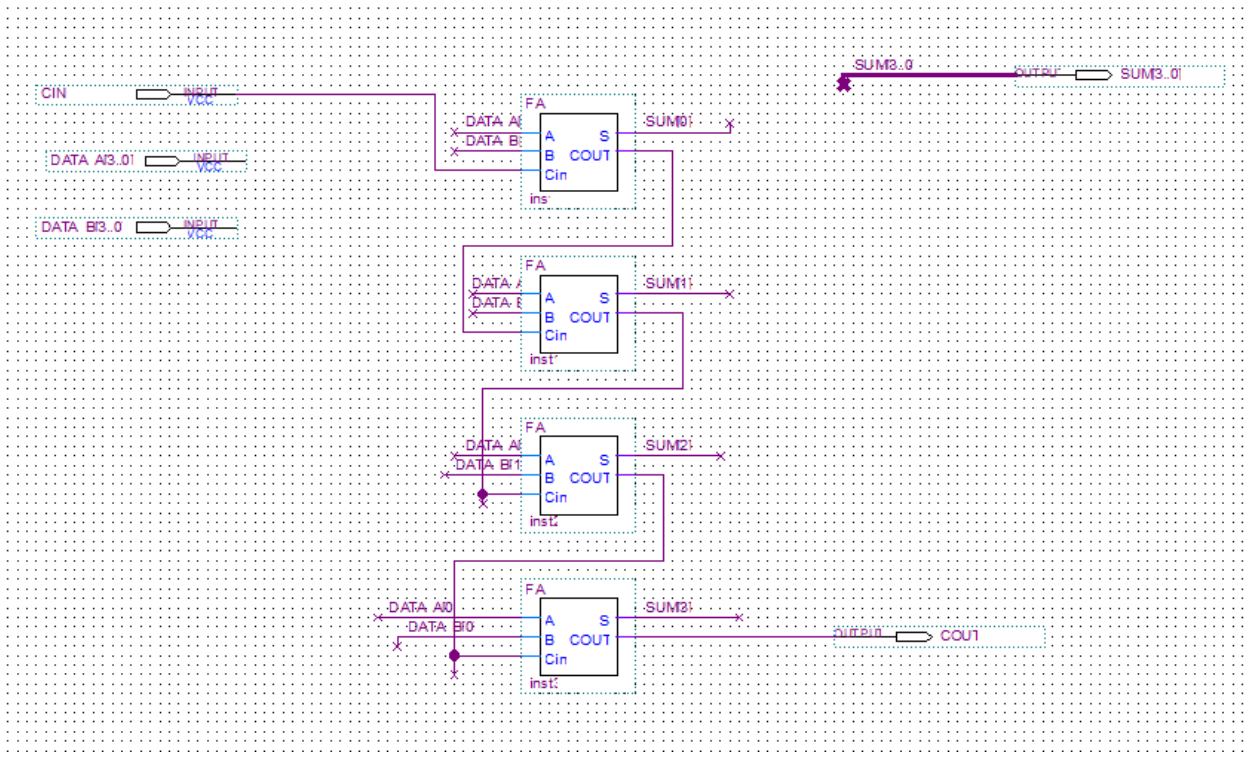
آریا همتی ۴۰۱۱۱۰۵۲۳ | فرزام کوهی ۴۰۱۱۰۶۴۰۳
تمرین عملی سوم معماری کامپیوتر، دکتر اسدی

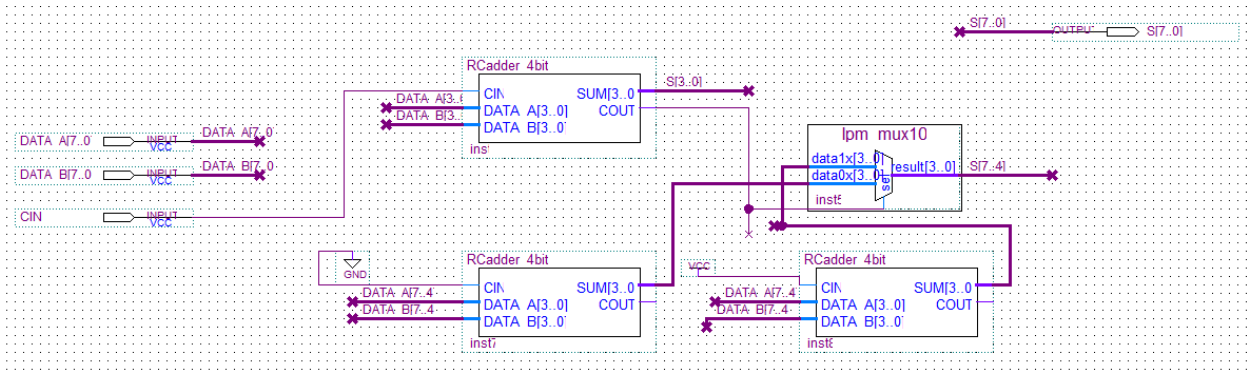
گزارش ساخت مدار:

برای ساختن این مدار ابتدا از یک Decoder، با 4 بیت آپکد به 16 حالت (عملیات) استفاده کردیم. ورودی آن یک OPCODE است و ۱۶ خروجی decoder نمایانگر تک تک operation ها است.
برای اجرای هر عملیات از یک بلوک استفاده شده، بلوک AND8bit و OR8bit و XOR8bit و SCAdder-8bit که در اصل یک سلکت کری ادر است، SUB_8bit، یک lpm_compare برای مقایسه دو عدد و اعلام جواب عملیاتهای Equal و N_Equal و یک ضرب کننده که با الگوریتم Booth ضرب را انجام میدهد به نام Booth_Multiplier.
بک سیگنال READY داریم که در عملیات‌های غیر ضرب که باید در یک سایکل انجام شوند، در یک سایکل، 1 می‌شود و در عملیات ضرب، هر 9 سایکل یکبار (عملیات ضرب طرق گفته‌های دستیاران آموزشی باید در چند سایکل صورت گیرد که طبق پیاده سازی ما، در 8 سایکل است.)
سه MUX داریم که برای تعیین مقادیر سیگنال‌های OUT, READY, ZERO بکار می‌رود.سیگنال ZERO تنها در دو عملیات Equal و N_Equal دستخوش تغییر می‌شود و عملیات هایی که مقدار 8 بیتی خروجی می‌دهند، سیگنال OUT را تغییر می‌دهند. سیگنال READY هم همانطور که گفته شد در دستورهای غیر ضرب در هر کلاک و در ضرب، در 9 کلاک یکبار تغییر می‌کند.

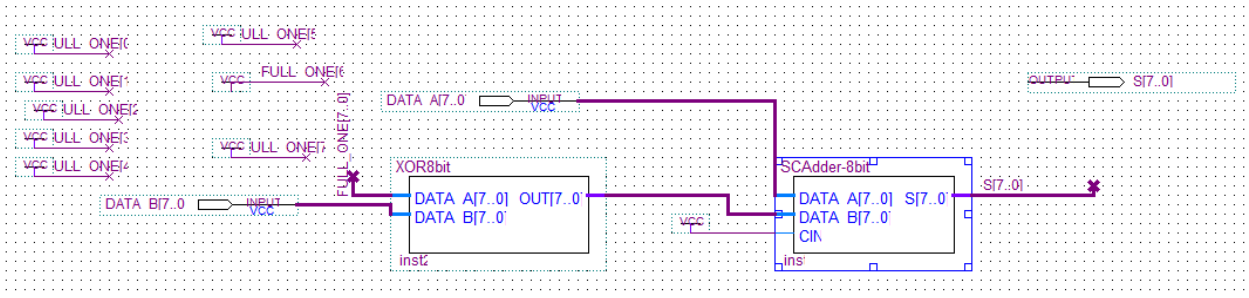
تنها دو بخش SCAdder و Booth_Multiplier نیاز به توضیح دارند و سایر بخش‌ها بدیهی هستند. SCAdder را به صورت دو بلوک 4 بیتی زدیم.
برای Booth_Multiplier، از یک Counter با 4 بیت استفاده کردیم که مراحل الگوریتم Booth را نشان می‌دهند. با مقایسه عدد counter با 0 میتوان فهمید در ابتدای ضرب هستیم یا خیر و با مقایسه آن با عدد 9 میتوان فهمید که آیا ضرب تمام شده یا خیر.
عدد حاصل از الگوریتم Booth را که با کنار هم قراردادن 8 بیت 0، Multiplier و یک بیت فرضی 0 در ابتدا initial شده را در یک رجیستر 17 بیتی ذخیره کردیم که مقدار آن را به همراه Multiplicand به یک بلوک سفارشی به نام Booth_Complex_Adder میدهیم که در اصل مغز متفکر اجرای الگوریتم Booth است و تصمیم گیری اینکه جمع انجام شود یا منها با توجه به دو رقم LSB این رجیستر 17 بیتی در این بلوک انجام می‌شود. در نهایت نیز این عدد با استفاده از یک Arithmetic Shifter، یک بیت به راست شیفت داده می‌شود.



