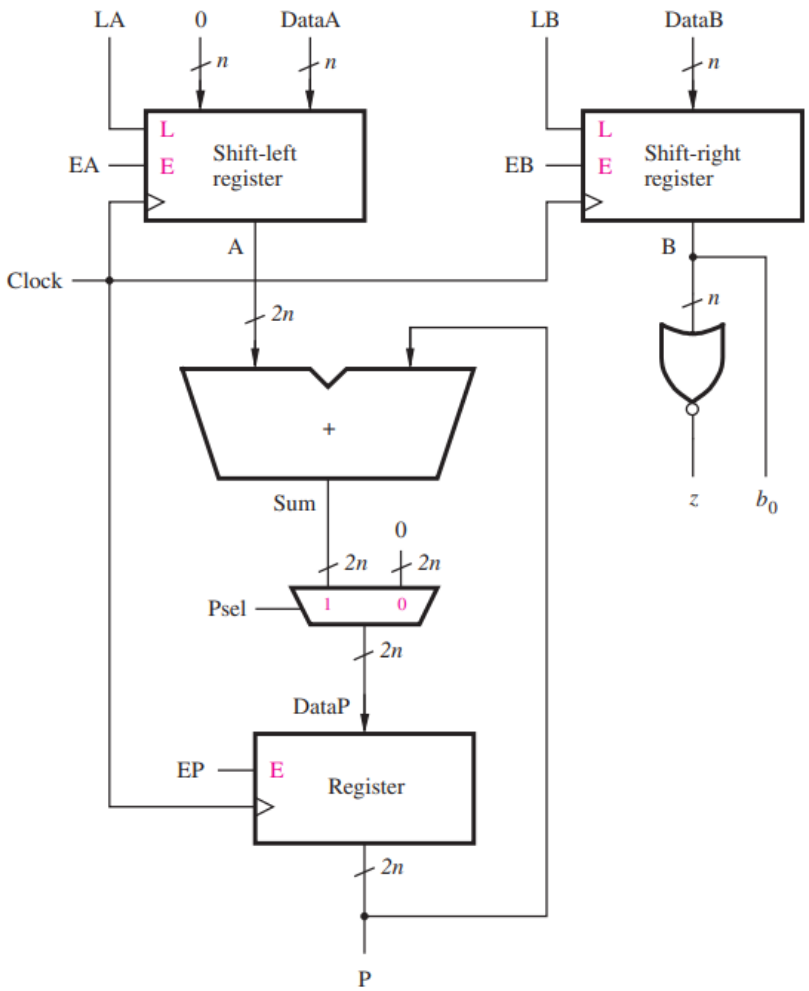
به نام خدا

پروژه کامپیوتری پنجم درس مدار منطقی ، استاد علیزاده

محمدشهاب شرافت ۸۱۰۱۰۲۴۶۷

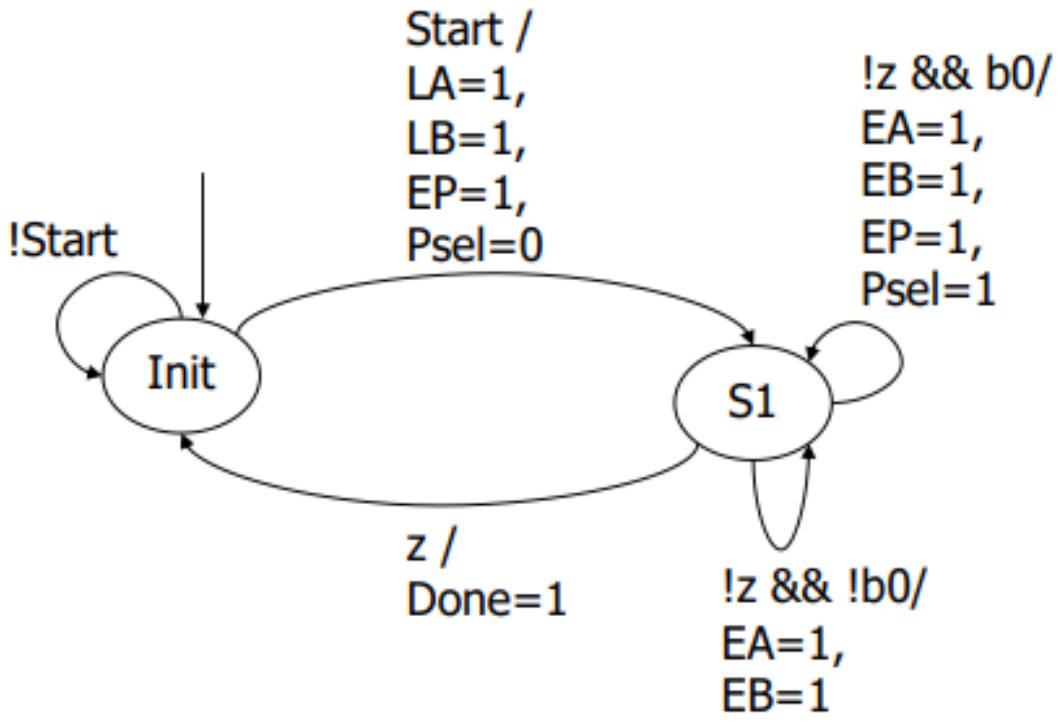
سوال ۱

برای این سوال ابتدا **Datapath** و سپس **FSM** را طراحی کردم. **Datapath** بدین شکل است:



این طراحی دو عدد **A** و **B** را میگیرد و **A** را به چپ و **B** را به راست شیفت میدهد و اگر بیت حذف شده ی **B** ۱ بود نتیجه را با مقدار فعلی **A** جمع میکند.

