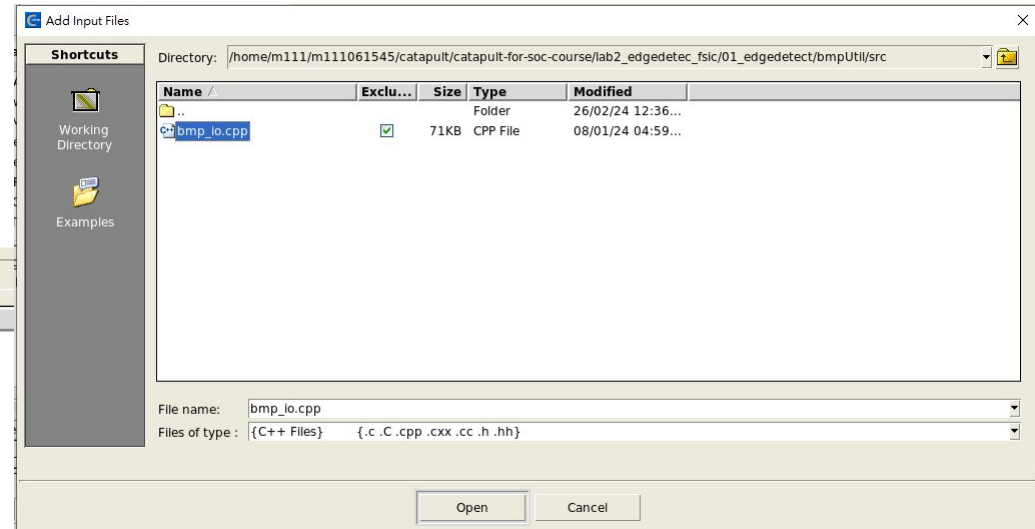
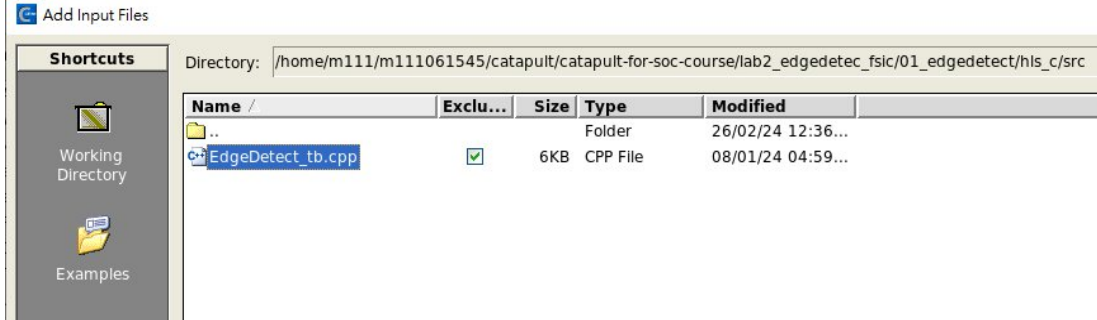
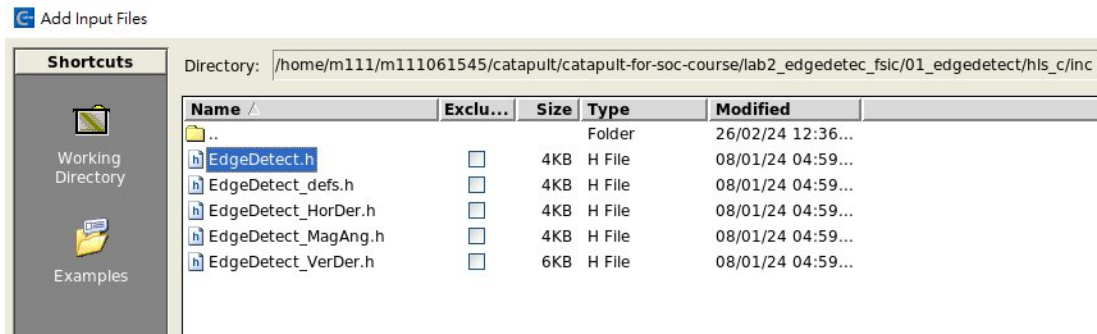


Catapult FPGA (Edge Detect) workflow

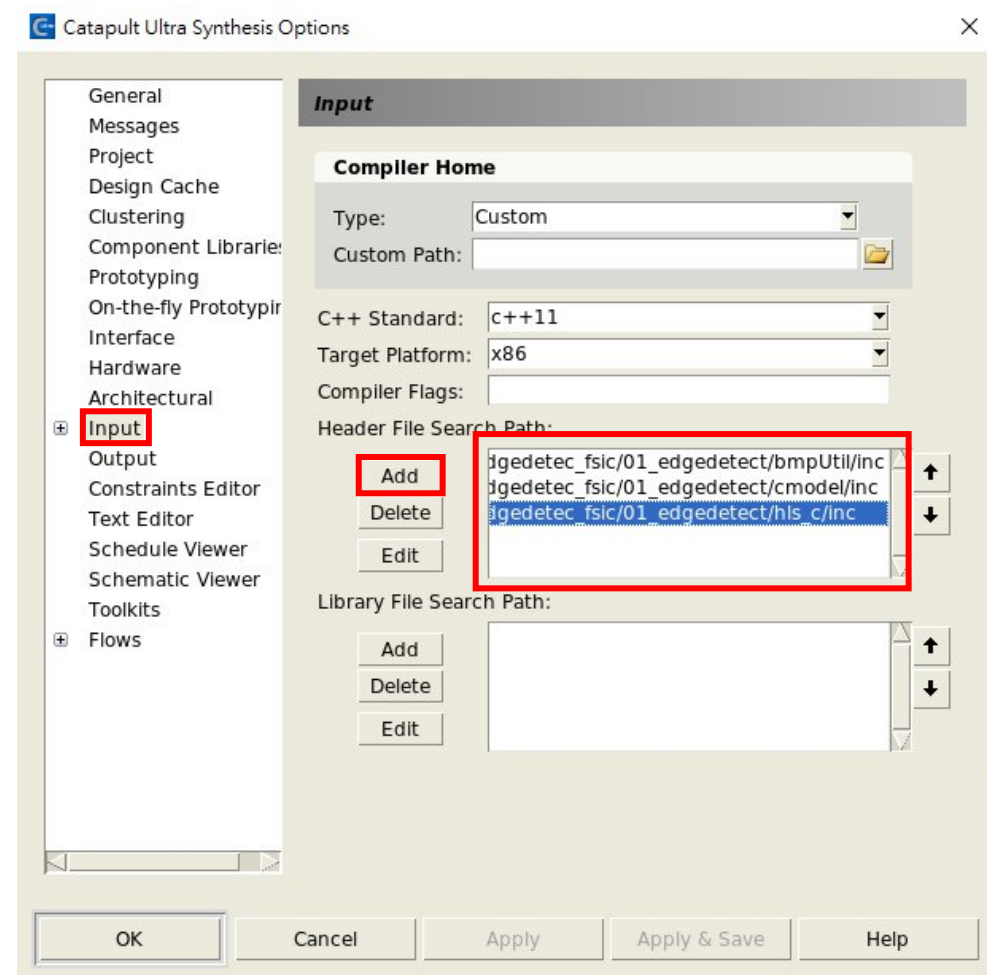
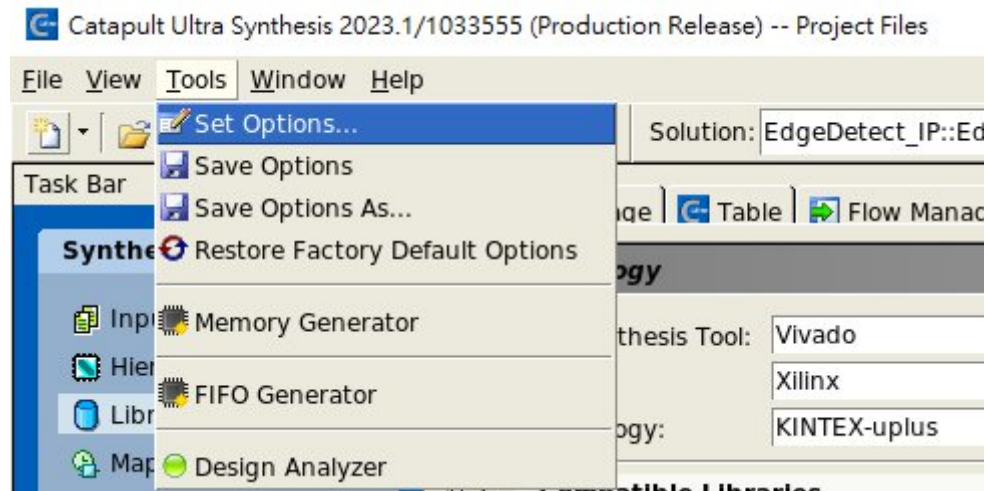
TA:陳揚哲

