LAUTERBACH
DEVELOPMENT TOOLS

NIOS II Instantiating the Off-chip Trace Logic

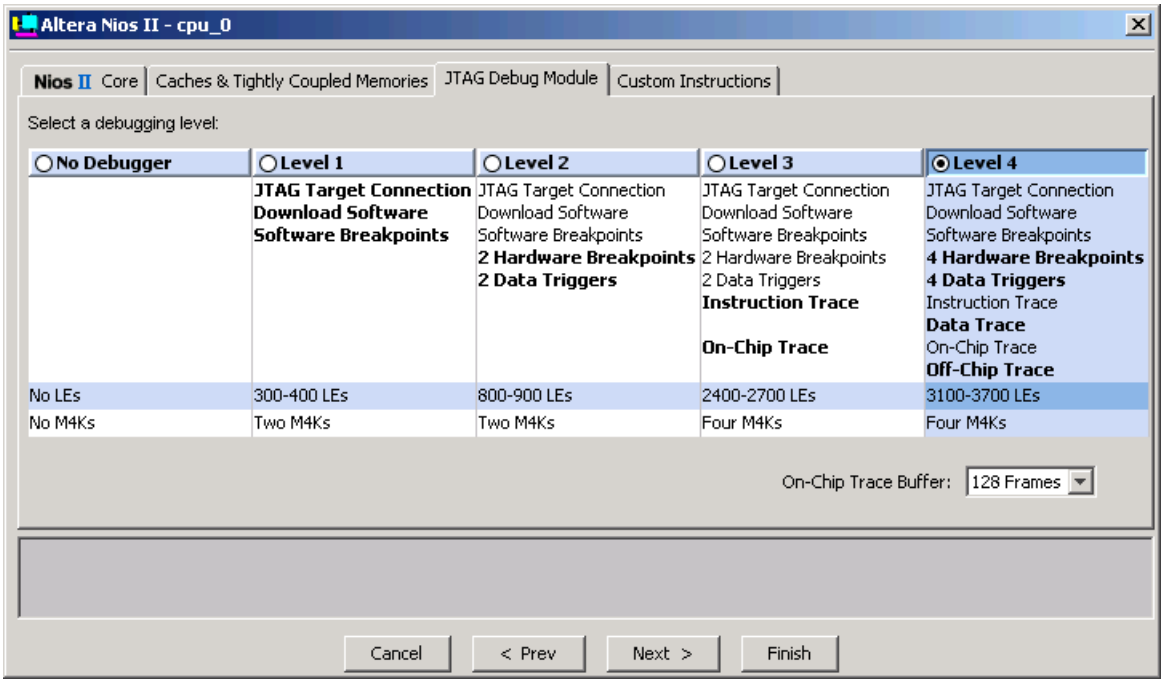
MANUAL

NIOS II Instantiating the Off-chip Trace Logic

NIOS II Instantiating the Off-chip Trace Logic

Version 06-Jun-2024

First of all, you have to tell the SOPC Builder in the Quartus II design software from Altera, that you want to enable Off-Chip tracing for your core. This is done by selecting 'Level 4' in the 'JTAG Debug Module' tab of the CPU settings window in the SOPC Builder.



After the SOPC Builder regenerated your system, you will have some additional off chip trace related output pins on your system, which are called:

jtag_debug_offchip_trace_clk	Trace clock for the trace data.
jtag_debug_offchip_trace_data[17..0]	18 Bit trace data.
jtag_debug_trigout	Trigger for off chip trace.

Disable Automatic PLL Instantiation

Without further modifications a compilation of the system will automatically implement an additional PLL, which is used to clock the internal trace buffer logic of the system. This PLL is needed, because the internal trace buffer logic needs a clock, which has to have twice the frequency of the cpu clock. Especially for Cyclone devices this automatic PLL instantiation is not optimal, because Cyclone devices only have two PLLs available.

Another issue is that you don't get access to the trace buffer clock if the PLL is instantiated automatically. Having access to this clock enables the implementation of further improvements.

To use the resources more efficiently and to get access to the clock for the internal trace buffer logic do the following steps:

1. Close the SOPC Builder.
2. Open the ".ptf" file, in which the SOPC Builder stores all settings for your system
3. In this file, look for a line which looks like
   ```
   oci_embedded_pll = "1";
   ```
   and change this line to
   ```
   oci_embedded_pll = "0";
   ```
   If you can't find a line like the above (this happens with older Quartus II versions), look for a line which looks like
   ```
   oci_offchip_trace = "1"
   ```
   and **insert** a new line under the above line, with the content
   ```
   oci_embedded_pll = "0";
   ```
4. Open the SOPC Builder and Re-Generate the system.

