

Lattice Diamond Software Simulation and Reveal Tools

Topic

- Use Simulation Wizard tool to simulate the design
- Use Reveal tool to debug the design

Simulation

Simulation Wizard

The diagram illustrates two methods to launch the Simulation Wizard:

- Method 1:** Clicking the Simulation Wizard icon (a lightning bolt) on the **Start Page** toolbar.
- Method 2:** Navigating to **Tools** > **Simulation Wizard** in the main application menu.

Both methods lead to the **Simulation Wizard** dialog box, which is shown on the right. The dialog contains the following fields and options:

- Simulator Project Name:** Enter a name for your simulator project and specify a directory where the project file will be stored.
- Project name:**
- Project location:** ...
- Simulator:**
 - ☒ Active-HDL
 - ☐ ModelSim/QuestaSim
- Navigation:** < Back, Next >, Cancel

Either way

Simulation Wizard

Variables
in the
testbench

Active-HDL 10.3 (simulation ,simulation) - untitled.awc *

File Edit Search View Workspace Design Simulation Waveform Tools Window Help

Design Browser

testbench

Hierarchy

testbench

Name	Value
leds	FF
clk	0
rst	1

Signal name Value

clk 0

leds FF

rst 1

Waveforms

Cursor 1

untitled.awc *

Console

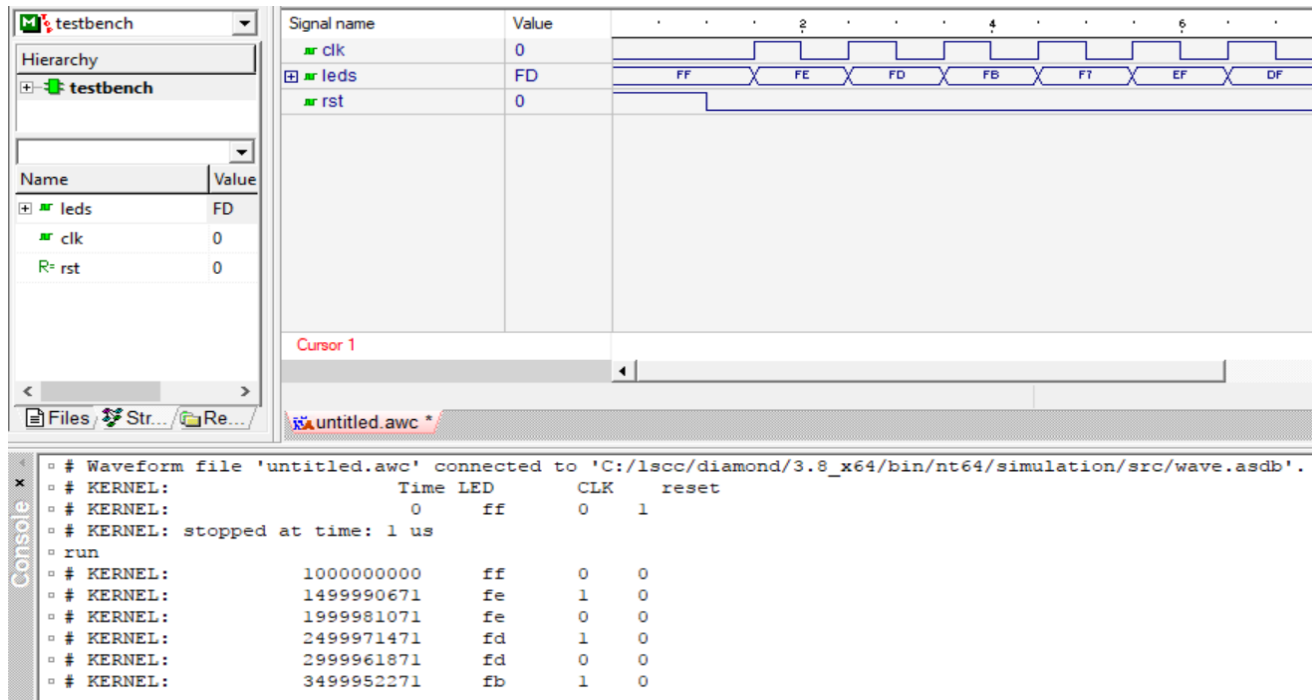
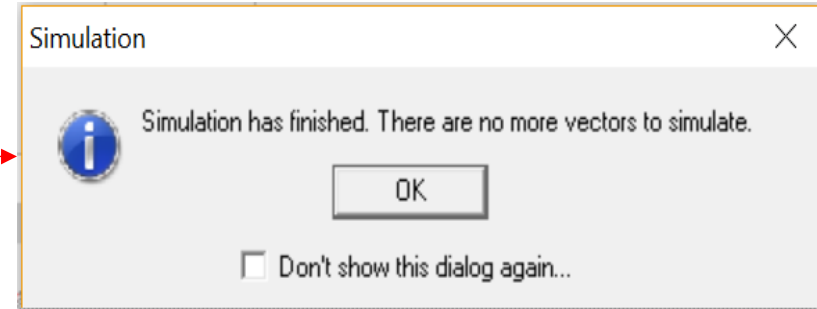
```
# Allocation: Simulator allocated 5553 kB (elbread=1280 elab2=4128 kernel=144 sdf=0)
# KERNEL: ASDB file was created in location C:\lsc\diamond\3.8_x64\bin\nt64\simulation\src\wave.asdb
# 8:18 PM, Thursday, December 14, 2017
# Simulation has been initialized
add wave *
# 3 signal(s) traced.
run 1000ns
# Waveform file 'untitled.awc' connected to 'C:/lsc/diamond/3.8_x64/bin/nt64/simulation/src/wave.asdb'.
# KERNEL: Time LED CLK reset
# KERNEL: 0 ff 0 1
# KERNEL: stopped at time: 1 us
```

