|  |  |  |
| --- | --- | --- |
|  | به نام خدا |  |
| **دانشگاه تهران**  **دانشکده‌ مهندسی برق و کامپیوتر**  **ESL**  **گزارش** **پروژه‌ چهارم** | | |

|  |  |
| --- | --- |
| امیرحسام جعفری راد | نام و نام خانوادگی |
| 810100247 | شماره‌ دانشجویی |
|  | تاریخ ارسال گزارش |

­

فهرست

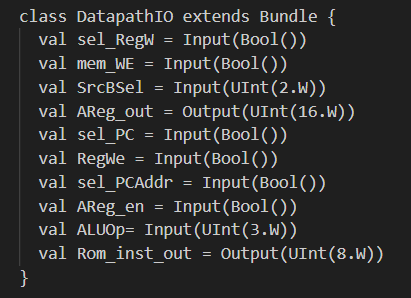
[توضیحات 2](#_Toc169463348)

[درستی سنجی طراحی 4](#_Toc169463349)

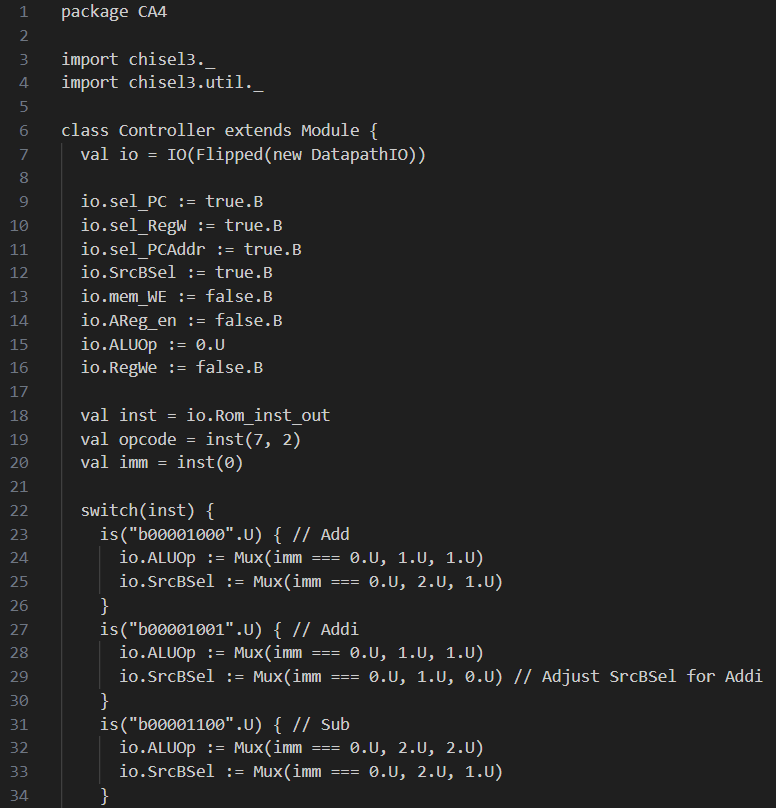
# توضیحات

برای انجام این پروژه نیز برای سهولت کار از repository خود chisel که در پروژه قبل استفاده شده بود استفاده می‌کنیم و تمامی فایل ها را در همان محل جایگذاری میکنیم. پوشه اصلی پروژه در محل chisel-tutorial\src\main\scala\CA4 ایجاد شده است. تمامی فایل هایی استفاده شده در مسیر داده نیز در این پوشه قرار داده شده اند بجز ALU که در محل پروژه قبلی بود و برای استفاده از آن، آن را در هرجا که نیاز بود با دستور روبه‌رو import می‌کنیم: import solutions.ALU

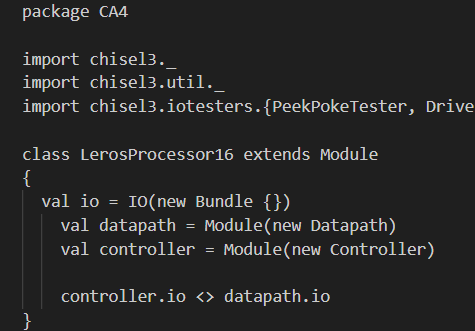
پس از نوشتن ماژول های مورد نیاز بصورت جداگانه، مسیرداده را با اینتستنس گرفتن از ماژول ها ایجاد میکنیم. برای اینکار ابتدا سیگنال های ورودی خروجی را متصل کرده و سپس بقیه مراحل را انجام می‌دهیم:



سپس ماژول کنترلر را با توجه به جدول داده شده در صورت سوال به کمک دستور switch/case مطابق چیزی که در پروژه سوم و ALU استفاده شده طراحی می‌کنیم:



در نهایت ماژول های مسیرداده و کنترلر را در top module که LoresProcessor16.scala نام دارد اینتنس میگیریم:

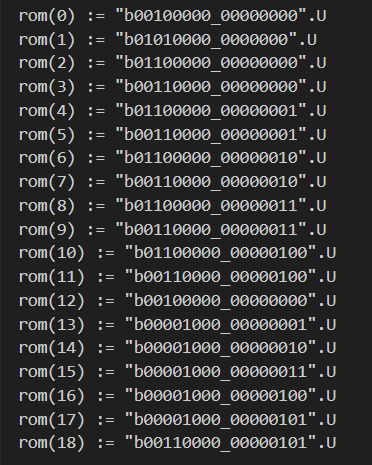


برای تست کردن این پردازنده مطابق خواسته صورت پروژه 5 عدد را در مموری گذاشته و در نهایت آن ها را به کمک instruction های مورد نیاز درون رجیسترفایل های 0 تا 4 ریخته و در رجیسترفایل با ایندکس 5، مجموع آن 5 عدد را محاسبه می‌کنیم. اعداد استفاده شده در این تست به شرح ذیل است:

حالت اول) 5, 7, 3, 42, 15

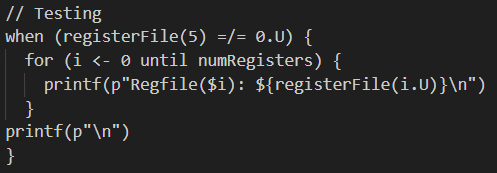
حالت دوم) 1, 2, 3, 4, 5

همچنین instruction های مورد نیاز برای این عمل به شرح ذیل می‌باشد:



همانطور که مشاهده می‌شود این کار توسط 19 دستور در این پردازنده قابل انجام می‌باشد.

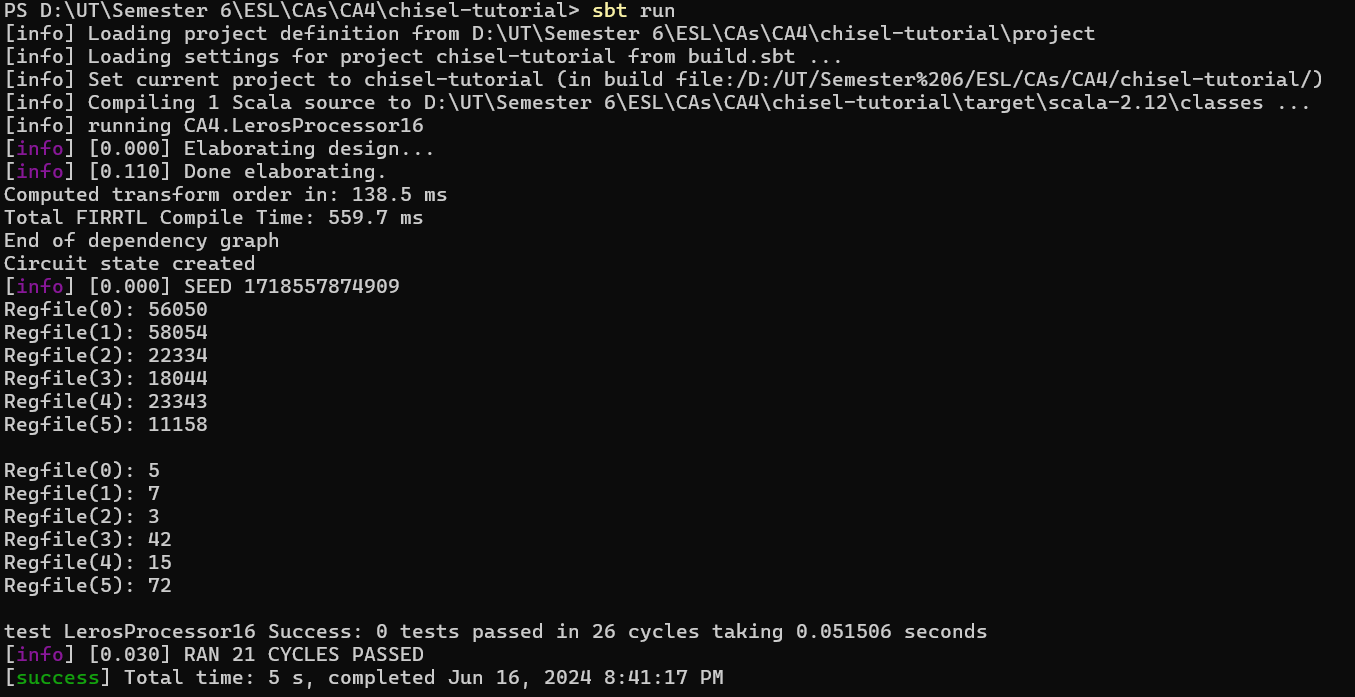
در حالت اول انتظار میرود خروجی در رجیستر با اندیس 5 قرار بگیرد و مقدارش برابر با 5 + 7 + 3 + 42 + 15 که برابر با 72 است دیده شود. جهت دیده شدن خروجی قطعه کدی در رجیسترفایل نوشته شده که تنها اگر مقدار رجیستر با اندیس 5 مخالف صفر بود (یعنی مقداری را گرفته بود)، تمامی مقادیر 6 رجیستر نمایش داده شود (5 رجیستر حاوی اعداد و رجیستر آخر حاوی حاصل جمع اعداد).



