Subject: 1

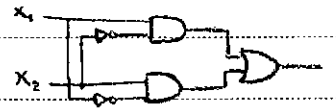
Year. Month. Date. (3)

* تعداد ورودی‌ها و خروجی‌های هر gate به علت اینکه از ترانزیستور ساخته شده اند دارای محدودیت‌هایی به نام fan in و fan out خواهد بود.

* AND Gate همواره کمترین ورودی را از خود عبور می‌دهد، چون در تکنولوژی CMOS، AND از زمین استفاده می‌کند. OR Gate همواره بیشترین ورودی را از خود عبور می‌دهد. NAND مثل کمترین ورودی را از خود عبور می‌دهد و NOR مثل بیشترین ورودی را از خود عبور می‌دهد. XOR به ازای تعداد ورودی‌های فرد، «یک» می‌دهد و XNOR به ازای تعداد ورودی‌های زوج، «یک» را نتیجه خواهد داد.

* تحلیل مدار یعنی پیدا کردن عملکرد مدار و ساختن مدار منطقی ساده (Forward ↔ Reverse Engineering)

Reverse Engineering ☑



* مدارهای ترکیبی از معادلات کانونی که به دو صورت POS و SOP هستند. برای ساده کردن آنها از دو روش Karno-Map و Quine-McClusky استفاده می‌شود.

* Negative logic : در مدارات CMOS حامل‌ها دارای حرکت می‌کنیم و مدارات Bipolar حامل‌ها حرکت می‌کنند. الکترون‌ها با سرعت ۳ برابر نسبت به حفره‌ها حرکت می‌کنند. الکترون حامل بار منفی هستند و حرکت آنها تولید حرارت می‌کند که خود، مانع سرعت است. برای جلوگیری از تولید حرارت، بجای استفاده از دیتاهای غیر، عدد منفی دیتا را نصب را بعنوان دیتا در نظر می‌گیریم.

* implicant ترکیبی است که حداقل دارای یک نقطه «یک» از جدول درستی تابع است.

* prime implicant ترکیبی است که تعدادی از تابع را پوشاند که تنها یک تابع می‌تواند آنها را بپوشاند.

* اگر مدار ⊕ باشد NOR بیشتر به درد می‌خورد، اگر مدار ⊖ باشد NAND بیشتر به درد می‌خورد.

* برای مدارات ترکیبی به True Table، Karno-Map احتیاج داریم. برای مدارهای ترکیبی از Moore، Mealy استفاده می‌کنیم.

Subject:

Year. Month. Date. ( )

☑ ماشین Mealy زیر را به Moore تبدیل کنید.

	0	1
q_1	$q_2, 0$	$q_3, 1$
q_2	$q_1, 0$	$q_2, 0$
q_3	$q_3, 1$	$q_1, 1$

⟹

	0	1	out
q_{10}	q_2	q_3	0
q_{11}	q_2	q_3	1
q_{20}	q_1	q_2	0
q_{30}	q_3	q_1	1

☑ ماشین Moore زیر را به Mealy تبدیل کنید.

	0	1	
q_1	q_2	q_3	0
q_2	q_1	q_4	1
q_3	q_2	q_2	0
q_4	q_1	q_4	1

⟹

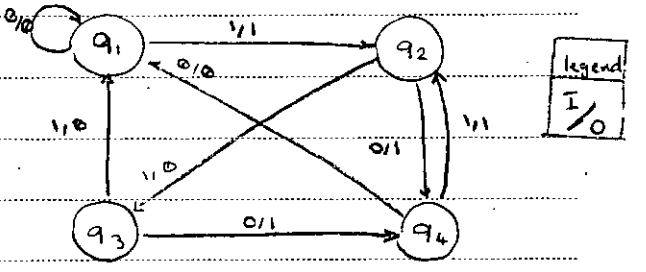
	0	1
q_1'	$q_2', 1$	$q_3', 0$
q_2'	...	...
q_3'		
q_4'		

	0	1
q_1	$q_1, 0$	$q_2, 1$
q_2	$q_4, 1$	$q_3, 0$
q_3	$q_1, 0$	$q_4, 1$
q_4	$q_2, 1$	$q_1, 0$

☑ خروجی جدول زیر پس از دریافت X = 100111 چیست؟

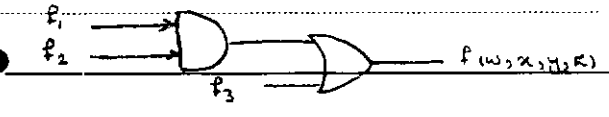
input	X	1	0	0	1	1
state	IS	q_1	q_2	q_4	q_2	q_3	q_4
output	Z	1	1	1	0	1	

☑ state diagram مربوطه را رسم کنید.

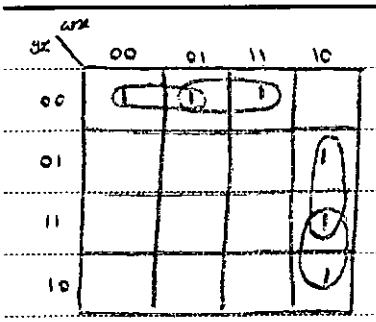


Given the following circuit determine the functions f_2 & f_3 if $f_1 = x\bar{z} + \bar{x}z$ ☑

and the overall transmission function is to be $f(w,x,y,z) = \sum(0,4,9,10,11,12)$

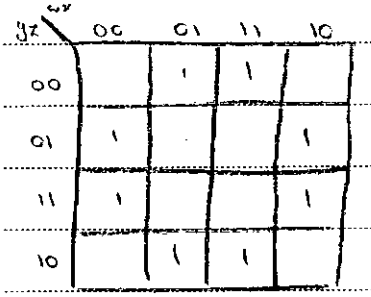


P4PCO



$\Rightarrow f = \bar{w}\bar{y}\bar{z} + x\bar{y}\bar{z} + w\bar{x}z + ywz = \sum(0,4,9,10,11,12)$

$f = f_1 \cdot f_2 + f_3 = ((x\bar{z} + \bar{x}z)f_2) + f_3$

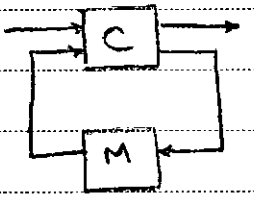


$\Rightarrow f_1 = \sum(1,3,4,6,9,11,12,14)$

* Don't Care = بی‌تفاوت نامیده می‌شود و برابر می‌گیرد:
- فرقی نمی‌کند.
- مقدار ممنوعه است.
- ممنوعه نیست ولی اتفاق نمی‌افتد.

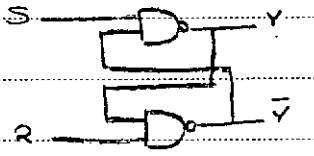
* Memory Elements :

S-R flip flop .

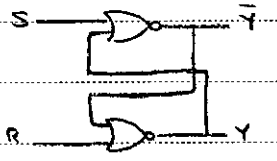


q \ SR	00	01	11	10
0	0	0	—	1
1	1	0	—	1

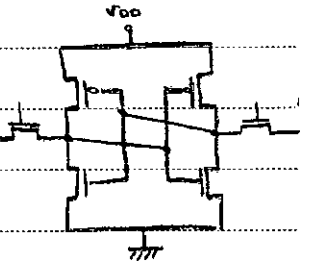
NAND



NOR



VLSI



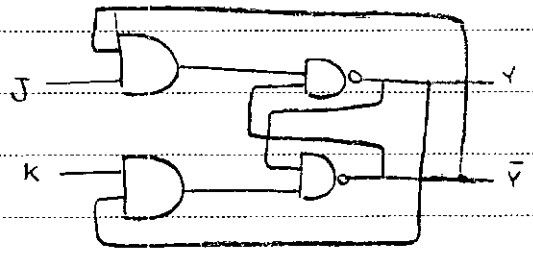
$Y = S + \bar{R}y$

P.S	N.S	S	R
0	0	0	—
0	1	1	0
1	0	0	1
1	1	—	0

Subject:

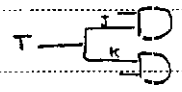
Year. Month. Date. ( )

J-K flip flop •



JK / q	00	01	11	10
0	c	c	1	1
1	1	0	c	c

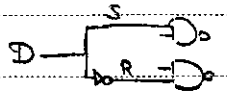
T flip flop •



y / T	0	1
0	0	1
1	1	0

$Y = \bar{T}y + T\bar{y}$

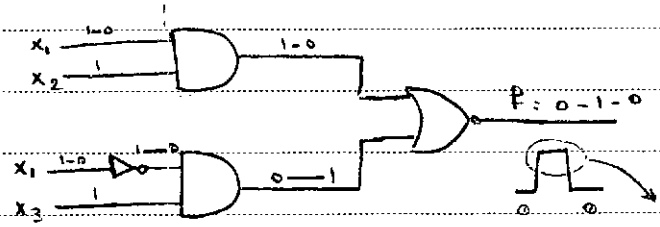
D flip flop •



D / q	0	1
0	0	1
1	0	1

$Y = D$

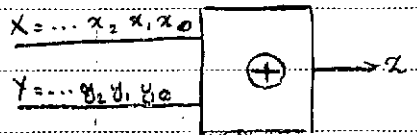
static zero hazard ☑



f — T — y: 1-0-0 x

f — D — T — y: ✓

serial adder ☑

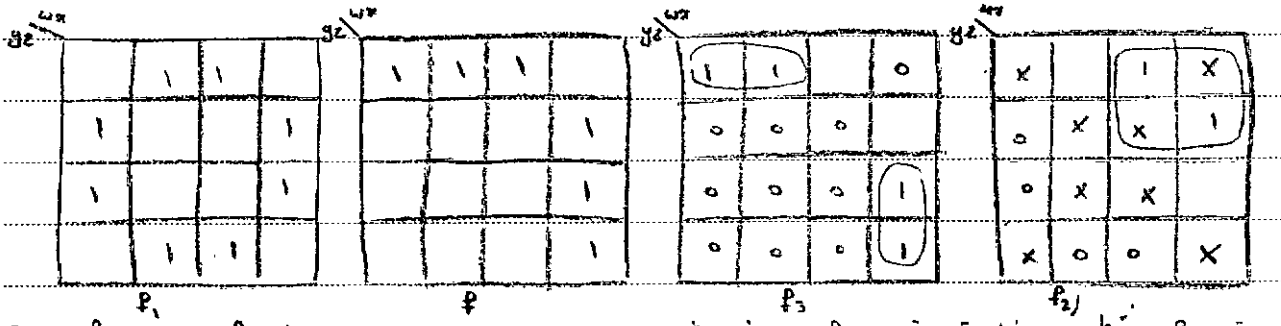


c: / xy	00	01	11	10
q_0	$q_{0,0}$	$q_{0,1}$	$q_{1,0}$	$q_{0,1}$
q_1	$q_{0,1}$	$q_{1,0}$	$q_{1,1}$	$q_{1,0}$

Subject: 3

Year. 84 Month. 7 Date. 17 (5)

حل ☑ $f_1 = x\bar{z} + \bar{x}z$, $f(w,x,y,z) = \sum(0,4,9,10,11,12)$



تابع f در نقاطی که ضرایب تابع f_3 یک هستند، یک است.

