

Macchina a Stati

Creo macchina stato che rappresenti il comportamento descritto.

Per semplicità una singola lettera (es: A) implica che solo A è 1, mentre B è 0.

$A \approx A\bar{B}$, $B \approx \bar{A}B$.

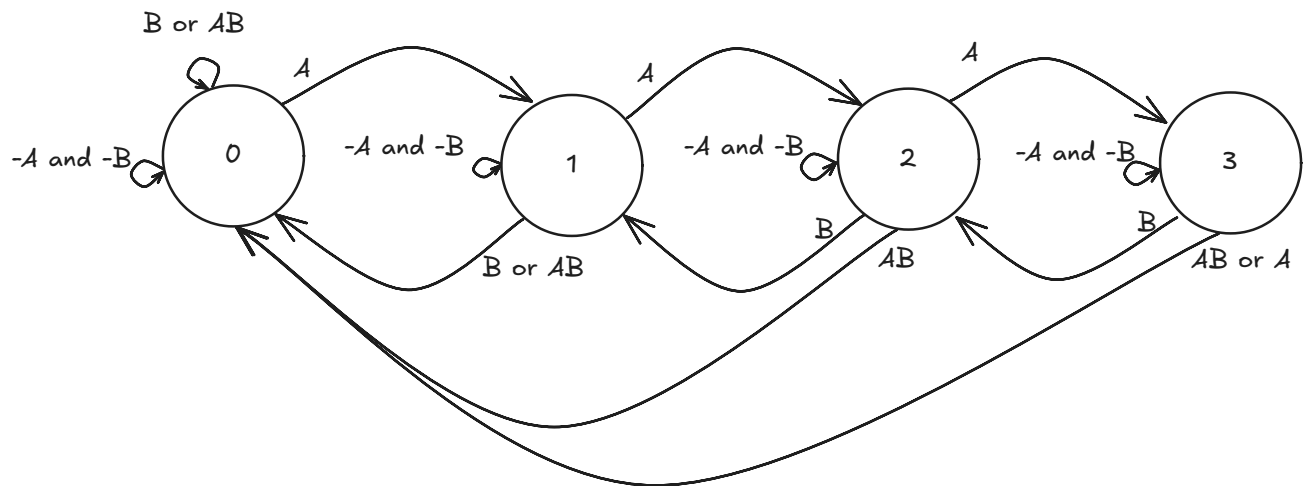


Tabella di Verità

Creo tabella di verità con stati e input come asse, e lo stato successivo all'interno delle celle.

Come si vede dalla macchina di stato, A aumenta, B decrementa, AB resettano, $\bar{A}\bar{B}$ mantiene stato.

Nella tabella, colonna $\bar{A}\bar{B}$, riscrivo lo stato come numero invece che mettere S come indicatore che l'output è lo stato corrente per trasformarlo nella codifica a 2 bit più velocemente ad occhio.

Stato ↓	Input →	$A\bar{B}$	$\bar{A}B$	AB	$\bar{A}\bar{B}$
0		1	0	0	0
1		2	0	0	1
2		3	1	0	2
3		0	2	0	3

