Vitis HLS Tool Command Line Interface

**Vitis HLS Tool Command Line Interface**

**2022.1**

# Özet

Bu laboratuvar, Vitis™ HLS komut satırı aracı arayüzünde temel eylemlerin nasıl gerçekleştirileceğini sunar (CLI).

Bu eğitim yaklaşık 30 dakika sürmektedir.

# Yalnızca CloudShare Kullanıcıları

Bir laboratuvara erişmeniz için size üç deneme hakkı verilir ve her bir laboratuvarı tamamlamanız için ayrılan süre, laboratuvarı tamamlamanız için beklenen sürenin 2 katıdır. Zamanlayıcı geri saymaya başladığında onu duraklatamazsınız. Ayrıca, her deneme önceki denemeyi sıfırlar; yani, önceki denemedeki ilerlemeniz kaydedilmez.

# Hedefler

After completing this lab, you will be able to:

* Vitis HLS tool Clı da yeni bir proje yaratma
* Kendi kendini kontrol eden bir test ortamında bir C tasarımı olarak simüle etme
* Tasarımı sentezleme
* C test tezgâhı kullanarak bir RTL tasarımını simüle etme
* Tasarımı uygulamak

# Giriş

Bu laboratuvar, Vitis Yüksek Düzeyli Sentez (HLS) aracı Komut Satırı Arayüzü (CLI) akışının ana özelliklerine bir giriş sağlar. Bir proje oluşturmak için CLI modunda Vitis HLS aracını kullanacaksınız. Ayrıca sağlanan tasarımı simüle edecek, sentezleyecek ve uygulayacaksınız.

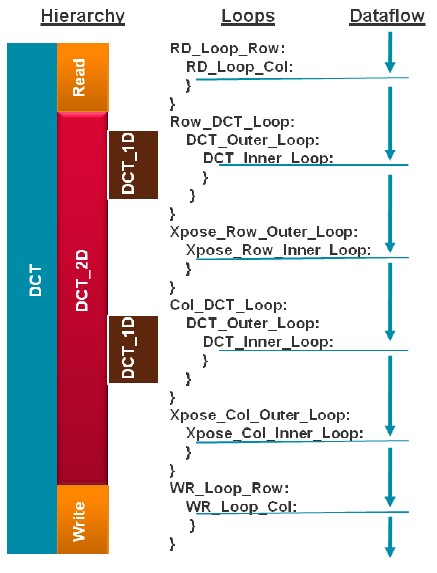
Bu tasarım, ayrık bir kosinüs dönüşümü (DCT) uygular ve C kaynağı olarak sağlanır.

İşlev, önce bir 1D DCT aracılığıyla giriş dizisinin her satırını işleyerek, ardından elde edilen dizinin sütunlarını aynı 1D DCT aracılığıyla işleyerek bir 2D DCT algoritmasından yararlanır. read\_data, dct\_2d ve write\_data işlevlerini çağırır.

read\_data işlevi 54. satırda tanımlanır ve iki döngüden oluşur: RD\_Loop\_Row ve RD\_Loop\_Col. write\_data işlevi 66. satırda tanımlanır ve sonucun yazılması için iki döngüden oluşur. 23. satırda tanımlanan dct\_2d işlevi, dct\_1d işlevini çağırır ve devrik gerçekleştirir.

Son olarak, 4. satırda tanımlanan dct\_1d işlevi, dct\_coeff\_table'ı kullanır ve 1D Type-II DCT algoritmasının temel yinelemeli biçimini uygulayarak gerekli işlevi gerçekleştirir

Aşağıdaki şekil, sol tarafta fonksiyonun prensibini, yürütüldükleri sırayla döngüleri ve sağ tarafta veri akışını göstermektedir.



**Şekil 2-1: Tasarım Hiyerarşisi ve Veri Akışı**

**Laboratuvar Ortamını Anlama**

Bu kursta sağlanan laboratuvarlar ve demolar, bir Linux platformunda çalışacak şekilde tasarlanmıştır.

