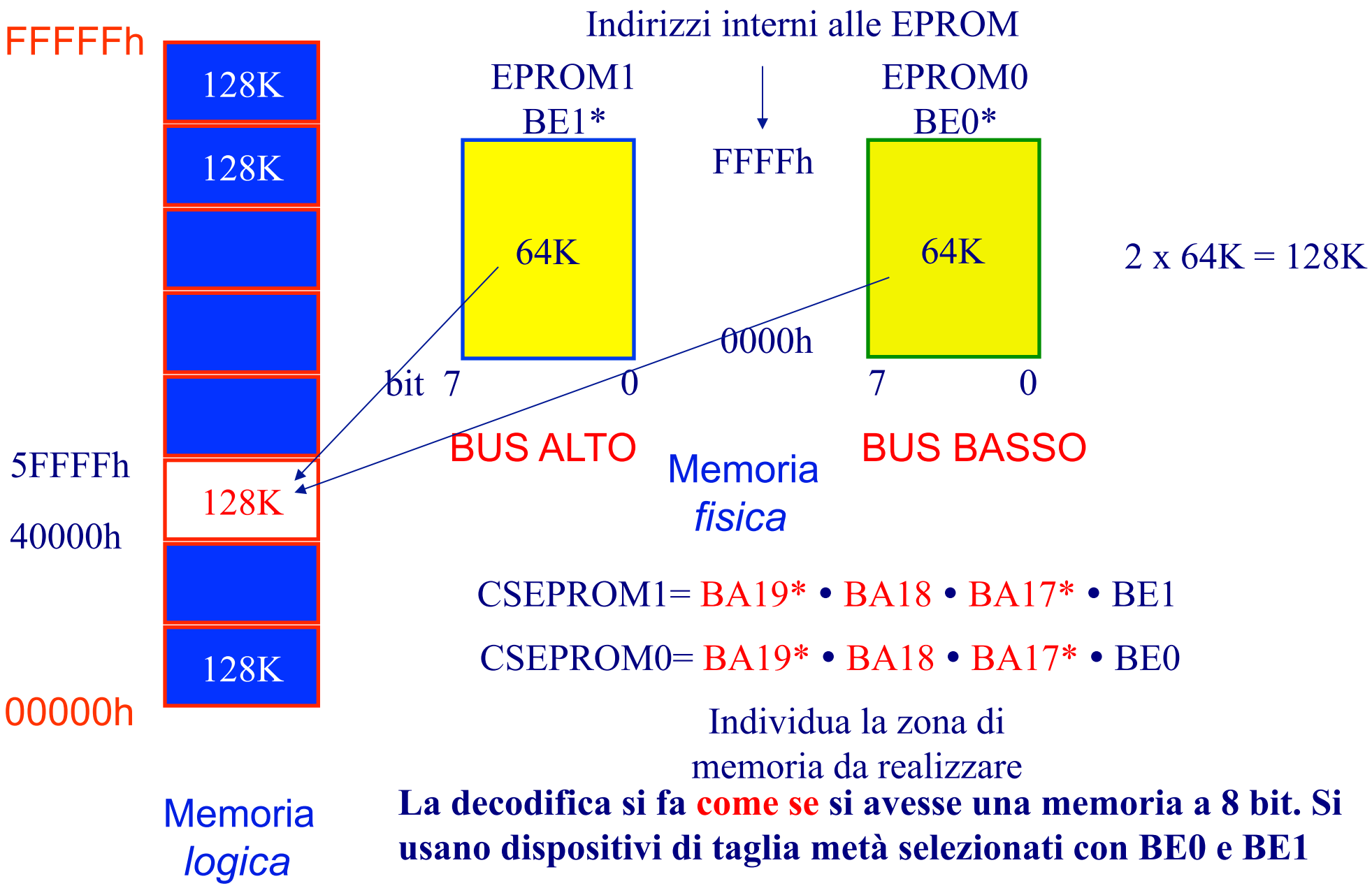
Lo scambio byte alto esterno, byte basso del registro e viceversa avviene *all'interno* del microprocessore

BE1	BE0	
1	1	Word
1	0	Byte alto (ind. dispari)
0	1	Byte basso (ind. pari)
0	0	Non possibile



BA0 ***del processore non viene generato (di fatto seleziona il banco - al suo posto BE0 e BE1)***
BA1 ***del processore connesso ai piedini A0 delle memorie***
BA2 ***del processore connesso ai piedini A1 delle memorie***
etc. etc.

