

論理回路基礎 (第5回)

坂井 修一

東京大学大学院 情報理工学系研究科 電子情報学専攻
東京大学 工学部電気工学科

- スケジュール
- 組合せ回路の実例 (その2)
 - ALU
 - デコーダ、エンコーダ
 - マルチプレクサ、デマルチプレクサ
 - パリティチェッカ、コンパレータ

講義の概要と予定

1. デジタル回路入門
2. 論理演算
3. 組み合わせ回路の構成法
4. 組合せ回路の実例
5. フリップフロップ
6. 基本的な順序回路
7. 一般的な順序回路
8. 論理回路の実現
9. 記憶回路
10. デジタル回路から電子計算機へ

休講: 12月2日、 試験: 3月3日(予定)

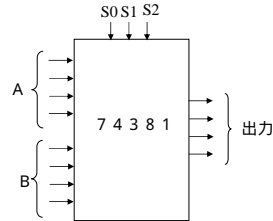
4.4 ALU

■ ALU = 算術論理ユニット(Arithmetic Logic Unit)

制御信号によって、どのような演算を行うかを決める組合せ回路

例 . 7 4 3 8 1 (という I C) by T I

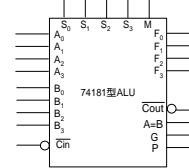
制御信号			出力
S2	S1	S0	
0	0	0	0000
0	0	1	B-A
0	1	0	A-B
0	1	1	A+B
1	0	0	A ⊕ B
1	0	1	A OR B
1	1	0	A AND B
1	1	1	1111



例 2 . 7 4 1 8 1

A+B, A-B, \bar{A} , \bar{B} , A ⊕ B, A+1, A-1, A OR B, A AND B, 1111, 0000, A, B,

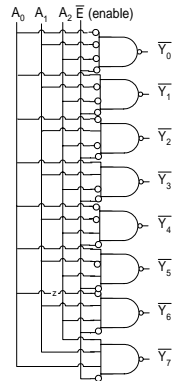
7 4 1 8 1 型 ALU



制御信号 S ₃ S ₂ S ₁ S ₀	M=1: 論理演算	M=0: 算術演算	
		Cin=0	Cin=1
0 0 0 0	$F=A$	$F=A$	$F=A \text{ PLUS } 1$
0 0 0 1	$F=A + B$	$F=A+B$	$F=(A+B) \text{ PLUS } 1$
0 0 1 0	$F=\bar{A} + B$	$F=A+\bar{B}$	$F=(A+\bar{B}) \text{ PLUS } 1$
0 0 1 1	$F=0$	$F=1111$	$F=\text{ZERO}$
0 1 0 0	$F=A \cdot B$	$F=A \text{ PLUS } A \cdot \bar{B}$	$F=A \text{ PLUS } A \cdot \bar{B} \text{ PLUS } 1$
0 1 0 1	$F=B$	$F=(A+B) \text{ PLUS } A \cdot \bar{B}$	$F=(A+B) \text{ PLUS } A \cdot \bar{B} \text{ PLUS } 1$
0 1 1 0	$F=A \oplus B$	$F=A \text{ MINUS } B \text{ MINUS } 1$	$F=A \text{ MINUS } B$
0 1 1 1	$F=A \cdot \bar{B}$	$F=A \cdot \bar{B} \text{ MINUS } 1$	$F=A \cdot \bar{B}$
1 0 0 0	$F=\bar{A} + B$	$F=A \text{ PLUS } A \cdot B$	$F=A \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 0 1	$F=A \oplus \bar{B}$	$F=A \text{ PLUS } B$	$F=A \text{ PLUS } B \text{ PLUS } 1$
1 0 1 0	$F=B$	$F=(A + \bar{B}) \text{ PLUS } AB$	$F=(A+\bar{B}) \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 1 1	$F=A \cdot B$	$F=A \cdot B \text{ MINUS } 1$	$F=A \cdot B$
1 1 0 0	$F=1$	$F=A \text{ PLUS } A$	$F=A \text{ PLUS } A \text{ PLUS } 1$
1 1 0 1	$F=A + \bar{B}$	$F=(A+B) \text{ PLUS } A$	$F=(A+B) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 0	$F=A + B$	$F=(A+B) \text{ PLUS } A$	$F=(A+B) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 1	$F=A$	$F=A \text{ MINUS } 1$	$F=A$

4.5 デコーダ (複合器、decoder)

- 2進符号 i (n -bit) / 本目の信号線をonにする、という論理回路



普通、出力はonのときに、0にする(active low)

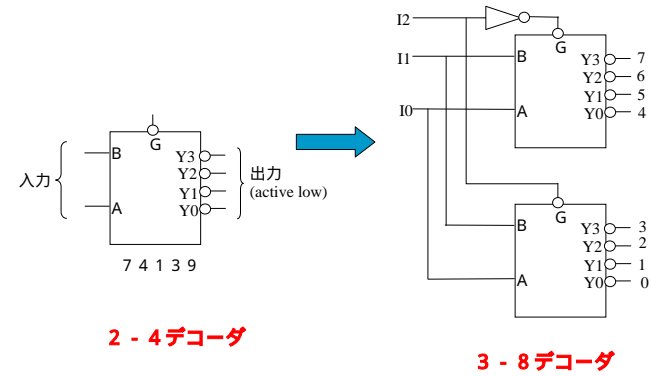
cf. active high: onのとき 1

e.g. $A_2A_1A_0=010$ のとき、 Y_2 が0になる

論理回路基礎

東大・坂井

デコーダの連結



2 - 4 デコーダ

3 - 8 デコーダ

論理回路基礎

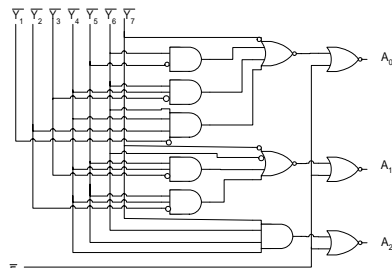
東大・坂井

4.6 エンコーダ (符号化器、encoder)

