BeMicro Max 10
FPGA Evaluation Kit

BeMicro Max 10 Reference Design containing interfaces to the ADC, Temperature Sensor, DAC, Accelerometer and Serial Flash

Version 14.0.2 **12/19/2014** **Reference Design**

Table of Contents

1. INTRODUCTION AND REQUIREMENTS

This is a comprehensive design example which demonstrates how to use the following interfaces on the BeMicro Max 10 FPGA Evaluation kit:

(i) Accelerometer
(ii) Temperature Sensor
(iii) DAC
(iv) ADC (Internal)
(v) Serial Flash

1.1 Hardware Requirements:

In order to run this example you will need the following Hardware:

- BeMicro MAX 10 FPGA Evaluation Kit
- Mini-USB cable for power

1.2 Software Requirements:

In order to run this example you will need the following software:

- Quartus II Programmer , Version 14.0.2 or Later
- The Altera Nios II Embedded Design Suite (EDS)

If you want to edit and re-create your version of the design, you will need the following additional software:

- Quartus II , version 14.0.2 or Later

2. DESIGN OVERVIEW

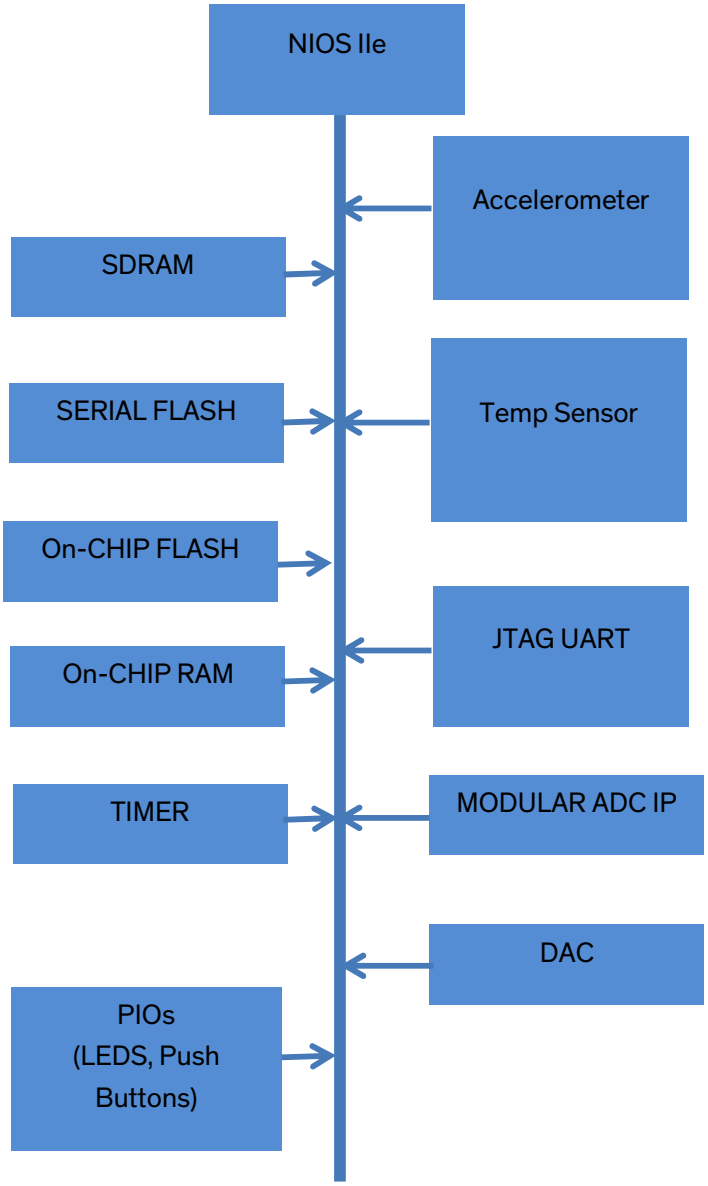


Figure 1: Qsys System

The Qsys system consists of all the interfaces shown in the figure 1. The .c source file explains how to use each of these interfaces can be found under the software folder. If you want to edit and re-create your version, keep only the needed interfaces and remove the other unnecessary interfaces from the Qsys system.

