EE4620/6620, CEG4324/6324
Digital Integrated Circuit Design
with PLDs and FPGAs

Project:

Xilinx Vivado Multiple 6-Bit Synchronous Pipeline Parallel Adders: Design and Simulation in FPGA

A. Introduction

The main objective of this lab is for you to explore the design space to design a **6-bit synchronous pipeline parallel adder**, using the Xilinx Vivado simulation environment. You will design a VHDL project in Xilinx Vivado and functionally verify and validate your design along with the implementation steps to upload the design through the Vivado simulation environment. Read through all of the steps before you begin the lab.

Instructions

Pipeline is an important technique used in several applications such as digital signal processing (DSP) systems, microprocessors, etc. Pipeline results in speed enhancement for the critical path in most DSP systems. For example, it can either increase the clock speed or reduce the power consumption at the same speed in a DSP system. The advantage of pipelining is that it increases the throughput of the system when processing a stream of tasks. However, applying too many pipelined functions can lead to increased latency - that is, the time required for a single task to propagate through the full pipe is prolonged.

Example: Non-pipeline 4-bit parallel adder verse 4-bit pipeline parallel adder

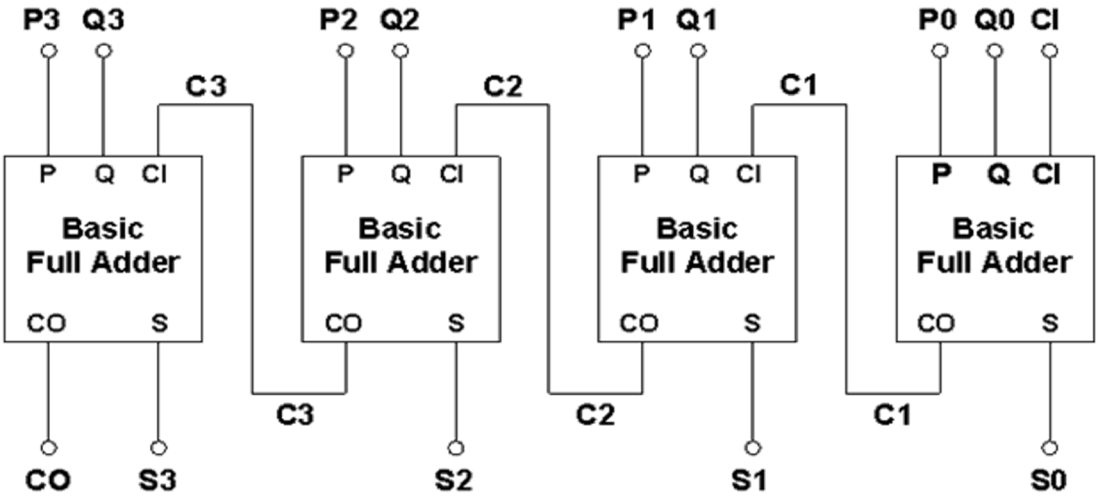

Figure 1. Example: 4-bit parallel adder architecture

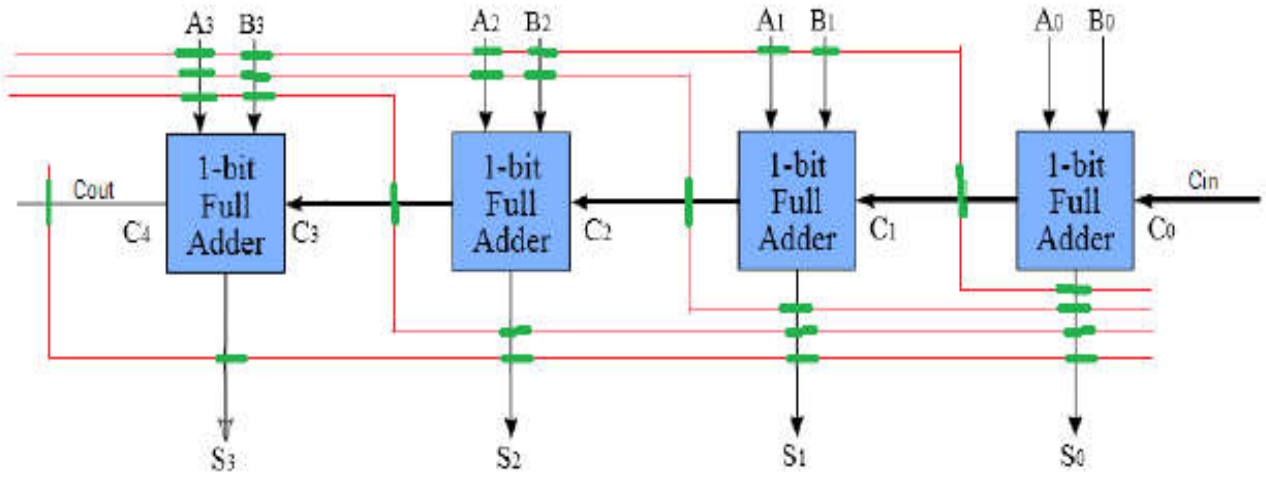

Figure 2. Example: 4-bit synchronous pipeline adder architecture (4 pipelines)

In the above pipeline adder, the green lines represent the registers, and the red lines are the guide for determining the distribution of the registers on the same synchronous pipeline.

B.1 6-bit synchronous pipeline parallel adder

In this project you will design a VHDL project for a **6-bit synchronous pipeline parallel adder**, using the Xilinx Vivado simulation environment.

B. Project Grading for EE4620L/CEG4324L Students [150 pts] and for EE6620L/CEG6324L Students [200 pts]

C.1 [100 pts]

Turn in a written report that captures your approach to a **6-bit synchronous 6-pipeline parallel adder**. It should include:

1. Pipeline architecture of your 6-bit design, similar to Fig. 1 of 4-bit pipeline adder architecture except there are **6 pipeline stages**.
1. VHDL code and testbench of your **6-bit synchronous 6-pipeline parallel adder**. Note: Complete the VHDL testbench and determine the minimum clock period to continue obtaining correct pipeline adder results.
2. Your functional verification.

   The following test cases for your synchronous pipeline adder are required:

   A_{10} = (7, 62)

   B_{10} = (30, 63)

   You must test for all A+B cases with initial carry-in C_0 = 0 and 1, which means you should verify a total of **4 x 2 = 8** input and output values.

Note: All answers in the waveforms must be labeled to show correct answers. Annotate the inputs and outputs in hexadecimal values next to their binary equivalents in the simulation waveform. Include all waveform snapshots.

C.2 [50 pts]

Turn in a written report that captures your approach to a **6-bit synchronous 2-pipeline parallel adder**. It should include:

2. New pipeline architecture of your design where **bit 0, bit 1, and bit 2 are in one pipeline stage**; **bit 3, bit 4, and bit 5 are in one pipeline stage**. So, there are **2 pipeline stages**.
3. VHDL code and testbench of the **6-bit synchronous 2-pipeline parallel adder**. Note: Complete the VHDL testbench and determine the minimum clock period to continue obtaining correct pipeline adder results.
4. Your functional verification.

   The following test cases for your synchronous pipeline adder are required:

   A_{10} = (7, 62)

   B_{10} = (30, 63)

   You must test for all A+B cases with initial carry-in C_0 = 0 and 1, which means you should verify a total of **4 x 2 = 8** input and output values.

   Note: All answers in the waveforms must be labeled to show correct answers. Annotate the inputs and outputs in hexadecimal values next to their binary equivalents in the simulation waveform. Include all waveform snapshots.

C.3 Additional lab requirement for EE6620L/CEG6324L Students [50 pts]

Turn in a written report that captures your approach to a **6-bit synchronous 3-pipeline parallel adder**. It should include:

5. New pipeline architecture of your design where **bit 0 and bit 1 are in one pipeline stage**; **bit 2 and bit 3 are in one pipeline stage; bit 4 and bit 5 are in one pipeline stage.** So, there are **3 pipeline stages**.
6. VHDL code and testbench of the **6-bit synchronous 3-pipeline parallel adder**. Note: Complete the VHDL testbench and determine the minimum clock period to continue obtaining correct pipeline adder results.
7. Your functional verification.

   The following test cases for your synchronous pipeline adder are required:

   A_{10} = (7, 62)

B_{10} = (30, 63)

You must test for all A+B cases with initial carry-in C_0 = 0 and 1, which means you should verify a total of **4 x 2 = 8** input and output values.

Note: All answers in the waveforms must be labeled to show correct answers. Annotate the inputs and outputs in hexadecimal values next to their binary equivalents in the simulation waveform. Include all waveform snapshots.