Add input file

- Add EdgeDetect.h(hls_c/inc), EdgeDetect_tb.cpp(hls_c/src), and bmp_io.cpp(bmpUtil/src)
 - Note: these file should be set as excluded

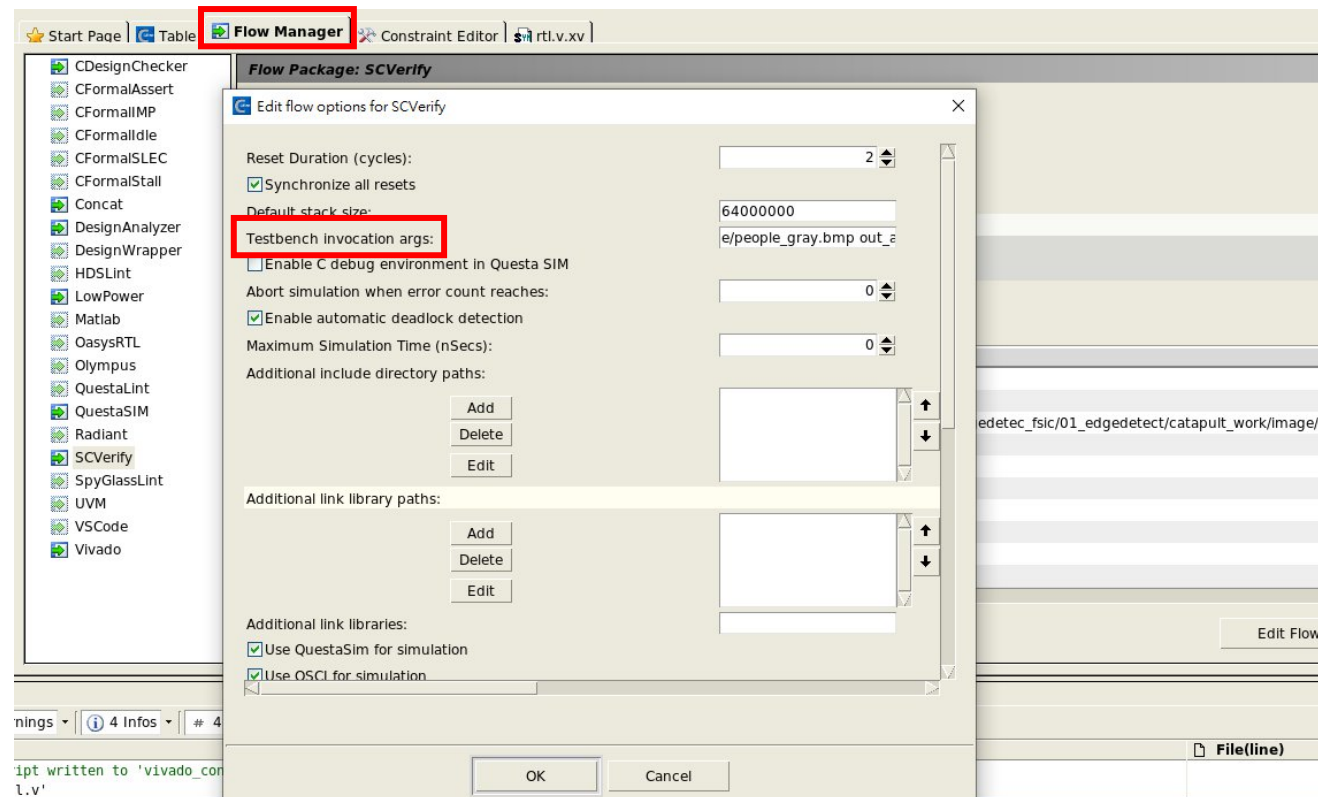


Add include path



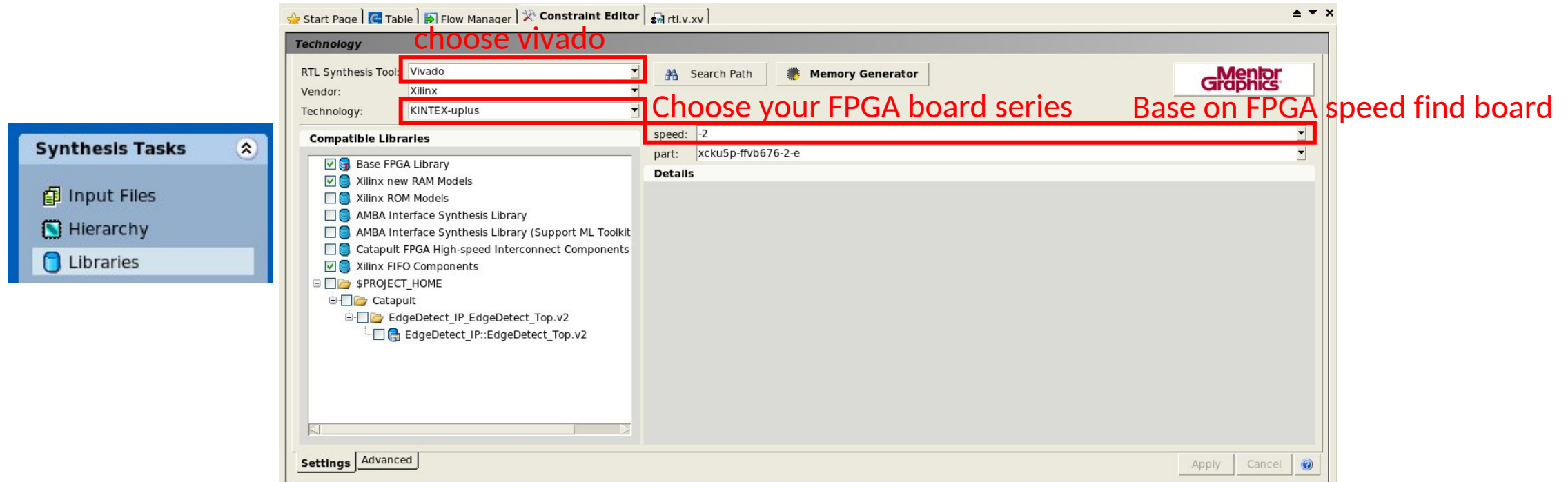
Add testbench invocation args

- Add the directory (~/.image/people_gray.bmp out_algorithm.bmp out_hw.bmp)



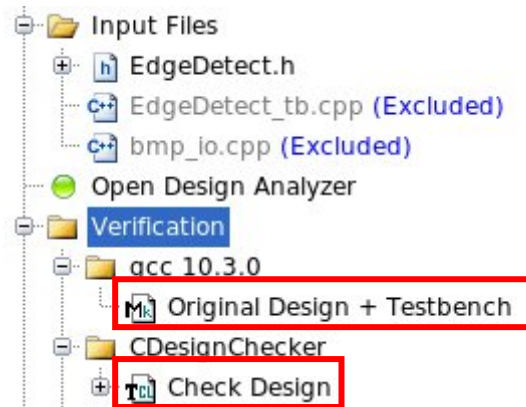
Choose library (FPGA board)

