Xenomai: Real-Time Framework for Linux

http://xenomai.org/

Xenomai on NIOS II Softcore Processor: a step by step Guide

Author	**:**	P. Kadionik and Ph. Gerum
Version	**:**	1.2
Date	**:**	04/01/2010

Patrice Kadionik: kadionik@enseirb-matmeca.fr
Philippe Gerum: rpm@xenomai.org

TABLE OF CONTENTS

Summary

Hardware	
Target processor	NIOS II
Vendor	Altera
Synthesis tools	Quartus II version 9.0 at least
Target board example	Altera Stratix 1S10
Software versions used in this guide	
Linux distribution	Fedora 12
Linux version for NIOS II	µClinux 2.6.30
Cross compiler for NIOS II	3.4.6
ipipe patch version for NIOS II	adeos-ipipe-2.6.30-nios2-1.1-00.patch
Xenomai version	2.5.2

Document version:

Date	**Version**	**Comments**
11/16/2009	1.0	Initial version
01/10/2010	1.1	Updated to Xenomai 2.5.0. Text corrections
04/01/2010	1.2	Updated to Xenomai 2.5.2. Text corrections. Adding design rule on timer creation

1. INTRODUCTION

This guide explains how to set-up your SoPC (*System on Programmable Chip*) design for running Xenomai. Altera's Stratix 1S10 board is the reference target throughout the document. However, the contents of this guide should be applicable to similar hardware as well.

2. THE ALTERA NIOS II SOFTCORE PROCESSOR

The NIOS II processor (second generation of the NIOS processor) is a RISC softcore processor with a Harvard architecture. It has up to 6 pipeline stages and has a 32-bit data bus.

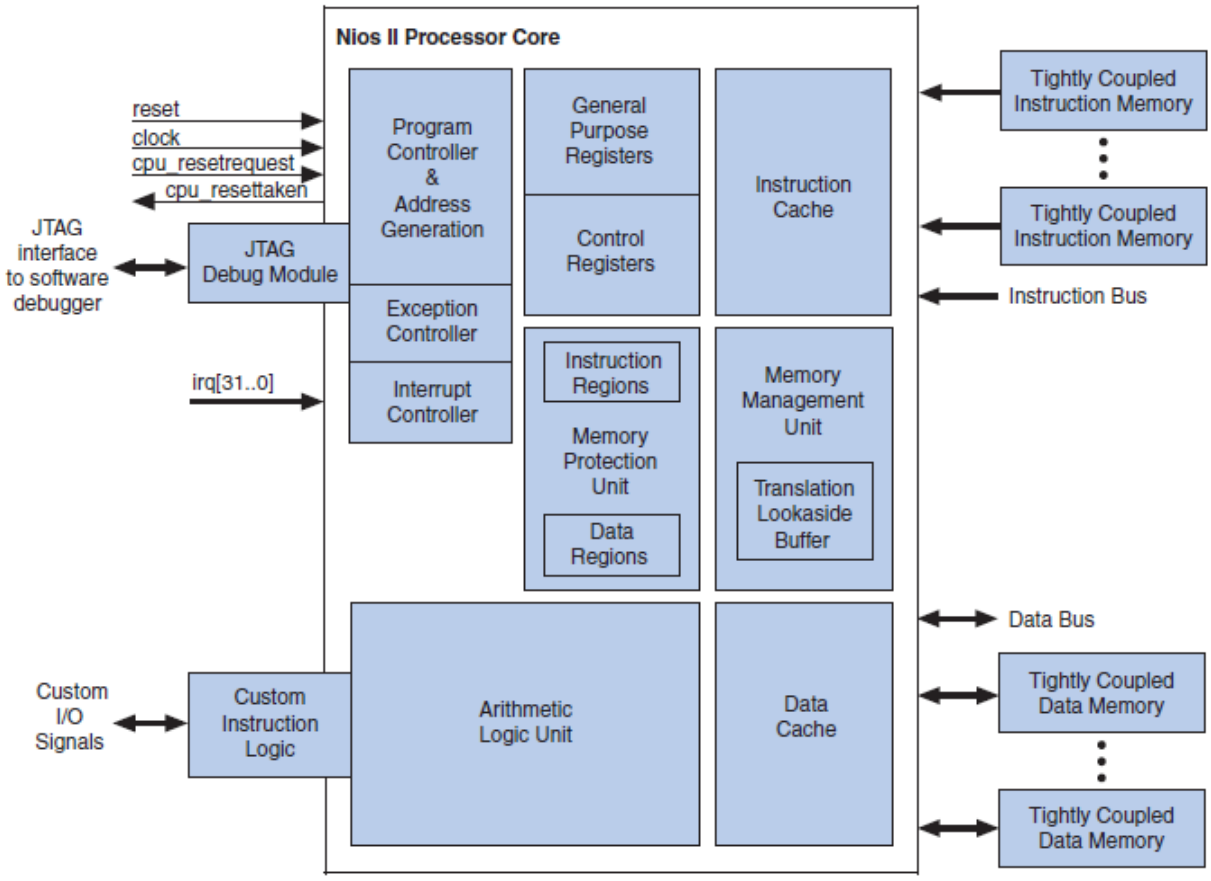


Figure 1: NIOS II processor architecture

The following table resumes its main characteristics:

NIOS II Processor
■ RISC architecture ■ 32-bit instructions ■ 32 32-bit general registers ■ 32 IRQs ■ Instruction and data caches ■ Custom instructions

Figure 2: Main characteristics of the NIOS II processor

During the configuration of the NIOS II processor with the *SOPC Builder* tool, you can choose between three versions for the processor:

- The *Economy* version that uses less silicon area on the FPGA circuit.
- The *Standard* version that allows a good compromise between area and speed.
- The *Fast* version is the fastest (in frequency) version.

	NIOS II /f	**NIOS II /s**	**NIOS II /e**
Pipeline	6 levels	5 levels	No
HW Multiplication	1 Cycle	3 Cycles	By SW
Branch Prediction	Dynamic	Static	No
Instruction Cache	Configurable	Configurable	No
Data Cache	Configurable	No	No
Custom Instructions	Up to 256		

Figure 3: The different versions of the NIOS II processor

The following figure shows the performances in DMIPS (*Dhrystone Million Instructions per Second*) and the used area in Altera Logic Elements for different versions of the NIOS II processor on different families of Altera FPGA circuits (Stratix, Stratix II, Cyclone, Cyclone II ...).

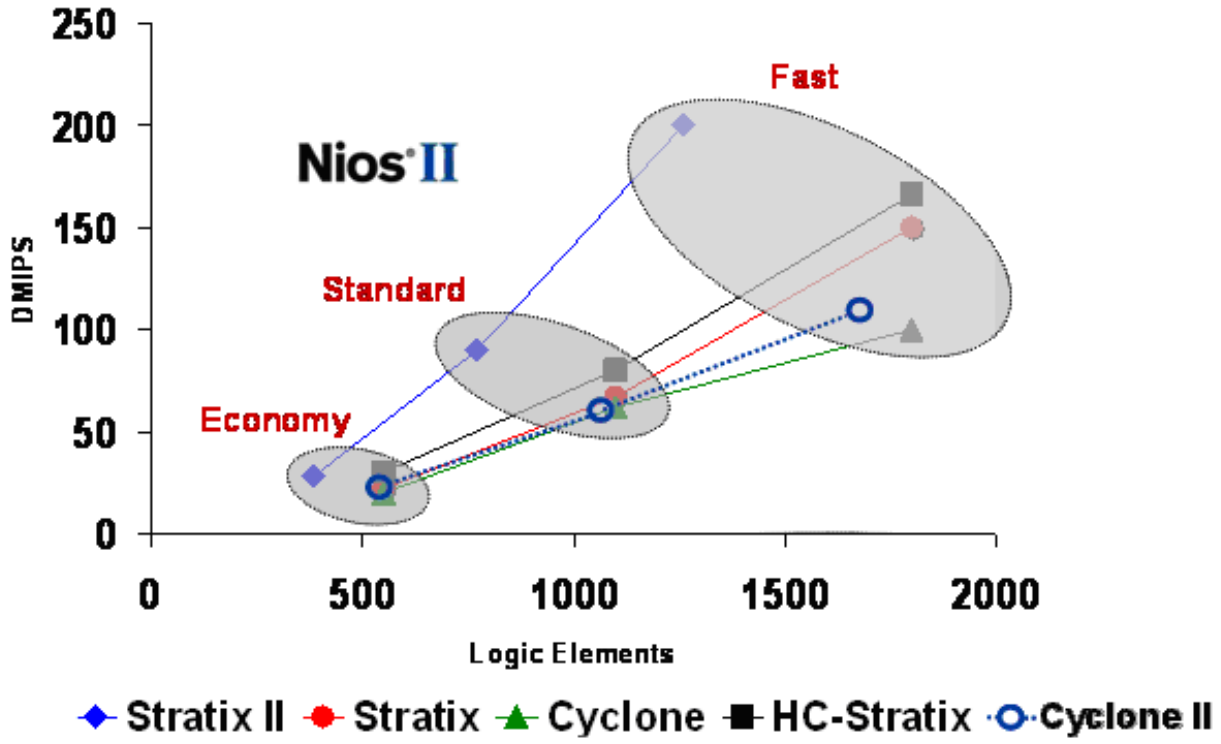


Figure 4: NIOS II processor performances

When you build your SoPC system with the *SOPC Builder* tool, it is possible to include various devices using the Altera Avalon bus:

- Memory.
- Timer.
- UART serial line.
- LCD screen.
- GPIO.
- Ethernet interface.
- CompactFlash interface.
- JTAG.
- …

