

Zhihong Lin,
Strategic Marketing Manager

Dr. Jagadeesh Sankaran,
Chief Architect,
Embedded Vision Engine

Tom Flanagan,
Director,
Technical Strategy
Texas Instruments

Empowering automotive vision with TI's Vision AccelerationPac

Introduction

By September of 2013, Google's self-driving car had completed over 500,000 miles of driving without a single accident under computer control^[1]. Google's disruptive driverless car project was aimed to improve car safety and efficiency by using a combination of video cameras, radar sensors and laser range finders to see and navigate the traffic (along with Google's map database). The Google driverless car prototype is equipped with \$150,000 robotic components, including a \$70,000 laser radar system which is far from commercial use. In August 2013, Nissan announced plans to release driverless cars by 2020 aiming to achieve zero fatalities^[2]. The journey to commercialize the self-driving car will be focused on how to make the autonomous car more affordable, more robust and safer in every corner cases. One of the key technologies to enable autonomous cars is computer vision, using camera-based vision analytics with the goal of providing highly reliable, low-cost vision solutions. While the cost of the camera-based sensor is lower than other technologies, it comes with a huge increase

in processing requirements. Today's systems require that we process image resolutions of 1280×800 at 30 frames/sec often running 5 or more algorithms concurrently. Texas Instruments' latest Application Processor TDA2x based on OMAP5 technology, features the state-of-the-art Vision AccelerationPac to enable advanced driver assistance systems (ADAS) with power efficiency, low cost, programmability and flexibility to power the 20/20 vision for the autonomous vehicles. The Vision AccelerationPac is a programmable accelerator which has specific hardware units and custom pipelines that are fully programmable from a high-level language. This allows vision developers to harness new levels of performance not available using standard processor architectures. The Vision AccelerationPac's programmability support enabled from high-level languages allows end car makers to innovate and explore various algorithmic trade-offs. This is an especially important feature given that these algorithms are far from mature, yet are critical in the accelerated goals for time to market.

The car that sees

Statistics from the United State Census Bureau indicates there is an average of 6 million motor vehicle accidents in the US each year. Young adults and teenagers from 16–24 years old have the highest fatality rate. The statistics also showed the majority of the accidents are caused by human error. It is believed that adding vision and intelligence into motor vehicles can reduce human error and reduce traffic accidents and as a result save lives. It is also believed that automotive vision systems can help reduce traffic congestion, increase highway capacity, increased fuel efficiency and enhance driver comfort on daily commutes.

The advanced driver assistance systems (ADAS) are a key step towards fully autonomous vehicles. ADAS systems include but are not limited to Adaptive Cruise Control, Lane Keep Assist, Blind Spot Detection, Lane Departure Warning, Collision Warning System, Intelligent Speed Adaptation, Traffic Sign Recognition, Pedestrian Protection and Object Detection, Adaptive Light Control and Automatic Parking Assistance systems.

Cameras provide a low-cost means to capture many of the traffic scenarios for intelligent analysis. Stereo Front Cameras can be used for adaptive cruise control to capture real-time traffic conditions to help maintain the optimal distance from the vehicle ahead. Front cameras can also be used for lane keep assist to keep the car centered in a lane, as well as for traffic sign recognition and object detection. Side cameras can be used for cross-traffic monitoring, blind spot detection and pedestrian awareness.

The analytics behind the cameras is what enables the car to have vision-like capabilities. A real-time vision analytics engine is needed to analyze each video camera frame to extract the correct information for intelligent decision. It not only needs enormous computing capacity to process data in the split second intervals required to allow a fast-moving vehicle to make the correct maneuver, it also needs wide I/O to feed the vision analytics engine inputs from multiple cameras to allow simultaneous correlation. Low power, low latency and reliability are also key aspects of the automotive vision systems.

TI technology enabler – Vision AccelerationPac

TI's Vision AccelerationPac is a programmable accelerator created specifically to enable the processing, power, latency and reliability needs found in computer vision applications in the automotive, machine vision, and robotics markets. The Vision AccelerationPac contains one or more Embedded Vision Engines (EVE) that deliver programmability, flexibility, low-latency processing as well as power efficiency and a small silicon die area for embedded vision systems. The result is an exceptional combination of performance and value. Each EVE delivers more than 8× improvement in compute performance for advanced vision analytics than existing ADAS systems at same power levels. See Figure 1 for details.

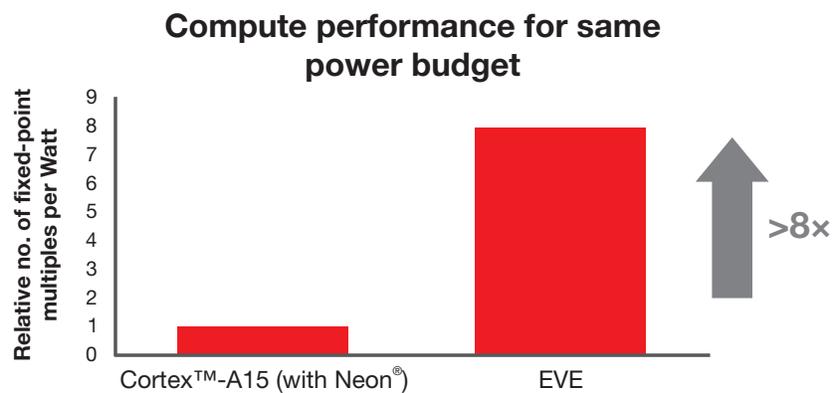


Figure 1: EVE: >8× Compute performance for same power budget with respect to Cortex-A15

Figure 2 on the following page shows the Vision AccelerationPac architecture.

Within a Vision AccelerationPac is one or more EVE, a vision-optimized processing engine that includes one 32-bit Application-Specific RISC Processor (ARP32) and one 512-bit Vector Coprocessor (VCOP) with built-in mechanisms and unique vision-specialized instructions for concurrent, low-overhead processing. The ARP32 includes 32KB of program cache to enable efficient program execution. It also features a built-in

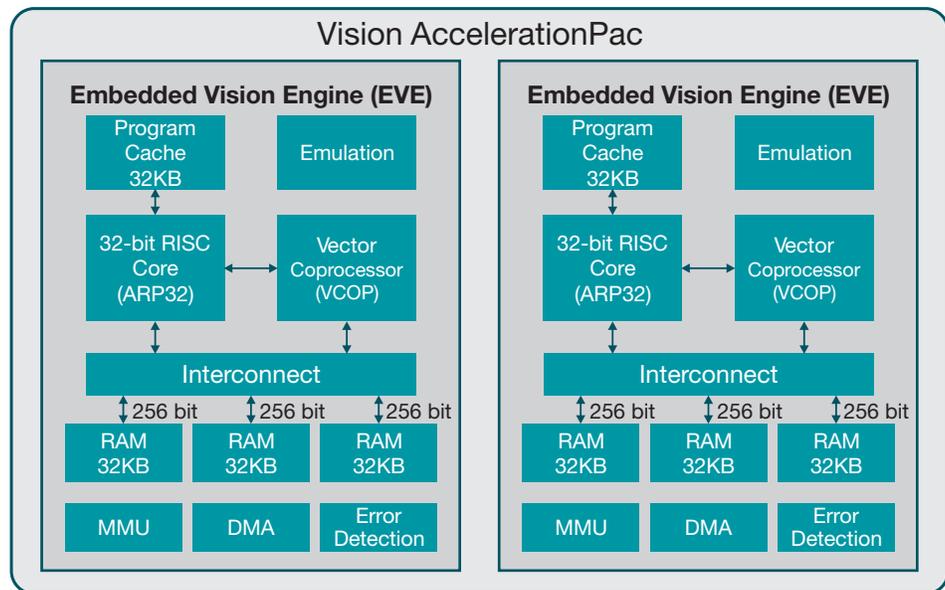


Figure 2: Vision AccelerationPac architecture

emulation module to simplify debugging and is compatible with TI's Code Composer Studio™ Integrated Development Environment (IDE). There are three parallel flat memory interfaces each with 256-bit load and store bandwidth providing a combined 768-bit wide memory bandwidth (6 times higher internal memory bandwidth than most other processors) and has a total of 96KB L1 data memory to enable simultaneous data movement with very low processing latency. Each EVE also has a local dedicated Direct Memory Access (DMA) for data transfer to and from the main processor memory for fast data movement, and a Memory Management Unit (MMU) for address translation and memory protection. To enable reliable operation, each EVE is further equipped with single-bit error detection on all data memories and double-bit error detection on program memory. A key architectural feature is the complete concurrency of the DMA engine, control engine (RISC CPU) and the processing engine (VCOP). This allows, for example, the ARP32 RISC CPU to process an interrupt or execute sequential code, in the meantime the VCOP executes a loop and decodes another in the background, while moving data without any architecture or memory sub-system stalls. It also has built-in support for inter-processor communication by way of hardware mailboxes. EVE enables 8 GMACS processing performance and 384-Gbps with only 290mW of worst-case total power consumption at 125°C, for a most power-efficient vision processing.

