

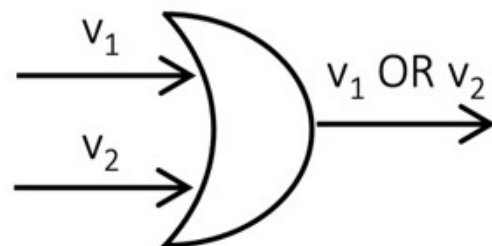
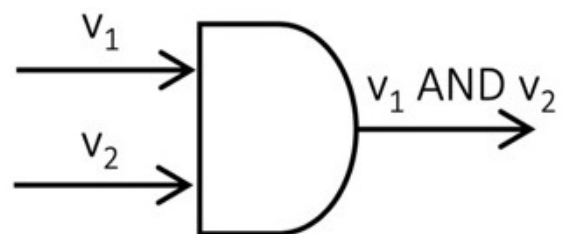
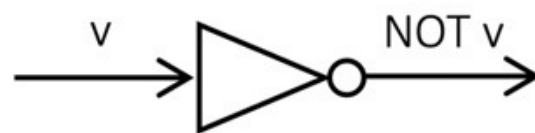
Principi e Metodi del Digitale

Lezione 3

Mario Verdicchio

Università degli Studi di Bergamo

A.A. 2026-2027



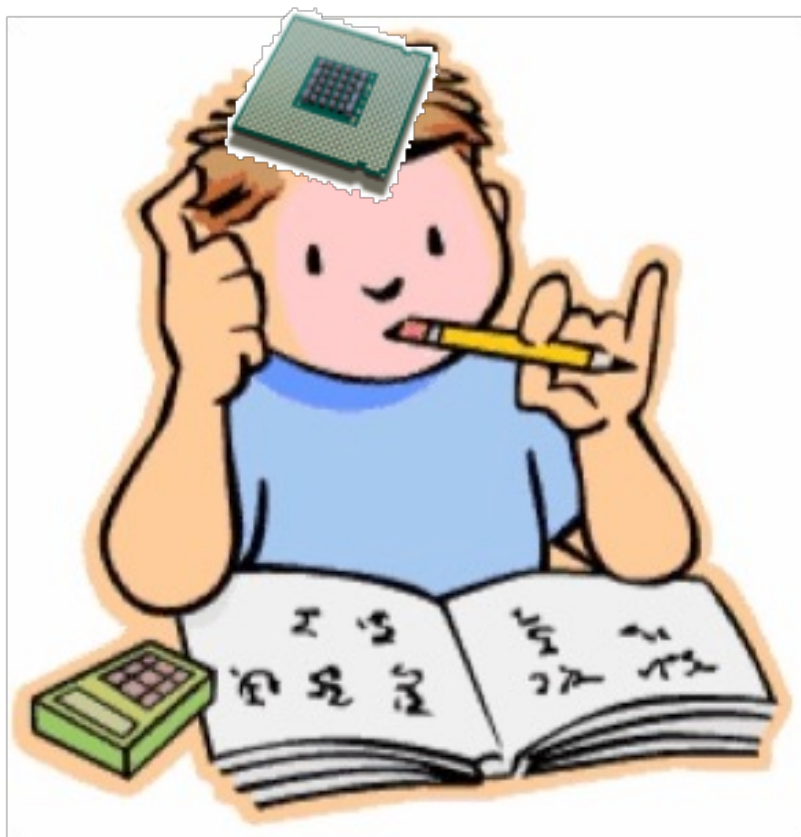


Andiamo a vedere dentro.

Hardware



Noto in italiano come “processore”, la CPU (Central Processing Unit) è la parte del computer che esegue le operazioni.



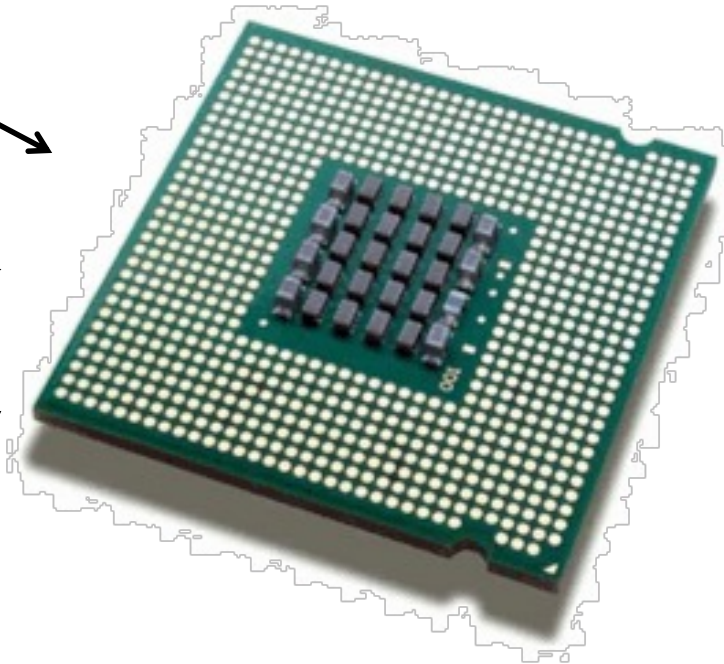
CPU

La CPU è fatta di due parti:
la ALU (Arithmetic-Logical Unit) esegue le operazioni
aritmetiche e logiche.

