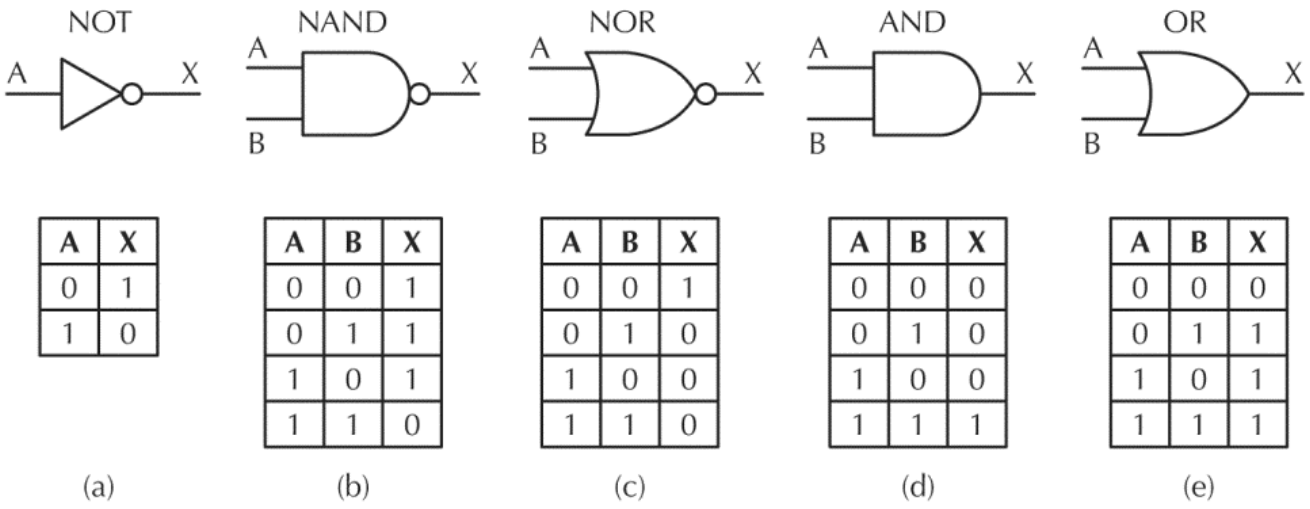
Livello Logico-Digitale

Prof. Ing. Donato Impedovo

Porte Logiche



A	X
0	1
1	0

(a)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

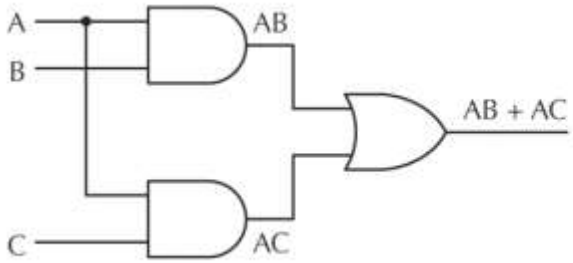
(d)

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)

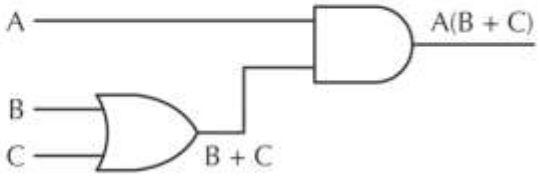
Figura 3.2 Simboli e comportamenti funzionali di cinque porte logiche elementari.

Funzioni



A	B	C	AB	AC	AB + AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

(a)



A	B	C	A	B + C	A(B + C)
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

(b)

Figura 3.5 Due funzioni equivalenti. (a) *AB + AC*. (b) *A (B + C)*.

Proprietà

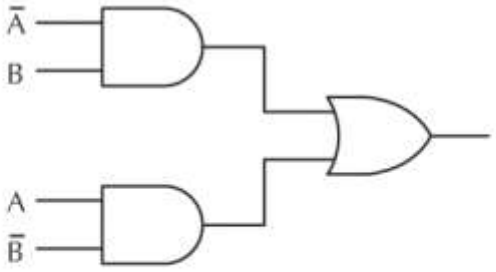
Nome	Forma AND	Forma OR
Elemento neutro	$1A = A$	$0 + A = A$
Assorbimento	$0A = 0$	$1 + A = 1$
Idempotenza	$AA = A$	$A + A = A$
Complementazione	$A\overline{A} = 0$	$A + \overline{A} = 1$
Proprietà commutativa	$AB = BA$	$A + B = B + A$
Proprietà associativa	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Proprietà distributiva	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Legge di assorbimento	$A(A + B) = A$	$A + AB = A$
Legge di De Morgan	$\overline{AB} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{AB}$

Figura 3.6 Identità dell'algebra booleana.

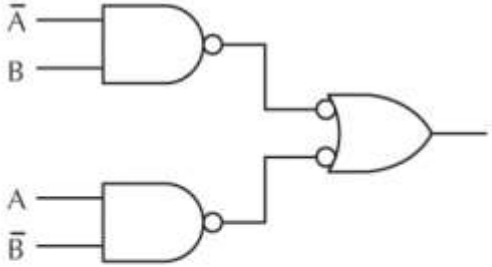
XOR

A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

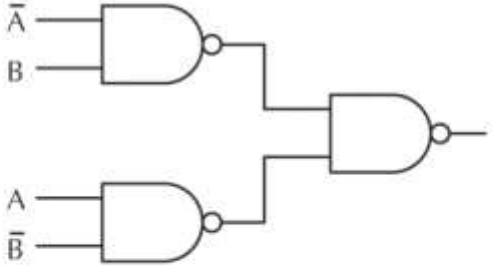
(a)



(b)



(c)



(d)

Figura 3.8 (a) Tabella di verità della funzione XOR. (b)-(d) Tre circuiti per calcolarla.

Usage

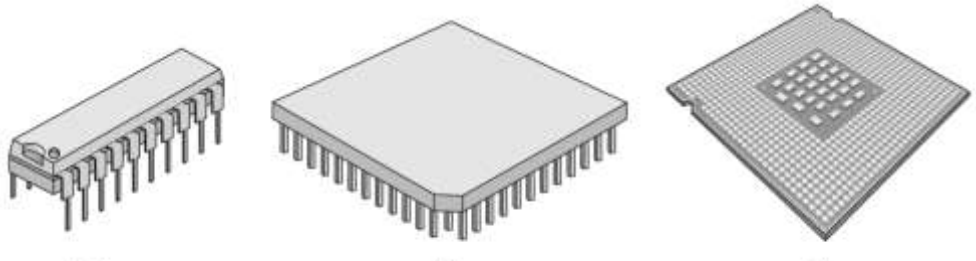


Figura 3.10 Tipologie di supporti per circuiti integrati: DIP (a), PGA (b), LGA (c).

DIP: Dual Inline Package
PGA: Pin Grid Arrays
LGA: Land Grid Arrays

Multiplexer

In funzione del valore sulle linee A,B e C solo una delle otto linee di entrata viene trasmessa in uscita.

2^n valore input
1 valore output
n valore controllo

A	B	C	D_i	F
0	0	0	D_0	D_0
0	0	1	D_1	D_1
0	1	0	D_2	D_2
0	1	1	D_3	D_3
1	0	0	D_4	D_4
1	0	1	D_5	D_5
1	1	0	D_6	D_6
1	1	1	D_7	D_7

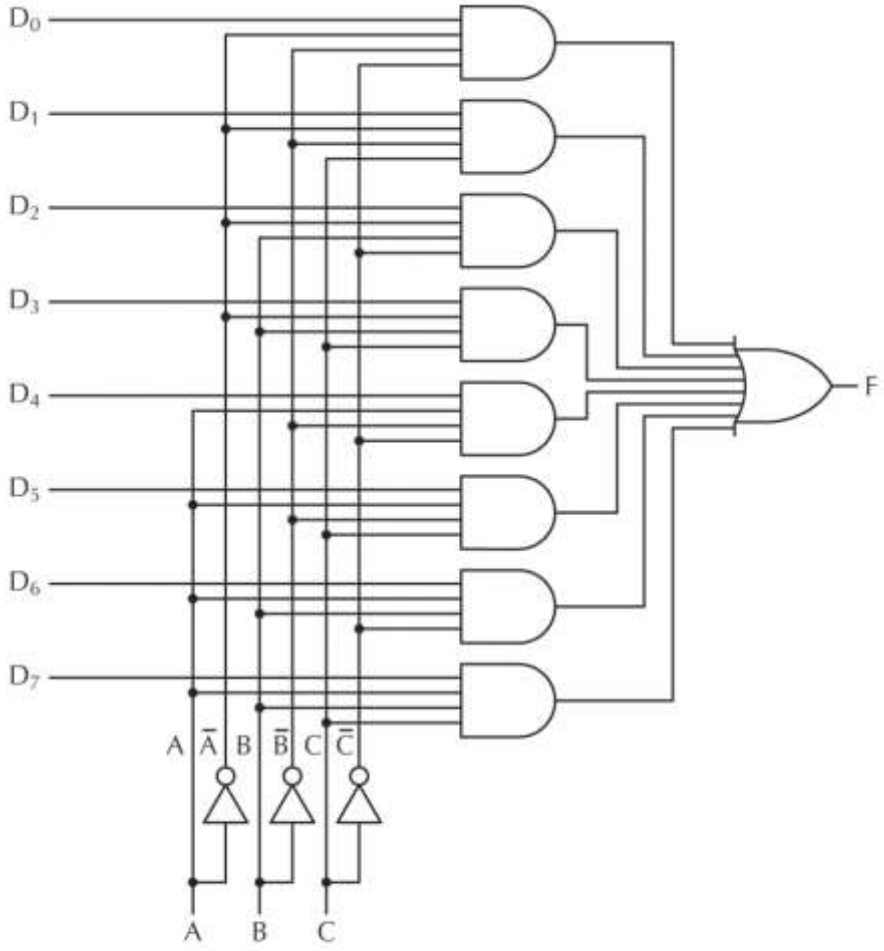


Figura 3.11 Multiplexer a otto vie.

Multiplexer

Usando un contatore e abilitando in uscita le diverse entrate in sequenza, un byte da parallelo può essere trasmesso in serie: conversione parallelo/serie.

Il multiplexer può essere utilizzato per trasformare il byte corrispondente al carattere di un tasto pigiato in una sequenza di bit, che attraverso echo può essere inviato ad uno schermo.

Il contrario del multiplexer è il demultiplexer che indirizza un segnale in ingresso verso una delle otto uscite selezionata in base al segnale di controllo sulle linee A,B,C., Viene utilizzato per trasformare un segnale da seriale a parallelo.

Decodificatore

n valore input (controllo)
2^n valore output

- Il circuito decodificatore abilita una sola delle 2^n uscite quando eccitato dalle n entrate, esattamente quella corrispondente all'indirizzo dell'entrata.

- E' il circuito utilizzato per il processing delle istruzioni dell'ISA durante il ciclo di **fetch**.

