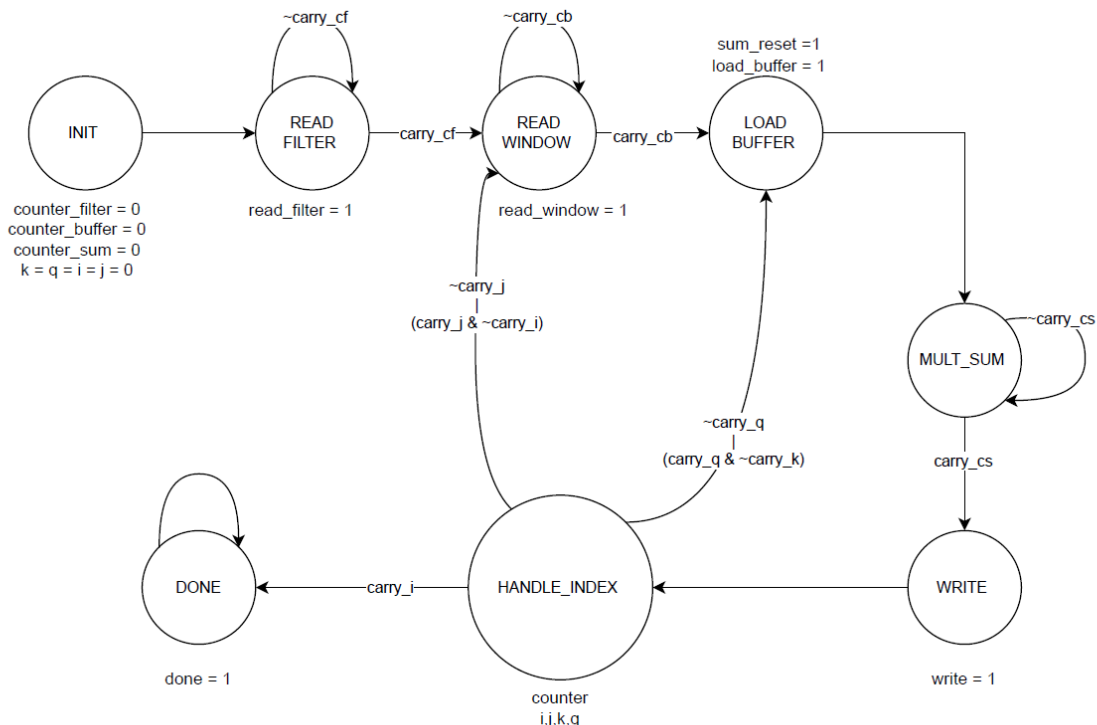


توضیح کلی:

- اکثر دیتا پث و کنترلر بدون تغییر مانده اند و با همان منطق توضیح داده شده کار میکنند.
- بر اساس سوال از بافر ۸×۸ استفاده شده است، این حالت بهینه است و تنها کافیت ۹ بار از مموری بخوانیم (کمترین تعداد ممکن).
- همانطور که نوشته شده است، آدرس دهی به کنترلر سپرده شده است. طبق صحبت استاد، نوشتن کد رفتاری برای آدرس دهی مانعی نداشت.

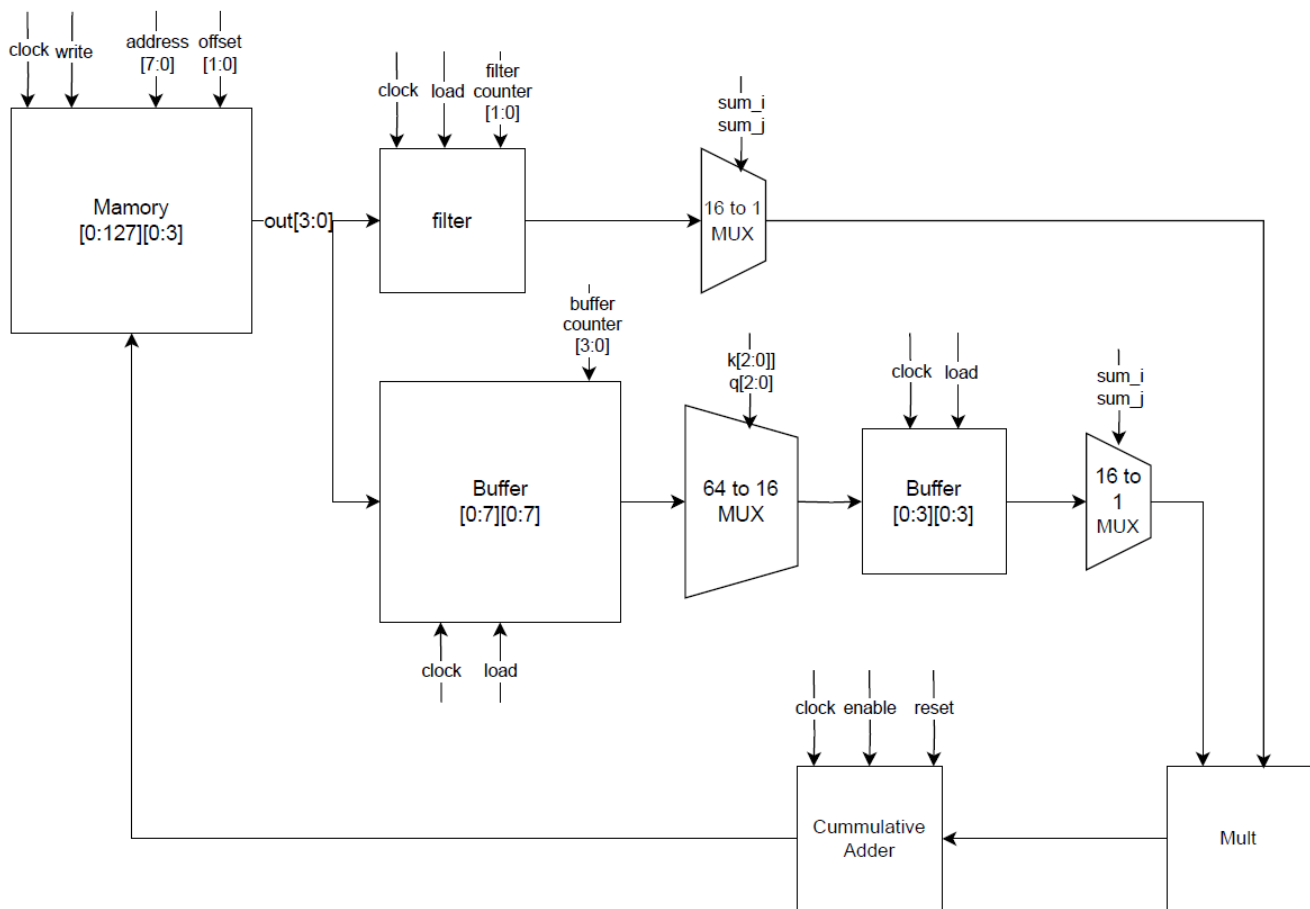
تغییرات کنترلر:

- طبق صحبت استاد، خواندن از مموری ۴ بایت در نظر گرفته شده است. به همین دلیل تعداد کلاک خواندن فیلتر به ۴ و خواندن بافر ۶۴ تایی به ۱۶ رسیده است.
- دو استیت SUM و MULT ادغام شده اند و به یک استیت MULT_SUM کاهش یافته اند و در ۱۶ کلاک کار میکنند. (طبق صورت پروژه)
- جمع کننده درختی به جمع کننده تجمعی تغییر یافت (بر اساس صحبت استاد مدرسی استفاده از جمع کننده درختی مانعی نداشت ولی برای تطبیق بیشتر با صورت پروژه این تصمیم را گرفتم).
- برای اندیس دهی از ۶ اندیس استفاده شده بود، برای بهینه شدن به ۴ اندیس کاهش یافت. (دو اندیس دیگر با روابطی از دیگر چهار اندیس به دست می آیند).
- به علت کمبود وقت، بسیاری از سیگنال های کنترلی نوشته نشده بودند.



تغییرات دیتا پث:

- تصحیح: در صفحه ۲ برای ضرب از بافر ۴ در ۴ استفاده شده ولی در صفحه ۵ رسم این بافر را فراموش کردم.
- در صورت سوال نحوه خواندن از مموری نیامده بود، و بعد از جلسه طبق صحبت اساتید، اندازه مموری ۱۲۸ خانه ۴ بیتی در نظر گرفته شده است.
- خروجی ضرب در ۱۶ بیت ذخیره می شود و ۸ بیت پرارزش آن خروجی داده می شود.
- خروجی جمع در ۱۲ بیت ذخیره می شود و ۸ بیت پرارزش آن خروجی داده می شود.
- به علت کمبود وقت، از اندیس دهی برای مشخص کردن خانه های حافظه شده است، در طراحی به مولتی پلکسر تغییر یافته اند.



نحوه کلی کار:

- با ۹ برش از تصویر (حالت بهینه) بافر ۸ در ۸ پر میشود.
- هر تصویر ۸ در ۸ به برش های ۲۵ برش ۴ در ۴ تقسیم می شود.
- کانولوشن حساب می شود و با توجه به اندیس ها در آدرس صحیح خود قرار می گیرند.
- حالت های تکراری مدیریت می شوند و بهینگی حفظ می شود.

تست:

■ آپدیت: این کد برای ضرب عادی در نظر گرفته شده بود. با توجه به گفته های جدید، دیگر این تست کیس ها کار نخواهند کرد.

- برای تست، فایل پایتونی به نام testGenerator.py در نظر گرفته است که دو خروجی input.mem و expectedOutput.mem را تولید می کند.
- در هر بار اجرا این برنامه، مقادیر با اعداد رندومی پر می شوند و این در نظر گرفته شده است که ضرب و جمع اعداد ایجاد شده overflow ایجاد نکند.
- در انتها مموری به طور ۱۲۸ در ۴ در فایل output.mem ذخیره می شود، و می توان با expectedOutput.mem مقایسه شود.
- همه ی این فایل ها در ./sim/file قرار دارند.

نکات دیگر:

- طراحی عدد های منفی را نیز مدیریت می کند.
- تصاویر با کیفیت تر کنترلر و دیتاپث در diagram.pdf قرار دارند.