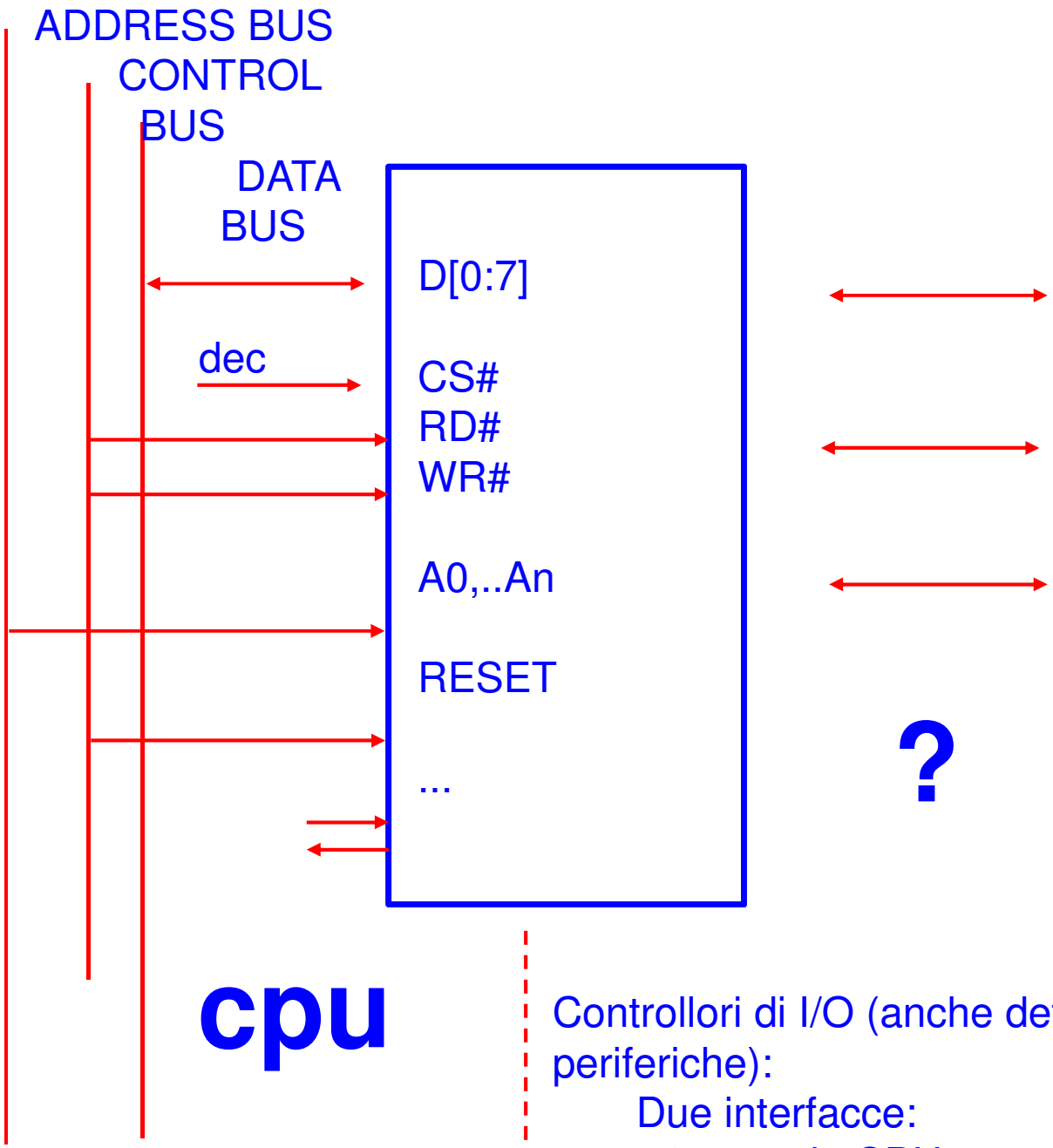
CONTROLLORI DI I/O



Controllori di I/O (anche detti periferiche):

Due interfacce:

- 1 verso la CPU, standard
- 1 verso l'esterno, specifica

Interfaccia con periferiche

Interfaccia con periferiche a 8 bit con un bus dati di 8 bit (esempio: 8088)

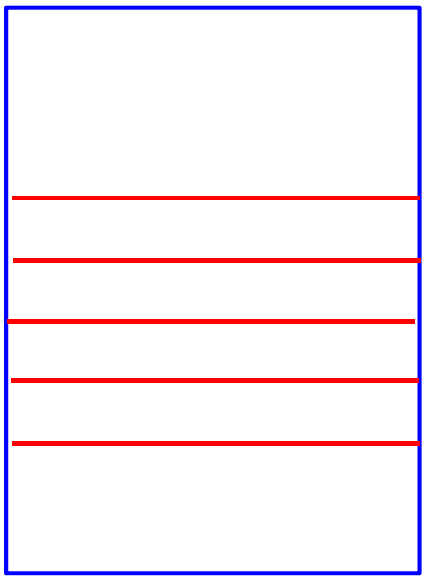
Esempio: periferica con 4 registri di configurazione R0.. R3

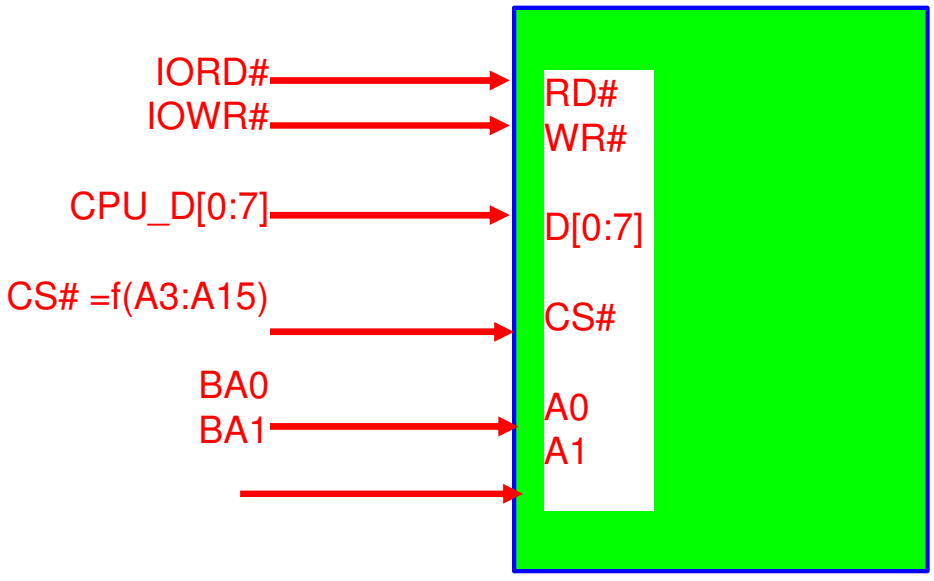
Necessità di distinguere nell'accesso ai 4 registri: servono 2 pin di indirizzo sul controllore, A0 e A1

A0 <-- BA0
A1 <-- BA1

SPAZIO DI INDIRIZZI DI I/0: 16 bit BA0...BA15
I 4 registri hanno indirizzi contigui:
Esempio:

CSR3 = 6533H
CSR2 = 6532H
CSR1 = 6531H
CSR0 = 6530H





Interfaccia su bus a più di 8 bit

- A differenza dell'interfaccia con la memoria, l'interfaccia con i controllori di I/O non ha in genere l'esigenza di bus a elevato parallelismo
- Molti controllori di I/O hanno una porta dati a soli 8 bit
- Quando si interfaccia un controllore di I/O a 8 bit con un bus di dati ad esempio a 16 bit, si devono operare delle scelte:
  - collegare il controllore al solo bus basso
  - collegare il controllore al solo bus alto
  - collegare il controllore alternatamente a entrambi i bus

Interfaccia con periferiche

Interfaccia con periferiche a 8 bit con un bus dati di 16 bit (esempio: 8086)

Scelta semplice: colleghiamo la periferica al solo bus basso D0-D7
! La periferica è raggiungibile solo a indirizzi pari!

