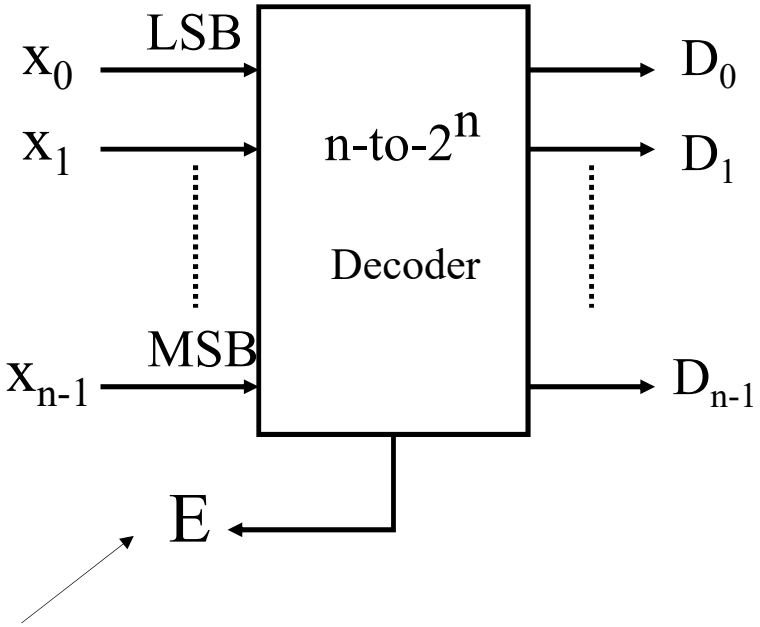
# دیکدر

دیکدر n به $2^n$ یک شبکه منطقی ترکیبی است با n خط ورودی و $2^n$ سیگنال خروجی.

عنصری است که مینترم ها را می سازد.

## ماجول دیکدر n به $2^n$

در حالت کلی برای nورودی، $2^n$ ترکیب مختلف وجود دارد، که به ازای هر ترکیب تنها یکی از خروجی ها "1" و بقیه "0" می باشند.

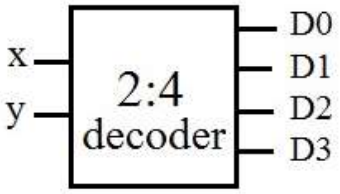



معمولا Active Low هستند.

# دیکدر (Decoder)

**ترکیب ورودی ها مشخص می کند کدام خروجی "1" شود، به عنوان مثال در حالت xy = 01 که معادل عدد ۱ می باشد، خروجی D1، "1" می شود و بقیه ی خروجی ها "0" می باشند.**

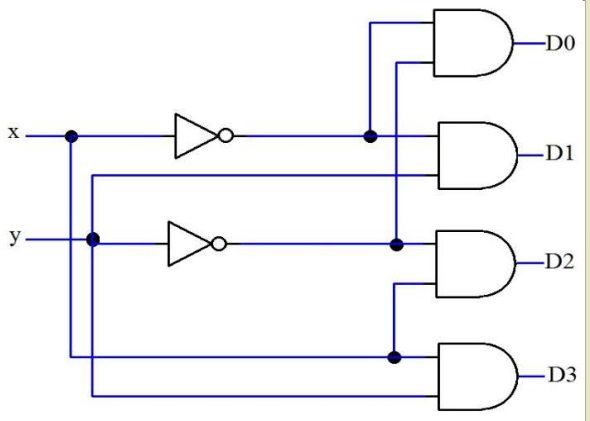
بلوک دیاگرام دیکدر 2:4




روابط منطقی خروجی ها بر حسب ورودی ها

| Inputs | | Outputs | | | |
|---|---|---|---|---|---|
| x | y | D0 | D1 | D2 | D3 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

$D0 = \bar{x}\bar{y} = m0$
$D1 = \bar{x}y = m1$
$D2 = x\bar{y} = m2$
$D3 = xy = m3$



# دیکدر (Decoder)

## دیکدر 3:8

| Inputs | | | Outputs | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| x | y | z | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
| 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 |

روابط منطقی خروجی ها بر حسب ورودی ها

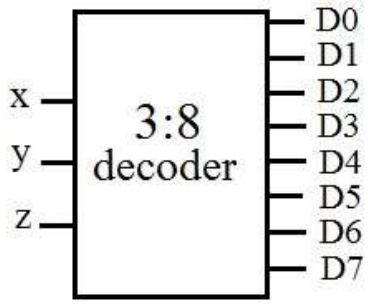
$D0 = \bar{x}\bar{y}\bar{z} = m0$ $D1 = \bar{x}\bar{y}z = m1$

$D2 = \bar{x}y\bar{z} = m2$ $D3 = \bar{x}yz = m3$

$D4 = x\bar{y}\bar{z} = m4$ $D5 = x\bar{y}z = m5$

$D6 = xy\bar{z} = m6$ $D7 = xyz = m7$

بلوک دیاگرام دیکدر 3:8

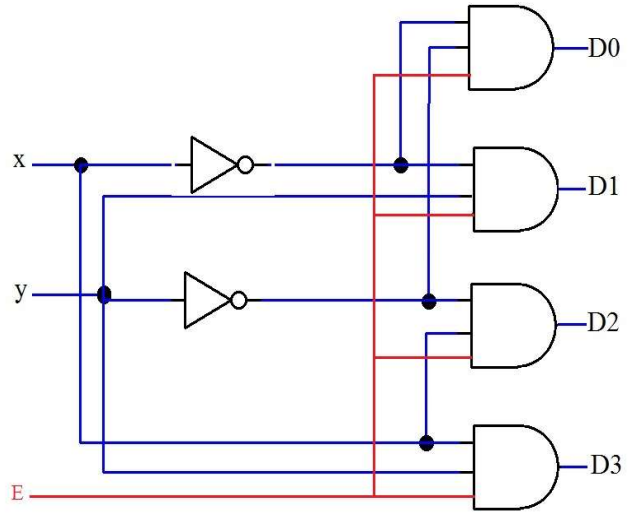


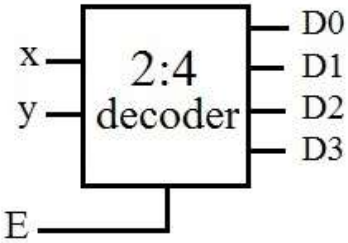
# دیکدر با پایه ی فعال ساز

دیکدر ها می توانند دارای خط فعال ساز باشند. هنگامی که این خط فعال باشد دیکدر فعال می شود و عملکرد نرمال خود را خواهد داشت، در غیر این صورت دیکدر غیر فعال است. بر این معیار دیکدر ها را می توان به دو گروه تقسیم کرد:

## ۱) دیکدر فعال بالا ( فعال "1") (Active – high)

طبق مدار روبرو هنگامی که $E = 0$ باشد تمامی خروجی ها "0" می شوند و دیکدر غیر فعال است، ولی وقتی $E=1$ شود دیکدر فعال می شود و ترکیب ورودی های x و y مشخص می کند کدام خروجی "1" شود.



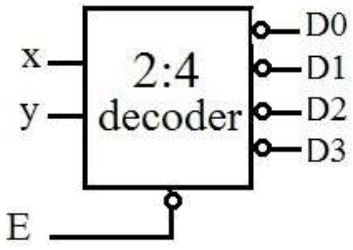

| E | x | y | D0 | D1 | D2 | D3 |
|---|---|---|---|---|---|---|
| 0 | X | X | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |

فعال

# دیکدر با پایه ی فعال ساز

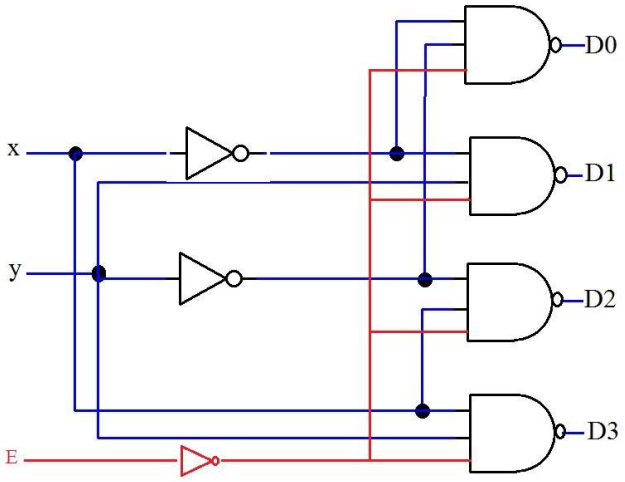
## ۲) دیکدر فعال پایین ( فعال "0") (Active – low)

دیکدر ها را می توان با گیت های NAND نیز پیاده سازی کرد. طبق مدار روبرو هنگامی که E = 1 باشد تمامی خروجی ها "1" می شوند و دیکدر غیر فعال است، ولی وقتی E=0 شود دیکدر فعال می شود و ترکیب ورودی های x و y مشخص می کند کدام خروجی "0" شود.




| E | x | y | D0 | D1 | D2 | D3 |
|---|---|---|---|---|---|---|
| 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 |

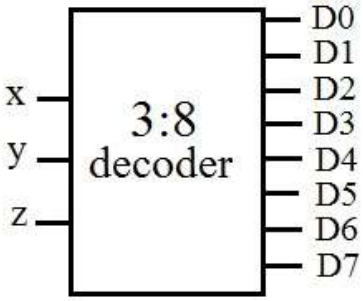
فعال



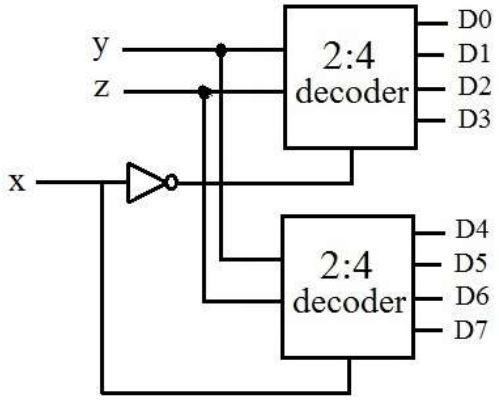
# طراحی دیکدر های بزرگ با دیکدر های کوچک تر

## طراحی دیکدر ۳:۸ به وسیله ی دیکدر های ۲:۴

با توجه به جدول درستی دیکدر ۳:۸ در ۴ حالت اول x=0 و ترکیب yz خروجی های $D_0D_1D_2D_3$ را فعال می کند، در این حالت $D_4D_5D_6D_7 = 0000$ می باشد. بنابراین x=0 باید یک دیکدر ۲:۴ را در حالت نرمال قرار دهد و دیگر دیکدر ۲:۴ را غیر فعال کند. استدلالی مشابه برای ۴ حالت بعدی برقرار است.




| Inputs | | | Outputs | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| x | y | z | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
| 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 |



# پیاده سازی توابع منطقی با دیکدر ها

**مثال:** **تابع رو به رو را با دیکدر وگیت های لازم پیاده سازی کنید.**

$$F(A , B ,C) = \sum m(0 ,1 ,4 ,6 ,7) = \prod M (2 ,3 ,5)$$

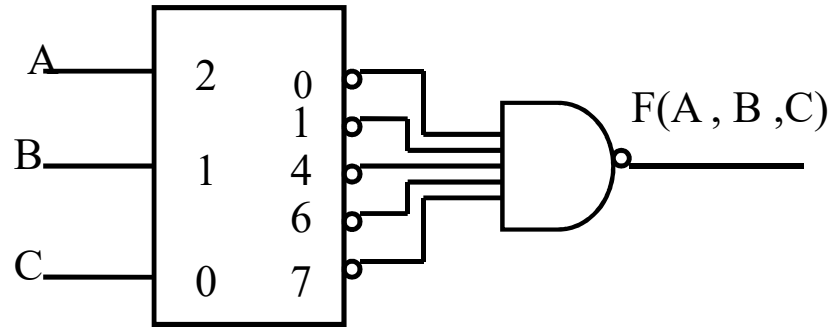
نکته:

تابع را به چندین طریق مي توانیم پیاده نماییم:

1. یک دیکدر (با خروجي فعال بالا) ویک گیت **OR** بکار بریم.
2. یک دیکدر (با خروجي فعال پایین) ویک گیت **NAND** بکار بریم.
3. یک دیکدر (با خروجي فعال بالا) ویک گیت **NOR** بکار بریم.
4. یک دیکدر (با خروجي فعال پایین) ویک گیت **AND** بکار بریم.

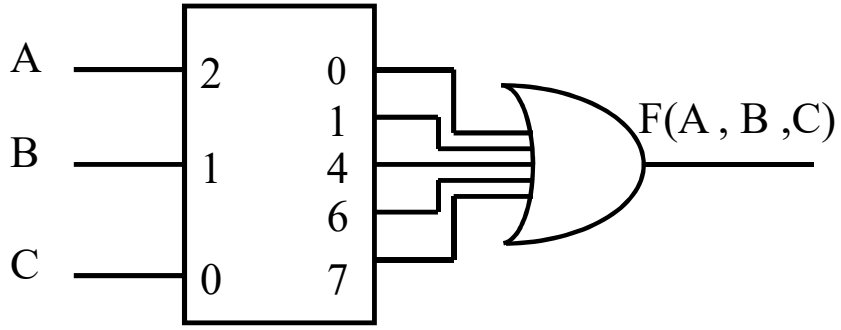
یک دیکدر (با خروجی فعال پایین) و یک گیت **NAND** بکار بریم.

$$F(A,B,C) = \overline{\bar{m}_0 . \bar{m}_1 . \bar{m}_4 . \bar{m}_6 . \bar{m}_7}$$



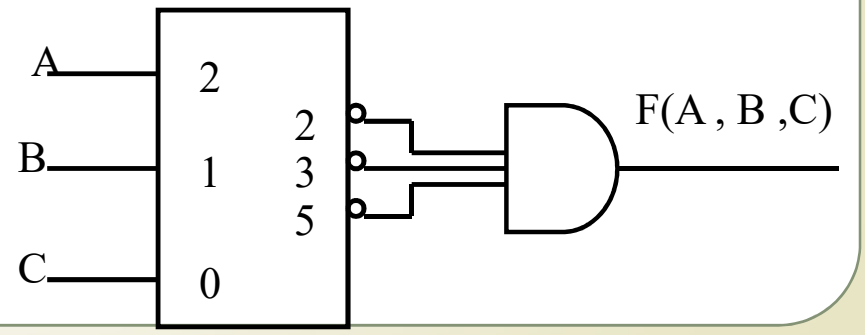

یک دیکدر (با خروجی فعال بالا) و یک گیت **OR** بکار بریم.

$$F(A,B,C) = m_0 + m_1 + m_4 + m_6 + m_7$$



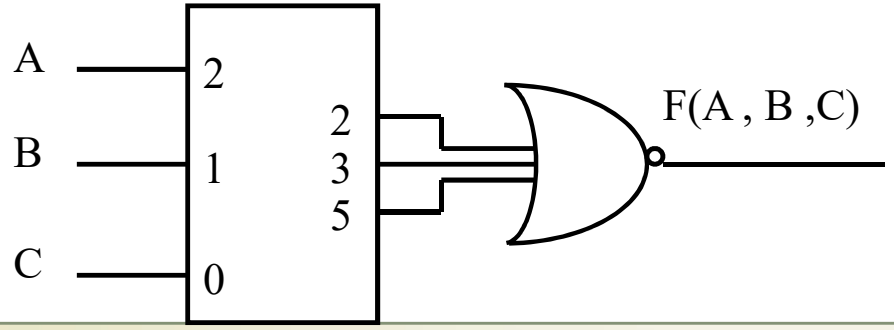

یک دیکدر (با خروجی فعال پایین) و یک گیت **AND** بکار بریم.

$$F(A,B,C) = \bar{m}_2 . \bar{m}_3 . \bar{m}_5$$




یک دیکدر (با خروجی فعال بالا) و یک گیت **NOR** بکار بریم.

