

1.5 Control bus

Control bus

Le azioni di lettura, scrittura, acquisizione ed emissione sono governate dalla CPU che sincronizza gli altri dispositivi attraverso il controllo bus.

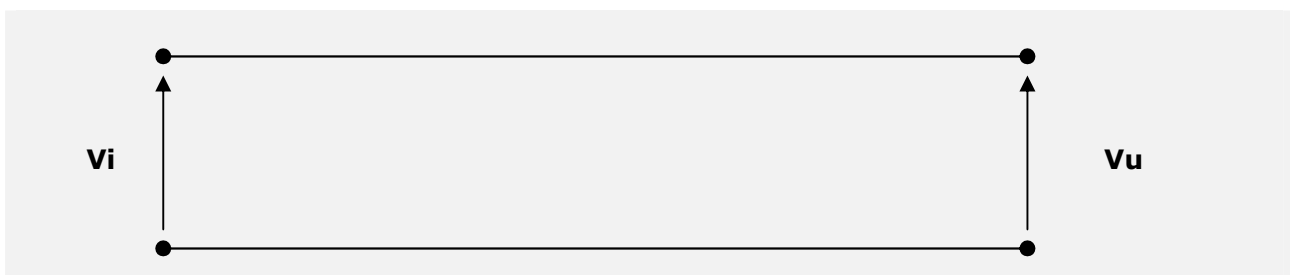
Il control bus è costituito da un insieme di segnali logici. Alcuni di questi escono dalla CPU e quindi sono comandi che la CPU dà agli altri dispositivi mentre altri entrano nella CPU e quindi sono informazioni che gli altri dispositivi danno alla CPU.

Si può quindi affermare che il control bus è bidirezionale, ma con un significato diverso rispetto al data bus. Infatti nel caso del control bus esistono segnali d'ingresso e segnali di uscita ma ciascun segnale non inverte mai la sua direzione mentre nel data bus tutti i segnali sono sia di ingresso che di uscita perché in funzione della operazione svolta la direzione si può invertire.

I segnali di uscita del control bus sono quindi comandi per le periferiche; questi segnali sono a loro volta governati dal 'clock' che è il motore che fa avanzare le azioni della CPU con una cadenza fissata dalla sua frequenza.

Per comprendere la necessità della sincronizzazione e quindi il significato della limitazione imposta dalla frequenza di clock si deve analizzare un modello semplificato di un filo di bus.

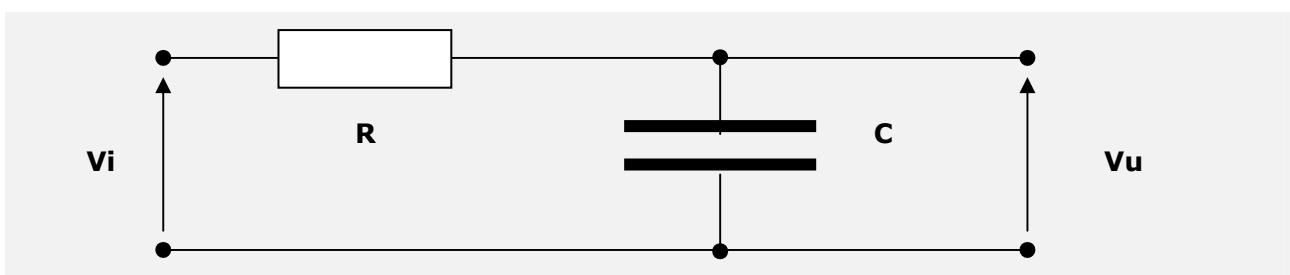
Una linea di data bus è composta da due fili (segnale e massa di riferimento) se questi due fili vengono considerati conduttori ideali il modello grafico che si ottiene è il seguente:



In questo modello un segnale viene trasferito dall'ingresso all'uscita senza distorsione (deformazione della sua forma).

