Technology Transfer in Computing Systems

D3.12: Individual TTP12 abstract

Project no.:	609491
Funding scheme:	Collaborative project
Start date of the project:	1st September 2013
Duration:	36 months
Work programme topic:	FP7-ICT-2013-10
Deliverable type:	Report
Deliverable reference number:	ICT-609491 / D3.12
WP and tasks contributing:	WP 3 / all
Due date:	31/01/2015
Actual submission date:	11/02/2015
Responsible Organization:	TUB
Dissemination Level:	Public
Revision:	1.0

TETRACOM D3.12: eGPU accelerated HEVC/H.265 video decoder

Mauricio Alvarez-Mesa, Chi Ching Chi, Ben Juurlink, Technische Universität Berlin, Germany / Georgios Keramidas, Iakovos Stamoulis, George Siridopoulos, Think Silicon Ltd. Greece

HEVC/H265 is a new video coding standard that provides higher video quality with reduced bandwidth compared to its predecessors. When performing video decoding and playback in embedded and mobile devices power consumption is the main design constraint. Several solutions are being used and proposed in the industry and academia for the execution of video decoding applications on mobile platforms, however all of them have significant shortcomings. Software only solutions with multicore CPUs are very flexible but are not power efficient. A software only solution using the GPU with GPGPU programming models such as OpenCL can be potentially more power efficient than CPU only solutions, but the OpenCL execution model does not match well with video decoding applications. Currently, the most widely used solution by the industry is to perform video decoding using fixed-function logic; this results in very low very power consumption and small silicon area per codec, but it is not flexible, and has an area overhead when multiple codecs need to be supported. In this project we designed an alternative architecture and programming model for video decoding and playback applications on mobile systems that combines hardware and software modules. The proposed model results in a flexible, and at the same time, power and area efficient solution.

H.265 Video Decoder Integrated into an Embedded GPU

In the proposed design the host CPU is only used as a general controller of the video player application, and the GPU is responsible for video decoding and rendering. The main design requirements were to achieve the target performance with a flexible and programmable solution, and to reuse as much as possible the existing GPU modules. Several optimization techniques have been proposed in order to achieve the design objectives:

- The integration of a multi-standard hardware accelerated Entropy Decoding (ED) module in the front-end of the GPU pipeline. ED is a strongly sequential kernel that cannot be efficiently mapped to the programmable GPU cores. The proposed unit can perform ED for H.265 and is extensible for other codecs, such as H.264/AVC, that use similar entropy coding schemes (e.g. CABAC).

- The integration of a new hardware unit called the Video Task Dispatch Unit (VTDU), that submit macro-operations to the Video Task Queues (VTQ) in each GPU core. The VTQs are similar to the existing Graphics Task Units but are optimized for video decoding kernels.

- Motion Compensation (MC), as one of the most time consuming parts of the decoding process, has been mapped to the existing Texture Units which contain fixed-function logic for performing pixel interpolation operations.

- General video kernels such as inverse quantization, inverse transform, intra-picture prediction, deblocking filter and SAO filter are executed on the GPU programmable cores.

- Video rendering is performed on the GPU cores assisted with fixed-function logic for the standard color conversion operations (e.g. YUV to RGB).

Programmable eGPU Video Decoder

A new programming model based on the concept of video macro-operations has been proposed and allows making the video decoder programmable and extensible to support multiple video codecs. The modules that have been accelerated using fixed-function logic are configurable, for example filter coefficients for MC interpolation. The integration of decoding and rendering in the GPU allows for additional optimizations, in particular block-level decoding can be coupled with tiled rendering for saving off-chip memory accesses. TU Berlin and Think Silicon are currently working on an implementation of the proposed design.