Decodificatore

3 valore input (controllo)
$8=2^3$ valore output

A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

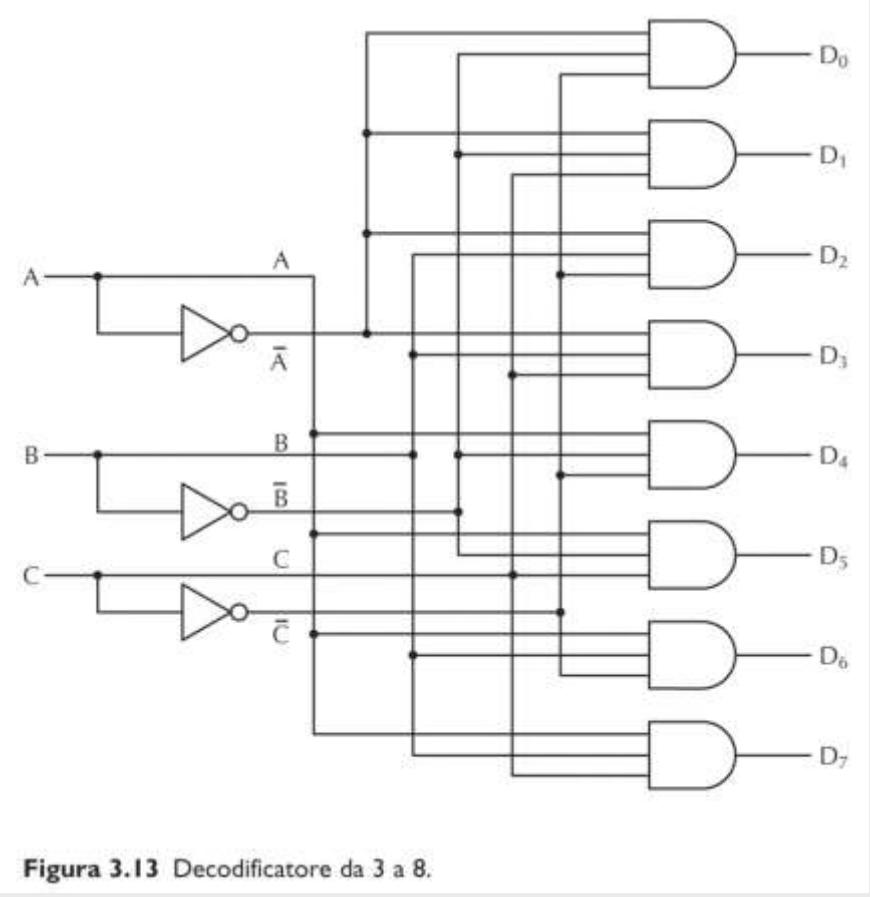
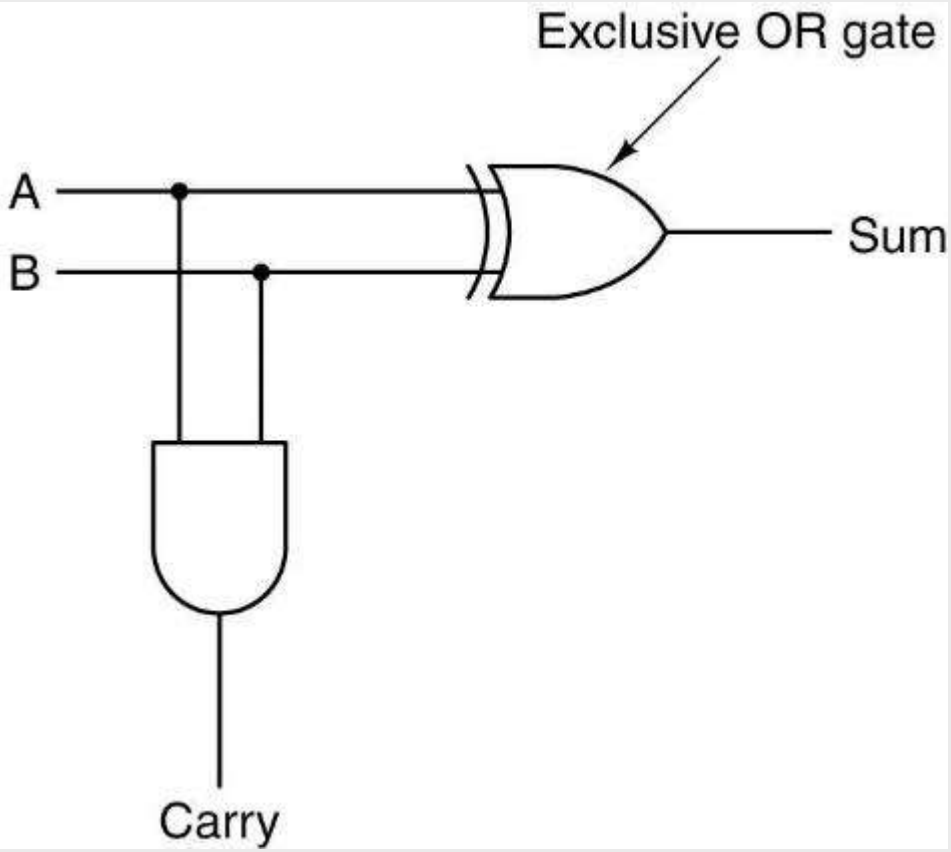


Figura 3.13 Decodificatore da 3 a 8.

Addizionatore ad un bit
HALF ADDER

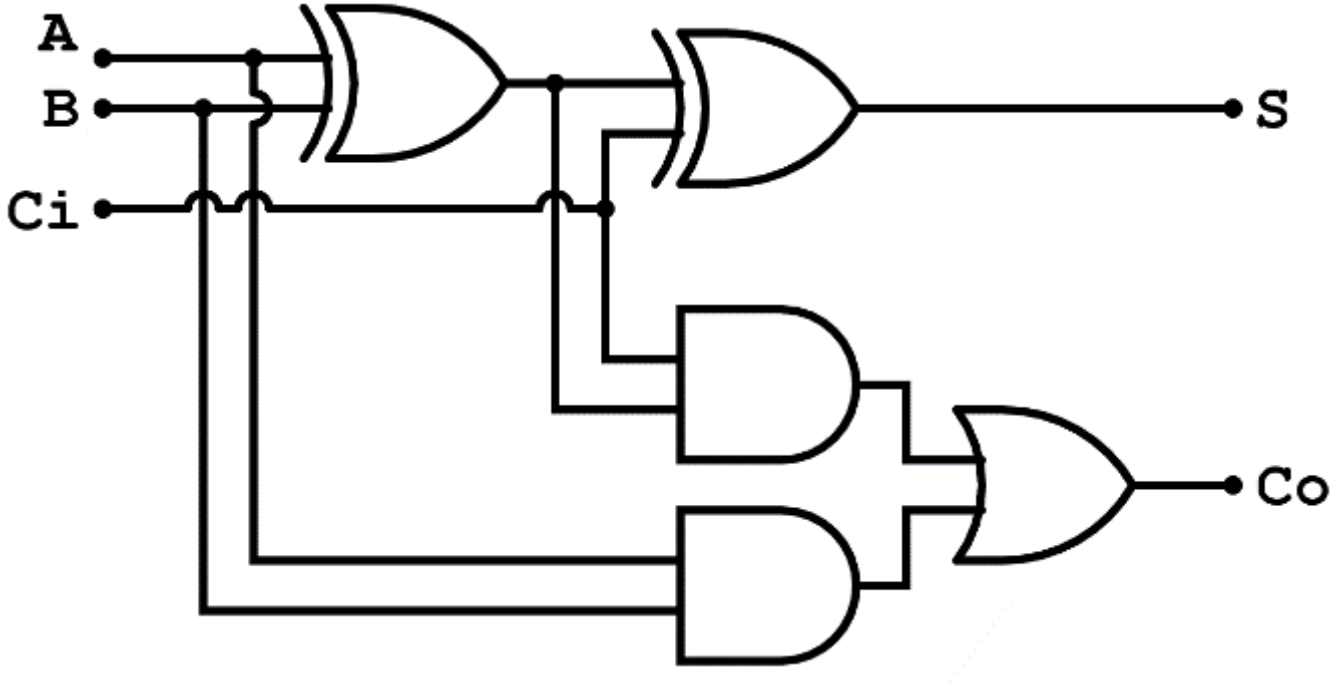
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



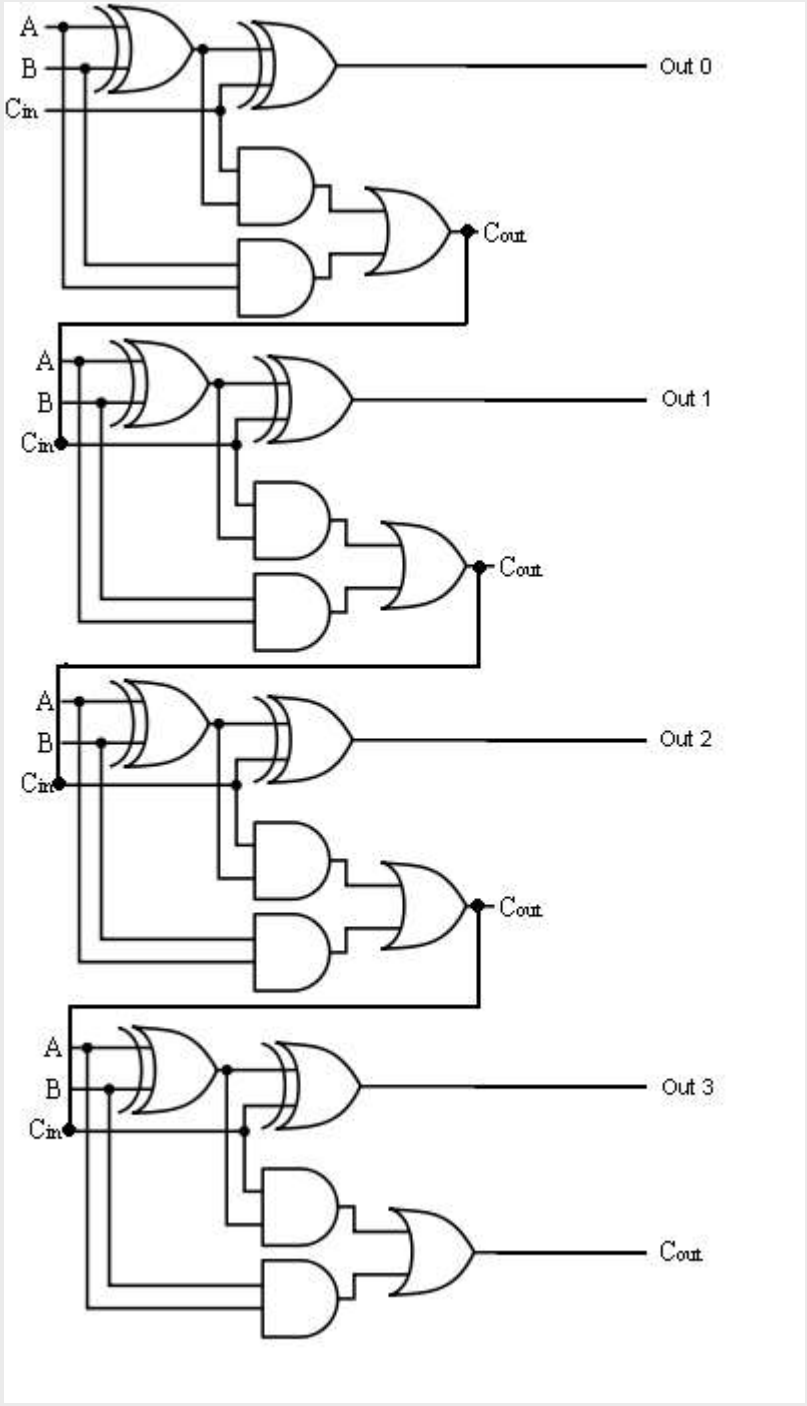
Esercizio: si disegni il circuito di un addizionatore a 2 bit (full adder) per sequenze più lunghe di un bit

FULL ADDER

Inputs			Outputs	
A	B	C_i	C_o	S
0	0	0	0	0
1	0	0	0	1
0	1	0	0	1
1	1	0	1	0
0	0	1	0	1
1	0	1	1	0
0	1	1	1	0
1	1	1	1	1



FULL ADDER A 4 bit



ALU a 1 bit

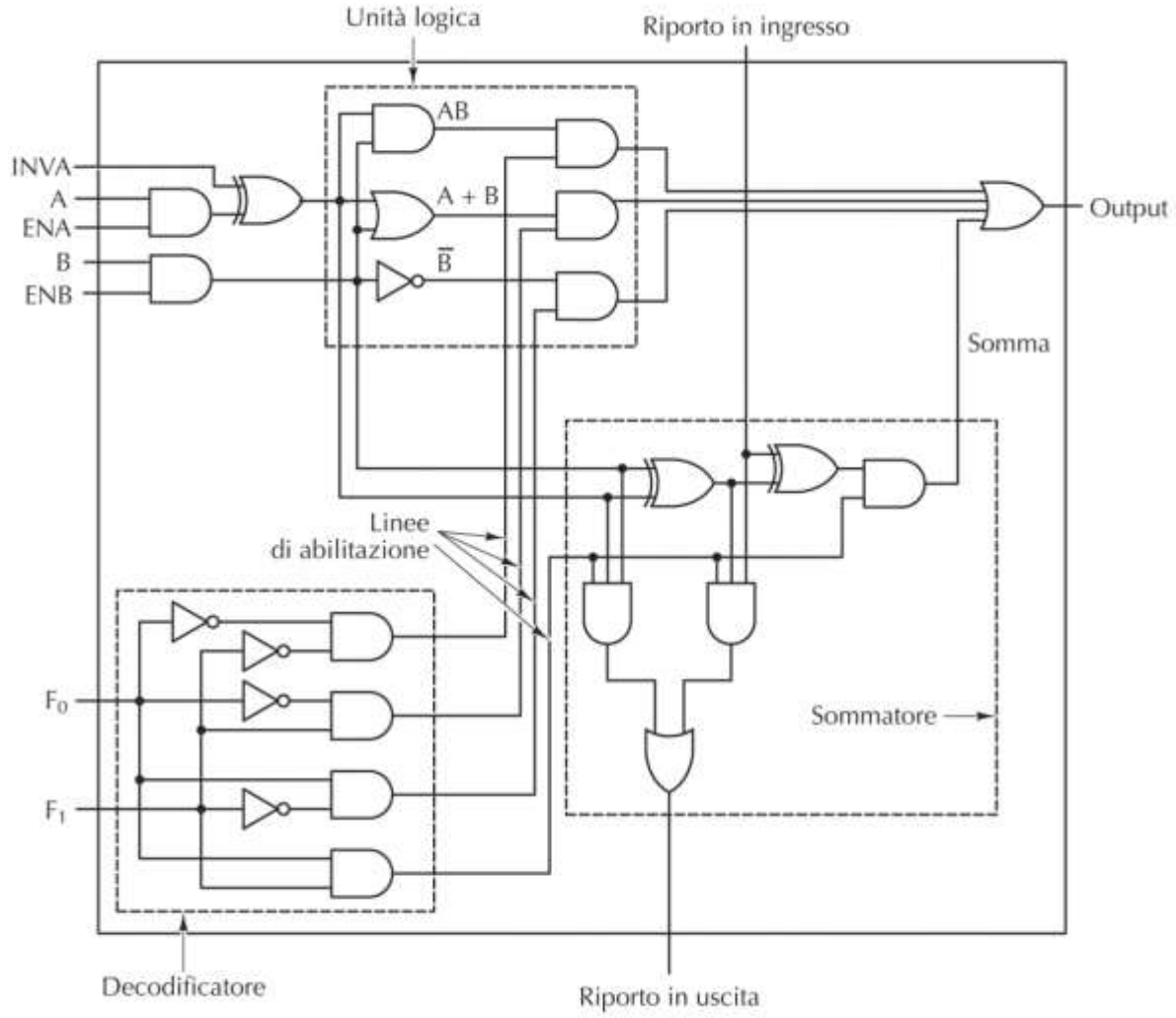


Figura 3.18 ALU a 1 bit.

