



Kurulum

PULPissimo açık kaynaklı bir işlemci olduğundan ötürü, süreç boyunca kullanılan tüm programların versiyonları belirtileceği şekilde kurulmalıdır. Farklı versiyon kombinasyonlarında farklı hatalar bulunmakta ve çözmesi vakit kaybına sebep olmaktadır. Bu sebepten ötürü, kurulması gereken program ve versiyonları şu şekildedir.

Ubuntu 18.06 LTS
Xilinx Vivado 2018.3
Quartus
QuestaSIM
RiscV-GNU-TOOLCHAIN
PULPissimo
Pulp-Runtime
Pulp-Runtime-Examples
OpenOCD
CuteCOM

Eğer FPGA üzerinde C kodu çalıştırılmak isteniyorsa, harici bir JTAG programlayıcı gereklidir.

Kurulum sırasında önerim, aşağıda belirtilecek kısımlardan Ubuntu kısmını bitirdikten sonra, Quartus'u ve Vivado'yu sırayla indirin. Birini kurmak için diğerinin inmesini beklerseniz kurulumlar çok daha uzun bir süre alacaktır. Önerim Quartus'u ve Vivado'yu indirirken bir yandan RISC-V-GNU-TOOLCHAIN adımlarını yapmanız olacaktır. Bu şekilde en optimal sürede kurulumu tamamlayabilirsiniz.

Ubuntu

Ubuntu işletim sistemi, Windows işletim sistemi yüklü bir bilgisayara dual-boot yöntemi ile yüklenebilir ve kullanılabilir. Bu işlemin nasıl yapılacağını öğrenmek için Ankara Üniversitesi Bilgisayar Topluluğundan Mehmet AFACAN'ın youtube videosuna bakabilirsiniz.

<https://youtu.be/BEEbB7D9tCE>

Ubuntu 18.06 LTS kurulduktan sonra, süreç için gerekli bir çok eklentinin kurulması gereklidir. Bu kurulumlardan bazılarında yönetici izni istenmektedir. Bu sebeple kurulumlar yapılırken garantiye alınmalı ve komut satırının başına sudo eklenmelidir.

Komut penceresine yapıştırılacak şekilde sırasıyla satırlar verilmiştir.

- 1- sudo apt install vim git python3-pip gawk texinfo libgmp-dev libmpfr-dev libmpc-dev
- 2- sudo pip3 install pyelftools
- 3- sudo apt-get install autoconf automake autotools-dev curl python3 libmpc-dev libmpfr-dev libgmp-dev gawk build-essential bison flex texinfo gperf libtool patchutils bc zlib1g-dev libexpat-dev
- 4- sudo apt install swig3.0 libjpeg-dev lib-core doxygen python-sphinx sox graphicsmagick-libmagick-dev-compat libsdl2-dev libswitch-perl libftdi1-dev cmake scons libsndfile1-dev
- 5- sudo pip3 install twisted prettytable pyelftools openpyxl xlswriter pyyaml numpy configparser pyvcd
- 6- sudo pip2 install configparser

Bu ve bundan sonraki tüm adımlar Ubuntu üzerinde gerçekleştirilmelidir.

Quartus 21.1 ve QuestaSim

QuestaSim, Quartus ile birlikte lisanslanabilir bir şekilde gelen bir simülasyon aracıdır. İlerleyen aşamalarda simülasyon yapabilmek için gereklidir. Kurulum için Marquette Üniversitesinden Cristinel Ababei'nin hazırlanmış olduğu aşağıda linki verilen adımların 1 ve 2.sinin yapılması yeterlidir.

http://dejazz.com/eece4740/lectures/1_Install_Quartus_Prime.pdf

- 1- Öncelikle <https://fpgasoftware.intel.com/21.1/?edition=liteplatform=linux> websitesine gidin.
- 2- Sol kısımda arama kısmına 21.1 yazın
- 3- Intel® Quartus® Prime Standard Edition Design Software Version 21.1.1 for Linux bağlantısına tıklayın
- 4- Bu bağlantıdaki yaklaşık 25 GB'lık .tar uzantılı dosyayı indirin ve indirdikten sonra bilgisayarınıza kurun. Kurulum esnasında simülasyon araçlarından QuestaSim'i seçmeyi unutmayın.
- 5- Questa lisansı almak için <https://licensing.intel.com/psg/s/sales-signup-evaluationlicenses> websitesine gidin.
- 6- Siteye edu.tr uzantılı mail adresiniz ile kayıt olun.
- 7- Kayıt olduktan ve hesabınızın mail adresini onayladıktan sonra, <https://licensing.intel.com/psg/s/sales-signup-evaluationlicenses> adresine gidin.
- 8- Questa*-Intel® FPGA Starter Edition SW-QUESTA 'yı seçip, # of Seats sütunundaki değeri 1 yapın ve sağ alttan Get Licence'a tıklayın

9- Generate licence penceresi açılacaktır. Bu pencerede Create Computer dedikten sonra, computer name'e herhangi bir şey yazabilirsiniz. Computer Type -i NIC ID, Licence Type -i Fixed olacak şekilde seçimleri yapın.

