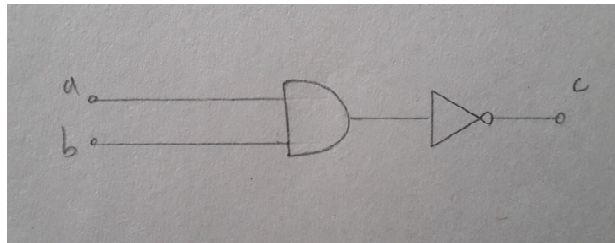


## گزارش دستور کار دوم آزمایشگاه معماری کامپیوتر

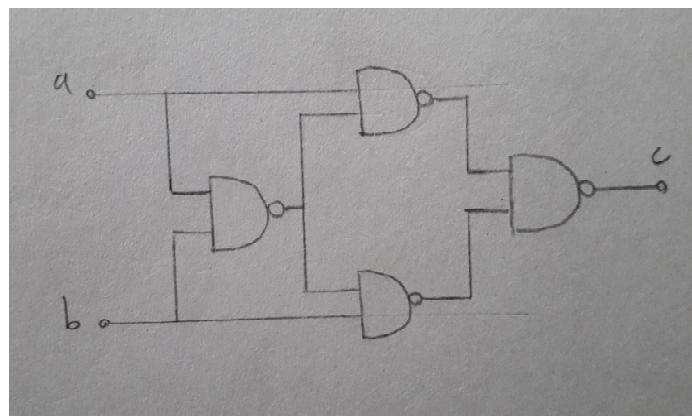
نگار موقتیان، ۹۸۳۱۰۶۲

### قسمت اول

با توجه به دستور کار، در ابتدا گیت XOR با استفاده از گیت‌های NAND به شکل زیر پیاده سازی شده است. برای آشنایی و تسلط بیشتر، گیت NAND استفاده شده در این قسمت نیز به صورت ساختاری و در ماژولی جداگانه طراحی شده.



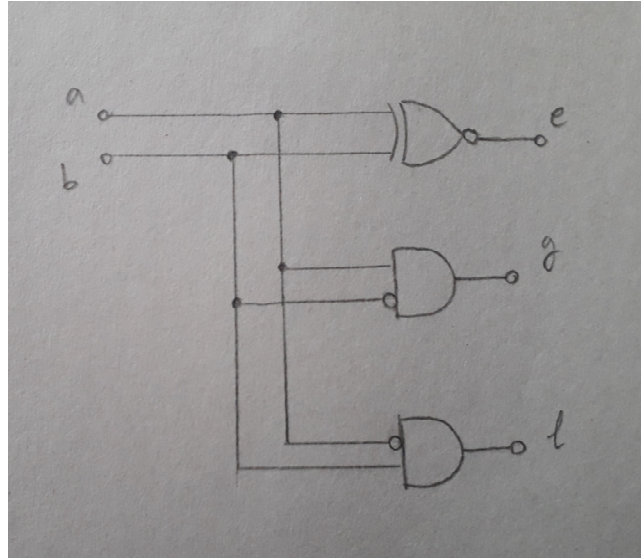
ساختار درونی گیت NAND



ساختار درونی گیت XOR

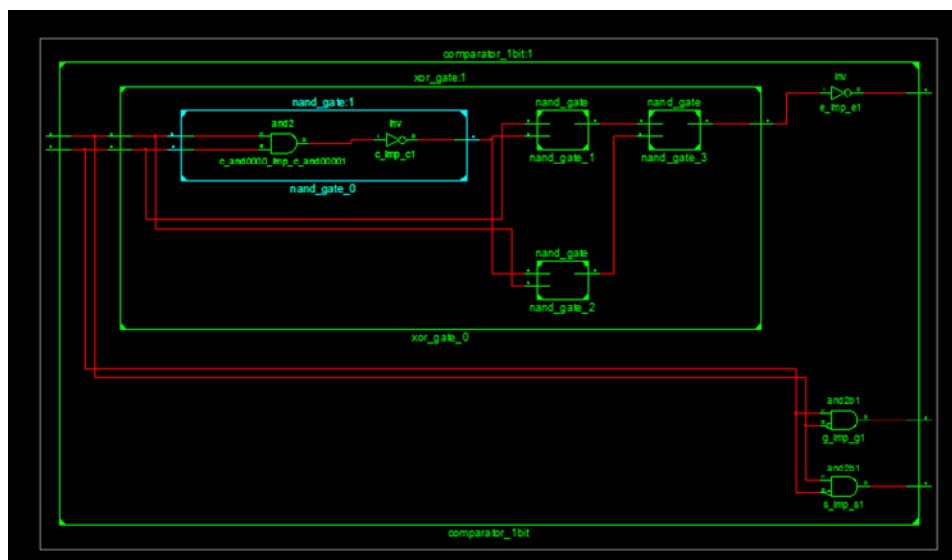
پس از آن با استفاده از گیت XOR ساخته شده، یک مدار مقایسه کننده تک بیتی به صورت زیر ساخته شده است. زمانی که  $a = 0$  و  $b = 1$  باشد یعنی  $a \cdot b = 1$  عدد  $a$  از  $b$  کوچکتر بوده و خروجی 1 برابر یک شده است.

