Reduction of a polynomial F *w.r.t.* another polynomial G is defined as follows:

$$F - \frac{lt(F)}{lt(G)} \cdot G = F + \frac{lt(F)}{lt(G)} \cdot G \tag{1}$$

where $lt(F)$ and $lt(G)$ denote the leading terms of polynomials, F and G, respectively. Note that - can be replaced with + as we are performing modulo 2 sum.

Of course, this equation holds only if $lt(G)$ divides $lt(F)$. As we are working on polynomials *modulo* 2, the coefficients in the polynomials are either 1 or 0. Therefore, the above expression can be written as,

$$F - \frac{lm(F)}{lm(G)} \cdot G = F + \frac{lm(F)}{lm(G)} \cdot G \tag{2}$$

where $lm(F)$ and $lm(G)$ denote the leading monomials of polynomials, F and G, respectively.

Consider, as an example, the polynomials F and G are,

$$F = f \cdot d + f + c \tag{3}$$

$$G = f + b + a \tag{4}$$

with monomial ordering $f > d > c > b > a$

We want to reduce F *w.r.t.* G. Redcution using equation 2 will require two steps, one for the term $f \cdot d$ and other for the term f in F. This reduction can be completed in one step if we know all the terms in F that have f in them. Note that the leading monomial of G will always be a single variable, as G models a gate. Now consider the ZBDDs of F and G in Fig. 1 and 2 respectively. The ZBDDs represent the polynomials as a set of monomials ($\{f \cdot d, f, c\}$ for F and $\{f, b, a$ for $G\}$) appearing in them. The CUDD manager creates the ZBDDs with the defined monomial order, and therefore, the topmost node in both diagrams is f. Checking if $lm(G)$ divides $lm(F)$ becomes trivial as we just need to compare the indices of top-most nodes of F and G, which in this case are equal.

We want to perform reduction of F *w.r.t.* G in one step. If we check the THEN branch of node f in F, we will find that it represents the polynomial, $d + 1$. Therefore, the THEN branch of the top-most node of F gives us all the terms that appear with f. So the reduction can be performed by multiplying $d + 1$ with G and adding this product to F *modulo* 2,

$$\begin{aligned}&(f \cdot d + f + c) + (d + 1) \cdot (f + b + a)\\ &= 2 \cdot (f \cdot d + f) + c + (d + 1) \cdot (b + a)\\ &= c + (d + 1) \cdot (b + a)\end{aligned}$$

Consider the follwing terminologies,

$$\begin{aligned}&head(F) = \text{THEN branch of top-most node of } F = d + 1\\ &tail(F) = \text{ELSE branch of top-most node of } F = c\\ &tail(G) = \text{ELSE branch of top-most node of } G = b + a\end{aligned}$$

The last step of reduction process can be written as,

$$= c + (d+1)\cdot(b+a)$$
$$tail(F) + head(f)\cdot tail(G)$$

The data structure for a ZBDD node has two pointers for the THEN child and ELSE child, respectively. Therefore, $head(F)$, $tail(F)$, and $tail(G)$ can be acquired by just accessing the respective pointers. So the reduction process effectively involves two operations, a modulo 2 sum (SUM) and a product (PROD). The CUDD package provides a function for computing PROD. The SUM operation of two polynomials, F and G, can be performed as follows,

$$SUM(F,G) = (F \cup G) - (F \cap G)$$

where $\cup$, $\cap$, and $-$ represents set union, set intersection, and set difference respectively. The functions for performing these three operations are present in the CUDD package.