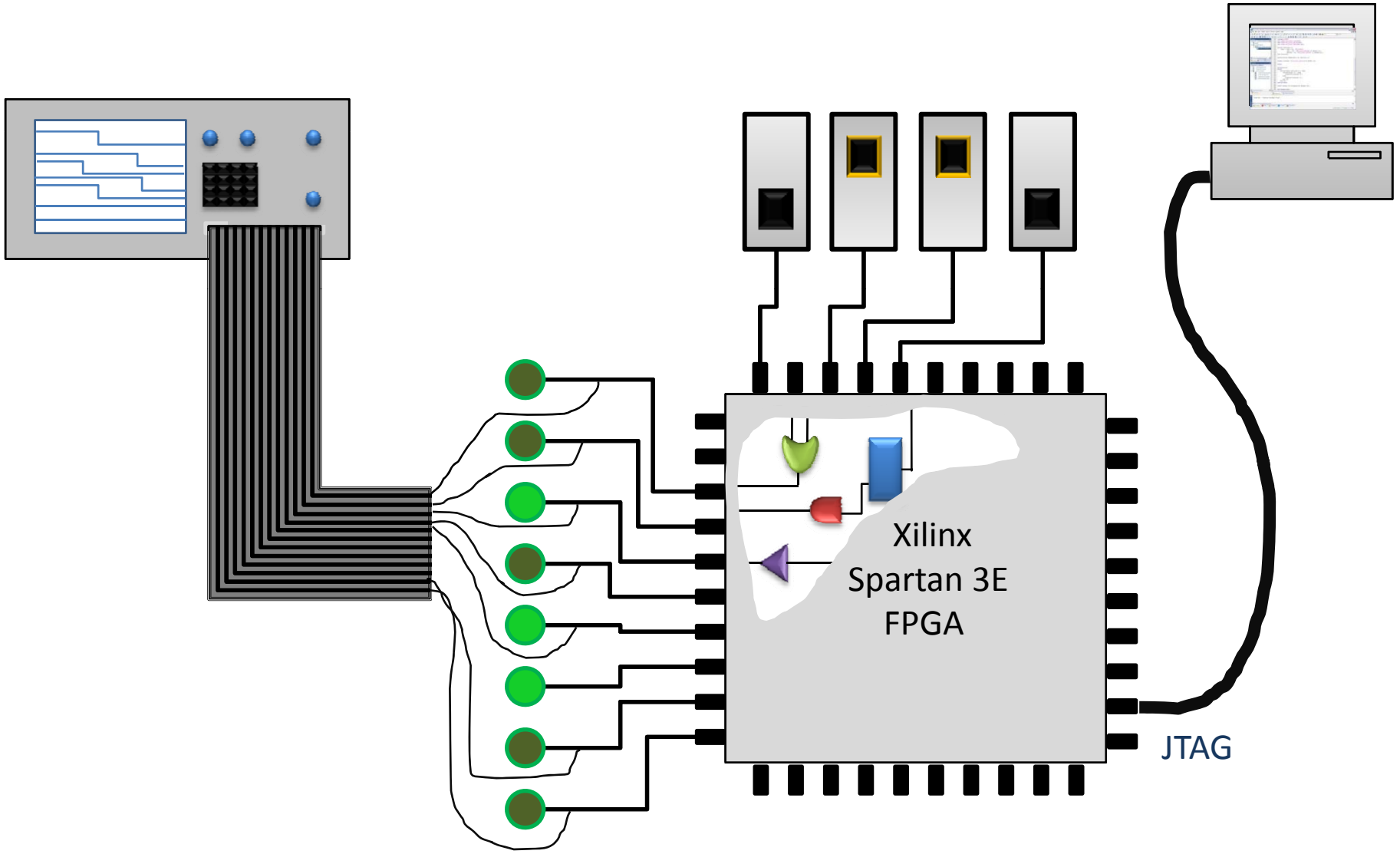

# COE758 - Xilinx ISE 9.2 Tutorial 2

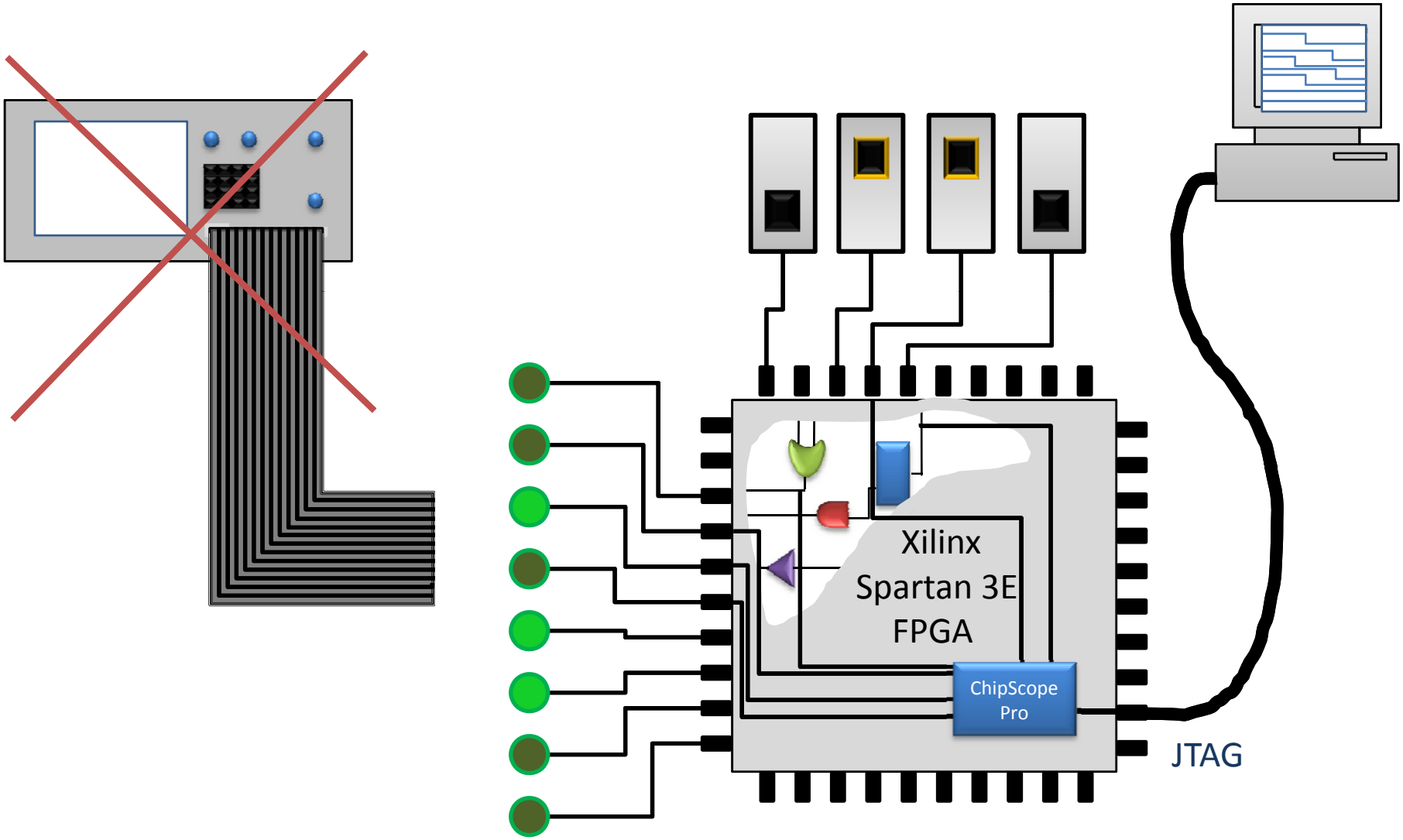
*ChipScope Overview*

*Integrating ChipScope Pro into a project*

# Conventional Signal Sampling



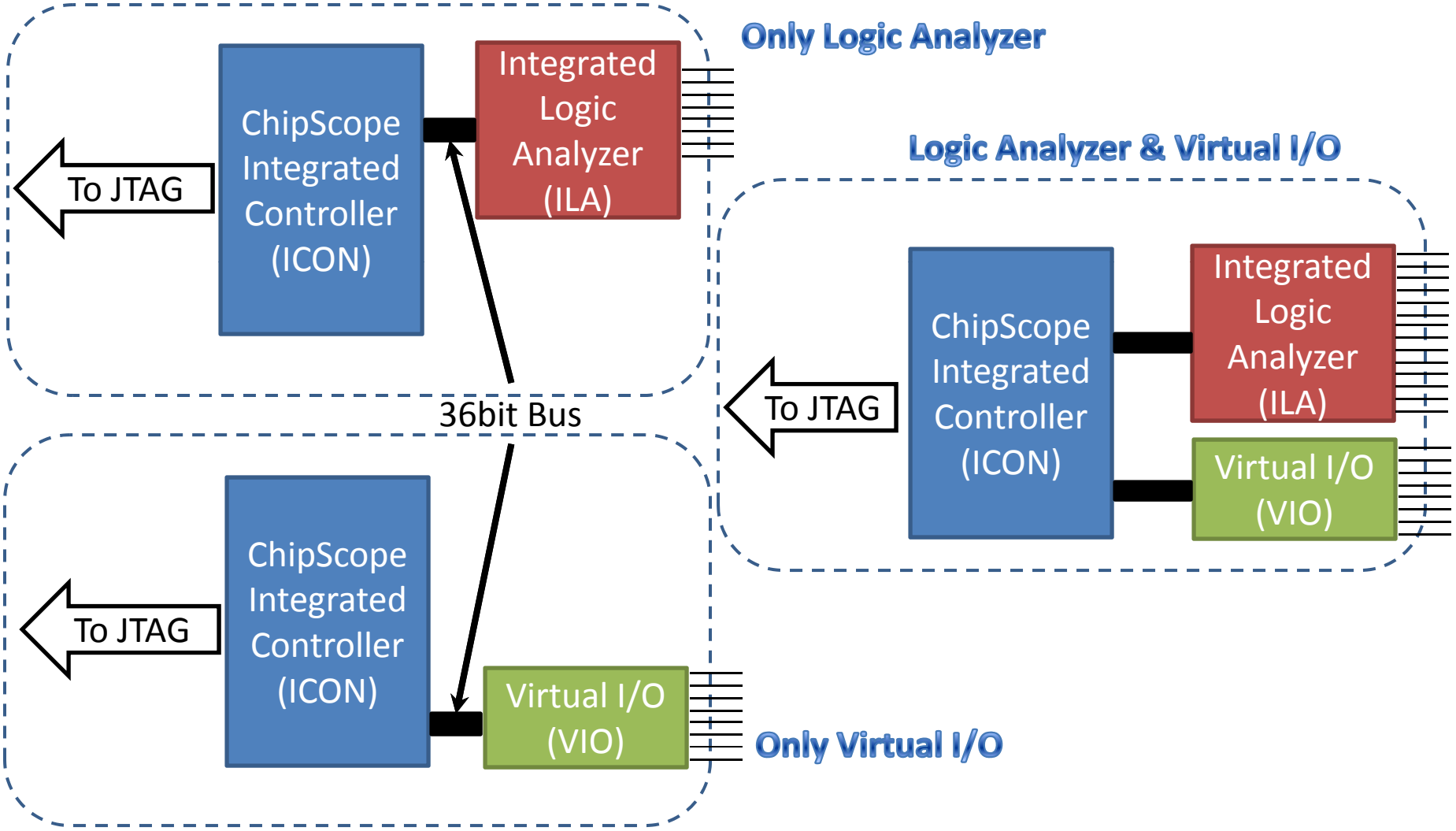
# ChipScope Pro Signal Sampling



# ChipScope Overview

- It is a software Logic Analyzer
- Can be integrated in ISE project as a component
- ChipScope has following features:
  - Can be connected to Inputs
  - Can be connected to Outputs
  - Can be connected to Intermediate signals
  - Can have a variable number of inputs/triggers
  - Records up to 16384 samples
  - Can create virtual internal inputs (such as buttons) that are activated from the GUI application on a PC

