International Master Program
Information & Communication Engineering

TECHNISCHE UNIVERSITÄT DARMSTADT

Reconfigurable Technologies

Dr. Gilles SASSATELLI
sassatelli@lirmm.fr
Gabriel Marchesan ALMEIDA
marchesan@lirmm.fr

Winter Semester 2009

Lab 3.1

Running a RTOS on a MIPS based processor

1. Introduction

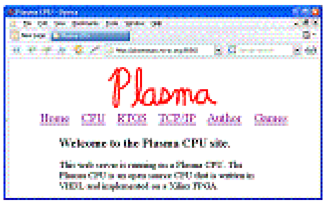
This Lab is devoted to the prototyping of a MIPS based processor called PLASMA using FPGA based boards and run an operating system on it. You will use Spartan 3 boards from Xilinx. Those are populated with a Xilinx Spartan 3 XC3S1000 FPGA and feature a lot of peripherals, from the classical pushbuttons and LEDs to Serial Interface and video. All the necessary information for this board is located on the Xilinx Web site (http://www.xilinx.com). Take a close look at the Spartan 3 starter kit manual (http://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf). The RTOS as well the Plasma processor you will prototype on the FPGA are both available on the OpenCores website (http://www.opencores.org/projects.cgi/web/mips/overview). After we have prototyped the Plasma processor on the board, you will compile the RTOS using a gcc-mips cross-compiler and upload it to the board.

2. Plasma Processor

The Plasma CPU is a small synthesizable 32-bit RISC microprocessor. It is currently running a live web server with an interrupt controller, UART, SRAM or DDR SDRAM controller, and Ethernet controller. The Plasma CPU executes all MIPS I(TM) user mode instructions except unaligned load and store operations [1].

This "clean room" CPU core is implemented in VHDL with either a two or three-stage pipeline. It is running at 25 MHz on a Xilinx FPGA and also verified on an Altera FPGA.

The Plasma CPU along with the Plasma RTOS and TCP/IP protocol stack are now running a live Web Server on a Xilinx FPGA.



Welcome to the Plasma CPU site.

This web server is running on a Plasma CPU. The Plasma CPU is an open source CPU that is written in VHDL and implemented on a Xilinx FPGA.

The Plasma RISC CPU was also successfully used to control four communication robots using Xilinx Virtex FPGAs.

3. Tools

1- Xilinx ISE (to synthesize the Plasma Processor) [2]
2- Impact (to program the FPGA with either PROM File (.mcs) or Bitstream (.bit)) [2]
3- Serial Terminal (to send/receive data to/from serial interface) [3]
4- Gcc-mips cross-compiler (to compile C ANSI applications for the MIPS Processor)

4. Lab exercises

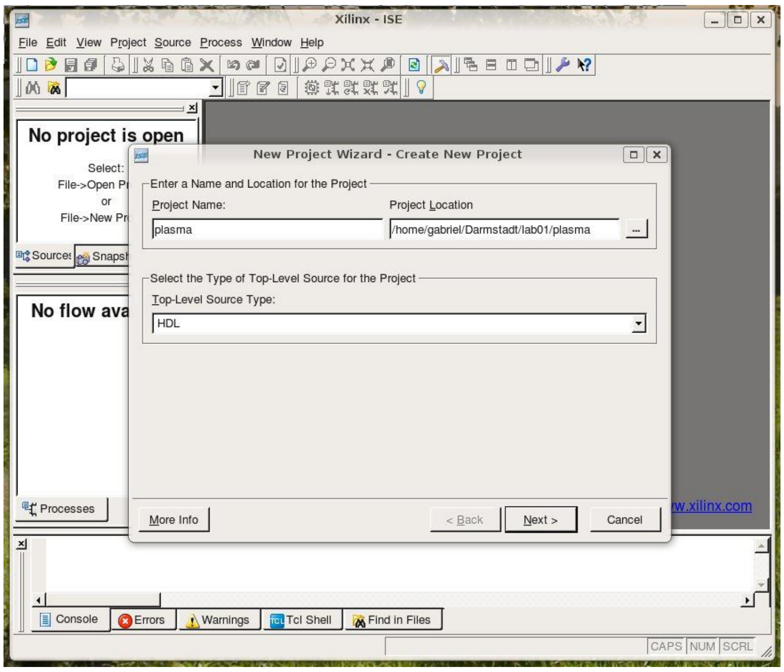
DOWNLOADING FILES

1. Download all the labs from either http://www.microelectronic.e-technik.tu-darmstadt.de/staff/sassate/ef.html or http://www.lirmm.fr/~marchesan/ (Menu 'lectures' -> 'Reconfigurable Technologies') and copy to your $(HOME) directory.

    a. Datasheet/
        i. Spartan3StarterKit/
            1. The Spartan 3 Starter Kit (PDF file)
    b. Lab3-1/
        i. Docs/
            1. The Lab3-1 manual
        ii. ISE_Project/
            1. It is an empty folder, but you will save your Xilinx ISE Project in this folder.
        iii. OS
            1. binary/
                a. All the binary files of the tasks.
            2. gcc_mips/
                a. The gcc mips cross-compiler
            3. kernel/
                a. The RTOS kernel.
            4. tools/
                a. The applications (opcodes, test1t, testmt, pi).
                b. apptools/
                    i. Some applications used when compiling the tasks. (You don't need to change it).
        iv. Screenshots/
            1. Some screenshots of the tools you have to use in this lab.
        v. Vhdl_files/
            1. Plasma/
                a. All the VHDL files of the Plasma Processor.
    c. Lab3-2/ and Lab3-3/ are coming soon.

