

# I/O Programmato

**Materiale di studio**

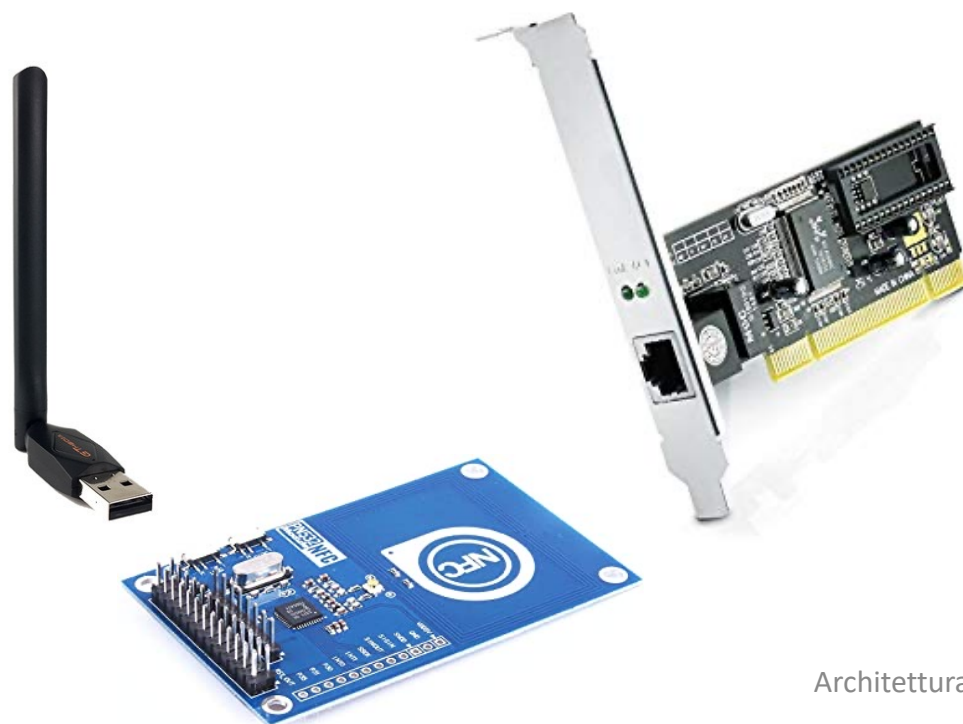
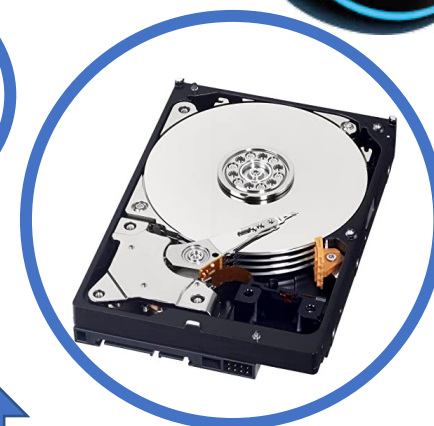
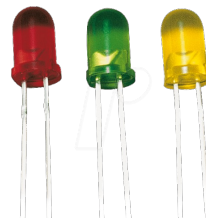
- Paragrafo 3.2
- Capitolo 7

