

# Introduction to computer architecture

## Carry select adder

**Submitted by:**

Ahmed Khaled 18101381

Saif Mohamed 18101353

Mohamed Abd El wahab 18102714

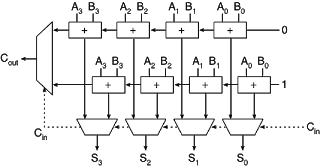
Samar Ibrahim 18102357

### Dr. Ahmed Sultan

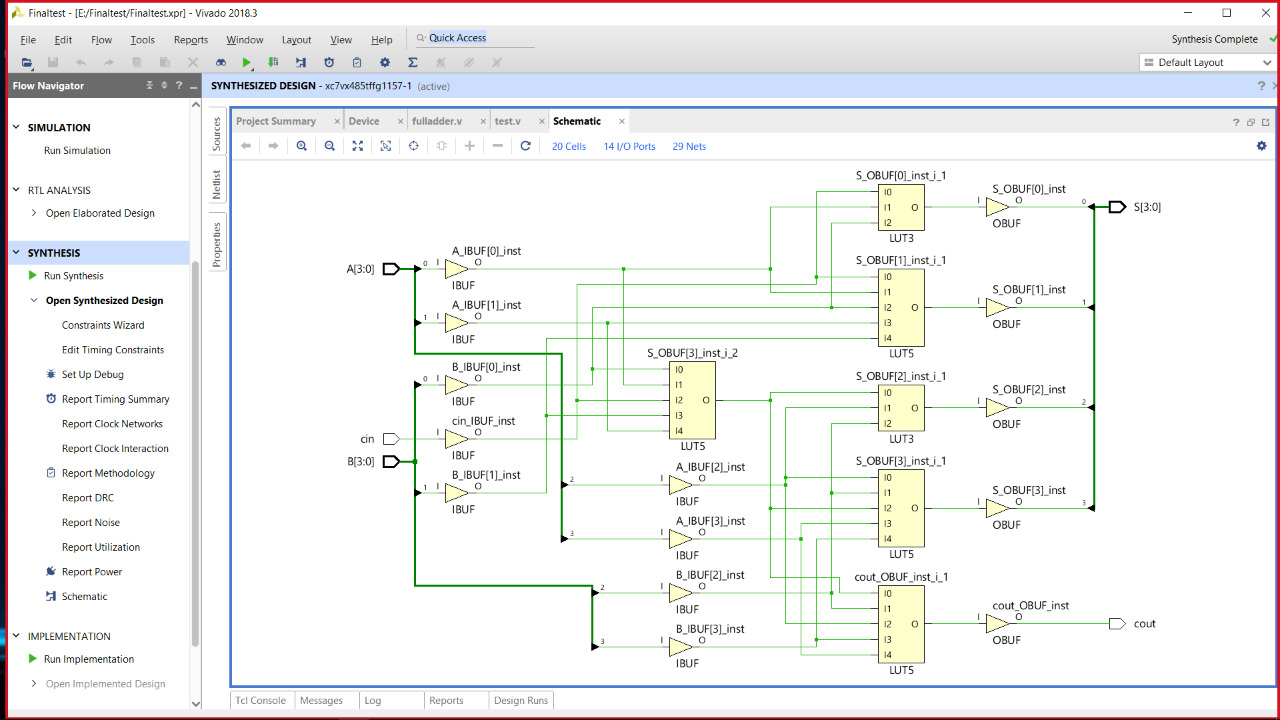
Spring 2021

**Abstract**

The trouble with the carry adder is that the high order bits have to wait for the carry-in from the low order bits, so to improve the latency of our addition circuit, we have to reduce the propagation delay of the circuit. A carry select adder is an arithmetic combinational logic circuit that adds two N-bit binary numbers and outputs their N-bit binary sum and a 1-bit carry. This is no different from a ripple carry adder in function, but in design, the carry select adder does not propagate the carry through as many full adders as the ripple carry adder does. This means that the time to add two numbers should be shorter.



we have two adders in parallel, we computed both answers at the same time, that compute one answer using a carry in od zero and at the same tome compute a different answer with a carry in of one. One of those two will be correct, so they doesn’t wait for the carry out from each other. Then each one can processing immediately. But, how we know which adder’s output should be used.? We need a sub circuit that conceptually acts like a switch (multiplexer).



Schematic of Carry Select Adder

**The propagation delay :**

How many times can we apply this pattern and continue to reduce the running time.? Each time we apply this pattern, the width of the component adders get cut in half. We go from 16 bit ripple carry adder to 8 bit to 4 bit and so on. That is the definition of log base 2 of n (). So if we apply this pattern all log n times. So the circuit will break into a set of independent full adders one bit full adder for each pair of input bits connected to a tree of multiplexers, each full adder just run in constant time, and all of the full adders can run in parallel because they are connected to the a and b inputs, so that section of the adder has a constant propagation delay. Each multiplexer also run in a constant time. But the multiplexers are a part of a tree with a depth of log n, so that tree has a propagation delay that O(log n). thus, the overall logarithmic running time is O(log n).

**Results:**

