Switching in Full Adder

Consider the selection part of the previous circuit.

The outputs from the AND gates are always either (x, 0) or (0, y).

Then for the OR gate, if one input is 0, the output reflects the other input.

So the output from the OR gate is always either (x) or (y), controlled by whether s (c_{in} in the other example) is 0 or 1.

If s is 0, output is x.

If s is 1, output is y.

This circuit is called a 2–1 line multiplexor (2–1 line MUX).

Where We're Going

Using multiplexors like this, we can create different circuits by connecting inputs to different outputs. This will allow us to create different pathways/instructions in our machine.

How Would We Choose Between 4 Inputs?

Use a 4–1 line multiplexor.

This takes 4 inputs, has 2 selectors (since 2 bits means $2^2 = 4$ possible states), and one output.

These are I_0 to I_3 and S_1 to S_0.

If (S_1, S_0) = (0, 0) then I_0 is output

(0, 1) -> I_1

(1, 0) -> I_2

(1, 1) -> I_3

Note it's by design that the binary numbers input are 0 for I_0 up to 3 for I_3.

For the 4–1 line MUX, you use a 4-input OR gate at the end, which is basically a cascading OR gate where you OR two inputs, then the result with the next input, and so on until finished.
The same can be done with an AND gate.