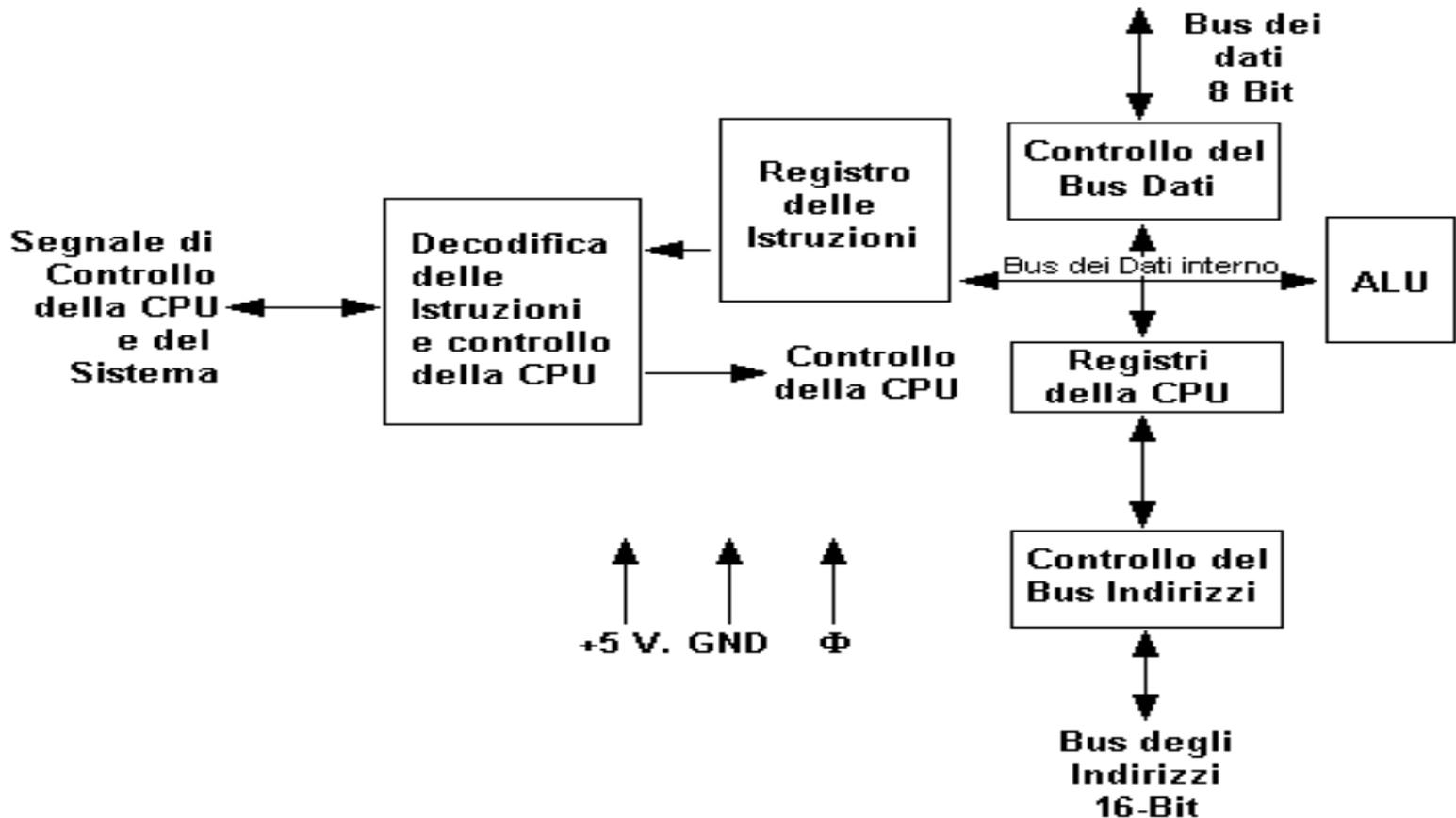


IL MICROPROCESSORE Z80

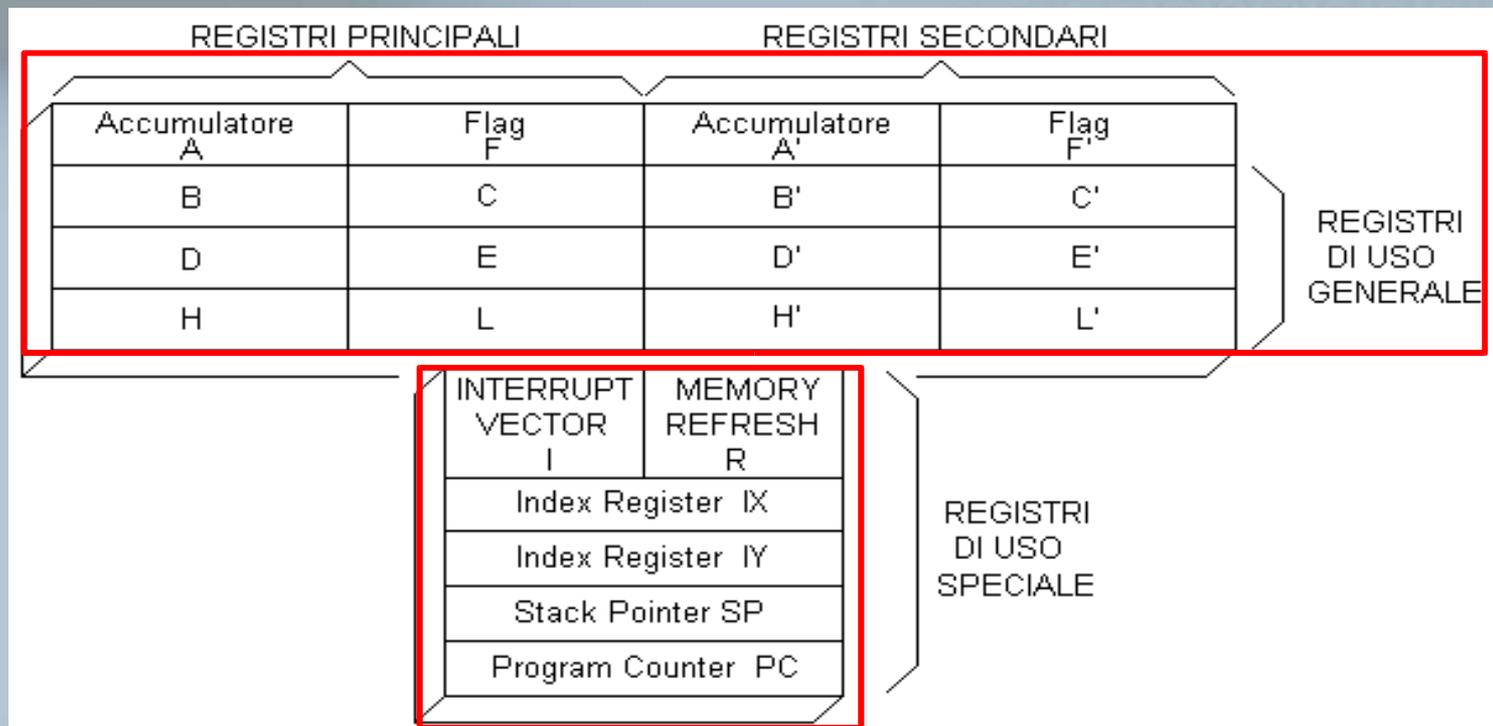
ARCHITETTURA INTERNA

- In figura é mostrato lo schema a blocchi dell'architettura interna della CPU Z80. Si tratta di un microprocessore con parola (Word) da 8 bit (1 byte) e con indirizzo da 16 bit corrispondente a 64 Kbyte di memoria indirizzabile.



ARCHITETTURA INTERNA

- Lo Z80 contiene 208 bit accessibili al programmatore in lettura/scrittura (R/W). La figura seguente illustra come questi bit siano organizzati in 18 registri a 8 bit e in 4 registri a 16 bit. Tutti i registri sono realizzati utilizzando memoria RAM statica. I registri comprendono due gruppi di sei registri di uso generale che possono essere utilizzati singolarmente come registri a 8 bit, oppure in coppia come registri a 16 bit. Oltre ai sei registri, nei due gruppi sono presenti due accumulatori e due registri di flag (singoli bit che indicano particolari stati della CPU).



Registri di uso speciale

- **Contatore di programma (Program Counter, PC)**

- Questo registro contiene l'indirizzo, a 16 bit, dell'istruzione in corso di lettura della memoria. Il PC é automaticamente incrementato ogni volta che il suo contenuto viene trasferito alle linee di indirizzamento. Quando si ha un salto nel programma, il nuovo valore dell'indirizzo viene posto nel PC, invece di eseguire l'operazione automatica di incremento.

- **Puntatore della catasta (Stack Pointer, SP)**

- Il puntatore contiene l'indirizzo a, 16 bit, dell'attuale limite superiore di uno stack posto in una qualunque zona della memoria esterna RAM. Lo stack é organizzato in modo LIFO (**L**ast-**I**n-**F**irst-**O**ut, ultimo-ingresso-prima-uscita). I dati possono essere inseriti od estratti dalla catasta (attraverso un generico registro) mediante istruzioni di PUSH (spingi) o di POP (estrai). L'uso della catasta semplifica la gestione delle interruzioni e delle subroutines.

Registri di uso speciale

- **Registri indice (Index Register, IX & IY)**

- Tali registri memorizzano un indirizzo di partenza, a 16 bit, che può essere utilizzato nei modi di indirizzamento indicizzato. Questo tipo di indirizzamento facilita grandemente la gestione di tabelle di dati. Nell'indirizzamento indicizzato il registro indice viene utilizzato come base per puntare ad una zona di memoria nella quale devono essere immagazzinati (o recuperati) dei dati. Un byte addizionale che specifica lo spostamento (displacement) dalla base é incluso nell'istruzione che utilizza questo tipo di indirizzamento. L'Indirizzo finale viene quindi ottenuto sommando all'indirizzo di partenza (contenuto nel registro indice) lo spostamento specificato. Lo spostamento é espresso come numero binario con segno in complemento a due

- **Registro dell'interruzione (Interrupt Page Address Register, I)**

- La CPU Z80 può operare in un modo in cui la risposta ad una interruzione può essere realizzata mediante una chiamata indiretta (call) ad una qualsiasi locazione di memoria. Il registro I viene usato per memorizzare gli 8 bit più significativi dell'indirizzo indiretto, mentre il dispositivo generante l'interruzione fornisce gli 8 bit meno significativi dell'indirizzo. Questa caratteristica permette che i programmi di gestione delle interruzioni siano posizionati in modo dinamico ovunque nella memoria, mentre il tempo di accesso ai suddetti programmi resta in ogni caso minimo.

Registri di uso speciale

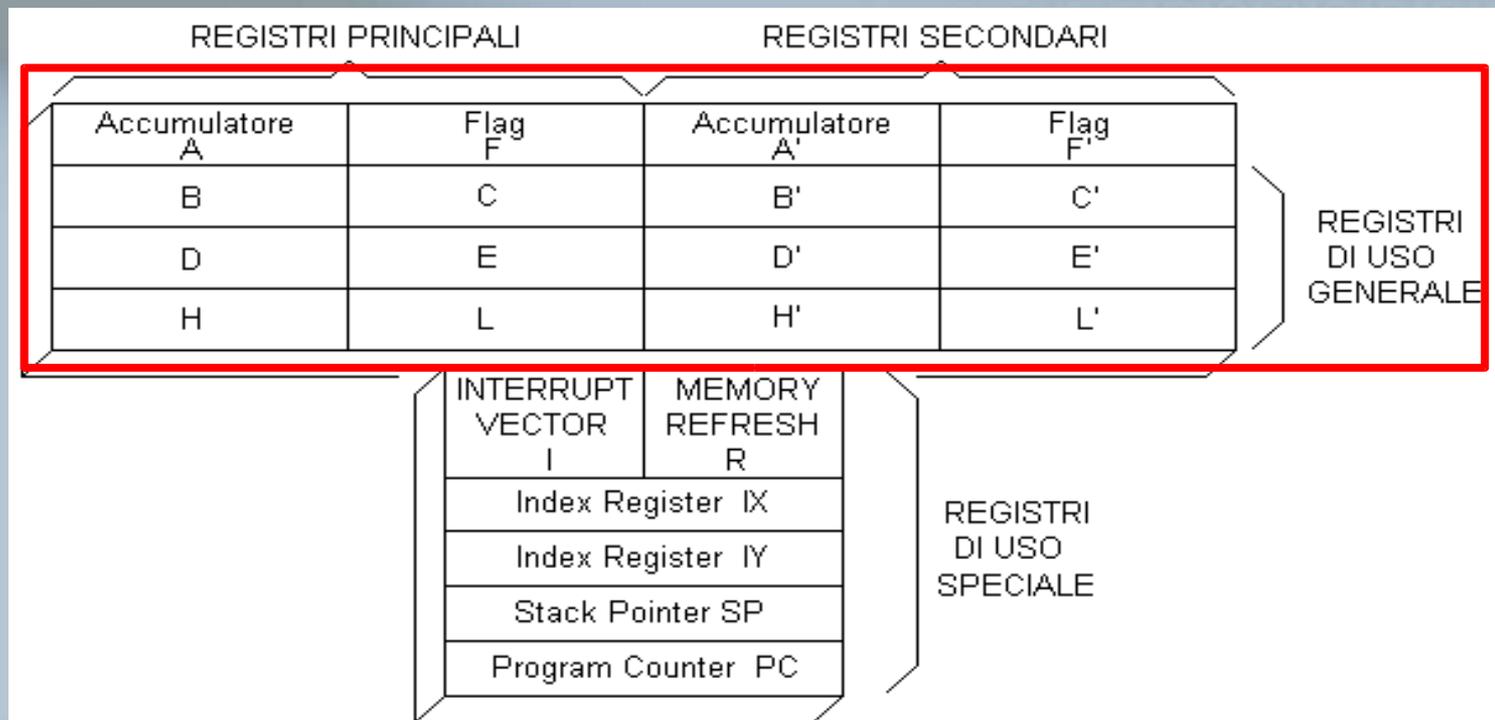
- **Registro di rinfresco della memoria (Memory Refresh Register, R)**
-
- Lo Z80 contiene un contatore di rinfresco della memoria, per facilitare l'uso delle memorie dinamiche, Sette degli 8 bit di questo registro sono automaticamente incrementati dopo ogni lettura di un'istruzione. L'ottavo bit, mediante un'istruzione apposita, rimane invece come programmato.
- **Accumulatore e registri delle flag**
-
- Nella CPU vi sono due accumulatori indipendenti ad 8 bit con i registri delle flag associati. L'accumulatore contiene il risultato di una operazione aritmetico/logica mentre il registro delle flag indica condizioni specifiche per le operazioni effettuate (p. es. se il risultato di una operazione é o non zero).

