بسمه تعالی

گزارش کار پروژه معماری کامپیوتر

دانشگاه صنعتی شریف

استاد: دکتر سربازی

اعضای گروه:

سیدمحمدرضا خسرویان

محمدرضا بدری

سام خانکی

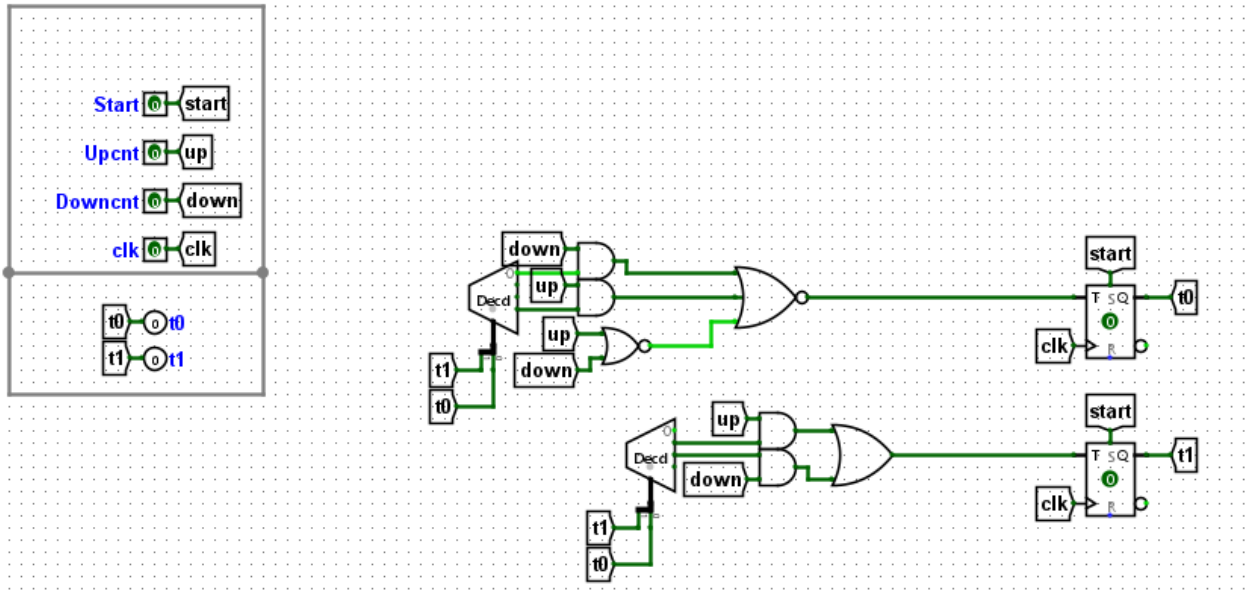
فاز اول پروژه

خرداد ۱۴۰۲

برای این فاز از پروژه کافی است branch prediction از نوع saturation را پیاده سازی کنیم، برای این کار سه مرحله کار باید انجام شود:

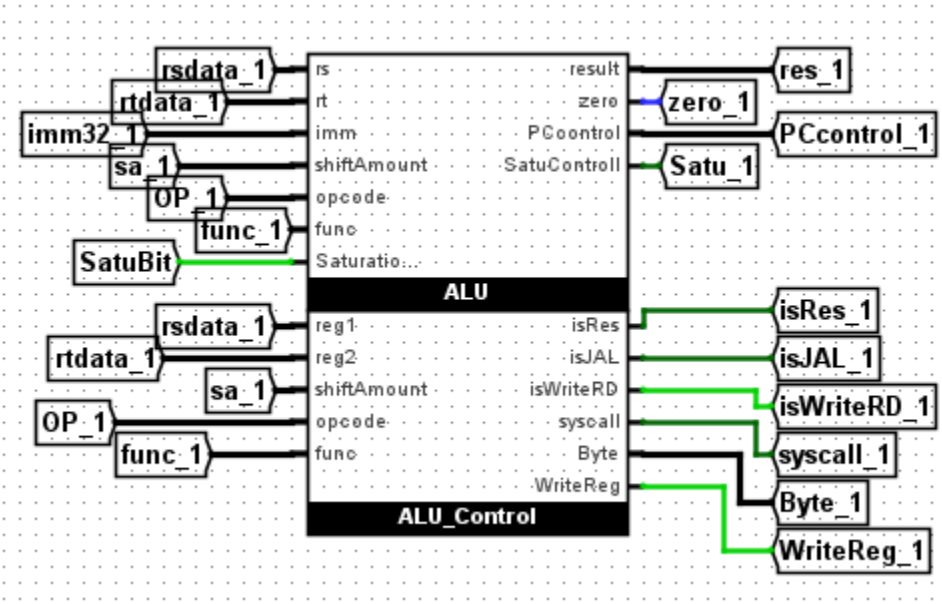
مرحله اول پیاده سازی saturation controller :

