

BLM 343 – SOC (System on Chip) Design

CORDIC Tabanlı HW/SW CoDesign

Recep Gemalmaz, Berk Tunç, Ogün Berat Gürses, Damla Su Karadoğan

Fenerbahçe Üniversitesi

Bilgisayar Mühendisliği

İstanbul, Türkiye

e-mail: [damla.karadogan@stu.fbu.edu.tr](mailto:damla.karadogan@stu.fbu.edu.tr), berk.tunc@stu.fbu.edu.tr, ogun.gurses@stu.fbu.edu.tr, recep.gemalmaz@stu.fbu.edu.tr,

1. **Giriş**

**Proje özeti:** Bu proje kapsamında donanım hızlandırıcı olarak Xilinx’in CORDIC (COordinate Rotation DIgital Computer) IP’si kullanıldığı ve işlemcinin hesap yükünü donanım hızlandırıcıya aktardığı bir proje yapılacaktır.

**Anahtar Kelimeler:** FPGA, CPU,SystemVerilog,Vitis,ZYNQ,CORDIC.

**Abstract:** A project to help budget hardware of Xilinx CORDIC (Coordinate Rotation DIgital Computer) IP training and equipment.

**Keywords:** FPGA, CPU,SystemVerilog,Vitis,ZYNQ,CORDIC.

Donanım hızlandırıcı olarak Xilinx’in CORDIC (COordinate Rotation DIgital Computer) IP’si kullanıldığı ve işlemcinin hesap yükünü donanım hızlandırıcıya aktardığı bir proje yapılacaktır.

ZYNQ mimarisine sahip olan PYNQ geliştirme kartı üzerinde proje geliştirilecektir. ZYNQ’in PS (Processor) bölümü, tasarlanacak özel bir modüle verileri besleyip, sonucunu alacak şekilde tasarlanacaktır. Özel modülün giriş ve çıkışları aşağıda verilmektedir.

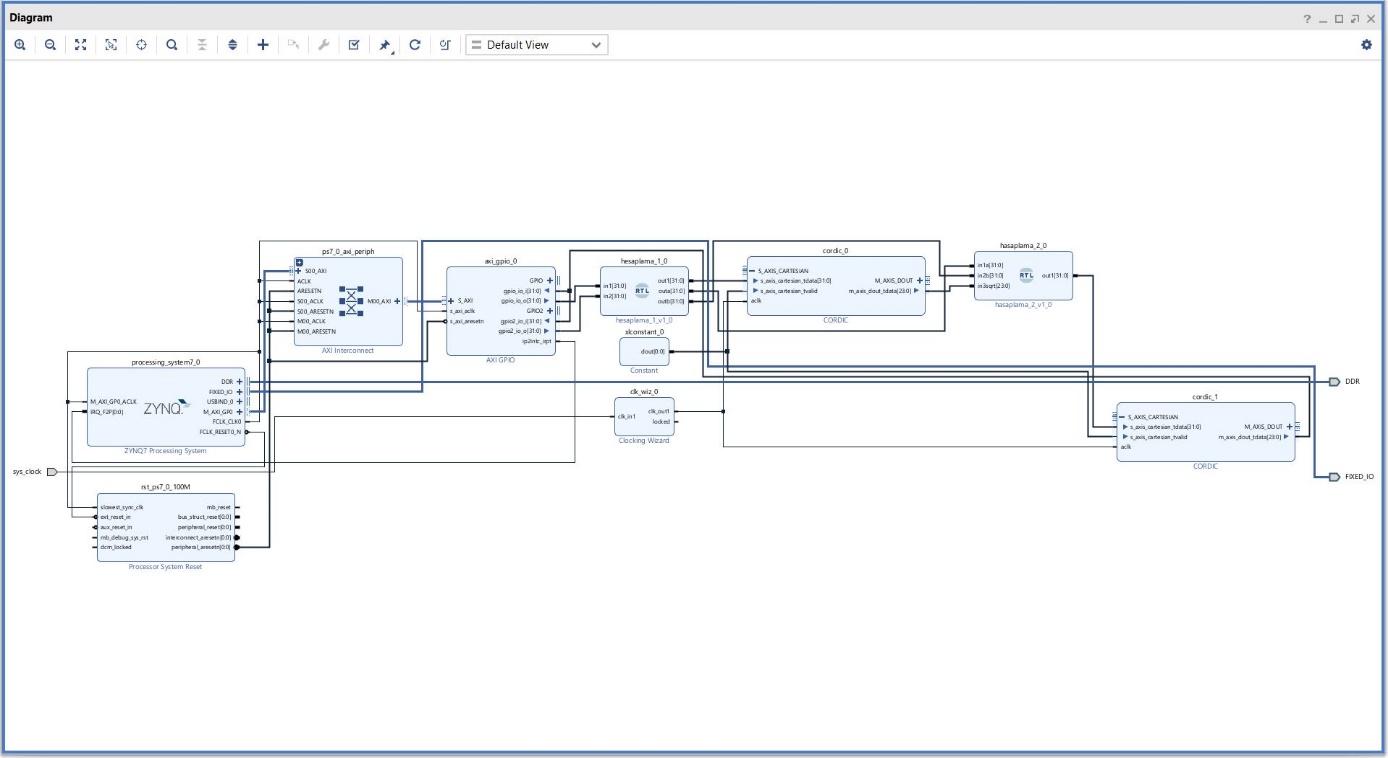
• clk, referans clock sinyali • A[31:0] ve B[31:0] giriş sinyalleri • C[31:0] çıkış sinyalleri

