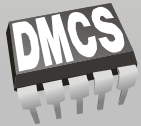


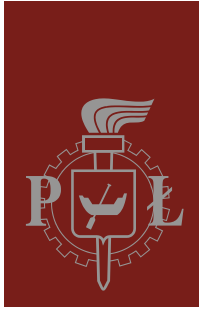
Implementation of nBLM algorithms

dr inż. Grzegorz Jabłoński,
dr inż. Wojciech Jalmużna,
dr inż. Rafał Kiełbik

12.02.2019



Department of Microelectronics and Computer Science, Lodz University of Technology
<http://www.dmcs.p.lodz.pl>



nBLM BEE hardware platform

IOxOS IFC_1410

FPGA Processing Unit

Xilinx Kintex UltraScale KU040 FPGA+1024MB dual channel DDR3L

Processor Unit

High-performance Freescale/NXP QorIQ T2081 processor

On-board 2GB DDR3L 1866 SDRAM

Powered by U-Boot/Linux and able to run EPICS-based applications

FMC Interfaces

Dual HPC VITA-57.1-compliant FMC slots

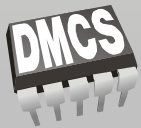
AMC Interface

Port 0: AMC.2-compliant gigabit Ethernet link with the processor

Ports 4 to 7: AMC.1-compliant PCI express x4 Gen3 link with the FPGA

Ports 12 to 15: point-to-point LVDS links with the FPGA

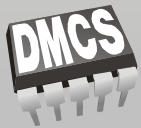
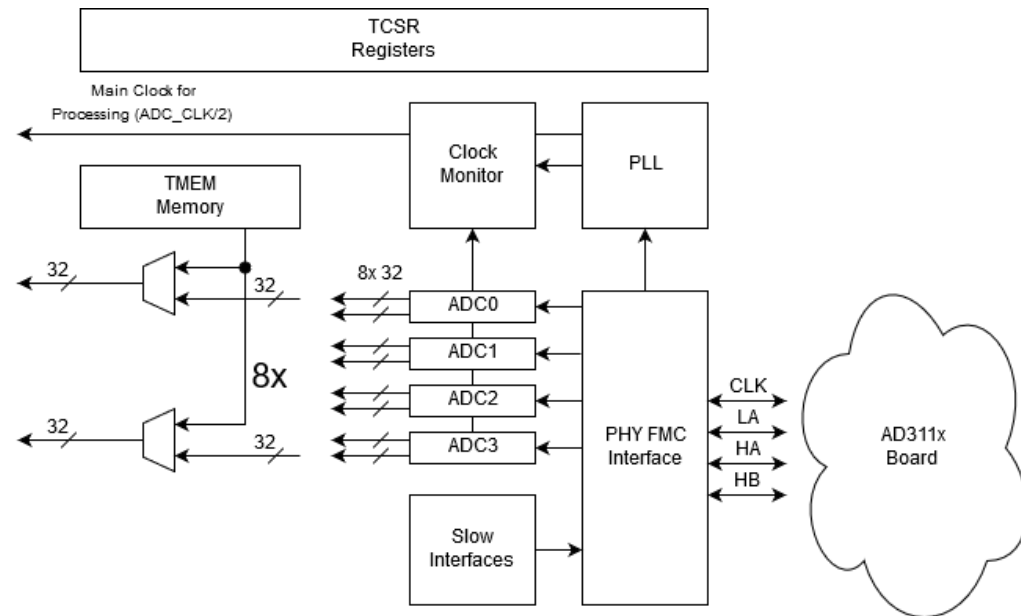
Ports 17 to 20: shared bus M-LVDS links with the FPGA





