工学部講義

論理回路基礎（1）

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　電子情報工学科／電気工学科

- はじめに
- スケジュール
- ２進数の計算

はじめに

- 本講義の目的
  - 電子計算機などに用いられるディジタル論理回路の基礎を学習する
- 時間・場所
  - 木曜日 9:00 - 10:30、駒場５９１
- ホームページ（ダウンロード可能）
  - url:　http://www.mtl.t.u-tokyo.ac.jp/~sakai/ronri/
- 教科書
  - 坂井修一『論理回路入門』（培風館）

上記教科書の通りやります。

講義の概要と予定（1／2）

１．ディジタル回路入門
ディジタルとアナログ、２進数
２．論理演算
論理演算とは、３つの基本論理演算、NORとNAND、完備性、ディジタル回路の表現、ブール代数、標準形
３．組み合わせ回路の構成法
真理値表と組合せ回路、組合せ回路の簡単化、カルノー図、クワイン・マクラスキー法
４．組合せ回路の実例
加算器、補数、減算器、ＡＬＵ、デコーダ、セレクタなど
５．フリップフロップ
ＦＦ、ＳＲラッチ、Ｄラッチ、非同期と同期、SR-FF、D-FF、マスタスレーブ形とエッジトリガ形, レジスタ、JK-FF, T-FF, FFの変換、レジスタ

講義の概要と予定（2／2）

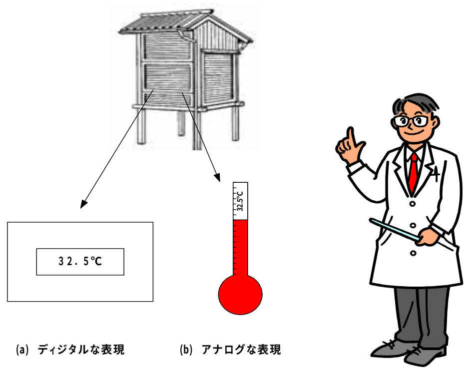
6．基本的な順序回路
順序回路とは、リプルカウンタ、並列カウンタ、カウンタ回路の構成法、シフトレジスタ
7．一般的な順序回路
同期式と非同期式の一般論、Mealy/Moore、状態割当て、状態レジスタと組合せ回路の結合、応用例
8．論理回路の実現
半導体デバイスと論理回路、PLA, FPGA、ゲートアレイとスタンダードセル、論理回路のCAD
9．記憶回路
ROM、SRAMとDRAM、メモリ設計、連想メモリ
10．ディジタル回路から電子計算機へ
電子計算機とは, 命令動作、計算機構成法

休講：　１２／２，　試験：（たぶん）３月３日
成績：　試験のみ　４年生：追試･レポート無し

１．ディジタル回路入門

- 内容
  - ディジタルとアナログ
  - 2進数
    - 数の表現
    - 数の変換
    - 2進数の演算
  - 組合せ回路と順序回路
  - 素子

ディジタルとアナログ



(a) ディジタルな表現　(b) アナログな表現

ディジタルとアナログの比較

ディジタルとアナログ

	ディジタル	アナログ
表現法	数字	別の「量」
ハードウェア量	多いが LSI により高集積化可能	少ない
処理速度	速い	問題による
記憶	容易	むずかしい
持ち運び、通信	容易	むずかしい
コスト	安い。特に汎用マイクロプロセッサ	問題・精度に大きく依存
精度	必要な桁数分のハードウェアを投入して達成する	構成要素（素子）に依存。一般にあまり高くない
雑音	強い	弱い
汎用性	高い	低いー
インタフェース	AD/DA 変換が必要	特別な変換は不要

ディジタル論理回路

- ディジタル論理回路
  - 電子計算機（コンピュータ）の基礎となる回路。通常、0，1の2種類の入出力をもつ（不連続な値）
    - 0： 電圧の低い状態（L）
    - 1： 電圧の高い状態（H）
- ビット (bit, binary digit, b)
  - 情報の単位。0か1の値を取る

2進数　- 数の表現 -

- **n進法でm+1桁の数：**
  - **位取り記数法：** $a_m a_{m-1} \ldots\ldots a_1 a_0$

    $$X = a_m n^m + a_{m-1} n^{m-1} + \ldots\ldots + a_1 n + a_0$$

  - 2進数：　$n = 2,\ a_i = 0, 1$
  - 8進数：　$n = 8,\ a_i = 0, 1, 2, 3, 4, 5, 6, 7$
  - 10進数：$n = 10,\ a_i = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$
  - 16進数：$n = 16,\ a_i = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, A, B, C, D, E, F

