particles

MDPI

Article

Versal Adaptive Compute Acceleration Platform Processing for ATLAS-TileCal Signal Reconstruction

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

Instituto de Física Corpuscular (CSIC-UV), catedràtic José Beltran 2, 46980 Paterna, València, Spain; alberto.valero@cern.ch (A.V.B.); luca.fiorini@cern.ch (L.F.); hector.gutierrez@ific.uv.es (H.G.A.); fernando.carrio@cern.ch (F.C.); sonakshi.ahuja@ific.uv.es (S.A.); francesco.curcio@ific.uv.es (F.C.)

* Correspondence: fran.hervas@ific.uv.es

Abstract: Particle detectors at accelerators generate large amounts 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. 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. The contribution presents the implementation of the deep-learning algorithm on a Versal ACAP device, as well as the system for transferring data in an efficient way.

Keywords: FPGA; SoC; HL-LHC; VHDL; power

check for updates

Academic Editors: Tommaso Dorigo, Roberto Ruiz de Austri Bazan, Jose Salt and Pietro Vischia

Received: 3 February 2025
Revised: 28 February 2025
Accepted: 19 April 2025
Published: 1 May 2025

Citation: Hervás Álvarez, F.; Valero Biot, A.; Fiorini, L.; Gutiérrez Arance, H.; Carrió, F.; Ahuja, S.; Curcio, F. Versal Adaptive Compute Acceleration Platform Processing for ATLAS-TileCal Signal Reconstruction. *Particles* **2025**, *8*, 49. https://doi.org/10.3390/particles8020049

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the new era of the Large Hadron Collider (LHC) [1], the High-Luminosity LHC (HL-LHC) [2], multiple challenges emerge. As a case of study, the ATLAS Tile Calorimeter [3] will have an upgrade to a luminosity corresponding to an average of up to 200 simultaneous proton–proton interactions per Bunch Crossing (BC) [4]. This phenomenon is known as signal pile-up, and to mitigate its impact new readout systems are needed, along with updated design requirements in the HL-LHC. These requirements include keeping the degradation of energy resolution for a single particle below 10% and the energy bias below 1%. Signals will be reconstructed for every BC at 40 MHz before the first level of trigger, and the processing will be made using Field-Programmable Gate Arrays (FPGAs) due to their low and deterministic latency for signal synchronization. In addition, more sophisticated algorithms are needed for signal reconstruction, such as deep-learning algorithms.

The goal of this study is to implement a system where multiple signal reconstruction algorithms can be tested in real hardware, using real electronic requirements. The device selected for this contribution is the AMD-Xilinx Versal ACAP VC1902 [5], which is implemented in the development board VCK190 [6].

2. Materials and Methods

2.1. Hardware Setup

FPGAs are ideal for implementing deep-learning algorithms because they perform parallel mathematical calculations, unlike traditional CPUs, which operate sequentially. The system is implemented using the Versal AI Core VC1902, a System on Chip (SoC) that combines a traditional CPU—Processing System (PS), FPGA—Programmable Logic (PL), and transceivers for external communication. This integration enhances overall performance. The device is integrated into the VCK190 development board made by AMD-Xilinx. The hardware setup is composed of a host computer that is connected to the VCK190 development board by PCIe 4th generation with eight lines.

2.2. Firmware

2.2.1. Signal Reconstruction Algorithm

The core objective of the signal reconstruction system is to process nine BCs during each clock cycle, in a similar way as will be done in the ATLAS Tile Calorimeter [3]. The algorithm will take these nine values as input and will focus on predicting the target amplitude corresponding to the central BC within the given window (Figure 1).

The proposed algorithm for the signal reconstruction is a MultiLayer Perceptron (MLP) [7], which is a type of feedforward artificial neural network known for its capability to model complex relationships between input and output data. The MLP utilized in this study consists of two layers, with a single neuron in each layer (Figure 2). This minimalist architecture is specifically designed to balance simplicity and effectiveness, ensuring efficient signal processing with minimal computational overhead.

