

seconda parte – bus del calcolatore

Si consideri l'andamento dei segnali di *bus sincrono* quando una periferica *legge* in DMA una parola da memoria. Il DMA è realizzato con arbitraggio centralizzato e l'arbitro di bus è il processore stesso.

Ecco le fasi logiche dell'operazione: la periferica richiede il bus; il processore glielo cede; la periferica lo prende; legge la parola da memoria; e lo restituisce al processore.

Ecco le specifiche del bus:

struttura del bus

- segnali di arbitraggio: busrequest, busgrant e busbusy
- segnali di controllo: read, write e wait
- indirizzo, dato e naturalmente il clock

caratteristiche logiche e temporali del bus

- i segnali di arbitraggio e controllo sono attivi bassi
- il ciclo di clock ha periodo di 200 ns
- i tempi di propagazione sul bus variano tra 20 e 40 ns

protocollo di bus per il trasferimento di parola

- le unità funzionali (master e slave) collegate al bus acquisiscono i segnali sul fronte di salita del clock e li emettono su quello di discesa
- il trasferimento di una parola tra master e slave richiede almeno due cicli: nel primo ciclo vengono attivati segnali di controllo e indirizzo, nel secondo la parola viene trasferita
- se le tempistiche dell'unità slave impongono più di due cicli, l'unità slave proroga il primo ciclo inserendo cicli aggiuntivi tra il primo e il secondo *

