

## Lezione 07-10-08

### LOCK (vedi slide 125)

Indica che un'istruzione in fase di esecuzione non può essere fermata.

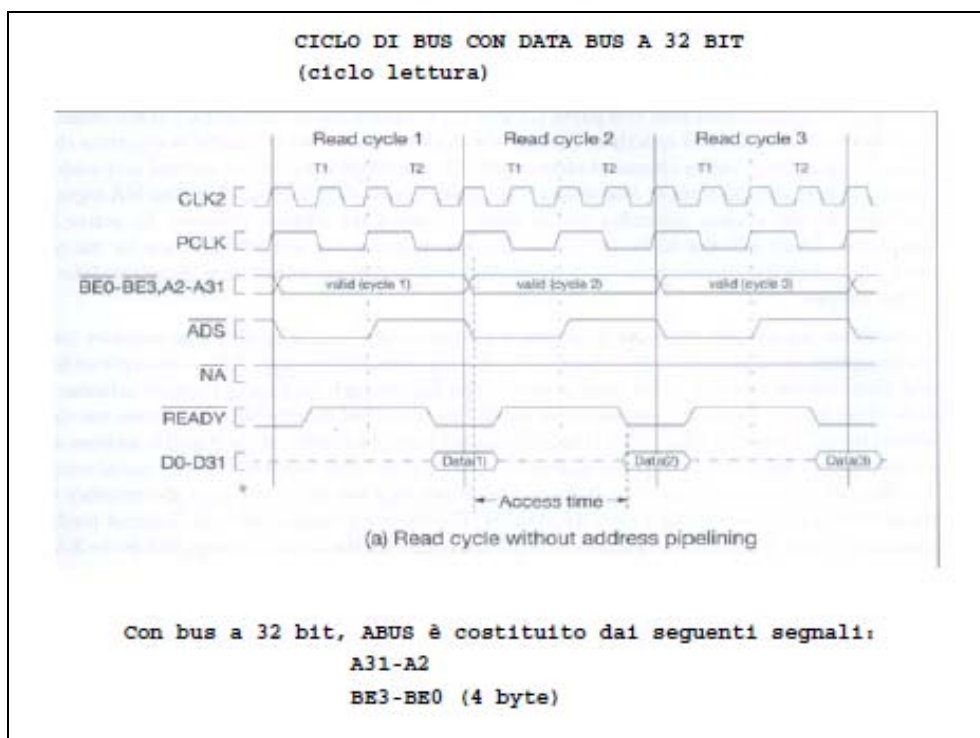
In altre parole, LOCK permette l'indivisibilità dell'esecuzione dell'istruzione. È utile ad esempio quando si usano i semafori, per realizzare l'operazione di "test and set".

### PENTIUM BUS CYCLE (vedi slide da 126 a 133)

Nei Pentium la durata di un ciclo di bus è di due periodi di clock ed esistono due tipi di cicli di bus:

- Single transfer: trasferimento di un solo dato (tutti i processori sequenziali sono fatti in questo modo)
- Burst cycle: trasferimento di 32 byte ( $4 \times 64\text{bit}$ ) effettuato per l'aggiornamento della L1 cache nei casi di cache miss.

Statisticamente un processore legge la memoria sostanzialmente per operazioni che riguardano la cache (aggiornamento della cache); per questo tipo di operazioni valgono i principi di località.



Nelle operazioni che interessano la cache si ha adiacenza di indirizzi e un tasso di hit di circa il 98% se la cache lavora bene. Senza cache la lettura di dati in memoria avverrebbe in modo sparso.

Essendo la main memory utilizzata soprattutto per copiare dati in cache si cerca di sfruttare l'adiacenza delle celle e inoltre la predizione di indirizzo (in questo modo si diminuisce notevolmente il tempo di accesso).

Nell'utilizzo di DRAM nel caso di single transfer si deve avere che il tempo di accesso alla memoria deve essere minore o uguale al tempo di CPU. Siccome il tempo di accesso è maggiore del tempo di CPU si inseriscono dei cicli di wait in modo da rispettare questa condizione. Al crescere della frequenza di clock del bus cresce anche il numero di questi cicli di wait.

