X پایان ترم ۱۳ سوال دارد. بدهد ما نیز این مثال

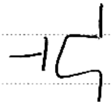
در طراحی باید H یا L یا HI مرحله مراقب باشیم

X-state نشود.

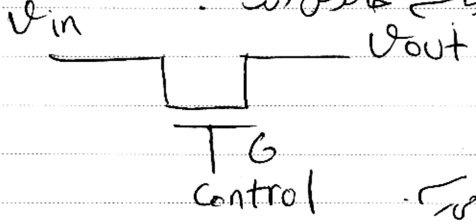
فصل ۵

مدارهای منطقی پویا

ترانزیستور عبور یا pass transistor



NMOS است اگر پایه G ولتاژ high داشته باشد روشن است اگر صفر باشد خاموش است.



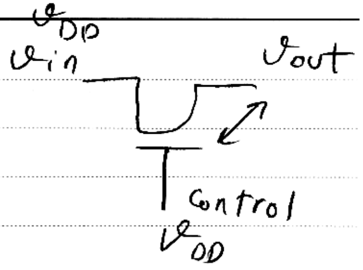
اگر کنترل ۱ باشد

ورودی عبور می‌کند به خروجی می‌رسد.

مقدار دقیقش با جدول بررسی می‌شود.

if Control = 1 => $v_{out} = v_{in}$

v_{in}	control	v_{out}
0	v_{DD}	0
v_{DD}	v_{DD}	$v_{DD}-v_t$
0	0	HI
v_{DD}	0	HI



$\Rightarrow v_{out}=v_{DD}-v_t$

NMOS اگر خاموش باشد خروجی ربطی به ورودی ندارد ← HI

اگر $v_{in}=v_{DD}$ کلید روشن است وقتی که روشن ورودی را به خروجی منتقل می‌کند.

کلید ورودی ← 0
کلید v_{DD} ← هرا منتقل می‌کند.

$v_{GS}>v_t$

$v_{in}\to v_{DD}$ $G\to v_{DD}$

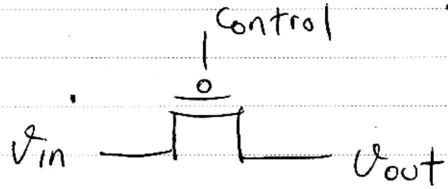
G , S ولتاژ یکسان دارند ⇐ به اندازه v_t اختلاف ولتاژ ندارد

⇐ $v_{DD}-v_t$ یعنی v_t کم کنیم حال ترانزیستور روشن است اما خروجی یک افت دارد.

اختلاف G , S به اندازه روشن شدن NMOS می‌شود.

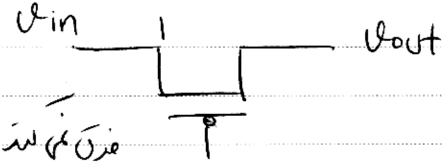
اینجا فقط به D ورودی می‌دهیم
S ورودی ندارد و از آن خروجی می‌گیریم

خروجی را در این جا از کجا می‌بریم.


Control
v_{in}
v_{out}

برای PMOS هم داریم.

برای PMOS برعکس است.



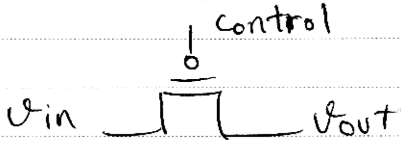
فرض می‌کند

برای v_{DD} ← HI است.

NMOS صفر را خوب رد می‌کند high را خوب رد نمی‌کرد.

Control این جا با صفر روشن است. چون PMOS است.
ولتاژ آستانه < ۰.

$0 - v_{out}$ منفی‌تر از ولتاژ آستانه شود.


control
v_{in}
v_{out}

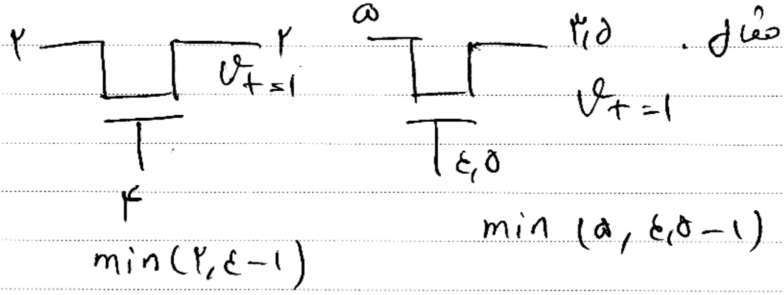
مثال آمد $v_{tp} = -1$
$v_{in} = 0$
Control = 0

$0 - v_{out} < -1$
$\Downarrow$
$v_{out} = 1 \Rightarrow$ ترانزیستور روشن است

if Control $= 0 \Rightarrow v_{out} = v_{in}$

$v_{out} = \max(v_{in}, v_G - v_{tp})$

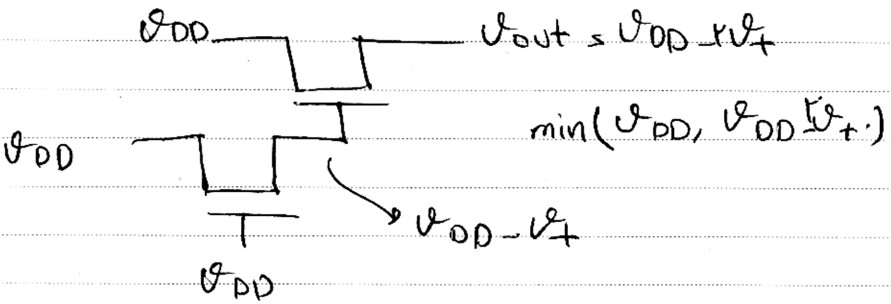
مثال.


۲
۲
$v_t = 1$
۴
۵
۳٫۵
$v_t = 1$
۴٫۵
min(۲، ۴−۱)
min (۵، ۴٫۵−۱)

$v_{out} = \min(v_{in}, v_G - v_t)$

ممکن است مثل ترانزیستور ↑ ها پشت سرهم داشته باشیم.

NMOS رایج‌تر است. با ترکیب P, N انتقال را ایده‌آل می‌کند.


v_{DD}
$v_{out} = v_{DD} - 2v_t$
$\min(v_{DD}, v_{DD} - v_t)$
v_{DD}
$v_{DD} - v_t$
v_{DD}

برای PMOS

v_{in}	Control	v_{out}
0	v_{DD}	HI
v_{DD}	v_{DD}	HI
0	0	$\lvert v_{tp} \rvert$ ← برعکس NMOS است
v_{DD}	0	v_{DD}

PMOS با صفر روشن می‌شود.

برای PMOS ⇐ $v_{GS} < v_t$ باید باشد.

$$G = 0$$

$$\Rightarrow v_G - v_S < v_t$$

$$v_G - v_{out} < v_{tp}$$ ← منفی است

$$0 - v_{out} < \text{''}$$

$$\Rightarrow v_{out} >$$ به اندازه v_{tp} باید تا ترانزیستور روشن شود.