Bu modül içerisinde aşağıda verilen aritmetik işlemi yapan donanımı içermelidir.

C = SQRT(SQRT(a ^ 2 + B ^ 2) + A \* B)

SQRT işlemi için CORDIC IP’si kullanılabilir. CORDIC IP’sinde bulunan SQRT fonksiyonu için unsigned integer seçeneği seçilebilir. AXI GPIO IP’si ile tasarlanan modülün giriş ve çıkışlarına bağlanmalıdır. PS tarafında A ve B sayıları örnek olarak 10 ve 20 olarak ayarlanıp giriş verilip, sonuç doğru üretildiğinde geriye değer alınmalıdır.

1. **Sistem Mimarisi**

****

Öncelikle ZYNQ IPsini ekliyoruz daha sonra gpio IP ekleniyor. Yapmaya çalıştığımız şey gpio IPsinin

çıkışlarından iki adet sayı göndermek, gönderilen sayıları hesaplama modülü içerisinde istenen hesaplama adımlarından 1. kısmını yapmak (hesaplama1). Bu sayıları Vitis içerisinde tanımlıyoruz. metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

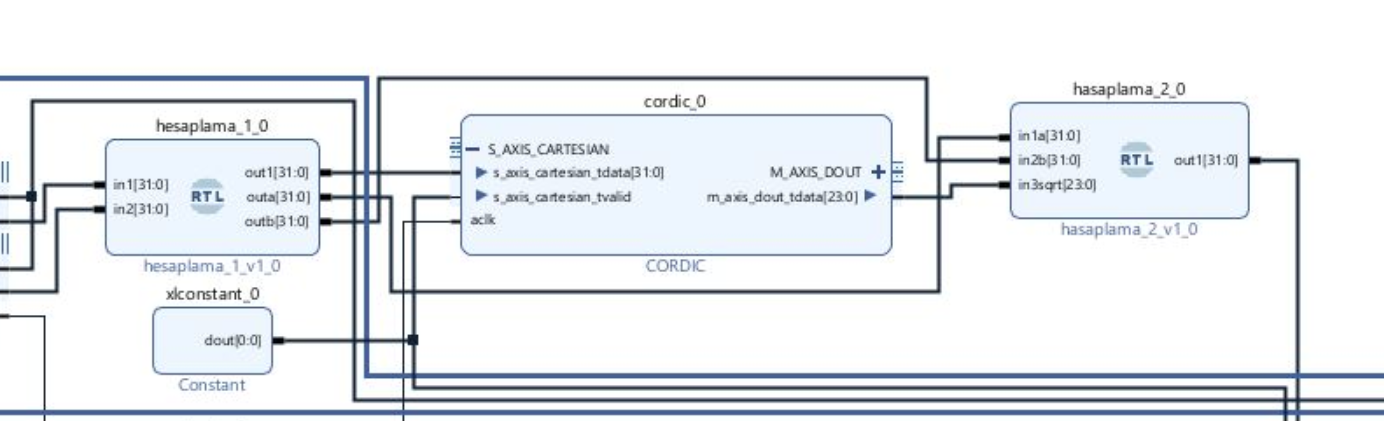
Bu kısımda görüldüğü üzere 10 ve 20 kısımları tanımlanmakta.