# Cosa è la gestione dell'I/O

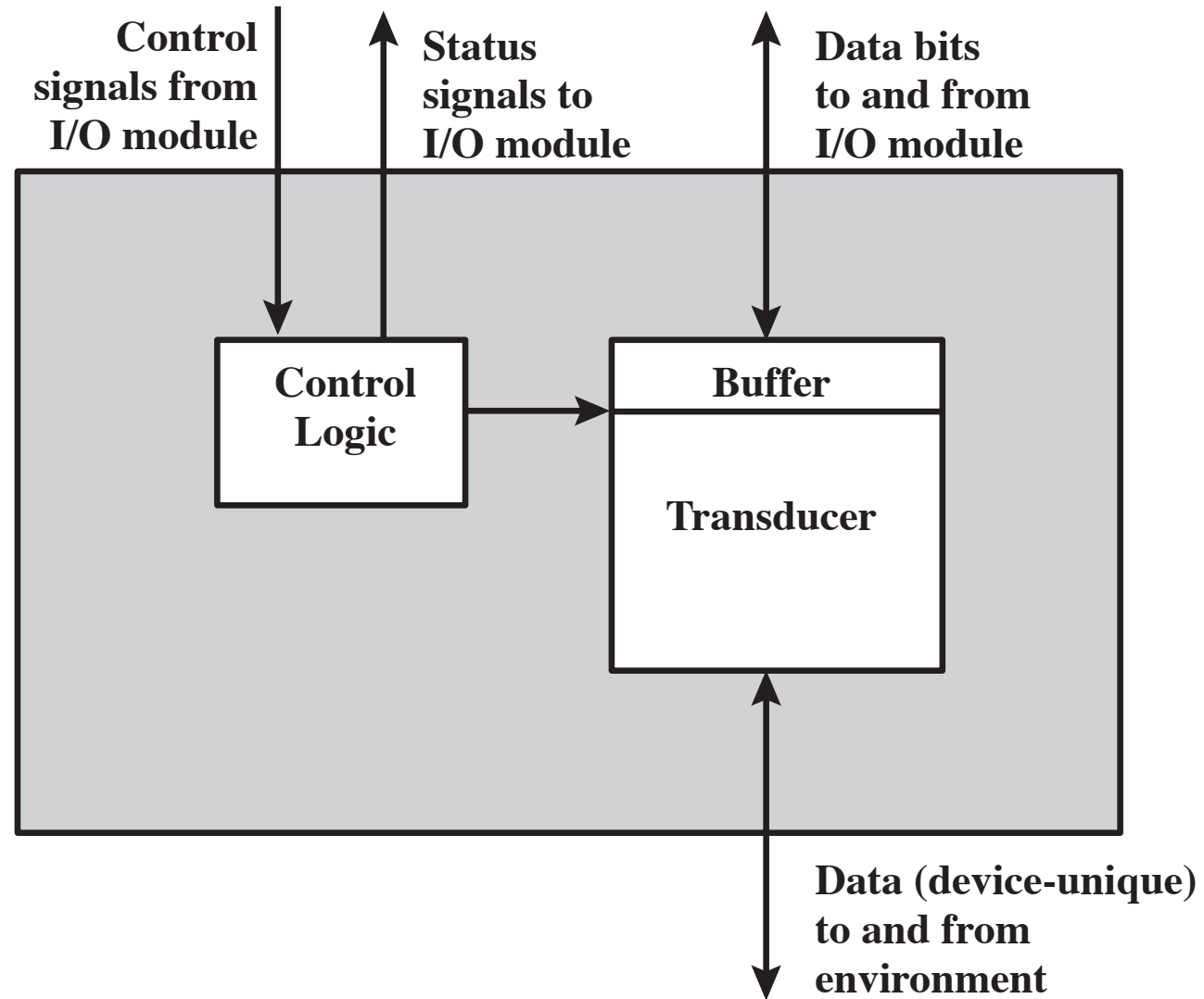
- La sequenza di operazioni che il processore deve eseguire per
  - Leggere dei dati da un dispositivo esterno
  - Scrivere dei dati su un dispositivo esterno

# Diversi tipi di I/O

- Human readable
- Machine readable
- Communication



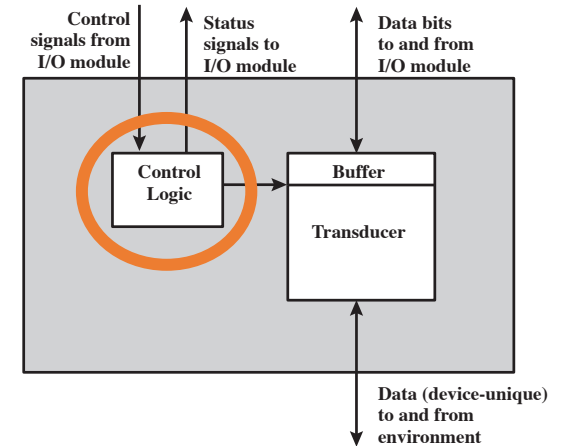
I dispositivi di memoria possono essere visti  
come dispositivi I/O