طراحی

می‌خواهیم با استفاده از اینها طراحی را انجام دهیم

" طراحی گیتهای منطقی با ترانزیستور عبور "

ابتدا یک روش کلی ذکر می‌کنیم.

می‌خواهیم گیت AND را طراحی کنیم (جدول بکشیم

A	B	OUT
0	0	0
0	1	0
1	0	0
1	1	1

هر تابعی با جدول داده باشد .

اگر n تا ورودی داشته باشیم .

n-1 ورودی را به عنوان کنترل

مثلا برای این دوتا ورودی داریم یکی را کنترل می گیریم ⇐ B کنترل
↓
G

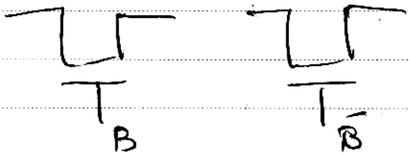
B صفر است یا یک است .

ما به کنترل است ← B است ① است

$\overline{B}$ است 0 است .

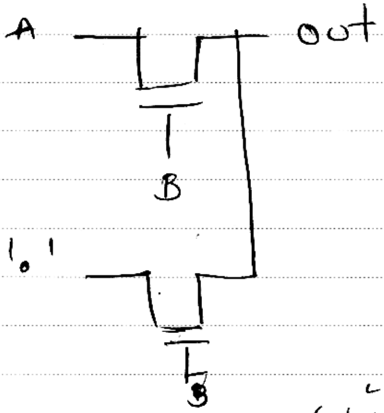
حال فراهم با ترانزیستور عملگر سیستم را می دانیم ورودی هایی که با یک ضرب

همان بیت . And را بدهد .



وقتی B روشن است
↙
1

فرض ← اگر $A = 0$ خروجی 0 (همیشه A را منتقل می کند .
$A = 1$ 1)



برای $\bar{B}$ A هرچی باشه

ضربش صفر است => صفر منطقی دارد

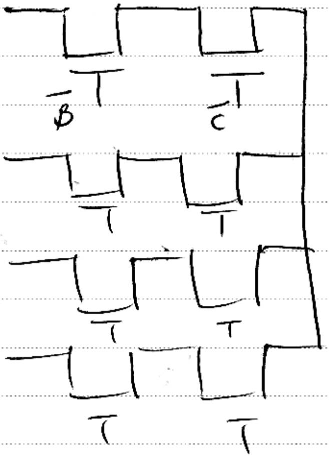
میشه ضربش هم صفر منطقی به ما میدهد.

اگر ۳ تا ورودی A, B, C

داشته باشیم به طور مثال B, C را نشان میدیم

همه حالت ها را نشان میدیم. هر ردیف داشتیم که ضربش بر ۳.

1
BC
$\bar{B}C$
$B\bar{C}$
$\bar{B}\bar{C}$

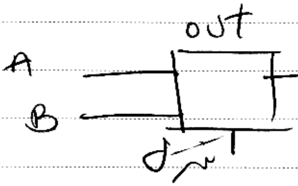


Subject:
Year. Month. Date. ( )

دقت کنید در روش استاندارد ضرب جمع، این صورت در نظر بگیریم

وقتی در روش تحلیلی: حالتی پیش بیاید اگر ورودی‌ها متغیری روشن شد یا ۱ متغیر همزمان روشن شد هر کس برای خودش یک مقدار بگذارد در مسیر. عملاً مدار می‌سوزد یا درست کار نمی‌کند.

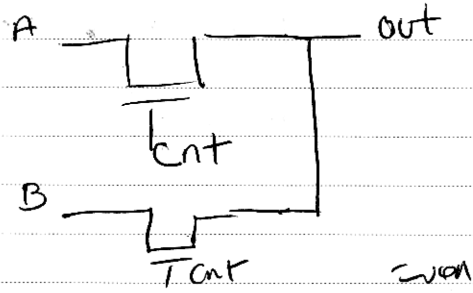
مثلاً می‌خواهیم MUX 2x1 بسازیم



برای هر حالت در هر لحظه زمان یکدام اتفاق می‌افتد به ازای ورودی‌های کنترل می‌دهیم. امکان ندارد که 1 BC و 2 $\bar{B}C$ با هم روشن باشد.

چون C یا صفر یا یک است.

← فرض‌مان این است که ۲ ورودی A,B داریم یک پایه کنترل می‌گذاریم



و یک خروجی داریم

اگر کنترل = 0 خروجی ← A

〃 = 1 〃 ← B

به جای کشیدن جدول با ۳ ورودی و ۸ حالت

Cnt 2⁰ [illegible] استفاده [illegible] که از pass ۲ ۲

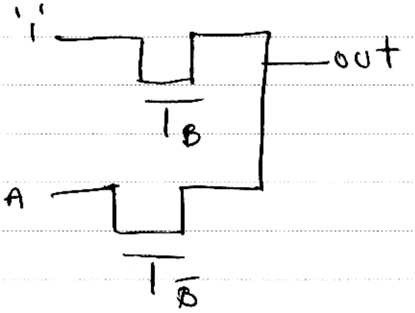
Cnt = ۰ → A منتقل شود.

Cnt = 1 → B ״

آن وقت که دو ترانزیستور ۸ تا می شه اگر از pass tran استفاده کنیم. و در انتها ۵ ولت مطمئن هستیم که خروجی درست می دهد.

گیت OR را هم می توانیم پیاده سازی کنیم.

A	B	out
0	0	0
0	1	1
1	0	1
1	1	1



هر زمانی B ← 1

$\bar{B}$ ← A منتقل می شود.

ترکیب های این بخش آیت الهی خبر است.

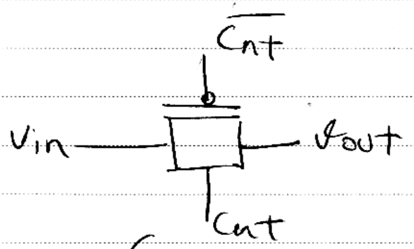
در عمل هم این ها در ترکیب های خوبی ندارد

Subject:
Year. Month. Date. ( )

اگر عدد غلط باشد بنویسیم که عددمان غلط است . واسش 0 در جا بگذاریم .

که دیگه پیشرفت صفر تا عکس را از کتاب گذشته بود در امتحان هم به همین صورت است

گیت انتقال Transmission gate



از ترکیب P و N استفاده کرده

از مزیتهای CMOS را دارد

عکسش هم ایراد ندارد.

در NMOS ما پایه بستر را داریم

مشخص نیست درین یا
روی صد در صد نوشتیم

اشکالش این است که منطق h را خوب

رد نمی‌کند P 0 را خوب رد نمی‌کند.

به همین علت این دو تا را ترکیب می‌کنیم

$Cnt = 1 \Rightarrow V_{out} = V_{in}$ صفر است $\overline{Cnt}$
N و P هر دو روشن است

اشکال قبل را رفع کرده

افت ندارد دقیقاً ورودی را به خروجی می‌رساند.

$Cnt = 0 \Rightarrow V_{out} = HI$
خروجی ندارد.

N با 0 خاموش می‌شود P ← notاش ← 1 خاموش

فراهمی گیت های منطقی با گیت انتقال

مثال. یک fulladder را فراهم با گیت انتقال فراهم می‌کنیم.

