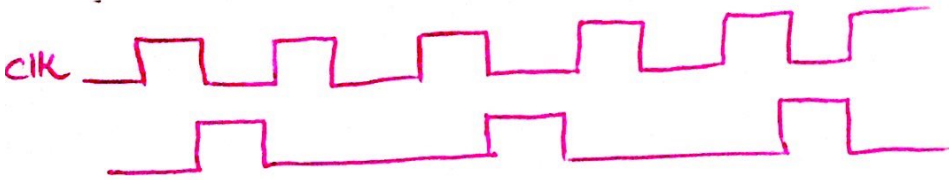


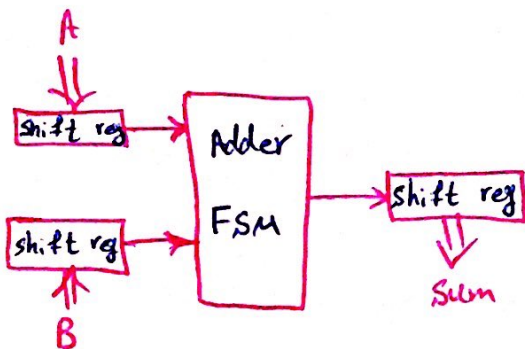
کدیز:

مداری طراحی کنید تا سیگنال ورودی X را از سیگنال کلاک ورودی طبق دیگرام زیر تولید کند.



- پیاده سازی جمع کننده سری Serial Adder :

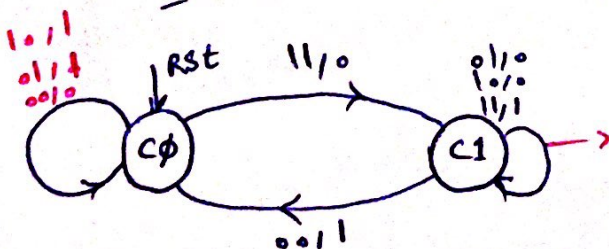
- دو عدد بدون علامت $A = a_{n-1} \dots a_0$ و $B = b_{n-1} \dots b_0$ را در نظر گرفته با فرض اینکه جمع آنها $S = s_{n-1} \dots s_0$ باشد. می خواهیم مداری طراحی کنیم که جمع را به صورت سری انجام دهد.



Mealy مدل
Moore مدل

Mealy مدل :

- در state در نظر بگیریم $c\phi$: carry-in (از مرحله قبل می آید).
 $c1$: ...

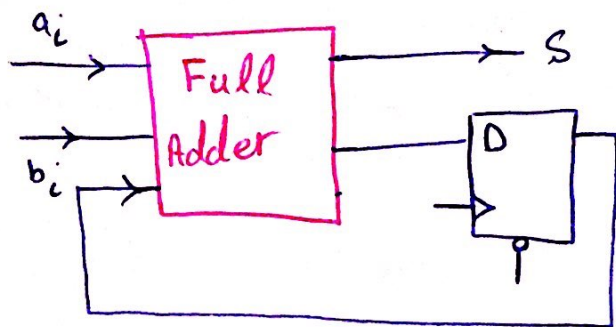


تیمین: کامل شود!
* کامل شد!

c_s	$NS(Y)$	S
y	$a \ b \ c$	
0	0 0 0 1	0 1 1 0
1	0 1 1 1	1 0 0 1

$$Y = ab + ay + by$$

$$S = a \oplus b \oplus y$$



مدار میانه

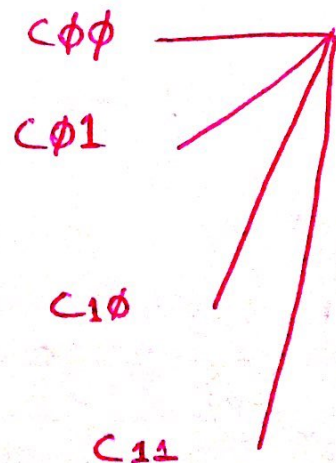
← آیا می توان این مدار را بصورت Moore ساده تر کرد؟
 که خودی تنها به c_s وابسته است