Registri di uso speciale: i Flag

- **Vediamo adesso il significato di ciascun bit del registro F:**
- **Il bit 0 (C).** E' il flag del carry che in genere è settato a 1 quando si verifica un riporto sull'ultimo bit. Inoltre è alterato dalle operazioni di shift e dalle operazioni logiche.
- **Il bit 1 (N).** E' il flag N che viene settato quando viene effettuata una sottrazione o un decremento. Non può essere testato mediante i salti condizionati, ma è sfruttato dalla operazione DAA.
- **Il bit 2 (P/V).** E' il flag di parità o di overflow. Quando lavora come flag di parità (in genere nelle operazioni logiche e negli shift) viene posto a uno quando il numero dei bit posti a 1 nel risultato è pari. Se invece lavora come flag di overflow (in genere nelle operazioni aritmetiche) viene settato a 1 quando il segno del risultato è errato. Supponiamo ad esempio di voler sommare -1 a -128. Le istruzioni sono:
 - LD A,255 ; -1=255 in complemento a 2
 - ADD A,128 ; -128=128 in complemento a 2Il risultato corretto sarebbe -129 ma invece nel registro A troviamo il valore 127. In questo caso il flag P/V sarà posto a 1.
- **Il bit 3.** Questo bit non è un flag e non ha alcun significato. Il suo valore è casuale.
- **Il bit 4 (H).** E' il flag dell'half-carry e indica il riporto che avviene al bit 3 dell'accumulatore. Non può essere testato con le istruzioni di salto condizionato, ma è sfruttato dall'operazione DAA.
- **Il bit 5.** Questo bit non è un flag e non ha alcun significato. Il suo valore è casuale.
- **Il bit 6 (Z).** E' il flag dello zero: viene posto a 1 nel momento in cui il risultato di un'operazione è 0.
- **Il bit 7 (S).** E' il flag del segno. Quando viene posto a 1 significa che il risultato dell'operazione è negativo (in altre parole il bit più significativo del risultato è uguale a 1).

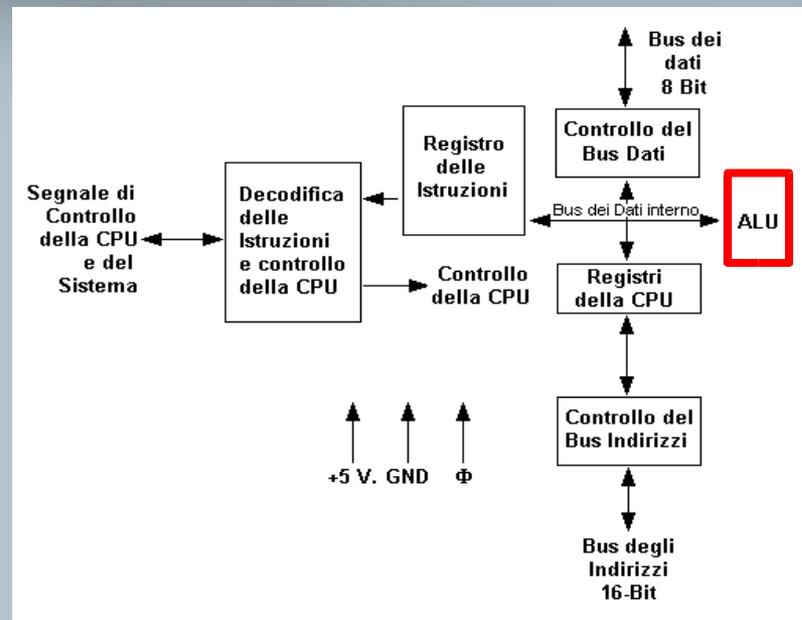
Registri di uso generale

- La CPU contiene due insiemi accoppiati di registri di uso generale; ciascun insieme é formato da sei registri ad 8 bit che possono essere utilizzati individualmente od a coppie come registri da 16 bit. Un insieme é formato dai registri BC, DE, HL e l'insieme complementare da B'C', D'É, H'L'



Unità aritmetica e logica (ALU)

- Le istruzioni aritmetiche e logiche ad 8 bit della CPU sono eseguite mediante la ALU. Internamente, la ALU comunica con i registri e con il bus dei dati esterno mediante il bus dei dati interno. I tipi di funzioni eseguite comprendono:
-
- Somma, Sottrazione, AND Logico, OR logico, OR ESCLUSIVO Logico, Confronto, Scorrimento o Rotazione a destra o sinistra, Incremento, Decremento, Set dei bit, Reset dei bit, Test dei bit.

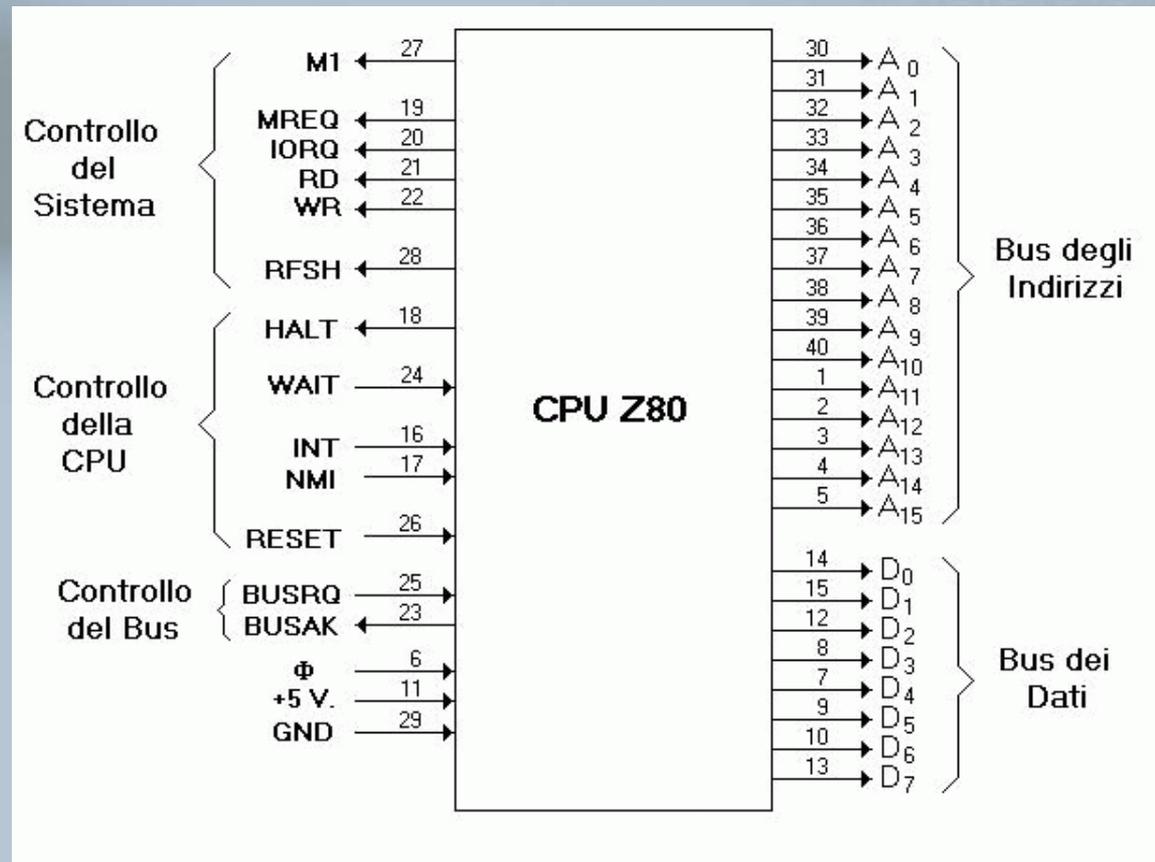


Funzionamento della CPU

- Quando lo Z80 esegue un programma che risiede nella memoria ad esso associata, le istruzioni vengono lette una dopo l'altra dalla memoria stessa.
- Questa é la fase di acquisizione (**fetch**) dell'istruzione: l'indirizzo contenuto nel contatore di programma viene posto sul bus degli indirizzi, vengono generati e posti sul bus di controllo i segnali necessari per attivare la memoria che preleva l'istruzione dall'indirizzo indicato e la invia sul bus dei dati; infine l'istruzione viene memorizzata dalla CPU nell'apposito registro delle istruzioni.
- Segue quindi la fase di esecuzione (**Execute**) dell'istruzione: il contatore di programma viene incrementato per prepararlo al successivo ciclo di acquisizione; l'istruzione viene decodificata ed eseguita. L'unità di controllo della CPU coordina le varie temporizzazioni assicurando il corretto funzionamento del decodificatore delle istruzioni e dell'unità aritmetico/logica (ALU) che esegue tutte le operazioni aritmetiche e logiche previste dal set di istruzioni dello Z80.

DESCRIZIONE DEI PIEDINI

- La CPU Z80 é incapsulata in un contenitore standard Dual in Line a 40 piedini; nella figura é mostrata la disposizione funzionale dei piedini.



DESCRIZIONE DEI PIEDINI

A0 – A15 (BUS DEGLI INDIRIZZI)

Uscite tri-state, attive alte, unidirezionali, **A0-A15** costituiscono il bus degli indirizzi in cui A0 é il bit meno significativo. Il bus fornisce gli indirizzi per lo scambio dei dati con la memoria (max. 64 Kbyte = 2¹⁶) e con i dispositivi di I/O (ingresso/uscita) ossia è il canale utilizzato per trasferire l'indirizzo di una locazione di memoria o di un dispositivo di I/O.

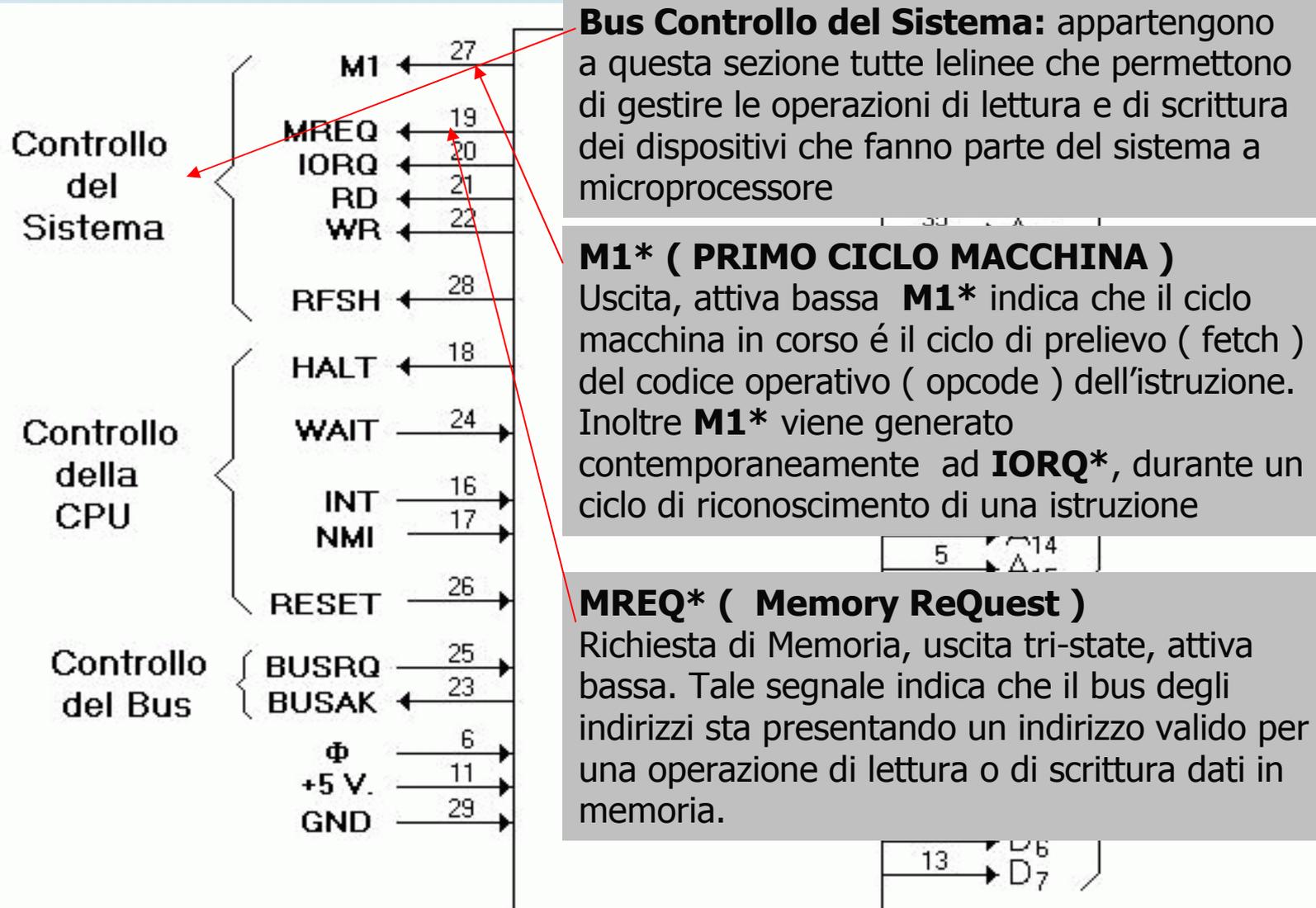
L'indirizzamento dei dispositivi di I/O viene effettuato utilizzando gli 8 bit meno significativi del bus degli indirizzi, A0.....A7; ciò consente di selezionare direttamente fino a 256 (2⁸) porte di ingresso o di uscita. Durante l'operazione di refresh della memoria, i sette bit meno significativi contengono un indirizzo di refresh valido.

