**"به نام یزدان پاک"**

**گزارش کار آزمایش دوم**

**اعضای گروه:**

**کیانا آقا کثیری 9831006**

**محمد چوپان 9831125**

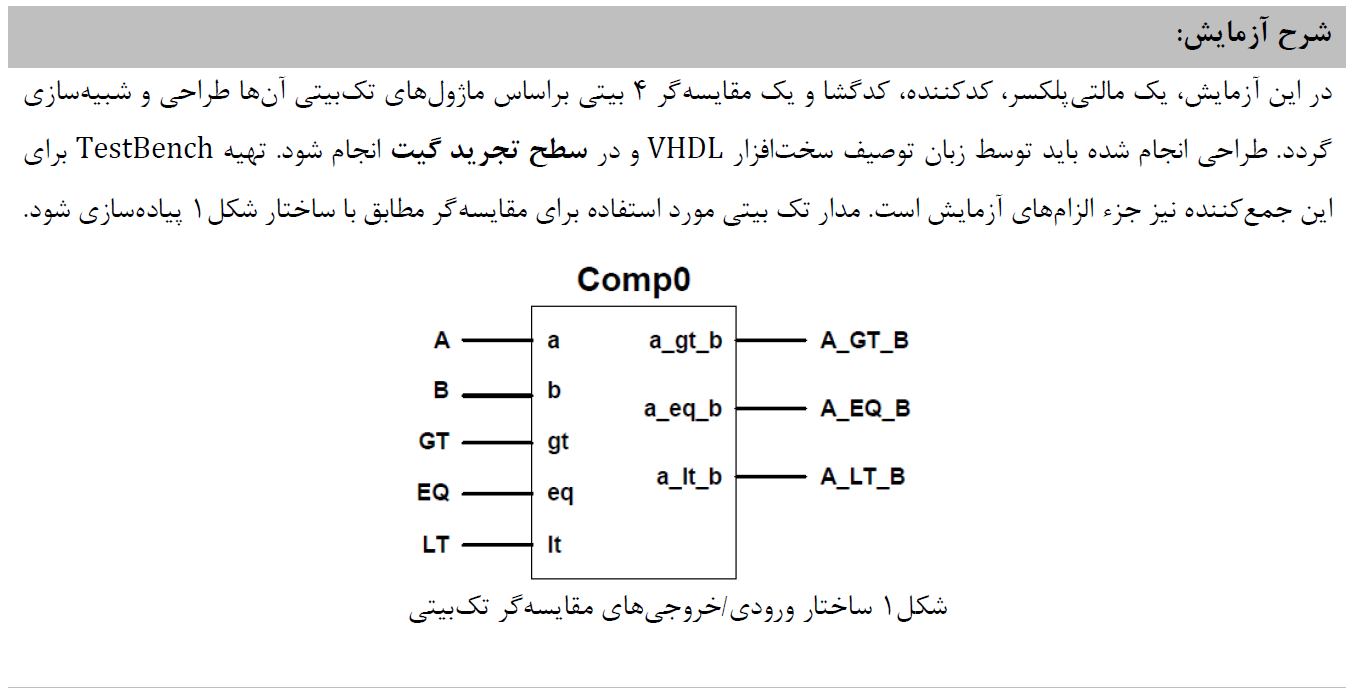
**سارا تاجرنیا 9831016**

**نویسنده گزارش : کیانا آقا کثیری**

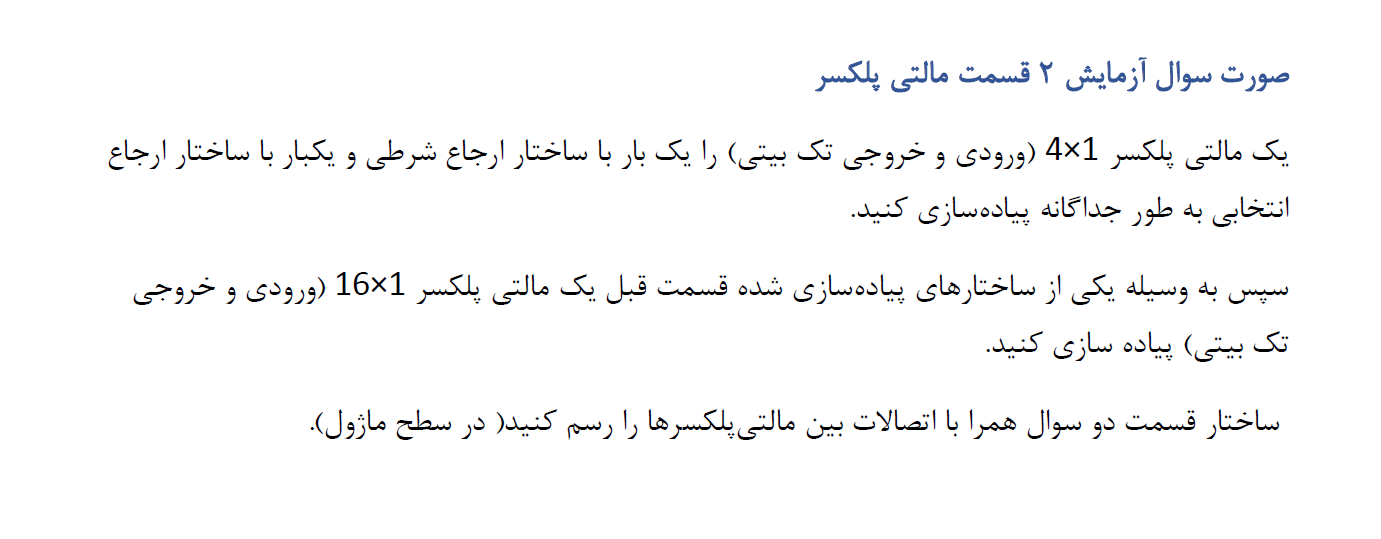
**تاریخ آزمایش : 20/12/99**

**تاریخ تحویل گزارش :62/12/99**

**شرح آزمایش:**

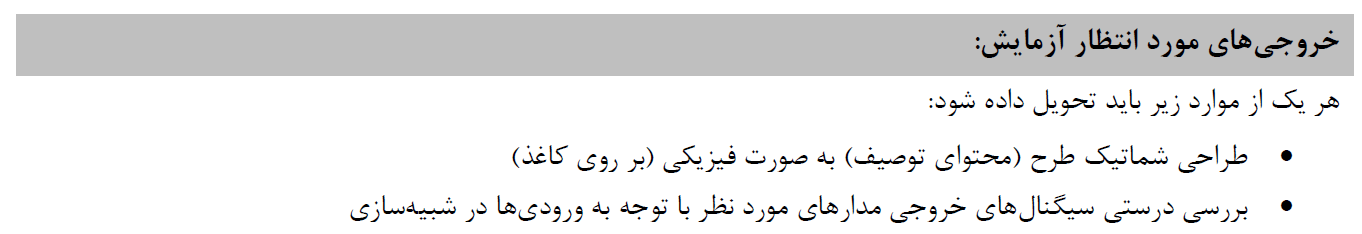


**شکل 1: شرح آزمایش**



**شکل 2: صورت سوال تغییر یافته آزمایش**

خروجی مورد انتظار آزمایش:

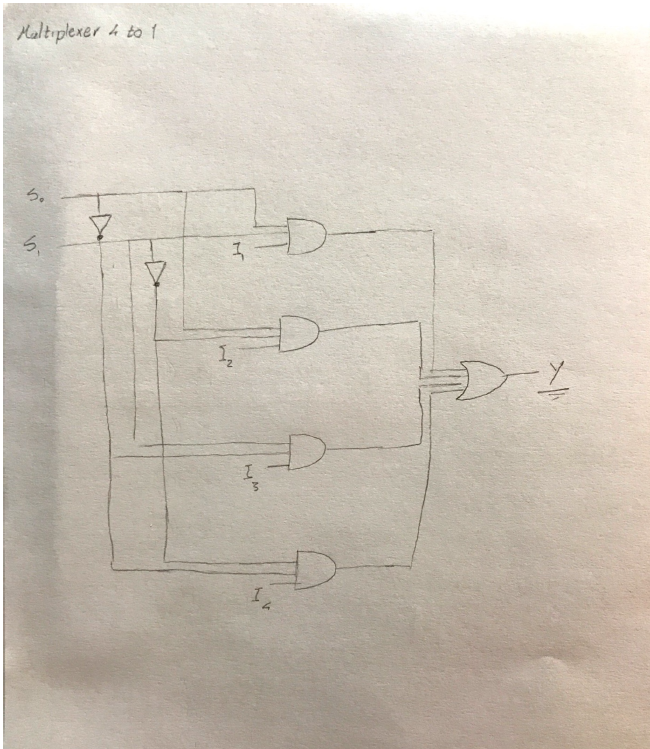


**شکل 3: خروجی مورد انتظار آزمایش**

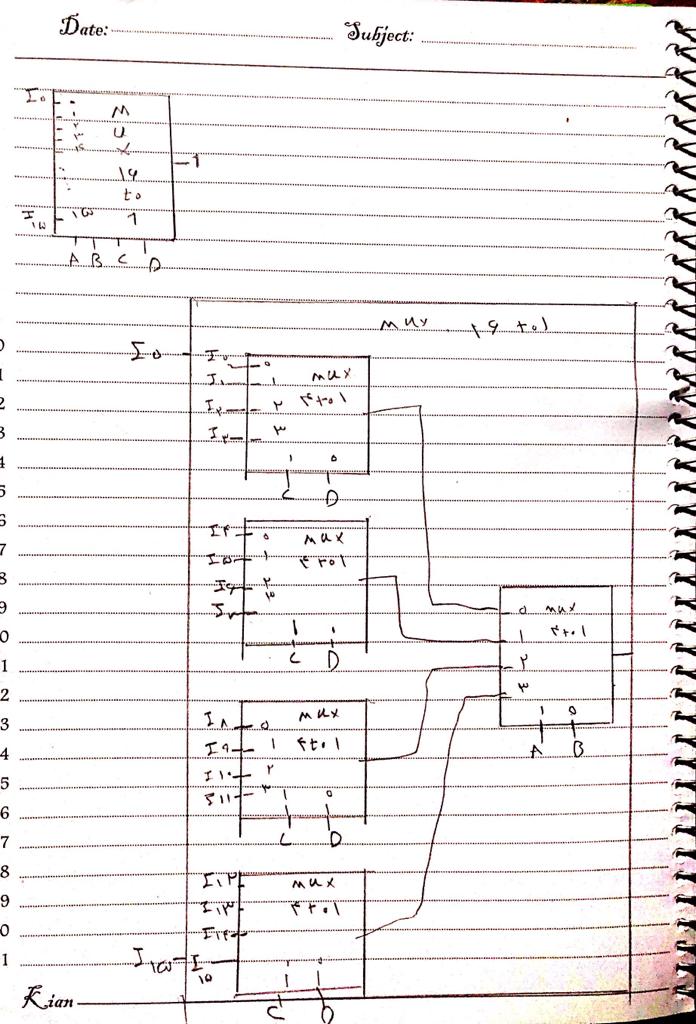
**توضیح آزمایش :**

**مالتی پلکسر (MUX ):**

با استفاده از شکل 4 که در پیش گزارش آمده یک مالتی پلکسر 4 به 1 را با استفاده از شرط های When -else (ارجاع شرطی ) و select (ارجاع انتخابی) طراحی کرده .و کد آن را با استفاده از زبان VHDL در ISE پیاده سازی میکنیم. در قسمت دوم سوال یک مالتی پلکسر 16 به یک را با استفاده از طراحی انجام شده در شکل 5 انجام می دهیم. برای پیاده سازی این ماژول در زبان VHDL ابتدا Component مالتی پلکسر 4 به 1 را تعریف میکنیم. سپس 5 نمونه از آن را ساخته و سپس ورودی و خروجی های ان را PORT MAP کرده و بهم وصل میکنیم.در نهایت برای تست کردن ماژول های خود Test Bench برای هر دو ماژول نوشته ورودی و خروجی های مرتبط را بهم وصل میکنیم و مقادیر دل خواهی را جهت تست به آن می دهیم.



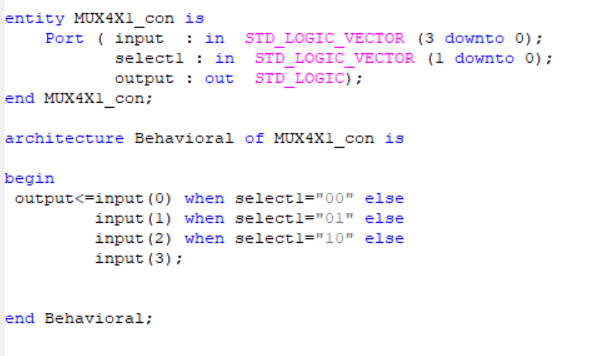
**شکل 4 :طراحی MUX در سطح گیت**



**شکل 5 : طراحی MUX 16 به 1 با استفاده از MUX 4 به 1**

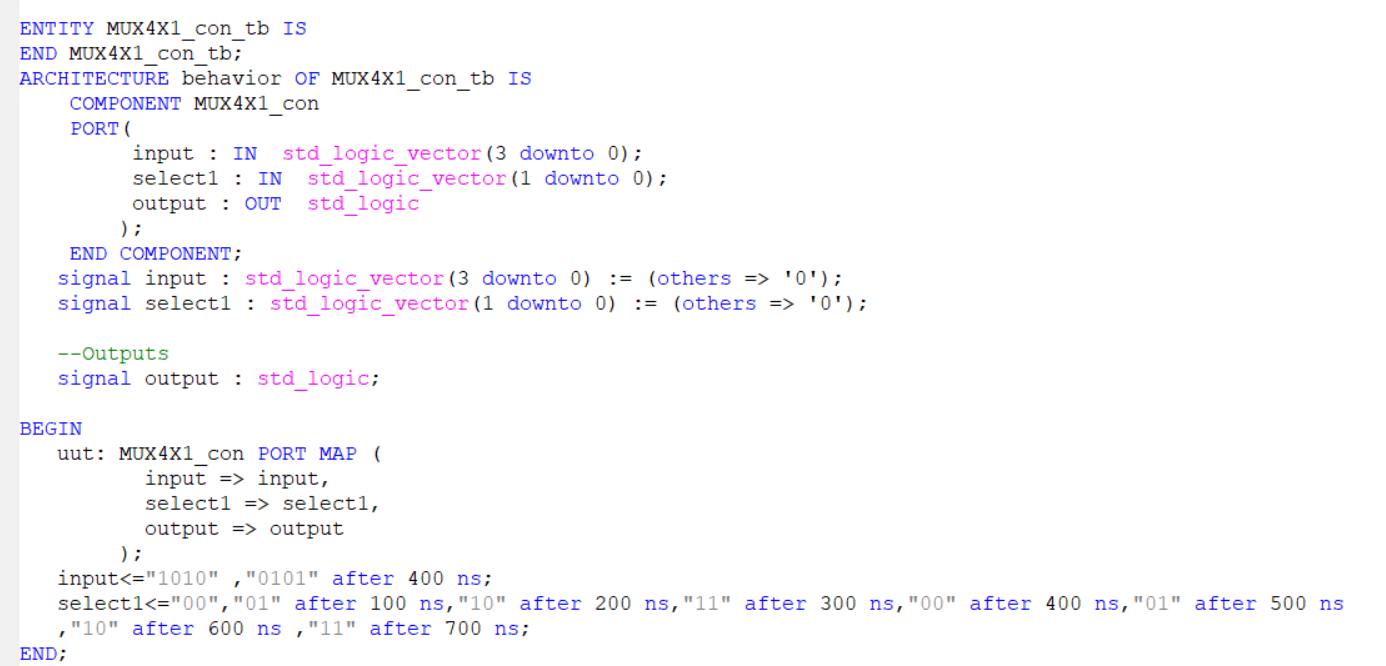
**Behavior of 4 to 1 MUX:**

**(Conditional with-else)**



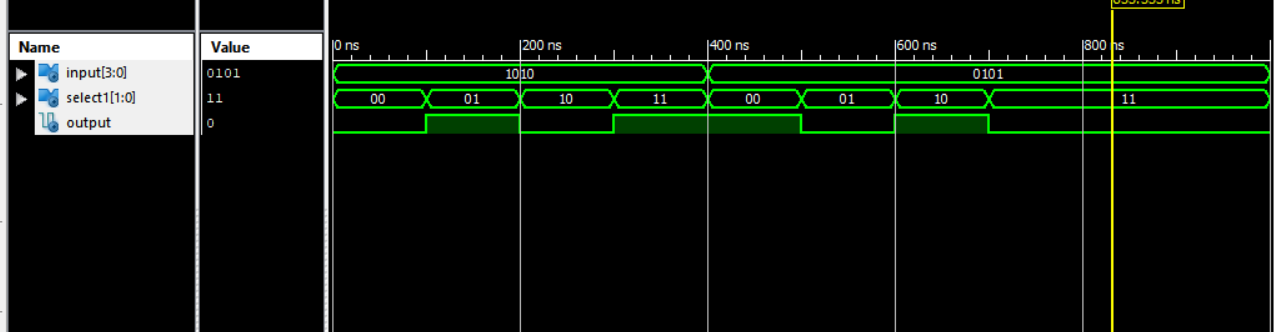
**شکل 6 : توصیف مالتی پلکسر 4 به 1 با ساختار شرطی**

**Test Bench :**



**شکل 7: تست بنچ مالتی پلکسر 4 به 1 با ساختار شرطی**

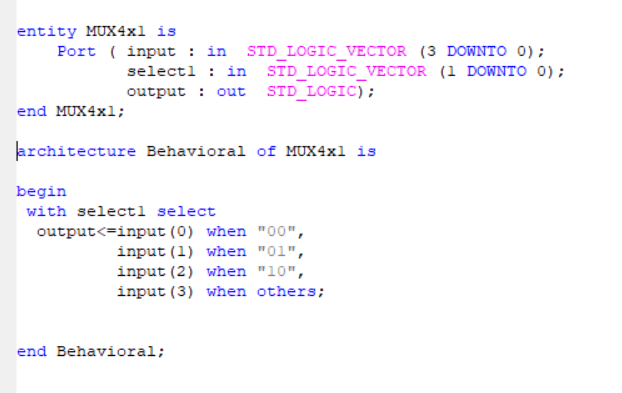
**Result of Simulation in isim** :



