DESCRIPTION :

This document includes a short How To on making a Nios II boot automatically, right after the DE0\_nano\_SoC (or other board) is powered up and before the Linux is initialized.

1) Quartus II project

a) Create a project based on GHRD

b) in Qsys setup a NIOS II according to **niosII\_gen2\_hardware\_tutorial.pdf** (C:\Users\gartin\QUADCOPTER\_DOC\nios\_II\altera\_example)

c) Generate files (in Qsys) and Compile (in Quartus II)

2) Eclipse NIOS II

a) Create project

New -> Project -> Nios II Application and BSP from Template

in window : - locate .socpinfo file generated by Quartus II project

- set project name

- choose Hello World example

b) write code

- example of simple main file using alt\_ functions (smaller memory footprint)

|  |
| --- |
| **#include** <unistd.h> // usleep()  **int** **main**()  {  alt\_printf("Hello from Nios II!\n");  **int** counter = 0;  **while**(1){  alt\_printf(".");  **usleep**(1000000);  counter++;  **if**(counter >= 128){  alt\_printf(".");  counter = 0;  }  }  **return** 0;  } |

c) build the project

- right-click the project folder -> Build Project

- the Eclipse Nios II produces .elf file in :

"C:\PROJECTS\Quartus\_II\_projects\QUADCOPTER\GHRD\_quad\_v3\sof\_file\software\quad\_nios\_boot\_v1\quad\_nios\_boot\_v1.elf"

- !!! the project path is according the .sopcinfo file, generated by Quatus II compilation !!!

4) Altera Nios2 Command Shell [GCC 4]

- this is a Nios2 console (**NOT** Altera Embedded Command Shell)

- the shell is located "<Quartus\_path>\nios2eds\Nios\_II\_Command\_Shell.bat"

a) navigate to .elf file ( see 2-c) )

- run command :

make mem\_init\_generate

- this converts the .elf to .hex, this can be used to initialize the On-Chip Ram in FPGA

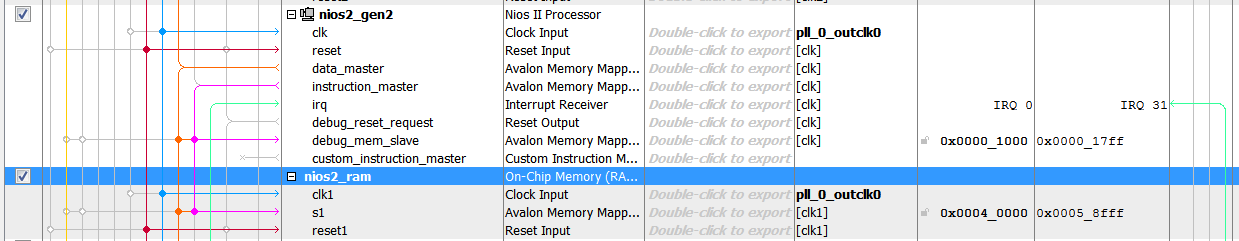
- the file is generated in :

"<Eclipse\_Nios\_II\_project>/mem\_init/hdl\_sim/soc\_system\_<OCRAM\_name>.hex"

