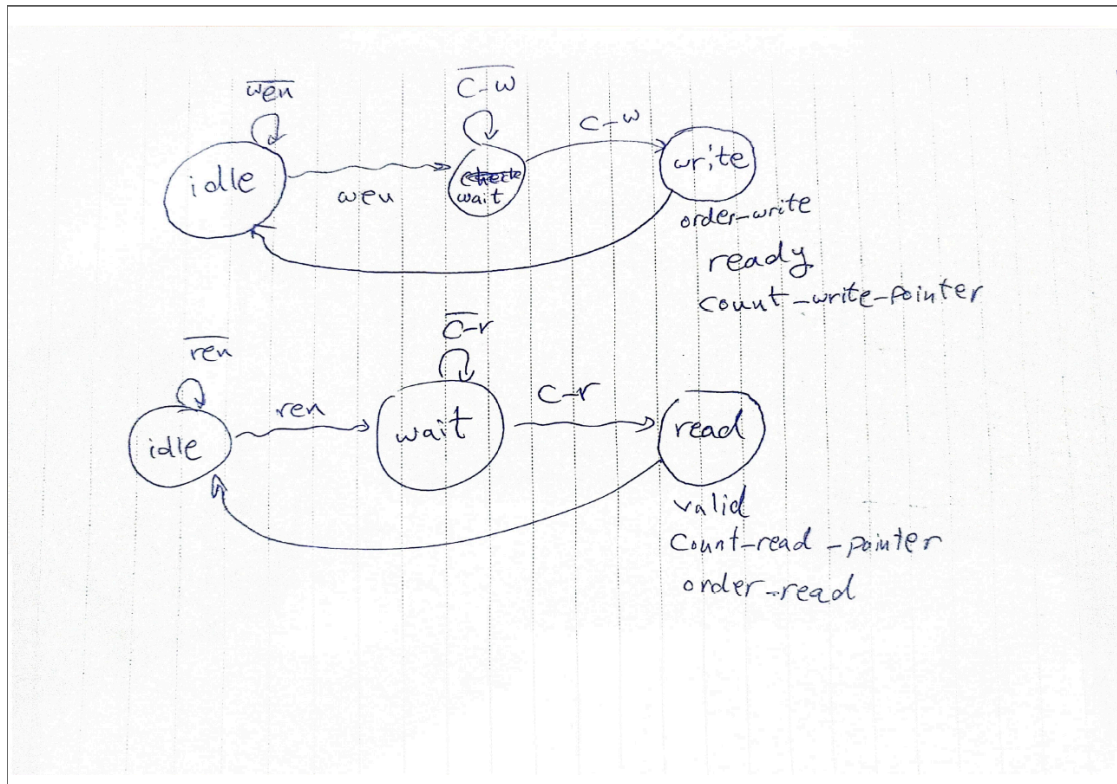
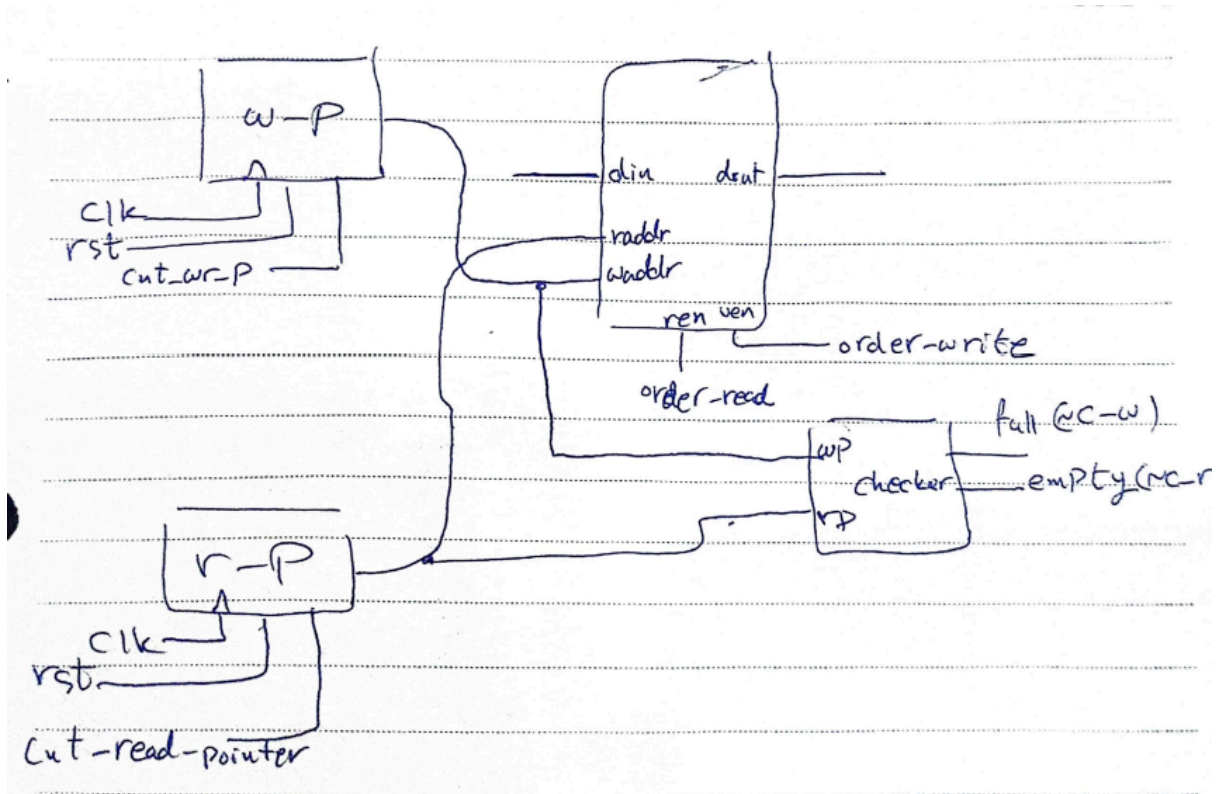