$f_1 = 1$, $f_3 = 0 \Rightarrow f_3 = 1$

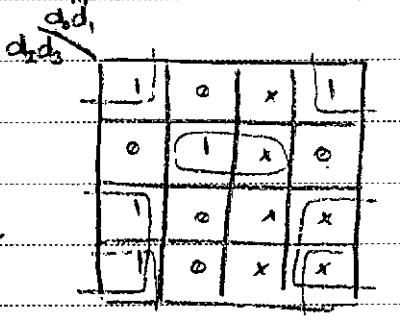
$f = 0$, $f_1 = 0 \Rightarrow f_1 = x$

$f = 1$, $f_3 = 1 \Rightarrow f_2 = x$

$f_2 = \bar{y}w$

$f_3 = \bar{w}\bar{y}\bar{z} + w\bar{x}y$ } minimal

A BCD message appears in 4 input lines of a switching circuit. Design ☑ a network which produces an output 1 whenever the input combination is 0, 1, 3, 5 or 8.



$f = \bar{d}_1\bar{d}_3 + d_1\bar{d}_2d_3 + \bar{d}_1d_2$

چون BCD از 0 تا 9 دارد پاسخی هم نیست.

☑ با استفاده از Quine - McClusky تابع Prime implicant را تابع زیر بیابید.

a) $f(w,x,y,z) = \sum(1,5,6,12,13,14) + \sum\Phi(2,4)$

b) $f(w,x,y,z) = \sum(0,1,4,5,6,7,9,11,15) + \sum\Phi(10,14)$

Solution a)

1	0 0 0 1 ✓	1,5	0 − 0 1 Ⓐ	4,5,12,13 − 1 0 − Ⓒ
2	0 0 1 0 ✓	2,6	0 − 1 0 Ⓑ	4,6,12,14 − 1 − 0 Ⓓ
4	0 1 0 0 ✓	4,5	0 1 0 − ✓	
5	0 1 0 1 ✓	4,6	0 1 − 0 ✓	
6	0 1 1 0 ✓	4,12	− 1 0 0 ✓	
12	1 1 0 0 ✓	5,13	− 1 0 1 ✓	
13	1 1 0 1 ✓	6,14	− 1 1 0 ✓	
14	1 1 1 0 ✓	12,14	1 1 − 0 ✓	
		12,13	1 1 0 − ✓	

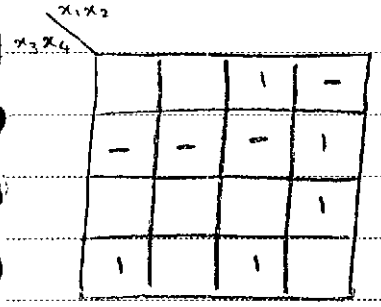
	1	5	6	12	13	14
A	(x)	x				
B			x			
C		x		x	(x)	
D			x	x		(x)

$f = \bar{w}\bar{y}z + x\bar{y} + x\bar{z}$

PAPCO

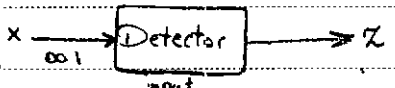
Subject:

Year. Month. Date. ( )



Prime implicant ☑ ی جدول کارنوی زیر را بدست آورید. سپس آنرا Quine - Mc Clusky تبدیل کنید.

sequence detector ← 001 ☑



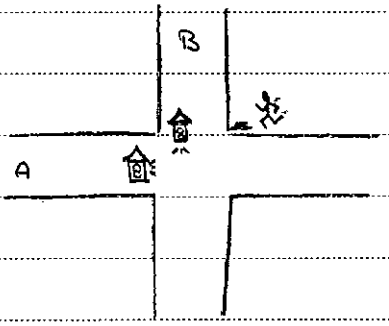
q_0 : آخرین ورودی صفر بوده است.

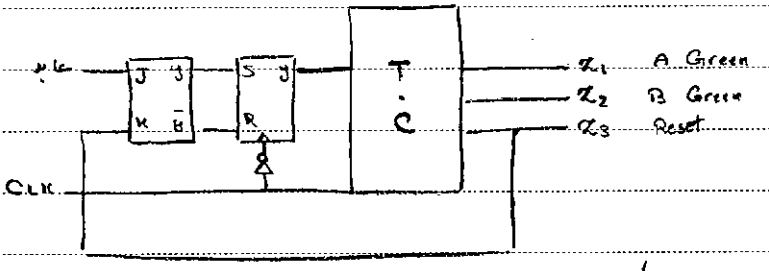
q_1 : آخرین ورودی یک بوده است.

q_2 : آخرین دو ورودی صفر بوده است.

state \ input	0	1
q_0	$q_{2},0$	$q_{1},0$
q_1	$q_{0},0$	$q_{1},0$
q_2	$q_{2},0$	$q_{1},1$

pedestrian traffic Controller ☑





چراغ هر طرف یک دقیقه سبز است. دکمه عابر کلید را فشار داده، پس از آن دوره چراغ سبز ۲ دقیقه تا چراغ قرمز شدن تداوم میکند. سپس چراغ سبز به طرفی نوبتش رسیده است داده می شود. مدار Traffic Controller را با T-flip flop طراحی کنید.

q_0 : سبز A قرمز B

q_1 : قرمز A سبز B

q_2 : هر دو قرمز، A سبز بوده

q_3 : هر دو قرمز، B سبز بوده

	عابر 0	عابر 1
q_0	$q_1,010$	$q_2,000$
q_1	$q_0,100$	$q_3,000$
q_2	$q_1,010$	$q_2,001$
q_3	$q_0,100$	$q_3,001$

خروجی Reset برای اعلام پایان سرویس دادن به عابر است یعنی پایان ۲ دقیقه را اعلام می کند.

Subject: 4

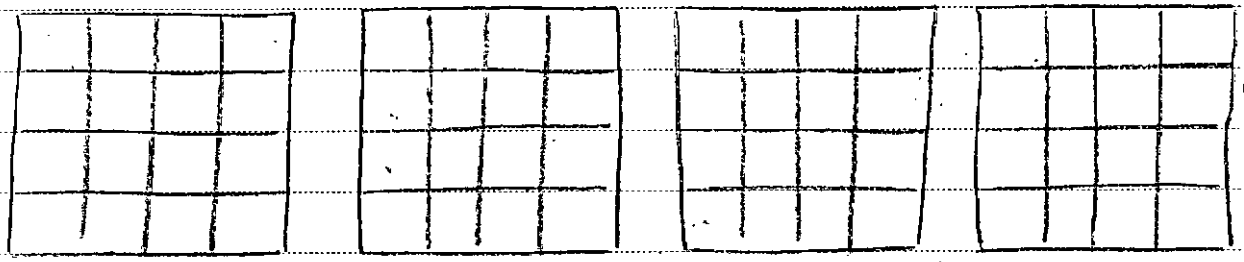
Year. 84 Month. 7 Date. 19 (6)

☑ جدول دو بردار با T Flip Flop پیاده‌سازی کنید

	X = 0	X = 1
q_1	$q_2, 0$	$q_2, 0$
q_2	$q_4, 1$	$q_5, 0$
q_3	$q_5, 0$	$q_2, 1$
q_4	$q_4, 1$	$q_3, 0$
q_5	$q_1, 1$	$q_4, 0$

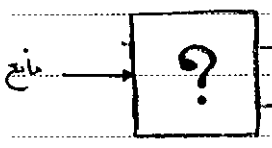
در اینجا احتیاج به ۳ تا T Flip Flop داریم. با این ۳ تا ۸ حالت مختلف تولید می‌شود. در مرحله state assignment، با کد gray code برای ۵ state ی مختلف کدگذاری می‌کنیم. مدار حاصل به دلیل تنها یک بیت تفاوت، hazard کمتری خواهد داشت. جدول حاصل «YZ matrix» نام دارد.

$y_3 y_2 y_1$ Gray Code	Present State	i	Next State	T_3	T_2	T_1	Z
000	q_1 : 0 0 0	0	0 0 1	0	0	1	0
		1	0 1 1	0	1	1	0
001	q_2 : 0 0 1	0	0 1 0	0	1	1	1
011		1	1 1 0	1	1	1	0
010	q_3 : 0 1 1	0	1 1 0	1	0	1	0
110		1	0 0 1	0	1	0	1
111	q_4 : 0 1 0	0	0 1 0	0	0	0	1
101		1	0 1 1	0	0	1	0
100	q_5 : 1 1 0	0	0 0 0	1	1	0	1
		1	0 1 0	1	0	0	0



☑ یک روبات داریم. اگر جلوی روبات مانع باشد و روبات دفعهٔ قبل به چپ چرخیده باشد آنگاه به راست می‌چرخد و مانع را رد کند. اگر روبات دفعهٔ قبل به راست چرخیده باشد آنگاه به چپ می‌چرخد تا مانع را رد کند.

