RISC-V Tabanlı İşlemci Tasarımı

Serhat Erdoğan, Hüseyin Berk Işıldak

Fenerbahçe Üniversitesi

Bilgisayar Mühendisliği

İstanbul, Türkiye

e-mail: {serhat.erdogan, huseyin.isildak }@fbu.edu.tr,

*Özetçe*— Bu proje kapsamında başlangıç tasarım verilen bir RISC-V işlemcisinin ALU ve instruction decoder blokları temel SystemVerilog dili özellikleri kullanılarak tasarım ve doğrulama çalışmaları yapılacaktır.

Anahtar Kelimeler — FPGA, CPU, RTL, System Verilog.

*Abstract*— Within the scope of this project, the ALU and instruction decoder blocks of a RISC-V processor, whose initial design is given, will be designed and verified using the basic SystemVerilog language features.

Keywords — FPGA, CPU, RTL, System Verilog.

# Giriş

Bu proje de bize bir RISC-V işlemcisi verilmiştir ancak bu tasarımın içinde gerçeklemesi yapılmamış modüller bulunmaktadır. Bunlar ALU ve Instruction Decoder modülleridir.

Bizim amacımız Alu ve Instruction Decoder modülleri SystemVerilog dili kullanılarak tasarlanıp, tamamlanan tasarımın doğruluğunu test etmek, SystemVerilog dilini ve RISC-V işlemcisini daha iyi tanımaktır.

# Sistem Mimarisi

**RISC-V:** RISC-V, yerleşik azaltılmış komut seti bilgisayar ilkelerine dayanan bir açık standart komut seti mimarisidir. Diğer ISA tasarımlarının çoğunun aksine, RISC-V ISA, kullanım için ücret gerektirmeyen açık kaynak lisansları altında sağlanır.

**ALU:**

İşlemcinin ALU’sunun destekleyeceği 11 adet işlem vardır. Bu işlemlerden hangisinin yapılacağı alu\_function girişinden gelmektedir. İşlemlere göre a ve b sayıları, result isminde sonuç çıkışı ve sonuç eğer sıfır ise, ayrı bir çıkış olarak sonucun sıfır olması durumunda 1 olan bir çıktı vardır.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 2.

Şekil 2’de alu’nun yapabildiği işlemler ve bu işlemlerin operasyon kodları verilmiştir.

• ADD: A + B

• SUB: A - B

• SLL: A << B

• SLR: A >> B

• SRA: A >>> B

• SEQ: A == B

• SLT: A < B

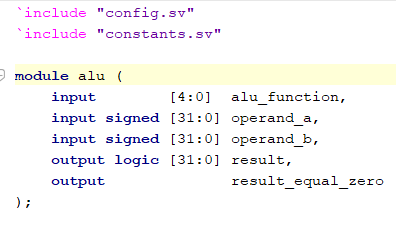
• SLTU: $unsigned(A) < $unsigned(B)

• XOR: A ^ B

• OR: A | B

• AND: A & B

Yukarıda operasyonların açıklamaları verilmiştir.

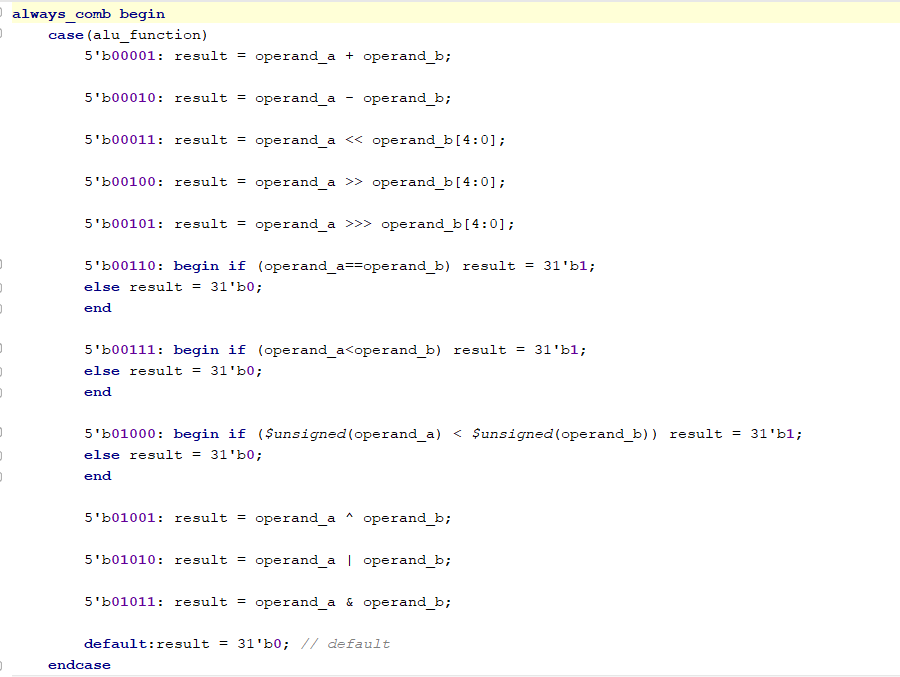


Şekil 3.

