

# Step-By-Step Lab1: FIR

Yuan-teng Chang, Catapult HLS AE  
[yuan-teng.chang@siemens.com](mailto:yuan-teng.chang@siemens.com)

# Objectives

- Learn the Catapult HLS C++ flow
- Understand the loop unrolling and pipelining
- Understand the memory interface
- Learn how to design with multiple blocks

# Environment Variables

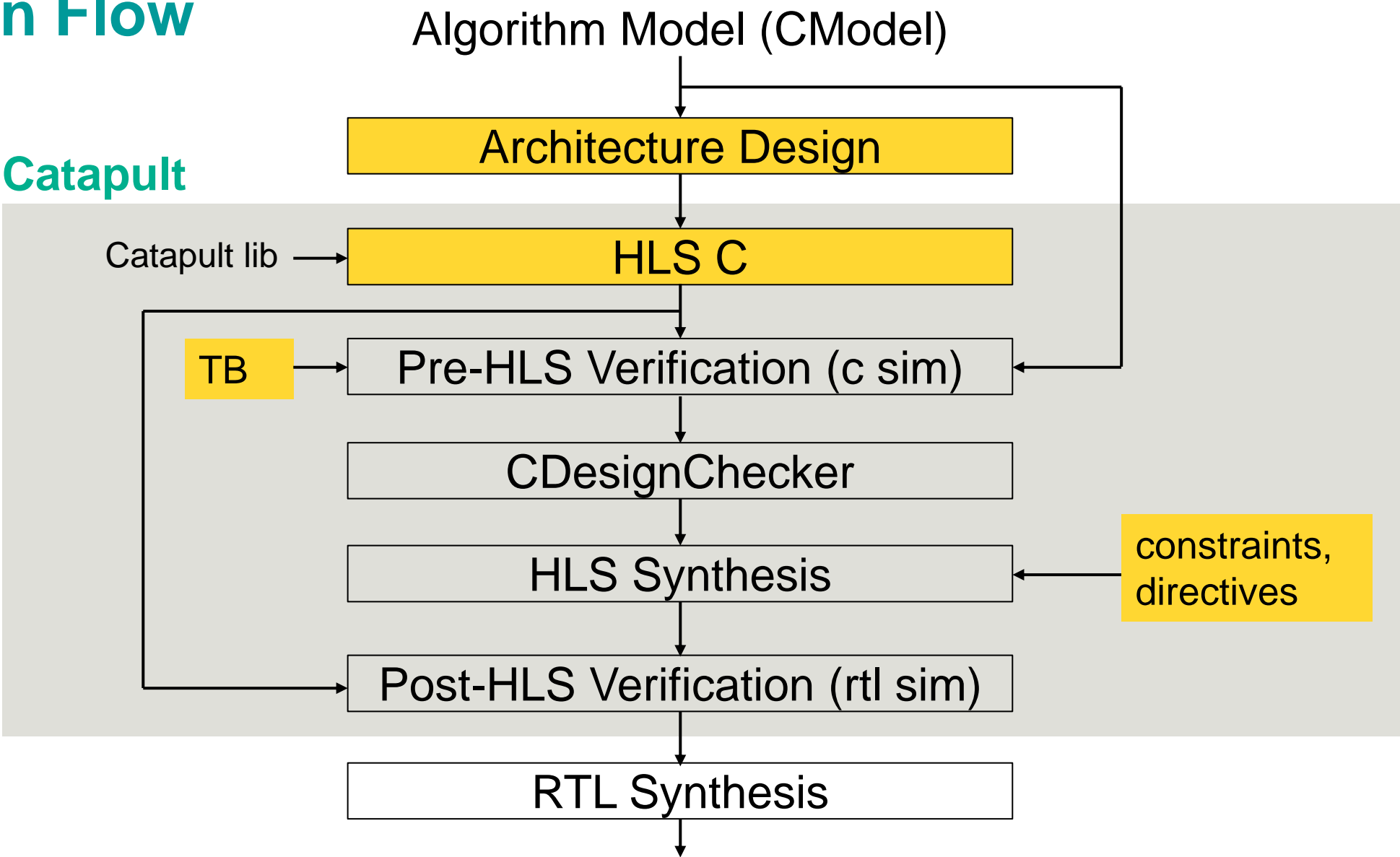
To run Catapult HLS and QuestaSim, a few environment variables are required to be defined

- Define them in a shell script like setup-catapult.csh and source it
- Modify the path depending on the tool install path in your workstation

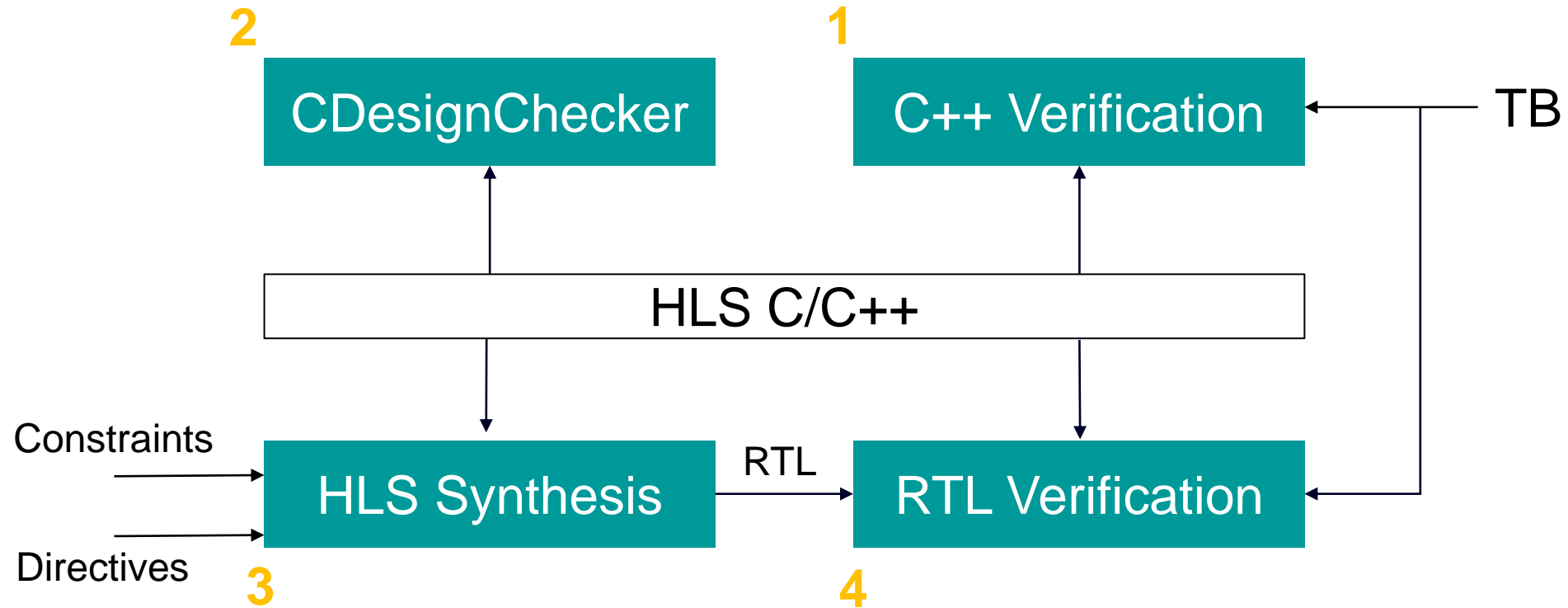
```
2 echo "Catapult setup"
3 setenv MGC_HOME /usr/local/bin/Siemens_EDA/Catapult_Synthesis_2022.2_1-1019737/Mgc_home
4
5 setenv CXX_HOME $MGC_HOME
6 setenv SYSTEMC_INCDIR $MGC_HOME/shared/include
7 setenv SYSTEMC_LIBDIR $MGC_HOME/shared/lib
8 set path = ($path ${MGC_HOME}/bin)
9
10 setenv LD_LIBRARY_PATH $SYSTEMC_LIBDIR
11 setenv LD_LIBRARY_PATH ${MGC_HOME}/lib:$LD_LIBRARY_PATH
12
13 echo "QuestaSim setup"
14 setenv MODEL_TECH /usr/local/bin/Siemens_EDA/Questa_Sim_2022_4/questasim/bin
15 setenv QUESTA_HOME /usr/local/bin/Siemens_EDA/Questa_Sim_2022_4/questasim
16 set path = ($path ${MODEL_TECH})
```

# Design Flow

## Catapult



# Catapult Flow



# | Walkthrough of Catapult C++ Flow

An 8-tap FIR

# fir\_ref.h (Reference model)

$$y = \sum_{i=0}^7 c_i \cdot r[i]$$
$$= c_0 r[0] + c_1 r[1] + \dots + c_7 r[7]$$

```
class fir_ref {  
    private:  
        int regs[8];  
  
    public:  
        fir_ref() {  
            for (int i = 7; i >= 0; i--)  
                { regs[i] = 0; }  
        }  
        ...  
};
```

```
void run(int input, int coeffs[8], int &output)  
{  
    int temp = 0;  
  
    for (int i = 7; i >= 0; i--) {  
        if ( i == 0 ) {  
            regs[i] = input;  
        } else {  
            regs[i] = regs[i-1];  
        }  
    }  
  
    for(int i = 0; i < 8; i++)  
        temp += regs[i] * coeffs[i];  
  
    output = temp >> 11;  
};
```

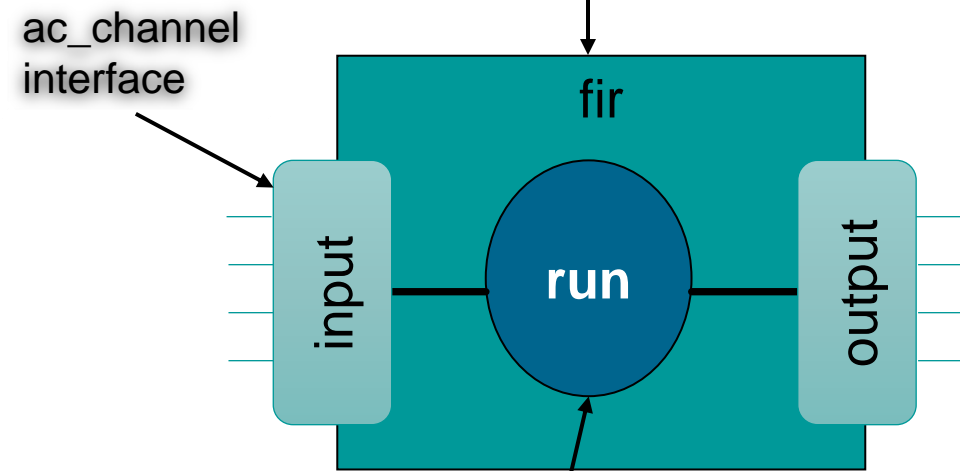
# fir.h (HLS C)

```
#include <ac_int.h>
#include <ac_channel.h>
#include <mc_scverify.h>
```

```
class fir {
private:
    int8 regs[8];
public:
    fir() {
        for (int i = 7; i >= 0; i--) {
            regs[i] = 0; }
    }
```

Required

Design block



```
#pragma hls_design interface
void CCS_BLOCK(run)(ac_channel<int8> &input,
                    int8 coeffs[8],
                    ac_channel<int8> &output)
```

Required

label

```
{
    int19 temp = 0;
SHIFT: for (int i = 7; i >= 0; i--) {
    if ( i == 0 ) {
        regs[i] = input.read();
    } else {
        regs[i] = regs[i-1];
    }
}
MAC: for (int i = 7; i >= 0; i--) {
    temp += coeffs[i]*regs[i];
}
output.write(temp >> 11);
};
```



# fir\_tb.cpp (TB)

```
#include "fir_ref.h"
#include "fir.h"
CCS_MAIN(int argc, char *argv[])
{
    int8 coeffs[8];
    ac_channel<int8> in_chn;
    ac_channel<int8> out_chn;
    int8 din ,dout;
    int coeffs_ref[8];
    int din_ref;
    int dout_ref;
    fir      dut;
    fir_ref ref_model;
    int pass_cnt = 0;
    int fail_cnt = 0;
    // Test Impulse
    for (int i=0; i < 8; i++)
        coeffs_ref[i]=coeffs[i]=-128+rand()%256;
    ...
}
```

```
for (int i = 0; i < 100; i++ ) {
    din_ref = din = -128 + rand()%256;
    in_chn.write(din);
    dut.run(in_chn, coeffs, out_chn);
    ref_model.run(din_ref, coeffs_ref,
dout_ref);
    dout = out_chn.read();
    if (dout.to_int()!=dout_ref) {
        printf("fail @ %3d: %4d != %4d \n", i,
dout.to_int(), dout_ref);
        fail_cnt ++;
    } else {
        printf("pass @ %3d: %4d == %4d \n", i,
dout.to_int(), dout_ref);
        pass_cnt ++;
    }
}
printf("\n");
printf("total pass count %d\n", pass_cnt);
printf("total fail count %d\n", fail_cnt);
CCS_RETURN(0);
}
```

# Pre-HLS Verification by gcc (C Sim)

Build with gcc

- `g++ -o3 -std=c++11 -I $MGC_HOME/shared/include fir_tb.cpp -o SCVerify_fir.exe`

```
[mentor@RHEL74 01_walkthrough_loops]$ make
/usr/local/bin/Siemens_EDA/Catapult_Synthesis_2023.2_1-1065141/Mgc_home/bin/g++ -g -o3 -std=c++11 -Wall -Wno-unknown-pragmas -Wno-unused-variable -Wno-unused-label -I./ -I/usr/local/bin/Siemens_EDA/Catapult_Synt
thesis_2023.2_1-1065141/Mgc_home/shared/include ./fir_tb.cpp -o bin/SCVerify_fir.exe
[mentor@RHEL74 01_walkthrough_loops]$
```

Run execution file

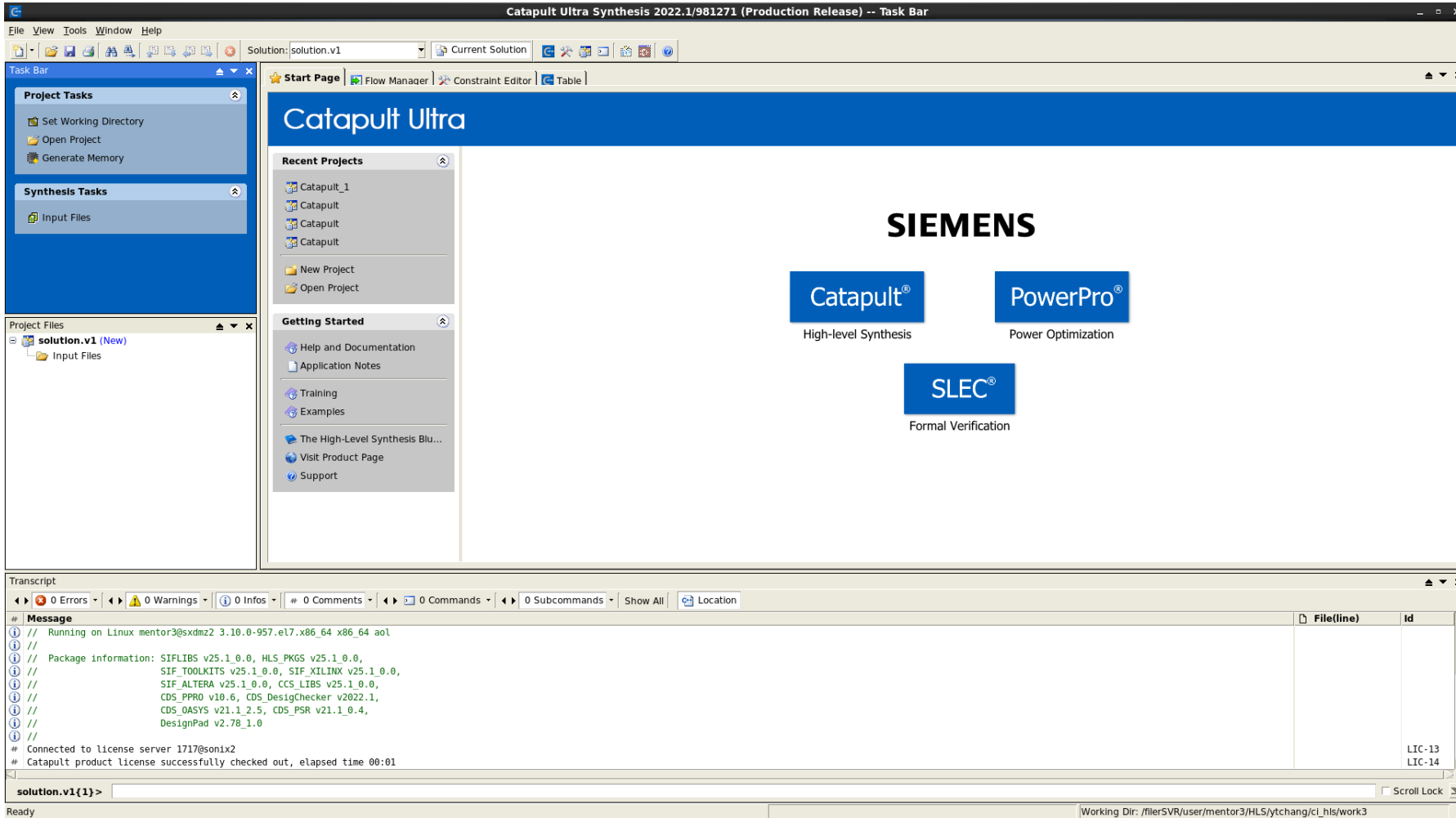
- `./SCVerify_fir.exe`

```
pass @ 89: 3 == 3
pass @ 90: -18 == -18
pass @ 91: 21 == 21
pass @ 92: -12 == -12
pass @ 93: 2 == 2
pass @ 94: -6 == -6
pass @ 95: 5 == 5
pass @ 96: -21 == -21
pass @ 97: 18 == 18
pass @ 98: -5 == -5
pass @ 99: 9 == 9
```

```
total pass count 100
total fail count 0
```

# Launch Catapult

catapult &



# Add input file

Project Tasks

Set Working Directory

Open Project

Generate Memory

Synthesis Tasks

Input Files

Project Files

solution.v1 (New)

Input Files

Catapult Ultra

Recent Projects

Catapult\_1

Catapult

Catapult

Catapult

New Project

Open Project

Getting Started

Help and Documentation

Application Notes

Training

Examples

The High-Level Synthesis Blu...

Visit Product Page

Support

Transcript

0 Errors

0 Warnings

0 Infos

0 Comments

0 Commands

0 Subcommands

Show All

Location

# Message

// Running on Linux mentor3@sxdmz2 3.10.0-957.el7.x86\_64 x86\_64 aol

//

// Package information: SIFLIBS v25.1.0.0, HLS PKGS v25.1.0.0,

// SIF\_TOOLKITS v25.1.0.0, SIF\_XILINX v25.1.0.0,

// SIF\_ALTERA v25.1.0.0, CCS\_LIBS v25.1.0.0,

// CDS\_PPRO v10.6, CDS\_DesignChecker v2022.1,

// CDS\_OASYS v21.1.2.5, CDS\_PSR v21.1.0.4,

// DesignPad v2.78.1.0

//

// Connected to license server 1717@sonix2

// Catapult product license successfully checked out, elapsed time 00:01

solution.v1(1)>

Ready

Add Input Files

Directory: /home/mentor/sda4/Project/Altek/training/walkthrough

Name	Exclu...	Size	Type	Modified
..			Folder	21/08/22 08:19...
fir.h	<input type="checkbox"/>	1KB	H File	22/08/22 09:02...
fir_ref.h	<input type="checkbox"/>	1KB	H File	22/08/22 09:00...
fir_tb.cpp	<input checked="" type="checkbox"/>	2KB	CPP File	22/08/22 09:02...

File name: fir.h fir\_tb.cpp

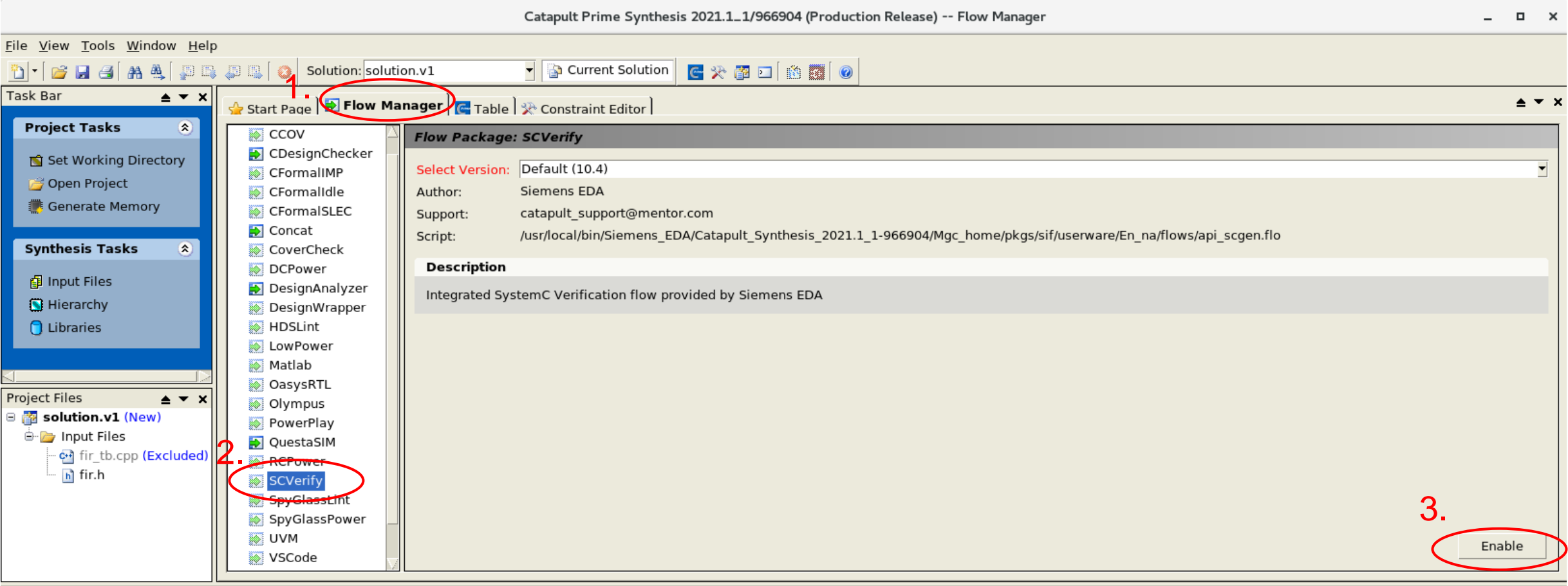
Files of type : {C++ Files} {.c .C .cpp .cxx .cc .h .hh}

Open

Cancel

TB is not required for synthesis

# Enable SCVerify



# Pre-HLS (C++) Verification by gcc

1. **Synthesis Tasks** - Libraries

2. **Project Files** - gcc 4.9.2 - Original Design + Testbench

3. **Transcript**

```
# Message
# pass @ 93: 2 == 2
# pass @ 94: -6 == -6
# pass @ 95: 5 == 5
# pass @ 96: -21 == -21
# pass @ 97: 18 == 18
# pass @ 98: -5 == -5
# pass @ 99: 9 == 9
#
# total pass count 100
# total fail count 0
```

Technology

RTL Synthesis Tool: DesignCompiler  
Vendor: Nangate  
Technology: 045nm

Compatible Libraries

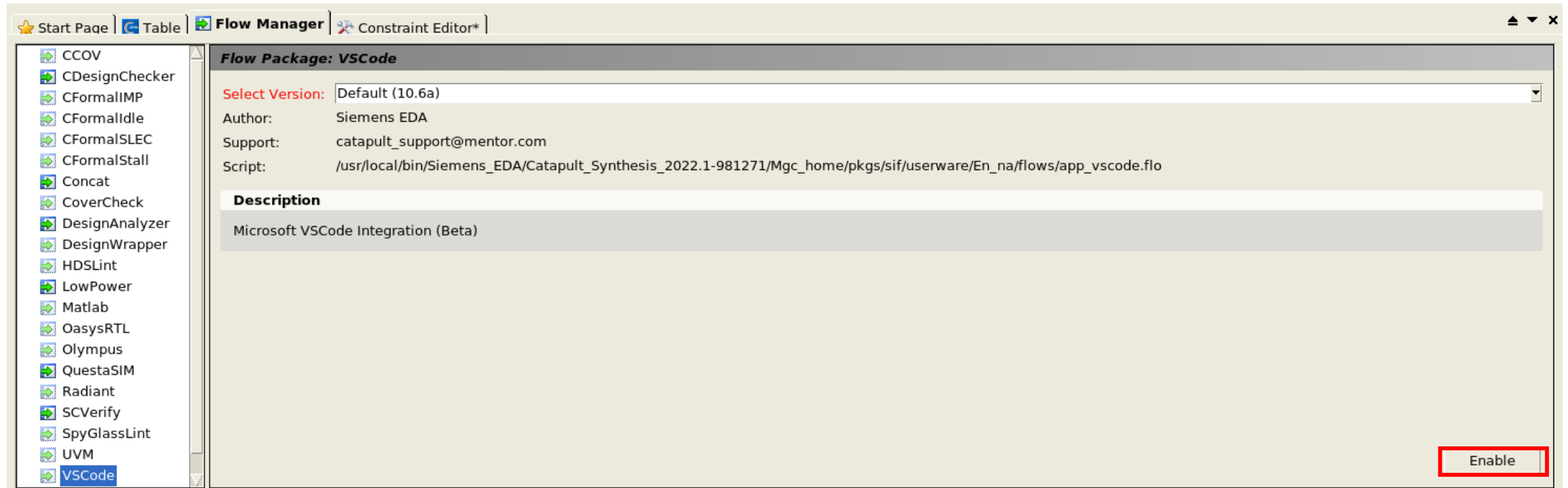
- ☒ Catapult base lib for NangateOpenCellLibrary t
- ☐ RAM - Pipe
- ☐ RAM - Sync w/Dual RW Ports
- ☐ RAM - Sync w/Separate RW Ports
- ☐ RAM - Sync w/Single RW Port
- ☐ Register File
- ☐ ROM - Asynchronous
- ☐ ROM - SynchronousRegIn
- ☐ ROM - SynchronousRegOut
- ☐ Sample RAMs

Settings Advanced

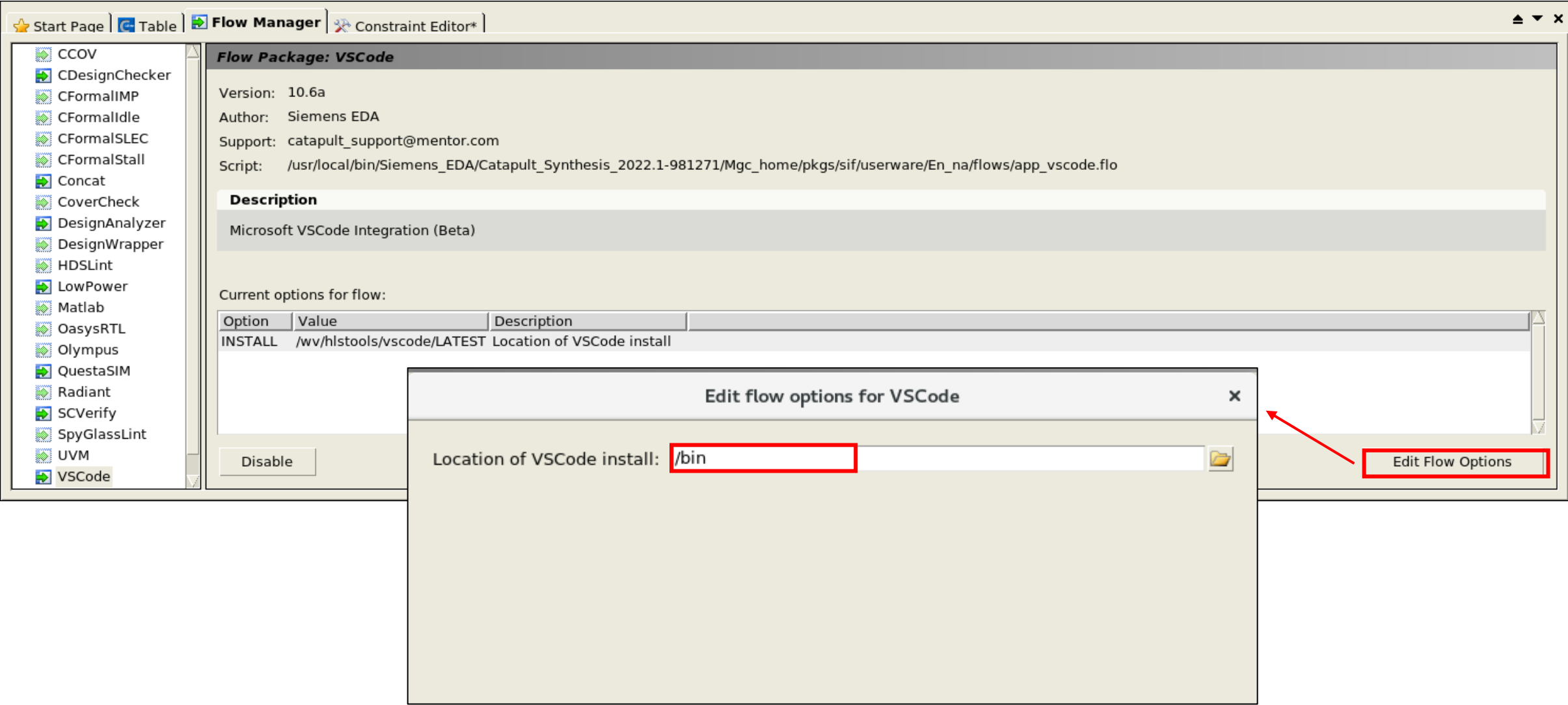
The exe can be found here

```
[mentor@RHEL74 walkthrough]$ ll ./Catapult/fir.v1/scverify/orig_cxx_osci/scverify_top
-rwxrwxr-x 1 mentor mentor 154728 Aug 22 09:29 ./Catapult/fir.v1/scverify/orig_cxx_osci/scverify_top
```

# Build VSCode Environment

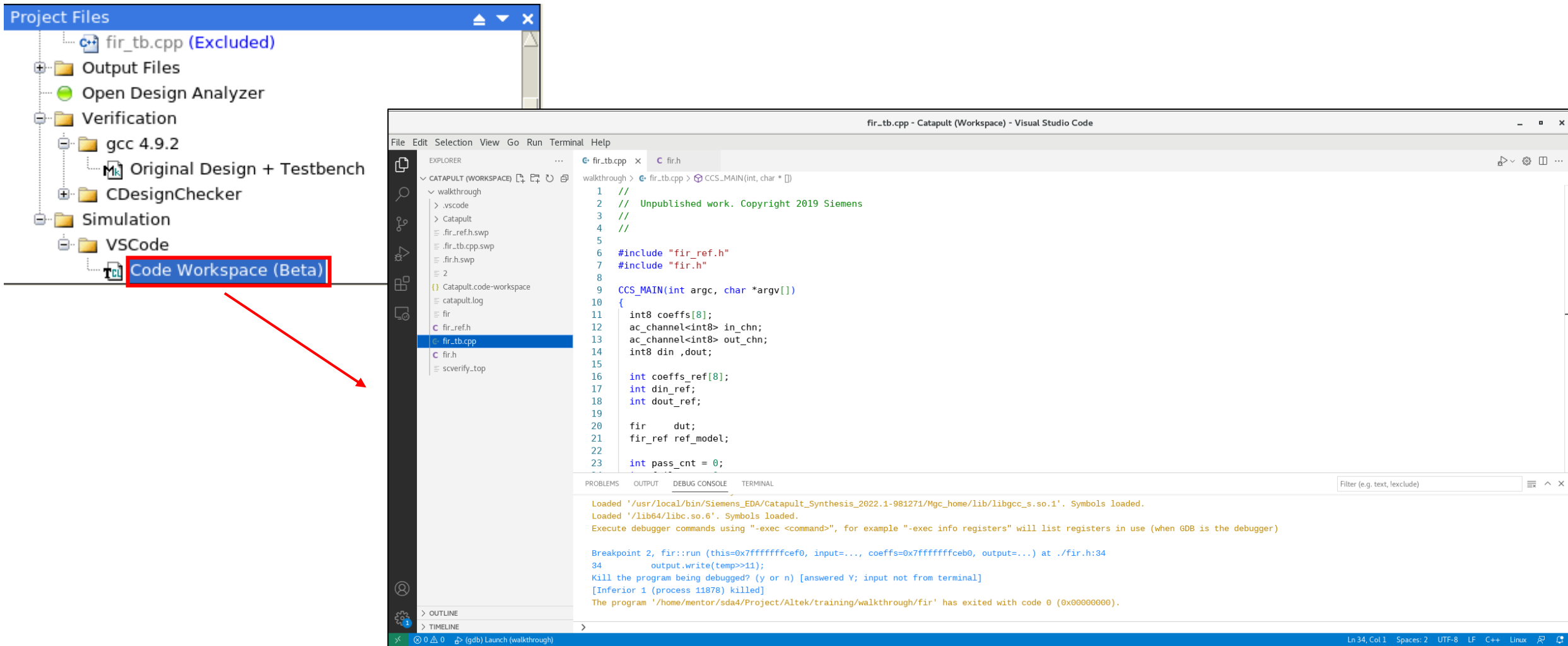


# Build VSCode Environment (Cont.)





# Build VSCode Environment (Cont.)



# Run CDesignChecker

Task Bar

Synthesis Tasks

Input Files

Hierarchy

Libraries

Project Files

fir.h

fir\_tb.cpp (Excluded)

Output Files

Open Design Analyzer

Verification

gcc 4.9.2

Original Design + Testbench

CDesignChecker

Check Design

Simulation

Transcript

0 Errors

0 Warnings

# Message

# total fail count 0

# flow package require /VSCode

# 10.6a

# flow run /VSCode

# /OUTPUTFILES/4/DEPENDENCIES/1/DEPENDENCY

# flow package option set /VSCode/INSTALL

# flow run /VSCode/launch\_vscode ../../Cat

# /bin/code /home/mentor/sda4/Project/...

Start Page

Table

Flow Manager

Constraint Editor\*

CCOV

CDesignChecker

CFormalIMP

CFormalIdle

CFormalSLEC

CFormalStall

Concat

CoverCheck

DesignAnalyzer

DesignWrapper

HDSLint

LowPower

Matlab

OasysRTL

Olympus

QuestaSIM

Radiant

SCVerify

SpyGlassLint

Flow Package: VSCode

Version: 10.6a

Author: Siemens EDA

Support: catapult\_support@mentor.com

Script: /usr/local/bin/Siemens\_EDA/Catapult\_Synthe

Description

Microsoft VSCode Integration (Beta)

Current options for flow:

Option	Value	Description
INSTALL	/bin	Location of VSCode install

Select Modes

Verification Mode: Custom

QofR Checks

SCVerify/ Synthesis-Simulation Mism

Check for Overflow/Underflow

Custom

OK

Cancel

Select Rules

<input checked="" type="checkbox"/> ABR - Array Bounds Read	ERROR
<input checked="" type="checkbox"/> ABW - Array Bounds Write	ERROR
<input checked="" type="checkbox"/> ACC - Accumulator of native C type	WARNING
<input checked="" type="checkbox"/> ACS - Accumulator of saturated type	WARNING
<input checked="" type="checkbox"/> AIC - Assignment used Instead of Comparison	WARNING
<input checked="" type="checkbox"/> ALS - Ac_int Left Shift check	WARNING
<input checked="" type="checkbox"/> AOB - Arithmetic Operator with Boolean	ERROR
<input checked="" type="checkbox"/> APT - Array Dimension Power of Two	INFO
<input checked="" type="checkbox"/> AWE - Assignments Without Effect	WARNING
<input checked="" type="checkbox"/> CAS - Incomplete Switch-Case	ERROR
<input checked="" type="checkbox"/> CBU - Conditional break in Unrolled Loop	WARNING
<input checked="" type="checkbox"/> CCC - Static constant comparison	WARNING
<input checked="" type="checkbox"/> CGR - Conditional Guard in Rolled Loop	WARNING
<input checked="" type="checkbox"/> CIA - Comparison Instead of Assignment	WARNING
<input checked="" type="checkbox"/> CMC - C style Memory Check	INFO
<input checked="" type="checkbox"/> CNS - Constant condition of if/switch	WARNING
<input checked="" type="checkbox"/> CWB - Case Without Break	WARNING
<input checked="" type="checkbox"/> DBZ - Divide By Zero	ERROR
<input checked="" type="checkbox"/> DIU - Dynamic Index in Unrolled Loop	WARNING
<input checked="" type="checkbox"/> FVI - For Loop with Variable Iterations	WARNING
<input checked="" type="checkbox"/> FXD - Mixed fixed and non-fixed datatypes	WARNING
<input checked="" type="checkbox"/> ISE - Illegal Shift Error	ERROR
<input checked="" type="checkbox"/> LRC - Long Reset Cycle	INFO
<input checked="" type="checkbox"/> MDB - Missing Default Branch	WARNING
<input checked="" type="checkbox"/> MXS - Mixed signed and unsigned datatypes	INFO
<input checked="" type="checkbox"/> NCO - No Contribution to Output	WARNING
<input checked="" type="checkbox"/> OSA - Optimal Size Accumulator	WARNING
<input checked="" type="checkbox"/> OVL - Overflow/Underflow	ERROR
<input checked="" type="checkbox"/> PDD - Platform dependent datatype (long)	WARNING
<input checked="" type="checkbox"/> RIU - Rolled loop Inside Unrolled loop	WARNING
<input checked="" type="checkbox"/> RRT - Reset referenced in thread	ERROR

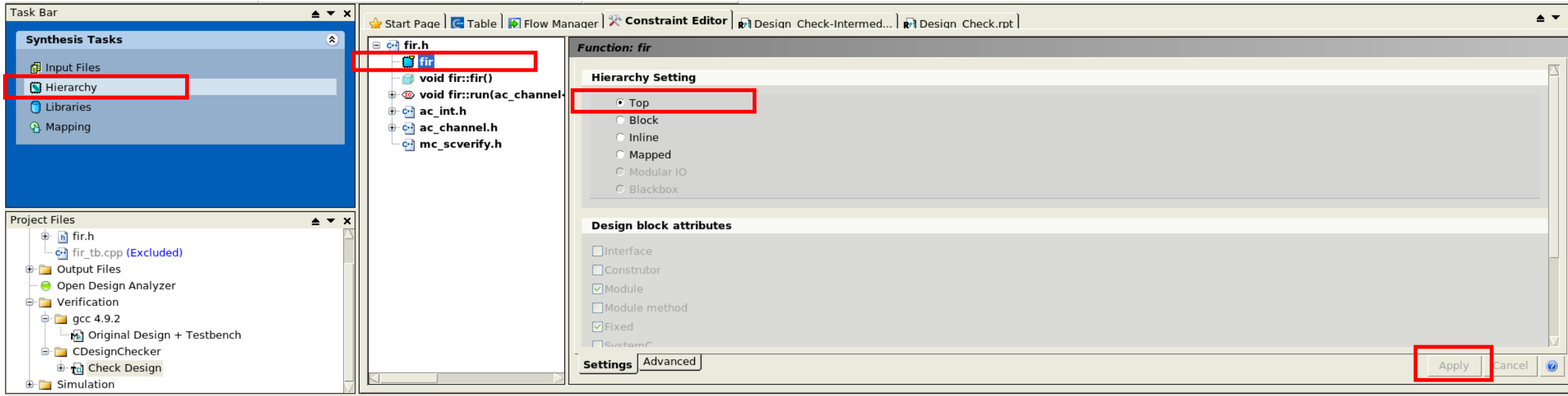
OK

Cancel





# HLS Synthesis – Set Top Design



# HLS Synthesis – Library Setup

Catapult Prime Synthesis 2021.1\_1/966904 (Production Release) -- Constraint Editor\*

File View Tools Window Help

Solution: fir.v1 Current Solution

Task Bar

Synthesis Tasks

- Input Files
- Hierarchy
- Libraries

Project Files

- fir.v1 (Passed Compile)
  - Input Files
    - fir.h
    - fir\_tb.cpp (Excluded)
  - Output Files
  - Open Design Analyzer
  - Verification

Technology

RTL Synthesis Tool: OasysRTL

Vendor: Nangate

Technology: 045nm

Search Path

Memory Generator

Compatible Libraries

- ☒ Catapult base lib for NangateOpenCellLibrary typical
- ☐ RAM - Pipe
- ☐ RAM - Sync w/Dual RW Ports
- ☐ RAM - Sync w/Separate RW Ports
- ☐ RAM - Sync w/Single RW Port
- ☐ Register File
- ☐ ROM - Asynchronous
- ☐ ROM - SynchronousRegIn
- ☐ ROM - SynchronousRegOut
- ☐ Sample RAMs
- ☐ AMBA Interface Synthesis Library
- ☐ AMBA Interface Synthesis Library (Support ML T

Details - Catapult base lib for NangateOpenCellLibrary typical

Name: nangate-45nm\_beh

Type: Technology

Path: \$MGC\_HOME/pkgs/siflibs/nangate/nangate-45nm\_beh.lib

Description

This Catapult library provides the HLS synthesis info for the Open Source Nangate 45nm technology. In order to simplify the downstream flows, the Liberty, LEF, and Verilog files are also included (\$MGC\_HOME/pkgs/siflibs/nangate). The Oasys RTL flow will work out-of-the-box. For Design Compiler, you must compile the Liberty file and name the resulting file 'NangateOpenCellLibrary\_typical\_ccs.db'. In Catapult, add the path to the DB file to the TechLibSearchPath option (Tools > Set Options.. > Component Libs)

Settings Advanced

Apply Cancel

Transcript

0 Errors 0 Warnings 7 Infos 19 Comments 4 Commands 0 Subcommands Show All Location

# HLS Synthesis – Mapping

## Frequency

Catapult Prime Synthesis 2021.1...1/966904 (Production Release) -- Constraint Editor\*

File View Tools Window Help

Solution: fir.v1 Current Solution

Task Bar

Start Page | Flow Manager | Table | Constraint Editor\*

**Synthesis Tasks**

- Input Files
- Hierarchy
- Libraries
- Mapping**

**Project Files**

- fir.v1 (Passed Libraries)**
  - Input Files
    - fir\_tb.cpp (Excluded)
    - fir.h
  - Output Files
  - Open Design Analyzer
  - Verification

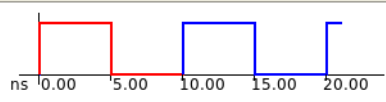
**Instance Hierarchy**

- Solution
  - fir

**Module**

- fir
  - run (clk)**

**Process: run**



Process Clock: clk

Add Clock X Delete Rename

Frequency: 100 MHz

Period: 10 ns

High-Time: 5 ns

Offset: 0 ns

Edge: rising

Uncertainty: 0 ns

Settings Advanced

Apply Cancel

# HLS Synthesis – Mapping

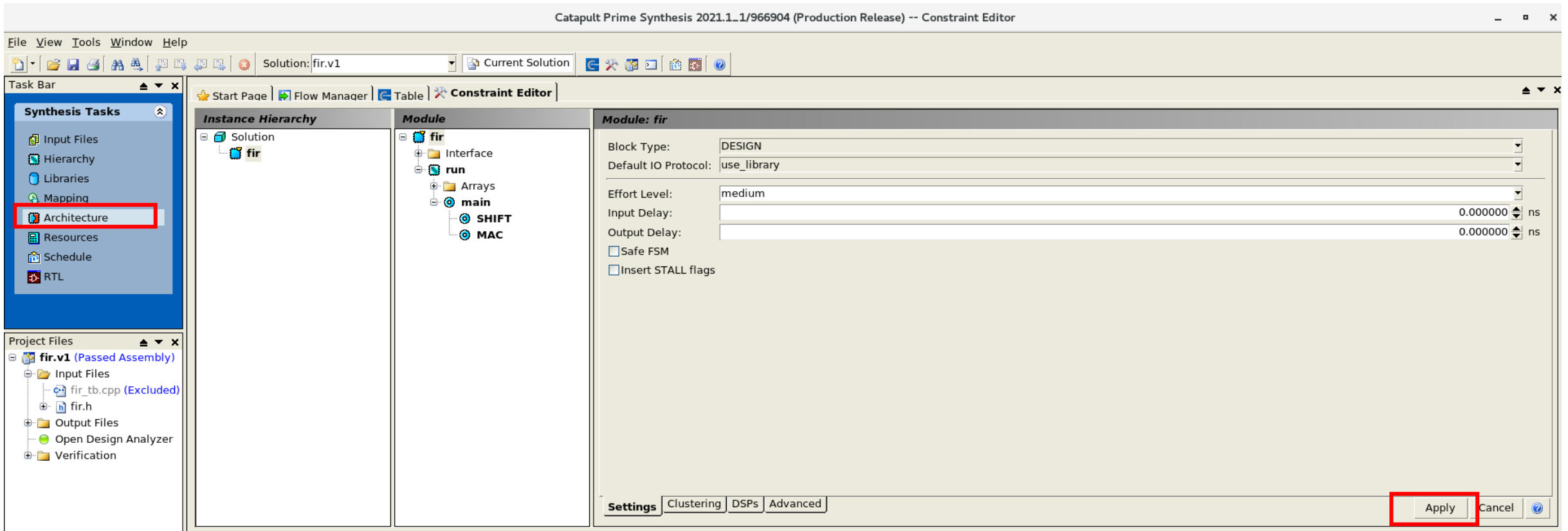
Sync / async reset

The screenshot displays the Siemens EDA software interface during the HLS Synthesis Mapping phase. The **Synthesis Tasks** pane on the left shows the **Mapping** tab selected. Below it, the **Project Files** pane shows the project structure for **fir.v1 (Passed Extract)**, including **Input Files** (fir.h, fir\_tb.cpp (Excluded)), **Output Files**, **Open Design Analyzer**, **Verification**, **Simulation**, and **Synthesis**.

The main workspace is divided into three panes:

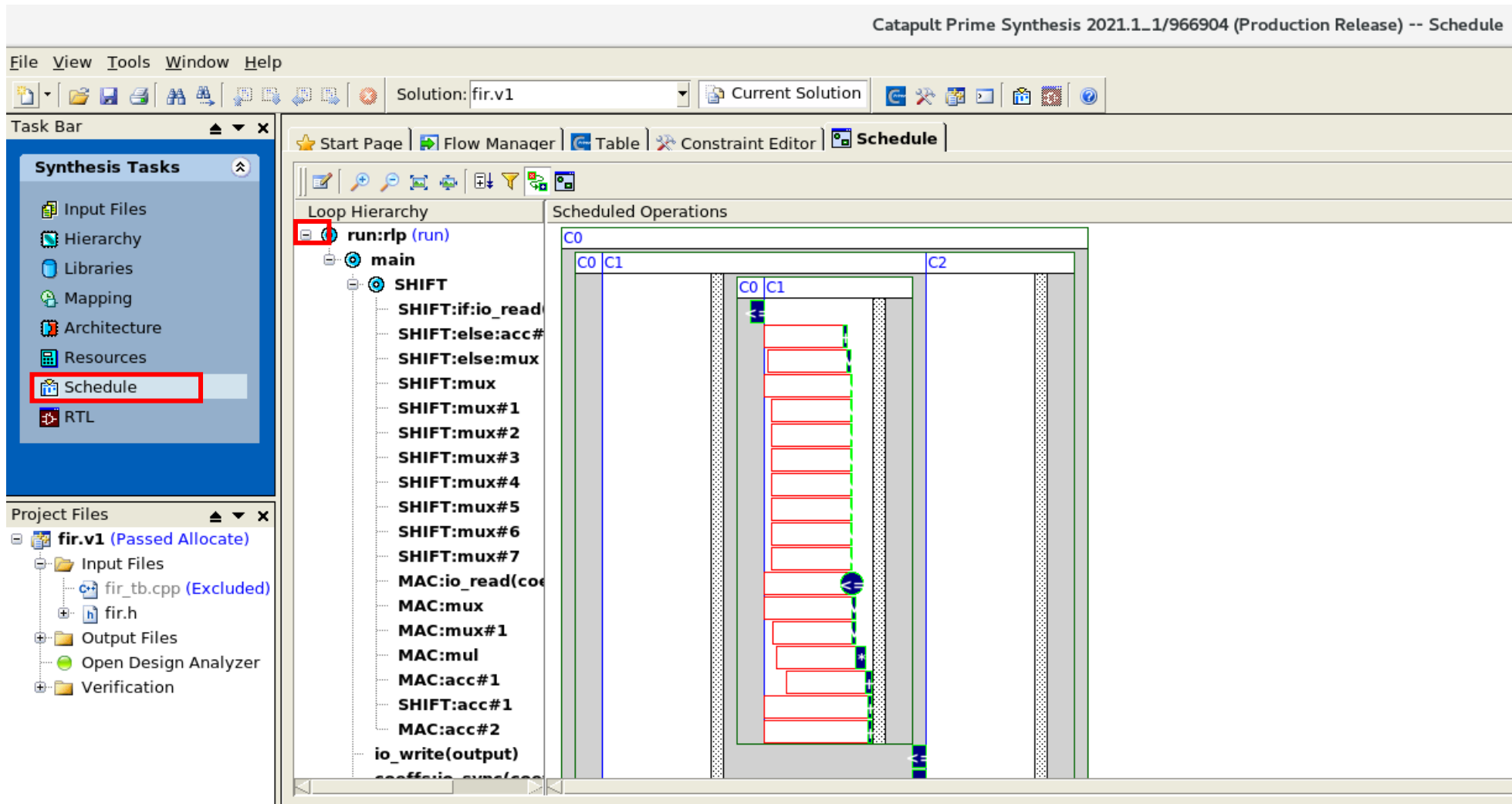
- Instance Hierarchy:** Shows a **Solution** containing the **fir** module.
- Module:** Shows the **fir** module with a **run (clk)** instance.
- Process: run:** Contains the reset configuration settings.
  - Synchronous Reset:** This option is selected (checked). The **Signal Name** is **rst** and the **Active** state is **high**.
  - Asynchronous Reset:** This option is not selected (unchecked). The **Signal Name** is **arst\_n** and the **Active** state is **low**.
  - Enable:** This option is not selected (unchecked). The **Signal Name** is **en** and the **Active** state is **high**.
  - Settings:** The **Advanced** button is highlighted.

# HLS Synthesis – Architecture





# HLS Synthesis – Scheduling



# HLS Synthesis – RTL

The screenshot shows the Siemens EDA software interface during RTL synthesis. The **Task Bar** on the left lists various tasks, with **RTL** highlighted. The **Project Files** pane shows a project structure with folders like Output Files, Reports, Schedule, Schematics, VHDL, SCVerify, and Verilog. The file **rtl.v** is selected under the Verilog folder. The main editor displays the Verilog code for the **fir** module, which is highlighted with a red box. The code defines the module with inputs **clk**, **rst**, **input\_rsc\_dat**, **input\_rsc\_vld**, **input\_rsc\_rdy**, **coeffs\_rsc\_dat**, and **coeffs\_rsc\_triosy\_lz**, and outputs **output\_rsc\_dat**, **output\_rsc\_vld**, and **output\_rsc\_rdy**. It also includes interconnect declarations for component instantiations.

```
750 module fir (  
751     clk, rst, input_rsc_dat, input_rsc_vld, input_rsc_rdy, coeffs_rsc_dat, coeffs_rsc_triosy_lz,  
752     output_rsc_dat, output_rsc_vld, output_rsc_rdy  
753 );  
754  
755 input clk;  
756 input rst;  
757 input [7:0] input_rsc_dat;  
758 input input_rsc_vld;  
759 output input_rsc_rdy;  
760 input [63:0] coeffs_rsc_dat;  
761 output coeffs_rsc_triosy_lz;  
762 output [7:0] output_rsc_dat;  
763 output output_rsc_vld;  
764 input output_rsc_rdy;  
765  
766  
767 // Interconnect Declarations for Component Instantiations  
768 fir_run fir_run_inst (  
769     .clk(clk),  
770     .rst(rst),  
771     .input_rsc_dat(input_rsc_dat),  
772     .input_rsc_vld(input_rsc_vld),  
773     .input_rsc_rdy(input_rsc_rdy),  
774     .coeffs_rsc_dat(coeffs_rsc_dat),  
775     .coeffs_rsc_triosy_lz(coeffs_rsc_triosy_lz),  
776
```

# HLS Synthesis – Report

Task Bar

Libraries

Mapping

Architecture

Resources

Schedule

RTL

Power Report (Pre Power Opt)

Power RTL

Project Files

fir\_tb.cpp (Excluded)

Output Files

Reports

Schedule

Schematics

VHDL

SCVerify

Verilog

rtl.v

concat\_rtl.v (extract)

concat\_sim rtl.v (extract)

Start Page | Table | Flow Manager | Constraint Editor | Design Check-Inter... | Design Check.rpt | Schedule | rtl.v

Report: General

Solution

General

Run Time

Memory Usage

Timing

Area Score

ency...	Latency...	Throug...	Throug...	Slack	Total Area
8	80.00	10	100.00	6.49	1422.44

# HLS Synthesis – Report (Cont.)

Catapult Prime Synthesis 2021.1\_1/966904 (Production Release) -- rtl.rpt

File Edit View Tools Window Help

Solution: fir.v1 Current Solution

Task Bar

Synthesis Tasks

- Input Files
- Hierarchy
- Libraries
- Mapping
- Architecture
- Resources
- Schedule
- RTL

Project Files

- fir\_tb.cpp (Exclude)
- fir.h
- Output Files
- Reports
- Messages
- Commands
- Messages
- Cycle
- RTL
- Schedule
- Schematics

Start Page | Flow Manager | Table | Constraint Editor | Schedule | rtl.rpt

Goto line...

```
1  |-- Catapult Prime Synthesis: Report
2
3  |-- Version:                2021.1_1/966904 Production Release
4  |-- Build Date:             Thu Nov 11 21:13:39 PST 2021
5
6  |-- Generated by:           mentor@RHEL74
7  |-- Generated date:         Tue Mar 01 17:04:01 CST 2022
8
9  Solution Settings: fir.v1
10 Current state: extract
11 Project: Catapult
12
13 Design Input Files Specified
14 $PROJECT_HOME/fir_tb.cpp
15 $PROJECT_HOME/fir.h
16 $MGC_HOME/shared/include/ac_int.h
17 $MGC_HOME/shared/include/ac_channel.h
18 $MGC_HOME/shared/include/mc_scverify.h
19
20 Processes/Blocks in Design
21 Process      Real Operation(s) count Latency Throughput Reset Length II Comments
22 -----
23 /fir/run      38      8      10      0 0
24 Design Total: 38      8      10      0 0
25
26 Bill Of Materials (Datapath)
27 Component Name      Area Score Area(Combinational) AreaSeq Delay Post Alloc Post Assign
28 -----
29 [Lib: ccs_ioport]
30 ccs_in(2,64)         0.000      0.000 0.000 0.000      1      1
31 ccs_in_wait(1,8)     0.000      0.000 0.000 0.000      1      1
32 ccs_out_wait(3,8)    0.000      0.000 0.000 0.000      1      1
```

# Design Analyzer

Analyze the design in various perspectives

The screenshot displays the Catapult Design Analyzer interface, which is used for analyzing digital designs. The main window is titled "/home/mentor/sda4/Project/Altek/training/walkthrough/Catapult - Catapult Design Analyzer".

**Project Files:** A sidebar on the left shows the project structure, including folders for SCVerify, Verilog, and Verification. The "Open Design Analyzer" option is highlighted in the Verification folder.

