

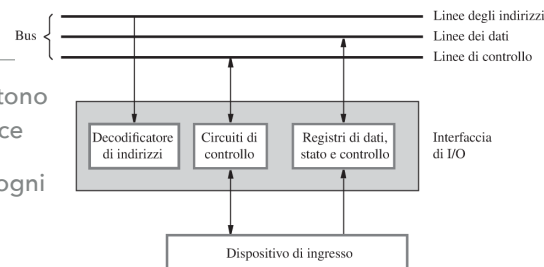
ARCHITETTURA DEGLI ELABORATORI

PROF. TRAMONTANA
www.dmi.unict.it/tramonta

Capitolo 7 Sistema di ingresso e uscita

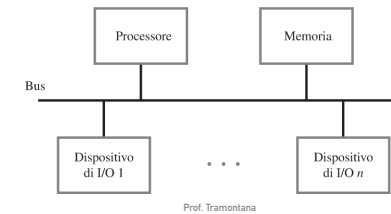
Interfacce di I/O

- ▶ Le tre parti del bus connettono parti distinte delle interfacce
- ▶ Nel memory-mapped I/O ogni interfaccia decodifica l'indirizzo e se questo appartiene all'insieme proprio dell'interfaccia, reagisce ai segnali di controllo e accede alle linee dati
- ▶ Si ha un funzionamento simile nel caso di spazi di indirizzamento separati



Struttura a bus (S. 7.1)

- ▶ Il bus è la più semplice struttura di interconnessione fra processore, memoria e periferiche (tramite le loro interfacce)
- ▶ Il bus è un insieme di linee parallele per la comunicazione fra una coppia di dispositivi alla volta
- ▶ Spesso è partizionato in tre insiemi di linee: bus **indirizzi**, bus **dati**, linee di **controllo**

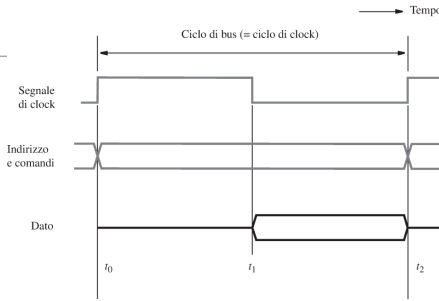


Funzionamento del bus (S. 7.2)

- ▶ Protocollo del bus: insieme di regole che ne governano l'uso. Il protocollo è realizzato mediante specifici segnali di controllo
- ▶ Es. la linea di controllo R/\overline{W} specifica l'operazione di I/O, dove lettura è attiva alta e scrittura è attiva bassa: se la linea vale 1 il processore vuol fare un'operazione di lettura, se vale 0 vuole fare un'operazione di scrittura
- ▶ Le linee di controllo segnalano pure il tempo e quindi la sincronizzazione fra le varie unità: il momento in cui processore e periferiche devono mettere il dato sul bus o acquisirlo dal bus
- ▶ I protocolli si dividono in due famiglie: bus sincroni e bus asincroni
- ▶ Ruoli dei due dispositivi coinvolti in un trasferimento di dati: **master**, ovvero dispositivo che dà inizio all'operazione; **slave**, l'altro dispositivo (una memoria è sempre slave)

Bus sincrono

- ▶ Temporizzazione dei segnali sul bus regolata da un segnale periodico: bus clock, su una linea di controllo, in due fasi (livelli del segnale)
- ▶ Operazione di lettura: a t_0 il master emette l'indirizzo e $R/\bar{W} = 1$; la linea dati da t_0 a t_1 non contiene valori significativi
- ▶ La transizione intermedia **attiva la risposta** dello slave al comando ricevuto sulla linea R/\bar{W} e a t_1 lo slave emette il dato
- ▶ $t_1 - t_0$ deve essere sufficiente per la propagazione nel bus e decodifica dell'indirizzo e dei segnali di controllo nello slave
- ▶ $t_2 - t_1$ deve essere sufficiente per la propagazione nel bus e tempo di setup del registro del master
- ▶ A t_2 il master legge il valore presente sulle linee dato e lo mette nei suoi registri interni



