**به نام خدا**

**آزمایش 2**

طراحي مدارهاي اوليه Multiplexer4to1, Encoder4to2, Decoder2to4 و Comprator 3 بيتي

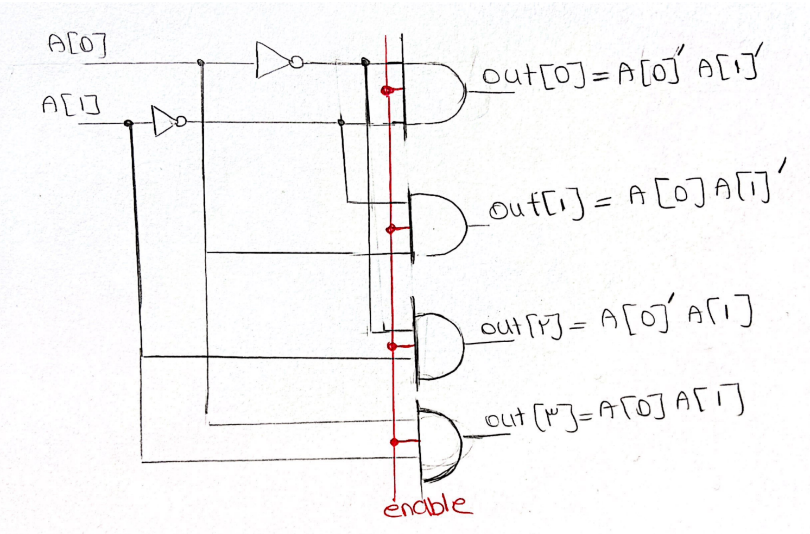
روژینا کاشفی-هلیا سادات هاشمی پور

================================================================

**Decoder(2x4)**

برای رسم دیکودر 2x4 به یک ورودی 2 بیتی و خروجی 4 بیتی نیاز داریم که به صورت vector تعریف کردیم و یک enable گذاشتیم .اگر enable فعال باشد مدار فعال و خروجی ها فعال می شوند در غیر اینصورت صرف نظر آنکه ورودی چی باشد تمام خروجی ها 0 می شود.

طراحی مدار Decoder 2x4 :

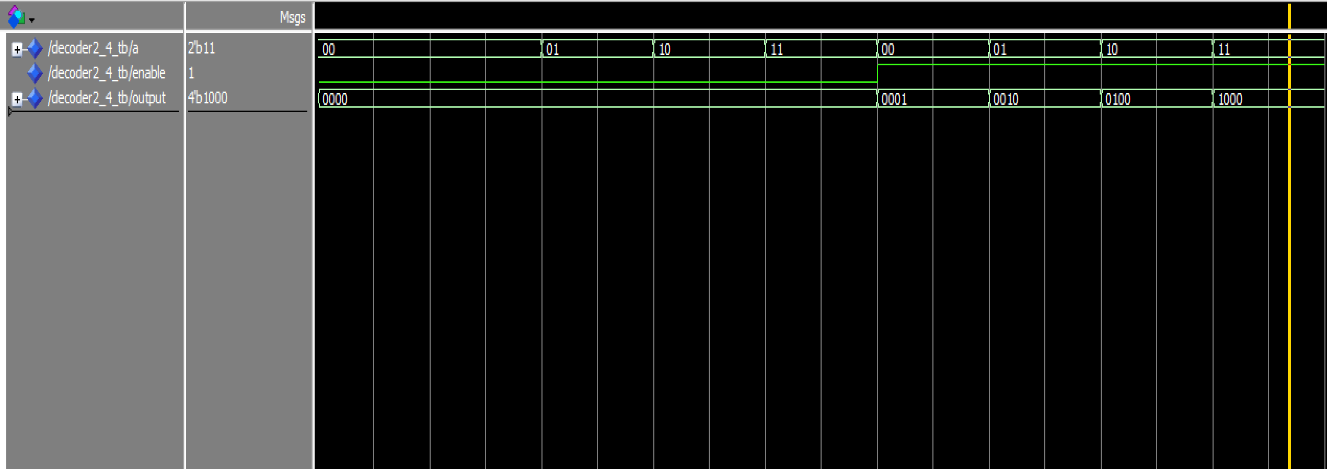


در جدول زیر نحوه کار کردن دیکودر را مشاهده می کنیم:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Output[3] | Output[2] | Output[1] | Output[0] | A[0] | A[1] | Enable | Decimal |
| 0 | 0 | 0 | 0 | X | X | 0 |  |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 2 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 3 |

خروجی کد نوشته شده:

همانطور که مشاهده می کنیم خروجی کد مانند جدول فوق است.

