

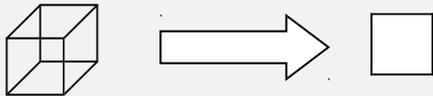
## 1.6 Modello della memoria

### Modello della memoria

In questa unità viene analizzata la memoria centrale da un punto di vista funzionale. Quindi, senza entrare nel merito delle specifiche tecnologie, viene descritto un modello sistemico a partire dal concetto di elemento di memoria fino ad arrivare al concetto di decodifica dell'indirizzo di memoria.

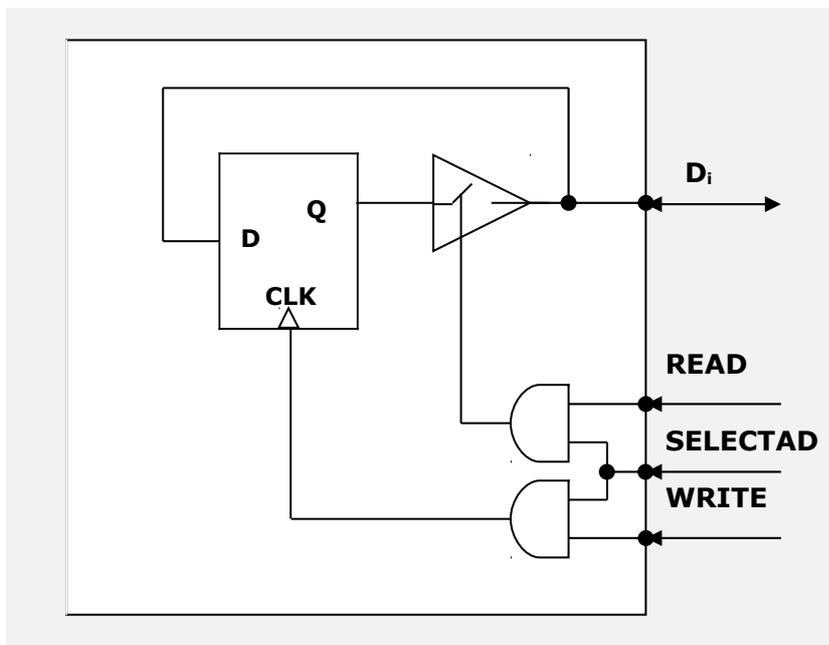
### Elemento di memoria

Il termine 'memoria' indica la capacità di ricordare una informazione. Nella tecnologia informatica (e anche nella logica) la più piccola quantità di informazione è definita informazione binaria, cioè una informazione che può assumere solo due valori. In informatica questo tipo di informazione viene chiamato 'bit' contrazione delle due parole inglesi "Binary digIT". Il più piccolo elemento di memoria è quindi un dispositivo elettronico in grado di conservare un bit. Negli schemi grafici che seguono sarà rappresentato da un piccolo quadrato il cui significato è quello di una scatola (memoria = contenitore) vista dall'alto:



Dal punto di vista elettronico lo si può immaginare come un flip-flop, che in effetti è un componente in grado di memorizzare una informazione binaria. In realtà esistono molte tecnologie diverse per realizzare le memorie (le RAM dinamiche ad esempio sono basate su fenomeni capacitivi) ma il modello flip-flop consente una semplice spiegazione.

### Modello dell'elemento di memoria



Questo modello descrive il funzionamento di un singolo elemento di memoria. In realtà le operazioni di accesso avvengono sempre a livello di Byte quindi si deve immaginare la presenza di altri sette elementi collegati agli altri fili di data bus ma comandati dallo stesso circuito di selezione.

L'elemento di memoria è descritto da un flip-flop di tipo D. Il flip-flop D (Delay=ritardo) mantiene inalterata la sua uscita Q indipendentemente dal valore di D. Il valore D viene copiato in Q quando si ha una commutazione sull'ingresso CLK.

In questo modo, quando il segnale sul data bus  $D_i$  commuta, durante le operazioni della CPU i valori variabili di  $D_i$  non influenzano lo stato dell'elemento di memoria (Q) che continua a conservare il suo valore.