- You can choose your own FPGA board for catapult project
- Here I use KCU116 board for example
- Check Xilinx new RAM Model, Xilinx FIFO Components



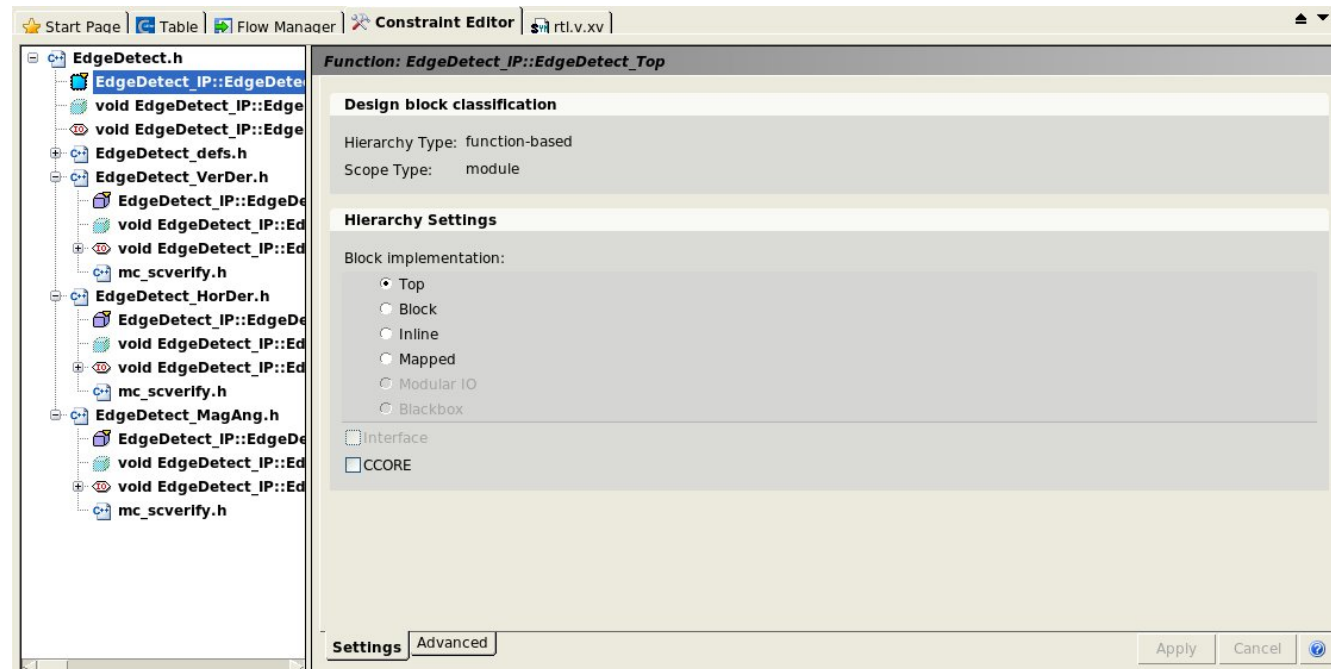
GCC and CDesignChecker

- Double click **Original Design + Testbench** and **Check Design**



Top-down Process

- Here use Top-down process for example
 - Choose top module as top in block implementation
- Suggest bottom-up process. (Step_by_step_lab2_EdgeDetect.pdf)
 - *_Ver -> *_Hor-> *_Mag -> *_Top



Mapping

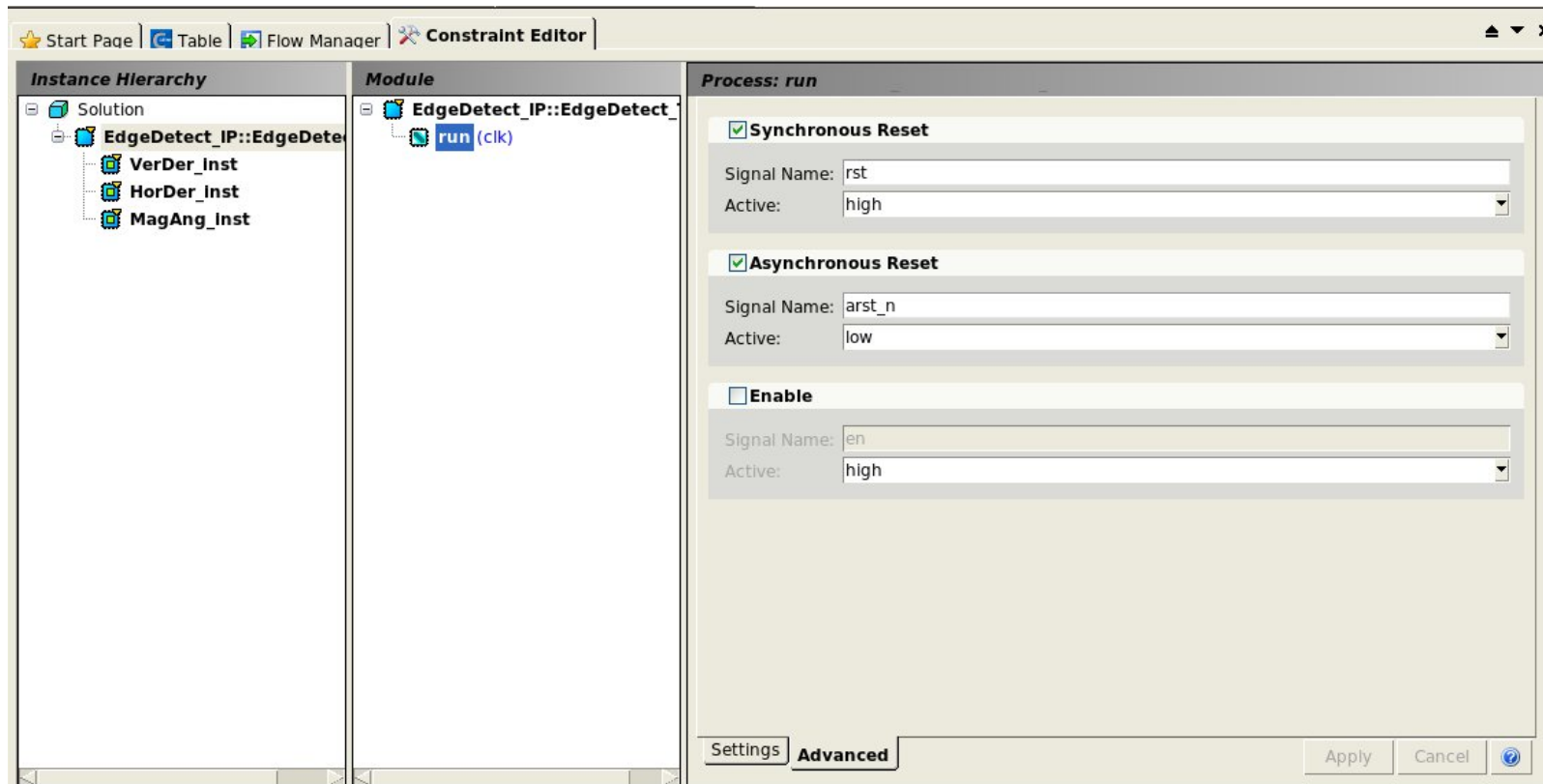
- Set clock as 100 MHz

The screenshot displays the Xilinx Constraint Editor interface. The top menu bar includes 'Start Page', 'Table', 'Flow Manager', and 'Constraint Editor'. The left pane, titled 'Instance Hierarchy', shows a tree structure with 'Solution' at the root, followed by 'EdgeDetect_IP::EdgeDetect', and then three sub-instances: 'VerDer_Inst', 'HorDer_Inst', and 'MagAng_Inst'. The middle pane, titled 'Module', shows the 'EdgeDetect_IP::EdgeDetect' module with a 'run (clk)' component. The right pane, titled 'Process: run', contains a timing diagram showing a clock signal 'clk' with a period of 10 ns. Below the diagram, the 'Process Clock' is set to 'clk'. The 'Frequency' is set to 100 MHz, 'Period' is 10 ns, 'High-Time' is 5 ns, 'Offset' is 0 ns, 'Edge' is 'rising', and 'Uncertainty' is 0 ns. A 'Clock States' table is also visible, showing a 'default' state with a 'Period' of 10 ns.

