



درس معماری کامپیوتر
CA1 (Booth Multiplier)

اعضای گروه:

سینا طبسی 810199554

سید حامد میرامیرخانی 810199500

مسیر داده:

مسیر داده شامل دو رجیستر برای اعداد ورودی (X و Y)، یک ALU برای انجام عملیات جمع و یا تفریق، یک رجیستر برای ذخیره کردن جواب جمع و تفریق حاصل از ALU (A) و یک فلیپ فلاپ برای ذخیره کردن X0 تا بتوانیم به وسیله آن و همچنین به وسیله X1، نوع عملیات را مشخص نماییم.

در هر مرحله از ضرب، به وسیله دو بیت X0 (خروجی فلیپ فلاپ) و X1 (کم ارزش ترین بیت رجیستر X) نوع عملیات (جمع و یا تفریق) را مشخص می کنیم. سپس عملیات مورد نظر را بر روی رجیستر Y و رجیستر A در ALU انجام می دهیم و جواب را در رجیستر A ذخیره می کنیم. سپس رجیستر A و X را یکبار شیفت به راست کرده تا برای مرحله بعد آماده باشد. خروجی ما برابر رجیستر A (5 بیت با ارزش) و رجیستر X (5 بیت کم ارزش) می باشد.

کنترلر:

در ابتدا در استیت S0 (Idle) بوده و سیگنال های init_cnt و Done را برابر یک قرار می دهیم. با دیدن سیگنال start شروع به کار می کند. در استیت بعدی ورودی را در رجیستر X و Y قرار می دهیم (Id_X و Id_Y) و init_A و init_ff را برابر یک قرار می دهیم تا رجیستر A و فلیپ فلاپ dff مقدار اولیه صفر را بگیرند. بعد از آن با استفاده از دو بیت X0 و X1 نوع عملیات را برای ALU مشخص می کنیم که یا add و یا sub را انجام بدهد. سپس با سیگنال Id_A مقدار محاسبه شده در ALU را در رجیستر A قرار می دهیم و در دو استیت بعدی shift_a و shift_x را به ترتیب فعال می کنیم تا رجیستر A و X شیفت به راست دهیم. در استیت بعدی C-up را کی مرتبه افزایش می دهیم. حال اگر counter ما برابر 5 و یا بیشتر شده بود، به آن معنی است که عملیات ضرب ما تمام شده و به استیت S0 می رویم و سیگنال Done و init_cnt را برابر یک قرار می دهیم (به این معنی است که عملیات تمام شده و منتظر ورودی جدید خواهیم بود). اگر مقدار counter کمتر از 5 بود، به استیت S5 می رویم تا برای ادامه عملیات ضرب تصمیم گیری کنیم.

ماژول ها:

Counter: در قسمت کنترلر ما قرار دارد تا به استفاده از آن حساب تعداد عملیات های جمع و

تفریق در هر بار ضرب کردن را بدانیم.

dff: فلیپ فلاپ مورد نظر ما می باشد.