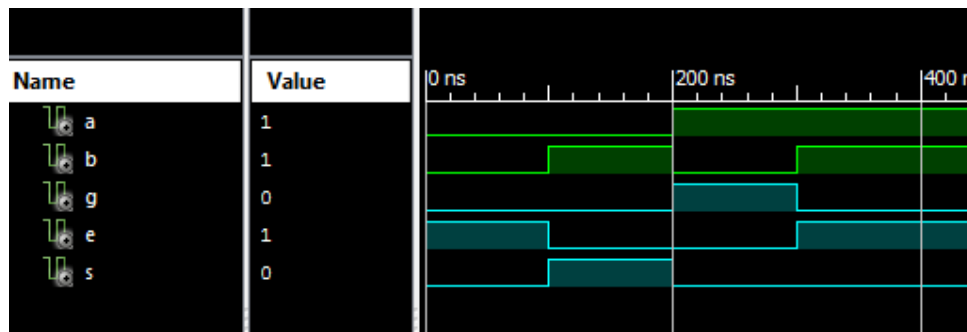
می‌شود. همچنین زمانی که  $a = 1$  و  $b = 0$  باشد یعنی  $a.b' = 1$ ، عدد  $a$  از  $b$  بزرگ‌تر بوده و خروجی  $g$  برابر یک می‌شود. در غیر این صورت اگر  $a$  و  $b$  مساوی باشند، XOR آن‌ها برابر با صفر شده و نقیض آن یعنی  $e$  برابر با یک می‌شود.



ساختار درونی گیت XOR

طرح شماتیک مدار شبیه‌سازی شده و همچنین شکل سیگنال‌های خروجی مدار به ازای مقادیر مختلف داده شده در test bench در ادامه آمده‌است. در شکل زیر همچنین ساختار درونی گیت‌های XOR و NAND مشاهده می‌شود.





## قسمت دوم

هدف از این قسمت از آزمایش، ساخت مقایسه کننده‌ای برای اعداد دوبیتی و با استفاده از یک MUX 4:1 و حداقل تعداد گیت می‌باشد. برای این کار دو بیت پر ارزش a و b را به ورودی select مالتی پلکسر متصل می‌کنیم.

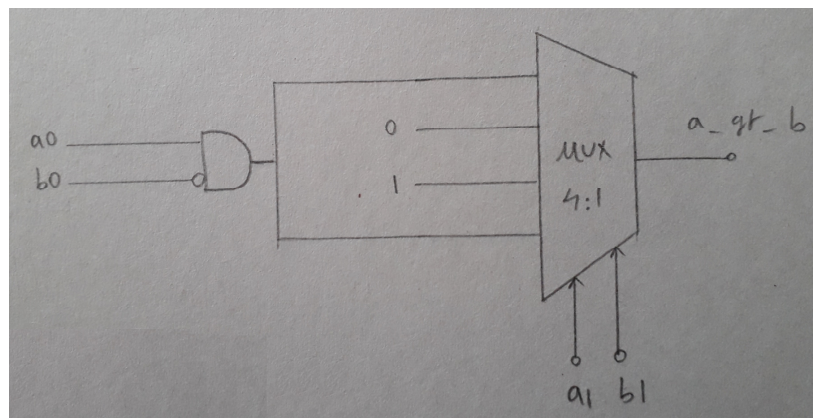
۱. اگر  $a_1 = 0$  و  $b_1 = 0$  آنگاه برای مقایسه این دو عدد باید به بیت کم‌ارزش مراجعه کنیم. اگر  $a_0 = 1$  و  $b_0 = 0$  باشد، یا به عبارتی اگر  $a_0b_0' = 1$  باشد عدد a بزرگ‌تر است و خروجی برابر با یک می‌شود.