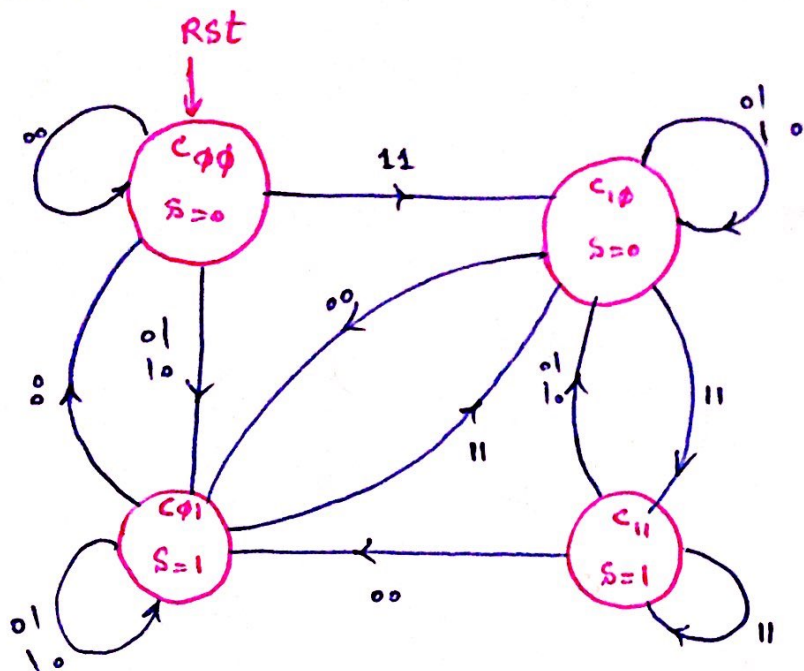
$$C = \phi, S = \phi$$

$$C = \phi, S = 1$$

$$C = 1, S = \phi$$

$$C = 1, S = 1$$





* در مدل Moore مجبوریم

تعداد states را زیاد کنیم

* کامل شد

(اینی می خواستم بگم کامل کنید خودتان)

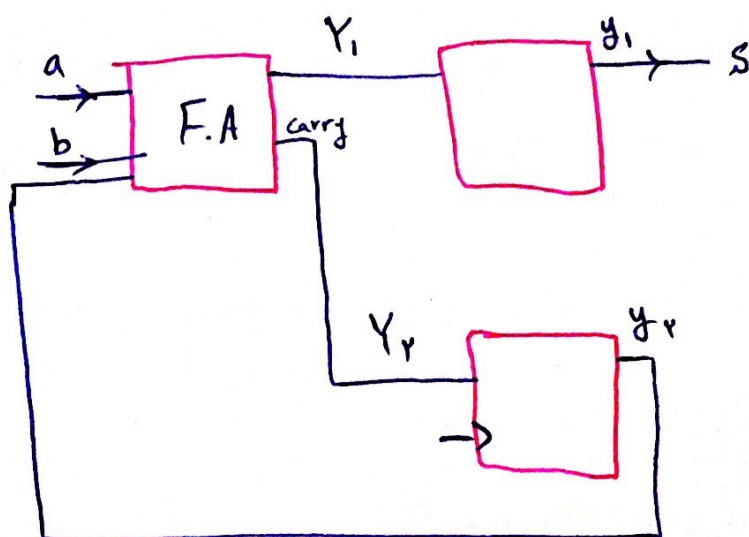
کامل کردن! :-D

قبلیم مهندسورا

$y_2 y_1$	$NS(Y_2 Y_1)$				output
	00	01	10	11	S
00	00	01	01	10	0
01	00	01	10	10	1
10	01	10	10	11	0
11	01	10	10	11	1

(submit) $Y_1 = a \oplus b \oplus y_2$

(carry) $Y_2 = ab + ay_2 + by_2$

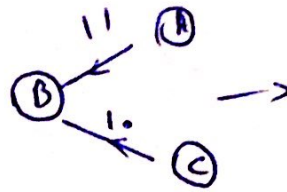


تبدیل مدارهای منطقی به مور: ← خروجی در دو حالت صفر و یک در هر دو حالت های A و B وجود دارد.