Memoria con processori a parallelismo > 8
Il caso dei 16 bit



CSEPROM1= BA19* • BA18 • BA17* • BE1

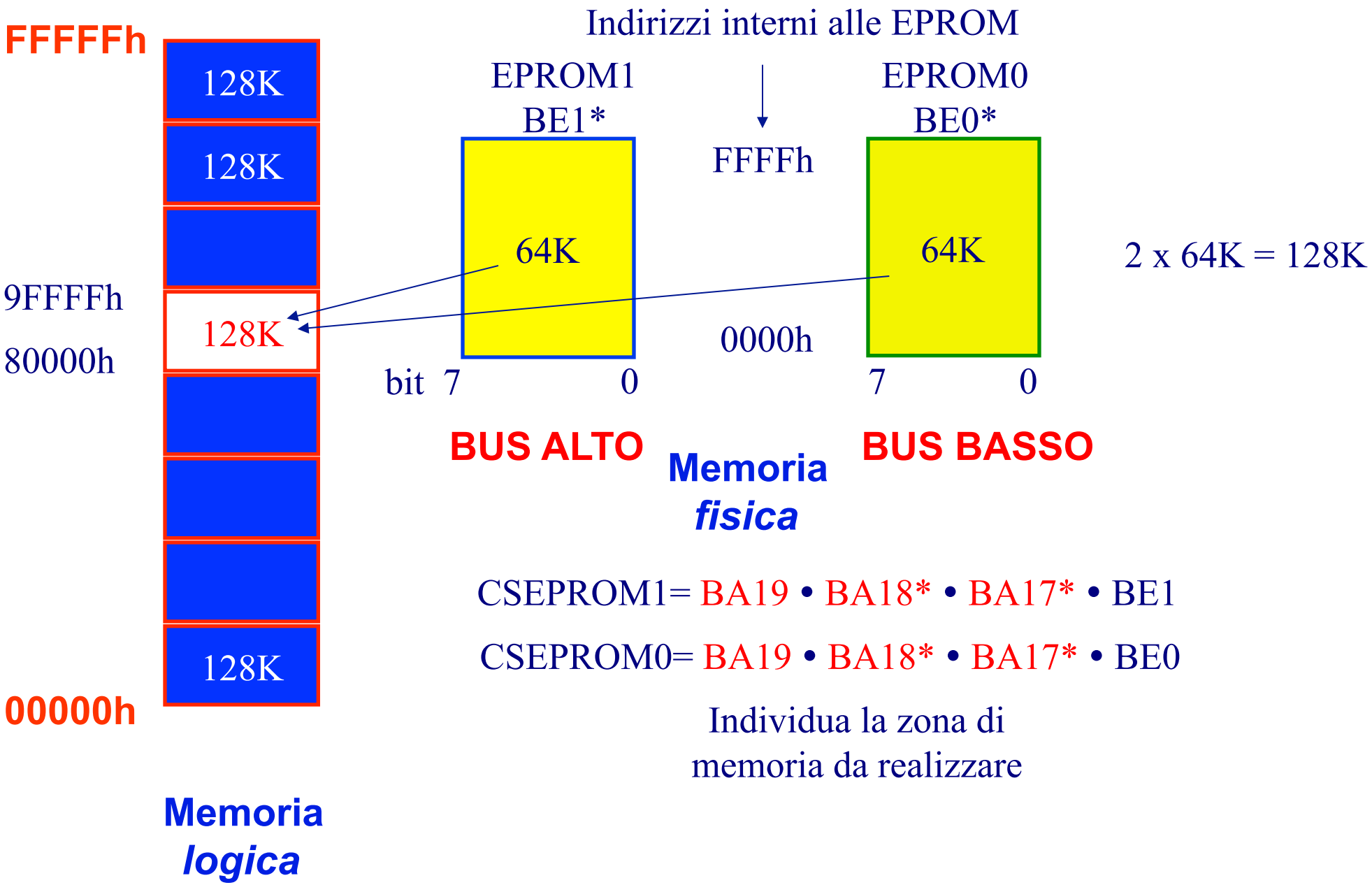
CSEPROM0= BA19* • BA18 • BA17* • BE0

Individua la zona di
memoria da realizzare

La decodifica si fa come se si avesse una memoria a 8 bit. Si usano dispositivi di taglia metà selezionati con BE0 e BE1

Le memorie vanno sempre in coppia (16 bit)

