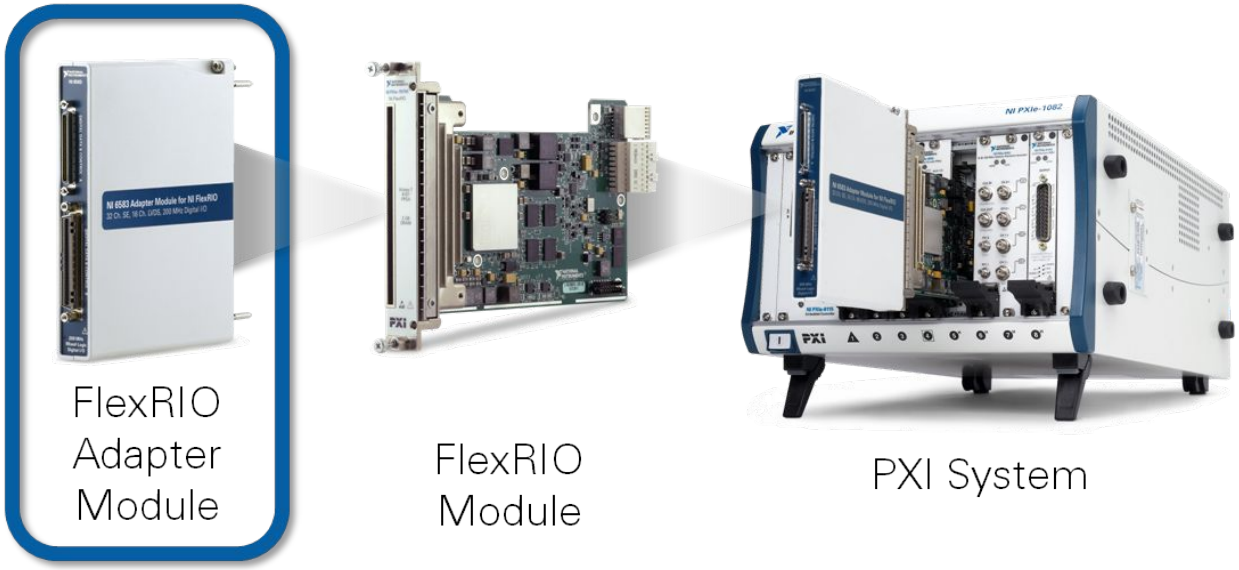


Valeo

SMART TECHNOLOGY
FOR SMARTER CARS

What's So Flexible about Flex RIO

Steven Dyra

Valeo

SMART TECHNOLOGY
FOR SMARTER CARS

Welcome

Agenda

SMART TECHNOLOGY
FOR SMARTER CARS

1. Valeo what we do
2. Test Scope
3. Test benches
4. Flex Rio Development
5. Adapter modules
6. The final application

VALEO Ireland what we do

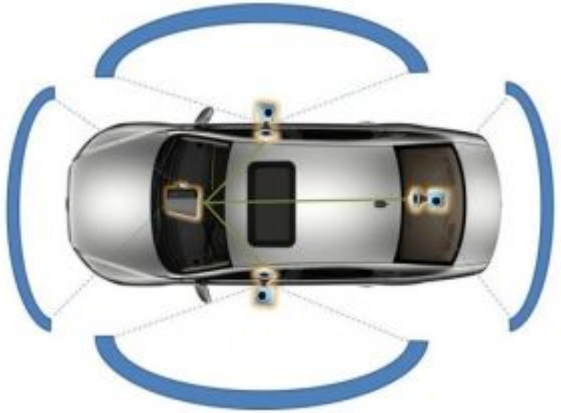
We are an automotive supplier that is based in tuam

We are a Certified Labview Center of Excellence

Valeo make cameras and design vision ECU's, VPM's (Vision Processing Modules)

The VPM is used for rendering images from the cameras and creating views to be displayed on the head unit for the driver.