数の変換

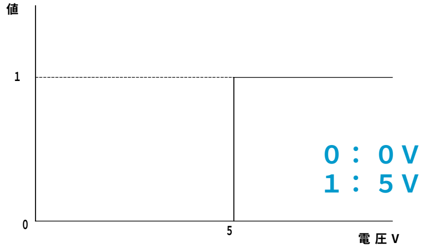
- **2進数⇒10進数**

$$X_{n-1}2^{n-1} + X_{n-2}2^{n-2} + \cdot\cdot\cdot + X_1 2 + X_0 + X_{-1}2^{-1} + X_{-2}2^{-2} + \cdot\cdot\cdot + X_{-m+1}2^{-m+1}$$

の計算を１０進数の加算・乗算のルールで行えばよい。

- **10進数⇒2進数**
  - 10進数を2で割り、商を2で割り、さらに商を2で割り、と、商が1になるまで2で割ることを繰り返し、最後の商、最後の剰余、2番目の剰余、・・・・　最初の剰余と順番に並べればよい。
  - 小数点以下は「2倍にすること」を繰り返して小数点から上にでてきた数（0または1）を並べていく

電圧による2進数の表現



- **電線で2進数を表現する**
  - 電線を束ねて、それぞれの線で「桁」を表現していると考える
- **バイト（B）**
  - 8ビットのこと。2進数で8桁、16進数で2桁

数の数え方

- 数の単位
  - m（ミリ、10^{-3}）、 μ（マイクロ、10^{-6}）、 n（ナノ、10^{-9}）、 p（ピコ、10^{-12}）、f（フェムト、10^{-15}）
  - K（キロ、10^{3}）、 M（メガ、10^{6}）、 G（ギガ、10^{9}）、T（テラ、10^{12}）、 P（ペタ、10^{15}）、E（エクサ、 10^{18}）
- 使用例
  - 現在のLSIに含まれるトランジスタの数
    - 100 M (mega)個 = 1.0 × 10^8個　⇒　1億個のトランジスタ
  - 光が真空中を30cm進むのに要する時間
    - 1 ns (nanosecond)= 10^{-9} s
  - 世界最高速級のコンピュータ：Earth Simulator (NASDA)
    - 40 TFLOPS (teraflops) = 4 × 10^{13} FLOPS

      ⇒　1秒間に40兆回浮動小数点数の計算をする
  - 2011年に予想されるメモリチップの容量
    - 64 Gb (gigabits) = 6.4 × 10^{10} b

      ⇒　64 Gb = 8 GB

負の数の表現

- **負の数： 2の補数表現を使う**
  - **$-X \rightarrow 2^n - X$**
  - **n： 数を表すのに用いるビット数（ワード長）**
    C言語の整数型ならば32

2進数の演算（1ビット）

1ビットの四則演算

X	Y	加算 X＋Y	桁上げ	減算 X－Y	借り	乗算 X×Y	除算 X÷Y
0	0	0	0	0	0	0	－
0	1	1	0	1	1	0	0
1	0	1	0	1	0	0	－
1	1	0	1	0	0	1	1

桁上がりのある加算

X	Y	桁上げ入力	X＋Y	桁上げ出力
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Nビット加算

- 正の数（０を含む）Ｘと正の数Ｙの加算は、各桁のビットについて加算を行い、最上位のビットが１になったら、桁あふれ（オーバフロー、overflow）とみなす
- Ｘ、Ｙのどちらか一方が負の数で他方が正の数だった場合も、各桁のビットについて加算を行えばよい。この場合は、桁あふれは起こらない
- X、Yの両方が負の数の場合も、各桁のビットについて加算を行えばよい。この場合は、最上位のビットが０となったら、桁あふれとみなす

加算の例

0011 + 0010 = 0101（3+2=5）
0011 + 0101 = 1000（3+5→桁あふれ）
(a) 正数＋正数

0011 + 1110 = 0001（3+(-2)=1）
0011 + 1011 = 1110（3+(-5)=-2）
(b)正数＋負数

1101 + 1110 = 1011（-3+(-2) = -5）
1101 + 1010 = 0111（-3+(-6) →桁あふれ）
(c) 負数＋負数

Nビット減算

- 減算は、演算数（引く数）の補数をとって被演算数に加える操作を行えばよい
  - 正数から正数を引く減算は、正数と負数の加算となる
  - 正数から負数を引く減算は、正数と正数の加算となる
  - 負数から正数を引く減算は、負数と負数の加算となる
  - 負数から負数を引く減算は、負数と正数の加算となる

Nビット乗算

(1) 被乗数を２Nビットに符号拡張する

２N〜N+1ビット目までに、下からNビット目の値を入れる（最下位を1ビット目と数えている）

(2)1ビット目からN−1ビット目まで普通に被乗数×乗数の計算をする。

(3) (2)の値からNビット目の乗算結果を引く（乗数が正の場合は(2)で終わり）

乗算の例

```
    00000101
  ×     0011
  ----------
    00000101
   00000101
  +       00
  ----------
    00001111
```

(a)　5×3＝15

```
       00000101
     ×     1101
     ----------
       00000101
  +   000001010
  -------------
          11001
  −     0101
  -------------
       11110001
```

