

Quad-Rail Combinational Circuit Design

The design process for NCL combinational circuits is similar to Boolean circuits, where a Karnaugh map, or other simplification technique, can be utilized to determine the simplified sum-of-product (SOP) expressions for each output. As in dual-rail optimization, a Karnaugh map can be constructed for each output, but instead of only $0s$ and $1s$, corresponding to a signal's $rail^0$ and $rail^1$, respectively, the K-map also contains $2s$ and $3s$, which correspond to a signal's $rail^2$ and $rail^3$, respectively. The 0 outputs are then grouped together to obtain a minimized expression for $rail^0$; the 1 outputs are grouped together to obtain a minimized expression for $rail^1$; the 2 outputs are grouped together to obtain a minimized expression for $rail^2$; and the 3 outputs are grouped together to obtain a minimized expression for $rail^3$. After expressions for the outputs have been obtained, an assessment must be made to ensure that the circuit is input-complete. If not, the missing input(s) must be added to the appropriate product term(s). The output equations must then be partitioned into sets of four or fewer variables to be mapped to the 27 NCL gates, while ensuring that the resulting circuit is observable. To minimize area and delay, partitioning should be performed such that the minimal number of sets is obtained, which will occur when the maximum number of product terms are grouped into each set.

Take for example the design of a quad-rail partial product (PP) generation component, depicted in Figure 1, for use in an unsigned quad-rail multiplier. Remember that each quad-rail signal corresponds to 2 bits, such that the quad-rail partial product (PP) generation component is equivalent to 2 bits \times 2 bits, which yields a 4-bit result, and hence 2 quad-rail signals, PPH and PPL . Figure 2 shows the Karnaugh maps for this component, along with the optimal coverings. Note that only 4-coverings can be utilized to eliminate a quad-rail signal from the corresponding product term; 2-coverings will not do so, and are therefore not used. Because of this, the input order does not need to be rearranged like required for Boolean and dual-rail K-maps (i.e., 0, 1, 2, 3 for quad-rail vs. 00, 01, 11, 10 for Boolean and dual-rail). Also note that 3 does not appear as an output in the PPH K-map. This is because the maximum value of PPH is 2, resulting when A and B are both 3 (i.e., $3 \times 3 = 9_{10} = 1001_2 = 21_4$); therefore PPH^3 is always 0, and can be treated as a don't care in subsequent circuits that use PPH as an input. The minimal SOP equations are derived directly from the K-map coverings as follows:

$$PPH^0 = A^0 + A^1 + B^0 + B^1 \rightarrow \text{TH14}$$

$$PPH^1 = A^2B^2 + A^2B^3 + A^3B^2 \rightarrow \text{THand0}$$

$$PPH^2 = A^3B^3 \rightarrow \text{TH22}$$

$$PPH^3 = 0$$

$$PPL^0 = A^0 + B^0 + A^2B^2 = A^0 \cdot (B^0 + B^1 + B^2 + B^3) + B^0 \cdot (A^0 + A^1 + A^2 + A^3) + A^2B^2$$

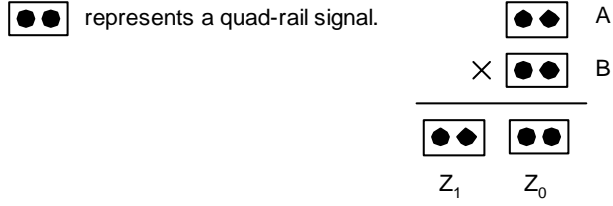
$$= A^0B^3 + A^0B^1 + A^3B^0 + A^1B^0 + A^2B^2 + A^2B^0 + A^0B^2 + A^0B^0$$

$$PPL^1 = A^1B^1 + A^3B^3 = A^1B^1 + A^3B^3 + A^1A^3 + B^1B^3 \rightarrow \text{TH24comp}$$

$$PPL^2 = A^2B^1 + A^2B^3 + A^1B^2 + A^3B^2$$

$$PPL^3 = A^1B^3 + A^3B^1 = A^1B^3 + A^3B^1 + A^1A^3 + B^1B^3 \rightarrow \text{TH24comp}$$

