TECHNICAL UNIVERSITY OF MADRID

HANDS-ON IRIO-GPU

June 6th 2016. Real Time Conference

"Advanced data acquisition and processing applications using NIRIO FPGAs-based technology and NVIDIA GPUs. Integration in EPICS"

Authors: Julian Nieto
Sergio Esquembri
Mariano Ruiz

Table of Contents

1 HANDS-ON IRIO-GPU

1.1 Objective

Analyse and modify an application performing the data acquisition and processing of images acquired by a cameralink framegraber (see Figure 1). The application is developed in C language using IRIO library. The camera has been replaced by a cameralink simulator generating images.

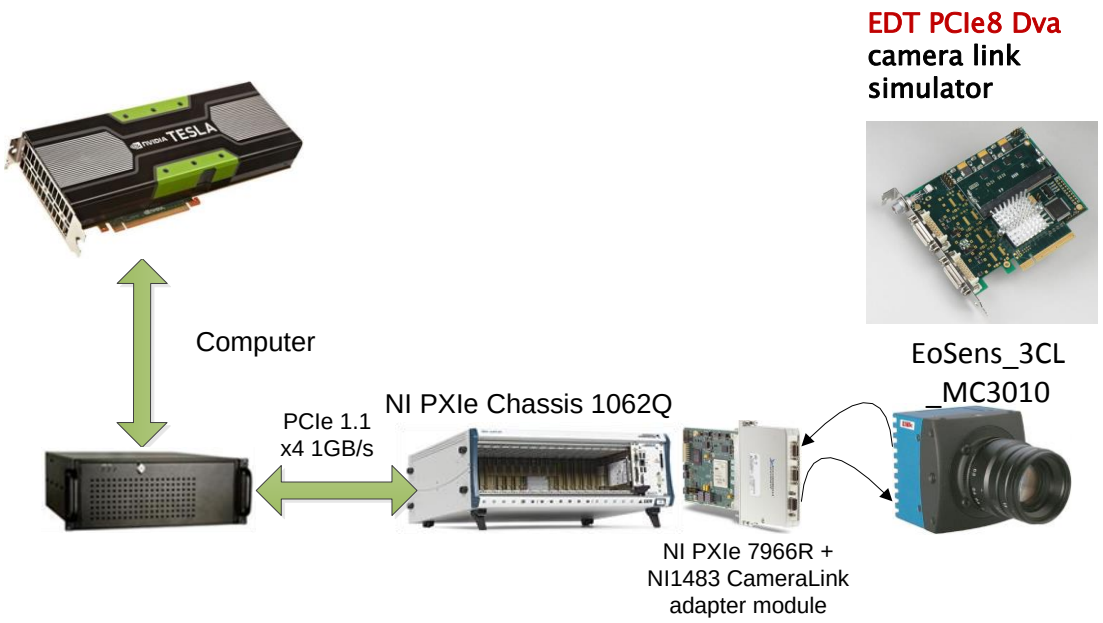


Figure 1: Architecture of the image and processing system implemented with PXIe Technology

The C application acquire images from the FPGA-based framegraber implemented in the FlexRIO device. The images are moved by DMA directly to a GPU and processed using an algorithm implemented with CUDA software tools. The images acquired can be displayed using a python script.

1.2 Material needed to execute this lab:

- Laptop with a ssh client in a Linux computer or applications like MobaXterm in a Windows computer (you can download it from indico conference WEB Server).
- Wifi connection to private network
  - SSID: RT2016-UPM key: rt2016upm

1.3 Steps

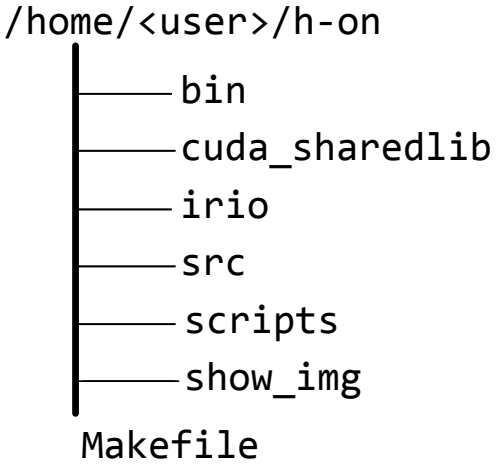
1. Connect to "short-course WIFI".
2. Connect to IRIO_GPU system using the following command line or use the MOBATERM application:

```
ssh -X rtsc<n>@192.168.1.104
```

n => 1..9

passwd: rtsc<n>.2016

3. Copy the compressed file /opt/h-on/h-on.tar.gz to your home folder. Decompress it.
4. Check that you have the following folder structure in your home folder:



5. Change to **/home/<user>/h-on/src** folder and open (using gedit or nano editor) the file "***FlexRIO_mod1483-7966-Imag-GPU.c***". This contains the source code of the application developed using IRIO software.
6. Review the content in this file and identify the different C sentences with the following flowchart.