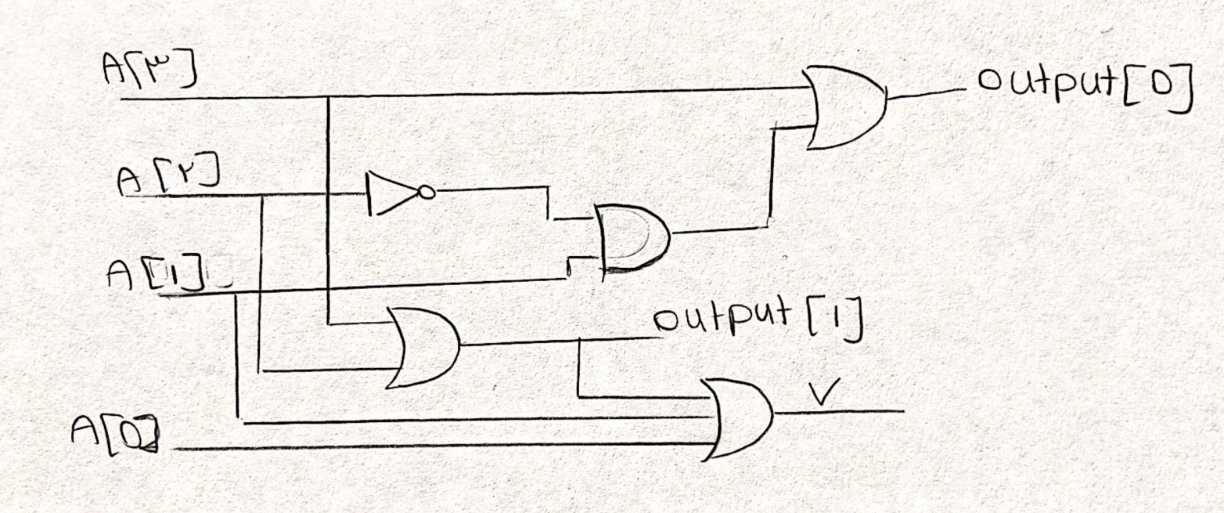


===========================================================================================

**Encoder (4x2)**

برای رسم encoder 4x2 باید یک vector 4 بیتی به عنوان ورودی دهیم و خروجی یک vector 2 بیتی است.ما در این قسمت یک priority encoder تعریف کردیم که زمانی که بیت v(valid) صفر است صرف نظر از انکه ورودی چیست خروجی مدار نامشخص است و همچنین اگر v یک باشد هر بیت صرف نظر از مقدار بیتهای قبلیش مقدارش 1 می شود و مقدار بیت های بعدیش 0 می شود.

طراحی مدار Encoder 2x4 :

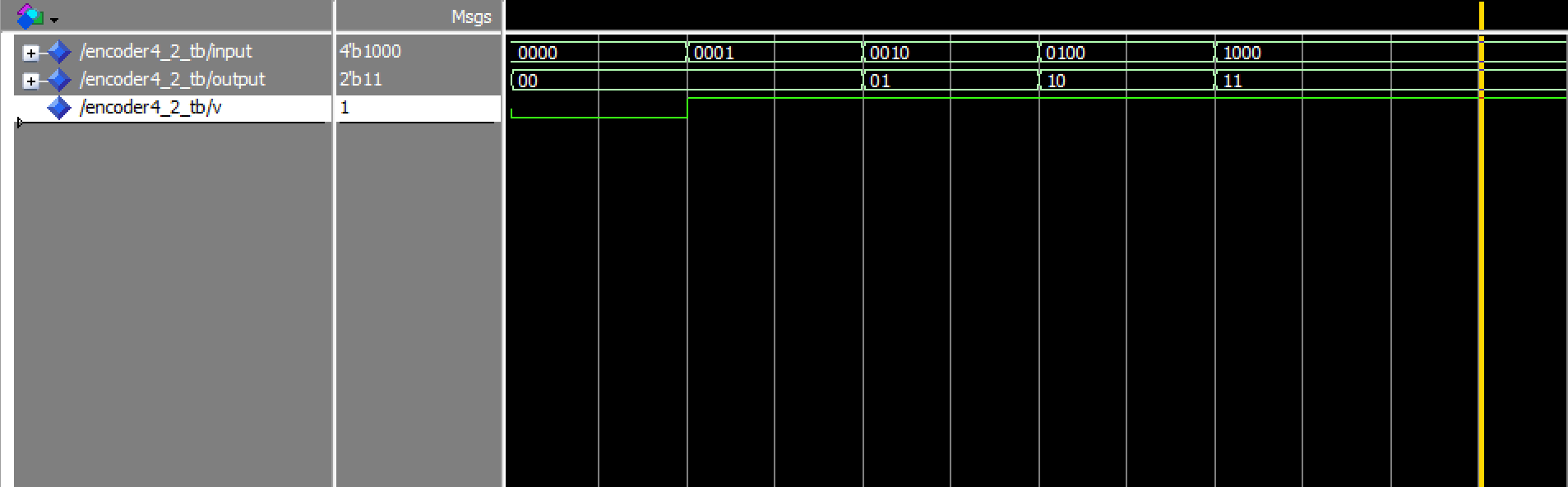


در جدول زیر نحوه کار کردن priority encoder 4x2را مشاهده می کنیم:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Output[0] | Output[1] | A[0] | A[1] | A[2] | A[3] | V | Decimal |
| X | X | 0 | 0 | 0 | 0 | 0 |  |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | X | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | X | X | 1 | 0 | 1 | 2 |
| 1 | 1 | X | X | X | 1 | 1 | 3 |

خروجی کد نوشته شده:

همانطور که مشاهده می کنیم خروجی کد مانند جدول فوق است.



===============================================================================================

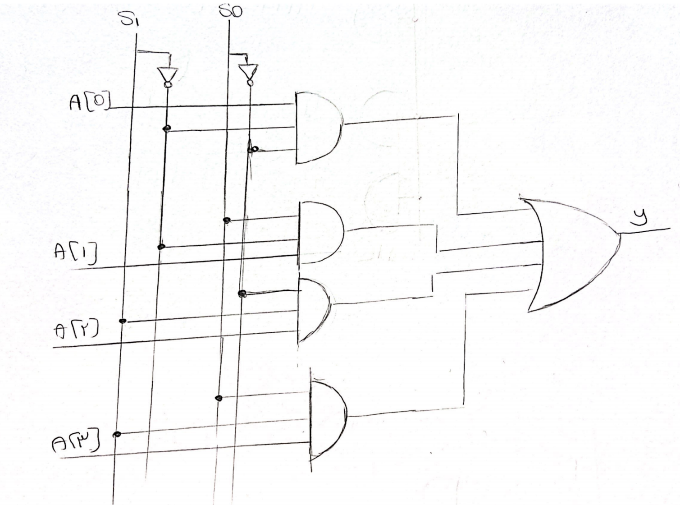
Multiplexer

هدف ما طراحی یک مالتی پلکسر 16x1 است برای اینکار ابتدا یک مالتی پلکسر 4x1 طراحی می کنیم و سپس با استفاده از 5 مالتی پلکسر 4x1 یک مالتی پلکسر 16x1 میسازیم.

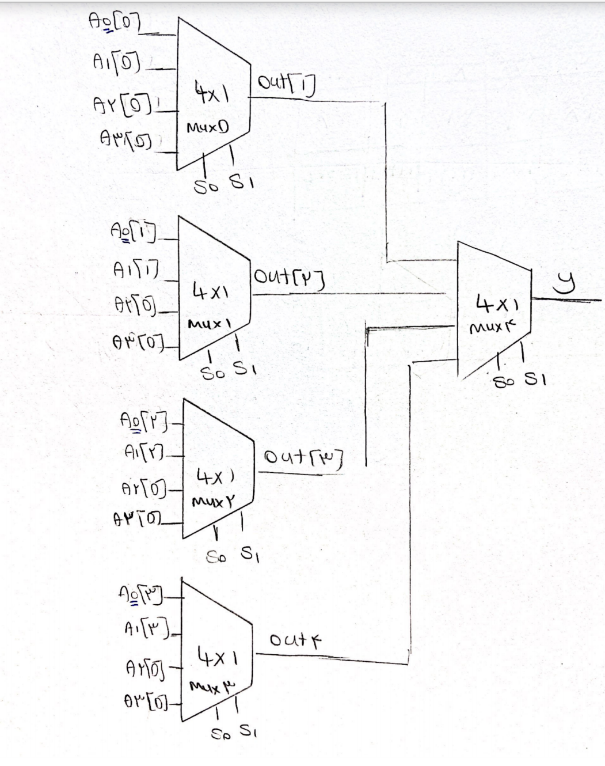
جدول یک مالتی پلکسر 4x1 :

|  |  |  |
| --- | --- | --- |
| Y | S0 | S1 |
| Input[0] | 0 | 0 |
| Input[1] | 1 | 0 |
| Input[2] | 0 | 1 |
| Input[3] | 1 | 1 |