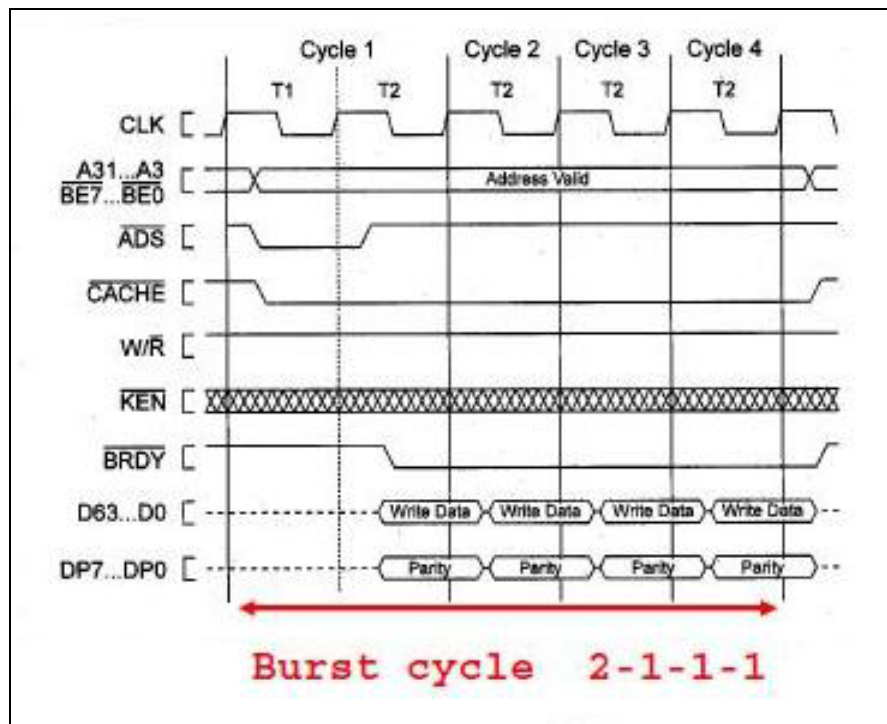
### **Burst Cycle (vedi slide da 134 a 138)**

Nei processori odierni con cache onboard, le interazioni tra processore e memoria avvengono principalmente per l'aggiornamento della cache in caso di miss oppure per scrittura di questa in memoria. Il parallelismo dei processori dipende dalla gestione efficiente della cache.

Durante l'aggiornamento della cache si deve:

- massimizzare il numero di byte associati ad una cache line per rendere più efficace il principio di località
- minimizzare il tempo di aggiornamento della cache (numero di cicli di bus per effettuare la lettura della memoria) per minimizzare il tempo di inattività della CPU.

Questi due parametri sono divergenti in quanto migliorandone uno si peggiora l'altro e viceversa. Il compromesso è stato ottenuto utilizzando 4 cicli di bus per aggiornare la cache ed aumentando il numero di segnali del DBUS in modo da inviare più byte per ogni ciclo di bus.



2-1-1-1 indica che occorrono 4 cicli di bus per trasferire i dati dalla memoria e che il primo dura 2 cicli di clock, mentre i rimanenti 3 impiegano un periodo di clock (in quanto gli indirizzi fisici di memoria interessati sono sequenziali).