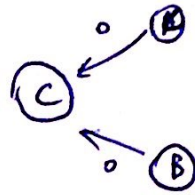
== برای تبدیل این مدار به Moore باید به ازای هر یک از حالت های A و B دو حالت در نظر بگیریم.

*** مهم *** اگر به ازای سیرهای ورودی به یک state (مثلاً A) مجموعه خروجی یک Z دریافت کنیم می توان برای C در مدار Moore نهایت state را گرفت.

CS	NS		Z	
	0	1	0	1
A	A	B	0	1
B	A	C	1	0
C	C	B	0	0



این دو جواب مختلف
صفحه یک یورد



CS	NS		Z
	0	1	
A_0	A_0	B_1	0
A_1	A_0	B_1	1
B_0	A_0	C	0
B_1	A_0	C	1
C	C	B_0	0

B ای که خروجی را یک کند
 (در جدول بالا در سطرها)
 به ازای $A=B$ خروجی هم یک
 شده است
 A ای که خروجی اش صفر است
 B ای که خروجی را صفر کنند

تجزیه: مدار زیر را به صورت تبدیل کنید.

CS	NS		Z	
	$w=0$	$w=1$	$w=0$	$w=1$
A	D	A	00	00
B	A	D	10	01
C	D	C	10	00
D	B	C	00	00

$A \rightarrow A_0$
 $A \rightarrow A_1$

$B \rightarrow B$
 $C \rightarrow C$

$D \rightarrow D_0$
 $D \rightarrow D_1$
 $D \rightarrow D_1$

کاهش تعداد states :

قبل گفتیم کاهش تعداد state منجر به
 FF کاهش
 comb logic

لذا در حالت کلی state reduction مطلوب است.

چگونه کم کردن تعداد states را کردیم، را تغییر می دهد؟ خیر

سوال کم کردن حالت :

CS	NS		Z	
	0	1	0	1
A	B	C	1	1
B	C	D	0	1
C	B	D	0	1
D	B	C	1	1

\Rightarrow A, D
 یکسان
 به نظر می آید
 حذف می کنیم

	NS		Z	
	0	1	0	1
A	B	C	1	1
B	C	A	0	1
C	B	A	0	1

ب, c هم شبیه هستند پس کلاً در state داریم

روشهای کم کردن حالت :

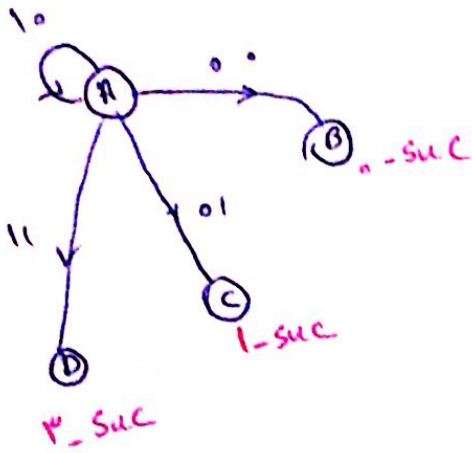
(1) Partitioning

(2) Implication graph

قبل از پرداختن به این روشها چند تعریف :

1 - successor , 0 - successor (1

(k - successor)



0 1 ... 1ⁿ - 1

- در یک FSM، حالت s_i را در نظر بگیرید.

s_w یا 0-suc حالت s_i است که اگر به ازای ورودی

$w = 0$ از s_i به s_w برویم.