سپس Controller:



پس از طراحی تست کیس های موجود در تست بنچ را تست کردم و نتیجه رضایت بخش بود.

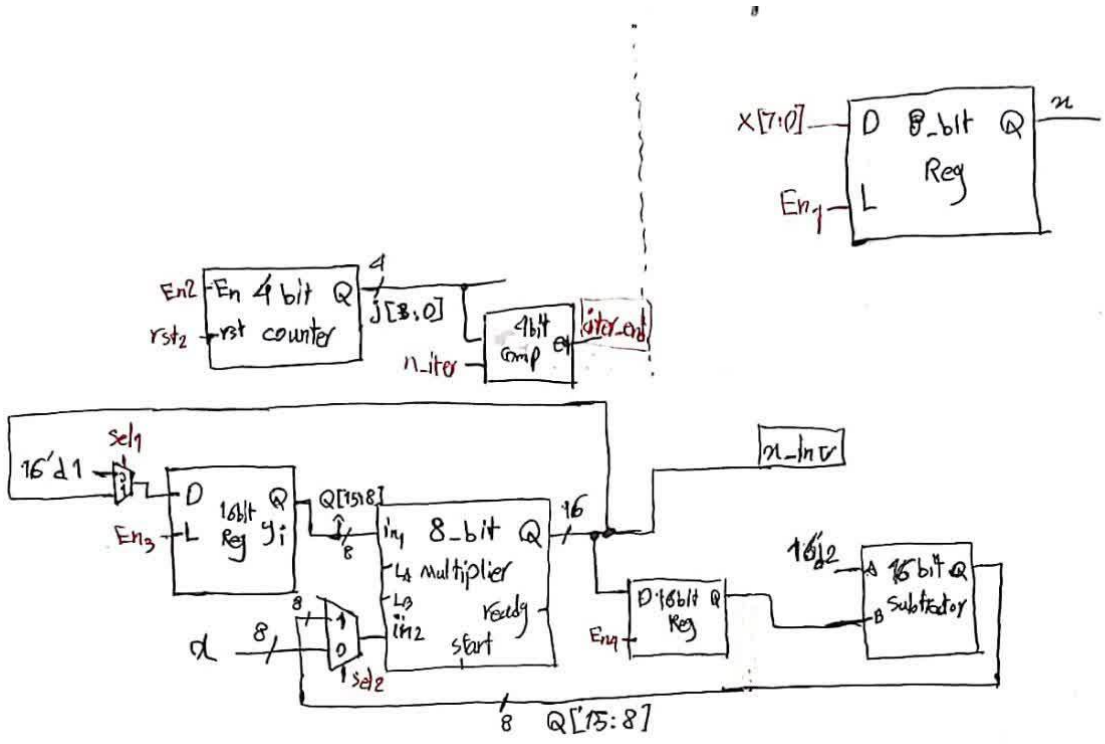
پاسخ سوال ۴: تنها تفاوتی که میکند این است که تعداد اعشار های عدد اول با عدد دوم جمع میشود و برابر است با تعداد اعشار های نتیجه. یعنی اگر مثلا عدد اول ۲ بیت اعشار داشت و عدد دوم ۱۴ بیت اعشار داشت، نتیجه را بدون احتساب اعشار ها حساب میکنیم و درنهایت روی نتیجه مان ۱۶ بیت اعشار میگذاریم.

