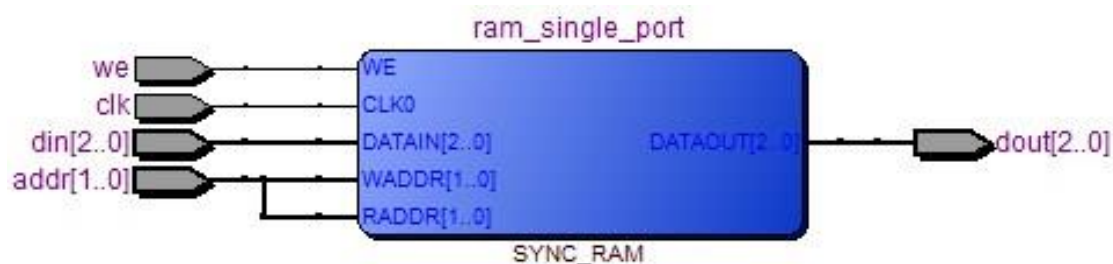


گزارش آزمایش ۱۰

سارا روحانی

امیر حسام مرادی

## Single-port-RAM



برای کار با این نوع حافظه به ۴ ورودی احتیاج داریم که در آخر یک خروجی به ما می دهد

در پیاده سازی این نوع حافظه تعداد و اندازه داده ها را به صورت generic مشخص می کنیم

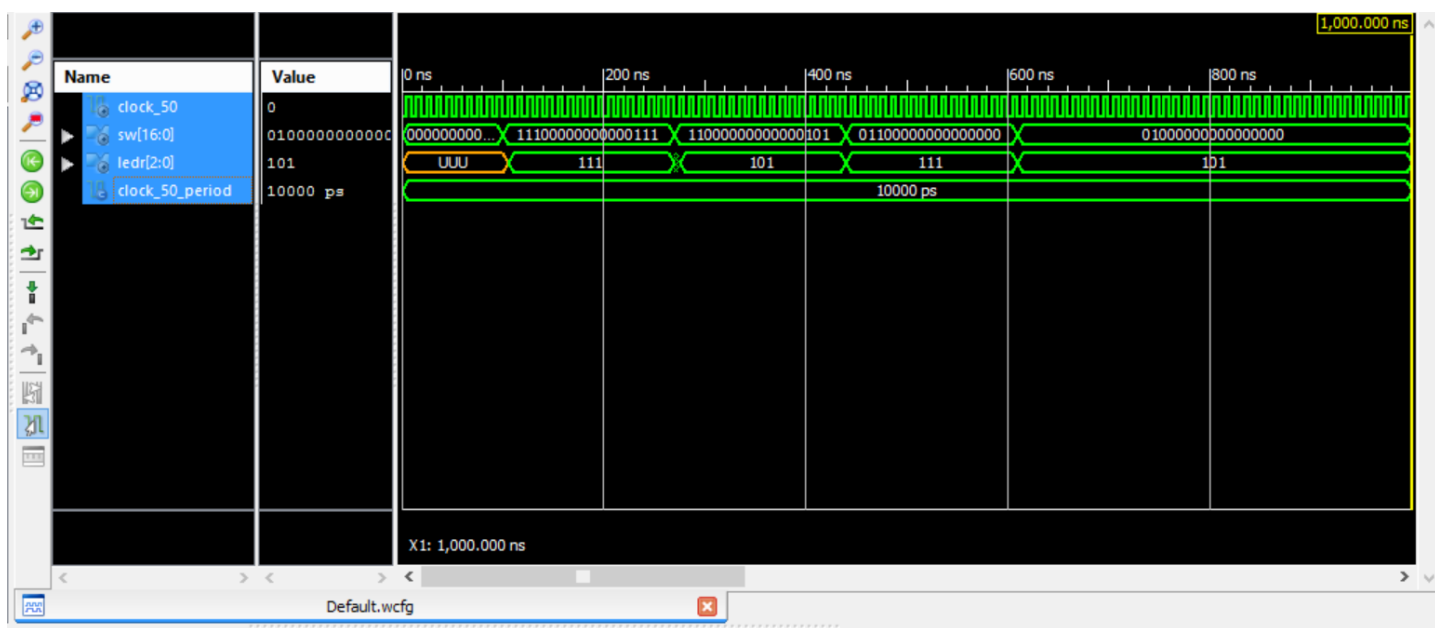
ورودی ها

۱- clk که ورودی کلاک است

۲- we از این ورودی برای مشخص می کند که آیا داده در حافظه ذخیره شود یا که اگر ۱ باشد می توان داده را ذخیره کرد

۳- din این برای گرفتن داده استفاده می شود

۴- addr از این ورودی برای مشخص کردن خانه مورد نظر در حافظه استفاده می شود . در حافظه های تک پورتی این آدرس هم برای خواندن و هم برای نوشتن استفاده می شود که تشخیص این ۲ دستور با ورودی we انجام می شود . در نتیجه در این نوع از حافظه امکان خواندن و نوشتن همزمان وجود ندارد .



