

Internship

SDSoC

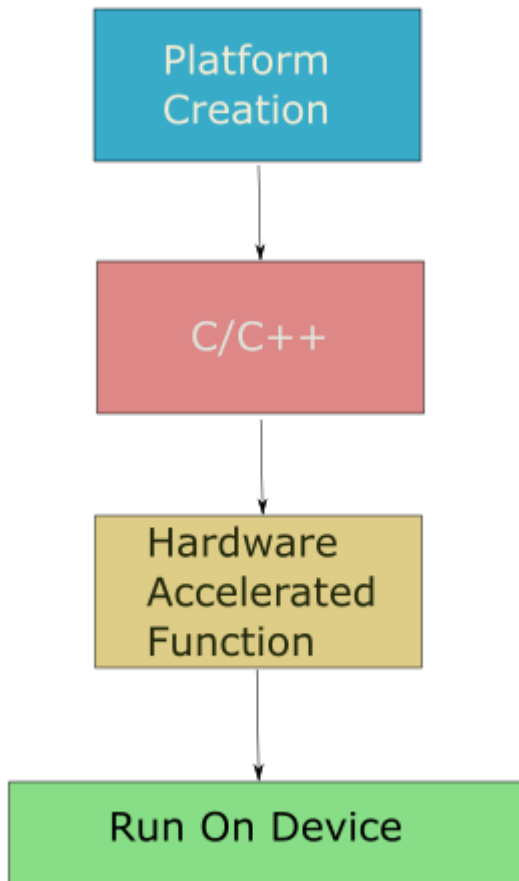
Andrew Maclellan



Introduction

- What is SDSoC?
 - Software Defined System on Chip.
 - Accelerate software functions (C/C++) within the FPGA on Zynq devices.
 - Aimed at software engineers with limited hardware knowledge.
 - Runs on top of Vivado, Vivado HLS, and SDK.

SDSoC Structure



Platform created in Vivado and linked to SDSoC.

C/C++ program written in SDSoC.

selected functions are built into hardware.

Run on device through JTAG or booted from SD card.

SDSoC Features

- Hardware acceleration
- Automatically creates boot files for baremetal, linux, and FreeRTOS.
- Vivado HLS pragmas. (Pipelining, unrolling etc...)
- Estimation Reports (Comparing software only to hardware accelerated).
- Axi Performance Interface.

Learning

- Zynq Book.
- 2 day training course SDSoC (Rutherford Appleton Laboratory).
- Creating own platform.
- Using HLS pragmas.
- Writing software to be synthesisable in hardware.



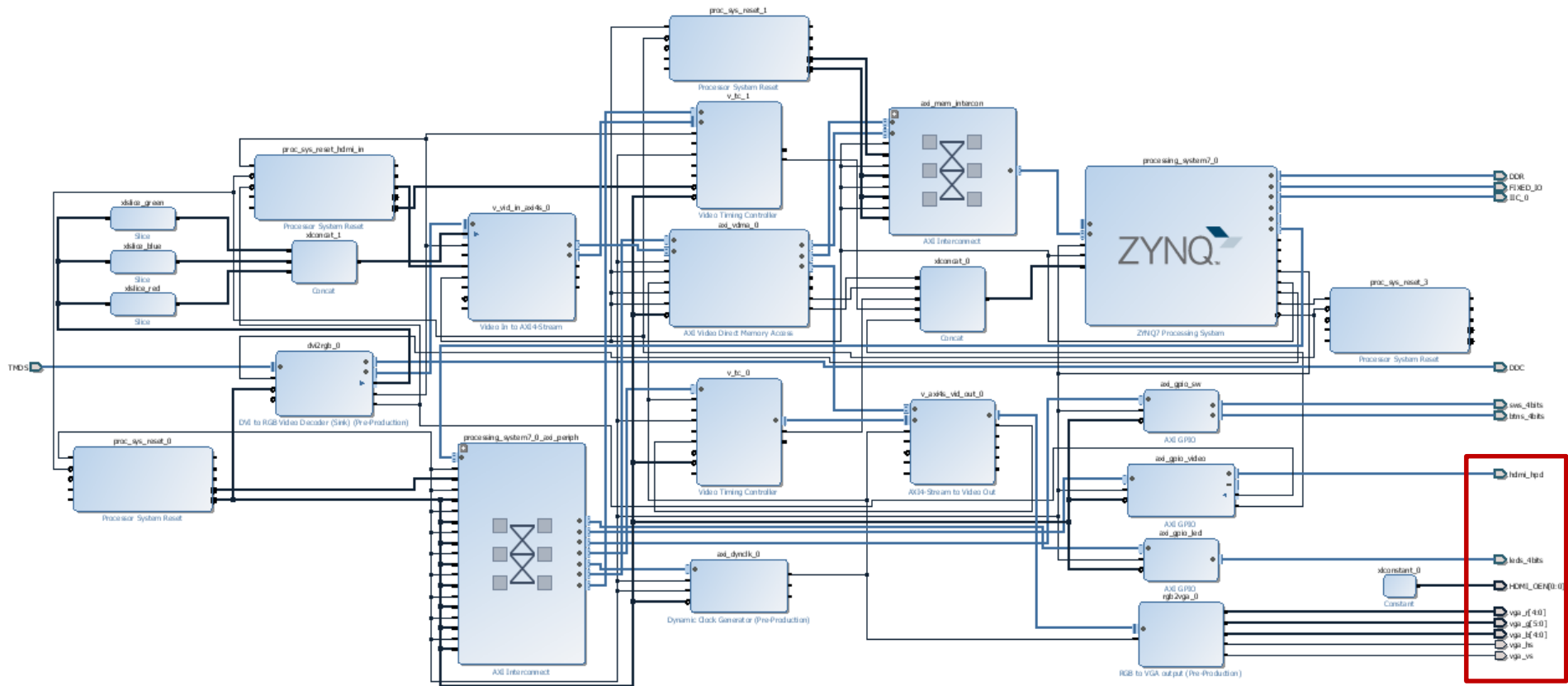
Application of SDSoC

- Zybo.
- Image processing.
- Following SDSoC structure.
- Program for hardware.
- Create filters for:
 - Pass-through and colour change.
 - Greyscale.
 - Average Filter.
 - Sobel Filter.

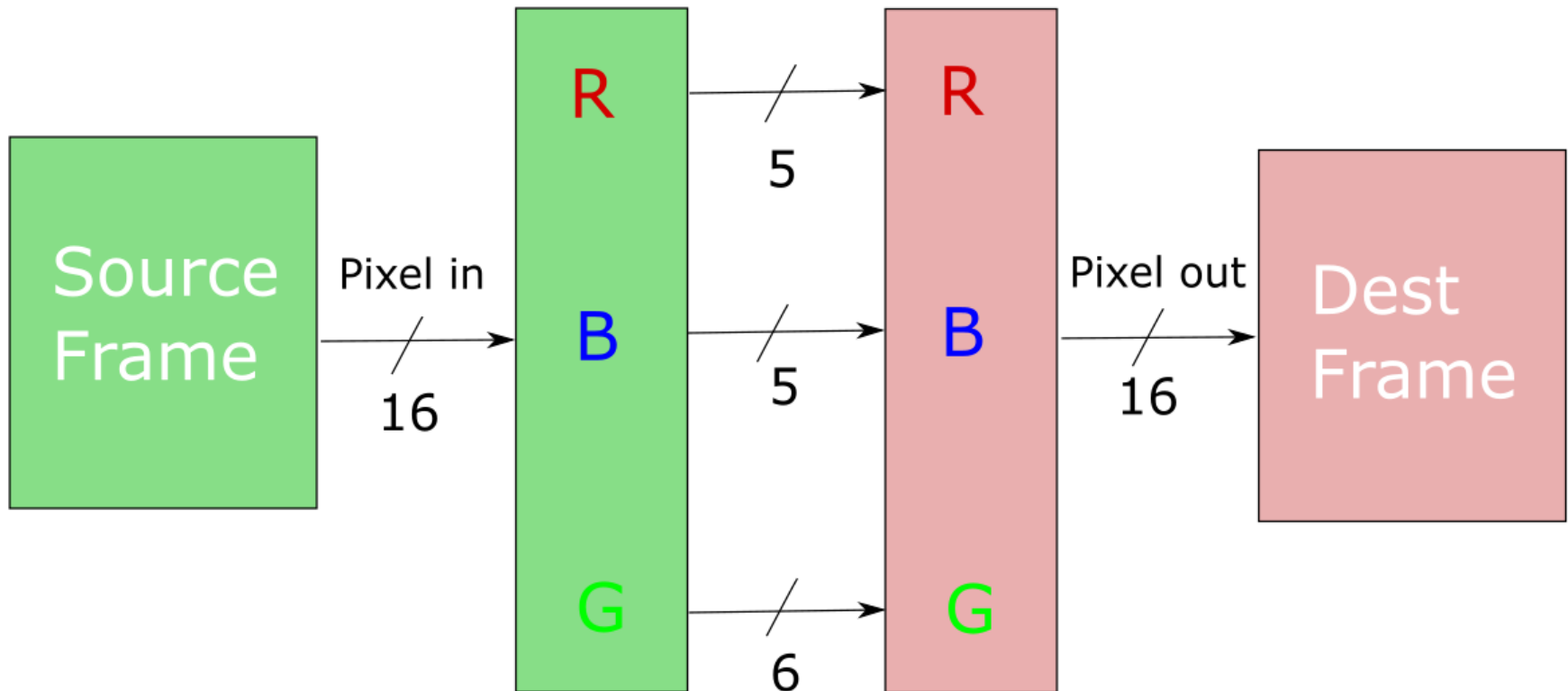


Platform

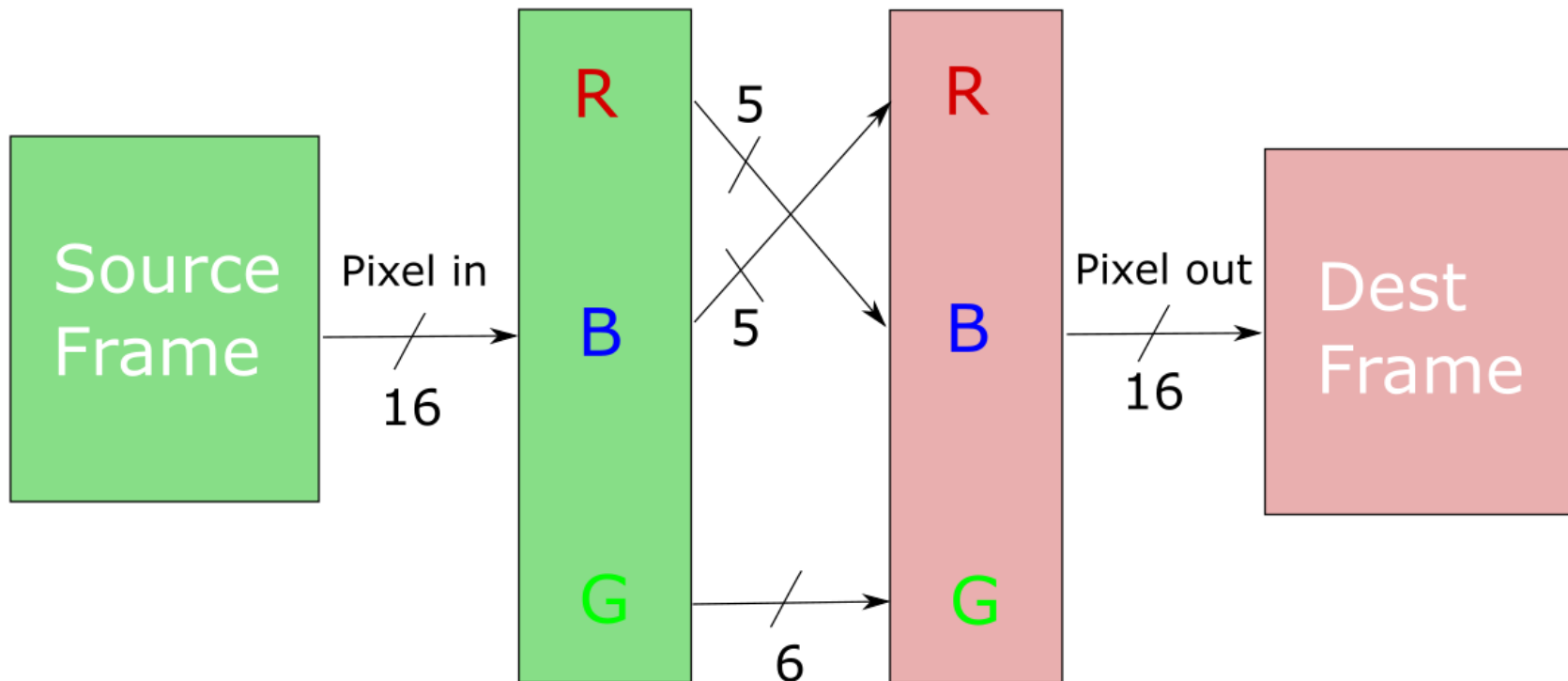
- Zybo Digilent platform.
- HDMI_in, VGA_out, CLKs, GPIO...



Pass-through

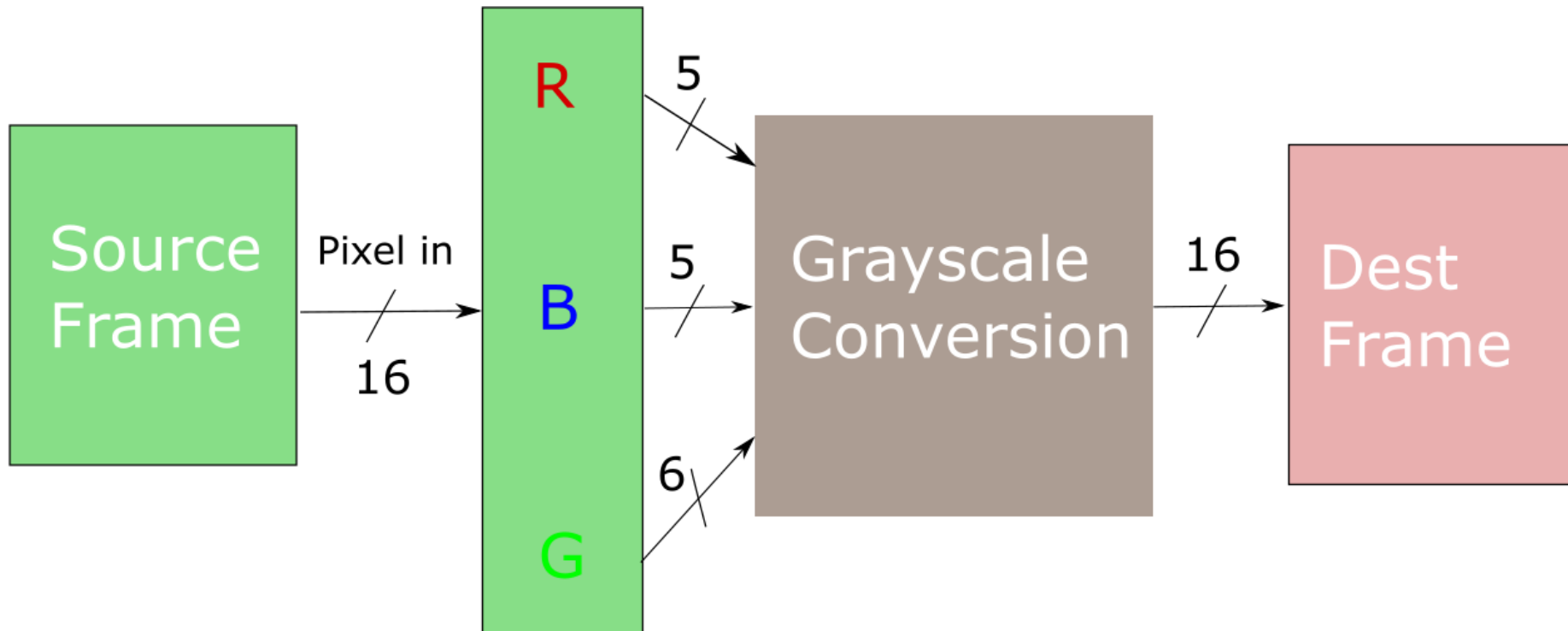


Pass-through





Grayscale



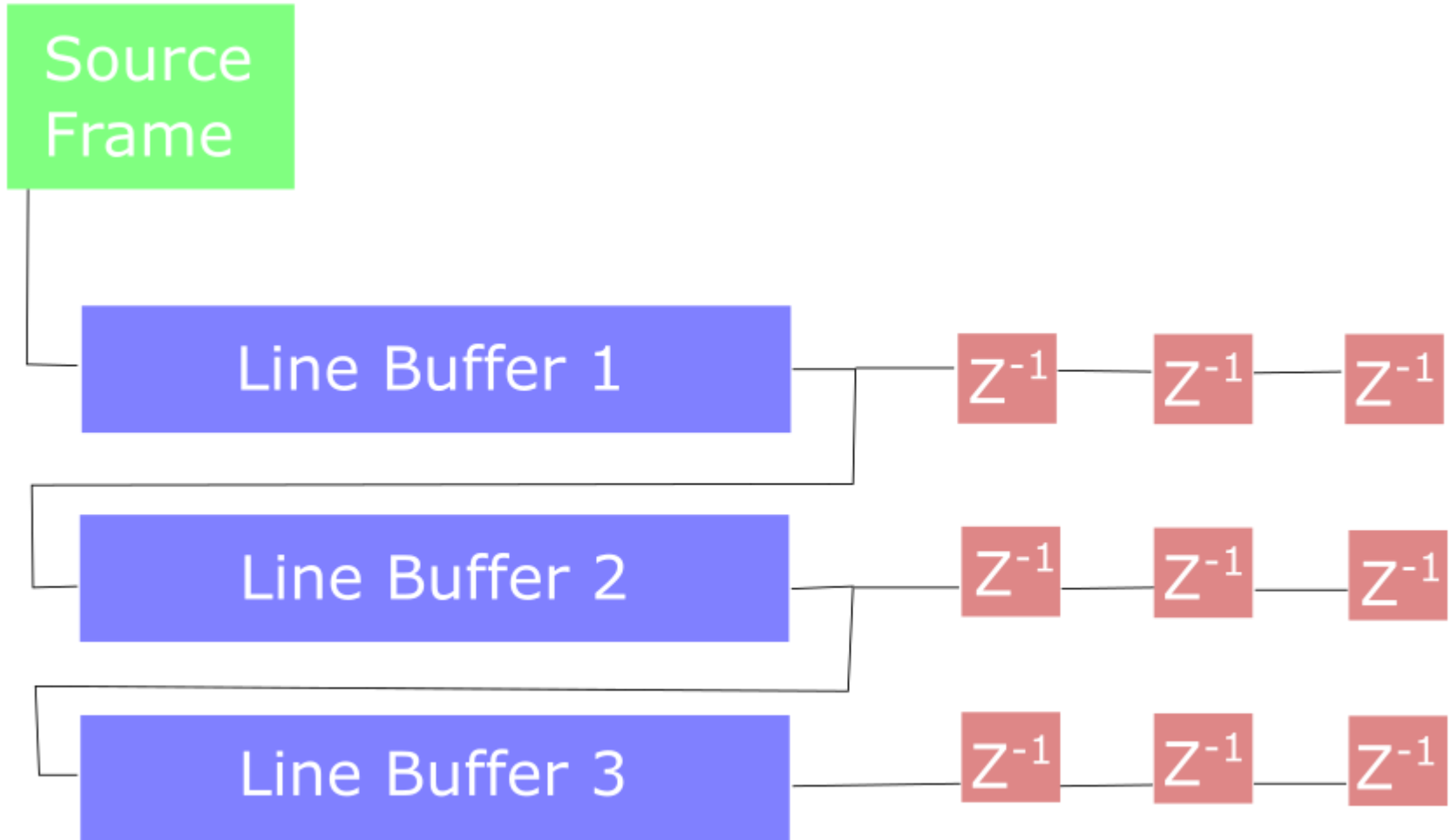
Greyscale



$$Y = (R * 76) + (B * 150) + (G * 29 + 128) >> 8$$

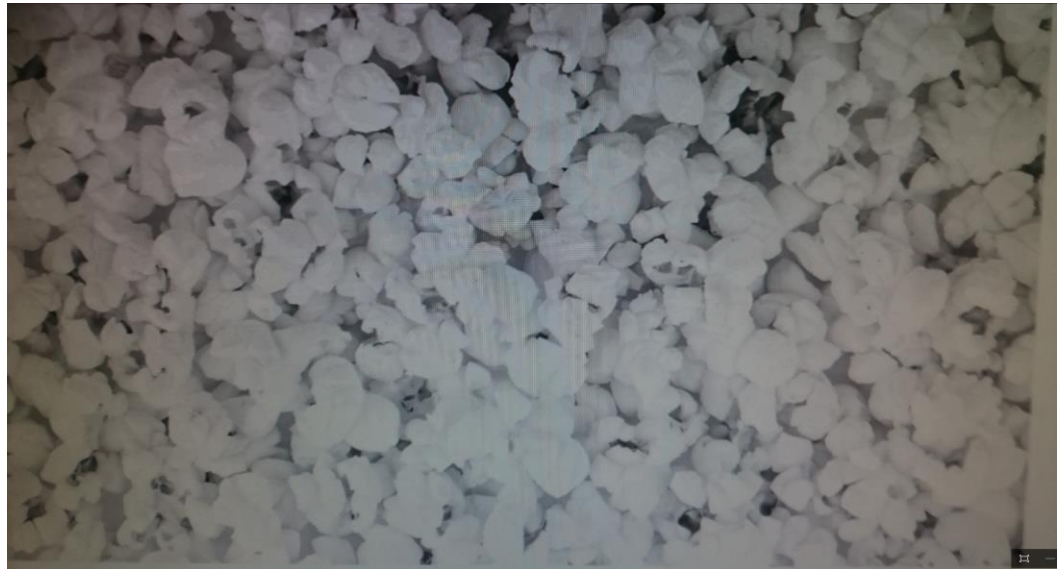


Average Filter

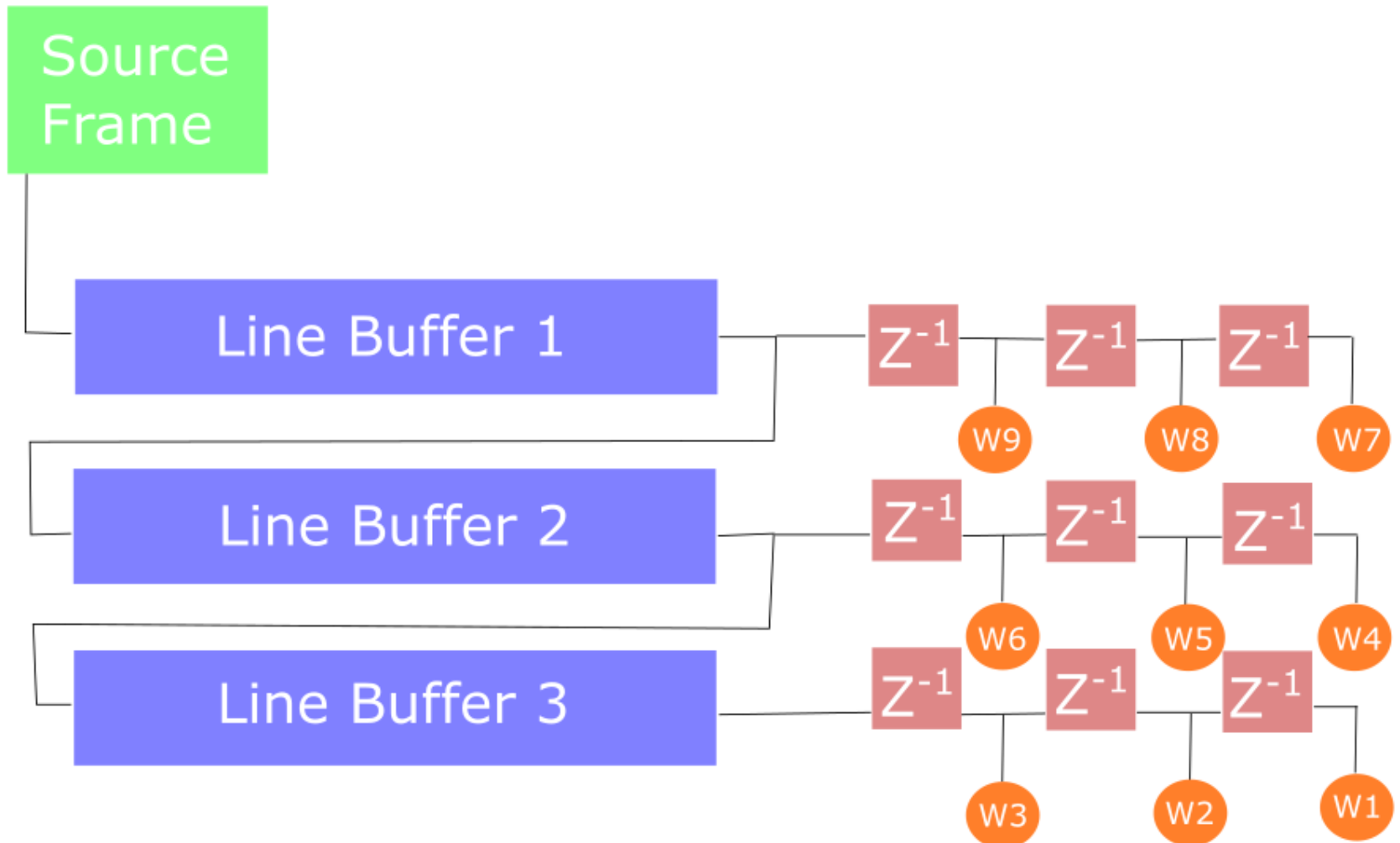


Average Filter

$$Z_{\text{Average}} = (z_1 + z_2 + z_3 + z_4 + z_5 + z_6 + z_7 + z_8 + z_9) * 1/9$$



Sobel Filter



Sobel Filter

Gx

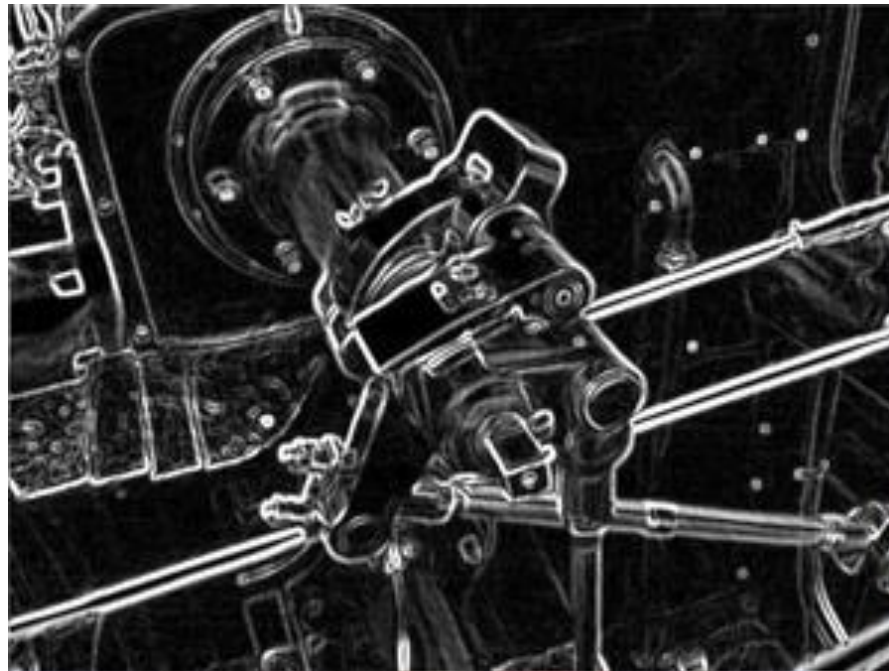
-1	0	-1
-2	0	-2
1	0	1

Gy

1	2	1
0	0	0
-1	-2	-1

Sobel Filter

$$G_mag = ABS(Gx) + ABS(Gy)$$



[https://upload.wikimedia.org/wikipedia/commons/thumb/d/d4/Valve_sobel_\(3\).PNG/300px-Valve_sobel_\(3\).PNG](https://upload.wikimedia.org/wikipedia/commons/thumb/d/d4/Valve_sobel_(3).PNG/300px-Valve_sobel_(3).PNG)

Sobel Filter

- Future implementation:
 - Edge thresholding.
- Gradient threshold at 100.
- ≥ 100 is set as white.
- < 100 is set as black.

Hardware implementation

- Using HLS pragmas:

Pipeline

```
for(ycor = 0; ycor < DEMO_HEIGHT; ycor++){  
    for(xcor = 0; xcor < DEMO_WIDTH; xcor++){  
#pragma HLS PIPELINE II=1
```

Unroll

```
        for(i = 0; i < 3; i++){  
#pragma HLS UNROLL  
            for(j = 0; j < 3; j++){  
#pragma HLS UNROLL
```

Memory attributes and access patterns

```
#pragma SDS data mem_attribute(srcFrame:PHYSICAL_CONTIGUOUS, dstFrame:PHYSICAL_CONTIGUOUS)  
#pragma SDS data access_pattern(srcFrame:SEQUENTIAL, dstFrame:SEQUENTIAL)  
void SobelFrameHw(u16 srcFrame[DEMO_PIXELS], u16 dstFrame[DEMO_PIXELS]);
```

Any Questions?





University of **Strathclyde** Glasgow