**شکل 7-1 : نتایج شبیه سازی**

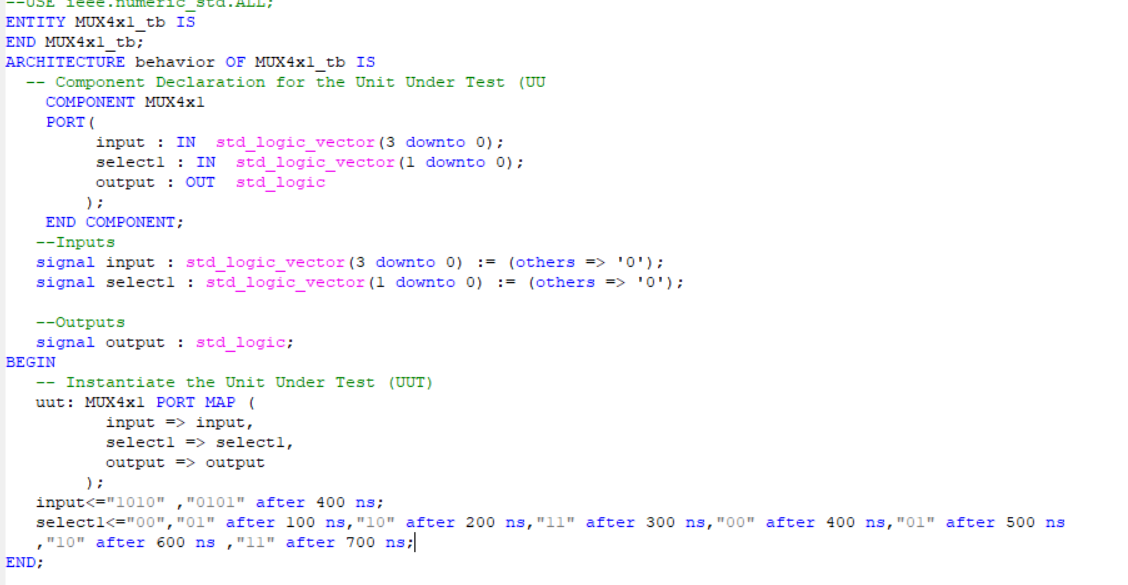
**Behavior of 4 to 1 MUX:**

**(with select )**



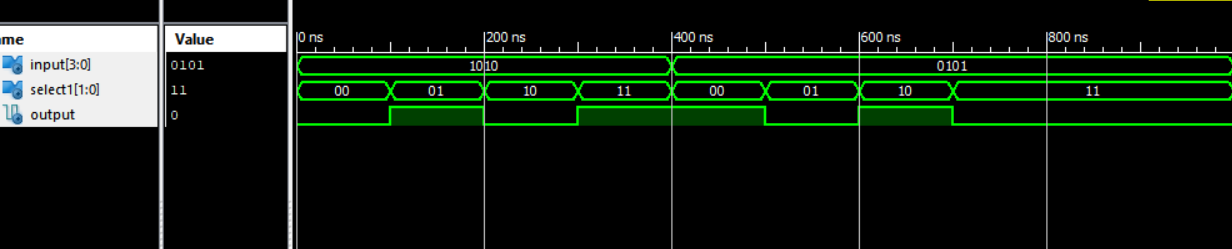
**شکل 8 : توصیف مالتی پلکسر با استفاده از ساختار انتخابی**

**Test Bench:**



**شکل 9: تست بنچ**

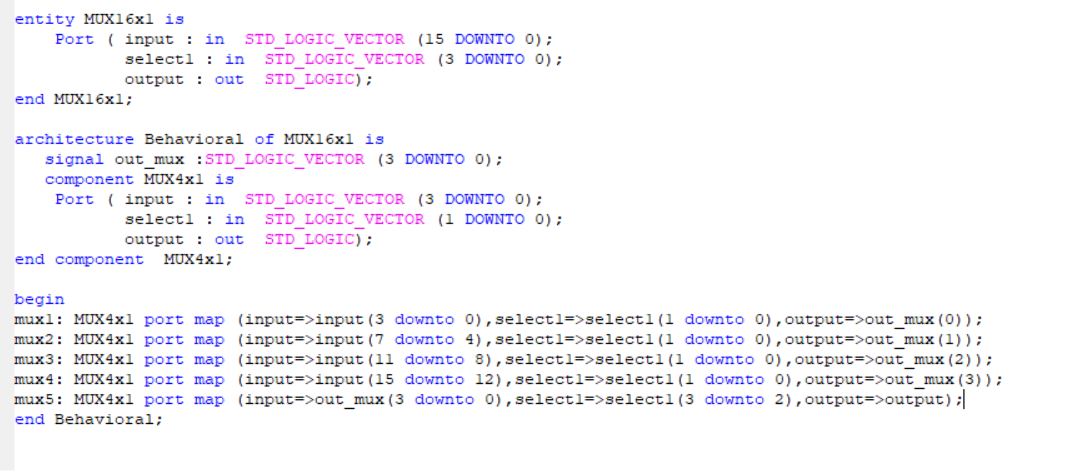
**Result of simulation in isim:**



**شکل 10: نتایج شبیه سازی**

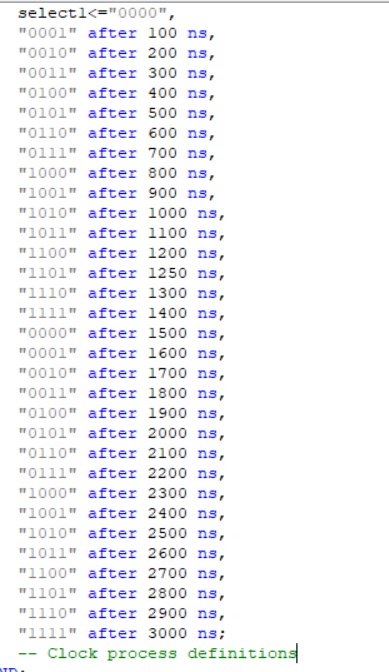
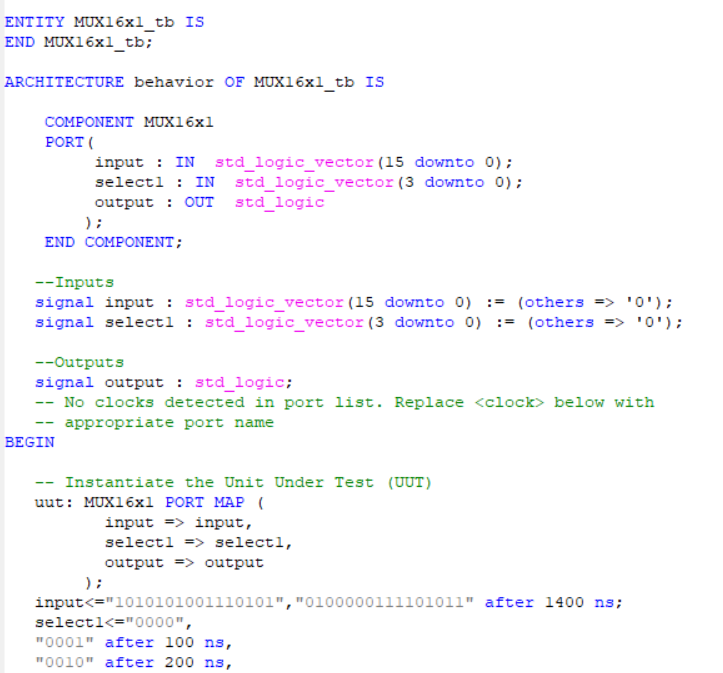
حال با استفاده از5 MUX 4 به 1 یک MUX 16 به 1 را به صورت زیر توصیف میکنیم.

