

LIFO - گروه اول

در این سوال شما باید یک LIFO پیاده‌سازی کنید که از RAM برای ذخیره‌سازی داده‌ها استفاده می‌کند. داده‌هایی که در آن ذخیره می‌کنیم 8 بیتی هستند و هم چنین این حافظه گنجایش 8 عدد به این شکل دارد. در اینجا ما دو سیگنال read و write را خواهیم داشت که مشخص می‌کند قصد خواندن یا نوشتن اطلاعات در RAM را داریم. هم چنین دو pointer هم برای عملیات خواندن یا نوشتن خواهیم داشت که مشخص می‌کنند در کدام قسمت از حافظه قرار داریم. دقت کنید که در زمانی که RAM خالی است توانایی خواندن داده و در زمانی که پر است توانایی نوشتن داده را نخواهیم داشت (می‌توانید از سیگنال‌های دلخواه برای تشخیص پر یا خالی بودن حافظه استفاده کنید).

گروه دوم - special queue

در این سوال شما باید یک queue خاص درست کنید. این صف، مانند صف های عادی از طریق inqueue کردن، ورودی میگیرد و ورودی آن تنها اعداد 1 تا 10 هستند. تنها تفاوت آن با صف های عادی این است که dequeue کردن آن به ترتیب نیست. خروجی این صف عضوی است که کمترین میزان وارد شدن را در این صف دارد.

اگر این میزان برای 2 ورودی یکسان بود، عضوی خارج میشود که مقدار آن کمتر است (مثلا اگر 2 و 4 یک میزان تکرار را داشتند، 2 خارج میشود).

برای درک بیشتر این موضوع این مثال را فرض کنید.

صف در ابتدا خالی است، سپس اعداد زیر وارد صف میشوند.

10-9-9-10-1-4-4-4

حال قرار است dequeue رخ دهد. از عضو های درون صف، کمترین میزان وارد شدن را {1} دارد، پس این عضو خارج میشود.

حال اگر 2 بار 1 دوباره به صف بالا inqueue شود، تنها تفاوتی که ایجاد میکند این است که تعداد "وارد شدن" های 1 برابر با 3 میشود! (1 بار قبلا در صف بوده که خارج شده، 2 بار هم الان وارد شده) پس این دفعه موقع dequeue کردن، باید از بین 10 و 9 انتخاب کنیم که چون 9 از 10 کمتر است، 9 خارج میشود. (با اینکه در صف الان، 1-9-10 هر 3، دو بار در صف تکرار شدند)

راهنمایی: سعی کنید یک آرایه به اندازه 10 خانه داشته باشید که مثلا خانه i آن، تعداد وارد شدن های عدد z داخل queue را نگه دارد.

FIFO - گروه سوم

در این سوال شما باید یک FIFO پیاده‌سازی کنید که از RAM برای ذخیره‌سازی داده‌ها استفاده می‌کند. داده‌هایی که در آن ذخیره می‌کنیم 8 بیتی هستند و هم چنین این حافظه گنجایش 8 عدد به این شکل دارد. در اینجا ما دو سیگنال read و write را خواهیم داشت که مشخص می‌کند قصد خواندن یا نوشتن اطلاعات در RAM را داریم. هم چنین دو pointer هم برای عملیات خواندن یا نوشتن خواهیم داشت که مشخص می‌کنند در کدام قسمت از حافظه قرار داریم. دقت کنید که در زمانی که RAM خالی است توانایی خواندن داده و در زمانی که پر است توانایی نوشتن داده را نخواهیم داشت (می‌توانید از سیگنال‌های دلخواه برای تشخیص پر یا خالی بودن حافظه استفاده کنید).

گروه چهارم - Mini ALU

در این سوال قصد داریم یک ALU بسیار بسیار ساده تنها با دو عملگر طراحی کنیم که این عملگرها شامل ک.م.م و log بر پایه ۲ میباشد. توجه کنید که این ALU یک بیت نیز به عنوان Selector ورودی می‌گیرد که در صورت ۰ بودن آن عملگر ک.م.م و در صورت ۱ بودن آن عملگر log فراخوانی میشود. توجه کنید که datatype ها برای ورودی اعداد آزاد است و میتوانید طبق صلاحدید خود از datatype های موجود استفاده کنید.

موفق باشید.