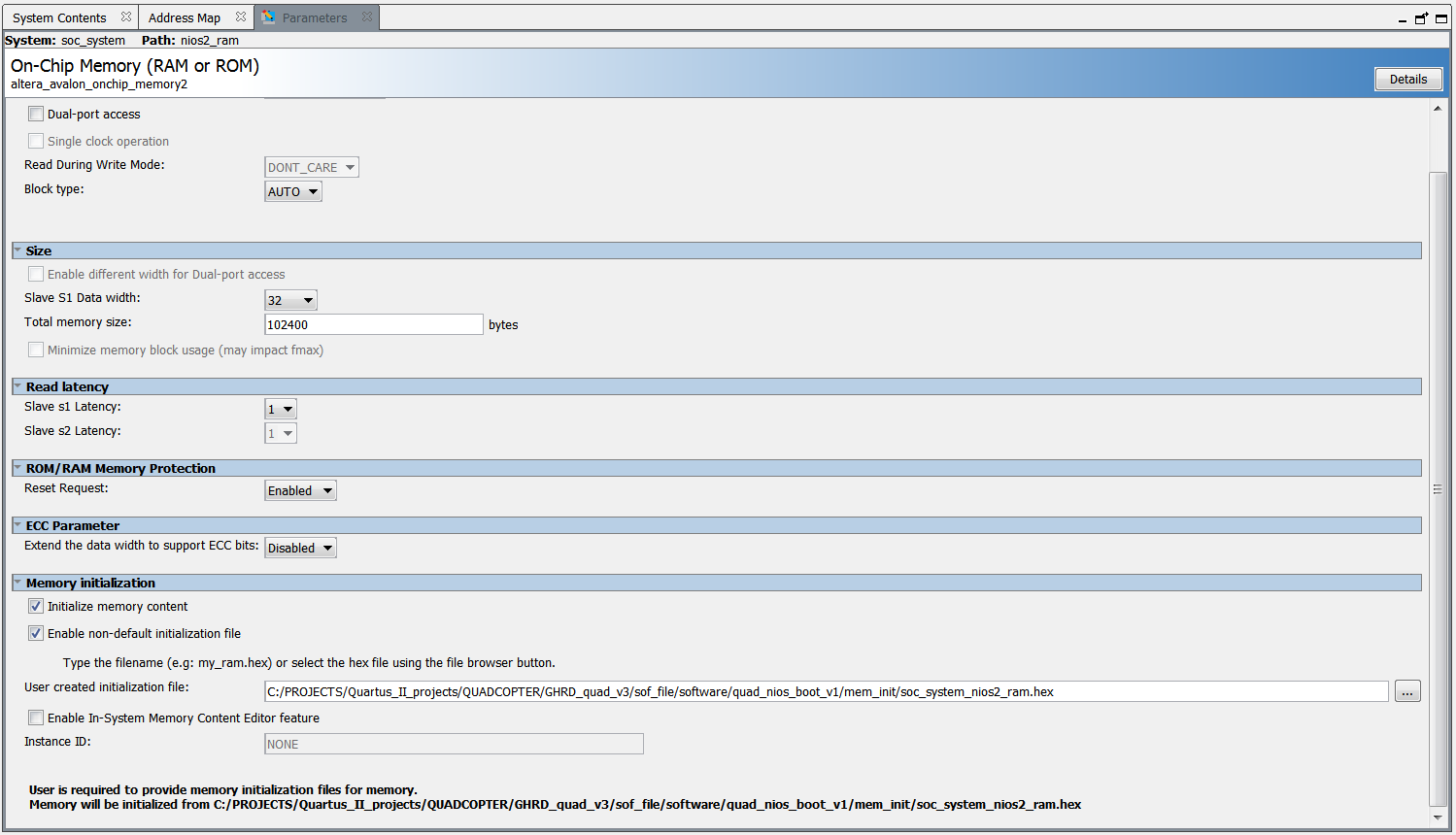
3) Quartus II

a) predefine memory content of On-Chip Ram

- the OCRAM is connected to Nios II (data\_master and instruction\_master port)



- double-click the OCRAM and tick the "Enable non-default initialization file" and the file path



- the path points to the " soc\_system\_<OCRAM\_name>.hex " mentioned in 4-a)