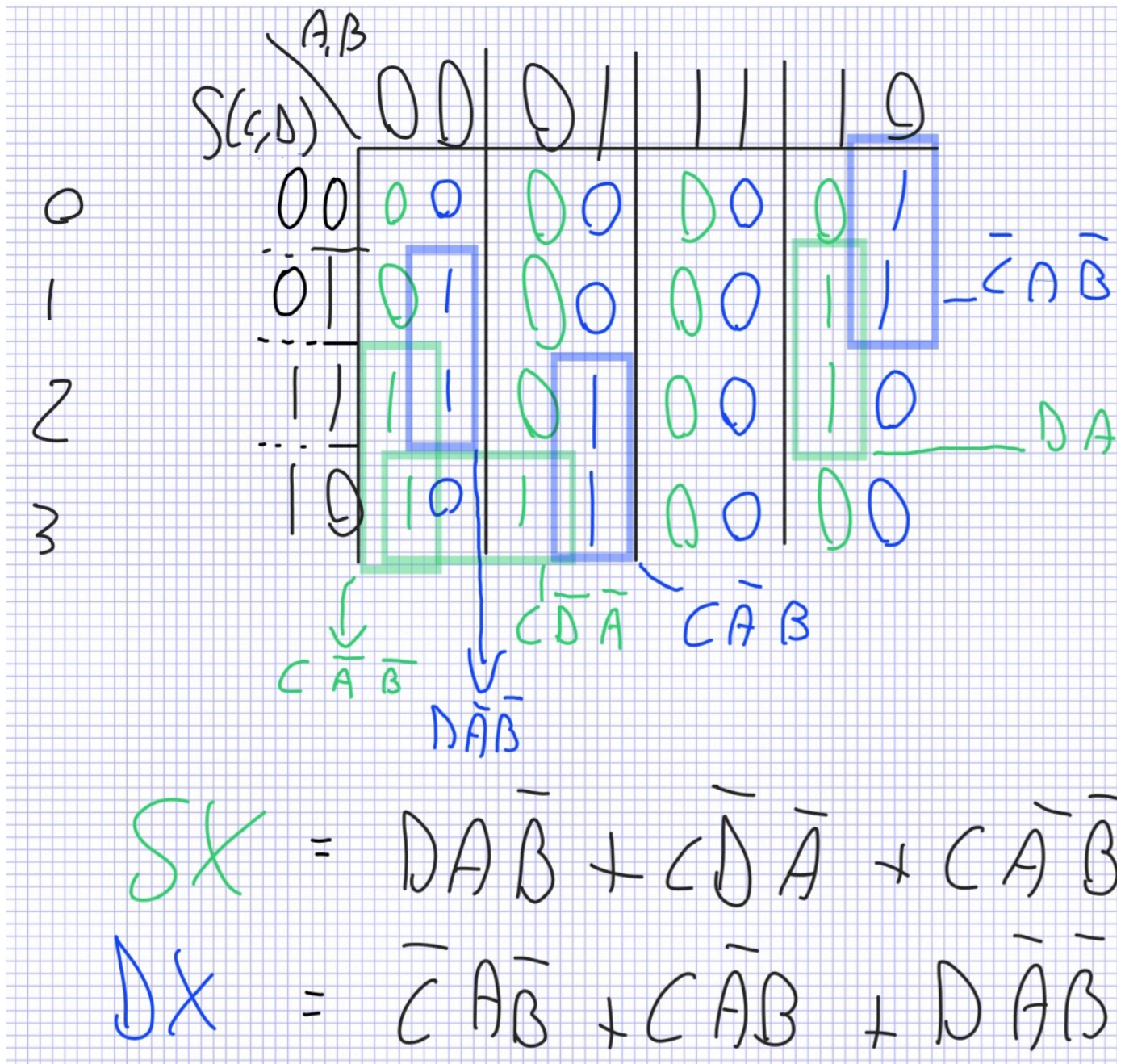
Mappa di Karnaugh

Converto i 4 stati in bit C e D ($2^2 = 4$ stati rappresentabili $\Rightarrow 2$ bit), e a causa delle restrizioni delle mappe di Karnaugh (differenza di un singolo bit tra colonne/righe adiacenti) decido di usare una codifica che associa $2_{dec} = 11_{bin}$ e $3_{dec} = 10_{bin}$ per averli in ordine guardando la tabella.

Gli stati 0 e 1 sono uguali alla rappresentazione posizionale binaria.

Gli stati output saranno ovviamente anch'essi in due bit, verde il sinistro e blu il destro nella mappa di Karnaugh per chiarezza.

L'output Z non l'ho messo nella mappa di Karnaugh in quanto ho scelto di ricavarlo dalla State Memory (i DFLIPFLOP) successivamente data la semplicità dell'implementazione (un singolo AND).



Sfruttando le tecniche di assorbimento identico gruppi di 1. Purtroppo in questa tabella ho solo coppie e non gruppi più grandi (da 2^n bit), che non è il migliore dei casi.

