CEVA®
The DSP Powerhouse

CEVA-XM4™

CEVA-XM4™

Intelligent Vision Processor

White Paper

Liran Fishel, Director of Architecture

Yair Siegel, Director of Product Marketing

February 2015

Table of Contents

1 Introduction

A change has come to consumer electronics. Once confined to the desktop, processing-intensive algorithms for image enhancement, computational photography and computer vision have moved en masse to camera-ready smartphones, tablets, wearables and other embedded mobile devices. This movement has already hit the limits of today's underlying hardware ability to keep pace in terms of performance, space and energy efficiency, yet we are only seeing the tip of the iceberg.

A clear and tangible indicator of recent advances in mobile imaging and vision that are pushing these limits of design is the dual-camera smartphone, with its accompanying sensor and signal-chain processing for 3D vision and scanning, along with many other image-enhancement features. While consumers may believe they are coming closer to the ideal camera-plus-phone converged solution, designers and equipment manufacturers understand that compromises have been made as the increasingly advanced algorithms are simply relying upon the pre-existing hardware.

This hardware, typically comprising a CPU and a GPU, was not designed to support such processing-intensive imaging algorithms, so it is forcing developers to compromise on features and image quality to match the processing capabilities of the hardware. Even so, the total application continues to consume too much power and drastically shortens battery life, too much so for the still unwary user.

As newer and more-complex algorithms develop to meet both consumer demand for increased functionality as well as manufacturers' need for differentiation, an alternate approach to the underlying vision processing architecture is required if the delicate balance between functionality and acceptable battery life is to be maintained. This alternate approach relies on the adoption of dedicated, on-chip vision processors that are able to cope with both current and future complex imaging and vision algorithms. CEVA-XM4 is exactly that, a fully programmable processor that was designed from the ground up to accelerate the most demanding image-processing and computer-vision algorithms.

This document supplies an overview of the CEVA-XM4 processor's capabilities, architecture, features, target applications, use cases and code examples.

2 Computer Vision Market

A few general trends are common in the various imaging and vision markets:

1. Need for devices and camera-feature differentiation.
2. User desire for single device for communications and photography/video

3. Movement of image processing to the end devices (vs. relying only on the cloud)
4. Explosion in computational load with a fixed power budget
5. Cost pressure (with similar features)

Device differentiation: In today's rapidly evolving and highly competitive market, device and original equipment manufacturers (OEMs) must show real differentiation. To achieve this, many are turning to the camera module. As they continually improve the camera itself, they are also adding new features and technology such as sophisticated computational photography, for better low-light handling, and natural user interfaces (NUIs) for gesture recognition, as well as augmented reality (AR) and depth-sensing capability.

At a higher level, the age of the Internet of Things (IoT) is upon us, so most devices are – or soon will be -- connected to the cloud, where much of the visual analytics is being done. However, as we see more sophisticated image and scene analysis taking place, the trend is to move more of the processing to the camera and reduce the cloud/server processing. The main reasons for this are:

- Camera processing is becoming cheaper, versus expensive cloud/control-room processing (where many end-point devices are controlled at once.)
- Improved real-time response (cloud searches/sorts, end devices supply the features to the cloud engines)
- Reduction in power consumption and cost (of sending raw videos to the cloud)
- Privacy concerns

Beyond smartphones and tablets, optimum energy efficiency is clearly mandated for very small battery-limited devices such as wearables, drones, and robots, but also for automobiles and surveillance systems. The latter are subject to extreme weather and condition changes, yet must remain cool and are quite sensitive to power usage.

Despite the rapidly increasing computational needs in all markets, batteries remain stubbornly consistent in their limitations, placing enormous efficiency pressures on designers of mobile and embedded vision and imaging systems for four key areas: smartphones and tablets, automotive, consumer & wearables and security & surveillance.

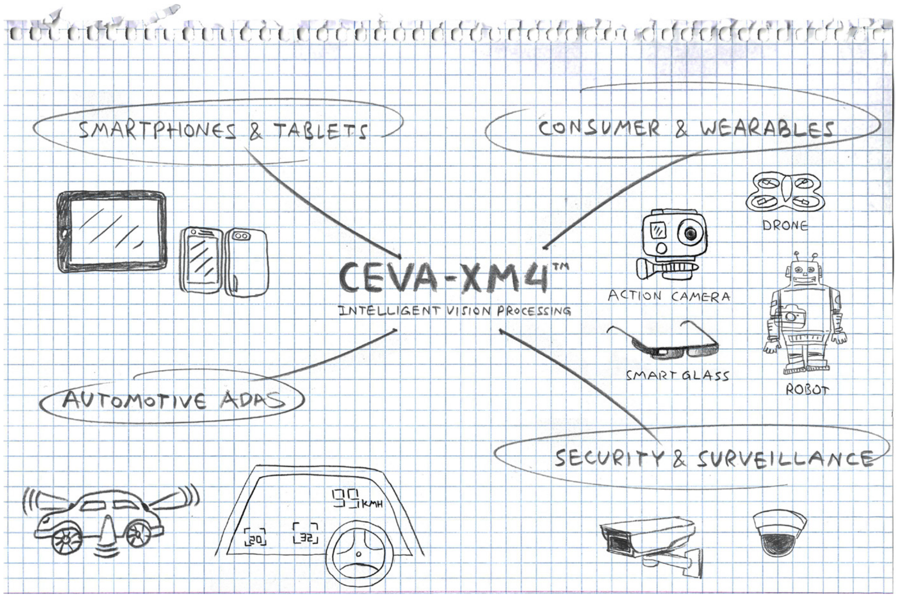


Figure 1: Designers of intelligent vision systems for major market segments are differentiating through the aggressive application of more advanced algorithms and technology, yet must balance that aggression with awareness of the stubbornly immovable limits of today's battery chemistries.

3 Platform Overview

The CEVA-XM4 is an extremely high performance, fully programmable, low-power, fully-synthesizable digital signal processor (DSP) and memory subsystem IP core that was designed specifically to most-efficiently meet the requirements of computer-vision and image-processing applications. The core architecture is a unique mix of scalar and vector units, very long instruction word (VLIW) and single instruction, multiple data (SIMD) functions. The DSP also includes support for both fixed-point and floating-point math, is able to easily connect to hardware accelerators via dedicated ports, and supports easy connectivity to the system bus via standard AXI buses.

The CEVA-XM4 also incorporates sophisticated power management in the form of a power scaling unit (PSU). This controls all clock signals in the system and facilitates power shutdown modes. The PSU thus allows the developer to scale to the required application horsepower, while minimizing the power consumption.

Along with the DSP itself, the CEVA-XM4 IP platform includes:

- A comprehensive application developer kit (ADK) including an extended computer vision pre-optimized library (CEVA-CV), a framework which plugs in directly to the host/CPU processor and enables easy offloading and acceleration of CV tasks from the CPU to the DSP, as well as software modules which automate handling of system & memory tasks at the frame level. All these ease the algorithm developers' task load and abstract the CPU offloading, while improving efficiency and reducing power consumption.
- Software product support, such as super resolution and digital video stabilizer.
- A comprehensive Eclipse-based software development environment, including an optimizing C/C++ compiler, debugger, profiler, and a cycle-accurate simulator.
- A fully featured hardware development platform, including relevant device drivers and peripherals to enable early and fast prototyping.

- An expanding partner ecosystem, which leverages industry-leading technology providers who collaborate with CEVA to offer ready-to-market optimized and efficient algorithms

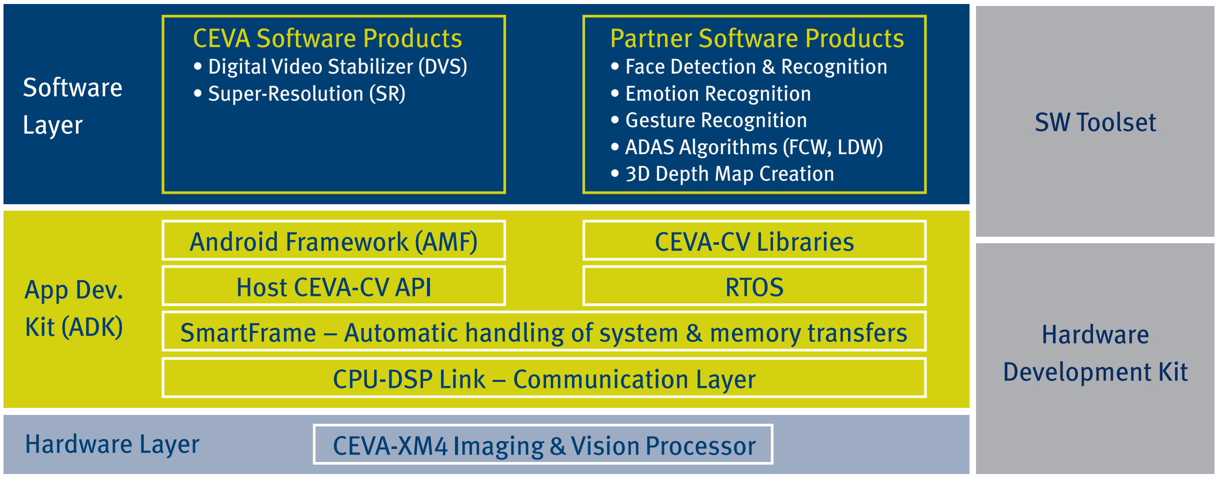
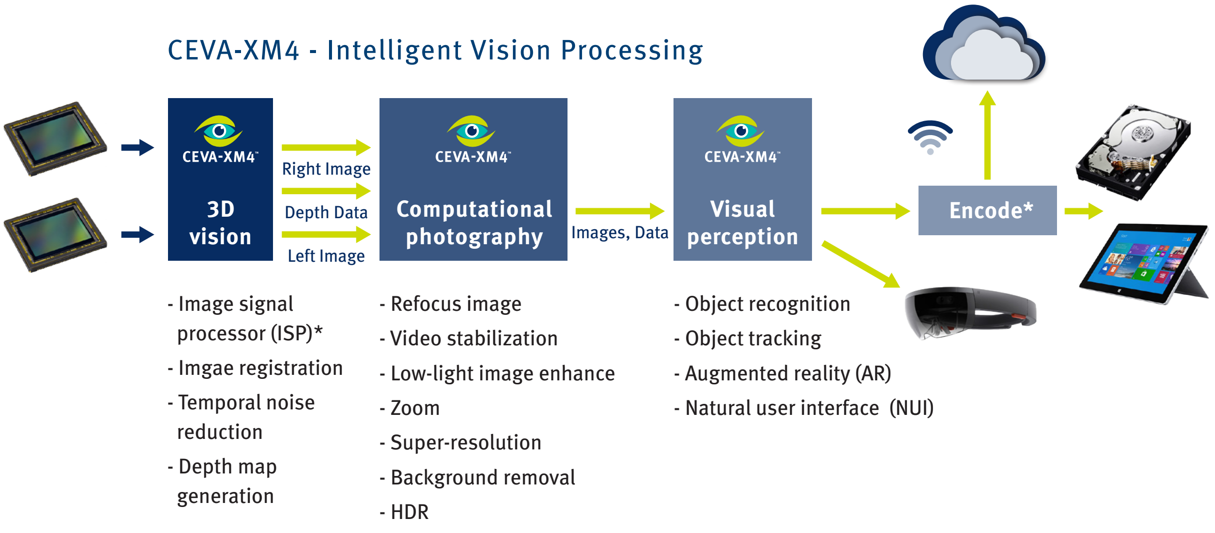


Figure 2: Along with the fully synthesizable DSP optimized for vision and imaging, the CEVA-XM4 platform includes software support at multiple layers, from hardware to a development kit to ready-to-use application libraries.

The CEVA-XM4 leverages the well proven infrastructure, tools and ecosystem of its predecessor the CEVA-MM3101 and is backwards compatible.

4 Target Applications

The CEVA-XM4 can be used in SoCs to perform intelligent vision processing and offload the CPUs and GPUs. The main areas of expertise relate to image preparation as in 3D vision, improving the image as in computational photography and generating sophisticated visual perception and analytics on the input data.



* these are most appropriately implemented by HW accelerators

Figure 3: The CEVA-XM4 offloads the CPU and GPU by performing processing-intensive functions along the imaging preparation as 3D data, computational photography, and visual perception and analytics on the incoming data.

Computer vision algorithms supported include real-time 3D depth map generation and point cloud processing for 3D scanning, object detection, object and image recognition algorithms, ranging from ORB, Haar, and LBP, all the way to deep learning algorithms that use neural network technologies such as convolutional neural networks (CNN).

Computational Photography algorithms supported include refocus, background replacement, zoom, super-resolution, image stabilization, HDR, noise reduction and improved low-light capabilities.

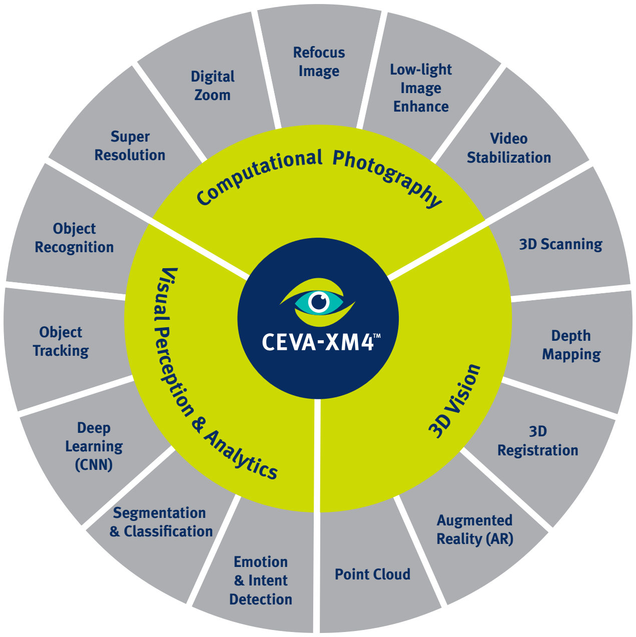


Figure 4: The CEVA-XM4 balances efficiency and high performance across a range of applications, including being able to perform computer vision processing on 1080p or 4K video streams, combine depth generation with vision processing, and run multiple applications in parallel.

Below are a few example use cases, showing what can be achieved using the CEVA-XM4:

- Computer vision processing on video streams (1080p, 4K).
- Combine depth generation with vision processing (e.g. depth + augmented reality, depth + 3D scanning).
- Multiple applications processing running in parallel (e.g. gesture recognition + face detection + emotion detection + eye-tracking + optional depth).
- Multi-frame algorithms on high-resolution images (e.g. super-resolution on 20Mpixel images) or video (e.g. refocus on 1080p or 4K)

5 CPU Offloading

CEVA-XM4 processor is augmented by a robust software infrastructure and framework called Application Developer Kit (ADK) – this is a full set of libraries, software modules and drivers which enable accessing the fully optimized libraries directly from the CPU, without having to program the CEVA-XM4 directly.
By offloading the device’s main CPU and the GPU for performance-intensive imaging & computer vision processing tasks, the highly-efficient CEVA-XM4 dramatically reduces the power consumption of the overall system, while providing complete flexibility. Algorithm developers can leverage the CEVA-XM4’s programmable architecture to implement their own proprietary software, thereby addressing unique use-cases and providing exceptional functionality and the opportunity to truly differentiate their products.

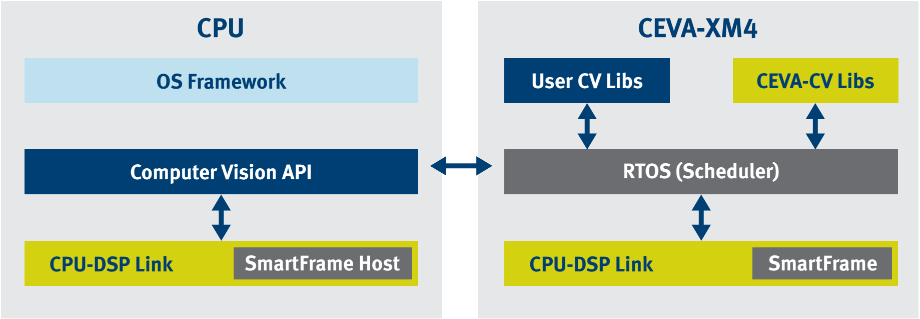


Figure 5: CEVA’s Application Developer Kit (ADK) is a full set of libraries, software modules and drivers that give access to the fully optimized libraries directly from the CPU.

Tools	Description	SW Developer’s point-of-view
CEVA-CV	• Computer-vision functions • OpenCV-based • Pre-optimized for CEVA-XM4	• Abstracts DSP ISA • Enables development using standard widely-used libraries such as OpenCV • Provides optimized performance
SmartFrame	• SW handling all data transfer, frames and tiles • Manages kernels execution and tunneling	• Abstracts all system and memory aspects • Saves memory bandwidth by linking multiple kernels and avoiding external memory accesses
RTOS	• Task scheduling on DSP	• Handles prioritization and task switching
CEVA-Link Driver	• CPU-DSP communication channels and relevant drivers	• Abstracts the CPU-DSP interface • Automates task offloading from CPU to DSP
CEVA-CV API	• Brings the CEVA-CV libraries into the developers CPU domain	• Abstracting CV modules usage • Easy access to utilize CV libraries
Android Multimedia	• Reference design of Android integrated to CEVA-CV	• Eases the integration of CEVA-CV functions with any Android-based application rocessor

Framework (AMF)	• Ideal for offloading CV modules to DSP directly from Android Applications to accelerate performance and lower power consumption

CEVA ADK Feature List

6 Processor Features and Configurations

Architecture Overview

At the heart of the CEVA-XM4 is an efficient and programmable vector-processing DSP. It is based on a dedicated pixel-processing VLIW/SIMD architecture with a 14-stage pipeline and contains nine different units that can work in parallel enabling flexible combination for different type of instructions. All instructions support conditional execution using predication, optimized to save code size.

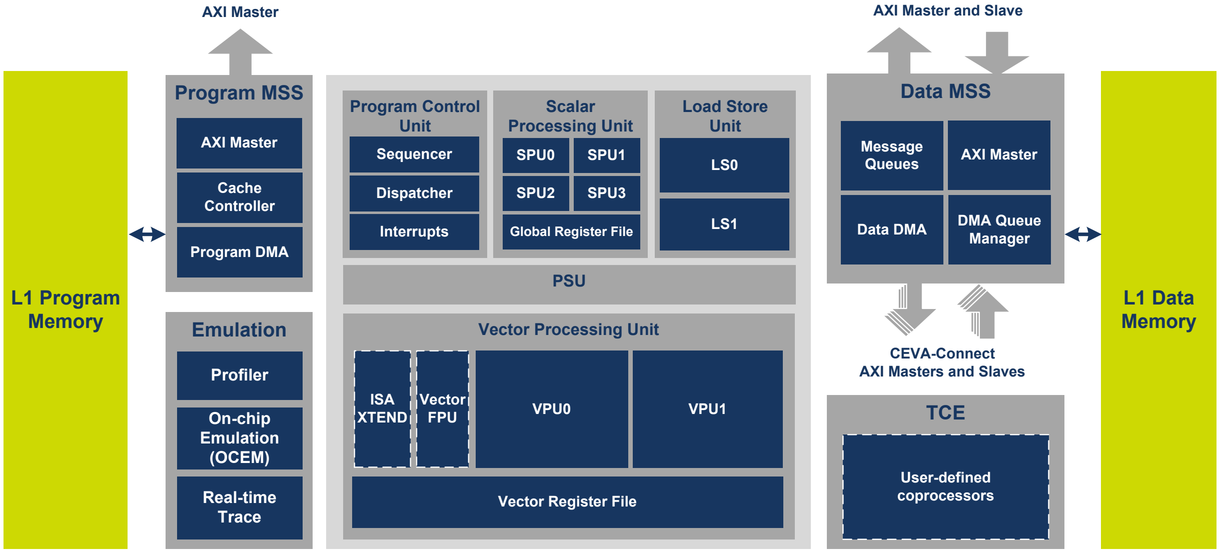


Figure 6: The CEVA-XM4 processor block diagram showing the unique mix of scalar and vector units and very long instruction word (VLIW) architecture. The DSP also includes support for both fixed-point and floating-point math, is able to easily connect to hardware accelerators via dedicated ports, and supports easy connectivity to the system bus via standard AXI buses.

The CEVA-XM4 can handle large amounts of data while keeping the required memory bandwidth at minimum. It achieves that using built-in techniques for reducing data bandwidth transfer from the DDR memory to the core and vice versa, and unique patented technology for data folding and processing on the fly and enhanced internal memory structure. The techniques not only lower the system load on the buses, but also significantly reduce power consumption.

Floating Point Capabilities

The capability to perform floating-point operations is a requirement in many computer vision applications where both a wide dynamic range and accuracy are needed. The CEVA-XM4 supports IEEE754-compliant, single-precision floating-point operations, in both scalar and vector units. Each scalar unit is capable of performing a single floating-point operation per cycle, while the vector units can perform multiple floating-point operations per cycle. CEVA-XM4 supports all rounding modes in hardware.

Non-Linear Functions

Many computer-vision algorithms such as Haar, Connected Components and SURF require the ability to efficiently divide and square root the input data. The CEVA-XM4 is able to perform multiple divisions, square root and inverse square operations per cycle. These operations are supported in both fixed- and floating-point precision.

Efficient MAC Use

Many imaging and computer vision algorithms require heavy use of multiply and multiply-accumulate (MAC) operations. In order to handle the extremely high processing required by the large images and high frame rates of today’s video streams, the CEVA-XM4 incorporates 128 MAC units. Using this amount of multipliers requires an extremely high memory bandwidth, which leads to high power consumption.
To reduce the bandwidth requirement and thus keep power consumption to a minimum, the CEVA-XM4 employs an innovative mechanism, based on data reuse, which is able to use its 128 MAC operations with only a fraction of the memory bandwidth. Kernels with overlapping within the input sources can employ this mechanism to increase the resource utilization. The mechanism is fully flexible and enables the developer to simply implement any two-dimension filter.

Some relevant algorithms that utilize this mechanism include Harris Corner Detector, Bi-lateral filter, 2D correlation, 2D convolution, Gaussian filter, KLT feature tracker, Nagao Matsuyama filter, algorithms that require the sum of absolute differences, and the Sobel filter.

Figure 7 illustrates an example of the data-processing operation. on consecutive data stream such as frame pixels. In this example the same input pixels and filter coefficients are used to calculate four output results in parallel. This example shows how 16 multiplies with an accumulated bandwidth of 512 bits are used with only 176 bits of input data.

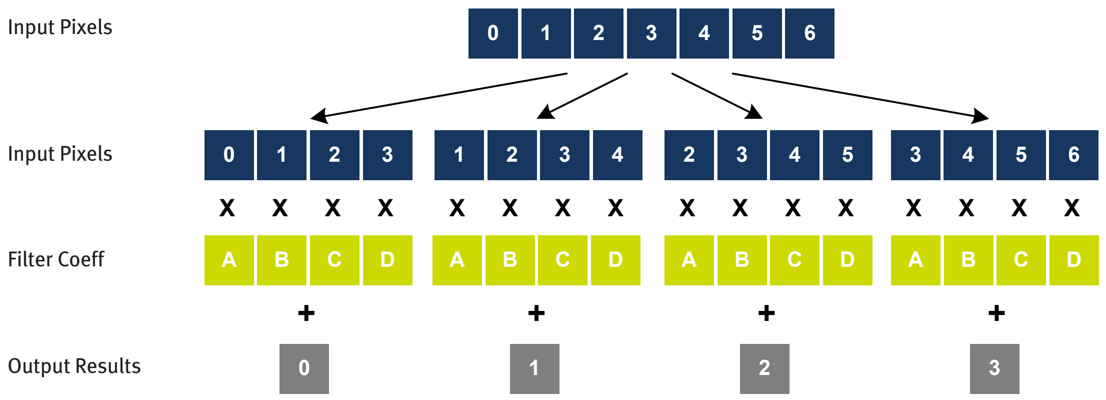


Figure 7: In this example of a pixel line data-processing operation, the same input pixels and filter coefficients are used to calculate four output results in parallel. It shows how 16 multiplies with an accumulated bandwidth of 512 bits are used with only 176 bits of input data.

Parallel Random Memory Access

One of the challenges faced by developers of imaging and computer-vision applications is the ‘randomness’ of the data that needs to be processed. For example, while extracting features of an image (using algorithms such as Canny, SURF and Harris), the extracted features are scattered around the image, limiting the ability to process these features with a standard vector-processing unit (VPU).

However, using the unique and innovative ability of the CEVA-XM4 to access multiple memory locations in a single operation, it is possible to load multiple features into a single vector and in that way use the VPU’s high-performance processing capabilities. Using the parallel memory access on a typical vision kernel, the CEVA-XM4 will achieve a performance improvement of 8x compared to alternative optimal code. This mechanism is extremely useful in taking standard scalar code and ‘vectorizing’ it into parallel operations.

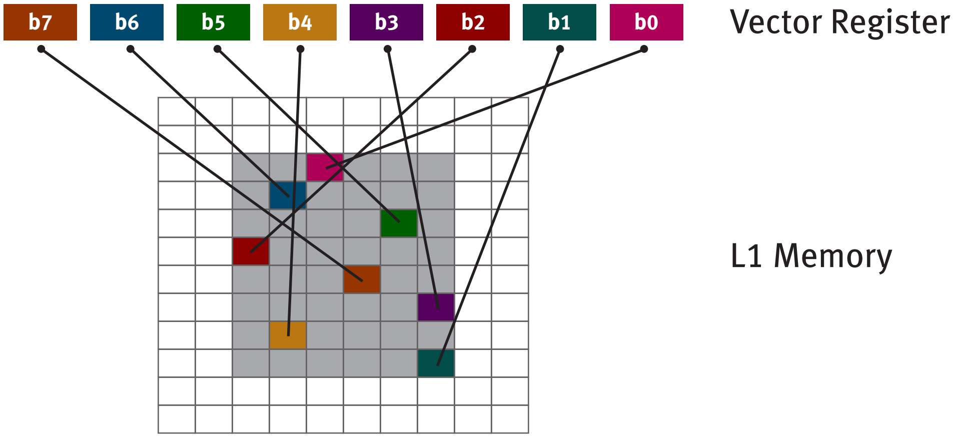


Figure 8: The CEVA-XM4's ability to access multiple memory locations in a single operation means it is possible to load multiple features into a single vector. This parallel memory access on a typical vision kernel will achieve a performance improvement, of 8x compared to alternative optimal code.

Memory Subsystem

The CEVA-XM4 memory subsystem (MSS) is an extended system that can be easily adapted for full SoC integration. The MSS consists of both a program memory subsystem (PMSS) and data memory subsystem (DMSS). The PMSS comprises an L1 instruction memory and an optional four-way cache.

The CEVA-XM4 supports up to 4GB of instruction memory and 4GB of data memory, and has up to nine separate physical interfaces -- with up to eight for data memory and one for program memory. This enables the core to simultaneously access both the program and data memories in parallel to make possible tightly-coupled extensions (TCEs).

The AXI ports integrated in the MSS are fully compliant with the Advanced Microcontroller Bus Architecture (AMBA) versions 3 and 4. The MSS also includes an I/O space that uses dedicated instructions and dedicated space configurations and ports. This space is useful for connecting peripherals to the processor.

7 CEVA-Connect

The CEVA-XM4 has a dedicated data DMA and up-to eight data traffic managers that automatically handle all the traffic coming in and out of the L1 memory.

The data DMA supports various types of memory transfers designed to offload and simplify the software developer effort. The data DMA supports byte-alignment transfers both the incoming and outgoing data; this enables scaling of data pyramids without any requirements from the user to keep alignment of the source and destination. An additional fundamental feature of the DMA is the support of 2D data transfers. This feature simplifies the data transfers of frame tiles and saves the user from having to handle line-by-line transfers.

The data traffic managers include up to eight queue managers. Each manager can handle incoming or outgoing data traffic passing to and from an external hardware accelerator, external memory, host processor or an additional CEVA-XM4 DSP. The queue managers closely monitor the buffers of the CEVA-XM4 L1 memory and the buffers of the external device using dedicated flow-control busses. They then initiate DMA transactions according to the buffers state. Using this method an efficient, closely coupled data transfer between the CEVA-XM4 and external resources is implemented without any overhead to the host processor.

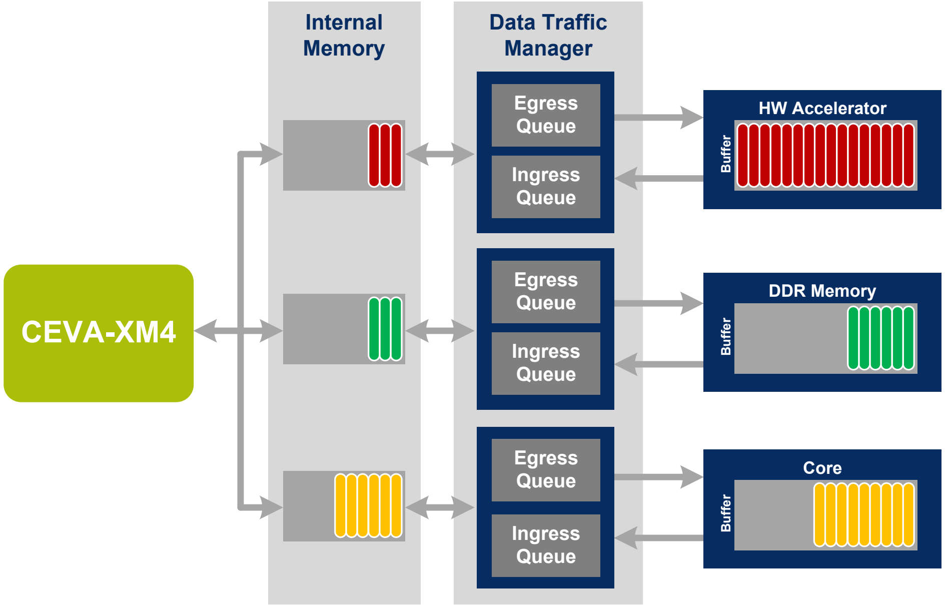


Figure 9: The CEVA-XM4 has up-to eight dedicated traffic managers that automatically handle all the L1 memory ingress (incoming) and egress (outgoing) traffic. These operate by initiating DMA transactions according to the buffer's state. Using this method an efficient, closely coupled data transfer between the CEVA-XM4 and external resources is implemented without drawing upon the DSP and the to the host processor.

8 Development Environment

The CEVA-XM4 has a wide range of CEVA tools and infrastructure, including an Eclipse-based software development environment, C/C++ compiler, documentation, linker, debugger and profiler:

- C/C++ compiler. Supporting auto-vectorization, C language extensions in OpenCL-like syntax, vector types for C programming, and extensive Vec-C support.
- Software simulators. High-speed instruction set and cycle-accurate simulations running on Windows or Linux. These models are used for software and application development and give full visibility to the processor resources and pipeline. These simulators can also be used for model prototyping such as ESL system simulation tools.
- On-chip emulation support. Supporting emulation function & cache profiling. Processor registers, disassembly and memory (internal and external) view.
- Multi-core debugging environment. Allows connection and simulation of multiple instances of the core, each running different application. Multi-core debugging is supported both in simulation and emulation.
- Development board. PCB with CEVA-XM4 core and memory subsystem. These boards can be used to run real-time applications, as well as for demos, hardware prototyping and performance evaluation.

9 Compiler Vectorization

The CEVA-XM4 compiler supports multiple programming levels that allow the developers to make the best trade-off between development effort and performance. The processor has high SIMD capabilities. To make optimum use of these capabilities, the compiler includes multiple C level coding options which developers can use.

Auto-Vectorizing C Code

The C compiler has built-in capabilities to convert C code to SIMD instructions using the VPU vector operations. The example below shows C code that will be automatically converted to a SIMD operation on the VPU. The compiler will be converted to a single SIMD operation that will execute up to 32 iterations in a single cycle.

```
for (i=0; i<len; i++)
{
  Y[i] = (A[i]+B[i])*C[i];
}
```

Although the compiler can automatically generate SIMD instructions, there are many restrictions in the C language that might limit the compiler's ability to automatically vectorize the code. These restrictions could be due to non-contiguous data in the memory, memory aliasing and issues relating to the order of operations such as read-after-write hazards.

Adding Vector Types

In order to avoid the auto-vectorization limitations and give the programmer better control of the results, the C language was extended with new vector types such as short8, ushort32. These types represent a vector of scalar data types as defined by the OpenCL standard. The example below shows C code with vector types providing the same functionality as the plane C example above by using the extended vector types. These variable type enhancements to ANSI-C are also referred to as Vec-C.

```
for (i=0; i<len; i+=32)
{
  vA = *pA++;
  vB = *pB++;
  vC = *pC++;
  vY = (vA + vB) * vC;
 *pY++ = vY;
}
```

Optimized Performance Using Intrinsic Operations

There are some cases where the developer will want full control over the generated vector SIMD operations to achieve the optimal performance. In these cases, additional level of control is provided such that the use can use the exact vector operation while keeping the rest of the code in C and letting the compiler handle the complex register allocation and instruction scheduling. The example below shows C code with vector types and intrinsic operations providing the same functionality as the two examples above by using the extended vector types and calling upon the *vmpyadd* instruction.

```
for (i=0; i<len; i+=32)
{
  vA = *pA++;
  vB = *pB++;
  vC = *pC++;
  vY = vmpyadd(vA, vB, vC);
  *pY++ = vY;
}
```

10 Summary

The CEVA-XM4 was designed to answer the strong need in the market to adopt dedicated vision processors into future devices which are able to cope with the upcoming complex imaging and vision algorithms and be able to address those in an energy efficient manner to maintain long battery life. With its innovative architecture the CEVA-XM4 was designed to answer this need while providing high-level C programmability. Its robust memory architecture and advanced system connectivity ensures that the CEVA-XM4 will be able to seamlessly integrate into complex SoCs with minimum effort while providing the best area and power efficiency to date.