ALU a 8 bit

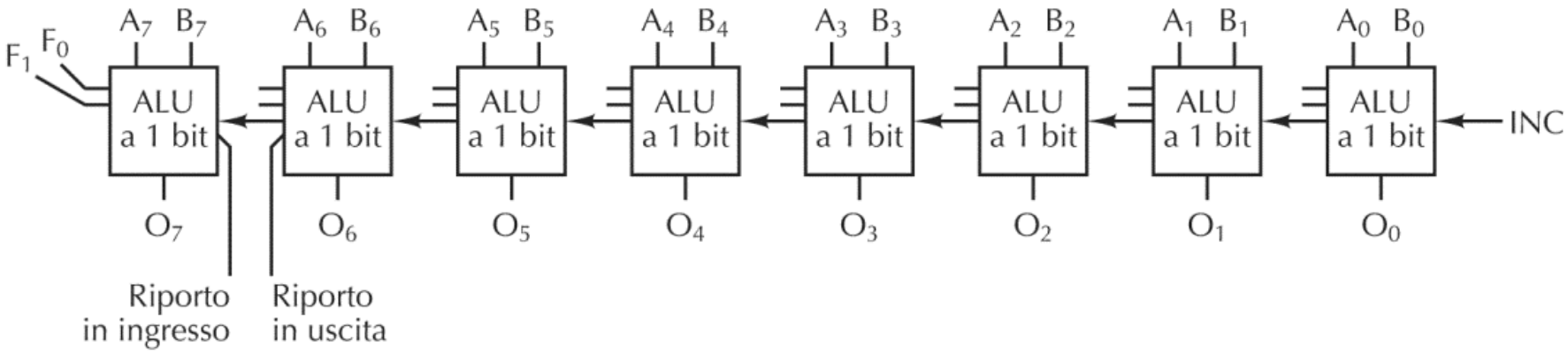
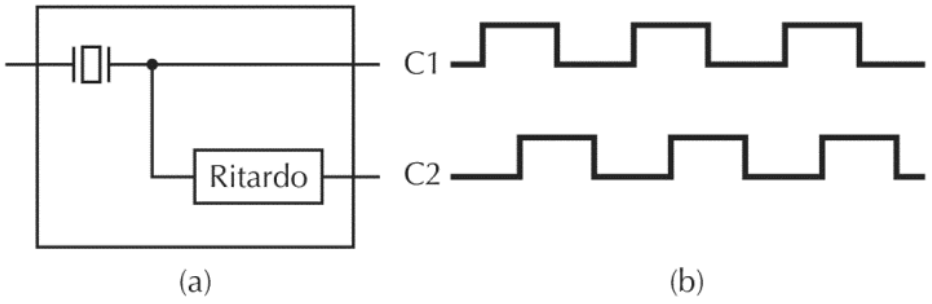


Figura 3.19 Otto ALU a 1 bit connesse per comporre una ALU a 8 bit. Per semplificare non sono mostrati segnali di abilitazione e segnali di inversione.

Clock



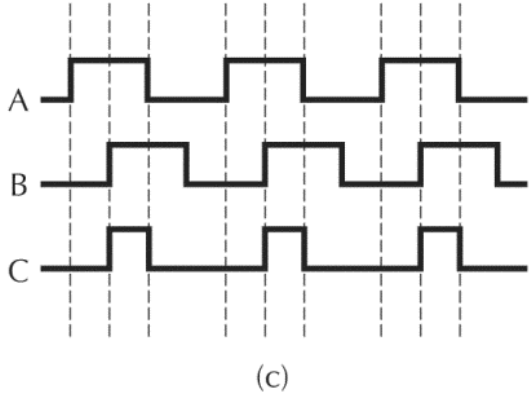


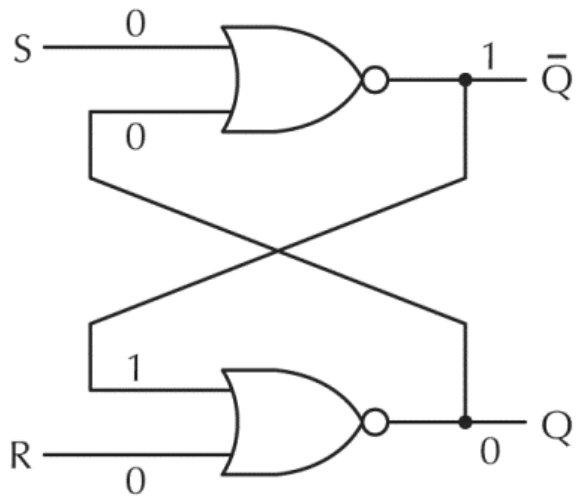
Figura 3.20 (a) Clock. (b) Diagramma di temporizzazione del clock. (c) Generazione di un clock asimmetrico.

Latch SR

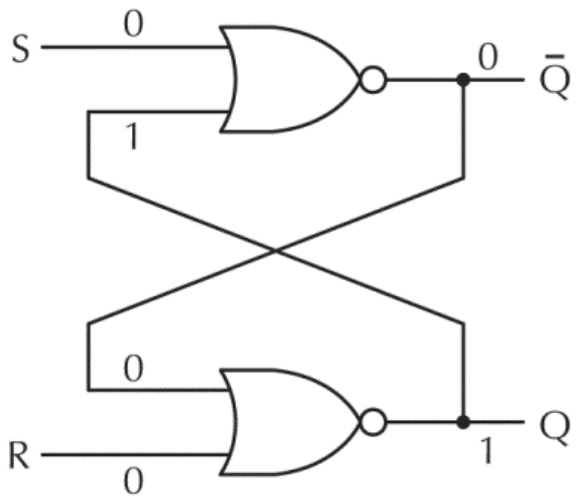
Un latch è un circuito con capacità di memoria
S = setting
R= resetting
a) Nello stato 0
b) Nello stato 1

BISTABILE!

S	R	Funzione
0	0	Latch (Memorizzazione)
0	1	Reset: Q = 0 , !Q = 1
1	0	Set: Q = 1 , !Q = 0
1	1	Non è ammesso



(a)



(b)

A	B	NOR
0	0	1
0	1	0
1	0	0
1	1	0

(c)

Figura 3.21 (a) Latch di tipo NOR nello stato 0. (b) Latch di tipo NOR nello stato 1. (c) Tabella di verità del NOR.

Latch SR temporizzato

Vogliamo che il cambio di valore avvenga solo in determinati istanti
S = setting
R= resetting
Il clock vale 0, viene impostato ad 1 quando si vuole modificare il contenuto della cella di memoria

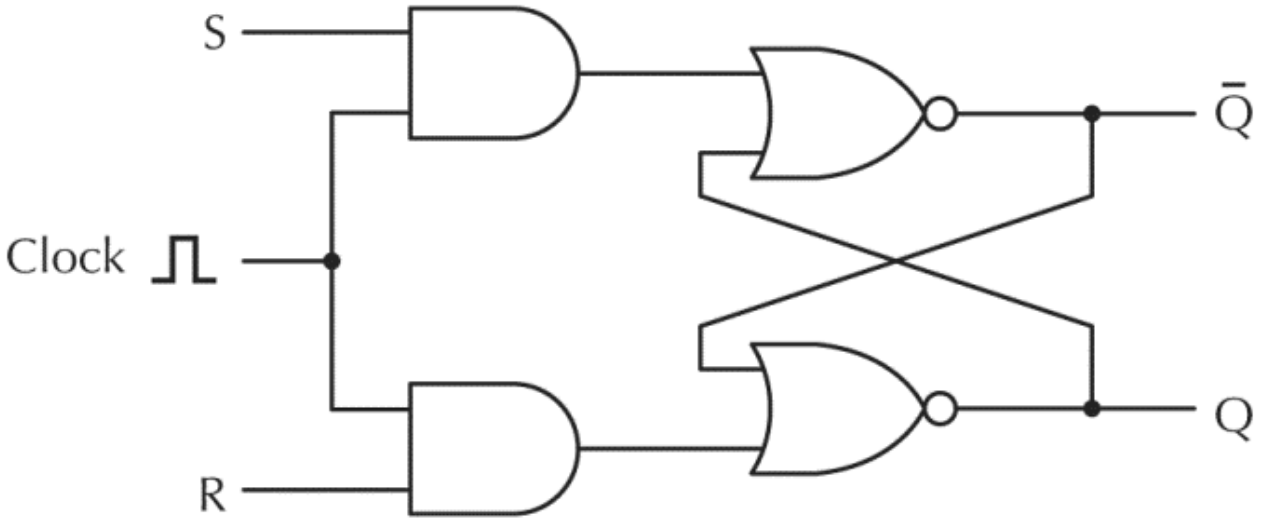


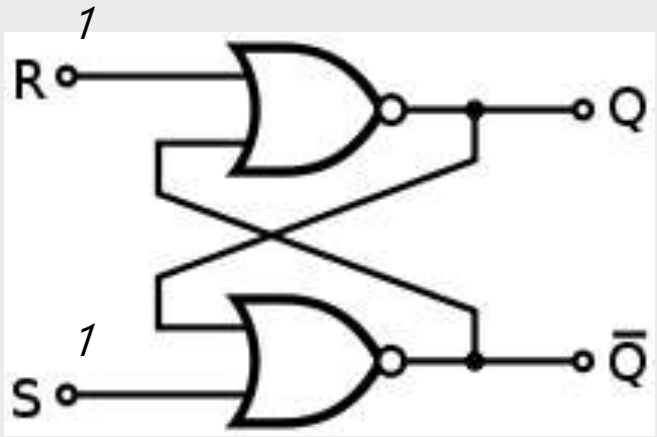
Figura 3.22 Latch SR temporizzato.

Ambiguità del Latch SR

Cosa accade se S=R=1???

NB: il latch è bistabile

S	R	Funzione
0	0	Latch (Memorizzazione)
0	1	Reset: Q = 0 , !Q = 1
1	0	Set: Q = 1 , !Q = 0
1	1	Non è ammesso



Latch D

Utilizzo di un solo input

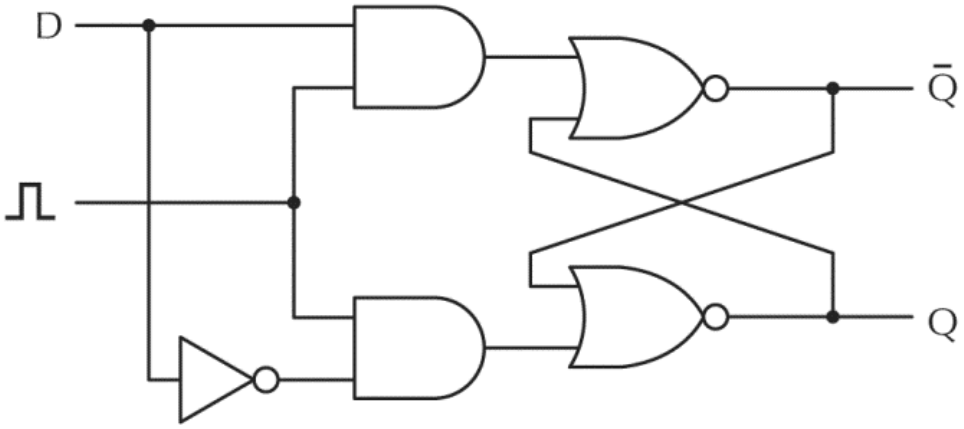


Figura 3.23 Latch D temporizzato.

