

Bilgisayar Organizasyonu – Proje 2 Raporu

mips_core modülü

mips_core modülü içinde, gelen 32bitlik instruction op-code, rs, rt, rd, shamt ve funct field olmak üzere 6 parçaya bölünür. Op code'un R Type olup olmadığının kontrolü yapılır. Daha sonra parçalara ayrılmış bitlerden rs, rt ve rd; signal_reg_write ve clk ile birlikte register modülüne gönderilir.

mips_registers modülü

core modülü tarafından gönderilmiş rs, rt ve rd bilgisi register modülünde integer'a çevrilir. Ayrıca registers.mem dosyasının okunması da burada always bloğu içerisinde yapılır. Dosya okunduktan sonra, integer'a çevrilmiş index'ler yardımı ile rs_content ve rt_content bulunur. signal_reg_write kontrolü yapılır eğer 1 ise register bloğuna yazma işlemi gerçekleştirilir, writememb komutu ile dosyaya da yazdırılır böylelikle değişimin gözle görülmesi sağlanır.

Proje içerisinde ayrıca bir ALU modülü oluşturulmamıştır. ALU işlemleri core modülünde rs_content ve rt_content register modülünden alındıktan sonra function code'a göre yapılmaktadır. Result hesaplandıktan sonra write_data'ya yazılmaktadır ayrıca clk ve signal_reg_write sinyalleri de 1 olarak ayarlanır böylece register modülüne yazma sinyali gönderilir. ALU işlemleri yapılırken function code kontrolü de yapılmaktadır bu sayede 9 instruction dışında herhangi bir instruction desteklenmemektedir.

Register modülünün sağlıklı çalıştığından emin olmak adına mips_registers_testbench yazılmıştır. Ayrıca core modülü için de mips_core_testbench yazılmıştır. Bu testbench içerisinde 9 farklı instruction'ı temsil eden instruction kodları mevcuttur. Testbench'lerde debug edebilmek için \$monitor() kullanılmıştır.