In recent Quartus II versions (7.1 and later) you will have to modify the ".sopc" file instead of the ".ptf" file.In this file you have to:

Look for a line which reads:
```
<parameter valueString="true" name ="debug_embeddedPLL"/>
```

and change this line to:
```
<parameter valueString="false" name ="debug_embeddedPLL"/>
```

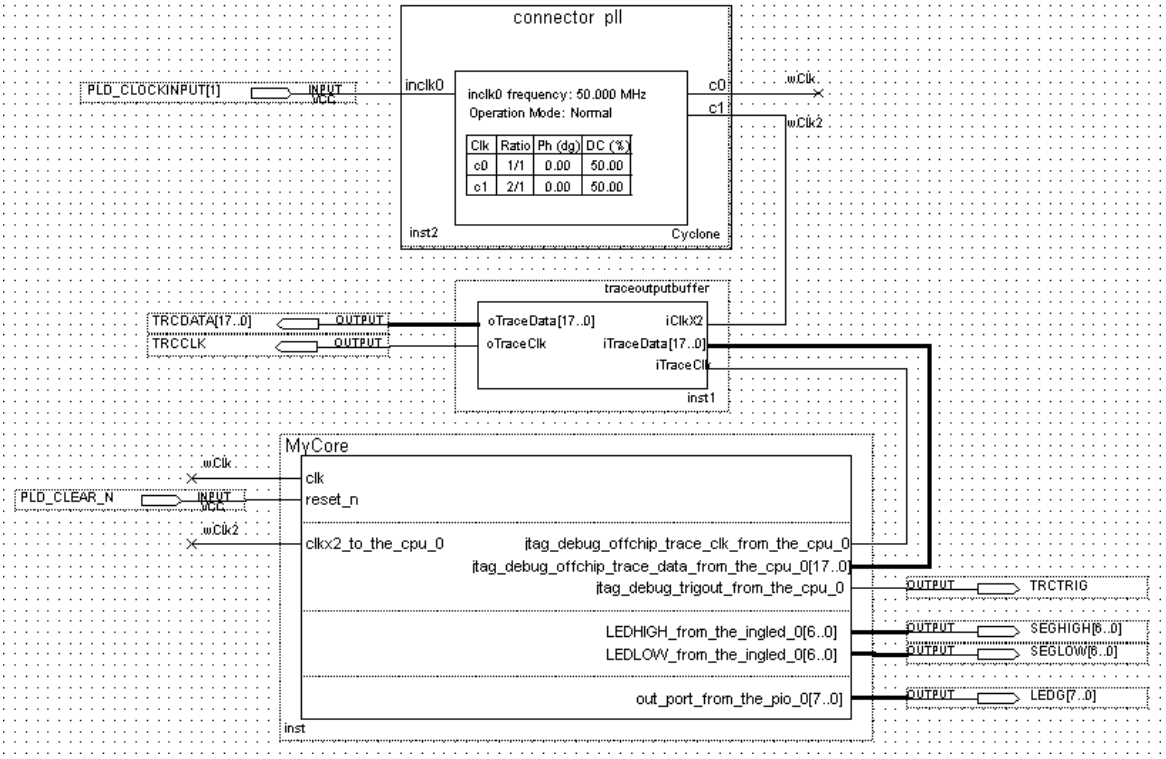
After this modification, you should let the SOPC builder regenerate your system. Then you should have an additional input which is called
```
clkx2_to_the_...
```

This extra input is used to clock the system internal trace buffer logic. You have to connect a clock to it, which has twice the frequency of the cpu clock. The simplest way to achieve that is to use the PLL which generates the cpu clock and add an additional output to this PLL, which has twice the frequency of the CPU clock. In this way you only need an additional PLL output to use the off chip trace and not a whole additional PLL.

Using Extra Registers for Better Timing

Another modification is needed to guarantee a unified timing on the trace pins and to improve the timing relations of the trace clock and the trace data. To achieve this we recommend to insert a buffer stage between the trace outputs of your system and the output pins. LAUTERBACH provides such a buffer stage on the DVD in the directory ~~/**demo/nios/trace**. When this buffer stage is used it is ensured that the trace pins use IO registers, which guarantees a uniform timing; additionally the trace clock is center aligned to the trace data eyes, which improves the reliability of the trace data a lot. Additionally the extra buffer stage improves the maximum frequency of the design, because the extra layer of registers simplifies fitting.

Here is an example of an extremely compact design schematic which uses an Off-Chip trace port.