3

+

5

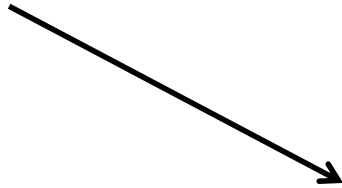


8

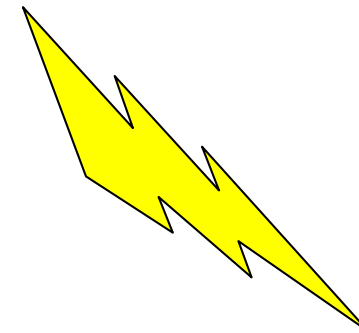
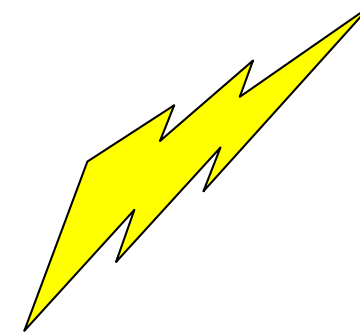
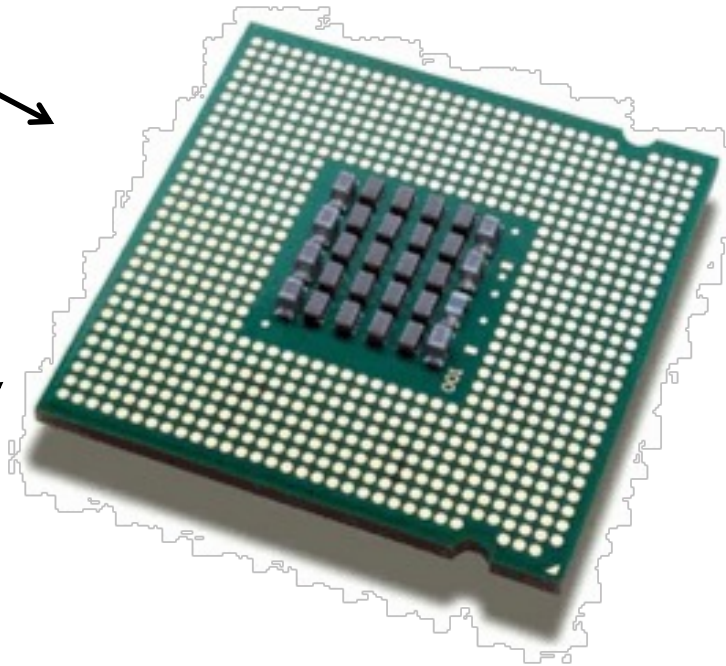
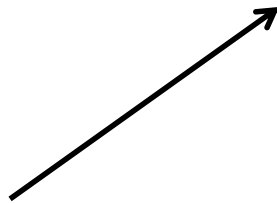
ALU

La CU (Control Unit) è la parte che controlla il trasferimento di dati da una parte all'altra del computer.

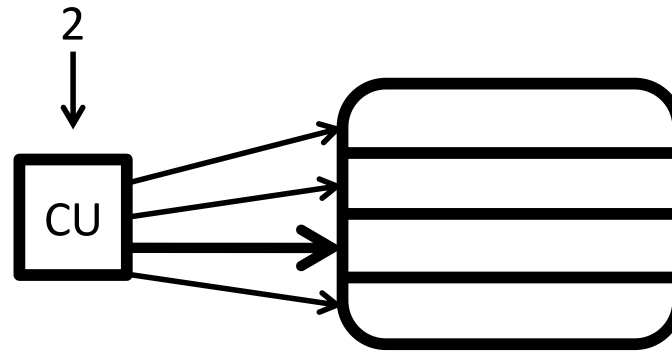
print



“ciao”



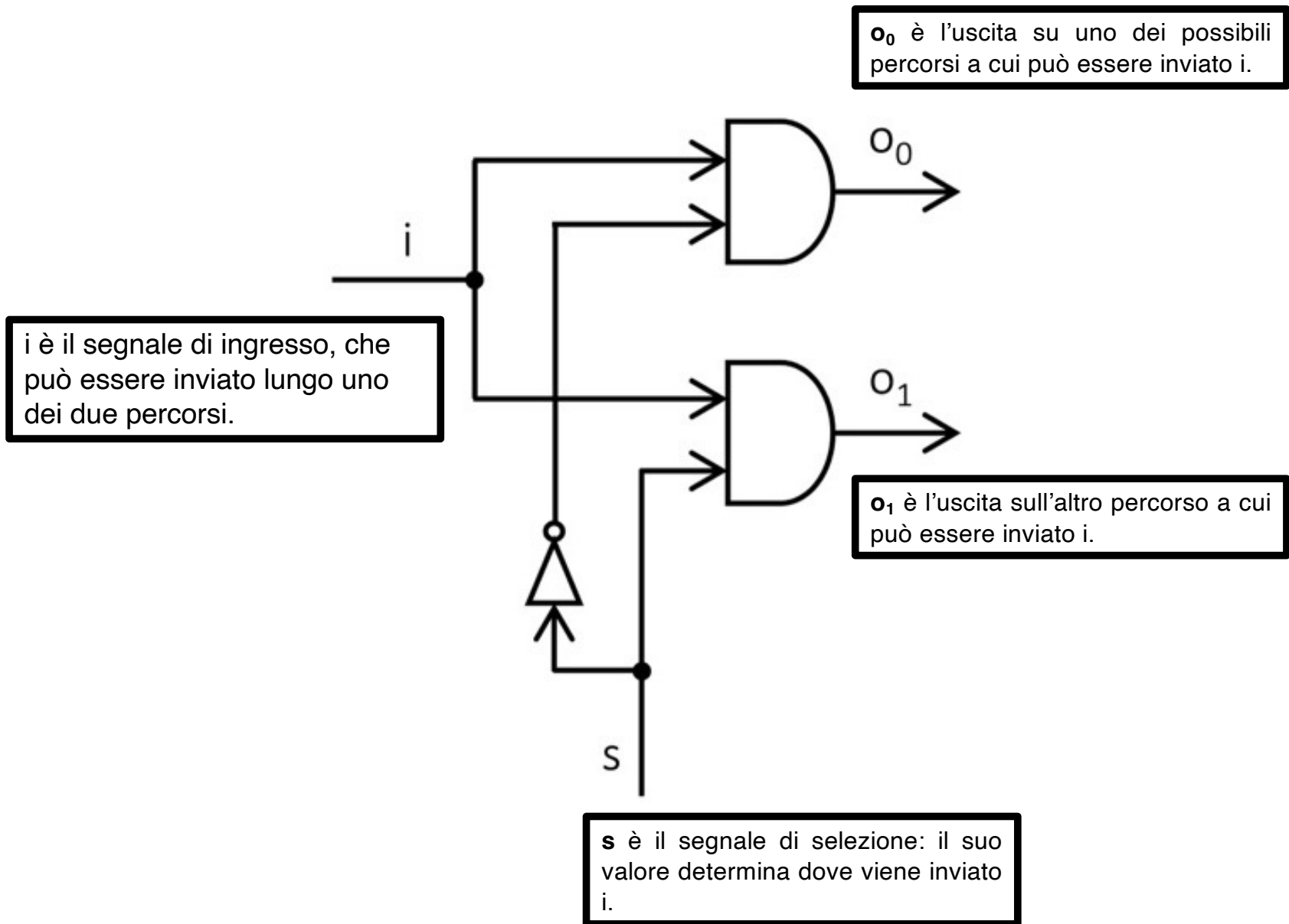
CU



Vediamo ora come queste porte possono essere combinate per effettuare la selezione di un percorso specifico.