Bir ortam değişkeni gereklidir: TRAINING\_PATH, laboratuvar dosyalarının bulunduğu yeri gösterir. Bu değişken, CloudShare/CustEd\_VM ortamlarında yapılandırılmış olarak gelir.

Bazı araçlar bu ortam değişkenini doğrudan kullanabilir (yani, $TRAINING\_PATH genişletilecektir) ve bazı araçlar manuel genişletme gerektirir (/home/xilinx/training için CloudShare/CustEd\_VM ortamları). Laboratuvar talimatları, her bir alet için ne yapılması gerektiğini açıklar.

Hem Vivado Design Suite hem de Vitis platformu, birçok laboratuvar tarafından kullanılan bir Tcl ortamı sunar. Araç başlatıldığında, araçların önceki başlatılmasından kalan hiçbir işlem veya değişken olmadan temiz bir Tcl ortamıyla başlar.

Bu, bir Tcl komut dosyasına kaynak sağladıysanız veya herhangi bir Tcl değişkenini manuel olarak ayarladıysanız ve aracı kapattıysanız, aracı yeniden açtığınızda, Tcl komut dosyasını yeniden kaynaklamanız ve laboratuvarın gerektirdiği tüm değişkenleri ayarlamanız gerekeceği anlamına gelir. Bu aynı zamanda terminal pencereleri için de geçerlidir; yeni bir terminal açıldığında tüm değişken ayarlar silinecektir.

# Genel Akış

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Aşama 1:  Vitis HLS Takım Projesi Yaratmak |  | Adım 2:  Çalıştırma  Simülasyon & Sentezleme | |  | |  | | --- | | Aşama 3:  Ortak simülasyon  & Dışa Aktarma | |

## Vitis HLS Aracı Projesi Oluşturma Adım 1

Bu adımda, yeni bir Vitis HLS aracı projesi oluşturacak, kaynak dosyaları ekleyecek ve Vitis HLS aracı komut istemini kullanarak varsayılan çözüm için çözüm ayarları sağlayacaksınız. Daha sonra, uygulanmış ayarların hızlı bir şekilde gözden geçirilmesi için oluşturulan projeyi Vitis HLS aracı GUI'sinde açacaksınız.

**1-1. Vitis HLS aracı komut istemini başlatın ve dizini şu şekilde değiştirin:**

**$TRAINING\_PATH/hls\_cli\_flow/lab/[zcu104 | vck190] working directory.**

**1-1-1. [Windows 10 kullanıcıları]:** Select **Start** > **Xilinx Design Tools** > **Vitis HLS 2022.1 Command Prompt** to launch the Vitis HLS tool command prompt.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**metin içeren bir resim

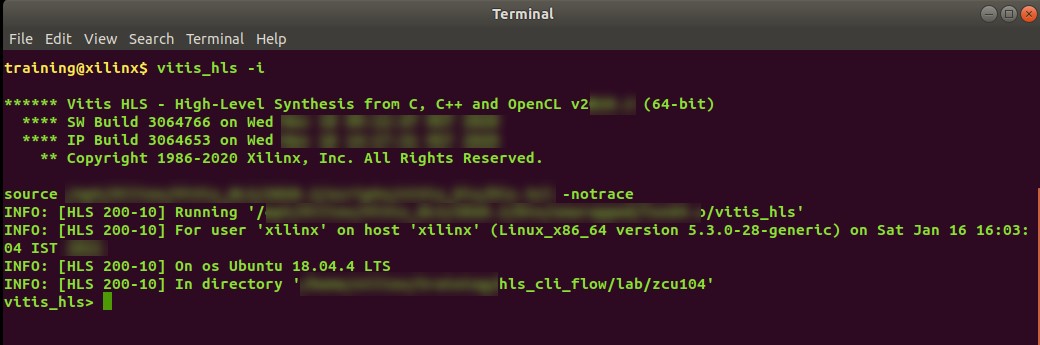
Açıklama otomatik olarak oluşturulduŞekil 2-2: Vitis HLS Komut İstemi**

**[Linux kullanıcıları]:** Set up the Vitis HLS environment. o Press <**Ctrl** + **Alt** + **T**> to open a new terminal window. o Change the directory to the Vitis HLS tool installation path:

**[host] $ cd $TRAINING\_PATH/hls\_cli\_flow/lab/[zcu104 | vck190]** o Enter the following command to set up the Vitis HLS environment appropriately:

**[host] $ source /opt/Xilinx/Vitis/2022.1/settings64.sh**

o Type **vitis\_hls -i** in the terminal and press <**Enter**> to launch the Vitis HLS tool command line interface.