۲. اگر  $a_1 = 0$  و  $b_1 = 1$  آنگاه عدد a به طور حتم از b کوچک‌تر است و خروجی صفر می‌شود.

۳. اگر  $a_1 = 1$  و  $b_1 = 0$  آنگاه عدد a به طور حتم از b بزرگ‌تر است و خروجی یک می‌شود.

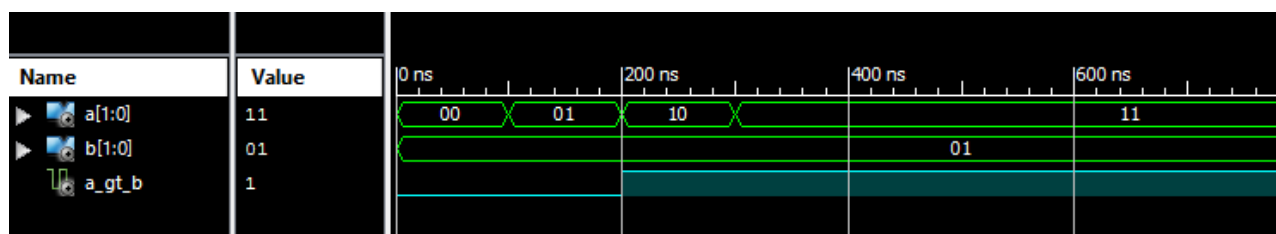
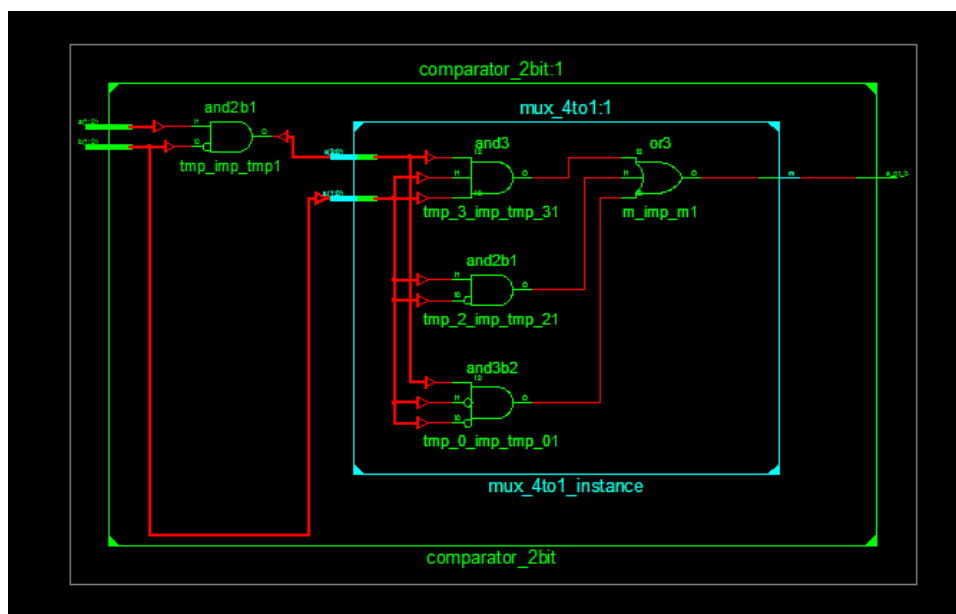
۴. اگر  $a_1 = 1$  و  $b_1 = 1$  آنگاه مانند قسمت اول باید به مقایسه بیت‌های کم‌ارزش عدد بپردازیم و اگر  $a_0b_0' = 1$  باشد عدد a بزرگ‌تر است و خروجی برابر با یک می‌شود.

با توجه به این رفتار می‌توان مدار مورد نظر را به شکل زیر طراحی کرد.



برای این کار تنها به یک مالتی پلکسر، یک گیت AND و یک گیت NOT نیاز است.