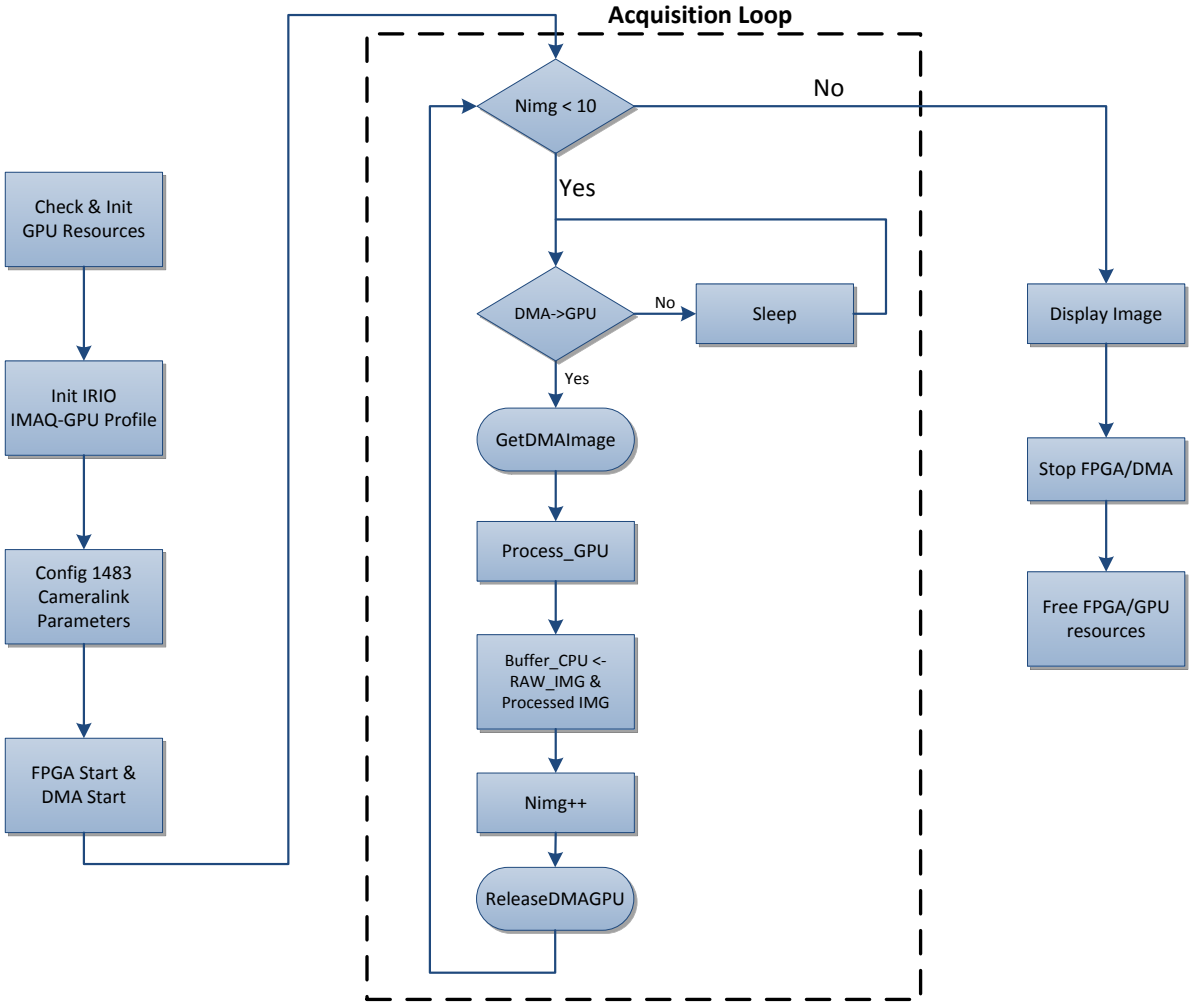


Figure 2: Flowchart of FlexRIO_mod1483-7966-Imag-GPU.c

7. Answer the following questions:
    a. What is the **GPU** buffer size in bytes?

    b. What is the **CPU** buffer size in bytes?

    c. What is the C variable used as image counter?

    d. How many images are processed?

    e. Which image is shown? (acquired, processed, both)

8. Identify the serial Number of the RIO

```
rtsc0@localhost~$ lsrio.py
```

You will see the different RIO devices installed in the computer. Ask the instructor which is the serial number for the RIO device with the NI1483 adapter module.

9. Compile the application executing ***make*** in the project root folder.

```
rtsc0@localhost~$ make
```

Warning. Please call to the instructor in this point to guarantee that you have exclusive access to hardware.

10. Before starting the application we need to run the cameralink simulator in order to send images. The commands to run the cameralink simulator are this:

```
rtsc0@localhost~$ cd /opt/EDTpdv
rtsc0@localhost~$ ./clsiminit -u 0 -C -l -f
camera_config/256_8Tap.cfg -v 1000 -g 1000
rtsc0@localhost~$$ ./simple_clsend -u 0 -m -l 0 -i imagelist.txt
```

Warning. If you are not the first user running the lab it is possible that the simulator is already running.

11. In the bin folder execute the application with this command:

```
rtsc0@localhost~$ ./FlexRIO_mod1483-7966-imag-GPU <S/N> 7966
<file>.raw
```

12. Run the program several times to see how you are acquiring and displaying the images generated by the simulator.

Warning. Please release the use of the hardware system to .other people in the short course

13. In this point you need to open the shared library with the GPU code implementing the processing algorithms. Have a look to the content and identify the functions implemented.
14. **Modify** the C source code file, in order to change the function performing the image processing.

- Once the modifications are saved, change to project root folder and execute ***make***.
- In this point your need exclusive access to the hardware again. Execute the program to see the differences with the previous application.