# LEZIONE 7 Ottobre 2008

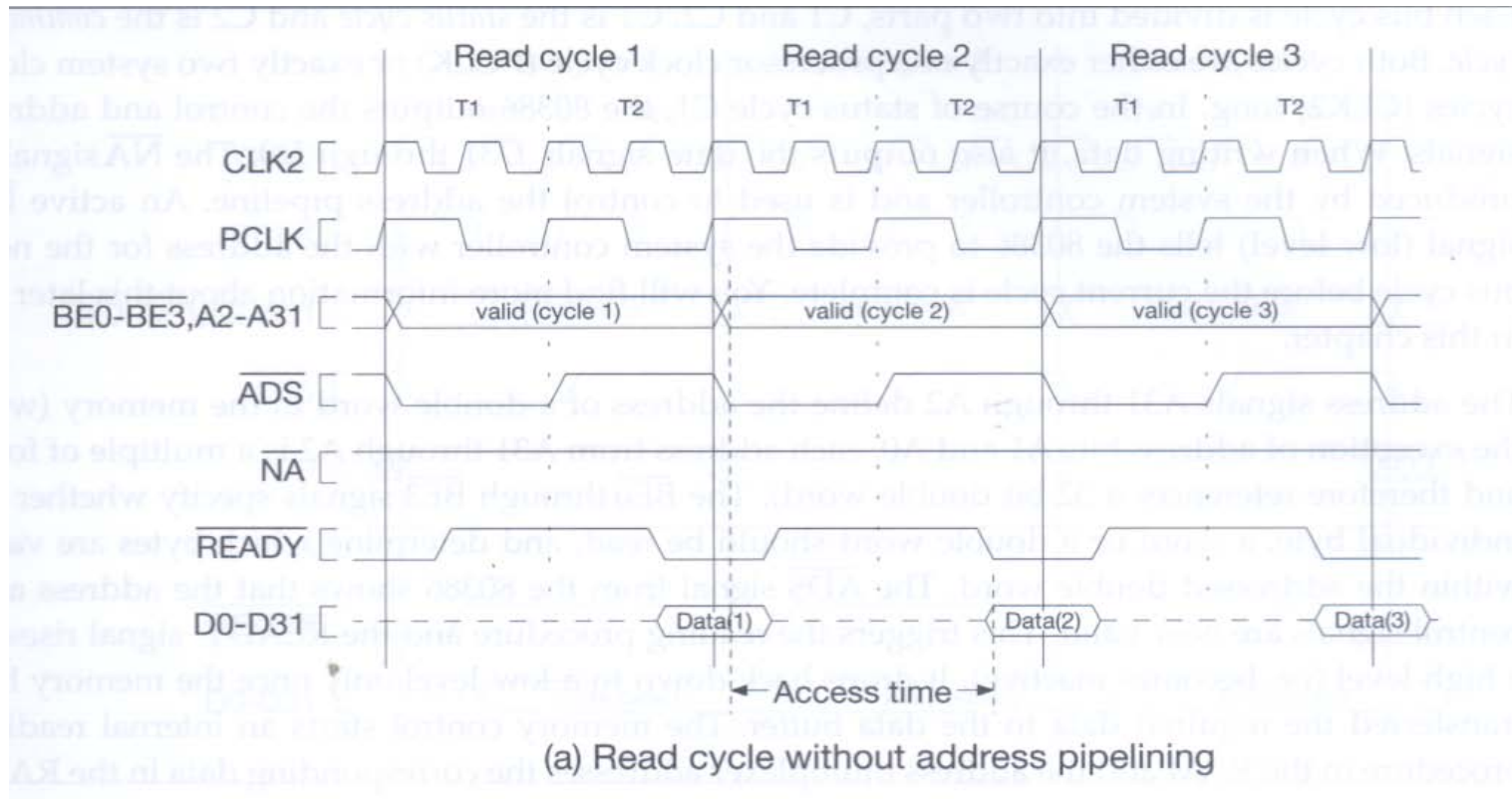
# LOCK

- **Indica che un'istruzione con il prefisso LOCK e' in corso di esecuzione e conseguentemente il bus non puo' essere utilizzato da un altro potenziale master.**