Memoria con processori a parallelismo > 8
Il caso dei 16 bit

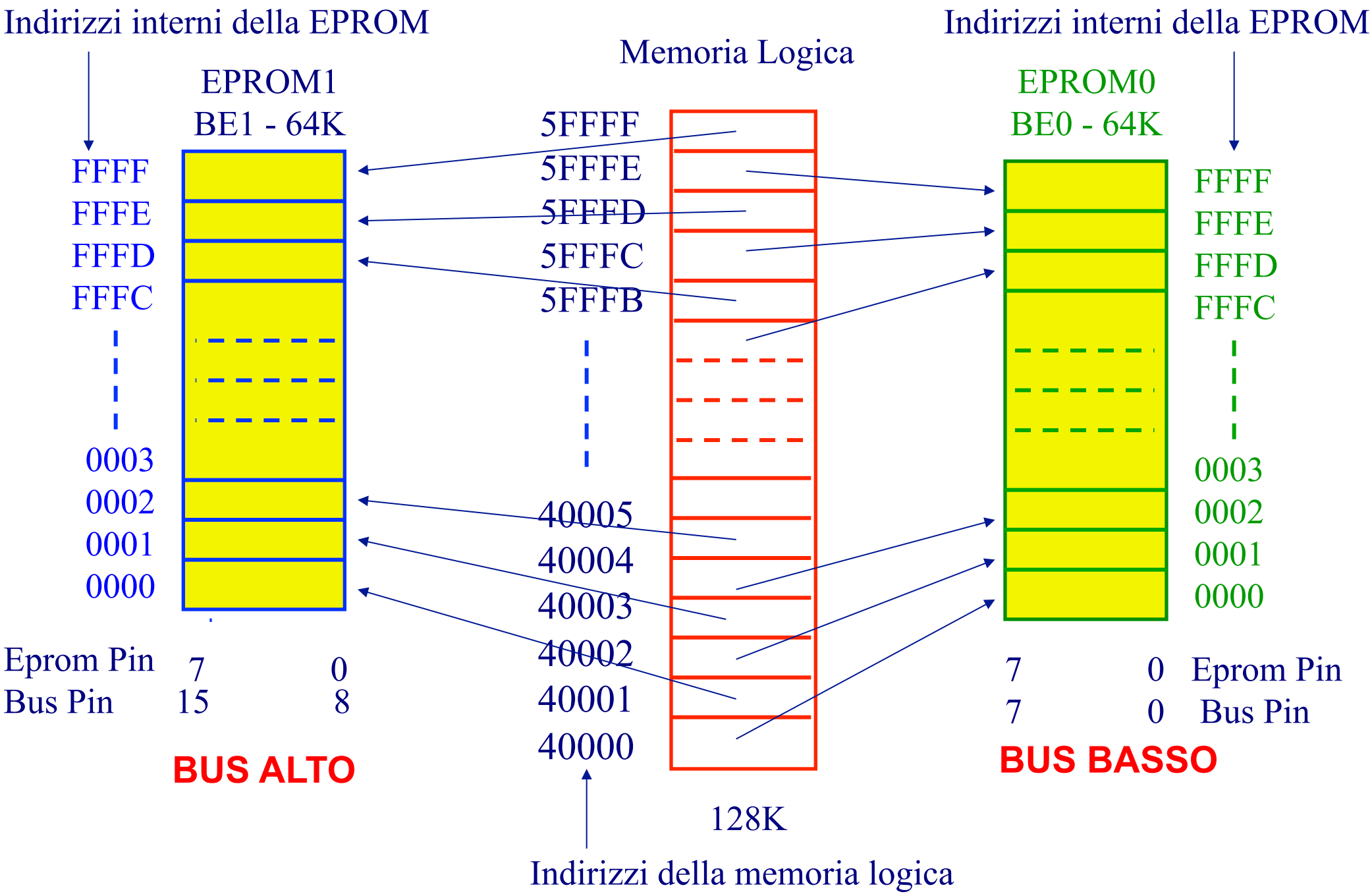


CSEPROM1= BA19 • BA18* • BA17* • BE1

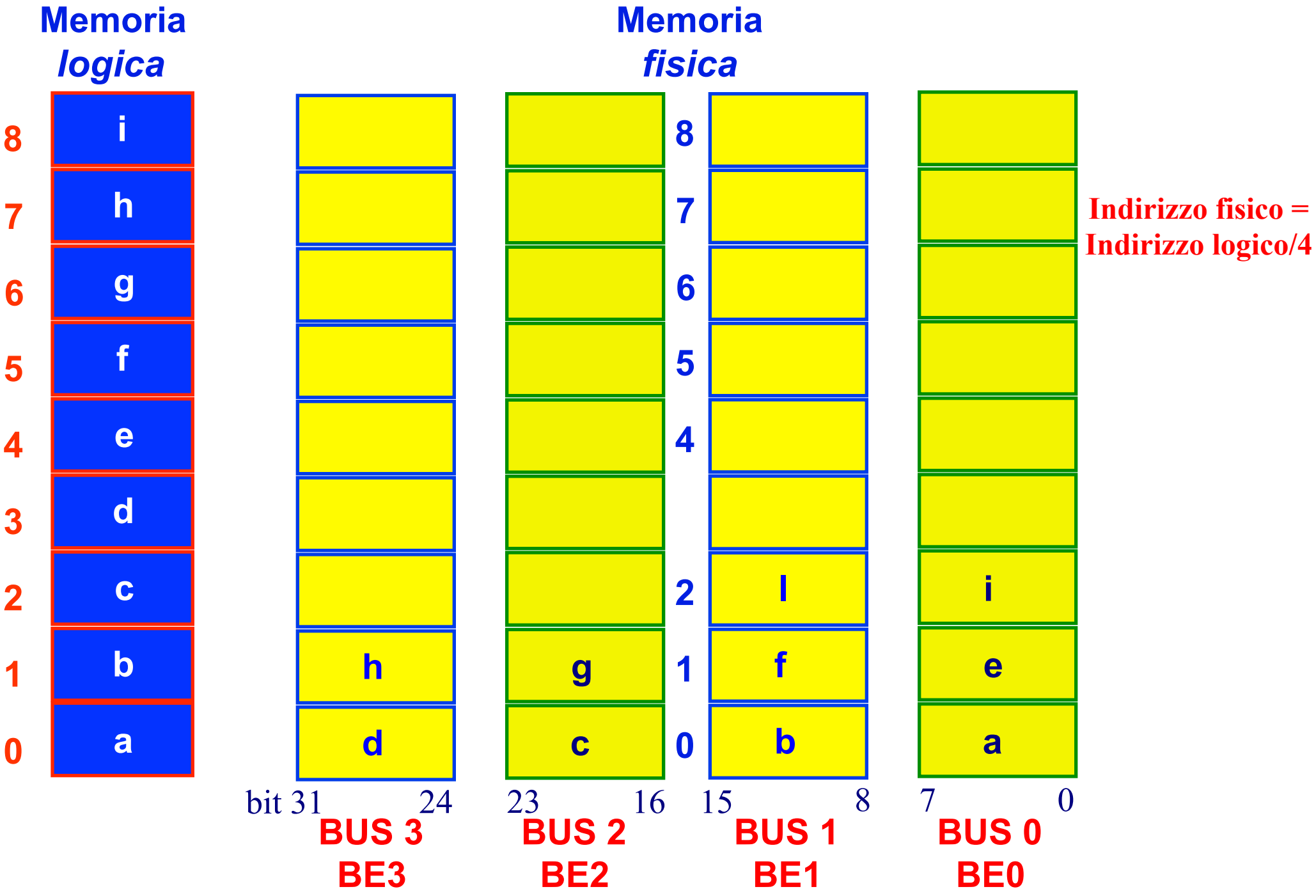
