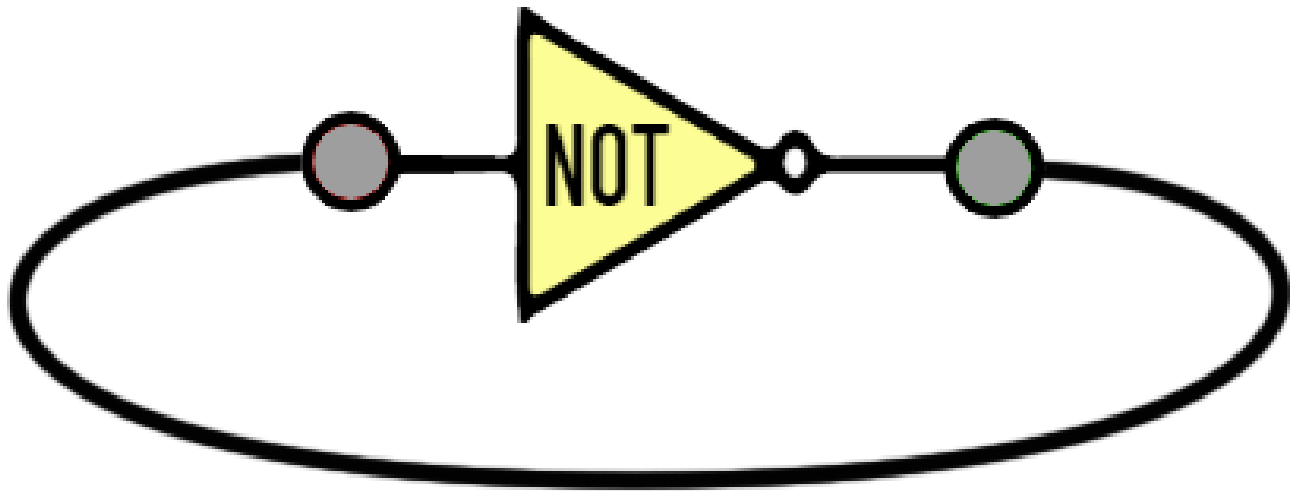




Questa frase è falsa

Cara lettrice, caro lettore,

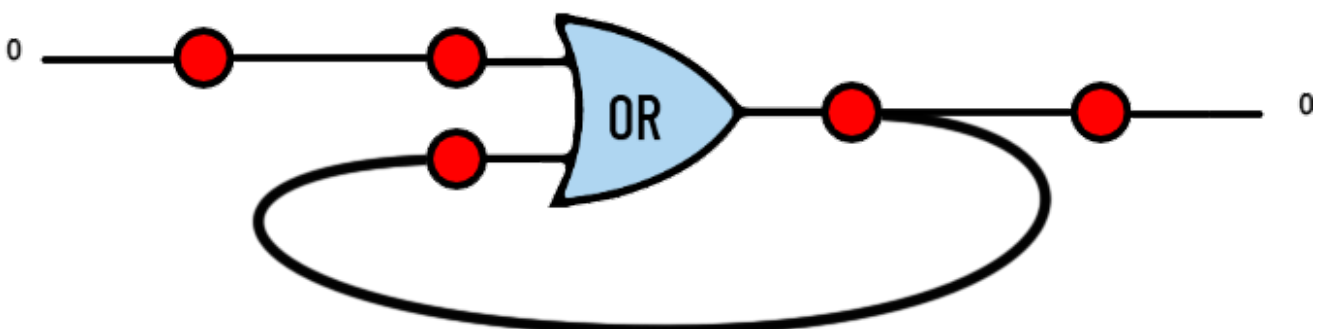
oggi esploreremo un concetto affascinante che collega la matematica, la logica e l'elettronica in modi sorprendenti: i **circuiti autoreferenziali di porte logiche**. Partiamo da un paradosso ben noto, quello racchiuso nell'affermazione "questa frase è falsa". Questa affermazione crea un dilemma logico: se la frase è vera, allora deve essere falsa, ma se è falsa, allora deve essere vera. Sorprendentemente, questo paradosso trova una **rappresentazione pratica** nel mondo delle porte logiche, specificamente attraverso una porta NOT collegata a se stessa. In questo scenario, l'uscita della porta NOT è reinserita come suo ingresso, creando un circuito che non ha uno stato definito, riflettendo il paradosso logico di partenza.



Un circuito creato collegando l'output di una porta NOT al suo input.