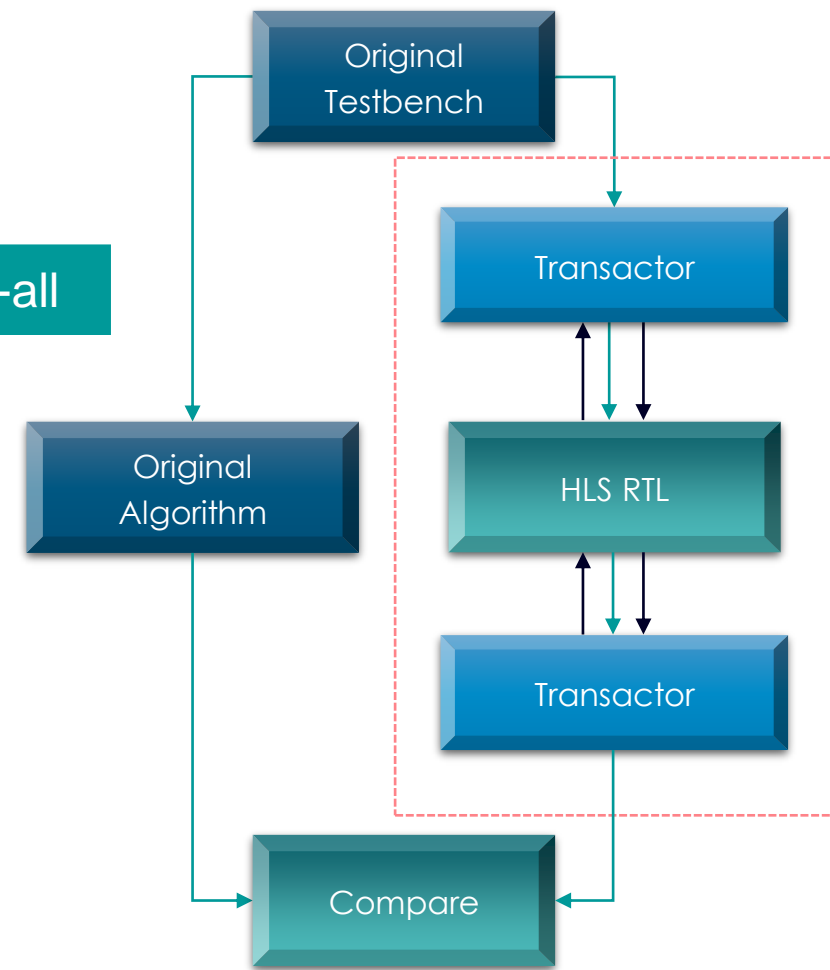
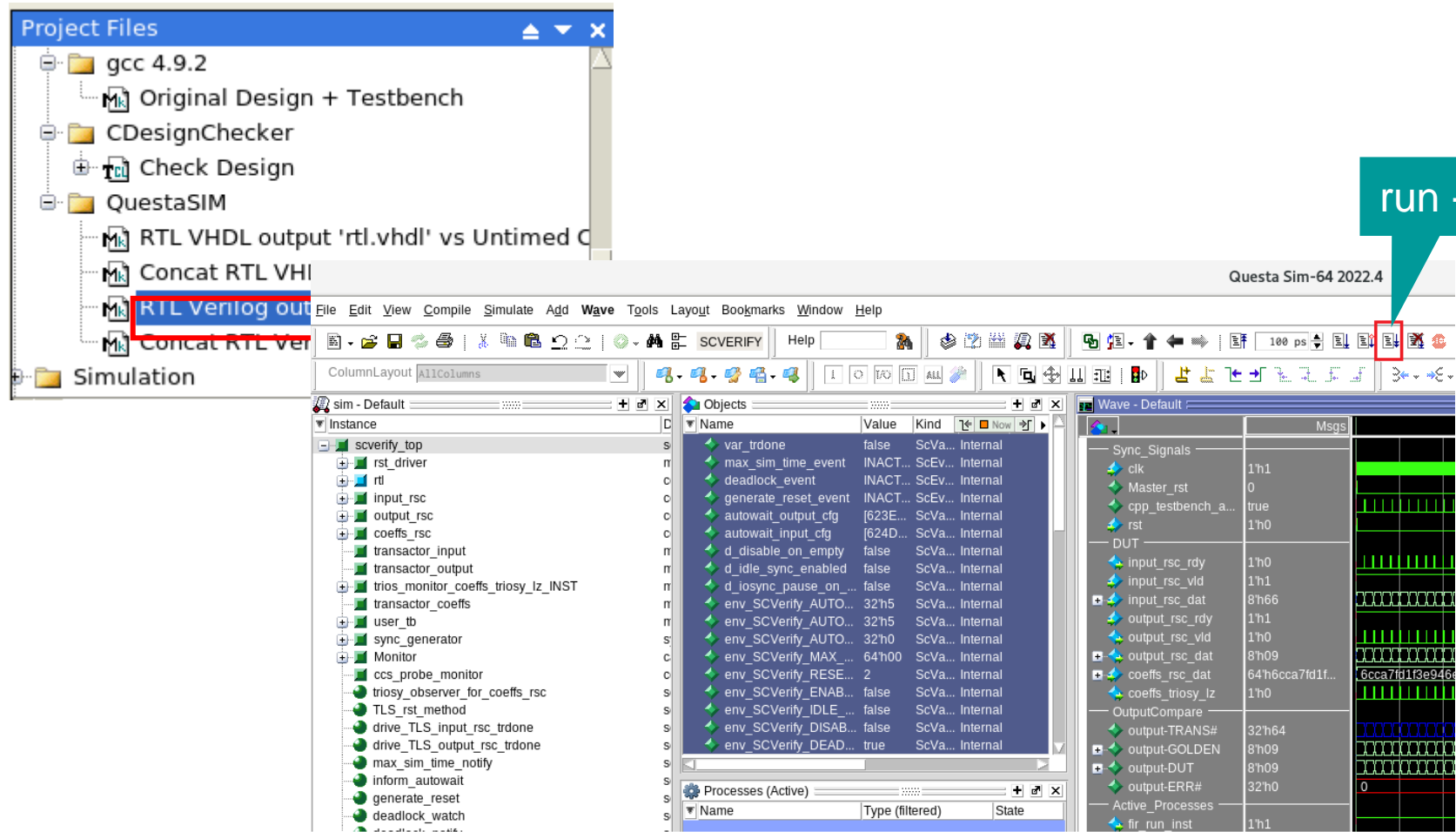
**Perspectives:** A menu in the top-left corner lists various analysis perspectives, including Schematic, Schedule, and Schematic. The "Schematic" perspective is currently selected, showing a detailed circuit diagram of a FIR filter. A red box highlights a specific part of the circuit, and the text "1 'X', 1 '+'" is overlaid on it.

**Code Views:** The bottom of the interface shows the source code for the design. The "fir.h (1)" file is open, displaying C code for a FIR filter. The code includes a loop for calculating the output, with a red box highlighting the calculation of the output value.

**Adviser:** A panel on the right side of the interface, titled "Adviser", shows the results of the analysis. It indicates that no issues were detected and provides reference information for the analysis.

# RTL Verification

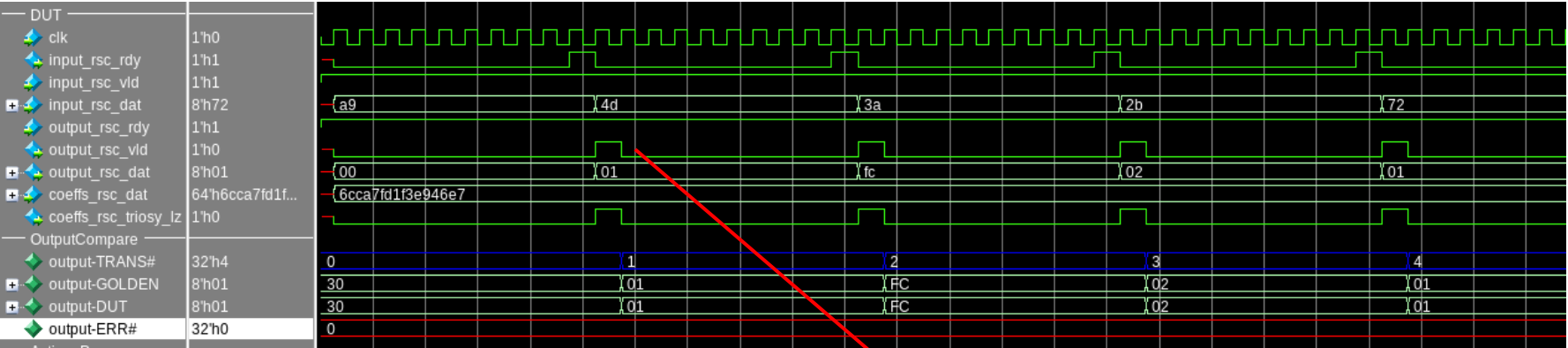
Automatically compare the outputs of HLC C and RTL



# RTL Verification

Automatically compare the outputs of HLC C and RTL

```
Transcript
# captured 100 values of output
# Info: scverify_top/user_tb: Simulation completed
#
# Checking results
# 'output'
#   capture count      = 100
#   comparison count   = 100
#   ignore count       = 0
#   error count        = 0
#   stuck in dut fifo  = 0
#   stuck in golden fifo = 0
#
# Info: scverify_top/user_tb: Simulation PASSED @ 10016 ns
# ** Note: (vsim-6574) SystemC simulation stopped by user.
# 1
#
VSIM 3>
```

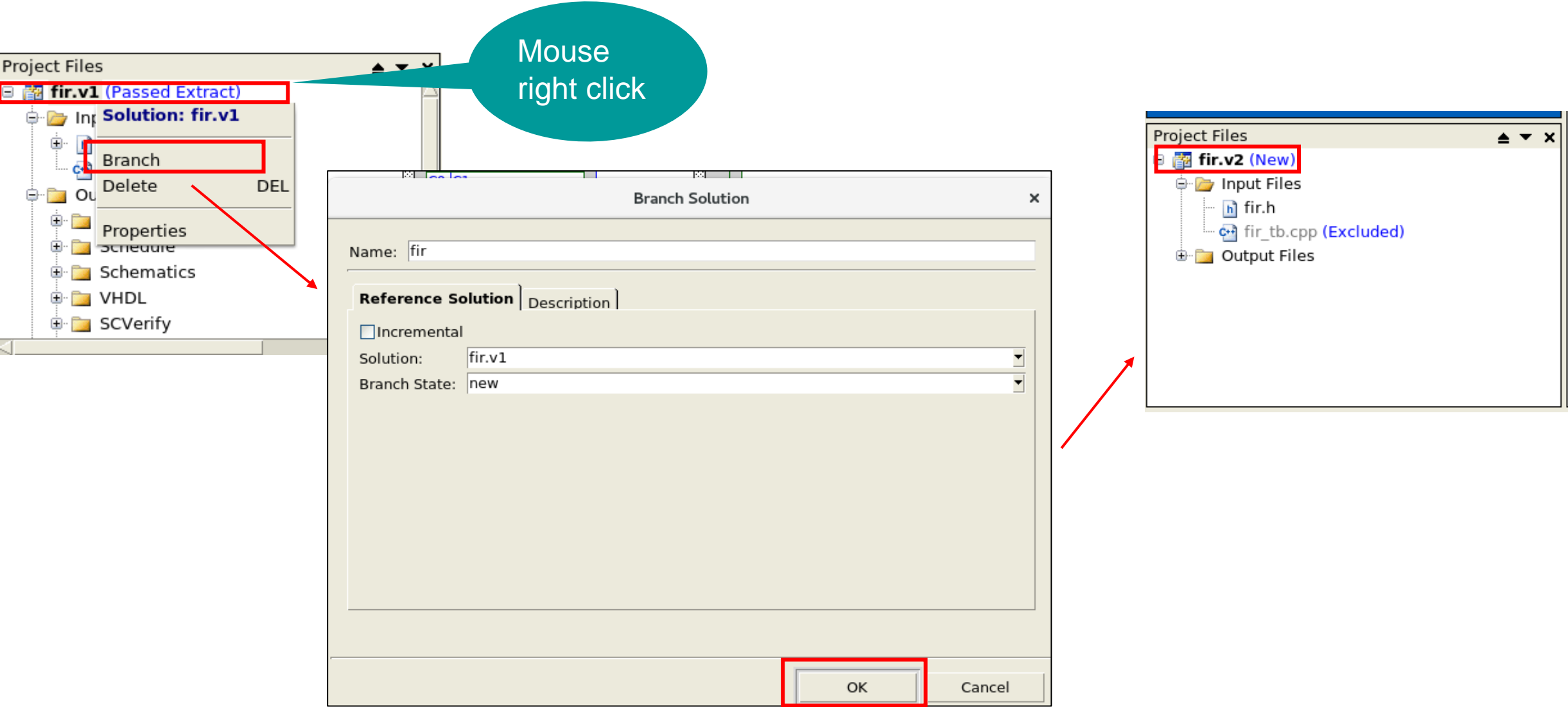


Throughput = 10

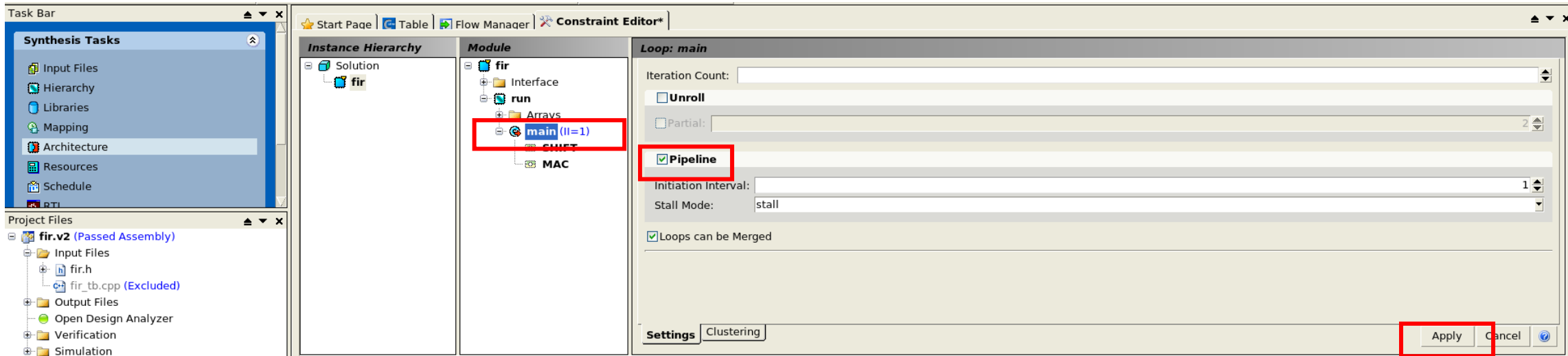
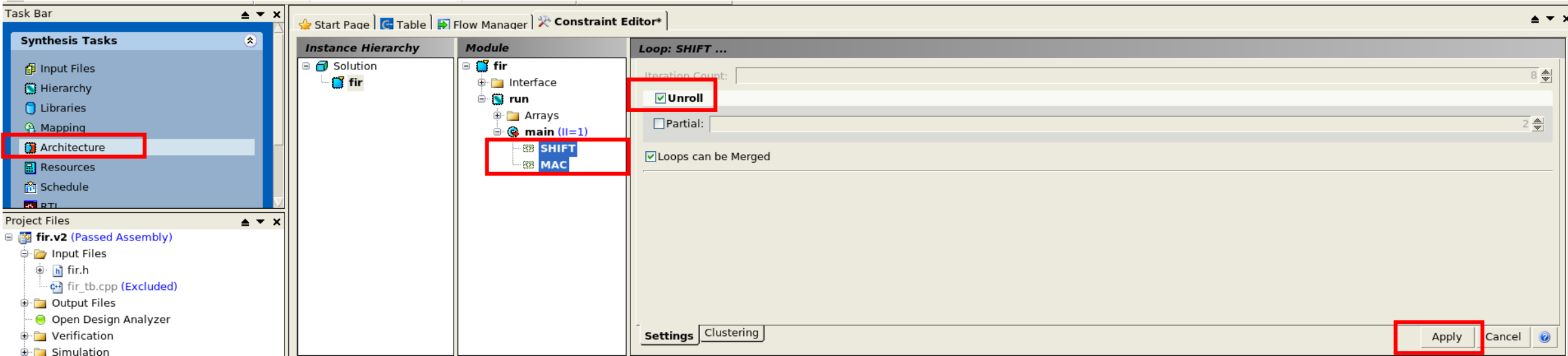
# | Loop Unrolling and Pipelining



