Management and analysis of physics datasets, Part. 1

Sixth Laboratory

Antonio Bergnoli bergnoli@pd.infn.it

8/1/2020

Laboratory Introduction

Goals

- Introduction to the IPBus framework.

IPBus (oriented for this course)

IPBus (1)

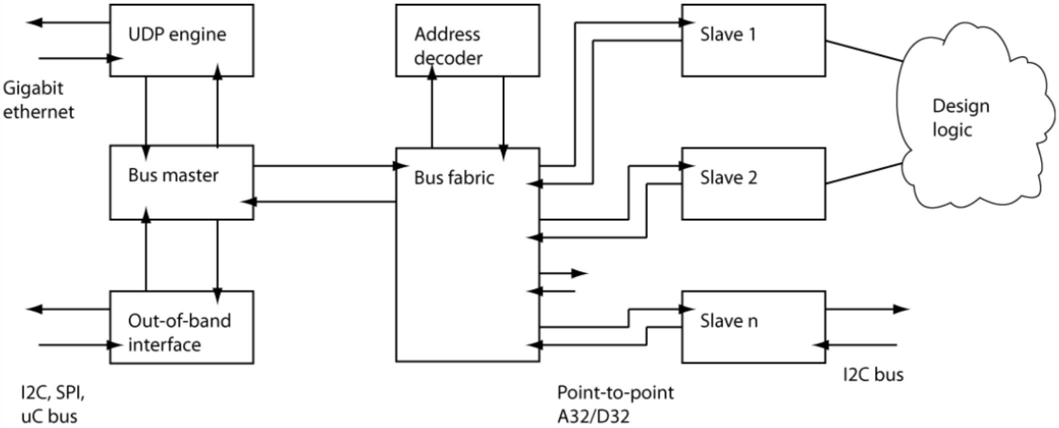
- The IPbus protocol is a packet-based control protocol for reading and modifying memory-mapped resources within FPGA.
- The memory-mapped resources are registers at 32 bit. They are addressed with addresses at 32 bit.
- The IPbus suite of software and firmware implement a high-performance control link for particle physics electronics, based on the IPbus protocol.
- It was developed for the control system in the CMS experiment. IPBus is replacing VME control in several large projects.
- It is also suitable for moderate speed DAQ systems

IPBus (2)

- The IPbus suite (for you) consists of two components:
  1. IPbus firmware. It is a module that implements the IPbus protocol within FPGA.
  2. uHAL. It is a library that provides a Python API for IPbus reads and writes transactions. In these laboratories ONLY the reads transactions are considered.
- Each IPbus host device (in these laboratories the evaluation board) has an IP address (10.10.10.100) and a port number through which it accepts IPbus control packets.

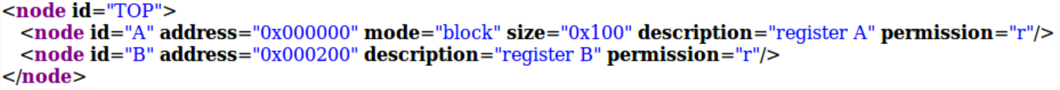
IPBus - Setup

IPBus - Firmware



IPBus - Software (1)

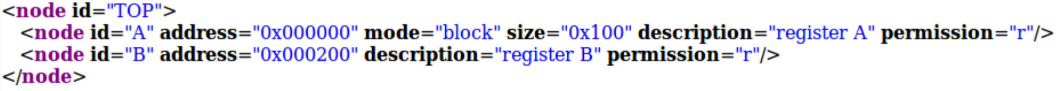
In uHAL the target's registers layout are specified by XML files, allowing a hierarchical address structure of the memory-mapped resources.



```
<node id="TOP">
  <node id="A" address="0x000000" mode="block" size="0x100" description="register A" permission="r"/>
  <node id="B" address="0x000200" description="register B" permission="r"/>
</node>
```

- **id**. It is the string identifier.
- **mode**. In this labs can be equals to single (single is the default mode if no mode is explicitly declared) or block. Single indicates that the node refers to a single word (i.e. 32 bits) register. Block indicates that the given address is the base address of a block of registers (number indicates from size tag) with a continuous address space (i.e. RAM).

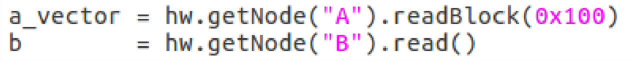
IPBus - Software (2)



```xml
<node id="TOP">
  <node id="A" address="0x000000" mode="block" size="0x100" description="register A" permission="r"/>
  <node id="B" address="0x000200" description="register B" permission="r"/>
</node>
```

- **permission**. r indicates a read-only IPbus endpoint (or slave).
- **description**. It is a way of documenting the address table. The content of the attribute can be accessed by an application.

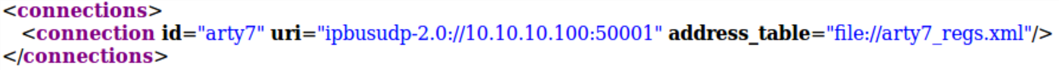
In order to read the memory block "A" and the memory location "B", you can use the following Python code:



```python
a_vector = hw.getNode("A").readBlock(0x100)
b        = hw.getNode("B").read()
```

IPBus - Connection

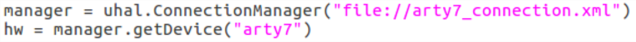
Another XML file is used also for configure the link connection.



```xml
<connections>
  <connection id="arty7" uri="ipbusudp-2.0://10.10.10.100:50001" address_table="file://arty7_regs.xml"/>
</connections>
```

- **id**. It is the string identifier.
- **uri**. It is the protocol (ipbusudp-2.0) and location (10.10.10.100) to access a target device in URI format.
- **address_table**. It is the location of the address table file which describes the register space of the FPGA.

The Python code to use this file is :



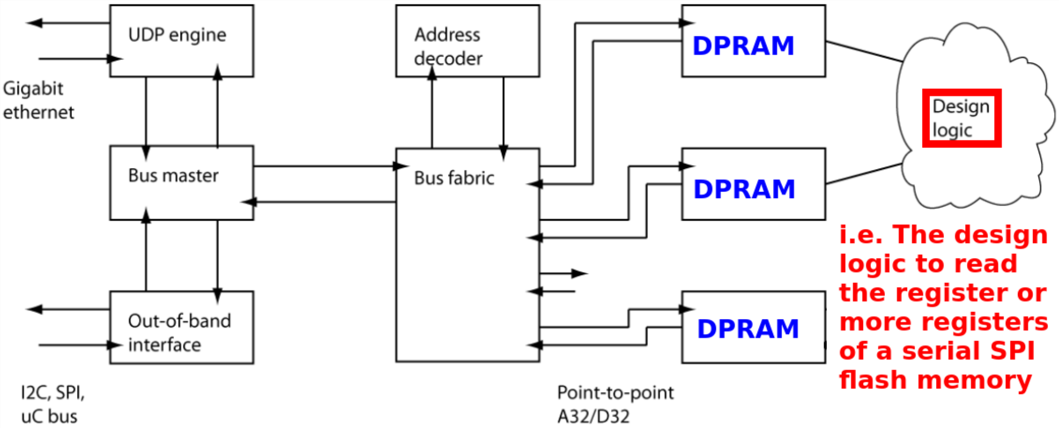
```python
manager = uhal.ConnectionManager("file://arty7_connection.xml")
hw = manager.getDevice("arty7")
```

IPBus for these labs

What to do?

1. You don’t have to write Python code (it is given);
2. You don’t have to write XML files (they are given);
3. You have to write only VHDL code. You have to write only the "design logic". The remaining code that implements the IPBus core, the interface from/to the PC and from/to the "design logic" is given.

Firmware



i.e. The design logic to read the register or more registers of a serial SPI flash memory

Hmw - IPBus Network Configuration

- The default IP address value was set to 10.10.10.100.
- You have to change the settings of your laptop network card in order to be in same subnetwork of the IPBus IP address; i.e. 10.10.10.1. Depending on your operating system, you can read these guides:
  1. Ubuntu
  2. Windows
  3. Mac
- If you want to change the IP address value of the evaluation board, you have to modify the field `uri` in the file *arty7_connection.xml* and the value assigned to the signal *s_ip_addr* in the file *top_level.vhd*.
- In order to check if the configuration is ok, text in the terminal "`ping 10.10.10.100`". The ping must be successful.

Hmw - IPBus OS Configuration

- IPBus works only in Linux operating systems (O.S.).
- Depending on which Linux distribution you have, you find the instruction to install the uHAL libraries at the following link:
  - uHAL
- In order to check if the installation of the libraries was successfully, open a terminal and text:
  - python
  - import uhal

  If no errors are given (i.e. `ImportError: No module named ...`), the installation is ok.
- Or if you do not have a machine with a Linux operating system you can use a virtual machine.