Discover NX Design Suite 23.5

NanoXplore

Agenda

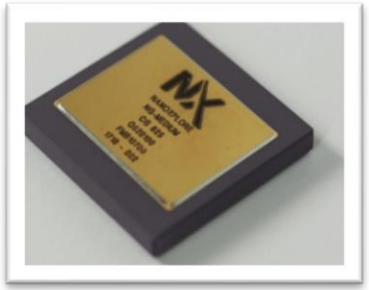
- Introduction
- Graphical User Interface
- Design Flow
- IP & Debug Tools
- Documentation
- Conclusion

NanoXplore

Introduction

NanoXplore

NX Products – Overview



NG medium

Low-End FPGA

- 35kLUTs/32kDFFs
- 3Mb RAM
- 112 DSP
- No HSSL
- No Hard IP Processor

- Companion chip

ESCC9000 qualified

ultra 300

Mid-End FPGA

- 290kLUTs/273kDFFs
- 21Mb RAM
- 896 DSP
- 16x HSSL 12G
- ADC/DAC

- Payload
- Platform
- Sensor control
- Power control loop

NG ultra

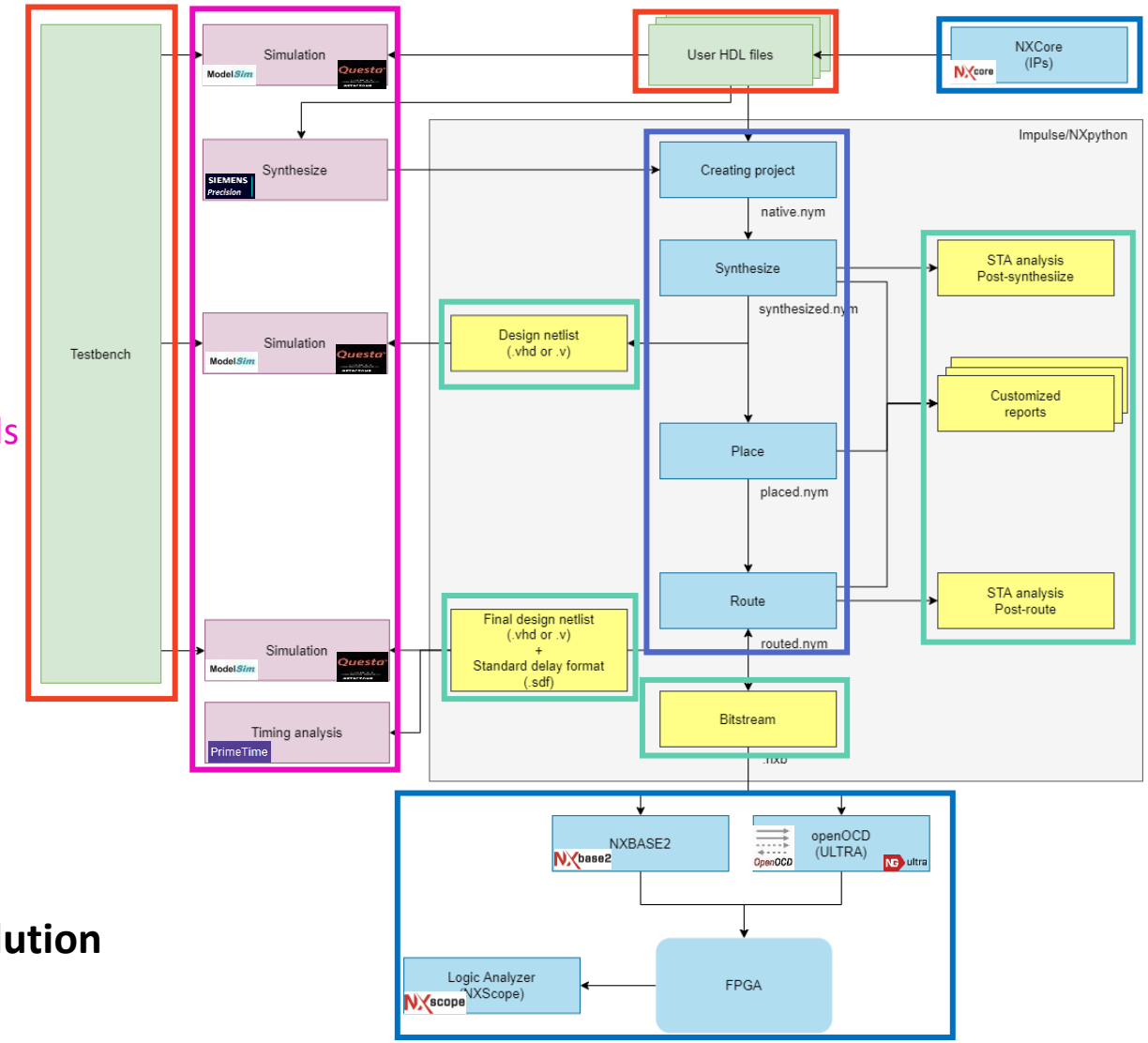
High-End FPGA

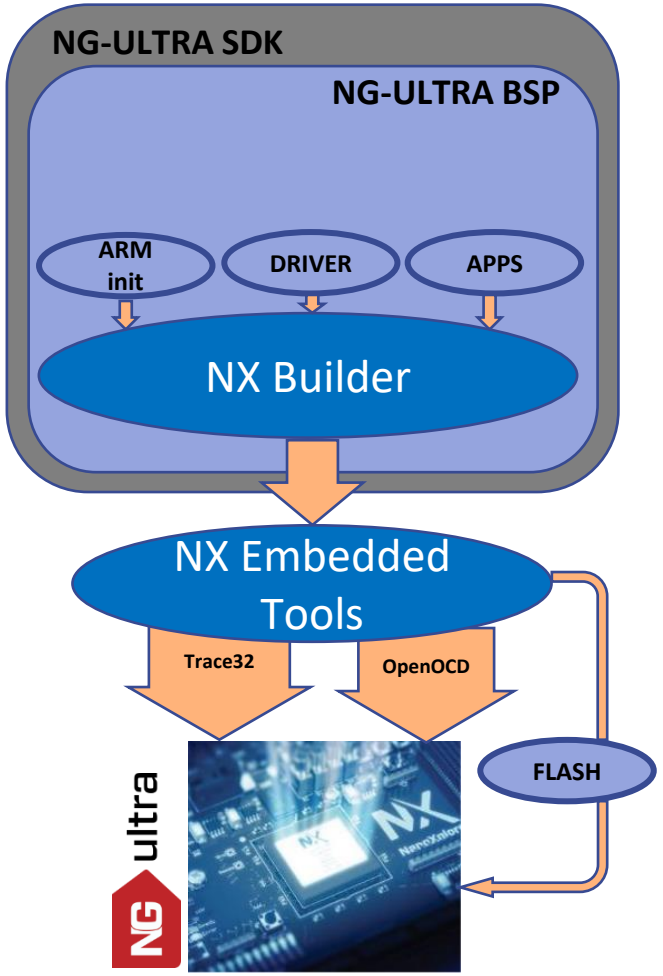
- 537kLUTs/505kDFFs
- 32Mb RAM
- 1344 DSP
- 32x HSSL 12G
- Quad-core ARM-R52 (SoC)

- Payload
- Platform

NX Design Suite – Overview

User Resources
NX Tools
NX Output Files
Third-Party Tools





Complete solution

Graphical User Interface

NanoXplore

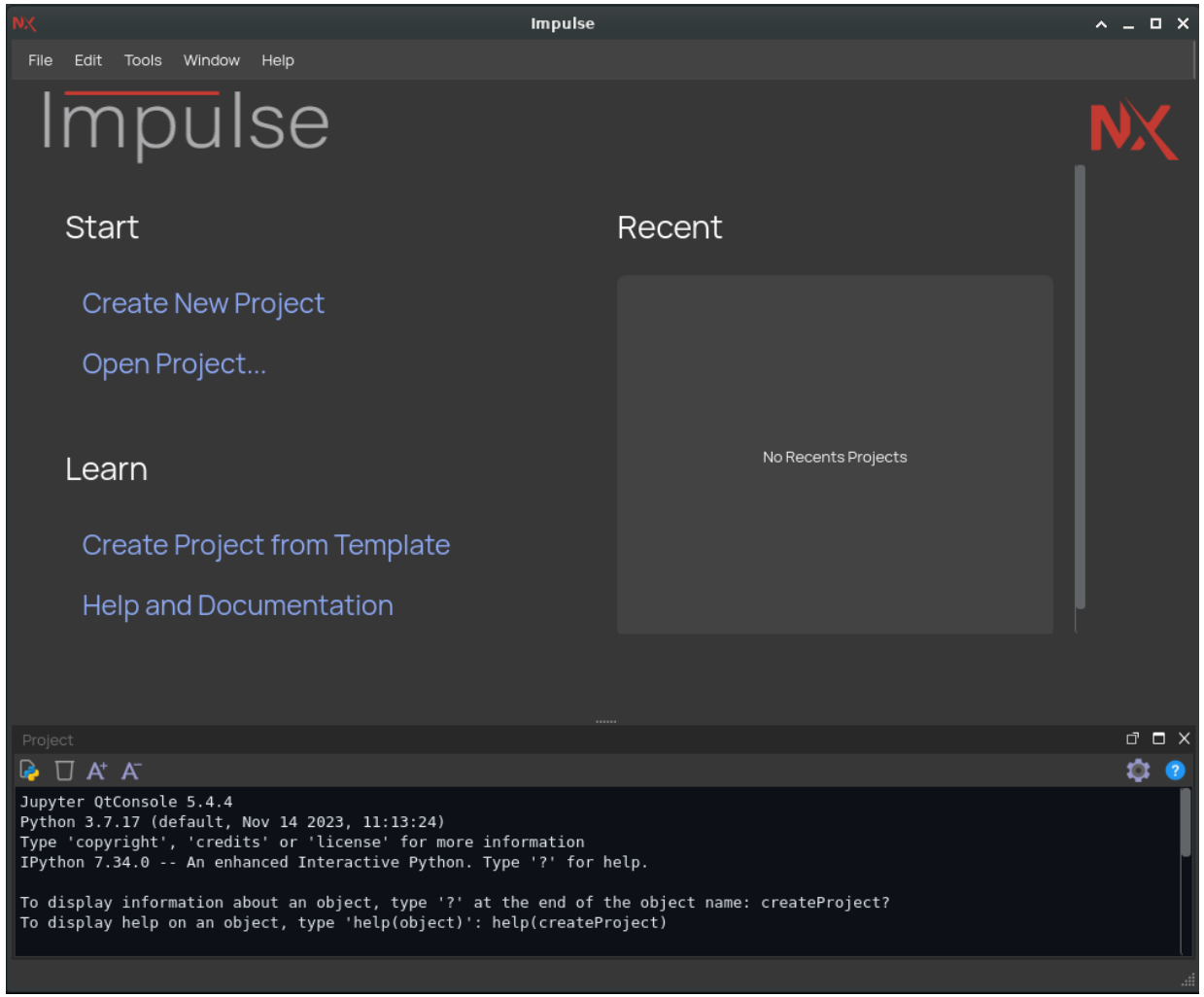
Impulse 23.5

To launch the design flow, you can:

- Use GUI
- Use NXpython (Python based tool)

The steps before launching the design flow:

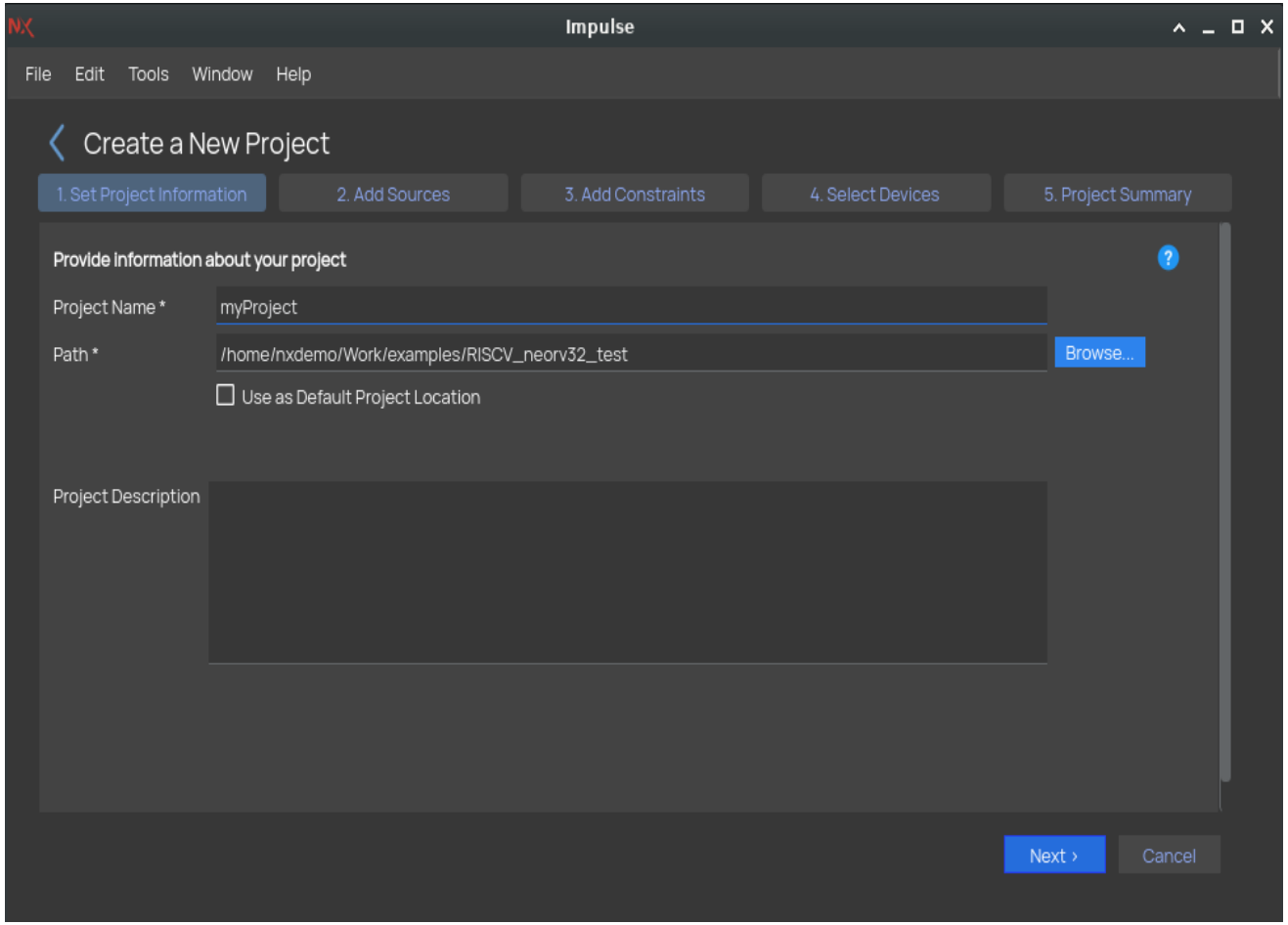
- Create and configure a new project
- Add source files
- Set tool options (Synthesis, Place, Route & Bitstream Generation)
- Add constraints (IOs, timings, placement)



Create New Project

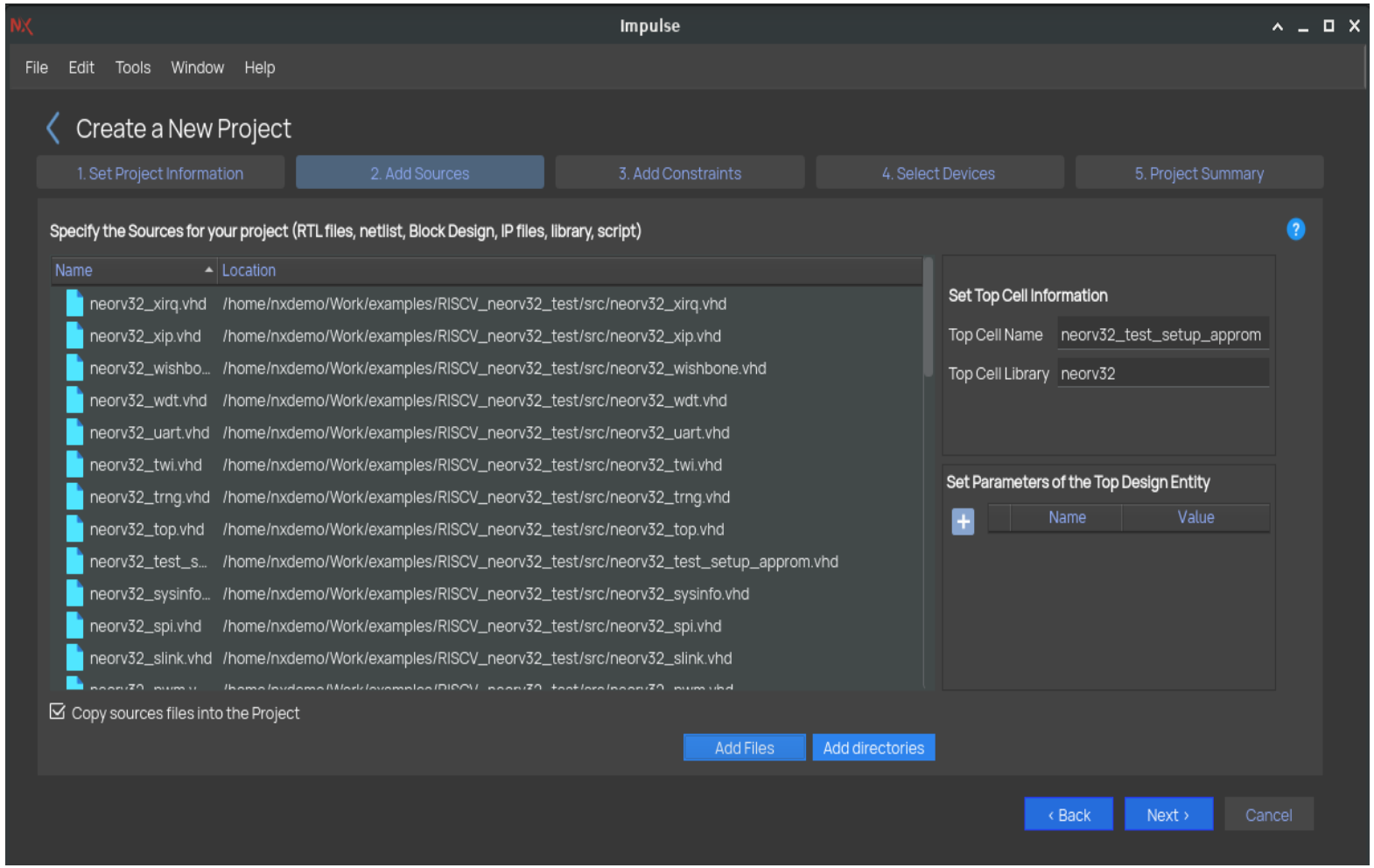
To create a project from scratch:

- Create new project
- Provide a project name
- Specify project path



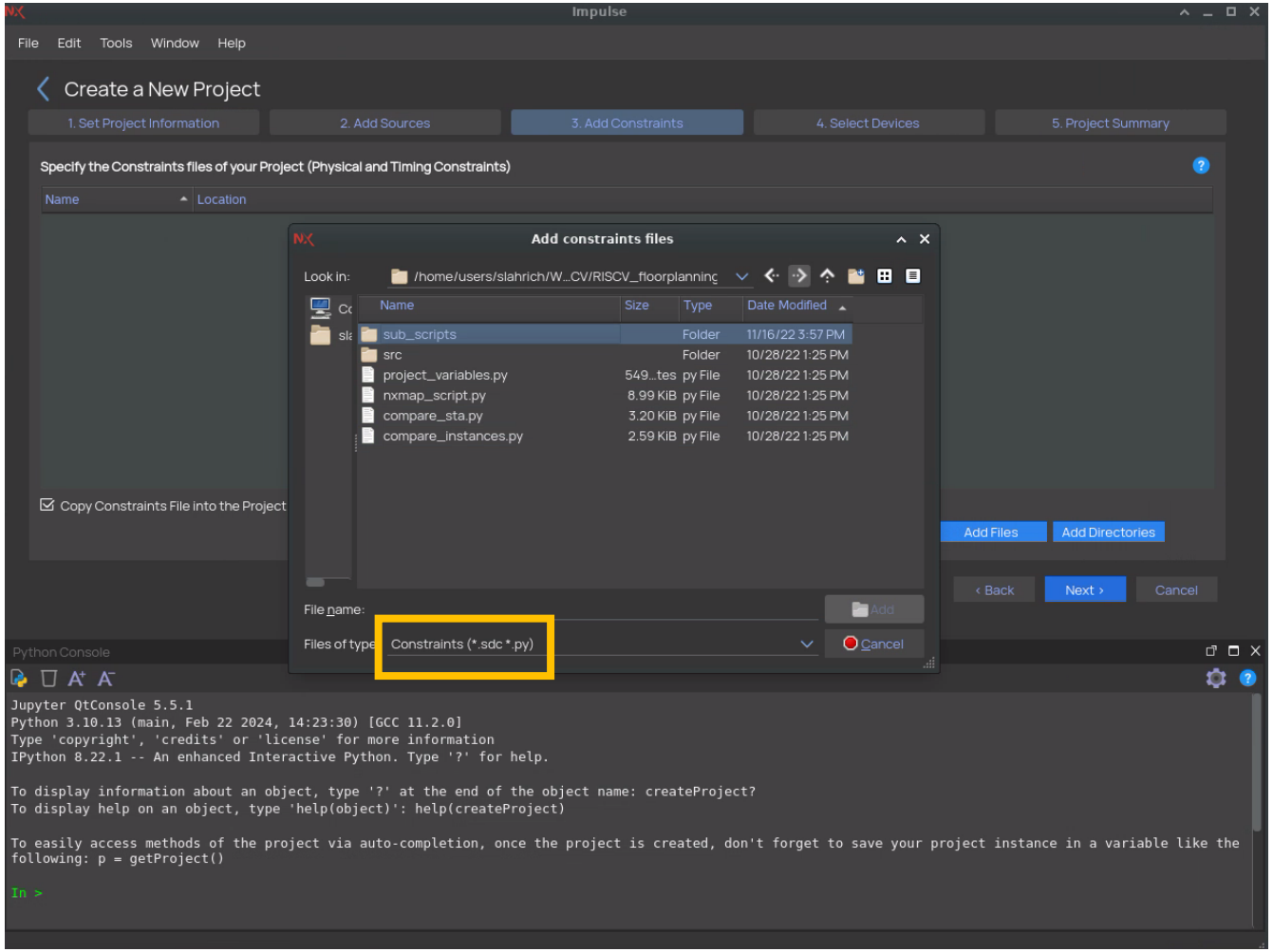
Add files

- Add source files
- Specify top cell name
- Specify top cell library



Add Constraints

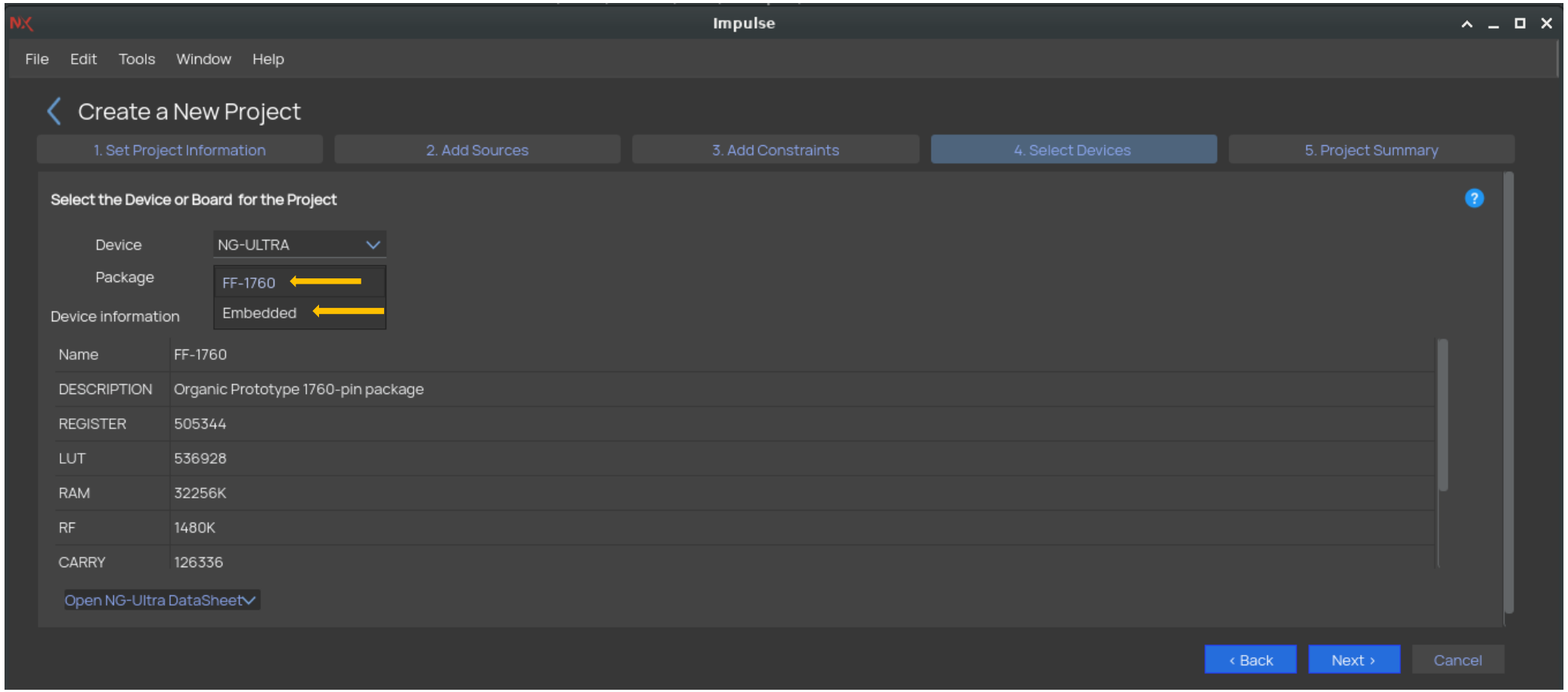
For timing constraints, SDC or Python file (NXDC) is used.