10- Primary Computer id kısmı için eğer windowstaysanız konsolu açın ve ipconfig /all yazın. Bu komutu yazdıktan sonra gelecek olan bilgilerden, bilgisayarınız Türkçe ise "Fiziksel Adres", İngilizce ise "Physical Address" olarak önünüze 12 adet hexadecimal sayıdan oluşan bir seri gelecektir. Bu sayıyı kopyalayıp Primary Computer ID kısmına yapıştırın.

11- Tüm adımları yaptıktan sonra mail adresinize mail ile birlikte .dat uzantılı bir lisans dosyası yollanacak. Bu dosyayı ubuntuda iken indirin.

12- Şimdi ubuntuda bir terminal açın ve sadece "cd" yazarak root'a gidin. Terminale "vim .bashrc" yazdıktan sonra bu dosyanın en alt kısmına "export LM_LICENSE_FILE=LISANSKONUMU" satırını yazın. Bu satırı yazdıktan sonra sırasıyla "ESC, :x , Enter" komutlarını uygulayın. Sizi terminalde geri döndürecek ve dosyayı kaydedecek. Sonrasında source .bashrc yazarak bu dosyayı kalıcı olarak bilgisayarınızda path'e eklemiş olacaksınız.

— Örnek komut: export LM_LICENSE_FILE=/home/erkmen/Desktop/LR-102459_License.dat

13- Yine terminale yalnızca "cd" yazarak root'a gidip vim .bashrc yazarak bu dosyanın en sonuna kurulmuş olan Quartus'un dosyalarındaki questa'nın bin klasörü kalıcı olarak pathe eklenmelidir. (Insert tuşu ile editleme moduna geçebilirsiniz) Örnek olarak:

```
export PATH=/home/erkmen/Desktop/intelFPGA/21.1/questa_fe/bin:$PATH
```

Bu adımdan sonra yine ESC, :x, Enter komutlarını uygulayıp source .bashrc yaparak kalıcı olarak PATH'inize QuestaSIM'i eklemiş olacaksınız.

Vivado 2018.3

PULPissimo'yu FPGA'e atmak için üretilcek bit dosyası, Vivado 2018.3 tarafından sorunsuz şekilde implemente edilmektedir. Bu sebeple bu versiyon kullanılmalıdır.

1- <https://www.xilinx.com/support/download.html> websitesine gidin. 2- Sol taraftaki menüden "Vivado Archive" seçeneğine tıklayın. 3- "Vivado HLx 2018.3: All OS installer Single-File Download" kısmından Vivado'yu indirin, .tar'dan çıkartın ve kurun.

RISCV-GNU-Toolchain

RISCV-Gnu-Toolchain, içerisinde bir çok tool ile birlikte, bir C dosyasını RISCV komutlarına çevirebilen GCC'yi de içermektedir. RISCV-GNU-TOOLCHAIN'in içerisindeki GDB ile JTAG üzerinden PULPissimo programlanabildiği gibi, GCC ile derlenmiş dosyayı disassembly yapabilme özelliğine sahip tool gibi birçok tool bulunmaktadır.

1- Yeni bir terminal açıp, RISCV-GNU-Toolchain'in kurulum dosyalarını indirmek istediğiniz klasöre "cd" komutları yardımı ile gidin. (Örneğin: cd /Desktop/kurulum)

2- git clone <https://github.com/pulp-platform/pulp-riscv-gnu-toolchain> komutunu uygulayın.

3- Kurulum dosyalarını indirdikten sonra terminalde bu dosyanın içine girin ve sırasıyla
"./configure --prefix=/opt/riscv --with-arch=rv32imc --with-cmodel=medlow --enable-multilib"
ve

"make" komutlarını uygulayın. Burada dikkat etmeniz gereken şey, --prefix=/opt/riscv kısmına istediğiniz dosya konumunu ekleyebilirsiniz. Bu dosya konumunu masaüstünüzde ayrıca bir RISCV-GNU-ROOT gibi bir adda klasör oluşturup bu klasörü de kurulum lokasyonu olarak tanımlayada

bilirsiniz. (Örneğin -prefix=/home/erkmen/Desktop/RISCV-GNU-FOLDER yazarsanız masaüstünüzde oluşturduğunuz RISCV-GNU-FOLDER dosyasına kurulum yapılacaktır.) Dipnot: Kurulum süresi biraz uzun olacaktır.