E	D	Q	$\bar{Q}$
1	0	0	1
1	1	1	0
0	-	Q	$\bar{Q}$

NB: per caricare in memoria il nuovo valore di D, occorre un valore 1 sulla linea del clock

Latch D…. Flip Flop

PRO:

- mantenimento dell'uscita
- adatto all'impiego come interfaccia-memoria nel comando di tastiere o visualizzatori.

CONTRO

- lo stato dell'ingresso si porta all'uscita nell'istante (e per tutto il tempo in cui) l'ingresso E vale 1. Questo può essere causa di comportamenti indesiderati nel caso il componente fosse montato in contesti in cui l'uscita è riportata negata in ingresso. In questo caso essa comincerebbe a oscillare e, non appena E torna a 0, si avrebbe un valore del tutto casuale.

Flip-flop D-Edge-Triggered (o semplicemente flip-flop) basate sull'idea di eseguire il campionamento e la marcatura in intervalli temporali ben distinti.

Flip Flop

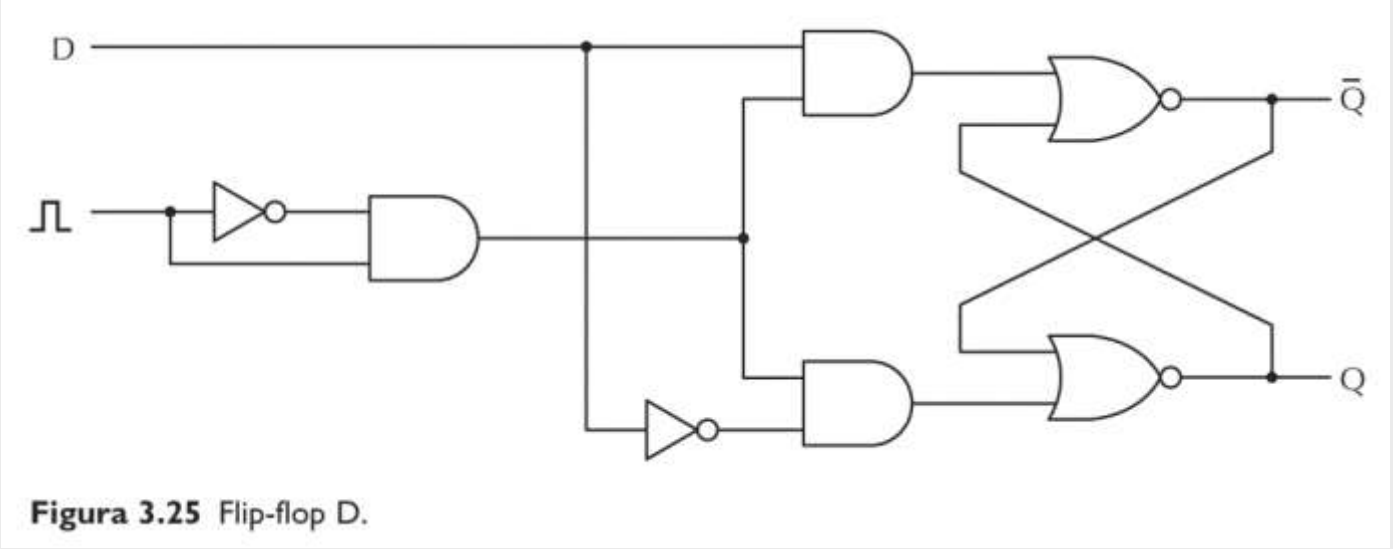


Figura 3.25 Flip-flop D.

La transizione di stato si verifica nella transizione del clock:

- Da 0 a 1 (fronte di salita)
- Da 1 a 0 (fronte di discesa)

La durata dell'impulso non ha importanza.
FLIP-FLOP: commutazione su fronte
LATCH: commutazione su livello

Registro

Registro a 8 bit

- *8 flip-flop*
- *Tutte le linee di clock sono collegate allo stesso clock*
- *Il valore di ogni bit del registro si modifica solo nell'istante del clk*

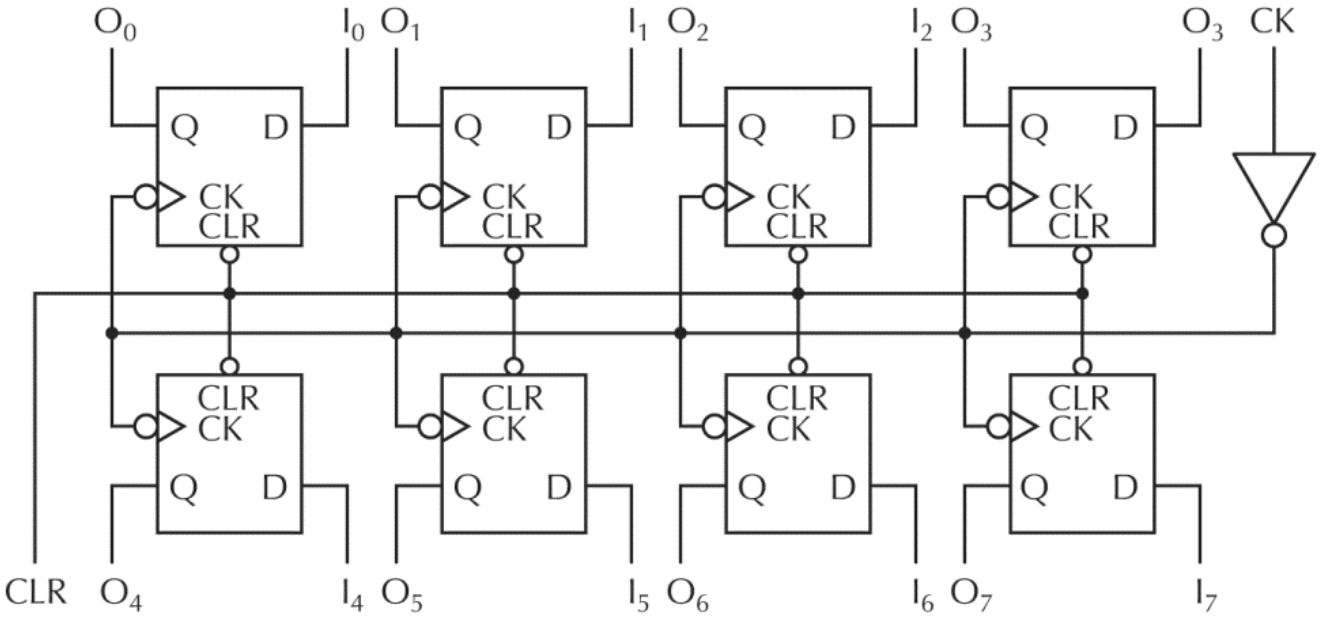


Figura 3.27 T Un registro a 8 bit costruito a partire da flip-flop a 1 bit.

Memoria

Memoria a 4 parole di 3 bit l'una

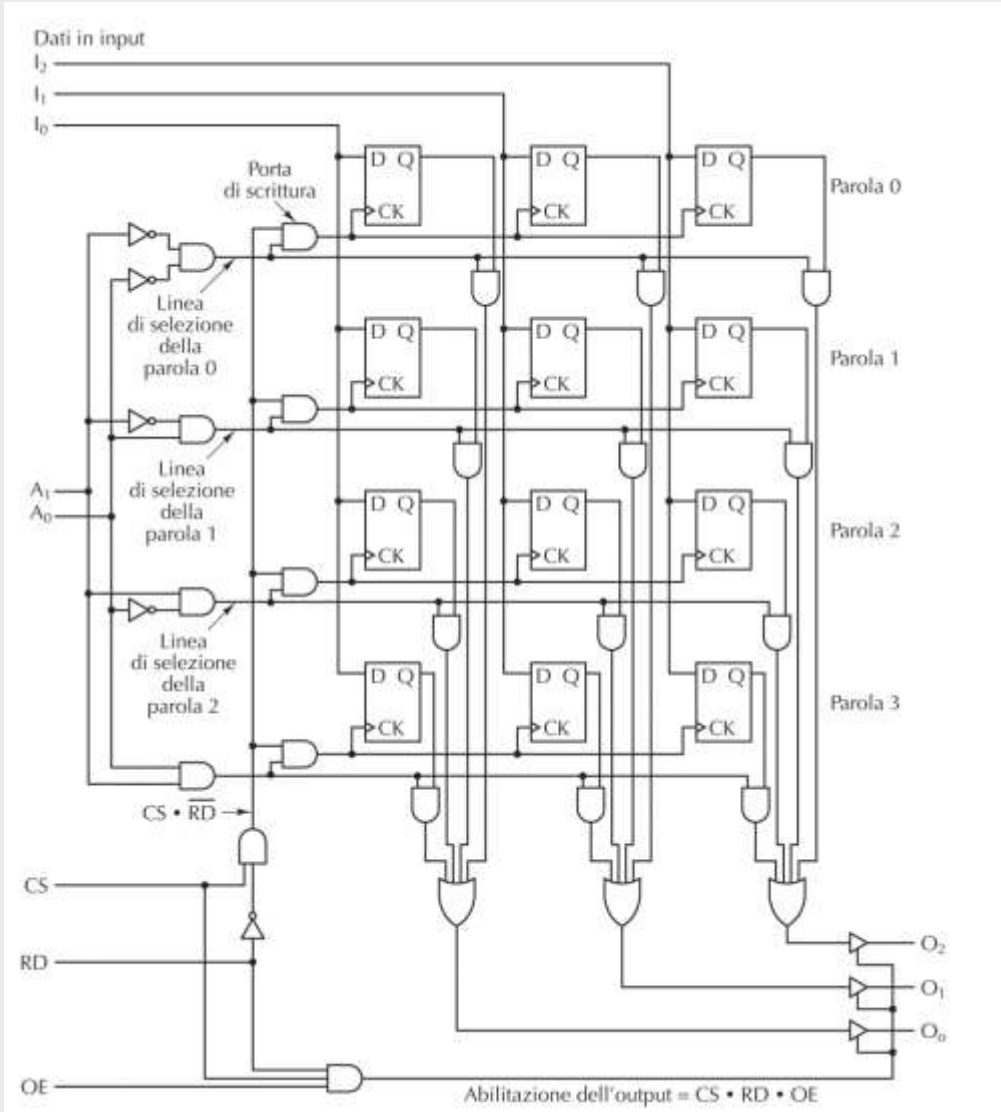
I0,I1,I2 : bit in input per ogni parola

A0,A1: indirizzano la parola

CS: selezione del chip

RD: seleziona la scrittura o la lettura

OE: abilita l'output (Output Enable)