**Figure 7.2 Block Diagram of an External Device**

# La logica di controllo

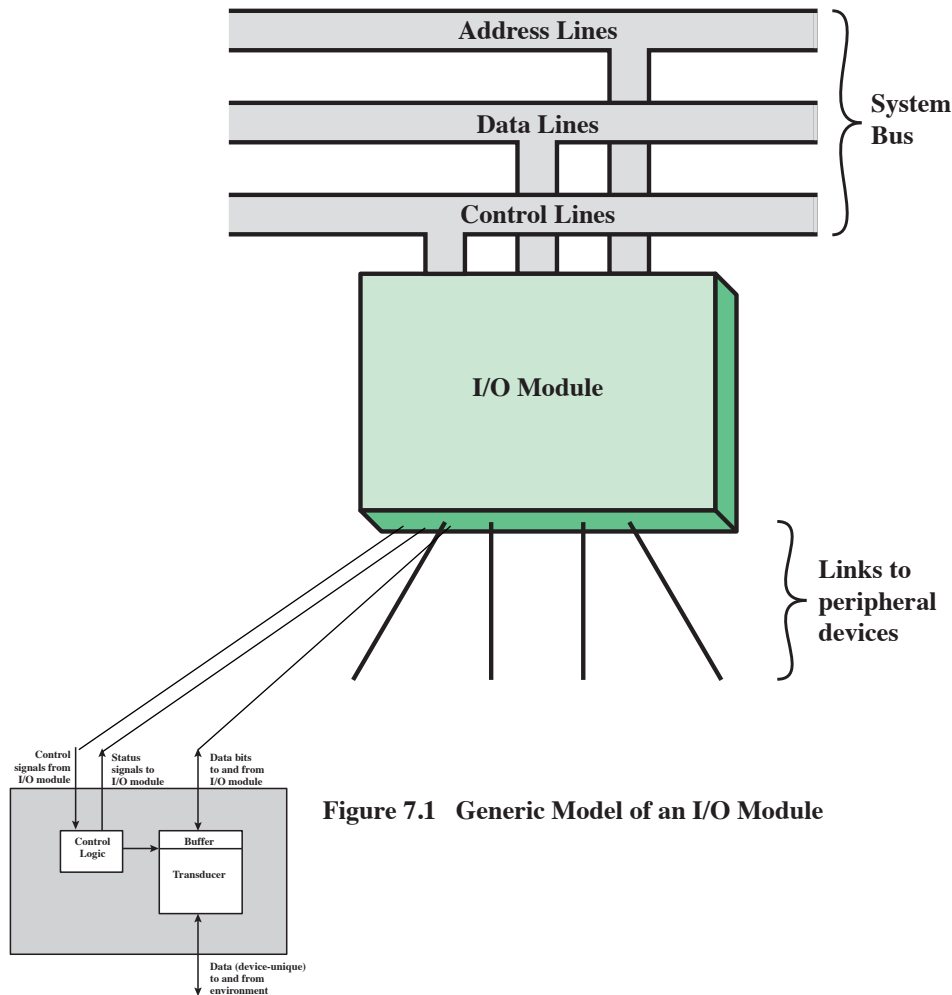
- Le operazioni da compiere possono essere molto diverse a seconda del dispositivo connesso!
- Per garantire che i dispositivi riescano a comunicare con il nostro elaboratore, la loro comunicazione deve avvenire attraverso un **protocollo di I/O** predefinito
- Inoltre i dispositivi saranno fisicamente connessi al nostro elaboratore attraverso una interfaccia standard, rappresentata da un **modulo di I/O**



# Il modulo di I/O

## Il modulo di I/O

- Da un lato è collegato al nostro elaboratore attraverso il bus di sistema
- Dall'altro è collegato ad (uno o più) dispositivi di I/O