Şekil 3’te ALU ünitesinin giriş ve çıkışları gösterilmektedir.

result == 0 olduğunda result\_equal\_zero değeri 1’e eşitleniyor.

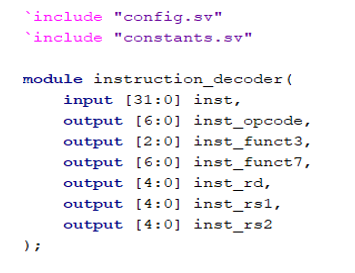
Result == 0 olmadığı durumlarda ise result\_equal\_zero değeri 0 oluyor.



Şekil 4.

Şekil 4’te ki tasarımda bizim eklediğimiz kodlar bulunuyor burada 11 tane operasyon kodlarına karşılık gelen işlemler yapılıyor.

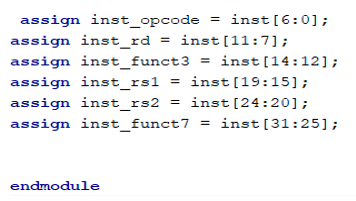
**Instruction Decoder:**

****

Şekil 5.

Şekil 5’te Instruction Decoder ünitesinin giriş ve çıkış sinyalleri verilmiştir.

Burada instruction\_decoder, inst isminde 32 bitlik bir girdi alıyor ve bu girdiyi, inst\_opcode, inst\_funct3, inst\_funct7, inst\_rd, inst\_rs1 ve inst\_rs2 olmak üzere 6 parçaya bölüyor.

