به نام خدا

پروژه ی درس CAD

دکتر خدادادی

اعضای گروه:

علیرضا اسلامی خواه(99521064)

امیرمحمد کمیجانی(99522032)

آرین عبدالهی ثابت نژاد(۹۹۵۲۱۴۴۲)

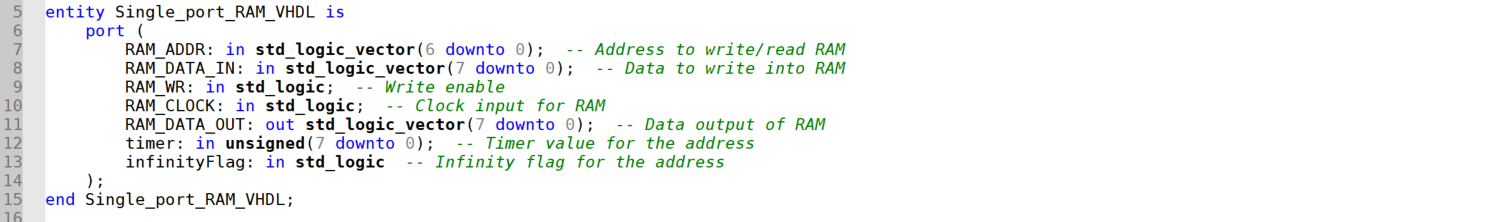
امیرعلی فرازمند(99522329)

موضوع پروژه:

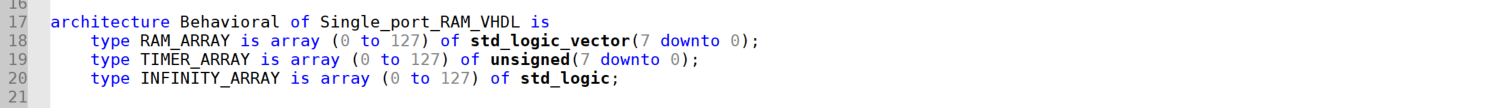
RAM تک پورتی 128 بایتی

با این قابلیت که خانه ها هرکدام در زمان مشخصی که از ورودی میگیرند میتوانند طوری تنظیم شوند که پاک شوند یا پاک نشوند

در این پروژه ما یک رم تک پورتی طراحی کردیم با حافظه ی 128 بایتی(128x8 بیتی).پورت انتیتی رم مان از این قرار است:

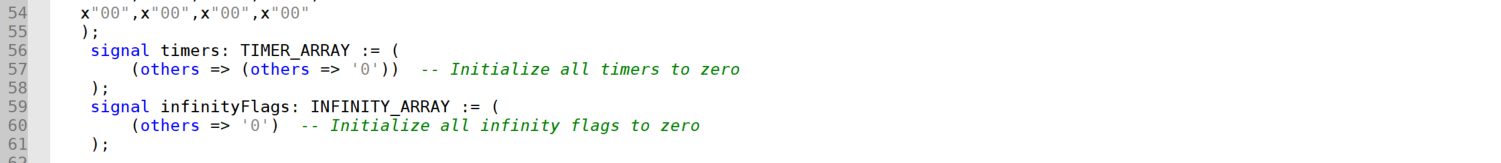


* **RAM\_ADDR**  پورت ورودی که آدرس نوشتن یا خواندن از RAM را نمایش می‌دهد(در این آدرس دیتا باید نوشته شود).
* **RAM\_DATA\_IN**  پورت ورودی که دیتا برای نوشتن در RAM میرود.
* **RAM\_WR** پورت ورودی که سیگنال فعال سازی نوشتن برای RAM را نمایش می‌دهد. (write enable)
* **RAM\_CLOCK** پورت ورودی که سیگنال کلاک برای RAM را نمایش می‌دهد.
* **RAM\_DATA\_OUT** پورت خروجی که داده خروجی از RAM را نمایش می‌دهد.
* **Timer** پورت ورودی که مقدار تایمر برای آدرس را نمایش می‌دهد (درصورتی که infinityFlag ما 0 باشد از آن کاسته میشود).
* **infinityFlag** پورت ورودی std\_logic که برای آدرس میگوید که آیا مقدار خانه باید تا بینهایت بماند یا که با کم شدن تایمر و در نهایت با به 0 رسیدن آن دیتا را از رم پاک کند.

