Estimating Accelerator Performance

Introduction

This lab guides you through the steps involved in estimating the expected performance of an application when functions are implemented in hardware, without going through the entire build cycle.

Objectives

After completing this lab, you will be able to:

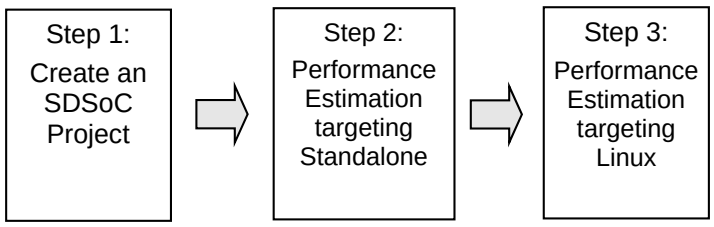
- Use the SDSoC environment to obtain an estimate of the speedup that you can expect from your selection of functions to accelerate
- Differentiate between the flows targeting Standalone OS and Linux OS

Procedure

This lab is separated into steps that consist of general overview statements that provide information on the detailed instructions that follow. Follow these detailed instructions to progress through the lab.

This lab comprises three primary steps: You will create an SDSoC project, estimate performance speedup targeting the Standalone OS and then estimate performance speedup targeting the Linux OS.

General Flow for this Lab



Create an SDSoC Project Step 1

1-1. Launch SDSoC and create a project, called *lab4*, using the *Empty Application* template and then using the provided source files, targeting the Zed or Zybo board and Standalone OS.

1-1-1. Open SDSoC, select **c:\xup\SDSoC\labs** as the workspace and click **OK**.

1-1-2. Create a new project called **lab4**, and select either *zybo* or *zed*

1-1-3. Select **Standalone** as the target OS, and click **Next**.

1-1-4. Select **Empty Application** and click **Finish**.

1-2. Import the provided source files from source\lab4\src folder.

1-2-1. Right click on *src* under **lab4** in the Project Explorer tab and select **Import…**

1-2-2. Click on **File System** under *General category* and then click **Next**.

1-2-3. Click on the **Browse** button, browse to *c:\xup\SDSoC\source\lab4\src* folder, and click **OK**.

1-2-4. Either select all the files in the right-side window or select *src* checkbox in the left-side window and click **Finish** to import the files into the project.

Performance Estimation Targeting Standalone Step 2

2-1. Mark sharpen_filter for the hardware acceleration. Run an initial performance estimate of the hardware only.

2-1-1. Click on the "+" sign in the Hardware Functions area to open up the list of functions which are in the source files.

2-1-2. Select the *sharpen_filter* function and click **OK**.

2-1-3. In the *Actions* panel of the SDSoC Project Overview, click on *Estimate Performance Speedup for HW functions*.

This selects the SDEstimate build configuration and performs the estimation flow.

2-1-4. The Build project dialog appears and asks if you want to build the project. Click **OK**.

The SDSoC environment builds the project. A dialog box displaying the status of the build process appears.

2-1-5. After the build is over, you can see an initial report. This report contains a hardware-only estimate summary and has a link that can be clicked to obtain the software run data, which updates the report with comparison of hardware implementation versus the software-only information.

Performance and resource estimation report for the 'lab4' project

Click Here **to get software-only application performance and speedup**

Note: Performance estimation assumes worst-case latency of hardware accelerators, it also assumes worst-case data transfer size for arrays (if transfer size cannot be determined at compile time). If the accelerator latency and data transfer size at run-time is smaller than such assumptions, the performance estimation will be more pessimistic than the actual performance.

Details

Performance estimates for 'sharpen_filter' function

HW accelerated (Estimated cycles)	9743741

Resource utilization estimates for hardware accelerators

Resource	Used	Total	% Utilization
DSP	0	220	0
BRAM	3	140	2.14
LUT	306	53200	0.58
FF	297	106400	0.28

(a) Zed

Performance and resource estimation report for the 'lab4' project

Click Here **to get software-only application performance and speedup**

Note: Performance estimation assumes worst-case latency of hardware accelerators, it also assumes worst-case data transfer size for arrays (if transfer size cannot be determined at compile time). If the accelerator latency and data transfer size at run-time is smaller than such assumptions, the performance estimation will be more pessimistic than the actual performance.

Details

Performance estimates for 'sharpen_filter' function

HW accelerated (Estimated cycles)	13502092

Resource utilization estimates for hardware accelerators

Resource	Used	Total	% Utilization
DSP	0	80	0
BRAM	3	60	5
LUT	276	17600	1.57
FF	207	35200	0.59

(b) Zybo

Figure 1. Initial estimate of hardware only performance

2-2. Connect and power up the board. Click on the `Click Here` link of the initial estimation report to run the application and get the entire application speedup.

2-2-1. Connect the board and power it ON.

2-2-2. Click on the **Click Here** link in the *SDSoC Report Viewer* tab to get the software only application performance and speedup.

Since the board is connected using JTAG and the OS is Standalone, the *Local* connection will be used.

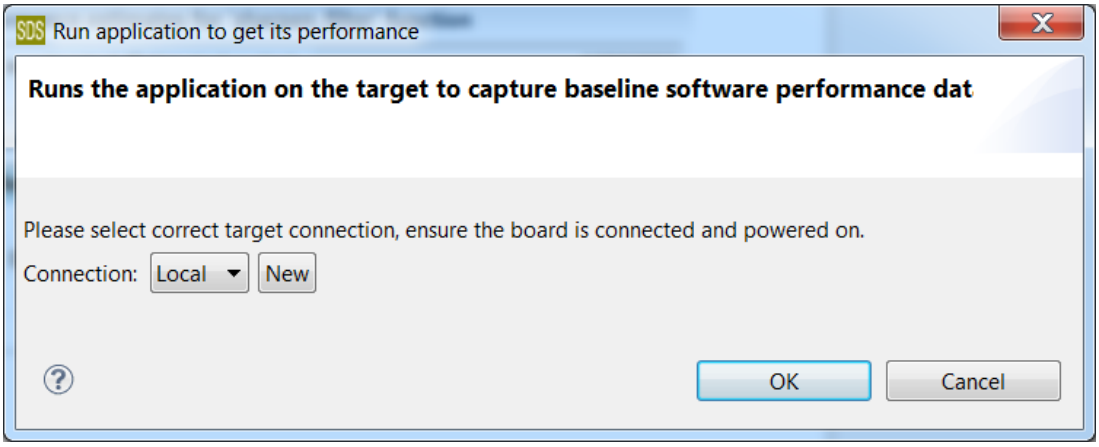


Figure 2. Making connection to download and running application

2-2-3. Click **OK**.

The default bitstream (without the accelerator functionality) and the application will be downloaded and executed.

Performance, speedup and resource estimation report for the 'lab4' project

Note: Performance estimation assumes worst-case latency of hardware accelerators, it also assumes worst-case data transfer size for arrays (if transfer size cannot be determined at compile time). If the accelerator latency and data transfer size at run-time is smaller than such assumptions, the performance estimation will be more pessimistic than the actual performance.

Summary

Performance estimates for 'main' function

SW-only (Measured cycles)	859112446
HW accelerated (Estimated cycles)	676601111
Estimated speedup	1.27

