9

A Scalable, Heterogeneous Hardware Platform for Accelerated AIoT based on Microservers

R. Griessl[1], F. Porrmann[1], N. Kucza[1], K. Mika[1], J. Hagemeyer[1], M. Kaiser[1], M. Porrmann[2], M. Tassemeier[2], M. Flottmann[2], F. Qararyah[3], M. Waqar[3], P. Trancoso[3], D. Ödman[4], K. Gugala[5], and G. Latosinski[5]

[1]Bielefeld University, Germany
[2]Osnabrück University, Germany
[3]Chalmers University of Technology, Sweden
[4]EMBEDL AB, Sweden
[5]Antmicro, Poland
E-mail: rgriessl@techfak.uni-bielefeld.de;
fporrmann@techfak.uni-bielefeld.de;
nkucza@techfak.uni-bielefeld.de; kmika@techfak.uni-bielefeld.de;
jhagemey@techfak.uni-bielefeld.de; mkaiser@techfak.uni-bielefeld.de;
mporrmann@uni-osnabrueck.de; marco.tassemeier@uni-osnabrueck.de;
mflottmann@uni-osnabrueck.de; qarayah@chalmers.se;
waqarm@chalmers.se; ppedro@chalmers.se; zouzoula@chalmers.se;
daniel@embedl.ai; kgugala@antmicro.com; glatosinski@antmicro.com

Abstract

Performance and energy efficiency are key aspects of next-generation AIoT hardware. This chapter presents a scalable, heterogeneous hardware platform for accelerated AIoT based on microserver technology. It integrates several accelerator platforms based on technologies like CPUs, embedded GPUs, FPGAs, or specialized ASICs, supporting the full range of the cloud−edge-IoT continuum. The modular microserver approach enables the integration

of different, heterogeneous accelerators into one platform. Benchmarking the various accelerators takes performance, energy efficiency, and accuracy into account. The results provide a solid overview of available accelerator solutions and guide hardware selection for AIoT applications from the far edge to the cloud.

Keywords: IoT, machine learning, AIoT, microserver, deep learning, (far) edge-computing, FPGA, accelerator, energy-efficiency, performance classification.

9.1 Introduction

Looking into novel architectures optimized to accelerate the computation of neural networks, adaptable and scalable hardware solutions tailored to the applications' requirements are a key component. A fully featured, heterogeneous hardware platform integrating several accelerators is described and evaluated in the following. Over the last years, a large number of diverse DL accelerators in the form of special ASICs or IP cores, as well as GPU- or FPGA-based solutions, have been introduced in the market. This chapter focuses on benchmarking, and a comparative evaluation of selected accelerators regarding performance, energy efficiency, and accuracy is performed. Together with the seamless integration of DL into the IoT hardware platforms, the benchmarking methodology is used for further optimizing applications toward performance and energy efficiency. The presented work has been part of the VEDLIoT project [1]. In this chapter, we present a summary of the results obtained. More details are available in the respective project deliverables [2], [3].

9.2 Heterogeneous Hardware Platform for the Cloud-edge-IoT Continuum

This section deals with the hardware architecture and presents the different accelerators evaluated. It also acts as an introduction and classification for the different accelerators used in the benchmarking section.

The hardware platform can be used as a joint infrastructure for different developments. It supports a wide range of AIoT applications that can be addressed using a flexible communication infrastructure and exchangeable microservers. Figure 9.1 shows the RECS platforms covering application

# Sites	>100K	>10K	100-10K	<100
Footprint	Custom	Compact (1RU)	Medium (2RU)	Large (3RU)
Power Budget	<30 W	< 500 W	500 W – 2 KW	> 2 KW
# Microserver	max 2	up to 3	up to 48	up to 144

Figure 9.1 Overview of modular and scalable RECS platforms.

domains from embedded/far-edge computing toward cloud computing. All platforms commonly target heterogeneous computing with tightly coupled microservers. The cloud computing platform RECS|Box consists of either two or three rack units and aims for high-density applications using hundreds of microservers with high-bandwidth communication requirements. t.RECS houses up to three microservers in one rack unit and focuses on edge computing scenarios with low-latency demands like image and video processing use cases or 5G base stations. u.RECS rounds off the range of the RECS family toward low-power and compact embedded computing.

Microservers are based on industry-standard computer-on-module (COM) form factors, allowing for flexible and heterogeneous processing. On the one hand, RECS|Box and t.RECS support microservers that are based on COM express and COM-HPC server and client standards. The u.RECS, on the other hand, supports multiple compact form factors for far-edge computing, including SMARC, Jetson NX, Xilinx Kria, and Raspberry Pi compute modules.

9.2.1 Cloud computing platform RECS|Box

The RECS|Box platform is available in two different chassis sizes. The small chassis with 2U (Durin) is meant as a starter chassis, mainly for evaluation and non-datacenter use cases, while the 3U (Deneb) chassis is to be used in larger installations. The RECS|Box server architecture supports microservers based on x86 (e.g., Intel Xeon), 64-bit ARM mobile/embedded SoCs, 64-bit ARM server processors, FPGAs, GPUs, as well as other PCIe-based acceleration units. The smaller Durin can be equipped with up to 9 high-performance (HP) microservers or with 48 low-power (LP) microservers, and the larger Deneb can host 27 HP microservers or 144 LP microservers. The large amount of microservers inside the systems requires a sophisticated

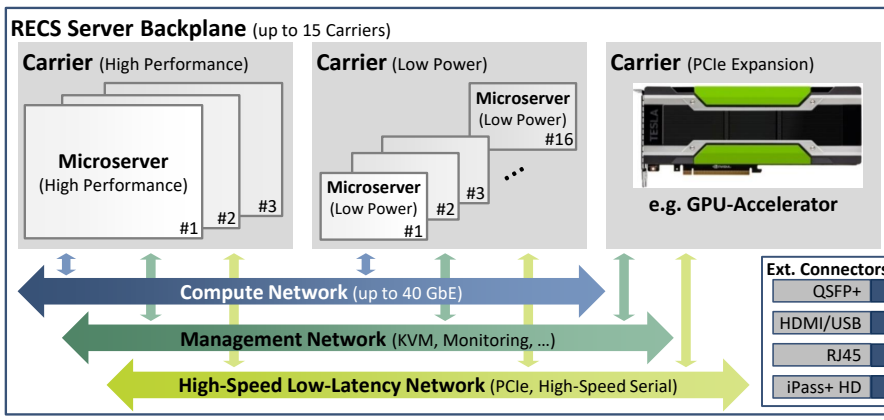

Figure 9.2 Communication architecture of RECS|Box platform.

communication infrastructure. Therefore, the RECS|Box comes up with multiple communication standards depicted in Figure 9.2.

The basis is the Ethernet network. It provides multiple 1 and 10 Gbit/s links to every microserver. Furthermore, it is internally switched and supports upstream bandwidth toward the top of the rack (ToR) switch up to 120 Gbit/s, combining three 40 Gbit/s links. In addition to the Ethernet communication infrastructure, a dedicated high-speed low-latency (HSLL) communication network is integrated into the RECS|Box architecture. It consists of two levels. On the physical level, the HSLL can directly connect high-speed serial links between microservers, as commonly available in FPGA modules. For processor-driven microservers (e.g., x86 based), the second level is PCIe-based direct host-2-host communication. Similar to the Ethernet network, it is internally switched and provides bandwidth of up to 56 Gbit/s to every microserver. The bandwidth toward a PCIe ToR switch is up to 336 Gbit/s, combining three 112 Gbit/s links.

9.2.2 Near-edge computing platform t.RECS

While the RECS|Box cloud hardware, described in the section above, focuses on data center applications, the edge server architecture supports local applications with high demands for low-latency, safety, and security. Especially applications with user interaction require local (pre-) processing and reduction of large amounts of data, which are difficult to achieve using a cloud-based approach. Three microserver modules of the COM-HPC standard

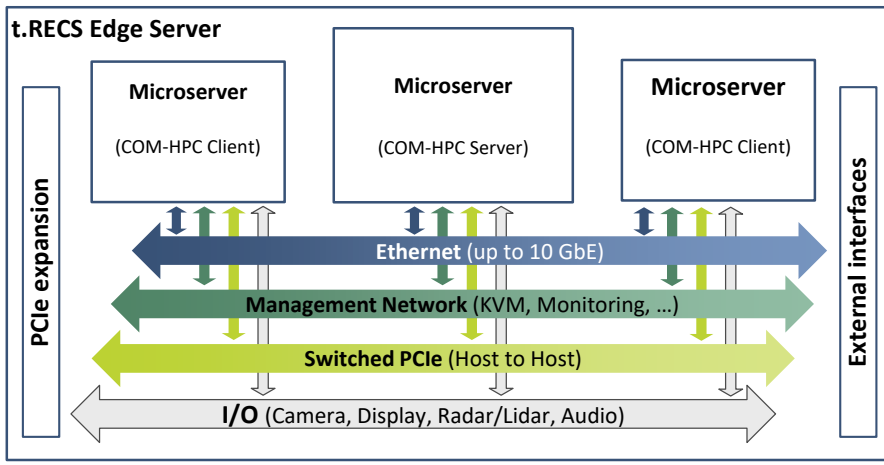

Figure 9.3 Communication architecture of t.RECS platform.

can be placed on the carrier board, supporting microservers based on x86 (e.g., Intel Xeon), 64-bit ARM server processors, FPGAs, GPU SoCs (e.g., NVIDIA Jetson AGX), as well as PCIe-based acceleration units via the PCIe expansion slot.

The t.RECS has a powerful and scalable communication infrastructure as shown in Figure 9.3. It is derived from the RECS|Box cloud platform and provides the basis for closely coupled heterogeneous compute nodes. The internal bandwidth for Ethernet, as well as HSLL, is the same as that in the RECS|Box, but the external bandwidth is reduced to single external links of 40 Gbit/s for Ethernet and 112 Gbit/s for HSLL.

9.2.3 Far-edge computing platform u.RECS

The architecture of the u.RECS is presented in Figure 9.4. The two integrated module slots support the SMARC 2.1 standard and the NVIDIA Jetson NX standard. In addition to the two module slots, a PCIe M.2 slot and an mPCIe slot are integrated, which can be used to add further accelerators or communication methods, such as 5G, to the u.RECS. Furthermore, communication options, e.g. Ethernet or PCIe and energy measurement methods, are integrated on the board to make the u.RECS a perfect fit for a wide range of AIoT use cases.

The NVIDIA Jetson NX module slot is capable of supporting Xavier NX and Orin NX SoC modules. These modules have ARM CPUs combined

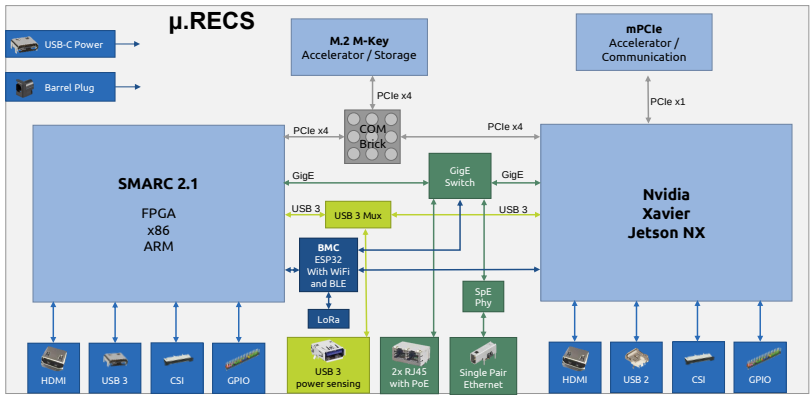

Figure 9.4 Architecture of u.RECS platform.

with latest NVIDIA GPU technology. Support for the SMARC 2.1 standard gives the u.RECS access to a wide range of COMs and ML accelerators, as SMARC modules are available in the market through different module manufacturers, such as Congatec, ADLINK Technology, or others. The SMARC slot can be equipped with, among others, the following types of microservers:

- ARM CPU (e.g., i.MX 8)
- x86 CPU (e.g., Atom CPU)
- FPGA (e.g., Xilinx Zynq UltraScale+)

There are a number of additional ML accelerators that can be equipped in or connected to an M.2 or mPCIe slot. Additionally, it is possible to connect accelerators via USB 3.0 and access them from one of the compute modules. Furthermore, with the u.RECS, it is possible to measure the energy of an accelerator connected via USB. Accelerators supported this way include:

- Intel Myriad X
- Hailo-8
- Google Coral

9.3 Accelerator Overview

There are many accelerators available for a wide range of applications, from small embedded systems with power budgets in the order of milliwatt to cloud platforms with a power consumption exceeding 400 W. Figure 9.5 provides an overview of the different accelerators using a double logarithmic plot, grouping them into three groups, depending on their peak performance values

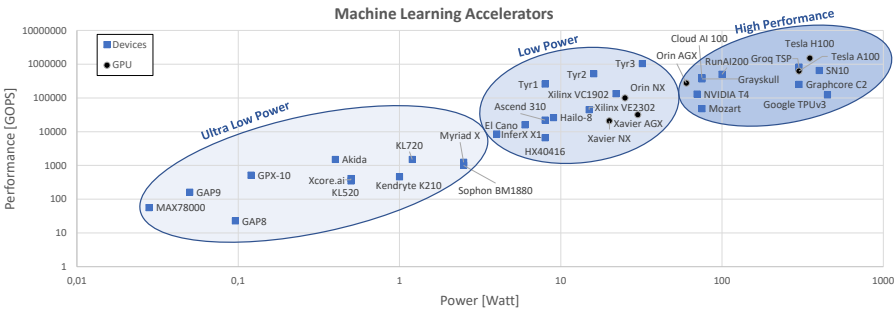

Figure 9.5 Theoretical performance of AI accelerators and classification into performance groups.

(in giga-operations per second). It should be noted that values provided by the vendors are used; so no normalization regarding technology, precision, or architecture is performed. On average, an energy efficiency of about 1 Tera Operation per W (1 TOPS/W) is achieved. In the following paragraphs, the main characteristics of the three performance groups are discussed.

Ultra-low power (<3 W): The ultra-low power group of accelerators is mainly devices integrating energy-efficient, microcontroller-style cores combined with compact accelerators for DL-specific functions. They are focusing on generic IoT applications like the Maxim MAX78000, the Ambient Scientific GPX-10, or the BrainChip Akida, providing only simple analog or digital interfaces. Other devices such as the Greenwave GAP 8 and GAP 9, the Canaan Kendryte K210, or the Kneron KL530 and KL720 also aim at vision processing, providing an additional camera interface. Typically, those devices are directly designed into the application itself without using a modular or microserver-based approach, simply because all interfaces and peripherals are integrated. Only the Bitmain Sophon BM1880 and Intel Myriad X are providing a generic USB interface and are designed to act as accelerator devices attached to a regular host processor. None of these devices integrates external memory controller interfaces. Based on its wide availability, the Intel Myriad X device is included in the benchmarking activity.

Low power (3−35 W): While the previous group of accelerators is focusing on applications with a very low-power envelope (often in a battery-powered environment with no special requirements regarding cooling), the low-power group of accelerators includes accelerators for a wide range of applications in automation and automotive. All devices include high-speed interfaces for external memories, and peripherals, as well as high-speed communication

toward other processing devices or host systems, such as PCIe, proving excellent capabilities for a modular, microserver-based approach as supported by the RECS platform. Apart from the Hailo-8, the FlexLogix InferX X1, and the VSORA Tyr family, which are designed as dedicated accelerators attached to an external host processor, all devices include powerful, general-purpose application processors, capable of running a fully fledged Linux operating system. In addition to specialized ASICs including the Coherent Logix HX40416, the Blaize El Cano, or the Huawei Ascend 310, this group also includes embedded GPUs from NVIDIA, in particular, the Jetson family, starting from the Nano and TX2, via the Xavier NX and Orin NX devices all the way up to the AGX Xavier. The Xilinx Versal Core AI VC1902 and Versal Edge AI VE2302 are explained in detail in the following section.

High performance (>35 W): The high-performance group of accelerators includes devices with up to 450 W of TDP, suitable for both inference and training use cases, typically deployed in the form of a PCIe extension cards for edge or cloud servers. Besides the classical NVIDIA Tesla GPGPUs including Tesla V100, A100, and H100, also dedicated ASICs like the Groq TSP, the SambaNova SN10, the Graphcore C2, or the Google TPUv3 are part of this cluster. In addition, also powerful inference ASICs like the SimpleMachines Mozart, the Tenstorrent Grayskull, the Qualcomm Cloud AI 100 Chip, or the Untether AI RunAI200 are included. As a reference, also a consumer-class NVIDIA Geforce GTX 1660 GPU has been included in the benchmarking. The NVIDIA Jetson AGX Orin is also part of this group due to its high power envelope, although it is part of the embedded NVIDIA Jetson family.

9.3.1 Reconfigurable accelerators

Field programmable gate arrays (FPGAs) are a promising alternative to GPUs and TPUs. Due to their reconfigurable architecture, these devices can be adapted to the specific requirements of an application, making them promising candidates for the resource-efficient processing of machine learning algorithms. For acceleration of deep learning models on their FPGAs, Xilinx provides a dedicated IP core, the deep-learning processor unit (DPU). Various FPGA devices are already available in the RECS system, and new devices like Xilinx Versal are expected to be added in the near future. For the easy yet efficient integration of new reconfigurable accelerators into the RECS system, an FPGA base design has been developed, supporting the flexible communication facilities of the RECS platform.

A key advantage of FPGAs over ASICs is their reconfigurability, enabling highly optimized designs for specific application scenarios. However, this reconfigurability comes at a significant overhead in terms of power and performance. This overhead is reduced by the integration of embedded processors and fixed-function units (like DSP blocks and embedded memories) in modern FPGAs. An additional method to increase the resource efficiency of reconfigurable architectures is partial dynamic reconfiguration, enabling, e.g., to switch between different accelerators at runtime. Dynamic reconfiguration can be used to enable the system to automatically adapt to changing environmental conditions, like weather changes, when running a neural network on camera data. In general, accelerators with different power, performance, and accuracy footprints can be selected at runtime.

Figure 9.6 provides an overview of the architecture and supported interfaces of the base design for the u.RECS. For heterogeneous systems, the PCIe interface connects the reconfigurable accelerator to other compute modules and accelerators on the u.RECS. The base design was created with the Xilinx Vitis Core Development Kit (2021.2) in the Vivado block design environment. When targeting different FPGAs or FPGA platforms, the base design needs to be adapted, e.g., because of changed internal or external interfaces. Additionally, other pre- or post-processing steps may be required, as well as a change of the complete application runtime. Hence, a wide variety of different FPGA implementations can be expected, which are difficult to

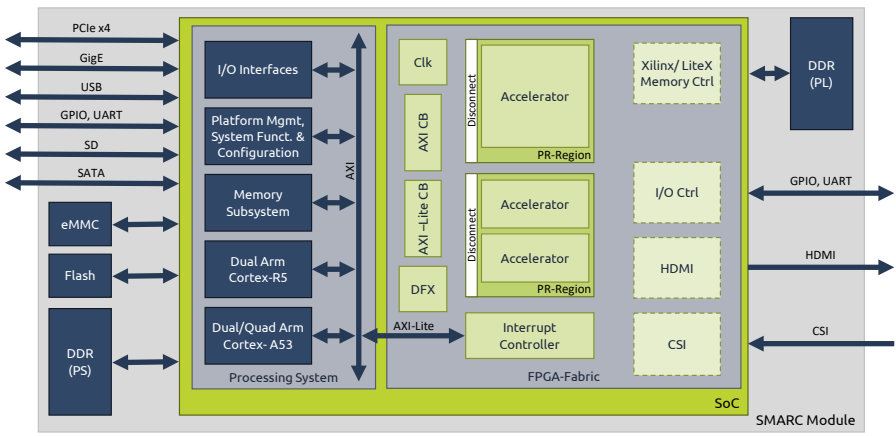

Figure 9.6 Block diagram of the FPGA base design supporting partial dynamic reconfiguration.

manage by hand. Therefore, we have set up a scripting environment that automates the configuration and build process. All necessary calls to the Vitis build system are automated, enabling an easy transition to new platforms. The entire hardware platform as well as the software infrastructure are built automatically, including the configuration of the processing system and the Linux environment. Changes to the FPGA base design, like additional interfaces, located in the FPGA fabric, can be done directly in the script. This is especially important for easy migration between the different FPGAs supported by the RECS platform.

For the evaluation of performance and energy efficiency, various combinations of Xilinx FPGAs and DPU configurations have been generated with the scripting approach described above. UltraScale+ FPGAs have been used, ranging from small (ZU3EG) to large (ZU15EG) devices. The DPUs can be parameterized to match the application requirements, e.g., by varying the inherent parallelism in terms of the peak number of operations per clock cycle. In the next section, FPGA implementations are named by the device and the integrated DPU variant. To give an example, ZU15 2xB4096 refers to a ZU15EG device that integrates the base architecture together with two B4096 DPUs, each capable of processing 4096 INT8 operations per clock cycle. The DPUs are running at a reduced clock frequency of 200 MHz, limited by power constraints of the used boards.

In addition to Xilinx UltraScale+ FPGAs, we have also evaluated the energy efficiency of the new Xilinx Versal architecture, utilizing a VC1902 on the VCK190 evaluation system. The reconfigurable SoCs combine an ARM processing system with a programmable logic fabric and a variety of I/O interfaces. In addition to the classical FPGA-based SoCs, the VC1902 integrates new DSP engines, AI engines, and a network-on-chip infrastructure for communication between the heterogeneous computing resources. For deep learning applications, especially the 400 AI engines are of high interest, promising a significant increase in performance and energy efficiency compared to DPU implementations on the reconfigurable fabric. For the development, Xilinx Vitis AI version 2.5 has been used together with Xilinx Vitis 2022.1. A wide range of configurations can also be selected for the Versal DPU. In our implementation, C32B6 refers to an architecture with six batch handlers, utilizing 32 AI engine cores per batch handler, for a total of 192 AI engines. The implementation runs at a clock frequency of 333 MHz for the programmable logic and 1.25 GHz for the AI engines.

9.4 Benchmarking and Evaluation

9.4.1 Methodology

The evaluation of different accelerators and their corresponding hardware manufacturer's optimization toolchains was conducted using a standard set of convolution neural network (CNN) models. The evaluation utilized three state-of-the-art CNNs − ResNet50 [4], MobileNetV3 [5], and YoloV4 [6] − all of which are from the domains of image recognition and classification. The models were represented using the open neural network exchange (ONNX) [7], which is an open standard for ML algorithms.

For evaluation purposes, two widely used benchmarking datasets were employed: common objects in context (COCO) [8], a comprehensive database for object detection, segmentation, and captioning, and ImageNet [9], the most frequently used dataset for image classification in the large-scale visual recognition challenge (ILSVRC). ImageNet contains 1000 object categories and has 1,281,167 training images, 50,000 validation images, and 100,000 test images. Three versions of each model (ResNet50, MobileNetV3, and YoloV4), each with a different precision, were evaluated. The first version was the original trained model with 32-bit floating-point precision (FP32), followed by two quantized versions of the original model: 16-bit floating-point precision (FP16) and 8-bit integer precision (INT8). The toolchains used for evaluation are summarized in Table 9.1.

In order to evaluate the merit of the hardware platforms for various deployment scenarios with different goals and constraints, we used the following metrics divided into four categories:

- System metrics: **peak performance** in giga-operations per second (GOPS) and **idle power**[1] in Watts (W).

[1] The idle power is measured as to determine a more accurate power consumption for the execution.

Table 9.1 Toolchains used for evaluation.

