RiscV32i Single Cycle Cpu

Temel Amacımız en basit yapılacak şekilde instructionları çalıştırmak.Bu amaçla bir data path bulduk ve bazı düzenlemeler yaptık.İlk önce düzenlemelerden bahsetmeden önce görev dağılımımızdan bahsedelim.3 Üyemiz vardı. Yiğit:Branch unit,muxlar ve alu kısmını yaptı.Emine:Control Unit,Alu,Reg file,imm gen’i yaptı.Metin:Data memory ,İnstruction memory ve test bench yazımını yaptı.Kısaca görev dağılımımız bu şekildeydi.

Hazır bir data path aldığımızdan bahsetmiştik.Bundaki yaptığımız en önemli değişiklik jump sinyalini brancha ayrı kablodan getirmekti.Aslında jump instructionları geldiğinde jump diye ayrı bir tel çekmek yerine branch işlemindeki koşul kontrol etme kablolarından birini koşul olmasada çalıştırıp branchtan bir sonuç elde edebilirdik.Gereksiz sinyal kullandık ama bizim için daha anlaşılır oldu.

İlk önce gelen instruction txt’den çekilip instruction mem.den alınıp 3 yere eş zamanlı gidiyor(paralel çalışma).İmm gen’den imm değeri elde ediliyor.eğer toplama işlemi imm ile yapılıyor ise kullanıyor.Devrede neyin nasıl kullanılıcağını karar veren sinyaller instruction’ın parçalanması ile Control Unitte elde ediliyor.kodun ilk 7 biti op code ,12-14. Bitler arası func 3 ve 25-31. Bitler arası func 7 (0.bitten saymaya başlanılmıştır).Bu aldığı karara göre devre’nin her bir yanına sinyaller gönderiyor çalışmayı ayarlayabilmek için.B tipi ve jal jalr tipleri atlama yapar.Jal ve jalr koşulsuz atlama yaparken B tipleri istenen koşul sağlandığında atlama yaparlar Bunu bizim devremizde branch ve alu işbirliğinde kontrol edilir Branch ve jump sinyalleri ve aludaki durum kontrol sinyalleri ile(less,zero).Aupic ve lui rd registeri olarak instructionda belirtilen registerdaki veriyi imm’indeki değere göre değiştirir ama aupic bu değiştirmeyi yaparken imm’indeki değer son 12 biti olmayan değer ile program counter değerini toplar.

R tipi instructionlar ve i tipi instructionlar temel mantık işlemlerini alu ile yaparlar ama i tipi imm değeri kullanırken r tipi iki registerin içindeki verileri kullanır.L tipi ve S tipi veriler kaldı.ikiside aynı şeyleri ifade ediyor gibi gözükse de l tipi load işlemi ile bellekteki veriyi okur ama store bellekteki veriyi değiştirir.Bu instructionlar data memory ile ilgili instructionlardır.