TULIPP

H2020-ICT-04-2015

Grant Agreement n° 688403

D2.2: Optimized instance of a power-efficient board that follows the design rules

Authors

Organization	Participant
SUNDANCE	Flemming Christensen
IOSB	Igor Tchouchenkov

Reviewers

Organization	Participant
THL	Philippe Millet
NTNU	Magnus Jahre
SUN	Timoteo García Bertoa

Document Description

Deliverable number	D2.2
Deliverable title	Optimized instance of a power-efficient board that follows the design rules
Work Package	WP2
Deliverable nature	Report
Dissemination level	Public
Contractual delivery date	2018-11-30
Actual delivery	2019-03-04
Version	1.3

	Written by	**Approved by**
Name Signature	Igor Tchouchenkov	

Version history

Version	Date	Description
1.0	2019-01-13	Created
1.1	2019-01-30	Re-worked
1.2	2019-02-15	Reviewed
1.3	2019-03-01	Second review

Executive Summary

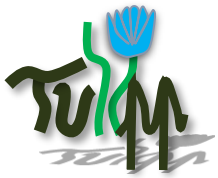
This deliverable describes the optimized instance of the TULIPP Starter Kit that follows the design rules developed in WP1 and WP2. The HW – especially when using in TULIPP platform – is more power-efficient than the first version, faster and supports more interfaces and features important for image processing applications.

The delay of approximately two months by the preparation of this deliverable is due to the document only because of justifications with D1.3. The optimized TULIPP boards were delivered to the consortium in time before M34.

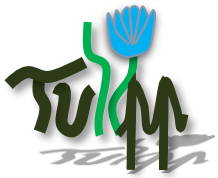
In addition to a description of the board and its benefits, a short tutorial is provided to help the reader get started with using the Starter Kit – the board together with other key TULIPP components (STHEM, real-time operating system and libraries).

Table of Contents

List of Figures



List of Tables



Acronyms

API Application Programming Interface

APU Application Processing Unit

CAN Controller Area Network

CPI Camera Parallel Interface

CPU Central Processing Unit

DSI Display Serial Interface

DVFS Dynamic Voltage and Frequency Scaling

DPM Dynamic Power Management

ESM Embedded System Module

FinFET Fin Field-Effect Transistor

FPGA Field Programmable Gate Array

GHHP Generic Heterogeneous Hardware Platform

GigE Gigabit Ethernet

GPU Graphics Processing Unit

HDMI High-Definition Multimedia Interface

I/O Input/Output

IP Intellectual Property

JTAG Joint Test Action Group

MPSoC Multi-Processor System on Chip

ONFI Open NAND Flash Interface

OS Operating System

PCB Printed Circuit Board

PCI Peripheral Component Interconnect

PL Programmable Logic

PMU Platform Management Unit

PS Processing System

RPU Real-Time Processing Unit

RTOS Real-Time Operating System

SMMU System Memory-Management Unit

SoC System on Chip

SoM System on Module

STHEM Supporting uTilities for Heterogeneous EMbedded image processing platforms

SWD Serial Wire Debug

TRP TULIPP Reference Platform

TSK TULIPP Starter Kit

UART Universal Asynchronous Receiver-Transmitter

UAV Unmanned Aerial Vehicle

USB Universal Serial Bus

1 INTRODUCTION

The hardware platform was built from the recommendations issued in WP1 deliverables D1.1, D1.2 and D1.3. The recommendations were used during the selection process of the processing unit and the interfaces and showed that the Zynq platform is a good choice for the implementation of the Tulipp hardware platform.

The initial choice of Xilinx Zynq 7030 SoC [1] for TULIPP reference platform instantiation and board construction was based on prior background research and experience from other projects (e.g. EMC2 Artemis [2]). This first instantiation, based on EMC2-Z7030 with dual-core ARM on PC/104 board [3], was delivered to all partners needing it and tested with several applications (TULIPP Use Cases). In parallel, the selection of suitable components and the development of new instantiation boards was started in close cooperation with WP1. The analysis and the selection of most suitable components for the second instantiation are well documented in the Deliverable D2.1. The board development for the selected Xilinx Zynq Ultrascale+ ZU4EV MPSoC was based on the latest technologies, the experience from the usage of the first instantiation as well as the principles and rules from WP1 (described in its Deliverables D1.2 and D1.3).

The new board is a part of TULIPP Starter Kit (see Fig. 1) whose design exploits the concept of low-power heterogeneous image processing systems providing a complete platform (containing also real-time OS, different development tools and libraries) useful for applications such as medical x-ray, embedded image processing for UAVs or automotive applications.

The slides used during the Tutorial [4] of the TULIPP platform, with the TULIPP Starter Kit, is attached to this deliverable. The EMC2-DP Starter's Guide (https://www.sundance.technology/wp-content/uploads/2019/03/EMC2-DP-Starters-Guide-v3.0-QCF32.pdf) can be used as additional source to better understand the hardware platform.

Figure 1: TULIPP Starter Kit with LynSyn

2 WAY TO THE DEVELOPMENT OF TULIPP BOARDS

Embedded solutions are always fighting with small or highly constrained volume, weight and reduced power consumption or heat dissipation. This is referred as SWaP (Size, Weight and Power) constraints or SWaPC when adding the cost constraint that often comes with these small, ubiquitous and pervasive architectures to which we are often exposed without noticing (smart watch, smart phones, video surveillance cameras, IoT sensors, car sensors, …).

These constraints have drastic impact on the computer architecture. The whole computing solution being limited, including the processor capabilities, the memories sizes (internal and external RAMs and caches). The storage capabilities are also limited; there is often no hard drive and the system can often only use limited flash memories.

An embedded system interacting with its environment is called a Cyber-Physical-System (CPS). When a CPS is dealing with a sensor that provides signal at a dedicated rate, the CPS has to work at the same pace. This requires the CPS to work at real-time. The real-time requirement becomes even stronger when the CPS deals with critical systems, such as pedestrian detection in cars or works within a system with regulation (avionic or medical systems). Since our use cases are dealing with such systems our platform needs to be able to handle real-time computation.

2.1 Requirements for Functions of Hardware for Image-Processing Platforms

An embedded image-processing platform must:

1) read images from one or several camera-like sensors
2) process the images through a dedicated algorithm(s)
3) output an information extracted from the (processed) image
    a) as “high-level” information to produce actions
    b) as “enhanced” image
4) support applications development (compiling, deployment, debugging etc.)

The common structure of an image processing system is shown in Fig. 2.

