Data\_array :

برای پیاده سازی این ماژول یک آرایه ی 64 تایی که هر خانه 32 بیت دارد در نظر میگیریم که بعدا با دو بار instantiate کردن از آن کل کش را در بر میگیرد. در صورت 1 شدن wren باید wrdata ورودی را در آن خانه ای از آرایه که اندیسش address است بریزیم.

Tag\_valid\_array :

یک output داریم که 4 خانه ی اول Tag و خانه ی آخر valid را مشخص می کند. در صورت 1 شدن wren باید wrdata که همان tag است را در خانه ی 0 تا 3 آرایه خروجی بریزیم.

Invalidate هم برای مشخص کردن valid است

Miss\_hit\_logic :

میایم چک می کنیم که آیا تک تک بیت های tag با w0 یا w1 مساوی است یا خیر و درصورت 1 بودن valid میگوییم hit اتفاق می افتد.

Lru\_array :

در این ماژول برای بررسی ایکه کدام خانه دیرتر آمده است از دو آرایه ی کمکی برای way اول و دوم استفاده می کنیم به این صورت که اگر خانه ای که اندیس آن address است 0 باشد در Way دوم میریزیم و 0 و 1 این دو way را برعکس می کنیم . درواقع آن خانه ای که صفر است قدیمی تر بوده و در آن باید آدرس جدید را بریزیم.

Controller :

همان طور که در تعریف پروژه گفته شد به ترتیب عمل می کنیم .

Ram :

پیاده سازی شبیه ماژول data\_array میباشد

Cache :

باید از ماژول ها component بگیریم و با استفاده از signal که آن را تغییر می دهیم به ماژول ها port map کنیم

Cache\_ram :

باید از ماژول ها component بگیریم و با استفاده از signal که آن را تغییر می دهیم به ماژول ها port map کنیم