D0 – D7 (BUS DEI DATI)

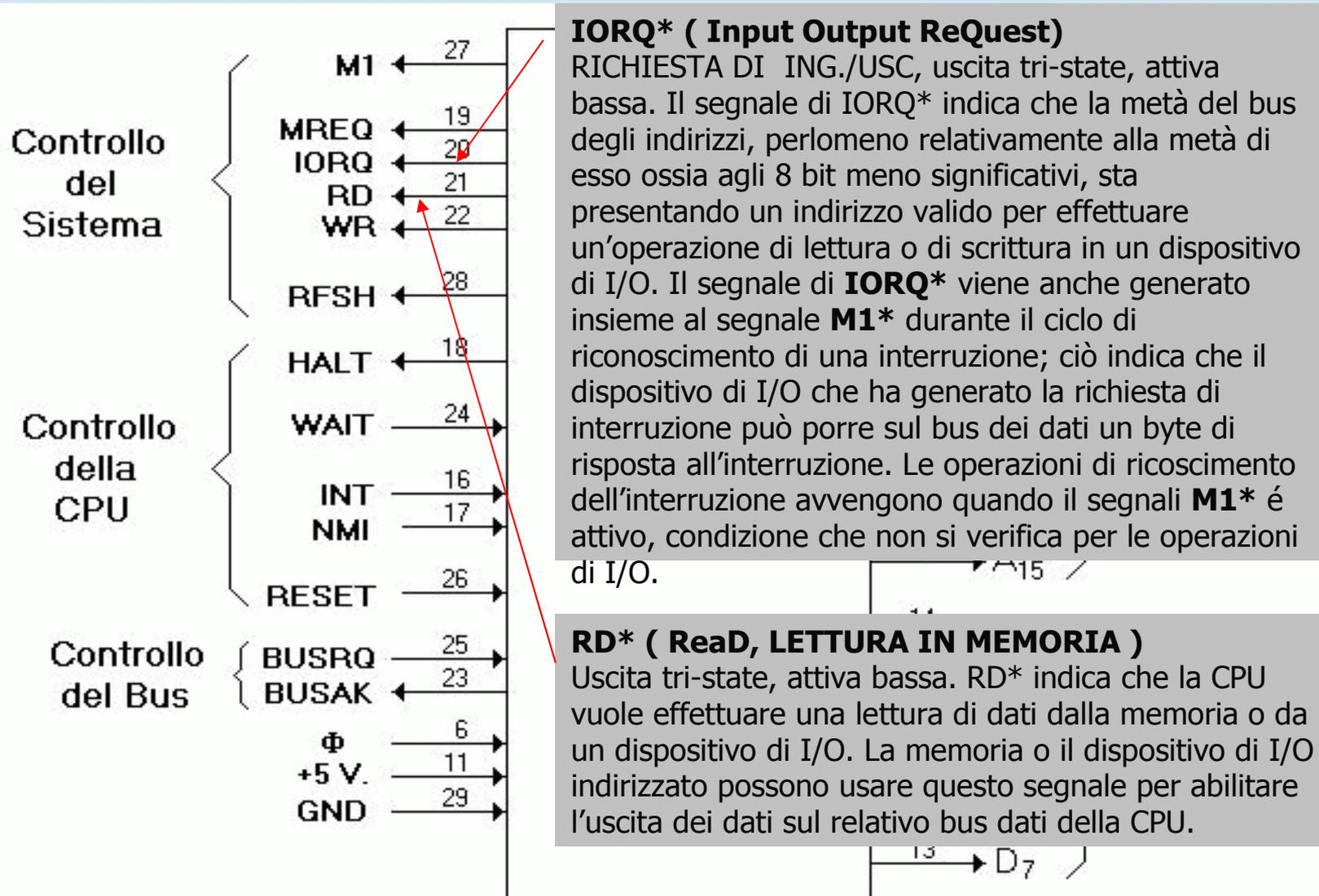
Ingressi /uscite, tri-state, attive alte. **D0-D7** formano il bus bidirezionale ad 8 bit dei dati, Tale bus viene utilizzato per lo scambio dei dati tra la CPU e la memoria e tra la CPU e i dispositivi di I/O.