Hardware	Toolchain	Version
NVIDIA GPUs	TensorRT SDK	7.1.3 and 8.0.1 [10]
Intel CPUs, Myriad	OpenVINO	2021.4.1 [11]
Xilinx FPGAs	Vitis AIVitis	1.3 and 2.5 (Versal)2021.2 and 2022.1 (Versal)
Google Coral TPU	TensorFlow [12]TensorFlow Lite	2.4 and 2.52.4 and 2.5
Hailo-8	Hailo Software Suite	4.8.1

- Performance metrics: **inference time** in seconds (s), **achieved performance** in GOPS, and **power consumption** in Watt (W).
- Quality metrics: **accuracy** in percentage (%) and **mean average precision** (mAP or mAP@X) in percentage (%).
- Efficiency metrics: **power efficiency** in GOPS per Watt (GOPS/W).

In this evaluation, two quality metrics were evaluated, each suited for the targeted CNN domain. For image classification, the most crucial quality metric is accuracy, which represents the number of correct classifications divided by the number of images. Accuracy was measured in two ways: top-1 that accuracy measures the frequency of the model prediction with the highest probability matching the ground truth; and top-5 accuracy that measures if the top 5 highest-probability predictions include the ground truth. For object detection, the relevant quality metric is mean average precision (mAP or mAP@X). mAP@X is the area under the precision−recall curve with an intersection over union (IoU) threshold X. For instance, mAP(.50) means that a positive detection must have a minimum IoU of 50%, with everything below being marked as a false detection with a precision of 0%. Another form of mAP is mAP@X:Y, calculated as the average AP over a range of minimum IoUs. We reported the mAP@X:Y from $X = 0.5$ to $Y = 0.95$, with a step size of 0.05.

To determine the power consumption, we utilized tools provided by the hardware vendors, and when these were not available, we used laboratory instruments. For the NVIDIA accelerators, we employed the utilities **Tegrastats** and **nvidia-smi**. The NVIDIA Jetson-Nano was an exception, where, due to the absence of integrated tools, we used an external power meter. The Intel Myriad and its host module were measured using a Tektronix MDO4054B oscilloscope. The Google Coral TPU and its host module were also measured with the same oscilloscope. The power consumption of Hailo-8 was measured inside an NVIDIA Xavier NX evaluation system by plugging it into the M.2 PCIe port and excluding the power consumed by the CPU module. For FPGA-based systems, the complete system power, including external memory and I/O interfaces, was measured. Notice that the power consumption values are also necessary to determine the efficiency metric (typically measured in GOPS/W).

It is important to mention that, due to the limited space, only the evaluation results for the YoloV4 model are presented in this chapter. However, the conclusions in this chapter are still relevant to the results of all other tested models.

9.4.2 Evaluation results

As mentioned before, the optimization toolchains for the evaluated accelerators are vendor-specific and vary between architectures. Despite using the same source for the DL models, we needed to ensure that all devices were performing the same computations and produce comparable results. To validate this, we measured the mAP(.50) and mAP(.50:.95) for each device. Our findings show that the mAP is significantly influenced by the software toolchain used to compile and quantize the models. Therefore, the mAP was grouped into categories based on vendor and quantization (FP32, INT8), as depicted in Figure 9.7.

The NVIDIA FP32 category encompasses all results obtained from NVIDIA devices that used 32-bit floating point (FP32) quantization. The OpenVINO FP32 category combines the results from x86-based processors and the Myriad DL accelerator that employed FP32 quantization.

Furthermore, tests were also conducted using FP16 quantization, but since they only show minor deviations from FP32 ($<0.1\%$), only FP32 and INT8 results are presented here. For the NVIDIA INT8 category, which encompasses all NVIDIA devices using 8-bit integer quantization, the quantization was done using training data from the COCO dataset with the toolchain. The Xilinx INT8 and Hailo-8 INT8 categories were based on pre-quantized models from each vendor's model zoo. Our attempts to quantize the YoloV4 model for these categories resulted in poor precision outcomes. This highlights the significant impact that specific toolchains and hardware expertise can have on quantization and precision.

Figure 9.7 compares the mAP of all tested architectures with the YoloV4 model. Most of the architectures show slight deviations of less than 5%, with the exception of the Xilinx INT8 result, which is nearly 8% lower. Further analysis was conducted by examining the recall−precision gradients for each of the 80 classes the YoloV4 model is trained on. Figure 9.8 presents an example of this analysis, showing the mAP(.50) recall−precision gradients, where objects with an IoU larger than 50% are considered positive detections and are displayed with their corresponding precision. Objects with an IoU less than 50% are considered negative detections and are set to a precision of 0%, which is why the orange and yellow precisions are not present in the figure. Class I (toothbrush) showed the highest deviation for INT8 quantization among the tested devices, with the NVIDIA and Xilinx accelerators performing relatively poorly compared to the Hailo-8 accelerator. This is by far the class with the highest deviation, unlike class II (vase), where all