nBLM BEE hardware platform AD3111

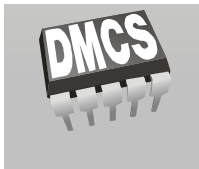
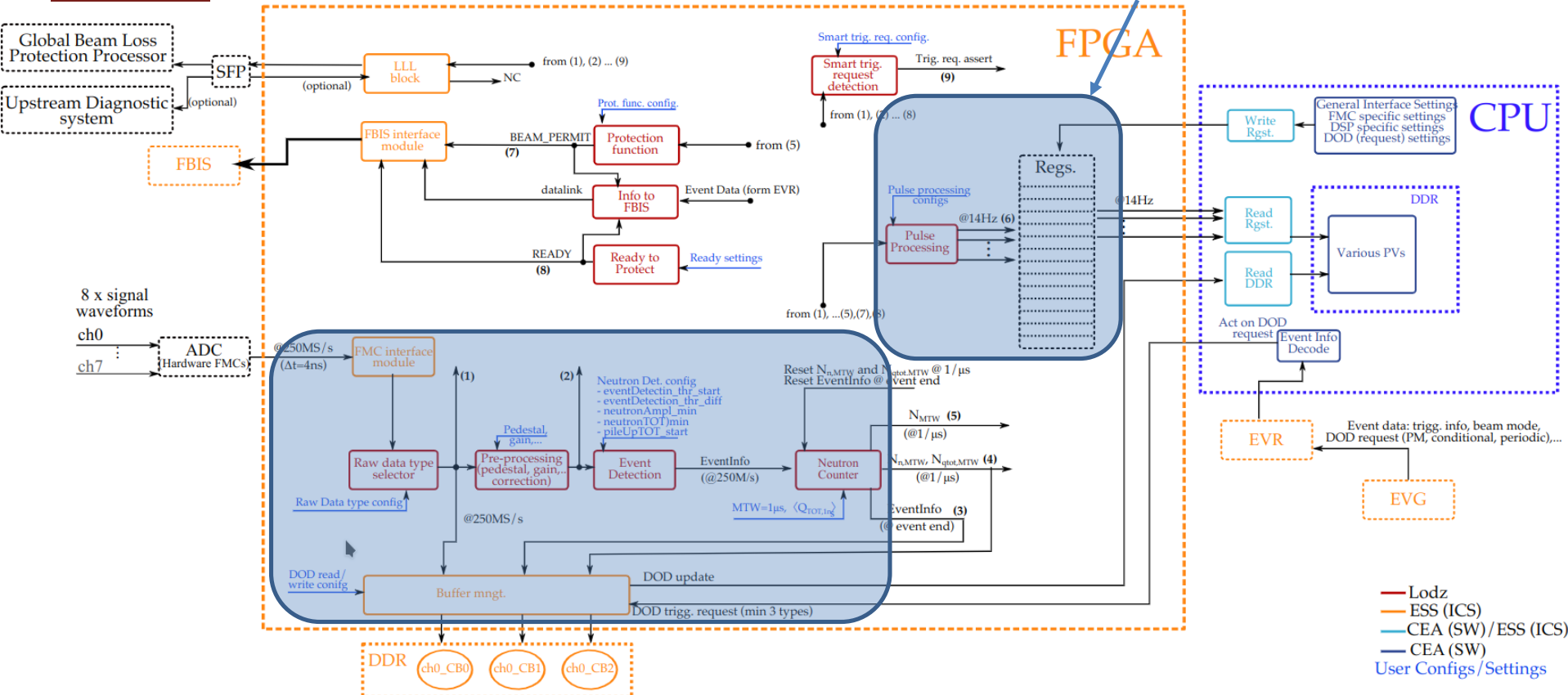
- Eight (8) channels 16-bit/250Msps ADC
- Sophisticated clock tree distribution
 - TI LMK4906 (dual PLL)
 - On-board ultra-low noise oscillator /VCXO
 - External SSMC Clock reference





Implementation status

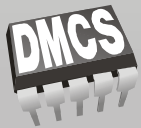
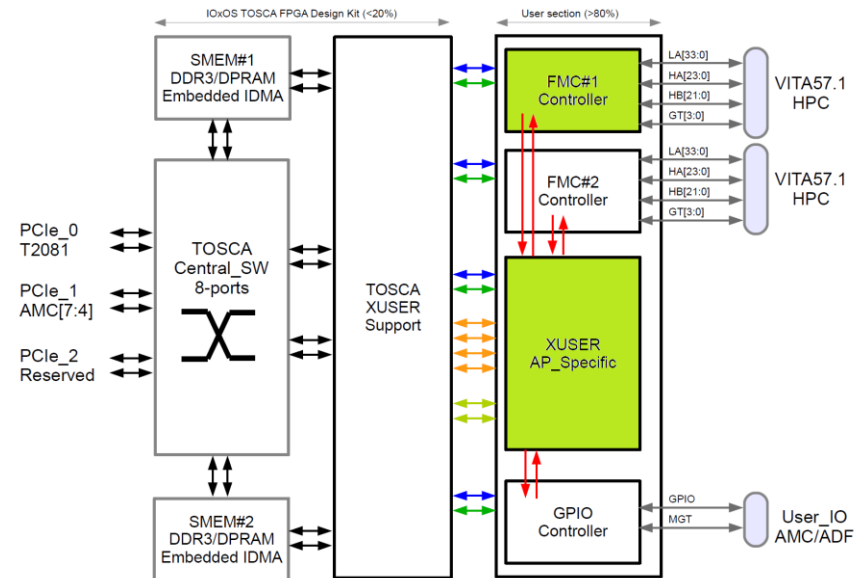
Periodic data is available via DDR