CSEPROM0= BA19 • BA18* • BA17* • BE0

Individua la zona di
memoria da realizzare

Memoria con processori a parallelismo > 8
Il caso dei 16 bit



Memoria con processori a parallelismo 32 bit



Bus enable con parallelismo 32 bit

BE3	BE2	BE1	BE0	
1	1	1	1	Word 32 bit
0	0	1	1	Half word bassa
1	1	0	0	Half word alta
0	0	0	1	Byte 0-7
0	0	1	0	Byte 15-8

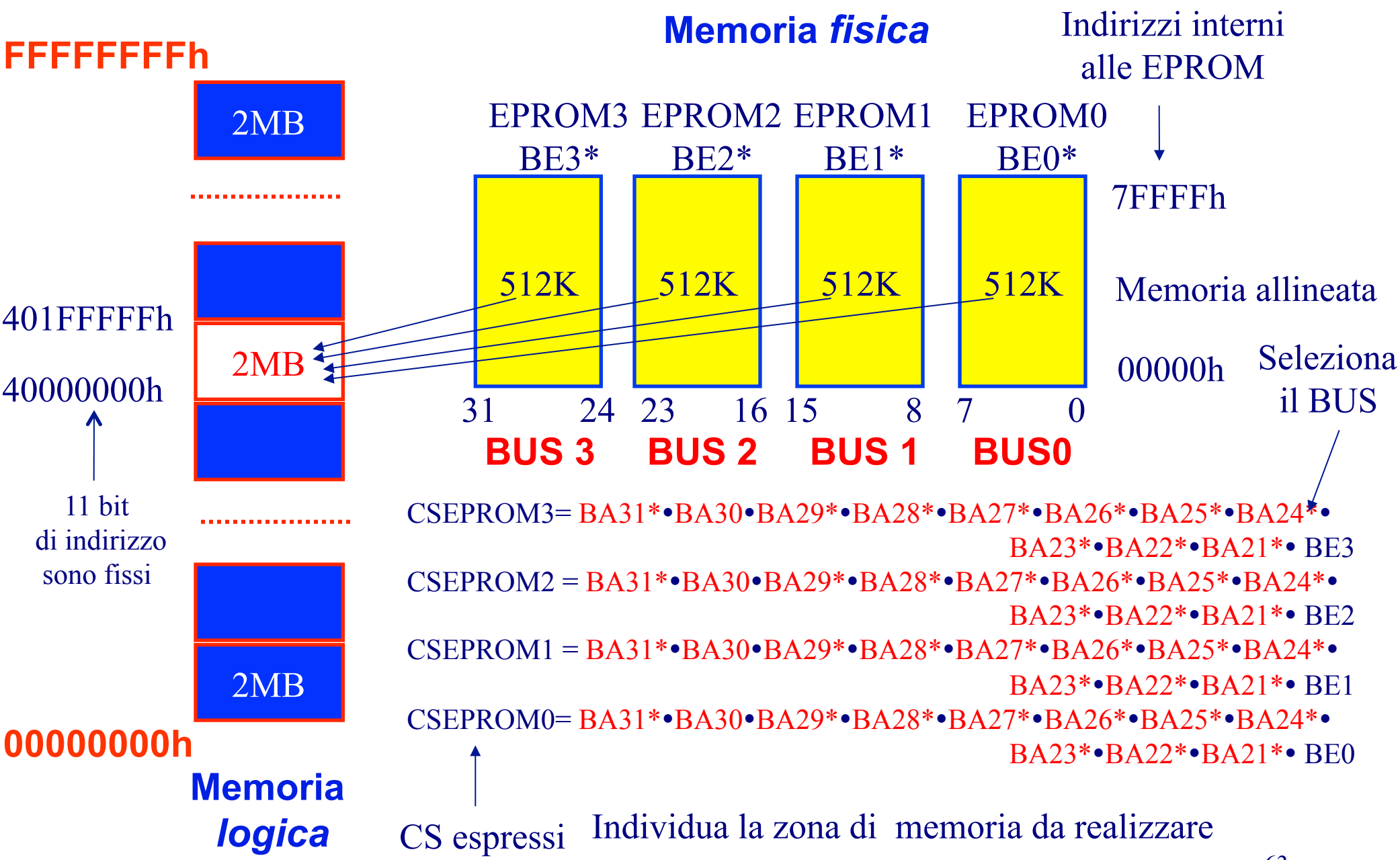
etc.

