به نام خدا

lC Se13 Dr. Shaabaany

Matin Barekatain

priority encoder:

highest w_n - - - - w_0 lowest

- ورودی‌ها دارای اولویت هستند
- خروجی نشان دهنده‌ی ورودی Active (1) شده‌ای است که دارای بالاترین اولویت است.

اولین بیت ورودی که 1 شده است

w_3	w_2	w_1	w_0	y_1	y_0	z
0	0	0	0	d	d	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

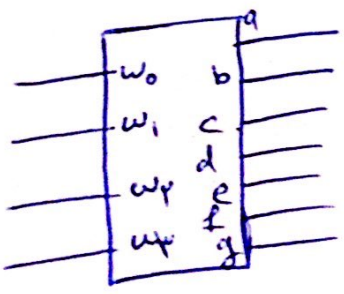
$y_0 = \overline{w_3}\,\overline{w_2}\,w_1 + w_3$

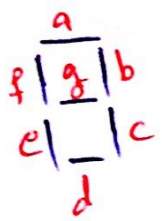
$y_1 = \overline{w_3}\,w_2 + w_3$

$z = w_0 + w_1 + w_2 + w_3$

مبدل‌های کد: code-converter

BCD-to-7seg:





w_3	w_2	w_1	w_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
⋮										

مبدل باینری به کد gray:

صفحه‌ی بعد :-D

b_2	b_1	b_0	g_2	g_1	g_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

* hamming distance ← باید بیت به بیت XOR کنیم و حاصل را هم جمع کنیم.

* الگوریتم ساخت گری کد:

۱) ابتدا 0 را اینگونه می نویسیم.

۲) سپس زیر آن آینه قرار می دهیم.

۳) در کنار دو کد بالایی 0 و در کنار دو کد پایینی 1 قرار می دهیم.

۴) هر ستون را مجدداً آینه ای قرینه می کنیم:

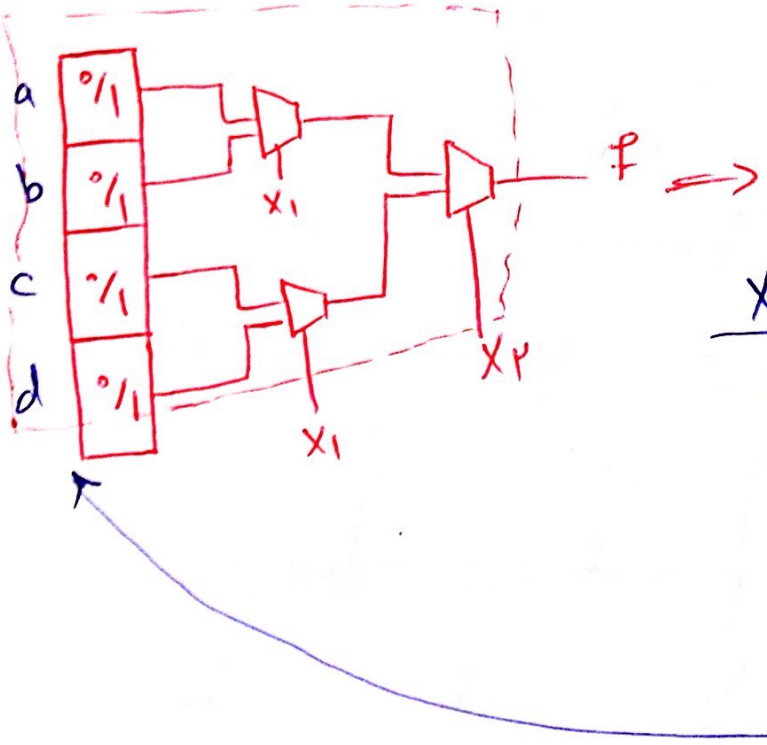
۵) در کنار ۴ کد بالایی صفر و در کنار ۴ کد پایینی 1 قرار می دهیم.

⇒ کد ساخته می شود.

$$g_2 = b_2$$
$$g_1 = b_1 \oplus b_2$$
$$g_0 = b_0 \oplus b_1$$

پیاده‌سازی برروی LUT : (Look-Up-Table)

- یک حافظه‌ی تک بیتی را در نظر بگیرید. %1
- یک LUT مجموعه‌ای از حافظه‌های تک بیتی است.
- به کمک یک LUT، 2^n بیتی می‌توان هر تابع n متغیره را پیاده‌سازی کرد.



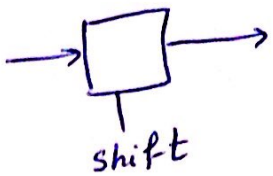
* ادعا می‌شود همه‌ی توابع دو متغیره با این سیستم قابل اجرا است:

X_1	X_2	f
0	0	a
0	1	b
1	0	c
1	1	d

* تغییر f باعث تغییر a و b و c و d می‌شود.
* مزیت بزرگ LUT این است که با آن هر تابعی را می‌توان پیاده‌سازی کرد.