Chip di Memoria

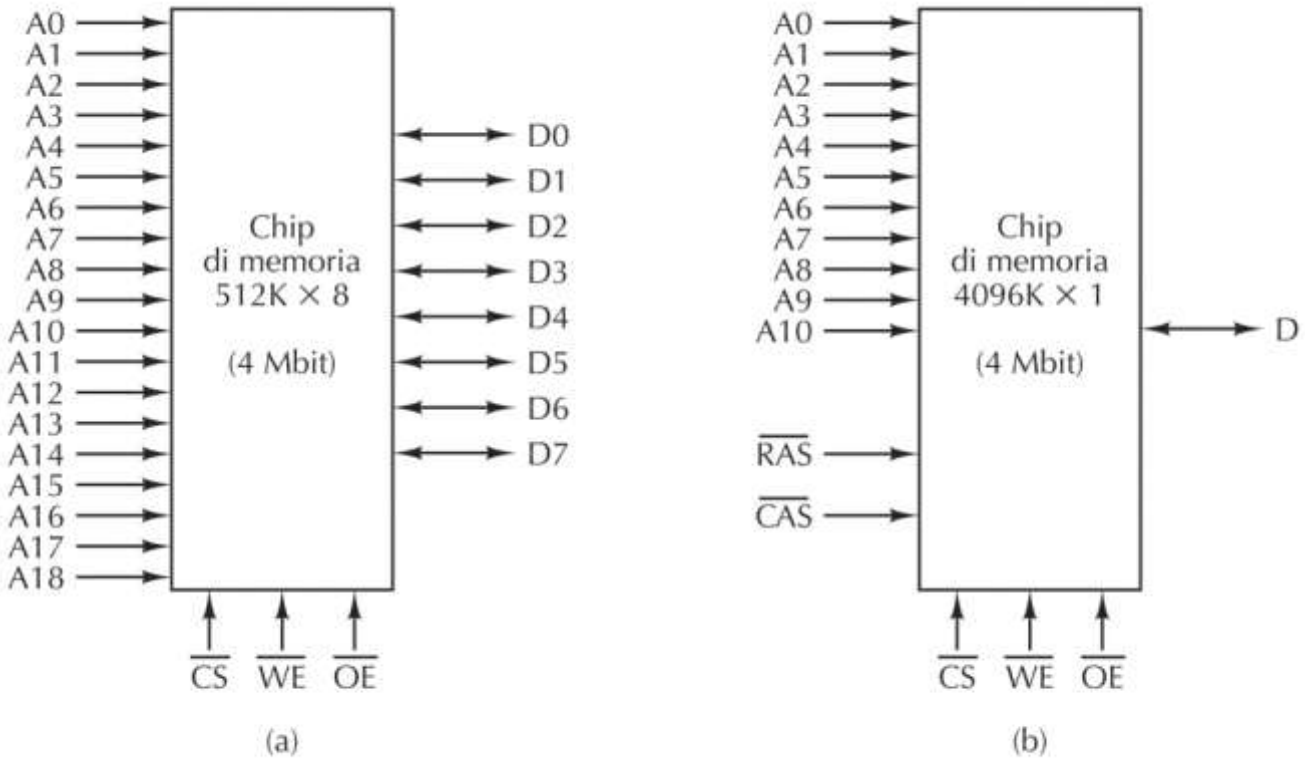


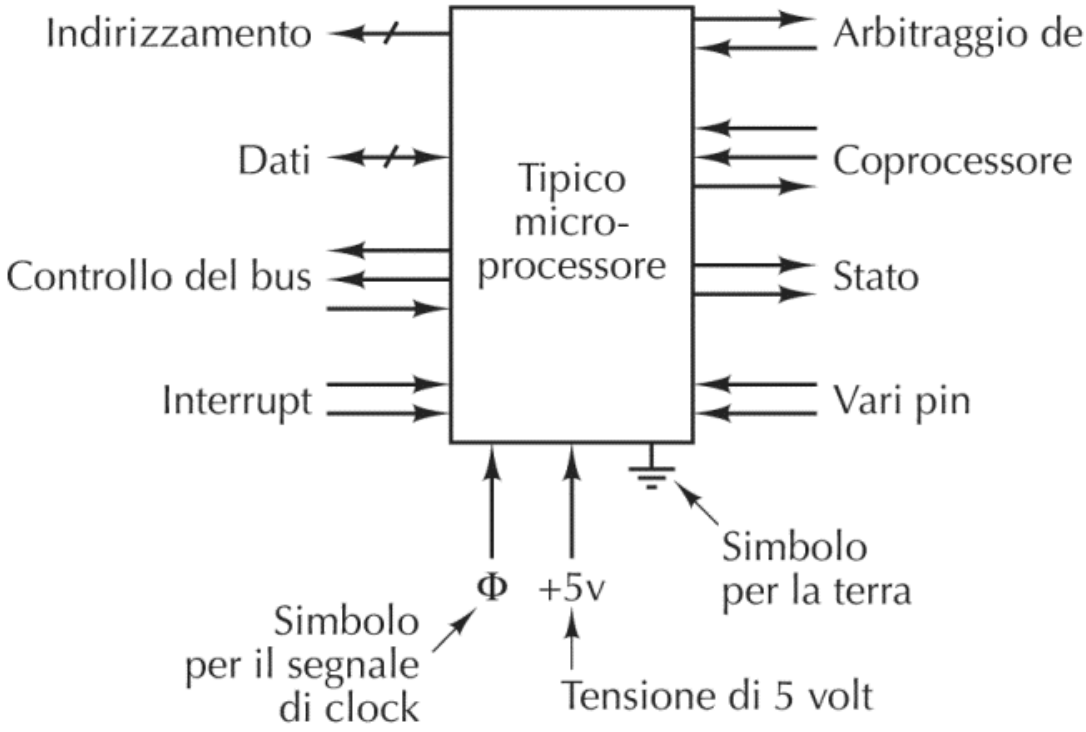
Figura 3.30 Due modi di organizzare un chip di memoria a 4 Mbit.

Tipologie di Memoria

Tipo	Categoria	Cancellazione	Byte modificabili	Volatile	Tipico utilizzo
SRAM	Read/write	Elettrica	Sì	Sì	Cache di secondo livello
DRAM	Read/write	Elettrica	Sì	Sì	Memoria centrale (vecchia)
SDRAM	Read/write	Elettrica	Sì	Sì	Memoria centrale (recente)
ROM	Read-only	Impossibile	No	No	Elettrodomestici (prodotti in grandi volumi)
PROM	Read-only	Impossibile	No	No	Dispositivi (prodotti in piccoli volumi)
EPROM	Read-mostly	Raggi UV	No	No	Prototipazione di dispositivi
EEPROM	Read-mostly	Elettrica	Sì	No	Prototipazione di dispositivi
Flash	Read/write	Elettrica	No	No	"Pellicola" per macchine fotografiche digitali

Figura 3.32 Confronto tra vari tipi di memoria.

CPU



- *linee di indirizzamento: : 4,8,16, 20, 22, 32.*
- *linee dati: 4,8,16, 32, 64.*
- *linee di controllo del bus: servono per sincronizzare le trasmissioni*
- *linee di arbitraggio sul bus: gestire i conflitti di richiesta del bus da parte dei diversi dispositivi*
- *Quelle di stato per garantire la compatibilità con precedenti processori*

Figura 3.34 Una generica CPU. Le frecce indicano i segnali di input e di output. I trattini diagonali indicano pin multipli; per una specifica CPU un valore ne indica la quantità.

CPU

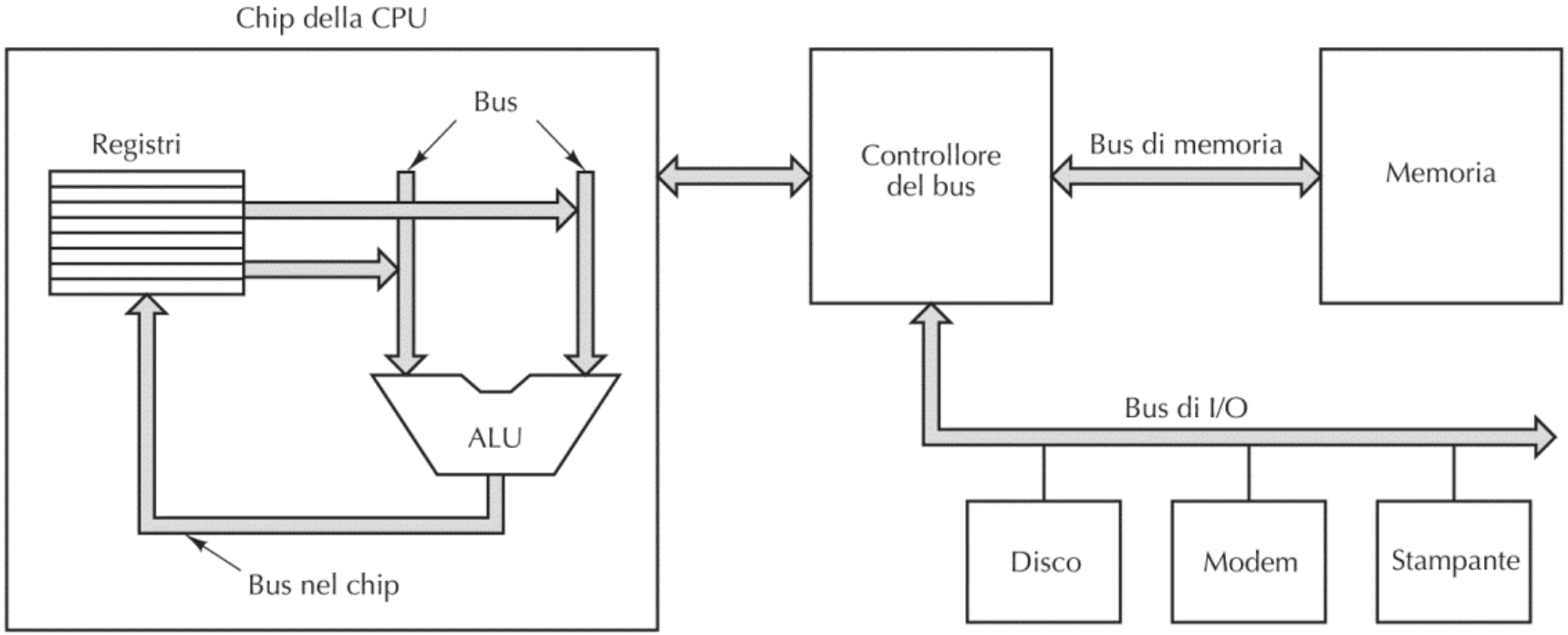


Figura 3.35 Sistema di un calcolatore con più bus.

BUS: master-slave

- Le relazioni tra i dispositivi sul bus sono regolati da una relazione di tipo Master/Slave.

Master	Slave	Esempio
CPU	Memoria	Prelievo delle istruzioni e dei dati
CPU	Dispositivo di I/O	Inizio del trasferimento dei dati
CPU	Coprocessore	Passaggio dell'istruzione al coprocessore da parte della CPU
I/O	Memoria	DMA (Direct Memory Access)
Coprocessore	CPU	Prelievo degli operandi dalla CPU da parte del coprocessore

Figura 3.36 Esempi di master e slave del bus.

Quando si progetta un bus si tiene presente:

- *L'ampiezza;*
- *La temporizzazione;*
- *L'arbitraggio;*
- *Le operazioni che consente.*

Da queste scelte dipende la velocità sul bus.

