

**Petalinux Tabanlı CORDIC HW/SW CoDesign Projesi**

Evrim Arda Kalafat, Arda Alhan, Cüneyt Balcı

Fenerbahçe Üniversitesi

Bilgisayar Mühendisliği

İstanbul, Türkiye

e-mail: [evrim.kalafat@stu.fbu.edu.tr](mailto:evrim.kalafat@stu.fbu.edu.tr), [arda.alhan@stu.fbu.edu.tr](mailto:arda.alhan@stu.fbu.edu.tr) , [cuneyt.balci@stu.fbu.edu.tr](mailto:cuneyt.balci@stu.fbu.edu.tr)

**Proje özeti:** Proje Petalinux ortamından UART arayüzü üzerinden yakalanan iki sayı girişi verilip, beklenen çıktının aynı sonucu elde edilip edilmediği kontrol edilecektir. Sonuç UART arayüzü üzerinden PC’e gönderilecektir.

**Anahtar Kelimeler:** FPGA, CPU, SystemVerilog, Petalinux, UART.

**Abstract:** Within the scope of this project, we are going to design ALU and Instruction decoder blocks of a RISC-V processor by using the basic SystemVerilog language features.

**Keywords:** FPGA, CPU, RISC-V, SystemVerilog, RTL.

1. **Giriş**

Donanım hızlandırıcı olarak Xilinx’in CORDIC (COordinate Rotation DIgital Computer) IP’si kullanıldığı ve işlemcinin hesap yükünü donanım hızlandırıcıya aktardığı bir proje yapılacaktır. Bu donanım tasarımı PL tarafında hazırlandıktan sonra Petalinux ortamından veriler aktarılacaktır.

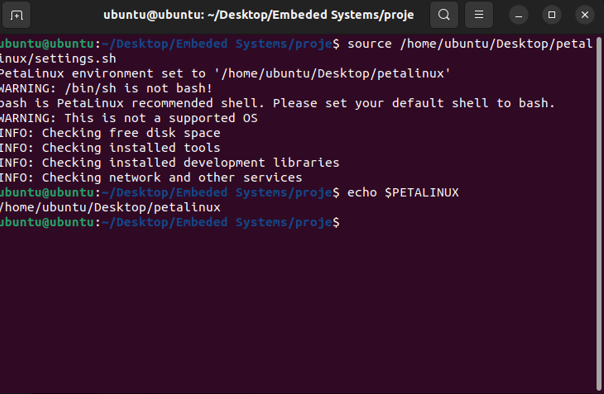
1. **Sistem Mimarisi**

**Xilinx Vivado Design Suite:** Xilinx Vivado Design Suite, FPGA geliştirme kartları üzerinde çalışmalar yapmak için gerekli olan tasarımı oluşturmak için kullanılmaktadır. Verilog, VHDL vb.. donanım tasarım dillerini alarak, FPGA’e konfigüre edilebilecek (Xilinx firması FPGA’leri için .bit uzantılı dosyalar) tasarım dosyasını oluşturur.

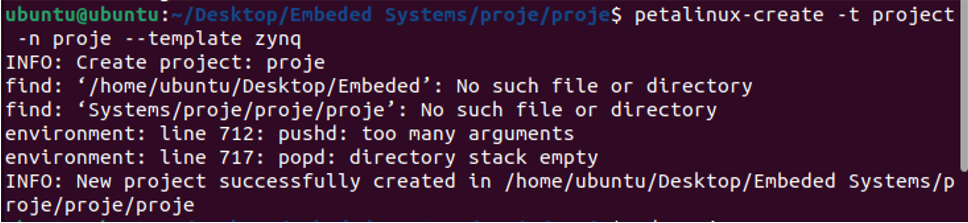
**Visual Studio Community:** Proje kapsamında, ödev ve LAB’larda kullanılan Microsoft’un derleyicisi olan Visual Studio Community kullanılacaktır.

