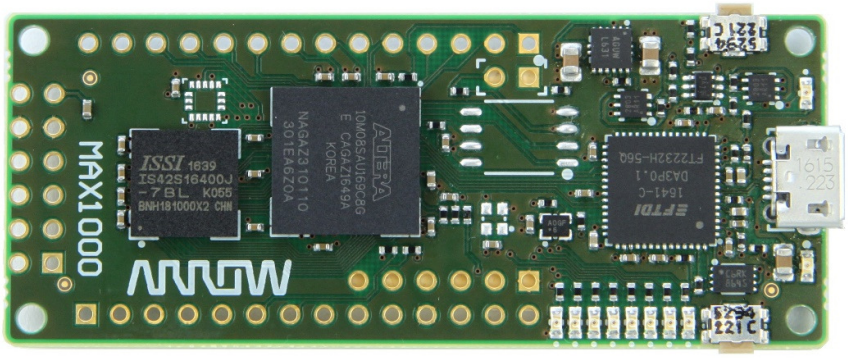


MAX 1000 Tutorial
Designing a Nios II System

Exercise Manual

for MAX1000 Board

Quartus Prime 18.1 / Platform Designer

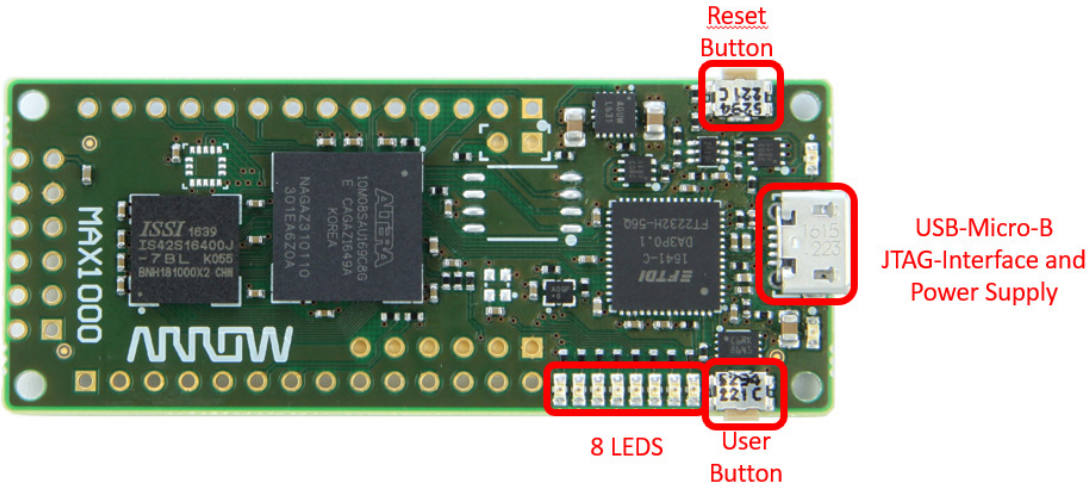
Version 2.0/ 2019-03-04

HW-Setup of the Nios II System

In this lab you will generate a small uController system containing a Nios II CPU, RAM, some peripherals and the internal User-FLASH block as program storage. The program code will be copied from the User-FLASH into the internal memory at start-up time.

Hardware setup requirements:

- Micro-USB cable connected between computer and MAX1000 board.
- Power supply is not needed because of the supply over USB.



Names (files, directories) and actions which needed to be done as in the instructions below are highlighted in red and bold!!!!

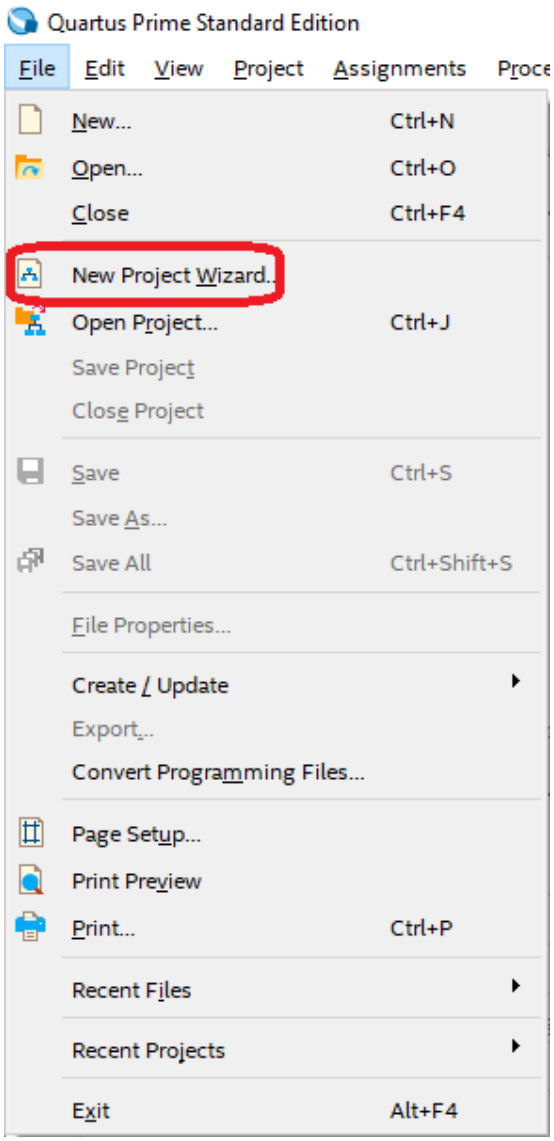
Copyright © 2019 www.arrow.com

1. From the provided USB Stick copy the Max1000_Nios_tutorial directory to your computer. For the training we recommend copying it to your desktop.

   Important: The directory path must not contain spaces
   If the path to your Desktop contains spaces please use a different location

2. Start the Quartus Prime software.

3. Form the File Menu select “New Project Wizard”

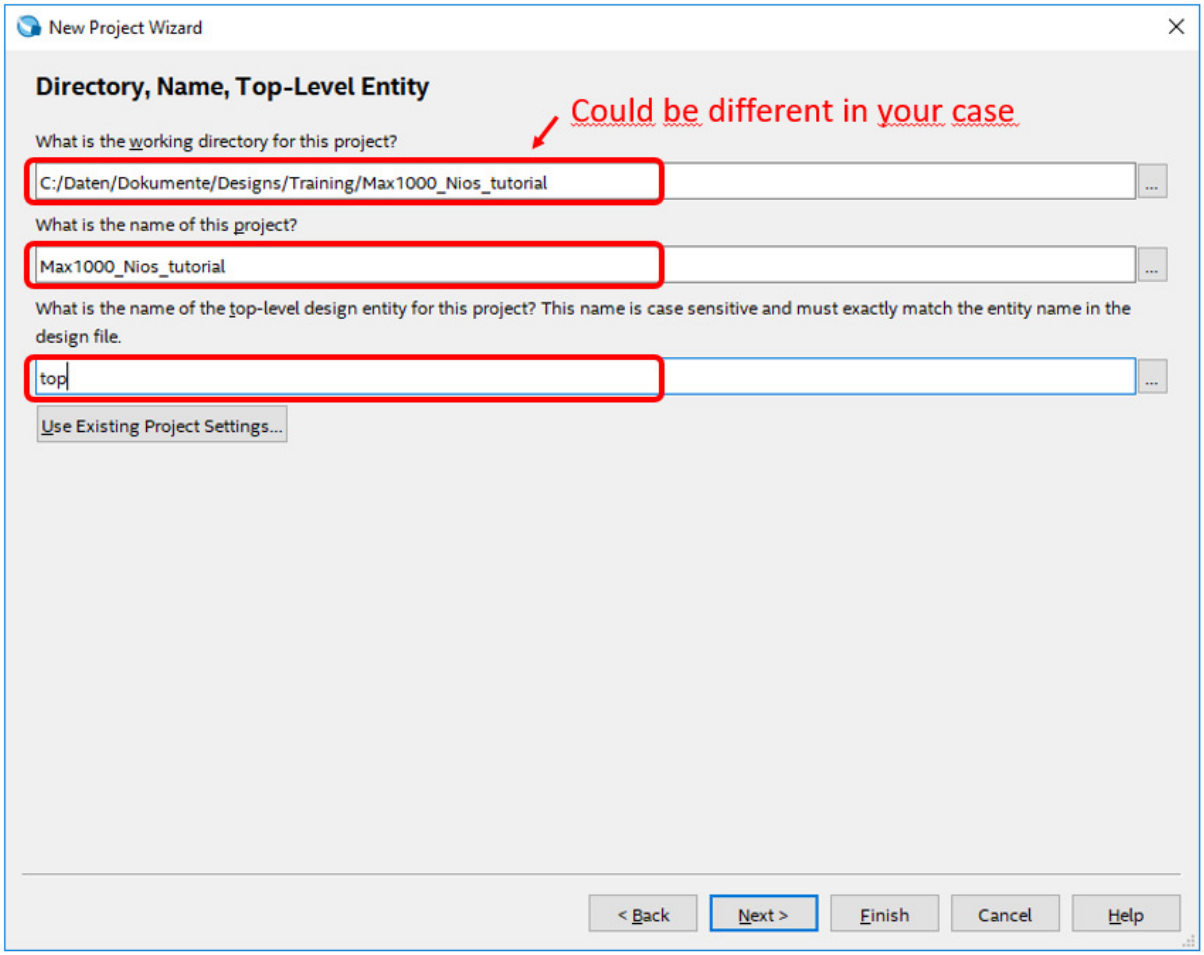


4. Click “Next” to get to the “Directory, Name, Top-Level Entry”

5. Select as Working Directory the in Step 1. copied directory

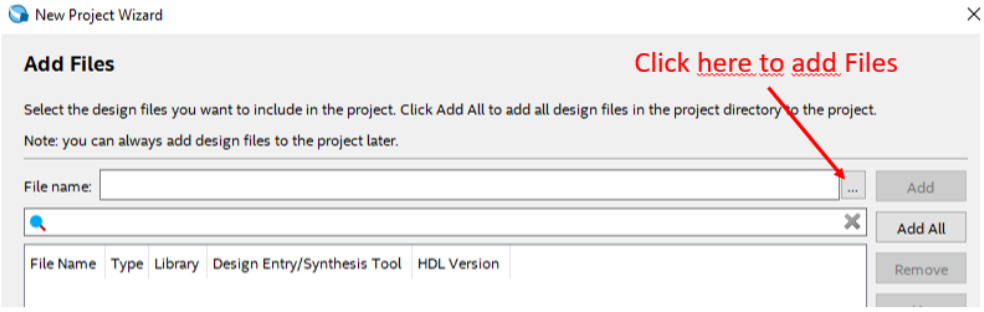
6. Name the project (second line) as: **top**

 Copyright © 2019 www.arrow.com



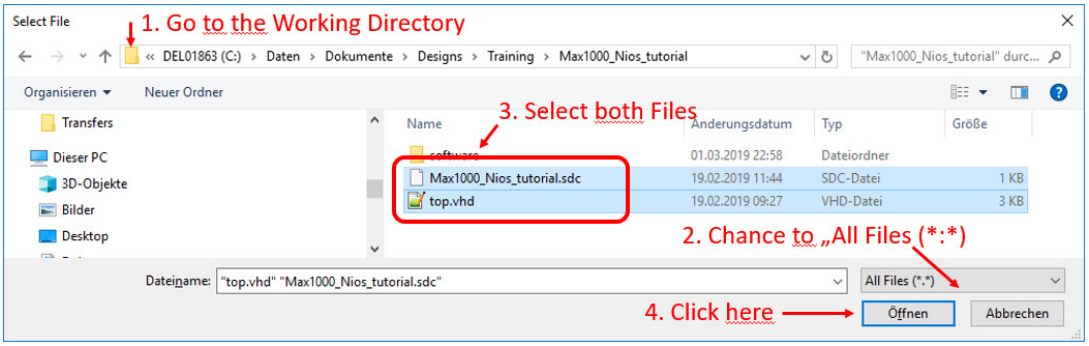
7. Click “Next”
8. Leave Project Type as “Empty project” and click “Next”
9. We need to add two already existing files

   a.

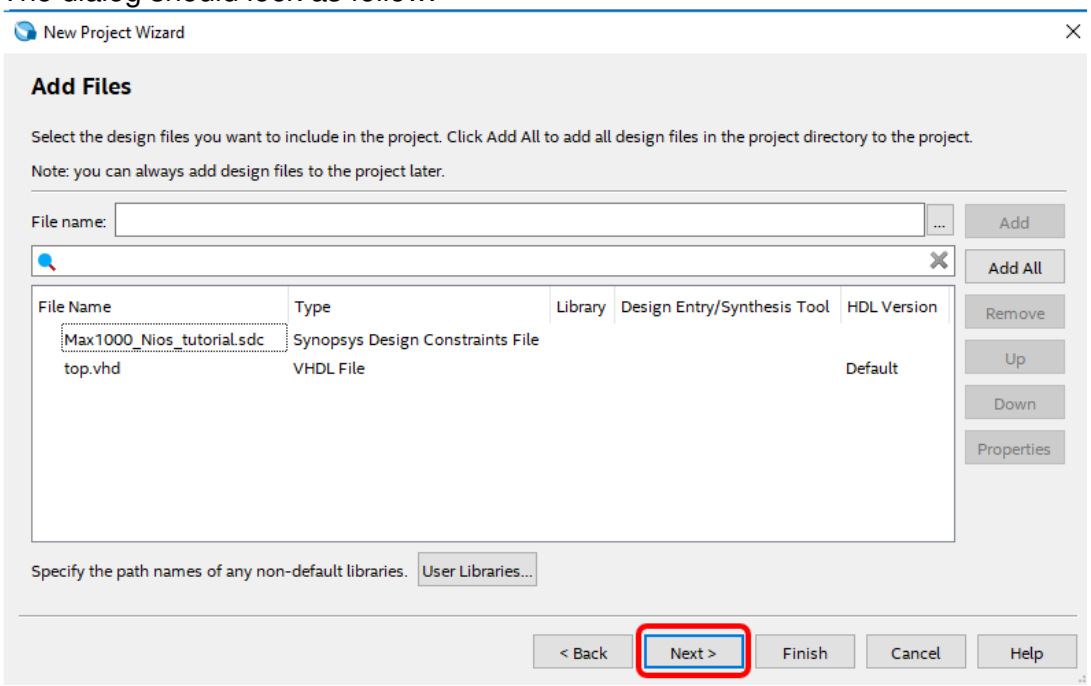


 Copyright © 2019 www.arrow.com

b.