Name	Period
default	10

Mapping

- Check Asynchronous Reset



Architecture

The screenshot displays the Xilinx ISE Constraint Editor interface, specifically the **Architecture** tab. The interface is divided into several panels:

- Task Bar:** Shows the current tab as **Constraint Editor**.
- Synthesis Tasks:** A list of tasks including Input Files, Hierarchy, Libraries, Mapping, **Architecture** (highlighted), Resources, Schedule, RTL, Power Report (Pre Power Opt), and Power RTL.
- Project Files:** A tree view showing the project structure, including **EdgeDetect_IP::EdgeDetect_Top.v2 (Passed)** and its sub-files like **EdgeDetect.h**, **EdgeDetect_tb.cpp (Excluded)**, and **bmp_io.cpp (Excluded)**.
- Instance Hierarchy:** A tree view showing the hierarchy of instances, including **EdgeDetect_IP::EdgeDetect** (highlighted), **VerDer_Inst**, **HorDer_Inst**, and **MagAng_Inst**.
- Module:** A tree view showing the module structure, including **EdgeDetect_IP::EdgeDetect** (highlighted), **Interface**, **dat in:rsc (1x8)**, **widthIn:rsc (1x11)** (highlighted), **heightIn:rsc (1x10)**, **magn:rsc (1x9)**, **angle:rsc (1x8)**, and **Interconnect**.
- Resource: widthIn:rsc:** A panel showing the properties of the selected resource. It includes a **Resource Type** dropdown set to **[DirectInput]**, and sections for **Input Delay** and **Output Delay**. Each section has fields for **Library Delay**, **Inherited Delay**, **Port Delay**, and **Total**, all set to **0 ns**. A note below each section states: **Total Delay = 'Library' + 'Port' if specified, else 'Inherited'**.

At the bottom of the Resource Properties panel, there are tabs for **Settings** and **Mapping**, and buttons for **Apply** and **Cancel**.

Architecture

The screenshot displays the Xilinx ISE Constraint Editor interface, showing the architecture of the **EdgeDetect_IP** module and the configuration for the **dat_in:rsc** resource.

Task Bar: Shows the current window as the **Constraint Editor**.

Synthesis Tasks: A list of tasks including Input Files, Hierarchy, Libraries, Mapping, Architecture (selected), Resources, Schedule, RTL, Power Report (Pre Power Opt), and Power RTL.

Project Files: A list of files including **EdgeDetect_IP::EdgeDetect_Top.v2** (Passed), **EdgeDetect.h**, **EdgeDetect_tb.cpp** (Excluded), **bmp_io.cpp** (Excluded), **Open Design Analyzer**, **Verification**, **Output Files**, and **Synthesis**.

Instance Hierarchy: Shows the hierarchy of the **EdgeDetect_IP** module, including **VerDer_Inst**, **HorDer_Inst**, and **MagAng_Inst**.

Module: Shows the **EdgeDetect_IP::EdgeDetect** module, including the **Interface** section with **dat_in:rsc (1x8)** (highlighted in red), **widthIn:rsc (1x11)**, **heightIn:rsc (1x10)**, **magn:rsc (1x9)**, and **angle:rsc (1x8)**, and the **Interconnect** section.

Resource: dat_in:rsc: Shows the configuration for the **dat_in:rsc** resource, including the **Resource Type** (ccs_ioport.ccs_in_wait_coupled), **Datasheet** link, **Input Delay** (Library Delay: 0 ns, Inherited Delay: 0 ns, Port Delay: 0 ns, Total: 0 ns), and **Output Delay** (Library Delay: 0 ns, Inherited Delay: 0 ns, Port Delay: 0 ns, Total: 0 ns). The **Settings** and **Mapping** tabs are visible at the bottom.

Architecture

- Set FIFO depth base on your design delay (By bottom-up step)

The screenshot displays the Xilinx ISE software interface with the **Architecture** tab selected in the left-hand menu. The **Instance Hierarchy** pane shows the design structure, including the **EdgeDetect_IP::EdgeDetect** module. The **Module** pane shows the internal components of this module, with the **dat:cns (1x8)** resource highlighted. The **Resource: dat:cns** configuration window is open, showing the **FIFO Depth** set to 0. The **Input Delay** and **Output Delay** sections are also visible, with all delay values set to 0 ns. The **Settings** and **Mapping** tabs are at the bottom of the configuration window.

Synthesis Tasks

- Input Files
- Hierarchy
- Libraries
- Mapping
- Architecture
- Resources
- Schedule
- RTL
- Power Report (Pre Power Opt)
- Power RTL

Project Files

- EdgeDetect_tb.cpp (Excluded)
- bmp_io.cpp (Excluded)
- Open Design Analyzer
- Verification
 - gcc 10.3.0
 - CDesignChecker
 - QuestaSIM
 - RTL VHDL output 'rtl.vhdl' vs Untimed
 - Concat RTL VHDL output 'concat_si
 - RTL Verilog output 'rtl.v' vs Untimed
 - Concat RTL Verilog output 'concat_
- Output Files

Instance Hierarchy

- Solution
 - EdgeDetect_IP::EdgeDetect
 - VerDer_Inst
 - HorDer_Inst
 - MagAng_Inst

Module

- EdgeDetect_IP::EdgeDetect
 - Interface
 - dat_in:rsc (1x8)
 - widthIn:rsc (1x11)
 - heightIn:rsc (1x10)
 - magn:rsc (1x9)
 - angle:rsc (1x8)
 - Interconnect
 - dat:cns (1x8)
 - dy:cns (1x9)
 - dx:cns (1x9)

Resource: dat:cns

Resource Type: ccs_ioport.ccs_pipe

[Datasheet](#)

FIFO Depth: 0

Input Delay

Library Delay: 0 ns
Inherited Delay: 0 ns
Port Delay: ns
Total: 0 ns
Total Delay = 'Library' + 'Port' If specified, else 'Inherited'

Output Delay

Library Delay: 0 ns
Inherited Delay: 0 ns
Port Delay: ns
Total: 0 ns
Total Delay = 'Library' + 'Port' If specified, else 'Inherited'

Settings **Mapping** **Apply** **Cancel**

Architecture

The screenshot displays the Xilinx ISE Constraint Editor interface, which is divided into several panes:

- Task Bar:** Located at the top left, it contains a list of synthesis tasks. The 'Architecture' task is currently selected and highlighted.
- Project Files:** A tree view on the left showing the project structure. It includes files like 'EdgeDetect_tb.cpp (Excluded)', 'bmp_io.cpp (Excluded)', and a 'Verification' folder containing various simulation and compilation outputs.
- Instance Hierarchy:** A tree view in the center-left showing the hierarchy of instantiated blocks. Under 'EdgeDetect_IP::EdgeDetect', there are instances for 'VerDer_Inst', 'HorDer_Inst', and 'MagAng_Inst'.
- Module:** A tree view in the center-right showing the internal structure of the selected module. It includes an 'Interface' section with signals like 'dat_in:rsc (1x8)' and 'widthIn:rsc (1x11)', and an 'Interconnect' section with resources like 'dat:cns (1x8)' and 'dy:cns (1x9)'. The 'dy:cns (1x9)' resource is highlighted with a red box.
- Resource: dy:cns:** A detailed configuration pane on the right for the selected resource. It shows the 'Resource Type' as 'ccs_loport.ccs_pipe'. A red box highlights the 'FIFO Depth' parameter, which is set to 2. Below this, there are sections for 'Input Delay' and 'Output Delay', each with fields for 'Library Delay', 'Inherited Delay', 'Port Delay', and 'Total'. The 'Total' field for both input and output delays is currently set to 0 ns.

