

Scalable Embedded Architecture for High-speed Video Transmissions and Processing

Jiří Halák, Sven Ubik, Petr Žejdl
 CESNET / CTU Prague
 Zikova 4, Prague 6 / Kolejni 550, Prague 6
 Czech Republic
 email: {halak,ubik,zejdl}@cesnet.cz

Abstract—In this paper, we present a scalable and extendable hardware architecture for processing and transfer of ultra-high-definition video over high-speed 10/40/100 Gbit networks with very low latency. We implemented this architecture in a single FPGA device. Data processing is divided between FPGA resources and an embedded operating system. The FPGA resources can be moved between various processing functions depending on the device mode. The resulting inexpensive and compact device is intended for high quality video transfers and processing with a low latency and to support deployment in education and remote venues.

Keywords—HD-SDI, video, FPGA, network communication, high-speed

I. INTRODUCTION

Video transfers are an expected driver application area of the future Internet. Picture resolution has been increasing over time. Better-than-high-definition-resolution video (such as 4K) is already used in some areas, such as scientific visualization, the film industry or even medical applications.

For the ultimate quality, required for instance in film post-production or live remote surgery transmissions, working with a signal that has not been compressed is preferable. The productivity of a distributed team can be significantly increased when the video signal can be transferred over the network in a real time to enable cooperation that is more effective. Two of the main technical issues are high-data volume and time synchronization when transferring over an asynchronous network such as the current Internet.

Currently available solutions mostly consist of multiple devices (computers, conversion boxes, sync boxes, audio boxes, etc.), which are expensive and harder to setup, increasing the logistics costs. We designed an embedded modular and scalable architecture which fits into a single mid-size FPGA device including all the required functionality and reducing the complexness and costs of this solution. We implemented this architecture and developed a device called MVTP-4K (Modular Video Transfer Platform). We have already used several prototypes in field tests to support applications in film post-production and live medical applications.

This paper is organized as follows: In Section II, we summarize the hardware requirements of our design. In Section III, we present our architecture for video transfers and processing. In Section IV, we present our prototype. In Section V, we summarize our experience with device field tests, In Section VI, we compare our solution with other available devices.

II. REQUIREMENTS

We have set the following set of requirements for our architecture:

- Video inputs and outputs SDI, HD-SDI or 3G channels
- 10/40/100 Gbit network interface or multiple interfaces
- Very small added latency
- Extendable design for additional processing such as compression or encryption
- Fit into available FPGA devices and fully implementable in one mid-size FPGA device with additional interfaces

The use of a single- and dual-link HD-SDI channel for the transmission of high definition video streams is now a common industry practice and it is specified in SMPTE 274 [1] and SMPTE 372 [2]. This includes HD (1920x1080) and 2K (2048x1080) formats. The 4K (4096x2160) signals are typically transferred in four quadrants, each in 2K format carried over a separate dual-link HD-SDI channel. 3D transmissions are typically transferred as two independent 2K or 4K channels, some require additional synchronization.

The FPGA circuit was chosen as the processing device due to its versatility that allows us to build a complete embedded solution and to host all required functionality to combine video transmissions with other functions, such as compression, encryption or transcoding.

The architecture must be scalable to allow multiple configurations based on currently available FPGA devices and interfaces, assuming the speed of communication interfaces will increase, and eventually be usable with future 100 Gigabit Ethernet networks and similar high-speed media.

We require an unnoticeable latency added to the network propagation delay for real-time applications. Unnoticeable

latency for audio/video applications is below 60 ms for un-trained audience and below 30 ms for professional audience.

III. THE ARCHITECTURE

This section describes the proposed architecture.

A. Background

In our previous work we designed and implemented a scalable hardware architecture for network packet processing [3], [4]. This architecture consists of a set of reconfigurable modules for packet processing and a communication interface. The architecture was designed for maximum flexibility and multi-gigabit speeds starting at 10Gb/s. The main processing core was designed to be fully scalable for 10/40/100Gb speeds.

We have designed an interface and developed a prototype for 40Gb/s SONET/SDH networks [5] for basic data processing and testing of 40Gb/s networks and currently we are experimenting with 100Gb Ethernet interface in FPGA devices.

B. Design Overview

Real-time processing of multi-gigabit data rates is difficult on PC-based platforms with standard operating systems not designed for real-time operation. We were looking for a real-time design that is scalable to higher data rates (such as for 8K or UHDTV2 format), higher network speeds (such as 40 and 100 Gb/s) and can be integrated with commonly requested video processing functions, such as encryption, transcoding or compression. Real-time operation means to add a very low latency to a network delay and enable true live experience. This design is fully automated and all embedded in a single FPGA device.

The embedded architecture for real-time video transport and processing is based on our previous work. The core is the scalable architecture for network packet processing [3], [4] designed especially for Ethernet networks. This whole architecture operates at network clock domain of attached network interface and can be used for various modular data packet processing. Since video signal consists of special packets, we can make a simple conversion to transport the video packets to a network clock domain and back. This way we can use a network packet processing architecture for video packet processing.

When we convert data packets from network clock domain to video clock domain certain mechanisms need to be used. Ethernet Network is an asynchronous network, on the other hand, HD-SDI is a synchronous channel thus an advanced techniques for synchronization of data packets crossing from network domain to video domain are required [6].

Address range of all processing modules, routes and hardware configuration registers is mapped to an embedded processor logic bus (PLB) address range using a simple bus bridge of our own design. An embedded processor can be a

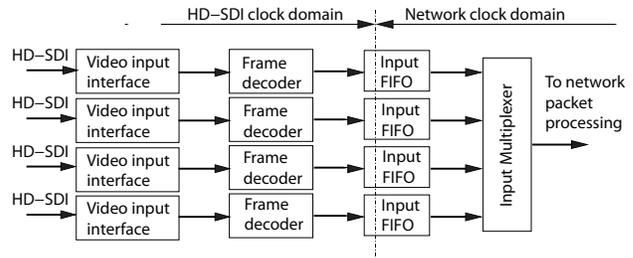


Figure 1. Input video processing and connection to packet processing for 4 channels.

dedicated Power PC processor or soft-core Microblaze [7] processor. An embedded processor is running a customized Linux distribution and all peripherals are managed by Linux drivers or dedicated software tools. The embedded Linux distribution also provides all means of communication with a device, such as ssh server, web server, display and keyboard controllers and eventually can also handle 10/100/1000 Mbit and multi-gigabit interfaces. The Multi-gigabit Ethernet Network Interface for an embedded processor is described in a subsection III-F.

C. Video Processing Modules

Video processing modules do a conversion between video and network packets, allowing video data to be processed in the network packet processing core (section III-D). There are two video processing modules, the input and output module, shown in Figures 1 and 2.

The input module consists of the video input interface and the frame decoder. The video input interface implements low-layer communication with the HD-SDI equalizer chips through Rocket IO channels and the frame decoder extracts video packets, converts them to network packets and attaches headers with video format parameters.

The output module consists of the video frame generator and the video output interface. The frame generator receives network packets and generates valid image to the video output interface based on information contained in network packet headers.

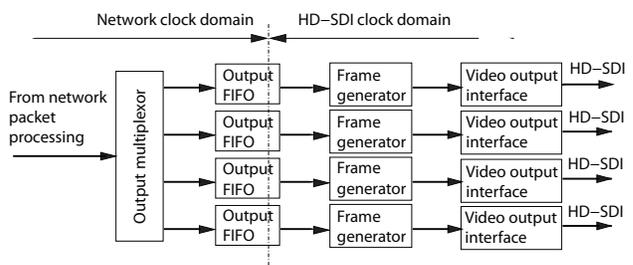


Figure 2. Output video processing and connection to packet processing for 4 channels.

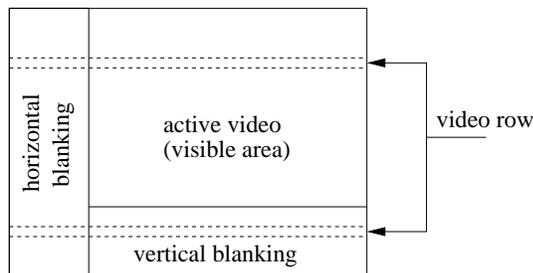


Figure 3. Video frame structure transported through HD-SDI channel

A video packet includes a video pixel row with specified headers and control characters. We use a dual-port memory as a packet FIFO to cross clock domain boundaries. The video processing modules are located in the HD-SDI clock domain and the network packet processing modules are located in the network clock domain. The example configuration in Figures 1 and 2 includes four HD-SDI channels. The channels are independent and can be added freely just with a simple modification of the channel multiplexer. This operation can be completely parameterized.

The HD-SDI interface has a bit rate of 1.485 Gbit/s [8] but not all data needs to be transferred. Video rows include blanking areas (horizontal blanking interval) and a video frame includes blanking video rows (vertical blanking interval). The whole situation is illustrated in Figure 3. Blanking areas can contain some secondary information such as audio, encryption or video format specification, which we can choose to transport or not. When we strip video packets of those blanking intervals, we get a bit rate between 1 Gb/s and 1.3 Gb/s depending on a picture resolution and frame rate. This means that the 10 Gbit Ethernet network can transfer up to eight HD-SDI video channels and with some video formats even including additional data, such as audio or encryption information.

The example bitrates of eight channels of selected video formats stripped of blanking intervals are summarized in Table I. The 30fps HD formats can be still transferred, but image crop must be applied.

TABLE I
VIDEO FORMATS BITRATES

Format	Bitrate eight channels (Gb/s)	Bitrate one channel (Gb/s)
2K/24	8,7	1,08
1080/24	8,2	1,025
1080/25	8,5	1,06
1080/30	10,4!	1,3
720/50	7,6	0,95
720/60	9,1	1,14

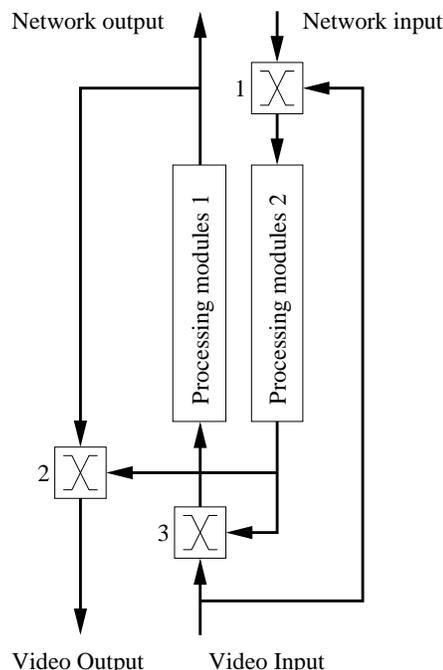


Figure 4. Schematic of interconnections in the processing core.

D. Network Packet Processing Core

The main processing core consists of two sets of processing modules. We have extended our original architecture [3], [4] with the video interface and video processing modules described in section III-C. The network packet processing core is divided into two parts. A set of switches can be arranged to allow a packet flow between the network and video domains in several ways. A schematic of this interconnection is shown in Figure 4. The following configurations are possible:

- From network input to network output through switch 1, processing modules 2, switch 3 and processing modules 1. All processing modules are dedicated for network to network packet processing.
- From network to video, full-duplex, one set of processing modules for each direction. From network input through switch 1 and processing modules 2 to video output. From video input through switch 3 and processing modules 1 to network output.
- From video input to video output through switch 3, processing modules 2, switch 3, processing modules 1 and switch 2. All processing modules are dedicated for video packet processing.

Data stream processing modules are inserted directly to the packet stream. Every processing module works as an individual processing unit. The advantage is that modules can occupy different FPGA devices. When we need to implement a complex module such as video encoder/decoder,

we may find more suitable to use more FPGA devices. For this purpose, the architecture is designed to relocate a packet stream through a high-speed FPGA ports to another device and make a cross-device interconnection of processing modules. Both options are shown in Figure 5. Option A: Modules are connected in a single device. Option B: Module interconnection crosses multiple devices over a high-speed interface.

E. Processing latency

The concept of intra-frame processing of video packets as network packets enables extra low latency of video processing and transmission. This opens a way to truly real-time collaboration support. The processing design itself has a low latency under 1 ms. Video packets are buffered only when synchronizing from the network asynchronous domain to the video synchronous domain. However, low delay variation in the network is required to allow design latency under 1 ms. In lower quality networks the buffering level needs to be obviously increased. The extreme cause is a single frame buffer adding a maximum latency of about 30 ms.

F. Network Interface

High-speed network interface consists of hardware and software parts, which are controlled by an embedded operating system. Incoming packets containing video data are sent to output video processing module, on the other hand network management packets such as ARP or ping are sent to software network driver. Outgoing packets have two different sources, packets containing video data are sent from input video processing module and network management packets are sent from software network driver.

The block diagram of the network interface is shown in Figure 6. Incoming packets are classified in the packet classifier and distributed between video processing modules (VPM) and RX FIFO. Outgoing packets are sent from VPM or from TX FIFO. Because there are two paths producing packets, packet multiplexer is included in the design. It is

multiplexing packets in a round-robin fashion. RX and TX FIFO are connected to the CPU through the processor local bus. Therefore, both memories are accessible from software. The packet classifier is also connected to the CPU, but the connection is not shown. The CPU is embedded inside FPGA either.

The packet classifier contains memory for four classification rules. Each rule can be marked as going to the VPM and/or going to RX FIFO. The memory is configured from software. Currently we use three rules. The first rule is marked as going to the VPM and the other two rules are marked as going to RX FIFO. The fourth rule is not used and is reserved for future use.

The rules are as follows:

- Rule for UDP packets containing video data.
- Rule for ARP packets for address resolution.
- Rule for ICMP packets (ping command).

The software part is based on embedded Linux. Network interface is accessible through the Linux TUN/TAP driver [9], which provides packet reception and transmission for user space programs. The program controlling network hardware is running as a daemon in the user space, and through TUN/TAP driver provides a new network interface. This new interface behaves like an ordinary network interface such as eth. Therefore, all networking services are available through this interface.

IV. MVTP-4K PROTOTYPE

We have designed and build a MVTP-4K (Modular Video Transfer Platform) device which implements proposed architecture and validates it in field tests. The MVTP-4K is a

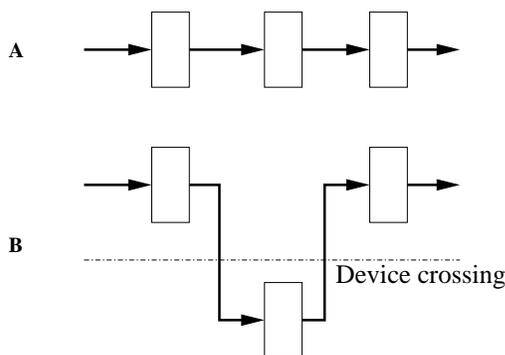


Figure 5. Processing modules

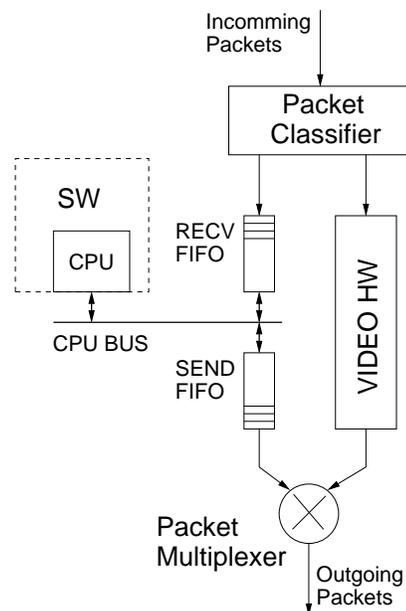


Figure 6. Ethernet Interface Block Diagram



Figure 7. System architecture overview.

portable device of our own construction for transmission of multiple high-definition video streams including 4k, 2k and HD over a 10 Gigabit Ethernet Network. The device consists of a main FPGA board with 8 HD-SDI video interfaces and one 10Gbit Ethernet interface. Brief structure is shown in Figure 7. The device supports all 4K, 2K and HD resolutions and all corresponding frame rates. 3D transmissions are also supported. Because the data processing is based on data packet processing we can even transport data not fully supported without the need of unpacking them from video signal. This allows us to transport audio data or encryption data embedded in the video stream.

We have chosen a Xilinx Virtex FPGA series because it provides all building blocks and tools required to implement our architecture. The prototype is based on an extended platform for network packet processing called MTPP [3]. Whole architecture fits to a mid-size FPGA device Virtex 5 series XCV5LX110T. We have experimentally confirmed that the device adds a low latency of less than 1 ms.

Mid-size Xilinx FPGAs can be obtained under 3000\$ a piece in a small quantities. For advanced hardware functions such as encryption or encoding, a larger FPGA or a second mid-size FPGA is required.

V. PRACTICAL EXPERIENCE

We have demonstrated our system at the Cinegrid 2009 and Cinegrid 2010 workshops. The aim was to demonstrate that such technology can enable real-time remote cooperation of a distributed team and thus increase productivity. In the first event, a stream of uncompressed 4K video was

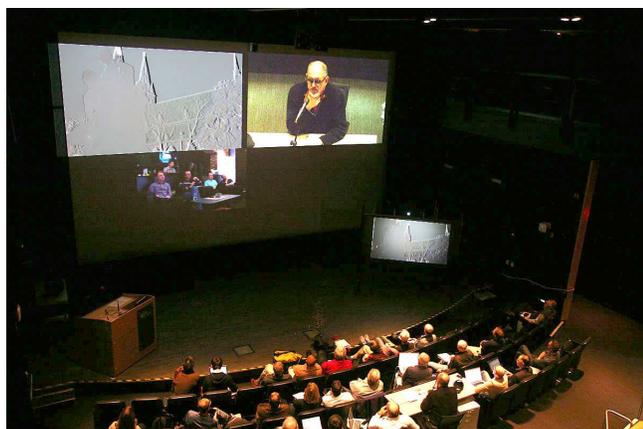


Figure 8. Practical use of the technology at Cinegrid 2010 event

transferred from the Barrandov studios in Prague to the venue in San Diego over a distance of more than 10000 km to perform remote color grading in a real-time. In the second event, a stream of 3D 2K video was transferred from the UPP Company in Prague to the venue in San Diego to perform remote real-time postproduction processing of 3D images. The 3D grading performed at the venue with the signal transferred by our device is illustrated in Fig.8.

In order to evaluate the system suitability for e-Health applications, we transferred several surgical operations from the daVinci Surgical System [10] which produces HD stereoscopic signal in 1080i format. The picture quality was subjectively approved by invited medical experts as suitable for highly illustrative student training or presentations of surgical procedures on symposia.

VI. RELATED WORK

There are several commercial products, which allow transport of SDI, HD-SDI or 3G channels over network.

Net Insight's [11] Nimbra 600 series switch can transport 8x HD-SDI or 3G SDI channels over an SONET/SDH network. There are several commercially available solutions for transport of compressed 4K video over the Internet, for example NTT Electronics [12] ES8000/DS8000 4K MPEG-2 encoder/decoder complemented with NA5000 IP interface unit and intoPIX's [13] system of PRISTINE PCI-E FPGA boards and JPEG 2000 IP cores.