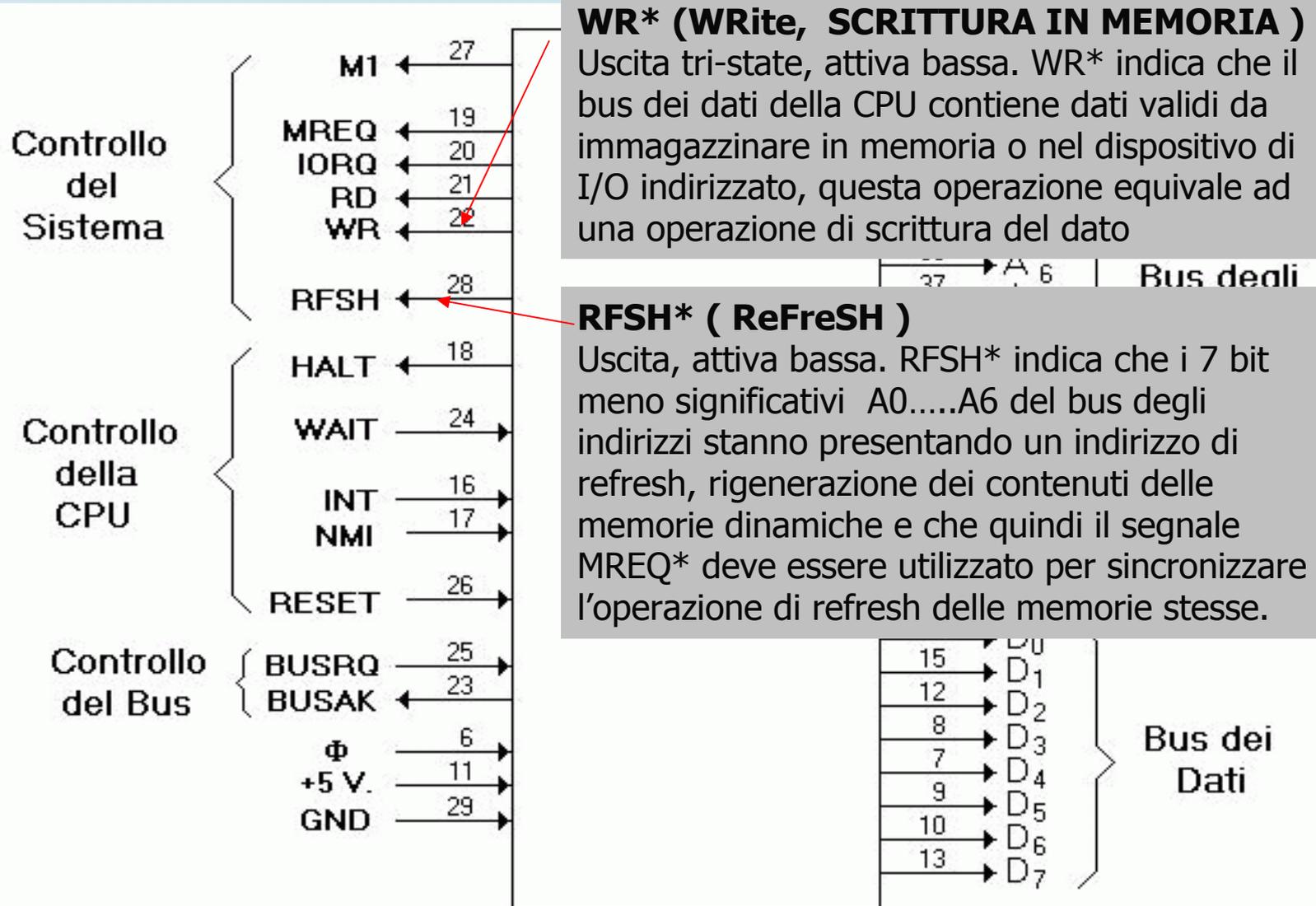
DESCRIZIONE DEI PIEDINI



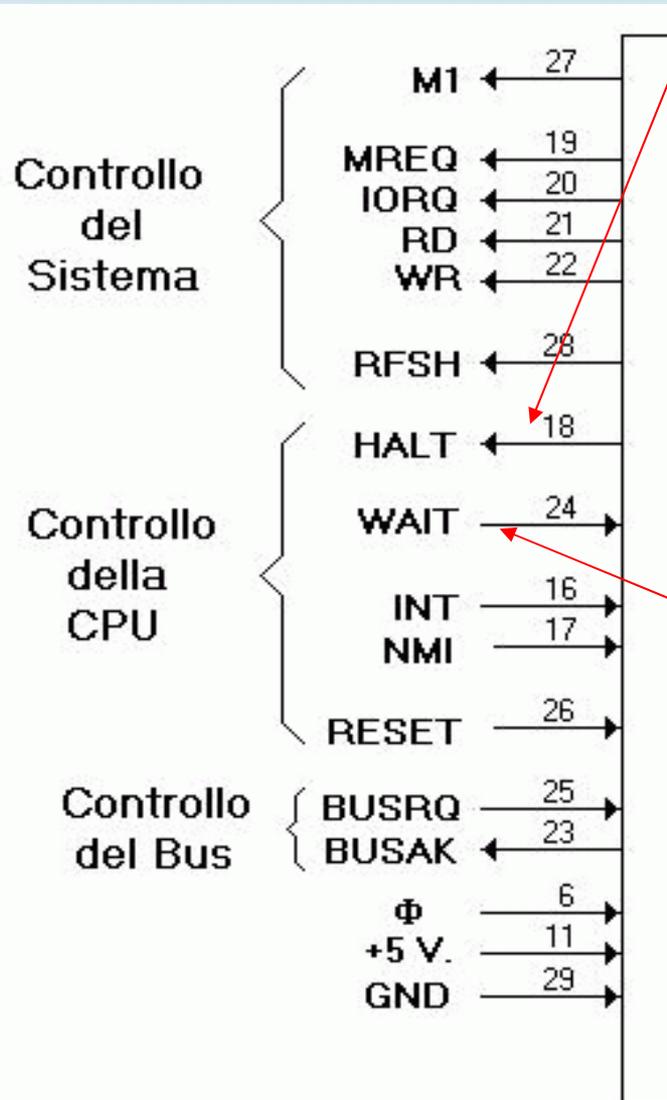
DESCRIZIONE DEI PIEDINI



DESCRIZIONE DEI PIEDINI



DESCRIZIONE DEI PIEDINI



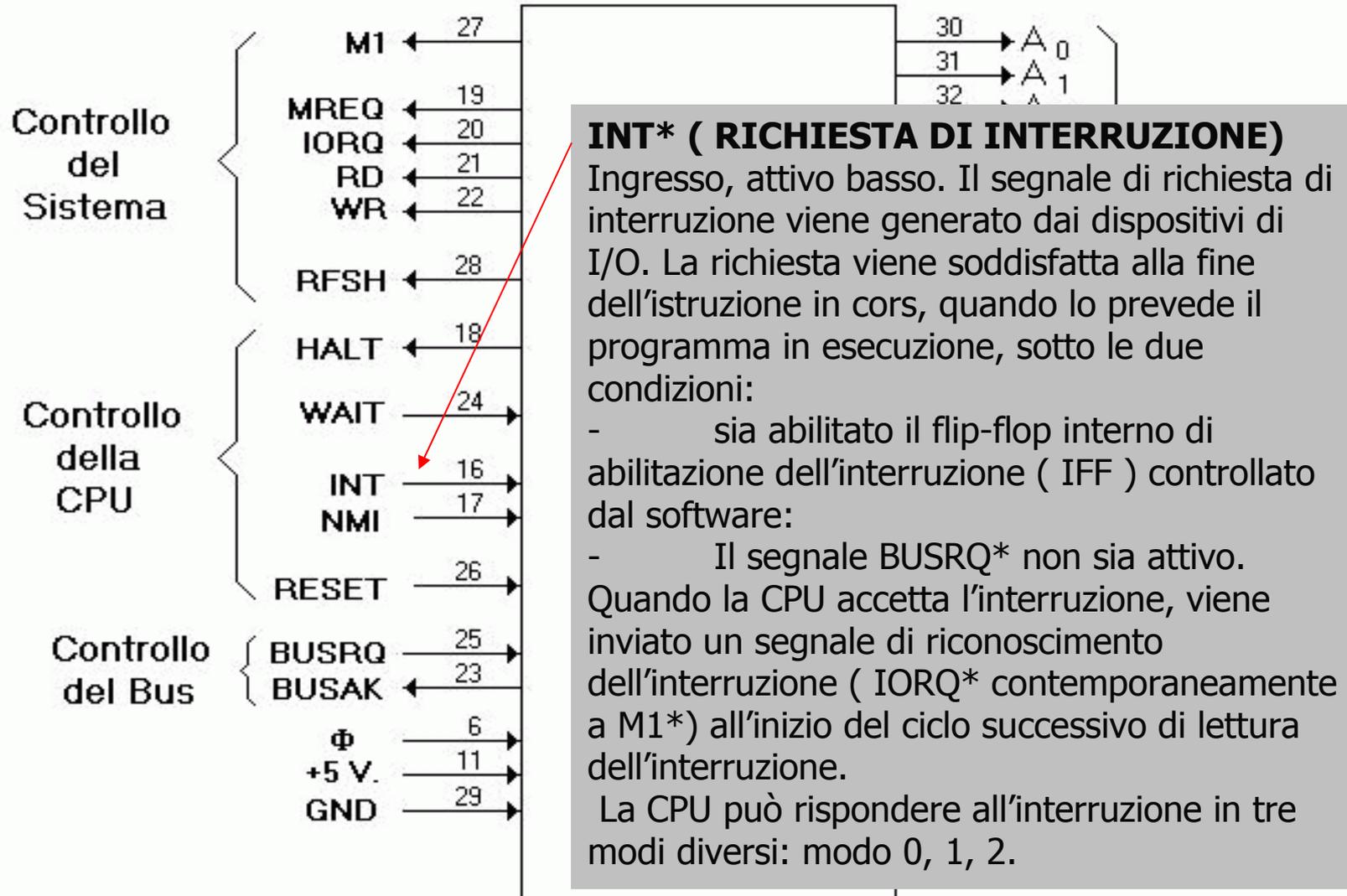
HALT* (STATO DI ALT)

Uscita, attiva bassa. HALT* indica che la CPU ha eseguito un'istruzione di HALT ed è in attesa di un segnale di interruzione non mascherabile o mascherabile (se quest'ultima è stata precedentemente abilitata) per poter riprendere l'esecuzione del programma. Finché perdura lo stato di alt, la CPU esegue continuamente delle istruzioni NOP (No Operation) per mantenere l'attività di refresh in quanto il perdurare di questo stato potrebbe portare alla perdita dei dati contenuti nelle memorie dinamiche, oltre che attraverso un segnale di interruzione è possibile agire sulla linea di RESET.

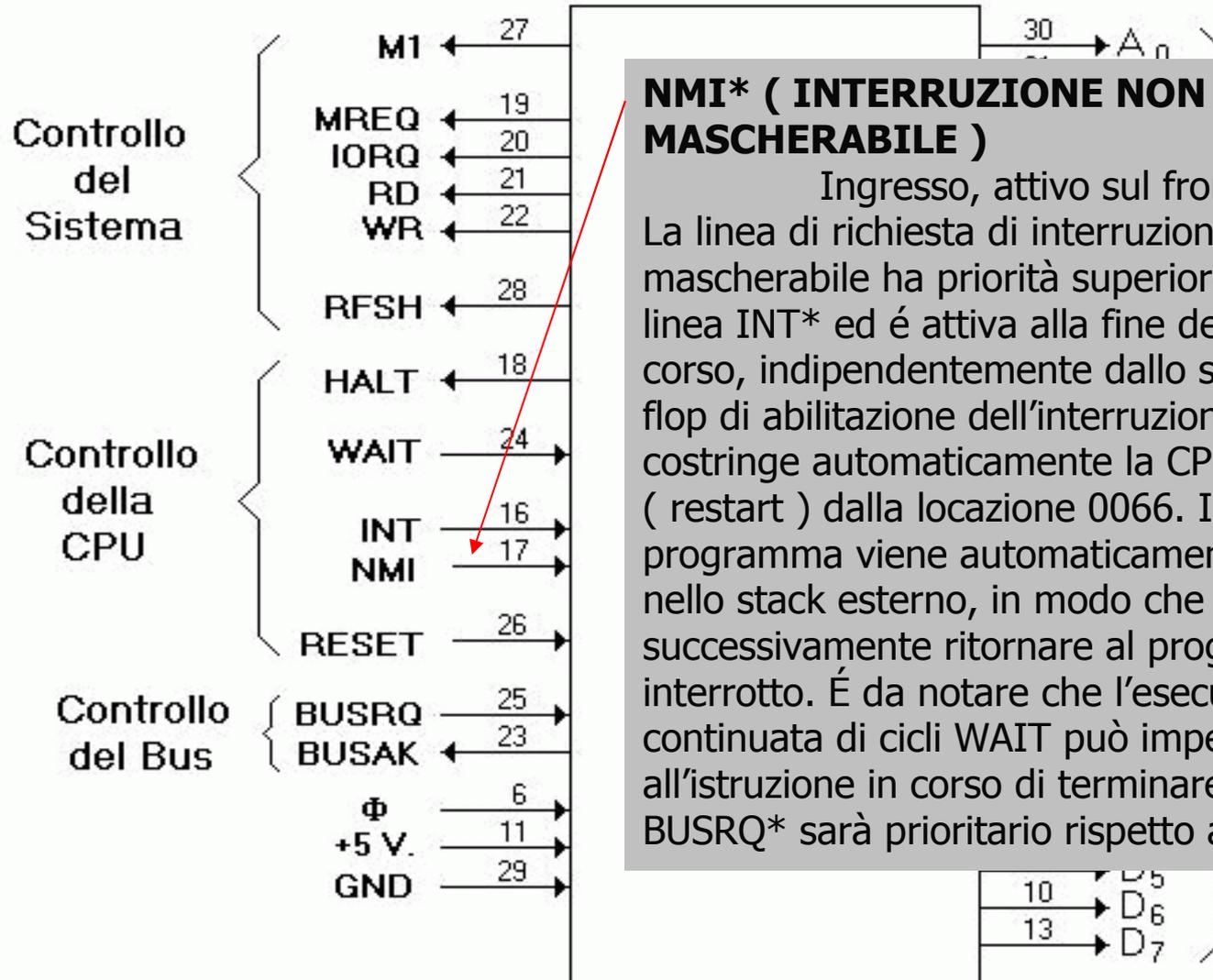
WAIT* (ATTESA)

Ingresso, attivo basso. WAIT* indica alla CPU che la memoria o il dispositivo di I/O non è pronto per un trasferimento dati. Finché questo segnale è attivo la CPU rimane in stato di attesa. Questo segnale permette che dispositivi di I/O o di memoria di qualunque velocità si sincronizzino con la velocità della CPU, rimane attivo fino a quando permane la situazione di indisponibilità. Questo segnale è inviato dal dispositivo di I/O interessato al trasferimento dei dati verso la CPU la quale, durante l'esecuzione delle singole istruzioni va a testare, in certi istanti prefissati, lo stato di questo terminale, se non lo trova attivo significa che i dispositivi coinvolti nello scambio di informazioni sono pronti a trasmettere o ricevere dati.