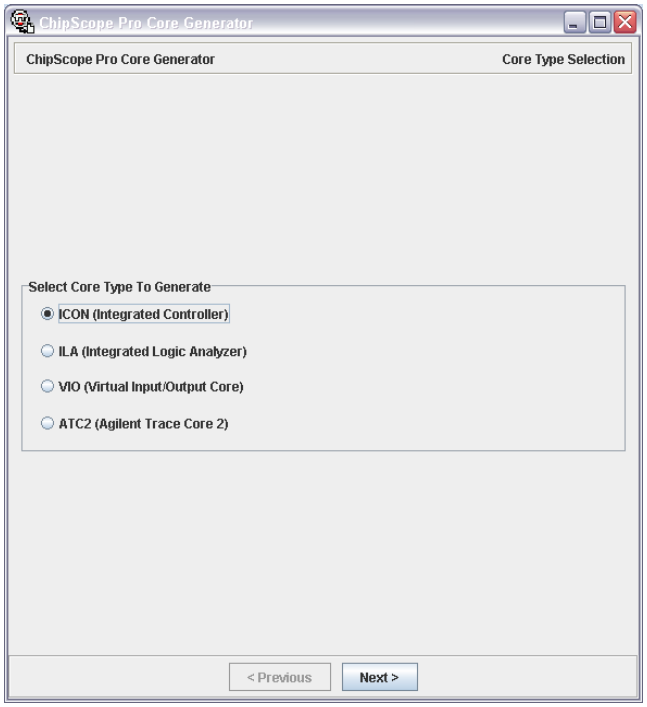
# ChipScope Configuration Variations

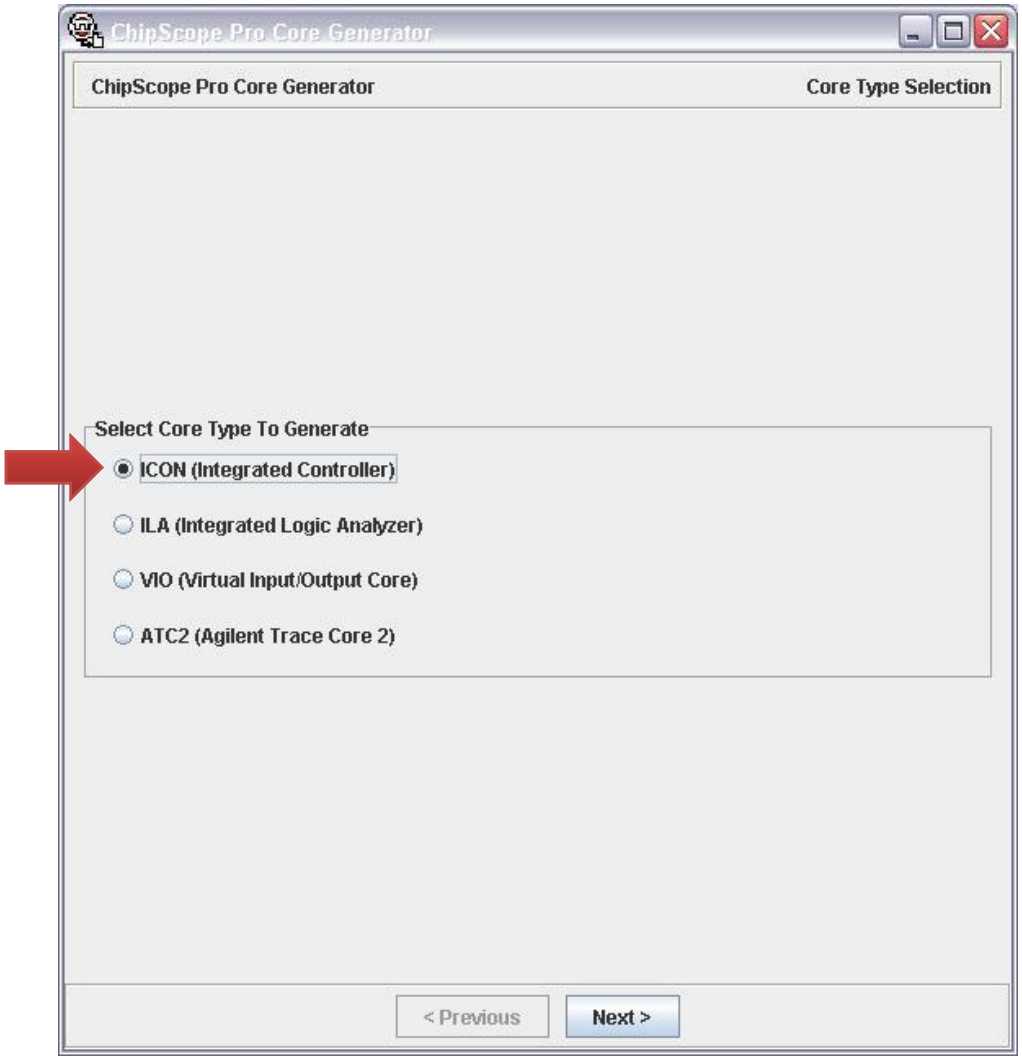


# Integration ChipScope ICON and ILA into Project from Tutorial 1

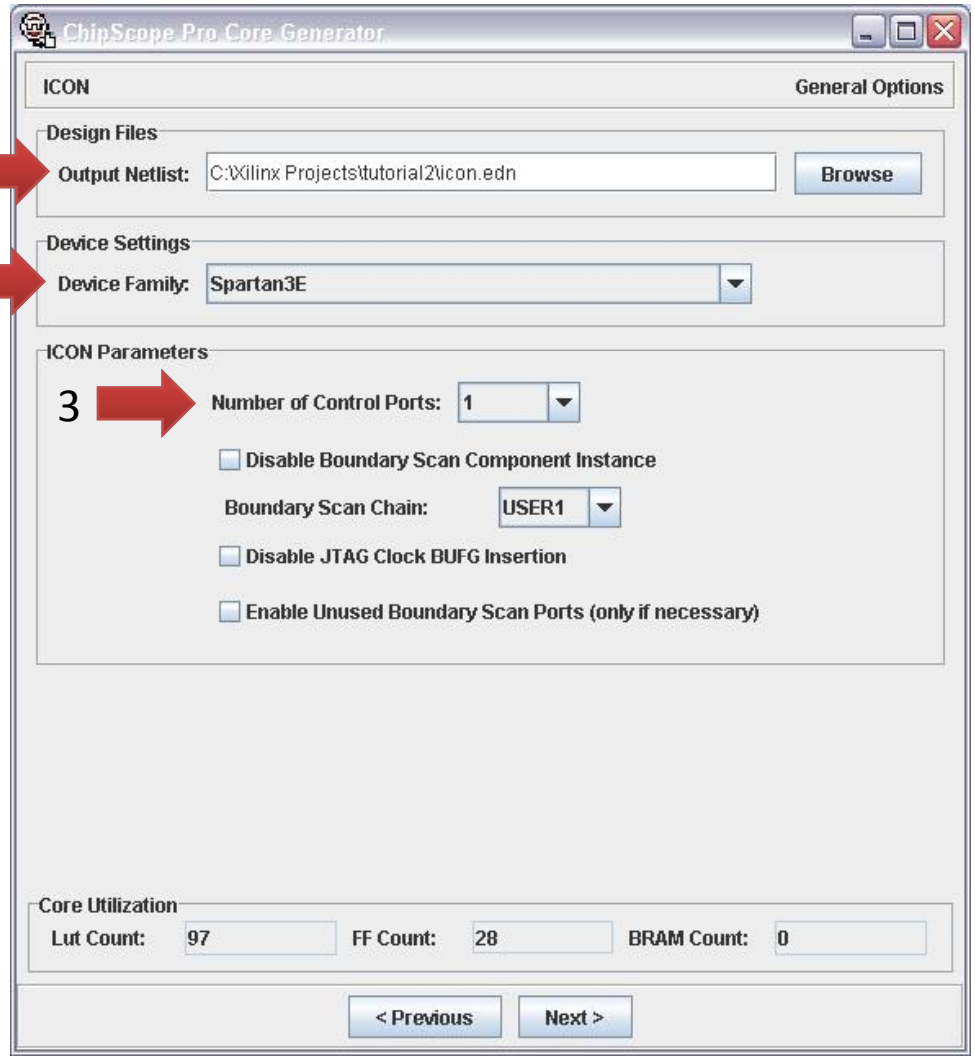
- Before inserting components we need to:
  - Generate ICON for the project
  - Generate ILA with appropriate amount of inputs, and triggers

Open ChipScope Pro Core Generator





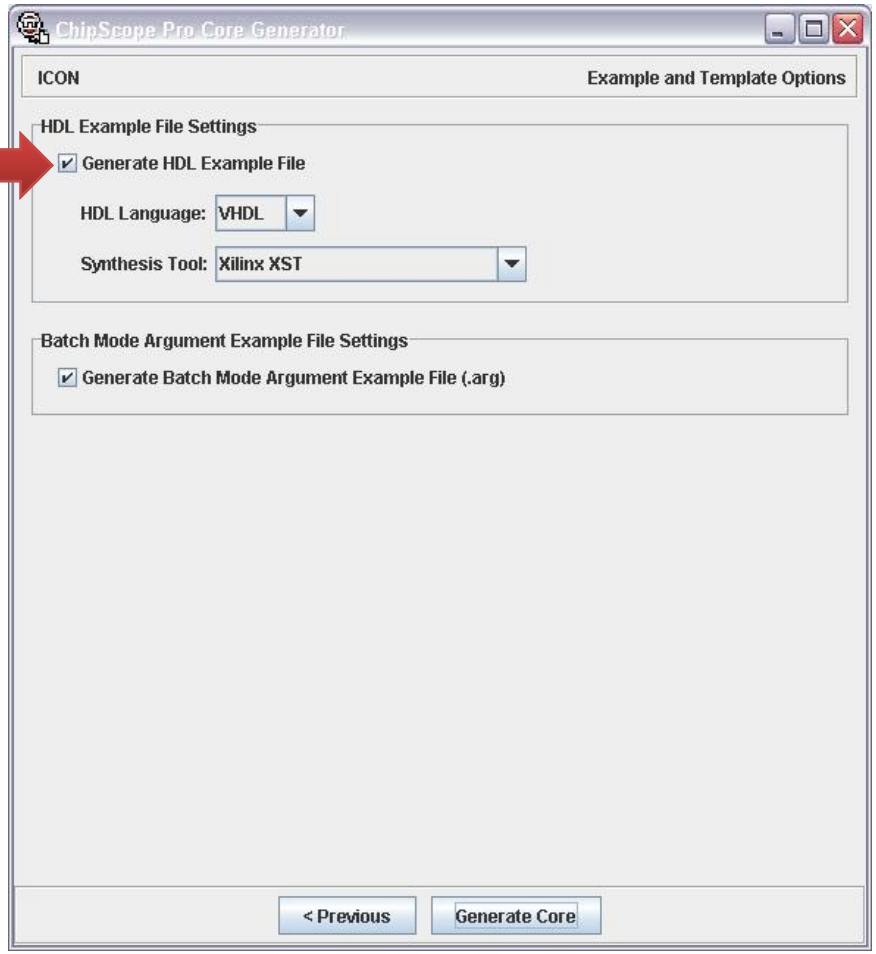

Select ICON (integrated Controller).
Press ***Next>***

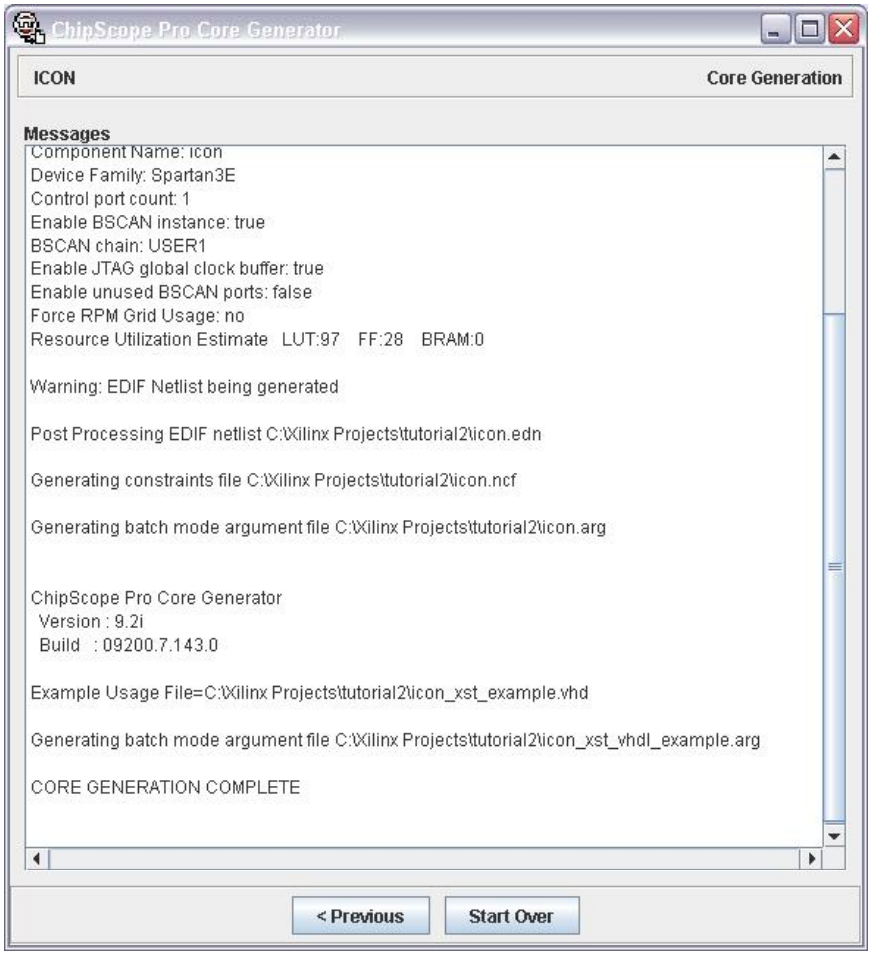



1
2
3

1. Specify directory where the project is located
2. Select appropriate FPGA device (Spartan3E)
3. Select number of ***Control Ports***. In this case it is 1 since we only have ILA

Press ***Next>***

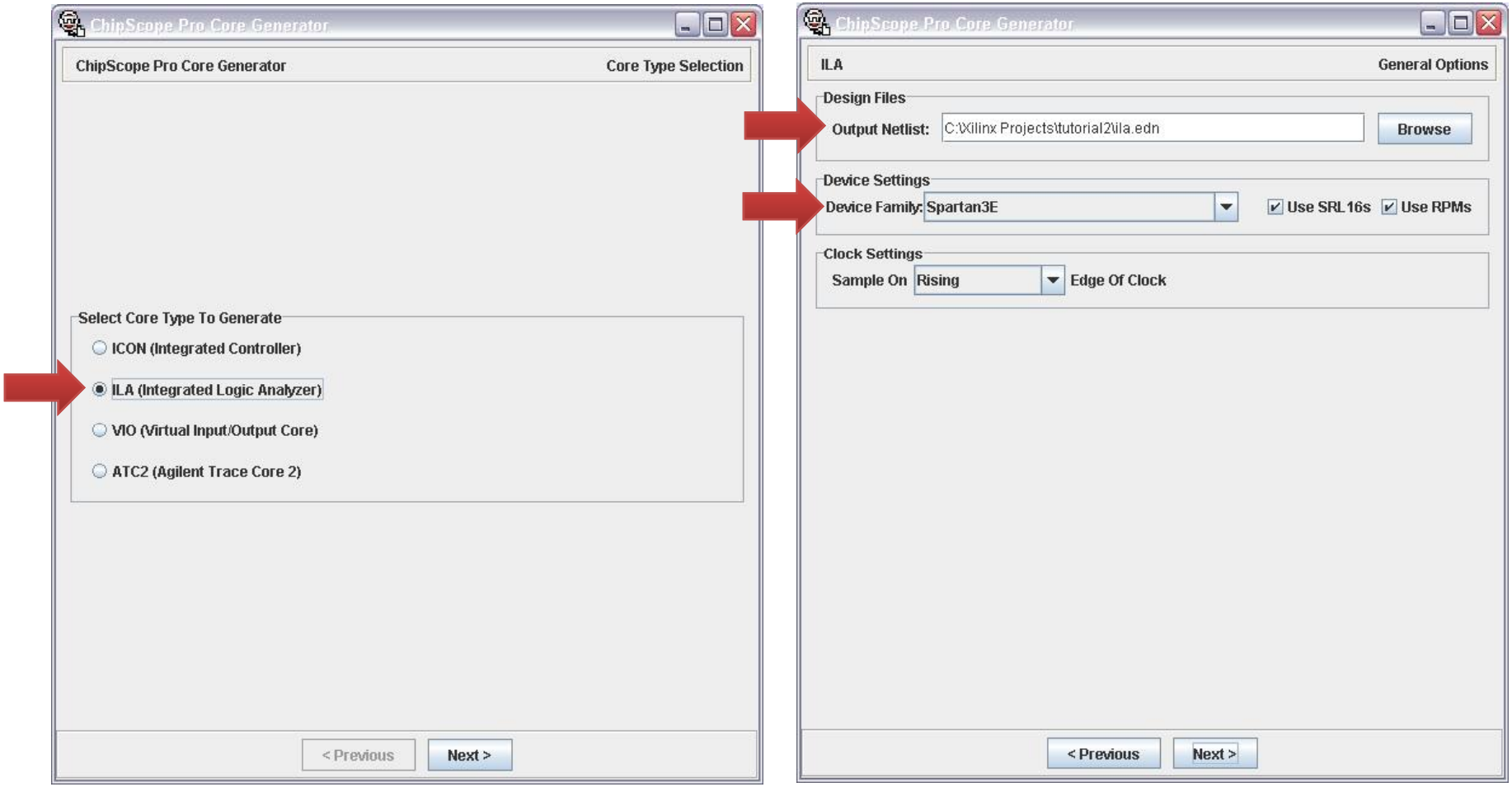



By selecting ***Generate HDL Example File*** ChipScope Pro Core generator will also generate the template which you will be able to copy directly to your design, thus minimizing possible errors.