4- Kurulum bittikten sonra yeni bir terminal açın ve sadece "cd" yazarak root'a gidin. Burada yine "vim .bashrc" komutuyla açılan dosyanın en altına sırasıyla şu komutları yazın (Insert tuşu ile editleme moduna geçebilirsiniz):

```
export PULP_RISCV_GCC_TOOLCHAIN=TOOLCHAINI_KURDUGUNUZ_DOSYA_KONUMU
```

```
export PATH=$PULP_RISCV_GCC_TOOLCHAIN/bin:$PATH
```

Örnek olarak yazılacak kodlar verilmiştir:

```
export PULP_RISCV_GCC_TOOLCHAIN=/home/erkmen/Desktop/RISCV-GNU-ROOT
```

```
export PATH=$PULP_RISCV_GCC_TOOLCHAIN/bin:* $PATH
```

Bu şekilde RISCV GNU Toolchain'i kalıcı olarak \$PATH'e eklemiş olacaksınız

PULPissimo

PULPissimo, ETH Zurich ve Bologna Üniversitesi tarafından işbirliği ile tasarlanmış açık kaynaklı bir mikrokontrolcüdür. PULPissimo'nun bu dökümantasyon sırasında 7. versiyonu kullanımdadır. Destekledikleri FPGA kartları github.com/pulp-platform/pulpissimo websitesinde verilmiştir.

- 1- Terminal açın ve Pulpissimoyu kuracağınız klasöre gidin. (Örn: cd /Desktop/proje)
- 2- Terminale git clone <https://github.com/pulp-platform/pulpissimo> yazarak pulpissimo reposunu çekin.
- 3- Terminale git clone <https://github.com/pulp-platform/pulp-runtime/> yazarak pulp-runtime reposunu çekin.
- 4- Terminale git clone <https://github.com/pulp-platform/pulp-runtime-examples> yazarak pulp-runtime-examples reposunu çekin.
- 5- Terminale "cd pulpissimo" yazarak çektiğiniz pulpissimo klasörüne gidin.
- 6- Pulpissimo klasörünün içindeyken terminalde
curl -proto 'https' -tlsv1.2 <https://pulp-platform.github.io/bender/init-sSf> — sh
komutunu uygulayarak bender'i indirmiş olacaksınız.
- 7- Bender'i pulpissimo klasörünün içerisine indirdikten sonra, yine pulpissimo klasörünün içerisinde terminalde sırasıyla şu komutlar uygulanmalıdır:

```
git checkout dde07e1  
make checkout
```

Bu komutlar ile FPGA'e atılacak bit file hazırlanması için gereken IP'ler kurulmuş olacak.

Simülasyon

Bu kısım, PULPissimo kullanılarak bir C kodunun simüle edilmesini anlatmaktadır.

- 1- Terminalden pulp-runtime klasörünün içerisine gidin.
- 2- pulp-runtime klasöründe "source configs/pulpissimo_cv32.sh" komutunu uygulayın.
- 3- "cd ." komutu ile geri çıkın ve pulpissimo klasörünün içerisine gidin.
- 4- Pulpissimo klasörünün içerisinde "source setup/vsim.sh" komutunu uygulayın.
- 5- Sonrasında yine Pulpissimo klasörünün içerisinde "make build" komutunu uygulayın.
- 6- İşlem bittikten sonra, "cd ." komutu ile geri çıkın ve "cd pulp-runtime-examples" komutunu uygulayarak pulp-runtime-examples klasörünün içerisine gidin.
- 7- Bu klasörün içerisindeki hello klasörüne gidin (cd hello).
- 8- Hello klasörünün içerisinde "make clean all run" komutunu uygulayın. Bu komut ile hello.c dosyası RISCv komutlarına göre derlenecek ve sonrasında Questa tarafından PULPissimo üzerinde simülasyonu yapılacak.

Bit File Oluşturma

Bu kısım, PULPissimo'nun FPGA'lere yüklenebilmesi için gerekli olan bit dosyasının nasıl oluşturulacağını ve nasıl FPGA'e yükleneceğini anlatmaktadır.

Terminale "sudo apt-get install cutecom" komutunu yazın.

