

## گزارش فاز اول پروژه معماری کامپیوتر - گروه Mascarpone

محمد مهدی قیدی - ۹۸۱۰۵۹۷۶

حسین آقایی - ۹۸۱۰۵۶۱۹

میلاد سعادت - ۹۸۱۰۰۴۴۲

پرهام چاوشیان - ۹۸۱۰۰۱۱۸

در پروژه با توجه به نیازهای موجود، ماژولهای مختلفی از جمله `fetch_operands`، `i_controller`، `PC_ALU`، `j_controller`، `r_controller`، `R_type_ALU`، `I_type_ALU` تعریف کردیم که در زیر کارایی هر کدام را شرح دادیم:

ماژول `fetch_operand` به واکشی دستورات و جداسازی بخشهای مورد نیاز از ۳۲ بیت ترجمه شده به زبان ماشین مربوط به هر دستور را به ما میدهد. به طوری که دستور را به صورت کامل به این ماژول میدهیم و بخشهای `opcode`، `rd`، `rt`، `rs`، `func`، `immediate` و ... در سیگنالهای خروجی اش قرار میدهد.

همچنین در این فاز با توجه به وجود دستورات `R-Type` و `I-Type` و `J-Type` ما به ۳ ماژول جداگانه نیاز پیدا کردیم تا هر کدام کار کنترل سیگنالهای این موارد را انجام دهند. به ورودی هر کدام از این ماژولها `opcode`های مربوطه را وصل کرده ایم تا نوع دستور را تشخیص دهند. همچنین برای خروجی این دستورات، به ازای هر کدام از دستورات مربوط در آن گروه یک سیگنال وجود دارد که در صورت `true` بودن نشان میدهد آن دستور باید اجرا شود. برای مثال `opcode 001001` مربوط به دستور `addiu` است که در صورتی که این آپکد به `i-controller` داده شود سیگنال `addiu` را روشن میکند.

ماژولهای `R-type-ALU` و `I-type-ALU` نیز وظیفه انجام دستورات داده شده را دارند و اینگونه پیاده سازی شده اند که سیگنالهای خروجی از کنترلرها و همچنین داده های رجیستر/مموری به این `ALU`ها وصل میشوند و هر کدام خروجی مطلوب ما را تولید میکنند و عملیاتهای محاسباتی را انجام میدهند.

ماژول `PC_ALU` وظیفه نگهداری درست `Program Counter` را دارد. همچنین توجه داشته باشید که دستورات `beq` و `bnq` هر دو `r-type` هستند اما برای اینکه دستورات `jal` و `jalr`، متغیر `program counter` را تغییر میدهند به جای پیاده سازی منطق دستورات `beq` و `bnq` در `r-type-alu` آنها را در `pc_alu` پیاده سازی کردیم تا مقدار `program counter` همیشه درست باشد.

در ماژول mips\_core به پیاده‌سازی کار کلی پردازنده پرداخته ایم، به این شکل که ابتدا رجیسترهای خود پردازنده که به آنها نیاز پیدا خواهیم کرد مثل rd rt rs و ... و همچنین wire های مورد نیاز را تعریف کرده ایم. سپس از ماژول های تعریف شده instance گرفتیم و همانند شکل زیر سیم‌کشی های بین ماژول ها را انجام دادیم.

همچنین در مسیر پروژه به تعدادی باگ و اشکالات برخوردیم و آنها را برطرف کردیم که ذکر آنها خالی از لطف نمیباشد. به طور مثال یک موضوع بسیار عجیب که به تی‌ای هم گزارش کردیم، این بود که Makefile تغییرات فایل های v. را هندل نمی‌کرد و فقط تغییرات فایل های sv. اعمال میشدند. همین موضوع باعث شد ما حدود یک روز یک بخش مشخص و ثابت از کد را به انواع شیوه های مختلف پیاده‌سازی کنیم و کار نکند تا در آخر مشکل را متوجه شدیم و با استفاده از sudo make clean و make verify-all همه تست‌ها را پاس کردیم (و متوجه شدیم حتی کد اولیه مان درست بود!).

همچنین یک موضوع قابل توجه دیگر این بود که در کلاک اول سیگنال rst\_b برابر صفر بود که این موضوع باعث می‌شد اولین instruction را اجرا نکنیم اما پس از اینکه متوجه این باگ شدیم اولین کلاک را رد کردیم و از کلاک بعدی به اجرای دستورات پرداختیم که مشکل را برطرف کرد.