**Behavior of 16 to 1 MUX:**



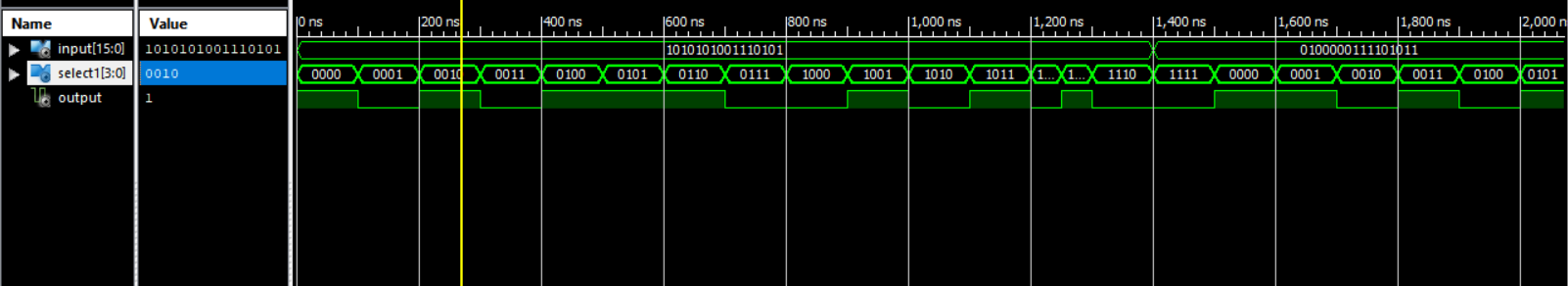
**شکل 11 :توصیف مالتی پلکسر 16 به یک در سطح گیت**

**Test Bench:**



شکل 12 :تست بنچ 16 به 1

**Result of simulation in isim:**

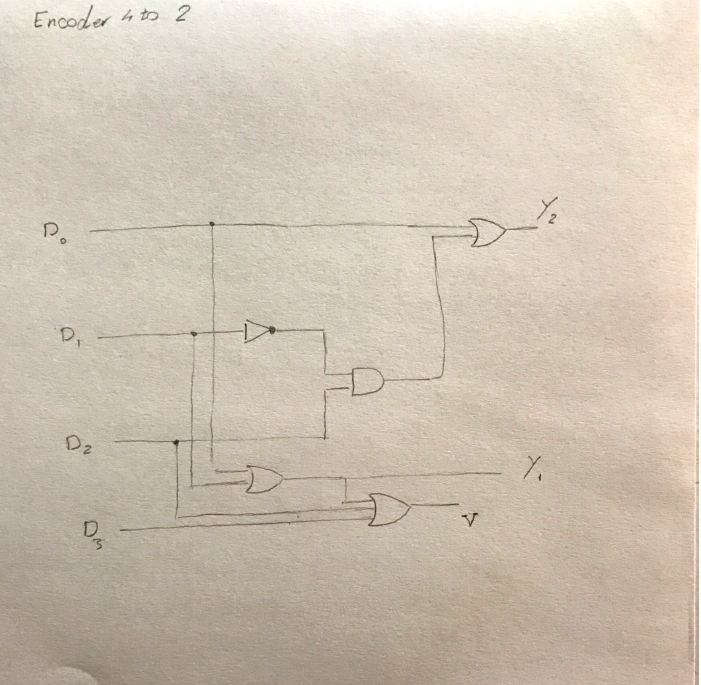


**شکل 13: نتایج شبیه سازی**

**بخش دوم:**

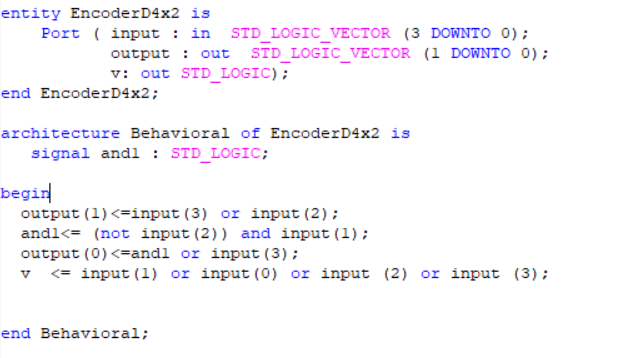
**(Encoder) انکودر:**

در این بخش هدف ما توصیف یک انکودر 4 به 2 در زبان VHDL در سطح گیت می باشد . این طراحی با استفاده از دستورات زبان و مدار طراحی شده در پیش گزارش انجام میدهیم. در شکل زیر یک انکودر الویت دار رسم شده است.



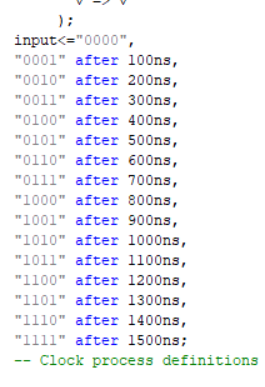
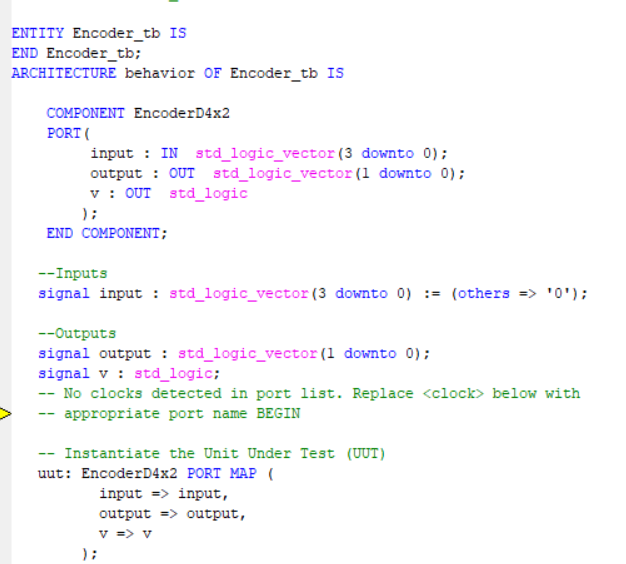
**شکل 14 : مدار انکودر الویت دار در سطح گیت**