Since all product terms in either PPL or PPH do not contain either an A or B , the resulting circuit is not input-complete with respect to either input; therefore additional terms must be added. Specifically, additional terms must be added to either PPL^0 or PPH^0 , since these are the input-incomplete rails of PPL and PPH , respectively. Making PPL^0 input-complete would require fewer additional terms and would not increase PPL 's worst-case delay, whereas making PPH^0 input-complete would increase PPH 's worst-case delay from 1 gate to 2 gates; therefore additional terms were added to PPL to make it input-complete with respect to both A and B , as shown above. The equations for PPH each directly map to an NCL gate, as shown above. The equations for PPL^1 and PPL^3 each map to TH24comp gates after adding two *don't care* terms, representing the cases when two rails of either A or B are simultaneously asserted, as shown above, resulting in two fewer transistors for implementing each rail (when using static gates). The equation for PPL^0 can be partitioned into one set of 4 variables and two sets of 3 variables, as highlighted above. The yellow and green terms each map to a TH33w2 gate; and the blue term maps to a TH24comp gate. The equation for PPL^2 can be partitioned into one set of 3 variables and a second set of 4 variables that contains the output of the first set as one input, as highlighted above. The red text maps to a TH33w2 gate; and the yellow term maps to a TH33w32 gate. Since no product terms were divided, the circuit is observable. The resulting optimized circuit is shown in Figure 3.



Example: $A = 2; B = 3 \rightarrow Z_1 = 1; Z_0 = 2$

Figure 1. Quad-rail PP generation component.

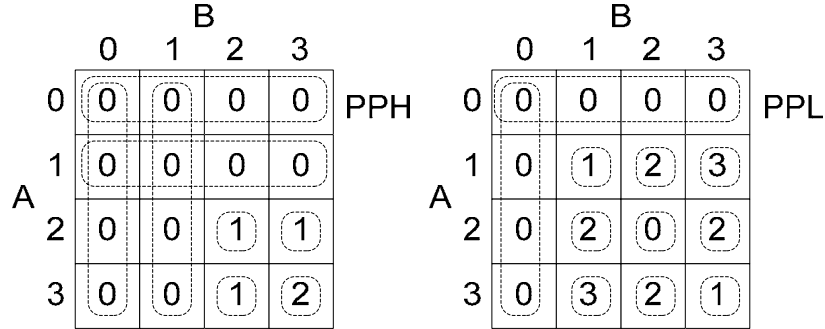


Figure 2. K-maps for quad-rail PP generation component.

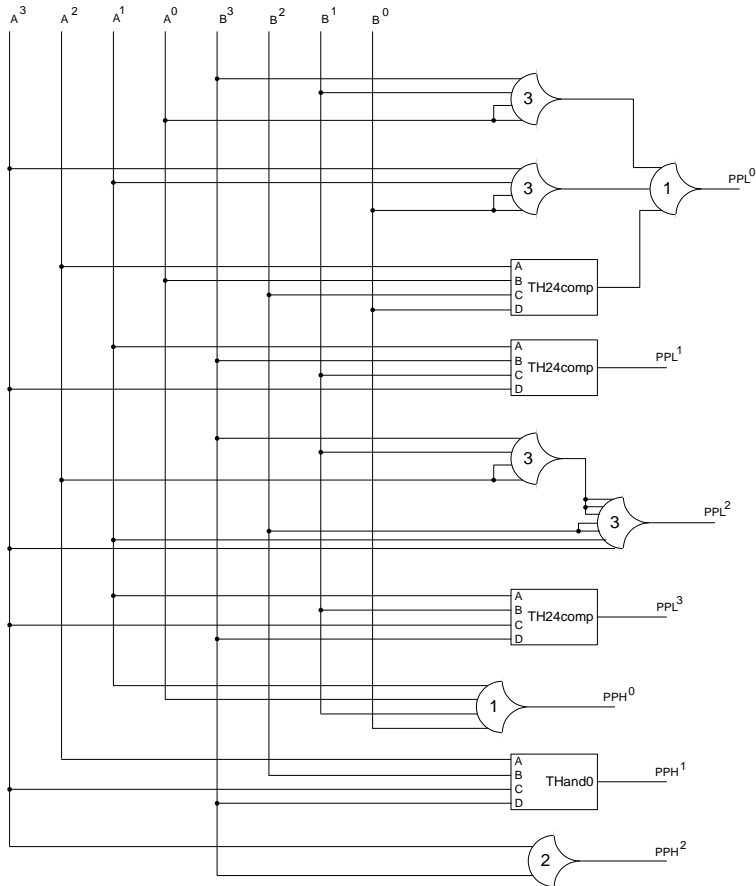


Figure 3. Optimized quad-rail PP generation component.

