

توضیح پروژه:

این مدار یک Bus configuration پایه است که هفت دستور ارجاع به حافظه در کامپیوتر پایه را تولید میکند و با استفاده از این هفت دستور میتوان مجموعه دستور العمل‌های مختلف را پیاده سازی کرد. یعنی در واقع این معماری صرفاً یک بستر برای پیاده سازی مجموعه دستور العمل‌های ارجاع به حافظه است و مشاهده کارکرد حقیقی کد نوشته شده پس از پیاده کردن یک مجموعه دستور العمل با بهره بردن از این دستورات امکان پذیر است. به این صورت که کاربر میتواند ماژول cpu در این کد را در ماژول خود فراخوانی کرده و گام به گام با مقدار دهی مناسب به سیگنال‌های این ماژول و ماژول خود برنامه ای در سطح سخت افزار نوشته و نتیجه را مشاهده کند.

نکته مهم: ما برای نوشتن این کد از دو مبحثی استفاده کردیم که در دانشگاه توسط استاد طاهری تدریس نشده است. شرح این دو مبحث:

1- non - blocking assignment (انتصاب غیر مسدود کننده): هنگامی که حساسیت بلاک always بر روی کلاک باشد، درون بلاک اگر برای انتصاب از $=$ به جای $=$ استفاده کنیم، تمام انتصابها به صورت موازی در لبه یک کلاک انجام خواهد شد. یک مثال از تفاوت non - blocking assignment و blocking assignment

$$a = a + 1$$

$$b = a$$

ابتدا a یک واحد اضافه میشود سپس نتیجه نهایی به b منصوب می شود.

$$a \leq a + 1$$

$$b \leq a$$

این دو عملیات همزمان در یک کلاک انجام میشوند. در نتیجه پس از صرف یک کلاک، a یک واحد زیاد میشود و b مقدار پیشین a (پیش از آنکه یک واحد زیاد شود) را می گیرد.

الحاق:

می توانیم چند سیگنال را به صورت موقت کنار هم قرار دهیم و از آنها استفاده کنیم و سپس آنها را به حالت پیشین خود بازگردانیم. سیگنال‌ها درون آکولاد نوشته میشوند و با کاما از هم جدا می شوند. می‌توانیم سیگنال ها را با ترتیب و جایگاه دلخواه مقدار دهی کنیم. سیگنال ها می توانند متغیرهایی باشند که از پیش به آنها مقدار دادیم یا میتوانند بیت یا بیت هایی خارج از متغیرهای تعریف شده باشند. الحاق بکار رفته در این کد : `mem_address<={4b0000 , IR_address}`

کد خط بالا به این معنی است که ۱۲ بیت IR_address در سمت راست mem_address قرار بگیرد و چهار بیت سمت چپ آن با ۴ تا صفر پر شود.

لینک گیت هاب پروژه: <https://github.com/aref-mipr/base-Busconfiguration.git>