Using mobile processors for general purpose industrial signal processing

Hans-Joachim Gelke
Zurich University of Applied Sciences
Institute of Embedded Systems
Winterthur, Switzerland
hans.gelke@zhaw.ch

Tobias Kammacher
Zurich University of Applied Sciences
Institute of Embedded Systems
Winterthur, Switzerland
tobias.kammacher@zhaw.ch

Matthias Rosenthal
Zurich University of Applied Sciences
Institute of Embedded Systems
Winterthur, Switzerland
matthias.roenthal@zhaw.ch

Amin Mazloumian
Zurich University of Applied Sciences
Institute of Embedded Systems
Winterthur, Switzerland
amin.mazloumian@zhaw.ch

The boom of mobile devices in recent years has brought efficient and cost-effective multicore processors to the market with perfect characteristics to realize a variety of different applications for computer vision, general signal processing, video encoding or network streaming.

This paper describes how to use so called “mobile processors” designed for tablet computers and smartphones in industrial designs. Various suppliers offer computer modules with extended temperature range, based on processors powering smart phones or tablets such that these devices may also be used in industrial applications. The module suppliers act also as the interface between the processor manufacturer and industrial user by supplying drivers and board support packages.

The current generation of mobile processors incorporates at least quad core processors with a power efficient operation, GPUs with up to 256 cores, High Efficiency Video Coding (HEVC), Ultra High Definition (UHD) capable MIPI interfaces and audio/video processing blocks. GPUs may be used for high-end graphics or general-purpose signal processing applications by using APIs like Open Compute Language (OpenCL) or Open Graphics Language (OpenGL). The powerful multicore architecture allows real-time processing entirely through software. Hence, mobile processors may replace FPGAs in certain applications.

Besides the interfaces most listeners are familiar with, mobile processors are equipped with so called Mobile Industry Processor Interfaces (MIPI), which serve to connect LCD Displays and camera modules. To adapt to older, more established interfaces, a variety of IPs and silicon is available. The author shares experiences with available IPs and MIPI driver development.

HEVC in mobile processors allows to drop the bandwidth to half or to double the video quality by maintaining the same bandwidth. The Ultra-HD capable video encoders and decoders of mobile processors provide state of the art HEVC in hardware. The specifics of HEVC are shortly discussed and video quality measurements are shared.

In order to access the many video processing blocks of the mobile processor we will describe a software framework for video processing is necessary. We show examples for Linux and Android.

The authors made good experiences with mobile processors and want to share the outcomes with the community.

Keywords—Mobile Processor, System-on-Chip SoC, Video Processing, H.265, HEVC, V4L2, GPU, UHD, FPGA, System-on-Module SOM, Camera Serial Interface CSI, MIPI, HDMI, Streaming

I. INTRODUCTION

The research field of the Institute of Embedded Systems (InES), an entity of the Zurich University of Applied Sciences (ZHAW), is related to real time signal processing in FPGA and
GPU in conjunction with wired and wireless real time transmission over networks.

InES projects include signal processing for video and audio, video encoding, decoding, forward error correction, real time data acquisition and video streaming over wide area networks.

In the recent years InES research projects were FPGA centric, however with the arrival of more high performance and low power CPUs and GPUs, many tasks, which were accomplished with FPGAs in the past, may now be done with high performance processing units as they can be found in mobile devices like smart phones and tablet computers.

This paper shows on two examples how general purpose signal processing and video signal processing can benefit from these powerful low cost processors.

As an example, Figure 1 shows a signal-processing path of a video encoder/decoder and recorder. At the beginning of the signal processing pipeline, there is a suitable, possibly latency free high speed I/O. If the signal is video and is sourced from a camera, color space conversion is required. Adapting to different resolutions is accomplished by scaling. Video encoders provide transmission and storage at low bandwidth. Some applications require also Forward Error Correction and Encryption in a networked environment.

In the past, the above described system required the development of an own ASIC or an FPGA. However since signal and video processing is part of the standard repertoire for smartphones and tablet processors, one might think about replacing FPGAs or special ASICs with processors found in smartphones or tablets.

II. MOBILE PROCESSORS

There are many processor products on the market, which are equipped with GPU and video processing, but are not solely designed for mobile applications; however, their characteristics are similar to processors designed specifically for smart phones. This paper will refer also to those kind of processors as “mobile processors”.

Current state of the art mobile processors provide up to eight 64-bit ARM CPU cores, 256 Core Graphics Processing Units supporting OpenGL 4.5, OpenGL ES 3.1 and AEP.

Low latency, high bandwidth Mobile Industry Processor Interfaces (MIPI) connect high-resolution cameras and displays.

MII interfaces may also be used as general purpose streaming interfaces and to feed a high-resolution video screen or are used for the HDMI input. Dedicated video processing units for scaling and format conversion are suited for 4K ultra high-resolution video.

For video encoding and decoding the latest H.265/HEVC video coding standard is supported [14].

Connectivity interfaces allow the easy integration of WLAN, Ethernet encapsulation, USB 3.0 and 4G wireless.

Figure 2 shows a typical mobile processor architecture. Refer to section VIII for a brief overview of the vendors.

III. PROCESSOR CORES

Typically, todays CPUs are multi GHz, 64-bit quad core ARM-A57. There are manufacturers, which use their own 64-bit CPU architectures. Some processors enclose up to two quad cores, functioning as so called “big.LITTLE architecture”. The two quad cores often consist of the bigger ARM-A57 and the smaller ARM-A53, where usually only one set of quad cores is active at a time. For saving battery life, the little CPU (typically ARM-A53) is active when low performance is required. If high performance is required, the system switches to the big CPU (typically ARM-A57).

High-speed interfaces include up to four lanes PCIe or USB3.0. Often pins are shared between PCIe and USB3.0 such that both cannot be used at the same time. Ethernet Media
Access Controllers (MAC) and Physical Interfaces are usually not integrated into Mobile processors. Ethernet MACs must connect via PCIe or USB.

IV. GRAPHICAL PROCESSING UNIT USED FOR GENERAL PURPOSE COMPUTING

Graphics Processing Units (GPUs) are many-core processors with high computational power and data throughput. The computational power has helped researchers and companies to achieve speedups from one to two orders of magnitude in their applications. The algorithms and computational-heavy tasks that can be divided into many time-independent subtasks can benefit most from the computational power of GPUs. In contrasts, the tasks with many synchronization parts are least optimized when utilizing GPUs. The wide range of GPU applications includes many-particle simulations, computational chemistry, gene sequencing, and medical imaging [2].

The computational power of GPUs is available for general purpose computing through parallel programming platforms [3], [4]. The two most prominent parallel programming platforms are CUDA (Compute Unified Device Architecture) and OpenCL (Open Computing Language). GPU manufacturer Nvidia developed the CUDA platform to allow developers to utilize the computational power of their GPU products [5]. The CUDA programming model lets developers write C code and make executables using Nvidia’s proprietary compiler. The executables perform thousands to millions of parallel task units, i.e. threads, which run on the hundreds of cores of Nvidia GPUs. The OpenCL platform on the other hand, is a royalty-free standard developed by Khronos Group for parallel programming of diverse processors [6][7]. An OpenCL source code executes on many types of CPUs, GPUs, DSPs, and FPGAs. The source code should be developed in OpenCL C which is the standard C99 adapted to fit the device model in OpenCL. Although an OpenCL source code natively executes on many devices, it does not mean that the same source code is optimal for all the devices. As one might expect, for Nvidia GPUs, it takes minimal effort to develop an optimal source code using CUDA with reduced portability rather than OpenCL.

In our application, an already-existing signal-processing product for a multi-core CPU platform was enhanced by offloading parallel floating-point calculations to Nvidia GPUs using CUDA. More specifically, computationally demanding floating point signal filters had to be applied on hundreds of 100 KHz input signals and generate output at very low latencies (less than 10 micro seconds). Performing signal-processing filters on many independent input signals has an extremely parallel nature, which makes GPU enhancement for the calculation an obvious choice. However, in latency-critical applications the main concern in using GPUs is the bi-directional data transfer between host memory to GPU memory. A typical GPU parallel program copies input from the host memory to the GPU memory, then the calculation is done with many parallel threads in GPU, and finally the results are copied back to the host memory. Such memory transfer method is a latency bottleneck. Because the required data for all working GPU threads transfer all at once in big chunks it requires synchronization between threads. In our application we could use the Nvidia unified memory scheme and fulfill the latency requirement because the size of signal sample data was relatively small. In this scheme, GPU threads can fetch byte-size data from host memory independently in a transparent way (Zero-copy method) (see Figure 4). Compared to the CPU-only implementation, using GPU for parallel floating point computations not only increased the computational power by more than an order of magnitude, but also significantly decreased the total cost. Because, thanks to the computational power of CUDA cores much cheaper CPUs were sufficient.

![Figure 4: CPU, GPU Architecture](image)

V. MULTIMEDIA INPUT AND OUTPUT

For video input, the Mobile Industry Processor Interface (MIPI) Camera Serial Interface (CSI) standard [18] is supported. On modern mobile processors, there are up to three four-lane MIPI Camera serial interfaces available. Each lane supports 1.5 Gbps, which is enough bandwidth for 4K video resolution. Usually these interfaces are designed to connect to cameras, however can be used for HDMI and Serial Digital Interface (SDI) video. For HDMI input, the industry provides converter chips, converting 4K HDMI to CSI [18]. For broadcast standard SDI Video input, FPGA IPs are available [12].

The mobile processors CSI interfaces are directly coupled with a video processor block, which may provide color space conversion from RGB to YUV, picture in picture, graphics overlay and scaling into different aspect ratios and resolutions.

Most processors provide a 4K capable HDMI 2.0 output. For Systems, which require a broadcast quality SDI output, the processor’s MIPI DSI outputs can be used. Also here, the established FPGA manufacturers offer MIPI CSI converters, as well as SDI outputs as FPGA IP blocks.

Many multimedia applications nowadays require at least eight channels of audio, which needs to be fed in and out of the processor. Most processors support at least eight channels of audio via a multiple of I2S or TDMI interfaces.

VI. HIGH EFFICIENCY VIDEO ENCODING AND DECODING (HEVC)

Video is supported at 4096 × 2160 pixels (4K) at a frame rate of 60 fps. At the lower HD resolutions, the maximum frame rate is even 120 fps. The hardware accelerated video encoders and
decoders support the established H.264 video encoding standards and the new High Efficiency Video Encoding (HEVC or H.265) standard [14] in 4K and 10-bit decoding quality, supporting main10 profile at H.265, as well as main and high profile at H.264.

HEVC is based on improved compression algorithms, which reduce the video data rate by half, compared to H.264. Figure 6 compares the bandwidth requirements between JPEG 2000, H.264 and HEVC.

HEVC outweighs its previous standards providing the following improvements:

- More flexible partitioning, from large to small partition sizes (see Figure 5)
- Greater flexibility in prediction modes and transform block sizes
- More sophisticated interpolation and de-blocking filters
- More sophisticated prediction and signaling of modes and motion vectors
- Features to support efficient parallel processing.

Figure 5: H.264 Macro block partitioning (left) vs. HEVC Quad Tree partitioning (right) [20]

HEVC can provide a better compression (up to 50%), at the cost of potentially increased processing power.

An encoding algorithm producing an HEVC compliant bit stream would proceed as follows:

- Partitioning each picture into multiple units
- Predicting each unit using inter- or intra-prediction, and subtracting the prediction from the unit
- Transforming and quantizing the residual (the difference between the original picture unit and the prediction).
- Entropy encoding the transform output, prediction information, mode information and headers.

Figure 6: Bandwidth requirements compared

Figure 7: HEVC Encoding, Decoding Pipeline

The two leading video coding expert groups, namely ITU-T and ISO/IEC, in 2012 released the HEVC standard [14]. The HEVC standard is superior to its widely used predecessor H.264/MPEG-4 AVC mainly in compression capability (up to 50%) and data-loss robustness [15], [16].

Figure 8 shows the relationship between video quality and bitrate when using either the H.264 or the HEVC codec (see Figure 8. The two video sequences ("Campfire Party" and "Tall Buildings") provide raw files in 4K resolution by Shanghai Jiao Tong University (SJTU) [8]. The video quality is represented by the Peak Signal-to-Noise-Ratio (PSNR), which compares the quality of the video after encoding-decoding to the original file.

If we look at a single video and compare the bitrate for the two different CODECs at a fixed quality (PSNR), we find that the bitrate is approximately halved when using the HEVC CODEC instead of the H.264 CODEC. During testing, we found a PSNR of 35 dB to be necessary for the quality to be subjectively acceptable (note that this is not an objective measure, but depends on various factors like the quality of the display and personal preferences). In order to achieve this quality the "Campfire Party" video needs to be encoded with a bit rate of 15 Mbit/s when using the H.264 CODEC. When using the HEVC CODEC on the other hand, a bit rate of 8 Mbit/s is sufficient.
VII. LINUX AND ANDROID ENVIRONMENT

Mobile processors provide a large number of application-specific hardware blocks and accelerators. Access to them is usually implemented as part of the Linux or Android operating system. While the Linux OS provides easy access to each part of the operating system, programming on Android is more restricted. For basic functionality on Android there are interfaces given in the Java programming language. However, for accessing low-level functions (e.g., direct hardware-acceleration for video CODECs) a native-code language is necessary. For this purpose, the Android NDK (Native Development Kit) allows the use of C/C++ libraries and source code inside an Android app [17].

In terms of product development, a mobile processor provides a complete operating system usually with extensive functionality. This allows for rapid implementations and testing. On the other hand, the system may be overly complex and thus only running at reduced performance. For this reason, and in order to restrict the access rights of the end user, it may be necessary to slim down the development system when going into production.

On the example of a video streaming device, we will explain how to access video processing components in the Linux OS. Mobile Processor manufacturers usually develop a specific Linux version for their hardware based on a vanilla (i.e., unmodified, general-purpose) Linux kernel. The given method is based on a mobile processor, which provides access to the video CODEC as well as a video input interface (MIPI CSI-2) [18].

A simplified view of the components involved in accessing the video CODEC blocks on a mobile processor from a user space application is given in Figure 9. A custom-built multimedia application initiates the access after receiving a corresponding user input. This application can for example be implemented using the GStreamer multimedia framework.

GStreamer is a highly flexible framework for creating applications involving audio / video processing and contains bindings to hardware CODEC accelerators. These bindings are implemented in the OpenMAX cross-platform API [6]. It is important to note that this has to be provided by the silicon manufacturer. In this example in the form of a hardware-specific GStreamer plugin.

The OpenMAX integration also represents the link between user space and kernel space, since the counterpart to the GStreamer OpenMAX plugin is the Bellagio OpenMAX IL (Implementation Layer). Access to the hardware blocks is performed either by the CPU or by means of a DMA (Direct Memory Access) block. This accelerated access requires a hardware-specific driver.

Another interesting application is video input. In modern Linux kernels the Video-for-Linux-2 (V4L2) subsystem handles (amongst others) video input hardware. The following paragraph explains the software requirements for connecting a video input device (e.g., a camera) to the video-input interface of the host processor. The video-input interface can be e.g., MIPI CSI-2 with data rates up to four Gbps on four differential lanes.

Back to the software requirements, the V4L2 subsystem requires three different drivers running on Linux on the host processor. A 'camera host' driver, which controls the video input interface of the host processor. Furthermore a 'camera sensor' driver, which connects to the external camera device (or bridge chip) and controls its functionality. Lastly, a DMA driver is required for controlling the data transfer between the image sensor and the host processor.

