به نام خداوند بخشنده و مهربان

گزارش پروژه cache

در این پروژه از ماژول های زیر استفاده شده:

1. Data\_arr : این ماژول برای نگهداری داده ها در کش هستند. زمانی که wren یک باشد در لبه بالارونده کلاک داده موجود در wrdata را در خانه ای از کش که address ان را مشخص می کند می نویسد. این ادرس در اصل 6 بیت کم ارزش ادرس درخواستی از طرف cpu می باشد یا به عبارتی همان index . برای خواندن از این ماژول نیازی به کلاک نیست.
2. Tag\_valid\_arr : در این ماژول بخش tag ادرس ذخیره می شود. در اصل همان 4 بیت پر ارزش ادرس درخواستی. و یک بیت و یک بیت valid که مشخص می کند ایا داده موجود در ان ادرس کش معتبر می باشد یا نه. (مثلا زمانی که هنوز کش پر نشده این داده نامعتبر است.) زمانی که wren یک باشد در لبه پایین رونده کلاک داده را در خانه ای از کش که ادرس مشخص می کند می نویسد. اگر سیگنال validate یک شود در لبه پایین رونده کلاک بیت valid ان خانه یک می شود. برای خواندن از این ماژول نیازی به کلاک نیست.
3. Hit\_miss\_logic : این ماژول برای مشخص کردن اینکه داده درخواستی ایا در کش موجود است یا نه استفاده می شود. (و اینکه در کدام way قرار دارد). ورودی این ماژول w0 و w1 (خروجی tag\_valid\_array های way0 و way1 می باشد) و همین طور tag (که همان 4 بیت پرارزش ادرس درخواستی است). در صورتی که tag در هرکدام از ماژول ها موجود باشد و داده ان هم معتبر باشد این ماژول خروجی w0\_valid و یا w1\_valid را یک می کند و همین طور خروجی hit را این ماژول نیازی به کلاک ندارد.
4. Mru : از این ماژول زمانی استفاده می شود که می خواهیم در کش بنویسیم و خانه های مشخص شده توسط ادرس way0 و way1 هر دو پر هستند. این ماژول مشخص می کند که اخرین استفاده از کدام یک از way ها بوده و در نتیجه در ان way نوشته می شود. این عمل به این صورت انجام می شود که زمانی که w0\_en و یا w1\_en یک شوند در لبه کلاک خانه مربوطه 1 می شود و خانه در مقابل ان 0 می شود. و به این روش که هربار که خواستیم از کش بخوانیم یا در ان بنویسیم سیگنال مربوط به ان way را یک می کنیم تا مشخص شود اخرین چیزی که از ان استفاد شده ان way بوده. خواندن از این ماژل نیازی به کلاک ندارد.
5. Controller : دارای 3 استیت است و در صورت نیاز تغییر استیت این اتفاق در لبه پایین رونده کلاک می افتد.

S0 : در این استیت همه سیگنال های خروجی 0 می شوند و در صورت یک بودن سیگنال iread و یا iwrite به S1 می رویم.

S1 : در این استیت در صورت hit شدن داده و 1 بودن iread همان جا dataready 1 میشود و داده از کش در خروجی قرار دارد استیت بعدی S0 می باشد. در صورت hit نشدن و 1 بودن iread سیگنال mem\_on\_data\_arr یک می شود تا خروجی مموری در ورودی data\_arr قرار بگیرد. همین طور سیگنال های readmem و mem\_on\_out هم یک می شوند تا داده از مموری خوانده شود و در خروجی قرار بگیرد استیت بعدی S2 در نظر گرفته می شود. اگر iwrite 1 بود سیگنال writemem 1 می شود تا داده در مموری نوشته شود. همین طور بر اگر هیچ way ای خالی بود که در ان و اگر خالی نبود با استفاده از mru مشخص می شود که در کدام way کش نوشته شود استیت بعدی S2 می شود.

S2 : در صورتی که iread 1 باشد به این معناست که داده درخواستی hit نشده بوده در خروجی قرار دارد(چون از مموری امده) پس dataready 1 می شود و سیگنال های مربوط به نوشته شدن در کش هم ست می شوند. استیت بعدی S0 در نظر گرفته می شود.

1. Cache : این ماژول عملا datapath می باشد و همه ماژول ها را به اضافه کنترلر به هم وصل کرده. تعدادی mux و بافر سه حالته نیز برای موارد مورد نیاز در این ماژول در نظر گرفته شده.
2. Cache\_mem : در این ماژول کش به مموری متصل شده و ماژول نهایی می باشد.