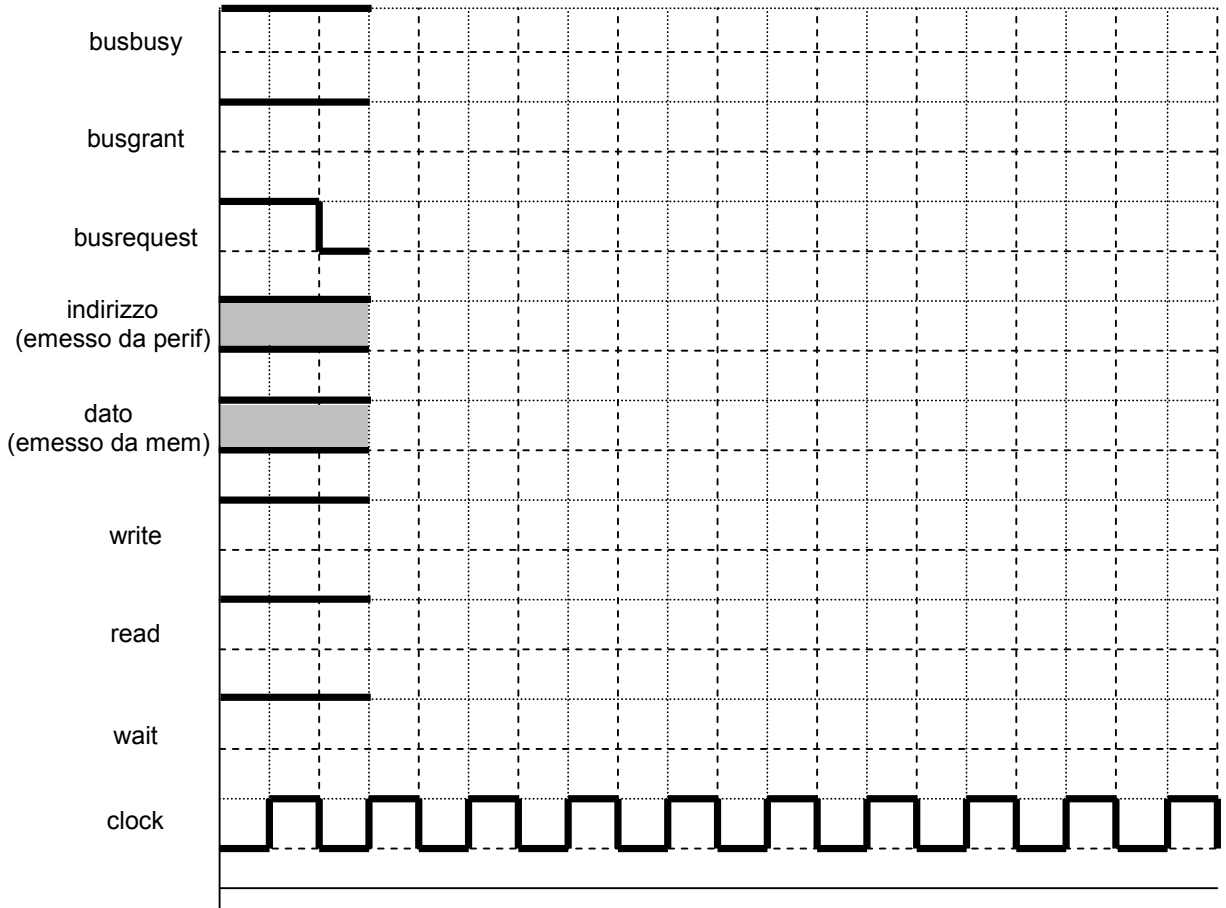
tempistiche di periferica e memoria

- la periferica prende il bus a seguito dell'attivazione di busgrant e lo restituisce quando il trasferimento termina completamente; in questo intervallo di tempo essa è master
- la memoria si attiva non appena acquisisce i segnali di controllo e riconosce l'indirizzo; dopo di ciò necessita di non meno di 200 ns per essere pronta a trasferire la parola

