به نام خدا

سایت گروه آموزشی آلم

ابتدایی، راهنمایی، دبیرستان، کنکور و دانشگاه

www.g-alm.ir	
www.g-alm.ir/ac	دانشگاه
www.g-alm.ir/forum	انجمن
www.g-alm.ir/azmoon	آزمون های آلم
www.g-alm.ir/shop	فروشگاه
www.film.g-alm.ir	فیلم های آموزشی

Subject: ①
Year. Month. Date.

حل تمرین‌ها ← logcir892@yahoo.com

user & pass : logic892

http://ee.sharif.edu/~logiccircuits892

مرجع: Digital Design (Mano)
1) Introduction to switching theory
2) logical design (Hill & Peterson)
3) مدارهای منطقی دیجیتال (Nelson , ...)

میان ترم: آخرین ۵ شنبه فروردین (۹۰/۱/۲۸) ساعت ۵ بعد از ظهر

تا آخر مدارهای ترکیبی

مباحث امتحان: تا آخر جلسه نهایی

PLD تا آخر فصل

نمره ۸۵ پایان ترم
نمره ۴ میان ترم
نمره ۳ آزمون
} با هم

نمره ۲۵ کوئیز، تمرین، پروژه

SM, 2's C, 1's C, BCD } نمایش اعداد

جمع و تفریق در مبناهای مکمل و نمایش‌های مختلف

طراحی مدارهای ترکیبی ماجول

۰ - ۵ ⟵ ۲۵۶ مقدار متمایز

۵V ⟵ ۰ ولتاژ

مبدل دیجیتال به ولتاژ D/A — اجرای عملیات — مبدل ولتاژ به دیجیتال A/D

ساده‌سازی توابع منطقی (sop, pos)

(Hazard-free) Glitch-free design

تشخیص و رفع خطا

مطالب:

۱) سیستم اعداد (مبنای ۲)
- مبناهای مختلف
- محاسبات
- نمایش اعداد منفی
- مکمل‌ها
- کدهای باینری

۲) جبر بول و گیت‌های منطقی
- تعریف و قضایای جبر بول
- توابع بولی و فرم‌های استاندارد
- گیت‌های منطقی

۳) ساده کردن توابع منطقی
- روش جدول کارنو
- استفاده از حالت‌های بی‌اهمیت

۴) مدارهای ترکیبی
- روش طراحی
- جمع کننده و تفریق کننده

۵) طراحی مدارهای دیجیتال
- مقایسه کننده‌ها
- Decoder
- Multiplexer
- RAM

۶) مدارهای ترتیبی

۷) رجیستر - RAM - شمارنده - PAL

۸) طراحی با استفاده از دیاگرام ASM

۹) مدارهای آسنکرون

$$N = a_n a_{n-1} \cdots a_0 . a_{-1} a_{-2} \cdots a_{-m} = (a_n r^n + \cdots + a_0 + a_{-1} r^{-1} + \cdots + a_{-m} r^{-m})$$

$$(11.01)_2 = 1\times2^3 + 1\times2^2 + 1\times2^0 + 1\times2^{-2} = 13.25$$

اعداد اعشاری { ۱- ممیز ثابت ۲- ممیز شناور

* تبدیل از مبنای ۱۰ به ۲:

قسمت صحیح عدد N = ...

قسمت اعشاری N = $0.a_{-1}a_{-2}\cdots a_{-m} = a_{-1}r^{-1} + \cdots + a_{-m}r^{-m}$

$r \times N = a_{-1} + (a_{-2}r^{-1} + \cdots + a_{-m}r^{-m+1})$

$(511.23)_{10} \longrightarrow (?)_2$

$0.23 \times 2 = 0.46 \rightarrow a_{-1}$

$0.46 \times 2 = 0.92 \rightarrow a_{-2}$

$0.92 \times 2 = 1.84 \rightarrow a_{-3}$

warning

$0.84 \times 2 = \cdots$

* نکته: اگر قسمت اعشاری به صفر تبدیل نشد، به دلخواه k رقم اول را ذخیره می‌کنیم و بقیه را دور می‌ریزیم.

Subject: ②
Year. Month. Date. ( )

مبنای ۲ ← باینری (دودویی) ، مبنای ۸ ← اکتال (octal) ، مبنای ۱۶ ← هگزادسیمال (Hexadecimal) ، مبنای ۱۰ ← دسیمال (دهدهی)

هر رقم در مبنای ۲ ← ۱ بیت (Bit) ، هرهشت Bit ← ۱ بایت (Byte) ، $1K \leftarrow 2^{10} = 1024$ ، $1M \leftarrow 2^{20}$

تبدیل از مبنای r_1 به r_2 : از r_1 ← ۱۰ ، از ۱۰ ← r_2

Warning → جداسازی از سمت راست

از ۲ به ۸ یا از ۲ به ۱۶ و بالعکس :

(مثال) $N = (\underbrace{11}_{3}\underbrace{010}_{2}\underbrace{101}_{5})_2 = (?)_8 \rightarrow (325)_8$

$N = (\underbrace{11}_{13}\underbrace{010}_{2}\underbrace{101}_{5})_2 = (?)_{16} \rightarrow (D5)_{16}$

* حاصل جمع دو عدد N رقمی حداکثر N+1 رقمی است.

$$\begin{array}{r} 10110110 \\ +\ 11101101 \\ \hline 1\ 10100011 \end{array}$$

carry Flag (CF)

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1\ 1\ 0 \\ -\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0 \\ \hline 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1 \end{array}$$

* نمایش اعداد منفی :

۱- سیستم اندازه علامت : (۱ منفی ، ۰ مثبت) : چون جمع و تفریق سخت است از این سیستم کمتر استفاده می‌شود.

$23 \rightarrow (10111)_2 \Longrightarrow -23 \rightarrow 110111$ ، $23 \rightarrow 10111$

(اندازه | علامت)

$(-23)_{10} = (?)_2$

۲- مکمل (r-1) : $C_{r-1}(N) = r^{n+1} - N - r^{-m}$

n: تعداد ارقام صحیح

m: تعداد ارقام اعشاری

مکمل ۹ : همه‌ی ارقام را از ۹ کم کنیم.

مبنای $r = 10$ → $N = 785 \Rightarrow C_9(N) = 214$

۳- مکمل (r) :

$$C_r(N) = \begin{cases} 0 & N = 0 \\ r^{n+1} - N & N \neq 0 \end{cases}$$

n: تعداد ارقام صحیح

مکمل ۱۰ : رقم سمت راست را از ۱۰ و بقیه را از ۹ کم می‌کنیم.

$N = 785 \Rightarrow C_{10}(N) = 215$

$N = 2817 \Rightarrow C_9(N) = 7182$ ، $C_{10}(N) = 7183$

* مکمل ۱ : همه‌ی بیت‌ها را معکوس می‌کنیم :

* مکمل ۲ : مکمل ۱ + $(r^{-m})1$

مکمل ۱

0000 +0
1111 -0

0001 +1
1110 -1

0010 +2
1101 -2

0111 +7
1000 -7

مکمل ۲

0000 +0
0000 -0

0001 → +1
1111 → -1

۴ بیت → $-8 \leq N \leq +7$

۸ بیت → $-128 \leq N \leq +127$

1000 0000 ... 0111 1111

* هرگاه جمع دو عدد مثبت، منفی شود یا جمع دو عدد منفی مثبت شود (اگر تغییر علامت ناخواسته داشته باشیم) بیت overflow یک می‌شود.

* بیت آخر علامت عدد را نشان می‌دهد (در سیستم مکمل ۲)

Subject: (۳)
Year. Month. Date. ( )

$A - B = A + C_r(B)$

* مکمل ۲ گرفتن از عدد منفی یعنی عدد مثبتش را به دست می‌دهد.

$+۷ = 0111$, $+۴ = 0100$, 1011

1011 ← خطا X ← (range از خارج شده) overflow ← صحیح نیست ← $-۸ \le < +۷$

-۵

⇓ مشکل overflow با افزایش تعداد Bit حل می‌شود.

مثال: $+۷ - ۴ = ۳$

$+۷ + (-۴) \Rightarrow 0111 + 1100 = 10011$

سرریز (carry) ← ۱ ، $0011 = ۳$

(مکمل ۲) * در سیستم مکمل ۲ اگر حاصل می‌بایست سرریز داشت، سرریز را دور می‌اندازیم و حاصل درست است.

(مکمل ۲) * اگر سرریز نداشت (در سیستم مکمل ۲) حاصل درست است.

مثال: $۷ - ۸ = -۱$

$0111 + 1000 = 1111$ ← -۱

$A - B = A + C_{r-1}(B)$ مثال: $۷ - ۴ = ۳$

$۷ \to +C_{r-1}(۴) \to 0111 + 1011 = 10010$

$10010 + 1 = 10011 = ۳$

۲

(مکمل $r-1$) * اگر سرریز داشتیم، سرریز را دور می‌اندازیم و یکی به حاصل اضافه می‌کنیم. } ← در سیستم مکمل $(r-1)$ ۱

* اگر سرریز نداشتیم، حاصل درست است.

* کدهای باینری:

۱) کد دودویی (اعداد همیشه مثبت) 1011 ← 8421 ← کدهای وزن دار

۲) 5043210 ← کد وزن دار

۰: ۰۱۰۰۰۰۱ ← "ما دانستی می‌کنیم که کدها همیشه دارای ۲ عدد "۱" و ۵ عدد "۰" هستند"

۱: ۰۱۰۰۰۱۰

⋮

۹: ۱۰۱۰۰۰۰ ⟹ 1011000 X

خطا (نویز، خرابی سیستم ...)

parity (توازن)

۳) کد با توازن زوج یا فرد

P d_2 d_1 d_0

۰ ۰ ۰ ۰

۱ ۱ ۰ ۰ ۱

۲ ⋮

۷ ۱ ۱ ۱

P را به گونه‌ای مقداردهی می‌کنیم که تعداد ۱ های کد حاصل عددی زوج یا فرد باشد.

* خطاهای $n = 2k+1$ بیتی را تشخیص می‌دهد.

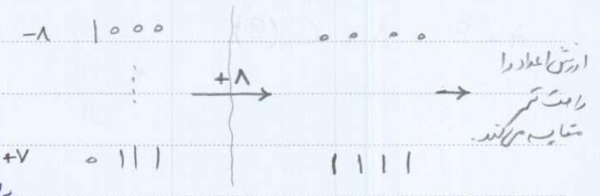
* خطا در صورتی قابل تشخیص است که کد معتبر به کد غیر معتبر تبدیل شود.

PAPCO

Subject: ②
Year. Month. Date. ( )

۴) کد ۲ از ۵

۵) کد k افزا (۸- افزا برای سیستم ۴ بیتی)



۶) کد ASCII (۷ بیتی) → برای نمایش اعداد - علائم - کاراکترها
۱ parity + ۷ بیتی (۸ بیتی) → به منظور تشخیص خطا

۷) کد همینگ (Hamming)

تعریف: d = hamming distance

مثال:
۱ ۱ ۰ ۱
۱ ۰ ۱ ۰
✓ × × × ⇒ d = ۳

(k parity bit , m data bit)

یک بیت تصحیح خطا ⇒ $m \leq 2^k - k - 1$ معادله

$k: ۳ \quad 2^k - k - 1 \geq 4 \quad m = 4\ bit$

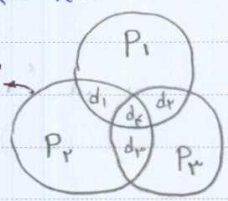
$k = ۴$ ⇐ تشخیص خطای دوگانه

اگر $2t + s \leq d_{min} - 1$

t: تعداد تصحیح خطا
s: " تشخیص خطا
d_{min}: کمینه فاصله همینگ بین کدهای یک مجموعه کد

کد همینگ ۱
H (7, 4)
۷ بیتی — ۴ داده

استاندارد
$m = 4$
$k = 3$



d_4	d_3	d_2	d_1	P_3	P_2	P_1	
۰	۰	۰	۰	۰	۰	۰	= ۰
۰	۰	۰	۱	۰	۱	۱	= ۱
۰	۰	۱	۰	۱	۰	۱	= ۲
۰	۰	۱	۱	۱	۱	۰	= ۳

$d_{min} = ۳ \Rightarrow 2t + s \leq ۲$

$t = 1$

$N = 1001011$ (X)

P_3	P_2	P_1
۱	۰	۰

$d_4 = ۰$ ⇑ P_3, P_2, P_1 ← هر سه اشتباهند پس d_4 خطا خورده.

$N = 0101011$

P_3	P_2	P_1
۱	۰	۱

$d_3 = ۰$ ⇐ P_3, P_2 متناقض ⇐

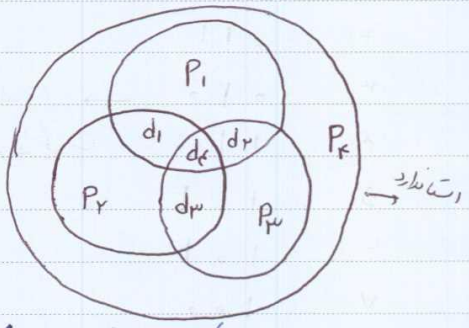
Subject: ۵
Year. Month. Date. ( )

روش یافتن کد معتبر در همینگ ۱ :

فاصله تمام کدهای معتبر را با کد دریافتی می‌سنجیم : اگر فاصله = ۰ بود معتبر است.
اگر یکی از فاصله‌ها = ۱ بود کد معتبر متناظر با همان کد است.

کدهای معتبر	کد خطا	فاصله
۰۰۰۰ ۰۰۰	۱۰۰۱۰۱۱	۴
۰۰۰۱ ۰۱۱	۱۰۰۱۰۱۱	۱ ⇐ کد معتبر
۰۰۱۰ ۱۰۱	۱۰۰۱۰۱۱	۵
⋮	⋮	⋮

کد همینگ ۲ H(8,4)
(4bit data 4bit parity)



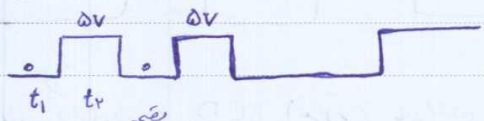
همینگ ۱	P_4
۰۰۰۰۰۰۰	۰
۰۰۰۱۰۱۱	۱
۰۰۱۰۱۰۱	۱

(مثال) همینگ ۱
خطا $N = 11010\,11$

کدهای معتبر	کد خطا	فاصله
۰۰۰۰۰۰۰		۵
۰۰۰۱۰۱۱		۲
⋮	N	⋮
۱۱۱۰۱۰۰		۵
۱۱۱۱۱۱۱		۲

← زیرا که فاصله‌اش ۱ باشد کد معتبر است.

انتقال داده
- موازی: (data bus) → گروه‌هایی از سیم‌ها
- سری



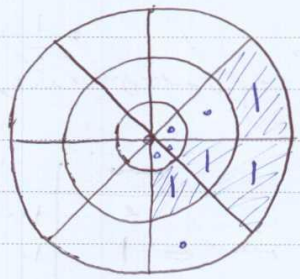
→ ۰۱۰۱۰۰۱۱

کد معتبر	تشخیص	d_{min}	$2t+s+1 \leq d_{min}$	$d+c+1 = d_{min}$ ← تشخیص / → تصحیح
	X	1	$t=0, s=0$	$d, c = 0$
	تشخیص یگانه	2	$t=0$ $s=1$	$d=1$, $c=0$
	تصحیح یگانه / تشخیص دوگانه	3 یا 3	$t=1, s=0$ / $t=0, s=2$	$d=1, c=1$
	تصحیح یگانه و تشخیص دوگانه	4	⊗ $t=1, s=1$	
	تشخیص سه‌گانه	4	$t=0, s=3$	

Subject: ⑥
Year. Month. Date. ( )

۸) Gray code

تشخیص موقعیت موتور →



۰	۰۰۰	۰۰۱
۱	۰۰۱	۰۰۰
۲	۰۱۰	۰۱۱
		۰۱۰

کدای مشاهده شده → دوبیت تغییر

۰	۰۰۰
۱	۰۰۱
۲	۰۱۱
۳	۰۱۰
۴	۱۱۰
۵	۱۱۱
۶	۱۰۱
۷	۱۰۰

نوعی کد حلقوی است. (Gray code) → در گذار از یک code به یک code دیگر فقط یک بیت تغییر دارد. → دو code

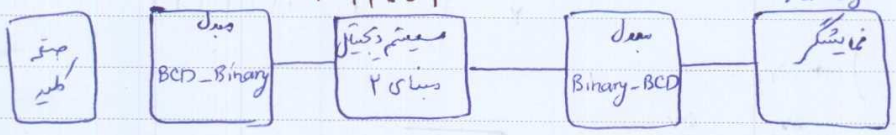
* کاربرد: مقایسه شمارنده های دو FIFO آسنکرون، تحقیقات، ...

۸) BCD کد : Binary Coded Decimal

نمایش اعداد مبنای ۱۰ (decimal)

	باکد دودویی (باینری)
۰	۰۰۰۰
۱	۰۰۰۱
۲	۰۰۱۰
۳	۰۰۱۱
⋮	⋮
۹	۱۰۰۱

۲۵ دهدهی →
1- باینری 00011001
2- BCD 0010 0101
۲ ۵



تمرین: در چه حالات دیگری در جمع BCD (تمرین) باید تصحیح BCD انجام دهیم؟ چگونه؟

```
  25
+ 65
------
 0010 0101
 0110 0101
-----------
 1000 1010
    8    A
+      6   (تصحیح BCD)
-----------
    9    0
```

* سریز در سطح دهگان داریم.
* در تبدیل ۱۰ رقم بالا کمک گرفتیم پس لزوم تصحیح نداریم.
* حاصل های را ۶ می کنیم.

Subject : (۷)
Year . Month . Date . ( )

اعداد صحیح:
a) مثبت $0 \le N \le 255$
b) مثبت و منفی:
- اندازه علامت: $-127 \le N \le 127$
- مکمل ۲: $-128 \le N \le +127$

?

اعداد اعشاری:
- ممیز ثابت — Fixed point — مکمل ۲ — $-8 \le N \le +7/\dots$ — $d = 0.0001 = \frac{1}{16}$ دقت (precision)
  * اگر ممیز وسط باشد
- ممیز شناور — Floating point

Range ↓↑ بستگی به جای ممیز

مثال ↓
مبنای ۱۰: $500 = 5.0 \times 10^1 = 5 \times 10^2 = (+)\ 0.5 \times 10^3$
(۳ → exponent (توان))
(علامت) sign — (ماتیس) Mantissa

(هنجارسازی) Normalization → $5.1\dots \times 10^2$ → نمایش بایاس

sign { 0 : + , 1 : − }

exponent $0 < e < 255$ → $-126 \le E \le 127$ ، $1 \le e \le 254$
۱۲۷ نفر — توان حقیقی — نمایش بایاس

$e = E + \frac{bias}{127\ \ 1023}$

IEEE 754

		s	e	M
single precision	32 bit →	1	8	23
double "	64 bit →	1	11	52

مبنای ۲ → 1. xxxx -- x → $M = 1.f$
- 1 → x حذف
- xxxx -- x → fraction
- M → 24 بیت (23+1)
- f → 23 بیت

(مثال) ۱) عدد صحیح 32 bit
$N = 1111\ 0000, 1100\ 1100, 1010\ 1010, 0000\ 1111$ (حذف ↑)
$= 1.111\ 0000\ 0000, 1100\ 1100, 1010\ 1010 \times 2^{31}$
(23 بیت)

خطای مطلق = ۲۵۵ ← هم مثبت: خطای نسبی هم است.

۱) $e = 0, f = 0 \Rightarrow N = 0$
۲) $e = 0, f \ne 0 \Rightarrow N = (-1)^s \times 0.f \times 2^{-b+1}$ → Denormalized اعداد
$b = bias$
(کاربرد: نمایش اعداد خیلی کوچک) مثال: $0.0000001 \times 2^{\dots}$
۳) $e \ne 0, e \ne \text{all } 1 \Rightarrow N = (-1)^s \times 1.f \times 2^{e-b}$ → Normalized اعداد
۴) $e = \text{all } 1, f = 0 \Rightarrow N = \pm\infty$
۵) $e = \text{all } 1, f \ne 0 \Rightarrow N = NaN$ Not a Number

Subject: ۸
Year. Month. Date. ( )

مثال) 111.1001

$= 11.11001 \times 2^{1}$ ← E

$= 1.111001 \times 2^{2}$ ← E=2

f

Denormal: $0.0000\ldots01 \times 2^{-126}$ (f) ... $0.11\ldots1 \times 2^{-126}$

* ممیز شناور سیستم نمایش اعداد ـ علامت دارد.

* محاسبات ممیز شناور:

1.01×2^{2}

$+ 1.10 \times 2^{3}$

e_2 e_1 → مقایسه کننده

0.101×2^{3}

مثال) $۲۵,۳۷۵ = (?)_{۲}$

1) fixed point: 11001.011

$0.375 \times 2 = 0.75$ ⓪

$0.75 \times 2 = 1.5$ ①

$0.5 \times 2 = 1.0$ ①

2) Floating point

$N = (+)\ 1.\underbrace{1001011}_{f} \times 2^{4}$ → E

$S = 0$ (+)

$S = 0$

$E = 4 \rightarrow e = 4 + 127 = 131 = 10000011$

$f = 1001011$

S | e | f

0 | 10000011 | 1001011000...0

⇒

(توابع بولی (منطقی)) Boolean functions

0 : False : نادرست : ⓪

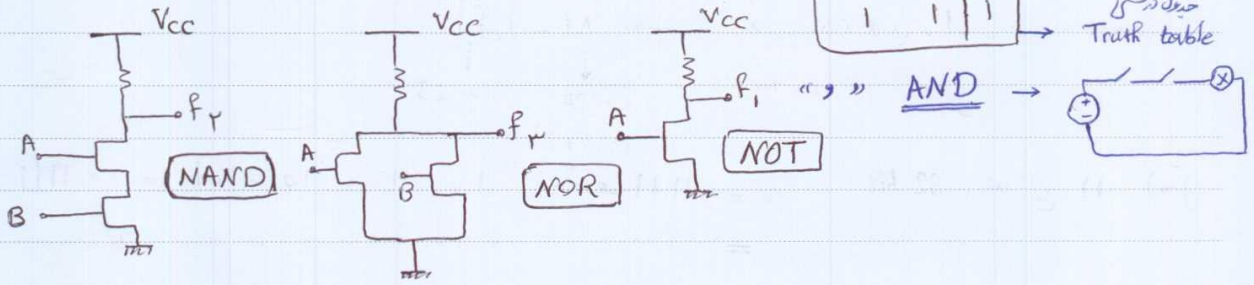
1 : True : درست : ①

سنسور ۲	سنسور ۱	f → توابع بولی
0	0	0
0	1	0
1	0	0
1	1	1

جدول درستی Truth table

تابع f

سنسور ۱ / سنسور ۲

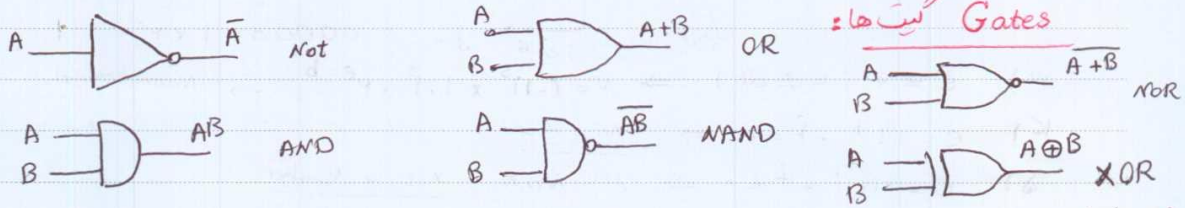


« و » AND →

$f_2 = \overline{A \cdot B}$ $f_3 = \overline{A + B}$ $f_1 = \overline{A}$

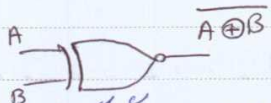
AND: ^ , "و" , "." , OR = ∨ , "یا" , "+"

گیت ها: Gates



Subject: ⑨
Year. Month. Date. ( )

XNOR (exclusive nor)



AB	Xor
0 0	0
0 1	1
1 0	1
1 1	0

* زمانی از ورودی‌ها صفر باشد ورودی دیگر برگردانده می‌شود.

تعریف: مجموعه‌ی B همراه با دو عمل + و . یک جبر بول نامیده می‌شود اگر:

۱) B نسبت به "+" و "." بسته باشد.

$$\left.\begin{matrix} x \in B \\ y \in B \end{matrix}\right\} \Rightarrow \begin{cases} x+y \in B \\ x \cdot y \in B \end{cases}$$

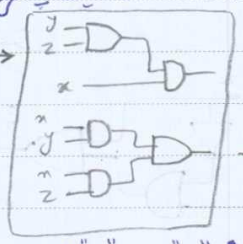
۲) عملگرهای "+" و "." جابه‌جایی داشته باشند.

$$x+y = y+x \quad , \quad x \cdot y = y \cdot x$$

۳) " " خاصیت پخشی داشته باشند.

$$x \cdot (y+z) = x \cdot y + x \cdot z$$

$$x + (y \cdot z) = (x+y) \cdot (x+z)$$



معادل هستند

۴) وجود عضو خنثی ("۰" , "۱")

$$x + 0 = x$$

$$x \cdot 1 = x$$

۵) وجود عضو متمم

$$x \in B \quad x + \bar{x} = 1$$

$$\exists \Rightarrow x' \in B \quad x \cdot \bar{x} = 0$$

۶) B حداقل ۲ عضو متمایز داشته باشد.

۷) خاصیت شرکت پذیری

$$a+(b+c) = (a+b)+c$$

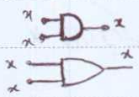
$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

قضایای جبر بول:

۱- متمم هر عضو مختصر به فرد است. $x \in B \Rightarrow x'$

۲- ۰ و ۱ مختصر به فرد هستند.

۳- هم توان

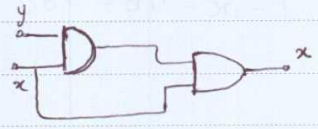


$$\leftarrow \begin{cases} x+x=x \\ x \cdot x = x \end{cases}$$

۴- متمم متمم x خود x است.

