デジタル（２進数）での加算・乗算

群馬大学
小林春夫

デジタル回路と2進数

- 人間はなぜ10進数を使うか？

  ⇒ 手の指が10本あるから。

- デジタルではなぜ2進数を使うか？

  ⇒ 2つの状態は技術的に容易かつ安定して実現可能。

例：　電圧の高いと低い

電流の流れる状態と流れない状態

パルスのあるとなし。

今か324年前、1692年のパリ

哲学者、数学者、科学者　ライプニッツ

（**Gottfried Wilhelm Leibniz**）

「全ての数を1と0によって表す驚くべき表記法」

を提案。

王立科学アカデミーに理解されず

学会誌にも掲載されなかった。

「誰も予想しなかった卓越した用途がありはずだ」

と語る。

ゴットフリート・ヴィルヘルム・ライプニッツ（Gottfried Wilhelm Leibniz, 1646年 - 1716年）

ライプニッツは哲学者、数学者、科学者など幅広い分野で活躍した学者・思想家として知られているが、また政治家であり、外交官でもあった。17世紀の様々な学問（法学、政治学、歴史学、神学、哲学、数学、経済学、自然哲学（物理学）、論理学等）を統一し、体系化しようとした。その業績は法典改革、モナド論、微積分法、微積分記号の考案、論理計算の創始、ベルリン科学アカデミーの創設等、多岐にわたる。ライプニッツは稀代の知的巨人といえる。

10進数と2進数

10進	2進	10進	2進
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

例　2進数　1011　を

10進数に変換

1x2x2x2+0x2x2+1x2+1

=11

デジタル加算

2進数の加算		10進数の加算
0011	(3)	437
+) 1011	(11)	+) 258
1110	(14)	695

2入力2進加算

0+0=00

0+1=01

1+0=01

1+1=10

3入力2進加算

0+0+0=00

0+0+1=01

0+1+1=10

1+1+1=11

論理積(AND)

論理変数　A,B, Z

A,B: 入力,　Z: 出力

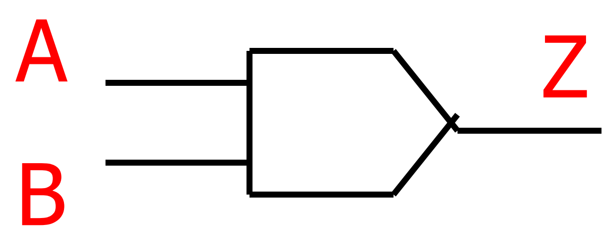
Z=　A･B

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

真理値表

ANDを実現する回路

→ AND回路



論理和(OR)

論理変数　A,B, Z

A,B: 入力,　Z: 出力

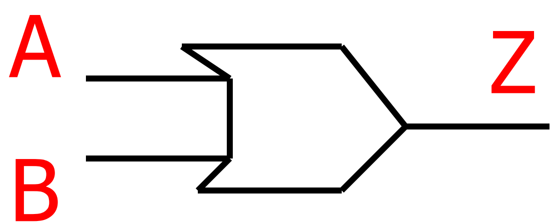
$Z = A + B$

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

真理値表

ORを実現する回路

→ OR回路



論理和と2進数の加算

論理和　Z=　A+B

右の真理値表は

論理和の定義

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

真理値表

論理和　　　　1+1=1

2進数の加算　1+1=10

同じ 記号 + でも意味が異なることに注意。

デジタル加算器の実現(1)
（半加算器； Half Adder）

2入力2進加算

A	0	1	0	1
+) B	+) 0	+) 0	+) 1	+) 1
Co S	0 0	0 1	0 1	1 0

真理値表

A	B	Co	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

1ビット加算器の実現

$$S = A \oplus B$$

$$Co = A \cdot B$$

デジタル加算器の実現(2)
（全加算器； Full Adder）

3入力2進加算

A　入力1

B　入力2

+) Cin　下からの繰り上がり

Co S

$S = A \oplus B \oplus Cin$

$Co = B \cdot Cin + A \cdot Cin + A \cdot B$

(Co は A, B, Cin の多数決)

真理値表

A	B	Cin	Co	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
1	0	0	0	1
0	1	1	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

全加算器 (Full Adder)の補足説明

Cin: Carry in (下位の桁からの繰り上げ)

S: Sum（加算結果）

Cout: Carry out （上位の桁への繰り上げ）

$$\begin{array}{rrr} & 0 & 1 \\ +) & 1 & 1 \\ \hline 1 & 0 & 0 \end{array}$$

を考える。

全加算器
の演算

$$\begin{array}{rrr} & \boxed{\begin{array}{c} 1 \\ 0 \\ 1 \end{array}} & \begin{array}{c} \\ 1 \\ 1 \end{array} \\ +) & & \\ \hline 1 & 0 & 0 \end{array}$$

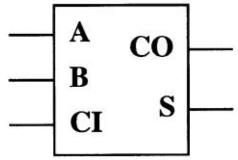
全加算器の補足説明(続き)