**Şekil 2-3: Linux'ta Vitis HLS Aracı Komut İstemi**

**1-2. Yeni bir proje oluşturmak, kaynak dosyaları ilişkilendirmek ve çözüm ayarlarını yapılandırmak için komutları yazın.**

**1-2-1.** Aşağıdaki konuma göz atın ve tercih ettiğiniz metin düzenleyici ile açın **dct.tcl** .

$TRAINING\_PATH/hls\_cli\_flow/lab/[zcu104 | vck190]

**1-2-2. 2** numaralı satıra aşağıdaki komutu girin. of *dct.tcl* to create a new project named *dct\_prj*:

### open\_project -reset dct\_prj

**Soru 1**

Önceki komutta bulunan –reset anahtarı ne yapar?

proje verilerini sıfırlar. Daha sonra projeyi yeniden çalıştırınca ,sıfırdan bir proje oluşturulmasını sağlar

**1-2-3.** 4numaralı satıra aşağıdaki komutu girin. dct'yi tasarımda sentezlenecek üst düzey biri şlev olarak belirtmek için: **set\_top dct**

**1-2-4.** 6 numaralı satıra projeye bir tasarım kaynak dosyası olarak dct.c eklemek için aşağıda belirtilen komutu girin:

### add\_files dct.c

**Soru 2**

Yukarıdaki komutun, projeye test ortamı dosyaları eklemekten farkı nedir?

add\_files'a "-tb" eklenmesidir.

**1-2-5.** dct\_test.c, in.dat ve out.golden.dat'ı projeyle test ortamı kaynak dosyaları olarak ilişkilendirmek için sırasıyla 8, 9, 10 satır numaralarına aşağıdaki komutları girin:

### add\_files –tb dct\_test.c add\_files –tb in.dat add\_files –tb out.golden.dat

**1-2-6.** çözüm1 adlı projeye çözüm oluşturmak için 12numaralı satıra aşağıdaki komutu girin:

### open\_solution solution1 -reset

**1-2-7** cihazı çözümle ilişkilendirmek için 14 numaralı satıra komutu girin**:**

For the ZCU104 board:

**set\_part {xczu7ev-ffvc1156-2-e}**

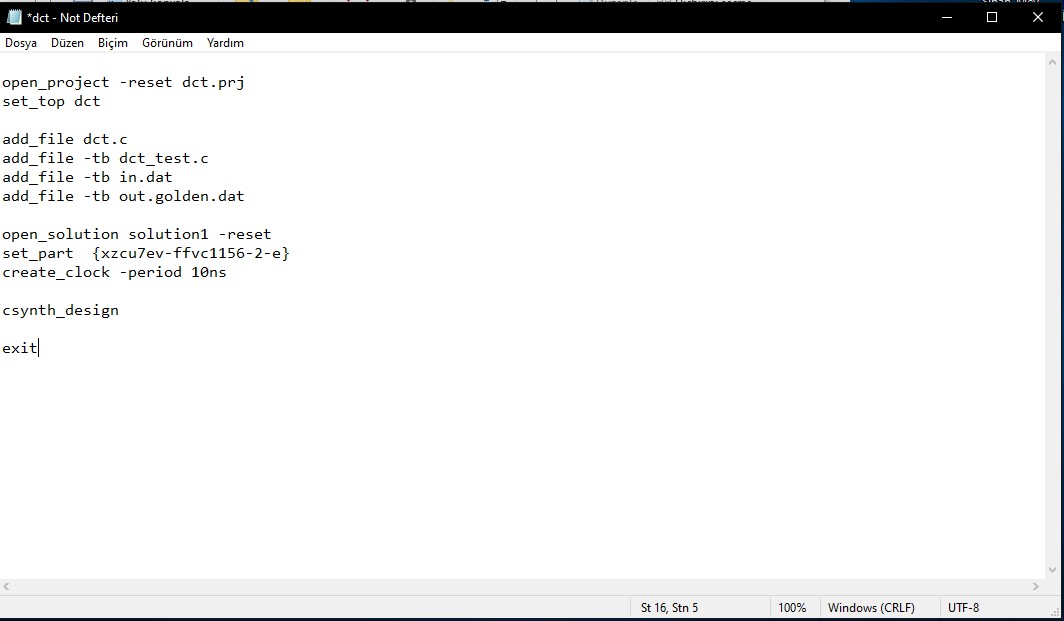
For the VCK190 board:

**set\_part {xcvc1902-vsva2197-2MP-e-S}**

**1-2-8** 16 numaralı satıra saati 10ns periyoduyla çözüm1 ile ilişkilendirmek için komutu girin:

### create\_clock -period 10ns

Yukarıdaki komutları yazdıktan sonra dct.tcl dosyası aşağıdaki şekildeki gibi görünmelidir..



**Şekil 2-4: Proje Bilgileri Örneği ile dct.tcl (ZCU104)**

**1-2-9.** dct.tcl dosyasını kaydet.

**1-2-10.** Tcl dosyasını kapatmak için metin düzenleyiciden çıkın.

**1-3. Projeyi oluşturmak için dct.tcl dosyasını çalıştırın ve oluşturulan projeyi Vitis HLS aracı GUI'sinde açın.**

**1-3-1.** *dct.tcl dosyasını çalıştırmak için Vitis HLS aracı komut istemine aşağıdaki komutu girin*

*dosya:*

### vitis\_hls -f dct.tcl

**Not**: "Çok fazla konumsal argüman belirtildi" şeklinde bir hata alırsanız, Tcl dosyasına geri dönün ve '–' yerine '-' koyun. Bu hata, komutu Windows ortamından VM'ye kopyaladığınızda, Linux VM'nin komutu farklı şekilde ele alması nedeniyle oluşur.

metin içeren bir resim

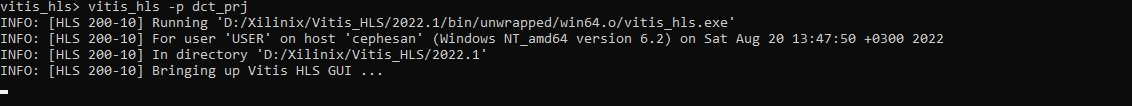
Açıklama otomatik olarak oluşturuldumetin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 2-5: Proje Oluşturma**

**1-3-2** Yakın zamanda oluşturulan projeyle GUI'yi başlatmak için aşağıdaki komutu girin**:**

### vitis\_hls -p dct\_prj



**Şekil 2-6: GUI'yi CLI'den başlatma**

Oluşturulan projeyi Explorer görünümünde görmelisiniz.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 2-7: Yeni Oluşturulan Proje ile Vitis HLS Aracı GUI**

**1-3-3.** Genişlet **dct\_prj** > **Source**.

Kaynak dosyanın eklendiğini gözlemleyin.

**1-3-4. dct.c** dosyasını açmak için çift tıklayın.

**1-3-5.** Seç **Project** > **Project Settings** Vitis HLS tool GUI.

Proje Ayarları iletişim kutusu açılır.

**1-3-6.** Tasarım ve test tezgahı kaynaklarının projeye Tcl dosyasına girildiği gibi eklendiğinden emin olmak için Simulation and Synthesis'e tıklayın.

**1-3-7.** Kaynak dosyalar doğrulandıktan sonra Proje Ayarları iletişim kutusunu kapatın.

**1-3-8.** Seç **Solution** > **Solution Settings**

Aktif çözüm için Çözüm Ayarları iletişim kutusu, yani bu mevcut projede var olan tek çözüm olan Çözüm 1 açılır

**1-3-9.** Saat frekansının ve parça numarasının Tcl dosyasında açıklananlar olduğundan emin olmak için Sentez'i seçin.

**1-3-10.** Saat frekansı ve cihaz ayarları doğrulandıktan sonra Çözüm Ayarları iletişim kutusunu kapatmak için İptal'e tıklayın.

**1-3-11.** Vitis Hls den çıkmak için **File** > **Exit**.

Vitis HLS Tool Command Line Interface

## C Simulation'ı Çalıştırma ve Tasarımı Sentezleme

## Adım[[1]](#footnote-1)

Bu adımda, C simülasyonunu çalıştıracak ve tasarımı Vitis HLS aracı komut isteminden sentezleyecek ve ardından proje durumunu ve Sentez raporunu gözden geçirmek için Vitis HLS aracı GUI'sini açacaksınız.

**2-1. Tasarımı simüle edin.**

**2-1-1.** C simülasyonunu çalıştırmak için Vitis Hls aracı komut istemine aşağıdaki komutu girin: csim\_design

Bu komut, sağlanan C test ortamını kullanılarak tasarımın sentez öncesi C simülasyonunu derler ve çalıştırır.

**-1-2.** Komut isteminde Sonuçlar iyi mesajına dikkat edin.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**2-2. Tasarımı Sentezleyin.**

**2-2-1** Tasarımı sentezlemek için Vitis HLS aracı komut istemine aşağıdaki komutu girin:

### csynth\_designmetin içeren bir resim Açıklama otomatik olarak oluşturuldu

Bu, projeye eklenen kaynak dosyalar üzerinde detaylandıracak ve üst düzey sentez gerçekleştirecektir. Bu aynı zamanda kaynak dosyalarını da analiz eder, varsa yönergeleri doğrular ve bazı ilk kod dönüşümlerini gerçekleştirir.

**Şekil 2-9: Komut İsteminden Sentez Gerçekleştirme**

**2-3. Vitis HLS aracı GUI'sinde Sentez raporunu doğrulayın.**

**2-3-1.** GUI'yi başlatmak için aşağıdaki komutu girin:

### vitis\_hls –p dct\_prj

**2-3-2.** Explorer bölmesind**e dct\_prj** > **solution1** > **syn** > **report** klasörünü genişletin.

**2-3-3.** Sentez raporunu görüntülemek içindct\_csynth.rpt dosyasına çift tıklayın.

**2-3-4.** Rapor dosyasını analiz edin.

**tablo içeren bir resim

Açıklama otomatik olarak oluşturuldumetin içeren bir resim

Açıklama otomatik olarak oluşturuldutablo içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Soru 3**

Sentez raporundan aşağıdaki ayrıntıları yazın:

* Tahmini saat frekansı: 3.59
* En kötü senaryo gecikmesi: 415
* BRAM\_18K numarası: 17
* kullanılan DSP48E: 16
* kullanılan FFs numaraları: 671
* Kullanılan LUTs numaraları: 1992

**2-3-**5 Sentez raporunda **Arayüz >** Özet'e gidin.

Rapor ayrıca araçlar tarafından üretilen üst düzey arayüz sinyallerini de gösterir.

**2-3-6.** Uygulamadan çıkmak için **dosya > çıkışı** seçin.

## RTL’yi birlikte simüle etme ve dışa aktarma Adım 3

Bu adımda, Vitis HLS aracı komut isteminden C test tezgahını kullanarak oluşturulan RTL dosyaları üzerinde C/RTL ortak simülasyonu gerçekleştireceksiniz. Ayrıca, IP'yi Vitis HLS aracı komut isteminden oluşturacaksınız..

**3-1. C/RTL ortak simülasyonu gerçekleştirin.**

**3-1-1.** Sentez sonrası ortak simülasyonu yürütmek için Vitis HLS aracı komut istemine aşağıdaki komutu girin: **cosim\_design**

metin içeren bir resim

Açıklama otomatik olarak oluşturuldumetin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 2-10: C/RTL Eş Simülasyonu**

**\*\*\* C/RTL co-simulation finished: PASS \*\*\*** adlı mesajı farkedin.

**3-2.** Tasarımı IP’ye aktarın**.**

**3-2-1.** Verilog HDL için bir RTL uygulaması gerçekleştirmek için aşağıdaki komutu girin:

### export\_design -flow impl -format ip\_catalog

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 2-11: Sentezlenmiş RTL'yi IP Örneği Olarak Dışa Aktarma(ZCU104)**

**Not:** Uyarıları dikkate almayın

-flow seçeneği, verilen seçeneğe (syn | impl) dayalı olarak oluşturulan IP üzerinde RTL sentezi veya RTL sentezi ve uygulamasını gerçekleştirir. Uygulama, RTL'nin tahmini zamanlama ve alan hedeflerini karşılayacağını değerlendirmek ve güven sağlamak için yürütülür. Bu sonuçlar, ihraç edilen paketin bir parçası olarak dahil edilmez.

-format seçeneği, IP'nin paketleneceği biçimi belirtir. Desteklenen formatlar aşağıdadır:

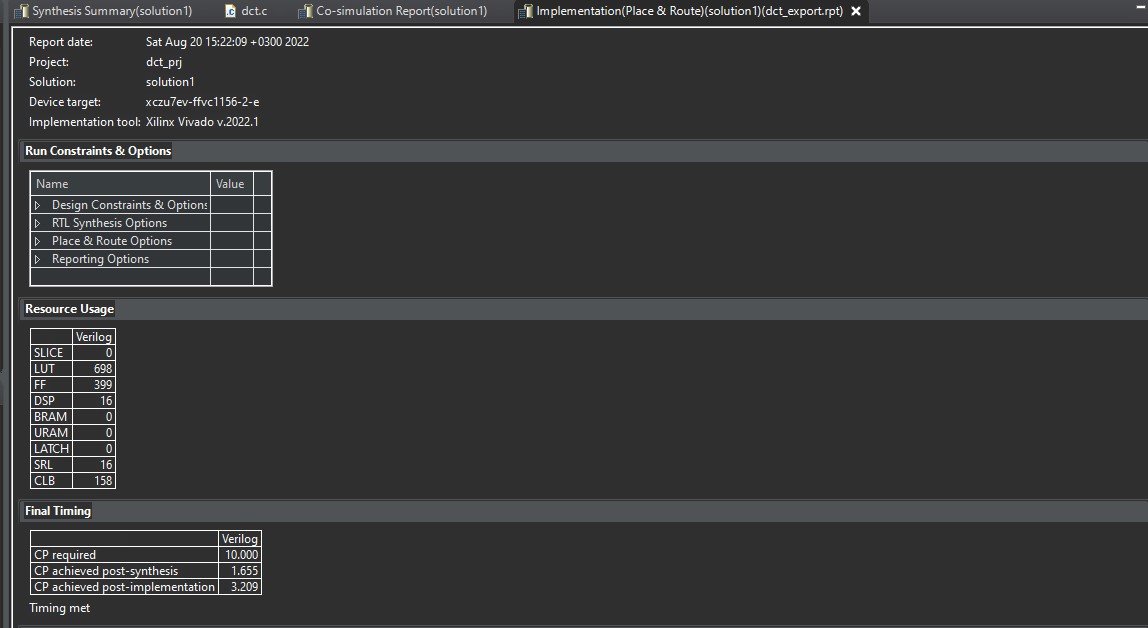
* ip\_catalog: Xilinx IP kataloğuna eklemek için uygun bir format.
* xo: Vitis ortamı uygulama hızlandırma akışında bağlantı oluşturmak için Vitis derleyicisi tarafından kabul edilen bir biçim.
* syn\_dcp: Vivado Design Suite için sentezlenmiş kontrol noktası dosyası. Bu seçenek kullanılırsa, RTL sentezi otomatik olarak yürütülür. Vivado uygulaması isteğe bağlı olarak eklenebilir.
* sysgen: System Generator'da kullanım için Vivado Design Suite IP ve .zip arşivi oluşturur.
* **3-2-2** Timing Met mesajını gözlemleyin.