Tra l'uscita Q del flip-flop ed il data bus  $D_i$  è interposto un buffer tristate che normalmente è in terzo stato (alta impedenza) in modo che lo stato dell'elemento di memoria non impegni permanentemente il filo di data bus.

Il circuito di selezione è formato da due AND che ricevono rispettivamente i segnali READ e WRITE dal control bus ed il segnale di selezione SELECT proveniente dall'ADDRESS DECODER che viene mostrato nel paragrafo successivo.

Quando è vero sia READ che SELECT (la CPU vuole leggere da questa cella) si chiude il buffer tristate e lo stato Q del flip-flop impegna il data bus e la CPU può campionare il valore.

Quando è vero sia WRITE che SELECT (la CPU vuole scrivere in questa cella) si crea una commutazione su CLK ed il valore di D (stato del data bus imposta dalla CPU) viene caricato dentro al flip-flop.

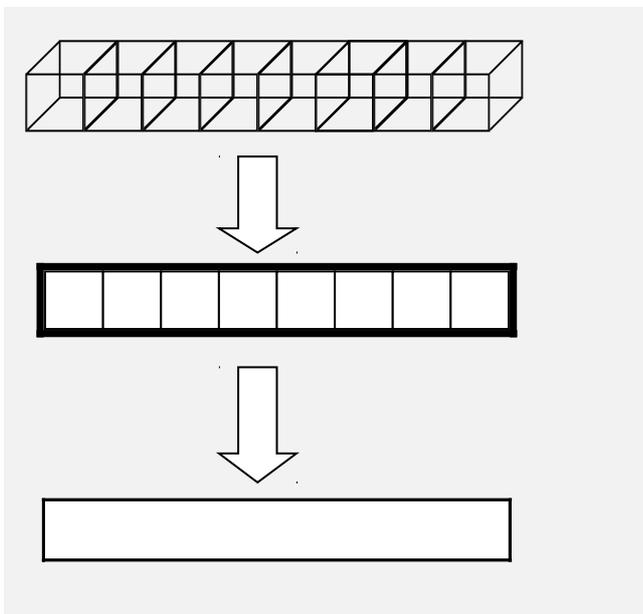
Per le regole di control bus non possono essere attivi contemporaneamente sia READ che WRITE.

Il segnale di SELECT proviene dal circuito di decodifica degli indirizzi. Questo circuito riceve in ingresso l'address bus ed il segnale di controllo MEM e decodifica il numero binario che costituisce l'indirizzo generando il segnale di selezione per la cella che ha l'indirizzo corrispondente.

### Cella di memoria

L'elemento di memoria non è accessibile singolarmente ma solo come parte di un blocco di elementi che vengono processati sempre insieme.

Il più piccolo blocco di memoria accessibile singolarmente è formato da otto bit (BYTE) ed è rappresentato simbolicamente da un rettangolo suddiviso in otto quadrati (i bit).