To introduce non-linearity and improve the model's ability to capture complex patterns in the input data, the hyperbolic tangent function, $tanh(x)$, is employed as the activation function in the hidden layer. Hyperbolic tangent is chosen because it is zero-centered, meaning that input data must be normalized in the range from -1 to 1, which helps in stabilizing weight updates during training.

A hyperbolic tangent, $tanh(x)$, is used to implement the non-linear activation function of the MLP first layer. However, implementing non-linear functions on an FPGA is resource-intensive. A quantized version of $tanh(x)$ is used via a Lookup Table (LUT), with 5000 discrete values in the range of -0.7 to 0.8 to address this issue [7].

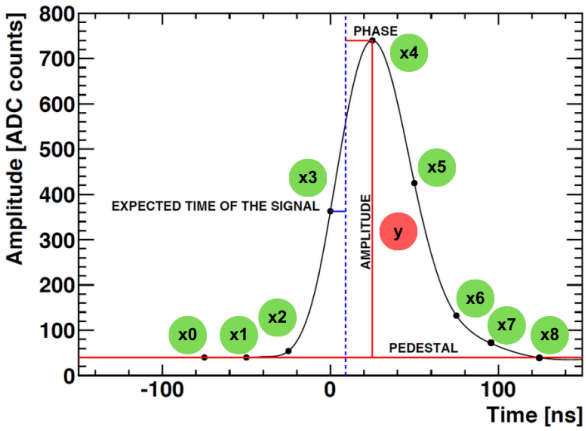


Figure 1. Analog pulse shape with the associated BC and the targeted amplitudes.

The Intellectual Property (IP) core used to implement the MLP is composed of an Advanced eXtensible Interface (AXI) Stream Slave and a Master interfaces, which are responsible for communicating with an external AXI Direct Memory Access (DMA) for moving the data across the algorithm, with 32 bits of data size for the AXI bus [8]. Also, pre-

processing and post-processing stages are implemented in the IP core, which is necessary for normalizing and denormalizing the data for the deep-learning algorithm. The signal of each read-out channel for each BC is stored on a First In First Out (FIFO) with serial input and parallel output. That is because there is a requirement of having the last 9 BCs accessible for the processing, targeting the amplitude of the central BC. Finally, the signal reconstruction algorithm is implemented. The hardware description language used for implementing the IP core is VHDL (Figure 3).

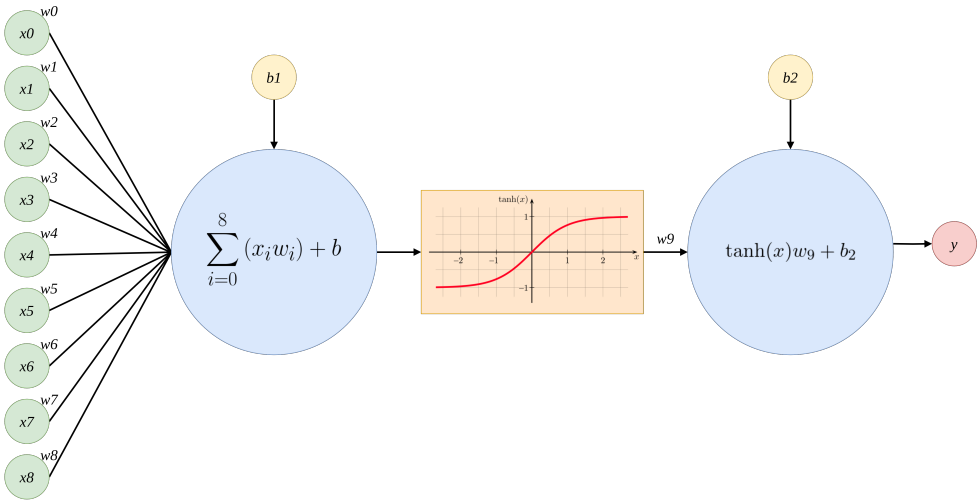


Figure 2. MLP composed of 2 layers and 1 neuron on each layer (Configuration 1-1). The activation function in the hidden layer is the $tanh(x)$ function.

