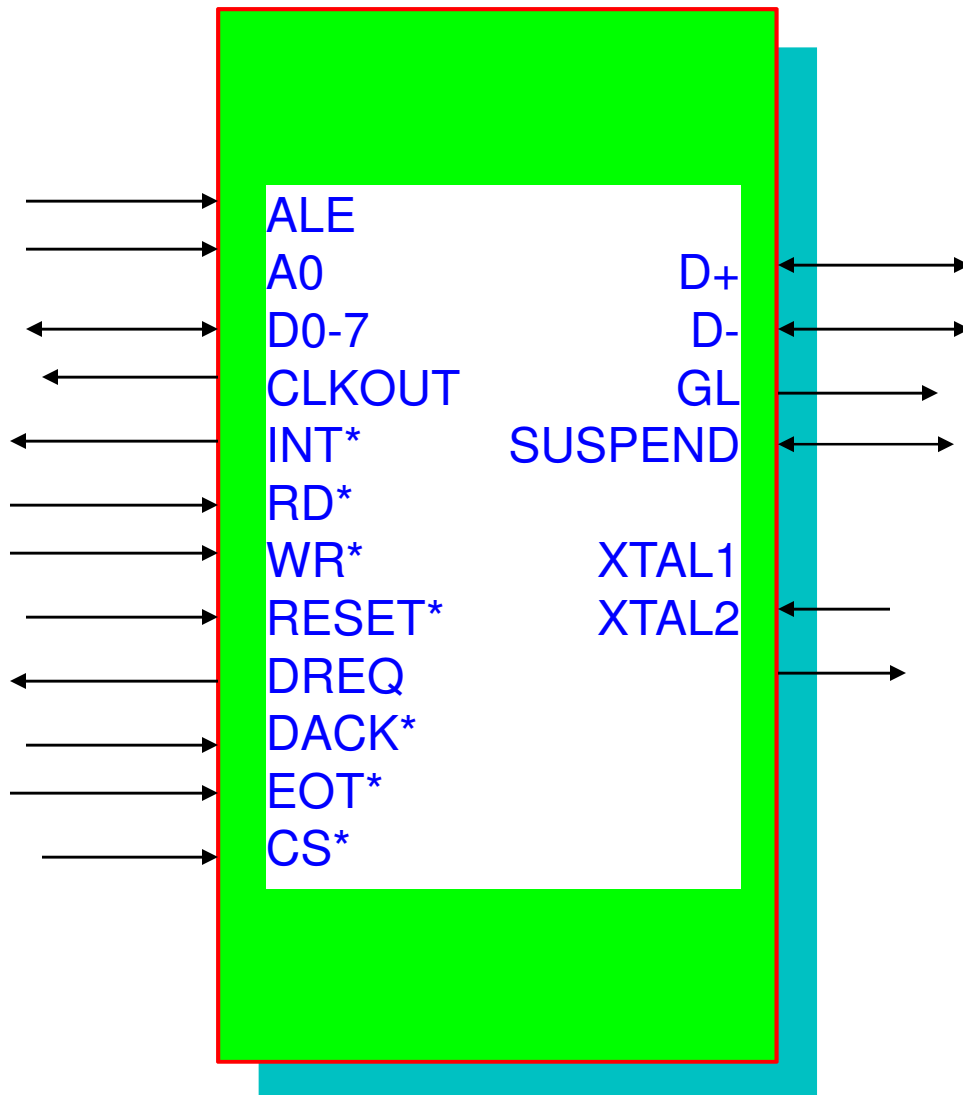
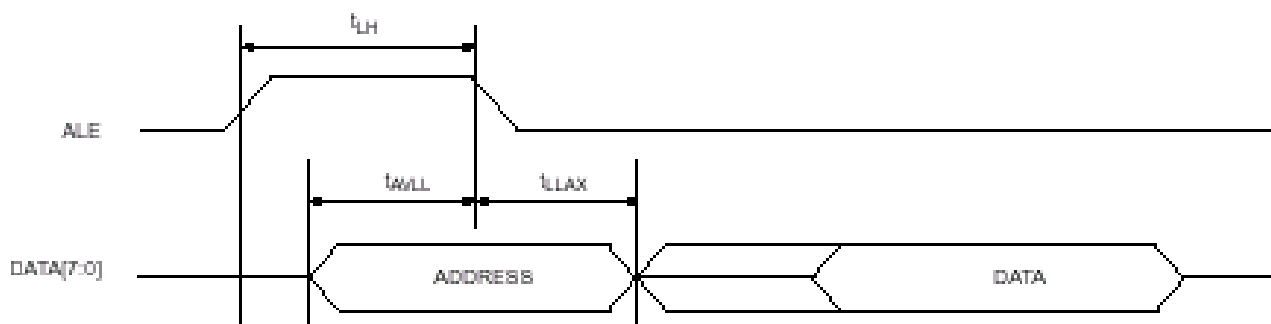