Un'interpretazione alternativa, meno logica e più pratica, si ottiene se nel nostro modello il circuito NOT stesso impieghi qualche istante prima di elaborare l'input. In questo caso, supponiamo che inizialmente l'input sia falso. La porta NOT lo elabora e, qualche istante dopo l'output di questa porta diventa vero, e così risulta vero anche l'input (dato che li abbiamo collegati). Ora che l'input è vero, dopo qualche istante la porta NOT elabora l'output falso, e così via, dando origine a un circuito il cui valore logico **continua a oscillare tra vero e falso** nel tempo di aggiornamento della porta NOT.

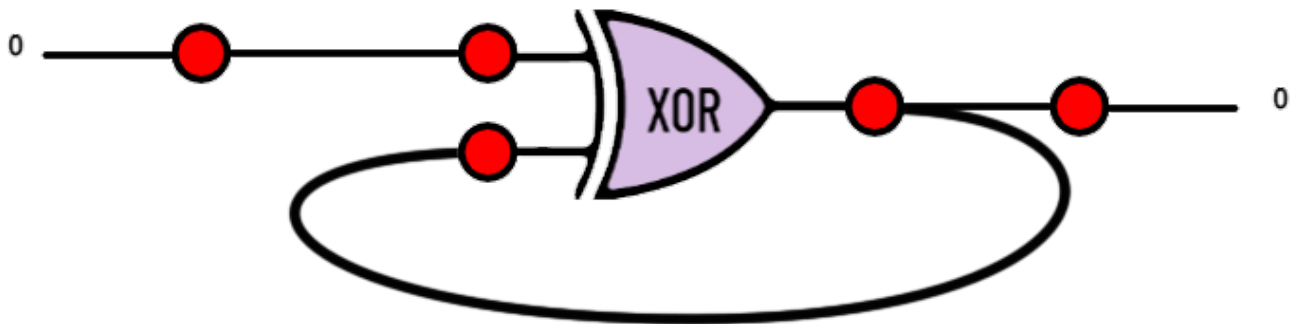
Questo tipo di comportamento richiede uno studio più accurato dal punto di vista matematico, ma è quello che si ottiene quando realizziamo nella pratica delle porte logiche, per esempio utilizzando dei transistor. Possiamo quindi provare a costruire circuiti più complessi che trovano applicazioni nella vita di tutti i giorni. Consideriamo, per esempio, un circuito OR in cui abbiamo un input collegato a un interruttore, e l'output collegato al secondo input rimasto libero. Inizialmente, il circuito è spento: **entrambi gli input valgono falso**, l'output di OR vale falso, e questo è consistente essendo l'output collegato a uno dei due input.



Un circuito autoreferenziale creato utilizzando una porta OR.

Una volta che si attiva l'interruttore, la porta logica elabora il risultato: vero OR falso = vero, e così, il secondo input diventa

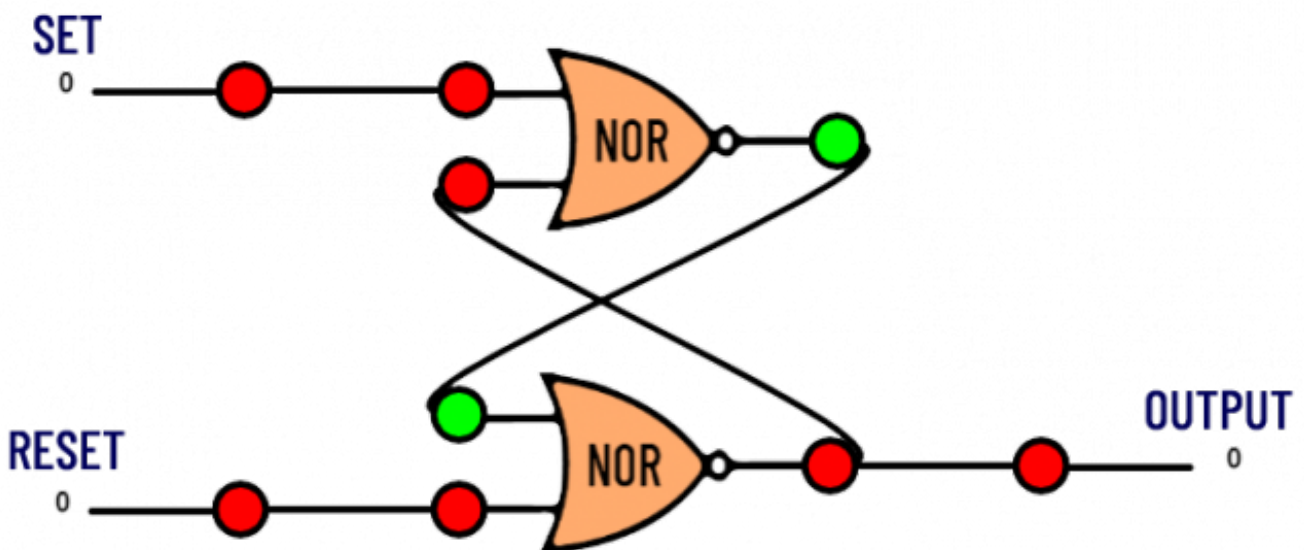
vero. A questo punto non importa più lo stato in cui impostiamo l'interruttore: dato che il secondo input è sempre vero, la porta OR avrà sempre come output vero. Questo meccanismo si può paragonare a un **sistema di allarme** che, una volta attivato, rimane in stato di allerta finché non viene resettato manualmente, per esempio togliendo l'alimentazione.