Si chiede di completare il diagramma temporale mostrato nella figura seguente.

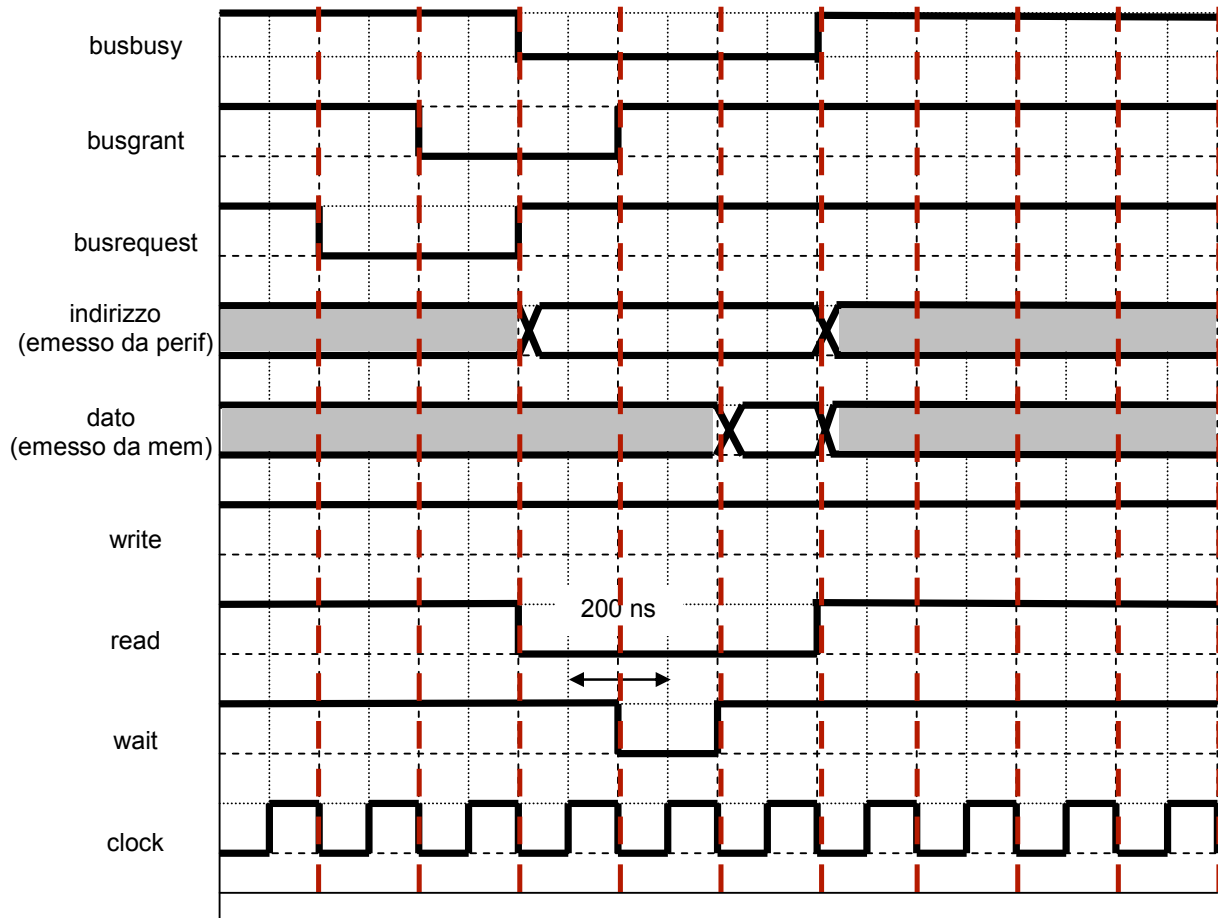
*** Vuol dire che ho delle wait prima del trasferimento della parola. Si nota infatti che il segnale di wait viene attivato proprio in corrispondenza del secondo ciclo durante il trasferimento della parola dalla memoria alla CPU.**