3. STEPS TO COMPILE AND LAUNCH HARDWARE:

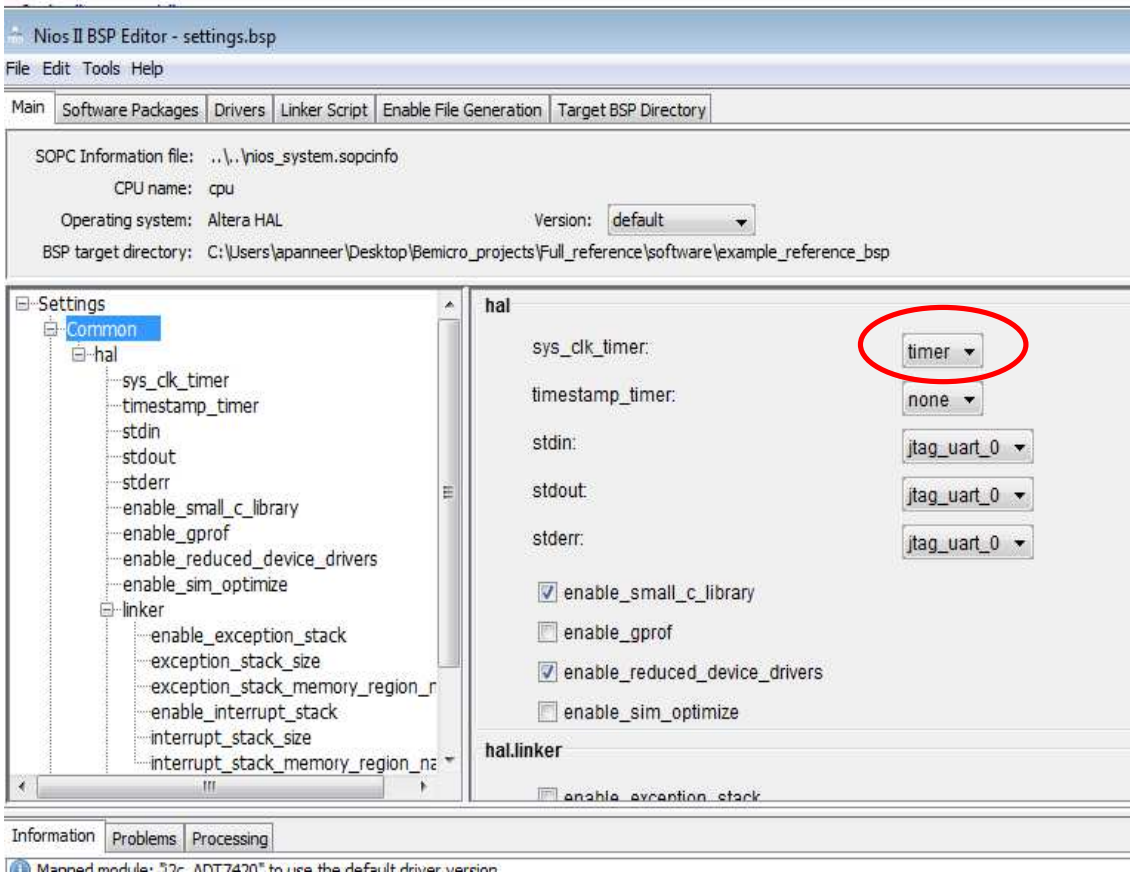
You can skip this step if you don't want to modify the Qsys component . We have provided a full_featured_reference.sof and full_featured_reference.pof under master_image folder. Note that the .sof file is loaded externally each time you power up the kit. The .pof file loads in to on board flash and will remain programmed in the FPGA even after power off. You can directly burn them into your BeMicro FPGA Evaluation kit using the Quartus II Programmer and proceed to Steps to Launch Software.

i) Launch Quartus II software and open the project top.qpf using File->Open Project.
ii) nios_system.qsys is the Qsys system which contains all the modules shown in Figure 1 . Open nios_system.qsys and modify the components if necessary.
iii) Compile the Project by clicking the  button.
iv) Launch the Quartus II programmer from the Tools menu or alternatively by clicking the  button.
v) Download the sof file output_files/top.sof and program the device. Alternatively create a .pof file by selecting Convert Programming Files from the File menu. Set Mode to Internal Configuration and use top.sof as an input file to convert.

4. STEPS TO LAUNCH SOFTWARE:

1) Open **Tools->Nios II Software build tools** from Quartus II. This launches the NIOS II Eclipse IDE where you can modify your C Code.
2) Select the workspace for Nios II Eclipse. Then select **File->New->Nios II application and BSP from Template**
3) In the Window which opens, select the nios_system.sopcinfo file in your Quartus project folder. The .sopcinfo file contains information about the Qsys system, each module instantiated in the project, and parameter names and values contained in the project.
4) Give the name to your Nios project as example_reference, select hello world as the project template and click finish. You can see that example_reference and example_reference _bsp is created on your workspace.
5) Now, we have to replace the hello_world.c source with our source code. The source code for various examples are present under the software/ folder. Choose one example at a time which you want to execute. Remove hello_world.c file from your example_reference project by right click → delete and import the .c example file from the software folder by following the sequence: right_click →import→General →File System → navigate to the desired example directory (e.g. accelerometer_example) → checkbox the source .c file → Finish . All of the examples write information out to the console that inform the user of what part of the devkit they are exercising.
6) Since the design needs the sys_clk_timer, we have to specify the sys_clk_timer in the bsp editor. Right click on example_reference_bsp and select Nios II -> BSP Editor
7) Select sys_clk_timer drop down list box and select timer as the sys_clk. Select File->Save and Click Generate

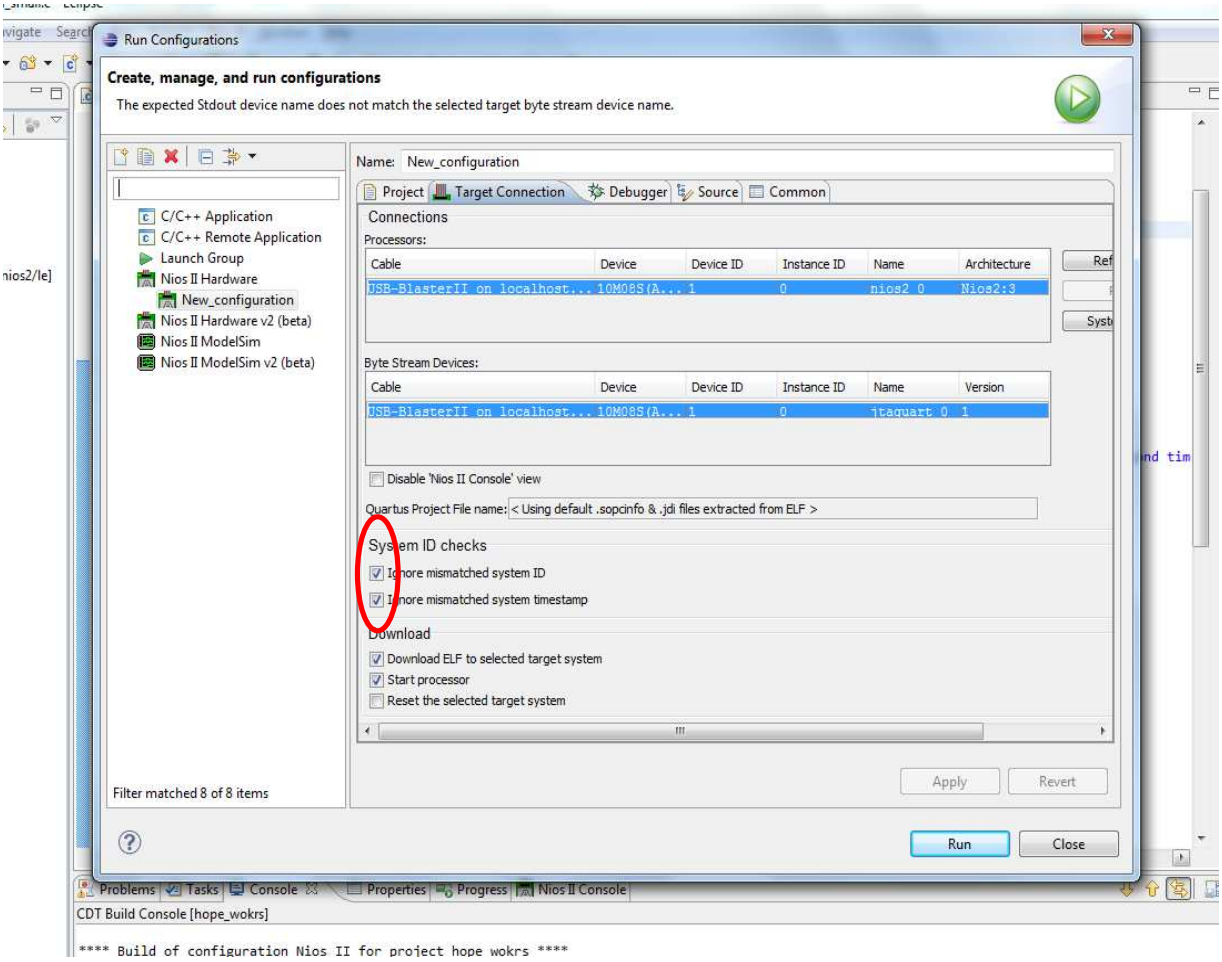
Steps to Launch Software:



8) Right Click on the example_reference and Select **Build Project** to build the project. The initial build may take some time.
9) Once the build is finished, to run the project, right click on the project and select Run As -> Run Configurations.
10) Double click on Nios II Hardware, and new configuration opens on the right pane. Make sure you select the project name as example_reference and .elf file as example_reference.elf .
11) Select Target Connection Tab. Then Check the following two check boxes

Ignore mismatched system ID
Ignore mismatched system timestamp

Steps to Launch Software:



12) Click Apply and Run. Wait for the program to complete load into the NIOS inside the FPGA. The lower right displays launching new configuration percentage complete status.
13) You will observe the results in the Eclipse console window.
14) Replace the source code with another example by following step 5.Then repeat step 8 and click run  . Each example has its own characteristics. Refer to the various peripheral datasheets and source code to get detailed information about the operation of each of the peripherals.