DESCRIZIONE DEI PIEDINI



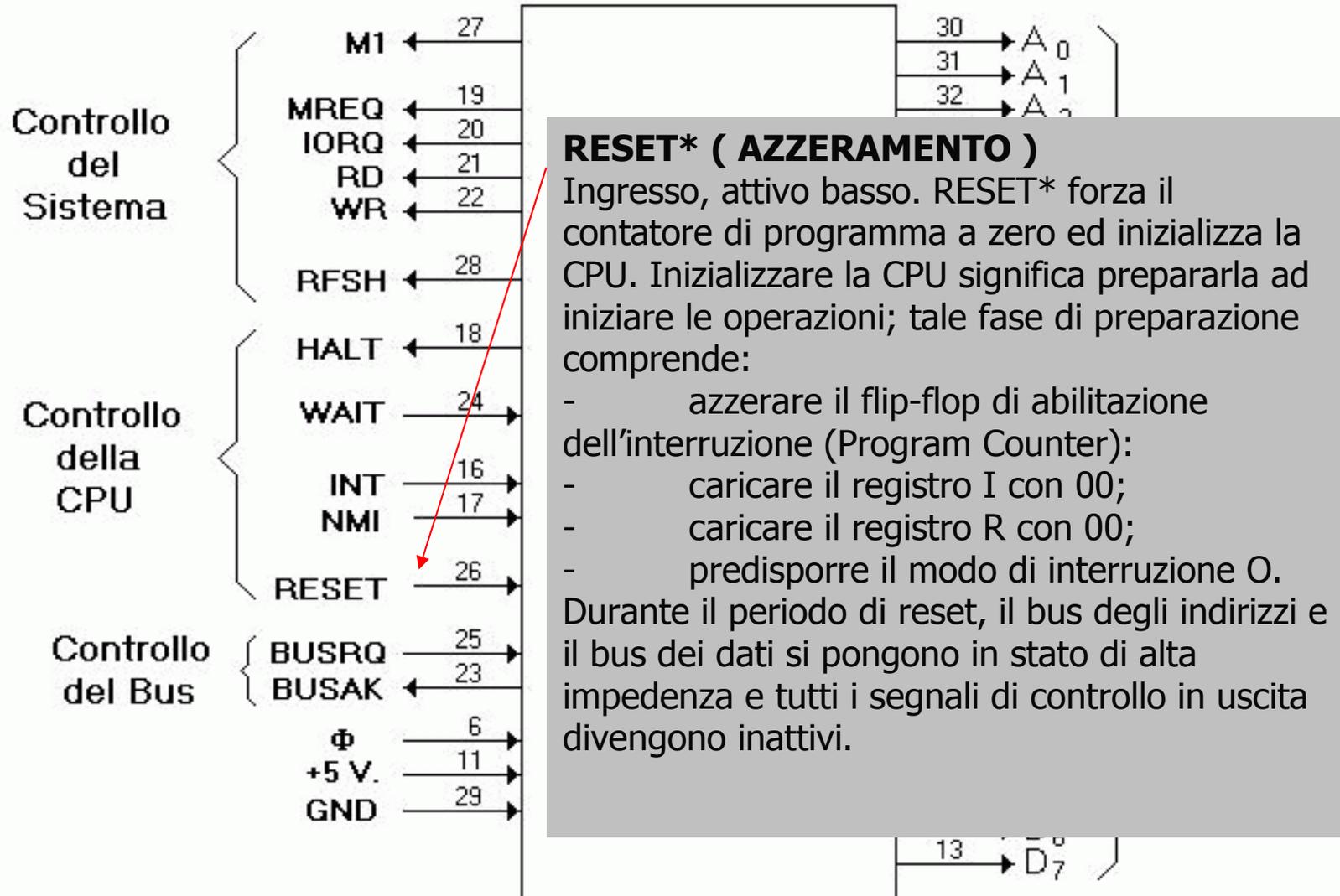
DESCRIZIONE DEI PIEDINI



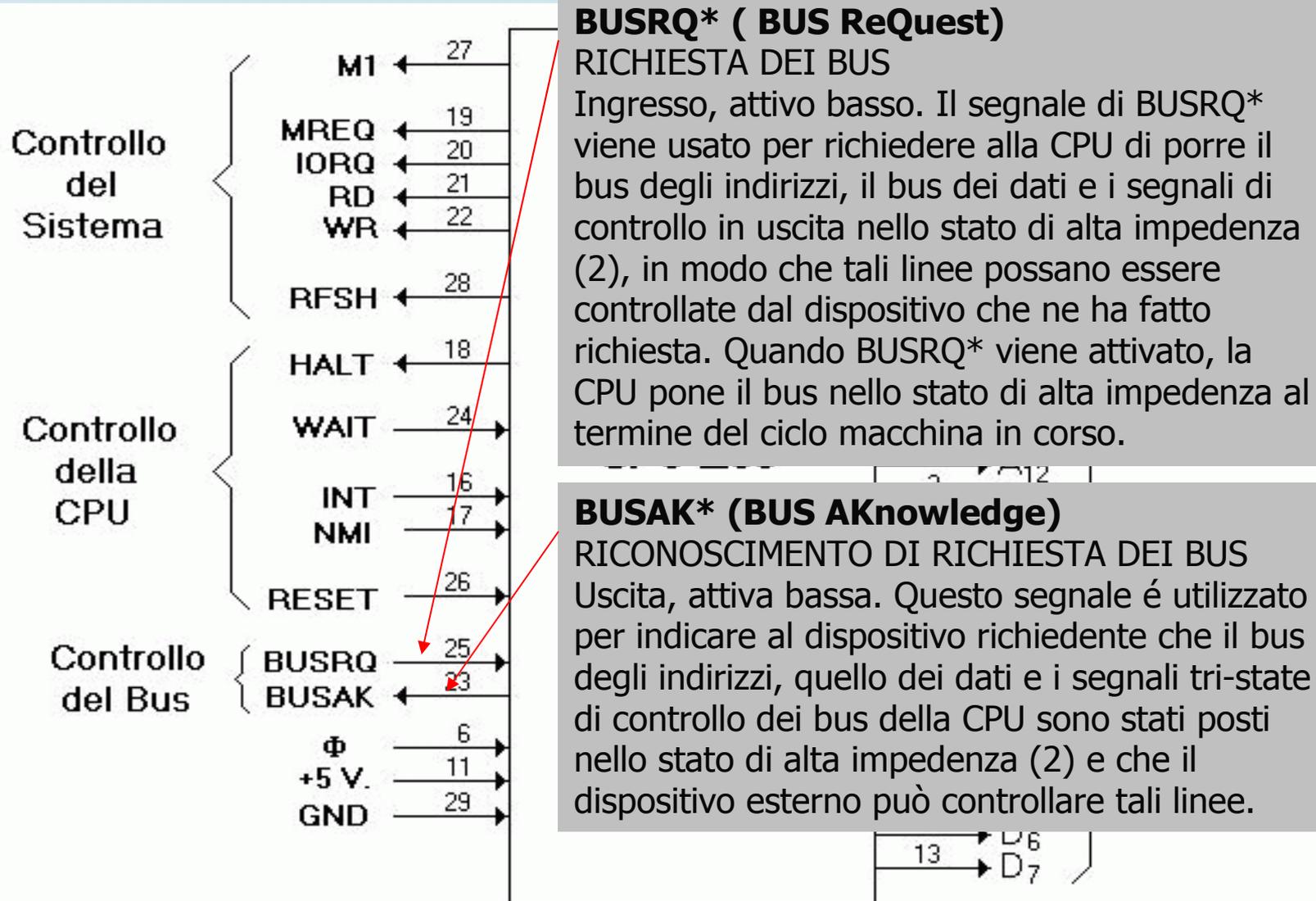
NMI* (INTERRUZIONE NON MASCHERABILE)

Ingresso, attivo sul fronte di discesa. La linea di richiesta di interruzione non mascherabile ha priorità superiore rispetto alla linea INT* ed è attiva alla fine dell'istruzione in corso, indipendentemente dallo stato del flip-flop di abilitazione dell'interruzione. NMI* costringe automaticamente la CPU a ripartire (restart) dalla locazione 0066. Il contatore di programma viene automaticamente salvato nello stack esterno, in modo che la CPU possa successivamente ritornare al programma interrotto. È da notare che l'esecuzione continuata di cicli WAIT può impedire all'istruzione in corso di terminare e che un BUSRQ* sarà prioritario rispetto ad un NMI*.

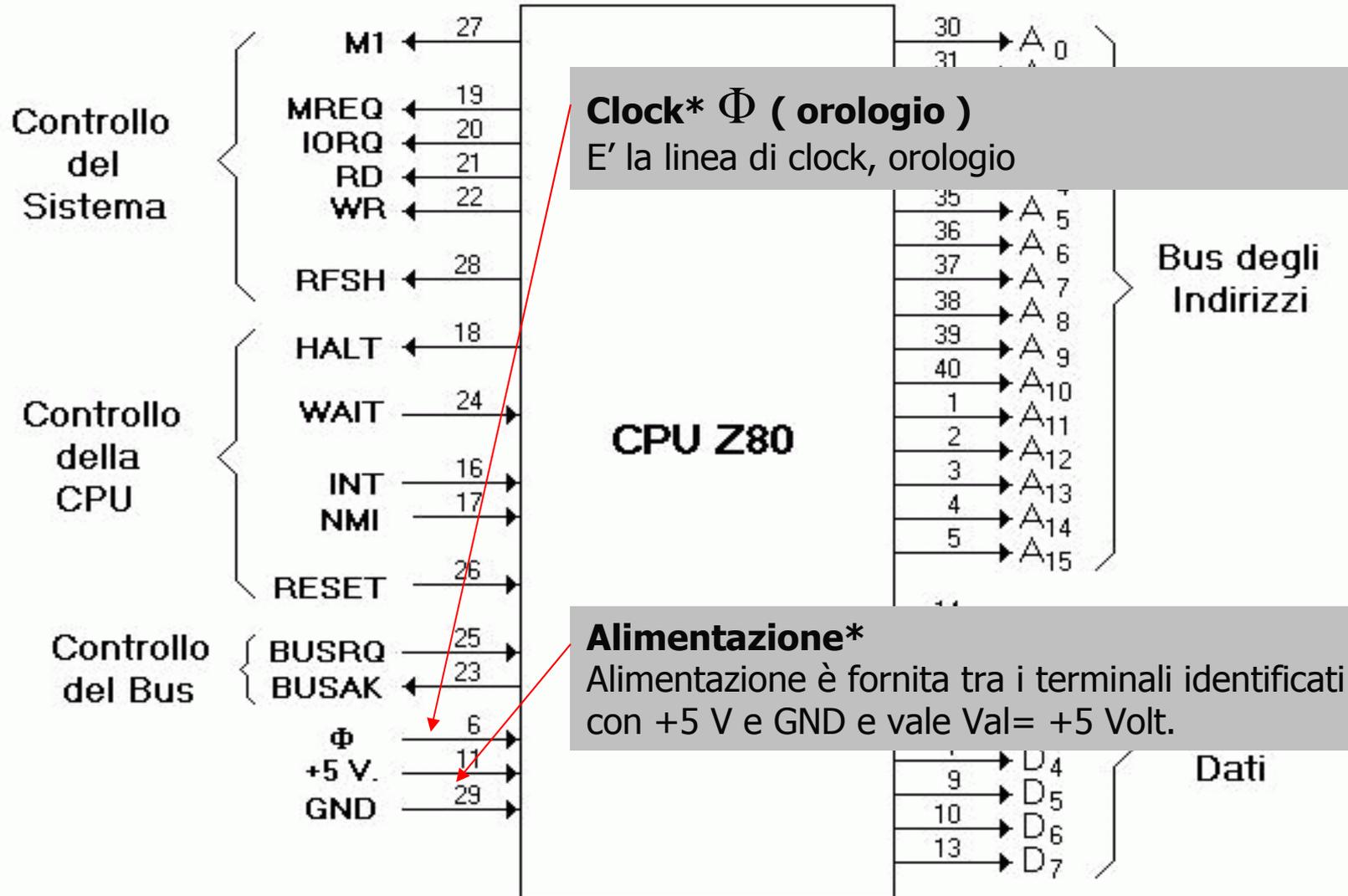
DESCRIZIONE DEI PIEDINI



DESCRIZIONE DEI PIEDINI

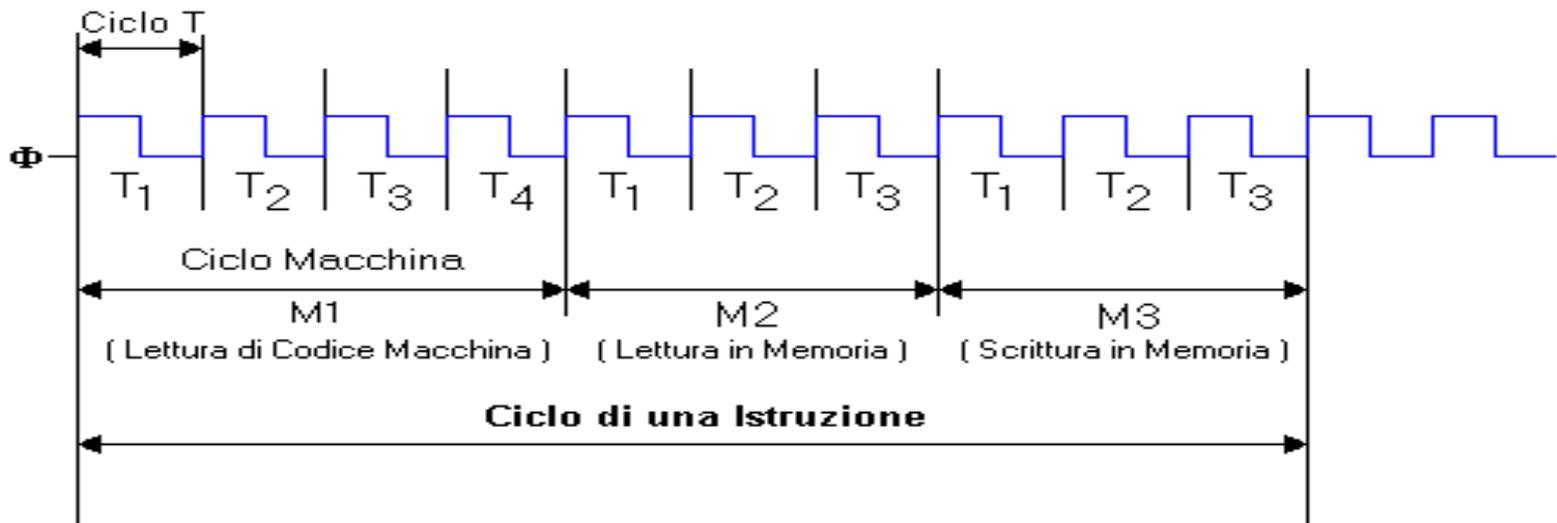


DESCRIZIONE DEI PIEDINI

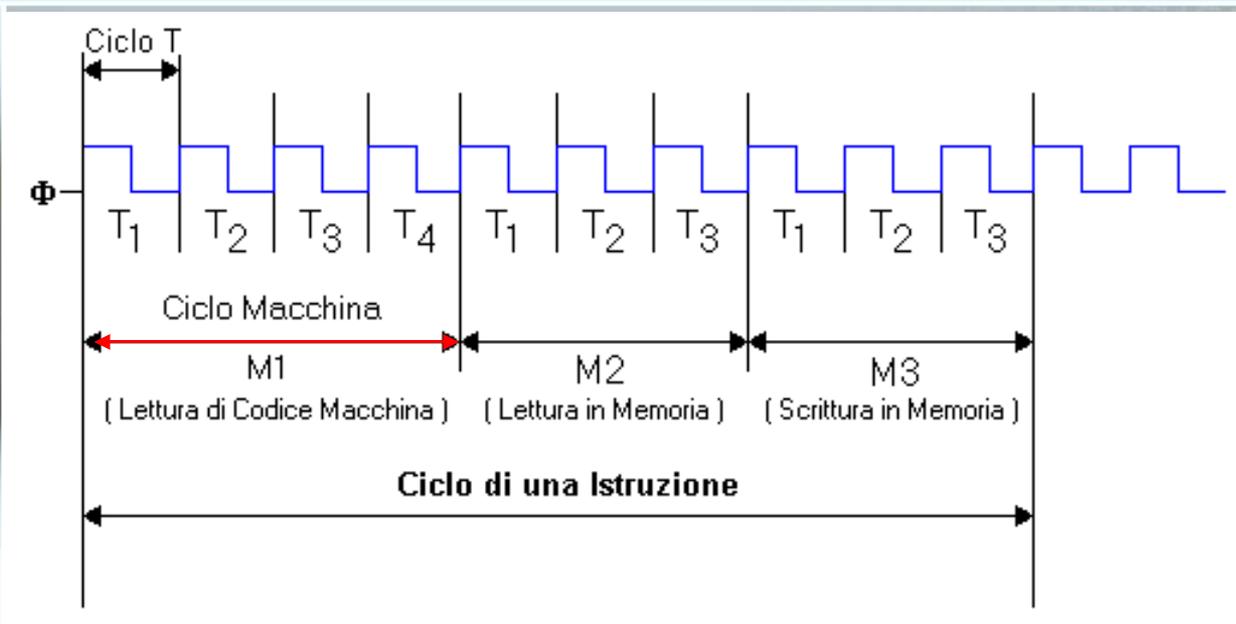


Segnali di TEMPORIZZAZIONE

- L'operatività di un sistema a microprocessore è funzione della logica che gestisce le temporizzazioni, infatti tutte le operazioni vengono svolte sotto l'azione dei segnali emessi dalla logica di controllo che scandisce i tempi di attivazione delle linee coinvolte nell'esecuzione delle varie istruzioni.
- La CPU esegue le istruzioni passando attraverso una serie ben precisa di operazioni fondamentali, od elementari:
 - - Lettura/Scrittura in memoria
 - - Lettura/Scrittura nei dispositivi di I/O
 - - Riconoscimento delle interruzioni.
- Tutte le istruzioni sono semplicemente una serie di queste operazioni fondamentali. Ognuna di queste operazioni può richiedere da tre a sei periodi di clock(Φ) per essere portata a termine; inoltre esse possono essere rallentate, allo scopo di sincronizzare la CPU con la velocità dei dispositivi esterni. Questi periodi di clock elementari sono chiamati cicli T, mentre le operazioni elementari cicli M (Macchina).
- La figura successiva mostra come una istruzione possa essere descritta semplicemente come una serie di cicli M e T.



