Recitation 3

Wallace Tree Multiplier

# Introduction

In this recitation, you will first learn about Wallace Tree Multipliers (WTM) from a 4x4 WTM example. Then, you will build your own 5x5 WTM and write a testbench to test it.

Collaboration Policy

You will be working in groups of 2-3. Groups are allowed to collaborate.

# Equipment

* Computer with Quartus Prime software

# Tasks

To receive credit for this recitation, you must complete:

* Task 1: Understand Wallace Tree Multipliers (WTM) from a 4x4 example and then build a 5x5 WTM.
* Task 2: Validate the correctness of your WTM with a testbench.

As you complete tasks, a TA must sign-off on the completion of each task. Ensure that the TA marks the completion of the tasks in Sakai. This recitation acts as a reference for writing and testing Verilog code. You do not need to finish all of the tasks by the end of recitation but you may not leave until either the end of the recitation period or you complete all the provided tasks. If you do not finish and would like to continue working on the tasks outside of recitation, you may go to office hours to receive assistance from TAs.

# Grading

* Completing Recitation Tasks: 1 point (pass/fail)

Wallace Tree Multipliers (WTM)

A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers. A detailed description could be found on [Wikipedia](https://en.wikipedia.org/wiki/Wallace_tree) with an 8x8 (8-bit times 8-bit) example. We are going to use a 4x4 WTM as an example. NOTE: Generally, Wallace Trees sacrifice area for speed.

As shown in the following illustration, a WTM operates in several stages (3 in this case). The first stage is obtained by doing the usual digit-by-digit multiplication (What is equivalent to multiplication for two 1-bit numbers?) we are used to. For an NxN WTM, we would have numbers in Stage 1. Then, at each stage, we use full adders (FA) and half adders (HA) to reduce the number of partial products until reaching the final result. As an example, for bit 2 in Stage 1, we use an FA to add up , and then pass the sum to the next stage. In Stage 2, the sum passed from Stage 1 is added with the carry of bit 1 from Stage 1 in a half adder. Note that there are multiple ways of doing this and this illustration is not the most efficient implementation in case of area. In the following figure, **the “Rectangle” which contains three elements is FA, the “Oval” which contains two elements is HA.**

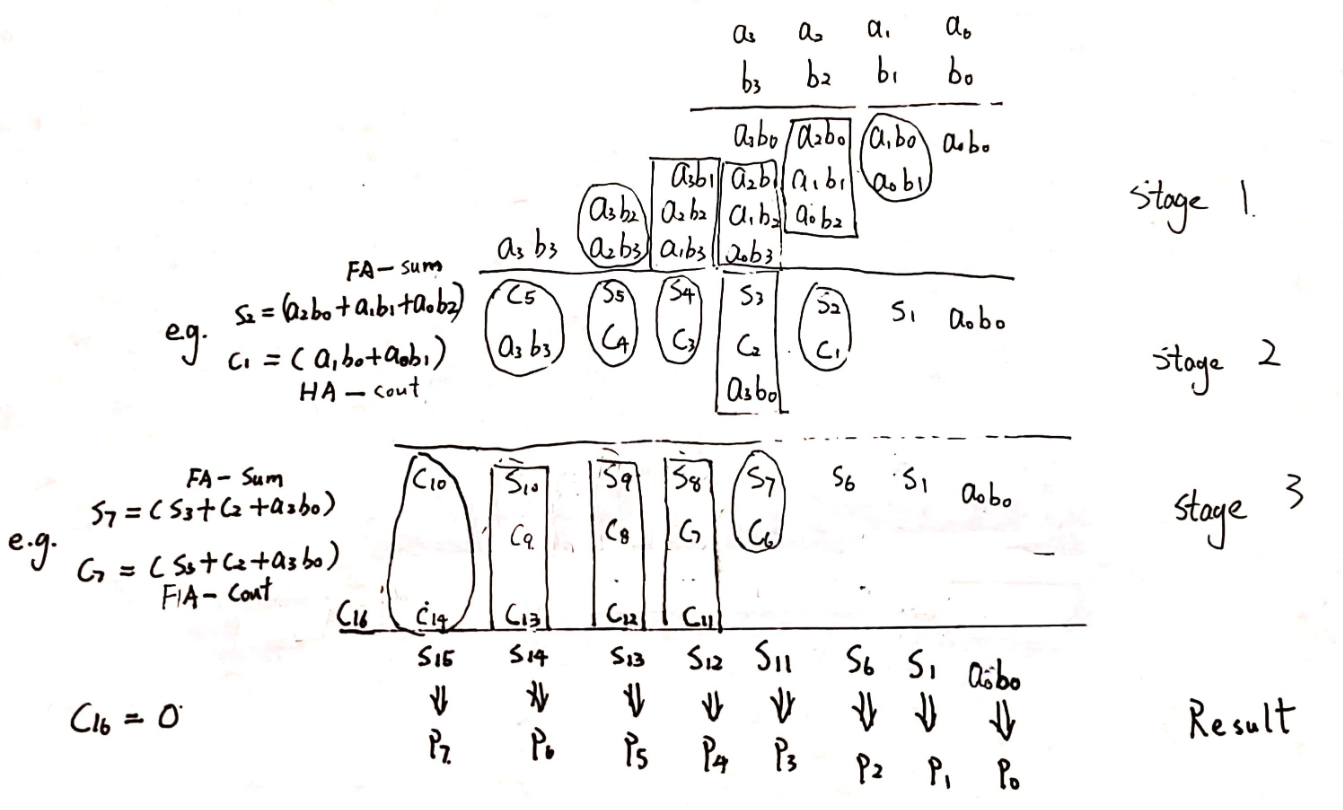
****

Figure 1. Operation of a 4x4 WTM shown in a multiplication table.

An example circuit diagram of a 4x4 WTM is shown in Figure 2 (NOTE: Figure 2 is NOT an implementation of Figure 1.)

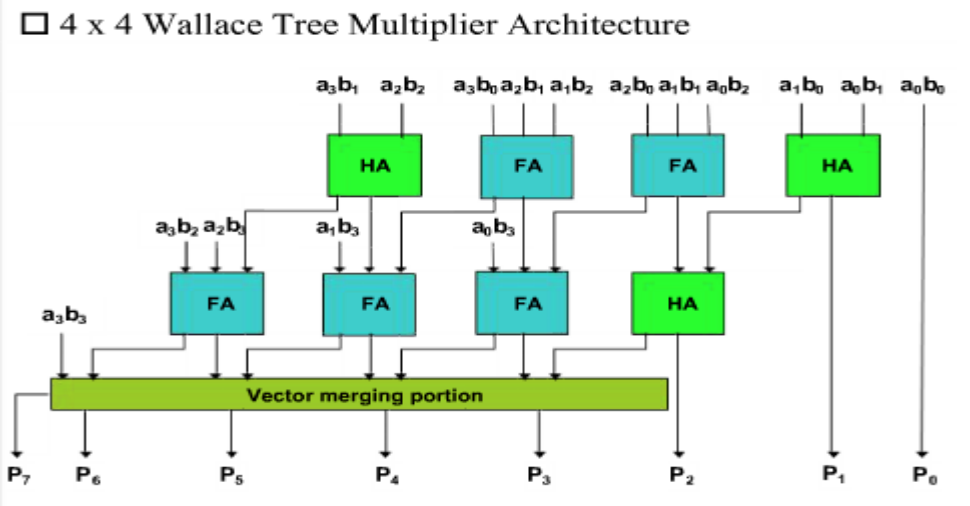


Figure 2. Example circuit diagram of a 4x4 WTM.

Your Wallace Tree Multiplier

Based on what you have learned about 4x4 WTMs, **build a 5x5 WTM**. The output would have 10 bits. Construct a table like the one in Figure 1 first and then build the circuit. You can use your implementation of a full adder in Recitation 2. You can either write your own half adder or use full adders instead.

