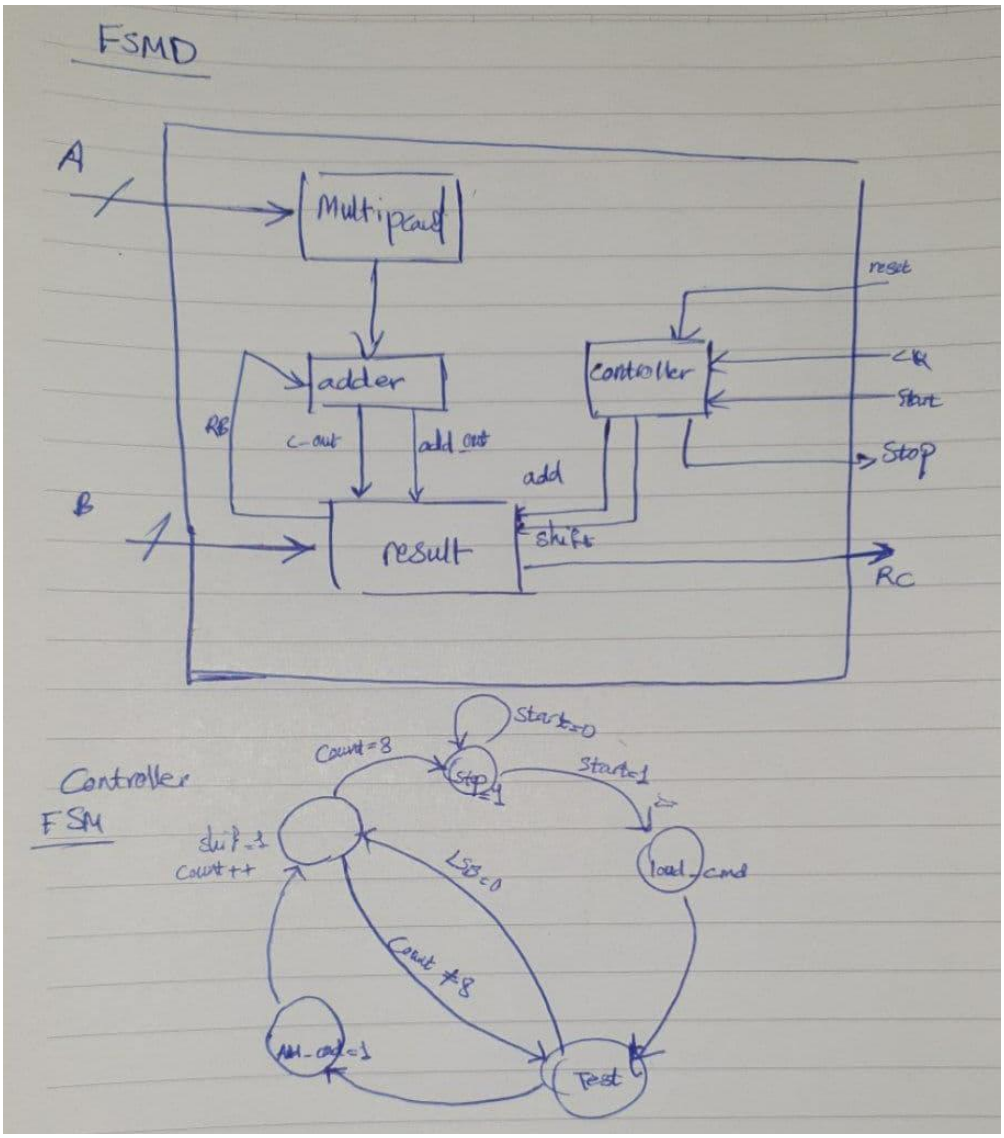
گزارش پیاده سازی تکلیف 4 هم طراحی سخت افزار و نرم افزار

غزل کلهری 97243058

عرفان مشیری 97243066

هدف از انجام این تمرین، طراحی یک ضرب کننده بر اساس عملیات شیفت و جمع برای ورودی های 6 بیتی بود.

ابتدا FSMD مرتبط با موارد خواسته شده را طراحی کردیم. در ادامه شمایی از طراحی انجام شده پیوست شده است.



در ادامه با استفاده از زبان GEZEL، کد مرتبط با FSMD مرحله قبل را پیاده سازی کردیم.

Datapath ضرب کننده ما شامل دو ورودی 6 بیتی بدون علامت x, y و دو بیت start, done برای شروع و پایان عملیات و در نهایت یک خروجی 12 بیتی بدون علامت است.

در ادامه از تعدادی رجیستر میانی برای ذخیره و شیفت دادن مقادیر میانی استفاده کردیم. سپس تعدادی sfg تعریف شده که از کنار هم قرار گرفتن آنها datapath ما تشکیل میشود. هریک از sfg های تعریف شده با توجه به فلوچارتی است که در ابتدای گزارش طراحی کردیم.

sfg init: مقداردهی اولیه به رجیستر ها و سیگنال های ما را بر عهده دارد.

sfg add: عملیات جمع نتیجه سابق ما با مقدار بدست آمده کنونی را نشان میدهد.

sfg shift: عملیات شیفت دادن حاصل بدست آمده را انجام میدهد. بدین ترتیب که در هربار اجرا، مقدار رقمی که در مرحله قبلی جمع شده و در آخرین بیت نتیجه ذخیره شده را تا زمان تمام شدن شمارشگر هربار یک رقم به سمت راست شیفت میدهد.

sfg IDLE: استیت اولیه که تا زمان آمدن سیگنال start، منتظر میماند تا عملیات را شروع کند.

sfg do_nothing: برای انتقال بهتر بین sfg ها.

always: همواره مقدار سیگنال product , Q را در کنار هم در خروجی قرار دهد تا در هر کلاک خروجی ما مقدار داشته باشد.

fsm mul_ctl: بخش کنترلر ما که با توجه به FSMD طراحی شده در مرحله اول، تعدادی استیت داریم که عملیات هر بخش از FSMD و انتقال به استیت های دیگر را با توجه به وضعیت یال هایی که دارد، تعیین میکند.

در نهایت برای برنامه خود تست بنچ نوشته و به ورودی های آن مقدار داده تا با نمایش خروجی از صحت اجرای آن مطمئن شویم.

خروجی های برنامه را با ورودی های مختلف در زیر مشاهده میکنید: