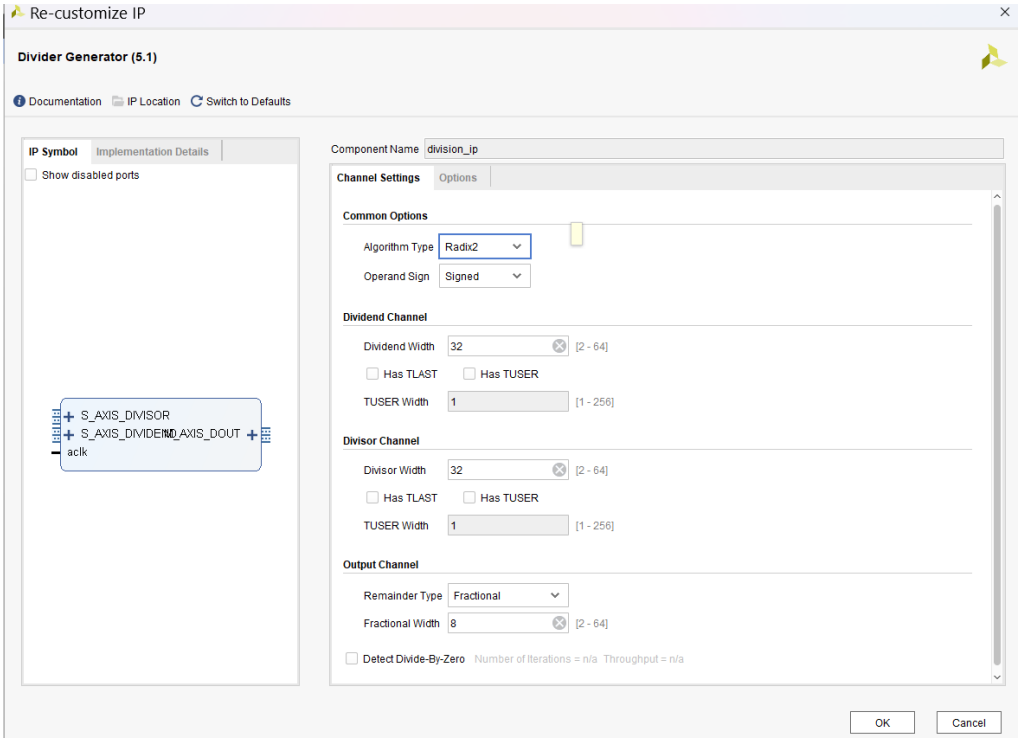
This code is available in github repo of Mr.Jeffrey Samuel. But actually his code have some errors. I have rectified this and implemented this on Zedboard on realtime.

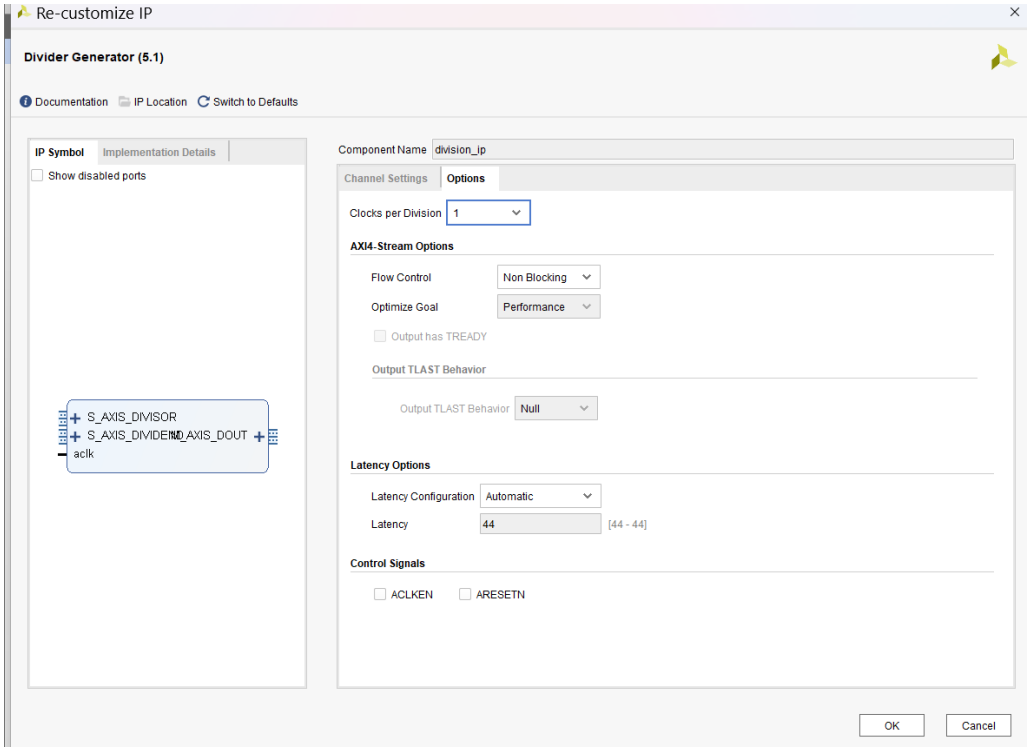
After copy pasting all the .v files for canny edge detection you will face some errors due to instantiation. To solve these follow the below steps:

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

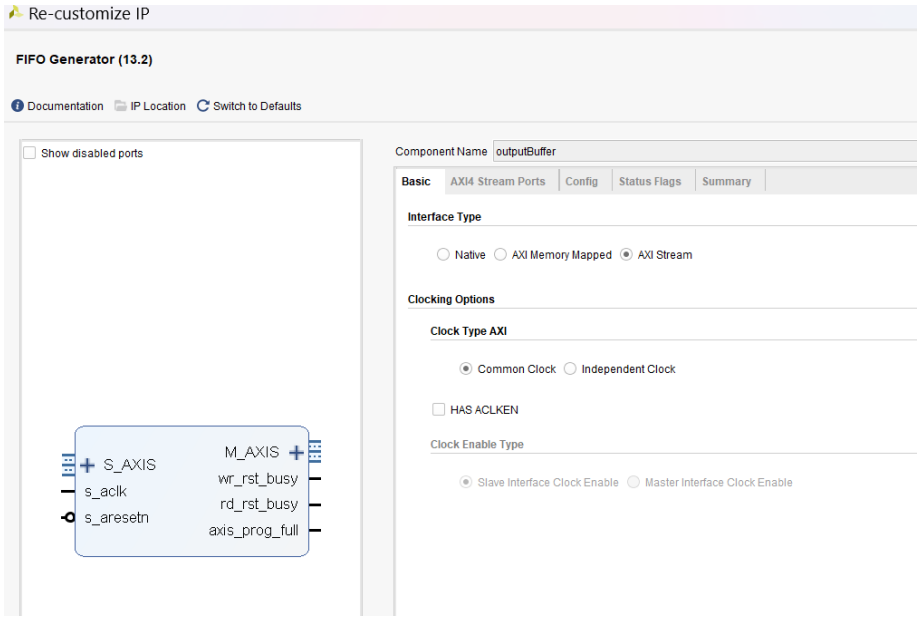


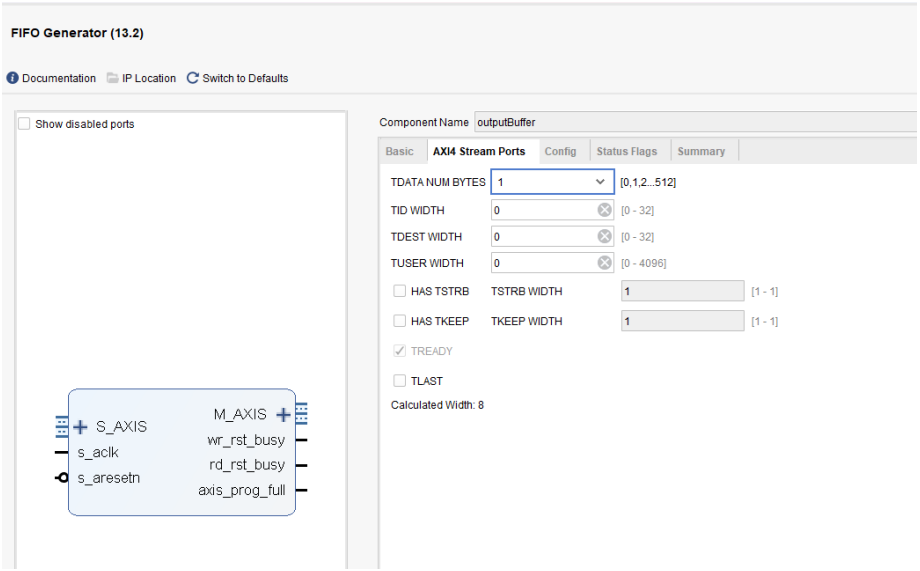


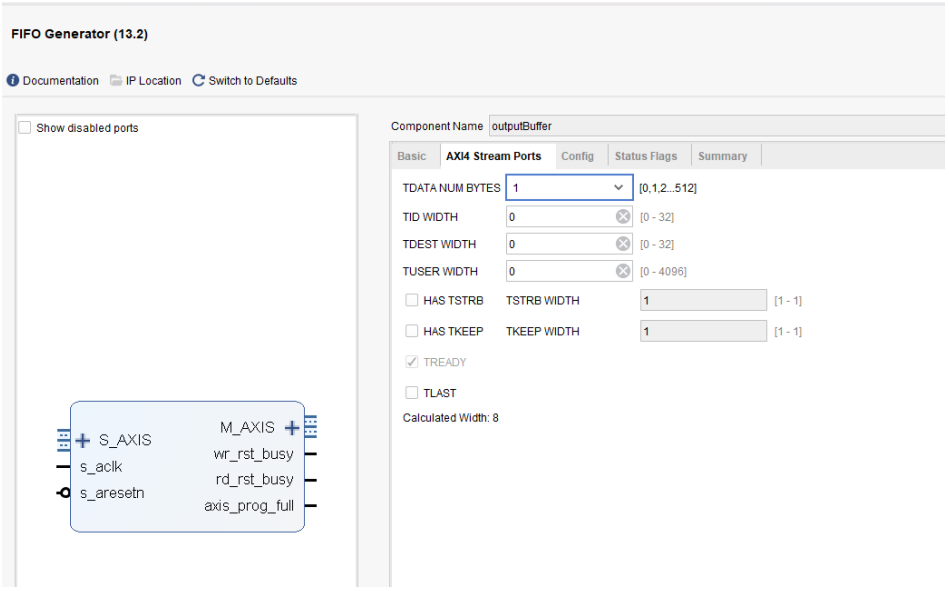