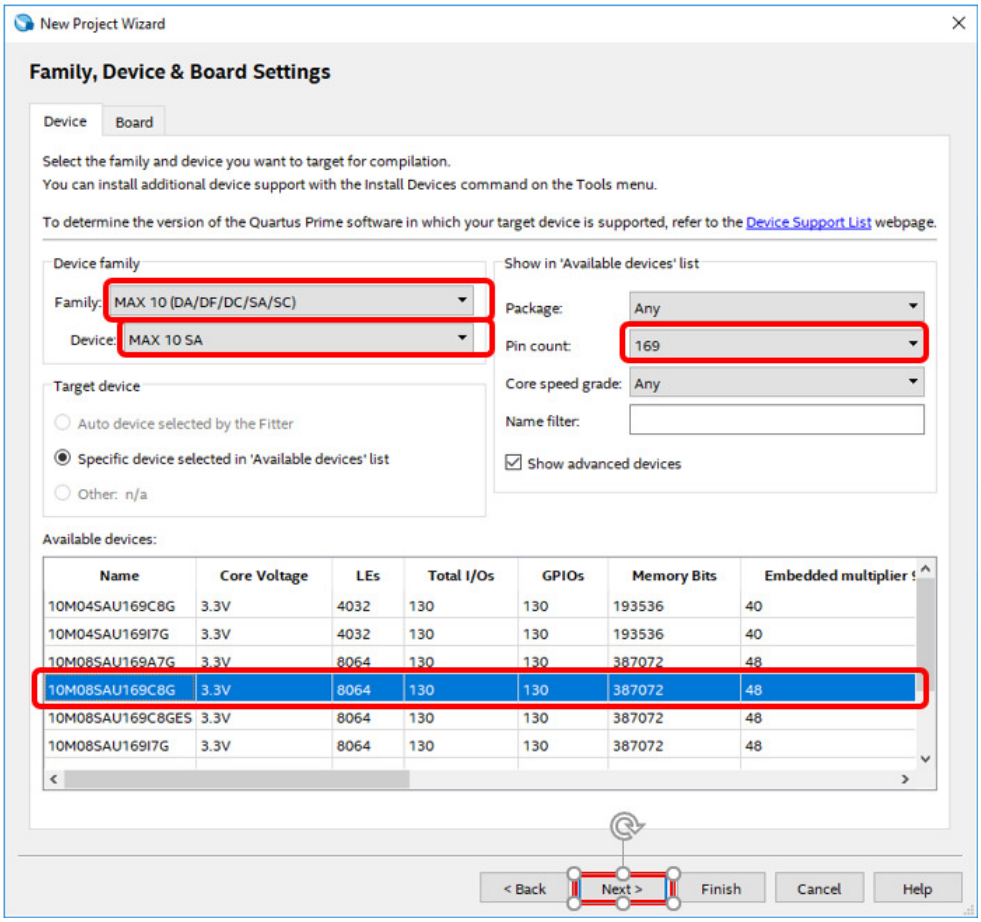
c. The dialog should look as follow:



d. Click “Next”

10. Now we select the MAX10 device
10M08SAU169C8G

Copyright © 2019 www.arrow.com



Click “Next”

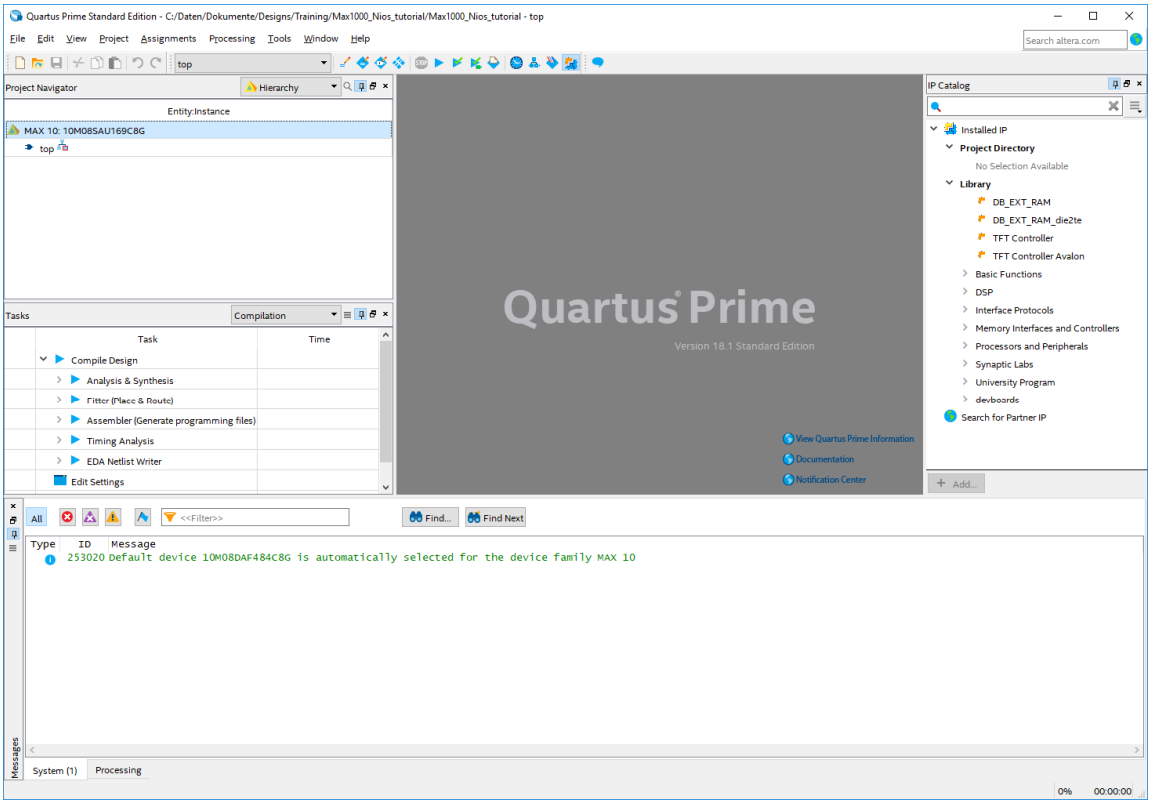
11. At “EDA Tool Settings” leave the defaults and click “Next”

12. In the “Summary” you can check if the device has been set correctly. If not go back to change.

    Click “Finish”

 Copyright © 2019 www.arrow.com

The Quartus Prime window should look like this:



13. We need to change the configuration mode of the MAX10 device and make some general IO assignments

    a. Double click on the MAX10 device in the “Project Navigator” window.

    b. Click on “Device and Pin Options”

    c. Select Category “General”

    d. **Deselect** the option “Enable nCONFIG, nSTATUS, and CONF_DONE pins”

    e. Select Category “Configuration”

    f. Change “Configuration Mode” to “**Single Compressed Image (1376Kbits UFM)**”

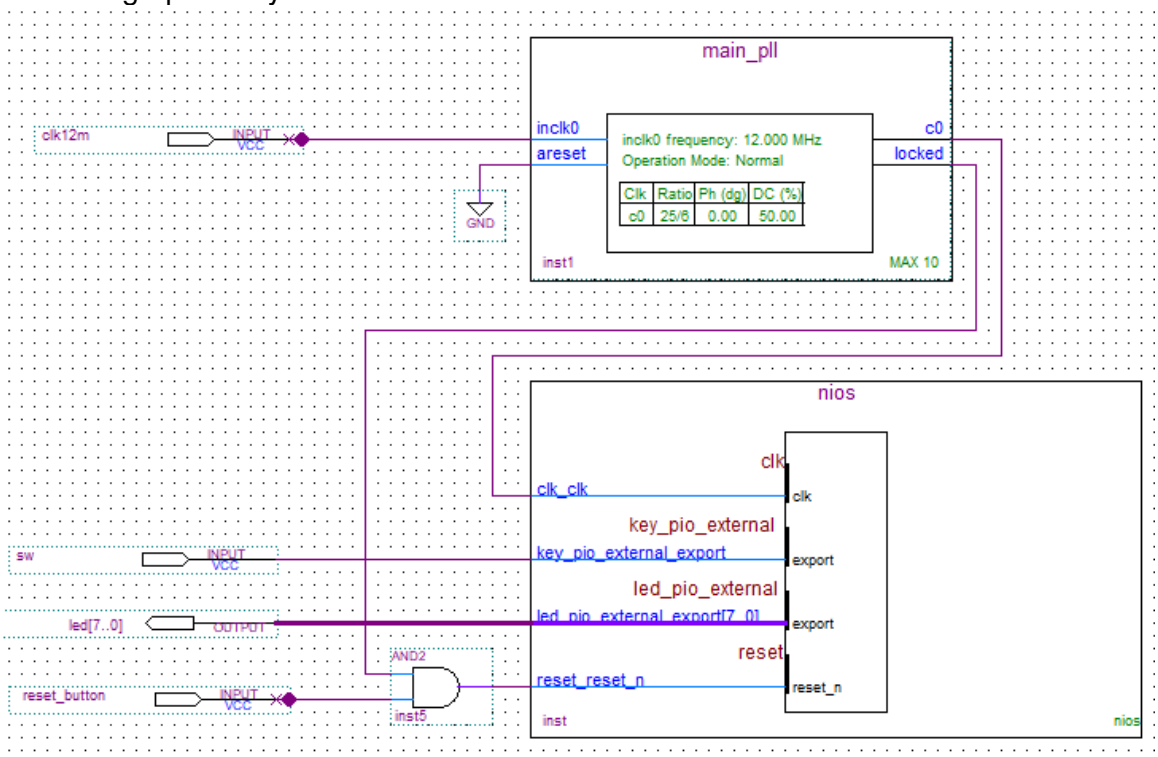
    g. Click “OK”

    h. Click “OK”

14. By double click on the “top” level instance in the “Project Navigator” window at the top left

 Copyright © 2019 www.arrow.com

The already provided top level VHDL file will be opened. You can have a look but should not change anything.
You will find two components used in this design:
main_pll
nios
This is a graphical system overview:



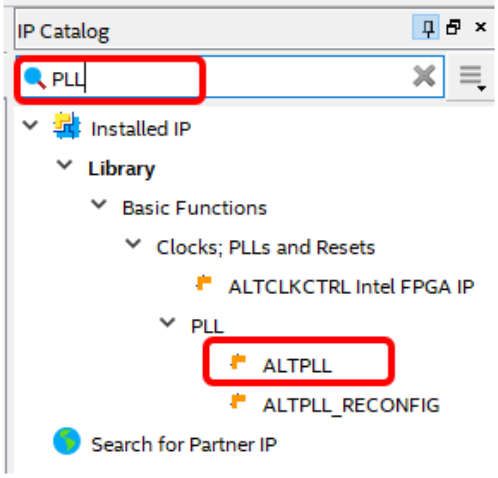
Both components will be generated in the next steps

15. First, we build a PLL to generate 50MHz for the NIOSII block

    a. At the right side of the Quartus Prime window you find the “IP Catalog”.

 Copyright © 2019 www.arrow.com

b. Type “PLL” in the search field and double click “ALTPLL”

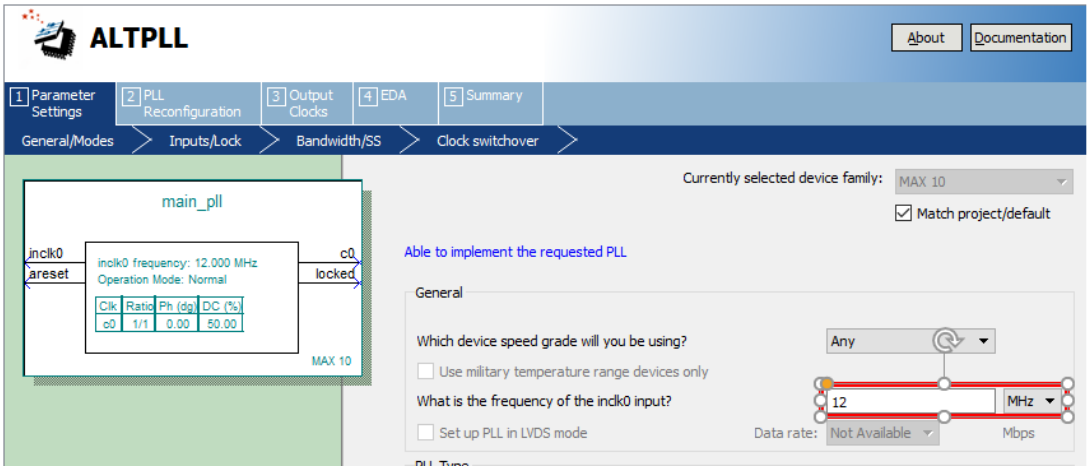


c. Name the IP: **main_pll** and select “**VHDL**” as IP variation file type.

d. Click “OK”

e. The MegaWizard Plug-In Manager is starting to guide you

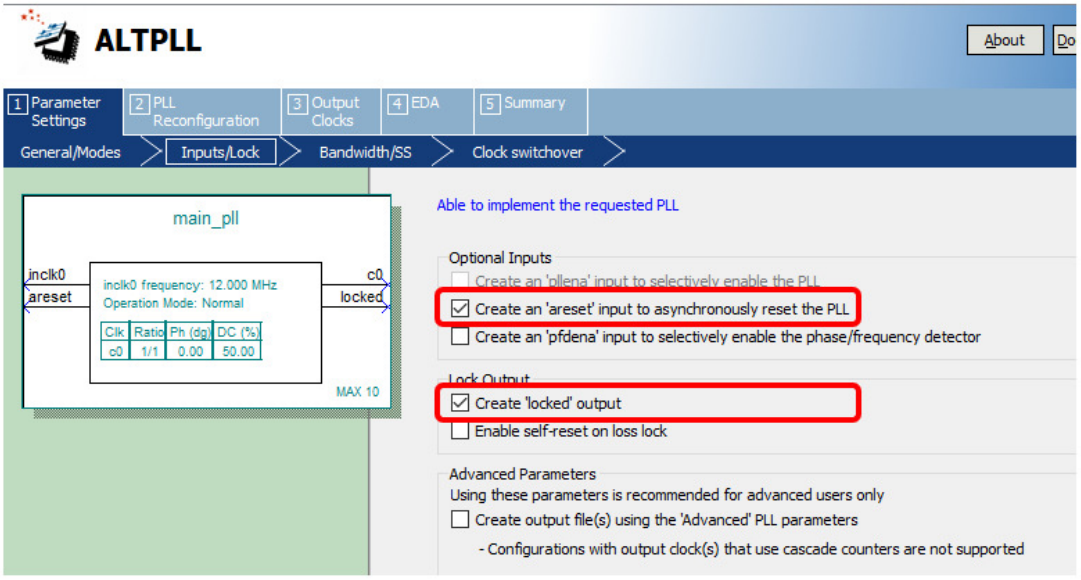
f. Set the input frequency to **12 MHz**



Click “Next”

 Copyright © 2019 www.arrow.com

g. Check these options and click “Next”

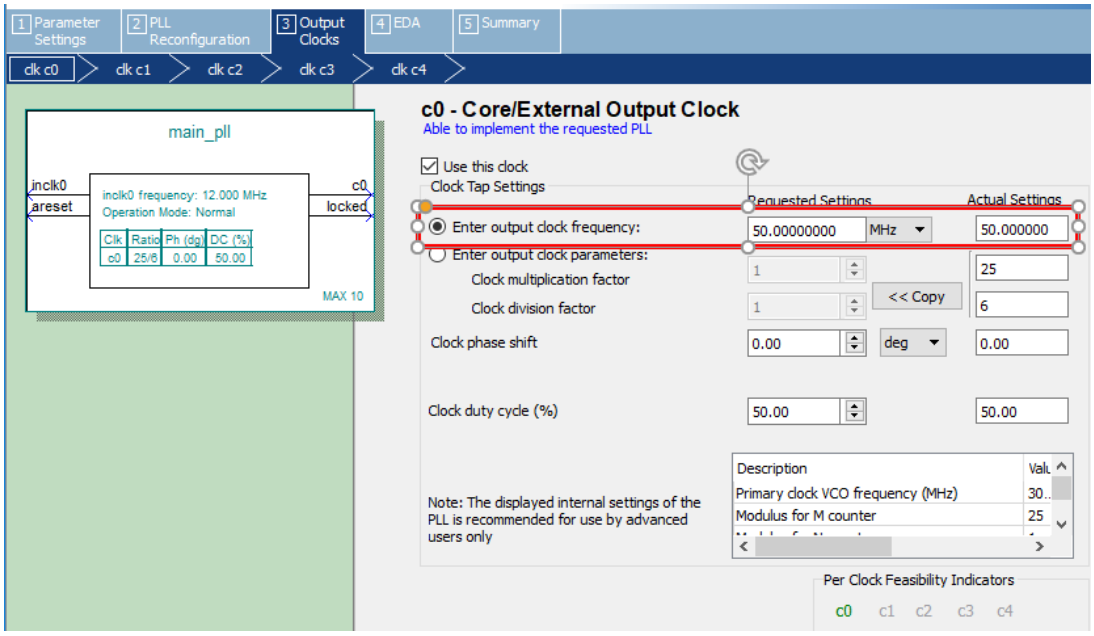


h. Click “Next”

i. Click “Next”

j. Click “Next”

k. Here we set the output frequency of “clk c0”
Set it to 50MHz



l. As we only need one output frequency click “Finish”

m. In the Summary you can select the output files to be generated.
main_pll.cmp is the component declaration
main_pll_inst is the component instantiation

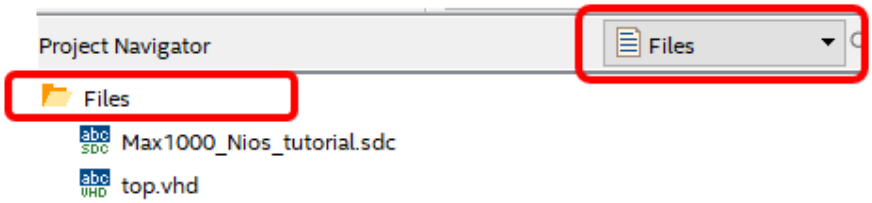
 Copyright © 2019 www.arrow.com

Click “Finish”

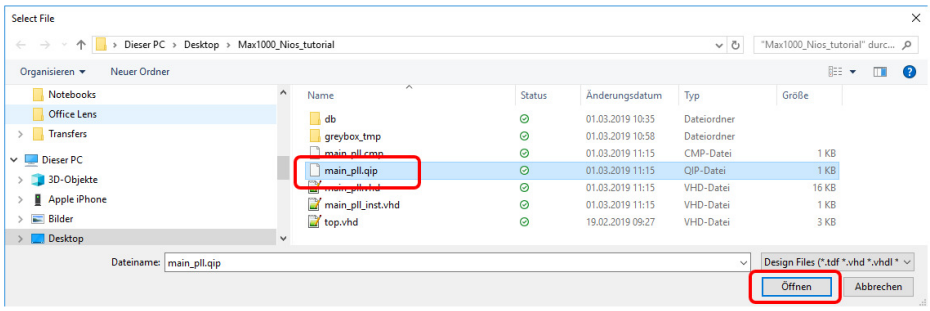
n. The generated files can be found in the working directory.

16. If the PLL IP is not added automatically, please add it manual to our project

a. Change the “Project Navigator” view to “Files and double click the first line “Files”



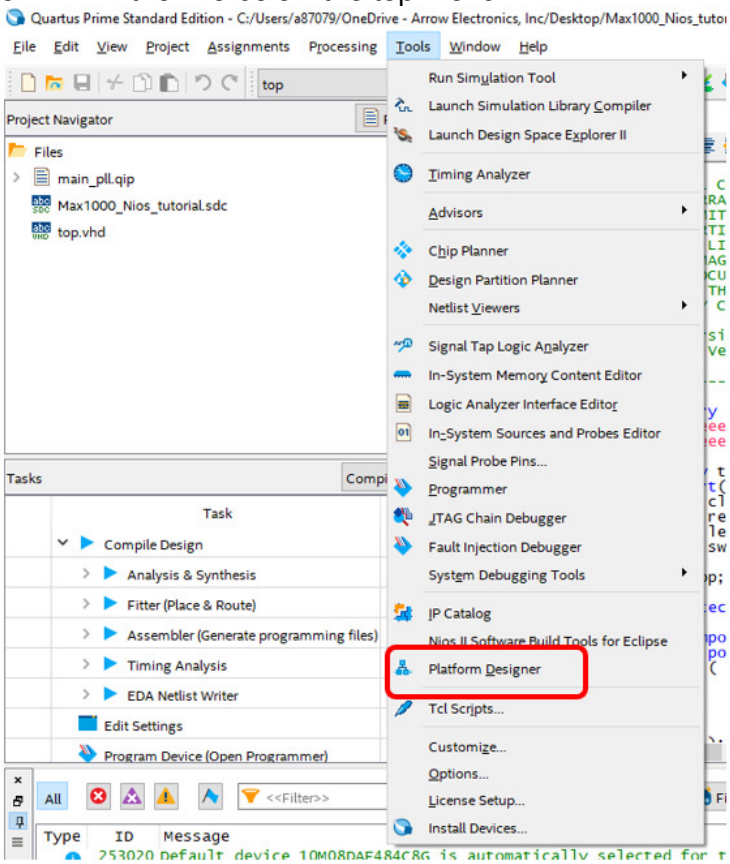
b. As in Step 9 add **main_pll.qip** to the project



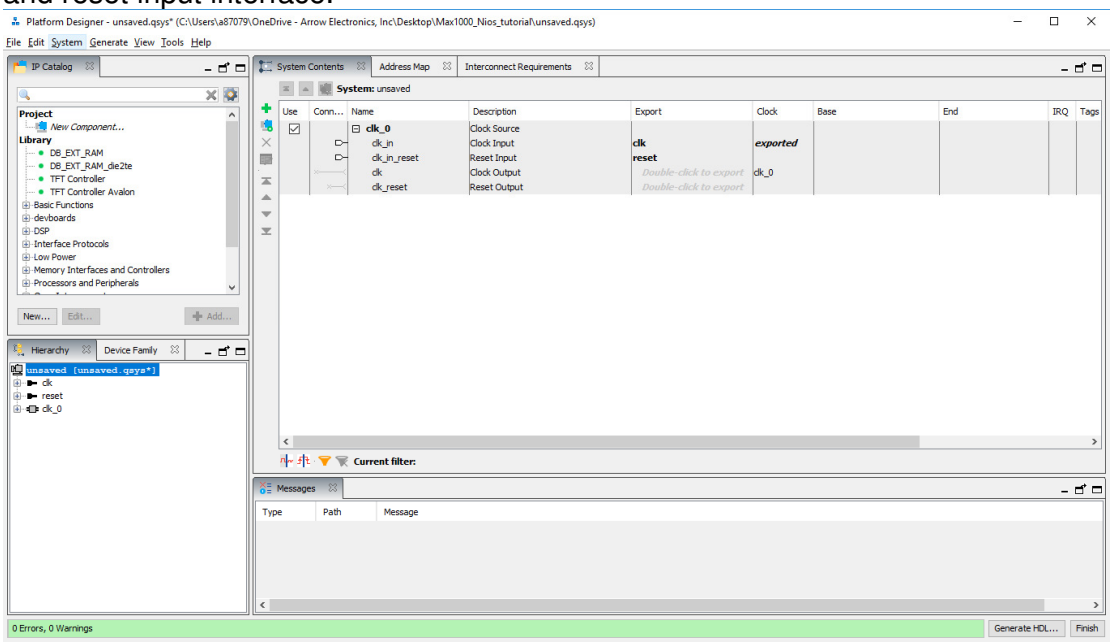
17. Now we are building the NIOSII system

 Copyright © 2019 www.arrow.com

a. Start Platform Designer from “Tools => Platform Designer” or by clicking the icon in the line below the top menu.

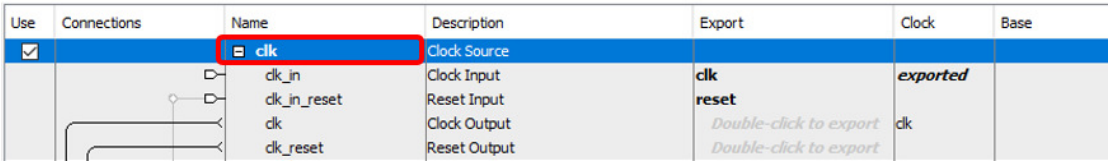


b. Platform Designer is starting with a blank project which contains only the clock and reset input interface.



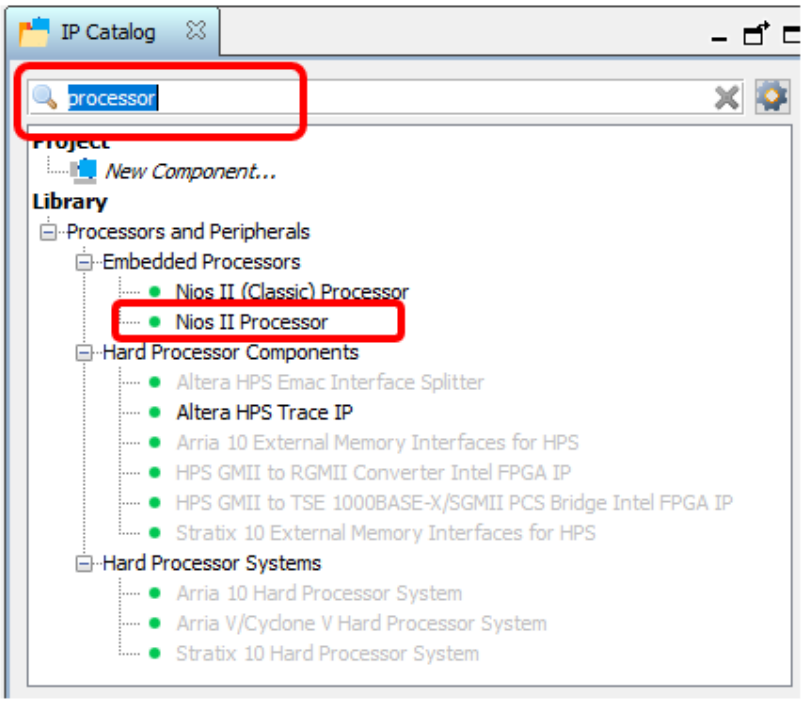
 Copyright © 2019 www.arrow.com

c. In the column "Name" right click "clk_0" and rename it to "**clk**"



18. Adding the NIOS II Processor

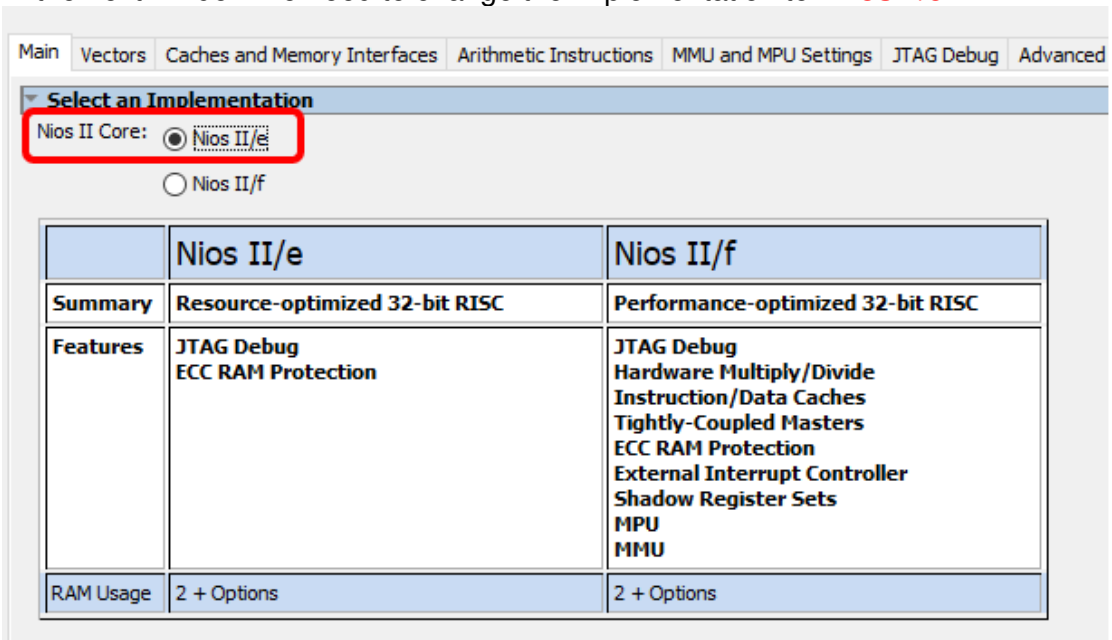
a. To add the NIOSII core type in the IP Catalog search path: "processor" and double click "**Nios II Processor**"



Make sure not to select the old classic version!!!!

 Copyright © 2019 www.arrow.com

b. In the next window we need to change the implementation to "**Nios II/e**"

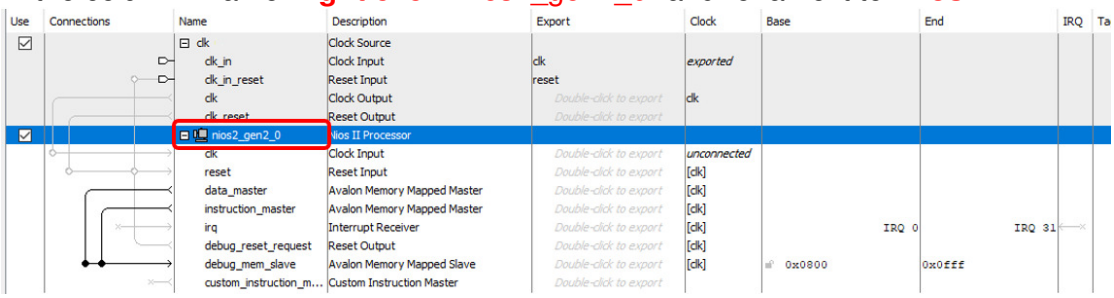


c. Click "Finish"

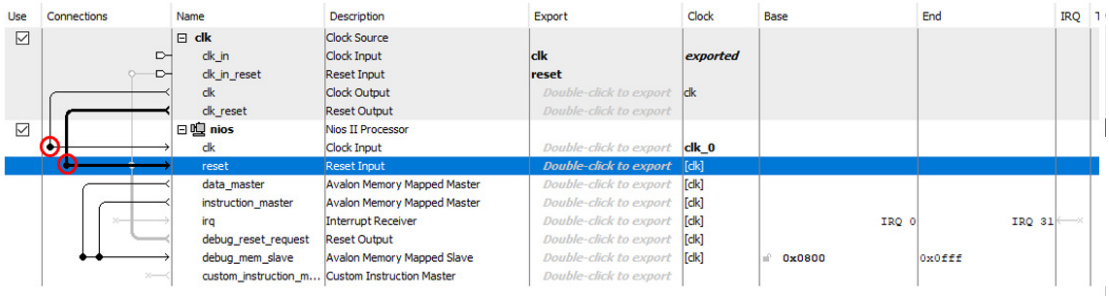
d. The Nios II Processor is added to the system

e. **Currently you can ignore all error messages. They will be fixed later**

f. In the column "Name" **right click** "nios2_gen2_0" and rename it to "**nios**"



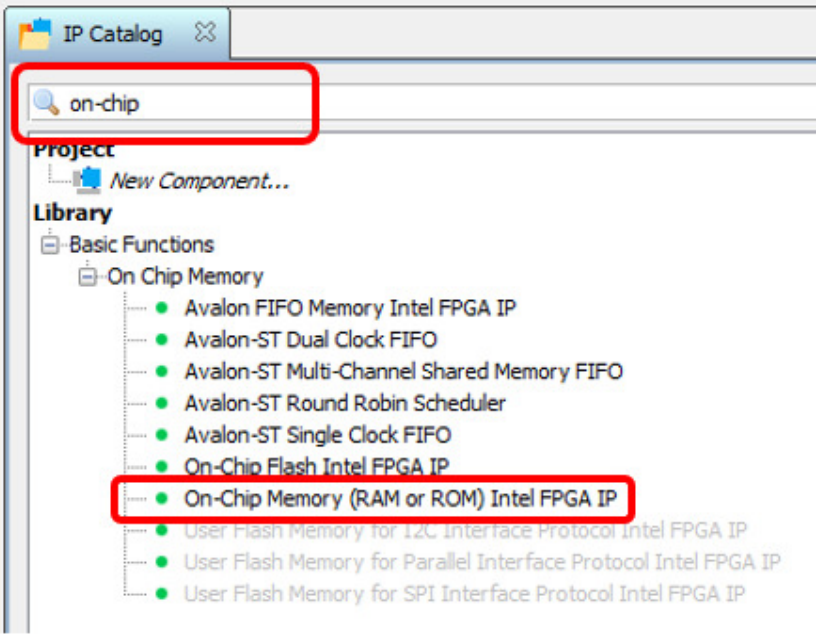
g. By clicking on the dots in column "Connections" connect the NIOS II Processor to Reset and Clock