Details

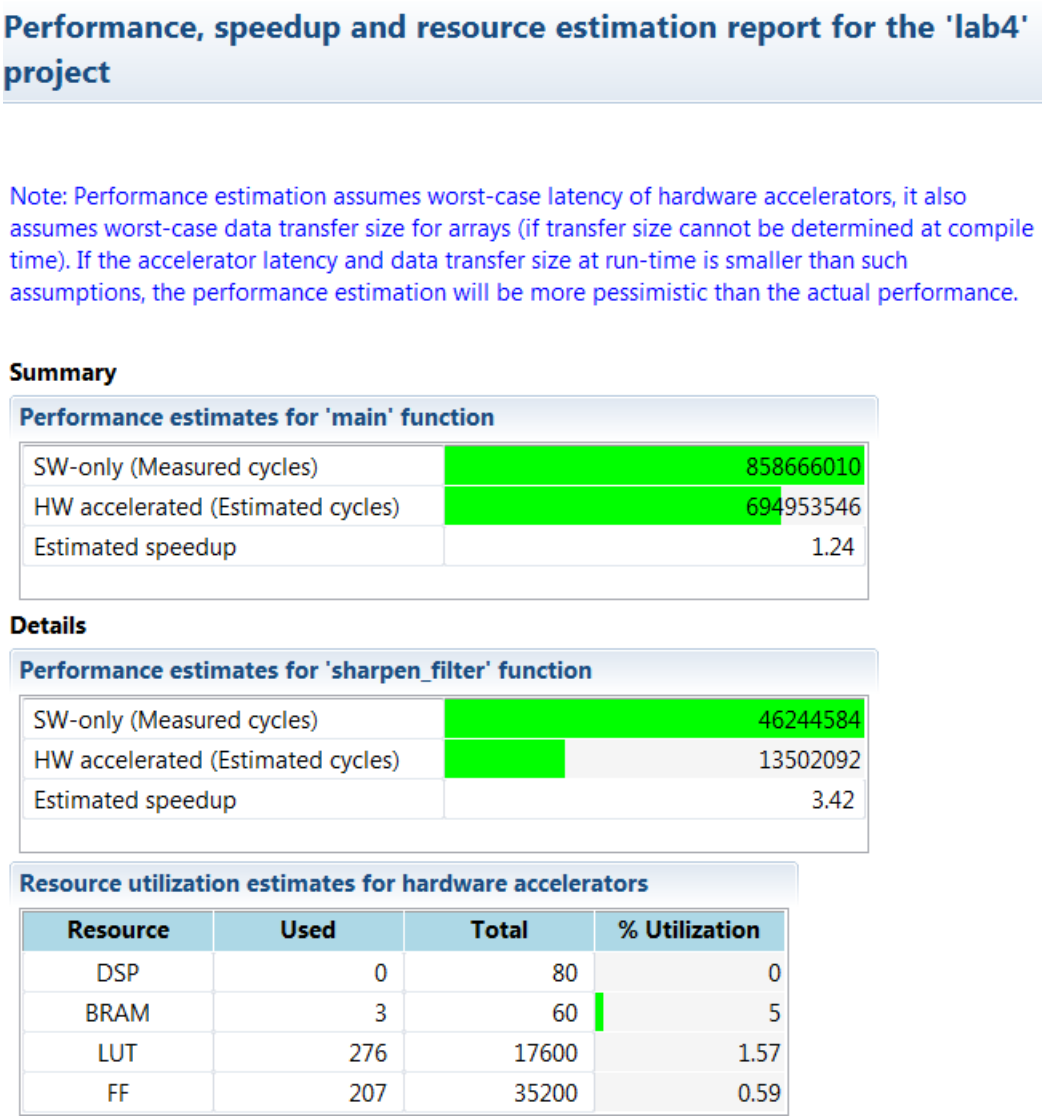
Performance estimates for 'sharpen_filter' function

SW-only (Measured cycles)	46246008
HW accelerated (Estimated cycles)	9743741
Estimated speedup	4.75

Resource utilization estimates for hardware accelerators

Resource	Used	Total	% Utilization
DSP	0	220	0
BRAM	3	140	2.14
LUT	306	53200	0.58
FF	297	106400	0.28

(a) Zed



Performance, speedup and resource estimation report for the 'lab4' project

Note: Performance estimation assumes worst-case latency of hardware accelerators, it also assumes worst-case data transfer size for arrays (if transfer size cannot be determined at compile time). If the accelerator latency and data transfer size at run-time is smaller than such assumptions, the performance estimation will be more pessimistic than the actual performance.

