WRIGHT STATE
UNIVERSITY

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

Project (Summer 2024)

Xilinx Vivado Multiple 8-bit Synchronous Pipeline Parallel Adders: Design and Simulation in FPGA

A. Introduction

This lab's main objective is for you to explore the design space to design **8-bit synchronous pipeline parallel adders** using the Xilinx Vivado simulation environment. You will create a VHDL project in Xilinx Vivado and functionally verify and validate your design and the implementation steps to upload the design through the Vivado simulation environment. Read through all of the steps before you begin the lab.

B. 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 system's throughput 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

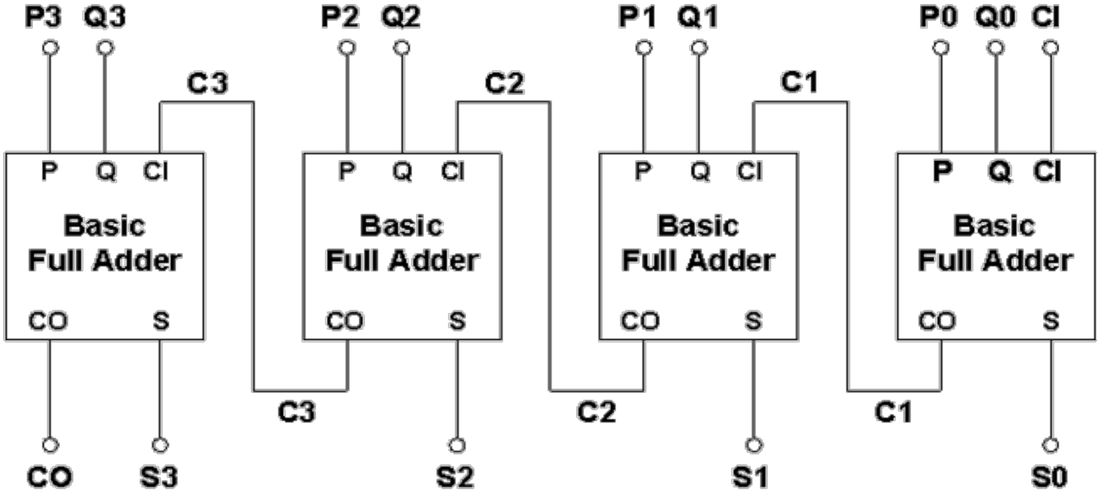

Figure 1. Example: 4-bit parallel adder architecture

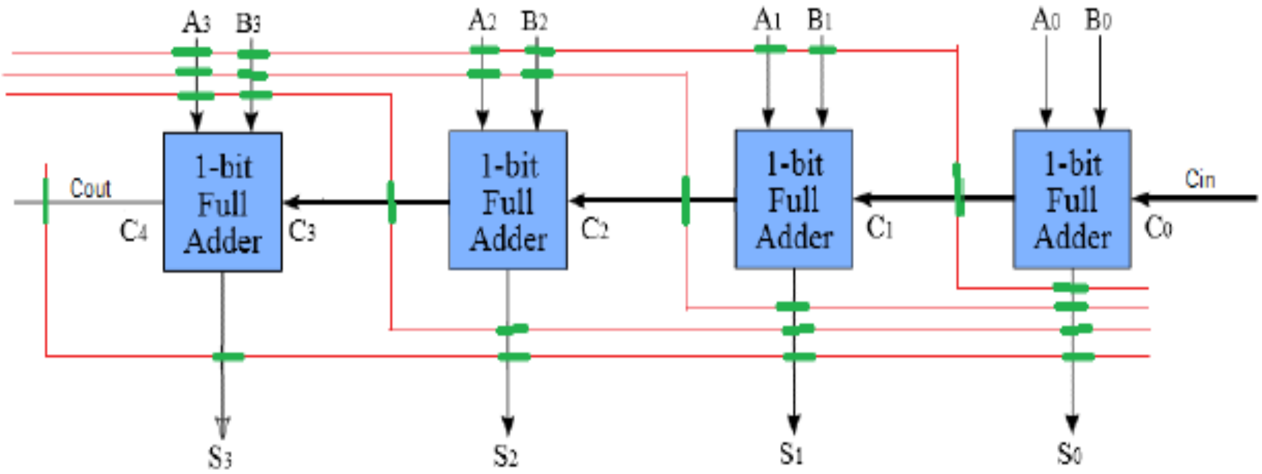

Figure 2. Example: 4-bit synchronous pipeline adder architecture (4 pipelines)