For more details => https://nanoxplore-wiki.atlassian.net/wiki/spaces/NAN/pages/357467312/NxDesignSuite+23.5+STA

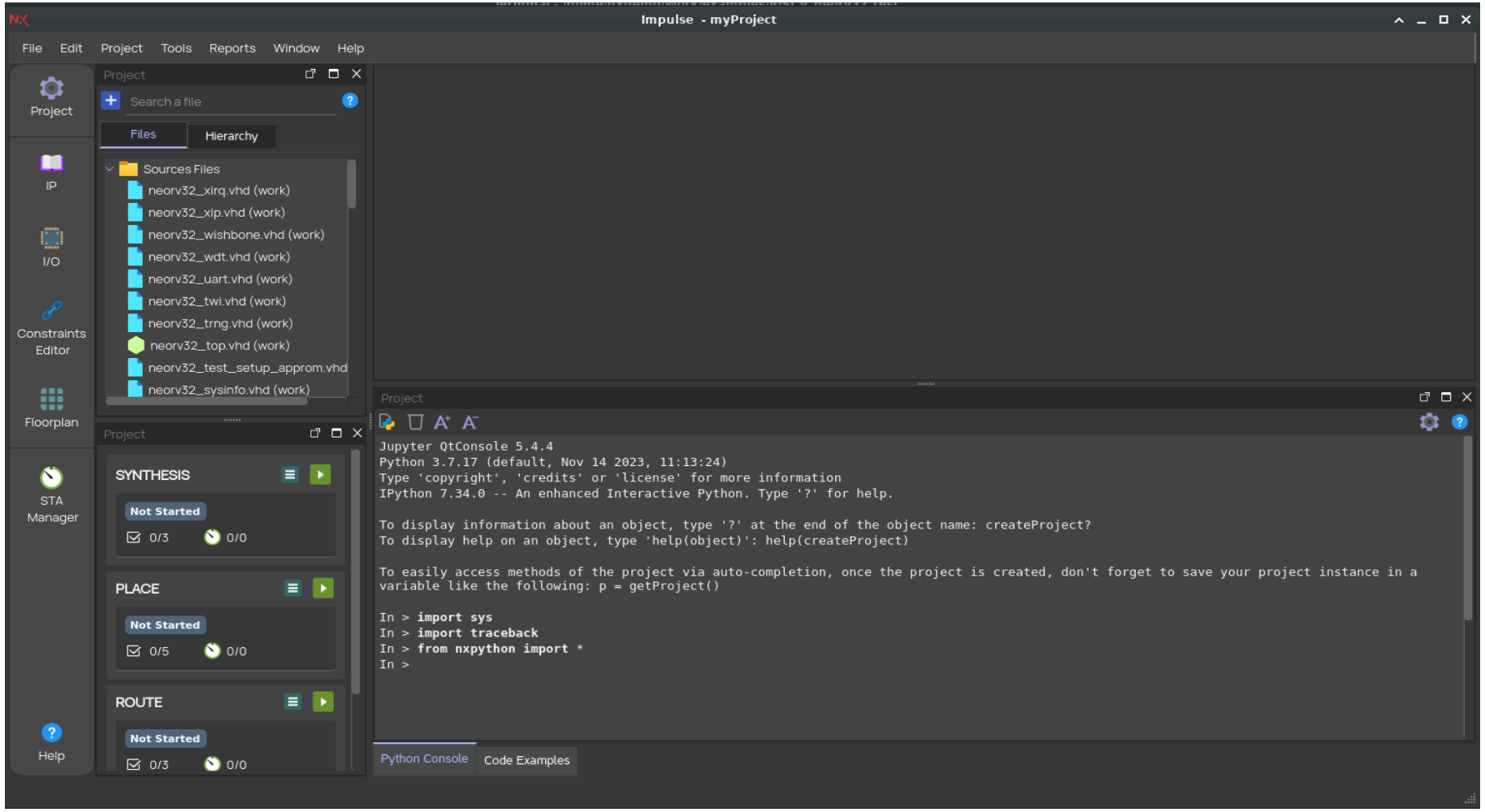
Select Devices

In addition to the device selection, the package can be specified.



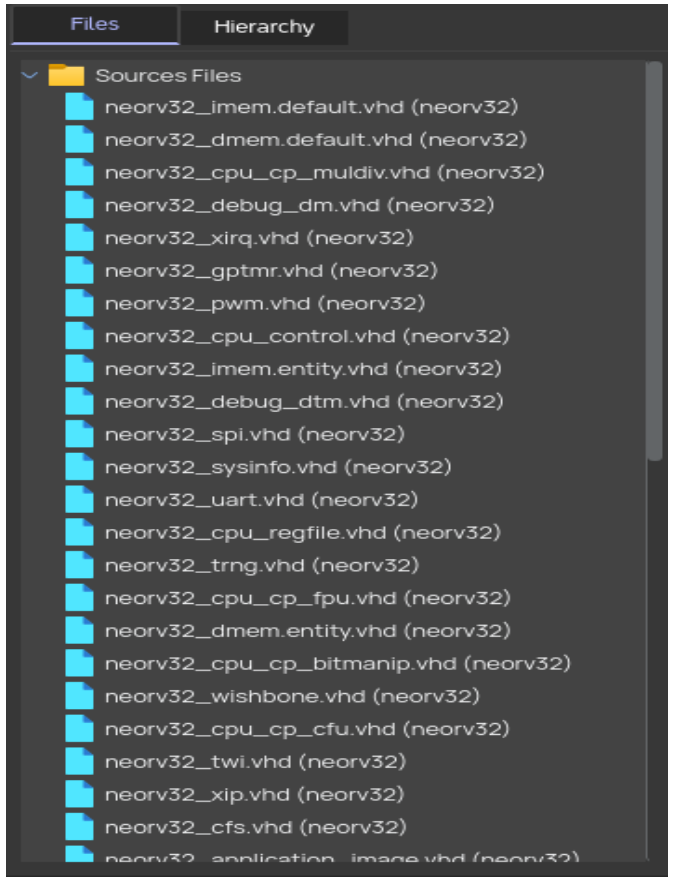
First step into Impulse

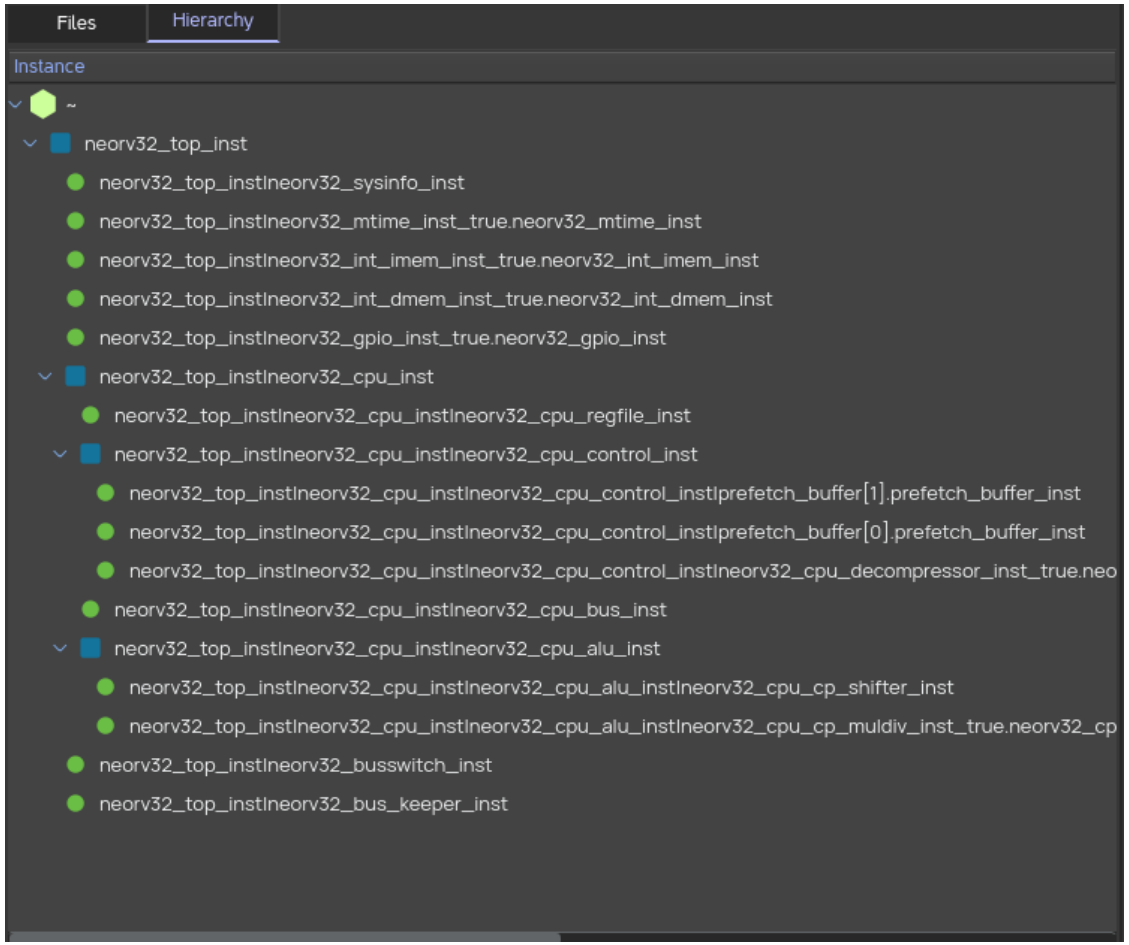
This window appears when project settings have been done.
Then you can launch the full flow or a step of the flow.



Widgets – Files and Hierarchy

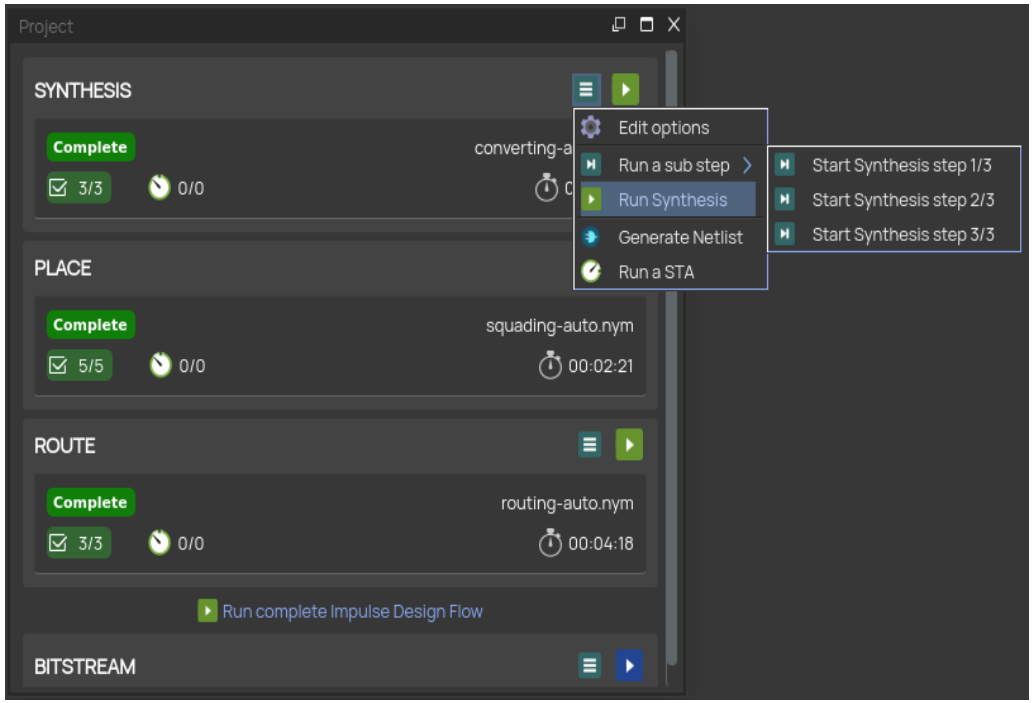
Hierarchy Project information is generated once the synthesis has been done.



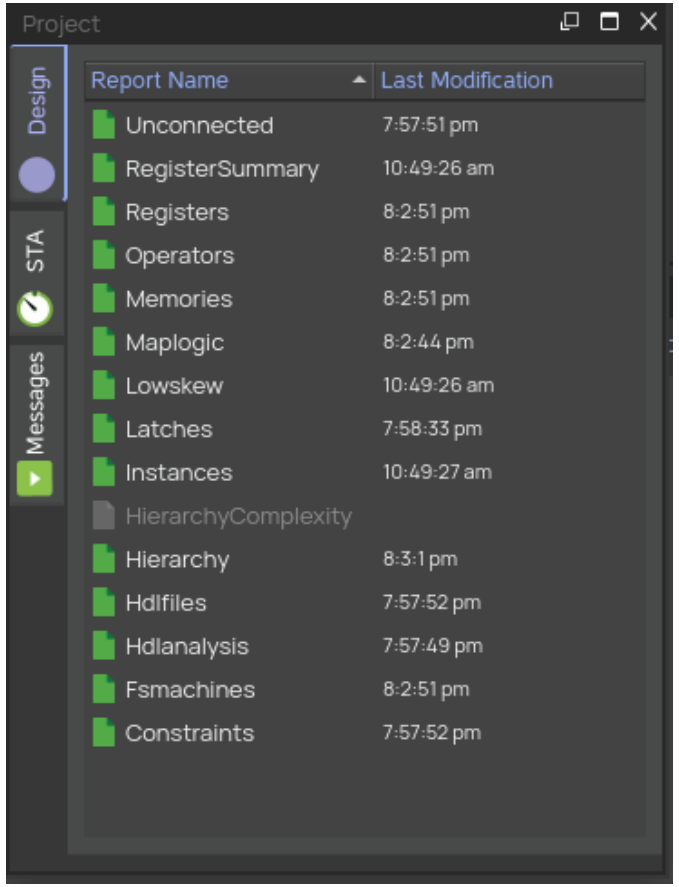


Widgets – Flow Navigator & Reports

Flow Navigator



Reports



Widgets

You can « **Edit files** » in Impulse

neorv32_test_setup_approm.vhd | neorv32_top.vhd

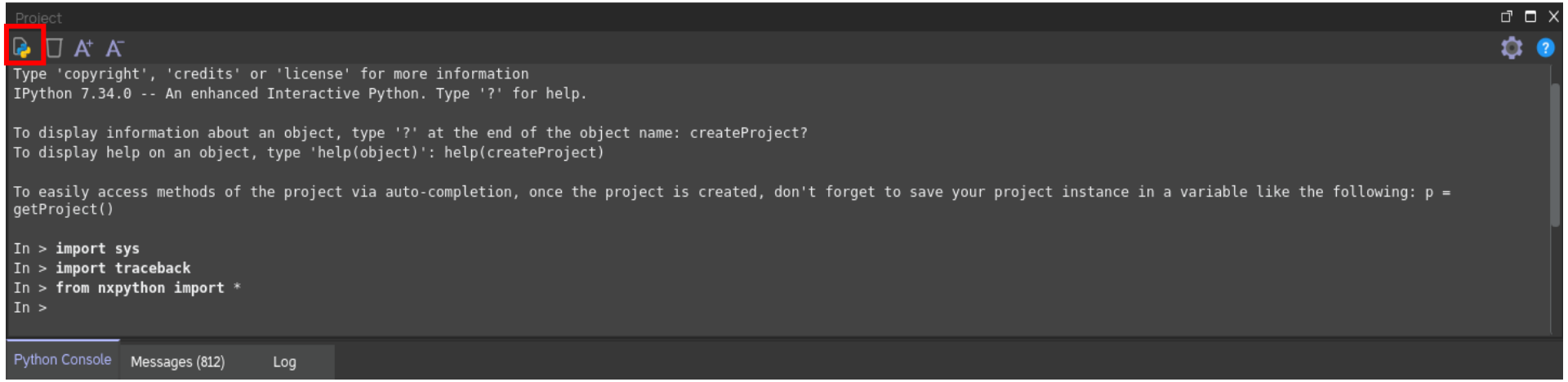
```vhdl
-- # The NEORV32 Processor - https://github.com/stnolting/neorv32              (c) Stephan Nolting #
-- ##############################################################################################

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

library neorv32;
use neorv32.neorv32_package.all;

entity neorv32_top is
  generic (
    -- General --
    CLOCK_FREQUENCY              : natural;           -- clock frequency of clk_i in Hz
    HW_THREAD_ID                 : natural := 0;      -- hardware thread id (32-bit)
    CUSTOM_ID                    : std_ulogic_vector(31 downto 0) := x"00000000"; -- custom user-defined ID
    INT_BOOTLOADER_EN            : boolean := false;  -- boot configuration: true = boot explicit bootloader; false = boot from int/ext (I)MEM

    -- On-Chip Debugger (OCD) --
    ON_CHIP_DEBUGGER_EN          : boolean := false;  -- implement on-chip debugger

    -- RISC-V CPU Extensions --
    CPU_EXTENSION_RISCV_B        : boolean := false;  -- implement bit-manipulation extension?
    CPU_EXTENSION_RISCV_C        : boolean := false;  -- implement compressed extension?
    CPU_EXTENSION_RISCV_E        : boolean := false;  -- implement embedded RF extension?
    CPU_EXTENSION_RISCV_M        : boolean := false;  -- implement mul/div extension?
    CPU_EXTENSION_RISCV_U        : boolean := false;  -- implement user mode extension?
    CPU_EXTENSION_RISCV_Zfinx    : boolean := false;  -- implement 32-bit floating-point extension (using INT regs!)
    CPU_EXTENSION_RISCV_Zicsr    : boolean := true;   -- implement CSR system?
    CPU_EXTENSION_RISCV_Zicntr   : boolean := true;   -- implement base counters?
    CPU_EXTENSION_RISCV_Zihpm    : boolean := false;  -- implement hardware performance monitors?
    CPU_EXTENSION_RISCV_Zifencei : boolean := false;  -- implement instruction stream sync.?
    CPU_EXTENSION_RISCV_Zmmul    : boolean := false;  -- implement multiply-only M sub-extension?
    CPU_EXTENSION_RISCV_Zxcfu    : boolean := false;  -- implement custom (instr.) functions unit?

    -- Tuning Options --
    FAST_MUL_EN                  : boolean := false;  -- use DSPs for M extension's multiplier
    FAST_SHIFT_EN                : boolean := false;  -- use barrel shifter for shift operations
    CPU_CNT_WIDTH                : natural := 64;     -- total width of CPU cycle and instret counters (0..64)
    CPU_IPB_ENTRIES              : natural := 2;      -- entries in instruction prefetch buffer, has to be a power of 2, min 2

    -- Physical Memory Protection (PMP) --
    PMP_NUM_REGIONS              : natural := 0;      -- number of regions (0..16)
    PMP_MIN_GRANULARITY          : natural := 4;      -- minimal region granularity in bytes, has to be a power of 2, min 4 bytes
```

/home/nxdemo/Work/examples/RISCV_neorv32_test/src/neorv32_top.vhd

Widgets

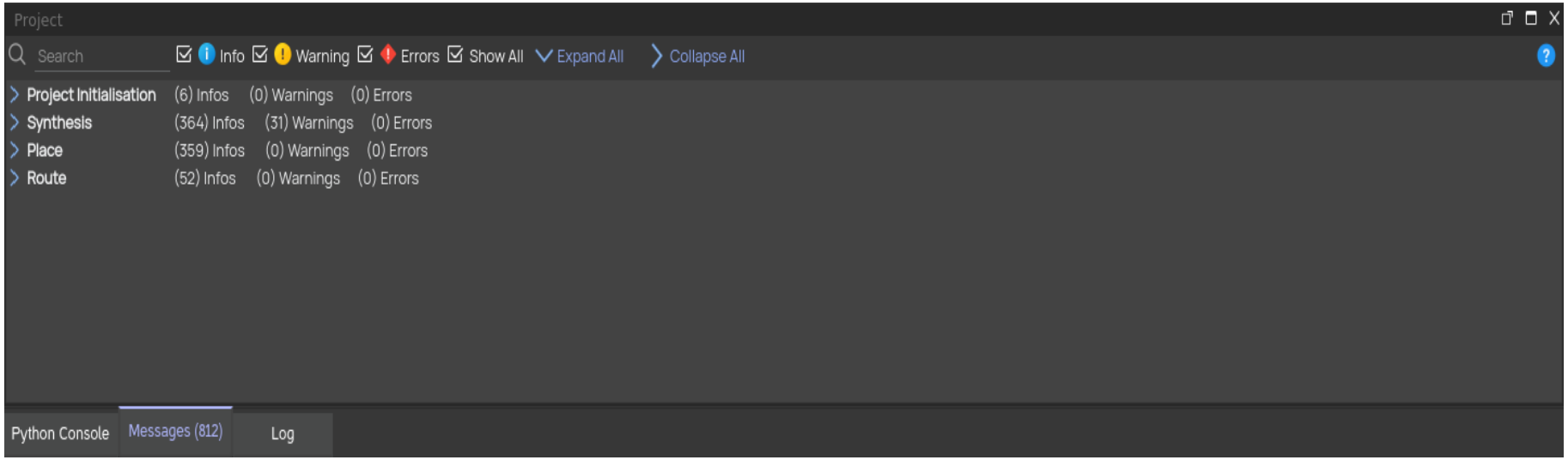
In **Python Console** you can:

- Launch NXpython commands
- Execute a NXpython script



In **Messages** field you can check at each step generated:

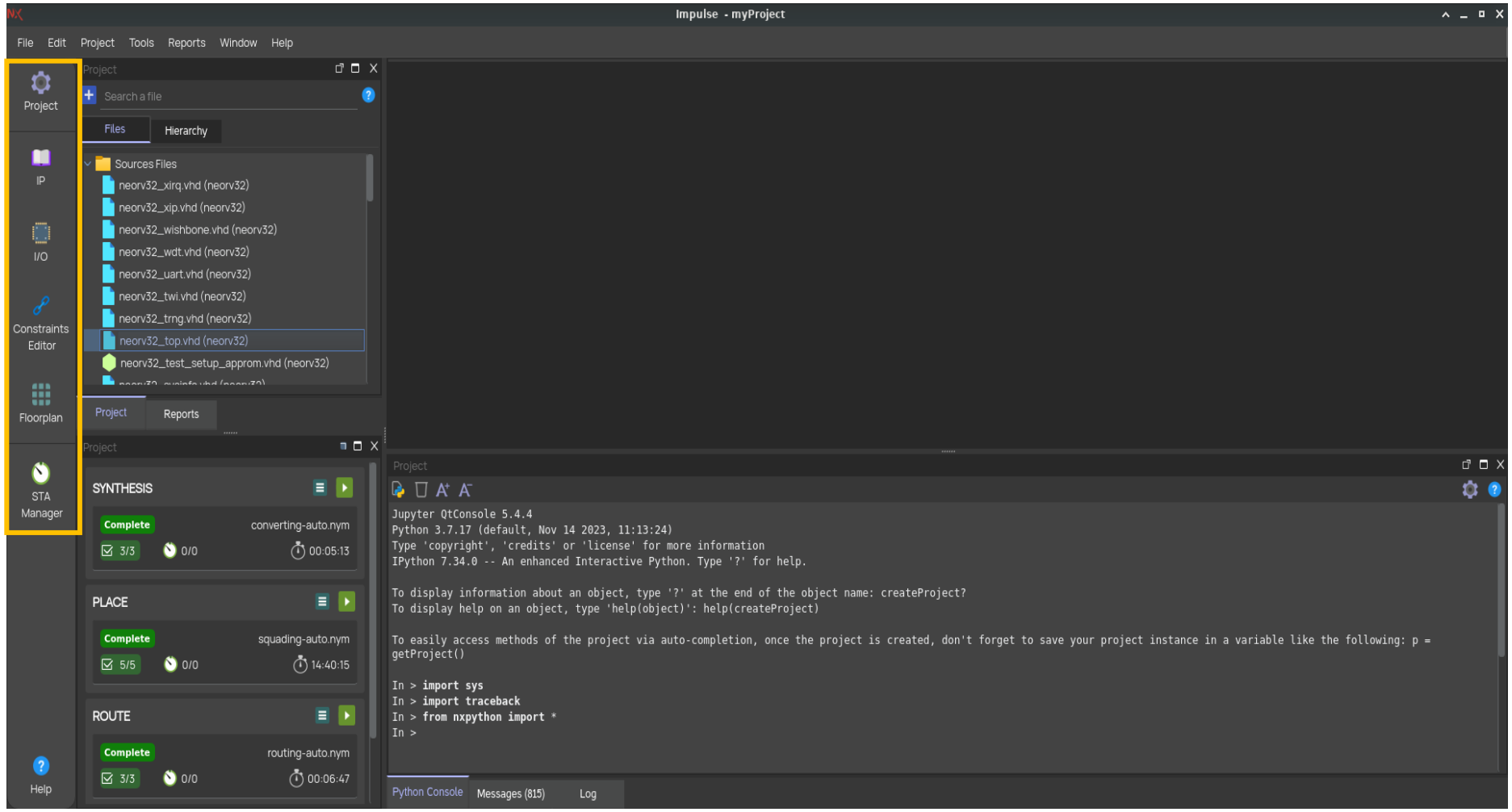
- Information
- Warnings
- Errors



Widgets

On Impulse interface, the widgets on the left gives you access to:

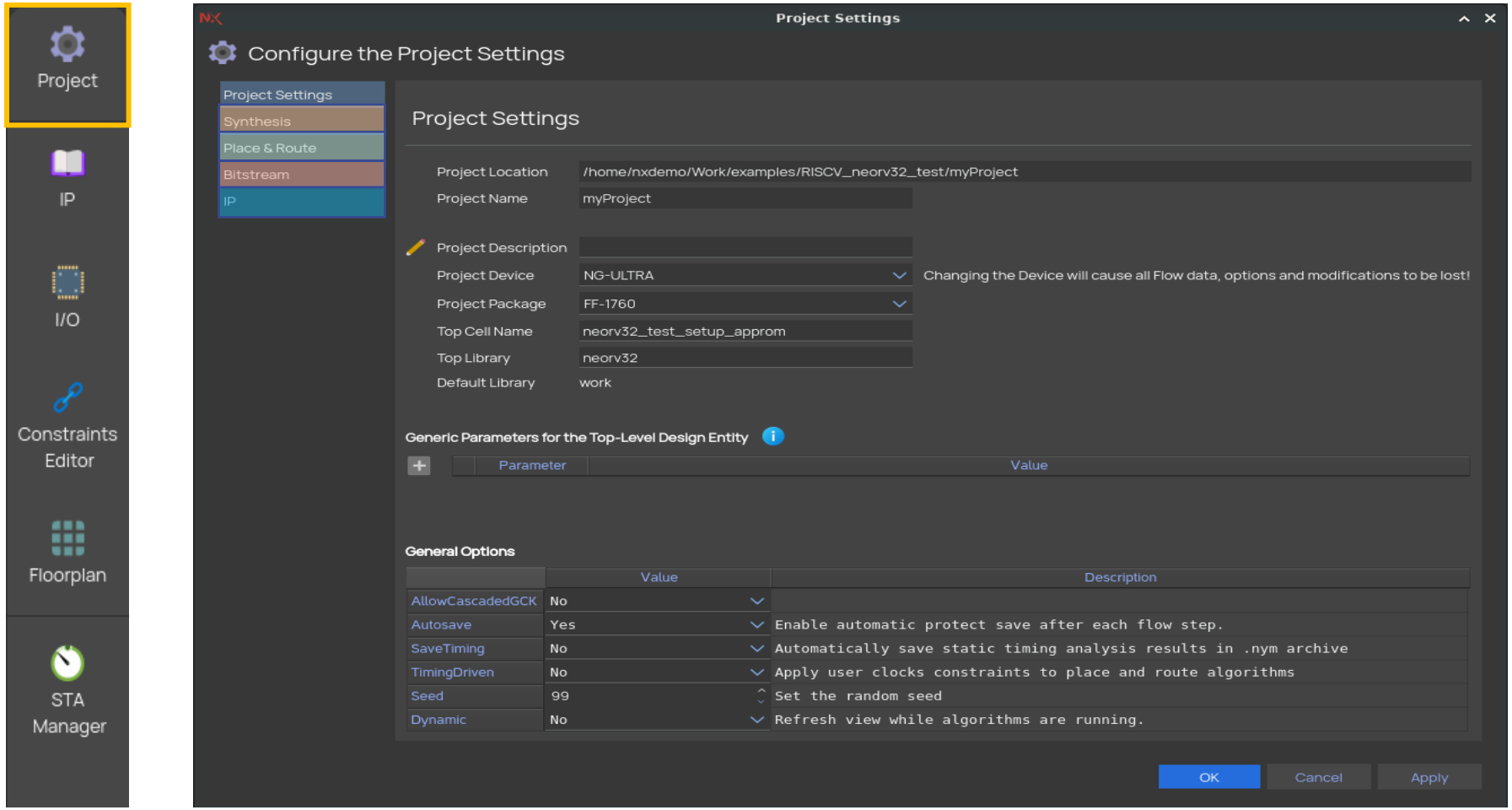
- Project Settings
- NXcore IP Catalog
- IOs Configuration
- STA Constraints Editor
- Floorplan Viewer
- STA Manager



Project Settings

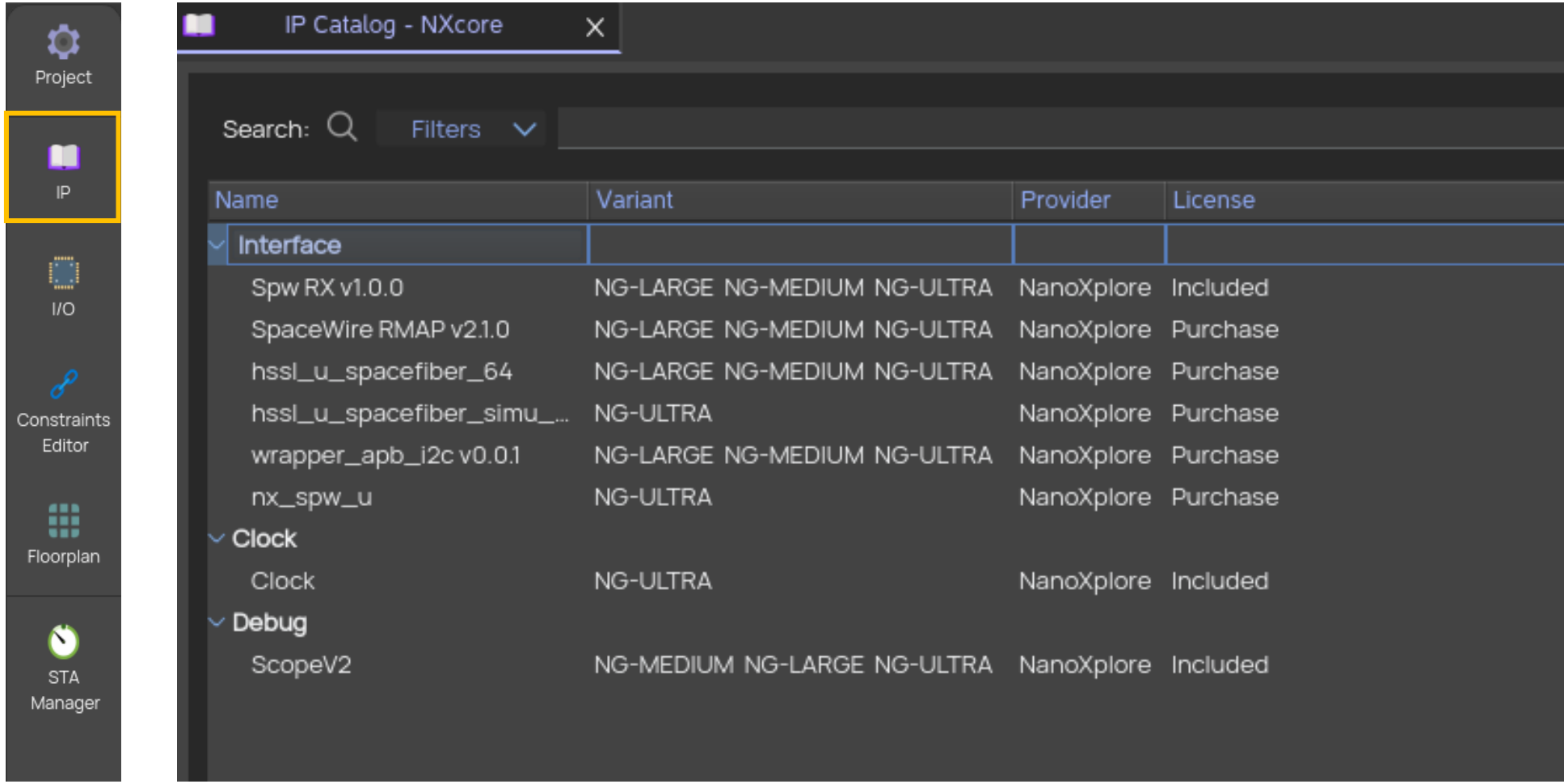
Depending on the step in the flow, it is possible to modify the project settings

- Update options
- Add generic parameters



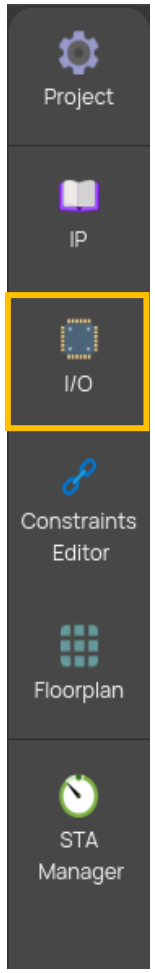
IP Catalog for NG-ULTRA

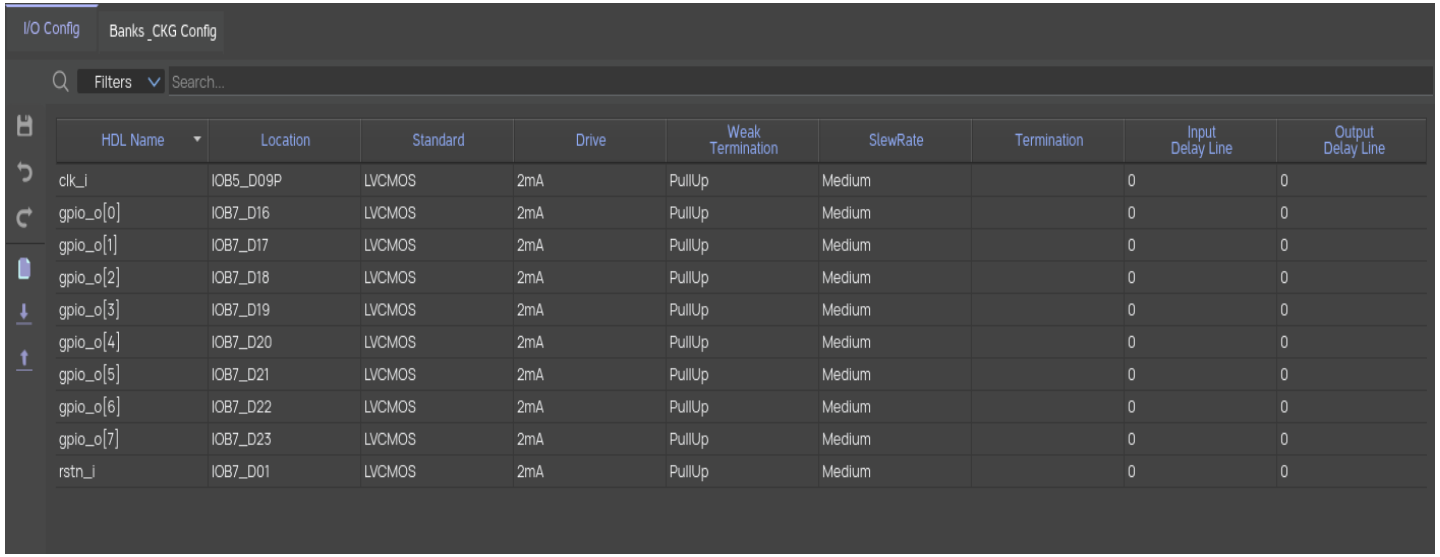
The IP Catalog shows the list of IPs available for a specific device.
In this case the device concerned is NG-ULTRA.



IO Editor

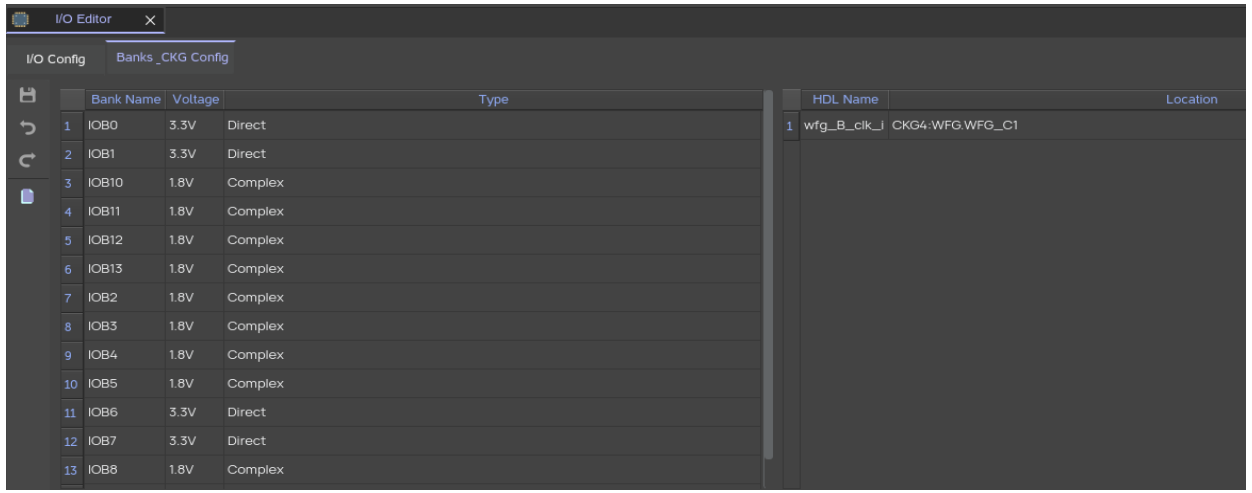
Depending on the step in the flow, you can update the ring configuration for:





HDL Name	Location	Standard	Drive	Weak Termination	SlewRate	Termination	Input Delay Line	Output Delay Line
clk_i	IOB5_D09P	LVCMOS	2mA	PullUp	Medium		0	0
gpio_o[0]	IOB7_D16	LVCMOS	2mA	PullUp	Medium		0	0
gpio_o[1]	IOB7_D17	LVCMOS	2mA	PullUp	Medium		0	0
gpio_o[2]	IOB7_D18	LVCMOS	2mA	PullUp	Medium		0	0
gpio_o[3]	IOB7_D19	LVCMOS	2mA	PullUp	Medium		0	0
gpio_o[4]	IOB7_D20	LVCMOS	2mA	PullUp	Medium		0	0
gpio_o[5]	IOB7_D21	LVCMOS	2mA	PullUp	Medium		0	0
gpio_o[6]	IOB7_D22	LVCMOS	2mA	PullUp	Medium		0	0
gpio_o[7]	IOB7_D23	LVCMOS	2mA	PullUp	Medium		0	0
rstn_i	IOB7_D01	LVCMOS	2mA	PullUp	Medium		0	0

IOs



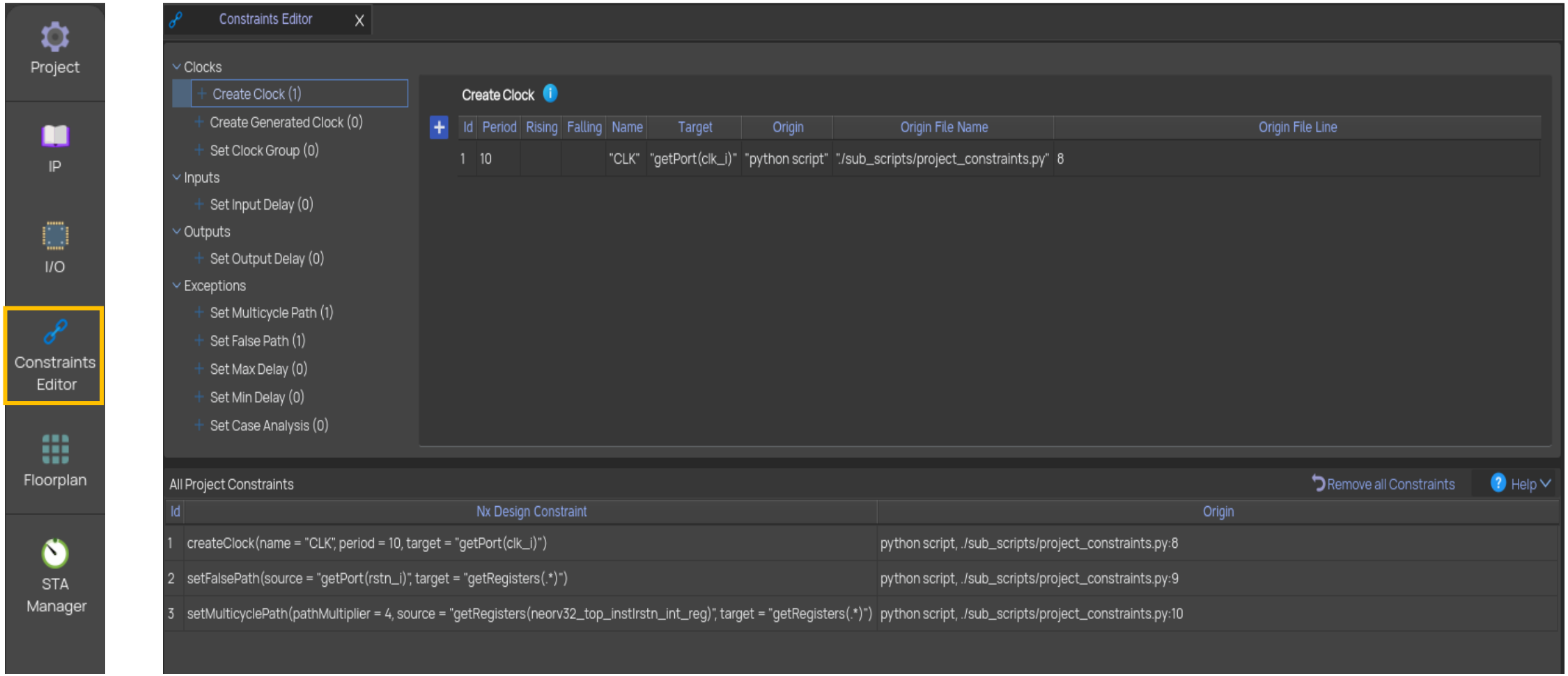
	Bank Name	Voltage	Type
1	IOB0	3.3V	Direct
2	IOB1	3.3V	Direct
3	IOB10	1.8V	Complex
4	IOB11	1.8V	Complex
5	IOB12	1.8V	Complex
6	IOB13	1.8V	Complex
7	IOB2	1.8V	Complex
8	IOB3	1.8V	Complex
9	IOB4	1.8V	Complex
10	IOB5	1.8V	Complex
11	IOB6	3.3V	Direct
12	IOB7	3.3V	Direct
13	IOB8	1.8V	Complex

	HDL Name	Location
1	wfg_B_clk_i	CKG4:WFG.WFG_C1

Banks

Constraints Editor

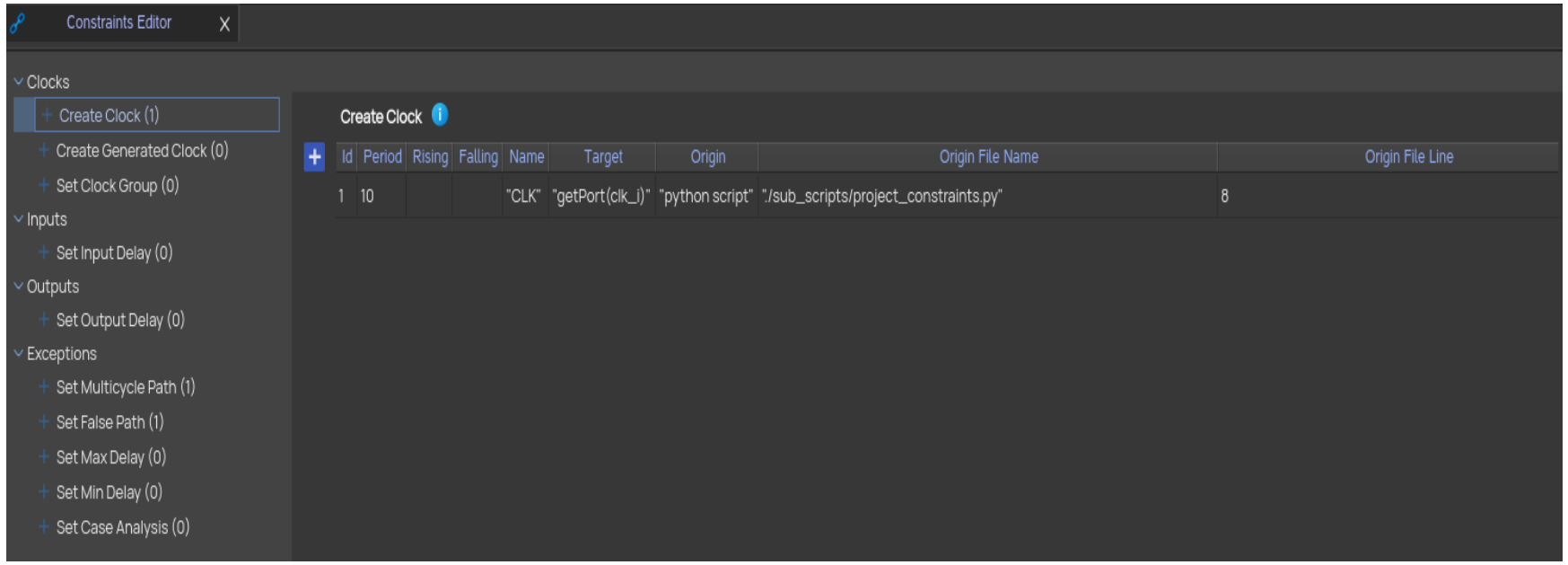
The constraints editor allows you to add or remove timing constraints



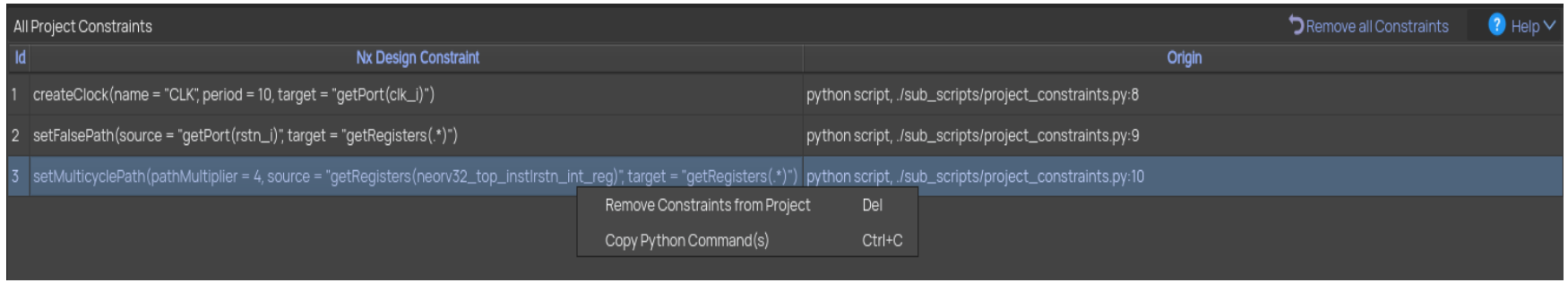
Constraints Editor – Update constraints

You can modify timing constraints before STA launch.

- Create a clock
- Remove a constraint



The constraint in python is available and you can copy it in a NXpython script.

