AvionCPU Design

Anıl İzgü
Istanbul Türkiye
e-mail: { anilizgu@marun.edu.tr},

***Abstract* —Bu proje kapsamında, Avion-CPU adlı bir işlemcinin RTL tasarımı Verilog dili kullanılarak gerçekleştirilmiştir. Tasarlanan işlemci üzerinde çeşitli makine dili kod parçacıkları çalıştırılmıştır. Proje sonunda, basit bir işlemcideki RAM, Kontrol Birimi ve Yazmaçların (Registerlar) nasıl birlikte çalışarak makine dili kod parçacıklarını yürüttüğü gözlemlenmiştir.**

Keywords — FPGA, CPU

***Abstract* — Within the scope of this project, the RTL design of a processor named Avion-CPU was implemented using the Verilog language. Various machine code snippets were executed on the designed processor. At the end of the project, it was observed how RAM, the Control Unit, and Registers in a simple processor can work together to execute machine code snippets**

Keywords — FPGA, CPU.

I. INTRODUCTION

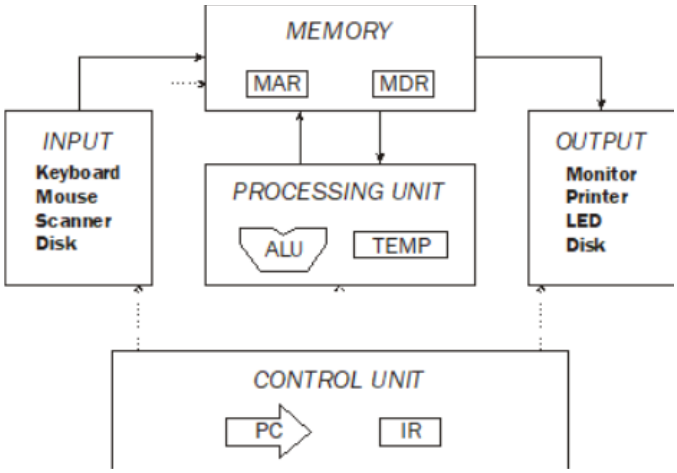
Bu proje, Avion-CPU adlı bir işlemcinin RTL (Register Transfer Level) tasarımının Verilog donanım tanımlama dili kullanılarak gerçekleştirilmesini amaçlamaktadır. Günümüz teknolojisinde işlemciler, elektronik cihazların temelini oluşturmakta ve performansları, enerji verimlilikleri ve karmaşıklıkları ile öne çıkmaktadır. Bu bağlamda, bir işlemcinin temel bileşenlerinin (RAM, Kontrol Birimi ve Yazmaçlar) nasıl entegre çalıştığını ve makine dili kod parçacıklarını nasıl yürüttüğünü anlamak,donanım tasarımı ve bilgisayar mimarisi alanında kritik bir öneme sahiptir.

Proje kapsamında, Avion-CPU'nun tasarımı ve doğrulaması yapılarak, işlemcinin farklı modülleri arasındaki etkileşimler detaylı bir şekilde incelenmiştir. Tasarım sürecinde,Verilog'un sunduğu imkanlar kullanılarak işlemcinin her bir bileşeni ayrı ayrı modellenmiş ve daha sonra bir araya getirilmiştir. Bu çalışma, öğrencilere ve araştırmacılara, basit bir işlemcinin iç işleyişini ve donanım ile yazılım arasındaki ilişkiyi pratik bir yaklaşımla deneyimleme fırsatı sunmaktadır. Projenin sonunda,tasarlanan işlemci üzerinde çeşitli makine dili kod parçacıkları çalıştırılarak, RAM,Kontrol Birimi ve Yazmaçların (Registerlar) birlikte çalışarak makine dili kod parçacıklarını başarıyla yürüttüğü gözlemlenmiştir. Bu rapor, projenin tasarım metodolojisini, sistem mimarisini, kullanılan yazılımları ve elde edilen sonuçları detaylandırmaktadır.

II. SYSTEM ARCHITECTURE

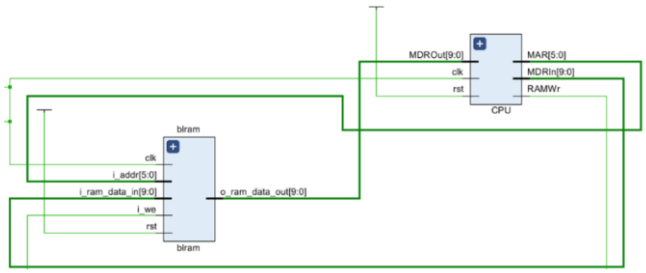
Avion-CPU projesinin sistem mimarisi, temel olarak bir Merkezi İşlem Birimi (CPU) vebir Blok RAM (BRAM) modülünden oluşmaktadır. Bu iki ana bileşen, işlemcinin makinedili kodlarını doğru bir şekilde yürütebilmesi için birbirleriyle entegre bir şekilde çalışmaktadır.