Press ***Generate Core>*** this will generate core and all required files

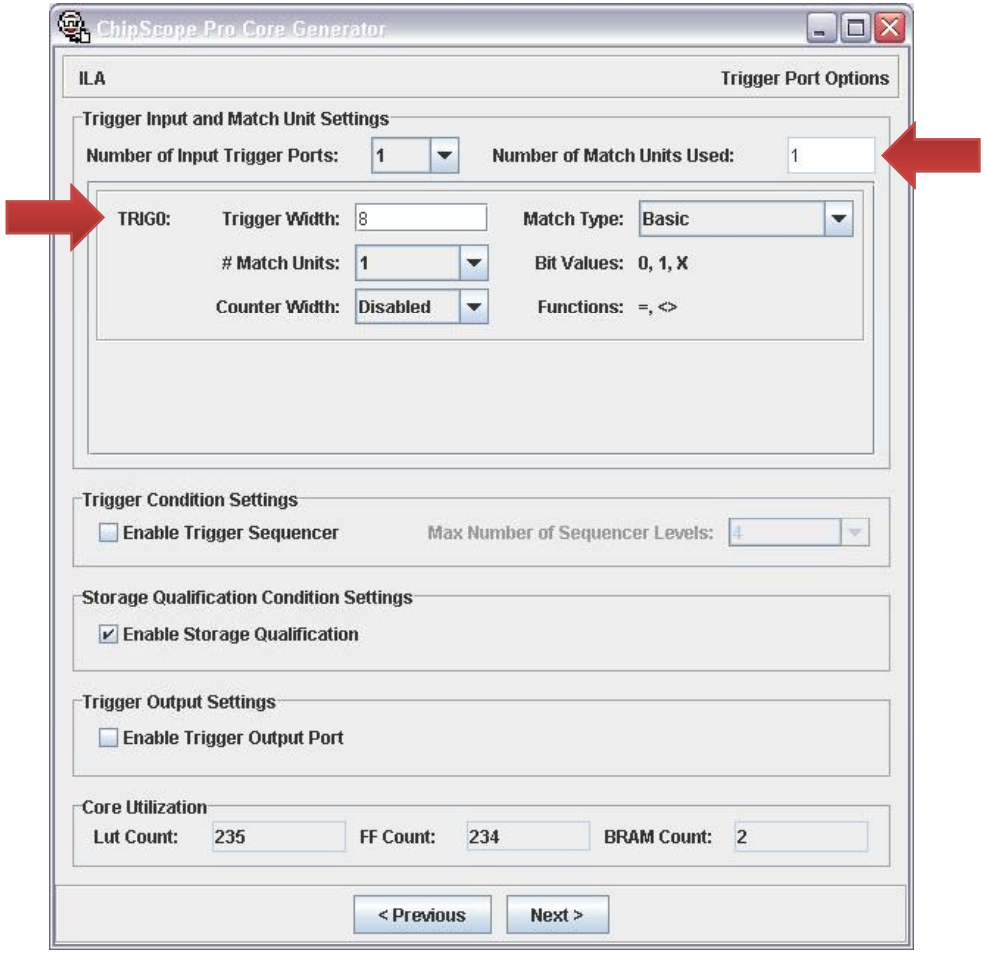
Press ***Start Over*** to continue in generating ILA component




This time select ***ILA (Integrated Logic Analyzer)*** and Press ***Next>***

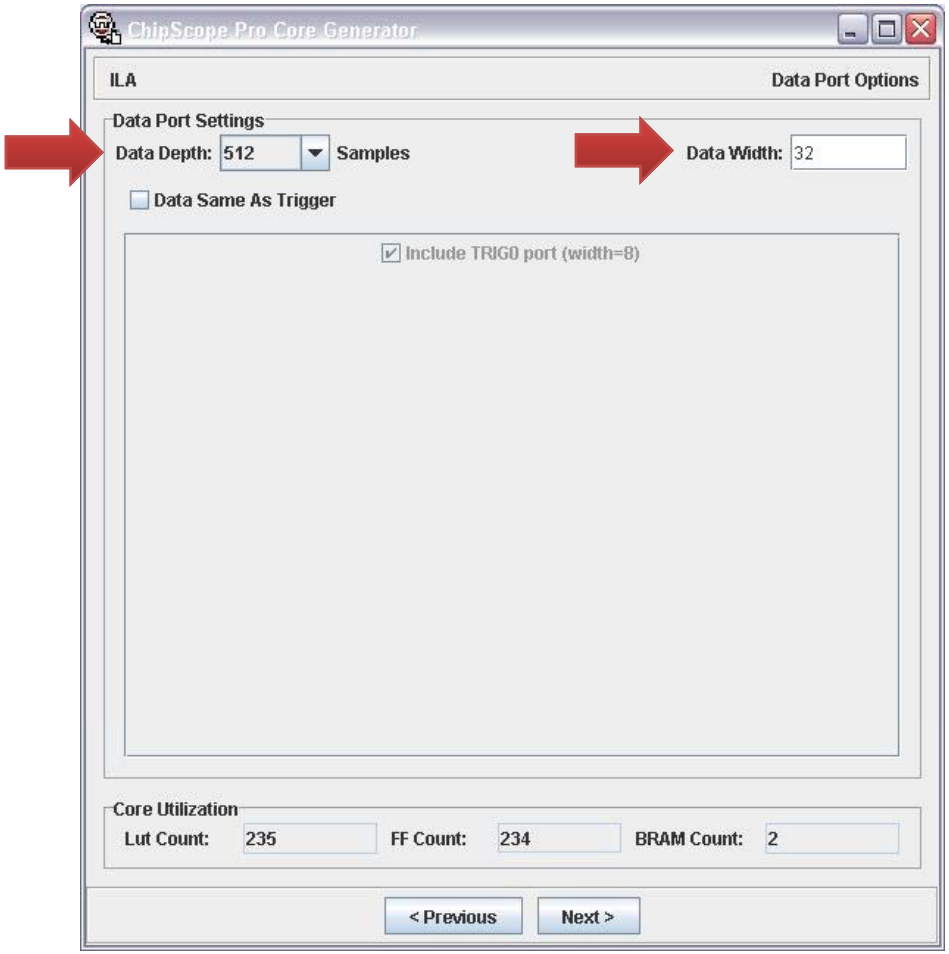
As before, make sure that ***Output Netlist:*** is placed in the project directory and ***Spartan3E*** is selected as ***Device Family***. By default these settings should be already present.

Press ***Next>***



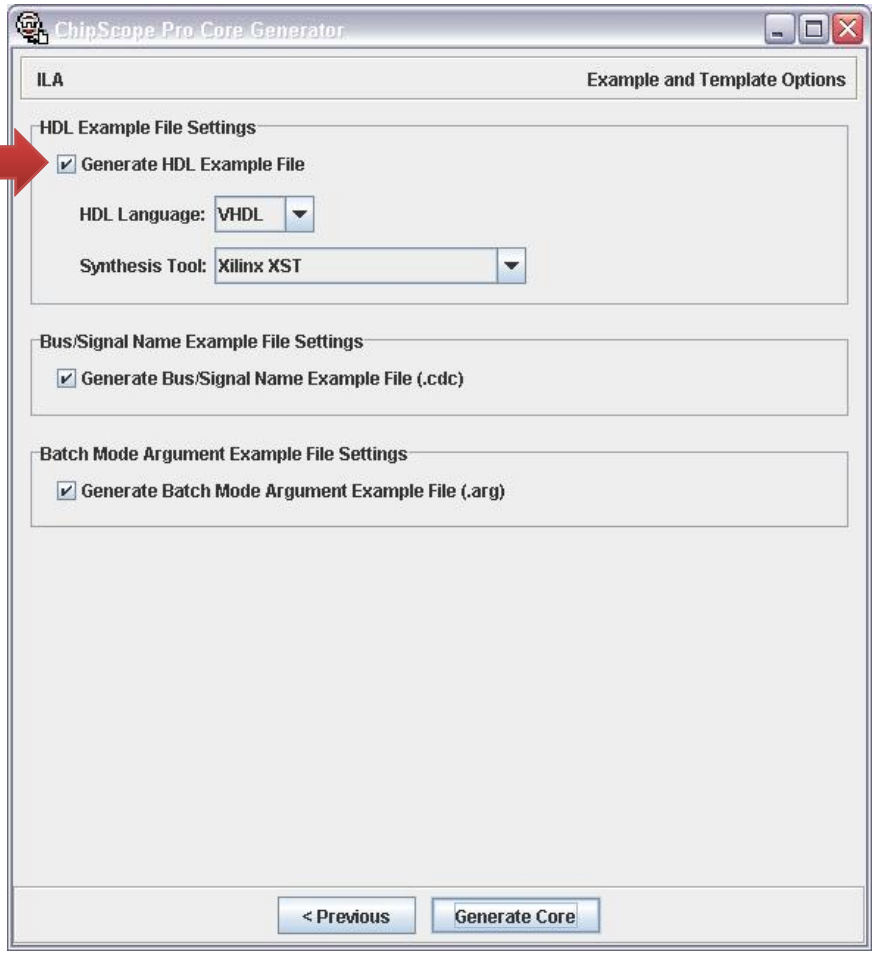

You can select number of triggers which can be setup to initiate the signal capture, same as you would do on physical logic analyzer. There are many options such as being able to have several trigger matches, but for now we will use a single match unit, and 8 bit trigger.

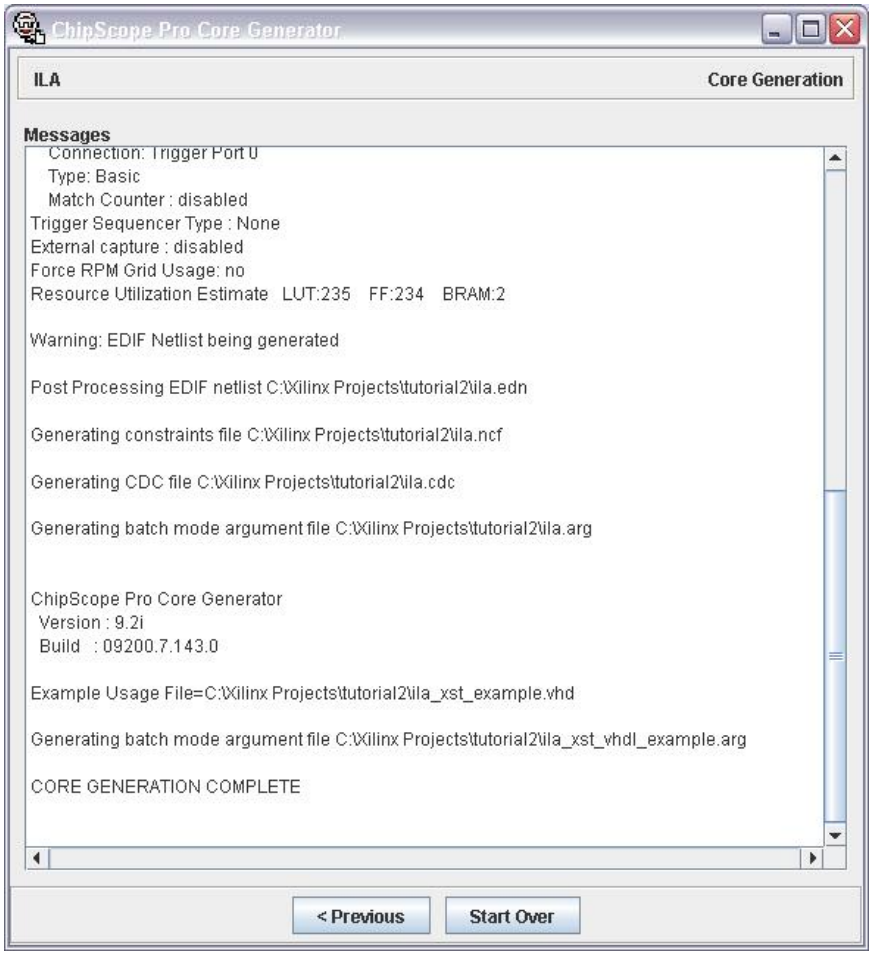
Press ***Next>***



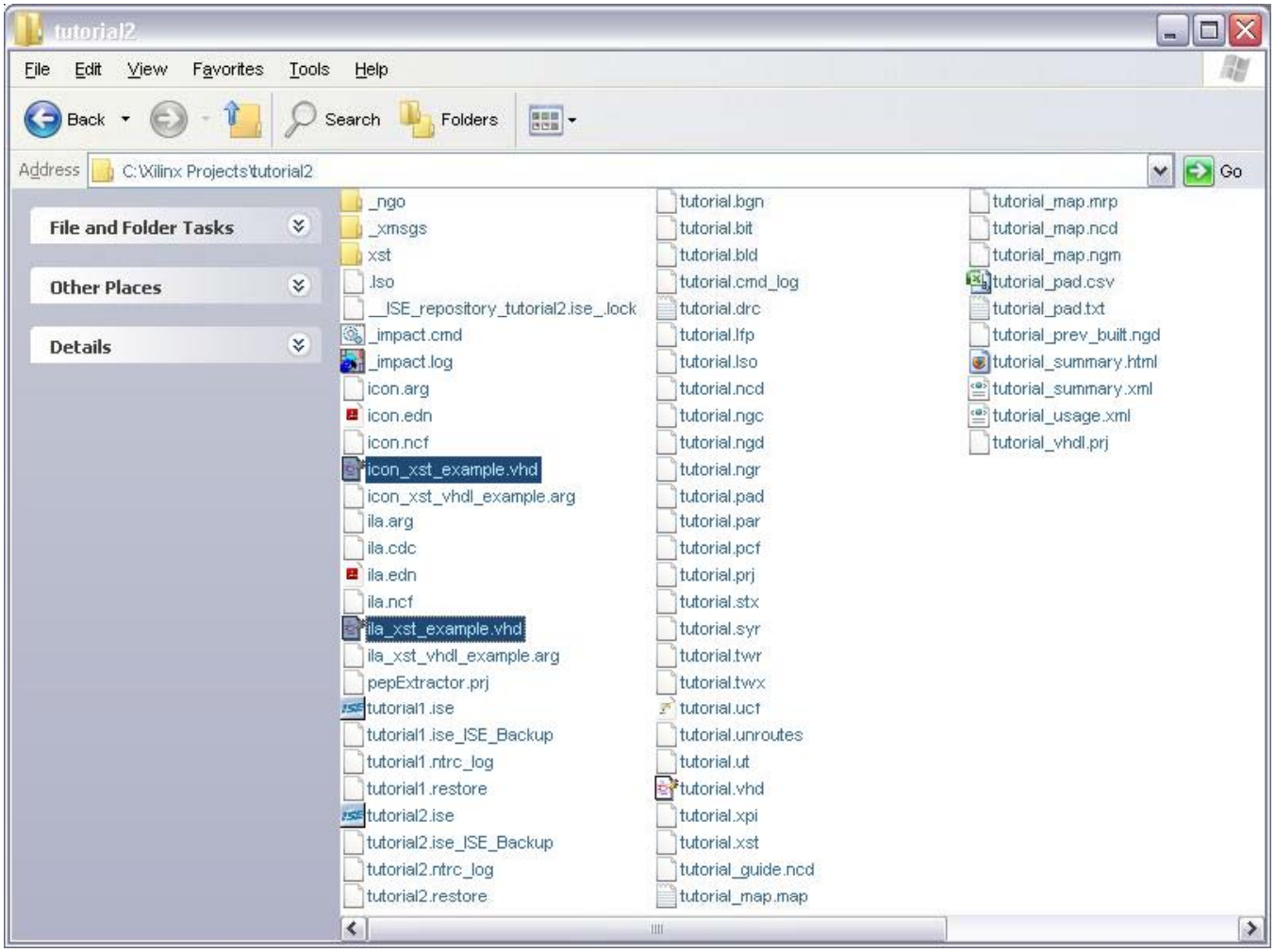

In this window you select number of samples that you want to be recorded at any given capture, and number of sample signal channels. For now we will pick 512 ***Samples*** with 32 bit ***Data Width.*** Note that increasing both of these parameters will require more internal memory, which might be needed by your design, therefore only use as much as you need.