da completare



soluzione

Occorre esattamente un ciclo di proroga.



seconda parte – bus di calcolatore

Si completi il diagramma temporale mostrato sotto, illustrando l'andamento dei segnali del bus sincrono di un calcolatore quando il processore preleva un'istruzione dalla cache istruzioni e l'istruzione va in esecuzione scrivendo un dato nella cache dati. Sia l'istruzione prelevata sia il dato scritto ingombrano **una parola**.

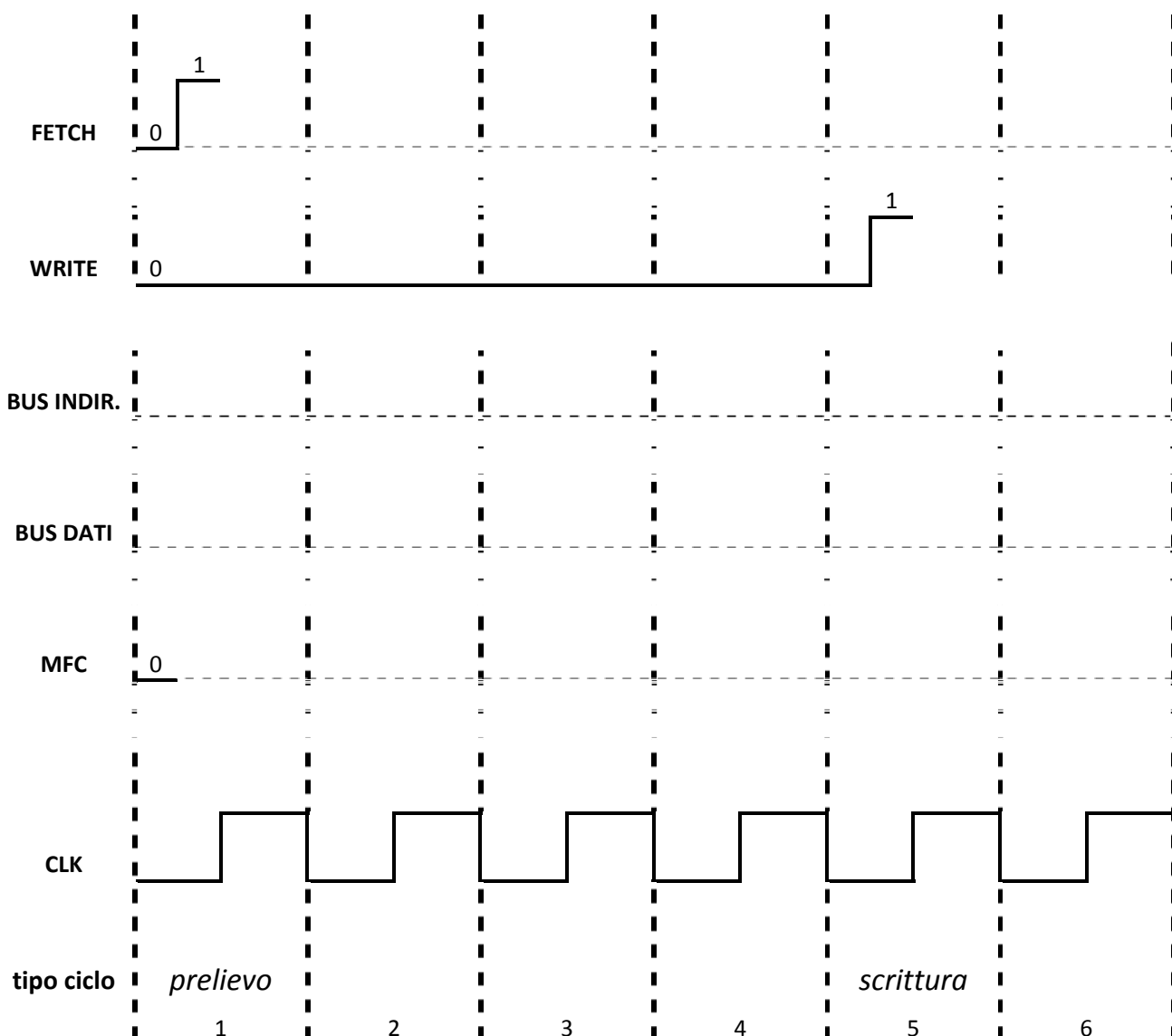
Il bus di controllo ha i tre segnali seguenti: **FETCH**, che contrassegna un indirizzo valido per prelievo d'istruzione; **WRITE**, che contrassegna un indirizzo valido per scrittura di dato; e **MFC** (Memory Function Completion), che indica il completamento di prelievo o scrittura, secondo il caso. Tutti e tre sono **attivi alti**.

Valgono le specifiche seguenti, che precisano la temporizzazione del protocollo di bus sincrono in esame:

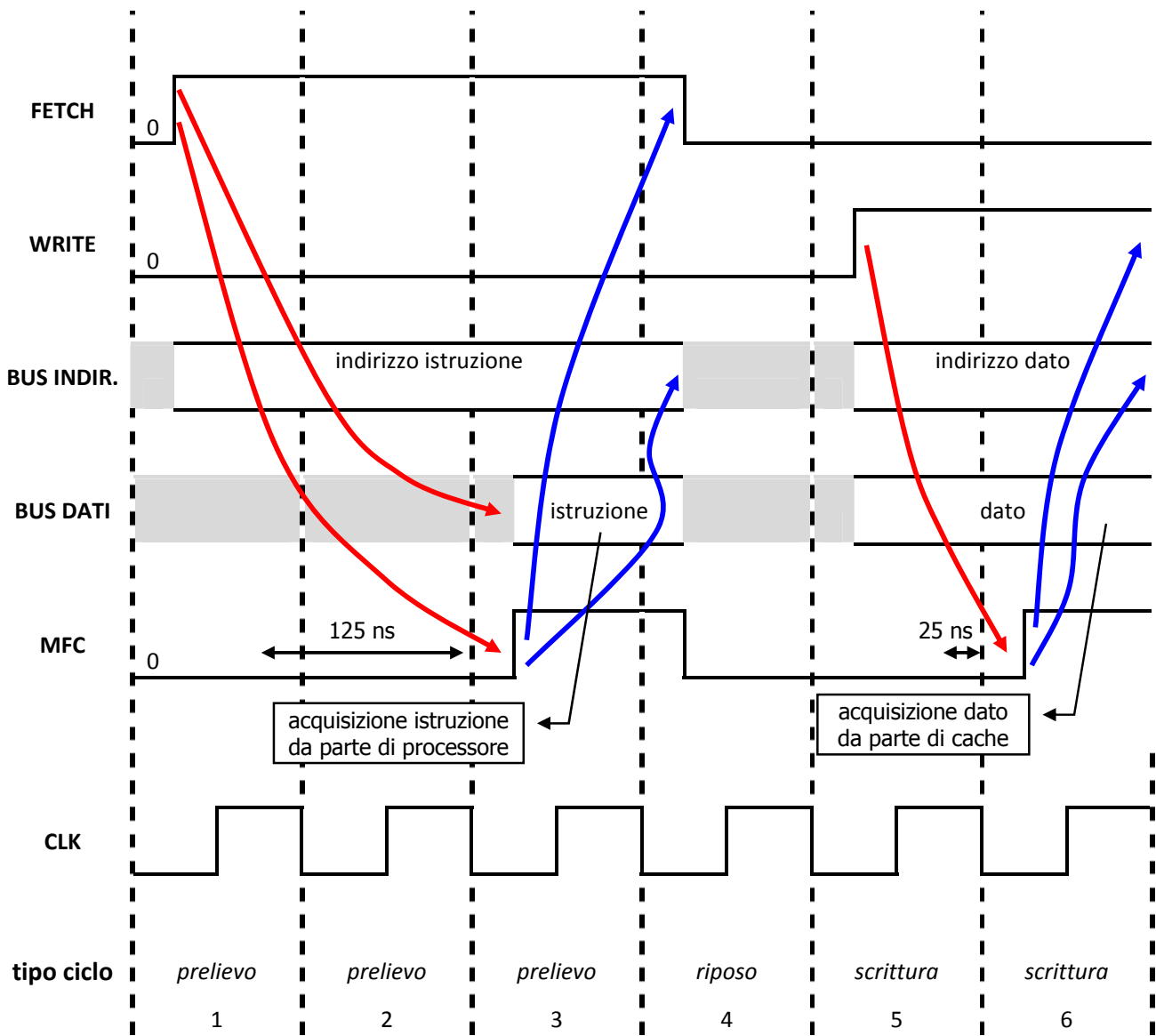
- il fronte di **discesa** del clock demarca il ciclo e il periodo di clock vale **100 ns**
- un segnale è acquisibile esattamente **25 ns** dopo il fronte di **salita** del clock
- un segnale è modificabile esattamente **25 ns** dopo il fronte di **discesa** del clock
- a partire da quando si accorge che è richiesto un trasferimento di parola:
 - la cache istruzioni necessita di almeno **125 ns** prima di attivare MFC ed emettere la parola
 - la cache dati necessita di almeno **25 ns** prima di attivare MFC e acquisire la parola
- il bus indirizzi e i segnali **FETCH** e **WRITE** vanno a riposo, rispettando le specifiche date sopra, dopo che la cache ha attivato il segnale **MFC** e il processore ne ha potuta rilevare l'attivazione

Nel diagramma temporale sono indicati solo i cicli di clock e quando il processore richiede prelievo o scrittura; il resto del diagramma va completato (il numero di cicli non è significativo).