مانع : X راست : Z_2 چپ : Z_1



		X = 0	X = 1
q_0		$q_0, 00$	$q_1, 01$
q_1	مانع نیست و چرخش به چپ بوده: q_0	$q_2, 00$	$q_4, 01$
q_2	مانع نیست و باید به راست بچرخد: q_1	$q_2, 00$	$q_3, 10$
q_3	مانع نیست و چرخش به راست بوده: q_2	$q_0, 00$	$q_3, 10$
q_4	مانع هست و باید به چپ بچرخد: q_3		

P4PCO

Subject:

Year. 84 Month. 7 Date. 24 (7)

serial adder ☑ طراحی کنید D ff با، T ff با، با

P.S	Inputs	N.S	out	D	T
0	0 0	0	0	0	0
	0 1	0	1	0	0
	1 0	0	1	0	0
	1 1	1	0	1	1
1	0 0	0	1	0	1
	0 1	1	0	1	0
	1 0	1	0	1	0
	1 1	1	1	1	0

y \ x_1x_2	00	01	11	10
0	0	0	1	0
1	0	1	1	1

c \ x_1x_2	00	01	11	10
0	0	0	1	0
1	1	0	0	0

$D = x_1x_2 + x_1y + x_2y$ $T = x_1x_2\bar{y} + \bar{x}_1\bar{x}_2y$

$z = x_1 \oplus x_2 \oplus y$

J-K ff $J = x_1x_2$ $K = \bar{x}_1\bar{x}_2$

پس J-K ff به بقیه ترجیح دارد.

state assignment ☑ تمایز انواع

	0	X	1
q_1	q_3		q_2
q_2	q_4		q_1
q_3	q_1		q_3
q_4	q_2		q_3

	A: y_1 y_2	B: y_1 y_2
q_1	0 0	0 0
q_2	0 1	1 1
q_3	1 0	1 0
q_4	1 1	0 1

مدار حاصل با D ff و T ff

بیازید و تمایز کنید

A:

P.S	I	N.S	D	T
00	0	10	10	10
	1	01	01	01
01	0	11	11	10
	1	00	00	01
10	0	00	00	10
	1	10	00	00
11	0	01	01	10
	1	10	10	01

$D_1 = \bar{I}\bar{x}_1 + Ix_2$ $D_2 = \bar{I}x_2 + \bar{I}x_1 + I\bar{x}_1\bar{x}_2$

$T_1 = \bar{I}$ $T_2 = I\bar{x}_1 + \bar{I}x_2$

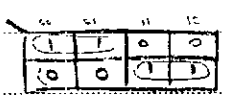
B:

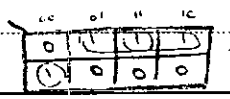
P.S	I	N.S	D	T
00	0	10	10	10
	1	11	11	11
11	0	01	01	10
	1	00	00	11
10	0	00	00	10
	1	10	10	00
01	0	11	11	10
	1	10	10	01

$D_1 = \bar{x}_1 + I\bar{x}_1\bar{x}_2$ $D_2 = \bar{I}x_2 + I\bar{x}_1\bar{x}_2$

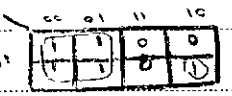
$T_1 = \bar{I} + I\bar{x}_1 + \bar{I}x_2$ $T_2 = \bar{I}\bar{x}_1 + Ix_2$

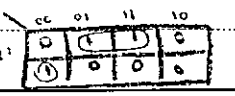
A





B





PAPCO

Subject: 5

Year. Month. Date. ( )

در روش لچ‌زنی A شبیه‌ی شود که مدار حاصل از TFF از مدار شبیه به لچ‌زنی B کوچکتر است، زیرا تغییرات بیشتری بین لچ حالتهای بیشتر به هم تبدیل می‌شوند وجود دارد تغییر بیشتر، تعداد ۱ بیشتری در جدول کارنوی TFF ایجاد می‌کند و ساده‌سازی را مشکل‌تر می‌سازد. در مثالیه، DFF لچ‌زنی A، طراحی TFF ترجیح دارد.

در روش لچ‌زنی B دیده می‌شود که با توجه به اشتراک دو minterm، T_2 و T_1، این طراحی از DFF کوچکتر باشد. در مثالیه طراحی DFF این روش در روش لچ‌زنی A به ترجیح نسبی روش B پی می‌بریم: چون اصولاً در لچ‌زنی B حالتهای که پرکاربردتر، تعداد ۱های بیشتری لچ‌زنی شده‌اند و در جدول کارنوی DFF ۱های بیشتری را ایجاد نمی‌کنند.

$$TFF_A > TFF_B > DFF_B > DFF_A$$

* چه جداولی را می‌توان minimize کرد؟

Completely Specified Tables ← don't care ندارند

*

State «q'» of table «M'» covers state «q» of table «M», if the output sequence of «M'» in initial state «q'» and «M» in initial state «q» are identical for any input sequence.

*

Table «M'» covers (is equivalent to) table «M» if every state of «M» is covered by some state of «M'».

*

If «M» is covered by «M'» which has less number of states than «M» then some states of «M» are covered by one state of «M'» which these states are called equivalent.

	0	1
q_1	$q_3, 0$	$q_4, 1$
q_2	$q_3, 1$	$q_2, 0$
q_3	$q_4, 1$	$q_1, 0$
q_4	$q_1, 1$	$q_3, 0$

q_2	X X		
q_3	X X	q_3, q_4 ✗ q_1, q_2 ✗	
q_4	X X	q_1, q_3 ✗ q_2, q_3	q_1, q_4 ✗ q_1, q_3
	q_1	q_2	q_3

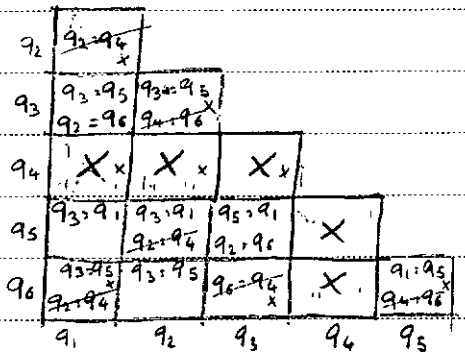
P4PCO

Subject:

Year. Month. Date. ( )

M

	0	1
q_1	$q_3, 0$	$q_2, 0$
q_2	$q_3, 0$	$q_4, 0$
q_3	$q_5, 0$	$q_6, 0$
q_4	$q_5, 1$	$q_1, 1$
q_5	$q_1, 0$	$q_2, 0$
q_6	$q_5, 0$	$q_4, 0$



$\Rightarrow q_1 \sim q_3$

$q_1 \sim q_5$

$q_2 \sim q_6$

$q_3 \sim q_5$

$\Rightarrow q_1 \sim q_3 \sim q_5 \,,\, q_2 \sim q_6$

* جواب ساده سازی باید state باشد، یعنی تمام covered باید در آن حضور داشته باشند.

$q_1 \sim q_3 \sim q_5 = A$

$q_2 \sim q_6 = B$

$q_4 = C$

M'

	0	1
A	A, 0	B, 0
B	A, 0	C, 0
C	A, 1	A, 1

* جواب ساده سازی باید closed باشد یعنی تمام Next state باید در یک دسته (مثلاً A) قرار بگیرد.

* جدول M' از نسبت به M سینگل باشد.

* Lemma 1 :

For a completely specified table «M», if «$q_i \sim q_k$» and «$q_k \sim q_j$» $\Longrightarrow$ «$q_i \sim q_k$»

proof :

Assume $q_i \not\sim q_j$

$\exists I$ such that $z_i \neq z_j$ but

$q_i \sim q_k$, for input I, $z_i = z_k$

$q_j \sim q_k$, for input I, $z_j = z_k$ $\Rightarrow z_i = z_k$

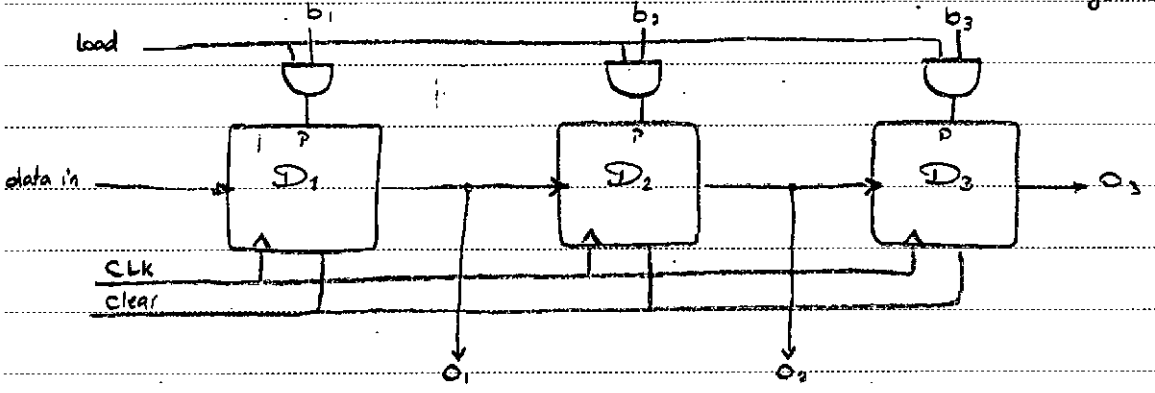
* Lemma 2 :

For a completely specified table in «M», if the set of states $q_1, q_2, \ldots, q_p$ are all equivalent to q_k that the set of states $\varphi = \{q_1, q_2, \ldots, q_p, q_k\}$ are equivalent.

Proof :

If the set φ is not equivalent there exist two states $q_i \not\sim q_j$ such that $q_i \not\sim q_j$ but $q_i \sim q_k$, $q_j \sim q_k \Rightarrow q_i \sim q_j$ X

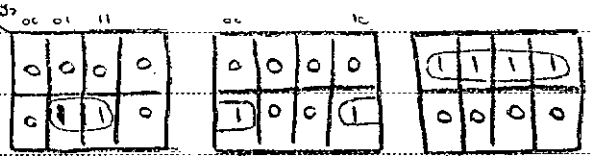
D ff ، shift reg. یک



* سه تابع احتیاج به طراحی ندارند : ring counter ، shift reg. ، reg.

یک counter ۸ تایی طراحی کنید، با T ff.

	P.S	N.S	T
q_0	0 0 0	0 0 1	0 0 1
q_1	0 0 1	0 1 1	0 1 0
q_2	0 1 1	0 1 0	0 0 1
q_3	0 1 0	1 1 0	1 0 0
q_4	1 1 0	1 1 1	0 0 1
q_5	1 1 1	1 0 1	0 1 0
q_6	1 0 1	1 0 0	0 0 1
q_7	1 0 0	0 0 0	1 0 0



$T_1 = y_2 y_3$ $T_2 = y_3 \bar{y}_2$ $T_3 = \bar{y}_3$

یک sequence generator برای دنباله دودویی 0,7,4,6,5,2,1,3 طراحی کنید.

