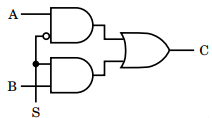
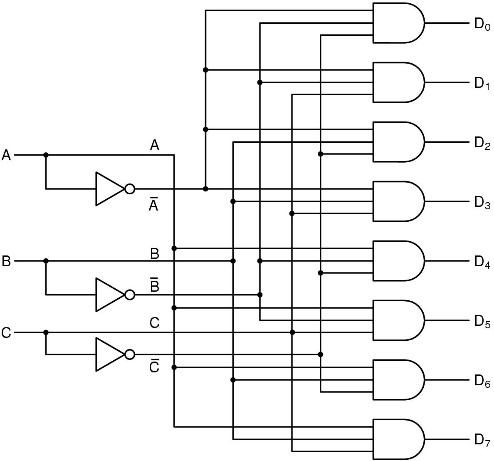
**CIRCUITI LOGICI DI BASE**

* **MULTIPLEXER: N segnali di controllo (S), 2^N segnali di ingresso, 1 uscita segnale. Lo scopo è di selezionare tramite S quale segnale in ingresso mandare in OUTPUT.**
* **DEMULTIPLEXER: N segnali controllo (S), 2^N uscite di segnale, 1 segnale ingresso: Si sceglie in quale delle uscite mandare il segnale di ingresso-**

**Quando S = 1 la AND prende il valore dell’ingresso**

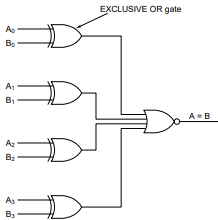
* **DECODER: N ingressi di controllo, 2^N uscite**

**Si attiva l’uscita corrispondente alla codifica in binario degli ingressi di controllo.**

* **ENCODER: N uscite, 2^N ingressi di segnale**

**Codifica in binario il numero dell’ingresso ATTIVO**

* **COMPARATORE:** serve a **verificare** che delle **sequenze di N bit siano uguali**

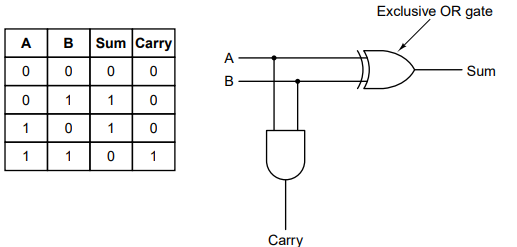
****

La **porta** **finale** è una **NOR**, perciò **tutti** i suoi **ingressi** devono essere a **0** **affinché** **produca** **1.**

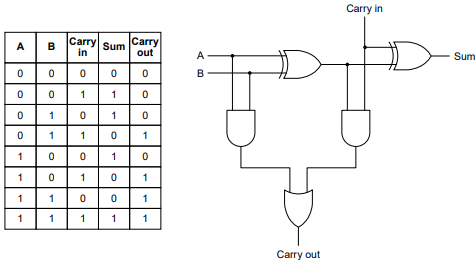
Se **A0** e **B0** sono **uguali** la **XOR** produce **0, altrimenti 1.**

Avendo **1** la **NOR** darà in **uscita 0**, **perciò** le due **sequenze** sono **diverse**.

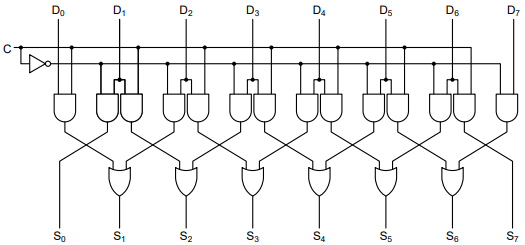
* **MEZZO SOMMATORE: effettua la somma tra due numeri, riportando risultato e eventuale riporto**



* **SOMMATORE COMPLETO: effettua la somma contando anche dell’eventuale riporto precedente**