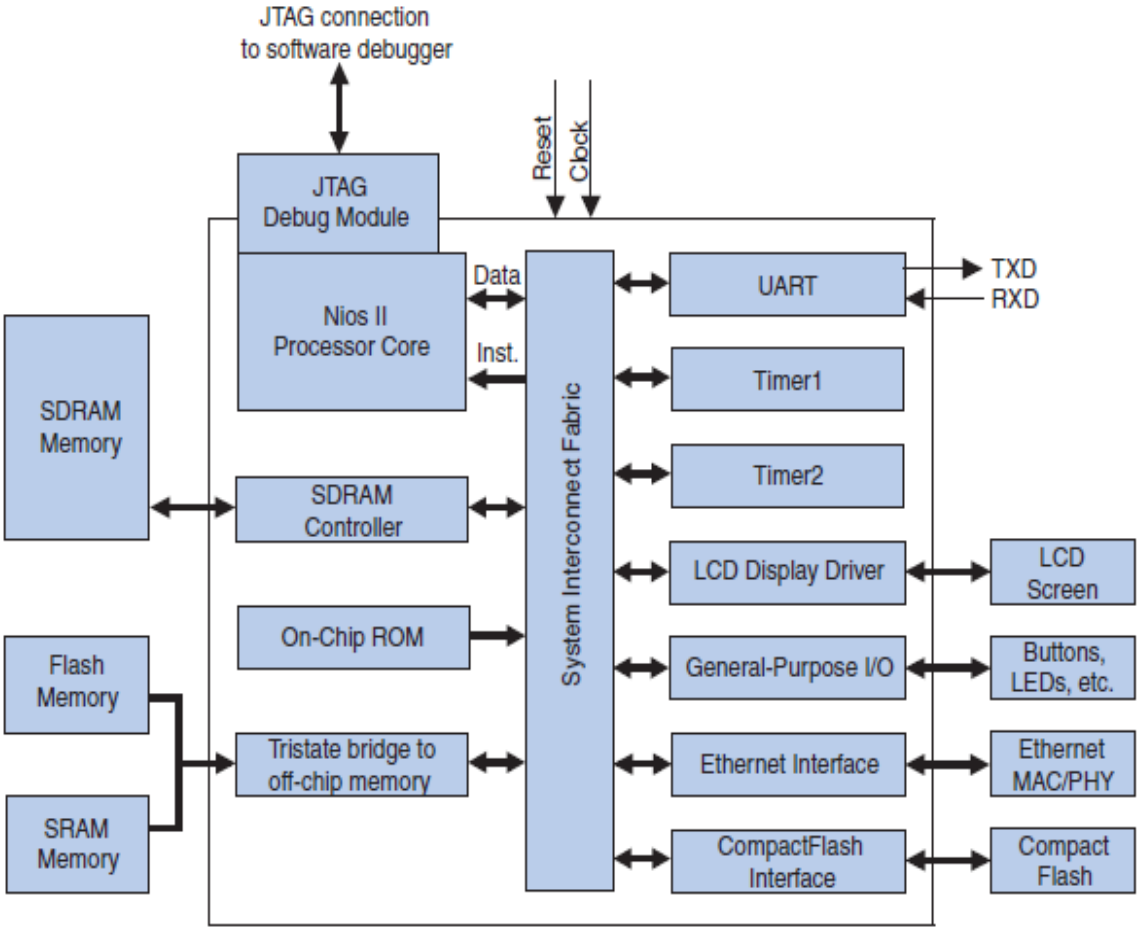


Figure 5: Altera Avalon bus

3. HARDWARE CONFIGURATION: A STEP BY STEP GUIDE

3.1. The target board used as an example

For this tutorial, we have chosen a development board from Altera: the Stratix 1S10 board.

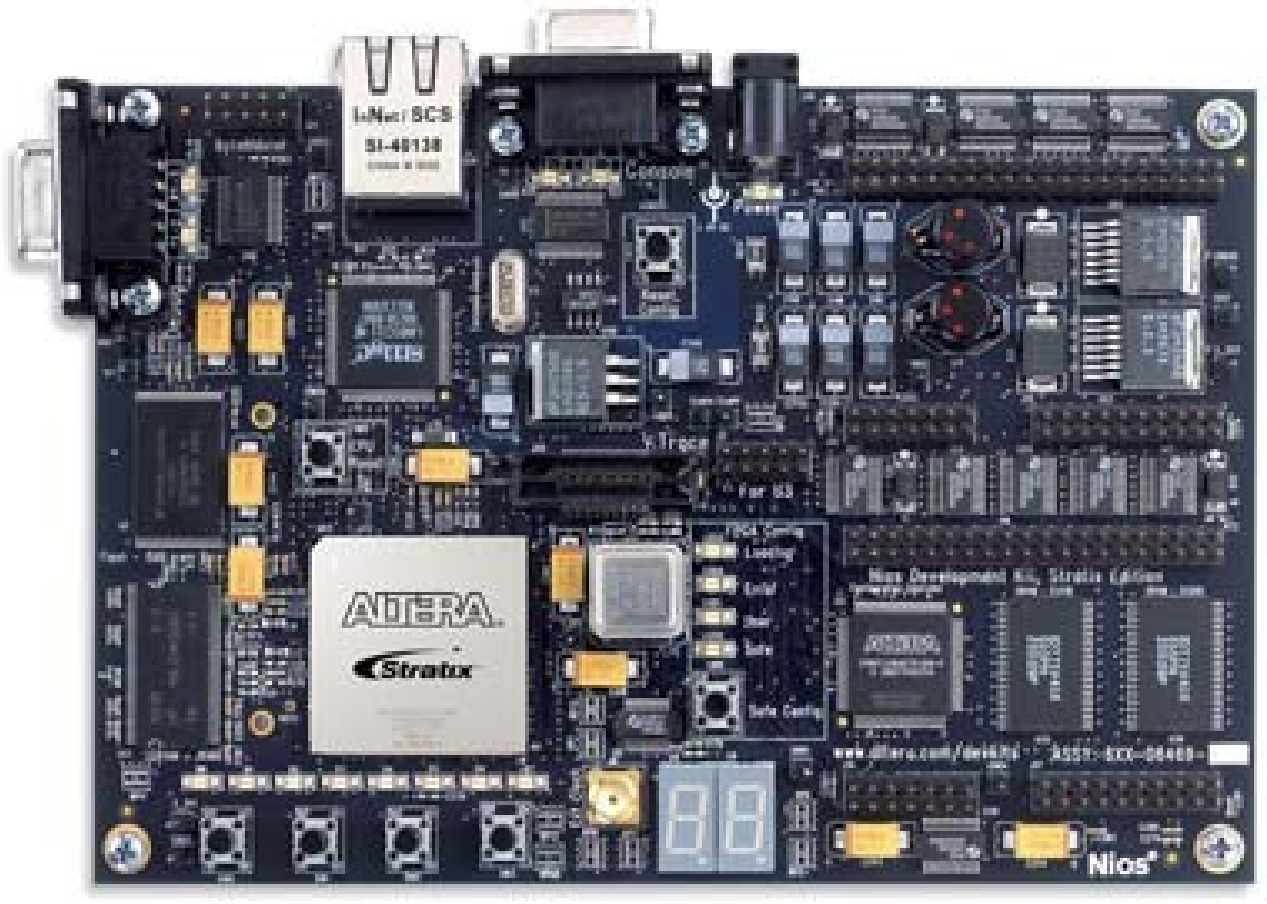


Figure 6: Altera Stratix 1S10 target board

The Stratix 1S10 board has the following features:

- A Stratix EP1S10F780C6 device.
- 8 Mbytes of flash memory.
- 1 Mbyte of static RAM.
- 16 Mbytes of SDRAM.
- On board logic for configuring the Stratix device from flash memory.
- On-board Ethernet MAC/PHY device.
- Two 5-V-tolerant expansion/prototype headers each with access to 41 Stratix user I/O pins.
- CompactFlash connector header for Type I CompactFlash (CF) cards.
- Mictor connector for hardware and software debug.
- Two RS-232 DB9 serial ports.
- Four push-button switches connected to Stratix user I/O pins.
- Eight leds connected to Stratix user I/O pins.

- Dual 7-segment LED display.
- JTAG connectors to Altera devices via Altera download cables.
- 50 MHz oscillator and zero-skew clock distribution circuitry.
- Power-on reset circuitry.

3.2. Building the SoPC system

You have to use the Altera *Quartus II* tool for building the SoPC system compatible with the Xenomai port. The FPGA synthesis process builds a *.sof* file, which will be usable in turn for programming the FPGA circuit on the target board with the JTAG module.

1. Reference design

The *Quartus II* reference design for embedded Linux with or without real-time extension has been built from the *standard* reference design provided by Altera for the Stratix 1S10 board.

This *standard* reference design is generally provided for all Altera boards and is a good start for using Embedded Linux.

The following figure shows the *standard* reference design for the Stratix 1S10 target board:

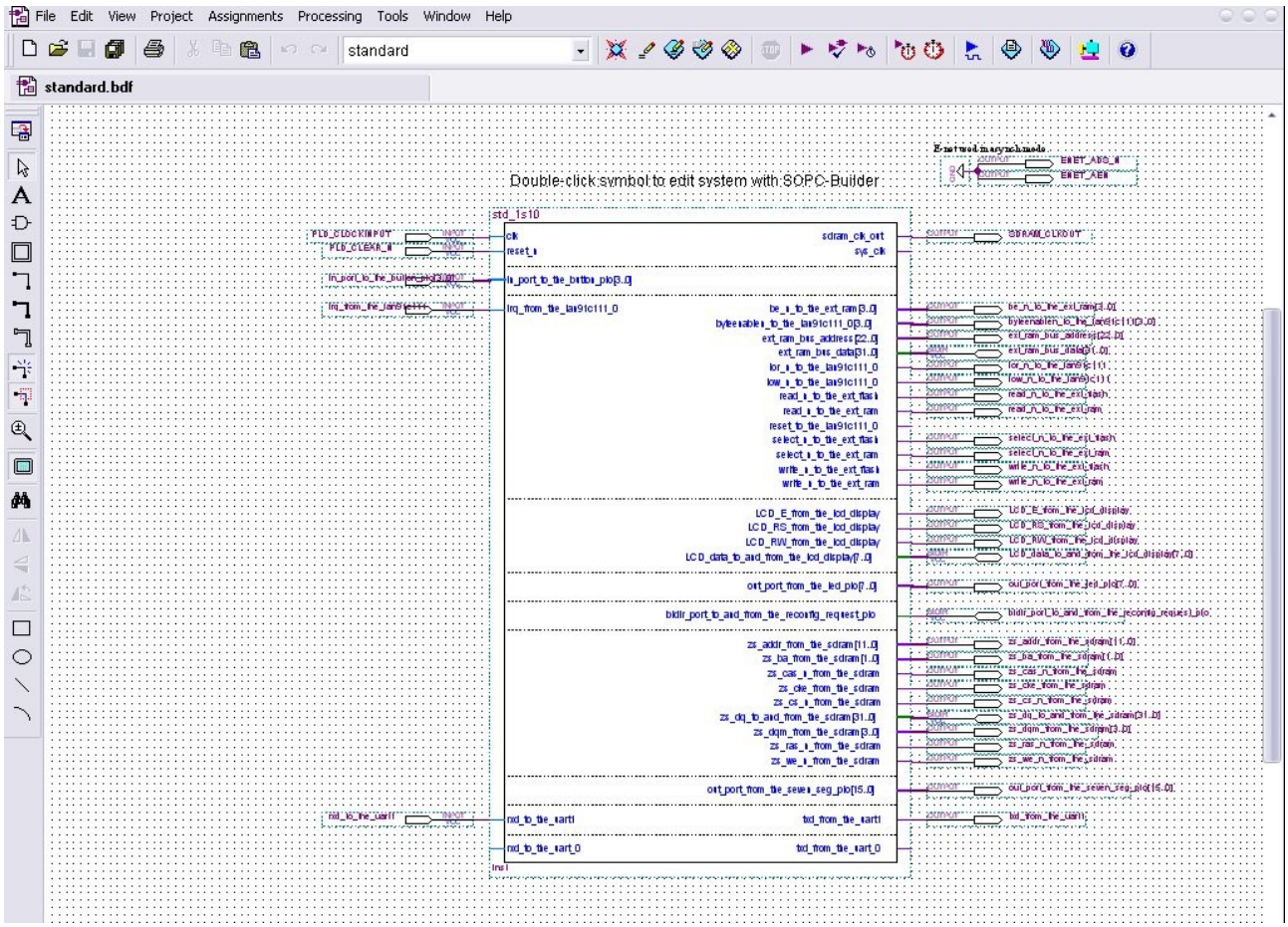
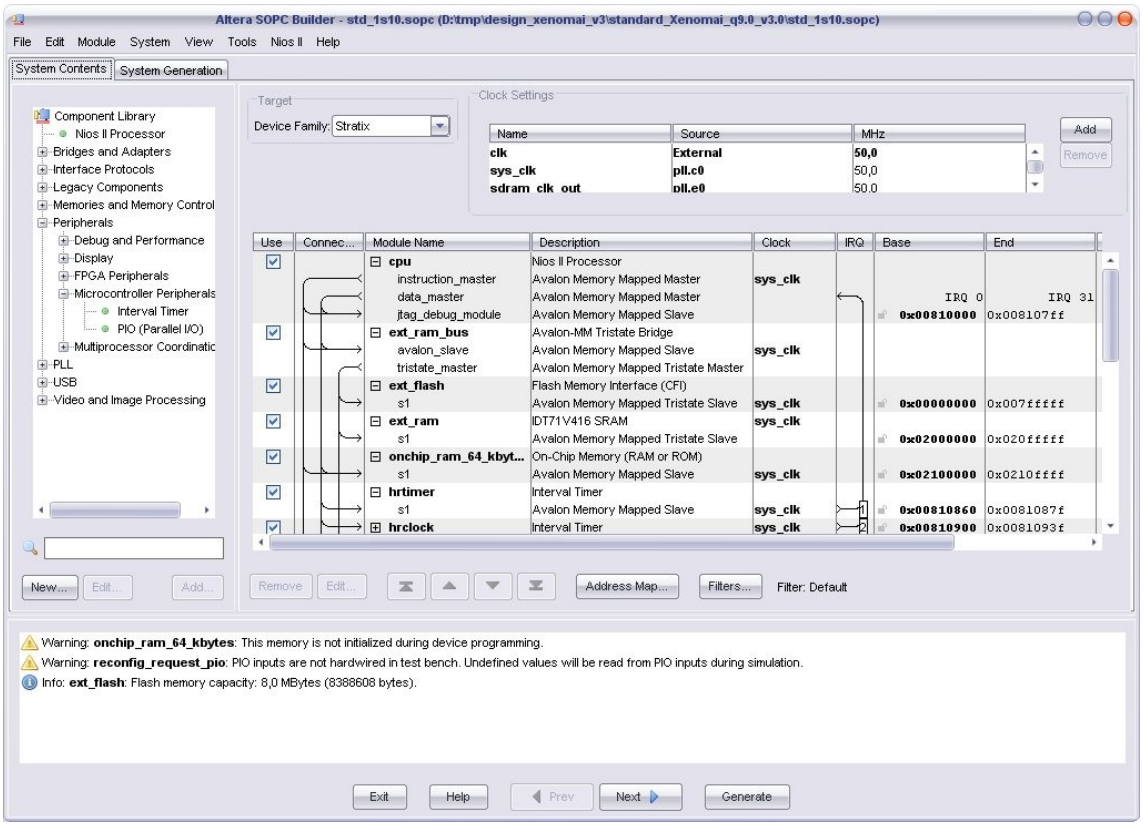
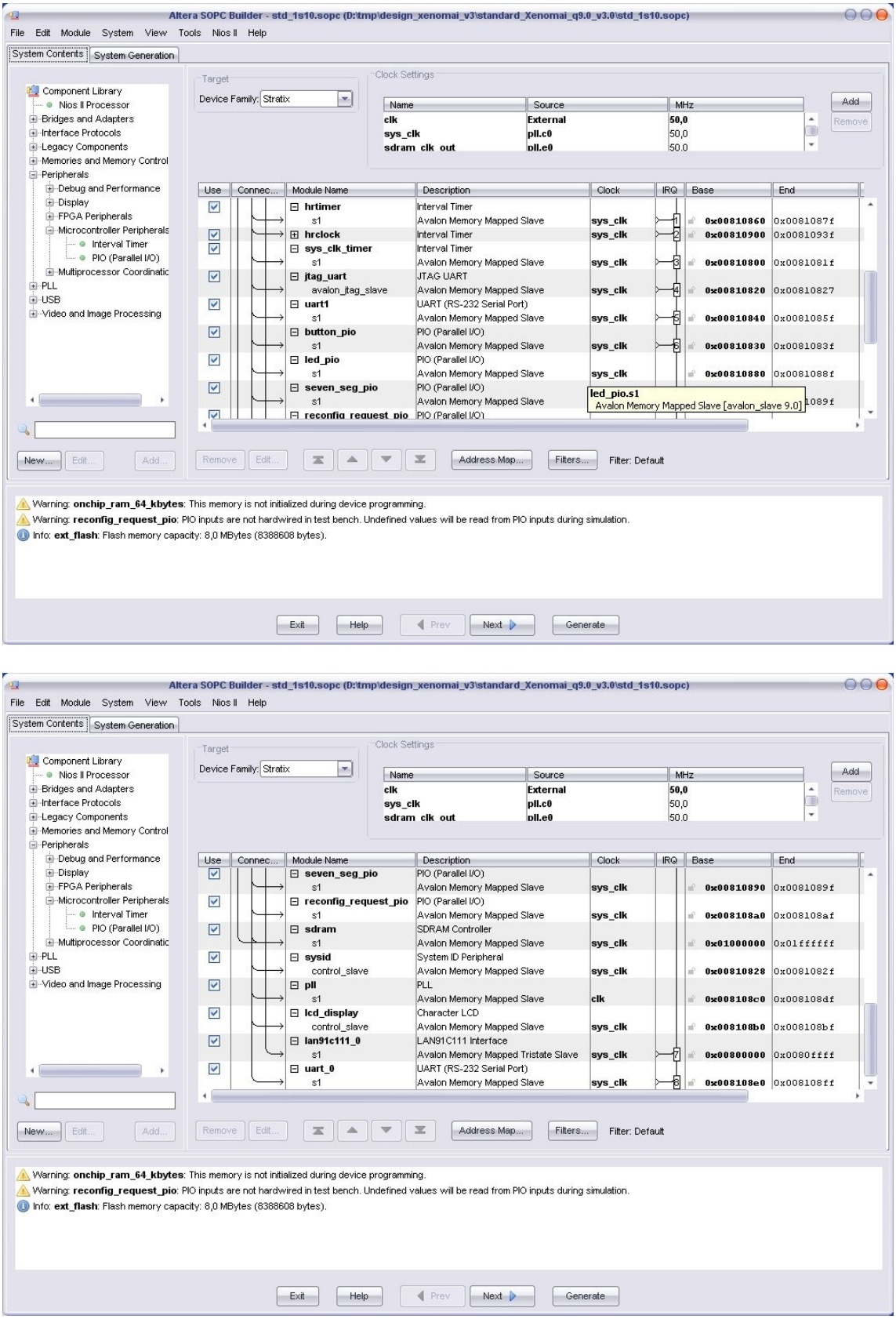


Figure 7: *standard* reference design for the Altera Stratix 1S10 target board

Modify this reference design with the *SoPC Builder* tool.

The three following figures show the new SoPC configuration. New timers have been introduced in the design for compatibility with the Xenomai port for the NIOS II processor.





Figures 8: Creation of new peripherals in the SoPC system

The main points to respect are:

- Peripheral name.
- IRQ attributions.
- Memory mapping for each peripheral.

It is very important to respect these rules in order to use the Xenomai port for NIOS II. All IRQ numbers must be different from 0 (Linux auto detection).

Peripheral Type	Peripheral Name	IRQ Number	Used by
32-bit Timer	*hrtimer*	1 *	Xenomai
64-bit Timer	*hrclock*	2 *	Xenomai
32-bit Timer	*sys_clk_timer*	3 *	Linux

Figure 9: IRQ attribution to the timers in the SoPC system

*: the IRQ number is not important according to the IRQ priority (just one priority in the NIOS II processor).

2. Memory mapping

The SoPC memory mapping must have no memory mapping overlapping.

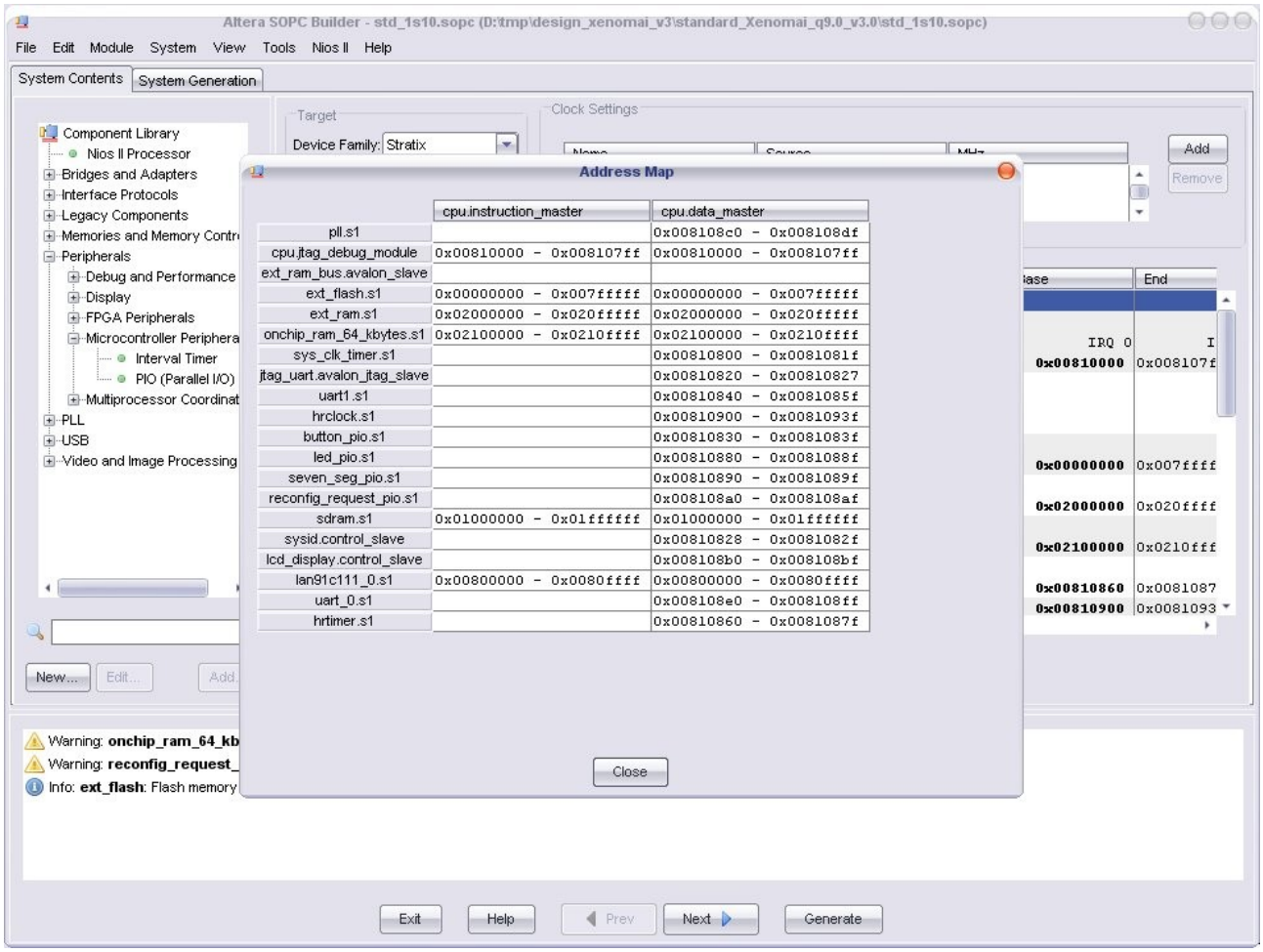


Figure 10: Memory mapping for the SoPC system (example)