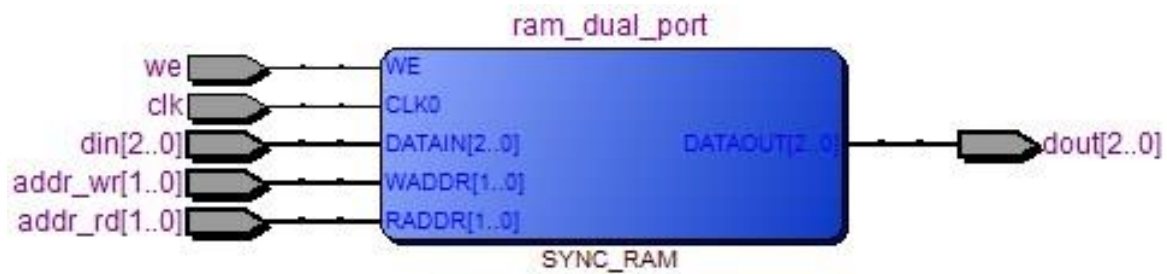
برای تست این حافظه ورودی ۱۷ بیتی SW که بیت های آن به صورت زیر است به سیستم می دهیم

Sw[16] -> we , sw[15:14] -> addr , sw[2:0] -> din

برای این تست ابتدا ۲ داده در حافظه می نویسیم برای این کار بیت سمت چپ SW باید ۱ باشد و سپس ۲ آدرس مشخص می کنیم که در این تست خانه های 10 و 11 را برای نوشتن حافظه انتخاب کردیم و بعد داده مورد نظر را در ۳ بیت سمت راست SW قرار می دهیم .

در مرحله بعد برای اطمینان از عملکرد صحیح حافظه ، مقدار این دو خانه را می خوانیم که برای این کار بیت سمت چپ SW را 0 قرار می دهیم و آدرس خانه های مورد نظر را موقعیت addr در sw وارد می کنیم همانطور که در شکل بالا پیداس حافظه به درستی عمل می کند .

## Dual-port-RAM

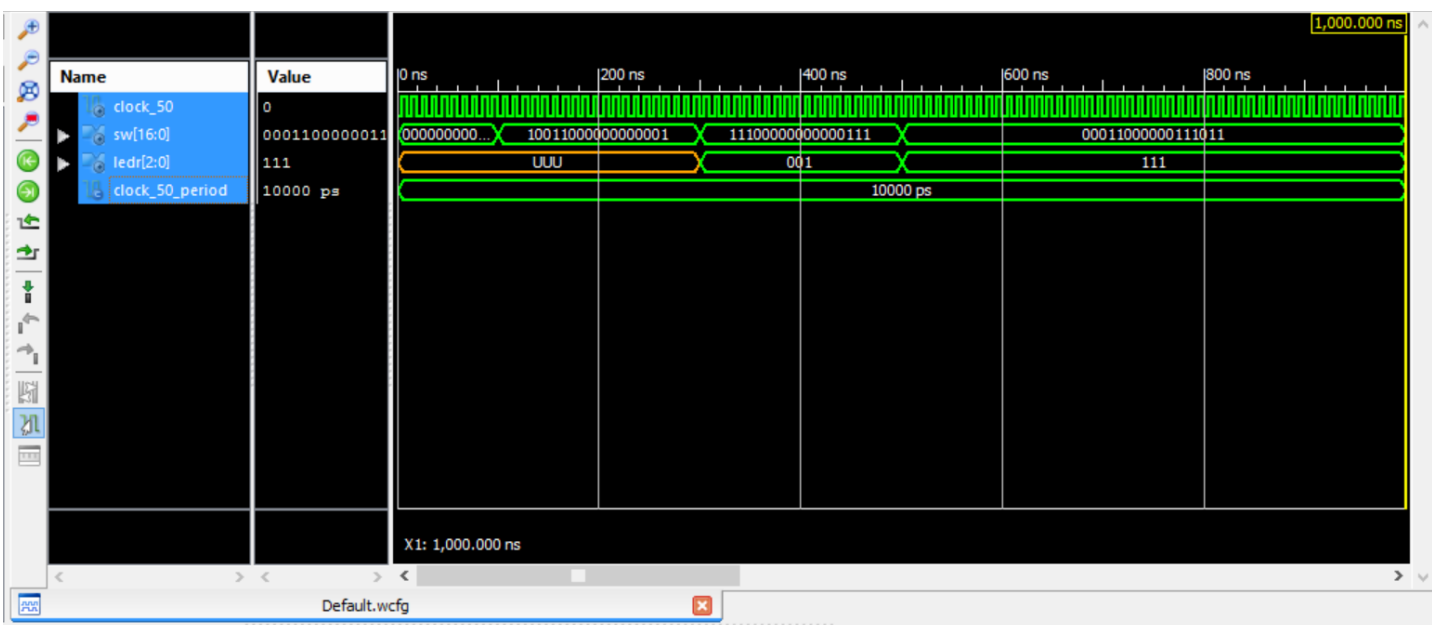


این نوع حافظه مشابه حافظه تک پورتی است با این تفاوت که می توان همزمان دستور خواندن و نوشتن را بر روی این نوع حافظه اجرا کرد . برای همین در این حافظه ۲ پایه ورودی برای دریافت آدرس داریم

`Addr_wr` آدرس خانه مورد نظر برای نوشتن داده

`Addr_rd` آدرس خانه مورد نظر برای خواندن داده

بقیه ورودی ها مانند حافظه قبلی است



برای تست این حافظه ورودی ۱۷ بیتی sw که بیت های آن به صورت زیر است به سیستم می دهیم

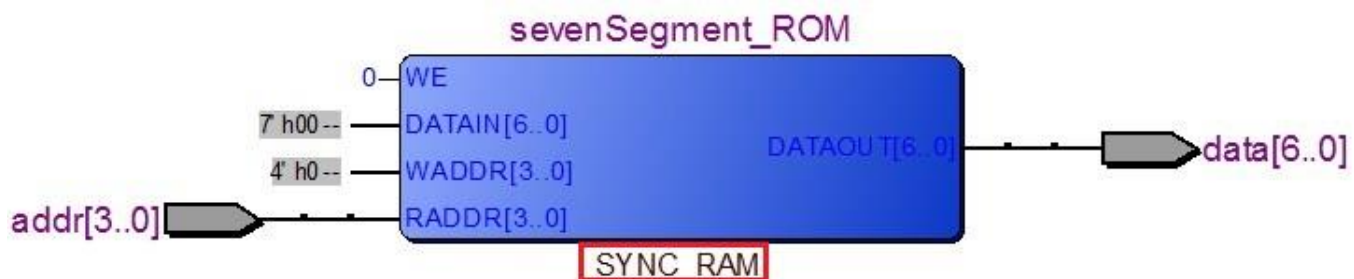
Sw[16] -> we , sw[15:14] -> addr\_wr , sw[13:12] -> addr\_rd , sw[2:0] -> din

ابتدا داده 001 را در خانه 00 می نویسیم و همزمان سعی می کنیم مقدار ذخیره شده در خانه 11 را بخوانیم و چون هنوز در خانه 11 مقداری نوشته نشده است در خروجی UUU نمایش داده می شود

در مرحله بعد داده 111 را در خانه 11 می نویسیم و مقدار خانه 00 را می خوانیم ، همانطور که در شکل پیداست همان مقداری که در مرحله قبل در خانه 00 ذخیره کردیم در خروجی نمایش داده می شود .

در مرحله بعد بدون نوشتن مقداری در حافظه داده موجود در خانه 11 را می خوانیم که برای این کار we را 0 و مقدار addr\_rd را برابر 11 قرار می دهیم و می بینیم که مقدار آن به درستی در خروجی قرار می گیرد .

## ROM



در این نوع حافظه امکان نوشتن وجود ندارد و حافظه در پیاده سازی مقدار دهی نیز می شود .

داده ها در این حافظه به صورت یک آرایه ۲ بعدی ذخیره شده است که هر سطر آن یک داده است .

در اینجا هر داده ۷ بیت است و در کل ۱۶ داده داریم . برای خواندن هر داده داشتن شماره سطر آن کافی است

این شماره سطر به صورت ورودی addr در ۴ بیت دریافت می شود و خروجی ۷ بیتی نمایش داده می شود .

