Current Status

- Driver for new DMA engine by Michele and Lorenzo (Partially Ready)
  - Verified **2.5** GB/s using data generator on x8 gen2 (Ready)
  - Faster version up to 3 GB/s + large DMA buffers (May)
  - High-speed 6 GB/s version (to be evaluated after I get hardware)
- Cleaner internal architecture enabling dynamic configuration (Finalized)
- New driver for CMOSIS IPECamera (On the way)
  - First test version (End of March - beginning of April)
  - Testing with all revisions of CMOSIS sensor (Mid – End of April)
- Driver for HEB (Unclear if still needed)
  - Do we need it? Which features (padding removal)? When?
- Fast IPE Camera (will not be fast, hardware needed for evaluation)
  - We need full-speed DMA first
  - Data flow and decoding optimizations probably required
- Further driver improvements (no definite schedule yet)
  - What else is needed on defined schedule?
- Testing platform (System ready, hardware is missing)

Driver for Michele's DMA

- Current status
  - Verified 2.5 GB/s using data generator on x8 gen2
  - Up to 3 – 3.2 GB/s should be easy to achieve
- Limitations
  - Only works with hardware IODMA enabled because hardware ignores lower 12 bits of the address.
  - 64-bit mode is failing on firmware I have (and required for PCIe gen3)?
  - In between of accessing certain registers non-documented sleeps are required (should not be a problem for performance)
- Outlook
  - Do not hurt performance when large amount of DMA buffers is used
  - More dynamic: configurable page size, pre-allocated contiguous memory
  - x8 gen3 hardware is needed for development 6 GB/s (I'll have some time in April - May)

Institute for Data Processing and Electronics
Karlsruhe Institute of Technology

New Architecture of Hardware Models

- Public API is unchanged
- IPECamera is now separate project from pcitool
- Easy development of new event models
  - We can implement HEB if needed
- Locking of CMOSIS registers to prevent crashes (student)
- Software registers (student)
- Register model described in XML (student)
  - Additional properties for registers (min, max, etc.)
  - Automatic unit conversion

IPECamera Event API

- CMOSIS Camera – 1 GB/s should be no problem
  - We need a testing stand (see next slides)
- Performance Issues
  - Support of 6 GB/s DMA required
  - Decoding could be moved to LibUCA to reduce memory copies (using raw data callback)
  - SIMD optimizations in LibUCA (complicated according to MV, we may require alternative data format)
- Work plan & schedule
  - To be defined when hardware available and all performance issues are evaluated

Institute for Data Processing and Electronics
Karlsruhe Institute of Technology

Advanced Features

- Kernel module simplification
  - Move more stuff to user-space (UIO, VFIO, etc.?)
- Evaluate Huge Pages in Linux
  - To enhance DMA performance and provide frames in contiguous memory
- Streaming in the user-supplied memory
  - Dangerous as hardware accessed memory is only valid during life-time of one application. Instead the kernel pages can be re-assembled as big buffer using vmmngr_map_page
  - Complex protocol is required to avoid overwriting data by the hardware
- Dynamic buffer management
  - No ring buffer, just send data in the newly configured buffers
  - Hardware: Can I add new buffers to the hardware while reading data? What is upper limit of buffer number?
  - Write-and-forget DMA operation mode?

Institute for Data Processing and Electronics
Karlsruhe Institute of Technology

Direct PCIe communication

- Reduce required memory copies
  - vmsplice to stream data directly to the network/storage
  - GPUDirect for Video to send data into the NVIDIA GPU
- Direct communication on PCIe Bus
  - GPUDirect for NVIDIA
  - DirectGMA for OpenCL
- Organize data flow
  - Camera → GPU → Infiniband (KIRO)
  - Camera → Infiniband (Hardware support needed)
- GPU-based preprocessing
  - Online data compression
  - Visualization
  - Integration with UFO (difficult due non-exisiting interface between CUDA and OpenCL) or other way to program complex GPU filtering and Visualization.
- Low power camera station: Intel Atom, Intel Galileo, ARM?
- Integration of Camera directly into the UFO cluster

Continuous Integration

- Testing of full software chain
  - Camera → Driver → LibPCO → KIRO → Infiniband → Server → UFO framework → Fast clustered storage
  - The place for setup and all systems are ready in the server lab
  - No free hardware
- Continuous integration of hardware and software
  - Manual testing is NOT working...
  - FPGA firmware should be under revision control
  - FPGA firmware should report actual revision
  - Automatically detect current revision, check for updates, load new firmware and check if full chain still operating
  - Automatically test all hardware flavours, i.e. 10 and 12 bit modes, 2 – 4 Mpix sensors, etc.?
- Advanced test scenarios are needed
  - All provided features have to be tested
  - Grabbing under different parameters (resolution, exposure, …)
  - Simulating user interactions

Work Plan

- Hardware requirements
  - 2x extra camera setups (testing platform, new student)
  - gen3 or x16 hardware counter for developing faster DMA driver
  - Hardware versioning and revision control for continuous integration
  - Data format documentation in BANK_REG_UFO_HEB_DMA.xls
- Hardware testing (setup is ready, we just need hardware)
- New internal architecture (March)
- Full support of CMOSIS-based IPECamera (April)
- Higher-speed DMA drivers (May, hardware needed)
- Dynamic DMA configuration (May)
- Support of large DMA buffers (May - June)
- DMA driver capable of 6 GB/s (Summer if hardware is provided soon)
- Software and XML registers, better locking (Fall if works fine with student)
- Continuous integration (Fall, but we need versioning)
- Drivers for high-speed IPE Camera (decided upon evaluation of hardware)
- GPUDirect and other advanced features (undefined)
- Any other requirements with defined deadline?
  - Lets use UFO tickets as much as possible in advance

Institute for Data Processing and Electronics
Karlsruhe Institute of Technology