Sul diagramma è preferibile indicare le relazioni di **causa-effetto**, dove ne esistono, tra le transizioni dei vari segnali, e indicare anche il **tipo di ciclo** (prelievo, scrittura o riposo).



soluzione



I cicli dove il processore richiede prelievo e scrittura sono: ciclo 1 e 5, rispettivamente, 25 ns dopo l'inizio del ciclo. La cache istruzioni rileva il prelievo (segnale FETCH e indirizzo) nel ciclo 1, 25 ns dopo il fronte di salita del clock; essa necessita di almeno 125 ns per decodificare l'indirizzo e disporsi a emettere la parola, dunque attiva il segnale MFC ed emette l'istruzione 25 ns dopo l'inizio del ciclo 3; il processore rileva il completamento del prelievo e acquisisce l'istruzione (segnale MFC e istruzione) nel ciclo 3, 25 ns dopo il fronte di salita del clock; pertanto il prelievo termina nel ciclo 3; non essendoci altri segnali attivi nel bus di controllo, il ciclo 4 è di riposo per il bus; la cache dati rileva la scrittura (segnale WRITE e indirizzo) nel ciclo 5, 25 ns dopo il fronte di salita del clock; essa necessita di almeno 25 ns per decodificare l'indirizzo e disporsi ad acquisire la parola, dunque attiva il segnale MFC 25 ns dopo l'inizio del ciclo 6 e acquisisce il dato 25 ns dopo il fronte di salita del clock nel ciclo 6; pertanto la scrittura termina nel ciclo 6; il ciclo 7 (qui non mostrato), in assenza di segnali attivi nel bus di controllo, sarebbe di riposo per il bus. Si badi che, nei tempi minimi di risposta precisati, la cache è comunque vincolata ad acquisire e modificare segnali (controllo, indirizzo e dato) in conformità alle specifiche di bus, valide per tutte le unità funzionali collegate al bus.

Qui processore e cache hanno ruolo di master e slave, rispettivamente. Le relazioni di causa-effetto sono di due tipi: in rosso, l'inizio di un'operazione da parte del master causa il completamento da parte dello slave, ossia l'attivazione del segnale MFC nonché l'emissione dell'istruzione o l'acquisizione dal dato secondo ci sia prelievo o scrittura; in blu, il completamento da parte dello slave (per entrambe le operazioni) causa la rimozione dell'indirizzo e la disattivazione dei segnali di FETCH o WRITE da parte del master.

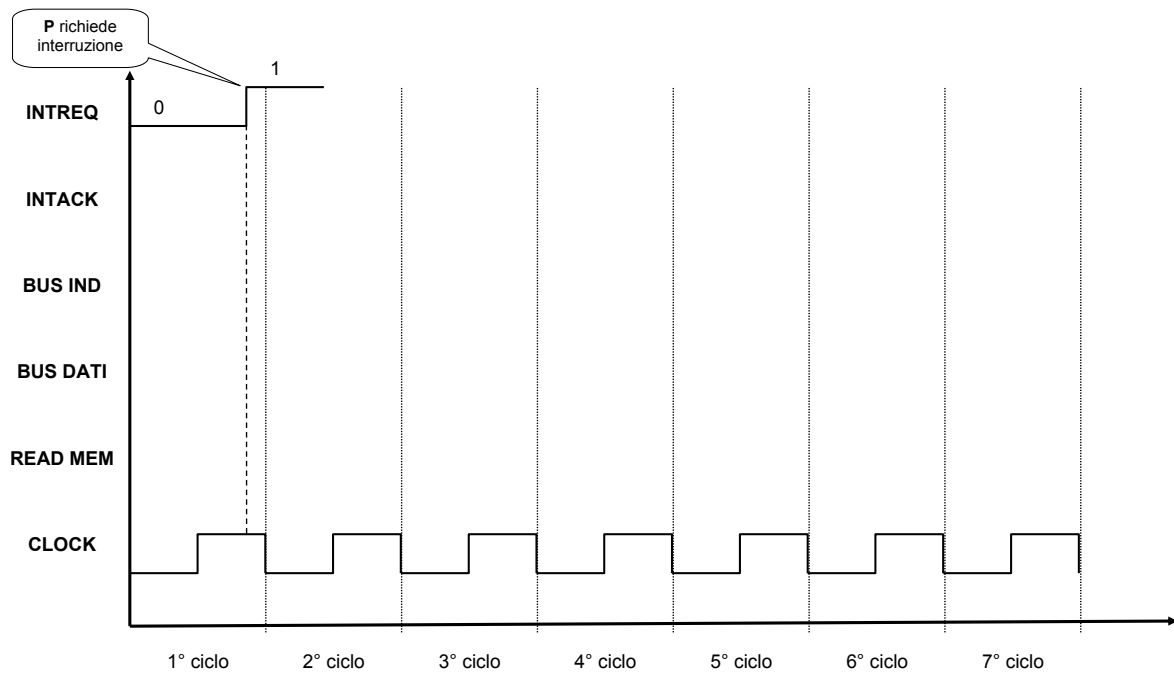
Ecco una possibile variante del protocollo di bus: disattivare i comandi FETCH e WRITE dopo un solo ciclo, giacché per le tempistiche specificate essi sono acquisiti nello stesso ciclo dove vengono attivati; tuttavia questa variante dipende strettamente dai tempi qui specificati, mentre attendere il completamento per disattivare i comandi è un protocollo di bus indipendente da tali tempi.