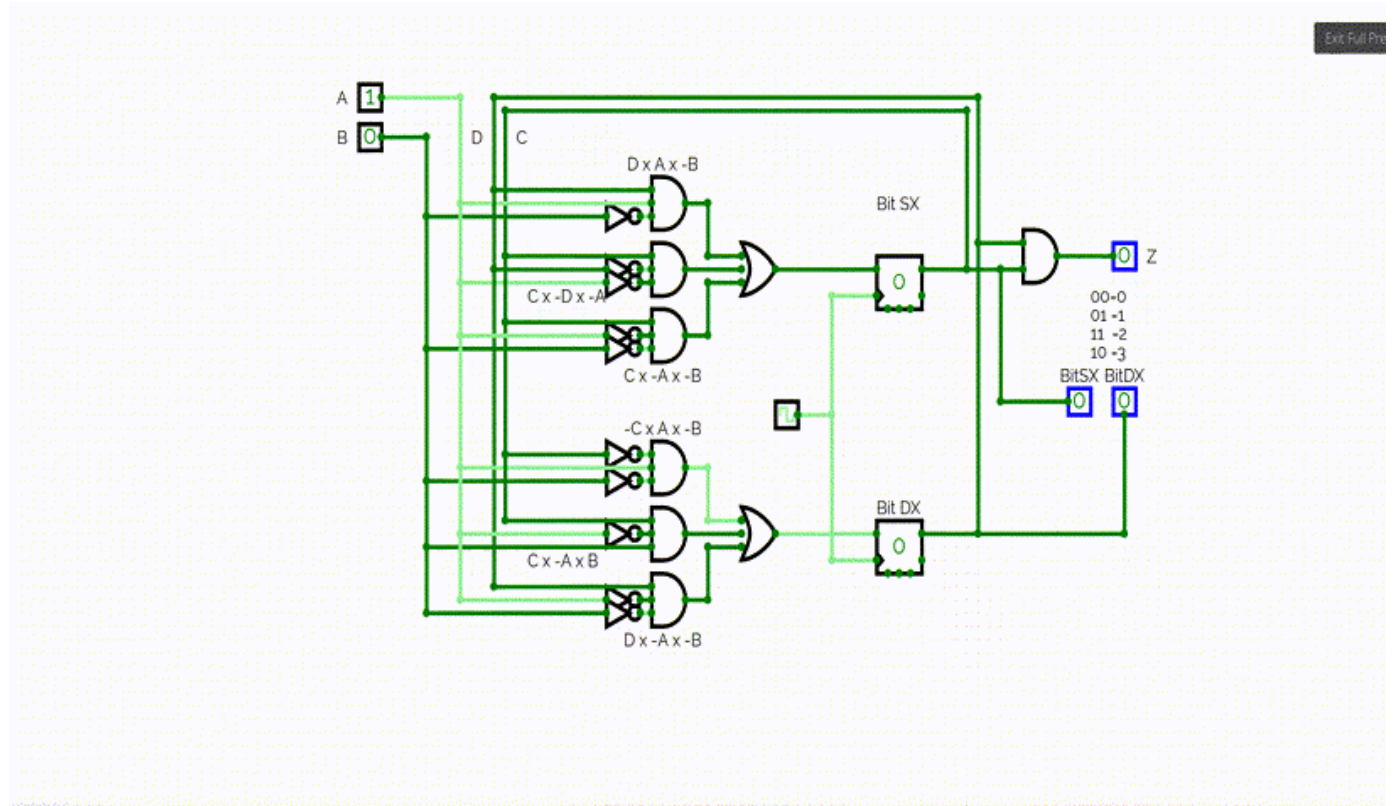
Ottengo quindi, guardando i bit in comune dei gruppi, le forme disgiuntive, separatamente per bit stato SX e DX.

