



CBPM development

Antoine, Nate

CBPM meeting
December 18, 2019

Recap' – test of the data throughput

Using an expanded version of the code I showed previously, I tested how fast we can read the FPGA version # from the module. The code flow is the following:

- x open socket
- x start timer
- x read the FPGA information a 1,000 times
- x stop timer
- x print read rate
- x close socket

The measured rate is about 300 Hz. This is only reading one register.

Next step: try reading out data from module issuing trigger and whatnot and measure the data throughput

Reading data from a CBPM module

Nate wrote MATLAB code that does minimalist data acquisition → translated that to Python3

After the usual struggles and making sure things were doing what they were supposed to → code seems to work: can trigger data acquisition of one module and read back the data

Example of read-out data (in hexadecimal):

```
b'\x08\x00' b'\x08\x00'  
b'\xdf\xff' b'\xdf\xff'  
b'\xde\xff' b'\xde\xff'  
b'\xbd\xff' b'\xbd\xff'
```

This data corresponds to 'pedestal' data and looks as expected

Data through-put

To measure the data throughput → loop over the entire code (except socket open/close) many times. Things about the code:

- x read one 16-bit data word per module (4 modules read sequentially)
- x process the data to put it in the proper format
- x nothing else is done to the data beyond that
- x all the registers configured every time → un-necessary over-head?

The typical data throughput is **6 Hz**

Removing what might be un-necessary register over-head for every data acquisition → data throughput is **30 Hz**

In principle, multi-threading would improve the throughput making tasks like writing to registers and reading from each module concurrent

Additional materials