Xilinx Vivado project setup

1. Open the development kit

- Download ADQ7WB firmware from https://transfereu.teledyne.com/link/s7R4U9ED7V3m1JAZbEE04N
- unzip the file
- Start Vivado
- In the Vivado menu select Tools/ Run TCL Script
- Select the file: devkit/implementation/scripts/devkit.tcl

2. Set up the project

- Go to the TCL console command field and type:

```
devkit_setup
```

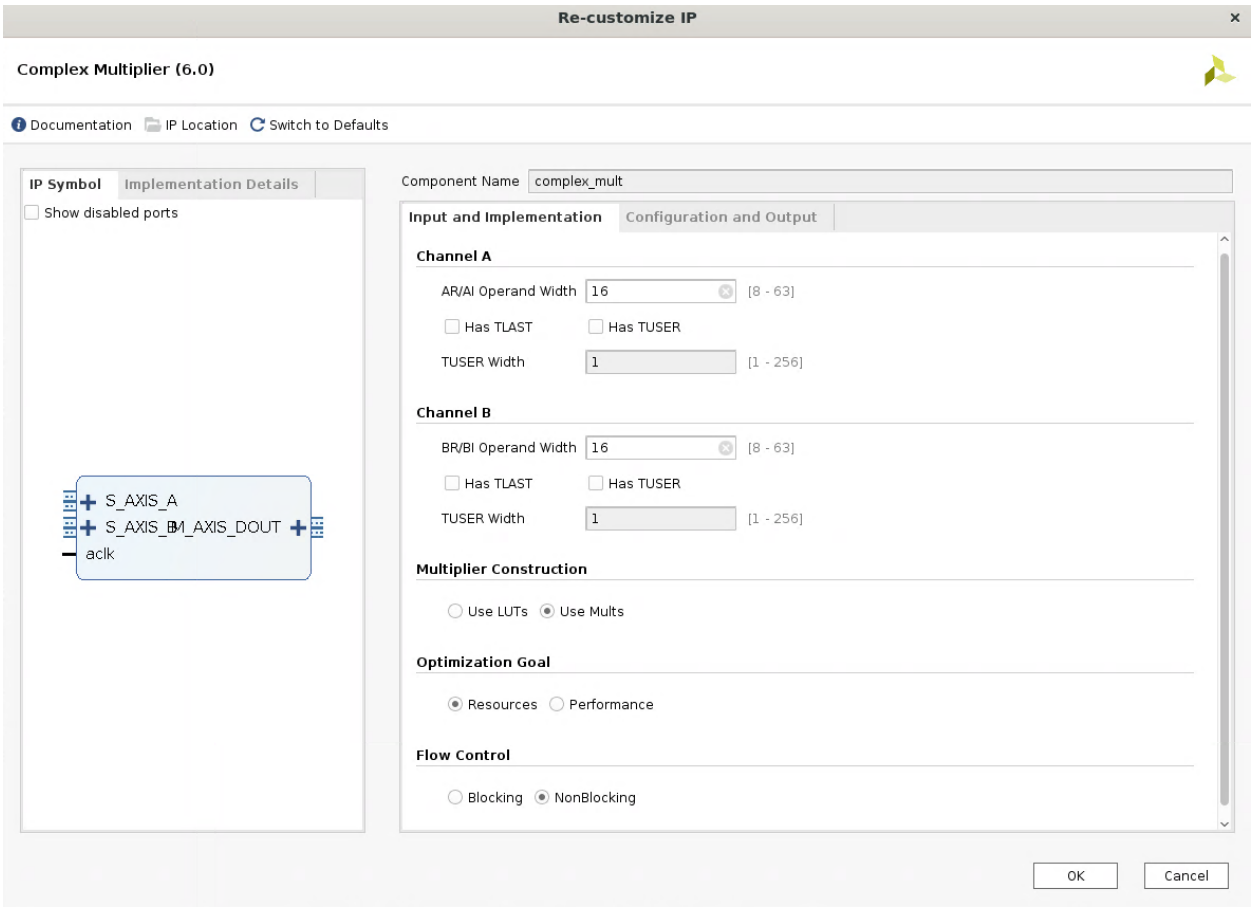
then press Return. The execution will take a while

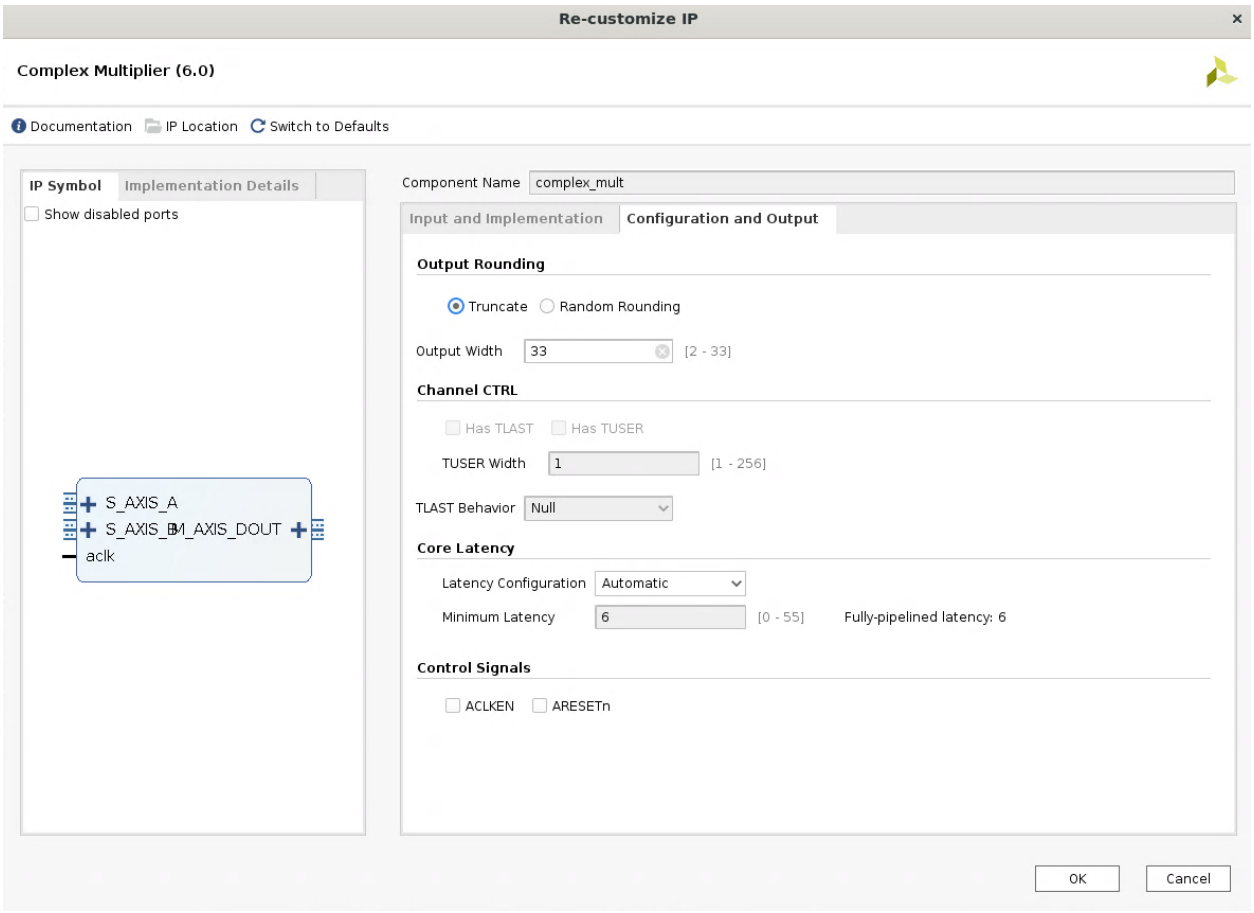
- In the Vivado menu select File/ Add source...
- In the Add Sources window, check 'Add or create design sources', then click 'Next'
- Click 'Add Files', select all the files in /frb-monitor/FPGA except for ul2_regfile.v, ul_coeff_mem.v, user_logic1.v and user_logic2.v. Then click 'Finish'