# Il modulo I/O

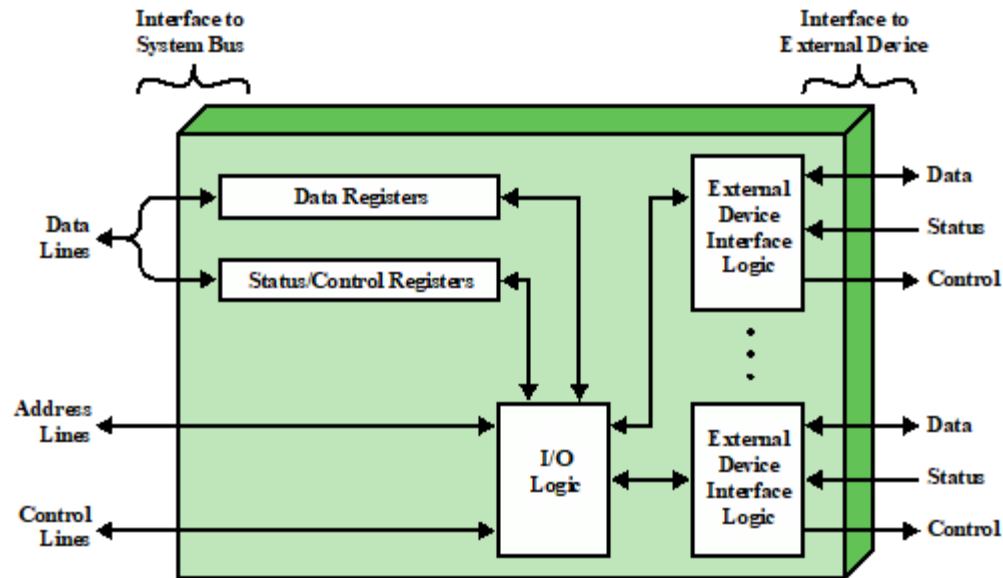


Figure 7.3 Block Diagram of an I/O Module

## Funzioni

- Controllo e temporizzazione
- Comunicazione con il processore
- Comunicazione con il dispositivo
- Data buffer
- Controllo degli errori

# I/O Programmato

- Nel Programmed I/O, il processore si fa carico direttamente dell'esecuzione delle istruzioni per la lettura dei dati da un dispositivo di input (e la scrittura dei dati su un dispositivo di output)
- Grazie al modulo di I/O il processore può fare questo usando quattro tipi di comando:
  - Controllo
  - Test
  - Lettura
  - Scrittura



