Lab 12

Lab Project Capstone

Description

In this lab, you will first implement and test increasing sizes of the carry-look-ahead adder discussed previously in the lecture section of the class. Then, you will use this hardware along with the components from Lab 11 to construct and test the full 64-bit two-level multiplier. This is the final lab of the lab project.

Procedure

Part A – 4 Bit and 16 Bit CLA Adders

- In this module, you will implement a 4-bit carry-look-ahead adder and then incorporate it into a 16-bit CLA adder to understand how CLA adders are recursively structured.
- Download 12A-handout and the files listed below from Canvas and complete this module, following along with the instructor.
- Files to download (for Part A only):
  - `lab12a.vhd`
  - `display_controller.vhd`
  - `cla_16.vhd`
  - `cla_4.vhd`
  - `cla_group_logic.vhd`
  - `partial_full_adder.vhd`
  - `lab12a.xdc`

Part B – 64 Bit CLA Adder using Serial Communication

- In this module, you will synthesize, implement, upload, and test a hardware description for a 64-bit carry-look-ahead adder, encapsulated within a serial transceiver.
- Download 12B-handout and the files listed below from Canvas and complete this module, following along with the instructor.

- Files to download:
  - CLA Files – for Parts B and C
    - `cla_pow_4.vhd`
    - `cla_group_logic.vhd`
    - `partial_full_adder.vhd`
  - Transceiver Files – for Parts B and C
    - `basys3_mk8_apex.xdc`
    - `d_flip_flop.vhd`
    - `mk8_apex_128.vhd`
    - `mk8_container_lab12.vhd`
    - `mk8_rx_module.vhd`
    - `mk8_tx_module.vhd`
    - `mk8_xcvr_generic.vhd`
    - `synchronizer_2ff.vhd`

Part C – 64 Bit Two-Level Multiplier

- In this module, you will finalize the lab work for the project. You will investigate, synthesize, implement, upload, and test the hardware description for a 64-bit two-level multiplier, encapsulated within a serial transceiver.
- Download 12C-handout and the files listed below from Canvas and complete this module, following along with the instructor.

- Files to download:
  - 2LMR Files – for Part C only
    - `mux_generic.vhd`
    - `priority_encoder_8.vhd`
    - `priority_encoder_64_2l.vhd`
    - `barrel_shifter_generic.vhd`
    - `decoder_generic.vhd`
    - `multiplier_mk2.vhd`
  - CLA Files – for Part C only
    - `cla_pow_2.vhd`
    - `cla_half_group_logic.vhd`
  - CLA Files – for Parts B and C
    - `cla_pow_4.vhd`
    - `cla_group_logic.vhd`
    - `partial_full_adder.vhd`
  - Transceiver Files – for Parts B and C
    - `basys3_mk8_apex.xdc`
    - `d_flip_flop.vhd`
    - `mk8_apex_128.vhd`
    - `mk8_container_lab12.vhd`
    - `mk8_rx_module.vhd`
    - `mk8_tx_module.vhd`
    - `mk8_xcvr_generic.vhd`
    - `synchronizer_2ff.vhd`

Deliverables

Lab Report

- Submit an **informal report** including the following:
  - Screenshots of Elaborated Design schematics for Parts A, B, and C
  - Screenshots of FPGA resource utilization for Parts A, B, and C
  - Pictures of the FPGA development board (as specified) for Part A
  - Printouts of data spreadsheets for Parts B and C
  - Screenshots of SerialTool with test results for Parts B and C
  - Description of whether experimental results match expected results for Part B

Outcomes

- Understand the recursive structure of a carry-look-ahead adder.
- Practice working with VHDL.
- Practice using Vivado for hardware synthesis and implementation.
- Practice programming and testing a hardware description on an FPGA development board.
- Practice using serial communication to test hardware descriptions by sending data to and receiving data from an FPGA development board.
- Practice working with Intellectual Property (IP) Cores.
- Practice using an onboard MMCM to generate a slower clock from the base 100 MHz clock on the FPGA development board.
- Understand a complex hardware description.
- Understand how high-precision integer multiplication can be implemented efficiently.