بسم الله الرحمن الرحیم

دانشگاه صنعتی شریف

دانشکده مهندسی برق

فاز سوم پروژه ساختار کامپیوتر و میکروپروسسور

طراحی پروسسور پایپ لاین(pipeline)

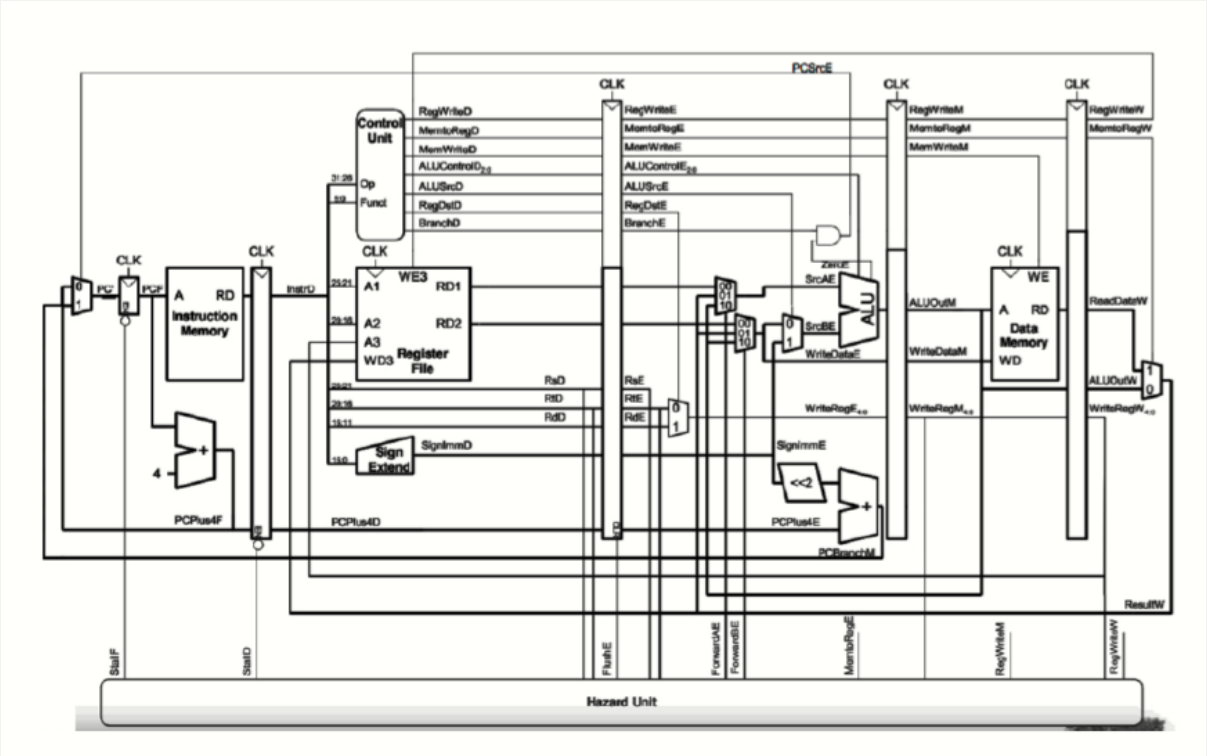
استاد : دکتر موحدیان

حسن طلائیان 95105708

بهار 97

1) شکل DataPath

در پیاده سازی و توصیف ساختار پروسسور با زبان وریلاگ سعی شد که DataPath مطابق شکل زیر باشد اما برخی تفاوت هایی در آن دیده میشود که در زیر لیست شده است.[1] کنترلر این پردازنده بصورت متمرکز میباشد.



تفاوت ها :

پورت های بین دو ماژول واحد هازارد و DataPath با این شکل متفاوت است و شامل موارد زیر است:

- ***stalF***
- ***stalD***
- ***flushE***
- ***RsD***
- ***RtD***
- ***RtE***
- ***forwardAE***
- ***forwardBE***

[1] تمامی این تفاوت ها مربوط به پیاده سازی ماژول هازارد و ارتباط آن با DataPath است. مگر در مورد OPE و وجود یک گیت not برای پیاده سازی دستور bne. مقدار OPE برابر با OP دستورالعمل در حال اجرا در قسمت execution میباشد.

- ***RegWriteM***
- ***WriteRegM***
- ***WriteRegE***
- ***LWE*** [2]
- ***RegWriteE***
- ***tontbE*** [3]
- ***tontbM***
- ***clk***

پیاده سازی ماژول Hazard Unit

این قسمت از مدار که جزوی از مدار کنترلر میباشد دو گونه از هازارد های ممکن در مدار را با مجموعه دستورالعمل های زیر شناسایی میکند و کنترل میکند.[4]

1. Data Hazard
2. Control Hazard

- قسمت کنترل Data Hazard بصورت ترتیبی و حساس به لبه بالارونده کلاک است و شامل دو بیت کنترلی forwardAE و forwardBE میباشد.[5]
- قسمت کنترل Control Hazard بصورت ترکیبی میباشد و شامل بیت های کنترلی ***stalD*** , ***flushE*** و ***stalF*** میباشد.
- در مورد LW، در صورت شناسایی هازارد یک stall رخ میدهد.
- در مورد branch، تشخیص در مرحله اجرا انجام میشود و در صورت taken بودن branch، 2 وقفه(stall) در اجرای دستورها ایجاد میشود.

[2] Load Word in execution stage (sets as LW is being executed)
[3] Taken or not taken branch in execution stage (sets when branch condition has been occurred)
[4] خوشبختانه این طراحی دارای structural hazard نمیباشد.
[5] رجیستر فایل به گونه ای تغییر کرد که در لبه پایین رونده کلاک داده های مربوط به آن نوشته شوند بنابراین حداکثر فاصله بین دو دستور که امکان هازارد بین آن دو وجود دارد 2 است.

مشکلات در حین انجام پروژه

1. در طراحی واحد هازارد، تشخیص اینکه مدار ترتیبی یا ترکیبی باید باشد. در واقع ابتدا تصور میشد مداری کاملا ترکیبی است. ابتدا بصورت کاملا ترکیبی و سپس کاملا ترتیبی پیاده سازی شد. با تشخیص خطا در اجرا که بسیار هم سخت بود، مدار ایده آل طراحی گردید.
2. خطاهای انسانی مربوط به متن کد مانند حروف کوچک و بزرگ در زبان انگلیسی
3. در طراحی روشی برای رفع باگ پردازنده طراحی شده

ساخت پردازنده با memory بسیار کند

اگر بخواهیم پردازنده ای بسازیم که سرعت آن بسیار بیشتر از memory باشد به گونه ای ک برای هر خواندن و نوشتن پردازنده را برای چندین کلاک از ادامه کار بازدارد میتوانیم :

1. فرض کنید طبقه مربوط به memory از ساختار pipeline را حذف کنیم و به گونه ای دسترسی به این قسمت فقط برای دو دستور LW و SW باشد مشکل اصلی وجود structural hazard خواهد بود که برای رفع آن لازم است کامپایلری بسیار هوشمندتر این کد را اسمبل کند.
2. میتوانیم یک یونیت جدا برای دو دستور LW , SW بگذاریم. در واقع یک پردازنده دو هسته که یکی از هسته ها همواره دستورات را جلوتر از دیگری خوانده و در صورت لزوم با حافظه کار کند که البته ساختار آن بسیار ساده تر از هسته ی اصلی و سرعت آن در اجرا کمتر است.