# I/O Programmato: lettura

- Controllo
- Test
- Lettura

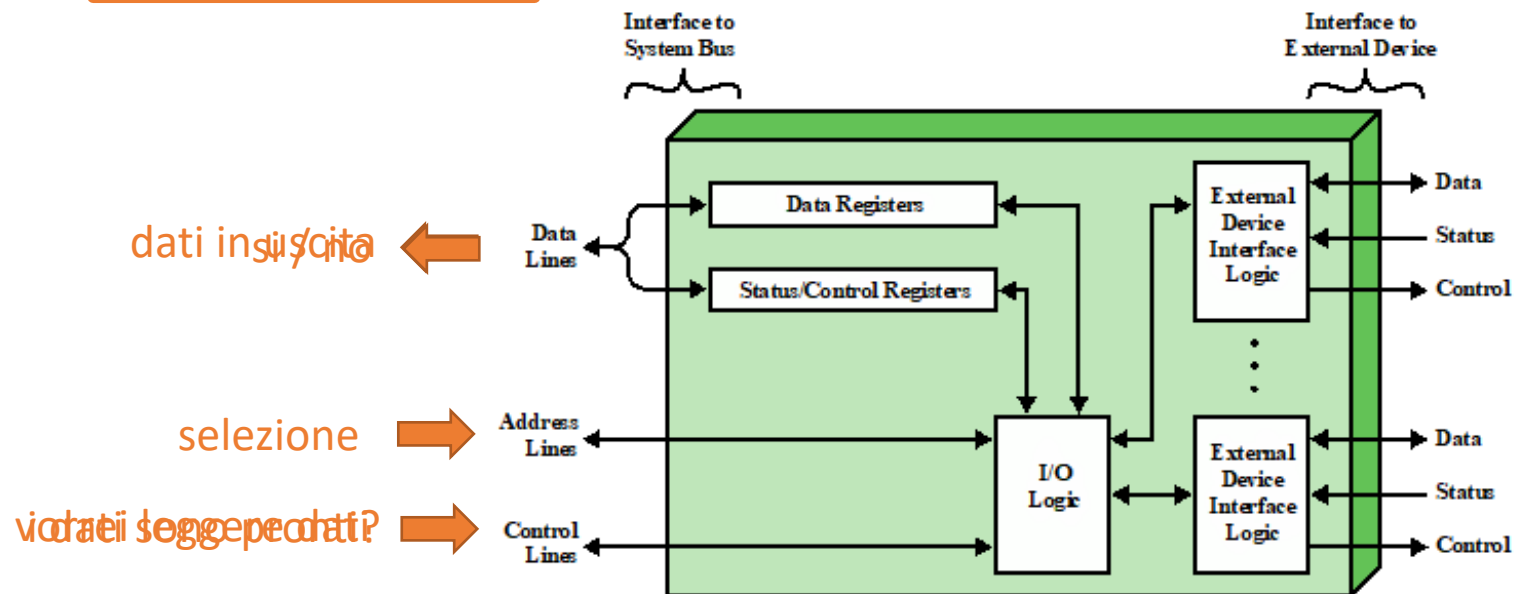
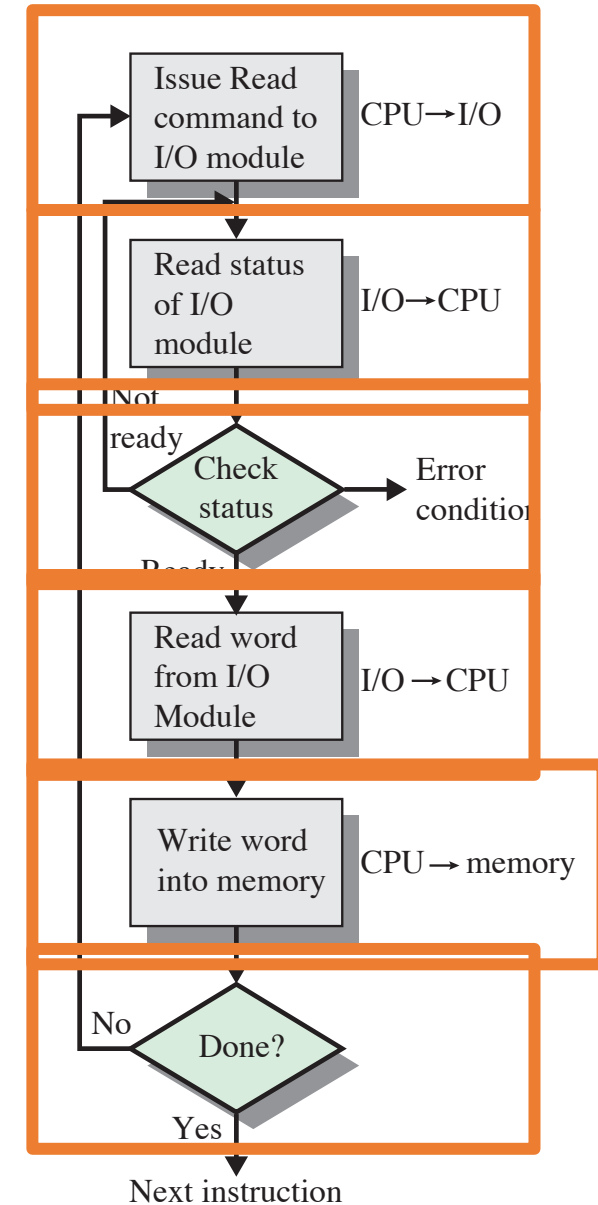


