Duckietown FPGA Development

HLS Final Project

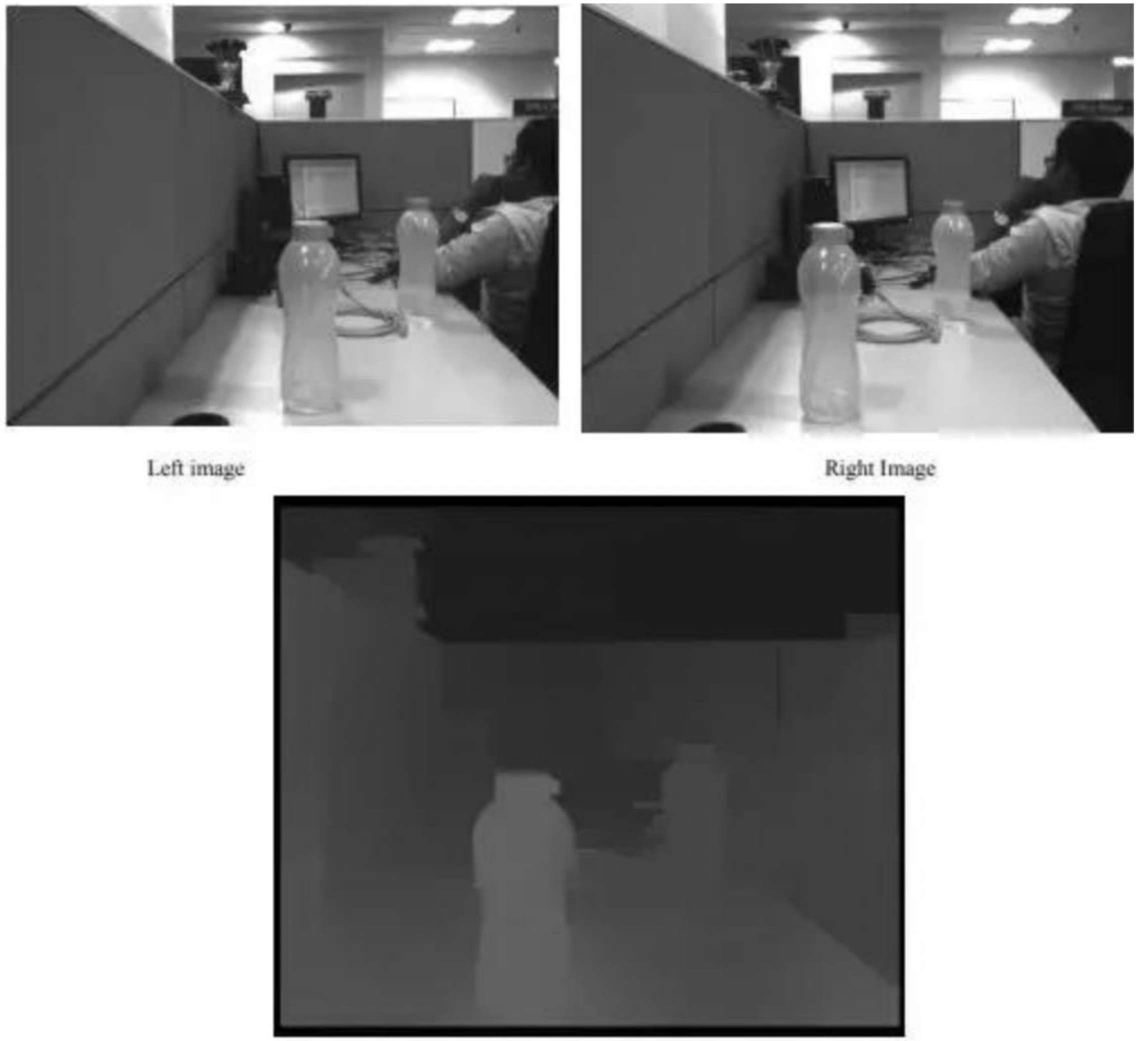
Team 2 陳聖文 徐志嘉

Outline

- Background
- Problem statement
- Project scope
- Debug & report & optimize
- Results

Background - SGBM

- Disparity Algorithm

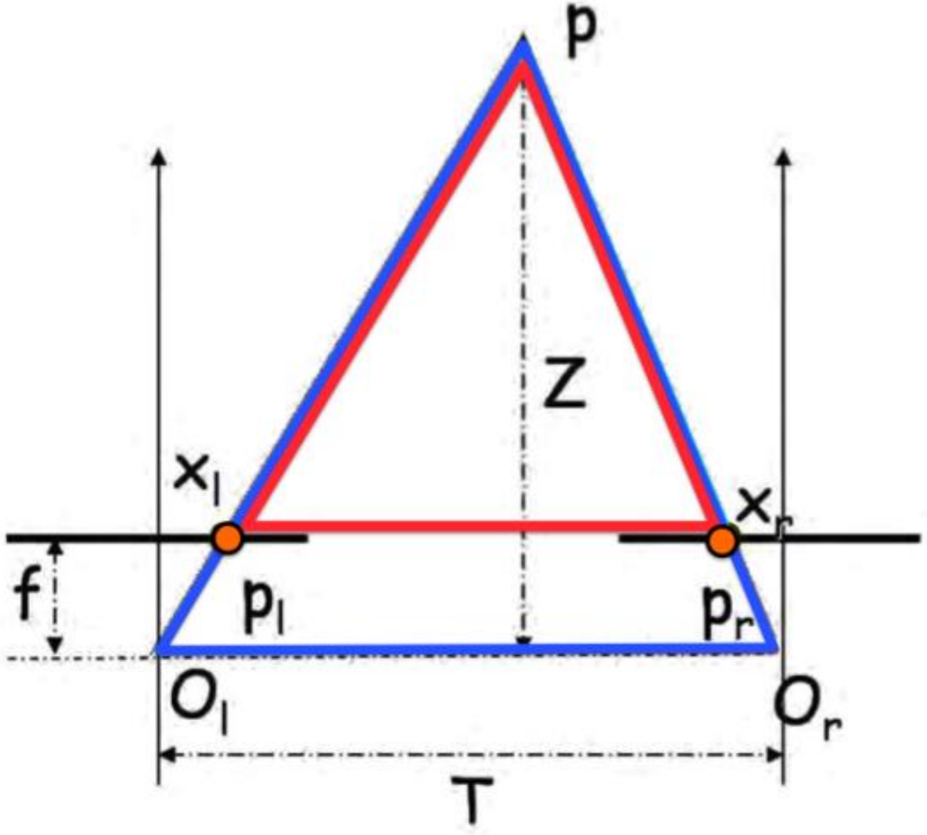


Left image

Right Image

Depth Image

Background – SGBM



Similar triangles:

$$\frac{T}{Z} = \frac{T + x_r - x_l}{Z - f}$$

$$\boxed{Z = \frac{f \cdot T}{x_l - x_r}}$$

baseline

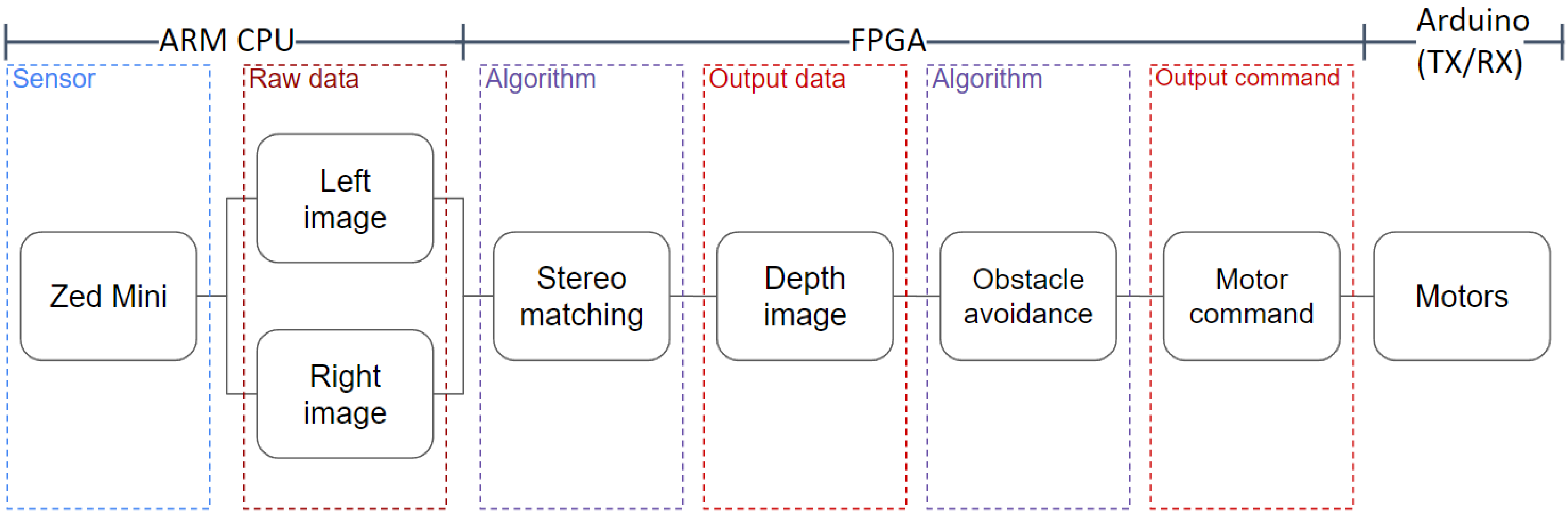
focal length

disparity

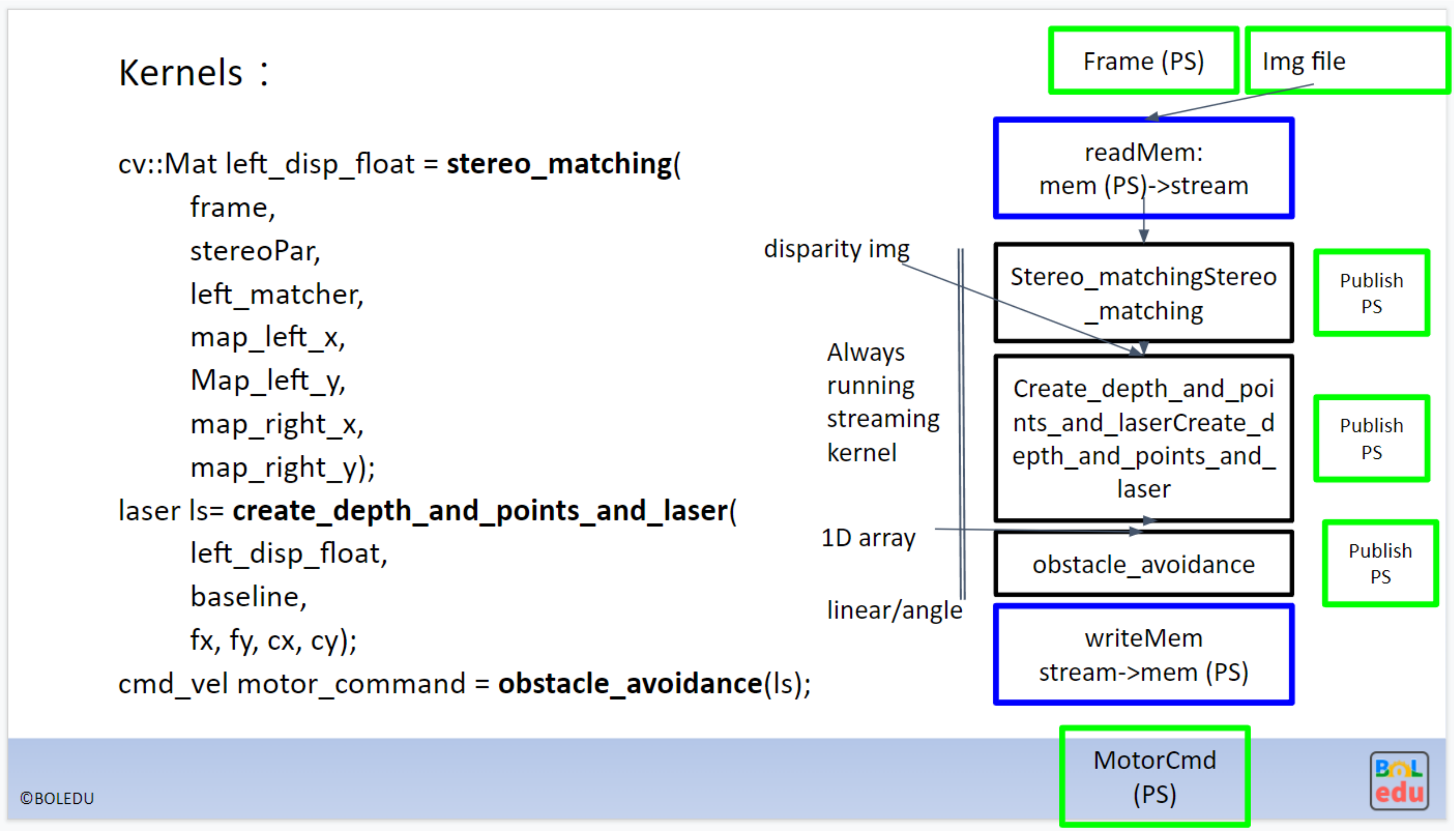
Problem statement

- Context: Obstacle avoidance by SGBM algorithm
- Issue: Only 2fps
- Objective: Real-time

Project scope



Project scope



Kernels：

```
cv::Mat left_disp_float = stereo_matching(
        frame,
        stereoPar,
        left_matcher,
        map_left_x,
        Map_left_y,
        map_right_x,
        map_right_y);
laser ls= create_depth_and_points_and_laser(
        left_disp_float,
        baseline,
        fx, fy, cx, cy);
cmd_vel motor_command = obstacle_avoidance(ls);
```

Frame (PS)

Img file

readMem:
mem (PS)->stream

disparity img

Stereo_matchingStereo_matching

Publish PS

Always running streaming kernel

Create_depth_and_points_and_laserCreate_depth_and_points_and_laser

Publish PS

1D array

obstacle_avoidance

Publish PS

linear/angle

writeMem
stream->mem (PS)

MotorCmd (PS)

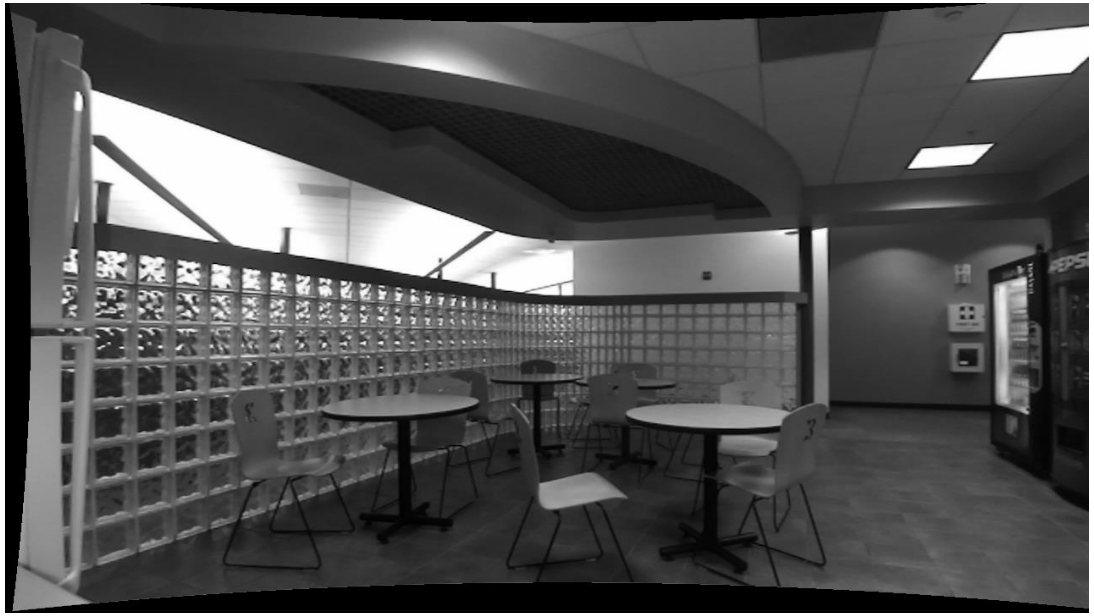
©BOLEDU

Project scope

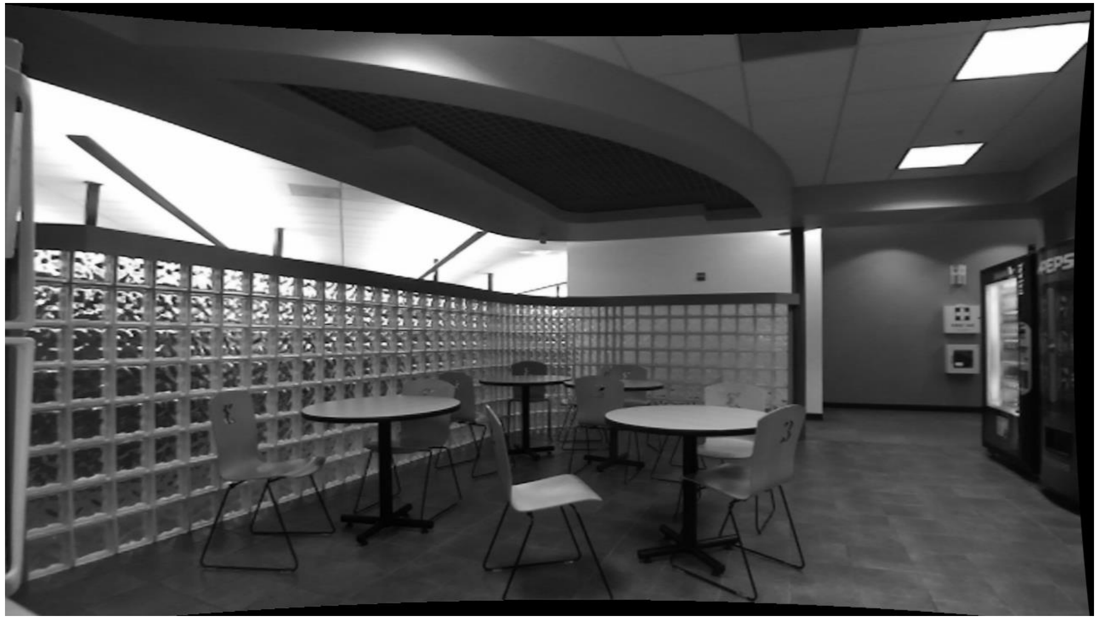
- Implement on KV260
- Target:
  - Real-time obstacle avoidance
  - At least 10fps

Input data (1280 x 720)

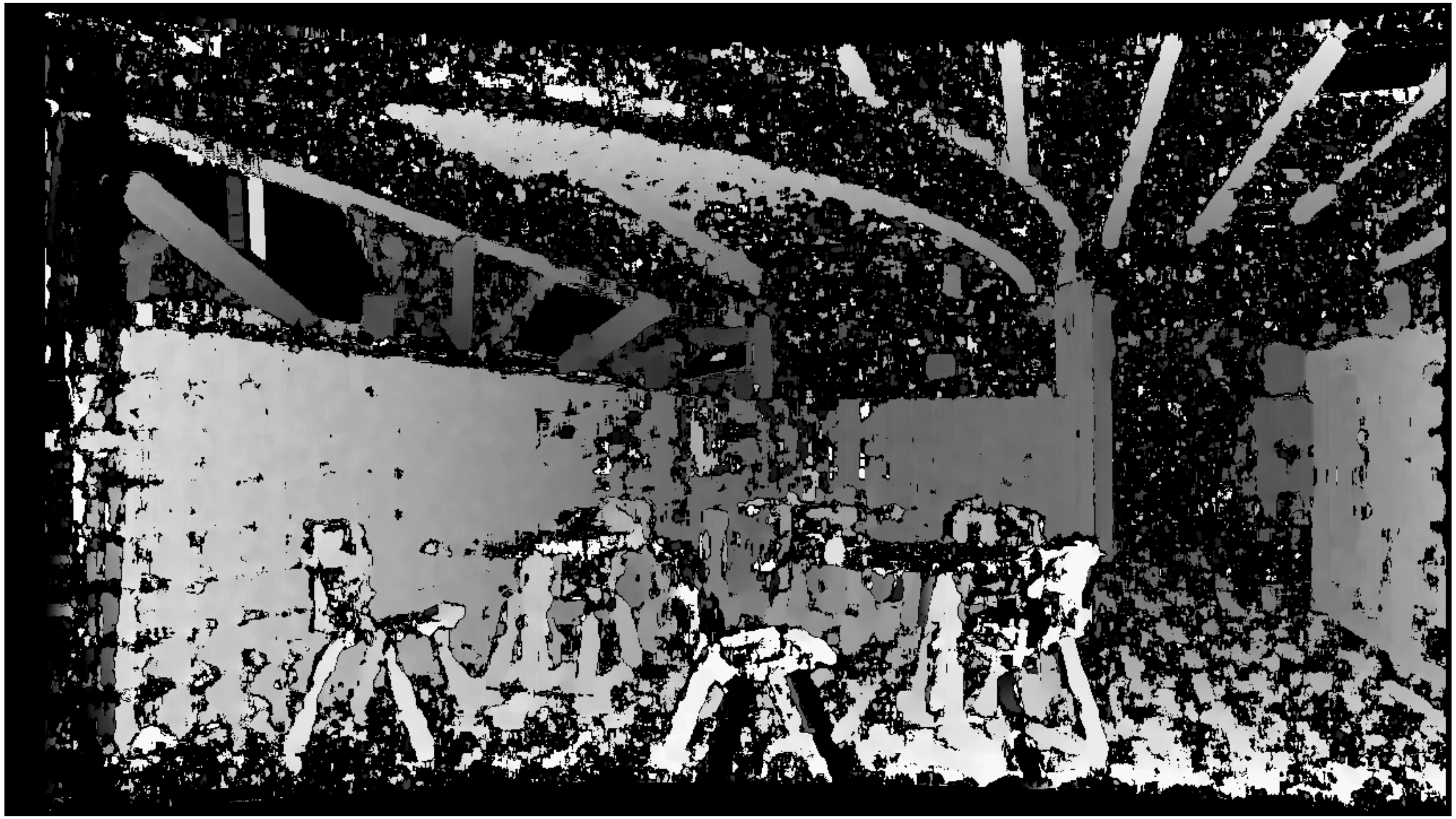
Left



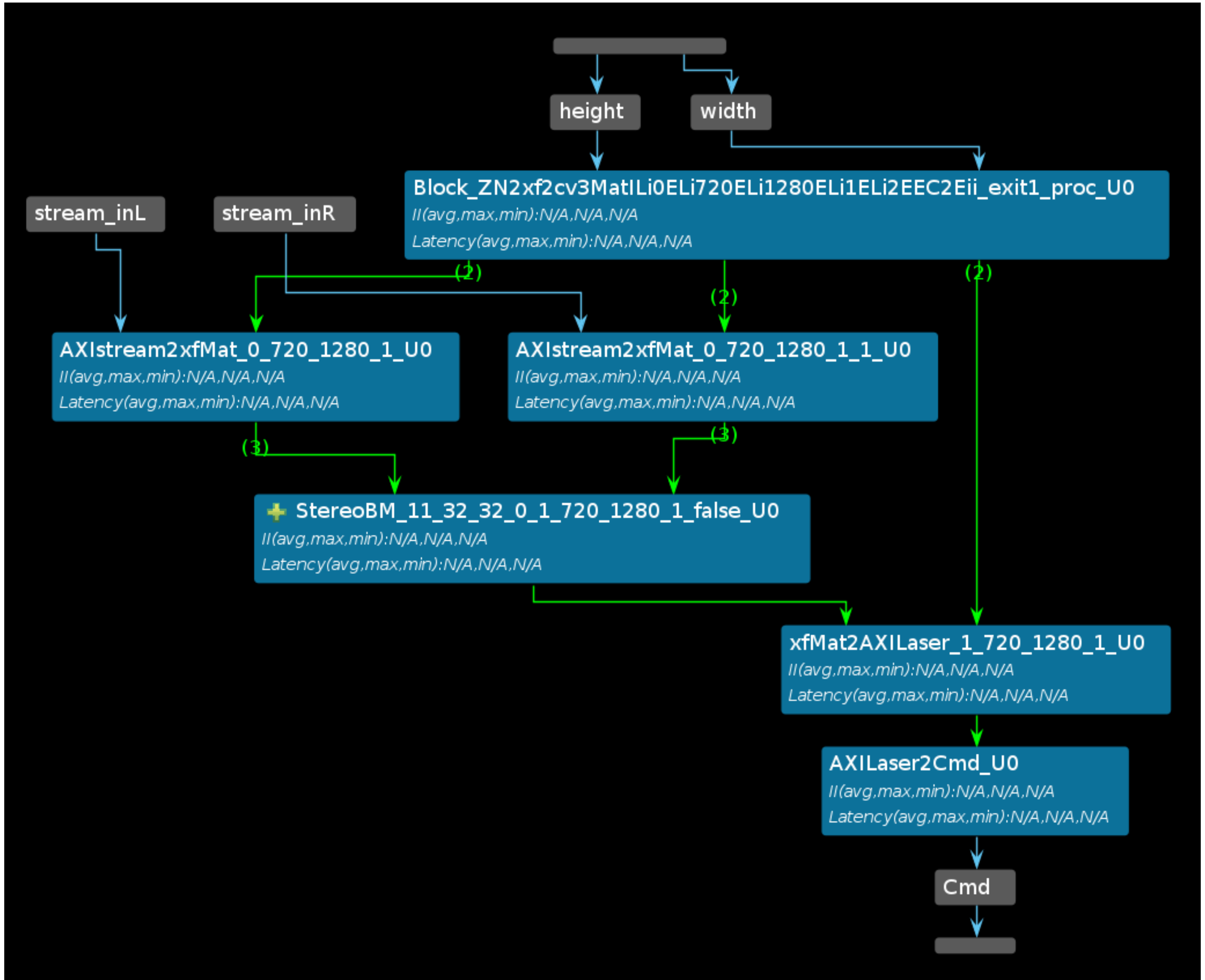
Right



Output disparity map

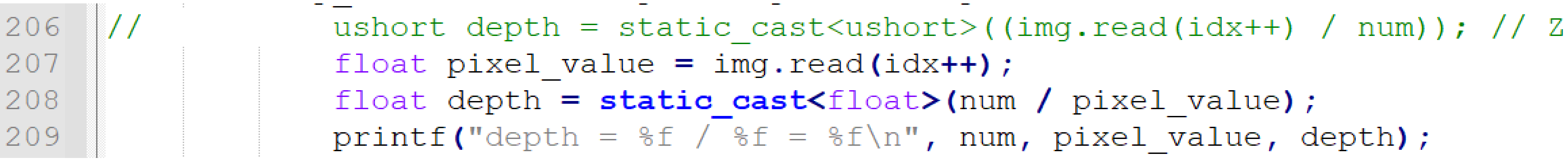


Dataflow

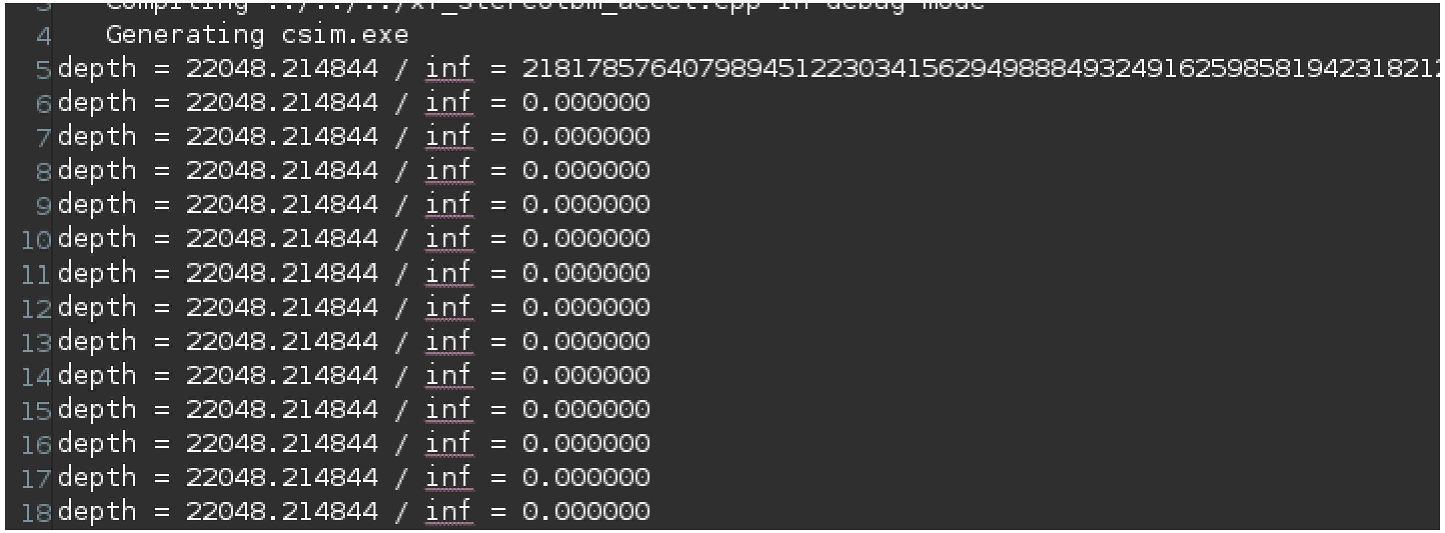

height
width
Block_ZN2xf2cv3MatILi0ELi720ELi1280ELi1ELi2EEC2Eii_exit1_proc_U0
II(avg,max,min):N/A,N/A,N/A
Latency(avg,max,min):N/A,N/A,N/A
stream_inL
stream_inR
(2)
(2)
(2)
AXIstream2xfMat_0_720_1280_1_U0
II(avg,max,min):N/A,N/A,N/A
Latency(avg,max,min):N/A,N/A,N/A
AXIstream2xfMat_0_720_1280_1_1_U0
II(avg,max,min):N/A,N/A,N/A
Latency(avg,max,min):N/A,N/A,N/A
(3)
(3)
StereoBM_11_32_32_0_1_720_1280_1_false_U0
II(avg,max,min):N/A,N/A,N/A
Latency(avg,max,min):N/A,N/A,N/A
xfMat2AXILaser_1_720_1280_1_U0
II(avg,max,min):N/A,N/A,N/A
Latency(avg,max,min):N/A,N/A,N/A
AXILaser2Cmd_U0
II(avg,max,min):N/A,N/A,N/A
Latency(avg,max,min):N/A,N/A,N/A
Cmd

Debug

- Not support ushort



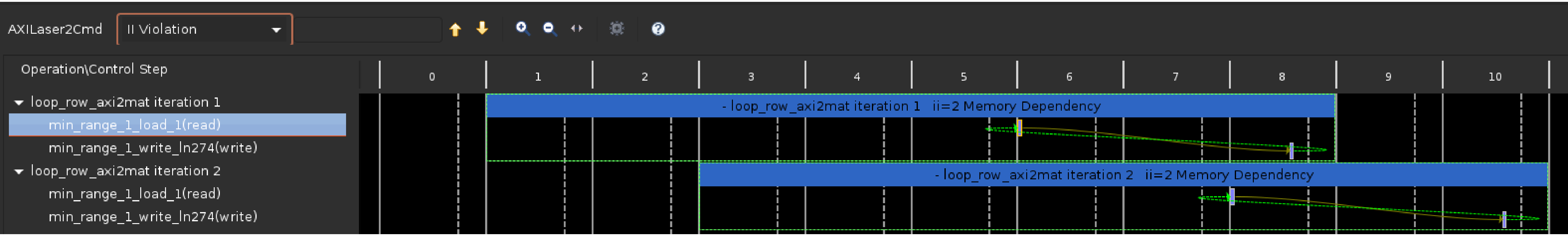
```
//              ushort depth = static_cast<ushort>((img.read(idx++) / num)); // Z
                float pixel_value = img.read(idx++);
                float depth = static_cast<float>(num / pixel_value);
                printf("depth = %f / %f = %f\n", num, pixel_value, depth);
```



```
   Compiling ../../../xf_stereobm_accel.cpp in debug mode
   Generating csim.exe
depth = 22048.214844 / inf = 218178576407989451223034156294988849324916259858194231821
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
depth = 22048.214844 / inf = 0.000000
```

Optimize: II Violation

- Memory dependency

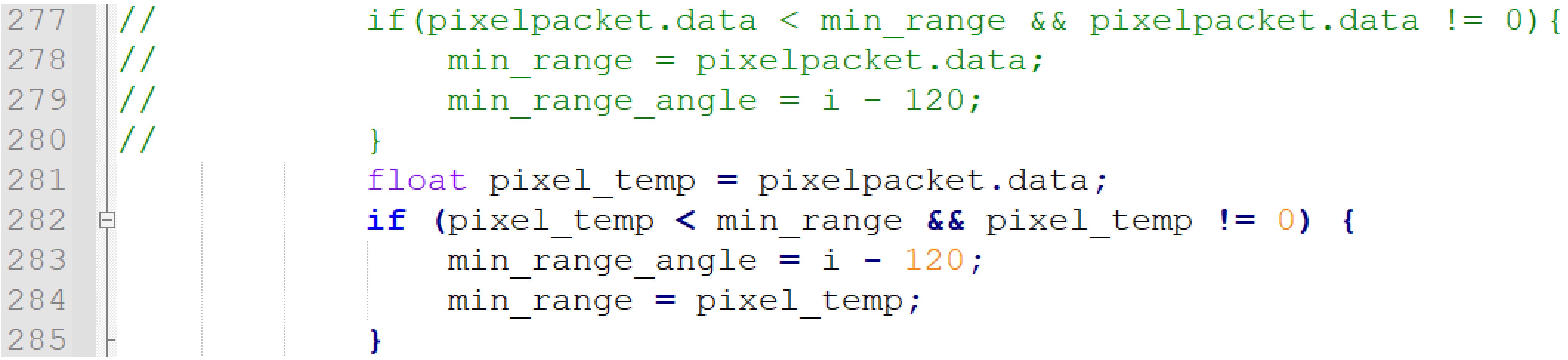


Operation\Control Step

- loop_row_axi2mat iteration 1
  - min_range_1_load_1(read)
  - min_range_1_write_ln274(write)
- loop_row_axi2mat iteration 2
  - min_range_1_load_1(read)
  - min_range_1_write_ln274(write)

- loop_row_axi2mat iteration 1 ii=2 Memory Dependency

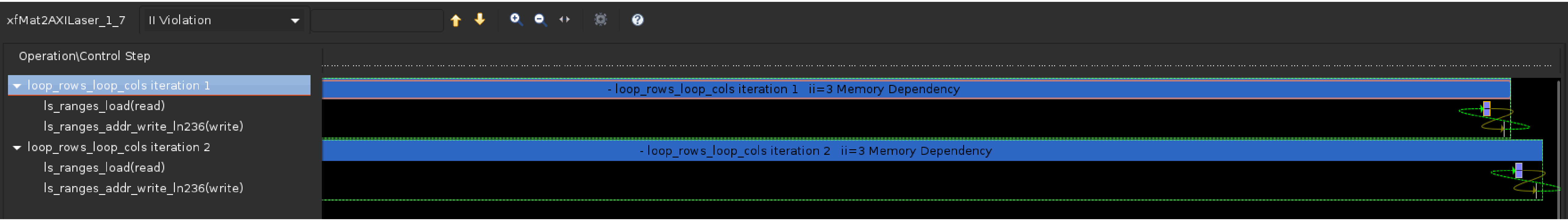
- loop_row_axi2mat iteration 2 ii=2 Memory Dependency



```
//          if(pixelpacket.data < min_range && pixelpacket.data != 0){
//              min_range = pixelpacket.data;
//              min_range_angle = i - 120;
//          }
            float pixel_temp = pixelpacket.data;
            if (pixel_temp < min_range && pixel_temp != 0) {
                min_range_angle = i - 120;
                min_range = pixel_temp;
            }
```

Optimize: II Violation

- Memory dependency



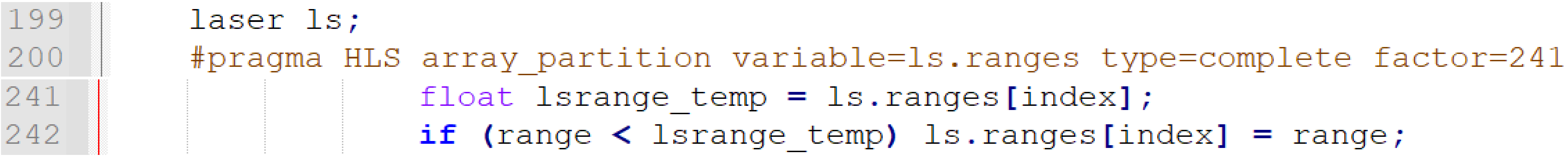
xfMat2AXILaser_1_7 II Violation

Operation\Control Step

- loop_rows_loop_cols iteration 1
  - ls_ranges_load(read)
  - ls_ranges_addr_write_ln236(write)
- loop_rows_loop_cols iteration 2
  - ls_ranges_load(read)
  - ls_ranges_addr_write_ln236(write)

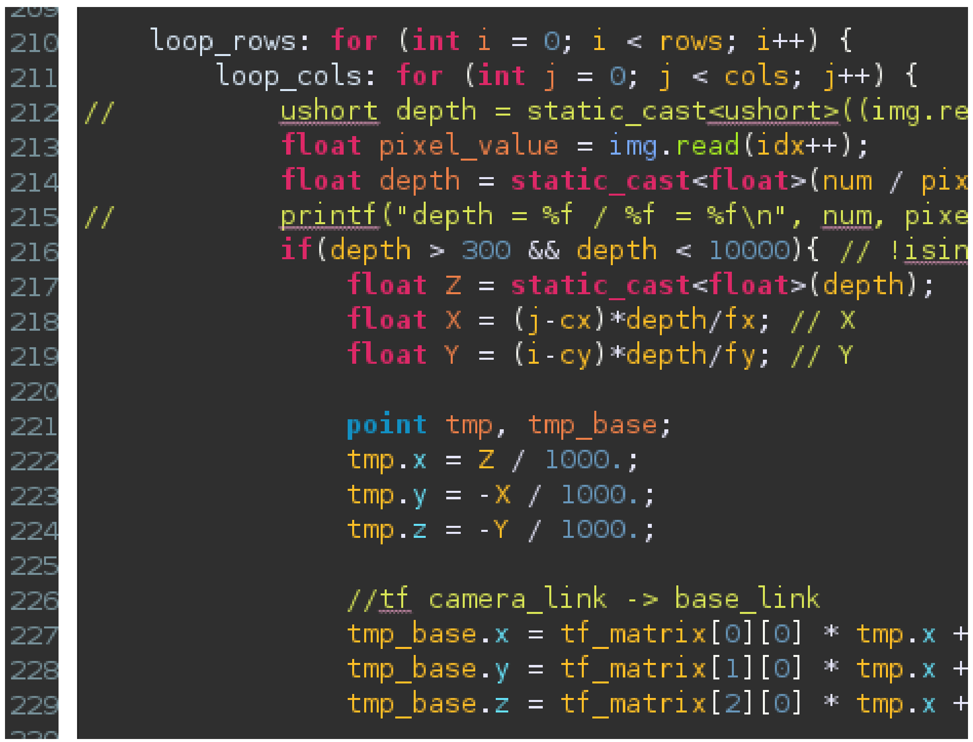
- loop_rows_loop_cols iteration 1   ii=3 Memory Dependency

- loop_rows_loop_cols iteration 2   ii=3 Memory Dependency

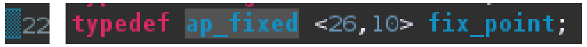


```
    laser ls;
    #pragma HLS array_partition variable=ls.ranges type=complete factor=241
                float lsrange_temp = ls.ranges[index];
                if (range < lsrange_temp) ls.ranges[index] = range;
```

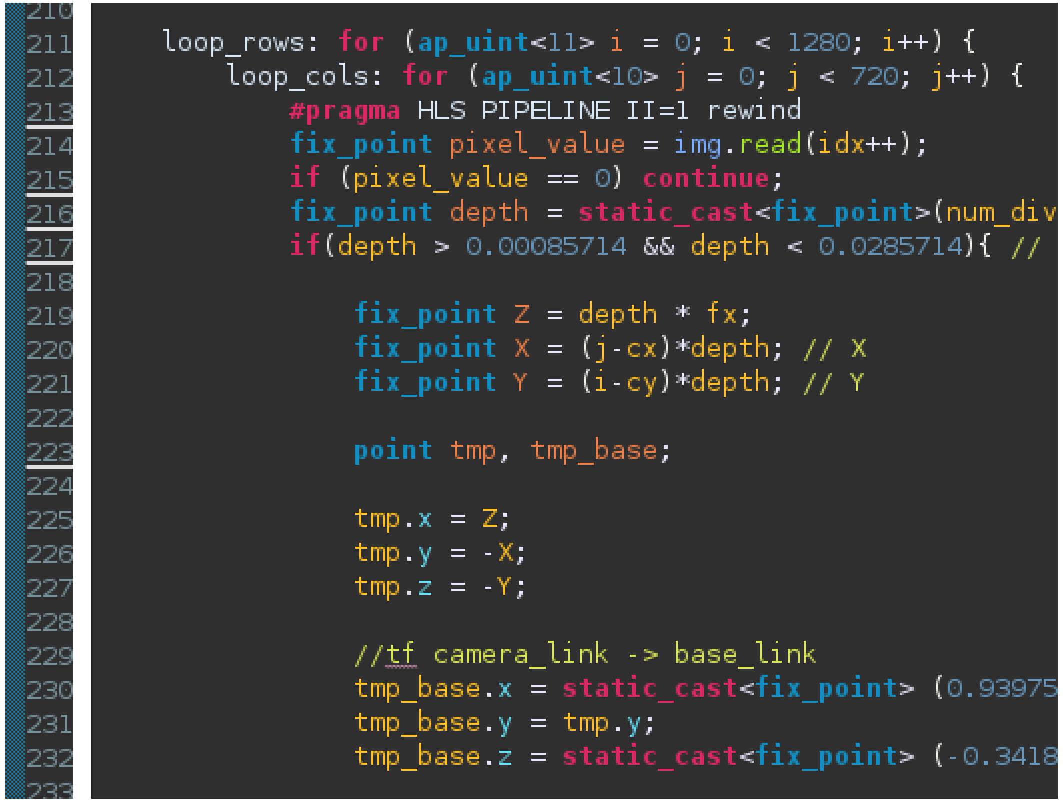
Optimize: fixed point



```
    loop_rows: for (int i = 0; i < rows; i++) {
        loop_cols: for (int j = 0; j < cols; j++) {
//          ushort depth = static_cast<ushort>((img.re
            float pixel_value = img.read(idx++);
            float depth = static_cast<float>(num / pix
//          printf("depth = %f / %f = %f\n", num, pixe
            if(depth > 300 && depth < 10000){ // !isin
                float Z = static_cast<float>(depth);
                float X = (j-cx)*depth/fx; // X
                float Y = (i-cy)*depth/fy; // Y

                point tmp, tmp_base;
                tmp.x = Z / 1000.;
                tmp.y = -X / 1000.;
                tmp.z = -Y / 1000.;

                //tf camera_link -> base_link
                tmp_base.x = tf_matrix[0][0] * tmp.x +
                tmp_base.y = tf_matrix[1][0] * tmp.x +
                tmp_base.z = tf_matrix[2][0] * tmp.x +
```



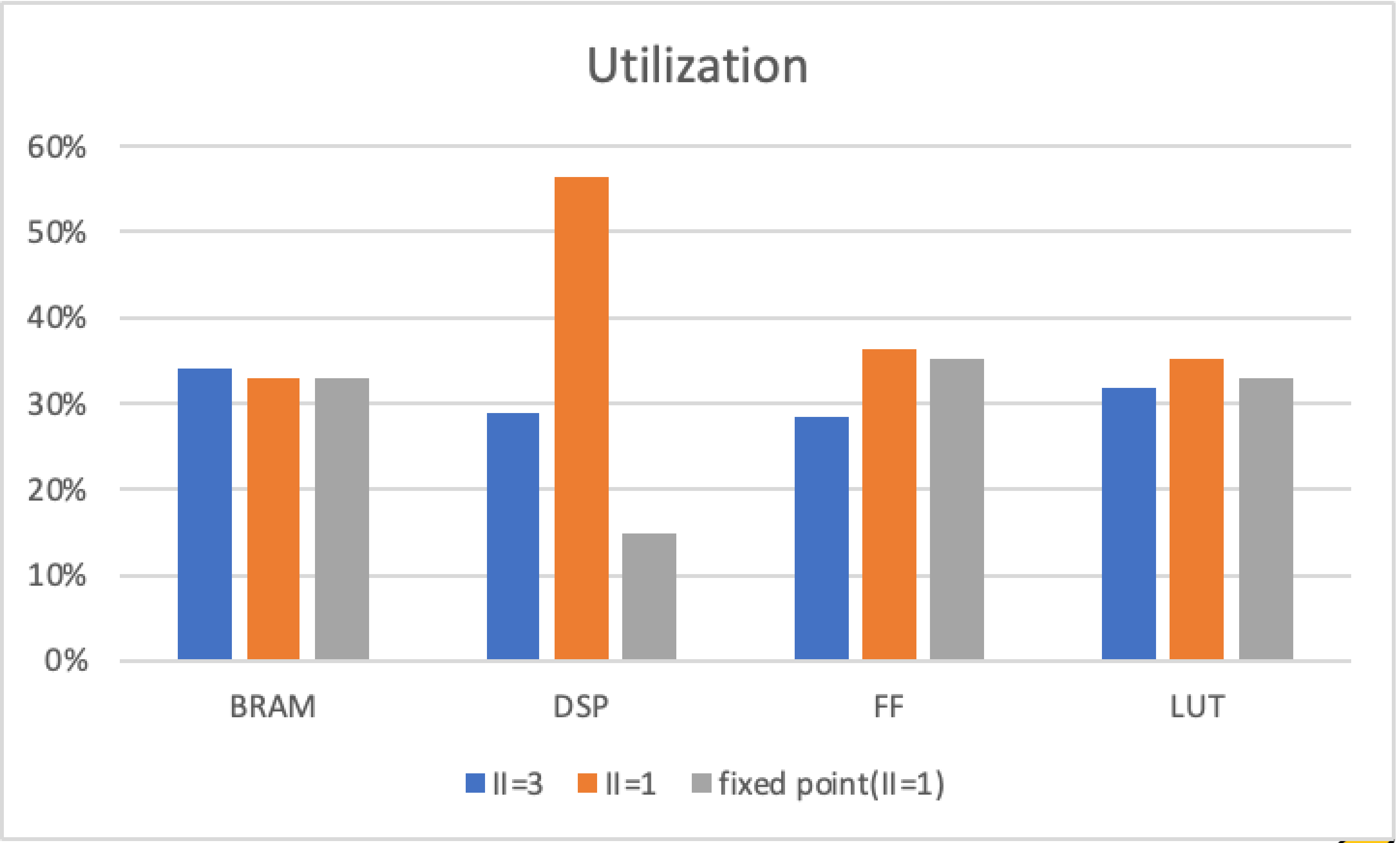
```
typedef ap_fixed <26,10> fix_point;
```



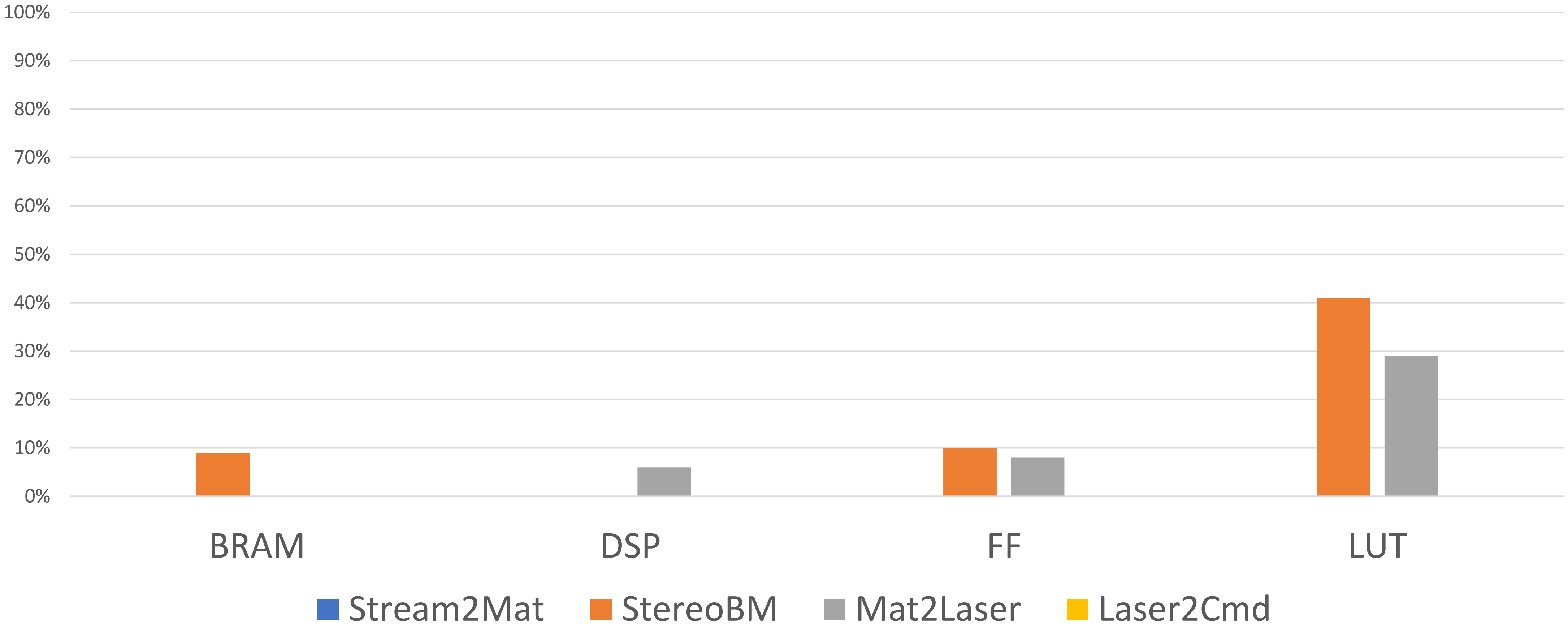
```
    loop_rows: for (ap_uint<11> i = 0; i < 1280; i++) {
        loop_cols: for (ap_uint<10> j = 0; j < 720; j++) {
            #pragma HLS PIPELINE II=1 rewind
            fix_point pixel_value = img.read(idx++);
            if (pixel_value == 0) continue;
            fix_point depth = static_cast<fix_point>(num_div
            if(depth > 0.00085714 && depth < 0.0285714){ //

                fix_point Z = depth * fx;
                fix_point X = (j-cx)*depth; // X
                fix_point Y = (i-cy)*depth; // Y

                point tmp, tmp_base;

                tmp.x = Z;
                tmp.y = -X;
                tmp.z = -Y;

                //tf camera_link -> base_link
                tmp_base.x = static_cast<fix_point> (0.93975
                tmp_base.y = tmp.y;
                tmp_base.z = static_cast<fix_point> (-0.3418
```

Utilization Comparison

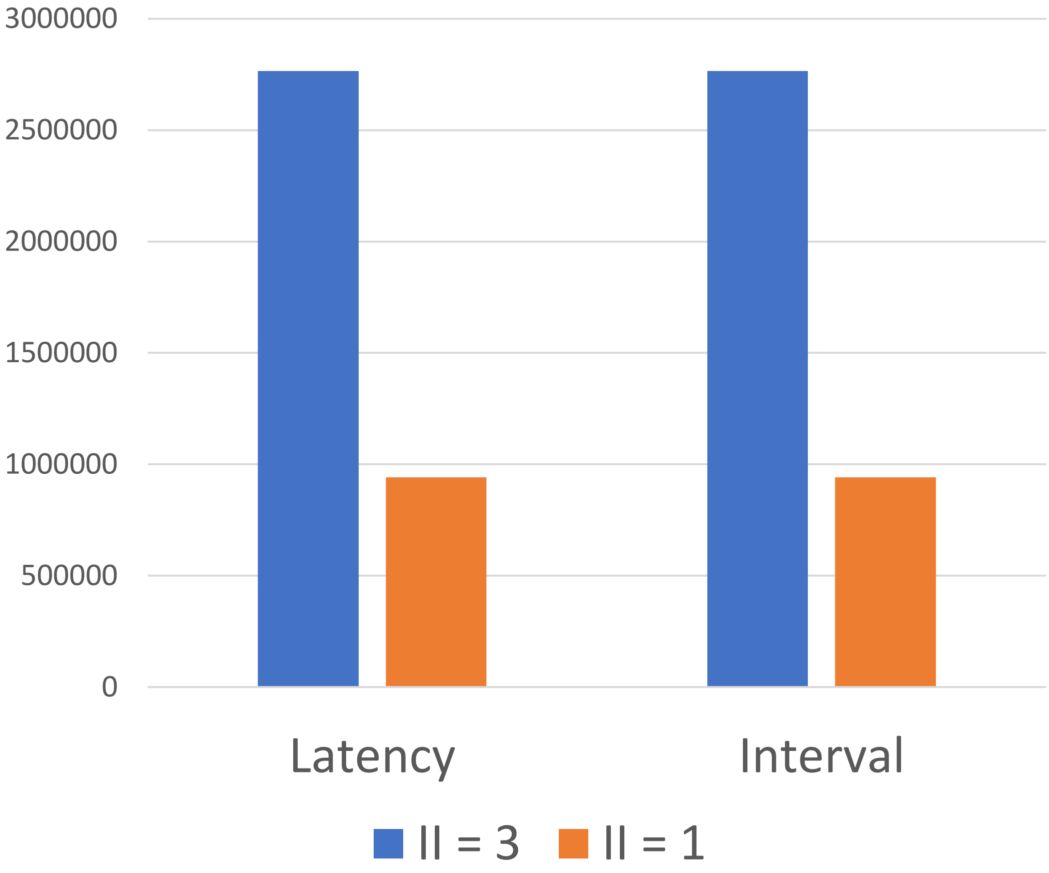
	BRAM	DSP	FF	LUT
II=3	27	43	36330	76307
II=1	26	84	46495	84890
fixed point(II=1)	26	22	44959	79286

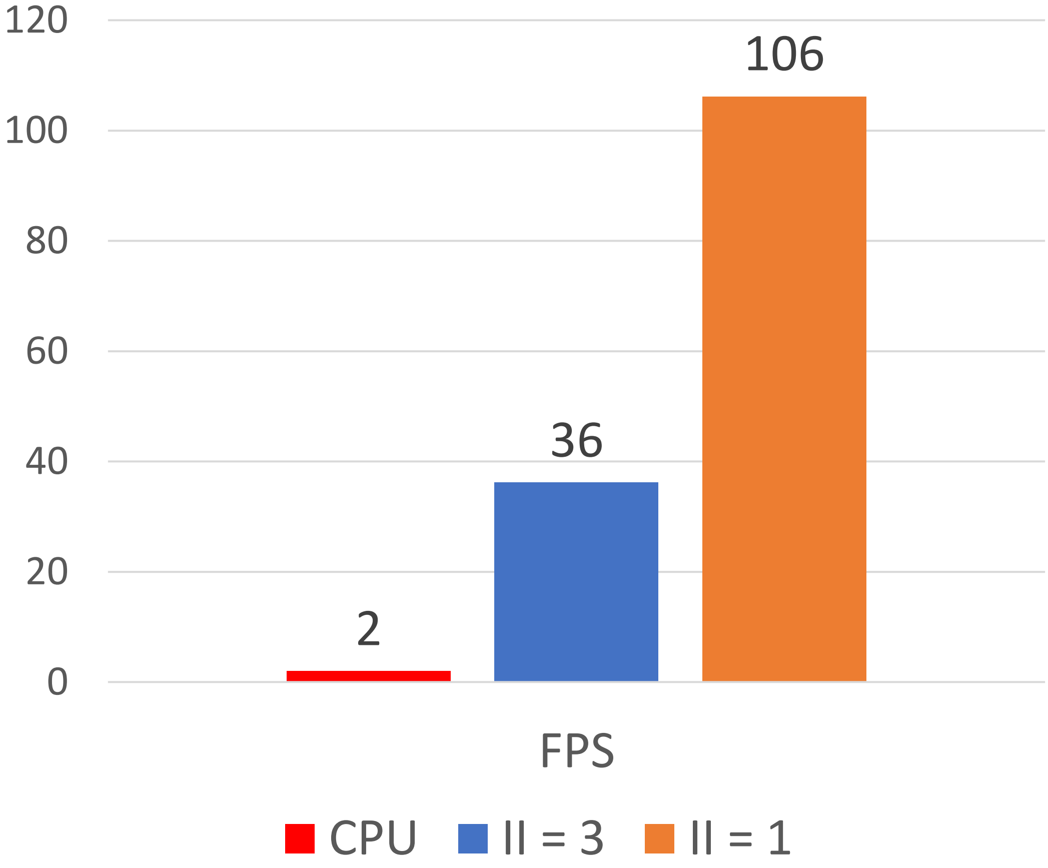


Area percentage of each function

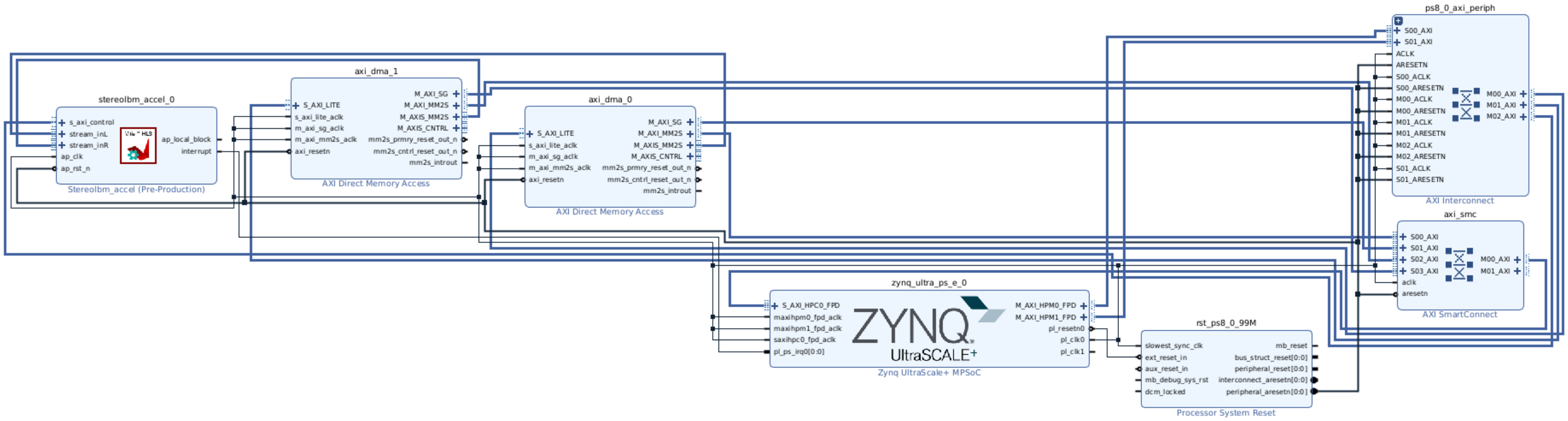


Performance





Implementation



Future Work

- One way
  - Pynq with ROOT permission
  - ROS with Python
- Another Way
  - XRT on KV260
  - ROS with C++

GitHub

- hsu880105/HLS_Final_Project (github.com)