Press ***Next>***

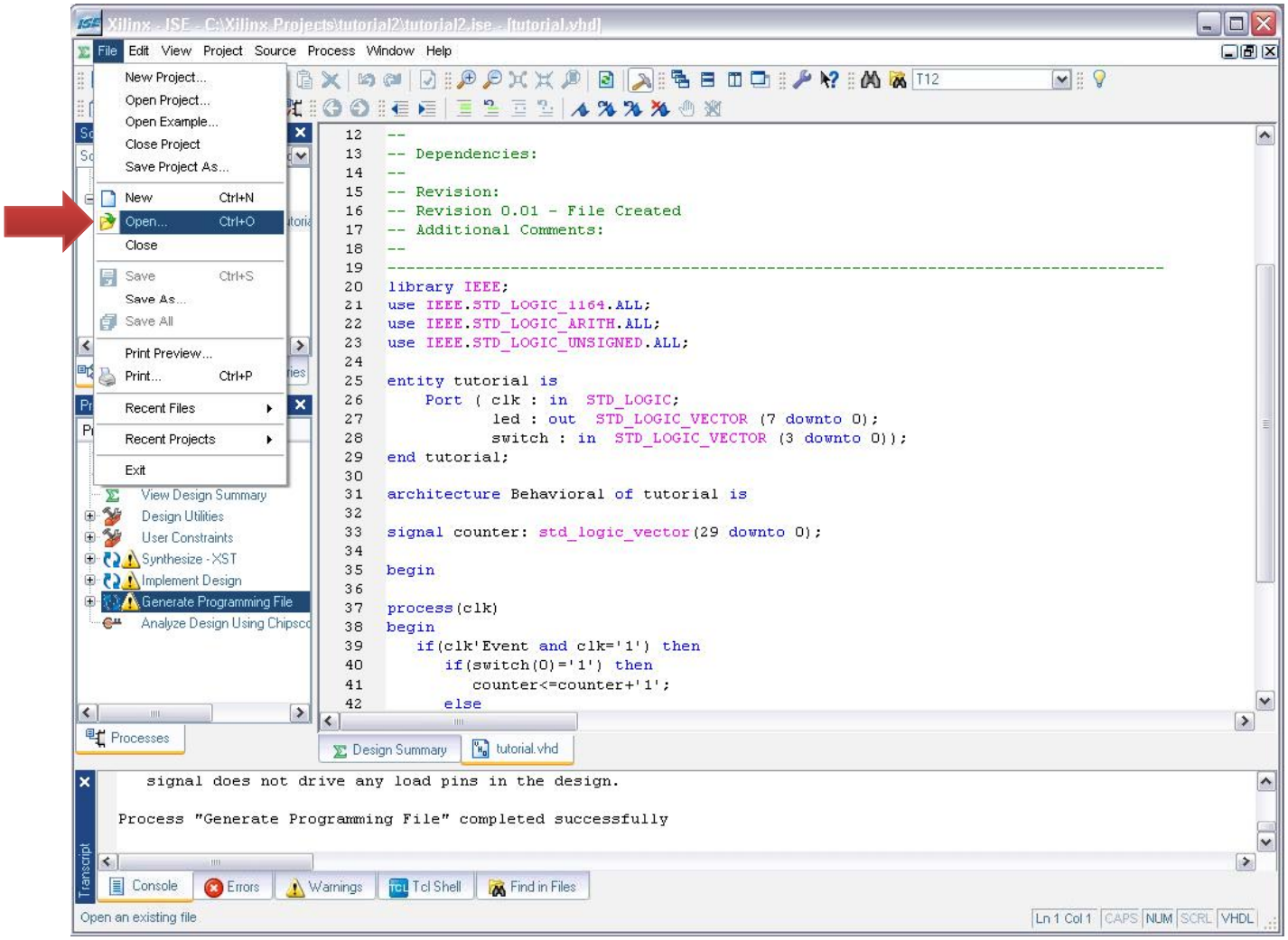



Same as for the ***ICON*** select ***Generate HDL Example File*** and
Press ***Generate Core***
After core is generated you can close ChipScope Pro Core Generator and go back to the project to start integrating ChipScope Pro components into it.

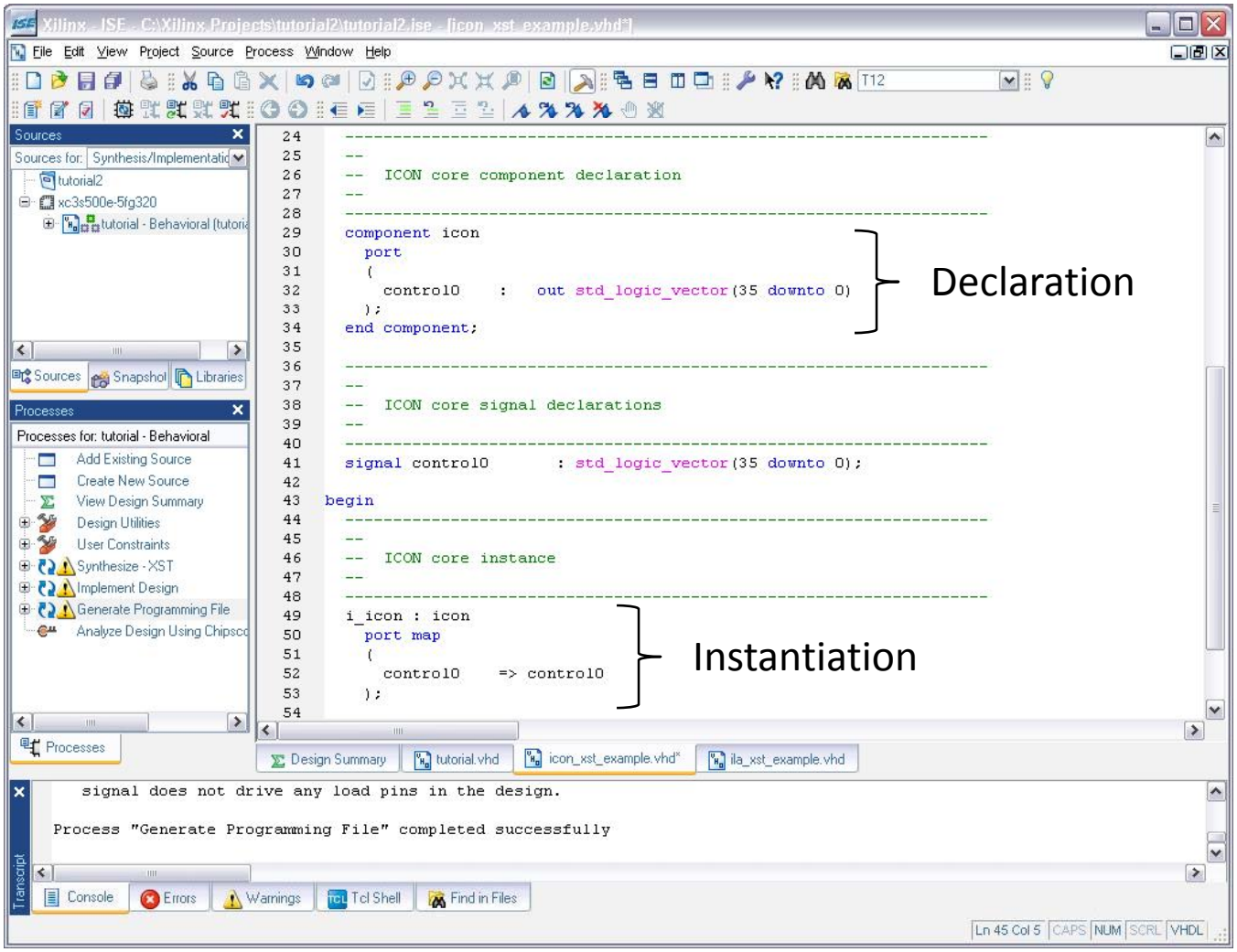



In the project directory you will notice that there are two VHDL example files for the ICON and ILA modules.

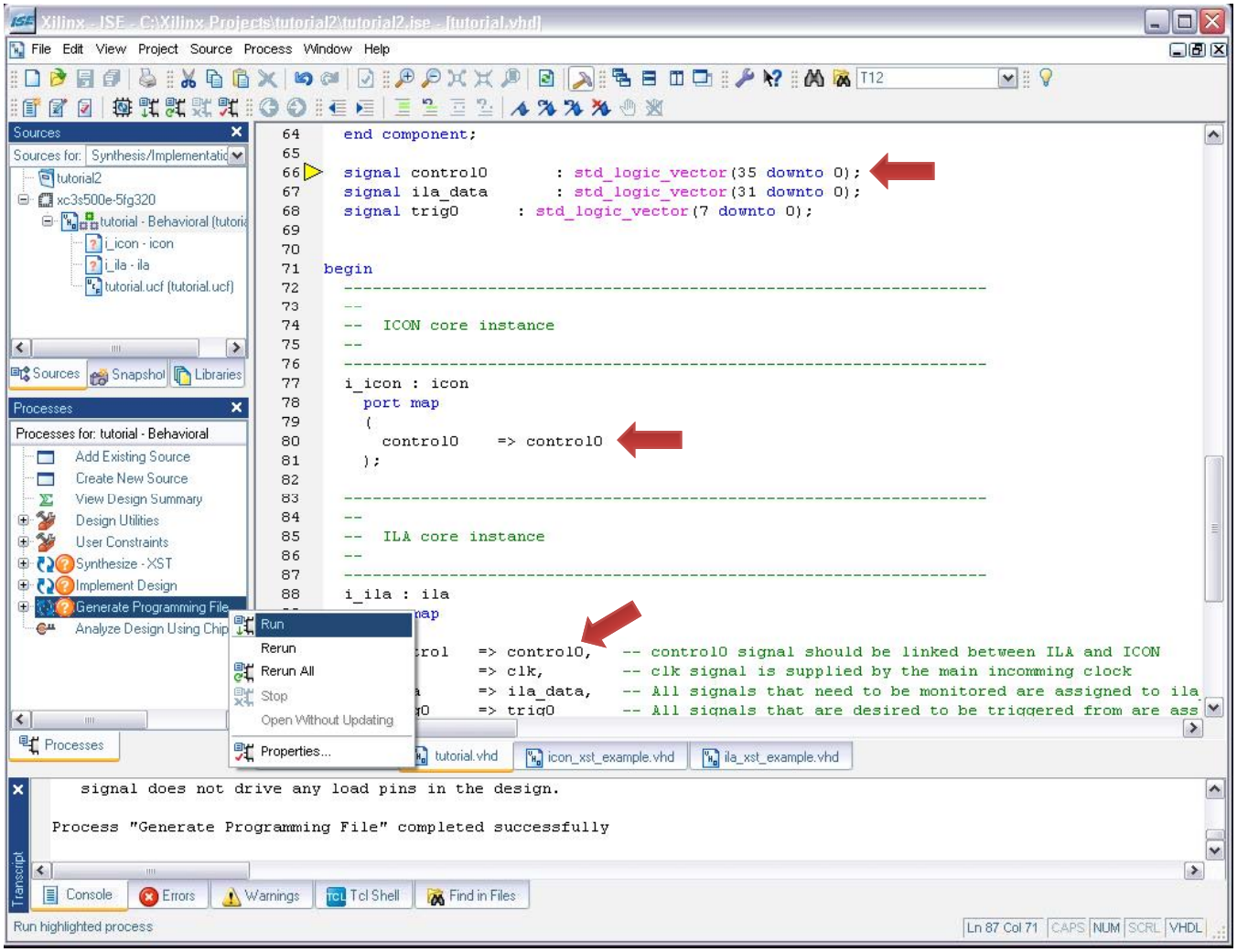



```
12  --
13  -- Dependencies:
14  --
15  -- Revision:
16  -- Revision 0.01 - File Created
17  -- Additional Comments:
18  --
19  ----------------------------------------------------------------------------------
20  library IEEE;
21  use IEEE.STD_LOGIC_1164.ALL;
22  use IEEE.STD_LOGIC_ARITH.ALL;
23  use IEEE.STD_LOGIC_UNSIGNED.ALL;
24
25  entity tutorial is
26      Port ( clk : in  STD_LOGIC;
27             led : out  STD_LOGIC_VECTOR (7 downto 0);
28             switch : in  STD_LOGIC_VECTOR (3 downto 0));
29  end tutorial;
30
31  architecture Behavioral of tutorial is
32
33  signal counter: std_logic_vector(29 downto 0);
34
35  begin
36
37  process(clk)
38  begin
39     if(clk'Event and clk='1') then
40        if(switch(0)='1') then
41           counter<=counter+'1';
42        else
```

You will need to open the example files either in a text editor, or directly in the Xilinx ISE.

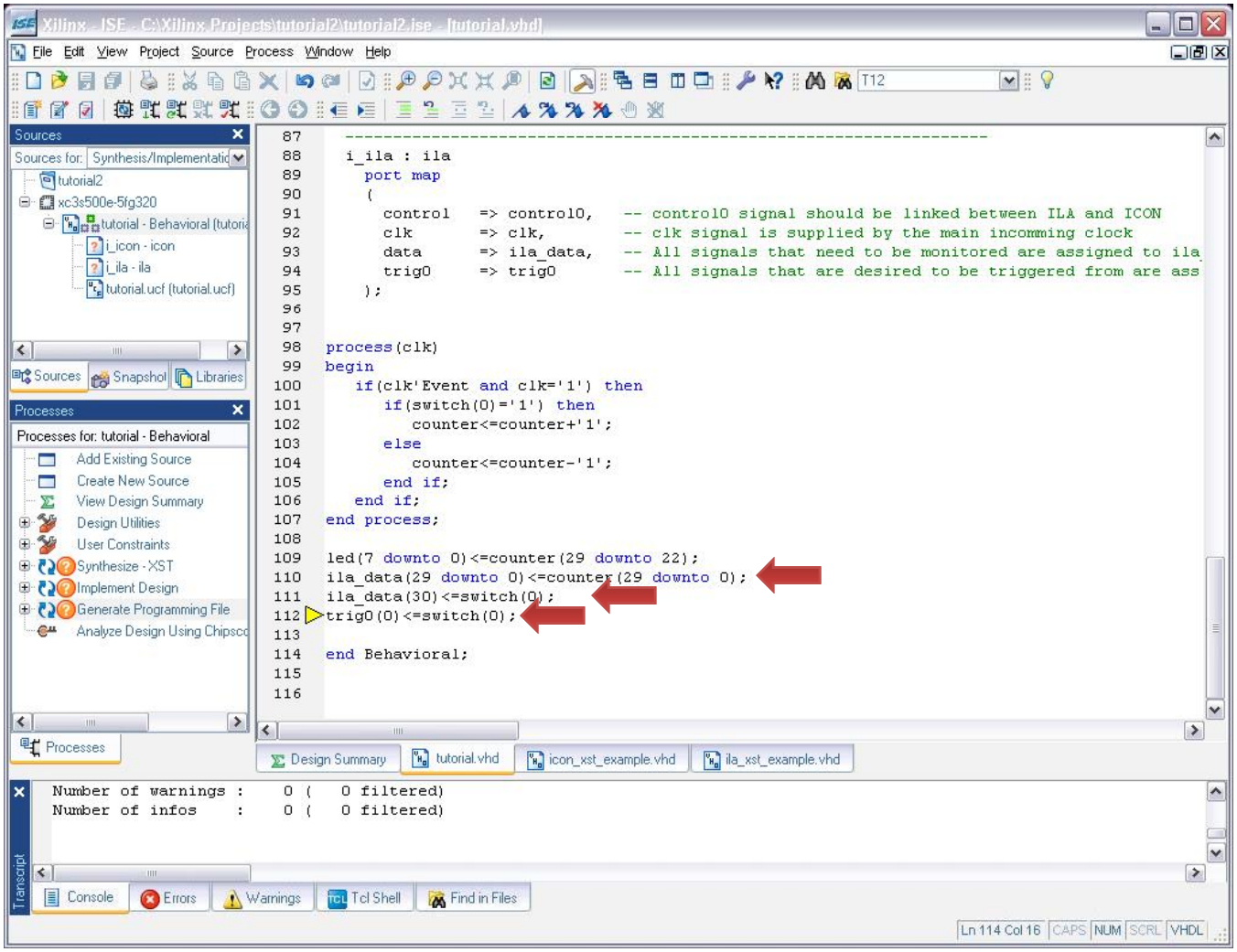



In these files you will find component declaration, and instantiation as shown in the window above. These sections can be directly copied to the project which in turn will initialize these components in your project.