۵- $x+1=1$, $x \cdot 0 = 0$

۶) قانون جذب



$$x + (x \cdot y) = x$$

$$x \cdot (x+y) = x$$

اثبات $x + xy = x \cdot 1 + x \cdot y = x(y+1) = x \cdot 1 = x$

۷- ۰ و ۱ دو عضو متمایزند و $0' = 1$

* خاصیت دوگانی :

اگر در یک رابطه همه‌ی + به ۰ و ۰ به + و ۱ به ۰ و ۰ به ۱ تبدیل شود دوگان آن رابطه به دست می‌آید.

$x + xy \overset{\text{دوگان}}{\Longleftrightarrow} x \cdot (x+y) = x$

کاربرد : برای یافتن متمم یک تابع.

$f = xy + \bar{x} + \cdots \Rightarrow \bar{f} = \overline{xy + \bar{x} + \cdots}$

۸ - قضیه‌ی دمورگان : $\overline{x+y} = \bar{x} \cdot \bar{y}$, $\overline{x \cdot y} = \bar{x} + \bar{y}$

(برای اثبات قضیه‌ی دمورگان تمرین)

۹ - قضیه‌ی تراکم :

$ab + \bar{a}c + bc = ab + \bar{a}c$ ← (راهنمای اثبات) $bc = bc(a+\bar{a})$

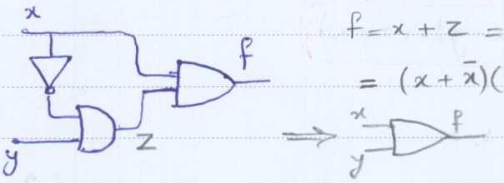
$(a+b)(\bar{a}+c)(b+c) = (a+b)(\bar{a}+c)$ ← دوگان

* هرگاه رابطه‌ای درست باشد، دوگان آن هم درست است.

مثال ۱

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

$= (x+\bar{x})(x+y) = x+y$



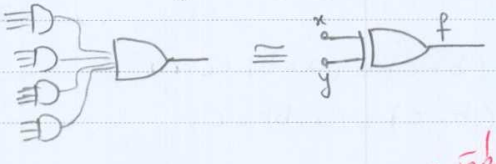
مثال ۲

$f = x'yz + x'yz' + xy'z' + xy'z$

$f = x'y(z+z') + xy'(z'+z)$

$= x'y + xy' = x \oplus y$

x	y	f
0	0	0
0	1	1
1	0	1
1	1	0



مراحل طراحی یک مدار منطقی :

۱ - نوشتن جدول درستی ۲ - نوشتن تابع بولی از روی جدول درستی ۳ - ساده کردن تابع بولی ۴ - رسم مدار

(جدول کارنو / جدول QM) ← * برای این که متمکن باشیم به ساده‌ترین مدار رسیده‌ایم.

مثال ۳

$f = \bar{c}(b+a\bar{b}) + ab + \bar{a}c$

$= c(b+a) + \bar{a}b + \bar{a}c$

$= cb + c(a+\bar{a}) + \bar{a}b$

$= cb + c + \bar{a}b$

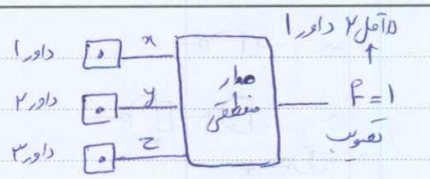
$= c + \bar{a}b$

$f = \bar{a}c + \bar{a}b + \underbrace{a\bar{b}c}_{\bar{a}b} + bc$

$= \bar{a}(b+c) + bc$

Subject: (11)
Year. Month. Date. ( )

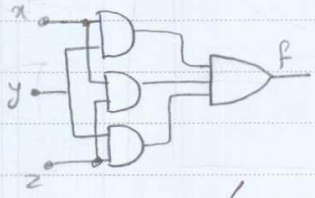
x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



شکل ۴)

$f = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz = \underline{\bar{x}yz} + \underline{x\bar{y}z} + \underline{xy\bar{z}} + \underline{xyz} + xyz + xyz$

$= yz + xz + xy$



مدار مربوطه

* متمم تابع اولی $\bar{f}$: یعنی در جدول درستی هرجا f ۱ بود حالا صفر شود و برعکس.

pos = product of sums

مثال) $f = x + \bar{x}y$

sop = sum of products

$\bar{f} = \overline{x + \bar{x}y} = \bar{x} \cdot (\overline{\bar{x}y}) = \boxed{\bar{x} \cdot (x + \bar{y})} = \bar{x}\bar{y}$

فرم کانونیک نمایش تابع
- sop ← فرم مینترم ← Minterm
- pos ← فرم ماکزیمم ← Maxterm

راه دوم: استفاده از دمورگان: هرگاه دمورگان آن را بطه هم ورودی ها معکوس شوند متمم آن به دست می آید.

مثال) sop: $f = \underbrace{x'yz'}_{\text{Minterm}} + xyz + xyz'$

pos: $\bar{f} = \underbrace{(x+\bar{y}+z)}_{\text{Maxterm}} \cdot (\bar{x}+\bar{y}+\bar{z})(\bar{x}+\bar{y}+z)$

Maxterm	x	y	z	Minterm
$x+y+z$	0	0	0	$\bar{x}\bar{y}\bar{z}$
	0	0	1	$\bar{x}\bar{y}z$

* فرم کانونیک در هر جدول باید حتما هر سه ورودی را داشته باشیم ← حالت خاصی از sop و pos شده اند.
* Minterm : ترکیبی از تمام ورودی ها یا معکوس آن ها که با یکدیگر AND

minterm	x	y	f	maxterm
$\bar{x}\bar{y}$	0	0	0	$x+y$
$\bar{x}y$	0	1	1	$x+\bar{y}$
$x\bar{y}$	1	0	1	$\bar{x}+y$
xy	1	1	0	$\bar{x}+\bar{y}$

sop: $f = \underbrace{\bar{x}y}_{\text{Minterm}} + \underbrace{x\bar{y}}_{\text{Minterm}}$ pos: $= \underbrace{(x+y)}_{\text{Maxterm}}(\bar{x}+\bar{y})$

Minterm هایی که تابع را ۱ می کنند. Maxterm هایی که تابع را صفر می کنند.

* Maxterm, Minterm با توجه به عدد معادل شان نام گذاری می شوند.

مثال) $f = x'yz + xy'z + xyz' + xyz$

$\quad 011 \quad 101 \quad 110 \quad 111$

$\quad m_3 \quad m_5 \quad m_6 \quad m_7$

$= m_3 + m_5 + m_6 + m_7 = \sum(3,5,6,7) \Rightarrow$ sop

$= \Pi(0,1,2,4) \Rightarrow pos$

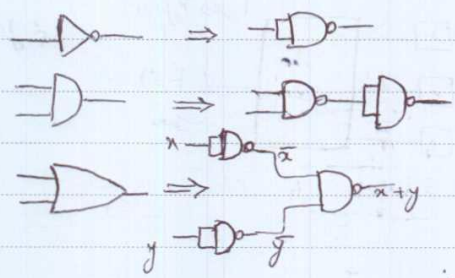
$= M_0 \cdot M_1 \cdot M_2 \cdot M_4$

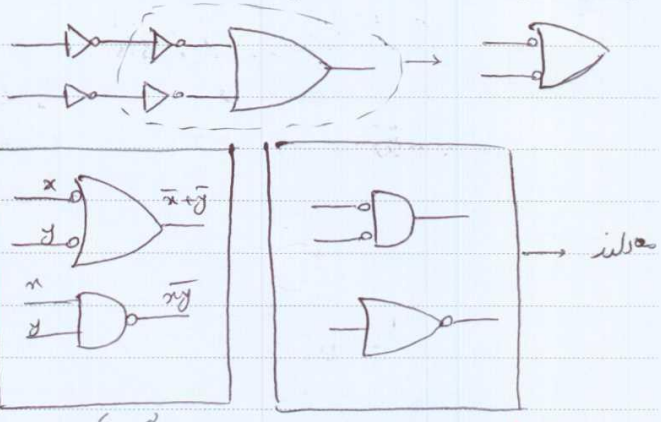
$= (x+y+z)(x+y+\bar{z})(x+\bar{y}+z)\cdots$

P4PCO

Subject: (۱۲)
Year. Month. Date. ( )

* گیت های NAND, NOR, ... universal هستند.

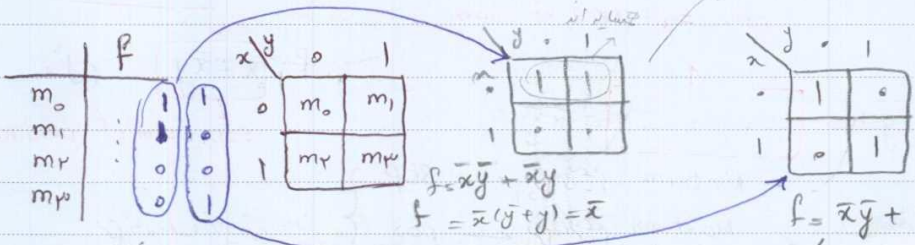




جدول کارنو :

$f(x,y) =$ sop

همسایگی مشترک دارند پس ساده می‌شود.

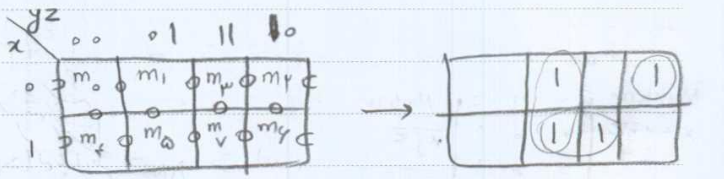


همسایه : خانه هایی که فقط در یک بیت با هم اختلاف دارند (بصورت gray code نوشته شود)

مثال $f(x,y,z) =$

x	y	z	f
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

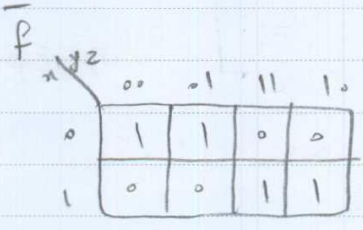
ترتیب مهم است.



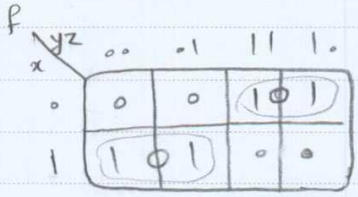
ساده‌ترین حالت نوشتن همان $f = \bar{y}z + xz + \bar{x}y\bar{z}$

مثال $f(x,y,z) = f = x'yz + x'yz' + xy'z' + xy'z$

$= \sum(2,3,4,5)$



$\bar{f} = \bar{x}\bar{y} + xy$ (XNOR)



$f = x\bar{y} + \bar{x}y$ (XOR)

if $x = y \Rightarrow f = 1$

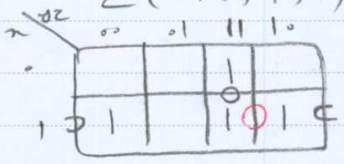
if $x \neq y \Rightarrow f = 1$

PAPCO

هدف جدول کارنو { ۱- کم‌ترین تعداد دسته ۲- دسته‌های بزرگ‌تر

مثال) $f = x'yz + xy'z' + xyz + xyz'$
011 100 111 110

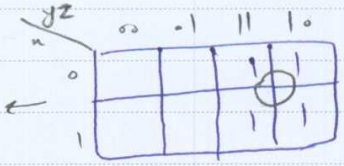
$= \sum(۳, ۴, ۶, ۷)$



$f = yz + x\bar{z} + xy$ ←

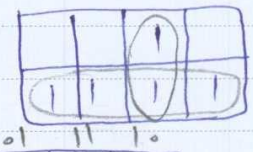
جمله اضافی (نیازی نیست)

مثال)



$f = y$ ←

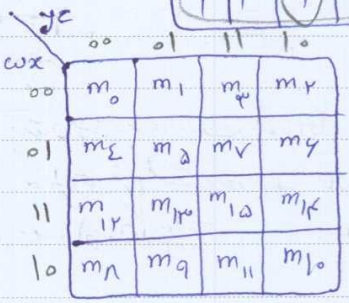
مثال)



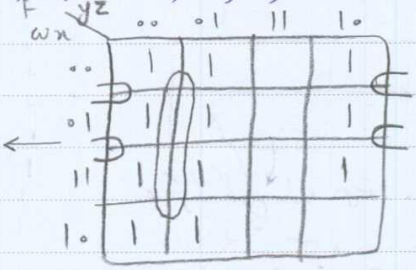
$f = x + yz$ ←

مثال)

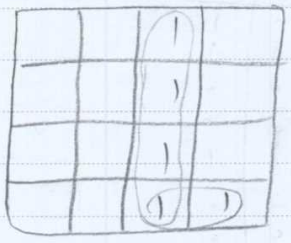
$f(\omega, x, y, z)$



مثال) $f(\omega, x, y, z) = \sum(0, 1, 2, 4, 5, 6, 8, 9, 12, 13, 14)$



$f = \bar{y} + \bar{\omega}\bar{z} + x\bar{z}$ ←

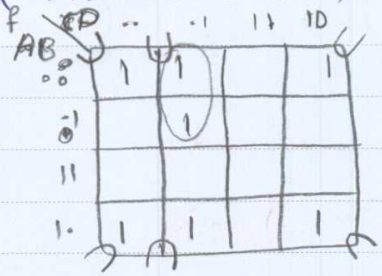


$\bar{f} = yz + \omega\bar{x}y$ ←

Subject: (۱۴)
Year. Month. Date. ( )

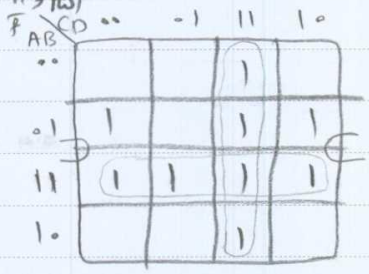
مثال) $f(A,B,C,D) = \sum(0,1,2,5,8,9,10)$

* شروع از بدترین حالت



$$f = \bar{B}\bar{C} + \bar{B}\bar{D} + \bar{A}\bar{C}D$$

$$\therefore \quad \bar{f} = \sum(3,4,6,7,11,13)$$



$$\bar{f} = CD + AB + B\bar{D}$$

$$f = (\bar{A}+\bar{B})(\bar{C}+\bar{D})(\bar{B}+D)$$

← f pos فرم

* اگر یک تابع f به فرم sop ساده شده باشد، $\bar{f}$ به فرم pos ساده شده است (و بالعکس)
* در مثال بالا فرم f pos ساده‌تر از فرم sop است به خاطر وجود ترم سه ورودی در sop ؛ بنابراین، گاهی اوقات فرم pos یک مدار ساده‌تر از sop آن است (و بالعکس).

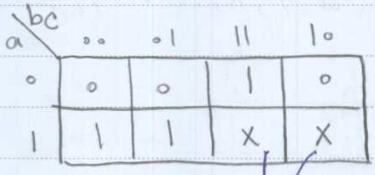
حالت بی اهمیت (don't care)

مثال

اگر عدد $> 3 \Rightarrow f = 1$

$0 \leq$ عدد ≤ 5

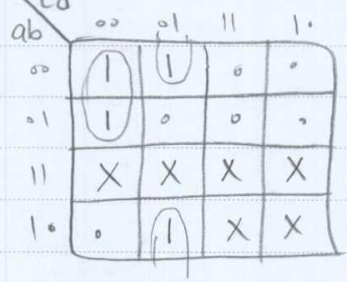
a b c	f
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	1
1 0 0	1
1 0 1	1
1 1 0	X → don't care
1 1 1	X



* اگر یک ترم مطلوب می‌شود. در این صورت:

$f = a + bc$

a b c d	β	$f = x^2$
0 0 0 0	0	1
0 0 0 1	1	1
0 0 1 -	2	0
0 0 1 1	3	0
0 1 0 0	4	1
0 1 0 1	5	0
0 1 1 0	6	0
0 1 1 1	7	0
1 0 0 0	8	0
1 0 0 1	9	1
	A	
	B	



مثال) اعداد BCD

در این جا X ها را صفر بگیریم بهتر است

$$f = \bar{a}c\bar{d} + \bar{b}\bar{c}d$$

P4PCO

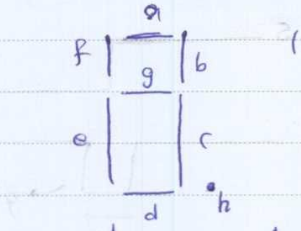
Subject: (۱۳)
Year. Month. Date. ( )

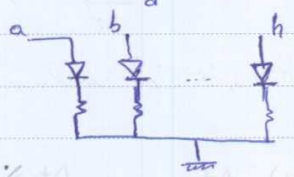
x y z w	a	b	c	d
0 0 0 0	1			
	0			
	1			
	1			
	0			
	1			
	1			
	1			
	1			
	1			
	X			
	X			
	X			

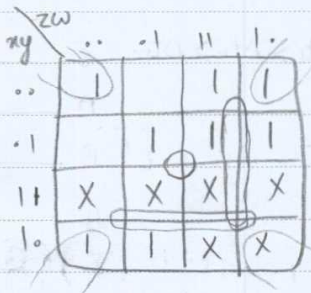
↳ برای اعداد بیشتر از ۹

abcdefgh
11110110

مثال)





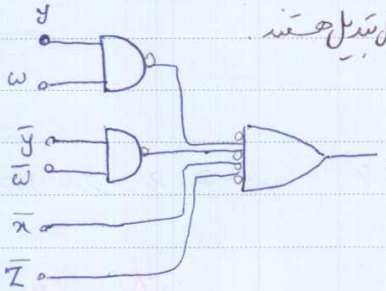


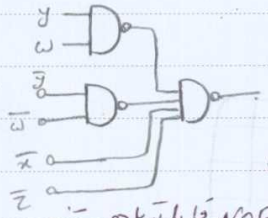
خانه‌های X را به دلخواه پر می‌کنیم

$a = x + z + y\omega + \bar{y}\bar{\omega}$

AND – OR

* شبکه‌های AND–OR را می‌توان به راحتی به NAND–NAND قابل تبدیل هستند



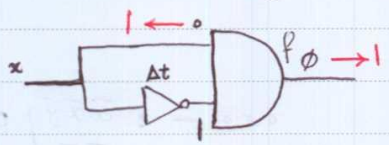


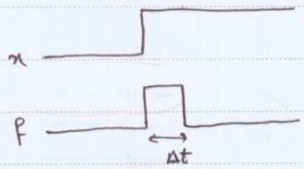
pos
OR–AND

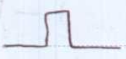
sop
AND – OR

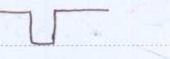
* شبکه‌های OR–AND (حاصل از pos) به راحتی به NOR–NOR قابل تبدیل هستند.

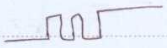
خاصره : hazard هر تغییر ناخواسته و نامطلوب در خروجی یک مدار منطقی که در اثر تغییرات ورودی‌ها حاصل می‌شود hazard گوییم.



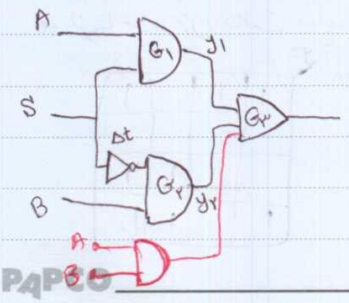


glitch  → (static) ایستا → glitch

 → یا (dynamic)



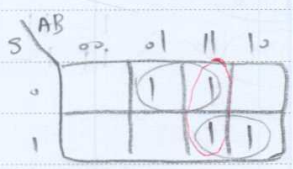
۱) $\Delta t_1 = \Delta t_2 = 0$ ، Δt ✓



$$\begin{cases} S = 0 : f = B \\ S = 1 : f = A \end{cases}$$

Multiplexer (MUX)

مثال)

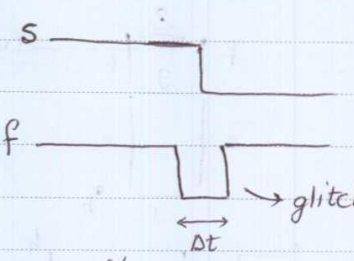


$f = \bar{S}B + SA + AB$

↳ برای از بین بردن glitch

PdPCC

Subject: (14)
Year. Month. Date. ( )

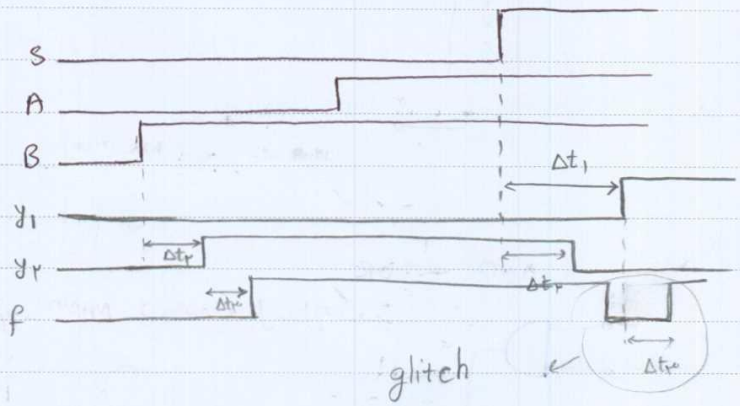


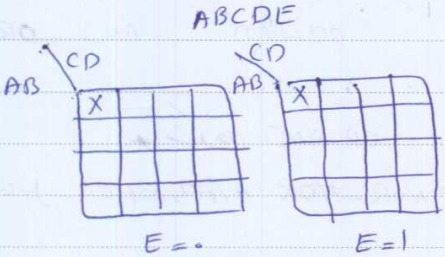
حالت روبرو را در نظر بگیرید: $AB = 11$, $S: 1 \to 0$

* warning : ۲ بی‌تغییر / درست → glitch نمی‌گویند.

* هنگام ساده کردن در جدول کارنو هر ۲ یک همسایه باید در یک دسته‌ی مشترک حضور داشته باشند. (برای جلوگیری از glitch)

۲) $\Delta t = 0$, $\Delta t_1 > \Delta t_2 > \Delta t_3$





جدول کارنو ≤ ۵ متغیره ۶ متغیره

روش QM

(مثال)

$f(w,x,y,z) = \sum (0,1,2,8,10,11,14,15)$

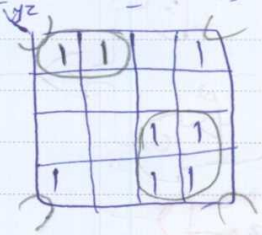
	مرحله اول	مرحله دوم	مرحله سوم
0	0000 ✓	000 —	
1	0001 ✓	00—0 ✓	—0—0
2	0010 ✓	—000 ✓	—0—0
8	1000 ✓	—010 ✓	1—1—
10	1010 ✓	10—0 ✓	1—1—
11	1011 ✓	1—10 ✓	
14	1110 ✓	1—11 ✓	
15	1111 ✓	11—1 ✓	

$\Rightarrow$ 000— : $\bar{w}\bar{x}\bar{y}$; —0—0 : $\bar{x}\bar{z}$; 1—1— : wy } P.I.

prime Implicant

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

* لزوماً ساده‌ترین حالت نیست.



Subject: (۱۷)
Year. Month. Date. ( )

PI	۰	۱	۲	۸	۱۰	۱۱	۱۴	۱۵
۰۰۰- $\bar{w}\bar{x}\bar{y}$	X	X						
-۰-۰ $\bar{x}\bar{z}$	X		X	X	X			
۱-۱- wy						X	X	X	X

* تکمیل جدول که بینیم می‌شود چیزی را حذف کرد یا خیر.

→ چون ■ = ۱ فقط در این دسته وجود دارد.

→

→ چون ۱۰ و ۱۴ و ۱۱ فقط با این دسته ساده‌تر می‌شود

$$F(w,x,y,z) = \sum(۱,۴,۶,۷,۸,۹,۱۰,۱۱,۱۵)$$

$\bar{w}\bar{x}\bar{y}z$ ۱	۰۰۰۱	- ۰۰۱	۱۰-- x
۴	۰۱۰۰	۰ ۱-۰	۱۰-- ✓
۸	۱۰۰۰	۱ ۰۰- ✓	
		۱ ۰-۰ ✓	
۶	۰۱۱۰	۰ ۱۱-	
$w\bar{x}\bar{y}z$ ۹	۱۰۰۱	۱۰-۱ ✓	
$-\bar{x}\bar{y}z$ ۱۰	۱۰۱۰	۱۰۱- ✓	
$--۰۱$			
۷	۰۱۱۱	- ۱۱۱	
۱۱	۱۰۱۱	۱-۱۱	
۱۵	۱۱۱۱		

		۱	۴	۶	۷	۸	۹	۱۰	۱۱	۱۵
- ۰۰۱	✓ $x'y'z$	X					X			
۰ ۱-۰	✓ $w'xz'$		X	X						
۰ ۱۱-	$w'xy$			X	X					
- ۱۱۱	✓ xyz				X					X
۱-۱۱	wyz									X
۱۰--	✓ wx'				X	X		X	X	X

$$\Rightarrow f = x'y'z + w'xz' + xyz + wx'$$

خانواده‌های logic

RTL → Resistor - Transistor logic

DTL → Diode - " "

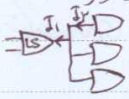
74XX TTL → Transistor - " " →

- 74HXX — high speed — سرعت دو برابر، مصرف دو برابر
- 74LXX — low speed — سرعت نصف، مصرف توان نصف
- 74SXX — schottky — سرعت سه برابر، مصرف دو برابر
- 74LSXX — low ~~speed~~ power schottky
- 74ALSXX — Advanced "
- 74FXX — Fast schottky

CMOS → MOSFET

- 74C
- 74HC
- 74HCT
- 74AC
- 74ACT

* Fan out : تعداد لیت‌هایی که می‌توانند در خروجی یک لیت مشابه قرار گیرند.