# Il controllore USB D12 (Philips PDIUSB12)



# Pin-out

- D-0-7 (I/O): dati
- A0 (I): 1 comandi  
0 dati
- ALE (I): il D12 può essere interfacciato direttamente a un bus multiplexato, in cui sono trasmessi indirizzi (1 bit) e dati in successione: ALE indica indirizzi validi. (è come se avesse un 373 interno per la registrazione degli ADDRESS)



Quindi se ho:

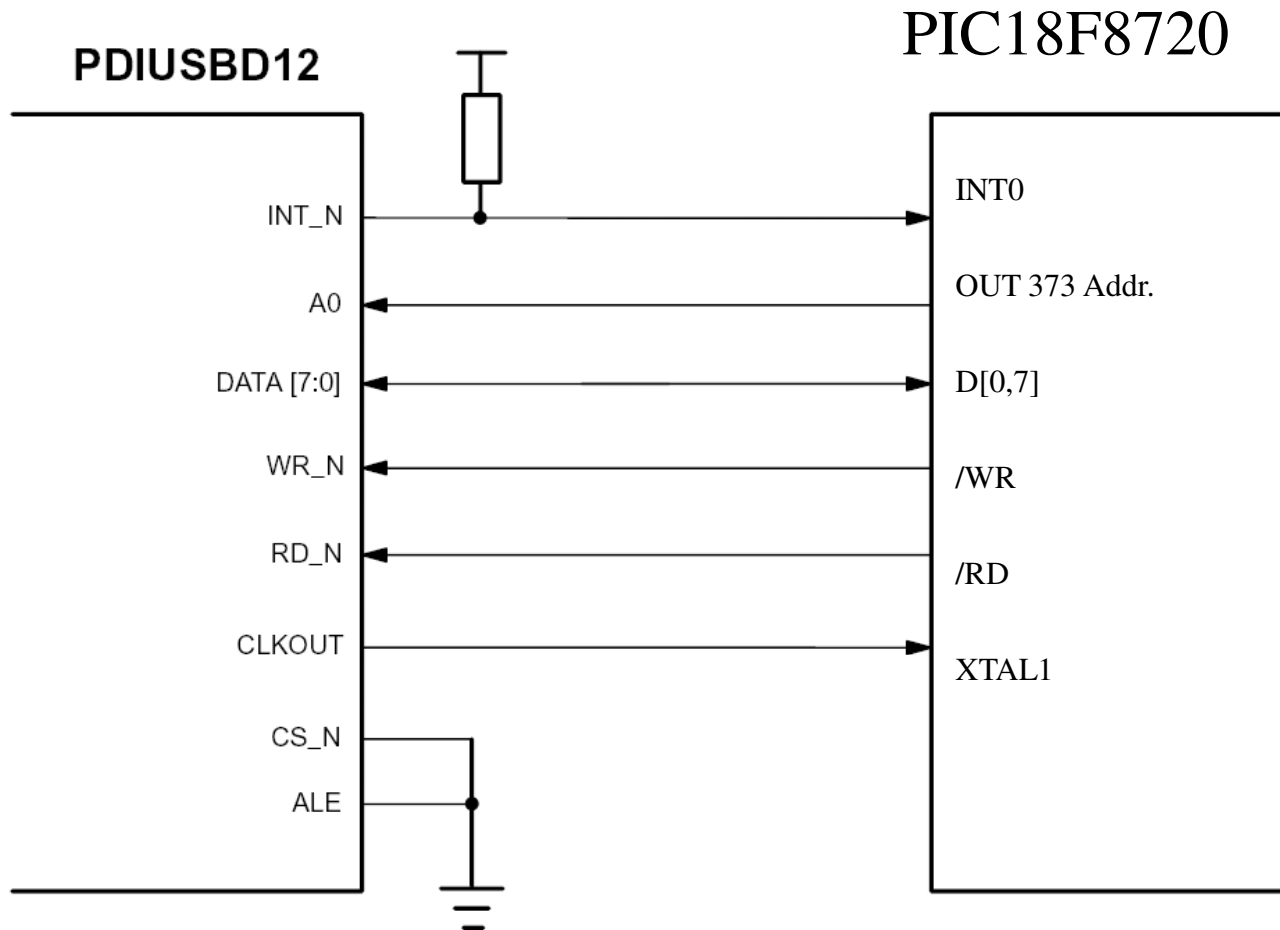
Bus multiplexato → A0 don't care, pull-up alto.