Writing
commands or
reading displays

Simulation Wizard

“stop” button

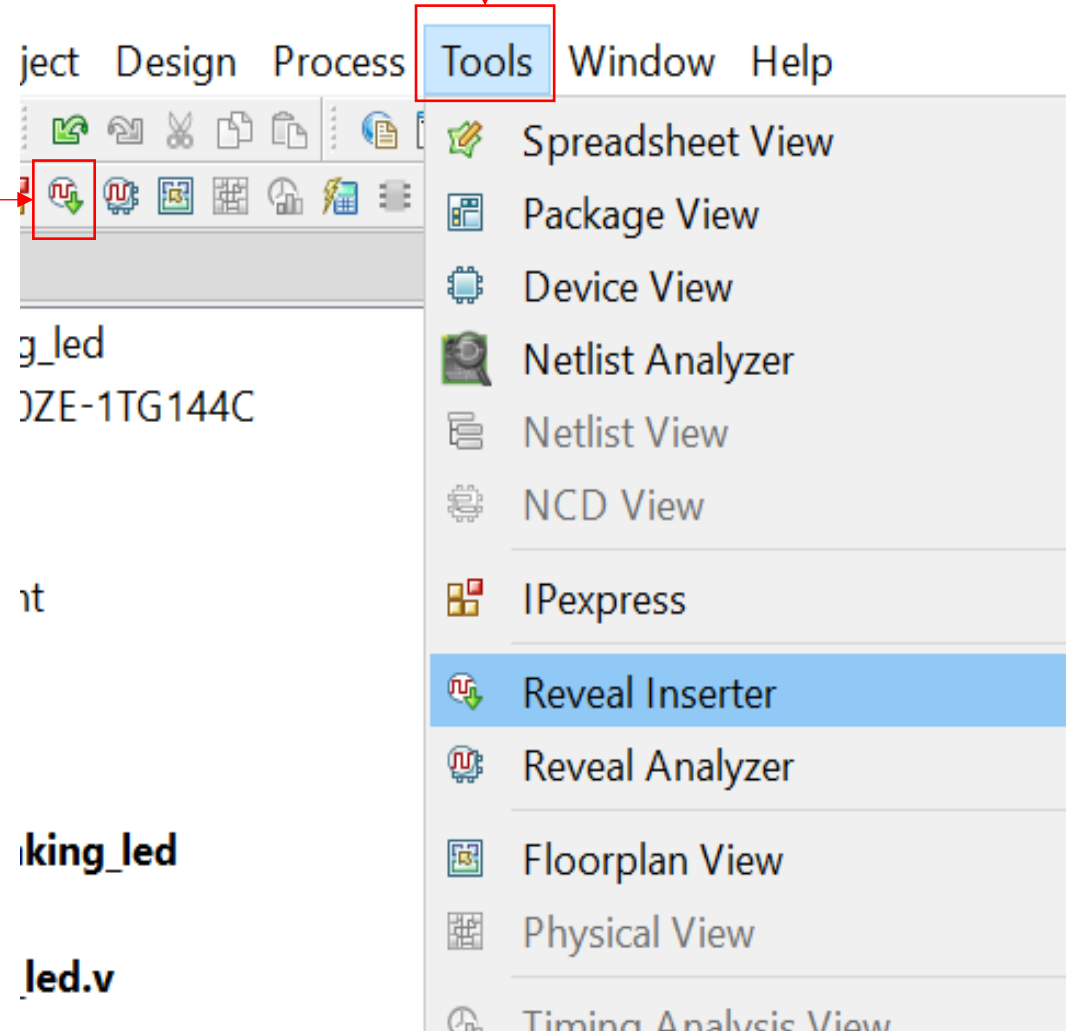
Hit the
“run”
button



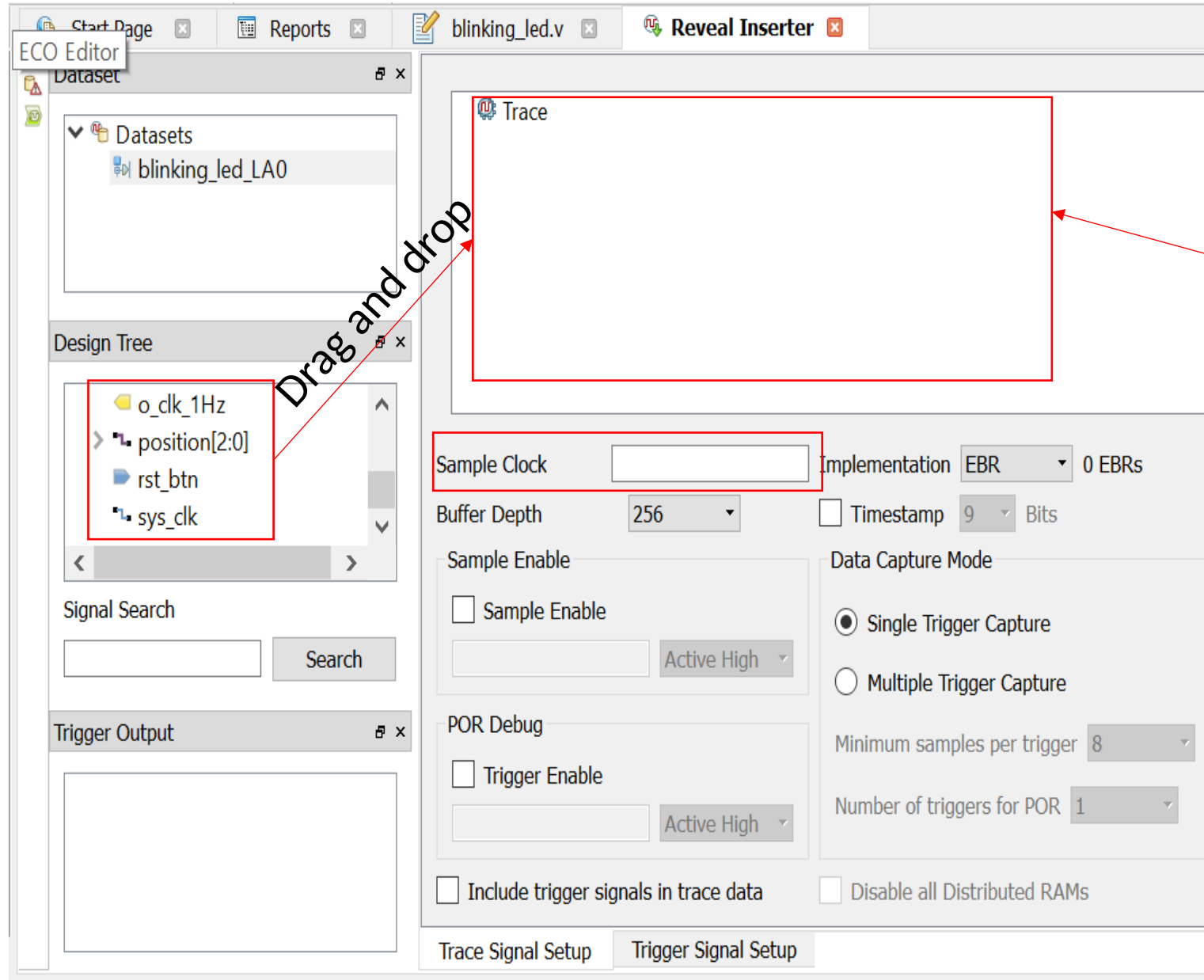
Reveal

Reveal

Either way



Reveal



Signals you want to look at

The "Sample Clock" is the clock for Reveal tool (must run faster or equal JTAG clock)

Reveal

Trigger types

Trace

- rst_btn
- clk_cnt
 - clk_cnt:0
 - clk_cnt:1
 - clk_cnt:2

Sample Clock: sys_clk Implementation: EBR 2 EBRs

Buffer Depth: 256

☐ Timestamp 9 Bits

Sample Enable

☐ Sample Enable Active High

POR Debug

☐ Trigger Enable Active High

☐ Include trigger signals in trace data