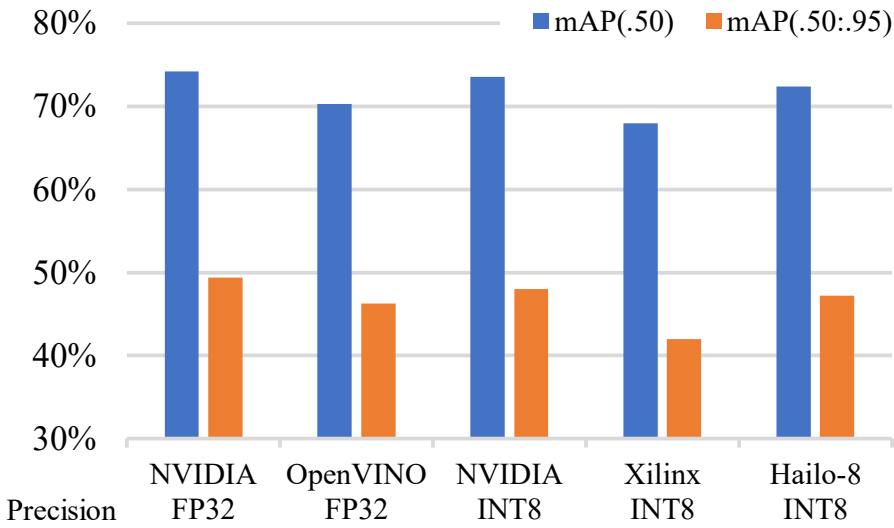

Figure 9.7 Accuracy evaluation of YoloV4.

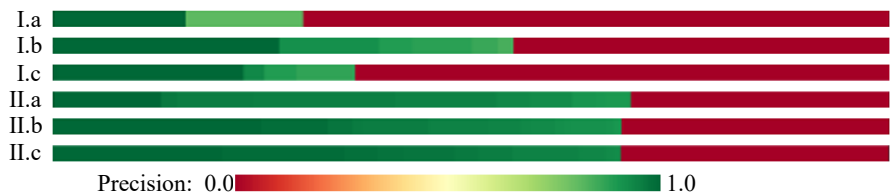

Figure 9.8 mAP(.50) recall−precision gradients using INT8 for classes I: toothbrush II: vase and accelerators. (a) NVIDIA; (b) Hailo-8 c: Xilinx.

accelerators performed similarly. A detailed analysis of the results, including all 80 COCO classes for each accelerator with floating point and integer quantization, showed that most classes behave like class II. This provides confidence that the accelerators in the evaluation are performing the same tasks and that the results are comparable.

The evaluation in Figure 9.9 shows the achieved performance in GOPS and the power consumption in Watt (W) for the execution of YoloV4 on the different hardware systems. Similar results are obtained for both ResNet50 and MobileNetV3. The notations next to the accelerators (B1, B4, and B8) indicate batch sizes of 1, 4, and 8. For those cases, the metrics are for the complete execution of the batch. It is important to note that the power consumption of all PCIe-based accelerators (Myriad, GTX1660, V100, and

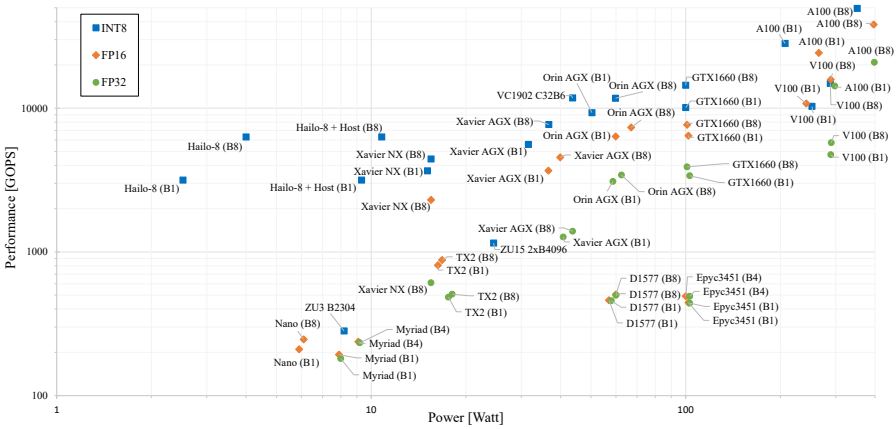

Figure 9.9 Performance evaluation of YoloV4.

A100) has been measured without the host system. For Hailo-8, both cases (with and without the host system) are considered.

Figure 9.9 serves as a reference for making hardware choices based on performance and power requirements. The results can be applied to a variety of use cases by taking into consideration the power domains depicted in Figure 9.5.

Two x86 systems (D1577 and Epyc3451) are provided as a reference to demonstrate the superiority of DL accelerators over traditional processing systems. In terms of energy efficiency, noteworthy platforms include Hailo-8, Xavier NX, Xavier AGX, VC1902, Orin AGX, and A100, catering to different domains, as shown in Figure 9.5.

