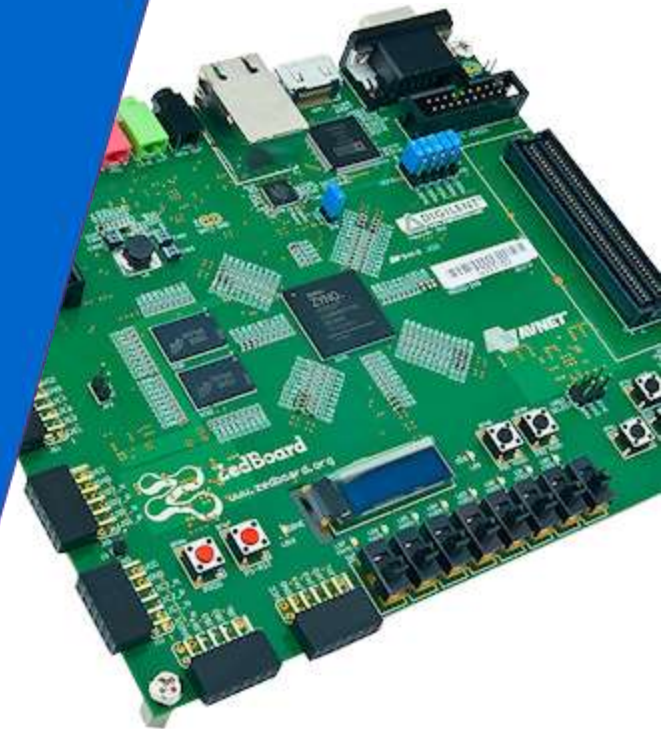


# Throughput Exploration and Optimization of a Consumer Camera Interface for a Reconfigurable Platform

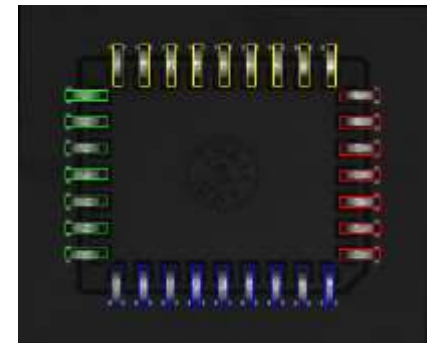
By: **Floris Driessen** ([f.c.driessen@student.tue.nl](mailto:f.c.driessen@student.tue.nl))



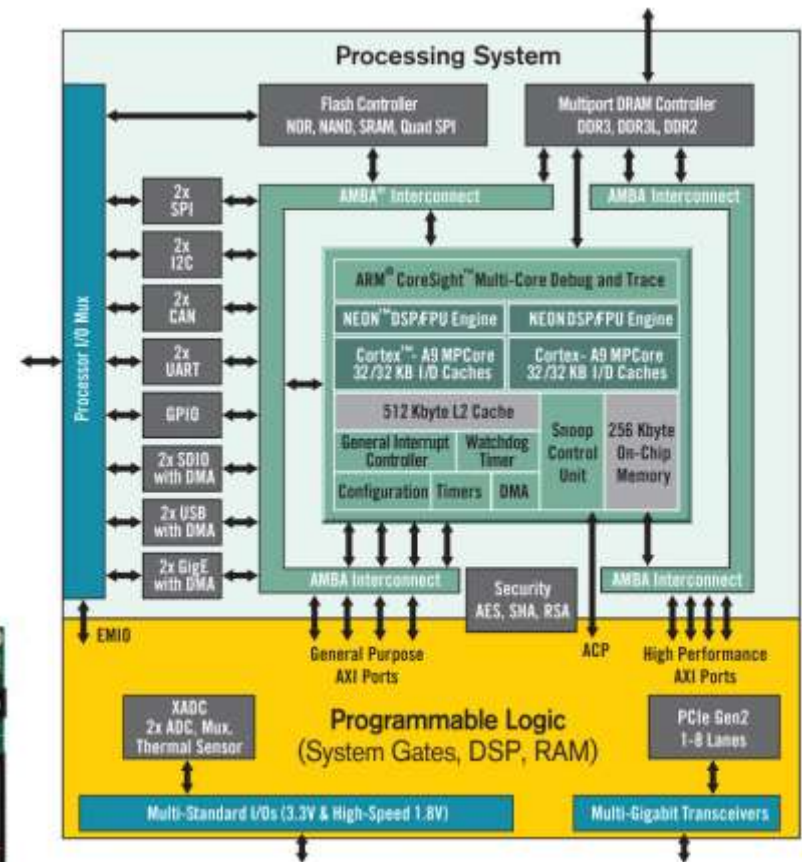
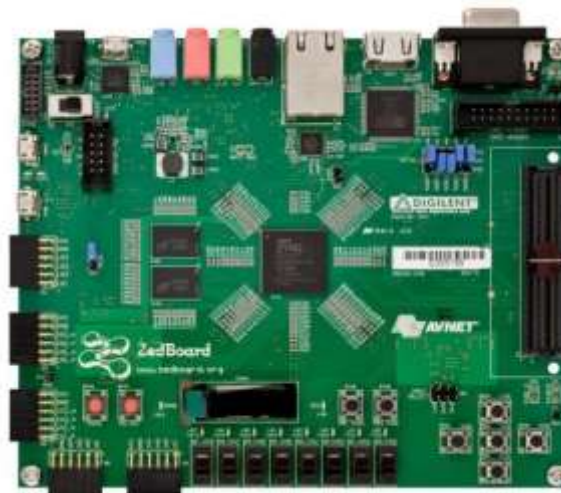
**TU** / **e**

Technische Universiteit  
**Eindhoven**  
University of Technology

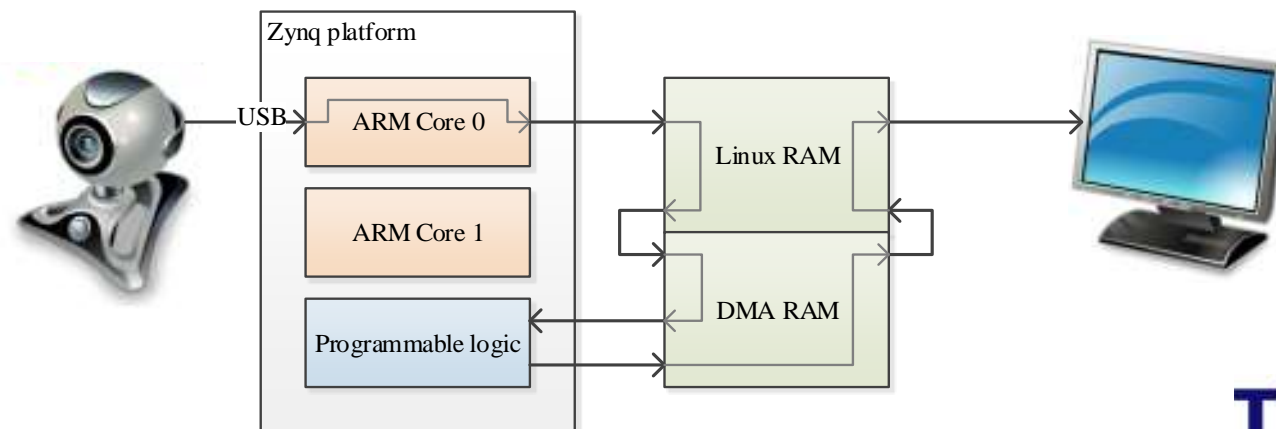
- Video applications on embedded platforms
- Use of accelerators
  - Faster
  - Energy efficiency
- USB camera



- Zedboard by Digilent
  - Xilinx Zynq platform
    - Dual core ARM Cortex A9
    - Programmable logic
  - 512 MB RAM
  - USB connectivity
  - HDMI output
  - USB camera

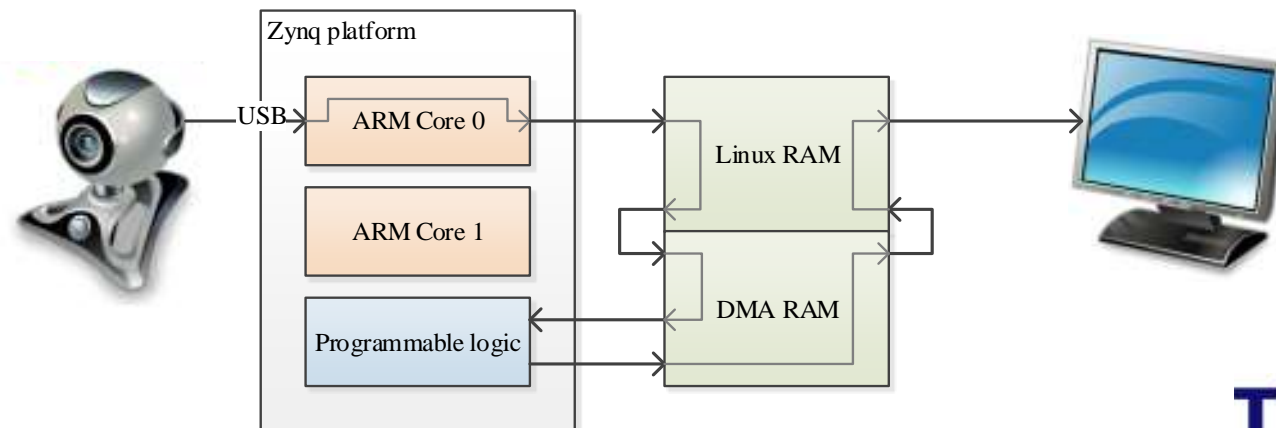


- **Software**
  1. Read camera frame
  2. Copy frame to DMA region
  3. Perform HW accelerated operation (Sobel)
  4. Copy result from DMA region
  5. Show result
- **Separate DMA region needed due to lack of DMA drivers**



- Performance limit
- Converting the format
  - Camera output to accelerator input
- Copying from/to DMA region
  - Mmap
    - Not cached
- Frame capturing

Routine	Time taken [s]	Percentage of time
Capture frame	0.066	1.5%
Convert and copy	1.948	45.1%
Sobel filter execution	0.007	0.2%
Convert and copy back	2.298	53.2%
Total	4.253	100%



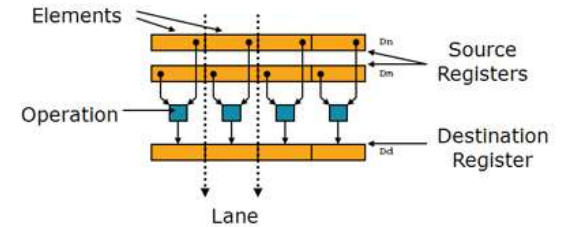
# Possible improvements

5

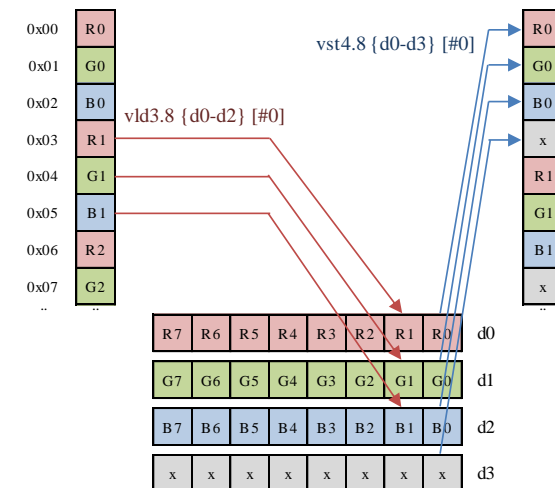
- **Exploiting scratchpad**
  - A frame would not fit
- **DMA driver support**
  - Not feasible within time frame of project
- **Optimize the current implementation**
  - Copying data
  - Converting format
  - Capturing camera frame

ARM® CoreSight™ Multi-Core Debug and Trace				
NEON™ DSP/FPU Engine		NEON DSP/FPU Engine		
Cortex™-A9 MPCore 32/32 KB I/D Caches		Cortex- A9 MPCore 32/32 KB I/D Caches		
512 Kbyte L2 Cache				Snoop Control Unit
General Interrupt Controller		Watchdog Timer	256 Kbyte On-Chip Memory	
Configuration	Timers	DMA		

- Naïve implementation
  - Combined conversion and copy
    - Writing small chunks to mmaped memory (slow)
- Split conversion and copy
- OpenCV mixChannels
- NEON interleaving
  - ARM SIMD
  - Next slide



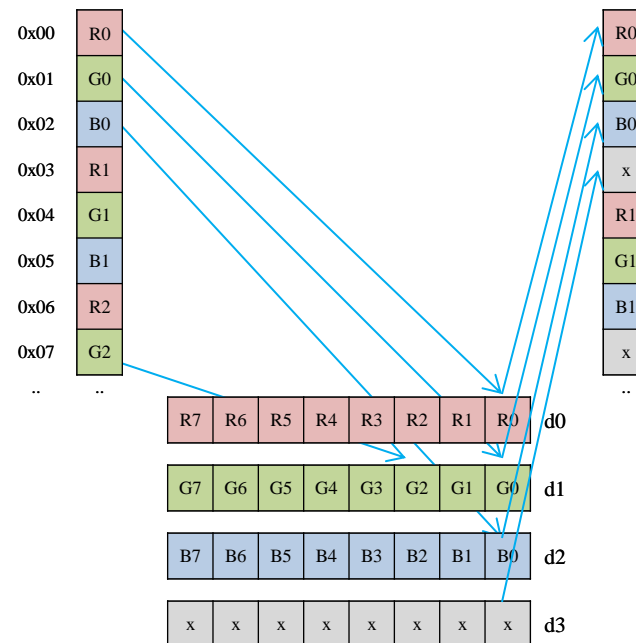
Implementation	Convert + copy [s]	Speed-up
Naïve	1,95	1x
Split	0,28+0,04=0,32	6,1x
OpenCV	0,05+0,04=0,09	21,7x
NEON	0,04	50,6x



# NEON RGB24 to RGB32 conversion example

7

```
void __attribute__((noinline))
neonRGBtoRGBA_gas(unsigned char* src, unsigned char* dst,
int numPix)
{
    asm(
        // numpix/8
        "    mov    r2, r2, lsr #3\n" // numpix/8
        // load alpha channel value
        "    vmov.u8 d3, #0xff\n"
        "loop1:\n"
        // load 8 rgb pixels with deinterleave
        "    vld3.8 {d0,d1,d2}, [r0]!\n"
        // preload next values
        "    pld    [r0,#40]\n"
        "    pld    [r0,#48]\n"
        "    pld    [r0,#56]\n"
        // subtract loop counter
        "    subs  r2, r2, #1\n"
        //"    vswp  d0, d2\n"
        // store as 4*8bit values
        "    vst4.8 {d0-d3}, [r1]!\n"
        // loop if not ready
        "    bgt  loop1\n"
    );
}
```

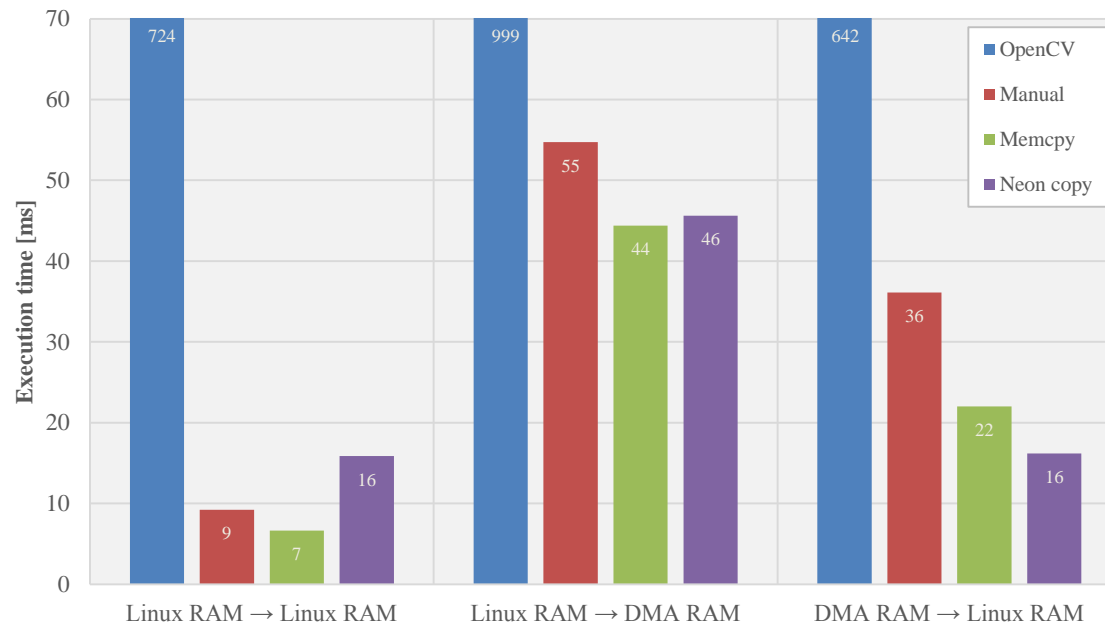




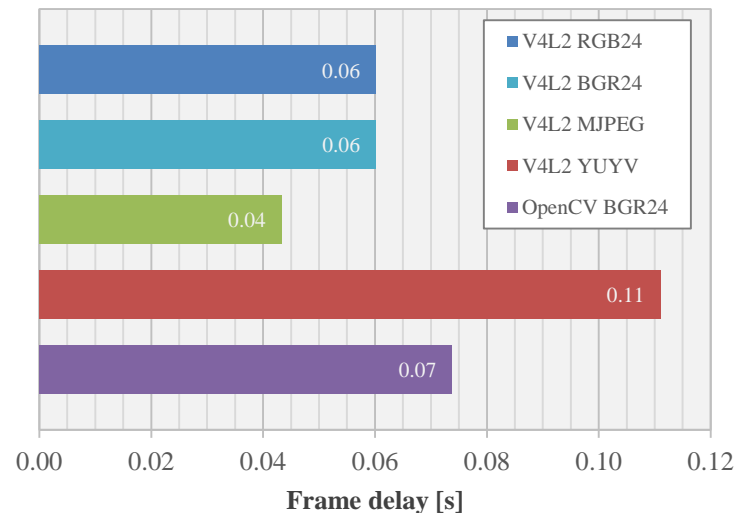
# Frame copy from/to DMA RAM

8

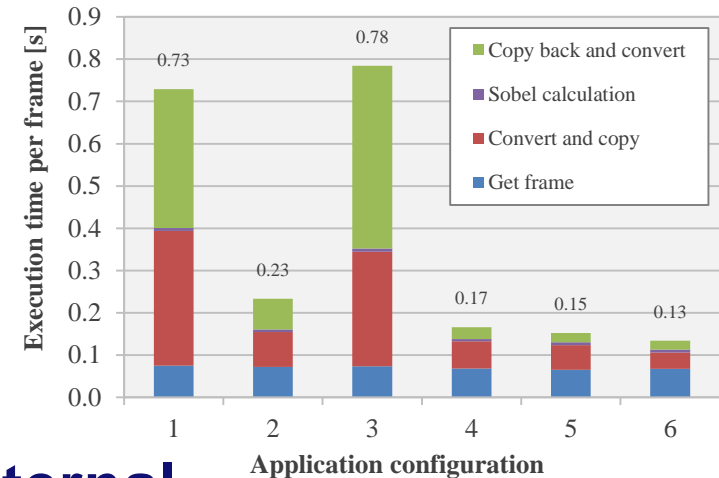
- **OpenCV (as used in the naïve implementation)**
- **Manual copy (loop over virtual contiguous memory)**
- **Memcpy from C library**
- **NEON accelerated copy**