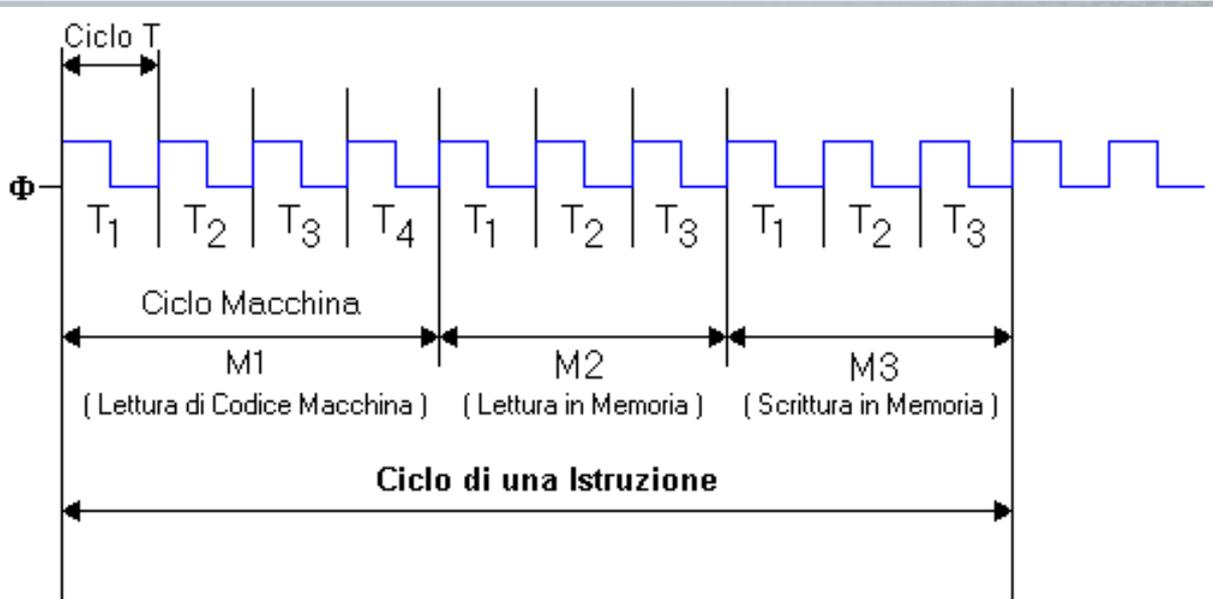
Segnali di TEMPORIZZAZIONE



Si noti il ciclo di fetch (Lettura del codice operativo) in questa istruzione. Il ciclo di fetch (M1) é utilizzato per leggere e decodificare il codice operativo (OP CODE) della istruzione da eseguire. I successivi cicli macchina muovono dati tra la CPU e la memoria od i dispositivi di I/O e possono essere costituiti da tre a cinque cicli T.

La durata di un ciclo T è funzione della frequenza di clock che può assumere i valori di 2,5 MHz per la versione base dello Z80, 4 MHz per lo Z80A e 6 MHz per lo Z80B.

Segnali di TEMPORIZZAZIONE



- Tutte le temporizzazioni della CPU possono essere spezzate in un certo numero di semplici diagrammi dei tempi; tali diagrammi mostrano le seguenti operazioni elementari:
 - - Ciclo di lettura del codice operativo di una istruzione.
 - - Cicli di lettura o scrittura di dati in memoria.
 - - Cicli di lettura o scrittura di I/O.
 - - Ciclo di Richiesta/Riconoscimento del bus.
 - - Ciclo di Richiesta/Riconoscimento di interruzione.
 - - Ciclo di Richiesta/Riconoscimento di interruzione non mascherabile.
 - - Uscita da una istruzione di HALT.

Segnali di TEMPORIZZAZIONE

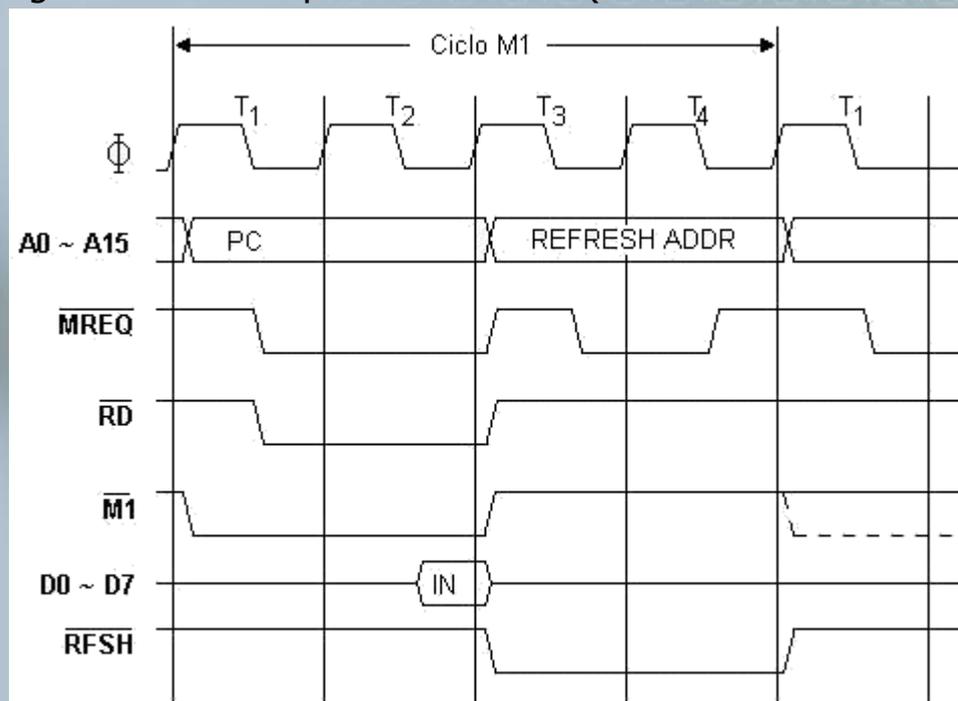
■ Ciclo di lettura del codice operativo (**fetch dell'istruzione**).

Nella figura seguente é riportato il diagramma dei tempi di un ciclo M1 (lettura di un codice operativo)

In Questo ciclo; fondamentale per l'operatività del microprocessore, si procede al prelievo in memoria di un dato che viene interpretato come codice operativo.

Il ciclo di fetch indicato con M1, comporta normalmente la durata di quattro cicli T e le linee interessate sono: MREQ, RD, M1, RFSH, WAIT.

Dall'esame della figura accanto si possono individuare le sequenze secondo le quali vengono attivate le linee suddette:



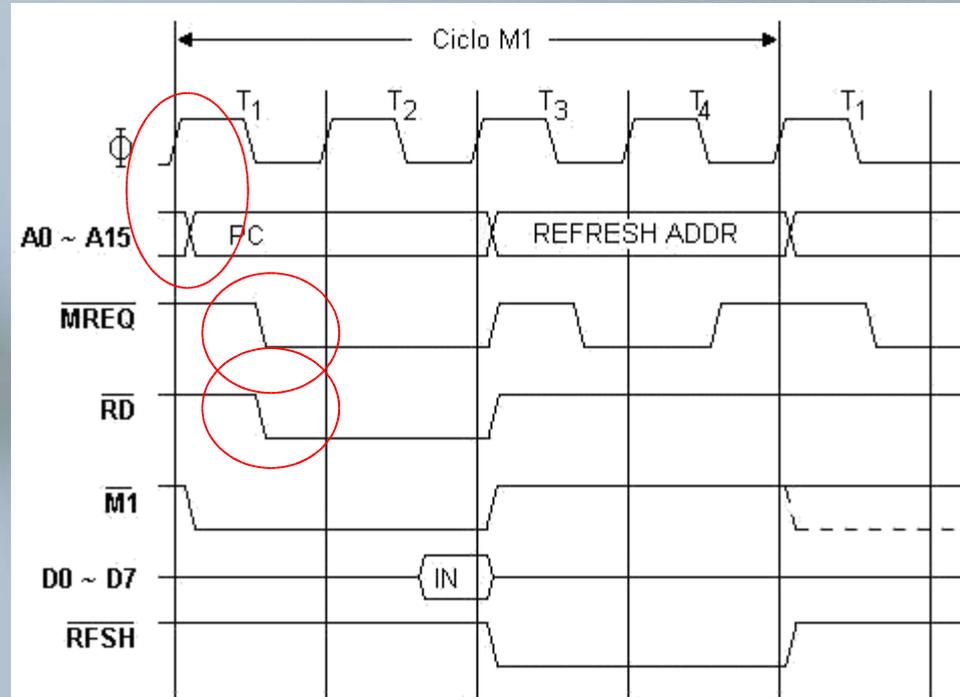
Temporizzazione della CPU Z80 nel ciclo M1 (prelievo del codice operativo)

Segnali di TEMPORIZZAZIONE

■ Ciclo di lettura del codice operativo (fetch dell'istruzione).

Si noti che il PC viene presentato sul bus degli indirizzi all'inizio del ciclo M1 (periodo iniziale, T1). Sul fronte di salita T1 si attiva la linea M1. Metà periodo di clock più tardi, il segnale MREQ* diventa attivo.

In questo tempo l'indirizzo ha avuto modo di stabilizzarsi, così che il fronte discendente di tale segnale può essere utilizzato direttamente come clock di abilitazione per le memorie dinamiche. La linea RD* diventa anch'essa attiva, ad indicare che il dato in lettura dalla memoria può essere presentato sul bus dei dati, rendendo pertanto possibile il trasferimento del contenuto della locazione di memoria indirizzata.



Temporizzazione della CPU Z80 nel ciclo M1 (prelievo del codice operativo)

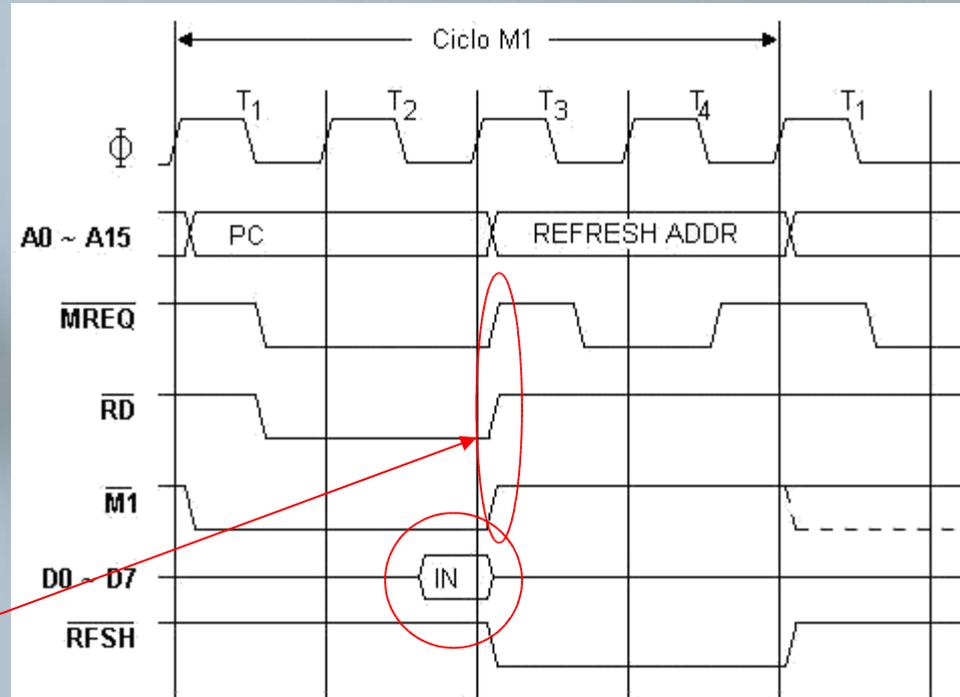
Segnali di TEMPORIZZAZIONE

■ Ciclo di lettura del codice operativo (fetch dell'istruzione).

A questo punto la CPU si dispone alla lettura del dato che avviene solo se la linea di WAIT non risulta attiva.

La CPU campiona il dato, proveniente dalla memoria e presente sul bus dei dati, con il fronte di salita del clock dello stato T3; questo stesso fronte è utilizzato dalla CPU per disattivare i segnali RD* e MREQ*. Quindi, il dato è campionato dalla CPU prima che il segnale RD* diventi inattivo.

Effettuata la lettura vengono disattivate le linee RD, M1 e MREQ e contemporaneamente è attivata la linea di RFSH.

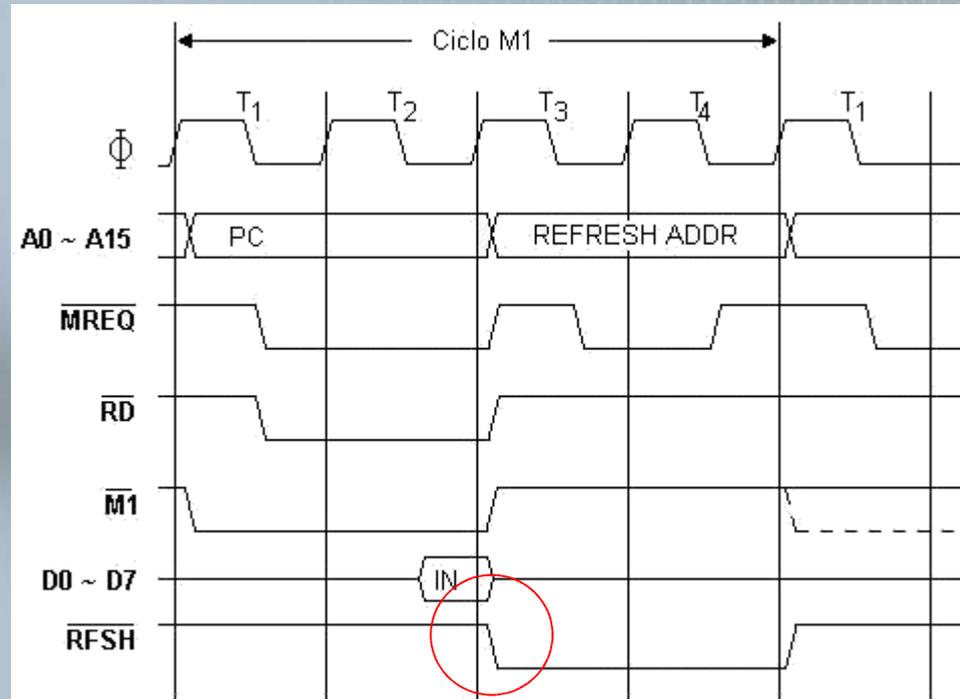


Temporizzazione della CPU Z80 nel ciclo M1 (prelievo del codice operativo)

Segnali di TEMPORIZZAZIONE

■ Ciclo di lettura del codice operativo (fetch dell'istruzione).