$$F(A,B,C) = \overline{m_2 + m_3 + m_5}$$

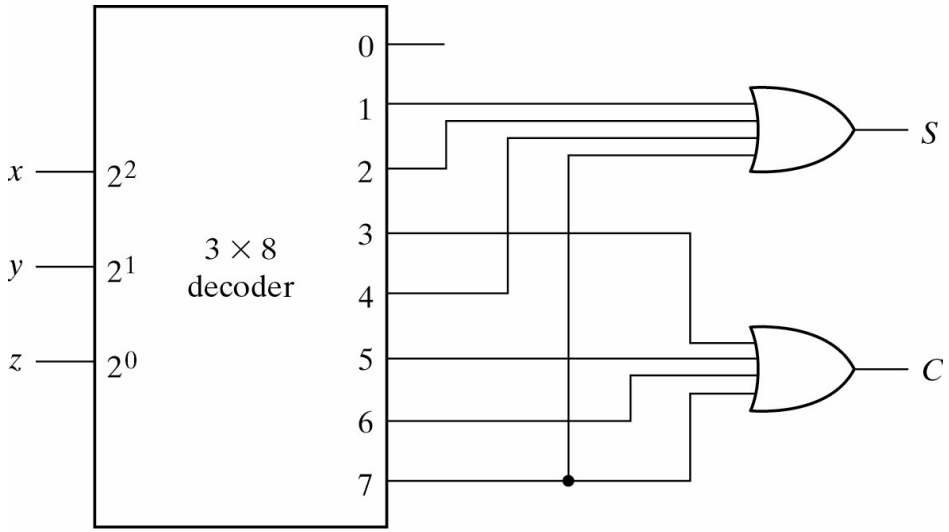


## ساختن Full Adder به وسیله دیکدر:

Truth Table

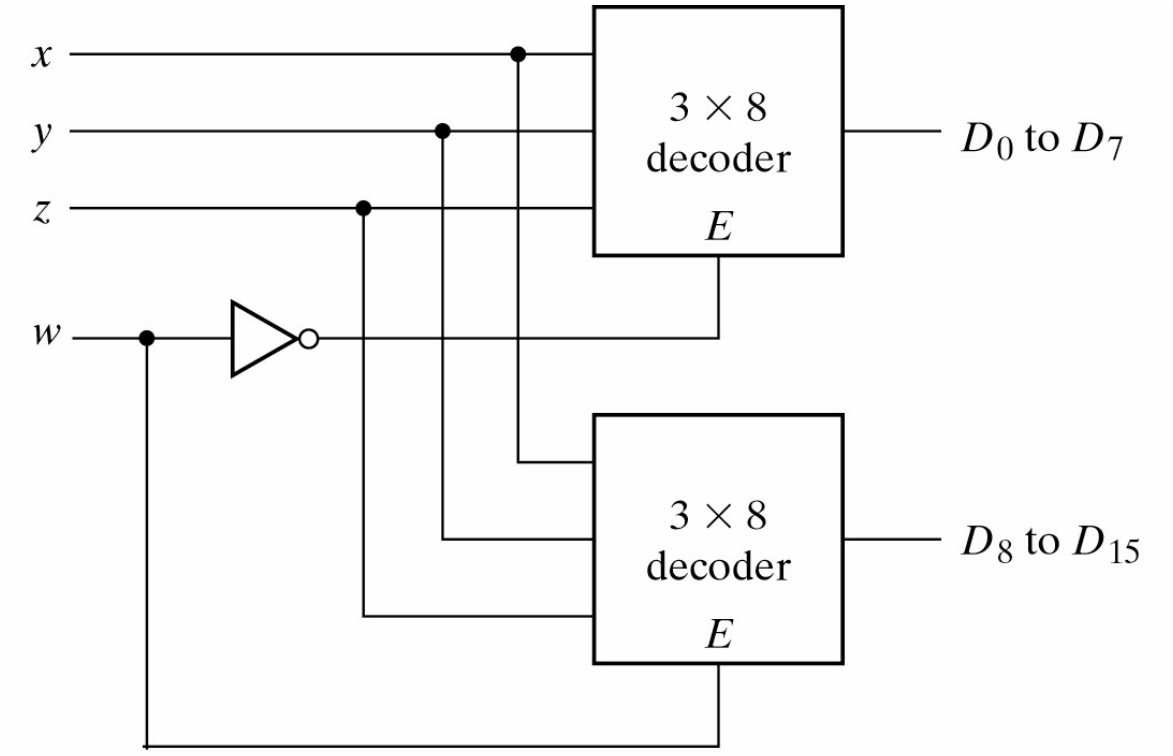
| $X_i$ | $Y_i$ | $C_{i-1}$ | $C_i$ | $S_i$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

$$\begin{cases} S_i = X_i \oplus Y_i \oplus C_{i-1} \\ C_i = X_i Y_i + X_i C_{i-1} + Y_i C_{i-1} \end{cases}$$



# ساختن دیکدر بزرگتر:

یک دیکدر 4*16 را با دو دیکدر 3*8 شبیه سازی کنید

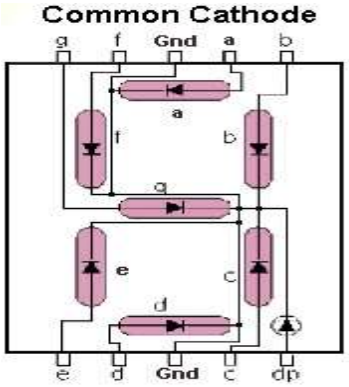


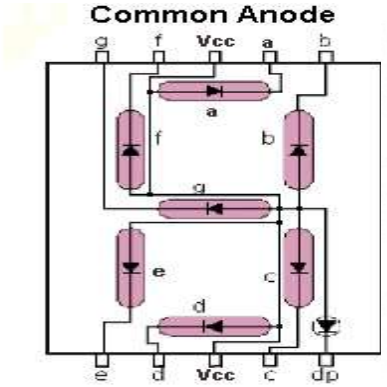
# دیکدر BCD به نمایشگر هفت قسمتی ( Seven Segment Display)

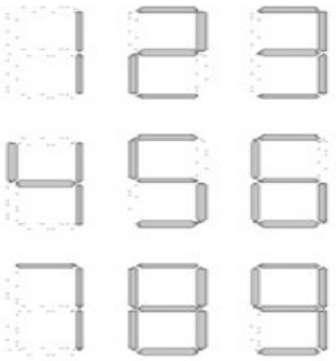
همان طور که از نام آن مشخص است از ۷ قسمت برای نمایش اعداد از ۰ تا ۹ تشکیل شده است. این هفت LED دو پیکربندی کلی می توانند داشته باشند.

1) آند های آن ها به هم وصل شده باشد که در این حالت ۷ سگمنت را آند مشترک (Common Anode = CA) گویند. در این حالت برای روشن شدن یک LED باید پایه ی مربوطه به "0" وصل شود.
2) کاتد های آن ها به هم وصل شده باشد که در این حالت ۷ سگمنت را کاتد مشترک (Common Cathode = CC) گویند. در این حالت برای روشن شدن یک LED باید پایه ی مربوطه به "1" وصل شود.

نحوه ی نام گذاری سگمنت ها و همچنین LED هایی که برای نمایش ارقام از ۰ تا ۹ باید روشن شوند (در حالت کاتد مشترک) در زیر نمایش داده شده است.

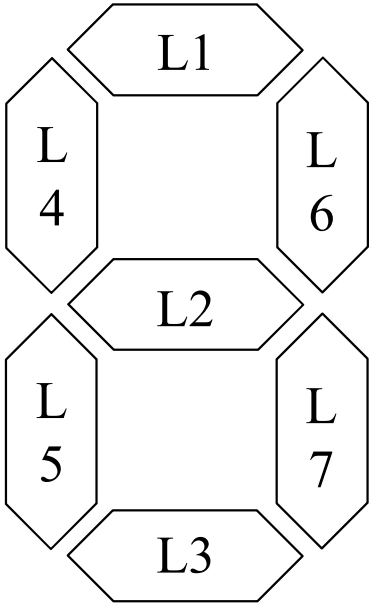


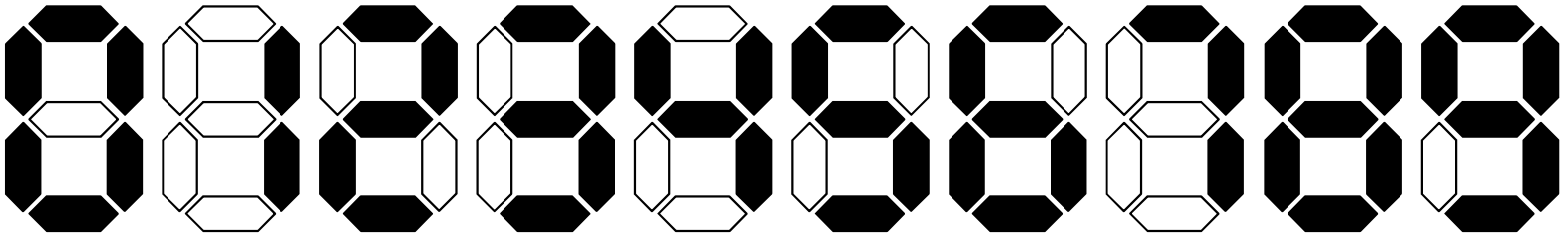
| Digit Shown | Illuminated Segment (1 = illumination) | | | | | | |
|---|---|---|---|---|---|---|---|
| | a | b | c | d | e | f | g |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 3 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 4 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 5 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 6 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 7 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