The VPM also runs Algorithms like Pedestrian Detection (PD), 3D object detection (3DOD) and Lane Sensing (LS)



Overview of Experience

Working on Test solutions for the last 5 years

Worked on solutions for software integration test for 1 Year

- Java scripting for JTAG
- Software unit test testing, Testing embedded C++

Working on solutions for software test for 4 Years

- Controlling hardware with python (Power supplies, image analysis)
- Creating prototype tools with C#
- Creating applications in Labview

I’m also a CLD

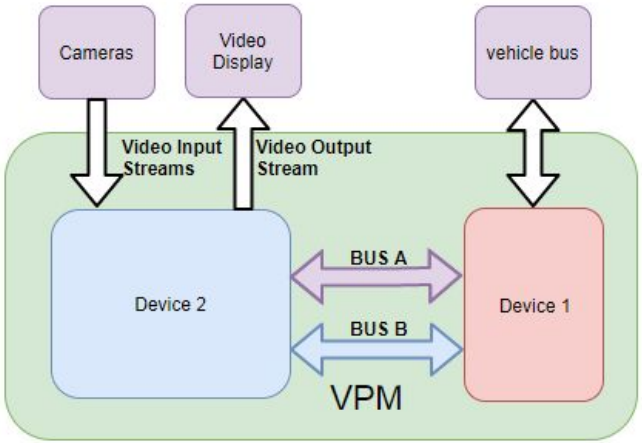
Test Scope Overview

The scope of the test team I am a part of is to test the DSP(Digital Signal Processor) SOC (System On Chip) on the VPM

Our Focus will be on the video pipeline and the error handling of the devices under test.

- Our objective is to prove the robustness of the devices in different error conditions for safety, as well as prove the framework which the algorithms use works as per the requirements

In the bottom right there is a small block diagram of the system that under test



Test bench: First Iteration

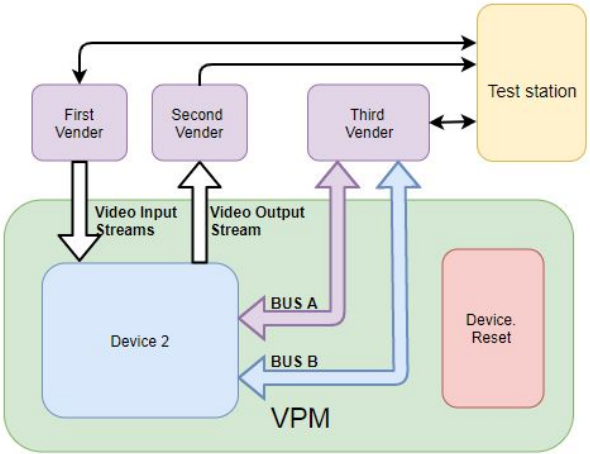
To be able to inject all the required faults the real devices could not be used

- We needed a solution that could handle the high throughput and all the different interfaces used

During the research phase we looked at lots of different tools from different vendors, this was the first Iteration

- One tool to simulate the camera streams
- Second tool to capture the video
- Third tool to simulate the embedded signals

When the requirements changed or the scope increased there was a cost to update the hardware or API's



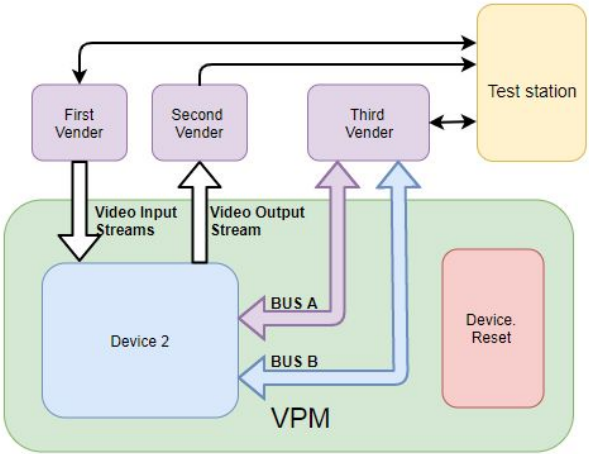
Test bench: First Iteration Pros & Cons

Advantages

- Hardware supplied by the vendor
- Software supplied by the vendor

Disadvantages

- Hard to have an integrated solution
- It was not feasible to synchronize these different tools to the level required for test
- Lots of different coding languages and API's used
- Vendors required different OS
- Changes to the software resulted in high costs
- Low level faults are not easy to create unless the requirements have been gathered early