****

* **SHIFTER: serve a traslare di 1 posizione verso DX (C=1) o SX (C=0)**

****

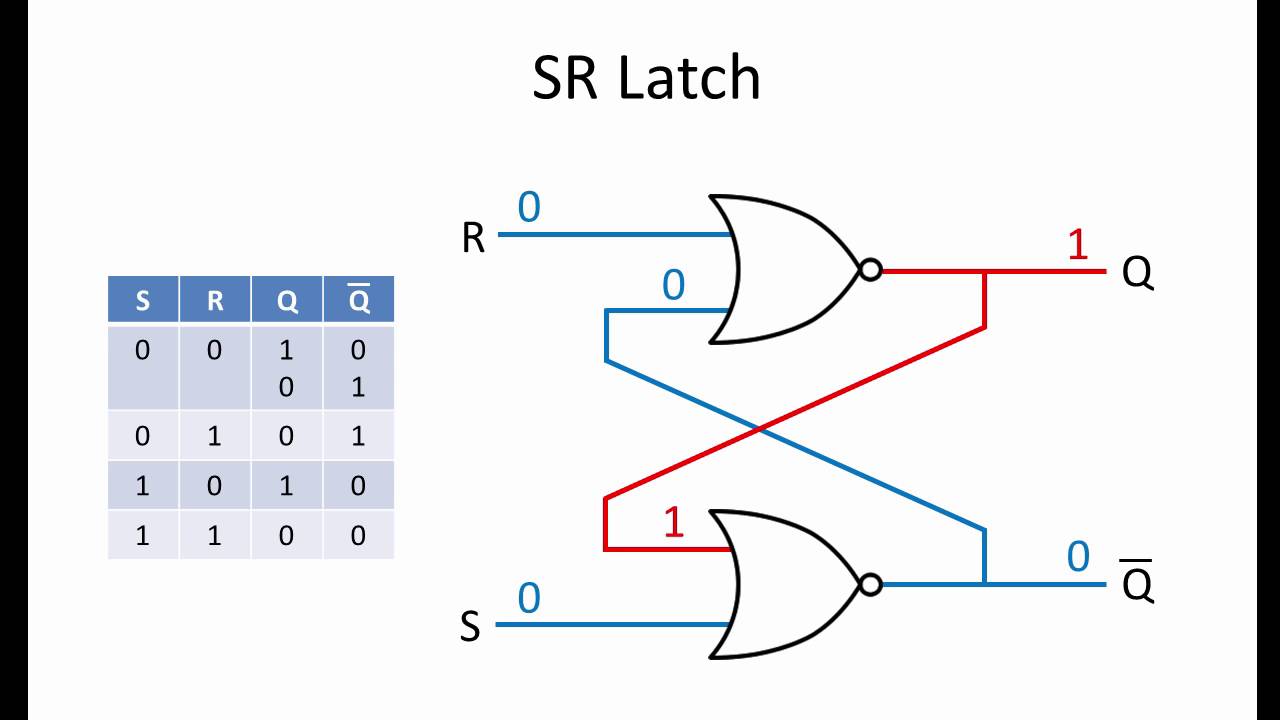
**RITARDI DEI CIRCUITI**

I **circuiti** **logici** **possiedono** un lieve **ritardo** nel **calcolo** dei risultati, nell’ordine di **0,1ns** = 10^(-10)s. Può sembrare un numero insignificante, ma è **molto** **influente**, infatti il **PERIODO** di **CLOCK** della **CPU** si aggira **attorno** a **1ns** (nel caso di F = 1GHz). Con **frequenze** **maggiori** il **periodo** **diminuisce**, **aumentando** **l’effetto** del **ritardo**.

**CIRCUITI COMBINATORI (CON MEMORIA)**

Questi circuiti producono un risultato in base al valore precedentemente calcolato.

**Retroazione:** l’output viene riusato come input



**S=SET**: se **S=1** imposta il valore di **Q = 1**

**R=RESET** se **R=1** imposta il valore di **Q = 0**

**Q** = valore di **output** che andrà successivamente letto

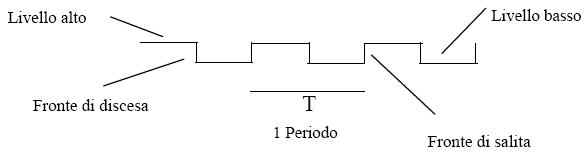
Quando **S=R=0** si è in una condizione di **LATCH**, cioè il valore di **Q dipende dal valore precedente**; infatti, in una NOR il valore 0 determina che nell’uscita ci sia il valore dell’altro ingresso.

Quando S=R=1 si è in una condizione non accettabile; infatti, Q e not(Q) non possono avere lo stesso valore.

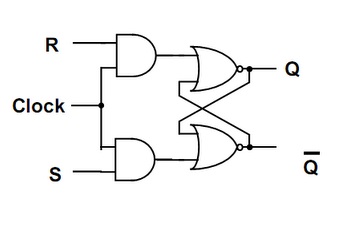
**1 LATCH MEMORIZZA 1 BIT**

**CLOCK**

Rappresenta il segnale di sincronismo, solitamente onda quadra, che fa da riferimento per tutte le operazioni che avvengono all’interno del PC.

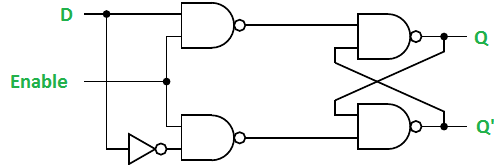
Il clock è la frequenza della CPU, cioè quante volte al secondo è in grado di eseguire le sue operazioni principali (Fetch, Decode e Execute).