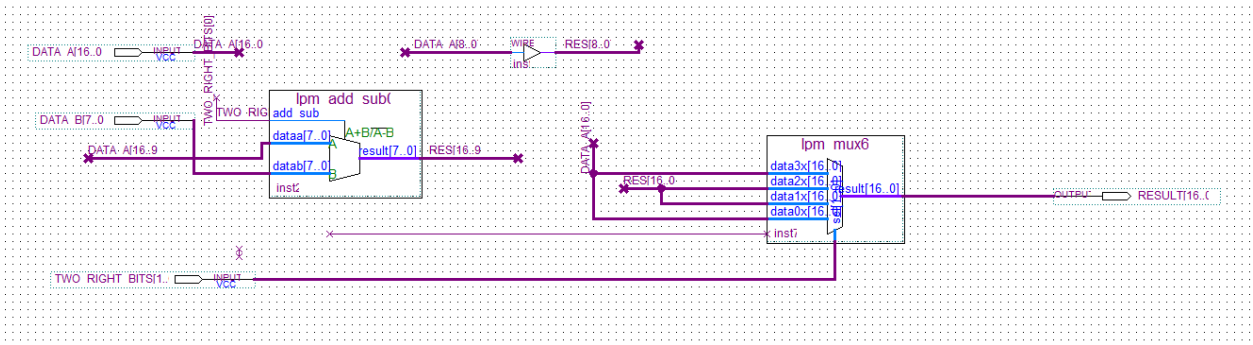
RCAdder_4bit



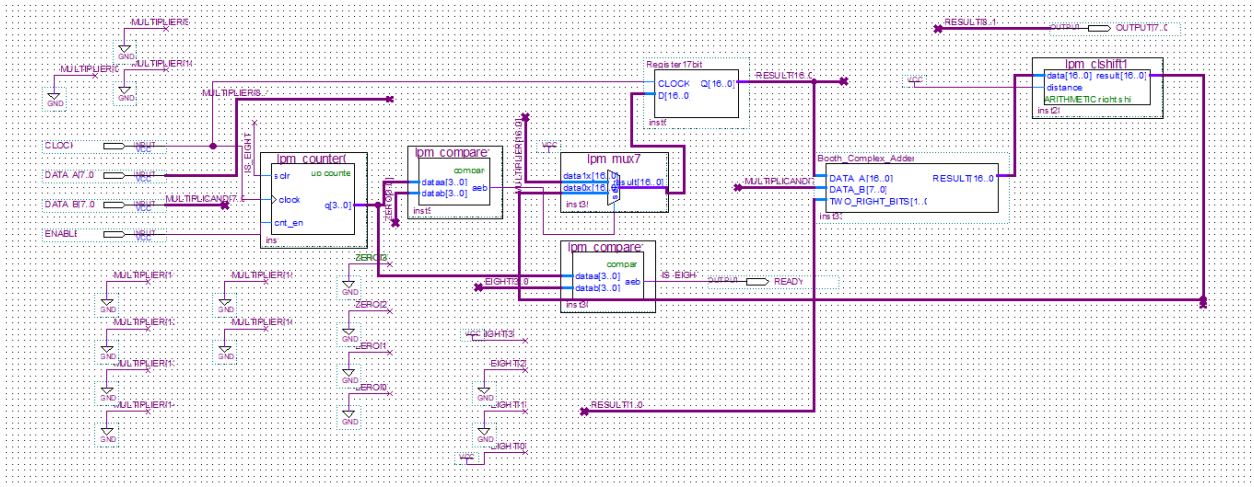
SCAdder-8bit



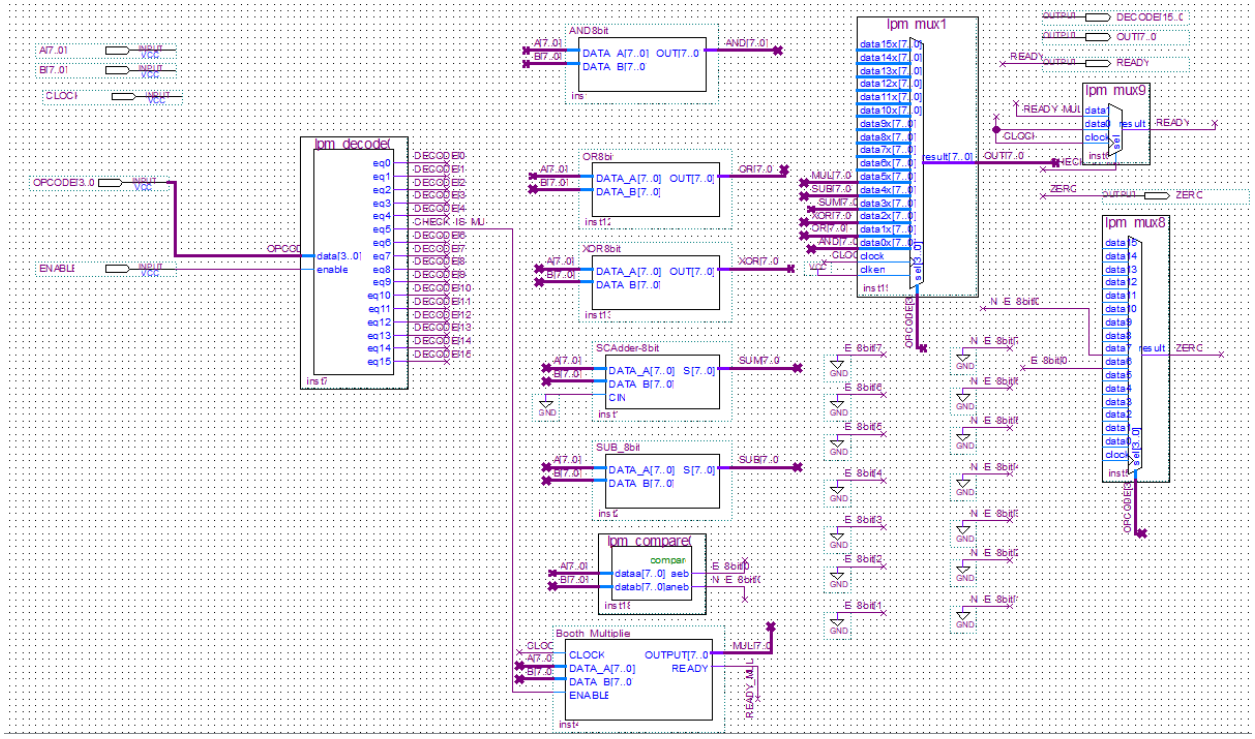
SUB_8bit



Booth_Complex_Adder: add or sub multiplicand from the left part of (current) result.



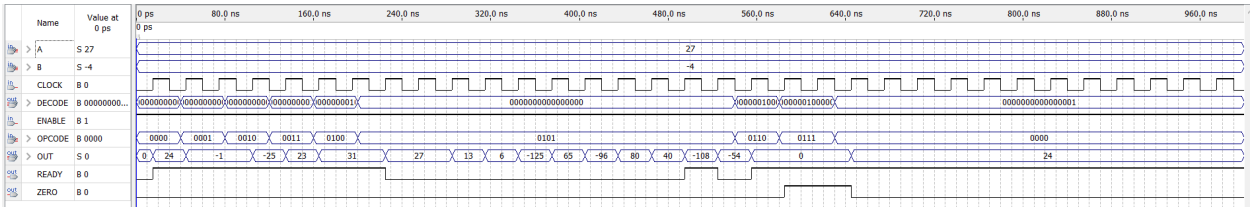
Booth_Multiplier: The part that uses “Booth_Complex_Adder” to do multiplication.



