**تمرین ششم** تاریخ تحویل: ۶ تیر

# ۱. طراحی سلسله‌مراتب حافظه

## هدف تمرین

هدف از این تمرین، درک عمیق تأثیر سلسله‌مراتب حافظه در پردازنده است. این تمرین از طراحی دو قسمت تشکیل شده است: طراحی حافظه نهان و به کارگیری آن در پردازنده.

در این تمرین شما باید یک حافظه نهان سطح اول [1] با نگاشت مستقیم [2] را طراحی کرده و آن را به یک حافظه اصلی [3] و پردازنده (طراحی شده در پروژه قبلی) متصل کنید. پردازنده پیش از مراجعه به حافظه اصلی، ابتدا با حافظه نهان شما تعامل خواهد داشت. پیاده‌سازی دقیق منطق تشخیص برخورد [4]، عدم برخورد [5] و جایگزینی داده‌ها، هدف اصلی این تمرین است. در این فرایند، شما باید با چالش‌های مربوط به مدیریت حافظه نهان و نحوه تأثیر آن بر عملکرد پردازنده روبرو شوید.

## طراحی حافظه نهان

### مشخصات ورودی‌ها و خروجی‌ها

ورودی‌ها:

- `Clk`: سیگنال ساعت
- `Rst`: سیگنال بازنشانی
- `CPU_DataIn`: داده دریافتی از پردازنده (۳۲ بیت)
- `CPU_Addr`: نشانی دریافتی از پردازنده (۳۲ بیت - مضرب ۴)
- `CPU_Write`: دستور نوشتن از پردازنده
- `CPU_Read`: دستور خواندن از پردازنده
- `MM_DataIn`: داده دریافتی از حافظه اصلی (۱۲۸ بیت)
- `MM_Done`: کامل شدن تراکنش حافظه اصلی

خروجی‌ها:

- `CPU_DataOut`: داده ارسالی به پردازنده (۳۲ بیت)

[1] L1 Cache
[2] Direct-Mapped
[3] Main Memory
[4] Hit
[5] Miss

- CPU_Done: کامل شدن تراکنش حافظه نهان
- MM_Write: ارسال دستور نوشتن به حافظه اصلی
- MM_Read: ارسال دستور خواندن به حافظه اصلی
- MM_Addr: نشانی ارسالی به حافظه اصلی (۳۲ بیت - مضرب ۱۶)
- MM_DataOut: داده ارسالی به حافظه اصلی (۱۲۸ بیت)

### منطق پیاده‌سازی

در این بخش، ما یک حافظه نهان با نگاشت مستقیم پیاده‌سازی خواهیم کرد. در این سیستم، پردازنده قبل از مراجعه به حافظه اصلی، ابتدا از حافظه نهان استفاده می‌کند. منطق حافظه نهان به این صورت است که ابتدا نشانی دریافتی از پردازنده به سه بخش Tag، Index و Byte Offset تقسیم می‌شود. سپس از بخش Index برای پیدا کردن بلاک مورد نظر در حافظه نهان استفاده می‌شود و مقدار Tag با تگ ذخیره‌شده در آرایه تگ مقایسه می‌گردد تا مشخص شود داده‌ی مورد نظر در حافظه نهان موجود است یا خیر. اگر داده در حافظه نهان موجود باشد (Cache Hit)، داده مستقیماً به پردازنده ارسال می‌شود. در غیر این صورت، یعنی در حالت عدم وجود داده (Cache Miss)، بلاک حاوی داده ازحافظه اصلی خوانده شده و در حافظه نهان ذخیره می‌شود. همچنین در این طراحی، از سیاست هم نویسی(Write-Through) به عنوان سیاست نوشتن استفاده کنید.

### نحوه آزمون

آزمون این حافظه نهان به این صورت است که تعدادی درخواست نوشتن و خواندن به این واحد داده می‌شود. سپس عملکرد درست آن بر مبنای تأخیر در پاسخگویی و درخواست‌های نوشتن به حافظه اصلی ارزیابی می‌شود. همچنین توجه داشته باشید که در هنگام عدم آماده بودن داده درخواست‌شده از حافظه اصلی، مقادیر تصادفی روی MM_DataIn قرار می‌گیرد.

## اتصال به پردازنده

ورودی‌ها:

- Clk: سیگنال ساعت
- Rst: سیگنال بازنشانی
- InstrMM_DataIn: دستور خوانده شده (۱۲۸ بیت)
- InstrMM_Done: کامل شدن تراکنش حافظه دستورات
- DataMM_DataIn: داده خوانده شده (۱۲۸ بیت)
- DataMM_Done: کامل شدن تراکنش حافظه داده

خروجی‌ها:

- R0، R2، ...،R31: ۳۲ ثبات ۳۲ بیتی
- InstrMM_Addr: نشانی دستور (۳۲ بیت - مضرب ۱۶)
- InstrMM_Read: دستور خواندن از حافظه دستورات
- DataMM_Addr: نشانی داده (۳۲ بیت - مضرب ۱۶)
- DataMM_Read: دستور خواندن از حافظه داده
- DataMM_Write: دستور نوشتن به حافظه داده
- DataMM_DataOut: داده در حال نوشته شدن به حافظه (۱۲۸ بیت)

**منطق پیاده‌سازی**

باید پردازنده پیاده‌سازی‌شده در تمرین قبلی را به حافظه نهان طراحی شده متصل کنید.

**نحوه آزمون**

پردازنده شما توسط کد تمرین ۵ مورد ارزیابی قرار می‌گیرد. در ابتدا مدار در وضعیت بازنشانی قرار می‌گیرد. سپس در حافظه مربوطه، دستورات مورد نظر قرار گرفته و حافظه داده ۰ می‌شود. در نهایت مدار از وضعیت بازنشانی خارج شده و پردازنده باید از همان چرخه ساعت شروع به کار کند. از این لحظه وضعیت ثبات‌های پردازنده شما با یک پردازنده معیار مقایسه‌شده و نمره‌دهی می‌شود.

## خروجی‌های مورد انتظار برای تحویل

- فایل طراحی مدار در محیط Logisim (نام فایل: HW6-STUDENT_ID.circ) یا کد پیاده‌سازی مدار به زبان Verilog.
- گزارش کاملی از نحوه طراحی و اجرای آزمون ارائه کنید (نام گزارش: HW6-STUDENT_ID.pdf). توجه داشته باشید که نمره‌دهی تنها بر اساس نتایج همین آزمون انجام می‌شود.