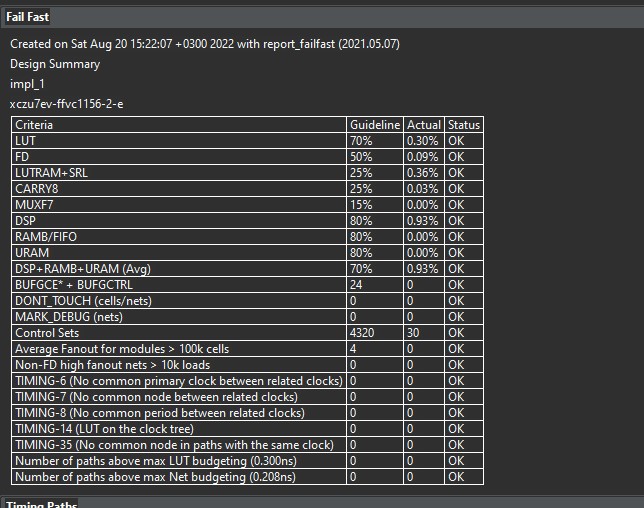
**3-3. Sonuçları Vitis HLS aracı GUI'sinde doğrulayın.**

**3-3-1.** GUI'yi başlatmak için aşağıdaki komutu girin:

### vitis\_hls –p dct\_prj

**3-3-2.** RTL Simülasyon raporunu açmak için Vitis HLS aracı GUI'sinde Çözüm> Raporu Aç > Kosimülasyon'u seçin.

**3-3-3.** Uygulama raporunu açmak için Çözüm > Raporu Aç > Uygulama > Yerleştir ve Yönlendir'i seçin. tablo içeren bir resim

Açıklama otomatik olarak oluşturuldutablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Not:** Bu raporları Flow Navigator'dan da C/RTL COSIMULATION ve UYGULAMA > Raporlar ve Görüntüleyici'yi genişleterek açabilirsiniz**.**.

**3-3-4.** Sonuçları analiz edin ve uygulamayı kapatın

**3-3-5.** Vitis HLS aracı komut istemini kapatın.

Some systems (particularly VMs) may be memory constrained. Removing the workspace frees a portion of the disk space, allowing other labs to be performed.

You can delete the directory containing the lab you just ran by using the graphical interface or the command-line interface. You can choose either mechanism. Both processes will recursively delete all the files in the $TRAINING\_PATH/hls\_cli\_flow directory.

**3-4. [Optional] [Only for local VMs—not for CloudShare] Clean up the file system.**

Using the GUI:

**3-4-1.** Using the graphical browser (Windows: press the <**Windows**> key + <**E**>; Linux: press <**Ctrl** + **N**>), navigate to $TRAINING\_PATH/hls\_cli\_flow.

**3-4-2.** Select **hls\_cli\_flow**.

**3-4-3.** Press <**Delete**>.

-- OR --

Using the command line:

**3-4-4.** Open a terminal window (Windows: press the <**Windows**> key + <**R**>, then enter **cmd**; Linux: press <**Ctrl** + **Alt** + **T**>).

**3-4-5.** Enter the following command to delete the contents of the workspace:

**[Windows users]:** **rd /s /q $TRAINING\_PATH/hls\_cli\_flow**

**[Linux users]:** **rm -rf $TRAINING\_PATH/hls\_cli\_flow**

# Summary

In this lab, you learned how to use the Vitis HLS tool command prompt to:

* Create the Vitis HLS tool project
* Create a solution with the desired settings
* Execute major tasks in the Vitis HLS design flow such as simulating, synthesizing, co-simulating, and exporting the RTL of the design

1. [↑](#footnote-ref-1)