Hardware-software interface

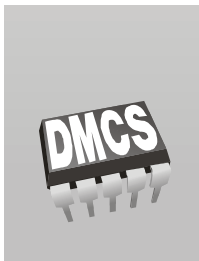
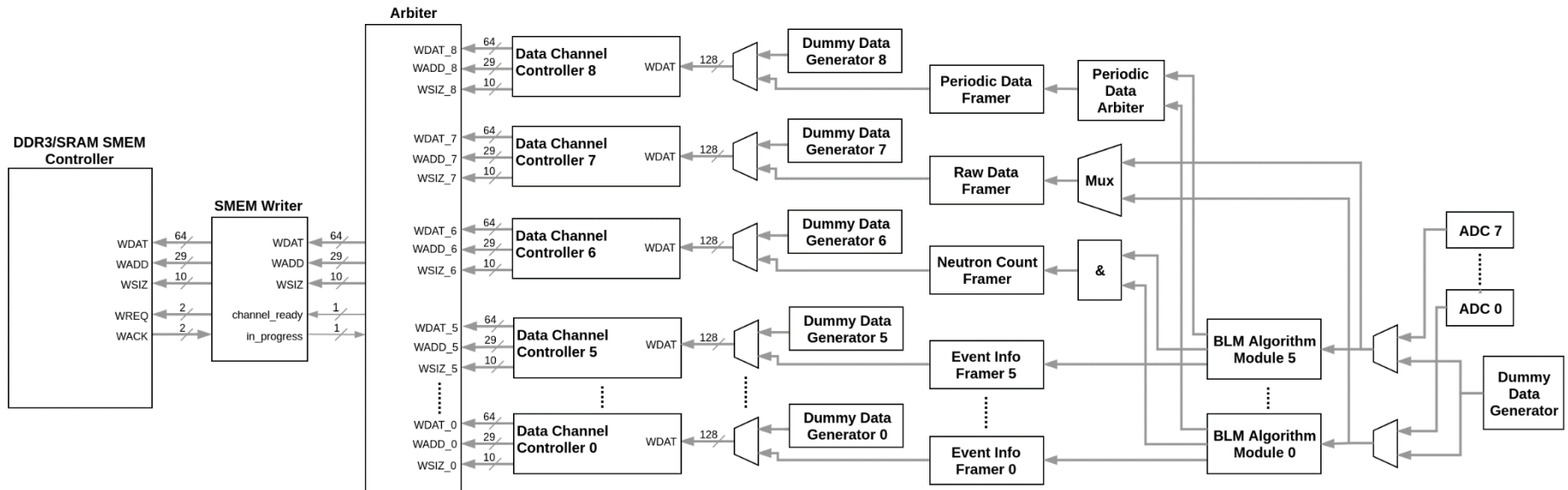
- ➔ Using circular buffers in DRAM to stream data to CPU
- ➔ Using TSCR interface for control and algorithm parameters
- ➔ Both DDR bank (currently) via SMEMDIR interface @275MHz, 1900 MB/s bandwidth
- ➔ Not needing such large Block RAMs overlapping initial part of DDR