طراحی یک مالتی پلکسر 4x1 :



طراحی یک مالتی پلکسر 16x1:



* دقت شود که خط select ها یکسان است .
* ما در طراحی خود به جای دادن یک وکتور 16 بیتی از 4 vector 4 بیتی استفاده کردیم به صورتی که a0 وکتور مربوط به بیت صفرم هر مالتی پلکسر , a1 وکتور مربوط به بیت 1 هر مالتی پلکسر , a2 وکتور مربوط به بیت 2 هر مالتی پلکسر و a3 وکتور مربوط به بیت 3 هر مالکتی پلکسراست.

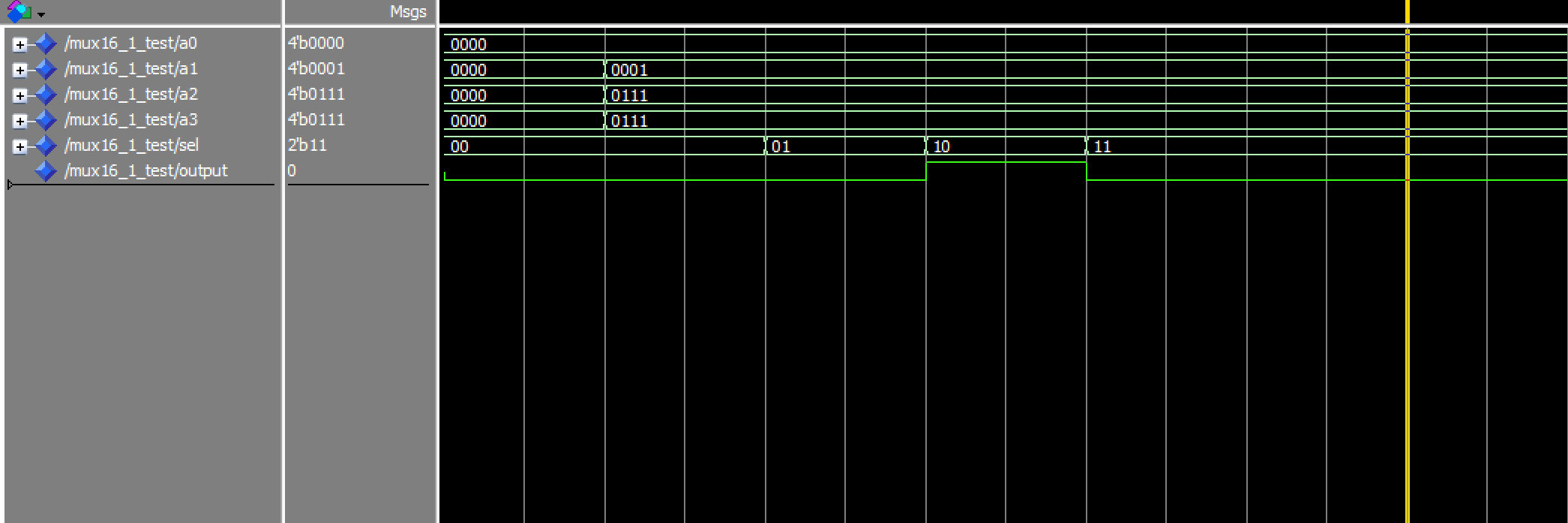
خروجی کد نوشته شده:

مشاهده می کنیم اگر خط select 0 باشد بیت 0 ام هر مالتی پلکسر به ماکس اخرمنتقل می شود و در نهایت بیت 0 a0 به عنوان خروجی نشان داده می شود.(خروجی 0)

اگر خط select 1 باشد بیت 1 ام هر مالتی پلکسر به ماکس اخرمنتقل می شود و در نهایت بیت 1 a1 به عنوان خروجی نشان داده می شود.(خروجی 0)

اگر خط select 2 باشد بیت 2 ام هر مالتی پلکسر به ماکس اخرمنتقل می شود و در نهایت بیت 2 a2 به عنوان خروجی نشان داده می شود.(خروجی 1)

