PROJECT 2

Öncelikle projede bizden istenen instructionların nasıl çalıştığının analizini yaparak , sadece
RD registerına yazma olduğunu ve her instruction için RD registerına yazma işlemini
yapıldığını analiz ederek mips register modülünü 2 şekilde kullandım.

1.Kullanımda register.mem dosyasından registerların okunması işlemi gerçekleştiriliyor , rs ve rt contentlerini okuyor.Fakat clock sinyalini 0 olarak verdiğim ve bu modülümü de bu kontrole göre (0 ise yazma işlemi yapılmaz ve register mem'de değişiklik olmaz) (1 ise yazma işlemi yapılır ve RD registerına write data yazılır)

2.Kullanımında ise always-begin-end bloğu içerisinde bu fonksiyonun argümanlarını initialize ederek o fonksiyonu tekrar çağırıyorum.Bu sefer clock 1 olmuş oluyor ve yazma işlemi yapılıyor.

Instruction işlemleri için ekstra module yazmadım çünkü yazmak istediğim case yapısında module çağırılması desteklenmiyordu bu sebepten ötürü hesaplamaları mips_core içerisinde case kontrolleri içerisinde yaptım.Case function field'a bakarak gelen function field'ın durumuna göre hangi işlemin yapılacağını seçer ve o işlemi yapar.

Her iki mips_core , mips_registers modülümde de always bloğunu kullandım , çünkü registerlar'a atama işlemini yapmak için bulabildiğim seçenek oydu.Result register'ına değer atanması ve mips_register modülündeki rs ve rt contentlerinin registerlara atanmaları için bu yapıyı kullandım.Onun dışında modüller her seferinde ilk olarak contentlerin okunması , case kontrolleri içerisinde hangi instruction çalıştırılcaksa buna karar verilip çalıştırılması , ardından sonucusun RD registerına yazılması (mips_register modülünü argümanlarını initialize ederek tekrar

çağırılarak oluyor ve clock 1 olmasıyla dosyaya ve register'a yazılabiliyor)

References :
https://stackoverflow.com/questions/10005411/assign-a-synthesizable-initial-value-to-a-reg-in-verilog

(Bu link üzerinden register'a nasıl değer atanabileceğini öğrendim.)

Muzaffer Metehan Alan
151044038