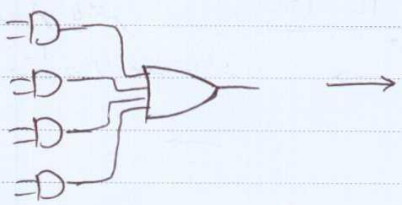


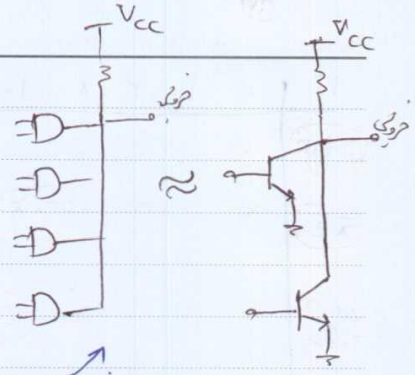
$$\text{Fan out} = \frac{I_1}{I_2} = \begin{matrix} ۵۰ \\ ۱۰ \end{matrix}$$

* حاشیه نویز : از سمش معلوم است.
* سرعت
* مصرف توان

Subject: ۱۸
Year. Month. Date. ( )

open collector





74LS 90

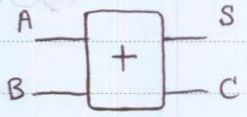
74LS01

↳ open collector → * برای مدار منطقی Fanout را مشخص می‌کند.

ساختار کلی

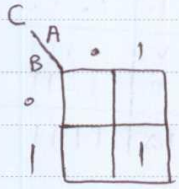
اجزای اساسی تشکیل دهنده‌ی مدارهای دیجیتال

۱- جمع کننده
۲- تفریق کننده
۳- مقایسه کننده
۴- MUX : Multiplexer
۵- DeMUX
۶- کدکننده Encoder
۷- کدگشا Decoder

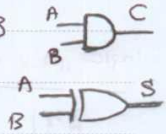


1 1 0
1 0 0
carry
sum

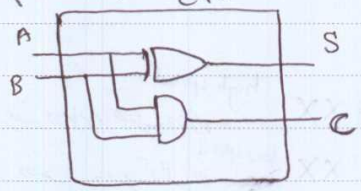
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

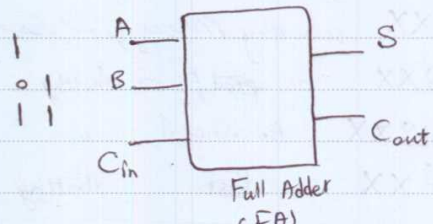


$C = AB$

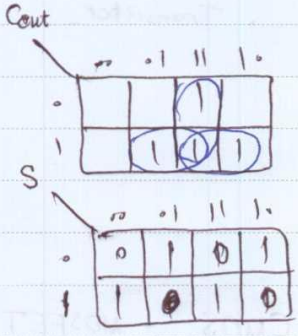


(HA) (Half Adder) نیم جمع کننده





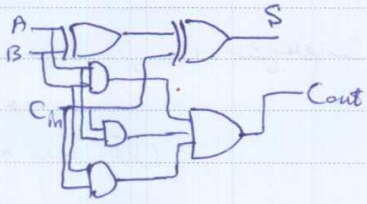
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



$C_{out} = AB + AC_{in} + BC_{in}$

$S = A \oplus B \oplus C_{in}$

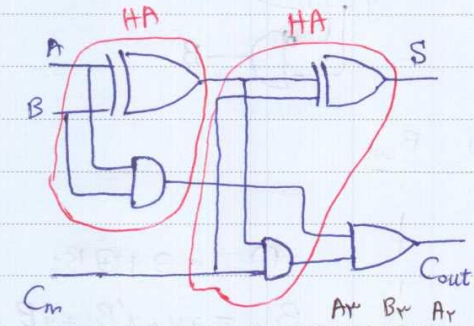
تمرین) کدام گیت ها خاصیت شرکت پذیری ندارند ؟



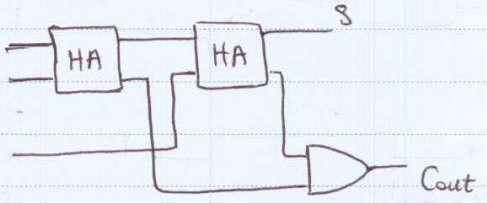
Subject: (19)
Year. Month. Date. ( )

$C_{out} = AB + AC_{in} + BC_{in} = AB + C_{in}(A+B) = AB + C_{in}(A \oplus B)$

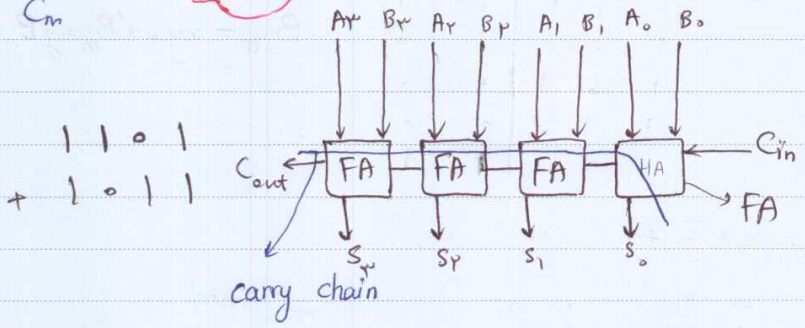
تبدیل به این شکل معادله



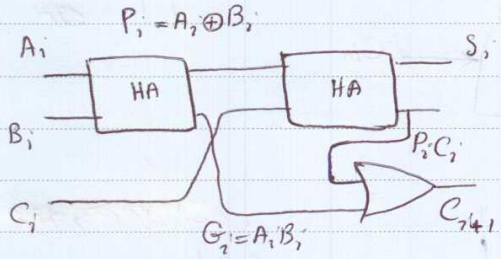
≅



مثال: جمع کننده ۴ بیتی:



carry look-ahead Adder



$C_{i+1} = G_i + P_i C_i$

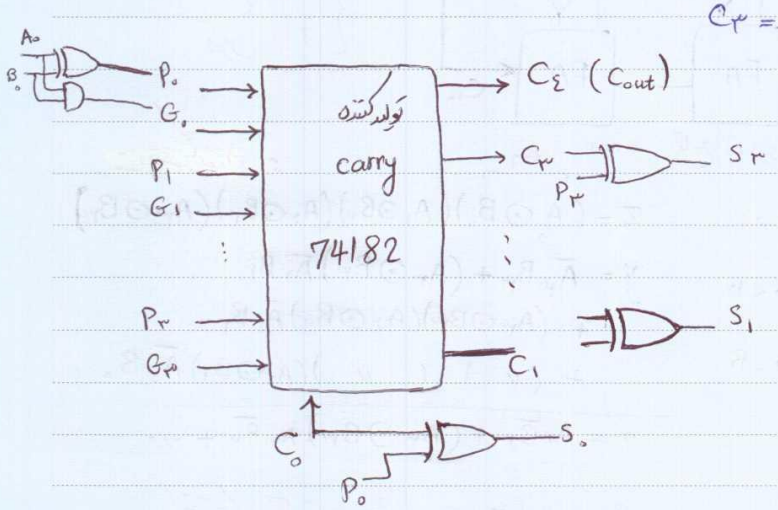
$C_1 = G_0 + P_0 C_0$

$= A_0 B_0 + (A_0 \oplus B_0) C_0$

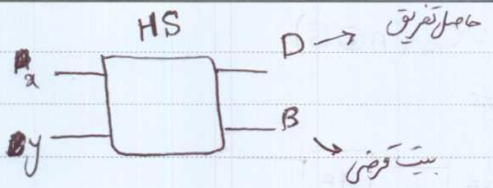
$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (A_0 B_0 + (A_0 + B_0) C_0)$

$= A_1 B_1 + (A_1 \oplus B_1)(A_0 B_0 + (A_0 + B_0) C_0)$

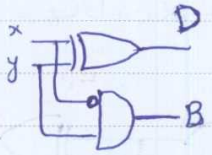
$C_3 = \cdots$

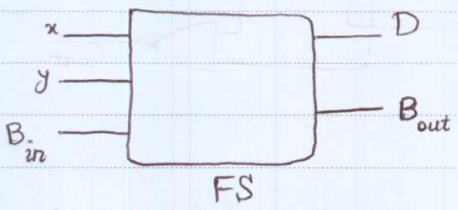


Subject: (۴۰)
Year. Month. Date. ( )



x	y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

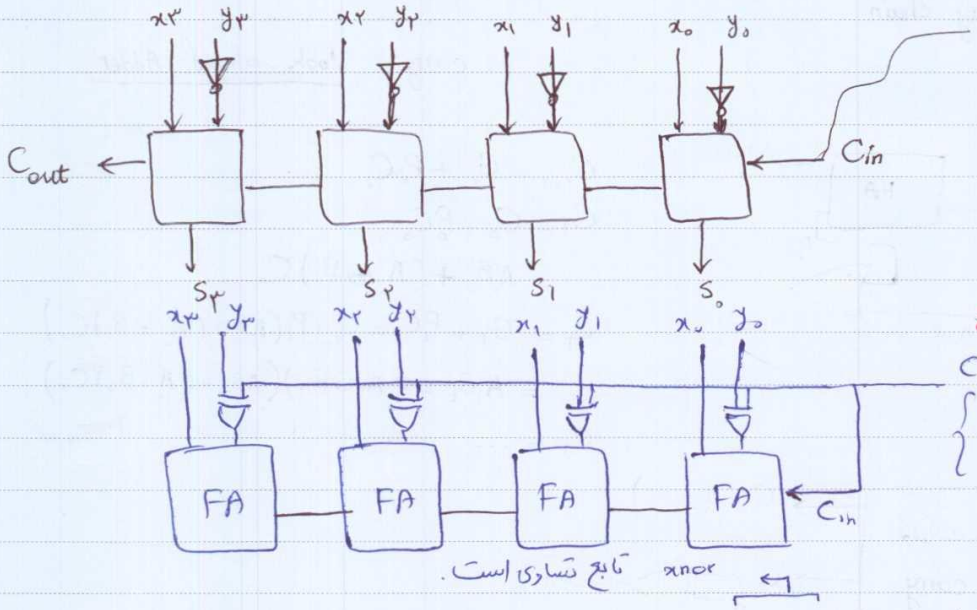




x	y	B_{in}	D	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = x \oplus y \oplus B_{in}$$
$$B_{out} = x'y + x'B_{in} + yB_{in}$$

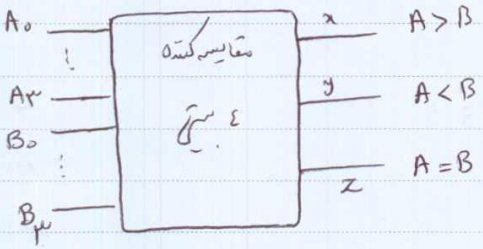
$$D = x - y = x + C_r(y) = x + \bar{y} + 1$$



جمع و هم تفریق کننده:

تابع تساوی است. xnor

مقایسه کننده:

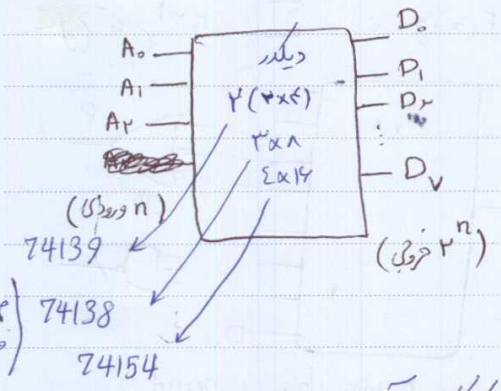


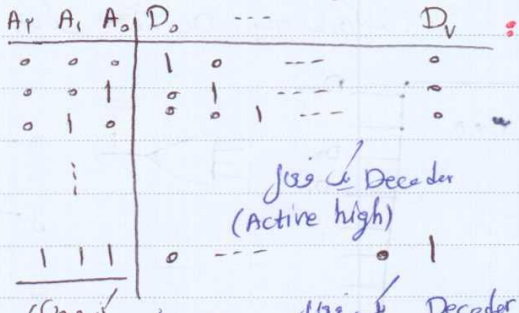
$$Z = (A_0 \odot B_0)(A_1 \odot B_1)(A_2 \odot B_2)(A_3 \odot B_3)$$
$$y = \bar{A}_3 B_3 + (A_3 \odot B_3)\bar{A}_2 B_2 + (A_3 \odot B_3)(A_2 \odot B_2)\bar{A}_1 B_1 + (\ \ )(\ \ )(A_1 \odot B_1)\bar{A}_0 B_0$$
$$x = A_3 \bar{B}_3 + (A_3 \odot B_3) A_2 \bar{B}_2 + \cdots$$

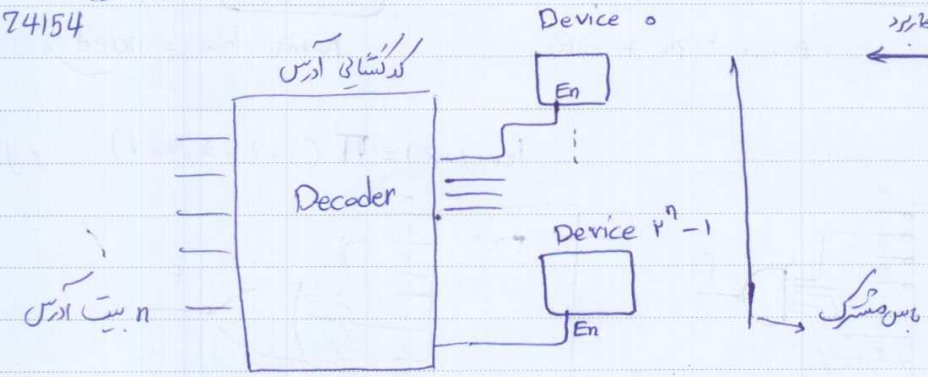
تمرین) تحقیق کنید آیا توابع بالا ساده‌تر می‌شوند یا خیر؟

رمزگشا

: Decoder







مثال جدول :

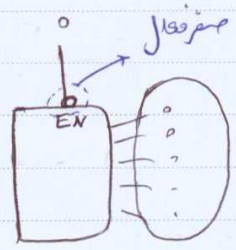
$$D_0 = \bar{A}_0\bar{A}_1\bar{A}_2$$

$$D_1 = \bar{A}_2\bar{A}_1 A_0$$

$$\vdots$$

$$D_7 = A_2A_1A_0$$

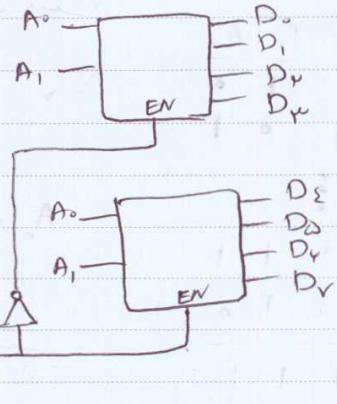
A_2	A_1	A_0	E_n	D_1	---	D_7
x	x	x	0	0	---	0
			1	1		
			1		1	
						1

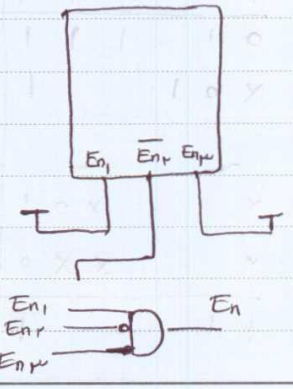


کافیست ورودی فعال‌ساز E_n رانیز در ورودی هرگیت and وارد کنیم.

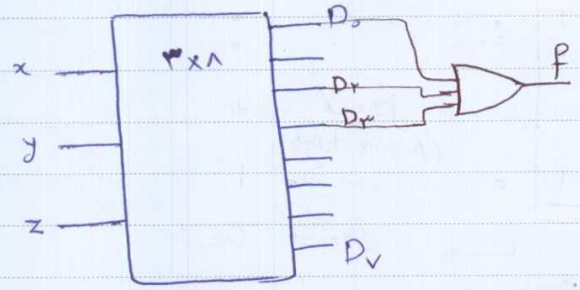
$3\times8 = 2(2\times4)$

A_2	A_1	A_0
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

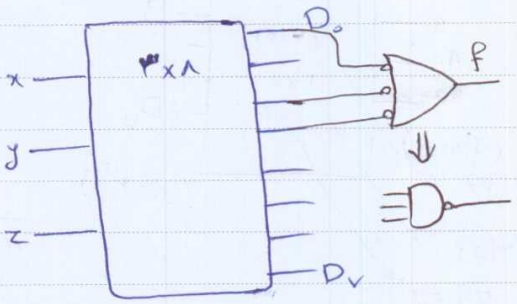




مثال) تابع $f(x,y,z)=\sum(0,2,3)$ را با یک Decoder پیاده سازی کنید.

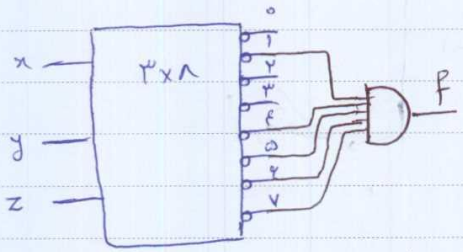


Active high + OR

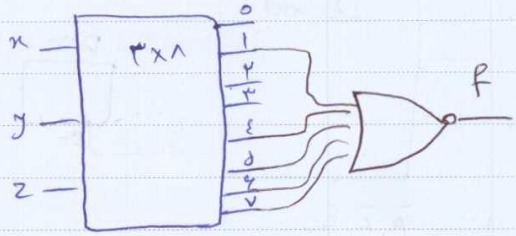


Active low + nand

مثال) $f(x,y,z)=\prod(1,4,5,6,7)$

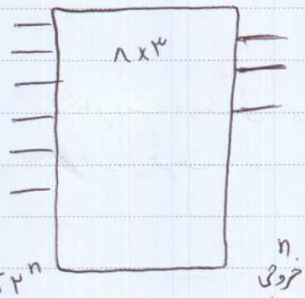


Active low + and



active high + nor

Encoder (رمزکننده، کدکننده)



D_7	---	D_0	A_2	A_1	A_0
0	---	0	1	0	0	0
	---	1	0	0	0	1
	---	1	0	0	1	0
1	0	0	0	---	0	1	1	1

don't care

don't care خوب نیست. ← محاط وجود

priority Encoder

با فرض هم‌زمان ورودی‌ها و خروجی‌ها فعال باشن

VΣ LS 1ΣV

۹	۸	۷	---	۱	۰	A_2	A_1	A_0
1	1	1	---	1	0	1	1	0
1	1	0	---	1	X	1	0	1
0	1	1	---	X	X	0	1	1
0	1	0	---	X	X	0	1	0
0	0	1	---	1	1	1	1	1

$A_0 = 1,3,5,7,9$

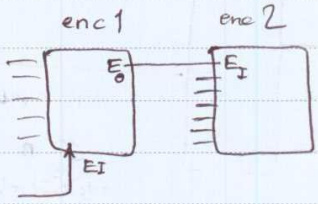
$= \overline{9} + 9\overline{8} + 9876\overline{5} + 98765\overline{43} + 9876543\overline{2}1$

$= \overline{9+8+876\overline{5}+8765\overline{43}+876543\overline{2}1}$

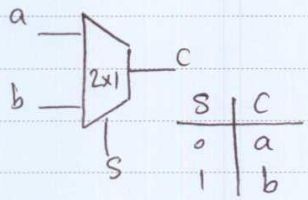
۷۴LS۱۴۸

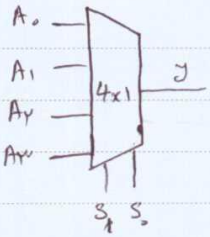
فعال سازی ورودی									وضعیت	
$\overline{EI}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	...	$\overline{7}$	A_2	A_1	A_0	G_S	E_O
1	X	X	...		X	1	1	1	1	1
0	1	1	...		1	1	1	1	1	0
0	X	X	...		0	0	0	0	0	1
⋮	X	X	...	0	1	0	0	1	0	1
						⋮			⋮	⋮
0	0	1	1	...	1	1	1	1		

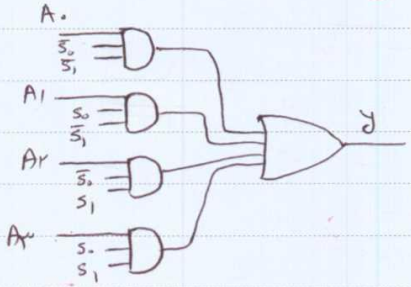
→ کنترل فعال پایین

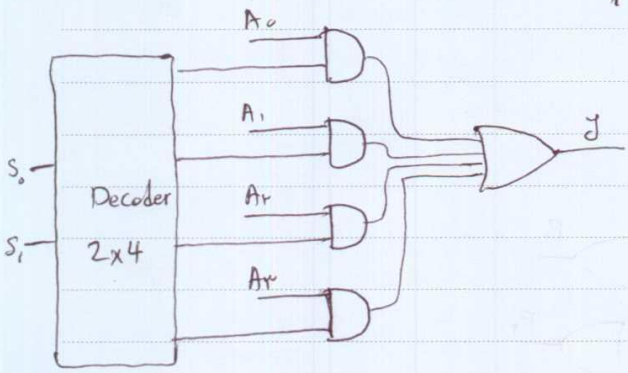


: MUltiplexer : مالتی پلکسر

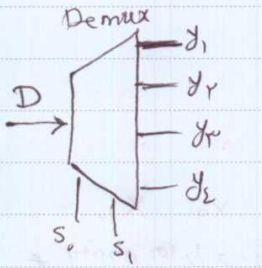




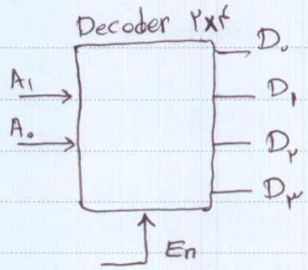


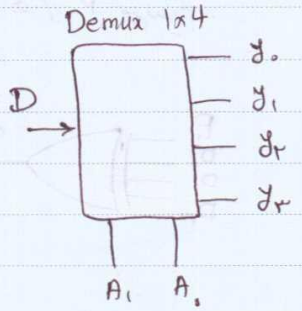


: Demultiplexer : دی مالتی پلکسر

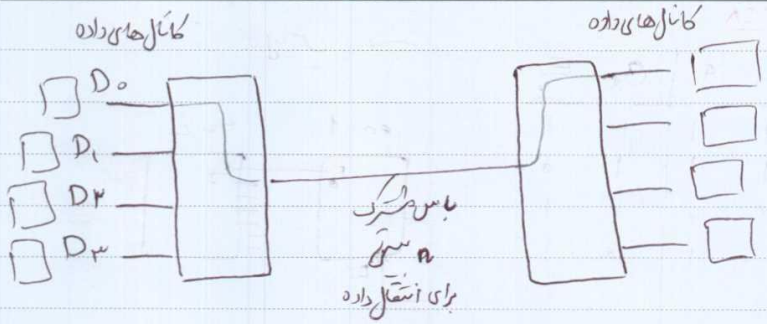


S_0	S_1	y_1	y_2	y_3	y_4
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D



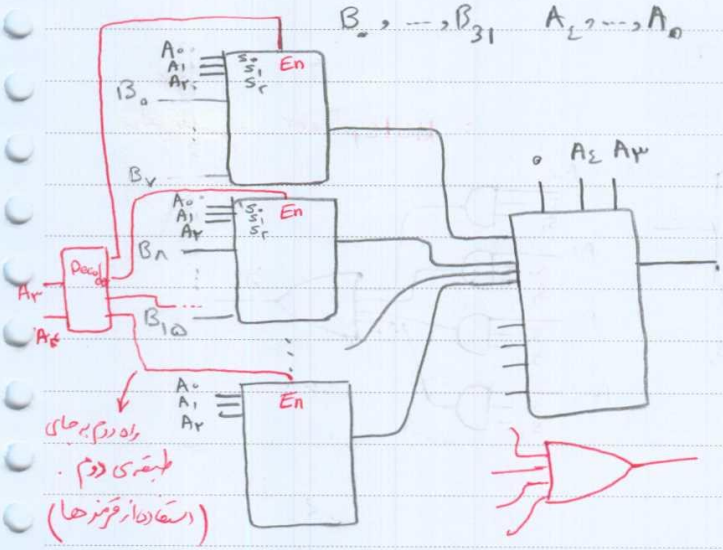


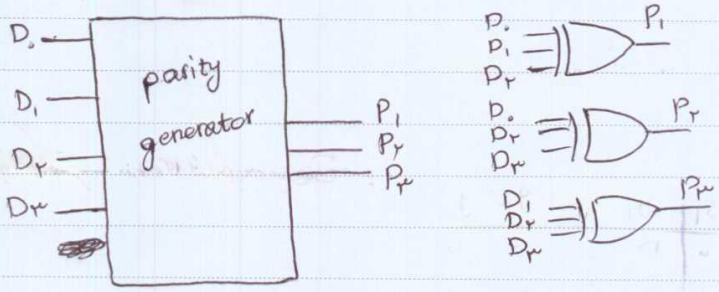
→ یک دی مالتی پلکسر



مثال: با چند MUX یک MUX 8x1 بسازید. MUX 32x1

$B_0 \to B_{31}$ $A_4 \to A_0$



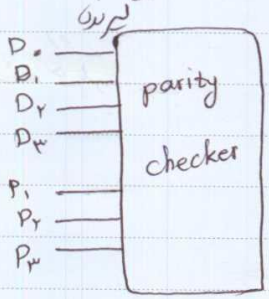


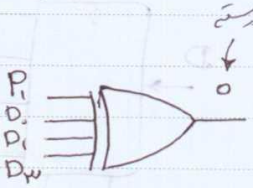
فرم استاندارد parity ها

۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ...

P_1 P_2 X_3 P_4 X_5 X_6 X_7 P_8 X_9 X_{10} ...

* parity ها را در مکان‌های توان‌های ۲ می‌نویسیم.