3. NIOS II processor configuration

The NIOS II processor is configured at least in its *standard* version:

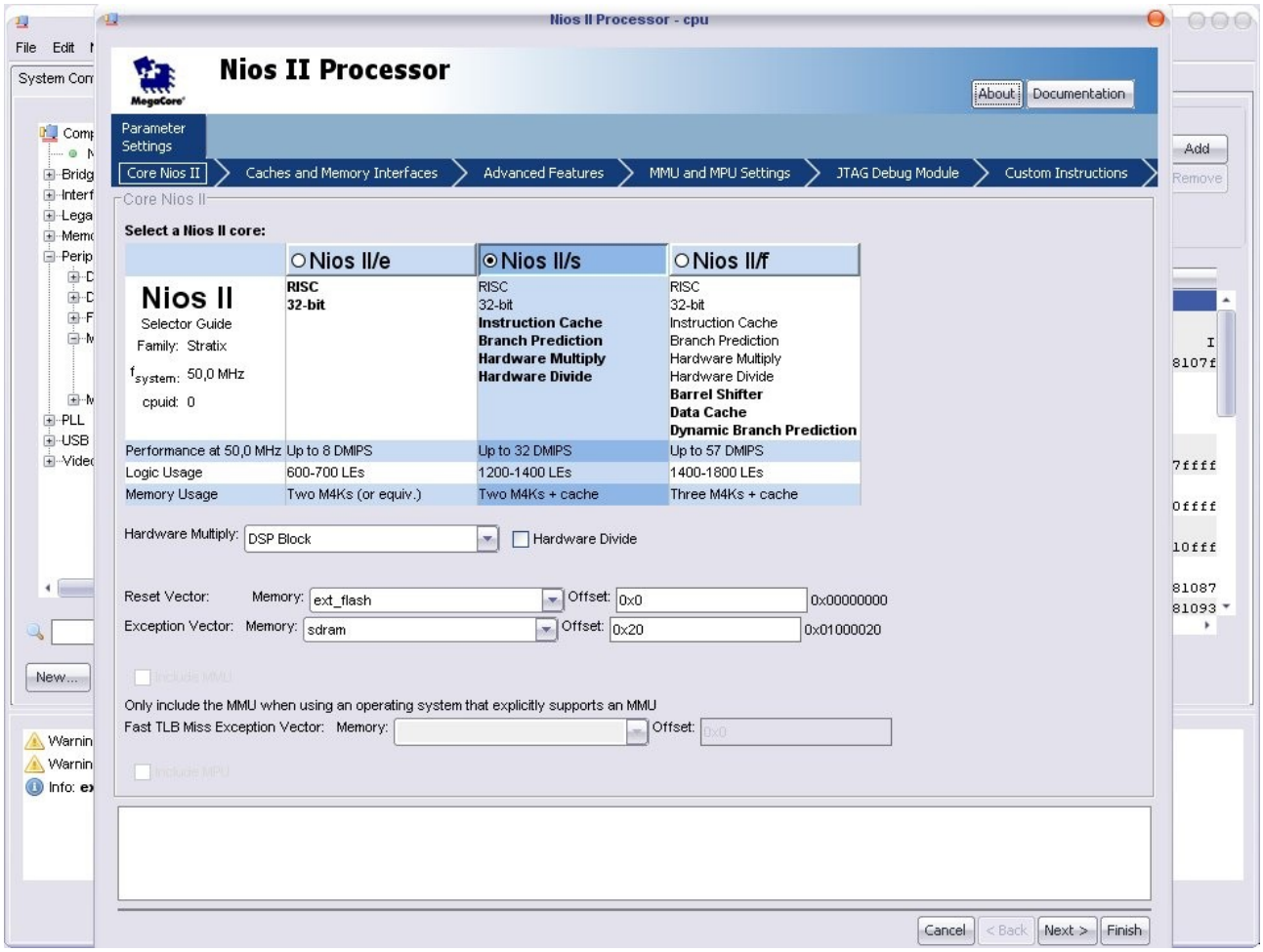


Figure 11: NIOS II processor configuration in its *standard* version

The instruction and data caches are enabled:

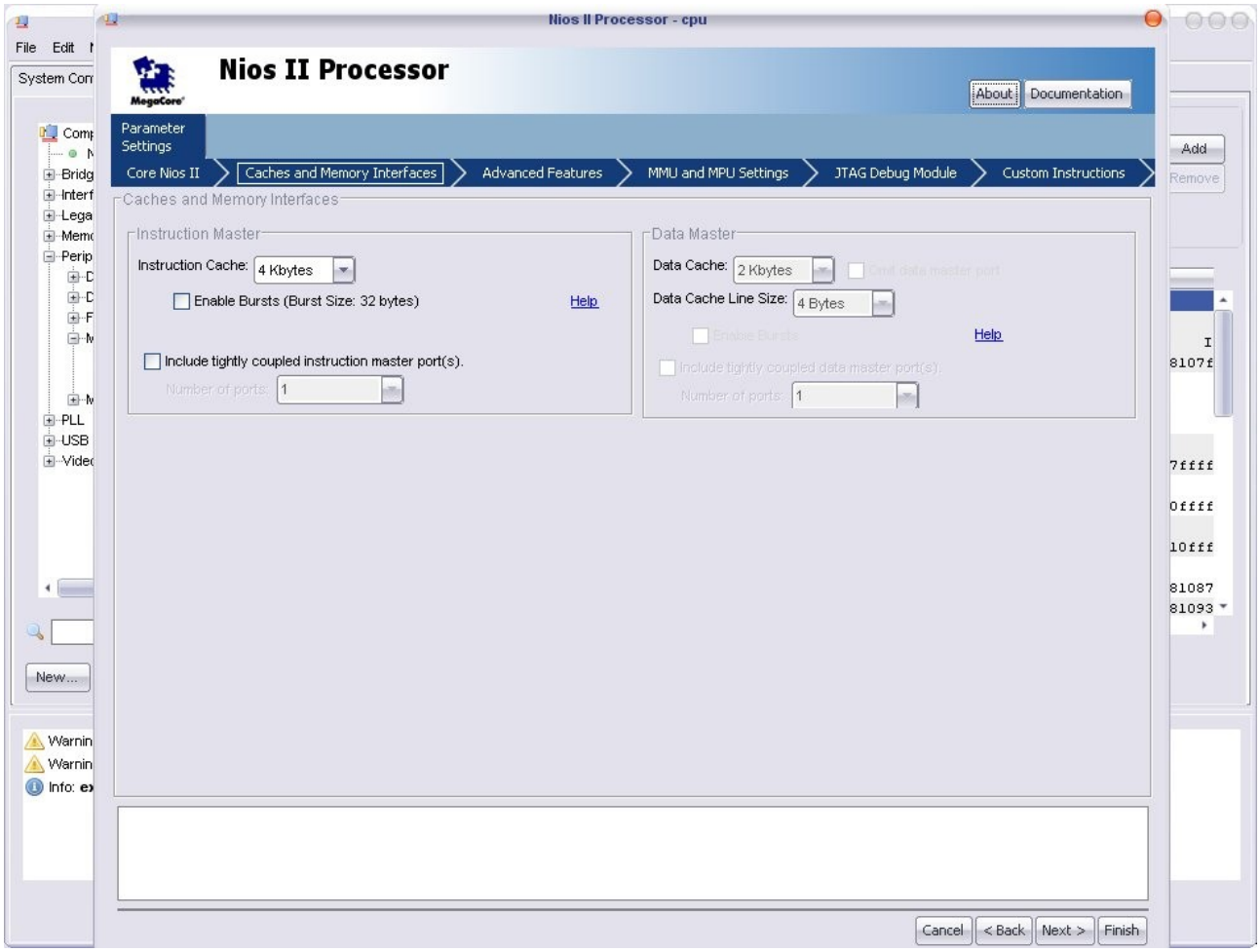
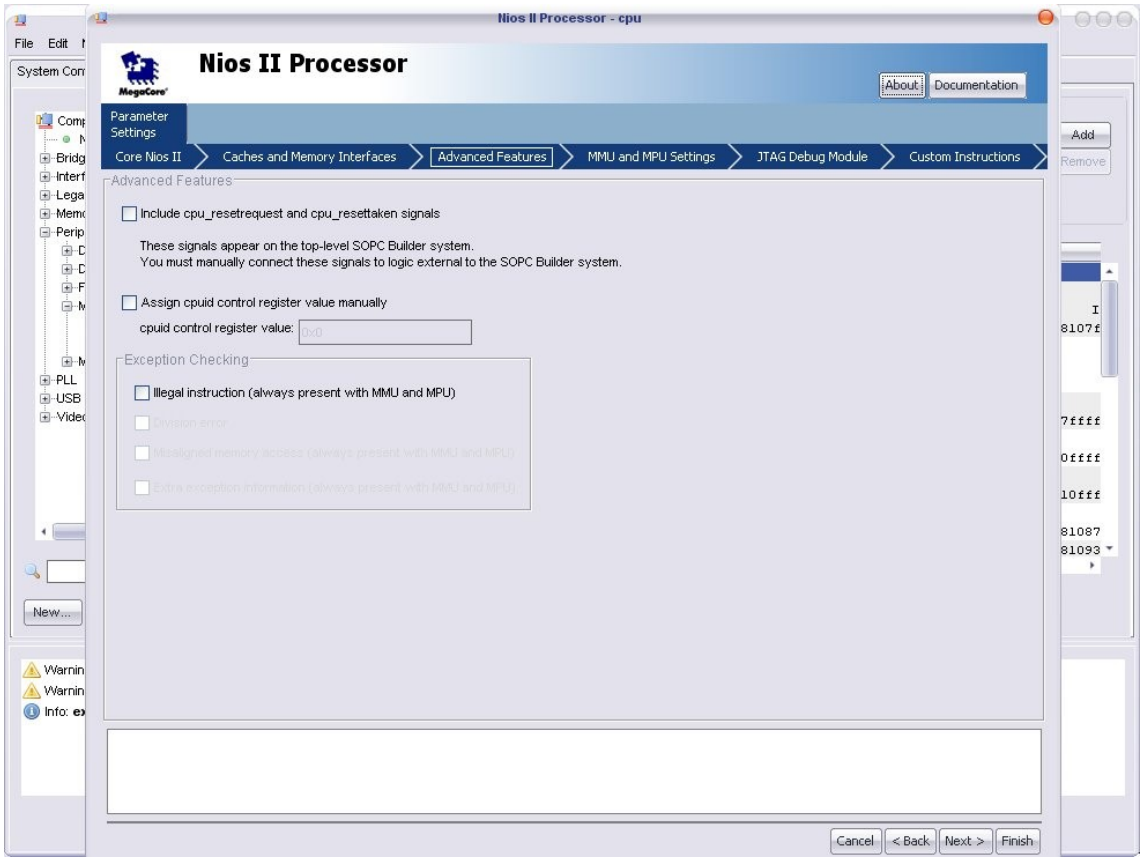
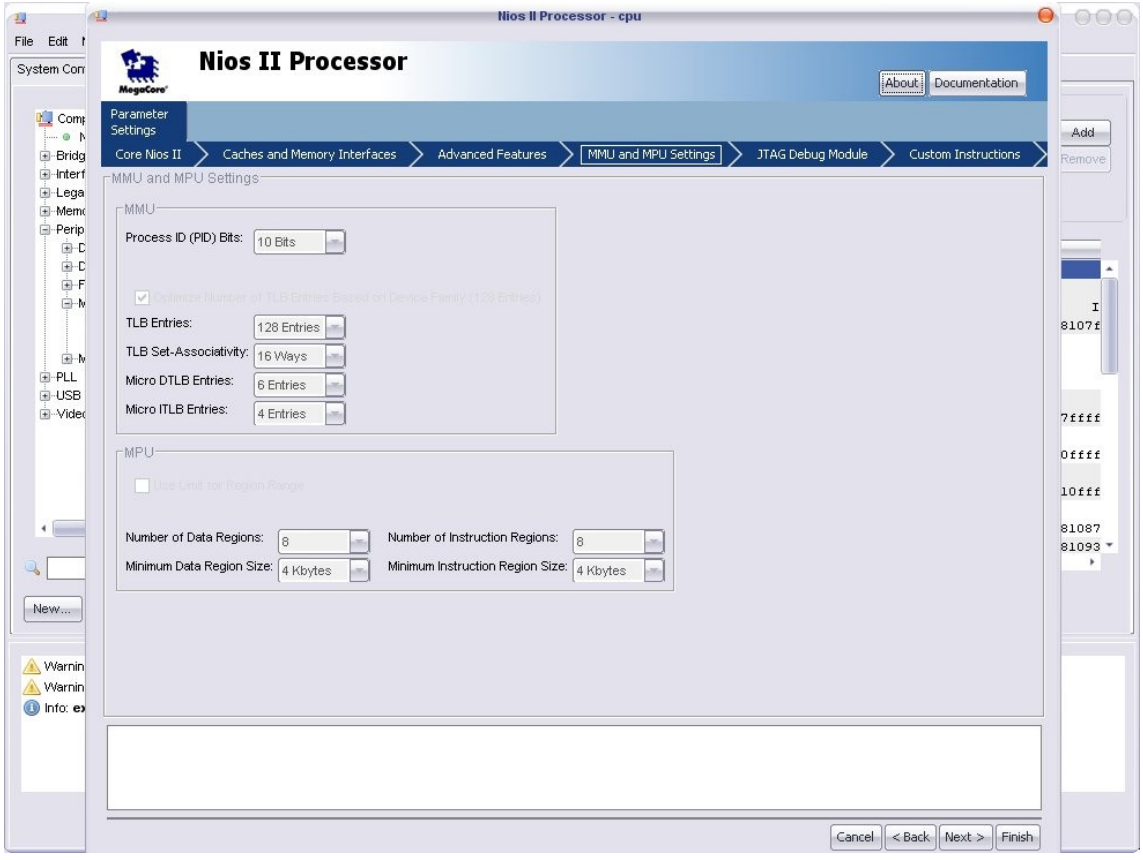