SYNTHESIZING THE PLASMA PROCESSOR

1. Launch Xilinx ISE Tool and go to Menu File->New Project (Figure 1).
2. Select the FPGA device (Figure 2).
3. Do not choose the option to add new VHDL source, in this step just click Next.
4. Add the VHDL files to your project (Figure 3).
5. In this step all your VHDL files are compiled, just click OK
6. Generate the bitstream file (.bit) (Figure 4).
7. Generate the ROM file (to be downloaded to the PROM)
    a. Launch Xilinx Impact
    b. Select 'create a new project' and click OK.
    c. Select the second option 'prepare a PROM file' and click Next.
    d. In this step give a name to your PROM file and choose the Location. (Figure 5).
    e. Let the first option checked (by default) and click Next (Figure 6).
    f. Check the first option (Auto Select PROM) and click Next.
    g. Click Finish
    h. In this step you have to browse your bitstream file (.bit), generated by the synthesis process. After uploading your file, you will be asked for adding a new device, just choose No (Figure 7).
    i. In Menu bar, click on the Operations menu and select Generate File (Figure 8).
    j. The next step is to program the ROM with your PROM file (.mcs). On the left Menu, click on Boundary Scan option and press (Ctrl + I) to initialize chain. The first window to appear will be related to the .bit file, just click Cancel. The second window will ask you to upload the .mcs file. Choose the right file and click Open.
    k. Right click on the 'xcf04s' device and select Program (Figure 9). In this step you should receive the following message 'Programming completed successfully', if it is not the case, redo the previous step 'j'.



New Project Wizard - Create New Project

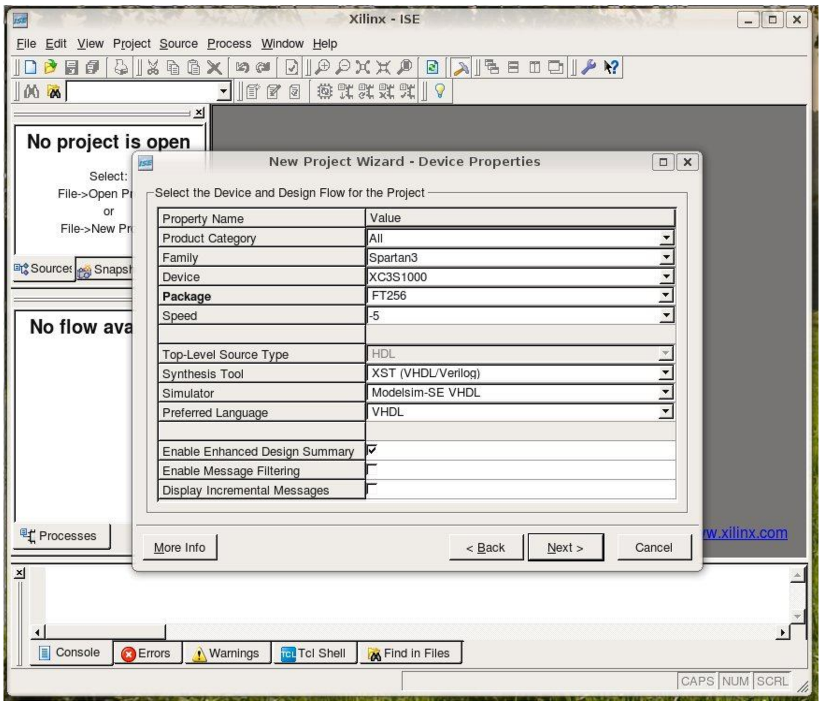
Enter a Name and Location for the Project

Project Name:	plasma
Project Location	/home/gabriel/Darmstadt/lab01/plasma

Select the Type of Top-Level Source for the Project

Top-Level Source Type: HDL

Figure 1 : Creating a new project



New Project Wizard - Device Properties

Select the Device and Design Flow for the Project

Property Name	Value
Product Category	All
Family	Spartan3
Device	XC3S1000
Package	FT256
Speed	-5
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	Modelsim-SE VHDL
Preferred Language	VHDL
Enable Enhanced Design Summary	✓
Enable Message Filtering	
Display Incremental Messages	

Figure 2 : Selecting the FPGA Device (Spartan3, XC3S1000, FT256)