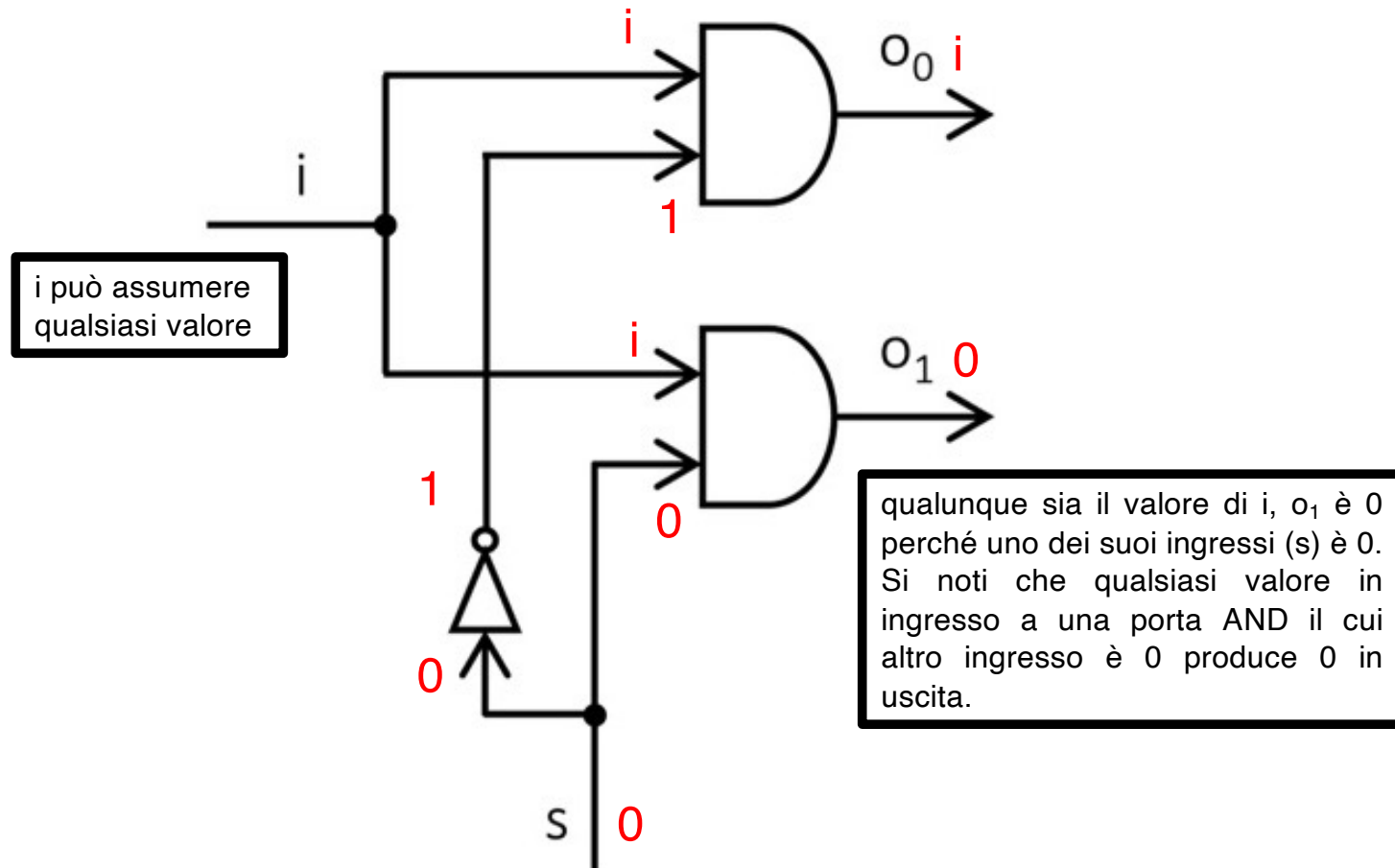
Concentriamoci su un caso più semplice di quello raffigurato nella figura, in cui la CU sceglie tra 4 percorsi diversi.

Vediamo di seguito come costruire un circuito che consenta di scegliere tra 2 percorsi.



se s è 0...

o_0 è uguale a i perché l'altro ingresso della porta AND è 1. Si noti che $i \text{ AND } 1$ equivale a i (che i sia 0 oppure 1 non fa differenza).