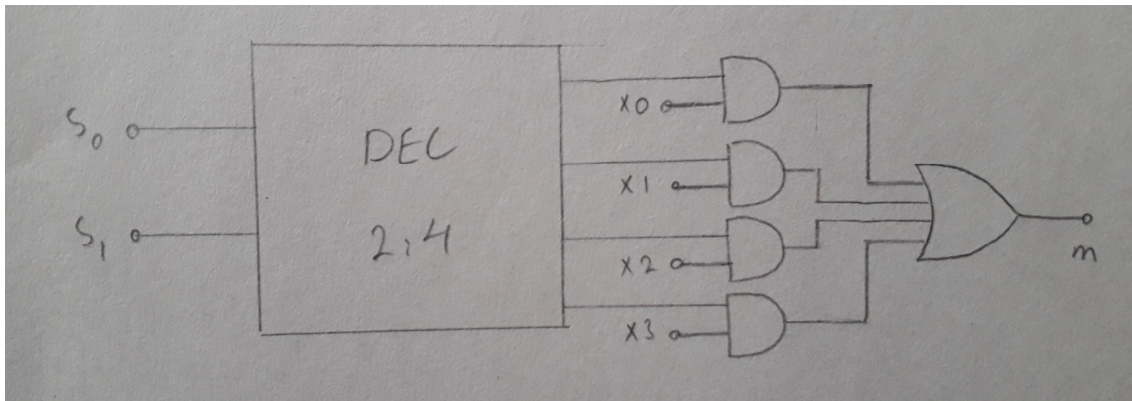
طرح شماتیک مدار شبیه سازی شده و همچنین شکل سیگنال های خروجی مدار به ازای مقادیر مختلف داده شده در test bench در ادامه آمده است. در شکل زیر همچنین ساختار درونی مالتی پلکسری که مطابق مدار موجود در پیش گزارش طراحی شده مشاهده می شود.



