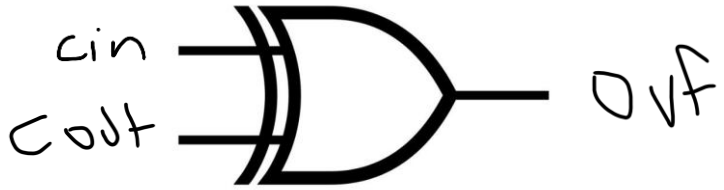
1.

Cin	0	1	1	0
MSB 1	1	0	1	0
MSB 2	1	0	1	0
Output	0	1	1	0
Cout	1	0	1	0
Overflow	Yes	Yes	No	No



There are two cases when overflow occurs (i.e., denoted in red above), both when Cin and Cout are opposites. Hence why the XOR circuit, with Cin and Cout is a good candidate for overflow detection. In other words, overflow really only happens when Cin and Cout are not the same.

2.

Ainvert = 0, Binvert = 0, Carry In = 0, Operation = 2 (adding), Output = A XOR B

3.

To achieve XOR, I added a single AND gate and NOT gate to compute (A∨B)∧¬(A∧B) (red box) using the existing AND and OR operations. The output of this result is wired to the rightmost multiplexer and output when the Op is 11. If you wanted XOR to only be output on the control signal 1110 all you would need to do is wire it up to the corresponding input for a 4 bit selection multiplexer – I have not done that here for simplicity.