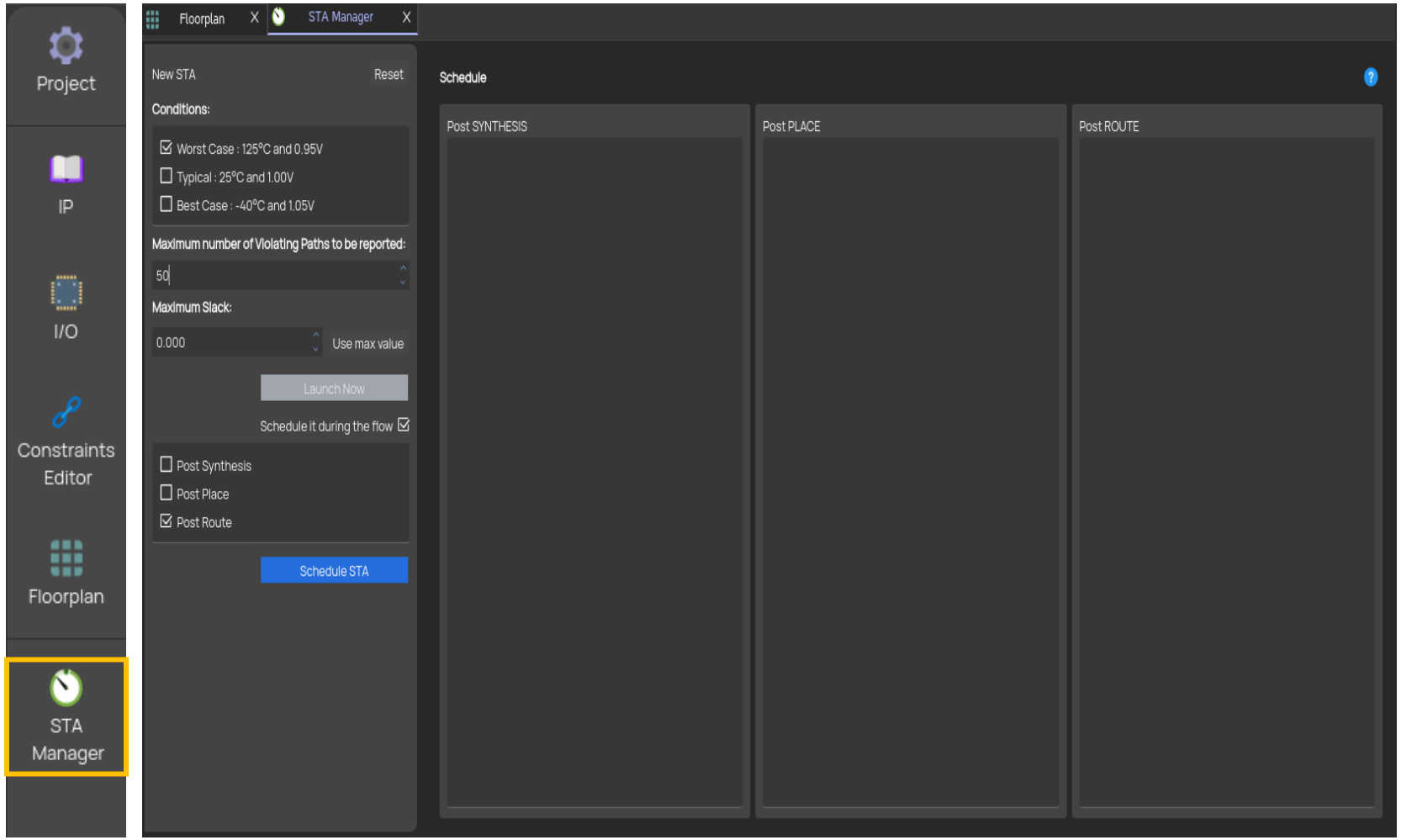


STA Manager

In STA Manager you can:

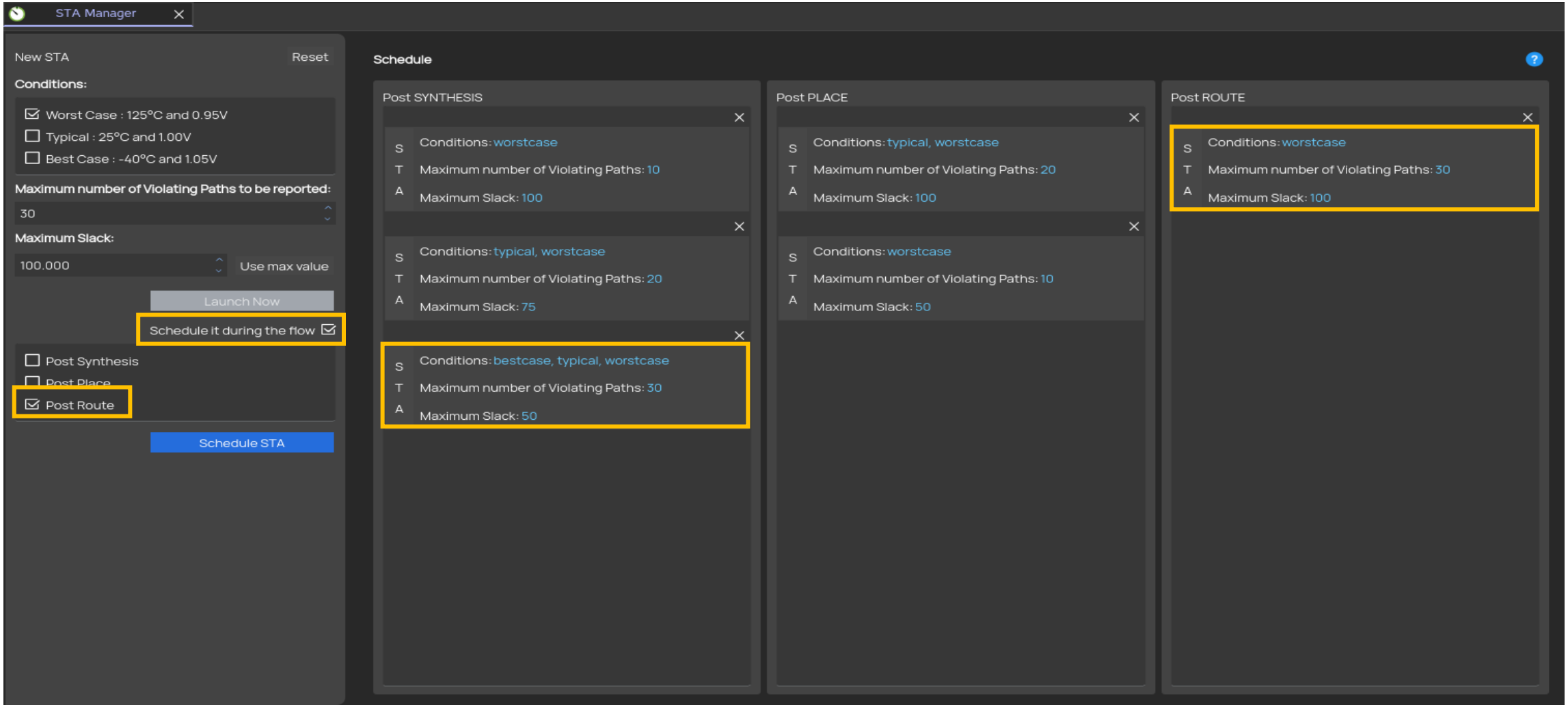
- Schedule STA after each step. Therefore, STA can be launched after
  - Synthesis
  - Place
  - Route

- Specify the STA conditions
  - Worst Case
  - Typical
  - Best Case



STA – Schedule STA

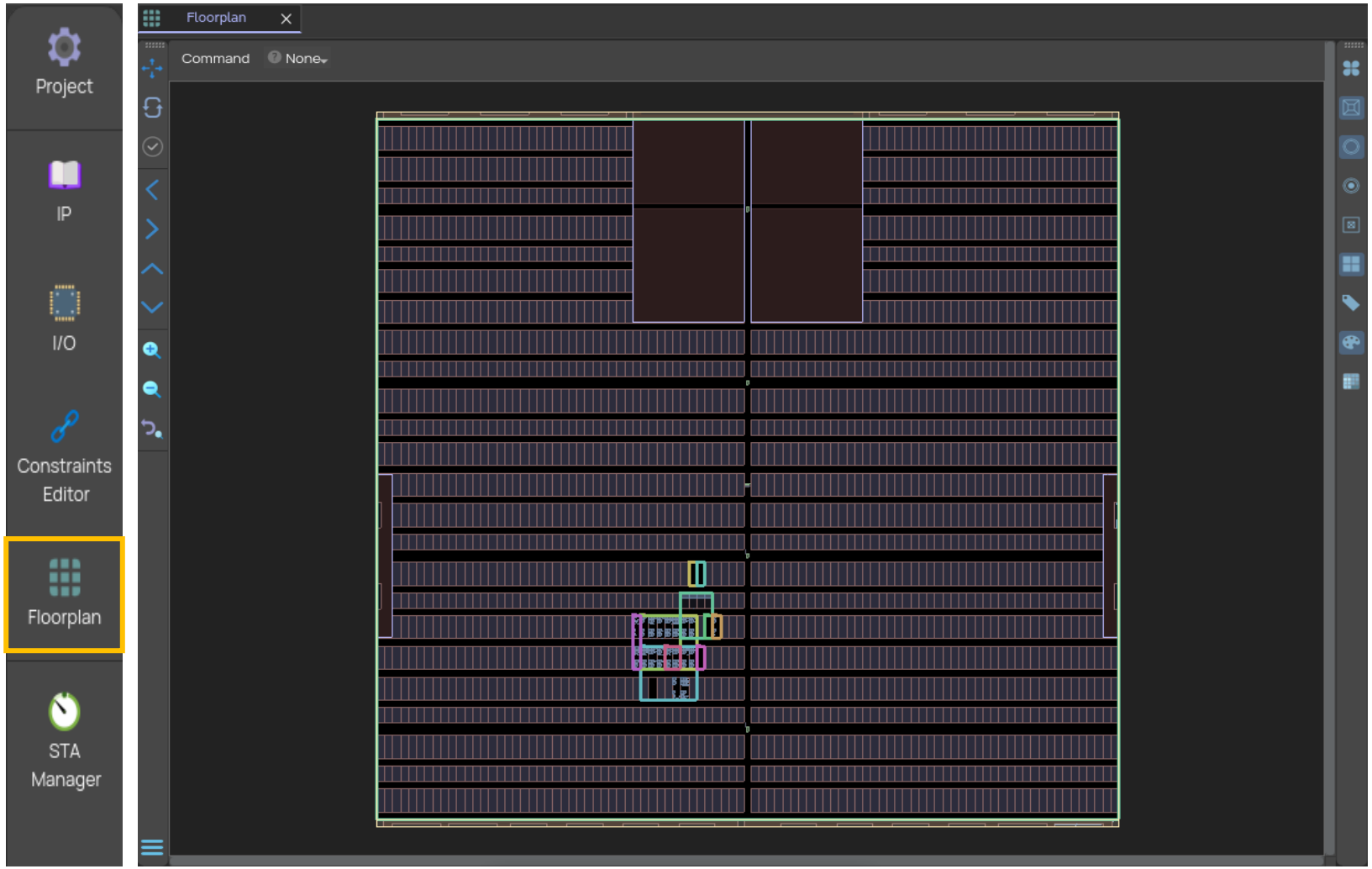
You can schedule as many STAs as you want through the GUI.



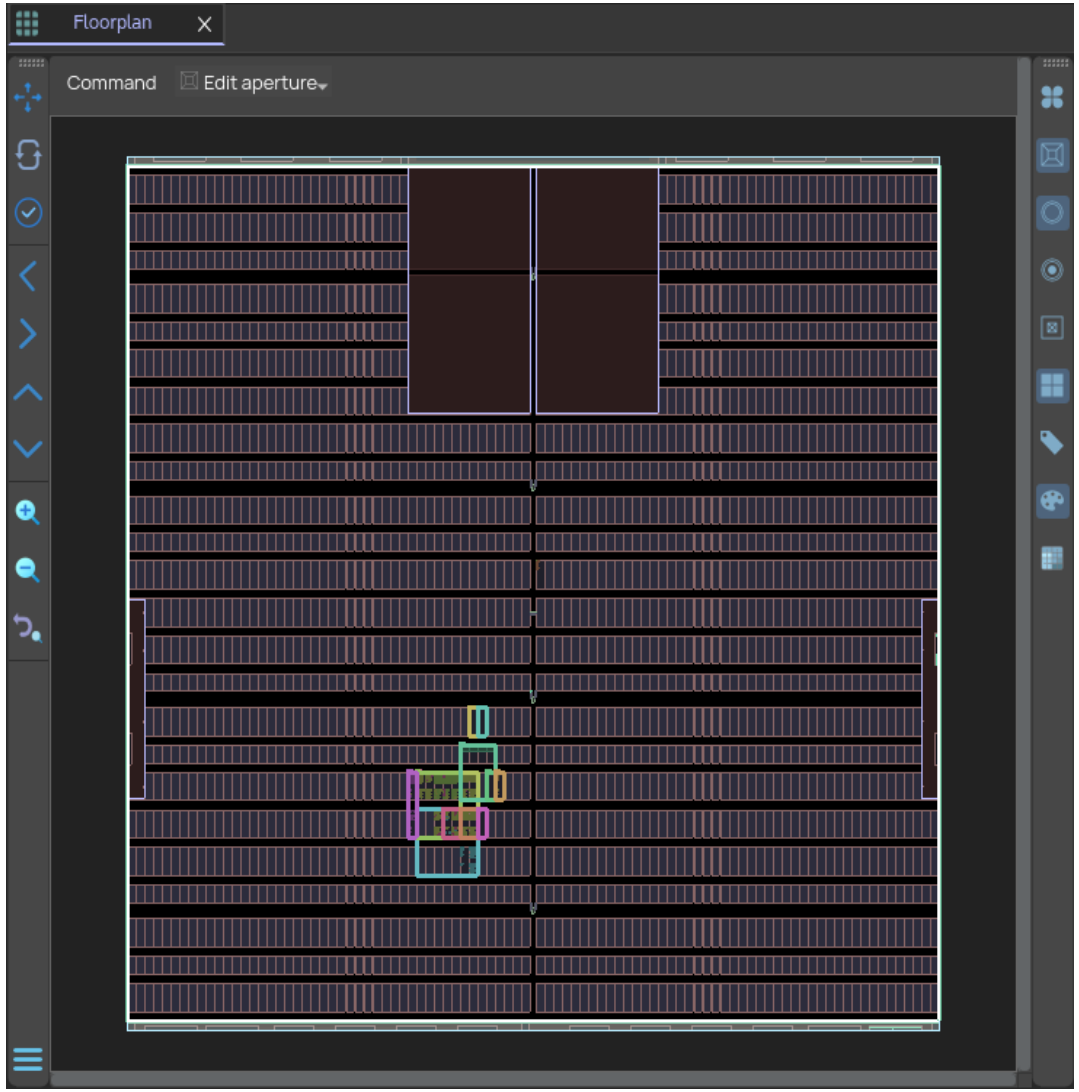
Floorplan Viewer

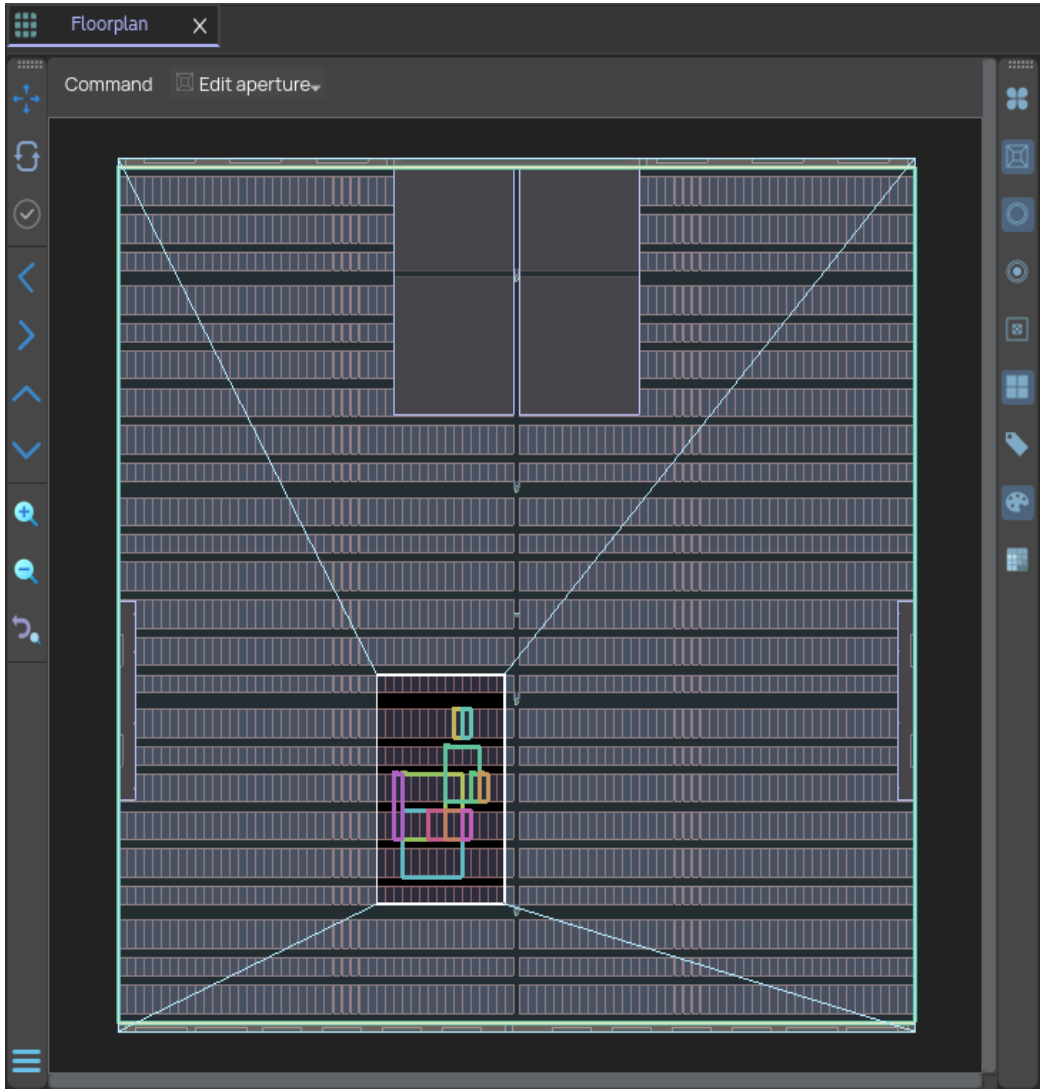
Through the floorplan viewer you can:

- Set global Aperture
- Create/Edit Regions
- View Nets
- View Instances
- View Paths
- View interconnexions between regions
- Display the routing Congestion Map

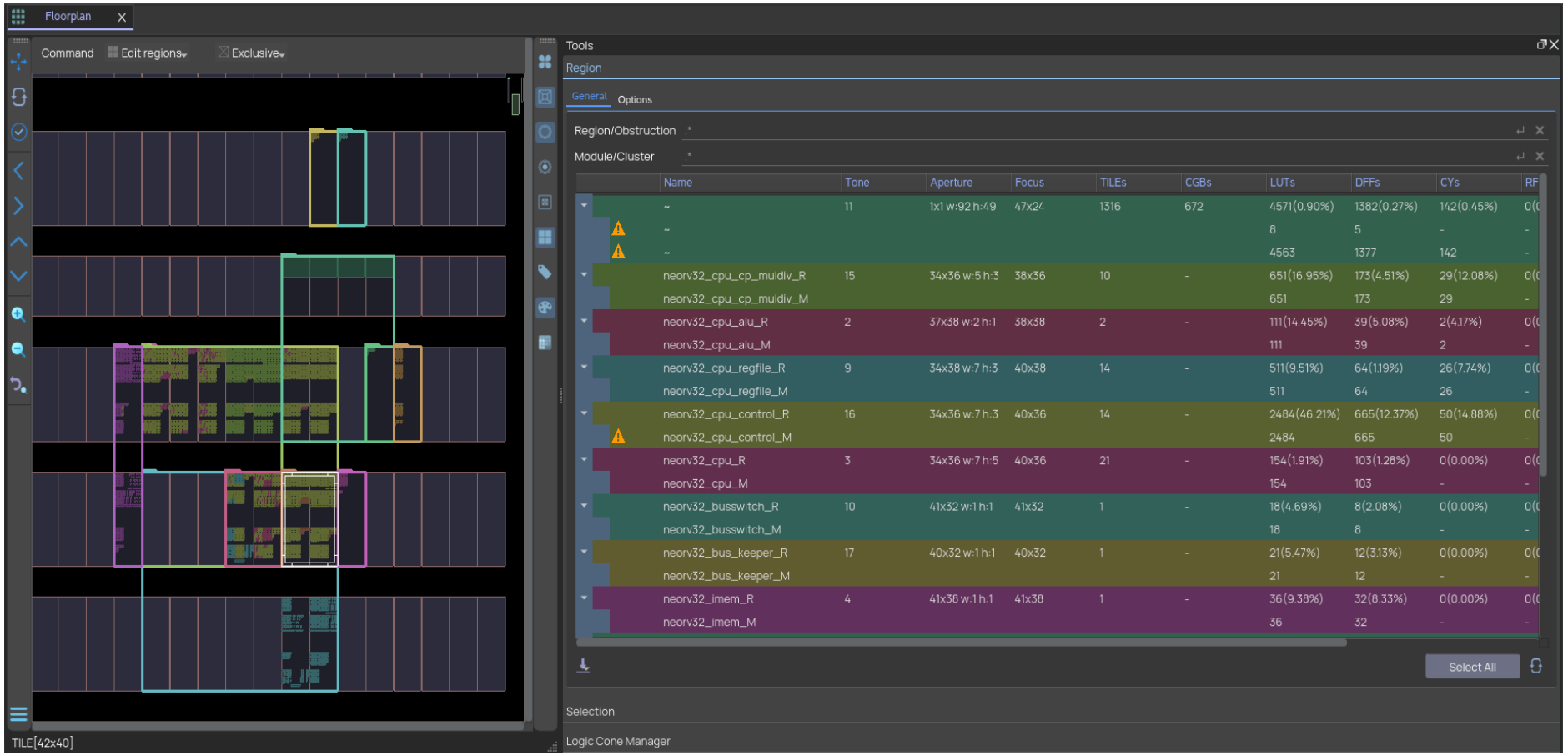


Floorplan – Edit aperture



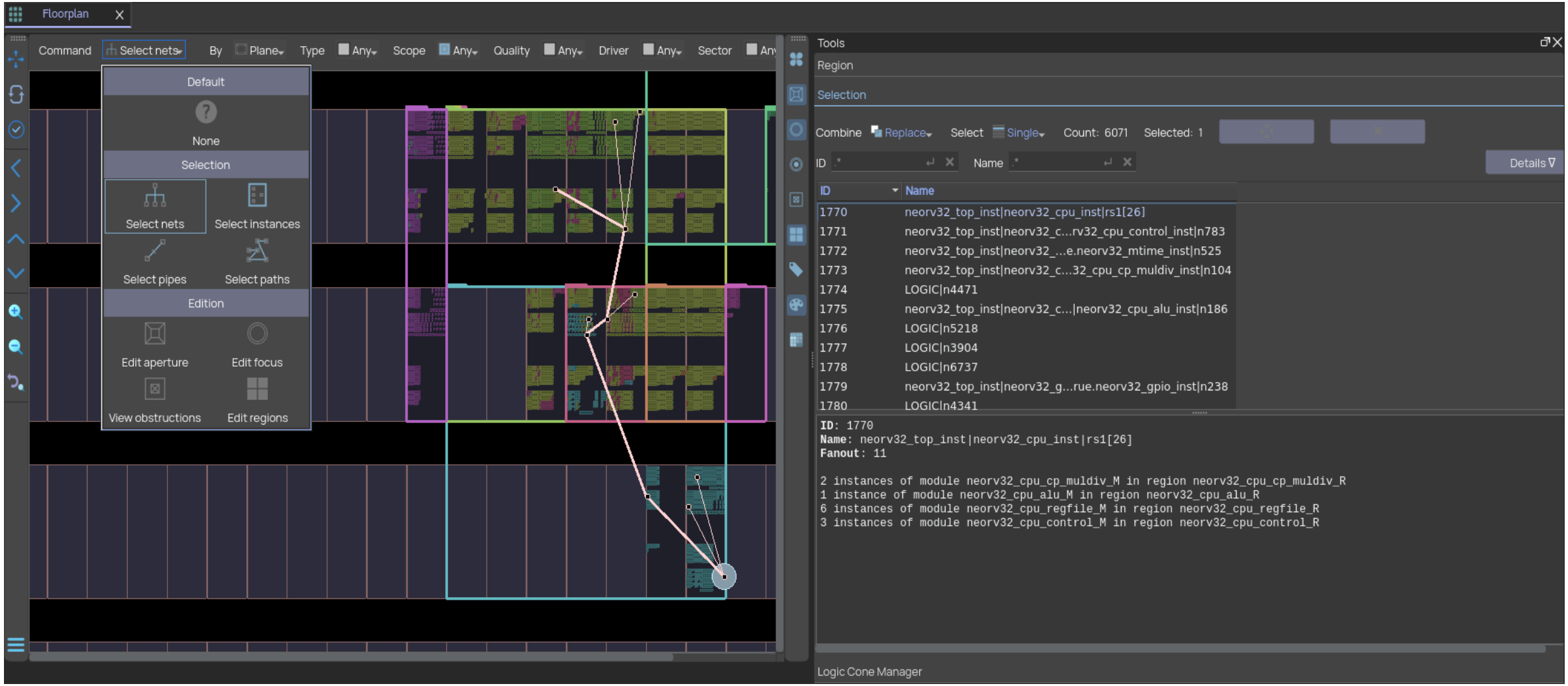


Floorplan - Regions



Name	Tone	Aperture	Focus	TILEs	CGBs	LUTs	DFFs	CYs	RF
~	11	1x1 w:92 h:49	47x24	1316	672	4571(0.90%)	1382(0.27%)	142(0.45%)	0(
~						8	5	-	-
~						4563	1377	142	-
neorv32_cpu_cp_muldiv_R	15	34x36 w:5 h:3	38x36	10	-	651(16.95%)	173(4.51%)	29(12.08%)	0(
neorv32_cpu_cp_muldiv_M						651	173	29	-
neorv32_cpu_alu_R	2	37x38 w:2 h:1	38x38	2	-	111(14.45%)	39(5.08%)	2(4.17%)	0(
neorv32_cpu_alu_M						111	39	2	-
neorv32_cpu_regfile_R	9	34x38 w:7 h:3	40x38	14	-	511(9.51%)	64(1.19%)	26(7.74%)	0(
neorv32_cpu_regfile_M						511	64	26	-
neorv32_cpu_control_R	16	34x36 w:7 h:3	40x36	14	-	2484(46.21%)	665(12.37%)	50(14.88%)	0(
neorv32_cpu_control_M						2484	665	50	-
neorv32_cpu_R	3	34x36 w:7 h:5	40x36	21	-	154(1.91%)	103(1.28%)	0(0.00%)	0(
neorv32_cpu_M						154	103	-	-
neorv32_busswitch_R	10	41x32 w:1 h:1	41x32	1	-	18(4.69%)	8(2.08%)	0(0.00%)	0(
neorv32_busswitch_M						18	8	-	-
neorv32_bus_keeper_R	17	40x32 w:1 h:1	40x32	1	-	21(5.47%)	12(3.13%)	0(0.00%)	0(
neorv32_bus_keeper_M						21	12	-	-
neorv32_imem_R	4	41x38 w:1 h:1	41x38	1	-	36(9.38%)	32(8.33%)	0(0.00%)	0(
neorv32_imem_M						36	32	-	-