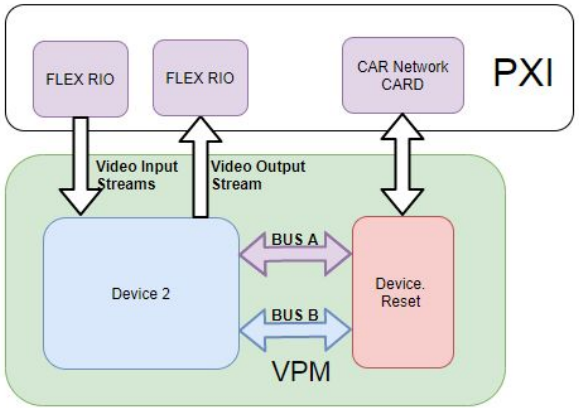


Why our team moved to Flex RIO

How our group decided to move to Labview and Flex RIO

- Chris Forristal gave a demo of a HIL (Hardware in the Loop) application he was working on for scene playback and capture for algorithm tests
- We observed the data throughput and its ability to capture all the data that was generated by the VPM
- The demo showed that the Flex RIO can have dedicated hardware on the front to convert the data to a format best suited to the FPGA

Once this was seen we changed our approach to work with the Flex RIO and NI PXI



Flex RIO solution

Advantages

- Development time reduced when libraries are used for code reuse
- Most of the Hardware supplied by NI; PXI, the FlexRIO card
- Software can be standardized with labview
- Can make custom adapters to match your hardware
- All Devices can be synchronized with the backplane trigger on the PXI
- Only need to use one OS (Windows)

Disadvantages

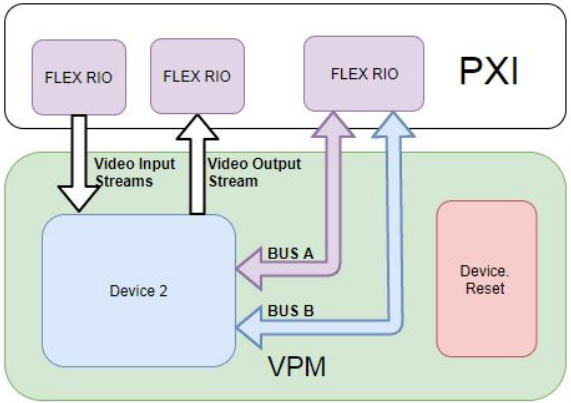
- The developer needs to know the protocols and Labview FPGA
- Change to custom adapters still has a cost
- Can be hard to debug the FPGA code

Test bench: The Flex RIO approach

The Flex RIO and the NI PXI will meet our requirements

- High Throughput with the option to perform high speed processing
- The approach covered simulation and capture of all required interfaces
- Quick implementation of new features
- Synchronized Clock timings with all tools in the same PXI
  - Using the PXI backplane triggers to synchronize clocks
  - On the FPGA using the 10 mhz clock on the PXI for the time stamping

We decided on the solution shown on the right



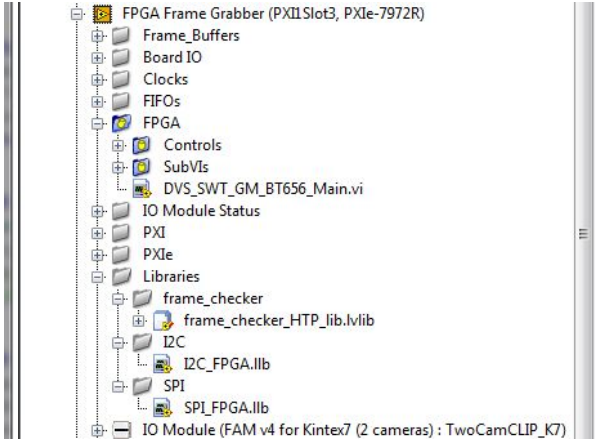
Flex Rio Code Reuse

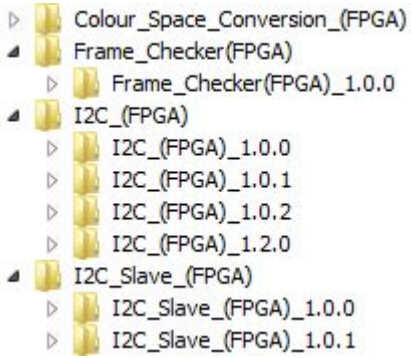
Why reinvent the wheel?