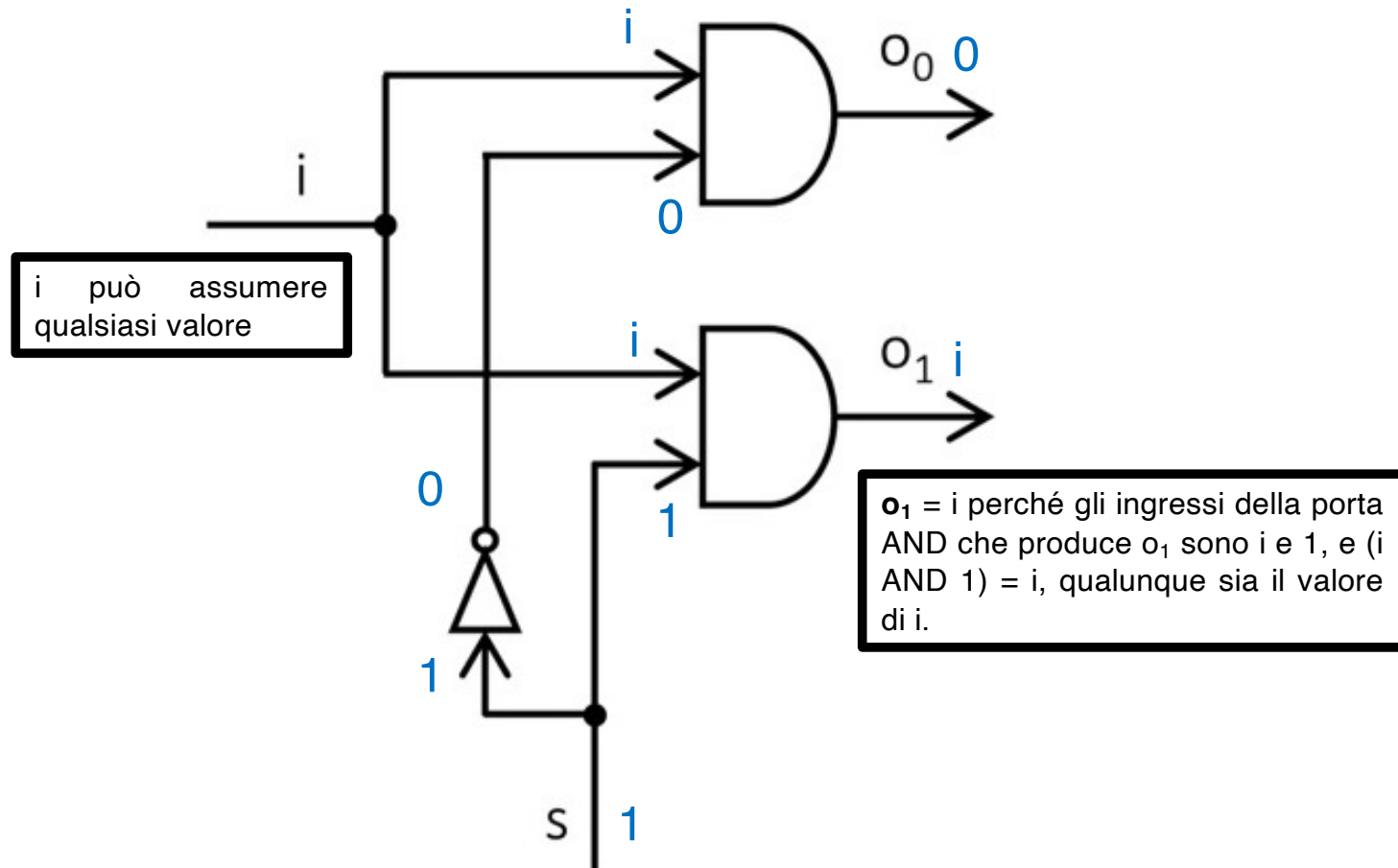


...allora $o_0 = i$.

In altre parole, i viene inviato al primo percorso.

Se s è 1...

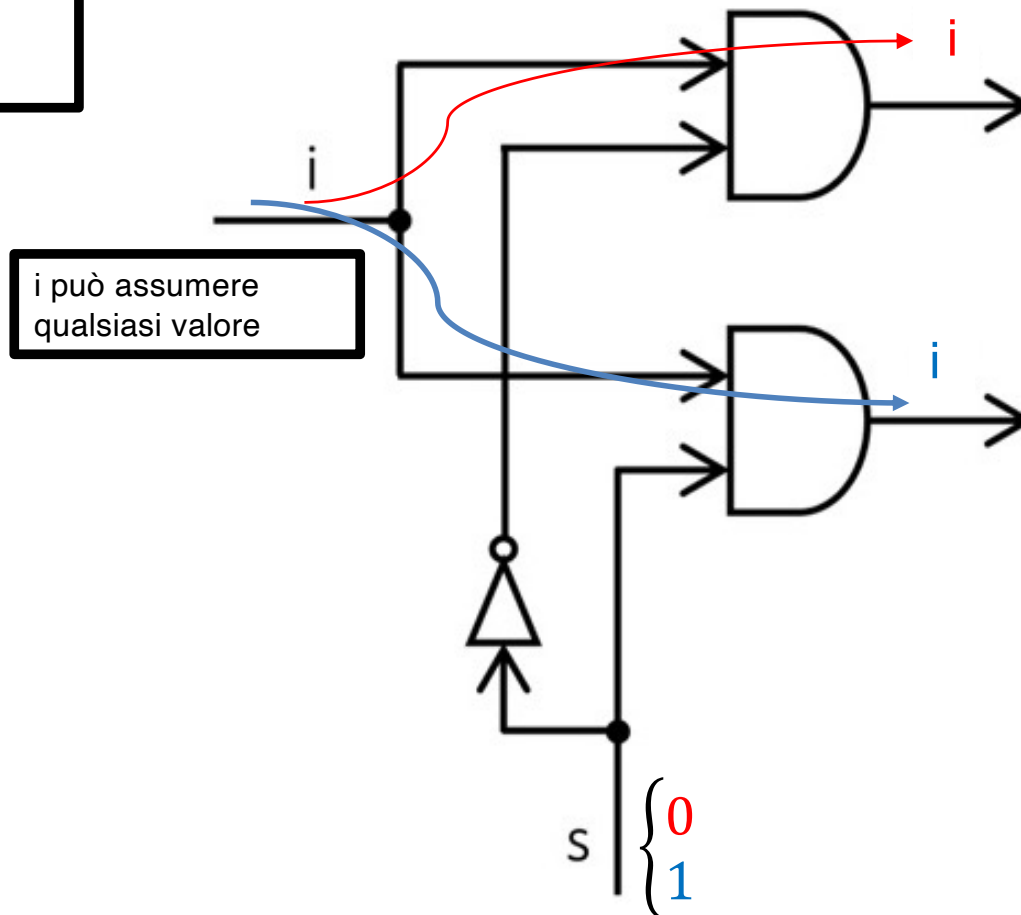
o_0 è 0 qualunque sia il valore di i , perché l'altro ingresso della porta AND è 0.