Figure 7.3 Block Diagram of an I/O Module



# I/O Programmato: accesso al dispositivo

## Memory Mapped

- I dispositivi di I/O sono collegati con le stesse linee del bus usate per la memoria
- I data buffer dei moduli I/O sono raggiungibili con dei normali indirizzi di memoria (tipicamente quelli più bassi)
- Il processore ci può scrivere con gli stessi comandi con cui accede alla memoria

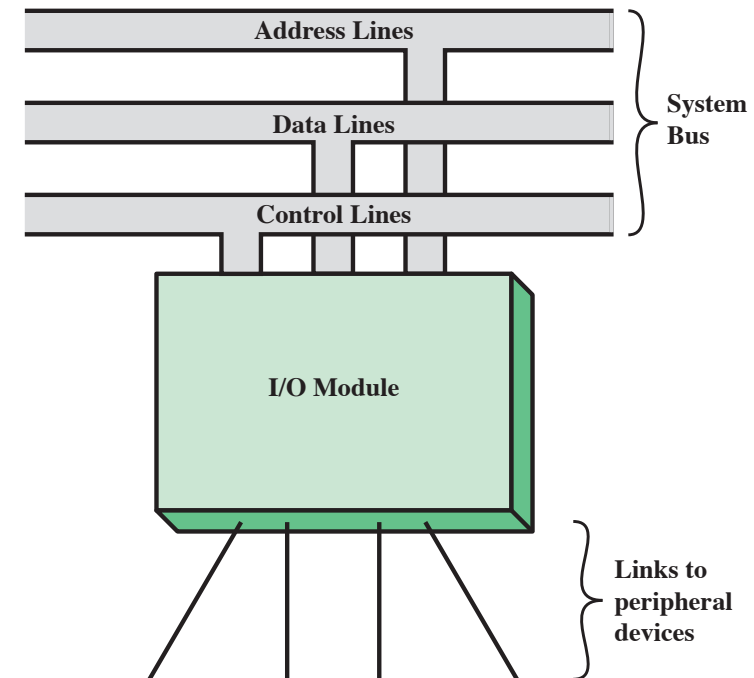
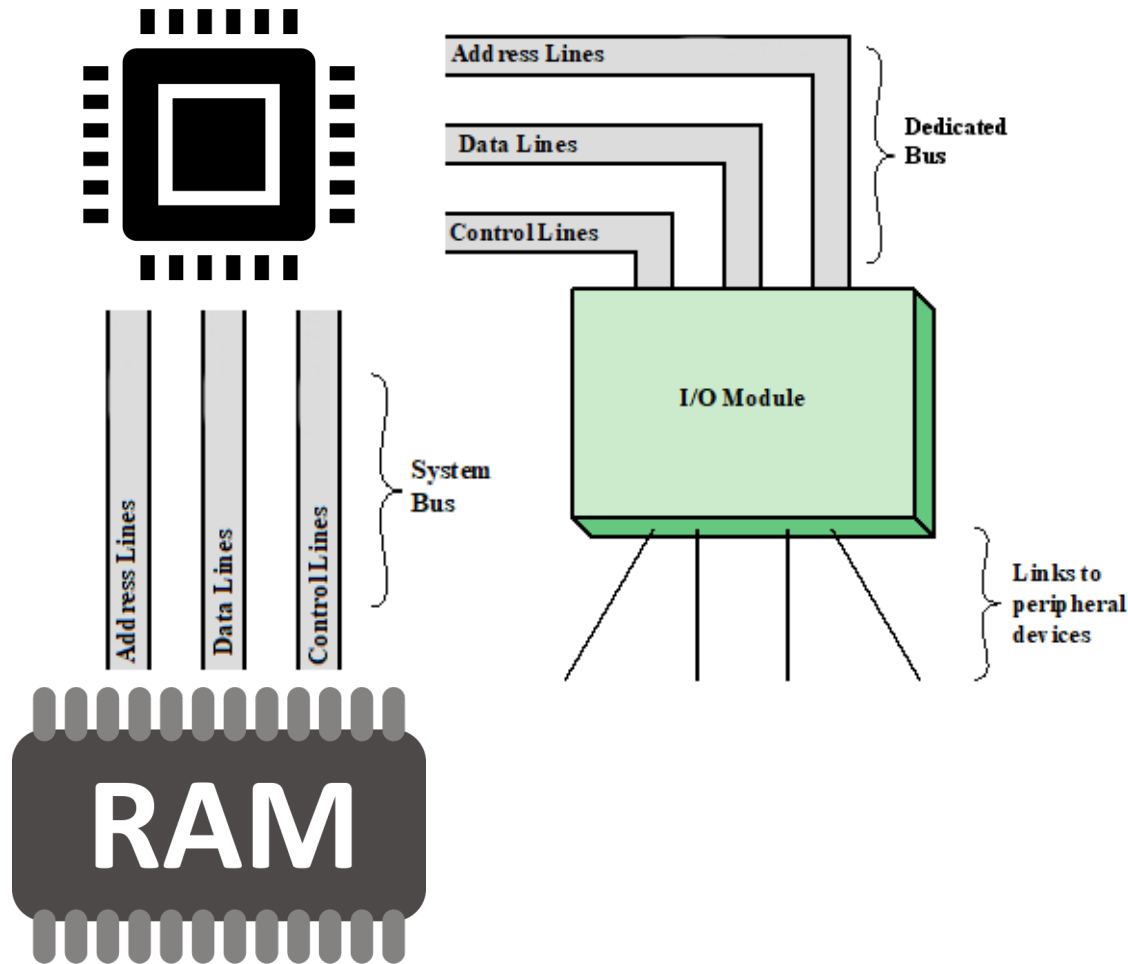


