

Computer Architecture Lab

Report #2

Implementing Hazard-Detect Unit

By: Edris Nasihatkon , Alireza Yazdanpanah

Lab partners

Ahmad Hassani 810194302

Nooshin Taghavi 810194289

در این مرحله ما یک ماژول به نام HazardDetect اضافه کردیم که وظیفه آن تشخیص رخداد مخاطره و به دنبال آن، تولید سیگنالی (freeze) برای متوقف کردن (stall) رجیستر های IF و PC است.

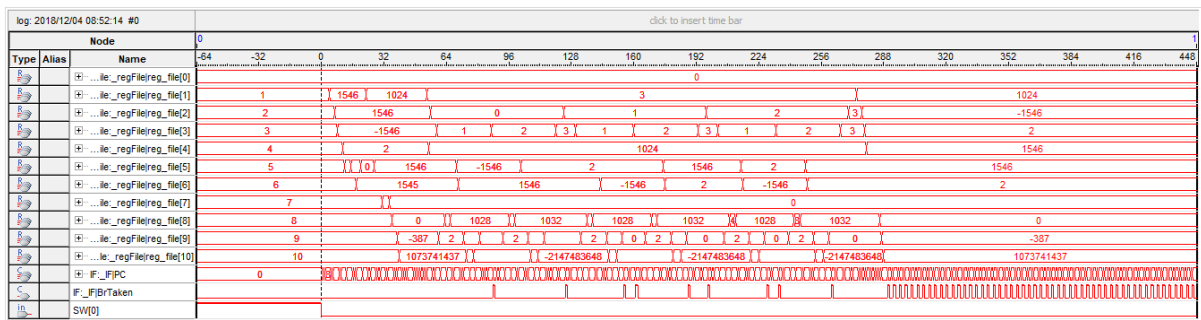
Hazard در پردازنده های in-order در صورتی رخ میدهد طی یک یا دو دستور متوالی، رجیستری دینای جدیدی بگیری و سپس خوانده شود. که چون به صورت پایپ لاین این پردازنده کار میکند، قبل اینکه داده در استیج WB ذخیره شود، داده قدیمی در استیج ID خوانده شده و برنامه طبق خواسته ما جلو نخواهد رفت.

طبق توضیحات گفته شده، Hazard-Detection unit باید شرط های زیر برای تشخیص مخاطره چک کند.

- اگر src1 دستور جدید (ID) با dest دستور قبلی (EXE) برابر باشد و نوشتن در رجیسترفایل داشته باشیم.
- اگر دستور دو source داشته باشد و src2 با dest استیج EXE برابر باشد و نوشتن در رجیسترفایل داشته باشیم.
- اگر src2 دستور جدید (ID) با dest دستور قبلی (MEM) برابر باشد و نوشتن در رجیسترفایل داشته باشیم.
- اگر دستور دو source داشته باشد و مانند بالا src2 با dest استیج MEM برابر باشد و نوشتن در رجیسترفایل داشته باشیم.

اگر یکی از این شرط ها برقرار باشد، سیگنال freeze را فعال میکنیم که رجیستر های IF و رجیستر PC به جای مقادیر جدید خود، مقادیر قبلی خود را نگه دارند. به این صورت دستور جدید بارگذاری نمیشود و به اصطلاح حبابی بین این دو دستور وارد میکنیم.

این سیگنال به استیج ID نیز میرود و selector یک مالتی پلکسر میشود که اگر 1 بود سیگنال های دستوری که بارگذاری و باعث مخاطره شده را 0 کند و این دستور به عنوان NOP جلو برود. پس از 0 شدن سیگنال freeze، این دستور دوباره از رجیستر های IF وارد ID و این دفعه مقادیر درست را از registerFile میگیرد.



شکل 1 – Signal Tap

CPI (Clock per Ins)	Total Logic Element	Total Combinational Functions	Dedicated Logic Registers	Runtime
4.64	<u>9140</u> 28%	<u>4357</u> 13%	<u>7760</u> 23%	288 Clock

جدول 1 – اطلاعات سنتر و اجرای دستورات

Flow Summary	
Flow Status	Successful - Tue Dec 04 08:49:35 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	EP2C35F672C6
Timing Models	Final
Total logic elements	9,140 / 33,216 (28 %)
Total combinational functions	4,357 / 33,216 (13 %)
Dedicated logic registers	7,760 / 33,216 (23 %)
Total registers	7760
Total pins	418 / 475 (88 %)
Total virtual pins	0
Total memory bits	200,192 / 483,840 (41 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

شکل 2 – اطلاعات سنتر (Flow Summary)

این پروژه نسبت به پروژه قبل، دستورات کمتری دارد (تقریباً نصف) زیرا دیگر احتیاجی به دستور NOP بین دو دستوری که باعث hazard میشود نیست. پس همان 62 دستور را داریم. مدت زمان اجرا نیز کمتر شده به این دلیل که در پروژه قبل بین دستورات دارای مخاطره 3 دستور NOP وارد کردیم (در حالیکه 2 دستور NOP کافی بود). پس برای هر مخاطره، 3 کلاک اضافه داشتیم. ولی در این پروژه برای هر مخاطره، 1 یا 2 کلاک اضافه داشتیم که باعث شد زمان اجرا حدود 100 کلاک کمتر شود.