Bus non multiplexato → ALE a GND.

ALE di tipo "a livello" se basso, porta sempre aperta

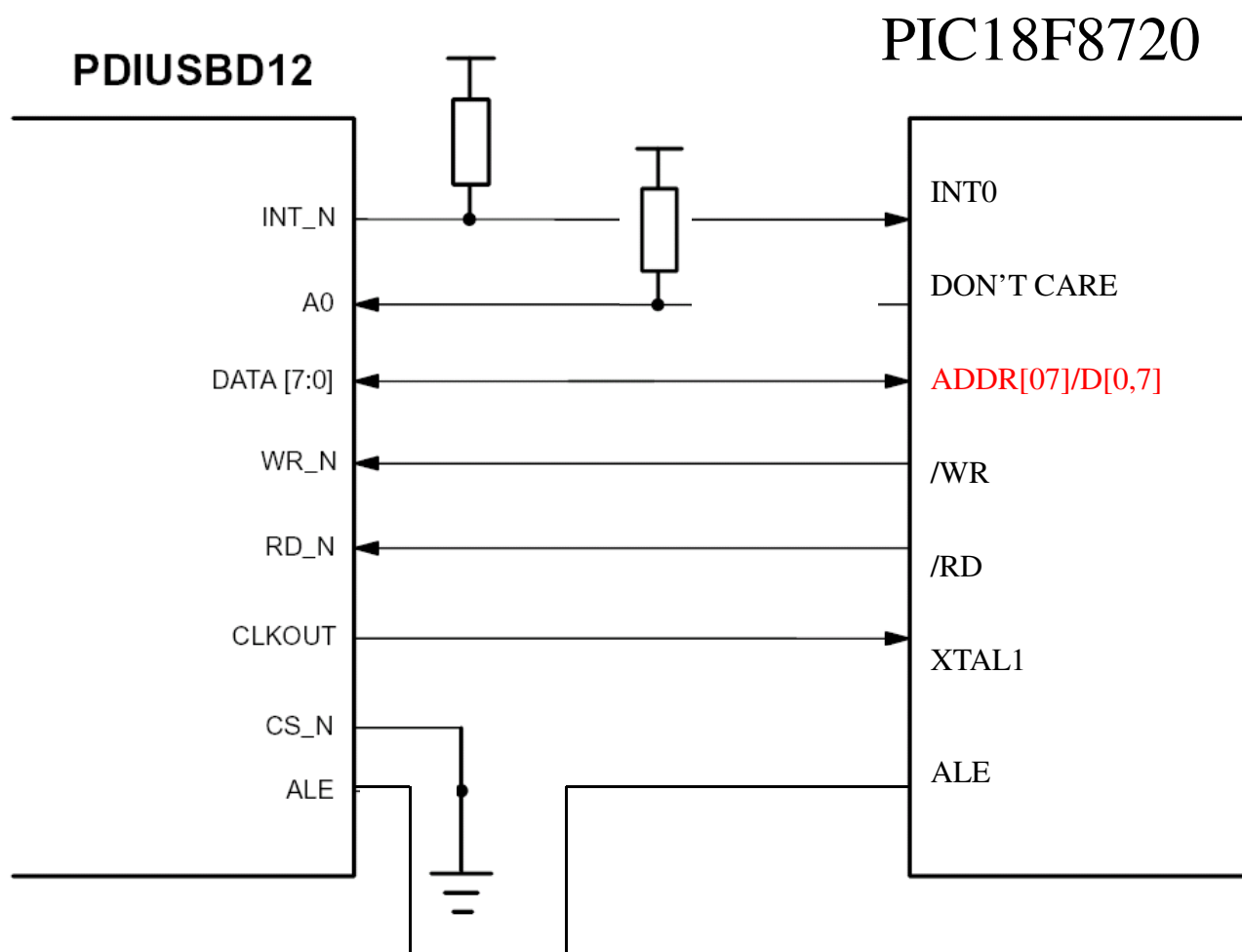
Bus multiplexato, ALE su fronte di discesa permette il campionamento del corretto valore di A0 (comandi/dati)

