Input-Completeness and Observability

Input-Completeness requires that all outputs of a combinational circuit may not transition from NULL to DATA until all inputs have transitioned from NULL to DATA, and that all outputs of a combinational circuit may not transition from DATA to NULL until all inputs have transitioned from DATA to NULL. In circuits with multiple outputs, it is acceptable according to Seitz's "weak conditions" of delay-insensitive signaling, for some of the outputs to transition without having a complete input set present, as long as all outputs cannot transition before all inputs arrive. For example, the NCL AND function in Figure 1 is not input-complete because the output, *Z*, will transition to DATA0 if either input is DATA0, even if the other input is NULL. However, the half-adder in Figure 2 is input-complete, even though C_{out} is not input-complete, because both inputs must be DATA in order for *S* to transition to DATA, such that the entire output set, $\{S, C_{out}\}$, cannot transition to DATA until both inputs transition to DATA. The hysteresis within each NCL gate ensures that all inputs must transition to NULL before a combinational circuit's output will transition to NULL, making the circuit input-complete with respect to NULL, assuming that the circuit is input-complete with respect to DATA.

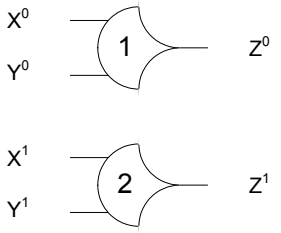

Figure 1. Input-incomplete NCL AND function.

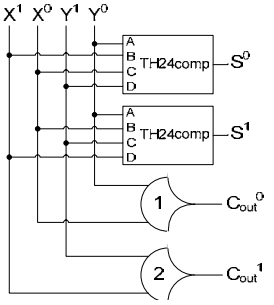

Figure 2. NCL half-adder.

To determine if a circuit is input complete, one must analyze the equation for each output. An output is input-complete with respect to a particular input iff every non-don't care product term in the output's equation (i.e. equations for all rails of the output) contains any of the rails of the particular input. Take Figure 3 for example. The output equations are as follows: $X^0 = B^0C^1 + B^0B^1 + C^0C^1 + C^0B^1$; $X^1 = A^1B^1A^0 + A^0B^0 + A^1B^1C^1$; $Y^0 = A^0C^0 + A^1C^1$; $Y^1 = A^1B^1C^0 + A^1B^1B^0 + C^1C^0 + C^1B^0$. Removing the don't cares, where two rails of the same signal are both asserted (i.e. both rails can never be simultaneously asserted; they are mutually exclusive), yields the following equations: $X^0 = B^0C^1 + C^0B^1$; $X^1 = A^0B^0 + A^1B^1C^1$; $Y^0 = A^0C^0 + A^1C^1$; $Y^1 = A^1B^1C^0 + C^1B^0$. *X* has a *B* in each product term, so it is input-complete with respect to *B*. Likewise, *Y* has a *C* in each product term, so it is input-complete with respect to *C*. To make the circuit input-complete with respect to *A*, *A* must be added to all product terms in which it is missing in either *X* or *Y*, but not both. Since *A* is only missing in one of the *Y* product terms, it is added here, resulting in the following equation: $Y^1 = A^1B^1C^0 + C^1B^0(A^1 + A^0) = A^1B^1C^0 + C^1B^0A^1 + C^1B^0A^0$. However, since Y^0 contains an A^1C^1 product term, the new $C^1B^0A^1$ product term in Y^1 must have been a don't care in the original expression, since both Y^0 and Y^1 cannot be simultaneously asserted; therefore, Y^1 can be simplified as follows: $Y^1 = A^1B^1C^0 + C^1B^0A^0$. The input-complete circuit can then be redrawn as shown in Figure 4.

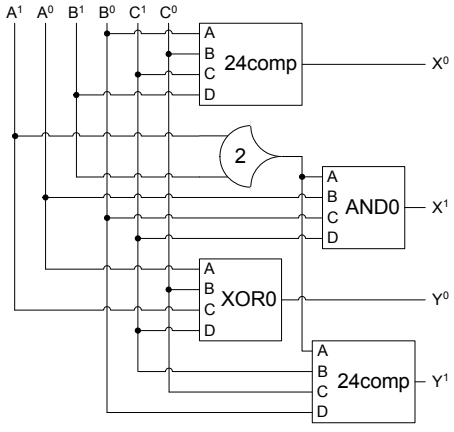

Figure 3. NCL circuit that's input-incomplete with respect to *A*.

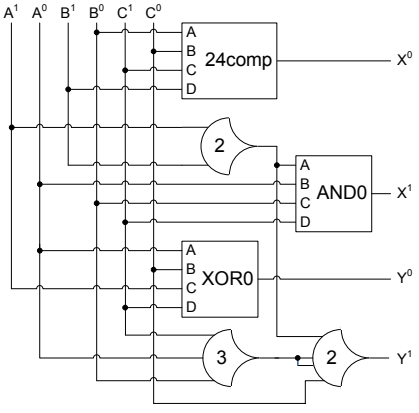

Figure 4. Input-complete NCL circuit.

Observability requires that no *orphans* may propagate through a gate. An orphan is defined as a wire that transitions during the current DATA wavefront, but is not used in the determination of the output. Orphans are caused by wire forks and can be neglected through the isochronic fork assumption (i.e. gate delays are much longer than wire delays within a component), as long as they are not allowed to cross a gate boundary. This *observability* condition, also referred to as indicatability or stability, ensures that every gate transition is observable at the output; which means that every gate that transitions is necessary to transition at least one of the outputs. Consider an unobservable version of an XOR function, shown in Figure 5, where an orphan is allowed to pass through the TH12 gate. For instance, when X=DATA0 and Y=DATA0, the TH12 gate is asserted, but does not take part in the determination of the output, Z=DATA0. This orphan path is shown in boldface in Figure 5. The equation for Z^l can be repartitioned to obtain a fully observable version of the XOR function, as shown in Figure 6. Here, the two internal TH22 gates are each connected to a TH23W2 output gate with a weight of 2, which is the same as the threshold, such that if either internal gate is asserted, its corresponding output gate will always become asserted. Note that this circuit is for example only, since the XOR function can be simplified to two TH24comp gates.

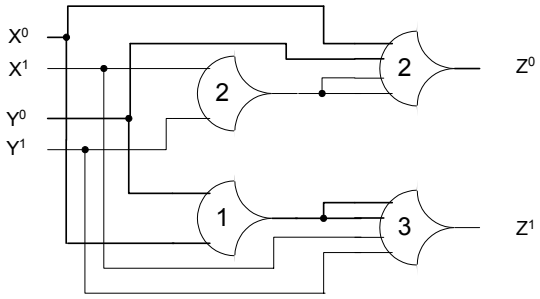

Figure 5. Unobservable NCL XOR function.

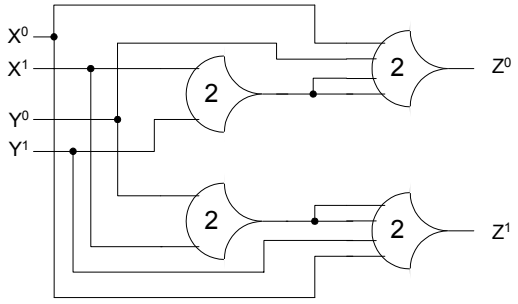

Figure 6. Observable NCL XOR function.

The best way to ensure that a circuit is observable is to not divide product terms when mapping equations to their corresponding gate-level circuits. This however is not required for a circuit to be observable, and is not always possible, for example when a product term contains more than four variables. The circuit in Figure 4 is observable even though a product term has been divided. The TH33 gate is observable because its output has the same weight as the output gate's threshold, similar to the previous example. This is not the case for the TH22 gate, so it must be analyzed more closely. The equation for the TH22 gate is A^1B^1; and its output is used in the X^l and Y^l product terms: $A^1B^1C^1$ and $A^1B^1C^0$, respectively. Therefore, if the TH22 gate is asserted, it will always cause either X^l or Y^l to become asserted because *C* must either be DATA0 or DATA1; hence the TH22 gate is observable.