Seeing as some of the required drivers where already working and in libraries they were reused

We follow a Configuration management plan for our libraries, so they are all:

- Code Reviewed (with our code review checklist)
- Baselined
- Tested
- Then used in application





Flex Rio Code Testing

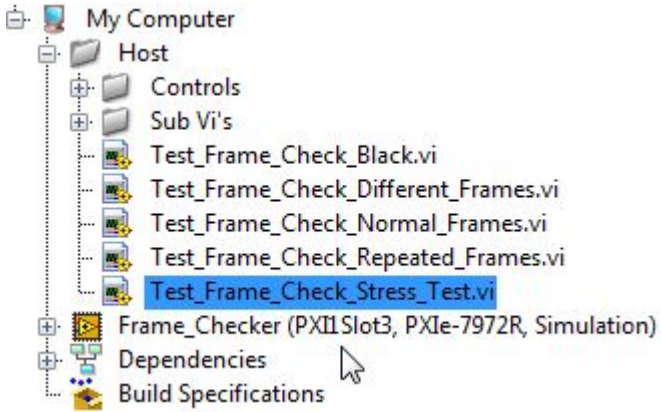
Testing your FPGA code can be tedious task

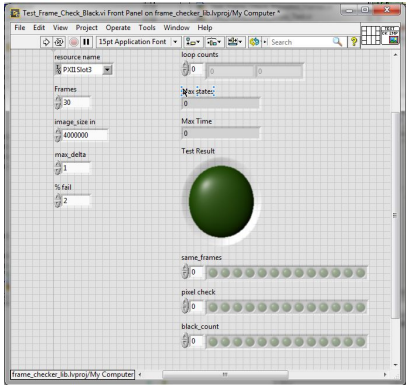
The best approach that we have used is making test applications for the libraries or the full FPGA application

- Note:The simulator will give false positives (there is never a time out)
- using the real compiled FPGA code is the best way to test
  - This also provides examples of how to use the Library for other developers

Example of test host applications for a Library on the right with a screenshot of a test VI

- Note: The big test result icon on the vi
- Each one of the test's shall be added to the release notes smoke test section

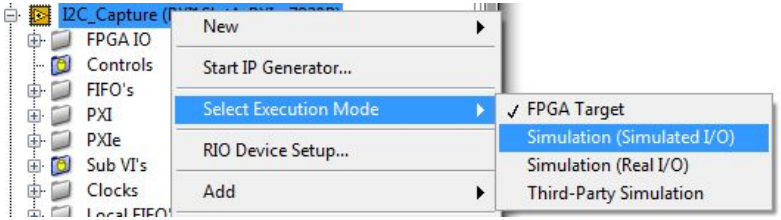




Flex Rio Quicker FPGA code development

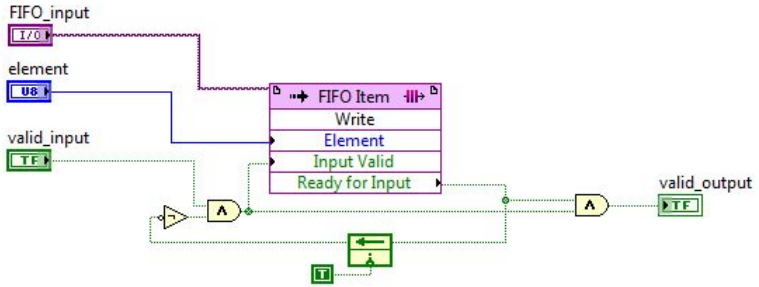
If any of you have worked with the NI FPGA you will have noticed it is quicker to build a prototype in normal labview

- This saves on developing time, as prototype with the real target FPGA involves compiling



Next step would be move to the simulator

- This helps validate the positive cases for the application
- Also helps with the building of the test application for the real FPGA code
- when using the simulator we will wrap the FIFO's in a way to create the negative cases



