

## • پیاده‌سازی توابع دودویی با استفاده از دیکدر

ورودی‌ها			خروجی‌ها							
a	b	c	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

- یاد آوری:
- دیکودر ۳ به ۸ با خروجی یک فعال

$$D_0 = a'b'c'$$

$$D_1 = a'b'c$$

$$D_2 = a'bc'$$

$$D_7 = abc$$

- هر خروجی متناظر با یک مین ترم است.

به نام خدا

## مدارهای منطقی

### مدارهای ترکیبی (ادامه)

- دیکودر ۳ به ۸ با خروجی صفر فعال

ورودی‌ها			خروجی‌ها							
a	b	c	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

$$D_0 = (a'b'c')' = a + b + c$$

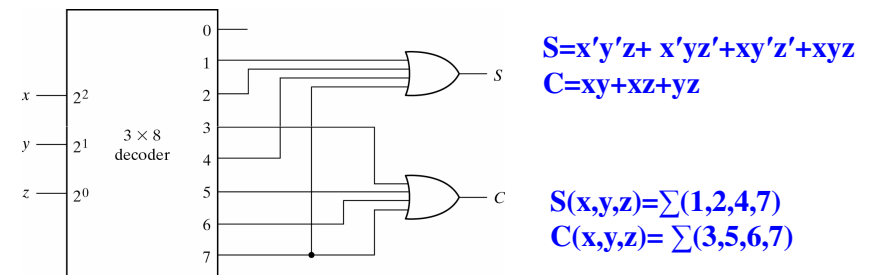
$$D_1 = (a'b'c)' = a + b + c'$$

$$D_7 = (abc)' = a' + b' + c'$$

- هر خروجی متناظر با یک ماکس ترم است.

## مثال: پیاده‌سازی تمام جمع‌کننده با دیکدر

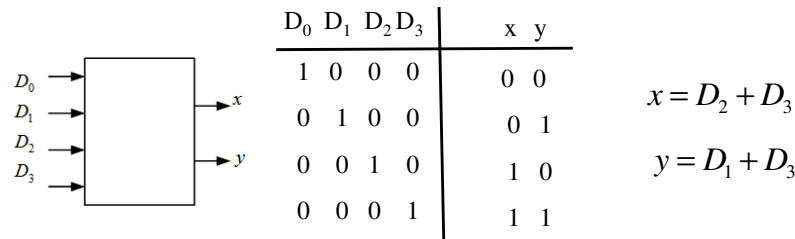
روابط مربوط به تمام جمع‌کننده:



دقت شود که با استفاده از دیکودر با خروجی صفر فعال و گیت AND نیز می‌توان توابع را پیاده‌سازی نمود.

## Encoder یا رمز کننده

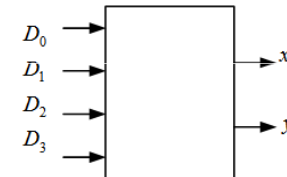
- عمل آن دقیقاً برعکس دیکدر است.
- مداری ترکیبی است که حداکثر  $2^n$  ورودی و  $n$  خط خروجی دارد. در خروجی کد دودویی متناظر با مقدار ورودی فعال تولید می‌شود.



- با استفاده از چند گیت OR براحتی ساخته می‌شود.

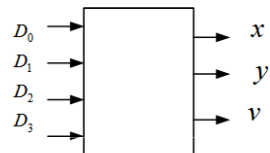
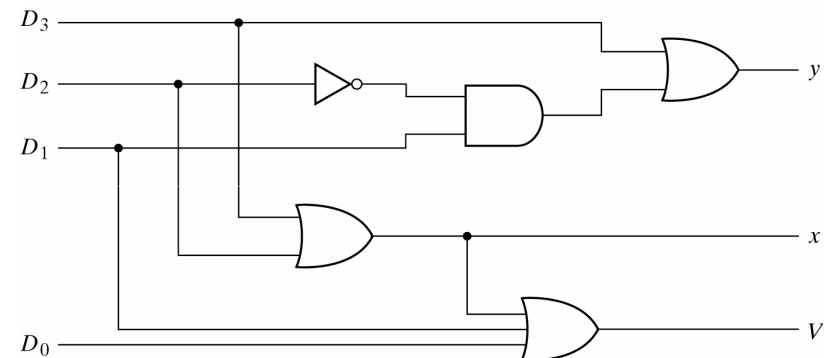
## Encoder با حق تقدم

- مدار رمزکننده‌ای است که شامل تابع حق تقدم می‌باشد.
- اگر دو ورودی یا بیشتر بطور همزمان مساوی ۱ شوند، ورودی با بالاترین حق تقدم برتری خواهد داشت.