# PENTIUM BUS CYCLE

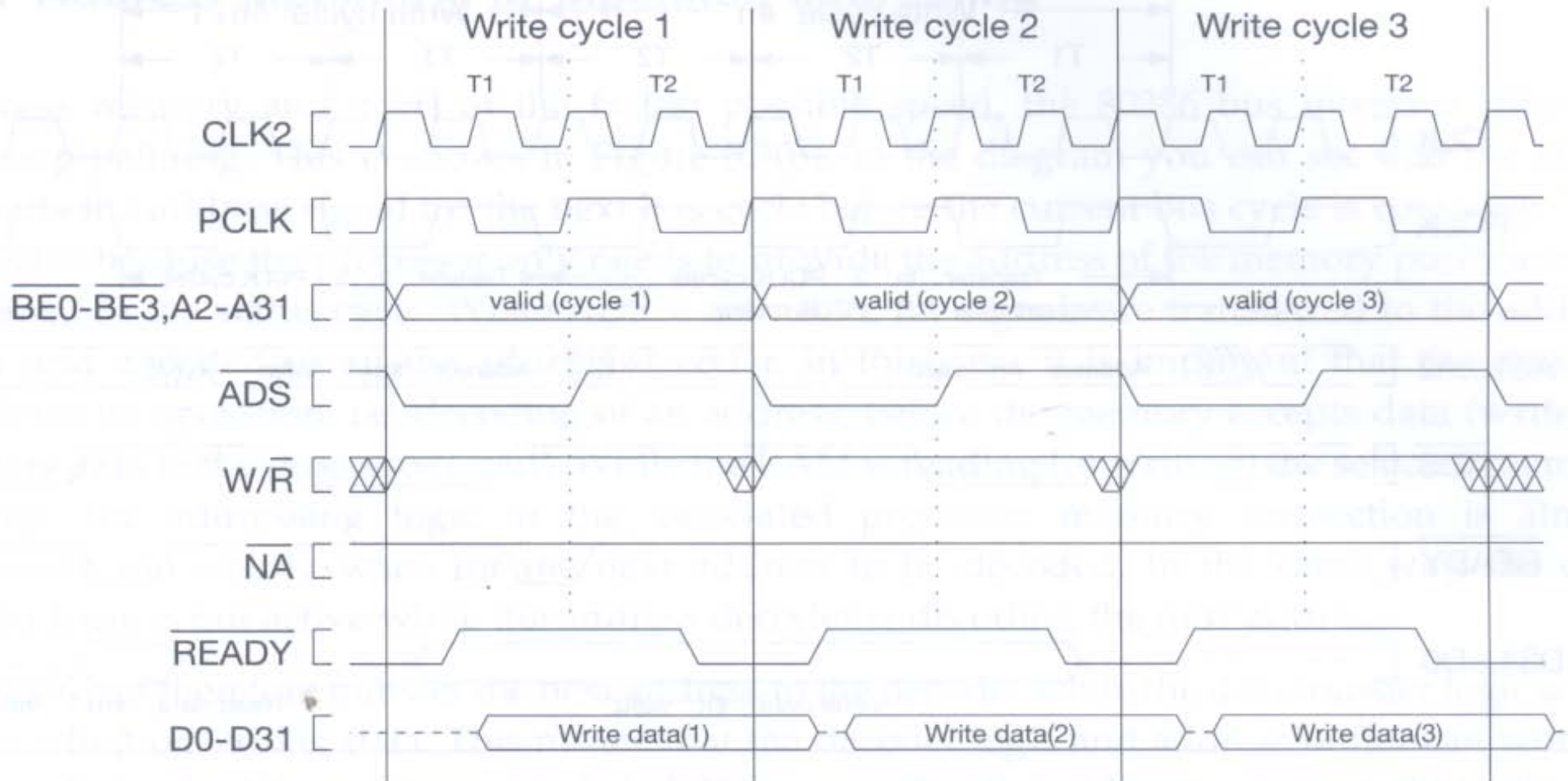
- Nel pentium esistono due tipi di cicli:
- **single transfert**: trasferimento di un solo dato
- **Burst cycle**: trasferimento di  $4 \times 64$  bit (32 byte) effettuato per