Lo scambio fra i bytes (half word) dei banchi di memoria e i byte (half word) dei registri e viceversa avviene *all'interno* del microprocessore

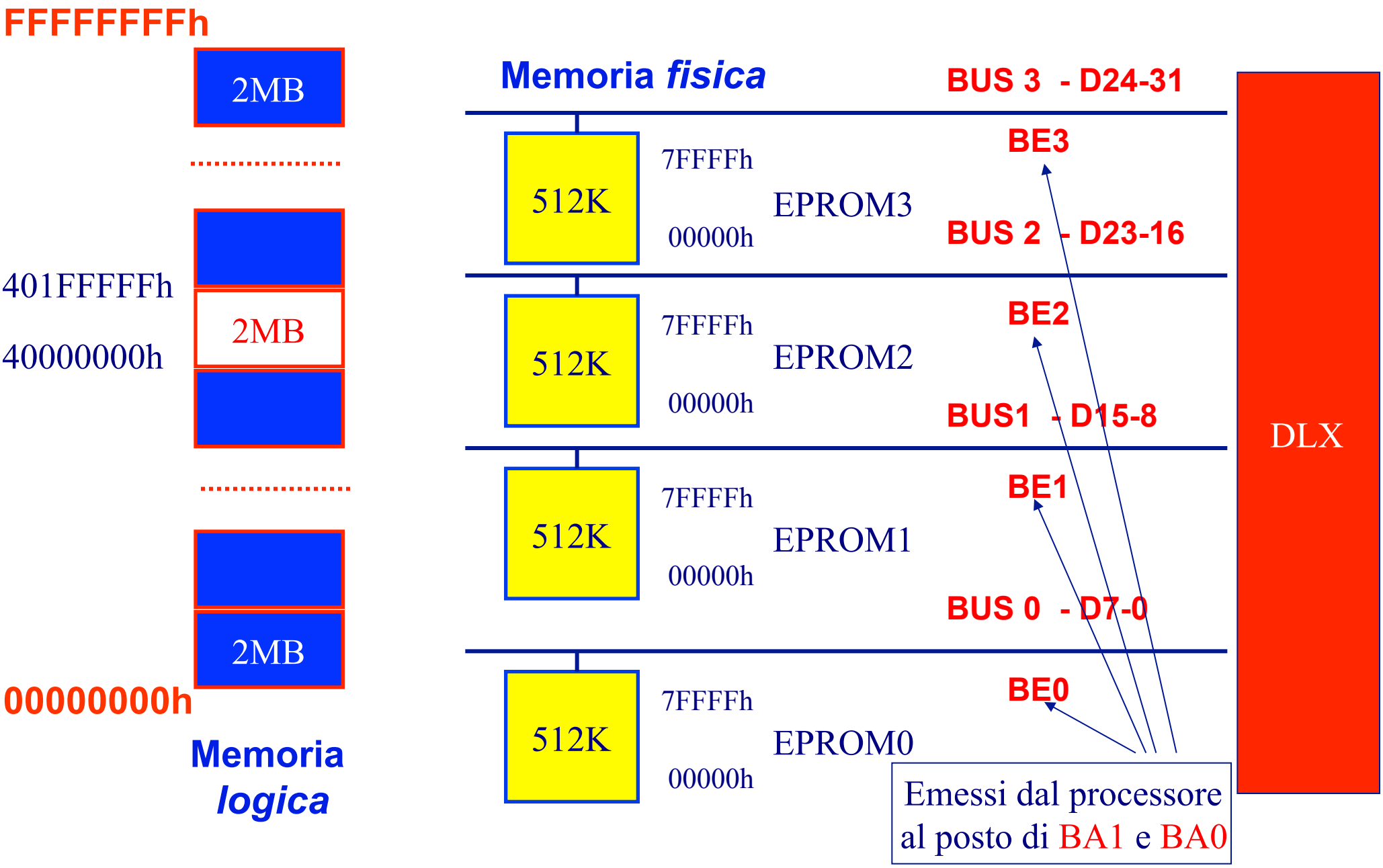
N.B. BA0 e BA1 ***del processore non vengono generati (di fatto selezionano uno dei banchi - al loro posto BE0, BE1, BE2, BE3)***