(a)　5×（−3）＝　−15

```
    11111011
  ×     0011
  ----------
    11111011
   11111011
  +       00
  ----------
    11110001
```

(c)　（−5）×3＝−15

```
       11111011
     ×     1101
     ----------
       11111011
  +   111110110
  -------------
       11100111
  −11111011
  -------------
       00001111
```

(d)　（−5）×（−3）＝　15

Nビット除算：正数どうし

(1) 被除数と除数の最上位の１がそろうまで除数を左シフト（２倍）する。このときのシフトの回数をＳとする。

(2) 被除数から除数を引く。

(3) 結果が正なら商に１をたてる。負なら０をたてて、除数を足しもどす。

(4) 除数を１ビット右シフトして、(1)(2)を繰り返す。除数のビットをＳ回右シフトしたときの操作で修了。

Nビット除算：一般形

(1) 被除数を２n－１ビットに符号拡張する。すなわち、２n－１～n＋１ビット目までに、下からnビット目の値を入れる。これをD_0とする。
(2) 除数をn－１ビット左シフトする。すなわち、除数に2^{n-1}をかける。これをD_1とする。
(3) Q＝０とする（商の初期値の設定)。
(4) D_0とD_1の符号が同じか、またはD_0＝０のとき、D_0＝D_0－D_1。そうでないとき、D_0＝D_0＋D_1とする。
(5) 新しいD_0とD_1の符号が同じか、または新しいD_0＝０のとき、Qの一番右のビットを１にする。そうでないとき、Qの一番右のビットを０とする。
(6) D_1を１ビット右シフトする。Qを１ビット左シフトする。
(7) (4)-(6)をn－１回繰り返す。最後に(4)(5)をもう一度行う。
(8) Qの一番右のビットが０のとき、D_0＝D_0＋D_1とする。
(9) 商はQ、余りはD_0となる。なお、このとき余りの符号は除数の符号に合わせてある。

本方式の正当性を確認せよ！

除算の例 (1)

				Q
	000$\boxed{0101}$	D_0		0
−	$\boxed{0011}$000	D_1		
	1101101	負	→	0
		↓	シフト	
＋	0001100			00
	1111001	負	→	00
		↓	シフト	
＋	0000110			000
	1111111	負	→	000
		↓	シフト	
＋	0000011			0000
	00000$\boxed{10}$	正	→	$\boxed{0001}$
	あまり			

(a)　5÷3＝1⋯2

除算の例 (2)

				Q
	000$\boxed{0101}$			0
＋	$\boxed{1101}$000			
	1101101	負	→	1
		↓	シフト	
−	1110100			10
	1111001	負	→	11
		↓	シフト	
−	1111010			110
	1111111	負	→	111
		↓	シフト	
−	1111101			1110
	0000010	正	→	$\boxed{1110}$
＋	1111101			
	$\boxed{1111111}$			
	あまり			

(b)　5÷（−3）＝−2⋯−1
(0101 ÷ 1101=1110 ⋯ 1111)

除算の例 (3)

				Q
	111$\boxed{1011}$			0
＋	$\boxed{0011}$000			
	0010011	正	→	1
		↓	シフト	
−	0001100			10
	0000111	正	→	11
		↓	シフト	
−	0000110			110
	0000001	正	→	111
		↓	シフト	
−	0000011			1110
	1111110	負	→	$\boxed{1110}$
＋	0000011			
	$\boxed{0000001}$			
	あまり			

(c)　(−5）÷3＝−2⋯1
(1011 ÷ 0011=1110 ⋯ 0000)

除算の例 (4)

			Q
	1 1 1 $\boxed{1\,0\,1\,1}$		0
−	$\boxed{1\,1\,0\,1}$ 0 0 0		
	0 0 1 0 0 1 1	正 →	0
		↓ シフト	
+	1 1 1 0 1 0 0		0 0
	0 0 0 0 1 1 1	正 →	0 0
		↓ シフト	
+	1 1 1 1 0 1 0		0 0 0
	0 0 0 0 0 0 1	正 →	0 0 0
		↓ シフト	
+	1 1 1 1 1 0 1		0 0 0 0
	$\boxed{1\,1\,1\,1\,1\,1\,0}$	負 →	$\boxed{0\,0\,0\,1}$
	あまり		

(d) （−５）÷（−３）＝１…−２

（1011 ÷ 1101=0001 … 1110）

2進数の演算：例題 ＝ 宿題

- アナログ方式の温度計とディジタル方式の温度計の特徴を比較せよ。
- 次の2進数を１０進数に変換せよ
  101011,　0.11,　110.101
- 次の１０進数を2進数に変換せよ。また、それぞれを１６進数に変換せよ。
  25,　102.5,　40.675, 0.4
- 引き放し法による除算の仕組みを解説せよ。