Summary

Performance estimates for 'main' function

SW-only (Measured cycles)	858666010
HW accelerated (Estimated cycles)	694953546
Estimated speedup	1.24

Details

Performance estimates for 'sharpen_filter' function

SW-only (Measured cycles)	46244584
HW accelerated (Estimated cycles)	13502092
Estimated speedup	3.42

Resource utilization estimates for hardware accelerators

Resource	Used	Total	% Utilization
DSP	0	80	0
BRAM	3	60	5
LUT	276	17600	1.57
FF	207	35200	0.59

(b) Zybo

Figure 3. Comparison between the pure software and hardware accelerated

The Summary section shows that the estimated speedup between the software only and one with the hardware accelerator is 1.24 whereas the Details section show the speedup of the targeted function to be 3.43 times.

2-3. Add sobel_filter for the hardware acceleration. Run an initial performance estimate of the hardware only.

2-3-1. Add the *sobel_filter* function to the accelerator list.

2-3-2. Select **lab4 > Clean Project**

2-3-3. In the *Actions panel* of the SDSoC Project Overview, click on *Estimate Performance Speedup for Hardware Functions*.

This selects the SDEstimate build configuration and performs the estimation flow.

2-3-4. The Build project dialog appears and asks if you want to build the project. Click **OK**.

2-3-5. After the build is over, the initial estimate and resources report will be displayed again.

Performance and resource estimation report for the 'lab4' project

Click Here to get software-only application performance and speedup

Note: Performance estimation assumes worst-case latency of hardware accelerators, it also assumes worst-case data transfer size for arrays (if transfer size cannot be determined at compile time). If the accelerator latency and data transfer size at run-time is smaller than such assumptions, the performance estimation will be more pessimistic than the actual performance.

Details

Performance estimates for 'sobel_filter' function

HW accelerated (Estimated cycles)	19479502

Details

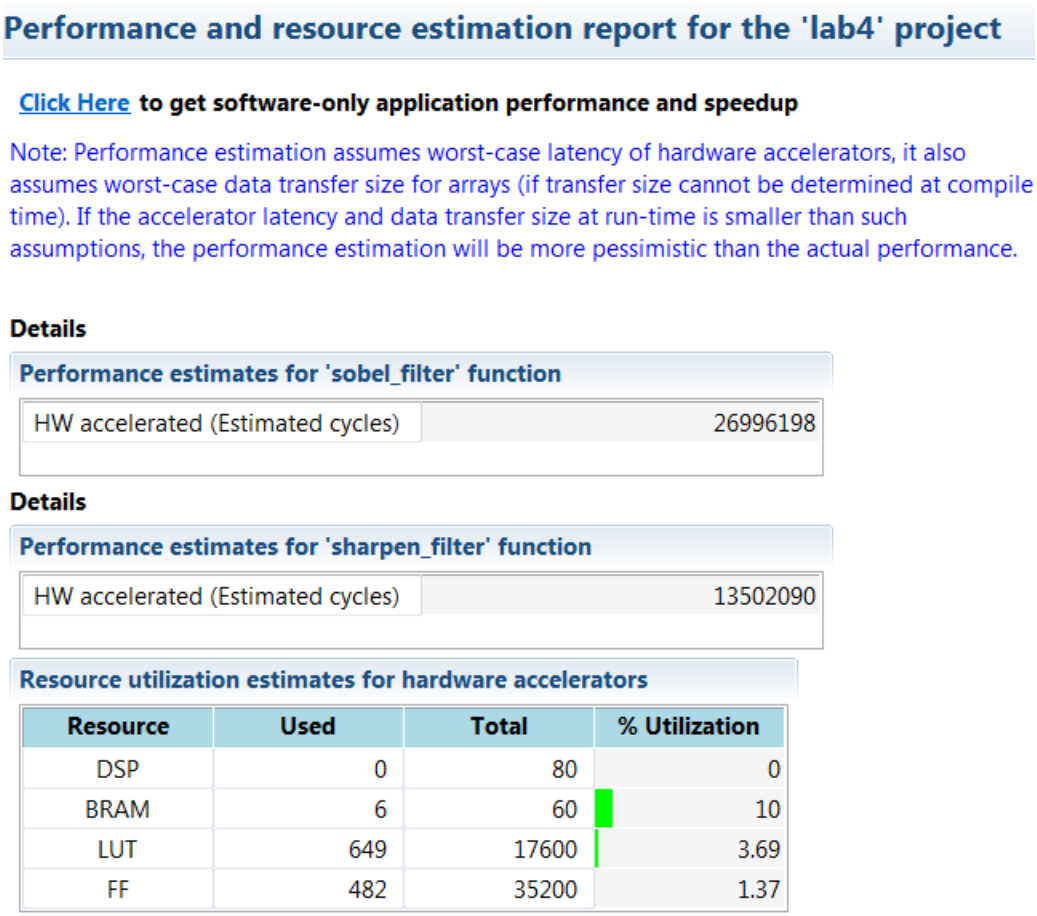
Performance estimates for 'sharpen_filter' function