یک دودویی در نظر می‌گیریم، با روشن شدن آن sequence پیش می‌رود.

Subject:

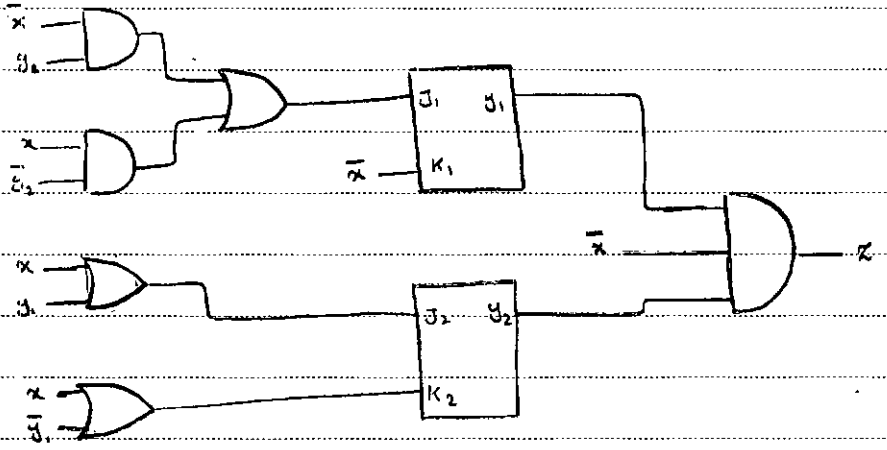
Year. 84 Month. 8 Date. 8 (8)

	⊗ x	y	gray code
q_0	q_0, 000	q_7, 111	0 0 0
q_7	q_7, 111	q_4, 100	0 0 1
q_4	q_4, 100	q_6, 110	0 1 1
q_6	q_6, 110	q_5, 101	0 1 0
q_5	q_5, 101	q_2, 010	1 1 0
q_2	q_2, 010	q_1, 001	1 1 1
q_1	q_1, 001	q_3, 011	1 0 1
q_3	q_3, 011	q_0, 000	1 0 0

* برای مدل بندی مکانیکی تعداد state زیاد شود (بیش از ۷ تا) ، مدارهای sequential طراحی خوبی نمی آیند

Analisys Of State Machine

وقتی است که مداری داریم و می خواهیم ببینیم چکار می کند. آن Reverse Engineering نیز می گویند



سراغ ورودی های FF و خروجی می رویم:

$Z = y_1 y_2 \bar{x}$

$J_1 = \bar{x} y_2 + x \bar{y}_2 = x \oplus y_2$ $\quad J_2 = x + y_1$

$K_1 = \bar{x}$ $\quad K_2 = x + \bar{y}_1$

$y_1 = J_1 \bar{y}_1 + \bar{K}_1 y_1$ $\quad y_2 = J_2 \bar{y}_2 + \bar{K}_x y_2$

Subject: 7

Year. 84 Month. 8 Date. 10 (9)

P.S	x=0 J_1	K_1	J_2	K_2	x=1 J_1	K_1	J_2	K_2
00	0	1	0	1	1	0	1	1
01	1	1	0	1	0	0	1	1
11	1	1	1	0	1	0	1	1
10	0	1	1	0	0	0	1	1

$\Rightarrow$

P.S	x=0	x=1
A 00	00 → A, 0	C, 0
B 01	D, 0	A, 0
C 11	B, 1	D, 0
D 10	B, 0	C, 0

* مدارات آسنکرون

در آن clock نداریم. بزرگترین مشکل در این مدارها race و hazard است (مخصوصاً در TFF) و در آن به هر کدام از این مشکل state خاص می‌شود.

معایب clock .

clock skew .

more gate required .

more wire required .

unbalance in distribution .

large size & low speed .

دقت در مداری در قسمتهایی نیاز به clock دارند دوره‌ی خاص خاتمه ندارند - آنها GAL می‌نویسند و چنین مداری را پیاده‌سازی می‌کنند.

* مدارهای آسنکرون

Pulse Type I، ورودی در زمان مشخص و با پهنای مشخص

Level Type II، پهنای مشخصی ندارد.

* مدارات آسنکرون sequential هستند پس جدول احتیاج داریم، برای هر ورودی یک ستون داریم حتی وقتی ورودی وجود نداشته باشد مدار تغییر می‌کند (حتی clock نداشته باشد). در این جدول flow table گرینه‌ایم Present State, Next State داشته باشد. چون clock نداریم باید در هر سطر یک حالت stable داشته باشد. به علت وجود مشکل پیاده‌ای، در هر تغییر تنها یک بیت می‌تواند تغییر کند

Subject:
Year. 84 Month. 8 Date. 15 (10)

به بیشتر (چون فعال شدن در یا چند ورودی به بیت بستگان clock برای synchronize کردن خلن است بدار
در حالات نشده بود. سایل با ورودی‌های pulse type تقریباً مانند داده‌ی شدن است.

* در Flow Table :
- در حالات پایدار دایره‌ی کشیم.
- در حالت تغییر بیش از یک ورودی، حالت یا don't care است یا خودش است.
- پیچیدگی نمی‌دان don't care با آدرس داد.

	00	01	11	10
1	①, 0	3, 1	①, 0	4, 0
2	②, 0	4, 1	—	4, 0
3	③, 1	③, 1	1, 0	4, 0
4	3, 1	④, 0	1, 0	④, 0

* برای حل مسائل آشکارون
- هر سطر باید یک حالت stable داشته باشد.
- هر transition باید از حالت پایدار شروع شده و به یک حالت پایدار خاتمه یابد.
- در هر transition نمی‌دان یک خانه don't care آدرس داد.
- در هر transition فقط یک ورودی تغییر می‌یابد.

* جداول Foundamental : به جدولی گویند از هر حالت پایدار شروع کنی به حالت پایدار ختم شود.

* جداول Foundamental ← (SOC) Single Output Change
(MOC) Multiple Output Change

Subject: 8

Year. Month. Date. ( )

معادل آن تنها یکی از عناصر MOC باشد، جدول MOC است. جدول SOC تمام عناصر SCC است.

* جدول زمان

① SOC باشد. (تمام عناصر SCC است)

② Fundamental خالص باشد (برای تمام عناصر باید Fundamental باشد)

③ هر عنصر تنها با یک پرش به عنصر بعدی برسد.

مثال: سیستمی با دو ورودی x_1 و x_2 و خروجی z داریم که:

$x_1 = 0 \Longrightarrow z = 0$

$x_1 = 1$ & x_2 changes $\Longrightarrow z = 1$ $\quad x_1\,0 \longrightarrow x_1\,0$

	00	01	11	10
1	①,0	2,0	—	3,0
2	1	②,0	4	—
3	1	—	5	③,0
4	—	2	④,0	6
5	—	2	⑤,1	6
6	1	—	5	⑥,1

برای هر state یا خانه در تغییر آزاد don't care می‌گیریم.
برای هر حالت ورودی خاص باید یک حالت stable وجود داشته باشد. در هر سطر یک حالت stable داریم.
این جدول Primitive Flow Table نامیده می‌شود.

مثال: Mod 4 Counter ← دو ورودی x_1, x_2 و خروجی z_1, z_2

00 → clear , 01 → 1 count , 10 → 2 count , 11 → no change

تعداد ستون‌ها برابر 4 است چون 4 ورودی فرض داریم که خود این 4 ورودی از ترکیب دو ورودی ایجاد می‌شود (x_2, x_1)

(+2) 10	3	1	③,2	7	9	1	⑦,3	1	⑨,0	13	3	1	⑬,1
N/A 11	1	4	5	④,1	⑤,2	5	10	10	11	⑩,3	⑪,0	11	4
(+1) 01	2	②,1	1	6	8	⑥,2	1	⑧,3	1	12	2	⑫,0	1
reset 00	0,0	1	1	1	1	1	1	1	1	1	1	1	1
	1	2	3	4	5	6	7	8	9	10	11	12	13

Subject:

Year. 84 Month. 8 Date. 22(11)

☑ یک Black Box استاندارد با ورودی $x_1 - x_2 - x_3$ از نوع pulse-type دارای خروجی z است که در صورت اعمال ورودی صحیح، $z=1$ می‌شود. مدار مربوط را بسازید.

$y_1\ y_2$	x_1	x_2
0 0	0	0
0 1	0	1
1 1	0	0
1 0	—	—

	x_1	x_2
00	00,0	01,0
01	00,0	11,0
11	00,0	11,0
10	—	—

	x_1	x_2
$x_1 \leftarrow q_1$	$q_1,0$	$q_2,0$
$x_1 - x_2 \leftarrow q_2$	$q_1,0$	$q_3,1$
$x_2 \leftarrow q_3$	$q_1,0$	$q_3,0$

$z = \bar{y}_1 y_2 x_2$

validation یعنی تحلیل مدار با حالتهای مختلف ورودی و ثابت بودن صحت عمل مدار به ازای این حالتهای ورودی.

$y_1\ y_2$	x_1	x_2	$Y_1\ Y_2$	T_1	T_2
0 0	0	1	00	0	0
0 1	1	0	01	0	1
1 1	1	0	11	1	0
1 0	—	—	10	—	—

x_1: 0 - 1 - 0 - 0

x_2: 0 - 0 - 1 - 1 $\Rightarrow$ z: 0 - 0 - 0 - 1 ✓

x_1: 0 - 0

x_2: 0 - 1 $\Rightarrow$ z: 0 - 0 ✓

$T_2 = y_2 x_1 + \bar{y}_2 x_2$ $\quad T_1 = x_1 y_1 + \bar{y}_1 y_2 x_2$

Subject: 9

Year. 84 Month. 9 Date. 6 (11)

* A set of compatible sets $\dot{C}=\{C_1, C_2, \dots\}$ is « Closed » : if every $C_i \in C$, all the implied sets (C_{ij}) are contained in some elements of C for all I_j.

