Towards Optimal Filtering on ARM for ATLAS Tile Calorimeter Front-End Processing

Mitchell A. Cox
School of Physics, University of the Witwatersrand, Johannesburg 2050, South Africa
E-mail: mitchell.cox@students.wits.ac.za

Abstract. The Large Hadron Collider at CERN generates enormous amounts of raw data which presents a serious computing challenge. After planned upgrades in 2022, the data output from the ATLAS Tile Calorimeter will increase by 200 times to over 40 Tb/s. Advanced and characteristically expensive Digital Signal Processors (DSPs) and Field Programmable Gate Arrays (FPGAs) are currently used to process this quantity of data. It is proposed that a cost-effective, high data throughput Processing Unit (PU) can be developed by using several ARM System on Chips in a cluster configuration to allow aggregated processing performance and data throughput while maintaining minimal software design difficulty for the end-user. ARM is a cost effective and energy efficient alternative CPU architecture to the long established x86 architecture. This PU could be used for a variety of high-level algorithms on the high data throughput raw data. An Optimal Filtering algorithm has been implemented in C++ and several ARM platforms have been tested. Optimal Filtering is currently used in the ATLAS Tile Calorimeter front-end for basic energy reconstruction and is currently implemented on DSPs.

1. Introduction
Projects such as the Large Hadron Collider (LHC) generate enormous amounts of raw data which presents a serious computing challenge. After planned Phase-II upgrades in 2022, the raw data output from the ATLAS Hadronic Tile Calorimeter (TileCal) will increase by 200 times to over 40 Tb/s (Terabits/s) [1, 2]. It is infeasible to store this data for offline computation.

A paradigm shift is necessary to deal with these future workloads and the cost, energy efficiency, processing performance and I/O throughput of the computing system to achieve this task are vitally important to the success of future big science projects [3].

ARM System on Chips (SoCs) are found in almost all mobile devices due to their low energy consumption, high performance and low cost [4]. The author is developing an ARM-based PU for the ATLAS TileCal as a high throughput, general purpose co-processor to the read-out system Super Read Out Driver (sROD) which can be used to combat the issue of out of time pile-up. Currently Optimal Filtering is used for signal processing of the raw data, but more sophisticated algorithms may be required in future [5, 7]. A general purpose co-processor is able to run more sophisticated and memory intensive algorithms than an FPGA-based device, although the latency cannot be controlled like it can with an FPGA, which is why the sROD is used in the data path [6].

A brief background on the ATLAS TileCal read out system is provided in Section 2. The results of a C++ Optimal Filtering algorithm on ARM are presented in Section 3 and the
implementation of a simple PCI-Express CPU interconnect and preliminary performance is
given in Section 4. Section 5 concludes with a brief discussion of future work.

2. The ATLAS Tile Calorimeter

The Hadronic Tile Calorimeter (TileCal) is made up by layers of scintillator and steel absorber. When a particle interacts with the scintillator, a pulse of light is produced which is converted to an electrical signal by PMTs. Bunch crossings in the ATLAS detector are separated by 50 ns, with 25 ns planned in future. In reality there can be over 20 separate collisions in a single bunch crossing which leads to a large number of particles interacting with TileCal which is called Pile-Up.

This electrical signal produced by each PMT is conditioned and spread into a pulse with a total length of 150 ns and full width at half maximum of 50 ns. An Analog to Digital Converter (ADC) samples this pulse every 25 ns resulting in seven samples per pulse. A reference pulse showing example sampling points is visible in Fig. 2 with three main parameters of the pulse also illustrated.

The TileCal read out architecture is required to digitize the analog signals produced by the PMTs located on the Tile Calorimeter. In the upgraded system, shown in Fig. 1, all of the PMT data is digitized and sent to the back-end where buffers and processing is done using the sROD [2].

The sROD is located in the back-end, off the detector to avoid the requirement for expensive, radiation-hard electronics. The sROD will be located in an industry standard AdvancedTCA (ATCA) chassis which enables comprehensive redundancy and monitoring to ensure maximum uptime.

In both the existing and the upgraded systems, a pipeline buffer is used to store events until the level one trigger provides an accept signal. This short (typically 5 μs) delay is required while the level one trigger performs computations. In the upgraded system the sROD is able to perform some calculations before sending data to the rest of the triggering and data acquisition system.