HW accelerated (Estimated cycles)	9743739

Resource utilization estimates for hardware accelerators

Resource	Used	Total	% Utilization
DSP	0	220	0
BRAM	6	140	4.29
LUT	701	53200	1.32
FF	716	106400	0.67

(a) Zed



Performance and resource estimation report for the 'lab4' project

Click Here to get software-only application performance and speedup

Note: Performance estimation assumes worst-case latency of hardware accelerators, it also assumes worst-case data transfer size for arrays (if transfer size cannot be determined at compile time). If the accelerator latency and data transfer size at run-time is smaller than such assumptions, the performance estimation will be more pessimistic than the actual performance.

Details

Performance estimates for 'sobel_filter' function

HW accelerated (Estimated cycles)	26996198

Details

Performance estimates for 'sharpen_filter' function

HW accelerated (Estimated cycles)	13502090

Resource utilization estimates for hardware accelerators

Resource	Used	Total	% Utilization
DSP	0	80	0
BRAM	6	60	10
LUT	649	17600	3.69
FF	482	35200	1.37

(b) Zybo

Figure 4. Initial hardware acceleration estimate for the two functions

2-3-6. Click on the **Click Here** link in the SDSoC Report Viewer tab and click **OK**.

Summary

Performance estimates for 'main' function

SW-only (Measured cycles)	854650422
HW accelerated (Estimated cycles)	252169119
Estimated speedup	3.39

Details

Performance estimates for 'sobel_filter' function

SW-only (Measured cycles)	103473158
HW accelerated (Estimated cycles)	19479502
Estimated speedup	5.31

Performance estimates for 'sharpen_filter' function

SW-only (Measured cycles)	46246343
HW accelerated (Estimated cycles)	9743739
Estimated speedup	4.75

Resource utilization estimates for hardware accelerators

Resource	Used	Total	% Utilization
DSP	0	220	0
BRAM	6	140	4.29
LUT	701	53200	1.32
FF	716	106400	0.67

(a) Zed

Summary

Performance estimates for 'main' function

SW-only (Measured cycles)	853274366
HW accelerated (Estimated cycles)	308181878
Estimated speedup	2.77

Details

Performance estimates for 'sobel_filter' function

SW-only (Measured cycles)	103271261
HW accelerated (Estimated cycles)	26996198
Estimated speedup	3.83

Performance estimates for 'sharpen_filter' function

SW-only (Measured cycles)	46245524
HW accelerated (Estimated cycles)	13502090
Estimated speedup	3.43

Resource utilization estimates for hardware accelerators

Resource	Used	Total	% Utilization
DSP	0	80	0
BRAM	6	60	10
LUT	649	17600	3.69
FF	482	35200	1.37

(b) Zybo

Figure 5. Actual performance estimation with two functions in hardware

Performance Estimation Targeting Linux Step 3

3-1. Create a new empty application project called lab4a targeting Linux OS. Import the provided source files from source\lab4\src folder

For this portion of the lab, you will need an Ethernet port on the PC configured with 192.168.0.1 as an IP address and an Ethernet cable.

3-1-1. Select **File > New > SDSoc Project** to open the New Project GUI.

3-1-2. Enter **lab4a** as the project name, select either *zybo* or *zed* (depending on the board you are using) via the drop-down button, select *Linux* as the target OS, and click **Next**.