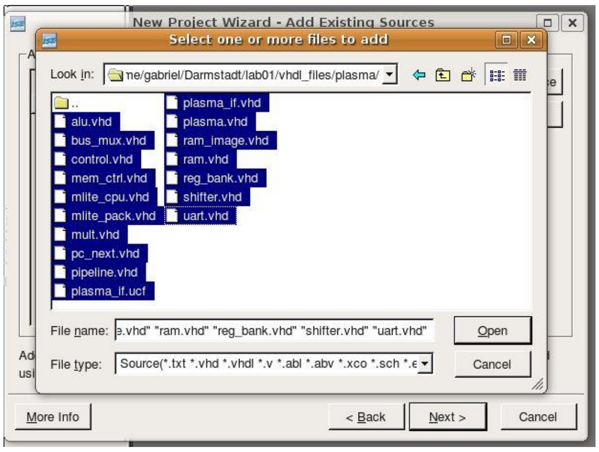
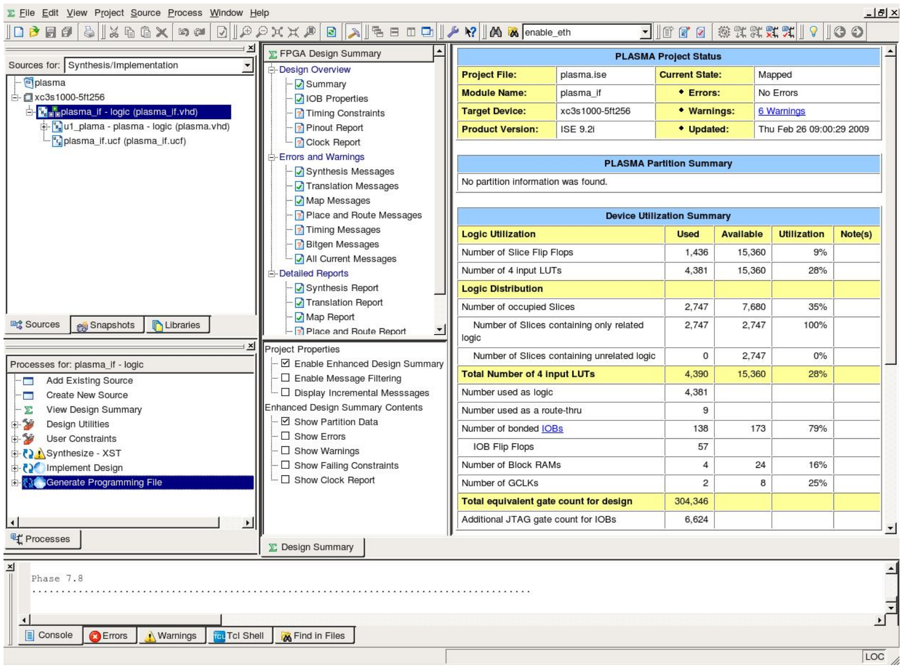


Figure 3 : Adding the VHDL files to the project



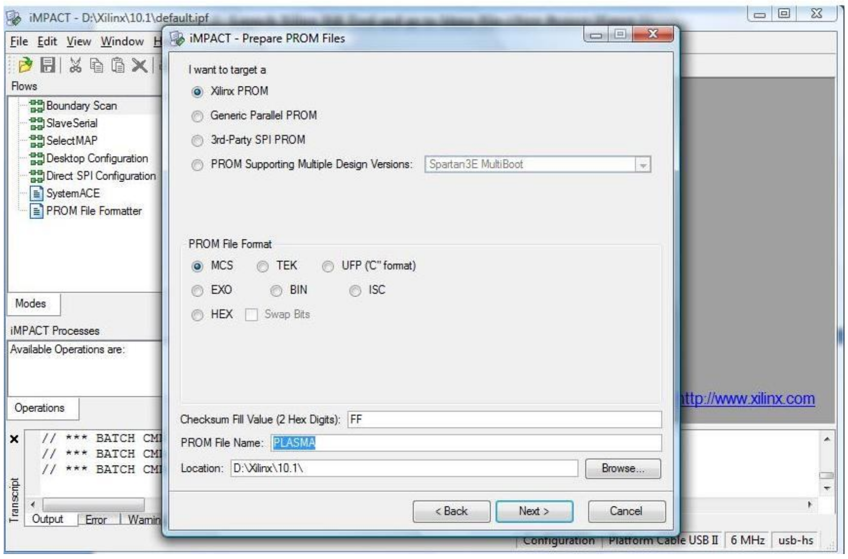
```
Phase 7.8
.................................................................................
```

PLASMA Project Status			
Project File:	plasma.ise	**Current State:**	Mapped
Module Name:	plasma_if	**• Errors:**	No Errors
Target Device:	xc3s1000-5ft256	**• Warnings:**	6 Warnings
Product Version:	ISE 9.2i	**• Updated:**	Thu Feb 26 09:00:29 2009

PLASMA Partition Summary
No partition information was found.

