

2.1.3 Dispositivi di ingresso uscita (Input/Output: I/O)

Per dispositivi di ingresso-uscita si intendono tutte quelle apparecchiature mediante le quali il calcolatore comunica con l'esterno, per acquisire nuove informazioni (**input**), da utilizzare nelle elaborazioni successive, o per inviare messaggi (**output**) relativi ai risultati ottenuti. È usuale indicare questi dispositivi anche con il nome di **periferiche**.

Si è visto che la elaborazione dei dati posti nella memoria centrale avviene ad un ritmo imposto dalla esecuzione delle istruzioni del programma, alla cadenza del segnale di clock di frequenza molto elevata. Per avere un dato di riferimento, per CPU e memorie poco veloci, si può supporre che l'esecuzione di una istruzione richieda un tempo medio di un microsecondo, anche se, in realtà, le istruzioni hanno durata che dipende dalla complessità dell'operazione eseguita. Con riferimento al programma di **Fig. 2.1.3.1**, si può, approssimativamente, ritenere che nel primo μsec la CPU raccolga un dato di 16 bit dalla memoria, nei successivi due μsec raccolga due volte lo stesso dato da sottrarre al primo e nell'ultimo μsec trasferisca il risultato in memoria. Le CPU di ultima generazione consentono, allo stato dell'arte, di ridurre di dieci volte, o più, il tempo di esecuzione dello stesso programma

Evidentemente, un modo così rigidamente preordinato di operare, male si adatta alla diversità delle periferiche, che si differenziano essenzialmente per:

- **Velocità di trasmissione dei dati:** uguale al numero di dati trasmessi nell'unità di tempo. Si possono considerare *periferiche lente* quelle che sono in grado di trasmettere pochi bit al secondo, come ad esempi semplici dispositivi elettromeccanici costituiti da pulsanti, fine corsa, relè, ecc. *Periferiche di media velocità* capaci di trasferire fino ad alcune migliaia di dati al secondo (terminali video da 300 a 19200 bit/sec, stampanti da 100 a 10000 byte/sec). *Periferiche ad alta velocità* capaci di trasferire fino ad alcune decine di milioni di dati al secondo (dischi magnetici dell'ordine di 10^7 byte/sec, convertitori A/D e D/A dell'ordine di 10^7 dati/sec, dispositivi di reti di calcolatori dell'ordine di 10^7 bit/sec). In conclusione, si può osservare che la velocità di funzionamento della CPU è notevolmente superiore a quella della maggior parte delle periferiche, che, frequentemente, per il loro funzionamento, attivano organi elettromeccanici (vedi ad esempio le stampanti), caratterizzati da velocità di movimento di gran lunga inferiore a quella con cui cambiano stato i componenti elettronici interni alla CPU.
- **Lunghezza della parole trasmessa:** uguale al numero di bit trasmessi contemporaneamente. Questo valore determina il parallelismo del dispositivo. In particolare si parla di **trasmissione seriale** quando la periferica riceve o trasmette un bit alla volta. In questo modo un byte viene trasmesso dopo otto impulsi di clock.
- **Natura e livello dei segnali:** si possono avere segnali di tensione e/o di corrente variabili in campi di valori (**range**) molto diversificati. I calcolatori utilizzano solitamente segnali binari di tensione con un range di pochi volt (tipico $0V \div 5V$).

Per adattare queste caratteristiche fisiche a quelle del calcolatore, in modo da consentire una corretta trasmissione delle informazioni, bisogna interporre tra i due sistemi degli appropriati **dispositivi di interfaccia**, come mostrato in **Fig.2.1.3.1**. Questi dispositivi sono sostanzialmente costituiti di due parti: 1) una collegata direttamente al bus del processore e capace di "comprendere" i segnali che quest'ultimo le invia; 2) una collegata direttamente alla periferica che realizza le funzioni logiche da essa previste.

L'interfaccia, quindi, ha il compito di sincronizzare due sistemi che hanno modalità intrinseche molto diverse di produrre e trasmettere informazioni. Il processore può dar inizio al trasferimento di un dato solo quando è sicuro che il dispositivo è pronto ad eseguire questa operazione, per cui debbono essere stabilite una serie di regole che entrambi gli interlocutori debbono rispettare affinché l'operazione vada a buon fine. L'insieme di queste regole definisce quello che si chiama **protocollo di comunicazione**.