The Templates page appears, containing source code examples for the selected platform.

3-1-3. Select **Empty Application**.

3-1-4. Click **Finish**.

3-1-5. Right click on *src* under **lab4a** in the Project Explorer tab and select **Import…**

3-1-6. Click on **File System** under *General category* and then click **Next**.

3-1-7. Click on the **Browse** button, browse to *c:\xup\SDSoC\source\lab4\src* folder, and click **OK**.

3-1-8. Either select all the files in the right-side window or select *src* checkbox in the left-side window and click **Finish** to import the files into the project.

3-2. Mark sharpen_filter for the hardware acceleration. Run an initial performance estimate of the hardware only.

3-2-1. Click on the "+" sign in the Hardware Functions area to open up the list of functions which are in the source files.

3-2-2. Select the *sharpen_filter* function and click **OK**.

3-2-3. In the Actions panel of the SDSoC Project Overview, click on *Estimate Performance Speedup for Hardware Functions*.

This selects the SDEstimate build configuration and performs the estimation flow.

3-2-4. The Build project dialog appears and asks if you want to build the project. Click **OK**.

The SDSoC environment builds the project. A dialog box displaying the status of the build process appears.

3-2-5. After the build is over, you will see an initial report that is the same as Figure 1 since the same function has been targeted for acceleration.

3-3. Copy the sd_card contents to the SD Card. Configure the board to boot from SD card. Connect and power up the board. Configure the board's Ethernet address to 192.168.0.10 and the PC's to 192.168.0.1

3-3-1. Configure the board to boot from SD card.

3-3-2. Using the Windows Explorer copy the content of the **lab4a > SDEstimate > sdcard** onto the (micro) SD card. Insert the SD card into the board.

3-3-3. Connect the board, including network cable, and power it ON.

The board will boot. Make a serial connection using the appropriate COM port.

3-3-4. Press the *PS-SRST* button on the board to reboot and notice Linux booting.

3-3-5. Once the board boot is complete, set the ip address of the board to 192.168.0.10 by typing the following command at the Linux prompt:

```
ifconfig
```

Note if any address is being assigned.

If not assigned then execute the following command to assign to the correct Ethernet adaptor.

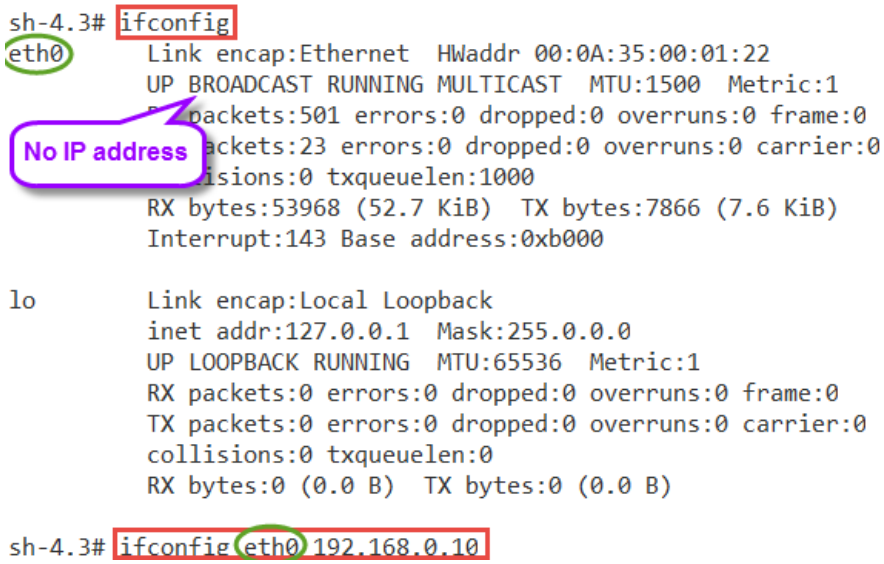


Figure 6. Assigning an IP address

3-3-6. Configure the Ethernet adaptor IP address on the Zynq board to 192.168.0.10

```
ifconfig eth0 192.168.0.10
```

3-3-7. Configure the PC Ethernet adaptor IP address to 192.168.0.1

3-3-8. Expand Linux TCF Agent in the Target Connection tab.