A0 <-- BA1
A1 <-- BA2

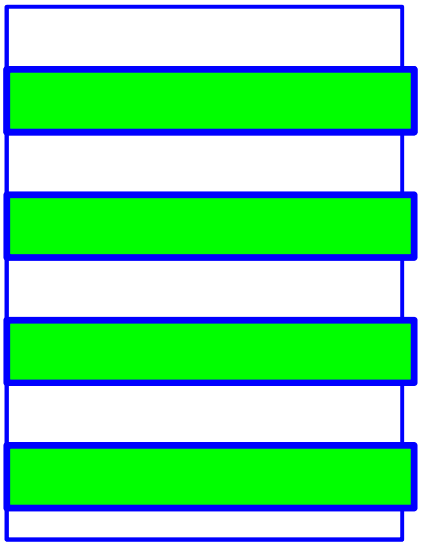
I 4 registri hanno indirizzi pari contigui:
Esempio:

CSR3 = 6536H

CSR2 = 6534H

CSR1 = 6532H

CSR0 = 6530H



Allo stesso modo i 4 registri avrebbero indirizzi dispari contigui se avessimo collegato la perifrerica sol o al bus alto
La periferica si comporta come un banco di memoria o pari o dispari

Con bus a 32 bit:
A0 <-- BA2
A1 <-- BA3

CSR0 = 6530h
CSR1 = 6534h
CSR2 = 6538h
CSR3 = 653Ch
..

Interfaccia con periferiche

Interfaccia con periferiche a 8 bit con un bus dati a 16 bit

!!! durante la scrittura di un byte l'8086 pilota tutti i 16 bit (8 con valore indefinito)