seconda parte – bus del calcolatore

Si chiede di completare il diagramma temporale mostrato nella figura seguente, illustrando l'andamento dei segnali del **BUS sincrono** di un processore in una situazione dove una **periferica P** effettua una richiesta d'interruzione (interrupt).

Il **bus di controllo** contiene i due segnali INTREQ e INTACK per la gestione del meccanismo d'interruzione (le interruzioni generate da **P** sono di tipo vettorizzato) e il segnale READ_MEM emesso dal processore, che contrassegna un indirizzo valido per un'operazione di lettura da memoria.

Nel diagramma temporale sono indicati solo i **cicli di clock** e l'**istante di tempo** in cui la periferica **P** invia al processore la richiesta d'interruzione; il resto del diagramma **va completato**.

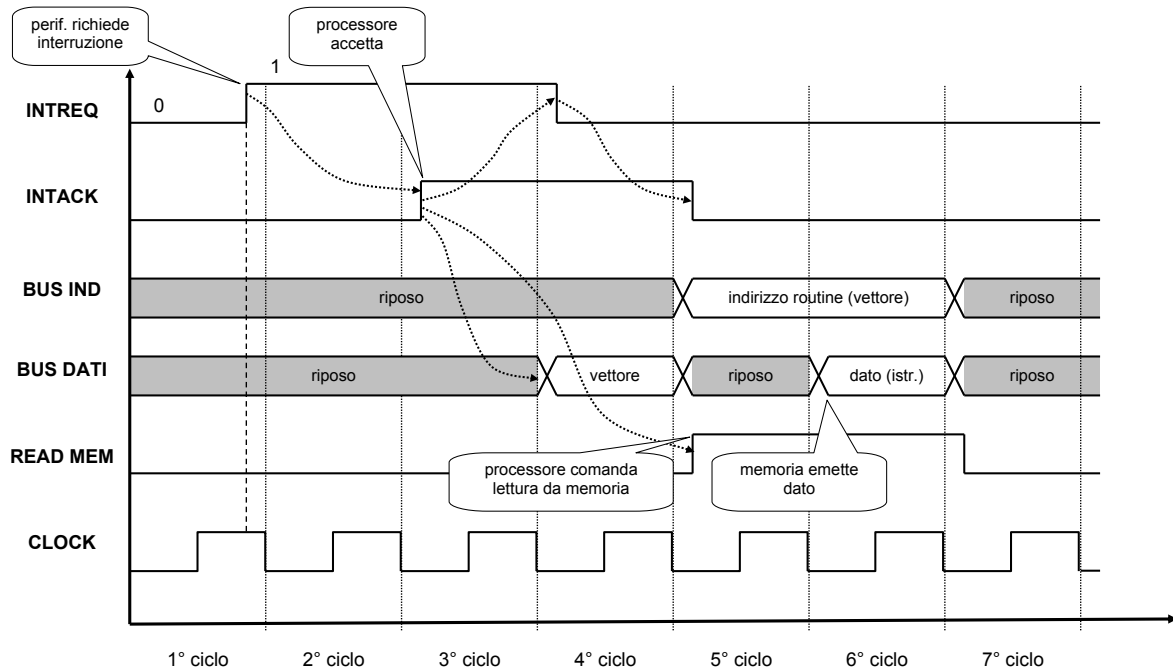


I segnali di controllo INTREQ, INTACK e READ_MEM sono di tipo **attivo alto**. I segnali INTREQ e INTACK osservano il protocollo di **handshake**: vengono attivati e disattivati l'uno in risposta all'altro, ma tenendo conto della presenza del clock come si dice più avanti. Si considerino le seguenti specifiche ulteriori:

- Il ciclo di clock è demarcato dal fronte di discesa del clock e il periodo di clock è di **100 ns**. Tutti i segnali vengono acquisiti (letti) entro **10 ns** (nanosecondi) a partire dal fronte di salita del clock e vengono emessi (scritti) entro **10 ns** a partire dal fronte di discesa del clock.
- A partire da quando si accorge che l'interruzione richiesta è stata accettata da parte del processore, la periferica **P** abbisogna di **50 ns** prima di potere emettere sul bus dati il suo vettore d'interruzione, e poi lo mantiene stabile sul bus dati per **100 ns**.
- Il **vettore d'interruzione** rappresenta direttamente l'indirizzo dell'istruzione macchina iniziale della routine di servizio d'interruzione. Nel ciclo successivo alla lettura del vettore, il processore emette sul bus indirizzi l'indirizzo della prima istruzione della routine di servizio d'interruzione, attiva il segnale di controllo READ_MEM e si dispone a leggere da memoria la prima istruzione della routine.
