CSE 331 Computer Organization

Project 3 – R-type Single cycle MIPS with Structural Verilog

Bu 3. projede önceki projenin üzerine devam ettik. Önceki projemizde ALU tasarlamıştık. 8x1 mux kullanarak select bitlerinden gelen işlemi yapıyorduk. 3. Projemizde bunların üzerine register bloğunu ve R Type instruction ları ekledik.

Mips_register:

Bunları yapabilmek için mips_register module'ünü behavioral kullandık. 32x32 registers kullandık. Bunun içinden okuyup, bunun içine yazdık. Bu registers'ı testbench'den okuduğumuz file ile dolduruyoruz. Yazma işlemini yapmak için 2 tane kontrol yapıyoruz. İlk olarak writeEnable biti 1 oldu mu ona bakıyoruz. Sonra yazılacak yer 0. Register ise oraya yazmıyoruz. Çünkü 0.register her zaman 0 olmalı. Bu şartlar sağlanırsa registers içine gelen değeri yazıyoruz.

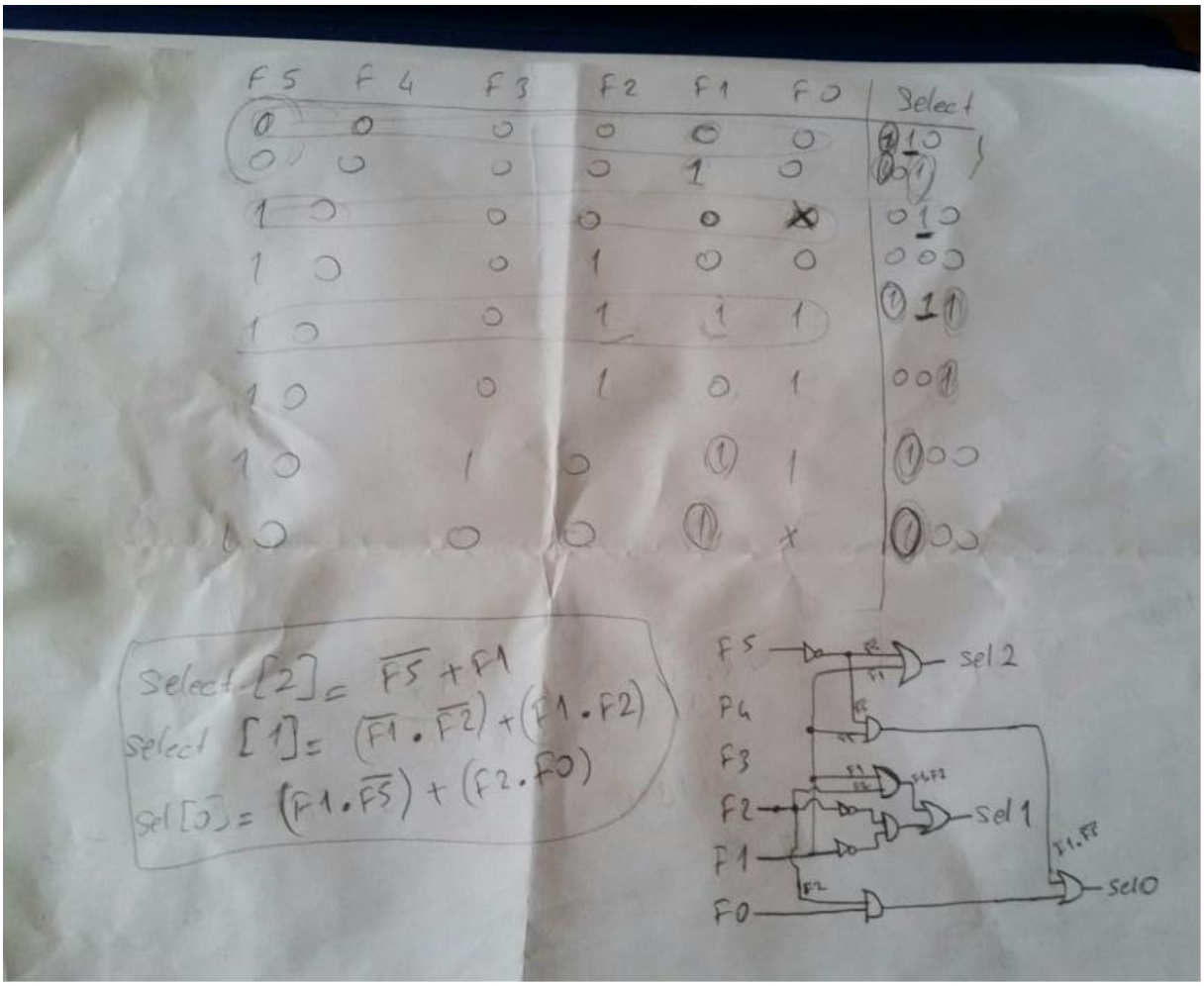
Mips32:

Instructions buraya geliyor ilk olarak. Burada R type'a göre funct, shamt, kısımlarını ayarlıyoruz. Register bloğu burada kullanıyoruz. Oradan gelen sonuçları 2 adet 2x1 mux işlemine tabii tutuyoruz. Bu niye yapıyoruz? Çünkü shift işlemlerinde (sll, srl) önceki projedeki gibi ilk sayıyı ikinci sayı kadar shift etmiyoruz. Bu projede ikinci sayıyı shamt sayısı kadar shift ediyoruz. Bundan dolayı register bloğundan gelen ilk sayı yerine ikinci sayıyı almamız gerekiyor ve ikinci sayı yerine de shamt sayısını almamız gerekiyor. Burada shamt 5 bit ama ikinci gelen sayı 32 bit. Bunu nedenle shamt sayısının 32 bite tamamlıyoruz. Sonra 2 adet 2x1 mux a gönderiyoruz. Burada seçici bit olarak shift lere özgü olarak funct'ın 5. biti bı kullanıyoruz. Bu sonuçlarıda ALU'ya gönderiyoruz. ALU önceki projedekinin aynısı.

Daha sonra ALU dan gelen sonucu alıyoruz. Bu sonucu register bloğuna yazmamız gerekiyor. Ama öncesinde sltu işlemi için biraz değişiklik yapmamız gerekiyor. Sltu için ALU da sub işlemi yapılıyor. Sonra bu sonucu alıp most sign bitini alıyoruz. Bu bitin sol tarafını 0 bit ile dolduruyoruz. Sltu nun sonucunu bulmuş oluyoruz. Eğer sltu sinyali gelirse bu sonucu göndereceğiz. Bunun için buraya da bir adet 2x1 mux koyuyoruz. Bu mux'un sonucunu register bloğuna gönderiyoruz.

ALUControl:

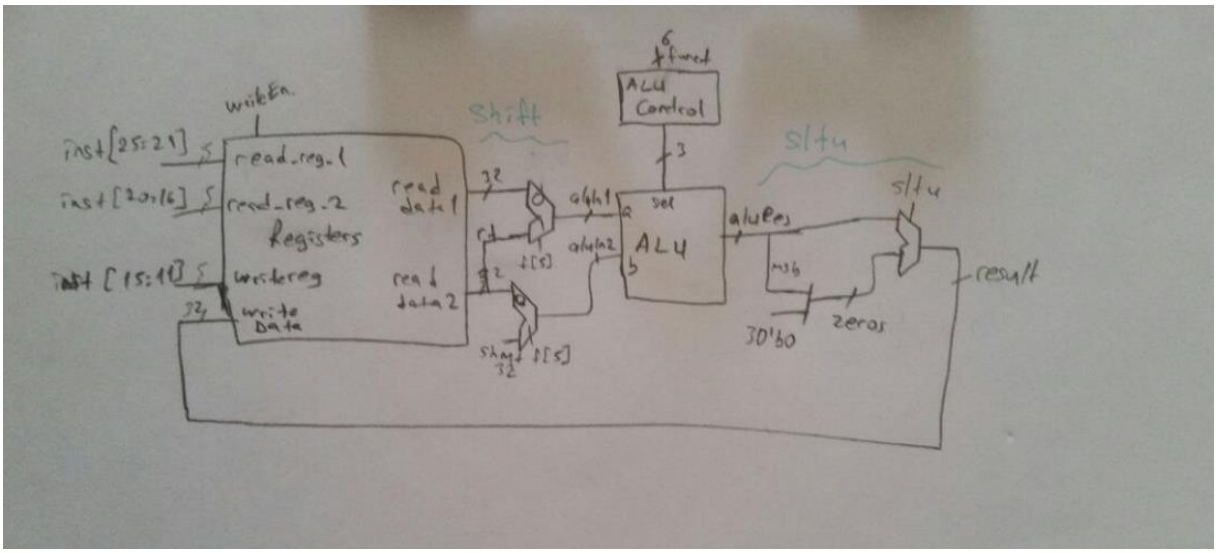
Burada instruction'dan gelen funct'a göre ALU'ya select bitlerini gönderiyoruz. Yani 6bit input funct geliyor ve 3 bit select biti ALU'ya gidiyor. Bunun için tüm funct lara göre bir devre kurmamız gerekiyor. Ben 7.slayttaki gibi tablo şeklinde yaptım sonra yanındaki gibi bir devre kurdum.