# دیکودر BCD به نمایشگر هفت قسمتی ( Seven Segment Display)



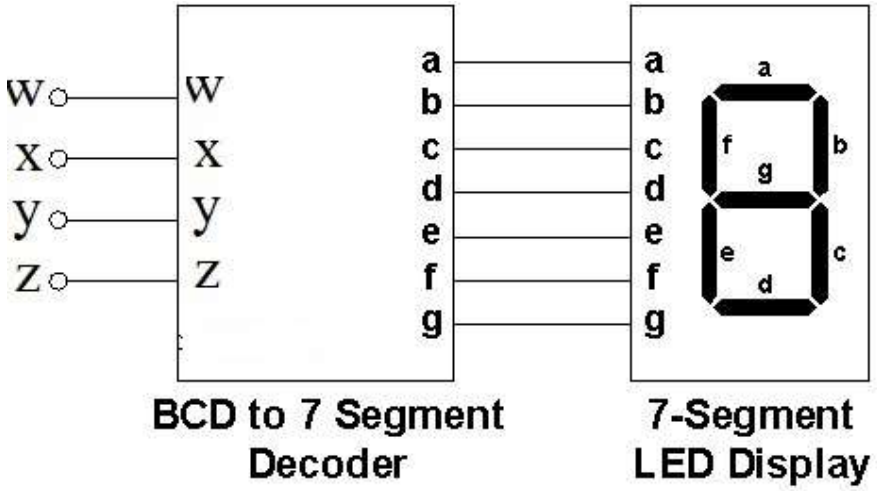

| B3 | B2 | B1 | B0 | Val | L1 | L2 | L3 | L4 | L5 | L6 | L7 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 3 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 4 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 5 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 6 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 7 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 9 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

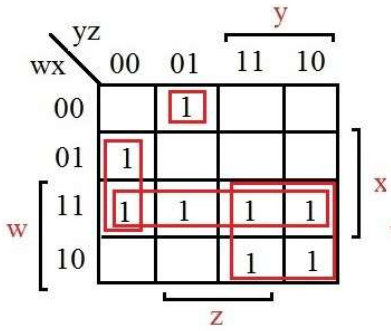
فرمت کاتد مشترک:
برای روشن شدن دیود باید به آن یک بدهیم



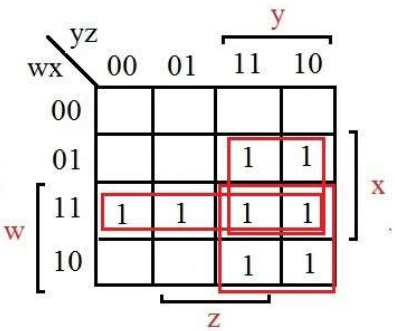
# طراحی مبدل BCD به نمایشگر 7 قطعه ای (آند مشترک)

| N | Inputs | | | | Outputs | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | w | x | y | z | a | b | c | d | e | f | g |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 6 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | … | … | … | … | … | … | … | … | … | … | … |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

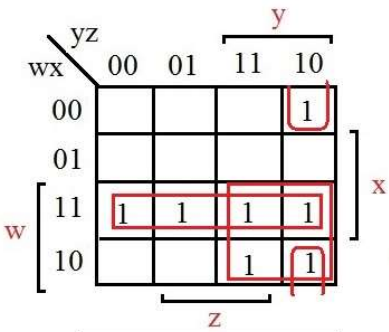




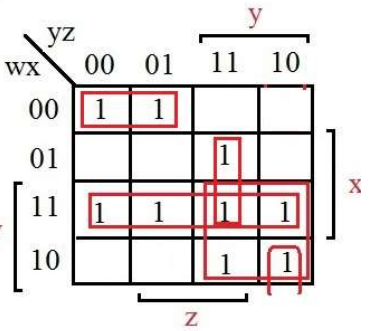

a = wx + wy + $x\bar{y}\bar{z}$ + $\bar{w}\bar{x}\bar{y}z$



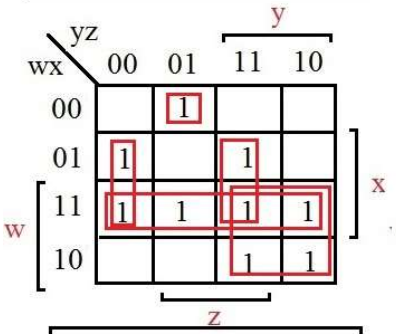

b = wx + xy + yz



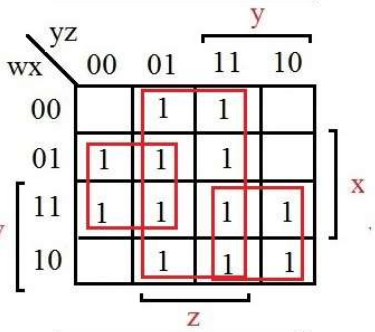

c = wx + yz + $\bar{x}\bar{y}z$



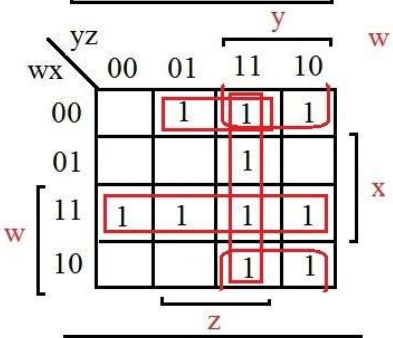

g = wx + wy + xyz + $\bar{w}\bar{x}\bar{y}$




d = wx + yz + xyz + $x\bar{y}\bar{z}$




e = z + $x\bar{y}$ + wy




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

## انکدر

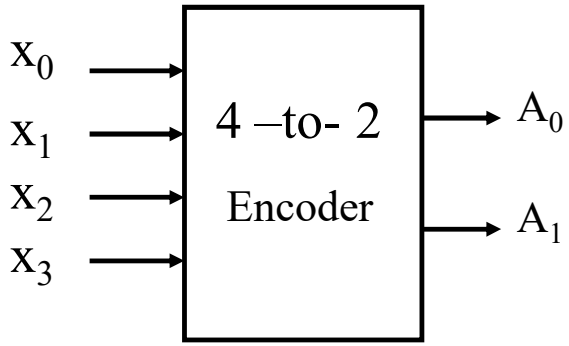
**اینکدر یک ماجول ترکیبی است که برای هر سیگنال ورودی به دستگاه یک کد خروجی منحصر به فرد را اختصاص می دهد.**

**اگر یک ماجول اینکدر n ورودی داشته باشد خروجی s باید در رابطه زیر صدق کند:**

$$\text{or} \begin{cases} 2^s \geq n \\ s \geq \text{Log}_2\, n \end{cases}$$

## مثال:

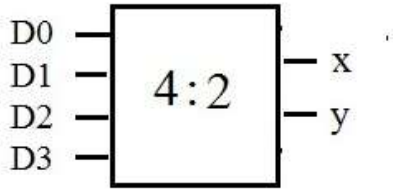
یک اینکدر براي چهار خط ورودي طراحي کنید بشرطي که در هر لحظه از زمان فقط یک ورودي فعال باشد.



# انکدر (Encoder)

## انکدر ۴:۲

در حالت کلی ۴ ورودی می توانند ۱۶ ترکیب مختلف داشته باشند. ۴ ترکیب مهم در انکدر، هنگامی است که تنها یکی از ورودی ها "1" باشد. در این حالت ها ورودی که "1" است مشخص می کند که خروجی ها چه ترکیبی داشته باشند.