Figure 12: Data and instruction cache configuration

MMU or MPU circuits are disabled (new functionalities appeared with the Quartus II version 8).

In consequence, you have to use the no-MMU Linux version (µClinux for NIOS II) with the NIOS II processor with or without Xenomai enabled…





Figures 13: MMU and MPU circuit configuration

Enable the JTAG (level 1) for debugging:

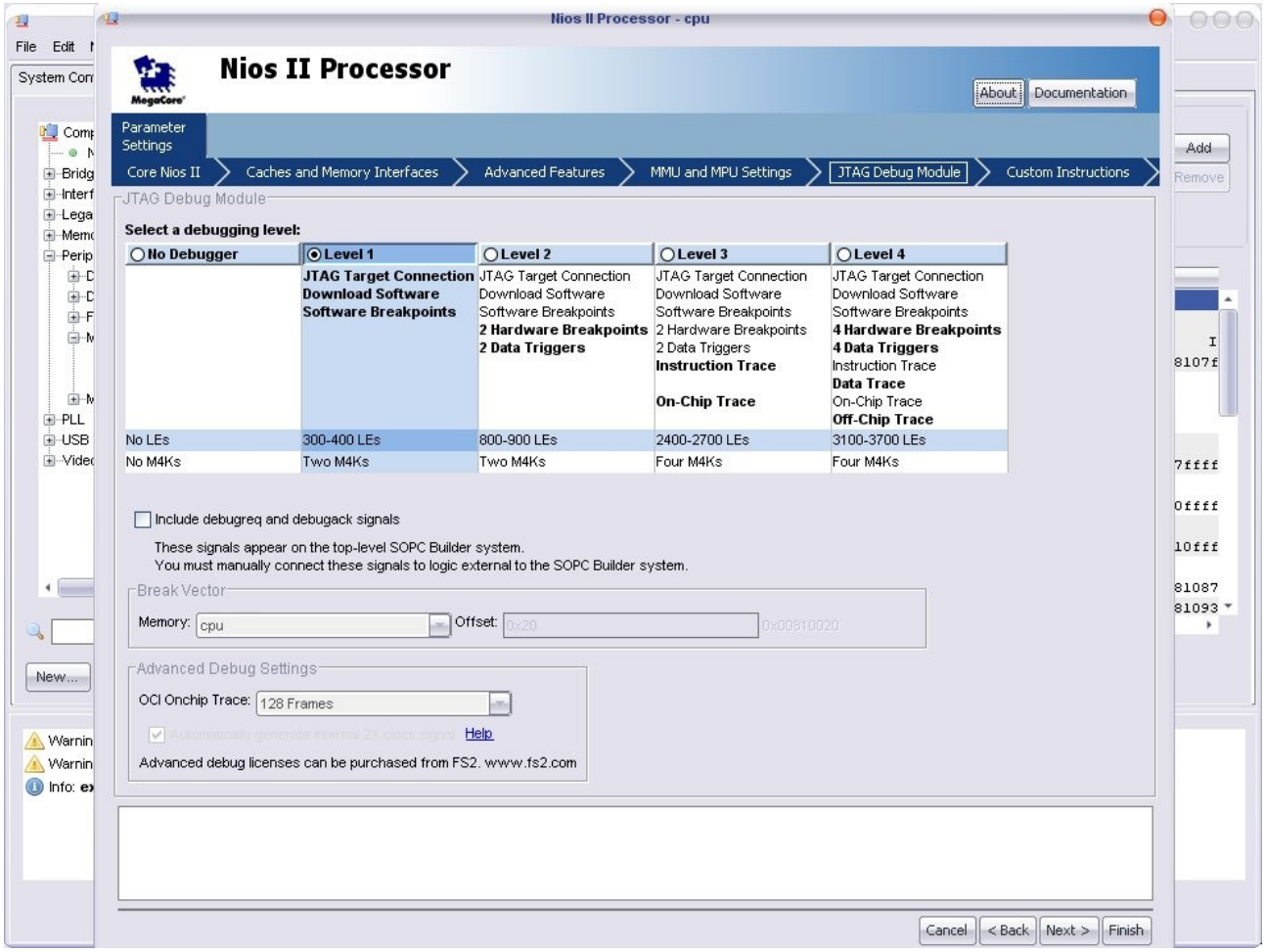


Figure 14: JTAG (*level 1*) configuration

You don’t use the custom instructions:

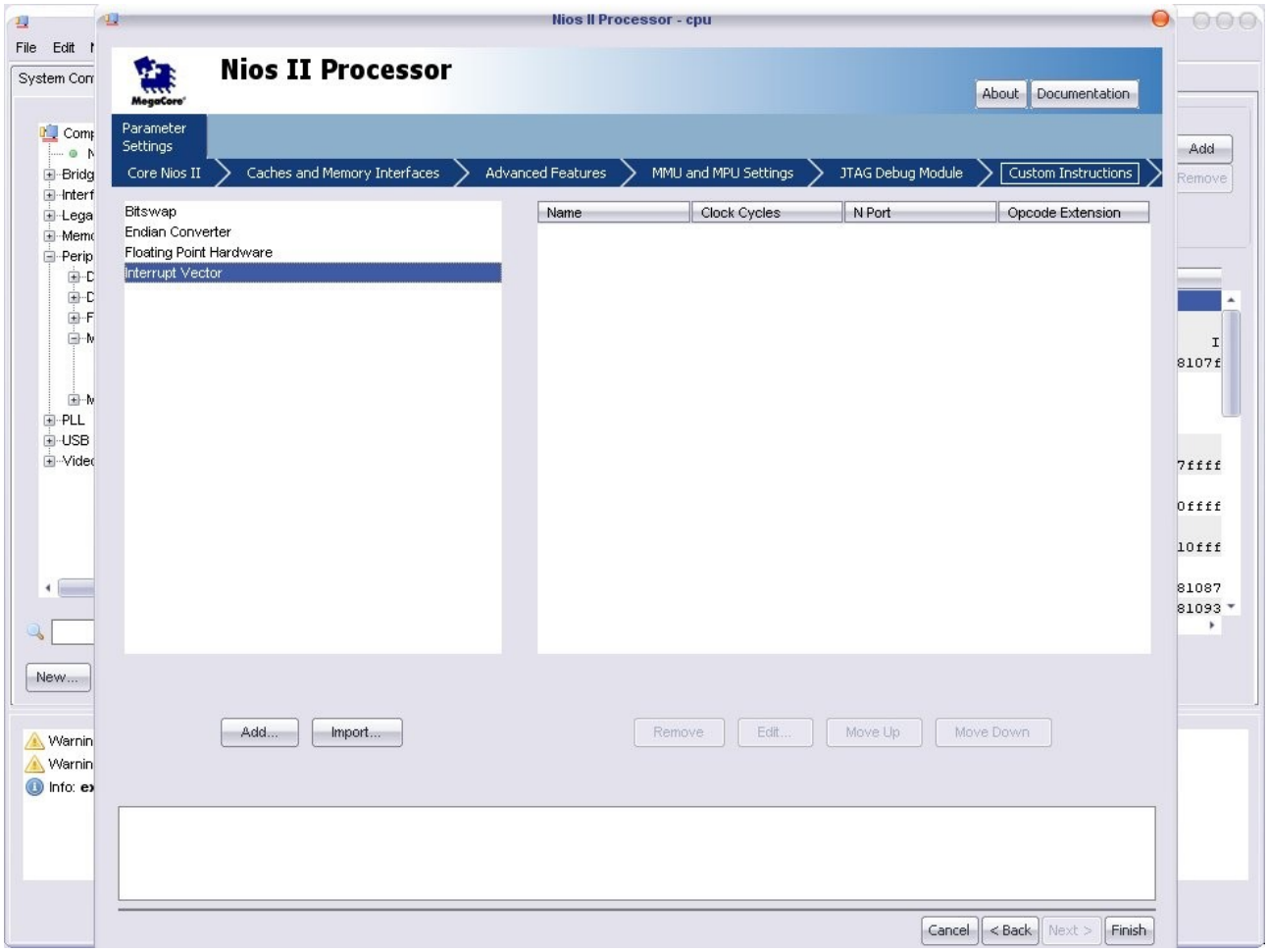


Figure 15: Custom instruction configuration

4. sys_clk_timer timer

IMPORTANT SOPC DESIGN RULE FOR XENOMAI:

When you instantiate your peripherals in your SoPC system with the *SOPC Builder* tool, the peripherals appear in the *.ptf* file in the order they have been created.
When you create the *nios2.h* file from your *.ptf* file (see 4.2), the used parser (Perl script) searches the first Altera timer and uses it as a Linux tick timer.

You must create first the *sys_clk_timer* timer in the SoPC system and then the other timers after.

If you have not done it, erase all Altera timers in your SoPC system (*SOPC Builder*), save and create again first the *sys_clk_timer* timer!

The *sys_clk_timer* timer is used by Linux as a tick timer. Its configuration with the *SoPC Builder* tool is:

- 32-bit timer.
- Timeout period: 10 ms.
- Preset: custom. Writable period, readable snapshot, Start/Stop control bits.

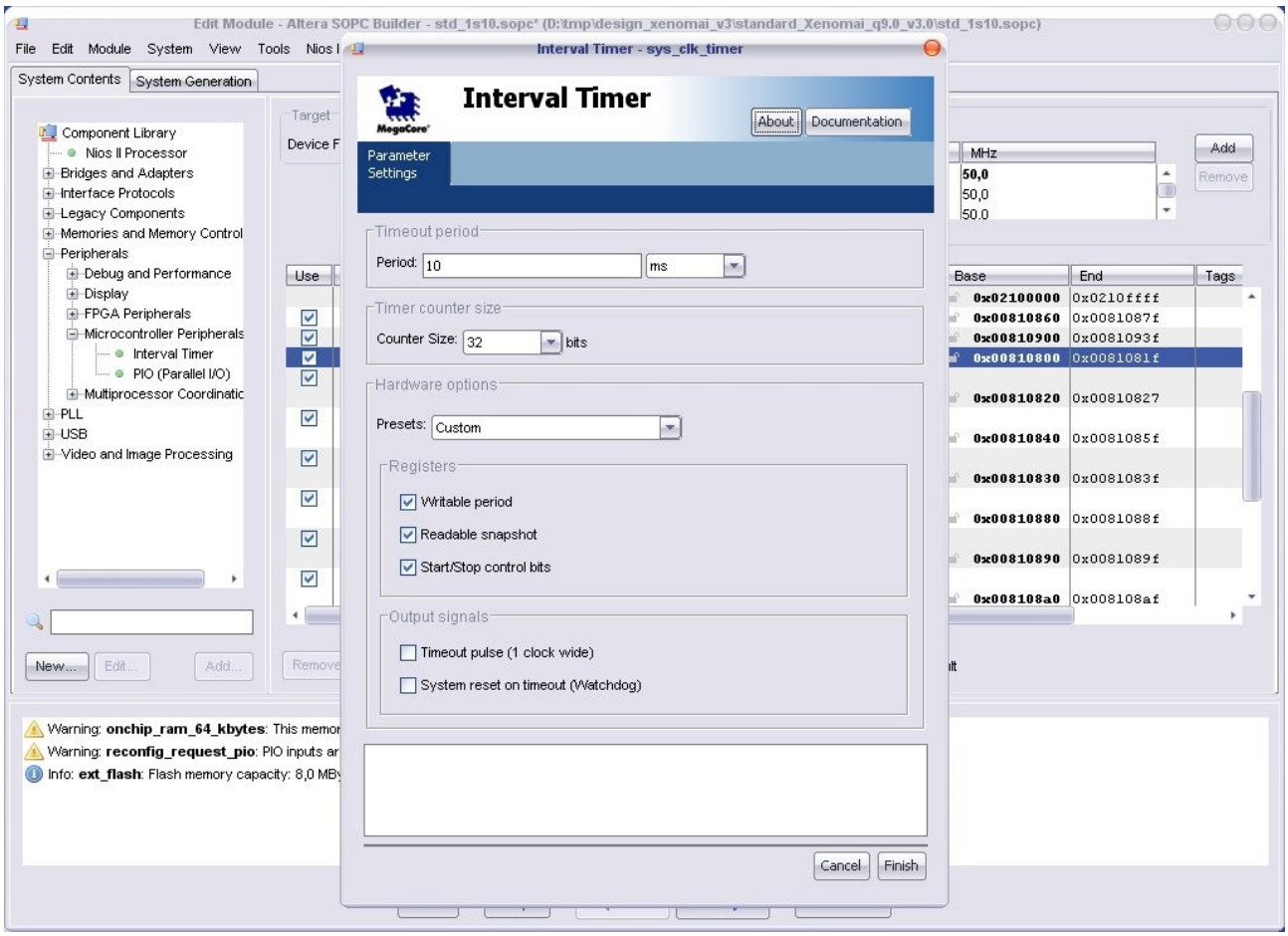


Figure 16: *sys_clk_timer* Linux timer configuration

5. hrtimer timer

The *hrtimer* timer is used by Xenomai as a high-precision clock event source. Its configuration with the *SoPC Builder* tool is:

- Timer: 32 bits.
- Timeout period: 1 µs.
- Preset : custom. Writable period, readable snapshot, Start/Stop control bits.

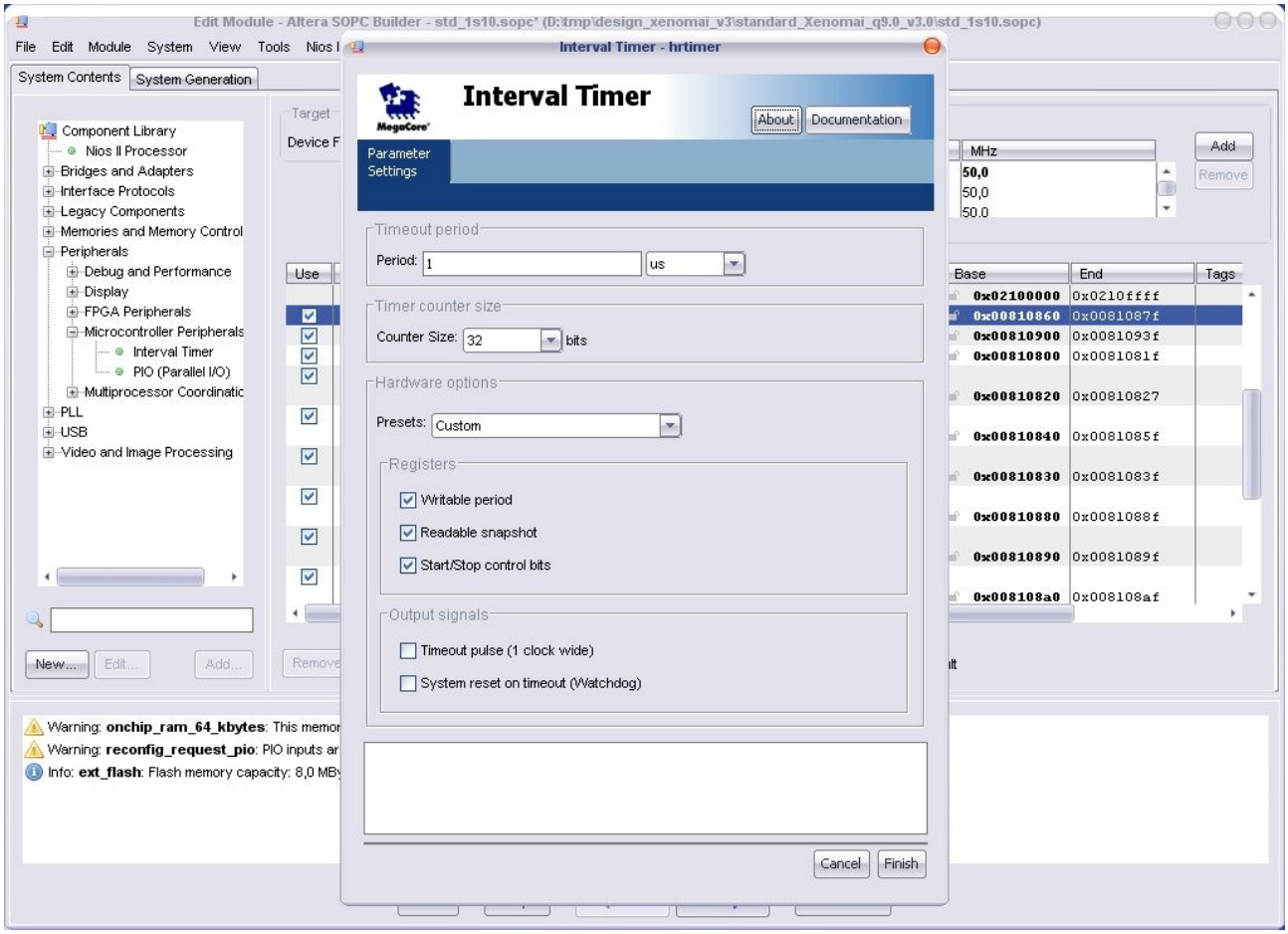


Figure 17: *hrtimer* Xenomai timer configuration

6. hrclock timer

The *hrclock* timer is used by Xenomai as a freerunning counter for precise time stamping purpose, in the snapshot mode. Its configuration with the *SoPC Builder* tool is:

- 64-bit timer.
- Timeout period: 1 µs. The timer functionality is not used by Xenomai.
- Preset: custom. Writable period, readable snapshot, Start/Stop control bits.

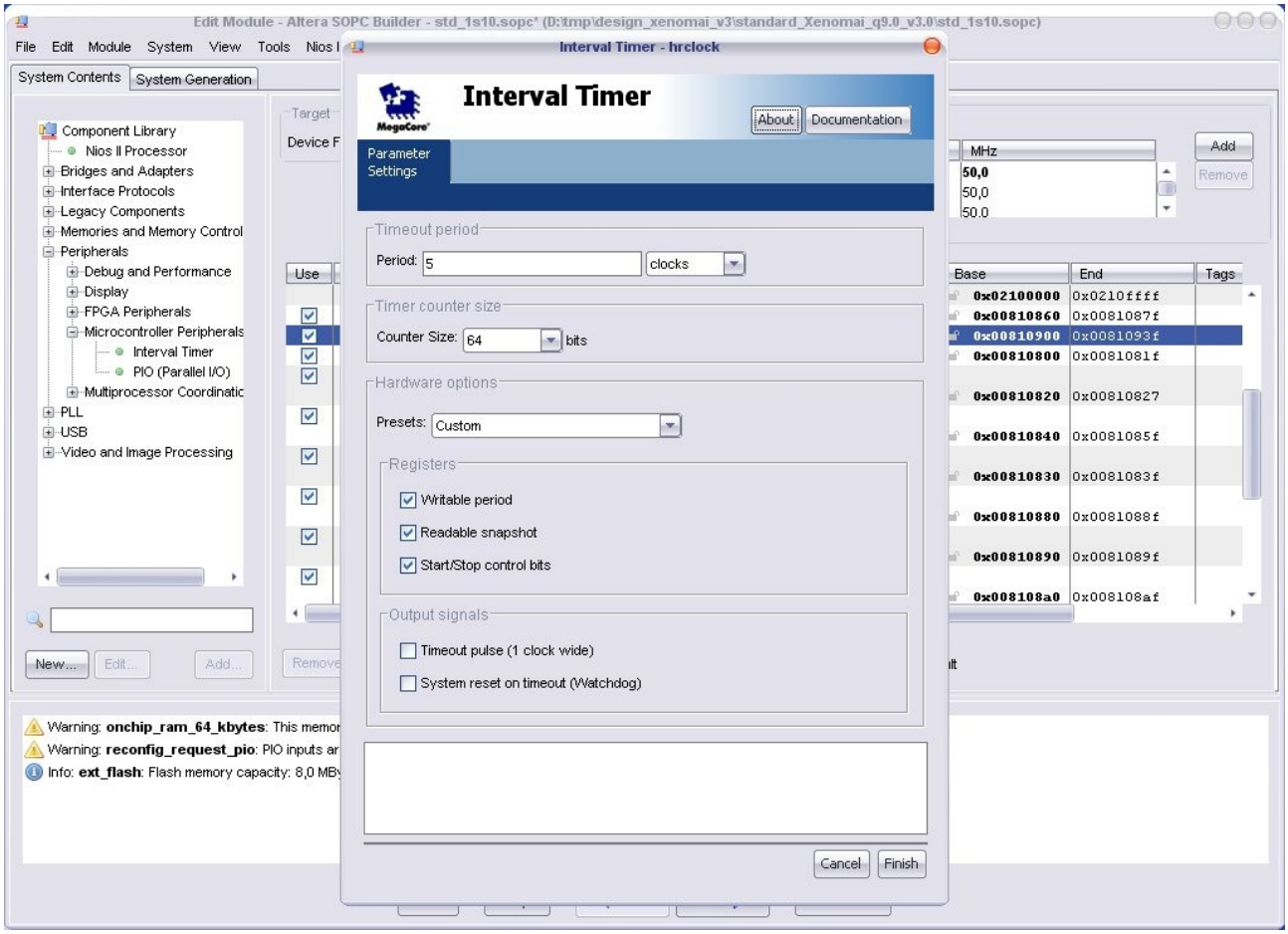


Figure 18: *hrclock* Xenomai timer configuration

4. SOFTWARE CONFIGURATION: A STEP BY STEP GUIDE

4.1. Altera tool installation under Linux

It is possible to use the Altera tools under Linux. In this way, we just need one machine under Linux for both HW and SW development.

The document [5] deeply explains how to do this installation. The installation procedure has been validated under Fedora 10 to Fedora 12.

We have used in this guide the version 9.0 of the *Quartus II* tools:

```
$ cd
$ wget ftp://ftp.altera.com/outgoing/release/90_quartus_linux.tar
$ wget ftp://ftp.altera.com/outgoing/release/90_nios2eds_linux.tar
$ wget ftp://ftp.altera.com/outgoing/release/90_modelsim_ae_linux.tar
$ tar -xvf 90_nios2eds_linux.tar
$ cd nios2eds
# ./install
$ cd
$ tar -xvf 90_modelsim_ae_linux.tar
$ cd modelsim_ae
# ./install
$ cd
$ tar -xvf 90_quartus_linux.tar
$ cd quartus
# ./install
$ cd
```

All the Altera software has been installed under the */opt/altera9.0* directory.

Create the *n2sdk* script that you put in your *~/bin* directory:

```
!/bin/bash
# Run this for a Nios II SDK bash shell
export LM_LICENSE_FILE=1700@localhost
SOPC_KIT_NIOS2=/opt/altera9.0/nios2eds
export SOPC_KIT_NIOS2
SOPC_BUILDER_PATH_90=/opt/altera9.0/nios2eds
export SOPC_BUILDER_PATH_90
unset GCC_EXEC_PREFIX
QUARTUS_ROOTDIR=/opt/altera9.0/quartus
export QUARTUS_ROOTDIR
export PERL5LIB=/usr/lib/perl5/5.10.0
bash --rcfile $QUARTUS_ROOTDIR/sopc_builder/bin/nios_bash
```

For using the Altera *Quartus II* and *ModelSim* tools for synthesis and simulation, you must acquire valid floating licences from Altera that you'll use with a *flexlm* server…

But you can use freely the Altera tools for programming the target board.

Install the JTAG module. In this guide, we use with the target board the USB Blaster module:

```
# mkdir /etc/jtagd
# cp /opt/altera9.0/quartus/linux/pgm_parts.txt /etc/jtagd/jtagd.pgm_parts
$ n2sdk
[NiosII EDS]$ su
Password:
# jtagd
# exit
[NiosII EDS]$ touch ~/.jtag.conf
[NiosII EDS]$ jtagconfig
1) USB-Blaster [USB 2-1.3.2]
  020010DD   EP1S10
```

Verify that the *nios2-download* command is ready for downloading a file into the target board:

```
[NiosII EDS]$ nios2-download
Using cable "USB-Blaster [USB 2-1.3.2]", device 1, instance 0x00
Pausing target processor: OK
Restarting target processor
[NiosII EDS]$
```

The Altera tools are now ready under Linux.

4.2. Linux configuration for the NIOS II processor

You have to use the NIOS processor without MMU or MPU, running a µClinux distribution.

Download the µClinux distribution for NIOS II processor [4]:

```
$ cd
$ wget http://www.niosftp.com/pub/uclinux/nios2-linux-20090730.tar
```

Install the archive file:

```
$ tar -xvf nios2-linux-20090730.tar
```

All the software is under the *nios2-linux* directory:

```
$ cd nios2-linux
```

Run the *checkout* script for recovering all source files and you use after the *update* script for updating:

```
$ ./checkout
$ ./update
```

Checkout the kernel baseline from the linux-2.6 git repository, according to the commit number mentioned for the target release, in the ksrc/nios2/patches/README file:

```
$ cd
$ cd nios2-linux/linux-2.6
$ git checkout -b ipipe d01303a1035a39e445007c7522d89ad985c4153c
```

Install the *gcc* crosscompiler for the NIOS II processor:

```
$ wget http://www.niosftp.com/pub/gnutools/nios2gcc-20080203.tar.bz2
# tar jxf nios2gcc-20080203.tar.bz2 -C /
```

Adjust your PATH variable in your *profile* file *(~/.bash_profile*):

```
PATH=/opt/nios2/bin:$PATH
export PATH
```

Verify that the *gcc* crosscompiler for NIOS II is ready:

```
$ nios2-linux-gcc -v
Reading specs from /opt/nios2/lib/gcc/nios2-linux-uclibc/3.4.6/specs
Configured with: /root/buildroot/toolchain_build_nios2/gcc-
3.4.6/configure --prefix=/opt/nios2 --build=i386-pc-linux-gnu
--host=i386-pc-linux-gnu --target=nios2-linux-uclibc --enable-
languages=c,c++ --disable-__cxa_atexit --enable-target-optspace
--with-gnu-ld --disable-shared --disable-nls --enable-threads
--enable-multilib
Thread model: posix
gcc version 3.4.6
```

We suppose now that *$uClinux_dist* variable is the directory containing the µClinux source files, in our case *nios2-linux/uClinux-dis*t directory.
Configure µClinux for the NIOS II processor:

```
$ cd $uClinux_dist
$ make menuconfig
```

In the *menuconfig* screens, verify that the following options are enabled:

```
Vendor/Product Selection --->
    --- Select the Vendor you wish to target
        Vendor (Altera)  --->
    --- Select the Product you wish to target
        Altera Products (nios2)   --->

Kernel/Library/Defaults Selection --->
    --- Kernel is linux-2.6.x
        Libc Version (None)  --->
    [*] Default all settings (lose changes)
    [ ] Customize Kernel Settings
    [ ] Customize Vendor/User Settings
    [ ] Update Default Vendor Settings
```

Create now the *nios2.h* file that is the link between the HW SoPC system and the µClinux software via the *.ptf* file of your design:

```
$ make vendor_hwselect SYSPTF=/path_to_your_design/your_design.ptf
```

For example here:

```
$ make vendor_hwselect SYSPTF=~/design_xenomai/std_1s10.ptf
```

```
RUNNING hwselect
--- Please select which CPU you wish to build the kernel against:
(1) cpu - Class: altera_nios2 Type: s Version: 7.080900

Selection: 1

--- Please select a device to execute kernel from:

(1) ext_flash
     Class: altera_avalon_cfi_flash
     Size: 8388608 bytes

(2) onchip_ram_64_kbytes
     Class: altera_avalon_onchip_memory2
     Size: 65536 bytes

(3) ext_ram
     Class: altera_nios_dev_kit_stratix_edition_sram2
     Size: 1048576 bytes

(4) sdram
     Class: altera_avalon_new_sdram_controller
     Size: 16777216 bytes

Selection: 4
--- Summary using
PTF: /home/kadionik/design_xenomai/std_1s10.ptf
CPU:                              cpu
Program memory to execute from: sdram
```

