EPJ Web of Conferences **337**, 01166 (2025) https://doi.org/10.1051/epjconf/202533701166
*CHEP 2024*

# Versal ACAP processing for ATLAS-TileCal signal reconstruction

*Francisco* Hervás Álvarez[1,*], *Alberto* Valero Biot[1], *Luca* Fiorini[1], *Héctor* Gutiérrez Arance[1], *Fernando* Carrió[1], *Sonakshi* Ahuja[1], and *Francesco* Curcio[1]

[1]Instituto de Física Corpuscular (CSIC-UV), catedràtic José Beltran 2, Paterna, València, Spain

**Abstract.** Particle detectors at accelerators generate a large amount of data, requiring analysis to derive insights. Collisions lead to signal pile-up, where multiple particles produce signals in the same detector sensors, complicating individual signal identification. This contribution describes the implementation of a deep learning algorithm on a Versal Adaptive Compute Acceleration Platform (ACAP) device for improved processing via parallelization and concurrency. The system will emulate the real conditions in the ATLAS Tile Calorimeter signal reconstruction module. Connected to a host computer via Peripheral Component Interconnect express (PCIe), this system aims for enhanced speed and energy efficiency over Central Processing Units (CPUs) and Graphics Processing Units (GPUs). In the contribution, we will describe in detail the data processing and the hardware, firmware and software components of the system, as well as the implementation of the deep learning algorithm on Versal ACAP device, and the system for transferring data in an efficient way.

## 1 Introduction

The High-Luminosity LHC (HL-LHC) [1] is a major upgrade project planned for the existing Large Hadron Collider (LHC) [2], intended to both preserve and extend its discovery capabilities. However, the substantially increased collision rate associated with this upgrade introduces a number of significant challenges that must be addressed. As a consequence the ATLAS experiment has conceived a comprehensive roadmap of upgrades to effectively accommodate and harness the higher collision rates. The ATLAS trigger and data acquisition systems are being upgraded, which involves the implementation of faster and more efficient readout electronics and data transfer technologies. To address these demands, the ATLAS Tile Calorimeter [3] data acquisition system has been redesigned for the increased data throughput. During HL-LHC operation, the detector signals will be transferred off-detector to the PreProcessor modules. These modules must reconstruct the deposited energy for approximately 10,000 channels for each Bunch Crossing (BC) at a 40 MHz rate, with a strictly fixed and deterministic latency of maximum 200 ns [4]. Complex signal reconstruction algorithms based on deep learning are being investigated for Field-Programmable Gate Array (FPGA) implementation under these stringent conditions.

*e-mail: fran.hervas@ific.uv.es

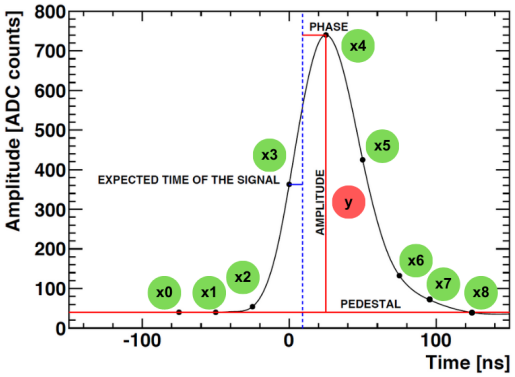


**Figure 1.** Analog pulse shape, represented in Analog to Digital Converter (ADC) counts, with the associated Bunch Crossings (BCs) ($x$) and the targeted amplitude ($y$), which is the central BC of the window.

## 2 Materials and Methods

### 2.1 Hardware setup

FPGAs are well suited to implement deep learning algorithms as they process math-based calculations simultaneously, whereas CPUs process tasks sequentially. The Versal AI Core VC1902 [5] is constituted by a System on Chip (SoC) that incorporates a traditional CPU—Processing System (PS)—and an FPGA-based Programmable Logic (PL) along with transceivers that can communicate outside of it. The chip is integrated on the VCK190 [6] development board, created by AMD-Xilinx. The hardware set-up is a host computer connected to the VCK190 development board using PCIe 4th generation through 8 lanes.

### 2.2 Firmware

#### *2.2.1 Signal reconstruction algorithm*

Before the signal reconstruction step the ATLAS Tile Calorimeter [3] pulses are acquired by the Photomultiplier (PMT) blocks and digitised by the Analog to Digital Converters (ADCs) at 40 MHz [4]. The aim of the signal reconstruction is to process these digital samples every clock cycle, targeting the amplitude of the central sample of the window (Figure 1).