Floorplan – Select nets



Floorplan – Select instances

Floorplan

Command | Select instances | By | Plane | Category | Any

Default
None
Selection
Select nets | Select instances
Select pipes | Select paths
Edition
Edit aperture | Edit focus
View obstructions | Edit regions

Tools
Region
Selection

Combine | Replace | Select | Single | Count: 9404 | Selected: 1 | Locking | Details

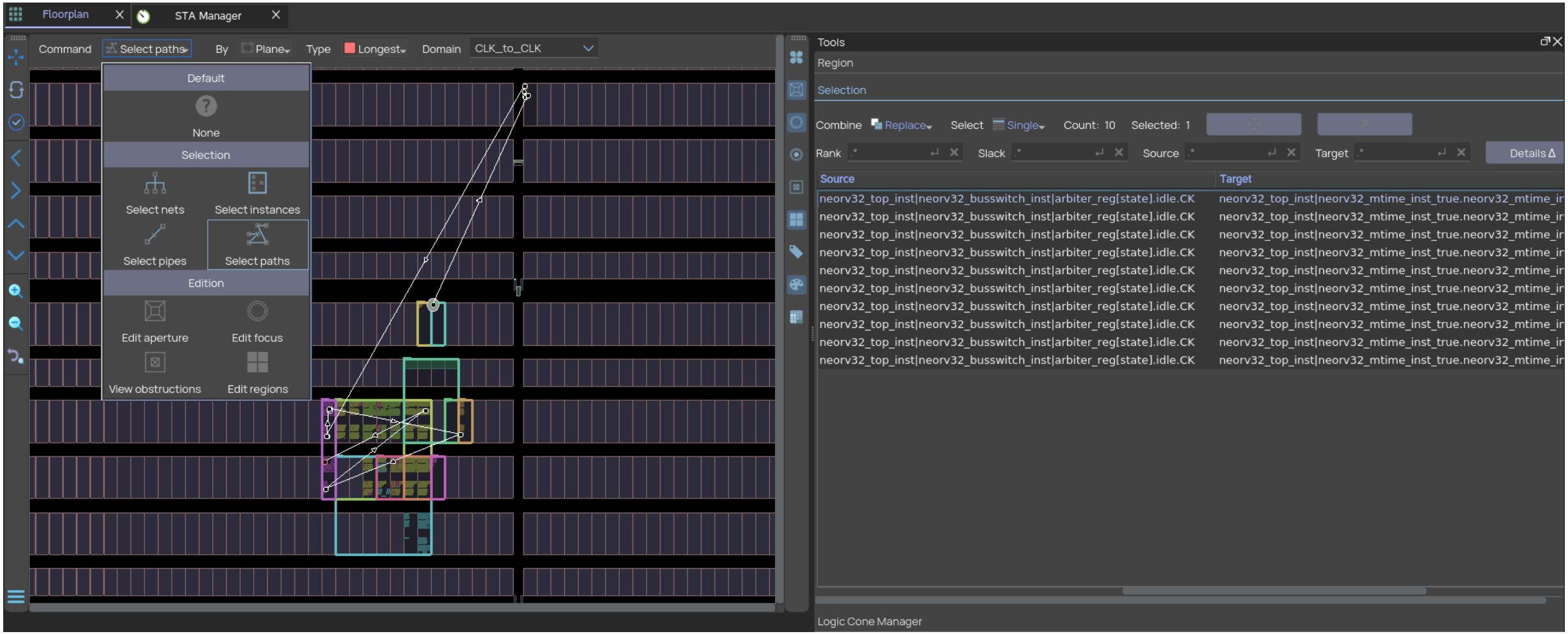
ID .* | Type .* | Location .* | Name .*

ID	Type	Location	Name
9403	LUT		neorv32_top_inst|neorv32_cpu_inst|neorv32_cpu_control_inst|trap_ctrl[env_start]_reg~csc~lu
9401	LUT		neorv32_top_inst|neorv32_cpu_inst|neorv32_cpu_control_inst|ctrl_reg[35]~csc~lut
9399	LUT		neorv32_top_inst|neorv32_cpu_inst|neorv32_c...t|execute_engine_reg[state].execute~csc~lu
9397	LUT		neorv32_top_inst|neorv32_cpu_inst|neorv32_cp...ntrol_inst|fetch_engine[restart]_reg~csc~lut
9395	LUT		neorv32_top_inst|neorv32_cpu_inst|neorv32_cp...p_muldiv_inst|ctrl_reg[state].s_idle~csc~lut
9393	LUT		LOGIC|lut_2668[D]~csc~lut
9391	LUT		neorv32_top_inst|neorv32_cpu_inst|neorv32_cp...er[1].prefetch_buffer_inst|lut_42[D]~csc~lut
9389	LUT		LOGIC|lut_3566[D]~csc~lut
9387	LUT		neorv32_top_inst|neorv32_cpu_inst|neorv32_cp...er[0].prefetch_buffer_inst|lut_41[D]~csc~lut
9385	LUT		neorv32_top_inst|neorv32_cpu_inst|neorv32_cpu_control_inst|lut_29[D]~csc~lut

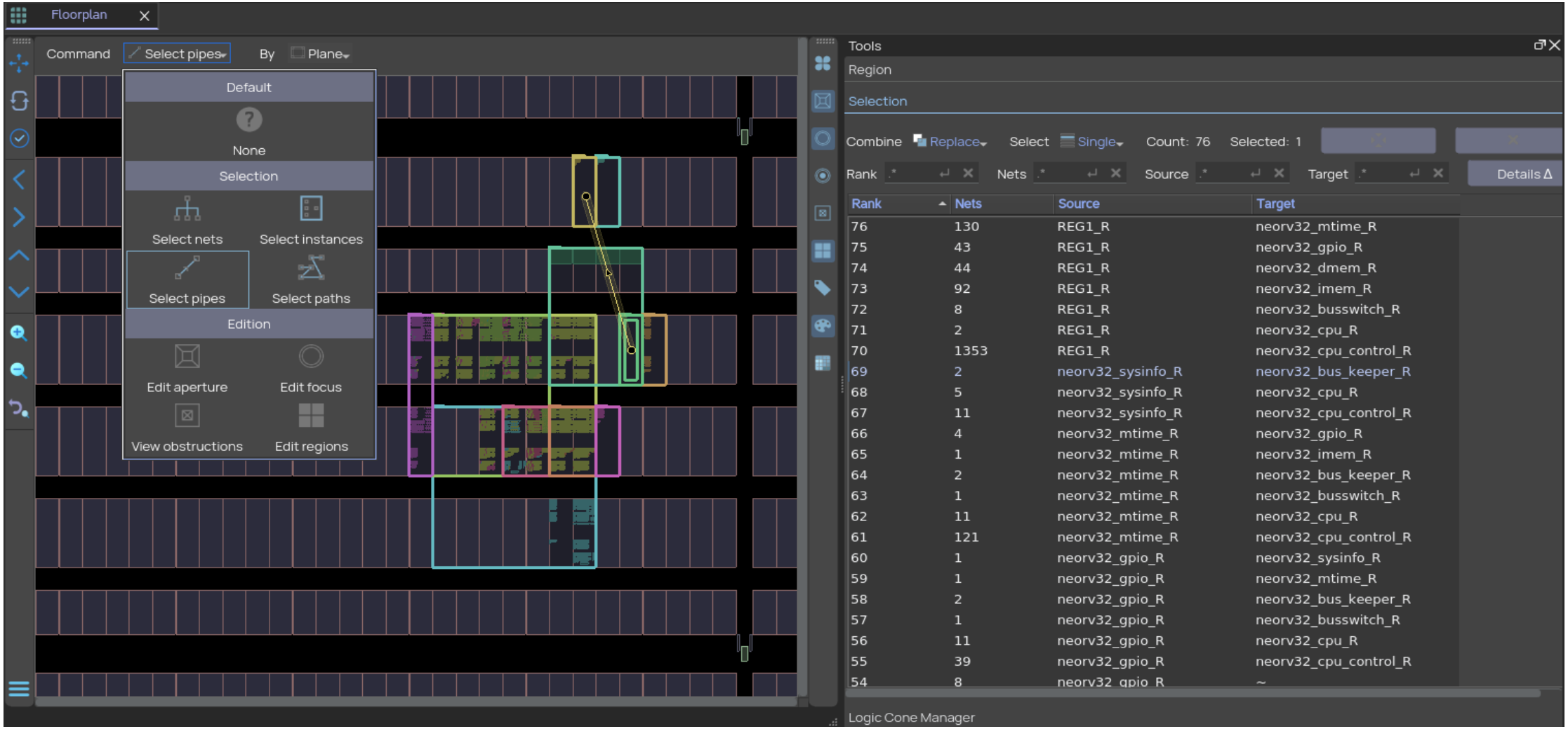
```
Name: neorv32_top_inst|neorv32_cpu_inst|neorv32_cpu_control_inst|trap_ctrl[env_start]_reg~csc~lut
Module: neorv32_cpu_control_M
Region: neorv32_cpu_control_R
Location: TILE[36x38]:S3.LUT65

Equation: O <= I1

I1: neorv32_top_inst|neorv32_cpu_inst|neorv32_cpu_control_inst|trap_ctrl[env_start]
I2: '0'
I3: '0'
I4: '0'
O: neorv32_top_inst|neorv32_cpu_inst|neorv32_cpu_control_inst|trap_ctrl[env_start]
```

Logic Cone Manager

Floorplan – Select paths



Floorplan – Select pipes



Synthesis

NanoXplore

Synthesis

FPGA synthesis is the process of converting a design into Netlist. From HDL code (VHDL, Verilog, etc.) to dedicated resources into FPGA.

Synthesis 1

Synthesis 2

Synthesis 3

PLACE

- Analysis & Elaboration
  - *Verilog IEEE 1164*
  - *VHDL IEEE-1076 standard (2008, 1993, 1987)*

- Mapping into NX resources
  (NX_LUT, NX_DFF, NX_RAM, NX_DSP…)
- Replacing BlackBoxes by Preplaced IPs

Post-Synthesis STA

Synthesis – Files Generation

- Logs
  - `general.log` : contains all messages printed during the Synthesis step

- Reports
  - `hierarchy.rpt` : Reports the hierarchy of the design
  - `constraints.rpt` : Reports the unused constraints and the constraints that were not applied
  - `memories.rpt` : Reports the information about memories used in the design
  - `operators.rpt` : Reports all operators (adder, subtractor, multiplier, comparator…) used in the design

- You can generate at synthesis stage:
  - Netlist in VHDL or Verilog

All synthesis steps are saved in different .nym files which could be loaded by the GUI and Nxpython method.

Design resources

The design resources is available after Synthesis.

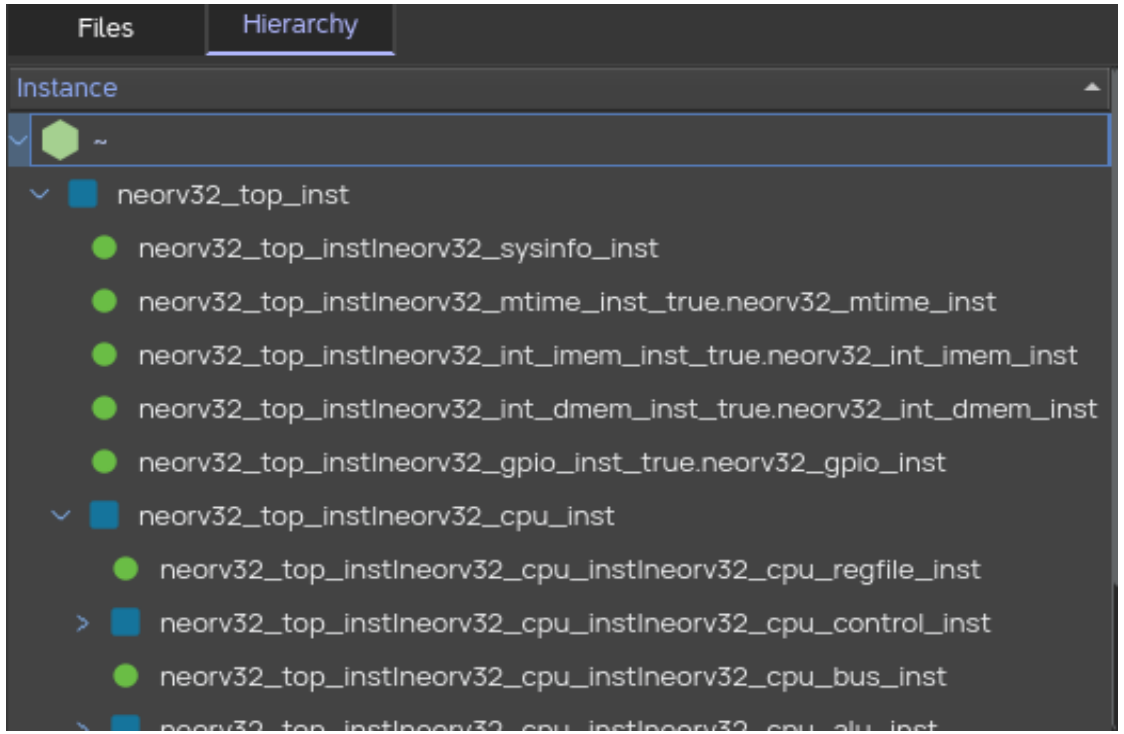
You can find design resources information in `general.log`.

```
Reporting instances at state 'Synthesized':
  +-----------------+-----------------+-------------+----------------+------------+------------+--------+------------+------------+-----------+----------+---------+-----------+
  |                 |                 |             |    1 - bit     | Register   |   Cross    | Clock  |   Clock    |  Digital   |  Memory   |          |         |           |
  |     4-LUT       |      DFF        |    XLUT     |     Carry      |   file     |  domain    | Buffer |  switch    |  signal    |  block    |   WFG    |   PLL   |    GCK    |
  |                 |                 |             |                |   block    |   clock    |        |            | processor  |           |          |         |           |
  +-----------------+-----------------+-------------+----------------+------------+------------+--------+------------+------------+-----------+----------+---------+-----------+
  | 3588/505344 (1%) | 1377/505344 (1%) | 0/31584 (0%) | 531/126336 (1%) | 4/2632 (1%) | 0/2632 (0%) |   0    | 0/5264 (0%) | 0/1344 (0%) | 4/672 (1%) | 0/70 (0%) | 0/7 (0%) | 0/160 (0%) |
  +-----------------+-----------------+-------------+----------------+------------+------------+--------+------------+------------+-----------+----------+---------+-----------+
```

Hierarchy

The design hierarchy is available after Synthesis.

```
----------------------------
HDL Modules Hierarchy Detail
----------------------------

~                       neorv32_test_setup_approm [ ~ ]
                        Resources:
                          NX_LUT : 3587
                          NX_DFF : 1377
                          NX_CY : 142
                          NX_RFB_U : 2
                          NX_RAM : 4
                          NX_IOB : 10
                          NX_BFR : 19
                          NX_WFG_U : 1
~                       |-> neorv32#neorv32_top(X8381227E) [ neorv32_top_inst ]
                        |-> Resources:
                        |->   NX_LUT : 2546
                        |->   NX_DFF : 1377
                        |->   NX_CY : 142
                        |->   NX_RFB_U : 2
                        |->   NX_RAM : 4
```



Mapping – Operators

Detailed info of each **operator**.

Operators (8) | Operators Mapping (21) | Notifications (0)

Search...

Name		
LessThan_32u_32u		
Instances (2)		
Name	File Origin	Line Origin
LessThan_cmp_lo_ge	../src/neorv32_mtime.vhd	196
LessThan_cmp_hi_gt	../src/neorv32_mtime.vhd	198
LessThan_33u_33u		
Instances (1)		
Name	File Origin	Line Origin
LessThan_cmp_o	../src/neorv32_cpu_alu.vhd	104
add_2u_2u		
Instances (3)		
Name	File Origin	Line Origin
add_L113	../src/neorv32_fifo.vhd	113
add_L119	../src/neorv32_fifo.vhd	119
add_level_diff	../src/neorv32_fifo.vhd	132
add_4u_4u		
Instances (1)		
Name	File Origin	Line Origin
add_L179	../src/neorv32_bus_keeper.vhd	179

```
-----------------------------------------
- Operator Cells analysis                -
-----------------------------------------

  Operator 'LessThan_32u_32u'
        : LessThan_cmp_lo_ge          (line 196 in ../src/neorv32_mtime.vhd)
        : LessThan_cmp_hi_gt          (line 198 in ../src/neorv32_mtime.vhd)

  Operator 'LessThan_33u_33u'
        : LessThan_cmp_o    (line 104 in ../src/neorv32_cpu_alu.vhd)

  Operator 'add_2u_2u'
        : add_L113          (line 113 in ../src/neorv32_fifo.vhd)
        : add_L119          (line 119 in ../src/neorv32_fifo.vhd)
        : add_level_diff    (line 132 in ../src/neorv32_fifo.vhd)

  Operator 'add_4u_4u'
        : add_L179          (line 179 in ../src/neorv32_bus_keeper.vhd)
```

Mapping – Memories

Detailed info of each **memory**.

Mem (17) | Mem Mapped (5) | Notifications (0)

Search...

Name	Type	Depth	Width	Instance Name	Read Type	Async	Generated
neorv32#neorv32_cpu_regfile(X5BDBBAD2)_reg_file	Ram	32	32	reg_file	ReadAfterWrite	false	true

Ports (2)

Addr Size	Din Size	Dout Size	Dout Type	Id	Wr
5	32	32	RS	0	true
5	0	32	RS	1	false

```
---------------------------------------------
- Memory Cells analysis                     -
---------------------------------------------

  HDL 'neorv32#neorv32_cpu_regfile(X5BDBBAD2)_reg_file' Raw description seems to be 'Read after Write'
       Physical implementation in RF/RAM may not follow it exactly without additional logic

  Ram 'neorv32#neorv32_cpu_regfile(X5BDBBAD2)_reg_file' Analysis:
       Port 0 :
                 Slc  Size: 0
                 Addr Size: 5
                 Din  Size: 32 (W)
                 Dout Size: 32 (RS)
       Port 1 :
                 Slc  Size: 0
                 Addr Size: 5
                 Din  Size: 0
                 Dout Size: 32 (RS)
       Array Depth 32
       Array Width 32
```

Mapping Directive

You can **force mapping** in specific resources

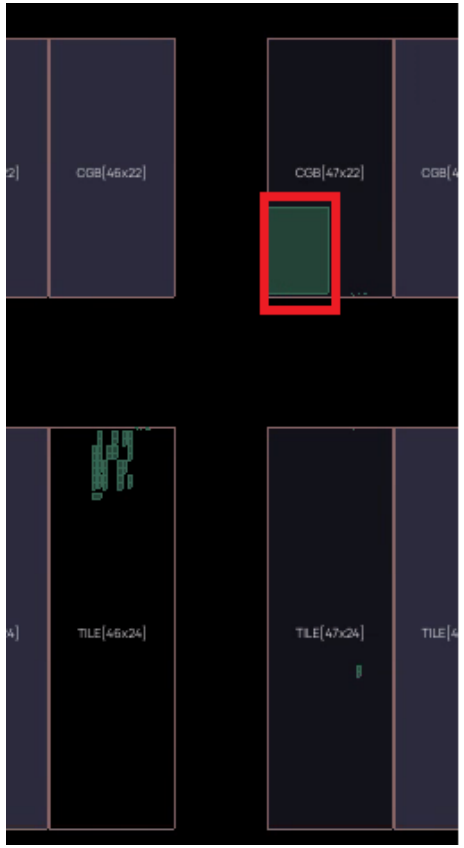
	Type	Mapping targets
Operators	ADD	LUT, CY, DSP
	LTN	LUT, CY, DSP
	MUL	CY,DSP
Memories	RAM	RF, RAM, RAM_ECC, DFF
	ROM	LUT, RF, RAM

Mapping Directive – Operators

The floorplan view of same **operator cell** mapped into different resources.

Adder mapped into CY

Adder mapped into DSP

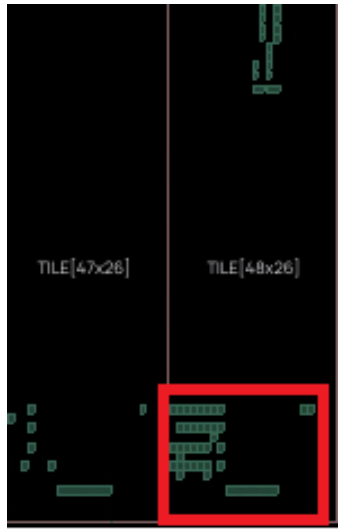


Adder mapped into LUT

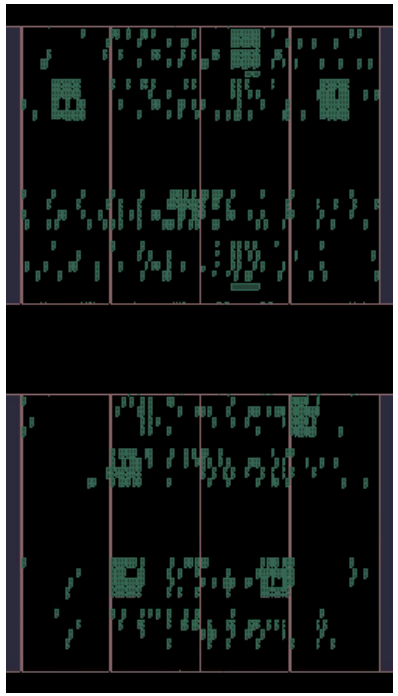
Mapping Directive – Memories

The floorplan view of **memory cell** mapped into different resources.

RAM mapped into RF

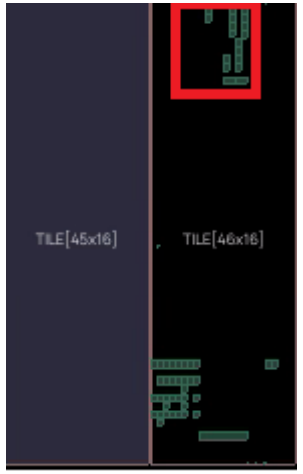


RAM mapped into DFF



RAM mapped into RAM block

ROM mapped into LUT



Post-Synthesis STA

Post-synthesis STA is done at the end of synthesis

This static timing analysis helps to know if the frequency requested can be reached.

```
|   Static Timing Analysis Conditions: typical
|   Voltage: 1.20V
|   Temperature: 25°C
|
|   Summary of DOMAIN_clk25m_to_clk25m
|     +-------------------------------------+-----------------------+-------------------------------------------+-------------------------------------------+
|     |                Domain               |       Frequency       |           Hold/Removal Summary            |          Setup/Recovery Summary           |
|     +-------------------+-----------------+----------+------------+----------+---------------+-----------------+----------+---------------+-----------------+
|     |      Source       |      Target     | Required |   Actual   |   Slack  | Minimum Data  | Minimum Required |   Slack  | Maximum Data  | Maximum Required |
|     |                   |                 |          |            |          | Arrival Time  |   Relationship   |          | Arrival Time  |   Relationship   |
|     +-------------------+-----------------+----------+------------+----------+---------------+-----------------+----------+---------------+-----------------+
|     | clk25m (Falling)  | clk25m (Rising) |        - |          - | 15.165ns |      2.246ns  |       -12.919ns  |  9.323ns |      3.596ns  |         12.919ns |
|     | clk25m (Rising)   | clk25m (Falling)|        - | 38.703 MHz | 14.310ns |      1.391ns  |       -12.919ns  |      0ps |     12.919ns  |         12.919ns |
|     | clk25m (Rising)   | clk25m (Rising) |        - |          - |    787ps |        787ps  |              0ps | 10.945ns |     14.893ns  |         25.838ns |
|     +-------------------+-----------------+----------+------------+----------+---------------+-----------------+----------+---------------+-----------------+
|     | Total                               |                                                                                                                 3 |
|     +-------------------------------------+-----------------------------------------------------------------------------------------------------------------+
|
|   Reporting longest paths for DOMAIN_clk25m_to_clk25m
|     +--------+---------+-------------------------------------------------------------------------------+-------------------------------------+------------+------------+----------------+--------+--------+
|     |   No.  |  Slack  |                                   Source                                      |                Target               | Data Delay | Clock Skew | Setup/Recovery |  Depth |  Note  |
|     +--------+---------+-------------------------------------------------------------------------------+-------------------------------------+------------+------------+----------------+--------+--------+
|     |      1 |     0ps | Pin: inst_AHBLite|AHBLite_0|matrix4x16|masterstage_0|SDATASELInt_reg[4].CK | Pin: inst_CoreMemCtrl|iSRAMWEN_reg.I |   12.682ns |      -10ps |          227ps |     14 |   (RF) |
|     |      2 |    14ps | Pin: inst_AHBLite|AHBLite_0|matrix4x16|masterstage_0|SDATASELInt_reg[5].CK | Pin: inst_CoreMemCtrl|iSRAMWEN_reg.I |   12.668ns |      -10ps |          227ps |     14 |   (RF) |
|     |      3 |    80ps | Pin: inst_AHBLite|AHBLite_0|matrix4x16|masterstage_0|SDATASELInt_reg[6].CK | Pin: inst_CoreMemCtrl|iSRAMWEN_reg.I |   12.602ns |      -10ps |          227ps |     14 |   (RF) |
|     |      4 |    97ps | Pin: inst_AHBLite|AHBLite_0|matrix4x16|masterstage_0|SDATASELInt_reg[7].CK | Pin: inst_CoreMemCtrl|iSRAMWEN_reg.I |   12.585ns |      -10ps |          227ps |     14 |   (RF) |
|     |      5 | 1.041ns | Pin: inst_AHBLite|AHBLite_0|matrix4x16|masterstage_0|SDATASELInt_reg[1].CK | Pin: inst_CoreMemCtrl|iSRAMWEN_reg.I |   11.641ns |      -10ps |          227ps |     13 |   (RF) |
|     |      6 | 1.041ns | Pin: inst_AHBLite|AHBLite_0|matrix4x16|masterstage_0|SDATASELInt_reg[1].CK | Pin: inst_CoreMemCtrl|iSRAMWEN_reg.I |   11.641ns |      -10ps |          227ps |     13 |   (RF) |
|     |      7 | 1.055ns | Pin: inst_AHBLite|AHBLite_0|matrix4x16|masterstage_0|SDATASELInt_reg[2].CK | Pin: inst_CoreMemCtrl|iSRAMWEN_reg.I |   11.627ns |      -10ps |          227ps |     13 |   (RF) |
|     |      8 | 1.107ns | Pin: inst_AHBLite|AHBLite_0|matrix4x16|masterstage_0|SDATASELInt_reg[2].CK | Pin: inst_CoreMemCtrl|iSRAMWEN_reg.I |   11.575ns |      -10ps |          227ps |     13 |   (RF) |
|     |      9 | 1.107ns | Pin: inst_AHBLite|AHBLite_0|matrix4x16|masterstage_0|SDATASELInt_reg[4].CK | Pin: inst_CoreMemCtrl|iSRAMWEN_reg.I |   11.575ns |      -10ps |          227ps |     13 |   (RF) |
|     |     10 | 1.121ns | Pin: inst_AHBLite|AHBLite_0|matrix4x16|masterstage_0|SDATASELInt_reg[5].CK | Pin: inst_CoreMemCtrl|iSRAMWEN_reg.I |   11.561ns |      -10ps |          227ps |     13 |   (RF) |
|     +--------+---------+-------------------------------------------------------------------------------+-------------------------------------+------------+------------+----------------+--------+--------+
|     | Total                                                                                                                                                                  10 |        |
|     +-------------------------------------------------------------------------------------------------------------------------------------------------------------------+--------+
|
```

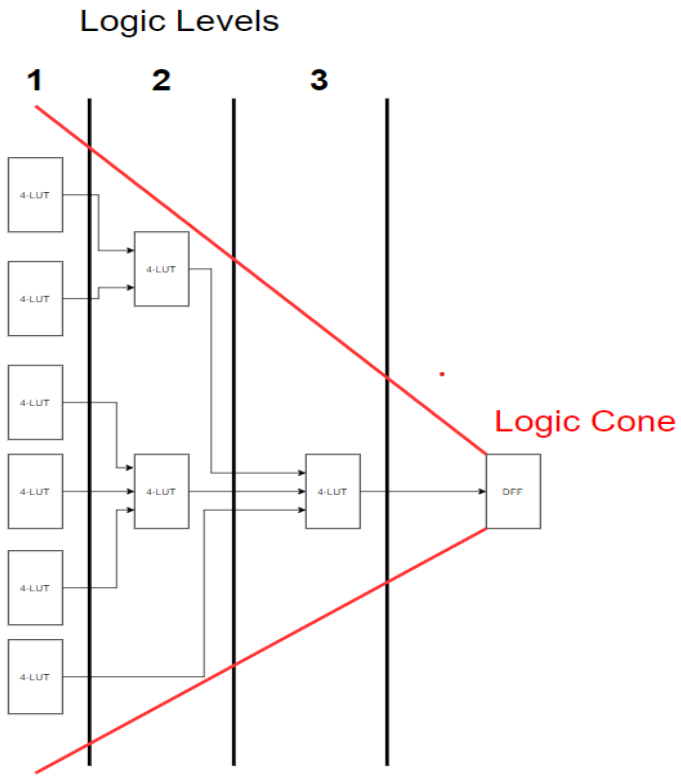
Logic Cone

Logic levels depend on the capacity of elements to perform an operation.

The lower the operational efficiency of a resource is, the higher its logic level will be high.

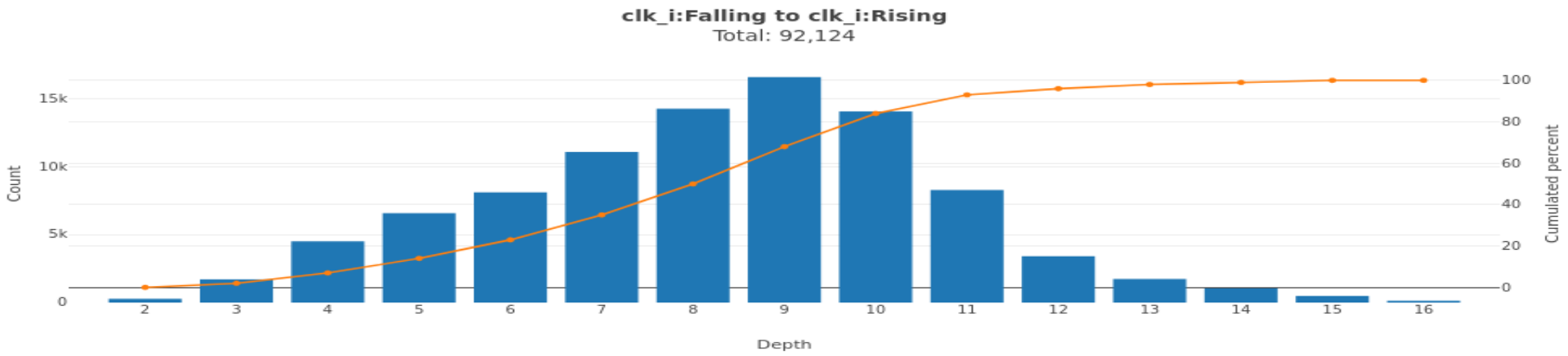
This creates **Logical depth**.

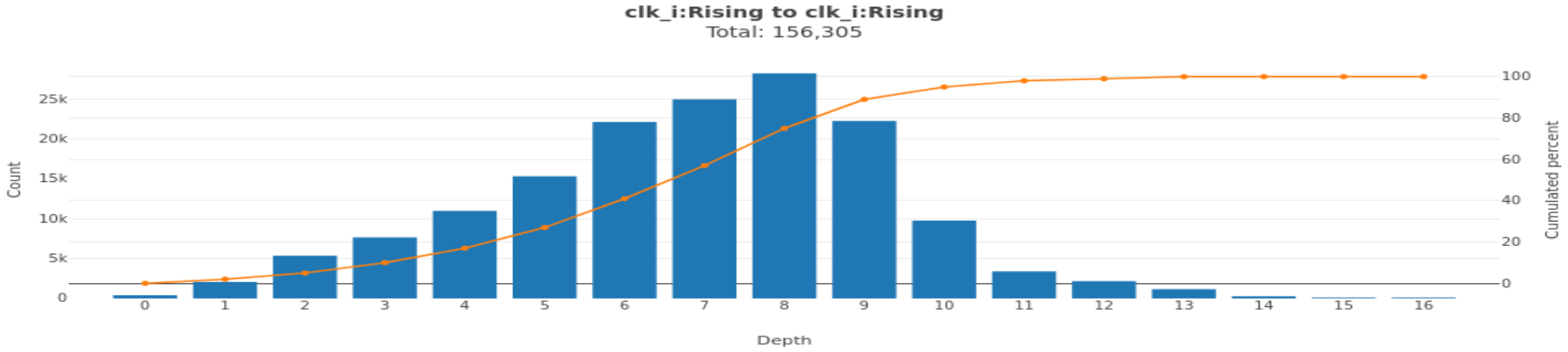
All the elements interconnected to generate one bit create one **Logic Cone**.



Design Complexity

Design complexity charts illustrates repartition of paths based on logic depth by clock domain.





Place

NanoXplore

Place

SYNTHESIS

Place 1

- Apply placement constraints

Place 2

- Place Lowskew Network

Place 3

- Place cells
- Place regions
- Clustering

Place 4

- Post-Place
- Optimize the placement by area

Place 5

- Pre-route

ROUTE

Place – Files Generation

- Logs
  - `general.log` : contains all messages printed during the Place step.

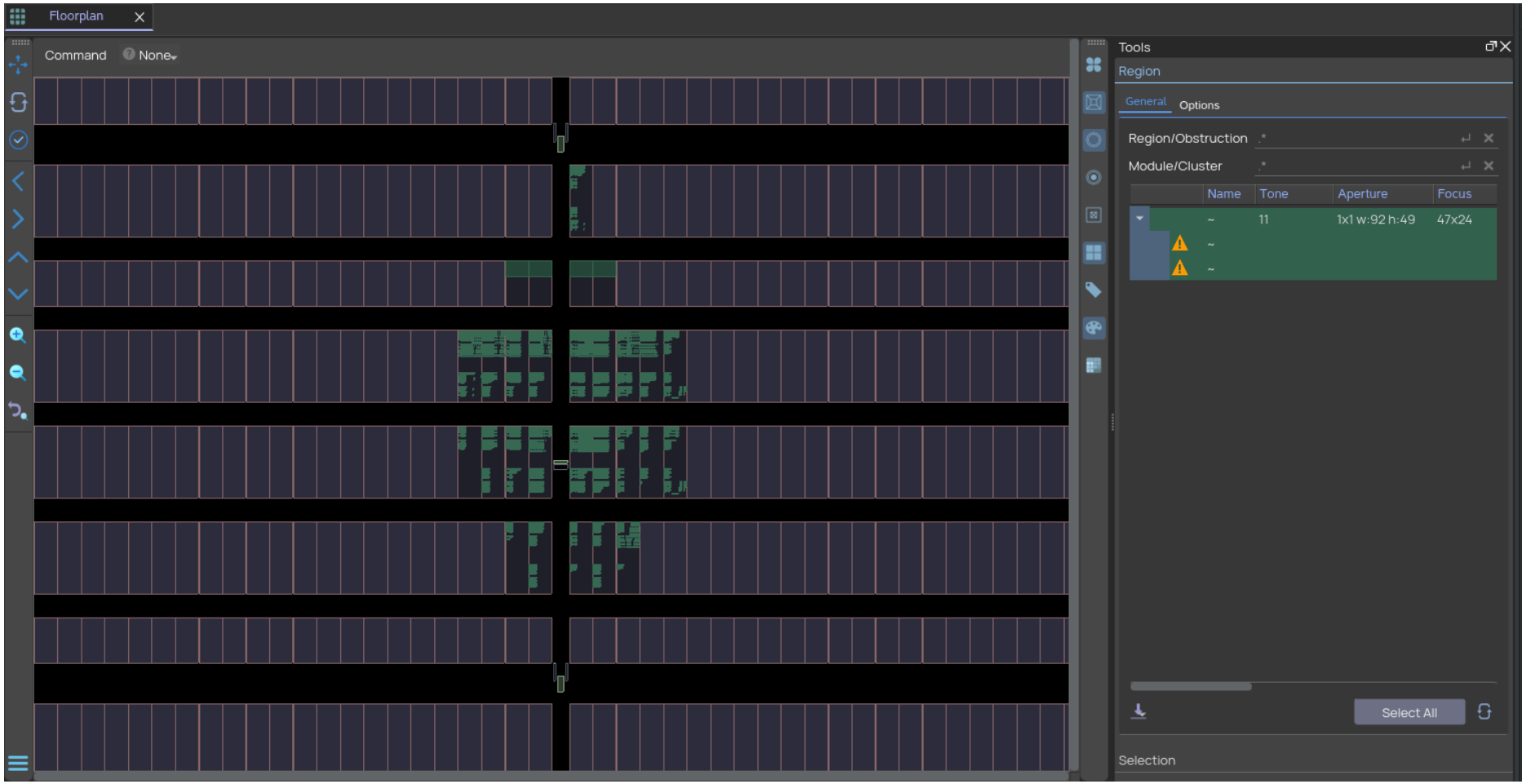
- Reports
  - `ios.rpt` : Reports the IOs of the design
  - `lowskew.rpt` : Reports all the lowskew network statistics at each step of the flow
  - `regions.rpt` : Reports the regions created of the design

- You can generate at place stage:
  - Netlist in VHDL or Verilog

All place steps are saved in different .nym files which could be loaded by the GUI and NXpython command.

Place 1/5

Automatic placement without floorplanning

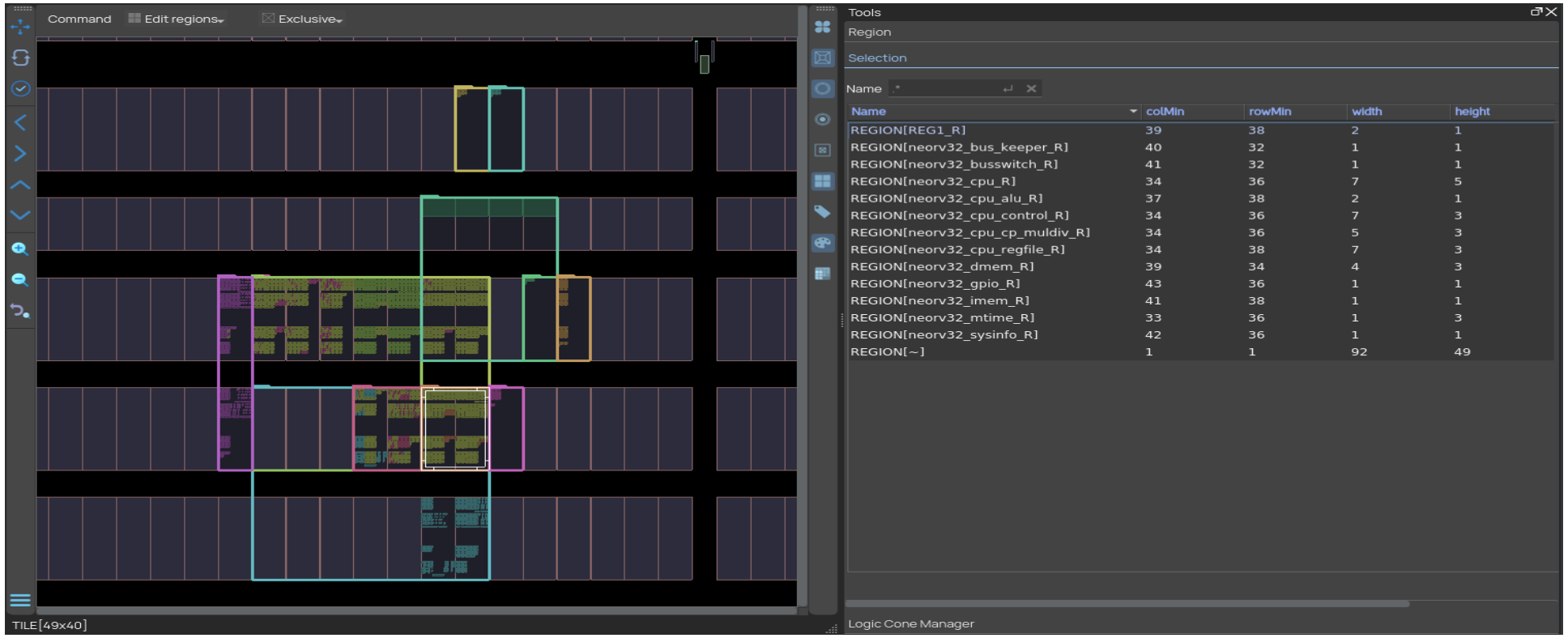


Floorplanning

To optimize performances, the design can be splitted into several regions.

Regions can be easily edited through the GUI or by NXpython method.

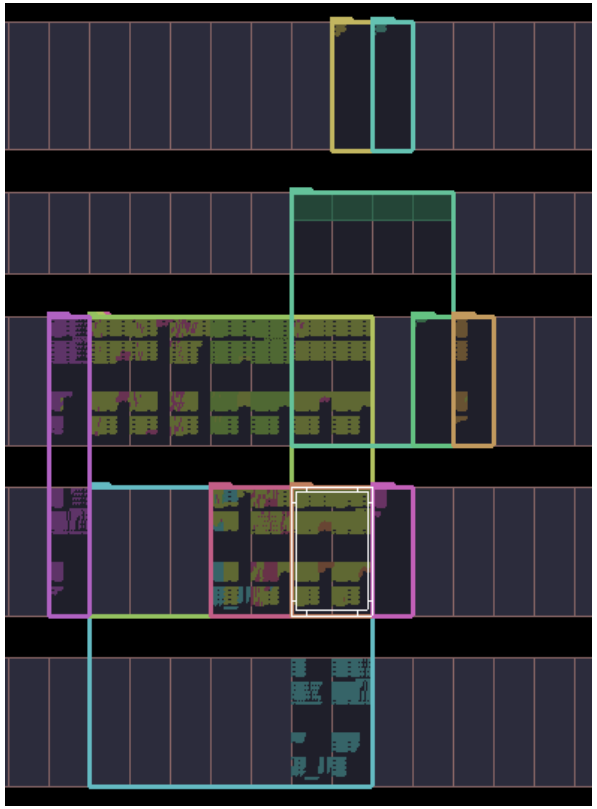
Regions are based on `hierarchy.rpt` file.

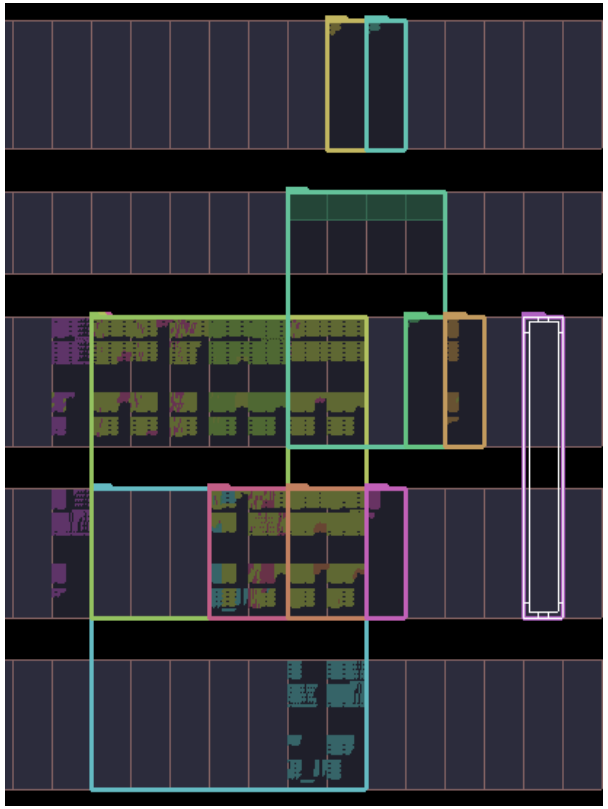


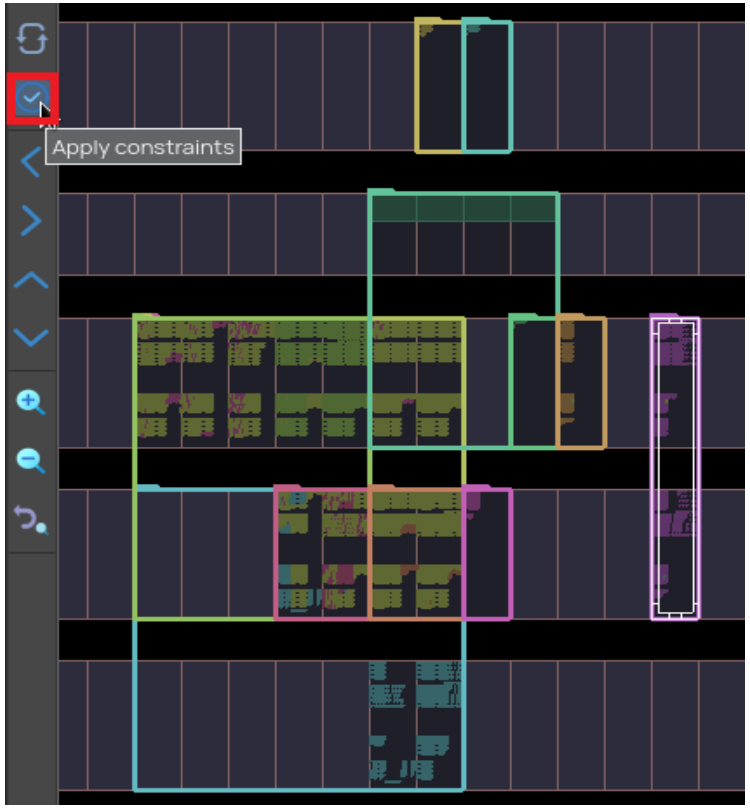
Name	colMin	rowMin	width	height
REGION[REG1_R]	39	38	2	1
REGION[neorv32_bus_keeper_R]	40	32	1	1
REGION[neorv32_busswitch_R]	41	32	1	1
REGION[neorv32_cpu_R]	34	36	7	5
REGION[neorv32_cpu_alu_R]	37	38	2	1
REGION[neorv32_cpu_control_R]	34	36	7	3
REGION[neorv32_cpu_cp_muldiv_R]	34	36	5	3
REGION[neorv32_cpu_regfile_R]	34	38	7	3
REGION[neorv32_dmem_R]	39	34	4	3
REGION[neorv32_gpio_R]	43	36	1	1
REGION[neorv32_imem_R]	41	38	1	1
REGION[neorv32_mtime_R]	33	36	1	3
REGION[neorv32_sysinfo_R]	42	36	1	1
REGION[~]	1	1	92	49

Refresh regionning

To update the region modification, you must click on “Apply constraints” button before continuing the flow.