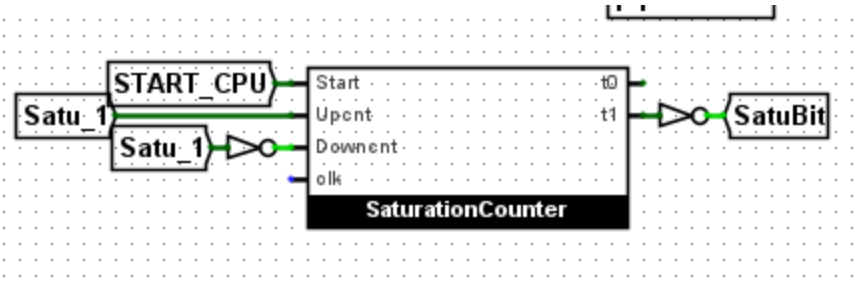
در این مرحله ابتدا counter برای saturationرا مانند شکل زیر پیاده سازی میکنیم:



این شمارنده بر اساس بیت های ورودی تا حد پایین صفر و حد بالای ۳ شمارش میکند.

حال یک شمارنده در مدار قرار میدهیم و بیت upcnt و downcnt آن را بر اساس رخ دادن جامپ یا ندادن جامپ تغییر میدهیم.(اینکه آیا در دستور جامپ رخ داده است یا خیر به راحتی در ALU قابل بررسی میباشد و یک بیت خروجی از ALU با نام Satu برای کنترل Saturation Counter خارج میشود:

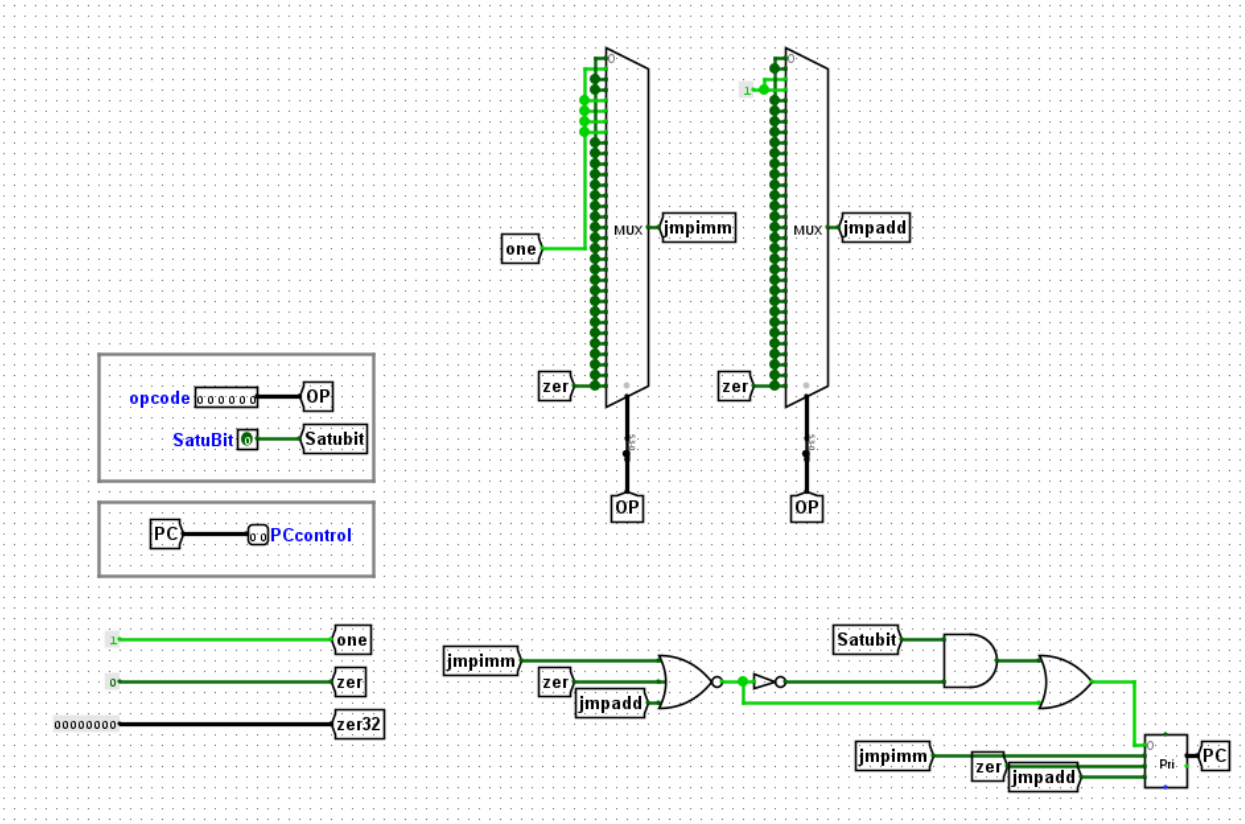




شماتیک این دو بخش در تصاویر بالا آمده است.

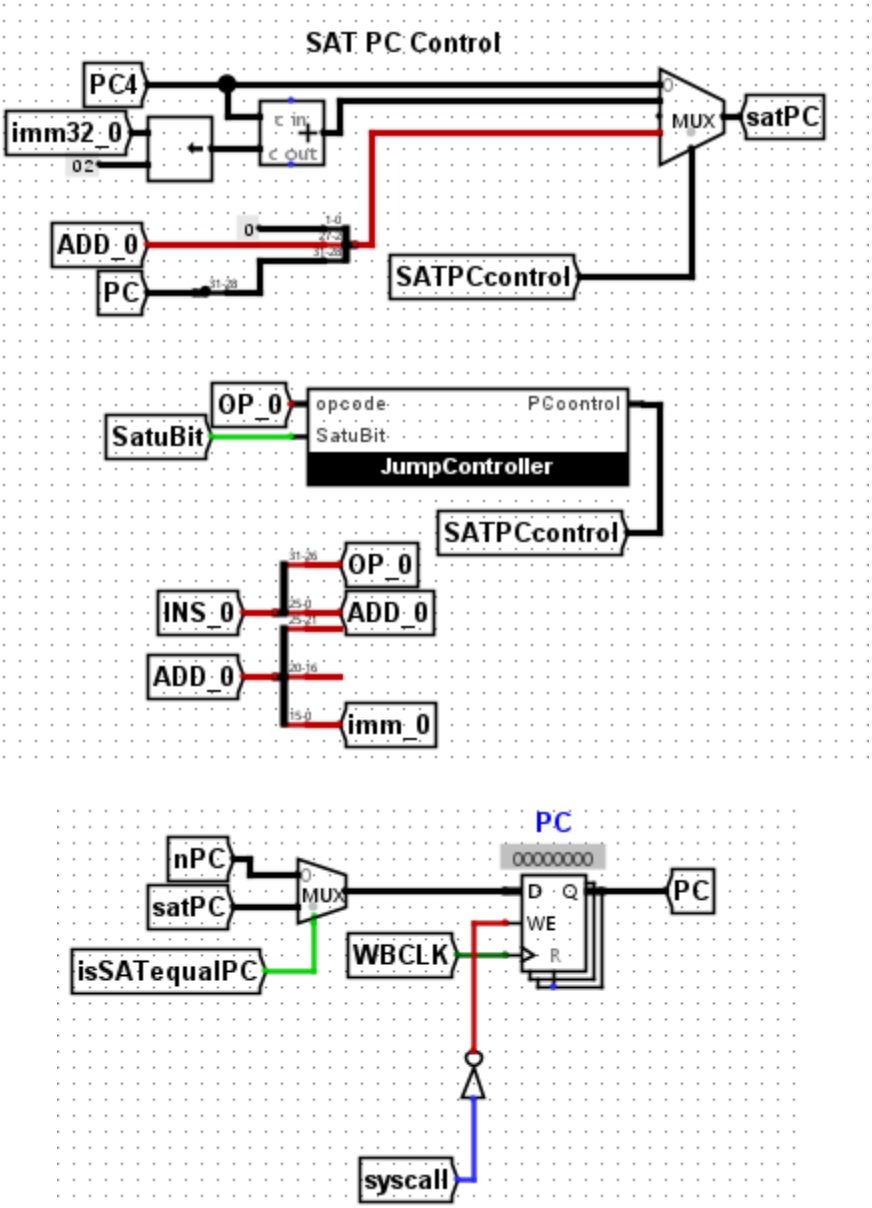
مرحله دوم پیاده سازی jmp with saturation :