Subject : (۴۵)
Year . Month . Date . ( )

عناصر حافظه و منطق برنامه پذیر :

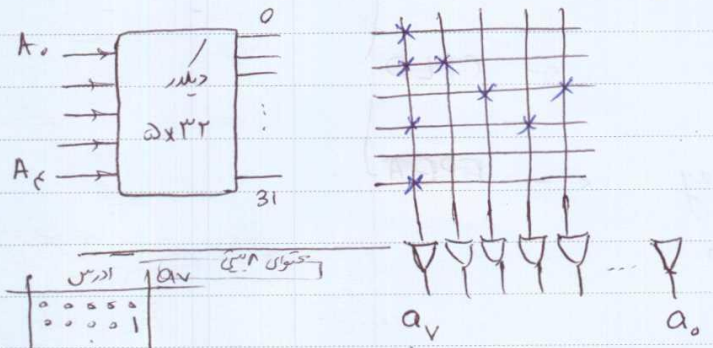
حافظه ها ← RAM : Random Access memory حافظه با دسترسی تصادفی — N خانه حافظه

- SRAM — static — ۶ ترانزیستور ـ ۴ ترانزیستور — هزینه مصرفی ↑ — سرعت ↑ — Cache / FPGA
- DRAM — dynamic — ۱ ترانزیستور + خازن — هزینه کم — فشرده — سرعت ↓ — Refresh دارد — حافظه معمولی کامپیوتر / DDR / Dual Data Rate

ROM : Read Only memory
- ROM
- PROM
- EPROM
- EEPROM
- Flash

۱) ROM

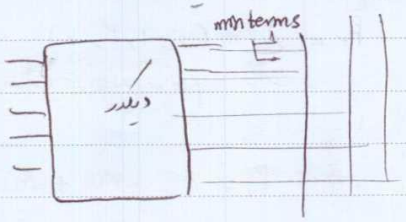
minterms



تعداد ورودی‌ها ← ۵

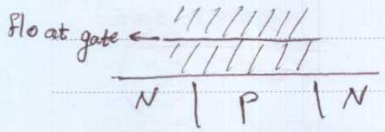
ظرفیت = $2^5 \times 8$ bit ← تعداد خروجی‌ها

۲) PROM : programmable ROM ← قابلیت یک بار برنامه‌ریزی دارد.



→ فیوز ← مناسب با جدول دیکدر کارخانه‌ها

۳) EPROM ← Erasable prog. ROM



programming
- drain = ۱۲v
- control gate = ۱۲v (hot-electron injection)
- source = ۰

سلول‌هایی که هدایت می‌کنند ← ۱
سلول‌هایی که هدایت نمی‌کنند ← ۰

Erasing → استفاده از اشعه UV

۴) EEPROM : Electrically Erasable Prog ROM ← استفاده می‌شود UV از

۵) Flash :

NOR-based } programming = جریان تونلی
NAND-based } Erasing = تونلی ″ ″

Subject: (۲۹)
Year. Month. Date. ( )

منطق برنامه پذیر : (PLD) ← Programmable logic device

ROM : دیکدر + آرایه OR برنامه پذیر

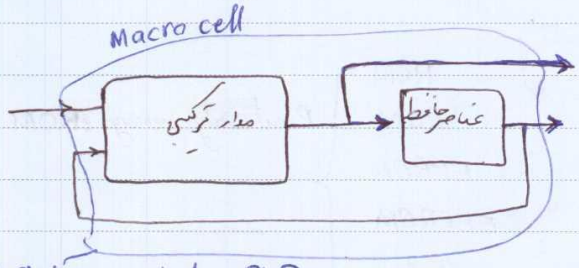
PLA : آرایه AND برنامه پذیر + " "

PAL : " AND آرایه + آرایه OR ثابت

GAL : هایی که قابلیت پاک شدن دارند PAL

Prog. logic Array ← PLA
Prog. Array logic PAL
Generic. " " GAL

ROM } ترکیبی

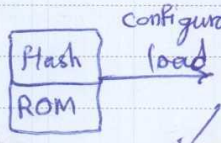


بلوک کلی PLD های ترکیبی

Macro cell ← simple PLD ← SPLD } ترکیبی (مدار ترکیبی + عناصر حافظه)

complex PLD ← CPLD

Field programmable Gate Array ← FPGA

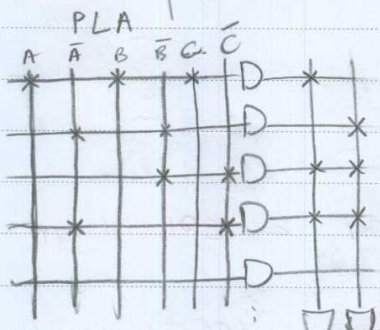


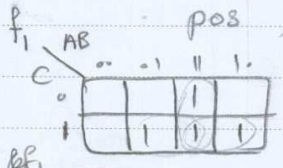
مثال) $f_1 = \sum m(3,5,6,7)$ را با استفاده از PLA طراحی کنید.

$f_2 = \sum m(0,1,3,7)$

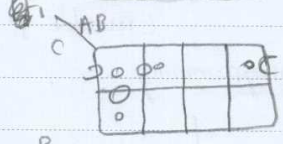
CAB (ABC)

۱) ساده سازی توابع به فرم sop , pos



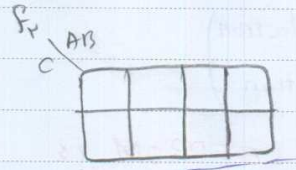


$f_1 = AB + BC + AC$

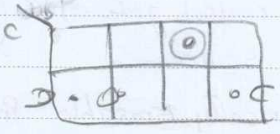


$f_1 = (A+C)(A+B)(B+C)$

$= \overline{\overline{AC} + \overline{A}\overline{B} + \overline{B}\overline{C}}$



$f_2 = \overline{ABC + \overline{B}\overline{C} + \overline{A}\overline{C}}$

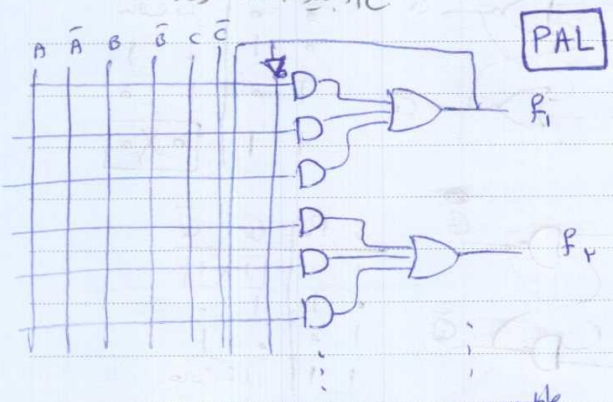


$f_2 = (\overline{A}+\overline{B}+C)(A+\overline{C})(B+\overline{C})$

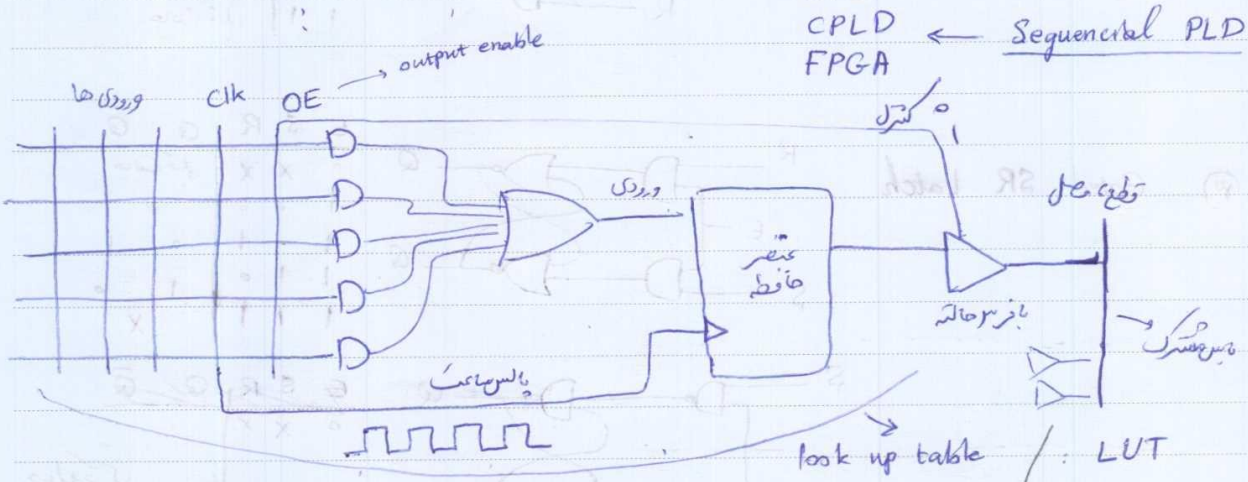
$= \overline{AB\overline{C} + \overline{A}\overline{C} + \overline{B}\overline{C}}$

حاصل‌ضرب‌ها	ABC	f_1	f_2
ABC	۱۱۱	۰	۱
$\bar{A}\bar{B}$	۰۰–	۱	۰
$\bar{B}\bar{C}$	–۰۰	۱	۱
$\bar{A}\bar{C}$	۰–۰	۱	۱
		C	T

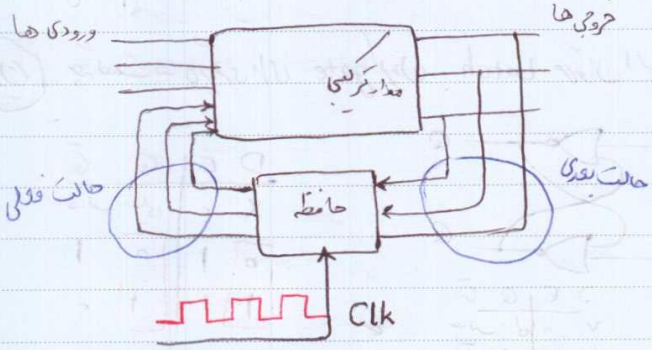
تابع f_1 نیز NOT شود.



SPLD
CPLD ← Sequencial PLD
FPGA



look up table : LUT

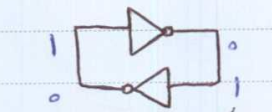


دیگر خط نقطه‌چین نداریم است.

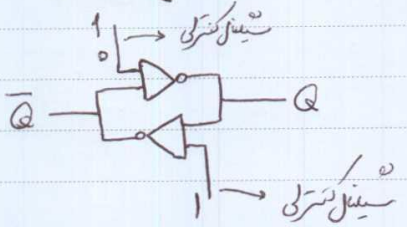
حساس به Clk { لبه / سطح }

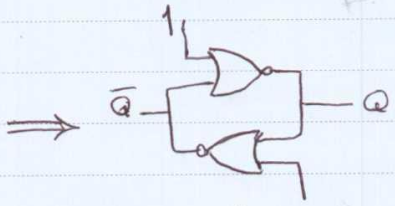
حافظه‌ها { Latch: نگهدارنده‌ها — حساس به سطح
Flip flop: فلیپ‌فلاپ‌ها — حساس به لبه }

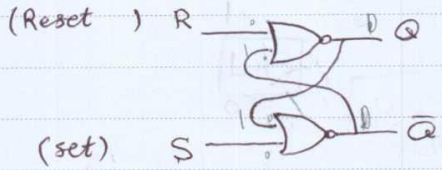
Subject: (۲۸)
Year. Month. Date. ( )



→ یک بیت را می‌تواند در خود ذخیره کند (دو حالت مختلف)

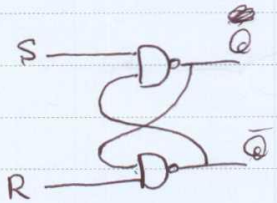






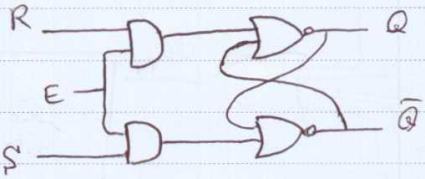
S	R	Q	$\bar{Q}$
0	0	حالت قبلی	
0	1	0	1
1	0	1	0
1	1	0	X 0

① SR Latch



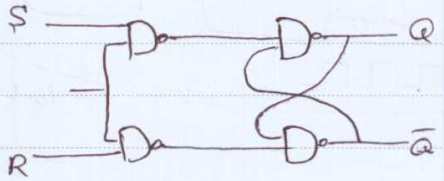
S	R	Q	$\bar{Q}$
0	0	1	X 1
0	1	1	0
1	0	0	1
1	1	حالت قبلی	

② $\bar{S}\bar{R}$ Latch
حساس به سطح صفر



E	S	R	Q	$\bar{Q}$
0	X	X	حالت قبلی	
1	0	0	〃	
1	0	1	0	1
1	1	0	1	0
1	1	1	X	

③ Gated SR Latch



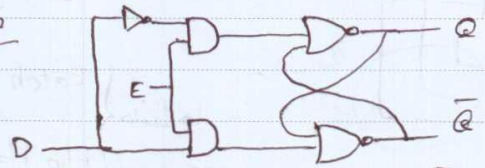
E	S	R	Q	$\bar{Q}$
0	X	X	حالت قبلی	

جدول درستی مثل حالت قبل

* تمرین: چه تغییراتی برای gate کردن Latch می‌توان استفاده کرد (جدول درستی چه تغییری می‌کند؟)؟

④ D Latch

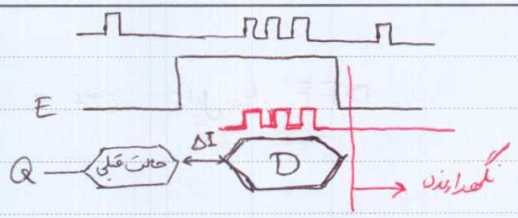
مشکل حالت X حل شد.



D	E	Q	$\bar{Q}$
X	0	حالت قبلی	
0	1	0	1
1	1	1	0

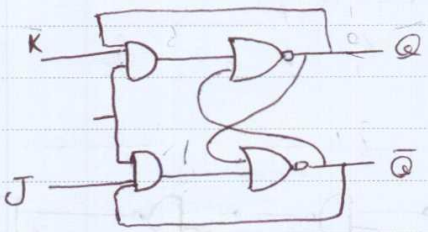
D	E	Q	$\bar{Q}$
X	0	حالت قبلی	
d	1	d	$\bar{d}$

P4PCO



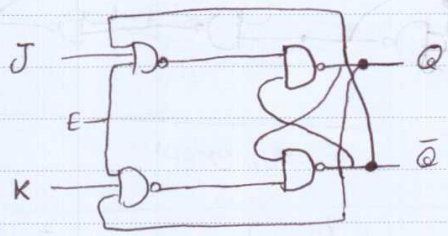
(۵) Jk Latch

S R



E	J	K	Q	Q̄
0	X	X	حالت قبلی	
1	0	0	حالت قبلی	
1	0	1	0	1
1	1	0	1	0
1	1	1	toggle	

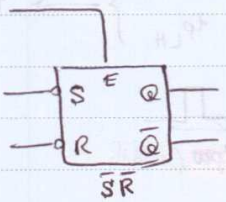
toggle = not حالت قبلی



حمل درستی

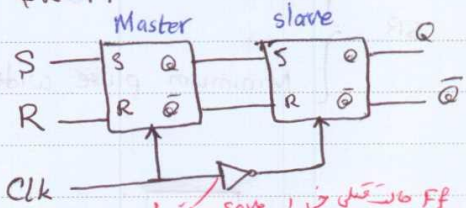
Flip_Flops

- Master_slave
  - از ابتدا حساس به لبه ساخته می‌شوند

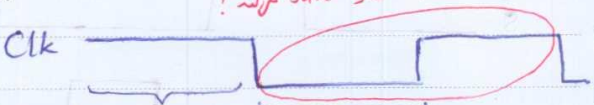


Nand-based Latch

① SR-FF → حساس به لبه‌ی پایین رونده



حالت قبلی خود را save می‌کند! FF

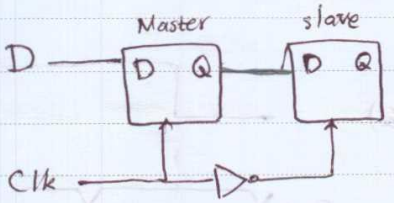


initialize اول در حالت SR — Latch اول SR

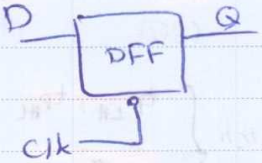
Latch دوم در حالت SR — initialize دوم SR

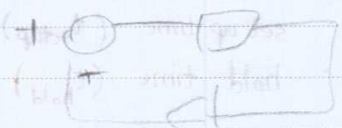
* با قرار دادن یک not سر clk می‌توان FF حساس به لبه‌ی بالا رونده ساخت.

② DFF → حساس به لبه‌ی پایین رونده



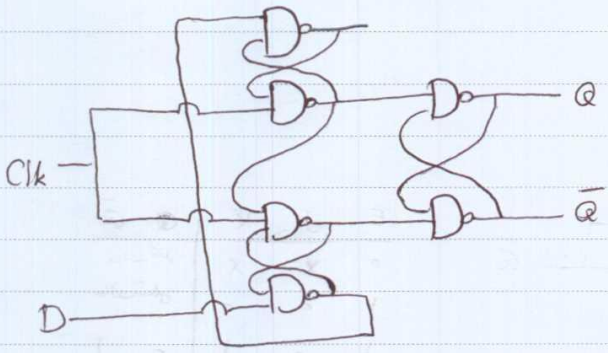
≅



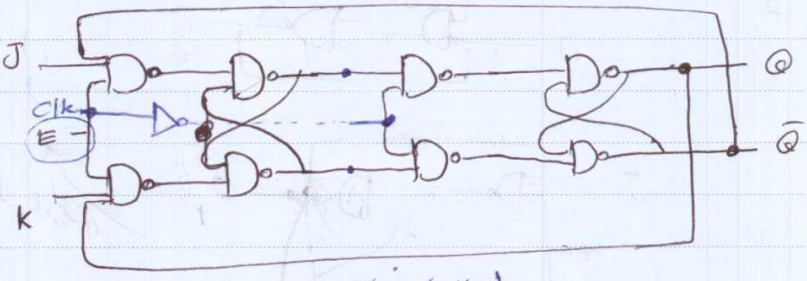


Subject: (۳۰)
Year. Month. Date. ( )

* تمرین : تکمیل مدار DFF روبرو



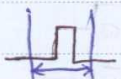
(۳۱) Jk-ff



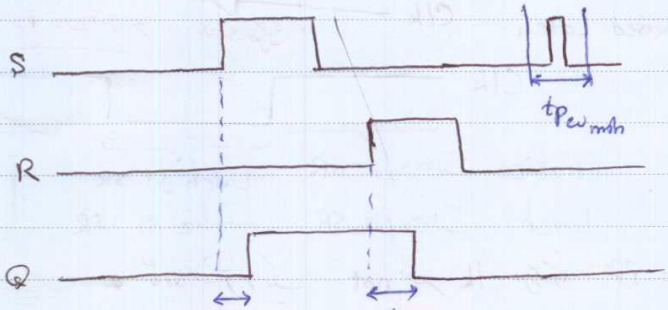
تمرین روبرو .

Latch SR { progration delay $tp\begin{cases} t_{P_{HL}} \\ t_{P_{LH}} \end{cases} \longrightarrow t_p = \frac{t_{P_{HL}} + t_{P_{LH}}}{2}$

Minimum pulse width



t_{pw} (width

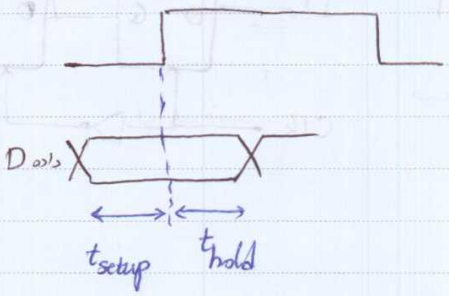


Gated Latch { $t_{P_{LH}}, t_{P_{HL}}$ (Clk → Q)

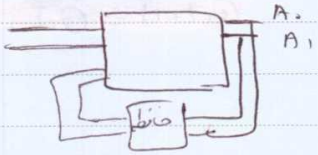
// (D → Q)

set up time (t_{setup})

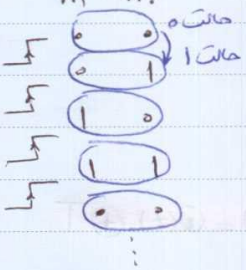
hold time (t_{hold})

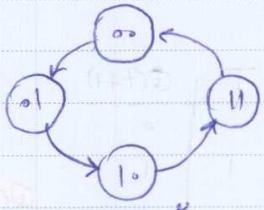


FF { t_{PLH} , t_{PHL} (C → Q)



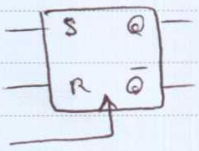
شمارنده ۲ بیتی





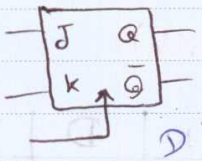
دیاگرام حالت

SR-ff



S	R	Q	Q̄
0	0	حالت قبلی	
0	1	0	1
1	0	1	0
1	1	X	

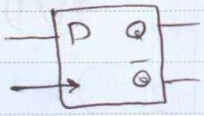
JK-ff



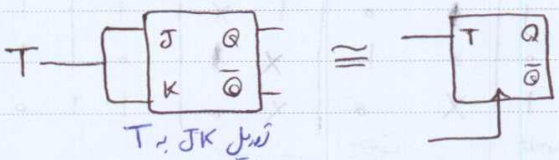
J	k	Q	Q̄
0	0	حالت قبلی	
0	1	0	1
1	0	1	0
1	1	toggle	

D T

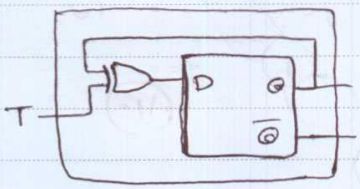
D-ff



D	Q
0	0
1	1



تبدیل JK به T



تبدیل D به T

جدول مشخصه

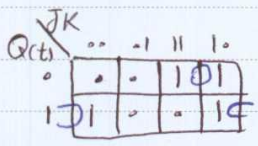
Q(t)	S	R	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

Q \ SR	00	01	11	10
0	0	0	X	1
1	1	0	X	1

$Q(t+1) = S + Q\bar{R}$ معادله مشخصه SR

کاربرد: تحلیل مدارهای ترتیبی

Q(t)	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0



$$Q(t+1) = \bar{Q}J + Q\bar{K}$$

Q(t)	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

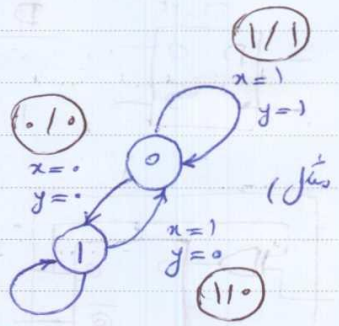
$$Q(t+1) = D$$

Q(t)	T	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

$$Q(t+1) = Q(t) \oplus T$$

جدول تحریک (Ecitation Table)

Q(t)	Q(t+1)	S	R	J	K	D	T → toggle
0	0	0	X	0	X	0	0
0	1	1	0	1	X	1	1
1	0	0	1	X	1	0	1
1	1	X	0	X	0	1	0



مثال)

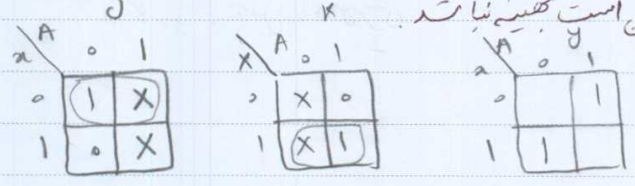
input: x — ورودی‌ها و حالت فعلی A(t)	x	حالت: A — حالت بعدی A(t+1)	خروجی: y — جدول تحریک J	K	خروجی‌ها y
0	0	1	1	X	0
0	1	0	0	X	1
1	0	1	X	0	1
1	1	0	X	1	0

2^{m+n}

m ورودی

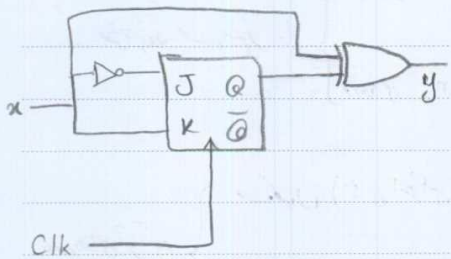
n فلیپ فلاپ

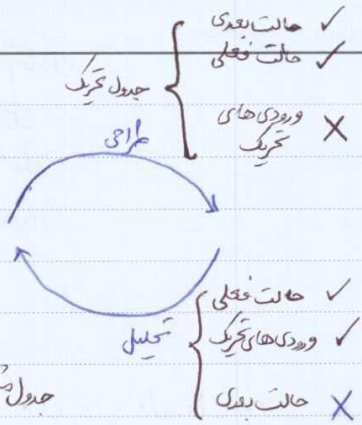
* هدف از جدول فوق طراحی مداری ترکیبی است که ورودی فلیپ فلاپ‌ها را تحریک کند تا خروجی مدار مطلوب باشد.
* با هر فلیپ فلاپی می‌شود ساخت فقط ممکن است بهینه نباشد.

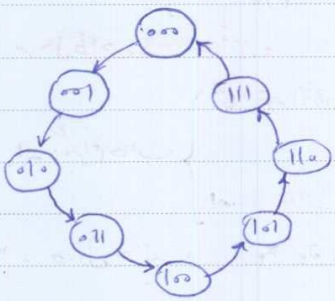


$J = \bar{x}$ $\quad K = x$

Subject: (۳۳)
Year. Month. Date. ( )

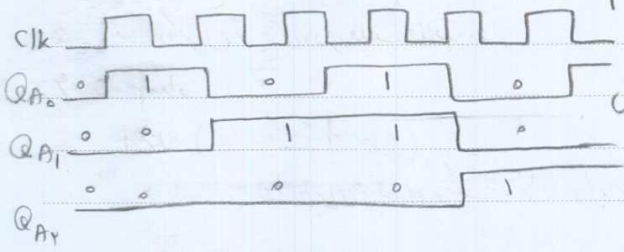


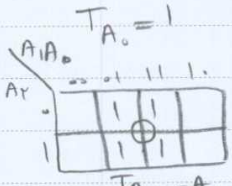


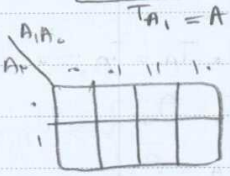


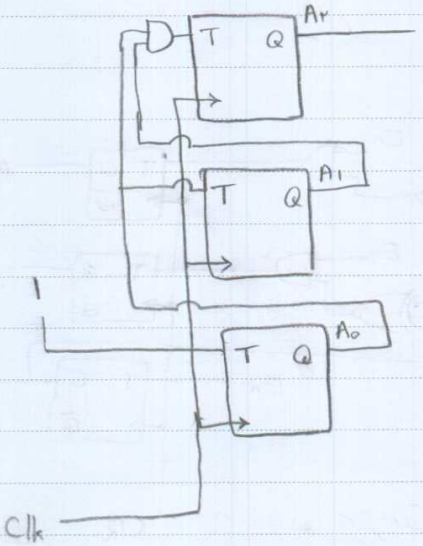
مثال) شمارنده ۳ بیتی سنکرون با فلیپ فلاپ T

حالت فعلی			حالت بعدی			جدول تحریک		
A_2	A_1	A_0	A_2	A_1	A_0	T_{A_2}	T_{A_1}	T_{A_0}
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1









تمرین جلسه آینده
شمارنده سنکرون با ورودی Enable و ورودی Direction
Enable: ۰ عدم شمارش، ۱ شمارش
Direction: ۰ نزولی، ۱ صعودی

مدارهای ترتیبی
- سنکرون (همه اجزای حافظه باهم سنکرون هستند.)
- آسنکرون (Clk ندارند) pulse mode (سنکرون نیستند ولی Clk موجود است.)