Ma questo modello è inadeguato per descrivere segnali che si evolvono rapidamente perché non tiene conto della resistività del conduttore e degli effetti capacitivi che si presentano tra due conduttori separati da un isolante.

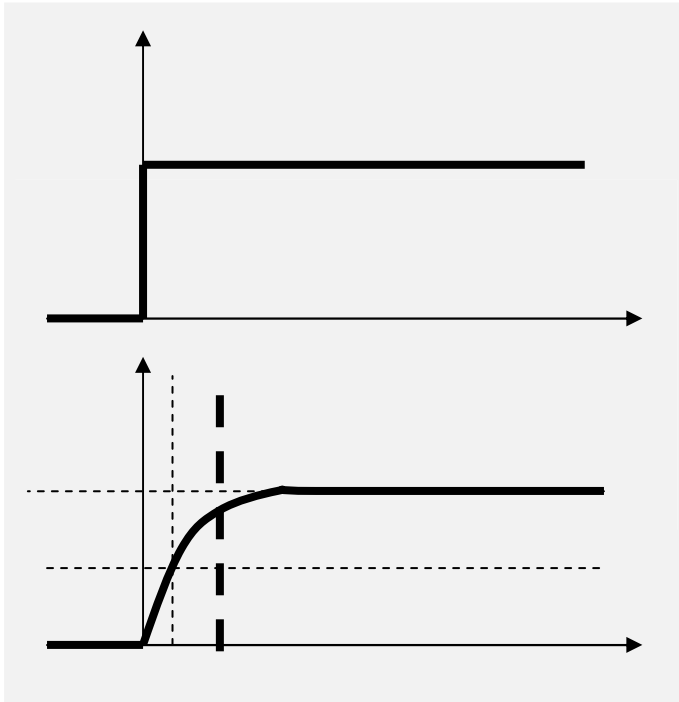
Per descrivere meglio il fenomeno si possono introdurre queste caratteristiche sotto forma di parametri concentrati, cioè continuare a considerare il filo ideale ma inserire in serie una resistenza fittizia che tiene conto della resistività del filo ed inserire in parallelo tra i due fili un condensatore fittizio che tiene conto della capacità tra i due fili.



Questo circuito è già stato studiato nel corso di terza e quindi vi rimando a quel corso per il dettaglio del funzionamento.

Attenzione: questi componenti non esistono fisicamente ma sono un modello per descrivere il fenomeno.

In queste condizioni esaminiamo l'andamento delle risposte ad un gradino in una coppia di grafici cartesiani temporali coordinati.



Il segnale di entrata fa una brusca variazione da 0 a 1. Il segnale di uscita, per il fenomeno di accumulo delle cariche segue un andamento esponenziale che lo porta asintoticamente ad 1. Ipotizzando che la soglia di scatto (livello di transizione tra 0 e 1) sia posta a metà dell'ampiezza del segnale si vede che il segnale di uscita raggiunge la soglia di scatto (e quindi viene riconosciuto come 1 da un circuito logico) in ritardo rispetto al momento in cui il segnale è stato generato.

Questo ritardo non è dovuto alla lentezza del movimento delle cariche ma alla deformazione che ha subito il segnale (distorsione).

