



دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

گزارشکار آزمایشگاه معماری کامپیوتر – شماره ۱۰

عنوان آزمایش: بررسی RAM, CAM

نام و نام خانوادگی گردآورندگان: حسنا اویار حسینی و پویا محمدی

استاد آزمایشگاه: جناب آقای مهندس عاروان

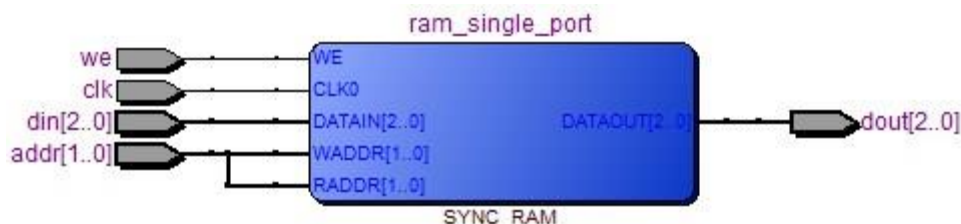
تاریخ آزمایش: ۱۴۰۰/۹/۳۰

آزمایش (۱)

نام آزمایش: پیاده سازی و بررسی single port RAM

شرح آزمایش: در این بخش از آزمایش میخواهیم یک single port RAM بنویسیم. ورودی و خروجی های یک single port RAM به صورت زیر می باشد:

ورودی ها		خروجی ها	
we	ورودی enable نوشتن در RAM (اگر $we = 1$ باشد عملیات نوشتن و اگر صفر باشد بازایی داده رخ میدهد)	Dout[2:0]	داده خروجی از RAM
clk	کلاک		
Din[2:0]	داده ورودی که قرار است در RAM ذخیره شود		
Addr[1:0]	آدرسی که داده باید در آن ذخیره شود و یا از آن خوانده شود		



و عملکرد آن به این صورت است که زمانی که we فعال باشد داده ای که به عنوان ورودی داده شده است در RAM و در آدرسی که توسط $Addr$ مشخص شده است ذخیره میشود و همچنین خروجی نیز همواره داده ای که در آدرس $Addr$ قرار دارد را نمایش میدهد. در مثالی که ما پیاده سازی کرده ایم din ۳ بیتی و $addr$ ۲ بیتی است پس یعنی RAM ما میتواند $2^2=4$ داده ۳ بیتی را در خود ذخیره کند.

برای پیاده سازی ابتدا باید طول داده و همچنین طول آدرس را مشخص کنیم که به وسیله دو متغیر از نوع generic این کار را انجام داده ایم ($data_width : integer := 3; addr_width : integer := 2$).

در VHDL، generic ها یک شکل محلی از Const هستند که وقتی یک جزء را نمونه سازی می کنیم، می توان مقداری به آن اختصاص داد. سپس یک آرایه با طول $addr_width$ میسازیم که عناصر آنها خود یک آرایه به طول $data_width$ هستند.

```
type ram_type is array (2**addr_width-1 downto 0) of std_logic_vector (data_width-1 downto 0);
```

با هر کلاک نگاه میکنیم اگر $we = 1$ بود din را در آرایه دو بعدی که تعریف کرده بودیم در خانه ای با شماره $addr$ مینویسیم. و در نهایت نیز داده ای که در درایه با شماره $addr$ وجود دارد را در خروجی نمایش میدهیم.

لازم به ذکر است با توجه به اینکه `addr` به صورت یک عدد باینری از ورودی گرفته شده است برای اینکه از آن بتوانیم به عنوان ایندکس برای دسترسی به درایه های آرایه استفاده کنیم باید `addr` را به `integer` تبدیل کنیم برای این کار از دستور زیر استفاده میکنیم:

```
to_integer(unsigned(addr))
```

