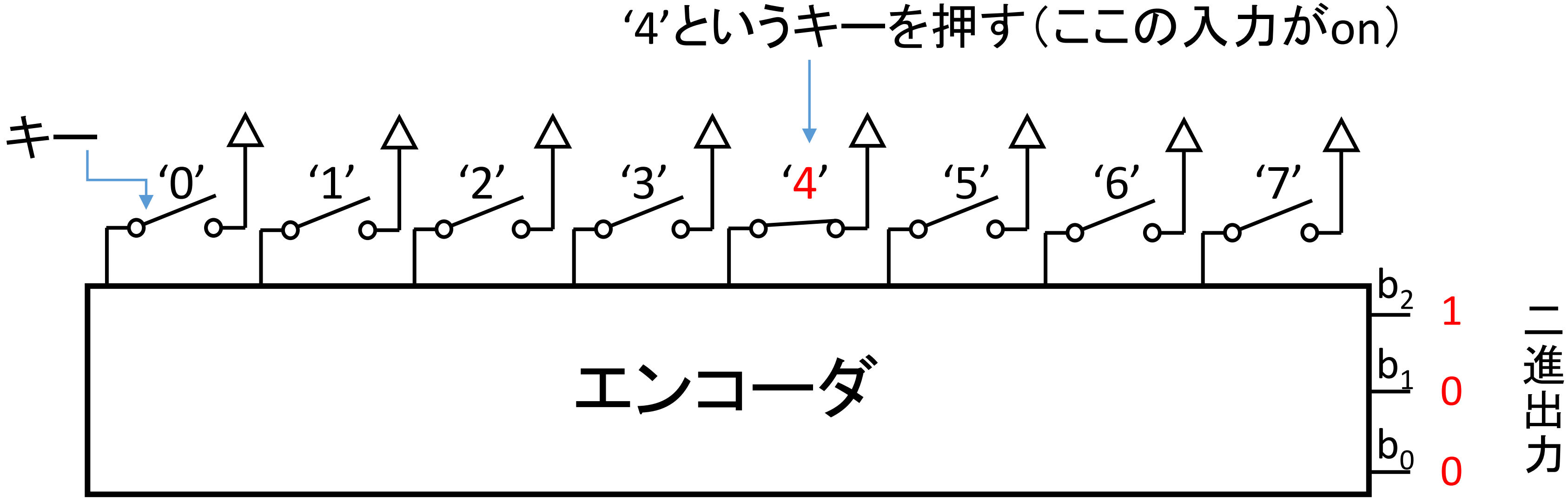
エンコーダ・デコーダ

エンコーダ

- ‘0’～’7’の数字入力のキーボードを考える
- これらは三桁の二進数で表現できる。ここではアスキーコードを無視
- たとえば’4’のキーを押した場合、100が出力されるような回路が必要
- このような、2^n個の入力に対し、nビットの符号を発生させる回路をエンコーダ（符号化回路）と呼ぶ



エンコーダ

真理値表（必要な部分のみ）

入力								出力		
‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	b_2	b_1	b_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

該当キーを押す:1　押さない:0

エンコーダ

真理値表（必要な部分のみ）

入力								出力		
'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	b_2	b_1	b_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

論理式 $b_0 = '1' + '3' + '5' + '7'$

エンコーダ

真理値表（必要な部分のみ）

入力								出力		
‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	b_2	b_1	b_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

