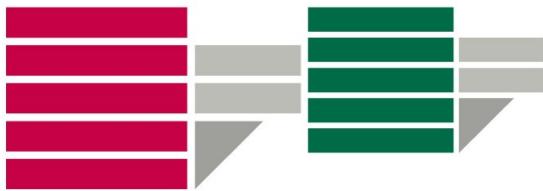


UNIVERSITÀ DELLA CALABRIA



DIMES - Dipartimento di INGEGNERIA INFORMATICA
MODELLISTICA, ELETTRONICA E SISTEMISTICA

Anno Accademico 2015-2016

Corso di Laurea Triennale in Ingegneria Informatica

Secondo progetto di Elettronica Digitale

“Circuito del Full Adder realizzato con gli interruttori”

Professoressa S.Perri

Ivonne Rizzuto
matricola 167058

Descrizione

Il progetto consiste nella realizzazione di un circuito sommatore costituito da un Full Adder, utilizzando gli interruttori.

Da questo bisognerà ricavare un circuito ad esso equivalente da definire in termini di resistenze, denominate R_{on} e R_{off} , per 4 combinazioni di bit in ingresso.

Full Adder

Questo componente circuitale è realizzato tramite due porte *xor* ed un *multiplexer*. Riceve in ingresso tre bit, quali i due operandi a e b ed il selettore cin , restituendo, in uscita il risultato della somma s ed il riporto finale $cout$.

La funzione logica utilizzata per la sua implementazione è la seguente:

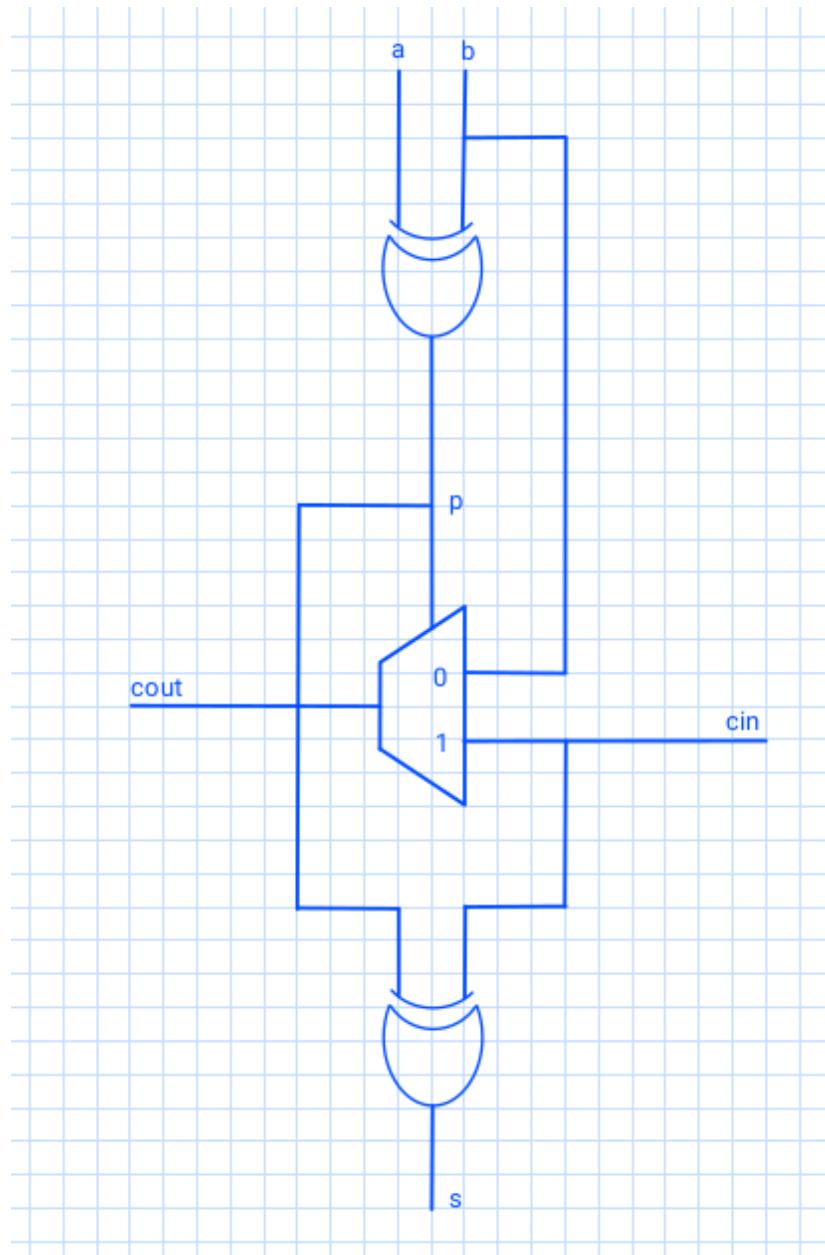
$$s = p \text{ xor } cin$$

dove p è il segnale ausiliario di propagate che descritto dall'espressione sottostante:

$$p = a \text{ xor } b$$

Essendo lo *xor* e il *mux* componenti logici complessi, si lavorerà sulle rispettive espressioni booleane ai fini di gestirle con più semplicità.

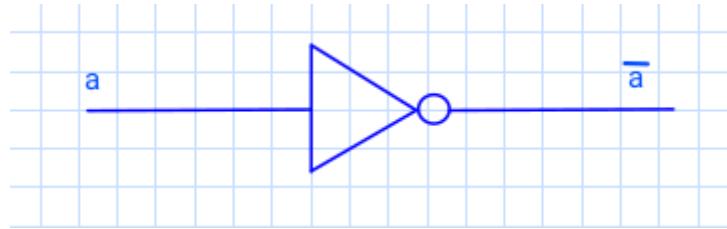
Segue il disegno del modello di Full Adder utilizzato:



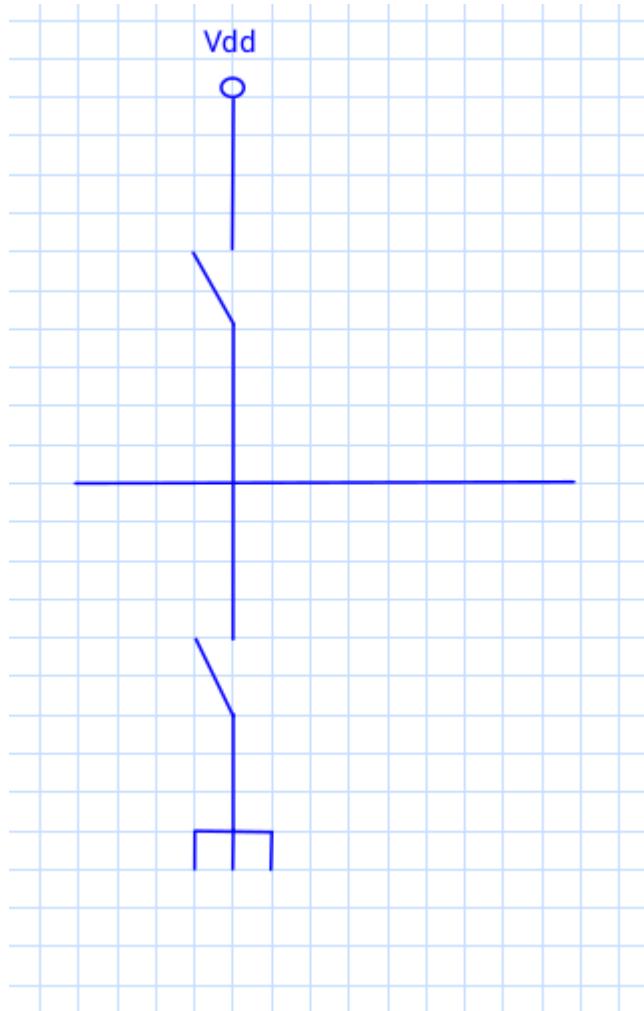
Si analizzano, di seguito, le componenti circuitali che lo costituiscono.

Inverter

Questo componente, dato un segnale a , svolge la funzione logica Not, per la quale si ottiene $a = \bar{a}$.