A MultiLayer Perceptron (MLP) [7] is one of the proposed algorithms for signal reconstruction. It is a type of feedforward neural network consisting of fully connected neurons with nonlinear activation functions distributed through layers. The MLP used here consists of two neurons within each of its two layers (Figure 2). The simplicity of its form ensures that the digital components used in the design are within the requirements of the signal reconstruction process.

To introduce non-linearity and make the model better suited to recognize complex relationships within the input data, the hyperbolic tangent function, $tanh(x)$, is employed as the hidden layer's activation function. However, applying non-linear functions on an FPGA is computationally costly. To avoid this, a quantized $tanh(x)$ is used through a Lookup Table (LUT) of 5000 discrete values between -0.7 and 0.8 [7].

The Intellectual Property (IP) core used to implement the MLP consists of an Advanced eXtensible Interface (AXI) Stream Slave and Master interfaces, that are in charge of communicating with an outside AXI Direct Memory Access (DMA) for data transfer through the

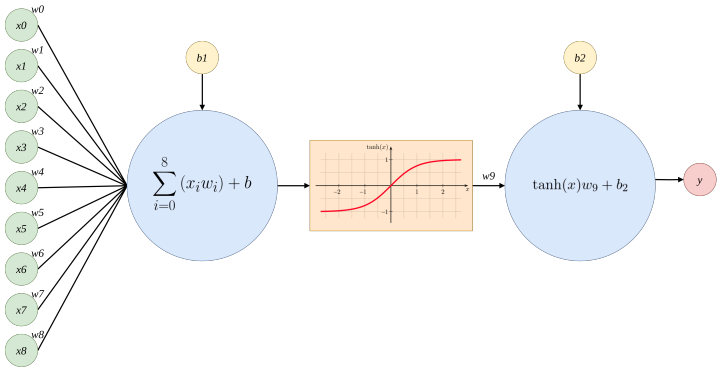


**Figure 2.** MultiLayer Perceptron (MLP) composed by 2 layers and 1 neuron on each layer (Configuration 1-1). The activation function in the hidden layer is the *tanh*(*x*) function. The first layer performs a weighted accumulation of the input window, adds a bias, and applies the activation function. The second layer applies a single weight to the output of the previous layer and adds a bias.

algorithm with a 32-bit data width on the AXI bus [8]. The IP core includes pre-processing and post-processing stages, which are necessary for normalizing and denormalizing data to be used by the deep learning algorithm. The read-out signal of every BC is held in a First In First Out (FIFO) buffer with serial input and parallel output. This ensures that all the required read-out BCs are accessible for processing[1], with the purpose of reconstructing the amplitude of the central BC. Finally, the signal reconstruction algorithm is implemented. VHDL is the hardware description language used to implement the IP core design (Figure 3).

Each stage of the IP core operates with a distinct clock domain. The design contains two clock domains: the BC clock domain, operating at 40 MHz, which is synchronous with the LHC operation, and the processing clock domain, which is 400 MHz. The BC clock domain is required to meet the needs of the ultimate real-time hardware device, and the processing clock domain is locked to a frequency to meet the processing latency needs for processing each BC. The final latency requirement is 8 BCs, or 8 clock cycles of the BC clock domain.

Fixed package as defined in the VHDL 2008 standard is used in the signal reconstruction algorithm, with fixed-point representation throughout the algorithm. All neurons' data input is 16 bits, with the decimal part occupying 10 bits and the integer part occupying 6 bits. The result of every neuron is 31 bits, 18 for the decimal part and 13 are for the integer part. Weights are encoded as 14-bit values, of which 8 are for the decimal and 6 are for the integer. Every bias is encoded as a 30-bit value, of which 18 are for the decimal part and 12 are for the integer part.

### *2.2.2 Programmable Logic integration*

The system is made up of several blocks to compose the entire PL. These components are described in the next list. The system representation in Vivado is shown in Figure 4.

- A central block known as the Control, Interfaces and Processing System (CIPS), which implements the CPU and the software it runs, as well as the PCIe interface for the physical port.

[1] In this implementation of the signal reconstruction algorithm, the samples of nine BCs are used as input.