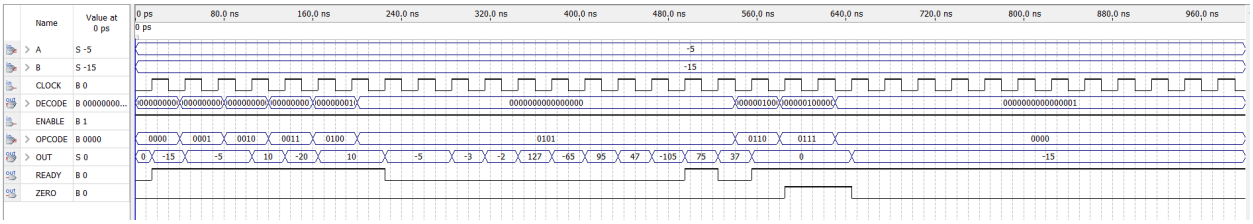
Practical3 - Main Circuit

شکل نهایی نیز، شکل نهایی مدار است.

تست مدار: در 3 تا عکس پیش رو، در هر عکس تمامی عملیات ها روی اعداد مشخصی تست شده اند که نتایج از روی عکس‌ها به وضوح قابل رویت است.

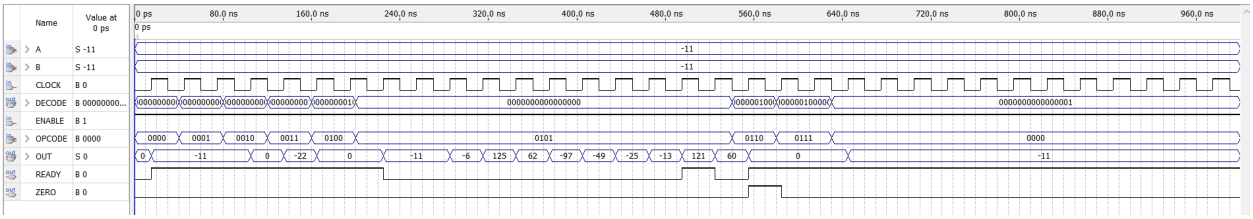


مقدارهای ورودی را برای مثال اول، 27 و 4- تعیین کردم. حاصل عملیات‌های MUL، SUB، ADD، XOR، OR، AND به ترتیب در سیگنال OUT واضح است (دقت شود که حاصل ضرب بعد از 9 کلاک می‌آید) که به ترتیب اعداد 24, -1, -25, 23, 31, 108- است. سیگنال READY نیز برای تمام دستورات تک سایکلی 1 است و برای دستور ضرب دقیقا زمانی که حاصل ضرب (108-) نمایان می‌شود، یک می‌شود. سیگنال ZERO نیز در دستورات Equal و N_Equal تحت تاثیر قرار می‌گیرد که چون این دو عدد نابرابرند فقط در N_Equal یک می‌شود.



مقادیر ورودی را برای این تست، 5- و 15- قرار دادم. حاصل‌های عملیات ها به ترتیب قابل رویت است:
حاصل ADD برابر 15-، حاصل OR برابر 5-، حاصل XOR برابر 10، حاصل ADD برابر 20-، حاصل SUB برابر 10، حاصل ضرب برابر 75.
حاصل دو عملیات Equal و N_Equal نیز طبق سیگنال ZERO، در حالت Equal برابر 0 و در حالت N_Equal برابر 1 است.

در تست آخر دو عدد برابر می‌دهم تا کارکرد Equal نیز تست شود.



مقادیر این تست به ترتیب 11- و 11- است. مقادیر خروجی OUT نیز برای عملیات‌ها مطابق روبه‌رو است: برای AND برابر 11-، برای OR برابر 11-، برای XOR برابر 0، برای ADD برابر 22-، برای SUB برابر 0، برای MUL برابر 121.
و در نهایت سیگنال ZERO اینبار در حالت Equal برابر 1 شده و در حالت N_Equal برابر 0 است. سیگنال READY نیز همانطور که گفته شد در دستورات تک سایکلی برابر 1 و در دستور ضرب نیز در سایکل 9 ام کلاک برابر یک می‌شود.