# Interfaccia con un microcontrollore Con bus non multiplexato



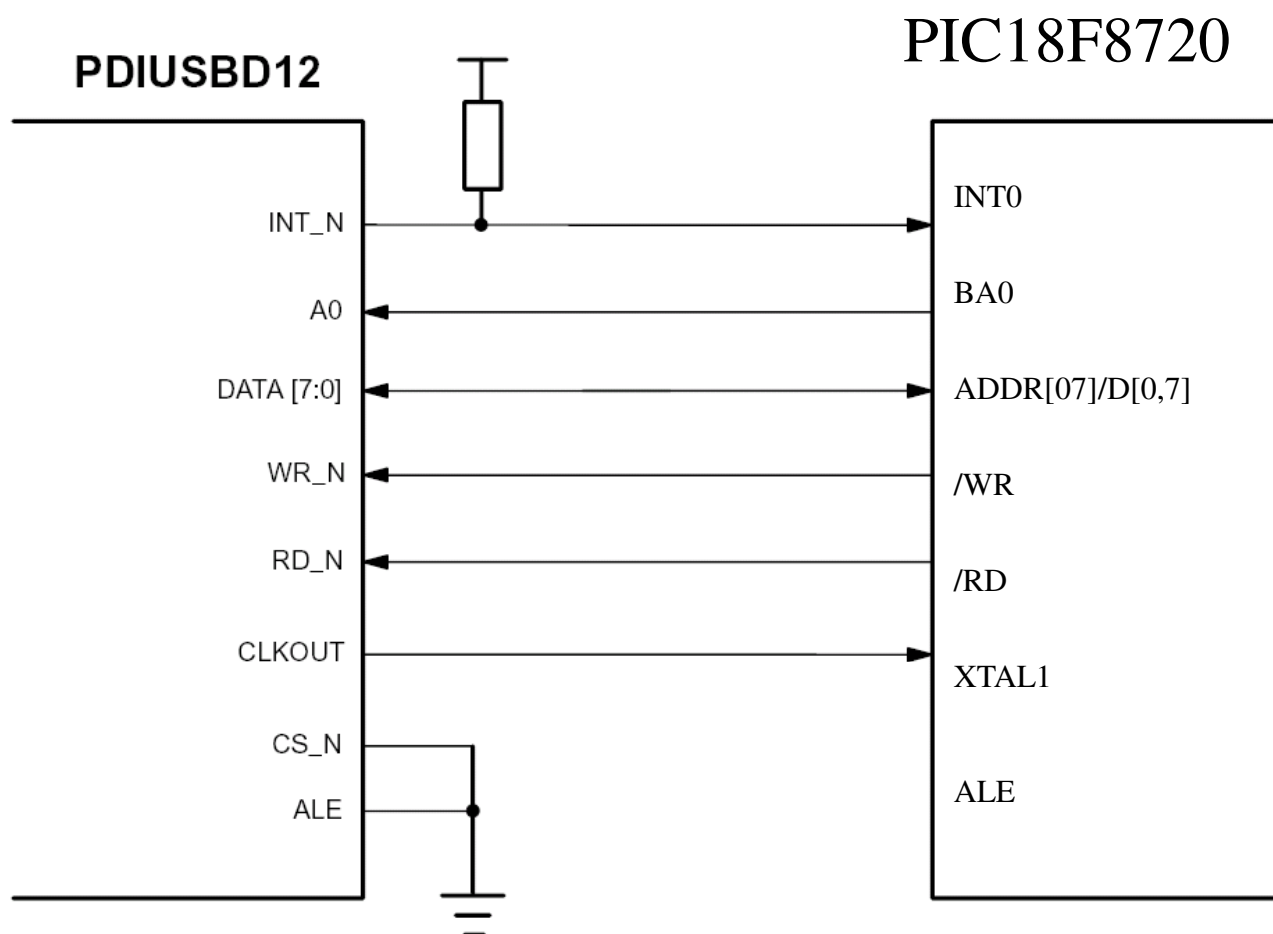
- RD\*, WR\* (I): comandi read e write
- CS (I): Chip select
- DREQ (O), DACK\* (I): richiesta e ack di DMA
- EOT\* (I): End of DMA Transfer; valido se asserito con DACK\* attivo e RD o WR attivi (è anche VBUS, l'eventuale alimentazione USB)