aggiornamento della L1 cache nei casi di cache miss

## CICLO DI BUS CON DATA BUS A 32 BIT (ciclo lettura)



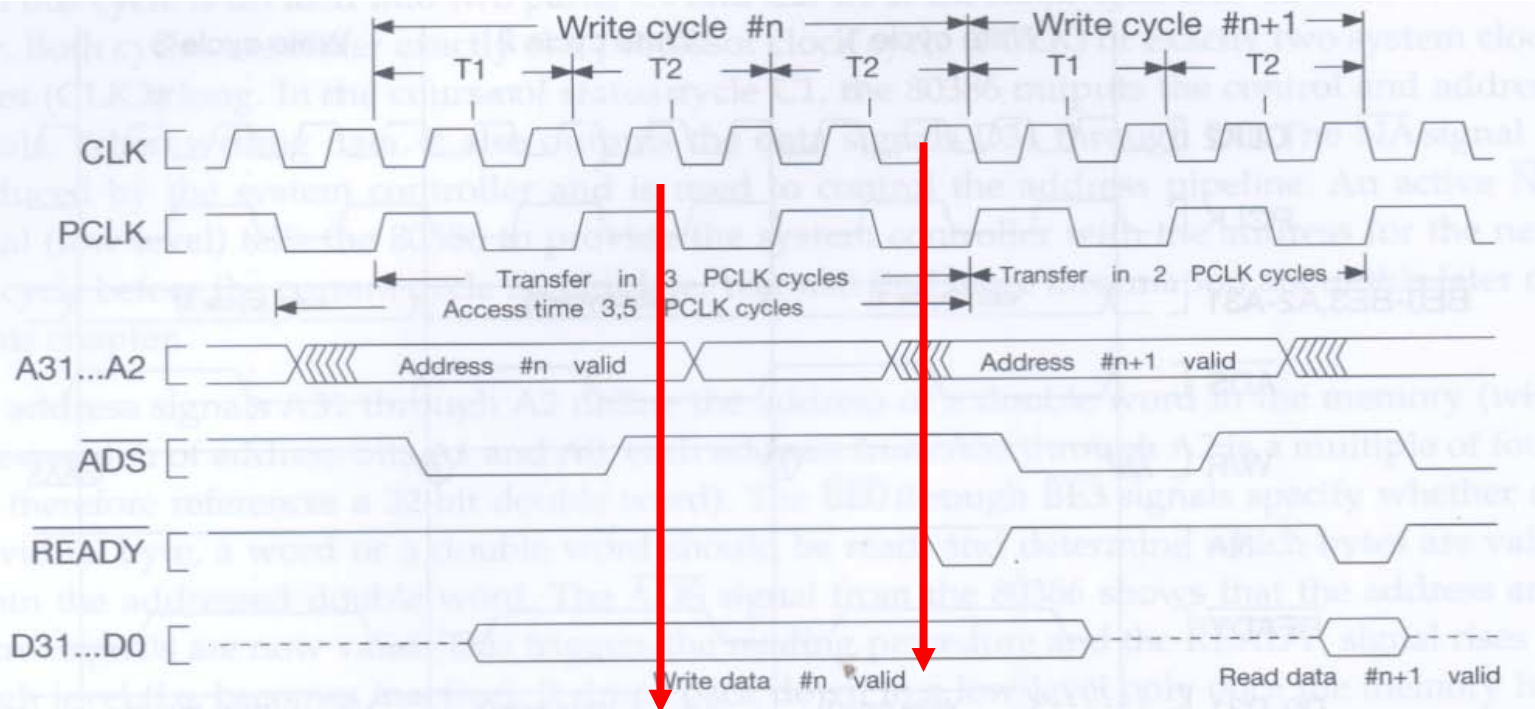
Con bus a 32 bit, ABUS è costituito dai seguenti segnali:  
**A31-A2**  
**BE3-BE0 (4 byte)**