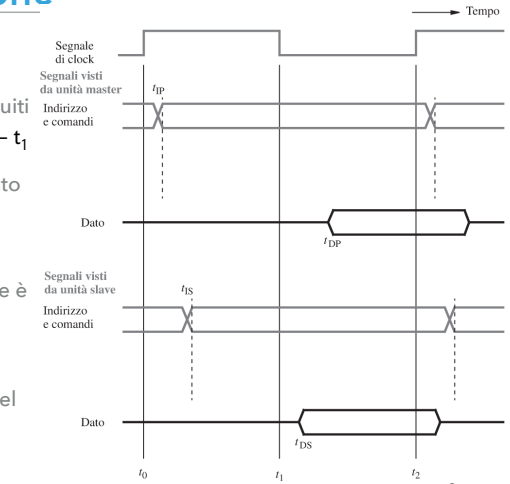
- ▶ I gruppi di linee del bus sono rappresentati tramite doppie linee che si incrociano quando una o più linee cambiano valore
- ▶ Quando il gruppo di linee del bus non ha un valore significativo o è in alta impedenza si disegna una linea intermedia

Prof. Tramontana

5

Ritardi di propagazione

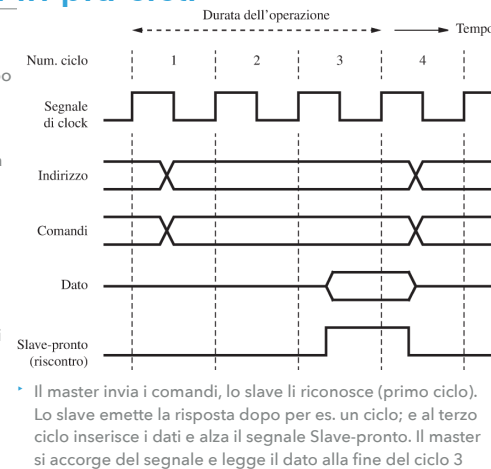
- ▶ Temporizzazione da due punti di vista diversi
- ▶ Ritardi di propagazione anche nei circuiti di output dei dispositivi: $t_{IP} - t_0$ e $t_{DS} - t_1$
 - ▶ IP: Indirizzo unità Principale; DS: dato unità Secondaria
- ▶ Il tempo utile allo slave per recepire indirizzo e comandi diventa $t_1 - t_{IS}$ che è più piccolo di $t_1 - t_0$
- ▶ Il dato deve rimanere stabile sul bus dopo t_2 per almeno il *tempo di hold* nel registro del master



Prof. Tramontana

Trasferimento di dati in più cicli

- ▶ Con il protocollo sincrono precedente, il periodo di clock è tarato sul tempo di risposta del dispositivo più lento e dal tempo di propagazione massimo. Tutte le unità lavorano con il periodo dell'unità più lenta
- ▶ Inoltre, non si ha garanzia che lo slave abbia eseguito il comando, e il master legge comunque il dato al tempo t_2
- ▶ Con il protocollo a più cicli si avrà il segnale di conferma dallo slave (Slave-pronto)
- ▶ Inoltre, la durata è variabile, si usa una frequenza elevata così da poter avere tempi brevi per slave veloci e si estende il numero di cicli per slave lenti
- ▶ Numero di cicli di attesa del segnale di conferma variabile ma limitato

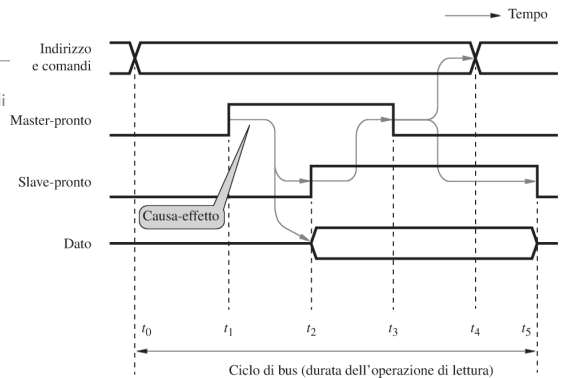


Prof. Tramontana

7

Bus asincrono

- ▶ Temporizzazione senza bus clock ottenuta tramite lo scambio di segnali di conferme
- ▶ Due linee di controllo: segnalazione (pilotata dal master) e riscontro (pilotata dallo slave), linee attive alte, usate per **handshake**
- ▶ Il master manda indirizzo e comando e quindi mette alta la linea Master-pronto
- ▶ Le unità decodificano l'indirizzo e interpretano il comando
- ▶ L'unità slave indirizzata mette il dato e alza la linea Slave-pronto
- ▶ Il master vista l'attivazione di Slave-pronto disattiva Master-pronto, toglie indirizzo e comando e legge il dato