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	1,436	15,360	9%	
Number of 4 input LUTs	4,381	15,360	28%	
Logic Distribution				
Number of occupied Slices	2,747	7,680	35%	
Number of Slices containing only related logic	2,747	2,747	100%	
Number of Slices containing unrelated logic	0	2,747	0%	
Total Number of 4 input LUTs	4,390	15,360	28%	
Number used as logic	4,381			
Number used as a route-thru	9			
Number of bonded IOBs	138	173	79%	
IOB Flip Flops	57			
Number of Block RAMs	4	24	16%	
Number of GCLKs	2	8	25%	
Total equivalent gate count for design	304,346			
Additional JTAG gate count for IOBs	6,624			

Figure 4 : Generating the Programming File (Bitstream .bit)



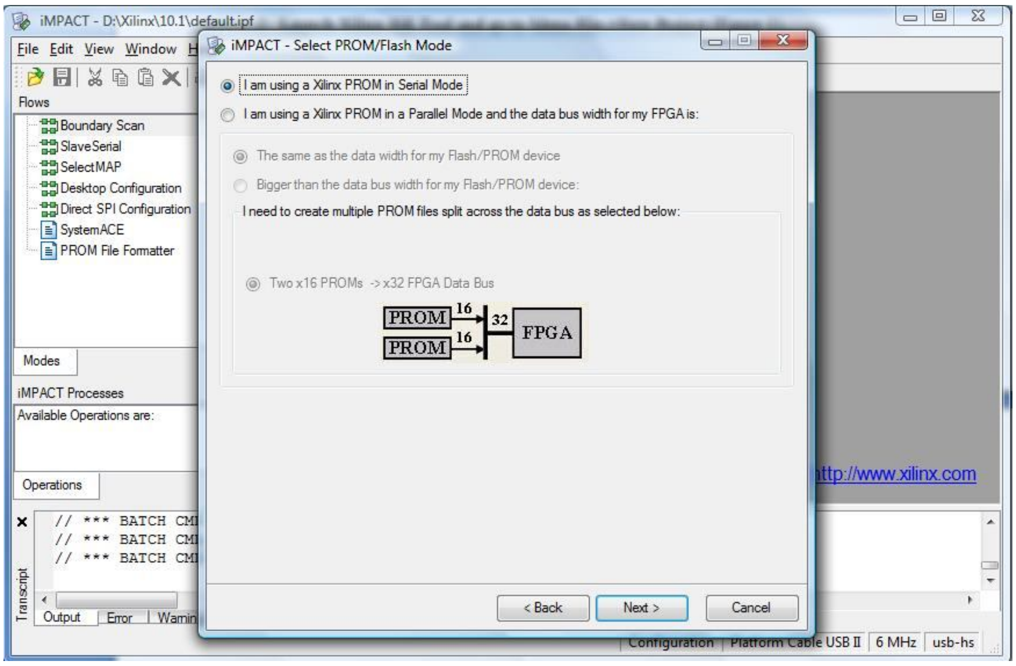
```
// *** BATCH CM
// *** BATCH CM
// *** BATCH CM
```

I want to target a
[✓ Xilinx PROM]
Generic Parallel PROM
3rd-Party SPI PROM
PROM Supporting Multiple Design Versions: Spartan3E MultiBoot

PROM File Format
[✓ MCS] TEK UFP ("C" format)
EXO BIN ISC
HEX Swap Bits

Checksum Fill Value (2 Hex Digits): FF
PROM File Name: PLASMA
Location: D:\Xilinx\10.1\

Figure 5 : Naming the PROM file and choosing the Location



iMPACT - Select PROM/Flash Mode

[✓ I am using a Xilinx PROM in Serial Mode]

I am using a Xilinx PROM in a Parallel Mode and the data bus width for my FPGA is:

The same as the data width for my Flash/PROM device

Bigger than the data bus width for my Flash/PROM device:

I need to create multiple PROM files split across the data bus as selected below:

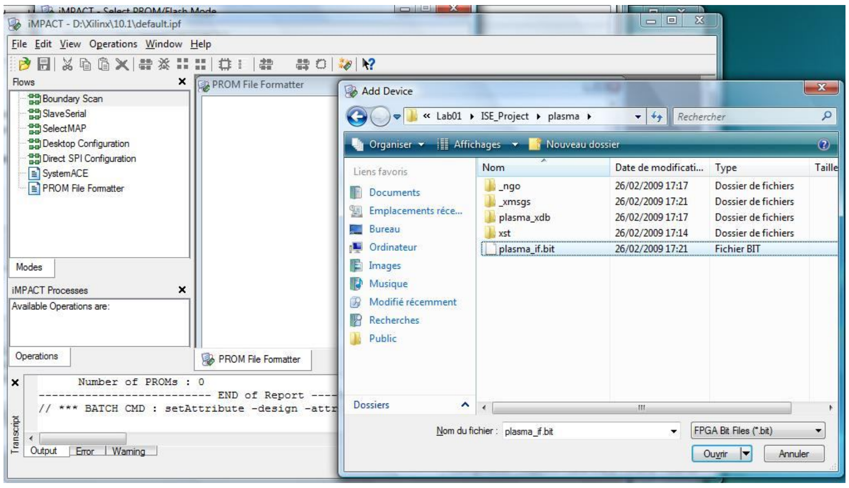
Two x16 PROMs -> x32 FPGA Data Bus

Available Operations are:

```
// *** BATCH CMI
// *** BATCH CMI
// *** BATCH CMI
```

Configuration | Platform Cable USB II | 6 MHz | usb-hs

Figure 6 : Selecting the PROM/Flash Mode



Available Operations are:

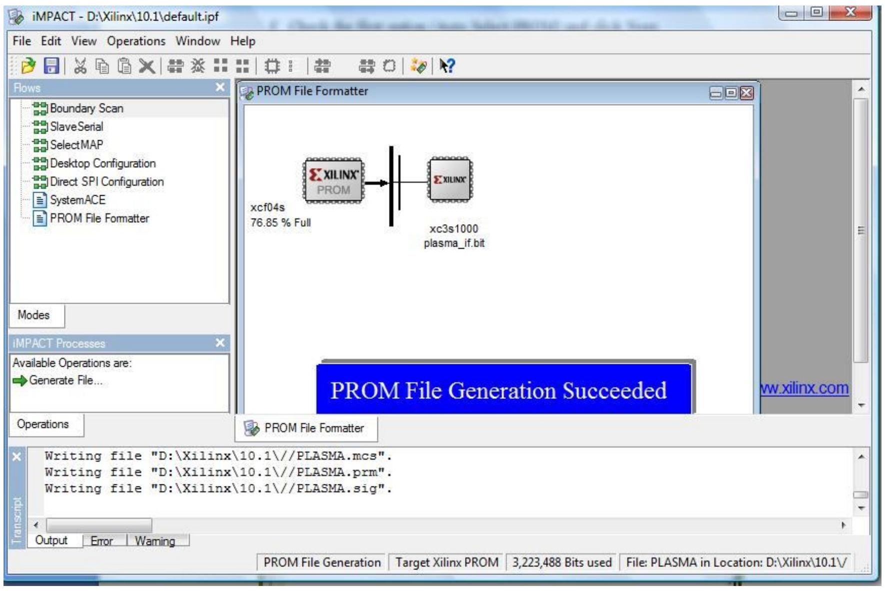
Nom	Date de modificati...	Type	Taille
_ngo	26/02/2009 17:17	Dossier de fichiers	
_xmsgs	26/02/2009 17:21	Dossier de fichiers	
plasma_xdb	26/02/2009 17:17	Dossier de fichiers	
xst	26/02/2009 17:14	Dossier de fichiers	
plasma_if.bit	26/02/2009 17:21	Fichier BIT	

Nom du fichier : plasma_if.bit

FPGA Bit Files (*.bit)

```
        Number of PROMs : 0
-------------------------- END of Report ----
// *** BATCH CMD : setAttribute -design -attr
```

Figure 7 : Choosing the bitstream file (.bit)



Available Operations are:

Generate File...