## CICLO DI BUS CON DATA BUS A 32 BIT (ciclo scrittura)





## CICLO DI BUS CON DATA BUS A 32 BIT (ciclo scrittura con wait)

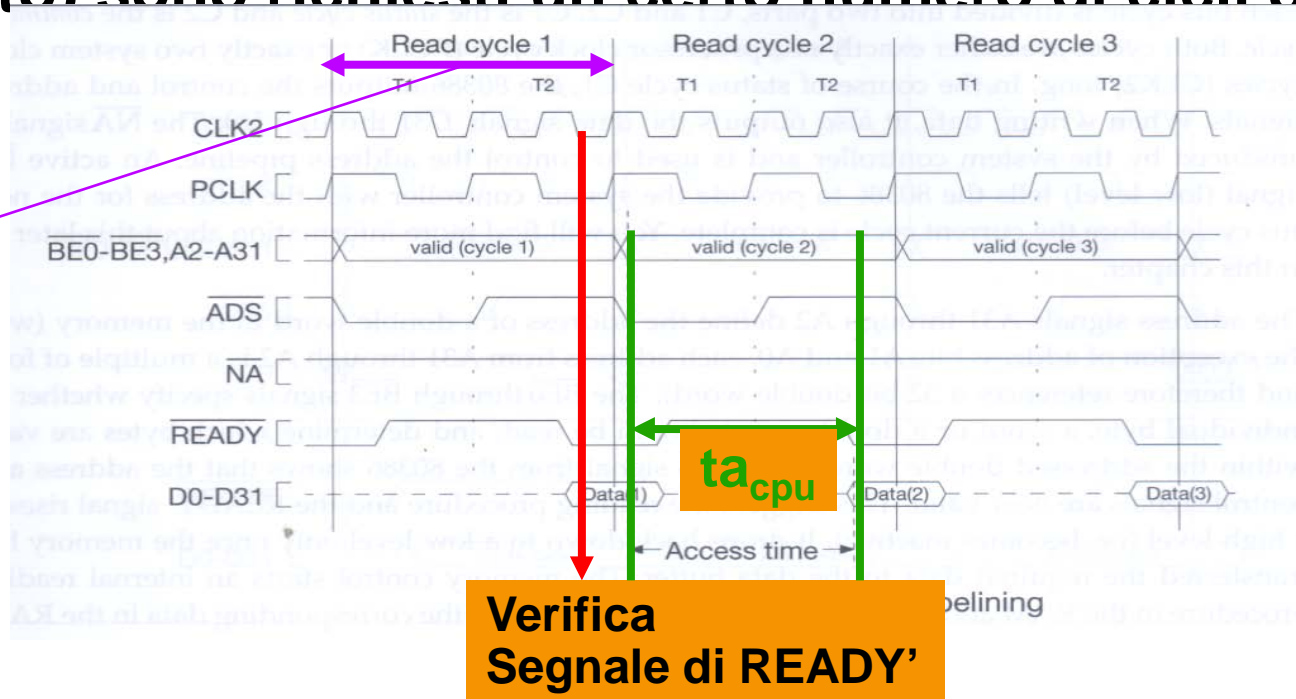


**Verifica  
Segnale di READY': nel primo caso  
La memoria NON è pronta, nel secondo SI**

# PENTIUM BUS CYCLE (esempio con Dram)

- Si supponga di avere un pentium con host bus a 300MHz che interfaccia una memoria dram con  $t_a = 80$  ns
- Nel caso di **single transfert** (trasferimento di un solo dato) sono richiesti un certo numero di cicli di wait.

Ciclo di bus  
( $2 \cdot t_{clock}$ )



# PENTIUM BUS CYCLE

## (esempio con Dram)

- Si possono fare le seguenti considerazioni:
- host bus a 30MHz significa che un ciclo di bus dura:
  - $33 \times 2 = 66 \text{ ns}$  ( $t_{\text{clock}} = 33\text{ns}$ )
- La cpu acquisisce i dati dal bus (memoria) in circa  $\frac{3}{4}$  di un ciclo di bus (si veda figura,  $t_{a_{\text{cpu}}}$ ). Nel caso in esame la memoria dovrebbe pertanto rispondere in circa  $50 \text{ ns}$  (75% di 66ns).
- Ogni ciclo di wait inserito ha durata ( $t_w$ ) pari ad un periodo di clock. Nel caso in esame  $33 \text{ ns}$ .
- In generale dovrà pertanto valere:
  - $t_a(\text{memoria}) \leq t_{a_{\text{cpu}}} + n \times t_w$

# PENTIUM BUS CYCLE

## (esempio con Dram)

- Applicando la:
  - $ta(\text{memoria}) \leq ta_{\text{cpu}} + n \cdot tw$
- al caso in esame, si ottiene:
  - $80 \leq 50 + n \cdot 33$
- Da cui si deriva che il numero di cicli di wait da inserire è pari ad  $n = 1$ .
- N.B. nel caso reale di DRAM è più corretto impiegare non il  $ta$ , ma il tempo di cyclo,  $tcycle$ , ponendosi sempre nel caso peggiore (worst case analysis). Si ricorda che  $tcycle$  è pari a circa  $2 \cdot ta$ .

# PENTIUM BUS CYCLE (esempio con Dram)

- Si consideri ora il caso:
- host bus a 100MHz significa che un ciclo di bus dura:
  - $10 \times 2 = 20 \text{ ns}$  ( $t_{\text{clock}} = 10\text{ns}$ ),  $t_{a_{\text{cpu}}} = 15\text{ns}$
- Applicando la:
  - $t_a(\text{memoria}) \leq t_{a_{\text{cpu}}} + n \times t_w$
  - al caso in esame, si ottiene:
    - $80 \leq 15 + n \times 10$
- Da cui si deriva che il numero di cicli di wait da inserire è pari ad  $n = 7$ .

