ارسلان فیروزی ۹۷۱۰۲۲۲۵ به نام خدا

تمام واحد های پیشین دوباره پیاده سازی شدند، برخی نیاز به اصلاح داشتند و تغییر داده شدند.

۱. واحد Main_PipeLine شامل سه واحد است که جدید طراحی شده اند: Datapath_PipeLine و Control_PipeLine و Hazard_PipeLine

در قسمت کنترلر از کنترلر واحد SingleCycle استفاده شده است.

بخش امتیازی در پوشه Optional است و گزارش آن در ته این گزارش آمده است.

۲. Data Forwarding:

برای سه حالت هازارد باید دیتا فوروارد کرد:

a) وابستگی به سه دستور قبل که از طریق نوشتن در لبه پایین رونده در رجیستر فایل و خواندن در لبه بالارونده کلاک انجام شود پس نیاز به فوروارد با این روش نیست.

b) وابستگی به یک دستور قبل از طریق فوروارد دیتا به stage محاسبه از طریق ALU انجام می شود.

حال اگر دستور قبل LW نبود، میتوان از طریق خروجی aluoutm دیتا را فوروارد کرد.

اگر دستور قبل LW بود، یک پالس کلاک stall میکنیم و وابستگی عینا مانند دو دستور قبل می شود. که در قسمت بعد توضیح دادم.

c) اگر به دو دستور قبل وابستگی وجود داشت، در execute stage داده را از write back stage فوروارد میکنیم.

در تمام حالت های بالا سیگنال های کنترلی توسط واحد هازارد تامین شده است.

یک مورد دیگر Data Forwarding وجود دارد: برای اینکه بتوان branch را با کمترین Delay Slot (در اصل با تاخیر یک دستور) اعمال کرد لازم است در صورت تشخیص وابستگی به یک دستور قبل باید یکبار stall کرد و در صورت تشخیص به وابستگی در دو دستور قبل اگر LW نبود آن را از stage Memory (aluoutm) فوروارد کرد و اگر LW بود یکبار stall کرد(شرایط مشابه وابستگی به سه دستور قبل شود) و در صورت وابستگی به سه دستور قبل مانند حالت قبل به دلیل نوشتن و خواندن در لبه های متفاوت پالس ساعت، مشکل برطرف می شود.

۳. در تست ISORT32 پس از گذشت 371270 نانوثانیه برنامه اجرا شد یعنی در ۳۷۱۲۷ پالس کلاک انجام شد. در مقایسه با Single_Cycle در صورت نبود هازارد انتظار داشتیم که در همان تعداد پالس کلاک (به اضافه ی ۴) بدست بیاید اما با وجود هازارد از ۲۸۸۳۵ بار پالس کلاک به ۳۷۱۲۷ پالس کلاک رسیدیم.

از لحاظ زمانی این مقایسه صحیح نیست چرا که T_{cp} در پایپ لاین به مراتب کوچک تر است بطوریکه در کل بازدهی افزایش می یابد.

۴. در تست Fib پس از گذشت ۱۵۷۰ نانوثانیه برنامه اجرا شد یعنی در ۱۵۷ پالس کلاک انجام شد.
در مقایسه با Single_Cycle در صورت نبود هازارد انتظار داشتیم که در همان تعداد پالس کلاک (به اضافه ی ۴) بدست بیاید اما با وجود هازارد از 126 بار پالس کلاک به 157 پالس کلاک رسیدیم.
از لحاظ زمانی این مقایسه صحیح نیست چرا که T_{cp} در پایپ لاین به مراتب کوچک تر است بطوریکه در کل بازدهی افزایش می یابد.

۵. در صورتیکه تعداد بیش تر از یک کلاک برای واحد مموری نیاز باشد، باید در صورت تشخیص دادن LW پایپ لاین را به تعداد مورد نیاز stall کرد و در غیر این صورت پایپ لاین عادی کار کند.
فرض من بر این است که در SW مشکلی نداریم و تنها در خواندن تاخیر زیادی داریم.

بخش امتیازی:

همین ایده ی بالا را پیاده سازی کردم و به نتیجه ی زیر رسیدم:

a) در تست ISORT32 با تعداد کلاک ۴۵۶۲۹ به ازای ND = 3 اجرا شد. چرا که دستورات زیادی برای خواندن از حافظه وجود داشت.

b) در تست Fib با تعداد کلاک ۱۵۷ اجرا شد. چون هیچ دستوری برای خواندن از حافظه وجود ندارد.