# Interfaccia con un microcontrollore Con bus multiplexato



- ALE è usata per campionare l'ADDR. Che è su linee multiplexate con i dati, in particolare A0 del D12 sarà la linea AD[0] del micro, se mappiamo il D12 solo sul data bus PARI (basso).

# Interfaccia con un microcontrollore Con bus multiplexato (2)



- ALE è inutile perché utilizziamo BA0 che è una linea non fisicamente multiplexata con il bus dati. A0 del D12 sarà la linea BA0] del micro, se mappiamo il D12 su tutti gli indirizzi (pari e dispari). Quindi pur essendo il bus multiplexato, non abbiamo bisogno di ALE

## Pin-out (2)

- INT\* (O): richiesta di interrupt
- XTAL1 (I), XTAL2 (O): per collegare un quarzo a 6 MHz.  
All'interno del D12 c'è un moltiplicatore di frequenza che porta il clock a 48 MHz
- CLKOUT (O): frequenza di uscita programmabile per il clock a un microprocessore.

$$\text{CLKOUT} = 48 \text{ MHz} / (\text{N} + 1), \text{ con } \text{N} = 0 \div 11$$

- GL (O): indicatore di stato a LED (Good Link)
- D+, D-: linee dato differenziali del bus USB
- SUSPEND (I/O): quando SUSPEND si porta a 1, dopo 1 ms CLKOUT va a 30 KHz (LazyClock)

# Modalità di funzionamento

- All'interno del D12 sono presenti 3 endpoint diversi: 0, 1, e 2 (endpoint principale). Ciascun endpoint può specializzarsi in una o due direzioni (input e output); l'**endpoint index** tiene conto delle direzioni.