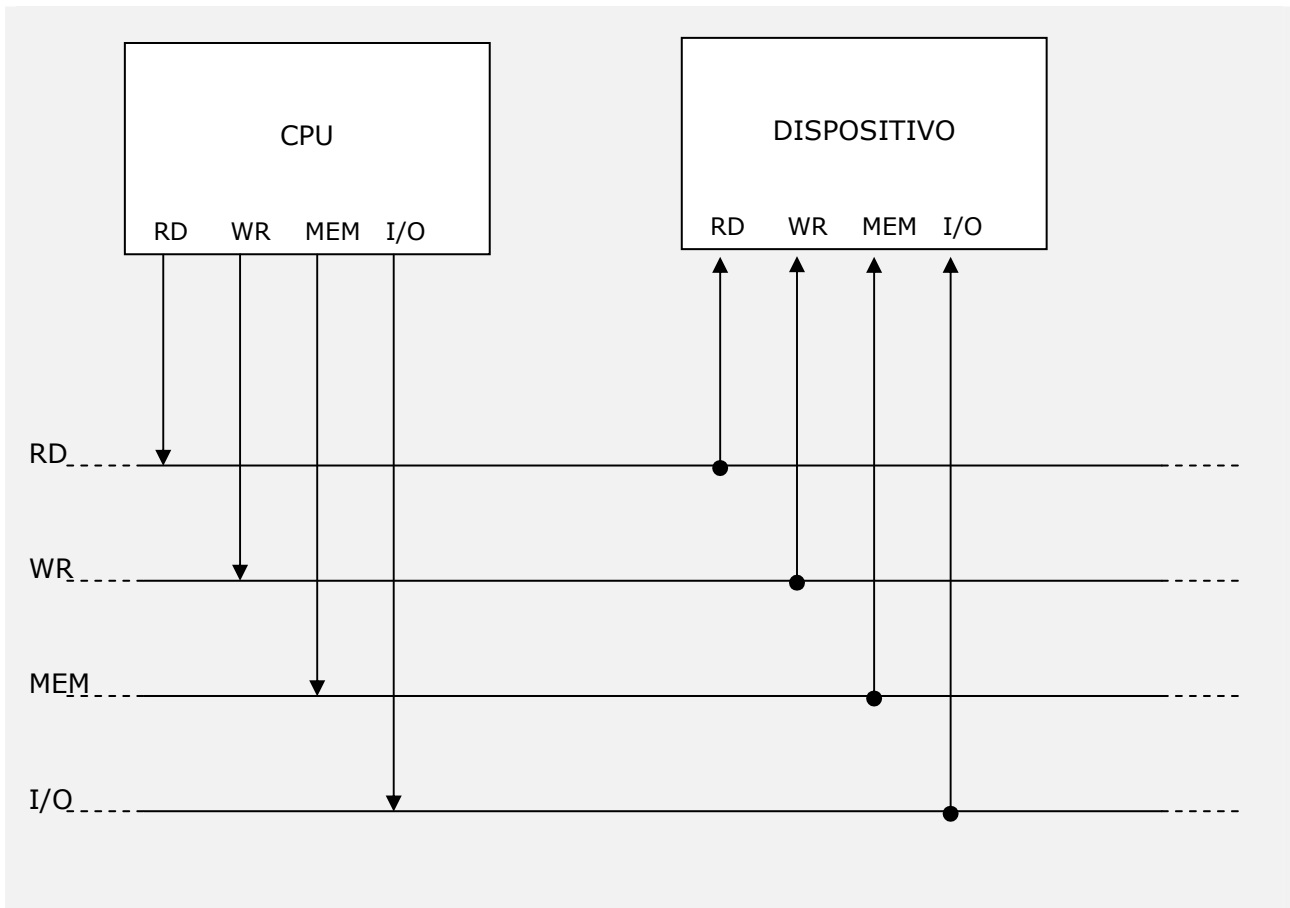
Un dispositivo che campiona un bus deve campionare solo dopo che il transitorio è passato. A questo provvede il segnale di controllo che abilita il campionamento solo quando il bus si è stabilizzato.

La presenza del clock quindi serve per dare tempo ai segnali sul bus di stabilizzarsi per essere campionati.

Controlli di uscita

In questo studio di carattere generale consideriamo solo quattro segnali di uscita del control bus:

- READ: segnale di controllo che indica che l'operazione da eseguire sul bus dati è una lettura, cioè un trasferimento dalla periferia (cella o porta) verso la CPU.
- WRITE: segnale di controllo che indica che l'operazione da eseguire sul bus dati è una scrittura, cioè un trasferimento dalla CPU verso la periferia (cella o porta).
- MEMORY: segnale di controllo che indica che l'operazione da eseguire sul bus dati (lettura o scrittura) è riferita ad una cella di memoria.
- I/O: segnale di controllo che indica che l'operazione da eseguire sul bus dati (lettura o scrittura) è riferita ad una porta di I/O.