* Let C_i = compatible set of states

C_{ij} : set of next states implied by C_i for input I_j

Class Set P_i implied by C_i is the set of all C_{ij} such that :

1. C_{ij} has more than one state or null
2. $C_{ij} \not\subset C_i$
3. $C_{ij} \not\subset C_{ik}$ if $C_{ik} \in P_i$

☑

	I_1	I_2	I_3
1	3, ⓪	—	2, —
2	—	4, ⓪	6, —
3	5, 1	—	—, ⓪
4	—	1, 1	—
5	1, —	—	6, —
6	4, —	5, —	6, —

mc's : {126, 14, 236, 34, 45, 256}

ظاهرا با این mc ها می‌توان به حداقل سه تا state داریم.

C_i	P_i
126	34, 45, ~~26~~ → C_i نباید باشد
12	26
256	14, 45
~~25~~	ϕ
~~56~~	14
1	ϕ
2	ϕ
3	ϕ
4	ϕ
5	ϕ
6	ϕ

~~26~~ — 45 — dominate نباید

16 — 34, 26

~~14~~ — ϕ

236 — 45

23 — ϕ

~~36~~ — 45

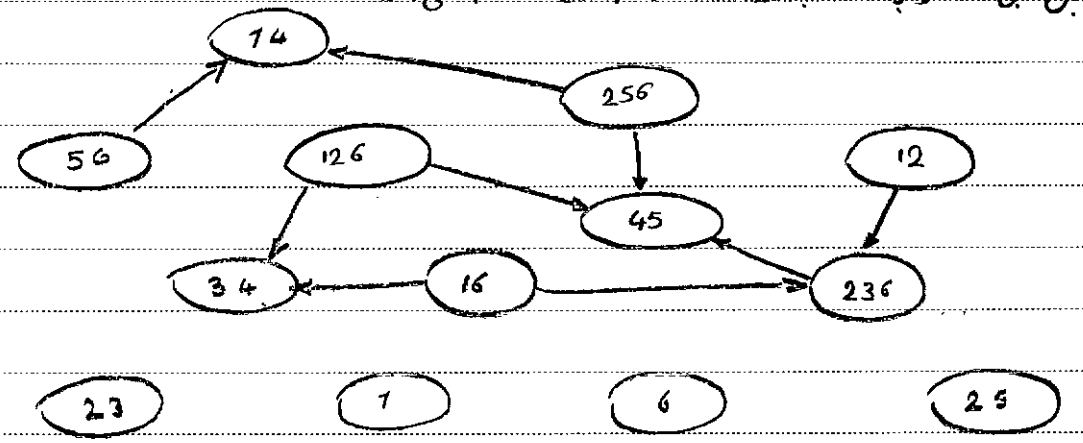
~~26~~ — ϕ

~~34~~ — ϕ

45 — ϕ

Subject:
Year. Month. Date. ( )

نودهایی که در جدول باقی ماند و بر آن علامت نشد، prime compatible می‌نامند.



* 26 توسط 236 علامت شده است.

دنبال نودی هستیم که بیشترین تعداد ورودی دلتای خودش را داشته باشد: «45»

236 به 45 اشاره می‌کند پس آن را انتخاب می‌کنیم (یکی از انتخاب‌ها)

12 به 236 اشاره می‌کند.

با سه state داریم (کاردینالیتی هایم ۳ بود): 16 ، 236 ، 45 «closed»

		I_1	I_2	I_3
Ⓐ	12	B, ∅	C, ∅	B, —
Ⓑ	236	C, 1	C, ∅	B, ∅
Ⓒ	45	A, —	A, 1	B, —

		I_1	I_2	I_3
Ⓐ	(126)	B, ∅	C, ∅	A, —
یا Ⓑ	(34)	C, 1	A, 1	—, ∅
Ⓒ	(45)	A, 1	A, 1	A, —

☑

1	3, ∅	—	6, —
2	—	4, ∅	6, —
3	5, 1	—	—, ∅
4	—	1, 1	1, —
5	1, —	—	6, —
6	2, —	5, ∅	6, —

mc's .

class set .

implication graph .

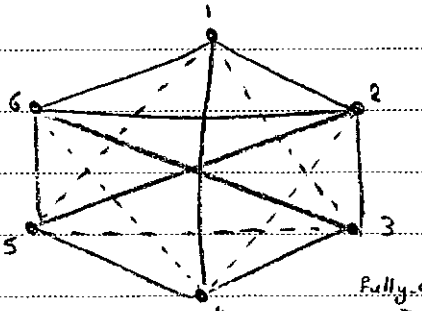
M' table .

P4PCO

Subject: 10

Year. Month. Date. ( )

	1	2	3	4	5
2	✓				
3	X x	✓			
4	1,6	X	✓		
5	1,3	✓	1,5	1,6	
6	3,2	4,5	2,5	1,6	✓



Fully connected گراف

mc's = {126, 236, 256, 14, 45, 35}

min comp's = {135, 24, 46}

$L = \{5, 6\}$

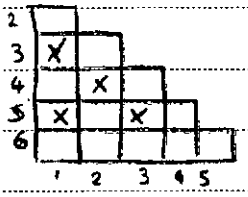
$S_4 = \{5\} \Rightarrow L = \{56, 45\}$

$S_3 = \{4\ 6\} \Rightarrow L = \{34, 36, 45, 56\}$

$S_2 = \{3\ 5\ 6\} \Rightarrow L = \{23, 236, 25, 256\}$

$S_1 = \{2\ 4\ 6\} \Rightarrow L = \{126, 14\}$

$L = \{34, 45, 14, 236, 256, 126\}$



$L = \{5, 6\}$

$S_4 = \{6\} \Rightarrow L = \{46, 5\}$

$S_3 =$

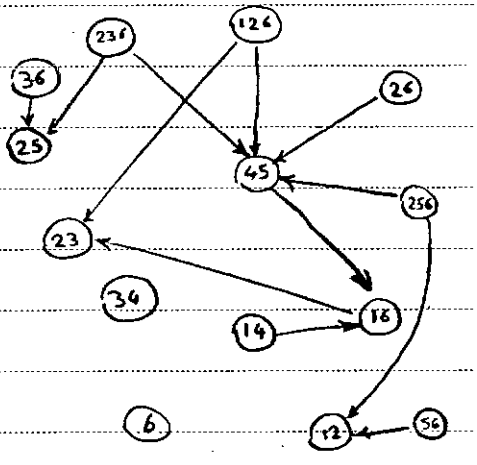
$S_2 =$

$S_1 =$

P_i	C_i
23, 45, ϕ 23 45	126 12 16 26
25, 45 ϕ 25	236 23 36
12, 45 ϕ 12	256 25 56
16	14
16	45
ϕ	34
ϕ ϕ ϕ ϕ ϕ ϕ	1 2 3 4 5 6

x x x x x

مقرب 2 — مقرب 23 — مقرب 23 — مقرب 36 — مقرب 45



مثالی : 45 , 16 , 23

PAPCO

		I_1	I_2	I_3
(A)	16	B, 0	C, 0	A, –
(B)	23	C, 1	C, 0	A, 0
(C)	45	A, –	A, 1	A, –

Subject:

Year. Month. Date. ( )

	K = 0	K = 1
1	5, 0	1, 0
2	4, 0	2, 0
3	5, –	3, –
4	1, 1	1, 1
5	1, –	2, –

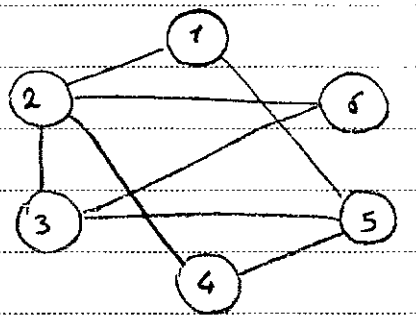
Subject: 11

Year. Month. Date. ( )

	x_1x_2 = 00	01	11	10
1	2	①	①	①
2	②	6	4	②
3	2	③	③	6
4	5	④	④	2
5	⑤	3	1	⑤
6	2	⑥	3	⑥

این جدول را کدگذاری و ماتریس YZ آن را بدست آورید.

node	adjacent → همسایه‌های پایدار
1	2, 5
2	6, 6, 3, 1
3	6, 5
4	2, 5
5	3, 1, 4
6	2, 3



$m = \lceil \log_2 6 \rceil = 3$

y_3 \ y_1y_2	00	01	11	10
0	stable 2	1	stable 5	3
1	buffer 2	4	5	6

جدول کدگذاری همسایه‌ها

باید adjacent باشند

نباید adjacent شوند.

unicode !!! نمی‌شود.

x_1x_2 = 00	01	11	10	$y_1y_2y_3$ P.S	
000	001	001	000	0 0 0	2^a
001	101	011	001	0 0 1	2^b
011	011	011	001	0 1 1	4
001	010	010	010	0 1 0	1
110	100	010	110	1 1 0	5^a
111	110	110	111	1 1 1	5^b
001	101	100	101	1 0 1	6
000	100	100	101	1 0 0	3

gray code بر اساس Present state

* تابعی universal

کمینه تعداد فلیپ فلاپی که برای درست کردن حداکثر احتیاج داریم: $S_0 = \lceil \log_2 n \rceil$

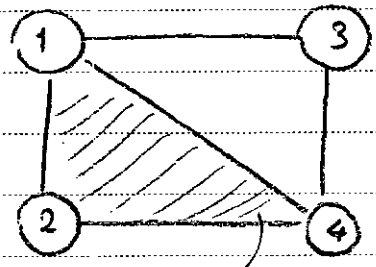
بیشینه تعداد فلیپ فلاپ‌ها می‌شود: $uni: 2S_0 - 1$

Subject:

Year. Month. Date. ( )

	00	01	11	10
1	①	2	3	①
2	②	②	4	1
3	1	4	③	③
4	1	④	④	3

node	adjacent
1	2, 3, 4
2	4, 1
3	4, 1
4	1, 2, 3



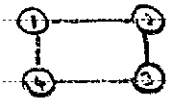
y_3 \ y_2y_1	00	01	11	10
0	1	4	4	2
1	1	3	—	—

مثلث را رد پس unicode می‌شود

اگر بخواهیم راه حل unicode پیدا کنیم چه می‌توان کرد؟

	00	01	11	10
1	①	2	3	①
2	②	②	4	1
3	1	4	③	③
4	1→3	④	④	3

node	adjacent
1	2, 3
2	4, 1
3	4, 1
4	3, 2



y_3 \ y_2y_1	0	1
0	1	2
1	3	4

به جای "10" می‌توان مثلث را زد.