- بالاترین حق تقدم متعلق به  $D_3$  و سپس به ترتیب به  $D_2$ ،  $D_1$  و نهایتاً  $D_0$  می‌باشد.

- به دست آوردن معادلات مربوط به خروجیها

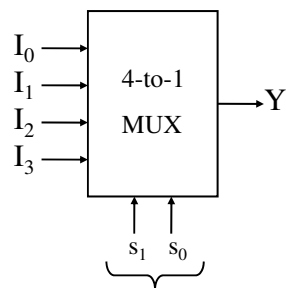


ورودیها				خروجیها		
$D_0$	$D_1$	$D_2$	$D_3$	$x$	$y$	$V$
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

- یک خروجی دیگر هم برای مدار در نظر گرفته می‌شود که همیشه مقدار آن ۱ است بجز حالتی که همه ورودیهای انکودر برابر با صفر باشند. بعبارت دیگر ۱ بودن آن نشان دهنده معتبر بودن خروجی انکودر است.

## مالتی پلکسر (Multiplexer) - تسهیم کننده

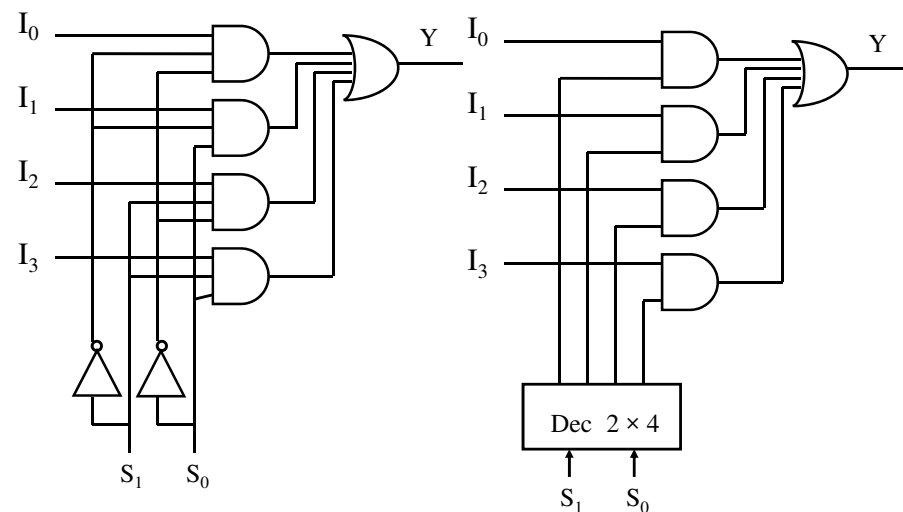
- مالتی پلکسر (انتخابگر داده) یک مدار ترکیبی است که دارای  $2^n$  ورودی و  $n$  پایه کنترلی و یک خروجی است.
- مالتی پلکسر در هر لحظه یکی از چند خط ورودی را انتخاب و آن را روی خط خروجی ظاهر می سازد. (مانند سوییچی عمل میکند که روی ورودیها حرکت می کند)



$S_1$	$S_0$	$Y$
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$

$$Y = S_1' S_0' I_0 + S_1' S_0 I_1 + S_1 S_0' I_2 + S_1 S_0 I_3$$

$$Y = S_1' S_0' I_0 + S_1' S_0 I_1 + S_1 S_0' I_2 + S_1 S_0 I_3$$

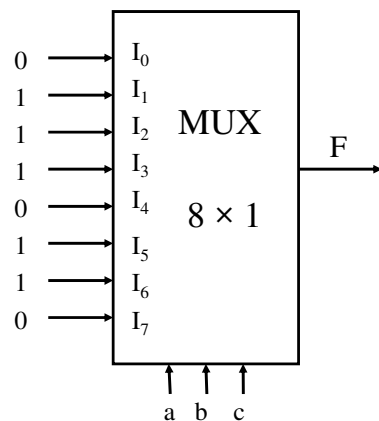


## پیاده سازی توابع دودویی با استفاده از مالتی پلکسر

- حالت اول: تعداد متغیرهای تابع با تعداد پایه های کنترلی مالتی پلکسر برابر باشد.