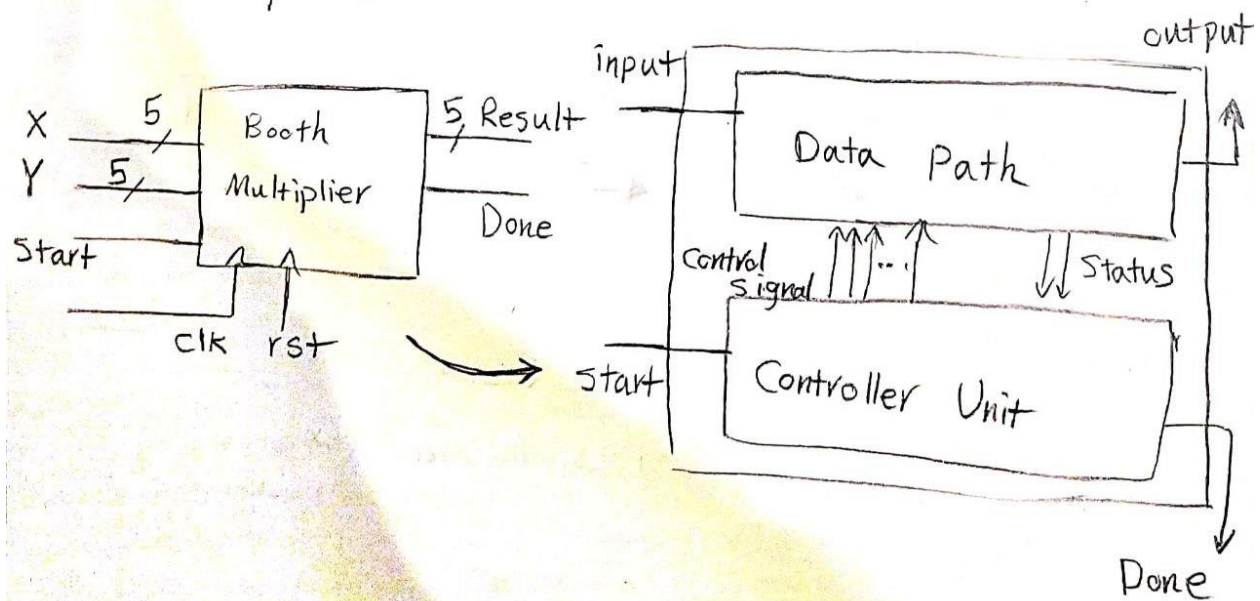
ALU: عملیات جمع و تفریق را انجام می دهد.

Y_reg: رجیستر Y می باشد.

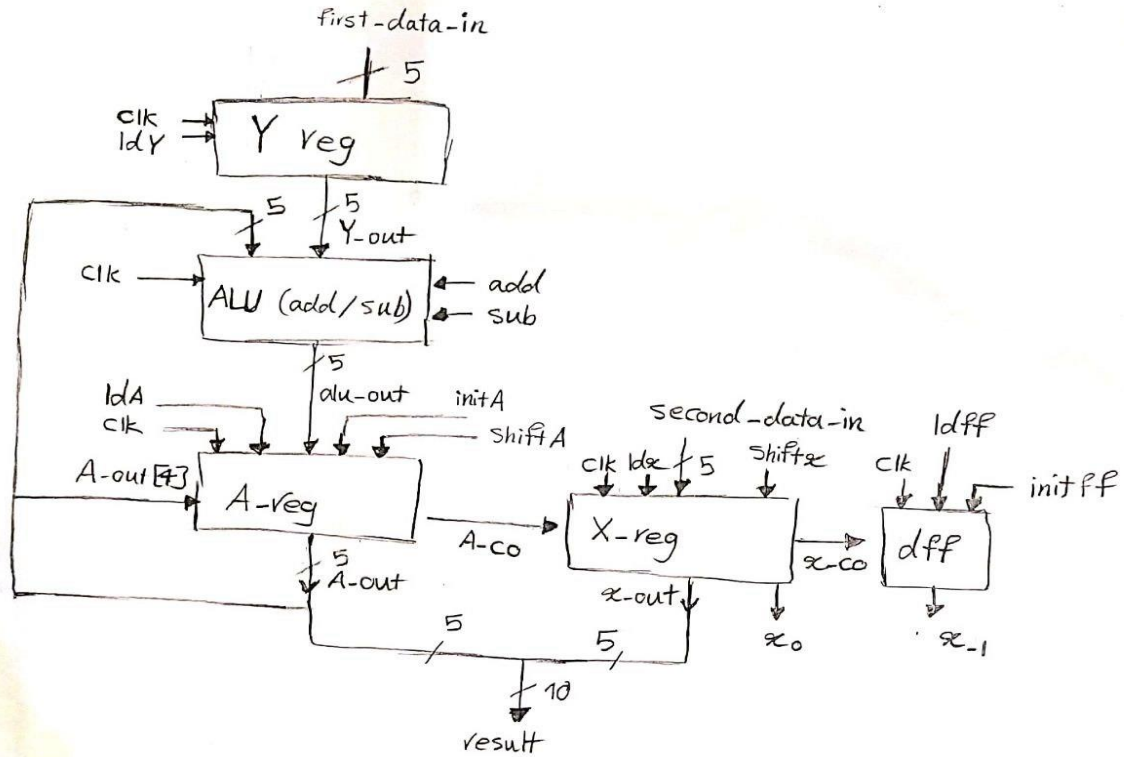
Shift_reg: رجیستر A و X از این نوع می باشد. زیرا این دو رجیستر باید قابلیت شیفت به راست را داشته باشند.