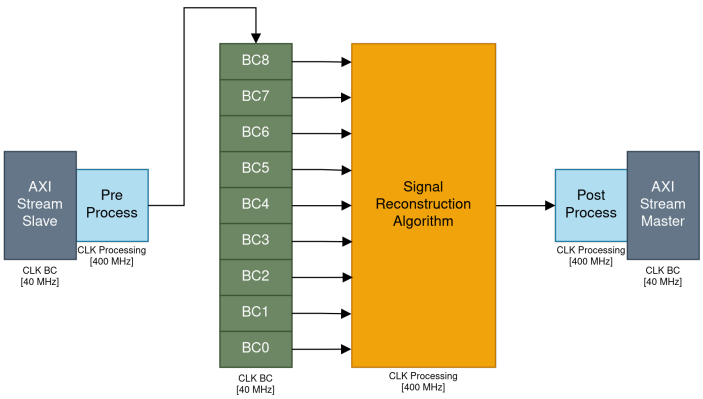


**Figure 3.** IP core implemented in Vivado. The implementation includes the AXI Stream (AXI) interface wrapper, the preprocess and postprocess stages, the FIFO and the signal reconstruction algorithm. Each clock has its own clock domain.

- The AXI Network on Chip (NoC), a high-bandwidth interconnect available in all Versal ACAP devices, connecting PL, PS, and other hardwired FPGA components like the Double Data Rate (DDR) memory controller.
- AXI SmartConnects (SMCs), which are multiplexers to interconnect various components within the PL side of the FPGA.
- An AXI DMA, which manages data transfer from DDR memory to the signal reconstruction IP core.
- The signal reconstruction core, which runs the deep learning algorithm via the AXIS interface.
- The clocking wizard, which converts the 40 MHz clock (BC clock domain) to the 400 MHz clock (processing clock domain).
- The processor system reset, which synchronizes the reset process.
- The Xilinx Direct Memory Access (XDMA) handshake IP, which manages data transfer between the device and the host computer.
- The XDMA interrupt generator IP, which generates an interrupt to the PS of the Versal device when new data is received from the host computer.

The system also implements a clock gating technique, based on applying an OR gate to the Global Clock Buffer with Clock Enable (BUFGCE) generated by the clocking wizard, with the valid signals of the AXIS bus, which turns off the processing IP core internal clock when not being used to reduce power consumption.

## 2.3 Software

### *2.3.1 Processing System*

The software running on the PS of the Versal device is built around three main routines: the main program, the interrupt handler for the PCIe reception flag, and the interrupt handler for the processed data flag (Figure 5).

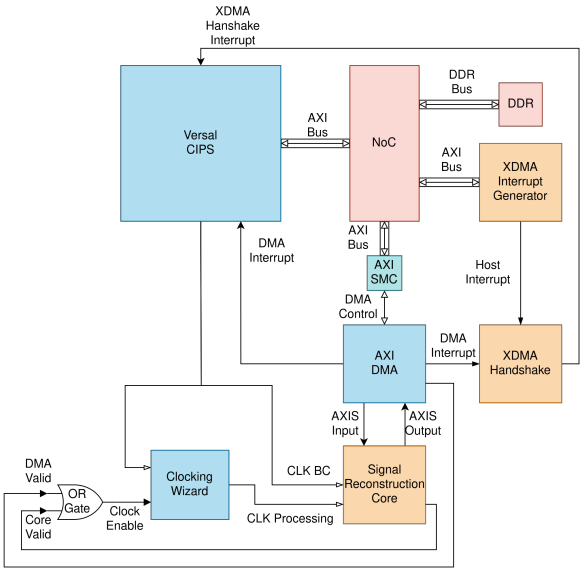


**Figure 4.** System integration at PL level in Vivado.

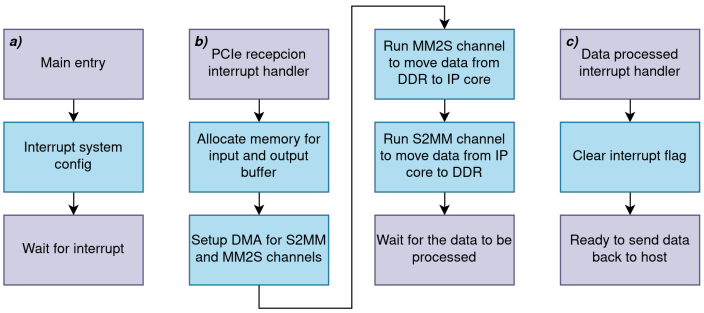


**Figure 5.** Simplified flow chart of the software located in the PS side of the Versal device. (**a**) Flow chart for the main loop of the program. (**b**) Interrupt handler for the PCIe reception flag. (**c**) Interrupt handler for data processed completion flag.

The main routine takes care of setting up the interrupt system, initializing the device, and then waiting for incoming interrupts. When new data arrives from the host computer to the DDR memory controller, the PCIe reception interrupt handler is triggered. At this point, it allocates memory for input and output buffers based on the number of BCs to be processed, configures the DMA transmission and reception channels, and starts the data transfer between DDR memory and the IP core.