مراحل طراحی مدارهای ترتیبی:
۱- دیاگرام حالت
- کاهش تعداد حالات
- انتساب حالت (نحوه ی کدکردن حالت ها)

۲- جدول حالت

A	B	x	A^*	B^*	J_A	K_A	J_B	K_B
حالات فعلی		ورودی ها	حالات بعدی		جدول تحریک			
۰	۰	۰	۰	۱				
		۱						
	۱							
۱	۰							
	۱							

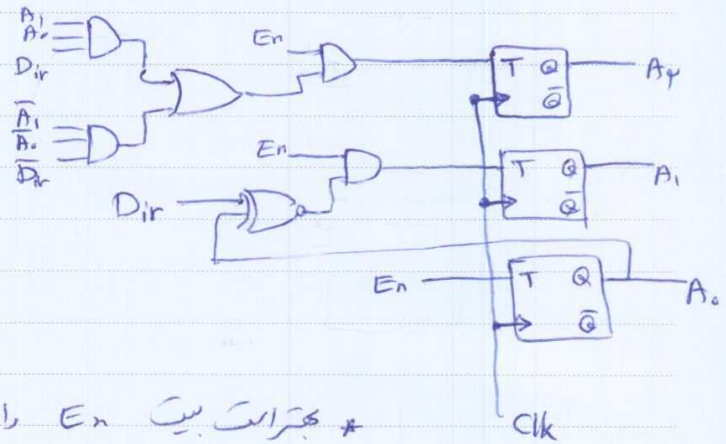
۳- انتخاب فلیپ فلاپ ها
۴- تعیین توابع ورودی فلیپ فلاپ ها از روی جدول تحریک
۵- ساده کردن توابع ورودی فلیپ فلاپ ها
۶- رسم مدار

حل تمرین (شمارنده سنکرون فوق)
* یک بار جدول را برای صعودی و یک بار برای نزولی می کشیم. دلیلم:

$En = 0 \Rightarrow T_{A_2} = T_{A_1} = T_{A_0} = 0$

$Dir = 1 \Rightarrow T_{A_1} = A_0$
$Dir = 0 \Rightarrow T_{A_1} = \bar{A}_0$
$\Rightarrow T_{A_1} = Dir \odot A_0$

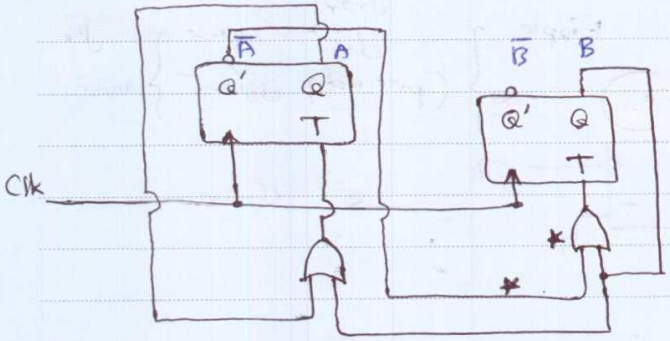
$T_{A_2} = Dir A_1 A_0 + \overline{Dir}\ \bar{A}_1 \bar{A}_0$



* بهتر است بیت En را در جدول منظور کنیم و حالاتی آن را در نظر بگیریم.

Subject : (۳۵)
Year . Month . Date . ( )

مثال ۱

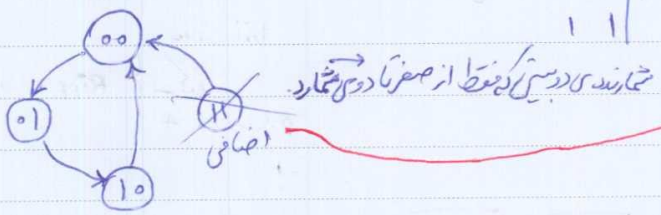


$T_A = A + B$

$T_B = \bar{A} + B$

A	B	T_A	T_B	A^*	B^*
0	0	0	1	0	1
0	1	1	1	1	0
1	0	1	0	0	0
1	1	X	X	X	X

معادلات ورودی ← جدول مشخصه



تحلیل مدارهای ترتیبی چ

۱ـ نوشتن معادلات ورودی

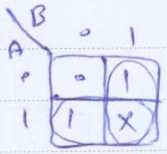
۲ـ جدول حالت

۳ـ رسم دیاگرام حالت

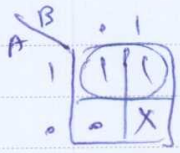
می‌نویسیم

A	B	A^*	B^*	T_A	T_B
0	0	0	1	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	X	X	X	X

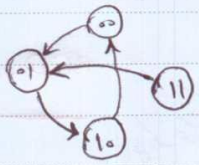
X ← 0 یا 1



$T_A = A + B$

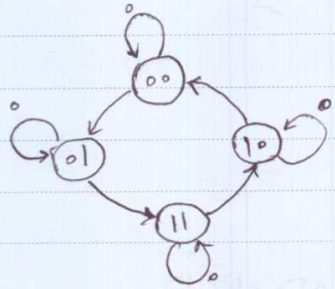


$T_B = \bar{A}$

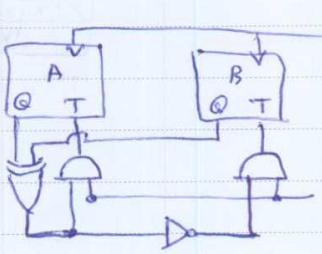


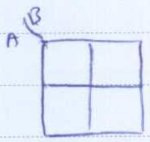
→ مدار ساده‌تری می‌دهد. ⟸ or * نیازی نیست رسم کنیم * مستقیماً به T وصل می‌شود.

مثال ۲ شمارنده‌ی ۲ بیتی گری با ورودی Enable



A	B	A^*	B^*	T_A	T_B
0	0	0	1	0	1
0	1	1	1	1	0
1	0	1	0	1	0
1	1	1	0	0	1



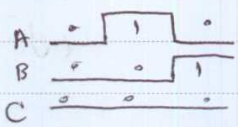


$T_A = A \oplus B$

$T_B = \overline{A \oplus B}$

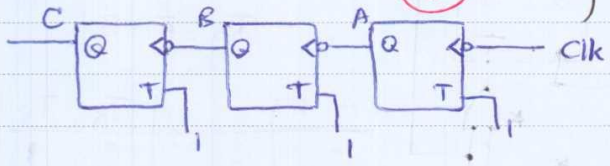
P4PCO

$T_A = E(A \oplus B)$ $T_B = E(\overline{A \oplus B})$

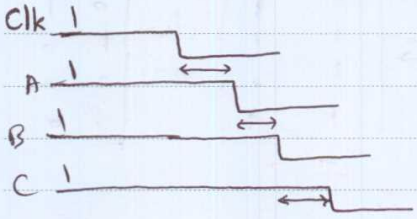


مثال شمارنده { سنکرون ✓ { صعودی / نزولی
آسنکرون { (pulse mode)

Ripple t_{PHL} (C-Q)

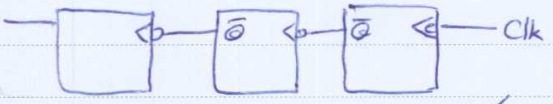


→ صعودی



7 6 4 0

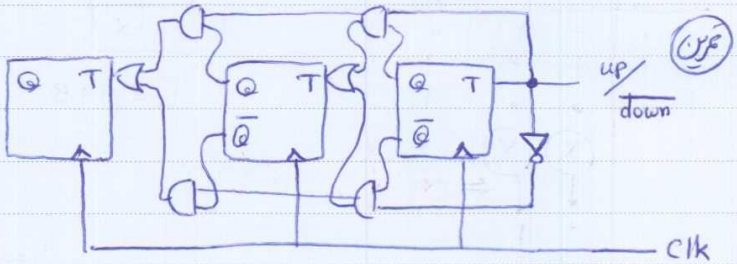
شمارنده Ripple { ـ حالت گذرا
ـ کند
4 مدار ساده



→ نزولی

تمرین تحقیق کنید ! این مدار چطور عمل می‌کند ؟

تمرین

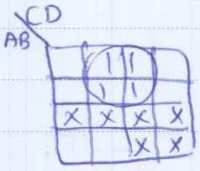


شمارنده BCD

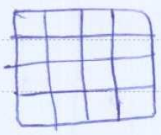
A	B	C	D	A^+	B^+	C^+	D^+	T_A	T_B	T_C	T_D	Carry
0	0	0	0	0	0	0	1	0	0	0	1	0
1	0	0	1	0	0	0	0	1	0	0	1	1
1	0	1	0	X				X				X
1	0	1	1									
1	1	0	0									
1	1	0	1									
1	1	1	0									
1	1	1	1									

پایانی یک می‌شود که شمارنده از آخر به اول برود.

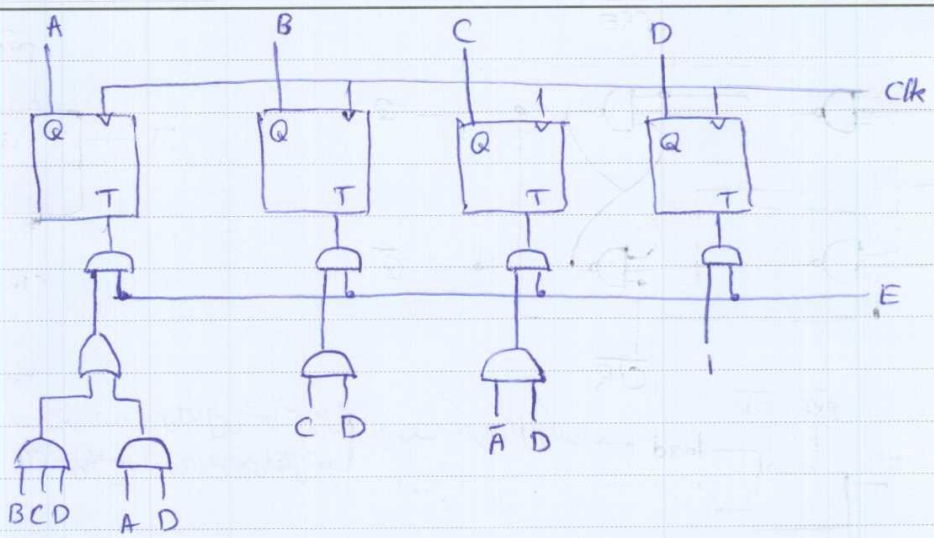
$T_D = 1$,



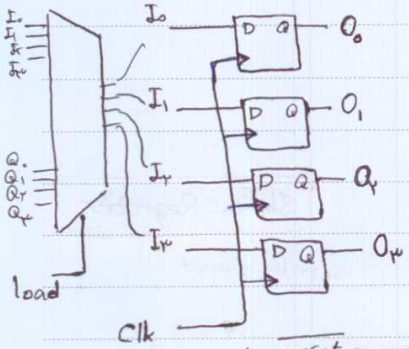
$T_C = \bar{A}D$

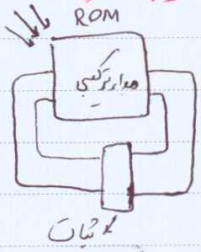


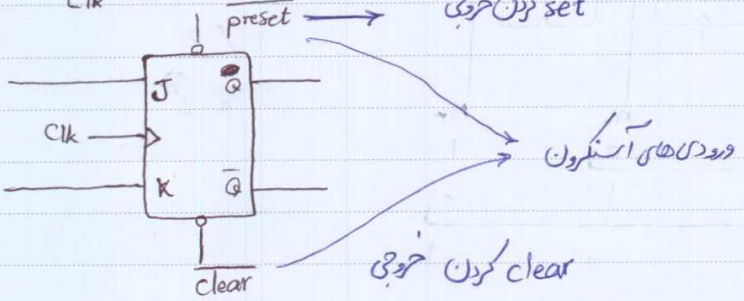
$T_B = CD$, $T_A = AD + BCD$

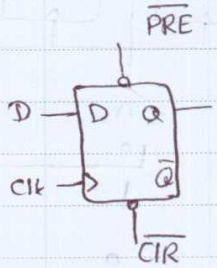


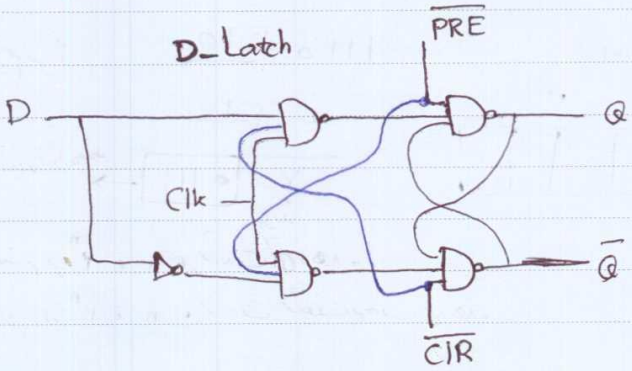
ثبات ها (Register) : آرایه ای از فلیپ فلاپ ها



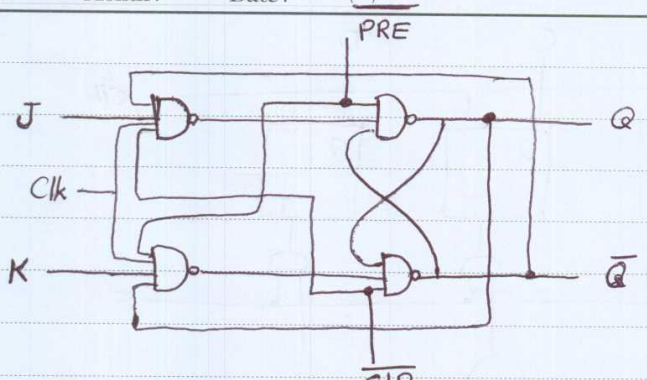


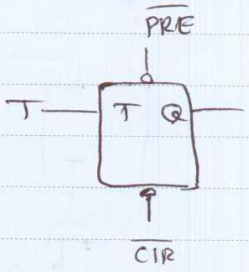


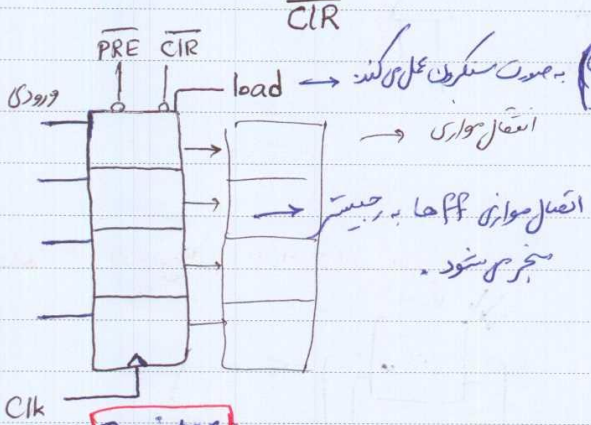




Subject: (۳۸)
Year. Month. Date. ( )







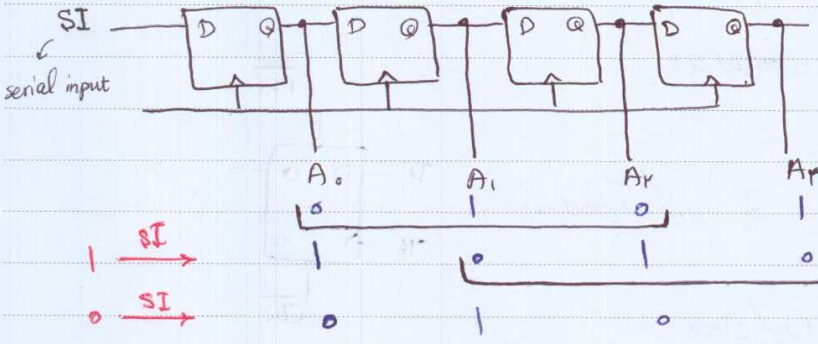
اول load یک می‌شود بعد ورودی اعمال می‌شود و سپس منتظر Clk می‌ماند تا ورودی به خروجی اعمال شود

به صورت سنکرون عمل می‌کند

انتقال موازی

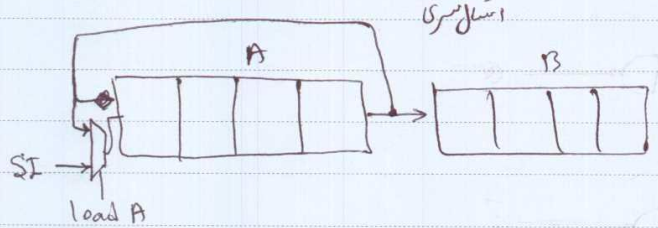
اتصال موازی FF ها به رجیستر منجر می‌شود.

Shift – Register



$/ 2^n$

$* 2^n$



انتقال سری

مثال) تقسیم بر ۲

1110 | 10

12 | 2

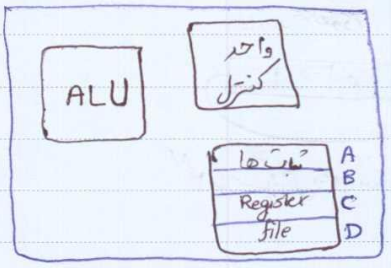
7 | 0111

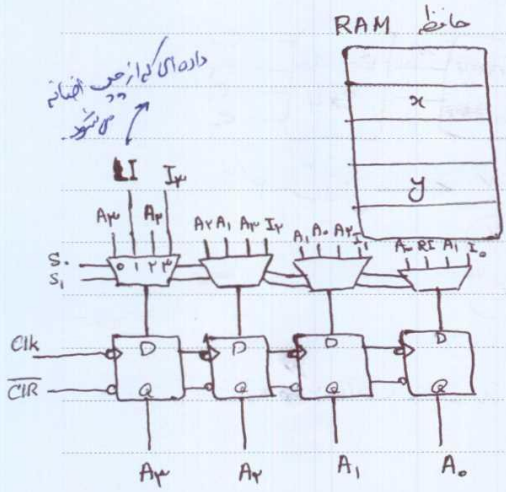
یک بار به راست شیفت داده شده.

برابند

* برای تقسیم بر 2^n ، n بار شیفت می‌دهد.
* برای ضرب در 2^n ، n بار شیفت به چپ می‌دهد.

پردازنده :





S_1	S_0	عملیات
0	0	حالت قبلی
0	1	شیفت به راست
1	0	" " چپ
1	1	بارگذاری موازی

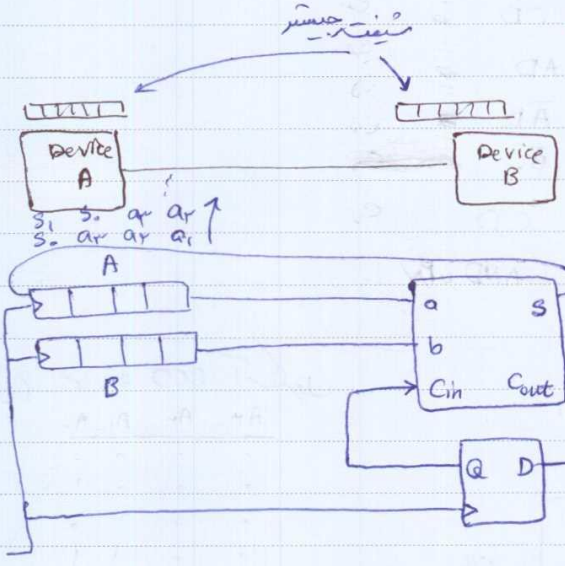
Universal shift – Register

طراحی مدارهای ترتیبی سنکرون

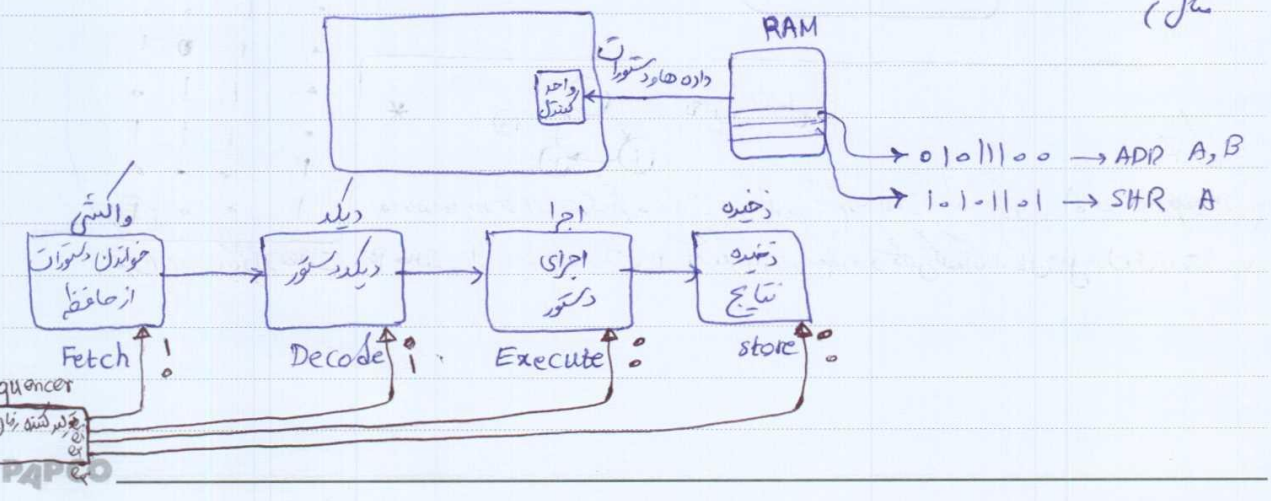
شمارنده ها — Binary (صعودی / نزولی) BCD

رجیسترها — AX BX CX DX (۱۶ بیتی)

شیفت رجیسترها

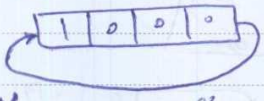


مثال) جمع کننده سری



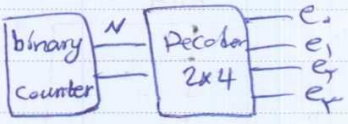
Subject: (۴۰)
Year. Month. Date. ( )

۱) شمارنده حلقوی — پیاده سازی



FF تا 2^N ← 2^N تا سیگنال زمان بندی

۲)

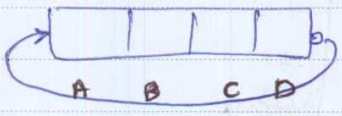


2^N سیگنال ← N سری FF

AND تا 2^N

~~NOT تا N~~

۳) شمارنده جانسون



FF تا N → $2N$ حالت

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

۸ حالت { X X X X

$\bar{A}\bar{D} = e_0$

$A\bar{B} = e_1$

$B\bar{C} = e_2$

$C\bar{D} = e_3$

$AD = e_4$

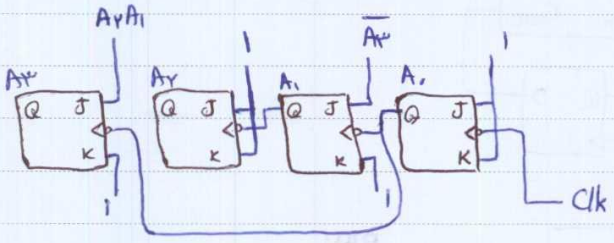
$\bar{A}B = e_5$

$\bar{B}C = e_6$

$\bar{C}D = e_7$

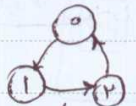
AND تا ۳ N

مثال) شمارنده BCD آسنکرون



A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

تعیین ناکامل



*

کلاک

* باید شمارنده جوری باشد که اگر به حالت غیرمطلوب رفتیم به حالات مطلوب برگردیم (با یک یا چند کلاک)

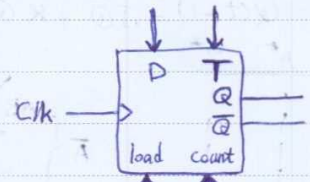
تمرین) دیاگرام حالت ۱۶ حالته مدار شمارنده BCD آسنکرون را رسم کرده و مشخص کنید آیا امکان قفل شدن شمارنده وجود دارد؟

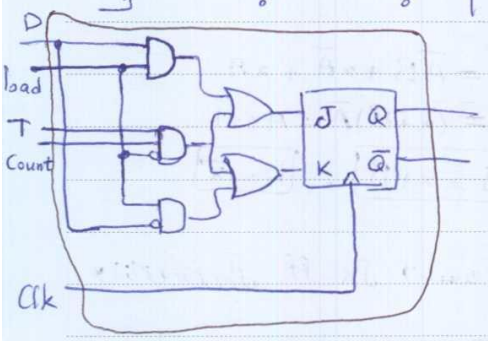
P4PCO

Subject: (۴۱) بارگذاری موازی
Year. Month. Date. ( )

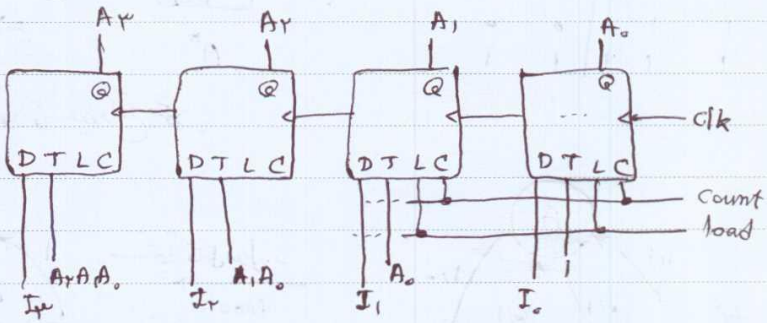
Clk	load	count	function	
↑	1	X	بارگذاری	D-FF
↑	0	1	ادامه شمارش	TFF
↑	0	0	توقف شمارش	

مثال) شمارنده با امکان بارگذاری موازی

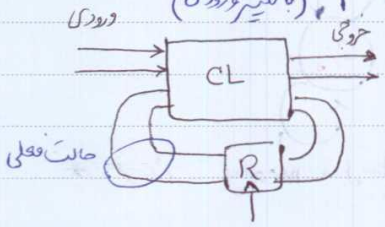




* این نوع طراحی لزوماً بهینه نیست، برای راحتی این طوری طراحی کردیم.