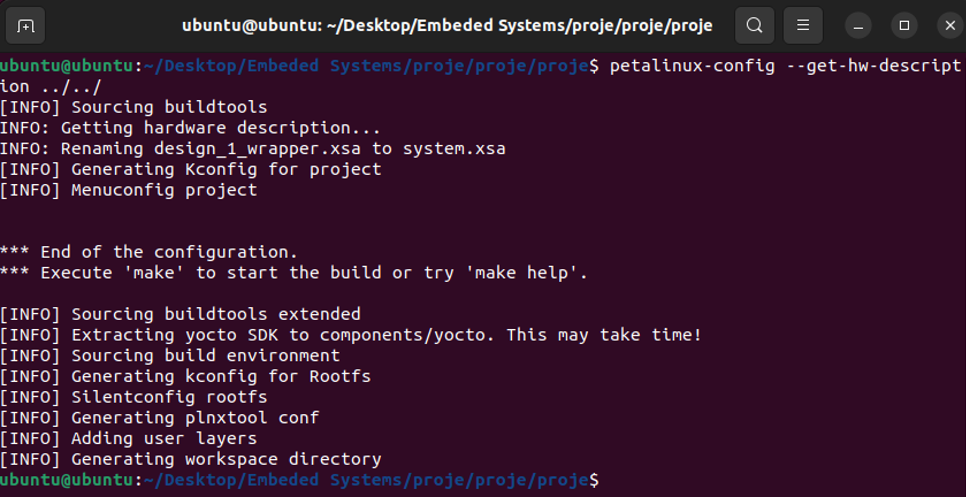
1. **Kullanılan Yazılım**

* Vivado’da dizaynımızı oluşturuyoruz ve ardından UART arayüzü ayarlamalarını yaptık. Burada kurduğumuz petalinux’ü çalıştırdık ve aktif ettik. Ardından da echo komutu ile doğru çalışıp çalışmadığını test ettik.

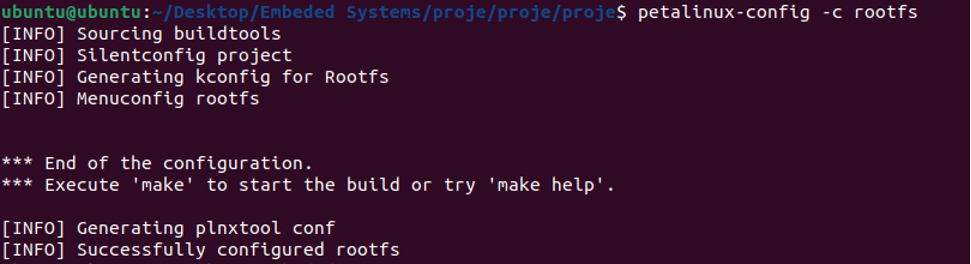


* Sonrasında proje taslağını oluşturup, içine girdik ve sonrasında da get-hw-description argümanıyla projeyi konfigüre ettik.

