INF2270, exercise in combinational logic two's complement arithmetic: example solution

P. Häfliger

March 1, 2010

Task 1

Figure 1 shows one possible implementation of a 4-bit decoder. Figure 2 shows the exact same solution, but with making good use of bus notation for avoiding extensive cabeling. The gates represent 16 gates in parallel. The most significant of these AND gates receives the input i(3),i(2),i(1),i(0), the next i(3),i(2),i(1),_i(0) and so forth down to _i(3),_i(2),_i(1),_i(0). (The underscore '_' indigates the inverse of a variable, since it's not possible to use an over-bar.)

Task 2

Figure 3 shows one possible implementation of a 4-bit comparator *for unsigned binary numbers* and figure 4 for signed (i.e. two's complement) numbers.

The 'equal' comparison per bit is rather straight forward: the XNOR checks if the bits at this position are equal and the output AND requires that also all more significant bits be equal.

The a>b comparison (written as 'agb') checks if a is greater than b for just the local position (first AND gate), then verifies that all more significant bits were equal (second AND), or if a>b has already been determined for a higher

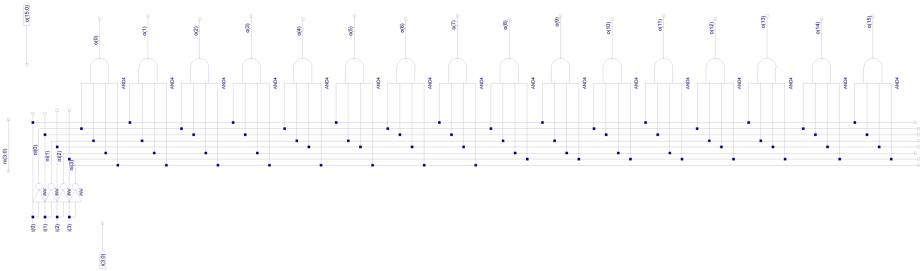

Figure 1: A 4-bit decoder with explicit routing.

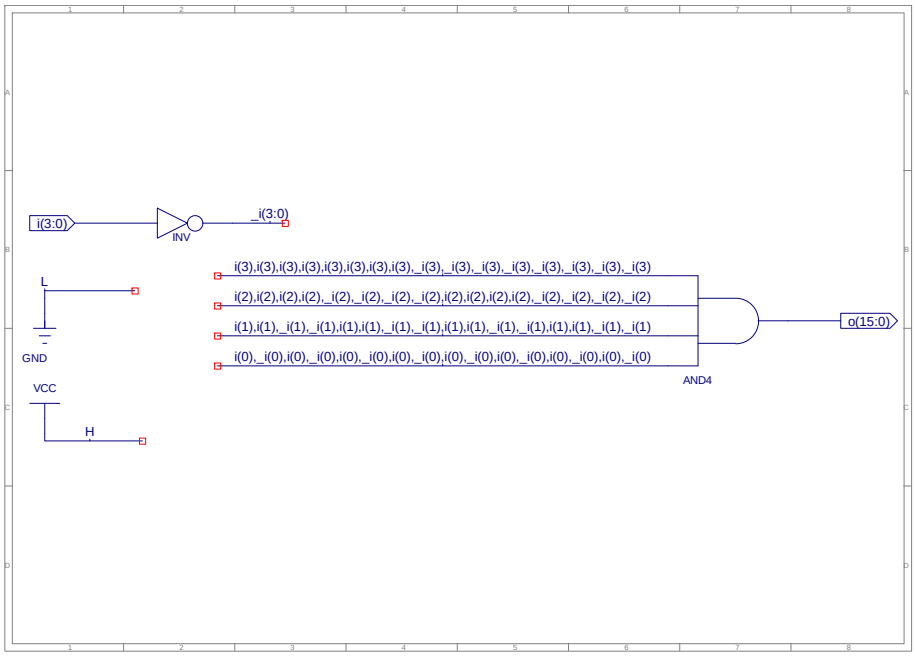

Figure 2: A 4-bit decoder using bus names for the routing. This figure implements the exact same circuit as in figure 1.

