Circuiti logici di base

- “Mattoncini” base per la costruzione di circuiti combinatori e sequenziali più sofisticati.
- Realizzano funzioni di utilità universale.
- Necessari nella progettazione (e descrizione) strutturata di circuiti complessi.

Vedremo comportamento, realizzazione, utilizzo dei più significativi rispetto a specifiche necessità:

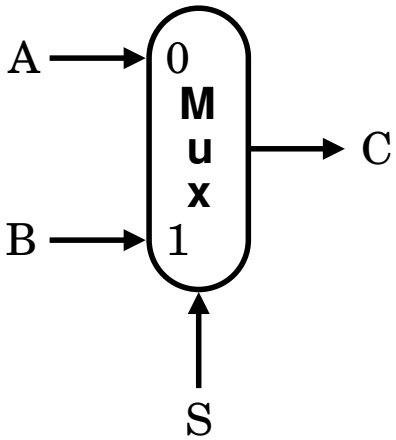
- multiplexer, decoder, demultiplexer
- comparatore
- mezzo sommatore, sommatore completo, traslatore
- latch, flip-flop.

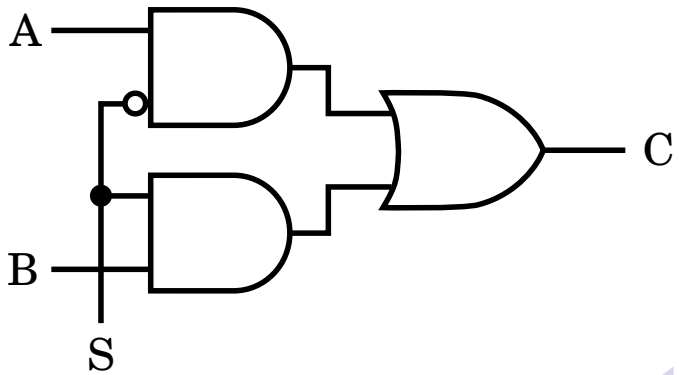
Multiplexer: comportamento

- n ingressi di controllo
- 2^n ingressi di segnale
- una uscita di segnale.

Gli ingressi di controllo selezionano un ingresso di segnale da instradare all'uscita.

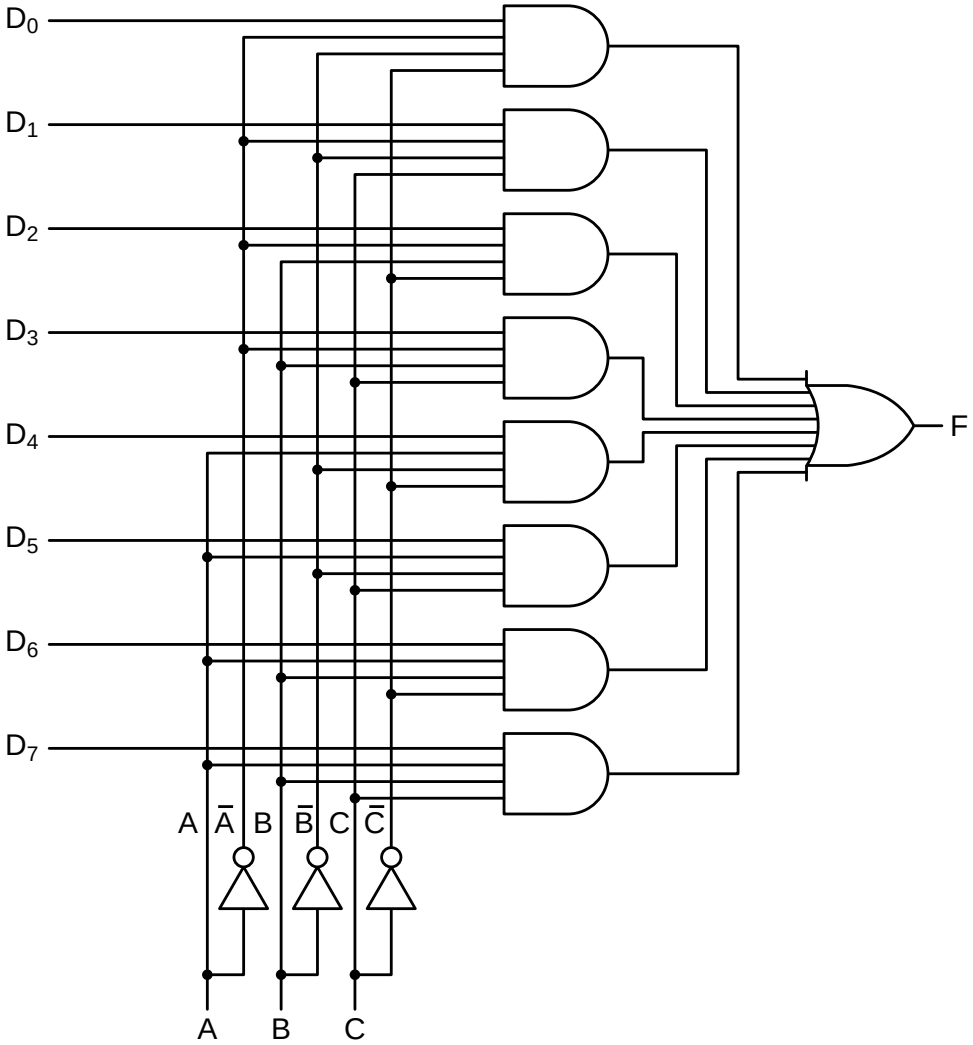
Es.: 2 ingressi di segnale, 1 di controllo





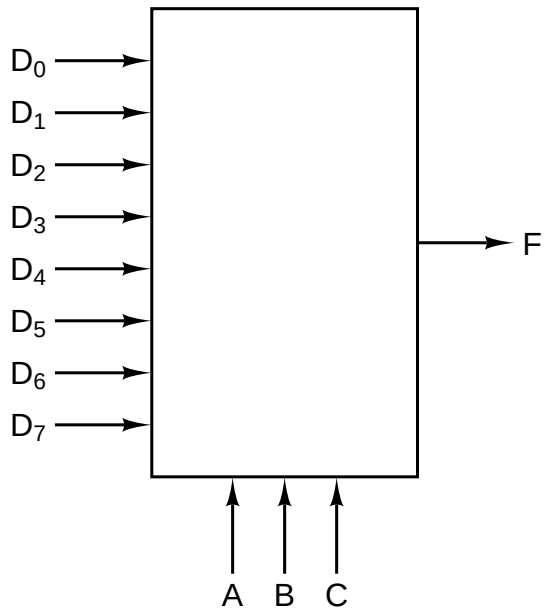
Multiplexer: realizzazione

Es.: 8 ingressi di segnale, 3 di controllo

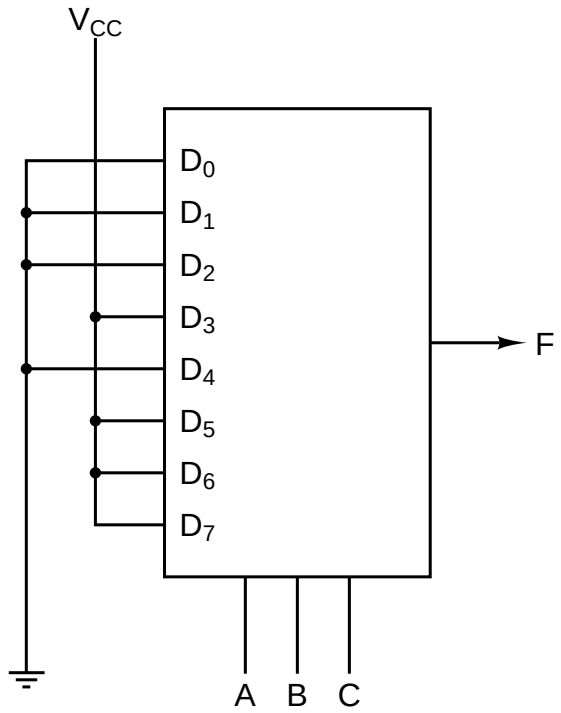


Multiplexer: utilizzo

- Trasformazione segnali parallela $\Rightarrow$ seriale
- Realizzazione di tabelle di verità



(a)

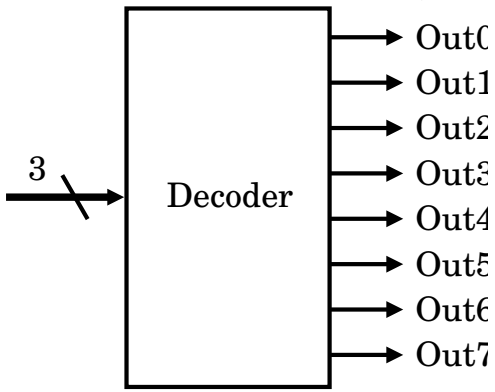


(b)

Decoder (decodificatore) a n bit

- n ingressi di controllo
- 2^n uscite di controllo
- 0 ingressi di segnale.

Gli ingressi di controllo selezionano una uscita.
L'uscita selezionata ha valore 1; tutte le altre 0.

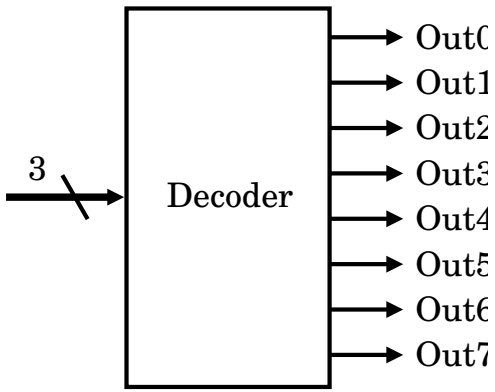


a. A 3-bit decoder

Decoder (decodificatore) a *n* bit

- *n* ingressi di controllo
- 2^n uscite di controllo
- 0 ingressi di segnale.

Gli ingressi di controllo selezionano una uscita.
L'uscita selezionata ha valore 1; tutte le altre 0.

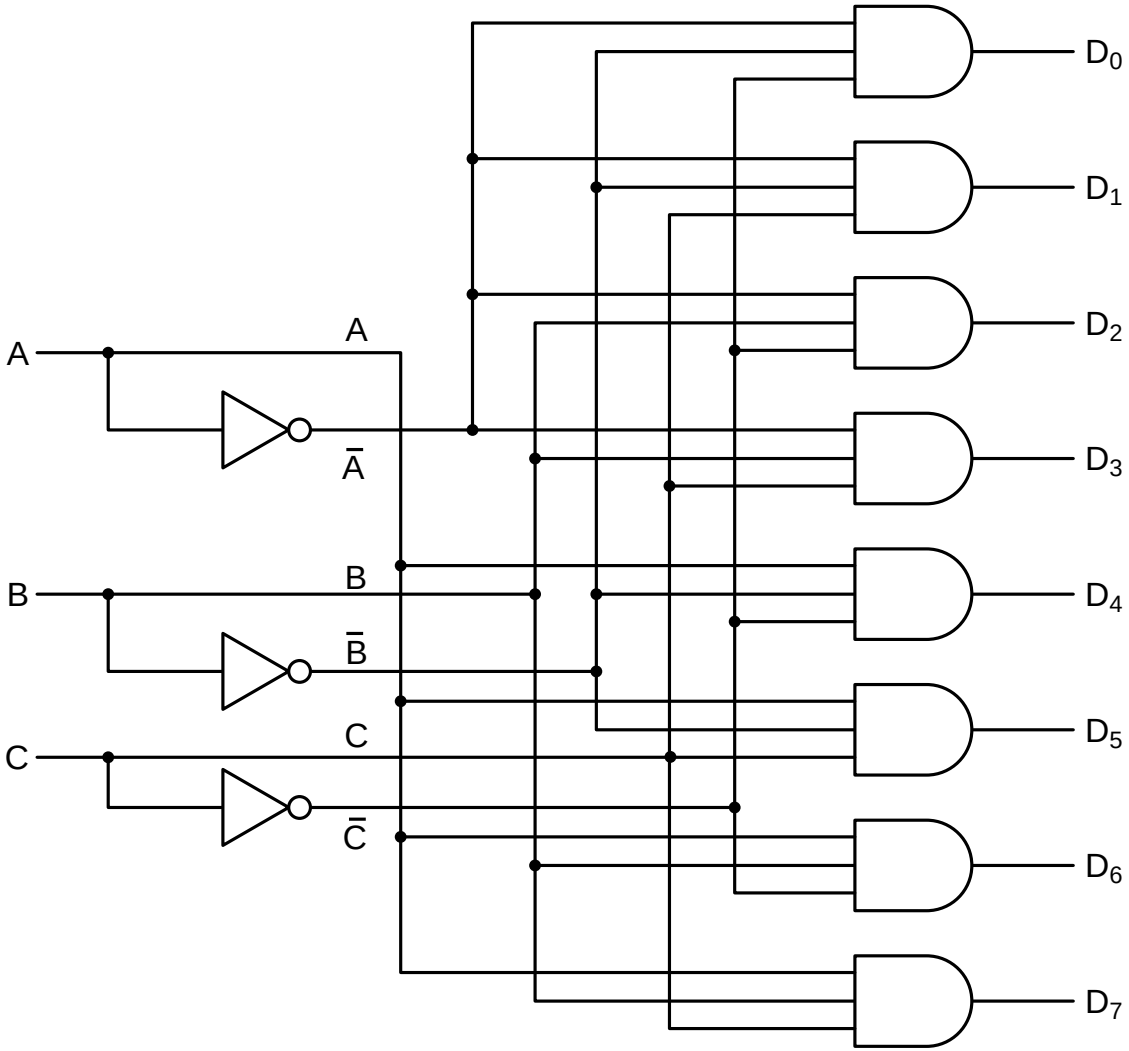


a. A 3-bit decoder

N.B.: nessuna parentela con il decoder di segnali in codice (che vedremo più avanti).

Decoder a n bit: realizzazione

Es.: $n = 3$

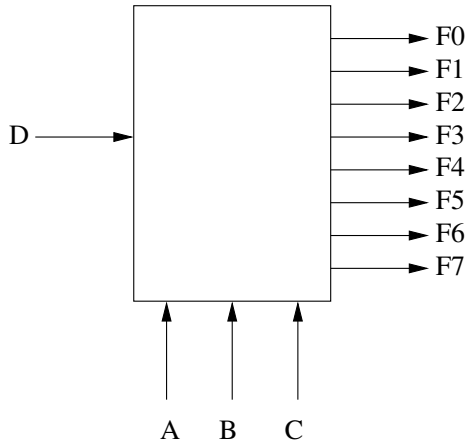


Decoder: utilizzo. Demultiplexer

Selezionare, tra molti dispositivi provvisti di un ingresso di attivazione, quello desiderato.

Es.: selezionare uno tra 2^n chip di memoria presenti nel calcolatore.

Dualmente al multiplexer, il Demultiplexer instrada un ingresso di segnale in una delle 2^n uscite di segnale.

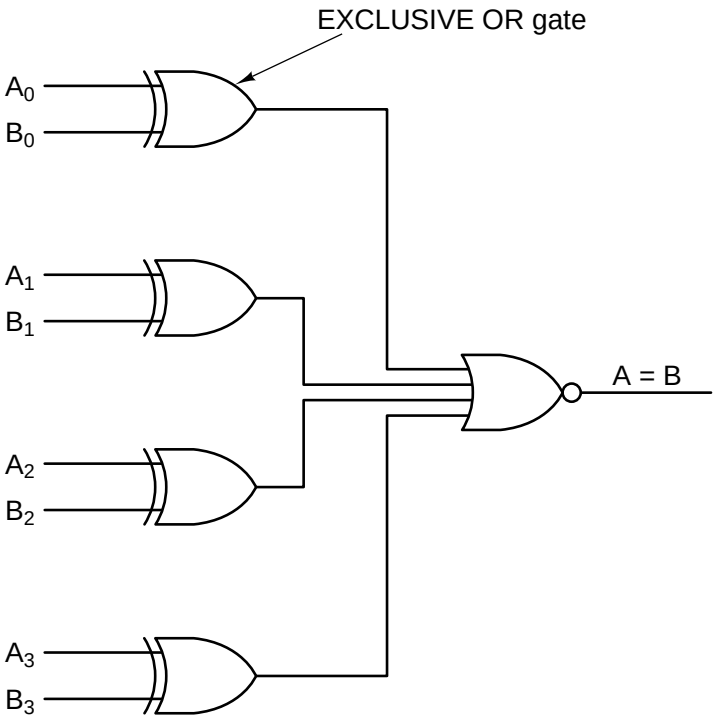


Esercizio: Realizzare, dal circuito del rispettivo decoder, un demultiplexer a un ingresso e 8 uscite.