Optimal Filtering (OF) or a Matched Filter (MF) are two methods by which the amplitude, A, phase, τ, and base-line pedestal, p, parameters can be calculated from the seven ADC samples of a pulse [7]. The pulse shape and therefore the energy can be reconstructed, when required, from these three parameters [5]. Figure 2 shows an ideal pulse shape with parameters illustrated.

For both the OF and MF algorithms, each parameter (A, τ or p) can be found by multiplying the ADC samples by a specific set of weights which are calculated ahead of time. Both algorithms work well in low luminosity operation where the background noise of the PMT signals is gaussian and uncorrelated. This assumption fails for high luminosity operation (above about $\sqrt{s} = 8$ TeV) where the background noise is no longer uncorrelated due to pile-up [7].
Figure 2: TileCal reference pulse from the PMT showing 7 sample points.

Figure 3: Single thread performance of an Optimal Filtering implementation with seven samples and two outputs with varying block size. An Intel i7-4770 CPU and three ARM platforms are shown.

3. Optimal Filtering on ARM

Optimal Filtering is a simple digital signal processing algorithm where several samples from an analog to digital converter are multiplied with a pre-calculated weight vector. The result of this dot product depends on the set of weights and is tuned to a specific characteristic of the input signal. In the case of the TileCal OF, the peak amplitude and the phase shift ($A$ and $\tau$) of the signal are calculated using two sets of weights.

This algorithm was implemented using the EIGEN C++ library which is one of the fastest linear algebra libraries that also supports ARM [8]. Vector sizes of seven were used to correspond to the current TileCal read-out and two dot products were performed with random weights to represent the amplitude and phase filters.

Figure 3 shows the performance which has been converted to MegaBytes per second (MiB/s) of filtered data where one megabyte is equal to $1024 \times 1024$ bytes. A high-end Intel i7-4770 CPU has been compared to an ARM Cortex-A9 (Freescale i.MX6), an ARM Cortex-A15 (NVIDIA Tegra-K1) and an ARMv8 (APM X-Gene 1) platform. The ARMv8 is the only currently
available 64 bit ARM System on Chip and is similar to a Cortex-A57.

Batches of seven samples were combined into blocks for more efficient processing. Block sizes from 28 B (one set of samples) to 128 kB were tested.

It is clear that the Intel performance is better than the ARM platforms. It is interesting to note that the performance of the Tegra-K1 is better than the X-Gene. This should not be the case but it was found that the compiler is not yet mature for the new ARMv8 architecture. Using GCC 4.9 instead of GCC 4.8 makes approximately 10% performance difference on the X-Gene but for uniformity GCC 4.8.2 was used on all platforms.

The Wandboard is capable of a peak of 350 MiB/s and the other platforms all achieve over 1 GiB/s of Optimal Filtering throughput. This is typically more than the external I/O available on the SoCs. It is important to note that the results are for a single thread and so if multiple cores are used then the results should scale accordingly.

4. PCI-Express CPU - CPU Interconnect

The external I/O interface of a computing system should be well balanced with the available processing power. Because this is highly dependent on the algorithms used, benchmarking is important. Based on the results in Section 3, even the lowest performing system, the Wandboard, requires up to 350 MiB/s external I/O for good system balance. Gigabit Ethernet, which is almost always available on a SoC, only allows up to 125 MB/s (119 MiB/s). This is clearly insufficient for optimal system balance in this application.

PCI-Express is a high-bandwidth external I/O interface that is energy efficient and simple to implement for the system designer as it only requires several PCB traces and no special and potentially expensive hardware. There is still a requirement to eventually use a standard interconnect such as Ethernet if connecting to an existing system, but this could be accomplished using a single fast adapter node such an a more expensive SoC or some other dedicated system. For the connection to the sROD, PCIe can be used directly.

PCI-Express throughput tests have been performed on a pair of i.MX6 quad-core ARM Cortex-A9 SoCs clocked at 1 GHz, located on Wandboard development boards. The results are presented in Table 1. Three tests were run to ascertain the maximum data throughput that can be obtained from the i.MX6 SoC: a simple CPU based memcpy command and two Image Processing Unit based Direct Memory Access (DMA) transfers, initiated by the Endpoint (EP) or slave and the Root Complex (RC) which is the host. The i.MX6 has no dedicated DMA controller on it’s PCI-Express controller.