- i 本目の信号線がon 2進符号 i (n -bit)を生成する論理回路

入力								出力			
\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3	\bar{Y}_4	\bar{Y}_5	\bar{Y}_6	\bar{Y}_7	E	A_0	A_1	A_2
*	*	*	*	*	*	*	*	0	0	0	1
*	*	*	*	*	*	*	0	1	0	0	1
*	*	*	*	*	0	1	1	0	1	0	1
*	*	*	0	1	1	1	0	0	0	1	1
*	*	0	1	1	1	1	0	1	1	0	0
*	0	1	1	1	1	1	0	0	0	1	0
*	0	1	1	1	1	1	0	1	0	0	0
*	1	1	1	1	1	0	0	0	0	0	0
*	*	*	*	*	*	*	1	0	0	0	0

(a) 真理値表



(b) 回路図

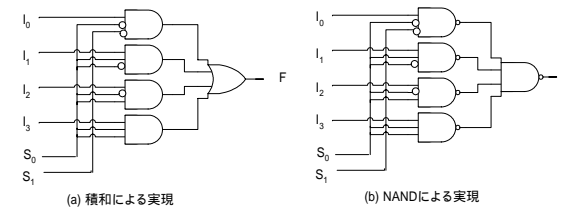
複数の入力 一つ (最小値) を出力する論理回路
優先度付きエンコーダ (Priority Encoder)

論理回路基礎

東大・坂井

4.7 マルチプレクサ (多重化装置、multiplexer)

- 複数の入力線からのデータを選択的に一本の共通線に接続する論理回路。各入力をチャンネル(channel)という



(a) 積和による実現

(b) NANDによる実現

4入力マルチプレクサ

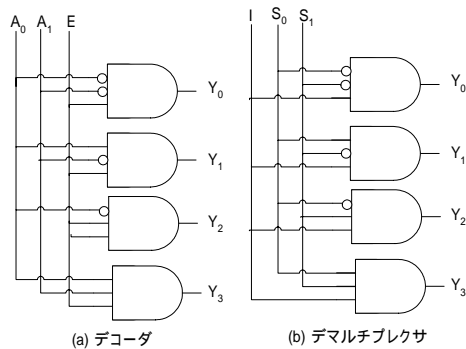
マルチプレクサをセレクトともいう

論理回路基礎

東大・坂井

4.8 デマルチプレクサ (多重分離装置、demultiplexer)

- マルチプレクサの反対で、一本の共通線からの信号を**選択的に**複数の出力線に接続する論理回路 = **イネーブル付きのデコーダと等価**

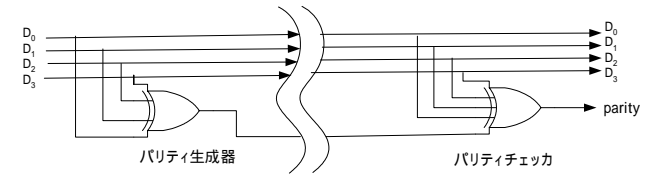


論理回路基礎

東大・坂井

4.9 パリティ生成器 (parity generator) と パリティチェッカ (parity checker)

- パリティ: 入力の中で、1の数が奇数個のときに1, 偶数個のときに0
- パリティ生成器: パリティを作る論理回路
- パリティチェッカ: パリティをチェックする論理回路



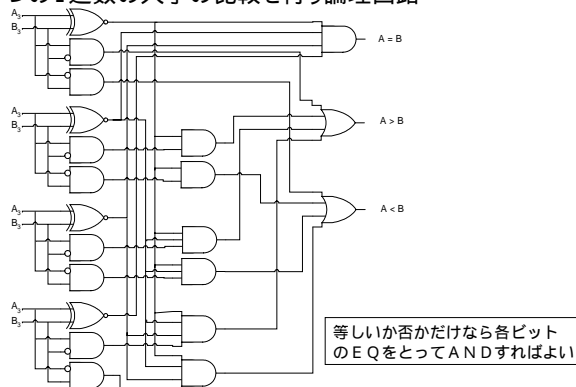
パリティ生成器は n 入力の排他的論理和になり、パリティチェッカは $n + 1$ 入力の排他的論理和になる
データ転送のエラーチェックに使われる

論理回路基礎

東大・坂井

4.10 コンパレータ (比較器、comparator)

- 2つの2進数の大小の比較を行う論理回路



等しいか否かだけなら各ビットのEQをとってANDすればよい

論理回路基礎

4ビットコンパレータ

東大・坂井

問題

1. 4ビットの2進数2つの積をとる回路を設計せよ。1ビット全加算器を基本回路として使ってよい
2. 8入力の優先度つきエンコーダ (図4.15) 2つといくつかの素子を組み合わせて、16入力の優先度つきエンコーダを作れ。
3. 減算器 (加算器とNOTの組合せ。前回の講義参照) を使って、コンパレータを作れ
 $(A, B) \rightarrow (A < B, A = B, A > B)$
4. ここで述べたパリティチェッカでは1ビットの誤りを検出することはできるが、どのビットが誤ったかを検出することができず、正しいデータに修復することができない。
1ビットの誤りを**修正する回路**を作るにはどうしたらよいか?

論理回路基礎

東大・坂井