ディジタル電子回路

第1回

講義のねらい（シラバスから抜粋）

論理回路の動作を理解する上で必要な基礎理論に習熟した後、各種論理ゲート、ディジタル演算回路への応用について講義する。コンピュータアーキテクチャの基礎となる2進法とそれらを用いた論理関数、および論理回路を組み合わせた演算回路の動作が理解できることを目標とする。

参考書：藤井信生『ディジタル電子回路』昭晃堂
：一色 剛, 熊澤 逸夫 『論理回路 』 数理工学社

【成績評価】
演習と試験による。
配点は，演習20点，中間試験40点，期末試験40点

講義のねらい

論理回路の動作を理解する上で必要な基礎理論に習熟した後、各種論理ゲート、ディジタル演算回路への応用について講義する。コンピュータアーキテクチャの基礎となる2進法とそれらを用いた論理関数、および論理回路を組み合わせた演算回路の動作が理解できることを目標とする。

じゃ、コンピュータの仕組みが判る？

命令セット　から　実際にコンピュータを動作させるには
コンピュータアーキテクチャの概念が必要

でもお金を入れられたら、それを表示して、ボタンを押されたら、ものを出して、おつりを渡す自動販売機なら作れる。

他の講義との関連

線形回路　3学期、　回路理論　4学期
回路理論の基礎

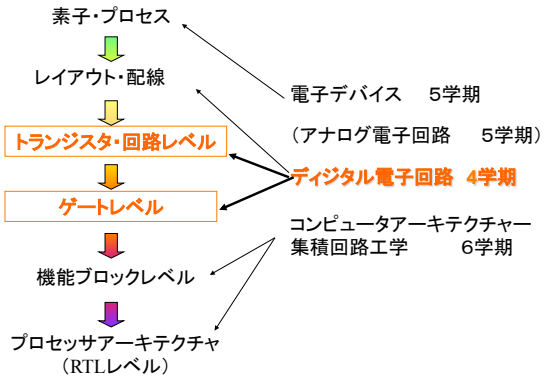
半導体物性　　4学期
半導体の基礎、pn接合

電子デバイス　　5学期
pn接合、MOSFET の基礎

アナログ電子回路　　5学期
トランジスタによる線形増幅回路、帰還回路、演算増幅器、発振回路

コンピュータアーキテクチャ　6学期
集積回路設計　　6学期
計算機のハードウェア、プロセッサの機能と設計手法

講義と論理LSI製造工程との関連



アナログ回路からデジタル回路へ

ディジタル回路（デジタル回路）⟷　アナログ回路

digital：（形）数字で計算する、計数式の、デジタル式の
数字式の、指の、指状の

回路の2つの動作状態を扱う。
・電流が流れているかどうか？
・電圧が高いか、低いか？　　電流、電圧の値にはよらないことが多い。

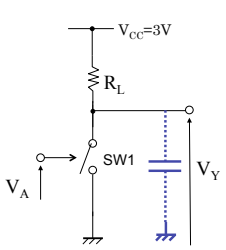
問：最も簡単に2値動作を実現するには？

スイッチの2値動作

電流の大小が問題ではなく、単にスイッチがオン、オフどちらの状態にあるかが重要。

スイッチによるディジタル回路の例

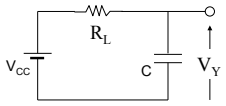
以下のようなスイッチと抵抗R_Lからなる回路を考える。このスイッチは外部からの電圧により制御されており、V_{CC}が印加された時にオン、0Vの時にはオフになる。出力電圧はどうなるか？



V_A	V_Y
0	V_{CC}
V_{CC}	

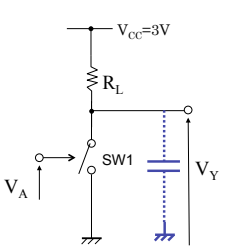
A	Y
0	1
1	

SW1がオフの時



スイッチによるディジタル回路の例

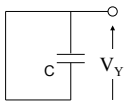
以下のようなスイッチと抵抗R_Lからなる回路を考える。このスイッチは外部からの電圧により制御されており、V_{CC}が印加された時にオン、0Vの時にはオフになる。出力電圧はどうなるか？



V_A	V_Y
0	V_{CC}
V_{CC}	0

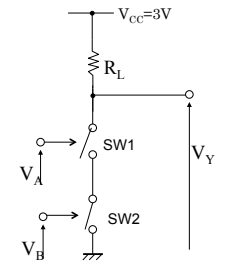
A	Y
0	1
1	0

SW1がオンの時



スイッチによるディジタル回路の例

以下のようなスイッチ1（SW1）、スイッチ2（SW1）、抵抗R_Lからなる回路を考える。これらのスイッチは外部からの電圧により制御されており、V_{CC}が印加された時にオン、0Vの時にはオフになる。表のように電圧が印加されたとき、出力電圧はどうなるか？