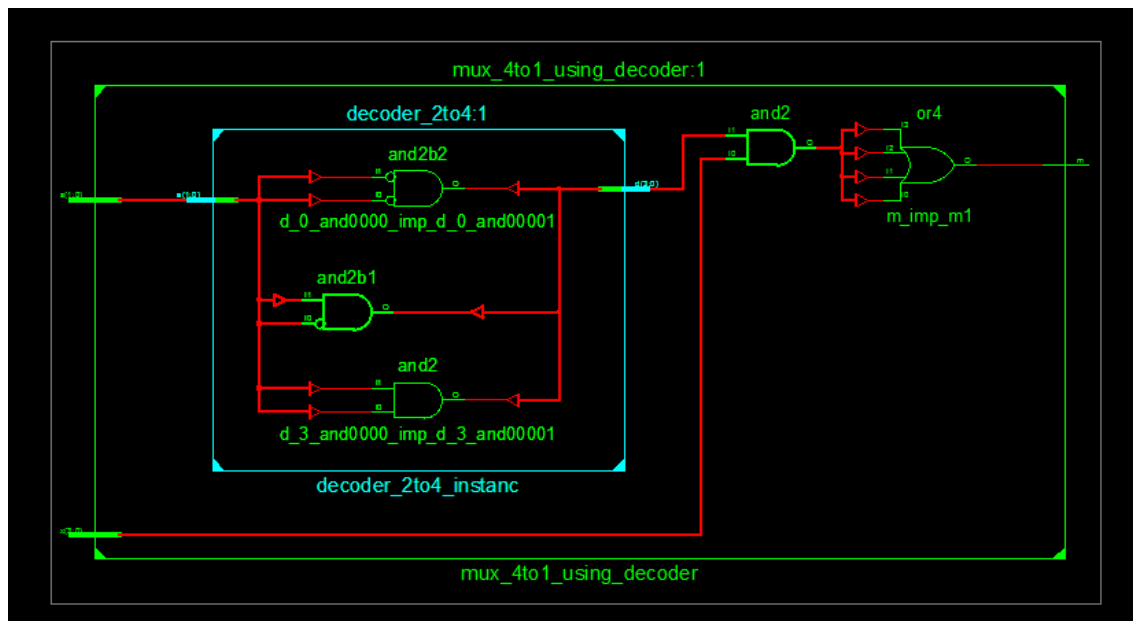
### قسمت سوم

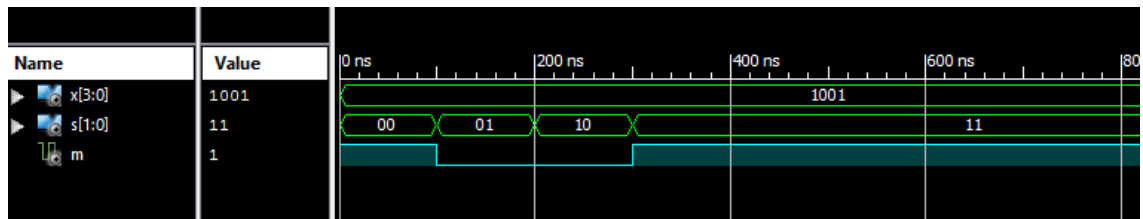
در این قسمت از آزمایش، مطابق گزارش کار باید یک مالتی پلکسر ۴:۱ با استفاده از Decoder 2:4 و گیت های پایه AND و OR طراحی شود. برای این کار به خواص MUX و همچنین Decoder توجه می کنیم. در یک مالتی پلکسر آدرس ورودی ای که می خواهیم در خروجی نمایان شود را به پایه های select می دهیم و انتظار

داریم مقدار خروجی برابر با ورودی مورد نظر شود. در یک کدگشا نیز یک آدرس به عنوان ورودی می‌دهیم و خروجی متناظر با آن یک می‌شود. حال اگر این خروجی‌ها را با ورودی‌ها AND کنیم تمام خروجی‌ها صفر می‌شوند به جز یکی که آدرس آن را داده‌ایم و این خروجی برابر با ورودی مورد نظر می‌شود. بنابراین اگر نتیجه تمامی AND ها را OR کنیم، مقدار ورودی خواسته شده به خروجی منتقل می‌شود. شکل مدار توصیف شده در بالا به صورت زیر می‌باشد.

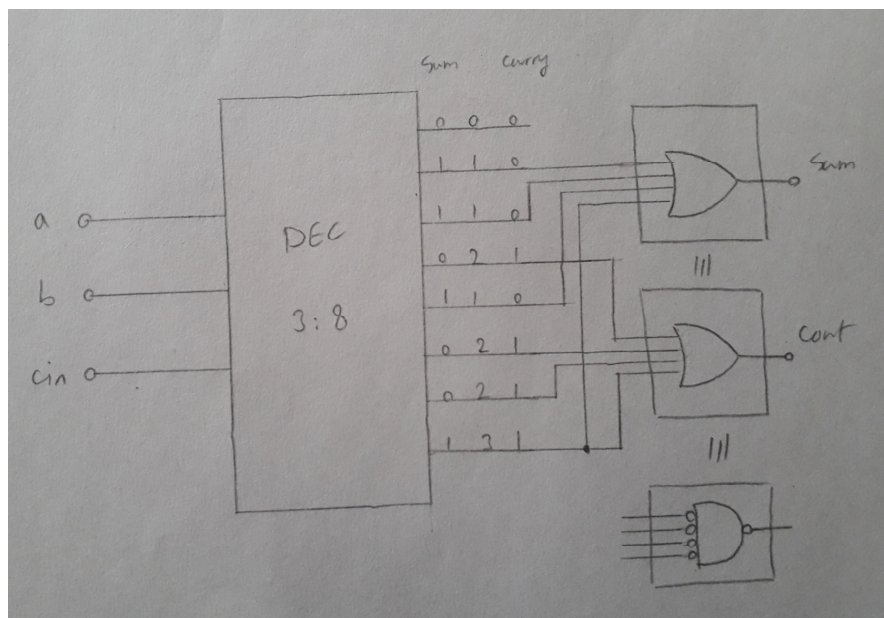


طرح شماتیک مدار شبیه‌سازی شده و همچنین شکل سیگنال‌های خروجی مدار به ازای مقادیر مختلف داده شده در test bench در ادامه آمده‌است. در شکل زیر همچنین ساختار درونی کدگشا که مطابق مدار موجود در پیش‌گزارش طراحی شده مشاهده می‌شود.