xcf04s
76.85 % Full

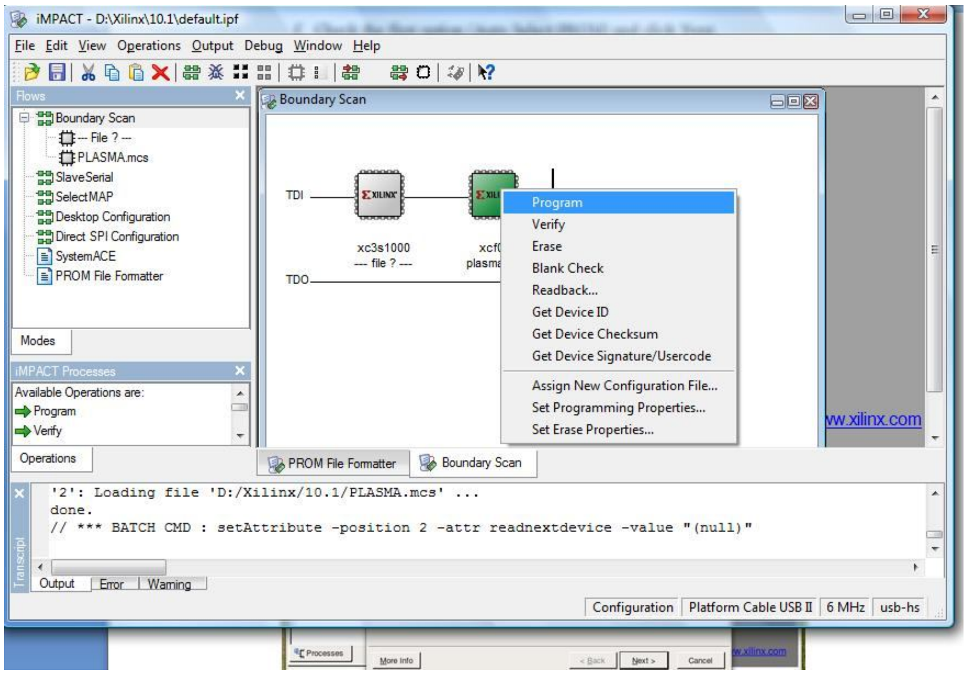
xc3s1000
plasma_if.bit

PROM File Generation Succeeded

```
Writing file "D:\Xilinx\10.1\//PLASMA.mcs".
Writing file "D:\Xilinx\10.1\//PLASMA.prm".
Writing file "D:\Xilinx\10.1\//PLASMA.sig".
```

PROM File Generation | Target Xilinx PROM | 3,223,488 Bits used | File: PLASMA in Location: D:\Xilinx\10.1\/

Figure 8 : PROM File Generation Succeeded



```
'2': Loading file 'D:/Xilinx/10.1/PLASMA.mcs' ...
done.
// *** BATCH CMD : setAttribute -position 2 -attr readnextdevice -value "(null)"
```

Figure 9 : Programming the ROM

COMPILING APPLICATIONS

1. Go to Start->Run-> and type cmd;
2. Go to $(HOME)/lab3-1/OS/tools/ directory
3. Compile the application opcodes by typing **gmake opcodes**. In this step the file opcodes.asm is compiled using gcc-mips cross compiler. The opcodes.bin file is generated and copied to $(HOME)/lab3-1/OS/binary/ directory. This file you will send through the serial port to the board.

SENDING APPLICATIONS TO THE BOARD

1. Connect the serial cable (male/female) to the Spartan 3 kit board.
2. Open the Serial Terminal [3].
3. You will receive a message asking you to establish a connection, just click Cancel.
4. In Menu bar, click on the Setup menu and select Serial Port, choose the Port (COM1, COM2, COM3, COM4) where you cable is connected on, configure the Speed to 57600 and click OK. (Figure 10).
5. Assuming your FPGA is already configured (ROM programmed), switch on the board and push the PROG Button (left side, near to VGA interface). In this step, the boot-loader, previously programmed on the FPGA will be executed.
6. After you have pushed the PROM button, you should receive a ‘greetings’ message in your serial terminal program. (Figure 11).

7. In Menu bar, click on the File menu and select Send File…, select your binary file (.bin) located in the $(HOME)/lab3-1/OS/Binary/ directory and check the box 'Binary' (Figure 12).

8. After sending the binary file to be executed on the Plasma Processor, you should receive the program output. (Figure 13).

9. Go to section 'COMPILING APPLICATIONS and redo the steps for the following applications:

    a. opcodes.asm (gmake opcodes)

        i. Application written in assembly code

    b. task1.c (gmake test1t)

        i. Application written in C code compiled with the OS (Operating System)

    c. tasks.c (gmake testmt)

        i. Application which creates two tasks running concurrently with the OS.

    d. pi.c (gmake pi)

        i. Application which calculates the PI value using the gettime() function to measure the computation time (clock cycles).

10. After you have compiled and tested the applications, you will implement the following application:

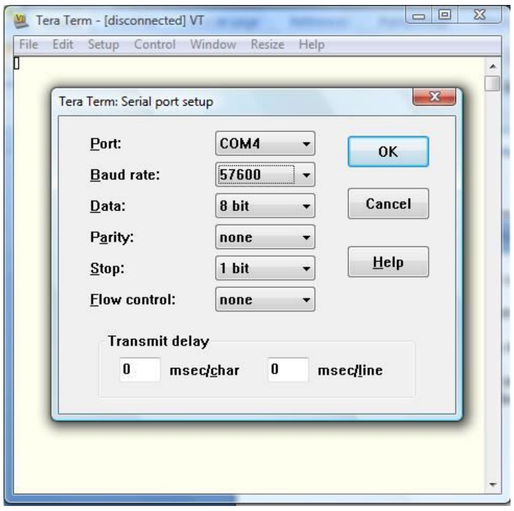
    a. Calculate the first 1000 prime numbers and print them on the screen (printf function). You can find in $(HOME)/lab3-1/OS/kernel/libc.c file some useful functions for helping you in this process. The output should be something like presented in Figure 15. The output should be something like shown in Figure 14.*

       *Run the application on the Plasma processor and get the performance of your application (in number of clock cycles). Before checking your application is working properly, remove the printf calls and measure the performance.

11. Write a short report describing the performance results for your application and attach the source code to the report.

