Specific Parallel Port IP

ProgInt
EmbSys
Avalon

Objective	**Be able to realize a programmable interface for a SOC: a specific Parallel Port**
Tools	Quartus II, Qsys, NIOSII, NIOS SBT, FPGA board
Preliminaries	VHDL, Embedded System on FPGA, Avalon Bus
Theory	Embedded System on FPGA, Avalon Bus
Material	FPGA board + Tools
Duration	2h

1 Introduction

In this laboratory, we have to design a specific parallel programmable interface module in VHDL. This module is on the Avalon bus in register mode, 8 bits width.

The designed interface has to be including in a NIOSII system, and tested with a small C program.

For the documentation on fpga board go to :

http://www.terasic.com.tw/attachment/archive/941/DE0-Nano-SoC_User_manual.pdf

2 Programmable interface specifications

The Parallel Port to design has the following characteristics:

- 8 bits width
- Programmable direction for each individual bit through a **Data Direction Register**
- A direct Write/Read access to the **Parallel Port Register**
- A **Pin Register** to read data at the pin level
- A write access to specify selected bits to be modify as '1', and no change for the other unselected bits. The selected bits are those when a '1' is write at the bit position, the bits with a '0' are not modified. The **Parallel Port Register** is modified by this access.
- Same as previous access but for putting a '0'. A write with a '1' will modify the port with a '0'. The bits with a '0' writing are not changed.

Question 1 *Parallel Port*

1.1 What is the register map for your interface?

1.2 How many wait states you need for your design on the Avalon bus?

1.3 How do you implement the register model of your interface?

Create a new project to realize your programmable interface and simulate the read and write accesses in Quartus using the simulator Tools ModelSim.

3 NIOS II design

On the simple NIOS II design with the following components, add your parallel port design. For this function, you need to create your own library component.

On Qsys → Create new component and provide the needed information. Generate the IP. Now it can be used as all the others components.

- Select NIOS II /s processor, with **4k bytes** instruction cache
- Add SRAM **128kB**ytes with **32 bits** width

- Add epcs controller
- Jtag interface
- **Your Parallel port 8 bits (ParPort0)**
- **Your Parallel port 8 bits (ParPort1)**
- Save
- Generate in VHDL
- Exit

Question 2

2.1 What are the addresses of your Parallel ports components?

3.1 Quartus Design

The Qsys symbol is added to the schematic or you can use the component to be included in a top level entity.

Add pins for:

- Clk,
- Reset
- ParPort0 (**only 4 bits for 4 switches as bits(3..0), 4 bits on an unused extension connector pins for highest 4 bits (7..4)**)
- ParPort1 (8x Green LEDs) parallel ports

Add pins number (search on the web site), they are the same as the previous laboratory.

- Compile the design and correct the errors.
- Connect the FPGA board on USB.
- Download the .sof generated → you are programming the hardware FPGA
- Open again the NIOSII System (clic-clic on the symbol)
- Select NIOS SBT → you will go to the Eclipse project manager

3.2 NIOS SBT

Create a Working Space in your account, but NOT in the project directly or use a previous one.

Program a InitPort() Function to select ParPor0 as input and ParPor1 as Output.

Make a simple loop for Reading ParPort0 and copying on the ParPort1.

Question 3

3.1 What is in your InitPort() function?

3.2 How do you make your copy loop?

Realize a Chenillard using the Write a '1', Write a '0' capability of your programmable interface.

Use the logic analyzer function of Quartus (Signal Tap Logic Analyzer) to see the Avalon and Port timing diagram.

Question 4

4.1 How do you use the specific function of your interface for the « chenillard » ?

4.2 Analyze the timing diagram of your accesses.

4.3 How long it take from the Avalon access to modify the Output?

Réponse 1

1.1 What is the register map for your interface?

1.2 How many wait states you need for your design on the Avalon bus?

1.3 How do you implement the register model of your interface?

Réponse 2

2.1 What are the addresses of your Parallel ports components?

Réponse 3

3.1 What is in your InitPort() function?

3.2 How do you make your copy loop?

Réponse 4

4.1 How do you use the specific function of your interface for the « chenillard » ?

4.2 Analyze the timing diagram of your accesses.

4.3 How long it take from the Avalon access to modify the Output?

Table des matières