ICON and ILA modules are linked by the common communication Bus ***control0.*** ila_data and trig0 Busses are connected to the actual inputs/outputs/signals in the design. Directory of all of the files can be downloaded from the coarse website.

After all of the components insertions Run Generate Programming File to make sure that there are no errors.



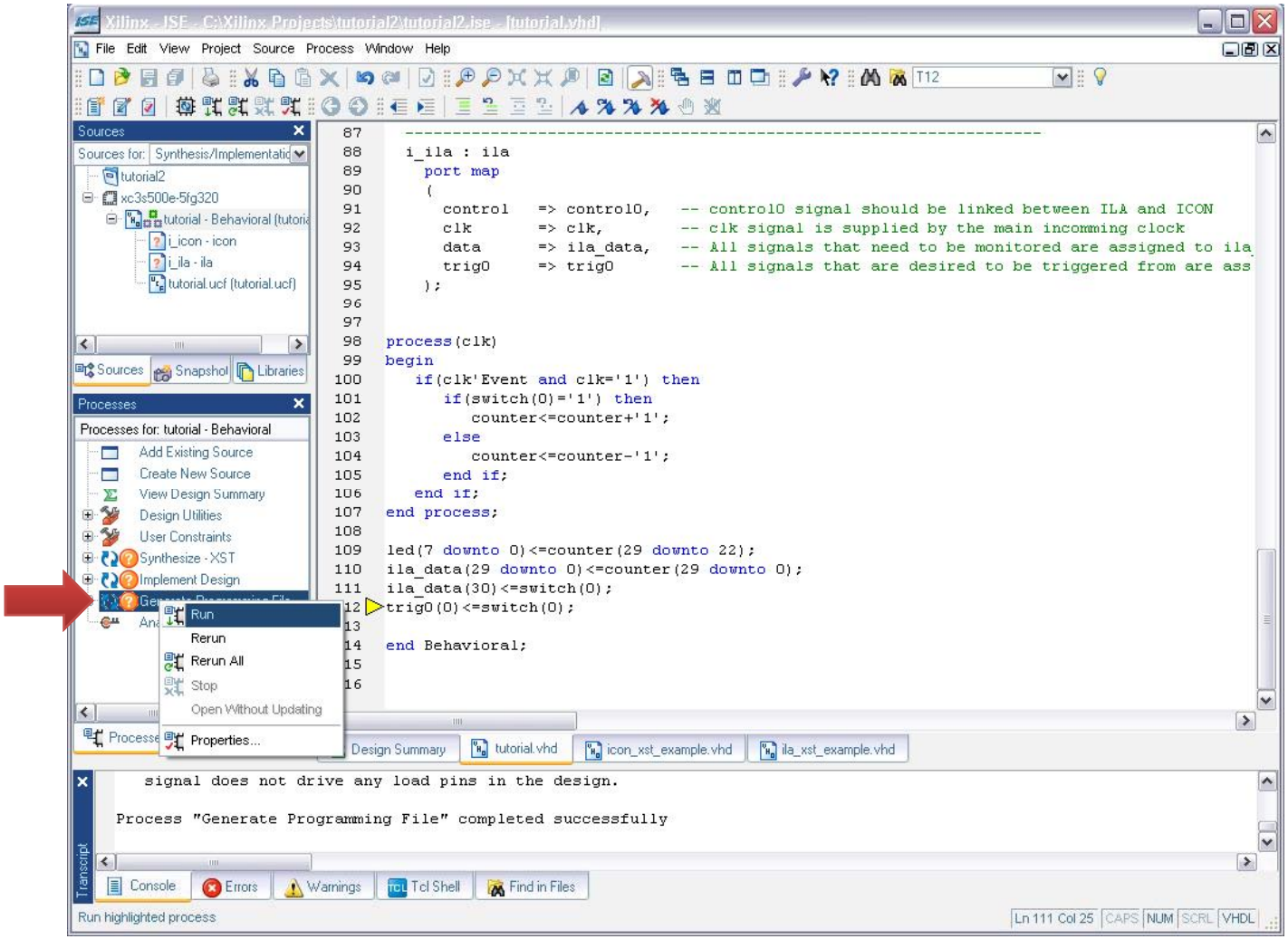

Output of internal signals of the ***counter*** to the ***ila_data***, as well as the input of the ***switch(0)***.

At the same time, set first bit of the trigger ***trig0*** to ***switch(0)***, so as an example when switch is set to ***'1'*** signals capturing will commence.

# Final Source Code

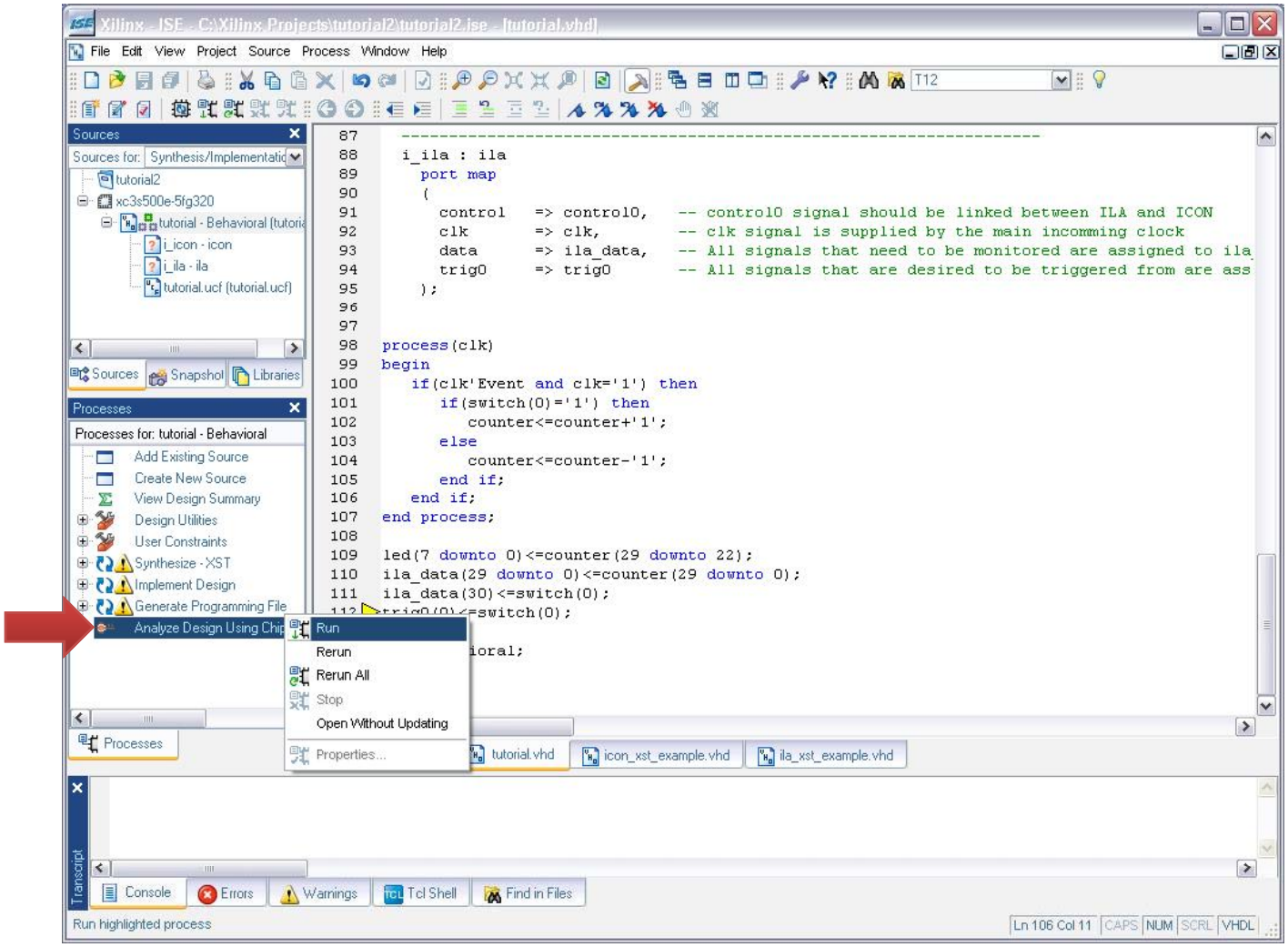
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity tutorial is
   Port ( clk : in  STD_LOGIC;
          led : out  STD_LOGIC_VECTOR (7 downto 0);
          switch : in  STD_LOGIC_VECTOR (3 downto 0));
end tutorial;

architecture Behavioral of tutorial is

component icon
   port
   (
     control0  : out std_logic_vector(35 downto 0));
end component;

component ila
   port
   (
     control    : in    std_logic_vector(35 downto 0);
     clk        : in    std_logic;
     data       : in    std_logic_vector(31 downto 0);
     trig0      : in    std_logic_vector(7 downto 0));
 end component;

signal counter: std_logic_vector(29 downto 0);
signal control0       : std_logic_vector(35 downto 0);
signal ila_data       : std_logic_vector(31 downto 0);
signal trig0      : std_logic_vector(7 downto 0);
```

```
begin

 i_icon : icon
 port map
  (
control0   => control0);

 i_ila : ila
  port map
  (
    control   => control0,   -- should be linked between ILA and ICON
    clk       => clk,        -- signal is supplied by the main clock
    data      => ila_data,
-- All signals that need to be monitored are assigned to ila_data Bus
    trig0     => trig0
-- All signals that are desired to be triggered from are assigned to trig0
   );

  process(clk)
  begin
     if(clk'Event and clk='1') then
           if(switch(0)='1') then
                 counter<=counter+'1';
            else
                 counter<=counter-'1';
             end if;
     end if;
  end process;

  led(7 downto 0)<=counter(29 downto 22);
  ila_data(29 downto 0)<=counter(29 downto 0);
  ila_data(30)<=switch(0);
  trig0(0)<=switch(0);