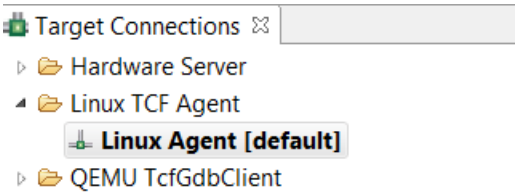


Figure 7. Configuring the Linux TCF Agent

3-3-9. Double-click on the **Linux Agent [default]** entry to open the connection form.

3-3-10. Set the *Host IP* address to **192.168.0.10**, the *Port* field to **1534,** and then click ***OK***.

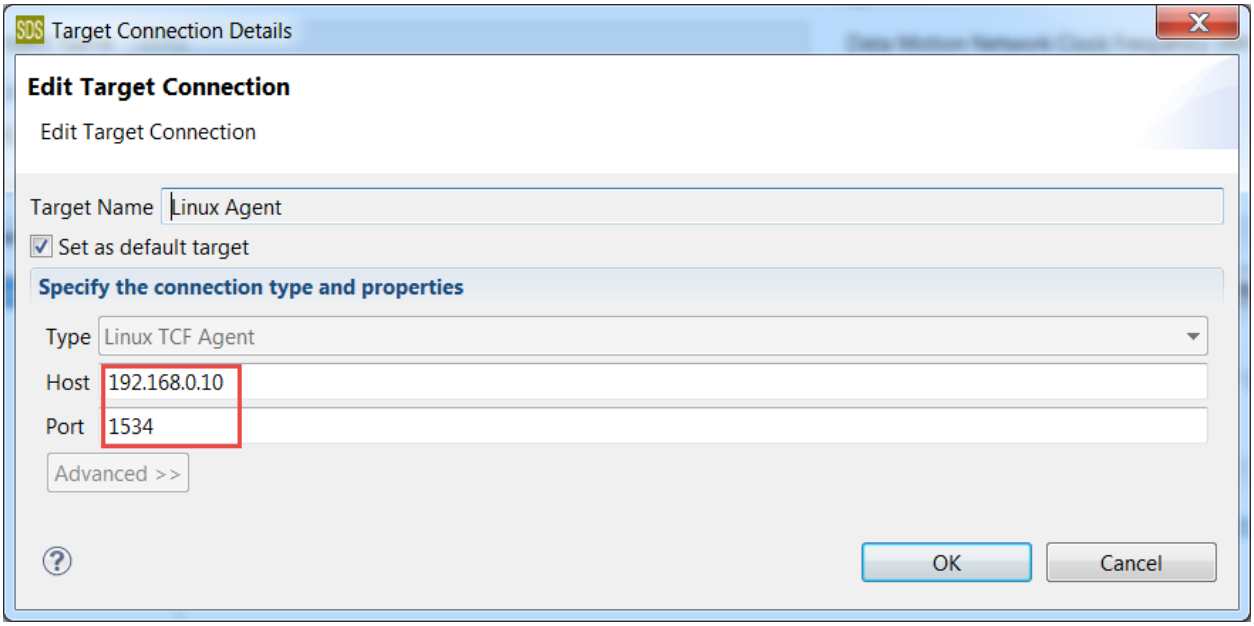
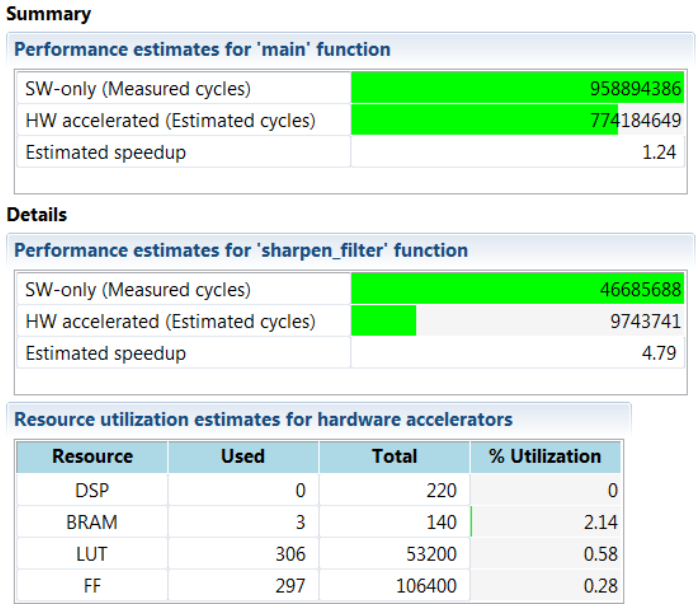