Per raggiungere questo scopo il dispositivo di interfaccia si può pensare costituito da un registro (o, eventualmente, più di uno) in cui viene temporaneamente memorizzato il dato scambiato, chiamato **buffer**, al quale si aggiunge un registro, generalmente di un solo bit, che codifica lo stato del dispositivo periferico, chiamato **registro di stato**. Dato che il protocollo cambia a seconda che la periferica sia di ingresso o di uscita, si descriveranno separatamente, in seguito, due possibili soluzioni adatte a risolvere questo problema di sincronismo.

2.1.4 Protocollo di comunicazione per un dispositivo di ingresso

La Fig. 2.1.4.1 mostra, in base alle considerazioni precedenti il sistema di interfaccia per una periferica di ingresso.

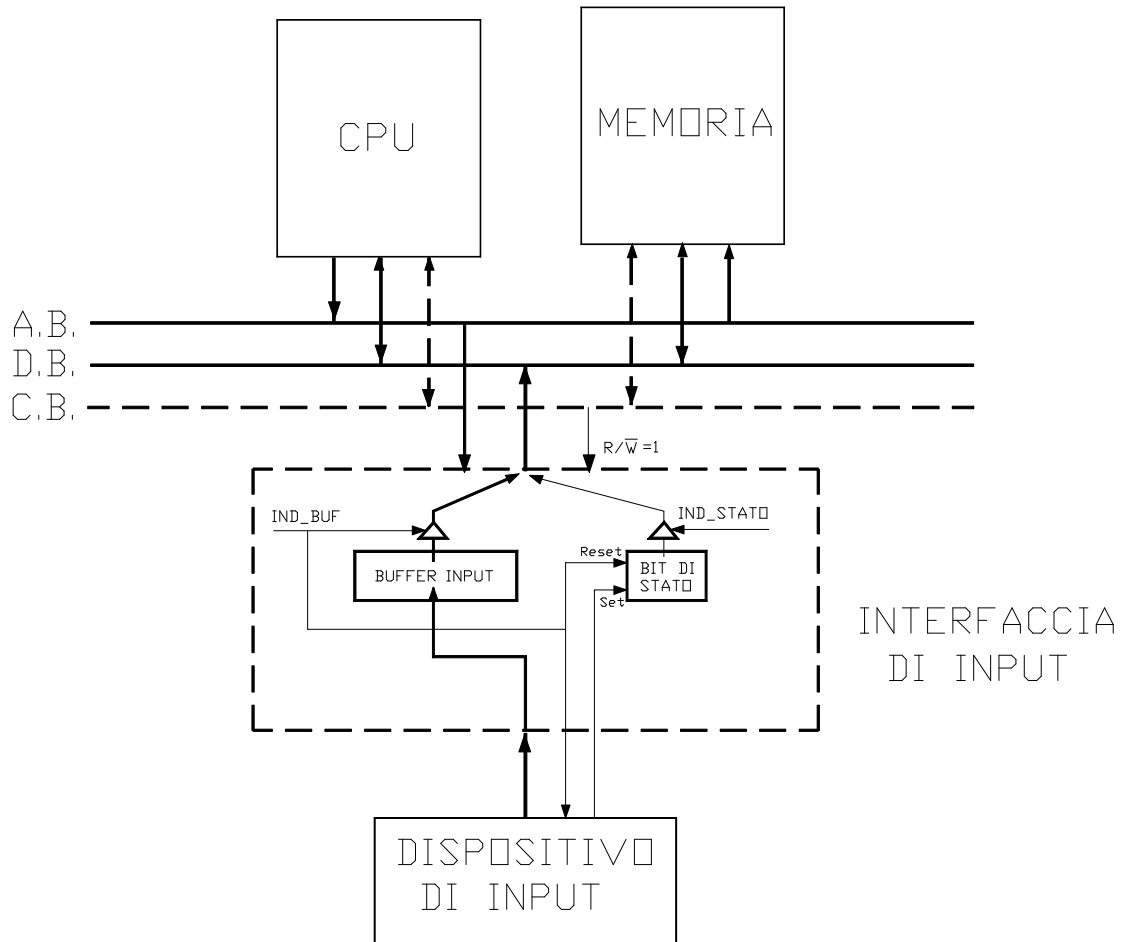


Fig. 2.1.4.1 Schema funzionale dell'interfaccia di una periferica di ingresso.