BUS: evoluzione nel tempo

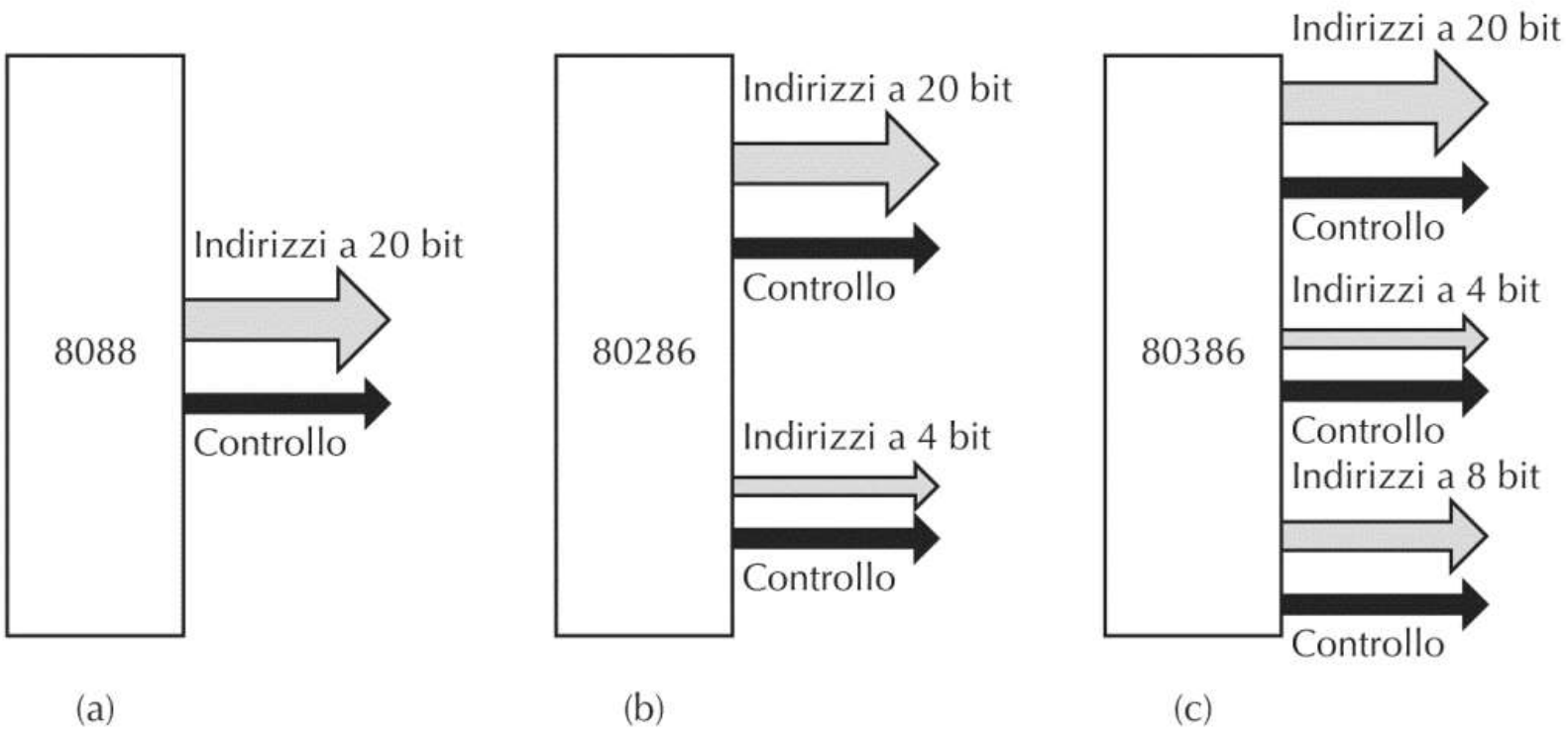


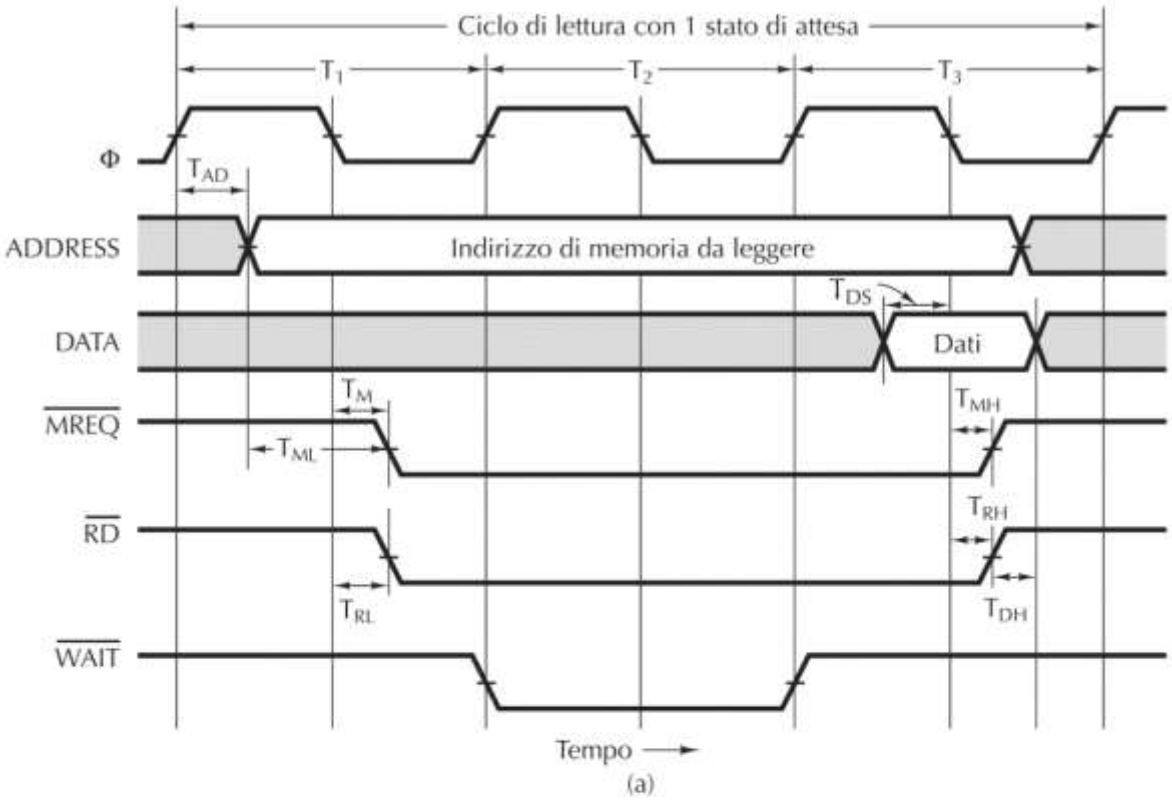
Figura 3.37 Crescita nel tempo degli indirizzi del bus.

BUS: temporizzazione

- I bus sincroni sono pilotati da un quarzo che genera un'onda quadra tra 5 e 133 MHz, tutte le operazioni vengono scandite da questi impulsi.

- I bus asincroni funzionano con il meccanismo dell'invio della domanda e dell'attesa della risposta - non hanno bisogno di clock.

BUS SINCRONO



(a)

Simbolo	Parametro	Min	Max	Unità
T_{AD}	Ritardo dell'output dell'indirizzo		4	nsec
T_{ML}	Indirizzo stabile prima di $\overline{MREQ}$	2		nsec
T_M	Ritardo di $\overline{MREQ}$ rispetto al fronte di discesa di Φ in T_1		3	nsec
T_{RL}	Ritardo di RD rispetto al fronte di discesa di Φ in T_1		3	nsec
T_{DS}	Tempo di impostaz. dei dati prima del fronte di discesa di Φ	2		nsec
T_{MH}	Ritardo di $\overline{MREQ}$ rispetto al fronte di discesa di Φ in T_3		3	nsec
T_{RH}	Ritardo di $\overline{RD}$ rispetto al fronte di discesa di Φ in T_3		3	nsec
T_{DH}	Tempo di mantenimento dei dati dopo la negazione di $\overline{RD}$	0		nsec

(b)

- T_1 , T_2 , T_3 : 3 clk
- Il primo ciclo inizia sul fronte di salita di T_1 e corrisponde all'invio dell'indirizzo della locazione
- MREQ indica la richiesta di accesso alla memoria
- RD indica se si vuole leggere o scrivere
- la memoria risponde asserendo il WAIT indicando alla CPU di non aspettarla
- All'inizio di T_3 la memoria nega il wait dichiarando di avere pronta
- al terzo clock i dati possono essere trasferiti

BUS: asincrono

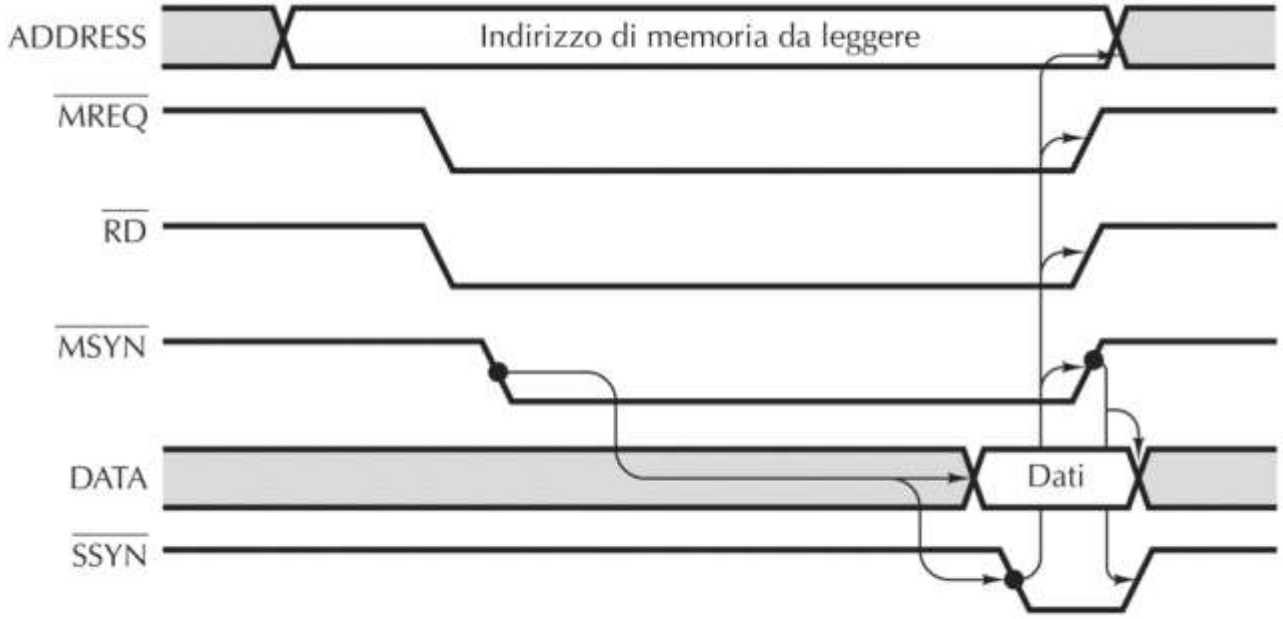


Figura 3.39 Funzionamento di un bus asincrono.

- No clock
- Quando ADDRESS viene lanciato l'indirizzo si propaga sul bus,
- subito dopo viene inviato MREQ e RD ed è con MSYN (master syncrinism) che si avvia l'operazione di trasferimento dati
- Concluso il trasferimento con la risposta SSYN (slave sincronism) dello slave, viene chiusa la comunicazione
- Successivamente può iniziare una nuova comunicazione

BUS: arbitraggio

PROBLEMA: Come decidere quale dispositivo può accedere al BUS

- **Daisy chaining**:
  - l'arbitro è integrato nel chip della cpu (quasi sempre)
  - tutti i dispositivi sono collegati ad una unica linea nell'ordine che si desidera dare loro (l'ordine corrisponde alla distanza del dispositivo dall'arbitro = livello di priorità)
  - L'arbitro vede solo se c'è una richiesta, se c'è concede il bus e se ne serve il dispositivo più vicino all'arbitro lungo il «festone», che è quello che ha fatto la richiesta, se non l'ha fatta si passa oltre al successivo, la concessione passa lungo tutto il festone di dispositivo in dispositivo fino al dispositivo che ha fatto richiesta.

In alcuni sistemi invece di un festone si usano anche 2, 4 o 8 livelli e così si può assegnare una priorità in base alla distanza e al livello di priorità del festone.

BUS: arbitraggio

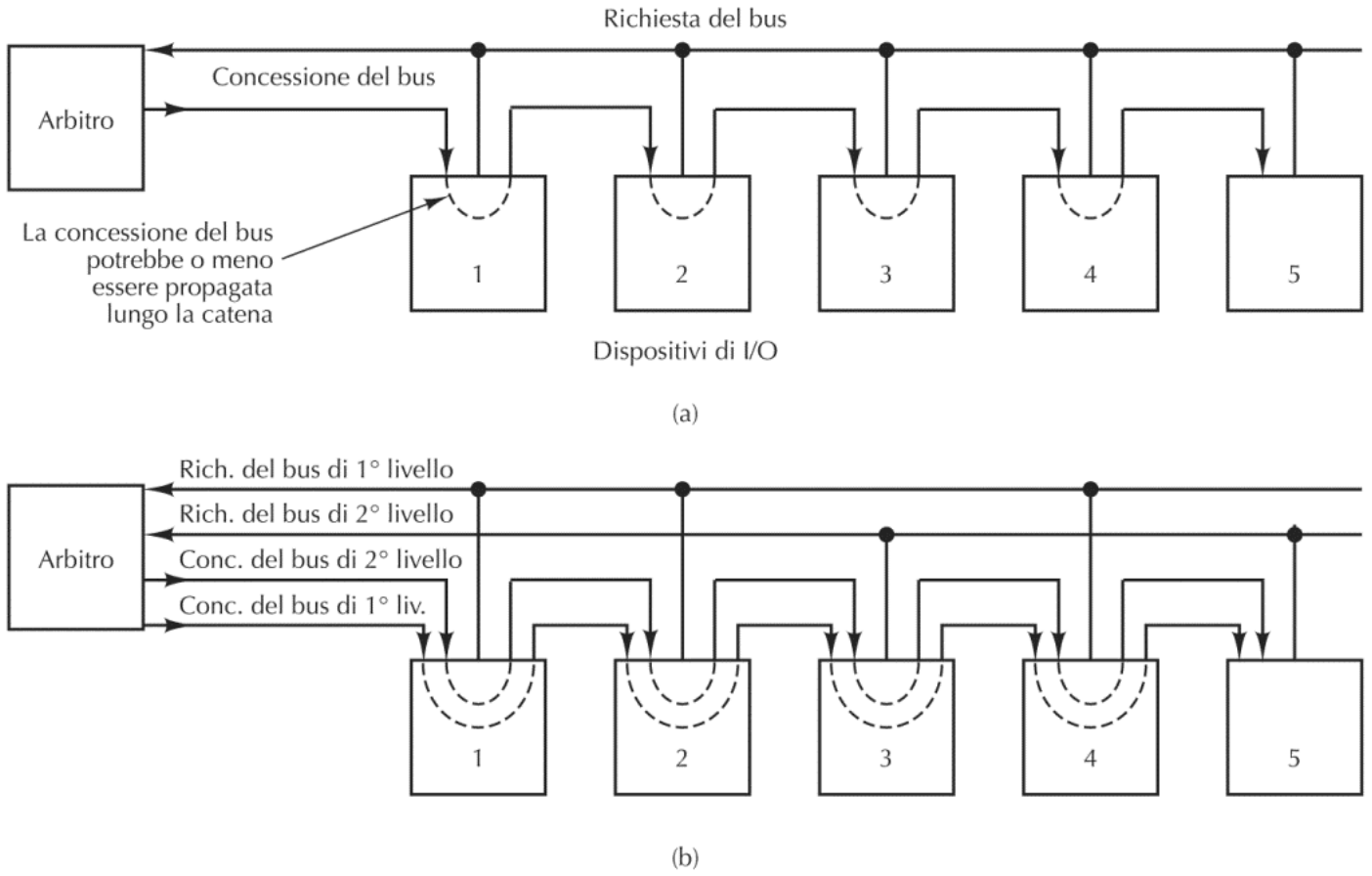


Figura 3.40 (a) Arbitro del bus centralizzato e a un livello che utilizza un collegamento a festone. (b) Stesso arbitro, ma a due livelli.