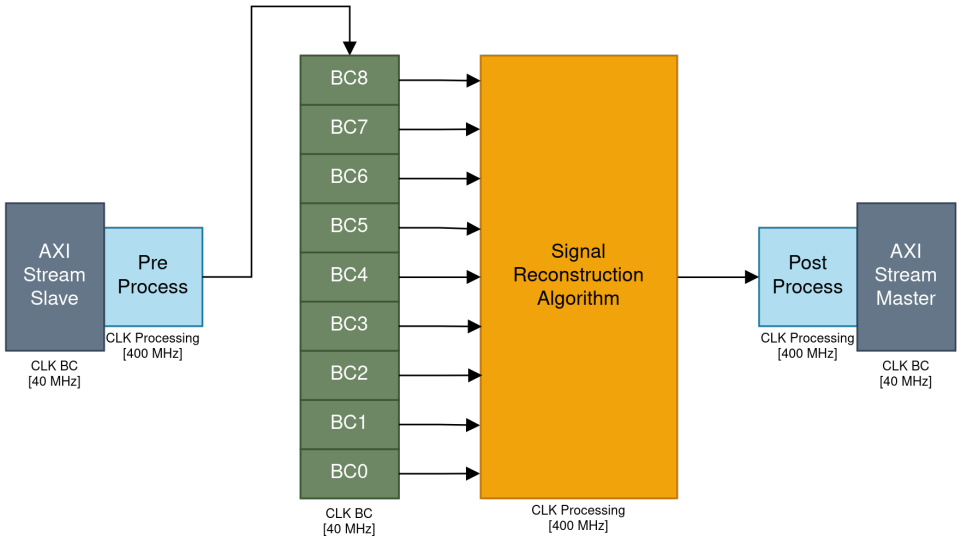


Figure 3. IP core implemented in Vivado. The implementation includes the AXI Stream (AXI) interface wrapper, the preprocess and postprocess stages, the FIFO and the signal reconstruction algorithm. Each clock has its own clock domain.

Each stage of the IP core is controlled by a different clock domain. There are two clock domains in the design: the BC clock domain, which is 40 MHz, and the processing clock domain, which is 400 MHz. The BC clock domain is the one that is the requirement of the final real-time hardware device, and the processing clock domain is the fixed frequency for achieving the latency requirements for the processing of each BC. The final latency requirement is 8 BCs, or 8 clock cycles of the BC clock domain.

In the signal reconstruction algorithm, a fixed package is used, which is a standard package of the VHDL 2008 standard [9]. Using this package, fixed point representation is used during all of the process. The input of each neuron data is 16 bit width, with the decimal part being 10 bit width and the integer part 6 bit width. The output of each neuron data is 31 bit width, with the deicmal part being 18 bit width and the integer part 13 bit width. The weights are stored with 14 bit width, with the decimal part being 8 bit width and the integer part 6 bit width. Each bias is stored with 30 bit width, with the decimal part being 18 bit width and the integer part 12 bit width.

2.2.2. Programmable Logic Integration

The system consists of a central block known as the Control, Interfaces and Processing System (CIPS), which implements the CPU and, therefore, the software that runs on it, and the PCIe interface for the physical port; the AXI Network on Chip (NoC), which is the high bandwidth interconnect that is implemented on all Versal ACAP devices, and is used for connecting PL, PS and other hard-coded parts of the FPGA, such as the Double Data Rate (DDR) memory controller; AXI SmartConnects (SMCs), which are multiplexers for connecting different parts in the PL side of the FPGA; the AXI DMA, which is the component that moves data from DDR memory to the signal reconstruction IP core; the signal reconstruction core, which is the deep-learning algorithm with the AXIS interface; the clocking wizard, which is used to convert 40 MHz clock (BC clock domain) to 400 MHz clock (processing clock domain); the processor system reset, which is used to synchronize the reset; the Xilinx Direct Memory Access (XDMA) handshake IP, which implements a handshake used by the XDMA responsible to move data between the device and the host computer; and the XDMA interrupt generator IP, which is in charge of generating an interrupt to the PS of the Versal device when new data are received from the host computer (Figure 4).

