

# CAD CA2

رضا چهرقانی 810101401

مهدی مختاری 810101515

---

در بافر معمولی یک حافظه با اندازه پارامتری تعریف می کنیم تا مقادیر را در آن ذخیره کنیم. یک `always` تعریف می کنیم تا با سیگنال `rstn` که ریست با مقدار صفر است این حافظه را برابر صفر کند. سپس برای نوشتن موازی در حافظه با لبه بالا رونده کلاک از `generate` و `for` کمک می گیریم تا بتوانیم به صورت موازی مقادیر را بنویسیم. از آنجایی که خواند به صورت آسنکرون می باشد از `assign` استفاده می کنیم تا مقدار مشخص شده با آدرس خواندن را بر روی خروجی قرار دهیم. البته برای خواندن موازی در اینجا هم از `for` استفاده می کنیم.

در بافر حلقوی برای آنکه در هر لحظه بدانیم که در کجا باید بنویسیم از `waddr` و برای دانستن آنکه از کجا باید بخوانیم از `raddr` استفاده می کنیم. نکته ای که وجود دارد این است که وقتی این دو پوینتر روی هم می افتند و دیگر نمی توان تشخیص داد که بافر خالی است یا اینکه پر است. برای حل این مشکل یک `flip-flop` با نام `last_op` قرار می دهیم تا به یاد داشته باشد که آخرین عملیات چه بوده است. زیرا وقتی آخرین عملیات خواندن باشد پس حتما در وضعیت خالی هستیم و اگر آخرین عملیات نوشتن باشد حتما در وضعیت پر هستیم.

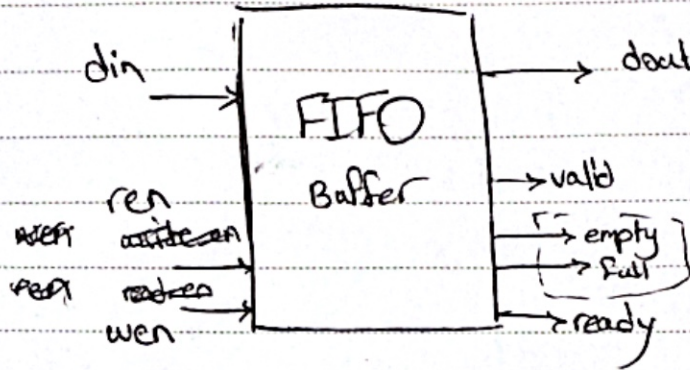
همچنین به دلیل مبهم بودن صورت پروژه سیگنال های خروجی را به این صورت تعریف می کنیم که `full` و `empty` زمانی رخ می دهد که بافر کاملاً پر و کاملاً خالی است و `ready` زمانی رخ می دهند که بافر به اندازه تعداد نوشتن موازی جا داشته باشد و `valid` زمانی رخ می دهد که بافر به تعداد خواندن موازی داده داشته باشد. در واقع ممکن است بافر کاملاً پر نباشد ولی تعداد

خانه های خالی کمتر از تعداد نوشتن به صورت موازی باشد در این صورت بافر full نیست ولی از آنجایی که به اندازه کافی خانه خالی نداریم ready هم نیست.

دو سیگنال do\_w و do\_r تعیین می کنیم تا در صورت برقراری شرط نوشتن و خواندن این عملیات ها را انجام دهد و در صورت برقرار نبودن شرایط سیگنال های ورودی wen و ren را در نظر نگیرد.

حال برای آنکه بفهمیم در حالت انجام عملیات موازی آیا خانه های جلوتر برای خواندن یا نوشتن آماده هستند باید بباییم تا به تعداد PAR\_WRITE و PAR\_READ خانه ی جلویی را چک کنیم که آیا waddr با raddr برابر می شود یا نه. نتایج این چک کردن ها را در is\_any\_equal\_w و is\_any\_equal\_r می ریزیم. در صورتی که هر کدام از بیت های این دو برابر یک شود دیگر نمی توانیم عملیات را انجام دهیم.

در انتها نیز اگر عملیات نوشتن یا خواندن انجام شد ما باید پوینتر های آن ها را یعنی waddr و raddr را آپدیت کنیم و به تعداد موازی سازی شده مقادیر آن ها را اضافه کنیم.



دین  
 رن  
 ون  
 دوت  
 وید  
 امی  
 فول  
 ریدی