論理式 $b_1 =' 2' + '3' + ' 6' + ' 7'$

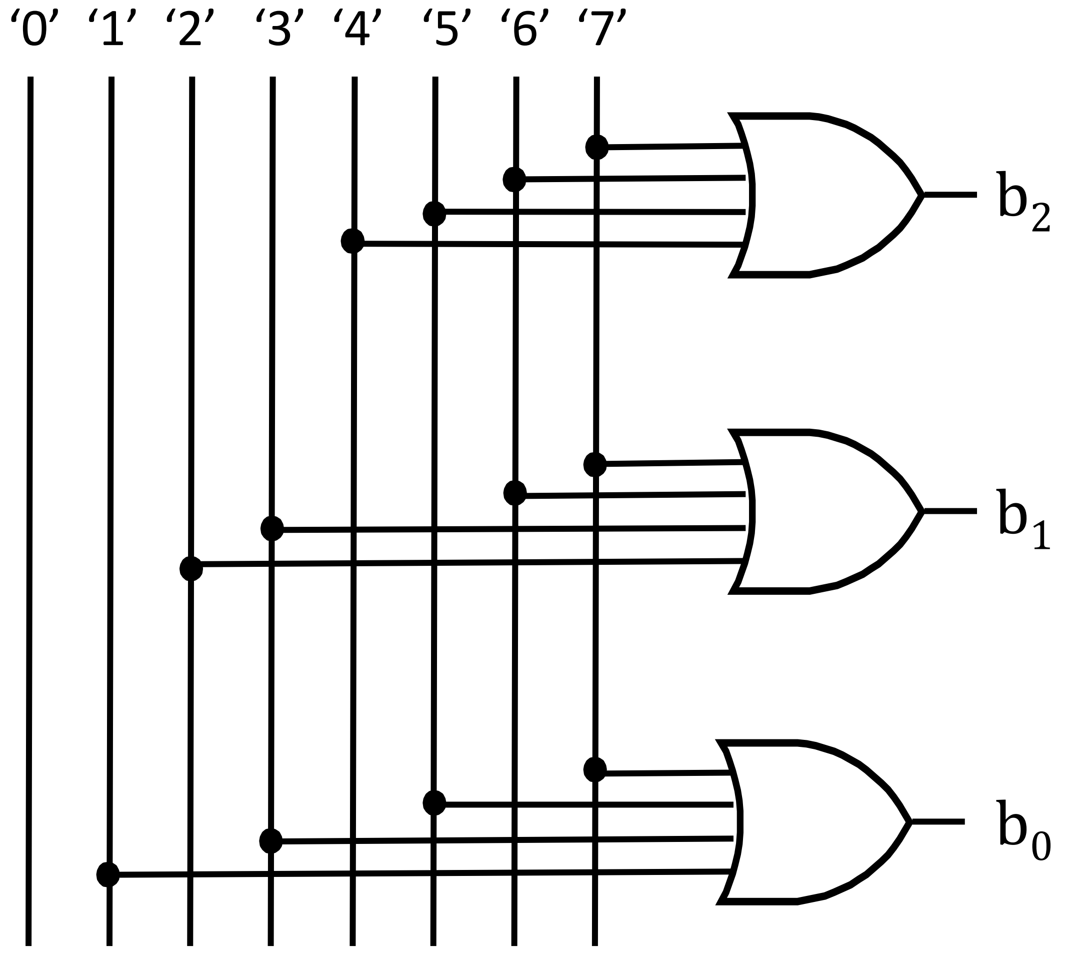
エンコーダ

真理値表（必要な部分のみ）

入力								出力		
‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	b_2	b_1	b_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

論理式　$b_2 = '4' + '5' + '6' + '7'$

エンコーダ



注意

- 上記の論理式はまず、直感的には理解できる。つまり、’1’ or ‘3’ or ‘5’ or ‘7’を押したらb_0は1

  → $b_0 = {'1'} + {'3'} + {'5'} + {'7'}$

- しかし、真理値表から加法標準形で論理式を求めると、

$$b_0 = \overline{'0'}\,{'1'}\,\overline{'2'}\cdots\overline{'7'} + \overline{'0'}\;\overline{'1'}\;\overline{'2'}\,{'3'}\cdots\overline{'7'} + \cdots$$

  となる

- つまり、論理式を簡単化する必要がある
- この簡単化については後述

Question

- ４入力ORゲートを2入力ORゲートで実現しなさい

エンコーダの論理式は真理値表から導出できる

- 3入力･2出力のケースを考える

A	B	C	X	Y
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
1	0	0	1	1
0	1	1	*	*
1	0	1	*	*
1	1	0	*	*
1	1	1	*	*