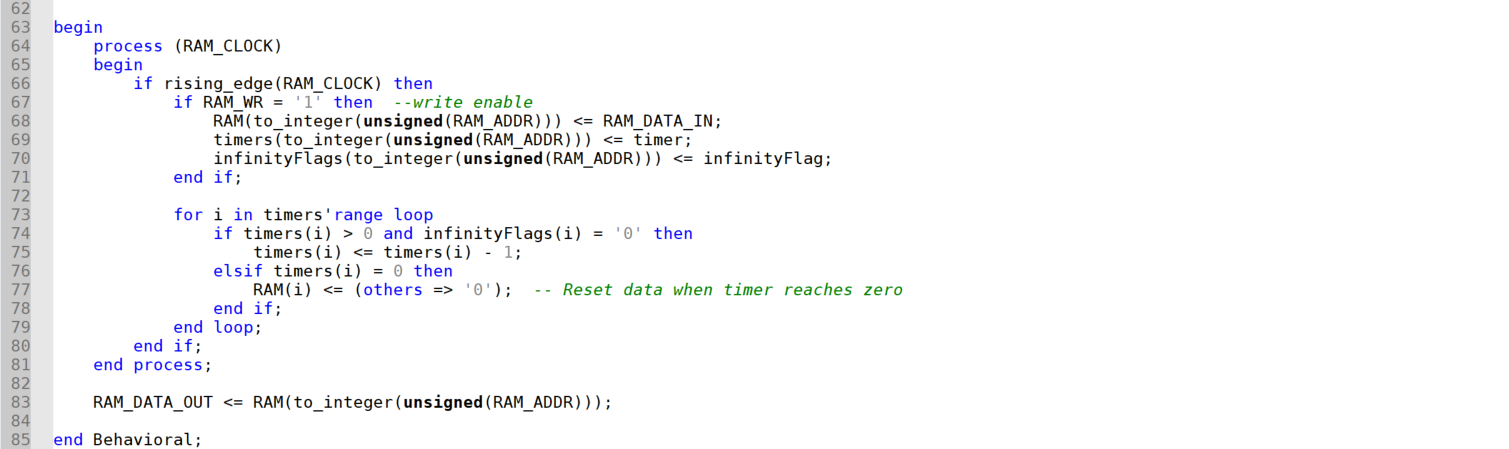
برای 3بخش گفته شده برای هر خانه در architectureمان تایپ های متناظری باید تعریف کنیم: 

* **RAM\_ARRAY** یک نوع آرایه که حافظه RAM را با ۱۲۸ عنصر نشان می‌دهد، هر عنصر یک std\_logic\_vector ۸ بیتی است.
* **TIMER\_ARRAY** آرایه ای از unsigned که تایمرها را برای هریک از ۱۲۸ عنصر ذخیره میکند.
* **INFINITY\_ARRAY** آرایه که با 0 و 1 برای 128 عنصر مشخص میکند که آیا با شروع به تمام شدن تایمرش کند و با به 0 رسیدن آن نابود شود یا نه.

در ادامه سیگنال هایی از تایپ های گفته شده میسازیم و تمام بیت های آن ها را با 0 در ابتدا مقداردهی میکنیم:



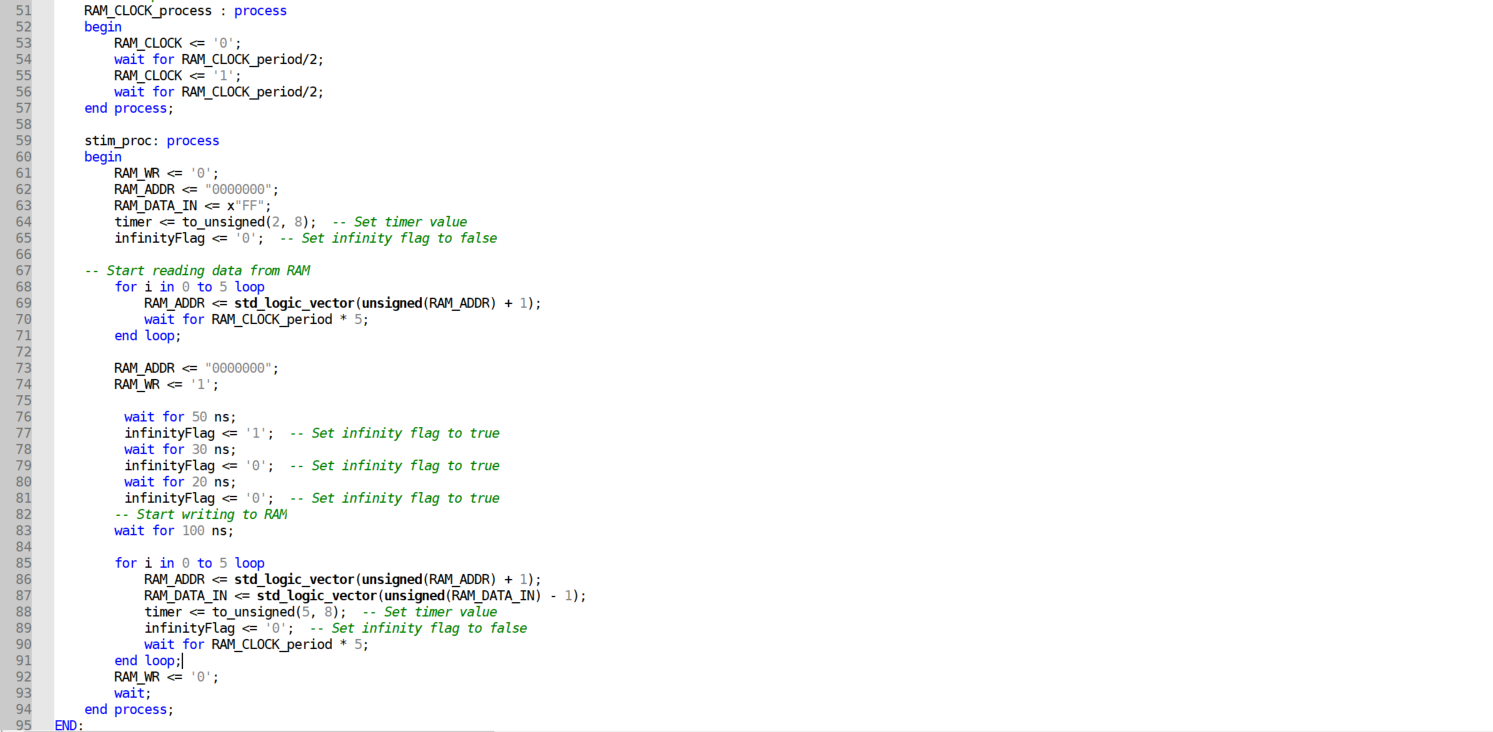
حال به بخش اصلی قطعه مان میرسیم، جایی که منطقش را تعریف کرده ایم:

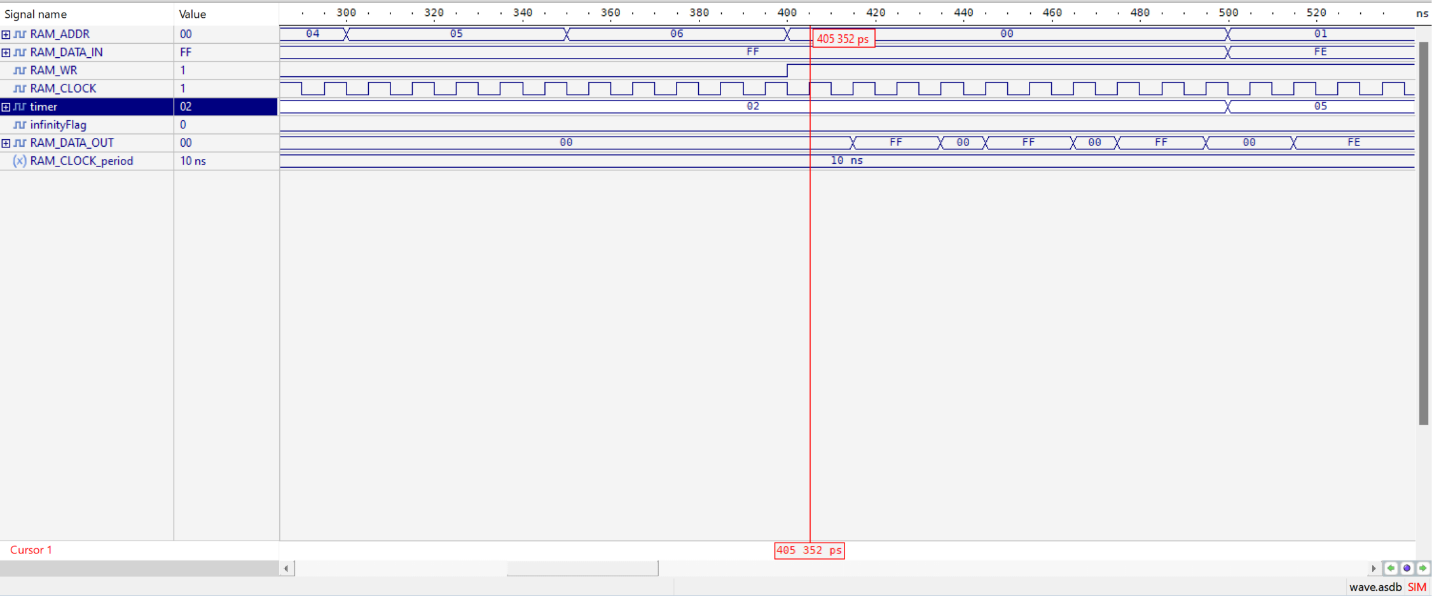


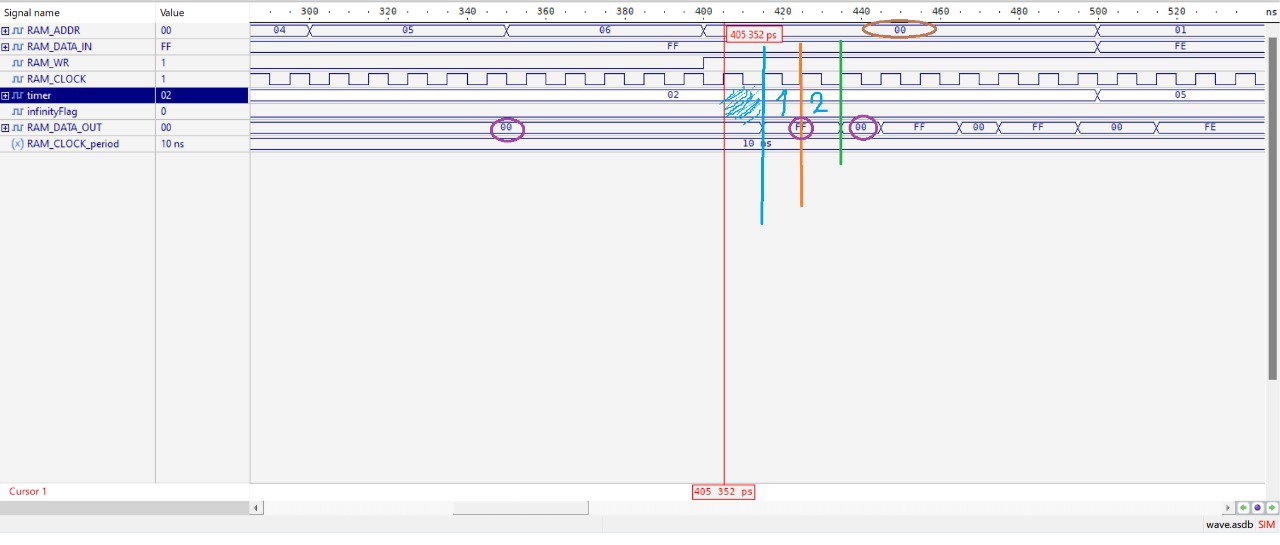
در هر کلاک بالارونده قطعه در خط 67 تا 70 چک میکند که اگر دیتای جدیدی باید وارد شود(با RAM\_WR)، آنرا و تایمر و infinityFlag مربوطه اش وارد رم میکند.

در خط 73 تا 79 هم برای 128خانه چک میکند که اگر infinityFlagشان 0 بود و تایمر هنوز به 0 نرسیده بود، از تایمر متناظر آن ها 1 واحد کم کند و اگر تایمر خانه ای 0 شد باید خانه ی آن را در رم 0 کند در RAM مان دیگر جای آن دیتا "00000000" خواهیم داشت، infinityFlagش تغیری نمیکند و تایمرش هم قاعتا به 0 رسیده و آن را هم تغیری نمیدهیم، چرا که نیازی نیست. همچنین در حالتی که infinityFlag خانه ای 1 بود کلا با آن سر این قضیه کاری نداریم.

نمونه از تستی که انجام داده ایم با کد مربوط بهش:







در این عکس که wave form ماست اومدیم تایمر رو برابر ۲ قرار دادیم و بعدش از زمانی که FF رو به یک خونه از رم assign میکنیم دو تا کلاک میگذرد و آن را ریست میکند.