- **OpenCV**
  - Always BGR24
- **Video4Linux**
  - Different formats
- **Not a big improvement**

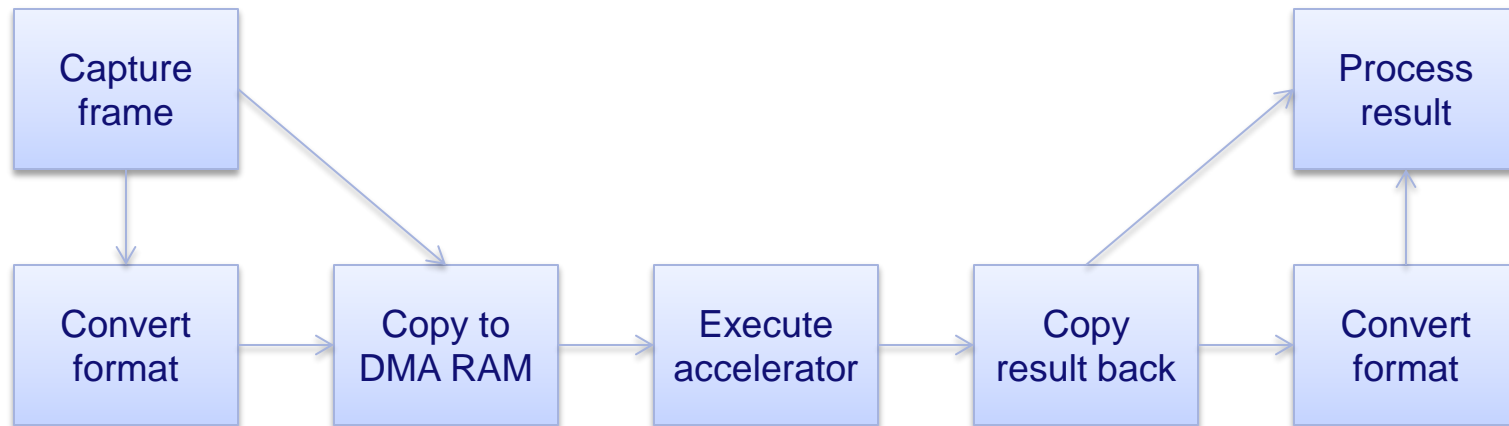


- **Multiple configurations**
- **Combined the conversion and copy (NEON accelerated)**
- **1: Split convert and copy**
- **2: OpenCV mixChannels**
- **3: Combined mixChannels to external**
- **4: No convert back + V4L capture**
- **5: NEON copy**
- **6: Combined NEON convert and NEON copy**



Routine	Time taken [s]	Percentage of time	Improvement
Capture frame	0.065	50.2%	1.1x
Convert and copy	0.038	28.8%	50.6x
Sobel filter execution	0.007	5.1%	1.0x
Convert and copy back	0.021	15.9%	108.2x
Total	0.131	100%	32.4x

- **Framework for combining USB camera with accelerators in programmable logic**
- **Multiple format conversion routines**
  - **NEON**
- **NEON copying routines**
- **Video4Linux frame capture**



- **Huge improvement 32x (0,2 to 7,7 FPS)**
- **Still one ARM core unoccupied for processing data after accelerator**
- **Make camera frame buffer available to DMA**
  - **DMA buffer sharing**
    - **Linux kernel 3.8**
- **Improve frame capture**
  - **Takes more than half of the time**
  - **Latency of ~4 frames**
  - **Driver from manufacturer**
  - **Consider other cameras**