

برای انجام این فاز ابتدا در ماژول Data\_Path تمامی مقادیر لازم را تعریف کردیم. از آن جایی که مقادیر رجیسترها با کلاک عوض می‌شود، آن دسته از مقادیر که به رجیسترها متصل بودند، به صورت reg و بقیه مقادیر را به صورت wire تعریف نمودیم. سپس ماژول‌های را که از فازهای قبلی موجود بودند، با کمی تغییر برای استفاده در این فاز مناسب کردیم و در ادامه ماژول Hazard\_Unit را ساختیم. در این مرحله به اتصال قسمت‌های مختلف به یکدیگر و تکمیل ماژول Data\_Path پرداختیم.

مشکلات موجود در نتیجه‌گیری از کد:

۱- در ابتدا با شروع تست پنج همه مقادیر ماژول Data\_Path، x می‌شدند و در نتیجه فقط دستور اول انجام می‌شد. علت این بود که بلاک initial فقط مقادیر reg را صفر می‌کرد و با رسیدن کلاک، دوباره مقدار x در تمامی مقادیر نوشته می‌شد. برای حل این مشکل علاوه بر بلاک initial از مقدار Reset نیز برای صفر کردن اولیه مقادیر استفاده می‌کنیم.

۲- مشکل بعدی در اجرای دستورات beq و bne بود زیرا تا زمان تصمیم‌گیری برای انجام دادن یا ندادن branch، دو دستور بعدی fetch می‌شدند و در نتیجه در صورت انجام شدن branch، رجیسترهای Decode و Execution، می‌بایست flush می‌شدند. اما در Data\_Path داده شده، برای این منظور سیگنالی در نظر گرفته نشده بود. برای حل این مشکل از مقدار PCSrcE استفاده کردیم؛ به این صورت که اگر در زمان رسیدن کلاک، مقدار آن ۱ باشد، رجیسترهای Decode و Execution، flush می‌شوند.

۳- از آن جایی که این Data\_Path برای تصمیم‌گیری در مورد دستورات branch، دو کلاک زمان لازم دارد، قبل از این که در مورد دستور branch یکی مونده به آخر تصمیم‌گیری شود، مقدار PC به مقدار نهایی خود می‌رسید و برنامه خاتمه می‌یافت. برای حل این مشکل دستور آخر را حذف کردیم و به جای آن از ۵ دستور nop استفاده نمودیم.

تاخیر در خواندن و نوشتن Memory:

یک راه ابتدایی استفاده از stall است. همواره می‌توان به اندازه تعداد کلاکی که Memory برای خواندن یا نوشتن نیاز دارد، با استفاده از stall، پروسسور را متوقف کرد. راه‌حل‌های بهتری نیز برای این موضوع وجود دارد. به عنوان مثال می‌توان از دو پورت برای نوشتن در Memory استفاده کرد. روشی دیگر استفاده از حافظه‌ای برای Memory است تا در زمانی که درصد دستوراتی که با Memory کار دارند، زیاد است، این دستورات در آن حافظه ذخیره شوند و در زمانی که درصد این دستورات کمتر است به ترتیب در Memory نوشته شوند. این کار باعث می‌شود زمان میانگین خواندن و نوشتن Memory کاهش یابد. روش دیگر نیز data forwarding است.