Hardware acceleration engines for Autonomous Driving at the level 4/5

Currently, there are three main acceleration engines for autonomous driving at the level 4/5: GPUs, FPGAs, and ASICs. Those platforms ensure the deployment of end-to-end autonomous driving systems that meets all computational performance, accuracy, and power requirements to provide cars with the capabilities to make safe operational decisions in real-time. The main processing requirements for the level 4/5 of autonomous driving are frame rates superior to 10 frames per second, and latencies smaller than 100 milliseconds in the end-to-end processing.

One important comparison factor between all the acceleration engines relies on the amount of tail latency that they can reduce while processing autonomous driving algorithms. In order to select the optimal acceleration engine for autonomous driving, it is necessary to comprehend where the main bottlenecks take place during the processing of the algorithms. The bottlenecks are present during object detection, object tracking, and localization. Those algorithms, which are based mainly on the execution of Deep Neural Networks (DNN) and Feature Extraction (FE), consume 99% and 85%, respectively, of the computation power during their execution. Therefore, they are appropriate candidates for acceleration to meet real-time processing constraints.

The following table provides all the details about the advantages and constraints of each architecture:

Architecture	Advantages	Constraints
Graphics Processing Units	+ High processing power. + They meet the latency constraints of 100ms for end-to-end processing in all autonomous driving algorithms. + High memory bandwidths for DNN architectures. Around 10GB/s for NVIDIA's Titan X GPU. + High internal clock frequencies. + Easy programmability due to extensive machine learning software libraries. + Good for training and inference.	- High power-hungry acceleration engine. - Reduce significantly vehicle driving range. - High thermal constraints. - Cooling represents an additional engineering challenge.
Field Programmable Gate Array	+ Mainly used for prototyping. + Highly customization. + High memory bandwidths. Around 6.4 GB/s in Altera's Stratix V FPGA. + Large external memory. + High energy efficiency in comparison to GPUs, and high latency reduction in comparison to multi-core CPUs.	- FPGAs with low amount of Digital Signal Processors (DPS) do not meet the latency constraints of 100ms for end-to-end processing when executing specific algorithms. - Limited on-Chip memory, which is not ideal for DNN architectures. - Hard programmability since mainly Hardware Development Languages (HDLs) are involved for prototyping. - Extensive design periods include code synthesis and hardware verifications.

Application Specific Integrated Circuit	+ Specific functionality means lower trade-offs in terms of power efficiency, computation processing, and memory bandwidth. + They meet the latency constraints of 100ms for end-to-end processing in all autonomous driving algorithms. + High energy efficiency. + Adaptable optimization for low-level optimization. + High thermal performance and high vehicle driving range. + Highly predictable behavior while computing AD algorithms.	- Limited clock frequencies in comparison with its counterparts. - Dependency on proprietary IP cores and machine learning software libraries. - Hard programmability since HDL code is required to achieve performance constraints. - Extensive design periods include code synthesis, verifications and tape-outs.

The technologies used for each hardware accelerator engines vary depending on the algorithm to be processed. Since the acceleration takes place for the detection (DET), tracking (TRA), and location (LOC) algorithms because they consume the vast majority of the processing power, the following table summarizes all the details about the current technologies employed for each architecture at the level 4/5 of autonomous driving:

Architecture	Technology
Graphics Processing Units	DET: Yolo object detection algorithm with NVIDIA's cuDNN software library. TRA: GO-TURN object tracking algorithm implemented on GPsU through Caffe2 and Caffe deep learning frameworks. LOC: ORB-SLAB algorithm is ported on NVIDIA's GPU with OpenCV libraries.
Field Programmable Gate Array	DET and TRA: development of different deep learning layers through hardware modules such as memory controllers, buffers for data fetching, header decoder units for filtering, and processing elements. LOC: based on the oFAST algorithm to extract feature points, and the rBIREF algorithm to compute a descriptor for each feature point. The implementation is done via synthesis tools provided by vendors such as Xilinx or Intel.
Application Specific Integrated Circuit	DET and TRA: hardware AI accelerators from different vendors. Mainly Synopsys and Cadence. Different research approaches exploiting locality to minimize on-Chip data movements to maximize performance. LOC: Uses the same principle of the FPGA's implementation.