

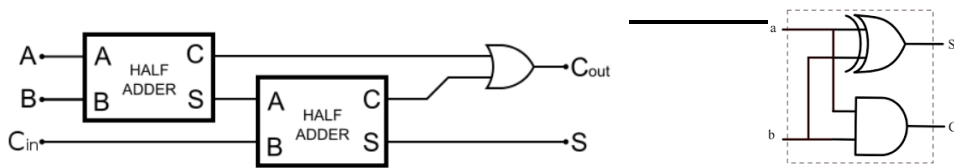


به نام خدا

تکلیف سری اول درس زبان‌های توصیف سخت‌افزار و مدارات

برای ارسال تکالیف، حتما به نکات زیر توجه کنید:

- برای هر سوال در نرم افزار ModelSim ، فایل جداگانه‌ای ایجاد کنید.
- تمامی سوالات می‌بایست شبیه سازی شوند؛ بنابراین لازم است برای هر سوال Test Bench مناسب نوشته و آن را ضمیمه کنید.
- علاوه بر ارسال فایل جواب سوال و فایل شبیه سازی، می‌بایست از شکل موج‌های موجود در شبیه سازی Screenshot گرفته و آن‌ها را با کیفیت مناسب (به طوری که اسامی سیگنال‌ها و شکل موج‌ها واضح باشند) ارسال کنید.
- لازم نیست تمامی فایل‌های موجود در پوشه پروژه را ارسال کنید!! تنها فایل v. جواب، فایل v. Testbench و تصاویر شکل موج‌های شبیه سازی شده را ارسال کنید.
- در صورت برخی سوالات قید شده است که برای سوال، گزارش کوتاهی بنویسید؛ این گزارش را به صورت مختصر و در فرمت pdf به همراه بقیه فایل‌هایی که در مورد قبل اشاره شد، ارسال کنید.
- حتی الامکان اسامی سیگنال‌ها و متغیرها را با مسمی انتخاب کنید و همچنین با نظم و ترتیب برنامه بنویسید.
- توجه کنید که برنامه‌ها باید تماما قابل سنتز باشند.
- برای نام‌گذاری فایل‌های ارسالی به شکل زیر عمل کنید:
فایل اصلی جواب : Q2_3.v
فایل Test Bench : TestQ2_3.v
تصاویر شکل موج‌های شبیه سازی: ScrQ2_3_1.jpg, ScrQ2_3_2.jpg, ScrQ2_3_3.jpg, ...
فایل گزارش (در صورت لزوم) : ReportQ_2_3.pdf
- تمامی فایل‌های خود را در یک فایل zip قرار دهید. نام این پوشه باید به فرمت Hwx_Student ID باشد. برای مثال : Hw1_9526795.zip
- در نهایت این فایل را در قسمت مربوطه در سامانه Yekta آپلود کنید.



طراحی اول: در این سوال می‌توانید از هر یک از سطوح Gate, Data flow, یا Behavioral استفاده کنید. در نهایت لازم است برای هر یک از ماژول‌ها یک test bench به گونه‌ای نوشته شود که حالات ممکن برای آن ماژول را دربر بگیرد. هر یک از این تست بنچ‌ها باید دربردارنده‌ی حد اقل سه جفت عدد ورودی متفاوت برای ماژول باشند.

(۱) یک half adder تک بیتی طراحی نمایید.

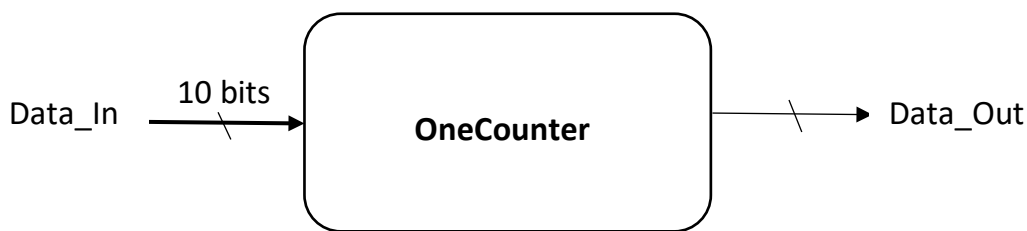
(۲) با استفاده از half adder قسمت ۱ یک full adder تک بیتی طراحی نمایید.

(۳) تنها با استفاده از ماژول قسمت ۲، ماژولی برای شمارش تعداد ۱‌های یک عدد باینری ۲۰ بیتی طراحی کنید. به عنوان مثال تعداد ۱‌های عدد باینری 1010100010 برابر ۴ است.

پورت‌های ماژول عبارت‌اند از:

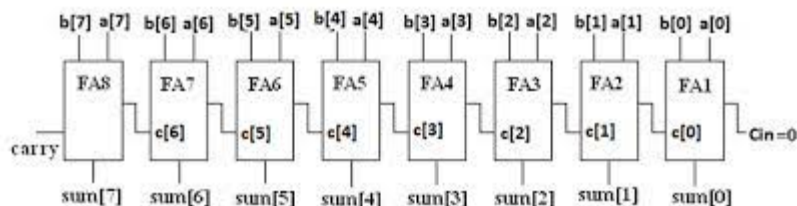
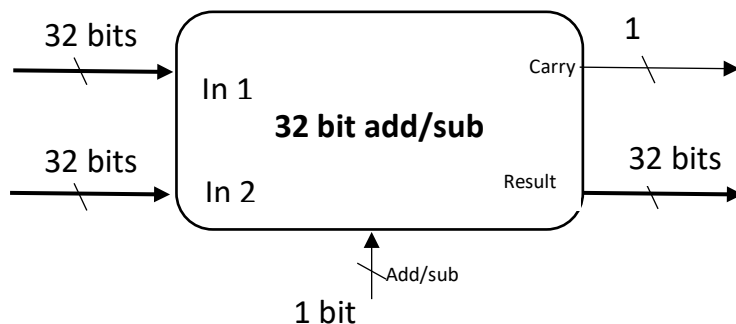
✓ ورودی 10 بیتی Data_In که باید تعداد ۱‌های آن شمرده شود.

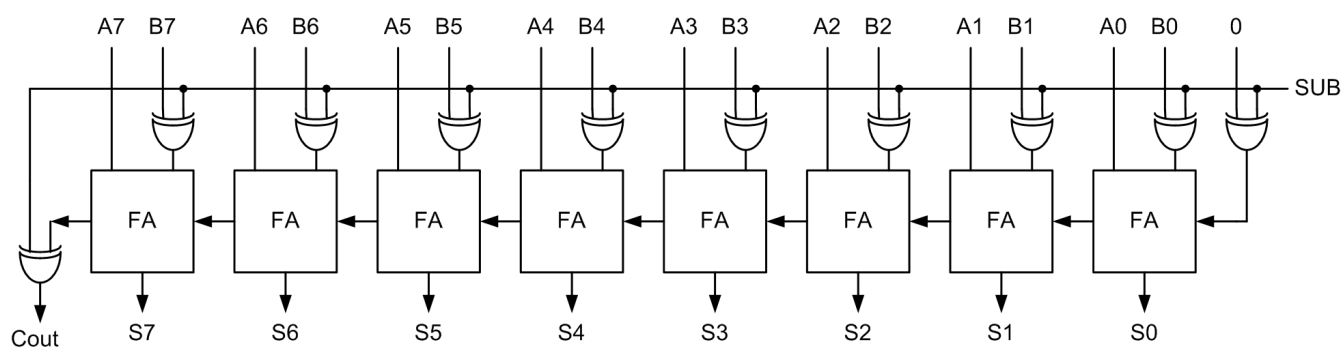
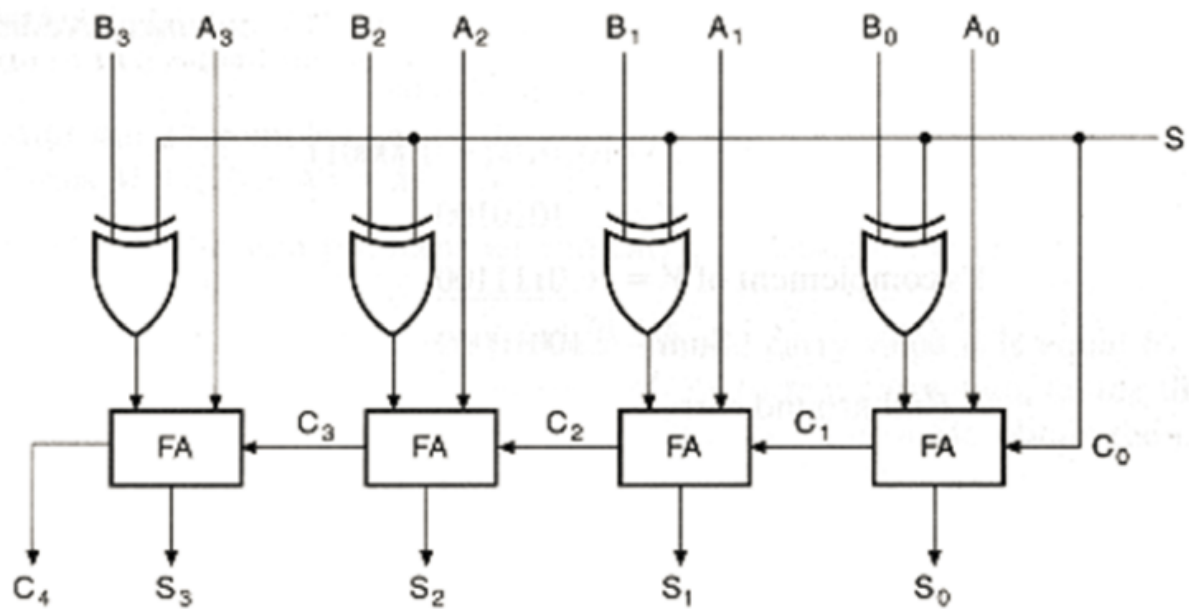
✓ خروجی 3 بیتی Data_Out که تعداد ۱‌ها را مشخص می‌کند.



(۴) با استفاده از full adder تک بیتی‌ای که در قسمت ۲ طراحی کردید، یک جمع‌کننده ۸ بیتی بسازید.

(۵) با استفاده از تعدادی نمونه از ماژول جمع‌کننده طراحی شده در قسمت ۴ یک جمع و تفریق‌کننده ۳۲ بیتی طراحی کنید. یک جمع و تفریق‌کننده ۳۲ بیتی ماژولی است که دارای دو ورودی ۳۲ بیتی برای اعداد ورودی و یک ورودی یک بیتی برای تعیین جمع یا تفریق کردن اعداد ورودی است. ماژول با یک شدن پایه کنترلی عمل جمع، و با صفر شدن عمل تفریق را انجام می‌دهد.





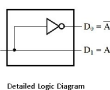
1-to-2 Line Decoder

Input		Output	
A	D ₂	D ₁	D ₀
0	1	0	0
1	0	1	1

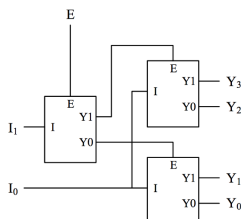
We need equation for every output wire

$$D_0 = A' \quad (\text{i.e. } m_0)$$

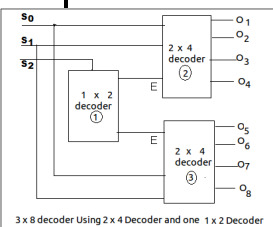
$$D_1 = A \quad (\text{i.e. } m_1)$$



طراحی دوم: در این قسمت هدف طراحی دیگر هایی با اندازه های مختلف با استفاده از یک دیگر پایه ۲ به یک است.

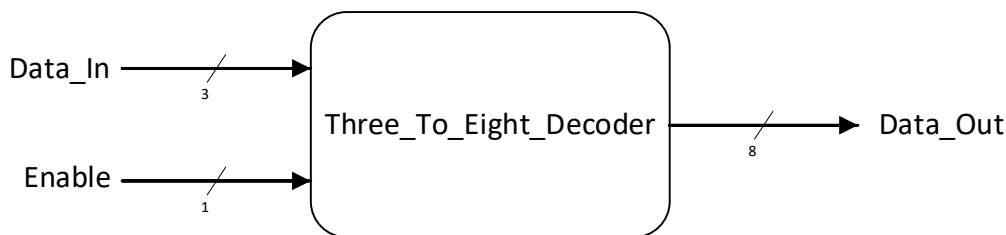


The circuit diagram above implements a 2-to-4 Decoder via cascading of 1-to-2 Decoders.



پورت های ماژول عبارت اند از:

- ✓ ورودی 3 بیتی Data_In که مشخص می کند کدام بیت خروجی با 1 شود.
- ✓ ورودی تک بیتی Enable که وقتی 1 باشد ماژول کار می کند و در غیر این صورت تمامی بیت های خروجی 0 می شوند.
- ✓ خروجی 8 بیتی Data_Out که در صورت 1 بودن Enable و با توجه به Data_In یکی از بیت های آن 1 می شود.

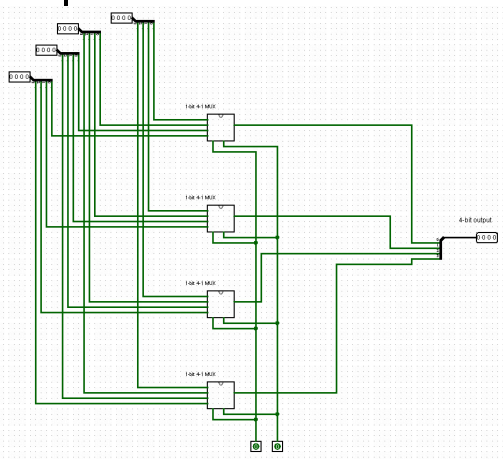


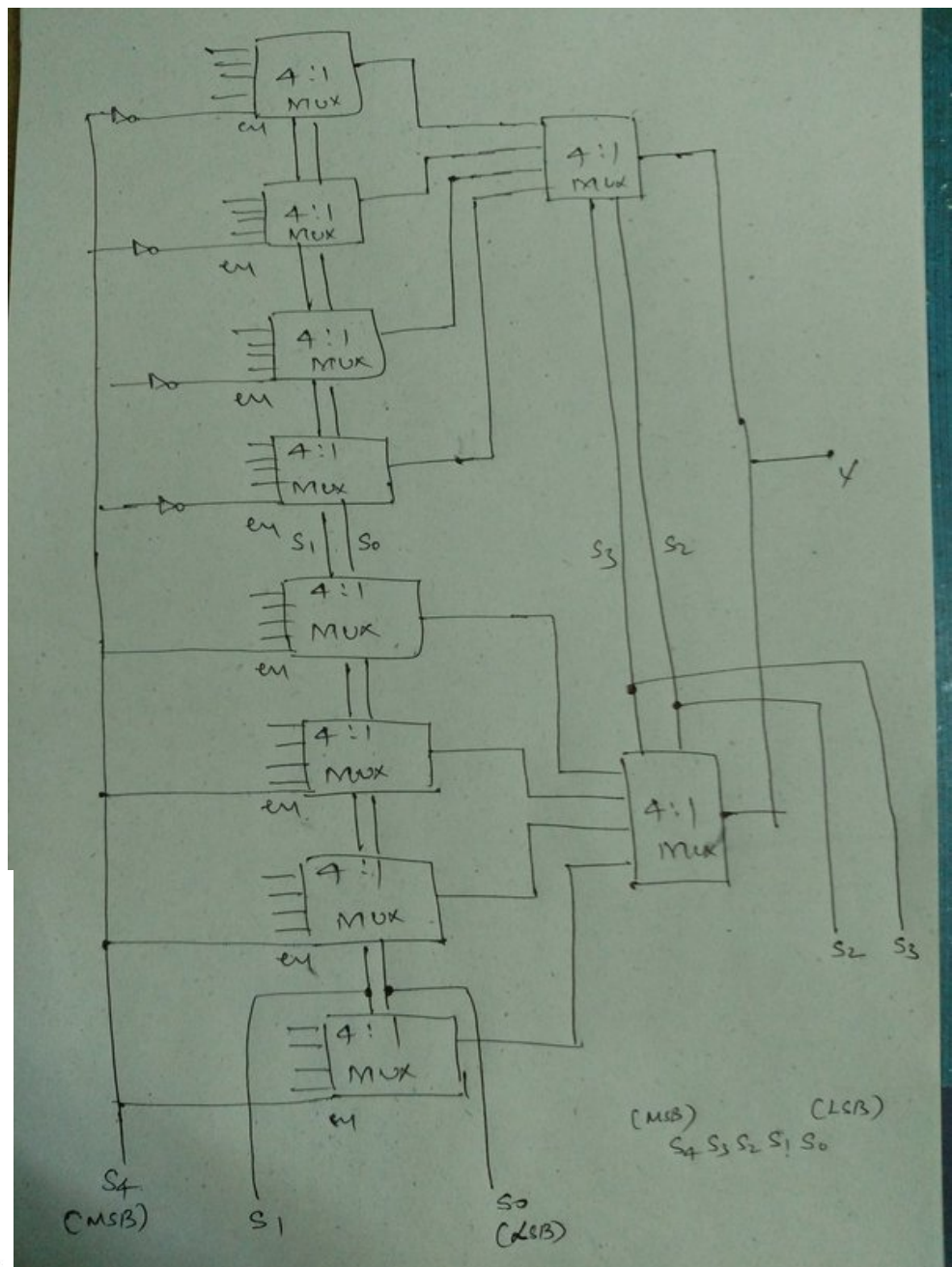
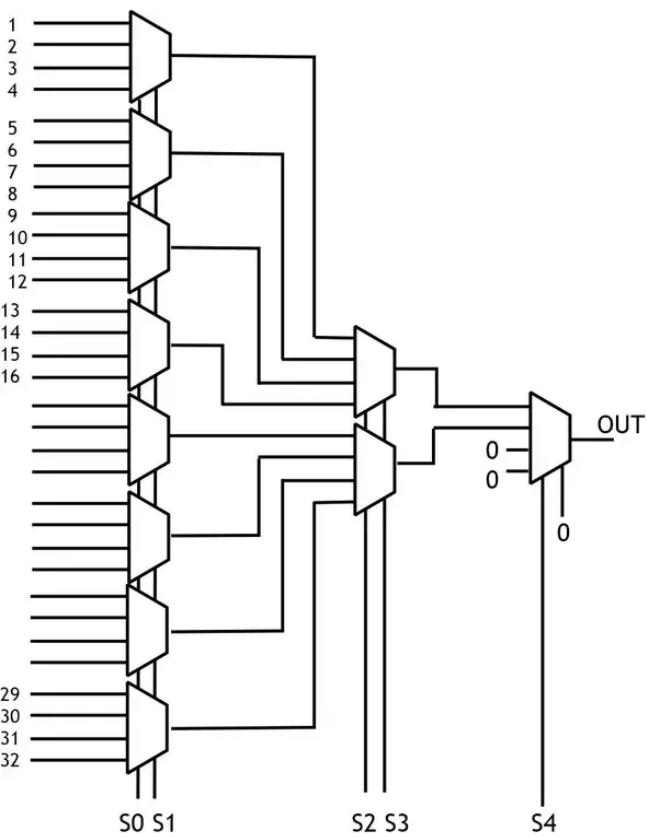
طراحی سوم: در این سوال میخواهیم یک مالتی پلکسر ۳۲ به ۱ با پهنای بیت ورودی ۱۶ بیت طراحی کنیم:

(۱) یک مالتی پلکسر ۴ به ۱ با پهنای بیت ورودی یک بیت در سطح گیت طراحی کنید.

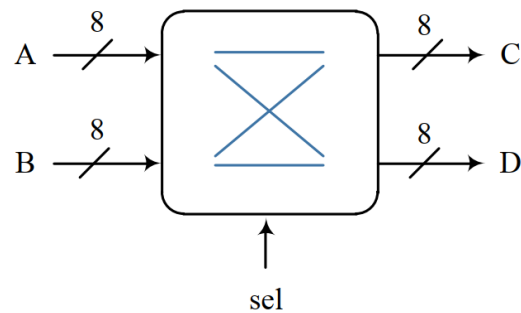
(۲) با استفاده از مالتی پلکسر های ۴ به ۱ پیاده سازی شده در قسمت ۱، یک مالتی پلکسر ۴ به ۱ با عرض بیت ورودی ۱۶ بیت طراحی کنید.

(۳) با استفاده از تعدادی مالتی پلکسر ۴ به ۱ با پهنای بیت ورودی ۱۶ بیت که در قسمت ۲ پیاده سازی کرده اید یک مالتی پلکسر ۳۲ به ۱ با پهنای ۱۶ بیت طراحی و پیاده سازی کنید.





طراحی چهارم: فرض کنید قرار است که مدار سوئیچی برای تنظیم خطوط ریلی چند ایستگاه راه آهن طراحی شود؛ به گونه‌ای که این خطوط ریلی اتصال بین دو ایستگاه مبدأ و دو ایستگاه مقصد را فراهم می‌کنند. برای طراحی چنین مداری فرض کنید که سیگنال انتخابی تحت عنوان sel تعیین کننده نحوه اتصال ایستگاه‌ها باشد که با پورت‌های ورودی و خروجی ۸ بیتی تعیین می‌شوند. به این ترتیب مطابق با شکل زیر اگر سیگنال تک بیتی sel برابر با صفر باشد، آنگاه خروجی C مدار برابر با A و خروجی D برابر با B خواهد بود. در غیر این صورت با ۱ بودن sel، خروجی C مقدار B و خروجی D مقدار A را خواهد داشت.



کد وریلاگی برای این مدار نوشته و عملکرد آن را با یک تست بنچ بررسی کنید.