**Behavior of 4 to 2 Encoder:**



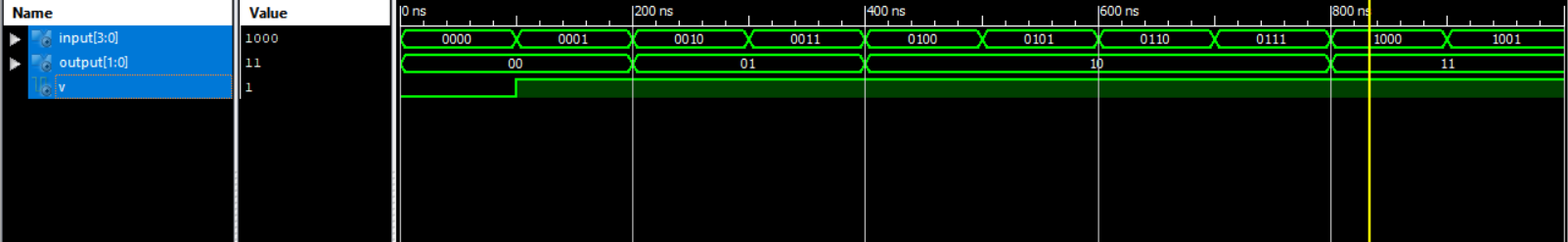
**شکل 15:توصیف انکودر در سطح گیت**

**Test Bench:**



**شکل 16 : تست بنچ انکودر 4 به 2**

**Result of simulation in isim:**

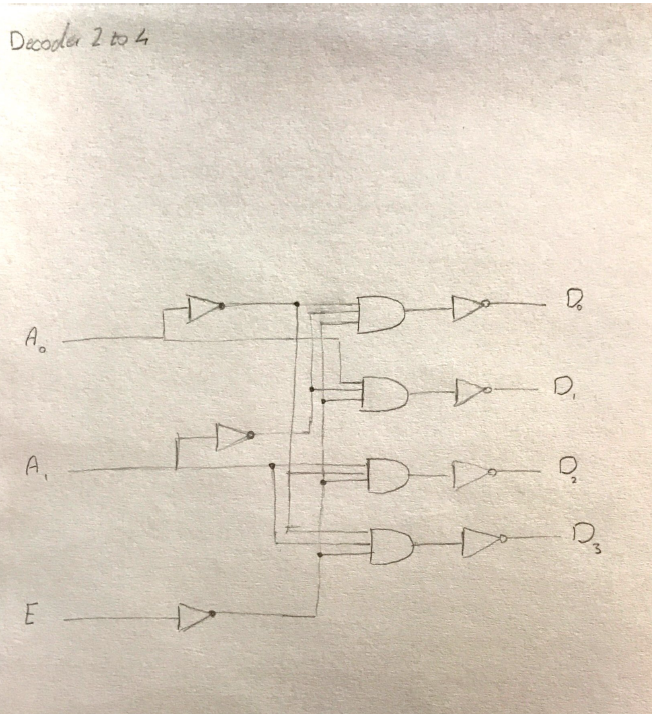


**شکل 17: نتایج شبیه سازی**

**بخش سوم:**

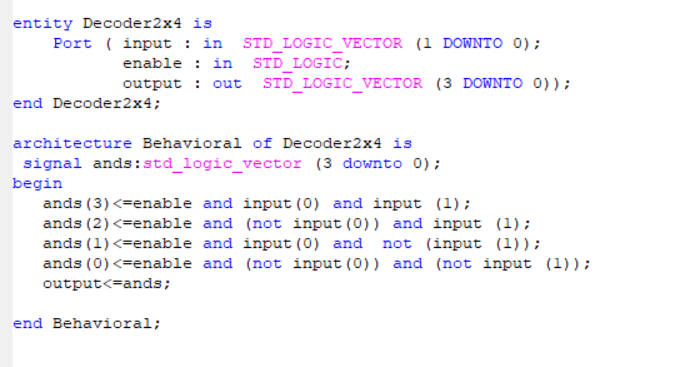
**دیکودر (Decoder) :**

در این بخش هدف ما توصیف یک دیکودر 2 به 4 در زبان VHDL در سطح گیت می باشد . این طراحی با استفاده از دستورات زبان و مدار طراحی شده در پیش گزارش انجام میدهیم.



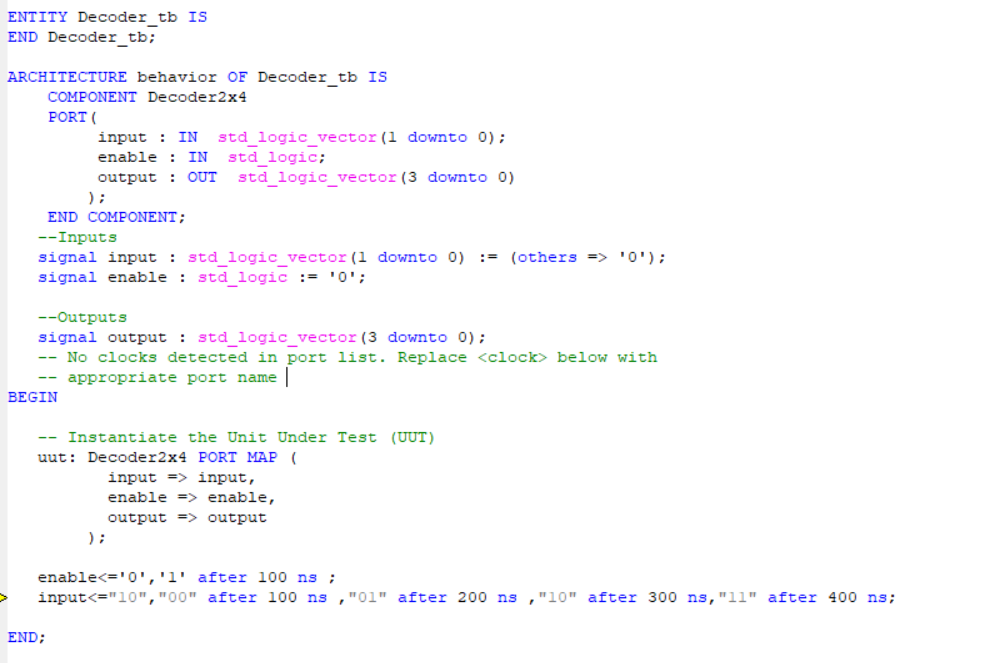
**شکل 18 : مدار رسم شده دیکودر 2 به 4**

**Behavior of 2 to 4 Decoder:**



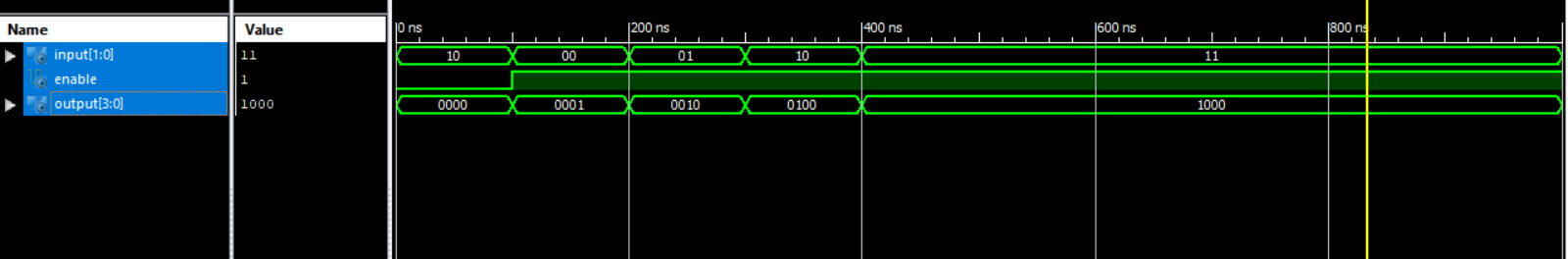
**شکل 19 : توصیف دیکودر 2 به 4 در سطح گیت**

