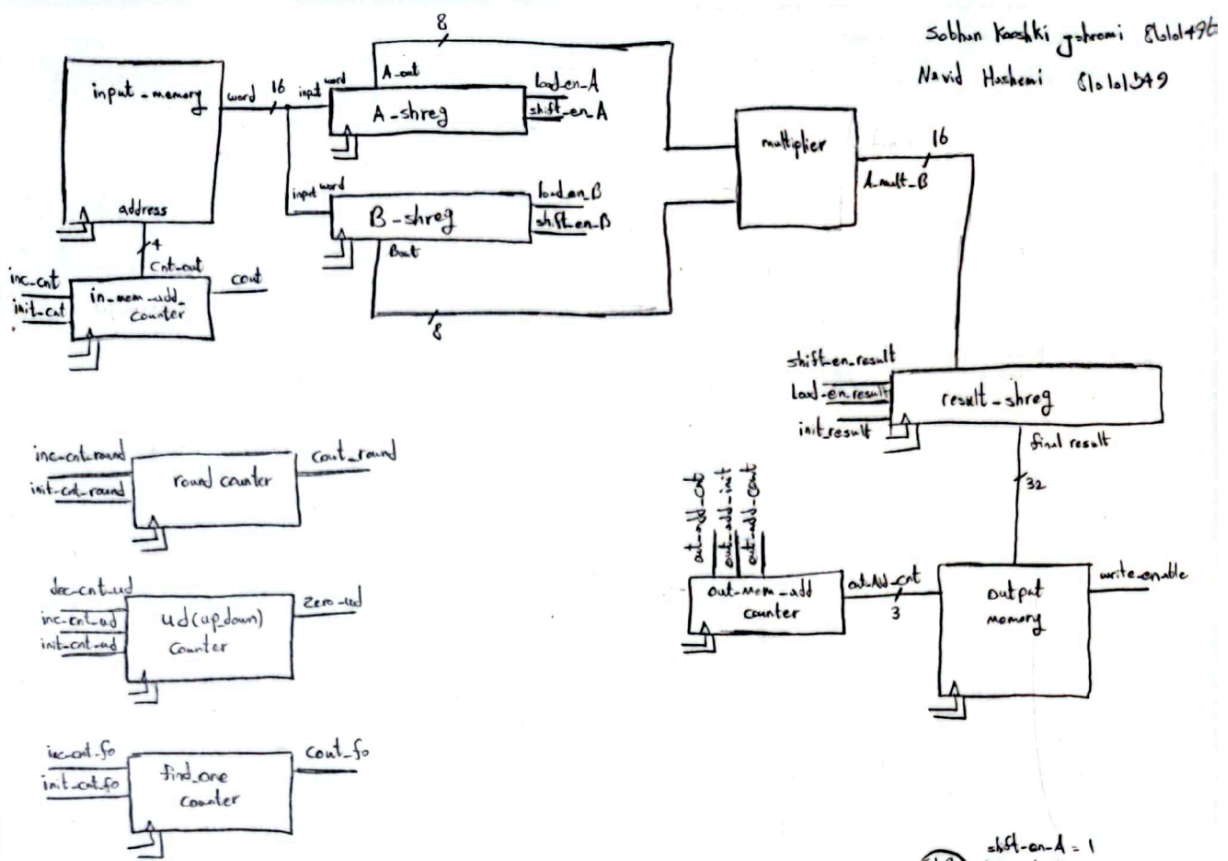


CAD (Computer Aided Design)