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

8-bit synchronous pipeline parallel adder

In this project, you will create a VHDL project for multiple **8-bit synchronous pipeline parallel adders** using the Xilinx Vivado simulation environment. Use the following components, REG and FA, in the design.

```
entity REG is
   port(D, CLK, RSTn: in std_logic;
         Q: out std_logic);
end entity;

architecture behaviour of REG is
begin
process(CLK)
   begin
      if (CLK'EVENT AND CLK = '1') then
         if(RSTn = '1') then
            Q <= '0' AFTER 10 ns;
           else
            Q <= D AFTER 10 ns;
         end if;
      end if;
   end process;
end architecture;

entity FA is
   port (A, B, Ci: IN STD_LOGIC ;
          S, Co: OUT STD_LOGIC ) ;
end entity;
```

```
architecture behaviour of FA is
   begin
      S  <= A XOR B XOR Ci AFTER 20 ns;
      Co <= (A AND B) OR (Ci AND A) OR (Ci AND B) AFTER 25 ns;
end architecture;
```

C. Project Grading for EE4620L/CEG4324L Students [150 pts for C.1 and C.2] and for EE6620L/CEG6324L Students [200 pts for C.1, C.2 and C.3]

(Submitted to your Pilot LAB Section Dropbox by 11:30 pm, Thursday, June 13, 2024)

C.1 [100 pts]

Turn in a written report that captures your approach to an **8-bit synchronous 8-pipeline parallel adder**. It should include:

1. [10 pts] Pipeline architecture of your 8-bit design, similar to Fig. 1 of 4-bit pipeline adder architecture, except there are **8 pipeline stages**.
2. [40 pts] VHDL code and testbench of your **8-bit synchronous 8-pipeline parallel adder**. Note: Complete the VHDL testbench and determine the minimum clock period to obtain correct pipeline adder results.
3. [40 pts] Your functional verification.

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

   A_{10} = (41, 213)

   B_{10} = (127, 253)

   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. In your testbench, use the sequence 41+127, 41+127+1, 41+253, 41+253+1, 213+127, 213+127+1, 213+253, 213+253+1.

   Note: Submit waveform snapshot of all 8 test cases. 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.
4. [10 pts] The post-synthesis logic design schematic and hardware report.

C.2 [50 pts]

Turn in a written report that captures your approach to an **8-bit synchronous 4-pipeline parallel adder**. It should include:

5. [10 pts] 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, and bit 6 and bit 7 are in one pipeline stage**. So, there are **4 pipeline stages**.
6. [10 pts] VHDL code and testbench of the **8-bit synchronous 4-pipeline parallel adder**. Note: Complete the VHDL testbench and determine the minimum clock period to obtain correct pipeline

adder results.

7. [20 pts] Your functional verification.

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

   A_{10} = (41, 213)

   B_{10} = (127, 253)

   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. In your testbench, use the sequence 41+127, 41+127+1, 41+253, 41+253+1, 213+127, 213+127+1, 213+253, 213+253+1.

   Note: Submit waveform snapshot of all 8 test cases. 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.

8. [10 pts] The post-synthesis logic design schematic and hardware report. Compare hardware and the minimum clock speed of the 8-bit synchronous 8-pipeline parallel adder and the 8-bit synchronous 4-pipeline parallel adder.

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

Turn in a written report that captures your approach to an **8-bit synchronous 2-pipeline parallel adder**. It should include:

9. [10 pts] New pipeline architecture of your design where **bit 0, bit 1, bit 2 and bit 3 are in one pipeline stage**; **bit 4, bit 5, bit 6 and bit 7 are in one pipeline stage**. So, there are **2 pipeline stages.**

10. [10 pts] VHDL code and testbench of the **8-bit synchronous 2-pipeline parallel adder**. Note: Complete the VHDL testbench and determine the minimum clock period to continue obtaining correct pipeline adder results.

11. [20 pts] Your functional verification.

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

    A_{10} = (41, 213)

    B_{10} = (127, 253)

    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. In your testbench, use the sequence 41+127, 41+127+1, 41+253, 41+253+1, 213+127, 213+127+1, 213+253, 213+253+1.

    Note: Submit waveform snapshot of all 8 test cases. 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.

12. [10 pts] The post-synthesis logic design schematic and hardware report. Compare hardware and the

minimum clock speed of the 8-bit synchronous 8-pipeline parallel adder, the 8-bit synchronous 4-pipeline parallel adder, and the 8-bit synchronous 2-pipeline parallel adder.