

【2】 =====

図 8.2(b) において, ゲート 6~9 から成るマルチプレクサの制御信号を $cntrl$ とすると, OR ゲート 15 の出力が AND ゲート 8 の出力の値を決定し, 外部出力 c_{i+2} の値を決定するためには, $cntrl = 0$ でなければならないが, この $cntrl$ は次式で表される.

$$cntrl = (x_i \oplus y_i) \cdot (x_{i+1} \oplus y_{i+1}) = (x_i + y_i) \cdot (\bar{x}_i + \bar{y}_i) \cdot (x_{i+1} + y_{i+1}) \cdot (\bar{x}_{i+1} + \bar{y}_{i+1})$$

今, $x_i \oplus y_i = 0$ であるとする, AND ゲート 4 の出力は入力 c_i の値に関わらず 0 であるから, 入力 c_i の値が AND ゲート 4 の出力の値を決定しない.

そこで, $x_i \oplus y_i = 1$ であるとする, AND ゲート 4 の出力は c_i の値で決定される. この値が OR ゲート 5 の出力 c_{i+1} の値を決定するためには, $x_i \cdot y_i = 0$ でなければならないが, $x_i \oplus y_i = 1$ であるから, これは満たされている. 従って, 入力 c_i の値で OR ゲート 5 の出力 c_{i+1} の値が決定される可能性がある.

しかし, c_{i+1} の値が AND ゲート 14 の出力 z の値を決定するには, $x_{i+1} \oplus y_{i+1} = 1$ でなければならないが, $x_i \oplus y_i = 1$ かつ $x_{i+1} \oplus y_{i+1} = 1$ であれば $cntrl = 1$ となってしまうため, AND ゲート 8 の出力は 0 になり, z の値が AND ゲート 8 の出力値を決定することはない.

従って, c_i から, AND ゲート 4, OR ゲート 5, AND ゲート 14, OR ゲート 15 を通り, マルチプレクサの AND ゲート 8 の出力に至るパスは, 偽パスであることが分かる.

このパスが偽パスであることは, パス上の各論理ゲートにおいて, パスの信号が論理ゲートの出力を決定する条件を全て満たす入力値の組があるか否かを調べることによって分かる. 各ゲートにおいて, パス上の入力が出力を決定する条件は, 下記のように書けるから, c_i の値が AND ゲート 8 の出力を決定するためには, これら全てが成立していなければならない.

AND ゲート 4 において, $x_i \oplus y_i = (x_i + y_i) \cdot (\bar{x}_i + \bar{y}_i) = 1$

OR ゲート 5 において, $x_i \cdot y_i = 0$, すなわち $\bar{x}_i + \bar{y}_i = 1$

AND ゲート 14 において, $x_{i+1} \oplus y_{i+1} = (x_{i+1} + y_{i+1}) \cdot (\bar{x}_{i+1} + \bar{y}_{i+1}) = 1$

OR ゲート 15 において, $x_{i+1} \cdot y_{i+1} = 0$, すなわち $\bar{x}_{i+1} + \bar{y}_{i+1} = 1$

AND ゲート 8 において,

$$cntrl = (x_i \oplus y_i) \cdot (x_{i+1} \oplus y_{i+1}) = (x_i + y_i) \cdot (\bar{x}_i + \bar{y}_i) \cdot (x_{i+1} + y_{i+1}) \cdot (\bar{x}_{i+1} + \bar{y}_{i+1}) = 0 ,$$

すなわち,

$$\begin{aligned} \overline{cntrl} &= \overline{x_i \oplus y_i} + \overline{x_{i+1} \oplus y_{i+1}} = \overline{(x_i + y_i) \cdot (\bar{x}_i + \bar{y}_i)} + \overline{(x_{i+1} + y_{i+1}) \cdot (\bar{x}_{i+1} + \bar{y}_{i+1})} \\ &= x_i \cdot y_i + \bar{x}_i \cdot \bar{y}_i + x_{i+1} \cdot y_{i+1} + \bar{x}_{i+1} \cdot \bar{y}_{i+1} = 1 \end{aligned}$$

これら全ての式が同時に成り立つような入力値の組があるか否かを調べることは、以下の論理式が1になるような入力値の組合せがあるか否かを調べることと同値である。

$$(x_i \oplus y_i) \cdot (\bar{x}_i + \bar{y}_i) \cdot (x_{i+1} \oplus y_{i+1}) \cdot (\bar{x}_{i+1} + \bar{y}_{i+1}) \cdot (\overline{x_i \oplus y_i + x_{i+1} \oplus y_{i+1}}) = 1$$

このように、偽パスか否かは、和積形で書かれた(排他的論理和を和積形で書き換えることは容易である)論理式が1となるような入力の値の組合せがあるか否かを判定すれば分かるが、それを効率的に行うには工夫が必要となる。

なお、図 8.2(b) の回路のゲート1から5およびゲート11から15は、共に全加算器であるから、この回路は2ビットの桁上げ伝搬加算器を含んでいることが分かるであろう。桁上げ伝搬加算器以外の回路(論理ゲート6~10)は、偽パスで計算する桁上げ(キャリー) c_{i+2} を少ない段数の論理ゲートで計算するための回路になっており、これらは6.2節で触れた桁上げ先見回路(Carry Look-ahead Circuit)である。このような回路が付加されると、元の桁上げ c_{i+2} を計算する回路は冗長(redundant)となる。