b) re-compile

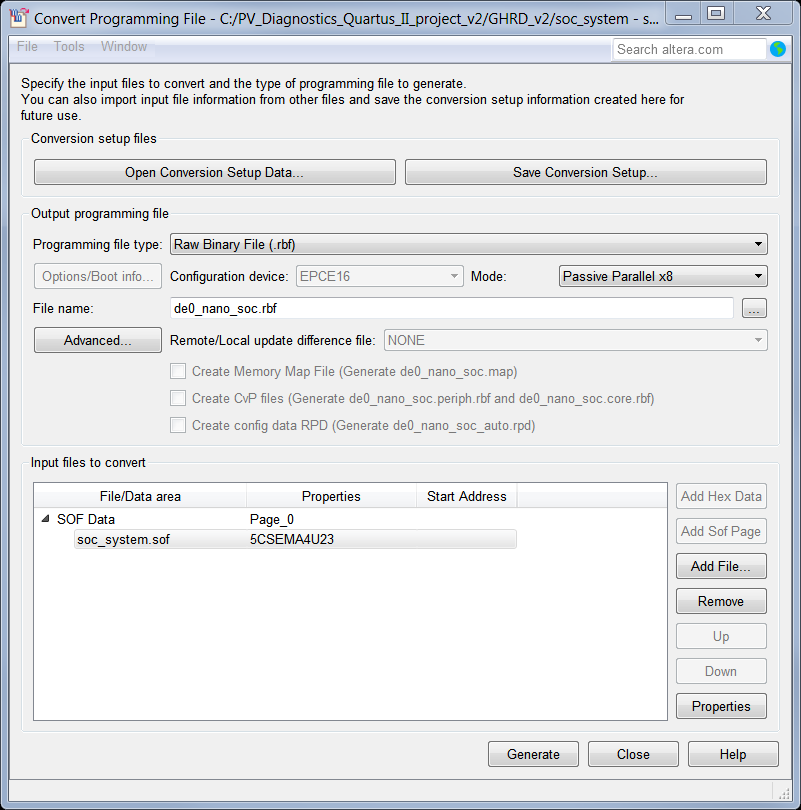
- Generate files in Qsys and re-compile the Quartus II project

- Quartus II creates .socpinfo, .sof, ... files in the Quartus II project root folder

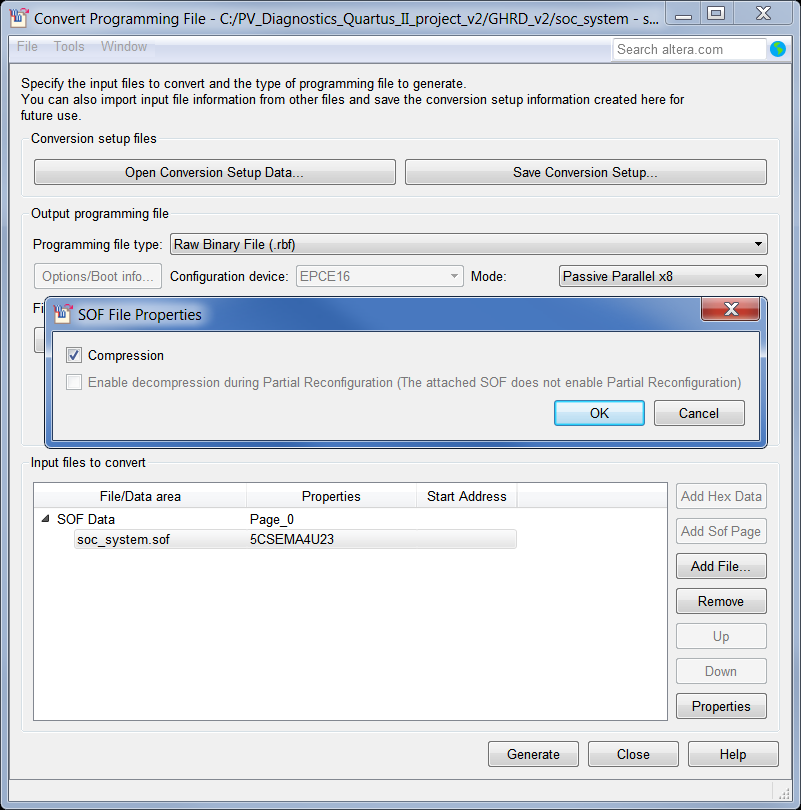
5) Convert

a) Convert files in Quartus II

Quatus II -> Files -> Convert Programming files



- mark the soc\_system.sof and click Properties button



- this produces "de0\_nano\_soc.rbf" in the Quatus II project root folder

- this can be put onto the SD card by copying the file from the host PC running Windows 7 directly to the visible partition of the SD card (under the assumption the new .rbf file replaces the old .rbf file) (check : <https://rocketboards.org/foswiki/Documentation/AVGSRDSdCard>)

6) JTAG Console

a) listen to Hello World program

- the Nios II is connected through JTAG connection, the transmission is then taking place through JTAG (and not COM port as usual)

- run SoC EDS 16.0 Command Shell, type the following command :

nios2-terminal

- the output should look like following :

Hello from Nios II!

..................................

- if more Nios II processors are present in the fpga fabric, the processor can be specified by nios2-terminal -i X, where X is the number/instance/id of the processor