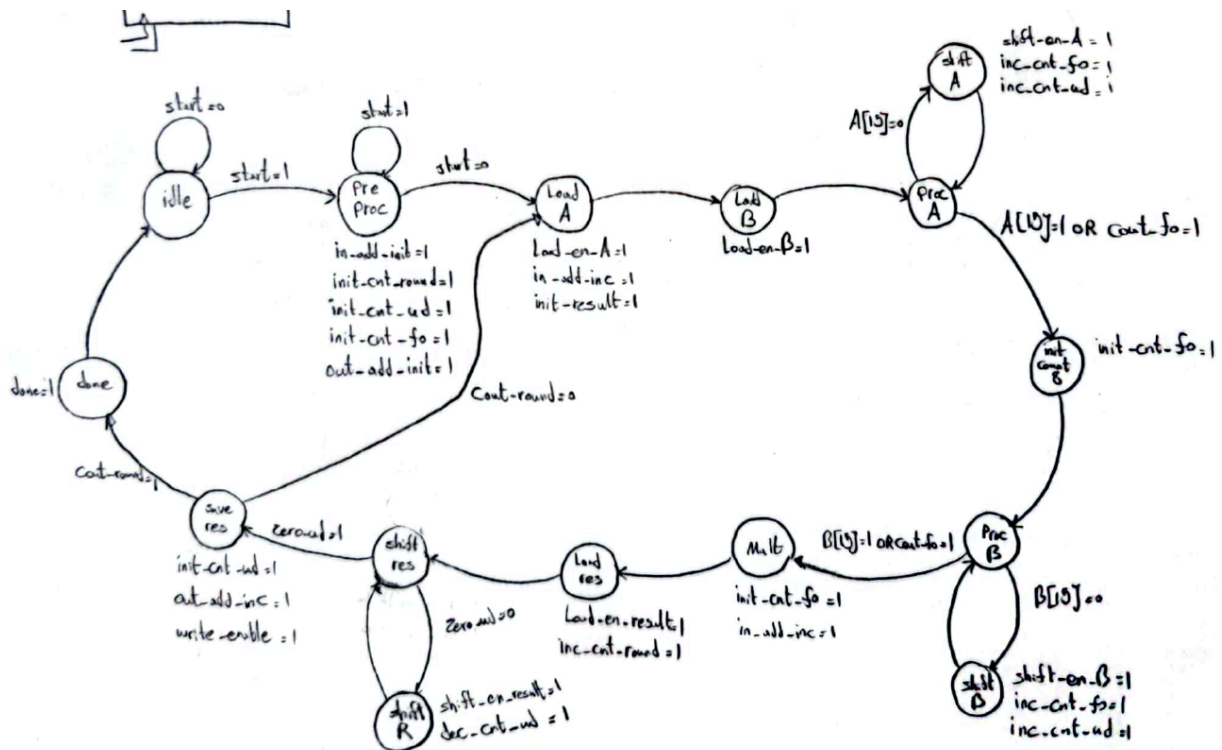
CA1

Sobhan Kooshki Jahromi 810101496
Seyed Navid Hashem 810101549

Datapath Design:



Controller Design:



Report & Check:

در این پروژه L به دنبال طراحی یک سیستم ضرب کننده تقریبی بودیم که می‌توانیم خاص و نسبتاً ساده ای دارد از طریق یک Bus ۱۶ بیتی در ورودی خوانده شد طبق الگوریتم ۸ بیت از هر کلام از این ورودی ما به ورودی ضرب ورودی و بعد از ضرب این دو عدد ۸ بیتی، ۱۶ بیت جواب تا تعدادی که لاگت می‌شود و در خروجی مقصد نوشته می‌شود

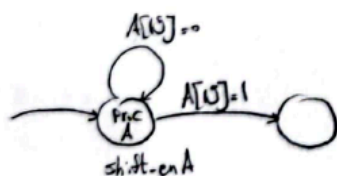
مرطه ۱ سے فائن دوری ما : ما در ابتدا کریک کریک که memory ما می توانیم دو خطه میزان را از آن بخوانیم همین دلیل ۲ خطه ما گذشتہ بودیم برای رجیسترهای A و B و بی باقی به عنوان TA ما می توانیم که معوی ما می تواند در هر لحظه ۱ دوری را برای ما بخواند بی آن ۲ بی را بین هم share کریک و در ۲ لاک سیکل از معوی اطلاعات را می خوانیم



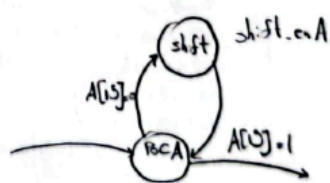
مرطه ۲ ← ابتدا کریک ۸ بیت موثر ضرب : ما رجیسترهای A و B را به صورت یک register می توانیم که بخوانیم با استفاده از اطلاعات جابجاء ۸ بیت موثر را به دست می آوریم اگر بیت ملی آنز A یا B ۰، ۰ بود ما رجیسترها را یک واحد شیت می دیم و اگر بیت آنرا بود یعنی به ۸ بیت رسیدیم و این مرطه را با موفقیت پشت سر گذاشتیم