برای اجرا کردن این پروژه تنها نیاز است دستور sbt run را در محل پروژه (chisel-tutorial) اجرا کنیم.

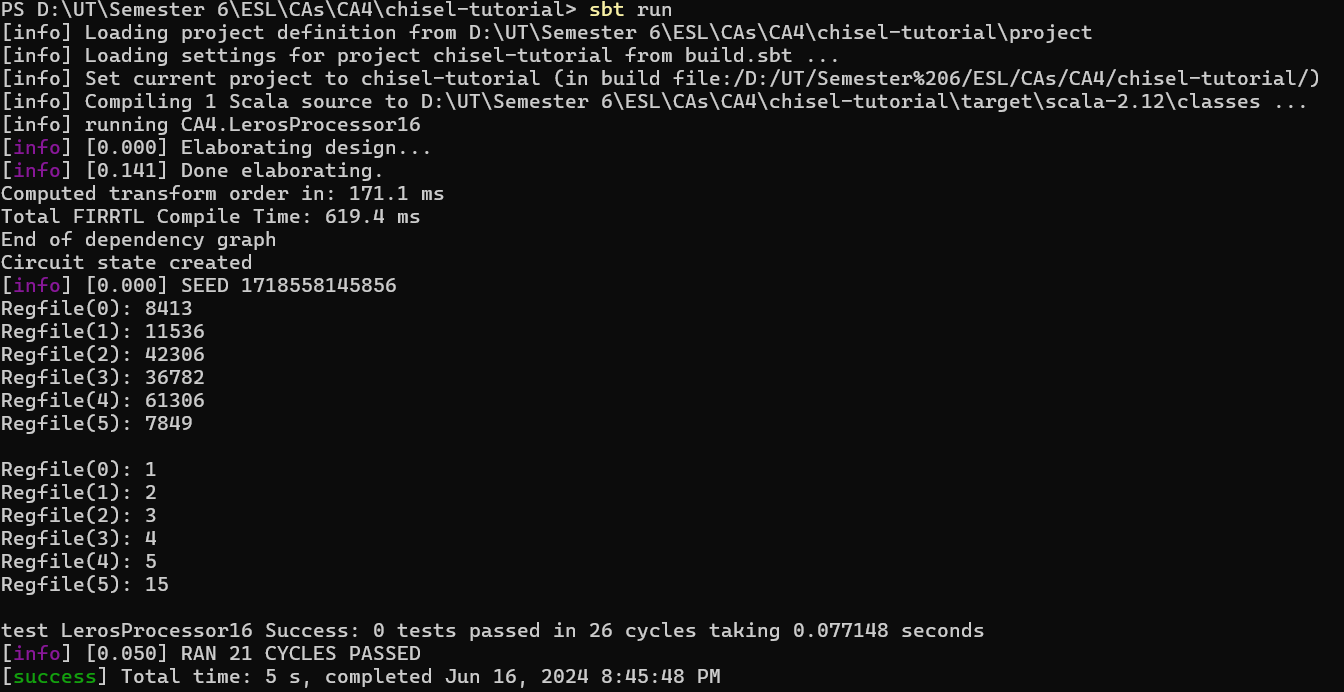
# درستی سنجی طراحی

**نتیجه تست حالت اول:**



علت اینکه بخش قرمز رنگ نمایش داده شده است این است که در لحظه اول تمامی رجیستر ها به دلیل عدم initialization مقداری غیرصفر بصورت رندوم دارند و چون مقدار ریجستر با اندیس5 مخالف صفر است مقادیر تمام ریجستر ها نمایش داده شده است. بخش آبی رنگ بخش درست و نتیجه اصلی می‌باشد که مطابق انتظار، رجیسترها مقادیر یه آرایه 5 تایی را به ترتیب گرفته و در خانه آخر مجموع آن ها نوشته شده است که برابر با 72 می‌باشد.

**نتیجه تست حالت دوم:**

در این حالت نیز انتظار میرفت که خروجی برابر با 1 + 2 + 3 + 4 + 5 یا برابر با 15 باشد که مطابق شکل این مقدار در رجیستر آخر ذخیره شده است.