In this evaluation, three reconfigurable devices (ZU3, ZU15, and VC1902) have also been studied. On the one hand, the Xilinx Zynq devices (ZU3 and ZU15) exhibit relatively low performance compared to the specialized accelerators, as they are basic FPGAs that utilize the Xilinx DPU accelerator. On the other hand, the Xilinx Versal (VC1902) boasts significantly higher performance and energy efficiency due to its built-in DL accelerators. Among all reconfigurable devices, the VC1902 shows the best energy efficiency with INT8 quantization.

The energy efficiency comparison in Figure 9.10 reveals a clear gap between classical processing systems (D1577 and Epyc3541) and DL accelerators. Even older DL accelerators (TX2, Nano, and Myriad) offer better efficiency. Newer GPU-based accelerators (Xavier NX, Xavier AGX, and Orin AGX) provide good efficiency but are obviously surpassed by dedicated

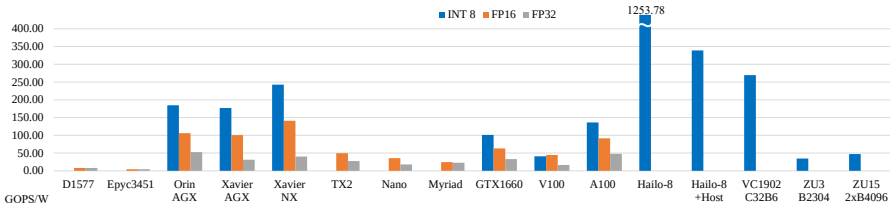

Figure 9.10 Efficiency evaluation of YoloV4. The missing bars for certain platforms represent cases where the precision is not supported.

ASIC-based accelerators (Hailo-8 and VC1902). It is important to note that the power measurement of all PCIe-based accelerators was reported without the power for the host system. The Hailo-8 presents a significant lead when compared to Xavier NX and VC1902.

Overall, this evaluation shows that, when considering the different points in the compute continuum, as presented in Figure 9.9, the Hailo-8 and Xavier NX are well-suited for far-edge computing platforms, while Xavier AGX, VC1902, and Orin AGX fit into near-edge computing platforms, and the A100 can be deployed in cloud computing platforms.

9.5 Conclusion

The main topic of this chapter is the evaluation of heterogeneous AIoT hardware, in particular, accelerators, for deep learning applications. In addition, the RECS hardware platforms are introduced, supporting the complete continuum of heterogeneous cloud, edge, and IoT applications. Especially for scenarios with low power budgets, energy efficiency is crucial, which is only achieved by using specialized hardware accelerators. A set of relevant accelerators was presented and classified into three different performance groups according to their processing capabilities. Besides ASIC- and GPU-based accelerators, emphasis has been put on reconfigurable architectures, presenting a DPU-based FPGA architecture for easy integration of dedicated DL algorithms.

The evaluation methodology was described in detail, discussing the used DL models, corresponding datasets, and used specific toolchains. The performance and efficiency metrics GOPS and GOPS/W were introduced and the quality metrics mAP(0.50) and mAP(0.50:0.95) were used for YoloV4. The power measurement used for this evaluation was described.

Since toolchains are vendor-specific, an evaluation of the accuracy, of the model running on different architectures, was performed. An in-depth analysis of recall−precision gradients per class shows that the results of different architectures using different toolchains are still comparable. The YoloV4 evaluation shows an extensive overview of modern DL accelerators and their performance as well as their energy efficiency. The outcome of this chapter provides a guideline for hardware selection in the area of DL accelerator, ranging from far-edge computing up to cloud computing.

Acknowledgements

This publication incorporates results from the VEDLIoT project, which received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No. 957197.

References

[1] Martin Kaiser, Rene Griessl, Nils Kucza, et al. VEDLIoT: Very Efficient Deep Learning in IoT. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 963-968, 2022.

[2] Rene Griessl, Karol Gugala, Elaheh Malekzadeh, et al. D 3.1 – Evaluation of existing architectures and compilers for DL, October 2021. VEDLIoT project deliverable.

[3] Rene Griessl, Marco Tassemeier, Pedro Trancoso, Karol Gugala, et al. D 3.3 – Evaluation of the DL accelerator designs, October 2022. VEDLIoT project deliverable.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770-778, 2016.

[5] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314-1324, 2019.

[6] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YoloV4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[7] Junjie Bai, Fang Lu, Ke Zhang, et al. ONNX: Open Neural Network Exchange. https://github.com/onnx/onnx,2019.

[8] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. *CoRR*, abs/1405.0312, 2014.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248-255. Ieee, 2009.

[10] Huang Rao, Chen et al. TensorRT. https://github.com/NVIDIA/TensorRT,2013.

[11] Paramuzov Lavrenov, Churaev et al. OpenVINO. https://github.com/openvinotoolkit/openvino,2013.

[12] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265-283, 2016.