Şekil 1, AvionCPU RTL dizaynında temel alınılan Von Neumann mimarisini göstermektedir.



Şekil 1: Von Neumann Mimarisi

Şekil 2'de, test ortamında modüllerin birbirleriyle olan bağlantılarını göstermektedir:



Şekil 2: Sistem mimarisi

A.CPU MODÜLÜ

CPU modülü, işlemcinin beyni olup, makine dili komutlarını yorumlama ve yürütme sorumluluğunu taşır. Bu modül, Kontrol Birimi, Yazmaçlar (Registers) ve AritmetikMantık Birimi (ALU) gibi temel işlemci bileşenlerini içerir. CPU, harici bellekle (BRAM)veri alışverişi yapmak için belirli sinyalleri kullanır:

clk (Saat Sinyali): Senkronize işlemleri sağlamak için kullanılan ana saatsinyalidir.

rst (Sıfırlama Sinyali): İşlemciyi başlangıç durumuna getirmek için kullanılır.

MDROut[9:0]: CPU'dan belleğe yazılacak 10-bitlik veriyi taşır.

MDRIn[9:0]: Bellekten CPU'ya okunacak 10-bitlik veriyi taşır.

MAR[5:0]: Bellek Adres Yazmacı (Memory Address Register) olup, bellekteki 6-bitlik adres bilgisini taşır.

RAMWr: Belleğe yazma işlemini kontrol eden sinyaldir (Write Enable).

B.BRAM (Blok RAM) Modülü

BRAM modülü, işlemcinin program kodlarını ve verilerini depoladığı bellek birimidir.Bu projede, basit bir bellek yapısı olarak kullanılmıştır ve CPU ile aşağıdaki sinyaller aracılığıyla iletişim kurar:

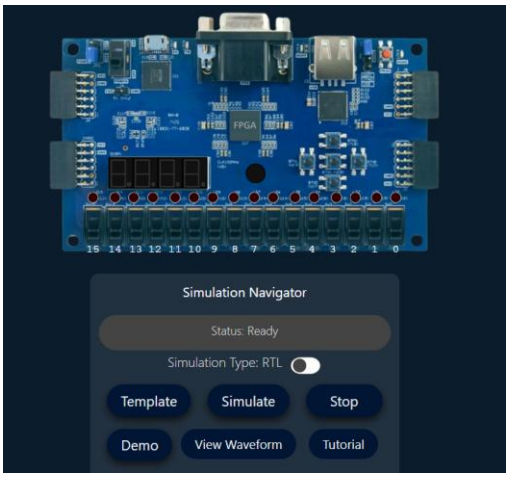
i_addr[5:0]: CPU'dan gelen 6-bitlik adres bilgisini alır.

i_ram_data_in[9:0]: CPU'dan gelen 10-bitlik yazılacak veriyi alır.

i_we (Write Enable): CPU'dan gelen yazma etkinleştirme sinyalidir.

o_ram_data_out[9:0]: Bellekten okunan 10-bitlik veriyi CPU'ya gönderir.

Projede, Avionchip firmasının simülasyon altyapısı kullanılmıştır.



Şekil 3: Simulasyon ortamı

III. SOFTWARE USED

Avion-CPU projesinin RTL tasarımı ve doğrulaması sürecinde simülasyon yazılımı kullanılmıştır. Bu yazılımda Verilog kodunu derleyerek ve sanal ortamda çalıştırarak,tasarımın beklenen davranışları sergileyip sergilemediğini kontrol edilmiştir. Simülasyon aracı, dalga formlarını görüntüleme ve hata ayıklama imkanını sunar.

IV. RESULTS

Tasarlanan AvionCPU ile memori işlemleri (load ve store), aritmetik işlemler (add,sub ve mul), dallanma operasyonları(jmp ve jmz), no operasyon (nop) ve halt (hlt) opearasyonları desteklenmektedir.

Proje kapsamında, Avion-CPU'nun tasarımı ve doğrulaması yapılarak, işlemcinin farklı modülleri arasındaki etkileşimler detaylı bir şekilde incelenmiştir. Verilog dilinin tasarımı, durum diyagramının kullanımı, tasarım aşamasında simülasyon aracının kullanımı ve faydaları, test senaryoları ile tasarımın doğrulama aşamaları bu projede öğrenilmiştir.

PROJECT TEAM

Proje ekibi tek kişiden oluşmaktadır. AvionCPU'nun tasarımında, Verilog dilinin kodlama ve simülasyon sürecinde bulundum.

REFERENCE FİLES

https://github.com/anilizgu1/Avion-CPU-Design
https://youtu.be/Nnj4aArhqrg