

## طراحی سیستم های دیجیتال

دکتر اجاللی

نیکا قادری  
بهار ۱۴۰۳

تمرین اول

## طراحی حافظه

تاریخ گزارش: ۲۳ فروردین ۱۴۰۳

## ۱ مقدمه

در این تمرین هدف بر آن است که با استفاده از توصیف رفتاری، یک حافظه یک کیلو بایتی طراحی شود که دارای ده خط آدرس و هشت خط داده می شود. داده باید به صورت *inout* تعریف شود بدین معنی که هم می تواند در نقش خروجی باشد و هم ورودی بگیرد.

## ۲ طراحی ماژول ها

ماژول حافظه ابتدا باید ۱۰۲۴ بایت حافظه داشته باشد، که با تعریف رجیستر های ۸ بیتی *ram*، این امر محقق می شود. سپس از آنجایی که نمی توان به دیتا مستقیماً مقدار داد، یک رجیستر موقتی به نام *data - reg* تعریف می کنیم و در مواقعی که ورودی نوشتن فعال باشد آن را به دیتا *assign* می کنیم. در غیر این صورت دیتا قطع می شود تا بتوان ورودی مناسب را (به طور مثال از تست بنچ) گرفت.

```

1 module customMemory(input [9:0] address, input readNot_write, inout [7:0] data);
2
3     reg [7:0] ram [1023:0];
4     reg [7:0] data_reg;
5
6     assign data = (readNot_write) ? {8{1'bz}} : data_reg;
7
8     always @(address or readNot_write or data) begin
9         if (readNot_write) begin
10             ram[address] = data;
11         end else begin
12             data_reg = ram[address];
13         end
14     end
15 endmodule

```

شکل ۱: حافظه

همان طور که مشاهده می شود در بلاک *always*، هر گاه که ورودی نوشتن غیر فعال باشد مقدار *ram[address]* روی رجیستر موقت - و در نتیجه روی سیگنال دیتا - نوشته می شود. در غیر این صورت تنها داده دریافتی را روی آدرس مناسب از حافظه می نویسیم.

## ۳ تست

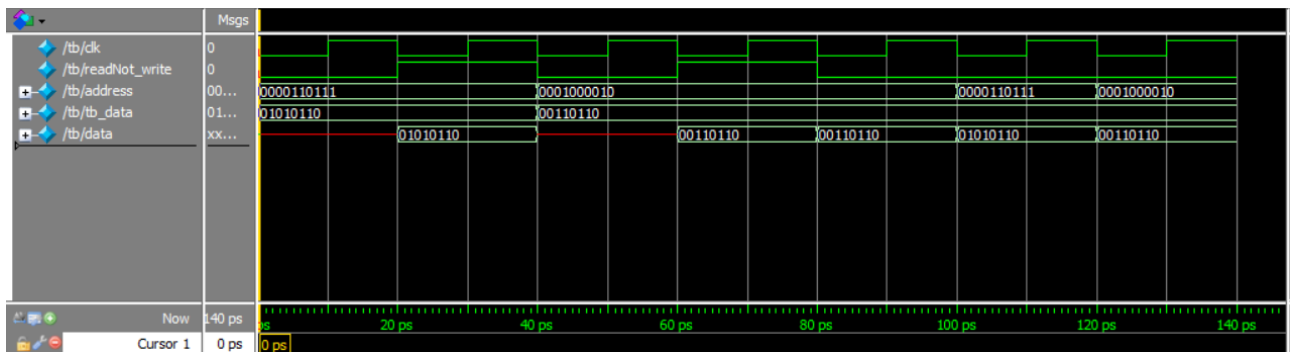
برای تست کد، در ماژول زیر، سیگنال های ورودی و خروجی و کلاک را تعریف می کنیم و در بلاک *initial* ورودی های مختلف را آزمایش می کنیم. البته از ماژول حافظه ابتدا باید اینستنس گرفته شود. نتایج این تست هم در قالب *wave* و هم از روی ترمینال قابل مشاهده هستند:

```

VSIM 51> run
# reading from an unknown address (uninitialized):
# readNot_write=0, address=0000110111, data=xxxxxxx
# writing to an address (55):
# readNot_write=1, address=0000110111, data=01010110
# reading from an unknown address (uninitialized):
# readNot_write=0, address=0001000010, data=xxxxxxx
# writing to an address (66):
# readNot_write=1, address=0001000010, data=00110110
# reading from the previous address (66):
# readNot_write=0, address=0001000010, data=00110110
# reading from the first address (55):
# readNot_write=0, address=0000110111, data=01010110
# reading from 66 again:
# readNot_write=0, address=0001000010, data=00110110
# Break in Module tb at D:/verilog/HW3/tb.v line 55

```

شکل ۲: خروجی ترمینال



شکل ۳: خروجی *wave*

## ۴ نتیجه گیری

با استفاده از توصیف رفتاری و بلاک های مطلوب، یک حافظه یک کیلو بایتی طراحی گردید و به ازای ورودی های مختلف، عملکرد آن در خواندن و نوشتن داده مورد بررسی قرار گرفت.