Figure 7.1 Generic Model of an I/O Module

# I/O Programmato: accesso al dispositivo

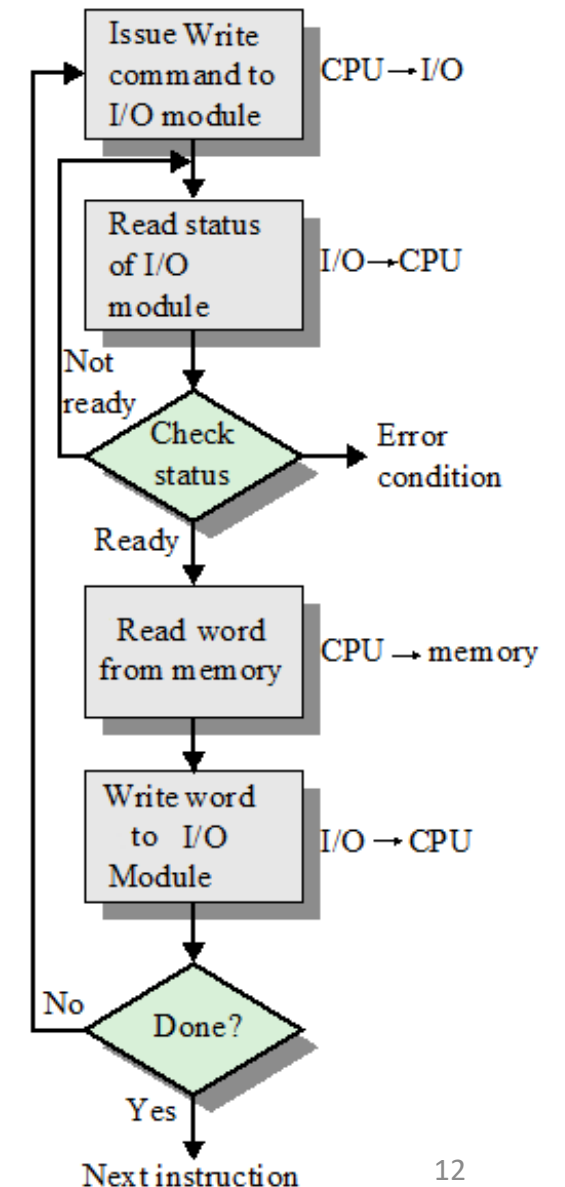


## Isolated

- I dispositivi di I/O sono collegati con linee di controllo e dati dedicate
- I dispositivi hanno uno spazio di indirizzi loro
- Il processore deve usare istruzioni dedicate per comunicare con i dispositivi