Create the IP cores as needed

complex_mult

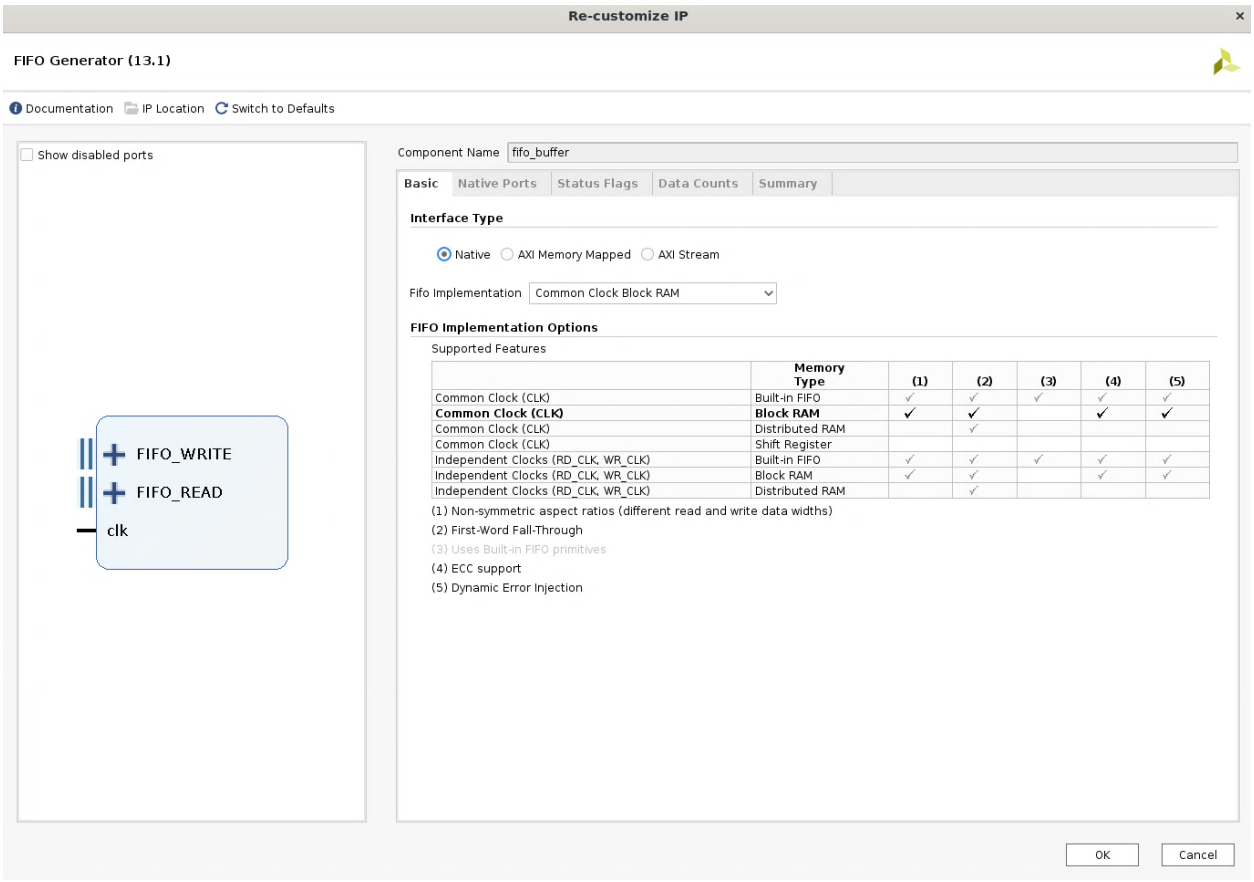
- In the 'Flow Navigator' window, select 'IP Catalog', search and open 'Complex Multiplier (6.0)'
- Config the IP as below:

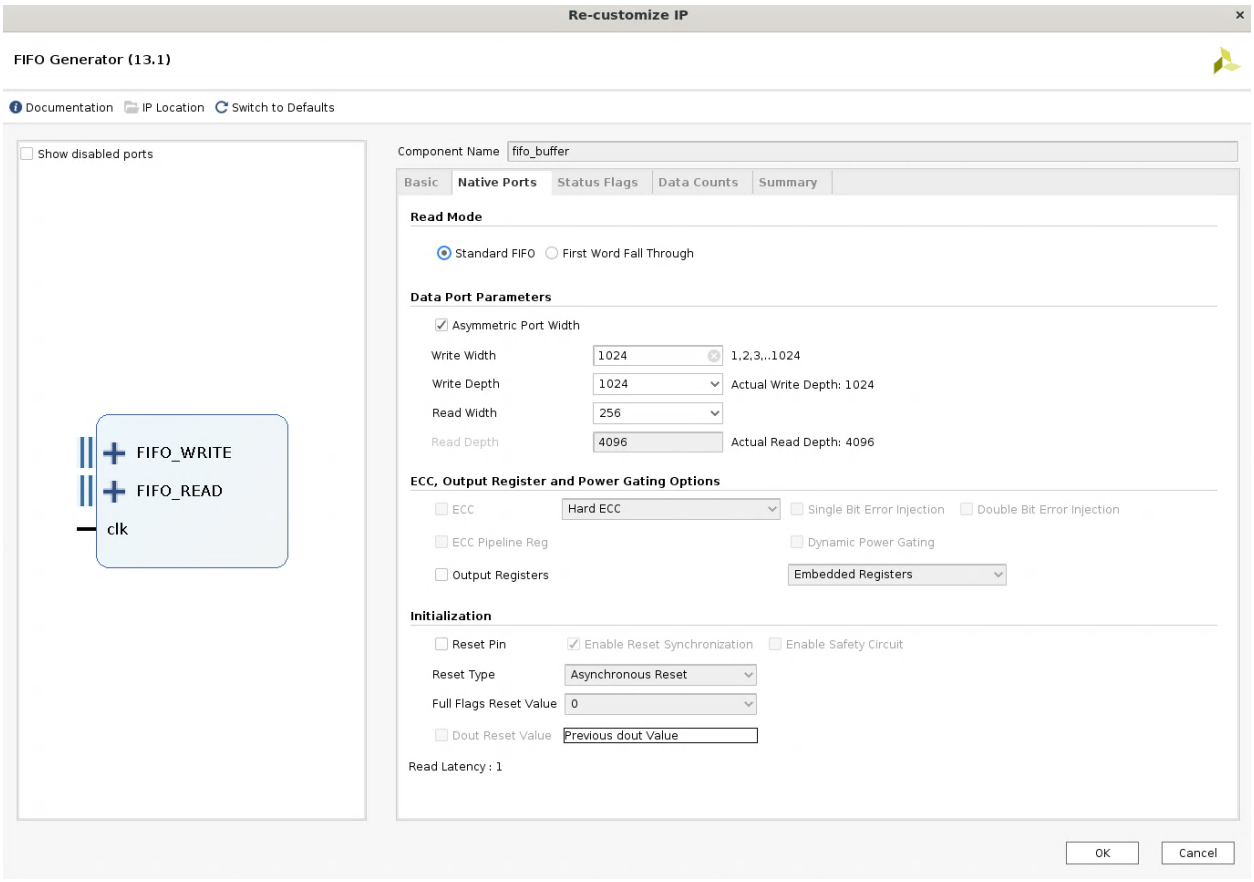

Re-customize IP
Complex Multiplier (6.0)
Documentation
IP Location
Switch to Defaults
IP Symbol
Implementation Details
Show disabled ports
S_AXIS_A
S_AXIS_B
M_AXIS_DOUT
aclk
Component Name
complex_mult
Input and Implementation
Configuration and Output
Channel A
AR/AI Operand Width
16
[8 - 63]
Has TLAST
Has TUSER
TUSER Width
1
[1 - 256]
Channel B
BR/BI Operand Width
16
[8 - 63]
Has TLAST
Has TUSER
TUSER Width
1
[1 - 256]
Multiplier Construction
Use LUTs
Use Mults
Optimization Goal
Resources
Performance
Flow Control
Blocking
NonBlocking
OK
Cancel