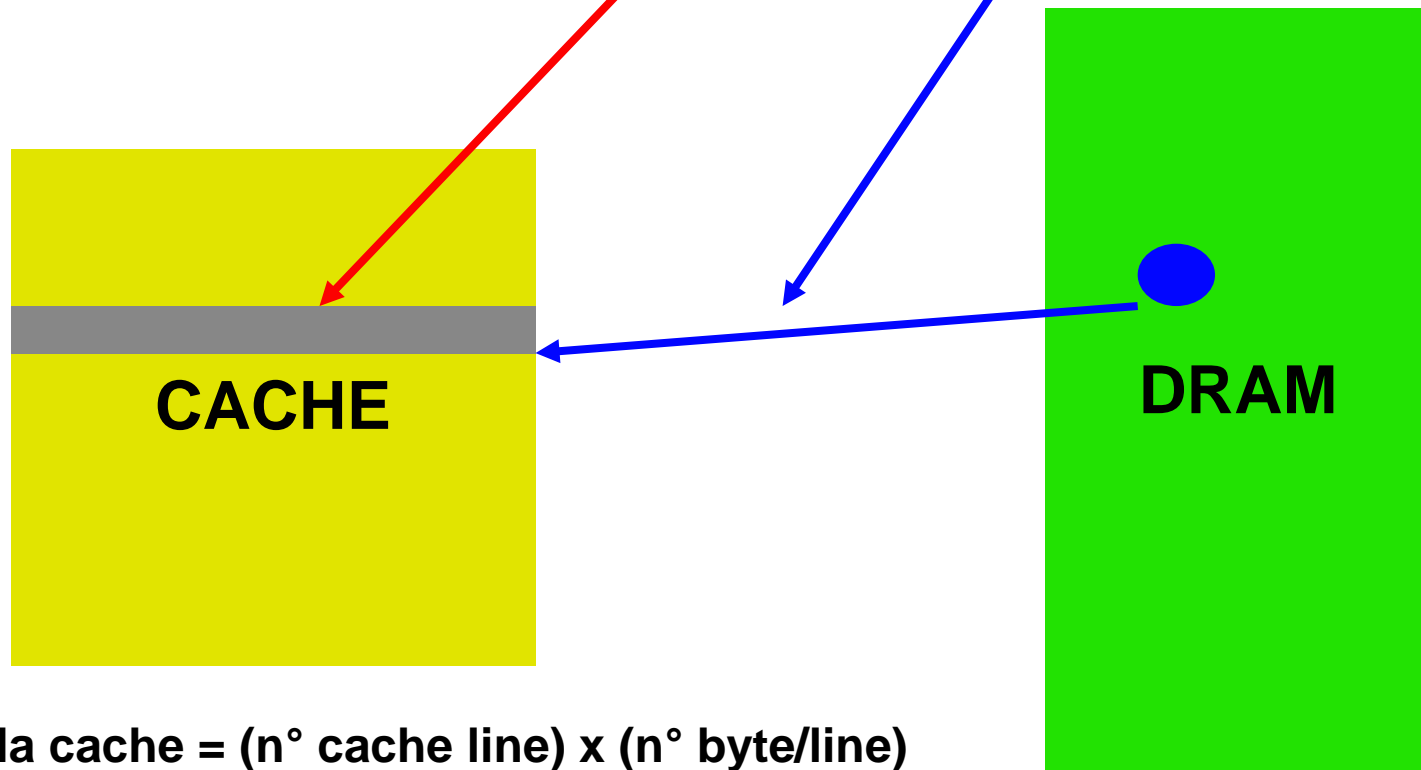
# PENTIUM BUS CYCLE

## (Burst cycle)

- Nei microprocessori con cache on –chip la maggior parte degli accessi alla memoria è dovuta alla necessità di aggiornare la cache in presenza di cache miss o di scritture della cache in memoria nel caso di write di dati sulla cache.
- Si ricorda che la cache è organizzata in cache line, ognuna delle quali riporta il contenuto di un certo numero di byte ADIACENTI della memoria
- Ne deriva che l'aggiornamento della cache consiste nel leggere nella cache stessa un dato numero di byte dalla memoria (principio di località). Pertanto tale operazione deve essere fatta nel più breve tempo possibile per evitare che la cpu resti in attiva per lungo tempo.
- Da un punto di vista progettuale due sono i parametri in gioco:
  - Il numero di byte associati ad una cache line
  - Il tempo di aggiornamento della cache, ossia il numero di bus cycle da effettuare per la lettura della memoria

# PENTIUM BUS CYCLE (cache)

- Il numero di byte associati ad una cache line
- Il tempo di aggiornamento della cache, ossia il numero di bus cycle da effettuare per la lettura della memoria



dimensioni della cache = (n° cache line) x (n° byte/line)

# PENTIUM BUS CYCLE

## (Burst cycle)

- Quanto maggiore è il numero di byte della cache line tanto più è “efficace” il principio di località, ma tanto maggiore è il tempo di attesa/inattività della cpu.
- Negli attuali processori il compromesso è ottenuto limitando a **4 i cicli di bus per aggiornare la cache**, ma incrementando il numero di segnali del DBUS in modo che ad ogni transazione si possano leggere un elevato numero di byte.
- Esempi
- DBUS 32 bit,  $(4 \text{ bus cycle} \times 4 \text{ byte}) = 16 \text{ byte cache line}$
- DBUS 128 bit,  $(4 \text{ bus cycle} \times 16 \text{ byte}) = 64 \text{ byte cache line}$