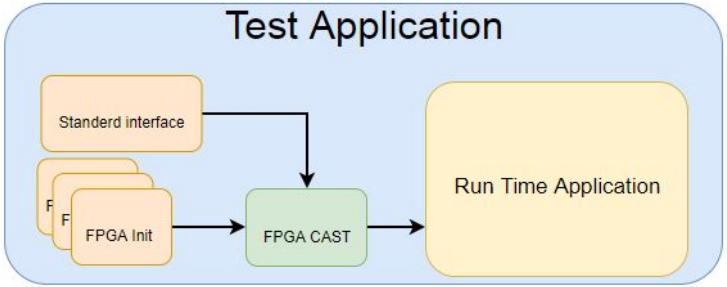
After these steps we would they on the real FPGA

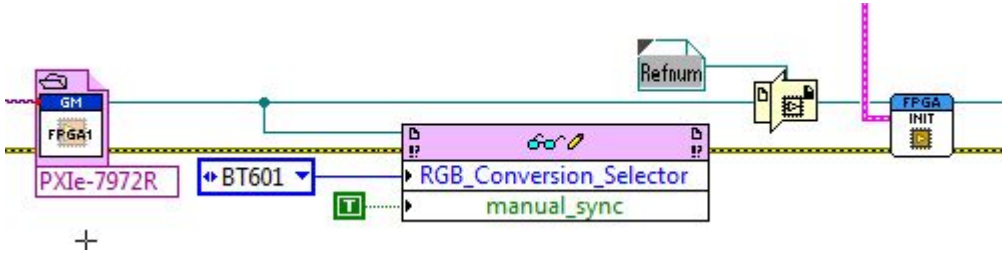
- Save a lot on time on completing FPGA bit files

Flex Rio on the Host

When there are multiple FPGA configurations or bit files, it is possible to cast the FPGA reference to have only the basic interfaces for your application

- When you can cast down to the run time interfaces the same host code can be used
- Note: to make your cast down FPGA reference, create a reference constant. Right click and select “Configure FPGA VI Reference” and remove any interfaces that are not needed in run time





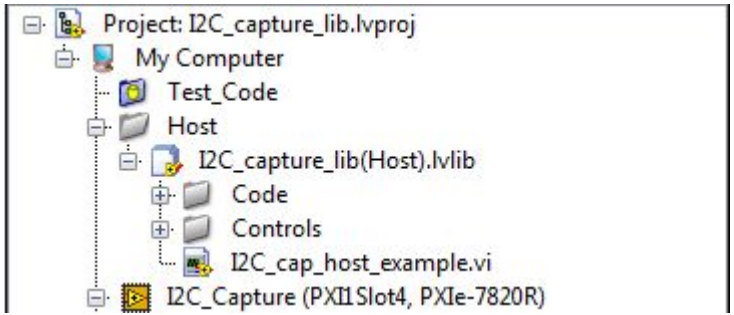
Flex Rio deploying bit files

We have moved to using the bit files as their own projects

- This allows use to develop the host applications in 64bit labview
- Labview FPGA is supported in 32 bit windows

With the bit files we add an interface library for the host application.

- saves on rebuilding interface code on the Host application

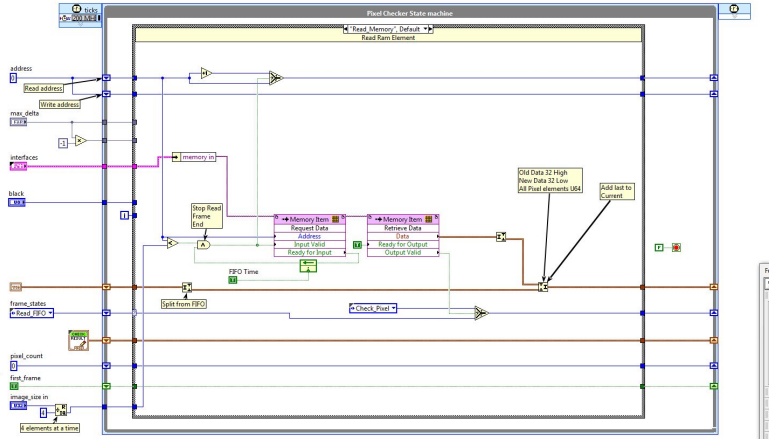


Flex RIO On board DDR

Last year I started using the DDR on the FPGA

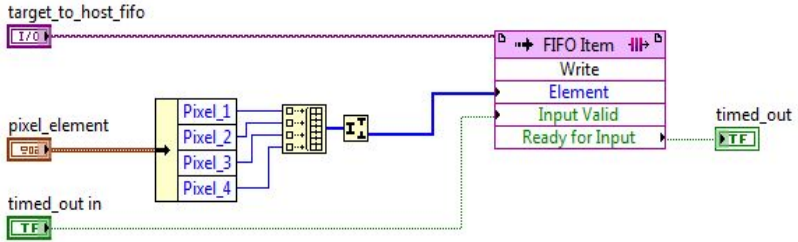
Have to say I found it more capable than expected