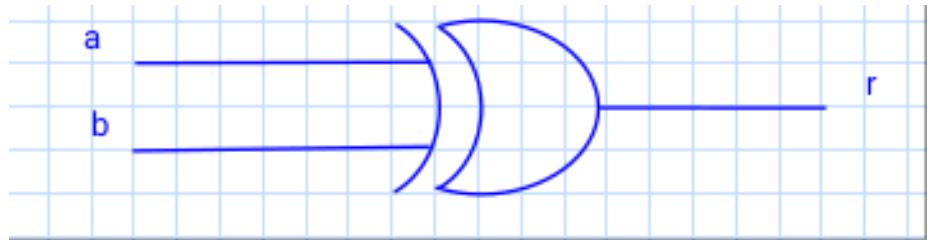


Tramite gli interruttori, lo si rappresenta così:



Porta XOR

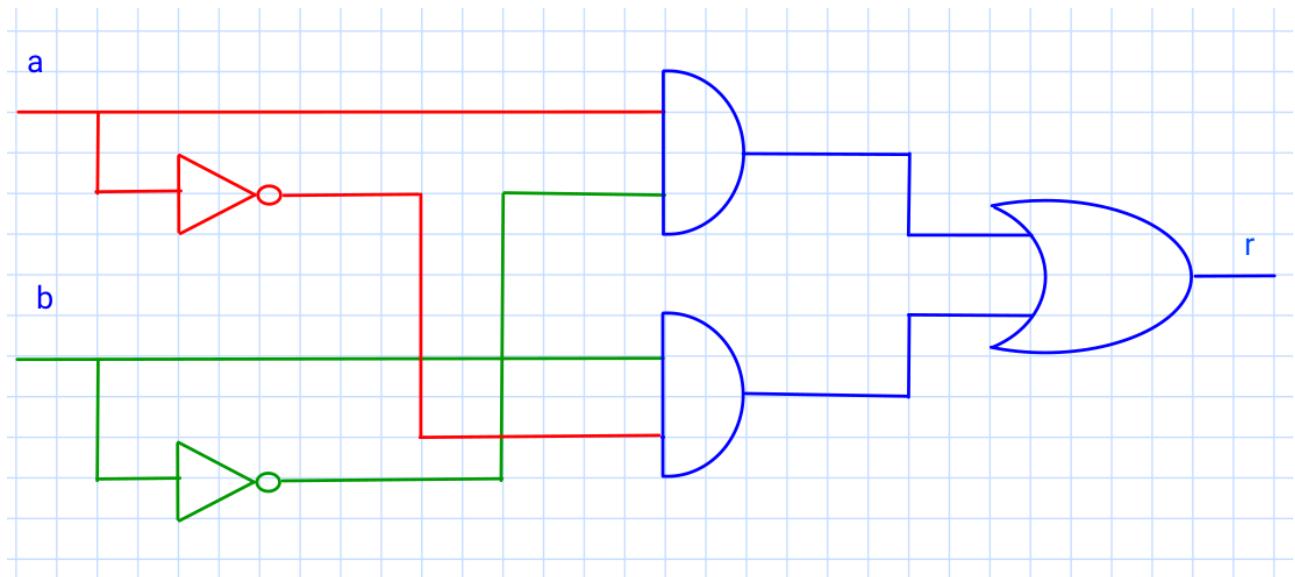
In generale, la porta logica xor, a due ingressi a e b , è rappresentata in questo modo:



L'espressione logica che la descrive è:

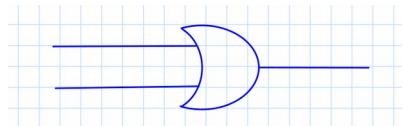
$$r = a \cdot \bar{b} + \bar{a} \cdot b$$

Dal punto di vista circuitale si rappresenta come segue:

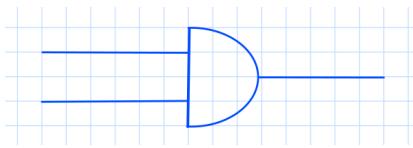


Ai fini della sua implementazione, allora, sono necessari:

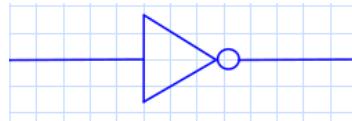
-un *or* ;



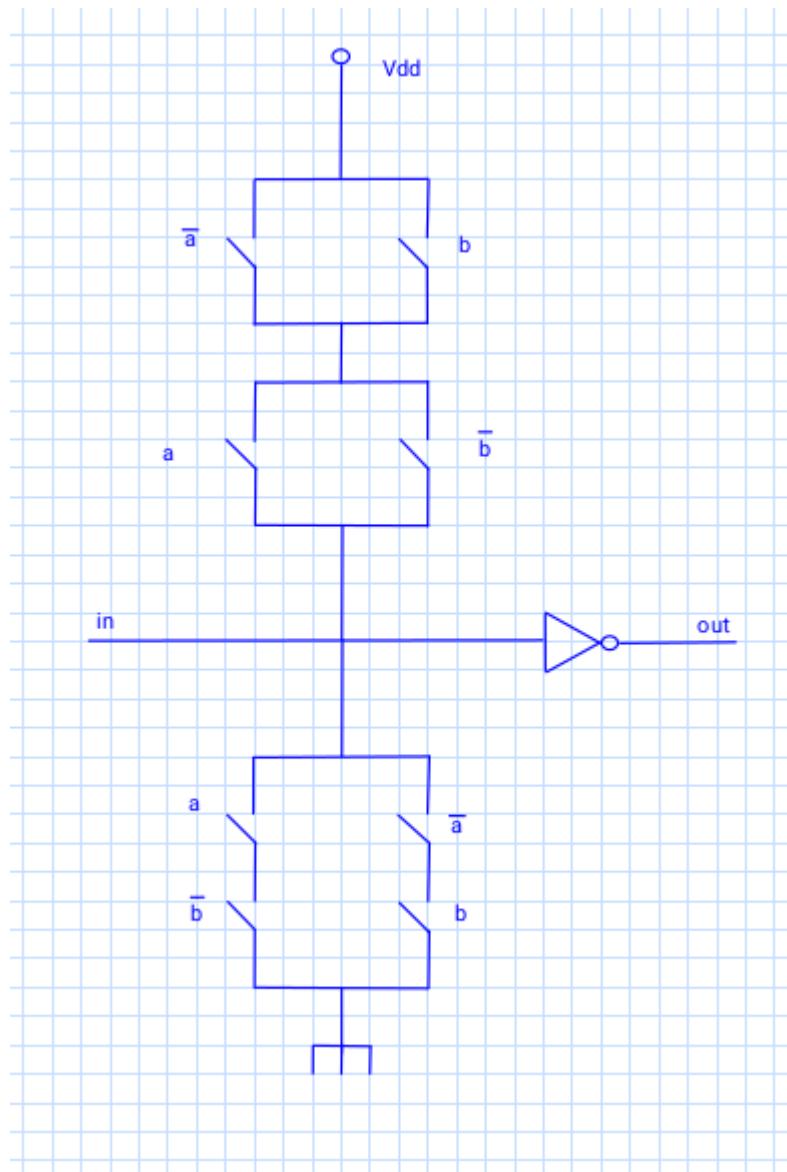
-due *and*;



-un *inverter*.