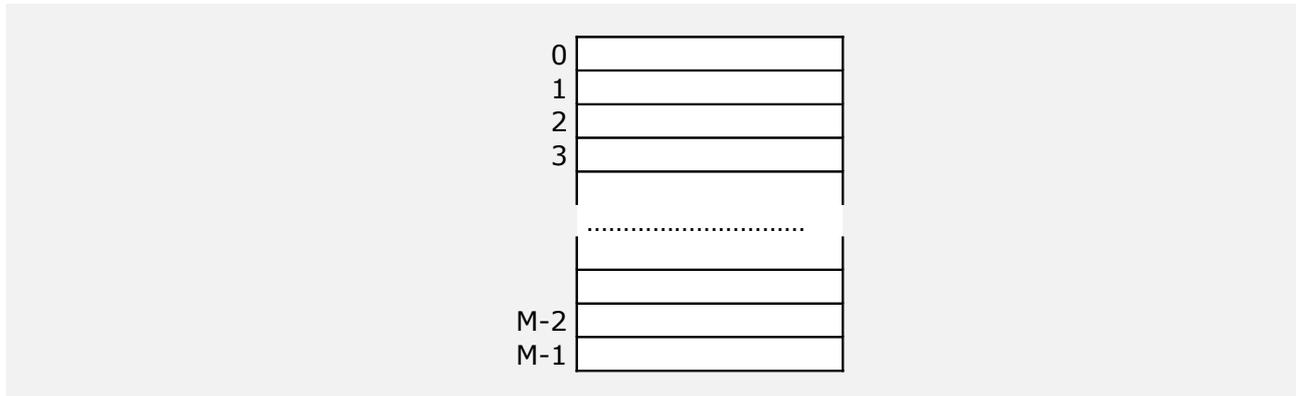
Ogni bit di una cella è collegato ad un diverso filo di data bus in modo che tutti i bit siano trasferiti contemporaneamente sul data bus. Quando la cella viene letta ogni elemento impone il suo contenuto sul rispettivo filo di data bus. Quando la cella viene scritta ogni elemento carica al suo interno il valore del rispettivo filo di data bus.

### Indirizzo della cella di memoria

Affinché ogni cella possa essere selezionata individualmente ad ogni cella deve essere associato un indirizzo che la identifichi univocamente.

Il modello descrittivo del meccanismo di indirizzamento è una schiera di celle di memorie poste una dopo l'altra. L'indirizzo di ciascuna cella è definito dalla posizione relativa della cella rispetto alla prima cella (si può anche definire come lo spostamento necessario per raggiungerla partendo dalla prima, quindi la prima ha indirizzo 0, la seconda 1 e così via ...).

Lo schema grafico è il seguente:



Il numero massimo di celle che possono essere presenti teoricamente in un sistema viene chiamato **"spazio di indirizzamento"** o **"spazio di memoria"** e il suo valore dipende dalle caratteristiche della CPU e si ottiene a partire dal numero di fili dell'address bus individuando tutte le possibili "disposizioni con ripetizione" di due elementi (0 e 1) N a N.

$$M = D^r_{2,N} = 2^N$$

Quindi se M è lo spazio di memoria ottenuto con N fili di address bus gli indirizzi di memoria vanno da 0 a M-1.

La quantità di celle presenti in memoria centrale si misura in byte (B) o suoi multipli (attenzione però i multipli sono potenze del 2):

1 KB = 1 KiloByte	=	1024 B
1 MB = 1 MegaByte	= 1024 KB =	1.048.576 B
1 GB = 1 GigaByte	= 1024 MB = 1024*1024 KB	= 1.073.741.824 B
1 TB = 1 TeraByte	= 1024 GB = 1024*1024*1024 KB	= 18.446.744.073.709.551.616 B
1 PB = 1 PetaByte	= 1024 GB ...	
1 EB = 1 ExaByte	= 1024 PB ...	

Spazi di indirizzamento tipici sono:

- 16 bit → 64KB
- 32 bit → 4GB
- 64 bit → 16EB

Note:

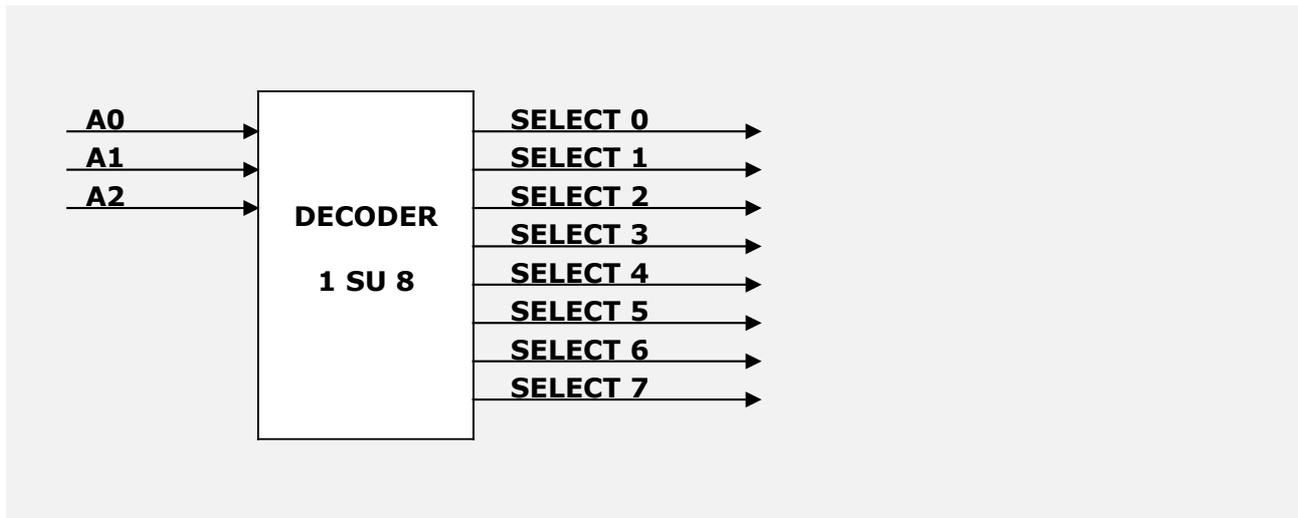
- Il numero definito dallo spazio di indirizzamento determina il numero teorico massimo di celle o porte che possono essere raggiunte dalla CPU e non il numero effettivo di elementi che è sempre notevolmente minore.
- Non deve trarre in inganno la sproporzione tra valori teorici e valori reali (un PC domestico a 64 bit può indirizzare 16 ExaByte di memoria ma in genere non monta più di alcuni GB di memoria fisica. Il software trae vantaggio dal maggior spazio di indirizzamento, a prescindere dalla effettiva memoria presente grazie ad un meccanismo denominato "memoria virtuale").

**Modello del circuito di decodifica**

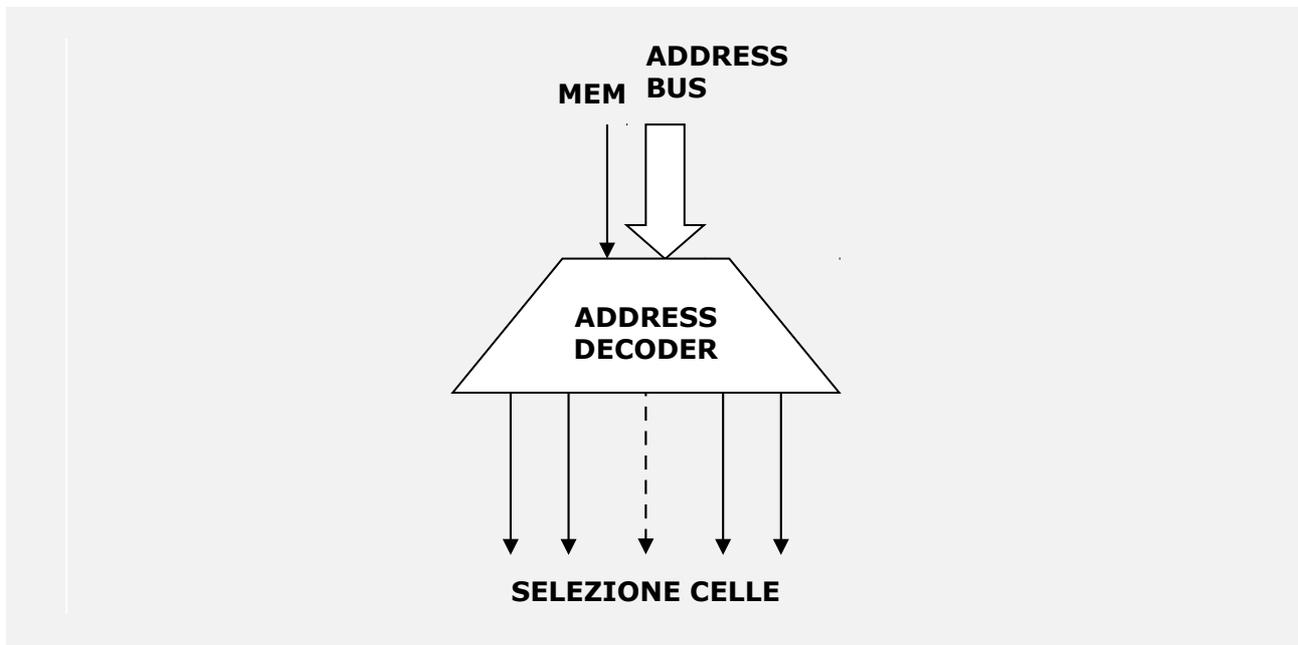
L'indirizzo della cella da selezionare viene fornito dalla CPU in forma codificata in binario sull'address bus.