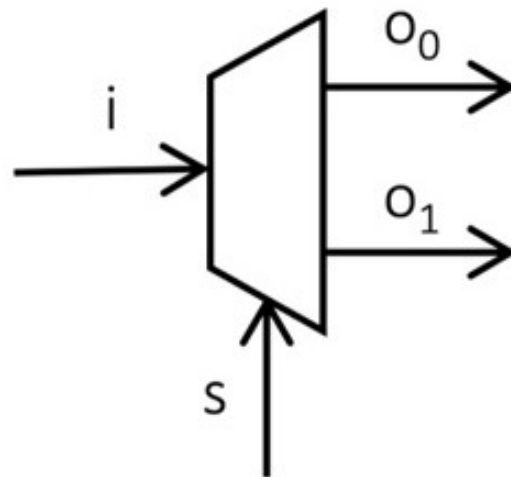
...allora $o_1 = i$.
Ossia, i viene inviato al secondo percorso.

Questo
significa

...

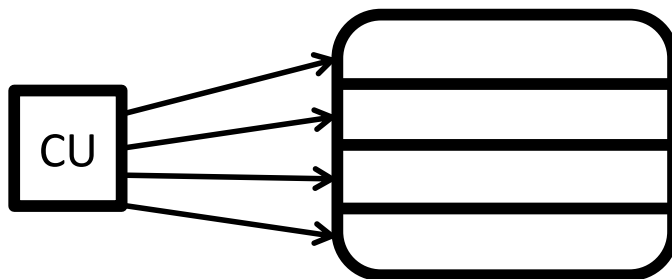


...che è come se s selezionasse
dove viene inviato i .



Questo è lo stile di progettazione dei circuiti usato per rappresentare un sistema di questo tipo, chiamato “demultiplexer” (o “DEMUX”).

Poiché la scelta avviene tra 2 percorsi, questo è chiamato DEMUX 1:2. In un DEMUX 1:2, il segnale di selezione è di un solo bit.



Nel nostro esempio originale, la scelta avviene tra 4 percorsi; abbiamo quindi bisogno di un DEMUX 1:4, con 2 bit per il segnale di selezione (s_0 e s_1), che consentono di scegliere tra 4 percorsi: O_{00} , O_{01} , O_{10} e O_{11} . L'elettronica è un po' più complessa, ma i principi sono gli stessi che guidano la progettazione del DEMUX 1:2.

Il punto della situazione

- Con tutte queste considerazioni sulla logica e sulla costruzione di circuiti elettronici che realizzano operatori logici come NOT, AND e OR, ci siamo concentrati sulla L di ALU
- Inoltre, abbiamo combinato porte NOT e AND per creare un dispositivo DEMUX, che permette di selezionare un percorso tra molti, attraverso il quale la CU può inviare i segnali dove servono
- Ma che ne è della A di ALU?
- In fondo, i computer non sono costruiti per calcolare, cioè per eseguire operazioni aritmetiche?

Ritorno alle basi dell'aritmetica

- I numeri possono essere usati come input di operazioni aritmetiche indipendentemente dal sistema numerico in cui sono espressi
- Possiamo calcolare $3 + 4$ e ottenere 7 (base 10)
- Allo stesso modo, possiamo calcolare $011 + 100$ e ottenere 111 (base 2)

Aritmetica semplice in binario

- Dati due bit in ingresso, le regole per sommarli sono molto semplici:
 - $0 + 0 = 0$
 - $0 + 1 = 1$
 - $1 + 0 = 1$
 - $1 + 1 = 10$, cioè 0 con riporto 1
- Poiché $+$ si applica a 2 bit e produce 1 bit in uscita (eventualmente con un riporto), proprio come AND e OR, possiamo immaginare di costruire un sistema di porte logiche che manipoli i bit in modo coincidente con l'operazione $+$

L'addizione nell'elettronica

- Per il momento ignoriamo il riporto nell'ultimo caso e concentriamoci su ciò che accade ai bit:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0$$

L'uscita di + è quasi identica all'uscita di OR

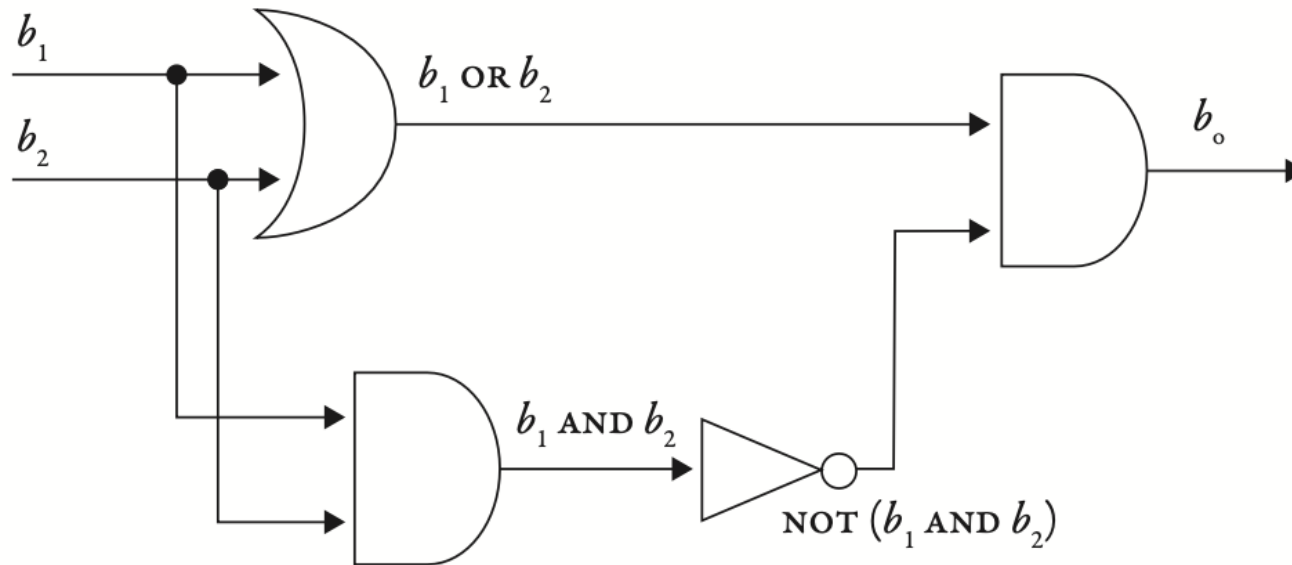
input ₁	input ₂	output
0	0	0
0	1	1
1	0	1
1	1	1

Tranne nell'ultimo caso, in cui entrambi i bit in ingresso sono 1 e l'uscita è 0 anziché 1.