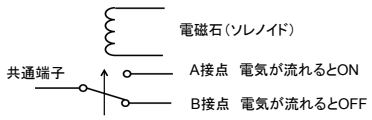
V_A	V_B	V_Y
0	0	V_{CC}
0	V_{CC}	V_{CC}
V_{CC}	0	V_{CC}
V_{CC}	V_{CC}	0

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

接点回路

スイッチは外部からの電圧により制御すること
電磁石により可能！　リレーと呼ばれている。

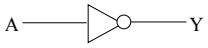
電磁石でスイッチを引っ張る



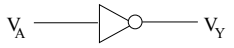
リレーのA接点、B接点、ソレノイドから構成されるラダーロジックと呼ばれる回路もある。
現在でもシーケンス制御/プログラマブルロジック等の
プログラマブルロジックコントローラ(PLC)はこのラダー言語で記述される例が多い
論理回路そのものは、トランジスタやダイオードとは関係がない

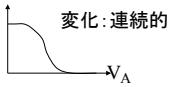
理想（論理回路理論）と実際

論理回路理論	実際の実現
論理値：	物理量：例えば電圧・電流
“1” (“T”:True、“H”:High)	“5V”（程度）
“0” (“F”: False、“L”:Low)	“0V”（程度）



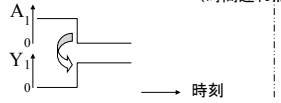
A	Y	変化：離散的
0	1	0→1
1	0	1→0



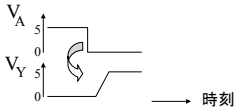


変化：連続的

変化に要する時間：一瞬
（時間遅れ無）



変化に要する時間：遅れる



2進符号による数の表現

10進数：0, 1, 2, …9の10種類の記号で数を表す。

16進数：0, 1, 2, …9, A, B, …F の16種類の記号で数を表す。
(10, 11, 15)　（昔の機械語はこれで書いていた）

2進数：0, 1の2種類の記号で数を表す。

<問題>

10進数	16進数	2進数
3	3	11
8	8	1000
28	1C	11100

2進符号による数の表現

10進数　$N = a_n a_{n-1} a_{n-2} \cdots a_0$

$$N = a_n \times 10^n + a_{n-1} \times 10^{n-1} + a_{n-2} \times 10^{n-2} \cdots + a_0 \times 10^0$$

2進数　$N_2 = (a_n a_{n-1} a_{n-2} \cdots a_0)_2$

$$N = a_n \times 2^n + a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} \cdots + a_0 \times 2^0$$

＜例＞
2進数の1011は10進数でいくつか？

$$N = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11$$

2進符号の1桁をビット（binary digitの略）という。

少数の2進数による表現

10進数の場合と同様に小数点を用いて

$$N_2 = (.\ a_{-1} a_{-2} \cdots a_{-m})_2$$

$$N = a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \cdots + a_{-m} \times 2^{-m}$$

従って小数も含めて一般には

$$N_2 = (a_n a_{n-1} \cdots a_0 .\ a_{-1} a_{-2} \cdots a_{-m})_2$$

重みのより大きいビット：上位ビット
重みのより小さいビット：下位ビット

重みの最も大きいビット：最上位ビット　MSB (most significant bit)
重みの最も小さいビット：最下位ビット　LSB (least significant bit)

負の数の2進数による表現

10進数では通常 “－”記号　を数字の前につけて表している。

2進数ではどうするか？

（1）最上位ビットの上にさらに1ビット設け、このビットの値が0の時に正の数を1の時に負の数を表すようにする。（符号付2進数）。

（2）負数を2の補数で表す。

負の数の2進数による表現

（1）　符号＋絶対値で表現する。すなわち最上位ビットの上にさらに1ビット設け、このビットの値が0の時に正の数を1の時に負の数を表す。

10進数	符号付き2進数
-7	1111
-6	1110
-5	1101
-4	1100
-3	1011
-2	1010
-1	1001
-0	1000

10進数	符号付き2進数
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

0が2通りできる。

最上位ビットの0を省略しない。（最上位ビットの0を省略すると、110 は、先頭の1が符号で、残りの10が絶対値なのか、先頭の0が省略されていて、絶対値が110なのか、区別がつかない。

