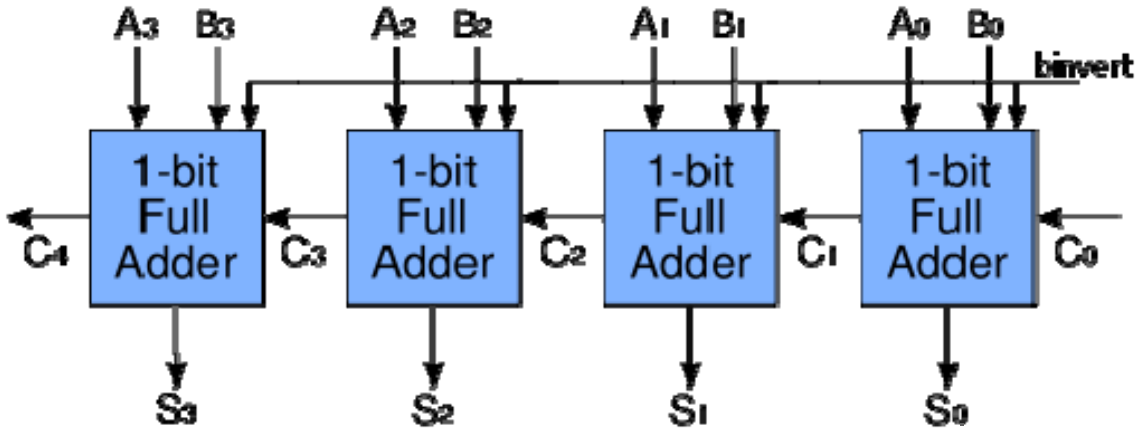
CPRE 381 Lab 5
Digital Arithmetic – Adder/Subtracter Design

In this lab you will construct two variations of an adder/subtracter. The first will be a standard ripple-carry design. The second will be based on a carry-look-ahead unit.

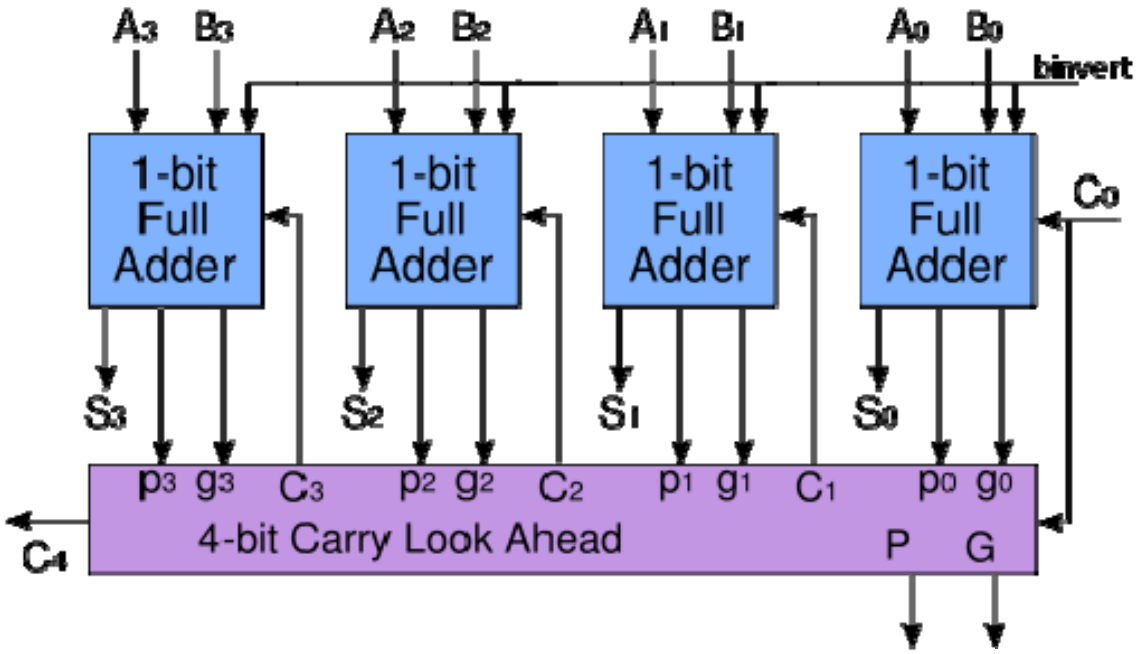
Ripple-Carry



1. Design a 1-bit adder/subtracter block with the following signals, four 1-bit inputs, a, b, carryin, and binvert, and two 1-bit outputs, sum and carryout. The outputs for sum and carryout are simply the addition of a, b and carryin. The binvert when high, inverts b prior to the addition, which will enable subtraction.
2. Implement your design in verilog.
3. Construct a 16-bit adder in verilog from 1-bit blocks by cascading the carryout ports to the carryin ports. (Hint: binvert is used as the carryin for the least significant bit to enable subtraction by converting b to its inverse in two's complement.)
4. Add a 1-bit overflow output signal to indicate whether the operation performed generates an overflow. There are four cases that can cause overflow, positive + positive yields a negative, negative + negative yields a positive, positive – negative yields negative, negative – positive yields positive. (Hint: overflow can be detected in all four cases by comparing the carryin to the MSB to the carryout from the MSB.)
5. Simulate and verify your design by performing several additions and subtractions with combinations of both positive and negative numbers. Also demonstrate all four instances of overflow.

Carry-Look-Ahead

Ripple carry is the most common implementation of an adder/subtracter because of its small area. However, in order to speed up computations and reduce the critical path, alternate arrangements have been designed. One such arrangement is the carry-look-ahead. More details can be found in appendix B38-47 in your book.



1. Design a new 1-bit block to support carry-look-ahead generator signals, p (propagate - a or b is 1) and g (generate – a and b are both 1). The inputs will remain the four 1-bit signals a, b, carryin and binvert. The output signals will be three 1-bit signals sum, p, and g. The p and g signals will be fed to an additional logic unit that will produce the carryin signals instead of relying on ripple-carry.
2. Design a 4-bit carry-look-ahead (CLA) unit that takes a carryin, four p signals, four g signals and produces four carry-out signals and a P (larger propagate – all four p signals are 1) and G (larger generate – one of the four g signals is 1 and it will be propagated to the last carryout).
3. Implement both of your new blocks in verilog.
4. Construct a 16-bit CLA adder/subtracter in two ways. First, create only one layer of CLA where each four bits is tied to a CLA unit and the four units are cascaded together (C4 tied to the C0 of the next unit as in ripple carry). Next also create a two-level CLA 16-bit adder/subtracter using a fifth CLA unit to use the P and G signals from the four 4-bit units to produce the carryin signals for the 4-bit units.
5. Add overflow detection to both of your 16-bit CLA adders.
6. Simulate and verify your two 16-bit CLA adders with the same tests performed on the ripple-carry adder/subtracter.