At the bottom of the 'Resource: dy:cns' pane, there are tabs for 'Settings' and 'Mapping', and buttons for 'Apply' and 'Cancel'.

Architecture

The screenshot displays the Xilinx ISE software interface, specifically the Architecture window. The left sidebar shows the 'Synthesis Tasks' pane with 'Architecture' selected, and the 'Project Files' pane showing the project structure for 'EdgeDetect_IP::EdgeDetect_Top.v2'. The main area is divided into three panes: 'Instance Hierarchy', 'Module', and 'Resource: line_buf0:rsc'. The 'Module' pane shows the 'run' module with 'line_buf0:rsc' and 'line_buf1:rsc' highlighted. The 'Resource: line_buf0:rsc' pane shows the 'Resource Options' tab with the following settings:

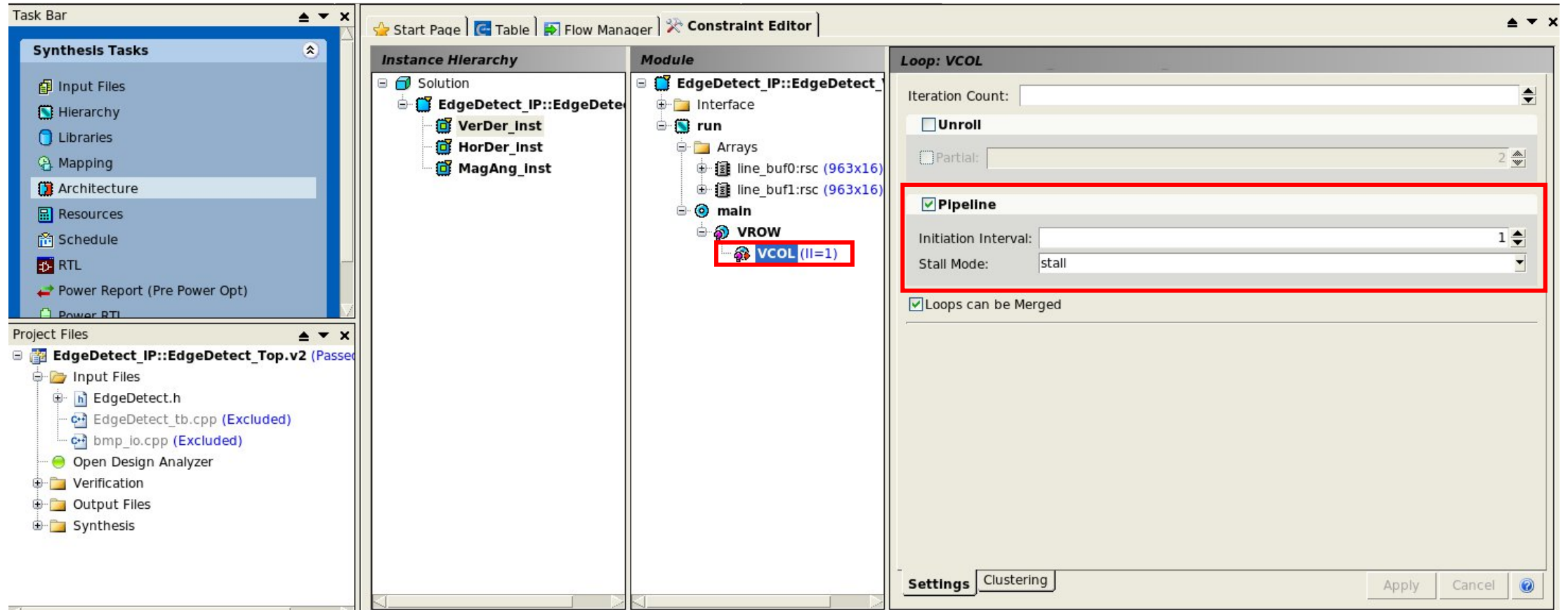
- Resource Type: **Xilinx_RAMSBLOCK_1R1W_RBW**
- latency: 1
- suppress_sim_read_addr_range_errs: 1
- Packing Mode: absolute
- Block Size: 0
- Interleave: 1
- ☒ Externalize
- ☒ Generate External Enable

The 'Input Delay' section shows the following values:

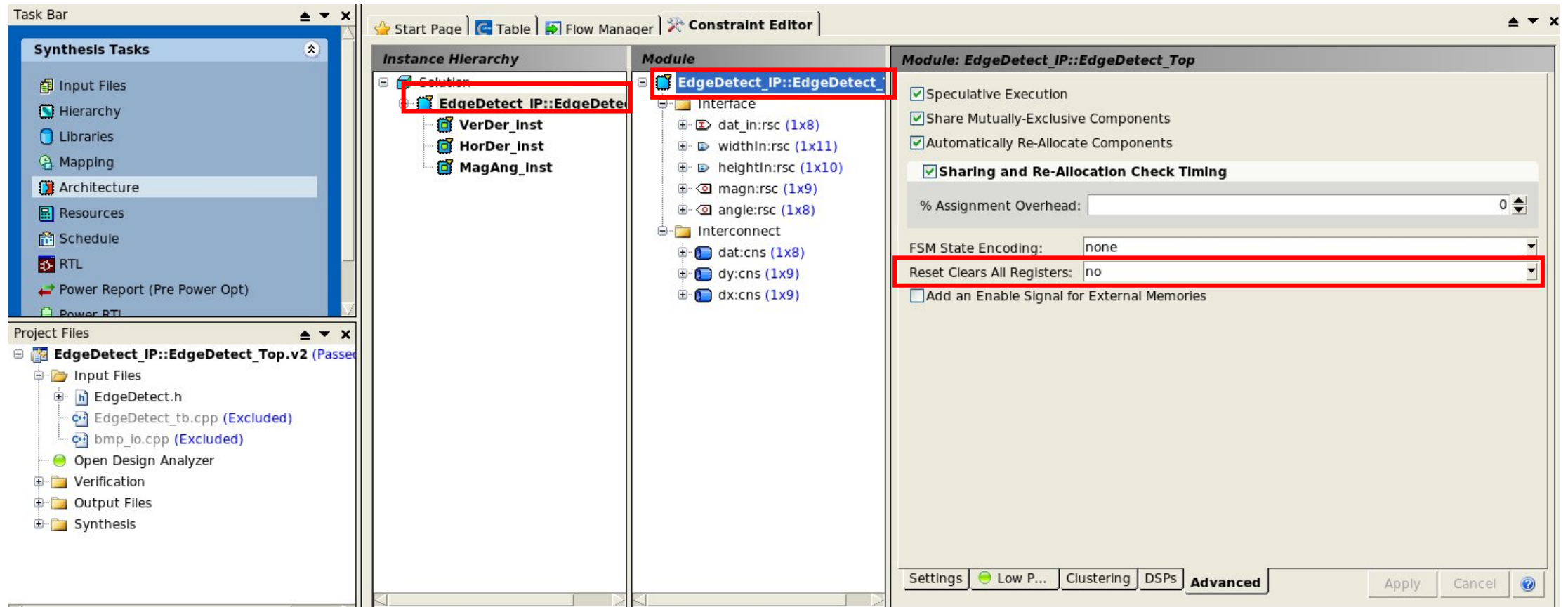
Delay Type	Value	Unit
Library Delay	1.62	ns
Inherited Delay	0	ns
Port Delay		ns
Total	1.62	ns

The total delay is calculated as 'Library' + 'Port' if specified, else 'Inherited'.