end Behavioral;
```

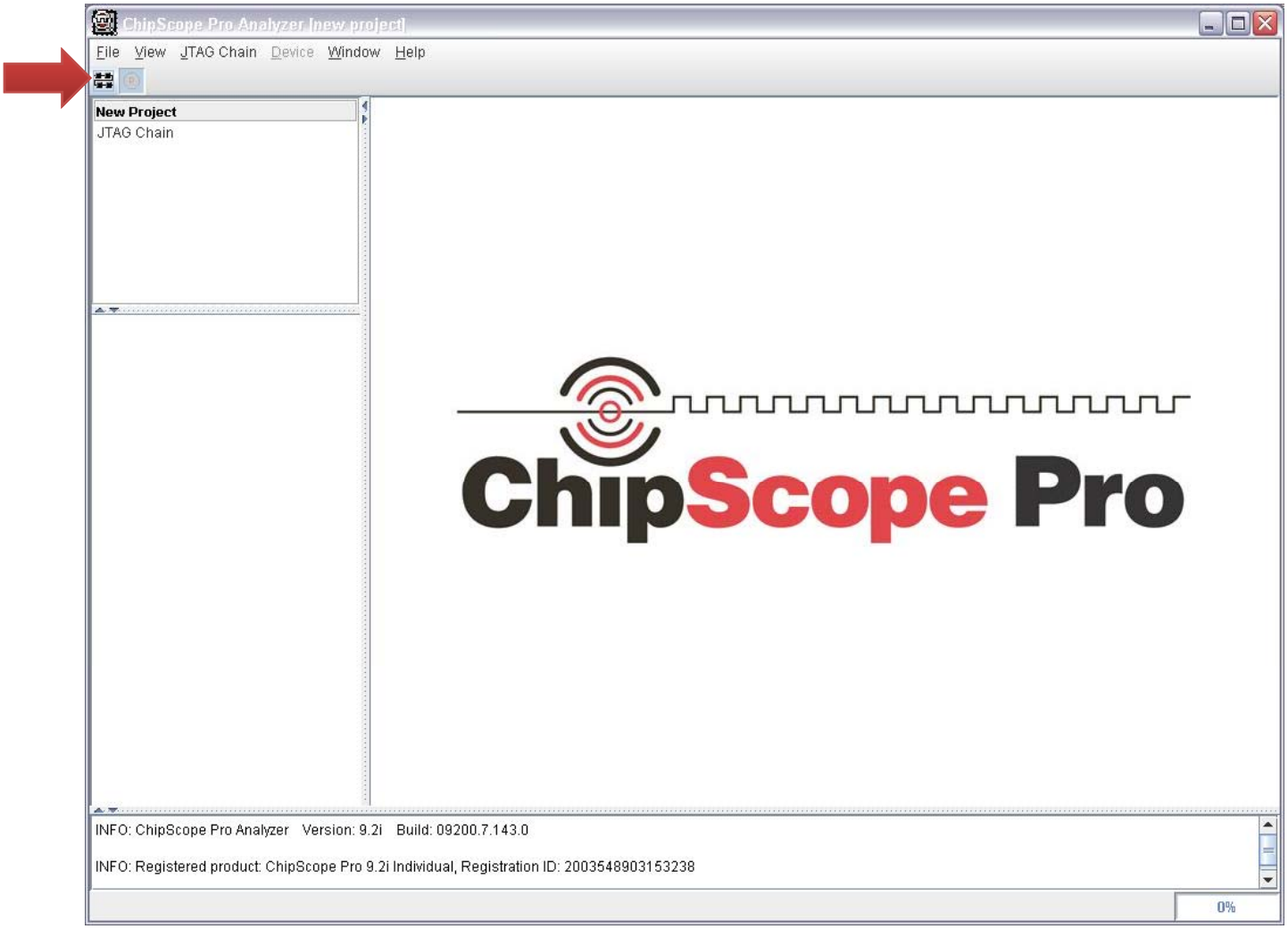



After all of the ILA signals were assigned ***Run*** the ***Generate Programming File*** and correct all of the errors that are encountered.

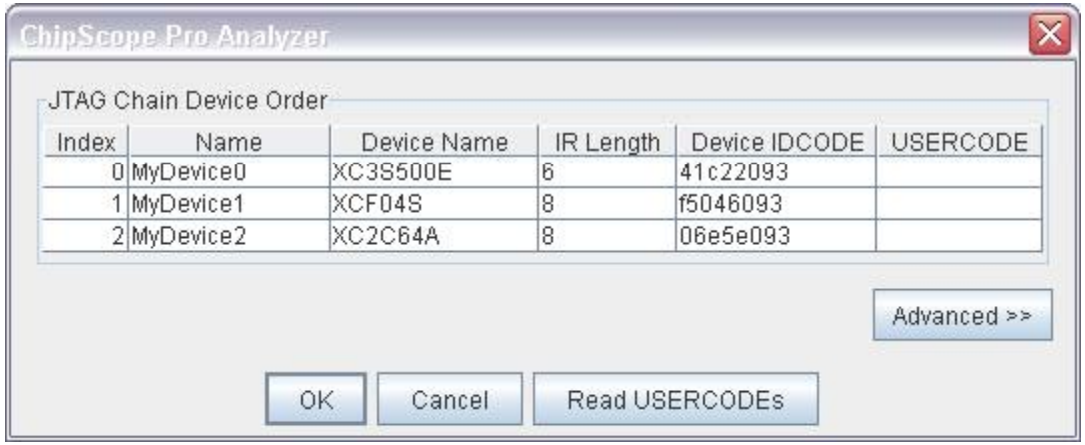
At this point platform can be configured with the configuration file and ChipScope Pro can be used to sample the signals.



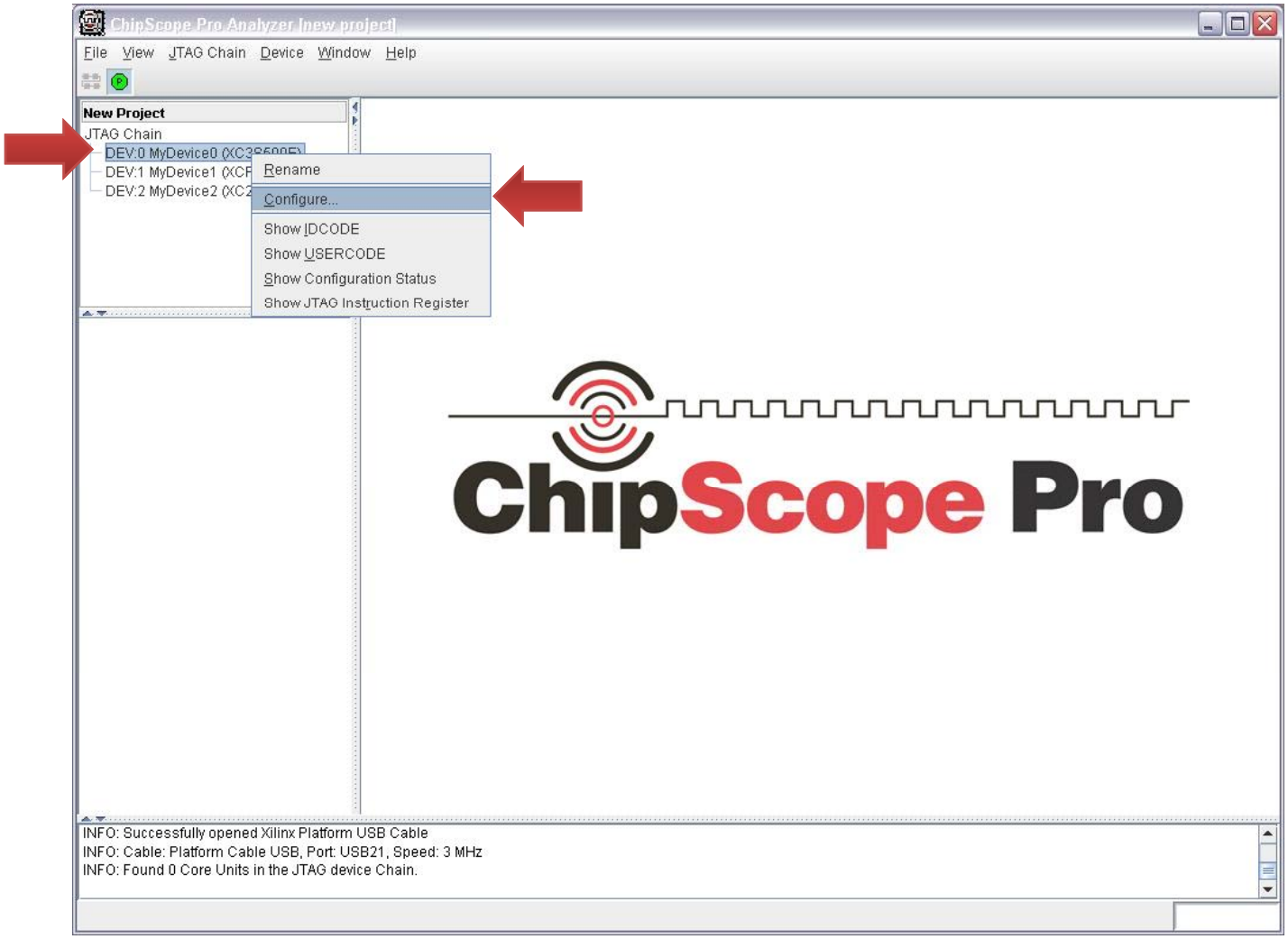

Instead of going through the ***iMPACT*** double click on the ***Analyze Design Using ChipScope.***



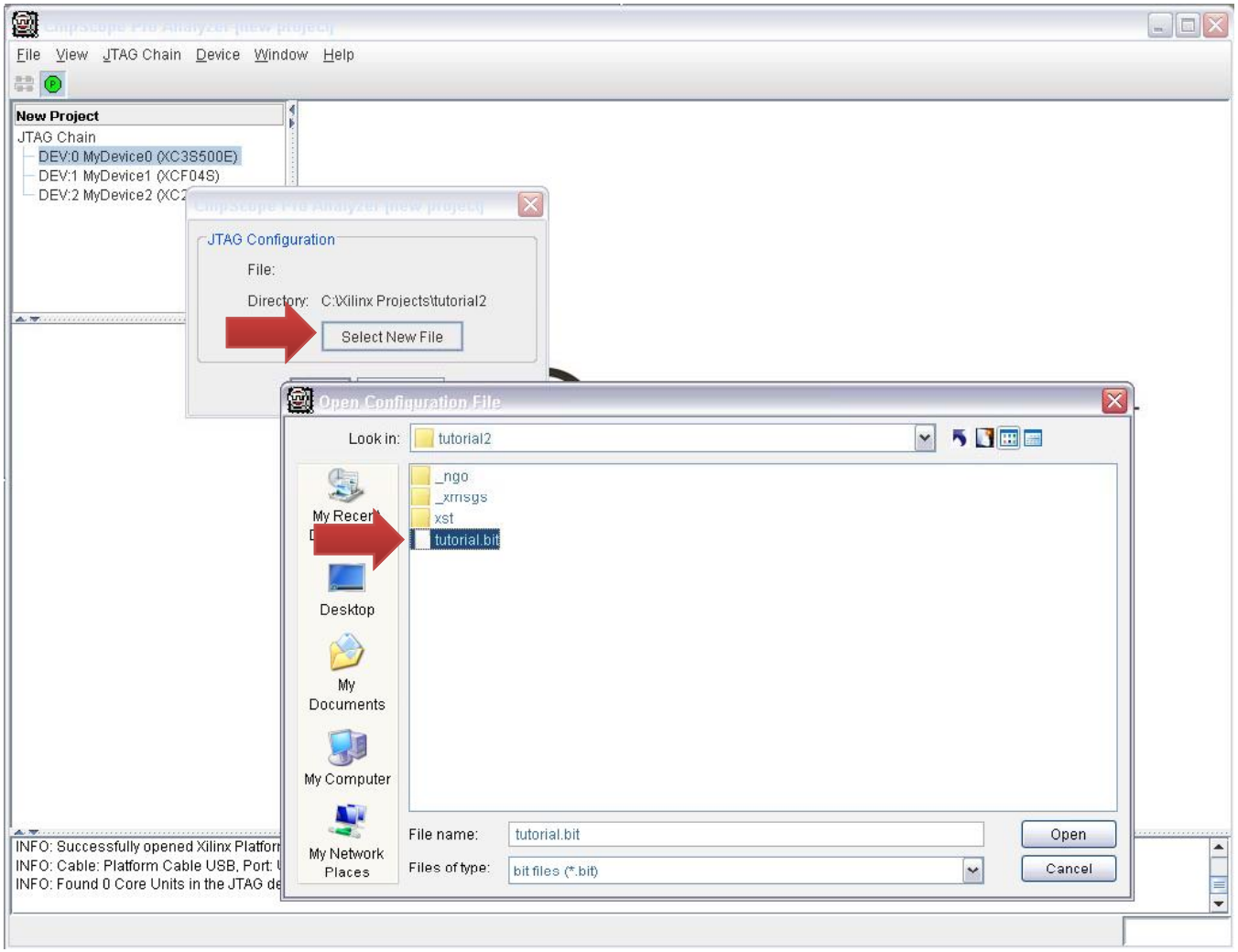

When the ChipScope Pro is loaded click on the JTAG scan button at the top left corner. This will initiate communication with the Spartan 3E platform, and identify all of the devices on the JTAG chain.




When communication is successful a window with a list of devices on the JTAG chain will appear.
Press ***OK***

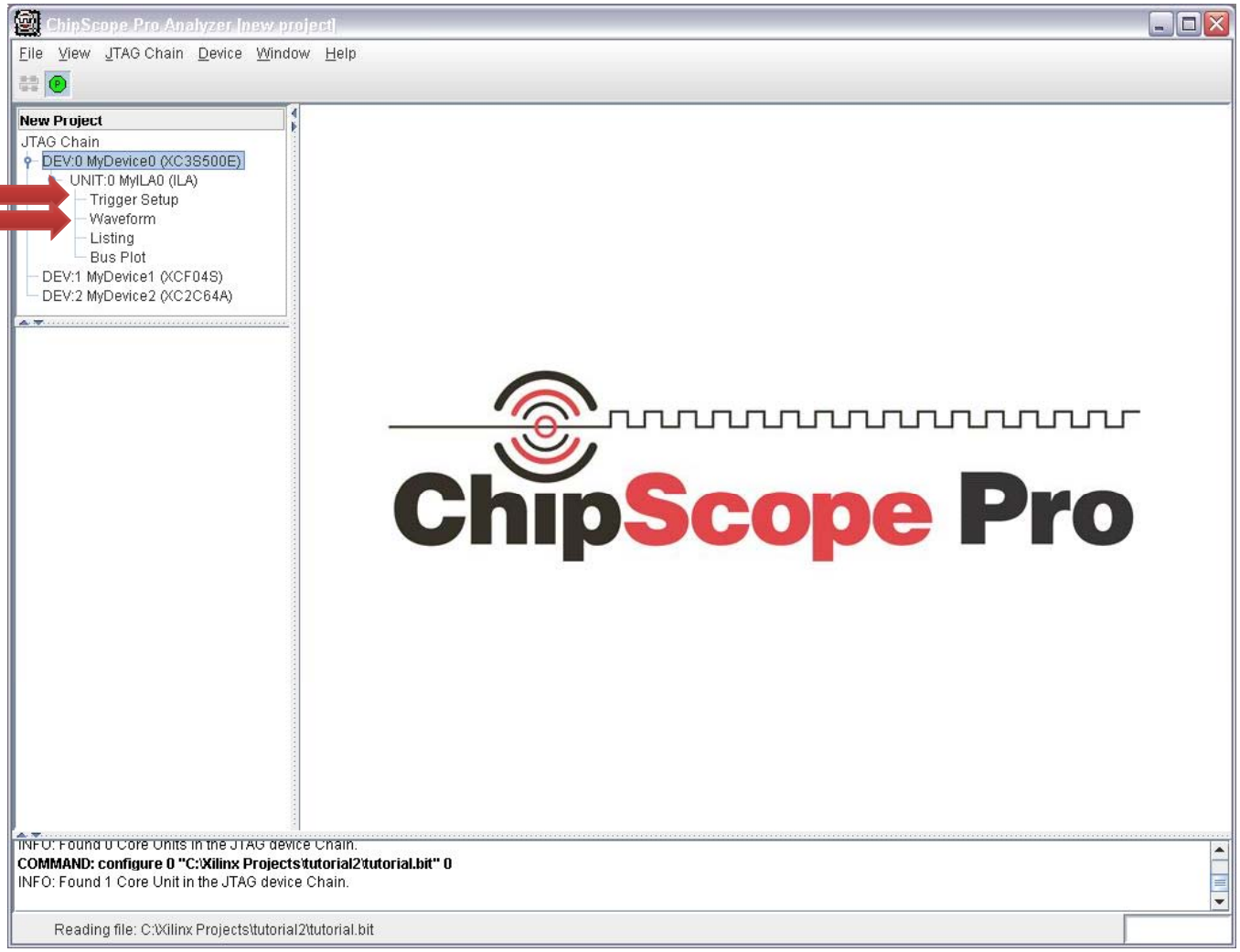



Identified devices will be added to the list. To upload the configuration bitstream to the Spartan 3E, right click on the ***DEV:0 MyDevice0 (XC3S500E)*** and select ***Configure.***

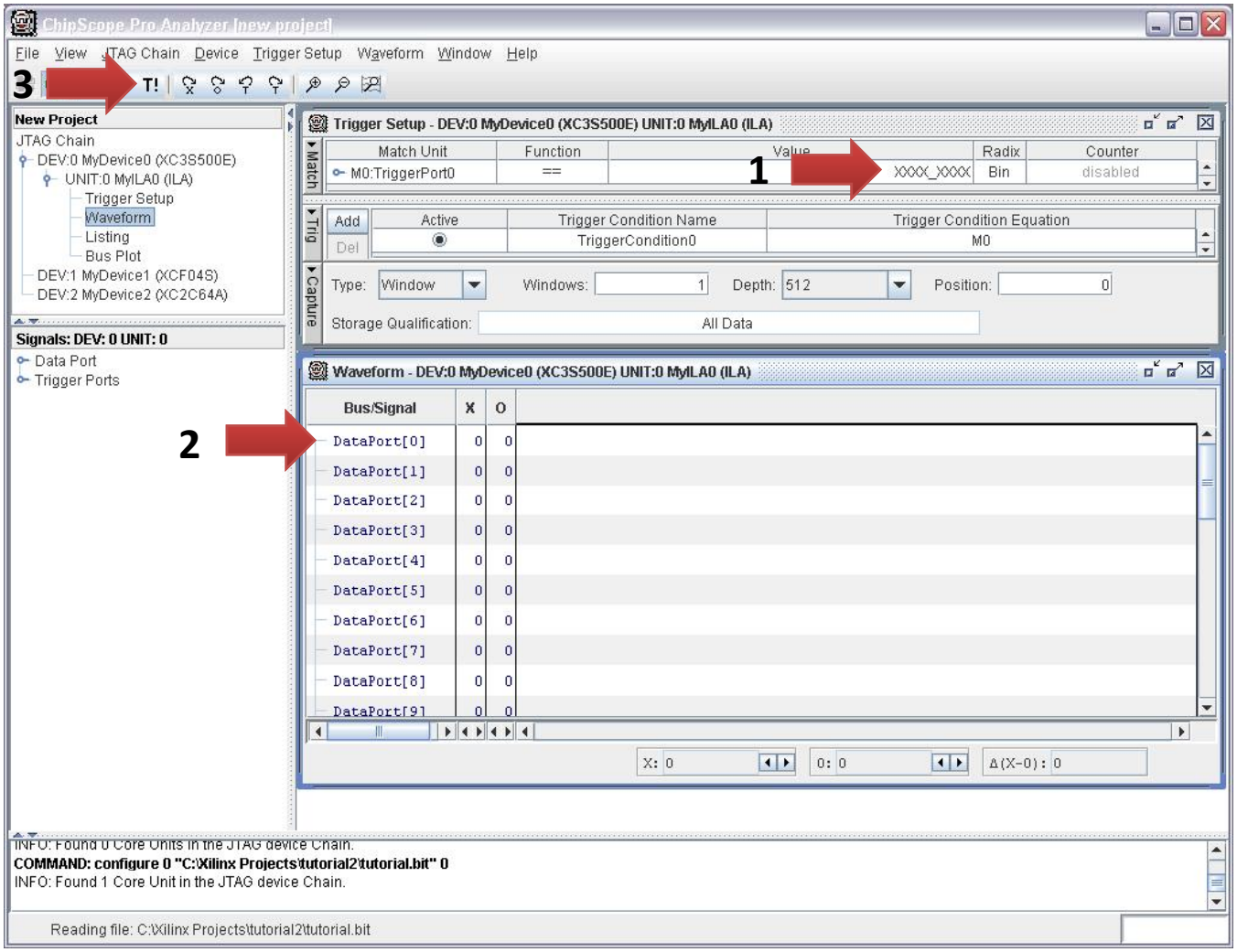



Select configuration file from your project directory with extention ***.bit*** and press ***Open***

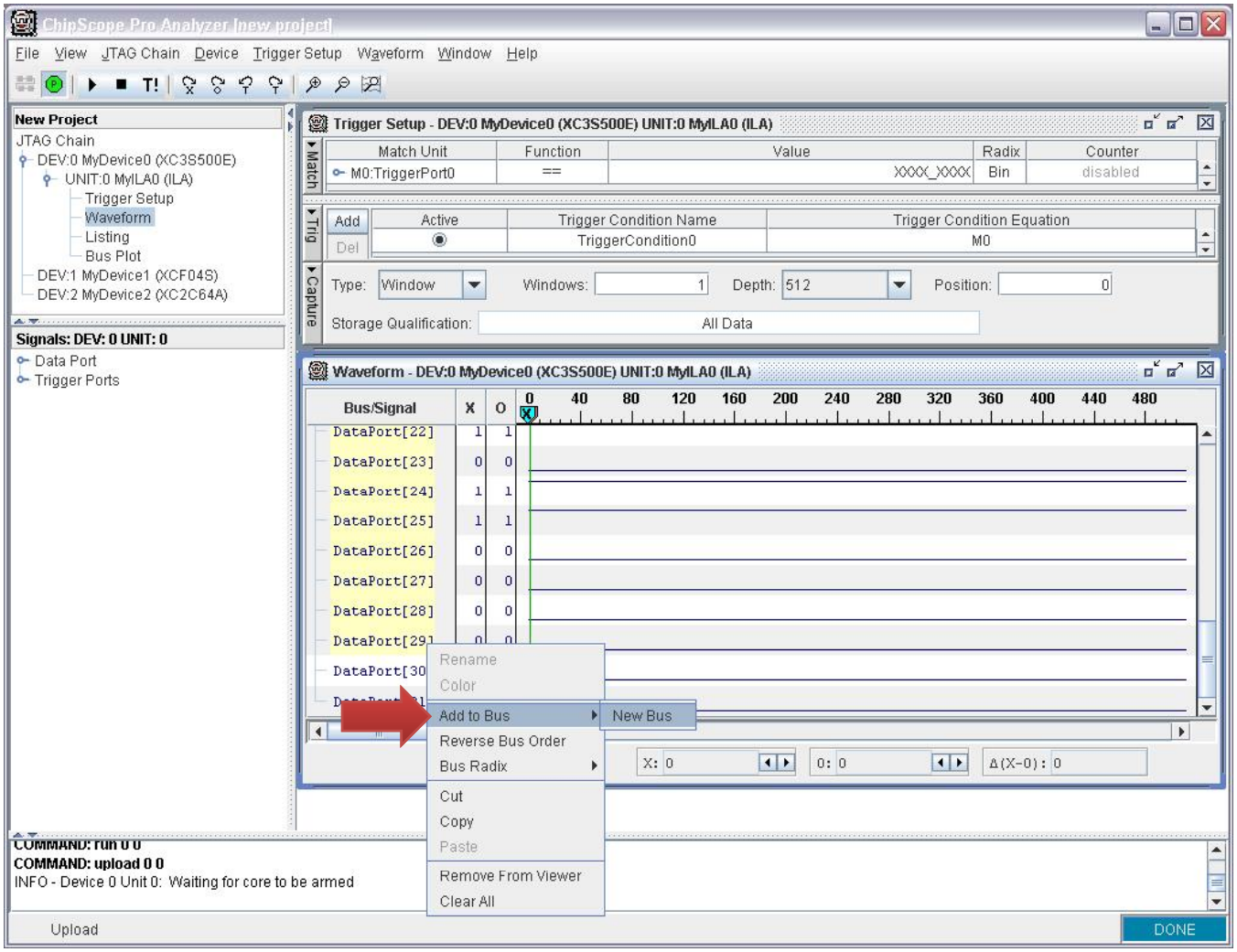
Press ***OK*** to upload the configuration bitstream file.



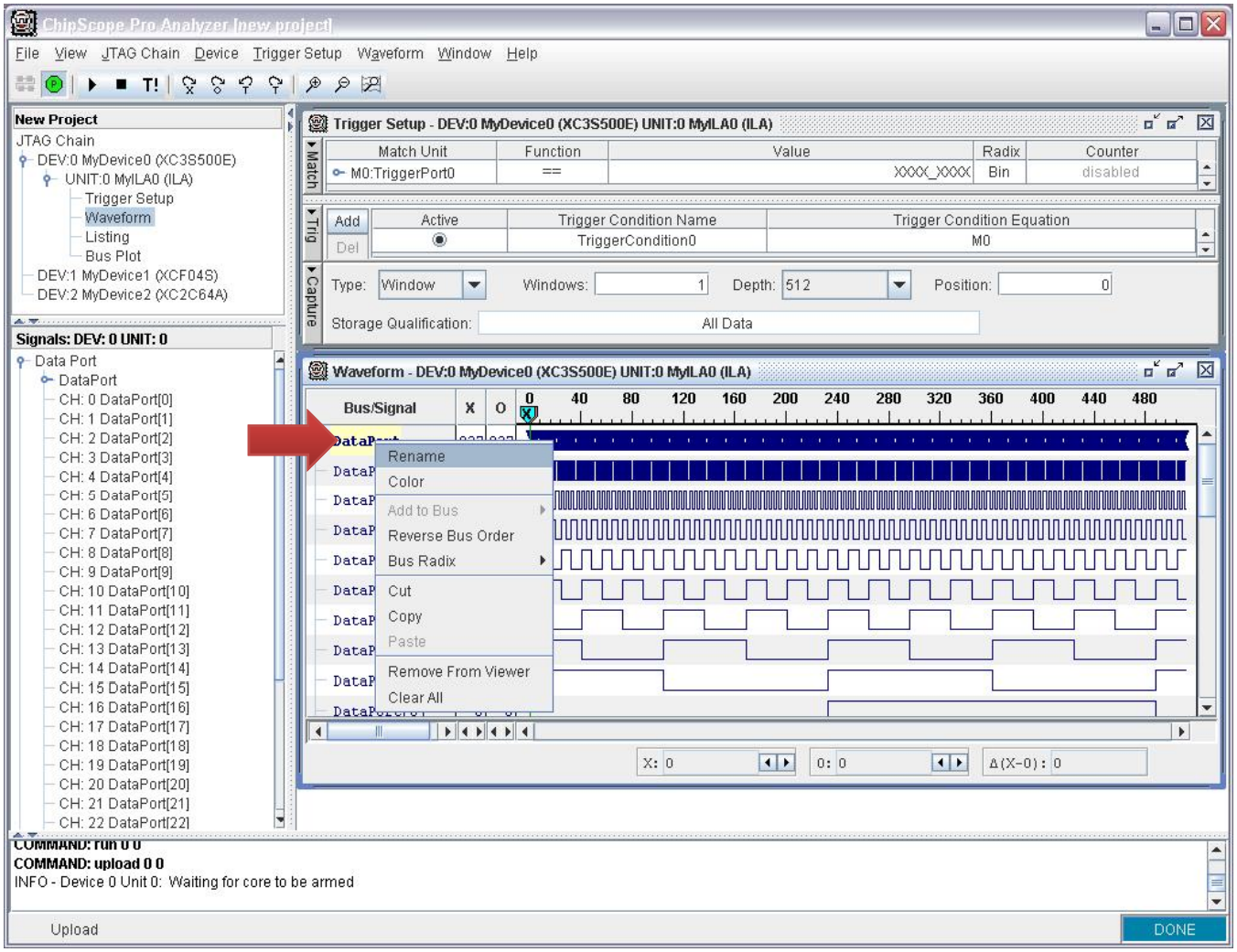