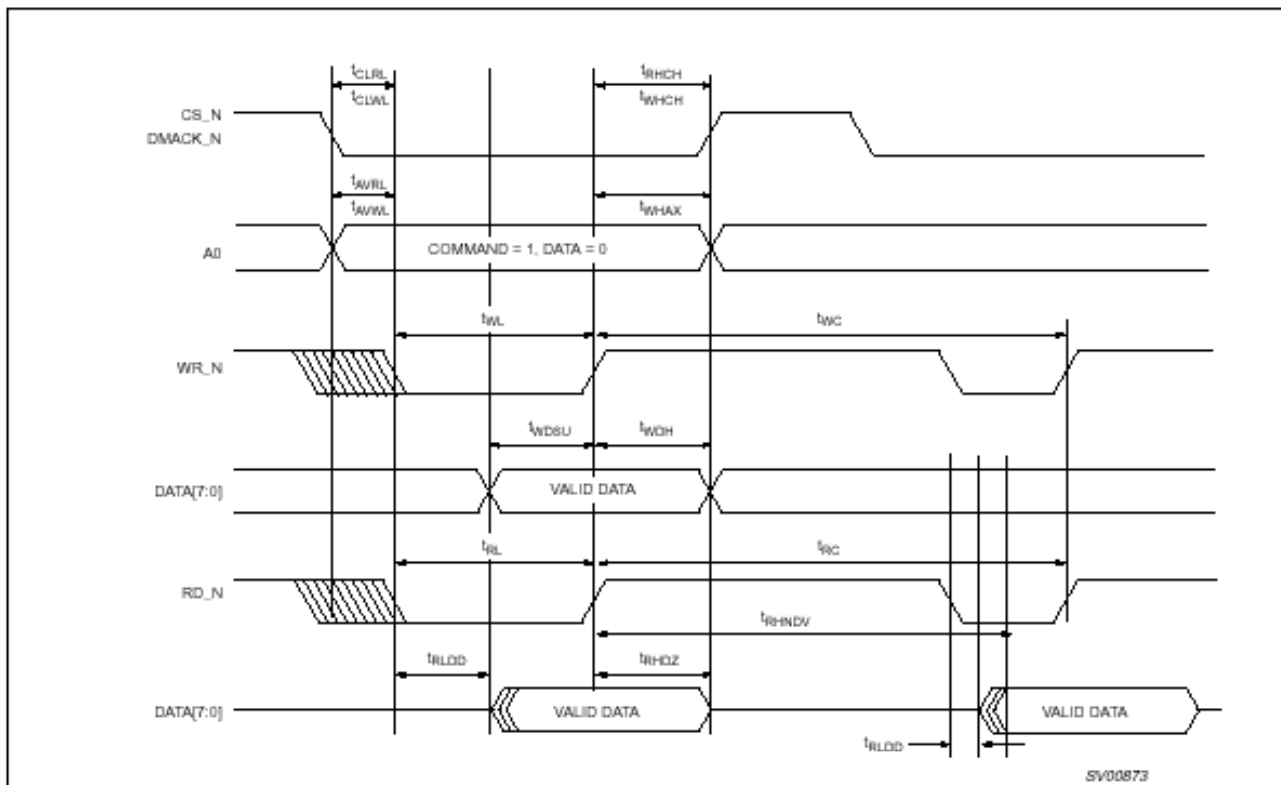
**endpoint 0**, **index** 0 e 1 (out e in): trasferimenti di controllo;

**endpoint 1**, **index** 2 e 3 (out e in): trasferimenti non isocroni;

**endpoint 2**, **index** 4 e 5 (out e in, se presenti): trasferimenti isocroni o non isocroni, a seconda del modo.

- 4 modi di funzionamento:
  - Modo 0: non isocrono input/output
  - Modo 1: isocrono output
  - Modo 2: isocrono input
  - Modo 3: isocrono input/output
- massima taglia del pacchetto:
  - endpoint 0 e 1: 16 byte
  - endpoint 2 in input/output: 64 byte
  - endpoint 2 in solo input o solo output: 128 byte

# Letture e scrittura



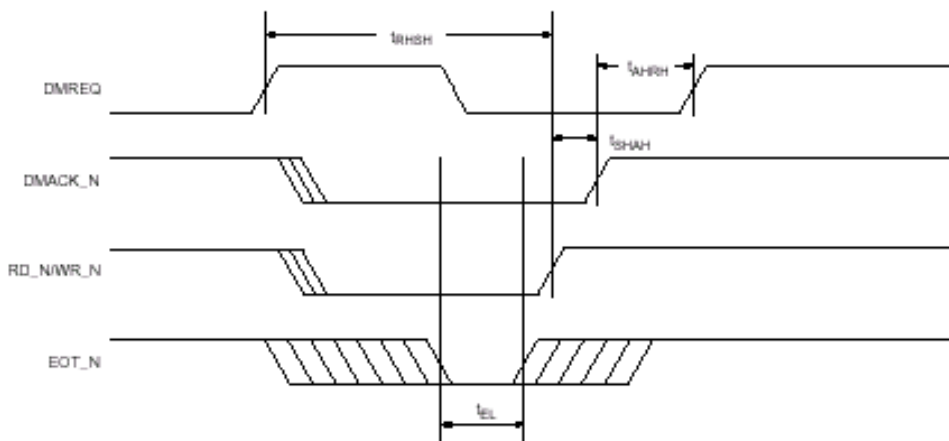
- TWL, TRL: durata WR e RD, min 20 ns
- TCLRL, TCWL: non significativi
- TRHCH, TWHCH: tempo tra fine di RD o WR e fine di CS, min 5 ns
- TWDSU, TWDH: setup e hold sui dati in scrittura rispetto alla fine di WR, 30 e 10 ns min
- TRLDD: tempo per dati validi in uscita dall'inizio di RD, max 20 ns
- TWC, TRC: tempo minimo di ciclo! per scrittura e lettura: 500ns