You may configure µClinux for using the UART serial line as a Linux console instead of the JTAG UART emulation:

```
$ make menuconfig
Menu Device Driver>Character devices>Serial drivers
    [ ] 8250/16550 and compatible serial support
        *** Non-8250 serial port support ***
    [ ] Altera JTAG UART support
    [*] Altera UART support
    (4)   Maximum number of Altera UART ports
    (115200) Default baudrate for Altera UART ports
    [*]   Altera UART console support
```

Compile now the Linux kernel:

```
$ make
```

You can download the *zImage* file into the target board with the JTAG module:

```
$ n2sdk
[NiosII EDS]$ nios2-download -g images/zImage
```

Use the *minicom* tool for having access to the target board serial line:
You can see the Linux traces:

```
$ minicom
Uncompressing Linux... Ok, booting the kernel.
```

```
Linux version 2.6.30 (kadionik@linux01) (gcc version 3.4.6) #2 PREEMPT Fri Jan 0

uClinux/Nios II
Built 1 zonelists in Zone order, mobility grouping off.  Total pages: 4064
Kernel command line:
NR_IRQS:32
PID hash table entries: 64 (order: 6, 256 bytes)
Dentry cache hash table entries: 2048 (order: 1, 8192 bytes)
Inode-cache hash table entries: 1024 (order: 0, 4096 bytes)
Memory available: 13604k/2492k RAM, 0k/0k ROM (1667k kernel code, 824k data)
Calibrating delay loop... 24.26 BogoMIPS (lpj=121344)
Mount-cache hash table entries: 512
net_namespace: 264 bytes
NET: Registered protocol family 16
init_BSP(): registering device resources
bio: create slab <bio-0> at 0
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 512 (order: 0, 4096 bytes)
TCP bind hash table entries: 512 (order: -1, 2048 bytes)
TCP: Hash tables configured (established 512 bind 512)
TCP reno registered
NET: Registered protocol family 1
io scheduler noop registered
io scheduler deadline registered (default)
ttyS0 at MMIO 0x810840 (irq = 5) is a Altera UART
console [ttyS0] enabled
ttyS1 at MMIO 0x8108e0 (irq = 8) is a Altera UART
smc91x.c: v1.1, sep 22 2004 by Nicolas Pitre <nico@cam.org>
eth0: SMC91C11xFD (rev 1) at 80800300 IRQ 7 [nowait]
eth0: Invalid ethernet MAC address.  Please set using ifconfig
dm9000 Ethernet Driver, V1.31
TCP cubic registered
NET: Registered protocol family 17
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
Freeing unused kernel memory: 596k freed (0x11da000 - 0x126e000)
Shell invoked to run file: /etc/rc
Command: hostname uClinux
Command: mount -t proc proc /proc -o noexec,nosuid,nodev
Command: mount -t sysfs sysfs /sys -o noexec,nosuid,nodev
Command: mount -t devpts devpts /dev/pts -o noexec,nosuid
Command: mount -t usbfs none /proc/bus/usb
mount: mounting none on /proc/bus/usb failed: No such file or directory
Command: mkdir /var/tmp
Command: mkdir /var/log
Command: mkdir /var/run
Command: mkdir /var/lock
Command: mkdir /var/empty
Command: ifconfig lo 127.0.0.1
Command: route add -net 127.0.0.0 netmask 255.0.0.0 lo
Command: cat /etc/motd
Welcome to
```

```
          ____ _  _
         /  __| ||_|
  _   _  | |  | | _ ____  _   _  _ _  _
 | | | | | |  | || |  _ \| | | |\ \/ /
 | |_| | | |__| || | | | | |_| |/    \
 |  ___\____|_||_|_| |_|\____|\_/\_/
 | |
 |_|
```

```
For further information check:
http://www.uclinux.org/
```

```
Execution Finished, Exiting

Sash command shell (version 1.1.1)
/>
/>
```

4.3. Xenomai configuration for the NIOS II processor

Download the latest Xenomai version from the *git* server:

```
$ cd
$ git clone git://xenomai.org/xenomai-head.git
```

For the latest stable version (2.5.2) :

```
$ wget http://download.gna.org/xenomai/stable/xenomai-2.5.2.tar.bz2
$ tar -xvjf xenomai-2.5.2.tar.bz2
```

We suppose now that the *$xenomai_root* variable is the directory containing the Xenomai source files.

Apply the *ipipe* patch for the NIOS II processor on the Linux kernel source files:

```
$ cd
$ cd $xenomai_root
$ export linux_tree=$HOME/nios2-linux/linux-2.6
$ ./scripts/prepare-kernel.sh --arch=nios2 \
        --adeos=$xenomai_root/ksrc/arch/nios2/patches/adeos-ipipe-2.6.30-nios2-* \
  --linux=$linux_tree
```

Compile the Xenomai utilities in order to integrate them into the *romfs* directory:

```
$ ./configure –host=nios2-linux
$ make install DESTDIR=$uClinux_dist/romfs
```

You may suppress all the Xenomai documentation in the *romfs* directory:

```
$\rm -rf $uClinux_dist/romfs/usr/xenomai/share/doc
$\rm -rf $uClinux_dist/romfs/usr/xenomai/share/man
```

Recompile the Linux kernel for including Xenomai support:

```
$ cd $uClinux_dist
$ make
```

Download the *zImage* file into the target board with the JTAG module:

```
$ n2sdk
[NiosII EDS]$ nios2-download -g images/zImage
```

Use the *minicom* tool for having access to the target board serial line:
You can see the Linux traces:

```
$ minicom
Uncompressing Linux... Ok, booting the kernel.
Linux version 2.6.30 (kadionik@linux01) (gcc version 3.4.6) #18 PREEMPT Tue Mar0


uClinux/Nios II
```

```
Built 1 zonelists in Zone order, mobility grouping off.  Total pages: 4064
Kernel command line:
NR_IRQS:32
PID hash table entries: 64 (order: 6, 256 bytes)
I-pipe 1.1-00: pipeline enabled.
Dentry cache hash table entries: 2048 (order: 1, 8192 bytes)
Inode-cache hash table entries: 1024 (order: 0, 4096 bytes)
Memory available: 11572k/4525k RAM, 0k/0k ROM (1931k kernel code, 2593k data)
Calibrating delay loop... 24.06 BogoMIPS (lpj=120320)
Mount-cache hash table entries: 512
net_namespace: 264 bytes
NET: Registered protocol family 16
init_BSP(): registering device resources
bio: create slab <bio-0> at 0
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 512 (order: 0, 4096 bytes)
TCP bind hash table entries: 512 (order: -1, 2048 bytes)
TCP: Hash tables configured (established 512 bind 512)
TCP reno registered
NET: Registered protocol family 1
I-pipe: Domain Xenomai registered.
Xenomai: hal/nios2 started.
Xenomai: scheduling class idle registered.
Xenomai: scheduling class rt registered.
Xenomai: real-time nucleus v2.5.2 (Souls Of Distortion) loaded.
Xenomai: starting native API services.
Xenomai: starting POSIX services.
Xenomai: starting RTDM services.
io scheduler noop registered
io scheduler deadline registered (default)
ttyS0 at MMIO 0x810840 (irq = 5) is a Altera UART
console [ttyS0] enabled
ttyS1 at MMIO 0x8108e0 (irq = 8) is a Altera UART
smc91x.c: v1.1, sep 22 2004 by Nicolas Pitre <nico@cam.org>
eth0: SMC91C11xFD (rev 1) at 80800300 IRQ 7 [nowait]
eth0: Invalid ethernet MAC address.  Please set using ifconfig
dm9000 Ethernet Driver, V1.31
TCP cubic registered
NET: Registered protocol family 17
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
Freeing unused kernel memory: 2276k freed (0x1232000 - 0x146a000)
Shell invoked to run file: /etc/rc
Command: hostname uClinux
Command: mount -t proc proc /proc -o noexec,nosuid,nodev
Command: mount -t sysfs sysfs /sys -o noexec,nosuid,nodev
Command: mount -t devpts devpts /dev/pts -o noexec,nosuid
Command: mount -t usbfs none /proc/bus/usb
mount: mounting none on /proc/bus/usb failed: No such file or directory
Command: mkdir /var/tmp
Command: mkdir /var/log
Command: mkdir /var/run
Command: mkdir /var/lock
Command: mkdir /var/empty
Command: ifconfig lo 127.0.0.1
Command: route add -net 127.0.0.0 netmask 255.0.0.0 lo
Command: cat /etc/motd
Welcome to

          ____ _  _
         /  __| ||_|
    _   _| |  | | _ ____  _   _  _  _
   | | | | |  | || |  _ \| | | |\ \/ /
   | |_| | |__| || | | | | |_| |/    \
```

```
   |  ___\____|_||_|_| |_|\____|\_/\_/
   | |
   |_|

For further information check:
http://www.uclinux.org/

Execution Finished, Exiting

Sash command shell (version 1.1.1)
/>
/> cd /usr/xenomai/bin
/usr/xenomai/bin> ./latency -t2
== Sampling period: 10000 us
== Test mode: in-kernel timer handler
== All results in microseconds
warming up...
RTT|  00:00:01  (in-kernel timer handler, 10000 us period, priority 99)
RTH|-----lat min|-----lat avg|-----lat max|-overrun|----lat best|---lat worst
RTD|      73.080|      83.223|      90.020|       0|      73.080|      90.020
RTD|       1.820|      55.164|     104.840|       0|       1.820|     104.840
RTD|       3.240|      65.454|     106.160|       0|       1.820|     106.160
RTD|       0.920|      63.167|     100.800|       0|       0.920|     106.160
RTD|       1.160|      65.735|     103.860|       0|       0.920|     106.160
RTD|      -0.600|      62.741|     105.180|       0|      -0.600|     106.160
RTD|       3.520|      64.406|     105.480|       0|      -0.600|     106.160
RTD|       1.720|      65.379|     105.100|       0|      -0.600|     106.160
RTD|       2.280|      66.060|     106.980|       0|      -0.600|     106.980
RTD|       4.900|      65.314|     106.320|       0|      -0.600|     106.980
```

5. REFERENCES

[1] The Xenomai project. http://www.xenomai.org

[2] The µClinux project for the NIOS II processor. http://www.nioswiki.com/

[3] The QuartusforLinux page.
http://www.nioswiki.com/OperatingSystems/UClinux/QuartusforLinux

[4] The InstallNios2Linux page. http://www.nioswiki.com/InstallNios2Linux

[5] Altera tools under Linux : ftp://ftp.altera.com/outgoing/release