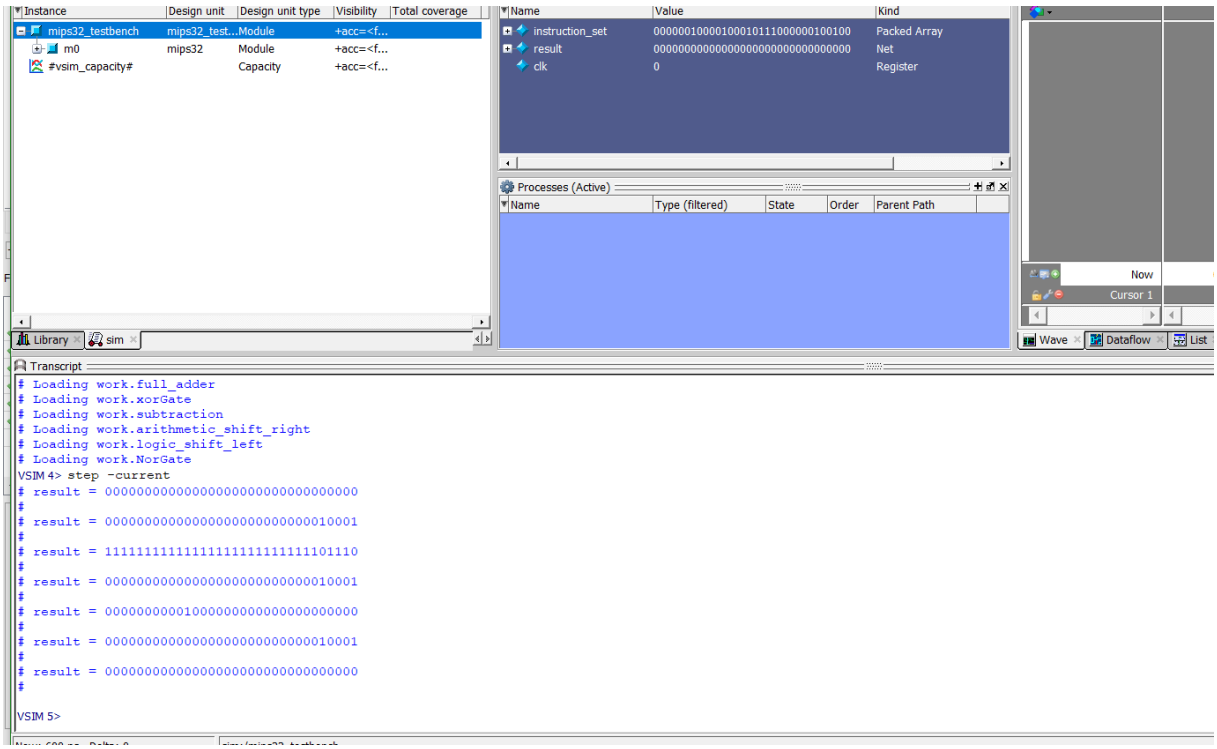
Mips32_testbench:

Burada çoğu şeyi yapan mips32 modülünü test edeceğiz. 32 bit instruction ve result tanımlıyoruz. Dosya okuma ve yazma kısımlarını burada yapıyorum. Dosyayı registers bloğundaki resigters kısmına direk okuyoruz. 10 tane farklı instruction belirleyip farklı işlemler uyguluyoruz. Ve sonuçları registers'dan alıp tekrar dosyaya yazıyoruz. Clk olayını da her instructiondan sonra değiştiriyoruz.

Sonuç olarak aşağıdaki gibi bir devre ortaya çıkıyor.



Modelsim sonucu da aşağıdaki gibi çıkıyor.



Ahmet Yuşa Telli

151044092