Embedded Systems
Hardware Design

Modern heterogenous computer architecture

System on chip (SoC)

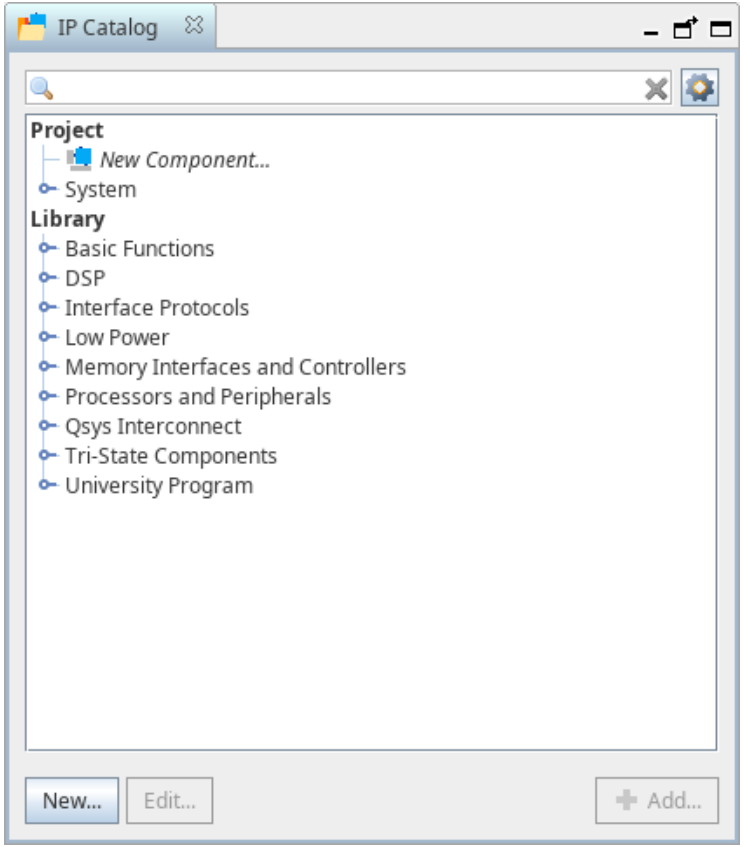
System on chip (SoC) can be defined as system made of various components, where each comoponent serves a unique functionality in the system. The structure of the system depends on the application, for example SoC installed in the mobile phones should have builtin CPU, GPU, 5G MODEM, AUDIO and VIDEO ENCODERS/DECODERS, etc. It's important that all components are made on the same chip.

Intellectual Property Core (IP Core)

The components from which the system is built are often named IP Cores. The main purpose of IP Cores is to provide specific functionality to the SoC. SoC manufactures very often buy the licenses of need IP Cores and build the system from them. Thanks to this, they do not incur additional costs resulting from designing their own solutions.

IP Cores library in Platform Designer

IP Cores are available in FPGA vendor software. Some of them can be free, some of them require vendor license.



Modern computer CPUs

Most modern CPUs are SoC and have builtin additional accelerators dedicated for different applications. Two example accelerators are described below.

Intel® Gaussian & Neural Accelerator - an ultra-low power accelerator block designed to run audio and speed-centric AI workloads. Intel® GNA is designed to run audio based neural networks at ultra-low power, while simultaneously relieving the CPU of this workload.

Intel® Image Processing Unit - an integrated image signal processor with advanced hardware implementation that improves image and video quality of cameras.

RISC-V

The RISC-V (pronounced as risk-five) architecture is an open-source instruction set architecture (ISA) that has gained significant attention in recent years due to its flexibility, modularity, and extensibility. This means, unlike proprietary architectures, you get access to the blueprints and can customize it as you see fit. As an open-source ISA, RISC-V allows for a wide range of customization options, enabling developers to create processors tailored to specific applications and use cases. This has led to its adoption in various industries, from embedded systems and IoT devices to high-performance computing and artificial intelligence. With RISC-V, the benefits are: cost-effective custom processors, innovative applications, and robust security implementations. The technology is considered as the future of processing, customizable in your hands.

Source https://www.wevolver.com/article/risc-v-architecture

RISC architecture vs CISC architecture

Aspect	RISC	CISC
Instructions Per Cycle	Small and fixed length	Large and variable length
Instruction Complexity	Simple and standardised	Complex and versatile
Instruction Execution	Single clock cycle	Several clock cycles
RAM Usage	Heavy use of RAM	More efficient use of RAM
Memory	Increased memory usage to store instructions	Memory efficient coding
Cost	Higher Cost	Cheaper than RISC

Source https://www.wevolver.com/article/risc-v-architecture

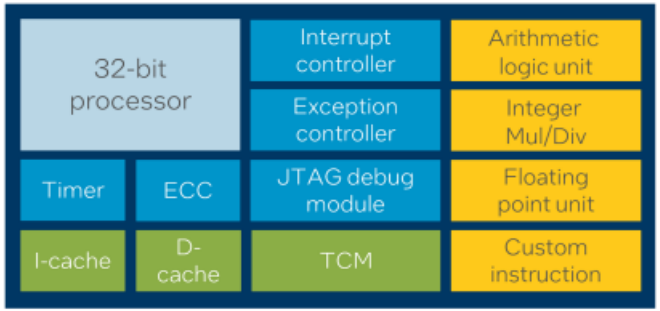
RISC-V standard extensions

- M-extension - adds support for integer multiplication and division instructions
- A-extension - rovides support for atomic memory operations
- F-extension - adds support for single-precision floating-point arithmetic operations
- D-extension - extends the F-extension by adding support for double-precision floating-point arithmetic operations
- C-extension - introduces a set of 16-bit compressed instructions
- V-extension - adds support for vector processing
- B-extension - provides a set of instructions for efficient bit-level manipulation

Source https://www.wevolver.com/article/risc-v-architecture

RISC-V implementations

- VexRiscv
- ITE IT8XXX2 series
- RV32M1 SoC
- Mi-V RISC-V
- SiFive FU540-C000
- Nios® Vr
- AMD MicroBlaze™ V Processor



Source https://docs.zephyrproject.org/2.7.5/boards/riscv/hifive_unleashed/doc/index.html
https://www.intel.com/content/www/us/en/products/details/fpga/nios-processor/v.html

Litex

The LiteX framework provides a convenient and efficient infrastructure to create FPGA Cores/SoCs, to explore various digital design architectures and create full FPGA based systems.

LiteX provides all the common components required to easily create an FPGA Core/SoC:

- Buses and Streams (Wishbone, AXI, Avalon-ST) and their interconnect.
- Simple cores: RAM, ROM, Timer, UART, JTAG, etc.…
- Complex cores through the ecosystem of cores: LiteDRAM, LitePCIe, LiteEth, LiteSATA, etc...
- Various CPUs \& ISAs: RISC-V, OpenRISC, LM32, Zynq, X86 (through a PCIe), etc...
- Mixed languages support with VHDL/Verilog/(n)Migen/Spinal-HDL/etc... integration capabilities.
- And a lot more... :)

Graphics processing unit (GPU)

GPUs are designed to accelerate generating of 3D graphics. Graphics cards have many other uses. Computing units placed in the GPU can accelerate many complex computes.

Applications of GPU:

- advanced simulations
- AI
- image processing
- signal processing
- optimization
- HPC
- etc.

Source https://www.intel.com/content/www/us/en/products/details/discrete-gpus/arc.html

Tensor Cores

Tensor Cores enable mixed-precision computing, dynamically adapting calculations to accelerate throughput while preserving accuracy and providing enhanced security. The latest generation of Tensor Cores are faster than ever on a broad array of AI and high-performance computing (HPC) tasks. From 4X speedups in training trillion-parameter generative AI models to a 30X increase in inference performance, NVIDIA Tensor Cores accelerate all workloads for modern AI factories.

GPU programming

There are dedicated toolchains and toolkits for GPU programming:

- CUDA (NVIDIA GPU)
- ATI Stream (ATI GPU)
- Intel® oneAPI Base Toolkit (Intel GPU)
- OpenCL (GPU-independent solution)

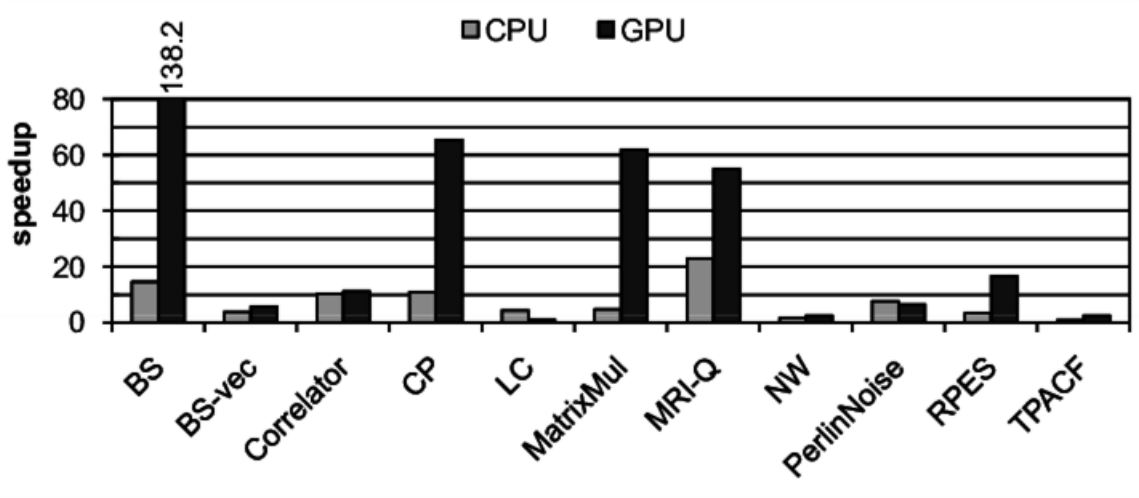
OpenCL

OpenCL™ (Open Computing Language) is an open, royalty-free standard for cross-platform, parallel programming of diverse accelerators found in supercomputers, cloud servers, personal computers, mobile devices and embedded platforms. OpenCL greatly improves the speed and responsiveness of a wide spectrum of applications in numerous market categories including professional creative tools, scientific and medical software, vision processing, and neural network training and inferencing.

Source https://www.khronos.org/opencl/

The speedup of the OpenCL applications with multicore CPUs and a GPU

OpenCL gives significant accelertion for GPU and even CPU.



Source An OpenCL Framework for Heterogeneous Multicores with Local Memory

Applications and libraries using GPU acceleration

Example of applications:

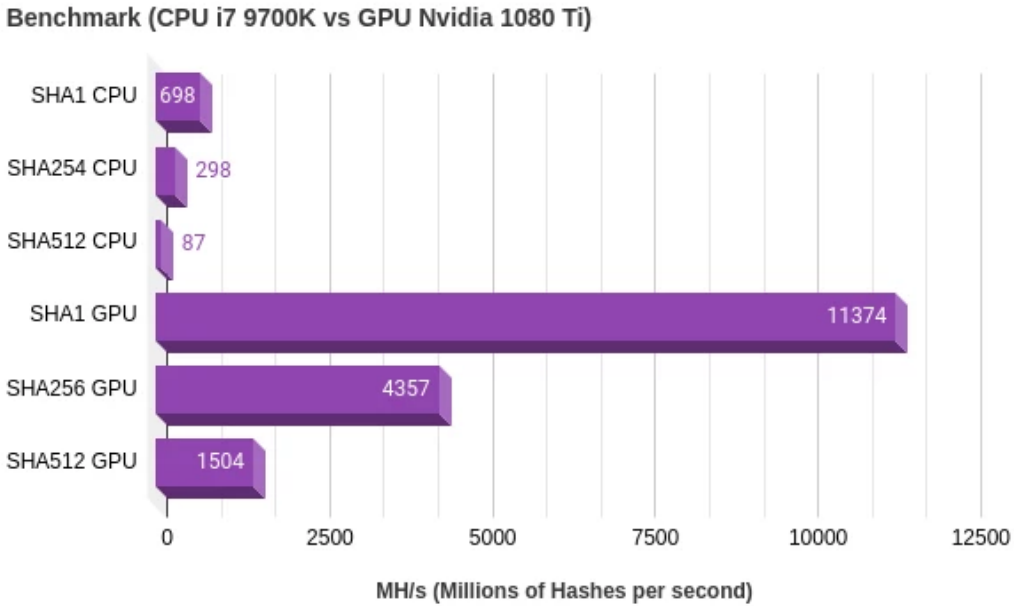
- Adobe's products
- Blender
- OBS Studio
- Corel Draw
- Cyberlink Powerdirector
- Sketchup
- Streamlabs
- etc.