E' necessario quindi nel blocco di memoria un circuito di decodifica che traduca la codifica binaria in un segnale di selezione per la cella coinvolta nell'operazione (il segnale SELECT dello schema precedente)

Da un punto di vista di principio questo circuito ha un funzionamento analogo ai circuiti di decodifica già visti in elettronica. Nell'esempio che segue viene mostrata una decodifica a tre bit che produce otto segnali di selezione ( $M=2^3=8$  quindi indirizzi da 0 a 7)



Il decoder degli indirizzi di un sistema a microprocessore deve decodificare un gran numero di indirizzi ed essere sottoposto alla abilitazione del segnale di controllo MEM perché il bus degli indirizzi è condiviso con gli I/O e il segnale MEM ha il compito di abilitazione della decodifica.



**Decodifica a più livelli**

In realtà il circuito di decodifica è integrato nei componenti di memoria e quindi in funzione della composizione della memoria è possibile che la decodifica si presenti a più livelli.

Consideriamo un esempio realizzato con numeri semplici.

Supponiamo che uno spazio di memoria totale di 64KB (16 bit di indirizzo) sia parzialmente occupato da 8 componenti di 4KB per un totale di 32KB di memoria effettivamente presente (indirizzi da 0 a 32767).

Un componente di memoria da 4KB contiene 4096 celle ed è dotato di un circuito di decodifica integrato in grado di selezionarli quindi il componente deve avere 12 ingressi  $A_i$  ( $A_0$  ad  $A_{11}$ ).

Tutti i componenti gli otto componenti sono uguali e quindi tutti decodificherebbero lo stesso campo di indirizzi ( da 0 a 4095) usando solo i primi dodici bit di indirizzo e ignorando i rimanenti quattro quindi sovrapponendosi sullo stesso spazio di memoria e ripetendo la selezione per indirizzi multipli di 4K.

Ad esempio se mando l'indirizzo 0 si seleziona la prima cella di tutti i componenti ma questa prima cella si seleziona anche se mando l'indirizzo 4096, 8192 e così via ...

Per comprendere meglio il fenomeno è opportuno analizzare l'indirizzo nella sua codifica binaria piuttosto che decimale:

Indirizzo decimale	Indirizzo in binario a 16 bit	Commento
0	0000000000000000	Prima cella del primo comp.
1	0000000000000001	Seconda cella del primo comp.
4095	0000111111111111	Ultima cella del primo comp.
4096	0001000000000000	Prima cella del secondo comp.
4097	0001000000000001	Seconda cella del secondo comp.
8191	0001111111111111	Ultima cella del secondo comp.
8192	0010000000000000	Prima cella del terzo comp.