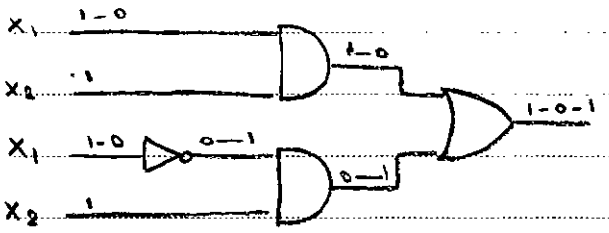
* انواع Delay

- طبیعی stray ← به ذات خود مدار ها موجود است.
- مصنوعی inserted ← زمانی که طراح ایجاد می کند تا بتواند زمانی را ایجاد کند.

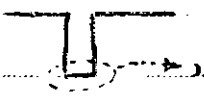
_ pure delay : تاخیر آن همواره مثبت است.

_ inertial delay : اگر پهنای باند سیگنال ورودی از تاخیر خود عنصر کمتر باشد، کنترلی خواهد داشت. سایر رفتارش مانند pure است.

* Hazard



static ☑



inertial delay این طور می شود ←

* انواع Hazard

Combinational < I

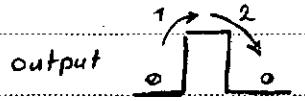
- static
- dynamic

Sequential < II

- steady state
- output
- essential

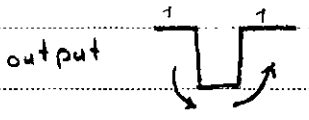
* Static Zero/One Haz.

S0:



تعداد تغییرات خروجی فرد است.

S1:



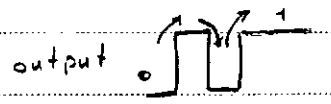
تعداد تغییرات خروجی زوج است.

Subject:

Year. Month. Date. ( )

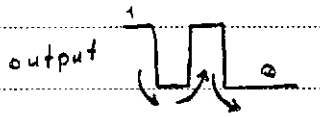
* Dynamic Hazard

D^{+1} :

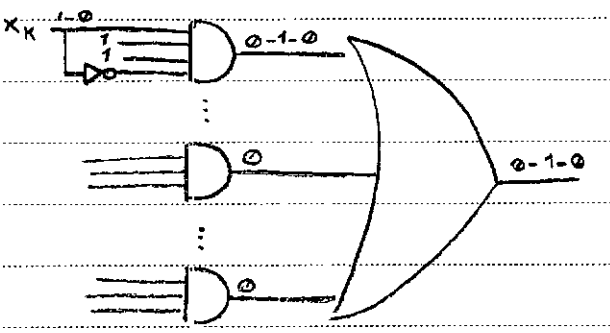


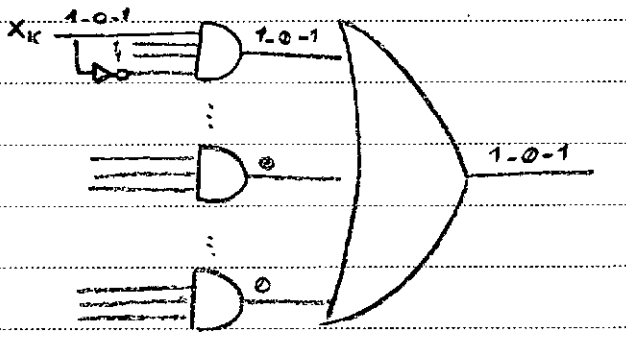
به تعداد فرد تغییر می‌کند

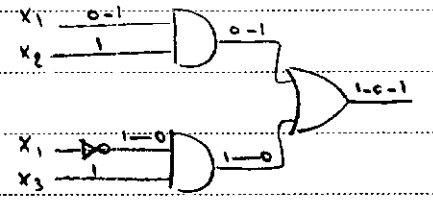
D^{-1} :

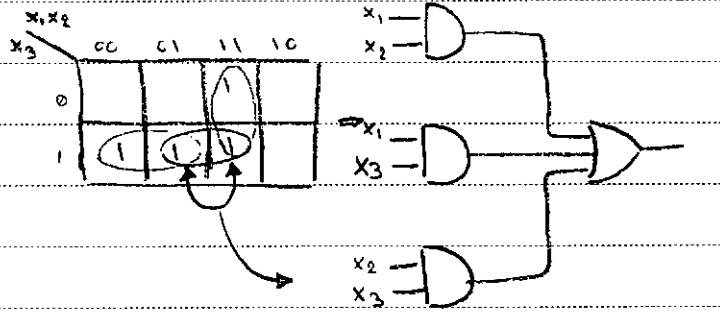


* اگر مدار دوسطحی به صورت Some Of Products داشته باشیم:









راه دیگر از بین بردن hazard استفاده از delay element است که البته مدار را کند می‌کند.

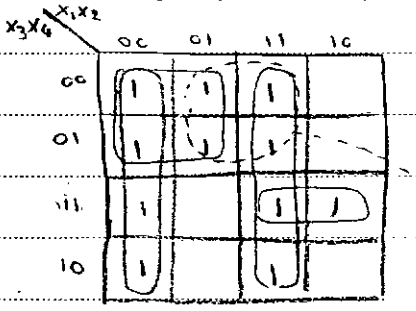
برای پیدا کردن سیستم‌های functional از مدارهای آشکارکننده استفاده می‌کنند.

Subject: 13

Year. Month. Date. ( )

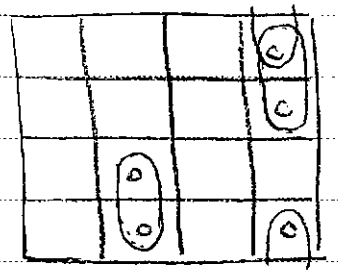
$E(x) = \bar{X}_1(\bar{X}_2 + \bar{X}_3) + (X_1 + \bar{X}_2)(X_2 + X_3X_4)$

☑ برنج static , dynamic را در این پیدا کنید.



$\bar{E}(x) = \bar{X}_1X_2 + \bar{X}_1\bar{X}_3 + X_1X_2 + X_1X_3X_4 + X_2\bar{X}_2 + \bar{X}_2X_3X_4$

آن - S1 مسئله‌ایم (پس عبارت دوم را اصلاح کنید) $(X_2\bar{X}_3)$



حال عبارت را معکوس می‌کنیم (اصل برای حالت static است می‌توان حذف کرد)

$\bar{E}(x) = X_1\bar{X}_2X_3 + X_1X_2\bar{X}_4 + \bar{X}_1X_2X_3$

آن - S0 مسئله‌ایم (از کاوشی نمی‌کنیم hazard ندارد)

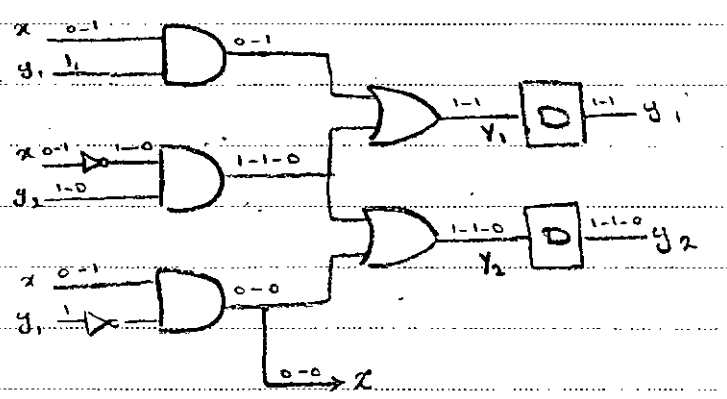
حال برای حالت dynamic متادیر $X_1 = 1$ و $X_3 = 1$ و $X_4 = 0$ در عبارت اولیه می‌گذاریم می‌بینیم که $E(x) = X_2 + X_2\bar{X}_2$ شده است. پس dynamic داریم.

* برای از بین بردن dynamic hazard می‌توان از inertial delay استفاده کرد.

* Sequential Hazard

	0 X 1	$y_1\ y_2$
1	①,0 | 2,1	0 0
2	3,0 | ②,1	0 1
3	③,0 | 4,0	1 1
4	1,0 | ④,0	1 0

☑

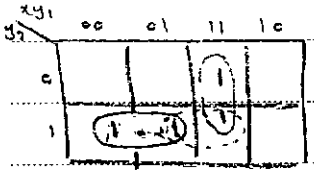


steady state : دوره‌ی تاریخ ی دیگر جدول کاوشی دوره‌ی کرا بی‌کنیم.

Subject:

Year. Month. Date. ( )

$Y_1 = xy_1 + \bar{x}y_2$



$Y_1 = xy_1 + \bar{x}y_2 + y_1y_2$

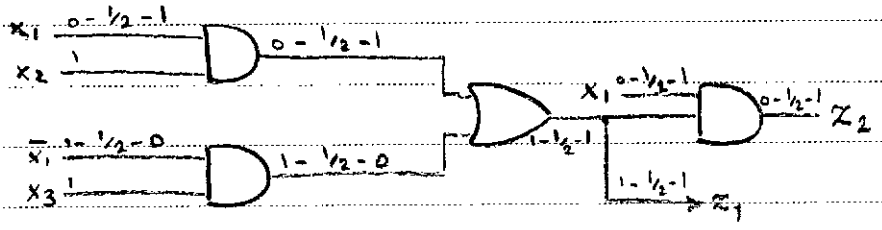
Let I_1 & I_2 be two adjacent input states and q_i an initial state of a flow table, such that $N(q_i, I_1) = q_i$ and $N(q_i, I_2) = q_j$ the flow table contain essential hazard for the $I_1 \rightarrow I_2$ transition from state q_i if the input sequence $I_1 I_2 I_1$ applied to the total state (q_i, I_2) results in a final state $q_k \neq q_j$

☑ آیا essential hazard دارد؟

	x_1x_2 00	01	11	10	y_1y_2
1	①	2	2	①	0 0
2	3	②	②	1	0 1
3	③	③	2	③	1 1

→ inserted / درج شده

* hazard ایستایی (یک) ثبت



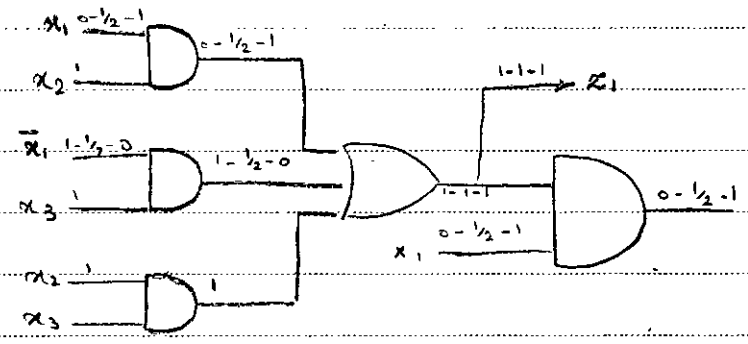
ثبت AND ورودی‌هایش ۱ می‌باشد.

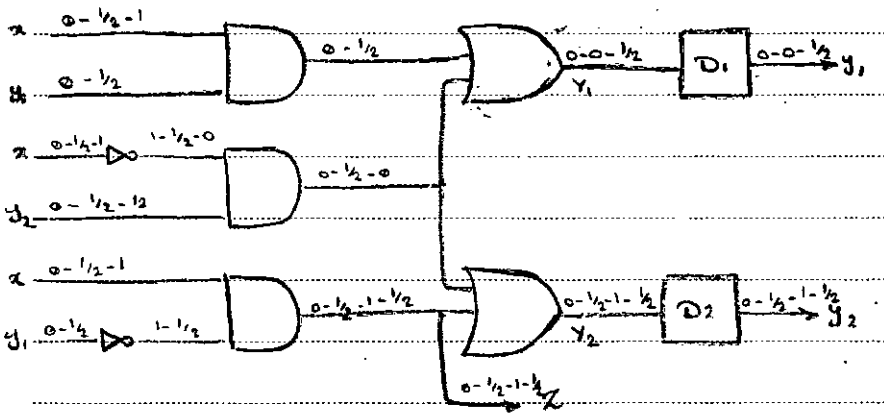
$1 - 1/2 - 1 \rightarrow$ S1

$0 - 1/2 - 0 \rightarrow$ S0

Subject: 14

Year. Month. Date. ( )





	x=0	x=1	$y_1 y_2$
1	①,0	2,1	0 0
2	3,0	②,1	0 1
3	③,0	4,0	1 1
4	1,0	④,0	1 0

* جبر ۳ تایی static hazard را هم در مدارهای ترکیبی، هم در مدارهای ترتیبی نشان می دهد فقط تایی برخلاف مدارهای feedback با چند loop باشند.

* جبر نشتایی

0 : low

1 : high

− : high → low

+ : low → high

S0 :

S1 :

D+ :

D− :

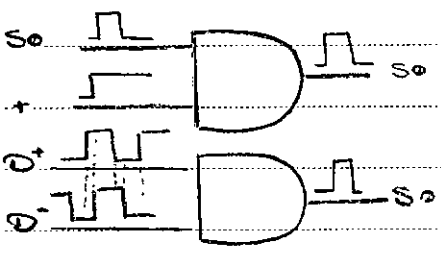
این موارد NOT باید به صورت دوتایی باشند.

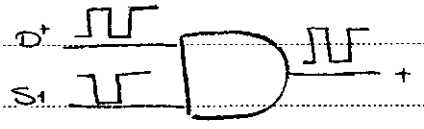
Subject:

Year. Month. Date. ( )

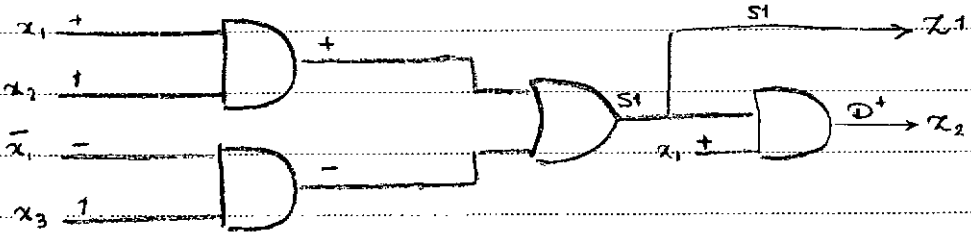
AND	1	0	+	−	S1	S0	D+	D−
1	1	0	+	−	S1	S0	D+	D−
0	0	0	0	0	0	0	0	0
+	+	0	+	S0	D+	S0	D+	S0
−	−	0	S0	−	D−	S0	S0	D−
S1	S1	0	D+	D−	S1	S0	D+	D−
S0	S0	0	S0	S0	S0	S0	S0	S0
D+	D+	0	D+	S0	D+	S0	D+	S0
D−	D−	0	S0	D−	D−	S0	S0	D−

OR	1	0	+	−	S1	S0	D+	D−
1	1	1	1	1	1	1	1	1
0	1	0	+	−	S1	S0	D+	D−
+	1	+	+	S1	S1	D+	D+	S1
−	1	−	S1	−	S1	D−	S1	D−
S1	1	S1	S1	S1	S1	S1	S1	S1
S0	1	S0	D+	D−	S1	S0	D+	D−
D+	1	D+	D+	S1	S1	D+	D+	S1
D−	1	D−	S1	D−	S1	D−	S1	D−





سرعت در حالت worst-case محاسبه می‌گردد یعنی بالاسب روی هم نمی‌افتند. سرعت AND در بهترین حالت minimum در نظر گرفته می‌شود.



عیب یابی و آزمون

* مدارهای ترتیبی

	x=0	x=1
A	A,1	B,0
B	B,0	C,0
C	C,0	D,0
D	A,1	D,1

$\overrightarrow{I} = \overrightarrow{111}$

state	output	final state
A	$\overleftarrow{000}$	D
B	100	D
C	110	D
D	111	D

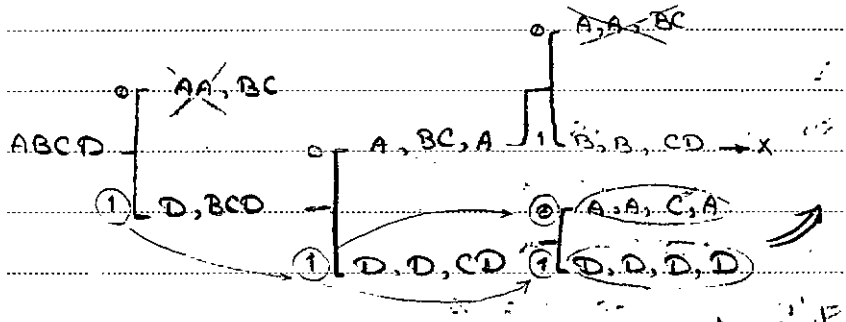
* Distinguishing Sequence

ترتیبی از ورود به صورت که با دادن آن به مدار، Initial State آن مشخص می‌شود.

* Homing Sequence

ترتیبی از ورودی‌ها است که با دادن آن به مدار، Final State آن مشخص می‌شود.

☑ استفاده از successor tree برای distinguishing sequence



∴ Distinguishing Sequence

Seq 1 = 111

Seq 2 = 110

① برنداد ستونی جدول رحمت استفاده کنید.

② برابرین خروجی‌ها آنرا جدا می‌کنیم.

③ اگر یکی از عناصر زوج تایی با بقیه آنرا ادامه می‌دهیم.

④ اگر عناصر یک زوج عیناً مانند یکی از اعداد خود باشند آنرا ادامه می‌دهیم.

⑤ اگر عناصر همه تک تک (single states) باشند به جواب کمینه می‌رسیم و بقیه شاخه‌ها را تا حد آن پیش می‌آوریم

Subject:

Year. Month. Date. ( )

I. State	output (I=111)	F. State	output (I=110)	F. State
A	000	D	000	C
B	100	D	100	A
C	110	D	110	A
D	111	D	111	A

راه ایجاد: جفت‌های output یکسان

□ استفاده از successor tree برای یافتن homing sequence

distinguishing seq پیدا می‌شود تا بتوانیم ابتدا اثر روی جفت‌های یکسان را پیدا کنیم

homing seq و distinguishing seq هم هستند (AA□) بنابراین آن را اجرا می‌کنیم. پس تمام

* شرط وجود داشتن homing seq.

1. fully specified ← don't care ندارد.

2. strongly connected ← از هر state به هر state دیگری می‌توان رفت

I. State	output (I=0,10)	F. State
A	001	B
B	000	C
C	100	A
D	001	B

→ احتمالاً output‌ها یکسان نیستند.

* Transfer Sequence

مجموعه ورودی‌هایی که روی state ها اعمال می‌شوند تا به state دیگری برسانند.

* Synchronizing Sequence

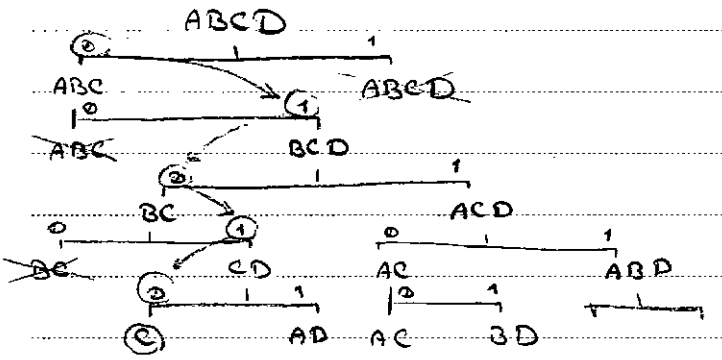
ماشین به هر state باشد با این seq به state خاصی می‌رسد. اکثر ماشین‌ها فاقد آن هستند.

این seq ها state نهایی‌اند واصلاً خروجی کاری ندارند. به همین دلیل نودهای آن تک‌پری

برادرهای نیز خط می‌خورند.

Subject: 16

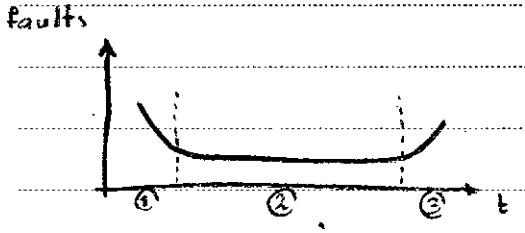
Year. 84 Month. 10 Date. 4 ( )



Sync: 01010

از هر حالتی به C می رود.

* منحنی زندگی یک تجهیز



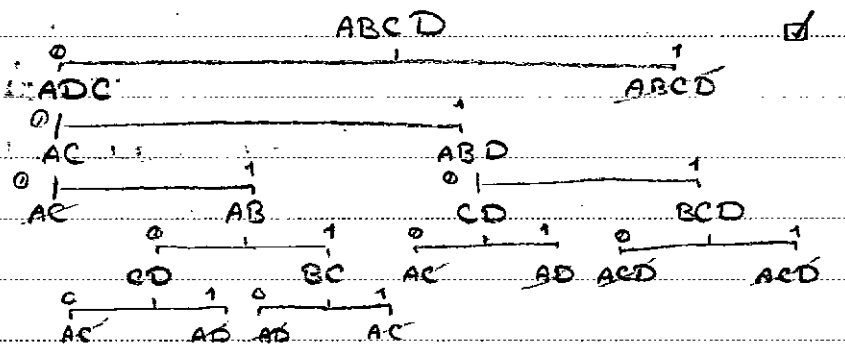
1. manufacturing
2. useful
3. old

* قابلیت تست در مدارهای با قابلیت زیر بالاست:

- Synchronizing Seq.
- Transfer Seq.
- Distinguishing Seq.
- Homing Seq. (حتماً مدار دارای آن نیست.)

و همچنین Checking Seq. می گویند.

	0	1
A	C, 0	B, 0
B	D, 1	C, 0
C	A, 1	A, 0
D	C, 0	D, 0



خروجی مهر نسبت پس:

«AD, CC» → «ADC»

Sync: 01010 → می دیم C.

Transfer To A: 010100

PAPCO

Subject:

Year. Month. Date. ( )

Distinguishing : x

ABCD

0 — 1

x AD, CC x ABCD

Homing : 00

ABCD

0 — 1

C, AD x ABCD

0 — 1

C, A BD, A

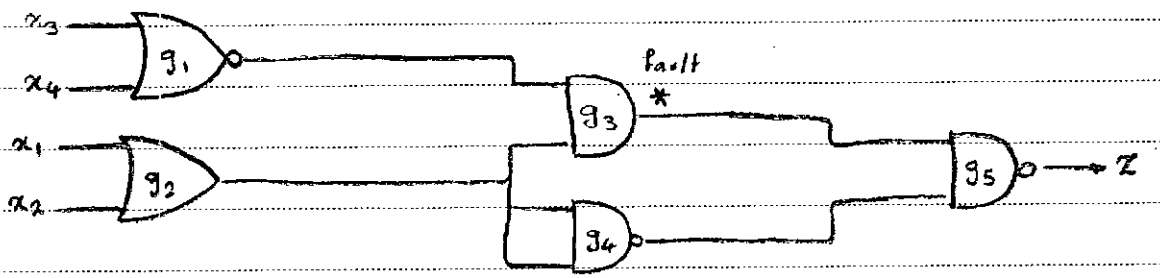
I. state	output	F. state
A	01	A
B	10	C
C	10	C
D	01	A

← Testing Table

Checking Seq: 01010 + 0 + 00

→ C → A → A (01 : خروجی)

Boolean Difference بررسی Fault کشف



$F = x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4$

فرض می‌کنیم جواب دارای fault باشد، اگر جواب مورد انتظار f باشد و جواب دوست آمده f_α باشد، وجود fault درصورتی است که:

$f \oplus f_\alpha = 1$

حال از معادله هر ورودی که در آن شک داریم، از آن مشتق می‌گیریم:

$\frac{df}{dx_i} = \dots$

$T_i = \overline{x_i} \cdot \frac{df}{dx_i} = \dots$ عبارت تست با جواب بود:

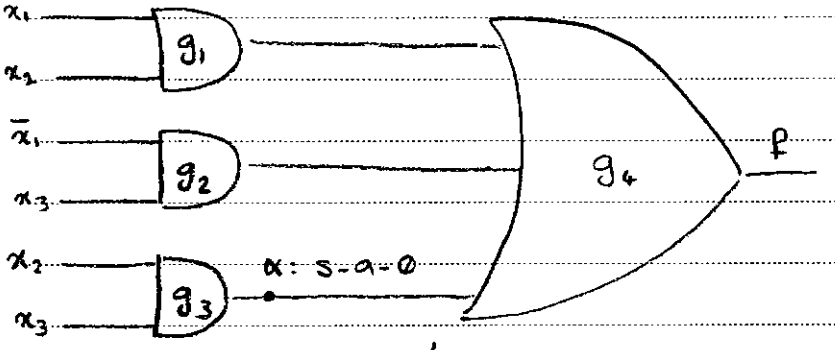
بسط شانون:

بر ورودی x_3 شک داریم که "stach at one" پس در $\bar{x}_3$ (یعنی خلاف آن) شک داریم.

$$T = \left[(f(x_3=0)) \oplus (f(x_3=1)) \right] \cdot \bar{x}_3$$

$$T = \left[(x_1x_4 + x_2x_4) \oplus (x_1+x_2) \right] \cdot \bar{x}_3 = \left[(\bar{x}_1+\bar{x}_4)(\bar{x}_2+\bar{x}_4)(x_1+x_2) \right] \bar{x}_3 = \dots + \dots$$

سمت برای این عبارت یک نقطه است



$\alpha : s-a-0 \longrightarrow$ در خط g_3 را داریم

$$f = x_1x_2 + \bar{x}_1x_3 + g_3$$

$$T = \left[(1) \oplus (x_1x_2 + \bar{x}_1x_3) \right] g_3 = (\bar{x}_1+\bar{x}_2)(x_2+\bar{x}_3)\, g_3 = (x_1\bar{x}_2 + \bar{x}_1\bar{x}_3 + \bar{x}_2\bar{x}_3)(x_2x_3) = \emptyset$$

این مدار نا تست است، fault آن بسط خود مدار خواهد بود. حال اگر $\alpha : s-a-1$ باشد fault را نشان می دهد.

* fault ← دائم
  - fault masking : جواب اصلا خطا اتفاق درست می شود!
  - failure : به خروجی دیده می شوند
* fault ← موقت
  - transient : با روشن شدن مجدد (اشعه α)
  - intermitant : با برآورده شدن شرایط (رطوبت، ...)

Subject:

Year. Month. Date. ( )

X : 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 Test Sequence

Z : 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 output (correct)

در سه state این جدول A, B, C باشد، Initial State آن A باشد جدول آنرا پیدا کنید.

	0	1
A	B, 0	B, 0
B	C, 1	C, 1
C	A, 0	C, 0

state : A B C A B C C A B C A B C A B

* ماشینهای FSM که دنباله ورودی را می‌گیرند، encode می‌کنند و output تولید می‌کنند. ماشینهای Information lossless خودی را گرفته، decode کرده و ورودی را درست می‌آورند. دو سه نوعند:

I > IL 1

output + Init state ⟹ input

II > IL 2

output + Final state ⟹ input

III > GIL (general)

output + Init state + Final State ⟹ input

این ماشین عمل خود را با یک state table انجام می‌دهد.

Subject: 16

Year. Month. Date. ( )

	0	1
1	1,0	2,1
2	1,1	4,0
3	4,0	3,1
4	2,0	4,1

output : 1 0 1

init state = 1

out = 1 0 1

state = 1 4 4

In = 1 1 1

اگر دنبال سطر چند خروجی ایستان داشتیم باید آنها را شناخت کنیم

Lemma 1:

A sequential machine M is IL1 iff $Z(q_i, I_j) \neq Z(q_i, I_K)$ for all $I_j \neq I_K$

Lemma 2:

A sequential machine M is IL2 if no next state output entry (q_i, z_j) appear more than once in the state table.

	0	1
1	1,0	2,0
2	2,1	3,1
3	4,1	1,1
4	4,0	3,0

Output = 0 1 1

Final state = 4

out = 0 1 1

state = 1 2 3 4

input = 1 1 0

Lemma 3:

If some (q_i, z_j) entry appears in state table more than once the machine is IL2 only if (q_i, z_j) does not appear in different columns of state table.

P4PCO

Subject:

Year. Month. Date. ( )

	0	1
1	3,0	2,0
2	3,0	2,0
3	4,1	1,1
4	4,1	2,1

⇒

F.S	Z(output) 0	1	
1	1	3	→ initial state
2	2	4	
3	12	—	Test Table
4	—	34	
12	12	34	
34	12	34	

out put : 0000 F.S : 3

out :	0	0	0	0	out :	1	1	1	1
state :	12	12	12	3	state :	34	34	34	4
input :	1	1	1	0	input :	0	0	0	0

out :	1	0	0	1	
state :	34	12	12	34	4
input :	1	1	0	0	

: Lemma 4

A sequence machine M is GIL iff no initial state has two or more identical entries, AND no two states with the same (N,Z) entry are in the same set of the test table.

	0	X	1
1	3,0		4,1
2	4,0		3,1
3	1,0		2,0
4	3,1		4,1

forbidden sets : 14 , 24

I.S	Z(output) 0	1	
1	3	4	→ next state
2	4	3	Test Table
3	12	—	
4	—	34	
12	34	34	
34	12	34	

Subject: 19

Year. Month. Date. ( )

output: 0 0 0 1 0 F.S = 2 I.S = 1

out: 0 0 0 1 0

state: 1 3 ~~2~~ ~~4~~ 3 ~~2~~

input: 0 ? 0 ? ?

1 0 1

	0	1
1	2,0	3,0
2	4,0	2,1
3	3,1	2,0
4	1,1	3,0

forbidden sets: 14 (3,0) , 13 (2,0)

I.S	0	1 (output)
1	23	—
2	4	2
3	2	3
4	3	1
23	24	23
24	34	12
34	23	13

output: 0 0 0 1 0 I.S = 1 F.S = 2

out: 0 0 0 1 0

state: 1 23 24 34 13 ~~23~~

input: ? ? ? 0 ?

در مسیر برگشت به مثل می‌خوریم.

برای بدست آوردن آن output ای که ماشین از حالت GIL خارج می‌کند از آخر جدول به اول آن می‌رویم و به آخر آن (2,0) اضافه می‌کنیم.

Subject:

Year. Month. Date. ( )