Il dispositivo di interfaccia mostra i seguenti collegamenti al sistema:

- **lato processore:** la presenza del segnale di $R/\bar{W}$, uguale a quello utilizzato per i trasferimenti con la memoria centrale, e dell'indicazione, con nome simbolico, dell'indirizzo a fianco di ogni registro (IND_BUF per il buffer e IND_STATO per il registro che contiene il bit di stato), vuol significare che la CPU "vede" il dispositivo di interfaccia come due normali locazioni di memoria (I/O *memory mapped*). Si è indicato, inoltre, il bit di controllo $R/\bar{W}=1$, per significare che il processore esercita, sui due registri, solo l'operazione di lettura. I due registri, interni al dispositivo, non sono indipendenti, come le locazioni della memoria, ma esiste una logica interna (non rappresentata) per cui l'operazione di lettura, effettuata dal processore, attiva una linea di Reset che annulla il bit di stato. In questo modo si stabilisce il fatto che il dato presente sul buffer è già stato utilizzato (in gergo "consumato") e che il prossimo dato utile sarà disponibile solo quando questo bit sarà posizionato a uno. La stessa linea di Reset agisce sul dispositivo e lo "stimola" alla produzione di un nuovo dato.
- **lato dispositivo di ingresso:** oltre al collegamento in ingresso con il buffer è presente un segnale, ancora di ingresso, che agisce sul Set del registro di stato. Il posizionamento a uno di questo bit, da parte del dispositivo, segnala, al processore, che nel buffer vi è un nuovo dato disponibile.

Con una struttura di questo tipo le regole da rispettare, da parte dei due sistemi, perché possano sincronizzarsi correttamente, sono le seguenti:

- 1) Il processore legge il registro di stato (un bit) e, se trova il bit a zero, *si pone in attesa* che questo bit diventi uno, rileggendo ripetutamente questo registro. Solo quando il bit di stato è uno il processore “capisce” che nel buffer vi è un nuovo dato e si appresta a leggerlo. La lettura del dato riporta a zero il bit di stato, per segnalare che il dato è stato consumato e per stimolare il dispositivo a produrne un altro.
- 2) Quando il dispositivo riceve dal processore il segnale che lo stimola a produrre un nuovo dato, con i suoi tempi di funzionamento, *completa l'operazione* di scrittura del dato sul buffer dell'interfaccia, agendo, con una propria linea, sul Set del registro di stato, per porre a uno il bit di stato necessario a comunicare al processore la presenza di un nuovo dato disponibile.

La logica di questo protocollo di comunicazione è illustrata dal flowchart di **Fig. 2.1.4.2**.