FA جدول زیر را دارد. نشانش را می‌دهیم

	A	B	Cin	S	C
$\bar{A}\bar{B}$	0	0	0	0	0
	0	0	1	1	0
$\bar{A}B$	0	1	0	1	0
	0	1	1	0	1
$A\bar{B}$	1	0	0	1	0
	1	0	1	0	1
AB	1	1	0	0	1
	1	1	1	1	1

روش رسم همان روشی است که در گیت‌های pass+ ، ترانزیستور عبور داشتیم.

یکبار هم فراهم S C را باید بیاوریم همه را پر می‌کنیم.

n تا ورودی

n −1 ← پایه های کنترل

یعنی pass+ رسم می‌کنیم سپس nmos pmos ها را رسم می‌کنیم

A و B ← مشترک می‌گیریم .

هر کدام را می‌خواهیم می‌توانیم مشترک بگذاریم چون محدودیت نداریم اگر داشتیم

همان طور عمل می‌کردیم

$\bar{A}\bar{B}$ می‌خواهد که بعدازش $0 \leftarrow S$

ورودی $0 \leftarrow C$

$1 \leftarrow S$

ورودی $1 \leftarrow C$

$\Leftarrow$ $\bar{A}\bar{B}$ دقیقاً Cin را وارد می‌کنیم .

حالت بعدی $\bar{A}B$

این بار $0 \leftarrow S$ $1 \leftarrow S$

$1 \leftarrow C$ $0 \leftarrow C$

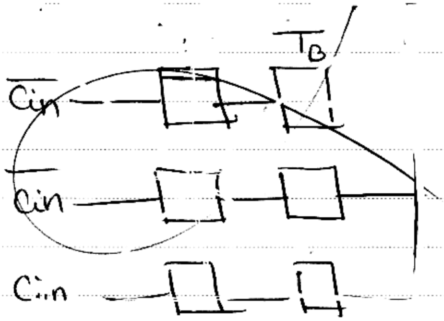
$\Leftarrow$ Cin را داریم می‌دهیم .

$A\bar{B}$ $0 \leftarrow S$ $1 \leftarrow S$

$1 \leftarrow C$ $0 \leftarrow C$

$\Leftarrow$ $\overline{Cin}$

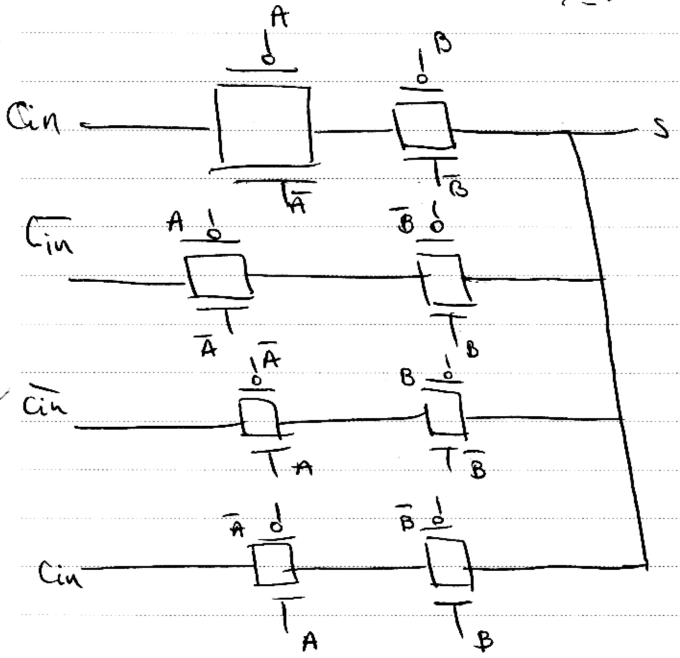
Cin منتقل می‌شود.　　AB


T_B
$\overline{Cin}$
$\overline{Cin}$
Cin

هرچی در N است
not اش را در P می‌دهیم.

اگر جمع‌کننده بخواهیم با PT
فقط N ها را اثر بدید

PS ← دو pmos هم برای P.


A
B
Cin
S
$T_{\overline{A}}$
$T_{\overline{B}}$
$\overline{Cin}$
A
$\overline{B}$
$\overline{A}$
T_B
$\overline{A}$
B
$\overline{Cin}$
T_A
$T_{\overline{B}}$
$\overline{A}$
$\overline{B}$
Cin
T_A
T_B

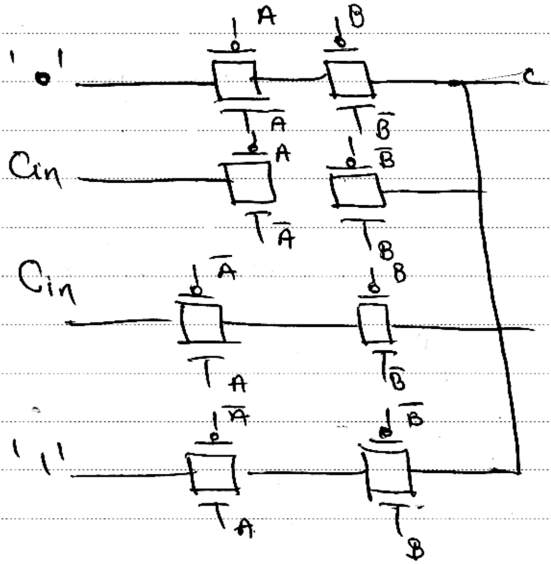
برای carry

$\bar{A}\bar{B}$ C ← 0 ؛ Cin کاری ندارد

$\bar{A}B$ 1 Cin را عبوراست داده کنیم.

$A\bar{B}$ 1 Cin را " "

AB 1 منطقی را وارد می‌کنیم؛ Cin کاری ندارد.


A
B
'0'
C
$\bar{A}$
$\bar{B}$
Cin
A
$\bar{B}$
$\bar{A}$
B
Cin
$\bar{A}$
B
A
$\bar{B}$
'1'
$\bar{A}$
$\bar{B}$
A
B

اگر ورودی‌ها A و B و Cin باشد.

دسته‌بندی عوض می‌شود.

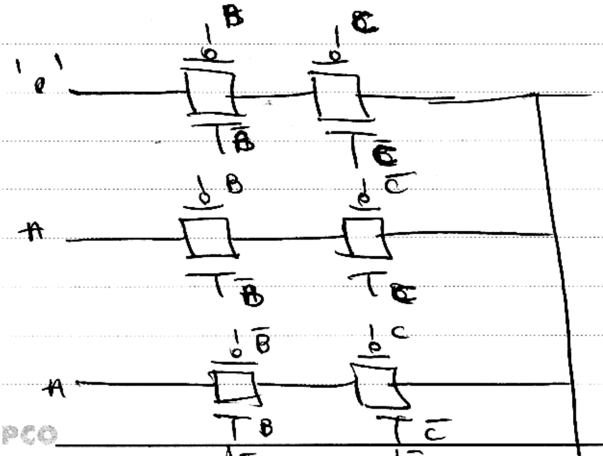
A	B	C	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0			0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1			1	1

$\bar{B}\bar{C}$ ← carry ربطی به ورودی ندارد '0' منطقی

$\bar{B}C$ ← عدد ~~A~~ را می‌دهم

$B\bar{C}$ ← " " "

BC ← '1' منطقی را می‌دهم.


