INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTATIONSDAG 2016-03-18

EXAMENSARBETE Image Processing Across Multiple Interconnected System-on-Chips
STUDENT Andrée Ekroth, Felix Mulder
HANDLEDARE Jonas Skeppstedt (LTH), Johan Rudholm (Axis), Magnus Mårtensson (Axis)
EXAMINATOR Per Andersson

Image Processing Across Multiple Interconnected System-on-Chips

POPULÄRVETENSKAPLIG SAMMANFATTNING **Andrée Ekroth, Felix Mulder**

This theis examines how best to divide the work of image processing across multiple interconnected System-on-Chips and shows by proof-of-concept that it is a feasible solution for bandwidth demanding applications.

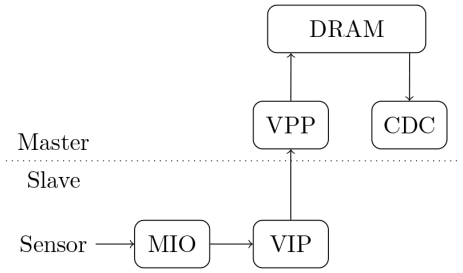
Splitting Up The Work

The motivation for the thesis was to allow Axis to utilize the hardware they designed themselves - instead of purchasing third party chips - to allow for high bandwidth applications like streaming at high resolutions.

During our thesis work, we have implemented a solution for dividing work between two SoC:s interconnected using a PCI Express (PCIe) bridge. It is easiest to imagine the relation between the two chips as a slave and master relationship, where the master simply tells the slave what to do.

Each chip contains several subsystems, and the good people at Axis have implemented an API to communicate with these subsystems. The API basically uses command structures sent from the operating system on the CPU to the subsystems containing Local CPUs (LCPUs). The beauty of this approach is that the system becomes very decoupled and modular - i.e. each system operates independently of the others.

In our thesis, we have split the work for different subsystems across two ARTPECs. In the single ARTPEC case, all subsystems are on the same chip.



Basically the figure illustrates the images' path through the different subsystems to the point at which it is made available for user consumption. To start with, we feed the system with a known image at the sensor stage on the slave side. When the image has been captured and processed by image improvement algorithms, it is sent via PCIe to the master ARTPEC. At this point the image is scaled and then sent for encoding. Once these steps have been performed, the image can be delivered to the user application.

We have chosen to divide the work by placing the image processing on the slave and the post-processing on the master ARTPEC. This is because the most bandwidth demanding subsystems are the image processing (VIP) and post-processing (VPP).

Results and Future Work

Our implementation results in a significant decrease of memory bandwidth usage on both the slave and the master ARTPEC. Most significantly it decreases the slave's bandwidth usage with 63% and the master's with roughly 25%.

Both the master and the slave ARTPEC contain all subsystems. This means that with our implementation the slave has subsystems and, perhaps more importantly, an entire CPU that is not being used. What this entails is that we have effectively enabled offloading of strenuous tasks!

In order to offload tasks to remote subsystems, it would be very easy for Axis to write an API that works in the same way as their current subsystem control mecha-nism. Examples of things they could easily offload are things like: audio re-sampling, user applications dubbed ACAP, facial recognition, and object recognition to name a few.