Figure 8. Making connection for Linux target

3-4. Estimate the accelerator speedup.

3-4-1. In the performance and resource estimate report, click on the **Click Here** link.

3-4-2. When the execution completes the performance estimate report will be displayed in the SDSoC report viewer.



Summary

Performance estimates for 'main' function

SW-only (Measured cycles)	958894386
HW accelerated (Estimated cycles)	774184649
Estimated speedup	1.24

Details

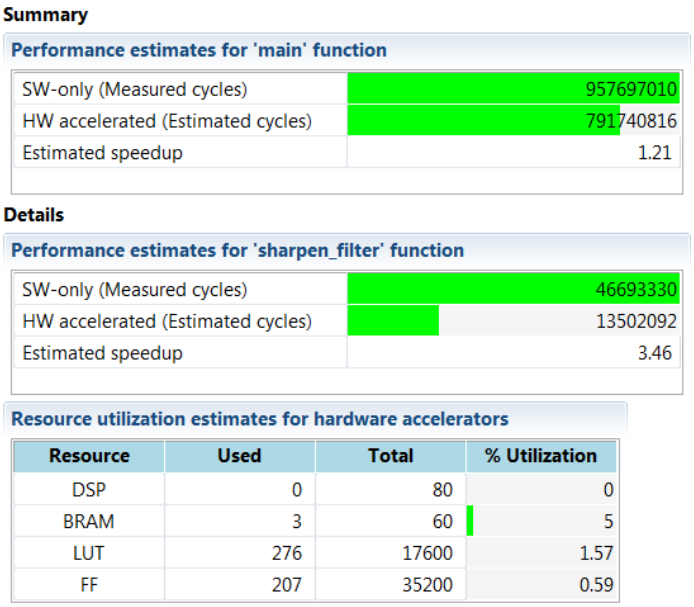
Performance estimates for 'sharpen_filter' function

SW-only (Measured cycles)	46685688
HW accelerated (Estimated cycles)	9743741
Estimated speedup	4.79

Resource utilization estimates for hardware accelerators

Resource	Used	Total	% Utilization
DSP	0	220	0
BRAM	3	140	2.14
LUT	306	53200	0.58
FF	297	106400	0.28

(a) Zed



Summary

Performance estimates for 'main' function

SW-only (Measured cycles)	957697010
HW accelerated (Estimated cycles)	791740816
Estimated speedup	1.21

Details

Performance estimates for 'sharpen_filter' function

SW-only (Measured cycles)	46693330
HW accelerated (Estimated cycles)	13502092
Estimated speedup	3.46

Resource utilization estimates for hardware accelerators

Resource	Used	Total	% Utilization
DSP	0	80	0
BRAM	3	60	5
LUT	276	17600	1.57
FF	207	35200	0.59

(b) Zybo

Figure 9. Performance estimation targeting Linux OS

Note that the number of SW-only cycles have increased compared to Figure 3. This is due to the overhead running in Linux.

3-4-3. Turn OFF the board and exit the SDSoC program.

Conclusion

In this lab, you performed speedup estimation of an application running under Standalone OS and Linux OS, after targeting desired function for acceleration. Performance estimation does not require the full bitstream generation and it gives speedup estimate by looking at the performance report generated by HLS for each function targeted in hardware.