- 1- Terminal açın ve pulpissimo klasörünün içerisine gidin.
- 2- "make scripts" komutunu uygulayın.
- 3- İşlem bittikten sonra "cd ." komutu ile geri çıkın ve "cd pulp-runtime" komutu ile pulp-runtime klasörünün içerisine gidin.
- 4- pulp-runtime klasörünün içerisinde "source configs/pulpissimo_cv32.sh" komutunu uygulayın.
- 5- Yine pulp-runtime klasörünün içerisinde "source configs/fpgas/pulpissimo/nexys.sh" komutunu uygulayın. **NOT: Burada nexys.sh , nexys kartları için gerekli olan dosyadır. Desteklenen kartlardan hangisi ile çalışıyorsanız, burada bulunan dosyalardan o karta ait dosyayı sourcelamalısınız.**
- 6- "cd ." ile geri çıkıp "cd pulpissimo/fpga" ile pulpissimo klasörünün fpga klasörüne gidiniz. Bu klasörde "make nexys rev=nexysA7-100T" komutunu uygulayınız. **NOT: Bu kısımda Nexys A7-100T'ye PULPissimo bit file'ı oluşturmak için gerekli parametreler girilmiştir. Diğer kartlar için bu parametrelerin ne olduğunu öğrenmek istiyorsanız, <https://github.com/pulp-platform/pulpissimo/tree/master/fpga> bağlantısına giderek kullandığınız kartın reposuna gidin ve aşağıda bulunan README kısmını okuyun.**
- 7- Terminalde bit file oluşturma süreci biraz uzun sürecektir. Bit file oluşturma süreci bittikten sonra bulunduğunuz fpga klasöründe bit file oluşmuş olacak.
- 8- FPGA kartınızı USB ile bilgisayarınıza bağlayın ve programlama modunu açın.
- 9- Terminal'e "vivado" yazarak Xilinx Vivado 2018.3'ü açın.
- 10- Vivado'ya kartınızı bağlayın ve program device seçeneği ile pulpissimo/fpga klasöründeki bit file'ı seçerek programlayın.
- 11- Eğer PULPissimo FPGA kartına doğru bir şekilde yüklendiyse, LED0 yanacaktır. **NOT: Güç bağlantısını kestiğiniz anda yüklü olan pulpissimo FPGA'den silinecektir. Bu yüzden fpga kartını bundan sonraki süreç boyunca bilgisayarınızdan çıkarmamalısınız.**
- 12- Hardware Manager ile FPGA kartının bağlantısını kestikten sonra önce hardware manager'i sonra Vivado'yu kapatın.

FPGA üzerinde C kodu kořturma

Bu kısımda pulpissimo yüklediđimiz FPGA üzerinde C kodu kořturmanın adımları anlatılmıřtır.

1- Öncelikle OpenOCD'nin kurulması gerekmektedir.

2- Terminal açın ve çalıştığınız ana klasörde "git clone <https://github.com/openocd-org/openocd>" komutunu uygulayın. 3- "cd openocd" ile bu klasörün içersine girin ve sırasıyla

"./bootstrap"

"./configure [options]"

"make"

"sudo make install" komutlarını çalıştırın. Bu şekilde OpenOCD kurulmuş olacak. 4- Şimdi pulp-runtime-examples/hello klasörüne gidin ve hello.c içersine yazdığınız C kodunu yapıştırın veyahut da Hello'yu görmek istiyorsanız dokunmayın. Terminal ile bu klasör içersinde "make clean all io=UART" komutunu uygulayın.

5- Kartınıza ait pulpissimo JTAG bağlantılarını <https://github.com/pulp-platform/pulpissimo/tree/master/fpga> bağlantısında kendi fpga kartınızın reposuna giderek README kısmından bulun ve bağlantıları yapın. Bağlantıları yaptıktan sonra harici JTAG programlayıcı bilgisayarınıza USB ile bağlayın.

6-Yen bir terminal açın ve bu terminalde "openocd -f .../pulpissimo/fpga/pulpissimo-nexys/openocd-nexys-hs2.cfg" komutunu uygulayın. **NOT: Bu komut Nexys kartları için geçerlidir. Diğer kartlar için kendilerine ait cfg dosyalarını seçmelisiniz. (...) diye belirtilen kısım pulpissimonun bulunduğu klasörün konumudur. Örneđin:**

openocd -f /home/erkmen/Desktop/kurulum/pulpissimo/fpga/pulpissimo-nexys/openocd-nexys-hs2.cfg

7- Yeni bir terminal daha açın ve bu terminalde

"riscv32-unknown-elf-gdb /home/erkmen/Desktop/kuurlum/pulp-runtime-examples/hello/build/test/test"

komutunu uygulayın. 8- Bu şekilde gdb açılacak. Şimdi elimizde 2 terminal var. Birinde OpenOCD, diğersinde gdb açık. 3. bir terminal açın ve bu terminalde "cutecom" komutu ile cutecom'u açın.

9- Cutecom'dan FPGA'inizin bağlı olduđu USB portunu, 115200 baudrate ile dinlemeye alın. 10- gdb terminaline dönün ve sırasıyla "load", ve sonrasında "continue" komutlarını uygulayın. Cutecom'dan çıkışı görmemiz gerekmektedir.

Süreç sırasında birçok hata alabilirsiniz, bunların çoğunun çözümü pulpissimo github sayfasının issues kısmında bulunmakta. Diğer basit hataları da google'a yazarak bulabilirsiniz.

Sorularınız için mail adresim: erkmen17@itu.edu.tr