**ECE 385**

**2021 FA**

**LAB 3**

**Introduction**

**to SystemVerilog, FPGA, CAD, and 16-bit Adders**

Zhou Qishen, Xie Mu

**Introduction**

In this lab, we designed three different adder: ripple carry adder, carry lookahead adder, and carry select adder , and extended 4-bit processor to 8-bit.

**Processor Part**

**Description**

In this lab, we made some changes on the original code to extend the 4-bit processor to 8 bit. The first part modified is all the input, output and register, all changing from 4 bit to 8 bit. The rest thing we need to do is modifying the state machine, to match the iteration number of the register.

**Block Diagram**

**Diagram, schematic

Description automatically generated**

Processor Block Diagram

**Simulation**

Graphical user interface

Description automatically generated with medium confidence

Simulation Result for 8 bit Processor

**Adder Part**

**Carry Ripple Adder**

Carry Ripple Adder is an adder with n full adder implemented in series.

Chart, box and whisker chart

Description automatically generated

**Diagram, schematic

Description automatically generated**

**Carry Lookahead Adder**

The carry-lookahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger-value bits of the adder.

P and G is calculated by formula:

A picture containing chart

Description automatically generated

**Graphical user interface

Description automatically generated**

**Carry Select Adder**

The carry-select adder generally consists of ripple-carry adders and a multiplexer, where adding two n-bit numbers is done with two adders in order to perform the calculation twice, one time with the assumption of the carry-in being zero and the other assuming it will be one.

One adder computes the sum and carry-out based on the assumption that the carry-in is 0, and the other assumes that the carry-in is 1. In this way, both possible outcomes are pre-computed. Once the real carry-in arrives, the corresponding sum and carry-out is selected to be delivered to the next stage.

Diagram, engineering drawing

Description automatically generated

**Diagram, schematic

Description automatically generated**

**Diagram, schematic

Description automatically generated**

**Designing Comparison**

For CRA, it is the simplest adder among three adders above. However, due to the nature of in series, it is quite slow. For SLA, the adder is more complex and more faster by predicting the future state, which means consuming much more energy. Finally, the CSA, it gained limited time improvement, while using twice of the space.

Chart, bar chart

Description automatically generated

**Module Description:**

**Module:​ carry\_lookahead\_adder.sv**

Inputs:​ logic[15:0] A, logic[15:0] B,

Outputs:​ logic[15:0] Sum, logic CO

Description:​ carry lookahead adder.

Purpose:​ carry lookahead adder.

**Module:​ carry\_select\_adder.sv**

Inputs:​ logic[15:0] A, logic[15:0] B,

Outputs:​ logic[15:0] Sum, logic CO

Description:​ carry selected adder.

Purpose:​ carry selected adder.

**Module:​ ripple\_adder.sv**

Inputs:​ logic[15:0] A, logic[15:0] B,

Outputs:​ logic[15:0] Sum, logic CO

Description:​ ripple adder.

Purpose:​ ripple adder

**Post Lab**

Generally speaking, LUT and DFLIP-FLOP match for small scale while memory is suitable for larger scale. LUT is the most simple but will cost most space, since we have to save all its possibility for future calculate. And for memory, it is well organized. Thus, it also will consume significant amount of resources.

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

The carry selected adder consumes the most energy, since it do the most calculation.

**Conclusion**

In conclusion, in this lab, we learnt how to compose and test System Verilog by expending 4-bit processor to 8 bit and designing three different adders.