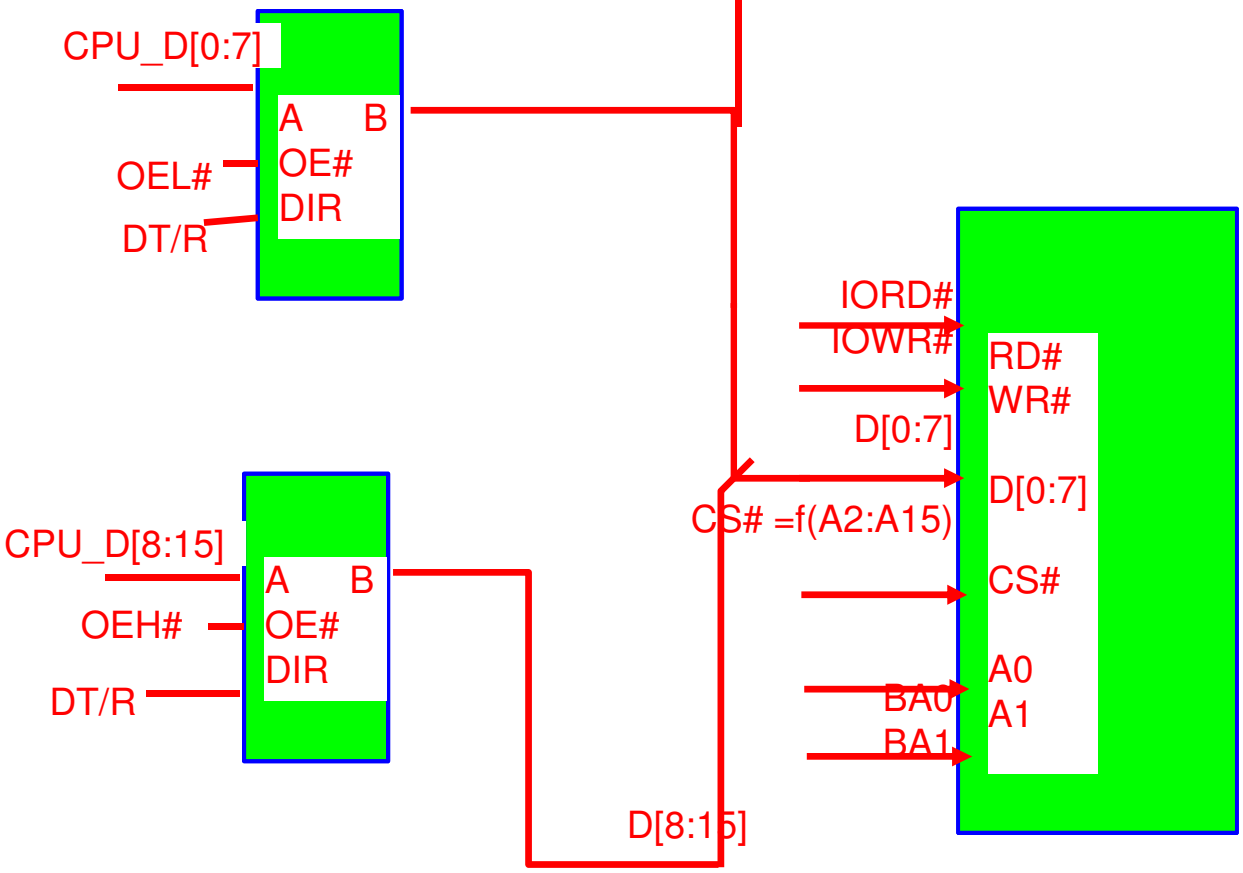
SE NECESSITA' DI INDIRIZZI CONTIGUI ->
CIRCUITI DI PILOTAGGIO CON TRANSCEIVERS

A0 <-- BA0
A1 <-- BA1
CSR0 = 6530H
CSR1 = 6531H
CSR2 = 6532H
CSR3 = 6533H

A0=0 PARI OEH# <-- off
OEL# <-- on
A0=1 DISPARI
OEH# <-- on
OEL# <-- off

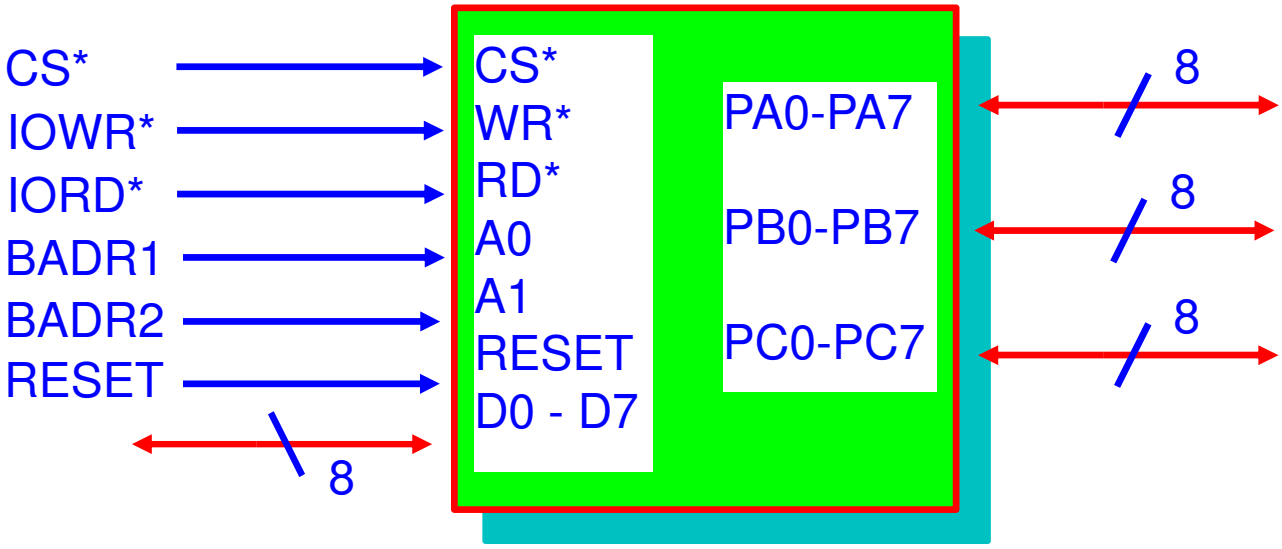
OEL#=A0

OEH#=$\overline{A0}$



CONTROLLORE PORTA PARALLELA 82(C)55

Gestione differenziata e programmabile di tre porte parallele bidirezionali (input o output)

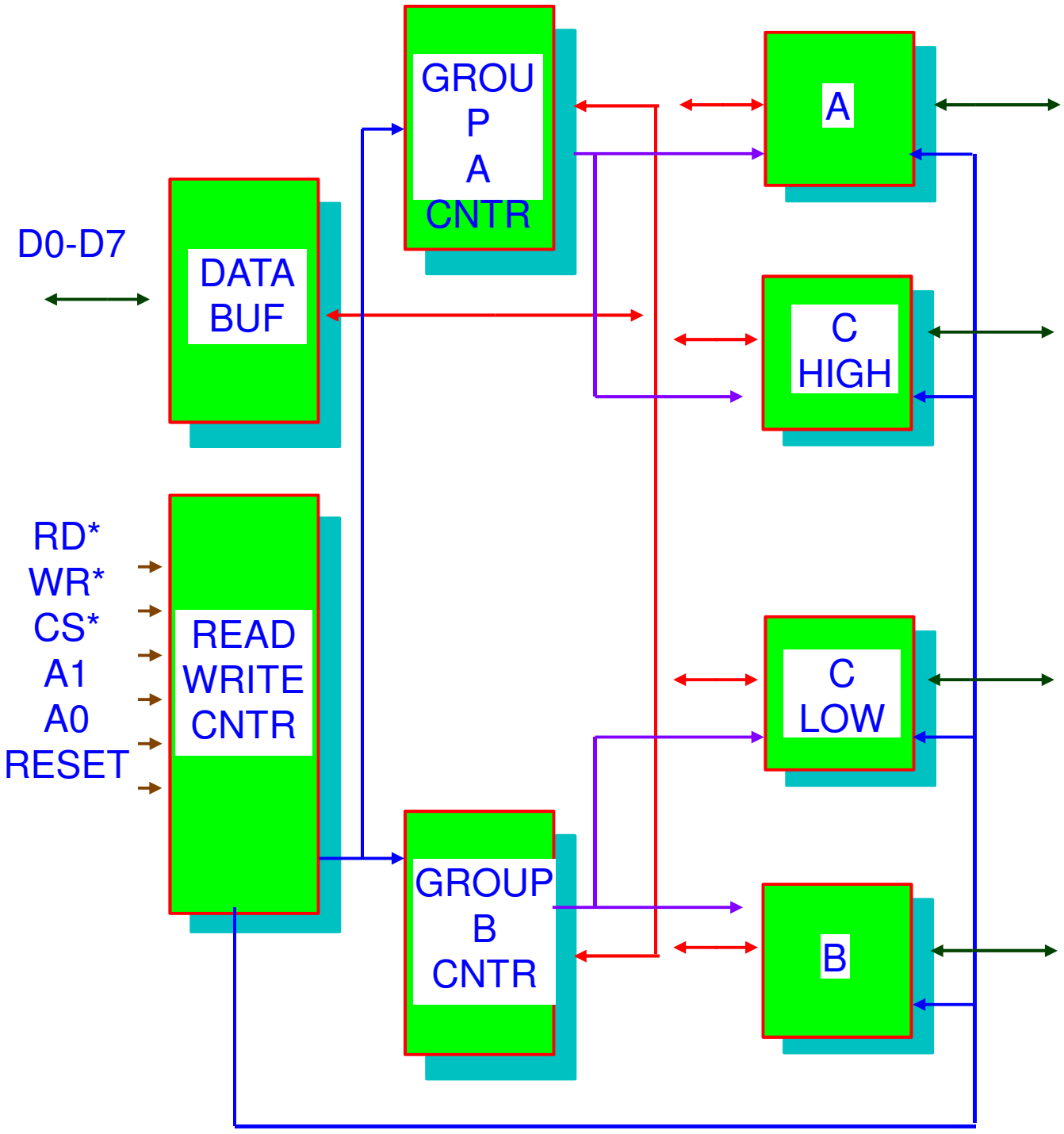


8255

Il dispositivo occupa 4 locazioni di indirizzo

Nel caso 8086 occupa 8 locazioni (minimo) se mappato solo a indirizzi pari o dispari

8255
STRUTTURA INTERNA



82C55 PROGRAMMAZIONE

Programmazione della **parola di controllo**:
- modo e direzione per ogni porta

Operazioni ammesse:
- Lettura / scrittura da ogni porta
- Scrittura di un singolo bit per volta (porta C)

A1	A0	RD*	WR*	CS*	
x	x	x	x	1	TRI-STATE
x	x	1	1	0	TRI-STATE
0	0	0	1	0	Read A
0	0	1	0	0	Write A
0	1	0	1	0	Read B
0	1	1	0	0	Write B
1	0	0	1	0	Read C
1	0	1	0	0	Write C
1	1	1	0	0	Write Control