برای این بخش هم درست بودن بخش قبل درستی این بخش را نتیجه میدهد.

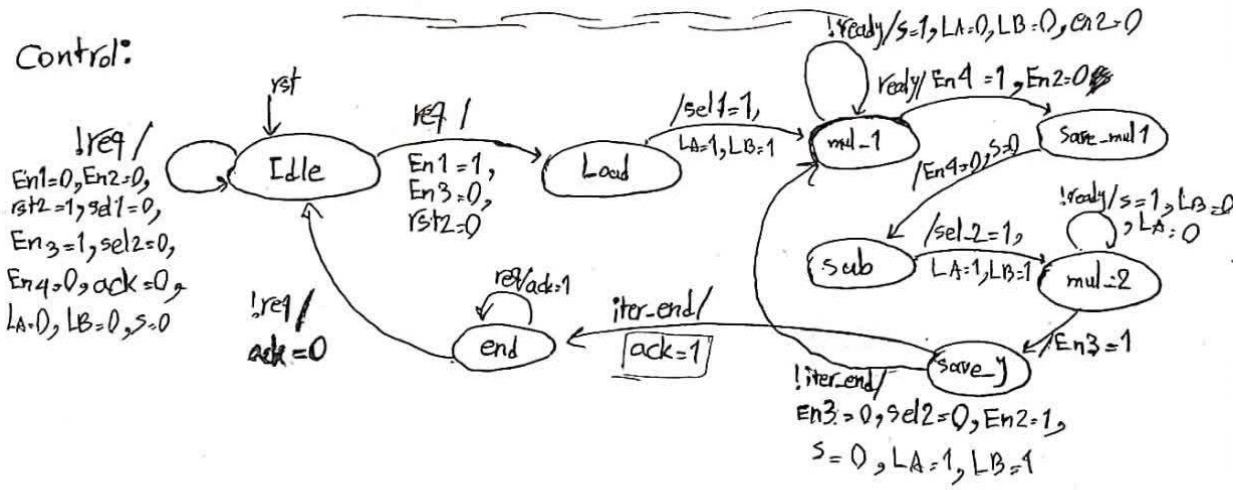
سوال ۲

برای این بخش ابتدا **x** را ذخیره کردم، سپس با استفاده از یک شمارنده به تعداد **n_iter** شمردم و تقریب زدم.

datapath به شکل زیر:



و **Controller** به این شکل شدند:



در این سوال، مدیریت اعشار اعداد خیلی سخت بود. برای بخش صحیح اعداد یک بیت و برای اعشارها ۲ بیت قرار دادم.

طراحی ام را برای چند ورودی آزمایش کردم و تا ۸ رقم اعشار درست کار میکرد ولی به دلیل اینکه برای هر بار اجرای کد نیاز به چندهزار نانوثانیه زمان بود و تایمینگ دیاگرام شبیه سازی به شدت بزرگ میشد، از آوردن نتیجه در گزارش امتناع ورزیدم.

سوال ۳

هدف از این سوال سنتز مدار طراحی شده در سوال ۲ بود.

سوال ۱) کوارتس را نصب کرده و پروژه را ساخته و کامپایل کردم. چند ارور مربوط به کامپایل وجود داشت که رفعشان کردم.

سوال ۲) در این بخش نکته این بود که باید دیفالت پارامتر های موجود در طراحی را تغییر میدادم، هم در فایل اصلی، هم در **datapath** و هم در ماژول **multiplier**.

n = 8

Flow Summary	
Flow Status	Successful - Fri Jan 31 00:00:21 2025
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	Reciprocal
Top-level Entity Name	Reciprocal
Family	Cyclone II
Device	EP2C70F896I8
Timing Models	Final
Total logic elements	149
Total combinational functions	127
Dedicated logic registers	87
Total registers	87
Total pins	32
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

1	226.86 MHz	226.86 MHz	clk

در این حالت، تعداد رجیستر ها برابر با ۸۷ و فرکانس کلاک برابر با ۲۲۶ مگاهرتز است. در مدار **datapath**، دوتا رجیستر ۱۶ بیتی، یک رجیستر ۸ بیتی، یک شمارنده ۴ بیتی و یک ضرب کننده که متشکل از یک دیتاپث با شیفت رجیستر ۱۶ بیتی، یک شیفت رجیستر ۸ بیتی و یک رجیستر ۱۶ بیتی و یک کنترلر با سه تا استیت (که یعنی ۳ فلیپ فلاپ دارد) است، وجود دارند. مجموع تعداد **d-flip-flop** هایی که گفتم این مدار برابر است با ۸۷ که دقیقا برابر با چیزی است که نرم افزار گفت.

