Transcript of the video “CMOSfunctiondesign”

In this problem, we are given a function Z equal to ((A OR B) AND C), the whole thing negated.

We are then asked to produce a CMOS circuit that implements the function Z.

We know that our CMOS circuits consists of a pull-down circuit made up purely of NFETs, and it is called pulldown because if it is on, it connects the output to ground to produce a low (or 0) output.

We also have a pull-up circuit made up purely of PFETs, and it is called a pull-up because if it is on, it connects the output to VDD to produce a high (or 1) output.

If Z = NOT ((A OR B) AND C), then NOT (Z) = ((A OR B) AND C).

To draw the pull-down portion of this circuit, we take a look at when the function produces Z = 0. This occurs when ((A or B) and C) equals 1, so the pull-down circuitry should be on when ((A or B) and C) equals 1. So (A OR B) = 1 and C = 1.

This means that we want a parallel (A, B) in series with C circuit for our pull-down.

The parallel (A, B) corresponds to (A OR B) = 1 because if either A or B equals 1 then we have a path from Z to the bottom of the parallel circuitry. Then if C is also 1, we complete our path between Z and ground.

So, if either (A = 1 and C = 1) or (B = 1 and C = 1), Z is pulled down to ground and produces a 0 output.

To generate our pull-up, we simply replace parallel circuits with series, and series with parallel. This will ensure that whenever our pull-down circuit is off, our pull-up circuit is on. So our pull-up circuitry is series (A, B) in parallel with C.

To convince ourselves that this is in fact the correct pull-up circuit, we know that the pull-up must make Z equal to 1.

Z = 1 if ((A OR B) AND C) the whole thing negated equals 1, or ((A OR B) AND C) = 0.

This is true when either (A OR B) = 0 or C = 0. This means that either A = 0 and B = 0 OR just C = 0.

Pull-up circuits use PFETs which are on when the input is low (or 0). So the equivalent circuit for this is series (A, B) ORed with C which is equal to series (A,B) in parallel with C.