Circuito

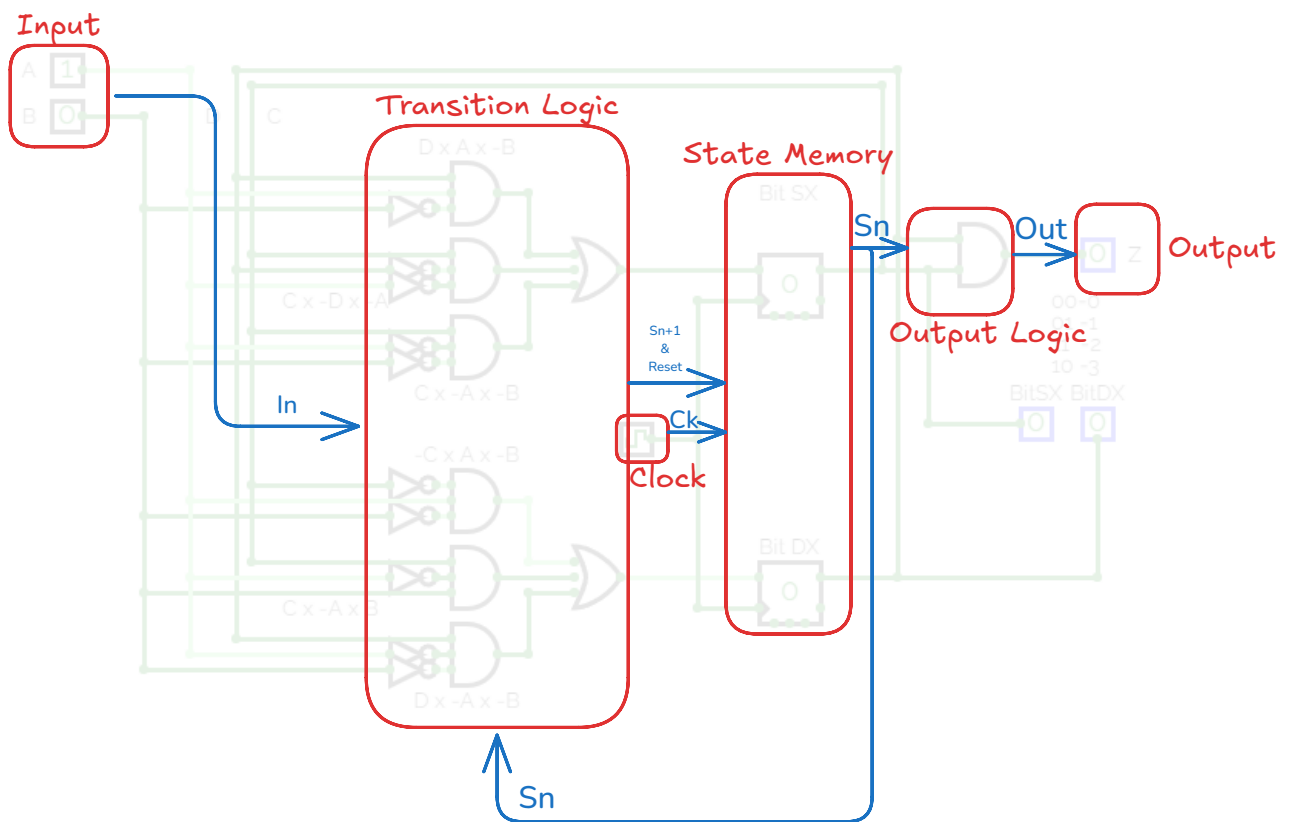
Queste formule le formerò con i gate logici elementari che prenderanno in input A, B, C, D (A e B input, C e D codifica stato attuale, poi precedente) e che collegherò individualmente ai 2 DFLIPFLOP (DFLIPFLOP bit SX e DFLIPFLOP bit DX) che rappresenteranno lo stato attuale letti insieme, e che porrò ad AND nel circuito per l'output Z .

Quando avrò $SX = 1$ e $DX = 1$ (mia codifica di 2), l'output Z sarà 1.

Avrò quindi bisogno di due input A e B , dei gate elementari NOT, AND, e OR per le forme disgiuntive, due DFLIPFLOP per lo stato dei 2 bit, il clock, e infine un AND per Z .



Modello di Moore



[Link a circuitverse](#)