19. Adding On-Chip RAM

 Copyright © 2019 www.arrow.com

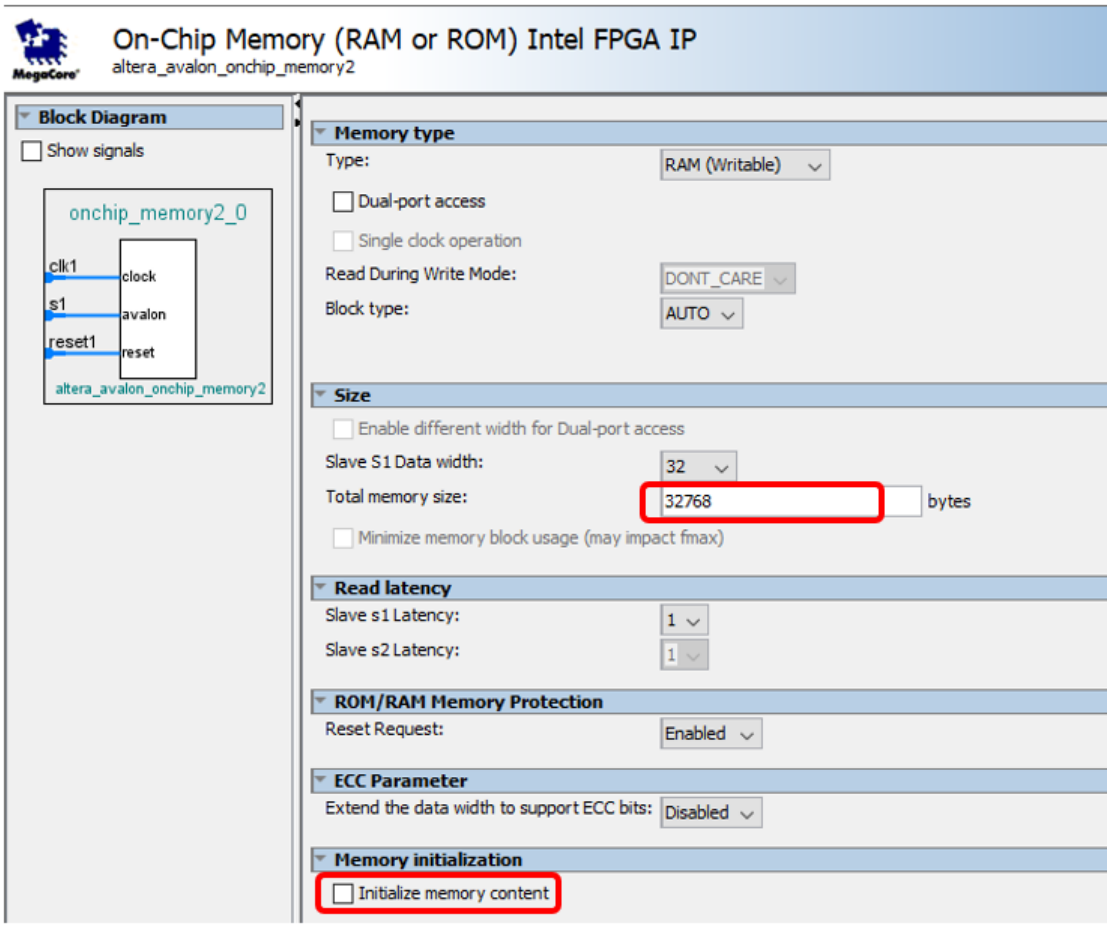
a. In the IP Catalog search for "**on-chip**" and double click "**On-Chip Memory (RAM or ROM) Intel FPGA IP**"



b. In the RAM configuration dialog change the "Total Memory Size" to "**32k**" (32k will automatically be changed to 32768) and deselect "**Initialize memory**

 Copyright © 2019 www.arrow.com

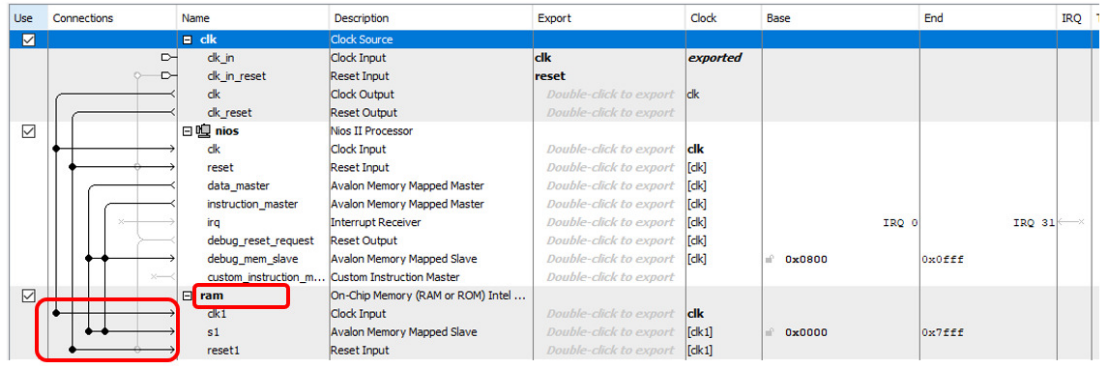
content”



c. Click “Finish”

d. In Column “Name” **right click “onchip_memory2_0”** and rename it to “**ram**”

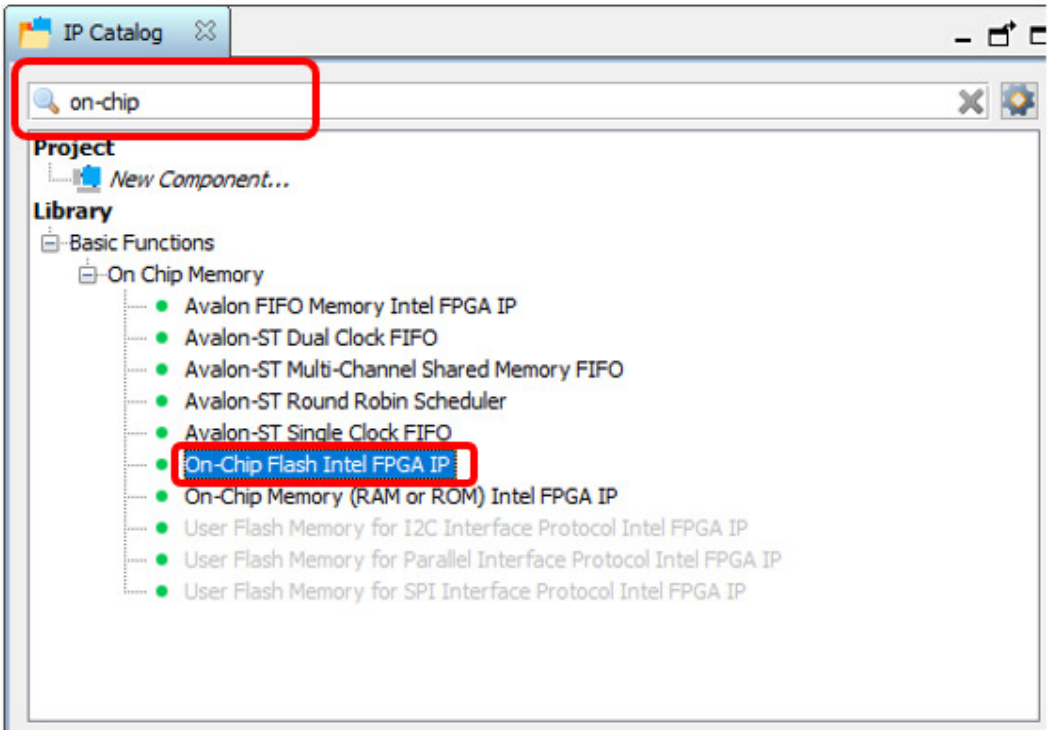
e. By clicking on the dots in column “Connections” connect the component to Clock, Reset and the s1 port to the NIOS data master and instruction master ports



20. Adding On-Chip Flash

 Copyright © 2019 www.arrow.com

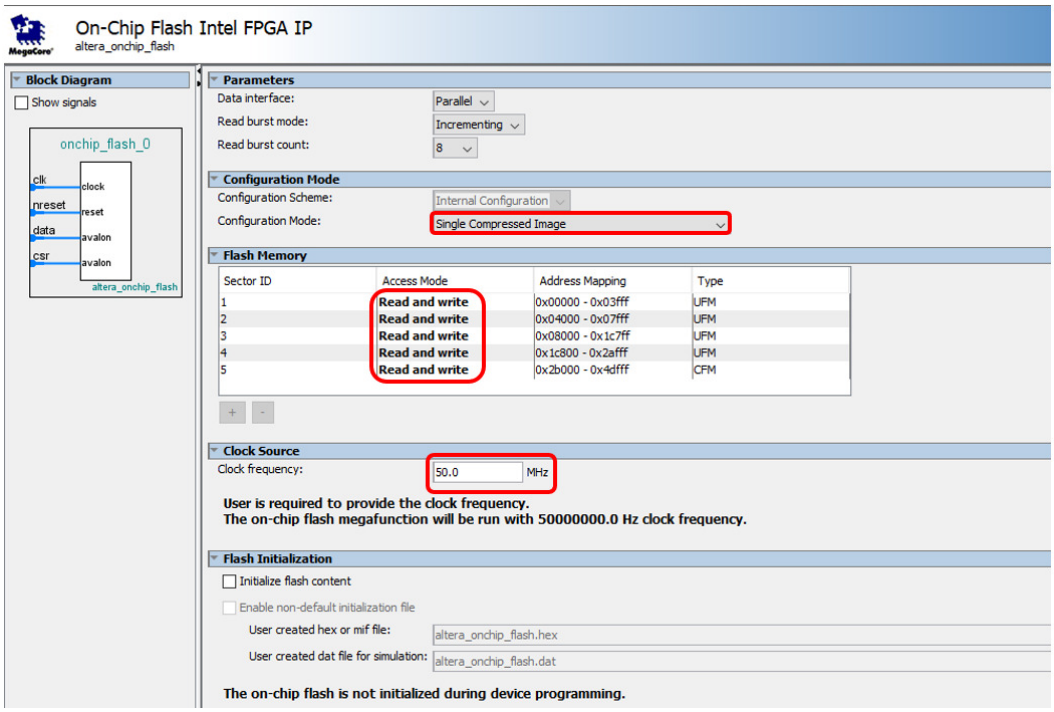
a. In the IP Catalog search for "**on-chip**" and double click "**On-Chip Flash Intel FPGA IP**"



b. Change the "Configuration Mode" to "**Single Compressed Image**"

c. Change the "Access Mode" of all Flash Memory Sectors to "**Read and write**"

d. Set the "Clock frequency" to **50.0** MHz

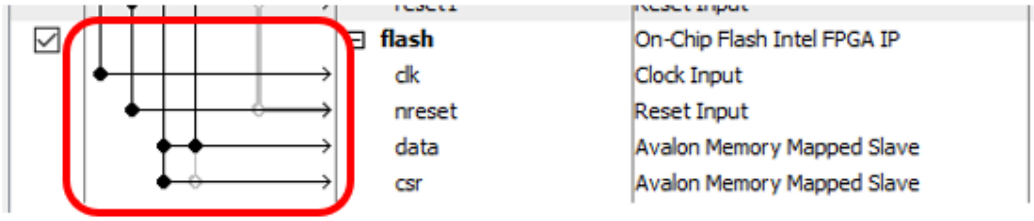


 Copyright © 2019 www.arrow.com

e. Click “Finish”

f. Rename the component to “**flash**”

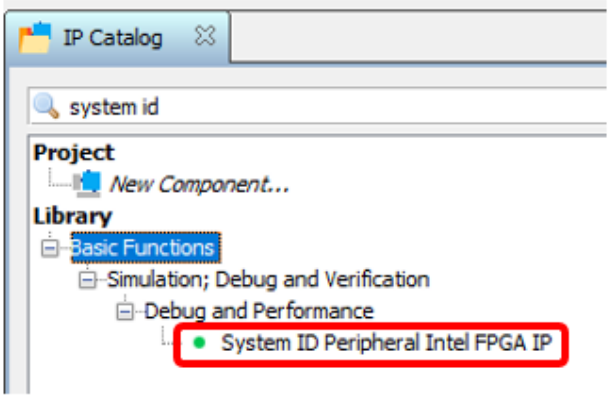
g. Connect the component like this:



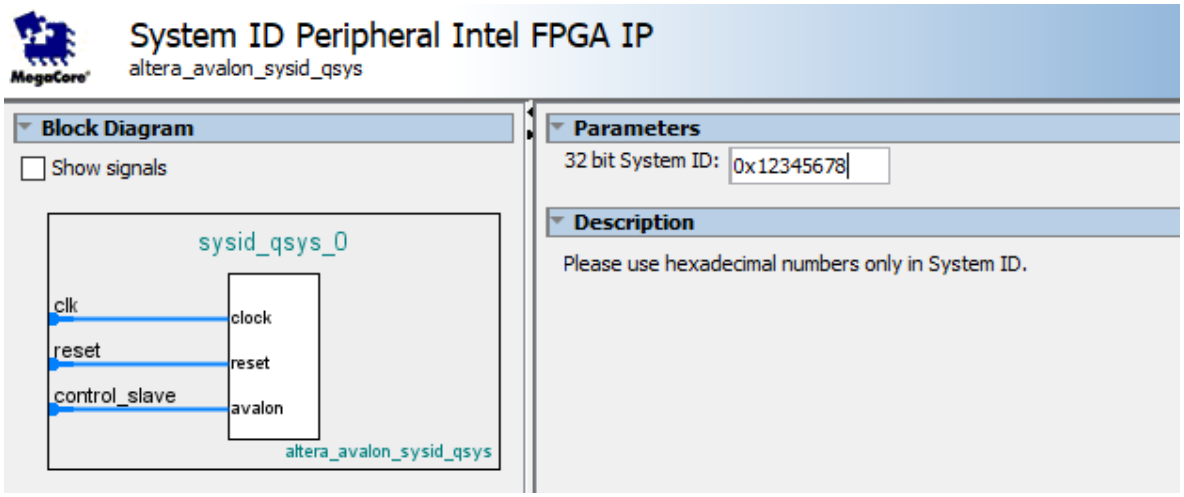
21. Adding System ID IP

a. Search for “system id” in the IP Catalog”

b. Double click “System ID Peripheral Intel FPGA IP”



c. You can enter any 32 bit hex number you like as “32 bit System ID”, e.g. 0x12345678

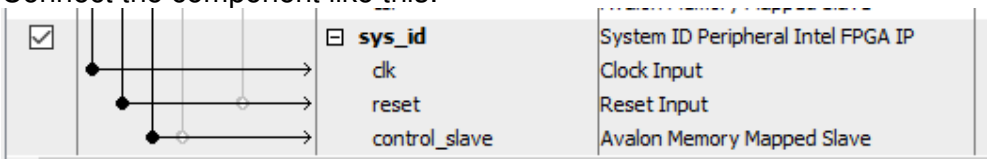


d. Click “Finish”

e. Rename the component to: “**sys_id**”

 Copyright © 2019 www.arrow.com

f. Connect the component like this:



22. Adding PIO IP for the LEDs

a. Search for "pio" in the IP Catalog and double click "**PIO (Parallel I/O) Intel FPGA IP**"

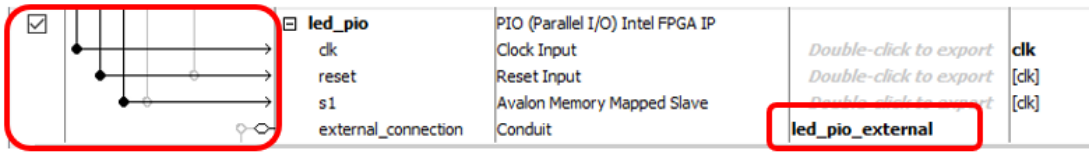
b. Change "Direction" to "**Output**"

c. Click "Finish"

d. Rename the component to "**led_pio**"

e. In the column "Export" double click the last line of the component to export the pio pins and give them the name "**led_pio_external**" (see picture at next step)

f. Connect the component like this:



23. Adding PIO IP for the Push-Button Key

a. Search for "pio" in the IP Catalog and double click "**PIO (Parallel I/O) Intel FPGA IP**"

b. Change the "With (1-32 bits):" to "**1**"

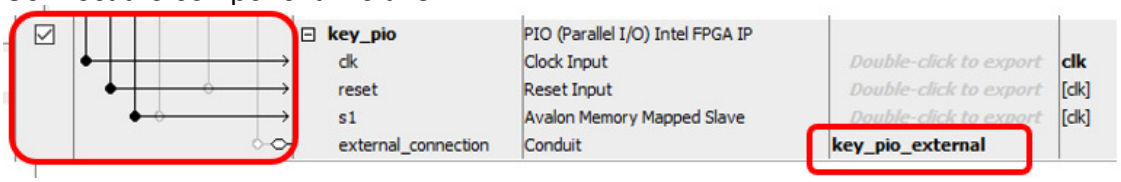
c. Change "Direction" to "**Input**"

d. Click "Finish"

e. Rename the component to "**key_pio**"

f. In the column "Export" double click the last line of the component to export the pio pins and give them the name "**key_pio_external**" (see picture at next step)

g. Connect the component like this:



24. Adding JTAG to UART IP

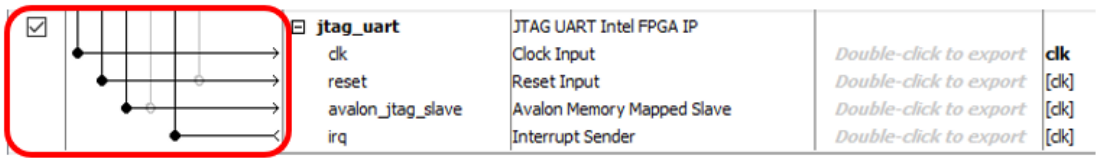
 Copyright © 2019 www.arrow.com

a. Search for "jtag uart" in the IP Catalog and double click "**JTAG UART Intel FPGA IP**"

b. Click "Finish"

c. Rename the component to "**jtag_uart**"

d. Connect the component like this:



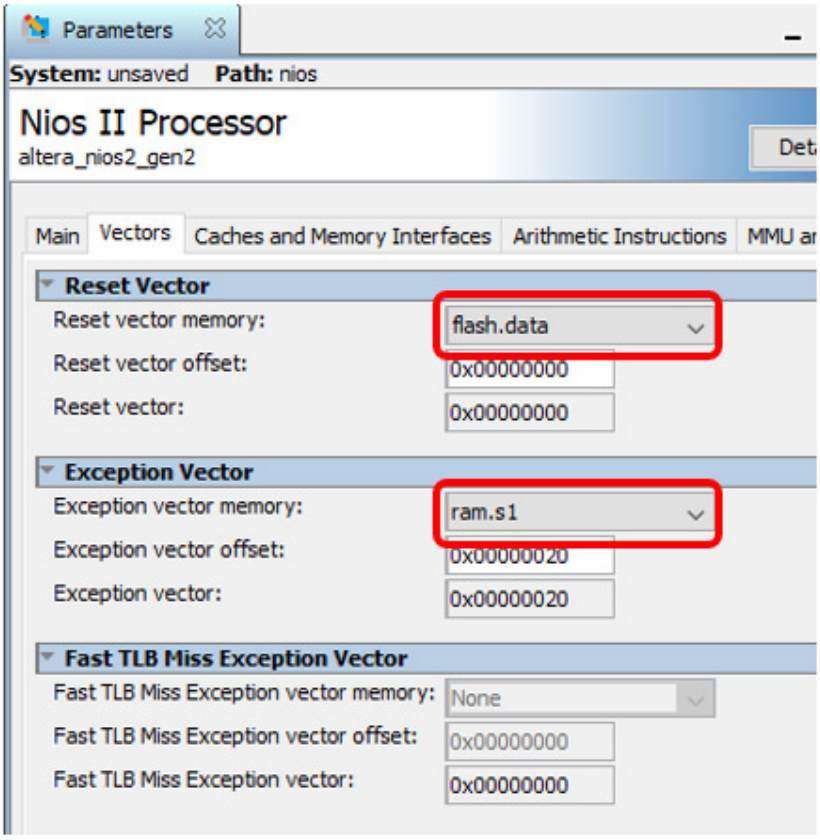
25. Now you are going to remove the error messages

a. Double click the first line of the NIOSII Processor IP to open the parameter window

b. Select the tab "Vectors"

c. Change the "Reset vector memory:" to "**flash.data**"

d. Change the "Exception vector memory:" to "**ram.s1**"



e. Close the parameters window

 Copyright © 2019 www.arrow.com

f. You need to assign the base addresses of all components. This can be done automatically
From the Platform Designer Menu select System -> Assign Base Addresses
Now all components now have unique base addresses.

26. Time to save the NIOSII system

a. Menu: File -> Save

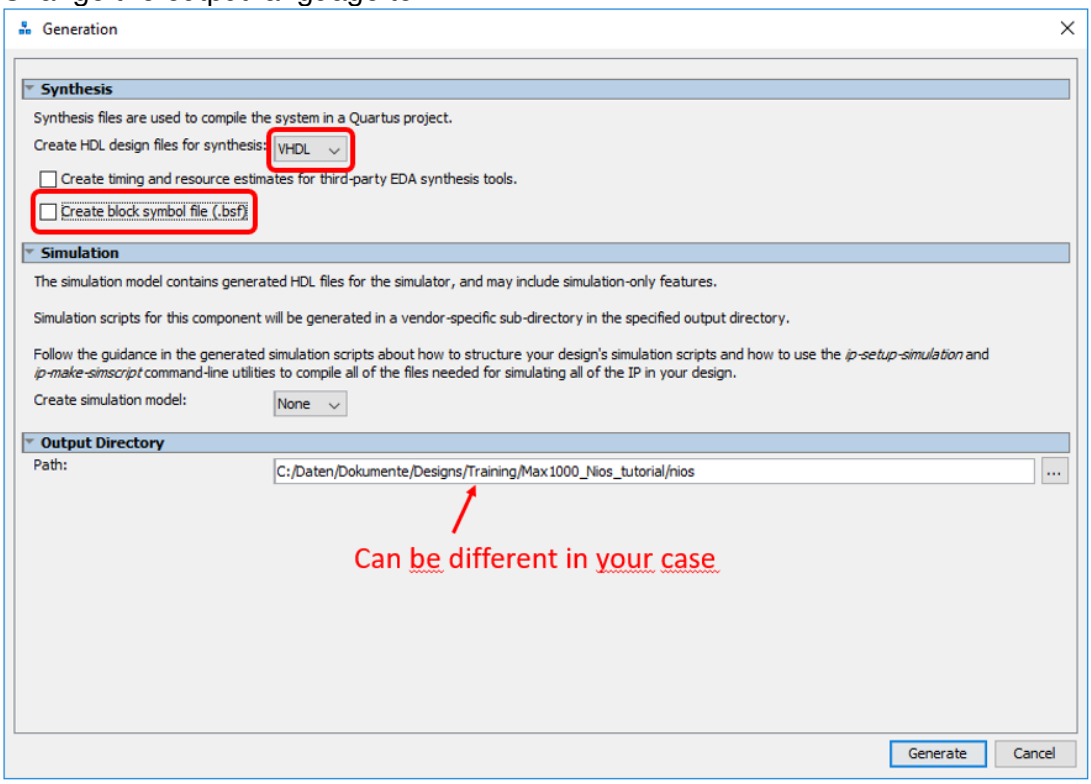
b. Filename has to be **nios.qsys**

c. Click “Save”

27. Generation of the NIOSII system

a. Click “Generate HDL…” at the bottom right of the Platform Designer window (There is also the option to use the menu “Generate -> Generate HDL…”)

b. Change the output language to “**VHDL**”



c. Deselect “Create block symbol file (.bsf)

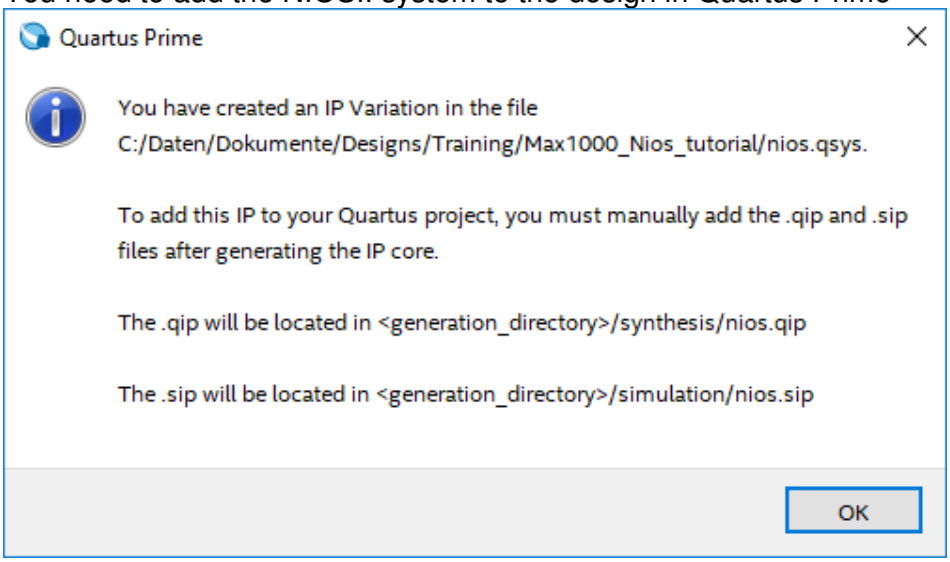
d. Click “Generate”

e. After the successful generation of the system close the summary window

f. Close Platform Designer

 Copyright © 2019 www.arrow.com

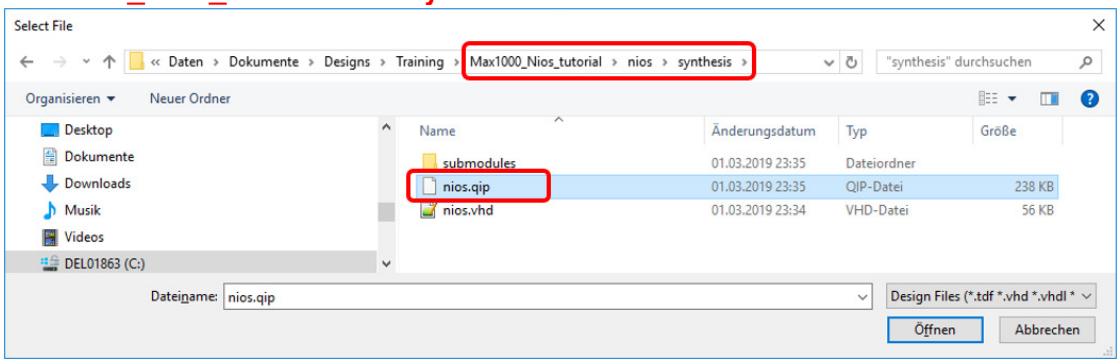
28. You need to add the NIOSII system to the design in Quartus Prime



Click "OK"

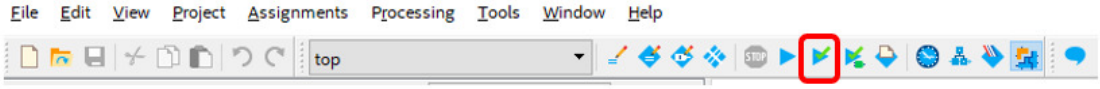
a. Change the Project Navigator view to "Files" and double click the first line (Files) to add the nios.qip file (see Step 9).

b. The file will be found in the subdirectory **Max1000_Nios_tutorial\nios\synthesis**



29. To be able to assign the IO pins to the signal you need to run "Analysis & Elaboration"

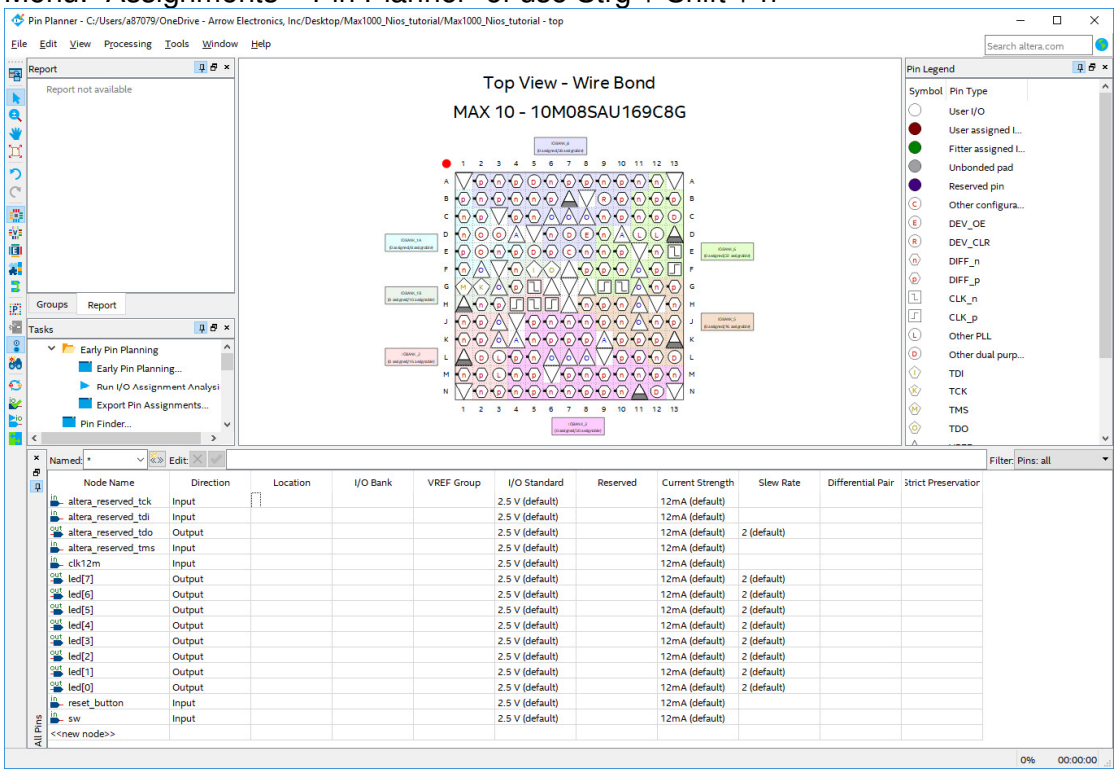
a. Click the icon for "Analysis & Elaboration" or use "Strg + k"



b. After the process is finished you can assign IOs. In this tutorial we will use the graphical "Pin Planner".

Copyright © 2019 www.arrow.com

c. Menu: “Assignments – Pin Planner” of use Strg + Shift + n”



Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation
altera_reserved_tck	Input				2.5 V (default)		12mA (default)			
altera_reserved_tdi	Input				2.5 V (default)		12mA (default)			
altera_reserved_tdo	Output				2.5 V (default)		12mA (default)	2 (default)		
altera_reserved_tms	Input				2.5 V (default)		12mA (default)			
clk12m	Input				2.5 V (default)		12mA (default)			
led[7]	Output				2.5 V (default)		12mA (default)	2 (default)		
led[6]	Output				2.5 V (default)		12mA (default)	2 (default)		
led[5]	Output				2.5 V (default)		12mA (default)	2 (default)		
led[4]	Output				2.5 V (default)		12mA (default)	2 (default)		
led[3]	Output				2.5 V (default)		12mA (default)	2 (default)		
led[2]	Output				2.5 V (default)		12mA (default)	2 (default)		
led[1]	Output				2.5 V (default)		12mA (default)	2 (default)		
led[0]	Output				2.5 V (default)		12mA (default)	2 (default)		
reset_button	Input				2.5 V (default)		12mA (default)			
sw	Input				2.5 V (default)		12mA (default)			
<<new node>>										

d. Here the list of the IO locations and IO Standards

Signal	Location	I/O Standard
clk12m	**H6**	**3.3-V LVCMOS**
led[7]	**D8**	**3.3-V LVCMOS**
led[6]	**C10**	**3.3-V LVCMOS**
led[5]	**C9**	**3.3-V LVCMOS**
led[4]	**B10**	**3.3-V LVCMOS**
led[3]	**A10**	**3.3-V LVCMOS**
led[2]	**A11**	**3.3-V LVCMOS**
led[1]	**A9**	**3.3-V LVCMOS**
led[0]	**A8**	**3.3-V LVCMOS**
reset_button	**E7**	**3.3 V SCHMITT TRIGGER**
sw	**E6**	**3.3 V SCHMITT TRIGGER**

The is no need to assign the “altera_reserved…” signals

a. You can drag and drop the Node Names from the table below to the pads in the “Top View – Wire Bond window”. Alternatively, you can type in the locations in the “Location” column.

b. To assign the I/O standard double click the cells in the “I/O Standard” column and select the right IO Standard from the drop-down menu.