Comparatore a n bit

2 ingressi a n bit: $A = (A_1, \ldots, A_n)$, $B = (B_1, \ldots, B_n)$.

L'uscita vale 1 se e solo se i due ingressi sono identici.



Uso: confronto di n-uple di bit.

Circuiti aritmetici

Come viene realizzata l'aritmetica nel calcolatore all'interno di una Arithmetic Logic Unit?

Vedremo i seguenti circuiti aritmetici:

- mezzo sommatore
- sommatore completo
- shifter.

Notazione posizionale

Ogni macchina calcolatrice deve

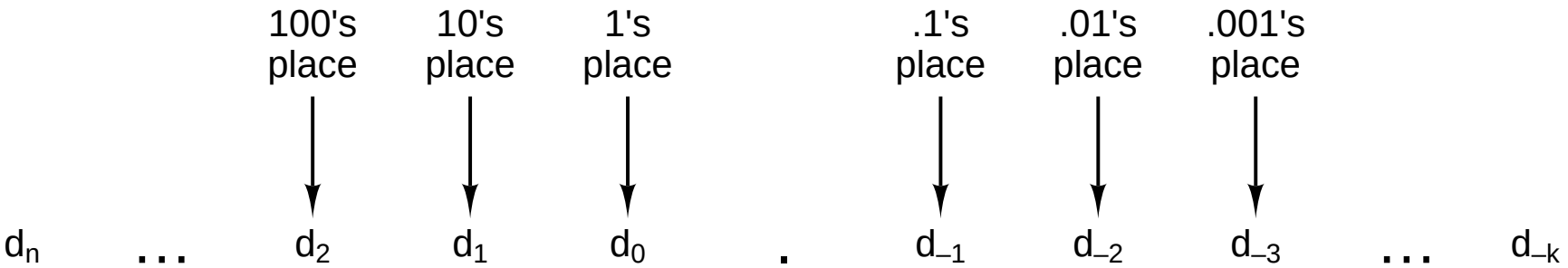
- rappresentare i numeri naturali (corrispondenti al valore di una quantità: 3 pecore, 8 dadi. . .)
- eseguire operazioni aritmetiche.

Notazione posizionale

Ogni macchina calcolatrice deve

- rappresentare i numeri naturali (corrispondenti al valore di una quantità: 3 pecore, 8 dadi...)
- eseguire operazioni aritmetiche.

Notazione posizionale: il peso di una cifra dipende dalla sua posizione (usiamo l'operatore somma $\sum$):



$$\text{Number} = \sum_{i=-k}^{n} d_i \times 10^i$$

Notazione posizionale in base B

Binary 1 1 1 1 1 0 1 0 0 0 1

$1\times2^{10} + 1\times2^{9} + 1\times2^{8} + 1\times2^{7} + 1\times2^{6} + 0\times2^{5} + 1\times2^{4} + 0\times2^{3} + 0\times2^{2} + 0\times2^{1} + 1\times2^{0}$

$1024 + 512 + 256 + 128 + 64 + 0 + 16 + 0 + 0 + 0 + 1$

Octal 3 7 2 1

$3\times8^{3} + 7\times8^{2} + 2\times8^{1} + 1\times8^{0}$

$1536 + 448 + 16 + 1$

Decimal 2 0 0 1

$2\times10^{3} + 0\times10^{2} + 0\times10^{1} + 1\times10^{0}$

$2000 + 0 + 0 + 1$

Hexadecimal 7 D 1 .

$7\times16^{2} + 13\times16^{1} + 1\times16^{0}$

$1792 + 208 + 1$

$$\text{valore} = \pm \sum_{i=-k}^{n} d_i \cdot B^i, \qquad 0 \leq d_i < B, \qquad B \text{ base}.$$

Occorrono B simboli per rappresentare un numero in base B: 0,1,...,9,A,B,C,D,E,F,...

Notazione binaria

Il calcolatore utilizza la base 2, con molteplici vantaggi:

- un bit è sufficiente per rappresentare una cifra
- un segnale digitale a due livelli può rappresentare una sequenza di bit
- un segnale digitale a 2^P livelli può rappresentare una sequenza di gruppi di P bit

Potendo rappresentare con una data precisione qualunque numero, l'hardware per il calcolo quindi può essere interamente progettato per funzionare nel dominio digitale.

Tipicamente i numeri interi sono rappresentati con 8, 16, 32 o 64 bit.

Operazioni aritmetiche: somma

Gli algoritmi che eseguono un'operazione aritmetica non dipendono dalla base scelta.

Algoritmo per la somma (quello delle elementari):

- si sommano coppie di cifre di pari peso a partire dalle meno significative
- eventualmente si generano riporti

1	1	1		
	1	1	1	+
	1	0	1	=
1	1	0	0	

Più numeri possono essere sommati iterando somme di due numeri.

Operazioni aritmetiche: sottrazione

Algoritmo per la sottrazione (quello delle elementari):

- si sottraggono coppie di cifre di pari peso a partire dalle meno significative
- eventualmente si riportano valori da sottrarre dalle cifre superiori

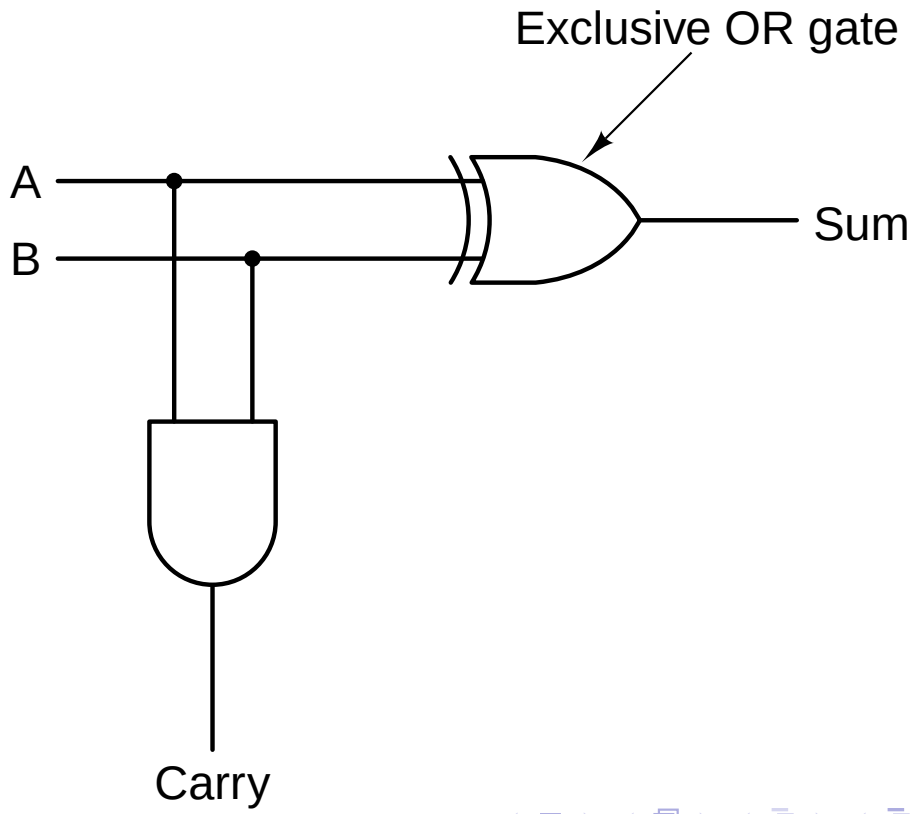
0	01	10	prestito sul prestito
0	10	0	prestito
1	0	0	-
	1	1	=
0	0	1	

Più numeri possono essere sottratti iterando sottrazioni di due numeri.

Mezzo sommatore

Esegue la somma su una cifra binaria, restituendo il risultato e il riporto

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

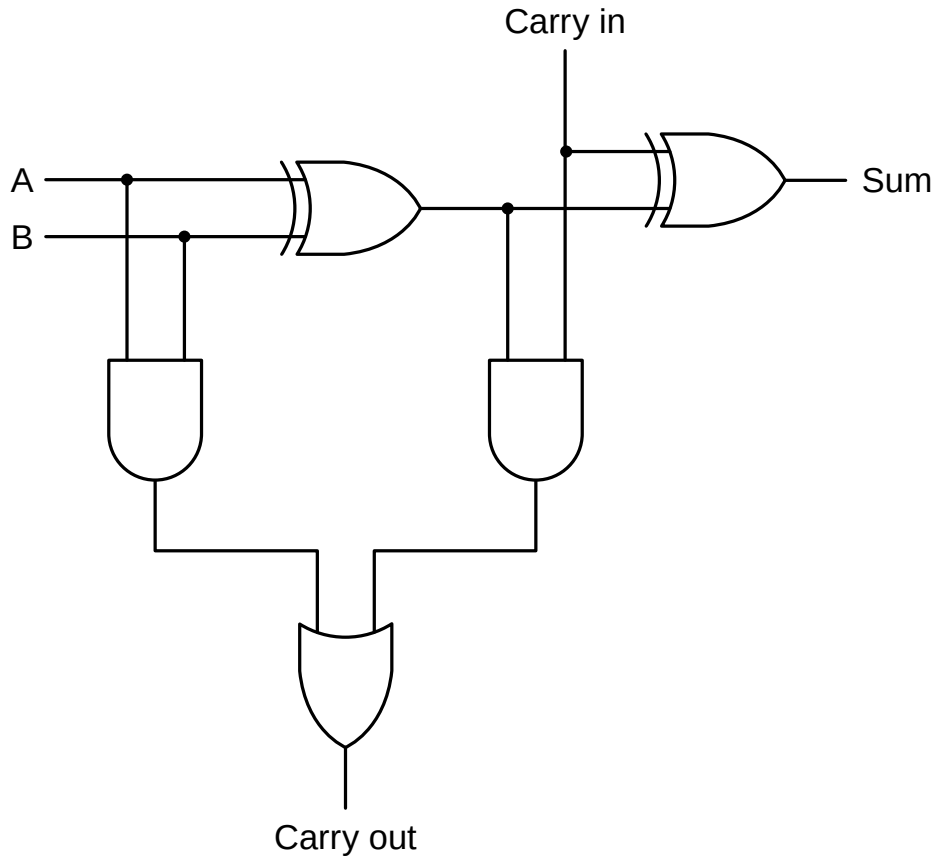


Sommatore completo

Accetta in ingresso anche il riporto della somma dalla cifra di peso immediatamente minore

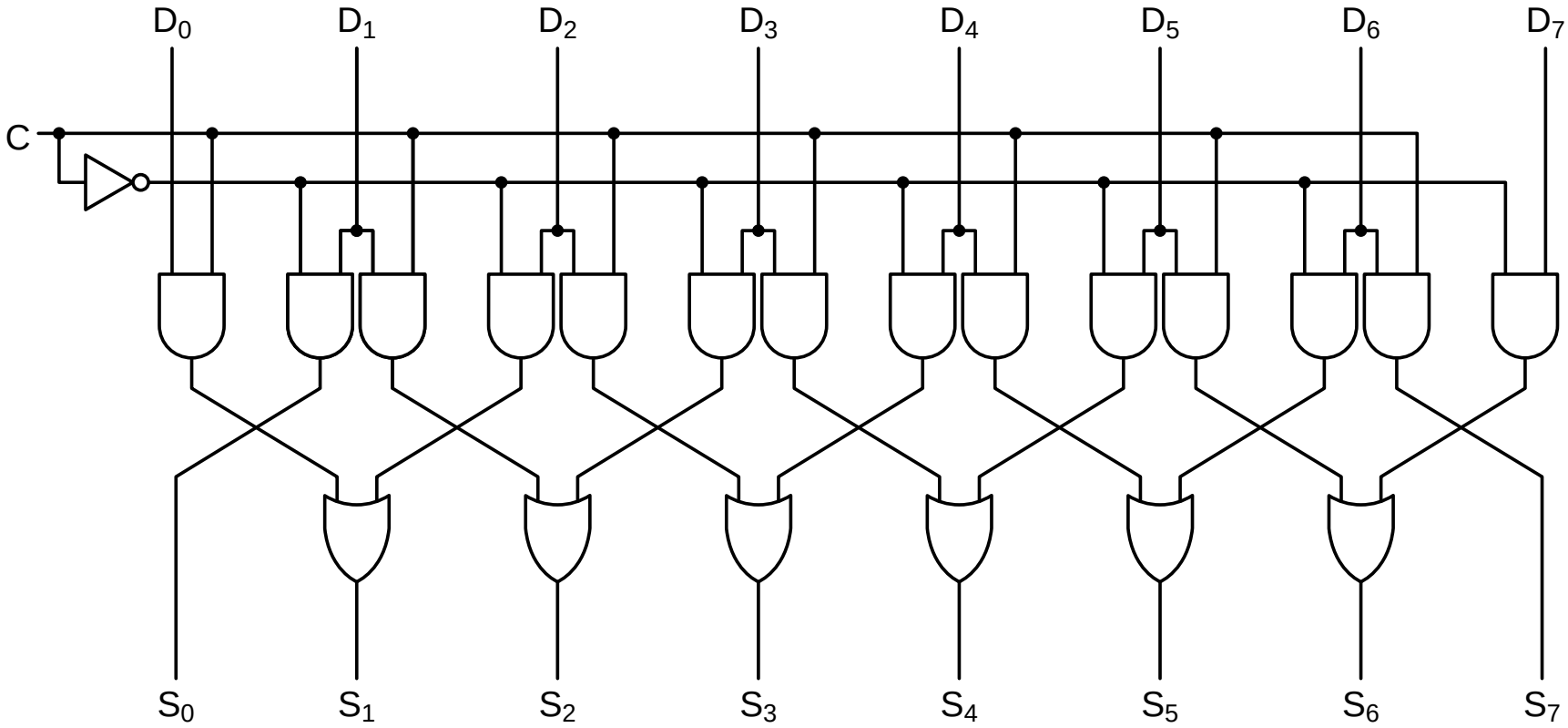
A	B	Carry in	Sum	Carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(a)



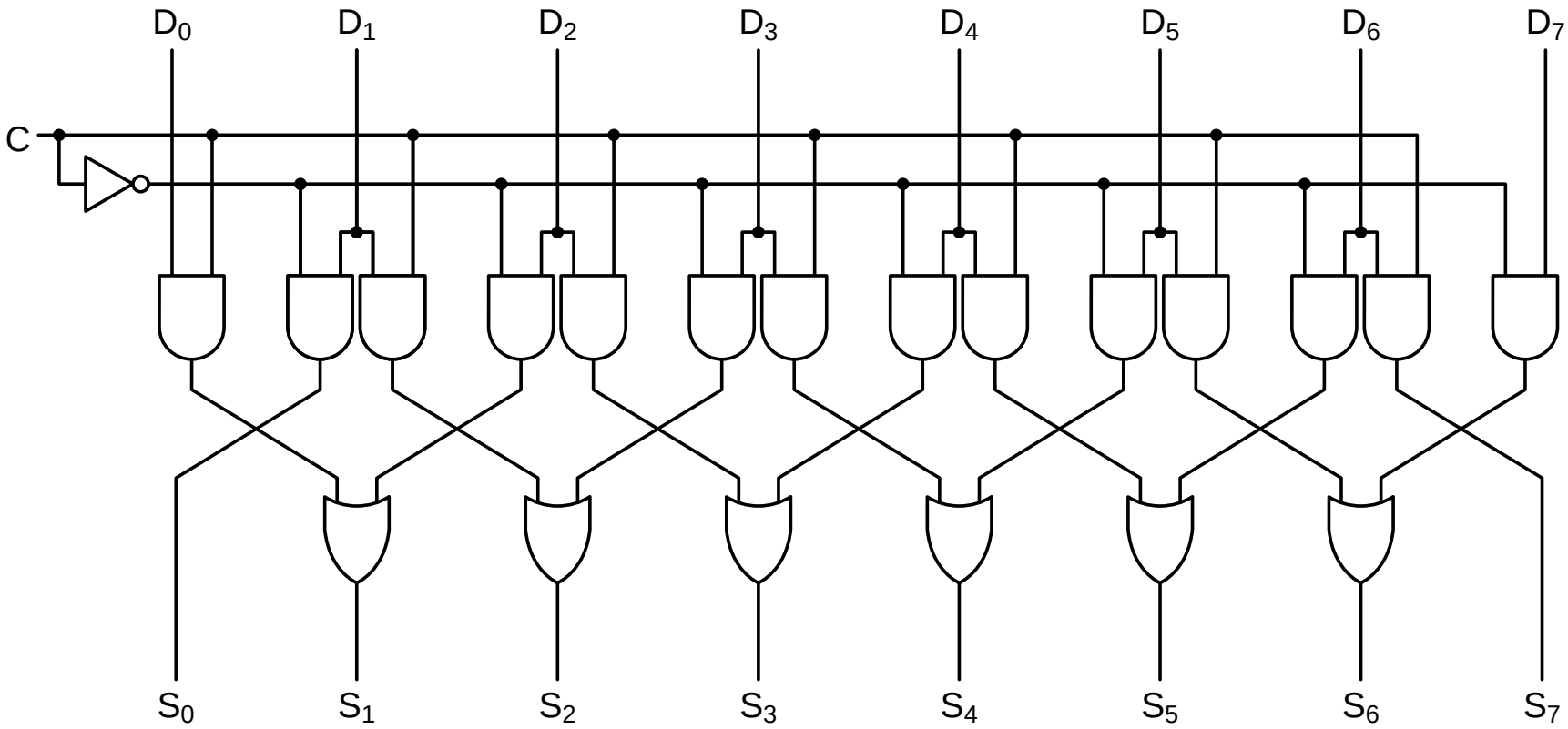
(b)

Shifter (traslatore)



Traslazione dei bit a dx ($C = 1$) oppure a sx ($C = 0$).