- Needed to be able to process 120 (MB/s) a second
- Ended up being able to process 960 (MB/s) a second and can scale to 1.92 (GB/s)
- just had to keep in mind every DDR element is 512 bits in size, so keep the cluster/array under the bit size



With the DDR we are not restricted to basic elements in the FIFO's on the FPGA only on the FPGA to Host?

It's possible to load multiple elements at once to the Host which helps reduce number of cycles in the single cycle timed loops



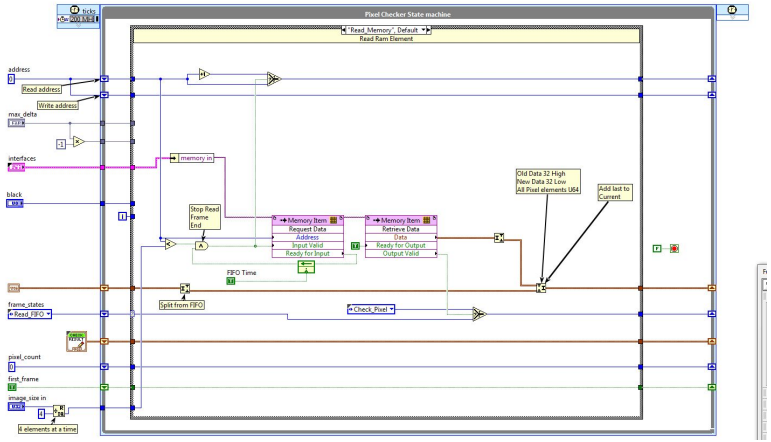
Flex RIO On board DDR

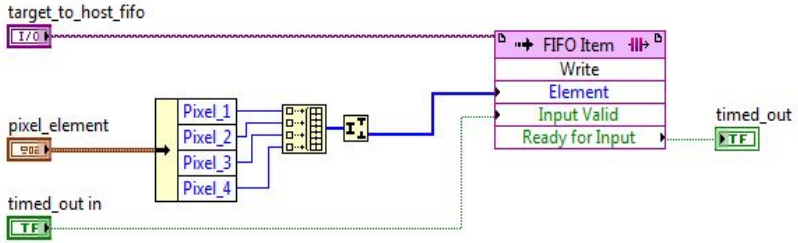
After we started using the onboard ram it freed up approx 25% of the CPU load on the benches

- Equivalent to two tabs in your chrome browser

It increased bench performance

- Reduced Cycle time of tests



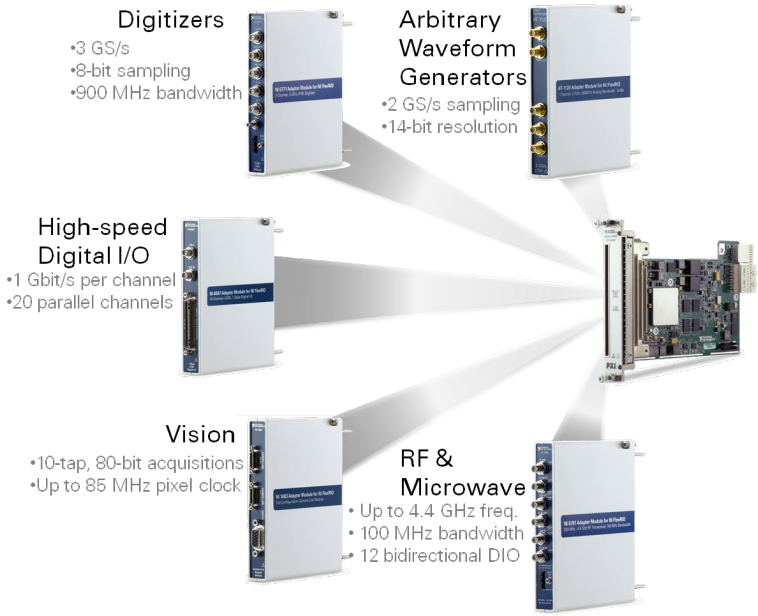


Flex Rio adapter modules

One of the most powerful features of the Flex RIO Cards

When dedicated hardware is needed to interface with a LVDS (Low Voltage Differential Signal) or CAN (Controller Area Network)