B
C
'1'
$\bar{B}$
$\bar{C}$
B
$\bar{C}$
A
$\bar{B}$
C
$\bar{B}$
C
A
B
$\bar{C}$
c

تمام اینها یکی با ترانزیستور
عبور طراحی می‌کند با استفاده
هم می‌توان طراحی کرد فقط
P باید اضافه کنیم.

Subject:
Year. Month. Date. ( )

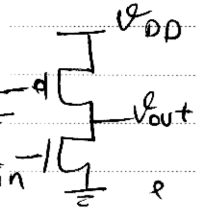
منطق شبه NMOS

ترکیب NMOS و CMOS به شبه nmos را جا می‌دهد.

nmos به pullup ضعیف ببرد.

cmos تراز nmos است اما به ازای آن nmos آن

هم pmos باید می‌داشتیم



در بالاکش pmos هم نداریم. و به زمین وصل کردند 6 به

همیشه روشن بود. ترانزیستور بالایی به افزایش، کاهش همیشه روشن

nmos هم ورودی می‌دهیم و خروجی می‌گرفتیم.

ظاهراً که شبیه not جا می‌دهد. اشکالش این است. وقتی ترانزیستور پایینی روشن

است خروجی کاملاً صفر نبود و دقیقاً یک را هم نمی‌دهد.

اشکال: در نقطه میانی V_m به اشتباه می‌رفت. عملکرد بالا با ترانزیستورهای

به درستی به خطر می‌رفت.

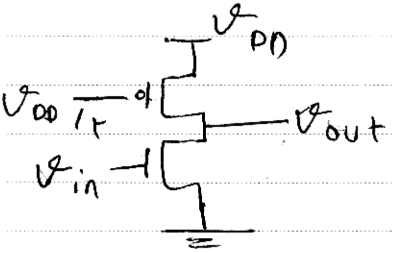
← $V_{DD}/2$ به V_{bios}

این کار را کردند.

شبه nmos

این فرم دیگه نداره دیگه .

هر تابعی را بخواهیم می‌توانیم با این طراحی کنیم .



$$v_{bios} = v_{DD}/2$$

$\boxed{\text{bios}}$

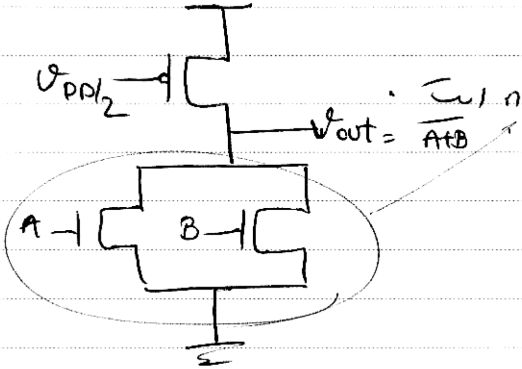
گاهی اوقات طراحی مان برای این فرض است که ترانزیستور حلقه باز .

کارشون درست باشه ؟ اشباع هم در این کار را با پارامترهای اینجا انجام دهیم.

به این عمل می‌گوییم bios کردن . یعنی با این فرض که مقاومت ها

طوری تنظیم شده که مدنظرمان را دارند .

گیت NOR با شبه nmos :

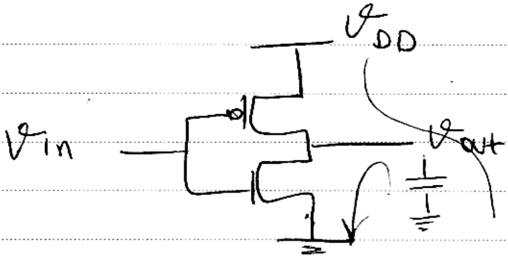


bios ← pullup

pd ← مثل nmos است .

منطق CMOS پویا :

مدار ایستا



این مدار در هر لحظه از زمان پایینی یا بالایی یکدام روشن است خروجی از مسیر پایینی یا بالا به زمین یا تغذیه وصل بوده است.

مدار ایستا باعث می‌شود اگر که در V_{out} یک خازن داریم در یک حالتهایی دشارژ می‌شود و در حالتهایی هم شارژ می‌شود.

در هر صورت در هر لحظه از زمان V_{out} به زمین یا منبع تغذیه راه دارد

اما در مدار پویا به این صورت نیست

مدار پویا

حالتی ممکن است وجود داشته باشد خروجی از هیچ مسیری به هیچکدام تغذیه یا زمین وصل نباشد.

مصرف نباشد اما حتی لایه ولتاژ دارد.

تفاوت دینامیک اگر فرض هم این جا وصل نباشد می‌گفتیم ← $\hat{HI}$ شناور است

ولی در دینامیک چون از زمان‌های خیلی کوچک بررسی می‌کنیم.

یک سیگنال clk داریم. به صورت یک پالس با دوره تناوب مشخص

اعمال می‌شود اگر clk = 0 شد

ترانزیستور پایینی ← خاموش است.

pmos

روشن ↓ هیچ ربطی به ورودی ندارد.

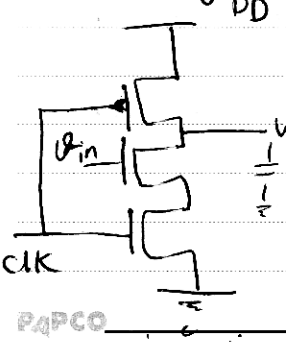
v_{out}

تا جایی که clk = 1 شود.

پایینی nmos است با clk = 1 روشن

بالایی pmos با " " خاموش

حالت 2) مد دوم یا همان مد پایینی است.



اگر $v_{in} = 1$

دشارژ می‌شود،

$v_{in} = 0$ ، clk = 1

به هیچ صورتی. مقدار h خود را حفظ کرده اما این حفظ کردن در حد نانوثانیه است.

در یک بازه کوچک خازن روشن مانده مقدارش هم با وجود $v_{in}=0$ است مقدارش را حفظ کرده به محض اینکه $clk=0$ شد. انگار به تعویق داریم می‌کنیم. چونکه h داریم می‌دهیم در این ناحیه مدار حالت شارژ دارد وقتی $clk=1$ می‌شود سوئیچ v_{in}.

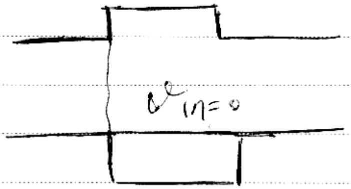
می‌توانیم

اگر دوره تناوب مناسب باشد. می‌توانیم حفظ دیتا ولتاژ می‌دهیم می‌توانیم از مدار استفاده کنیم توان مصرف نمی‌کنیم ولی از این مقدار ولتاژ در بازه‌ی کوچکی نگه داشته.