The VCOP vector coprocessor is a Single Instruction Multiple Data (SIMD) engine with built-in loop control and address generation. It provides a dual 8-way SIMD with 16 16-bit multipliers per cycle, for 8 GMACS per second at 500-MHz frequency of sustained throughput, sustained by associated loads/stores and built-in zero-looping overheads with rounding and saturation. It has three-source operations, allowing the two vector units to gain an additional 2× and compute 32 32-bit additions in each cycle. VCOP also has eight address generation units each capable of 4-dimensional address to sustain address for four nested loops and three memory interfaces, resulting in zero overhead for four levels of nested looping. This significantly reduces

the compute cycles needed for iterative pixel operations. The vector coprocessor has specialized pipelines for accelerating histogram, weighted histogram and lookup tables and supports common computer-vision processing stages such as gradients, orientation, sorting, bit interleaving/de-interleaving/transposing, integral images and local-binary patterns. In addition, the vector coprocessor has specialized instructions with flexible and concurrent load store operation to accelerate the region of interest decoding and a scatter/gather operation for efficient processing of data from non-contiguous memory. This minimizes the conventional data movement and copying needed for traditional image-processing procedures resulting in ultra-fast processing performance. Speedups of 4× to 12× on a diverse range of functions relative to standard processor architectures are common. VCOP natively supports scatter/gather and region of interest processing. Sorting is a common computer vision function that occurs in multiple-use cases such as identifying target features to track, and matching in dense optical flow searches. EVE significantly accelerates sorting with custom instruction support, allowing EVE to sort an array of 2048 32-bit data points in < 15.2 μsecs.

The Vision AccelerationPac is fully programmable with a standard set of TI code-generation tools, allowing software to be directly compiled and run on a PC for simulation. The ARP32 RISC core can run full C/C++ programs along with TI's real-time operating system BIOS (RTOS). The VCOP vector coprocessor is programmed with a specialized subset of C/C++ created by TI called VCOP Kernel C. VCOP Kernel C is a templated C++ vector library which exposes all the capabilities of the underlying hardware through a high-level language. Algorithms written in VCOP Kernel C can be emulated and validated on a standard PC or workstation by using standard compilers such as GNU GCC or Microsoft® MSVC. This allows developers to incorporate vectorization and verify bit exactness early in the algorithm development process and test with extensive data sets to ensure the robustness of the algorithm. By simply recompiling the source code with TI's code-generation tool, the algorithms can directly run on Vision AccelerationPac. Programs written in VCOP Kernel C have many advantages; they are optimized to leverage the Vision AccelerationPac architecture and instruction set, they have specific loop structure, they operate on vector data and they have an almost one-to-one mapping between C statements and assembly language, resulting in a very efficient code with a small code size and memory footprint.

There are over 100 programming examples available for Vision AccelerationPac. A simple example of array addition using ARM® NEON® SIMD compared with VCOP Kernel C shows ARM can add four 32-bit values in six cycles, achieving an inner loop performance of 1.5 cycles per output, while VCOP achieves eight outputs in one cycle employing full use of its 768 bits of load-store bandwidth. The result is comparable throughput in 1/8th of a cycle per output which achieves an overall cycle to cycle speedup of 12× compared to ARM.

EVE's ARP32 RISC core inside of the Vision AccelerationPac is optimized for control code and sequential processing. It supports running SYS/BIOS, TI's real-time operating system, thereby offering support for threads, semaphores and other RTOS features.

EVE is supported by a full set of code-generation tools including an optimizing compiler, simulator integrated into TI's Code Composer Studio IDE. EVE has built-in support for non-intrusive performance monitoring

through hardware counters. This allows users to monitor multiple performance signals while the application runs and allows in-depth monitoring of the application's run-time performance without having to make any code modifications.

Circular traffic sign recognition example using Vision AccelerationPac

A typical vision analytics processing involves several stages, as depicted in Figure 3 below, including image pre-processing and feature detection, object of interest recognition, image and pattern matching and finally decision making. TI's Vision AccelerationPac is optimally designed to offload the intensive calculation in the first three stages of the vision analytics processing. The decision making often involves classifier, floating-point operation, and matrix conversion which are most effectively processed by C66x DSP cores. For this reason the Vision AccelerationPac is paired with one or more DSP cores in an SoC. The result is an optimal partition of vision analytics workloads.

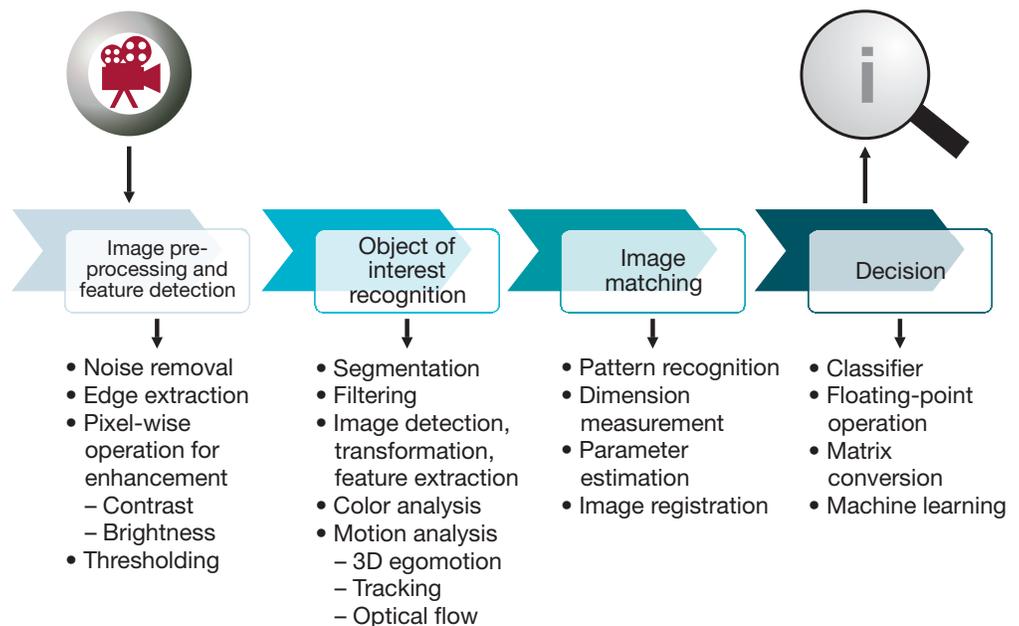


Figure 3: Vision analytics processing flow

TI's TDA2x enables low-power, high-performance vision-processing systems. It features two C66x DSP cores and a Vision AccelerationPac with two embedded vision engines (EVE). It also includes a video front end and vehicle interface for automotive connectivity. Figure 4 on the following page shows the TDA2xx SoC block diagram.

The following explores how the Vision AccelerationPac can accelerate ADAS Circular Traffic Sign Recognition.

In some parts of the world circular traffic signs have a red circle as boundary, so the first step is extracting red-only pixels from YUV422 input data. Next is the calculation of the horizontal and vertical gradients, using brightness and contrast to confirm the red boundary. The Hough transform algorithm is then deployed to find

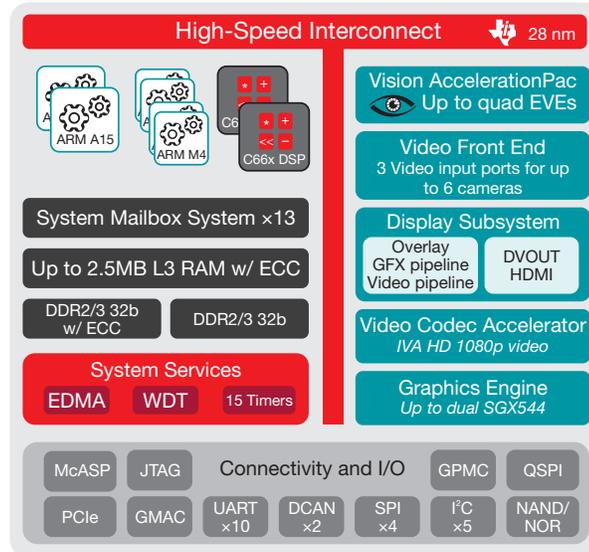


Figure 4: TDA2x block diagram

the circle shape. Now the image in the identified region of interest inside the circle is correlated using patterns stored in a data base to decode the traffic sign (80 MPH) and eventually come to the decision making point. In this case, the conclusion is that the speed limit is 80 miles per hour.

