

Comunicazione di I/O



Daniele Paolo Scarpazza
Dipartimento di Elettronica e Informazione
Politecnico di Milano

May 12th 2004

Nota sui termini

- Nelle slide che seguono presentiamo le sequenze degli eventi di alcuni meccanismi della comunicazione di I/O;
 - Non indichiamo se i segnali presentati sono attivi alti o attivi bassi;
 - Con il termine *set* indichiamo attivare il segnale, cioè alzarlo se è attivo alto, abbassarlo se è attivo basso;
 - Con il termine *reset* indichiamo disattivare il segnale, cioè abbassarlo se è attivo alto, alzarlo se è attivo basso;

Esempio: bus sincrono

- Parametri e convenzioni:

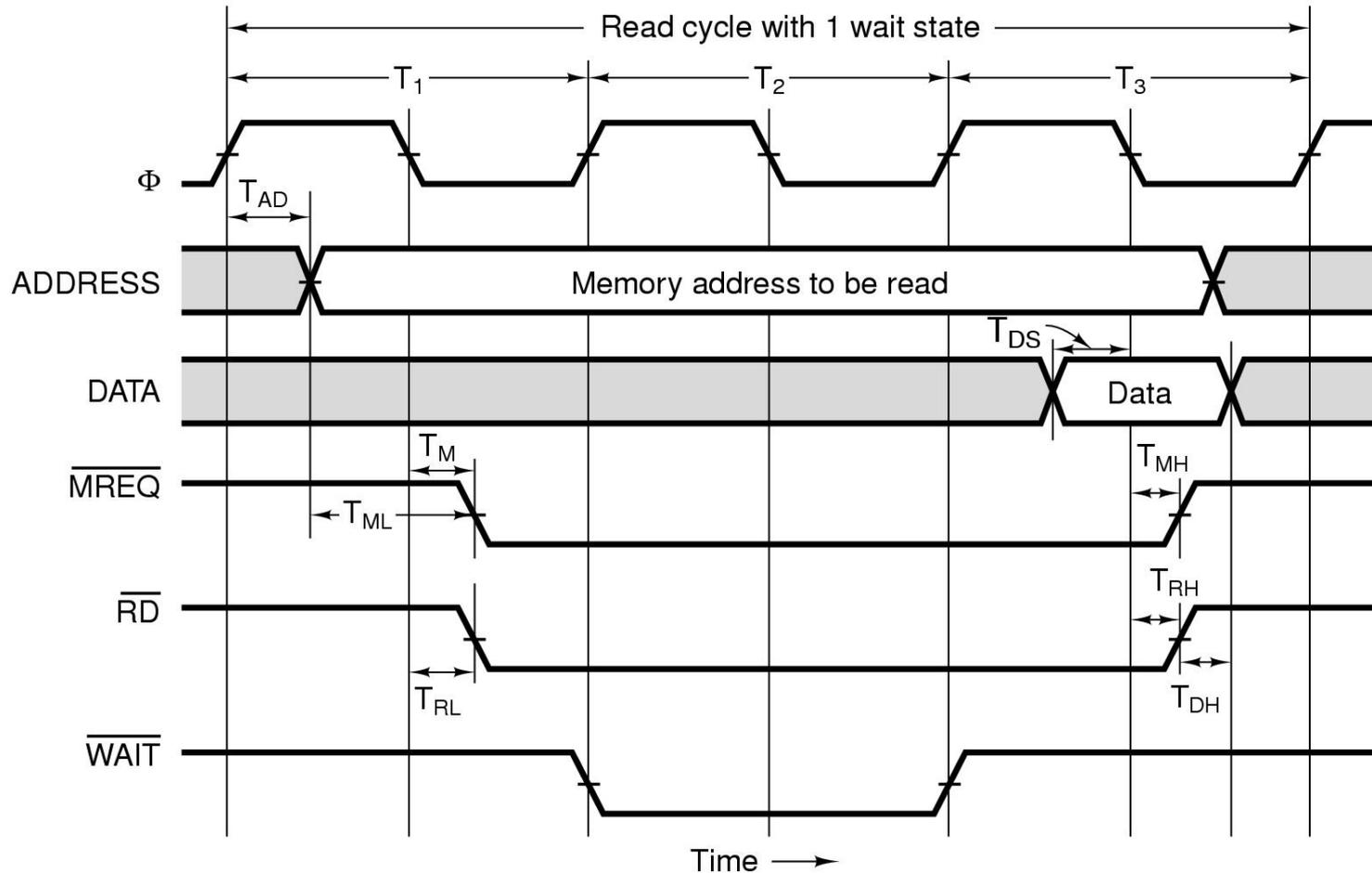
- Clock = 40 MHz ($T_c = 25$ ns)
- il clock inizia col fronte di salita

