

گزارش کار فاز اول پروژه اول درس طراحی سیستم‌های دیجیتال

پریا پاسه ورز 810101393

آریان رجبی 810199421

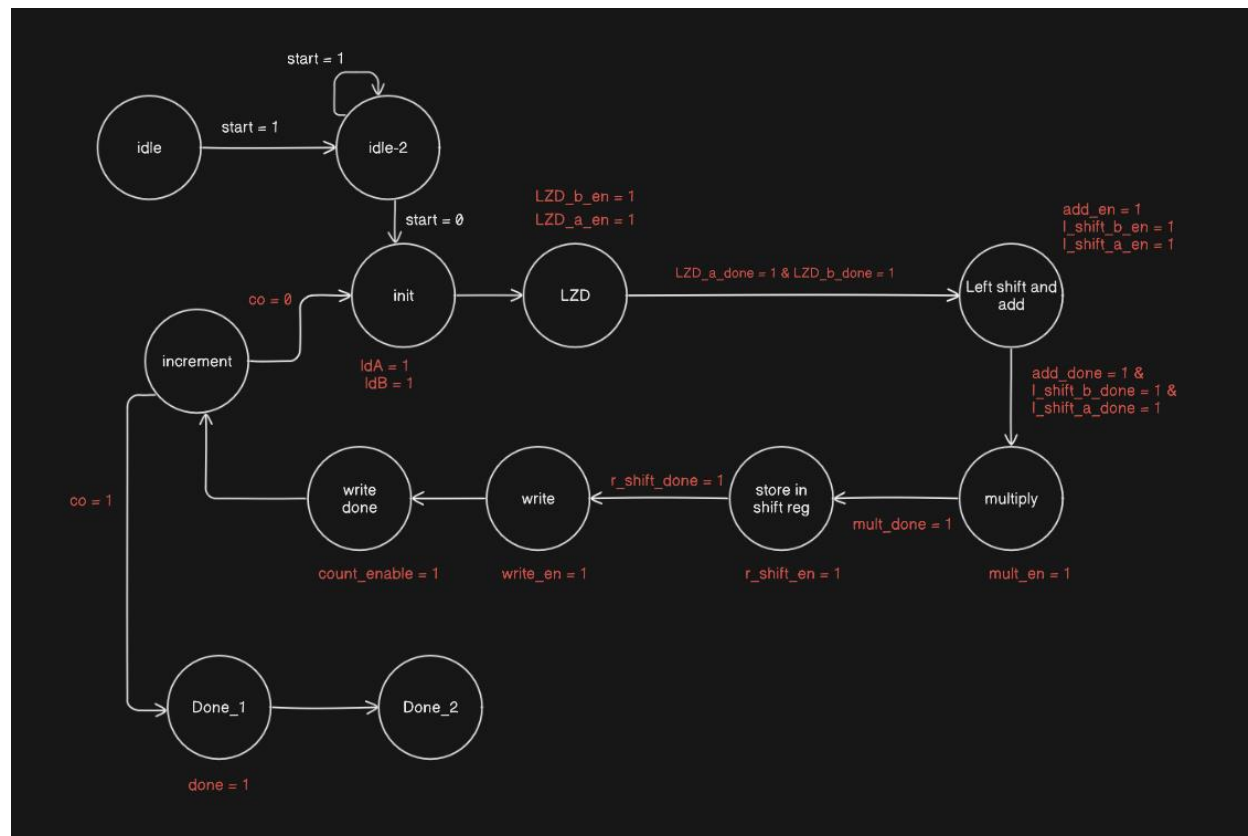
نکات:

- تمام کامپوننت ها به کلاک حساس هستند و سیگنال های ورودی را با لبه بالا رونده کلاک می خوانند.
- خواندن و نوشتن در فایل در datapath و controller نیامده اند چون با یک دستور ویلاگ قابل اجرا هستند.
- تقریباً تمام کامپوننت ها سیگنال enable دریافت و done را منتشر میکنند، این کار برای اطمینان از آماده بودن ورودی های یک کامپوننت قبل شروع کار آن انجام شده است.