Shifter (traslatore)



Traslazione dei bit a dx ($C = 1$) oppure a sx ($C = 0$).

Uso: moltiplicatore per 2 oppure divisore per 2.

$$\left(\sum_{i=-k}^{n} d_i \cdot 2^i\right) \cdot 2 = \sum_{i=-k}^{n} d_i \cdot 2^{i+1} = \sum_{i=-k+1}^{n+1} d_{i-1} \cdot 2^i.$$

Propagazione del ritardo

- I circuiti logici rispondono con un piccolissimo ritardo: circa 10^{-10} s
- in presenza di porte logiche in cascata i ritardi si sommano
- la realizzazione tradizionale del sommatore richiede una lunga cascata per la propagazione del riporto: implementazione lenta.

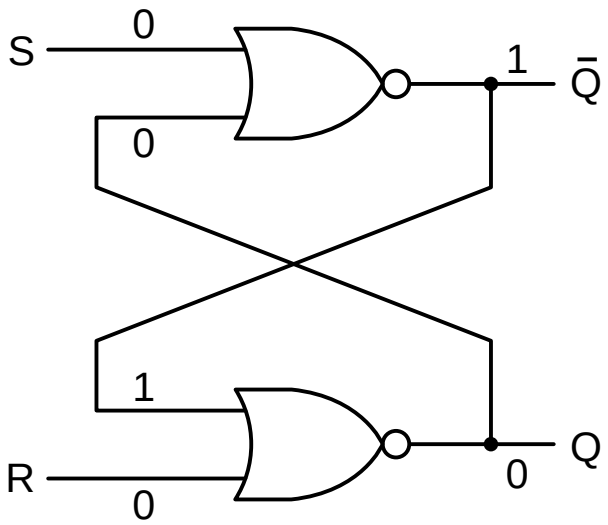
Per ottenere circuiti più veloci la somma usa circuiti più sofisticati, che non vediamo.

Circuiti con memoria

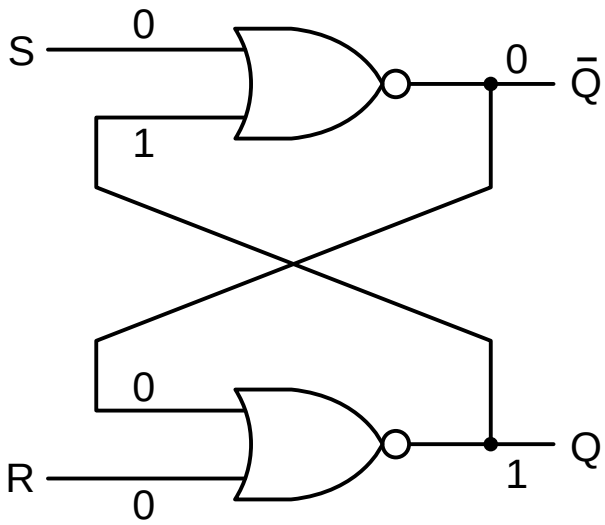
Circuiti con memoria: il comportamento passato ha effetti sull'uscita all'istante presente.

Retroazione: riutilizzo dell'uscita nell'ingresso.

Il più semplice circuito con memoria è il latch set–reset (SR).



(a)



(b)

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

(c)

Latch SR

- Il segnale S (set) a 1 porta l'uscita Q a 1.
- Il segnale R (reset) a 1 porta l'uscita Q a 0.

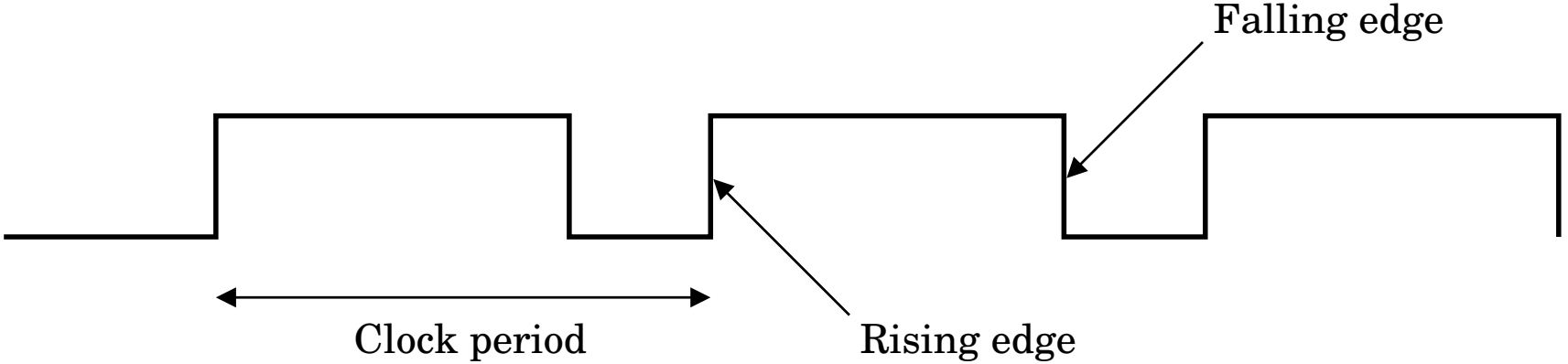
Posso memorizzare un bit.

Infatti, cessato il segnale d'ingresso, con input $S = 0$, $R = 0$ assume uno tra due stati stabili complementari in dipendenza dal passato.

Nota: l'input $S = 1$, $R = 1$ forza le due uscite complementari ad assumere lo stesso valore 0, incoerente rispetto al funzionamento del latch SR.

Clock

Segnale periodico che scandisce il funzionamento dei circuiti sequenziali. Tipicamente un'onda quadra.



Segnale periodico: esiste un intervallo temporale detto periodo in cui il segnale si ripete identico. Tipicamente ogni periodo contiene un valore di tensione alto e uno basso.

Frequenza di clock [Hz] = 1 / periodo di clock.

Periodo di clock

Vincoli contrapposti:

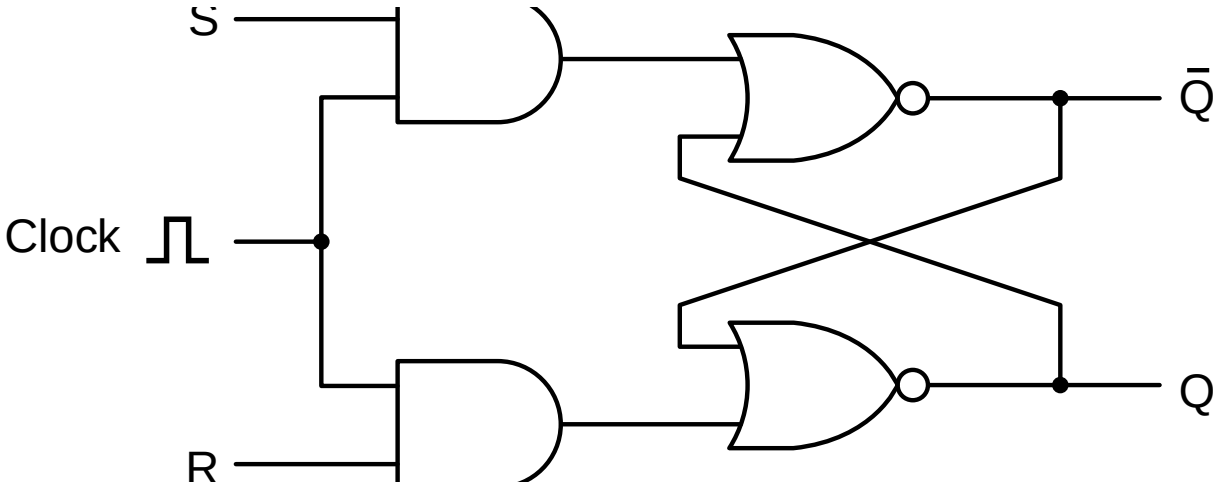
- la massimizzazione delle prestazioni vorrebbe il periodo di clock più breve possibile
- ogni circuito ha un tempo di commutazione che non può essere superiore al periodo di clock.

Ordine di grandezza del periodo di clock: 1$\sim$10 ns, corrispondenti a una frequenza di 100 MHz$\sim$1 GHz.

In un calcolatore coesisitono vari segnali di clock, per sincronizzare processore, scheda grafica, bus di sistema, ...

Latch SR sincronizzato

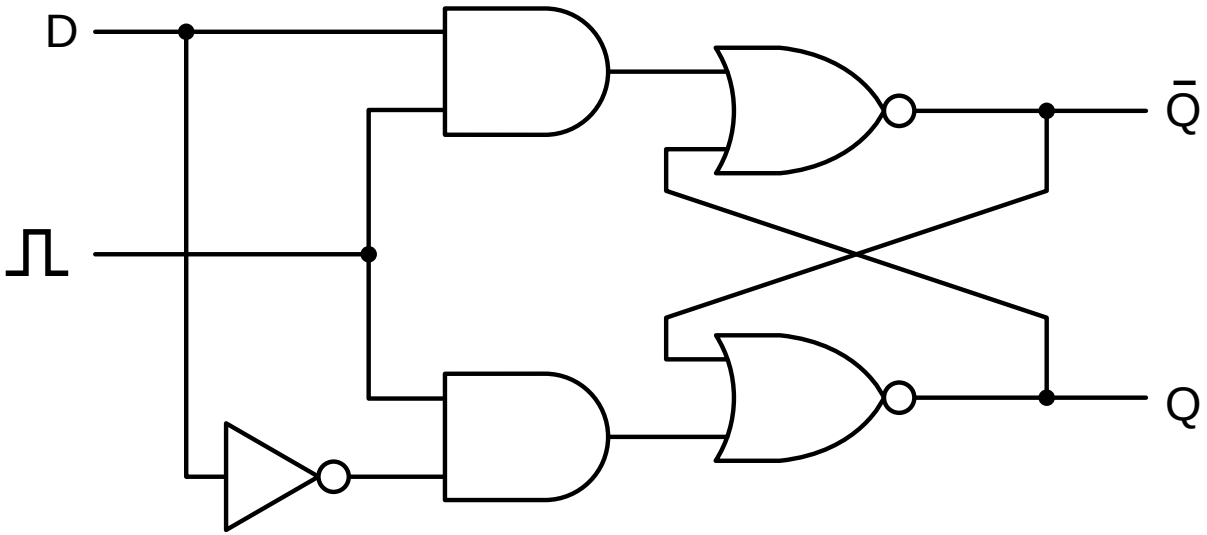
Nei latch la funzione del clock è abilitare la scrittura (segnali di enable, strobe).

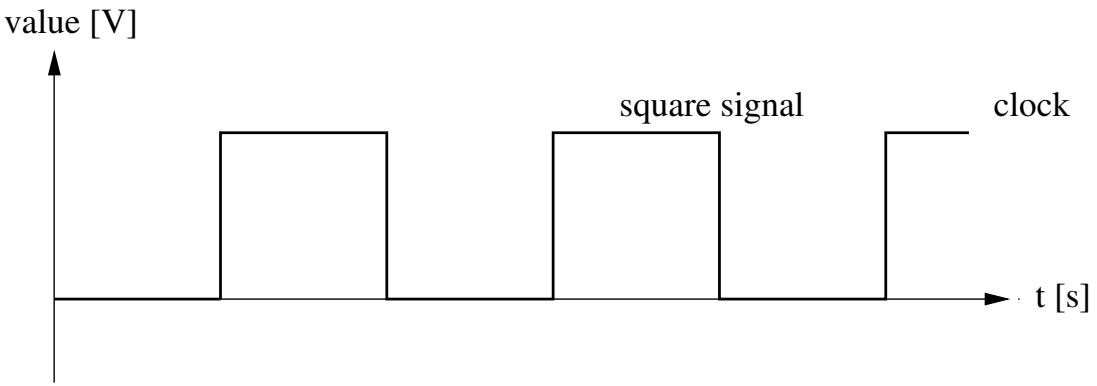


Quando il segnale di clock ha tensione bassa la scrittura è disabilitata. Il funzionamento può essere reso facilmente complementare negando il clock.

Latch D sincronizzato

Il set/reset è reso da un segnale D unico e non ambiguo.

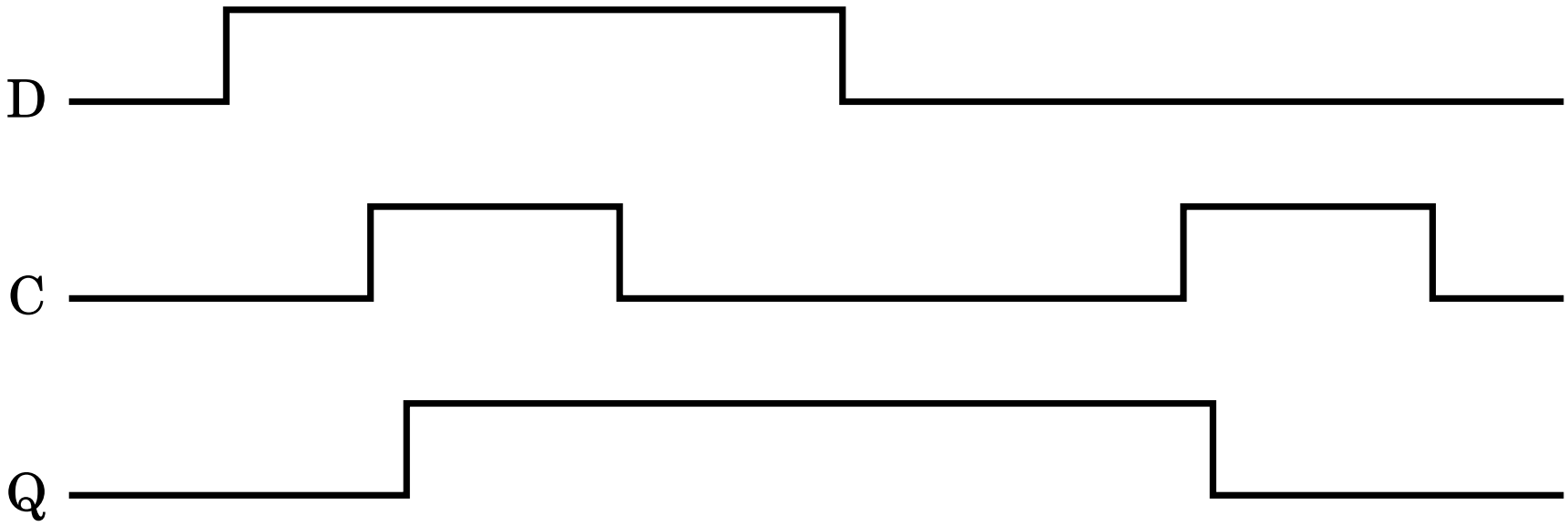




Flip-flop

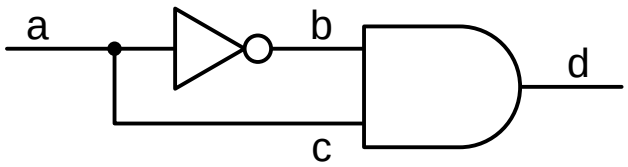
Diversi dai latch per il comportamento rispetto al clock: possono cambiare stato solo nell'istante in cui il clock cambia valore.