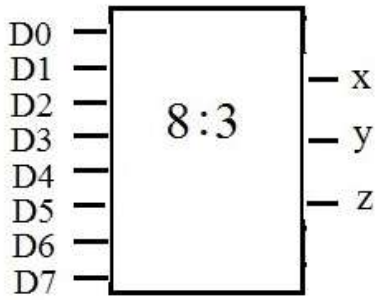

| Inputs | | | | Outputs | |
|---|---|---|---|---|---|
| D0 | D1 | D2 | D3 | x | y |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |

$$x = D2 + D3$$
$$y = D1 + D3$$

# انکدر (Encoder)

## انکدر ۸:۳

اگر دو ورودی به طور همزمان"1" شوند، خروجی حالت نامعینی دارد. به عنوان مثال اگر D2=D5 = 1 خروجی طبق روابط بالا xyz = 111 می شود که معادل حالتی است که D7=1 و بقیه ی ورودی ها "0" باشند! بنابراین در چنین حالت هایی باید ورودی ها اولویت بندی شده باشند.



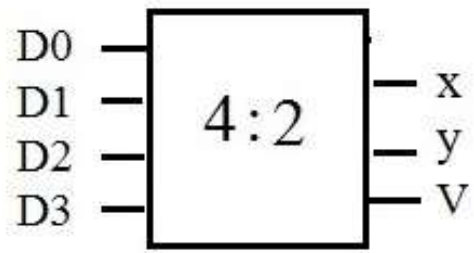

| Inputs | | | | | | | | Outputs | | |
|---|---|---|---|---|---|---|---|---|---|---|
| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | x | y | z |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

$x = D4 + D5 + D6 + D7$
$y = D2 + D3 + D6 + D7$
$z = D1 + D3 + D5 + D7$

## انکدر اولویت (Priority Encoder)

برای اینکه در هر ترکیب از ورودی ها تنها یک ورودی کدگذاری شود، در مدارات انکدر باید ورودی ها اولویت بندی شوند. در انکدر اولویت هنگامی که دو ورودی "1" شوند، ورودی که عدد بالاتری را نشان می دهد، اولویت بیشتری داشته و خروجی را تعیین می کند.

انکدر ها می توانند دارای بیت اعتبار(V) (Valid bit) در خروجی باشند که تنها در حالتی که همه ی ورودی ها صفر باشند یا به اصطلاح نامعتبر باشند، V=0 می باشد، در بقیه ی ترکیبات ورودی ها معتبر بوده و V=1 می باشد.

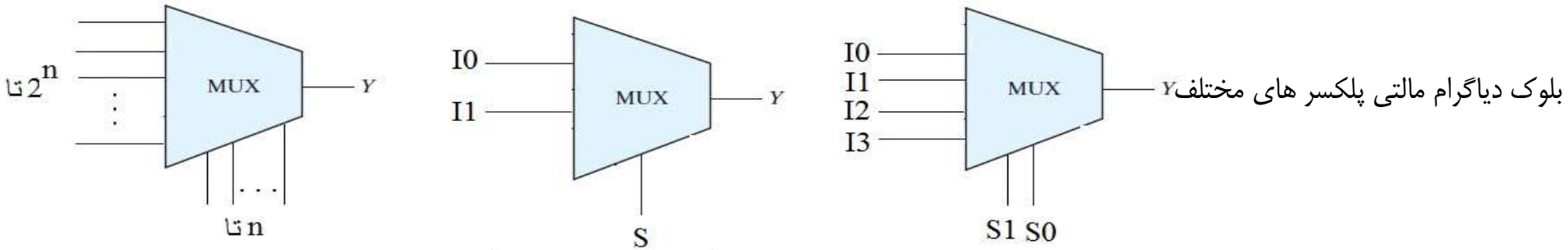



| Inputs | | | | Outputs | | |
|---|---|---|---|---|---|---|
| D0 | D1 | D2 | D3 | x | y | V |
| 0 | 0 | 0 | 0 | X | X | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| X | 1 | 0 | 0 | 0 | 1 | 1 |
| X | X | 1 | 0 | 1 | 0 | 1 |
| X | X | X | 1 | 1 | 1 | 1 |

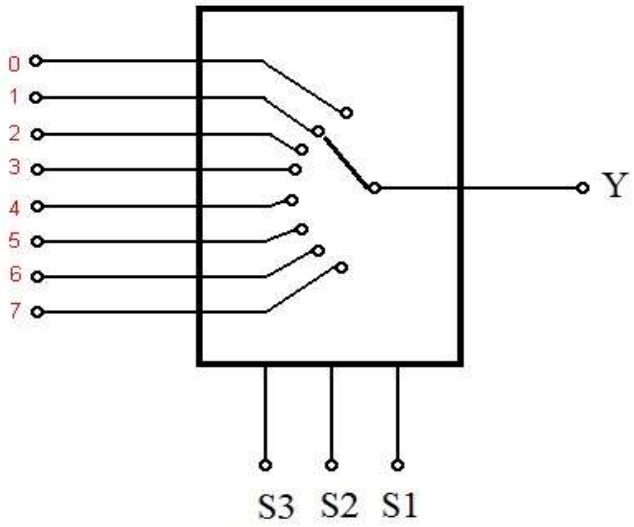
در حالتی که همه ی ورودی های انکدر "0" باشند، بیت اعتبار "0" بوده و دیگر خروجی ها حالت نامعینی دارند.

# مالتی پلکسر (Multiplexer)

مالتی پلکسر یک مدار ترکیبی است که از بین چند خط ورودی یکی را انتخاب کرده و به تک خروجی خود می رساند. مالتی پلکسر ها در حالت کلی $2^n$ ورودی، n خط انتخاب و ۱ خروجی دارند.




بلوک دیاگرام مالتی پلکسر های مختلف

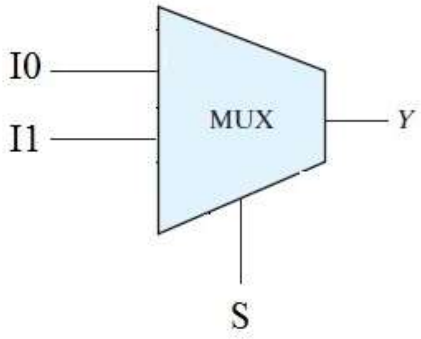



هر مالتی پلکسر را می توان به صورت یک سوئیچ چند حالته فرض کرد، که خطوط انتخاب وضعیت سوئیچ را مشخص می کنند.

# مالتی پلکسر (Multiplexer)

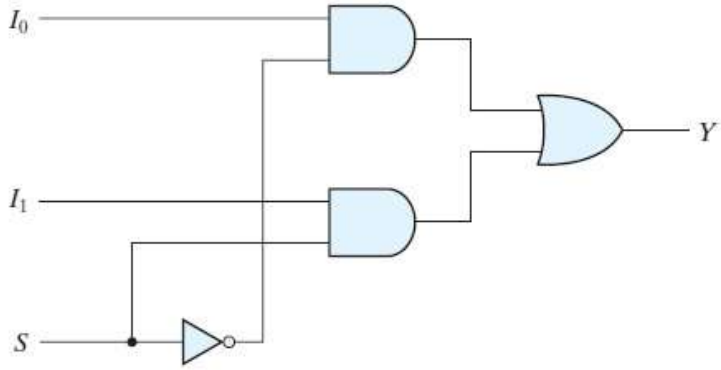
## مالتی پلکسر ۲:۱

این مالتی پلکسر یک خط انتخاب دارد که وضعیت آن مشخص می کند، کدام یک از دو ورودی به خروجی می رسد.