 Copyright © 2019 www.arrow.com

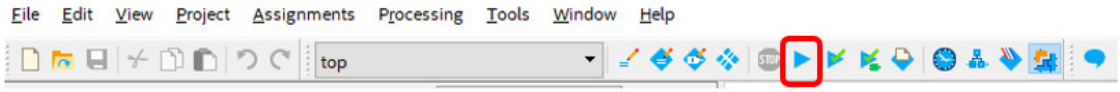
c. The table should look like

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation
altera_reserved_tck	Input				2.5 V (default)		12mA (default)			
altera_reserved_tdi	Input				2.5 V (default)		12mA (default)			
altera_reserved_tdo	Output				2.5 V (default)		12mA (default)	2 (default)		
altera_reserved_tms	Input				2.5 V (default)		12mA (default)			
clk12m	Input	PIN_H6	2	B2_N0	3.3-V LVCMOS		2mA (default)			
led[7]	Output	PIN_D8	8	B8_N0	3.3-V LVCMOS		2mA (default)	2 (default)		
led[6]	Output	PIN_C10	8	B8_N0	3.3-V LVCMOS		2mA (default)	2 (default)		
led[5]	Output	PIN_C9	8	B8_N0	3.3-V LVCMOS		2mA (default)	2 (default)		
led[4]	Output	PIN_B10	8	B8_N0	3.3-V LVCMOS		2mA (default)	2 (default)		
led[3]	Output	PIN_A10	8	B8_N0	3.3-V LVCMOS		2mA (default)	2 (default)		
led[2]	Output	PIN_A11	8	B8_N0	3.3-V LVCMOS		2mA (default)	2 (default)		
led[1]	Output	PIN_A9	8	B8_N0	3.3-V LVCMOS		2mA (default)	2 (default)		
led[0]	Output	PIN_A8	8	B8_N0	3.3-V LVCMOS		2mA (default)	2 (default)		
reset_button	Input	PIN_E7	8	B8_N0	3.3 V Schmitt Trigger		8mA (default)			
sw	Input	PIN_E6	8	B8_N0	3.3 V Schmitt Trigger		8mA (default)			

d. Close the Pin Planner. The assignments will be stored

30. Back in Quartus Prime you need to compile the design to get the needed programming files.

a. Click the “Start Compilation” icon or use “Strg + I”

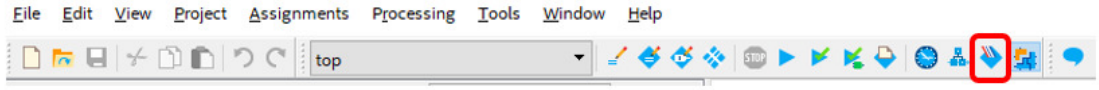


b. This process will take a while. In the Tasks window you can follow the status

c. After the compilation process has finished successfully the Compilation Report shows the result of the compilation process.

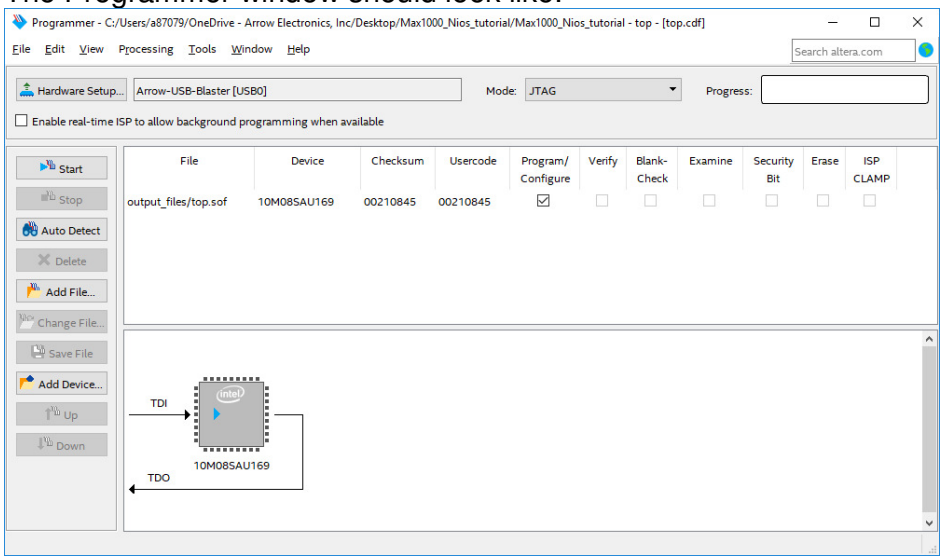
31. To be able to program the SRAM layer of the MAX10 you need to connect the MAX1000 Board via a Micro-USB cable to your computer.

a. Open the Programmer tool by clicking the “Programmer” icon or use the “Tools” menu



 Copyright © 2019 www.arrow.com

b. The Programmer window should look like:



c. The top.sof file is used to program the SRAM layer of the MAX10 device

d. Click “Start” to start the program process

e. After the process has finished you can start the NIOSII Software tutorial.

 Copyright © 2019 www.arrow.com

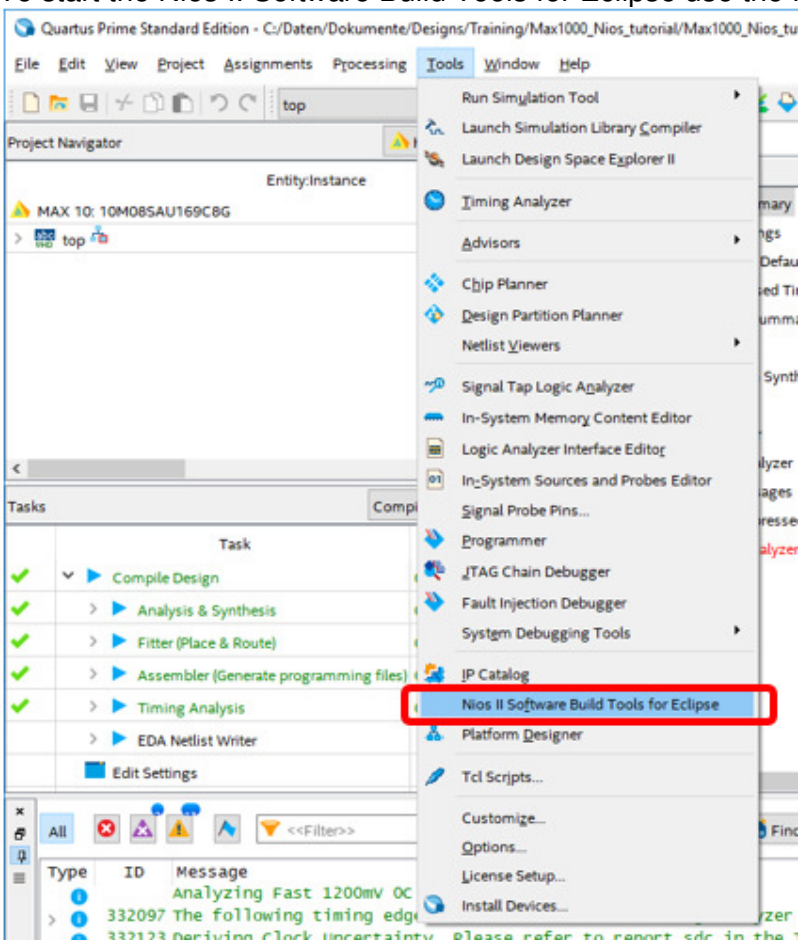
SW-Setup of the Nios II System

In this lab you will generate a C++ based NIOSII program, debug it on the MAX1000 board and generate the HEX file needed to program the UFM.

Hardware setup requirements:

- Micro-USB cable connected between computer and MAX1000 board.
- SOF file programmed to the MAX10 device

1. To start the Nios II Software Build Tools for Eclipse use the Menu “Tools”



2. You need to specify a workspace

   a. Browse to the MAX1000_Nios_tutorial directory and select the directory “**software**”

   b. Click “OK”

 Copyright © 2019 www.arrow.com

c. Nios II – Eclipse starts with an empty window

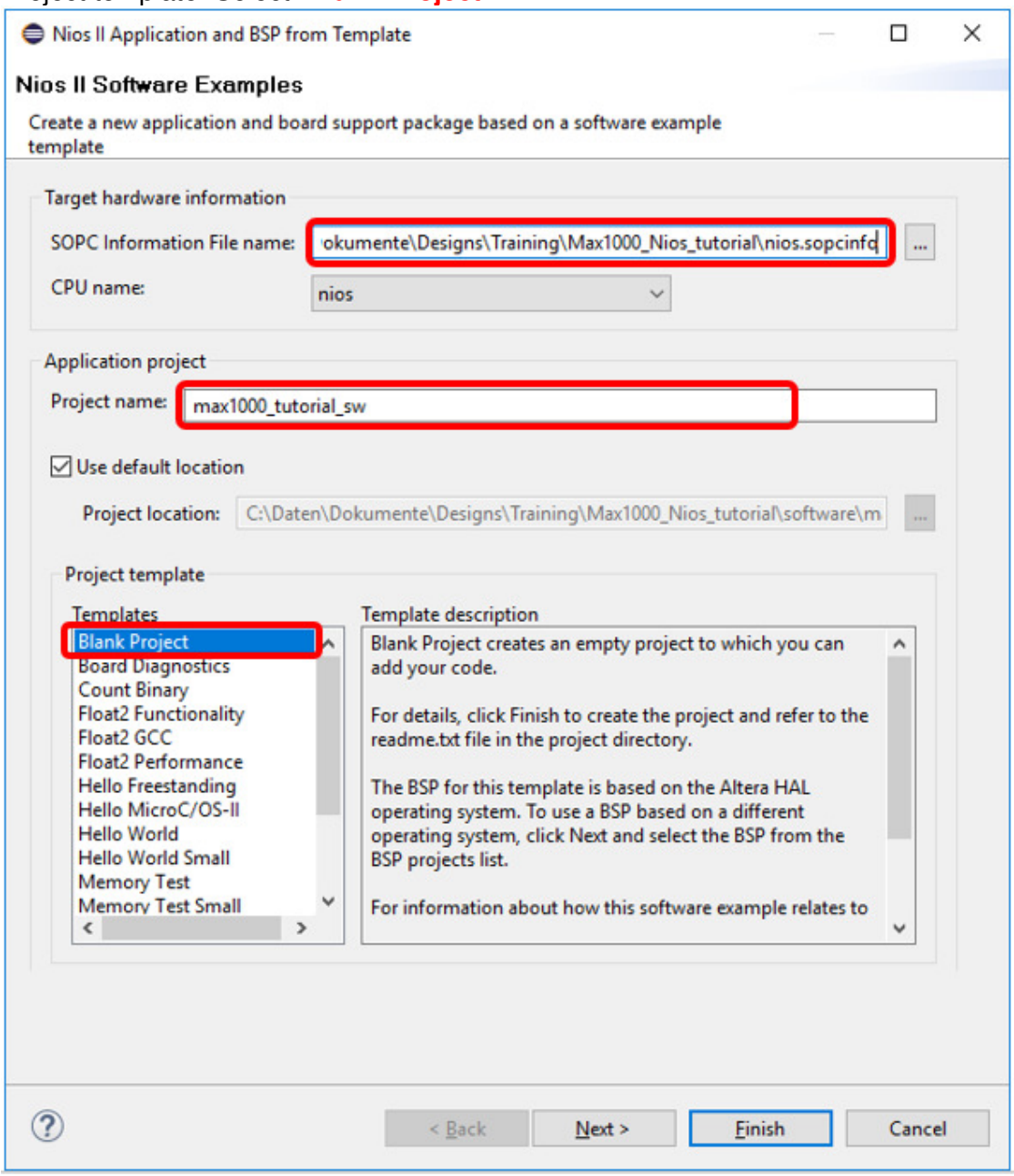
3. Now you generate a new Nios II Application

a. File -> New -> Nios II Application and BSP from Template

b. Select the SOPC Information File name: "**nios.sopcinfo**"

c. Project name: "**max1000_tutorial_sw**"

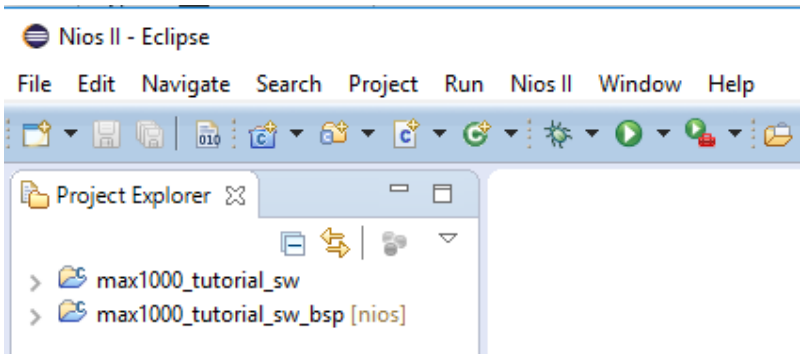
d. Project template: Select "**Blank Project**"



e. Click "Finish

 Copyright © 2019 www.arrow.com

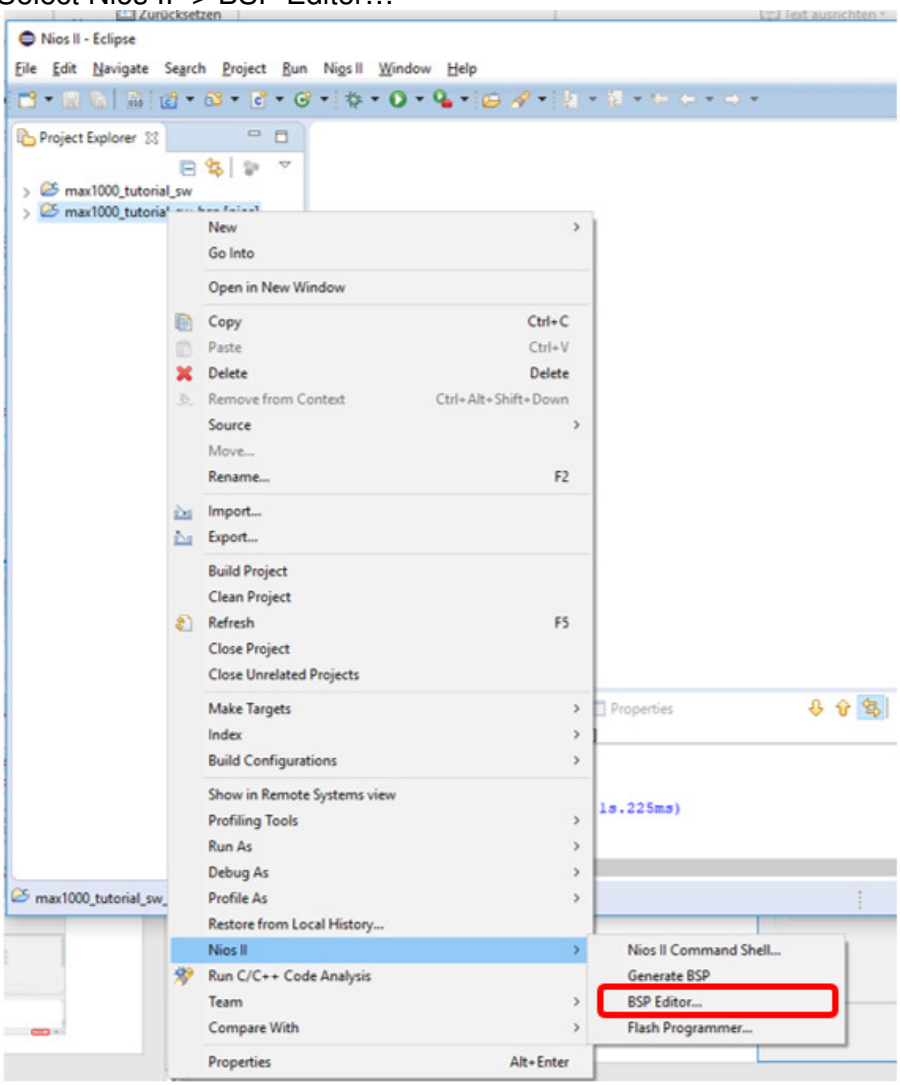
f. In the “Project Explorer” window you now can find two folders