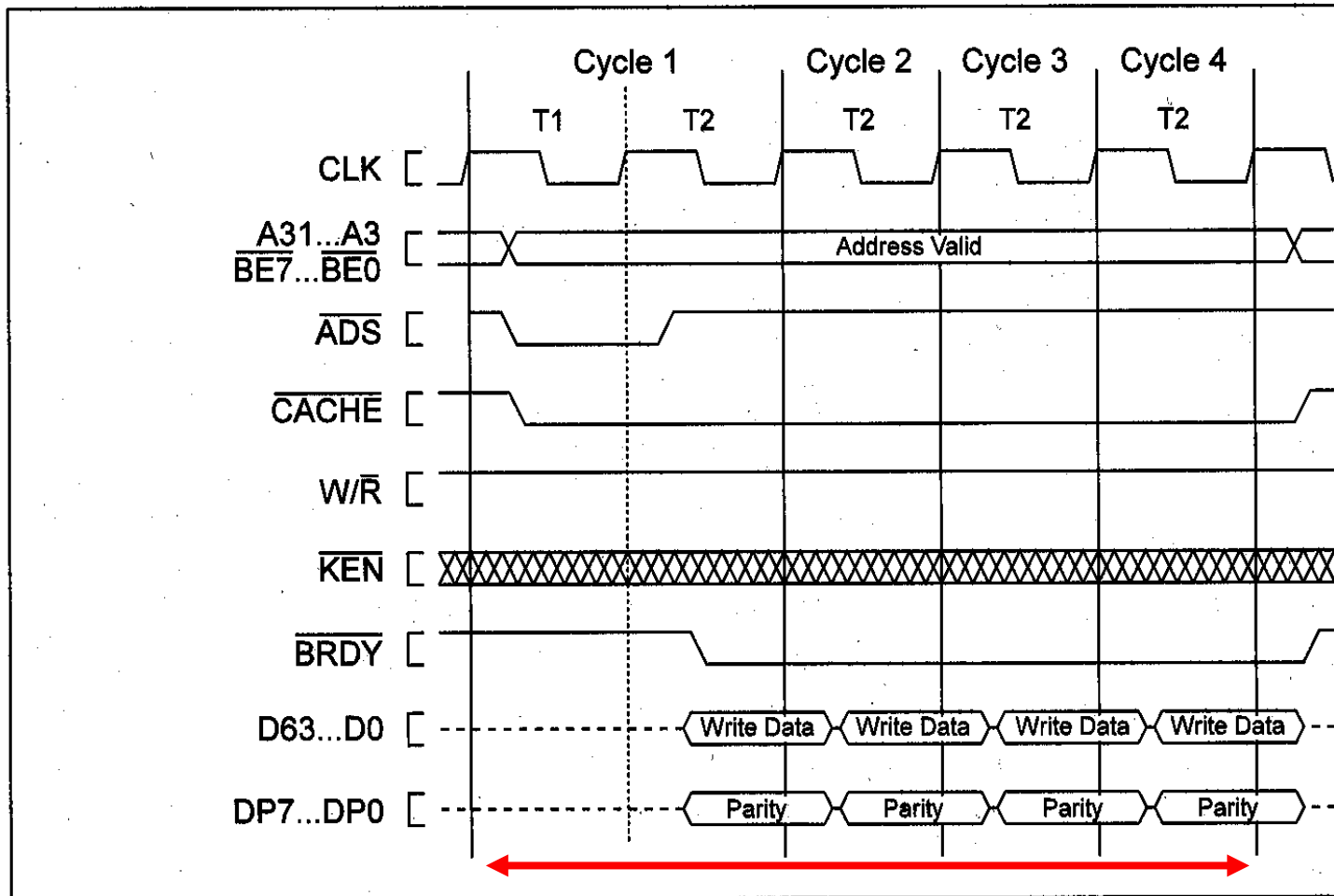


# **PENTIUM BUS CYCLE**

## **(Burst cycle)**

- Per ottimizzare i tempi di trasferimento si sfrutta il fatto che i byte sono adiacenti e quindi gli indirizzi noti a priori.
- In tal modo la BIU realizza un ciclo di burst, composto da 4 cicli di bus di cui solo il primo deve essere di 2 clock. I restanti sono di un clock solo. Si ha cioè un ciclo di burst detto “2-1-1-1”
- Ad esempio nel caso di bus da 100 MHz, si ha un ciclo burst della durata di  $20+10+10+10 = 50$ , rispetto agli 80 di 4 cicli di bus normali.
- Va da se che le memorie devono essere in grado di rispondere in tali tempi: da qui le nuove categorie di memoria dette “fast operative”, tipo le DDR DRAM.

# PENTIUM BUS CYCLE (Burst cycle)



**Burst cycle (2-1-1-1)**