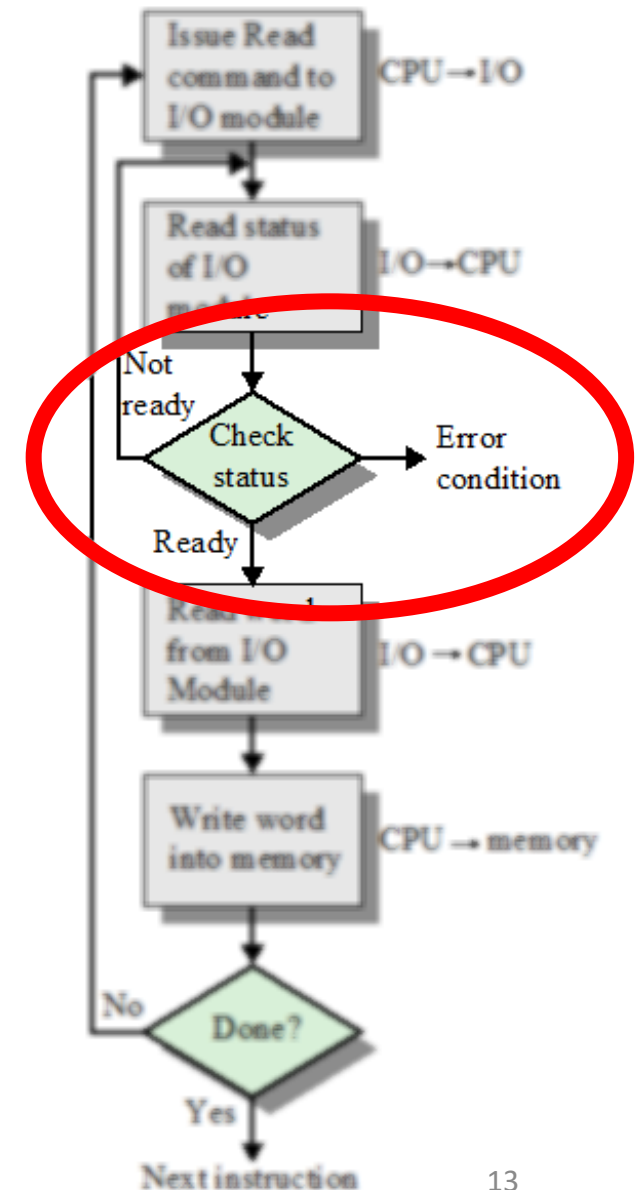
# I/O Programmato: scrittura

- La procedura è la stessa, solo che il processore invece di prelevare dei dati dal data buffer per metterli in memoria, li prenderà dalla memoria per scriverli nel data buffer
- Come per la lettura, prima di accedere al data buffer, dovrà assicurarsi che il dispositivo di I/O sia pronto



# Il problema del «busy waiting»

- Sia in lettura che in scrittura, il processore deve aspettare che il dispositivo sia pronto
- I dispositivi possono essere MOLTO più lenti del processore
- Mentre il processore aspetta, non può fare niente altro: è «busy waiting»
- Questo è un grande spreco di cicli!  
Es: un HDD a 7200 rpm ha un tempo di accesso di circa 9ms: in questo tempo, mentre il disco accede al dato, una cpu da 1GHz avrebbe potuto fare 10 milioni di operazioni !





Architettura degli Elaboratori



# I/O Programmato: pro e contro

## PRO

- Molto facile da realizzare
- Tempo di risposta molto veloce(\*): appena il dato è pronto viene letto / inviato

## CONTRO

- Mentre è in attesa che il dispositivo sia pronto, il processore non può fare niente altro!
- L'inefficienza peggiora più alta è la differenza di velocità tra il processore e il dispositivo di I/O

(\*) se il dispositivo è pronto ...

# Come possiamo migliorare?