On the other hand, the interrupt handler for the processed data flag steps in once the data has been processed. At this stage, the processed data is prepared for transmission back to the host computer, completing the cycle.

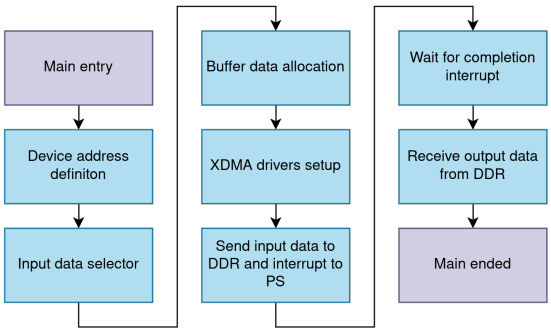


**Figure 6.** Simplified flow chart of the host software.

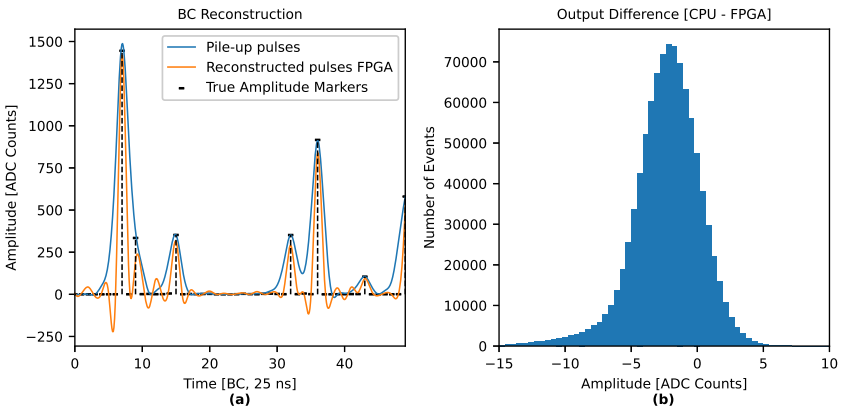


**Figure 7.** (**a**) Accuracy comparison between the input signal and the reconstructed signal by the Versal ACAP, with the reference of the true amplitude. The reconstructed pulses have been obtained from the processing in the VC1902 device. (**b**) A histogram showing the difference between the CPU (single precision floating point) implementation output and the FPGA (fixed point) implementation output.

### *2.3.2 Host drivers*

On the host computer side, a C-based driver is used for device communication (Figure 6). It begins with the declaration of the device's write buffer address and read buffer address and the selection of the data to be written. Memory space for the data is allocated and the XDMA drivers are initialized. After initialization is completed, the data is transferred to the DDR memory of the device, resulting in an interrupt to the PS. After waiting for the completion interrupt, the output data that is processed is read from the DDR memory of the device.

## 3 Results

The data used in this study has been generated using a custom simulator that emulates the detector read-out. Figure 7 (**a**) illustrates a 50 BCs time series with the amplitude reconstructed by the FPGA system, and the true amplitude generated by the simulation.

**Table 1.** Resource utilization of the signal reconstruction algorithm.

| Resource Type | IP Core Usage | System Usage | Max. Cores |
|---|---|---|---|
| FF (Slices) | 975 | 0.054 % | 1845 |
| LUT (Slices) | 825 | 0.092 % | 1090 |
| PL Memory (kb) | 216 | 0.113 % | 884 |
| DSP (Slices) | 7 | 0.356 % | 281 |
| BC Freq. (MHz) | 40 | - | - |
| Processing Freq. (MHz) | 400 | - | - |
| Bus data width (bits) | 32 | - | - |
| Bandwidth (MB/s) | 160 | 0.156 % | 640 |

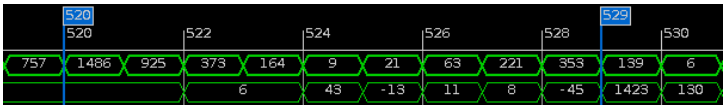


**Figure 8.** Input stream from the DMA to the algorithm and output stream from the algorithm to the DMA. The latency measured with the Integrated Logic Analyzer (ILA) is 9 clock cycles.

For the specified fixed-point lengths, Figure 7 (**b**) illustrates the difference between the algorithm output for single-precision floating-point and fixed-point representations. The value of the difference is biased towards negative values because the most typical amplitude after normalization is under zero. Since the normalized range is between -1 and 1, and the most typical amplitude value of the simulated data is in the range from -1 to 0, fixed-point truncation increases the value of the data, resulting in a higher output value than the floating-point version. Due to that, the Gaussian distribution is slightly under 0.