☐ Disable all Distributed RAMs

Trace Signal Setup **Trigger Signal Setup**

Data Capture Mode

☒ Single Trigger Capture

☐ Multiple Trigger Capture

Minimum samples per trigger: 8

Number of triggers for POR: 1

blinking_led.v Reveal Inserter *

Trigger Unit

	name	Signals (MSB:LSB)	Operator	Radix	Value
1	TU1		==	Bin	0

Add Remove Default Trigger f Bin

Trigger Expression

	name	Expression	RAM Type	Sequence Depth	Max Sequence Depth	Max Event Counter
1	TE1		0 EBR	0	1	1

Add Remove

Event Counter

☐ Enable final trigger cou Event Counter V 8

Trigger Out

☐ Enable Trigger C Net NET al_debug_core_LA0_net

Trace Signal Setup **Trigger Signal Setup**

Reveal

Double Click

blinking_led.v x Reveal Inserter *

Trigger Unit

	name	Signals (MSB:LSB)	Operator	Radix	Value
1	TU1		==	Bin	0

Add Remove Default Trigger F Bin

Trigger Expression

	name	Expression	RAM Type	Sequence Depth	Max Sequence Depth	Max Event Counter
1	TE1		0 EBR	0	1	1

Add Remove

Event Counter

☐ Enable final trigger cou Event Counter V 8

Trigger Out

☐ Enable Trigger C Net NET al_debug_core_LA0_net

Trace Signal Setup Trigger Signal Setup

TU Signals

Select Signals (3)

(1) (2)

clk_cnt[31:0]@1
int_LED[7:0]
o_clk_1Hz
position[2:0]
rst_btn@Tc,Tg
sys_clk@C
Trigger Output S

>
<

(3)

rst_btn

LSB
↑
↓
MSB

(4) OK Cancel Help

Reveal

Set trigger on rising edge

Trigger Unit

	name	Signals (MSB:LSB)	Operator	Radix	Value
1	TU1	rst_btn	rising edge	Bin	1

Add Remove

Default Trigger F Bin

Trigger Expression

	name	Expression	RAM Type	Sequence Depth	Max Sequence Depth	Max Event Counter
1	TE1	TU1	EBR	1	1	1

Add Remove

Event Counter

☐ Enable final trigger cou Event Counter V 8



Trace Signal Setup Trigger Signal Setup

The “Value” will indicate what bit control the trigger. For example, if the signal is 4-bit type, setting “Value” to “**1100**” will tell the Reveal to trigger on the **rising edge** of the **2 MSB**.

Powerful option, but we don’t use it in this tutorial. Setting **TU1** to be the only trigger unit in Expression

Reveal

[http://
www.latticesemi.com/en/Products
/DesignSoftwareAndIP/FPGAandLDS
/LatticeDiamond.aspx](http://www.latticesemi.com/en/Products/DesignSoftwareAndIP/FPGAandLDS/LatticeDiamond.aspx)

<input type="checkbox"/>	<input type="checkbox"/>	Reveal 3.10 User Guide 
<input type="checkbox"/>	<input type="checkbox"/>	Timing Closure 3.10 
<input type="checkbox"/>	<input type="checkbox"/>	Select All Notify Me of Changes Download Selected as Zip File

Page 31

Operators You can use the following operators to connect trigger units:

- ▶ & (AND) – Combines trigger units using an AND operator.
- ▶ | (OR) – Combines trigger units using an OR operator.
- ▶ ^ (XOR) – Combines trigger units using a XOR operator.
- ▶ ! (NOT) – Combines a trigger unit with a NOT operator.
- ▶ Parentheses – Groups and orders trigger units.
- ▶ THEN – Creates a sequence of wait conditions. For example, the following statement:

```
TU1 THEN TU2
```

means “wait for TU1 to be true, then wait for TU2 to be true.”


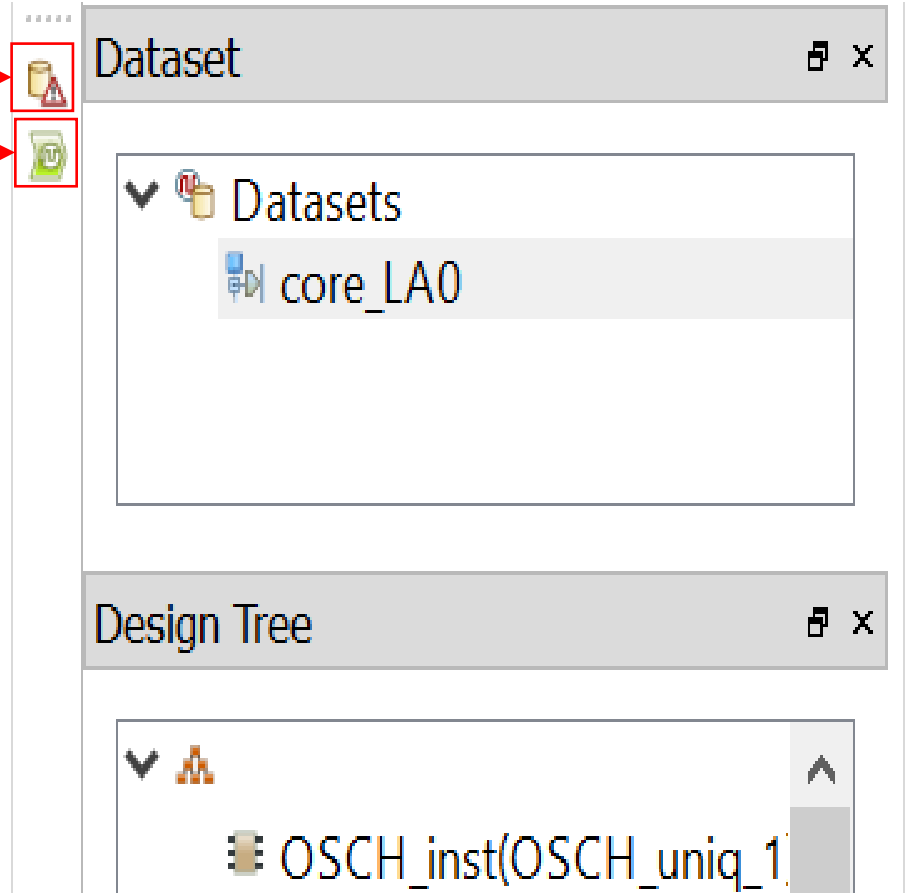
The following expression:

```
(TU1 & TU2) THEN TU3
```