エンコーダの真理値はここまで

*は1,0のどちらでもOKだが、ここでは1と見なす。

エンコーダの論理式は真理値表から導出できる

Xについて、加法標準形で論理式をもとめ、変形していくと以下が得られる。Yについても同様に求められる。

$$
\begin{aligned}
X &= \overline{A}\cdot B\cdot\overline{C} + A\cdot\overline{B}\cdot\overline{C} + \left(\overline{A}\cdot B\cdot C + A\cdot\overline{B}\cdot C + A\cdot B\cdot\overline{C} + A\cdot B\cdot C\right) \\
&= \overline{A}\cdot B\cdot\left(\overline{C}+C\right) + A\cdot\overline{B}\cdot(\overline{C}+C) + A\cdot B\cdot\left(\overline{C}+C\right) && \leftarrow \text{分配則} \\
&= \overline{A}\cdot B + A\cdot\overline{B} + A\cdot B && \leftarrow \text{相補則} \\
&= \overline{A}\cdot B + A\cdot\overline{B} + A\cdot B + A\cdot B && \leftarrow \text{べき等則} \\
&= \left(\overline{A}+A\right)\cdot B + A\cdot\left(\overline{B}+B\right) && \leftarrow \text{分配則} \\
&= B + A = A + B && \leftarrow \text{相補則}
\end{aligned}
$$

カルノー図

- 論理式の変形の代わりに、カルノー図と呼ばれる方法で論理式を簡単化することもできる
- まず、真理値表から以下のカルノー図(X)を得る

		C	
AB		0	1
	00	0	0
	01	1	*
	11	*	*
	10	1	*

グレイ符号

カルノー図

- 1か*のマス目を、2^n個(n=0から)ずつ、できるだけ大きく(つまりnを大きく)囲む(グループ化する)
- べき等則A+A=Aを利用すると、(論理式が等価という意味では)下記のように範囲を重ねてもよい

		C	
		0	1
AB	00	0	0
	01	1	*
	11	*	*
	10	1	*

カルノー図

- 上の楕円（Xの値がCとAの値に影響を受けないため）: B
- 下の楕円（Xの値がCとBの値に影響を受けないため）: A
- ∴X = B + A (= A + B)

AB		C	
		0	1
	00	0	0
	01	1	*
	11	*	*
	10	1	*

manaba小テスト：演習10-1

- Yのカルノー図について
- 20分, 6点

演習

- 講義資料では（上の楕円は「Xの値がCとAの値に影響を受けないため」）…で論理式を導出しているが、その二つの楕円について、それぞれの論理式を立てて、それを変形して論理式X=A+Bを導きなさい。