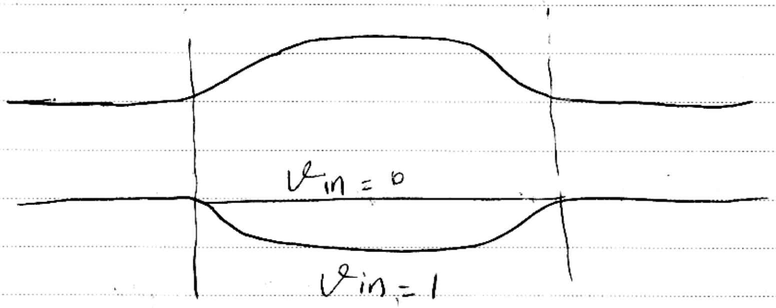
از نظر کاربری کارایی زیادی داشت چون در مصرف توان خیلی صرفه‌جویی می‌شود.

$clk=1$ nmos روشن $\begin{cases} v_{in}=1 & \text{دشارژ} \\ v_{in}=0 & \text{نگه شارژ} \end{cases}$

$clk=0$ pmos روشن $v_{out}=1$ مدت زمان



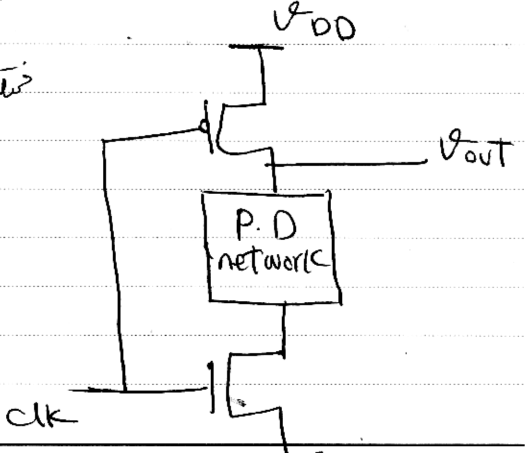
آخر سر گفته شد مدار دوپایی پیش شارژ بود.



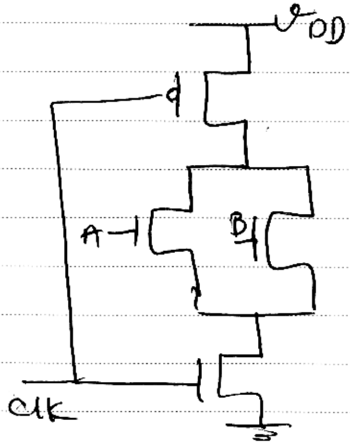
بعدا این را نخواهیم توانیم ادامه دهیم

فرم کلی مدارات پیش شارژ

در فرم کلی یک nmos برای clk داریم و یک شبکه P.D
pmos
از nmos ها استفاده می کنیم که تابع را پیاده کنیم . تا ورودی های مختلف را دارا شود.


v_{DD}
v_{out}
P.D network
clk

مثال. مدار NOR را با پیش شارژ رسم کنید.



در هنگامی که $clk=0$ ارزیابی انجام
می دهیم.

↓

$V_{out}=1$ ربطی به ورودی
ندارد

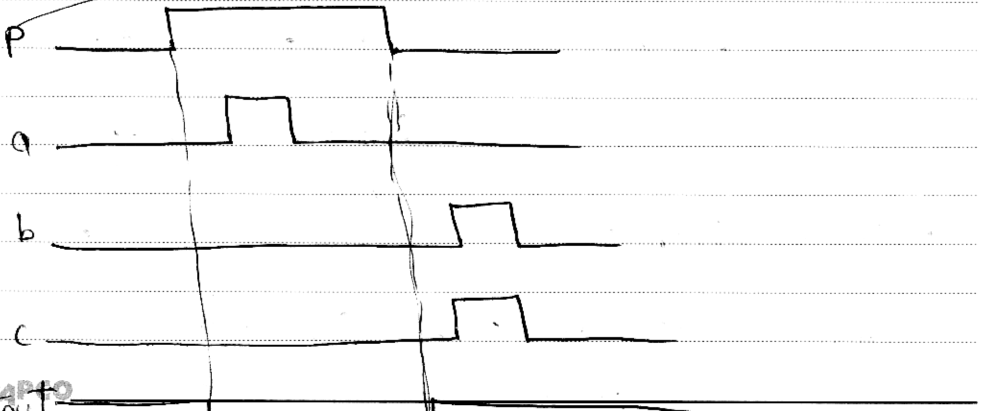
در حالتی که ارزیابی انجام دهیم
$clk=0$
NOR پیاده سازی می شود

(مثل [illegible] H مهم شود)

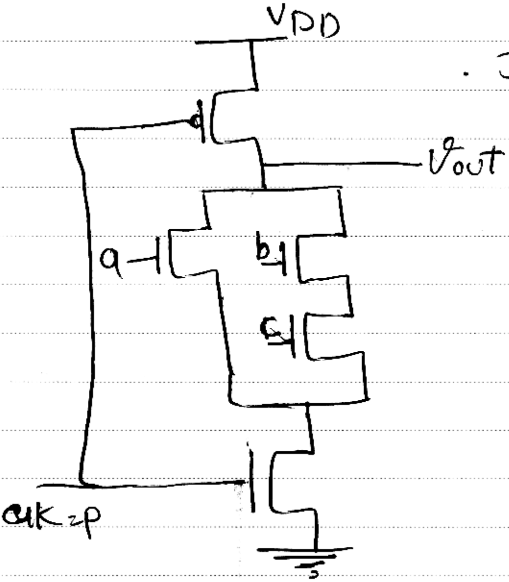
مثال. تابع $F = \overline{a+bc}$ را با مدار دومینوی پیش شارژ رسم کنید.

وضعیت را برحسب ورودی ها بدست آورید.

نقش clk را دارد



clk = 0



بالایی روشن پایینی خاموش است .

p هم clk است لازم نیست

که اسمش clk باشه .

می‌خواهیم خروجی را ثابت بیاوریم

clk = 0 pullup داریم

↙

Vout = h

clk = 1 P.U = خاموش
P.D = روشن

می‌خواهیم ببینیم مداری هست که خروجی را حفظ کند. (مرحله ارزیابی)

در اولین لحظه a,b,c = 0 ⇐ هر سه ترانزیستور خاموش است .

تا جایی که a = 1 شود .

مدار شارژ پیدا می‌کند. حالا شارژ ادامه پیدا می‌کند تا زمانی که clk عوض شود.

clk هر وقت وارد شد ← clk = 0 خروجی‌ها P.U ←

h

تأثیری روی پایینی کاری ندارد.

Subject:

Year. Month. Date. ( )

وقتی a از یک به صفر عوض شود. تا اون قدر روشن بود بعداً خاموش شد.

درست است اما از بین رفتن وقتی ست ریست شد داشتیم.

a خاموش شده اما دوباره نمی‌تواند out ← high شود.

علت: مقدار b,c ذخیره کرده بود در خروجش از دست داد.

و امکان مقدار گرفتن ندارد تا clk بعدی بیاید.

وقتی low شد و مقدارش را از دست دادیم در بازه‌ی ارزیابی دوباره نمی‌تواند high شود

هر مقدار را اضافه کند تا به مرحله‌ی شارژ

برسد.