* خروجی می‌تواند مستقل از clk تغییر کند. (با تغییر ورودی)

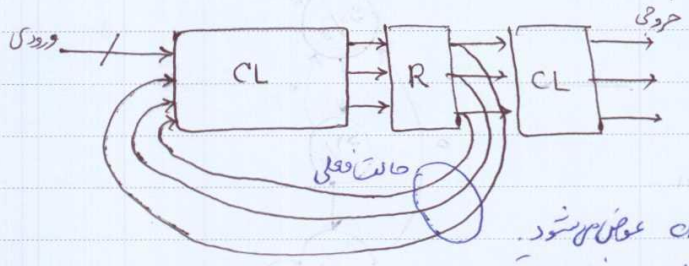


مدل‌های ترتیبی سنکرون

(Mealy) ← خروجی تابعی از ورودی و حالت فعلی

خروجی فقط تابعی از حالت فعلی

(Moore)



* خروجی فقط با clk عوض می‌شود.
* امکان glitch در خروجی حداقل است.

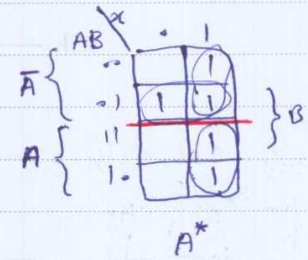
Mealy
+ حالت کلی ← انعطاف پذیرتر ← مدار بهینه
− تأثیر آنی ورودی در خروجی ← glitch

Moore
− حالت خاص ← انعطاف پذیری کم‌تر ← مدار غیربهینه یا تعداد حالات بیش‌تر
+ خروجی سنکرون با clk

مثال) پیاده سازی مدارهای ترتیبی سنکرون با استفاده از معادلات حالت:

JK-ff $Q(t+1) = J\bar{Q} + \bar{K}Q$

A	B	x	A^*	B^*
0	0	0	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

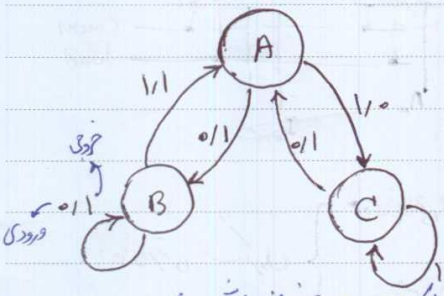


$A^* = \bar{A}B + x\bar{A} + xA$

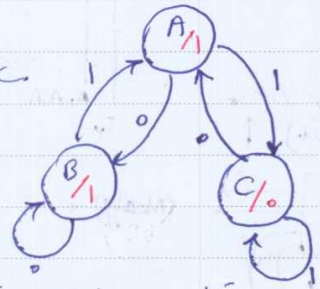
$= (x+B)\bar{A} + A(x)$

$\Rightarrow \boxed{J = x+B}$, $\boxed{K = \bar{x}}$

* از این روش برای JK-ff استفاده می‌شود تا مفید واقع شود.

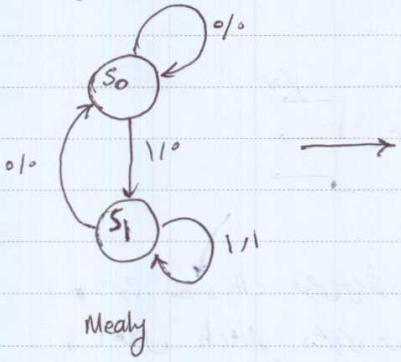


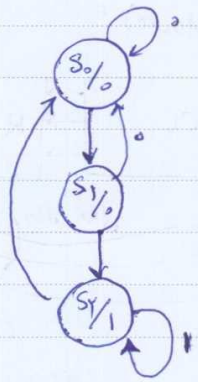
مستقیماً قابل تبدیل به Moore



* هر مدار Mealy قابل تبدیل به Moore هست. (ممکن است اضافه کردن حالت نیاز باشد.)

مثال)





طراحی مدارهای ترتیبی سنکرون:

۱- تعیین نوع مدار (میلی / مور)

۲- رسم دیاگرام حالت.

۳- ساده سازی دیاگرام حالت. state reduction

۴- انتساب حالت state assignment

۵- رسم جدول حالت

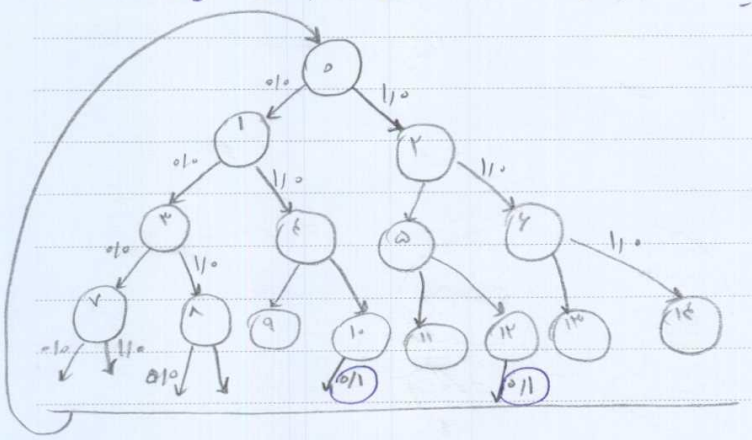
۶- ساده سازی توابع تحریک

۷- رسم مدار

Subject: (۴۳)
Year. Month. Date. ( )

مثال) مداری طراحی کنید که با ورودی x و ترتیب ۰۱۱۰ یا ۱۰۱۰ در ورودی با خروجی ۱ تشخیص داده شود.

۱- مدار میلی



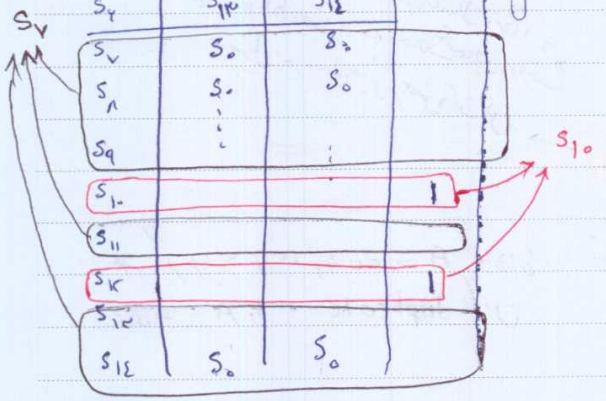
ساده سازی دیاگرام حالت:

۱- Row matching

۲- Implication Table

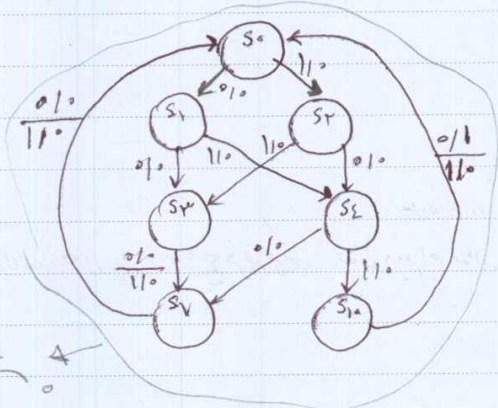
* حالات معادل: دو حالت که به ازای تمام حالات ورودی‌ها، حالت بعدی یکسان و خروجی یکسان دارند، معادل‌اند.

حالت فعلی	حالت بعدی x=0	حالت بعدی x=1	خروجی x=0	خروجی x=1
S_0	S_1	S_2		
S_1	S_3	S_4		
S_2	S_5	S_6		
S_3	S_7	S_8		
S_4	S_9	S_{10}		
S_5	S_{11}	S_{12}		
S_6	S_{13}	S_{14}		
S_7	S_0	S_2		
S_8	S_0	S_0		
S_9	⋮	⋮		
S_{10}			1	
S_{11}				
S_{12}			1	
S_{13}				
S_{14}	S_0	S_0		



حالت فعلی	حالت بعدی x=0	حالت بعدی x=1	خروجی x=0	خروجی x=1
S_0	S_1	S_2		
S_1	S_3	S_4		
S_2	S_4	S_3		
S_3	S_5	S_5		
S_4	S_5	S_{10}		
S_5	S_5	S_{10}		
S_6	S_5	S_5		
S_7	S_0	S_0		
S_{10}	S_0	S_0	1	

حالت فعلی	حالت بعدی x=0	حالت بعدی x=1	خروجی x=0	خروجی x=1
S_0	S_1	S_2		
S_1	S_3	S_4		
S_2	S_4	S_3		
S_3	S_5	S_5		
S_4	S_5	S_{10}		
S_5	S_0	S_0		
S_{10}	S_0	S_0	1	



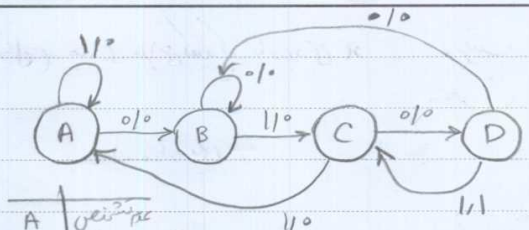
۱۵ حالت ← ۴ ff

۷ حالت ← ۳ ff

Subject: (۴۴)
Year. Month. Date. ( )

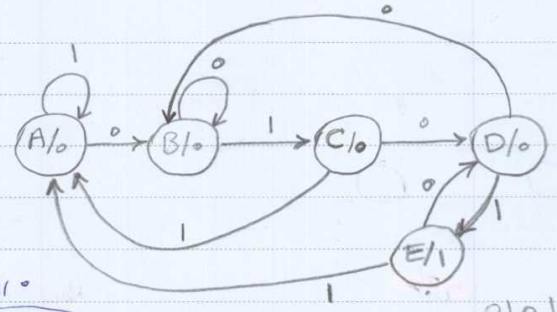
مثال) طراحی مدار تشخیص کد 1101 .

Mealy



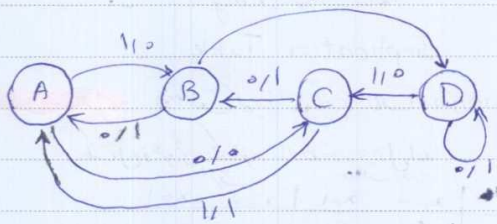
A	عدم تشخیص
B	تشخیص 0
C	01 " / 0101
D	010 "

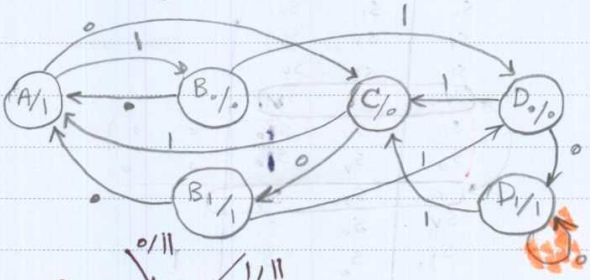
Moore ⟶



0101010

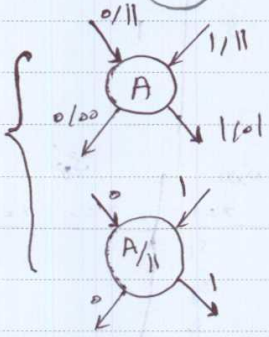
مثال) تبدیل مدار میلی به مور .





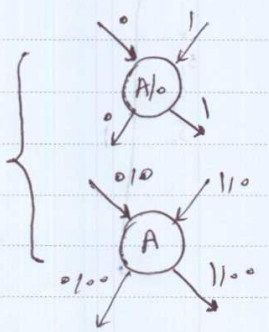
برای تمام حالات ورودی حالت بعدی یکسان خروجی یکسان داشته باشد (طبیعتاً) ← به حالت خاصی وارد میشود

در تبدیل مور به میلی بیش‌ترین کاهش در تعداد این



قوانین تبدیل میلی به مور

⇒ * اگر خروجی های فلش‌هایی که به A وارد می‌شوند متفاوتند ، برای تعداد خروجی ها A را duplicate کن .



قوانین تبدیل مور به میلی

⇒ ساده سازی دیاگرام

* روش پیاده سازی مدارهای میلی و مور هیچ فرقی نمی‌کند . فقط دیاگرام حالت عوض می‌شود .

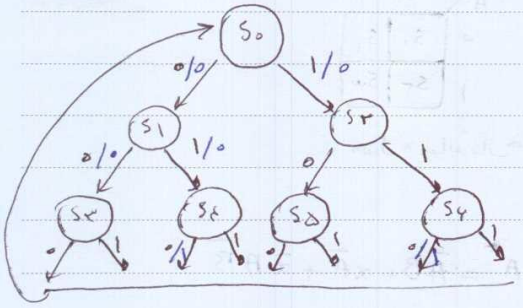
P4PCO

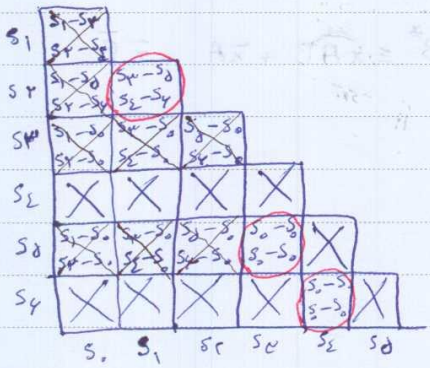
ساده سازی دیاگرام حالت

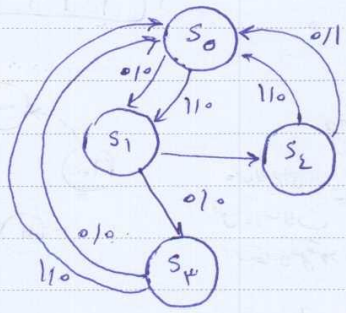
Row matching

✓ Implication Table

حالت فعلی	حالت بعدی x=0	حالت بعدی x=1	خروجی x=0	خروجی x=1
S_0	S_1	S_2	0	0
S_1	S_3	S_4	0	0
S_2	S_5	S_4	0	0
S_3	S_0	S_0	0	0
S_4	S_0	S_0	1	0
S_5	S_0	S_0	0	0
S_6	S_0	S_0	1	0







طراحی مدارهای سنکرون:

۱- انتخاب مبلی یا مور ✓

۲- رسم دیاگرام حالت ✓

۳- ساده سازی دیاگرام حالت

۴- انتساب حالت state statement

۵-

انتساب حالت

۱) binary encoding

+ تعداد FF بهینه ($\log_2 N$)

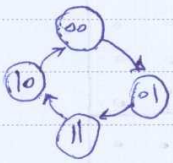
− مدار ترکیبی غیر بهینه

۲) Gray encoding

+ تعداد FF بهینه

+ توان مصرفی

− حالات خاص



۳) One-Hot encoding

− تعداد FF زیاد (N)

+ logic کم

+ توان مصرفی نسبتاً کم

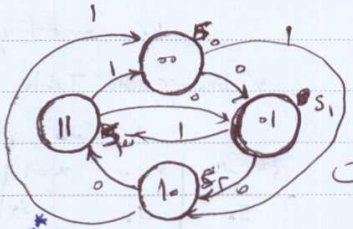
+ سرعت زیاد

− تعداد حالات غیر مجاز بیشتر

Subject: (۴۶)
Year. Month. Date. ( )

State Encoding

1) binary



A \ B	0	1
0	S_0	S_1
1	S_2	S_3

جدول کارنو انتساب حالت

x	A	B	A^*	B^*
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

A^*

x \ AB	00	01	11	10
0		1		1
1	1	1		

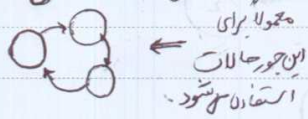
$$A^* = \bar{A}B + x\bar{A} + \bar{x}A\bar{B}$$

B^*

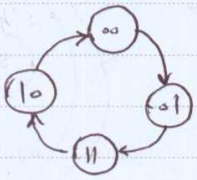
x \ AB	00	01	11	10
0	1	1	1	
1		1		

$$B^* = x\bar{A}B + \bar{x}A + \bar{x}\bar{B}$$

2) Gray encoding



← مجبوراً برای این چنین حالات استفاده می‌شود.



3) one-hot encoding

$S_0 : 0001$
$S_1 : 0010$
$S_2 : 0100$
$S_3 : 1000$

پیاده‌سازی: کاربرد → FPGA

* تعداد فلیپ فلاپ‌ها در این روش زیاد است.

x	A_0	A_1	A_2	A_3	A_0^*	A_1^*	A_2^*	A_3^*
0	0	0	0	1	0	0	1	0
	0	0	1	0	0	1	0	0
	0	1	0	0	1	0	0	0
	1	0	0	0	0	0	1	0
1	0	0	0	1	0	1	0	0
	0	0	1	0	1	0	0	0
	0	1	0	0	0	0	0	1
	1	0	0	0	0	0	0	1

$$A_0^* = \bar{x}\,\bar{A_0}A_1\bar{A_2}\bar{A_3} + x\bar{A_0}\,\bar{A_1}A_2\bar{A_3}$$
$$= \bar{x}A_1 + xA_2$$
$$A_1^* = \bar{x}A_2 + xA_3$$
$$A_2^* = \bar{x}A_3 + \bar{x}A_0$$
$$A_3^* = xA_0 + xA_1$$

4) Enumerated Encoding

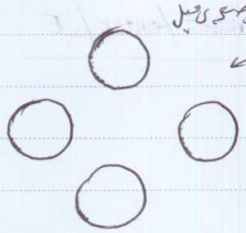
* هدف: کدگذاری حالات به طریقی که در جدول کارنوی تحریک، 1های بیشتری باهم همسایه شوند.

توابع → با فرض DFF

- به حالت‌هایی که به ازای یک ورودی، حالت بعدی یکسان دارند کد مجاور می‌دهیم. (بالاترین تقدم)
- به ورودی‌های مجاور، حالت‌های بعدی یک حالت هستند کد مجاور می‌دهیم. (تقدم متوسط)
- به ازای یک ورودی، خروجی یکسان دارند کد مجاور می‌دهیم. (کم‌ترین تقدم)

P4PCO

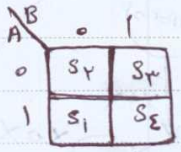
دیاگرام حالت مدل میلی



CS	x=۰	x=۱
S_1	S_2	S_3
S_2	S_3	S_4
S_3	S_4	S_1
S_4	S_2	S_1

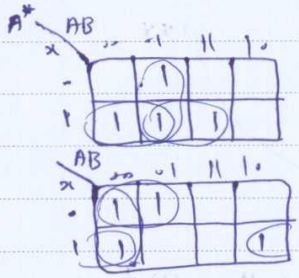
۱) (۱,۴) (۳,۴)

۲) (۲,۳) (۳,۴) (۱,۴) (۱,۲)



CS	x=۰	x=۱
۱۰	۰۰	۰۱
۰۰	۰۱	۱۱
۰۱	۱۱	۱۰
۱۱	۰۰	۱۰

NS



$$A^* = x\bar{A} + x\bar{B} + \bar{A}B$$

$$B^* = \bar{x}\bar{A} + x\bar{B}$$

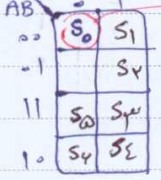
CS	x=۰	x=۱	خروجی x=۰	خروجی x=۱
S_0	S_1	S_2	۱	۰
S_1	S_3	S_4	۱	۰
S_2	S_4	S_4	۰	۱
S_3	S_5	S_5	۰	۱
S_4	S_3	S_4	۱	۰
S_5	S_0	S_0	۰	۱
S_6	S_0	X	۱	X

مثال)

۱) (۳,۴) (۵,۶) (۱,۲)

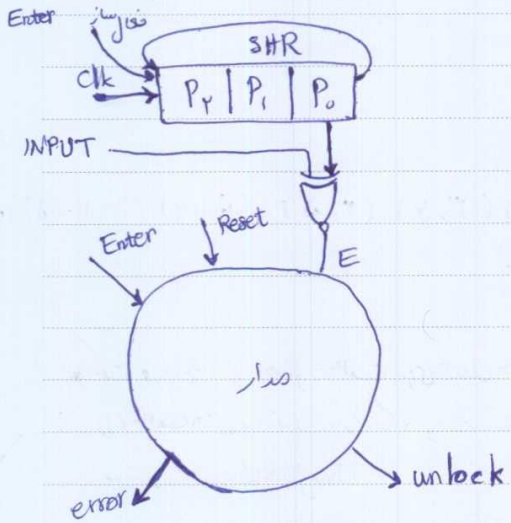
۲) (۱,۲) (۳,۴) (۵,۶)

۳) (۰,۱,۴,۶) (۲,۳,۵)



معمولاً راحت ترو که S_0 را در خانه‌ی ۰۰ بذاریم

(برای reset)



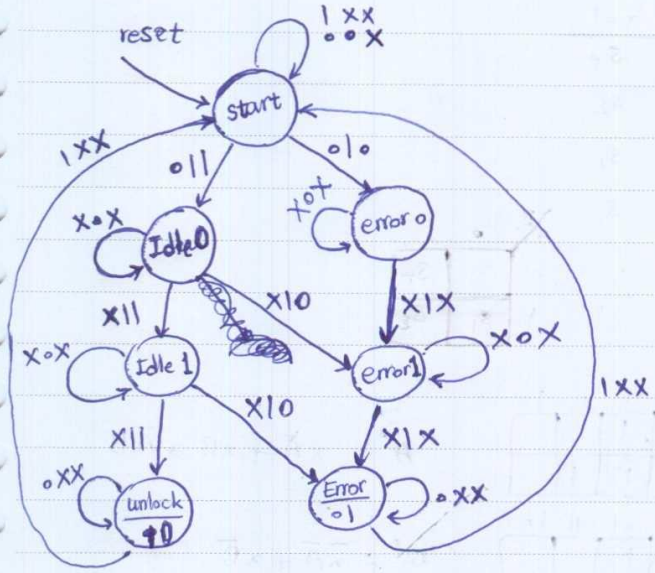
مثال) طراحی قفل سه بیتی.

ورودی:
- ۱ بیتی INPUT
- ۱ بیتی Clk
- ۱ بیتی Reset
- ۱ بیتی Enter

خروجی:
- // Unlock
- // Error

Subject: (۴۸)
Year. Month. Date. ( )

reset /enter/ E



CS	NS 000	001	010	011	100	101	110	111	خروجی unlock	error
S_0	S_0	S_0	S_2	S_1	S_0	S_0	S_0	S_0	0	0
S_1	S_1	S_1	S_4	S_3	S_1	S_1	S_4	S_3	0	0
S_2	S_2	S_2	S_4	S_4	S_2	S_2	S_4	S_4	0	0
S_3	S_3	S_3	S_4	S_5	S_3	S_3	S_4	S_5	0	0
S_4	S_4	S_4	S_4	S_4	S_4	S_4	S_4	S_4	0	0
S_5	S_5	S_5	S_5	S_5	S_0	S_0	S_0	S_0	1	0
S_6	S_6	S_6	S_6	S_6	S_0	S_0	S_0	S_0	0	1

انتساب حالت

(۱) حالت‌هایی که به ازای ورودی مشابه NS مشابه دارند.

$(3,6)(0,5,6)(4,6)(3,5)(3,4)(1,2)$

(۲) حالت‌هایی که به ازای ورودی‌های مجاور NS یک حالت هستند.

$(0,5)(4,6)(0,4)(5,6)(3,6)(3,5)(2,4)(3,4)(1,4)(1,3)(1,2)(0,1)(0,2)$

(۳) حالت‌هایی که به ازای ورودی مشابه خروجی مشابه دارند:

$(0,1,2,3,4)$

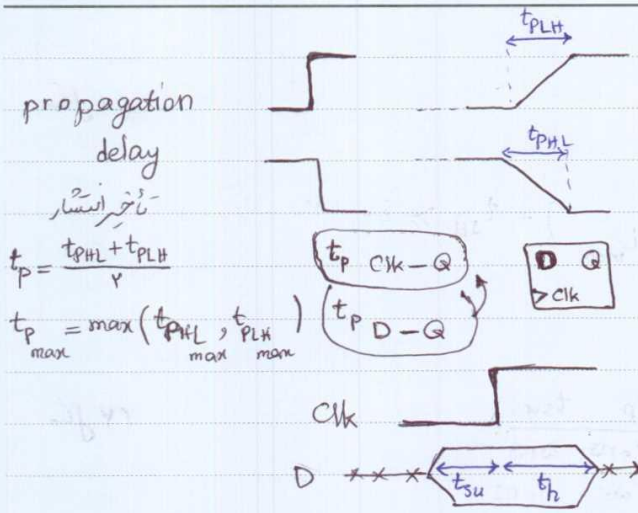
	0	1
00	S_0	S_5
01	S_1	S_4
11	S_2	S_4
10		S_6

* نکته: در مرحله‌ی کاهش حالت برای جدول حالت با تعیین ناکامل می‌توان جدول انجاب را با حالت‌های بی‌اهمیت رسم کرد و با استفاده از نمودار ادغام تعداد حالات را به نحو بهینه کاهش داد.