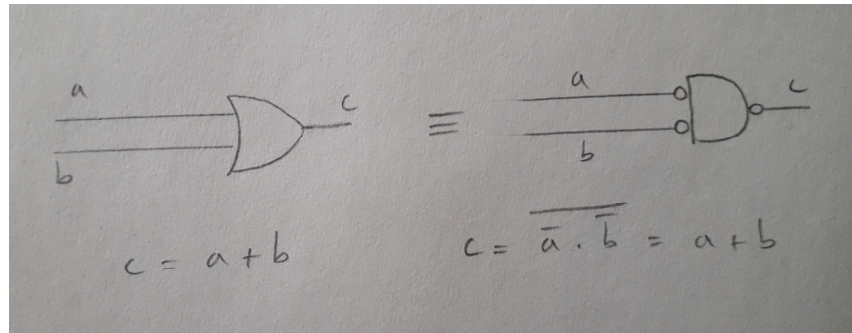


در قسمت بعدی آزمایش ساخت یک تمام جمع کننده با استفاده از یک کدگشا و گیت های NAND خواسته شده. به دلیل این که یک تمام جمع کننده سه ورودی  $a$ ،  $b$  و  $cin$  دارد از یک کدگشای ۳:۸ استفاده شده است. در شکل زیر بر روی خروجی های کدگشا سه عدد نوشته شده که اطلاعات ورودی ها را در صورتی که این خروجی یک شود بیان می کنند. عدد میانی جمع دسیمال  $a$ ،  $b$  و  $cin$  می باشد. همچنین ستون سمت راست carry ایجاد شده از جمع این سه عدد و ستون چپ  $sum$  این سه عدد می باشد. برای مثال اگر  $a = 1$ ،  $b = 1$  و  $cin = 0$  باشد ورودی کدگشا به صورت 011 بوده و در نتیجه خروجی شماره سه روشن می شود. در این حالت جمع اعداد ورودی ۲ بوده و باید داشته باشیم  $sum = 0$  و  $cout = 1$ . بنابراین با توجه به مقادیر این سه ستون، خروجی هایی که ستون  $sum$  آن ها برابر ۱ است را با هم OR کرده و خروجی هایی که ستون  $carry$  آن ها برابر ۱ است را نیز با هم OR می کنیم و به عنوان  $sum$  و  $carry$  نهایی ارائه می دهیم.

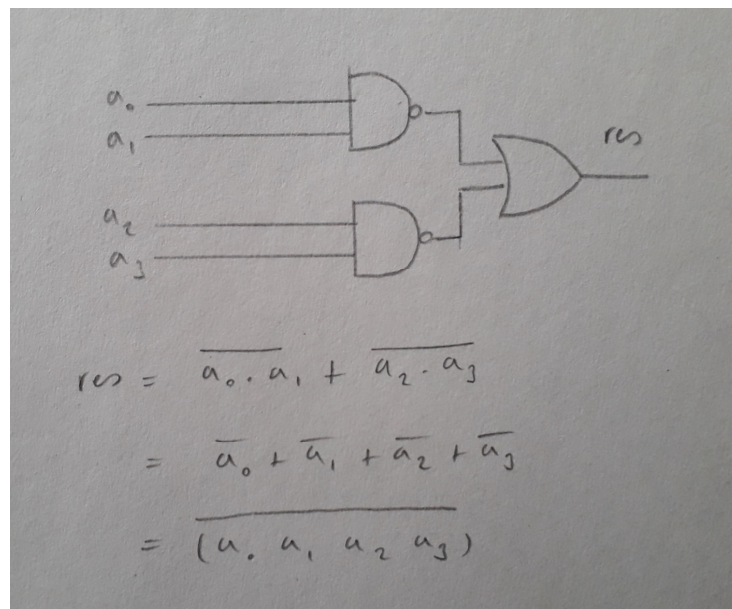




تنها بحثی که باقی می ماند این است که در صورت آزمایش خواسته شده که تنها از گیت های NAND استفاده کنیم. طبق شکل زیر هر گیت OR را می توان به یک گیت NAND که ورودی هایش نقیض شده اند تبدیل کرد.



بنابراین در شکل اصلی می توان گیت OR را که در یک مستطیل مشخص شده با گیت NAND مشخص شده جایگزین کرد. برای ساخت گیت NAND چهار ورودی نیز می توان از مدار زیر استفاده کرد.



و به این شکل مدار خواسته شده کامل می شود. طرح شماتیک مدار شبیه سازی شده و همچنین شکل سیگنال های خروجی مدار به ازای مقادیر مختلف داده شده در test bench در ادامه آمده است.