Now let's consider the design of the increment circuitry for the 4-bit counter shown in Figure 4. The specifications for this counter included a full NCL interface with request and acknowledge signals labeled K_i and K_o , respectively. Functionality was specified to reset *count* to 0000₂ when the *reset* signal is applied, to increment *count* by 1 when *inc* = 1, and to keep *count* the same when *inc* = 0. The counter will rollover to 0000₂ when *count* = 1111₂ and *inc* = 1.

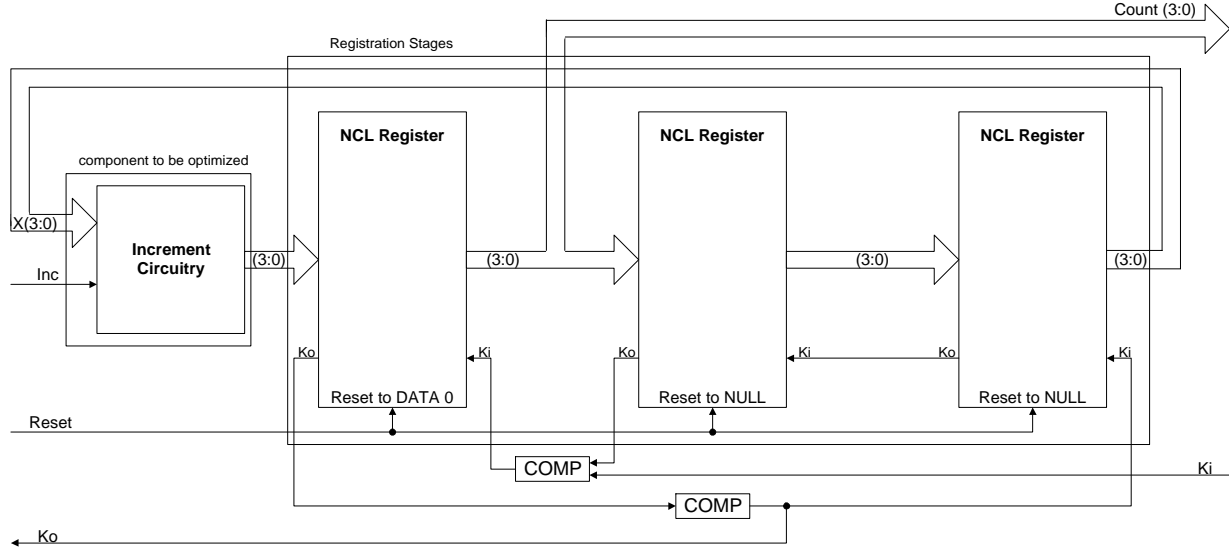


Figure 4. NCL up-counter with three-register feedback.

To design the increment circuitry using quad-rail logic requires a dual-rail *Inc* input and two quad-rail inputs, X_i and X_o , and two quad-rail outputs, S_i and S_o . Figure 5 shows the Karnaugh maps for the increment circuitry, along with the optimal coverings. Note that not all of the coverings that eliminate the dual-rail input, *Inc*, are shown, so as not to clutter the drawing. The minimal SOP equations are derived directly from the K-map coverings as follows:

$$\begin{aligned}
 S_0^0 &= \text{Inc}^0 X_0^0 + \text{Inc}^1 X_0^3 = \text{Inc}^0 X_0^0 + \text{Inc}^1 X_0^3 + \text{Inc}^0 \text{Inc}^1 + X^0 X^3 \rightarrow \text{TH24comp} \\
 S_0^1 &= \text{Inc}^0 X_0^1 + \text{Inc}^1 X_0^0 = \text{Inc}^0 X_0^1 + \text{Inc}^1 X_0^0 + \text{Inc}^0 \text{Inc}^1 + X^0 X^1 \rightarrow \text{TH24comp} \\
 S_0^2 &= \text{Inc}^0 X_0^2 + \text{Inc}^1 X_0^1 = \text{Inc}^0 X_0^2 + \text{Inc}^1 X_0^1 + \text{Inc}^0 \text{Inc}^1 + X^1 X^2 \rightarrow \text{TH24comp} \\
 S_0^3 &= \text{Inc}^0 X_0^3 + \text{Inc}^1 X_0^2 = \text{Inc}^0 X_0^3 + \text{Inc}^1 X_0^2 + \text{Inc}^0 \text{Inc}^1 + X^2 X^3 \rightarrow \text{TH24comp} \\
 S_1^0 &= \text{Inc}^0 X_1^0 + X_0^0 X_1^0 + X_0^1 X_1^0 + X_0^2 X_1^0 + \text{Inc}^1 X_0^3 X_1^3 = X_1^0 \bullet (\text{Inc}^0 + X_0^0 + X_0^1 + X_0^2) + X_1^3 \bullet (\text{Inc}^1 X_0^3) \\
 S_1^1 &= \text{Inc}^0 X_1^1 + X_0^0 X_1^1 + X_0^1 X_1^1 + X_0^2 X_1^1 + \text{Inc}^1 X_0^3 X_1^0 = X_1^1 \bullet (\text{Inc}^0 + X_0^0 + X_0^1 + X_0^2) + X_1^0 \bullet (\text{Inc}^1 X_0^3) \\
 S_1^2 &= \text{Inc}^0 X_1^2 + X_0^0 X_1^2 + X_0^1 X_1^2 + X_0^2 X_1^2 + \text{Inc}^1 X_0^3 X_1^1 = X_1^2 \bullet (\text{Inc}^0 + X_0^0 + X_0^1 + X_0^2) + X_1^1 \bullet (\text{Inc}^1 X_0^3) \\
 S_1^3 &= \text{Inc}^0 X_1^3 + X_0^0 X_1^3 + X_0^1 X_1^3 + X_0^2 X_1^3 + \text{Inc}^1 X_0^3 X_1^2 = X_1^3 \bullet (\text{Inc}^0 + X_0^0 + X_0^1 + X_0^2) + X_1^2 \bullet (\text{Inc}^1 X_0^3)
 \end{aligned}$$

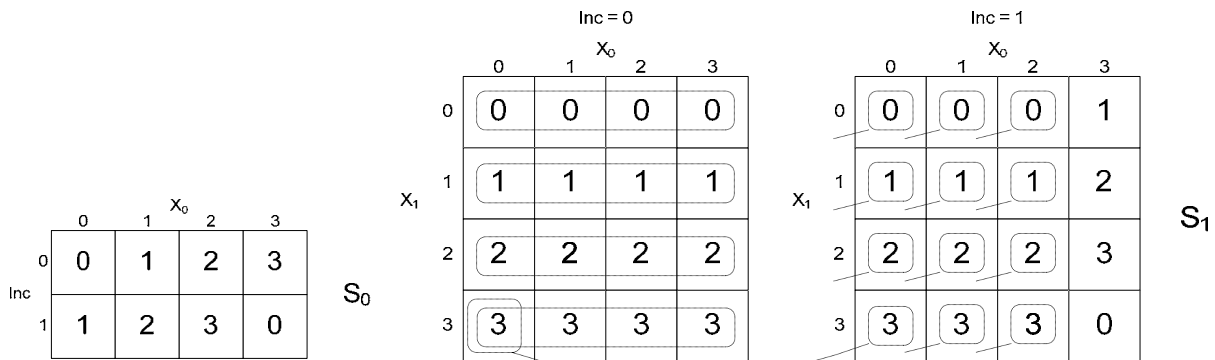


Figure 5. K-maps for quad-rail increment circuitry.

S_0 is input-complete with respect to *Inc* and X_o , since they appear in all S_0 product terms; and S_i is input-complete with respect to X_i , since it appears in all S_i product terms. Therefore, the circuit is input-complete with respect to all inputs. Furthermore, this circuit, is inherently input-complete since it is impossible to determine the

value of S without knowing the value of both X and Inc . The equations for S_0 each directly map to a TH24comp gate, after adding two *don't care* terms. The equations for all S_i rails contain two like terms, $(Inc^0 + X_0^0 + X_0^1 + X_0^2)$ and $(Inc^1 X_0^3)$, such that these can each be implemented using a single gate, and reused for all S_i rails, as shown in the resulting optimized circuit in Figure 6. Note that these two terms are mutually exclusive, such that their product can be added as a *don't care* term, along with the product of two X_i rails, as shown below, such that the equations for S_i each map to a TH24comp gate.