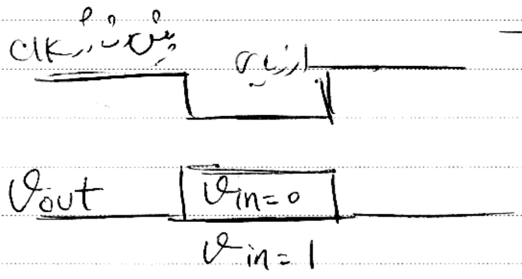
پس b, c تغییر نکرده اما چون در مرحله‌ی ارزیابی نیست. out تغییر نکرده

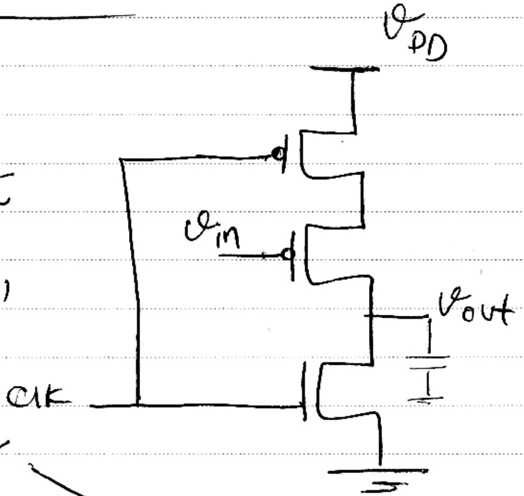
آمد b,c تغییر می‌کرد در مرحله‌ی ارزیابی ولی چون در مرحله‌ی شارژ

است تغییر نمی‌کند. اثری ندارد. دوباره out ← سالمش.

و البته کلاک اثری ندارد.
تغییر ورودی

مدار پیش دشارژ


clk پیش شارژ
ارزیابی
Vout
Vin = 0
Vin = 1


V_{DD}
V_{in}
V_{out}
clk

تنها ترانزیستوری است که از pmos استفاده می‌کنیم .

clk = 1 برعکس حالت قبل → خروجی . P.U خاموش

P.D روشن

out = 0

clk = 0 P.U = روشن

P.D = خاموش } V_{in} بی‌اثر می‌شود

$V_{in} = 0$ → خروجی می‌تواند High شود

$V_{in} = 1$ → خروجی مقدار صفر خودش را حفظ می‌کند .

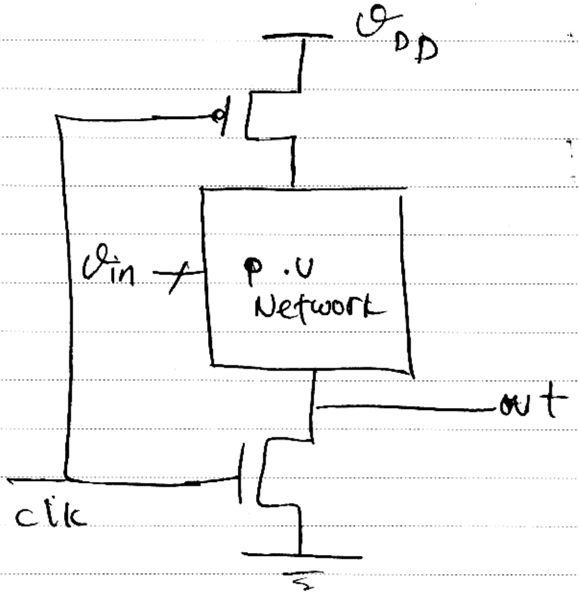
اگر در ابتدای عملیات چک کردیم می‌فهمیم hight است.

اما در پایان عملیات چک می‌کنیم عمق یک خازن در out داریم که اشتباه

دشارژ کردیم مقدار صفر دست رسیده مدار دارد در انتهای بازه‌ی کوتاه

صفر شدنش را صفر کرده.

فرم کلی مدارات پیش دشارژ pre discharge

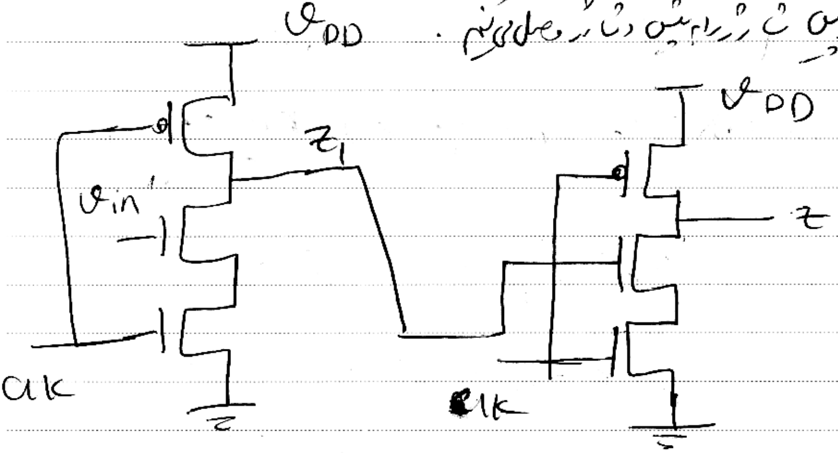


Subject:
Year. Month. Date. ( )

اتصال زنجیره‌ای : cascade

اتصالات زنجیره‌ای مدارات پیش دشارژ رو دارد.

خروجی مدار پیش شارژ را به پیش دشارژ وصل می‌کنیم.



اگر یک clk داشته باشیم کلاکی که وارد سیستم می‌شود، شکل مربعی کامل نیست

در واقعیت هیچ وقت مربعی کامل نداریم.

مقداری زمان طول می‌کشد تا L H برود یا برعکس؛ همین خاطر

مقداری تأخیر داریم و مربعی کامل نیست.

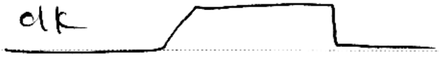
در عمل‌هایی که انجام می‌دهیم سیگنال داریم در هر لحظه زمانی کار می‌کنیم اگر مشکل

در سطح نانو کار می‌کند و تأخیر در حد نانو ثانیه است عملاً در کار ما تأثیر

ندارد و از آن صرف‌نظر می‌کنیم

Subject:

Year. Month. Date. ( )



ما در لبه پایین حس می‌کنیم

⇐ تأخیر کم جلوی تأثیرگذار است .

فرض کنید v_{in} با یک تأخیر اعمال شد .

قبلی $clk = 0$ (مدار طبقه اول) جدا بوده که است .

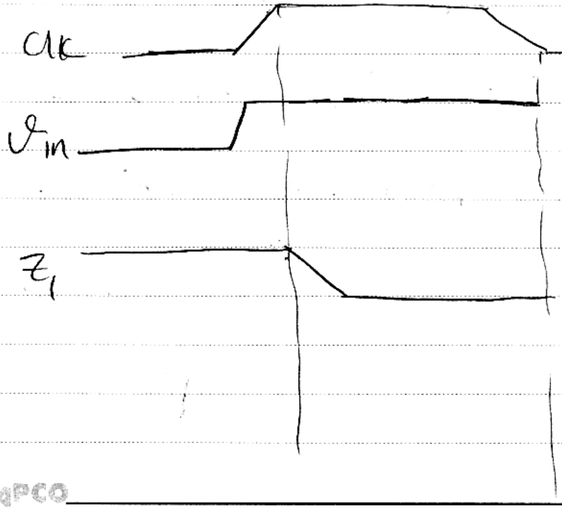
$z_1 \to H$

clk نزدیک به $clk = 1$ آغاز تغییر دارد

$clk = 1$ شد . نتوانند تغییر یابند روشن می‌شود .

