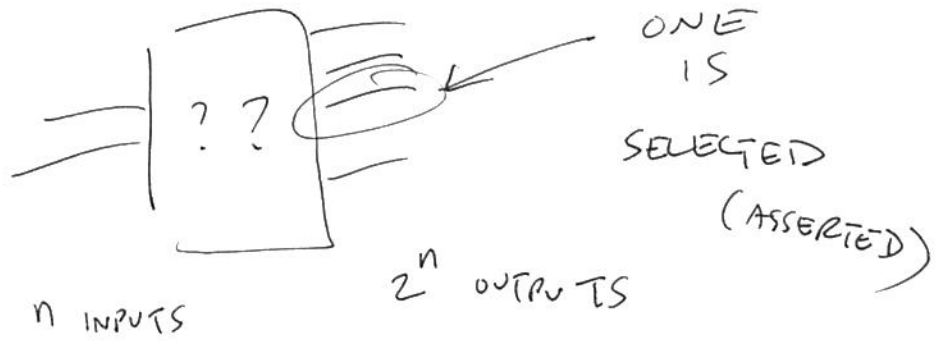

Floating Point

- What's the first stab at representing real numbers? Well, a fractional component (like we saw above) with an integer component
- We still need negative reals! How? (sign bit again)
- We now want to extend the range of our magnitude. How?
- One solution, add more bits
- A better solution -> add an exponent field
- What happens if we treat it as unsigned?
- We introduce a bias so that we get more
- IEEE 754 FP: 1 bit sign, 8 bits exponent, 23 bits mantissa (fraction)
- We assume a leading 1 before the mantissa, but we don't include it (we call this *normalized*)
- The general equation for a FP number is $-1^S * 1.M * 2^{E-127}$ where S is the sign bit, E is the unsigned exponent value, and M is the mantissa

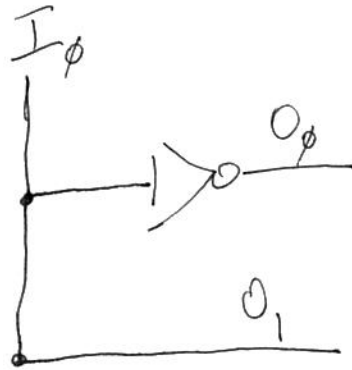
THE DECODER



How?

1-2 LINE DEC.

I_0	I_1	O_0
0	1	0
0	0	1
1	0	0
1	1	1



3-8 DECODER

I_0	I_1	I_2	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7
0	0	0	1							
0	0	1		1						
0	1	0			1					
0	1	1				1				
1	0	0					1			
1	0	1						1		
1	1	0							1	
1	1	1								1