Multiplier: تاپ ماژول ما بوده و دارای دو ماژول **datapath** و **controller** می باشد.

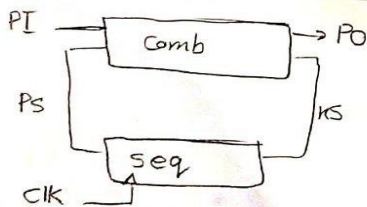
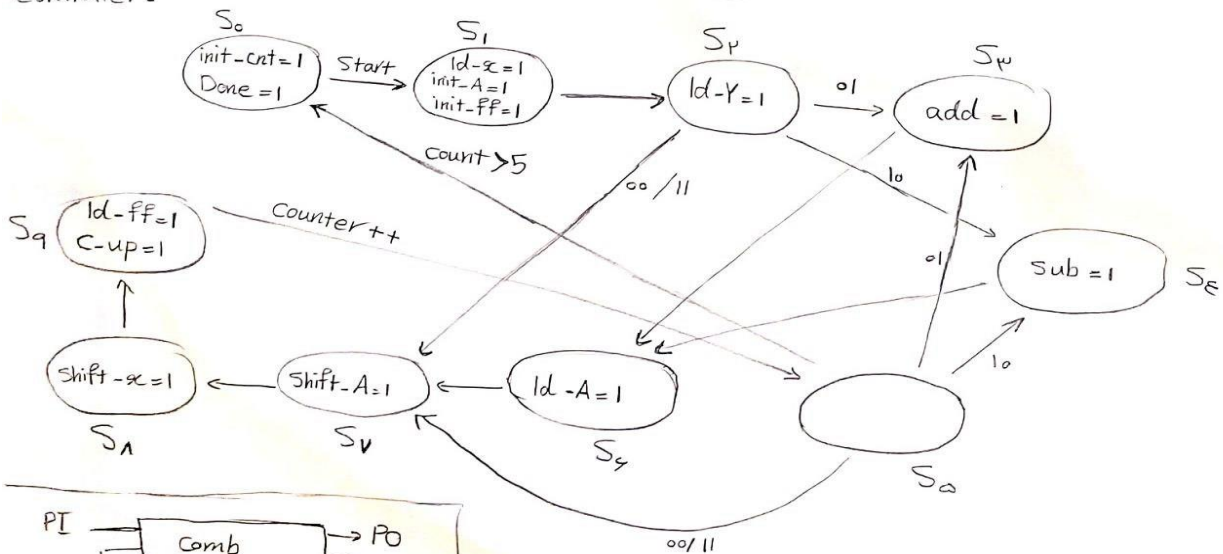
Booth Multiplier :



Data Path:

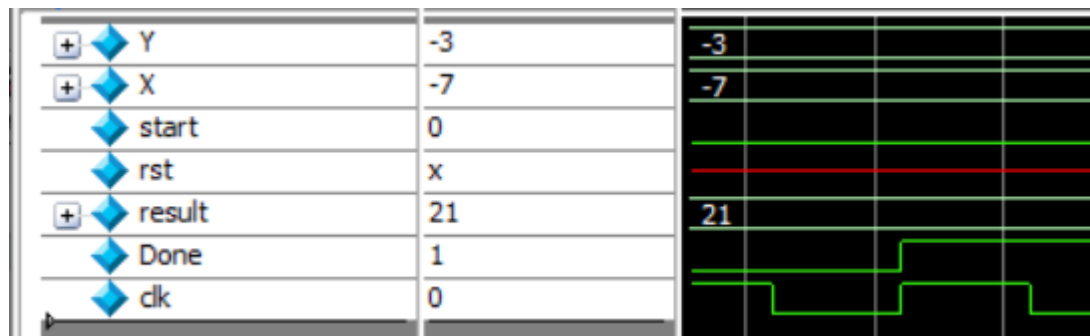


Controller:

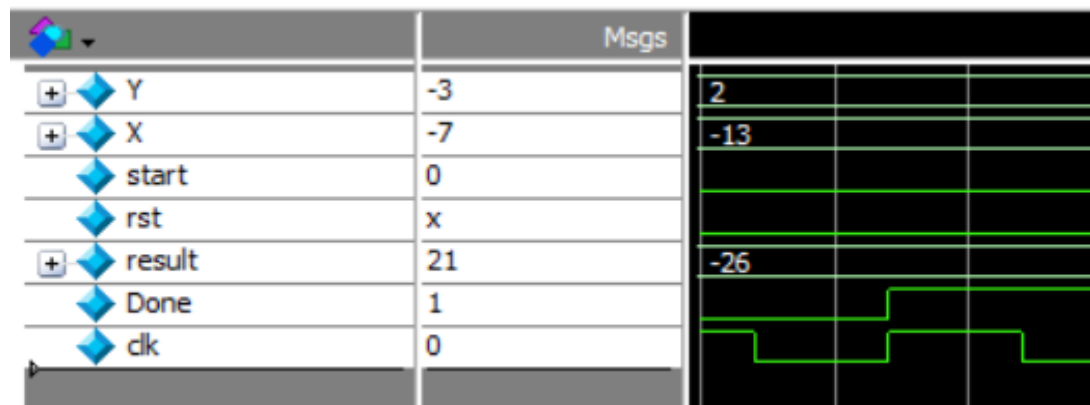


نتایج برای 5 عدد باینری تصادفی:

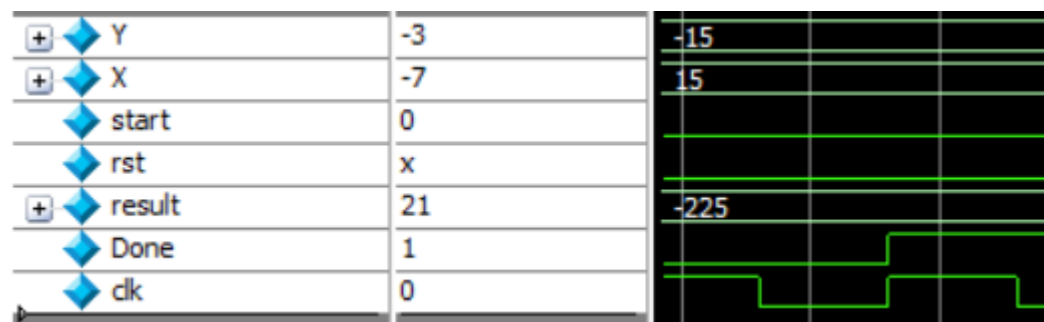
تست اول:



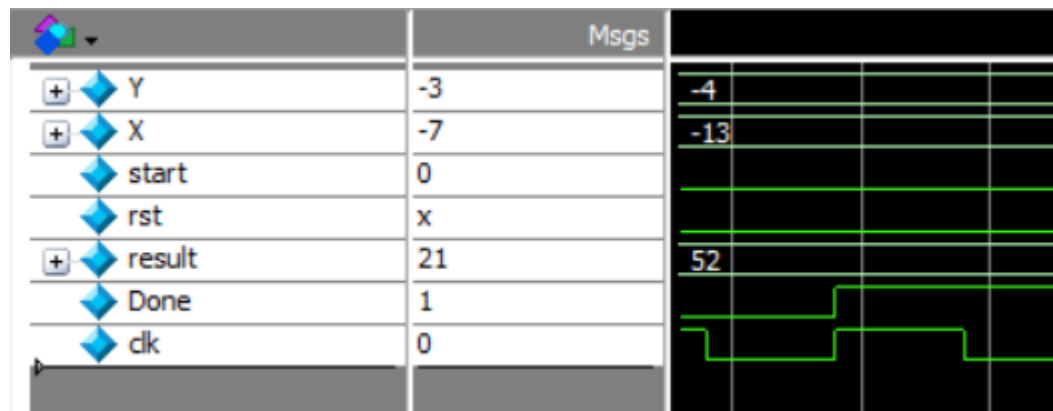
تست دوم:



تست سوم:



تست چهارم:



تست پنجم:

