Conditional Sum Adders: Detailed Implementation

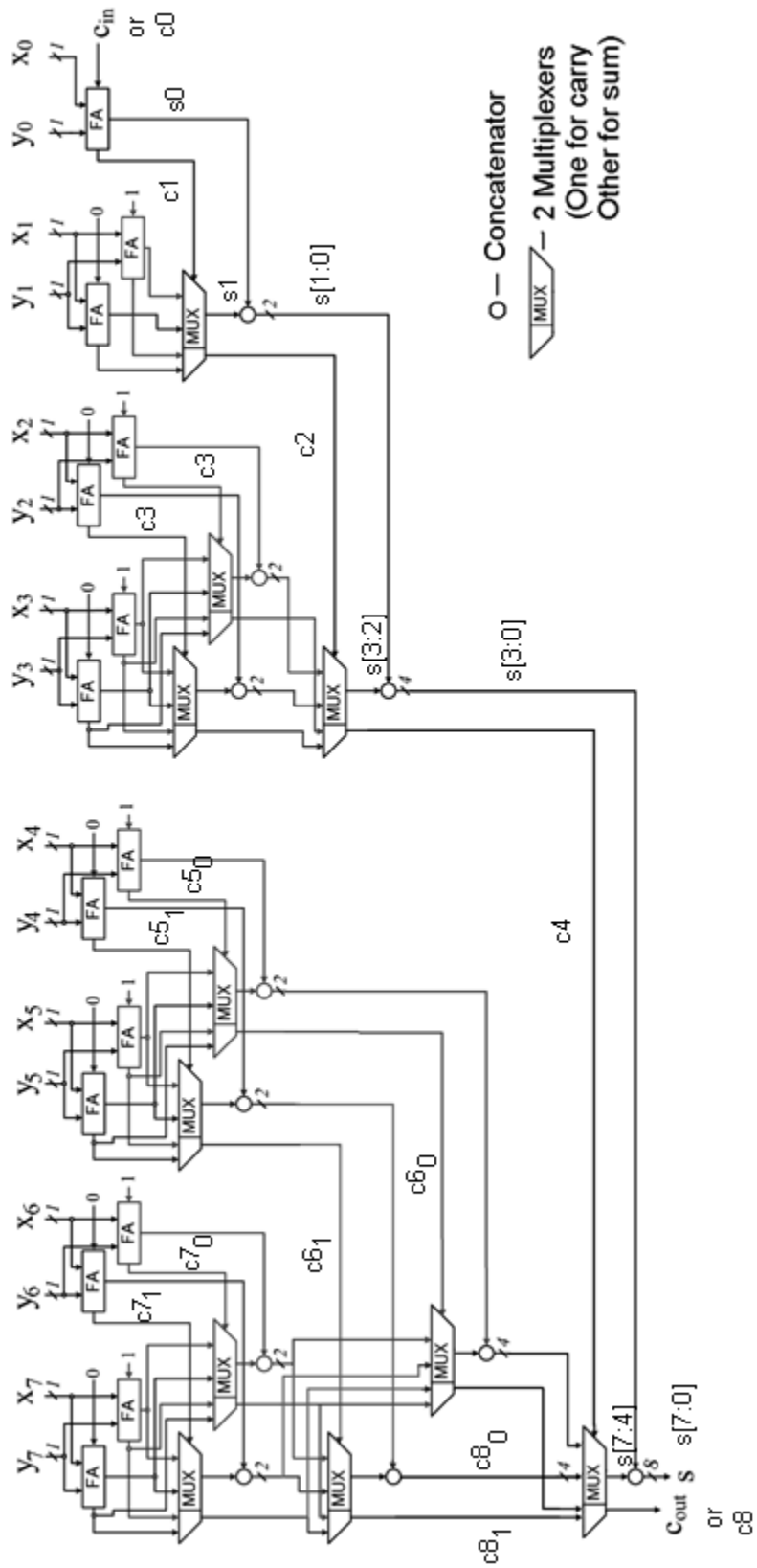
In this handout, we illustrate the functionality of an 8-bit conditional sum adder. The block diagram of 8 bit Conditional Sum Adder is drawn on next page.

The basic idea in the conditional sum adder is to generate two sets of outputs for a given group of operand bits, say, k bits. Each set includes k sum bits and an outgoing carry. One set assumes that the eventual incoming carry will be zero, while the other assumes that it will be one. Once the incoming carry is known, we need only to select the correct set of outputs (out of the two sets) without waiting for the carry to further propagate through the k positions.

In this 8-bit Conditional Sum Adder, the first FA (with x0 and y0 input) takes cin as carry in and generates carry-out. For x1 and y1, we have two sets of adders with carry-in as 0 and 1 respectively. The final sum s1 is the result of multiplex operation between the two sum generated with the select lines as carry-out generated by previous FA. The carryout of second addition is also multiplexed and sent as a select signal for higher bit additions as shown in figure. The concatenator simply groups the incoming bit lines to a wider sum-bus: it is not an actual hardware component, but rather indicates that two fields of input wires will be treated as a combined larger field of output wires.

Advantage: The addition is much faster as higher bit addition operation doesn't rely on carry propagation. However, the selection of correct result waits for the previous carry-out to be set.

Disadvantage: This is a relatively expensive design, as for $N=2^n$-bit addition, it requires 2N-1 full-adders and $2^{n+1} – n – 2$ multiplexers, which are not present in traditional ripple-carry adders.


Concatenator
2 Multiplexers
(One for carry
Other for sum)
MUX
FA
c_{in}
or
c0
x_0
y_0
s0
c1
x_1
y_1
s1
s[1:0]
c2
x_2
y_2
c3
x_3
y_3
s[3:2]
s[3:0]
c4
x_4
y_4
$c5_0$
$c5_1$
x_5
y_5
$c6_0$
x_6
y_6
$c7_0$
$c6_1$
$c7_1$
x_7
y_7
$c8_0$
$c8_1$
s[7:4]
s[7:0]
c_{out}
or
c8