PROJECT: Implementing Arithmetic device and STA

Introduction

Full adder architecture: Carry-chain forms the critical path in any full adder circuit. Various architectures have been purposed to optimize the carry chain path. Widely used adder architectures are Ripple Carry Adder, Carry Skip Adder, Carry Select Adder, Square Root Carry Select Adder and Parallel Adders. We introduce most of them during the lectures.

In this project, you will be asked to design a pipeline adder and exploring the Synopsys Design Compiler (DC) and the Synopsys Prime Time (PT) to optimize it.

Task 1:

Follow the getting started guide to set up environment and path for DC and PT. Briefly, read through the user manual to understand the basic command and functional option.

Task 2:

1. Fig 1 is a 4 bit pipelined carry select adder using 2-bit full adder as a basic block. The second and third adders are duplicated to calculate the sum of bit 2:3 with carry 0 and 1.

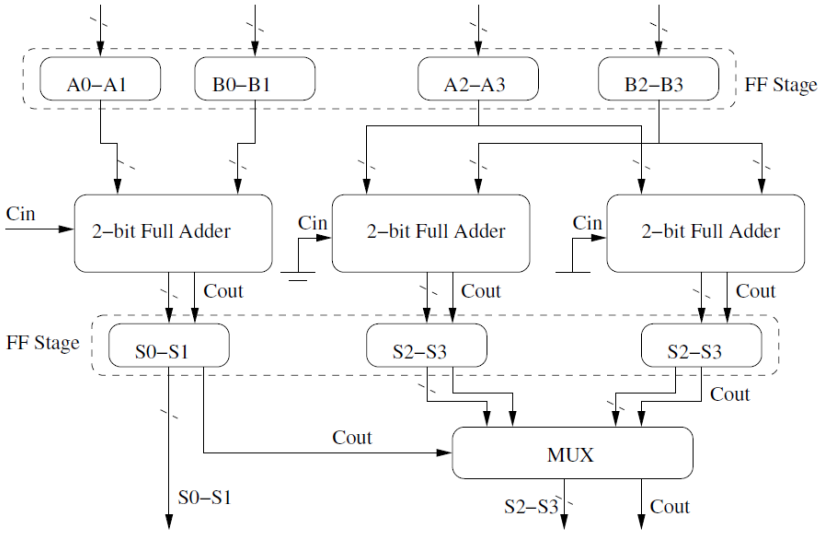

Fig 1

2. Implemented this adder with Verilog and write a simple test bench to show it works correctly.
3. Synthesize it with DC to net-list and figure out what is the maximum delay with DC's report and then calculate the minimum clock period you can have for your adder.
4. Import the net-list to PT for static timing analysis (STA).
   - ✓ Use the following constraints: all input has a delay of 0.6 ns, external clock latency is 0.2 ns, clock skew between first and second level FF is 0.5 ns.
   - ✓ Use the minimum clock period you drive before for constraint and run STA with PT.
5. Check for violation, if there is no violation, you can still try to reduce your clock period until it fails to meet the constraint, then you write down your minimum clock period and calculate the throughput for your pipelined adder.
6. Turn in all the codes, script and timing report.

Task 3:

Implement an 8-bit pipeline adder, you can either expand the carry select adder above or use any adder implementation you want. The minimum requirement is that it has to be pipelined, at least have two stages. The goal is to derive a pipeline adder with the maximum throughput. You are encouraged to explore more optimize functions in DC and PT, and use them to speed up your design.

Requirements for the report:

1. Your result of throughput calculation for both Task 2 and 3.
2. STA report.
3. Explain the process of how you choose your adder and what optimization function you use from EDA tools

Notes (read the note before you work on the project)

1. **Do not distribute the Tools manuals to Non-PSU students.**
2. **Contact TA if you have any question regarding this project.**
3. **In the sample script, DC and prime time are using two different libraries, and it works for that design, but it may cause a problem for your own design, you should set the target library as the same library in both DC and PT. You can use either library.**
4. **When you compile Verilog RTL design into Verilog netlist, it may generate a hierarchical design which RT will have the problem to read. So when you compile your design in DC add follow option "compile -ungroup_all ", then after it generates the net-list Verilog you can check the code and see if there is any sub-module in your design.**
5. **Go over the getting start guide first and try to understand the command in the script by checking the user manual. Not all the command we had in the example script is necessary for your design, you are free to modify it base on your need.**

Honor Pledge

On the cover page of your project report, please type/write down the following sentence in the underlined section below and sign/type your name under it.

"On my honor, I have neither given nor received unauthorized aid on this project."

Do NOT copy and paste the sentence, please do type/write every word in this sentence. Failure to write down this sentence and sign your name or failure to honor the content of this sentence will result in 0 points in the project.

Write the above sentence in underlined section below:

Type or Sign Your Name here:

_____