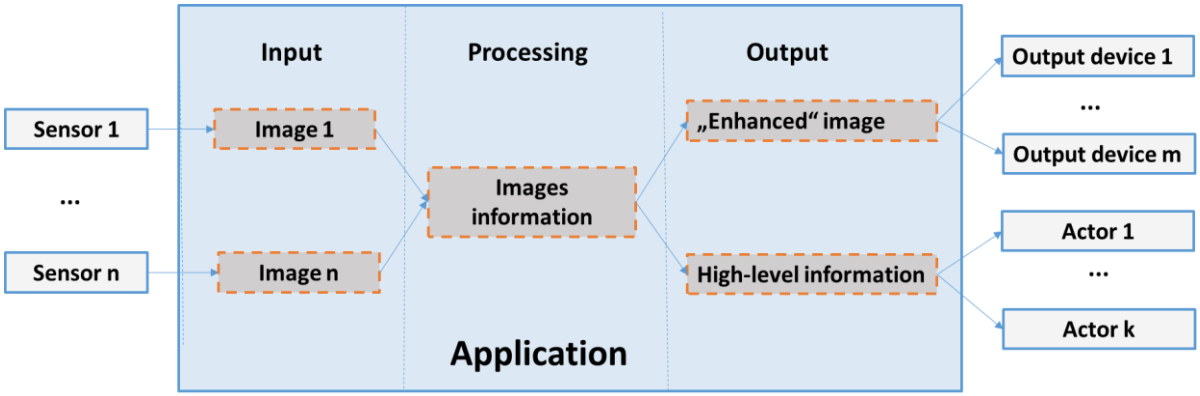
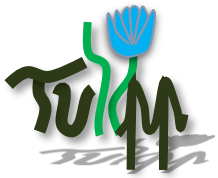


Figure 2: Common Structure of an Image Processing System



The TULIPP hardware platform is energy-efficient and able to support all three functions at the sensor rate without losing information. Only if the result of the computation is not linked to any safety issue, it might be allowed to lose some of the frames, but for the information to have any meaning, the system must be designed to not lose any frame – i.e., be "real-time".

Only systems that are user-friendly enough for handling and programming can become popular. For example, for application development debug interfaces are needed. A hardware that would not include any debug interface would not be easy to program and would be unlikely to become popular. One way to be widely used is to be open and to use standards whenever possible. The corollary is that when a technology is widely used it becomes the standard. The TULIPP hardware platform is open and uses standards. Thanks to the ecosystem we built in the project, we hope it will become widely used.

In the next part, we analyse these requirements briefly from the perspective of hardware implementation. The requirements were analysed in different TULIPP Reports and Deliverables (e.g., D1.3) more concretely from different perspectives.

2.1.1 Reading images from one or several camera-like sensors

There are a lot of different interfaces and standards to connect camera-like sensors to processing boards (they were described in D1.1 and D2.1). A "nice" hardware platform must ensure that all vision interfaces that users could need are or might be implemented. The challenge by the design is to decide which interfaces must be implemented "standardly" and which could be implemented with additional components if needed. The problem here: each interface consumes resources (volume, weight and energy) so that too many interfaces make the system bigger and more energy-hungry.

According to specifications from WP1, the most suitable visual input interfaces for TULIPP are GigE, CPI, USB3 and CameraLink. The interfaces are described in Deliverables D1.2 and D1.3 of WP1.

2.1.2 Processing the images through a dedicated algorithm(s)

For real-time image processing very different algorithms can be used, so there is no chip or SoC optimal in all the parameters for all the use cases. The FPGA acceleration is a good compromise as it can optimize in hardware a wide set of functions. SoCs with graphical accelerators (like NVidia Tegra SoC or Myriad VPU) could also be a good fit for application with functions that could not benefit from the FPGA acceleration.

Adding processing devices also get the solution to a higher cost. Including additional resources also adds internal and external interfaces. Even when idled resources also consume energy. The "dark silicon" concept that we know at the integrated circuit level, is also a reality at the board level when all the components cannot be switched on to keep the TDP at the required level. So, the components must not be "over dimensioned".

The hardware designer must find a trade-off between maximum computing performance and minimum energy usage. Higher performance will lead to higher energy consumption that will limit the lifetime of the system when on batteries. Lower energy usage will allow longer lifetime but will also delivering less capabilities to the application developers.

2.1.3 Output an information extracted from the image

For output an information as original or "enhanced" image there are two possibilities: Direct visualisation on a display over a (standard) interface like HDMI, DVI etc.; or transmission to visualisation over another interface, like Ethernet.

For the output of high-level information (detected objects, features, action commands etc.) different interfaces could be used depending on the throughput needed: Ethernet, USB, CAN, RS232 etc. As already written in 2.1.1, each interface consumes resources so that only essential interfaces shall be implemented.

According to specifications from WP1, the most suitable video output interface for TULIPP is HDMI. The interface is described in deliverable D1.3 of WP1.

2.1.4 Programming and Handling

A support of all possible interfaces, high processing speed and energy efficiency is meaningless if the hardware is difficult to program or to use. Especially the transfer of legacy software from older implementations and the support of "standard" image processing libraries (like OpenCV, OpenCL or OpenVX) are essential. Of course, the full application development cycle must be well supported too including deployment and debugging. The hardware must have a suitable debug interface, and additional test possibilities (like measurement of time and energy consumption of program parts) are highly appropriate.

There are only a few interfaces dedicated to debug. JTAG and SWD interfaces are included in WP1 specifications for TULIPP platform.

Convenient handling refers to dimensions, working temperatures/cooling, shock and vibration resistance, positioning of interfaces, additional modules, power supply etc.

2.2 Hardware specification for TULIPP Platform instantiation

The system specification for TULIPP use cases (Medical, Automotive and UAV) is based on the analysis of use cases requirements which was made in D2.1. This specification was used as a base for TULIPP board development and was derived from Tables 1 and 2:

	Sizes, mm	Weight, g	Interfaces		Resolution		Power Consumption	Progr. Language
			Input	Output	minimal	optimal		
Medical Imaging			1x PCIe 2.0 or 1x GigE	1x GigE	1024x1024	1344x1344	< 25 W, better 10W	C/C++, CUDA, OpenCL
Automotive			1x Camera Link	Ethernet…	640x480	1024x512	Few watts	C/C++, CUDA, OpenCL
UAV	120 x 120	< 300	2x Camera Link	USB,CAN, Ethernet	376x240	640x480	< 25 W, better 5W	C/C++, CUDA, OpenCL, OpenMP

Table 1: Key TULIPP Use Cases requirements

	Input, MBits/sec		Output, MBits/sec		Latency, msecs
	minimal	preferred	minimal	preferred	
Medical Imaging	420 (2 bytes/pixel)	870	900	940	< 170
Automotive	222 (3 bytes/pixel)	378	<1 for control 250 for video	<1 for control 400 for video	< 150
UAV	7 (1 byte/pixel)	73	<1 for control 8 for video	<1 for control 80 for video	< 100 (optimal 10)

Table 2: TULIPP Use Cases requirements to I/O speed and latency

These additional requirements must also be taken into account:

- Scalability (at least one possibility):
  - Usage of different components of the same type/family (e.g. Zynq Ultrascale+) with suitable computing power/energy consumption
  - Usage of different component types
  - Connection of many boards (via PCIe >= 2.0)
- Physical compatibility
  - Work temperatures: –40 - +125 or better
  - Small sizes and weight
  - Shock and vibration resistant (especially UAV and automotive)

The specification contains many more or less "hard" requirements, but it does not select the hardware components definitely in all cases. In our case, additional parameters were analysed to take a decision: costs, availability, reliability, support, additional interfaces, etc.

The common TULIPP hardware platform structure (developed in WP1) is shown in the Figure 3. It is a *Generic Heterogeneous Hardware Platform (GHHP)* described in D1.3 containing a collection of computing substrates (i.e., CPUs, DSPs, GPUs and an FPGA fabric) with different performance as well as an interconnection network and input/output devices. The GHHP also contains on-chip memory in the form of scratchpads or caches, but this is not shown in the figure as platforms vary widely.

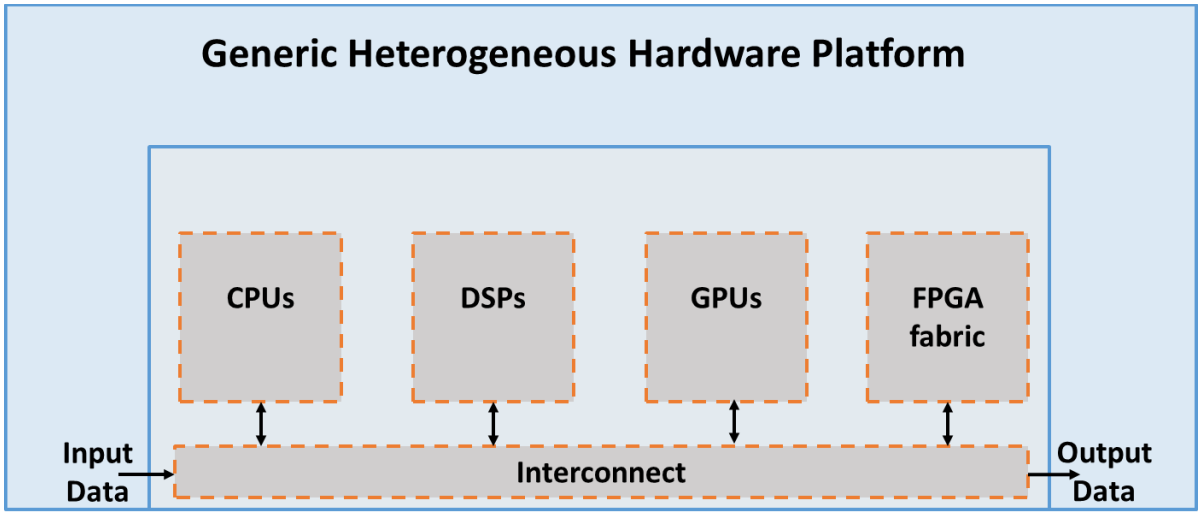


Figure 3: Generic Heterogeneous Hardware Platform

However, we must ensure that the OS, drivers, tool-chain and libraries are compatible with the selected hardware platform — as no user code can be ported without these components. To support complex algorithms and multiple video sources in parallel, the solution must support connection of multiple MPSoCs (or boards). HW selection was described more detailed in D2.1 and D1.3.

*The most heterogeneous component on the market today containing CPUs, GPUs and FPGA is the **Xilinx Zynq Ultrascale+*** [4]. Note however that the GPU can only be used for displaying.

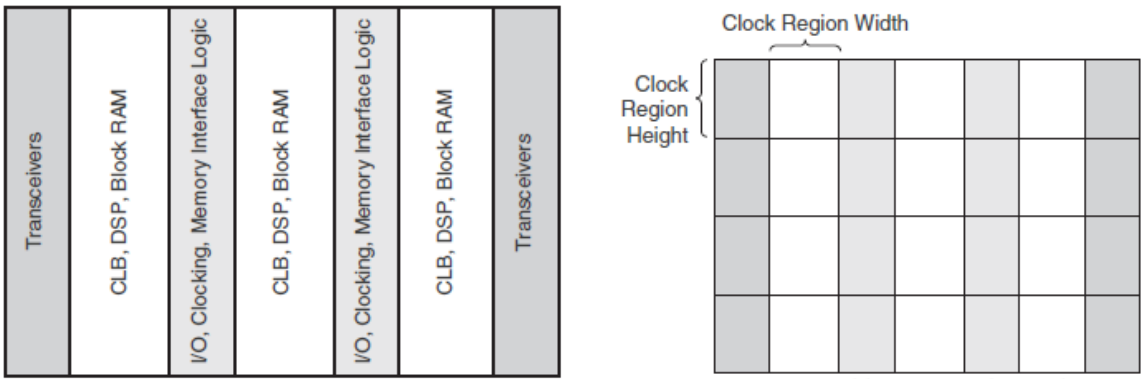
Since TULIPP use cases need different I/Os and have several productivity requirements, we chose to develop a carrier board for the TULIPP platform based on the selected device. An additional benefit of this choice is the scalability and possibility to move to more powerful computing modules when these become available. This choice is a direct impact from D1.2 and D1.3 that would probably not have been made without the work carried out in WP1. Technical solutions and steps needed for developing of such boards are discussed in the next section.

2.3 Zynq UltraScale+ Family: Key Features and Parameters

Xilinx's 16nm Zynq UltraScale+™ families are based on the all programmable architecture to span multiple nodes from planar through FinFET technologies and beyond. The families combine new memory, 3D-on-3D, and multi-processing SoC (MPSoC) technologies.

The Zynq UltraScale+ MPSoC family consists of a SoC style integrated processing system (PS) and a Programmable Logic (PL) unit. The PS comprises an Application Processing Unit (APU), a Real-Time Processing Unit (RPU), a GPU and many peripherals for connecting the multiple processing engines to external components [5].

Zynq UltraScale+ devices are arranged in a column-and-grid layout. Columns of resources are combined in different ratios to provide the optimum capability for the device density, target market or application, and device cost. At the core of Zynq UltraScale+ MPSoCs is the processing system that displaces some of the full or partial columns of programmable logic resources. Figure 4 shows a device-level view with resources grouped together [6]. For simplicity, certain resources such as the processing system, integrated blocks for PCIe, configuration logic, and System Monitor are not shown.



a. FPGA with Columnar Resources b. FPGA divided into Clock Regions

Figure 4: Zynq UltraScale FPGA with Columnar Resources and clock regions

Resources within the device are divided into segmented clock regions. The height of a clock region is 60 CLBs. A bank of 52 I/Os, 24 DSP slices, 12 block RAMs, or 4 transceiver channels also matches the height of a clock region. The width of a clock region is essentially the same in all cases, regardless of device size or the mix of resources in the region, enabling repeatable timing results. Each segmented clock region contains vertical and horizontal clock routing that span its full height and width. These horizontal and vertical clock routes can be segmented at the clock region boundary to provide a flexible, high-performance, low-power clock distribution architecture.

The structure of the device is shown in Figure 5.

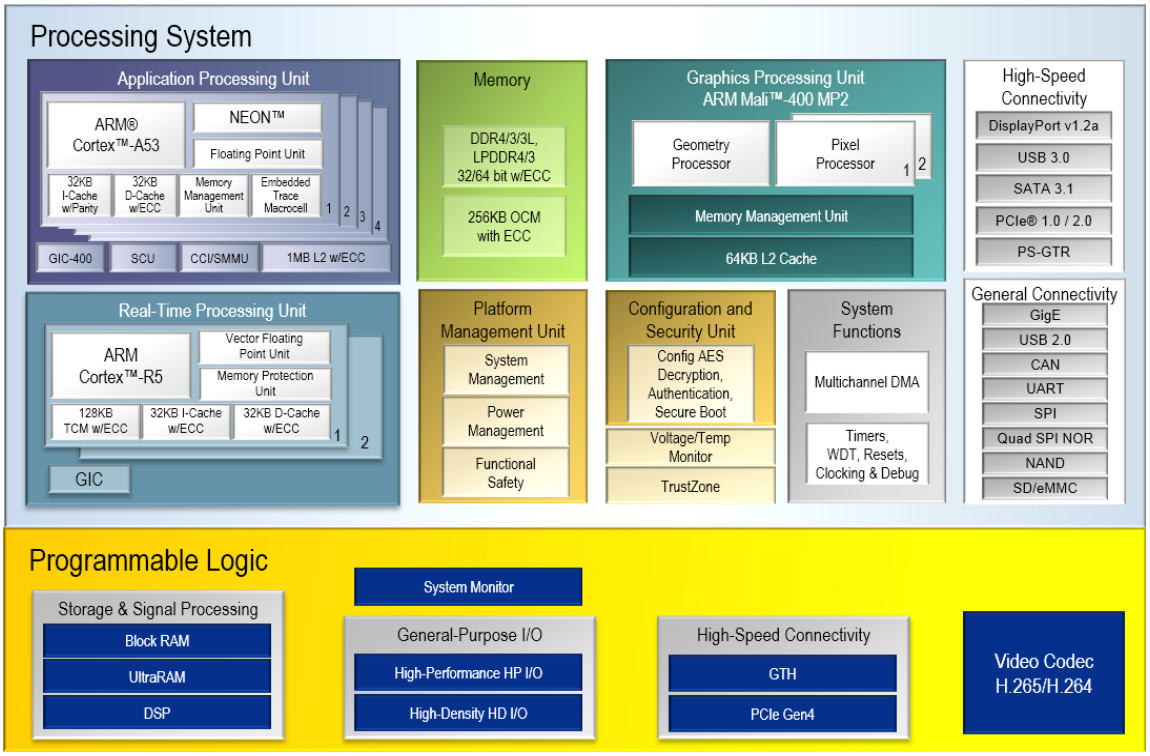


Figure 5: MPSoC Xilinx Zynq UltraScale+ [5]

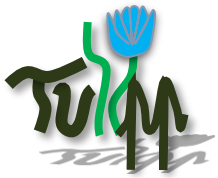
A short description of main components, features and interfaces Zynq UltraScale+ MPSoC is given below.

Application Processing Unit (APU)

The APU has a feature-rich Quad-core ARM Cortex-A53 processor [7]. Cortex-A53 cores are 32-bit/64-bit application processors based on ARM-v8A architecture, offering the best performance-to-power ratio and hardware virtualization. Each of the Cortex-A53 cores has:

- 32KB of instruction and data L1 caches, with parity and ECC protection respectively;
- NEON SIMD engine;
- single and double precision floating point unit

In addition to these blocks, the APU consists of a snoop control unit and a 1MB L2 cache with ECC protection to enhance system-level performance. The snoop control unit keeps the L1 caches coherent thus eliminating the need of spending software bandwidth for coherency. The APU also has a built-in



interrupt controller supporting virtual interrupts. The APU communicates to the rest of the PS through 128-bit AXI coherent extension (ACE) port via Cache Coherent Interconnect (CCI) block, using the System Memory Management Unit (SMMU). The APU is also connected to the PL, through the 128-bit accelerator coherency port (ACP), providing a low latency coherent port for accelerators in the PL. To support real-time debug and trace, each core also has an Embedded Trace Macrocell (ETM) that communicates with the ARM CoreSight Debug System.

Real-Time Processing Unit (RPU)

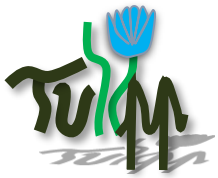
The RPU in the PS contains a dual-core ARM Cortex-R5 PS. Cortex-R5 cores are 32-bit real-time processor cores based on ARM-v7R architecture. Each of the Cortex-R5 cores has 32KB of L1 instruction and data cache with ECC protection. In addition to the L1 caches, each of the Cortex-R5 cores also has a 128KB Tightly Coupled Memory (TCM) interface for real-time single cycle access. The RPU also has a dedicated interrupt controller. The RPU can operate in either split or lock-step mode. In split mode, both processors run independently of each other. In lock-step mode, they run in parallel with each other, with integrated comparator logic, and the TCMs are used as 256KB unified memory. The RPU communicates with the rest of the PS via the 128-bit AXI-4 ports connected to the low power domain switch. It also communicates directly with the PL through 128-bit low latency AXI-4 ports. To support real-time debug and trace each core also has an ETM that communicates with the ARM CoreSight.

Graphics Processing Unit (GPU)

The dedicated ARM Mali-400 MP2 GPU [8] in the PS supports 2D and 3D graphics acceleration up to 1080p resolution. The Mali-400 supports OpenGL ES 1.1 and 2.0 for 3D graphics and Open VG 1.1 standards for 2D vector graphics. It has a geometry processor (GP) and 2 pixel processors to perform tile rendering operations in parallel. It has dedicated Memory management units for GP and pixel processors, which supports 4 KB page size. The GPU also has 64KB L2 read-only cache. It supports 4x and 16x Full scene Anti-Aliasing (FSAA). It is fully autonomous, enabling maximum parallelization between APU and GPU. It has built-in hardware texture decompression, allowing the texture to remain compressed (in ETC format) in graphics hardware and decompress the required samples on the fly. The GPU also supports efficient alpha blending of multiple layers in hardware without additional bandwidth consumption. It has a pixel fill rate of 2Mpixel/sec/MHz and a triangle rate of 0.1Mvertex/sec/MHz. The GPU supports extensive texture format for RGBA 8888, 565, and 1556 in Mono 8, 16, and YUV formats. For power sensitive applications, the GPU supports clock and power gating for each GP, pixel processors, and L2-cache. During power gating, GPU does not consume any static or dynamic power; during clock gating, it only consumes static power.

External Memory

The PS can interface many types of external memories through dedicated memory controllers. The dynamic memory controller supports DDR3, DDR3L, DDR4, LPDDR3, and LPDDR4 memories. The multi-protocol DDR memory controller can be configured to access a 2GB address space in 32-bit addressing mode and up to 32GB in 64-bit addressing mode using a single or dual rank configuration of 8-bit, 16-bit, or 32-bit DRAM memories. Both 32-bit and 64-bit bus access modes are protected by ECC using extra bits.



The SD/eMMC controller supports 1 and 4 bit data interfaces at low, default, high-speed, and ultra-high-speed (UHS) clock rates. This controller also supports 1-, 4-, or 8-bit-wide eMMC interfaces that are compliant to the eMMC 4.51 specification. eMMC is one of the primary boot and configuration modes for Zynq UltraScale+ MPSoCs and supports boot from managed NAND devices. The controller has a built-in DMA for enhanced performance.

The Quad-SPI controller is one of the primary boot and configuration devices. It supports 4-byte and 3-byte addressing modes. In both addressing modes, single, dual-stacked, and dual-parallel configurations are supported. Single mode supports a quad serial NOR flash memory, while in double stacked and double parallel modes it supports two quad serial NOR flash memories.

The NAND controller is based on ONFI3.1 specification [9]. It has an 8-pin interface and provides 200Mb/s of bandwidth in synchronous mode. It supports 24 bits of ECC thus enabling support for SLC NAND memories. It has two chip-selects to support deeper memory and a built-in DMA for enhanced performance.

General Connectivity

There are many peripherals in the PS for connecting to external devices over industry standard protocols, including CAN2.0B, USB, Ethernet, I2C, and UART.

USB 3.0/2.0

The pair of USB controllers can be configured as host, device, or On-The-Go (OTG). The core is compliant to USB 3.0 specification and supports super, high, full, and low speed modes in all configurations. In host mode, the USB controller is compliant with the Intel XHCI specification. In device mode, it supports up to 12 end points. While operating in USB 3.0 mode, the controller uses the serial transceiver and operates up to 5.0 Gb/s. In USB 2.0 mode, the Universal Low Peripheral Interface (ULPI) is used to connect the controller to an external PHY operating up to 480Mb/s. The ULPI is also connected in USB 3.0 mode to support high-speed operations.

Ethernet MAC

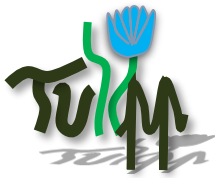
The four tri-speed Ethernet MACs support 10Mb/s, 100Mb/s, and 1Gb/s operations. The MACs support jumbo frames and time stamping through the interfaces based on IEEE Std. 1588v2. The Ethernet MACs can be connected through the serial transceivers (SGMII), the MIO (RGMII), or through EMIO (GMII). The GMII interface can be converted to a different interface within the PL.

High-Speed Connectivity

The PS includes four PS-GTR transceivers (transmit and receive), supports data rates up to 6.0Gb/s and can interface to the peripherals for communication over PCIe, SATA, USB 3.0, SGMII, and DisplayPort.

PCIe

The integrated block for PCIe is compliant with PCI Express base specification 2.1 and supports x1, x2, and x4 configurations as root complex or end point, compliant to transaction ordering rules in both configurations. It has built-in DMA, supports one virtual channel and provides fully configurable base address registers.



SATA

Users can connect up to two external devices using the two SATA host port interfaces compliant to the SATA 3.1 specification. The SATA interfaces can operate at 1.5Gb/s, 3.0Gb/s, or 6.0Gb/s data rates and are compliant with advanced host controller interface (AHCI) version 1.3 supporting partial and slumber power modes.

Debug Connectivity

JTAG

JTAG is the industry-standard interface used to download and debug programs on a target processor, as well as many other functions. It is a 5 wire interface. JTAG offers a convenient and easy way to connect to devices and is available on all ARM processor-based devices.

SWD

The Serial Wire Debug (SWD) mode is an alternative to the standard JTAG interface. It uses only two wires (with an optional third wire for console output) and provides the same debug functionality as JTAG with no performance penalty.

The SWD interface pins can be overlaid with JTAG signals, allowing the standard target connectors to be used.

As written before, ARM processors provide CoreSight debug and trace technology on-chip. It allows real-time trace solution for the entire SoC, making ARM processor-based SoCs the easiest to debug and optimize.

Power Management

For power management, there are four independently controllable power domains: the PL plus three within the PS (full power, lower power, and battery power domains). They are controlled by Platform Management Unit (PMU) that supports also sleep state management. Additionally, many peripherals support clock gating and power gating to further reduce dynamic and static power consumption.

2.4 Board Development

Together with the rising complexity of chip architectures, many versions in the same family are built to cover a maximum of application cases. For FPGAs for example, several SoCs have different Programmable Logic (PL) sizes and may have also different pinouts and power requirements. For board manufacturer it is very expensive to build as many board types as the number of available chips so there are different solutions to simplify the development and the production.

The solution adopted is to solder a chip on a smaller board called a module. The module has standardised interfaces that allows the board manufacturer to develop different carrier boards keeping the same interfaces with the processing module – also in case of the evolution of the chips. It is also good for application designer – he can easily change the processing element if the application requirement changes.

There are several approaches but there is not a clear standard so we analyse different approaches to assemble such a module and connect it to the main board.

Physical connectors:

SODIMM connector allows connecting the module to the carrier with a limited thickness. For higher mechanical strength, the module can be screwed on the carrier.

Connectors like Samtec QSH-060-01-H-D-A, which is used by NVidia for its Jetson module [10], have better mechanical parameters because they are located under the board and allow more wires between the module and the carrier.

Other modules use (located under the module) pitch board-to-board connectors that are optimized for smaller and thinner electronic consumer products.

All these connectors make no differences in power efficiency. Only the form and weight factors, the mechanical constraints and the power supported are relevant factors for selection.

System on Module (SoM) is a printed circuit board that integrates the functions of a whole system in a single board. It allows to select components to match application requirements better and ensures more control of the system design process and better stability. That reduces design costs.

A SoM is often connected to a carrier board that might integrate several SoMs. There are several ways to connect SoM to carrier boards and no standard.

Embedded System Module (ESM) typically includes a CPU, memory and module-specific I/O connectors. It is aimed at being plugged to a carrier board and relies on the standard PCI bus. For example, the company standard by MEN Micro [11] specifies the ESM concept and the different types of modules with one form factor 149 x 71 mm.

PC/104: PC/104 (or PC104) is a family of embedded computer standards which define both form factors and computer buses [3]. PC/104 is intended for challenged environments where a small, rugged computer system is required. The standard is modular and allows to stack together boards from a variety of COTS manufacturers to produce a customised embedded system as shown in Fig. 6. It currently relies on PCI / PCI-Express buses.

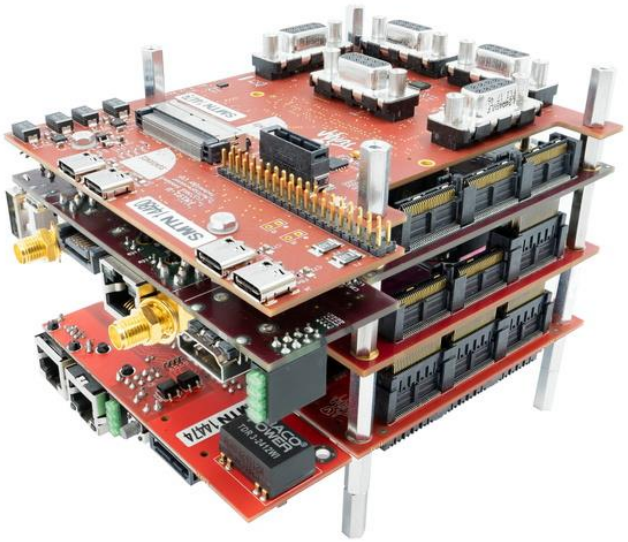


Figure 6: PC104 Stack

The TULIPP carrier board follows the PC104 standard. We chose this standard to simplify mounting the platform environments such as cars and UAVs. Another benefit of choosing the PC104 is that it supports stacking individual boards and thereby improve performance. The board is based on 40x50 mm System on Module that ensures scalability and easy upgrading.

TULIPP platform uses open standard interfaces where it is possible. WP1 defined three groups of interfaces for camera, display and hardware connection (described for example in D1.2). Here is the list of most relevant TULIPP interfaces:

Camera interfaces

- GigE
- CameraLink

Display interfaces

- HDMI

Hardware interconnection interfaces

Hardware interconnection interfaces are needed for connection of different processing boards, external components (like hard drive or SSD) and actors (like autopilot of UAVs). Especially following interfaces are important:

- GigE
- PCIe
- USB

Image processing challenges

The development was based on rules produced in WP1 and WP2. The main goal of the development was to match as good as possible the challenges of embedded image processing, contained in the Table 3 which were produced in WP1:

Type	Challenge
Sensors	More sensors to capture the whole environment. More cameras with better quality and bigger image sizes
Algorithms	More complex, requiring more processing from the hardware. More information will be extracted from the images. More intelligence from the images and from other sources of information (other kind of sensors, communications between drones or cars…)
Energy	The energy shall ideally remain constant. While this might not be possible, it must be mastered as more energy means bigger batteries with higher costs and weight. Mastering the power-budget means much higher processing-efficiency is required.
Development Costs	Development costs must be as low as possible and time-to-market as short as possible. To achieve this, the development must rely on standard libraries and APIs. An operating system is required to capitalize on the implementation of optimized power-efficient libraries based on standard APIs.
Customer Price	The markets addressed by TULIPP are highly competitive. Therefore, the final cost of the solution must be controlled to be able to offer it at a price customers can afford.

Table 3: Energy-efficient embedded image processing applications challenges

The Table 3 takes into account not only current situation, but also tendencies of progress in the application domain.

3 CAPABILITIES OF TULIPP BOARDS

This part contains briefly description of TULIPP carrier boards and its key parameters.

3.1 TULIPP board description

The carrier TULIPP board (type EMC²-ZU4EV by Sundance) is in the PC/104 standard form-factor (90mm x 96mm) and has an expansion board that is connected to a board with GigE and USB3 Video Connectors (Figure 7). Via a cable it can be connected to 'breakout' board that provides low-speed I/O, like RS232, CAN, etc. Both of the expansion boards are four-layer PCBs (low-cost) and might be customized for different user applications.

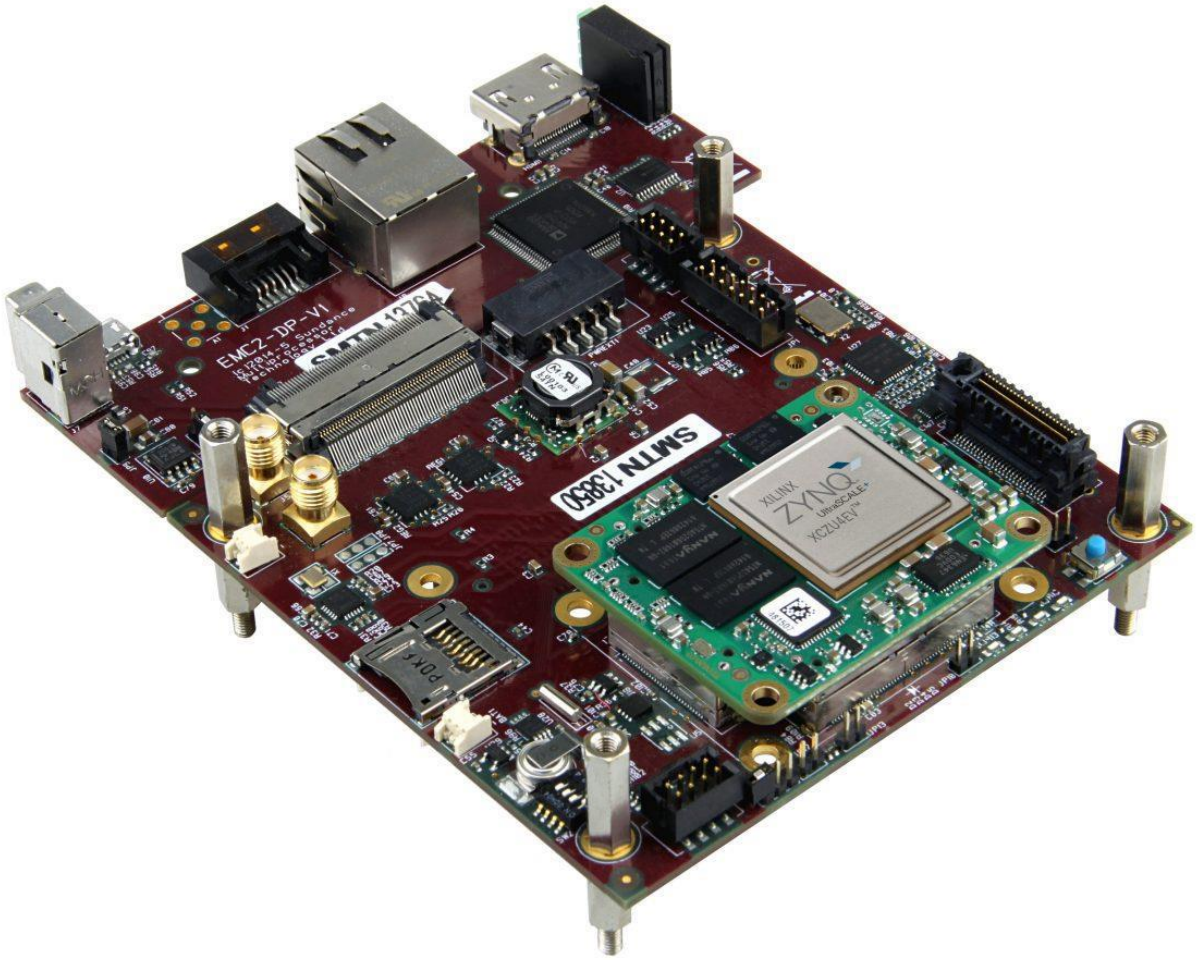


Figure 7: TULIPP board - PCIe/104® Compatible Single Board Computer by Sundance

The TULIPP board is a PCIe/104 OneBank™ SBC with a Xilinx Zynq MPSoC and a VITA57.1 FMC™ LPC I/O board. The main processing power of the board is a Quad Core A53 combined with traditional FPGA fabrics/gates + High-Speed I/O interfaces, like USB2.0, HDMI, 1Gb Ethernet and SATA.

The PCIe/104 OneBank™ design enables the board to be added to robust and rugged installations for various applications. To enable more versatility the TULIPP board also incorporates a VITA57.1 FMC™ LPC connector opening up a range of extra possibilities and expansion options.

The Key Features of the TULIPP board are:

- PCIe/104 OneBank SBC with Quad ARM53 & 1Gbyte DDR4
- Xilinx Zynq SoC FPGA for I/O Interface and processing
- Integrated 1Gb Ethernet, combined w. USB2, SATA-2
- PCI Express Gen 2 compatible and integrate PCI Express switch
- Infinite number of EMC²-ZU4EV can be stacked for large I/O solutions
- Expandable with any VITA57.1 FMC I/O Module for more flexibility
- 96mm x 90 mm PC/104 Form-Factor with Cable-less Break-Out PCB Connector

The main part of the board is 40x50 mm SoM with Zynq UltraScale+ from Trenz Electronic (Figure 8):

Figure 8: Xilinx SoM Module with Zynq Ultrascale+ for EMC2 PC/104 Carrier Board

The Key Features of the SoM are:

- Xilinx Zynq UltraScale+ XCZU4EV-1SFVC784E
- ZU4, 784 Pin Packages
- 4 x 5 cm form factor
- Rugged for shock and high vibration
- 2 x 512 MByte 32-Bit width DDR4 SDRAM
- 2 x 32 MByte (2 x 256 MBit) SPI Boot Flash dual parallel
- 4 GByte eMMC Memory (up to 64 GByte)
- Graphic Processing Unit (GPU) + Video codec unit (VCU)
- B2B Connectors: 2 x 100 Pin and 1 x 60 Pin •14 x MIO, 132 I/O's x HP (3 banks)
- Serial transceiver: PS-GTR 4
- GT Reference clock input
- PLL for GT Clocks (optional external reference)
- 1 GBit Ethernet PHY
- USB 2.0 OTG PHY
- Real Time Clock
- All power supplies on board.
- Evenly spread supply pins for good signal integrity.

Board files from Trenz Electronic can be used for program deployment depending on the module installed on the carrier. Trenz Electronic IP Cores [12] can be also used (for example, for HDMI interface).

To ensure maximal energy efficiency and scalability, the TULIPP carrier board has most of its capabilities available at the extension board, accessible through the SEIC connector – see Figure 9:

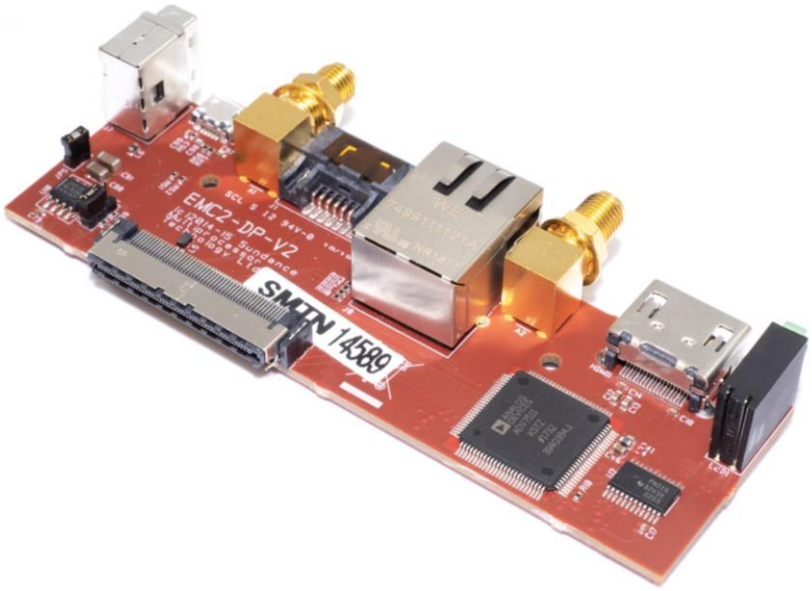


Figure 9: SEIC extension board

A sample structure of TULIPP board is shown in Figure 10. It depends on SoM type and SEIC extension boards. Afterwards, because some components (especially interfaces) can be implemented with FPGA, the structure shown in Figure 10 is quite relative.

The TULIPP board needs an external power supply of 3.3V for most of the components on board, as well as 5V and 12V for the PCIe and FMC ports. The IO voltages for the FPGA banks (3.3V, 2.5V or 1.8V) can be selected through the jumpers. The board works well with any power supply which provides the 12, 3.3 and 5V necessary for the board to work with all its capabilities.

The TULIPP board can work in host or add/on mode, depending on the application and the needs of the user. It has two boot possibilities which can be selected through the jumpers:

- QSPI flash mode
- SD card mode

The board is well supported in Vivado 15.2 and newer versions. It's recommended to use always the newer versions – board files were steady actualised by Sundance.

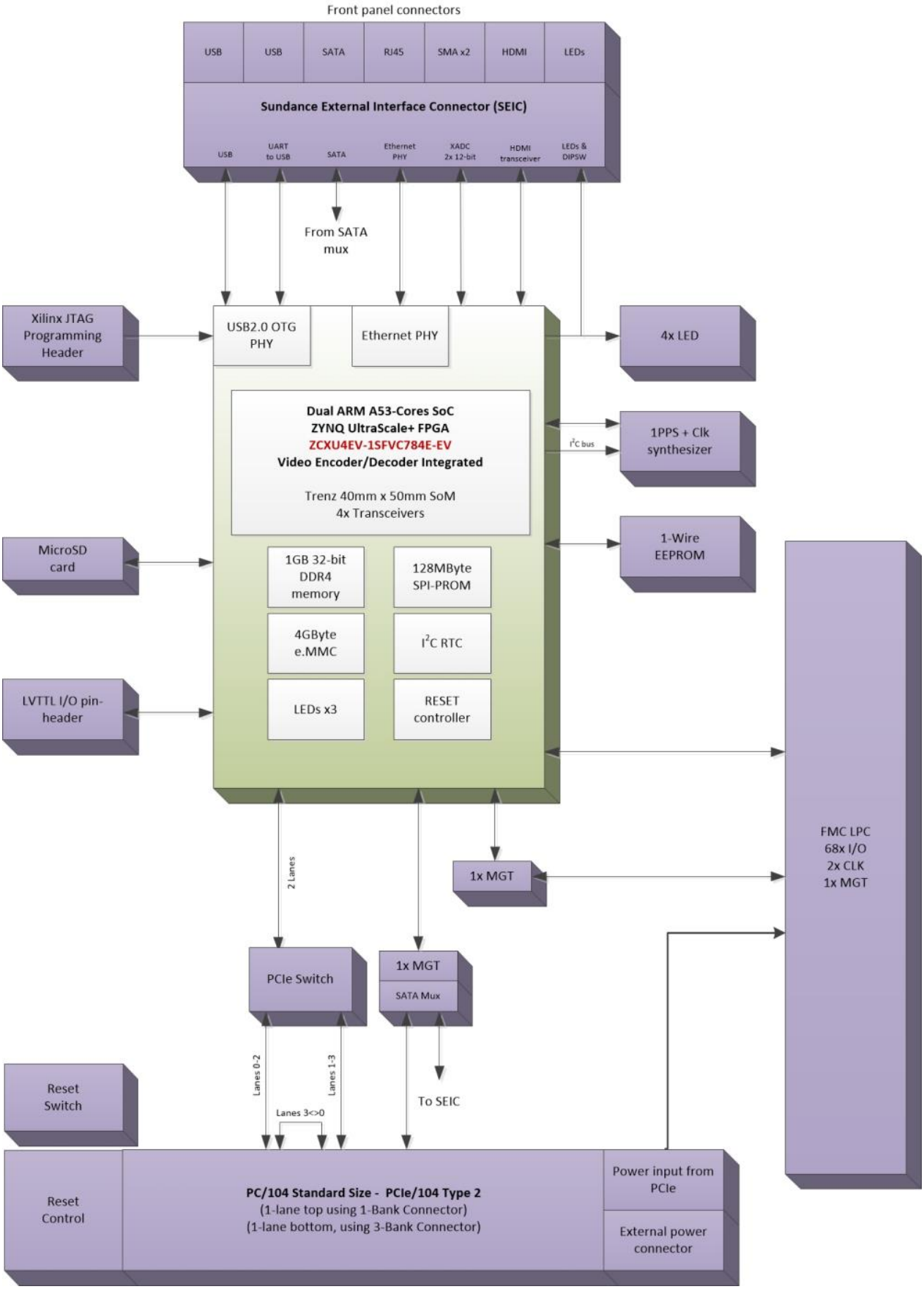
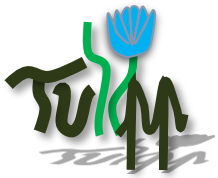


Figure 10: Structure of TULIPP Board PCIe/104® Compatible [13]

3.2 Performance and energy consumption parameters

The main energy consumer of TULIPP board is the Zynq Ultrascale+ MPSoC so that energy consumption is significantly decreased – it consumes a factor 2 less energy than the first board instance. The performance for UAV use case is 2 times better as the first board based on Zynq 7000 SoC (29 fps instead of 14,5 fps) so that energy efficiency of the optimized TULIPP board is potentially 4 times better. Additional performance is provided by faster interfaces and memory. Additional reduction of energy consumption depends on the usage of power saving capabilities provided by the board and by the Operating System, libraries and applications. Not only the hardware plays a role in energy consumption but the whole TULIPP platform instantiation with its software stack and tool chain. In comparison with the first board instance, the new board provides four power domains instead of two, power gating and retention states management that also ensures additional power saving capabilities.

