eSAME 2015 Conference

EFFICIENT FPGA-BASED EMBEDDED VIDEO SYSTEMS AND THE MOVE TOWARD SOFTWARE DEFINED VIDEO INFRASTRUCTURE

AARON BEHMAN, Xilinx, Inc.

San Jose, CA, USA

Abstract: *As video systems grow in complexity with content creators pursuing immersive experiences and the distribution of signals transition from baseband to packet-based protocols, professional video creators must employ more scalable and power efficient techniques. This paper will present how FPGA technologies are used to implement efficient real time embedded video systems and will also discuss the trend toward virtualizing these functions in the data center.*

I. INTRODUCTION

FPGA and FPGA-based system on chip (SoCs) devices are best suited in applications where massive parallelism provides computational advantages over alternatives. This can be seen in markets like professional video where FPGAs and FPGA-based SoCs offer high computational productivity relative to central processing units (CPUs) and graphics processing units (GPUs).

II. MARKET PROBLEM

A. The Consumer Demand for Immersive Experiences:

Consumers are demanding more immersive experiences. This is being addressed through the application of more, better and faster pixels in video and vision system design [1]. The following are ways in which these highly immersive systems are being created: 1) through more pixels, resolutions are scaling beyond High Definition (HD) 1080p60 to 4K (4x HD) and 8K (16x HD) and support for fully immersive virtual reality with 360° surround viewing experiences; 2) through better pixels, bit depths are increasing from 8-bit to 12-bit, deeper color gamut, e.g. ITU-R BT.Rec 2020 released in support of wider color gamut for 4K and high dynamic range (HDR) or enhanced dynamic range (EDR) to produce more stunning and realistic images that are more like the human visual system; 3) through faster pixels, refresh rates are increasing from 24 frames per second (fps) to 48fps in digital cinema and 60Hz to 120/240Hz in video. The advances in video and vision systems create challenges in terms of how to manage this additional video bandwidth.

B. Advertising Channels Coming from New Sources:

Streaming and over-the-top (OTT) services like Amazon Prime, Netflix and Hulu are threatening traditional linear broadcast. User generated content (UGC), e.g. YouTube, Twitch.tv, is gaining more share of viewers. Cellular carriers are getting into the content space as well with AT&T recently acquiring DirecTV [2]. This transition in how content is consumed creates new opportunities for revenue, through viewer subscriptions to channels and new targeted advertising models (e.g. pay-per-click, overlaid ad graphics).

C. Remaining relevant requires scalability and flexibility:

In order to continue to harness advertising revenue opportunities broadcasters must become more agile and scalable. Coax is now being replaced by Ethernet in the studio in emerging cases and capital expense in the form of broadcast infrastructure equipment is starting to be replaced by operating expense through scalable and more cost efficient cloud-based data center services. Virtualization of channel production proves a compelling technology to economically scale up more channels of content and therefore exploit new opportunities for revenue [3].

Standard silicon solutions cannot adapt fast enough to meet the evolving needs of the market or intersect rapidly evolving standards.

III. FPGAS IN THE PROFESSIONAL VIDEO VALUE CHAIN

The professional video equipment market can be categorized into four stages of creation and consumption of high production value content. The initialism to describe this market and its key functions is ACDC.

"A" for acquisition, "C" for creation or contribution, "D" for distribution, and the last "C" for consumption. Some examples of the types of equipment used in each stage are as follows:

Acquisition, e.g. professional broadcast cameras, integrated receiver decoders;

Contribution, e.g. video routers, switcher decks, multi-viewers, satellite uplinks;

Distribution, e.g. RF modulators for satellite, terrestrial, cable, and distribution encoders;

Consumption, e.g. projectors for cinema exposition, and reference studio monitors.

Fig. 1 is an abstract representation of the video value chain.

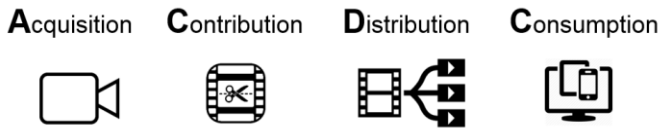

Fig. 1: The professional video value chain – ACDC

IV. Typical functions implemented in an FPGA-based embedded video system

A typical embedded video system is comprised of the following functions: input interfaces; image processing algorithms (in the case of vision-based systems), video processing algorithms, graphics, compression algorithms (codecs), video analytics algorithms, encryption algorithms, infrastructure or video system framework elements like memory controllers or frame buffers, and output interfaces. At a high level this represents a large majority of the functions that are implemented in a typical video system. Fig. 2 depicts this general architecture.

Modern FPGAs are capable of implementing these functions and are able to be reconfigured to support additional standards as they emerge (such as those to support more, better and faster pixels) or are capable of loading alternative configurations to support other use cases on the same hardware platform. Furthermore, partial reconfigurability enables hardware designers to configure a system and reconfigure partitions of a systems on an as needed basis without impacting the general system implementation. With the advent of fully integrated ARM®-based processing systems complete self-contained systems have emerged with control plane and data plane all implemented in a single monolithic piece of silicon.

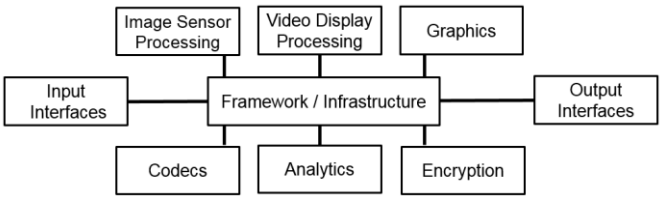

Fig. 2: General architecture representing a video or image processing system

Input/Output Interfaces are industry standard interfaces that allow for the transmission and reception of video data that are either compressed or uncompressed. Typical interfaces that can be implemented in FPGAs include: high-definition multimedia interface (HDMI) versions 1.4b and 2.0; DisplayPort version 1.2; the Society for Motion Picture and Television Engineers' (SMPTE) serial digital interface (SDI) (including the recently ratified 6 Gbps and 12 Gbps versions; mobile industry processor interface (MIPI); low voltage differential signaling (LVDS) interface. In addition to these baseband interfaces there has been a recent industry emergence of packet-based interfaces for transporting video over Ethernet. Some of these interfaces include: SMPTE ST 2022 (versions 1, 2, 5, 6 and 7), Ethernet Audio Video Bridging (AVB) for both 1 Gbps and 10 Gbps networks.

Image Sensor Processing (ISP) are a specific set of algorithms that are used in conjunction to form an image processing pipeline. Foundational algorithms include defective pixel correction, color filter array interpolation, image statistics for generating histograms for correcting exposure, focus and white balance, color correction matrix, gamma correction, image enhancement, motion adaptive noise reduction, color space conversion and chroma resampling. When combined these algorithms can take raw Bayer data from image sensors and create video from those sensors. Having these algorithms implemented in FPGAs enables the developer to continually improve the algorithm. High-level synthesis (HLS) tools have emerged that now enable developers to maintain algorithms in higher-level languages like C versus having to covert the algorithms to very high speed integrated hardware description language (VHDL) or Verilog.

Video Display Processing, like ISP, is a specific set of algorithms for processing and grooming video data. Typical algorithms include cross conversion from one video interface standard to another, color correction, motion adaptive deinterlacing, noise reduction, scaling and down scaling, alpha blending and graphics overlay or on-screen display (OSD). As with ISP-based algorithms an emerging trend is to leverage new high-level synthesis tools to generate the FPGA bitstream from higher-level languages like C to configure the devices to run those algorithms. This greatly improves development time and enable algorithm developers a way to validate performance in real time hardware versus simulating algorithms on personal computers (PCs).

Graphics, 3D graphics engines that are compliant with OpenGL® ES 1.1 API are available for FPGAs and can be used to implement sophisticated 3D graphics as well as graphical user interfaces (GUIs) [4]. Xilinx has released a next generation FPGA-based SoC known as the UltraScale+® Multi Processing SoC (MPSoC) which integrates the ARM® Mali® 400 GPU and is capable of supporting 3D graphics and OpenGL ES 2.0.

Codecs are highly complex algorithms that are used to compress video bandwidth so that it is easier to manage in or transport from a video system. There are two high level categories for codecs: 1) those that are visually lossless; and 2) those that are visually lossy. Codecs can be applied to a number of use cases and tend to target two primary use cases:

1) video contribution applications where video quality must be preserved, but compressed to a level where it is more manageable in the system and over the network; and 2) video distribution applications where quality can be compromised in the interest of having much lower bitrates to enable the distribution of video over consumer intended networks like cable, satellite, terrestrial and Internet streaming networks. Codecs can be proprietary or standards-based, but most codecs widely used tend to be standards-based. The following are some of the widely used codecs in the first use case (contribution) which are high bitrate very low-latency visually lossless codecs: JPEG 2000, VC-2, Tiny Codec (TICO) (a proprietary codec developed and marketed by intoPIX, s.a.), and display stream compression (DSC) version 1.0 which is part of the DisplayPort version 1.3 specification. The following are some of the widely used codecs in the second use case (distribution) which are lower bitrate low-latency visually lossy codecs: MPEG-2, MPEG-4/AVC/H.264, HEVC/H.265, VP8, VP9.

Analytics: also known as video content analysis is a set of specialized algorithms that are a subset of the field of computer vision. The goal of these algorithms is to detect objects, extract features from those objects and then classify those features. Given the parallel nature of these techniques and the computational intensity of these techniques FPGAs are used in embedded system architectures to efficiently analyze video in real-time. FPGAs and FPGA-based SoCs are also applied to deep learning algorithms like convolutional neural networks where embedded systems are being optimized to recognize images on a more power efficient basis than alternative computing resources like CPUs or GPUs [5].

Encryption: is used in an embedded video system context to protect or decrypt secure content. Digital rights management (DRM) media is typically encrypted to prevent misuse or piracy of proprietary content. Complex schemes like high-bandwidth digital content protection (HDCP) is used to encrypt and protect this class of video content. HDCP can be implemented in FPGAs along with other processing capabilities to enable developers to architect systems that can protect/encrypt original proprietary content or display/decrypt that content.

Framework/Infrastructure: lastly, this element of an embedded video system is implemented to link various external components and peripherals with algorithmic sub-systems (as described in the previous paragraphs) into a cohesive embedded vision or video system. A key type of framework/infrastructure includes the ARM Advanced Microcontroller Bus Architecture (AMBA®) Advanced eXtensible Interface version 4 (AXI®4) interconnect to enable a common protocol to connect disparate sub-systems, memories and other peripherals into a connected embedded system. The protocol is further nuanced in support of control plane functions (AXI4 lite), external memory mapping (AXI4 memory mapped) and streaming (AXI4 streaming) in this context video pixel streaming. Xilinx's software defined SoC (SDSoC) environment enables developers to maintain algorithms and define complete systems at the C level and then leverage the environment to profile the elements of the codebase that are best accelerated in FPGA programmable logic versus the ARM-based processing system.

V. An efficient embedded video system

Having now presented a general architecture for an embedded video system based on FPGAs and having gone through, in some detail, the various sorts of interfaces, algorithms and protocol frameworks that can be instantiated in FPGAs let's now take a closer look at a specific example.

The Xilinx Real Time Video Engine (RTVE) is a reference design that is offered by Xilinx to enable embedded video system developers to quickly implement all of the requisite elements of a video system in either a single Xilinx FPGA or Xilinx All Programmable SoC. The RTVE is built around Omnitek's Scalable Video Processor (OSVP). Omnitek [6] is an FPGA and SoC intellectual property (IP) developer, design services house and product company based in the United Kingdom. At a high level this system can support up to eight channels of 1080p60 video or two channels of 2160p60 (4K) video in a single mid-range Xilinx device. These video channels can be further groomed and enhanced applying Xilinx and Omnitek IP to those channels to produce professional quality video.

Referring to the general architecture presented in Fig. 2, Fig. 3 highlights the specific elements of that architecture that are represented in the RTVE.

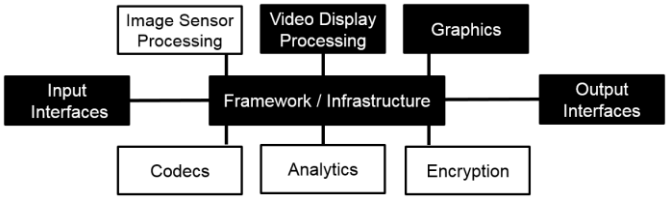

Fig. 3: Relevant elements of the general architecture implemented in the Xilinx Real Time Video Engine FPGA-based reference design.

Taking a closer look at the RTVE relative to the general architecture in Fig. 2, let's describe each element of that system:

Input/Output interfaces: The RTVE implements several video interfaces. SDI is a typical interface used in professional broadcast use cases. The SDI interface can be implemented in FPGAs using IP cores that are made available from FPGA vendors. These IP cores are typically free to license and rely on the multi-gigabit transceivers that are built in to modern FPGAs. HDMI is another common video interface that is used in both professional and consumer use cases. As of this writing the Xilinx RTVE does not natively implement the HDMI

interface; however, the IP core exists to allow a developer to natively implement the interface in the device without the need for an external physical HDMI receiver, transmitter or transceiver device (a subsequent version will natively implement the HDMI interface). Lastly, DisplayPort version 1.2 can be natively implemented in the device. Multiple instances of any of these video interfaces can be implemented in an FPGA or FPGA-based SoC. In the case of the RTVE it can support up to eight 1080p60 channels in a single Xilinx® Kintex®-7 K325T device. Additionally, depending on the version of the reference design the user targets, two channels of 2160p60 (4K2K) video can be supported. Fig. 4 represents the numerous video interfaces that can be supported in the RTVE reference design, and provides a complete abstracted view of the system.

Video Display Processing: Is the most significant portion of this particular system. It is in this part of the architecture where the transformation of the video content occurs through the application of key video processing algorithms. The algorithms applied include a video deinterlacer which converts interlaced video, i.e. video that is produced in interlaced sequential fields of even or odd lines from a particular source frame and need to be deinterlaced into a cohesive frame of video. One of the challenges with these sorts of algorithms is they produce unwanted artifacts known as jaggies that are particularly troublesome at low angles. The deinterlacer implemented in this reference design is motion adaptive and has been refined to reduce the amount of those sorts of artifacts. Once the video content has been deinterlaced another common operation to perform on a video stream is to up-scale or down-scale the content. The scalar contained in the OSVP can support scaling and resizing video in real time with support for input resolutions up to 4,096 x 2,160 60Hz and output resolutions up to 4,096 x 2,160 120Hz. Additionally the video display processing pipeline supports alpha blending allowing multiple streams to be overlaid on top of one another at varying levels of transparency supporting picture-in-picture or real time compositing of multiple video streams. Other processing functions available to system developers include algorithms to detect and correct film cadence which can produce spurious artifacts such as Moiré patterns and noisy text overlays in 2:2 and 3:2 cadence material. Chroma resampling is also implemented to convert source video streams to 4:4:4 chroma format as subsequent processing in the system requires 4:4:4 for signal processing. Color space conversion is implemented to convert 1080p60 streamed content from ITU-R recommendation BT.709 to ITU-R recommendation BT.2020 which is required for 2160p60 video. Color correction algorithms are also implemented to enable developers to make lift and gain adjustments to video as well as adjust brightness, saturation and hue.

Framework / Infrastructure: This system also implements the AMBA® AXI4 interconnect protocols to support AXI4 lite for controlling the parameters of each of the afore mentioned blocks as well as the CPU control plane (from the ARM®-based processing system in the case of a Zynq® 7000-based system). AXI4 memory management is implemented to support video memory management as pictured in Fig. 4. AXI4 streaming is implemented to handle the interconnect between the various algorithm blocks, e.g. the deinterlacer, scalar, color correction, etc. The system runs on Linux which is implemented on the Zynq 7000 ARM® processing system or on MicroBlaze (Xilinx's soft 32-bit microprocessor architecture) in the case of a Kintex-7-based implementation. In both the FPGA and SoC implementations these systems are controlled via a web-based GUI which runs on a web server implemented in Linux. This particular reference design targets hardware such as the Zynq®-7045 SoC-based OZ745 from Omnitek that is shown in Fig. 5.

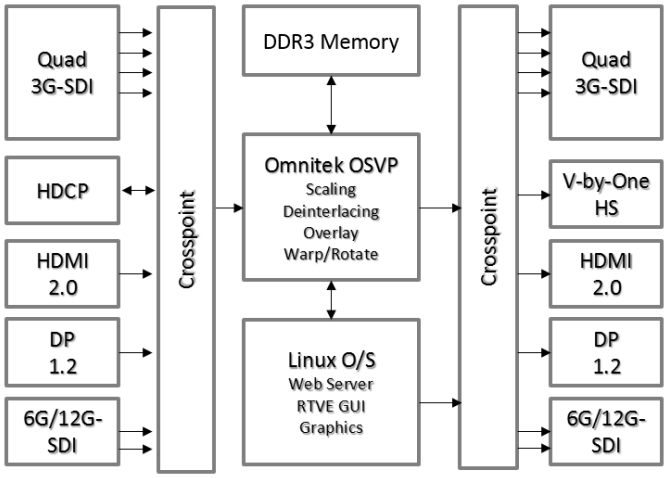

Fig. 4: Xilinx Real Time Video Engine (RTVE) based on the Omnitek Scalable Video Processor (OSVP)

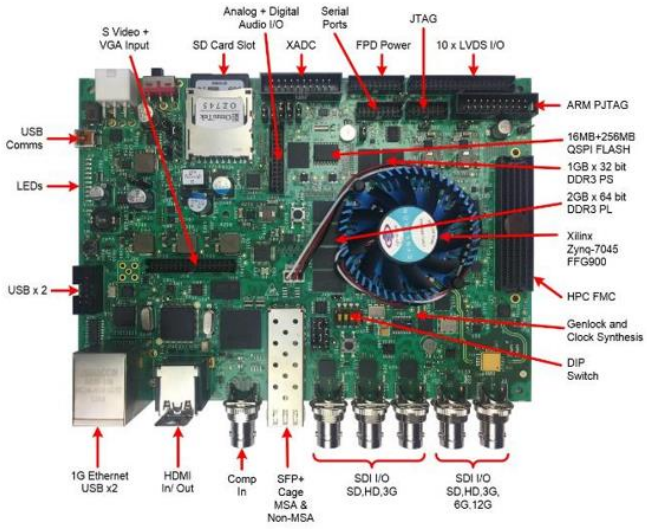

Fig. 5: Target Hardware the Omnitek OZ745 Zynq 7000-based board

VI. Software Defined Video Infrastructure

Internet streaming continues to gain momentum. In the beginning, user generated content (UGC) on services like YouTube gained popularity with a particular segment of consumers. There is increasingly a larger percentage of high production value content that is now being streamed on services like Netflix. Consumers are demanding that more of their content be delivered over the Internet or over-the-top (OTT) as it enables the consumption of media anywhere, anytime on any device. This phenomenon presents unique opportunities for the content creation community to create more targeted video programming and exploit new opportunities for advertising revenue from smaller micro-targeted demographic groups. In order to fully exploit this opportunity, however, requires more efficient ways to produce the content. It becomes cost prohibitive for content creators to address new content projects leveraging traditional capital-intensive infrastructure like television studios or outside broadcast vehicles (OBVs). With the decline in the cost of 10 Gbps Ethernet infrastructure; improvements in the ease of use in working with FPGA technology, through tools and environments like HLS and SDSoC; and the processing efficiencies FPGAs offer, which are especially deft at handling real time high resolution video, it is becoming more realistic for content creators to produce live content directly in the cloud. By being able to provision video infrastructure on an as needed basis, content creators can produce content at an unprecedented rate on a more economical basis.

Aperi Corporation [7] is an early stage technology startup in California that is working to develop a platform comprised of industry standard hardware, commercial off-the-shelf (COTS)-based servers using advanced telecommunications computing architecture (ATCA) microserver modules (Fig. 8) and a development framework that enables its developers and licensees of its technology to develop FPGA-based applications that can be provisioned and orchestrated over the Internet much the way one can create a virtual machine using the Amazon Web Service (AWS) platform-as-a-service (PaaS) model. The main difference being that Aperi's FPGA-powered applications are much more capable at handling real time high resolution content in both the compressed and uncompressed domains, are more flexible and scalable. Nothing about what Aperi has developed is specific to broadcast or professional video use cases and can be applied to any problem where power efficient computing is needed; however, Aperi possesses a strong background in this industry space and is first applying its technology to the broadcast industry.

For the purpose of this paper we will explore a single application of a JPEG 2000 encoder/decoder. This system first encodes or decodes a live video stream using the JPEG 2000 standard and encapsulates the stream using the SMPTE 2022 standard enabling real time transport of the content to the next processing node where further processing can be done. Fig. 6 highlights the relevant elements of the generic reference architecture that are implemented in this system and Fig. 7 provides additional details at a functional level.

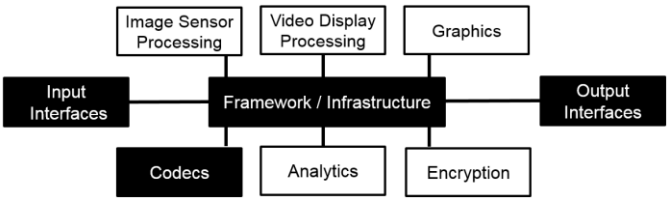

Fig. 6: Relevant elements of the general architecture implemented in the Aperi virtualized system for broadcast contribution applications.

Input/Output interfaces: SDI is the primary baseband video interface that is integrated into Aperi's hardware platform. The various processing and interface blocks together represent what Aperi refers to as an Aperi BIOS or application. In order to further process the video content over the Internet the streaming content needs to be encapsulated using the SMPTE 2022 standard, specifically the -1 variant that integrates forward error correction (FEC). This encapsulation stage occurs after the streamed content has been compressed using the JPEG 2000 codec.

Codecs: In this example JPEG 2000 is used as it is a wavelet-based codec that is visually lossless and ideal for contribution use cases where video must be kept in pristine condition before being passed on to further processing nodes.

Framework/Infrastructure: AXI4 remains the predominant interconnect in this example as well and serves as the basis for Aperi's API which enables developers to call processing functions in the various Aperi blocks using familiar RESTful APIs.

A more detailed view of the Aperi implementation on their microserver is shown in Fig 7.

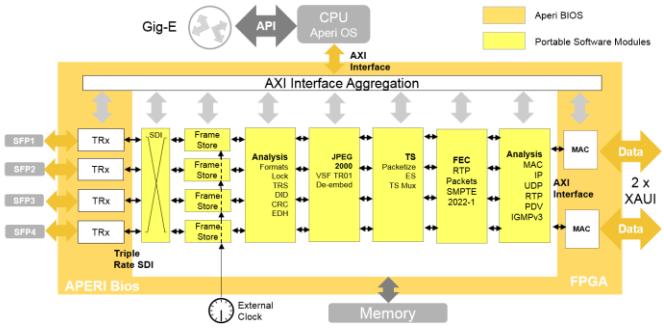

Fig. 7: Aperi SMPTE 2022 JPEG 2000 encoder application (patent pending)

Fig. 8: Aperi FPGA-based Micro Server Hardware

A software-defined platform such as that offered by Aperi means that the previously complex task of integrating system level video functions can be simplified into effectively purchasing and downloading apps to run on the FPGA. Abstraction away from the hardware results in a much faster implementation time (Aperi recently implemented an RFC4175 video packetization demonstration using Xilinx HLS code in a couple of weeks), a more scalable and agile implementation which can quickly adapt to the broadcaster's needs without expensive hardware changes, and which still benefits from the real-time video processing performance of the underlying FPGAs.

VII. Conclusion

The intent of this paper was to introduce the capabilities FPGAs and FPGA-based SoCs offer the professional video and vision equipment market. The PLD industry has reached a point of maturation on several technology vectors: from massive parallelism at economic price points relative to CPUs and GPUs; to significant advances in high-level synthesis tools and software defined environments, to heterogeneous acceleration environments for OpenCL and FPGA-based SoCs (e.g. Xilinx's SDSoC environment for FPGA-based SoCs). Through these advances in economics, power efficiency and ease of use the FPGA and FPGA-based SoCs have become a more accessible tool to software engineers not only in the professional video space, but in all disciplines where the need for efficient computational power exists. FPGAs and FPGA-based SoCs offer the equipment development community advanced tools and capabilities enabling them to maintain competitive advantages in their markets through flexibility and differentiation.

References

[1] P. Putman, "More, Better, Faster Pixels," SMPTE Motion Imaging Journal, May/June 2014, pg. 20

[2] S. Xue, "Drama in the TV Industry: A Study of New Entrants, New Services, and New Consolidations," University of Washington Libraries, May 2014, http://hdl.handle.net/1773/25957

[3] T. Edwards, A. Bechtolsheim, and W. Belkin, "Video Processing in an FPGA-Enabled Ethernet Switch," SMPTE Motion Imaging Journal, 123(2):4, March 2014

[4] http://www.logicbricks.com/Products/logi3D.aspx

[5] K. Ovtcharov, O. Ruwase, J. Kim, J. Fowers, K. Strauss, E. Chung, "Accelerating Deep Convolutional Neural Networks Using Specialized Hardware," Microsoft Research, 2/22/2015, http://research.microsoft.com/apps/pubs/?id=240715

[6] http://www.omnitek.tv

[7] http://www.apericorp.com