**Test Bench:**



**شکل 20: تست بنچ**

**Result of simulation in isim:**

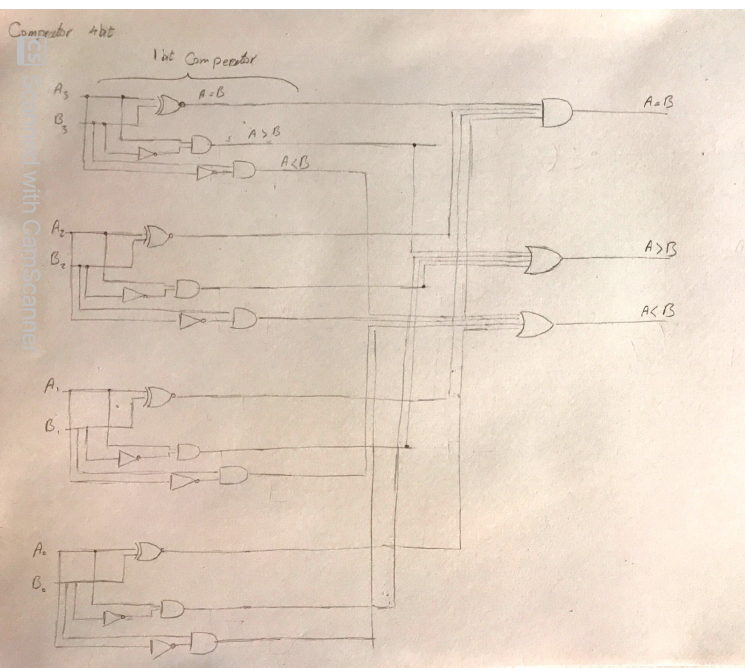


**شکل 21 : نتایج شبیه سازی دیکودر**

**بخش چهارم:**

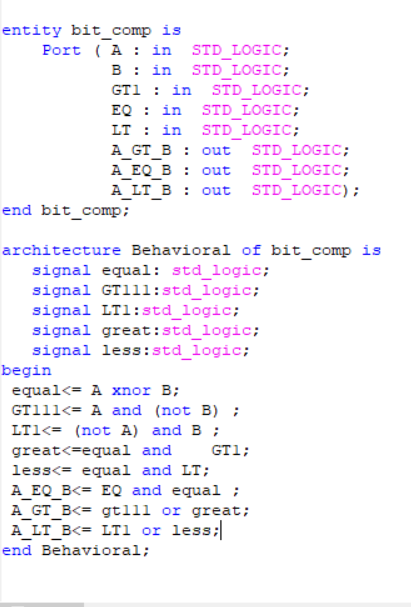
**مقایسه گر (Comparator )**

در این بخش ابتدا یک مقایسه کننده تک بیتی را در سطح گیت در زبان VHDL با استفاده از مدار های طراحی شده در پیش گزارش توصیف میکنیم. سپس با استفاده از 4 مقایسه گر تک بیتی یک مقایسه گر 4 بیتی را طراحی میکنیم. که ابتدا از با ارزش ترین بیت شروع به مقایسه کرده و در هر مرحله در صورت برابر بودن آن دو بیت به بیت کم ارزش تر مراجعه میکنیم. در زبان VHDL ابتدا Component مقایسه گر تک بیتی را برای مقایسه گر 4 بیتی تعریف میکنیم. سپس 4 نمونه از آن را ساخته و ورودی و خروجی ها را با استفاده از Port MAP به هم وصل می کنیم و در نهایت نیز با نوشتن حالت های مختلف در Test Bench این دو مقایسه گر خود را آزمایش میکنیم و با استفاده شبیه سازی نتیجه آن را مشاهده میکنیم.



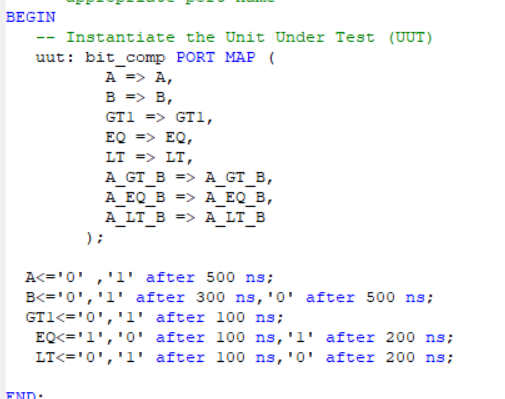
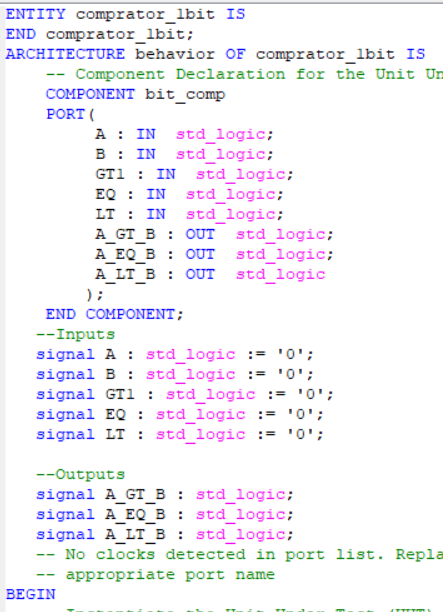
**شکل 22: توصیف مقایسه گر 4 بیتی با استفاده از تک بیتی**