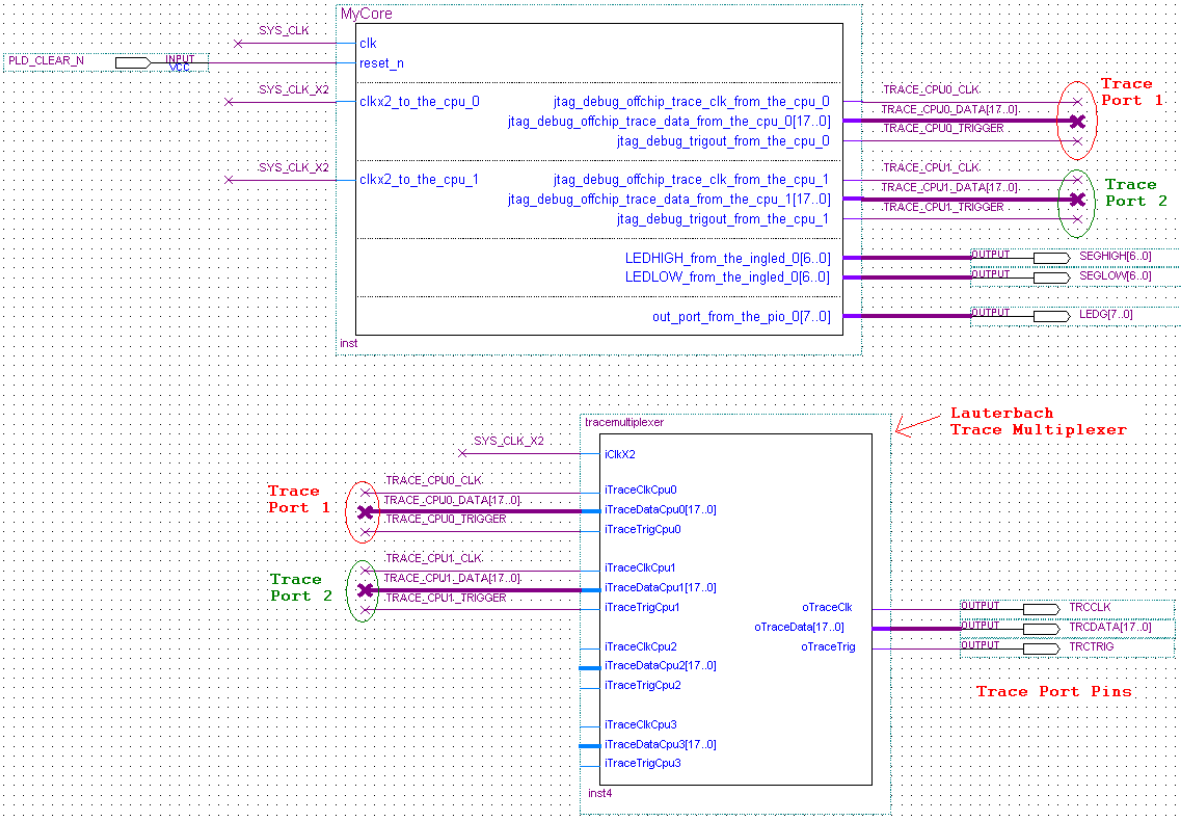
Multiplexing Trace Outputs of Multiple CPUs

Recently multi-core designs become more and more popular. If you implement a Design which contains multiple Nios II CPUs and several of the Nios II CPUs implement an off-chip Trace Port, then LAUTERBACH offers a solution to multiplex these outputs to only one set of Trace Pins. With this approach it is possible to decide which CPU will be selected to output its trace via the off chip trace pins.

To use this multiplexer approach, several constraints have to be met:

One CPU clock	All CPUs, which you want to connect to the off-chip trace pins have to run with the same CPU clock.
No internal Trace PLL	You have to follow the above recommendation about “Disabling automatic PLL instantiation”. So you need access to the trace clock, which runs with twice the frequency of the CPU clock.

Here is an excerpt of an extremely compact design schematic, which shows how to include the Trace Multiplexer:



The Trace Multiplexer module can also be found in the **demo/nios/trace** on the CD.

If your design contains a Trace Multiplexer module, then in the PowerView Software you can switch the TraceMultiplexer to the CPU you are currently debugging with the command SYStem.Option.TracePortEna On.
This will automatically happen, when you only debug one CPU at the same time.

Since only one CPU can be switched onto the Trace Port, the command will be ignored, if you already assigned the trace port to another CPU. In this case you first have to disable the use of the TracePort in all CPUs, before you can enable the TracePort for a different CPU.