Note:

- Questo schema non si riferisce ad alcuna CPU in particolare ma è un modello generale di studio che ricalca situazioni realmente esistenti.
- Allo stesso modo i segnali sono considerati in logica positiva (attivi quando valgono 1) indipendentemente dalla effettiva struttura elettronica di particolari CPU.
- La presenza della coppia di segnali I/O e MEMORY serve per discriminare lo spazio di indirizzamento visto che l'address bus è comune per entrambi gli spazi.
- La coppia READ WRITE non può essere attiva contemporaneamente mentre è lecito che nessuno dei due sia attivo (significa che i bus non sono impegnati)
- La coppia MEM I/O non può essere attiva contemporaneamente mentre è lecito che nessuno dei due sia attivo (significa che non c'è ancora selezione e quindi i bus non sono impegnati)

Tabella di verità dei segnali di controllo di uscita

READ	WRITE	MEM	I/O	Azione
0	0	0	0	Nessuna azione
1	0	0	0	Nessuna azione
0	1	0	0	Nessuna azione
1	1	0	0	Illecita
0	0	1	0	Nessuna azione
1	0	1	0	Lettura della memoria
0	1	1	0	Scrittura della memoria
1	1	1	0	Illecita
0	0	0	1	Nessuna azione
1	0	0	1	Acquisizione di un ingresso
0	1	0	1	Emissione di una uscita
1	1	0	1	Illecita
0	0	1	1	Illecita
1	0	1	1	Illecita
0	1	1	1	Illecita
1	1	1	1	Illecita

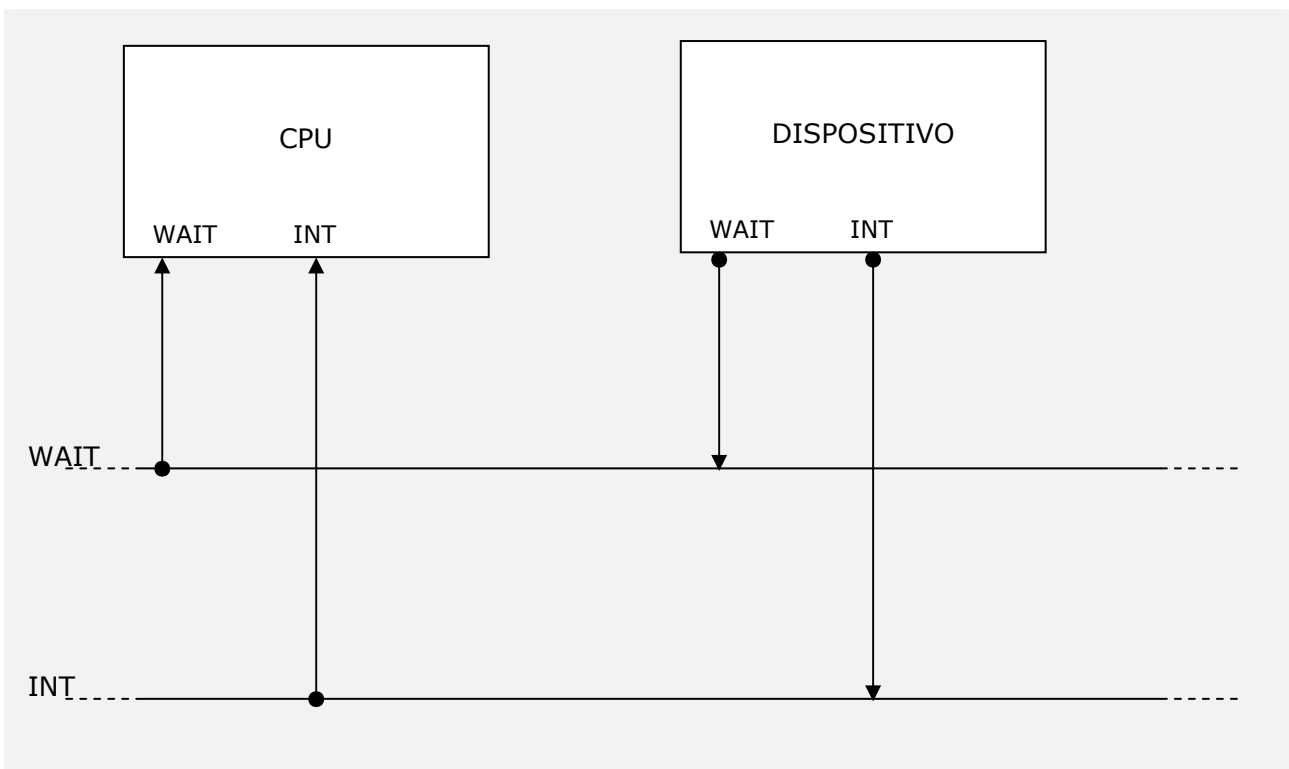
Note:

- Le azioni illecite non si possono verificare perché l'unità di controllo (CU) interna alla CPU è programmata in modo da non generarle mai.
- Le situazioni di nessuna azione invece sono possibili perché ci sono cicli in cui la CPU fa azioni interne che non coinvolgono l'uso dei bus oppure sono possibili durante fasi transitorie per passare da un ciclo ad un altro. In ogni caso in queste fasi il bus non è impegnato da alcun dispositivo e quindi non contiene alcun segnale (bus flottante o floating).

Controlli di ingresso

In questo studio di carattere generale consideriamo solo due segnali di ingresso del control bus:

- **WAIT**: segnale di controllo che indica che la periferica che la CPU ha selezionato con l'address bus non è in grado di reagire subito al comando. La CPU campiona lo stato dell'ingresso WAIT ad ogni ciclo di clock e se lo trova attivo introduce un ciclo fittizio (IDLE) in cui non fa niente attendendo che la periferica si dichiari pronta.
- **INT**: segnale di controllo, proveniente dalle periferiche di I/O che segnala alla CPU la richiesta di attenzione. Questo segnale si rende necessario perché le periferiche di I/O sono enormemente più lente della CPU (ordine di grandezza 1 milione di volte più lente) e quindi i programmi eseguiti dalla CPU non possono attendere il completamento delle operazioni, pena il bloccaggio dell'intero sistema per tempi lunghissimi. Quindi quando viene iniziata una operazione di I/O la CPU passa ad eseguire altri programmi fino a quando non viene interrotta da un segnale di INTERRUPT che segnala che una periferica ha dati da fornire alla CPU oppure a completato una operazione che aveva iniziato. La CPU interrompe immediatamente il programma in corso, passa alla esecuzione di un particolare sottoprogramma (chiamato 'procedura interrompente') che effettua le operazioni di I/O richieste e termina restituendo il controllo della CPU al programma che era stato interrotto.



Note:

- Questo schema non si riferisce ad alcuna CPU in particolare ma è un modello generale di studio che ricalca situazioni realmente esistenti.
- Allo stesso modo i segnali sono considerati in logica positiva (attivi quando valgono 1) indipendentemente dalla effettiva struttura elettronica di particolari CPU.
- La reale connessione del segnale di INT è molto più complessa, questo è solo uno schema di principio. L'effettiva connessione si vedrà più avanti studiando i cicli macchina.
- IL comportamento dei segnali WAIT e INT, è molto diverso, nel primo caso la presenza del segnale blocca l'avanzamento di qualsiasi programma in attesa che la periferica completi il suo compito e quindi viene usato solo per sincronizzare periferiche di poco più lente rispetto alla CPU mentre nel secondo caso la CPU continua a lavorare eseguendo i programmi caricati in memoria centrale interrompendosi quando c'è una richiesta di attenzione ma per eseguire in effetti un altro programma.