

7セグメントLEDを用いた2進-10進デコーダについて

表 1: 0/1 表現による真理値表 (テキスト記載の表の OFF/ON を 0/1 に変換)

スイッチ入力				7セグメントLED							
I3	I2	I1	I0	a	b	c	d	e	f	g	dec.
1	1	1	1	0	0	0	0	0	0	1	1
1	1	1	0	1	0	0	1	1	1	1	1
1	1	0	1	0	0	1	0	0	1	0	1
1	1	0	0	0	0	0	0	1	1	0	1
1	0	1	1	1	0	0	1	1	0	0	1
1	0	1	0	0	1	0	0	1	0	0	1
1	0	0	1	0	1	0	0	0	0	0	1
1	0	0	0	0	0	0	1	1	1	1	1
0	1	1	1	0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0	1	0	0	1

論理式の求め方

表 1 の a ~ g について、結果が “1” になる I0 ~ I3 の組合せを OR で並べる。

(例 1) ”a” の場合

注) I0 ~ I3 の並びが表と逆順になっていることに注意。

- (I0, I1, I2, I3) が (0, 1, 1, 1)、(1, 1, 0, 1) の 2 つの組合せを or でつなげる。

$$a = \overline{I0} \cdot I1 \cdot I2 \cdot I3 + I0 \cdot I1 \cdot \overline{I2} \cdot I3 \quad (\text{論理式})$$

$$a <= (\text{not } I0 \text{ and } I1 \text{ and } I2 \text{ and } I3) \text{ or } (I0 \text{ and } I1 \text{ and not } I2 \text{ and } I3) \quad (\text{VHDL 表記})$$

(例 2) ”g” の場合

- (I0, I1, I2, I3) が (1, 1, 1, 1)、(0, 1, 1, 1)、(0, 0, 0, 1) の 3 つの組合せを or でつなげる。

$$a = I0 \cdot I1 \cdot I2 \cdot I3 + \overline{I0} \cdot I1 \cdot I2 \cdot I3 + \overline{I0} \cdot \overline{I1} \cdot \overline{I2} \cdot I3$$

$$a <= (I0 \text{ and } I1 \text{ and } I2 \text{ and } I3) \text{ or } (\text{not } I0 \text{ and } I1 \text{ and } I2 \text{ and } I3) \text{ or } (\text{not } I0 \text{ and not } I1 \text{ and not } I2 \text{ and } I3)$$

サンプル3 : 7セグメントLEDを用いた2進-10進デコーダ

以下のサンプルプログラムでは b, d, f が未完成である。

```
--  
-- binary to digit decoder using 7-segment  
--  
library IEEE;  
use IEEE.std_logic_1164.all;  
  
entity SEGMENT7 is  
    port ( I0, I1, I2, I3 : in std_logic;  
          a, b, c, d, e, f, g : out std_logic);  
end entity;  
  
architecture DATAFLOW of SEGMENT7 is  
begin  
  
    a <= (not I0 and I1 and I2 and I3) or (I0 and I1 and not I2 and I3);  
  
    b <=  
  
    c <= (I0 and not I1 and I2 and I3);  
  
    d <=  
  
    e <= (not I0 and I1 and I2 and I3) or (not I0 and not I1 and I2 and I3) or  
        (I0 and I1 and not I2 and I3) or (not I0 and I1 and not I2 and I3) or  
        (not I0 and not I1 and not I2 and I3) or (not I0 and I1 and I2 and not I3);  
  
    f <=  
  
    g <= (I0 and I1 and I2 and I3) or (not I0 and I1 and I2 and I3) or  
        (not I0 and not I1 and not I2 and I3);  
  
end DATAFLOW;
```