When configuration is completed successfully ***DEV:0 MyDevice0 (XC3S500E)*** will expand and show ***MyILA0*** component with the ***Trigger Setup***, ***Waveform***, etc… Double click on ***Trigger Setup*** and ***Waveform*** to bring the ILA interface on screen.



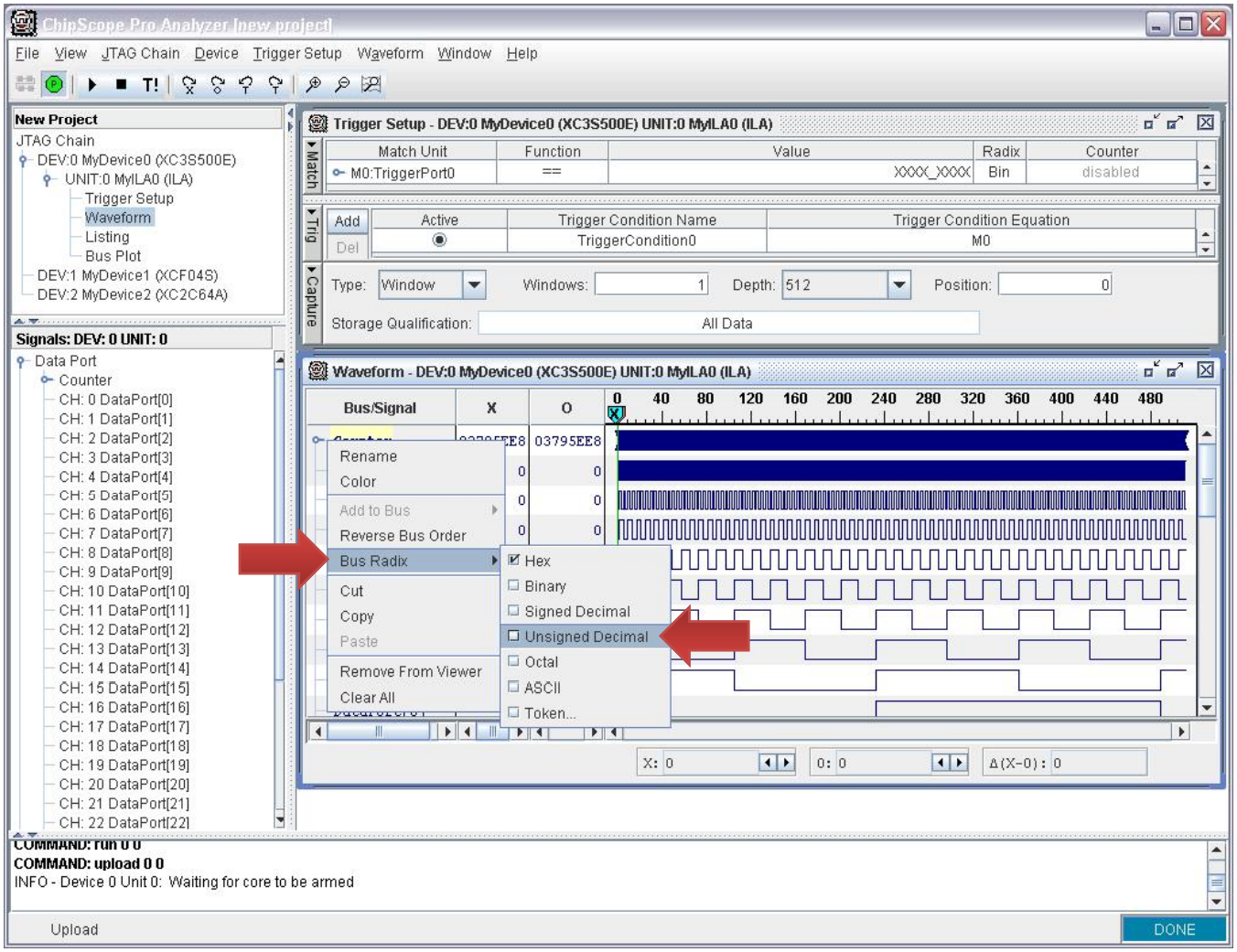

1. Trigger Setup window allows to configure ChipScope Pro to commence capturing when combination of the particular signals is encountered.
2. Middle window shows the signal captures as it would on any logic analyzer.
3. To initiate capture at any point, without trigger press on ***T!*** button.



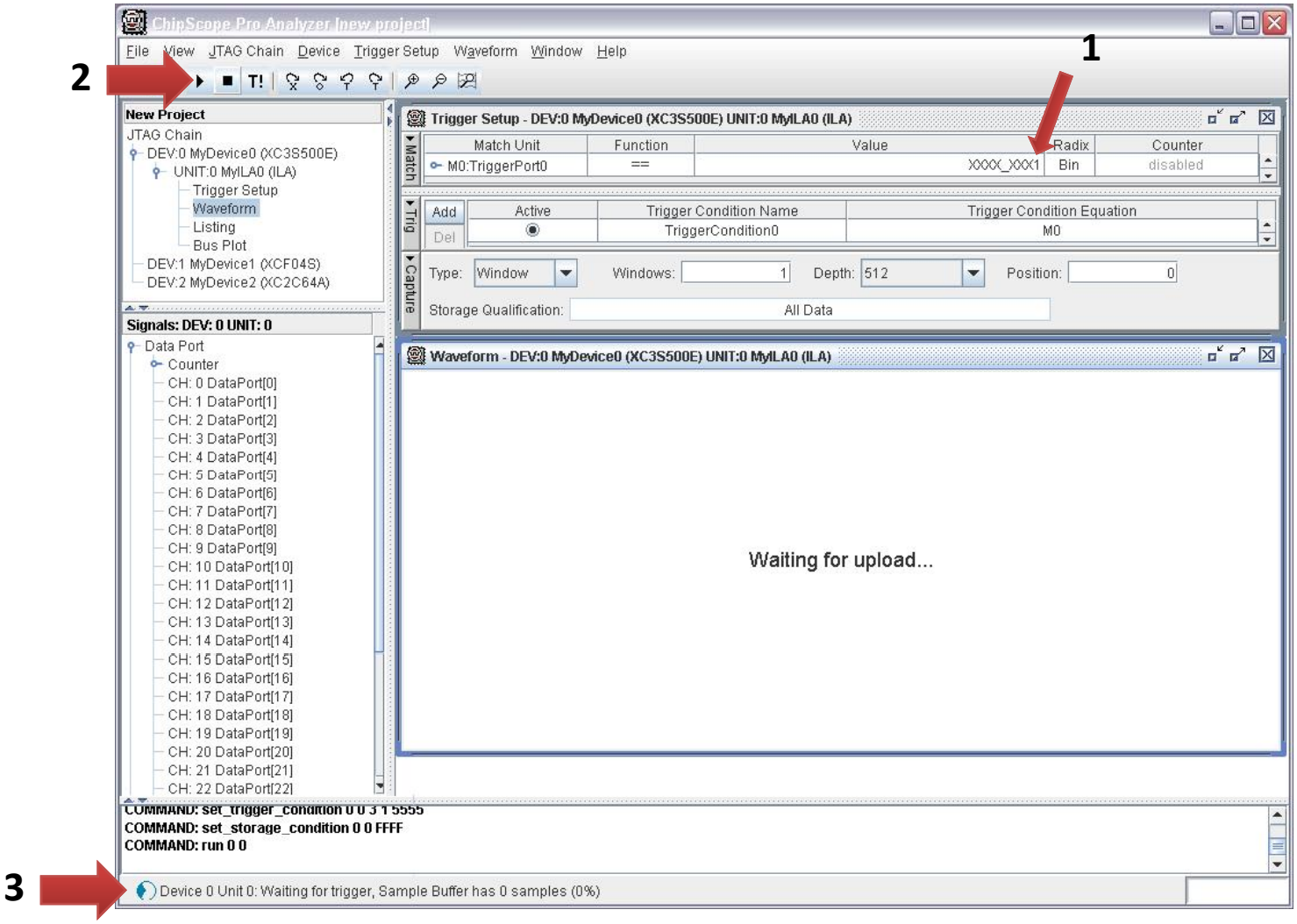

After the immediate capture you will see all of the levels of individually captured signals. If some signals represented a Bus you can group them into the Bus for easier readout. Select signals that you want to place in a Bus by holding ***Ctrl*** and clicking on desired signals, after which right click over the selected signals and select ***Add to Bus > New Bus***



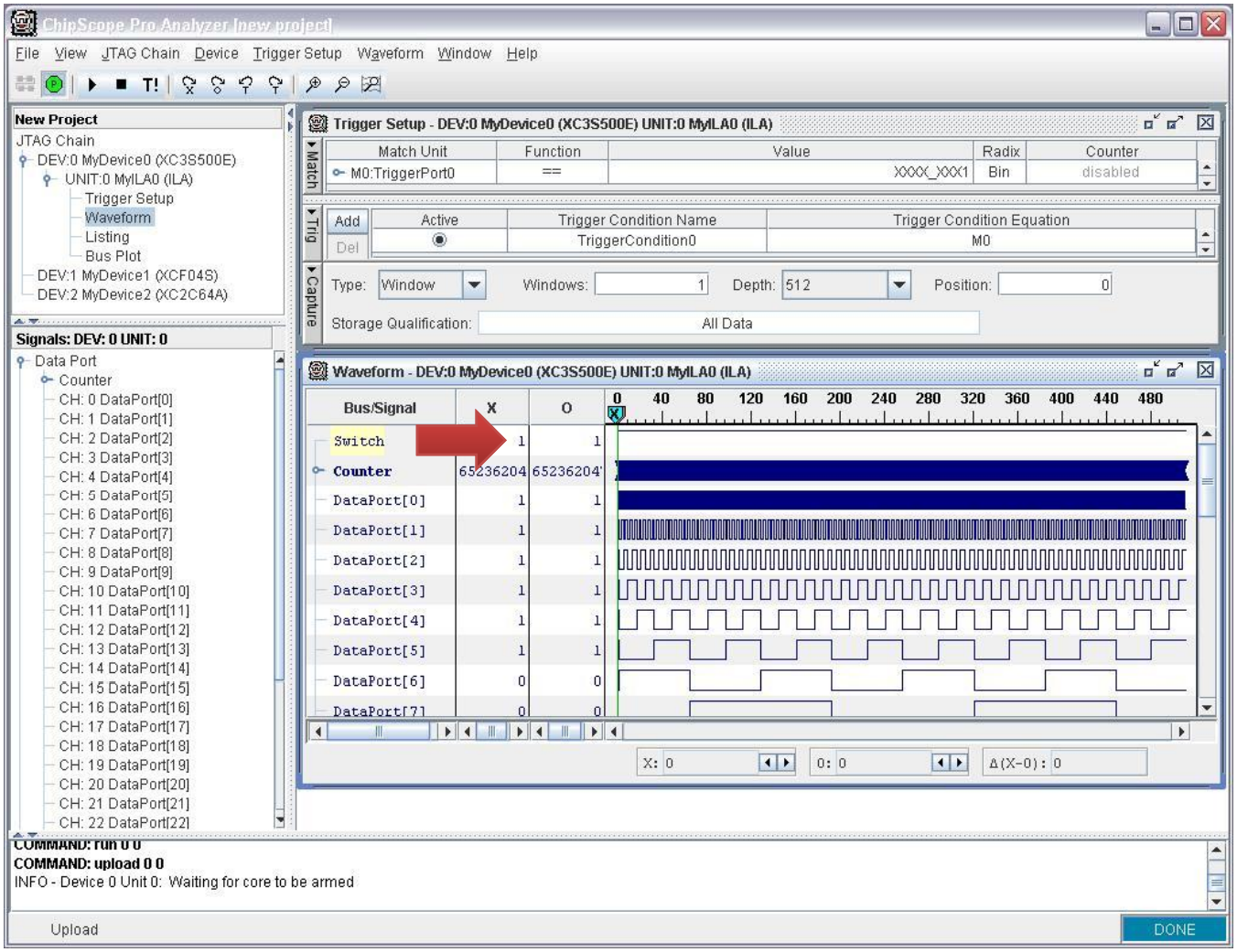

Grouped Bus will be added at the top of the list of signals. Grouped Bus can be renamed by right clicking on the Bus and selecting ***Rename.***



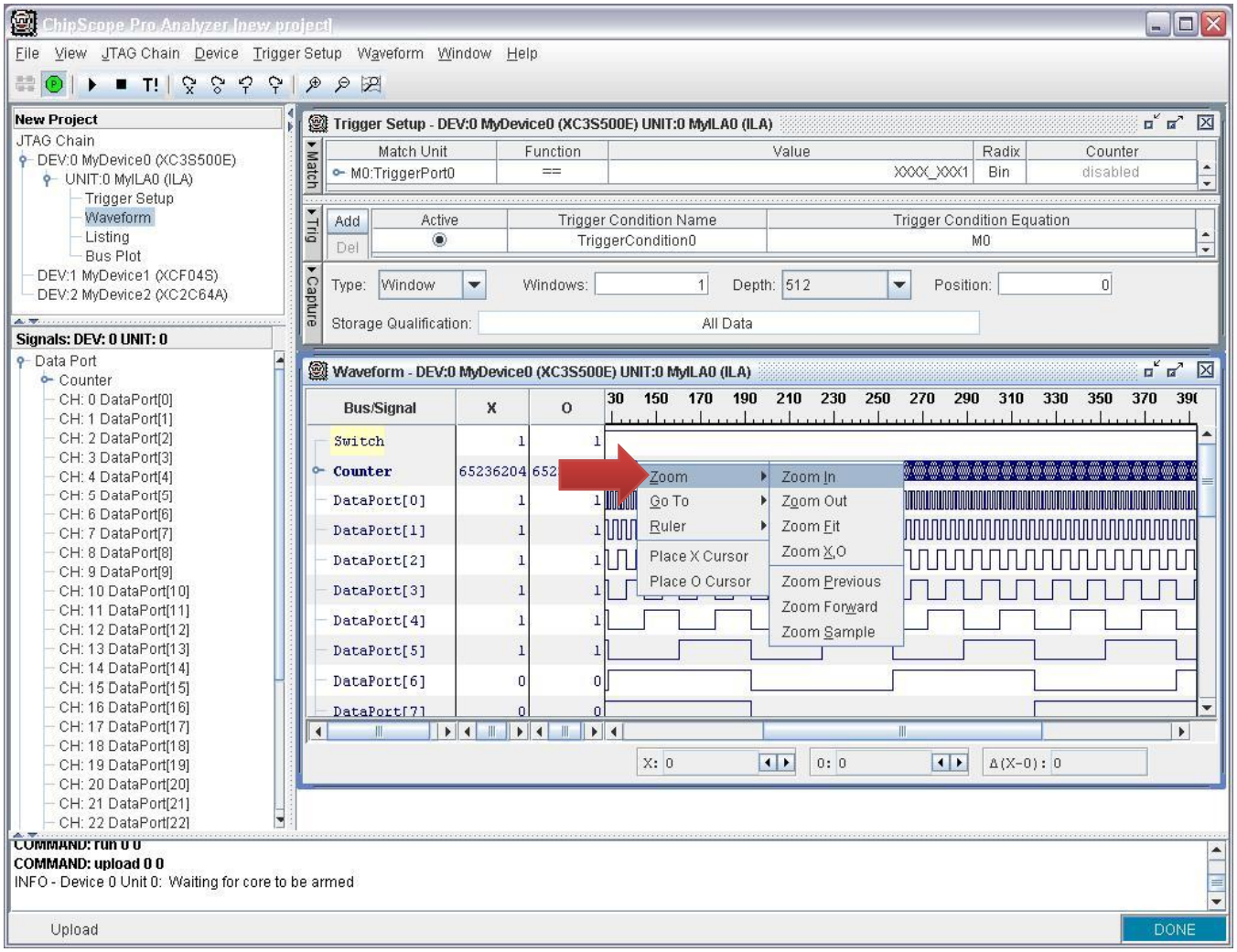

There are several ways of representing grouped value. Options are available by right clicking on the bus and selecting ***Bus Radix*** menu. In this menu you have many options such as ***Hex***, ***Dec***, ***Binary***, and more.