As illustrated in Figure 5 below, Vision AccelerationPac can efficiently offload most of the processing in the circular traffic sign recognition use case, including in this case a cross-correlation-based block-matching template accelerator. The DSP cores are used to improve the robustness of the final decision making. Using Hough transform to find circles is a very compute-intensive task but using a Vision AccelerationPac, for Hough transform for circles only takes 140 Bytes of code space and about $(1.88 * \text{NUM_RADIUS}) + 1.81$ cyc/pix cycles of processing time, where NUM_RADIUS is the number of radii we choose to search in the Hough space, which delivers very fast vision-recognition time with low power consumption and cost-effective silicon die area. The entire traffic sign recognition for a 720×480 frame at 30 frames per second is about 50 MHz or $< 10\%$ of EVE's cycles. Ample processing capacity remains, illustrating that one EVE can run multiple vision algorithms concurrently.

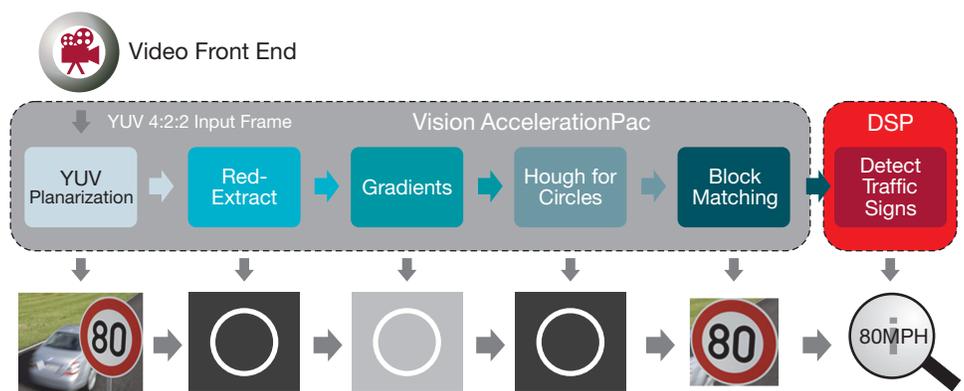


Figure 5: Circular Traffic Sign Recognition using Vision AccelerationPac

Vision AccelerationPac for machine vision – Beyond cars or camera processing

There are many more areas where the Vision AccelerationPac will excel. In addition to video camera analytics processing, Vision AccelerationPac's fixed-point multipliers and hardware pipeline are ideal for radar analytics processing due to its efficient processing of Fast Fourier Transform (FFT) and beam-forming algorithms. Using the Vision AccelerationPac to process a 1024-point FFT takes less than 3.5 μ secs. Thus, radar can be used to complement a camera-in-car vision system to visualize many different traffic and weather conditions.

The same mechanism that is used for vision in cars can be applied to many other industries for machine vision; industrial automation, video security monitoring and alert systems, traffic monitoring and license plate recognition are a few examples. The Vision AccelerationPac can be used to augment the DSP to solve many of today's vision analytics challenges, in a more deterministic and power-efficient manner.

Conclusion

Vision AccelerationPac is Texas Instruments innovative vision analytics solution. With a flexible SIMD architecture that is highly optimized for efficient embedded vision processing, the Vision AccelerationPac delivers very low power consumption and exceptional silicon die area efficiency. The Vision AccelerationPac used in conjunction with C66x DSP cores enables floating-point and matrix computation that significantly accelerate the complete processing chain for embedded vision applications. In addition to being an efficient and reliable architecture, Vision AccelerationPac uses a straight forward C/C++ based programming model whose output is very compact code. This means Vision AccelerationPac-enabled systems have a very low memory footprint to further reduce vision system cost and power. TDA2x SoC with its Vision AccelerationPac is an ideal platform for vision-based analytics driving intelligent automotive systems, industrial machines and “seeing” robots to enhance our lives.

For additional information about the TDA2x SoC, please visit www.ti.com/TDA2x.

References

- [1] [Google tests self-driving car at Va. Tech](#)
- [2] [Nissan Announces Unprecedented Autonomous Drive Benchmarks](#)

Acknowledgement

The authors would like to thank Stephanie Pearson, Debbie Greenstreet, Gaurav Agarwal, Frank Forster, Brooke Williams, Dennis Rauschmayer, Jason Jones, Andre Schnarrenberger, Peter Labaziewicz, Dipan Mandal, Roman Staszewski and Curt Moore for their contributions to this paper.

Important Notice: The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

Code Composer Studio is a trademark of Texas Instruments. All other trademarks are the property of their respective owners.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2020, Texas Instruments Incorporated