| S | Y |
|---|---|
| 0 | $I_0$ |
| 1 | $I_1$ |

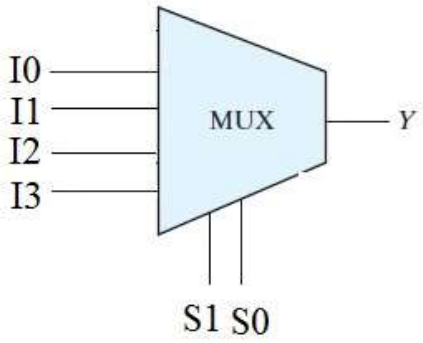
$$Y = I_0\bar{S} + I_1.S$$



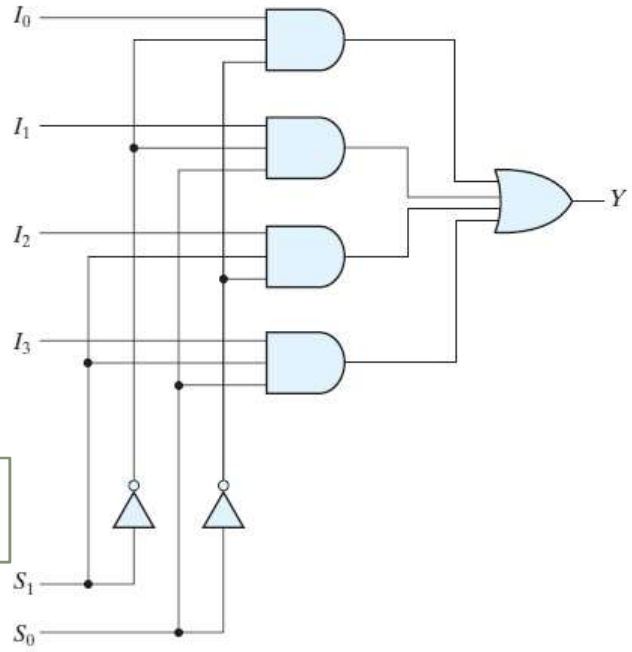
# مالتی پلکسر (Multiplexer)

## مالتی پلکسر ۴:۱

این مالتی پلکسر ۲ خط انتخاب دارد که وضعیت آن ها مشخص می کند، کدام یک از ۴ ورودی به خروجی می رسد.



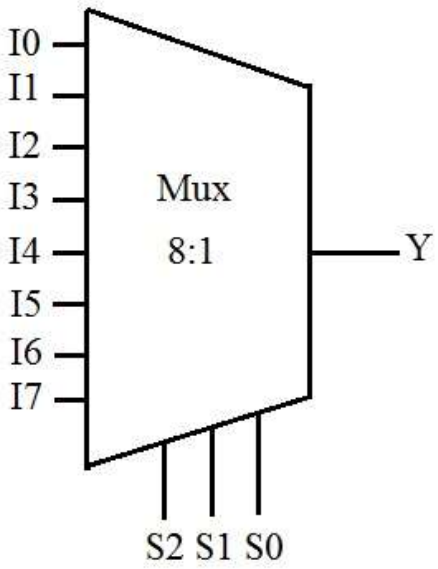

| S1 | S0 | Y |
|---|---|---|
| 0 | 0 | $I_0$ |
| 0 | 1 | $I_1$ |
| 1 | 0 | $I_2$ |
| 1 | 1 | $I_3$ |

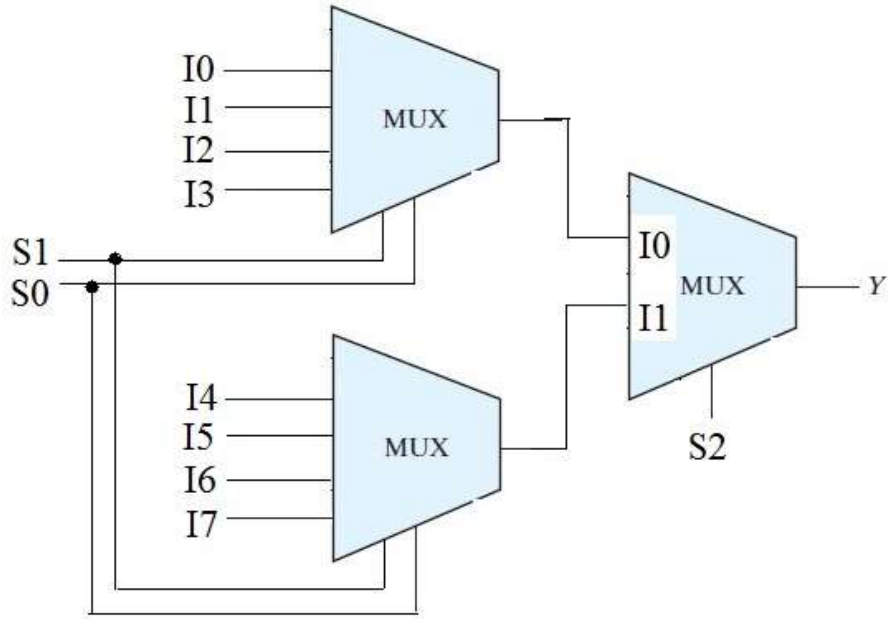



$$Y = I_0 \overline{S_1}\ \overline{S_0} + I_1 \overline{S_1}\ S_0 + I_2\ S_1 \overline{S_0} + I_3\ S_1\ S_0$$

## طراحی مالتی پلکسر های بزرگ با مالتی پلکسرهای های کوچک تر

**طراحی مالتی پلکسر 8:1 با مالتی پلکسرهای کوچکتر**

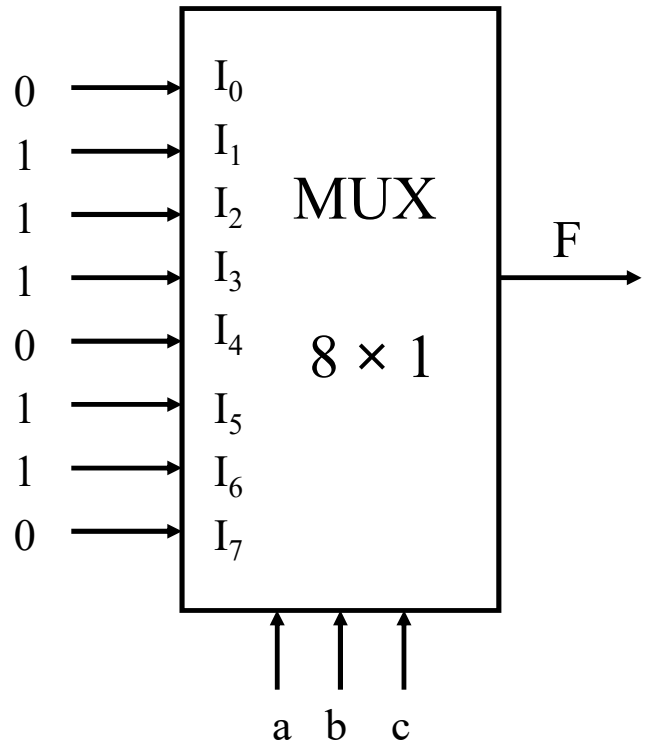


مثال : تابع $F(A,B,C,D) = \sum m(1,2,3,5,6)$ را با مالتی پلکسر $1*8$و گیت های لازم پیاده سازی کنید.

$F(A , B ,C) = \sum m(1, 2 , 3 , 5 ,6)$

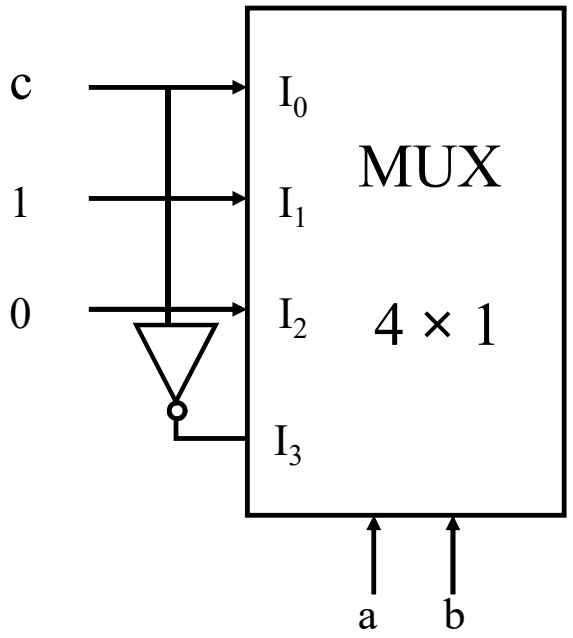
| a | b | c | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |



مثال : تابع $F(A,B,C,D) = \sum m(1,2,3,,6)$ را با مالتی پلکسر 4*1 و گیت های لازم پیاده سازی کنید.

