

## Transistor-level NCL Design

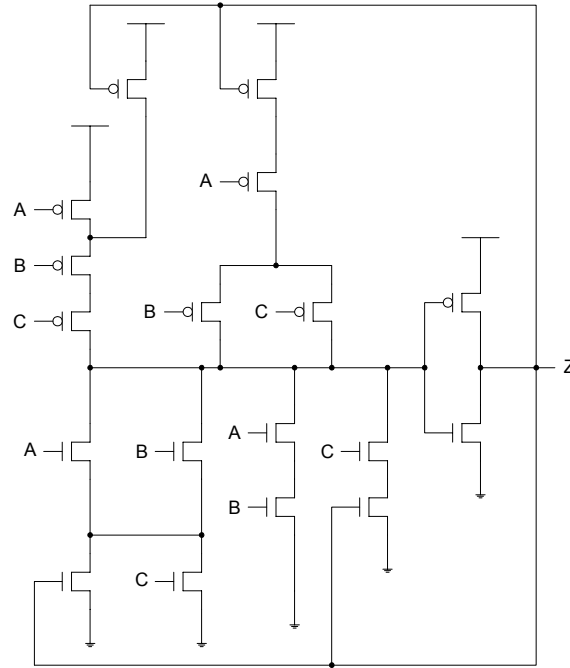
As explained in the Intro to NCL module, NCL threshold gates are designed with *hysteresis* state-holding capability, such that after the output is asserted, all inputs must be deasserted before the output will be deasserted. Therefore, NCL gates have both *set* and *hold* equations, where the *set* equation determines when the gate will become asserted and the *hold* equation determines when the gate will remain asserted once it has been asserted. The *set* equation determines the gate's functionality as one of the 27 NCL gates, as listed in Table 1, whereas the *hold* equation is the same for all NCL gates, and is simply all inputs ORed together. The general equation for an NCL gate with output  $Z$  is:  $Z = \text{set} + (Z \bullet \text{hold})$ , where  $Z$  is the previous output value and  $Z$  is the new value. Take the TH23 gate for example. The *set* equation is  $AB + AC + BC$ , as given in Table 1, and the *hold* equation is  $A + B + C$ ; therefore the gate is asserted when at least 2 inputs are asserted and it then remains asserted until all inputs are deasserted.

**Table 1. 27 fundamental NCL gates.**

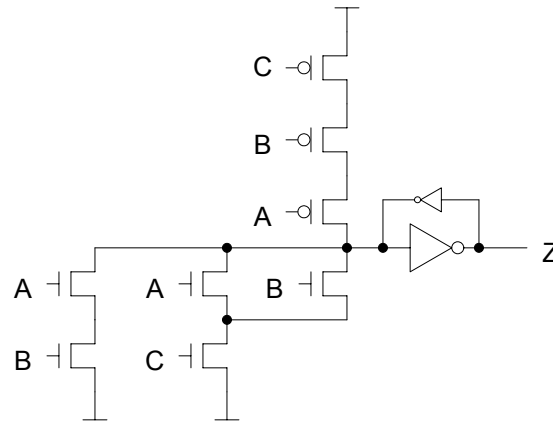
NCL Gate	Boolean Function	Transistor Count (static)	Transistor Count (semi-static)
TH12	$A + B$	6	6
TH22	$AB$	12	8
TH13	$A + B + C$	8	8
TH23	$AB + AC + BC$	18	12
TH33	$ABC$	16	10
TH23w2	$A + BC$	14	10
TH33w2	$AB + AC$	14	10
TH14	$A + B + C + D$	10	10
TH24	$AB + AC + AD + BC + BD + CD$	26	16
TH34	$ABC + ABD + ACD + BCD$	24	16
TH44	$ABCD$	20	12
TH24w2	$A + BC + BD + CD$	20	14
TH34w2	$AB + AC + AD + BCD$	22	15
TH44w2	$ABC + ABD + ACD$	23	15
TH34w3	$A + BCD$	18	12
TH44w3	$AB + AC + AD$	16	12
TH24w22	$A + B + CD$	16	12
TH34w22	$AB + AC + AD + BC + BD$	22	14
TH44w22	$AB + ACD + BCD$	22	14
TH54w22	$ABC + ABD$	18	12
TH34w32	$A + BC + BD$	17	12
TH54w32	$AB + ACD$	20	12
TH44w322	$AB + AC + AD + BC$	20	14
TH54w322	$AB + AC + BCD$	21	14
THxor0	$AB + CD$	20	12
THand0	$AB + BC + AD$	19	13
TH24comp	$AC + BC + AD + BD$	18	12

To implement an NCL gate using CMOS technology, an equation for the complement of  $Z$  is also required, which in general form is:  $Z' = \text{reset} + (Z' \bullet \text{set}')$ , where *reset* is the complement of *hold* (i.e., the complement of each input, ANDed together), such that the gate is deasserted when all inputs are deasserted and remains deasserted while the gate's *set* condition is false. For the TH23 gate, the *reset* equation is  $A'B'C'$  and the simplified *set'* equation is  $A'B' + B'C' + A'C'$ . Directly implementing these equations for  $Z$  and  $Z'$ , after simplification, yields the

static transistor-level implementation of an NCL gate, as shown in Figure 1 for the TH23 gate. This requires the output,  $Z$ , to be feedback as an input to the NMOS and PMOS logic to achieve hysteresis behavior. NCL gates can also be implemented in a semi-static fashion, where a weak feedback inverter is used to achieve hysteresis behavior, which only requires the *set* and *reset* equations to be implemented in the NMOS and PMOS logic, respectively. The semi-static TH23 gate is shown in Figure 2. In general, the semi-static implementation requires fewer transistors, but is slightly slower because of the weak inverter. Note that TH1n gates are simply OR gates and do not require any feedback, such that their static and semi-static implementations are exactly the same.



**Figure 1. Static CMOS implementation of a TH23 gate:  $Z = AB + AC + BC$ .**



**Figure 2. Semi-static CMOS implementation of a TH23 gate:  $Z = AB + AC + BC$ .**