- The developer can get a custom front end for their Flex RIO depending on the hardware requirements
- If national instruments don't have the required card there are third party suppliers who will make custom adapter modules
- Generally this is used to create the physical layer that converts high speed or high voltage communications to parallel



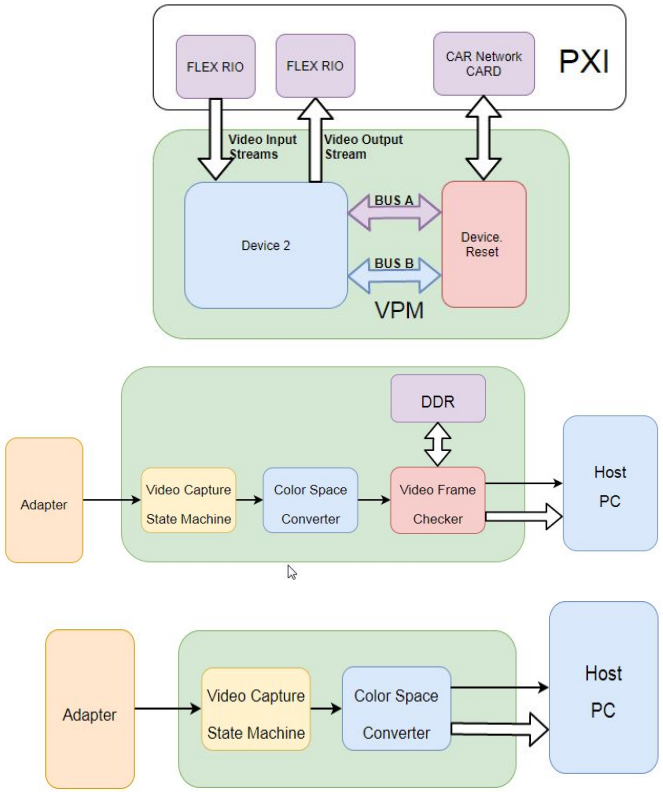
The applications we developed

There was a requirement for 4 simulated cameras

- These cameras didn't need to pass video for algorithm test, the only requirement was to pass frames
- We stored the frames in the DDR and had the host application logging every action and gathering stats
- The host also had the option to change data on the Flex RIO for fault injection

There was a requirement for 1 video capture port

- This was a requirement to capture and save images and display them
- There was also a requirement to check the images, So DDR RAM again stored the images in the pipeline for analysis.
  - This was a huge saving on time and processing power (approx 20%)



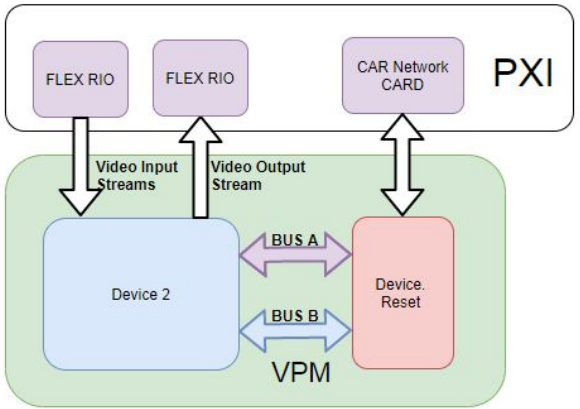
The applications we developed

There was a requirement to simulate a device on the ECU

- This required a DIO module like the NI 6851B or the NI 7820R (I know the 7820R is not a Flex RIO but its a beast for IO control under 80mhz, it complies faster to!)

This application has to emulate all the signals of the real device, and synchronise all the FPGA clocks

- The requirement for my tools was for the plug in to be the clock master
- There are also requirements for fault injection, on both the hardware and protocol levels



Questions & Answers

Valeo
SMART TECHNOLOGY
FOR SMARTER CARS

THANK YOU!


Valeo
SMART TECHNOLOGY
FOR SMARTER CARS