BUS: operazioni

SMP – macchine odierne multiprocessore

Cosa accade se più dispositivi richiedono l'uso del bus contemporaneamente?

Soluzione:

- dotare il bus di una ulteriore linea:
  - 0 il bus può essere concesso
  - 1 bus occupato, chi fa richiesta deve aspettare.
  - Ma se quando è 0 sono in due a fare richiesta contemporanea????
    - Soluzione: leggi-modifica-scrivi operazione atomica sul bus

Tipologie di BUS

- **Universal Serial Bus**, anche noto come bus “Industrial Serial Architecture” (ISA)
  - utilizzato per connettere tastiere, mouse, stampanti ecc… cioè dispositivi lenti

- **PCI (Peripheral Component Interconnected).**
  - bus sincrono con relazione master slave tra trasmittente e ricevente,
  - le stesse 64 linee del bus vengono utilizzate sia per indirizzi che per dati con il meccanismo del multiplexing:
    - nel primo ciclo viene immesso l’indirizzo sul bus,
    - nel secondo viene rimosso l’indirizzo ed il bus viene invertito in maniera che lo slave possa utilizzarlo,
    - nel terzo si generano in output i dati richiesti.

Tipologie di BUS: PCI

Il Bridge coordina i dispositivi e le loro diverse velocità.

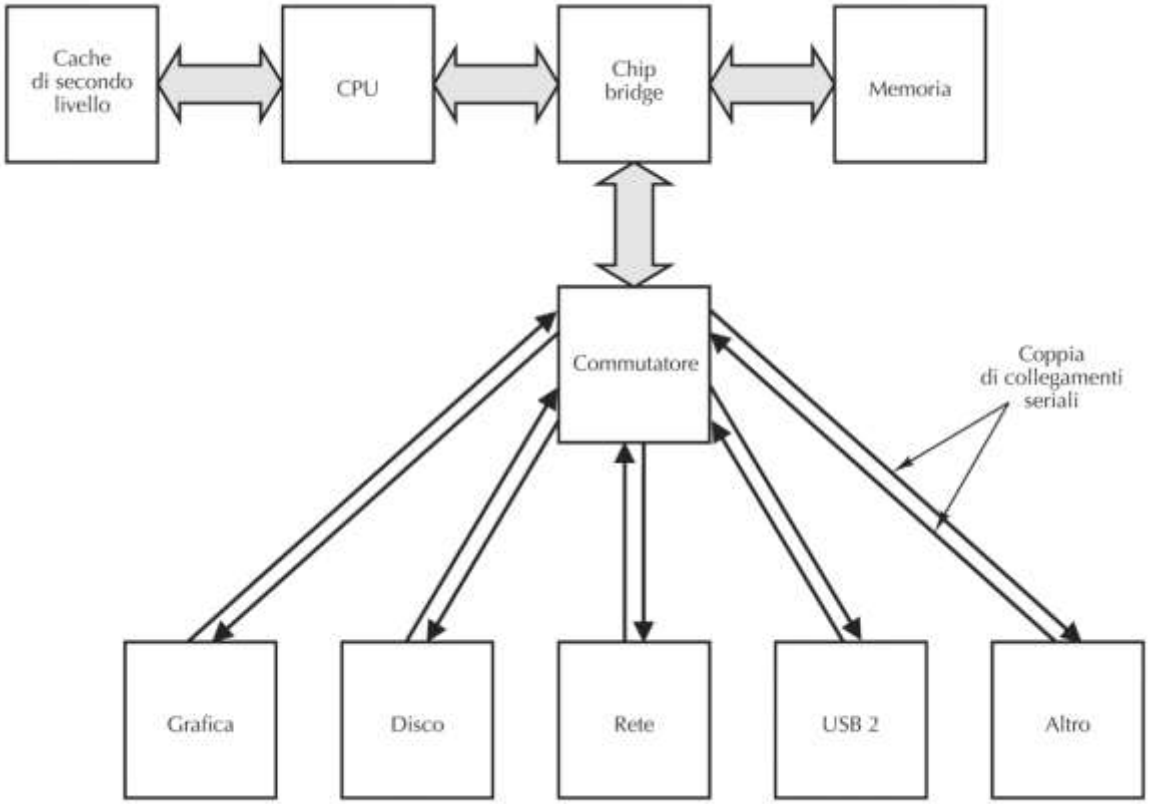


Figura 3.56 Tipico sistema PCI Express.

- Il commutatore che può stare nel Bridge oppure nello stesso chip della CPU.
- Le connessioni consistono di due fili, uno per la massa e l'altro su cui viaggiano i segnali.
- La connessione è punto punto e seriale.
- Possibilità di inserimento e soppressione a caldo dei dispositivi.
- I commutatori sono piccoli dispositivi viene favorita la migliore integrazione e l'uso in computer di più piccole dimensioni (smart phone, tablet, embedded).

Tipologie di BUS: USB

UNIVERSAL SERIAL BUS

Il sistema USB è composto da un **Hub principale** che si collega:

- al bus del sistema
- ai cavi dei dispositivi organizzati ad albero per dare la possiblità di connessioni multiple. La radice dell'albero è nell'hub principale.

Il cavo di connessione consiste di 4 fili:

- 2 per i dati,
- 1 per GND,
- 1 per la tensione a 5V.

Tipologie di BUS: USB

Invio di Frame

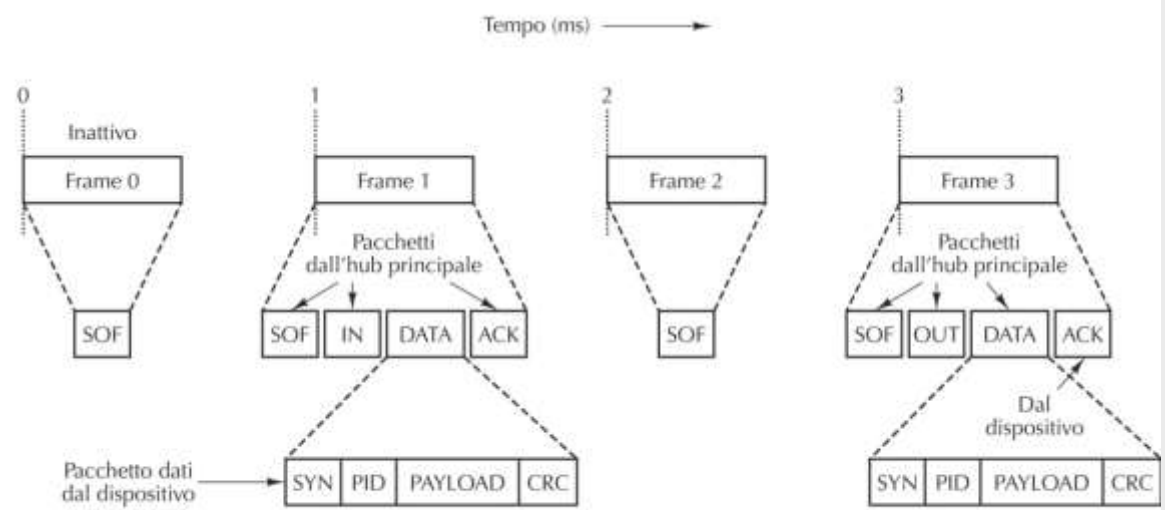


Figura 3.58 L'hub principale USB spedisce frame ogni millisecondo.

4 tipi di frame

1. **controllo** utilizzati per configurare i dispositivi, assegnare loro i comandi e interrogarli;
2. **Isocroni** (microfoni, altoparlanti, ecc.): spediscono a precisi intervalli di tempo ma non possono richiedere la trasmissione in caso di errore;
3. **Bulck** usati per la tramissione di grandi volumi di dati;
4. **Interrupt**

BUS: operazioni e interrupt CHIP 8259A Intel

Come fa un dispositivo ad indicare di voler essere servito?

Interrupt..problemi analoghi ai precedenti

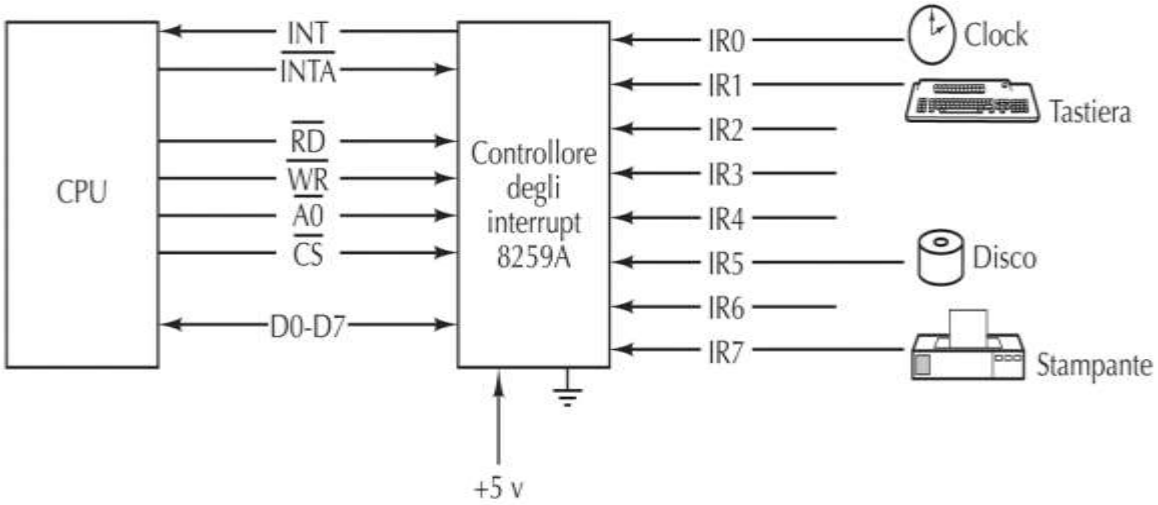


Figura 3.43 Utilizzo del controllore degli interrupt 8259A.

1. *IRx (interrupt Requet): richiesta di interrupt da un dispositivo*
2. *Il chip 8259A asserisce la linea INT (INTerrupt)*
3. *Quando la CPU può gestire l'interupt , asserisce INT A' (INTerrupt Aknowledgement),*
4. *Il chip 8259A specifica l'input che ha generato l'interrupt,*
5. *La CPU usa quel numero come vettore di interrupt in una tabella che seleziona la procedura.*
6. *La CPU può leggere o scrivere nei registri del chip 8259A utilizzando i cicli di bus ed i pin: RD (ReaD), WR (Write), CS (ChipSelect) e AO (Anable Output).*

Un esempio di architettura: Intel i7

Intel Core i7… un po' di numeri

- Introdotto nel novembre 2008
- 4 processori, frequenza di 3,2 GHz
- 447 linee per segnali + 286 per alimentazione + 360 di terra + 62 riservate per utilizzazi futuri
- La prima versione che usava l'architettura "Nahalem" è poi stata sostituita da una nuova versione "Sandy Bridge": contiene fino a 6 processori con frequenza di 3.5 GHz

Un esempio di architettura: Intel i7

Intel Core i7… un po' di sostanza

- 64bit.
- standard IEEE 754 per implementazione in virgola mobile
- numero di processori che va da 2 a 6
- l'hyperthrading (cioè il multithreading simultaneo)
- può eseguire fino a 4 istruzioni per volta: macchina superscalare a 4 livelli
- Ogni processore ha 3 livelli di cache:
  - L1-Dati da 32 KB,
  - L1-Istruzioni da 32 KB,
  - L2 Unificata da 256 KB,
  - L3 Unificata da 4 a 15 MB

Un esempio di architettura: Intel i7

- Due bus sincroni:
  - uno interno di tipo DDR3 (Dinamic Direct Random Access memory) per l'accesso alla memoria centrale
  - uno PCIe (PCI Express) per i dispositivi di I/O
- Può consumare da 17 a 150W … problemi di dissipazione del calore.