AB	C	
	0	1
00	0	0
01	1	*
11	1	*
10	*	*

補足

AB	CD			
	00	01	11	10
00				
01				
11				
10				

論理式は？

論理式は？

- 隣接しているマス目は論理変数1つのみ異なる
- 隣接しているマス目はその値が1または*の場合、まとめること(簡単化すること)は可能

補足

AB	CD				
		00	01	11	10
	00				
	01				
	11				
	10				

論理式は？

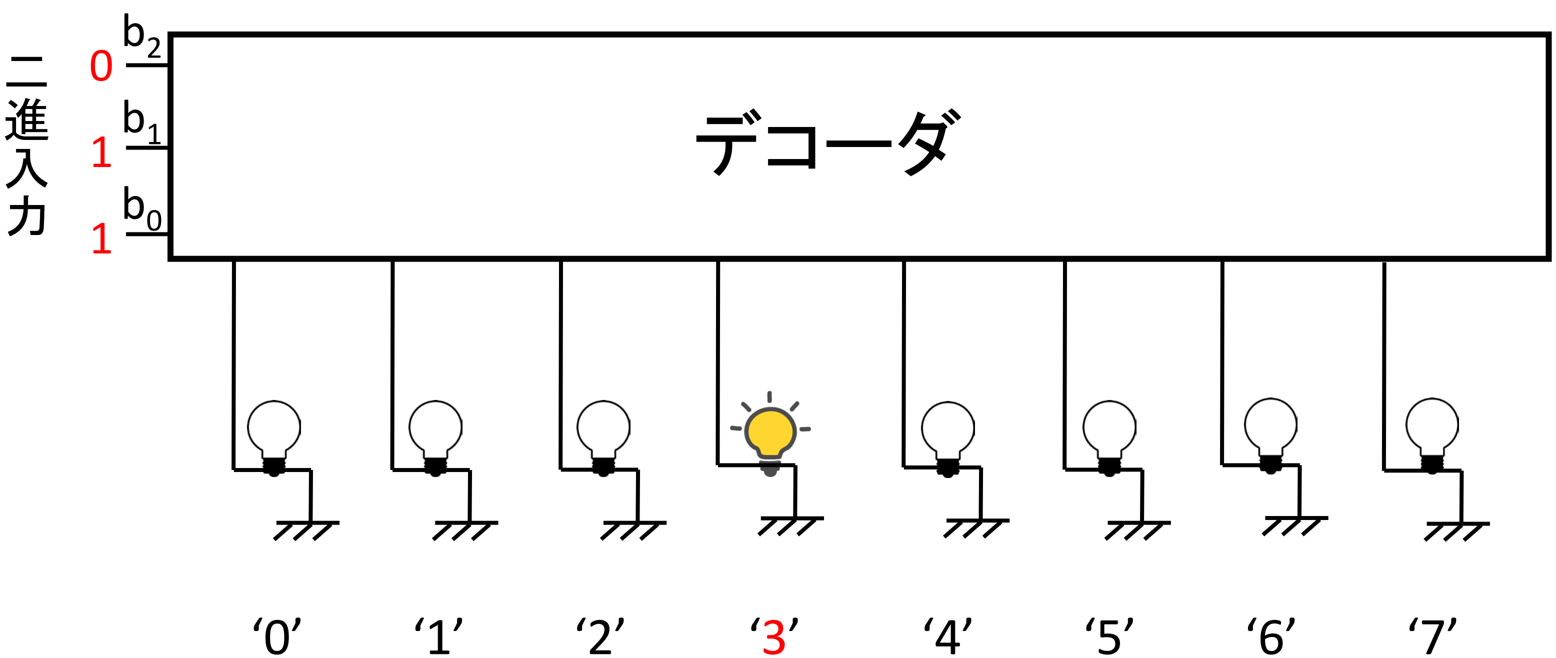
- 上記マスも論理変数1つのみ異なるので、隣接しているとみなす

manaba小テスト:演習10-2

- 論理式をカルノー図で簡単化する問題
- 15分, 6点

デコーダ

- エンコーダと逆のことを行う回路。つまり、nビットの入力（符号）から2^n個の出力（信号）生成するもの、解読回路とも呼ぶ
- 二進数で符号化されている命令やメモリアドレスの解読に使われる
- 以下、まず特定の二進数に反応するランプを点灯する回路を示す



デコーダ

真理値表

入力			出力							
b_2	b_1	b_0	‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

1 : 点灯

デコーダ

論理式

$$'0' = \overline{b_2} \cdot \overline{b_1} \cdot \overline{b_0}$$

$$'1' = \overline{b_2} \cdot \overline{b_1} \cdot b_0$$

$$'2' = \overline{b_2} \cdot b_1 \cdot \overline{b_0}$$

$$'3' = \overline{b_2} \cdot b_1 \cdot b_0$$

$$'4' = b_2 \cdot \overline{b_1} \cdot \overline{b_0}$$

$$'5' = b_2 \cdot \overline{b_1} \cdot b_0$$

$$'6' = b_2 \cdot b_1 \cdot \overline{b_0}$$

$$'7' = b_2 \cdot b_1 \cdot b_0$$

Question

- $'4' = \mathrm{b}_2 \cdot \overline{\mathrm{b}_1} \cdot \overline{\mathrm{b}_0}$を実現する回路を論理ゲートで作成しなさい。ただし、ゲートの入力数は2以上でも可とする。