مدار shift به صورت است:

input
w_3, w_2, w_1, w_0



$y_3 y_2 y_1 y_0$

shift = 0 → No Action
= 1 → right shift (افزودن صفر)

assign Y = X >> 1;

Barrel shifter :

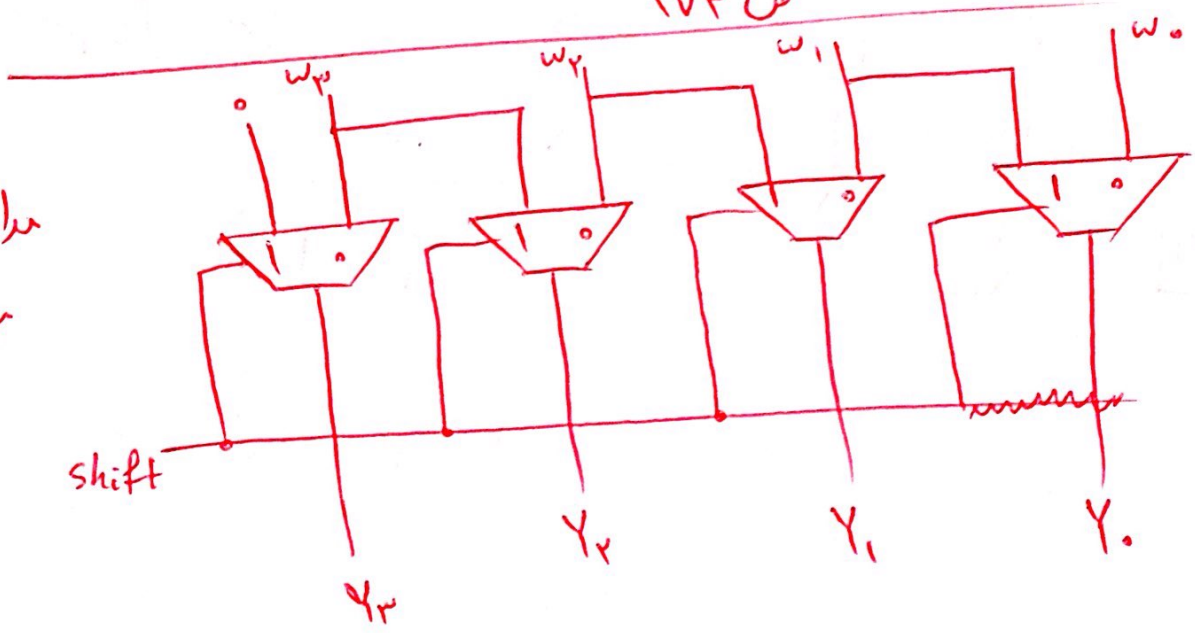
← یک حالت کلی تر یک shifter که تعداد شیفت آن قابل کنترل است و بیت‌های خالی توسط چرخیده شده‌ی بیت‌های قبل پر می‌شود.

S_1	S_0	Y_3	Y_2	Y_1	Y_0	
0	0	w_3	w_2	w_1	w_0	→ no shift
0	1	w_0	w_3	w_2	w_1	
1	0	w_1	w_0	w_3	w_2	
1	1	w_2	w_1	w_0	w_3	

assign Y = X >> n;

