

## CSE 331- Computer Organization

### Proje-2: R-Type MIPS

#### PROJE RAPORU

Bu projedeki amacımız sadece R-Type instructionları destekleyen bir MIPS processor tasarlamaktır. Bunu yapmak için 3 module oluşturdum.

Bu modüllerden top modüle olan mips\_core.v modülüm 32 bitlik bir instruction alıp sonucu output olarak resulta yazıyor. Öncelikli olarak 32 bitlik instructionumu parse etmekle başladım. 6 bitlik opcode, 5 bitlik rs, rt, rd ve shift amount, 6 bitlik function code parse işlemi sonucunda buldum. Ardından bu parçalanmış bitleri mips\_registers isimli modülüne gönderdim.

Diğer bir modülüm olan mips\_register.v isimli modülüm top modülünde çağrılır. Parametre olarak parse edilmiş instruction bitlerini alır. Ayrıca input olarak write sinyali ve clock sinyalini alır. Write sinyali register bloğunda yazma gerçekleştiği zaman 1 olarak gelir. Diğer durumlarda sıfırdır. Clock sinyali bu proje de kullanılmamıştır. Bu modül output olarak da 2 tane 32 bitlik register döndürür. Bu outputlar read\_data\_1 ve read\_data\_2 'dir. Register contentlerin bulunduğu "registers.mem" isimli dosyayı okur. Okuduğu bu dosyayı 32X32'lik registers isimli değişkene atar. Buradan input olarak gelen rs değeri ile read\_data\_1'i rs contenti ile doldurur. Aynı işlem read\_data\_2 içinde gerçekleştirilir. Read\_data\_2' de rt'in contentini içerir. Ve mips\_registers.v read\_data\_1 ve read\_data\_2'yi doldurup işini bitirir.

Diğer bir modülüm olan alu, mips\_core tarafından mips\_registers modülünden sonra çağrılır. Alu.v modülü parametre olarak mips\_register tarafından doldurulan rs, rt ve rd contentlerini alır. Ayrıca shift amount, function code ve opcode da input olarak gelir. Function code göre işlemi gerçekleştirip output olarak resulta yazar. 9 tane instructionu gerçekleştirip resulta yazar. Bu işlemlerin hangisi olduğu function code bakıp karar verdim. R-Type olup olmadığını da opcode bakarak karar verdim. Opcode 0 ise R-Type'dir. Aksi durumda hata veriyor. Ayrıca bu 9 instructiondan biri değilse de hata veriyor. Bu işlemleri gerçekleştirip sonucu doldurup işini bitiriyor.

Result bulunduktan sonra mips\_core signal\_reg\_write'ı 1 yapar. Always bloğunda bu işlem gerçekleştiği için mips\_register tekrar çağrılır. Mips\_register modülünde yaptığım check ile write\_data inputu ile gelen sonuç rd registerına yazılır.

Böylelikle istenilen tüm işlemler yapılmış olup sadece R-Type'ı destekleyen Mips sağlanmış olur. Tüm bu modüller için de testbench yazılıp test edilmiştir.