Libraries:

- PyTorch
- JAX
- TensorFlow
- PyTorch Geometric
- DGL
- etc.

Benchmark SHA512 with Hashcat (CPU vs GPU)

The acceleration provided by the GPU is at least 10x compared to the CPU.



Source https://passwordrecovery.io/sha512/

Field-programmable gate array (FPGA)

FPGAs emerged as simple ‘glue logic’ technology, providing programmable connectivity between major components where the programmability was based on either antifuse, EPROM or SRAM technologies.

Source https://www.raypcb.com/intel-altera-cyclone-v-fpga-boards/

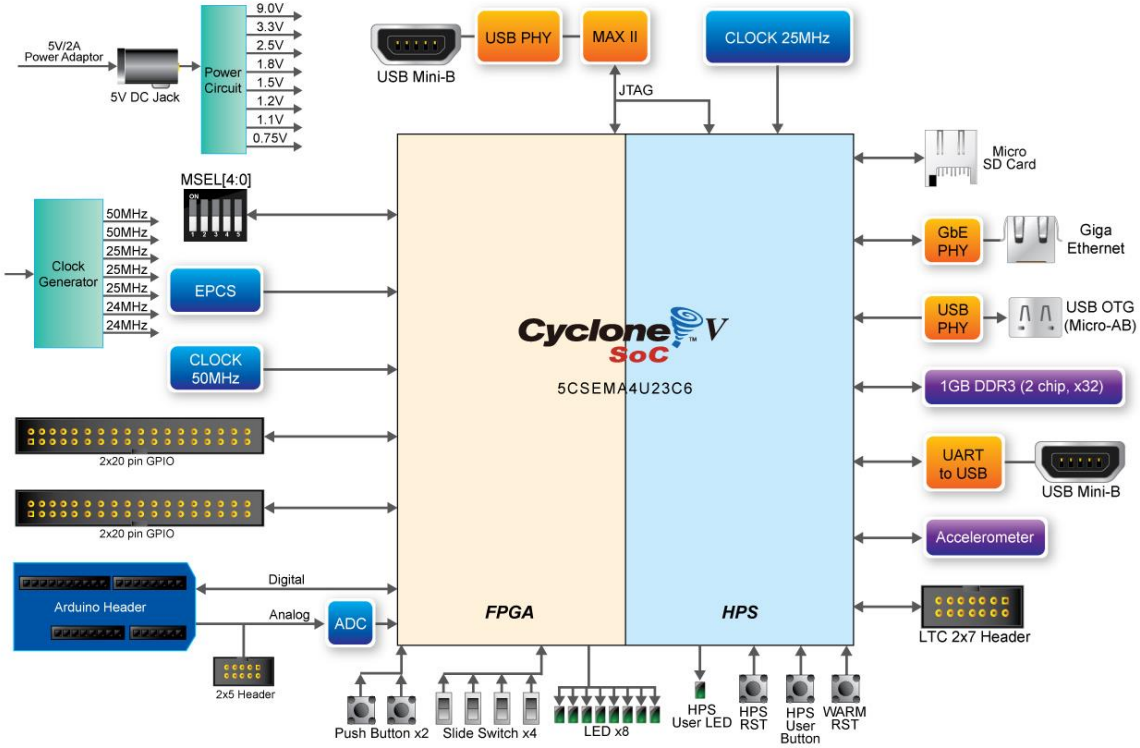
Real-World Applications of the FPGA

- Digital Signal Processing (DSP)
- Software-Defined Networking (SDN
- Embedded Systems
- High-Performance Computing (HPC)
- Cryptocurrency Mining
- IoT and Edge Computing
- Aerospace and Defense
- Scientific Research

Source https://www.mirabilisdesign.com/working-applications-of-fpga/

Field-programmable gate array (FPGA)+Hardware Process System (HPS)

FPGA SoC is a hybrid of FPGA and HPS. They are connected to each other by HPS-to-FPGA bridge. Thanks to this solution, the CPU can control the FPGA.

