

## SoC Design on FPGA devices

### Polytech Nice Sophia Antipolis

During this first lab you will design your first System on Chip (SoC) by using the Qsys tool of the Quartus Prime Framework.

## I. Installation

Install the following software to use the DE1-SoC board integrating a Altera FPGA device :

- quartus prime Lite edition version 16.1
- nios II EDS 16.1
- Altera University Program 15.0

Plug the DE1 SoC board to a free USB port, power on the board and install the driver.

## II. Hardware design

Run Quartus Prime and create a new project and choose the following device family :

Cyclone V : 5CSEMA5F31C6

Launch QSYS from the Tool menu.

### a. Generate code of your SoC

QSYS is a graphical editor to build a custom SoC from the library of IP (Intellectual Property) blocks provided by Altera.

The main component of your SoC architecture is an Embedded Processor : Nios II Processor.

Instantiate the following components in your design:

- Nios II processor,
- On-chip memory (RAM or ROM)
- PIO (Parallel Input / output) : direction=Input, size=8, name=switches
- PIO (Parallel Input / output) : direction=Output, size=8, name=ledr
- PIO (Parallel Input / output) : direction= Output, size=8, name=hex0
- PIO (Parallel Input / output) : direction= Output, size=8, name=hex1
- PIO (Parallel Input / output) : direction= Output, size=8, name=hex2
- PIO (Parallel Input / output) : direction= Output, size=8, name=hex3
- PIO (Parallel Input / output) : direction= Output, size=8, name=hex4
- PIO (Parallel Input / output) : direction= Output, size=8, name=hex5
- PIO (Parallel Input / output) : direction=Input, size=4 name=key
- JTAG UART
- Interval Timer, period = 1ms

After having instantiating the components you still have to configure your system by the following steps:

- Connect the ports of the components to the data master, instruction master, clk\_0.clk, clk\_0.clk\_reset ports,
- Export the external connection of each PIO by clicking in the corresponding entry in column 5,
- Assign the base addresses (menu System),
- choose the reset vector and the exception vectors of the processor into the on-chip memory,
- configure the interrupt priorities in the column 9,
- generate the HDL code of your design.

#### b. VHDL description of your design

You can observe the interface of your on-chip design in menu Generate -> Show Instantiation Template.

Copy the template and paste it in a new VHDL File by coming back to Quartus Prime.

Complete the template with the following lines and replace the CONNECTED\_TO\_ signals by one of the main circuit pin name.

```
LIBRARY ieee;
```

```
USE ieee.std_logic_1164.ALL;
```

```
USE ieee.std_logic_unsigned.ALL;
```

```
ENTITY CPU IS
```

```
PORT (
```

```
    CLOCK_50 : IN STD_LOGIC;
```

```
    KEY : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
```

```
    SW : IN STD_LOGIC_VECTOR (9 DOWNTO 0);
```

```
    HEX0 : OUT STD_LOGIC_VECTOR(7 downto 0);
```

```
    HEX1 : OUT STD_LOGIC_VECTOR(7 downto 0);
```

```
    HEX2 : OUT STD_LOGIC_VECTOR(7 downto 0);
```

```
    HEX3 : OUT STD_LOGIC_VECTOR(7 downto 0);
```

```
    HEX4 : OUT STD_LOGIC_VECTOR(7 downto 0);
```

```
    HEX5 : OUT STD_LOGIC_VECTOR(7 downto 0);
```

```
    LEDR : OUT STD_LOGIC_VECTOR (9 DOWNTO 0)
```

```
);
```

```
END CPU;
```

```
ARCHITECTURE my_rtl OF CPU IS
```

```
BEGIN
```

```
LEDR(9 downto 8) <= SW(9 downto 8);
```

```
END my_rtl;
```

### c. Pin assignment to adapt to the board and compilation

Once your VHDL description is ended, you can import the pin assignment of the DE1-SoC board from the file DE1\_SoC\_Pin\_assignment.QSF

You can then compile your SoC architecture in order to generate the Bitstream file which will be used to configure the FPGA device on board.

If the compilation is correct (warnings are normal), program the board:

- menu Tool -> Programmer
- auto-detect
- add device -> SoC Series V -> SOCVHPS
- add file -> Output File -> ProjectName.sof

At this step, the hardware design phase is finished. If you encountered errors, use the solution file in the corresponding directory.

## III. Software design

### a. First program for your SoC

Launch the Nios II EDS software (a eclipse plugin) and create a new "Nios II application and BSP from template".

Select the SOPC info file of your hardware design, your project name and the Small Hello World template.

Build the software project and run it onto the board to validate the hardware design.

### b. Address base access to the peripherals

Modify your C code to display on the 7-segments the number of iteration of the infinite loop.

As a reminder, the following table contains the configuration of the 7 segments for the 10 first hexadecimal digits :

```
#include "alt_types.h"
```

```
#include "system.h"
```

```
#include "altera_avalon_pio_regs.h"
```

```
int iter = 0;
```

```
static alt_u8 segments[16] = {  
    0x81, 0xCF, 0x92, 0x86, 0xCC, 0xA4, 0xA0, 0x8F, 0x80, 0x84, /* 0-9 */  
    /* to be completed */}; /* a-f */
```

```
alt_u32 data = segments[iter];
```

```
IOWR_ALTERA_AVALON_PIO_DATA(SEVEN_SEG_PIO_BASE, data);
```