**Behavior of 1 bit comparator :**



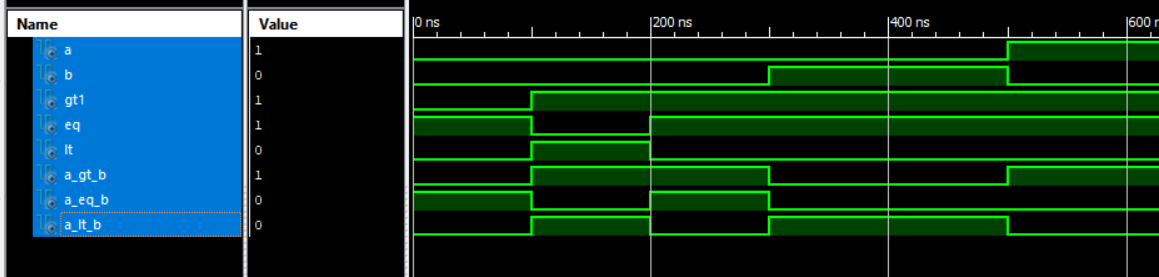
**شکل 23: توصیف مقایسه گر تک بیتی**

**Test Bench:**



**شکل 24: تست بنچ**

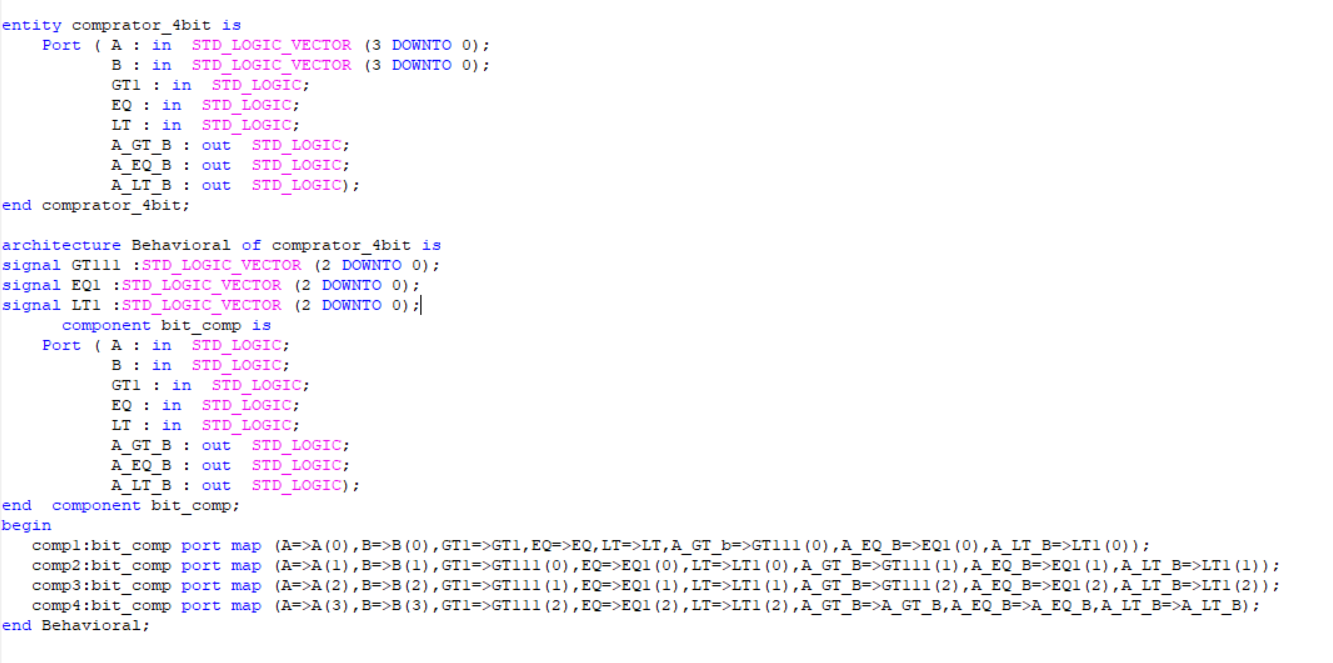
**Result of simulation in isim:**



**شکل 25 : نتایج شبیه سازی**

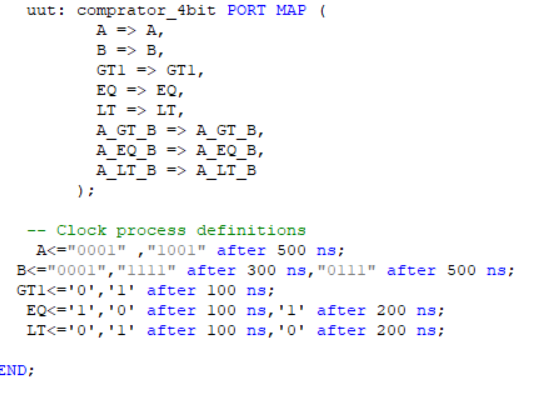
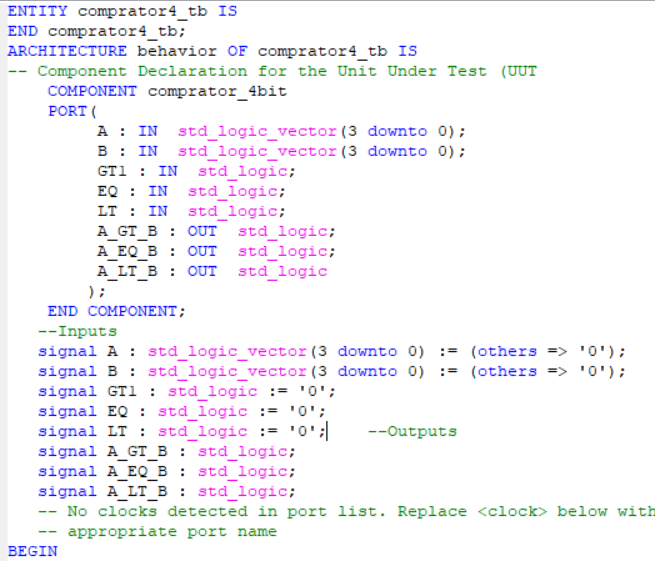
حال با استفاده از 4 مقایسه گر تک بیتی یک مقایسه گر 4 بیتی زا توصیف میکنیم.

**Behavior of 4 bit comparator:**



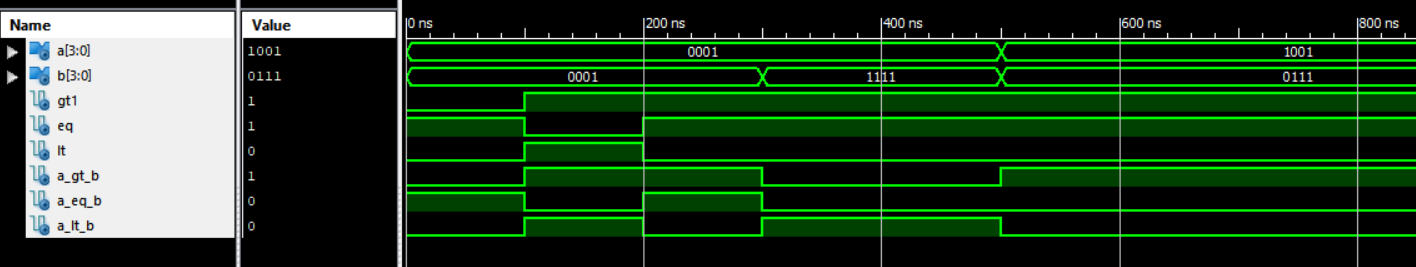
**شکل 26: توصیف مقایسه گر 4 بیتی در زبان**

**Test Bench:**



**شکل 27: تست بنچ**

**Result of simulation in isim:**



**نتیجه گیری:**

به طور کلی و در اين آزمايش هدف آشنايي با نحوه عملکرد و پياده سازي هر يک از مدارهاي پايه در سطح تجريد گيت است.و یادگیری بهتر زبان VHDL و کار با ساختار ارجاع شرطی و ارجاع انتخابی که با پیاده سازی ماژول های مالتی پلکسر این کار را انجام دادیم . سپس یاد آوری درس مدار منطقی و آموزش کار با زبان VHDL که با استفاده از ماژول های دیکودر انکودر و مقایسه گر ها این کار به درستی انجام شد.