P4PCO

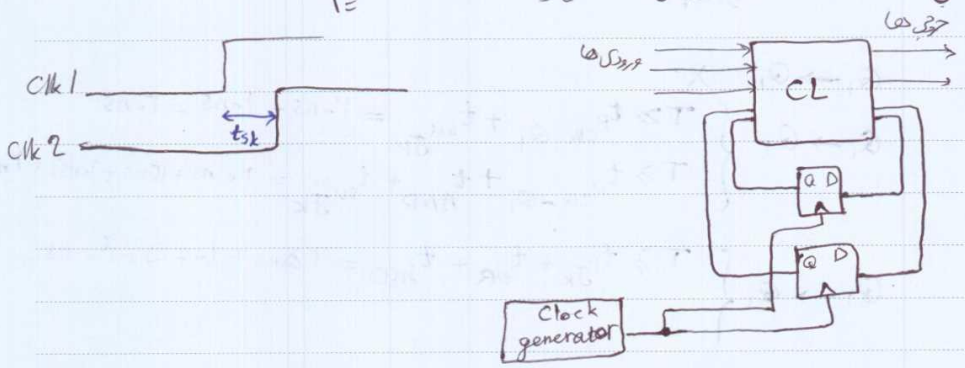
مدل‌های سنتروز

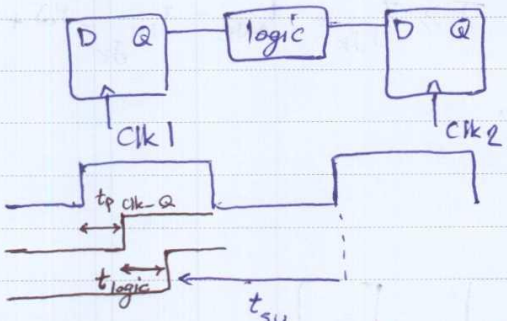
پارامترهای زمانی



- logic
  - $t_{PLH\,min}$
  - $t_{PHL\,min}$
  - $t_{PLH\,max}$
  - $t_{PHL\,max}$
- FF
  - $(t_{PLH}, t_{HL})_{min\,max}$
  - t_{su} setup time
  - t_h hold time
  - t_{sk} skew time

skew : تأخیر یک سیگنال clk نسبت به سیگنال clk دیگر clock skew گوییم.





۱) setup

$$t_{p_{Clk-Q}\,max} + t_{logic\,max} \leq T - t_{su}$$

$$\Rightarrow \boxed{T \geq t_{p\,max} + t_{logic\,max} + t_{su}}$$

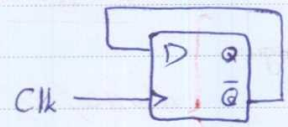
از روی طولانی‌ترین مسیر logic تعیین می‌شود.

۲) hold

$$t_{p_{Clk-Q}\,min} + t_{logic\,min} \geq t_h$$

Subject: ۵۰
Year. Month. Date. ( )

7474

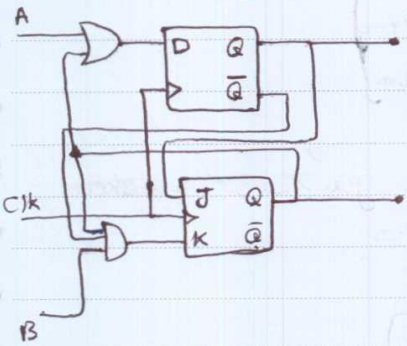


$t_{PLH_{max}} = ۲۵ ns$
$t_{PHL_{max}} = ۴۰ ns$
$t_{su} = ۲۰ ns$

(۱ نمره)

$T \geq \max(t_{PLH_{max}} + t_{PHL_{max}}) + t_{su} \geq ۴۰ + ۲۰ = ۶۰ ns$

✓ $t_{p_{min}} + ۰ \geq t_h$



	t_p	t_{su}
DFF	۲۰ns	۵ns
JK	۲۵ns	۱۰ns
AND	۱۲ns	
OR	۱۰ns	

(۲ نمره)

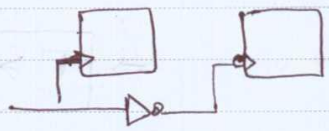
$Q_1 \rightarrow Q_1$ X

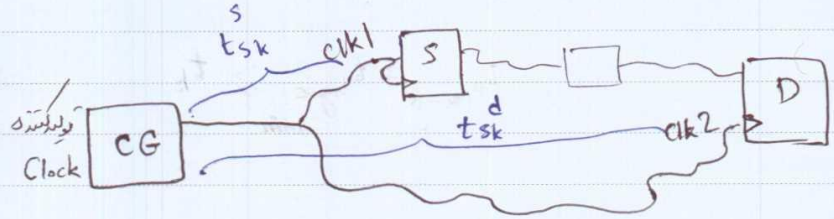
$Q_1 \rightarrow Q_۲$ $\begin{cases} T \geq t_{p_{Clk-Q_1}} + t_{su_{JK}} = ۲۰ns + ۱۰ns = ۳۰ns \\ T \geq t_{p_{Clk-Q_1}} + t_{AND} + t_{su_{JK}} = ۲۰ ns + ۱۲ns + ۱۰ns = ۴۲ns \end{cases}$

$Q_۲ \rightarrow Q_1$ $\begin{cases} T \geq t_{p_{JK}} + t_{OR} + t_{su_D} = ۲۵ns + ۱۰ + ۵ = ۴۰ ns \end{cases}$

$Q_۲ \rightarrow Q_۲$ $T \geq t_{p_{JK}} + t_{AND} + t_{su_{JK}} = ۲۵ + ۱۲ + ۱۰ = ۴۷ ns$

skew { تاخیر سیم
logic تاخیر { Clock gating, INV, بافر

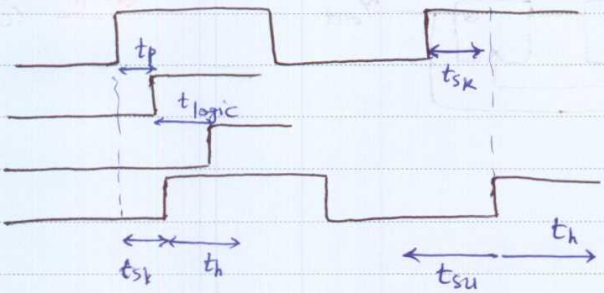




$t_{sk} = t_{sk}^{d} - t_{sk}^{s}$

P4PCO

Subject: (۵۱)
Year. Month. Date. ( )



$$t_{p_{max}} + t_{logic_{max}} \leq T - (t_{su} - t_{sk})$$

$$\Rightarrow \boxed{T \geq t_{p_{max}} + t_{logic_{max}} + (t_{su} - t_{sk})}$$

F_{Clk} ↑ ← بدون skew ← $\infty \gg t_{sk} > 0$ ازای

$$\boxed{t_{p_{min}} + t_{logic_{min}} \geq t_h + t_{sk}}$$

$$t_{sk} \leq t_{p_{min}} + t_{logic_{min}} - t_h$$

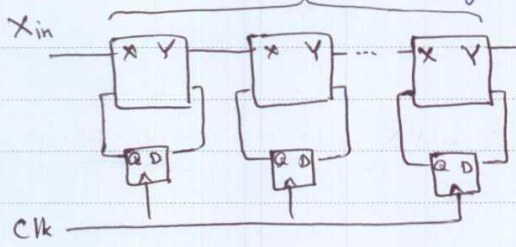
maximum allowable clock skew

$0 < t_{sk} \ll \infty$ → بدون

$t_{sk} \gg 0$ → hold violation (داده از دو یا چند ff در یک clk رد می‌شود)

$t_{sk} \ll 0$ → setup violation (از یک ff تا ff بعدی)

Metastability (بعد)



(از طرف clk)

$Freq_{max} = ?$

$Delay_{max} = ?$

$Delay_{min} = ?$

$$T \geq t_{pff_{max}} + t_{Q_Y{}_{max}} + (n-1)\, t_{xy_{max}} + t_{x_D{}_{max}} + t_{su}$$

$$T \geq t_{pff_{max}} + t_{Q_D{}_{max}} + t_{su}$$

$$T \geq t_{pff_{clk-Q_{max}}} + t_{path_{Q-D}} + t_{su} - (t_{sk}^{d} - t_{sk}^{s})$$

← در نظر گرفتن clk skew ازای (*)

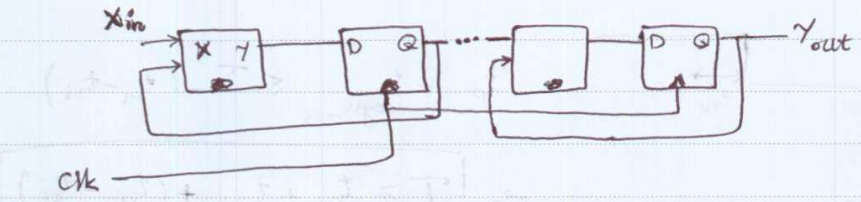
Max delay $\Rightarrow T_{C_{Yout}} \leq t_{pff_{max}} + t_{Q_Y{}_{max}}$

Min delay $T_{C_{Yout}} \geq t_{pff_{min}} + t_{Q_Y{}_{min}}$

Y_{out} ← clk ماکزیمم تأخیر بین (*)

Subject: (۵۲)
Year. Month. Date. ( )

مثال Moore

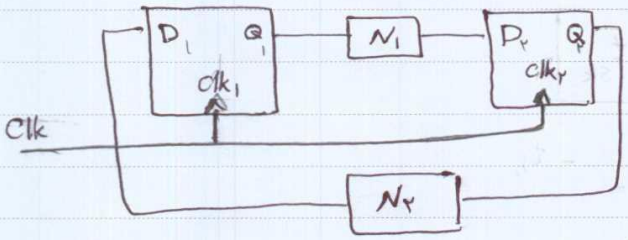


$T \geq t_{pff_{max}} + t_{X_{max}} + t_{su}$

Max delay $\longrightarrow$ $T_{cy} \geq t_{pff_{max}}$

Min delay $\longrightarrow$ $T_{cy} \geq t_{pff_{min}}$

مثال $clk_2 - clk_1 = 10ns$ حداقل فرکانس کاری = ؟



	min	max
t_{pff}	10ns	15ns
t_{N_1}	4ns	5ns
t_{N_2}	2ns	25ns
t_{su}		20ns
t_h		0

$T \geq t_{pff_{max}} + t_{N_1} + t_{pff_{max}} + t_{N_2} + t_{su} + clk_2 - clk_1$

$t_{sk_{min}}$

$T \geq t_{pff_{max}} + t_{N_{1max}} + (t_{su} - t_{sk}) = 15 + 50 + 20 - 10 = 75ns$

FF_2 FF_1

$T \geq t_{pff_{max}} + t_{N_{2max}} + (t_{su} - t_{sk}^{d} + t_{sk}^{s}) + t_{sk_{max}} = 15 + 25 + 20 + 50 = 110ns$

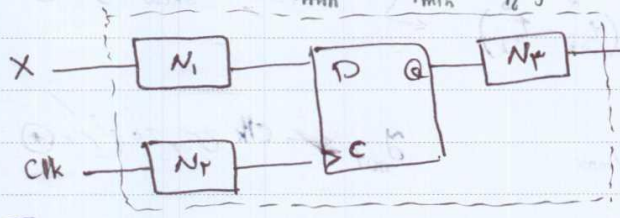
$t_{sk} + t_h \leq t_{pff_{min}} + t_{N_{1min}}$

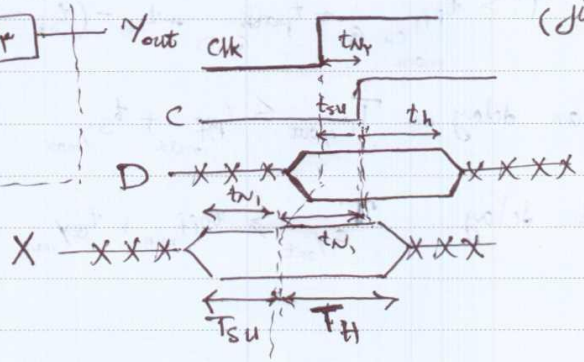
$\Rightarrow t_{sk} \leq t_{pff_{min}} + t_{N_{1min}} - t_h$

$\longrightarrow$ حداکثر t_h و حداقل t_{sk}

$\longrightarrow t_{sk_{max}} = t_{pff_{min}} + t_{N_{1min}} - t_h = 50ns$

مثال

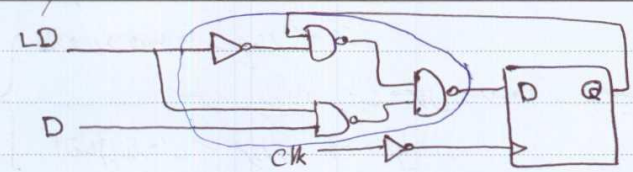


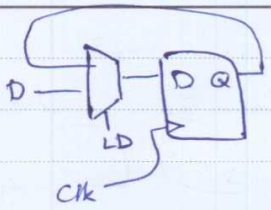


$T_{su} = t_{su} + t_{N_{1max}} - t_{N_{3min}}$

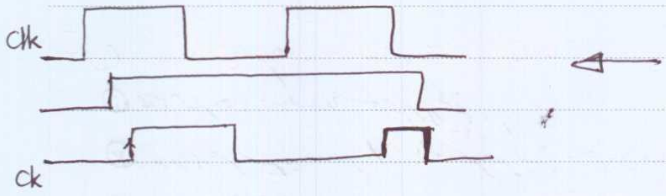
$T_H = t_h - t_{N_{1min}} + t_{N_{3max}}$

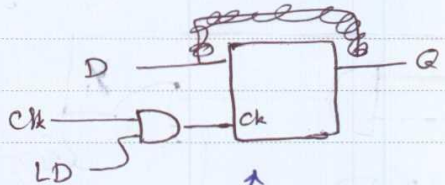
$tp_{clk-Yout} = t_{N_{2max}} + t_{clk-Q_{max}} + t_{N_{3max}}$





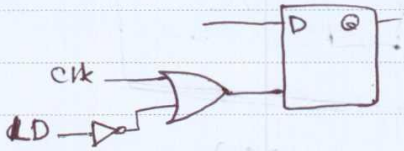
مثال (۱۲)





دارد skew .

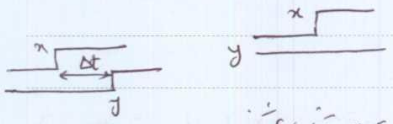
تحقیق کنید ④



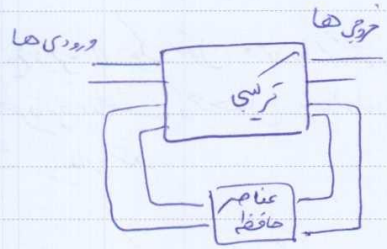
مدارهای آسنکرون

مدارهای پالسی
- در هر لحظه فقط یکی از ورودی ها پالس داریم.
- پالس های ورودی به اندازه‌ی Δt تأخیر دارند.
- فقط به ورودی های متمم شده یا متمم نشده نیاز داریم.

مدارهای وجه بنیادی Fundamental Mode
- در هر لحظه فقط یکی از ورودی ها تغییر می کند.
- تغییرات ورودی ها به اندازه‌ی Δt تأخیر دارند.
- ورودی ها هم متمم شده و هم متمم نشده ممکن است استفاده شوند.



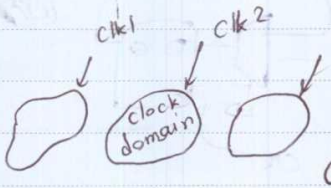
وجه تمایز مدارهای آسنکرون و سنکرون
- مدارهای آسنکرون Clk ندارند.
- تغییر حالت مدار پس از تغییر ورودی به صورت آنی صورت می گیرد.



latch
ff
تأخیر Δt

Subject: (۵۴)
Year. Month. Date. ( )

مدارهایی که باید پاسخ سریع داشته باشند.
" داشتن مدار تولید clk متمرکز به صرفه و اقتصادی نیست.
اتصال مدارهای سنکرون که clk جدا دارند.

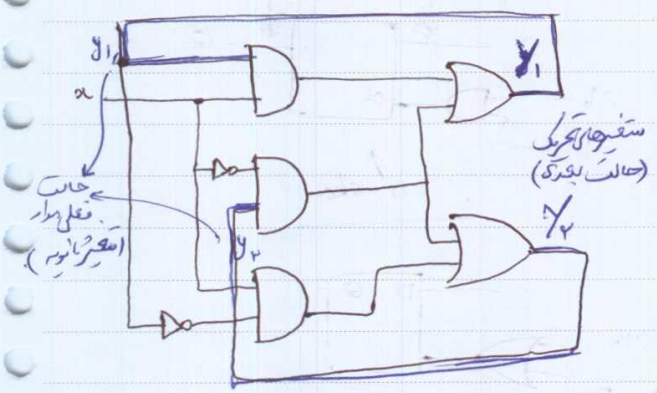


Globaly Asynchronous locally synch.

مثال) تحلیل مدارهای سنکرون وجه بنیادی:

① تمام مسیرهای فیدبک را مشخص می‌کنیم.

② برای هر مسیر، متغیر Y (متغیر فیدبک) در خروجی مسیر و متغیر y (متغیر حالت دوم) به عنوان ورودی مسیر در نظر می‌گیریم.



c) Δt_1 : تأخیر
Δt_2 : تأخیر فیدبک

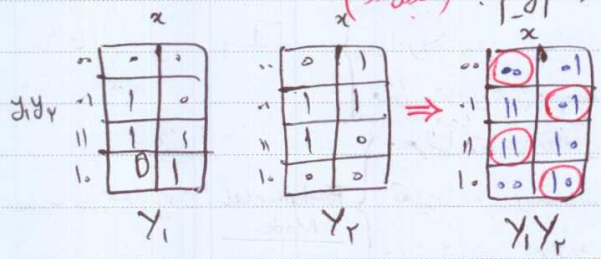
③ معادلات بولی Y برحسب ورودی‌ها و y را می‌نویسیم.

$Y_1 = xy_1 + \bar{x}y_2$ ، $Y_2 = x\bar{y}_1 + \bar{x}y_2$

بعد از تغییر در x ($\Delta t_1 + \Delta t_2$)

④ جدول کارنوی Y_1Y_2 را برحسب ورودی‌ها و y_1y_2 رسم می‌کنیم. (جدول نذر)

* دقت: متغیرهای حالت را در سطرها بنویسید.



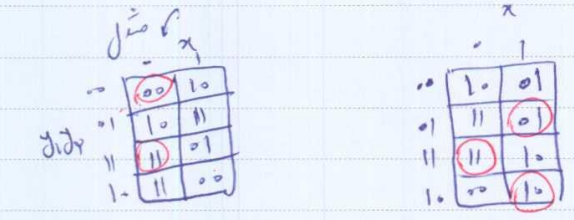
⑤ حالات پایدار مدار را با دایره مشخص می‌کنیم (خانه‌هایی که $y_1y_2 = Y_1Y_2$)

* حالت کلی مدار (xy_1y_2)
* حالت داخلی مدار (y_1y_2)

حالات کلی پایدار: {000, 011, 110, 101}

* اگر در جدول حالتی هیچ دایره‌ای در یک ستون وجود نداشته باشد، آن گاه مدار هیچ گاه پایدار نمی‌شود به ازای آن ورودی.
مدار ناپایدار کلی است.
* ولی می‌توان در یک سطر دایره نداشت و همچنان پایدار بود.
* شروع باید از یک حالت پایدار باشد.

مثال



مثال از جدولی که با وجود دایره در همه ستون‌ها باز هم مدار ناپایدار است.

P4PCO

(Flow table) جدول روند ⊛

جدول گذاری است که { حالات مدار که نشده اند. / خروجی ها نیز مشخص شده اند.

	x_1x_2 = 00 (خروجی)	01	11	10
a	$a_{,0}$	$a_{,0}$	$a_{,0}$	$b_{,0}$
b	$a_{,0}$	$a_{,0}$	$b_{,1}$	$b_{,0}$

⊛ $x_1x_2 = 01$ ⟸ $x_1x_2 = 11$
حالت فعلی = a ⟸ حالت فعلی = a

⊛ $x_1x_2 = 10$ ⟸ $x_1x_2 = 11$

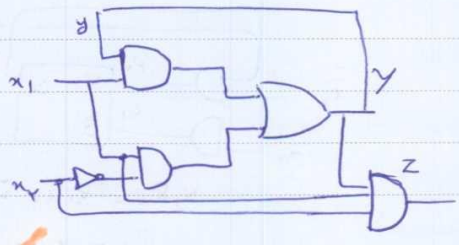
⟹ جدول فلز
$a = 0$
$b = 1$

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

Y

y \ x_1x_2	00	01	11	10
0				
1			1	

Z → خروجی



$Y = x_1y + x_1\bar{x_2}$ ، $z = x_1x_2y$

y \ x	0	1
00	00	11
01	11	11
11	11	(11)
10	11	11

هدف: 00 → 11 (خروجی)
در عمل: 00 → 01 → 11
00 → 10 → 11

Race رقابت } مدارهای آسنکرون
مثل مثال فوق: حالت نهایی همه ی چرخه ها یکی باشد.

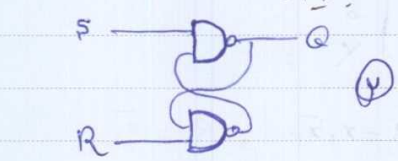
Race → رقابت غیر بحرانی
رقابت بحرانی → (مثال)

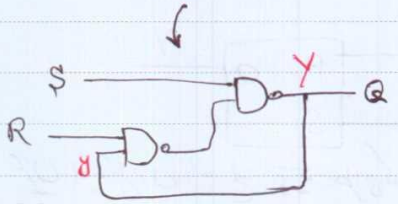
برعکس حالت فوق.

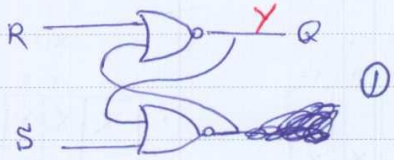
00	11
01	
11	
10	

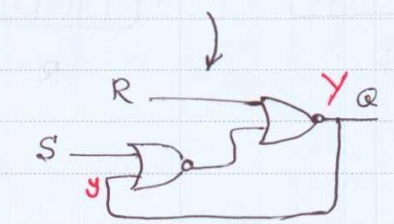
00 → (11)
00 → (01)
00 → (10)

نحوه ی تخصیص حالت: Race - free : کدهایی که در نظر گیریم برای جاوره های جدول فوق.

(۲)




(مثال ۱)




Subject: ۵۶
Year. Month. Date. ( )

① $Y = \overline{R + (\overline{S+y})} = \bar{R}(S+y) = S\bar{R} + y\bar{R}$ ← فرض کنیم $SR=0$ ⟹ $\boxed{Y = S + \bar{R}y}$ معادله‌ی مدار NOR SR

② $Y = \overline{S(\overline{Ry})} = \bar{S} + Ry$ ← فرض کنیم $\bar{S}R=0$ ⟹ $\boxed{Y = \bar{S} + Ry}$ → معادله‌ی مدار $\bar{S}R$

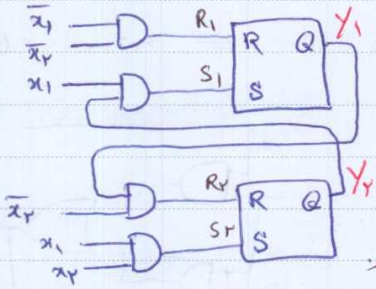
① :

y \ SR	00	01	11	10
0	0	0	-	1
1	1	0	0	1

② :

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

مثال)



① معادلات ورودی لچ‌ها

$S_1 = x_1 y_2$

$R_1 = \bar{x}_1 \bar{x}_2$

$S_2 = x_1 x_2$

$R_2 = \bar{x}_2 y_1$

② چک می‌کنیم شرط $SR=0$ یا $\bar{S}\bar{R}=0$ برقرار باشد.

$S_1 R_1 = x_1 \bar{x}_1 \bar{x}_2 y_2 = 0$ ✓

$S_2 R_2 = x_1 x_2 \bar{x}_2 y_1 = 0$ ✓

⟹ پس می‌توانیم از معادلات مثال قبل استفاده کنیم.

$Y_1 = S_1 + \bar{R}_1 y = x_1 y_2 + x_1 y_1 + x_2 y_1$

$Y_2 = S_2 + \bar{R}_2 y = x_1 x_2 + x_2 y_2 + \bar{y}_1 y_2$

y_1y_2 \ x_1x_2	00	01	11	10
00	00	00	01	00
01	01	01	11	11
11	00	11	11	10
10	00	10	11	10

مثال) طراحی با لچ SR

a / b \ x_1x_2	00	01	11	10
a	a	a	a	b
b	a	a	b	b

رسم مدار ← جدول کارنو ← جدول نهایی ← انتساب حالت ← جدول روند ← کاهش حالات ← توصیف : طراحی مدارهای کنترل

انتساب حالت : $a=0$, $b=1$

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

y	Y	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

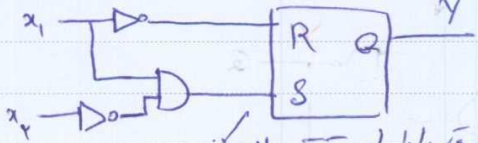
S

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

R

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

$S = x_1 \bar{x}_2$, $R = \bar{x}_1$



تمرین) مثال فوق را با لچ $\bar{S}R$ طراحی کنید.

P4PCO

Subject: ۵۷
Year. Month. Date. ( )

طراحی مدارهای آسنکرون

۱ـ مشخصات طراحی
۲ـ جدول روند اصلی (جدول روندی که در هر سطر فقط یک حالت پایدار دارد.) ← Primitive flow table
۳ـ کاهش حالات (جدول ایماب)
۴ـ انتساب حالت
۵ـ به حالت بی اهمیت خروجی در حالات ناپایدار، خروجی قطعی نسبت می دهیم!
۶ـ توابع بولی خروجی ها و تحریک (Y) در جدول کارنوی آن ها ساده کرده، مدار را رسم می کنیم.

مثال) یک لچ D با ورودی (Gate) (D, G): ورودی ها

خروجی Q

ترتیب حالات:
a 01
b 11
c 10
d 10

مراحل ① و ②

DG

	00	01	11	10
a	c/-	ⓐ,0	b/-	-/-
b	-,-	a/-	ⓑ/1	e,-
c	ⓒ/0	a/-	-,-	d/-
d	c/-	-,-	b/-	ⓓ/0
e	f,-	-,-	b/-	ⓔ,1
f	ⓕ,1	a/-	-/-	e/-

کاهش حالت ← حالات سازگار: حالاتی که بعد از تعیین مقادیر don't care معادل می شوند.