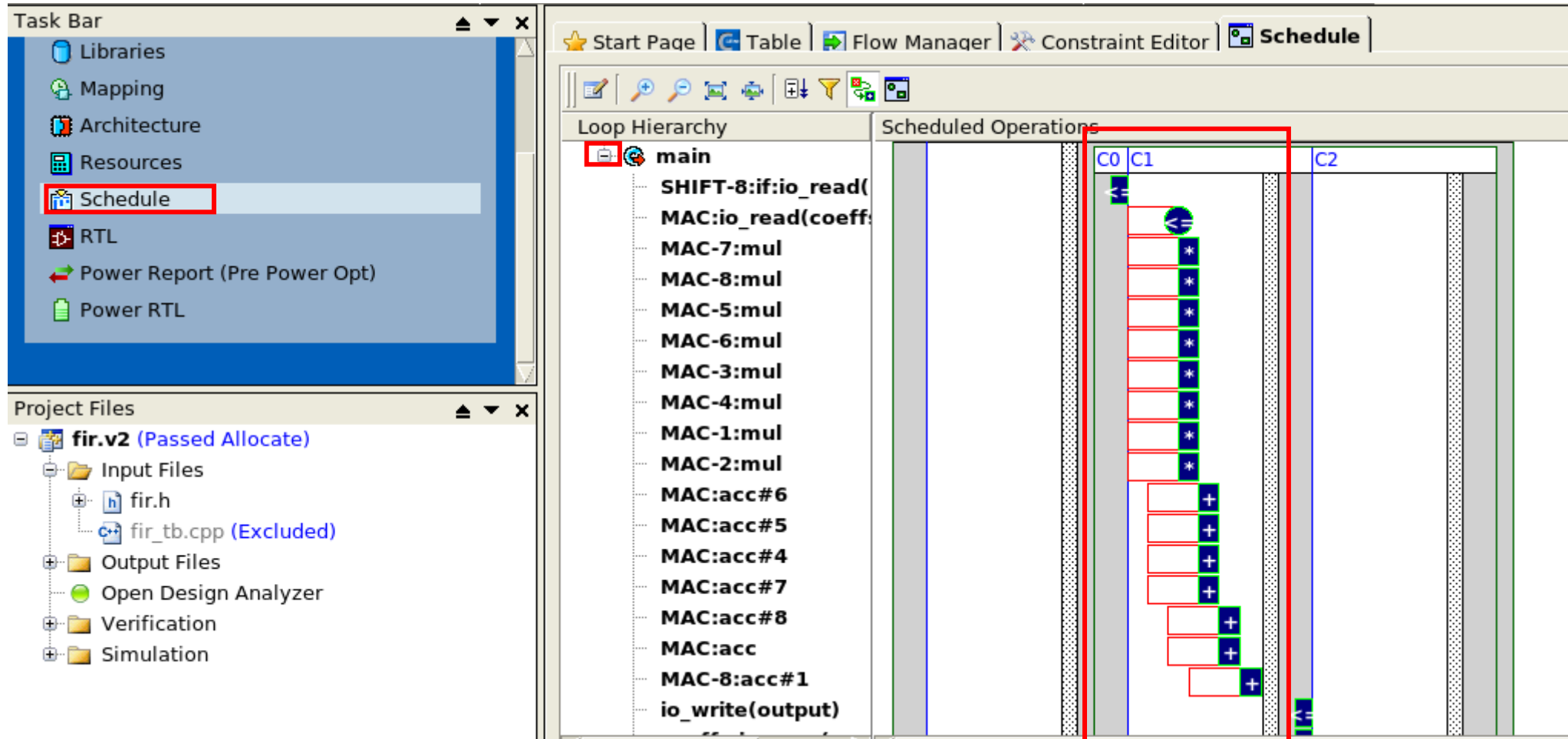
# Branch a New Solution



# HLS Synthesis – Architecture



# HLS Synthesis – Scheduling



# HLS Synthesis – RTL

The screenshot shows the Siemens EDA software interface for RTL synthesis. The **Task Bar** on the left lists various tasks, with **RTL** highlighted. The **Project Files** pane below it shows a project structure with folders for Schedule, Schematics, VHDL, SCVerify, and Verilog. Under the Verilog folder, the file **rtl.v** is highlighted. The main editor window displays the Verilog code for a module named **fir**. The code includes input and output declarations for clock, reset, data, valid, and ready signals, as well as coefficient data. A component instantiation **fir\_run\_inst** is shown at the bottom.

```
589 // Design Unit:   fir
590 // -----
591
592
593 module fir (
594     clk, rst, input_rsc_dat, input_rsc_vld, input_rsc_rdy, coeffs_rsc_dat, coeffs_rsc_triosy_lz,
595     output_rsc_dat, output_rsc_vld, output_rsc_rdy
596 );
597     input clk;
598     input rst;
599     input [7:0] input_rsc_dat;
600     input input_rsc_vld;
601     output input_rsc_rdy;
602     input [63:0] coeffs_rsc_dat;
603     output coeffs_rsc_triosy_lz;
604     output [7:0] output_rsc_dat;
605     output output_rsc_vld;
606     input output_rsc_rdy;
607
608
609
610 // Interconnect Declarations for Component Instantiations
611 fir_run fir_run_inst (
612     .clk(clk),
613     .rst(rst),
614     .input_rsc_dat(input_rsc_dat),
615     .input_rsc_vld(input_rsc_vld),
616
```

# HLS Synthesis – Report

Task Bar

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

Start Page | Table | Flow Manager | Constraint Editor | Schedule | rtl.v

Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
fir.v1 (extract)	8	80.00	10	100.00	6.49	1422.44
fir.v2 (extract)	1	10.00	1	10.00	4.82	3355.97

# Design Analyzer

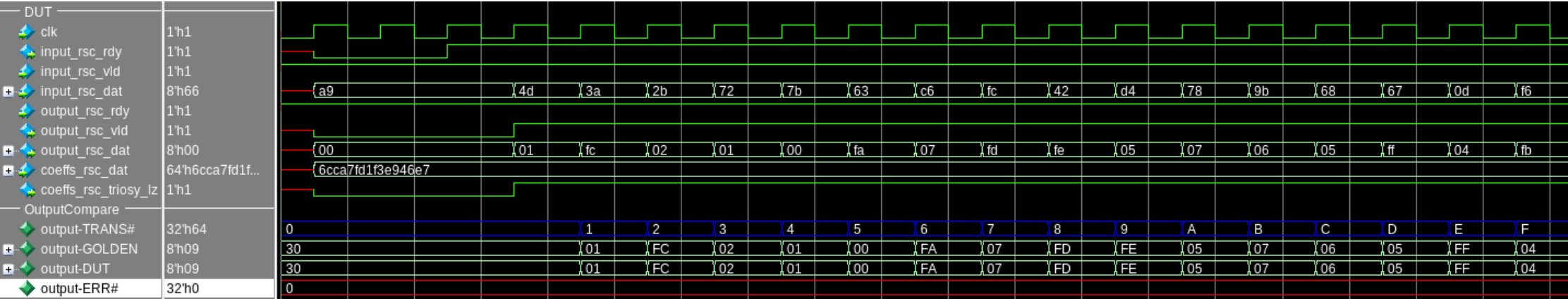
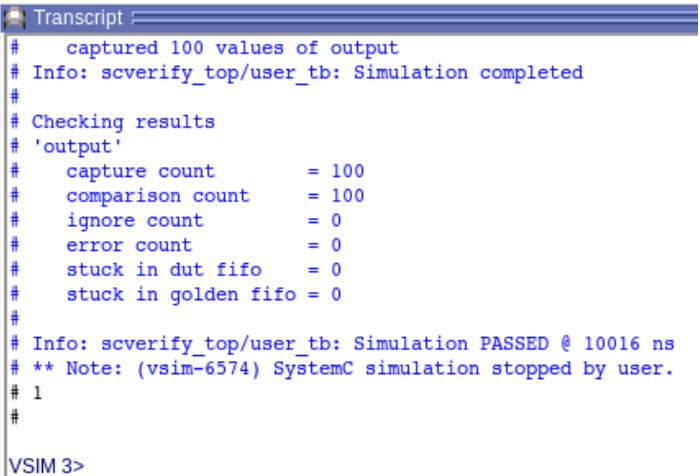
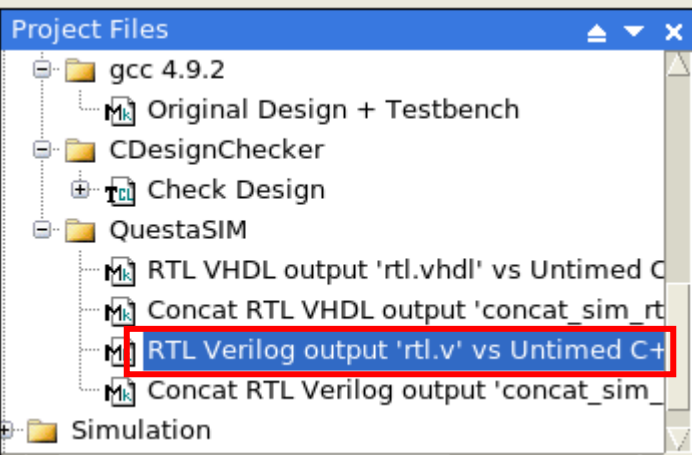
Analyze the design in various perspectives

The screenshot displays the Siemens Design Analyzer interface with several panels:

- Modules and Scopes:** Shows a tree view with 'fir' and 'run' modules. The 'run' module is selected, showing its details.
- Module Details:** Displays the 'Ports' and 'Calls' for the 'run' module. The 'Calls' section shows a call to 'SHIFT:if:io\_read(i)'. The 'Ports' section shows 'coeffs', 'input', and 'output'.
- Scope:** Shows the C code for the 'fir0' scope. The code includes a loop for 'SHIFT:for (int i = 7; i >= 0; i--)' and a call to 'func : temp >> 11 -> 0'.
- Schematic - fir:** A hardware schematic diagram showing a series of multipliers (red circles) and adders (green circles). A red box highlights a section of the circuit, and the text '8 'x', 7 '+'' is written in red next to it.
- Schedule - run:rlp:** A resource usage diagram showing the allocation of resources (multipliers and adders) across different stages of the design.
- Input Files:** A list of files including 'fir.h (3)' and 'fir\_tb.cpp'.
- Adviser:** A panel showing 'Detected Issues' (no issues detected) and 'Reference Information' (Conditional Assignment, Dynamic Write Index, Nested Conditions, Resource Competition).

# RTL Verification

Automatically compare the outputs of HLC C and RTL



# Reproduce the Solution

All command has been recorded in a tcl script

```
[mentor@RHEL74 walkthrough]$ ll ./Catapult/fir.v2/directives.tcl
-rw-rw-r-- 1 mentor mentor 3531 Aug 22 11:45 ./Catapult/fir.v2/directives.tcl

go new
go compile
solution library add nangate-45nm_beh -- -rtlsyntool DesignCompiler -vendor Nangate -technology 045nm
go libraries
directive set -CLOCKS {clk {-CLOCK_PERIOD 10.0 -CLOCK_EDGE rising -CLOCK_UNCERTAINTY 0.0 -CLOCK_HIGH_TI
go assembly
directive set /fir/run/main -PIPELINE_INIT_INTERVAL 1
directive set /fir/run/SHIFT -UNROLL yes
directive set /fir/run/MAC -UNROLL yes
go architect
█
```

The same result can be produced with this .tcl

- catapult -f ./Catapult/fir.v2/directives.tcl &



# | Memory Interface

# Source Code (fir.h)

```
void CCS_BLOCK(run)(ac_channel<ac_int<8>> &input,  
                    ac_int<8> coeffs[32][8],  
                    ac_channel<ac_int<5,false>> &coeff_addr,  
                    ac_channel<ac_int<8>> &output) {  
    ac_int<19> temp = 0;  
    ac_int<5,false> addr = coeff_addr.read();  
  
    SHIFT:for (int i = 7; i>=0; i--) {  
        if ( i == 0 ) {  
            regs[i] = input.read();  
        } else {  
            regs[i] = regs[i-1];  
        }  
    }  
    MAC:for (int i = 7; i>=0; i--) {  
        temp += coeffs[addr][i]*regs[i];  
    }  
    output.write(temp>>11);  
}
```

# TestBench (fir\_tb.cpp)

```
#include "fir.h"

CCS_MAIN(int argc, char *argv[])
{
    ac_int<8> coeffs[32][8];
    ac_channel<ac_int<8>> input;
    ac_channel<ac_int<5,false>> coeff_addr;
    ac_channel<ac_int<8>> output;
    ac_int<8> data;
    ac_int<5,false> addr;
    fir inst;
    ...
}
```

```
...
// Test Impulse
for (int j=0; j < 32; j++)
    for (int i=0; i < 8; i++)
        coeffs[j][i] = rand();
for (int i = 0; i < 10; i++ ) {
    data = rand();
    input.write(data);
    addr = rand();
    coeff_addr.write(addr);
    inst.run(input, coeffs, coeff_addr, output);
}
while (output.available(1))
    printf("Output = %3d\n",output.read().to_int());
CCS_RETURN(0);
}
```

