

141044080

Yunus ÇEVİK

Bilgisayar Organizasyonu

Proje - 2

Module mips_core:

```
module mips_core(instruction, result);
```

Bu modül çağrıldığında instructionları alıp 6 bit opcode, 5 bit rs, 5 bit rt, 5 bit rd, 5bit shamt ve 6 bit funct lara bölüp mips_registers modülüne gönderir. Dönen rData1 ve rData2 outputlarını ALU içine aktarır ve işlemlerin yapılmasını sağlar son olarak result parametresiyle output gönderir. Genel olarak işlemlerin çalıştığı yer mips_core dur.

Module mips_registers:

```
module mips_registers
```

```
( read_data_1, read_data_2, write_data, read_reg_1, read_reg_2, write_reg,  
signal_reg_write, clk );
```

Bu modülün temel amacı signal_reg_write in değerine göre dışarıdan bir dosyayı okuyup registerler a koyma ve registerleri okuyup dosyaya koymasidir.

Module ALU:

```
module ALU (opCode,rd,rs,rt,shamt,funct);
```

ALU modülü içinde r-type lar için çalışan işlemler mevcuttur. Bu işlemler;

add, addu, sub, and, or, sra, srl, sll, sltu instructionları mevcuttur.

Not: Bütün işlemler eksiksiz çalışmasına rağmen register içindeki veriyi array içine alıp işlem yapıyorum ancak tekrardan işlem yapılmış hallerini array e ya da dosyaya yazdırdığımda hata aldığımdan yazma işlemini yorum satırı içine aldım. Verilog içinde de yorum satırında açıklamalarımı yazdım. "registers.mem" dosyası içindekileri değiştirmeden ALU sorunsuz çalışmaktadır.