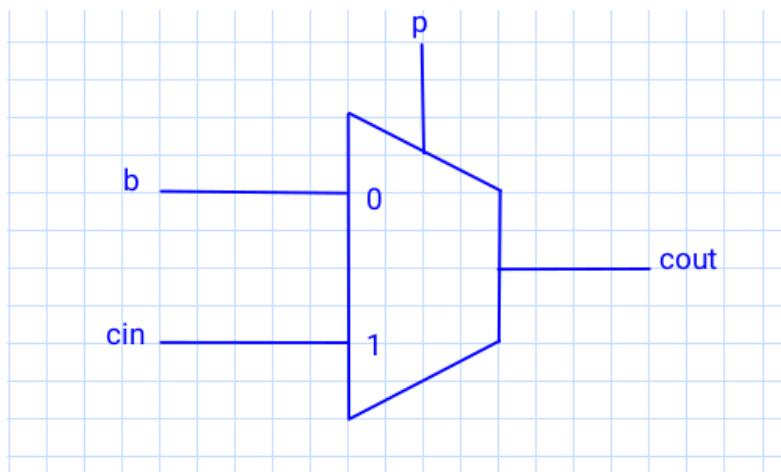
Allora, la sua espressione logica può essere descritta, tramite gli interruttori:



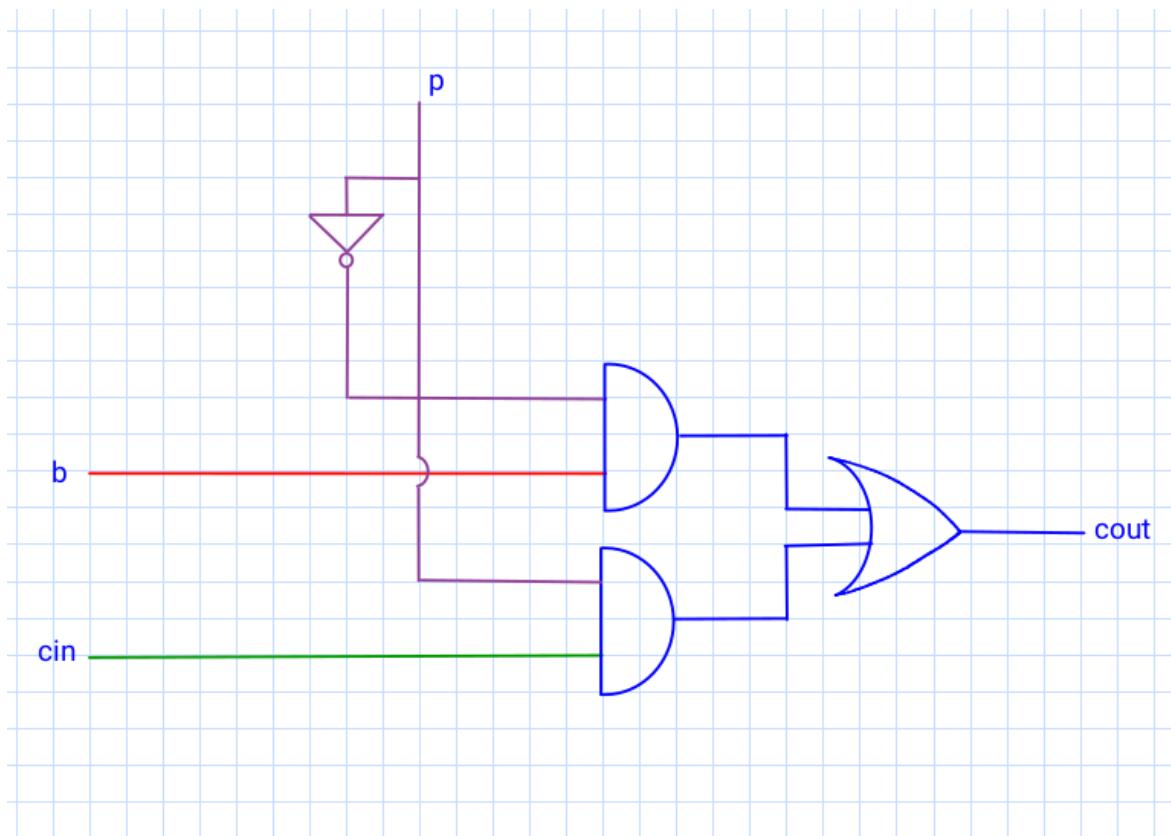
Multiplexer

Calcolo del riporto cout

Questo componente si occupa, tramite il selettore p , di generare il riporto $cout$.



Il Mux è stato realizzato sdoppiando il bit di controllo p , che è lo xor tra gli ingressi a e b , nel segnale stesso ed il suo negato.



Nello specifico, il Mux sceglie una dei due ingressi, b e cin (scegliere a , al posto di b , sarebbe stato indifferente) e lo instrada sull'uscita $cout$.

In particolare, posto p pari ad $a(XOR)b$, se il risultato di questa operazione sarà 0, allora verrà scelto b , se sarà 1, allora si instraderà cin .

Utilizzando i simboli “·” per rappresentare la funzione AND e “+” per la OR, si esplicitano le espressioni che descrivono p e $cout$, dove:

$$p = a \text{ XOR } b$$

$$cout = (\bar{p} \cdot b + p \cdot cin)$$

Allora:

$$p = (a \cdot \bar{b} + \bar{a} \cdot b)$$

$$\begin{aligned} cout &= \overline{(a \cdot \bar{b} + \bar{a} \cdot b)} \cdot b + (a \cdot \bar{b} + \bar{a} \cdot b) \cdot cin = \\ &= [(\bar{a} + b) \cdot (a + \bar{b})] \cdot b + (a \cdot \bar{b} \cdot cin) + (\bar{a} \cdot b \cdot cin) = \\ &= [(\bar{a} \cdot a) + (\bar{a} \cdot \bar{b}) + (a \cdot b) + (\bar{b} \cdot \bar{b})] \cdot b + (a \cdot \bar{b} \cdot cin) + (\bar{a} \cdot b \cdot cin) = \\ &= \bar{a} \cdot \bar{b} \cdot b + a \cdot b \cdot b + a \cdot \bar{b} \cdot cin + \bar{a} \cdot b \cdot cin = \\ &= a \cdot b + a \cdot \bar{b} \cdot cin + \bar{a} \cdot b \cdot cin \end{aligned}$$

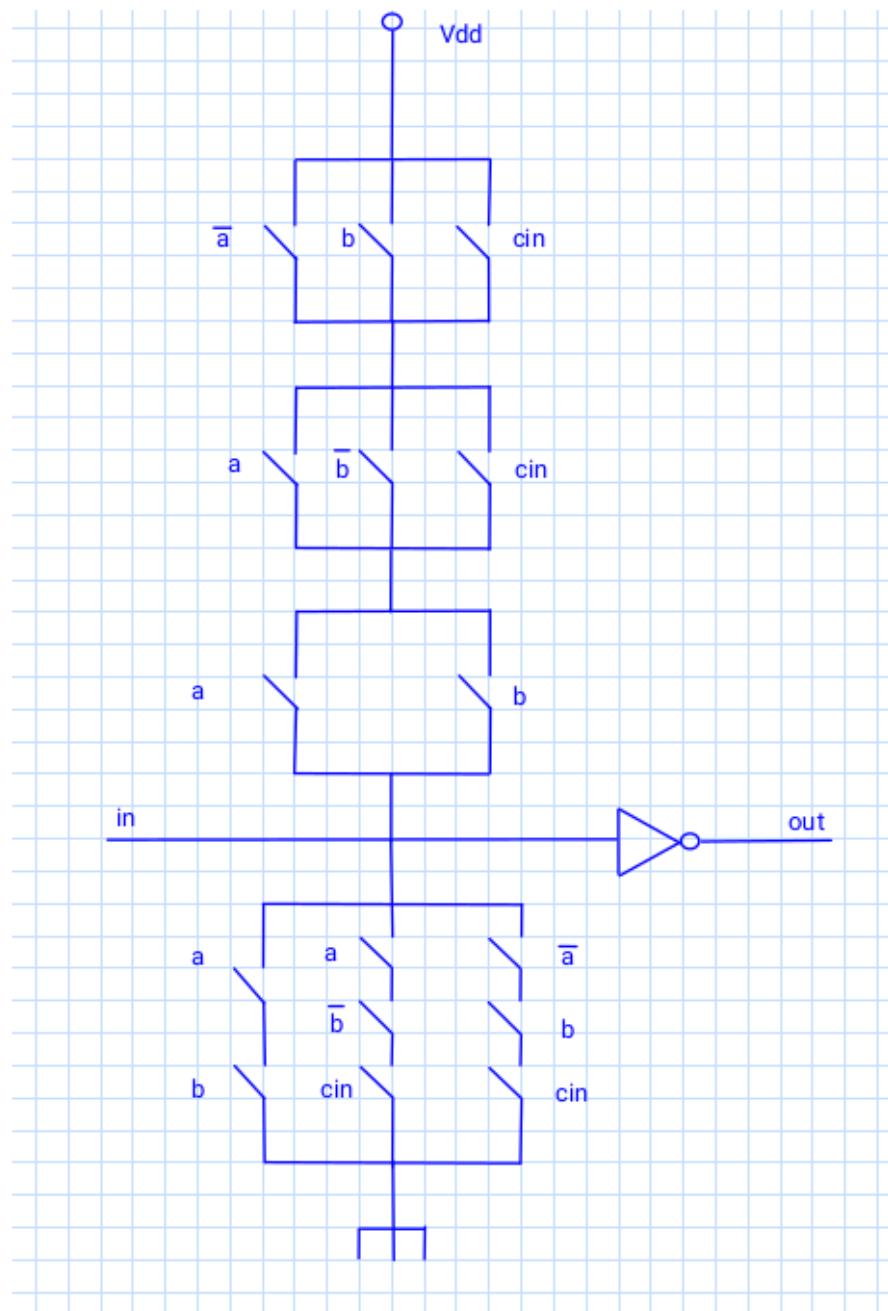
Dall'espressione che si è ottenuta dopo aver adottato delle semplificazioni tramite l'algebra booleana, si può vedere come si comporta il circuito da essa descritto in base ai vari casi:

-se tutte e tre le variabili, cioè a , b e cin , valgono 1, allora nessuno dei tre AND viene soddisfatto ed il valore della funzione di uscita $cout$ sarà 0;

-se l'AND tra a e b restituisce 1, ciò significa che i due segnali valgono entrambi 1, allora il riporto cout varrà 1, indipendentemente dal riporto iniziale cin;

-altrimenti, almeno uno tra i due ingressi a e b sarà negato, allora, nel caso in cui soltanto uno dei due valga 0 e il bit cin valga 1, la funzione cout restituirà 1.

Si riporta il Mux rappresentato tramite gli interruttori, con Reti di Pull-Up e di Pull-Down:



Calcolo della somma s

Come si evince dal modello circuitale riportato per il *Full Adder*, la somma s è generata dalla cascata tra due XOR.

L'espressione logica che la descrive è la seguente:

$$s = (a \text{ } XOR \text{ } b) \text{ } XOR \text{ } cin$$

Semplificando, si ottiene:

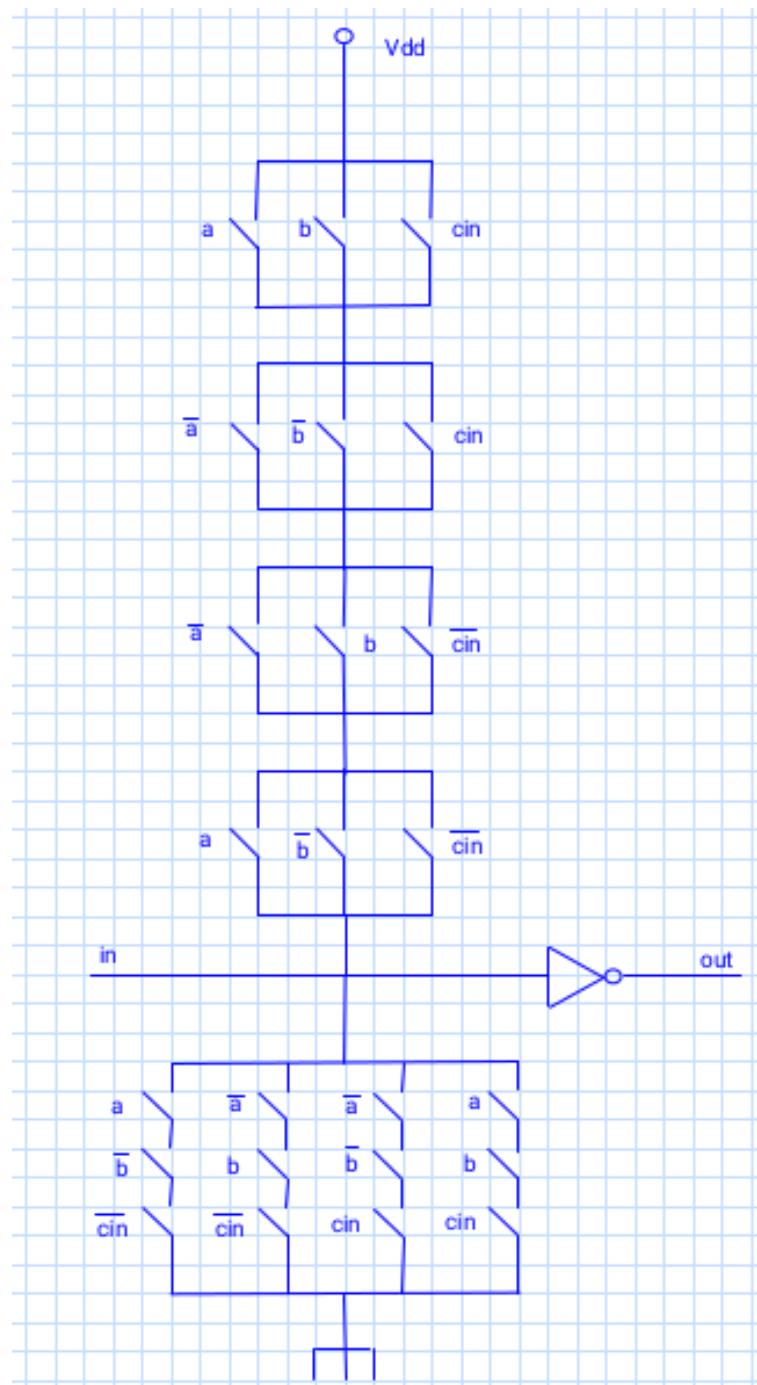
$$\begin{aligned} s &= (a \cdot \bar{b} + \bar{a} \cdot b) \text{ } XOR \text{ } cin = \\ &= (a \cdot \bar{b} + \bar{a} \cdot b) \cdot \bar{cin} + \overline{(a \cdot \bar{b} + \bar{a} \cdot b)} \cdot cin = \\ &= (a \cdot \bar{b} \cdot \bar{cin}) + (\bar{a} \cdot b \text{ } cadot \text{ } \bar{cin}) + [(\bar{a} + b) \cdot (a + \bar{b})] \cdot cin = \\ &= (a \cdot \bar{b} \cdot \bar{cin}) + (\bar{a} \cdot b \cdot \bar{cin}) + [\bar{a} \cdot a + \bar{a} \cdot \bar{b} + a \cdot b + b \cdot \bar{b}] \cdot cin = \\ &= a \cdot \bar{b} \cdot \bar{cin} + \bar{a} \cdot b \cdot \bar{cin} + \bar{a} \cdot \bar{b} \cdot cin + a \cdot b \cdot cin \end{aligned}$$

Considerando i primi tre AND, la funzione in uscita s restituisce 1 soltanto se due segnali su tre sono negati.

L'ultimo AND, infatti, restituisce 1 soltanto quando tutti e tre gli ingressi assumono valore logico alto, cioè 1.

Allora, la somma varrà 1, se tra i segnali ricevuti in ingresso, si verifica che un numero dispari di questi assume il valore 1; in caso contrario, la somma varrà 0.