Reveal

Click on “Design Rule Check”

Click on “Insert
Debug” after
passing the
“Design Rule
Check”



```
Output
Starting: "rvl_tu set -name TU1 -val 1"

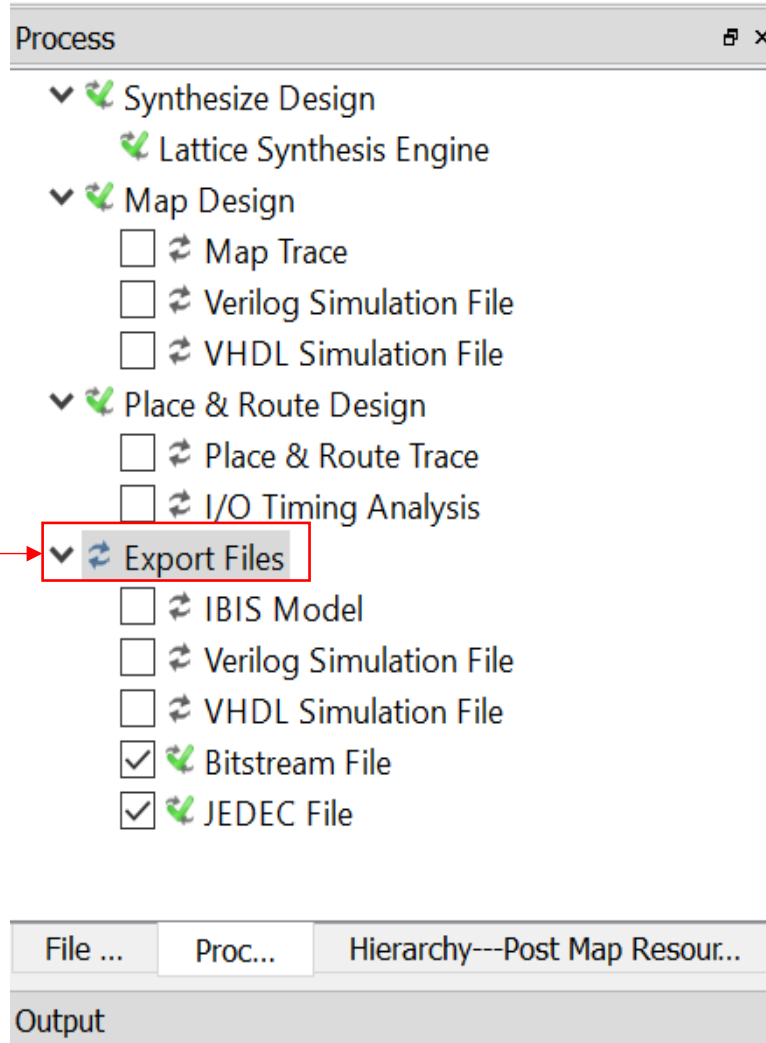
Starting: "rvl_te set -expression TU1 TE1"

Starting: "rvl_project save -overwrite C:/lscd/diamond/3.8_x64/b

Checking design rules ...
INFO - The number of EBRs needed is 3.
INFO - The number of DistrAM (logic/ROM/RAM) slices needed is 0.
Design Rule Check PASSED.
```

Reveal

Double Click to
re-generate
the .bit and .jed
files



The screenshot shows the 'Output' window in the Lattice IDE. The 'UFM Summary' section is displayed, showing the UFM Size and Utilization. A red arrow points from the 'Export Files' step in the 'Process' window to the 'UFM Summary' section. The output text is as follows:

```
UFM Summary.  
=====
```

UFM Size:	511 Pages (128*511 Bits).
UFM Utilization:	General Purpose Flash Memory.