全体で何ビット使うか決めておく。

演算が以外と大変

負の数の2進数による表現

（2）負数を2の補数（2^n-x）で表す。

最上位ビットと正負の関係の確認

正の数a+正の数bの和で最上位ビットへ繰り上がり　なっては駄目

正の数a+負の数で絶対値がxの和で、解が正(a>x)
$a+2^n$(負を表す正の最上位ビット)+ (2^n-x)(2の補数)=2^{n+1}+a-x

正の数a+負の数で絶対値がxの和で、解が負(x>a)
$a+2^n$(負を表す正の最上位ビット)+ (2^n-x)(2の補数)
=2^n(負を表す最上位ビット)+{2^n-(x-a)}(2の補数)

負の数で絶対値がxと負の数で絶対値がyの和 x+y
(2^n)(負を表す正の最上位ビット)+ (2^n-x)(2の補数)
+(2^n)(負を表す正の最上位ビット)+ (2^n-y)(2の補数)
=2^{n+1}+2^n(負を表す最上位ビット)+{2^n-(x+y)} (2の補数)

さっき無視した最上位ビットからの繰り上げ

負の数の2進数による表現

（2）負数を2の補数（2^n-x）で表す。

10進数	4ビット符号付き2進数
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111

10進数	4ビット符号付き2進数
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

加減算はそのまま行えばよい。
たとえば、$(-6) + 2 = (1010)_2 + (0010)_2 = (1100)_2 = -4$
$(-3)+(-2) = (1101)_2 + (1110)_2 = (1011)_2 = -5$
（最上位ビットからの繰り上げは無視：11011は1011）

補足

符号付2進数（負の数は**2の補数**）の作り方

1～8の正の数を$(10000)_2$から引いて、－1～－8を作る。 2^n-x

＜例＞ $-6=(10000)_2-(0110)_2=(1010)_2$
$-8=(10000)_2-(1000)_2=(1000)_2$

または、1～8の正の数のビットを反転させて1を足す。(2^n-1)-x+1

＜例＞ －6 は +6 $(0110)_2$ の ビット反転$(1001)_2$に1を足して $(1010)_2$

10進数	4ビット符号付き2進数
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111

10進数	4ビット符号付き2進数
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

2値が最も効率的か？

p進法、n桁で表現できる整数の数：$N=p^n$
この時の数字の総数 ： $W=p \times n$
N一定の条件下でWの最小値を求めると・・・

$$\ln N = n \ln p \qquad W = p \times n = p \times \frac{\ln N}{\ln p}$$

$$\frac{dW}{dp} = \frac{d}{dp}\left(\frac{p \ln N}{\ln p}\right) = \frac{\ln N}{(\ln p)^2}(\ln p - 1) = 0$$

p=e=2.71・・・ つまり3進数が一番効率が良い。

＜問題＞
2進法15桁、3進法10桁、10進法3桁で表せる整数の個数を求めよ。

2^{15}=32,768　　3^{10}=59,049　　10^3=1,000

論理関数と論理式

論理関数：2つの状態”1”または”0”をとる変数A,B,C,…と、これらの変数と変数を関連づけるいくつかの記号を用いて表現される関数 f(A,B,C,…)

＜例＞ f(A,B)=A+B

OR（論理和）

真理値表
数字1，0を用いて2値回路の動作状態を表した表

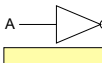
A	B	f(A,B)
0	0	0
0	1	1
1	0	1
1	1	1

論理変数：A,B,C,…（論理関数に用いられる変数）
論理記号：＋，・ など
論理式 ：論理変数と論理記号で表した式

基本演算 NOT、AND、OR

1．NOT（否定回路）

入力A	出力f(A)
0	1
1	0

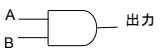


$f(A)=\overline{A}$　$\overline{A}$をAのNOT（否定）という。
－ はNOT演算を表す論理記号。

2．AND（論理積）回路

A	B	出力f(A,B)
0	0	0
0	1	0
1	0	0
1	1	1

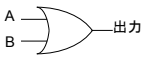
f(A,B)=A・B
・がAND演算を表す論理記号



3．OR（論理和）回路

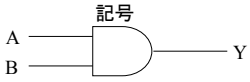
A	B	出力
L	L	L
L	H	H
H	L	H
H	H	H

f(A,B)=A＋B
＋がOR演算を表す論理記号



AND

記号



論理式による表現

Y= A・B

(Y=A ∧ B、Y= A B とも書く)

真理値表

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

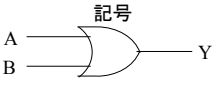
f(A, B, C,…)＝A・B・C・…のように
n個の変数のANDも可能

⇩

すべての変数が1の時だけ1となる
1つでも0になると0を出力

OR

記号



論理式による表現

Y= A + B

(Y=A ∨ B とも書く)