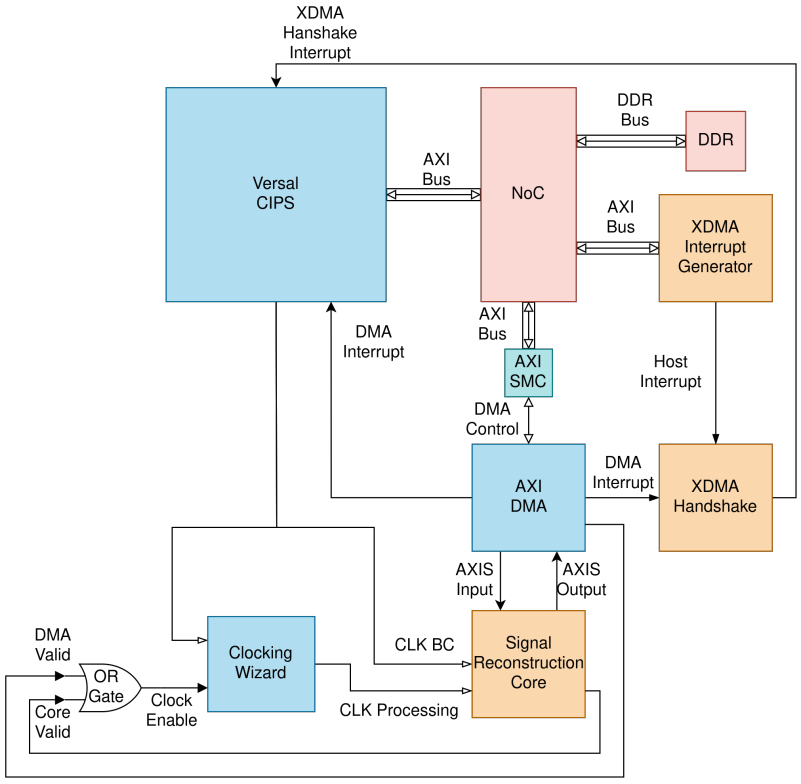


Figure 4. System integration at PL level in Vivado.

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 in order to reduce power consumption.

2.3. Software

2.3.1. Processing System

The PS software of the Versal device is based on three main routines: the main program, the interrupt handler for the PCIe reception flag and the interrupt handler for the data processed flag (Figure 5). The main routine configures the interrupt system and initializes the device and starts waiting for interrupts. The interrupt handler for the PCIe reception flag is run when there is new data coming from the host computer to the DDR memory controller and allocates memory for the input and output buffers, depending on the number of BCs to process, configures the transmission and reception channels of the DMA and runs those channels to move data from DDR memory to the IP core, runs those channels to move data from the IP core to the DDR memory, and waits for the data to be totally processed. The interrupt handler is run for the data processed flag, which clears the interrupt flag. When the interrupt handler for the data processed routine is complete, the data are ready to be sent back to the host computer.

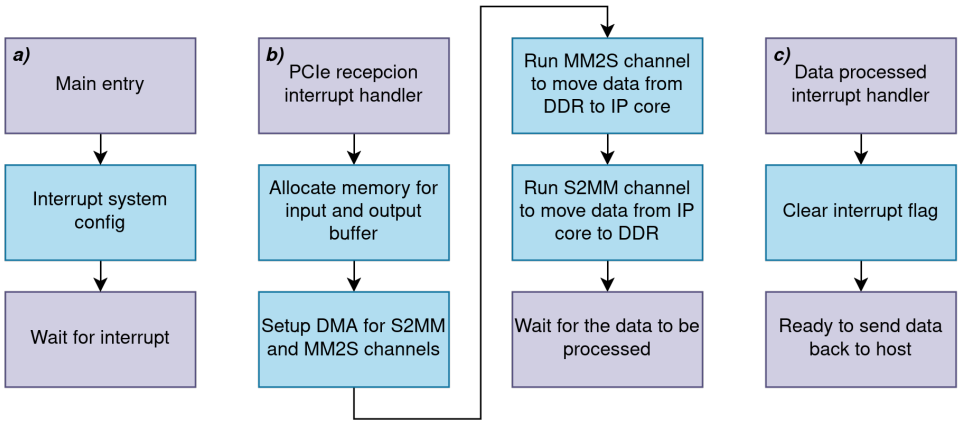


Figure 5. Simplified flow chart of the software located in the PS side of the Versal device. (**a**) Flow chart for the main loop of the program. (**b**) Interrupt handler for the PCIe reception flag. (**c**) Interrupt handler for data processed completion flag.