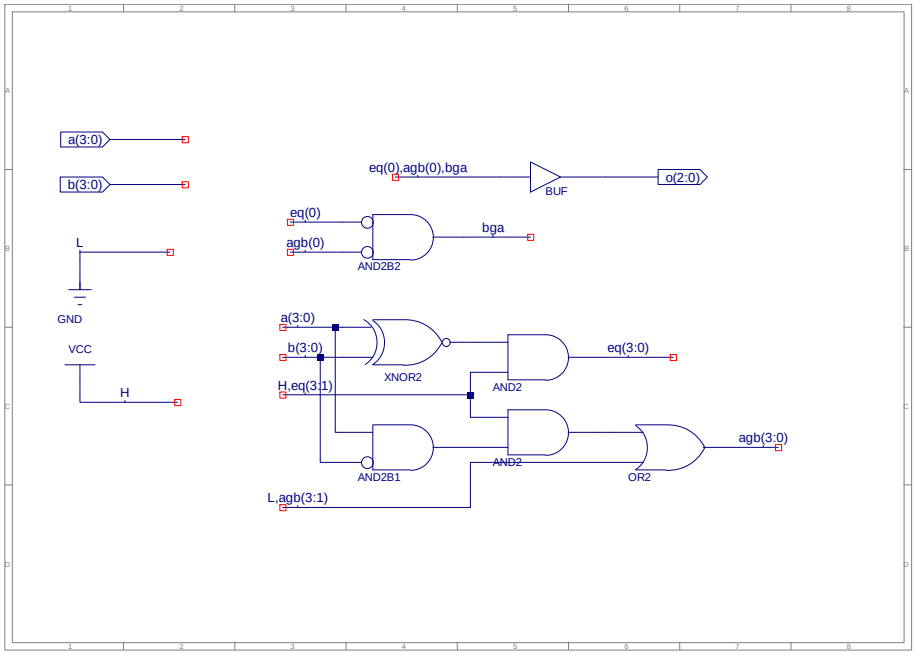

Figure 3: A 4-bit comparator for unsingned binary numbers

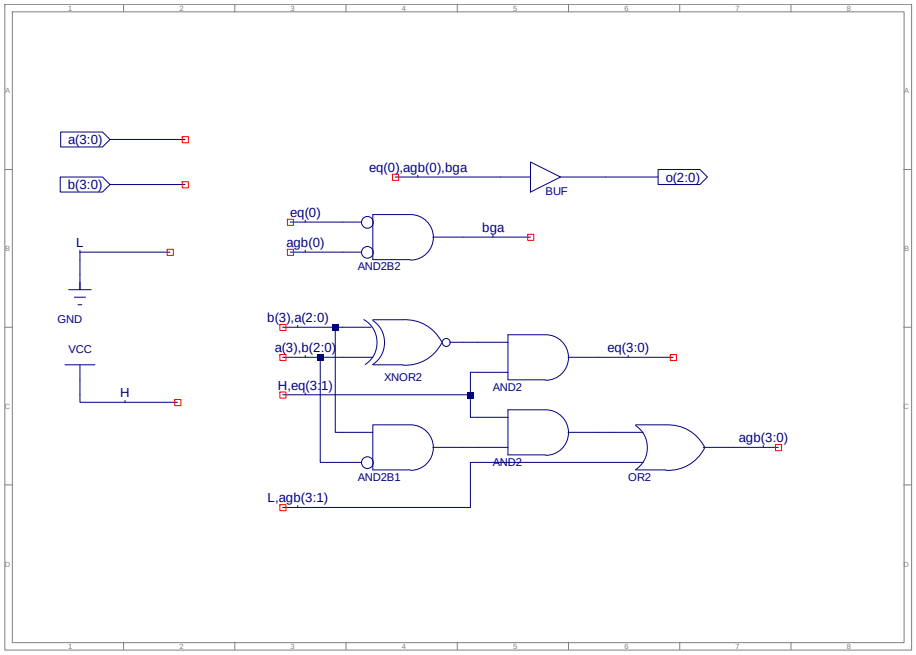

Figure 4: A 4-bit comparator for two's complement binary numbers

order bit (rightmost OR). Note the difference between signed and unsigned here: the MSB is the sign bit. Therefore that number is greater theat has a '0' as MSB if the other has a '1' as MSB. Thus, the MSB 1-bit comparison is reversed as compared to the less significant bits. In figure 4, the MSBs on the input bus are swapped as a consequence. The least siginficant output bits (eq(0) and agb(0)) are the final result.

The a<b (bga) is then simply NOT agb(0) AND NOT eq(0).