پروژه دوم درس طراحی کامپیوتری سیستم های دیجیتال

محمد امین توانایی رنجبر : 810101396

سید محمد حسین مظهري : 810101520



کنترلر و دیتابث بالا مربوط به بافر چرخشی هستند .

بافر معمولی مانند یک حافظه عمل می کند که ورودی و خروجی دارد و آدرسی برای خواندن و آدرسی برای نوشتن میگیرد .

در بافر چرخشی این آدرس خواندن و نوشتن را خود ما به آن می دهیم و از بیرون صرفاً دستور نوشتن یا خواندن دریافت می شود و مانند یک FIFO برخورد می شود .

وقتی ما درخواست خواندن یا نوشتن را دریافت می کنیم ، در ابتدا بررسی میکنیم که آیا امکان پاسخگویی به این درخواست هست یا خیر. یعنی مثلاً برای خواندن بررسی میکنیم آیا داده ای داخل بافر هست و برای نوشتن بررسی میکنیم آیا جایی برای نوشتن وجود دارد ؟

سپس **valid** و **ready** را بر اساس پاسخ به این جواب مقدار دهی می کنیم و در صورت نوشتن ، داده ی جدید را در بافر مینویسیم و در صورت خواندن ، داده مناسب را به خروجی میدهیم .

وقتی درخواست نوشتن داده می شود و امکان نوشتن وجود ندارد ، ما منتظر می مانیم تا تعداد مناسبی خواندن صورت بگیرد تا بافر جا برای نوشتن داشته باشد ، سپس عمل نوشتن را انجام می دهیم.

وقتی درخواست خواندن داده می شود و امکان خواندن وجود ندارد ، ما منتظر می مانیم تا تعداد مناسبی نوشتن صورت بگیرد تا اطلاعات کافی برای خواندن داشته باشد ، سپس عمل خواندن را انجام می دهیم .

این بافر امکان نوشتن همزمان در چند خانه و خواندن همزمان از چند خانه را فراهم میسازد . پس هنگام بررسی شروط بالا ، باید بررسی کنیم به تعداد مناسب خانه خالی / پر وجود داشته باشد نه صرفاً یک خانه .

با خواندن و نوشتن ، باید پوینتر اشاره کننده به محل خواندن و نوشتن را تغییر دهیم تا FIFO به درستی پیاده سازی شود . یعنی داده ای که اول نوشته شده است ، اول هم خوانده شود .

بافر دو سیگنال **full , empty** را نیز خروجی می دهد که به ترتیب به معنای خالی یا پر بودن آن است . (خالی بودن یعنی مثلاً ممکن است **PAR_READ** برابر با 4 باشد و ما 3 داده داخل بافر داشته باشیم ، آنگاه چون نمیتوانیم از بافر بخوانیم ، بافر خالی حساب می شود)

برای پر بودن نیز همینطور است . (یعنی ممکن است 2 جای خالی داشته باشیم ولی **PAR_WRITE** برابر با 4 باشد و نتوان داده را در بافر نوشت)

در این موقع باید صبر کنیم تا فضای مناسب برای عملیات مورد نظر به وجود آید .

دو تا کنترلر جداگانه نیز برای خواندن و نوشتن در نظر گرفته ایم که خواندن **valid** و نوشتن **ready** را ست میکند .

هنگام ورودی و خروجی دادن ، جای اینکه به تعداد **PAR_WRITE / PAR_READ** تا خروجی/ورودی داشته باشیم ، یک خروجی / ورودی به طول **PAR_READ * WORD_LENGTH** و **PAR_WRITE*WORD_LENGTH** داریم .

تصویر شبیه سازی :