The theoretical maximum throughput for the PCI-Express Gen 2 x1 link used is 500 MB/s. The best result is using DMA initiated by the RC but it is only 72% of the theoretical maximum. The RC-mode drivers are more optimized than the EP-mode drivers due to limited manufacturer support for EP-mode. The read results are lower than write because of overheads to initiate the read.

It is not trivial to use the IPU DMA for generic data transfer as the data is reformatted during movement. This is typical for image based data (scaling, pixel format, etc.) but would increase overhead in the end application that uses the data.
The X-Gene and Tegra-K1 system on chips have more advanced PCI-Express controllers which do support DMA. A test system for these platforms has not been built but in theory there should be no significant issues.

5. Discussion, Conclusions and Future Work

High data throughput computing is required for projects such as the LHC which produce enormous amounts of raw data. A general purpose ARM System on Chip-based processing unit is being developed which will be used as a co-processor to the sROD to help mitigate the energy reconstruction issues caused by pile-up under higher luminosity operation of the LHC.

A PCI-Express interface will be used for the raw data transfer between the sROD and the PU. 2.8 GiB/s of data throughput is required to sustain the raw data from the sROD prototype. Initial throughput measurements presented for a pair of Freescale i.MX6 quad-core Cortex-A9 SoCs are between 283 and 350 MiB/s of the theoretical maximum 488 MiB/s (500 MB/s) for the available x1 link.

An Optimal Filtering algorithm was implemented and tested on ARM Cortex-A9, A15 and X-Gene (similar to Cortex-A57) SoCs. The slowest SoC, the Cortex-A9, achieved 350 MiB/s throughput on the algorithm. The use of gigabit Ethernet would not lead to an optimal system balance between CPU and external I/O and PCIe is the only alternative in this case. On the other platforms the system balance by using gigabit Ethernet is even worse.

The X-Gene SoC supports 10 Gb/s Ethernet which is approximately 1 GB/s of external I/O. This would result in a good system balance if only one CPU core was used. Four or eight cores are present on the X-Gene and so PCI-Express, where the maximum throughput is almost 16 GB/s on the X-Gene, would be a very powerful and flexible solution as an sROD co-processor or general purpose processing unit.

Acknowledgements

The financial assistance of the National Research Foundation towards this research is hereby acknowledged. We would also like to acknowledge the School of Physics, the Faculty of Science and the Research Office at the University of the Witwatersrand, Johannesburg.

References

[1] The ATLAS Collaboration 2010 *The European Physical Journal C* **70** 1193–1236
[2] Carrió F, Castillo V, Ferrer A, Fiorini L, Hernández Y, Higón E, Mellado B, March L, Moreno P, Reed R, Solans C, Valero A and Valls J A 2014 *Journal of Instrumentation* **9** C02019–C02019
[3] Rajovic N, Vilanova L, Villavieja C, Puzovic N and Ramirez A 2013 *Journal of Computational Science* **4** 439–443
[4] Cox M A and Reed R 2014 *Proceedings of SAIP2014, the 59th Annual Conference of the South African Institute of Physics* ed Engelbrecht C and Karataglidis S (Johannesburg) pp 180–184 ISBN 978-0-620-65391-6 URL http://events.saip.org.za
[5] Fullana E, Castelo J, Castillo V, Cuenca C, Ferrer A, Higon E, Iglesias C, Munar A, Poveda J, Ruiz-Martinez A, Salvachúa B, Solans C and Valls J 2005 Optimal Filtering in the ATLAS Hadronic Tile Calorimeter URL http://ific.uv.es/tical/Publications/InternalNotes/tilecal-2005-001.pdf
[6] Cox M A, Reed R and Mellado B 2015 *Journal of Instrumentation* **10** C01007–C01007
[7] Peralva B S 2013 *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)* (Seoul: IEEE) pp 1–6 ISBN 978-1-4799-0533-1
[8] Jacob B and Guennebaud G 2014 Eigen C++ Library URL http://eigen.tuxfamily.org