- A partire da quando si accorge che è in corso un'operazione di lettura, la memoria abbisogna di **50 ns** prima di potere emettere sul bus dati la parola da leggere e poi la mantiene stabile per **100 ns**.
- I segnali sul bus indirizzi vengono portati a **riposo** e il segnale di controllo READ_MEM viene **disattivato** non appena i segnali sul bus dati non sono più stabili.

Completando il diagramma temporale è preferibile indicare anche le relazioni di **causa-effetto** tra le transizioni dei vari segnali, laddove esse esistono, come elemento di **maggior chiarezza**.

SOLUZIONE BEN COMMENTATA



Osservazioni: il segnale INTACK viene attivato dalla periferica troppo tardi per essere acquisito dal processore nel ciclo corrente e l'acquisizione è differita al fronte di salita del secondo ciclo; dati i ritardi indicati nelle specifiche, quando un segnale viene emesso all'inizio del ciclo, viene acquisito sul fronte di salita del ciclo stesso e l'eventuale reazione del ricevente si manifesta all'inizio del ciclo successivo.

Andamento: il processore acquisisce INTREQ nel secondo ciclo ed emette INTACK nel terzo ciclo; come reazione INTREQ viene disattivato dalla periferica nel quarto ciclo e INTACK viene disattivato dal processore nel quinto ciclo; questo completa lo handshake tra periferica e processore; la periferica emette il vettore nel quarto ciclo (come reazione a INTACK nel terzo ciclo) e il processore lo acquisisce nel ciclo stesso; il processore attiva READ_MEM ed emette l'indirizzo (ossia il vettore) nel quinto ciclo e la memoria li acquisisce nel ciclo stesso; la memoria emette il dato (istruzione iniziale della routine d'interruzione) nel sesto ciclo e il processore lo acquisisce nel ciclo stesso; nel settimo ciclo i segnali vanno a riposo e l'operazione descritta è conclusa; le frecce tratteggiate indicano i rapporti di causa-effetto e qua e là ci sono commenti esplicativi.