真理値表

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

f(A, B, C,・・)＝A＋B＋C＋…のように
n個の変数のORも可能

⇩

すべての変数が0の時だけ0となる
1つでも1になれば1を出力

ANDとORの比較

AND回路

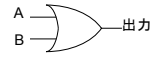
A	B	出力
L	L	L
L	H	L
H	L	L
H	H	H

LとHを入れ替え

A	B	出力
H	H	H
H	L	H
L	H	H
L	L	L

OR回路

A	B	出力
L	L	L
L	H	H
H	L	H
H	H	H



正論理と負論理

正論理

Hレベル → 1
Lレベル → 0

負論理

Hレベル → 0
Lレベル → 1

<例> AND回路

A	B	出力
L	L	L
L	H	L
H	L	L
H	H	H

正論理

A	B	出力
0	0	0
0	1	0
1	0	0
1	1	1

負論理

A	B	出力
1	1	1
1	0	1
0	1	1
0	0	0

正論理のOR

演算の順序

NOT、AND、ORには演算の優先順位がある。
（通常の数字の計算と同様）

AND演算はOR演算に先立って行う。

<例> $f = A + B \bullet C$

B･Cを先に求め、それからAとのORを求める。

$f = (A + B) \bullet C$

（ ）により演算の順序を変更可能。
A+Bを先に求め、それとCのANDをとる。

$f = \overline{A + B}$

NOTは単項の演算。
A+Bを求めてから、そのNOTを求める。

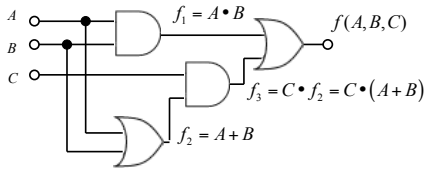
論理回路を論理式で表現する

論理回路が与えられる ⟶ 論理式

$f_1 = A \bullet B \qquad f_2 = A + B$

$f_3 = C \bullet f_2 = C \bullet (A + B)$

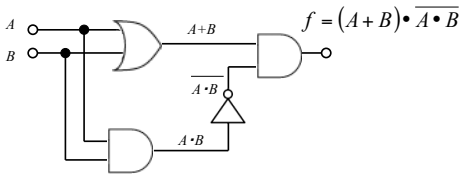
$f(A,B,C) = f_1 + f_3 = A \bullet B + C \bullet (A + B)$



論理式を論理回路で表現する

論理式が与えられる ⟶ 論理回路を構成

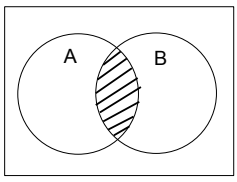
$f = (A + B) \bullet \overline{A \bullet B}$



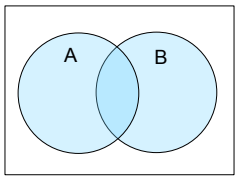
（注） 論理関数が与えられた場合、これを実現する論理回路は1種類ではない。
与えられた論理式は、様々な形に変形可能。

論理式とVenn図

AをAの円内で表す。



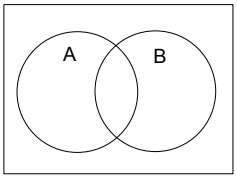
A･B



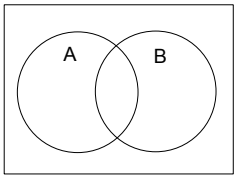
A+B

論理式とVenn図

$\overline{A}\cdot B + A\cdot\overline{B}$ はどの領域か？

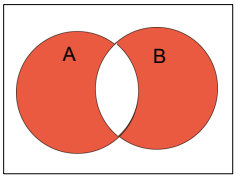


$A\cdot B + \overline{A}\cdot\overline{B}$ はどの領域か？

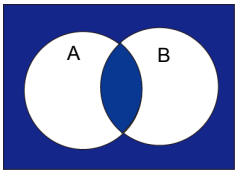


論理式とVenn図

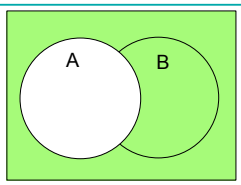
$\overline{A}\cdot B + A\cdot\overline{B}$ はどの領域か？



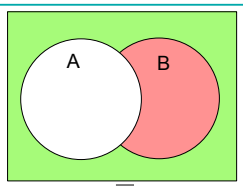
$A\cdot B + \overline{A}\cdot\overline{B}$ はどの領域か？



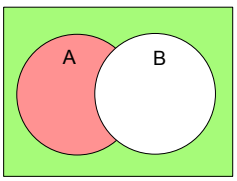
論理式とVenn図



$\overline{A}$



$\overline{A}\cdot B$



$A\cdot\overline{B}$