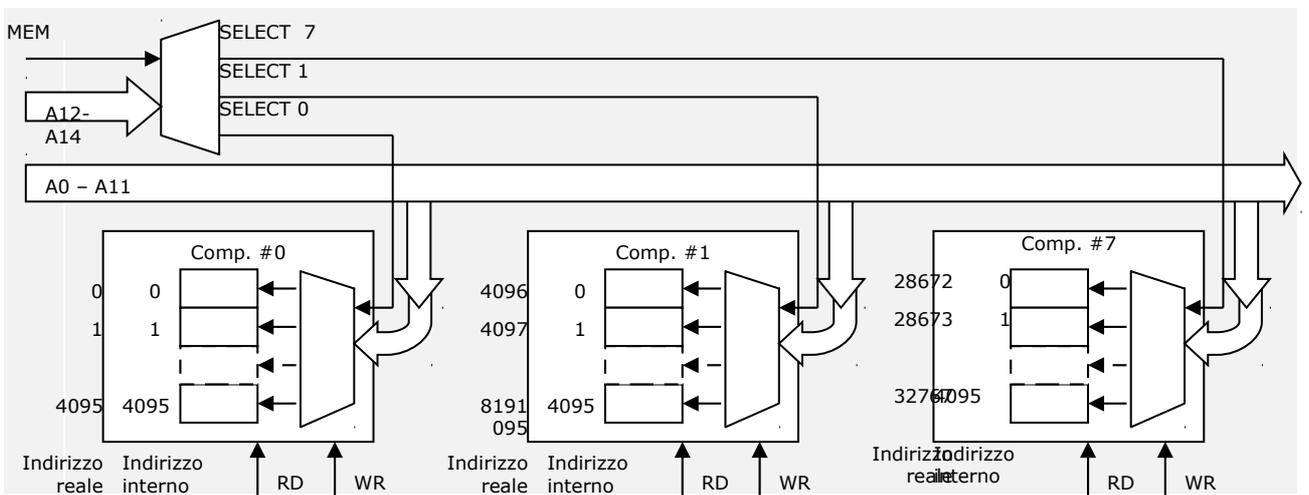
Per un componente che riceve solo i primi dodici fili di Address bus l'indirizzo 0 , 4096 e 8192 coincidono (parte in grassetto) perché la parte alta dell'indirizzo (corsivo) è ignorata.

Ogni componente è però dotato di un ingresso di selezione (CHIP SELECT) che abilita/disabilita l'intero componente indipendentemente dall'indirizzo che riceve.

Si può usare questo segnale di controllo per "traslare" gli indirizzi di un componente nello spazio di memoria. Infatti mandando tre dei quattro bit più significativi (l'ultimo non serve perché l'esempio prevede di mappare solo metà dello spazio teorico) ad un decoder si può discriminare se la parte bassa dell'indirizzo appartiene al primo, secondo ... n-esimo banco di memoria, cioè se è compreso tra gli indirizzi *nnn*000000000000 e *nnn*111111111111.

Mandando al CHIP SELECT di ciascun componente il risultato della decodifica di un diverso valore *nnn* il componente reagisce solo a 4K di indirizzi che hanno come parte alta il corrispondente valore *nnn*.

In questo esempio il 16-esimo bit di indirizzo non viene utilizzato.



**Relazione tra indirizzamento ed ampiezza di parola**

Come abbiamo visto il più piccolo blocco di memoria accessibile singolarmente è il Byte. Questo significa che ogni Byte di memoria (cella) deve avere un distinto indirizzo.

Da questa situazione deriva anche il modo di indicare le dimensioni dei componenti di memoria che sono espresse in quantità di Byte.

Ogni sistema a microprocessore però è caratterizzato da una propria "ampiezza di parola" che è definita come il numero di bit che possono essere trasferiti sul data bus in un'unica operazione.

Se l'ampiezza di parola supera il Byte (otto bit) la CPU è in grado di leggere e scrivere dalla memoria più di una cella contemporaneamente.

Ad esempio un Pentium 4 ha una ampiezza di parola di 64 bit (otto bytes) e quindi è in grado di leggere e scrivere 8 celle di memoria contemporaneamente.

Quindi un Pentium 4 ha il data bus composto dai fili D<sub>0</sub>, D<sub>1</sub>, ... D<sub>63</sub>.

Sui fili D<sub>0</sub>-D<sub>7</sub> si affacciano le celle di indirizzo 0, 8,16,24,32 ...