$$F(a, b, c) = \sum m(1, 2, 3, 5, 6)$$

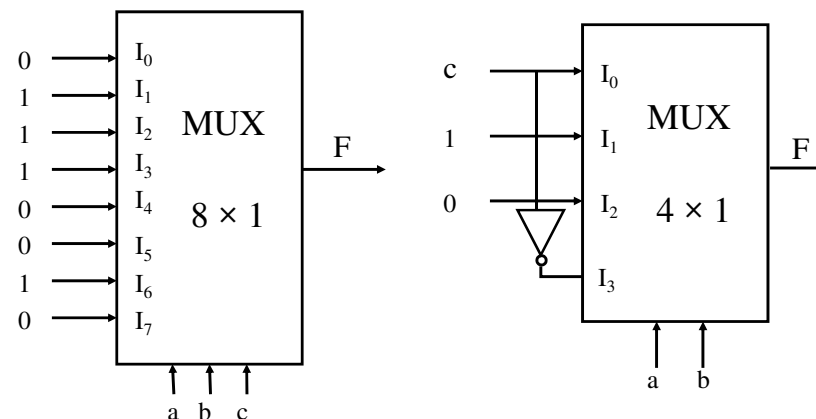
a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



## پیاده سازی توابع دودویی با استفاده از مالتی پلکسر

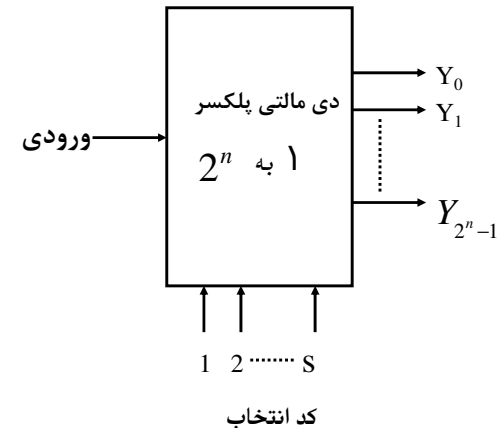
- حالت دوم: تعداد متغیرهای تابع یکی بیشتر از تعداد پایه های کنترلی مالتی پلکسر باشد.
- مثال: تابع زیر را ابتدا با مالتی پلکسر ۸ به یک و سپس با مالتی پلکسر ۴ به یک پیاده سازی نمایید.

$$F(a, b, c) = \sum m(1, 2, 3, 6)$$



## دی مالتی پلکسر (Demultiplexer) - پخش کننده داده ورودی

- یک مدار منطقی ترکیبی که یک خط ورودی را به یکی از  $2^n$  خط خروجی وصل می کند.
- انتخاب خط خروجی توسط  $n$  پایه کنترلی صورت می گیرد.

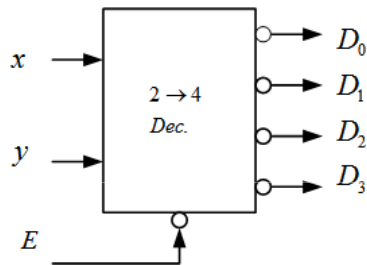


## پیاده سازی توابع دودویی با استفاده از مالتی پلکسر

- حالت سوم: اختلاف تعداد متغیرهای تابع و تعداد پایه های کنترلی مالتی پلکسر برابر ۲ باشد.
- مثال: تابع زیر را ابتدا با مالتی پلکسر ۸ به یک و سپس با مالتی پلکسر ۴ به یک پیاده سازی نمایید.

$$f(a,b,c,d) = \sum m(1,2,8,9,12,13,14,15)$$

- یک دیکودر دارای خروجیهای صفر فعال و پایه فعال ساز صفر فعال می تواند مانند یک Demultiplexer عمل کند.