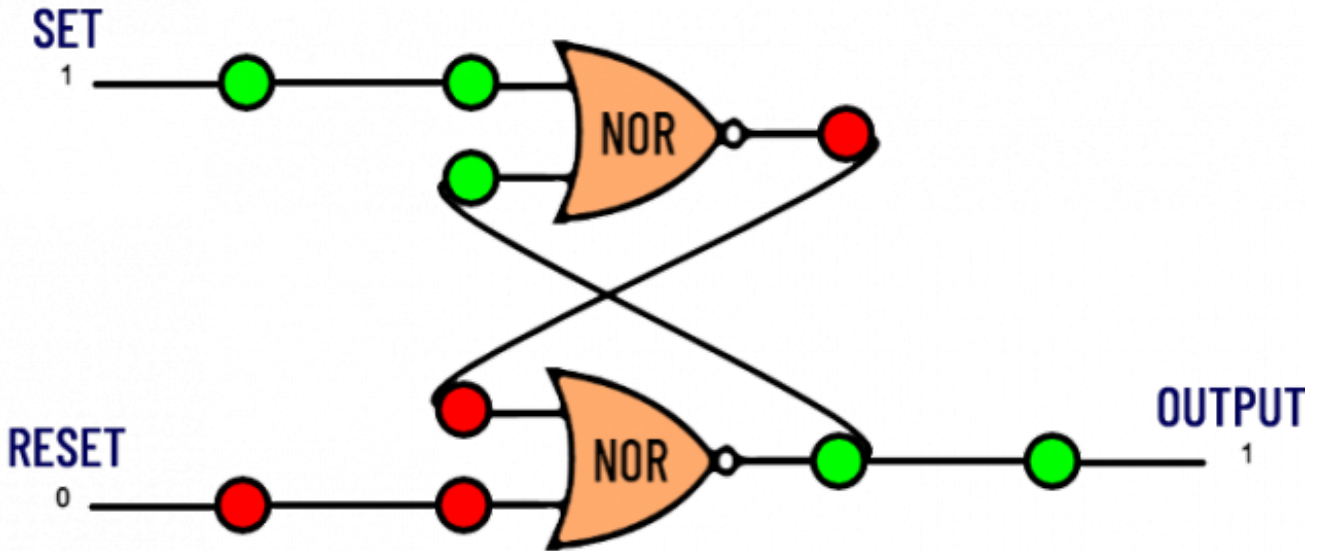
Un circuito autoreferenziale creato utilizzando una porta XOR.

Un altro esempio intrigante è il **circuito XOR autoreferenziale**, dove l'uscita è connessa a uno degli ingressi. In modo simile all'OR, supponiamo che inizialmente tutti gli input valgano falso. In questo caso, quando operiamo sull'input libero la porta logica impiega qualche istante per elaborare $1 \text{ XOR } 0 = 1$. Ma ora il secondo input vale 1, dato che è collegato all'output. Ancora una volta, dopo qualche istante, la porta logica elabora $1 \text{ XOR } 1 = 0$, e così via. L'uscita oscilla indefinitamente tra acceso e spento finché non si riporta l'interruttore a 0, interrompendo questa oscillazione. Questa configurazione può essere paragonata a un sistema di illuminazione a intermittenza, dove l'alternanza continua di stato simula un effetto lampeggiante.