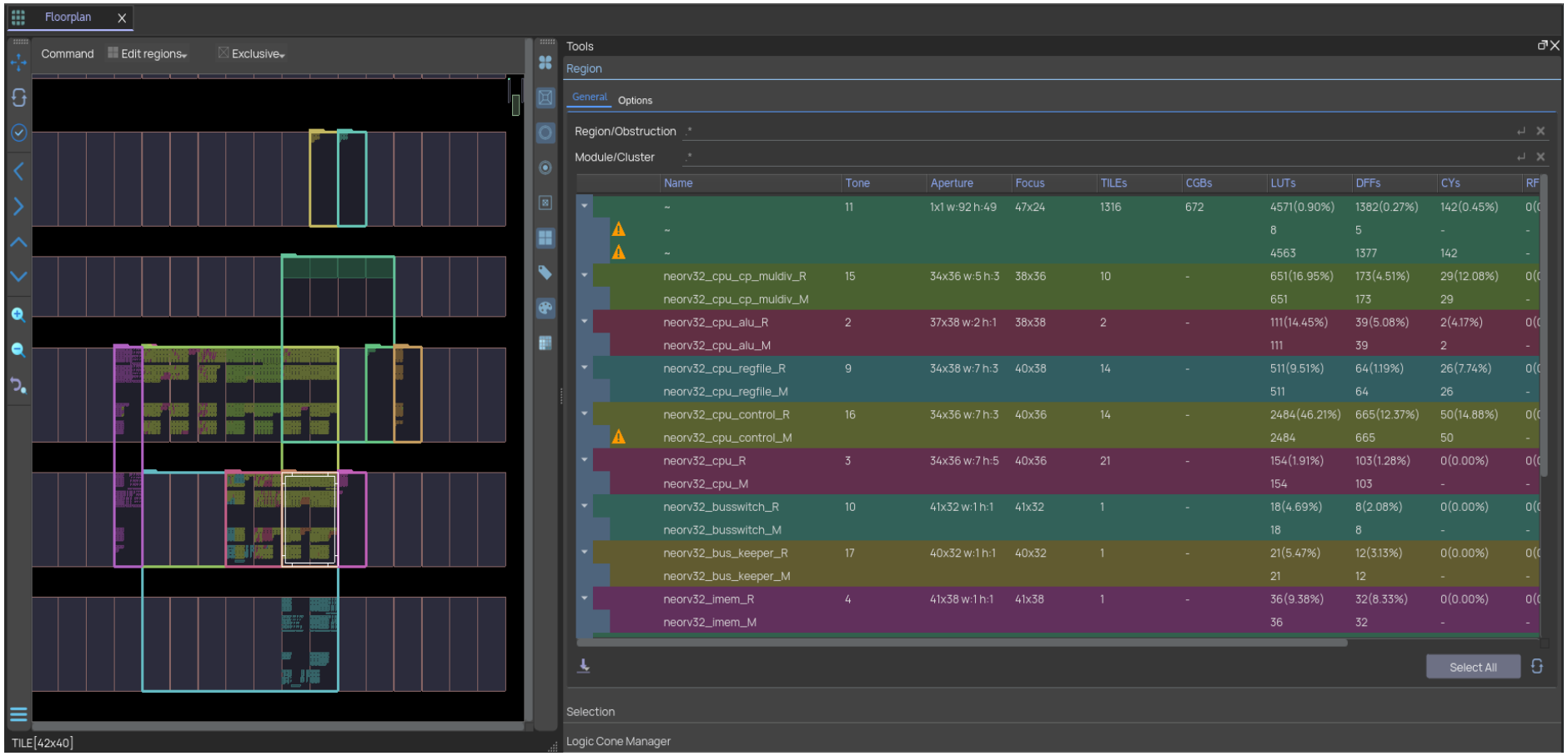




Regions

Once regions are created, you can see:

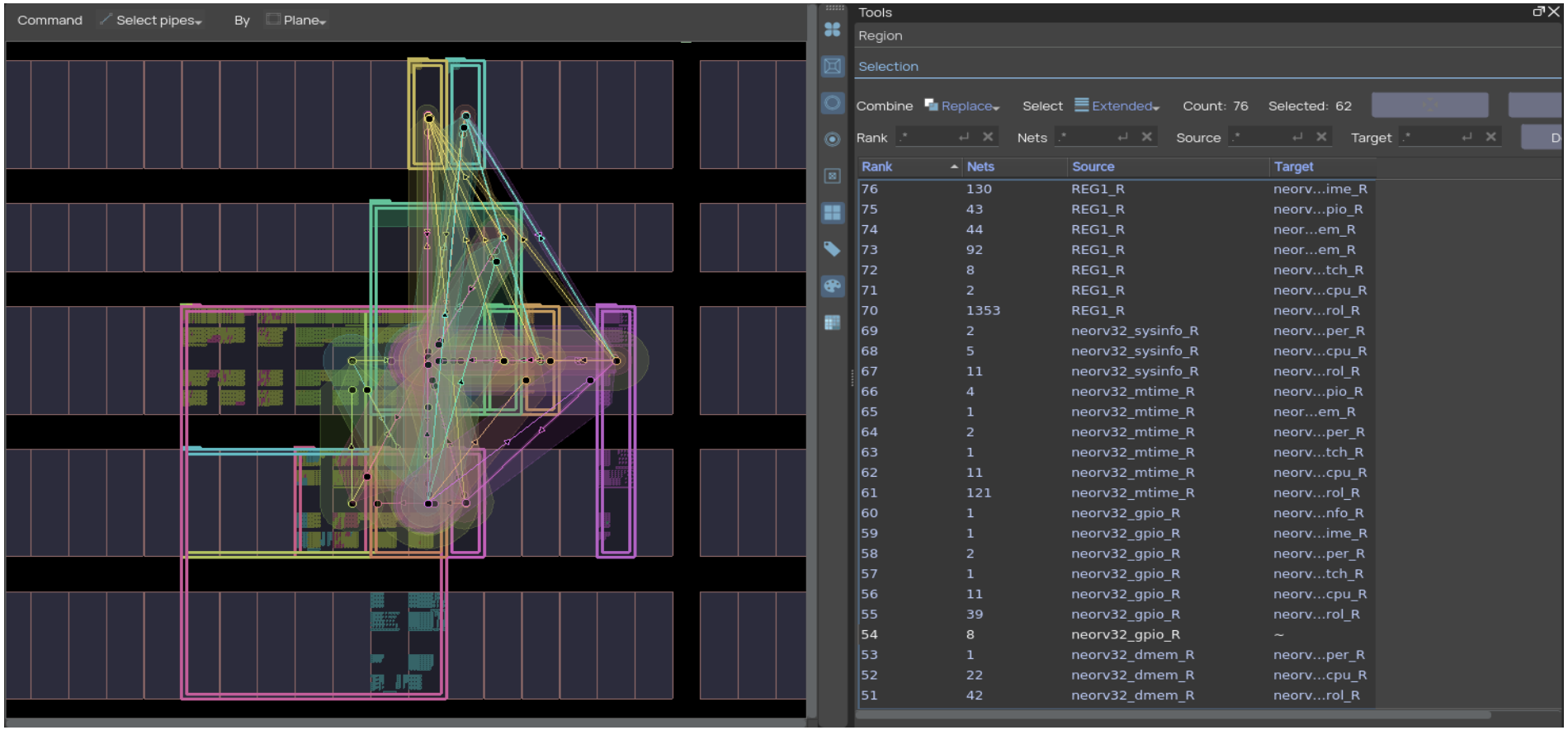
- The regions in floorplan.
- The resources in each region
- If there is any overflow



Name	Tone	Aperture	Focus	TILEs	CGBs	LUTs	DFFs	CYs	RF
~	11	1x1 w:92 h:49	47x24	1316	672	4571(0.90%)	1382(0.27%)	142(0.45%)	0(
~						8	5	-	-
~						4563	1377	142	-
neorv32_cpu_cp_muldiv_R	15	34x36 w:5 h:3	38x36	10	-	651(16.95%)	173(4.51%)	29(12.08%)	0(
neorv32_cpu_cp_muldiv_M						651	173	29	-
neorv32_cpu_alu_R	2	37x38 w:2 h:1	38x38	2	-	111(14.45%)	39(5.08%)	2(4.17%)	0(
neorv32_cpu_alu_M						111	39	2	-
neorv32_cpu_regfile_R	9	34x38 w:7 h:3	40x38	14	-	511(9.51%)	64(1.19%)	26(7.74%)	0(
neorv32_cpu_regfile_M						511	64	26	-
neorv32_cpu_control_R	16	34x36 w:7 h:3	40x36	14	-	2484(46.21%)	665(12.37%)	50(14.88%)	0(
neorv32_cpu_control_M						2484	665	50	-
neorv32_cpu_R	3	34x36 w:7 h:5	40x36	21	-	154(1.91%)	103(1.28%)	0(0.00%)	0(
neorv32_cpu_M						154	103	-	-
neorv32_busswitch_R	10	41x32 w:1 h:1	41x32	1	-	18(4.69%)	8(2.08%)	0(0.00%)	0(
neorv32_busswitch_M						18	8	-	-
neorv32_bus_keeper_R	17	40x32 w:1 h:1	40x32	1	-	21(5.47%)	12(3.13%)	0(0.00%)	0(
neorv32_bus_keeper_M						21	12	-	-
neorv32_imem_R	4	41x38 w:1 h:1	41x38	1	-	36(9.38%)	32(8.33%)	0(0.00%)	0(
neorv32_imem_M						36	32	-	-

Select pipes

How many connections there are between two regions?

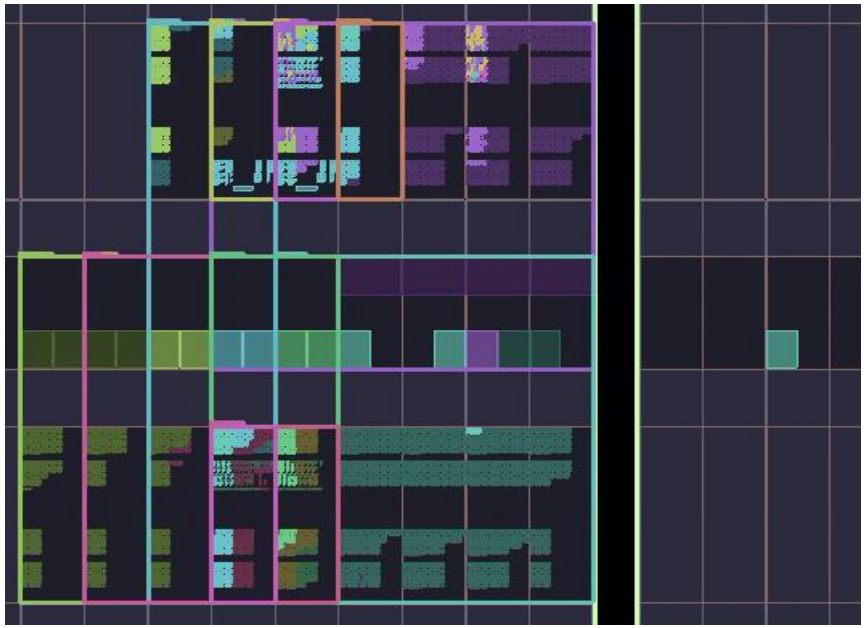


SetSite

This feature allows you to:

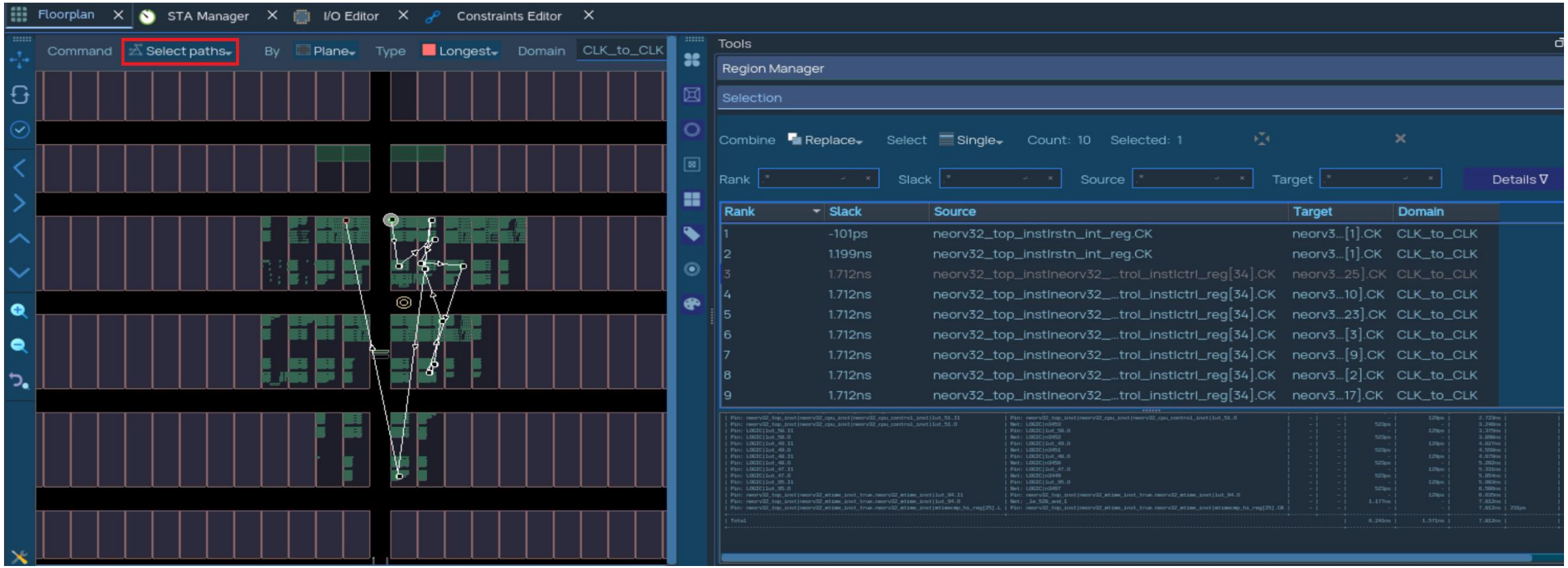
- Select one or several elements to spot in a specific location.
- Spot one or several elements override the region constraints

SetSite can be used through NXpython method and GUI.



STA and Floorplanning

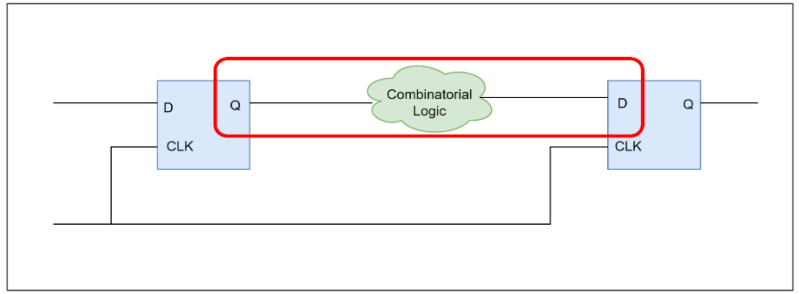
Select Path to check critical paths in the GUI
The floorplan shows the critical paths which can be selected by domain and by type (setup and hold).

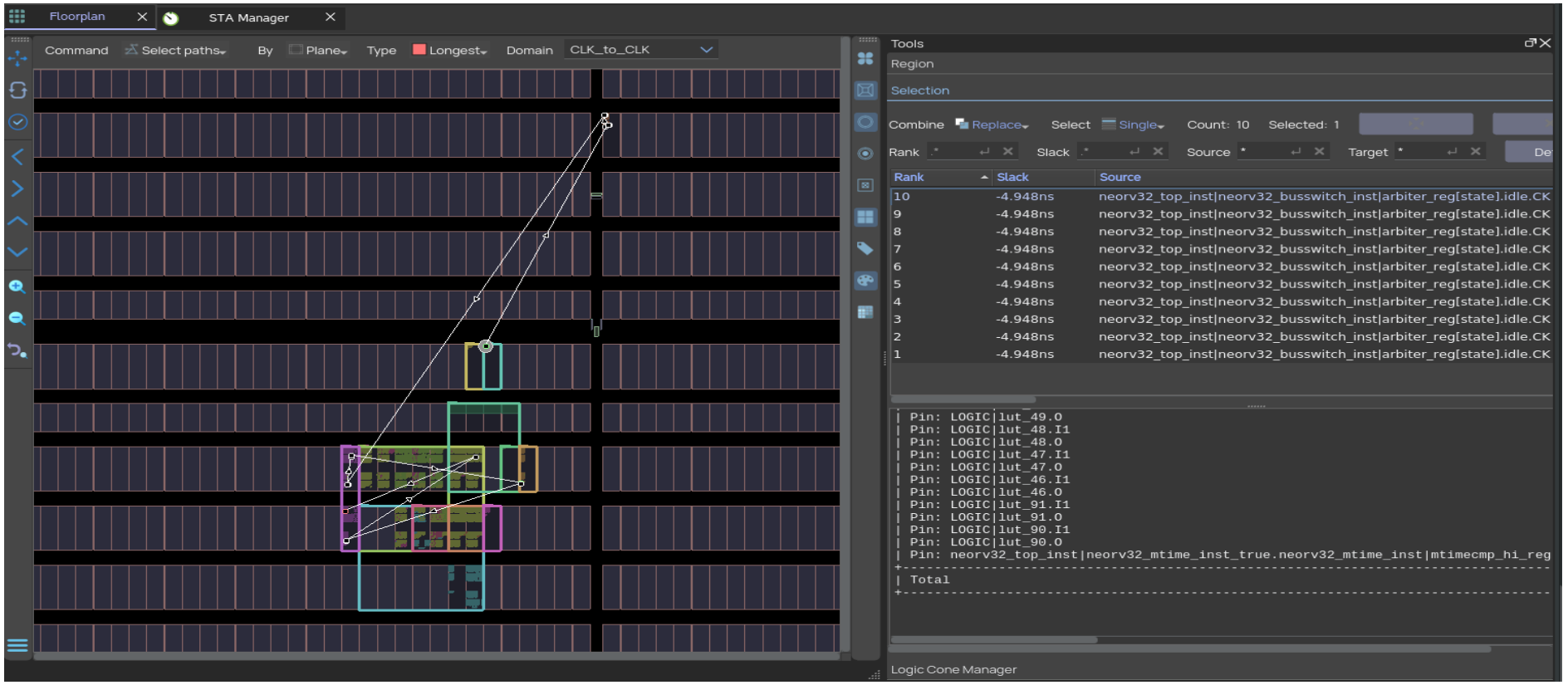


Manage Critical Path

There is a feature to manage specific critical path between two registers.

To reduce the propagation timing, you can constrain the path in a region.

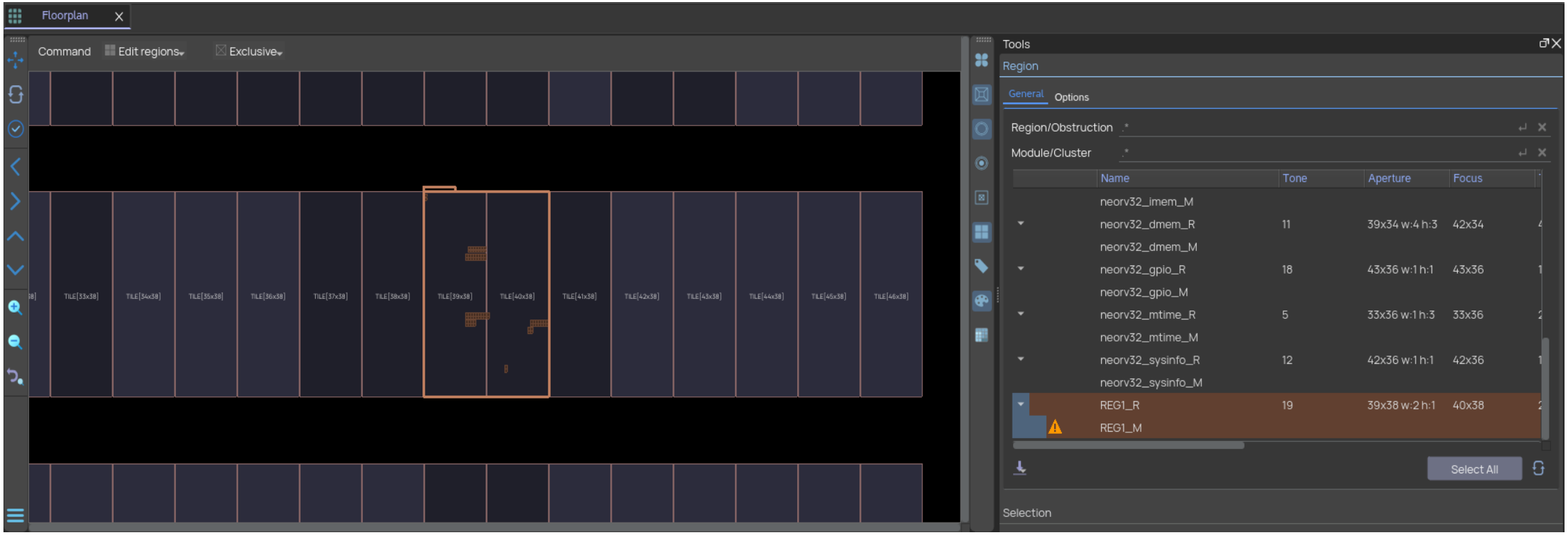




ConstrainPath

ConstrainPath method allows you to catch all resources between 2 registers and create a specific region to constrain these resources.

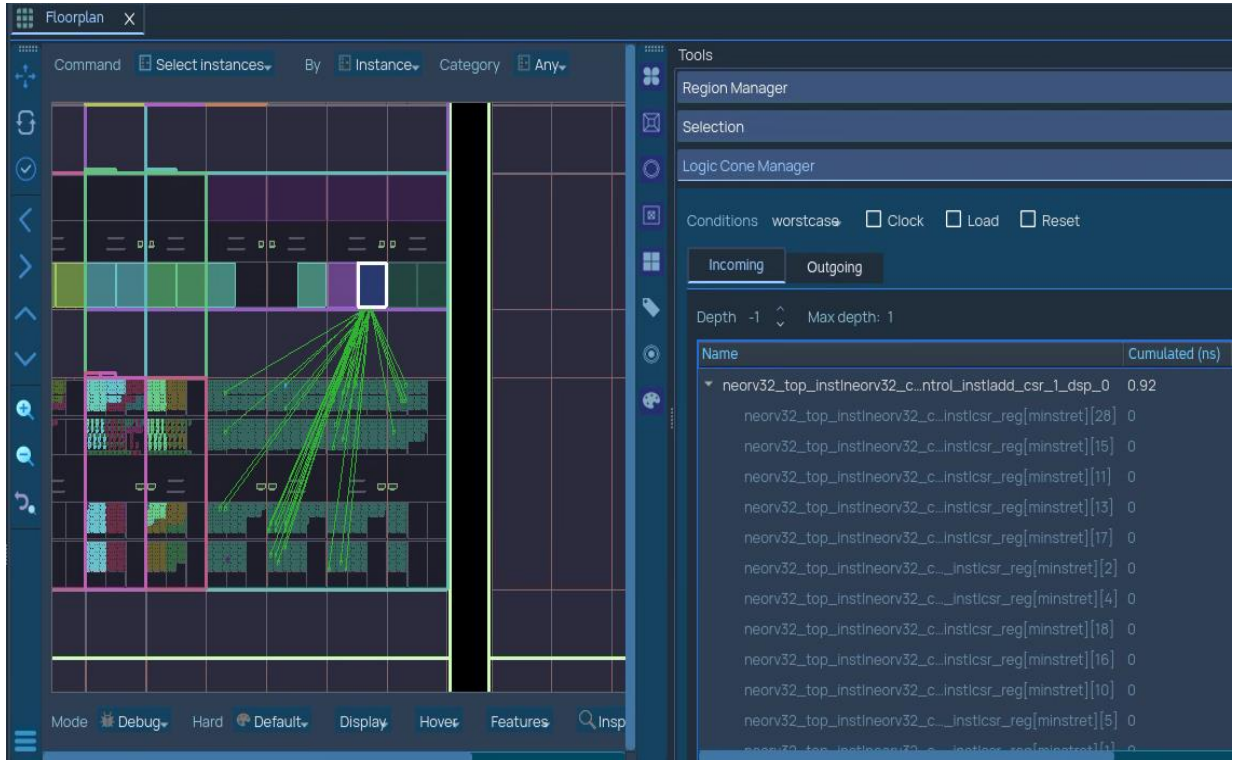
Both registers could be in different regions.

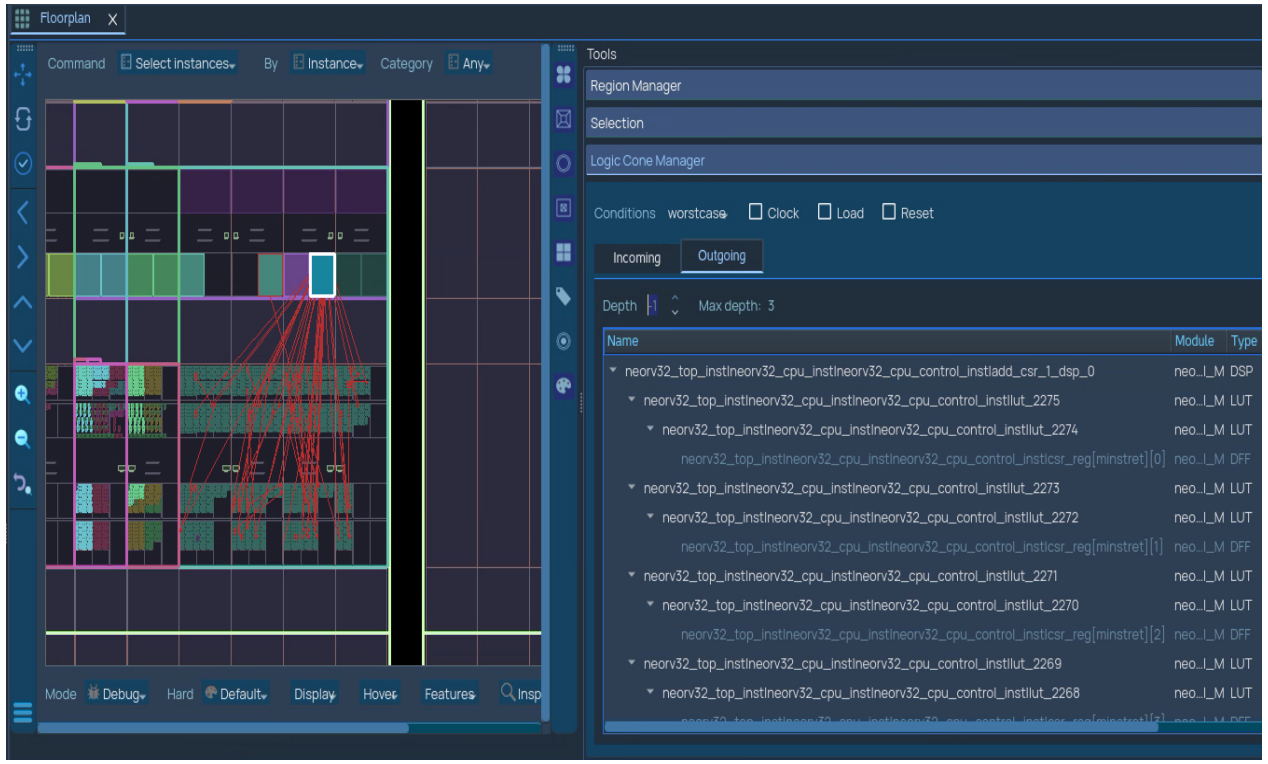


Logic Cone

The Logic Cone Interface is only available in GUI.
It must be only used for **test** or when the **code is fixed**.
The Logic Cone Interface applies SetSite constraint on each element.

What is Logic Cone Interface?