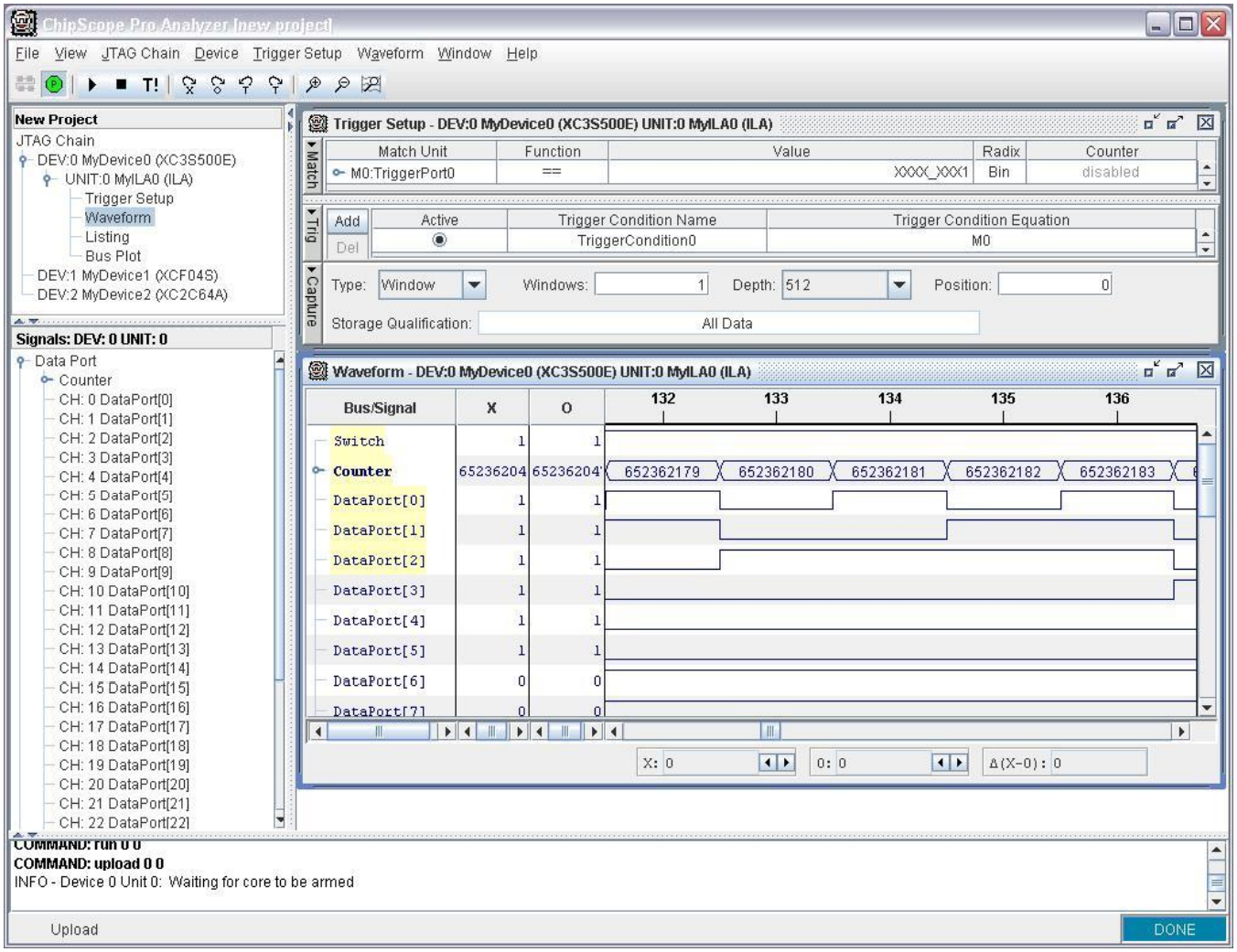

1. To capture signals based on trigger input, set the first bit of the trigger vector to ‘1’.
2. Click on the Play button at the top left corner to start monitoring for the trigger.
3. When switch is set to the top position signals would be captured based on the transition.




As shown, when trigger (switch 0) was set to ‘1’ signals were captured and shown in the Waveform window.




To see in detail the bus output you can right click on the bus and select Zoom>Zoom In. Resulting zoom is shown on the next page, where transition from one value to the other is displayed  along with sample number.




This concludes the introduction into the ChipScope Pro with the use of a single ILA module.

# Conclusion

This complete ChipScope Pro tutorial which included:

- Overview of the ChipScope Pro
- Generation of ChipScope Pro cores
- Insertion of ChipScope Pro cores into the project from Tutorial 1
- Identification and Configuration of the FPGA device directly from ChipScope Pro
- Capturing signals using ILA module and displaying in the Waveform window.
- Capturing signals based on the trigger setup.

*Next tutorial will cover the Virtual I/O module which allows for generation of virtual inputs and outputs inside the FPGA using the ChipScope Pro GUI. This benefits the designer on the stage of debugging of the firmware and hardware .*