در این مرحله مدار jumpController در ALU اضافه میشود که هدف آن بررسی نوع دستور در مرحله اول و خواندن از ممووری است و بر اساس بیت های OP تعیین میکند که آیا دستور از نوع jump immediate یا jump address هست یا خیر:



همانطور که مشاهده میکنید مدار به آپکد های مرتبط به پرش غیر رجیستر حساس است. اگر دستور از نوع جامپ نباشد PC=0 میشود که در MUX انتخاب آدرس بعدی PC+4 آدرس بعدی میشود، در غیر اینصورت با توجه به Satubit که خروجی saturation پیاده سازی شده است تصمیم گرفته میشود که در این دستور پرش رخ دهد یا خیر و PC برابر با ۰، یا ۲ و ۳ بسته به نوع پرش میشود.

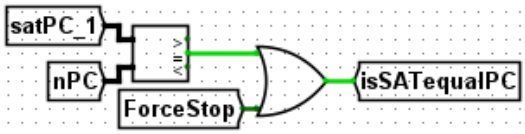
حال در بخش پردازنده آدرس جدید را میان آدرس خروجی Saturation jmp control و nPC که واقعیت پرش است و در دو کلاک بعدی در مرحله کلاک Execution مشخص میشود با بیت isSatEqualPC انتخاب میشود:



بیت isSatEqualPC بیتی است که در دو کلاک بعد پرش را بررسی میکند که آیا پرش درستی انجام داده ایم یا نه، اگر پرش غلط بوده باشد این بیت ۰ میشود و آدرس درست وارد PC میشود و دو کلاک اشتباه زده شده و در غیر اینصورت پروسه بدون مشکل ادامه پیدا میکند.

مرحله سوم کنترل isSatEqualPC:

در این مرحله، ابتدا satuPC را باف میکنیم و در مرحله ALU آدرسی که به آن رفتیم را با آدرس واقعی مقایسه میکنیم، درصورت برابری مشکلی رخ نداده و روند ادامه پیدا میکند و satubit برابر با یک میشود. اما درصورت نابرابری، دو اتفاق باید رخ دهد، دو کلاک دستوری اشتباه وارد شده باید از روند اجرای برنامه حذف شوند و همچنین این دستورات در ادامه جامپ اشتباه ایجاد نکنند، برای اینکار بیت forsestop را به بافر میدهیم به صورتی که استاپ اگر یک بود BigBuffer ورودی ها را باف نکند و ورودی نداشته باشیم، و همچنین در دو مرحله بعدی صرف نظر از برابری جامپ انجام شده و واقعیت isSatEqualPC یک باشد تا جامپ اشتباهی رخ ندهد.



satPC v npc age barabar isSATequlPC = 1

dar do ta clock avval hatman bayad isSatequalPC = 1

agar dar clk i npc != satPC dar clk i isSatequalPPc = 0 va dar clk i+1, i+2 stop = 1 is SATequlPC = 1