N.B.: la parola di controllo non puo' essere riletta

Modi di funzionamento

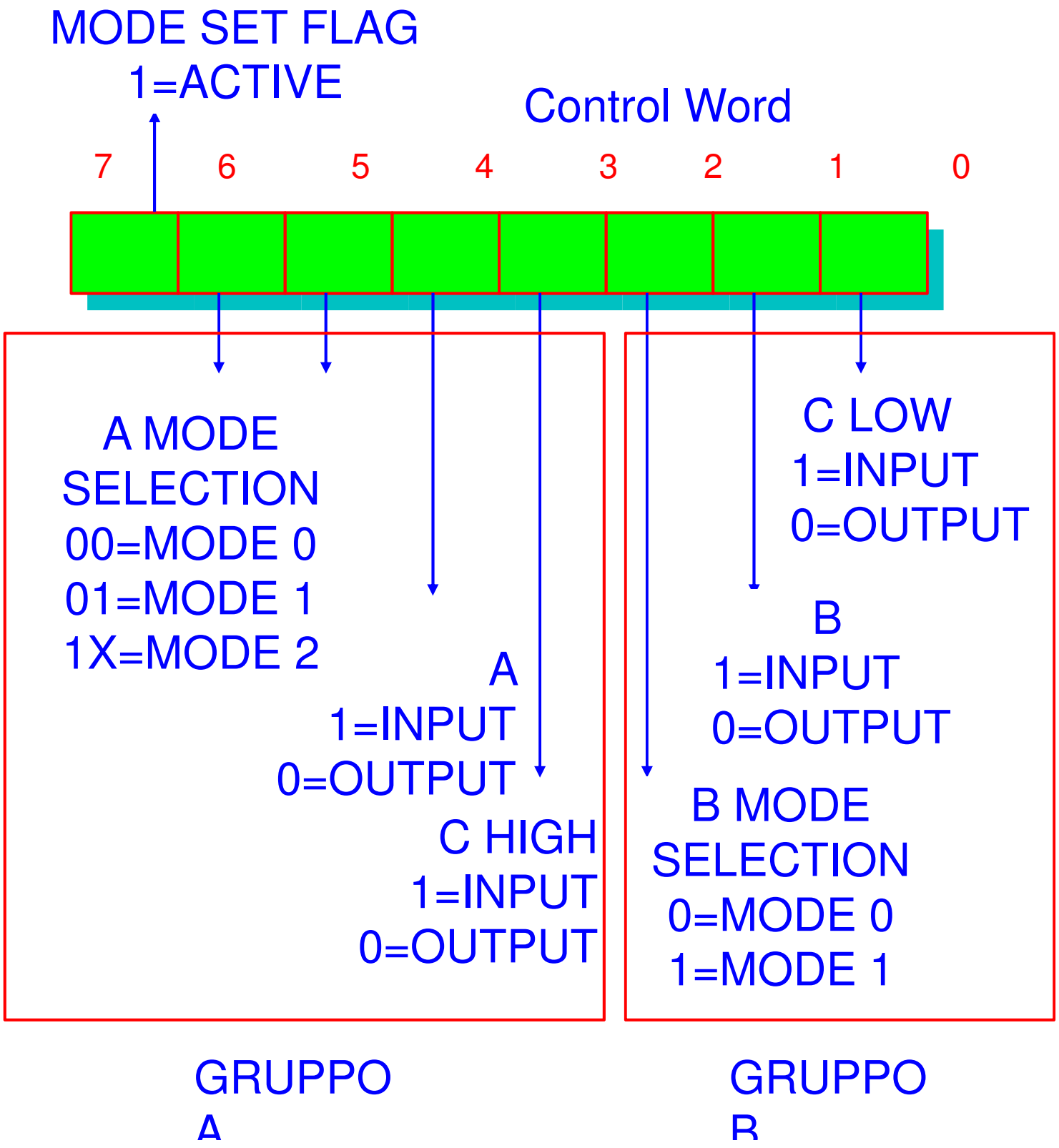
1) Mode 0: Basic I/O
LA CPU MASTER DECIDE SENZA SINCRONIZZAZIONE
I TEMPI DI LETTURA E SCRITTURA SULLA PORTA

2) Mode 1: Strobed I/O
L'INTERFACCIA CON L'ESTERNO E' SINCRONIZZATA DA UN PROTOCOLLO AD HANDSHAKE

in questo caslo la CPU deve
1) ESEGUIRE PRIMA UN CONTROLLO SOFTWARE DELLO STATO DI REGISTRI INTERNI,
oppure:
2) LAVORARE AD INTERRUPT

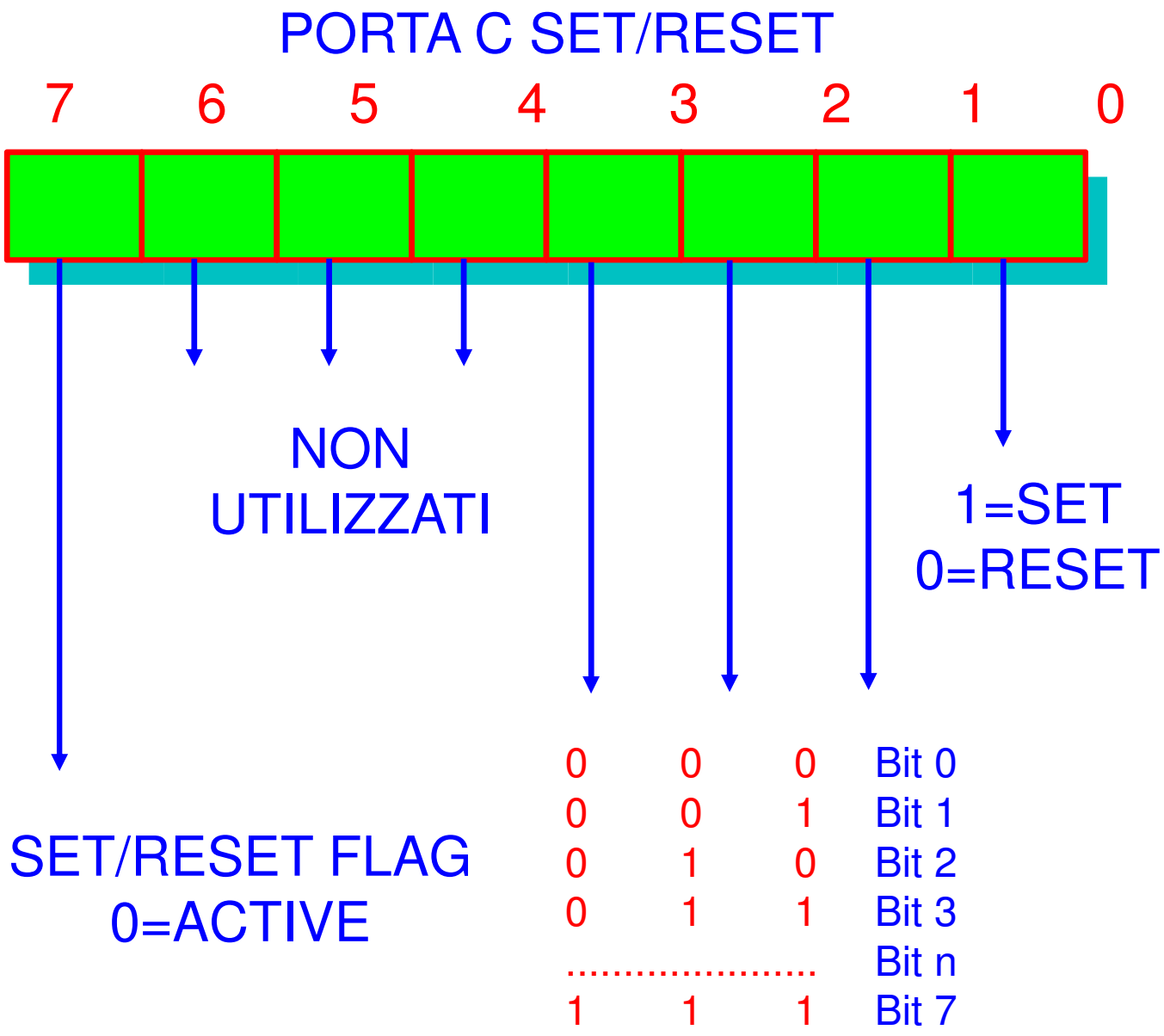
3) Mode 2: Strobed I/O bidirezionale
con doppio handshake in trasmissione e ricezione

In Modo 0 possono essere programmate le porte A, B, C.
In Modo 1 possono essere programmate le porte A, B.
In Modo 2 può essere programmata solo la porta A.



