

Computer Architecture Lab

Report #1

Implementing MIPS

By: Edris Nasihatkon , Alireza Yazdanpanah

Lab partners

Ahmad Hassani 810194302

Nooshin Taghavi 810194289

پردازنده ی طراحی شده شامل ۵ module با نام های IF ، ID ، EXE ، MEM ، WB است. چهار module اول داری دو module هستند که module اول شامل قسمت های کوچک تری هستند که وظیفه ی خاصی انجام می دهند و module دوم رجیستر بین این قسمت و قسمت بعدی است.

IF (1

- در module ، IFSub یک رجیستر PC داریم که اشاره گر به دستوری است که باید اجرا شود و یک instruction memory که شامل دستوراتی است که اجرا می شود. روند تغییر PC بدین صورت است که بعد از load شدن هر دستور چهار تا اضافه می شود و به دستور بعدی اشاره می کند، مگر آنکه دستور branch باشد. برای اینگونه دستورات یک MUX در ورودی رجیستر PC وجود دارد که برنامه به درستی اجرا شود.
- در module ، IFReg رجیستر های این قسمت و module بعدی است. که شامل PC و Instruction است.

ID (2

- در module ، IDSub شاهد چند قسمت هستیم :
 - Controller که سیگنال های کنترلی را با توجه به opcode دستور مربوطه تعیین می کند. این سیگنال ها عبارتند از :
 - WB_EN : آیا باید در register file شود؟
 - MEM_Signal : مشخص کننده ی دستورات load و store در module حافظه.
 - Branch_Type : نوع branch را مشخص می کند.
 - EXE_CMD : دستور اجرا شونده توسط ALU در module ، EXE را مشخص می کند.
 - Is_imm : آیا دستور immediate است؟
 - RegisterFile ، شامل سی و دو رجیستر سی و دو بیتی است. که وظیفه ی خواندن و نوشتن روی این رجیستر ها را بر عهده دارد. مقادیر Reg1 و Reg2 را خوانده و در صورتی که enable آن فعال شود WB_Data را در WB_Dest (Dest) می نویسد.
 - signExtend ، از عدد شانزده بیتی را با توجه به بیت آخر که بیت علامت است به سی و دو بیت extend می کند.
- همچنین دو MUX که بین Reg2 ، خروجی signExtend به عنوان val2 و مشخص کننده ی Dest وجود دارد.
- در module ، IDReg سیگنال های کنترلی و Dest ، PC ، Val1 ، Val2 ، Reg2 را در خود رجیستر می کند.

EXE (3

- در module ، EXESub شاهد چند قسمت هستیم:
 - ALU : وظیفه ی اجرای دستورات محاسباتی روی مقادیر Val1 و Val2 با توجه به سیگنال EXE_CMD را دارد.
 - AdderBranch : مقدار جدید PC برای branch را مشخص می کند.
 - ConditionCheck : با توجه به نوع branch و چک کردن شرط آن مشخص می کند که باید به دستور خاصی که PC آن توسط AdderBranch تولید می شود، branch کنیم یا خیر
- در module ، EXEReg سیگنال های WB_EN ، MEM_Signal ، PC ، Dest ، ALU_Result ، Reg2 در خود رجیستر می کند.

MEM (4

- MEMSub شامل یک حافظه همانند قسمت Register file است که تعداد رجیستر های آن بیشتر است که امکان خواندن و نوشتن روی حافظه را فراهم می کند.
- MEMReg سیگنال های WB_EN ، MEM_R_EN ، Dest ، ALU_Result ، DataMemOut را در خود رجیستر می کند.

شامل یک MUX است که تعیین می کند مقدار ALU_Result یا DataMemOut در Register file نوشته شود.

◀ Signal Tap



◀ Compilation Report

Flow Summary	
Flow Status	Successful - Tue Nov 27 09:52:24 2018
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	MIPS
Top-level Entity Name	MIPS
Family	Cyclone II
Device	EP2K35F672C6
Timing Models	Final
Total logic elements	11,733 / 33,216 (35 %)
Total combinational functions	6,102 / 33,216 (18 %)
Dedicated logic registers	9,335 / 33,216 (28 %)
Total registers	9335
Total pins	418 / 475 (88 %)
Total virtual pins	0
Total memory bits	267,776 / 483,840 (55 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

برای جلوگیری از Hazard بین دستورات لازم سه دستور NOP قرار دادیم که مجموع دستورات به صد و هفده دستور افزایش یافت.

Total logic element	Total combinational functions	Dedicated logic registers	Runtime	CPI
11733	6102	9335	342	342/117 = 2.92

◀ مشکلات

- 1) اشتباه در متصل کردن سیم ها و فراموشی تعریف wire: بررسی دقیق تر کد و استفاده از modelSim و signalTap
- 2) در دستوراتی که پرش داشتیم به مشکل برخوردیم که با بررسی دقیق تر متوجه شدیم که با گذاشتن دو NOP مشکل حل میشود و در فاز بعدی متوجه شدیم که میبایست از دستور flush استفاده کنیم.