



## آزمایش دوم: پیاده سازی پردازنده ARM

گرد آورنده: علیرضا یزدان پناه- ادریس نصیحت کن - مرضیه رستگار

## قسمت دوم آزمایش ARM: ( جلسه دوم)

در جلسه پیش مرحله واکنشی دستور را پیاده نموده اید در این جلسه مرحله کدگشایی را پیاده خواهید کرد. برای این کار به روش زیر عمل نمایید:

۱- مازول ثبات های عمومی (Registers File) را با پورت های زیر ایجاد نمایید. برای پیاده سازی آن از یک آرایه دوبعدی استفاده نمایید.

```

1
2 module RegisterFile (
3     input clk, rst,
4     input[3:0] src1, src2, Dest_wb,
5     input[31:0] Result_WB,
6     input writeBackEn,
7     output [31:0] reg1, reg2
8 );

```

۲- مازول ثبات های عمومی قابلیت خواندن همزمان از ۲ ثبات به صورت ناهمگام و نوشتن در رجیستر به صورت همگام (با لبه پایین رونده) را دارد.

۳- مازول ثبات های عمومی دارای ۱۶ ثبات ۳۲ بیتی است. ثبات ۱۵ آن خاص منظوره بوده و به عنوان شمارنده دستورات مورد استفاده قرار می گیرد برای سادگی کار از مازول ثبات های عمومی خارج شده است بنابراین مازول ثبات های عمومی را با ۱۵ رجیستر پیاده سازی کنید. همه ثبات ها را مطابق با شماره آن مقداردهی اولیه نمایید.

۴- مازول Control Unit را ایجاد نمایید. این مازول دو بیت mode و ۴ بیت Op-code به همراه بیت S را دریافت می کند و سیگنال های خروجی شامل این موارد است:

الف) Execute Command (ریز دستوره های واحد حساب و منطق مطابق جدول ۵ از دستور کار ARM).

ب) سیگنال های مرحله حافظه شامل خواندن از حافظه mem\_read و نوشتن در حافظه mem\_write.

ج) سیگنال مربوط به فعال سازی مرحله باز نشانی WB\_Enable.

د) سیگنال مربوط به دستور پرش B

هـ) سیگنال مربوط به به روزرسانی ثبات وضعیت. این سیگنال در تمامی دستورات محسابی منطقی (mode=0) برابر ورودی S خواهد بود.

توجه: در صورت استفاده مستقیم از کد دستور (op-code) در هر مازولی بجز Control Unit، نمره کسر می گردد.

۵- مازول Condition\_Check را تعریف کنید. این مازول بیت های مربوط به شرط (cond) را از دستور ورودی را دریافت می کند و به کمک ثبات وضعیت برقرار بودن شرط را بررسی می کند. بیت های مربوط به رجیستر SR از ورودی مرحله کدگشایی دریافت می شود. شرایط برقرار بودن شرط در جدول ۳ از ARM ذکر شده است.

۶- مازول کدگشایی و رجیسترهای پس از آن را مطابق با ورودی های مشخص شده در دستور کار ARM ایجاد نمایید.

۷- از مازول ثبات های عمومی در آن نمونه گیری کنید سپس پایه های مربوط آن را متصل نمایید. ورودی src2 ثبات های عمومی به کمک سیگنال mem\_write تعیین می شود (دستور STR رجیستر مقصد را به عنوان src2 در نظر می گیرد و در دیگر دستورات Rm ورودی دوم رجیستر فایل است).

۸- از مازول control\_unit در مرحله کدگشایی نمونه گیری کنید.



## آزمایش دوم: پیاده سازی پردازنده MIPS

گرد آورنده: علیرضا یزدان پناه - ادریس نصیحت کن - مرضیه رستگار

۹- از ماژول condition\_Check در مرحله کدگذاری نمونه گیری کنید. در صورت برقرار نبودن شرط دستور تمامی خروجی های control\_unit صفر می شود بنابراین یک mux بعد از خروجی این ماژول قرار می گیرد. Selector آن خروجی condition\_Check خواهد بود.

۱۰- سیگنال های خروجی مرحله کدگذاری را متصل کنید.

۱۱- از ماژول کدگذاری و رجیستر بعد از آن در ماژول اصلی (Top Module) نمونه گیری کنید.

۱۲- سیگنال هایی که هنوز پیاده سازی نشده اند را به صفر متصل کنید.

۱۳- ۱۸ دستور اول برنامه محک را در حافظه دستورالعمل قرار دهید.

۱۴- تمامی سیگنال های کنترلی خروجی مرحله کدگذاری و رجیسترهای بعد از آن به همراه رجیسترهای Rn و Rm را در Signal-Tap نمایش دهید.

۱۵- درستی سیگنال های تولید شده را بررسی نمایید. توجه داشته باشید که این سیگنال ها باید با دستورات مطابقت داشته باشد به طور مثال سیگنال mem\_read فقط در دستور LDR باید برابر یک شود.

۱۶- مقادیر رجیسترهای Rn و Rm باید مطابق عدد رجیستر باشد زیرا براساس عدد آن مقداردهی اولیه شده است.

۱۷- در صورت درستی، نتایج را به دستیار آموزشی نشان دهید.