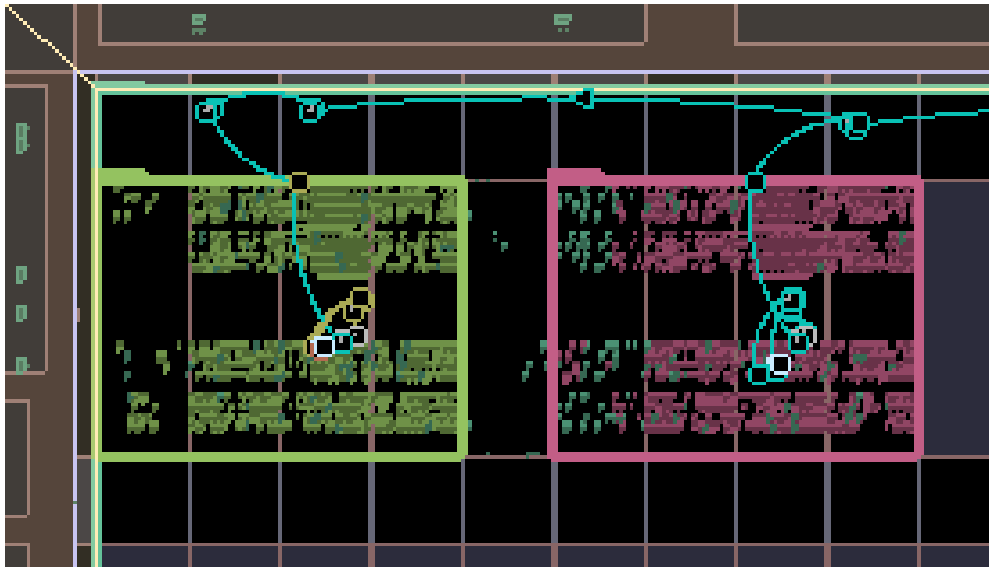
PreplaceIP

PreplaceIP gives the possibility to save a design as an IP and reuse it.

Only instances position are saved.

The spot (fix position) will be different.

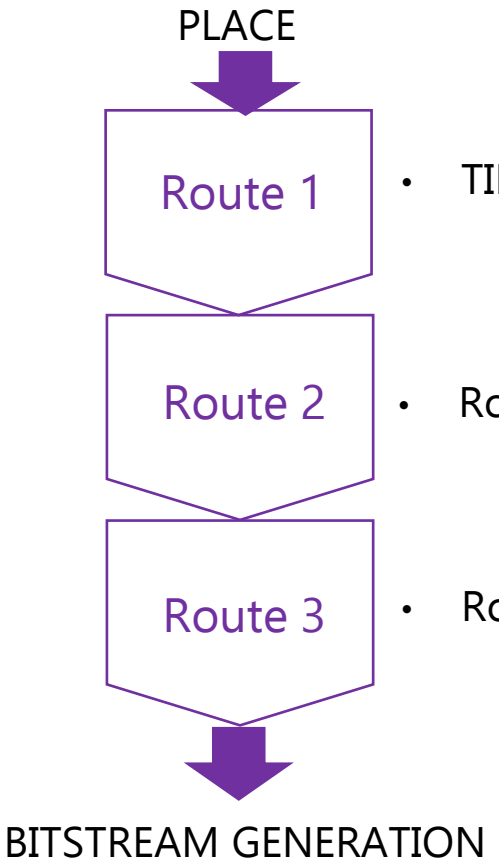
The PreplaceIP regions are not exclusive



Route

NanoXplore

Route



- TILE Routing (large grain)
- Routing inside the TILEs (fine grain)
- Routing connections between TILEs and CGB

Route – Files Generation

- Logs
  - general.log : contains all messages printed during the Route step.

- You can generate at last route stage:
  - Netlist in VHDL or Verilog
  - SDF (Backannoted)

All route steps are saved in different .nym files which could be loaded by the GUI and NXpython command.

You can change several IO parameters as driver value.

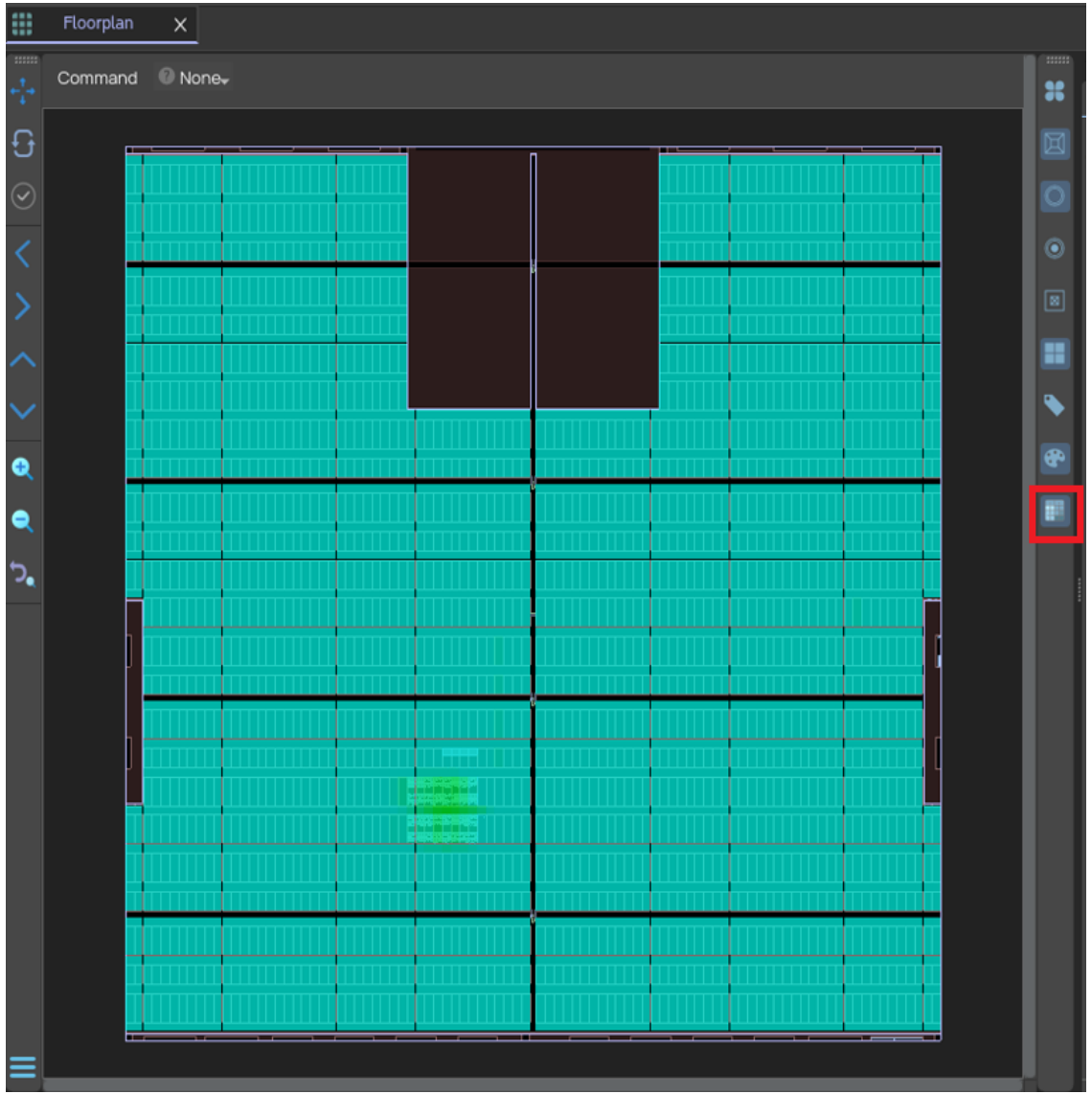
- Generate Bitstream can be done by GUI or NXpython method.

- Bitstream size depends on your design.

Congestion map

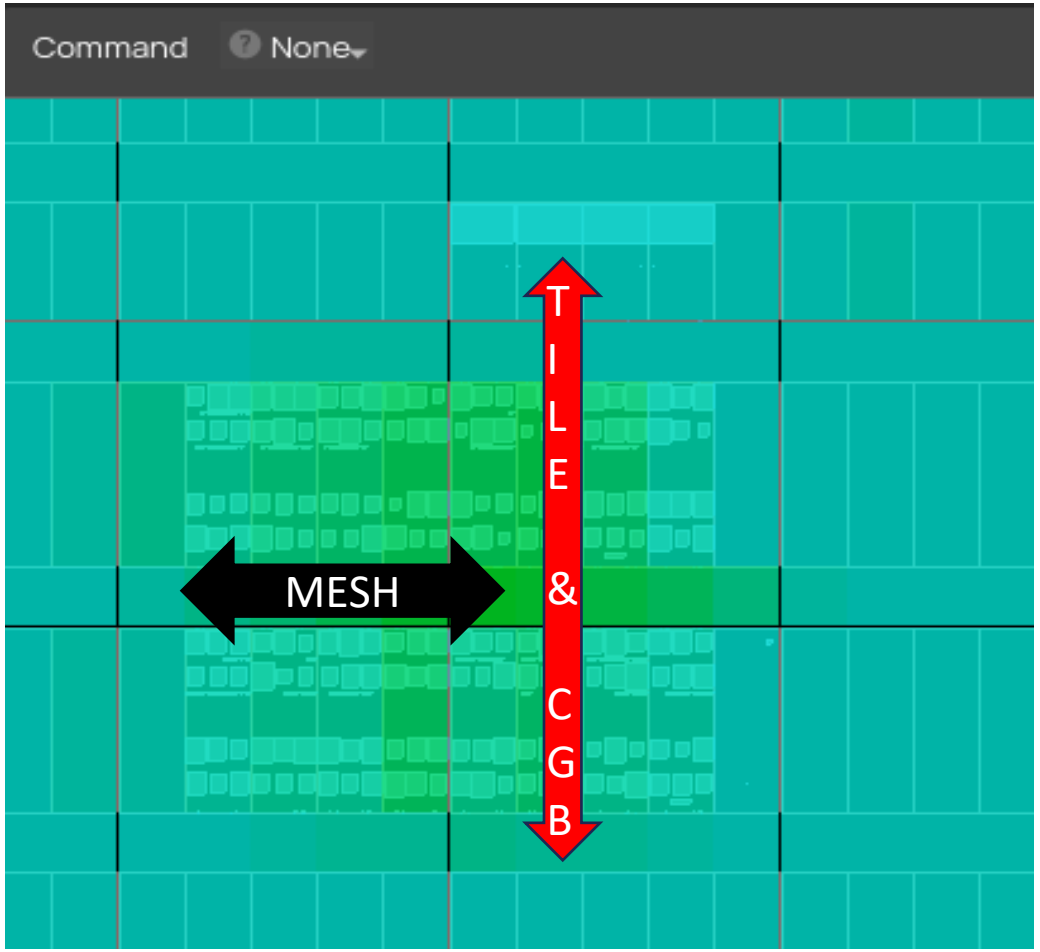
A congestion map is available with the button “Show Congestion Map”

- Vertical and horizontal congestion
- Available only when design is routed



Congestion map

- MESH represents horizontal congestion.
- TILE and CGB represent vertical congestion.



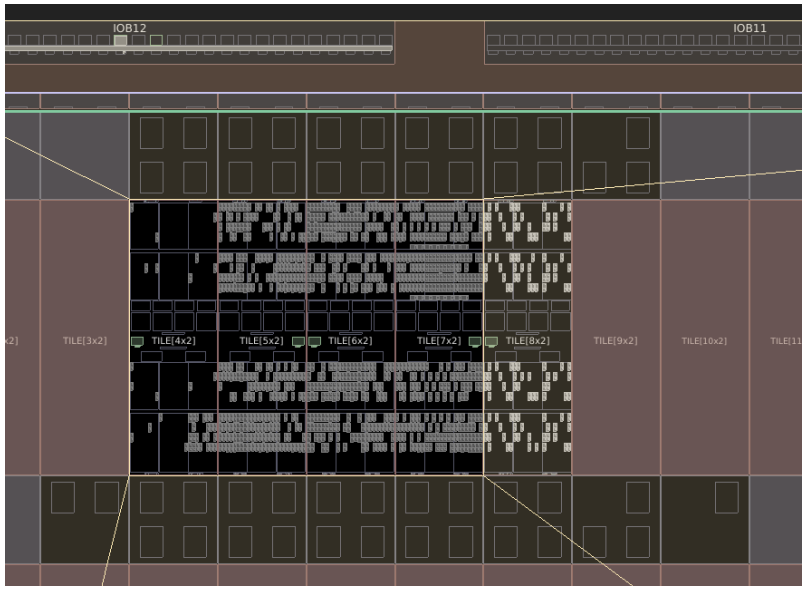
PreRoutedIP

BETA

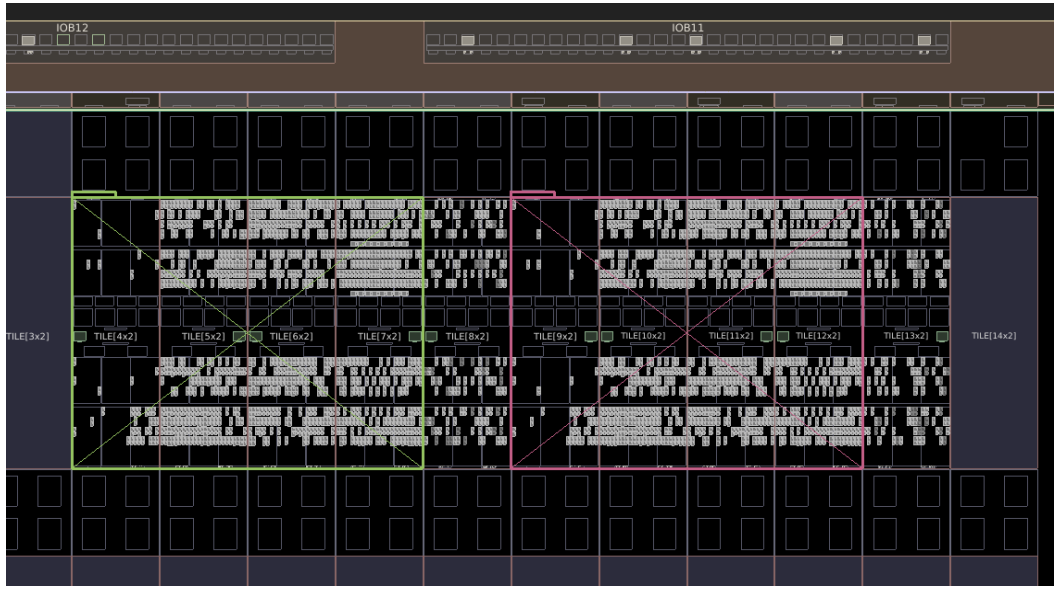
Integrates some optimized PreRouted IPs into a top design with the same performances.

For PreRoutedIP:

- The spot and route are saved
- The regions are exclusive



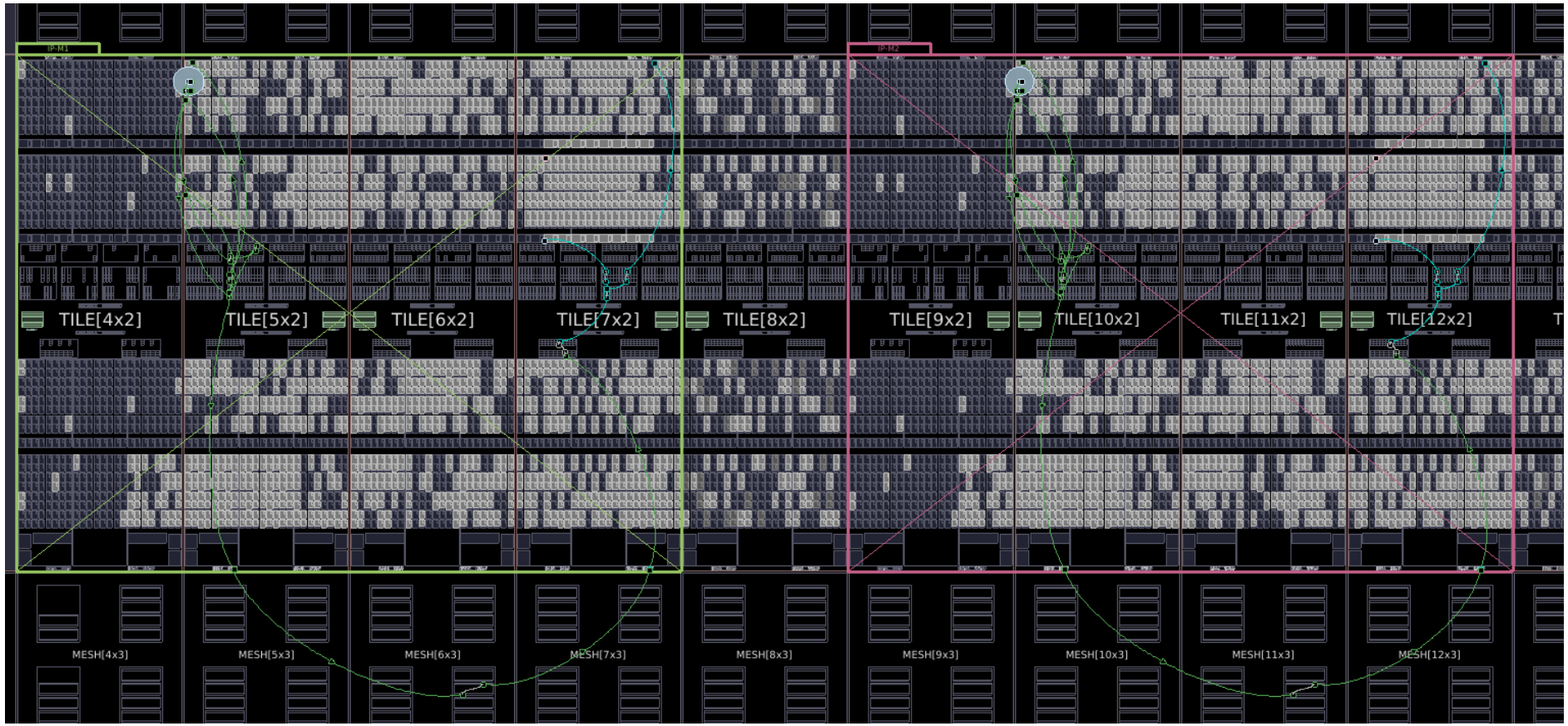
IP



TOP of another project

Routing is preserved in all instantiated IPs

BETA



IP & Debug Tools

NanoXplore

NxCore Catalog – 23.5.1

Clock Configuration

PLL	NG medium NG ultra **Ultra 300**

NX Scope

Scope	NG medium NG ultra **Ultra 300**

Memory

DFI	NG medium
Asynchronous FIFO	NG medium

Interface

APB Slave Controller	NG medium
HSSL Wrapper for Spacefiber Interface 64 bits	NG ultra
QSPI Controller (Simulation only)	**Ultra 300**
I2C	NG medium NG ultra **Ultra 300**

NxCore Catalog - Roadmap

Interface

SpaceWireRMAP	NG medium NG ultra Ultra 300
SpaceWire Bank	NG ultra Ultra 300
DDR3/DFI	NG ultra
JESD204B Prototype	NG ultra Ultra 300
SpaceWire IOM	NG ultra Ultra 300
ESIstream Interface DAC and ADC	NG ultra Ultra 300
Aurora	NG ultra Ultra 300

Bridge

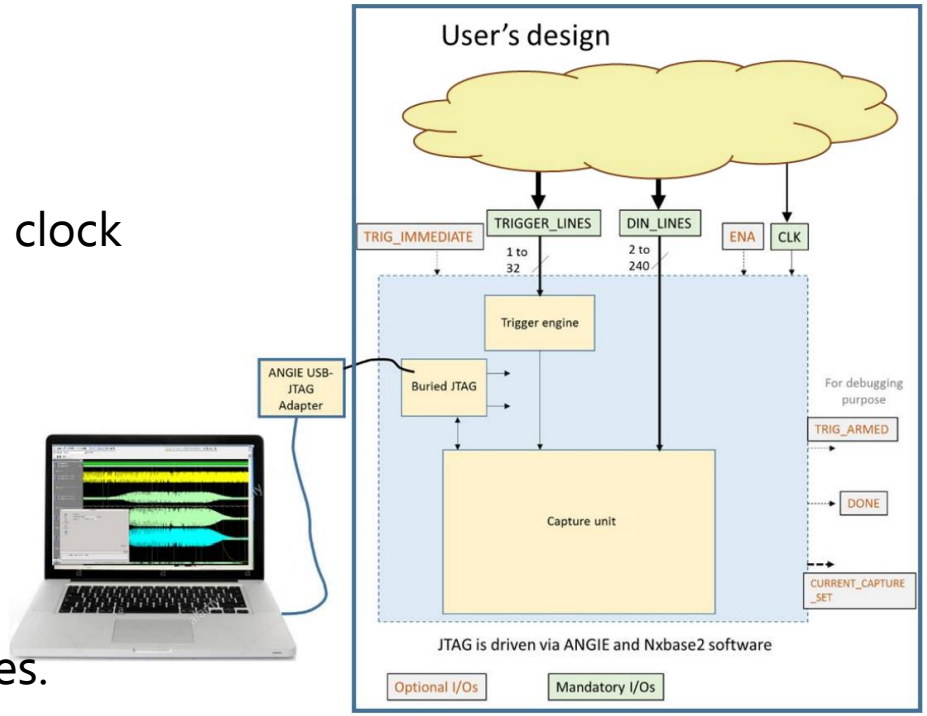
AXI4 to AHB	NG medium NG ultra Ultra 300
AHB to AXI4	NG medium NG ultra Ultra 300
AHB to IP1RW	NG medium NG ultra Ultra 300
AXI4 Full Interconnect	NG medium NG ultra Ultra 300

Debug Tool - NxScope

- NxScope is a NanoXplore IP Core.
  It is an embedded logic analyzer which allows you to :
  - Sample a collection of internal data synchronously with a user's clock
  - Analyze the sampled results in a waveform viewer

- The results can be displayed and analyzed either with
  - ModelSim waveform viewer
  - GTKWave (free waveform display tool)

NxScope logic analyzer is built with the FPGA available logic resources.



Link => https://nanoxplore-wiki.atlassian.net/wiki/spaces/NAN/pages/357467425/NxDesignSuite+23.5+NxCore

Documentation

NanoXplore

NanoXplore Wiki

Welcome to the NanoXplore Wiki

NanoXplore is a privately owned fabless company based in France. The company offers a comprehensive portfolio of SoCs and FPGA devices for aerospace, defense and industrial markets. Products include a leading radiation hardened FPGA portfolio.

Devices

NanoXplore develop devices focusing on three axes:

- A complete **Radiation Hardened FPGA** portfolio qualified for space applications and technologies including STM 65nm and 28nm FD-SOI
- Hard block embedded FPGA core IPs labelled as eFPGA products
- The development of ASIC devices

Our Radiation Hard FPGA range each include specific technical documentation, evaluation boards and DevKits to help discover and use all the available features.

NG-MEDIUM	NG-LARGE	NG-ULTRA	NG-ULTRA300
NG medium	NG large	NG ultra	NG ultra 300
Documentation	Documentation	Documentation	Documentation
Evaluation boards & DevKits	Evaluation Boards & DevKits	Evaluation Boards & DevKits	Evaluation Boards & DevKits

Development Tools

To accompany the hardware portfolio, NanoXplore has conceived a complete dedicated software chain to map a hierarchical design logic application into the programmable core matrix of any NX FPGA device:

- NXmap was designed by NanoXplore as the main software tool to perform a full design flow compilation to programme FPGAs, including synthesis, place, route, static timing analysis and bitstream generation
- NXbase2 is a command line tool that can interact with evaluation kit boards for NanoXplore's chips. It provides a way to upload bitstream files into the chip and is able to control some of the hardware features of the related evaluation kits. The communication between computer and evaluation kit board is done by JTAG via the ANGIE USB-JTAG adapter provided with the evaluation kit board. NXboard is a graphical user interface, using NXbase2, that allows easy interaction with the evaluation kit boards
- SDK gathers embedded software tools and features for Development Kits to use efficiently complex parts of dedicated FPGAs like the R5 processor for NG-Large and the NG-Ultra System On Chip (SOC) functionalities
- NanoXplore provides its own License Manager Daemon (NXLMD) which must be installed to use NanoXplore tools

NXmap	NXbase2	SDK	Server license
Documentation & Download versions	Documentation & Download versions	Documentation & Download	Documentation & Download versions

IPs

NanoXplore develops its own portfolio of soft IPs designed to configure FPGA dedicated functionalities:

- DDR2 DFI IP
- SerDes IP
- SpaceWire bank IP

NanoXplore also provides IPs designed by partners.

Information on NanoXplore Wiki

Current list of devices are RH FPGA:

- NG-MEDIUM
- NG-LARGE
- NG-ULTRA
- ULTRA300

For each device, there is a main page with overview and links to download:

- Datasheet
- Pinout Allocation
- Ball Out
- IBIS models
- Power Estimator

Other documents are available:

- Configuration Guide
- Application Notes
- Cookbook
- Radiative Test Report

Link: https://nanoxplore-wiki.atlassian.net/

No need to subscribe !

Conclusion

NanoXplore

Training Package

Shows several design examples.
Helps to learn how to use our software.

NxPython | STA | NxCore | NxBase2 | NxScope

Link => https://nanoxplore-wiki.atlassian.net/wiki/spaces/NAN/pages/357468388/NxDesignSuite+23.5+Training+Package

NanoXplore Support

Any questions about our products or licenses?

- Contact customer support by mail at support@nanoxplore.com
- Create an account on support platform at https://support.nanoxplore.com

NanoXplore

Thank you