# Start with running directives.tcl

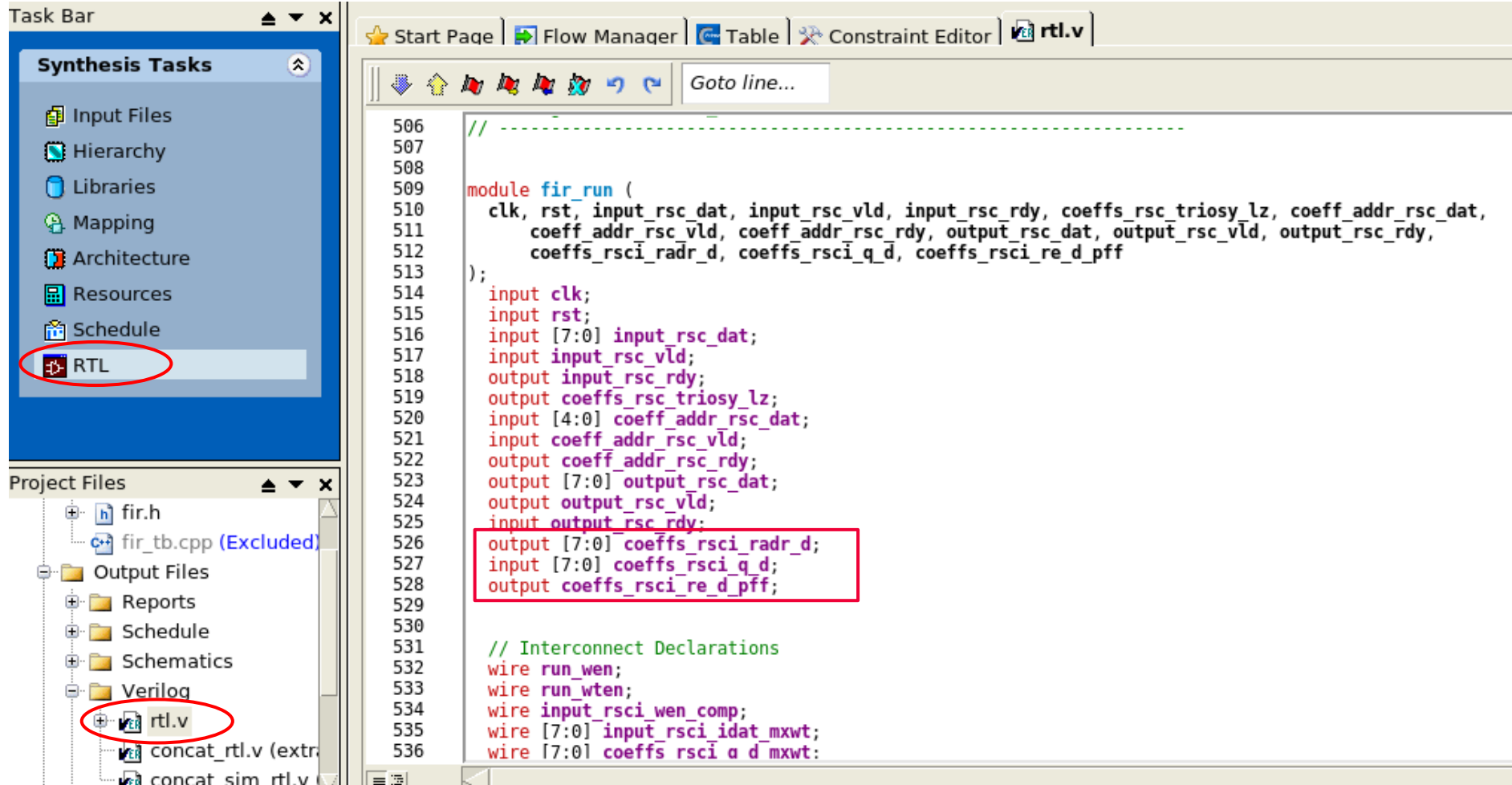
catapult -f directives.tcl

Check coeffs SRAM: (1) size: 256x8b (2) resource type: 1R1W RAM

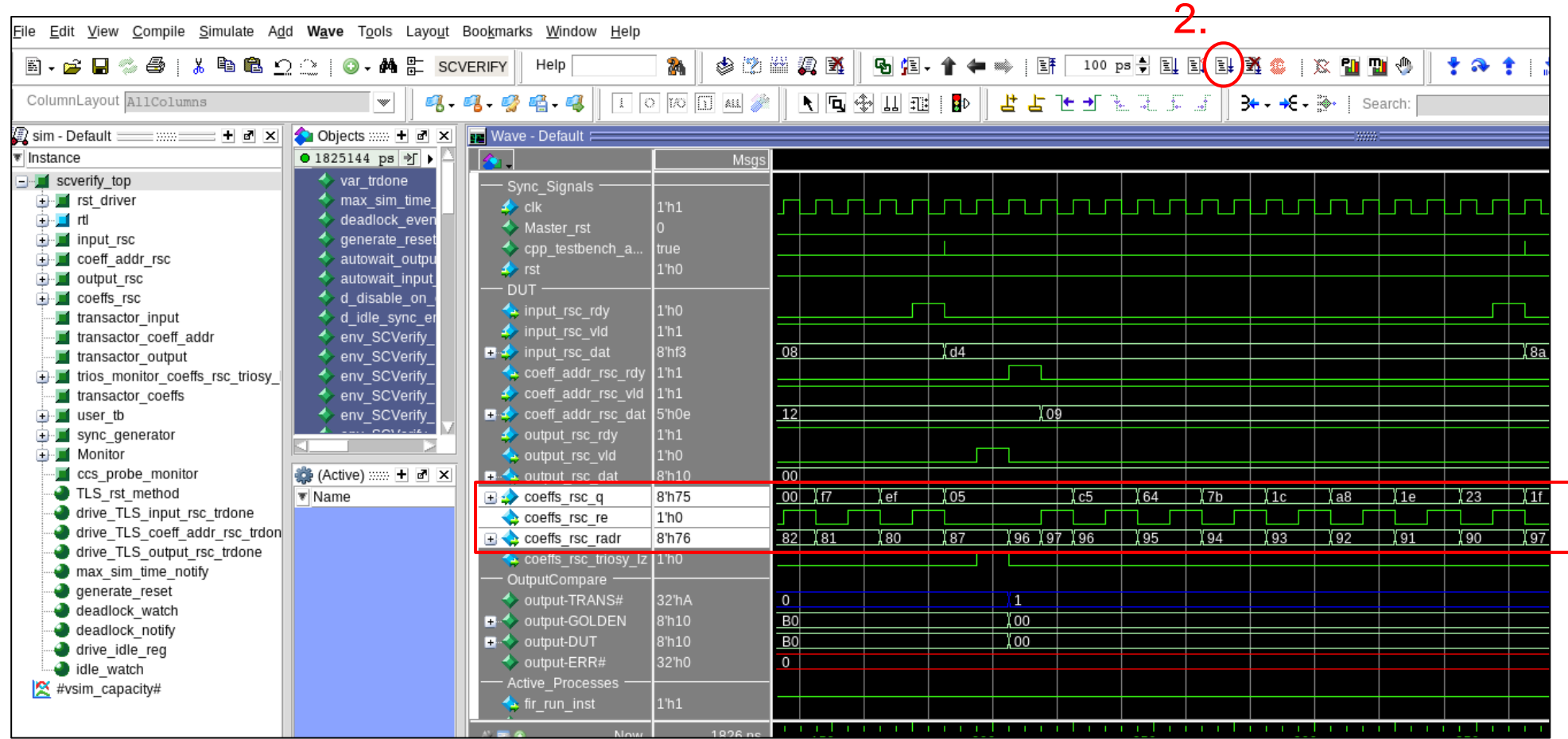
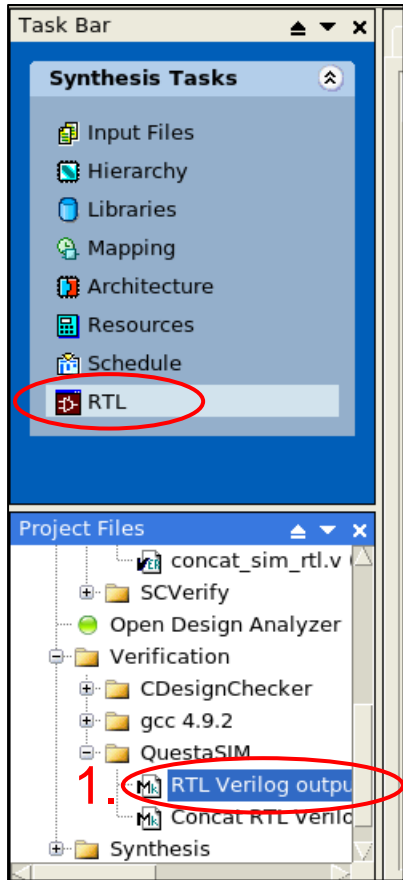
The screenshot displays the Siemens Catapult IDE interface. On the left, the 'Task Bar' shows 'Synthesis Tasks' with 'Architecture' highlighted. Below it, 'Project Files' lists 'fir.v1 (Passed Extract)' and its sub-files. The main area is divided into three panels: 'Instance Hierarchy' showing a tree with 'fir' module containing 'Interface' and 'run' sub-modules; 'Module' showing details of the 'coeffs:rsc (256x8)' resource; and 'Resource: coeffs:rsc' showing configuration options. In the 'Resource: coeffs:rsc' panel, the 'Resource Type' is set to 'ccs\_sample\_mem.ccs\_ram\_sync\_1R1W', which is circled in red. Other options include 'Resource Options' (RdDelay\_100ps: 5), 'Packing Mode' (absolute), 'Block Size', 'Interleave', 'Externalize' (checked), and 'Generate External Enable' (unchecked). The 'Input Delay' section shows fields for Library Delay, Inherited Delay, Port Delay, and Total Delay.

# Check SRAM Interface in RTL

\*\_re\_\*: ram read enable; \*\_radr\_\*: ram read address; \*\_d": ram read data



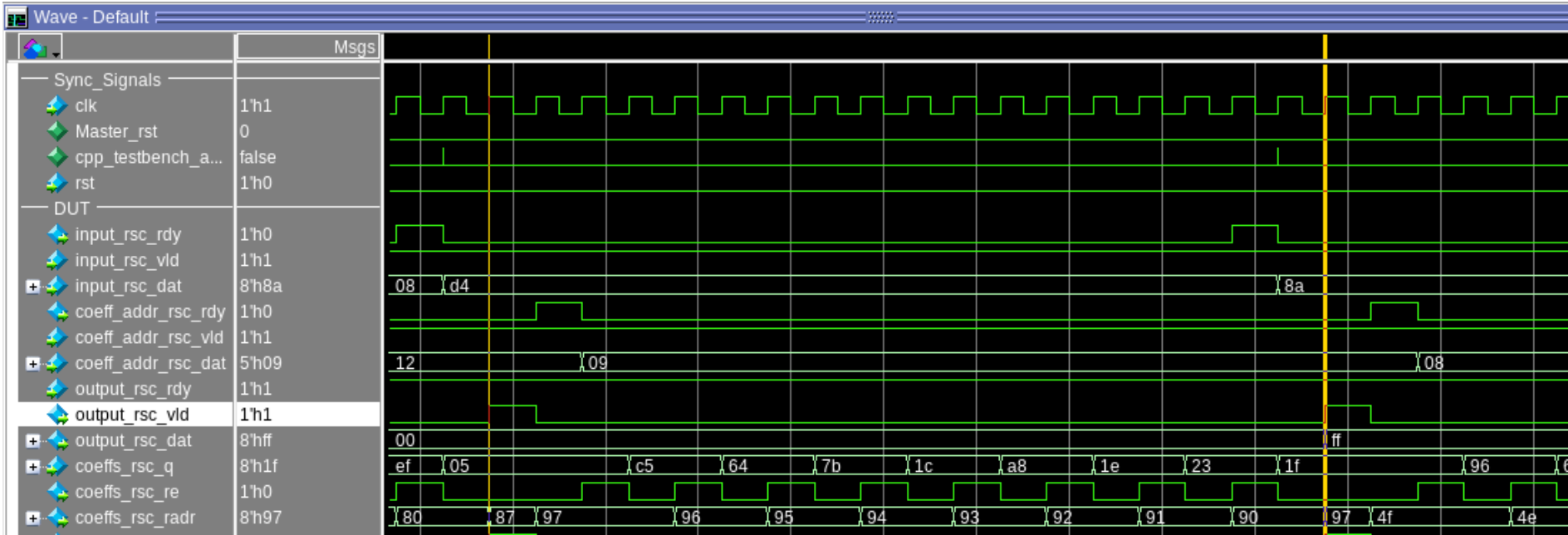
# Check SRAM Interface in Waveform



# How do we improve the throughput

Current throughput is 18

Report: General						
Solution	Latency...	Latency...	Throughput Cycles	Thro...	Slack	Total Area
solution.v1 (new)						
fir.v1 (extract)	17	170.00	18	180.00	7.68	1461.69



# Make convolution in parallel

```
void CCS_BLOCK(run)(ac_channel<ac_int<8>> &input,
                   ac_int<8> coeffs[32][8],
                   ac_channel<ac_int<5,false>> &coeff_addr,
                   ac_channel<ac_int<8>> &output) {
    ac_int<19> temp = 0;
    ac_int<5,false> addr = coeff_addr.read();

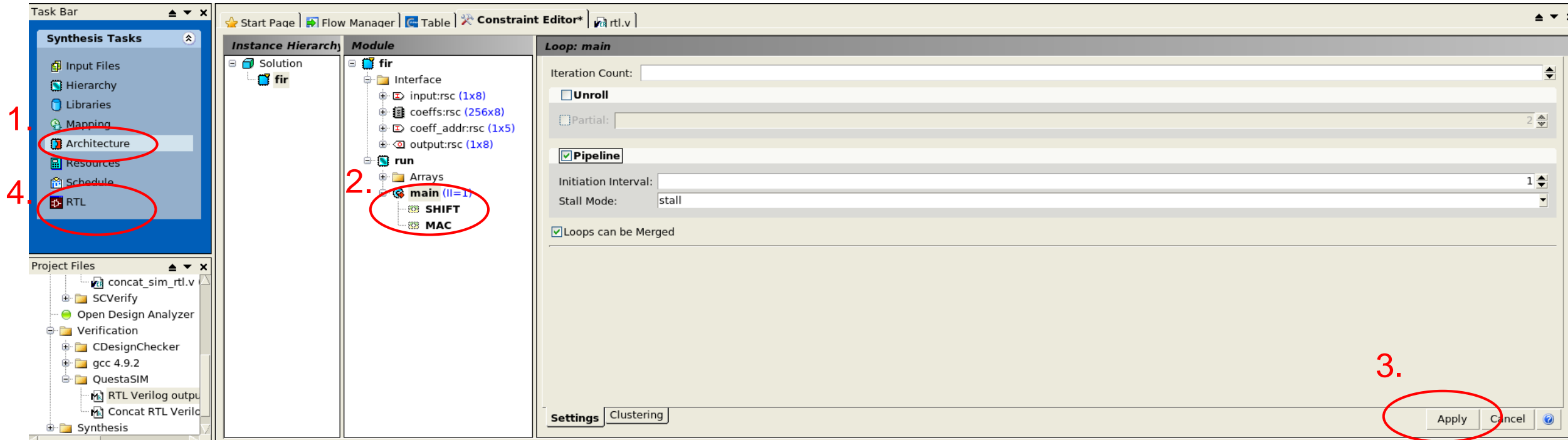
    SHIFT:for (int i = 7; i>=0; i--) {
        if ( i == 0 ) {
            regs[i] = input.read();
        } else {
            regs[i] = regs[i-1];
        }
    }
    MAC:for (int i = 7; i>=0; i--) {
        temp += coeffs[addr][i]*regs[i];
    }
    output.write(temp>>11);
}
```

Make convolution parallel



# Change the Architecture

Unroll 'SHIFT' and 'MAC', and Pipeline 'main'

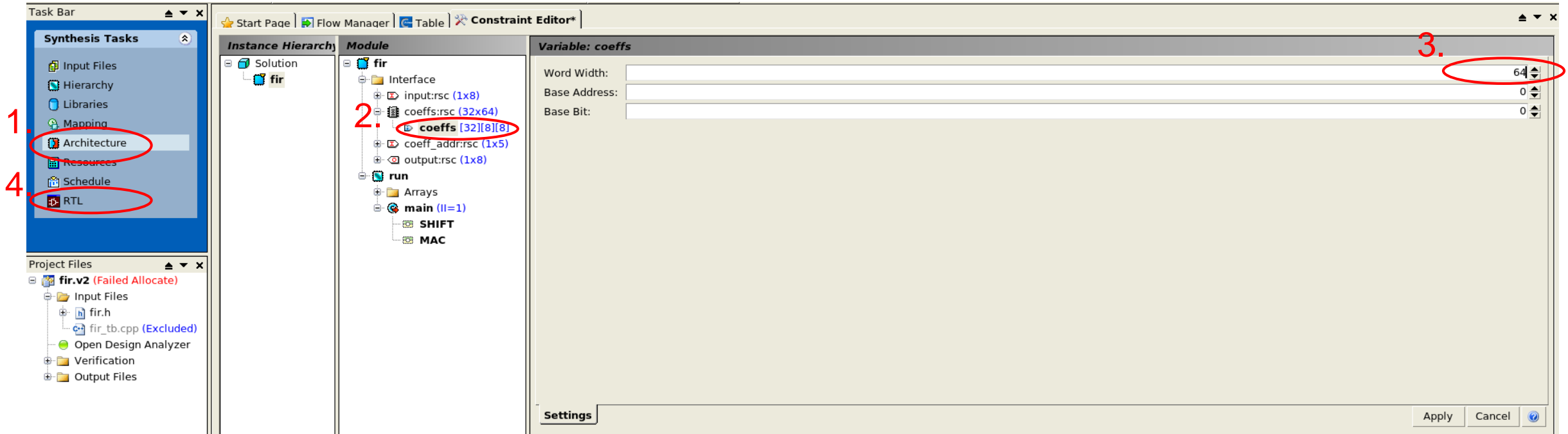


```
# Prescheduled SEQUENTIAL '/fir/run' (total length 4 c-steps)
# insufficient resources 'ccs_sample_mem.ccs_ram_sync_1R1W_rport(2,8,8,256,256,8,5)' to schedule '/fir/run'. 8 are needed, but only 1 instances are available
# please merge accesses to 'coeffs:rsc' to reduce the number of required resources
# Netlist written to file 'schedule.gnt'
# Completed transformation 'architect' on solution 'fir.v2': elapsed time 0.25 seconds, memory usage 1364772kB, peak memory usage 1364772kB
# Design complexity at end of 'architect': Total ops = 68, Real ops = 16, Vars = 23
# Design 'fir' could not schedule partition '/fir/run' - resource competition
```

```
fir.h(23)
fir.h(24)
SCHD-8
SCHD-4
SCHD-39
NET-4
SOL-9
SOL-21
```

# Increase Memory Data Width




Word Width = 64 (why ?)