12. Send it for both of us sassatelli@lirmm.fr, marchesan@lirmm.fr with the following subject:

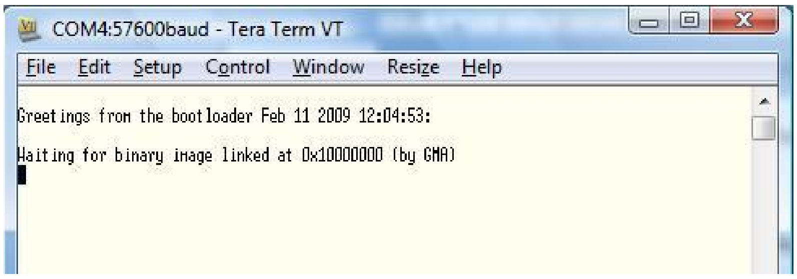
    Subject: Reconfigurable Technologies TUD: LAB3.1 [Yourname]


Tera Term - [disconnected] VT
File Edit Setup Control Window Resize Help
Tera Term: Serial port setup
OK
Cancel
Help

Port: COM4
Baud rate: 57600
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none

Transmit delay
0 msec/char 0 msec/line

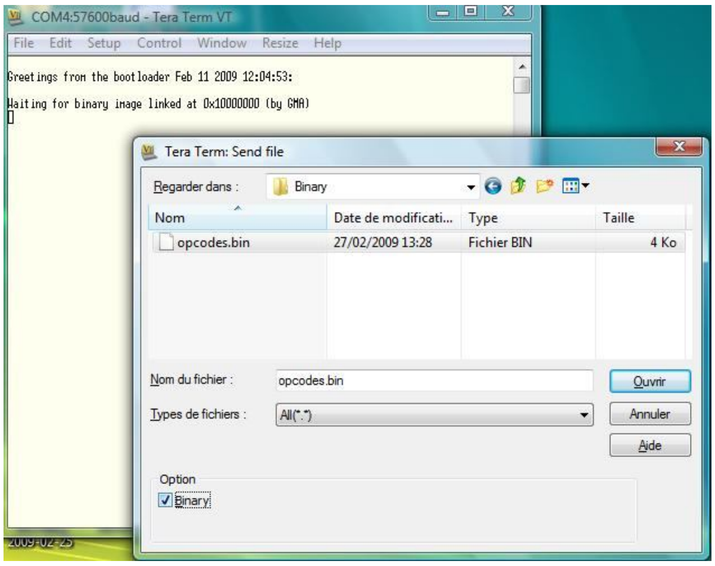
Figure 10 : Serial Port Setup


COM4:57600baud - Tera Term VT
File Edit Setup Control Window Resize Help

```
Greetings from the bootloader Feb 11 2009 12:04:53:

Waiting for binary image linked at 0x10000000 (by GMA)
```

Figure 11 : Greetings from the boot loader


COM4:57600baud - Tera Term VT
File Edit Setup Control Window Resize Help
Tera Term: Send file
Ouvrir
Annuler
Aide
2009-02-25

```
Greetings from the bootloader Feb 11 2009 12:04:53:

Waiting for binary image linked at 0x10000000 (by GMA)
```

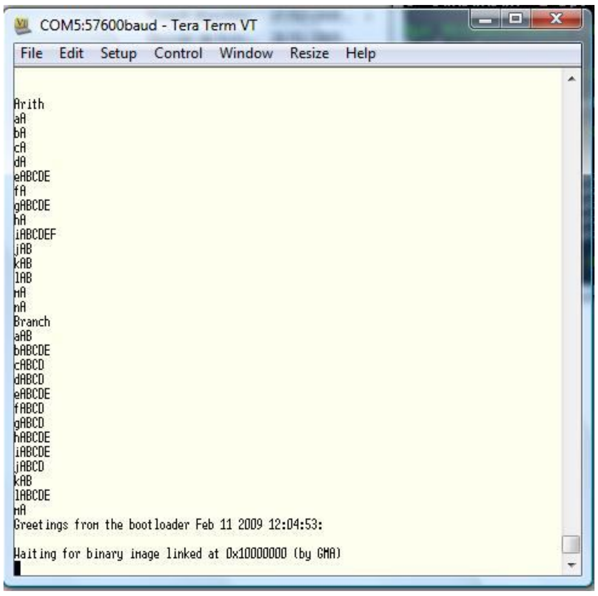
Regarder dans : Binary

Nom	Date de modificati...	Type	Taille
opcodes.bin	27/02/2009 13:28	Fichier BIN	4 Ko

Nom du fichier : opcodes.bin
Types de fichiers : All(*.*)