(a,b)

(a, c, d) → a
(b, e, f) → b
⇐ کاهش حالت

a	ⓒ/0	ⓐ,0	b,-	ⓓ,0
b	ⓕ,1	a,-	ⓑ/1	ⓔ,1

DG

	00	01	11	10
a	ⓐ/0	ⓐ/0	b/-	ⓐ/1
b	ⓑ/1	a,-	ⓑ/1	ⓑ,1

انتساب حالت: a=0, b=1

DG

	00	01	11	10
0	⓪,0	⓪,0	11/-	⓪,0
1	①,1	01/-	①/1	①,1

جدول گذر

حالت بعدی Y

	00	01	11	10
0	0	0	1	0
1	1	0	1	1

$Y = DG + \bar{G}y$

خروجی Z

	00	01	11	10
0	0	0	X	0
1	1	X	1	1

$Z = y$

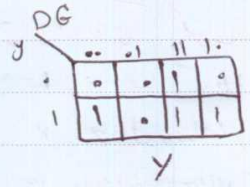
جدول کارنوی Y

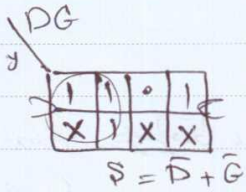
دقت: ساده کردن جدول کارنو باید با توجه به گروه صورت گیرد نه فقط با توجه به خود جدول کارنو.

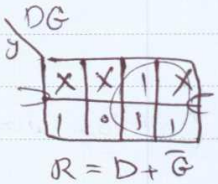
Subject: (۵۸)
Year. Month. Date. ( )

لچ D

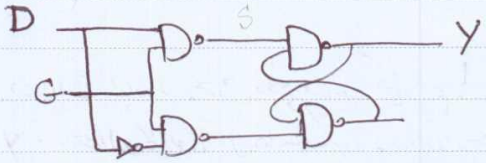
پیاده سازی با لچ $\overline{SR}$







y	Y	S	R
0	0	1	X
0	1	0	1
1	0	1	0
1	1	X	1



* لذا از یک حالت پایدار به یک حالت ناپایدار و دوباره به حالت پایدار بعدی نباید با تغییر ناخواسته در خروجی همراه باشد.

مثال

y \ x	0	1
a	(a)/0	b/-
b	c/-	(b)/0
c	(c)/1	d/-
d	a/-	(d)/1

* ولی وقتی از (b) ورودی را صفر می‌کنیم تا به c برویم خروجی را X نمی‌گذاریم چون خروجی های دو حالت پایدار متفاوت است.

کاهش حالت:

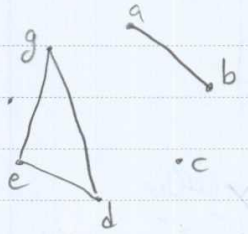
* ممکن است حالاتی داشته باشیم که حاوی don't care باشند. (مثال سنگرودی)

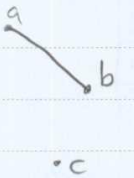
حالت فعلی	حالت بعدی x=0	حالت بعدی x=1	خروجی x=0	خروجی x=1
a	d	b	0	0
b	e	a	0	0
c	g	f	0	1
d	a	d	1	0
e	a	d	1	0
f	c	b	0	0
g	a	e	1	0

	a	b	c	d	e	f
b	e-d					
c	X	X				
d	X	X	X			
e	X	X	X	✓		
f	~~c-d~~	~~a-e~~ a-b	X	X	X	
g	X	X	X	e-d	e-d	X

d-e
a-b
e-g
d-g

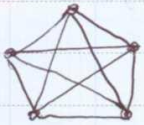
غیرقابل قبول

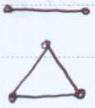




(e, g, d) (a, b) (c) (f)

* دسته‌های کامل به شکل های زیر هستند:





PdPCO

Subject: ۵۹
Year. Month. Date. ( )

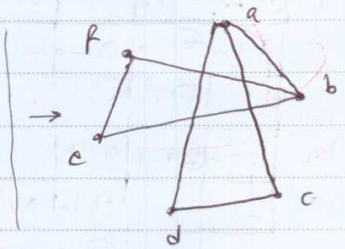
	11	10	01	00
a	b/-	-/-	a/0	c/-
b	a/-	b/1	-/-	e/-
c	a/-	-/-	-/-	d/-
d	b/-	-/-	-/-	c/1
e	b/-	-/-	-/-	f/-
f	a/-	-/-	-/-	f/1

1

	a	b	c	d	e
b	✓				
c	✓	d-e			
d	✓	d-e	✓		
e	c-f	✓	d-e, c-f	X	
f	e-f	✓	X	d-e	✓

→ بعضی از زوج حالت‌ها تشکیل مجموعه سازگاری‌ها نمی‌دهند.

مثال:



* در این مثال حالت موجهی نداشتیم چون همه خط خوردند.

کاهش حالت در مدارهای آسنکرون و سنکرون با تعیین ناکامل.

۱) جدول انجاب ۲) یافتن تمام حالات سازگار ۳) نمودار ادغام
۴) یافتن سازگارهای ماکزیمال ۵) یافتن مجموعه های مینیمال.

(b,f,e) (a,c,d) (a,b)

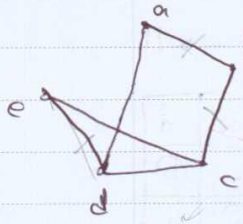
* نکته: مجموعه‌ی نهایی باید
- بسته ← اگر حالت شرط (موجهی) وجود دارد درون مجموعه باشد.
- پوشا ← همه‌ی حالات مدار پوشش داده شده باشند.

پس در مثال فوق می‌توان (a,b) را حذف کرد.

(b,f,e) → 1
(a,c,d) → 2

	11	10	01	00
1	2/-	1/1	1/1	1/1
2	2/0	2/0	1/-	2/0

مثال:



(e,d,c) (a,d) ~~(a,b)~~ (b,c)

	a	b	c	d
b	b/c			
c	X	d/e		
d	b/c	X	a/d	
e	X	X	✓	b/c

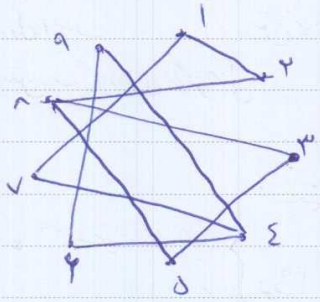
حذف
- d,c
- c,e
- a,d

(a,b)(b,c)(d,e) → مجموعه‌ی مینیمال

Subject: (40)
Year. Month. Date. ( )

	۰۰	۰۱	۱۱	۱۰
1	①۱۰۰	۳،۰۰	-،--	۷،۰۰
2	۱،۰۰	②۱۰۰	۵،-۰	-،--
3	۱،-۰	③۱،۰	۵،۱۰	-،--
4	۱،۱۰-	④۱،۰۱	۹،۰۱	-،--
5	-،--	۳،۱۰	⑤۱،۱۰	۸،۱۰
6	-،--	۴،۰۱	⑥۱،۰۱	۹،۰۱
7	۱،۰۰	-،--	۹،۰-	⑦۱،۰۰
8	۱،-۰	-،--	۵،۱۰	⑧۱،۱۰
9	۱،۱۰-	-،--	۹،۰۱	⑨۱،۰۱

	۱	۲	۳	۴	۵	۶	۷	۸
۲	✓							
۳	X	X						
۴	X	X	X					
۵	X	X	✓	X				
۶	X	X	X	✓	X			
۷	✓	~~X~~	X	✓	X	X		
۸	X	✓	✓	X	✓	X	X	
۹	X	X	X	✓	X	✓	X	X



(۲,۸)(۱,۲)(۱,۷)(۴,۷)(۳,۵,۸)(۴,۶,۹)

a ✓ b ✓ c X d ✓ X

* چون حالت شرط ندارم به راحتی قابل حذف هستند.

(۴,۶,۹) ⟶

۱۱۰-	⑤۱،۰۱	⑨۱،۰-	⑨۱،۰۱

⇓

a

⑥۱،۰-	⑥۱،۰۱	⑥۱،۰۱	⑥۱،۰۱

انتساب حالت

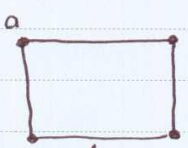
۰۰ ⟶ ۱۱

۰۰ ⟶ ۰۱ ⟶ ۱۱

۰۰ ⟶ ۱۰ ⟶ ۱۱

a	ⓐ	d	ⓐ	d
b	ⓑ	d	c	ⓑ
c	a	ⓒ	ⓒ	b
d	b	ⓓ	a	ⓓ

جدول انتقال

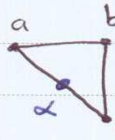


جدول کارنوی انتساب

yz \ wx	۰۰	۰۱	۱۱	۱۰
۰	a	d		
۱	c	b		

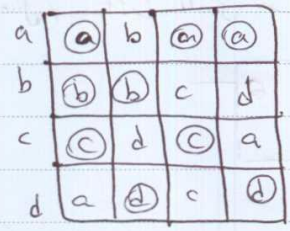
* اگر نشد که جا ورداد فقط برای جرانی ها که جا ورمی دهم (اگر بشود.)

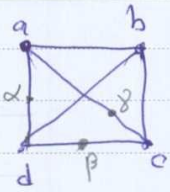
	۰۰	۰۱	۱۱	۱۰
a	ⓐ	b	ⓐ	ⓐ
b	a	ⓑ	ⓑ	c
c	ⓒ	b	a	ⓒ
α	-	-	b a	-

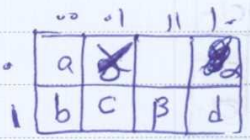


	۰	۱
۰	a	b
۱	α	c

Subject: (۶۱)
Year. Month. Date. ( )

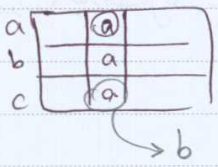


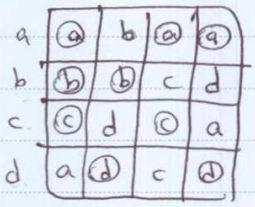


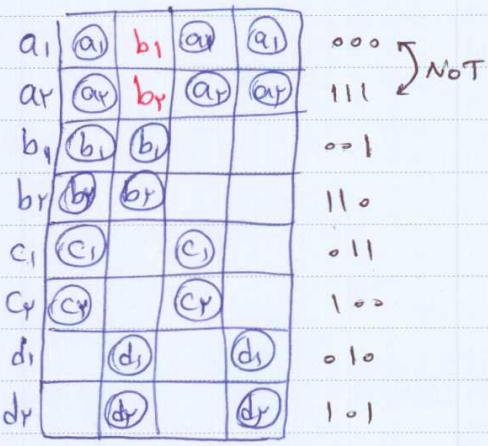


* ممکن است از حالت های غیبی مدار سوال برای همسایه کردن استفاده کرد.

مثال ↓

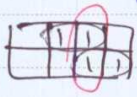






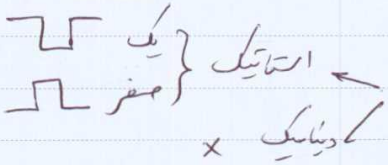
روش چند سطری

مثال)

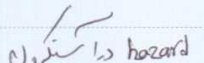


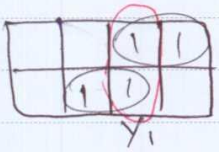
sop

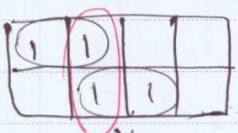
pos



hazard







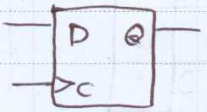
توصیف مدار به صورت جدول روند اصلی

ساده سازی جدول روند

انتساب حالت (Race free)

CD	00	01	11	10
a	(a)1-	b1-	-1-	c1-
b	a1-	(b)1-	d1-	-1-
c	a1-	-1-	e1-	(c)1-
d	-1-	g1-	(d)1-	f1-
e	-1-	b1-	(e)1-	(e)1-
f	h1-	-1-	d1-	(f)1-
g	h1-	(g)1-	d1-	-1-
h	(h)1-	g1-	-1-	c1-

مثال) Dff با لبه‌ی مثبت :



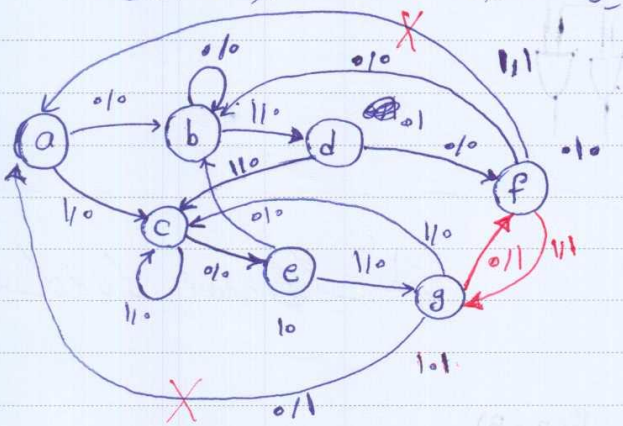
* نکته اساسی : در هر مرحله از پر کردن جدول روند نگاه می‌کنیم در ستون متصدا یا حالت پایداری وجود دارد که هم
- خروجی مطلوبی داشته باشد.
- رفتن به آن عملکرد مدار را دچار مشکل نکند.

Subject :
Year . Month . Date . ( )

* ذیل جدول مشترک تشکیل دادیم .

	a	b	c	f_1	f_2
$a'b'$	0	0	—	1	
$a'c'$	0	—	0	1	
ac	1	—	1	1	
				—	
				—	

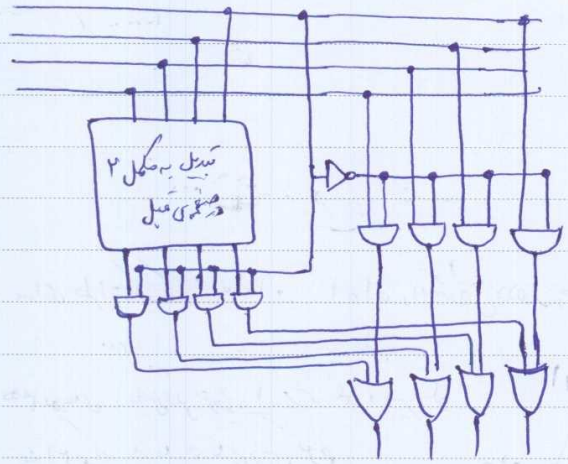
* مداری طراحی کنید که ۱۰۱۰ و ۱۰۱۰ را تشخیص دهد خروجی را set کند. هم پوشانی مجاز نیست.



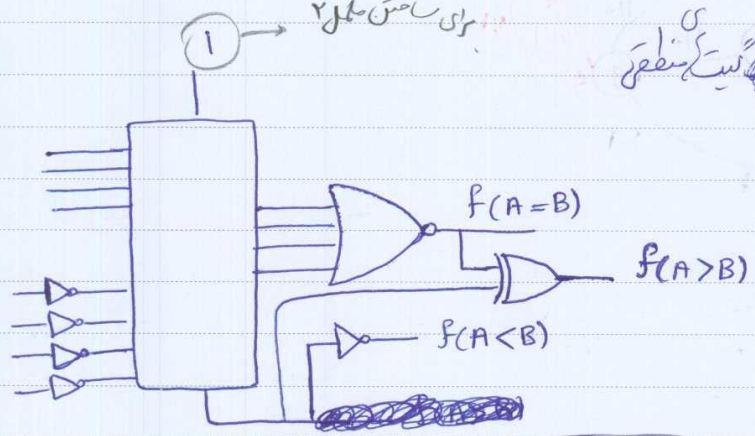
هم پوشانی : یعنی اگر خروجی 1 شد کم کم با بیت جلو
تا این که کم کم بررسی کن .

طراحی با هم پوشانی

Subject:
Year. Month. Date. ( )

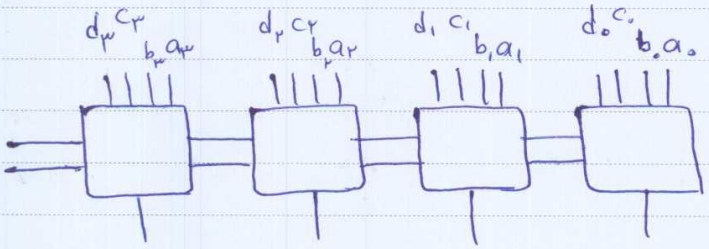


تبدیل اندازه علامت به مکمل ۲



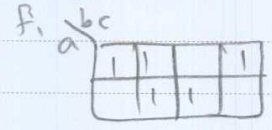
برای ساختن مکمل ۲

مقایسه کننده با یک adder و گیت منطقی



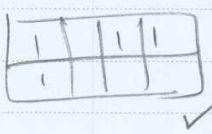
PLA ی بسازید.

$f_1 = \sum(0,1,2,5,7)$, $f_2 = \sum(0,2,3,6)$



$f_1 = a'b' + a'c' + ac$ ✓

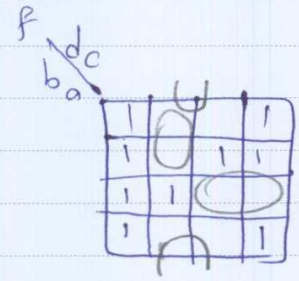
$f_1' = ac' + a'bc$



$f_2 = b'c' + a'b$

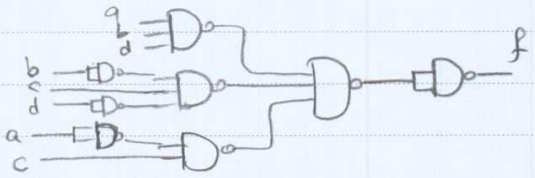
✓ $f_2' = b'c + ab$

Subject :
Year . Month . Date . ( )



$\rightarrow \bar{f} = ca' + dba + d'cb'$

7410 2x 7400
سه ورودی ← 3 4



$f = d'c' + b'c' + a'c' + d'ba + db'a$

* هرچند دیدید تعداد گیت ها زیاد است و پیچیدگی زیاد دارد، از طراحی معکوس هم بدنیست استفاده کنید برای گیت کم تر.

گذری :

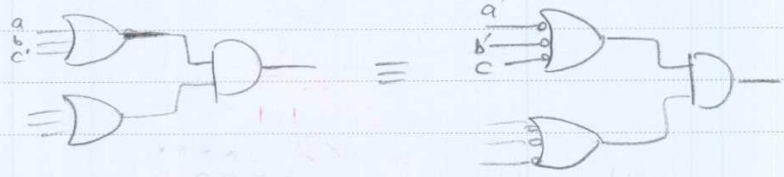
0	0	0	
0	0	1	
0	1	1	
0	1	0	
1	1	-	
1	1	1	
1	0	1	
1	0	0	

---

a) NAND - AND

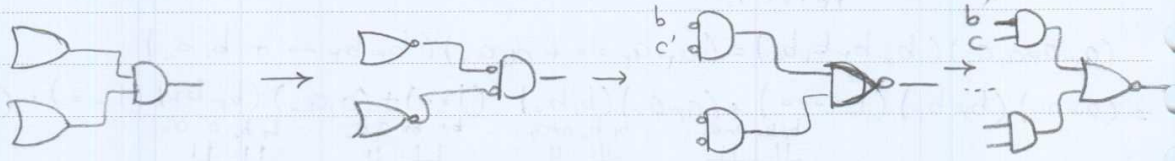
OR NOR

Pos →



b) AND - NOR

NOR → pos



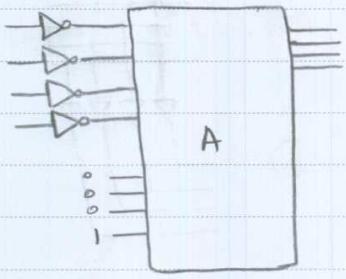
C) OR - NAND

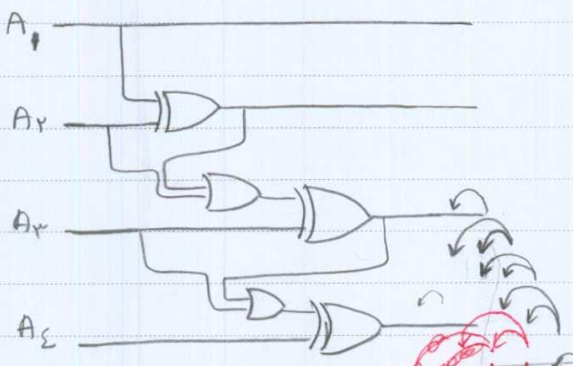
NAND

d) NOR - OR → sop →

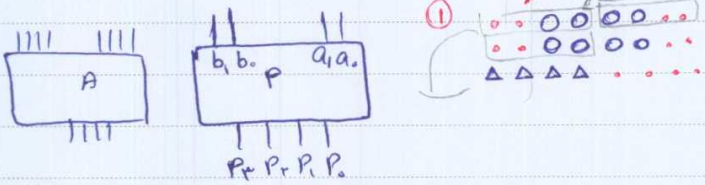
AND

Subject :
Year . Month . Date . ( )



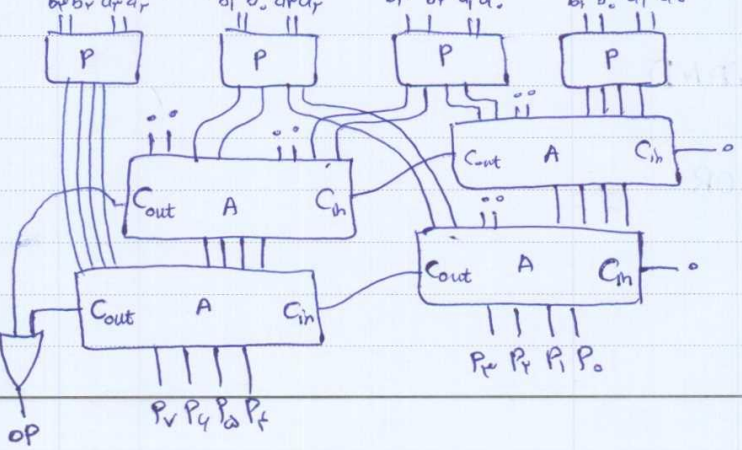


→ مثال ۴ بیتی



$$(a_3a_2a_1a_0)(b_3b_2b_1b_0) = (a_3a_2 00 + a_1a_0)(b_3b_2 00 + b_1b_0)$$

$$= (a_3a_2)(b_3b_2)(1000) + (a_3a_2)(b_1b_0)(100) + (a_1a_0)(b_3b_2)(100) + (a_1a_0)(b_1b_0)$$



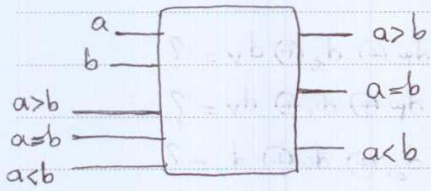
PAPCO

server → Quartus

Subject:
Year. Month. Date. ( )

mojgan.ghasemi@gmail

① مقایسه کننده یک بیتی extendable



1 0	1
1 0	0

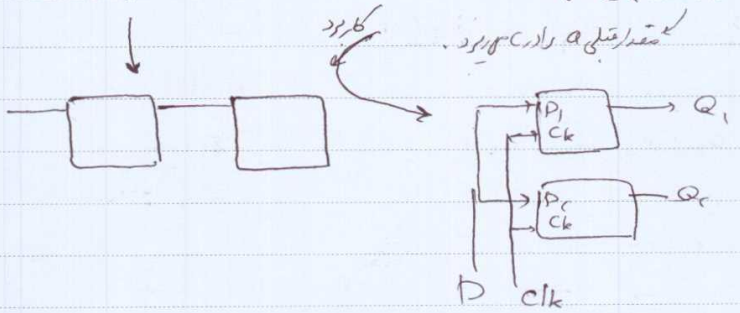
gate

② Data ← مبع کننده بیتی

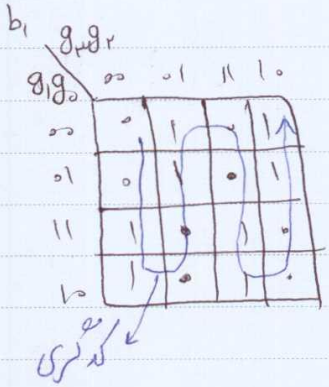
a=b; ترکیبی
c=a;

assignment : a<=b; / c<=a; ← همزمان

مقدار قبلی a وارد c میشود. ← کلاک



ترکیبی				دودویی			
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0



$b_0 = g_0 \oplus g_1 \oplus g_2 \oplus g_3$

$b_1 = g_1 \oplus g_2 \oplus g_3$

$b_2 = g_2 \oplus g_3$

$b_3 = g_3$

Subject:
Year. Month. Date. ( )

P_1	P_2	d_3	P_4	d_5	d_6	d_7

$P_1 = d_3 \oplus d_5 \oplus d_7$

$P_2 = d_3 \oplus d_6 \oplus d_7$

$P_4 = d_5 \oplus d_6 \oplus d_7$

↓ XOR

$S_0 = P_1 \oplus d_3 \oplus d_5 \oplus d_7 = ?$

$S_1 = P_2 \oplus d_3 \oplus d_6 \oplus d_7 = ?$

$S_2 = P_4 \oplus d_5 \oplus d_6 \oplus d_7 = ?$

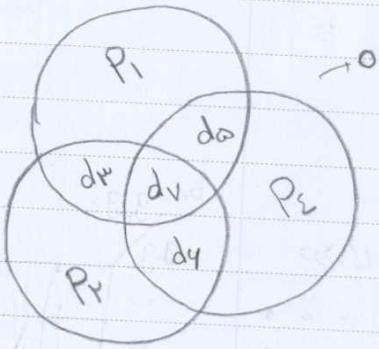
$(S_2 S_1 S_0)_2$ ← هر عددی را نشان دارد همان بیت خراب شده است.

$(P_1 P_2 X_3 P_4 X_5 X_6 X_7)_2 = (1100111)_2$ مثال)

$S_0 = (P_1, x_3, x_5, x_7) = (1, 0, 1, 1) = 1$

$S_1 = (P_2, x_3, x_6, x_7) = (1, 0, 1, 1) = 1$ → $(111)_2$ → بیت شماره 7 خراب است.

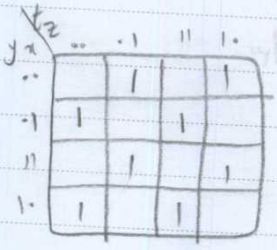
$S_2 = (P_4, x_5, x_6, x_7) = (0, 1, 1, 1) = 1$



→ ok!

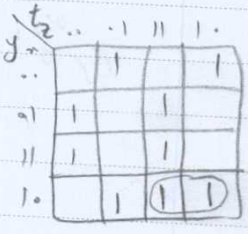
001 → d_7

$(z+x')(x+z') = z \odot x$ ↓ xnor



$x \oplus y \oplus z \oplus t$

$yz + y'z' = (y \odot z)$



$f = (x \oplus z \oplus t) + x'yt$

PqPCO