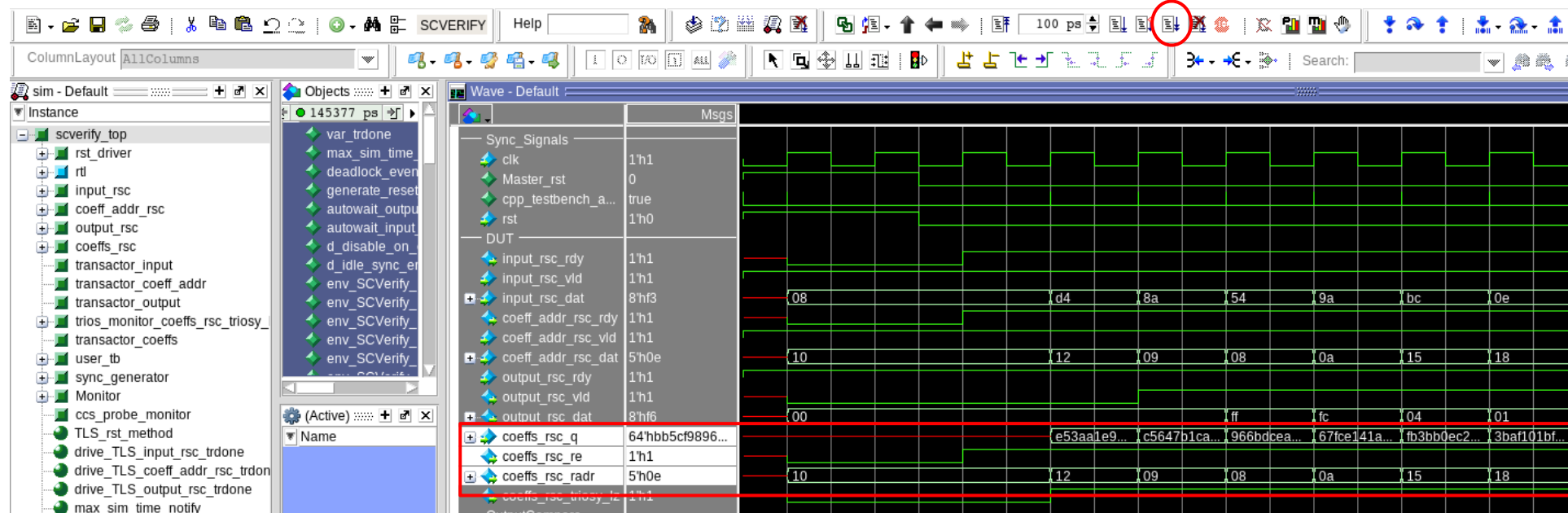
```
## Generating Verilog testbench files
## Makefile for RTL Verilog output 'rtl.v' vs Untimed C++ written to file './scverify/Verify_rtl_v_msim.mk'
## Makefile for Concat RTL Verilog output 'concat_sim_rtl.v' vs Untimed C++ written to file './scverify/Verify_concat_sim_rtl_v_msim.mk'
(i) Completed transformation 'extract' on solution 'fir.v3': elapsed time 3.66 seconds, memory usage 1364772kB, peak memory usage 1364816kB
(i) Design complexity at end of 'extract': Total ops = 269, Real ops = 84, Vars = 199
```

SOL-9  
SOL-21

# Throughput = 1

Report: General						
Solution	Latency...	Latency...	Throughput Cycles	Thro...	Slack	Total Area
 solution.v1 (new)						
 fir.v1 (extract)	17	170.00	18	180.00	7.68	1461.69
 fir.v2 (extract)	2	20.00	1	10.00	7.58	4024.42

2



# | Multiple Blocks

A Design with Two 8-Tap FIRs and One Decimator

# top.h

```
#include "fir.h"
#include "decimator.h"
class top {
    ac_channel<ac_int<8> > connect0;
    ac_channel<ac_int<8> > connect1;
    fir block0;
    fir block1;
    decimator block2;
public :
    top () {}
    #pragma hls_design interface top
    void CCS_BLOCK(run) (ac_channel<ac_int<8> > &din,
                        ac_int<8> coeffs[8],
                        ac_channel<ac_int<8> > &dout)
{
    block0.run(din,coeffs,connect0);
    block1.run(connect0,coeffs,connect1);
    block2.run(connect1,dout);
    ...
}
```

# decimator.h

```
class decimator {
    int count;
public :
    decimator () {
        count = 0;
    }
    #pragma hls_design interface
    void CCS_BLOCK(run) (ac_channel<ac_int<8> > &din,
                        ac_channel<ac_int<8> > &dout) {
        if (count==4)
            count = 0;
        ac_int<8> temp = din.read();
        if (count==0)
            dout.write(temp);
        count++;
    }
    ...
}
```

# fir.h

```
class fir {
    ac_int<8> regs[8];
public:
    fir() {
        for (int i = 7; i>=0; i--)
            { regs[i] = 0; }
    }
    ...
}
```

```
#pragma hls_design interface
void CCS_BLOCK(run)(ac_channel< ac_int<8> > &input,
                    ac_int<8> coeffs[8],
                    ac_channel< ac_int<8> > &output) {
    ac_int<19> temp = 0;
    SHIFT:for (int i = 7; i>=0; i--) {
        if ( i == 0 ) {
            regs[i] = input.read();
        } else {
            regs[i] = regs[i-1];
        }
    }
    MAC:for (int i = 7; i>=0; i--) {
        temp += coeffs[i]*regs[i];
    }
    output.write(temp>>11);
    ...
}
```

# Start from the 'Architecture' Stage

catapult -f directives.tcl &

There are three design blocks: block0, block1, block2

Set 'DirectInput' for the coeffs

The screenshot displays the Siemens Catapult Architecture stage interface. The left sidebar shows the 'Synthesis Tasks' panel with 'Architecture' selected (1). The 'Instance Hierarchy' panel shows the project structure with 'top' selected (2). The 'Module' panel shows the 'coeffs:rsc' resource selected (3). The 'Resource: coeffs:rsc' configuration panel shows the 'Resource Type' set to 'ccs\_ioport.ccs\_in' (4). The 'Input Delay' and 'Output Delay' sections are visible. The 'Apply' button is circled (5). A red box at the bottom contains the warning: 'Resource '/top/coeffs:rsc' with variable connected to multiple sub-blocks not mapped to '[DirectInput]''. The 'Settings' and 'Mapping' tabs are at the bottom.

1. Architecture

2. top

3. coeffs:rsc (1x6)

4. Select [DirectInput]

5. Apply

Resource: coeffs:rsc

Resource Type: ccs\_ioport.ccs\_in

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

Resource '/top/coeffs:rsc' with variable connected to multiple sub-blocks not mapped to '[DirectInput]'



# Set FIFO Depth as '0'

Set FIFO depth as '0' for connect0 and connect1

The screenshot displays the Siemens EDA software interface, specifically the Constraint Editor. The left sidebar shows the 'Synthesis Tasks' panel with 'Architecture' selected, and the 'Project Files' panel showing the project structure. The main workspace is divided into three panes: 'Instance Hierarchy', 'Module', and 'Resource: connect0:cns'.

In the 'Module' pane, the 'Interconnect' section shows two resources: 'connect0:cns (1x8)' and 'connect1:cns (1x8)', both of which are circled in red. The 'Resource: connect0:cns' pane shows the 'FIFO Depth' field set to 0, also circled in red. Below this, the 'Input Delay' and 'Output Delay' sections show various delay fields (Library Delay, Inherited Delay, Port Delay, Total) all set to 0 ns.

At the bottom right, the 'Apply' button is circled in red. The 'Settings' and 'Mapping' tabs are visible at the bottom left of the main workspace.

# Look Report

Latency = ?, Throughput = ?

Area: block0=?, block1=?, block2=?

Task Bar

Synthesis Tasks

Input Files

Hierarchy

Libraries

Mapping

Architecture

Resources

Schedule

RTL

Start Page | Flow Manager | **Table** | Constraint Editor

Report: General

Solution	Latency...	Latency...	Throug...	Throug...	Slack	Total Area
solution.v1 (new)						
top.v1 (extract)	9	90.00	10	100.00	6.38	2888.22

Task Bar

Synthesis Tasks

Input Files

Hierarchy

Libraries

Mapping

Architecture

Resources

Schedule

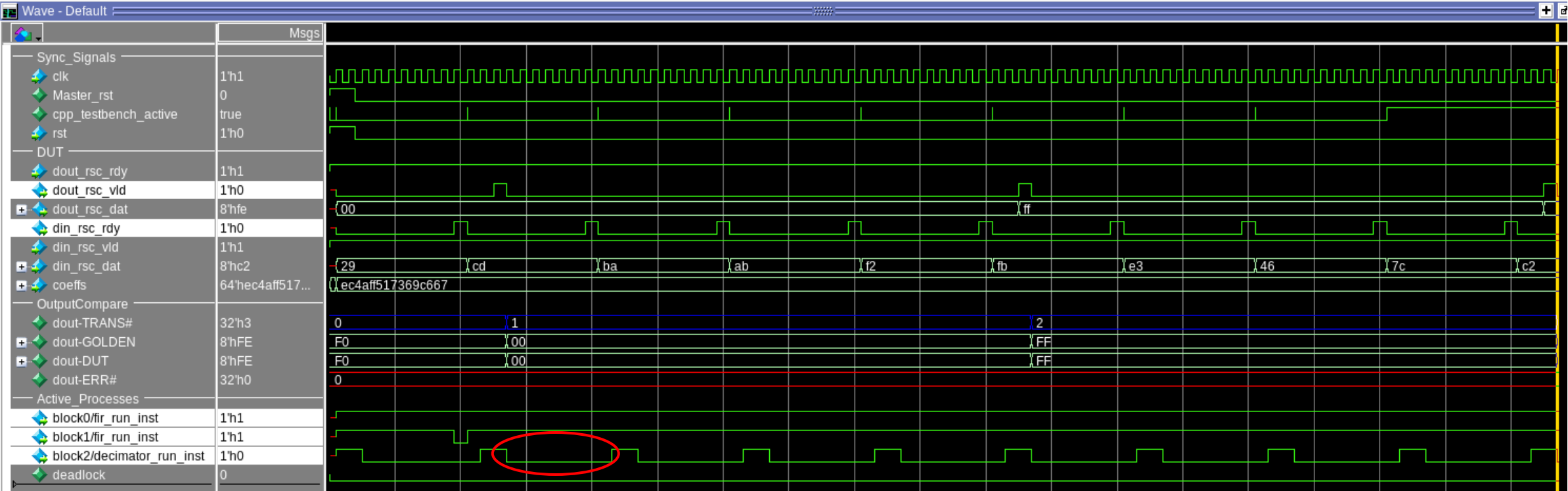
RTL

Start Page | Flow Manager | **Table** | Constraint Editor

Report: Area Score

Solution	Registers	MUX	Functional	Logic	Memory	FSM Reg	FSM Comb	FSM	Datapath	Total Reg	Total Area	Rom
solution.v1 (new)												
top.v1 (extract)	1340.64	406.45	859.24	228.89	0.00	48.00	5.00	53.00	2835.22	1388.64	2888.22	0.00
allocate	1998.19	621.38	1076.28	3.19	0.00	0.00	0.00	0.00	3699.04	1998.19	3699.04	0.00
dpfsm	1340.64	510.72	1076.28	238.21	0.00	48.00	5.00	53.00	3165.85	1388.64	3218.85	0.00
extract	1340.64	406.45	859.24	228.89	0.00	48.00	5.00	53.00	2835.22	1388.64	2888.22	0.00
block0	632.02	203.22	426.16	108.86	0.00	16.00	2.00	18.00	1370.26	648.02	1388.26	0.00
block1	632.02	203.22	426.16	108.86	0.00	16.00	2.00	18.00	1370.26	648.02	1388.26	0.00
block2	76.61	0.00	6.92	11.17	0.00	16.00	1.00	17.00	94.70	92.61	111.70	0.00

# RTL Simulation



why idle ?

# Improve Throughput

Improve the throughput from 10 to 1 by yourself

Task Bar

Synthesis Tasks

- Input Files
- Hierarchy
- Libraries
- Mapping
- Architecture
- Resources
- Schedule
- RTL

Start Page | Flow Manager | **Table** | Constraint Editor

Report: General

Solution /	Latency Cycles	Latency Time	Throughput Cycles	Throughput Time	Slack	Total Area
solution.v1 (new)						
top.v1 (extract)	9	90.00	10	100.00	6.38	2888.22
top.v2 (extract)	2	20.00	1	10.00	4.59	7005.00

# RTL Simulation (Throughput=1)

One dout every four din because of the decimation