Architecture



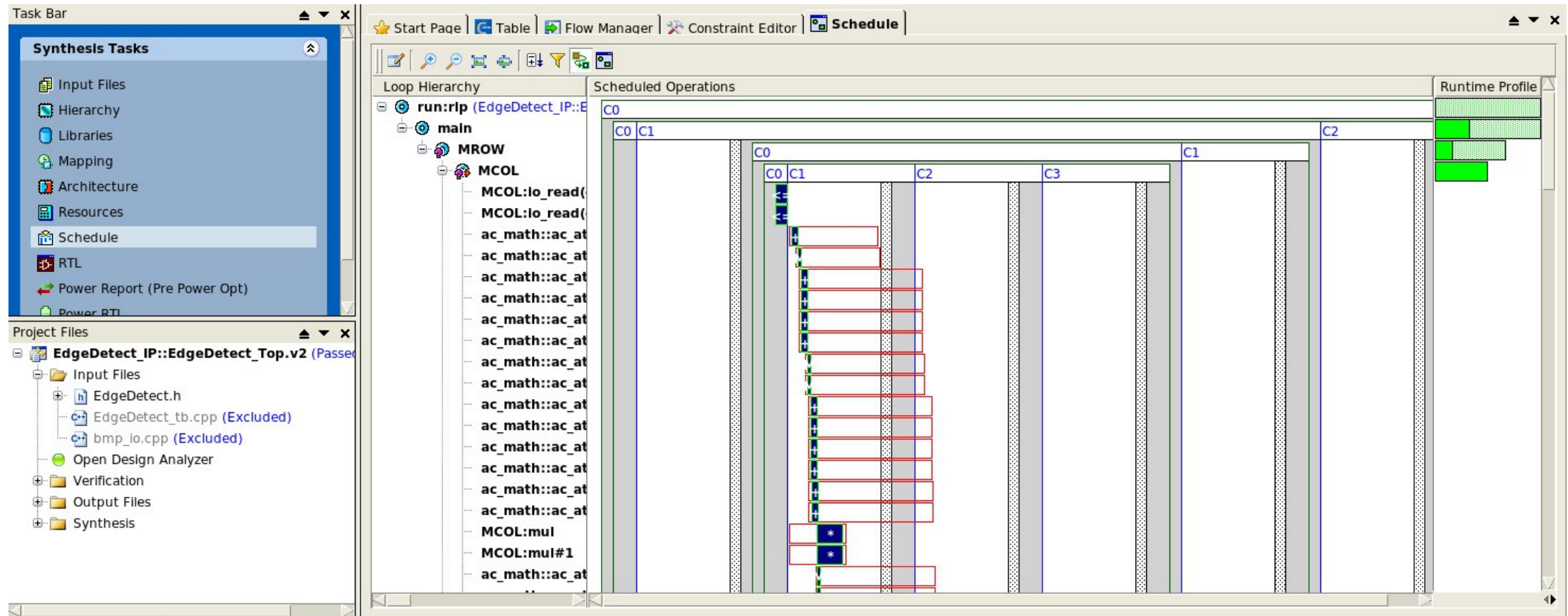
Architecture



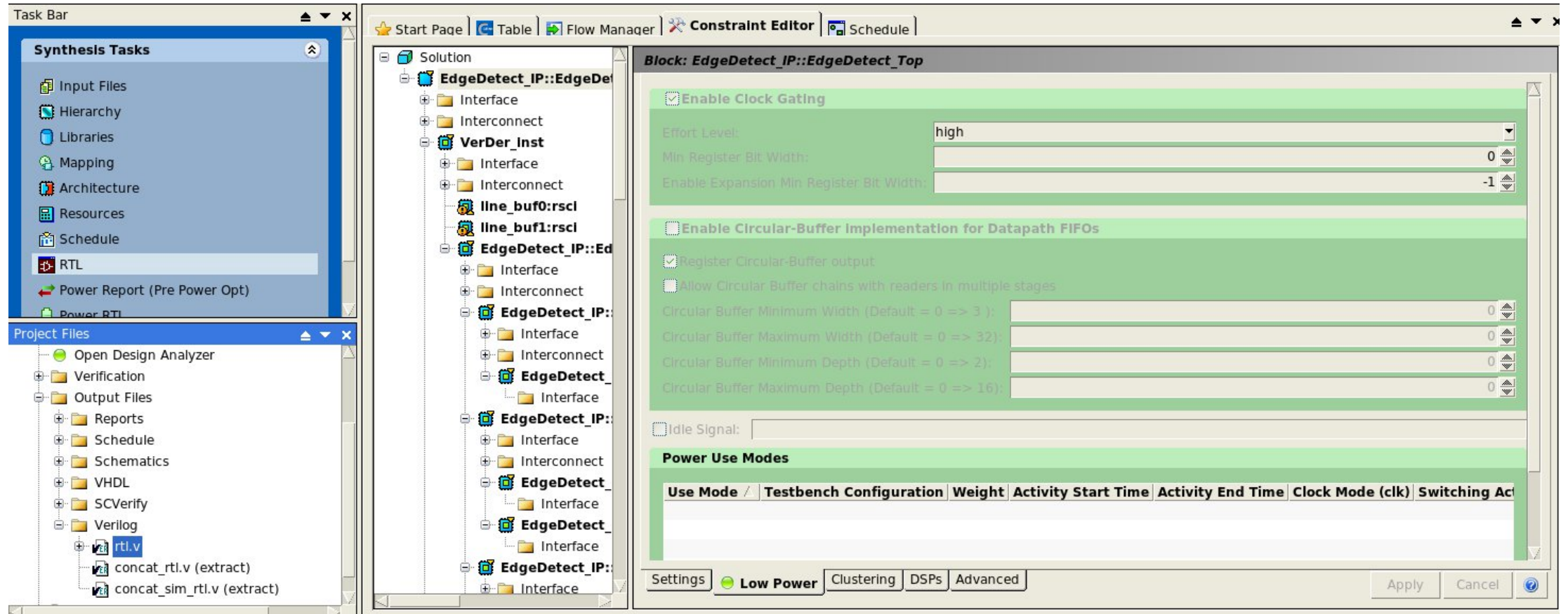
Architecture

The screenshot displays the Xilinx Vivado IDE's Architecture window, which is divided into three main panes. The left pane, titled 'Task Bar', contains a 'Synthesis Tasks' list with options like 'Input Files', 'Hierarchy', 'Libraries', 'Mapping', 'Architecture' (selected), 'Resources', 'Schedule', 'RTL', 'Power Report (Pre Power Opt)', and 'Power RTL'. Below this is the 'Project Files' pane, showing a tree structure for 'EdgeDetect_IP::EdgeDetect_Top.v2 (Passed)' with subfolders for 'Input Files', 'Verification', 'Output Files', and 'Synthesis'. The middle pane is split into two views: 'Instance Hierarchy' and 'Module'. The 'Instance Hierarchy' view shows a tree starting with 'Solution', followed by 'EdgeDetect_IP::EdgeDetect', which contains 'VerDer_Inst', 'HorDer_Inst', and 'MagAng_Inst' (highlighted with a red box). The 'Module' view shows the internal structure of 'EdgeDetect_IP::EdgeDetect', including 'Interface', 'Constant Arrays', 'run', 'main', 'MROW', 'MCOL (II=1)', and 'ac_math::ac_at' (highlighted with a red box). The right pane shows the 'Loop: ac_math::ac_atan2_cordic<9,9,AC_TRN,AC_WRAP,9,9,AC_TRN,AC_WRAP,8,3,AC_TRN,A' configuration. It includes an 'Iteration Count' set to 8, an 'Unroll' checkbox (checked and highlighted with a red box), a 'Partial' checkbox (unchecked), and a 'Loops can be Merged' checkbox (checked). At the bottom, there are tabs for 'Settings' and 'Clustering', and buttons for 'Apply', 'Cancel', and a help icon.