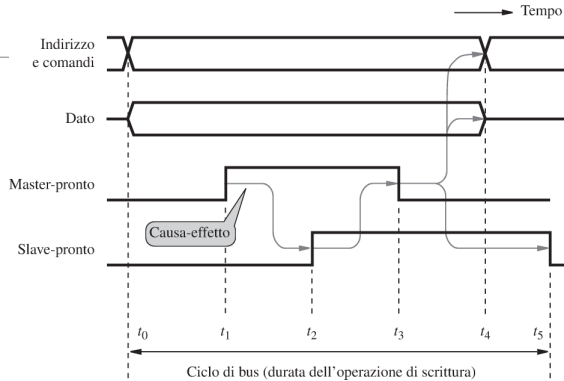
- ▶ Il diagramma mostra tramite gli archi la relazione di causalità fra transizioni di stato delle linee del bus
- ▶ L'intervallo $t_1 - t_0$ è lo sfasamento temporale (skew) fra le unità, dovuto a ritardi variabili del segnale che dal master si propaga agli slave presenti a distanze diverse

Prof. Tramontana

8

Bus asincrono

- ▶ Nel caso di un'operazione di scrittura, il master imposta le linee del bus dati insieme a quelle di indirizzo e di comando
- ▶ Il segnale di conferma dello slave segnala l'avvenuto caricamento dei dati nel suo registro



Prof. Tramontana

9

Pilotaggio del bus

- ▶ In ogni momento un solo dispositivo può essere abilitato all'invio del segnale, tutti gli altri devono avere i loro bus driver disabilitati e per questo si usano porte a tre stati (tri-state)
- ▶ Bus driver (pilota del bus): porta logica che è abilitata a inviare l'output su una linea del bus

Prof. Tramontana

11

Confronto bus sincrono e asincrono

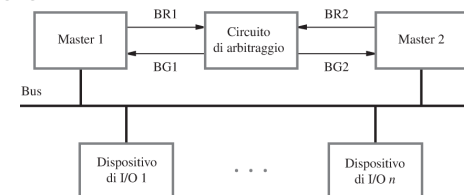
- ▶ Fattori di valutazione: semplicità del circuito di interfaccia al bus, flessibilità di aggiungere unità a velocità differenti, tempo di trasferimento sul bus, capacità di rilevare errori dovuti a unità inesistenti o malfunzionamenti
- ▶ Il protocollo del bus asincrono grazie alla procedura di handshake fa a meno del segnale di clock
- ▶ Con il bus asincrono gli eventuali ritardi (dovuti a sfasamenti temporali, carico elettrico sul bus) mutevoli, si riescono ad accettare, poiché la temporizzazione dei segnali di controllo si adatta alla nuova situazione
- ▶ Col bus sincrono, si deve distribuire lo stesso segnale di clock a tutte le unità, e tenere sotto controllo i ritardi poiché non posso eccedere limiti molto stretti
- ▶ Col bus asincrono la procedura di handshake limita la frequenza di trasferimento, poiché i segnali di handshake devono viaggiare avanti e indietro per tutto il bus due volte
- ▶ Col bus sincrono le frequenze di trasferimento sono più elevate e per unità lente basta aggiungere cicli di clock per prolungare l'operazione di trasferimento
- ▶ La maggior parte dei calcolatori attuali usa bus sincroni

Prof. Tramontana

10

Arbitraggio del bus (S. 7.3)

- ▶ Più dispositivi collegati al bus possono avere il ruolo di master (processore, controllori di I/O), quindi vi può essere la necessità di accedere a un certo slave nello stesso momento
- ▶ La soluzione è presa tramite un circuito di arbitraggio
 - ▶ Ciascun dispositivo invia una richiesta di uso
 - ▶ L'arbitro associa una priorità ai dispositivi e se riceve richieste nello stesso tempo concede la priorità al dispositivo a priorità più alta
- ▶ La priorità può derivare dalla necessità di accedere al bus senza ritardi per evitare errori



Due master, due linee di richiesta (BR1, BR2) e due linee per concedere (grant) il bus (BG1, BG2)

Prof. Tramontana

12

Esempio di Arbitraggio

- ▶ Priorità decrescente, tre master
- ▶ Il master due invia la richiesta BR2 e non essendoci altre richieste ottiene l'uso BG2. Quando completa la sua operazione di trasferimento rilascia il bus disattivando BR2.
- ▶ Quindi, entrambi i master 1 e 3 avevano attivato la richiesta, ma 1 ha priorità e BG1 viene attivato dopo che BG2 è stato disattivato
- ▶ Quando BR1 viene disattivato, BG1 viene disattivato e BG3 viene attivato