Es.: comportamento di un flip-flop di tipo D

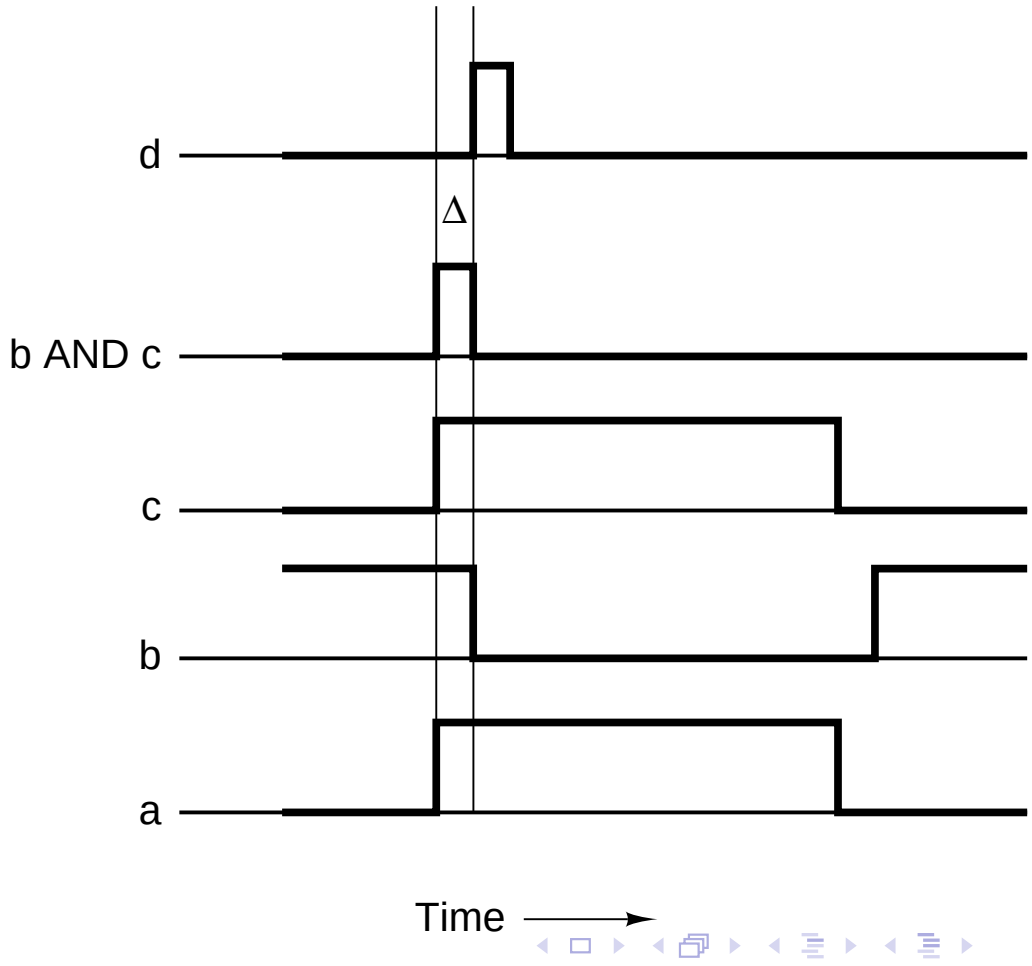


Realizzazione di un clock a impulsi

Si sfruttano i ritardi delle porte logiche per generare un segnale 1 brevissimo (impulso) da un segnale a onda quadra:

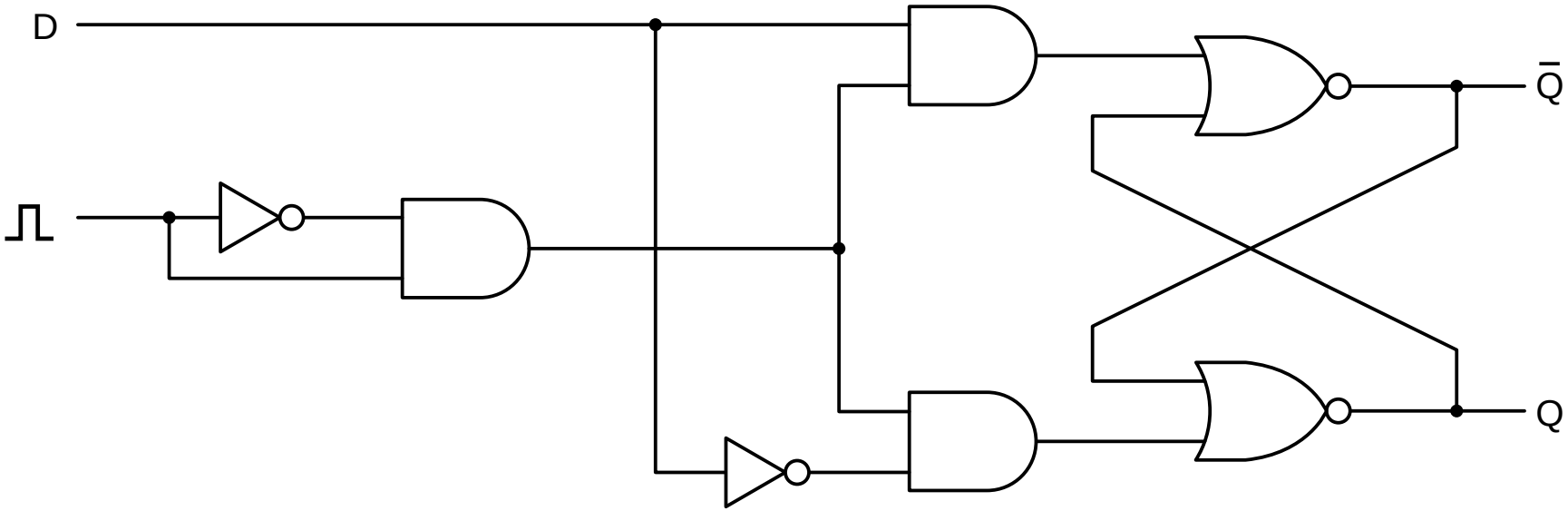


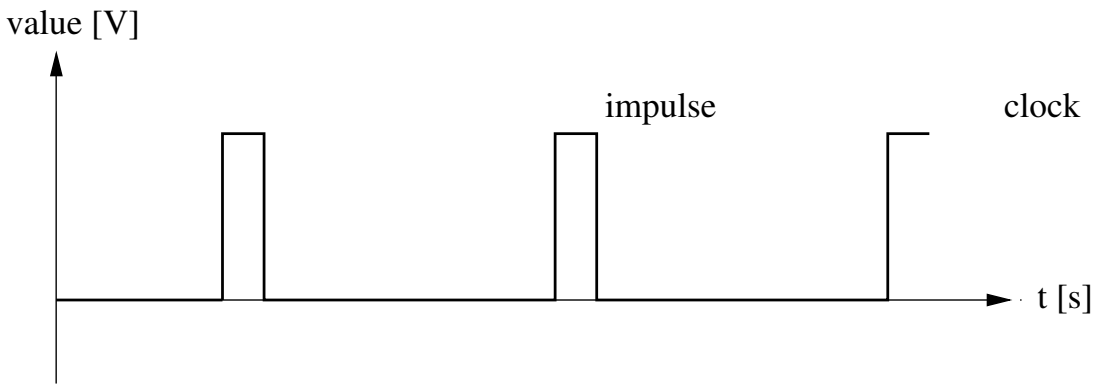
(a)



Flip-flop D sincronizzato

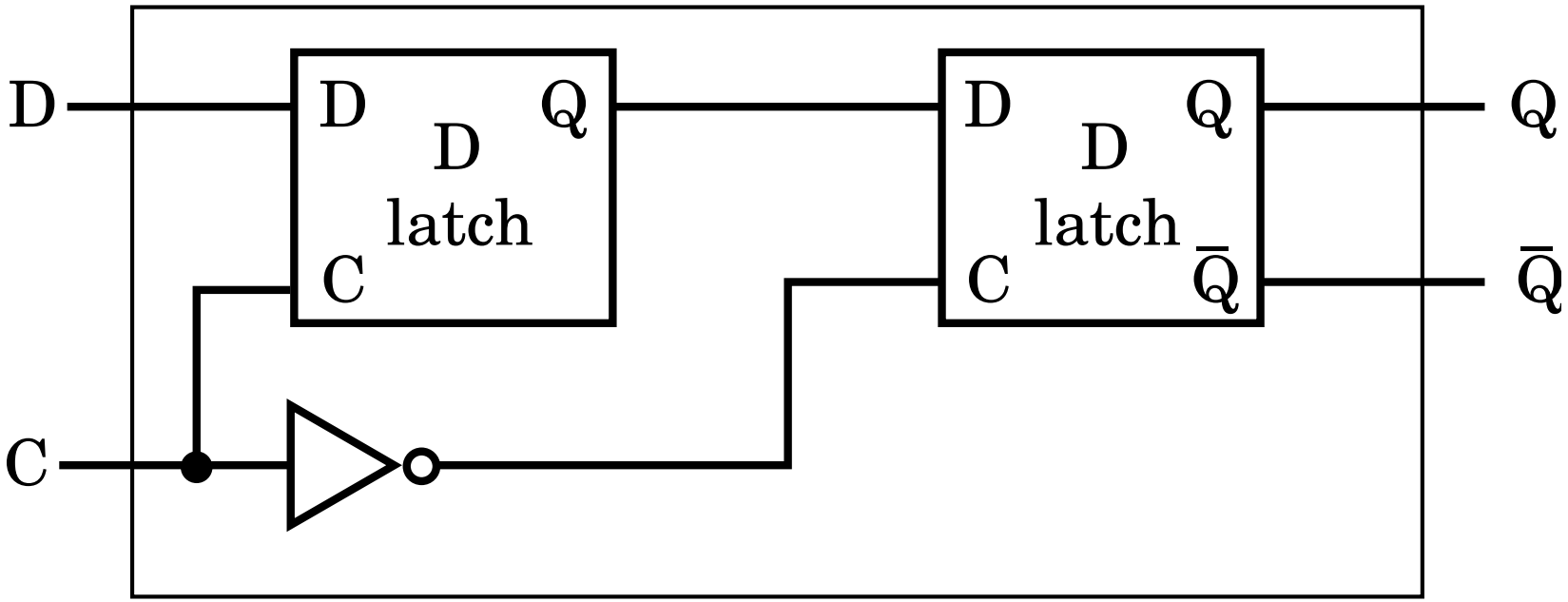
La scrittura è abilitata solo durante l'impulso





Realizzazioni master-slave

Più affidabili, in quanto rinforzano ulteriormente la stabilità del circuito: lo slave fintantoche è abilitato riceve dal master un ingresso stabile.



Confusione terminologica: il termine “latch” non viene adoperato nelle configurazioni master-slave.

Tipi di latch e flip-flop

Classificazione in base al tipo di clock:

- latch: level-triggered (azionato dal livello)
- flip-Flop: edge-triggered (azionato dal fronte).

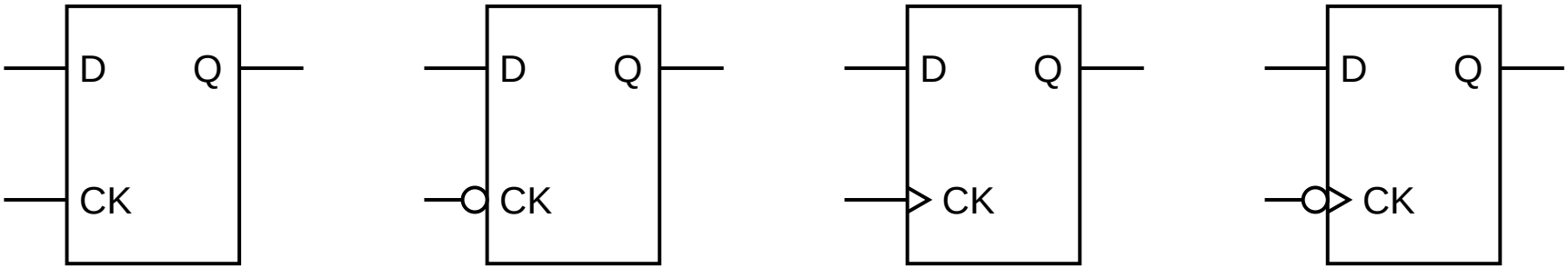
Vari tipi:

- S-R
- D
- J-K: come S-R ma cambia stato con J=1, K=1
- T: come D ma cambia stato con T=1.

N.B: non tutti concordano sulla classificazione e terminologia qui adottata!

Rappresentazione grafica

Alcuni esempi:

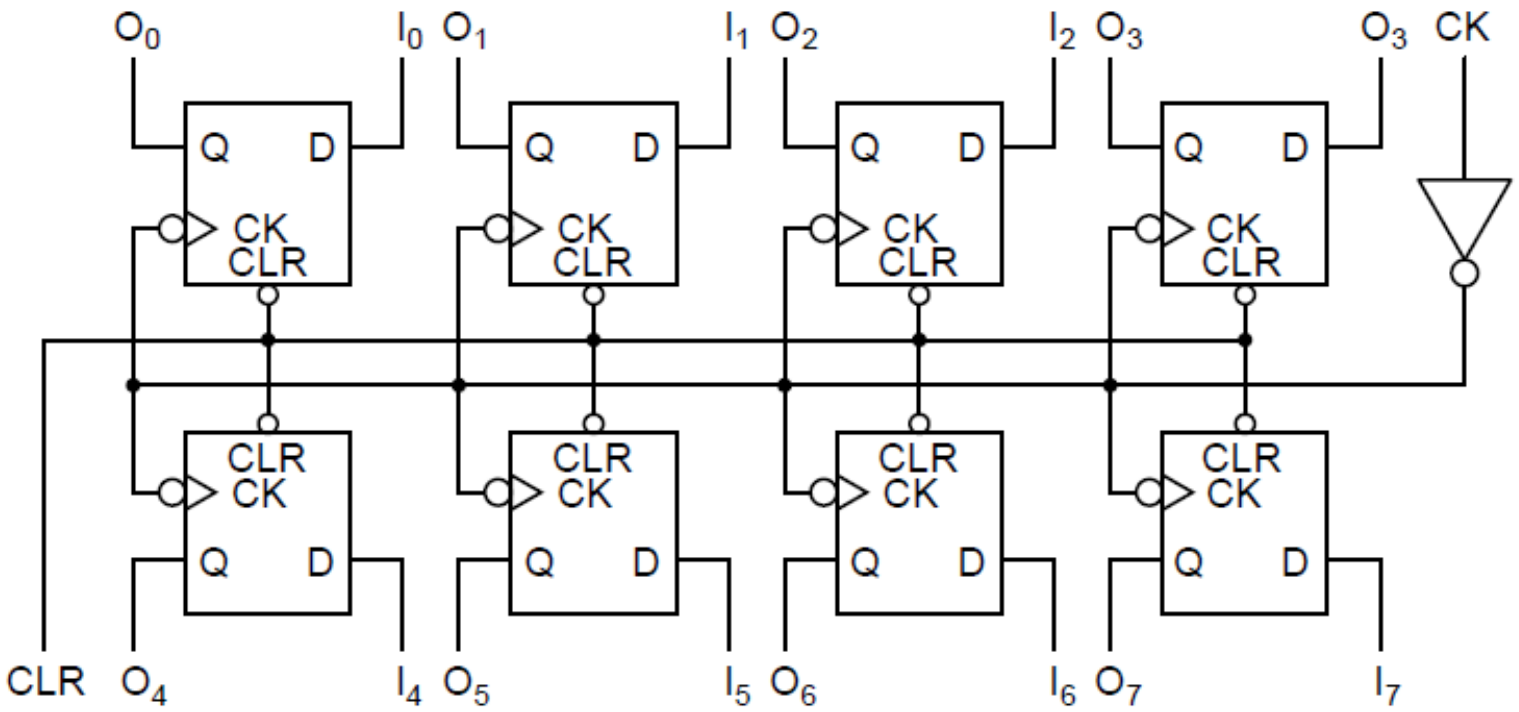


- (a): latch D
- (b): latch D con clock complementare
- (c): flip-flop D
- (d): flip-flop D con clock complementare.