n = 16

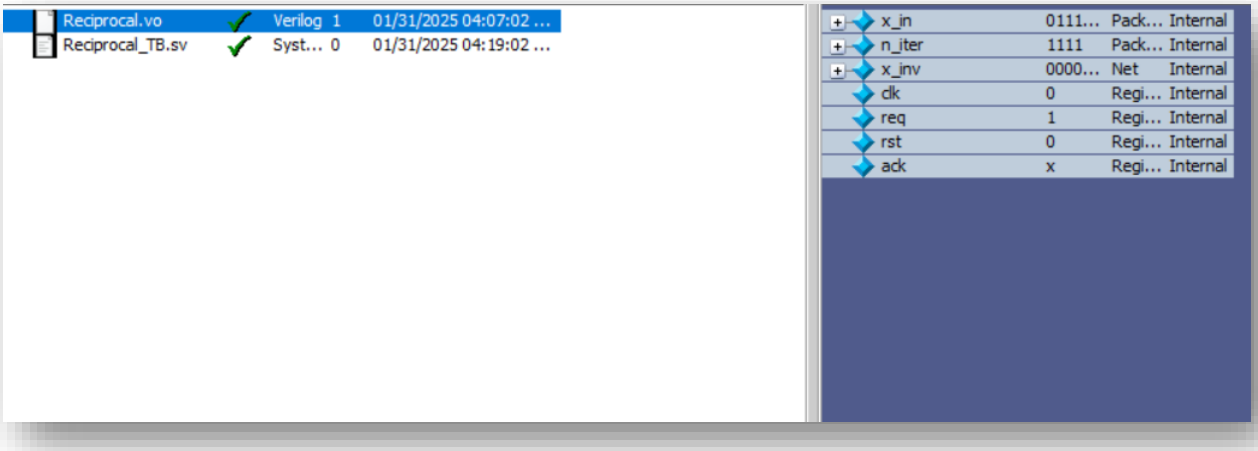
Flow Summary	
Flow Status	Successful - Fri Jan 31 01:24:05 2025
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	Reciprocal
Top-level Entity Name	Reciprocal
Family	Cyclone II
Device	EP2C70F896I8
Timing Models	Final
Total logic elements	224 / 68,416 (< 1 %)
Total combinational functions	201 / 68,416 (< 1 %)
Dedicated logic registers	167 / 68,416 (< 1 %)
Total registers	167
Total pins	56 / 622 (9 %)
Total virtual pins	0
Total memory bits	0 / 1,152,000 (0 %)
Embedded Multiplier 9-bit elements	0 / 300 (0 %)
Total PLLs	0 / 4 (0 %)

1	166.25 MHz	166.25 MHz	clk	

n = 32

Flow Summary	
Flow Status	Successful - Fri Jan 31 01:25:27 2025
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	Reciprocal
Top-level Entity Name	Reciprocal
Family	Cyclone II
Device	EP2C70F896I8
Timing Models	Final
Total logic elements	386 / 68,416 (< 1 %)
Total combinational functions	351 / 68,416 (< 1 %)
Dedicated logic registers	327 / 68,416 (< 1 %)
Total registers	327
Total pins	104 / 622 (17 %)
Total virtual pins	0
Total memory bits	0 / 1,152,000 (0 %)
Embedded Multiplier 9-bit elements	0 / 300 (0 %)
Total PLLs	0 / 4 (0 %)

2	111.99 MHz	111.99 MHz	clk

هرچه مقدار n بالا میرود، زمان مورد نیاز برای المان های داخل مدار هم زیادتر میشود و هرچه زمان زیادتر میشود فرکانس که عکس زمان است هم کمتر میشود.

سوال ۳) برای این سوال کلی وقت گذاشتم! و فایل های **vo.** و **sdo.** را به پوشه ی **simulated_codes** منتقل کرده و پروژه جدیدی در **modelsim** ساختم. برای تست بنچ هم از همان تست بنچ قبلی استفاده کردم. سپس کد را شبیه سازی کرده کردم. ولی نتیجه شباهتی به شبیه سازی کد خودم نداشت و درست کار نکرد.