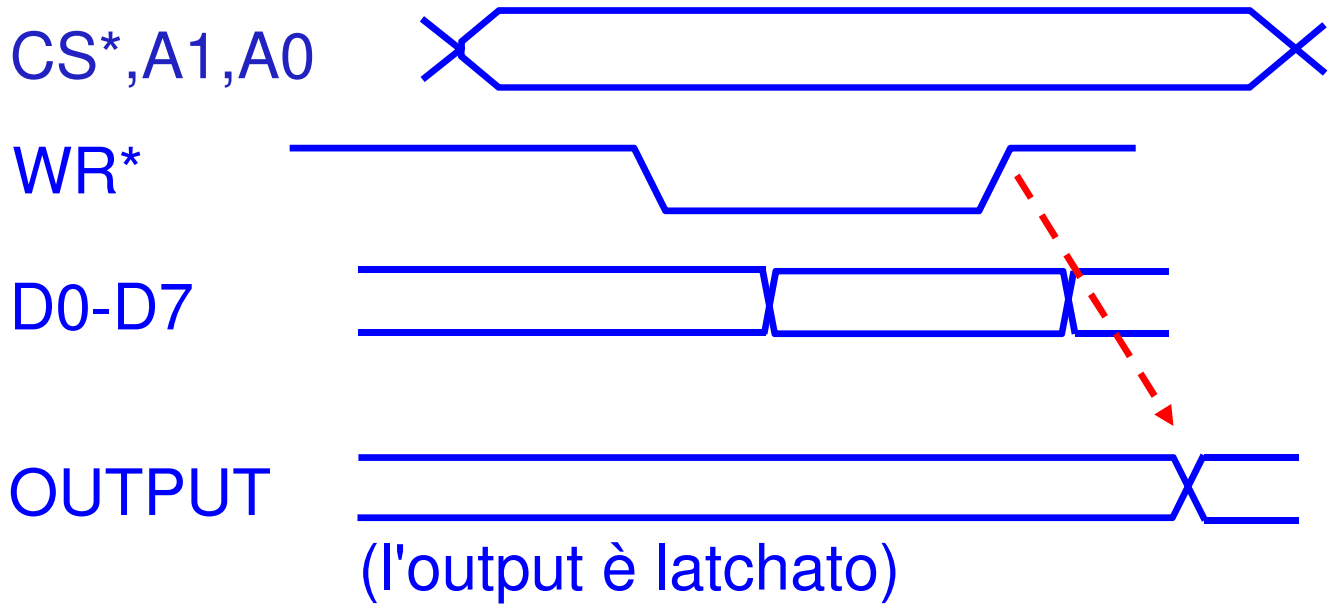
- ESEMPIO: PORTA C LOW = INPUT
  PORTA C HIGH = OUTPUT
  PORTA B = OUTPUT MODO 1
  PORTA A = INPUT MODO O
  CONTROL WORD = 10010101=95H
- N.B.: la programmazione della porta C si riferisce ai pin non usati per il controllo di A o B

8255 PROGRAMMAZIONE



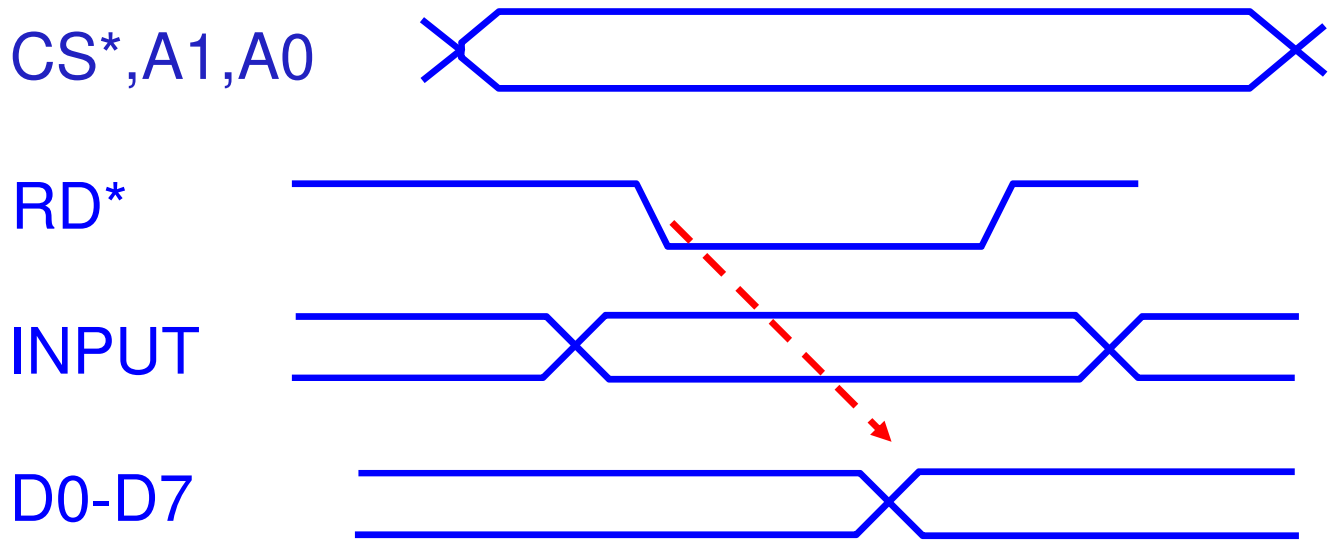
8255 INTERFACCIA CON LA CPU

MODO 0 -SCRITTURA



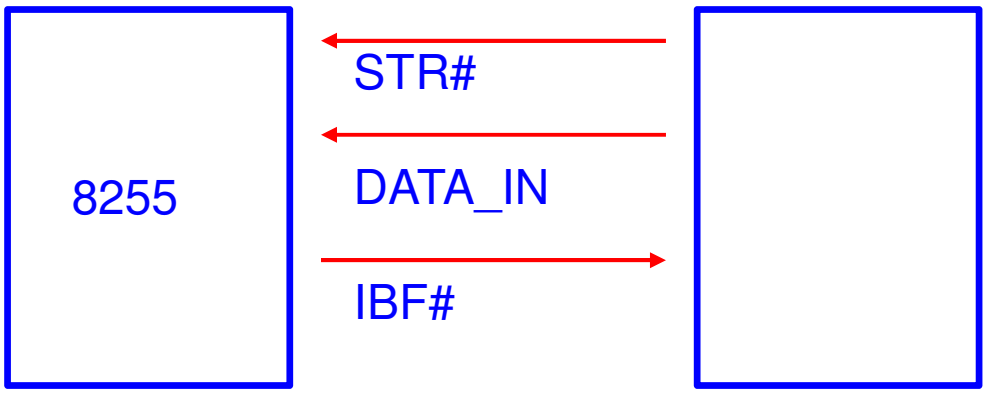
(l'output è latchato)