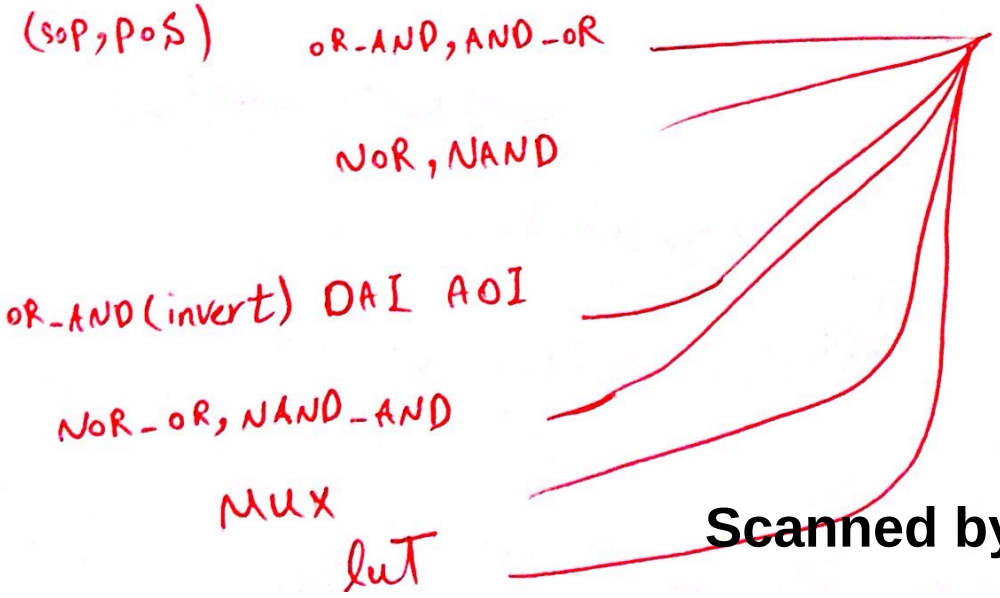
پیاده‌سازی → brown

ص ۳۷۴



مدار شیفت محوری

پلتفرم‌های پیاده‌سازی مدارهای ترکیبی :



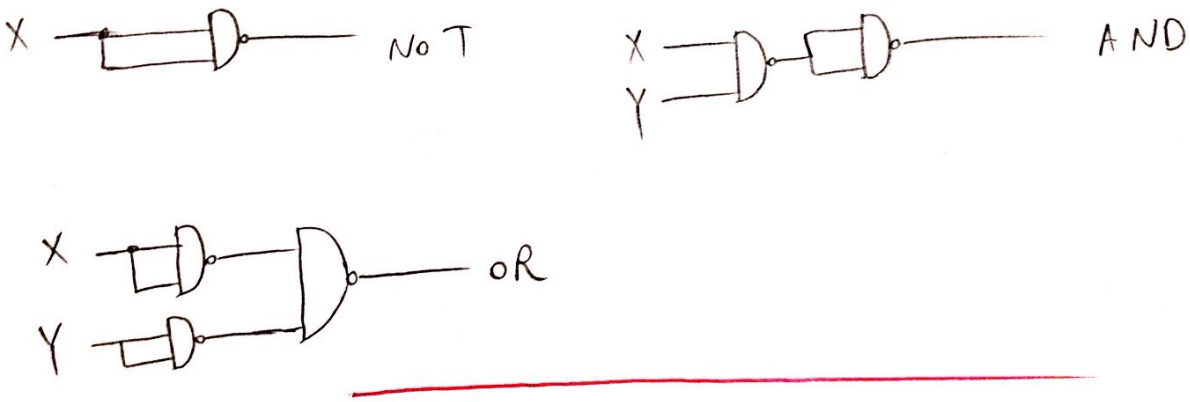
- قبلاً دیدیم هر تابع منطقی را می‌توان با گیت‌های AND ، OR ، NOT پیاده‌سازی کرد.

گیت کامل: یک گیت را کامل گوییم اگر بتوان گیت‌های AND ، OR ، NOT را با آن ساخت

مانند: NAND

← هر تابع منطقی را می‌توان بوسیله‌ی یک گیت کامل پیاده‌سازی کرد.

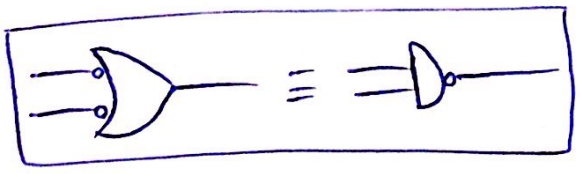
مثال: نشان دهید NAND یک گیت کامل است.

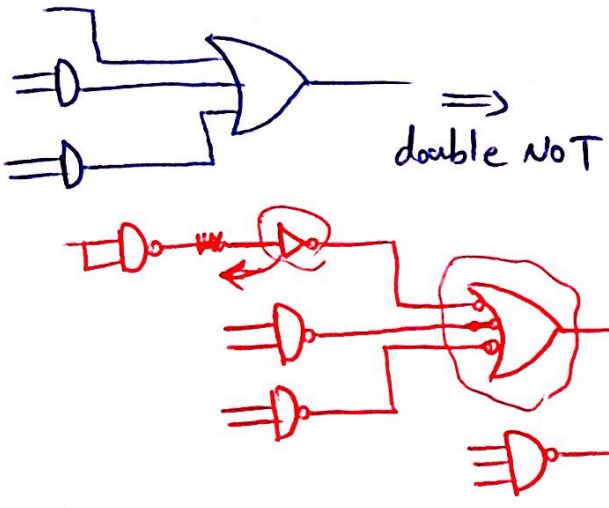


پیاده‌سازی برمبنای NAND و NOR:

- برای پیاده‌سازی براساس NAND ← از AND-OR استفاده می‌کنیم.

ایده اصلی:





مثال: