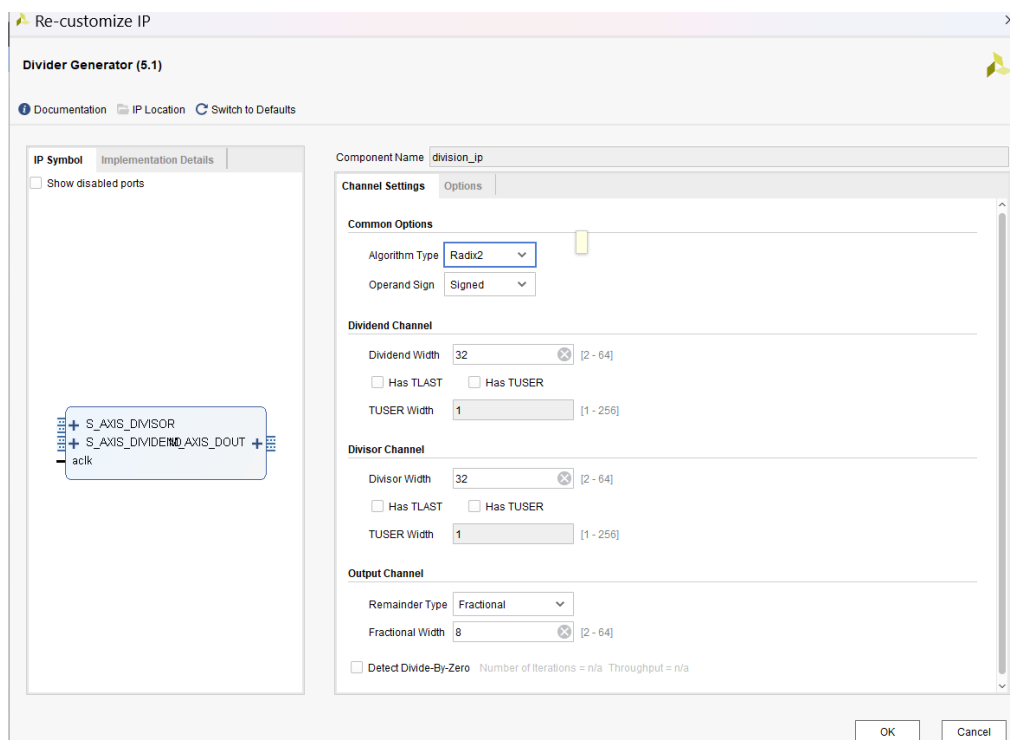


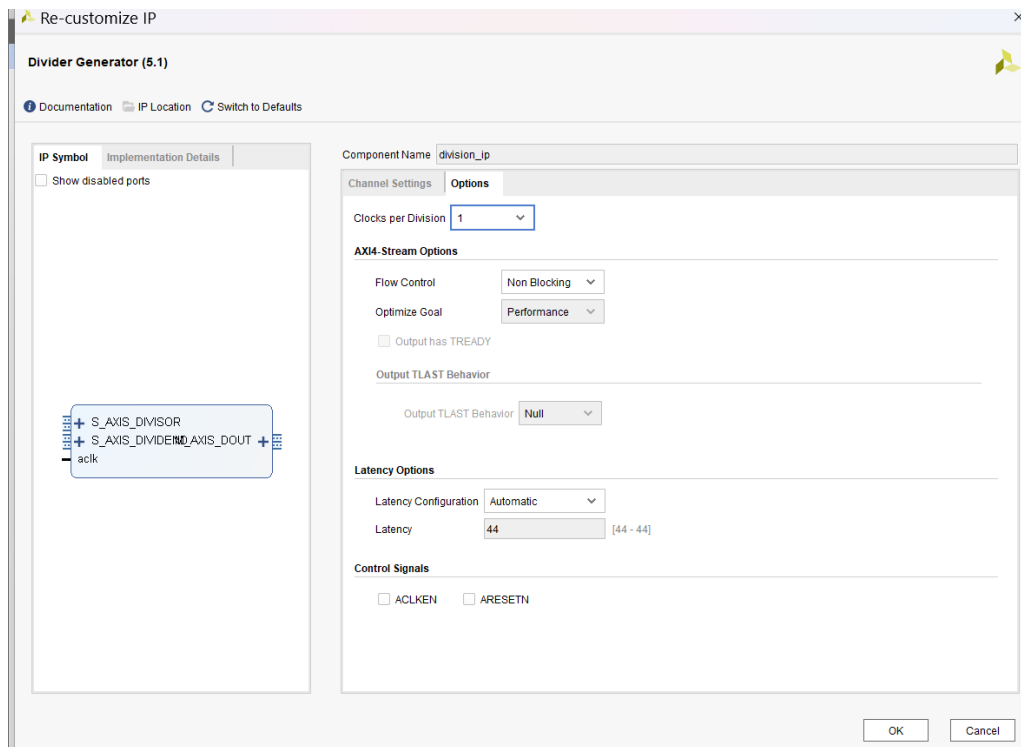
After copy pasting all the .v files for canny edge detection you will face some errors due to instantiation. To solve these follow the below steps:

Inside the Sobel module there is a DIVIDER GENERATOR instantiation named division\_ip.

For creating the instance follow the steps:

Go to IP Catalog in project manager >>> search for divider generator IP >>> make its settings as shown below

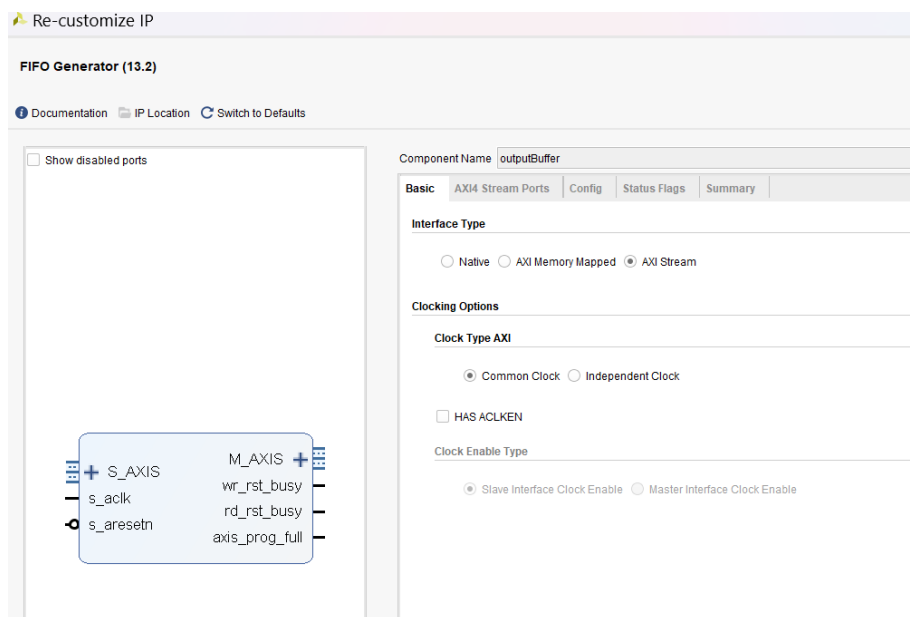




NB – change the component name as division\_ip

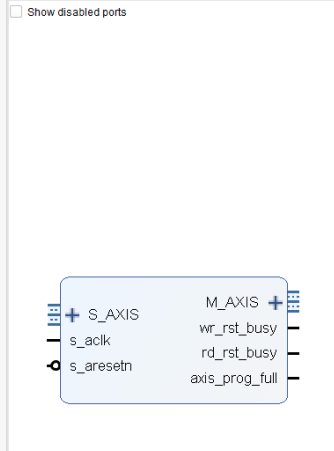
We also want to instantiate an output buffer, for these follow the steps:

Go to IP Catalog in project manager >>> search for fifo generator IP  
>>> make its settings as shown below



FIFO Generator (13.2)

Documentation IP Location Switch to Defaults

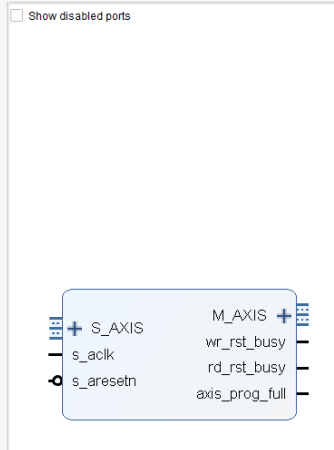


Component Name: outputBuffer

Basic	AXI4 Stream Ports	Config	Status Flags	Summary
TDATA NUM BYTES: 1 [0,1,2..512]				
TID WIDTH: 0 [0 - 32]				
TDEST WIDTH: 0 [0 - 32]				
TUSER WIDTH: 0 [0 - 4096]				
<input type="checkbox"/> HAS TSTRB TSTRB WIDTH: 1 [1 - 1]				
<input type="checkbox"/> HAS TKEEP TKEEP WIDTH: 1 [1 - 1]				
<input checked="" type="checkbox"/> TREADY				
<input type="checkbox"/> TLAST				
Calculated Width: 8				

FIFO Generator (13.2)

Documentation IP Location Switch to Defaults

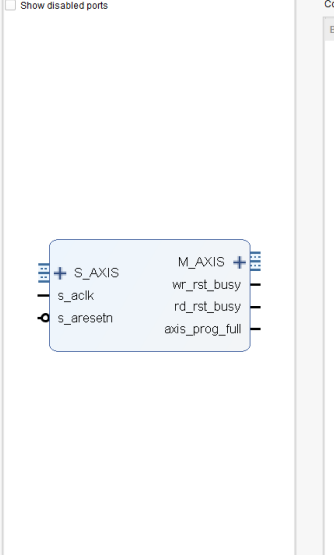


Component Name: outputBuffer

Basic	AXI4 Stream Ports	Config	Status Flags	Summary
TDATA NUM BYTES: 1 [0,1,2..512]				
TID WIDTH: 0 [0 - 32]				
TDEST WIDTH: 0 [0 - 32]				
TUSER WIDTH: 0 [0 - 4096]				
<input type="checkbox"/> HAS TSTRB TSTRB WIDTH: 1 [1 - 1]				
<input type="checkbox"/> HAS TKEEP TKEEP WIDTH: 1 [1 - 1]				
<input checked="" type="checkbox"/> TREADY				
<input type="checkbox"/> TLAST				
Calculated Width: 8				

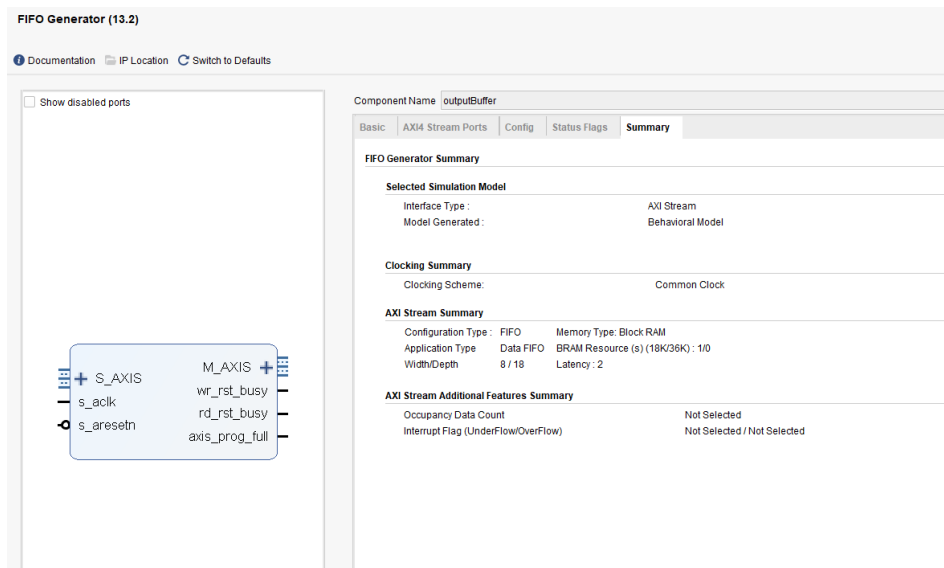
FIFO Generator (13.2)

Documentation IP Location Switch to Defaults



Component Name: outputBuffer

Basic	AXI4 Stream Ports	Config	Status Flags	Summary
<b>Configuration Options</b>				
<input checked="" type="radio"/> FIFO <input type="radio"/> Register Slice				
<b>FIFO Options</b>				
FIFO Implementation Type: Common Clock Block RAM				
FIFO Application Type				
<input checked="" type="radio"/> Data FIFO <input type="radio"/> Packet FIFO <input type="radio"/> Low Latency Data FIFO				
Latency: 2				
FIFO Width: 8 FIFO Depth: 16 Actual FIFO Depth: 18				
<b>ECC And Output Register Options</b>				
<input type="checkbox"/> ECC <input type="checkbox"/> Single Bit Error Injection <input type="checkbox"/> Double Bit Error Injection				
<input type="checkbox"/> Embedded Registers				
<b>Data Threshold Parameters</b>				
Programmable Full Type: Single Programmable Full Threshold Constant				
Full Threshold Assert Value: 8 [5 - 15]				
Programmable Empty Type: No Programmable Empty Threshold				
<input type="checkbox"/> Provide FIFO Occupancy Data Counts				



After setting all these you can add the tb.v as testbench for simulation. The input is trucks.bmp and output can be find in simulation folder after simulation named as trucks\_canny.bmp.

After that you can package the IP and test it on hardware by following the below youtube tutorial:

[https://youtu.be/Zm3KzhahbUg?si=vD47kpG\\_jnosRNxy](https://youtu.be/Zm3KzhahbUg?si=vD47kpG_jnosRNxy)

Here instead of using sobel ip in block diagrams use the canny IP that you have packaged. You can use the same sdk code as in the youtube video for testing this on Zedboard FPGA.