$$S_1^0 = X_1^0 \bullet (Inc^0 + X_0^0 + X_0^1 + X_0^2) + X_1^3 \bullet (Inc^1 X_0^3) + (Inc^0 + X_0^0 + X_0^1 + X_0^2) \bullet (Inc^1 X_0^3) + X_1^0 X_1^3$$

$$S_1^1 = X_1^1 \bullet (Inc^0 + X_0^0 + X_0^1 + X_0^2) + X_1^0 \bullet (Inc^1 X_0^3) + (Inc^0 + X_0^0 + X_0^1 + X_0^2) \bullet (Inc^1 X_0^3) + X_1^0 X_1^1$$

$$S_1^2 = X_1^2 \bullet (Inc^0 + X_0^0 + X_0^1 + X_0^2) + X_1^1 \bullet (Inc^1 X_0^3) + (Inc^0 + X_0^0 + X_0^1 + X_0^2) \bullet (Inc^1 X_0^3) + X_1^1 X_1^2$$

$$S_1^3 = X_1^3 \bullet (Inc^0 + X_0^0 + X_0^1 + X_0^2) + X_1^2 \bullet (Inc^1 X_0^3) + (Inc^0 + X_0^0 + X_0^1 + X_0^2) \bullet (Inc^1 X_0^3) + X_1^2 X_1^3$$

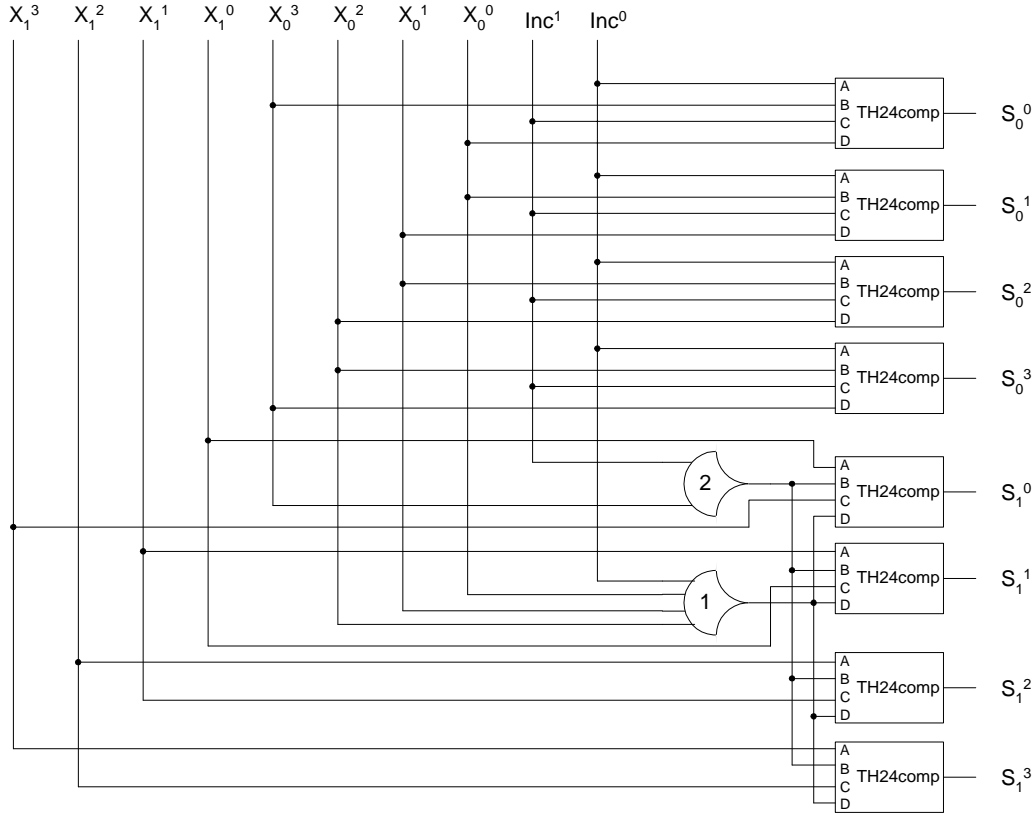


Figure 6. Optimized quad-rail increment circuitry.