		0	1	0	1	(5)
+)		0	1	1	1	(7)
	0	1	1	0	0	(12)

を考える。

		1	1	1		Carry（桁上げ）
		0	1	0	1	
+)		0	1	1	1	
	0	1	1	0	0	Sum（加算結果）

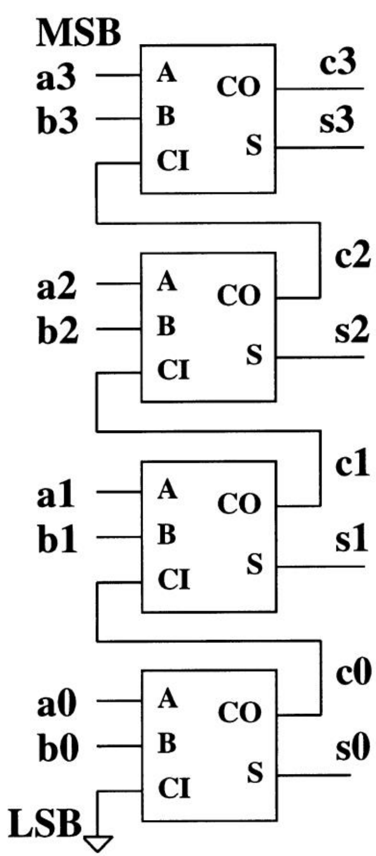
Full Adder



A	B	CI	S	CO
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	0	0	1	0
0	1	1	0	1
1	1	0	0	1
1	0	1	0	1
1	1	1	1	1

CI Carry In
CO Carry Out
S SUM

Adder & Carry Propagation



Full Adderで、A,B,CIの値が確定し
から COの値が確定するまでの遅延時
をTとする。
左図で、a3~a0, b3~b0の値が
確定してから
C0の値が確定するまで T
C1 〃 2T
C2 〃 3T
C3 〃 4T かかる。
→ Carryの伝播がスピードのネックにな

Example :

$$\begin{array}{r} 0\,1\,1\,1 \\ +\ 0\,1\,0\,1 \\ \hline 1\,1\,0\,0 \end{array}$$

⇓

(a3 a2 a1 a0) = (0 1 1 1)
(b3 b2 b1 b0) = (0 1 0 1)

(s3 s2 s1 s0) = (1 1 0 0)
(c3 c2 c1 c0) = (0 1 1 1)

デジタル乗算

２進数の乗算

```
      0101   (5)
   x) 1011  (11)
   -----------------
      0101
     0101
    0000
   0101
   -----------------
   0110111    (55)
```

10進数の乗算

```
      437
   x）258
   ----------
     3496
    2185
    874
   ----------
   112746
```

デジタル乗算器の直接的構成

Multiplier

$$X = \sum_{i=0}^{m-1} X_i 2^i$$

$$Y = \sum_{j=0}^{n-1} Y_j 2^j$$

$$P = X \times Y = \sum_{i=0}^{m-1} X_i 2^i \cdot \sum_{j=0}^{n-1} Y_j 2^j$$

$$= \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (X_i Y_j)\, 2^{i+j}$$

$$= \sum_{k=0}^{m+n-1} P_k 2^k.$$

TABLE 8.2 4-bit Multiplier Partial Products

				X3	X2	X1	X0	Multiplicand
				Y3	Y2	Y1	Y0	Multiplier
				X3Y0	X2Y0	X1Y0	X0Y0	
			X3Y1	X2Y1	X1Y1	X0Y1		
		X3Y2	X2Y2	X1Y2	X0Y2			
	X3Y3	X2Y3	X1Y3	X0Y3				
P7	P6	P5	P4	P3	P2	P1	P0	Product

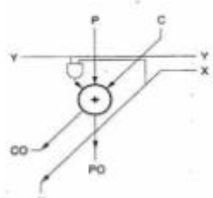
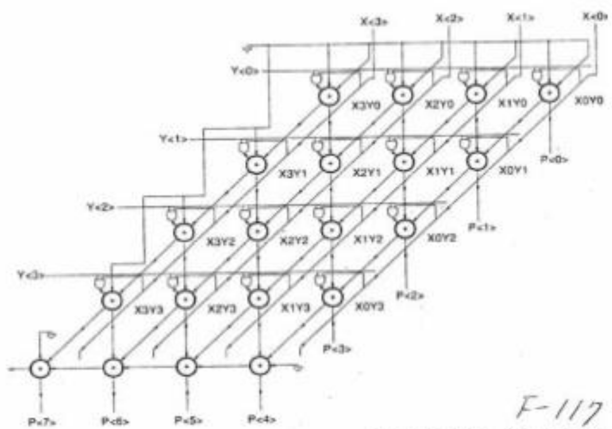


FIGURE 8.35 Array multiplier cell



F-117

FIGURE 8.36 A 4 × 4 array multiplier