Oluşturulan sayıları gpio IPsi ile hesaplama modülüne yolluyoruz.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu kısımda input ve outputlar görülmekte. hesaplama modülü görüldüğü üzere iki sayının da karelerini alıp topluyor ve out1 çıkışına besliyor. Başlangıçtaki input değerlerini de başka modülde kullanmak için 2 farklı çıkışa besliyor.

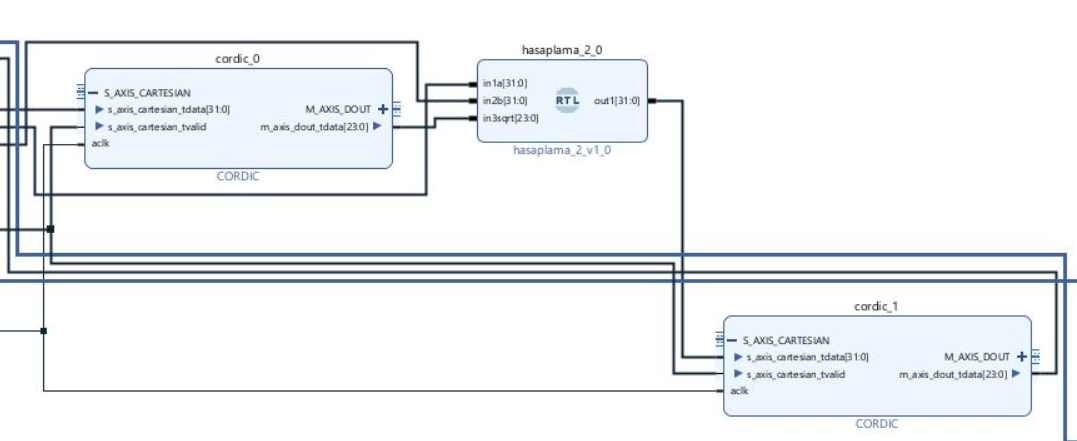


diyagramda da görüldüğü üzere yapılanişlem out1 çıkışından kök alma işlemini yapan cordic IPsine gidiyor. outa ile outb çıktılar ise tekrar hesaplama yapmak için hesaplama2 modülüne gidiyor.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

hesaplama 2 modülü input olarak cordicten çıkan kök sonucunu ve önceki modülden gelen outa ve outb yi alıyor. bu kısımda yapılan işlem önceki modülden gelen iki sayıyı çarpıp onları kökü alınmış değer ile toplamaktır.



hesaplama2 modülünde işlem tamamlandıktan sonra elde edilen çıktı yeni oluşturulan yine kök alma için kullanılacak olan cordic IPsinin içine gidiyor ve son elde edilen sayının kökü alınıyor.