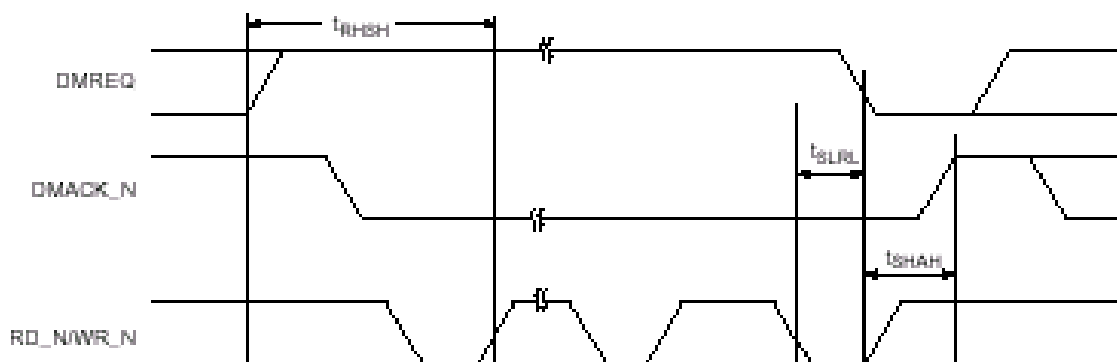
# DMA

- Il D12 prevede l'utilizzo con un DMA controller per non caricare la CPU dei trasferimenti verso la memoria.
- La CPU riceve dall'host USB le informazioni relative alla programmazione del DMA controller, tipicamente demand transfer mode e i valori di byte count e address count.
- 2 modi di DMA:
  - *single cycle*: ad ogni DACK, DREQ è disattivato
  - *burst mode*: DREQ è disattivato da EOT alla fine di tutti i trasferimenti in burst programmati. EOT genera una richiesta di interrupt su INT.
- Il trasferimento dal D12 alla memoria (DMA read nei data sheet) inizia solo quando il buffer di ricezione ha ricevuto un pacchetto (DREQ si attiva). Mentre avviene il trasferimento in DMA, un secondo buffer di ricezione è attivo. Se il D12 riceve dall'USB host meno dei byte programmati, interrompe comunque il DMREQ.
- In modalità isocrona, un pacchetto di taglia massima (128 byte) va ricevuto in 1 ms.

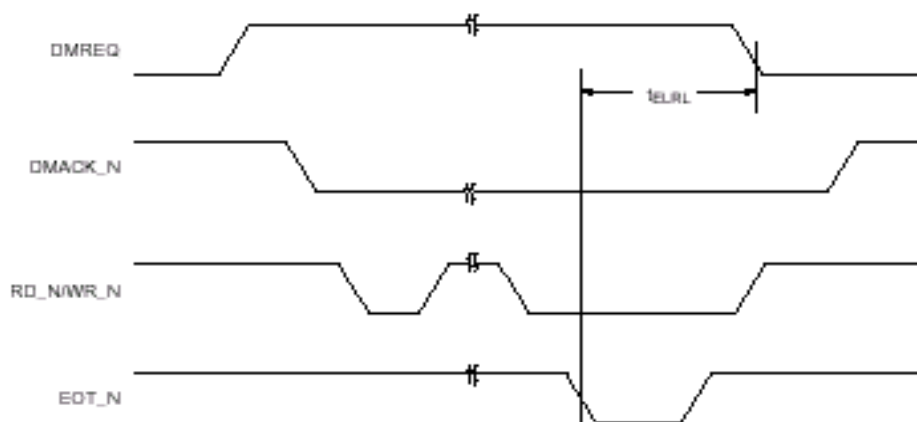
# Temporizzazioni in DMA



- Single cycle DMA



- Burst mode DMA



- EOT in Burst mode DMA

## Temporizzazioni in DMA (2)

- TEL: durata impulso di EOT min 10 ns contati con DACK e RD o WR attivi (bassi)
- TRHSH: tempo per cui si può disattivare RD/WR da quando si è attivato DREQ, min 120 ns
- Soltanto per il burst read, deve essere rispettato TRHNDV: tempo dalla disattivazione del RD precedente all'uscita dei dati correnti, min 420 ns (vedi diagramma del ciclo di bus). Questo perché un burst read è costituito da una sequenza di RD/dati in rapida successione.
- TELRL: si abbassa DREQ dall'inizio di EOT, max 40 ns
- TSLRL: si abbassa DREQ dall'inizio dell'ultimo RD, max 40 ns
- TSHAH: tempo minimo per poter alzare DACK dalla fine dell'ultimo RD