4. Next you need to generate the BSP for the Nios II system

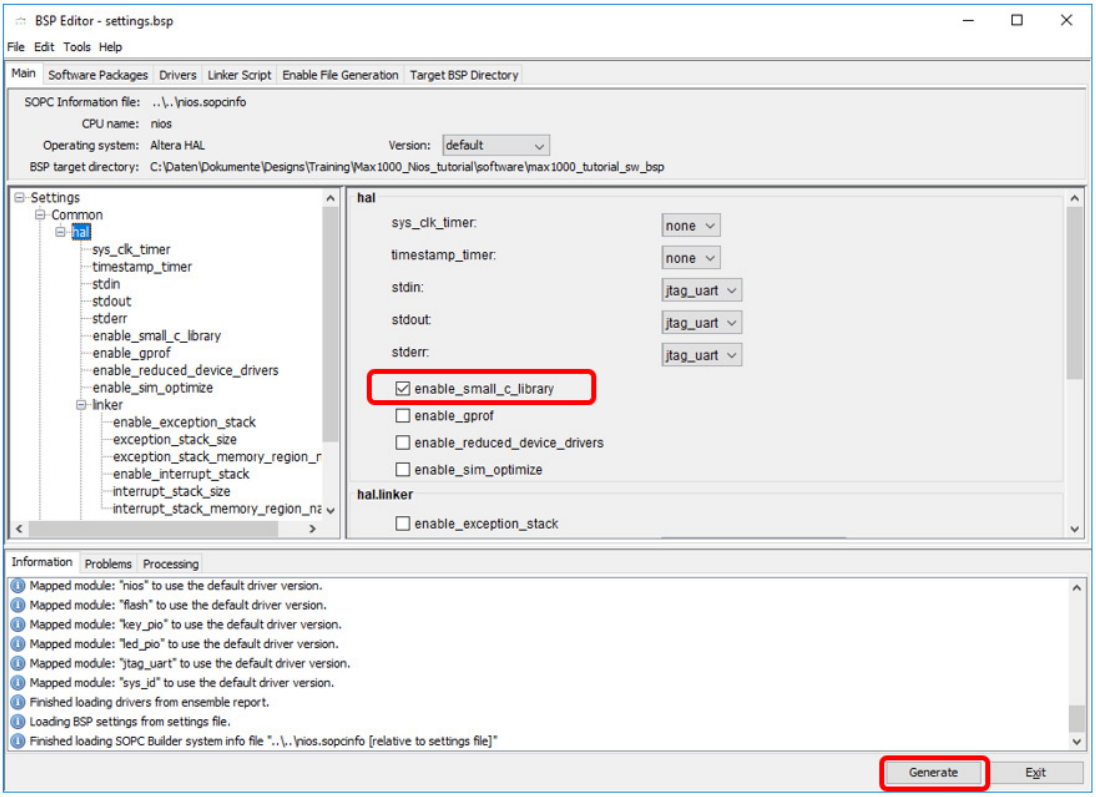
a. Right click the “max100_tutorial_sw_bsp [nios] folder

b. Select Nios II -> BSP Editor…



 Copyright © 2019 www.arrow.com

c. Select “enable_small_c_library”



d. Click “Generate” and “Exit”

5. In the Max1000_Nios_tutorial\software directory is a C file led_blinking.c. This file must be copied into the max1000_tutorial_sw folder.

   a. Use a file browser to navigate to the Max1000_Nios_tutorial\software and copy the led_blinking.c file.

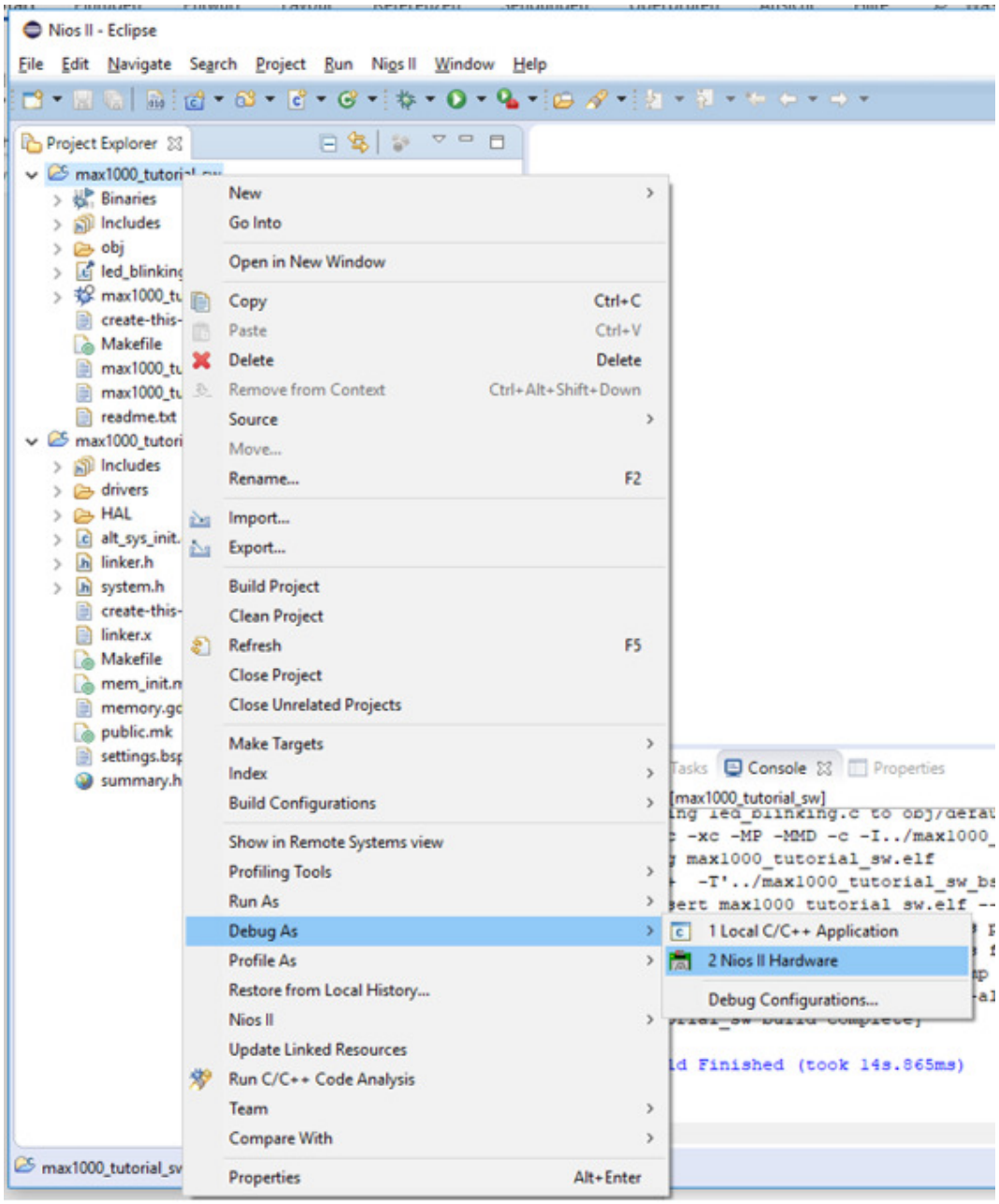
   b. In the Nios II – Eclipse window right click the max1000_tutorial_sw folder and select “Paste”

6. To build the project right click the max1000_tutorial_sw folder and select “Build Project”

7. Now you can debug the project

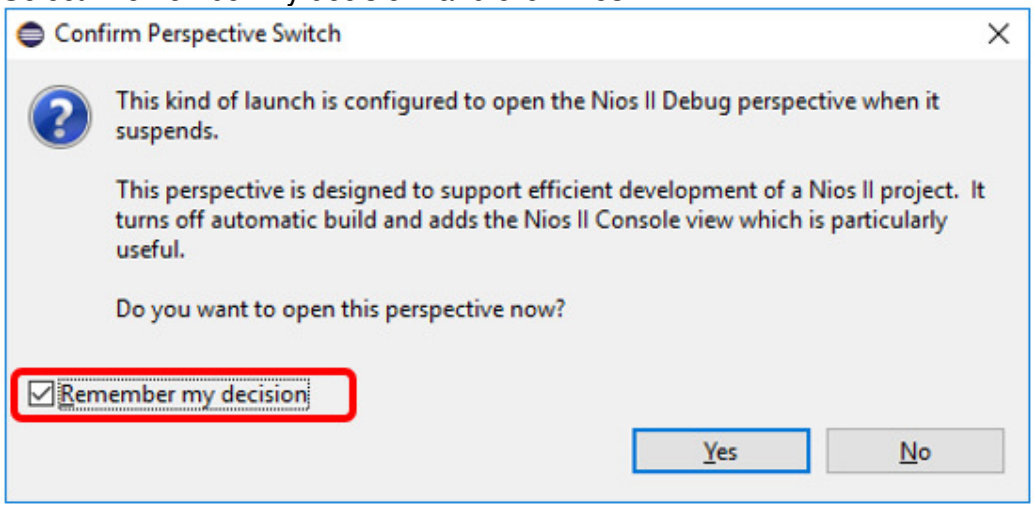
 Copyright © 2019 www.arrow.com

a. Right click the max1000_tutorial_sw folder and select “Debug As” -> “2 Nios II Hardware

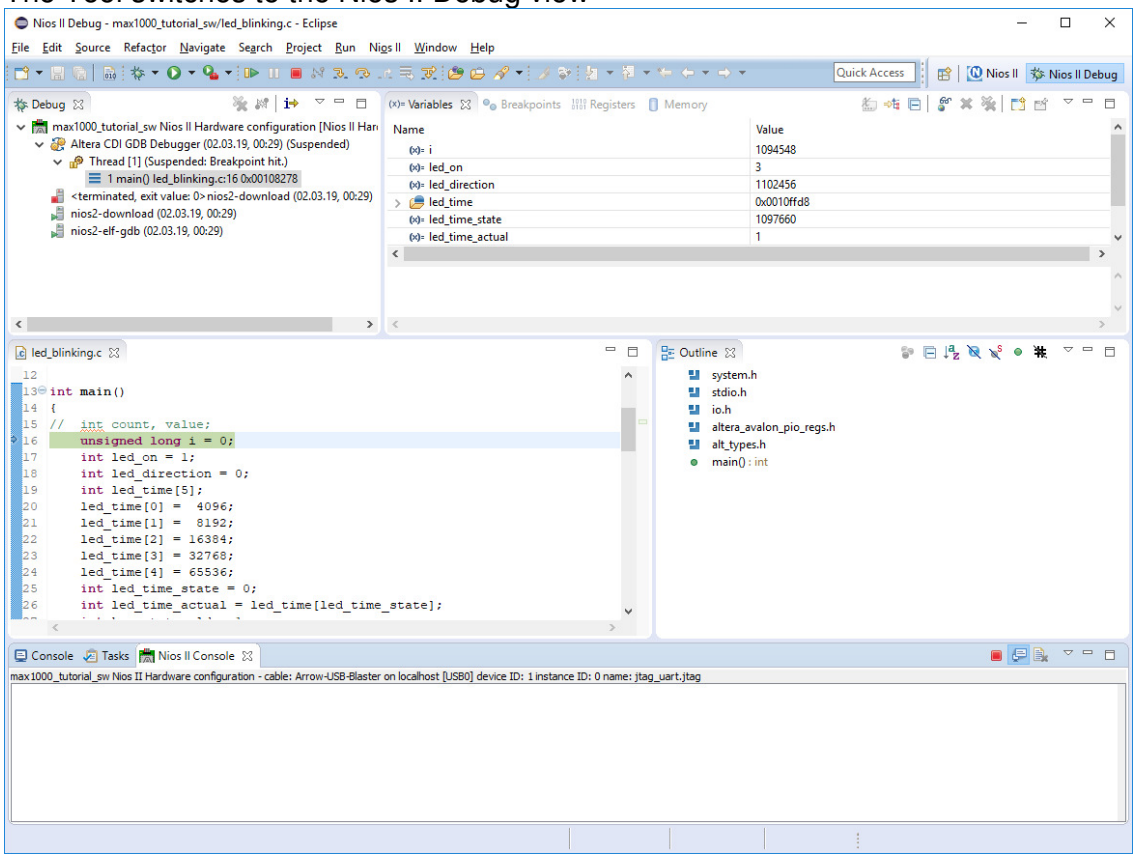


 Copyright © 2019 www.arrow.com

b. Select “Remember my decision” and click “Yes”

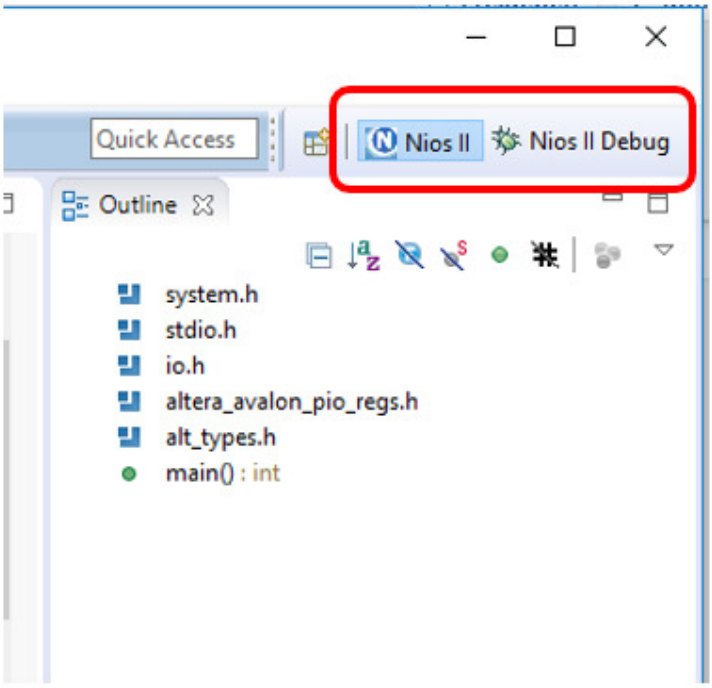


c. The Tool switches to the Nios II Debug view



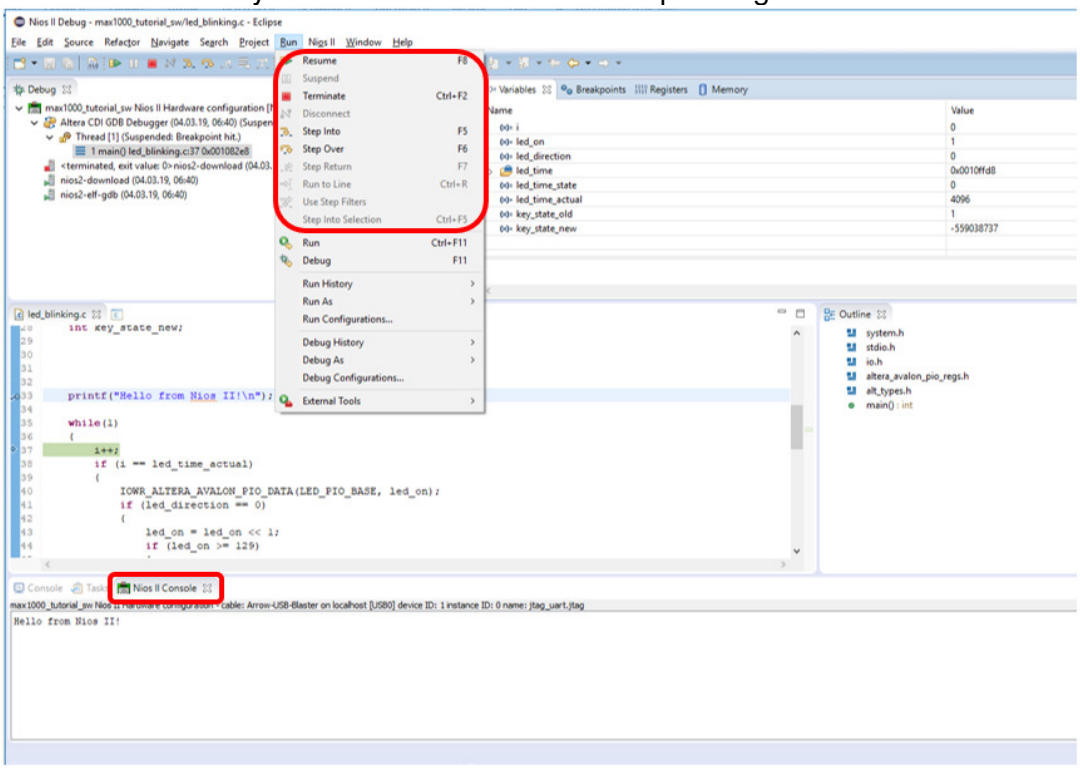
 Copyright © 2019 www.arrow.com

d. Switching between the two views can be made with the two buttons on the top right corner of the Nios II Eclipse window



e. After loading the ELF File to the hardware, the program stops at the first executable line of code.

f. In the menu “Run” you can find the controls to step through the code:



With “Step Over” or F6 the code is executed line by line

 Copyright © 2019 www.arrow.com

With “Step Into” or F5 you will also see the subroutines of every line
With F8 the whole code is executed until the next breakpoint (if there is any)
By double clicking on a line number in the code window you can set and delete breakpoints

g. In the “Nios II Console” you will find the outputs of printf commands.

h. Press F8 to let the program run.