2.3.2. Host Drivers

On the host computer side of the system, a driver is implemented to communicate with the device, written in C programming language (Figure 6). First, the write and read buffer addresses of the device are defined, and the input data to be sent are selected. Next, memory is allocated for the data, and the XDMA drivers are configured. Once everything is set up, the data are transferred to the device's DDR memory, and an interrupt is triggered to the PS. After waiting for the completion interrupt, the output data are retrieved from the device's DDR memory.

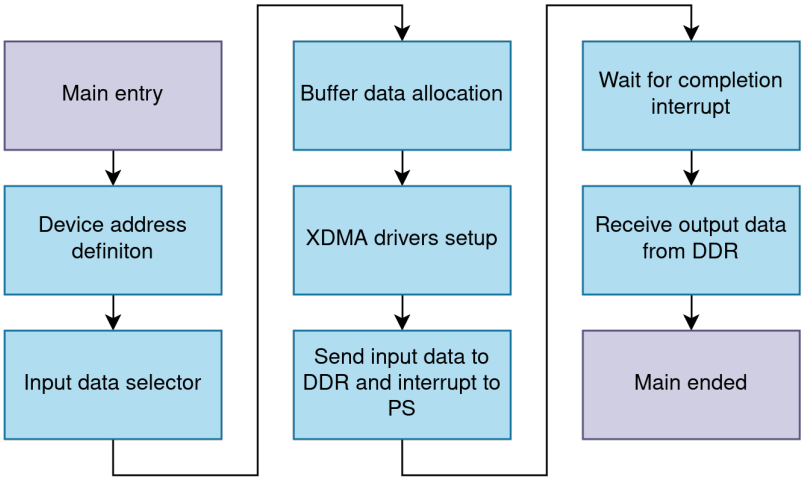


Figure 6. Simplified flow chart of the host software.

3. Results

The results obtained for this study are shown in this section. A time series of 50 BC with pile-up, the reconstructed amplitude using the actual FPGA system and the true amplitude are shown in Figure 7a.

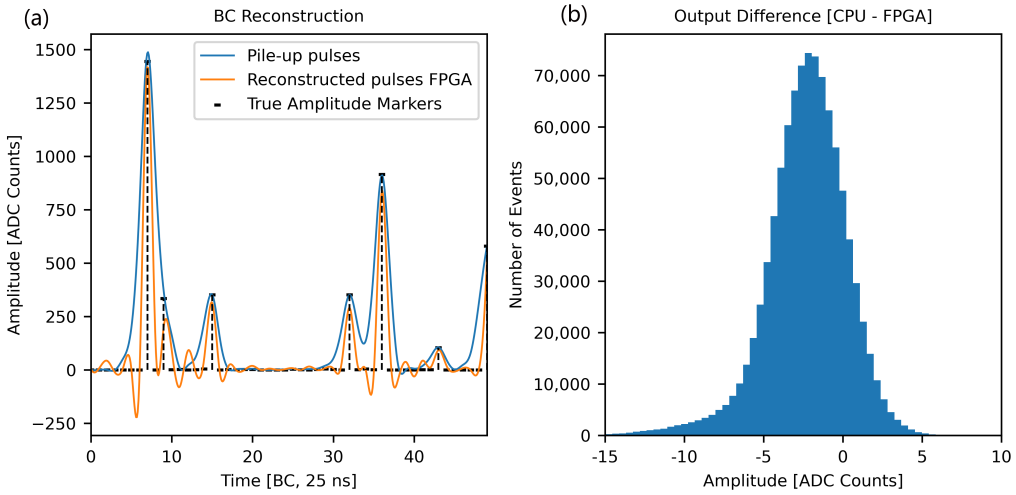


Figure 7. (**a**) Accuracy comparison between the input signal and the reconstructed signal by the Versal ACAP, with the reference of the true amplitude. The reconstructed pulses have been obtained from the processing in the VC1902 device. (**b**) A histogram showing the difference between the CPU (single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

For the specified fixed point lengths, the difference between the output of the algorithm using single precision floating point and the fixed point representations is shown in Figure 7b. The normal difference is centered below 0 because normalizing the inputs in the preprocessing stage makes them negative in the majority of cases for the FPGA processing. That is because, in the normalized range between -1 and 1, the typical amplitude of the test data is between -1 and 0. Therefore, negative numbers always have a greater value after the fixed point truncation. Due to this phenomenon, the fixed point implementation result is greater than the floating point implementation result.