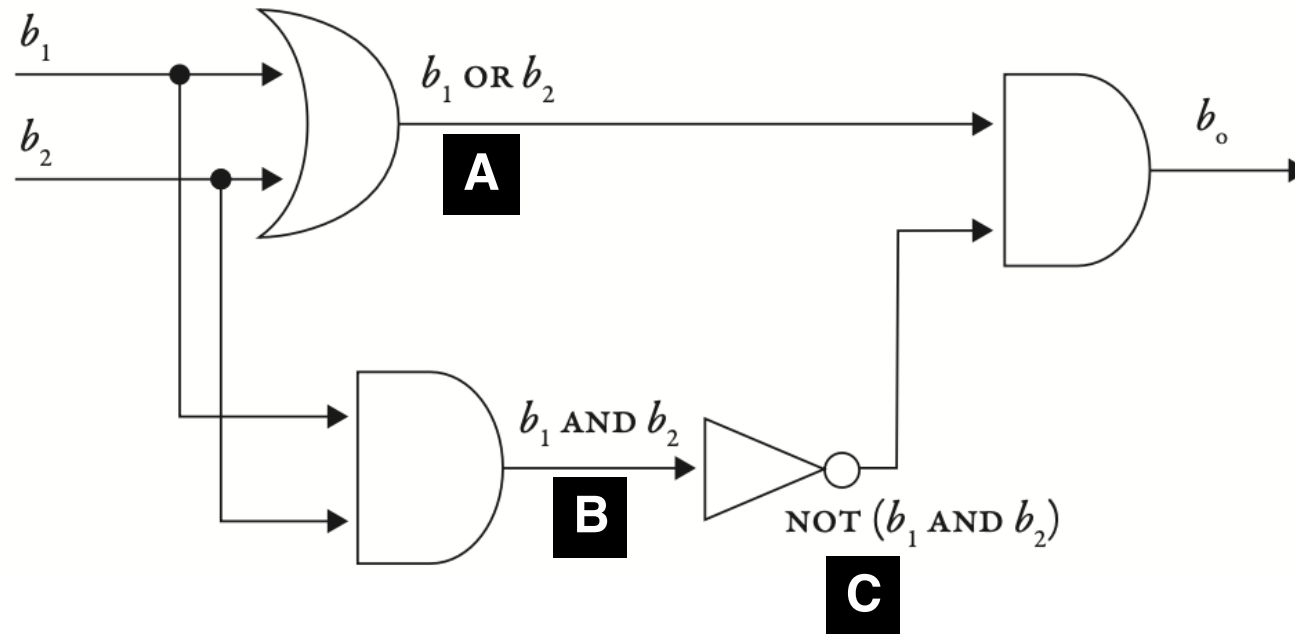
Quindi, dati due bit in ingresso, b1 e b2, b1 + b2 si comporta come b1 OR b2, tranne quando entrambi valgono 1.

Dati due bit in ingresso, b_1 e b_2 ,
 $b_1 + b_2$ si comporta come b_1 OR b_2 ,
tranne quando entrambi valgono 1.

Per progettare un sistema che realizzi elettronicamente questo comportamento, possiamo combinare le uscite di:
 b_1 OR b_2
AND
NOT (b_1 AND b_2),
come mostrato nel sistema in figura (tratto da “Che cos’è un computer” di Mario Verdicchio, Carocci, 2023).



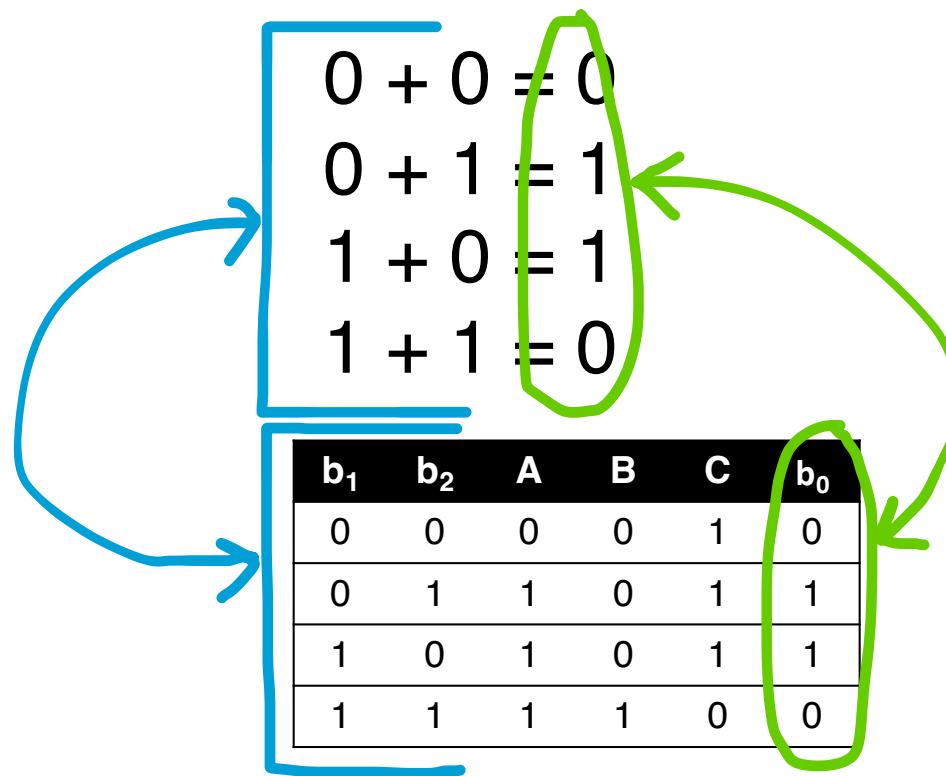
Vediamo nella slide seguente come funziona questo sistema.