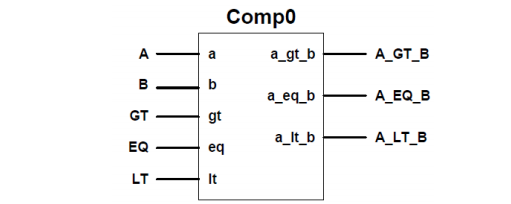
اگر خط select 3 باشد بیت 3 ام هر مالتی پلکسر به ماکس اخرمنتقل می شود و در نهایت بیت 3 a3 به عنوان خروجی نشان داده می شود.(خروجی 0)



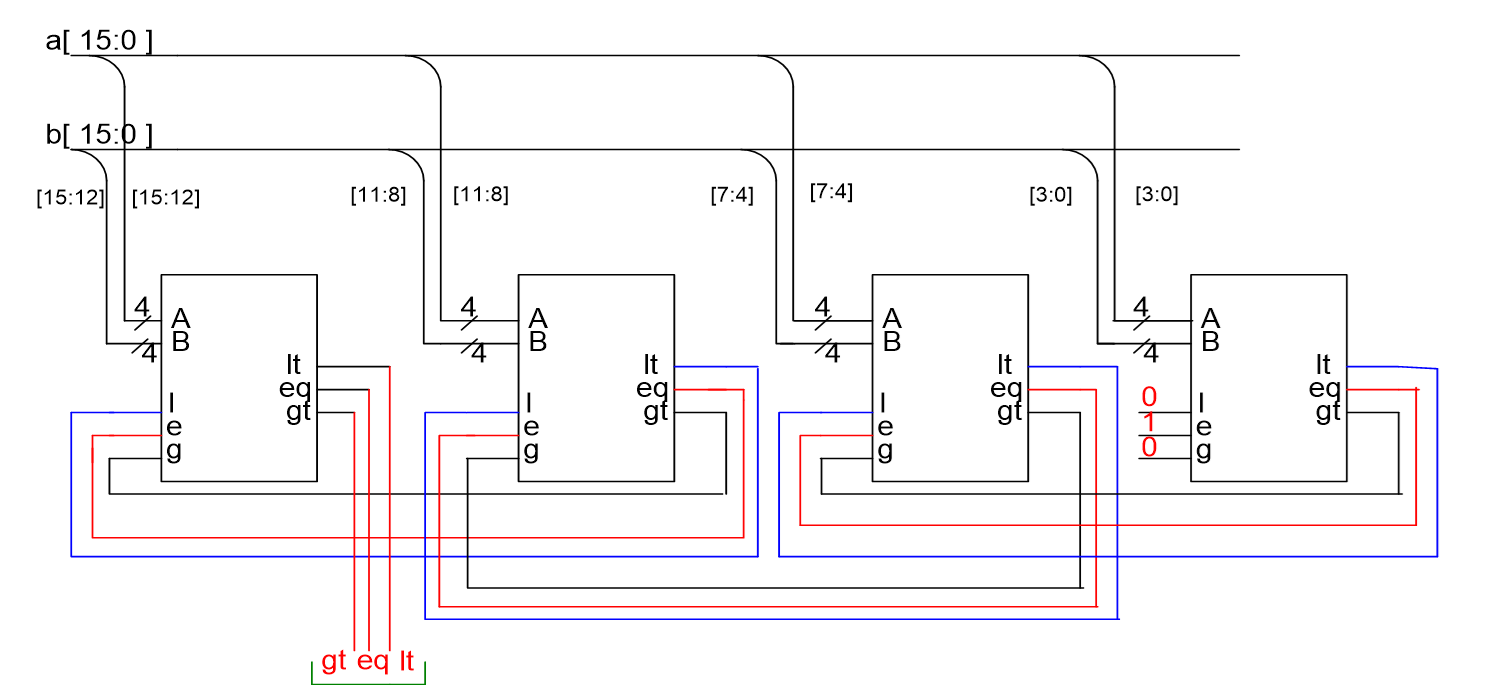
===========================================================================