The software components mentioned in this article (e.g., the Linux OS, the multimedia APIs or the GStreamer multimedia framework) are all open source software and published under a certain kind of open-source license. These licenses allow the right to study, change and distribute the software. Nonetheless, certain licenses may impose restrictions on the user of the software, e.g., copyleft licenses. Developing in a collaborative manner can yield great results, but the inclined reader is advised to get familiar with the exact licensing terms of particular interesting software projects.

Figure 9: Accessing video CODEC accelerator from Linux user space application [10]
VIII. MOBILE PROCESSOR MARKET PLAYERS

Several Semiconductor vendors offer attractive solutions. Qualcomm’s latest Snapdragon-820 processor is powered with a 64-bit Kryo 14nm quad-core CPU. The Kryo core is a Qualcomm custom designed architecture based on ARMv8 technology. The Adreno 530 GPU supports OpenGL ES 3.1 and OpenCL2.0. The presence of a X12 LTE modem, 802.11ad Wi-Fi and a Hexagon DSP processor shows that the Snapdragon-820 is specifically designed for mobile applications.

Nvidia’s latest TX1 processor is not exclusively designed to be used in mobile applications, which shows on the features and power dissipation. Nvidia strength is its 256 core 1024 GFLOPS GPU which is supported by OpenGL, OpenCL and the proprietary CUDA programming environment. The TX1 is a typical big.LITTLE architecture powered by four 64-bit 1.9 GHz ARM Cortex A57 big cores and 4 ARM Cortex A53 little cores and a Neon FPU.

Intel’s X7-Z8700 is based on x86 Atom Quad Core architecture and is used in Microsoft Surface Tablets.

Allwinner is a Chinese company whose A64 processor offers low cost ($5) with a 64-bit ARM Cortex A53 quad core and GPU.

IX. INDUSTRIAL BOARD COMPUTERS

For smaller projects, it could be difficult to get direct support and literature from some vendors of mobile processors. However, several industrial board computer manufacturers offer attractive System-on-Module (SOM) solutions, equipped with processors from Qualcomm, NVIDIA, Intel and Allwinner.

Avionic Design offers a module, powered by the NVIDIA TK1 and includes 2GB of DDR3 RAM and 16GB eMMC. Avionic Design also provides software packages in the form of a Linux device tree with CSI drivers for several cameras and Toshiba HDMI input.

Toradex just announced a similar module with TK1 and up to 4 GB DDRAM3 and 16GB eMMC.

Intrinsic offers the Open-Q 820 System on Module, which features the latest Snapdragon 820 processor from Qualcomm.

Theobroma systems offers a module with the Allwinner A80 Octo-Core and Cortex-A7/A15 with up to 8 GB DDR3 and up to 16Mbit NOR Flash and up to 128GB eMMC solid state disc on module [19].

For its latest TX1 processor, NVIDIA now offers the Jetson-TX1 module directly. The Jetson-TX1 module offers 4GB of DDRAM4, 16 GB Flash, IEEE 802.11ac 2x2 Wi-Fi transceiver and 1000 BASE-T LAN.

X. CONCLUSION

The Institute of Applied Sciences at the Zurich University of Applied Sciences is currently using mobile processors in three of its research projects for signal processing, control and user interface. We found that in certain applications, time consuming and custom specific FPGA development, may replace real-time software on mobile processors. However, one has to consider that the developer must deal with a multi core environment. An operating system, possibly with a real time patch, is required. A suitable software environment to program GPU and DSP is necessary. Special drivers are necessary for many of the hardware accelerated blocks like video codec, scaling and fast I/O. New standards for fast I/O like MIPI CSI are still a new field and new in applications other than video.

One main advantage of using the processors in a Linux environment is the huge availability of software from the open source community. Support by the module manufacturers is excellent; however, the access to more specific information directly from the CPU vendors is sometimes difficult to get by.

REFERENCES

[1] Nvidia Corp. X1 preliminary Datasheet NVIDIA Corp. 2015
[19] https://www.theobroma-systems.com/a80-a7-tech-specs