b_1	b_2	A	B	C	b_0
0	0	0	0	1	0
0	1	1	0	1	1
1	0	1	0	1	1
1	1	1	1	0	0

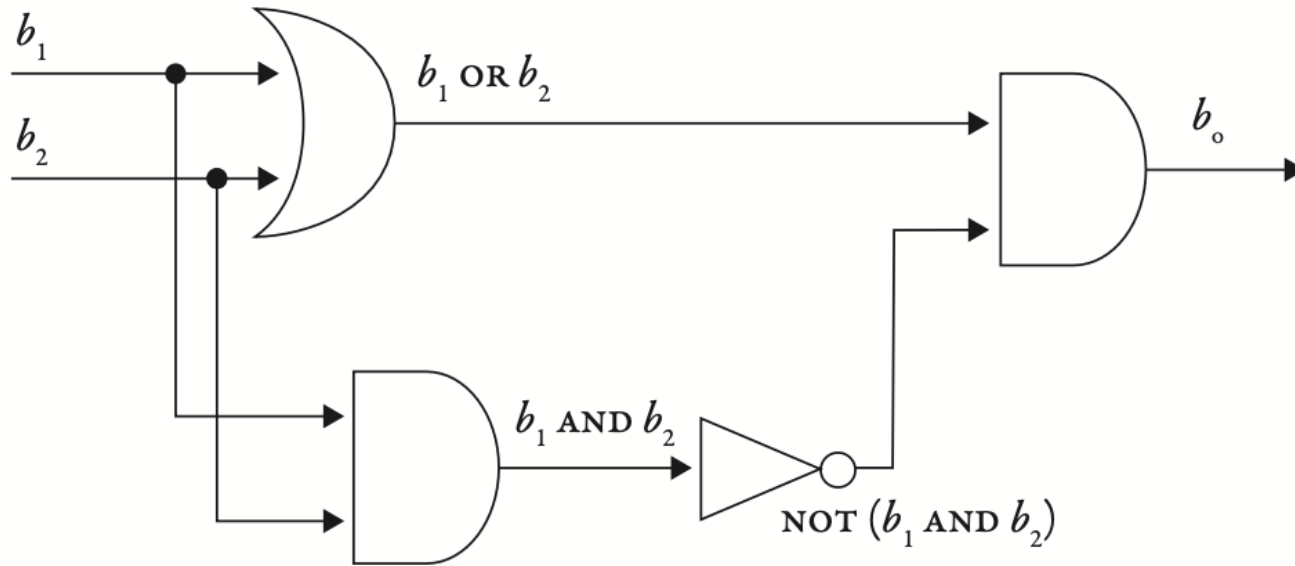
Il sistema usa gli ingressi b_1 e b_2 per calcolare, da una parte, $(b_1 \text{ OR } b_2)$ (che abbiamo chiamato **A**) e, dall'altra, la negazione di $(b_1 \text{ AND } b_2)$. La negazione di $(b_1 \text{ AND } b_2)$ viene calcolata mediante una porta AND (ottenendo **B**) e poi una porta NOT (ottenendo **C**).

Il calcolo finale, che produce l'uscita, combina **A** e **C** con una AND: ciò equivale a porre il risultato finale a 1 in tutti i casi in cui $(b_1 \text{ OR } b_2)$ è 1, tranne quando $(b_1 \text{ AND } b_2)$ è 1. In quel caso **C** è 0 e, attraverso l'ultima porta AND, porta il risultato finale a 0.



Quel circuito elettronico, composto da una porta OR, una porta NOT e due porte AND combinate come mostrato in precedenza, produce la stessa uscita dell'addizione applicata a due ingressi da un solo bit. Può quindi funzionare come SOMMATORE (ADDER).

Dal punto di vista fisico, è semplicemente un circuito che manipola tensioni elettriche; tenendo però presente la codifica fondamentale, possiamo interpretare il suo funzionamento (elettrico) come un'operazione (aritmetica) di addizione.



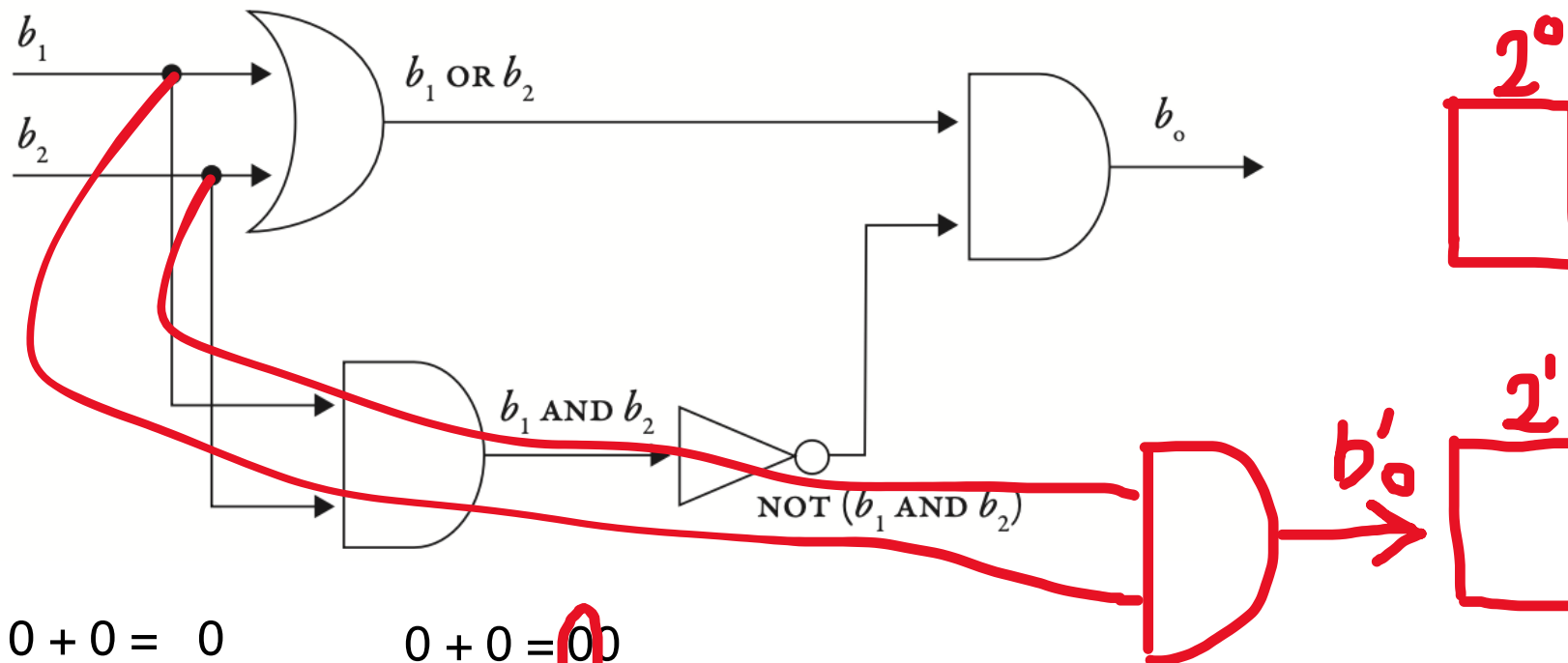
$0 + 0 = 0$
 $0 + 1 = 1$
 $1 + 0 = 1$
 $1 + 1 = 10$