- Sequenza degli eventi:

- CPU: set ADDRESS
- CPU: set MREQ, RD
- RAM: set WAIT @ clock t ↗
(se non riesce a fornire DATA @ clock t ↘)
- RAM: set DATA @ clock ↗ ... ↘
- CPU: sample DATA @ clock ↘
- CPU: reset MREQ, RD

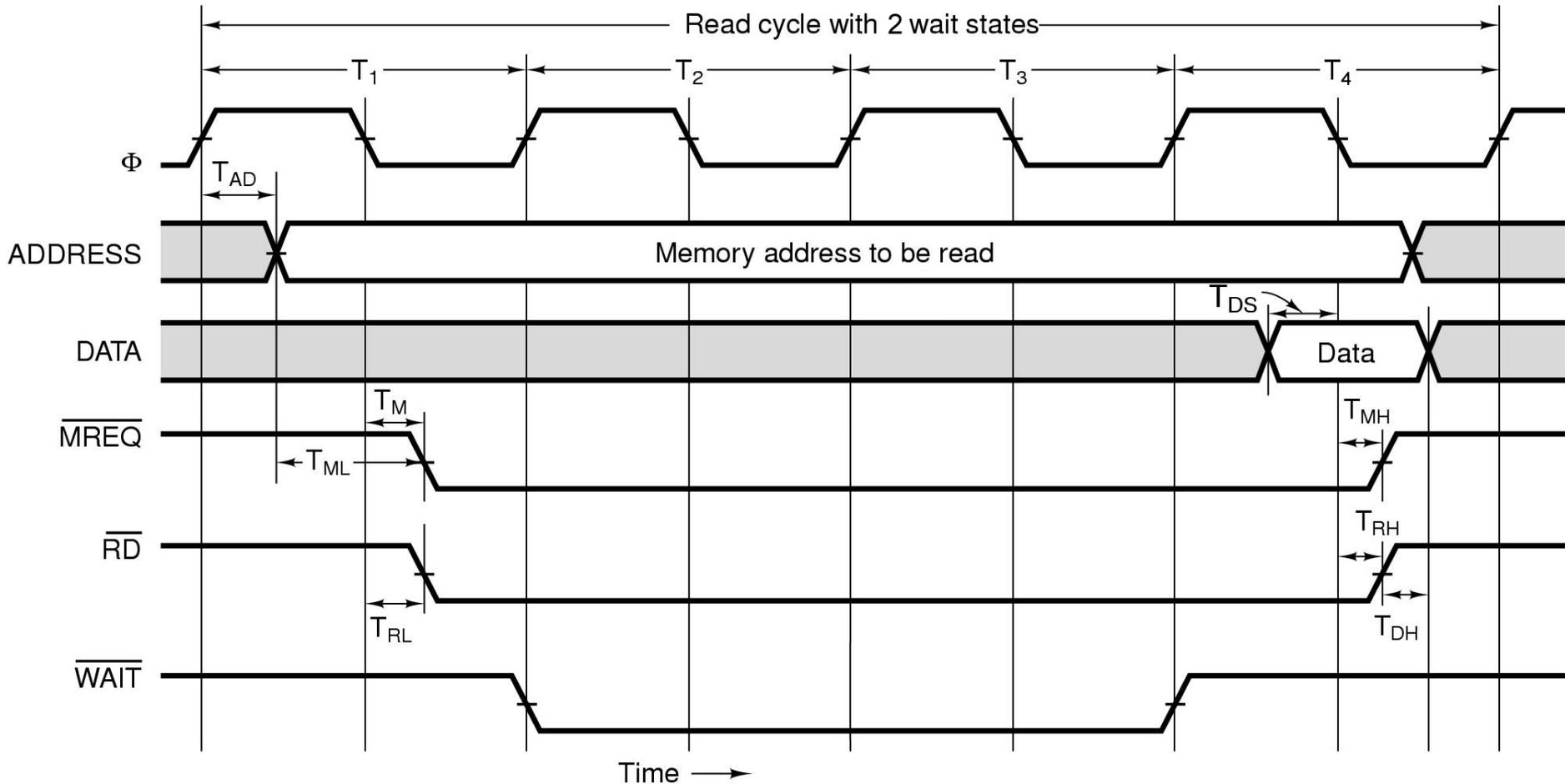
Esempio: bus sincrono

Latenza della memoria: 40ns; 1 wait state