The most relevant piece of information in the resource utilization chart (Table 1) is the number of Digital Signal Processors (DSPs) used in the signal reconstruction IP core,

which is the constraining resource for replicating the core across the FPGA. Therefore, the maximum number of cores that can be implemented in the VC1902 device is 281.

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 *

* Those numbers are based on the DDR Memory bandwidth.

The latency requirement is met since the latency of the IP core is 9 clock cycles due to the AXIS interface, which implements 2 clock cycles of latency (Figure 8). Removing this interface in the real-time application reduces the total latency of the IP core by 2, having 7 clock cycles of latency, which is below the requirement of 8 clock cycles.

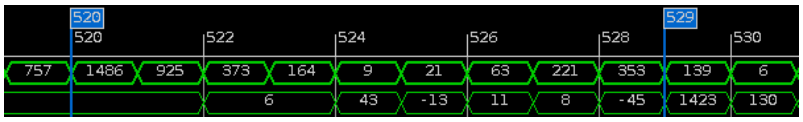


Figure 8. Input stream from the DMA to the algorithm and output stream from the algorithm to the DMA. The latency measured with the Integrated Logic Analyzer (ILA) is 9 clock cycles.

The power consumption, using only 1 processing core, is lower on the FPGA than on the CPU and the GPU. The power consumption has been measured by implementing 1 core on the FPGA and using 1 thread on the CPU and on the GPU (Figure 9). On an FPGA, power consumption is proportional to the logic used. That is, only the necessary operations are implemented, unlike CPU or GPU, where the architecture is fixed regardless of the application. Additionally, an FPGA can achieve the same throughput as other devices at much lower clock speeds through hardware concurrency, leading to power savings.

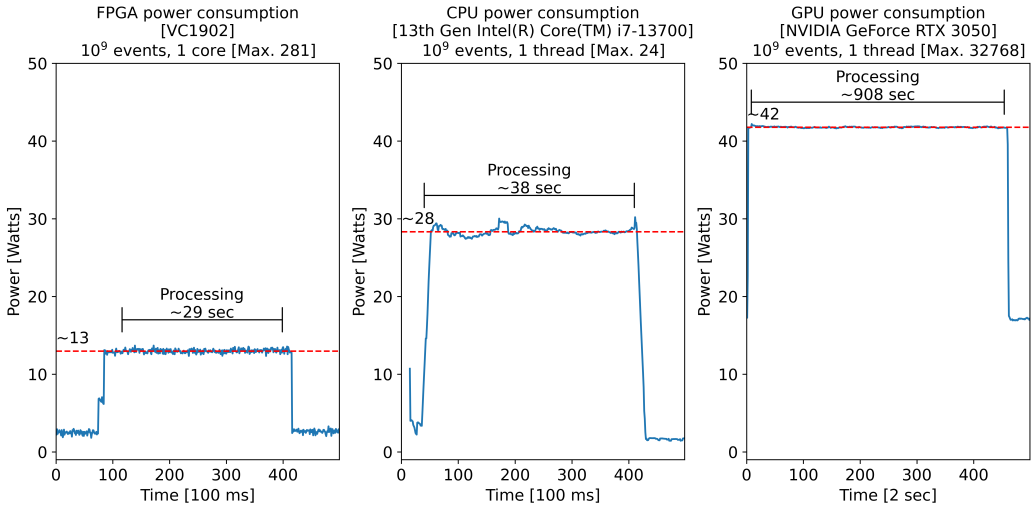


Figure 9. Power consumption measured for FPGA, CPU and GPU implementations. The number of events used for the test is 10^9.