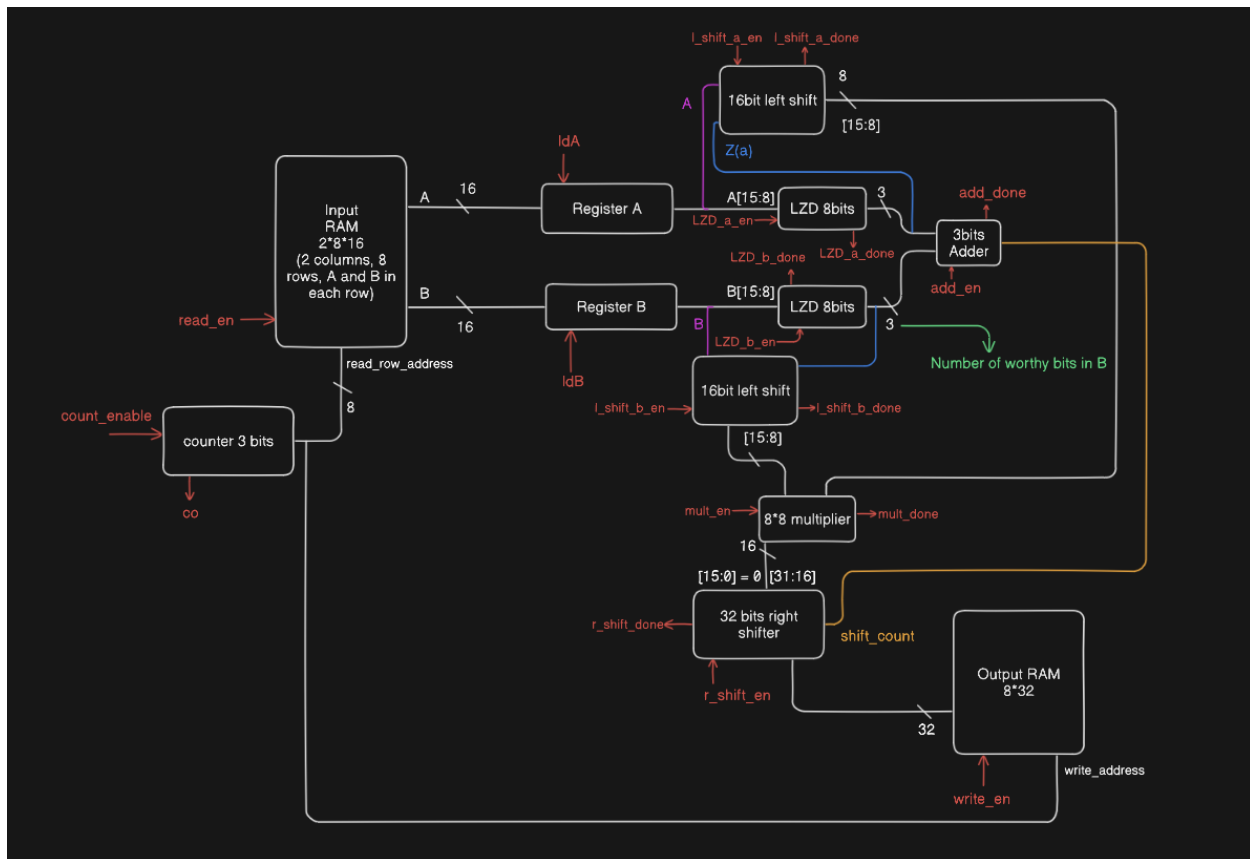
ایده کلی:

- اعداد A و B به صورت همزمان از مموری در رجیستر های مربوطه ریخته می شوند
- کامپوننت LZD مخفف Leading Zero Detector است که تعداد بیت های بی ارزش را محاسبه میکند، در ادامه کامپوننت Left Shift به این تعداد عدد را به چپ شیفت می دهد و 8 بیت با ارزش را بر میدارد.
- جمع بیت های بی ارزش دو عدد در Adder محاسبه می شوند.
- 8 بیت با ارزش در هم ضرب می شوند.
- حاصل 16 بیتی در نیمه چپ 32 Right shifter بیتی قرار میگیرد و به تعداد مجموع بیت های بی ارزش، عدد به راست شیفت داده می شود
- جواب نهایی در رم نوشته می شود Counter افزوده شده و اگر شمارنده overflow نکند به مرحله بعد می رویم.