UltraGrid from Laboratory of Advanced Networking Technologies is a software for real-time transmissions of high-definition video [14]. This solution is a fully software based and requires dedicated PC with specialized hardware.

The architecture and design described in this article differs in that it is a hardware solution fully scalable to higher speeds. The number of video and network interfaces is parameterized and can be easily extended. The FPGA enabled parallelism allows our architecture to process several video channels at once and to transfer every video format contained in SDI, HD-SDI or 3G interface. The architecture is designed to be embedded to a single FPGA device but some larger processing modules can be relocated to other FPGA devices. Our design has a very small added latency around 1 ms that enables a true real-time distributed team cooperation.

VII. CONCLUSION

We have extended a scalable architecture for network packet processing [3], [4] by video interfaces options. The resulting architecture is designed to process or transport video data over an asynchronous network with very low added latency. The design enables true real-time distributed team cooperation. The real-time team cooperation was demonstrated in several applications in the cinema industry and e-Learning in medicine. The architecture also fulfills the hardware requirements that we set and we successfully

implemented this architecture in a single FPGA device and presented its capabilities.

ACKNOWLEDGMENT

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research intent MSM6383917201 Optical Network of National Research and Its New Applications.

REFERENCES

- [1] "1920 x 1080 Image Sample Structure, Digital Representation and Digital Timing Reference Sequences for Multiple Picture Rates." Society of Motion Picture and Television Engineers., 2005.
- [2] "Dual Link 1.5 Gb/s Digital Interface for 1920 x 1080 and 2048 x 1080 Picture Formats." Society of Motion Picture and Television Engineers., 2009.
- [3] J. Halak and S. Ubik, "MTPP - Modular Traffic Processing Platform," in *12th IEEE Symposium on Design and Diagnostics of Electronic Systems, DDECS 2009*, Liberec, Czech Republic, April 2009, pp. 170–173.
- [4] J. Halak, "Multigigabit network traffic processing," in *Proc. The International Conference on Field Programmable Logic and Applications, FPL 2009*. Prague, Czech Republic: IEEE Computer Society, August 2010, pp. 521–524.
- [5] J. Halak, S. Ubik, and P. Zejdl, "Data stream processing for 40 Gb/s networks," in *Proc. Fifth International Conference on Digital Telecommunications, ICDT 2010*. Athens/Glyfada, Greece: IEEE Computer Society, June 2010, pp. 149–152.
- [6] —, "Receiver synchronization in video streaming with short latency over asynchronous networks." Vienna, Austria: IEEE Computer Society, April 2010, pp. 403–405.
- [7] MicroBlaze Soft Processor Core. (Last accessed: July, 2011). [Online]. Available: <http://www.xilinx.com/tools/microblaze.htm>
- [8] "1.5 Gb/s Signal/Data Serial Interface." Society of Motion Picture and Television Engineers., 2008.
- [9] Universal TUN/TAP device driver, Linux Kernel Documentation. (Last accessed: July, 2011). [Online]. Available: <http://kernel.org/doc/Documentation/networking/tuntap.txt>
- [10] The da Vinci Surgical System, Intuitive Surgical. (Last accessed: July, 2011). [Online]. Available: <http://www.intuitivesurgical.com/products/faq/index.aspx>
- [11] Net Insight AB. (Last accessed: July, 2011). [Online]. Available: <http://www.netinsight.se>
- [12] NTT Electronics. (Last accessed: July, 2011). [Online]. Available: <http://www.ntt-electronics.com>
- [13] intoPIX. (Last accessed: July, 2011). [Online]. Available: <http://www.intopix.com>
- [14] P. Holub, L. Matyska, M. Liska, L. Hejtmanek, J. Denemark, T. Rebok, A. Hutanu, R. Paruchuri, J. Radil, and E. Hladk, "High-definition multimedia for multiparty low-latency interactive communication," *Future Generation Computer Systems*, vol. "22", no. "8", pp. "856 – 861", October "2006". [Online]. Available: "http://www.sciencedirect.com/science/article/pii/S0167739X06000380"