# Target Imaging System - Electronics development

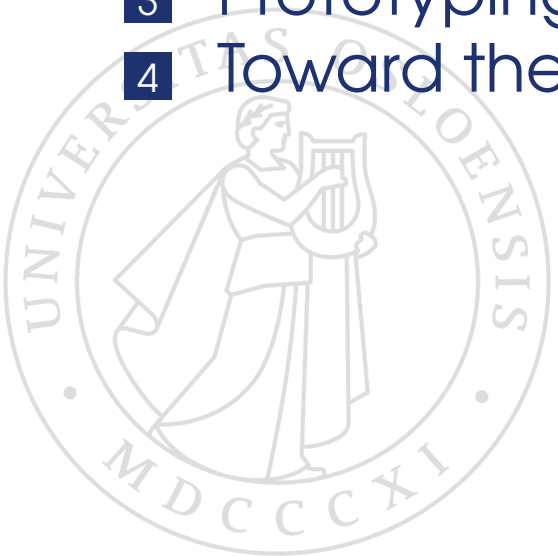
**David Michael Bang**[1] **Ole Myren Røhne**[1]

[1]University of Oslo

Trieste 3rd Beam Diagnostics Forum

# Outline



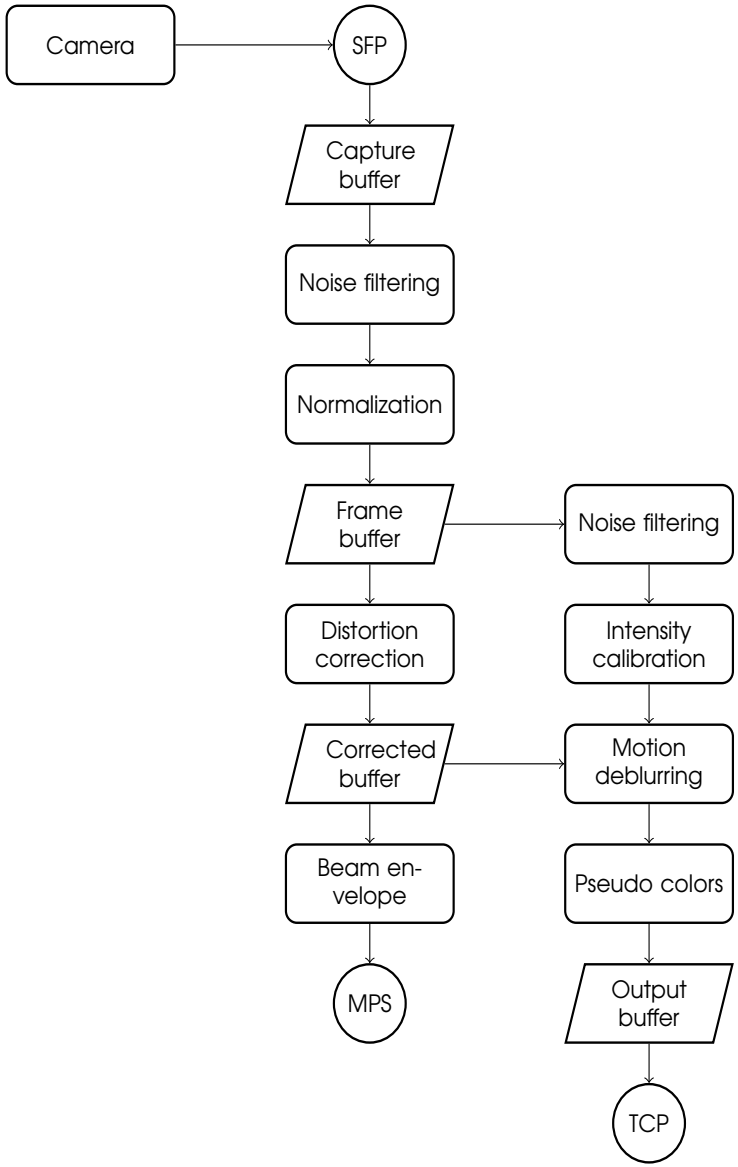


# Functional requirements

| Preprocessing | | | |
|---|---|---|---|
| Noise removal | Median filtering | tbd | tbd |
| Dead/hot pixels interpolation | Background image can be used | yes | tbd |
| Vignetting correction | | yes | tbd |
| **Calibration and correction** | | | |
| Geometrical distortion correction | Static look-up tables | yes | yes |
| Motion blurring | Subtracting constant offset might be sufficient | tbd | tbd |
| Normalization to beam current | Not clear if best strategy (makes IMG dependent BCM) | tbd | tbd |
| Correction for temperature (coating light yield) | Look-up table from test results. | tbd | tbd |
| **Parameter extraction** | | | |
| Centroid position | | yes | yes |
| Peak density | Default avering area 1 x 1 cm2 | yes | yes |
| Percentage outside defined footprints | Static rectangles: 99%, 99.9%, 99.99%. | yes | yes |
| Position of fiducials | Fast and slow changes in geometry | tbd | n/a |
| Histogram of actual beam, iso-curves | Might not be needed, see percentage outside defined footprint | tbd | n/a |
| **Presentation** | | | |
| Background subtract | Background frame stored on request | yes | tbd |
| Image display, 14 fps video | Including all corrections back to physical beam | yes | n/a |
| Beam diagnostic pulse | Might include non-standard processing | tbd | n/a |
| Functionality | | SW | FPGA |

(From `https://confluence.esss.lu.se/display/BIG/Software+and+FPGA+functionality+summary`)

# Data flow



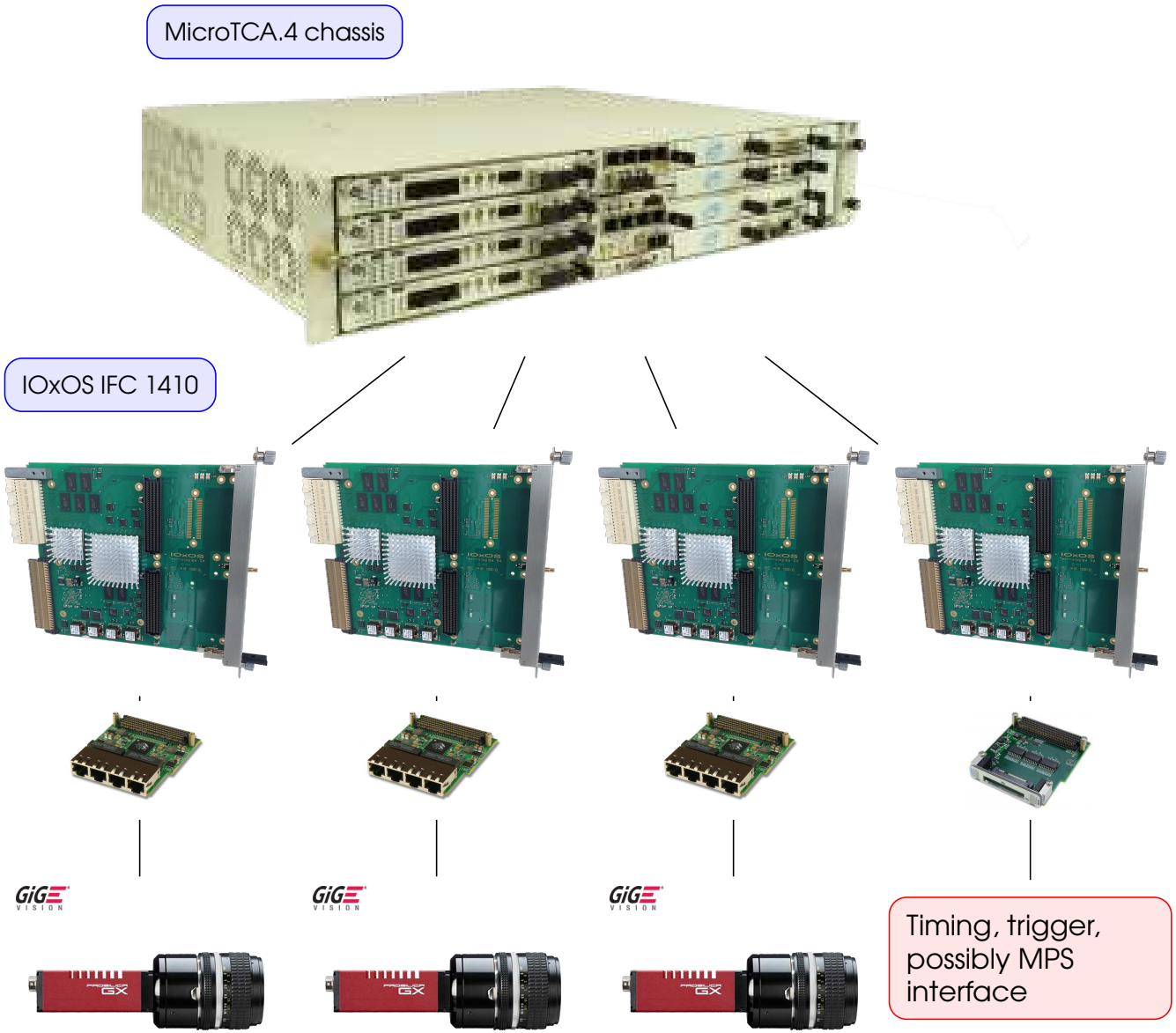

## Deterministic latency path (FPGA)

- Noise filtering
- Normalization
- Pixel mapping
- Distortion correction
- Apply critical limits

## Medium latency path (CPU/GPGPU)

- Intensity cross-calibration
- Motion de-blurring
- Apply pseudo colors
- Video stream codec

# Electronics system



# Front-end components

## Camera

- PBW and TW cameras are similar: friendly environment
- TD: harsher environment
- Trigger synchronization, $1\ \mathrm{ms}$ or better?
- Frame rate, $n \times 14\ \mathrm{fps}$
- Sensor size, is 1/3-format enough?
- Pixel resolution, few $\mathrm{MPix}$ enough
- Digitization, 14-16 bits for dynamic range

In-hand: Allied Vision Manta G-125 GigE/PoE

## FMC GigE Frame grabber/splitter

- GigE Vision, just needs a PHY-implementation
- Cameralink FMC also exists
- (assuming USB is not industrial-grade)

E.g. FMC103 - FMC Quad 10/100/1000 GbE Module

# Back-end components

## AMC IOxOS IFC1410

- FPGA: Xilinx Kintex UltraScale KU040/KU060
- CPU: NXP QorIQ T2081 Quad core Power/AltiVec @ 1.8 GHz
- PCIe: Data stream to CPU-board

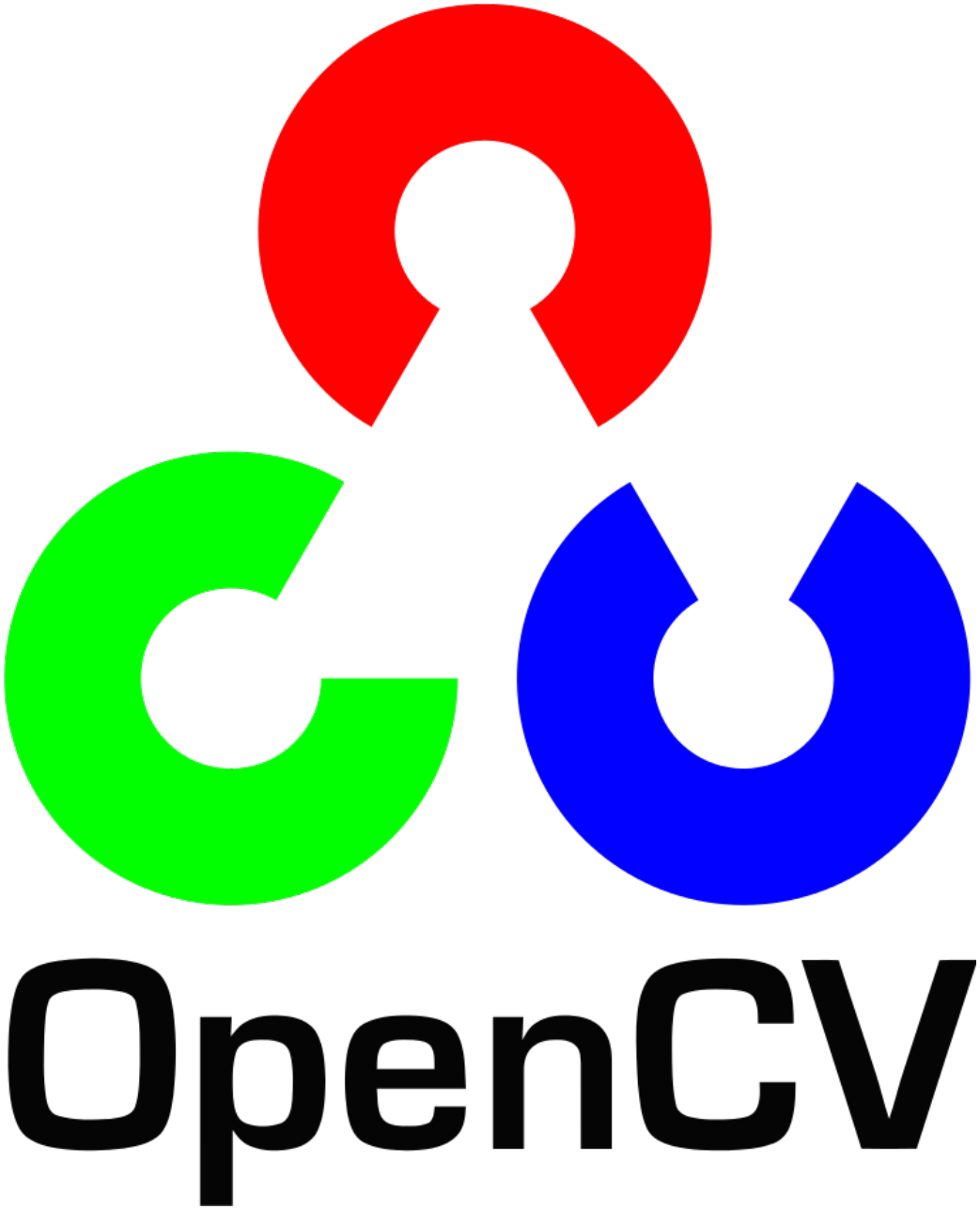
## MicroTCA.4 chassis

- Power supply, possibly redunant
- MicroTCA Carrier Hub and CPU
- Rear Transition Module, for synchronous signals

## Services infrastructure

- Power, cooling networking
- Remote power-up, booting
- Data recording/storage

# Software algorithms platform




## Open source software library

- Targeting computer vision and machine learning
- Video/imaging framework and 2500 algortims
- Implemented in C++/STL, also on GPU (CUDA and OpenCL)
- Easily usable from a wide range of languages including Python

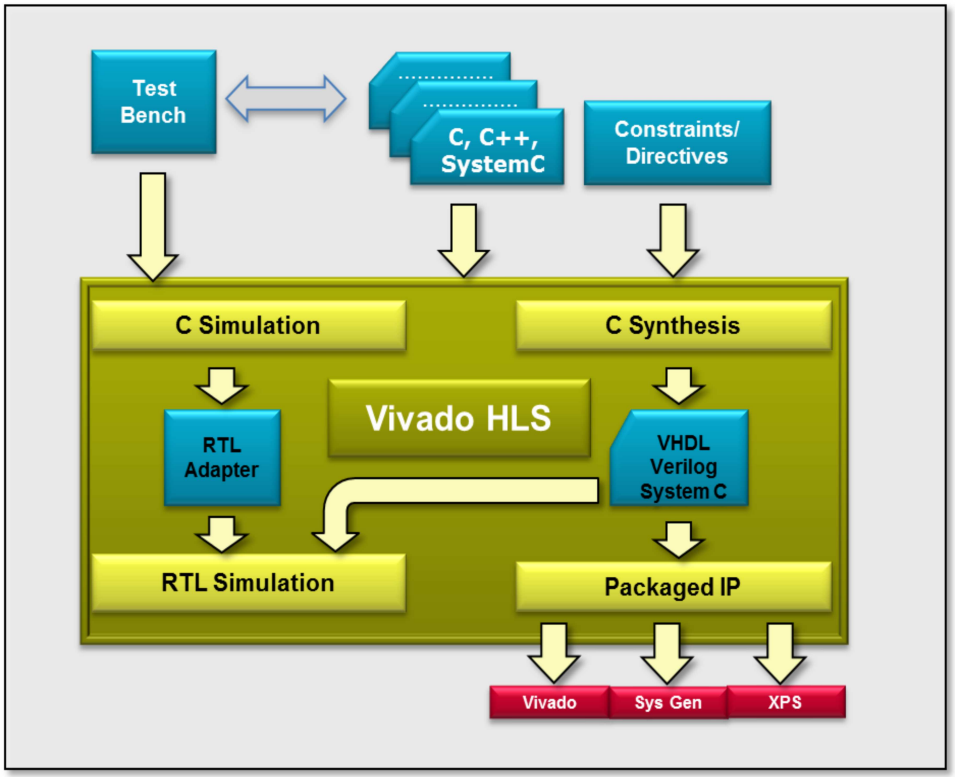
## Benefits for Target Imaging

- Xilinx Vivado HLS video processing cores refers to OpenCV
- EPICS areaDetector plugins already exists

What more can you ask?

# Hardware algorithms environment



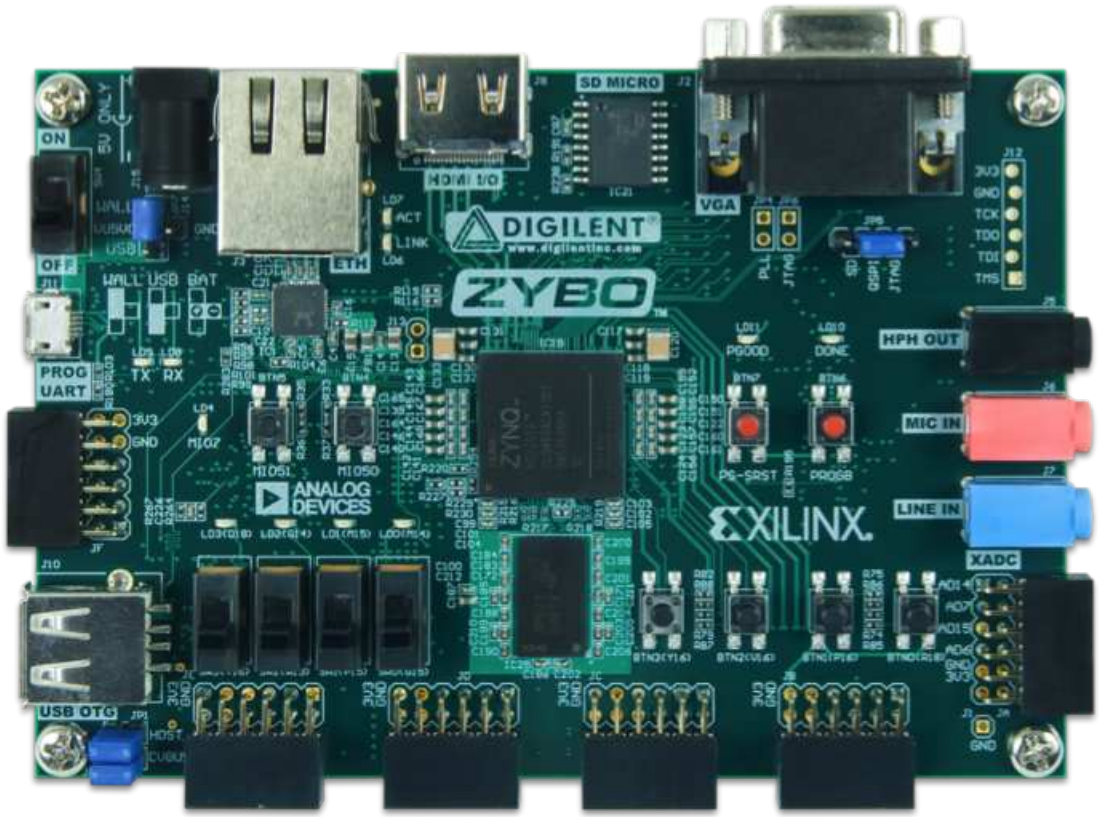



## Xilinx Vivado HLS

- Write algorithms in C language
- Synthesises through RTL-representation
- Integrates with lower-level (VHDL/Verilog) IP
- RTL-level simulation, resources utilisation, algorithm verification

# Prototype platform

Digilent Zybo Zynq-7000 ARM/FPGA SoC Trainer Board




## Key features

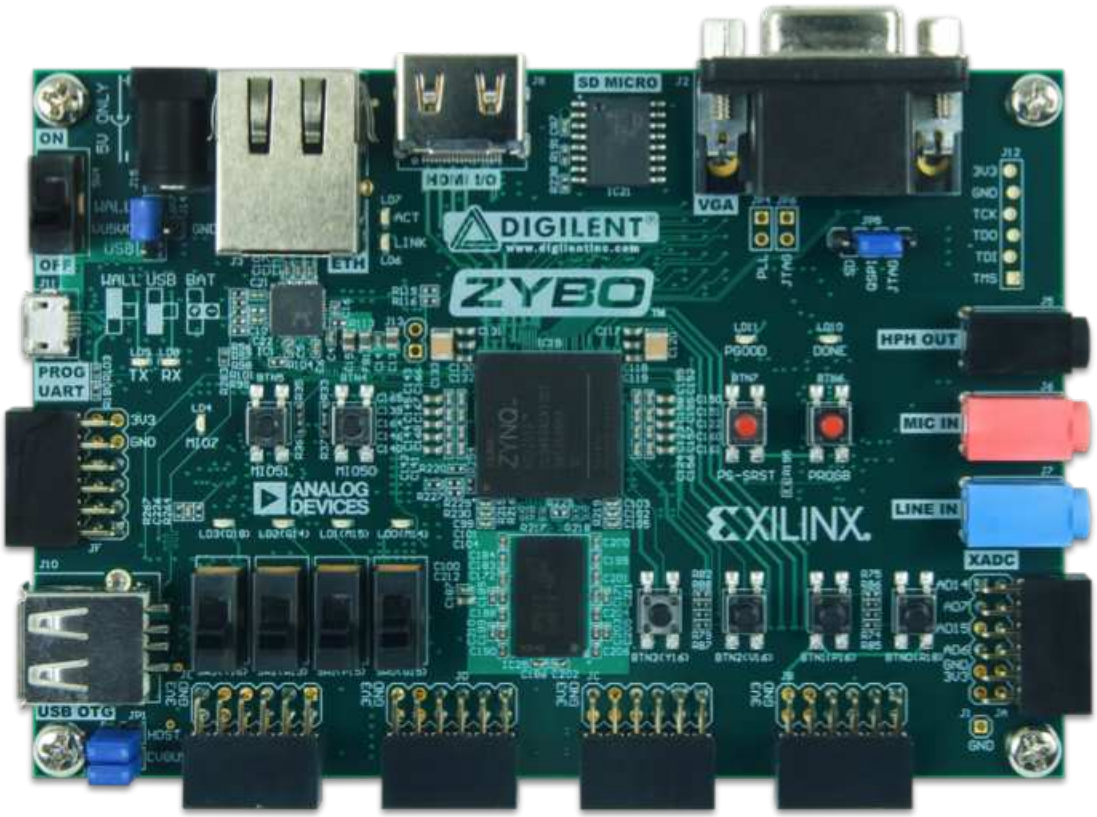
- Xilinx Zynq-7000 (XC7Z010-1CLG400C)
- 512 MB x32 DDR3 w/ 1050Mbps bandwidth
- 16-bits per pixel VGA output port
- High-bandwith peripheral controllers: 1G Ethernet, USB 2.0, SDIO

## Xilinx ZYNQ 7000 XC7Z010

- 28,000 logic cells
- 240 KB Block RAM
- 650 MHz dual-core Cortex~A9 processor

# Prototype platform (1)

Digilent Zybo Zynq-7000 ARM/FPGA SoC Trainer Board



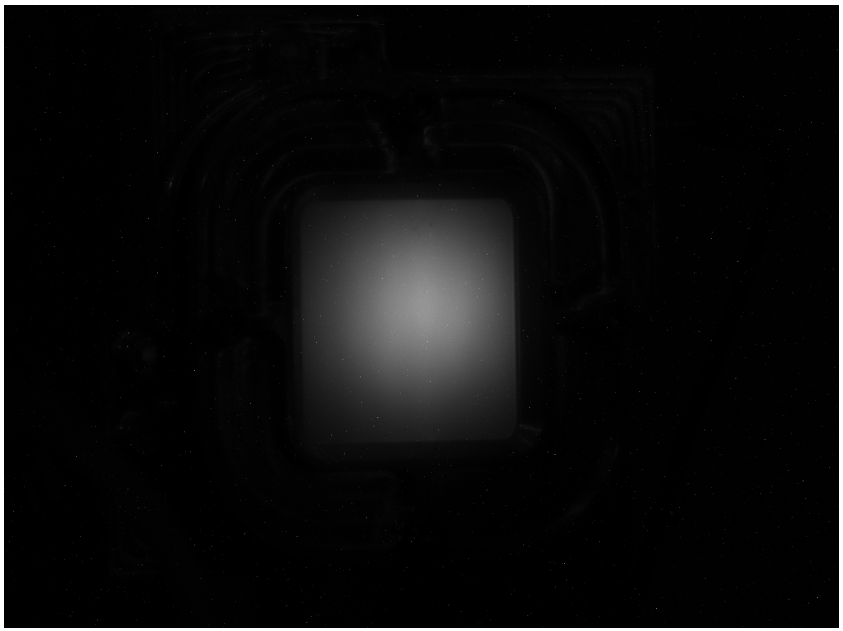

## Workflow

- Serial transmission of test image on UART from PC to ZYNQ stored on DRAM
- Bare metal program running on ARM used for controlling and configuring FPGA IP cores
- VDMA creates a video stream from stored test image
- VGA output for displaying processed video stream
- CPU/FPGA interface is AXI and VDMA/AXI-stream

XILINX ALL PROGRAMMABLE

DIGILENT A National Instruments Company

# Centroid position



Test Frame 1292x964

$$C_x = \frac{\sum_x \sum_y xP(x,y)}{\sum_x \sum_y P(x,y)}$$

$$C_y = \frac{\sum_x \sum_y yP(x,y)}{\sum_x \sum_y P(x,y)}$$

$$RMS\_C_x = \sqrt{\frac{\sum_x \sum_y x^2 P(x,y)}{\sum_x \sum_y P(x,y)} - C_x^2}$$

$$RMS\_C_y = \sqrt{\frac{\sum_x \sum_y y^2 P(x,y)}{\sum_x \sum_y P(x,y)} - C_y^2}$$

# Centroid position

```
function [cog_X, cog_Y, rms_X, rms_Y] = E200_im_cog_rms_XY(myimg, cut_back);

if nargin < 2
  cut_back = 0; % extra background cut needed to avoid centroid bias
end % if

myimg(myimg < cut_back) = 0;

X_hist=sum(myimg,1);
Y_hist=sum(myimg,2);
X=1:size(myimg, 2);
Y=1:size(myimg, 1);
cog_X=sum(X.*X_hist)/sum(X_hist);
cog_Y=sum(Y'.*Y_hist)/sum(Y_hist);

rms_X = sqrt( sum((X-cog_X).^2.*X_hist)./sum(X_hist) );
rms_Y = sqrt( sum((Y'-cog_Y).^2.*Y_hist)./sum(Y_hist) );
```

Matlab Implementation

Cx = 641.7417

Cy = 462.6578

RMS_Cx = 169.8422

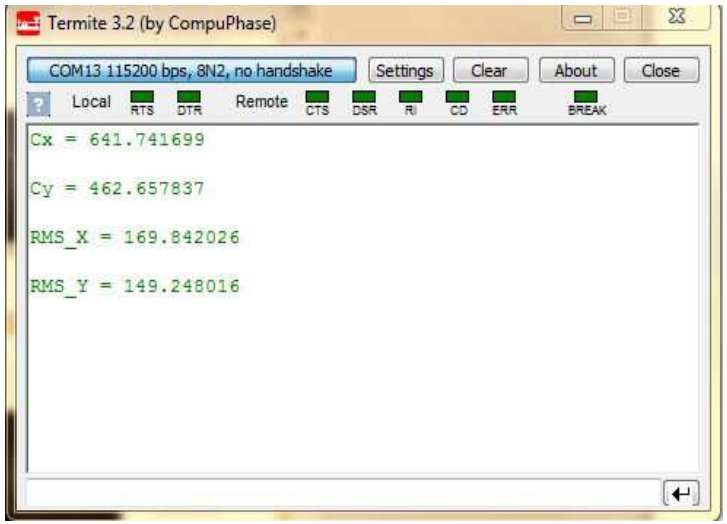
RMS_Cy = 149.2480

# Centroid position

FPGA Implementation

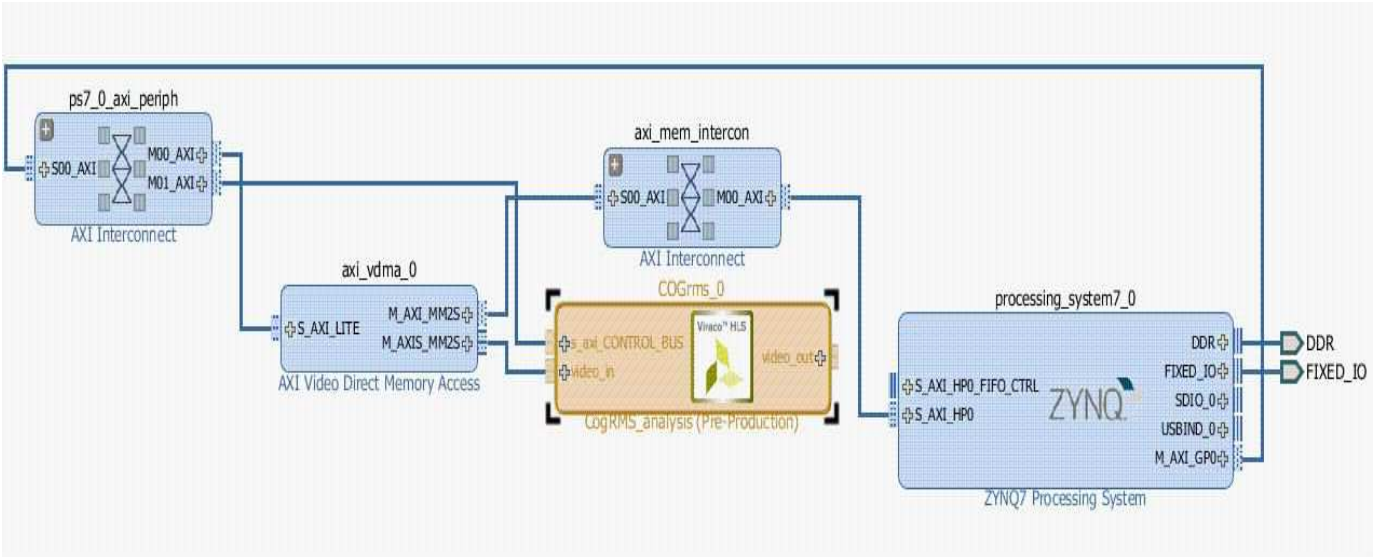
```
LOOP0: for (int n = 1; n <= 964; n++){
        for (int m = 1; m <= 1292; m++)
            {
            #pragma HLS PIPELINE
            video_in >> Apixel;
            video_out << Apixel;
            p_sum_x2 += m*m*Apixel.data;
            p_sum_x1 += m*Apixel.data;

            p_sum_y2 += n*n*Apixel.data;
            p_sum_y1 += n*Apixel.data;

            p_sum += Apixel.data;
            }
        }

mp_sum_x1 = (float) (p_sum_x1)/(p_sum);
mp_sum_y1 = (float) (p_sum_y1)/(p_sum);

mp_sum_x2 = (float) (p_sum_x2)/(p_sum);
mp_sum_y2 = (float) (p_sum_y2)/(p_sum);

*p_sum_out = (float)p_sum;

*Cog_X = mp_sum_x1;
*Cog_Y = mp_sum_y1;

*RMS_X = sqrtf((mp_sum_x2 - (mp_sum_x1)*(mp_sum_x1)));
*RMS_Y = sqrtf((mp_sum_y2 - (mp_sum_y1)*(mp_sum_y1)));
```

Snippet of Vivado HLS C-code



# Centroid position

FPGA Implementation




Vivado: Block diagram with HLS generated Centroid Position IP showing interface connection only

# Vivado HLS Report Comparison

**solution1:** xc7z010clg400-1
**solution2:** xcku040-sfva784-1-i
**solution3:** xcku040-fbva676-3-e

**Timing (ns)**

| Clock | | solution1 | solution2 | solution3 |
|---|---|---|---|---|
| ap_clk | Target | 10.00 | 10.00 | 10.00 |
| | Estimated | 8.22 | 8.29 | 8.65 |

**Latency (clock cycles)**

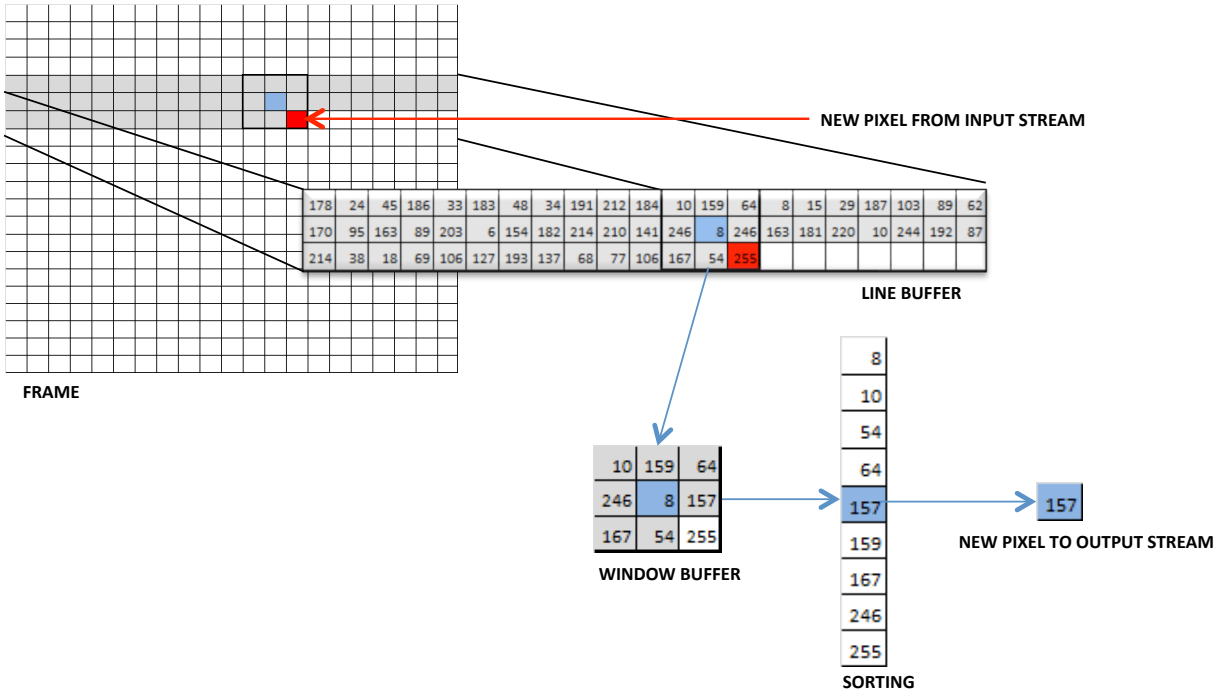
| | | solution1 | solution2 | solution3 |
|---|---|---|---|---|
| Latency | min | 1245536 | 1245520 | 1245514 |
| | max | 1245536 | 1245520 | 1245514 |
| Interval | min | 1245494 | 1245493 | 1245493 |
| | max | 1245494 | 1245493 | 1245493 |

Resource Usage

| | solution1 | solution2 | solution3 |
|---|---|---|---|
| BRAM_18K | 0 | 0 | 0 |
| DSP48E | 17 | 17 | 17 |
| FF | 5525 | 4017 | 3576 |
| LUT | 8060 | 6043 | 6007 |

Resolution = 1292x964 = 1245488 pixels

# Median filter



# Median filter

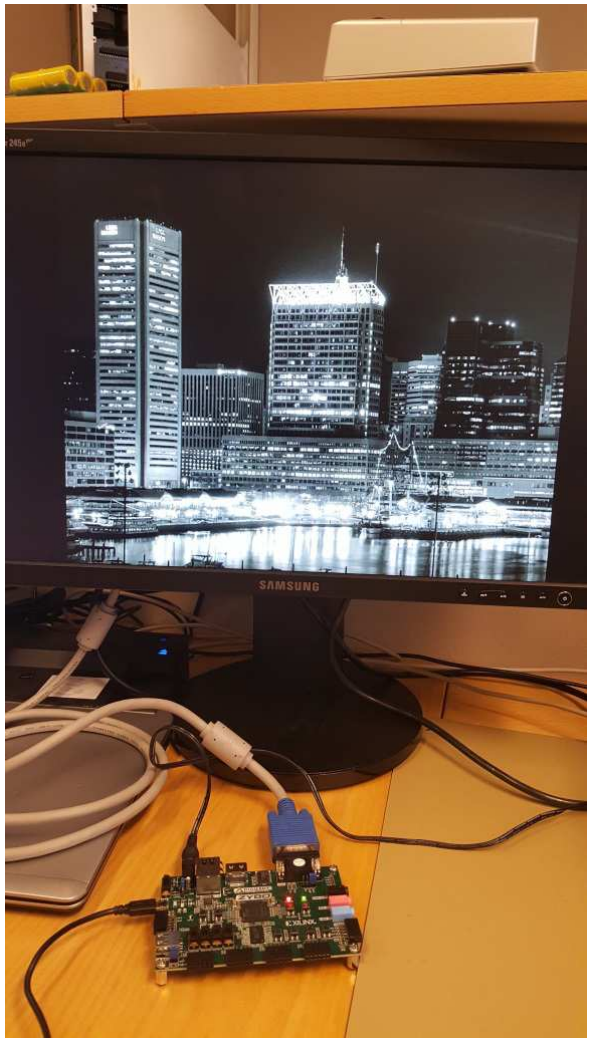
Original Picture

Added salt and pepper noise

C/RTL co-simulation

Median Filter

# Median filter



# Vivado HLS Report Comparison

**solution1:** xc7z010clg400-1
**solution2:** xcku040-sfva784-3-e
**solution3:** xcku040-sfva784-1-i

**Timing (ns)**

| Clock | | solution1 | solution2 | solution3 |
|---|---|---|---|---|
| ap_clk | Target | 10.00 | 10.00 | 10.00 |
| | Estimated | 8.73 | 8.39 | 8.05 |

**Latency (clock cycles)**

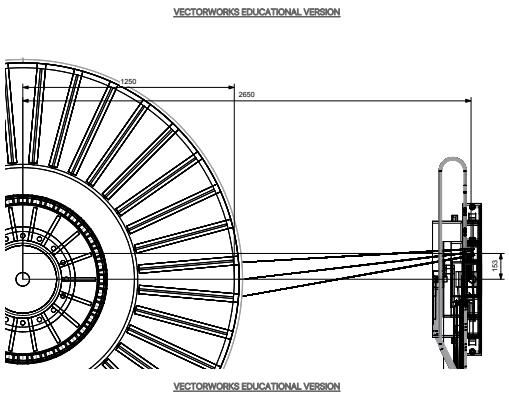
| | | solution1 | solution2 | solution3 |
|---|---|---|---|---|
| Latency | min | 8 | 3 | 8 |
| | max | 1245500 | 1245494 | 1245499 |
| Interval | min | 9 | 4 | 9 |
| | max | 1245501 | 1245495 | 1245500 |

Resource Usage

| | solution1 | solution2 | solution3 |
|---|---|---|---|
| BRAM_18K | 2 | 2 | 2 |
| DSP48E | 4 | 4 | 4 |
| FF | 651 | 545 | 586 |
| LUT | 1051 | 1048 | 1051 |

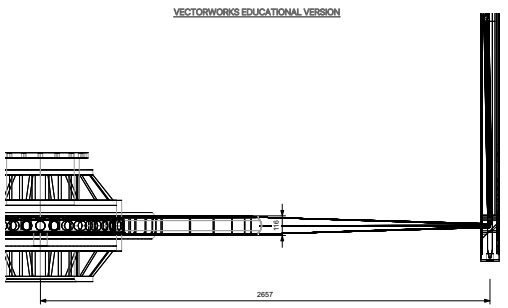
Resolution = 1292x964 = 1245488 pixels

# Geometrical distortion correction



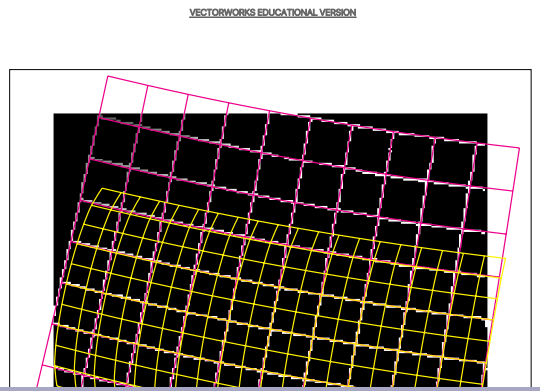

## Beam projection

- Rectilinear beam system
- Doubly-curved outer surface wheel
- Convex hull of torus
- Variable depth object




## Camera matrix

- Pinhole projection
- Off-axis wrt variable depth
- Closed solution exists




## Optical distortion

- Trivial image rotation
- Some barrel distortion
- Center-of-correction well outside frame

# Geometrical distortion correction

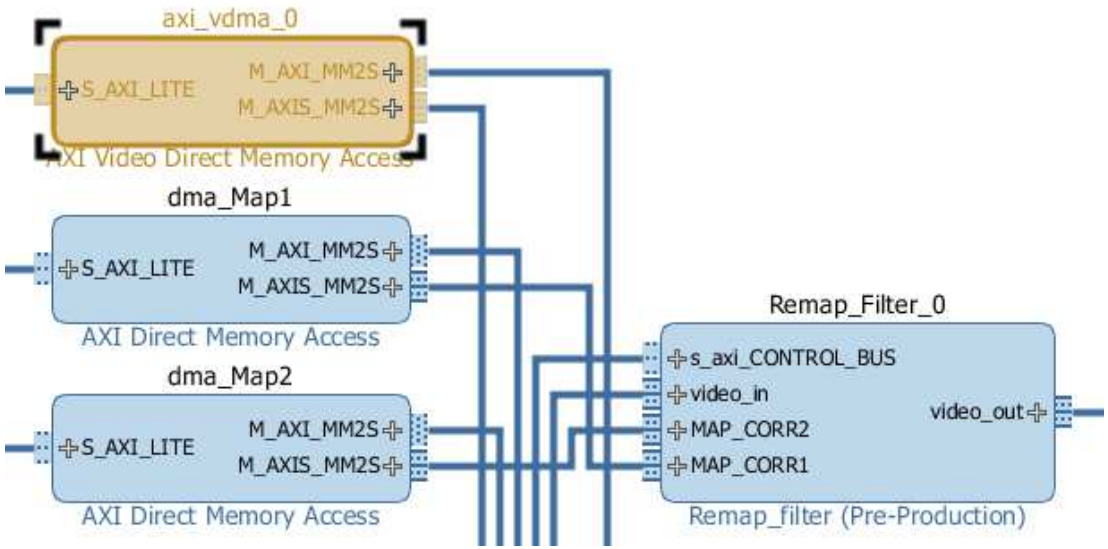
May use the Remap function in HLS video library

```
hls::AXIvideo2Mat(video_in, img_0);
hls::AXIvideo2Mat(MAP_CORR1, map_1);
hls::AXIvideo2Mat(MAP_CORR2, map_2);
hls::Remap<128>(img_0, img_1, map_1, map_2, HLS_INTER_LINEAR);
hls::Mat2AXIvideo(img_1, video_out);
```

Remaps the source image to the destination image according to the given remapping

Contains a linebuffer that buffers the required amount of rows needed for vertical displacement