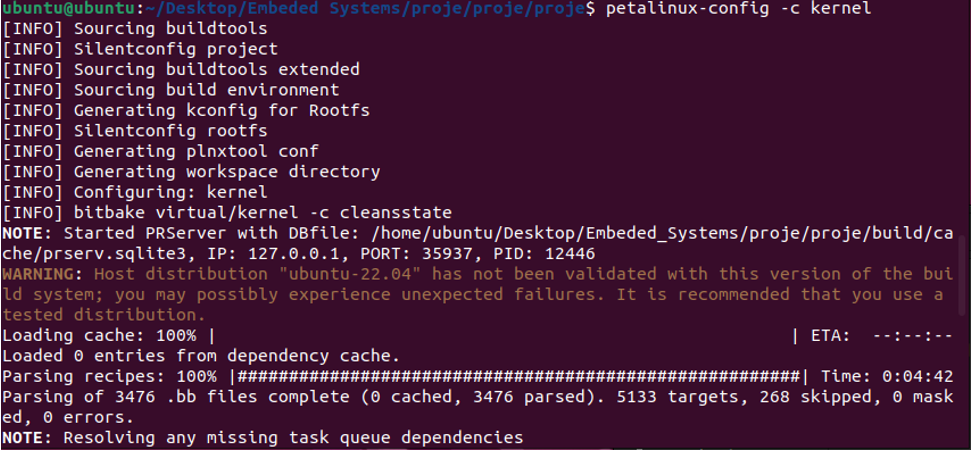




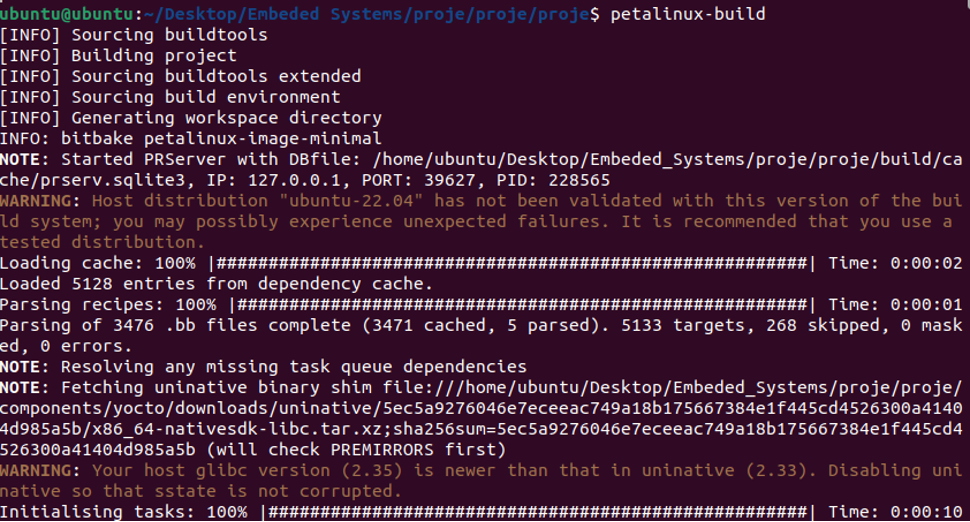
* Sonrasında ethernetten veri yakalayacağımız ve bunu C kodlarıyla yapacağımız için de içeride compiler’lara ihtiyacımız olacak. Bu yüzden de ‘petalinux config –c rootfs’ komutu ile içeriye compiler paketi koyulmasını sağladık.



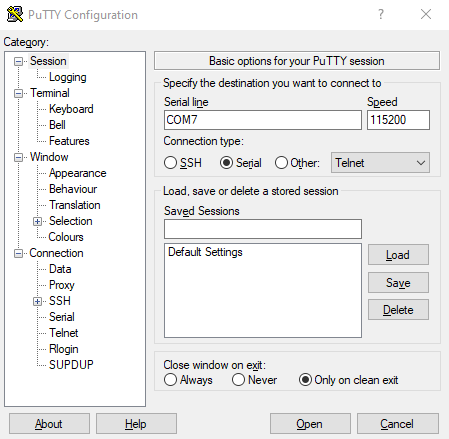
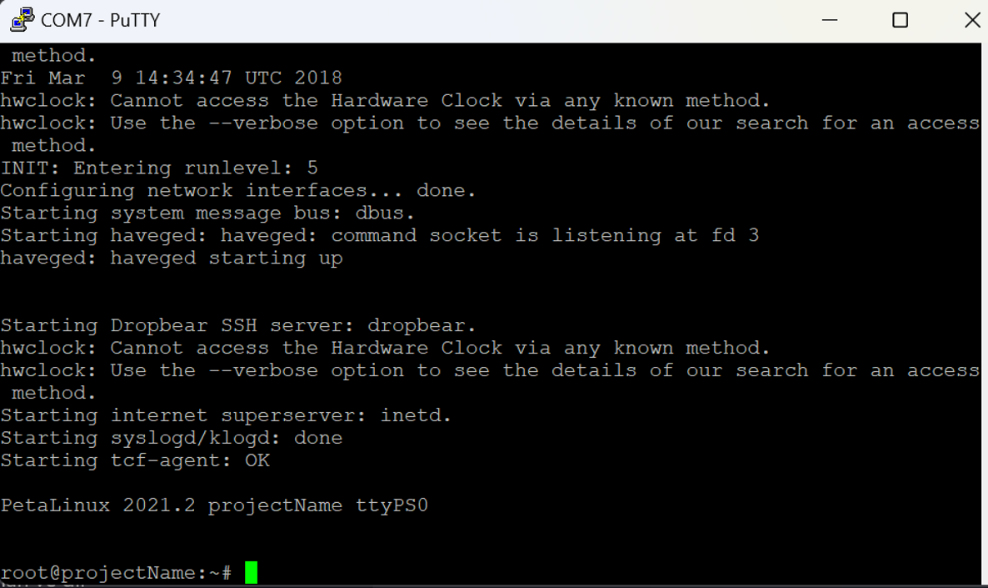
* ‘petalinux-config –c kernel’ komutu ile kernel için gerekli olan ayarların yapıldığı kısımdan kernel’ı modifiye ettik.



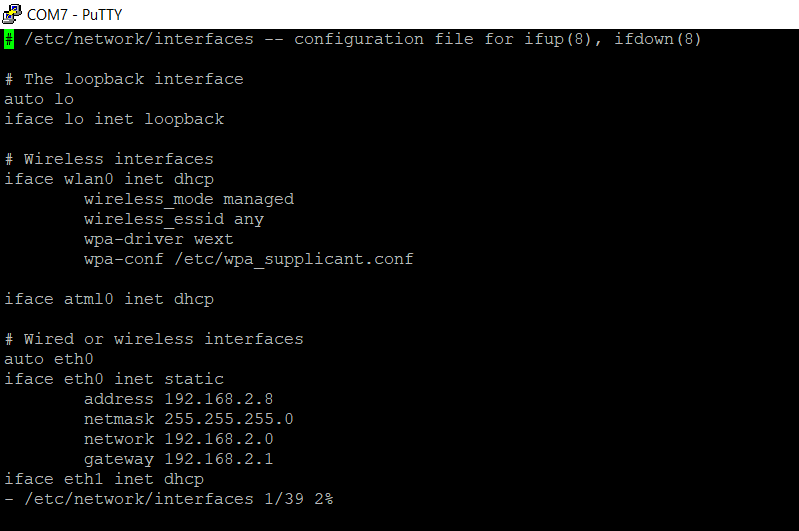
* Sonrasında ise ‘petalinux-build’ komutu ile paketi compile ettik.

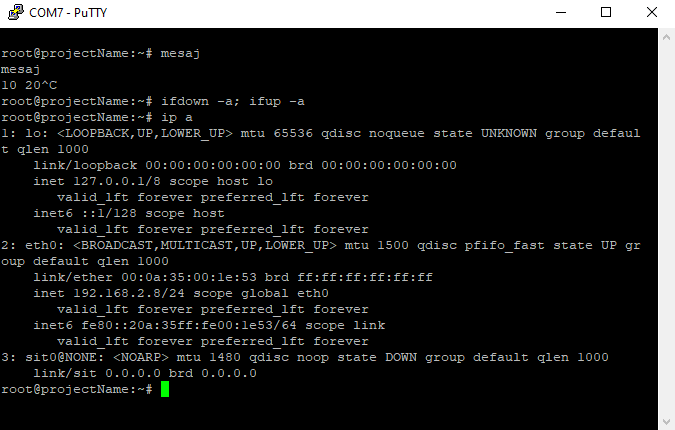


* Burada bilgisayara bağladığımız Petalinux kurulu sd kartı PYNQ FPGA geliştirme kartına taktıktan sonra gücü verdik. Ardından PuTTY’den seri porttan çıktıları almak için line ve speed ayarlamalarını yapıp seri port bağlantısını açtık.

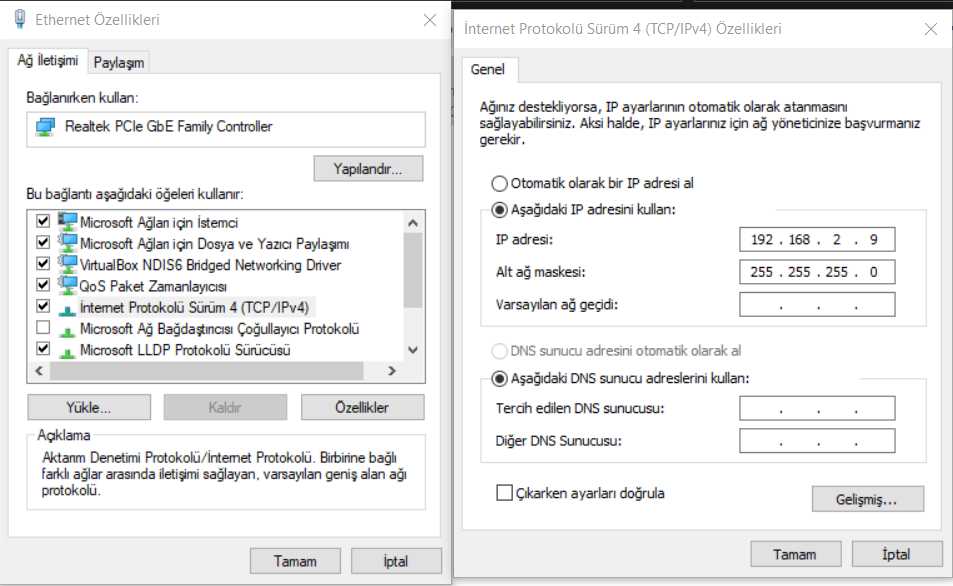


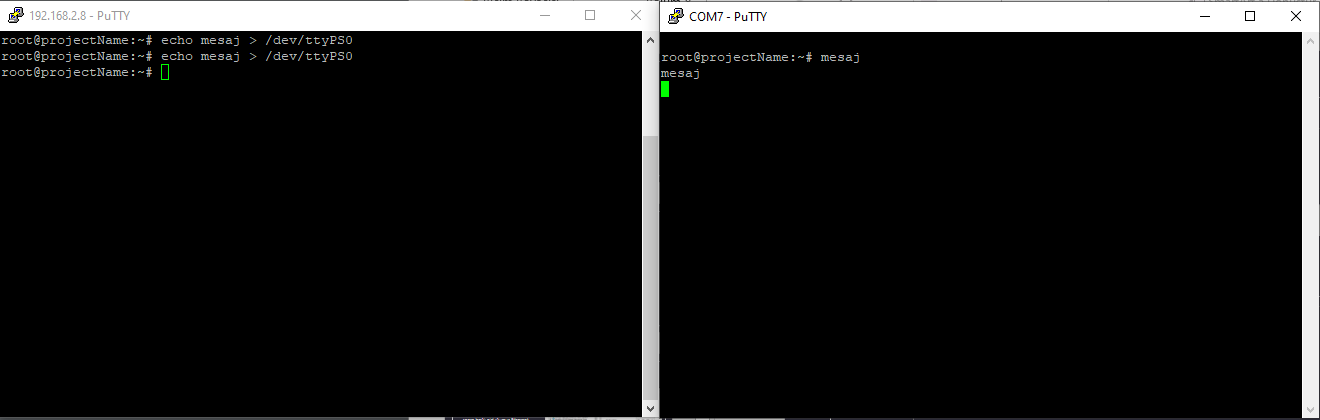
* vi /etc/network/interfaces komutu ile FPGA’in IP atamasını (192.168.2.8) yaptık. Ardından network arayüzünü yeniden başlatmak için ‘ifdown –a; ifup –a’ komutunu kullandık ve sonrasında da ‘ip a’ komutu ile FPGA’in IP adresinin gelmiş olduğunu gördük.

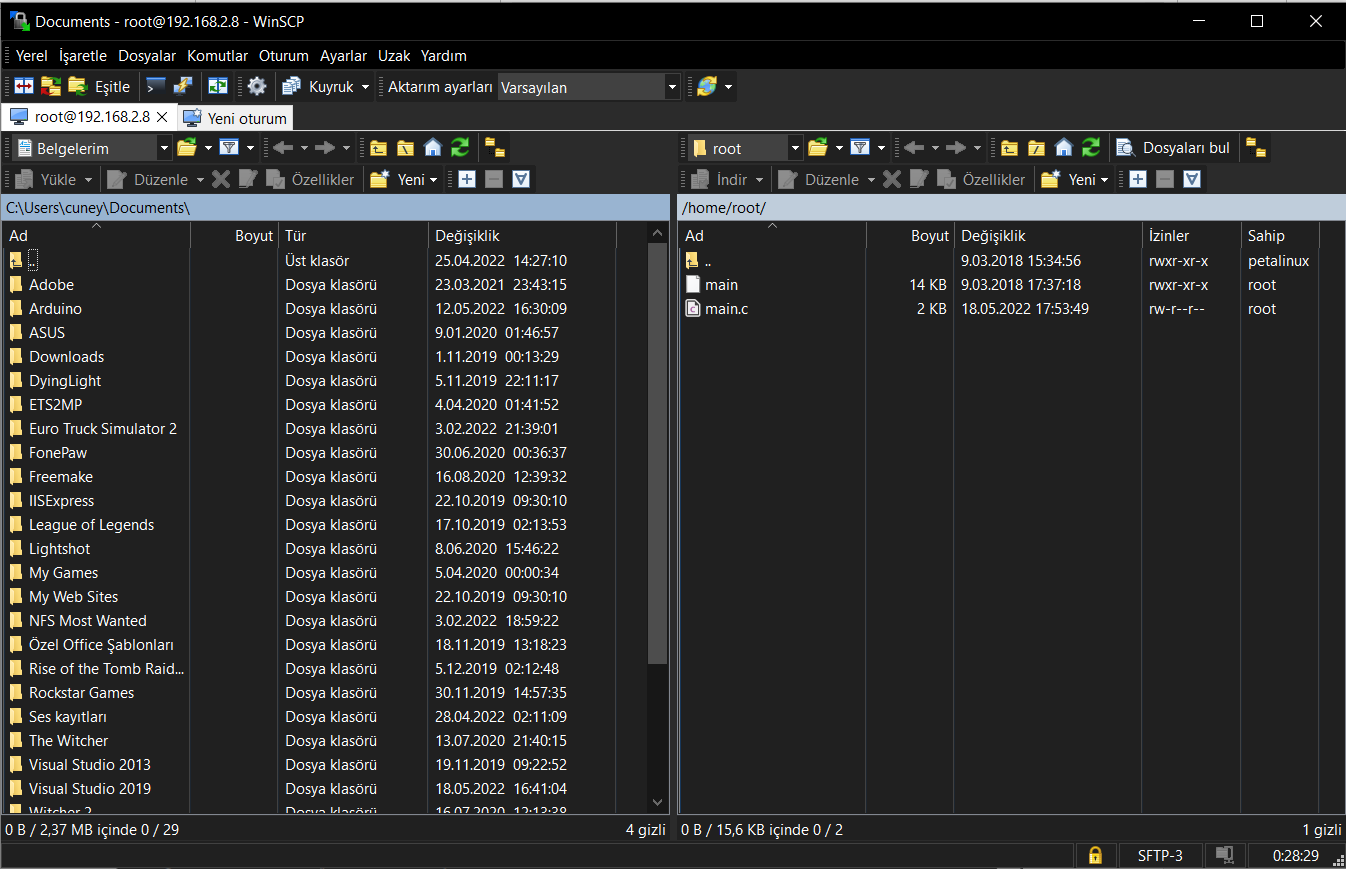




* Ethernet kablosu ile bilgisayar ve PYNQ Kart ile bağlantı kurduktan sonra bağlı olduğumuz Ethernet özelliklerinden, TCP/IPv4’e tıklayarak statik IP(192.168.2.9) verdik.