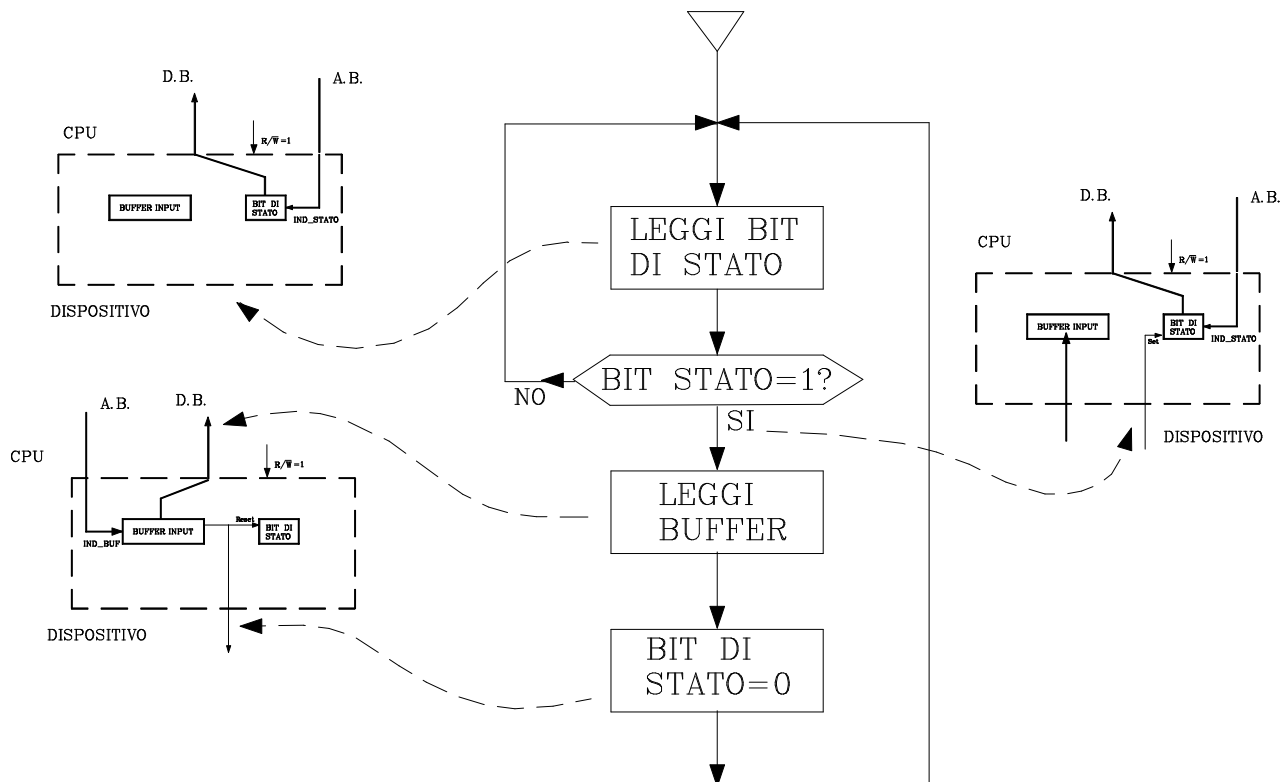


Fig. 2.1.4.2 Flowchart che rappresenta il protocollo di comunicazione di una periferica di ingresso.

La logica del flowchart di **Fig. 2.1.4.2** mostra chiaramente che, qualunque sia la velocità del dispositivo periferico, vi è la corretta sincronizzazione tra i due sistemi, dovuta al fatto che il processore aspetta, in un loop, che il dispositivo gli comunichi, tramite il Set del bit di stato, l'informazione di *dato pronto* e, d'altro canto, il dispositivo aspetta che il processore gli comunichi, tramite il segnale di Reset del bit di stato, l'informazione di *dato ricevuto* e la disponibilità di riceverne un altro.

2.1.5 Protocollo di comunicazione per un dispositivo di uscita

La Fig. 2.1.5.1 mostra, sulla base delle stesse considerazioni fatte per la periferica di ingresso, il sistema di interfaccia di una periferica di uscita.