Schedule



Generate RTL



Synthesis report

The screenshot displays a software interface with a 'Task Bar' on the left and a main workspace on the right. The 'Task Bar' contains a 'Synthesis Tasks' panel with a list of tasks: Input Files, Hierarchy, Libraries, Mapping, Architecture, Resources, Schedule, RTL (highlighted), Power Report (Pre Power Opt), and Power RTL. Below this is a 'Project Files' panel showing a tree structure: Open Design Analyzer, Verification, Output Files, Reports, Schedule, Schematics, VHDL, SCVerify, Verilog, and a sub-folder containing rtl.v, concat_rtl.v (extract), and concat_sim_rtl.v (extract).

The main workspace has a top menu bar with 'Start Page', 'Table' (highlighted with a red box), 'Flow Manager', 'Constraint Editor', and 'Schedule'. Below the menu bar is a 'Report' dropdown menu (also highlighted with a red box) showing a list of reports: General (selected), Run Time, Memory Usage, Timing, and Area Score. To the right of the dropdown is a table with the following data:

	Frequency...	Latency...	Throug...	Throug...	Slack	Total Area
Solution General						
Edge Run Time						
Edge Memory Usage						
Edge Timing	6	60.00	6	60.00	0.55	3273.35
Edge Area Score						

Design Analyzer

The screenshot displays the Catapult Design Analyzer interface. On the left, the 'Project Files' pane shows a tree structure for 'EdgeDetect_IP::EdgeDetect_Top.v2 (Passed)'. The 'Input Files' folder is expanded, showing 'EdgeDetect.h', 'EdgeDetect_tb.cpp (Excluded)', and 'bmp_io.cpp (Excluded)'. The 'Open Design Analyzer' button is highlighted with a red box. Below it are 'Verification', 'Output Files', and 'Synthesis' folders.

The main window shows a circuit diagram of 'EdgeDetect_Top'. A menu is open over the diagram, with the 'Reset Layout' option highlighted by a red box. The menu options include: 'Save Perspective', 'Reset Layout', 'Adviser', 'Bill Of Materials', 'C++', 'Critical Paths', 'Failed Schedule', 'Power Report', 'Register Sharing', 'Resource Sharing', 'RTL vs C++', 'Reports', 'Physical Data', 'Adviser', 'Critical Paths', 'Failed Schedule', 'Input Files', 'Message Log', 'Modules and Scopes', 'Object Details', 'Objects', 'Output Files', 'Project Contents', 'Schedule', 'Schematic', 'Search Files', 'Status Window', 'Files', 'Classes', 'Call Stack', and 'References'.

On the right, the 'Objects' pane lists the components of the design:

Name	Type
EdgeDetect_IP::EdgeDetect_Top	Hier Part
VerDer_inst	INSTANCE
HorDer_inst	INSTANCE
MagAng_inst	INSTANCE
angle:asn(angle:rsc.dat)	Assign
angle:asn(angle:rsc.vld)	Assign
angle:conc	Concurrent
dat:conc	Concurrent
dat_in:asn(dat_in:rsc.rdy)	Assign

At the bottom right, a 'What's New' sidebar for 'Catapult Design Analyzer 2023.1' is visible, featuring a green play button icon and the text 'Here's what's new in Design Analyzer 2023.1: Incremental Project Loading with Active'.

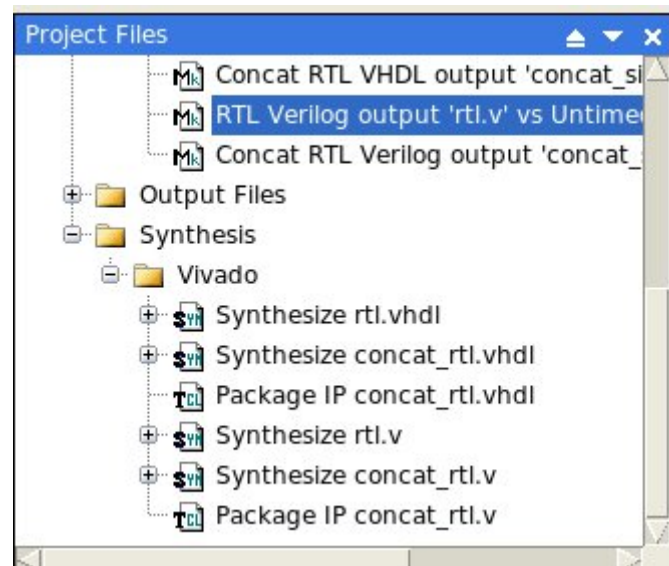
RTL Verification (Cont.)

Run all

The screenshot displays the Questa Sim-64 2021.1.1 interface. The **Project Files** window on the left shows a project structure with folders like **gcc 10.3.0**, **CDesignChecker**, and **QuestaSIM**. Under **QuestaSIM**, there are files for RTL VHDL and Verilog output, with the Verilog output file **'rtl.v' vs Untimed** highlighted in a red box. The **Instance** window in the center lists various design units and their types. The **Objects** window on the right shows a list of variables and their values, with the **var_idone** variable highlighted. The **Wave** window on the far right displays a timing diagram with signals like **clk**, **Master_rst**, and **cpp_testbench_a...**.

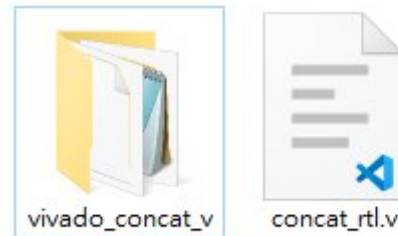
Synthesis workflow (Catapult GUI)





- If your environment have Vivado project, set up Vivado directory and double click Synthesize concat_rtl.v
- If not, follow the synthesis workflow



Synthesis workflow (Vivado)

- Download the file of concat_rtl.v and the folder of vivado_concat_v
- File under vivado_concat_v folder



 concat_rtl.v.xv	2024/2/28 上午 04:45	XV 檔案	6 KB
 concat_rtl.v.xv.sdc	2024/2/28 上午 04:45	SDC 檔案	3 KB
 concat_rtl.v.xv.signoff.sdc	2024/2/28 上午 04:45	SDC 檔案	1 KB
 concat_v_package_ip.tcl	2024/2/28 上午 04:45	TCL 檔案	2 KB

Change all of the directory in xv file

```
1## HLS SP synthesis script
2## Generated for stage extract
3## Vivado mode is Non-project
4
5# Version check
6set vv [version -short]
7if { [regexp {(\\d+)\\.((\\d+).*)} $vv all major minor] } {
8    if { ($major < 2020) || ($major == 2020 && $minor < 2) } {
9        puts "Vivado version (v${vv}) is not compatible with version used for the Catapult library (v2020.2).\"
10    }
11}
12# Reporting settings
13puts "-- Requested 4 fractional digits for design 'EdgeDetect_IP_EdgeDetect_Top' timing\"
14puts "-- Requested 4 fractional digits for design 'EdgeDetect_IP_EdgeDetect_Top' capacitance\"
15puts "-- Characterization mode: p2p \"
16
17puts "-- Synthesis Timing report: '/home/ubuntu/catapult_test/vivado_concat_v/timing_summary_synth.rpt' \"
18puts "-- Synthesis Utilization report: '/home/ubuntu/catapult_test/vivado_concat_v/utilization_synth.rpt' \"
19if { ([info exists env(Xilinx_RUN_PNR)] && $env(Xilinx_RUN_PNR) ) ||
20      ([info exists env(Xilinx_BITGEN)] && $env(Xilinx_BITGEN) ) } {
21    puts "-- Routed Timing report: '/home/ubuntu/catapult_test/vivado_concat_v/timing_summary_routed.rpt' \"
22    puts "-- Routed Utilization report: '/home/ubuntu/catapult_test/vivado_concat_v/utilization_placed.rpt' \"
23}
24
```