Option
[✓] Binary

Figure 12 : Sending binary files using Tera Term

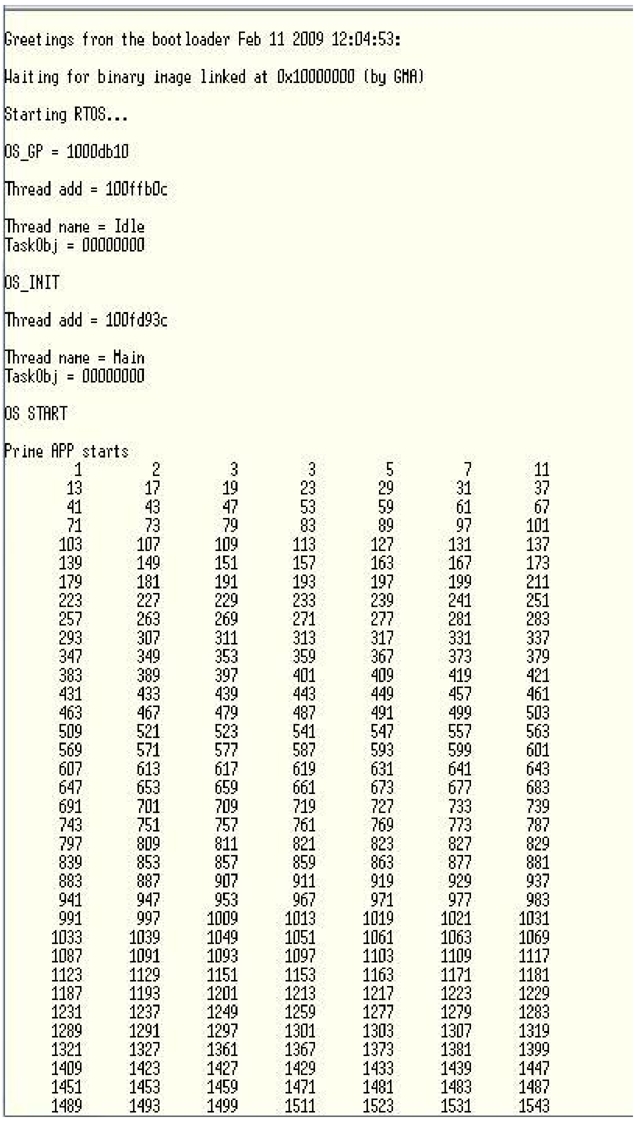


COM5:57600baud - Tera Term VT
File Edit Setup Control Window Resize Help

```
Arith
aA
bA
cA
dA
eABCDE
fA
gABCDE
hA
iABCDEF
jAB
kAB
lAB
mA
nA
Branch
aAB
bABCDE
cABCD
dABCD
eABCDE
fABCD
gABCD
hABCDE
iABCDE
jABCD
kAB
lABCDE
mA
Greetings from the bootloader Feb 11 2009 12:04:53:

Waiting for binary image linked at 0x10000000 (by GMA)
```

Figure 13 : Application output



```
Greetings from the bootloader Feb 11 2009 12:04:53:

Waiting for binary image linked at 0x10000000 (by GMA)

Starting RTOS...

OS_GP = 1000db10

Thread add = 100ffb0c

Thread name = Idle
TaskObj = 00000000

OS_INIT

Thread add = 100fd93c

Thread name = Main
TaskObj = 00000000

OS START

Prime APP starts
         1         2         3         3         5         7        11
        13        17        19        23        29        31        37
        41        43        47        53        59        61        67
        71        73        79        83        89        97       101
       103       107       109       113       127       131       137
       139       149       151       157       163       167       173
       179       181       191       193       197       199       211
       223       227       229       233       239       241       251
       257       263       269       271       277       281       283
       293       307       311       313       317       331       337
       347       349       353       359       367       373       379
       383       389       397       401       409       419       421
       431       433       439       443       449       457       461
       463       467       479       487       491       499       503
       509       521       523       541       547       557       563
       569       571       577       587       593       599       601
       607       613       617       619       631       641       643
       647       653       659       661       673       677       683
       691       701       709       719       727       733       739
       743       751       757       761       769       773       787
       797       809       811       821       823       827       829
       839       853       857       859       863       877       881
       883       887       907       911       919       929       937
       941       947       953       967       971       977       983
       991       997      1009      1013      1019      1021      1031
      1033      1039      1049      1051      1061      1063      1069
      1087      1091      1093      1097      1103      1109      1117
      1123      1129      1151      1153      1163      1171      1181
      1187      1193      1201      1213      1217      1223      1229
      1231      1237      1249      1259      1277      1279      1283
      1289      1291      1297      1301      1303      1307      1319
      1321      1327      1361      1367      1373      1381      1399
      1409      1423      1427      1429      1433      1439      1447
      1451      1453      1459      1471      1481      1483      1487
      1489      1493      1499      1511      1523      1531      1543
```

Figure 14 : Prime application output

4. References

[1] Steve Rhoads. Plasma - most MIPS I(TM) opcodes: Overview. Available at http://www.opencores.org/projects.cgi/web/mips/overview.htm, 2009.

[2] Xilinx ISE Foundation. Available at http://www.xilinx.com/ise/logic_design_prod/webpack.htm, 2009.

[3] Tera Term. Free Terminal Emulator. Available at http://ttssh2.sourceforge.jp/, 2009.