Registri

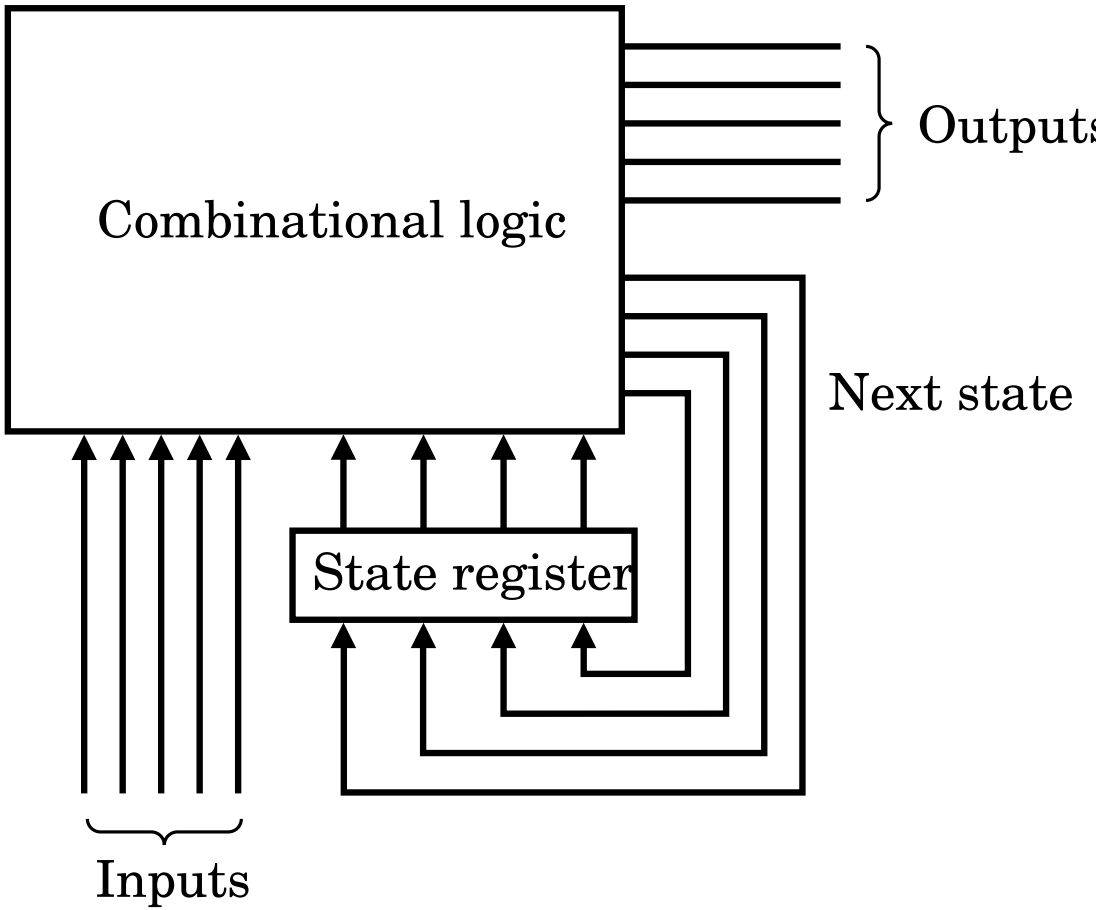
Un registro è in grado di memorizzare sequenze di N bit (tipicamente $N = 8, 16, 32, 64$).

Realizzazione: gruppo di N flip-flop sincronizzati mediante uno stesso segnale di clock.



Realizzazione di circuiti sequenziali

L'uscita dipende dall'ingresso e dallo stato:



$$\begin{cases} \boldsymbol{y}_n = g(\boldsymbol{x}_n, \boldsymbol{s}_n) \\ \boldsymbol{s}_{n+1} = h(\boldsymbol{x}_n, \boldsymbol{s}_n) \end{cases}$$

Presenza del registro di stato.

Evoluzione del circuito sequenziale

A ogni ciclo di clock il contenuto del registro può cambiare. Ciò corrisponde a una variazione della configurazione dello stato.

Di conseguenza, a ogni ciclo di clock può cambiare:

- il valore dell'uscita
- la configurazione dello stato.

Evoluzione del circuito sequenziale

A ogni ciclo di clock il contenuto del registro può cambiare. Ciò corrisponde a una variazione della configurazione dello stato.

Di conseguenza, a ogni ciclo di clock può cambiare:

- il valore dell'uscita
- la configurazione dello stato.

Lo stato del circuito dunque evolve in ogni caso.

Una variazione nell'input e/o nello stato genera questa evoluzione.

L'input è sincrono rispetto al segnale di clock: input e stato agiscono simultaneamente sulla parte combinatoria del circuito.

Progettazione di circuiti sequenziali

Circuito sequenziale = circuito combinatorio + memoria + clock.

La progettazione di semplici circuiti sequenziali segue i passi seguenti:

1. progettazione di una macchina astratta che risolve il problema
2. quantificazione dei bit necessari a memorizzare lo stato
3. sintesi della rete combinatoria sincronizzata suggerita dalla macchina.

In generale i circuiti combinatori sono complessi, e contengono molteplici registri e circuiti combinatori.

Macchina a stati finiti

Descrizione astratta del circuito sequenziale come una macchina a stati finiti (MSF, FSM).

A ogni istante la macchina si trova in una determinata situazione, data da:

- configurazione dello stato
- valore delle variabili d'ingresso.

Questa situazione determina univocamente

- il valore delle variabili d'uscita
- la configurazione dello stato all'istante successivo.

La nozione di tempo discreto e di sincronizzazione è intrinseca alla definizione di MSF.

Rappresentazione di MSF

Una macchina a stati finiti è rappresentata da un grafo: nodi connessi tramite archi orientati.

- Nodo: configurazione dello stato.
- Arco orientato: transizione conseguente a un particolare ingresso.

Nella macchina di Mealy ogni arco è etichettato da

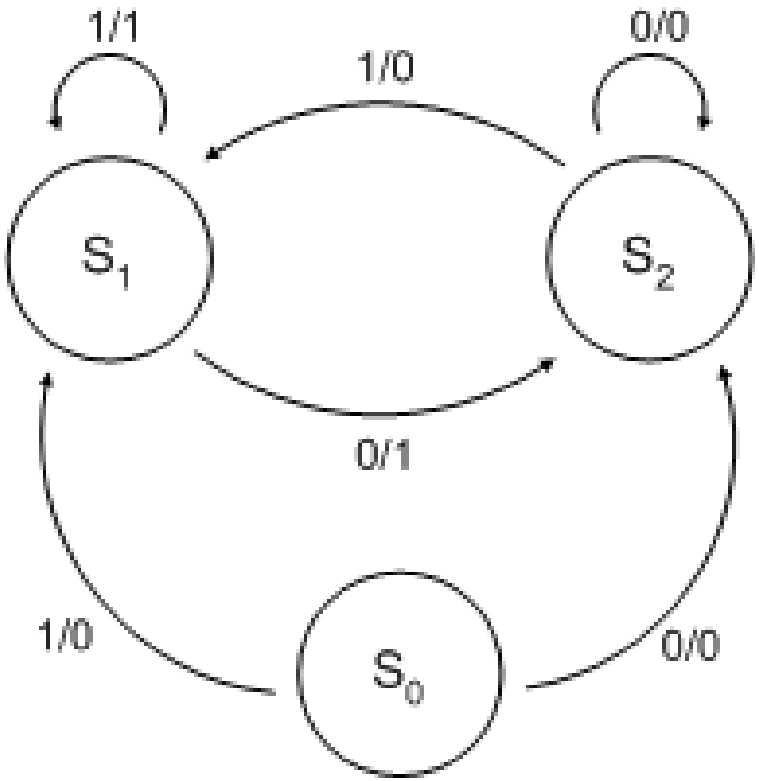
- un valore dell'input
- i valori del conseguente output.

Nella macchina di Moore ogni nodo è etichettato da

- una configurazione dello stato
- i valori del conseguente output.

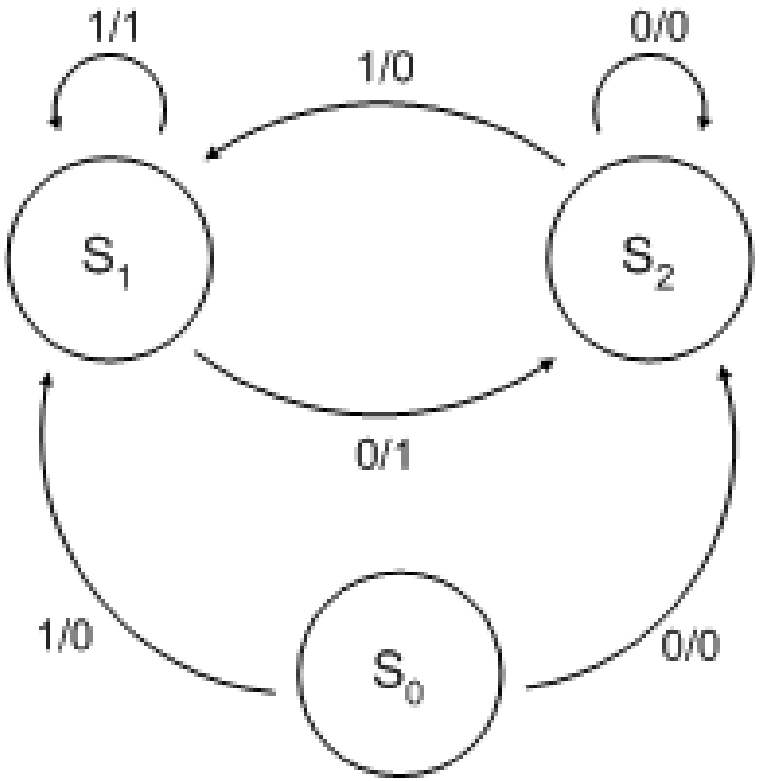
Macchina di Mealy

Es.: sia S_0 lo stato iniziale della MSF in figura. Che comportamento ha la macchina?



Macchina di Mealy

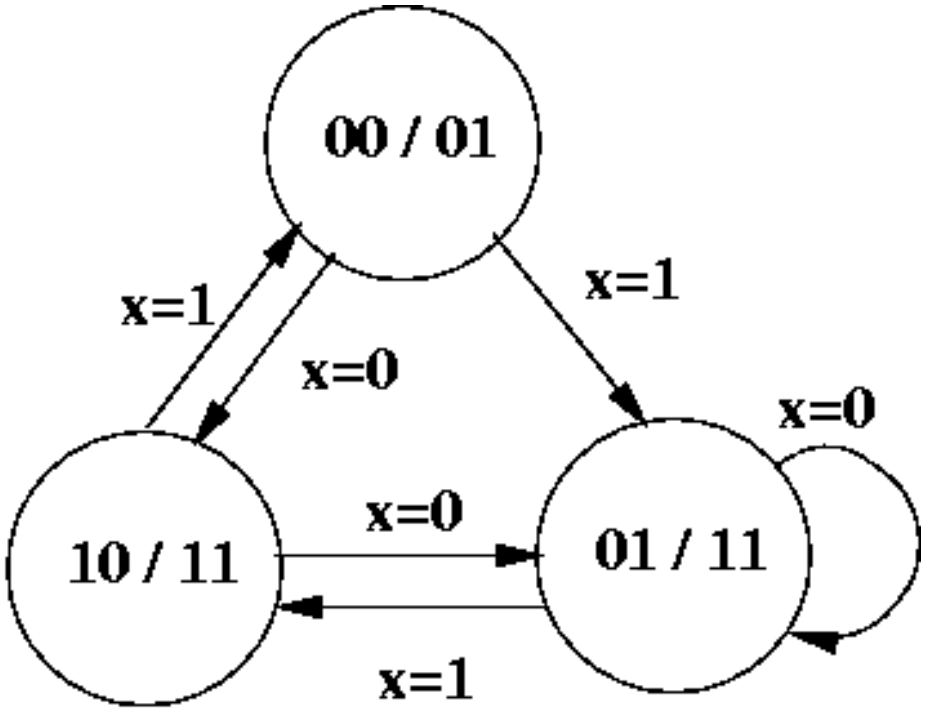
Es.: sia S_0 lo stato iniziale della MSF in figura. Che comportamento ha la macchina?



Copia l'ingresso attuale nell'uscita all'istante successivo.

Macchina di Moore

Es.:



L'output della macchina di Moore dipende esclusivamente dallo stato.

Non è meno potente della macchina di Mealy.

Progettazione della rete combinatoria

Una volta determinata una MSF dal problema,

- a ogni nodo si associa un numero binario (etichetta)
- i bit necessari a etichettare i nodi determinano l'estensione del registro di stato
- ogni bit è una variabile d'ingresso alla rete in aggiunta alle variabili di input
- output e stato successivo sono determinati da stato (Moore) o da stato + input (Mealy)
- si sintetizza il rispettivo circuito combinatorio, ad esempio utilizzando le mappe di Karnaugh
- il clock abilita l'aggiornamento dello stato
- lo stato iniziale dev'essere non ambiguo.

Esempi: semaforo passivo

Circuito per il controllo di un semplice semaforo

- il semaforo cambia stato ad ogni ciclo di clock (es.: 30 secondi)
- due configurazioni: rosso oppure verde.

Esempi: semaforo passivo

Circuito per il controllo di un semplice semaforo

- il semaforo cambia stato ad ogni ciclo di clock (es.: 30 secondi)
- due configurazioni: rosso oppure verde.

Controllo di un semaforo con tempo del verde diverso per le due strade

- tempo di verde su una strada frazione del tempo di verde sull'altra strada.

Esempi: semaforo passivo

Circuito per il controllo di un semplice semaforo

- il semaforo cambia stato ad ogni ciclo di clock (es.: 30 secondi)
- due configurazioni: rosso oppure verde.

Controllo di un semaforo con tempo del verde diverso per le due strade

- tempo di verde su una strada frazione del tempo di verde sull'altra strada.

Circuito di controllo di un "vero" semaforo

- inclusione della configurazione verde+giallo.

Esempi: semaforo attivo

Controllo di un semplice semaforo dotato di rilevatori di traffico

- lo stato del semaforo cambia solo se sono presenti dei veicoli in attesa
- due configurazioni: rosso oppure verde.

Traccia della soluzione

- 2 variabili di input: presenza di traffico sulla strada 1, presenza di traffico sulla strada 2
- 1 variabile di stato
- uscita corrispondente allo stato.

Esercizi su circuiti sequenziali

- Contatore “up/down” a 2 bit:
  - 2 ingressi: *A* abilita il conteggio, *B* determina il verso (crescente o decrescente) del conteggio
  - 2 variabili di stato per il contatore
  - uscita = valore del contatore.
- Circuito sequenziale per riconoscere una stringa di cifre binarie. Es.: 1100.
- Contatore di “uni” negli ultimi tre valori dell’ingresso
  - a ogni nuovo ingresso considera la terna formata dai 3 bit in ingresso più recenti
  - 2 uscite: assumono il valore 00 in corrispondenza del primo e del secondo istante di funzionamento; poi di volta in volta restituiscono il numero di bit uguali a uno contenuti nella terna più recente.

Esercizi su circuiti sequenziali

- Progettare un circuito sequenziale che è in grado di riconoscere tutte e sole le sequenze in una stringa binaria contenenti ripetizioni non vuote della sottostringa 01. Il circuito produce in uscita 1 non appena il riconoscimento di una di queste sequenze termina; 0 altrimenti.
  Es.: . . . 11010100101000011 . . .
- Adoperando una mcchina di Moore, progettare un circuito sequenziale che riconosce ogni ricorrenza disgiunta di caratteri doppi appartenenti all'alfabeto $A = \{a, b, c\}$. Il circuito produce 1 non appena il riconoscimento di due caratteri identici termina; 0 altrimenti.

Circuiti integrati

Circuiti integrati (Integrated Circuit, IC, chip): unità fisicamente identificabili contenenti circuiti logici.