$F(A , B ,C) = \sum M(1 ,2 , 3, 6)$

| | a | b | c | F |
|---|---|---|---|---|
| $I_0$ | 0 | 0 | 0 | 0 |
| | 0 | 0 | 1 | 1 |
| $I_1$ | 0 | 1 | 0 | 1 |
| | 0 | 1 | 1 | 1 |
| $I_2$ | 1 | 0 | 0 | 0 |
| | 1 | 0 | 1 | 0 |
| $I_3$ | 1 | 1 | 0 | 1 |
| | 1 | 1 | 1 | 0 |

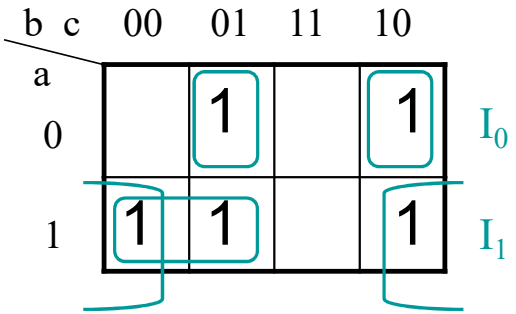


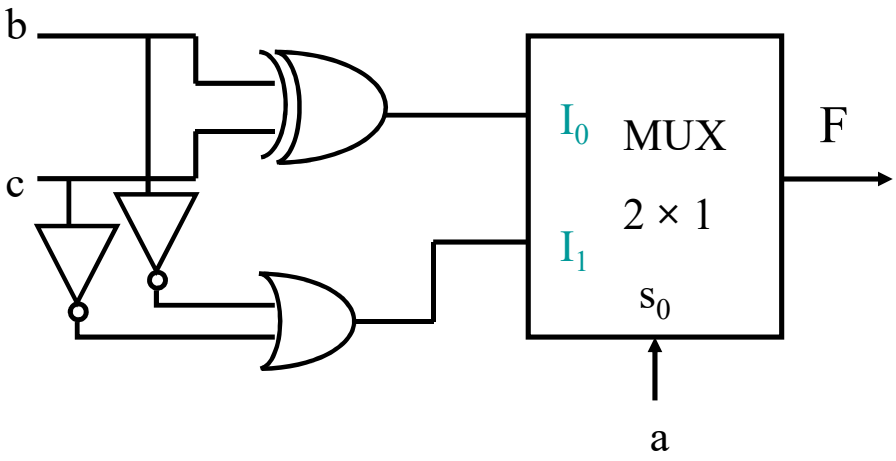
مثال : تابع $F(A,B,C,D) = \sum m(1,2,4,5,6)$ را با مالتی پلکسر 1*2 و گیت های لازم پیاده سازی کنید.

$F(A , B ,C) = \sum m(1, 2 , 4 , 5 ,6)$

| | a | b | c | F |
|---|---|---|---|---|
| $I_0$ | 0 | 0 | 0 | 0 |
| | 0 | 0 | 1 | 1 |
| | 0 | 1 | 0 | 1 |
| | 0 | 1 | 1 | 0 |
| $I_1$ | 1 | 0 | 0 | 1 |
| | 1 | 0 | 1 | 1 |
| | 1 | 1 | 0 | 1 |
| | 1 | 1 | 1 | 0 |

$$\begin{cases} I_0 = b \oplus c \\ I_1 = \bar{b} + \bar{c} = \overline{bc} \end{cases}$$



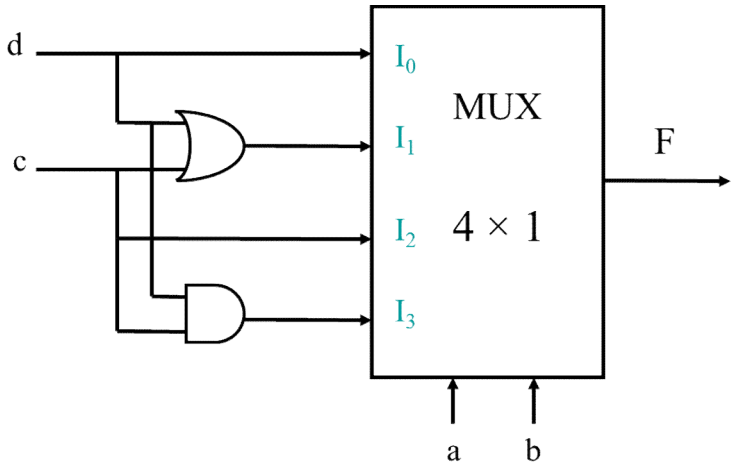



مثال : تابع $F(A,B,C,D) = \sum m(1,3,5,67,10,11,15)$ را با مالتی پلکسر 1*4 و گیت های لازم پیاده سازی کنید.

| | a | b | c | d | F |
|---|---|---|---|---|---|
| $I_0$ | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 1 | 1 |
| | 0 | 0 | 1 | 0 | 0 |
| | 0 | 0 | 1 | 1 | 1 |
| $I_1$ | 0 | 1 | 0 | 0 | 1 |
| | 0 | 1 | 0 | 1 | 1 |
| | 0 | 1 | 1 | 0 | 1 |
| | 0 | 1 | 1 | 1 | 1 |
| $I_2$ | 1 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 1 | 0 |
| | 1 | 0 | 1 | 0 | 1 |
| | 1 | 0 | 1 | 1 | 1 |
| $I_3$ | 1 | 1 | 0 | 0 | 0 |
| | 1 | 1 | 0 | 1 | 0 |
| | 1 | 1 | 1 | 0 | 0 |
| | 1 | 1 | 1 | 1 | 1 |

| a b \ c d | 00 | 01 | 11 | 10 | |
|---|---|---|---|---|---|
| 00 | | 1 | 1 | | $I_0$ |
| 01 | | 1 | 1 | 1 | $I_1$ |
| 11 | | | 1 | | $I_2$ |
| 10 | | | 1 | 1 | $I_3$ |

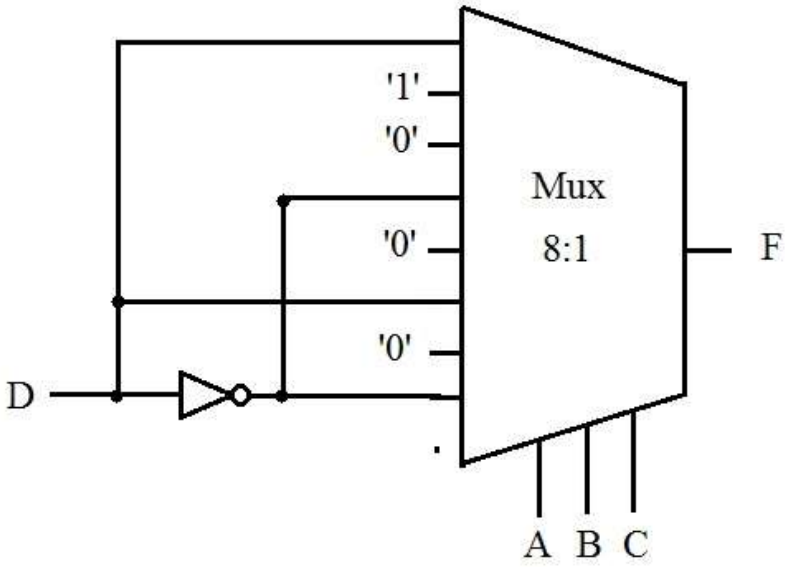
$$\begin{cases} I_0 = d \\ I_1 = d + c \\ I_2 = C \\ I_3 = cd \end{cases}$$



مثال : تابع $F(A,B,C,D) = \sum m(1,2,3,6,11,14)$ را با مالتی پلکسر و گیت های لازم پیاده سازی کنید.