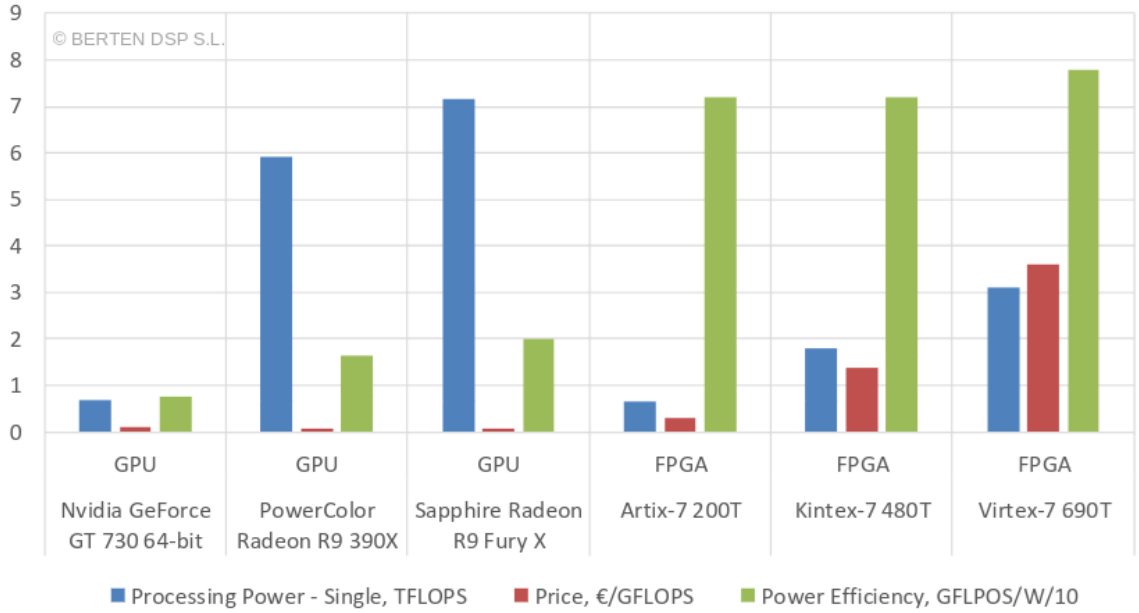


Real-World Applications of the FPGA+HPS

The application of FPGA+HPS is the same as FPGA itself. The only difference is that the FPGA does not have to be used to synthesize the softCPU, because it can use the HPS to control processes and support external peripherals. This allows more FPGA resources to be used.

GPU vs FPGA

FPGA doesn't have such power of floating-point processing as GPU but FPGA is more energy efficient than GPU.



WHITE PAPER GPU vs FPGA Performance Comparison

AI accelerators

AI accelerators are deisgned to accelerate artifical inteligence and machine learning applications.

Examples of AI accelerators:

- Loihi
- Loihi 2
- Akida

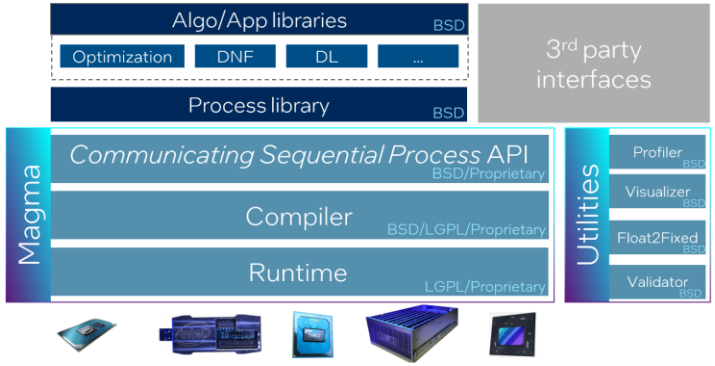
Loihi 2

Loihi 2 is Intel's latest neuromorphic research chip, implementing spiking neural networks with programmable dynamics, modular connectivity, and optimizations for scale, speed, and efficiency. Early research demonstrates promise for low-latency intelligent signal processing.

Loihi 2 is an efficient system supporting applications based on artificial intelligence. In addition to performance, it is also characterized by low energy consumption, ensuring sufficiently high computing power.

Lava

Lava is an open source SW framework to develop applications for neuromorphic hardware architectures. It provides developers with the abstractions and tools to develop distributed and massively parallel applications. These applications can be deployed to heterogeneous system architectures containing conventional processors as well as neuromorphic chips that exploit event-based message passing for communication. The Lava framework comprises high-level libraries for deep learning, constrained optimization, and others for productive algorithm development. It also includes tools to map those algorithms to different types of hardware architectures.



Source https://github.com/lava-nc/lava?tab=readme-ov-file