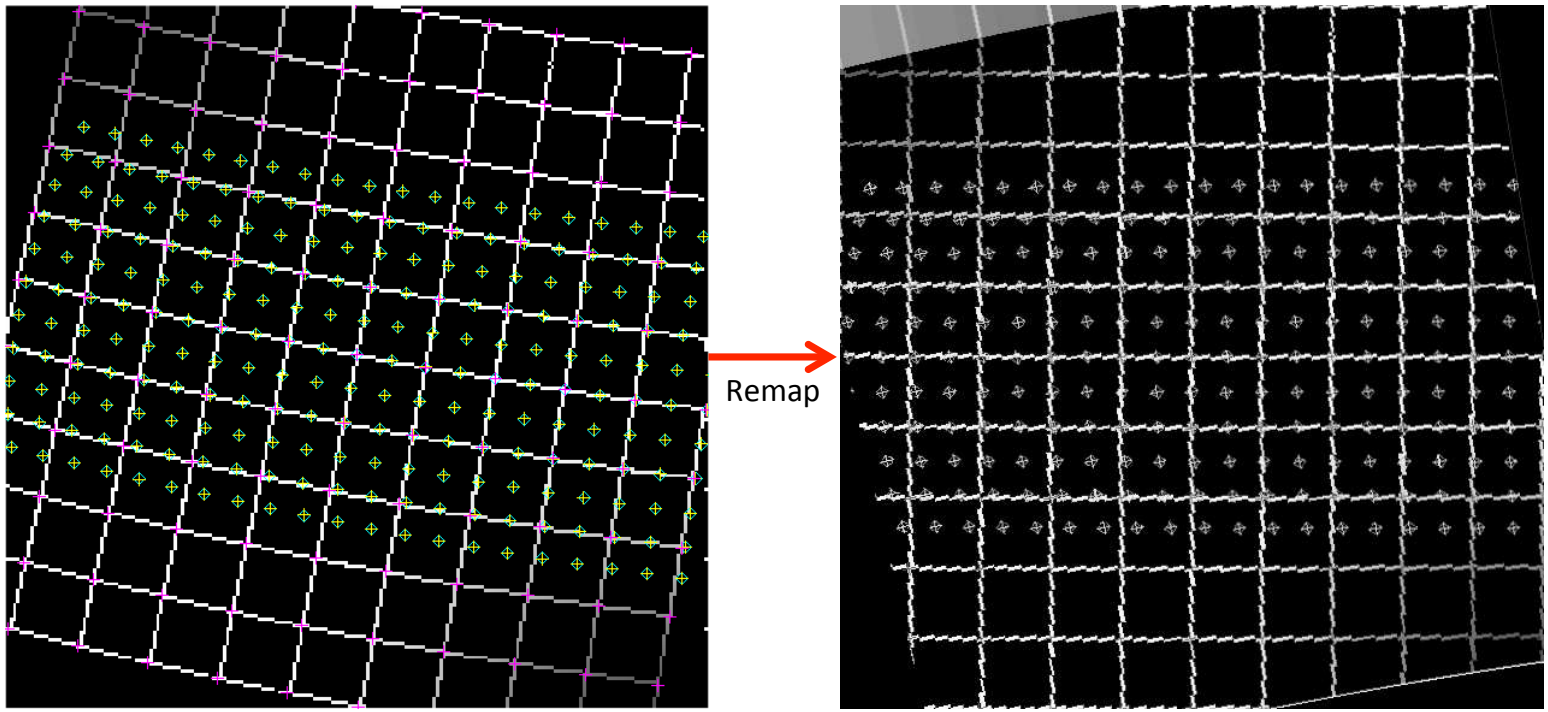
# Geometrical distortion correction



# Geometrical distortion correction

C/RTL cosimulation

Image resolution: 585x585 LineBuffer: 128 Lines



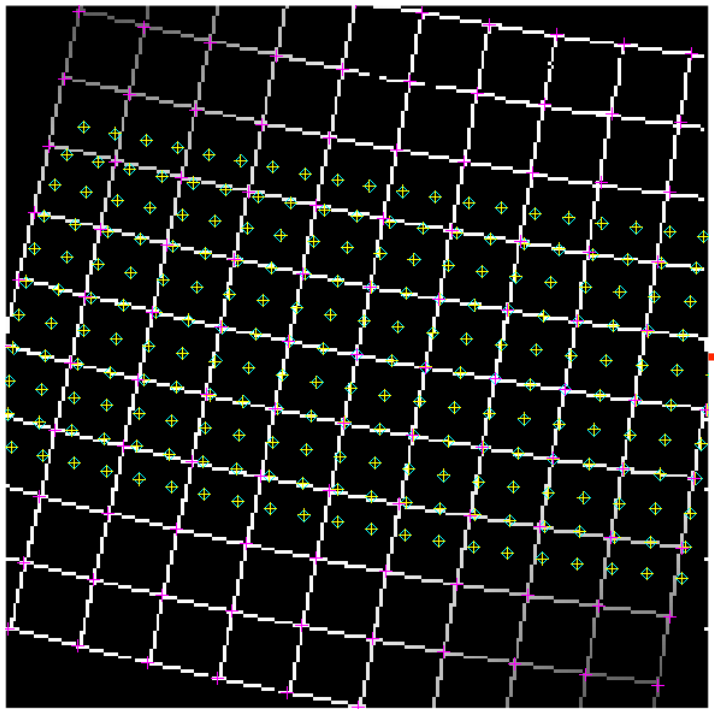

Original image

Rotation

# Geometrical distortion correction

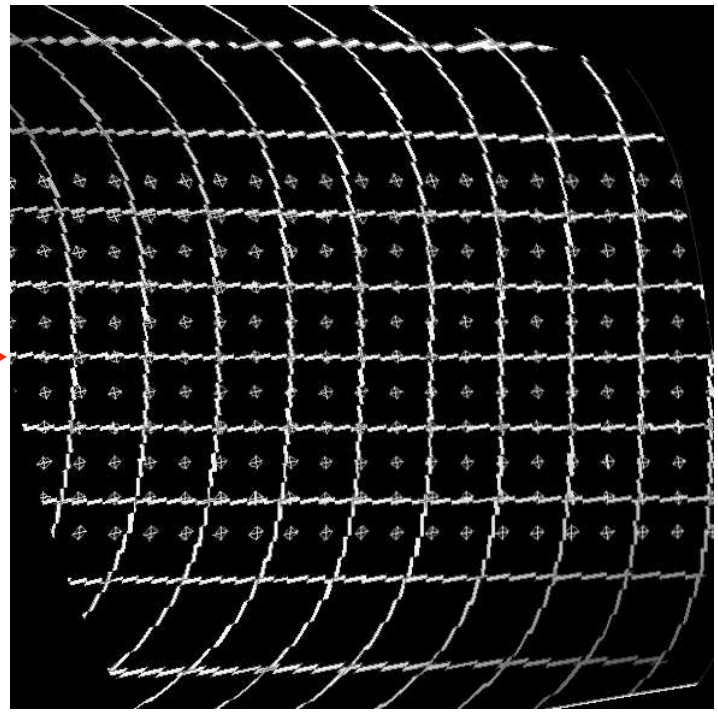
C/RTL cosimulation

Image resolution: 585x585 LineBuffer: 128 Lines



Original image

Remap



Rotation and distortion correction

# Vivado HLS Report Comparison

**solution1:** xc7z010clg400-1
**solution2:** xcku040-sfva784-1-i
**solution3:** xcku040-sfva784-3-e

**Timing (ns)**

| Clock | | solution1 | solution2 | solution3 |
|---|---|---|---|---|
| ap_clk | Target | 10.00 | 10.00 | 10.00 |
| | Estimated | 9.40 | 9.58 | 8.75 |

**Latency (clock cycles)**

| | | solution1 | solution2 | solution3 |
|---|---|---|---|---|
| Latency | min | 346323 | 346323 | 346323 |
| | max | 386417 | 386417 | 385119 |
| Interval | min | 346324 | 346324 | 346324 |
| | max | 386414 | 386414 | 385116 |

Resource Usage

| | solution1 | solution2 | solution3 |
|---|---|---|---|
| BRAM_18K | 66 | 42 | 42 |
| DSP48E | 10 | 10 | 10 |
| FF | 1658 | 1531 | 1453 |
| LUT | 2130 | 2128 | 2128 |

Resolution = 585x585 = 342225 pixels

# Toward the CDR

Timeframe: 2017-Q3 - end of September

- Functional requirements
- Functional prototype
- Hardware demonstrator
- Implementation plan