تا $v_{in} = 1$ ⇐ خازن دشارژ می‌شد .

ولی با یک تأخیر انجام می‌شد ، مقدار صفر نیز تا جایی دوباره



کلاک تغییر وضعیت می‌دهد

مدار هم حالت

ساکن نشده بود .

z_2 هم اساساً z پیش تر بود

ادامه تا جایی که کلاک ۱

شود ⇐ نتوانند تغییر یابند

روشن .

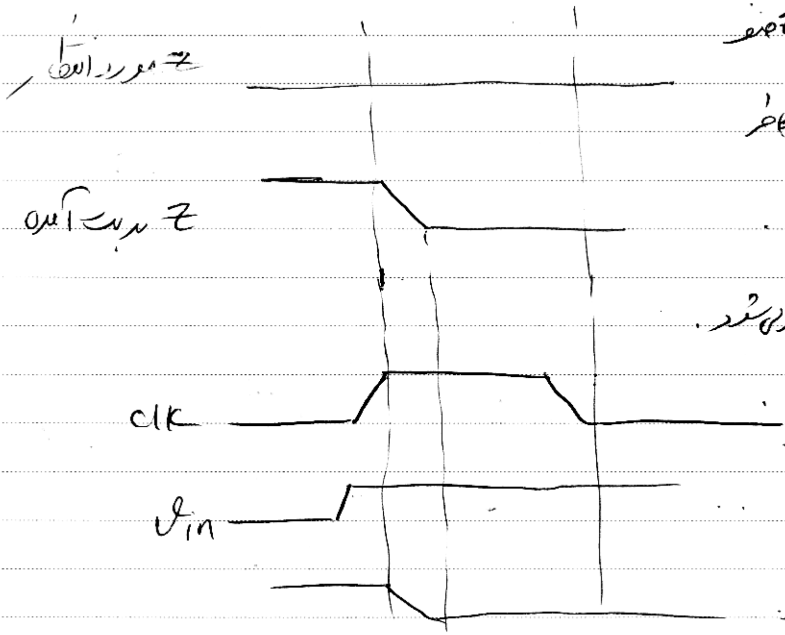
مقدار واقعی z_1 صفر است. اما به خاطر تأخیر مقدار قبلی‌اش یعنی

$z_1=1$ می‌بیند. ترانزیستور روشن می‌شود. دشارژ می‌شود.

در حالی که خروجی ناخواسته است خروجی مطلوب ما نیست.

مقدار صفر نمی‌دهد.

z نهایی



مقدار واقعی z_1 صفر

است. =۱ به خاطر

تأخیر آن می‌بیند.

و z نهایی صفر می‌شود.

اگر تأخیر را نمی‌دید

کلاک که ۱ بود

خروجی ۱ بود.

به محض اینکه کلاک

0 می‌شد. اگر نمی‌دید z_1 صفر است و مقدار z ۱ خروجی‌اش را حفظ می‌کرد.

اما low شده چون دید z_1 صفر شده کل تأخیر باعث شده خروجی اشتباه

باید تدبیری برای حل این مشکل بیاوریم ؟

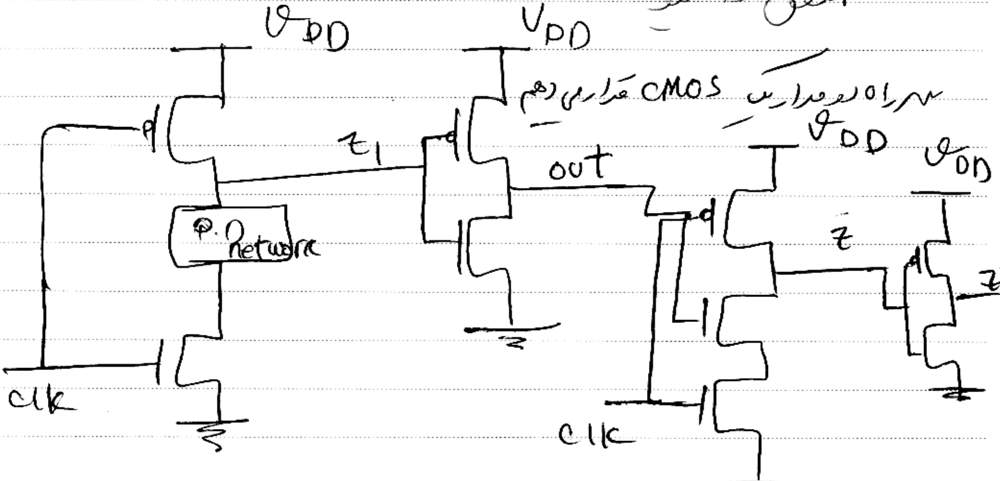
۲ راه پیشنهاد می‌شود: ① اگر در طبقه‌ی اول با کلاک کار می‌کنیم. طبقه‌ی بعد با $\overline{clk}$ کار کند اگر پیش شارژ است پیش دشارژ شود.

② منطق دامینو

یکی از راه‌های پیشنهادی برای اتصال چندین مدار از این نوع

منطق دامینو

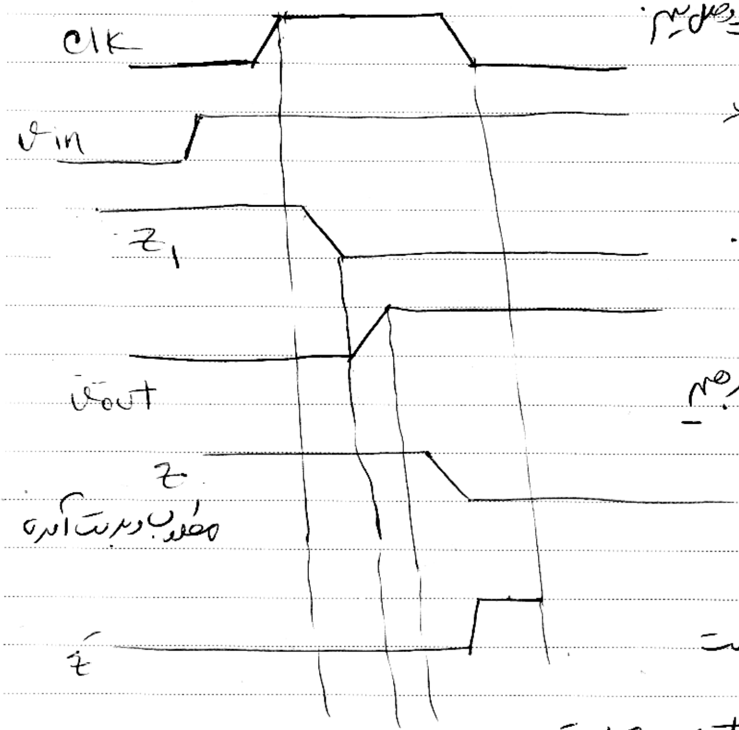
سر راه یک مدار CMOS قرار می‌دهم


VDD
VDD
VDD
VDD
Z1
out
P.D network
Z
Z'
clk
clk

Z قبل از این است not کنم

این ۲ تا دامینو است.

