Capitolo 2.2

domenica 24 dicembre 2023 13:22

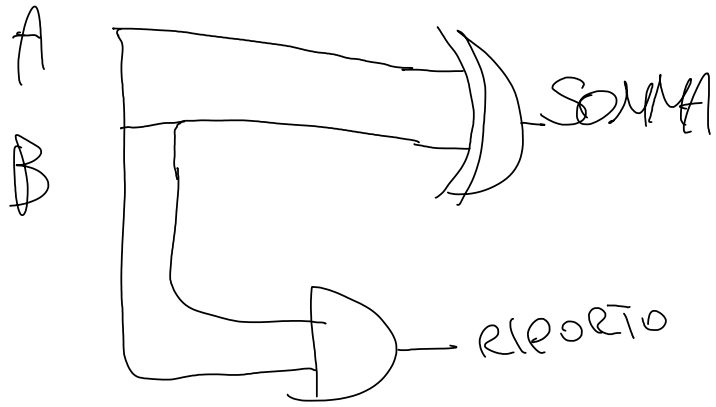
Home

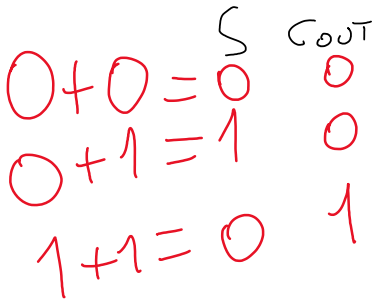
BLOCCHI FUNZIONALI

n.b gli input e gli output possono essere riempiti da alt o altri blocchi

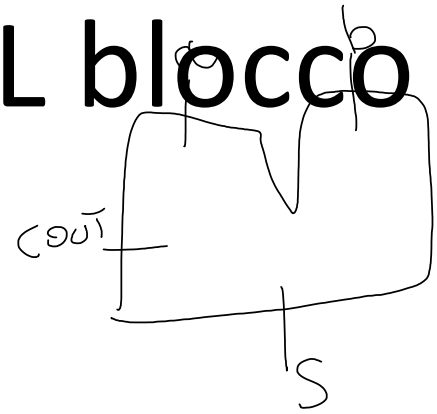
HALF ADDER

È un blocco funzionale che rappresenta la somma e prende 2 input e dà due output(S e Cout)
Il riporto è dato dagli and dei due e il risultato dallo xor dei due
Sarebbe tipo così:





IL blocco



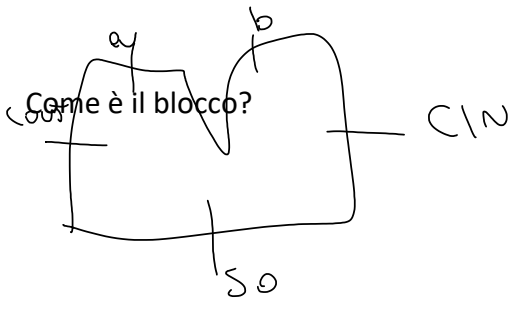
FULL ADDER

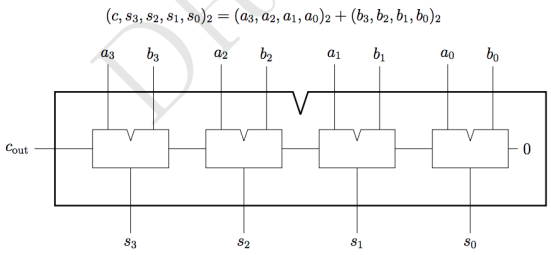
Il full addder è tipo un half adder ma tiene conto dei resti,ti permette di sommare più bit insieme(dividendoli due a due)

COME SI RAPPRESENTA dentro?

Sono due half adder e un or che tiene del resto

Come è il blocco?