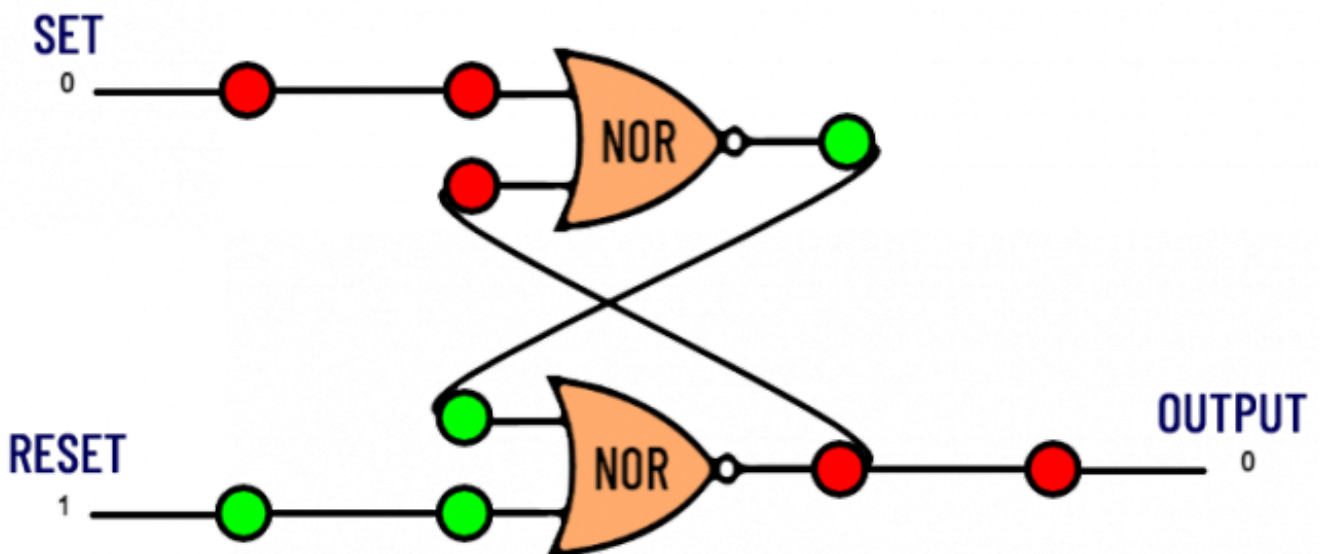
Vi proponiamo ora un circuito più complesso, chiamato **SR-Latch**, costruito utilizzando porte logiche in configurazioni autoreferenziali. La "S" nel nome sta per set e la "R" sta per reset, la funzione dei due pulsanti che controlleranno lo stato del circuito. Un modo per realizzarlo è tramite porte logiche NOR, ma si può ottenere un risultato simile anche con NOR. L'effetto finale, che vi invitiamo a verificare, è il seguente: attivando "set", l'output del circuito si accende, e rimane acceso anche quando "set" viene disattivato. Attivando "reset", l'output del circuito si spegne, e rimane spento anche quando "reset" viene disattivato.



Un esempio di circuito SR-latch costruito utilizzando delle porte NOR. Ora lo si vede nella configurazione di riposo, a output spento.



Un esempio di circuito SR-latch costruito utilizzando delle porte NOR. Ora lo si vede quando viene attivato l'input SET. L'output diventa uguale a 1 e, quando si smette di attivare SET, il circuito rimane in una posizione di riposo con output acceso.



Un esempio di circuito SR-latch costruito utilizzando delle porte NOR. Ora lo si vede quando viene attivato l'input RESET. L'output diventa uguale a 0 e, quando si smette di attivare RESET, il circuito rimane in una posizione di riposo con output spento.

Un circuito simile è quello per esempio dei circuiti di controllo di alcune macchine utensili, che utilizzano interruttori a pulsante per spegnere o accendere i dispositivi. Inoltre, questi circuiti funzionano come **memorie elementari**, capaci di impostare e poi mantenere lo stato di output fino a che non vengono cambiati da un input esterno. In questo contesto, questo circuito è una memoria da 1 bit. Questo principio trova applicazioni pratiche in una varietà di dispositivi, dalla conservazione dei dati nei computer fino ai sistemi di bloccaggio elettronico nelle porte di sicurezza, dimostrando come concetti apparentemente astratti possano avere impatti concreti e tangibili nel mondo reale.

Attraverso questi esempi, vediamo come le porte logiche, oltre a essere fondamentali per la costruzione di circuiti elettronici, ci offrono anche un mezzo per esplorare e comprendere concetti logici e matematici complessi, dimostrando ancora una volta l'**intreccio profondo tra teoria e pratica** e quanto la matematica abbia un ruolo fondamentale nella tecnologia che usiamo ogni giorno. Questi esempi provengono dalla mostra "Circuiti invisibili" di Curvilinea, che esplora le applicazioni delle porte logiche da varie angolazioni, inclusi linguaggio e matematica, ma facendo anche ponti verso il mondo dell'informatica.