Data transmission protocol layers

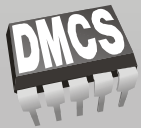
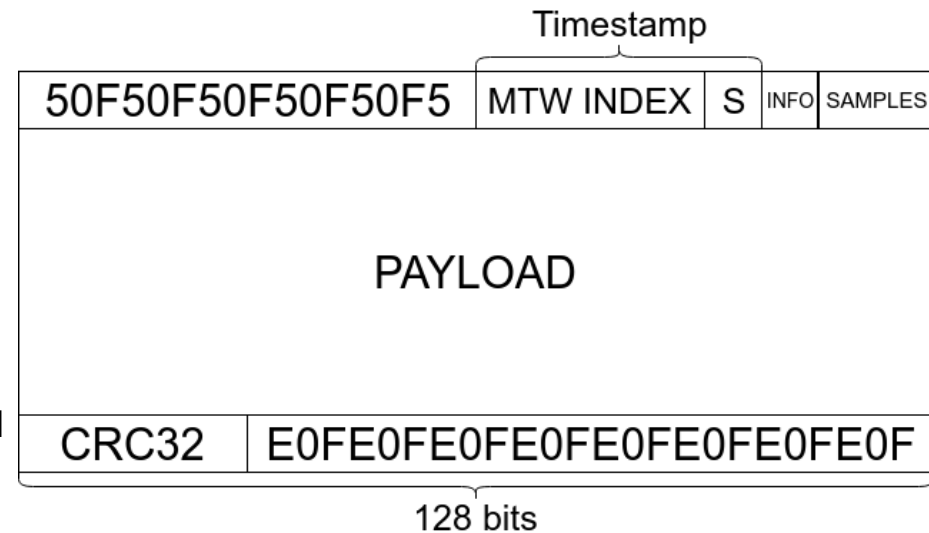
- ➔ Circular buffers – stream of unstructured data
- ➔ Data frames – timestamping and integrity check
- ➔ Periodic data – content-specific header





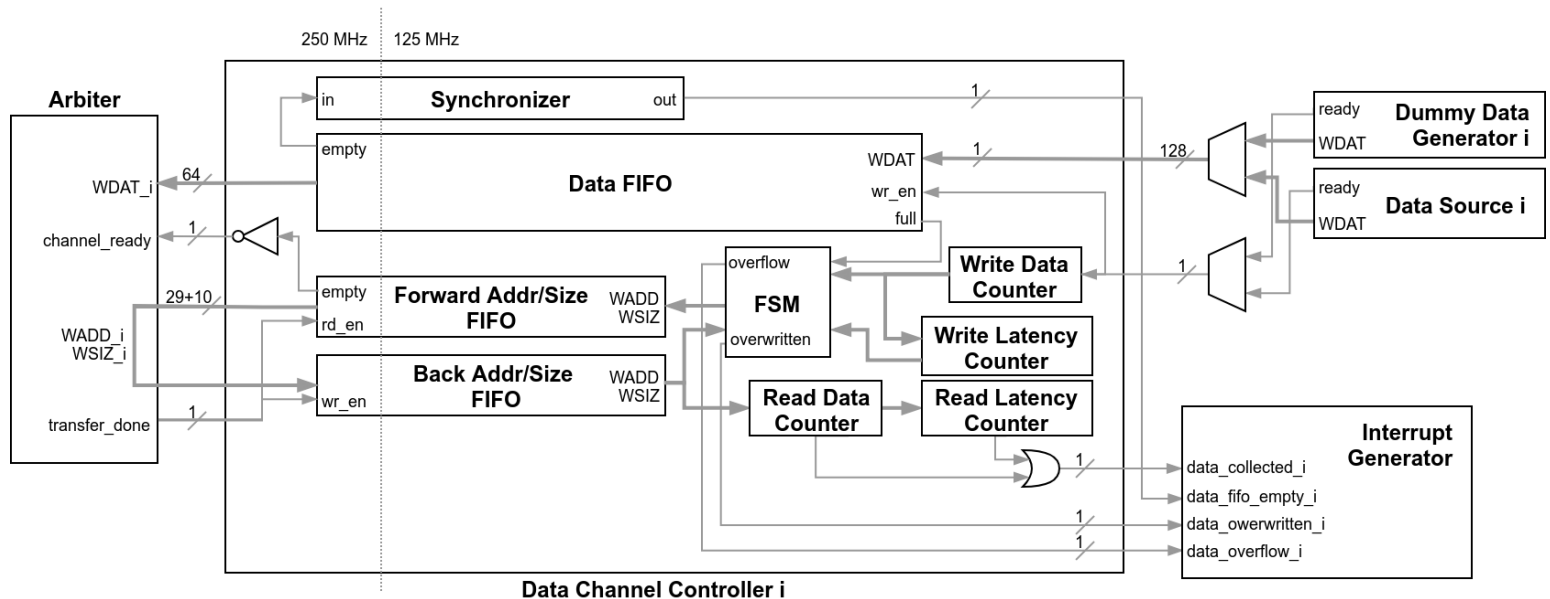
Frame layout

- Start-of-frame pattern
- Timestamp
 - Serial number of the 1-microsecond window
 - Sample index within the window.
- 1 generic information byte
- 16-bit number of samples in the frame
- Payload packed back-to-back on the bit level without any additional padding
- 32-bit CRC of all the previous words in the frame followed by the End-of-frame pattern.