**Comparator**

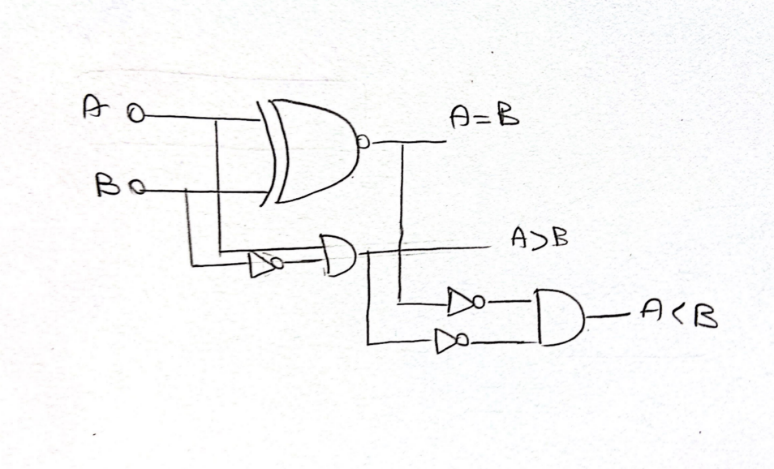
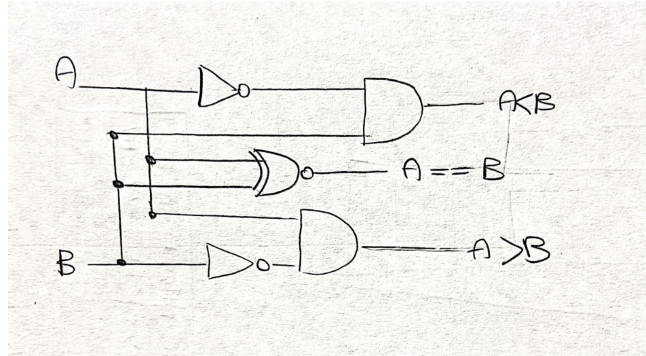
ابتدا باید یک comparator 1 بیتی طراحی کنیم که 2 ورودی دارد و نشان دهد در صورتی که اگر ورودی ها 1 بیت بود چگونه ان را مقایسه کند اما برای طراحی comparator های بیش از 1 بیت باید 5 ورودی داشته باشیم.2 تا از وزودی ها بیت های برای مقایسه کردن و 3 ورودی دیگر برای نشان دادن برابر یا بزرگتر با کوچکتر بودن بیت های قبلی است.



که مثالی از آن را میبینیم:

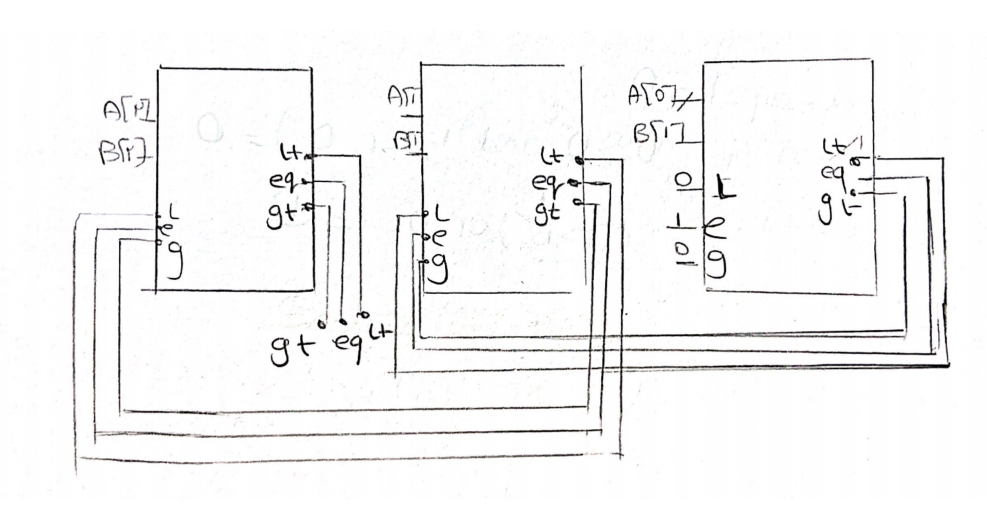


طراحی comparator 1 بیت:

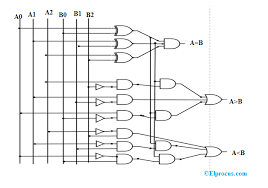


دو طراحی فوق یکسان است.

طراحی comparator 3 بیت:



اگر بدون comparator 1 bit می خواستیم انجام دهیم به صورت زیر می بود.



خروجی کد نوشته شده:

همانطور که مشاهده می کنیم زمانی که ورودیها 000 و 001 است a\_lt\_b=1

زمانی که ورودی های 010 و 001 است a\_gt\_b=1

زمانی که ورودی ها 100 و 100 است a\_eq\_b=1