Sui fili D<sub>8</sub>-D<sub>15</sub> si affacciano le celle di indirizzo 1, 9,17,25,33 ...

Sui fili D<sub>16</sub>-D<sub>23</sub> si affacciano le celle di indirizzo 2,10,18,26,34 ...

Sui fili D<sub>24</sub>-D<sub>31</sub> si affacciano le celle di indirizzo 3,11,19,27,35 ...

Sui fili D<sub>32</sub>-D<sub>39</sub> si affacciano le celle di indirizzo 4,12,20,28,36 ...

Sui fili D<sub>40</sub>-D<sub>47</sub> si affacciano le celle di indirizzo 5,13,21,29,37 ...

Sui fili D<sub>48</sub>-D<sub>55</sub> si affacciano le celle di indirizzo 6,14,22,30,38 ...

Sui fili D<sub>56</sub>-D<sub>63</sub> si affacciano le celle di indirizzo 7,15,23,31,39 ...

Per motivi di efficienza in ogni operazione di data bus viene sempre trasferito l'intero data bus questo significa che indipendentemente dalla cella su cui effettivamente si deve fare l'operazione (cioè anche se l'operazione coinvolge un solo byte) ne vengono trasferite 8 sempre a partire da un indirizzo multiplo di 8 (quindi indirizzi 0-7, 8-15, 16-23 ...). La cella interessata si deve trovare all'interno di questo blocco di memoria.

Naturalmente la memoria deve essere strutturata in modo da presentare le sue celle a blocchi di 8 sul data bus di 64 bit (I moduli DIMM sono proprio strutturati in questo modo)

Poiché la CPU legge e scrive sempre blocchi di otto celle ad indirizzi consecutivi non è necessario specificare i primi 3 bit di indirizzo (A<sub>0</sub>,A<sub>1</sub> e A<sub>2</sub>) che in effetti nella architettura del Pentium 4 non esistono (il bus degli indirizzi parte da A<sub>3</sub> ad A<sub>35</sub>).

La seguente tabella spiega il comportamento ricorrendo ancora alla codifica binaria dell'indirizzo.

Indirizzo decimale	Indirizzo in binario a 36 bit	Commento
0	000000000000000000000000000000000000	Tutte le celle di questo blocco hanno lo stesso indirizzo nella porzione A <sub>3</sub> -A <sub>35</sub> (0) e quindi vengono selezionate insieme. Gli ultimi tre bit sono condizioni di indifferenza xxx
1	000000000000000000000000000000000001	
2	000000000000000000000000000000000010	
3	000000000000000000000000000000000011	
4	000000000000000000000000000000000100	
5	000000000000000000000000000000000101	
6	000000000000000000000000000000000110	
7	000000000000000000000000000000000111	Tutte le celle di questo blocco hanno lo stesso indirizzo nella porzione A <sub>3</sub> -A <sub>35</sub> (1) e quindi vengono selezionate insieme. Gli ultimi tre bit sono condizioni di indifferenza xxx
8	000000000000000000000000000000001000	
9	000000000000000000000000000000001001	
10	000000000000000000000000000000001010	
11	000000000000000000000000000000001011	
12	000000000000000000000000000000001100	
13	000000000000000000000000000000001101	
14	000000000000000000000000000000001110	
15	000000000000000000000000000000001111	Blocco 2
16	000000000000000000000000000000010000	

Si deve notare che il fatto che vengano trattati otto byte alla volta non vuole dire che non esistono più gli indirizzi intermedi (cioè che le celle diventano larghe 64 bit ed assumono indirizzi consecutivi). Infatti il software deve comunque essere in grado di accedere singolarmente alle celle di dimensioni di un byte (si pensi al tipo char) che però per motivi di efficienza viene sempre trasferito insieme ad altri byte adiacenti. Per questo motivo nella piedinatura del Pentium 4 viene conservata la denominazione a partire da A<sub>3</sub> sebbene non esistano i bit di peso più basso.