***BA2** del processore connesso ai piedini **A0** delle memorie*
***BA3** del processore connesso ai piedini **A1** delle memorie*
etc. etc.

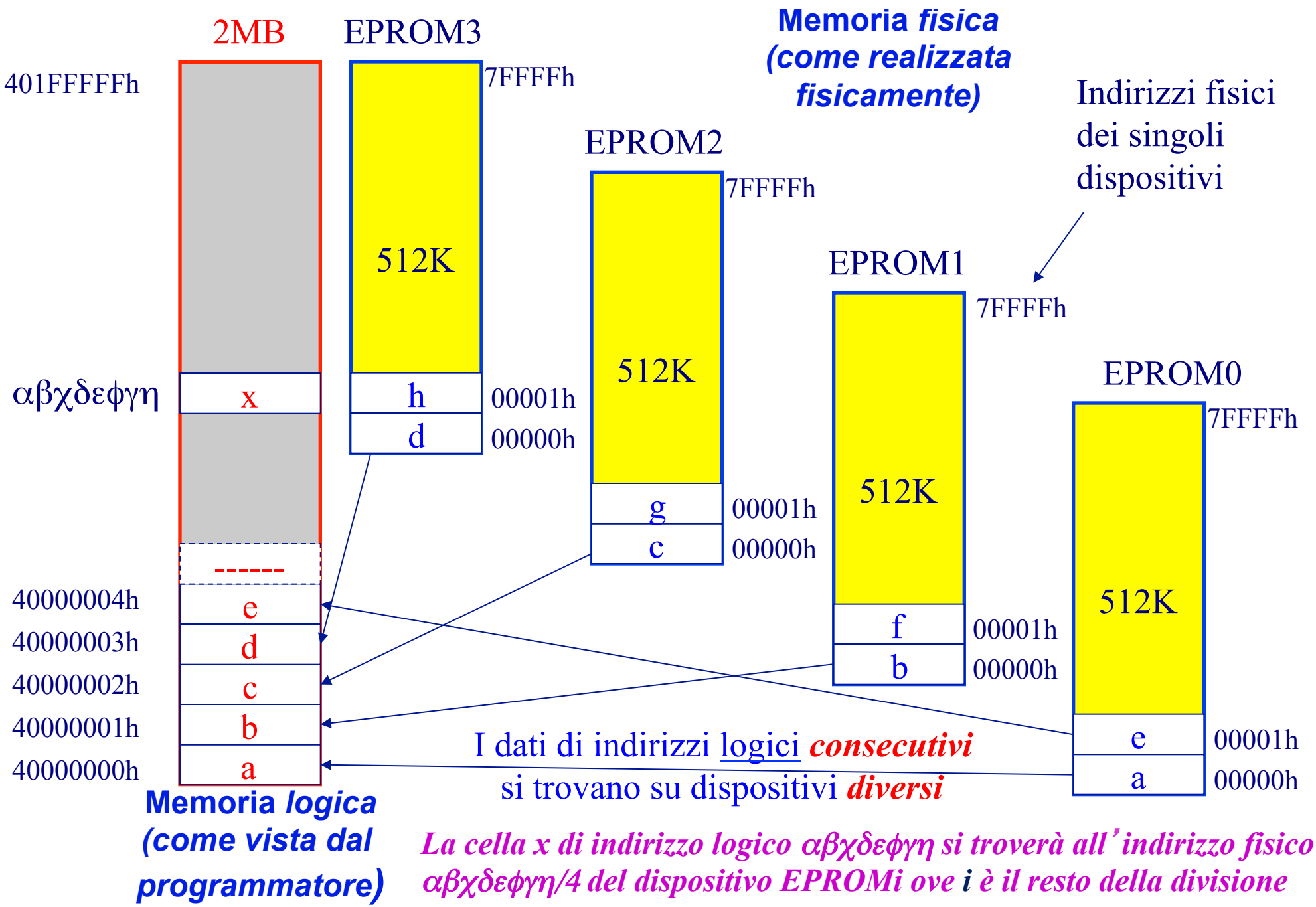
Memorie con parallelismo 32 bit



Memorie con parallelismo 32 bit



Memorie con parallelismo 32 bit



Memorie con parallelismo 32 bit

Esempio: si vuole realizzare nel DLX (bus 32 bit) una memoria RAM da 256K posta all'indirizzo 84000000 (allineata). Campo di indirizzamento 84000000-8403FFFF. Dispositivi: 8 RAM da 32 K (le RAM da 64K statiche NON esistono !!!!)
Di fatto quindi vi sono due banchi da 128K l'uno: il primo realizza la memoria da 84000000 a 8401FFFF e l'altro da 84020000 a 8403FFFF. I chip di memoria da 32 K utilizzano al loro interno (fisicamente) come indirizzi di selezione delle celle i pin A14-A0 che sono però collegati rispettivamente agli indirizzi emessi dal DLX BA16-BA2 (ricordiamo infatti che BA1 e BA0 del DLX NON sono emessi e al loro posto vengono emessi BE3, BE2, BE1 e BE0). Si noti il ruolo dell'indirizzo DLX BA17 che divide i due banchi