4. Conclusions

The implemented system serves as a base for testing new deep-learning algorithms for signal reconstruction in real hardware, using the real conditions of HL-LHC calorimeter electronics. The reason for using fixed point arithmetic is not only due to the resource utilization limitation but also because of the need of having a deterministic result. It is true that using floating point arithmetic increases dynamic range, but at the cost of increasing the randomness of the result, while fixed point computing gives the same result for the same inputs every time.

The obtained results show better performance for offline processing in terms of power consumption for the FPGA implementing only 1 processing core. That is because, in an FPGA, only one core can be implemented without using the rest of the logic and, for example in a GPU, all the hardware is defined, even without using it for the processing task. Additionally, clock gating techniques are used to disable the processing logic when not being used.

These results can be used in the future to implement a complete system where all the resources of the FPGA are used to parallelize all of the processing tasks and increase the performance and the efficiency over traditional processing systems.

Author Contributions: Conceptualization, A.V.B., F.C. (Fernando Carrió), L.F., F.H.Á. and H.G.A.; data curation, A.V.B., F.C. (Francesco Curcio) and S.A.; formal analysis, F.H.Á., A.V.B., F.C. (Francesco Curcio) and S.A.; funding acquisition, L.F. and A.V.B.; investigation, A.V.B., F.H.Á., F.C. (Fernando Carrió) and H.G.A.; methodology, F.H.Á., A.V.B. and H.G.A.; project administration, L.F. and A.V.B.; resources, L.F. and A.V.B.; software, F.H.Á. and H.G.A.; supervision, A.V.B. and L.F.; validation, H.G.A. and F.H.Á.; visualization, F.H.Á.; writing—original draft, F.H.Á.; writing—review and editing, L.F., A.V.B. and H.G.A. All authors have read and agreed to the published version of this manuscript.

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

Data Availability Statement: https://gitlab.cern.ch/fhervasa/versal_signal_reco/-/releases/v1.0.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Brüning, O.S.; Collier, P.; Lebrun, P.; Myers, S.; Ostojic, R.; Poole, J.; Proudlock, P. *LHC Design Report*; CERN Yellow Reports: Monographs; CERN: Geneva, Switzerland, 2004. [CrossRef]
2. Aberle, O.; Béjar Alonso, I.; Brüning, O.; Fessia, P.; Rossi, L.; Tavian, L.; Zerlauth, M.; Adorisio, C.; Adraktas, A.; Ady, M.; et al. *High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report*; CERN Yellow Reports: Monographs; CERN: Geneva, Switzerland, 2020. [CrossRef]
3. ATLAS Collaboration. Operation and performance of the ATLAS tile calorimeter in LHC Run 2. *Eur. Phys. J. C* **2024**, *84*, 1313. [CrossRef]
4. Einsweiler, K.; Pontecorvo, L. *Technical Design Report for the Phase-II Upgrade of the ATLAS Tile Calorimeter*; Technical report; CERN: Geneva, Switzerland, 2017.
5. AMD-Xilinx. *Versal Adaptive SoC Technical Reference Manual (AM011)*, 1.6 ed.; AMD, Inc.: Sunnyvale, CA, USA, 2023. Available online: https://docs.amd.com/r/en-US/am011-versal-acap-trm (accessed on 15 April 2025).
6. AMD-Xilinx. *VCK190 Evaluation Board User Guide (UG1366)*, 1.2 ed.; AMD, Inc.: Sunnyvale, CA, USA, 2024. Available online: https://docs.amd.com/r/en-US/ug1366-vck190-eval-bd (accessed on 15 April 2025).
7. Arciniega, J.; Carrió, F.; Valero, A. FPGA implementation of a deep learning algorithm for real-time signal reconstruction in particle detectors under high pile-up conditions. *J. Instrum.* **2019**, *14*, P09002. [CrossRef]

8. ARM. *AMBA® AXI Protocol Specification*, issue K ed.; Arm Ltd.: Cambridge, UK, 2023. Available online: https://developer.arm.com/documentation/ihi0022/latest/ (accessed on 15 April 2025).
9. *IEEE Std 1076-2008*; IEEE Standard VHDL Language Reference Manual. IEEE: New York, NY, USA, 2009. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.