Subject:
Year. Month. Date. ( )



اگر بخواهیم بار اضافه وصل کنیم:
عملاً یک P.D
network
nmos است.
خروجی را هم می‌توان
از یک not عبور بدهیم
مثل z تا را منفی
ثبت شده هم وصل کنیم.
تأثیرش به این صورت است

که V_{out} ← not z_1 است.

high می‌شود تا به بعد برسد. مشکلی که در clk صفر ← z=h , z=h

مشکلی که clk تغییر وضعیت می‌دهد. و ورودی تغییر می‌کند $V_{out}=0$ ← z

روشن نمی‌شود تا 0 ← z شود 1 ← z خروجی روشن ← دشارژ انجام

می‌شود چون اینکه تأخیر کوچکی دشارژ شده مشکلی نداریم

ابتدا چون برد
مورد انتظار ما را خروجی معکوس
است. تأخیر که بیش از موعد دشارژ شده.

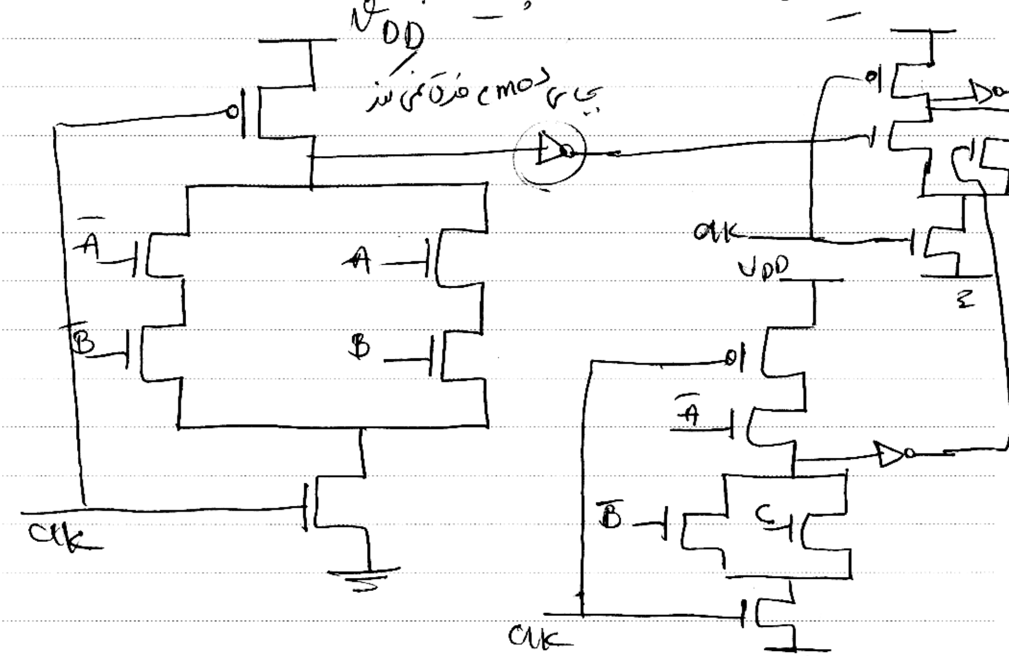
ج) عملاً not آن منفیه است . که خروجی اصلی همان است .

ب) از این not باید بگذاریم .

با توجه به شرایط جوردانی مدار ج مورد نظر

با ج بدست آمده یکی است . با توجه به اینکه مثبت مدار عوض شود .

تمرین : تابع F چیست؟


VDD
بجای دو cmos فرق می کند
A̅
B̅
A
B
Clk
Clk
VDD
A̅
B̅
C
Clk
ج

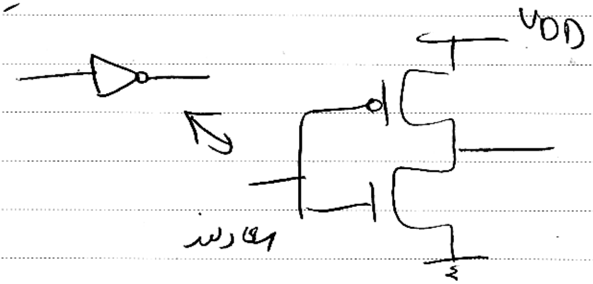
فقط فرض می کنیم باشد هر تابعی که بخواهیم پیاده سازی کنیم

Subject:
Year. Month. Date. ( )

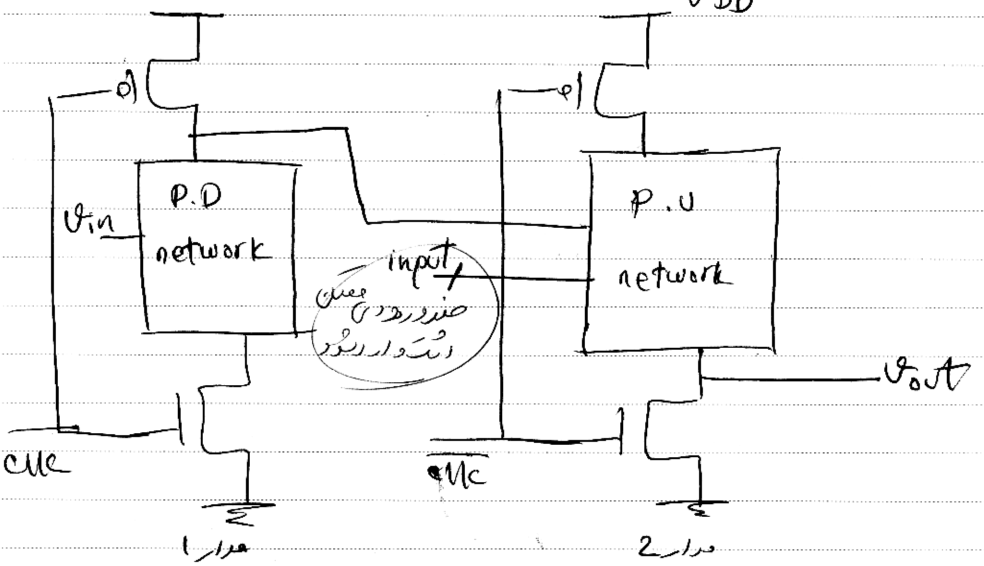
برای مدار

همیشه نمی‌بینیم مشکل اجرایی پیش آمده باشد. اگر دارد بنویسیم.

clk صفر یا یک باشد. ضرورتی است بنویسیم.


VDD
معادلند

منطق np_cmos


VDD
VDD
P.D
network
P.U
network
V_{in}
input
صفر و رو دوم یعنی
اشتباه را دارد
V_{out}
clk
$\overline{clk}$
مدار 1
مدار 2

در طبقه اول n ها را می‌بریم
" " دوم p " "