fifo_buffer

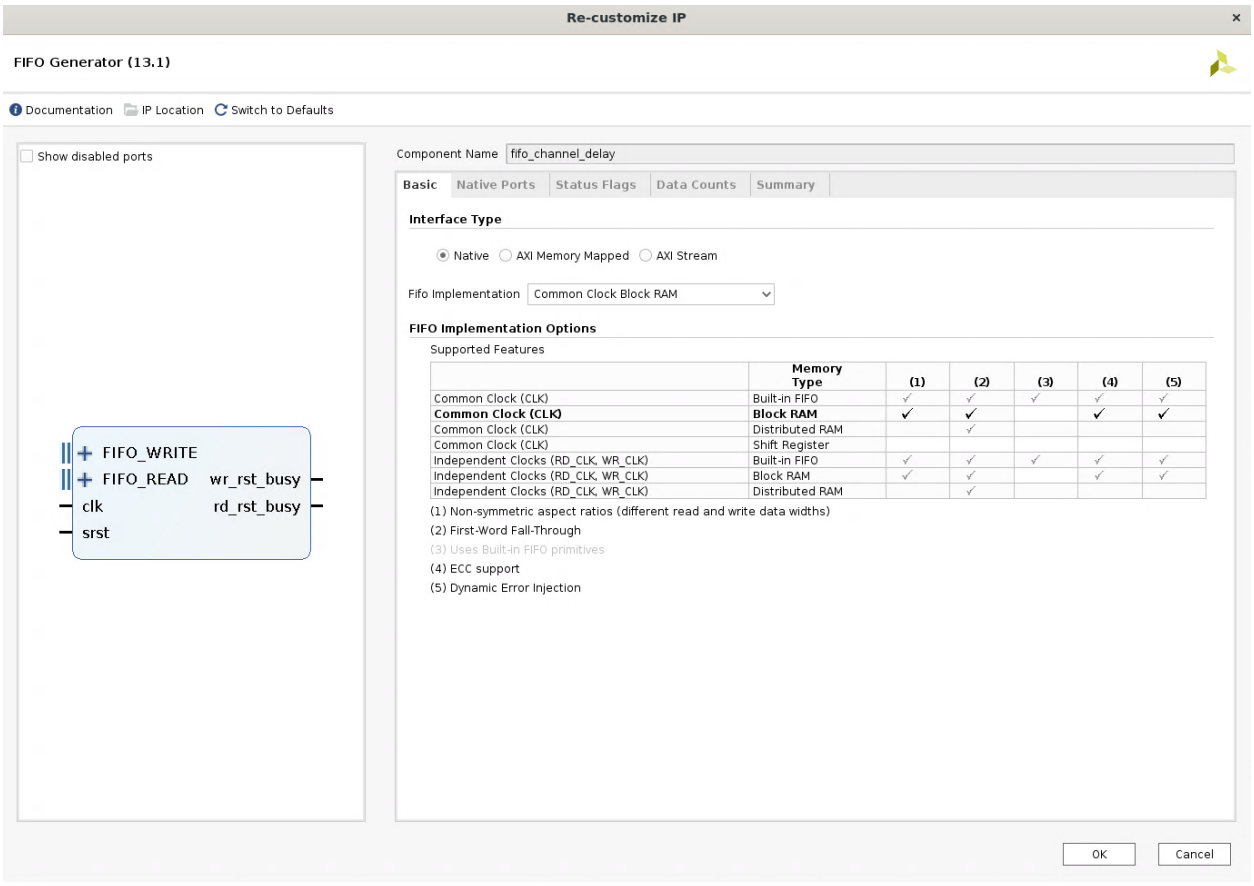
- In the 'Flow Navigator' window, select 'IP Catalog', search and open 'FIFO Generator (13.1)'
- Config the IP as below:

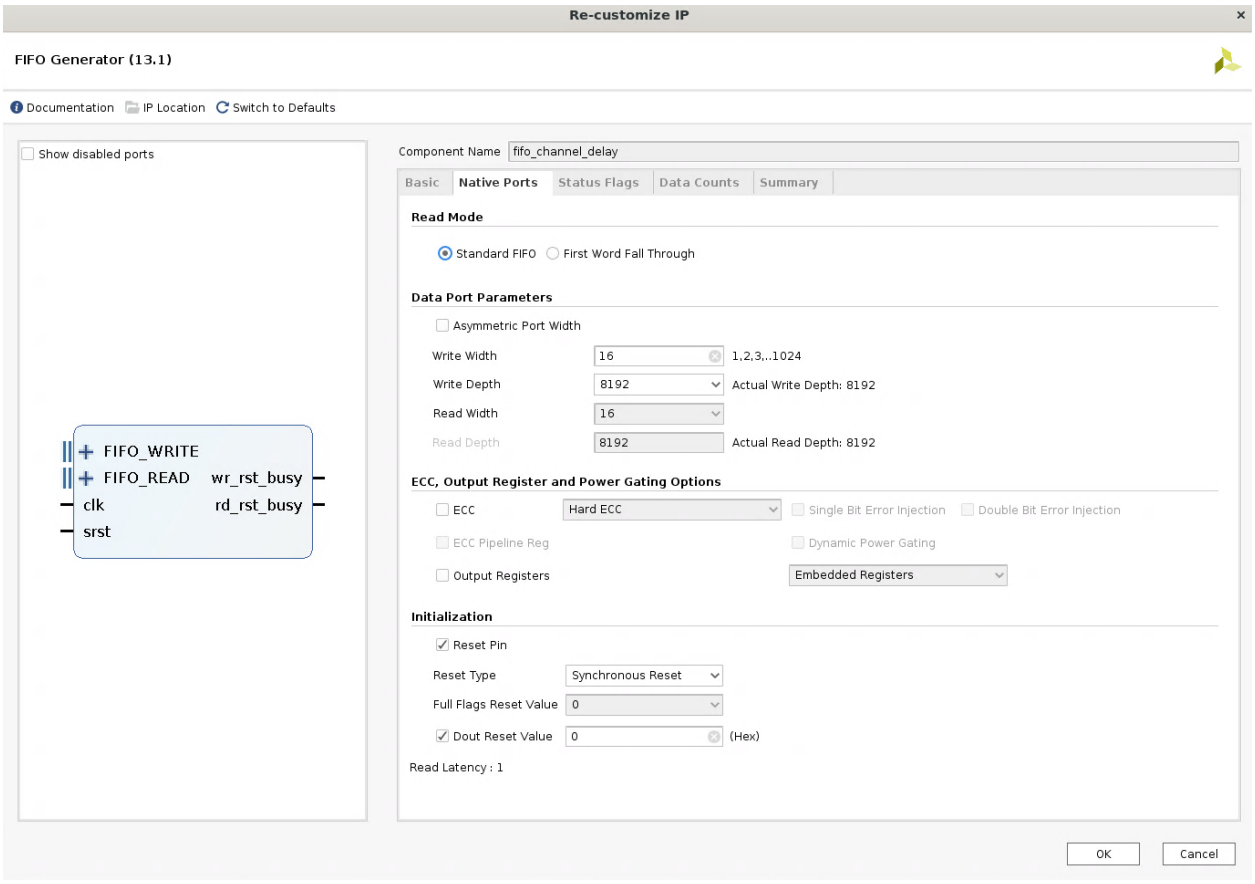

Re-customize IP
FIFO Generator (13.1)
Documentation
IP Location
Switch to Defaults
Show disabled ports
FIFO_WRITE
FIFO_READ
clk
Component Name
fifo_buffer
Basic
Native Ports
Status Flags
Data Counts
Summary
Interface Type
Native
AXI Memory Mapped
AXI Stream
Fifo Implementation
Common Clock Block RAM
FIFO Implementation Options
Supported Features
Memory Type
(1)
(2)
(3)
(4)
(5)
Common Clock (CLK) Built-in FIFO
Common Clock (CLK) Block RAM
Common Clock (CLK) Distributed RAM
Common Clock (CLK) Shift Register
Independent Clocks (RD_CLK, WR_CLK) Built-in FIFO
Independent Clocks (RD_CLK, WR_CLK) Block RAM
Independent Clocks (RD_CLK, WR_CLK) Distributed RAM
(1) Non-symmetric aspect ratios (different read and write data widths)
(2) First-Word Fall-Through
(3) Uses Built-in FIFO primitives
(4) ECC support
(5) Dynamic Error Injection
OK
Cancel



fifo_channel_delay

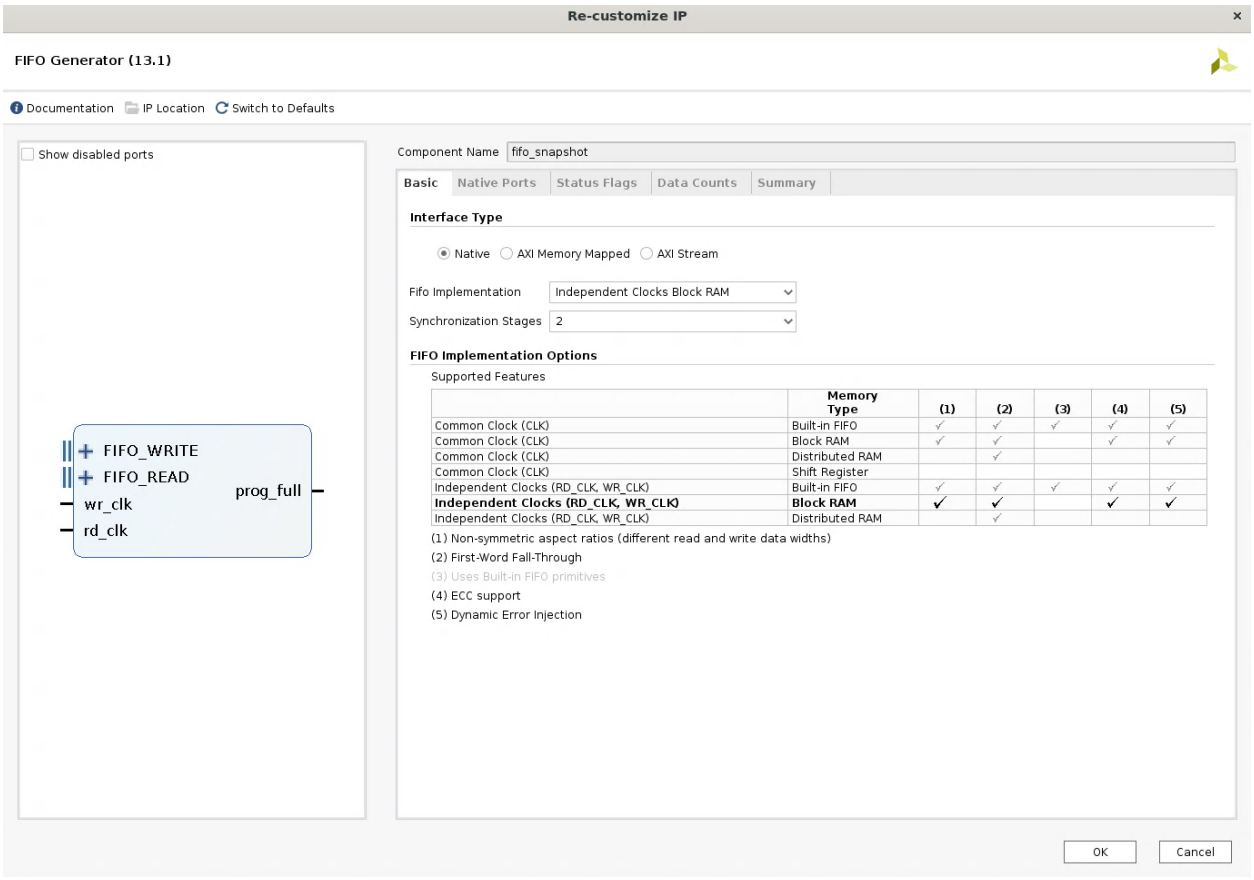
- In the 'Flow Navigator' window, select 'IP Catalog', search and open 'FIFO Generator (13.1)'
- Config the IP as below:

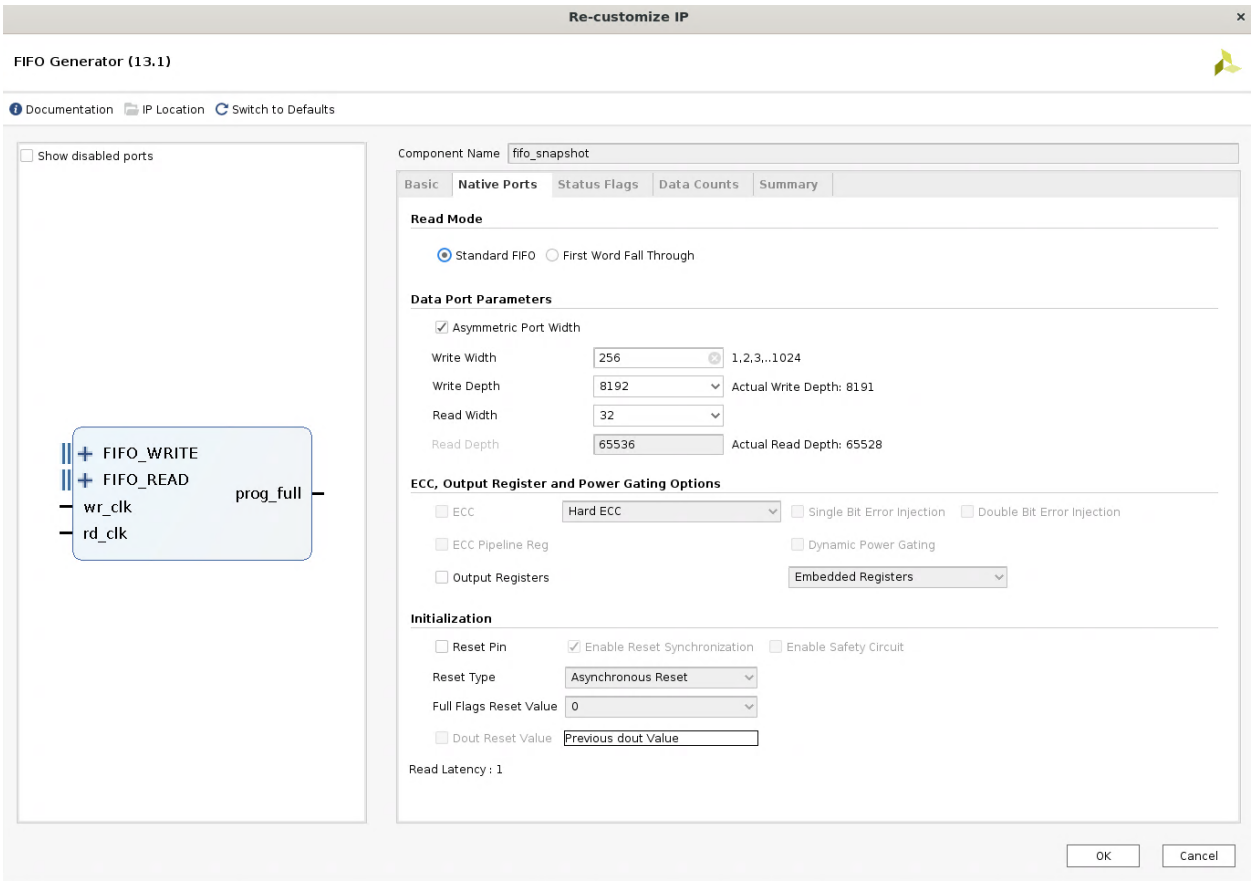

Re-customize IP
FIFO Generator (13.1)
Documentation
IP Location
Switch to Defaults
Show disabled ports
FIFO_WRITE
FIFO_READ
clk
srst
wr_rst_busy
rd_rst_busy
Component Name
fifo_channel_delay
Basic
Native Ports
Status Flags
Data Counts
Summary
Interface Type
Native
AXI Memory Mapped
AXI Stream
Fifo Implementation
Common Clock Block RAM
FIFO Implementation Options
Supported Features
Memory Type
(1)
(2)
(3)
(4)
(5)
Common Clock (CLK) Built-in FIFO
Common Clock (CLK) Block RAM
Common Clock (CLK) Distributed RAM
Common Clock (CLK) Shift Register
Independent Clocks (RD_CLK, WR_CLK) Built-in FIFO
Independent Clocks (RD_CLK, WR_CLK) Block RAM
Independent Clocks (RD_CLK, WR_CLK) Distributed RAM
(1) Non-symmetric aspect ratios (different read and write data widths)
(2) First-Word Fall-Through
(3) Uses Built-in FIFO primitives
(4) ECC support
(5) Dynamic Error Injection
OK
Cancel

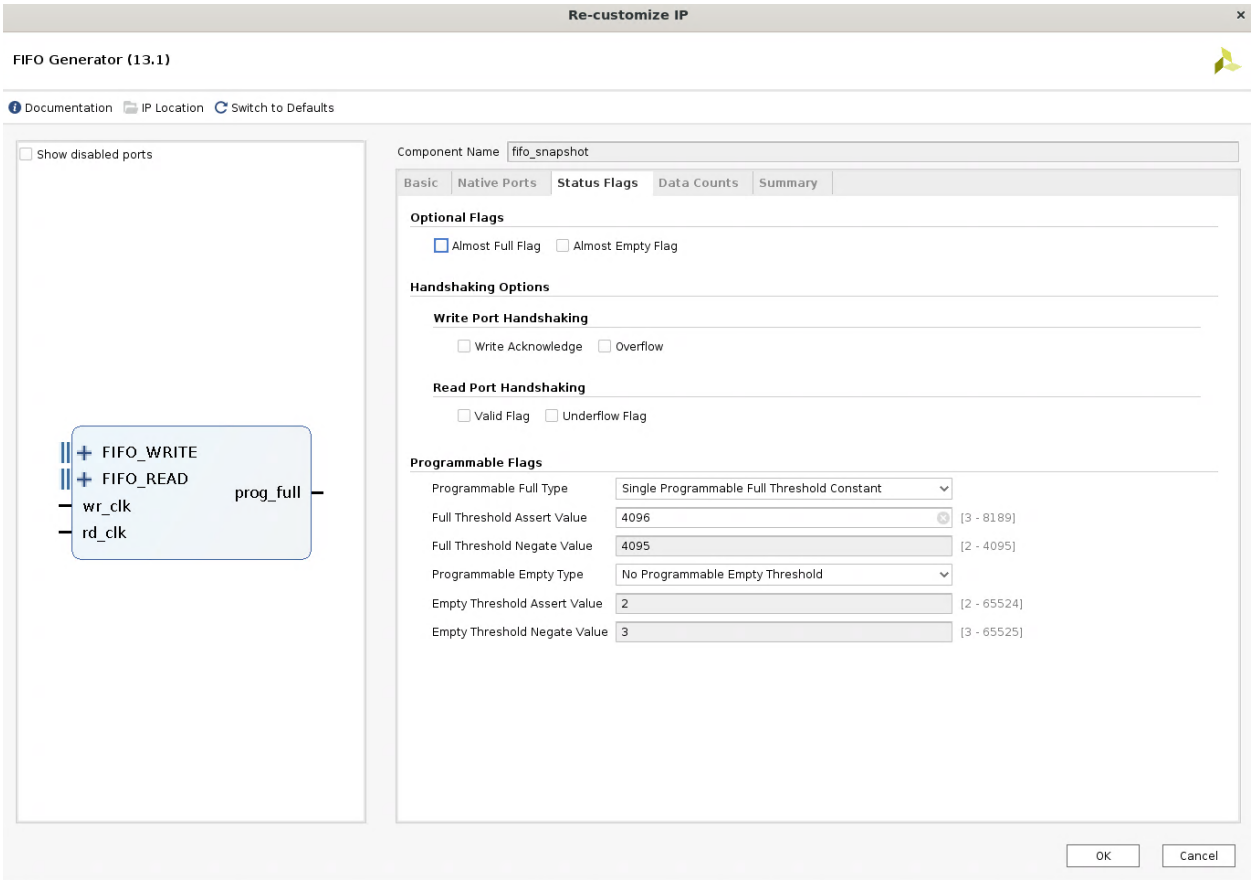


fifo_snapshot

- In the 'Flow Navigator' window, select 'IP Catalog', search and open 'FIFO Generator (13.1)'
- Config the IP as below:

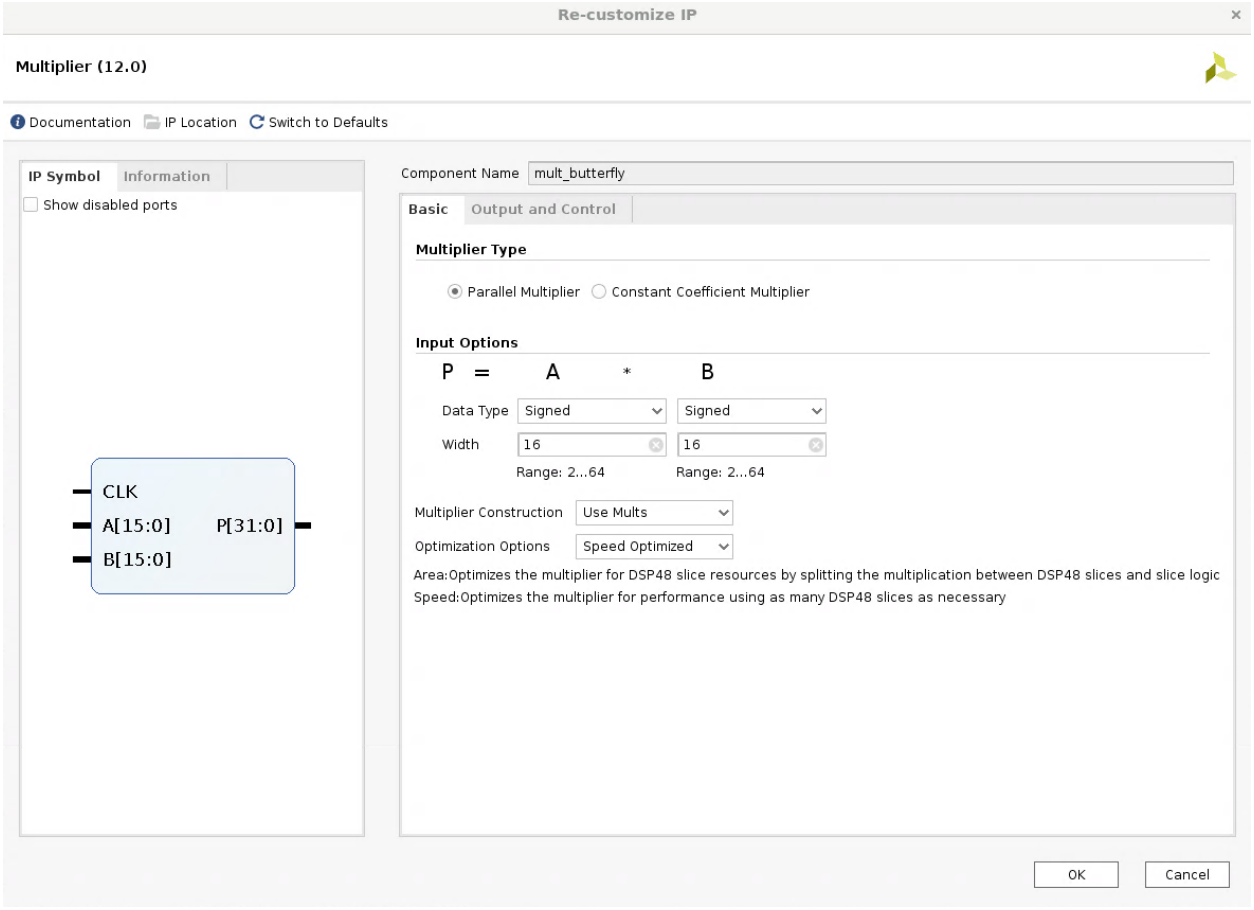

Re-customize IP
FIFO Generator (13.1)
Documentation
IP Location
Switch to Defaults
Show disabled ports
FIFO_WRITE
FIFO_READ
wr_clk
rd_clk
prog_full
Component Name
fifo_snapshot
Basic
Native Ports
Status Flags
Data Counts
Summary
Interface Type
Native
AXI Memory Mapped
AXI Stream
Fifo Implementation
Independent Clocks Block RAM
Synchronization Stages
2
FIFO Implementation Options
Supported Features
Memory Type
(1)
(2)
(3)
(4)
(5)
Common Clock (CLK) Built-in FIFO
Common Clock (CLK) Block RAM
Common Clock (CLK) Distributed RAM
Common Clock (CLK) Shift Register
Independent Clocks (RD_CLK, WR_CLK) Built-in FIFO
Independent Clocks (RD_CLK, WR_CLK) Block RAM
Independent Clocks (RD_CLK, WR_CLK) Distributed RAM
(1) Non-symmetric aspect ratios (different read and write data widths)
(2) First-Word Fall-Through
(3) Uses Built-in FIFO primitives
(4) ECC support
(5) Dynamic Error Injection
OK
Cancel

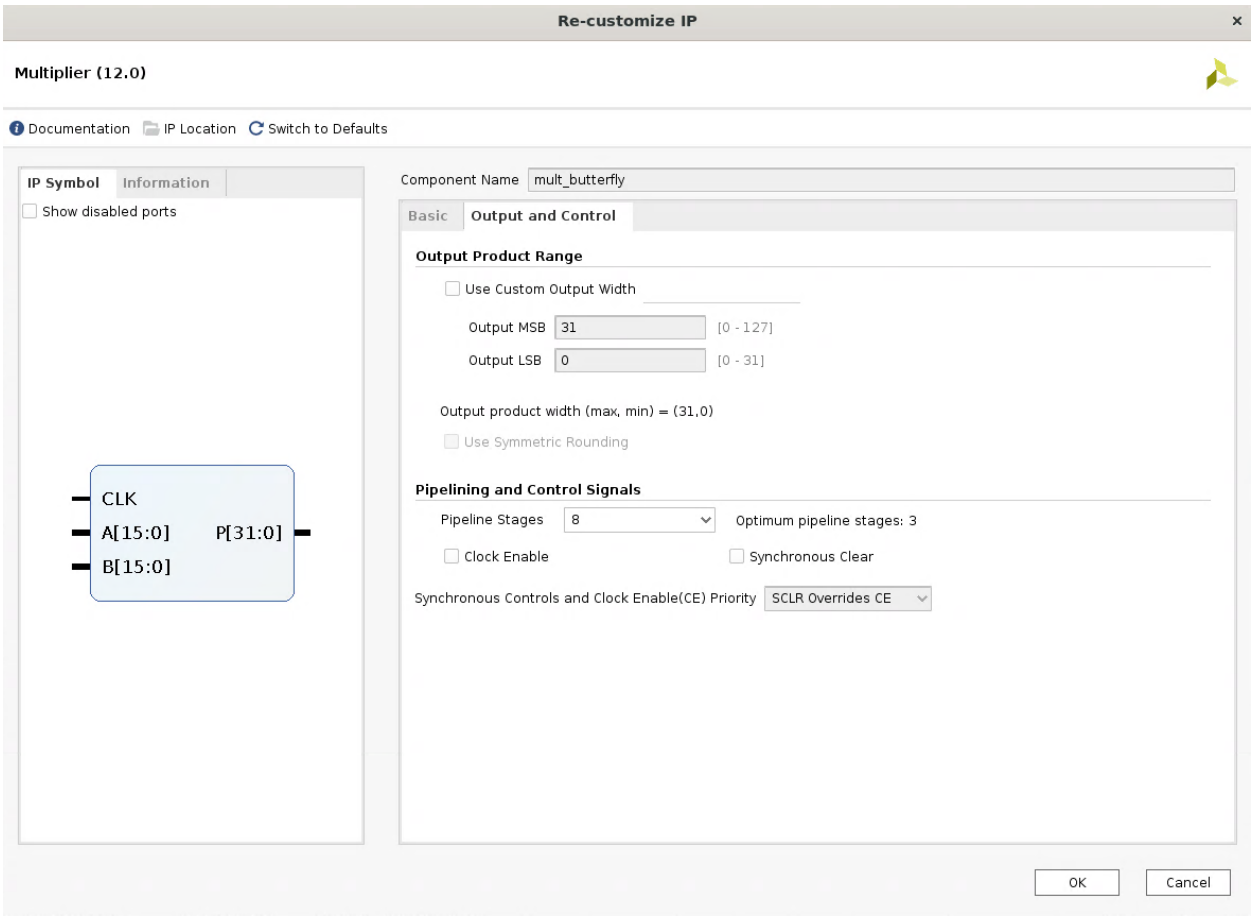

Re-customize IP
FIFO Generator (13.1)
Documentation
IP Location
Switch to Defaults
Show disabled ports
FIFO_WRITE
FIFO_READ
wr_clk
rd_clk
prog_full
Component Name
fifo_snapshot
Basic
Native Ports
Status Flags
Data Counts
Summary
Read Mode
Standard FIFO
First Word Fall Through
Data Port Parameters
Asymmetric Port Width
Write Width
256
1,2,3,..1024
Write Depth
8192
Actual Write Depth: 8191
Read Width
32
Read Depth
65536
Actual Read Depth: 65528
ECC, Output Register and Power Gating Options
ECC
Hard ECC
Single Bit Error Injection
Double Bit Error Injection
ECC Pipeline Reg
Dynamic Power Gating
Output Registers
Embedded Registers
Initialization
Reset Pin
Enable Reset Synchronization
Enable Safety Circuit
Reset Type
Asynchronous Reset
Full Flags Reset Value
0
Dout Reset Value
Previous dout Value
Read Latency : 1
OK
Cancel



mult_butterfly

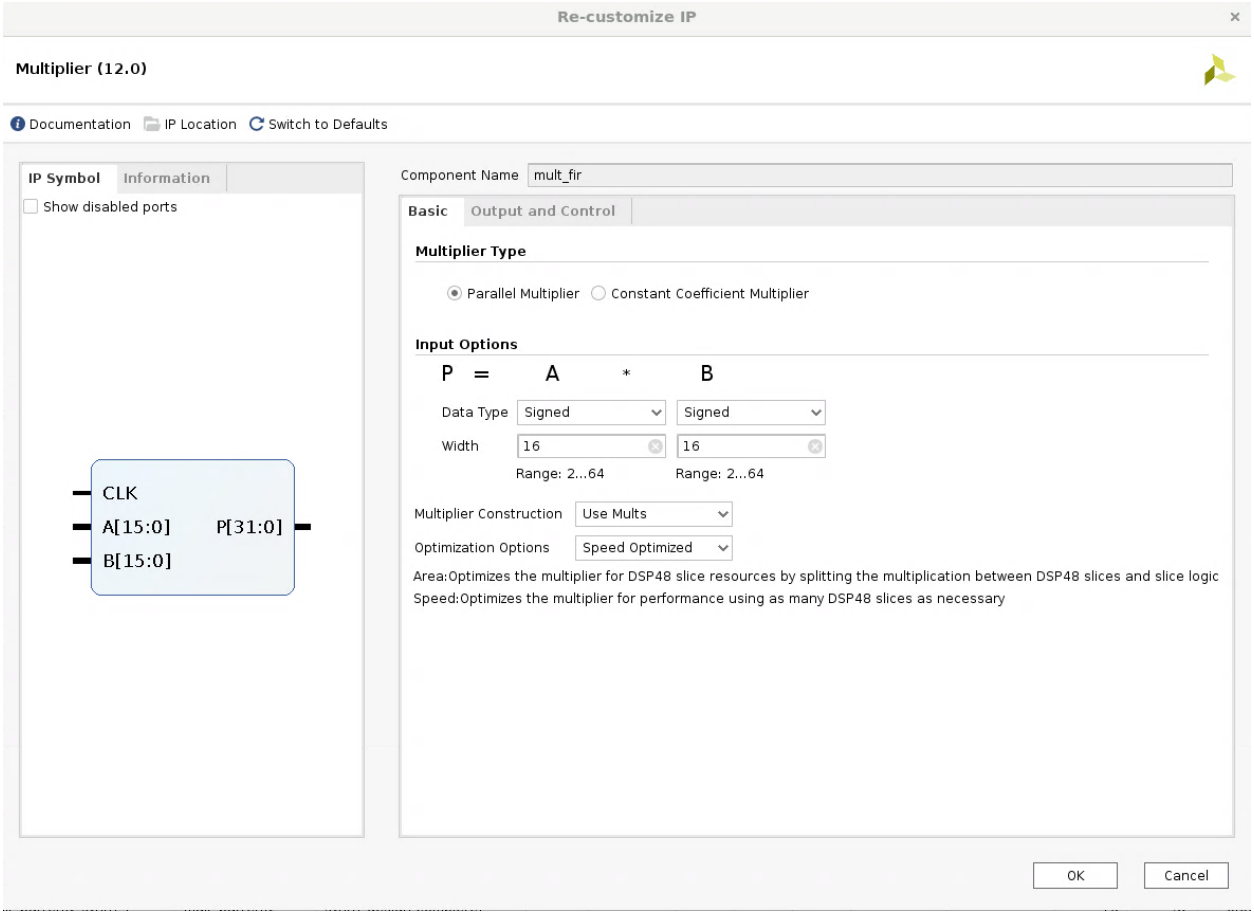
- In the 'Flow Navigator' window, select 'IP Catalog', search and open 'Multiplier (12.0)'
- Config the IP as below:

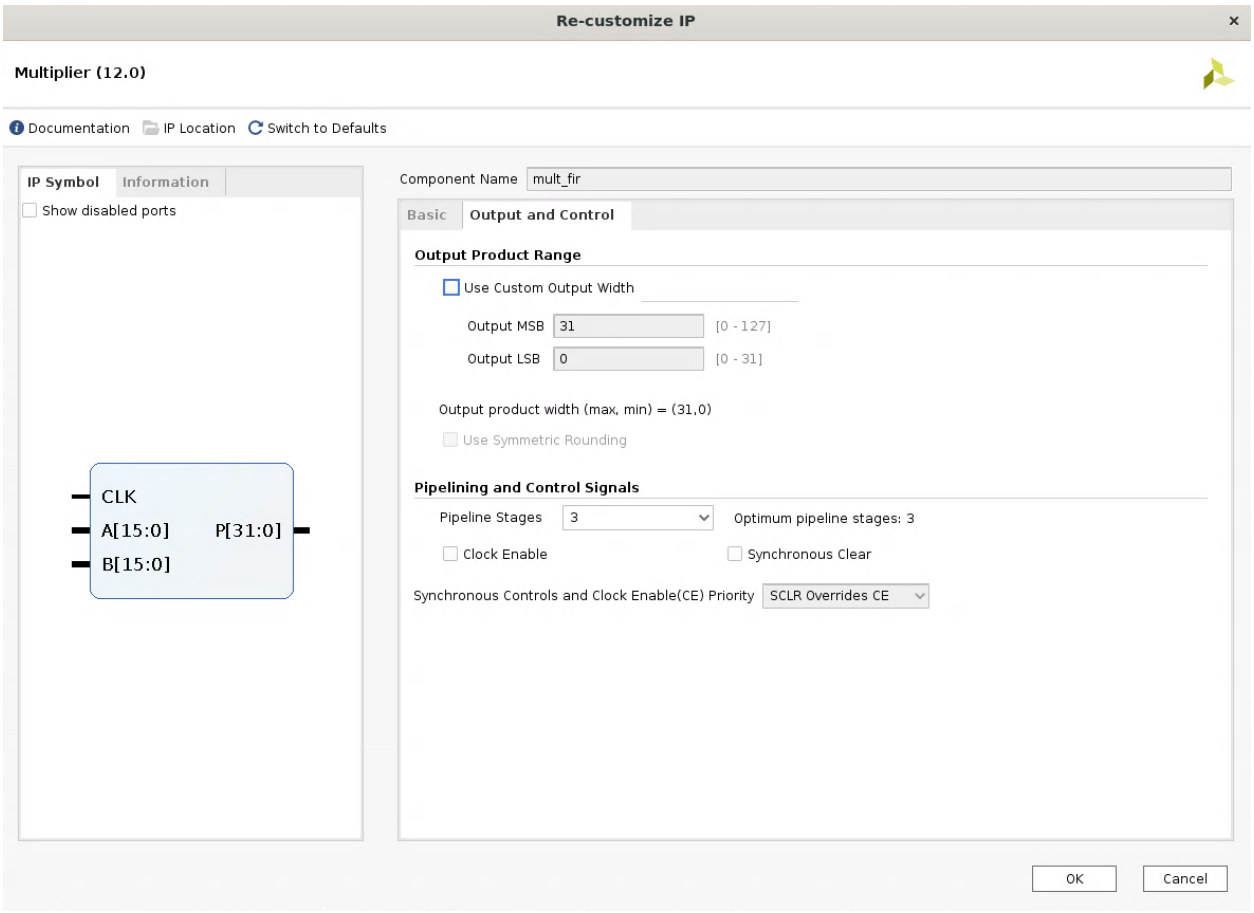

Re-customize IP
Multiplier (12.0)
Documentation
IP Location
Switch to Defaults
IP Symbol
Information
Show disabled ports
CLK
A[15:0]
B[15:0]
P[31:0]
Component Name
mult_butterfly
Basic
Output and Control
Multiplier Type
Parallel Multiplier
Constant Coefficient Multiplier
Input Options
P = A * B
Data Type
Signed
Signed
Width
16
16
Range: 2...64
Range: 2...64
Multiplier Construction
Use Mults
Optimization Options
Speed Optimized
Area:Optimizes the multiplier for DSP48 slice resources by splitting the multiplication between DSP48 slices and slice logic
Speed:Optimizes the multiplier for performance using as many DSP48 slices as necessary
OK
Cancel



mult_fir

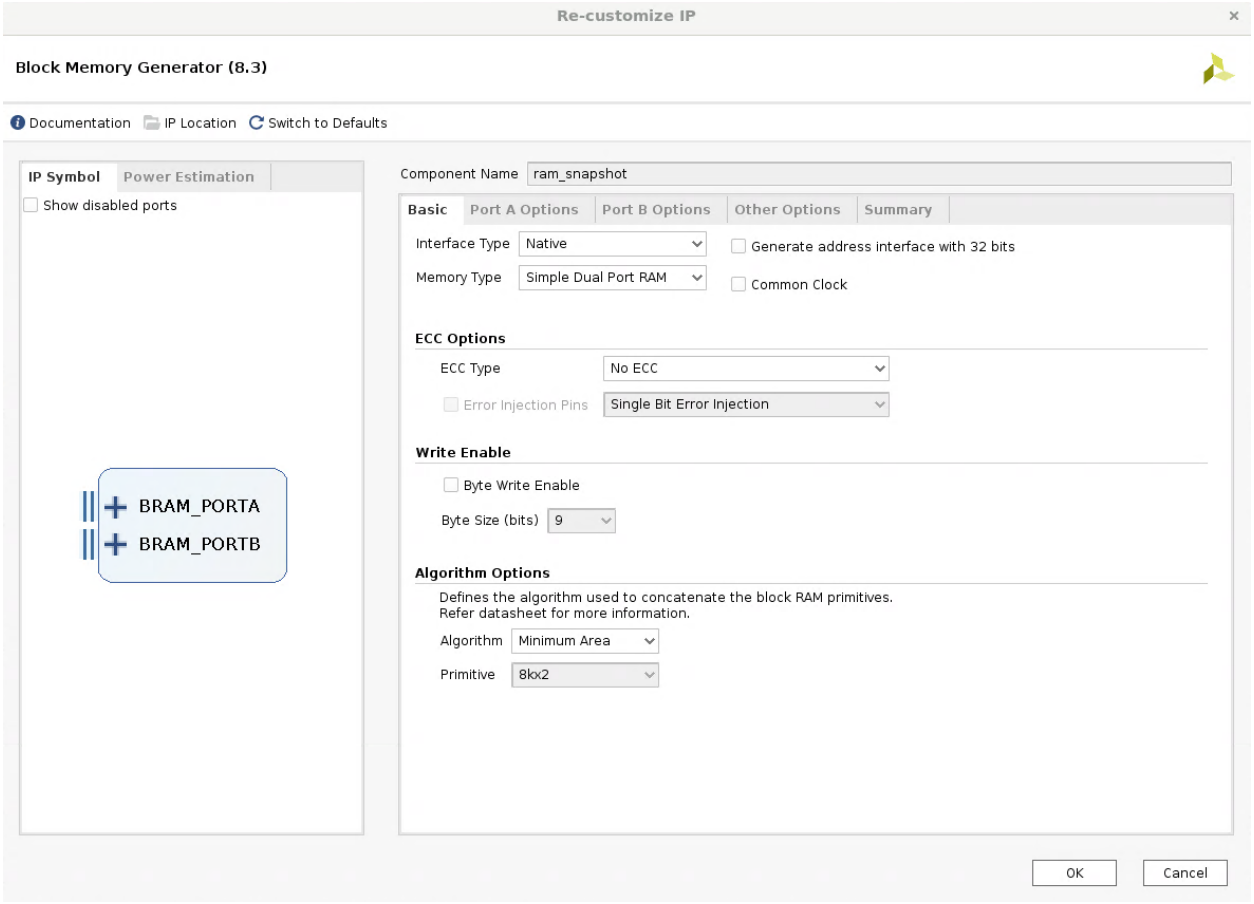
- In the 'Flow Navigator' window, select 'IP Catalog', search and open 'Multiplier (12.0)'
- Config the IP as below:

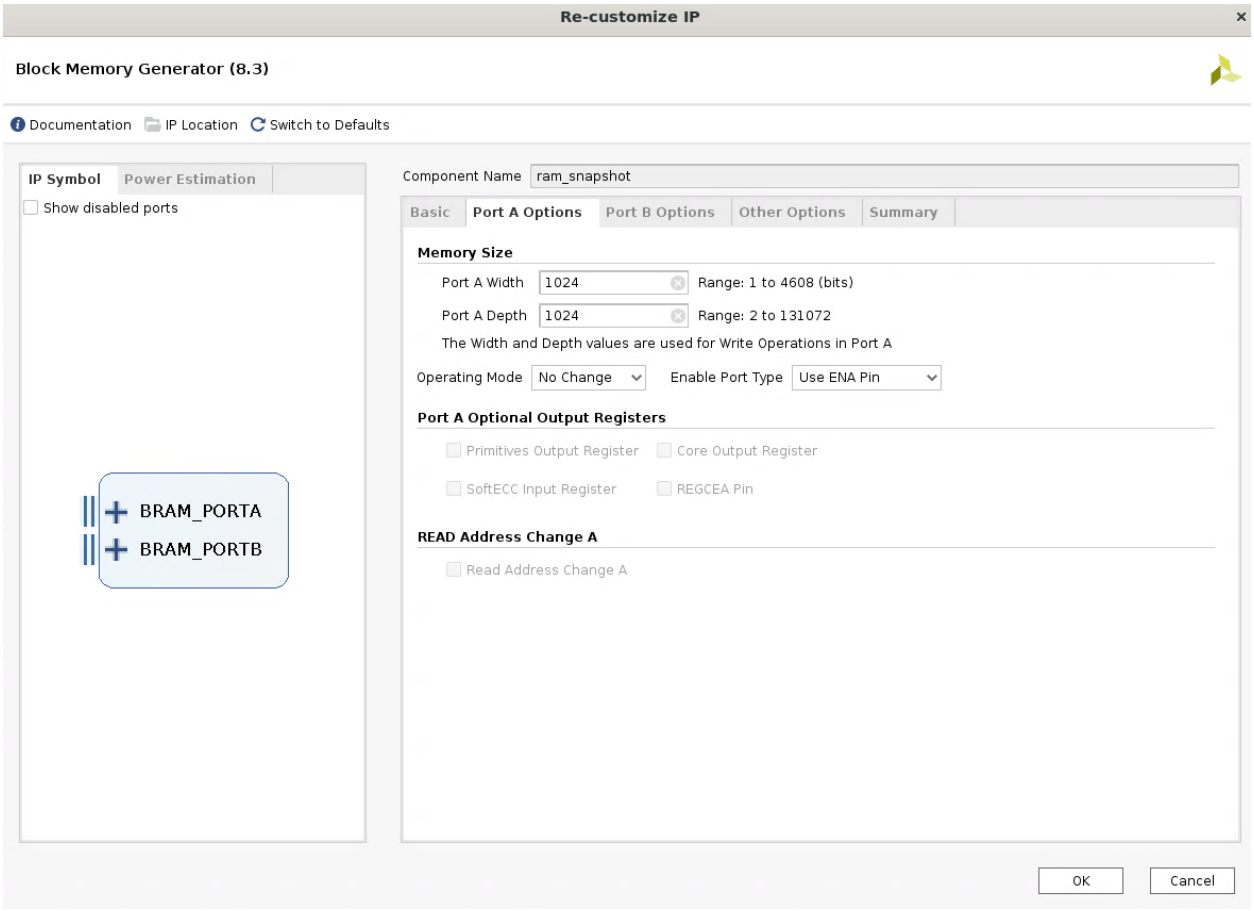

Re-customize IP
Multiplier (12.0)
Documentation
IP Location
Switch to Defaults
IP Symbol
Information
Show disabled ports
CLK
A[15:0]
B[15:0]
P[31:0]
Component Name
mult_fir
Basic
Output and Control
Multiplier Type
Parallel Multiplier
Constant Coefficient Multiplier
Input Options
P = A * B
Data Type
Signed
Signed
Width
16
16
Range: 2...64
Range: 2...64
Multiplier Construction
Use Mults
Optimization Options
Speed Optimized
Area:Optimizes the multiplier for DSP48 slice resources by splitting the multiplication between DSP48 slices and slice logic
Speed:Optimizes the multiplier for performance using as many DSP48 slices as necessary
OK
Cancel

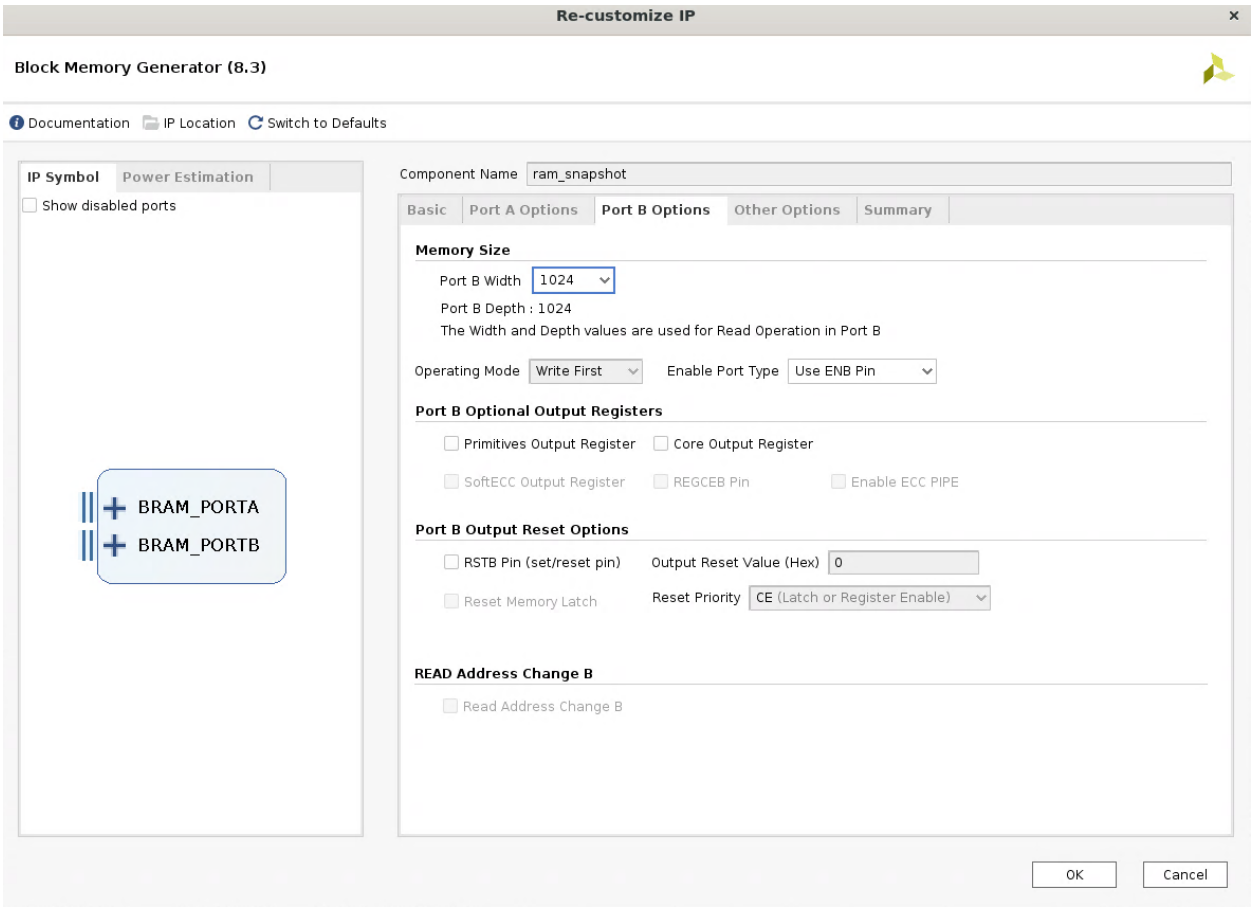


ram_snapshot

- In the 'Flow Navigator' window, select 'IP Catalog', search and open 'Multiplier (12.0)'
- Config the IP as below:


Re-customize IP
Block Memory Generator (8.3)
Documentation
IP Location
Switch to Defaults
IP Symbol
Power Estimation
Show disabled ports
BRAM_PORTA
BRAM_PORTB
Component Name
ram_snapshot
Basic
Port A Options
Port B Options
Other Options
Summary
Interface Type
Native
Generate address interface with 32 bits
Memory Type
Simple Dual Port RAM
Common Clock
ECC Options
ECC Type
No ECC
Error Injection Pins
Single Bit Error Injection
Write Enable
Byte Write Enable
Byte Size (bits)
9
Algorithm Options
Defines the algorithm used to concatenate the block RAM primitives.
Refer datasheet for more information.
Algorithm
Minimum Area
Primitive
8kx2
OK
Cancel


Re-customize IP
Block Memory Generator (8.3)
Documentation
IP Location
Switch to Defaults
IP Symbol
Power Estimation
Show disabled ports
BRAM_PORTA
BRAM_PORTB
Component Name
ram_snapshot
Basic
Port A Options
Port B Options
Other Options
Summary
Memory Size
Port A Width
1024
Range: 1 to 4608 (bits)
Port A Depth
1024
Range: 2 to 131072
The Width and Depth values are used for Write Operations in Port A
Operating Mode
No Change
Enable Port Type
Use ENA Pin
Port A Optional Output Registers
Primitives Output Register
Core Output Register
SoftECC Input Register
REGCEA Pin
READ Address Change A
Read Address Change A
OK
Cancel



3. build the firmware bit file

Go to the TCL console command field and type:

```
devkit_build
```

4. load the bit file to target FPGA