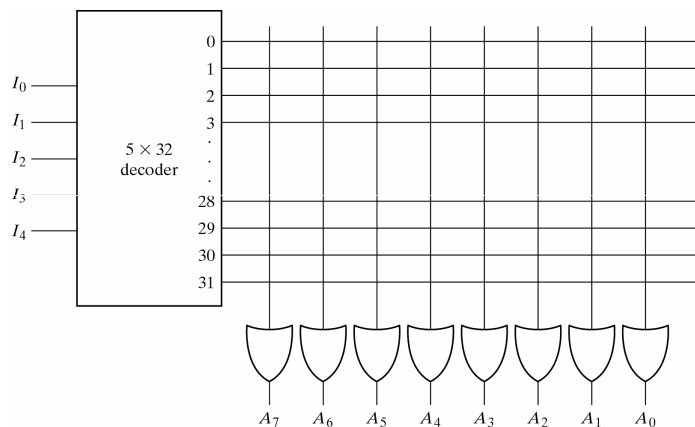


ورودی ها			خروجی ها			
E	x	y	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

## ROM (Read Only Memory) حافظه فقط خواندنی

- مداری ترکیبی است با  $n$  ورودی و  $m$  خروجی
- از دو طبقه تشکیل شده است:
- طبقه اول: AND هایی با خروجی ثابت که خروجی هر AND برابر یک مین ترم است.
- طبقه دوم: OR هایی که قابلیت برنامه ریزی دارند یعنی میتوان ورودیهای آنها را به دلخواه انتخاب نمود.

## ساختار داخلی یک ROM ۸\*۳۲



- مثال: یک ROM طراحی نمایید که دارای ۳ خط ورودی و ۲ خط خروجی باشد و توابع خروجی به صورت زیر باشند:

$$f_1(a,b,c) = \sum m(0,2,7)$$

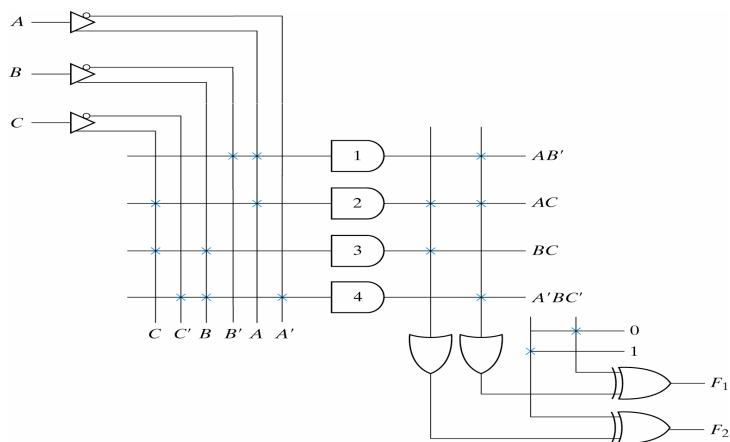
$$f_0(a,b,c) = \sum m(1,2,4,5,7)$$

آدرس خانه های حافظه	محتوای خانه های حافظه
000	10
001	01
010	11
011	00
100	01
101	01
110	00
111	11

## Programmable Logic Array (PLA)

### آرایه منطقی برنامه پذیر

- ساختار آن مشابه ROM است با این تفاوت که طبقه اول آن (AND ها) نیز قابلیت برنامه پذیری دارند.
- مثال: ساختار یک PLA با ۳ ورودی و ۲ خروجی



## (PAL)

- ساختار آن مشابه ROM و PLA است با این تفاوت که طبقه اول آن (AND ها) قابلیت برنامه پذیری دارند و طبقه دوم آن (OR ها) ثابت هستند.

- مقایسه ROM و PLA و PAL

	طبقه اول (AND ها)	طبقه دوم (OR ها)
ROM	ثابت	قابل برنامه ریزی
PLA	قابل برنامه ریزی	قابل برنامه ریزی
PAL	قابل برنامه ریزی	ثابت