MODO 0 -LETTURA



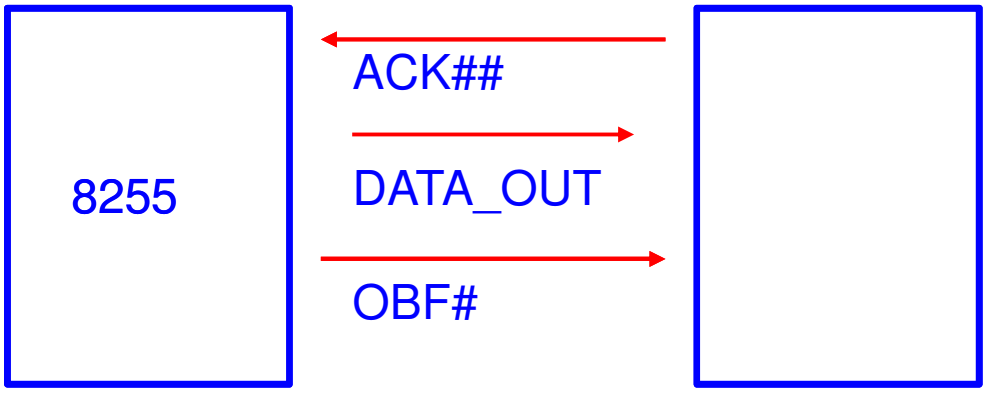
8255 MODO 1

input



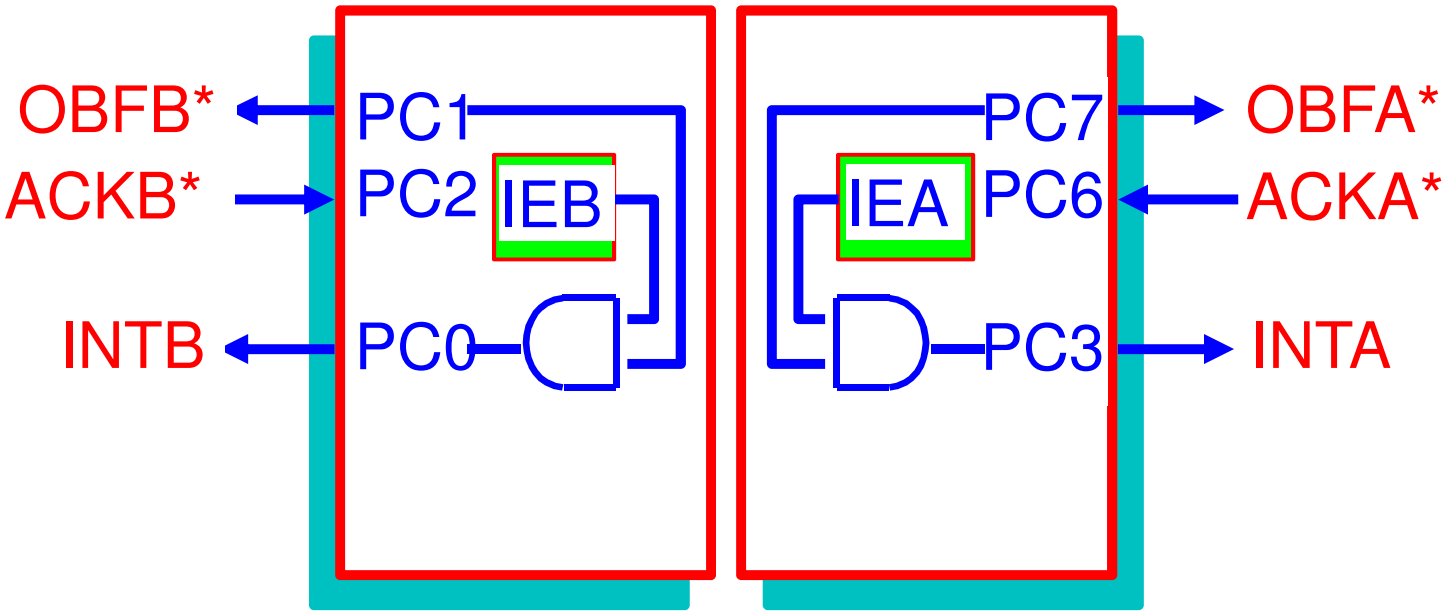
Handshake input:la CPU comunica al dispositivo esterno (sulla destra in figura) quando è pronta a ricevere un carattere; il dispositivo lo trasmette quando è disponibile e segnala con strobe

output



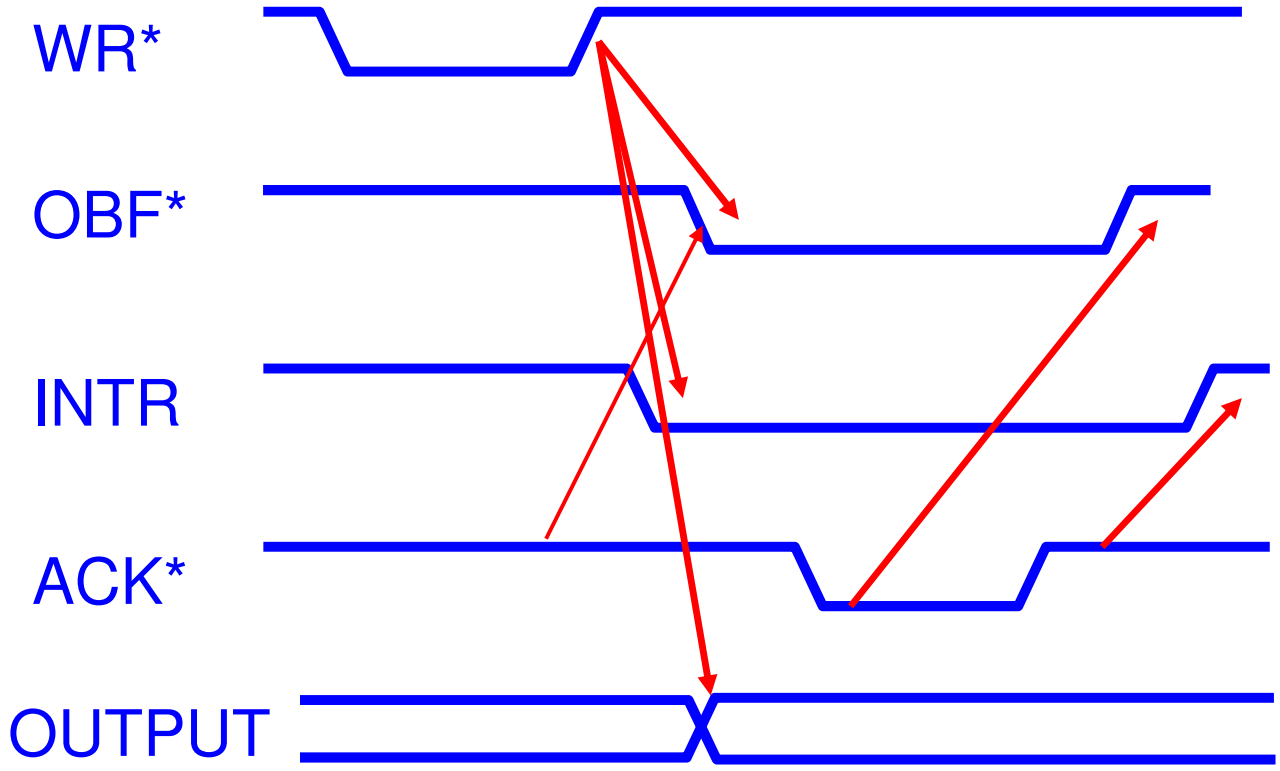
Handshake output: la CPU comunica al dispositivo esterno (sulla destra in figura) quando è pronta a trasmettere un carattere; il dispositivo lo riceve quando è disponibile e segnala con ack