یک نکته اینجا وجود دات دات این بود که اگر بعد از ۸ بار انجام به تکلیف باز هم ۱ را در بیت آنز A یا B نمی بینیم این ۸ بیت با این را در نظر می گیریم که در این صورت ضرب ما دیگر ضرب کامل است و تغییری نیست این شکل را ما با یک counter ۸ بیتی درست کریک و اگر ۸ بیت، برابر شد دیگر شیت نمی دیم و عملیات ما به پایان رسیده است.

یک نکته دیگر این بود که در اینجا ما این صورت تکلیف را انجام می دیم پس به دلیل اینکه بعد از دستور له معامی نباید در ۸ بیت ملی آن تغییریم این روش بکل دات و بعد از آنکه ۱ را در ۸ بیت تکلیف را انجام می ده و این شکل را با اضافه کردن یک عملیات



این روش خط است



این درست است

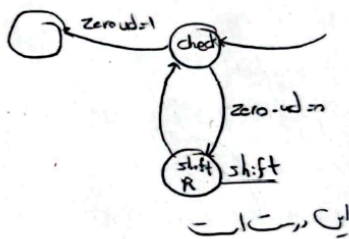
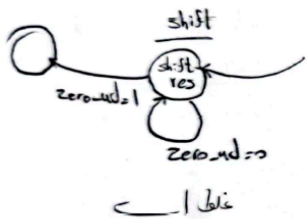
مرطه سوم ← ضرب کس : با استفاده از مرطه قبل ما می تونیم ۸ بیت باقی به تکلیف و عمل ضرب ما شده و آن ما را از ۲۴ بیکریک و ضرب می کریک

مرطه چهارم ← save کردن ضرب و تکلیف ملی ۰ : ما برای اینکه بتوانیم تکلیف را هم به دوری بگیریم از یک ۲۴ بیت shift register درست لان جواب ۳۳ بیتی آماده کریک اما این رجیستر را با ۰ initial کریک (در آلوی که گفته شد که بعد از یک کریک ۸ بیت موثر به تعداد رقم ملی که بعد از این ۸ بیت در ۸ و ۵ آمده است به دست راست فوپی ضرب کننده اضافه کنید این initial با ضربین کار را برای ما انجام می ده و دیگر نیازی نیست که آن رقم ما را از تعدادشان را بشماریم و در کنار جواب آن ما را بخوانیم) پس ۱۶ بیت فوپی ضرب کننده را در ۱۶ بیت بالا له می کریک.

حال بیت shift کردن ۰ ما به ست بیت رجیستر نوی است . در مرحله ای که دیتیر A و B را آنالیزی کنیم ما با استفاده از یک counter تعداد هابی را که بیت shift کنیم و از رجیسترهای A و B خروجی می گیریم و یک counter تعریف می کنیم

این تازمان ۰ شدن این هابیت ما از ست بیت به دون shift reg نوی ۰ اخته می گرد این هم بخشی نویی الگوریتم ضرب تقریبی بود

پس از پایان بیت کردن الان ما جواب یک ضرب ۱۶ و ۱۶ تقریبی را داریم فقط باید آن را در همی نویی دیتیر کنیم یک نکته دیگر هم مانده که مربوط به رجیسترهای A و B این است که دیتیر در انجام باید نویی shift ۰ ما به دون رجیستر نویی از ۲ بیت استفاده کنیم و گرنه یک بیت اخته به دون رجیستر shift نویی



Check Final Answer:

```

1110111000101110
0011011101110001
0011110001010011
1100010111111000
0010011010101111
1100001110000010
1000100111101000
0100001001111010
0000111101001100
0000010010001000
1101011001100101
0010100100100000
1010111001000101
0100100000100111
1100000101100100
1000000011100101

```

Input Memory

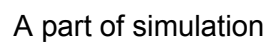


```

335d8000
2e5d4000
1d538000
23520000
00451a00
22460000
30f00000
60800000

```

Output Memory



A part of simulation