

教科書にも出てくる オーソドックスな素材で初体験! ハードウェア記述でつくる 汎用ロジックIC回路②

～グレイ・コード生成回路, タイマ回路ほか～

中 幸政 Yukimasa Naka

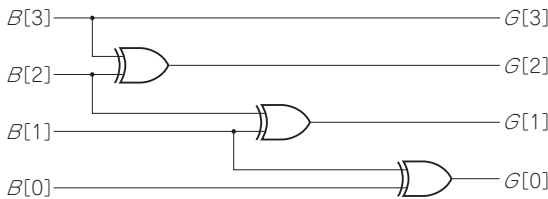


図1 バイナリ・コード(B)をグレイ・コード(G)に変換する回路

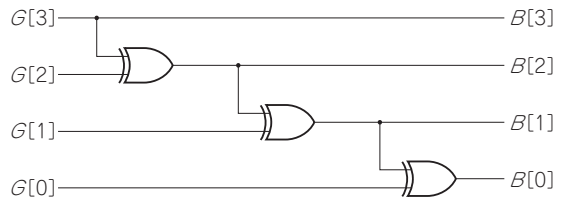


図2 グレイ・コード(G)をバイナリ・コード(B)に変換する回路

本コーナではFPGAを使って回路を作るためのノウハウやテクニックを紹介します。今回は、スイッチやセンサの信号を入力する時に使うオーソドックスな回路をVHDLで記述していきます。〈編集部〉

[回路1] グレイ・コード変換回路 相当する汎用ロジックIC: 7486

図1はバイナリ・コードをグレイ・コードに変換する回路です。図2はグレイ・コードをバイナリ・コードに変換する回路です。それぞれの回路を論理合成できるコードで記述してみましょう。このとき、入力のビット数は可変できるようにして、procedure文を使って記述することにします。

● グレイ・コードは何のためにある?

表1に4ビットのグレイ・コードとバイナリ・コードの対応表を示します。グレイ・コードは回転を検出するロータリ・エンコーダの出力信号としてよく使われています。

グレイ・コードでは、ある符号(例えば"0011")の隣り合う符号("0001"と"0010")が元の符号から、ある1ビットだけを変えた符号になっています。そのため値が変化している途中で、元の値と大きく異なる値を読み出してしまうことがなく、バイナリ・コードに比べて安全という特徴があります。

表1 16進数, バイナリ・コード, グレイ・コードの対応表

16進数	バイナリ・コード	グレイ・コード
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
A	1010	1111
B	1011	1110
C	1100	1010
D	1101	1011
E	1110	1001
F	1111	1000

● グレイ・コード変換回路の記述

リスト1が記述例です。サブプログラムはarchitecture宣言の中に記述することもできますが、packageの中に記述するのが一般的です(コラム参照)。

procedure文がfunction文と異なるのは、戻り値がないことです。その代わりに出力パラメータがあるため、入出力パラメータの宣言はport文の文法と同じようになります(リスト1の6, 7, 9, 10行目)。

サブプログラムを記述する時は、いくつかの工夫が必要です。たとえば、17行目では、入力にわざわざ別名を付けて、入力と出力の範囲を揃えています。入力が4 downto 1で、出力が3 downto 0のとき