**گزارش دستورکار نهم آزمایشگاه معماری کامپیوتر**

نگار موقتیان، 9831062

**ماژول ROM\_16x8**

در این قسمت از آزمایش می­خواهیم یک حافظه از نوع ROM طراحی کنیم، به صورتی که 16 ردیف 8 بیتی داشته باشد (در حقیقت در این حافظه word = 8 bit). مدار توصیف شده چنین شکلی خواهد داشت:

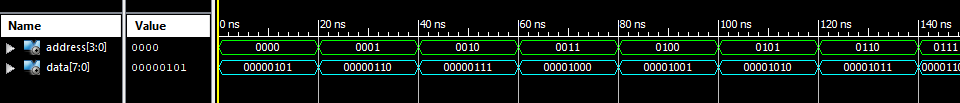
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| . . . | . . . | . . . | . . . | . . . | . . . | . . . | . . . |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

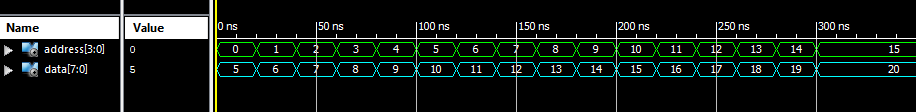
16

word = 8 bit

در ضمن می­دانیم حافظۀ ROM یک حافظۀ فقط خواندنی است، بنابراین در ابتدای برنامه به خانه­های آن مقداری پیش­فرض می­دهیم و پس از آن (پس از کلید واژۀ begin و همچنین در ماژول­های دیگری که از این حافظه استفاده خواهند کرد) اجازۀ تغییر مقادیر موجود داخل این حافظه را نداریم. در این آزمایش برای سادگی بررسی درستی رفتار این ماژول، محتوای داخل هر خانه مانند شکل بالا، 5 به علاوۀ شمارۀ خانه (یا همان ردیف) در نظر گرفته شده است.

این حافظه 16 خانه با شماره­ها 0 تا 15 دارد و می­دانیم این بازه از اعداد را می­توان توسط یک عدد دودویی 4 بیتی نمایش داد (از 2(0000) تا 2(1111)) بنابراین برای آدرس دهی به این حافظه یک گذرگاه 4 بیت کافی است. به عبارتی با دریافت یک آدرس 4 بیتی (گذرگاه address)، محتوای خانۀ مربوطه از این حافظۀ ROM را به خروجی 8 بیتی (گذرگاه data) متصل می­نماییم.

با توجه به این توضیحات حافظۀ مورد نظر را پیاده سازی کرده و در test bench مربوطه، با دادن مقادیر مختلف به گذرگاه آدرس درستی عملکرد آن را بررسی می­کنیم.

می­دانیم آدرس یک عدد بی علامت است و فرض می­کنیم اعداد داخل ROM به صورت بی علامت ذخیره شده­اند. بنابراین برای سهولت بررسی خروجی، ورودی و خروجی مدار را به صورت unsigned decimal نمایش می­دهیم.

همانطور که مشاهده می­شود این مدار عملکرد درستی دارد و مقادیر پیش­فرض موجود در ROM را با توجه به آدرس داده شده در خروجی انعکاس می­دهد.