Primo banco (decodifica non semplificata)

CSRAM00= (BA31•BA30*•BA29*•BA28*•BA27*•BA26• ….•BA18*•BA17*) •BE0
CSRAM01= (BA31•BA30*•BA29*•BA28*•BA27*•BA26• ….•BA18*•BA17*) •BE1
CSRAM02= (BA31•BA30*•BA29*•BA28*•BA27*•BA26• ….•BA18*•BA17*) •BE2
CSRAM03= (BA31•BA30*•BA29*•BA28*•BA27*•BA26• ….•BA18*•BA17*) •BE3

Secondo banco (decodifica non semplificata)

CSRAM10= (BA31•BA30*•BA29*•BA28*•BA27*•BA26• ….•BA18*•BA17) •BE0
CSRAM11= (BA31•BA30*•BA29*•BA28*•BA27*•BA26• ….•BA18*•BA17) •BE1
CSRAM12= (BA31•BA30*•BA29*•BA28*•BA27*•BA26• ….•BA18*•BA17) •BE2
CSRAM13= (BA31•BA30*•BA29*•BA28*•BA27*•BA26• ….•BA18*•BA17) •BE3

Ovviamente nel caso di decodifica semplificata (memoria logica incompletamente realizzata fisicamente) le funzioni di decodifica vengono ridotte di complessità. Ove questi due banchi fossero gli unici da realizzare i CS dipenderebbero solo da BA17 e da BEi