**LATCH SR SINCRONO**



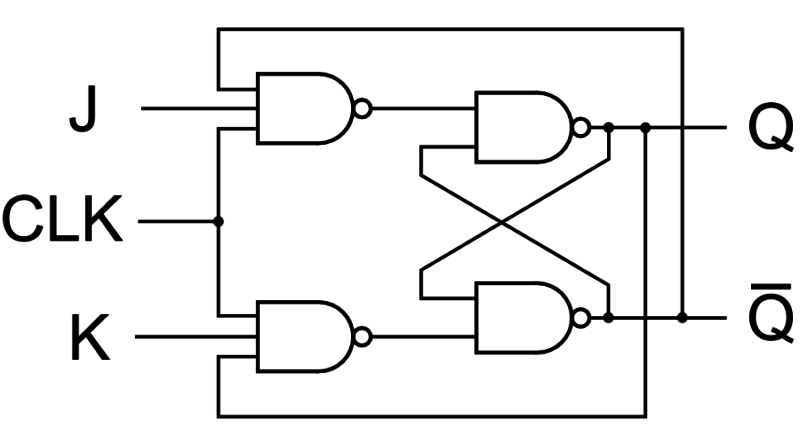
Si comporta come il circuito precedente; l’unica differenza sta nella presenza di un segnale di abilitazione, rappresentato dal CLK. Quando il CLK è alto (=1) abilita la AND di S o R, in base a quale dei 2 viene attivato.

**LATCH D**



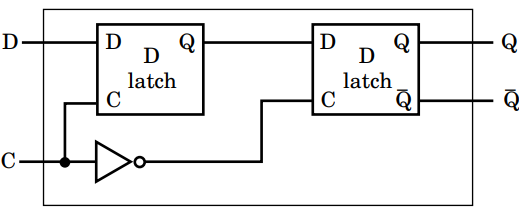
La variabile D sostituisce S e R, e elimina il caso di S=R, in modo che le uniche operazioni che possono avvenire siano il Set e il Reset.

**FLIP – FLOP**



A differenza del Latch, il FLIP-FLOP si attiva quando il CLK è in uno dei due fronti, salita o discesa. La logica di funzionamento resta invariata.

* LATCH : level-triggered (azionato dal valore del livello, alto o basso)
* FLIP-FLOP: edge-triggered (azionato dal fronte di salita o discesa)

**ARCHITETTURA MASTER – SLAVE**

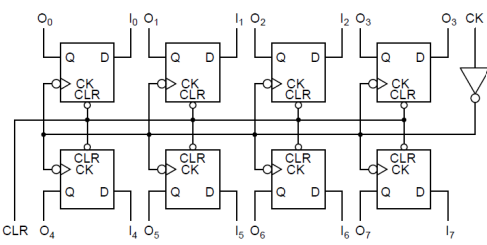
Il latch più a SX rappresenta il MASTER, e il suo valore di uscita aziona lo SLAVE.

Come si nota il CLK viene negato nello SLAVE, in modo che i due circuiti si attivino separatamente.

**REGISTRO**

Insieme di FLIP-FLOP / LATCHES che possono memorizzare sequenze di bit.

Un registro contiene 1 bit



**MACCHINA A STATI FINITI**

In ogni istante la macchina si trova in una determinata situazione, descritta da:

* Configurazione dello **stato**
* **Valore** delle variabili di **input**

Di conseguenza a ogni ciclo di CLOCK può variare:

* **Stato**
* Valore di **output**

Perciò **a ogni ciclo** lo **stato** della macchina **si evolve** in ogni caso, **anche** **se** non **cambia l’output**.

**CIRCUITI INTEGRATI (CHIP)**

Sono delle unità fisicamente separate dal resto e possiedono uno scopo ben preciso.

Immagine che contiene persona, interni

Descrizione generata automaticamenteVengono fatti interagire con altri CHIP o parti dell’HW di un calcolatore.

* Sono delle **piastrine** di **Silicio** su cui sono **incisi transistor, condensatori** e **resistori**.
* I **componenti** elettronici sono **ottenuti** **esponendo** il **Silicio** ad altri agenti, quali **Boro**, **Arsenico** o **Fosforo** (fenomeno della **drogatura**)
* I **collegamenti fisici** sono ottenuti mediante **Rame**

**FOTOLITOGRAFIA:** fenomeno tramite cui il circuito viene coperto da materiali fotosensibili e successivamente esposti a fonti luminose… la parte di circuito illuminata si solidifica.

Successivamente vengono inseriti all’interno di celle frigorifere a bassissime temperature, attorno a -270°C, circa 0 Kelvin, in modo che tutti i componenti si solidificano e si resettano. A quelle temperature la corrente non passa.

**CHIP DI MEMORIA**

I **chip** di **memoria** possono **contenere** **moltissimi** **registri**, ma essi **non** sono **accessibili** **individualmente**. I **registri** sono **raggruppati** in **locazioni** di **memoria**, **accessibili** **tramite** **l’indirizzo** di memoria.

* **Indirizzo di memoria** che identifica una locazione -> **BUS INDIRIZZI**
* **Input:** dato da SCRIVERE in memoria -> **BUS DATI**
* **Output:** dato LETTO dalla memoria -> **BUS DATI**
* **Segnali di controllo** provenienti dal -> **BUS CONTROLLO**
  + **Chip Select:** seleziona quale chip di memoria contattare
  + **Write Enable:** concede il permesso di scrittura in memoria
  + **READ or WRITE:** specifica quale operazione andare ad effettuare.
    - In base alla operazione il **BUS DATI** si comporterà come **input** o **output**