Controller:



Datapath:

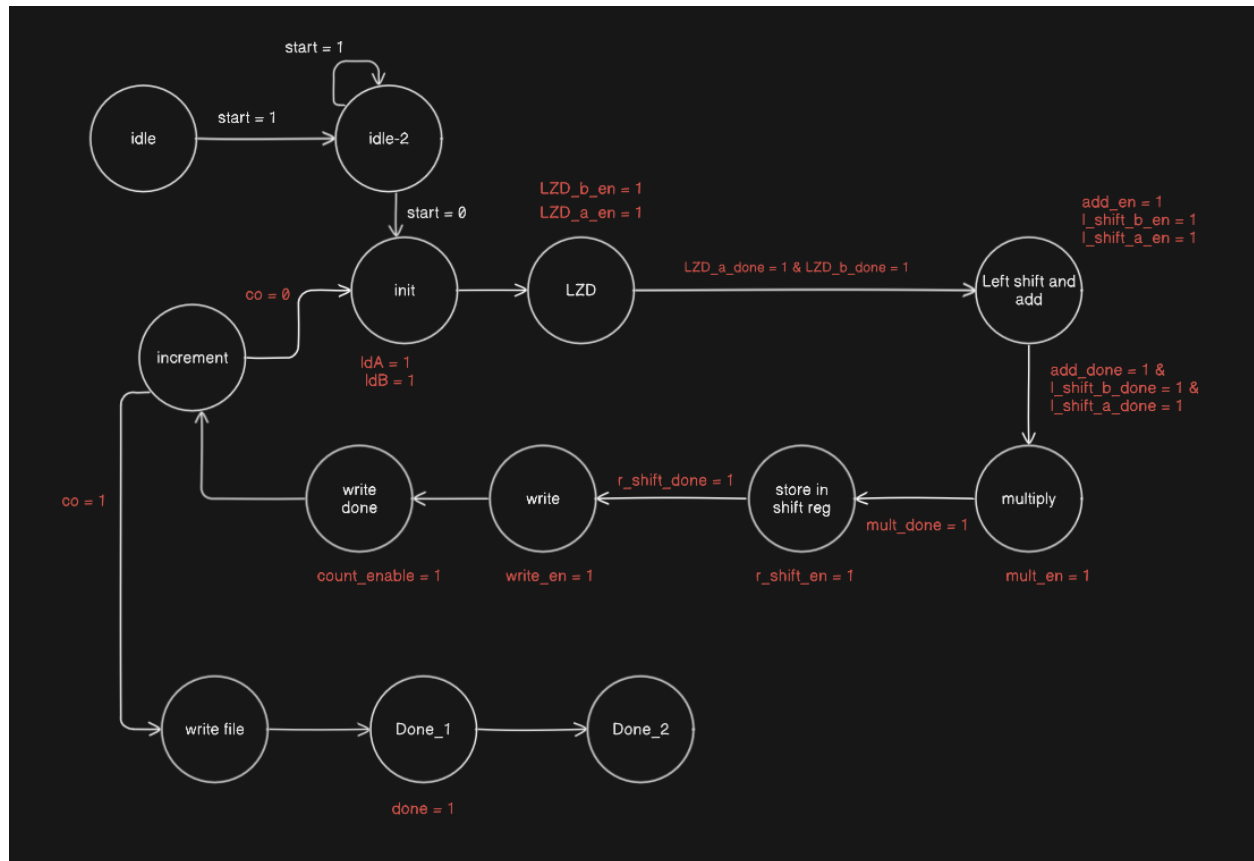


گزارش کار فاز دوم پروژه اول درس طراحی سیستم‌های دیجیتال

تغییرات لحاظ شده در فاز دوم:

1. سیگنال write file را به output ram اضافه کرده‌ایم، تا مشخص کنیم چه زمانی باید در ram خروجی نوشته شود.
2. همچنین یک state برای نوشتن داخل فایل اضافه کردیم.
3. تعداد بیت‌های خروجی leading zero باید چهارتا باشد، نه سه تا.
4. به همین دلیل سوم، shift count برای shifter 16 بیتی چهار بیتی می‌شود، نه سه بیتی.
5. به همین دلیل سوم، باید Adder نیز چهار بیتی باشد نه سه بیتی.
6. تغییر تعداد بیت‌های ورودی Adder سبب تغییر تعداد بیت خروجی آن می‌شود و دیگر چهار نیست بلکه پنج است، پس shift count برای shifter 32 بیتی 5 بیتی خواهد بود.
7. چون registerهایی سر خروجی Input ram قرار دارند، پس وجود سیگنال read enable برای آن ضروری نیست.

Controller:



Datapath:

