

Edge detection in a FPGA

EE 564

Hasith Perera

What I wanted to achieve...

- Basically wanted to implement a simple 2D image processing algorithm in an FPGA
 - Needed to get an image in to the FPGA
 - Perform the actual processing
 - Get the result back to a computer
- Wanted do this whole sequence in an FPGA
- Chose an Edge detector as my algorithm. Specifically a **Canny-Deriche** edge detector

Canny - Deriche Edge detector

- Introduced in 1987 by Rachid Deriche.
- Basic design - Has 4 stages that can be expressed as IIR filters
- Advantages
 - Has one parameter alpha which can change the smoothing/detection
 - Has a fixed processing time.

Canny - Deriche Edge detector (contd)

$$y_j^1 = a_1x_j + a_2x_{j-1} + b_1y_{j-1}^1 + b_2y_{j-2}^1 \quad (1)$$

$$y_j^2 = a_3x_{j+1} + a_4x_{j+2} + b_1y_{j+1}^2 + b_2y_{j+2}^2 \quad (2)$$

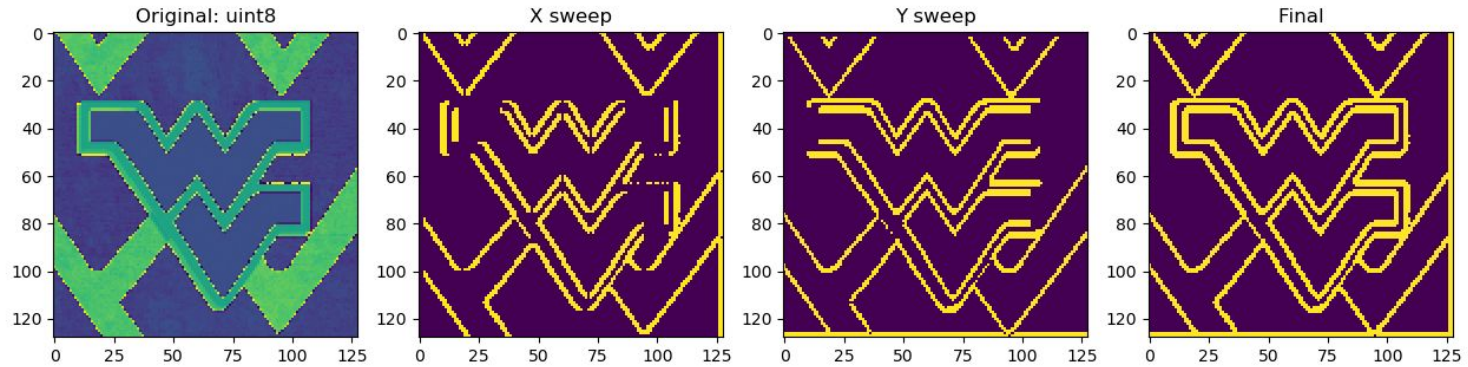
$$\theta_j = c_1(y_j^1 + y_j^2) \quad (3)$$

$$y_j^1 = a_5\theta_j + a_6\theta_{j-1} + b_1y_{j-1}^1 + b_2y_{j-2}^1 \quad (4)$$

$$y_j^2 = a_7\theta_{j+1} + a_8\theta_{j+2} + b_1y_{j+1}^2 + b_2y_{j+2}^2 \quad (5)$$

$$\Theta_j = c_1(y_j^1 + y_j^2) \quad (6)$$

CPU implementation



My implementation : [on Github](#)

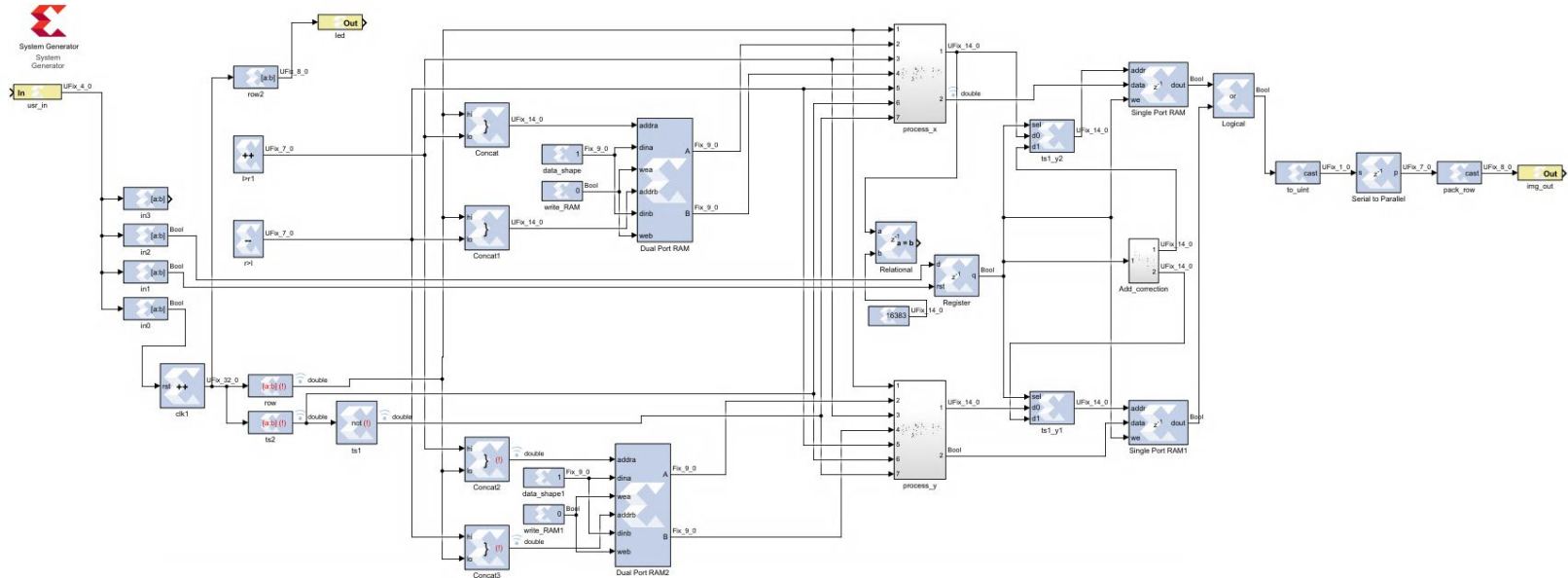
CPU implementation (contd)

- For the scale of images I am working with 128 X 128 second stage doesn't do much
 - This makes sense since that is intended to be a smoothing stage
 - Makes my life easy **don't need to run it for two passes**

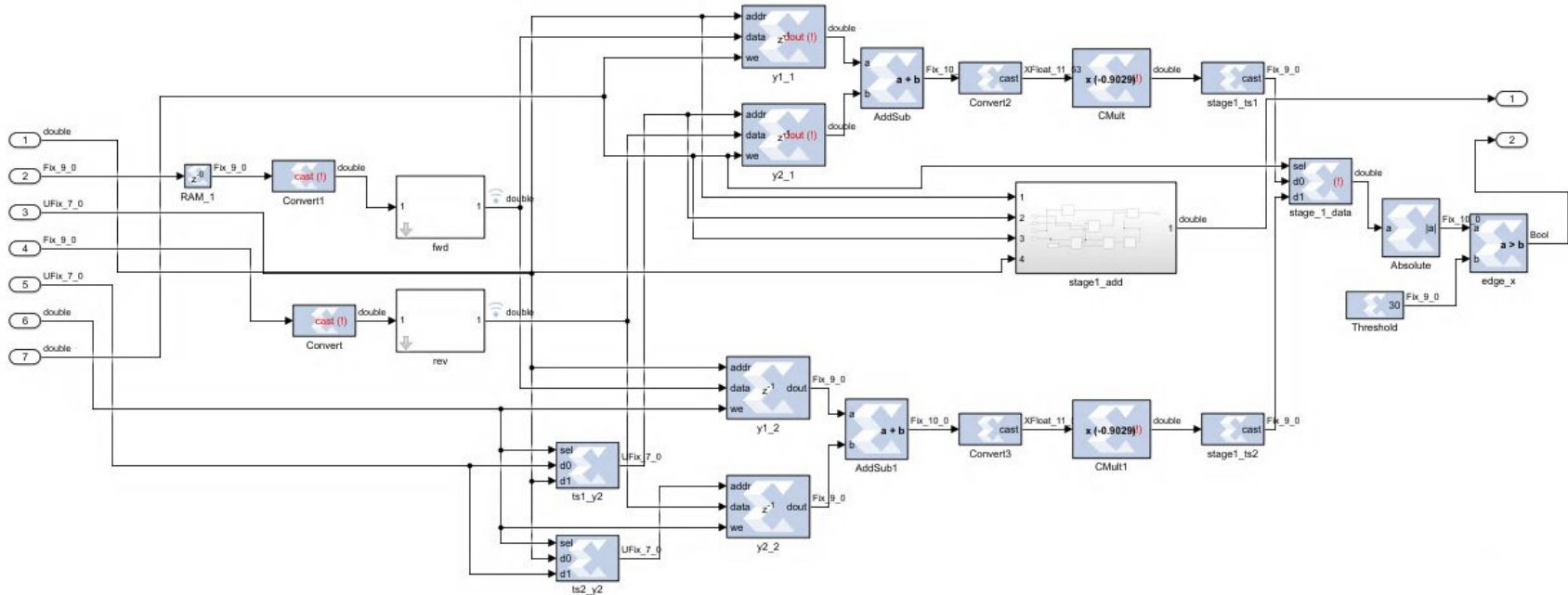
FPGA implementation

If you guys are interested I will do a run through my Xilinx program live.

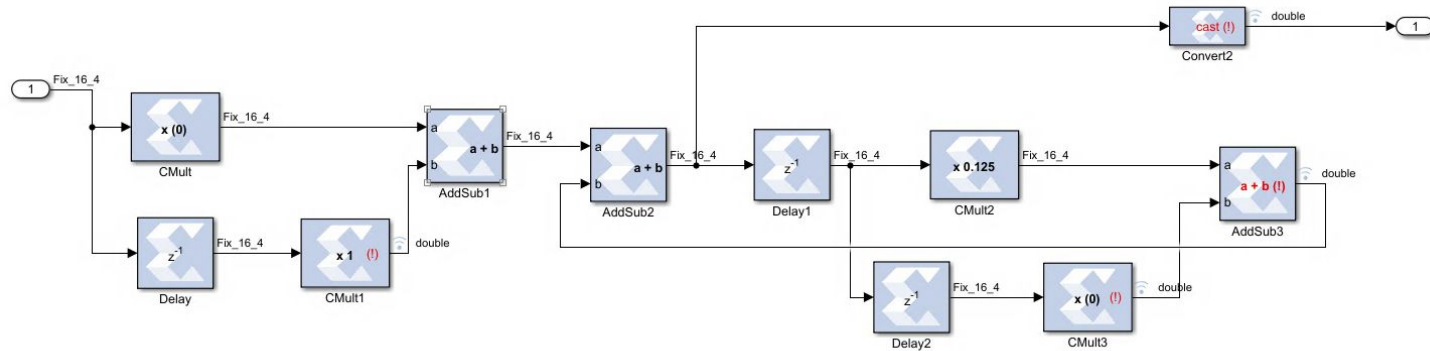
For completeness added a couple of screenshots



Internal blocks (contd)

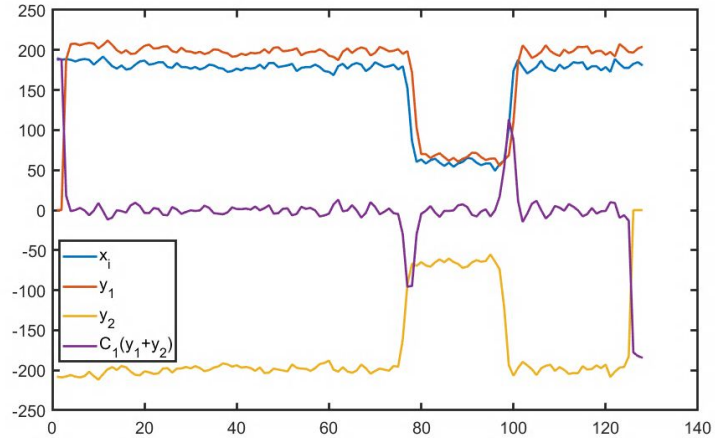


Internal blocks (contd)

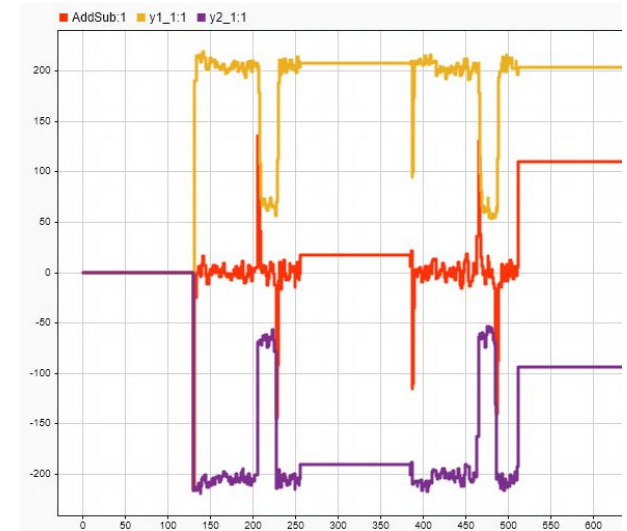


Simulation results

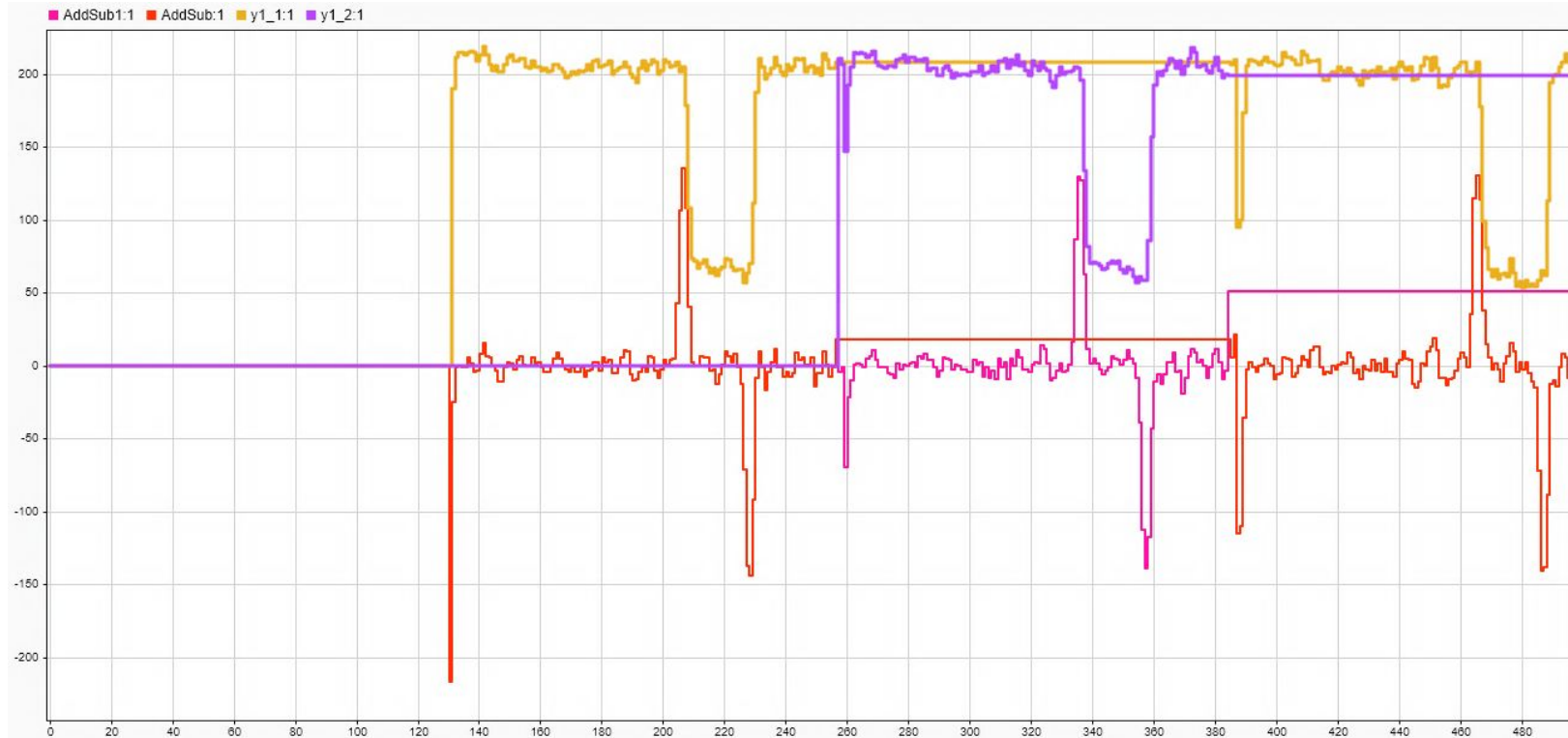
CPU - Version



Simulink



Pipeline - Use the time needed for reversal



Timing issues !

- Now that I had a confirmed working processing stage it's pretty simple right ?
 - Testing revealed that ethernet packets were not generated
 - Implementation shows a timing error
 - Spent a lot of time on figuring out what causes it
 - Cmulty take a lot of space when you use double
 - **Xilinx “Cmulty” block crashes Matlab on linux :(**

What worked !

- As I showed the basic signal processing stages are correctly calculated
- I have a python script that can get data from ethernet
- Sending data is also possible via python - Did not test it

Thank you !

Current design.