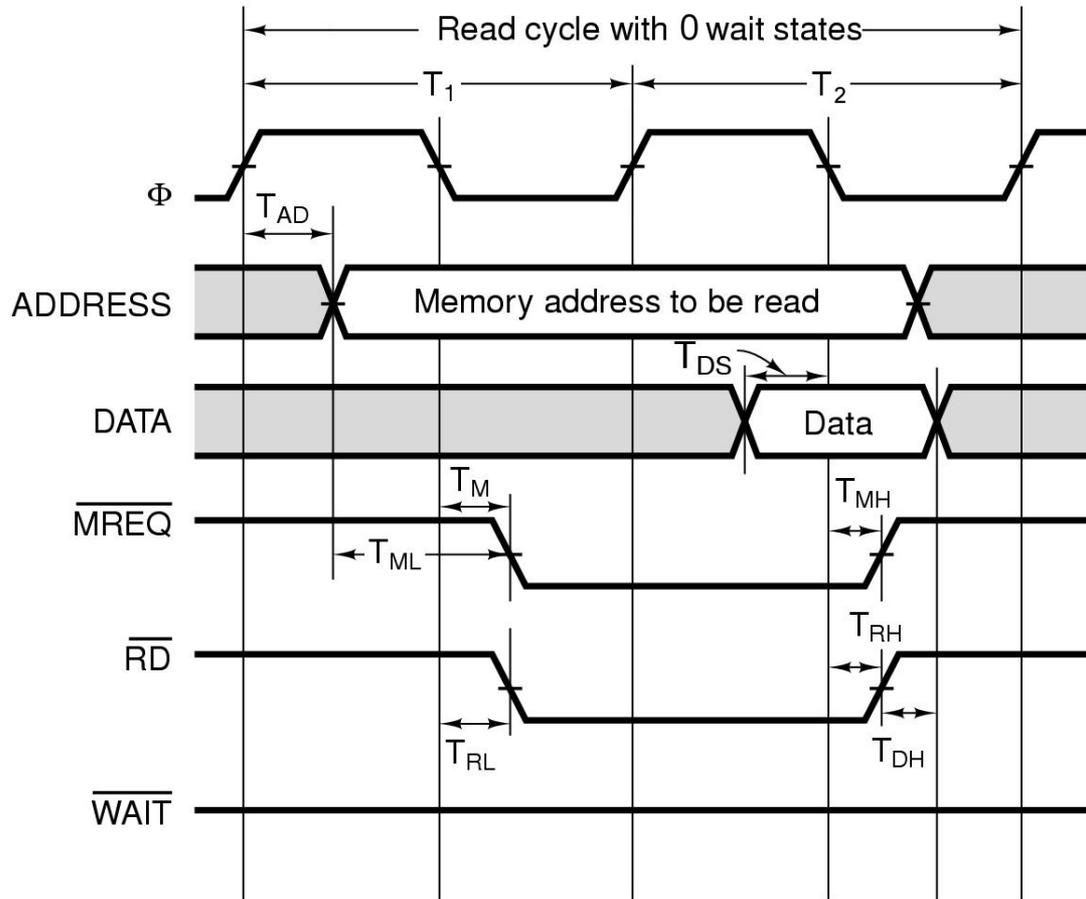
Esempio: bus sincrono

Latenza della memoria: 65ns; 2 wait states



Esempio: bus sincrono

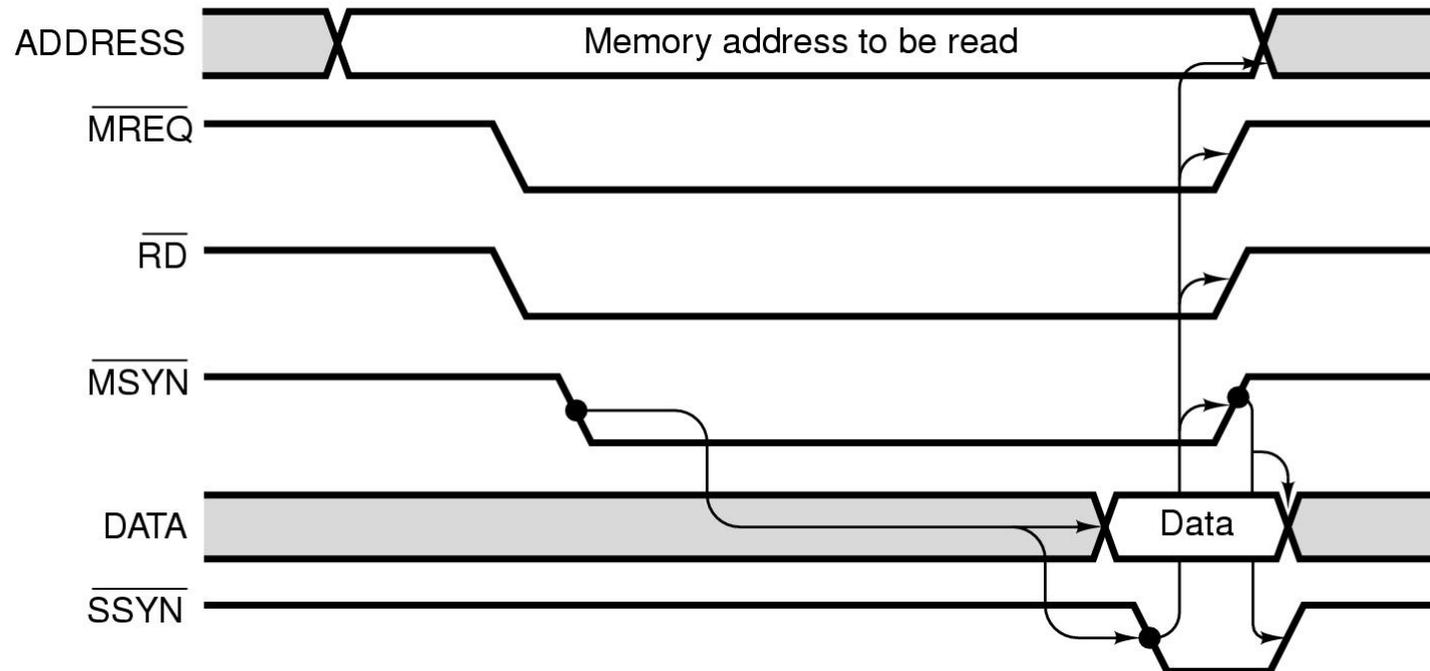
Latenza della memoria: 20ns; 0 wait states



Esempio: bus asincrono

- Non prevede l'uso del clock; le operazioni possono avvenire al massimo della velocità, senza attendere fronti di clock;
- Introduce i due segnali MSYN (Master SYNchronization) e SSYN (Slave SYNchronization);
- Sequenza degli eventi:
 - CPU: set ADDRESS
 - CPU: set MREQ, RD
 - CPU: set MSYN
 - RAM: set DATA ASAP
 - set SSYN
 - CPU: sample DATA
reset MSYN, MREQ, RD
- RAM: reset SSYN