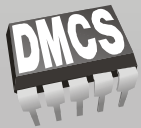




Circular Buffer Implementation



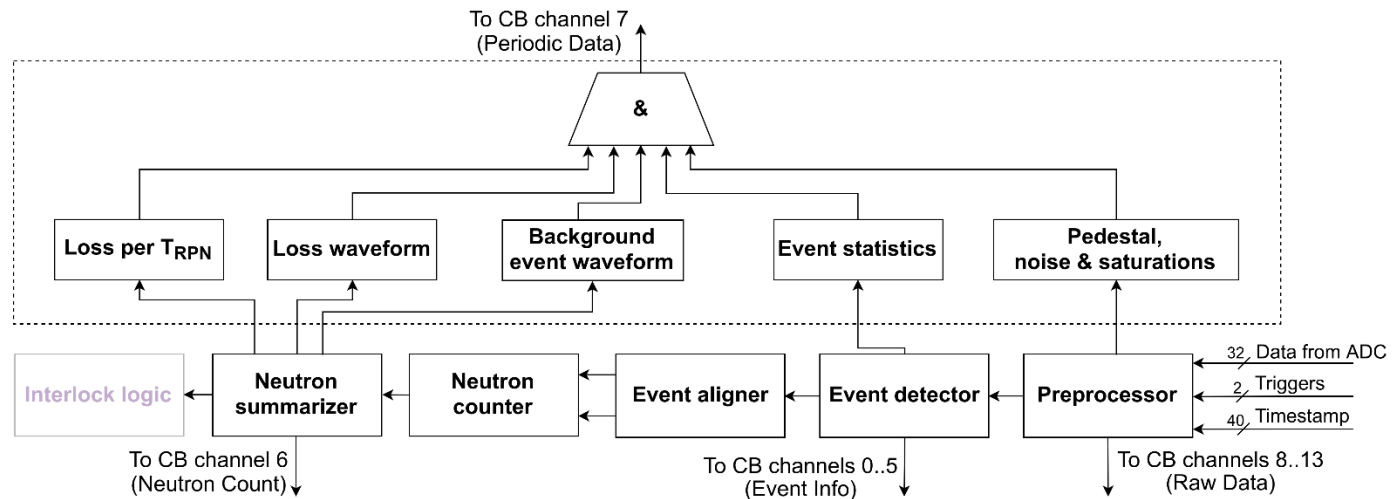
- ➔ Read pointer, write pointer for each channel
- ➔ Overflow – not able to write data to DDR at given input data rate
- ➔ Overwrite – data readout by DMA too slow
 - Read pointer at the moment of overwrite recorded





nBLM algorithms block diagram

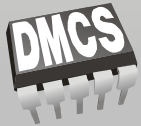
- ➔ Main flow – 5 pipelined processing blocks
- ➔ Implemented in C++ with High Level Synthesis
- ➔ Data in raw data, neutron summary and raw events channels timestamped by MTW number and sample number within MTW (unique within more than 1 hour, cycle-accurate)
- ➔ Periodic data – timestamped in the same way, but not cycle-accurate





Event detection algorithm

```
void
detect (hls::stream<preprocessedData>& A, hls::stream<eventInfo>& E, hls::stream<eventInfo>& E2,
hls::stream<pedestalComputationData>& PC, hls::stream<eventInfoForArchiving>& event_stream,
uint16_t neutronTOT_min_indx, uint16_t pileUpTOT_start_indx)
{
#pragma HLS LATENCY min=1 max=1
#pragma HLS PIPELINE II=1
#pragma HLS INTERFACE axis off port=A
#pragma HLS DATA_PACK variable=A
// ...
for (int i = 0; i < 2; ++i)
//...
    if (data.belowThr1 || (data.belowThr2 && ended_by_frame))
    {
        //start an event
        if (!bEve)
        {
            MTWindx = data.frame_index;
            bEve = true;
            TOTstartTime = data.sample_index;
            peakValue = data.adjusted_sample;
            peakTime = 0;
            peakValid = false;
            TOTvalid = false;
            pileUp = false;
            TOTlimitReached = false;
            event_isPart2 = ended_by_frame;
            TOT = -1;
            Q_TOT = 0;
            peakCounter = 0;
```