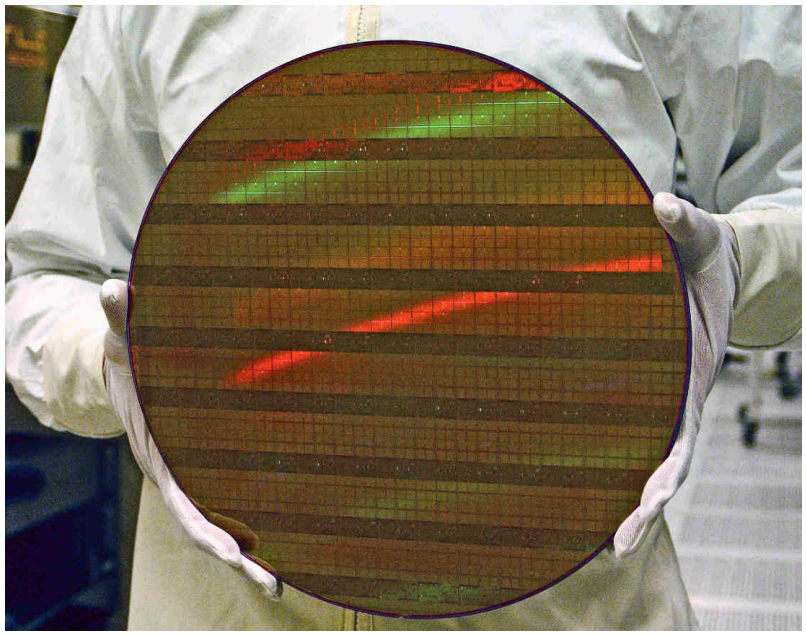
- Si parte da una piastrina di silicio di qualche cm^2
- sulla sua superficie vengono formati transistor, resistori, condensatori e i relativi collegamenti
- i componenti elettronici sono ottenuti esponendo il cristallo, in forno, ai vapori di altre sostanze: boro, arsenico, fosforo (drogatura)
- i collegamenti tra componenti sono ottenuti depositando uno strato di materiale conduttore (rame o alluminio)
- gli isolamenti elettrici sono ottenuti ossidando il silicio in forno a ossigeno.

Tecniche di fotolitografia

Il silicio viene selettivamente coperto con uno strato conduttivo di materiale fotosensibile, e poi illuminato in maniera differenziata.

- La parte illuminata solidifica e forma il circuito.
- la parte al buio viene rimossa.

Anche 50 diverse lavorazioni per singolo chip.

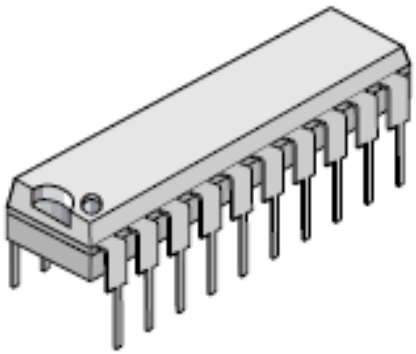


wafer

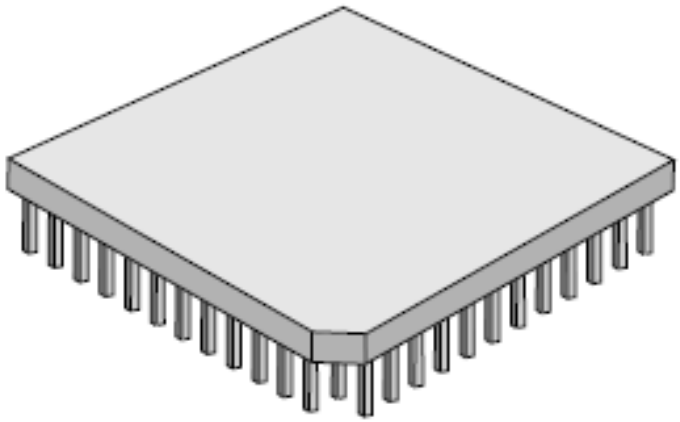
Packaging del chip e piedinatura

Ogni chip è impacchettato in un supporto isolante (plastica): package. Connessioni mediante piedini:

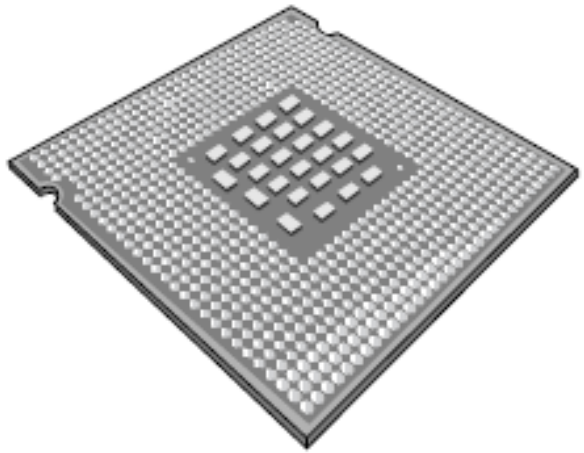
- chip di memoria e semplici processori: due file di piedini (dual in line package)
- processori per computer, schede video: centinaia di connessioni (pedinatura più complessa).



(a)



(b)



(c)

Chip di memoria

Circuiti integrati in grado di memorizzare un notevole numero di bit logicamente suddivisi in locazioni.

I singoli bit non sono individualmente accessibili.

Ogni locazione è individualmente accessibile tramite il suo indirizzo.

Per operare su un dato in memoria:

- si seleziona la locazione contenente il dato specificando il suo indirizzo
- si definisce l'operazione da eseguire: lettura o scrittura).

Accesso alla memoria

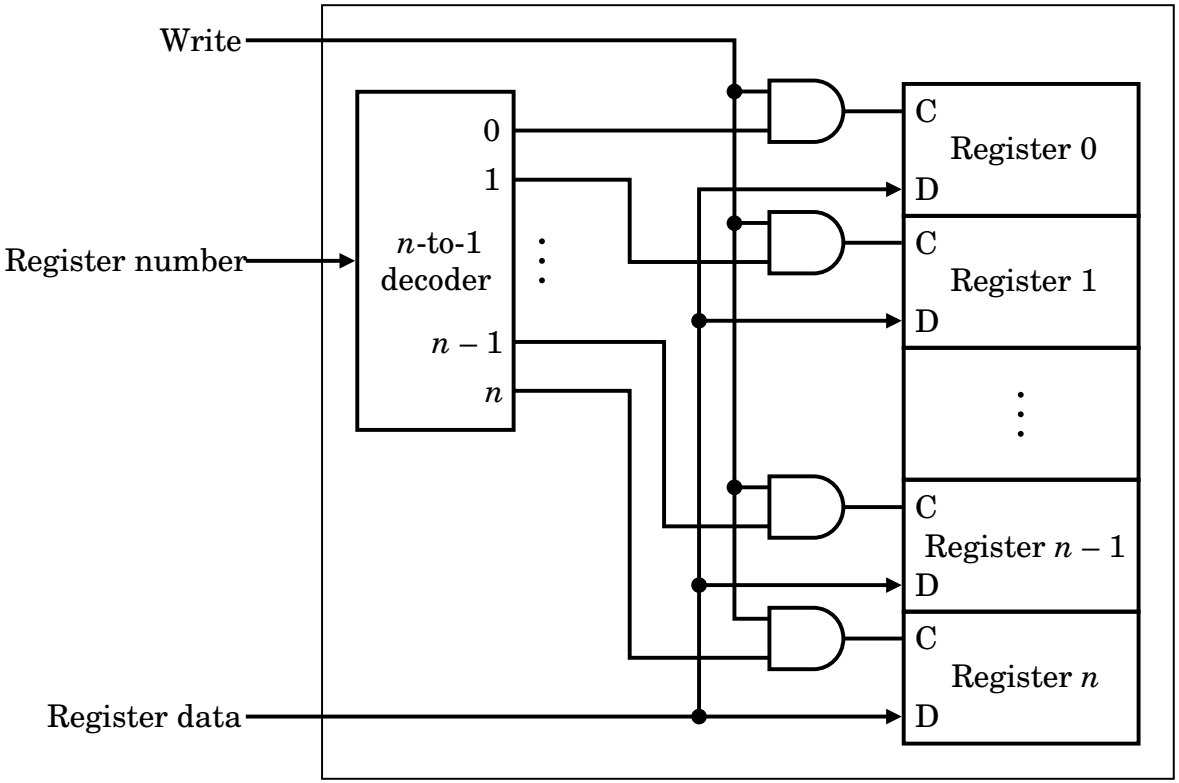
Una memoria deve assicurare l'accesso ai seguenti segnali di input/output (I/O):

- indirizzo (specifica la locazione su cui si opera)
- dato in ingresso (da scrivere nella locazione)
- dato in uscita (da leggere dalla locazione)
- segnali di controllo:
  - chip select (CS) – attiva il chip
  - read, write (R/W o RD, WR) – specifica/specificano se l'operazione è di lettura o scrittura
  - output enable (OE) – abilita l'uscita.

Ingresso e uscita quasi sempre avvengono sugli stessi piedini.

Es.: scrittura in sequenza di un dato

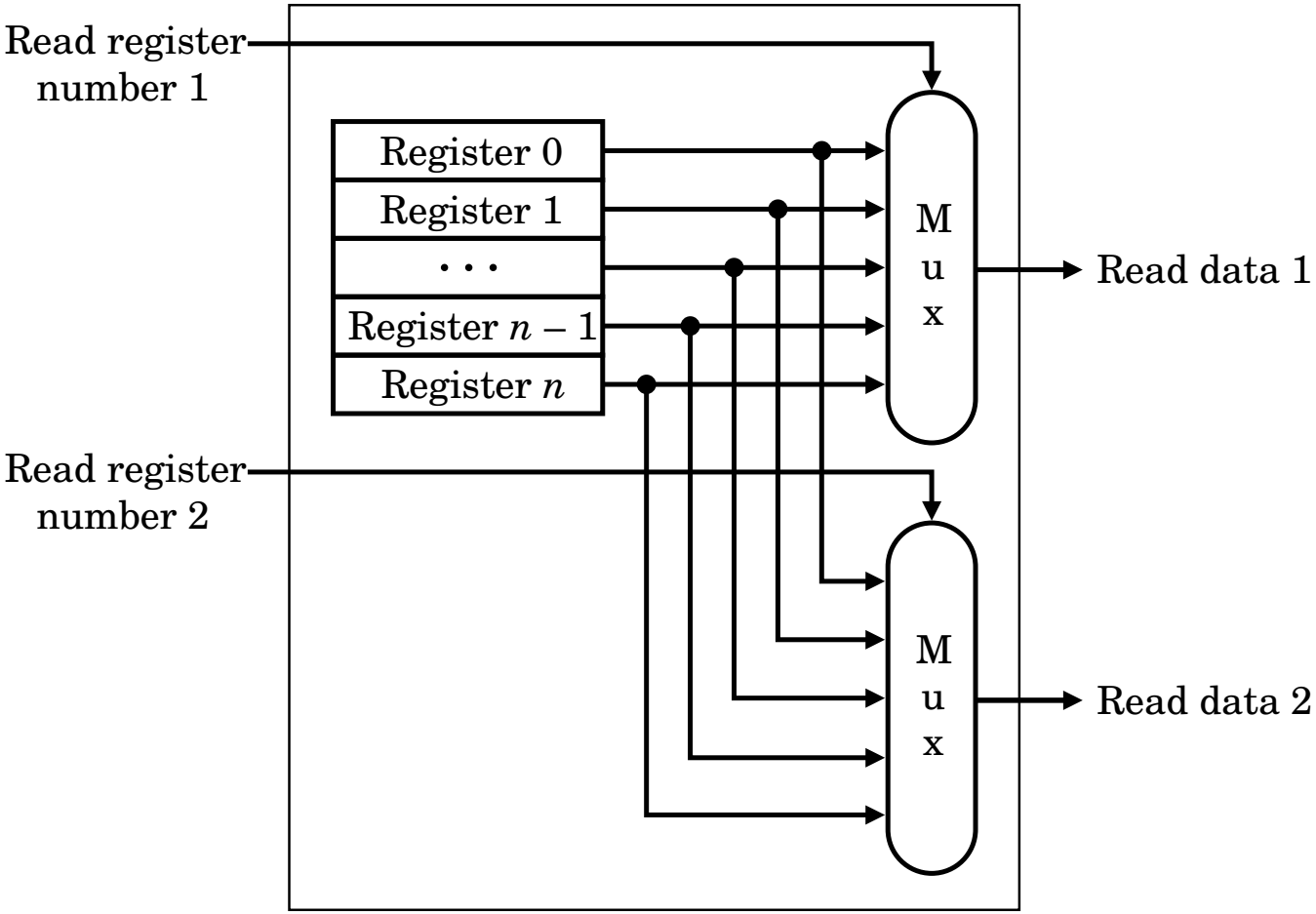
Scrittura di una sequenza di $n + 1$ bit in una locazione di memoria



N.B.: la locazione di memoria è costituita da $n + 1$ registri di un bit.

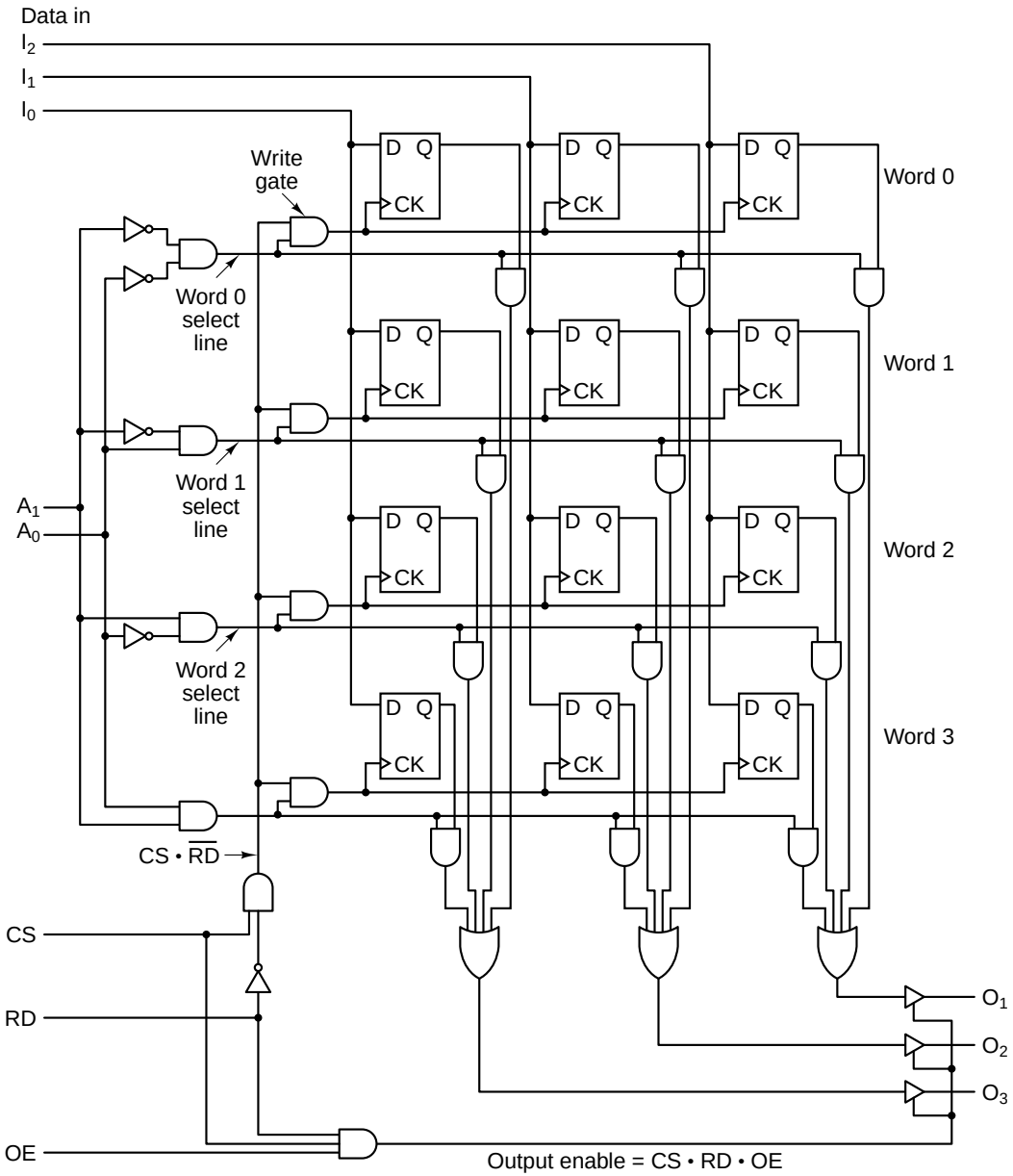
Es.: lettura in sequenza di un dato

Acquisizione del dato da $n + 1$ registri di un bit



I bit sono instradati a coppie.