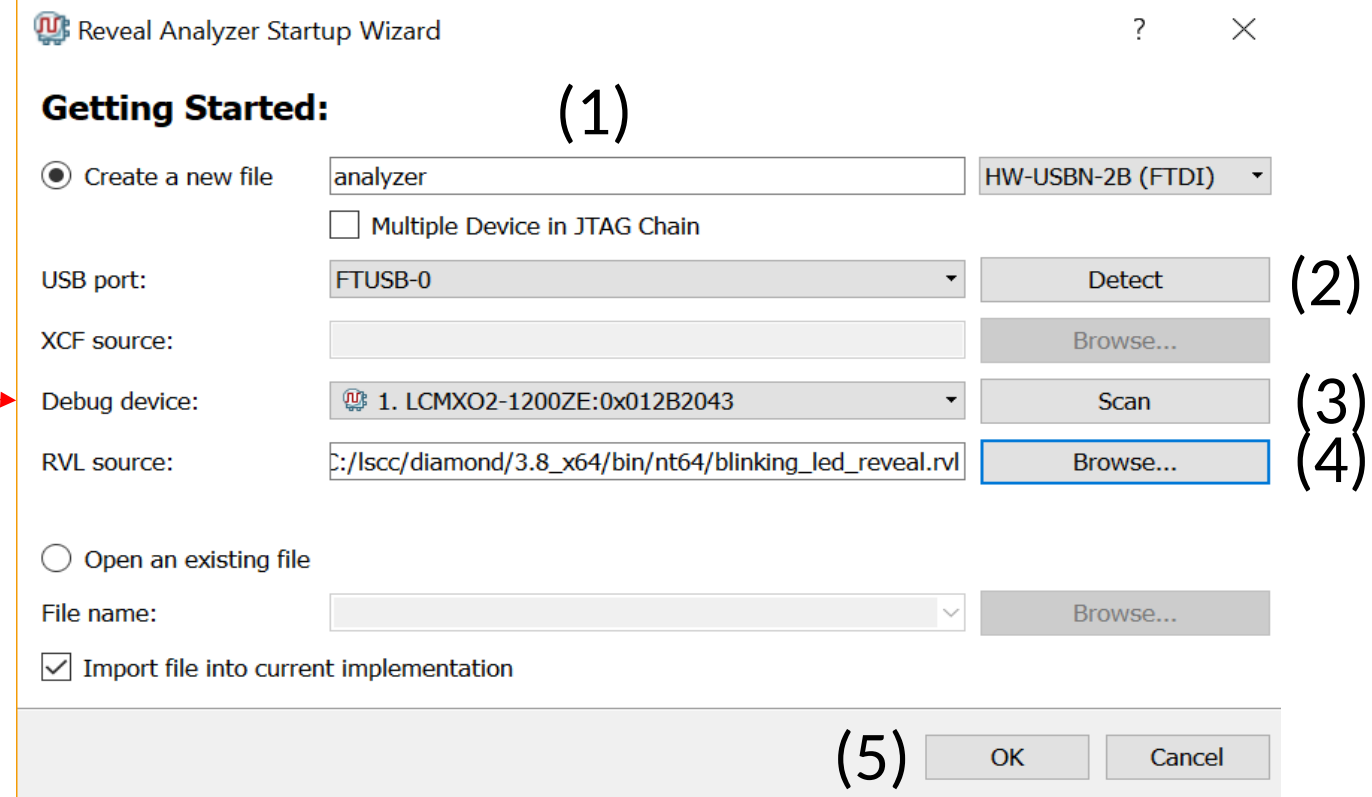
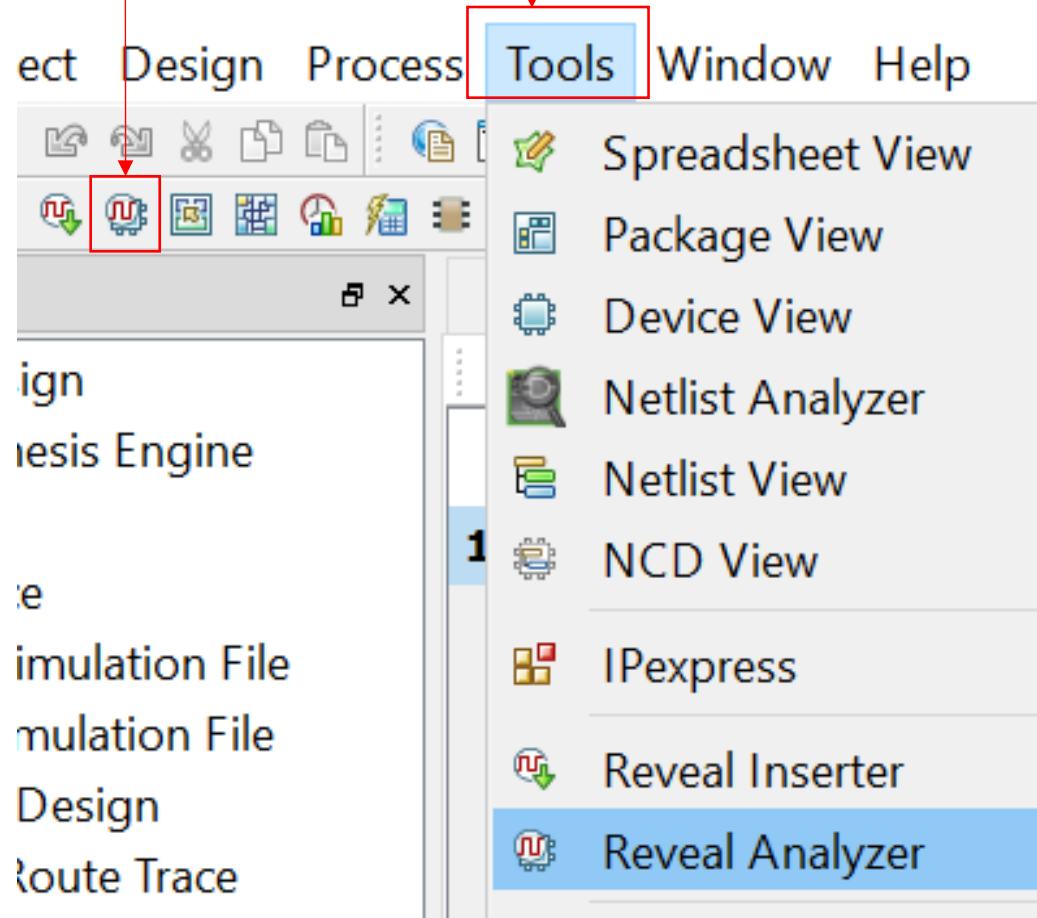
Available General Purpose Flash Memory:	511 Pages (Page 0 to Page 510).
Initialized UFM Pages:	0 Page.

Done: completed successfully

Re-program the MachXO2


Reveal

Either way



Reveal


Click to run

Ready  ☒ core_LA0

Trigger Unit


Name	Signals (MSB:LSB)
TU1	rst_btn

Wait for
trigger signal

Running  ☒ core_LA0

Bus/Signal	Data
rst_btn	
> clk_cnt	

Click to trigger again

Completed  ☒ core_LA0

Bus/Signal	Data	0:15	0:16	0:17	0:18	0:19	0:20
rst_btn	1						
> clk_cnt	012529B1	012529AF	012529B0	012529B1			00000000