Because the TULIPP Use Cases use different board configurations, the measurement results for their processing time and energy consumption are not comparable one another. The Use Cases implementation algorithms developed for first version of TULIPP board were adapted as well as the OS, the tool chain and the libraries, so that a comparison of energy consumptions can only be done at platform level. A pure comparison at the hardware level only of the processing times of each implementation for the same Use Cases on different hardware would not be informative for hardware as other parameters also impact the energy consumption.



4 CONCLUSION

This deliverable describes the optimized instance of the power-efficient TULIPP board developed in WP2 that follows the design rules developed in WP1. The board uses the hardware selected in cooperation with WP1 and provides all features and interfaces needed for the development and usage of several applications in the embedded image processing domain. It provides all modern possibilities for higher energy efficiency, scalability and is also usable for applications in other domains where high energy-efficiency and real-time capabilities are required.

The TULIPP boards were delivered to all the consortium members that needed them (THA, SYN, FRA, TUD, NTNU and HIP) in time and are also the basis for TULIPP Starter Kit.

5 REFERENCES

[1] Xilinx. Zynq-7000 SoC, https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html#productTable, 2019.

[2] Embedded Multi-Core systems for Mixed Criticality applications in dynamic and changeable real-time environments. https://www.artemis-emc2.eu/, 2014.

[3] PC/104. PC/104 Boards. http://pc104.org, 2018.

[4] HiPeac 2019 Valencia Workshop + Tutorial, http://support.tulipp.eu/viewtopic.php?f=20&t=288, 2019.

[5] Xilinx. Zynq UltraScale+ MPSoC, https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html, 2019

[6] Xilinx. Zynq UltraScale+ MPSoC Data Sheet: Overview, https://www.xilinx.com/support/documentation/data_sheets/ds891-zynq-ultrascale-plus-overview.pdf, 2018

[7] ARM CPU Cortex-A53, https://www.arm.com/products/silicon-ip-cpu/cortex-a/cortex-a53, 2019.

[8] ARM Mali-400 GPU, https://developer.arm.com/products/graphics-and-multimedia/mali-gpus/mali-400-gpu, 2019.

[9] Open NAND Flash Interface (ONFI), http://www.onfi.org, 2019.

[10] Nvidia JETSON, https://developer.nvidia.com/embedded/develop/hardware, 2019.

[11] MEN Micro. Computer-On-Modules, https://www.men.de/products/computer-on-modules-com/, 2019.

[12] Trenz Electronic. Vivado IP Core, https://shop.trenz-electronic.de/de/Download/?path=Trenz_Electronic/Software/Vivado_IP_Core , 2019

[13] EMC2-ZU4EV, https://www.sundance.technology/som-cariers/pc104-boards/emc2-zu4ev/ , 2019.

6 APPENDIX:

https://www.slideshare.net/TulippEu/presentations

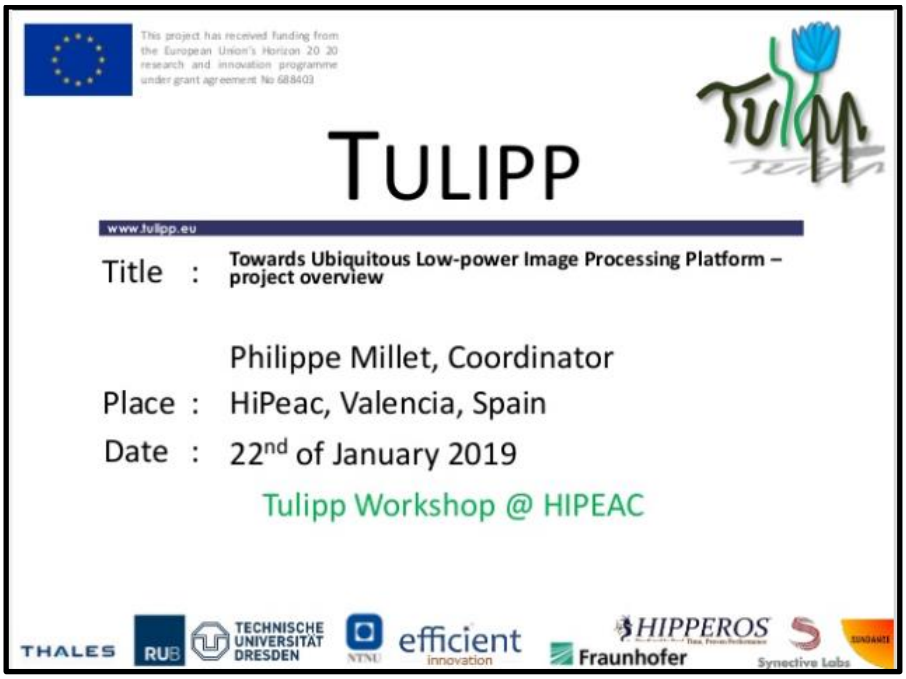


Figure 11: HiPEAC Workshop Presentation - Overview

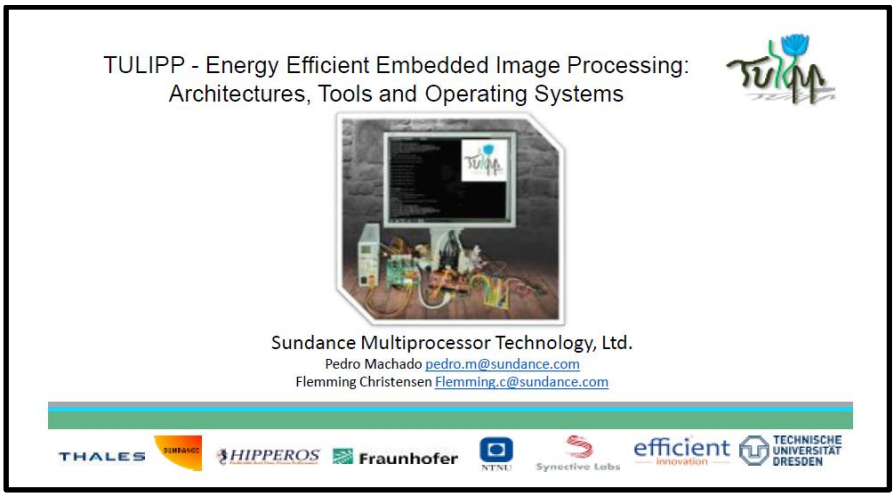


Figure 12: TULIPP Starter Kit – Hardware Platform

Click on the figure to get access to the full set of presentations on TULIPP's SlideShare.