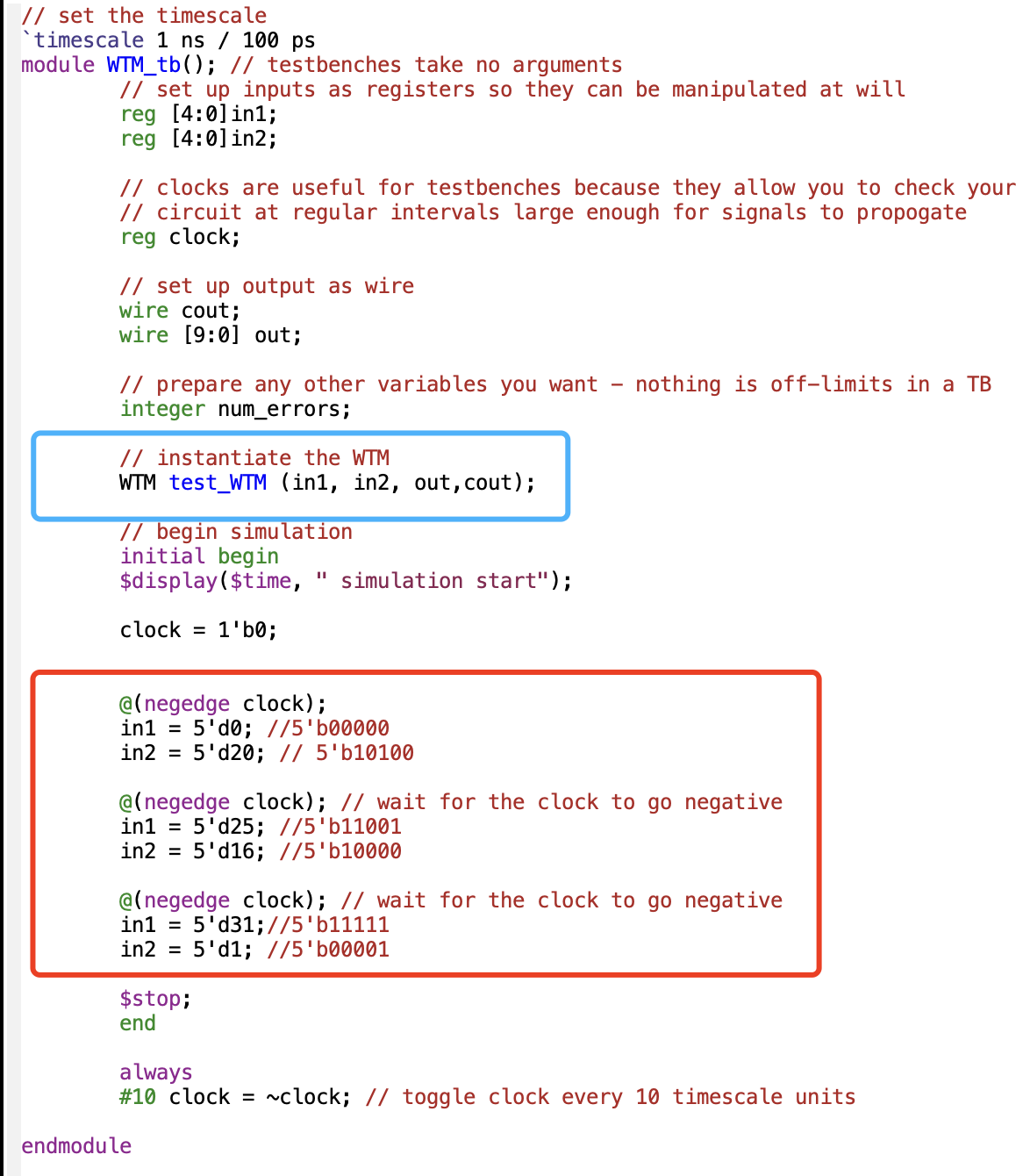
Testing Your Multiplier

Now, write a testbench to test your Wallace Tree. Print out the inputs and outputs in decimal for each test case and show the results to your TA. You must include at least three test cases.

From now on, you will gradually learn how to write testbench files. It’s extremely helpful because it gives you more flexibility when testing for your recitation or project implementations. For this recitation, a version of our testbench is shown in the image below. Please go through the following steps when using it:

1. Understand the meaning of the code before you type it into Quartus.
2. Make sure the line in the blue box matches your design for WTM.
3. Change and/or add more test cases in the redbox.

Of course, if you are comfortable writing your own testbench file, feel free to do so.

****