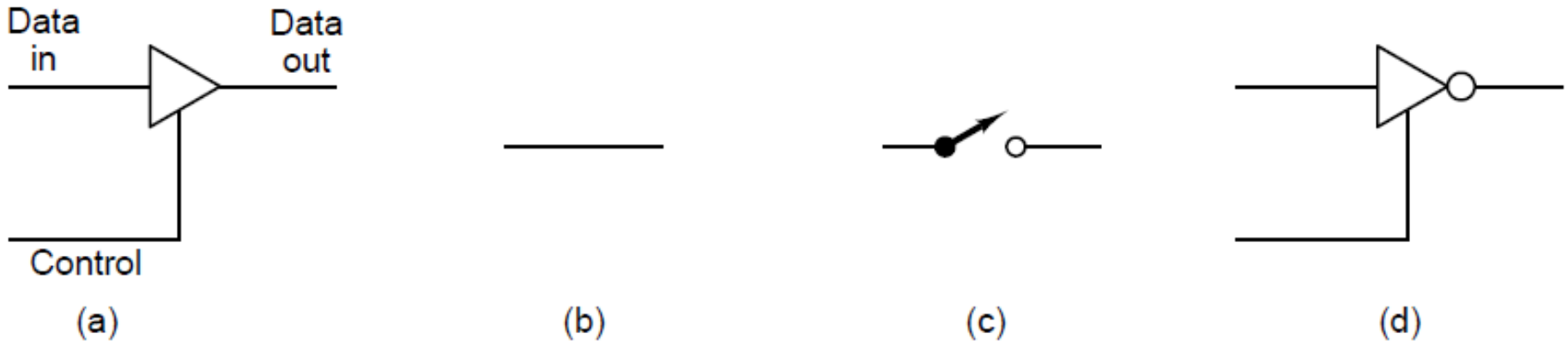
Es.: 4 locazioni di 3 bit



Buffering delle uscite

Le uscite dalle memorie non possono essere unite senza degli accorgimenti. Le porte AND sono inadeguate in quanto non permettono uno stato indeterminato dell'uscita.

Il buffer invertente equivale a una porta NOT "disattivabile", tornando utile allo scopo.



Memorie RAM

I circuiti di memoria riscrivibile vengono chiamati tradizionalmente RAM (Random Access Memory).

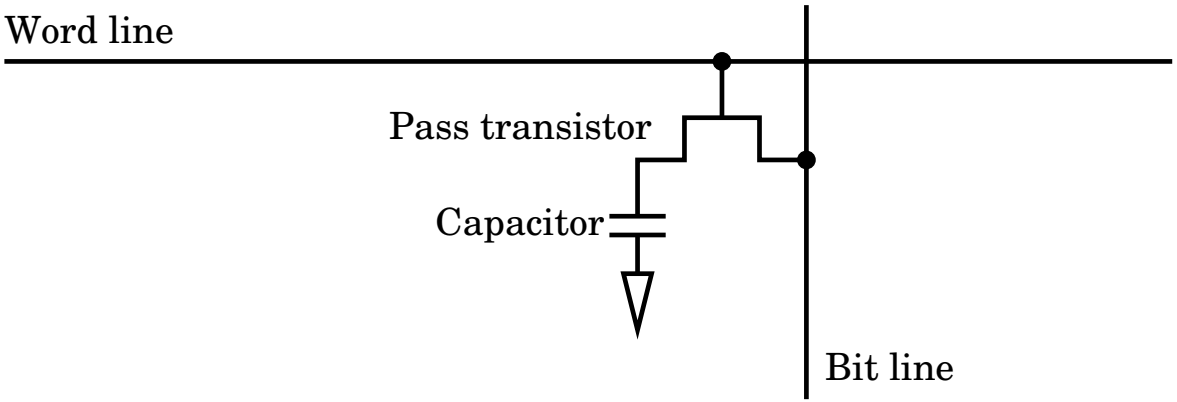
Le RAM costituiscono la memoria principale del calcolatore.

Due tipi di RAM:

- statica (SRAM): i singoli bit vengono memorizzati con latch. Veloce e costosa. Utilizzata nella memoria di basso livello
- dinamica (DRAM): usa altre tecnologie di memorizzazione. Lenta e capiente. Utilizzata nella memoria di alto livello.

RAM dinamica (DRAM)

Un singolo transistor più un condensatore per memorizzare un bit.



Il condensatore è controllato dal transistor, aperto o chiuso in base alla tensione sulla word line:

- Se il transistor è aperto il condensatore in un attimo assume la tensione sulla bit line.
- Se il transistor è chiuso il condensatore per un po' conserva il livello di tensione.

Caratteristiche DRAM

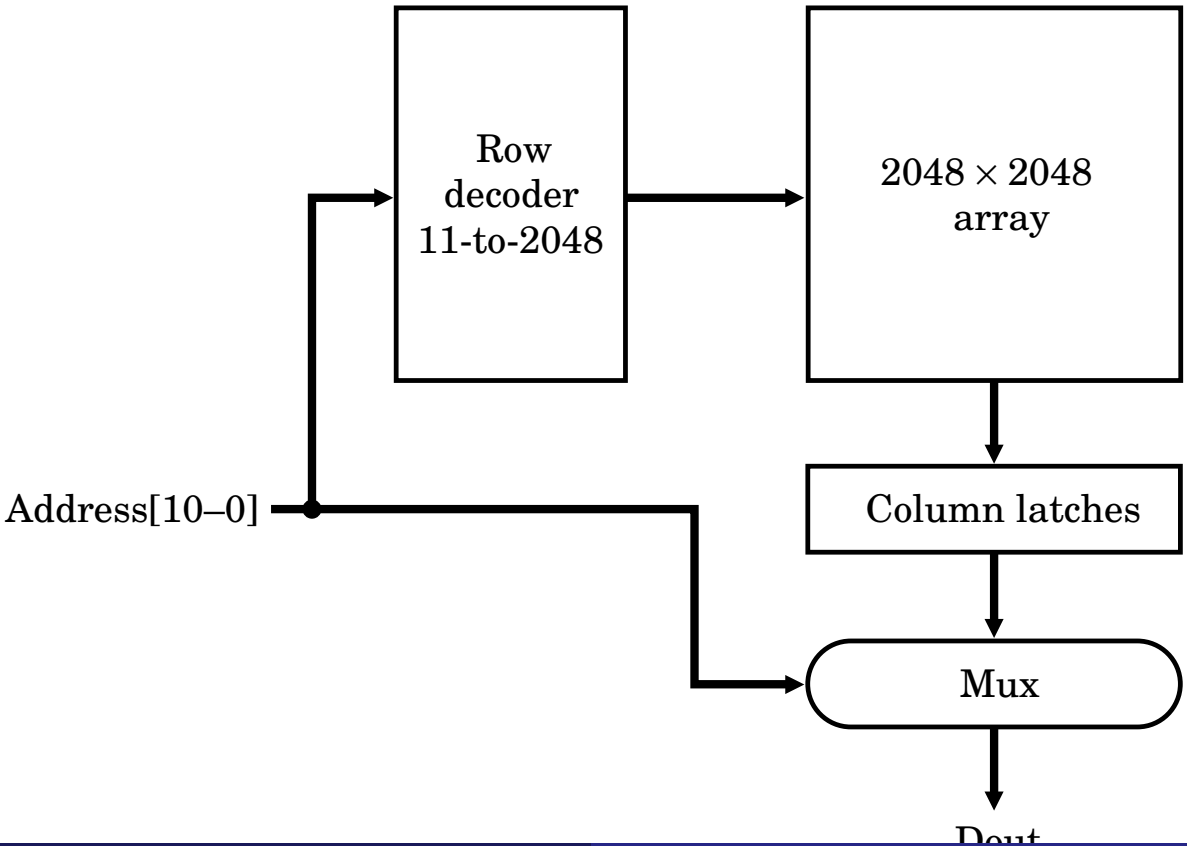
Pregi e difetti:

- più economiche e compatte delle SRAM: si possono inserire molte più locazioni per cm^2
- più lente delle SRAM
- complessità della conservazione dello stato: i condensatori si scaricano in circa 1 ms; occorre quindi dell'elettronica aggiuntiva per il refresh della carica che impegna circa il 10% di ogni ciclo di clock.

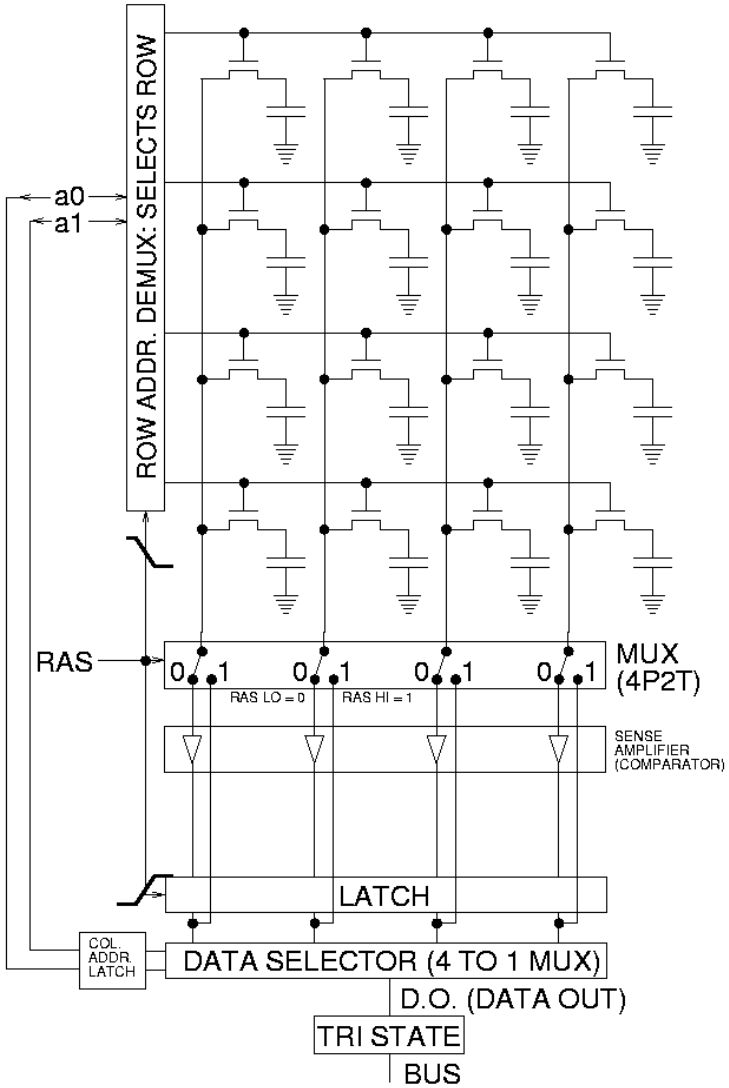
DRAM: lettura in due fasi

La lettura della DRAM normalmente avviene in due fasi:

- RAS: row access strobe
- CAS: column access strobe



Es.: DRAM di 4x4 locazioni di un bit



(by Glogger at English Wikipedia).

Tecnologie DRAM

I miglioramenti nei tempi di risposta delle DRAM sono state inferiori a quelli del processore; $\sim$ 10% vs. $\sim$ 50% di aumento in certi anni.

Risultato: processore 100-1000 volte più veloce della DRAM.

Idea: accedere quando possibile a locazioni consecutive di memoria invece che casualmente distribuite nella DRAM.

Se la riga di memoria nei latch contiene più locazioni consecutive con dati in uso a un programma allora non occorre ripetere la fase RAS a ogni accesso. Ciò velocizza di molto la lettura.

Tipi di DRAM

Evoluzione delle DRAM:

- FPM RAM (Fast page mode)
- EDO RAM (Extended data output)
- SDRAM (Synchronous DRAM)
- DDR3 SDRAM (Double Data Rate SDRAM)
- RDRAM (Direct Rambus DRAM)
- GDDR4 (Graphic Double Data Rate, schede grafiche)
- ...

Stessa tecnologia, diversi metodi di accesso.
Cambia l'interfaccia con il processore.

Double Data Rate Synchronous DRAM

- Synchronous: accesso regolato da clock.

  Trasmissione di pacchetti di dati contenuti in locazioni consecutive.

  Vantaggio: trasmissione di un pacchetto a ogni ciclo di clock.
  Svantaggio: necessari diversi cicli di clock per il primo pacchetto.
- Double Data Rate: a ogni ciclo di clock vengono spediti due pacchetti di dati.

Banda passante delle DRAM

Le DRAM hanno migliorato più la banda passante rispetto al tempo d'accesso.

- Banda passante: quantità di dati trasmessi nell'unità di tempo.
- Tempo d'accesso: tempo necessario per completare una singola operazione in memoria.

In generale non sono correlati.

NB: significato originale di Random Access Memory: non si accede a ogni dato nello stesso tempo.

Capacità della memoria

- Crescita della capacità nel numero n di periodi triennali $\approx 4^n$, secondo la legge di Moore.
- Le unità di memoria più capienti sono più costose.
- Una stessa quantità di memoria può essere distribuita su un numero variabile di unità.
- La lunghezza delle locazioni può cambiare con l'unità di memoria scelta.

Capacità di memoria: esempio

Un memoria da 1 Gbit può essere realizzata per esempio con:

- 1 G di locazioni di 1 bit
- 512 M di locazioni da 2 bit
- 256 M di locazioni da 4 bit
- 128 M di locazioni da 8 bit.

Organizzazioni diverse portano a diverse quantità di

- linee indirizzo (indirizzo locazione)
- linee dato (contenuto locazione).

Capacità =

Capacità di memoria: esempio

Un memoria da 1 Gbit può essere realizzata per esempio con:

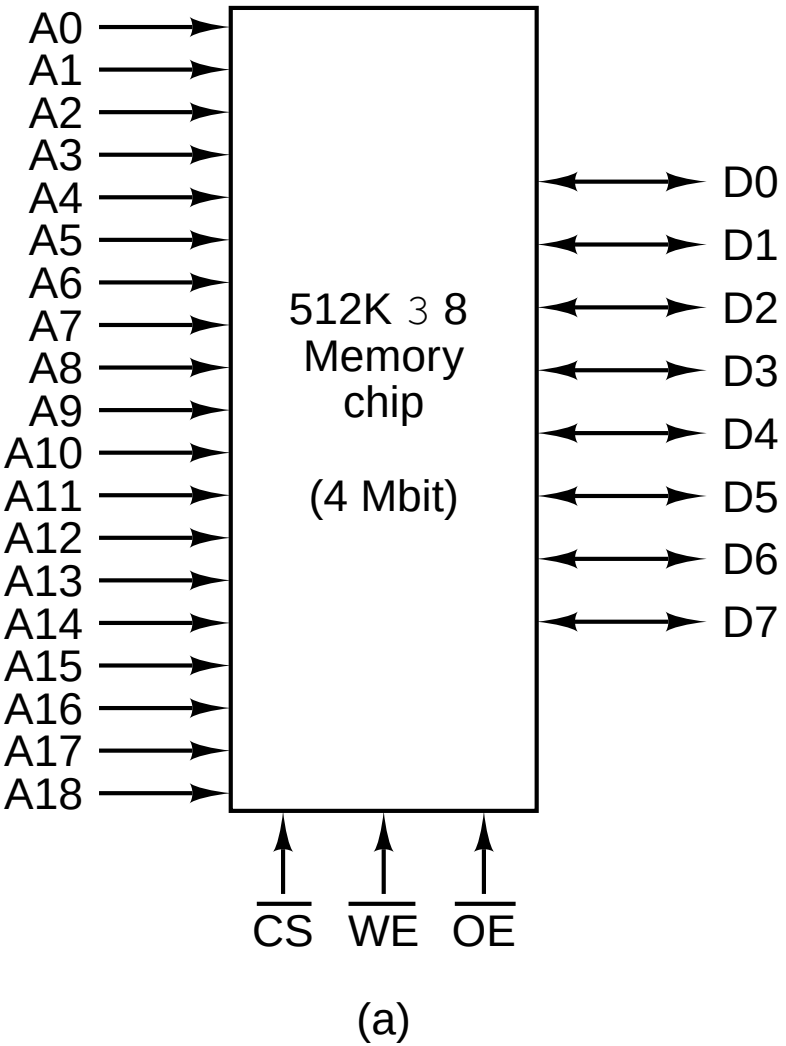
- 1 G di locazioni di 1 bit
- 512 M di locazioni da 2 bit
- 256 M di locazioni da 4 bit
- 128 M di locazioni da 8 bit.