$0 + 0 = 00$
 $0 + 1 = 01$
 $1 + 0 = 01$
 $1 + 1 = 10$

Non dobbiamo dimenticare il riporto.

Il riporto è un altro bit che fa parte del risultato finale. Nei primi 3 casi vale 0 e nell'ultimo vale 1. I bit di riporto corrispondono esattamente all'uscita di $(b_1 \text{ AND } b_2)$.

Per costruire il circuito che calcola il riporto, basta quindi far passare i due ingressi attraverso una porta AND.



$0 + 0 = 0$
 $0 + 1 = 1$
 $1 + 0 = 1$
 $1 + 1 = 10$

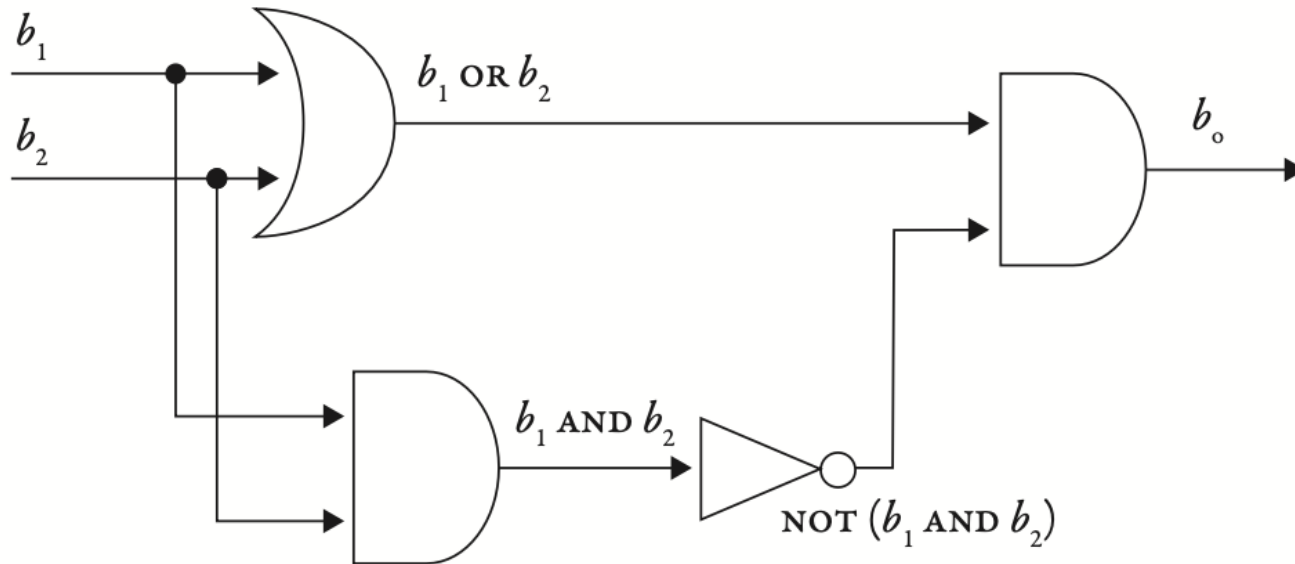
$0 + 0 = 00$
 $0 + 1 = 01$
 $1 + 0 = 01$
 $1 + 1 = 10$

Non dobbiamo dimenticare il riporto.

Il riporto è un altro bit che fa parte del risultato finale. Nei primi 3 casi vale 0 e nell'ultimo vale 1. I bit di riporto corrispondono esattamente all'uscita di $(b_1 \text{ AND } b_2)$.

Per costruire il circuito che calcola il riporto, basta quindi far passare i due ingressi attraverso una porta AND.

Il bit in uscita dal primo sistema (in nero, stampato) viene memorizzato in uno spazio di memoria da 1 bit che rappresenta quante unità (2^0) sono presenti nel risultato finale. Il bit in uscita dal secondo sistema (in rosso, scritto a mano) viene memorizzato in un'altra unità di memoria, che indica quanti due (2^1) sono presenti nel risultato finale.



Dato che abbiamo costruito un sistema aritmetico combinando un sistema logico di AND, OR e NOT, significa forse che l'aritmetica si basa sulla logica?

Assolutamente no. Nel mondo reale, aritmetica e logica hanno origini ben distinte. Anzi, poiché contare con le dita ha preceduto la formalizzazione del ragionamento, semmai l'aritmetica potrebbe essere considerata alla base di un modo di pensare che ha portato l'umanità a concepire la logica in un secondo momento; ma queste sono questioni per storici e filosofi della scienza.

Il punto è semplicemente che, nel mondo materiale dei circuiti elettronici costruiti mediante transistor, gli operatori logici sono molto più semplici da implementare delle operazioni aritmetiche. Lo si vede considerando che basta un solo transistor per costruire una porta NOT, mentre un SOMMATORE ha una struttura molto più complessa.

L'informatica ci mette a disposizione dispositivi estremamente utili, ma questi dispositivi sono il risultato sia di scoperte scientifiche (per esempio i semiconduttori) sia di accordi sociopolitici (per esempio codifiche come RGB e standard come JPG). Non vanno interpretati come indicazioni di principi fondamentali sul funzionamento dell'universo o del nostro pensiero e ragionamento.

Eppure, in alcuni sottocampi di questa disciplina, gli esperti se ne dimenticano. Voi non fatelo.