- Al di sopra può essere montata una ventola di raffreddamento.
- 5 stati diversi con diversi livelli di raffreddamento che vanno dal sonno profondo alla piena utilizzazione.
- Nei vari livelli vengono disattivati alcune funzionalità

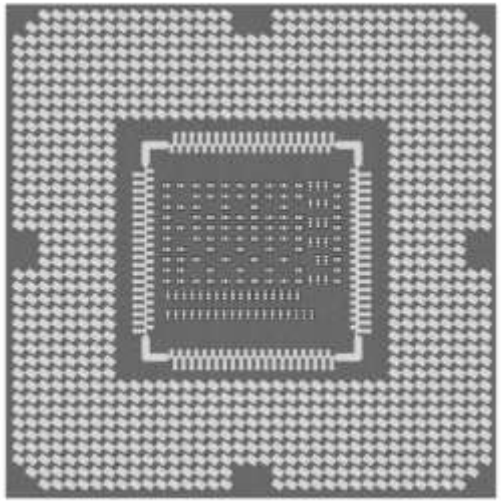


Figura 3.44 Disposizione fisica dei contatti del Core i7.

Un esempio di architettura: Intel i7

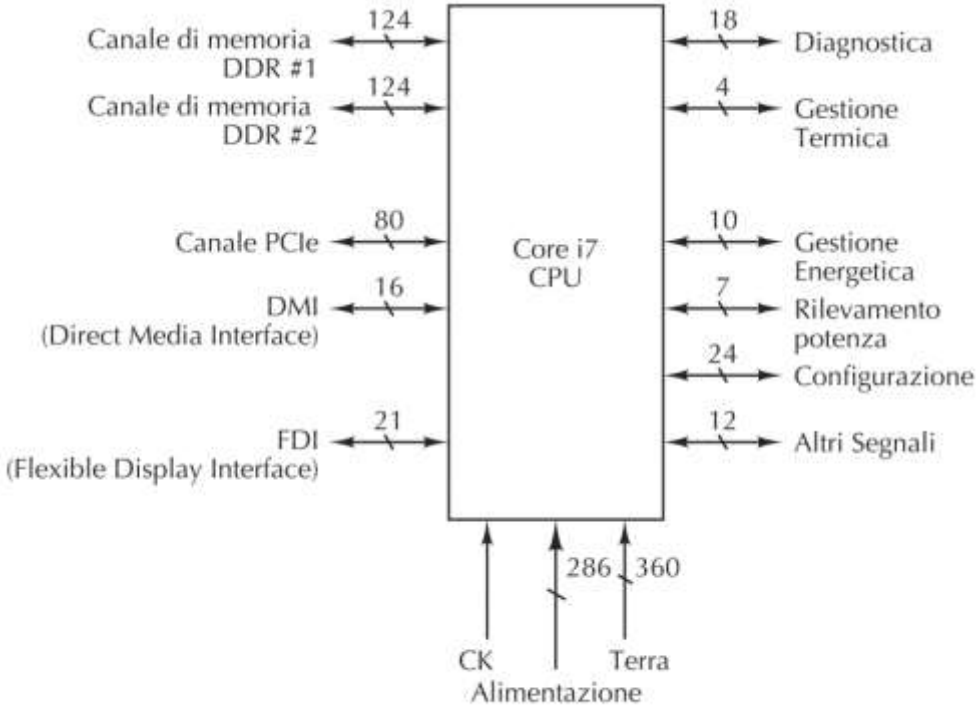


Figura 3.45 Disposizione logica dei contatti del Core i7.

Lato sinistro:

- Accesso alle memorie,
- gestione delle periferiche
- accesso diretto multimediale
- interfaccia flessibile al display.

Lato destro:

- gestione termica,
- gestione energetica,
- rilevamento della potenza,
- diagnostica,

Un esempio di architettura: Intel i7

- Due banchi memoria: **DDR #1 e DDR #2**, sono compatibili con la DDR3 e lavorano a 666 MHz ciascuno e quindi permettono 1333 milioni di transizioni al secondo.

  La DDR3 ha un'ampiezza a 64 bit e i due banchi lavorano in tandem potendo sopportare programmi fino a 20 GB di dati al secondo.

- Connessione tra CPU e periferiche via bus **PCI Express:** interfaccia seriale veloce in cui ogni singolo collegamento forma una "lane". Il Core i7 consente fino a 16 lane offrendo 16 GB/s di trasferimento potendo trasferire comandi read, write, interrupt, comandi di configurazione oltre ovviamente ad indirizzi e dati.

- **DMI (Direct Media Interface)**: connessione tra CPU e chipset, ossia insieme di chip che servono alla gestione delle porte, USB, audio, PCIe, Flash ed anche il DMA e lo fa attraverso il chip ICH10 che al suo interno ha anche il controllo di clock real time.

Un esempio di architettura: Intel i7

Il bus di memoria DDR3 è strutturato a pipeline a 3 fasi:

ACTIVATE della memoria: si apre una riga della DRAM per accessi successivi:

READ o WRITE: possono effettuarsi anche accessi multipli;

PRECHARGE: chiude la riga corrente della DRAM e prepara la memoria per il prossimo ACTIVATE.

L'organizzazione della memoria su più banchi consente l'accesso contemporaneo sui diversi banchi concorrenti.

Un esempio di architettura: Intel i7

In questo esempio si effettuano 4 accessi di memoria a 3 distinti banchi DRAM.

NB: Le letture avvengono in parallelo sullo stesso chip.

clock

Specifica l'attività

Indicazione del banco su cui operare

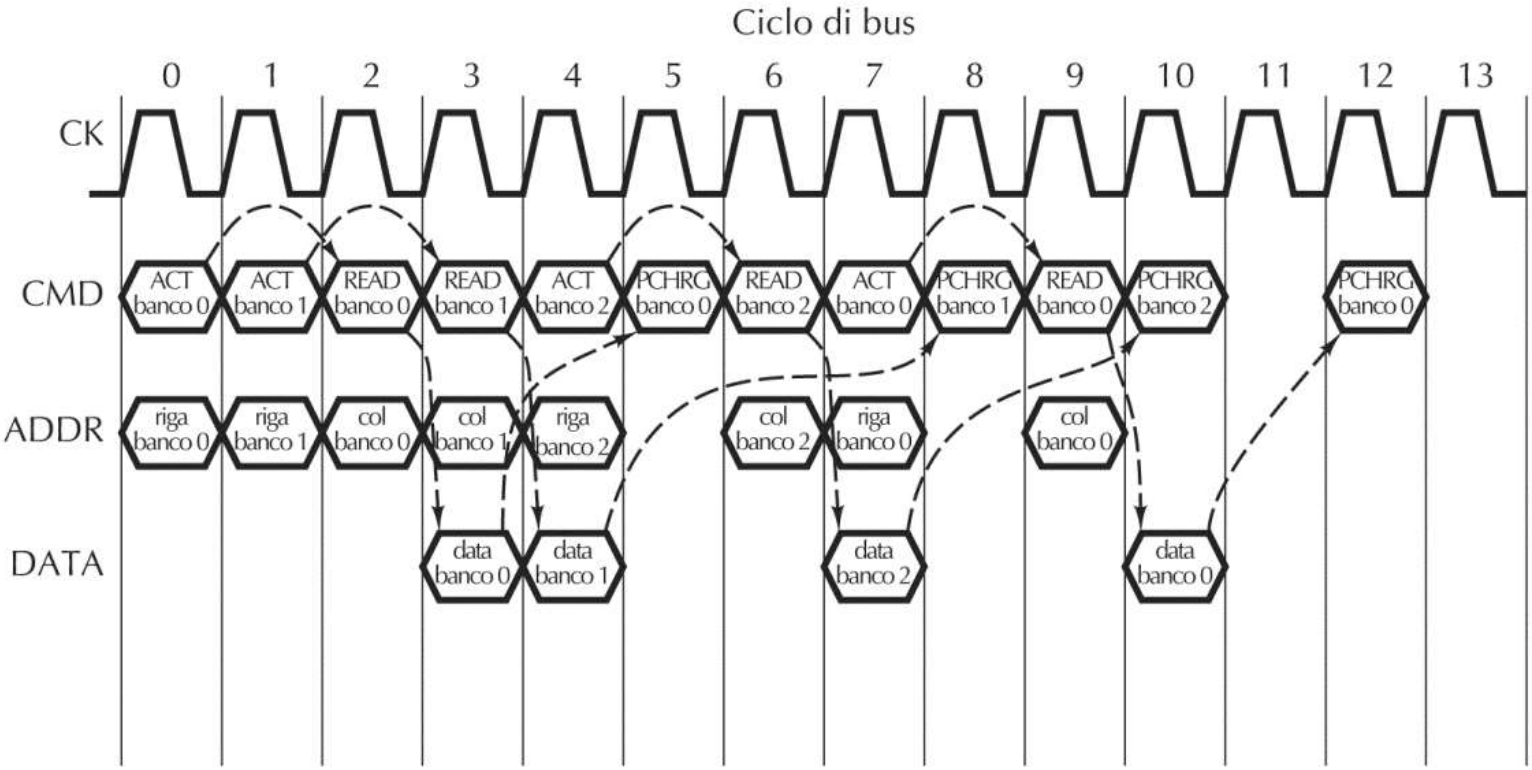


Figura 3.46 Richieste di memoria sull'interfaccia DDR3 del Core i7 gestite con pipeline.

Un esempio Intel i7

STRUTTURA DEL BUS

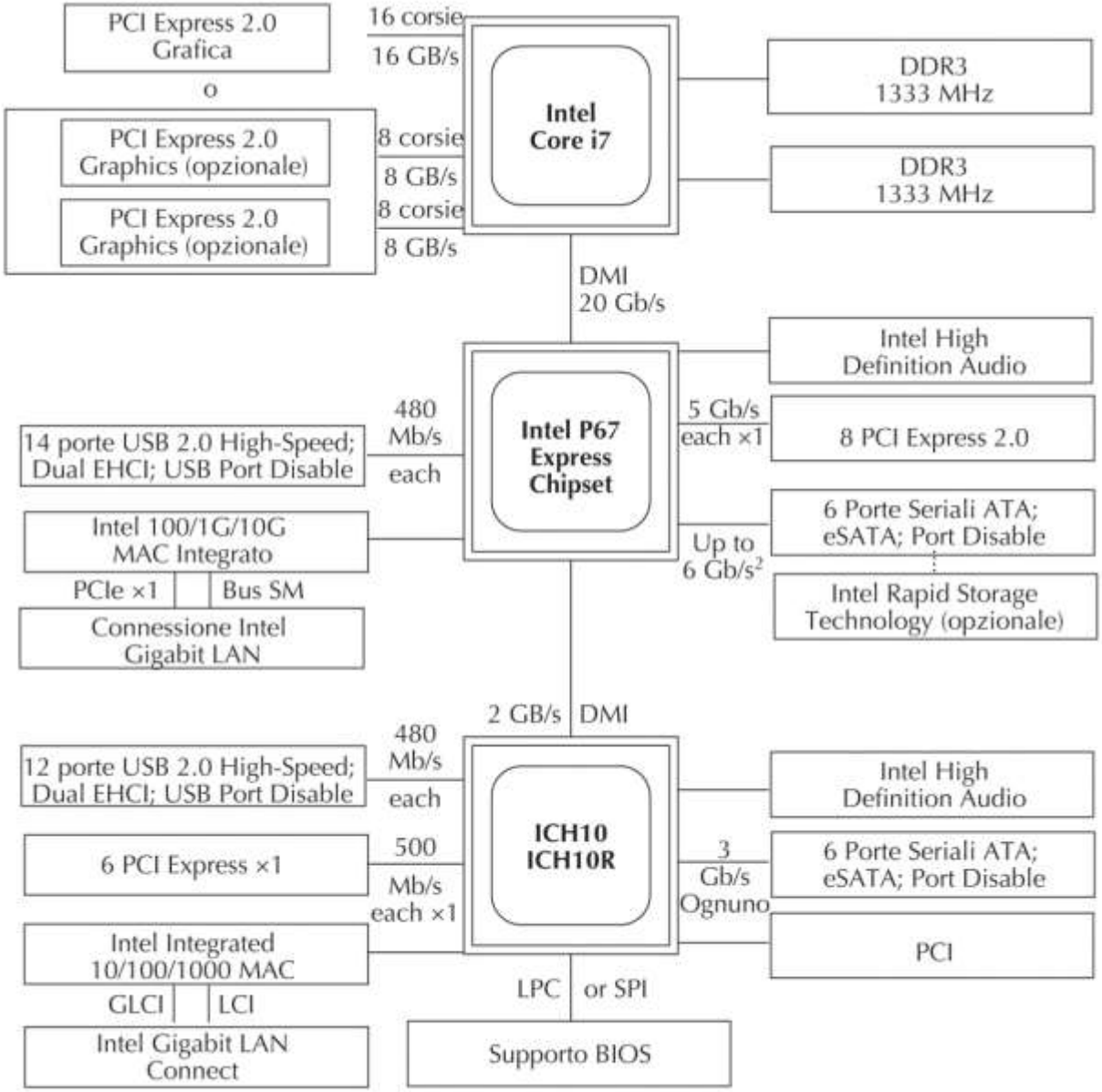


Figura 3.52 La struttura del bus di un moderno sistema Core i7.