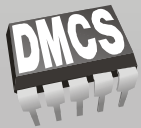
Solved problems

- ***DDR Calibration***

The TOSCA framework has had a bug, causing segmentation violation errors on the host CPU during the on-board DDR memory calibration procedure most of the time. Several reboots of the IFC_1410 modules were needed to make the board operate properly. The problems became more severe after migration of the Vivado version from 2016.4 to 2018.2. The problem has been finally identified by IOxOS and fixed in December 2018

- ***Timing closure problems***

The examples provided by IOxOS do not synthesize without timing errors “out of a box”. Most implementation runs of our project resulted in pulse width timing violations related to the 250 MHz clock driving the DDR memory controller. Finally we have been able to fix the problem by constraining the global buffer output driving this clock to a specific location. With this constraint, the memory clock can be increased to 275 MHz.

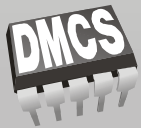




Solved problems

- ***DMA performance problems***

Initially we were instructed to use the Tosca kernel driver for the communication between CPU and FPGA. This driver provided very low DMA throughput of order of 550 MiB/s. Later on we have transitioned to the ESS-Tsc driver with the throughput of 950 MiB/s.

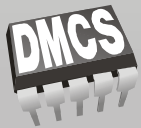




Current problems

- ***Loss of data during simultaneous DMA transfers from both DDR banks***

There is a bug in the TOSCA firmware or device driver causing data loss when the DMA transfer is performed simultaneously from two different memory banks using different DMA channels from different threads. The workaround is protections of the DMA transfer function in the userspace program by mutex, but this solution reduces performance.

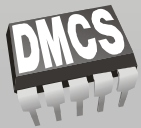




Current problems

- ***Tsc driver not supporting scatter-gather operations and prone to resource leaks***

Tsc driver has high performance, but requires allocation of contiguous physical memory buffers for operation. When memory is fragmented, it is not possible to allocate these memory buffers and a CPU reboot is required. Tosca driver did not have this problem, but it offered much lower performance. These memory buffers have to be manually allocated and are not freed automatically when the device file is closed. It requires appropriate signal handlers in the userspace program to avoid resource leaks.





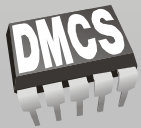
Current problems

- ***Problems with PCIe Gen3 link to Concurrent CPU***

The PCIe link speed between Concurrent CPU and the IOxOS board alternates between Gen1 and Gen3 after each system reboot.

- ***Readback from algorithm parameters register blocks***

Readback from algorithm parameters block does not work. It is a firmware issue and will be fixed.





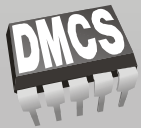
Current problems

- ***No separate interrupt support for both memory banks***

Only one interrupt, common for both memory banks, is supported. It is a firmware issue and will be fixed.

- ***Tsc Driver cannot access PON space when run on Concurrent CPU***

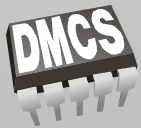
TscMon (and custom software applications) fail to access registers placed on PON configuration space (such as Power On for FMC cards) when run on Concurrent CPU. The same software and TscMon commands can be successfully run on embedded IFC1410 processor. The bug prevents proper behaviour of the software when only one PCIe endpoint (to Concurrent) is present in the design. Reportedly can be fixed by the PON FPGA firmware update.

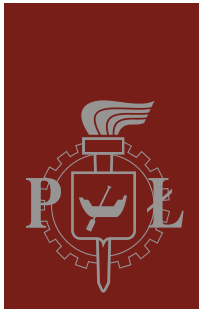




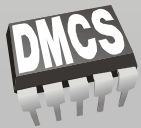
Current status

- ⇒ Fully implemented main data processing chain for 6 ADC channels (selectable from 8 inputs) with 8 data streams
 - ⇒ Neutron, saturations and background event count every microsecond from all channels
 - ⇒ Raw data from any of the 6 channels
 - ⇒ Raw events from 6 channels
- ⇒ Periodic data partially implemented
- ⇒ Tsc driver supported on both Concurrent and POWER CPU
 - ⇒ Possible to record 2 s of raw data from 1 channel with Tsc driver
 - ⇒ Able to acquire raw data from two channels simultaneously if DMA bug is fixed
 - ⇒ Able to acquire 9 ms windows of raw data from every 14 Hz period together with all auxiliary data channels (even without DMA bug fix)





Thank you for your attention



Department of Microelectronics and Computer Science, Lodz University of Technology
<http://www.dmcs.p.lodz.pl>