I periodi di clock T3 e T4 del ciclo macchina M1 sono utilizzati per rinfrescare le memorie dinamiche. La CPU utilizza questo tempo per decodificare ed eseguire l'istruzione appena letta, cosicché nessun'altra operazione potrebbe essere eseguita durante questo tempo. Durante T3 e T4 i 7 bit meno significativi del bus degli indirizzi presentano un indirizzo valido di rinfresco della memoria ed il segnale RFSH* diventa attivo, ad indicare che si sta eseguendo un'operazione di rinfresco di tutte le memorie dinamiche.



Temporizzazione della CPU Z80 nel ciclo M1 (prelievo del codice operativo)

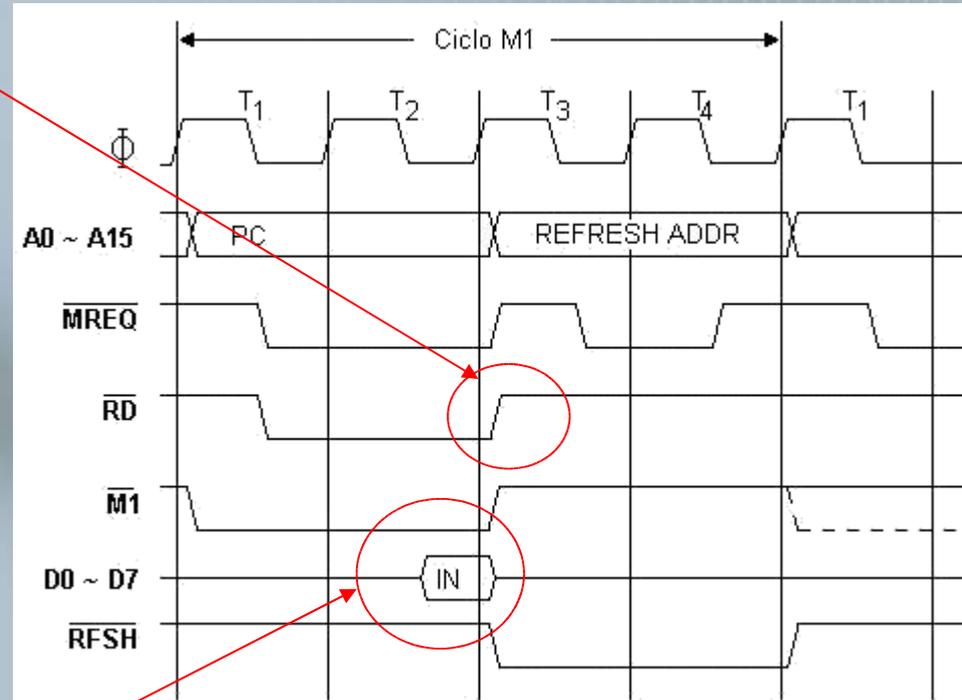
diagramma dei tempi di un ciclo M1 (lettura di un codice operativo)

Segnali di TEMPORIZZAZIONE

Ciclo di lettura del codice operativo (fetch dell'istruzione).

Si noti che il segnale RD* non viene generato durante il tempo di rinfresco, per evitare che i dati provenienti da diversi segmenti di memoria siano presentati contemporaneamente sul bus dei dati. Il segnale MREQ*, durante il tempo di rinfresco, può essere utilizzato per realizzare l'operazione di lettura di rinfresco di tutti gli elementi di memoria. Il segnale di rinfresco non può essere utilizzato da solo, in quanto l'indirizzo di rinfresco è garantito stabile solamente durante il tempo MREQ*.

Le linee D0.....D7 sono poste nello stato di alta impedenza per evitare la presenza di dati sul bus dati contemporaneamente all'esecuzione del ciclo di rinfresco.



Temporizzazione della CPU Z80 nel ciclo M1 (prelievo del codice operativo)

diagramma dei tempi di un ciclo M1 (lettura di un codice operativo)

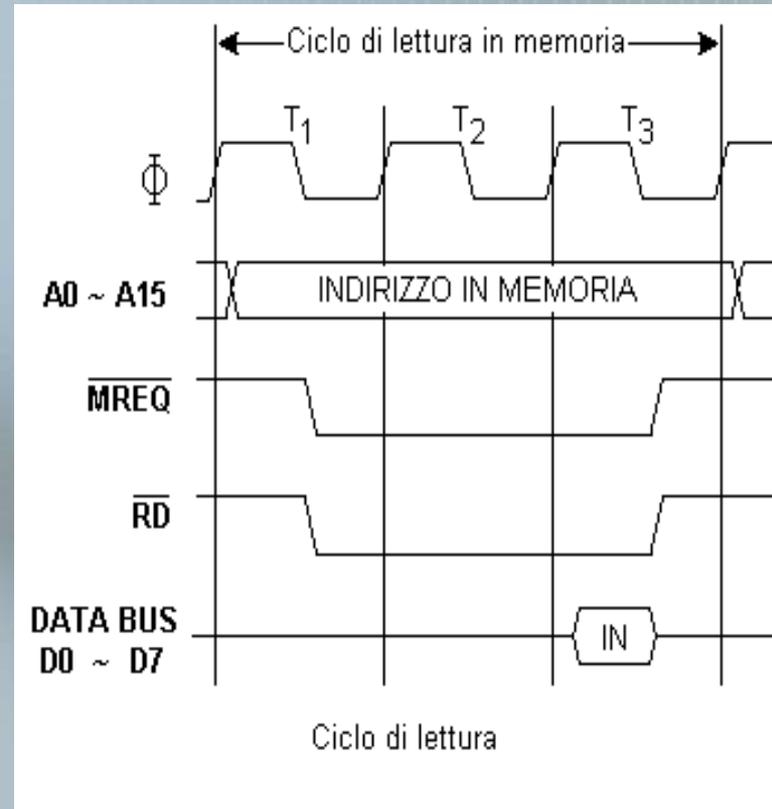
Segnali di TEMPORIZZAZIONE

Ciclo di lettura e scrittura in memoria

La figura seguente mostra le relazioni di tempo esistenti durante i cicli di lettura o di scrittura in memoria. Questi cicli sono composti generalmente da tre periodi di clock.

Il ciclo di lettura risulta molto simile al ciclo di fetch anche se le linee interessate sono unicamente RD e MREQ. Anche in questo caso è possibile inserire dei cicli di attesa in relazione allo stato della linea WAIT che potrebbe essere testata durante il fronte di discesa di T2.

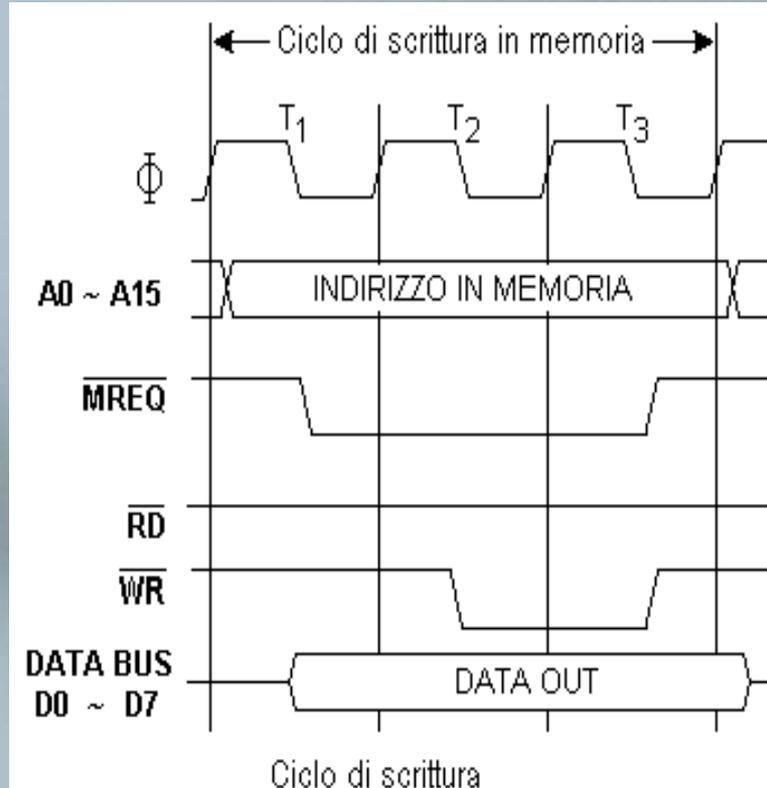
La lettura dei dati avviene sul fronte di discesa T3 per cui in questo intervallo i dati dovranno essere stabili sul bus dati.



Segnali di TEMPORIZZAZIONE

■ Ciclo di lettura e scrittura in memoria

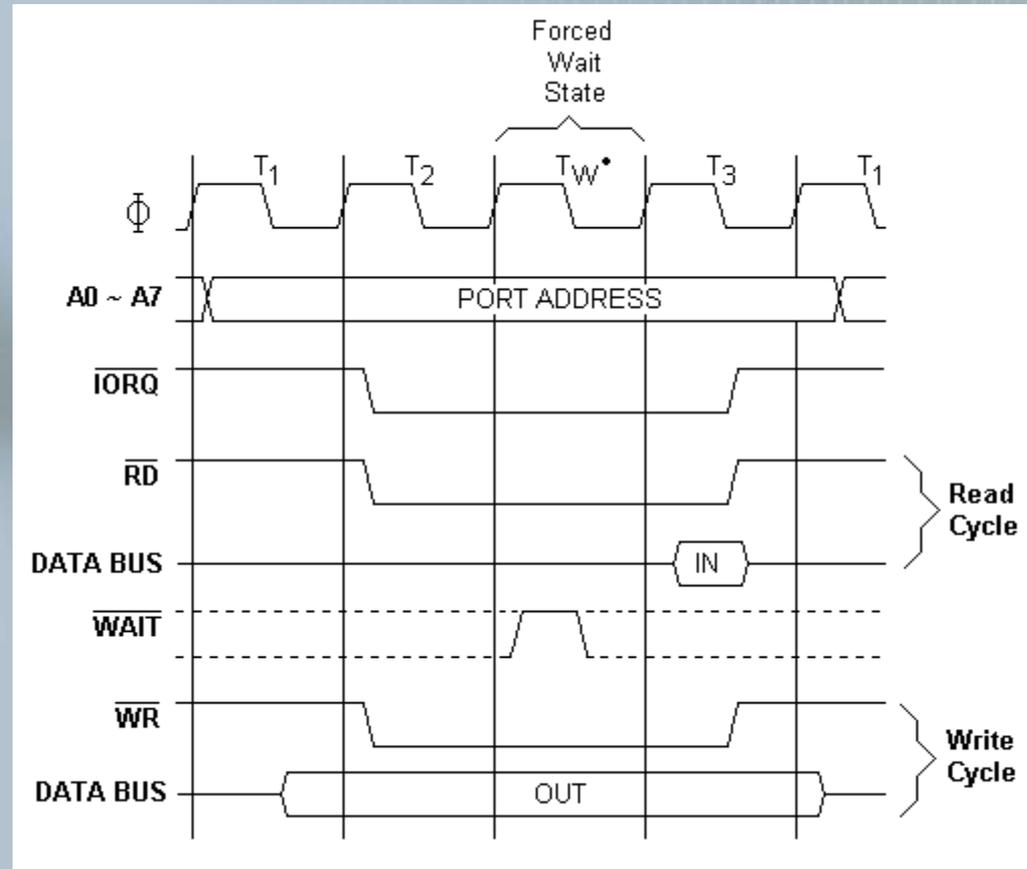
Nel ciclo di scrittura vengono invece attivate in tempi successivi le linee MREQ e WR, il segnale WR deve essere abilitato solo quando i dati presenti sul bus sono stabili per evitare errori di lettura, la disabilitazione delle linee WR e MREQ avviene circa mezzo periodo prima che i dati ed indirizzi perdano la loro stabilità



