Lab ?: ??????????

# Design Explanation

## NAND Implement Lawrence

1. Design Concepts
2. Schematic Graph

## NAND Implement Ariel

## Ripple Adder

## Decode and Execute

## Carry Lookahead Adder

1. Design Concepts

We divide CLA (Carry Lookahead Adder) into 3 modules, adder unit, 4-bit CLA generator, 2-bit CLA generator.

First, the adder unit process inputs into p (propagate) flag and g (generate) flag. Second, 4-bit CLA process p flag and g flag into c (carry) flags. 4-bit CLA would also condense p flags and g flags. Last, 2-bit CLA generator process the condensed p flags and g flags into c flag.

Such procedure could significantly reduce the latency from gates to gates, we will dig into the benefits in following discussion.

1. Schematic Graph
2. How it works?
3. Benefits

A ripple adder is a processing chain. We compute the first bit and carry the overflow to next bit. Next, we compute the second bit and carry the overflow to next bit, on and on.

Since ripple adder is combinatorial circuit, we shall reform the circuit to a DAG (Direction Acyclic Graph) by replacing gates with nodes and wires with edges.

Picking the greatest carry flag as root of the tree, the depth of the tree is if the ripple adder supports -bit addition. That is, it takes gates from input to output for a -bit addition. Since it takes to propagate through all gates in the tree, the latency from input to output is around .

A carry lookahead adder is a flattened processing graph. Picking the greatest carry flag as root, the depth of the tree is only if the CLA supports -bit addition. That is, the depth of the tree is not related to the bits of addition. Despite the propagate latency stays constant in respect to , CLA requires gates. Therefore, we should always consider the space-time trade off on CLAs.

In practice, we divide CLA to smaller CLAs. Smaller CLAs leads to less gates thus less power consumption. In this case, we sacrificed time to consume less space.

In a nutshell, ripple adders require gates and takes to compute. In contrast, CLAs require gates but takes to compute. Also, divide CLA into multiple smaller CLAs could significantly reduce gates in practice.

1. Circuits Explanation

## 4-bit Multipler

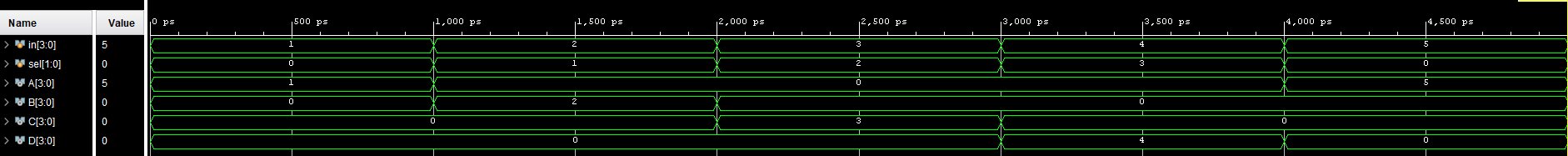
1. How it works?
2. Schematic Graph

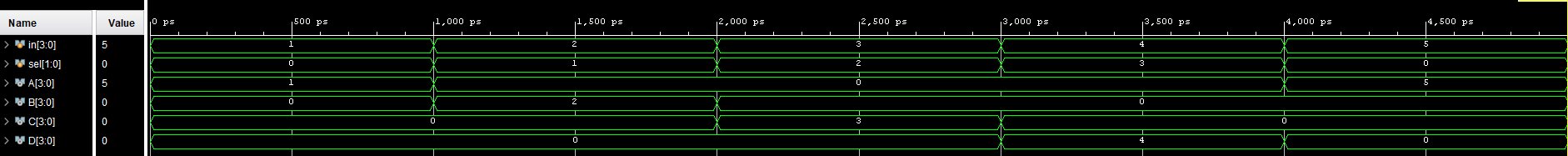
## Exhausted Testbench

# Design Verification

## DMUX

Enumerate all possible inputs, and verify the outputs manually. The pictures below are the representation with verification.





## ??????

????????

# Contribution

## Lawrence Wu

## Ariel Chang

# What have we learned?