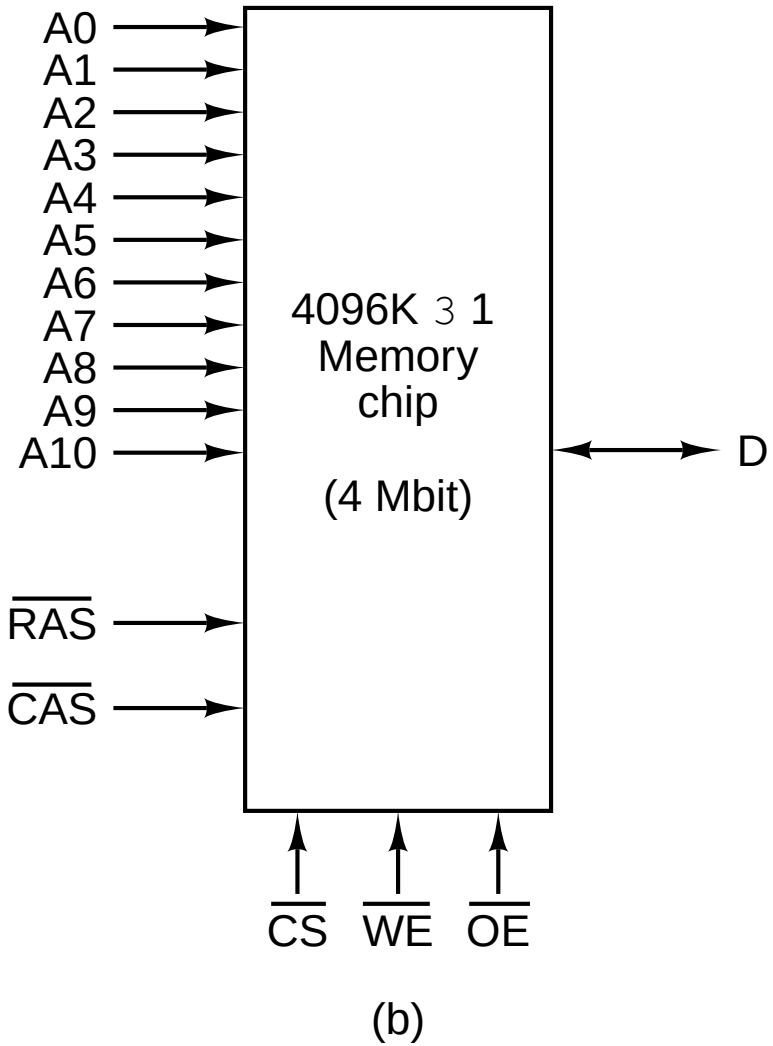
Organizzazioni diverse portano a diverse quantità di

- linee indirizzo (indirizzo locazione)
- linee dato (contenuto locazione).

Capacità = $2^{\text{linee indirizzo}} \times$ linee dato.

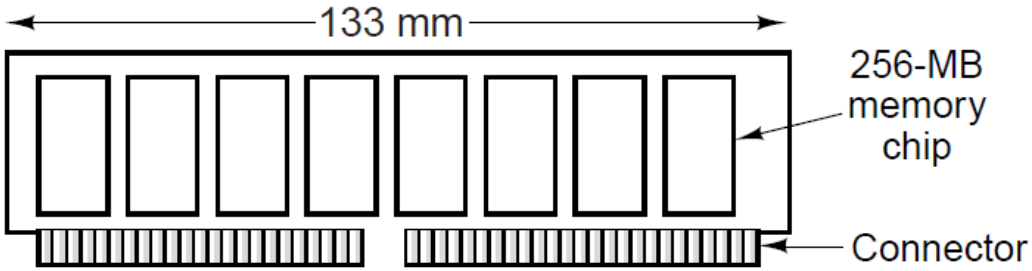
Es.: due unità di 4 Mbit

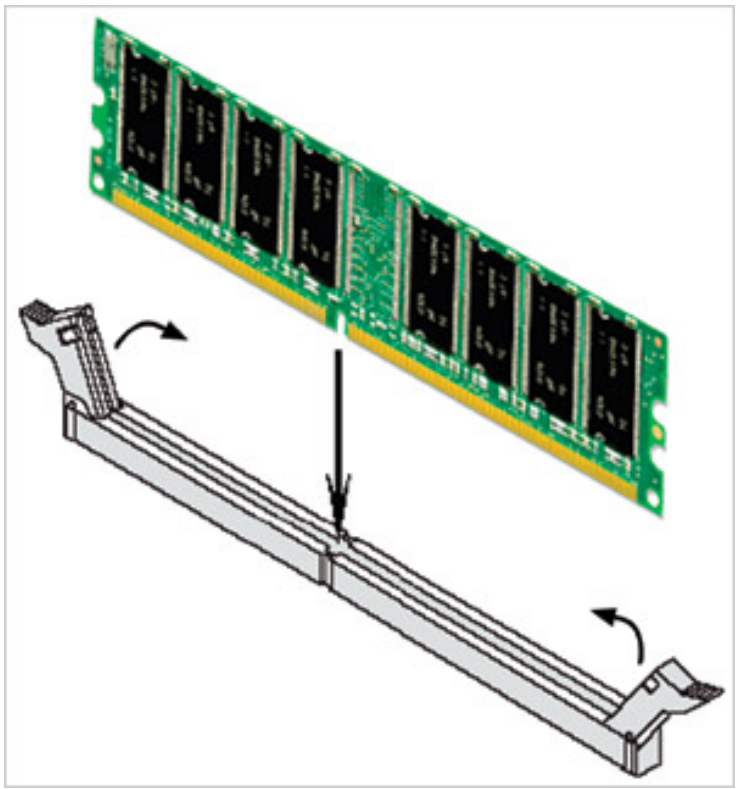




Moduli di memoria

Es.: modulo di memoria di 8 unità (chip)





Schede di memoria

Scheda di memoria:

- circuito stampato che realizza il modulo di memoria
- contiene solitamente alcuni (tipicamente 2^n) chip
- si innesta in appositi slot
- diversi tipi di connessioni (moduli):
  - DIMM Double Inline Memory Module
  - SO-DIMM Small Outline DIMM.

Le schede di memoria sono spesso incompatibili: moduli differenti adottano indirizzamenti e tempi d'accesso diversi.

Memorie permanenti

Le RAM perdono i dati se non sono alimentate.
Memorie permanenti necessarie per:

- il funzionamento di semplici architetture che eseguono sempre uno stesso programma
- memorizzare i programmi di avvio del calcolatore (BIOS).

Tipi di memorie permanenti

- ROM (Read Only Memory), scritte nel momento in cui sono prodotte

Tipi di memorie permanenti

- ROM (Read Only Memory), scritte nel momento in cui sono prodotte
- PROM (Programmable ROM), scrivibili un'unica volta (bit: fusibile non ripristinabile)

Tipi di memorie permanenti

- ROM (Read Only Memory), scritte nel momento in cui sono prodotte
- PROM (Programmable ROM), scrivibili un'unica volta (bit: fusibile non ripristinabile)
- EPROM (Erasable PROM), cancellabili mediante esposizione a raggi ultravioletti (bit: carica elettrica)

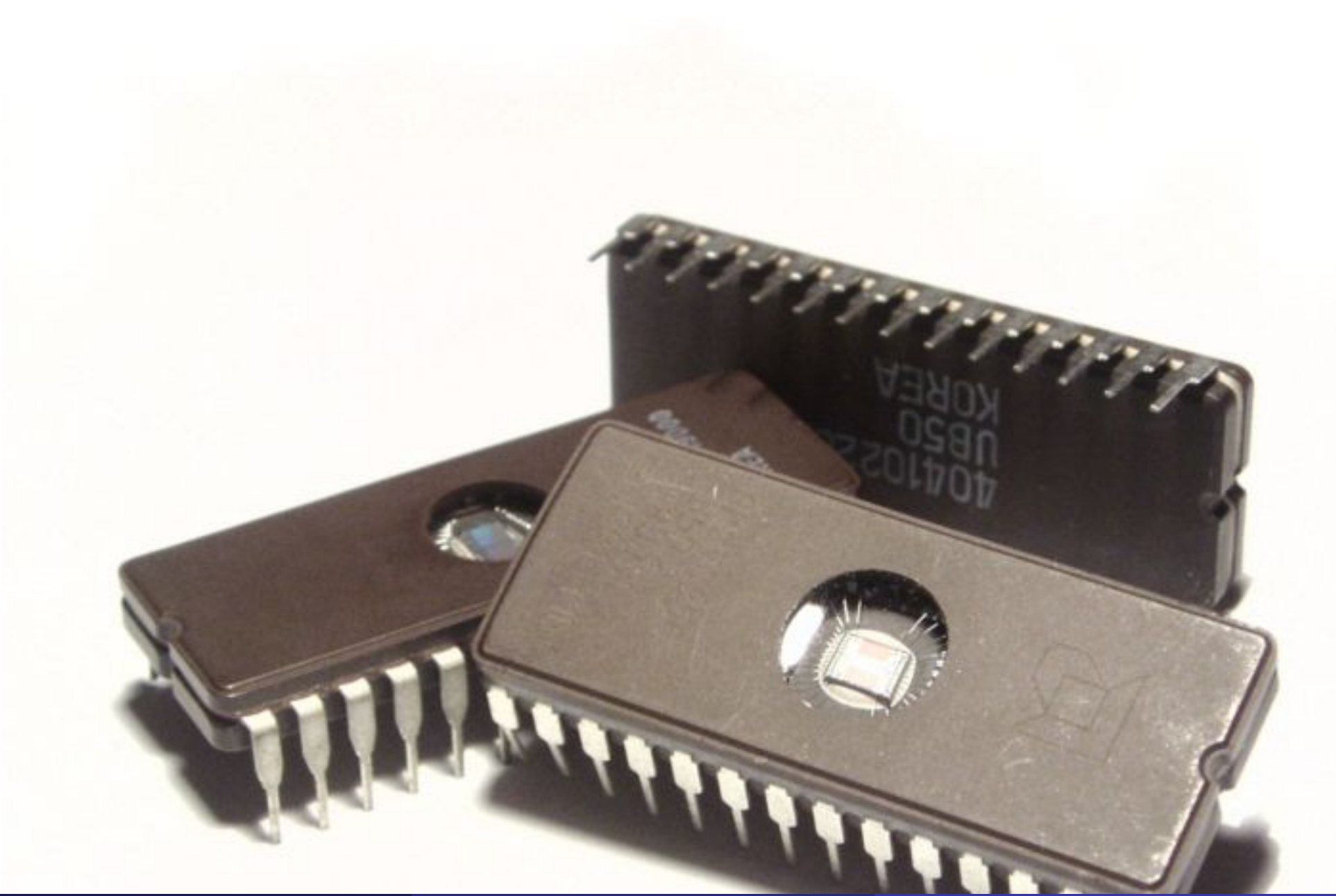
Tipi di memorie permanenti

- ROM (Read Only Memory), scritte nel momento in cui sono prodotte
- PROM (Programmable ROM), scrivibili un'unica volta (bit: fusibile non ripristinabile)
- EPROM (Erasable PROM), cancellabili mediante esposizione a raggi ultravioletti (bit: carica elettrica)
- EEPROM (Electrically EPROM), cancellabili elettricamente (bit: carica elettrica)

Tipi di memorie permanenti

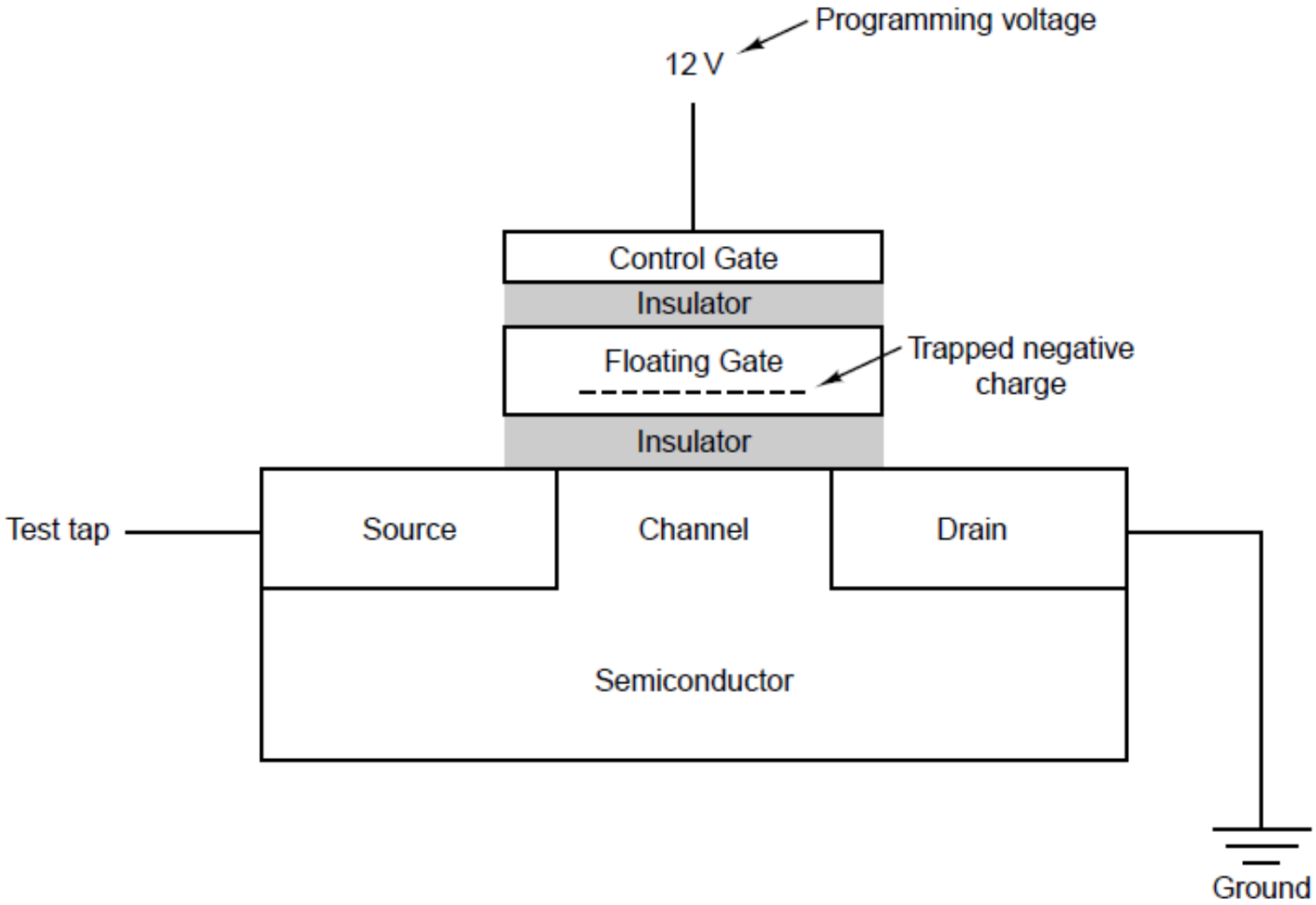
- ROM (Read Only Memory), scritte nel momento in cui sono prodotte
- PROM (Programmable ROM), scrivibili un'unica volta (bit: fusibile non ripristinabile)
- EPROM (Erasable PROM), cancellabili mediante esposizione a raggi ultravioletti (bit: carica elettrica)
- EEPROM (Electrically EPROM), cancellabili elettricamente (bit: carica elettrica)
- Flash: particolari EEPROM cancellabili a banchi (SSD: dischi a stato solido).

Memorie EPROM



Memorie EEPROM, Flash

Floating-gate MOSFET



Classificazione delle memorie

Type	Category	Erasure	Byte alterable	Volatile	Typical use
SRAM	Read/write	Electrical	Yes	Yes	Level 2 cache
DRAM	Read/write	Electrical	Yes	Yes	Main memory (old)
SDRAM	Read/write	Electrical	Yes	Yes	Main memory (new)
ROM	Read-only	Not possible	No	No	Large-volume appliances
PROM	Read-only	Not possible	No	No	Small-volume equipment
EPROM	Read-mostly	UV light	No	No	Device prototyping
EEPROM	Read-mostly	Electrical	Yes	No	Device prototyping
Flash	Read/write	Electrical	No	No	Film for digital camera