Change all of the directory in xv file

```
25 # Environment variable settings
26 global env
27 ## set CATAPULT_HOME "/usr/cadtool/mentor/Catapult/2023.1/Mgc_home"
28 ## Set the variable for file path prefixing
29 set RTL_TOOL_SCRIPT_DIR /home/ubuntu/catapult_test/vivado_concat_v
30 set RTL_TOOL_SCRIPT_DIR [file dirname [file normalize [info script] ] ]
31 puts "-- RTL_TOOL_SCRIPT_DIR is set to '$RTL_TOOL_SCRIPT_DIR' "
32 # Vivado Non-Project mode script starts here
33 puts "=====
34 puts "Catapult driving Vivado in Non-Project mode"
35 puts "=====
36 set outputDir /home/ubuntu/catapult_test/vivado_concat_v
37 set outputDir $RTL_TOOL_SCRIPT_DIR
38 #file mkdir $outputDir
39 #
40 # STEP#1: setup design sources and constraints
41 #
42 create_project -force tcl_concat_v
43 read_verilog /home/ubuntu/catapult_test/concat_rtl.v
44 # set up XPM libraries for XPM-related IP like the Catapult Xilinx_FIFO
45 set_property XPM_LIBRARIES {XPM_CDC XPM_MEMORY XPM_FIFO} [current_project]
46 read_xdc /home/ubuntu/catapult_test/vivado_concat_v/concat_rtl.v.xv.sdc
47 set_property part xcku5p-ffvb676-2-e [current_project]
48 #
49 # STEP#2: run synthesis, report utilization and timing estimates, write checkpoint design
50 #
51 synth_design -cascade_dsp auto -top EdgeDetect_IP_EdgeDetect_Top -part xcku5p-ffvb676-2-e -mode out_of_context -include_dirs ""
52 write_checkpoint -force $outputDir/post_synth
53 set viv_report_dir /home/ubuntu/catapult_test/vivado_concat_v
54 report_utilization -file $viv_report_dir/utilization_synth.rpt
55 report_timing_summary -path_type summary -file $viv_report_dir/timing_summary_synth.rpt
56 report_timing -nworst 1 -from [all_inputs] -to [all_outputs] -file $viv_report_dir/timing_summary_synth.rpt -append
57 if { [llength [all_clocks] ] > 0 } {
58     report_timing -nworst 1 -from [all_inputs] -to [all_clocks] -file $viv_report_dir/timing_summary_synth.rpt -append
59     report_timing -nworst 1 -from [all_clocks] -to [all_clocks] -file $viv_report_dir/timing_summary_synth.rpt -append
60     report_timing -nworst 1 -from [all_clocks] -to [all_outputs] -file $viv_report_dir/timing_summary_synth.rpt -append
61 }
```

Synthesis by GUI

- Open Vivado
- Create a project with any board
- Execute xv file, (or the command in xv file), in tcl console
- It will generate the design of the board you choose in catapult

Synthesis by Command Shell (Linux)

- Create file named run_vivado.
- Specify the directory of xv file and source it by vivado command.
- Then execute run_vivado.



```
echo "start vivado project"

# ----- remove vivado project log files -----#
rm -rf ./NA
rm -rf ./Xil
rm -f timing_report.log
rm -f vivado*.jou
rm -f vivado*.log

# ----- Re-build vivado project -----#

vivado -source concat_rtl.v.xv -mode tcl
```

A terminal window titled 'ubuntu@ubuntu2004: ~/catapult_test/vivado_concat_v'. The terminal shows the output of the script, including mounting messages and Vivado version information.

```
ubuntu@ubuntu2004: ~/catapult_test/vivado_concat_v
mount: /home/ubuntu/tools: /dev/sdb1 already mounted on /home/ubuntu/tools.
mount: /tools: /dev/sdb1 already mounted on /home/ubuntu/tools.
ubuntu@ubuntu2004:~/catapult_test/vivado_concat_v$ ./run_vivado
start vivado project

***** Vivado v2022.1 (64-bit)
**** SW Build 3526262 on Mon Apr 18 15:47:01 MDT 2022
**** IP Build 3524634 on Mon Apr 18 20:55:01 MDT 2022
** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.
```

For implement and bitstream (opt.)

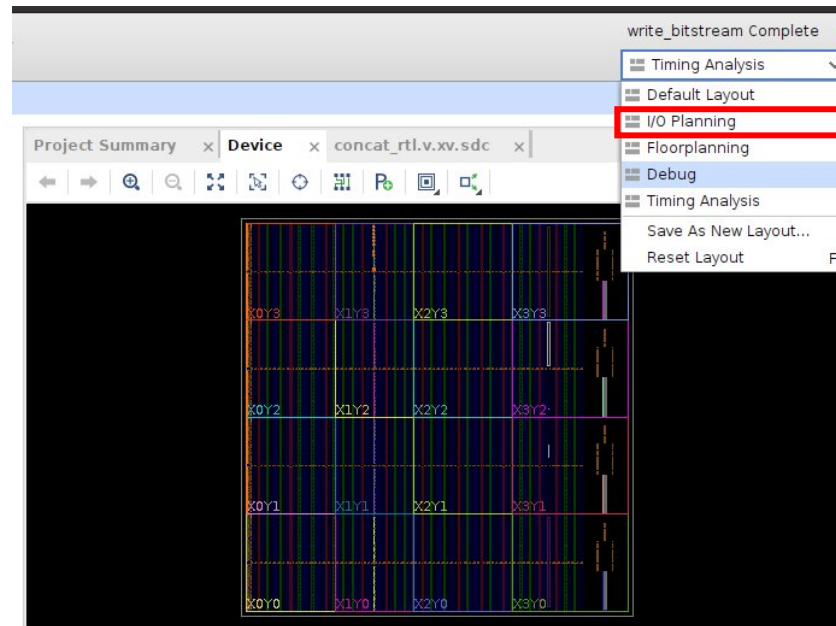
- For implement and generate bitstream, you need specify the constraints.
- You can specify the constraints on your own, or assigned by vivado
- Note: If you change the design board, you need to make sure the timing is satisfied.

Set up constraints

- Set up the arst_n constraints

```
set_input_delay -clock [get_clocks clk] -min -add_delay 0.000 [get_ports arst_n]  
set_input_delay -clock [get_clocks clk] -max -add_delay 0.000 [get_ports arst_n]
```

- Open synthesis design, and choose I/O planning for set up constraints



Check Fixed and set I/O standard

Tcl Console Messages Log Reports Design Runs Package Pins I/O Ports x										
?										
Q [Zoom In] [Zoom Out] [Fit] + [Fit] [Settings]										
Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	S
v [Icon] All ports (165)										
> [Icon] angle_rsc_dat (8)	OUT			<input checked="" type="checkbox"/>	87	LVC MOS18	1.800		12	
> [Icon] dat_in_rsc_dat (8)	IN			<input checked="" type="checkbox"/>	67	LVC MOS18	1.800			
> [Icon] heightIn (10)	IN			<input checked="" type="checkbox"/>	65	LVC MOS18	1.800			
> [Icon] line_buf0_rsc_d (16)	OUT			<input checked="" type="checkbox"/>	65	LVC MOS18	1.800		12	
> [Icon] line_buf0_rsc_q (16)	IN			<input checked="" type="checkbox"/>	67	LVC MOS18	1.800			
> [Icon] line_buf0_rsc_radr (10)	OUT			<input checked="" type="checkbox"/>	65	LVC MOS18	1.800		12	
> [Icon] line_buf0_rsc_wadr (10)	OUT			<input checked="" type="checkbox"/>	66	LVC MOS18	1.800		12	
> [Icon] line_buf1_rsc_d (16)	OUT			<input checked="" type="checkbox"/>	66	LVC MOS18	1.800		12	

Generate bitstream

- Click generate bitstream, it will generate the bitstream file under the directory of ***project_name***.runs/impl_1/***design.bit***