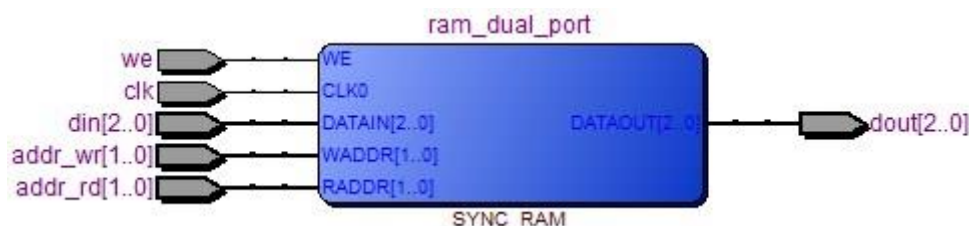
و اما علتی که به این `RAM` ، `single port RAM` گفته میشود این است که `RAM` تک پورت دارای یک پورت ورودی (به عنوان مثال خط آدرس) است که برای ذخیره و بازیابی داده ها استفاده می شود. در اینجا پورت `Addr` برای هر دو عملیات خواندن و نوشتن استفاده می شود.

آزمایش ۲)

نام آزمایش: پیاده سازی و بررسی dual port RAM

شرح آزمایش: در این بخش از آزمایش میخواهیم یک dual port RAM بنویسیم. ورودی و خروجی های یک dual port RAM به صورت زیر می باشد:

ورودی ها		خروجی ها	
we	ورودی enable نوشتن در RAM (اگر $we = 1$ باشد عملیات نوشتن و اگر صفر باشد بازایابی داده رخ میدهد)	Dout[2:0]	داده خروجی از RAM
clk	کلاک		
Din[2:0]	داده ورودی که قرار است در RAM ذخیره شود		
Addr_wr[1:0]	آدرسی که داده باید در آن ذخیره شود		
Addr_rd[1:0]	آدرسی که داده باید از آن خوانده شود		



و عملکرد آن به این صورت است که زمانی که we فعال باشد داده ای که به عنوان ورودی داده شده است در RAM و در آدرسی که توسط $Addr_wr$ مشخص شده است ذخیره میشود و همچنین خروجی نیز همواره داده ای که در آدرس $Addr_rd$ قرار دارد را نمایش میدهد. در مثالی که ما پیاده سازی کرده ایم din ۳ بیتی و $addr$ ۲ بیتی است پس یعنی RAM ما میتواند $2^2=4$ داده ۳ بیتی را در خود ذخیره کند. برای پیاده سازی ابتدا باید طول داده و همچنین طول آدرس را مشخص کنیم که به وسیله دو متغیر از نوع generic این کار را انجام داده ایم ($data_width : integer := 3; addr_width : integer := 2$).
*در VHDL، generic ها یک شکل محلی از Const هستند که وقتی یک جزء را نمونه سازی می کنیم، می توان مقداری به آن اختصاص داد.
سپس یک آرایه با طول $addr_width$ میسازیم که عناصر آنها خود یک آرایه به طول $data_width$ هستند.

```
type ram_type is array (2**addr_width-1 downto 0) of std_logic_vector (data_width-1 downto 0);
```

با هر کلاک نگاه میکنیم اگر $we = 1$ بود din را در آرایه دو بعدی که تعریف کرده بودیم در خانه ای با شماره $addr_wr$ مینویسیم. و در نهایت نیز داده ای که در درایه با شماره $addr_rd$ وجود دارد را در خروجی نمایش میدهیم.

لازم به ذکر است با توجه به اینکه $addr$ به صورت یک عدد باینری از ورودی گرفته شده است برای اینکه از آن بتوانیم به عنوان ایندکس برای دسترسی به درایه های آرایه استفاده کنیم باید $addr$ را به $integer$ تبدیل کنیم برای این کار از دستور زیر استفاده میکنیم:

```
to_integer(unsigned(addr))
```

و اما علتی که به این RAM ، $dual\ port\ RAM$ گفته میشود این است که در این نوع RAM بر خلاف نوع قبل که فقط یک خط آدرس برای هر دو عملیات خواندن و نوشتن داشت، خطوط آدرس اختصاصی برای عملیات خواندن و نوشتن (به ترتیب $addr_rd$ و $addr_wr$) ارائه شده است.

تفاوت $dual\ port\ RAM$ و $single\ port\ RAM$:

همانطور که گفته شد تفاوت این دو نوع رم در پورت های ورودی است اولی فقط یک پورت مشترک برای تعیین آدرس داده ورودی و خروجی دارد ولی دومی دو پورت جداگانه برای این کار دارد.