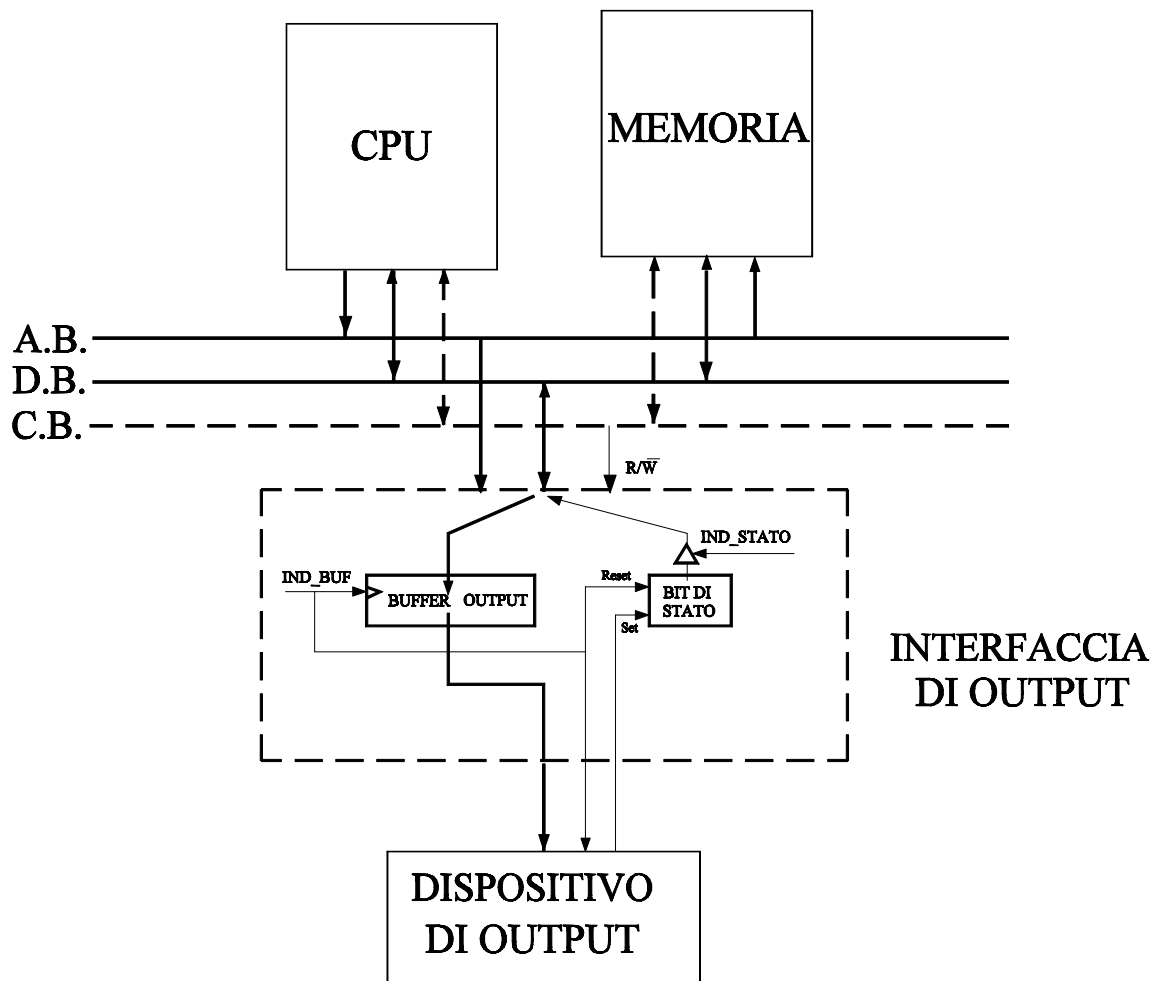


Fig. 2.1.5.1 Schema funzionale dell'interfaccia di una periferica di uscita.

Il dispositivo di interfaccia mostra i seguenti collegamenti al sistema:

- **lato processore:** del tutto simile a quello di ingresso, con la differenza che per il D.B. si è mantenuta la possibilità di trasferimenti bidirezionali e che per il bit di $R/\bar{W}$ sono previsti entrambi i valori dato che il processore deve poter esercitare sia l'operazione di lettura che quella di scrittura sul modulo di interfaccia. L'operazione di lettura viene eseguita sul registro di stato mentre quella di scrittura sul buffer di uscita. Anche in questo caso i due registri, interni al dispositivo, non sono indipendenti, ma esiste una logica interna (non rappresentata) per cui l'operazione di scrittura, effettuata dal processore, attiva una linea di Reset che annulla il bit di stato. In questo modo il processore comunica che vi è un nuovo dato presente sul buffer, pronto per esser utilizzato (in gergo "prodotto") e che verrà scritto un nuovo dato solo quando questo bit sarà posizionato a uno. La stessa linea di Reset agisce sul dispositivo e lo "stimola" a "consumare" il dato.
- **lato dispositivo di uscita:** oltre al collegamento in uscita con il buffer è presente lo stesso segnale di ingresso, visto per la periferica di input, che agisce sul Set del registro di stato. Il posizionamento a uno di questo bit, da parte del dispositivo, segnala, al processore, che il dato presente nel buffer è stato consumato.