Table 1 shows the resources used by the algorithm implementation. The most critical information is the usage of Digital Signal Processors (DSPs) in the signal reconstruction IP core because this resource limits the number of core replications on the FPGA. Thus, the maximum number of cores that can be instantiated in the VC1902 device is 281.

The latency requirement is fulfilled since the IP core imposes 9 clock cycle latency due to the AXIS interface, which imposes a delay of 2 clock cycles (Figure 8). In a real-time application, removal of the interface reduces the total latency by 2 clock cycles, making it 7 clock cycles, which is less than the required limit of 8 clock cycles.

When a single processing core is utilized, the FPGA consumes less power than the CPU and GPU. Power consumption was measured by running one core on the FPGA and one thread on the CPU and GPU (Figure 9).

## 4 Conclusions

The installed setup provides a testbed for testing novel deep learning algorithms to reconstruct signals emulating the hardware of the ATLAS Tile Calorimeter [3] for the HL-LHC. Fixed-point computation is employed not only to optimize the usage of resources but also to supply deterministic results. While floating-point computation offers higher dynamic range, it is nondeterministic in output, while fixed-point processing is deterministic for the same input.

The results indicate that for data processing, a single-core FPGA achieves less power consumption compared to a CPU or GPU. This is because an FPGA can execute a single core without activating the remaining logic, whereas a GPU executes all of its pre-defined hardware even though it may not be fully engaged in processing. Additionally, clock gating

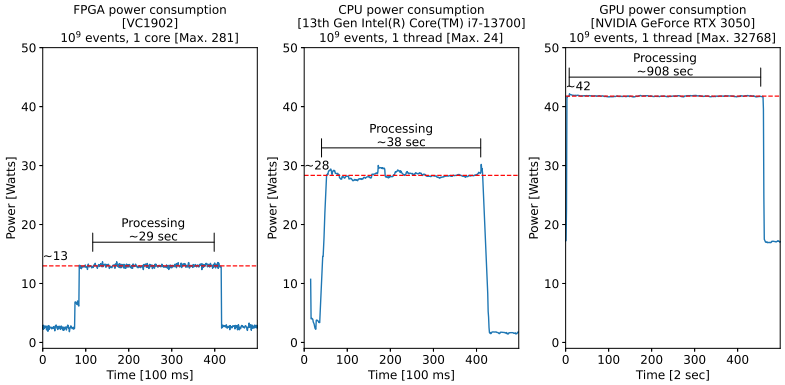


**Figure 9.** Power consumption measured for FPGA, CPU and GPU. The number of events used for the test is $10^9$.

techniques are employed in a way to shut down processing logic when idle, reducing power consumption even further.

All these outcomes can be employed in the future to create a complete system, where all the FPGA resources are utilized in an effort to parallelize processing tasks, while performance as well as efficiency is boosted compared to traditional processing systems.

The publication is part of the project TED2021-130852B-I00, financed by MCIN/AEI/10.13039/501100011033 and the EU "NextGenerationEU"/PRTR.

## References

[1] O. Aberle, I. Béjar Alonso, O. Brüning, P. Fessia, L. Rossi, L. Tavian, M. Zerlauth, C. Adorisio, A. Adraktas, M. Ady et al., High-Luminosity Large Hadron Collider (HL-LHC): Technical design report, CERN Yellow Reports: Monographs (CERN, Geneva, 2020), `https://cds.cern.ch/record/2749422`

[2] O.S. Brüning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, P. Proudlock, LHC Design Report, CERN Yellow Reports: Monographs (CERN, Geneva, 2004), `http://cds.cern.ch/record/782076`

[3] ATLAS Collaboration, Operation and performance of the ATLAS tile calorimeter in LHC Run 2, The European Physical Journal C **84**, 1313 (2024). 10.1140/epjc/s10052-024-13151-4

[4] K. Einsweiler, L. Pontecorvo (ATLAS Collaboration), Tech. rep., CERN, Geneva (2017), `http://cds.cern.ch/record/2285583`

[5] AMD-Xilinx, Versal Adaptive SoC Technical Reference Manual (AM011), 1st edn. (2023)

[6] AMD-Xilinx, VCK190 Evaluation Board User Guide (UG1366), 1st edn. (2024)

[7] J. Arciniega, F. Carrió, A. Valero, FPGA implementation of a deep learning algorithm for real-time signal reconstruction in particle detectors under high pile-up conditions (2019). 10.1088/1748-0221/14/09/P09002

[8] ARM, AMBA® AXI Protocol Specification (2023)