Segue la rappresentazione del circuito che realizza la somma, tramite gli interruttori:



Full Adder in Reti di Pull-Up e di Pull-Down

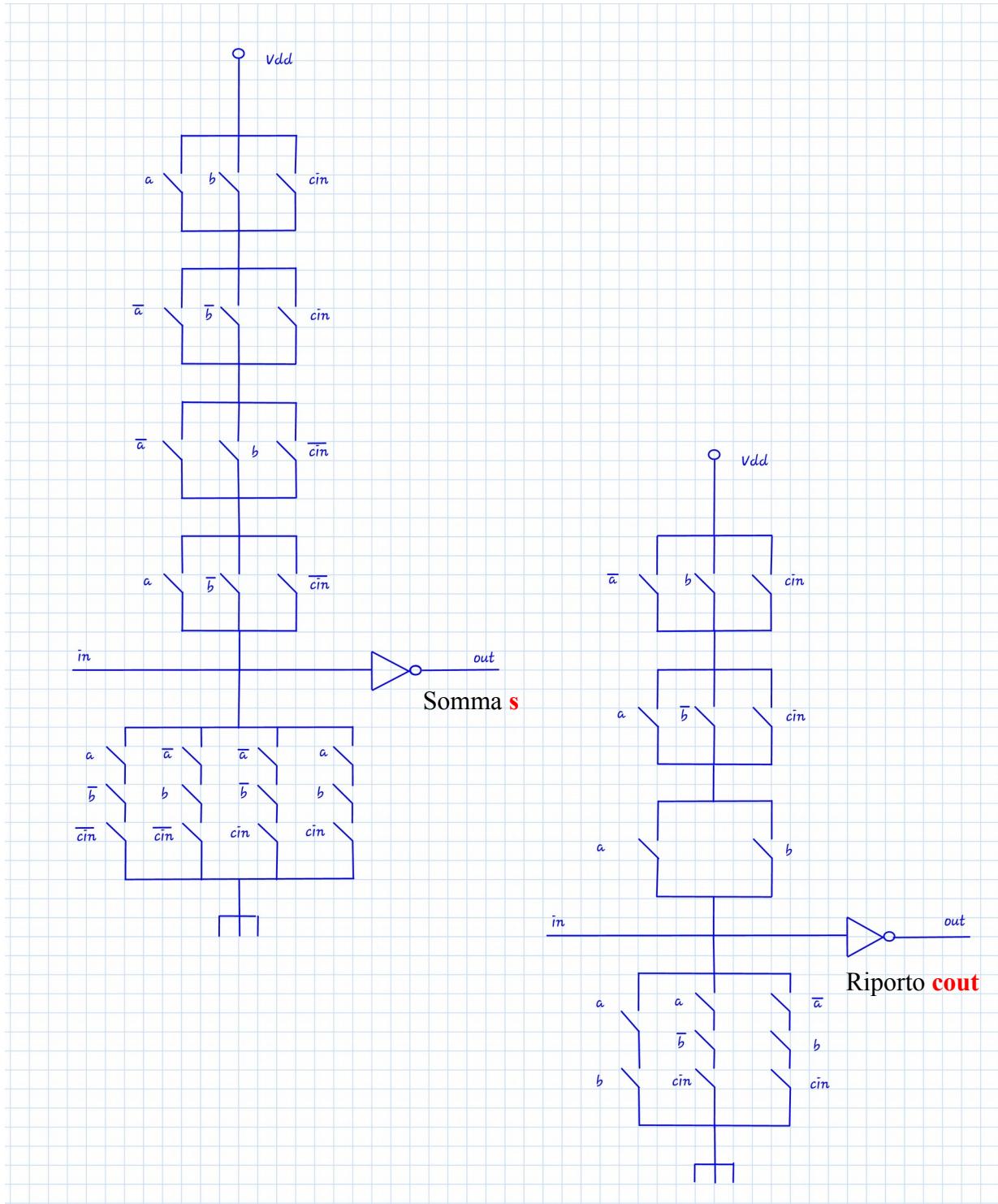


Tabella degli stati degli interruttori

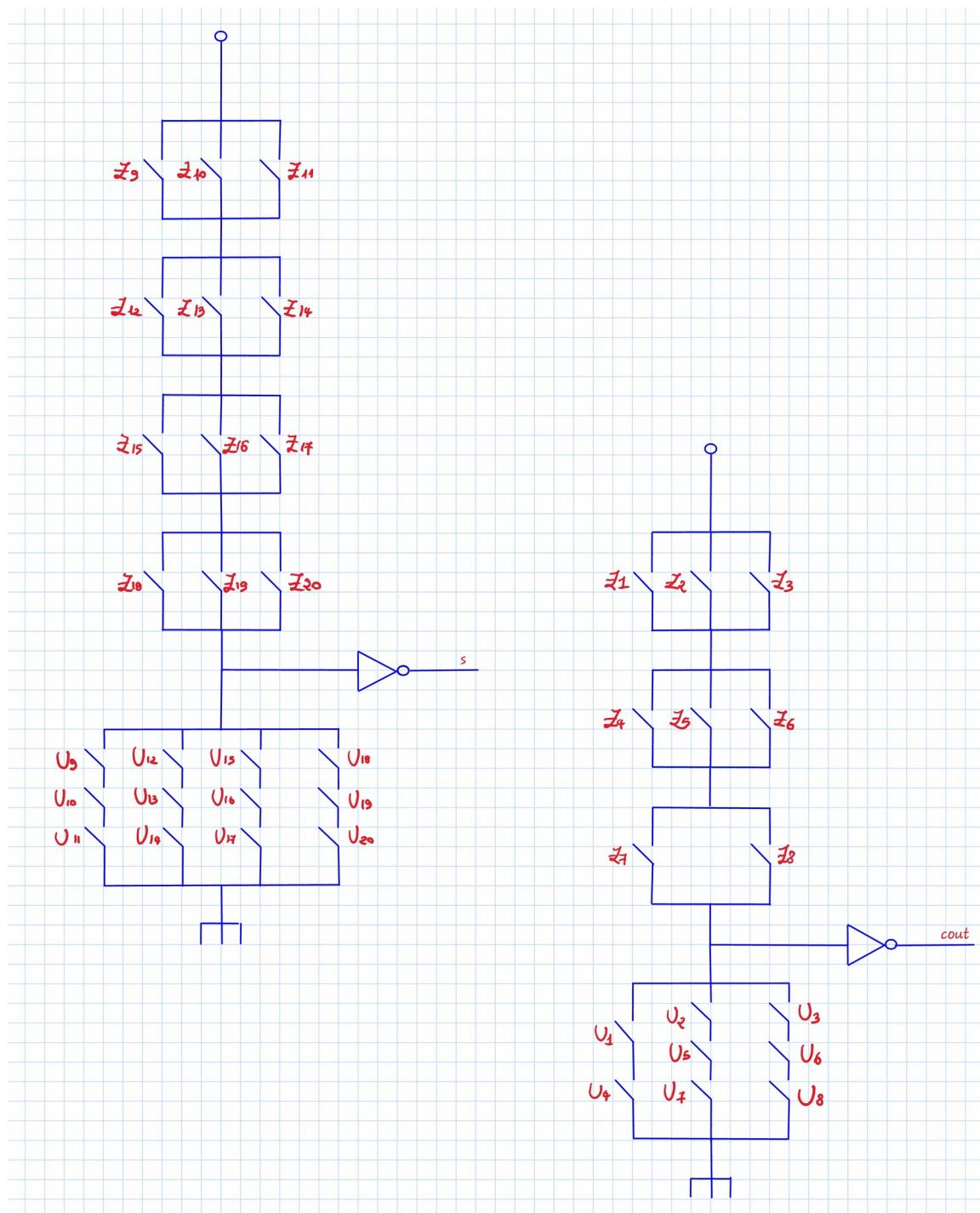
Ai fini di testare il circuito verranno scelte quattro diverse combinazioni di ingressi, quali:

- 1) $a=0, b=0, \text{cin}=1.$
- 2) $a=1, b=0, \text{cin}=1.$
- 3) $a=1, b=1, \text{cin}=1.$
- 4) $a=1, b=1, \text{cin}=0.$

Per questioni di leggibilità, si riporta il disegno del circuito completo comprensivo della nomenclatura assegnata a tutti gli interruttori dello stesso.

Per convenzione, si è scelto di indicare con il simbolo “ U ” gli interruttori appartenenti alla rete di *Pull-Down* e con il simbolo “ Z ” gli interruttori della rete di *Pull-Up*, essendo i primi di *tipo 1* ed i secondi di *tipo 0*.