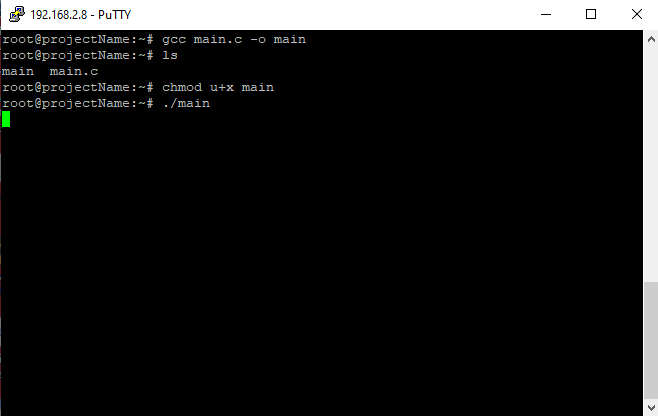


*  ls /dev/tty\* komutundan sonra FPGA’imizin seri port arayüzüne sahip olan yer olan ttyPS0’dan veriyi dışarı transfer ettik. “echo” komutu ile ssh arayüzünden basılan mesaj’ların COM4(bilgisayarın seri portu)’ten geldiğini görmüş olduk.
* Daha sonra WinSCP arayüzünden ethernet altyapısını ayağa kaldıracak kod parçasını ekledik.

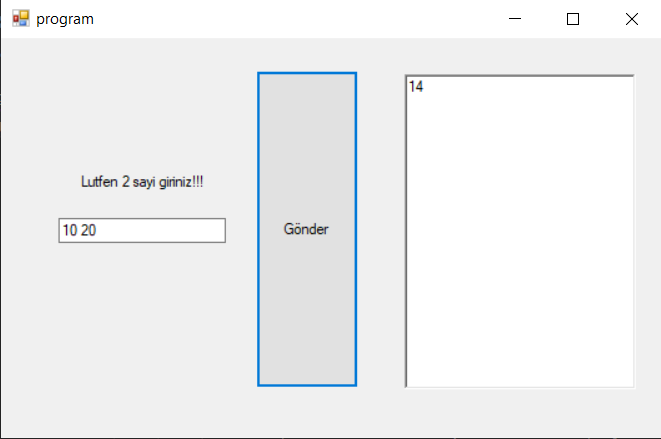


(İçerideki C kodu github’da paylaşılmıştır.)

* Sonrasında ssh’tan ‘gcc main.c –o main’ komutu ile kodumuzu derledik. Sonrasında ‘chmod u+x’ komutu ile çalışma yetkisi verdik. Ardından ‘./main’ komutu ile çalıştırdık.



* Daha sonra Visual Studio’da C# ile gerekli olan arayüzü oluşturuyoruz ve petalinux olan FPGA’e göndereceğimiz sayıları girdik.



1. **Sonuçlar**

Petalinux ortamından UART arayüzü üzerinden yakalanan iki sayı girişi verildi ve ardından beklenen çıktının aynı sonucu elde edildi. Test kapsamında sonuç UART arayüzü üzerinden PC’ye gönderildi.

1. **Proje Ekibi**

**Evrim Arda KALAFAT**, 25.09.2001 yılında istanbulda doğdu. 2019 yılında Kadıköy Final Temel Lisesi’nden mezun oldu. Şu anda Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. Bilgisayar Mühendisliği Bölüm Temsilciği yapmaktadır. C, C++ ve Pyhton dillerinde bilgili. Programlama, yapay zeka ve siber güvenlik ile ilgileniyor.

**Cüneyt BALCI**, 28.08.2000 yılında İstanbul'da doğdum. 2018 yılında Final Temel Lisesi'nden mezun oldum. Şu anda Fenerbahçe Üniversitesi - Bilgisayar Mühendisliği bölümünde ve çift ana dal programı kapsamında Ekonomi (İngilicze) bölümünde lisans eğitimi almaktayım. C, C++ ve Python dillerinde bilgili. Network ve blockchain alanlarıyla ilgileniyor.

**Arda ALHAN,** 18.05.2001 yılında İstanbul’da doğdu. 2019 yılında Eyüp Anadolu Lisesi’nden mezun oldu. Şu anda Fenerbahçe Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimi almakta. C, C++, C#, Selenium ve Pyhton dillerinde bilgili. Programlama ile ilgileniyor.

1. **Referans Dosyalar**
2. <https://youtu.be/VKGzSj22D90>
3. <https://github.com/rhgod/RISC-V_Project>
4. **Kaynaklar**

[1] http://www.levent.tc/courses/embedded-systems

[2] http://www.onurbabur.com/linux-sistemlerde-statik-ip-atama-ip-degistirme-islemleri/