Segnali di TEMPORIZZAZIONE

Ciclo di ingresso/uscita

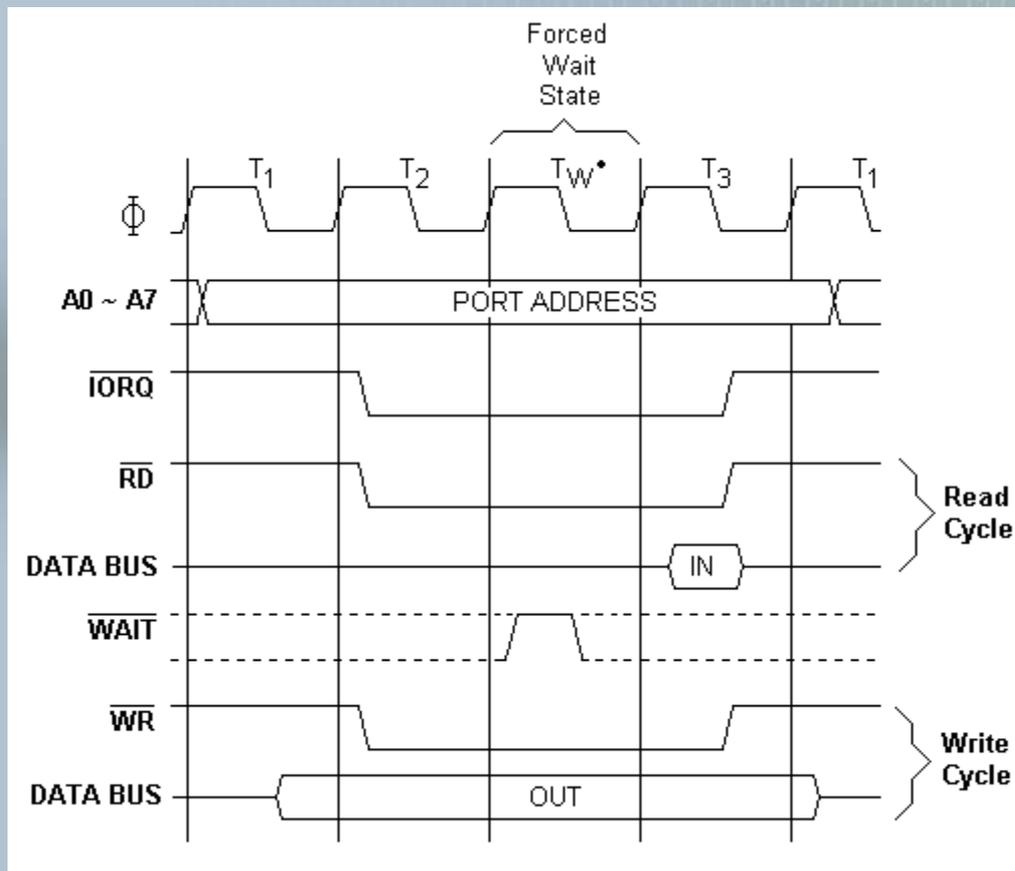
I cicli che consentono di effettuare operazioni di lettura e scrittura da un periferico sono riportati in figura e risultano molto simili ai cicli di lettura e scrittura. Nella figura seguente è mostrata un'operazione di lettura di un dispositivo di I/O ed una scrittura in un dispositivo I/O. Si noti che durante tali operazioni viene aggiunto automaticamente uno stato di attesa. La ragione di ciò è che, altrimenti, durante un'operazione di I/O, il tempo esistente da quando il segnale di IORQ* diventa attivo a quando la CPU avrebbe campionato la linea di WAIT*, sarebbe stato molto breve; quindi, senza questo stato aggiuntivo, non vi sarebbe stato tempo sufficiente per una porta di I/O di decodificare l'indirizzo ed attivare eventualmente la line di WAIT*.



Segnali di TEMPORIZZAZIONE

■ Ciclo di ingresso/uscita

I Durante questo tempo di attesa, la line di WAIT* viene campionata. Durante un'operazione di lettura di un dispositivo di I/O, la linea RD* é utilizzata per abilitare la porta indirizzata a presentare i propri dati su bus, esattamente come nel caso di una lettura in memoria. Durante le operazioni di scrittura in un dispositivo I/O, la linea WR* é utilizzata come clock per la porta di I/O, ed anche in questo caso con una temporizzazione che fornisce automaticamente una stabilit  dei dati tale da permettere l'uso del suo fronte di salita come clock dei dati.



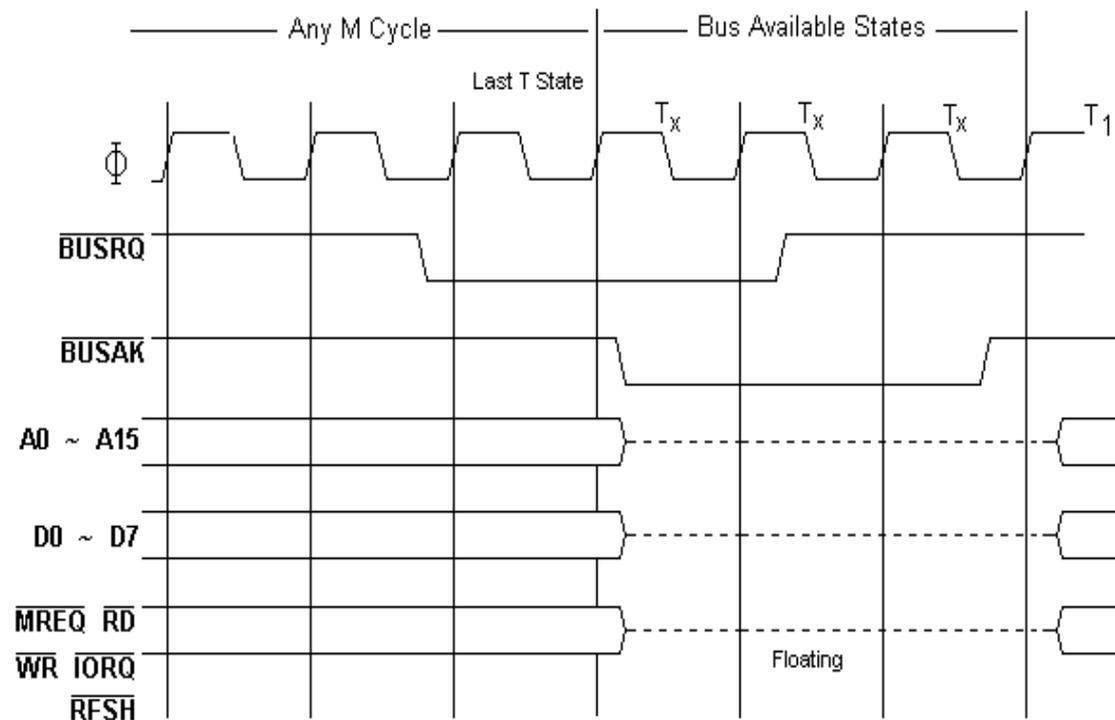
Segnali di TEMPORIZZAZIONE

Ciclo di richiesta/riconoscimento dei bus

Quando il microprocessore non è chiamato ad operare direttamente può cedere il controllo e la gestione di una parte delle sue linee a dei dispositivi esterni. Esempio costituito da operazioni DMA (Direct Access Memory) in cui la memoria scambia direttamente dati con il periferico senza l'intervento della CPU. Il segnale $\overline{\text{BUSRQ}}^*$ viene campionato dalla CPU con il fronte di salita dell'ultimo periodo di clock di ogni ciclo macchina. Se tale segnale è attivo, la CPU emette il segnale di $\overline{\text{BUSAK}}$ che segnala al periferico che ne ha fatto

richiesta
l'avvenuto

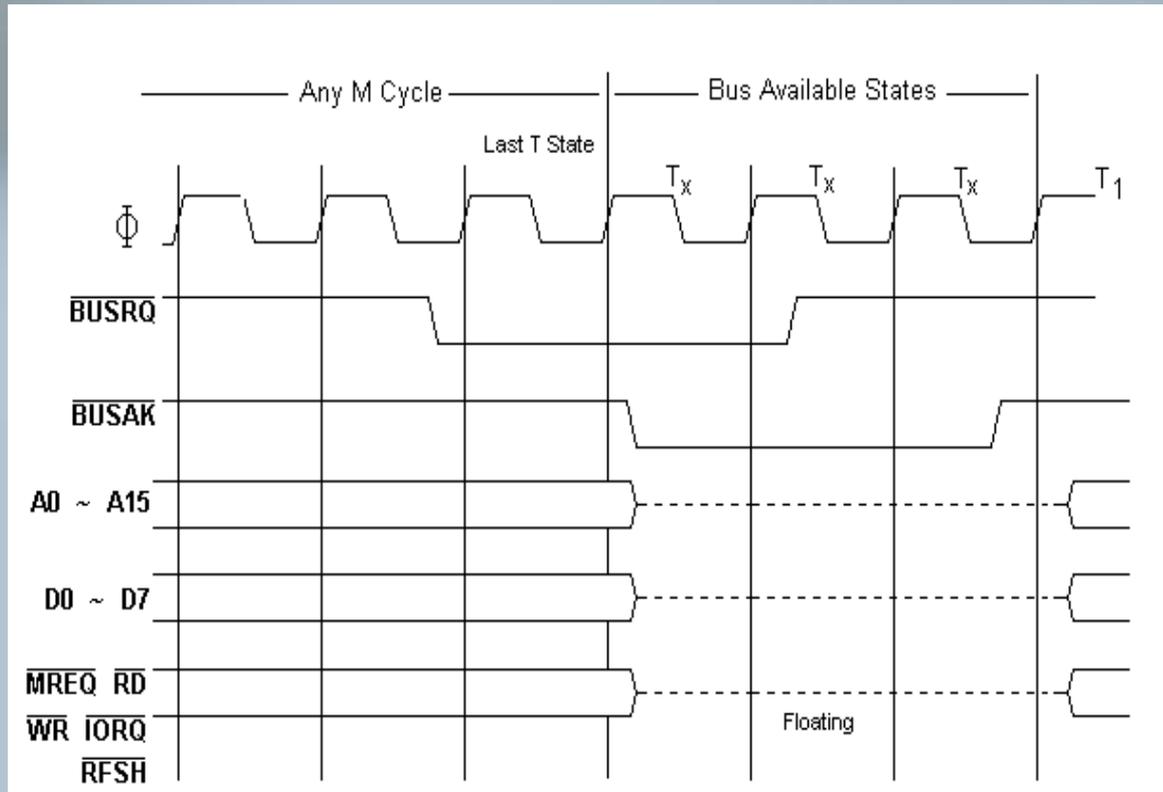
riconoscimento,
contemporaneamente
pone le proprie uscite
tri-state, in stato di alta
impedenza in
coincidenza del fronte di
salita del prossimo
impulso di clock. Da
questo momento,
qualsiasi dispositivo
esterno può controllare il
bus per trasferire dati tra
la memoria ed i
dispositivi di I/O



Segnali di TEMPORIZZAZIONE

■ Ciclo di richiesta/riconoscimento dei bus

Al termine dell'operazione per la cui esecuzione il periferico ha chiesto il servizio viene disattivata la linea BUSRQ. Siccome la CPU testa questa linea sul fronte di salita di ciascun ciclo T, quando la trova a livello alto disattiva la linea di BUSAK e riprende il controllo del sistema



Segnali di TEMPORIZZAZIONE

Richiesta di Interruzione

Durante il fronte di salita dell'ultimo ciclo T viene eseguito un test sulle linee INT e NMI e, nel caso in cui entrambe siano a livello basso, viene soddisfatta quella con priorità più elevata e cioè l'interruzione non mascherabile. Successivamente la CPU accetta la richiesta di interruzione mascherabile solo se il F-F di abilitazione delle interruzioni è settato a "1" e se non è attiva la linea di BUSRQ. E' importante notare che nel ciclo relativo all'interruzione mascherabile sono sempre presenti almeno due cicli di attesa per consentire al periferico che ha presentato la richiesta di interruzione di farsi riconoscere dalla CPU inviando sul bus dati il proprio indirizzo.

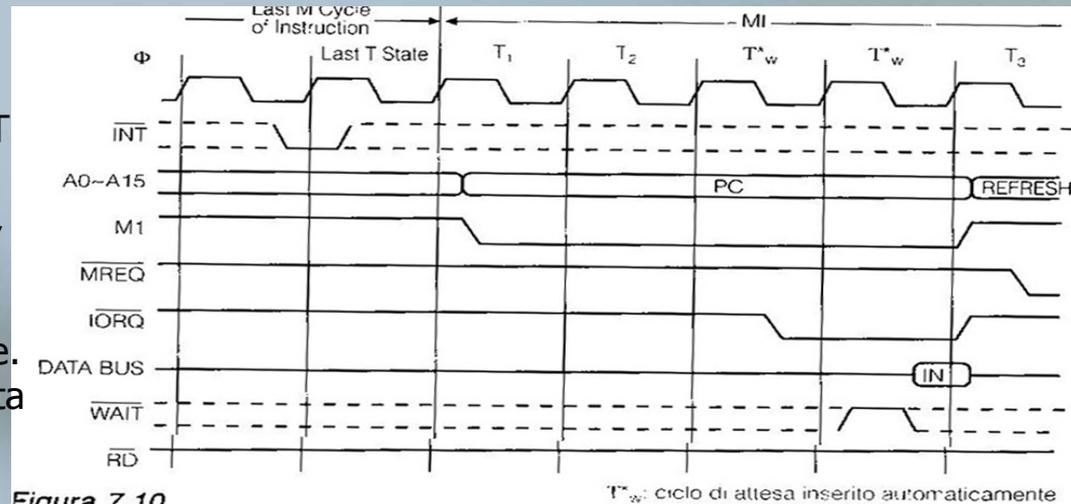


Figura 7.10