Con una struttura di questo tipo le regole da rispettare, da parte dei due sistemi perché possano sincronizzarsi correttamente, sono le seguenti:

- 3) Il processore legge il registro di stato (un bit) e, se trova il bit a zero, *si pone in attesa* che questo bit diventi uno, rileggendo ripetutamente questo registro. Solo quando il bit di stato è uno il processore “capisce” che il dato nel buffer è stato utilizzato e si appresta a scriverne uno nuovo. La scrittura del dato riporta a zero il bit di stato, per segnalare la presenza nel buffer del nuovo dato e per stimolare il dispositivo a consumarlo.
- 4) Quando il dispositivo riceve dal processore il segnale che lo stimola a utilizzare un nuovo dato, con i suoi tempi di funzionamento, *completa l'operazione* di lettura del dato dal buffer dell'interfaccia, agendo, con una propria linea, sul Set del registro di stato, per porre a uno il bit di stato necessario a comunicare al processore che il dato presente nel buffer è stato regolarmente consumato.

Questo protocollo di comunicazione è illustrato dal flowchart di **Fig. 2.1.5.2**.

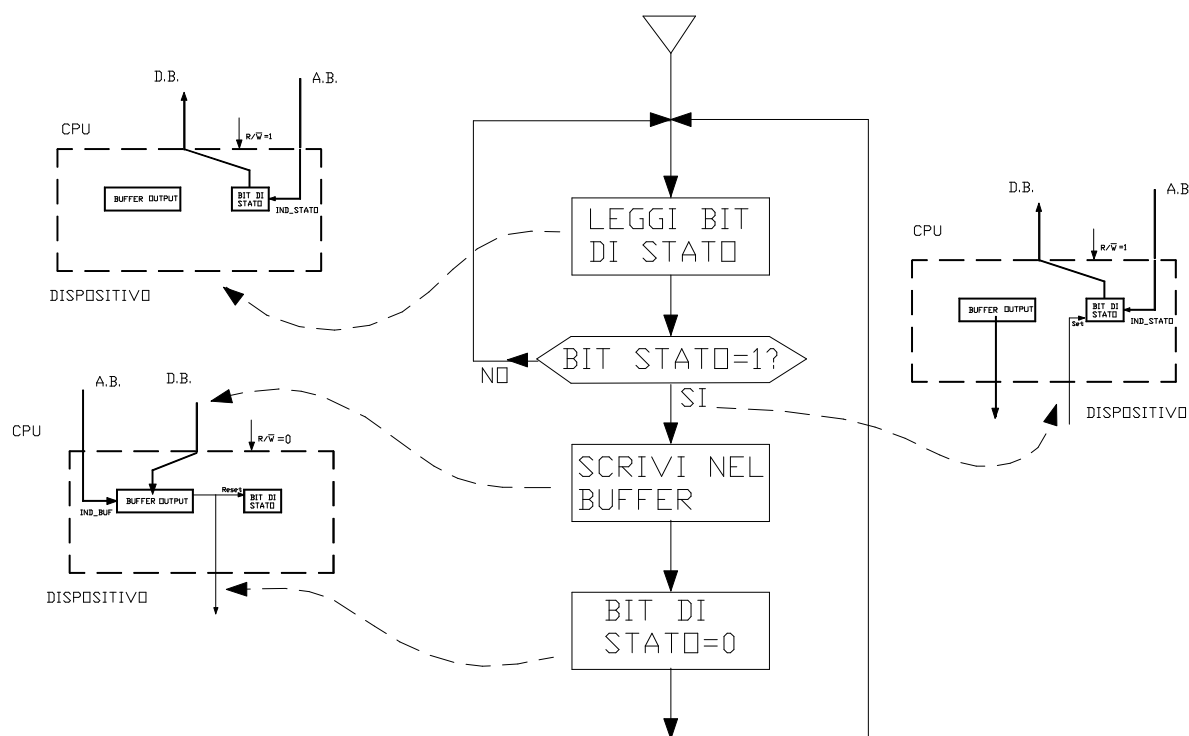


Fig. 2.1.5.2 Flowchart che rappresenta il protocollo di comunicazione di una periferica di uscita.

La logica del flowchart di **Fig. 2.1.5.2** è simile a quella vista per la periferica di ingresso e differisce da questa solo per il tipo di operazione effettuata dal processore, scrittura invece di lettura del buffer di interfaccia. Va rilevato che, in questo caso, l'informazione di *dato pronto*, inviata dal processore al dispositivo, è associata al valore zero del bit di stato, d'altro canto, mentre l'informazione di *dato ricevuto*, inviata dal dispositivo al processore, è associata al valore uno del bit di stato.