MODO 1 - SCRITTURA CAMPIONATA

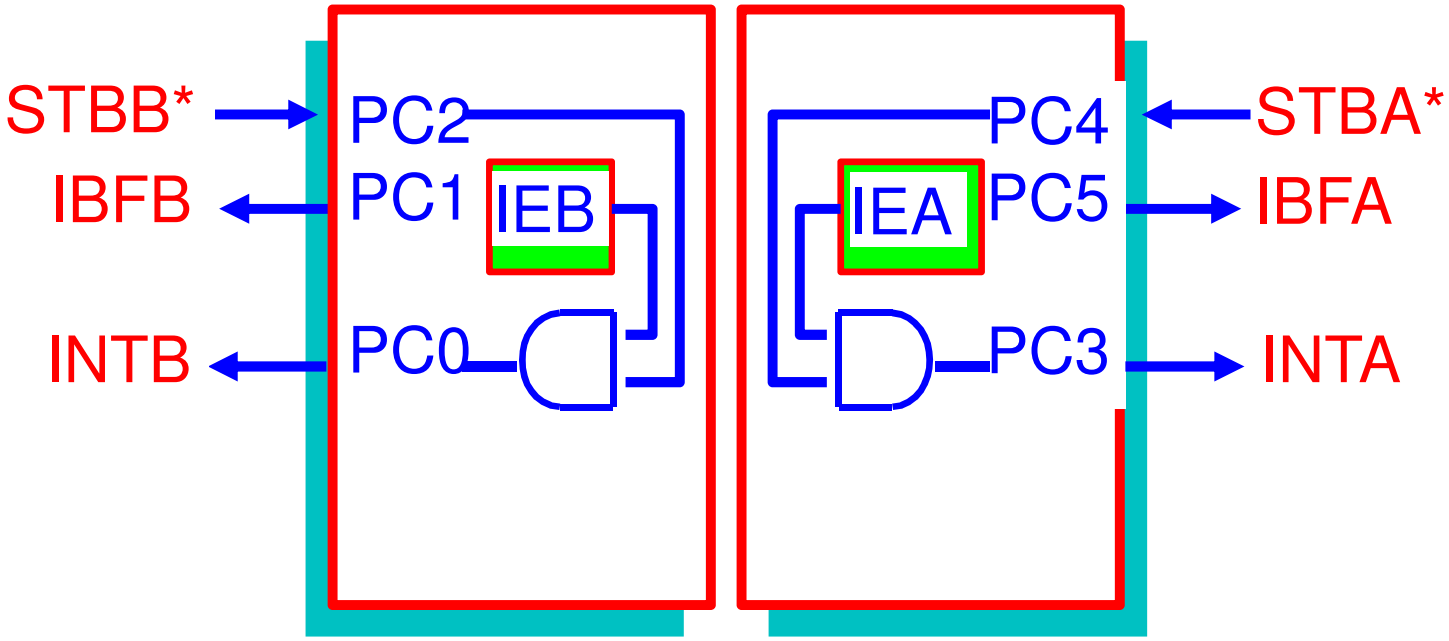


- IEA = FF 6 del registro C
  IEB = FF 2 del registro C

- Alcuni pin della porta C requisiti: in questo caso il pin esterno corrispondente è disconnesso dal flip flop interno