# Programmazione

- Tre tipi di comandi: inizializzazione, flusso dei dati, generali.
- Ogni comando consiste nell'invio di un byte alla porta comandi (A0 = 1) con un valore che specifica il comando, e della scrittura o lettura successiva di un numero prefissato di byte dalla porta dati (A0 = 0). E' una forma di indirizzamento seriale dei registri di comando simile a quella dell'8274.

- **Inizializzazione**

Set Address/Enable                      D0h Write 1 byte

Scrive l'indirizzo del dispositivo nel registro apposito (bit 0-6) e lo abilita (bit 7); all'inizio il registro ha contenuto 0 per default.

Set Endpoint Enable                      D8h Write 1 byte

abilita gli endpoint (bit 0 = 1)

Set Mode                                      F3h Write 2 bytes

il primo byte è la configurazione (tra cui bit 6 e 7 specificano il Modo); il secondo è il divisore N per CLKOUT.



- **Flusso dei dati**

<b>Comando</b>	<b>Destinatario</b>	<b>Valore</b>	<b>N. byte</b>
Read Interrupt Register	Device	F4h	Read 2 bytes
Select Endpoint	Control OUT	00h	Read 1 byte (opt)
	Control IN	01h	Read 1 byte (opt)
	Endpt 1 OUT	02h	Read 1 byte (opt)
	Endpt 1 IN	03h	Read 1 byte (opt)
	Endpt 2 OUT	04h	Read 1 byte (opt)
	Endpt 2 IN	05h	Read 1 byte (opt)
	Read Last Trans. Status	Control OUT	40h
Control IN		41h	Read 1 byte
Endpt 1 OUT		42h	Read 1 byte
Endpt 1 IN		43h	Read 1 byte
Endpt 2 OUT		44h	Read 1 byte
Endpt 2 IN		45h	Read 1 byte
Read Buffer	Select. Endpt	F0h	Read <b>n</b> bytes
Write Buffer	Select. Endpt	F0h	Write <b>n</b> bytes
Set Endpoint Status	Control OUT	40h	Write 1 byte
	Control IN	41h	Write 1 byte
	Endpt 1 OUT	42h	Write 1 byte
	Endpt 1 IN	43h	Write 1 byte
	Endpt 2 OUT	44h	Write 1 byte
	Endpt 2 IN	45h	Write 1 byte
Acknowledge Setup	Select. Endpt	F1h	Nessuno
Clear Buffer	Select. Endpt.	F2h	Nessuno

- Il Read Interrupt Register restituisce due byte: nel primo è indicato per ogni endpoint index se ci sono dati validi nel buffer, nel secondo lo stato di EOT
- Con i Select Endpoint si seleziona un opportuno endpoint; i successivi Read Buffer e Write Buffer ne leggono e scrivono il buffer rispettivamente; il formato degli n byte è il seguente:
  - byte 0: riservato
  - byte 1: numero dei dati
  - byte 2: primo dato
  - byte 3: secondo dato...

Nei cicli di DMA byte 0 e byte 1 non sono inseriti per semplicità: in lettura si determina l'ultimo byte sulla base del pacchetto EOP sul bus, in scrittura o perché il buffer è pieno (max 130 byte) o perché è arrivato l'EOT dal DMA controller opportunamente programmato

- Clear Buffer dichiara che il buffer è stato svuotato e può essere riusato
- Validate Buffer dichiara che il buffer è stato riempito e può essere trasmesso
- Set Endpoint Status manda in STALL l'endpoint
- Acknowledge setup: riabilita i comandi Clear e Validate disabilitati da un SETUP