Esempio: bus asincrono



Arbitraggio del bus con DMA

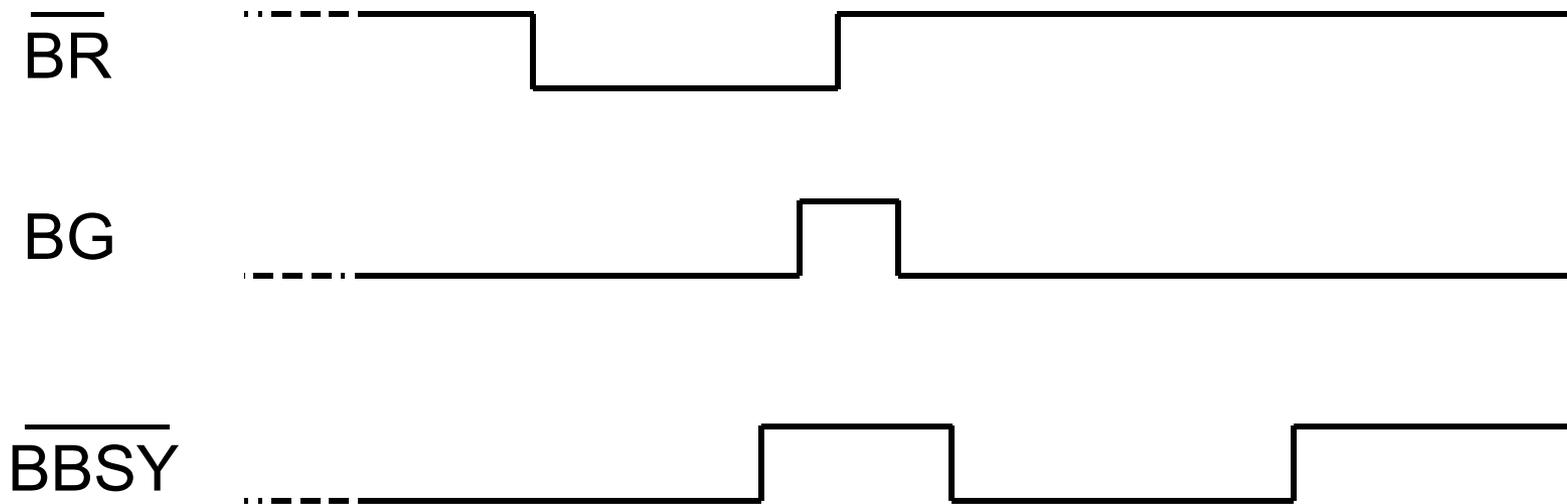
- DMA:
 - i dispositivi abilitati (*DMA controller*) comunicano con la memoria senza impegnare il processore;
 - ad un dato istante, uno solo dei DMA controller è il dispositivo abilitato a trasferire dati sul bus, cioè il *bus master*;
 - il passaggio dei poteri di bus master da un DMA controller ad un altro avviene ad opera del *bus arbiter*, solitamente la CPU;

Arbitraggio del bus con DMA

- Segnali:
 - Bus Request (BR) richiesta del bus
 - Bus Granted (BG) autorizzazione all'uso del bus
 - Bus BuSY (BBSY) bus occupato
- Sequenza degli eventi:
 - PERIF: set BR
 - ARB: attende reset BBSY
 - set BG
 - PERIF: reset BR
 - ARB: reset BG
 - PERIF: set BBSY
 - usa il bus
 - reset BBSY
- **Attenzione: numerose varianti possibili**

Arbitraggio del bus con DMA

Diagramma temporale
(in accordo con la variante descritta precedentemente)



Interruzione vettorizzata

- Ogni dispositivo ha un suo codice univoco n (*vettore di interrupt*) e si identifica ponendo tale codice sul bus;
- In risposta all'IRQ, la CPU deve leggere n dal bus e saltare alla locazione indicata nella n -esima della della tabella dei vettori di interrupt;
- Il bus potrebbe essere in uso, nel momento in cui il dispositivo deve generare l'interrupt, quindi prima di porre il proprio codice sul bus attende il segnale INTA (INTerrupt Acknowledgde)
- A seconda dell'architettura adottata può esserci un segnale di WAIT per permettere al dispositivo di indicare alla CPU che il codice non è ancora pronto sul bus;

Interruzione vettorizzata

- Sequenza degli eventi:
 - PERIF: set IRQ
 - CPU: set INTA ASAP
 - PERIF: reset IRQ
 - set WAIT se necessario
 - CPU: reset INTA
 - PERIF: set DATA
 - reset WAIT
 - CPU: attende reset WAIT
 - sample DATA

Interruzione vettorizzata

Diagramma temporale

