PGA33X6 Quick Reference

Introduction

The PGA33X6 allows us to store one of certain Boolean functions on four Boolean parameters $f_3 : B^4 \to B$ where one of the parameters is tied to an embedded SR flip-flop, as well as two other functions f_1, f_2 with further limitations. Note that any f_3 dependent on only 3 parameters can be represented in the PGA33X6.

To begin programming the PGA33X6 to represent a function, we need to first express the output of the function as a sum of products of the parameters, where each term of the product is an input parameter or its complement. Each of the terms of the sum are a product, and we call them **minterms.** We call such an expression the **canonical normal form (CNF).**

How many minterms do we need?

The CNF of a function with n parameters has 2^n minterms. Nevertheless, we can use **Karnaugh maps** to minimize the number of minterms, and we find that 4 minterms and hence 4 AND gates represent any f such that $f(i_0, i_1, i_2) = o$. We actually need 8 AND gates to represent any f such that $f(i_0, i_1, i_2, i_3) = o$, hence we can only represent certain functions with 4 parameters.

Say we have defined f_3 and possibly f_1. If there are n unique minterms that together make up the definition of f_3 and f_1, the PGA33X6 allows us to define an f_2 as a sum on up to $6 - n$ distinct minterms, as well as any of the n minterms used for f_3 and f_1.

Other limitations of the PGA33X6

The PGA33X6 makes the following convention: an AND gate with no input outputs 0, and an OR gate with no input outputs 0. Hence to represent the function $f\left(i_0, i_1, i_2\right) = 0$ for any i_0, i_1, i_2 is easy: connect no input to the OR gates. However, to represent $f\left(i_0, i_1, i_2\right) = 1$, we need at least 2 minterms—$i + i'$ for any input i.

Notation

We extend the Boolean algebra $B = (\{0, 1\}, +, \cdot)$ to $B_\varnothing = (\{0, \varnothing, 1\}, +, \cdot)$ so that

$$\begin{array}{|c|ccc|}\hline a+b & 0 & \varnothing & 1 \\ \hline 0 & 0 & 0 & 1 \\ \varnothing & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ \hline \end{array} \quad \begin{array}{|c|ccc|}\hline ab & 0 & \varnothing & 1 \\ \hline 0 & 0 & 0 & 0 \\ \varnothing & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ \hline \end{array} \quad \begin{array}{|c|c|}\hline a & a' \\ \hline 0 & 1 \\ \varnothing & \varnothing \\ 1 & 0 \\ \hline \end{array} \; . \tag{1}$$

In addition, we write $\{k\}_{k=a}^{b} = \{k \in \mathbb{Z} : a \leq k \leq n\}$.

Definitions

Inputs. From the top pin to the bottom, we denote the state of the inputs at step t with $i_{(0,t)}, i_{(1,t)}, i_{(2,t)}$. Then, denote the state `data` of the SR flip-flop at step t with $i_{(3,t)}$. **Input values.** For all $t \in \{t\}_{t=0}^{\infty}$ and $k \in \{k\}_{k=0}^{3}$,

$$i_{(k,t)} \in \{0, 1\}. \tag{2}$$

Outputs. From the top pin to the bottom, we denote the state of the outputs at step t with $o_{(0,t)}, o_{(1,t)}, o_{(2,t)}$, and the input `s` to the SR flip-flop at step t with $o_{(3,t)}$. Note that $o_{(1,t)}$ is also the input `r` to the SR flip-flop, and that the outputs of the OR gates are, from top to bottom, $o_{(3,t)}, o_{(1,t)}, o_{(2,t)}$. **Output values.** For all $t \in \{t\}_{t=0}^{\infty}$ and $k \in \{k\}_{k=0}^{3}$,

$$o_{(k,t)} \in \{0, 1\}. \tag{3}$$

AND Inputs. Let us denote using $g_{(k,n)}$ whether it is $i_{(k,t)}$ or $i'_{(k,t)}$ or neither of the two that is connected to the nth AND gate. For all $k \in \{k\}_{k=0}^{3}$ and $n \in \{n\}_{n=0}^{5}$,

$$g_{(k,n)} \in \{0, \varnothing, 1\}. \tag{4}$$

OR Inputs. Let us denote using $h_{(k,n)}$ whether the output of the nth AND gate is connected to the kth OR gate. For all $k \in \{k\}_{k=1}^{3}$ and $n \in \{n\}_{n=0}^{5}$,

$$h_{(n,k)} \in \{0, 1\}. \tag{5}$$

SR flip-flop selector. Finally, there is a programmable bit b that determines if $o_{(0,t)}$ takes its values from $i_{(3,t)}$ or from $o_{(3,t)}$.

$$b \in \{0, 1\}. \tag{6}$$

Operation

Say we have programmed/set all $g_{(k,n)}$ and $h_{(n,k)}$ and b. In addition, it is now step t and we know the input values $i_{(k,t)}$. We want to know $o_{(k,t)}$ and $i_{(3,t+1)}$.

Minterms. Then for all $t \in \{t\}_{t=0}^{\infty}$ and $n \in \{n\}_{n=0}^{5}$, the output of the nth AND gate at step t is the modified minterm

$$w_{t,n} = \left(\sum_{k=0}^{3} g_{(k,n)} + g'_{(k,n)} \right) \left(\prod_{k=0}^{3} i_{(k,t)} g_{(k,n)} + i'_{(k,t)} g'_{(k,n)} \right). \tag{7}$$

Outputs. Still at step t, for all $k \in \{k\}_{k=1}^{3}$, the output of kth OR gate is

$$o_{(k,t)} = \sum_{n=0}^{5} w_{(t,n)} h_{(n,1)}. \tag{8}$$

In addition,

$$o_{(0,t)} = b' o_{(3,t)} + b i_{(k,t)}. \tag{9}$$

SR Flip-flop. Finally, the output of the SR flip-flop for the next step $t+1$ is set depending on $\mathtt{s} = o_{(3,t)}, \mathtt{r} = o_{(1,t)}, \mathtt{data} = i_{(3,t)}$.

$$i_{(3,t+1)} = o_{(3,t)} + i_{(3,t)} o'_{(1,t)}. \tag{10}$$