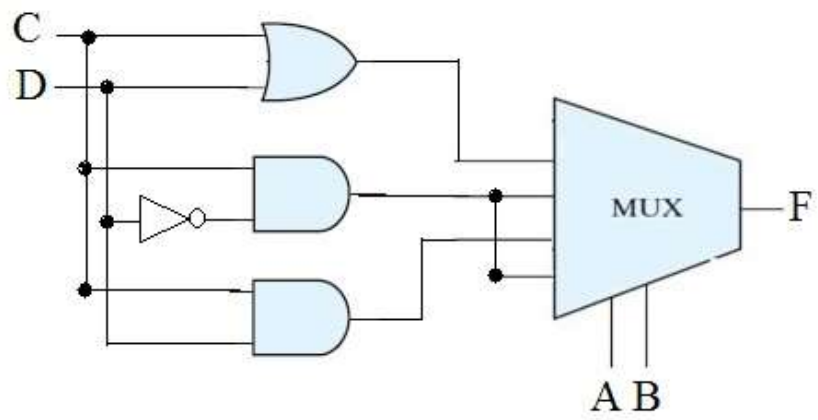
| A | B | C | D | F | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | F = D |
| 0 | 0 | 0 | 1 | 1 | |
| 0 | 0 | 1 | 0 | 1 | F = 1 |
| 0 | 0 | 1 | 1 | 1 | |
| 0 | 1 | 0 | 0 | 0 | F = 0 |
| 0 | 1 | 0 | 1 | 0 | |
| 0 | 1 | 1 | 0 | 1 | $F = \overline{D}$ |
| 0 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | F = 0 |
| 1 | 0 | 0 | 1 | 0 | |
| 1 | 0 | 1 | 0 | 0 | F = D |
| 1 | 0 | 1 | 1 | 1 | |
| 1 | 1 | 0 | 0 | 0 | F = 0 |
| 1 | 1 | 0 | 1 | 0 | |
| 1 | 1 | 1 | 0 | 1 | $F = \overline{D}$ |
| 1 | 1 | 1 | 1 | 0 | |

## روش اول)



## روش دوم)

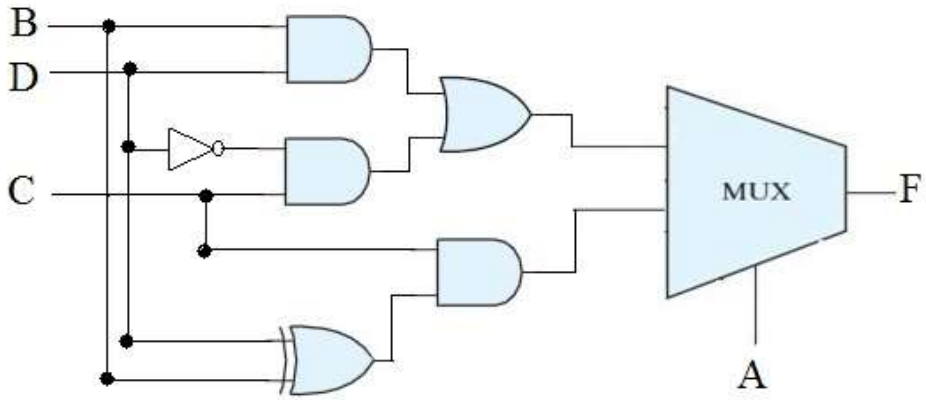
| A | B | C | D | F | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 1 | 1 | $F = C + D$ |
| 0 | 0 | 1 | 0 | 1 | |
| 0 | 0 | 1 | 1 | 1 | |
| 0 | 1 | 0 | 0 | 0 | |
| 0 | 1 | 0 | 1 | 0 | |
| 0 | 1 | 1 | 0 | 1 | $F = C\overline{D}$ |
| 0 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 0 | 1 | 0 | |
| 1 | 0 | 1 | 0 | 0 | $F = CD$ |
| 1 | 0 | 1 | 1 | 1 | |
| 1 | 1 | 0 | 0 | 0 | |
| 1 | 1 | 0 | 1 | 0 | |
| 1 | 1 | 1 | 0 | 1 | $F = C\overline{D}$ |
| 1 | 1 | 1 | 1 | 0 | |



## روش سوم)

| A | B | C | D | F |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

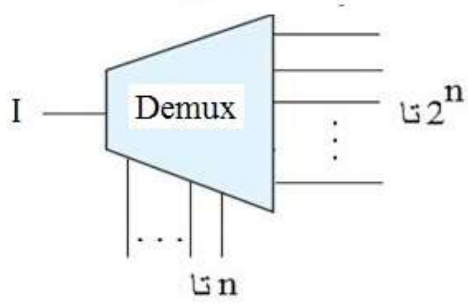
$F = \bar{B}D + C\bar{D}$

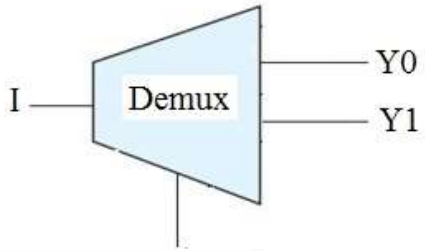


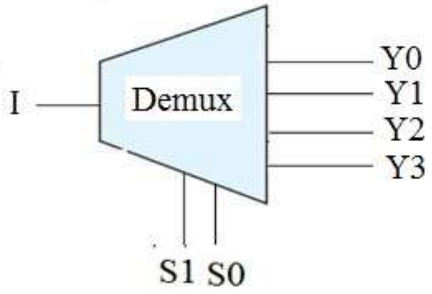

$F = C.(B \oplus D)$

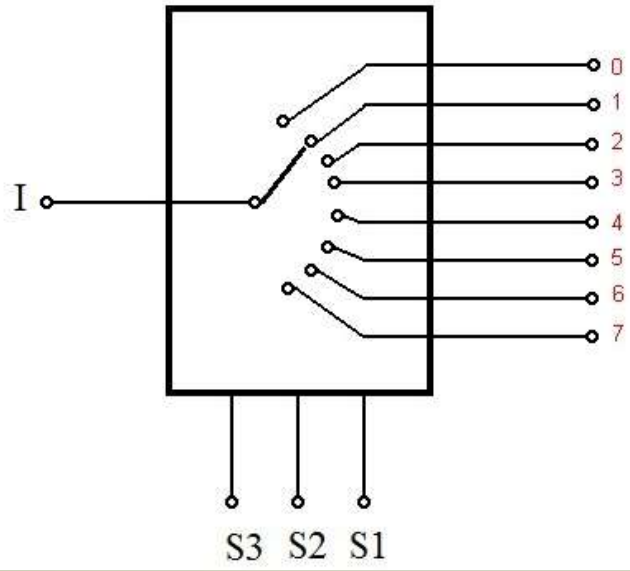
# دیمالتی پلکسر (Demultiplexer)

به صورت عکس مالتی پلکسر عمل می کند، یعنی ۱ ورودی، $2^n$ خط خروجی و n خط انتخاب دارد. وضعیت خطوط انتخاب مشخص می کند، ورودی به کدام یک از خطوط خروجی وصل شود.



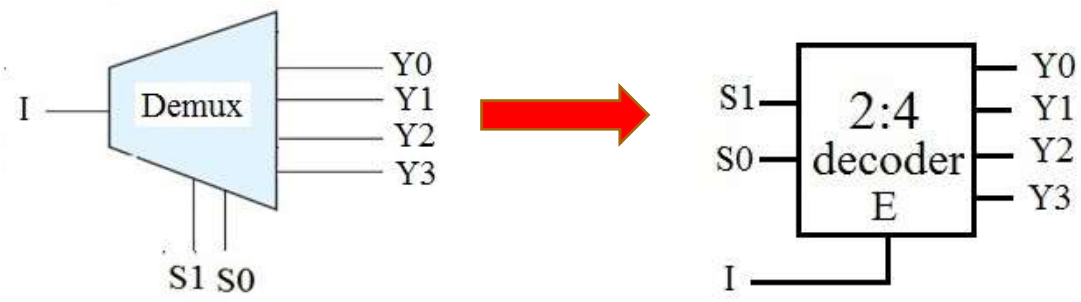




بلوک دیاگرام دمالتی پلکسر های مختلف

# دیمالتی پلکسر (Demultiplexer)

دمالتی پلکسر ها را می توان به صورت دیکدر های با خط فعال ساز در نظر گرفت، که خط فعال ساز ورودی دمالتی پلکسر، ورودی های دیکدر، خطوط انتخاب دمالتی پلکسر و خروجی های دیکدر، خروجی های دمالتی پلکسر در نظر گرفته می شوند.




| S1 | S0 | Y0 | Y1 | Y2 | Y3 |
|---|---|---|---|---|---|
| 0 | 0 | I | 0 | 0 | 0 |
| 0 | 1 | 0 | I | 0 | 0 |
| 1 | 0 | 0 | 0 | I | 0 |
| 1 | 1 | 0 | 0 | 0 | I |