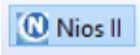
i. By pressing the button next to the LEDs the flashing speed will be reduced and you can find the value of variable “led_time_actual” in the “NIOS II Console”.

j. Press the button several times to see the changes.

k. Important: If you need to reprogram the MAX10 you need first to terminate the “Debug” or “Run” session

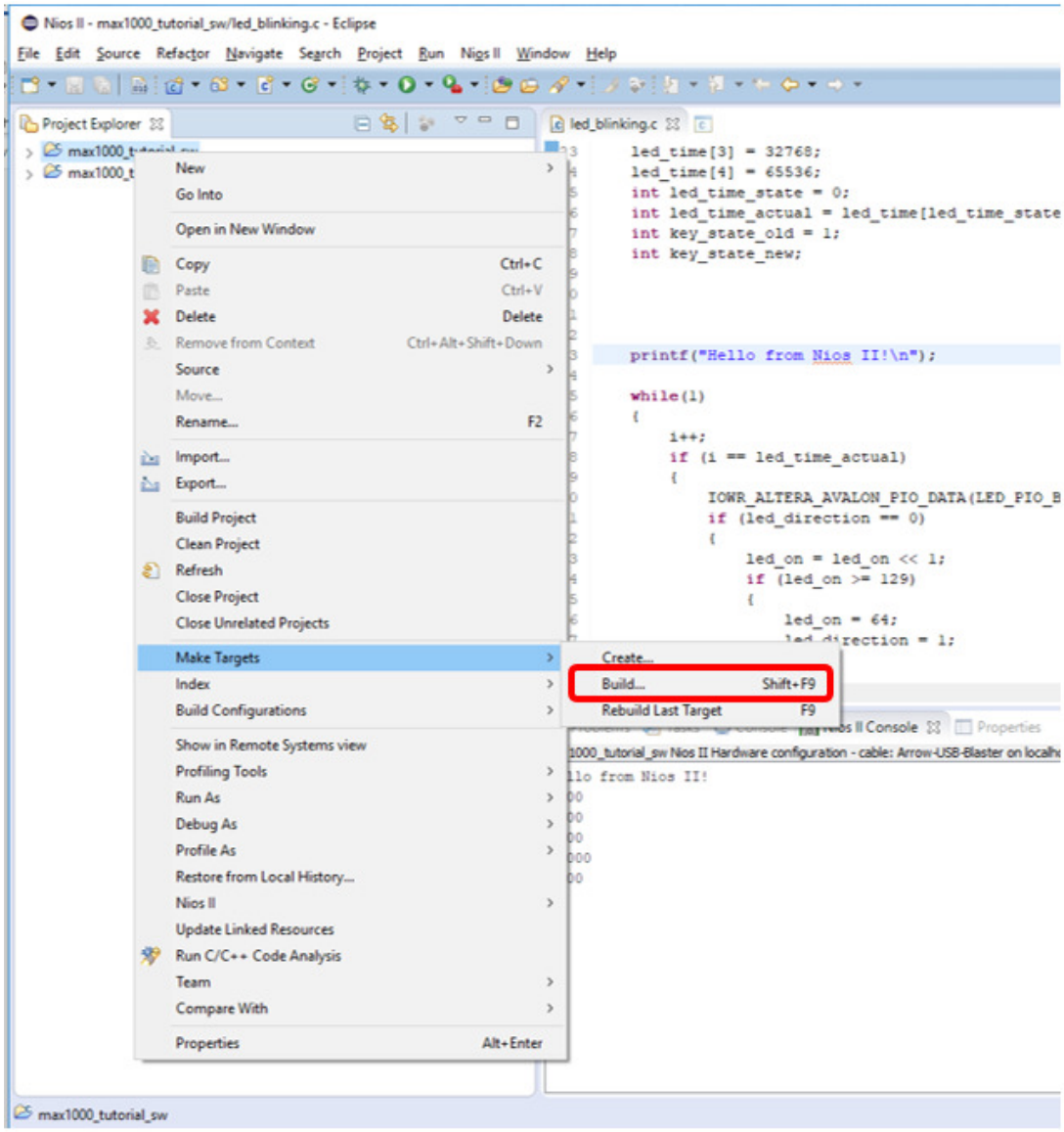
8. After the code is successfully debugged you can deploy the Nios II code to the MAX10 Userflash to have a fully programmed board starting the application by powering the board. Therefore, you need to generate the memory content.

   a. Switch to the Nios II view

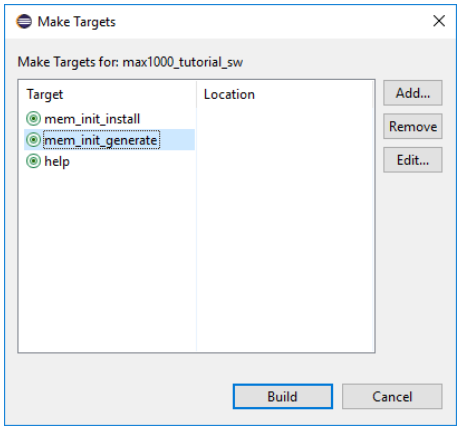


 Copyright © 2019 www.arrow.com

b. Right click the “max1000_tutorial_SW” folder and select “Make Targets” - > “Build…”



c. Double click “mem_init_generate”



 Copyright © 2019 www.arrow.com

d. The “Console” tab shows the process.

e. Terminate the debug session.

f. Switch to the Quartus Prime window

9. To program the Configuration Flash Memory (CFM) and the Userflash Memory (UFM) we need to generate a POF containing the FPGA configuration and the Nios II HEX file.

   a. Select “File” -> “Convert Programming Files…”
   With this tool you can generate different programming file outputs.

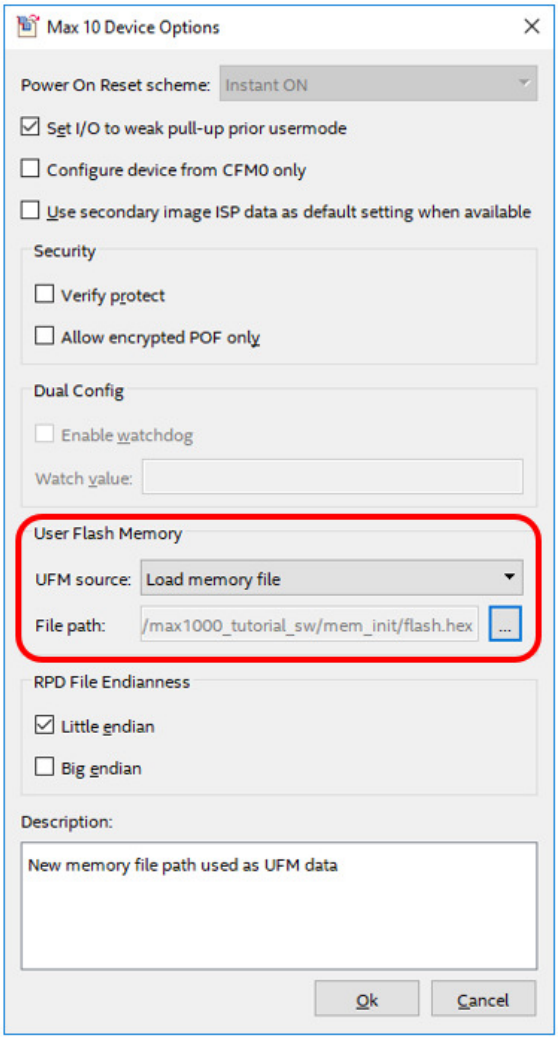
   b. Change the “Mode:” from “1-bit Passive Serial” to “Internal Configuration”

   c. Click on “Options/Boot info”

   d. In the “User Flash Memory” sector change “UFM source:” to “Load memory file”

 Copyright © 2019 www.arrow.com

e. Browse to the location of the generated flash.hex file "..\ Max1000_Nios_tutorial\software\max1000_tutorial_sw\mem_init"



f. Click "Ok"

g. You need to add the SOF file of the FPGA configuration

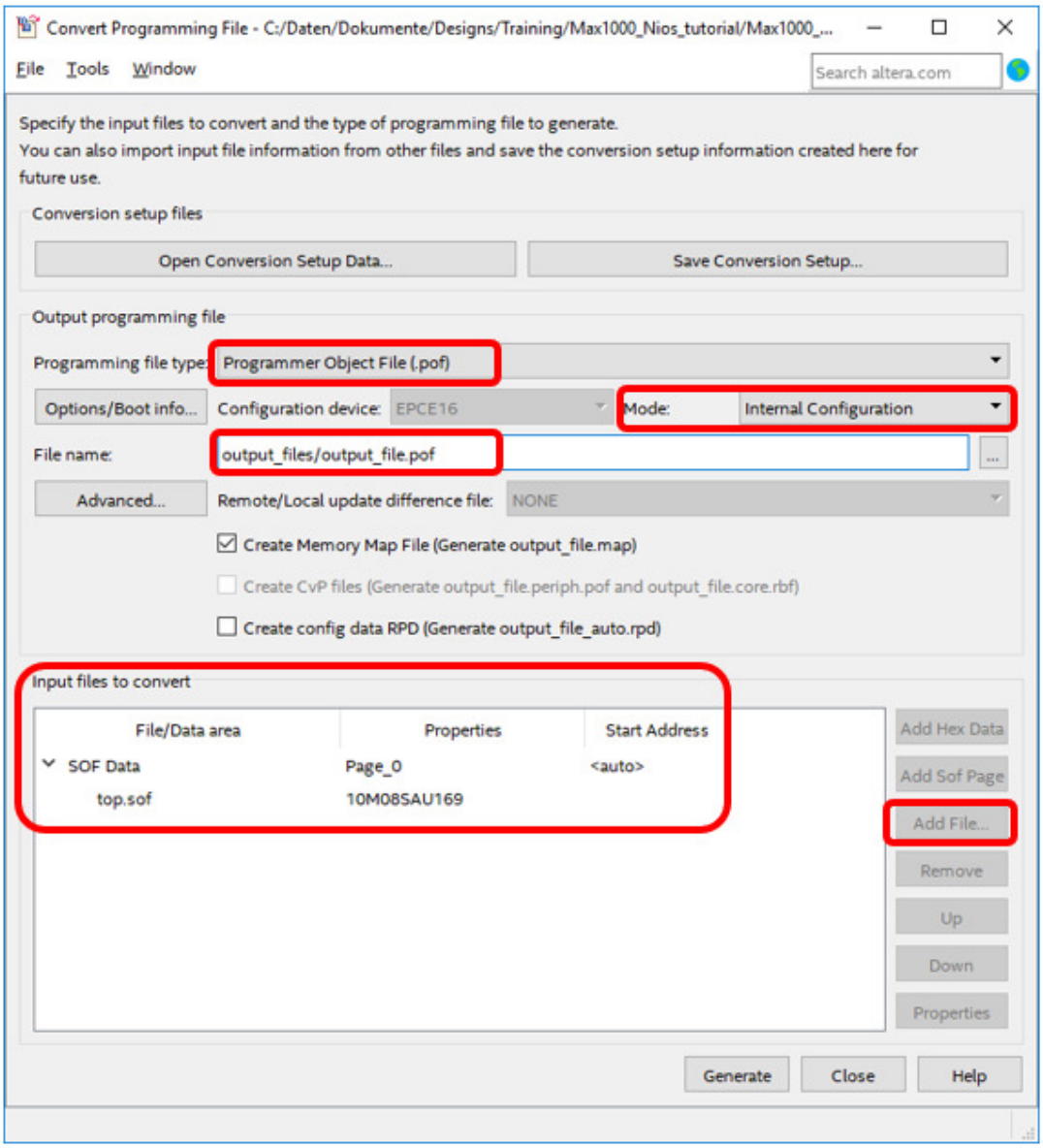
h. Select the "SOF Data" line in the "Input files to convert" section and click on "Add File…"

i. In the "output_files" directory select "**top.sof**" and click "Open"

j. Make sure that the "File name:" is set to "**output_files/output_file.pof"**

 Copyright © 2019 www.arrow.com

k. Check that the window looks like this



l. Click “Generate and “output_file.pof” will be generated

m. Close the “Convert Programming File” window.

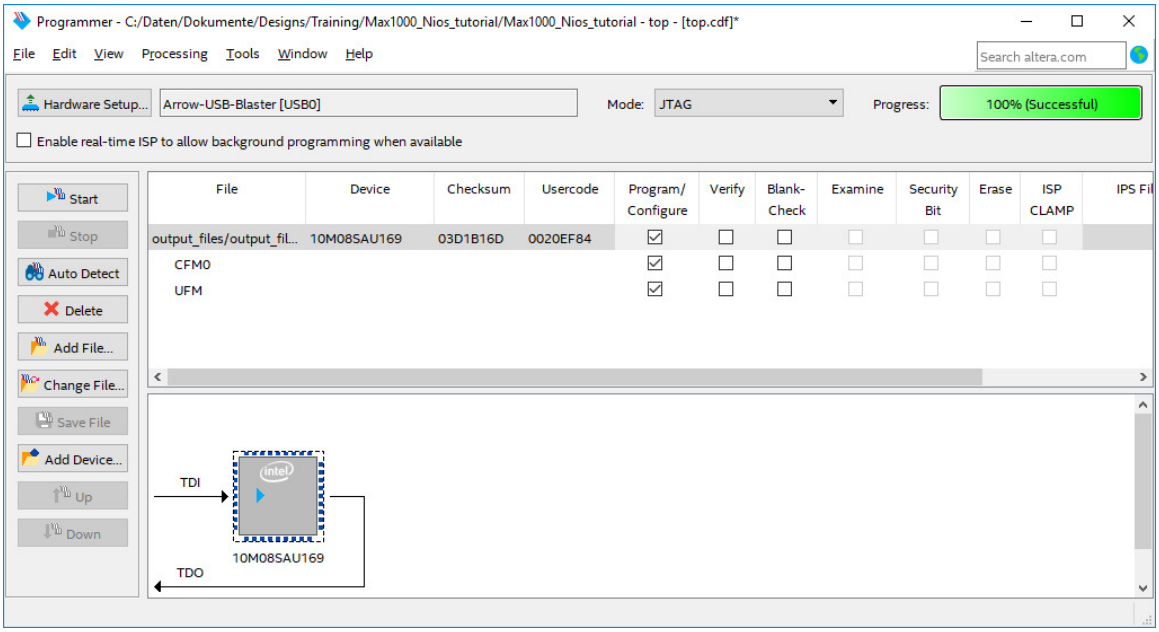
10. Switch to the “Programmer” tool.

11. Select the first line and click “Change File…”

12. Browse to the file “..\output_files\output_file.pof” and click “Open”

13. You will now see three lines. In the first line select the “Program/Configure” checkbox

 Copyright © 2019 www.arrow.com



14. By clicking “Start” the flash of the MAX10 will be erased and programmed with the new content. This will take some time. The progress will be shown by the green bar.

15. After successful programming you can repower the board and the application will start.

Congratulation, you have successfully finished the MAX1000 tutorial.

 Copyright © 2019 www.arrow.com