Segue il disegno del circuito in Reti di Pull-Up e Pull-Down completo della nomenclatura utilizzata per gli interruttori:



Si riportano, di seguito, gli stati degli interruttori in base alle combinazioni di bit ingresso:

Prima combinazione

a=0, b=0, cin=1

Risultati attesi

s=1, cout=0

Tabella per la somma s

Interruttori Z	Stato	Interruttori U	Stato	Funzione
Z9	CHIUSO	U9	APERTO	A
Z10	CHIUSO	U13	APERTO	B
Z11	APERTO	U17	CHIUSO	CIN
Z12	APERTO	U12	CHIUSO	NOT(A)
Z13	APERTO	U10	CHIUSO	NOT(B)
Z14	APERTO	U20	CHIUSO	CIN
Z15	APERTO	U15	CHIUSO	NOT(A)
Z16	CHIUSO	U19	APERTO	B
Z17	CHIUSO	U11	APERTO	NOT(CIN)
Z18	CHIUSO	U18	APERTO	A
Z19	APERTO	U16	CHIUSO	NOT(B)
Z20	CHIUSO	U14	APERTO	NOT(CIN)

Tabella per il riporto cout

Interruttori Z	Stato	Interruttori U	Stato	Funzione
Z1	APERTO	U3	CHIUSO	NOT(A)
Z2	CHIUSO	U4	APERTO	B
Z3	APERTO	U7	CHIUSO	CIN
Z4	CHIUSO	U1	APERTO	A
Z5	APERTO	U5	CHIUSO	NOT(B)
Z6	APERTO	U8	CHIUSO	CIN
Z7	CHIUSO	U2	APERTO	A
Z8	CHIUSO	U6	APERTO	B

Già dalle prime due tavole, è possibile comprendere il funzionamento del circuito in esame, infatti, se si considera la parte che genera la somma s, si può vedere che, alla

ricezione dei segnali in ingresso $a=0$, $b=0$ e $cin=1$, accade che:

-nella *rete di Pull-Down*, viene reso disponibile un percorso di interruttori chiusi, cioè **U15**, **U16**, **U17** (i quali svolgono, rispettivamente, le funzioni di $\text{not}(a)$, $\text{not}(b)$ e cin) che fa passare il segnale della *massa*, che corrisponde allo 0 logico, all'inverter finale, che tramuta quest'ultimo in 1 logico. Quello che si ottiene è, effettivamente, il risultato corretto, poiché concorde a quello atteso pari a $s=1$.

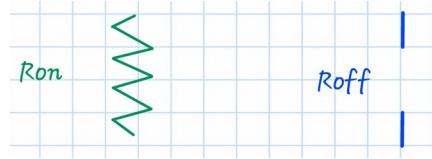
-nella *rete di Pull-Up*, invece, tutti i potenziali percorsi che potrebbero portare il segnale della Vdd all'inverter finale vengono interrotti, in ogni caso, dalla combinazione di interruttori **Z12**, **Z13**, **Z14** (che hanno la funzione di $\text{not}(a)$, $\text{not}(b)$, cin), poiché costituisce un parallelo di tre interruttori aperti. Questo comporta che la Vdd, pur attraversando la prima di interruttori, si blocchi alla seconda, senza raggiungere la terza.

Nella rappresentazione del circuito equivalente gli interruttori verranno sostituiti dalle rispettive resistenze.

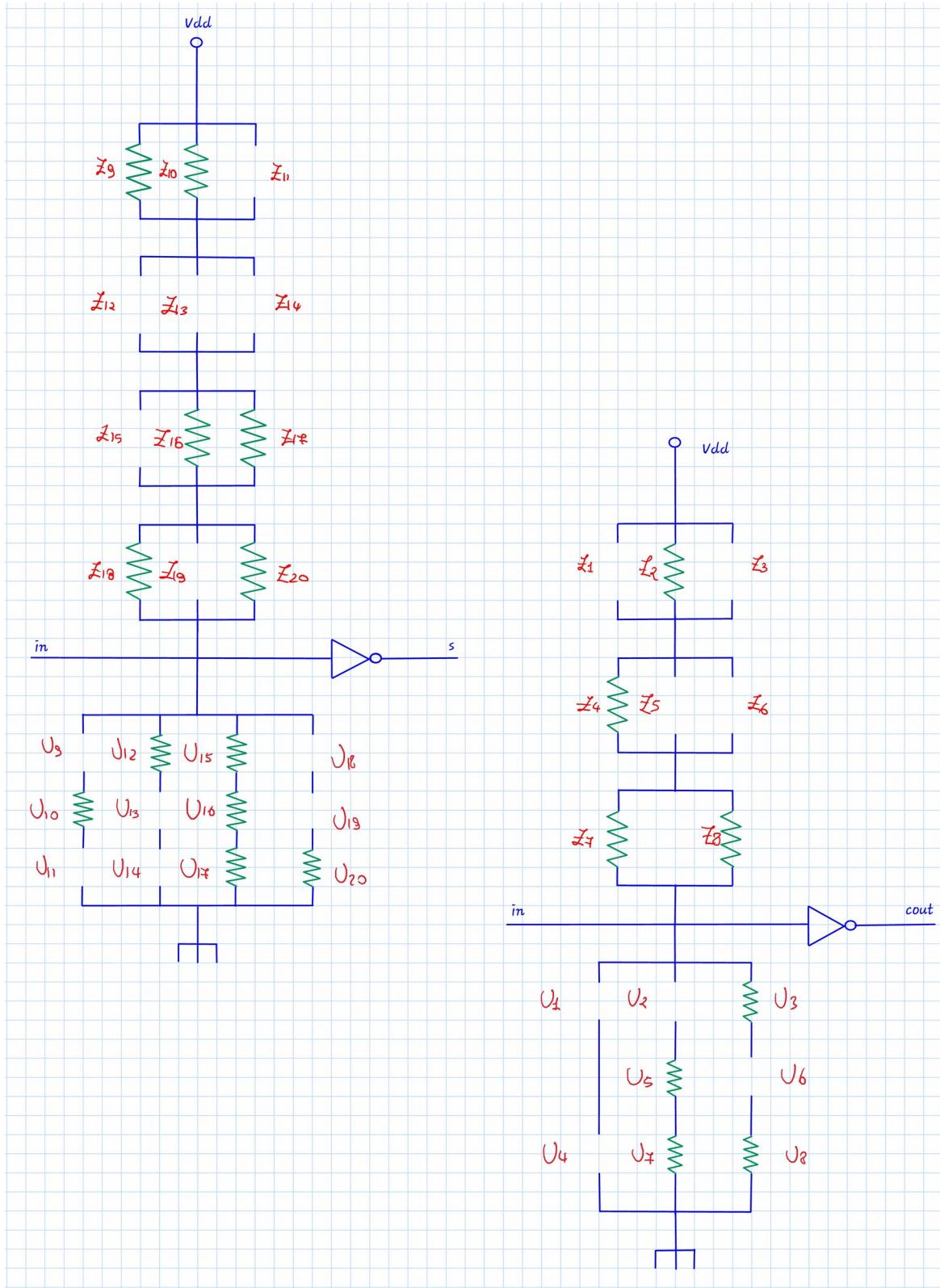
Nello specifico, all'interruttore contraddistinto dallo stato "aperto" verrà sostituita la resistenza R_{off} ed all'interruttore che si trova nello stato "chiuso" verrà, invece, sostituita la resistenza R_{on} .

Per comodità, si è scelto di disegnare con il convenzionale simbolo di resistenza soltanto le resistenze R_{on} , poiché le resistenze R_{off} , rappresentando un valore di resistenza che è massimo, possono essere sostituite da un circuito aperto.

