

# ECE 3550 DIGITAL DESIGN

FALL 2011

## Project Assignment #1

(Team Project)

**Total: 40 pts. (10 % of the course grade)**

**Due 4:00pm, Thursday, November 3, 2011**

**Design, simulate, build, and demonstrate** the correct operation of the **serial data processing module** specified below. A **functional block diagram** is given on Page 2. Use one Xilinx **Spartan3E FPGA** chip (the one that is mounted on your Xilinx Nexys 2 Board), as well as **off-the-shelf ICs** and **other parts** (as needed). Input signals are to be driven by **micro switches**, output signals are to be displayed by **Bar-LED** Units. Pay attention to the control of the bi-directional I/O Port of the 'HC299 Shift Register: after an operand has been loaded into the shift register the micro switches should be electronically separated from the port (by means of a Three-State buffer chip) to allow displaying the contents of the shift register. You may assume that the input signals to the system are properly synchronized with respect to the **CLK** signal (i.e., you will set/reset them that way). Make sure that your CLK signal will be **bounce-free**. **You are to provide for your own Nexys 2 Board, Circuit Bread Board and electronic parts, respectively.**

### Description of Operation

Initially, signal **DONE** (output) is **asserted** (High), signal **OVF** (arithmetic overflow, output) is **deasserted** (Low), signal outputs **OP\_LD** (Operand Load) and **RES\_LD** (Result Load) are also **deasserted** (Low). Signal **START** (input) should be in **deasserted** (Low) state. The two shift registers must be in **Hold Mode**, i.e., their contents are being displayed on Bar-LED modules and the image won't change while a train of clock pulses is received.

In order to begin with a **new arithmetic operation**, the user should **set up the two operands (Operand\_1 and Operand\_2 in 8-bit, in 2's complement representation)** using DIP Switches and **then assert the START signal**. In response, the System Controller (SC) should **deassert DONE**, and **assert OP\_LD**. **While OP\_LD is high**, the SC should first **load in parallel** (broadside) the two operands into the associated shift registers ('**HC299 Universal Shift Register with 3-State Outputs**, or compatible part). After the parallel load is executed the SC should control the **shifting** of the two operands from the shift registers into **temporary storage registers** on the **Spartan3E** chip by placing the shift registers in **Shift Right Mode**. At some point during the load process signal **START** should be **deasserted** by the user. After the two operands are moved into the storage registers signal **OP\_LD** should be **deasserted** by the SC.

After the two operands are secured in their storage registers the arithmetic operation should be performed as follows: **RES = (Operand\_2 – Operand\_1) signed-multiplied by 4**, where **Operand\_1** stands for the data received from Shift Register #1, **Operand\_2** stands for the data received from Shift Register #2 and **RES** stands for the result in **8-bit, 2's complement representation**. The SC should then **shift out the bits of RES** into

**Shift Register #1** and simultaneously **assert** signal **RES\_LD**. When the shift out operation is concluded **RES\_LD** should be **deasserted** and **DONE** should be **asserted**. **OVF** should only be asserted if an **arithmetic overflow** has occurred. **Both** shift registers should now be in **Hold Mode**. The whole operation may be repeated at this point

Assume **8-bit 2's complement** representation for the **two operands** (that is, you will be setting up the data in 2's complement form, and the results will also be generated in 2's complement form).

The system shouldn't change its state (RES remains unchanged) by receiving CLK pulses while the DONE signal is asserted (Hold Mode). You should provide for a **RESET#** signal (**active low**) to reset the whole system to its initial state at any time.

In order to facilitate the testing and debugging of the project a **set of test data along with the expected results** will be posted on the Class Web Page.

### Tasks

- a) Give the control algorithm for the System Controller (SC) in the form of an **ASM chart along with comments**. (4 pts.)
- b) Draw a **detailed schematic diagram** for the whole system. Clearly show the partitioning of the system into standard ICs, other parts, and the FPGA chip. If you have implemented any logic functions **external** to the FPGA then you should document them by Boolean expressions. (6 pts.)
- c) Use the **Xilinx ISE 13.x or Webpack 13.x tools along with Mentor Graphics ModelSim** to develop the FPGA segment of the system. You should prove the correctness of your design by **post-route simulations**. Turn in the printouts of your design files (**.vhd**, and **.do** files, as well as **sections of your Pad Report** with the **resource utilization** and the **device pinout diagram**, respectively), and your **simulation timing diagrams**.  
**Comment** on your simulation results. Your comments should refer to **each** pair of operands applied along with the **result** and the status of the **OVF** signal.  
**Hint: Force** pin assignments for your signals through the User Constraints Utility so when you need to modify your **VHDL** design file then you can still keep the wiring to the Nexys 2 Board unchanged. (10 pts.)
- d) **Build the whole system** using your Nexys 2 Board and your Breadboard and **demonstrate** its correct operation to the course instructor. (15 pts.)
- e) Submit a Project Report. (5 pts.)

Each Team should submit a **joint Project Report**. The report should have the main sections as follows: **Introduction**, **Design**, **Design Evaluation**, and **Conclusion**. Your solutions to Tasks a) to d) should be represented in the report.

**Demonstrations** will be arranged in the Microcomputer Lab, **Room B-214**. Dates and times will be posted.

**Note:** you will lose **4 pts. by each day** your project (**a demonstrated circuit along with the report**) is tardy. **No** credit will be given if the project is late by **five business days**.

## **Bonus Assignment #1**

**(Team Project)**

**Total: up to 6 pts. (1.5 % of course grade)**

**Due 4:00pm, Thursday, November 3, 2011**

**Similar** problem to Project #1 but now the **data flow is more parallel** in nature. In addition, **no** circuit-level demonstrations are required. You should prove the correct operation of your design by **post-route simulations only**.

The external shift registers will be eliminated. Assume that the user will load the two operands (in sequence) through an **8-bit bi-directional Three-State Data Bus D7 – D0** (D0 is LSB) into storage registers that are implemented on the FPGA. The SC should then **calculate** the result of **RES = (Operand\_2 – Operand\_1) · signed-multiplied by 4** **as soon as possible** and then **display** it on the **D7 – D0 Bus**. You should **define all the necessary input and output signals** of the SC to accomplish this task and to **avoid** any potential **data bus contention** on the D7 –D0 Data Bus.

You should **first** demonstrate and turn in your report for **Project #1** in order to get any credit for a solution to this Bonus Assignment. **Tasks a), b), c) and e) apply** here, too.