NB – change the component name as division_ip

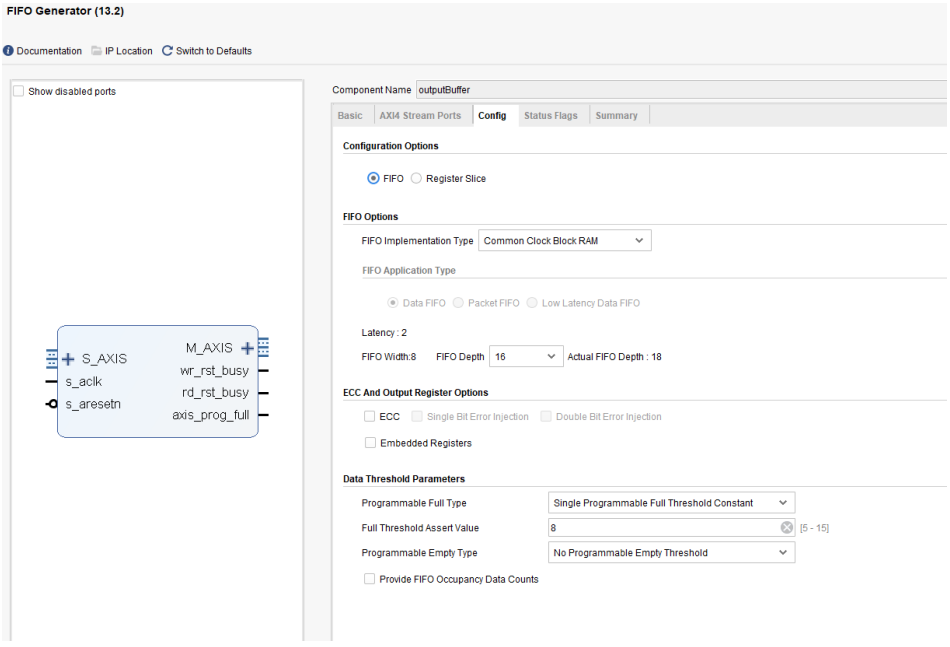
We also want to instantiate an output buffer, for these follow the steps:

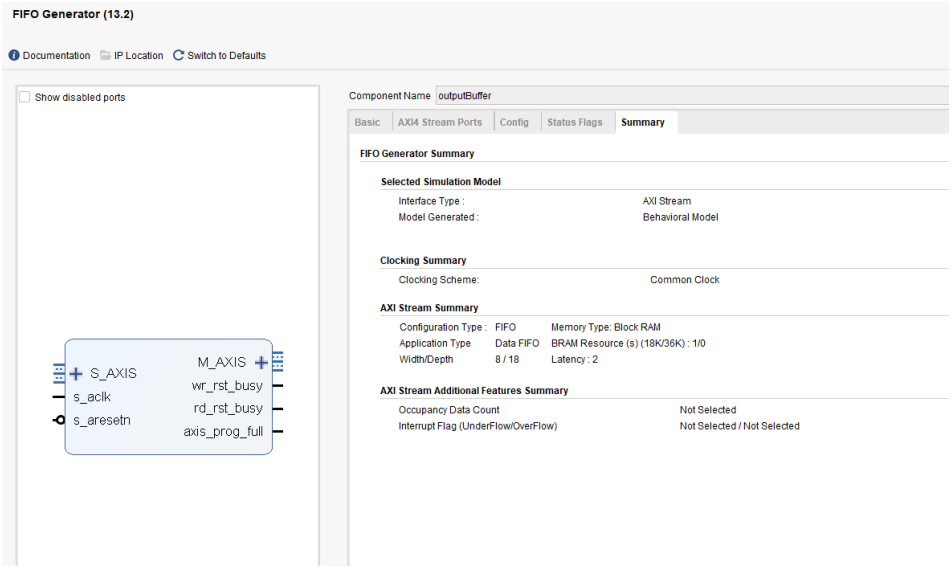
Go to IP Catalog in project manager >>> search for fifo generator IP >>> make its settings as shown below




FIFO Generator (13.2)
Documentation
IP Location
Switch to Defaults
Show disabled ports
S_AXIS
s_aclk
s_aresetn
M_AXIS
wr_rst_busy
rd_rst_busy
axis_prog_full
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]
HAS TSTRB
TSTRB WIDTH
1
[1 - 1]
HAS TKEEP
TKEEP WIDTH
1
[1 - 1]
TREADY
TLAST
Calculated Width: 8


FIFO Generator (13.2)
Documentation
IP Location
Switch to Defaults
Show disabled ports
S_AXIS
s_aclk
s_aresetn
M_AXIS
wr_rst_busy
rd_rst_busy
axis_prog_full
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]
HAS TSTRB
TSTRB WIDTH
1
[1 - 1]
HAS TKEEP
TKEEP WIDTH
1
[1 - 1]
TREADY
TLAST
Calculated Width: 8


FIFO Generator (13.2)
Documentation
IP Location
Switch to Defaults
Show disabled ports
S_AXIS
s_aclk
s_aresetn
M_AXIS
wr_rst_busy
rd_rst_busy
axis_prog_full
Component Name
outputBuffer
Basic
AXI4 Stream Ports
Config
Status Flags
Summary
Configuration Options
FIFO
Register Slice
FIFO Options
FIFO Implementation Type
Common Clock Block RAM
FIFO Application Type
Data FIFO
Packet FIFO
Low Latency Data FIFO
Latency : 2
FIFO Width:8
FIFO Depth
16
Actual FIFO Depth : 18
ECC And Output Register Options
ECC
Single Bit Error Injection
Double Bit Error Injection
Embedded Registers
Data Threshold Parameters
Programmable Full Type
Single Programmable Full Threshold Constant
Full Threshold Assert Value
8
[5 - 15]
Programmable Empty Type
No Programmable Empty Threshold
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

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.