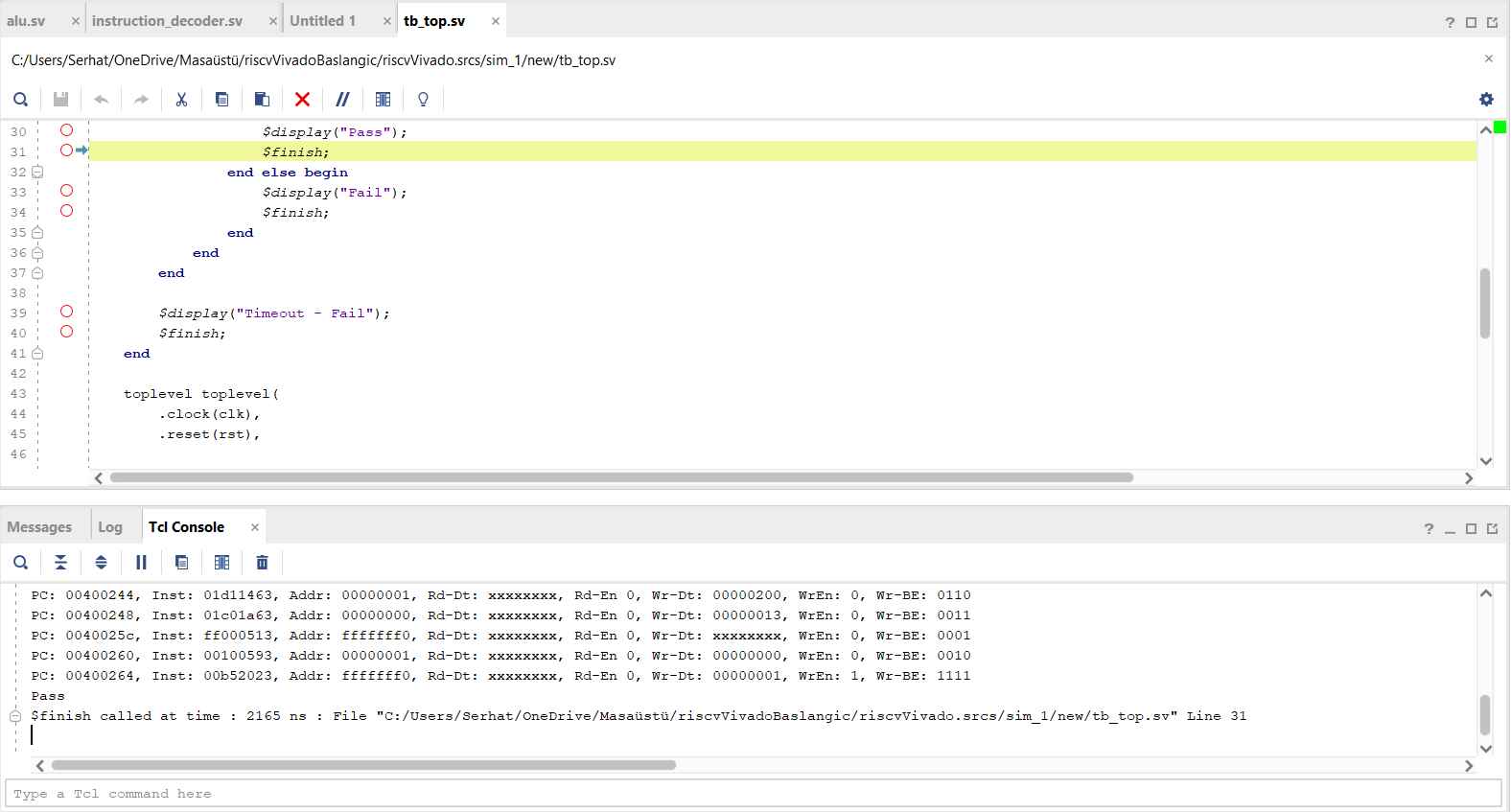


Şekil 6.

Şekil 6’da ise bizim eklediğimiz kod parçacığı bulunuyor. Burada inst girdisinden gelen 32 biti 6 parçaya ayırıyoruz.

# Kullanılan Yazılım

Tasarımları yapabilmek için Xilinx tarafından geliştirilen Vivado Design Suite yazılımını kullandık. Vivado Design Suite, HDL tasarımlarının sentezi ve analizi için üretilmiş bir yazılım paketidir ve Xilinx ISE'nin yerine çip geliştirme ve üst düzey sentez sistemi için ek özellikler sunar. Vivado, tüm tasarım akışının baştan aşağı yeniden yazılmasını ve yeniden düşünülmesini temsil eder. SystemVerilog dili kullanılarak Vivado üzerinde tasarımlar yapıldı. System Verilog IEEE 1800 olarak standartlaştırılmış SystemVerilog ise elektronik sistemleri modellemek, tasarlamak, simüle etmek, test etmek ve uygulamak için kullanılan bir donanım açıklaması ve donanım doğrulama dilidir.



Şekil 2.

Şekil 2.’de ki gibi console kısmında fail çıktısı veriyorsa test başarısız demektir ancak pass çıktısını alıyorsak tasarım başarılı ve testleri geçmiş demektir.

# Sonuçlar

Geliştirilen RISC-V işlemcisi ADD, SUB, SLL, SLR, SRA, SEQ, SLT, SLTU, XOR, OR ve AND işlemlerini destekliyor. Bu proje ile birlikte RISC-V işlemcisinin tasarımını, nasıl çalıştığını, hangi işlemleri yapabildiğini ve mimarisini öğrenmiş olduk. Verilog ve SystemVerilog dilini iyice kavrayıp kendimizi daha da geliştirebildik. Bu işlemcide tamamlanmamış Alu ve Instruction\_decoder tasarımını tamamladık ve bu tasarladığımız işlemciyi Vivado programında test edip doğru çalışıp çalışmadığını kontrol ettik ve tasarımın başarıyla çalıştığını gördük.

##### Proje Ekibi

Serhat Erdoğan

03.06.2000 tarihinde Doğdu 2018 yılında Yalova Uğur Okulları’ndan mezun oldu. Fenerbahçe Üniversitesi’nde Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta.İngilizce biliyor. Orta seviye C, c++ ve python dili ve giriş seviyesi HTML ve CSS biliyor.

Hüseyin Berk Işıldak

25.04.2000 tarihinde doğdu.2018 yılında Yusuf Kemalettin Perin Fen Lisesi'nden mezun oldu. Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. Orta seviye C, c++ ve python dili biliyor.

##### Referans Dosyalar

<https://www.youtube.com/watch?v=bzLLqrXZBE8&ab_channel=BerkI%C8%99%C4%B1ldak>

<https://github.com/serhaterdogan77/BLM202_RISC-V_Proje>

##### Kaynaklar

1. Levent, Vecdi Emre (2021) “Rısc-V Proccessor’’,

*Computer Architecture Lecture Notes*.

1. Levent, Vecdi Emre (2021) “Rısc-V CPU Design’’,

*Computer Architecture Lecture Notes*.