2. oluşturduğumuz cordic IPsinin çıktısı ise Gpio IPsinin girişlerine besleniyor. Gpio IPsinin girişleri değiştiği andan itibaren interrupt üretiliyor ve bu interrupt ZYNQ Ipsine geliyor ve dallanma oluşuyor.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

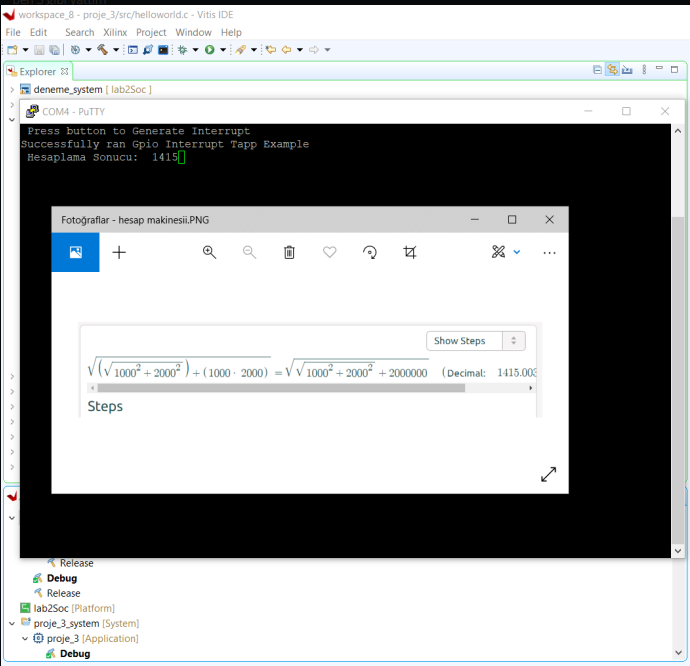
dallanma ile GpioHandler'ın içerisine giriyor ve işlem sonucunda oluşan sayı burada bastırılmış oluyor. Ve Proje tamamlanıyor.

**Sonuçlar**

Projenin sonuç çıktıları şu şekildedir.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu



1. **Kullanılan Yazılım**

Tasarımlarımızı yapmak 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. Biz de SystemVerilog dilini kullanarak Vivado üzerinde tasarımımızı yaptık. 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.

1. **Proje Ekibi**

**Damla Su KARADOĞAN**, 11.02.2001 yılında doğdu. 2019 yılında Özel Envar Anadolu Lisesinden mezun oldu. Şu anda Fenerbahçe Üniversitesinde Endüstri Mühendisliği bölümünde lisans eğitimini almakta ve Bilgisayar Mühendisliğinde ÇAP eğitimi alıyor. Öğrenci numarası, 190302016.

**Ogün Berat GÜRSES**: 10.11.2000 Sakarya ili Adapazarı ilçesinde doğdu. 2018 yılında Sakarya Anadolu Lisesi’nden mezun oldu. Şu anda Fenerbahçe Üniversitesi’nde Bilgisayar Mühendisliği bölümünde eğitim almaktadır. C, C++, mySQL, Verilog, Sys-Verilog ve Python ile ilgilenmektedir.

**Berk TUNÇ:** 04.02.2000 yılında Yalova ili Merkez ilçesinde doğdu. 2018 yılında Şehit Osman Altınkuyu Anadolu Lisesi'nden mezun oldu. Şu anda Fenerbahçe Üniversitesi’nde Bilgisayar Mühendisliği bölümünde bölümünde eğitim almaktadır. mySQL, HTML, C, C++, Verilog, Sys-Verilog ve Python ile ilgilenmektedir.

**RECEP GEMALMAZ:** 16.10.2000 tarihinde Kadıköy’de dünyaya geldi. 2018 yılında Alparslan Anadolu Lisesi’nden Mezun oldu. Şu anda Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. Java, C, C++ ve Pyhton dillerinde bilgili. Android Programlama, Verilog ve Sys-Verilog ile ilgilenmektedir.

1. **Referans Dosyalar**

**Github Dosyası:**

<https://github.com/brktnc/SOC-System-on-Chip-Design-CORDIC-Tabanl-HW-SW-Codesign>

**Youtube Adresi:**

<https://www.youtube.com/watch?v=jaxvgW-2OZs>

1. **Kaynaklar**

[1] http://www.levent.tc/courses/system-on-chip-soc-design