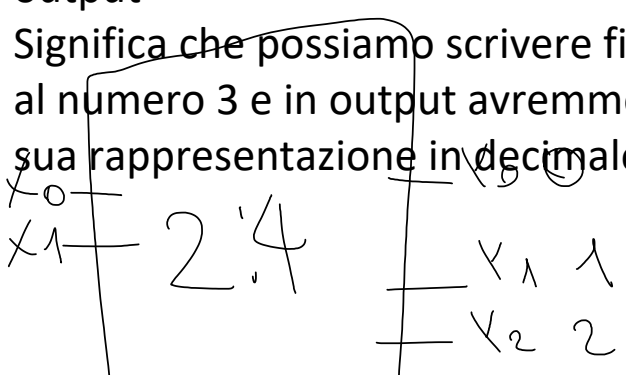
Con le sottrazioni basta fare la somma con il complemento a due

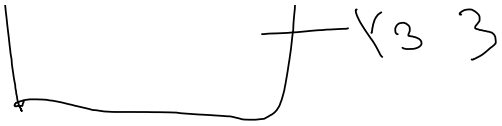
DECODER

IL DECODER è un blocco funzionale che ti permette di prendere tutte le varie combinazioni e ha n:2^n
Ovvero n input e 2^n output

Rappresenta la conversione binaria a decimale

Come funziona?
Abbiamo ad esempio 2 input e 4 output
Significa che possiamo scrivere fino al numero 3 e in output avremmo la sua rappresentazione in decimale





La y si accenderà in base all'input che mettiamo, se scrivessimo 01 si accenderebbe y1 se scrivessimo 11 si accenderebbe y3

Circuito e tabella di verità:
Se vedete si accende una sola y che è quella che ci serve infatti, e nel circuito ci sono tutte le combinazioni possibili

Se $n = 1$ abbiamo un solo input x_0 e due output y_1, y_0; y_0 deve essere 1 quando $x_0 = 0$ mentre y_1 deve essere 1 quando $x_0 = 1$:

x_0	y_1	y_0
0	0	1
1	1	0

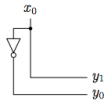


Figura 1: DECODER 1:2

Se $n = 2$ abbiamo due input x_0, x_1 e quattro output y_3, y_2, y_1, y_0

x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

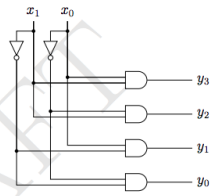
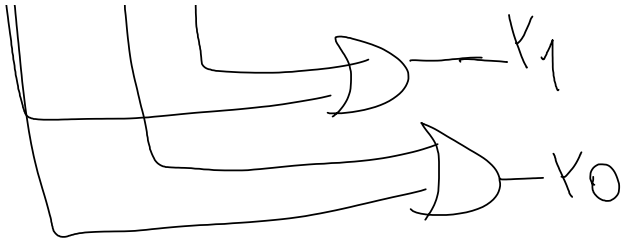


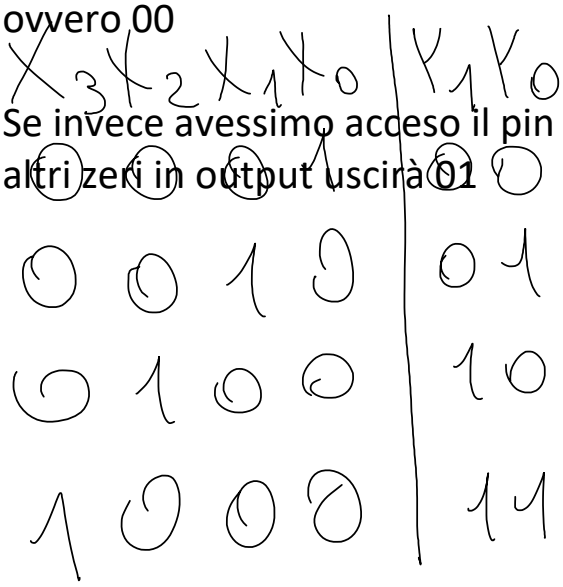
Figura 2: DECODER 2:4

ENCODER

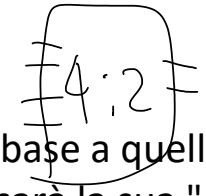
È l'opposto del decoder ovvero hai 2^n:n
Abbiamo in input il numero in decimale che si accende invece in output avremo il numero in binario
Il blocco funzionale è così:



Perché x0 viene ignorato?
Perché x0 non vale un cazzo c'è se tu hai tutte che valgono 0 e x0 che vale 1 basta fare l'or tra gli zeri e avrai il tuo risultato ovvero 00



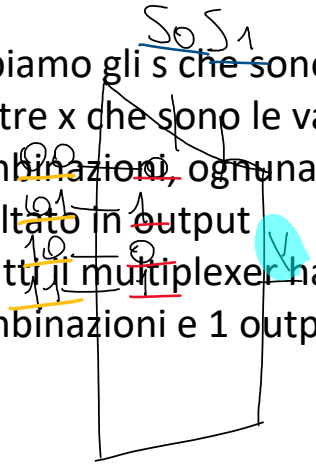
Se invece avessimo acceso il pin 1 e gli altri zeri in output uscirà 01



In base a quello che si accende in output ci sarà la sua "conversione"

MULTIPLEXER

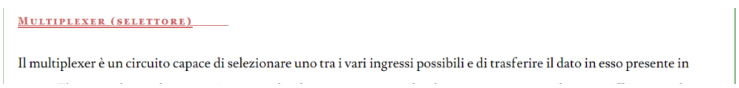
il multiplexer è un blocco funzionale che è tipo un if, per spiegarlo meglio facciamo subito un esempio



Abbiamo gli s che sono gli input e le nostre x che sono le varie possibili combinazioni, ognuna di loro ritorna un risultato in output
Infatti il multiplexer ha n input e avrà 2^n combinazioni e 1 output

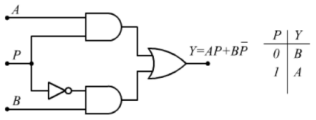
Questo blocco se riceverà 01 o 11 ritornerà 1 negli altri casi 00 e 10 ci sarà 0

Blu sono gli input rosso sono i return e arancione sono le varie possibilità e il printf è l'azzurro


MULTIPLEXER (SELETTORE)

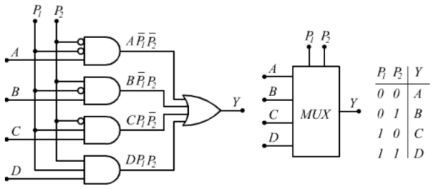
Il multiplexer è un circuito capace di selezionare uno tra i vari ingressi possibili e di trasferire il dato in esso presente in

uscita. E' sempre dotato di uno o più ingressi di selezione: m ingressi di selezione servono per pilotare $n=2^m$ ingressi dati.



P	Y
0	B
1	A

Nel disegno precedente è rappresentato un multiplexer nella sua forma più semplice, con 2 ingressi dati ed 1 ingresso di selezione.
E' l'ingresso di selezione P a decidere quale delle due variabili di ingresso viene trasferita in uscita. Se P=1 si ha Y=A; se invece P=0 si ha Y=B.
Qui di seguito è disegnato un multiplexer con 4 ingressi dati e 2 ingressi di selezione.



P1	P2	Y
0	0	A
0	1	B
1	0	C
1	1	D

La funzione logica che viene realizzata è

CON IL DECODER.

Possiamo costruire un MULTIPLEXER 4:1 anche usando un DECODER 2:4, quattro porte AND e una porta OR, come in Figura 4

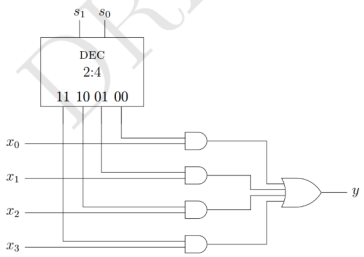


Figura 4: MULTIPLEXER 4:1