La simbologia usata, allora, è la seguente:



Si riporta il circuito equivalente:



Seconda combinazione

a=1, b=0, cin=1

Risultati attesi

s=0, cout=1

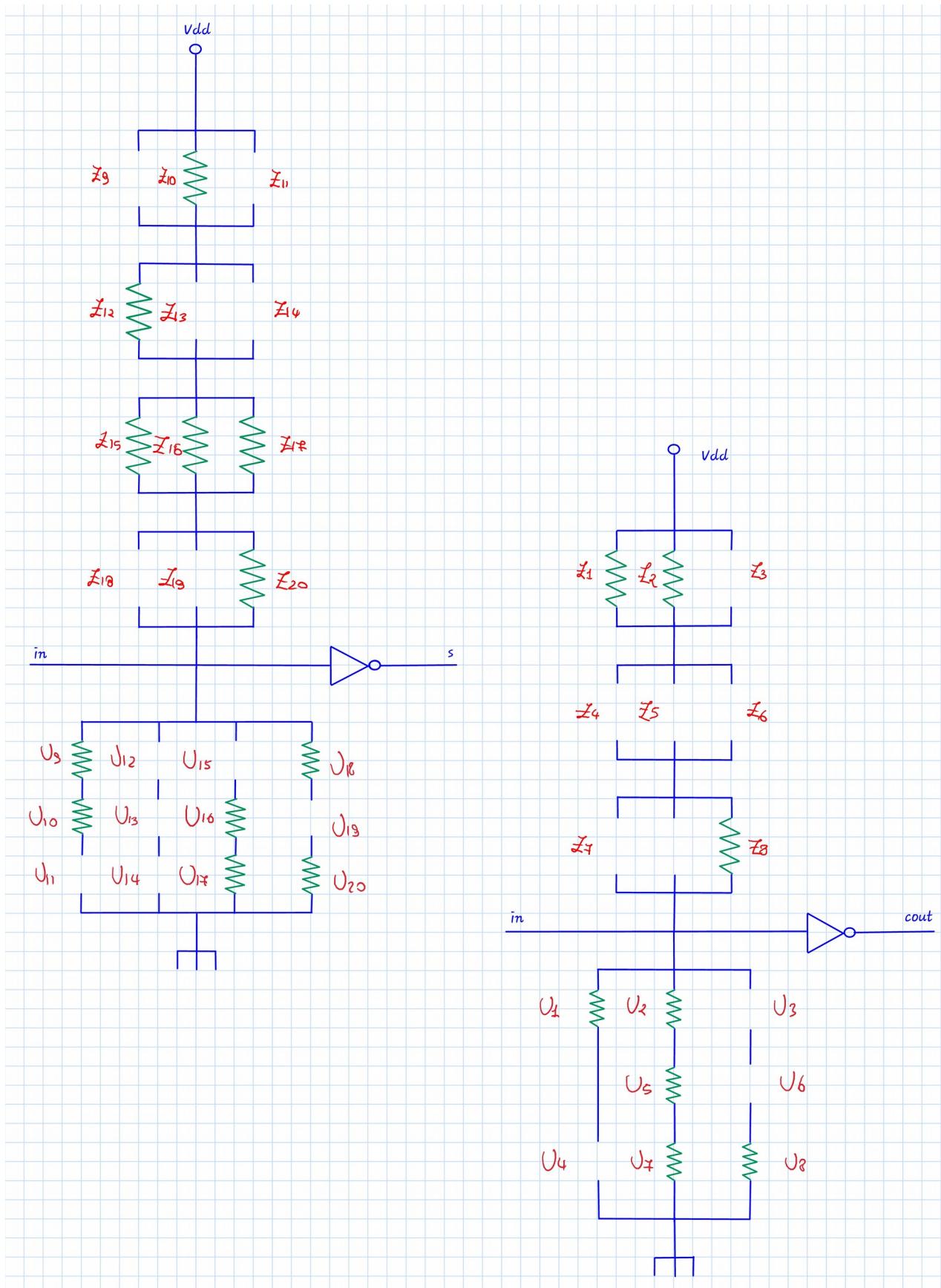
Tabella per la somma s

Interruttori Z	Stato	Interruttori U	Stato	Funzione
Z9	APERTO	U9	CHIUSO	A
Z10	CHIUSO	U13	APERTO	B
Z11	APERTO	U17	CHIUSO	CIN
Z12	CHIUSO	U12	APERTO	NOT(A)
Z13	APERTO	U10	CHIUSO	NOT(B)
Z14	APERTO	U20	CHIUSO	CIN
Z15	CHIUSO	U15	APERTO	NOT(A)
Z16	CHIUSO	U19	APERTO	B
Z17	CHIUSO	U11	APERTO	NOT(CIN)
Z18	APERTO	U18	CHIUSO	A
Z19	APERTO	U16	CHIUSO	NOT(B)
Z20	CHIUSO	U14	APERTO	NOT(CIN)

Tabella per il riporto cout

Interruttori Z	Stato	Interruttori U	Stato	Funzione
Z1	CHIUSO	U3	APERTO	NOT(A)
Z2	CHIUSO	U4	APERTO	B
Z3	APERTO	U7	CHIUSO	CIN
Z4	APERTO	U1	CHIUSO	A
Z5	APERTO	U5	CHIUSO	NOT(B)
Z6	APERTO	U8	CHIUSO	CIN
Z7	APERTO	U2	CHIUSO	A
Z8	CHIUSO	U6	APERTO	B

Il circuito equivalente è:



Terza combinazione
a=1, b=1, cin=1

Risultati attesi
s=1, cout=1

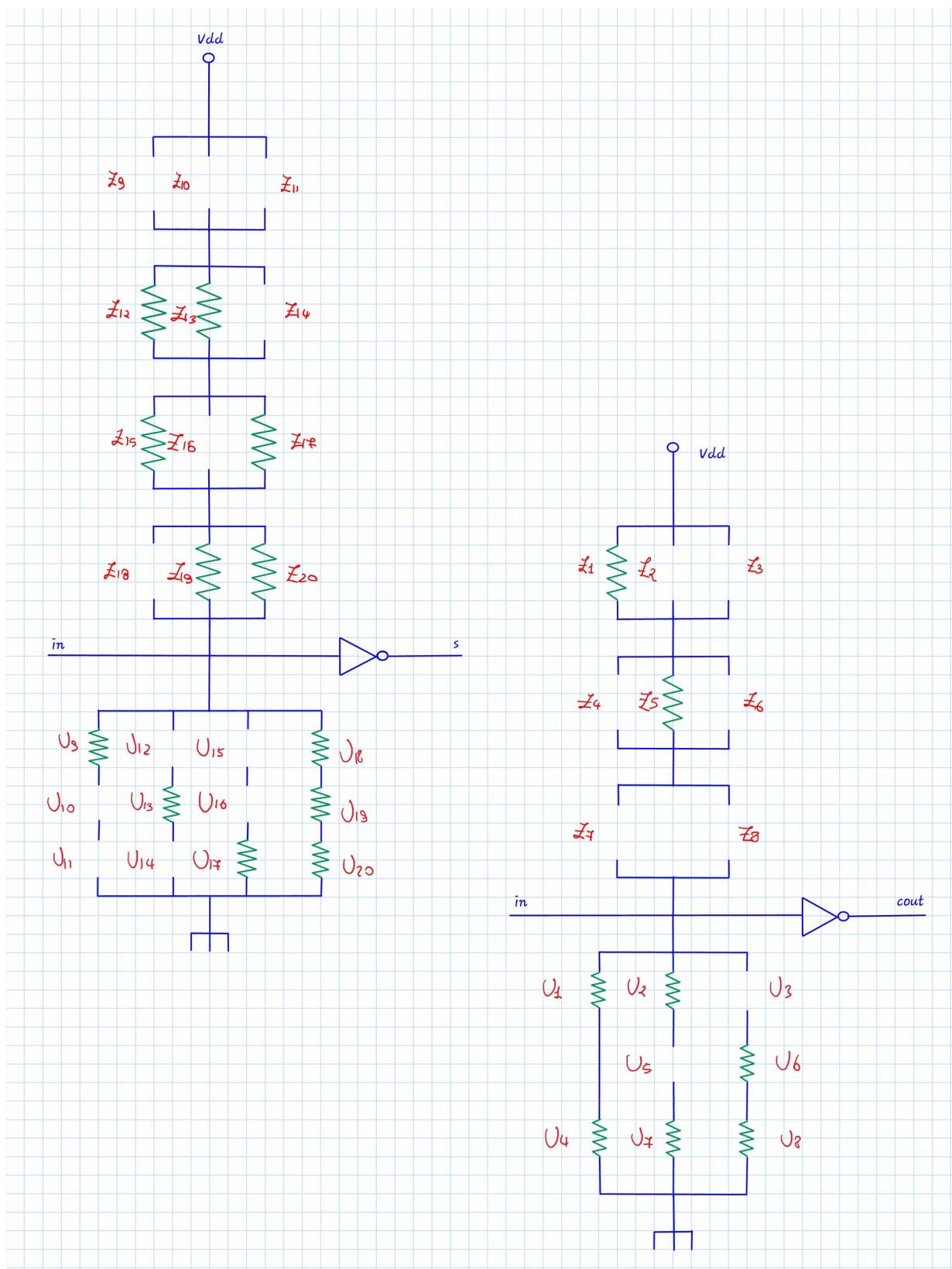
Tabella per la somma s

Interruttori Z	Stato	Interruttori U	Stato	Funzione
Z9	APERTO	U9	CHIUSO	A
Z10	APERTO	U13	CHIUSO	B
Z11	APERTO	U17	CHIUSO	CIN
Z12	CHIUSO	U12	APERTO	NOT(A)
Z13	CHIUSO	U10	APERTO	NOT(B)
Z14	APERTO	U20	CHIUSO	CIN
Z15	CHIUSO	U15	APERTO	NOT(A)
Z16	APERTO	U19	CHIUSO	B
Z17	CHIUSO	U11	APERTO	NOT(CIN)
Z18	APERTO	U18	CHIUSO	A
Z19	CHIUSO	U16	APERTO	NOT(B)
Z20	CHIUSO	U14	APERTO	NOT(CIN)

Tabella per il riporto cout

Interruttori Z	Stato	Interruttori U	Stato	Funzione
Z1	CHIUSO	U3	APERTO	NOT(A)
Z2	APERTO	U4	CHIUSO	B
Z3	APERTO	U7	CHIUSO	CIN
Z4	APERTO	U1	CHIUSO	A
Z5	CHIUSO	U5	APERTO	NOT(B)
Z6	APERTO	U8	CHIUSO	CIN
Z7	APERTO	U2	CHIUSO	A
Z8	APERTO	U6	CHIUSO	B

Si riporta il circuito equivalente:



Quarta combinazione

a=1, b=1, cin=0

Risultati attesi

s=0, cout=1

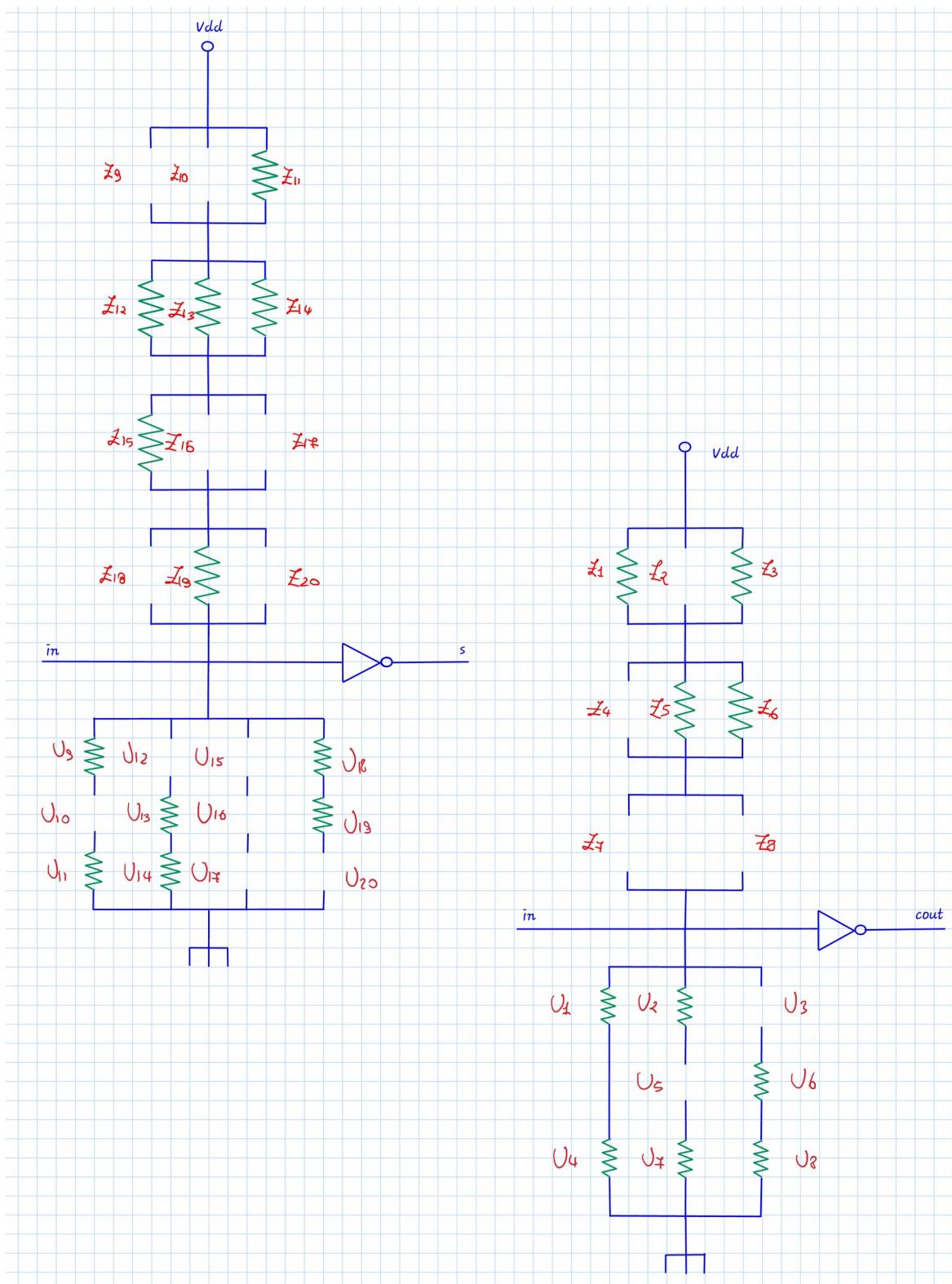
Tabella per la somma s

Interruttori Z	Stato	Interruttori U	Stato	Funzione
Z9	APERTO	U9	CHIUSO	A
Z10	APERTO	U13	CHIUSO	B
Z11	CHIUSO	U17	APERTO	CIN
Z12	CHIUSO	U12	APERTO	NOT(A)
Z13	CHIUSO	U10	APERTO	NOT(B)
Z14	CHIUSO	U20	APERTO	CIN
Z15	CHIUSO	U15	APERTO	NOT(A)
Z16	APERTO	U19	CHIUSO	B
Z17	APERTO	U11	CHIUSO	NOT(CIN)
Z18	APERTO	U18	CHIUSO	A
Z19	CHIUSO	U16	APERTO	NOT(B)
Z20	APERTO	U14	CHIUSO	NOT(CIN)

Tabella per il riporto cout

Interruttori Z	Stato	Interruttori U	Stato	Funzione
Z1	CHIUSO	U3	APERTO	NOT(A)
Z2	APERTO	U4	CHIUSO	B
Z3	CHIUSO	U7	APERTO	CIN
Z4	APERTO	U1	CHIUSO	A
Z5	CHIUSO	U5	APERTO	NOT(B)
Z6	CHIUSO	U8	APERTO	CIN
Z7	APERTO	U2	CHIUSO	A
Z8	APERTO	U6	CHIUSO	B

Segue il circuito equivalente:



Si può vedere, dalla verifica effettuata per le quattro combinazioni di bit in ingresso, come il circuito funzioni correttamente.