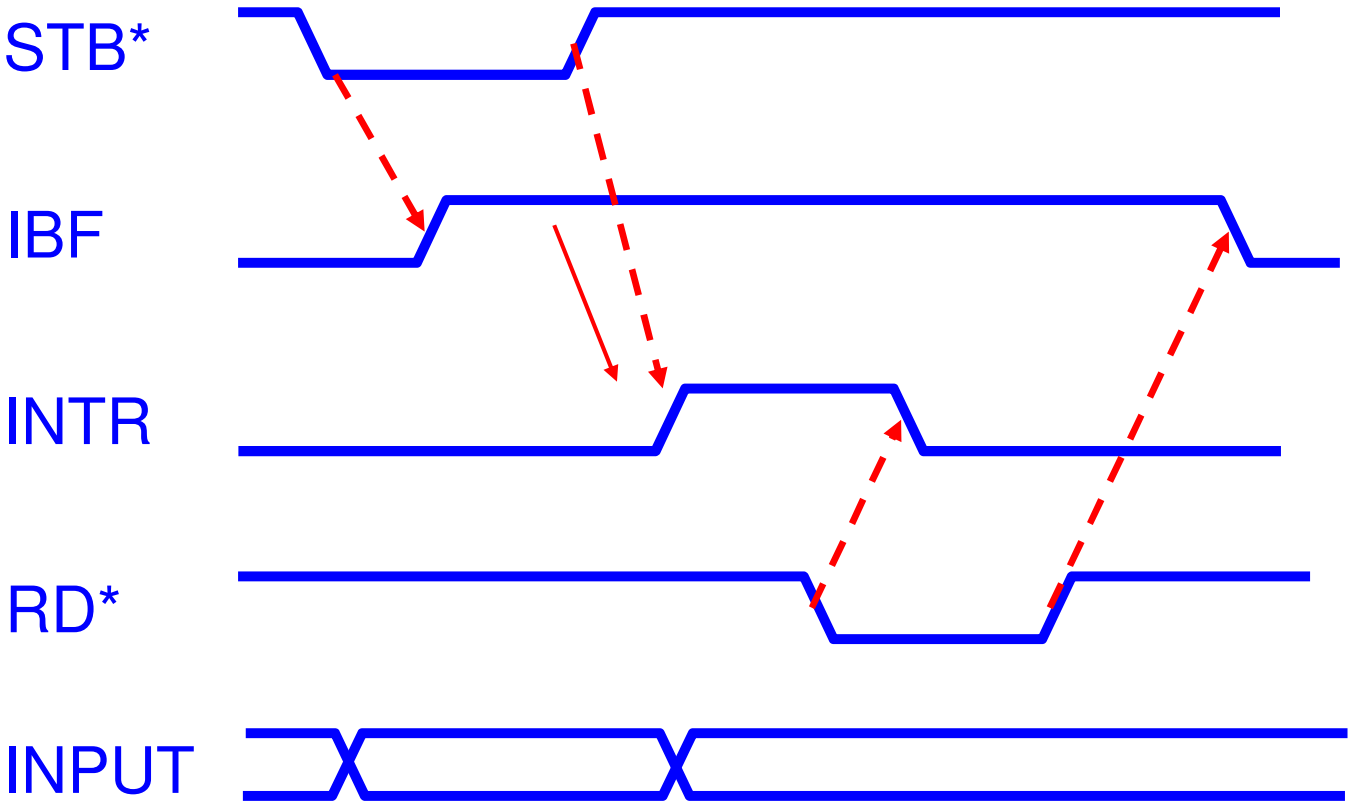


MODO 1 - LETTURA CAMPIONATA

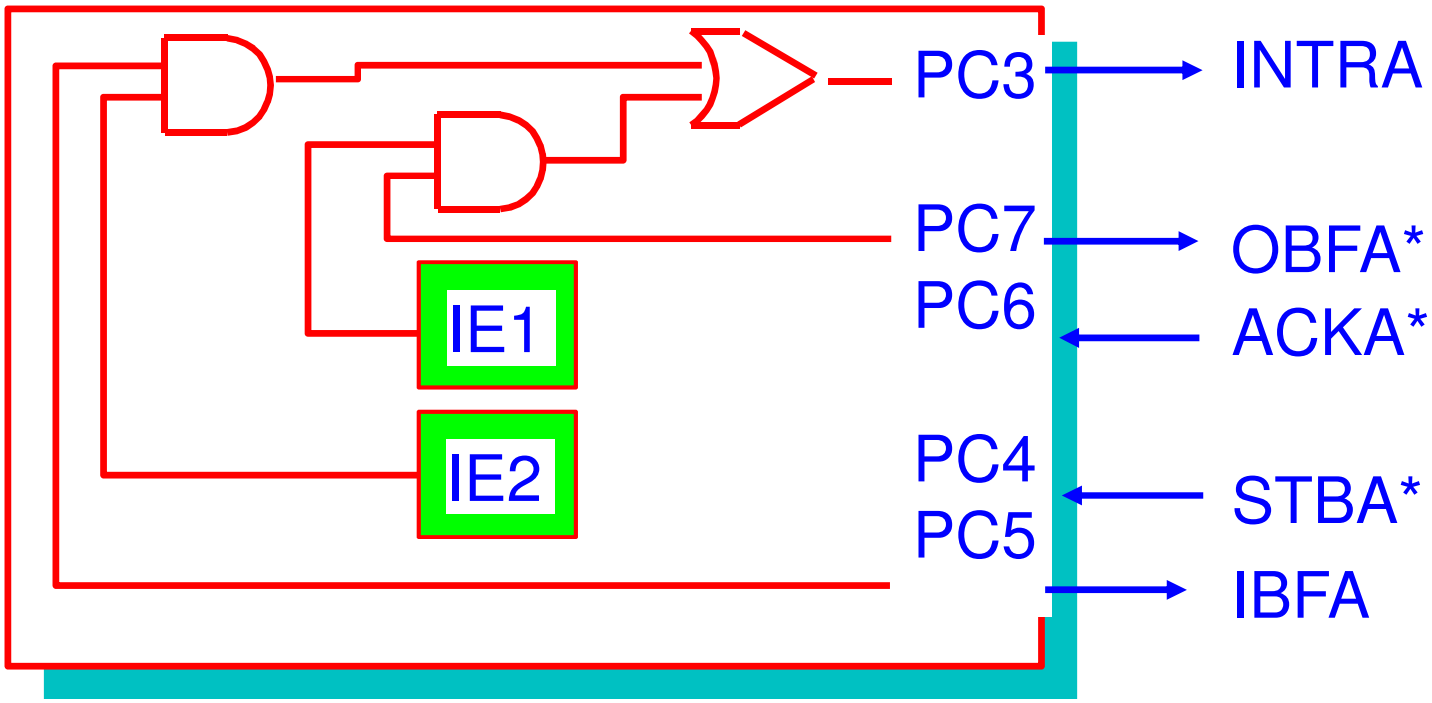


- IEA = FF 4 del registro C
  IEB = FF 2 del registro C

- Alcuni pin della porta C requisiti



MODO 2 - TRASFERIMENTO BIDIREZIONALE CAMPIONATO

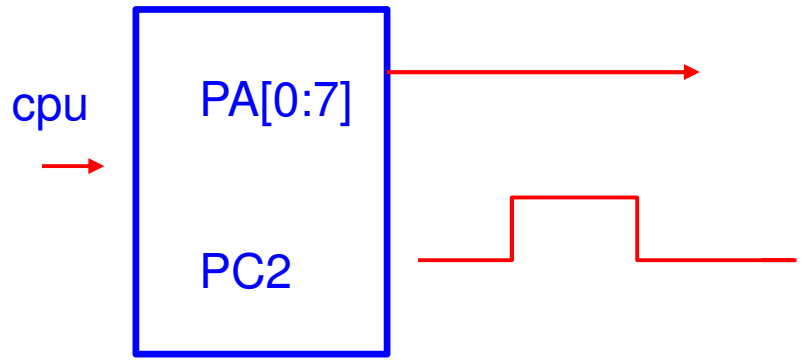


- La sincronizzazione è effettuata dal dispositivo esterno che pilota ACKA* e STBA*
- I dati sono depositati in buffer temporanei
- IE1 = PC6 - IE2 = PC4
- La temporizzazione è la combinazione dei due diagrammi precedenti

- Rileggendo la porta C quando alcuni dei suoi bit sono requisiti si legge lo stato del relativo flip flop

Programmazione

USO DEL PIN 2 DELLA PORTA C
PER GENERARE UN IMPULSO DI SCRITTURA
DI UN DATO CHE ESCE DALLA PORTA A



Esempio di programma in DJGPP, con definizioni in C
e istruzioni in assembler

```
#include <stdio.h>

unsigned char *PortaA_Address = (unsigned char *) 0x378;
unsigned char PortaA_Value = 100;
unsigned char *PortaC_Address = (unsigned char *) 0x37C;
unsigned char PortaC_P2ON  = 0x4; // 00000100b
unsigned char PortaC_P2OFF = 0x0 // 00000000b;
unsigned char *Control_Address = (unsigned char *) 0x37E;
unsigned char Control_Word = 0x80; // ( output, Modo 0)
```

esempio

```
main (void)
{
asm ("
mov Control_Address, %dx
mov Control_Word, %al
out %al, %dx

mov PortaA_Address, %dx
mov PortaA_Value, %al
out %al, %dx

mov PortaC_Address, %dx
mov PortaC_P2ON, %al
out %al, %dx

mov PortaC_Address, %dx
mov PortaC_P2OFF, %al
out %al, %dx
");
return 0;}
```

Qual è la durata dell'impulso su P2? Dipende dalla velocità del processore?