

演算機能回路

第2週

2006/10/09

泉知論

立命館大学 理工学部 電子情報デザイン学科

符号無整数

unsigned integer

二進数

- “0” と “1” の組合せで値を表現する。
- 電子回路的には GND と VDD で表現する。
- 通常、固定ビット長で表す。
- 4bit, 8bit, 32bit, 64bit など、2のべき乗の長さを用いることが多いが、例外も多い。
- 1単位長のビット列を“ワード”と呼ぶ。

符号無整数

- unsigned int と呼ばれる。
 - n-bit ワードの第i番目のビットの値を x_i とすると 値 = $\sum_{i=0}^{n-1} x_i \times 2^i$
- 8bit 符号無し整数の例

x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
$\times 2^7$	$\times 2^6$	$\times 2^5$	$\times 2^4$	$\times 2^3$	$\times 2^2$	$\times 2^1$	$\times 2^0$

例: 00000000 (2進) = 0 (10進), 00001000 (2進) = 8 (10進),
10000000 (2進) = 128 (10進), 11111111 (2進) = 255 (10進)

- 最下位($i=0$)ビットを LSB (Least Significant Bit) と呼ぶ。
- 最上位($i=n-1$)ビットを MSB (Most Significant Bit) と呼ぶ。

符号無整数(2)

- n bit で表現できる値は $0 \sim 2^n - 1$ の 2^n とおり
- 逆に N 未満の値を表現するには $\lceil \log_2 N \rceil$ ビット必要

※ $2^{10} = 1024 \div 1000$ なので、一般には“キロ”というと1000倍だが、デジタル分野では1024倍を表すことが多い。“メガ”、“ギガ”、“テラ”も同様。

演習2A

- 次の符号無し2進数は10進数では値はいくらになるか
 1. 1000000000
 2. 00000011
 3. 11111101
- 動画のフレームに通し番号をつけたい。毎秒24フレームで最大3時間の動画を対象とする。何ビット必要か？

加算器

(順次桁上加算器)

adder (ADD)

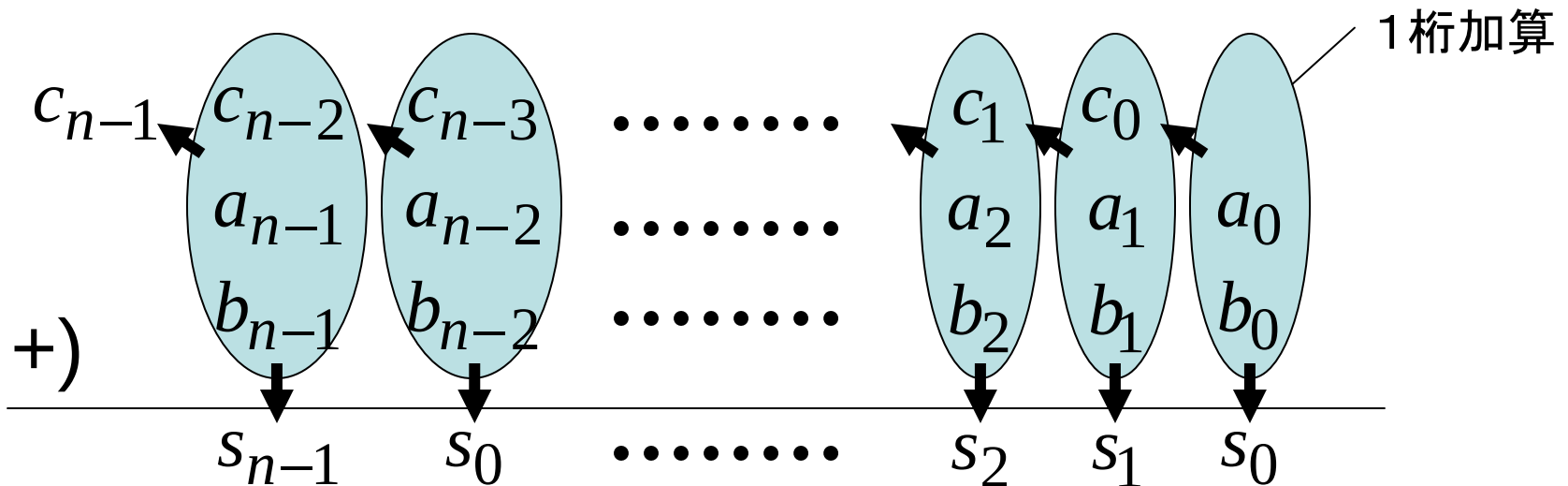
ripple carry adder (RCA)

順次桁上加算

- 筆算の手順を踏めばできる！

$$A = a_{n-1}a_{n-2} \cdots a_0 \quad B = b_{n-1}b_{n-2} \cdots b_0$$

として、第*i*桁の“繰り上がり”(carry)を c_i とすると...



全加算器(1)

- 1桁分の加算
- 全加算器 (full adder)
- s_i は排他的論理和
- c_i は多数決

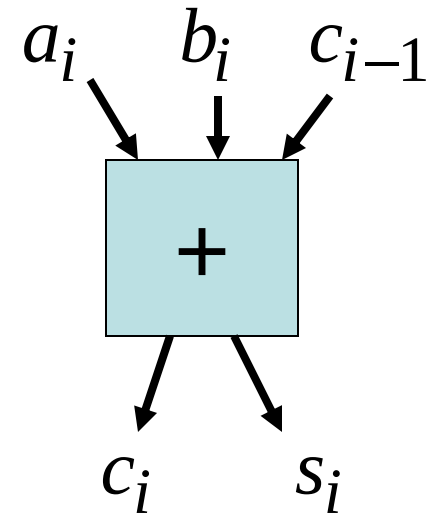
a_i	b_i	c_{i-1}		c_i	s_i
0	0	0		0	0
0	0	1		0	1
0	1	0		0	1
0	1	1		1	0
1	0	0		0	1
1	0	1		1	0
1	1	0		1	0
1	1	1		1	1

全加算器(2)

- 1桁分の加算

$$s_i = a_i \oplus b_i \oplus c_{i-1}$$

$$c_i = a_i b_i + a_i c_{i-1} + b_i c_{i-1}$$

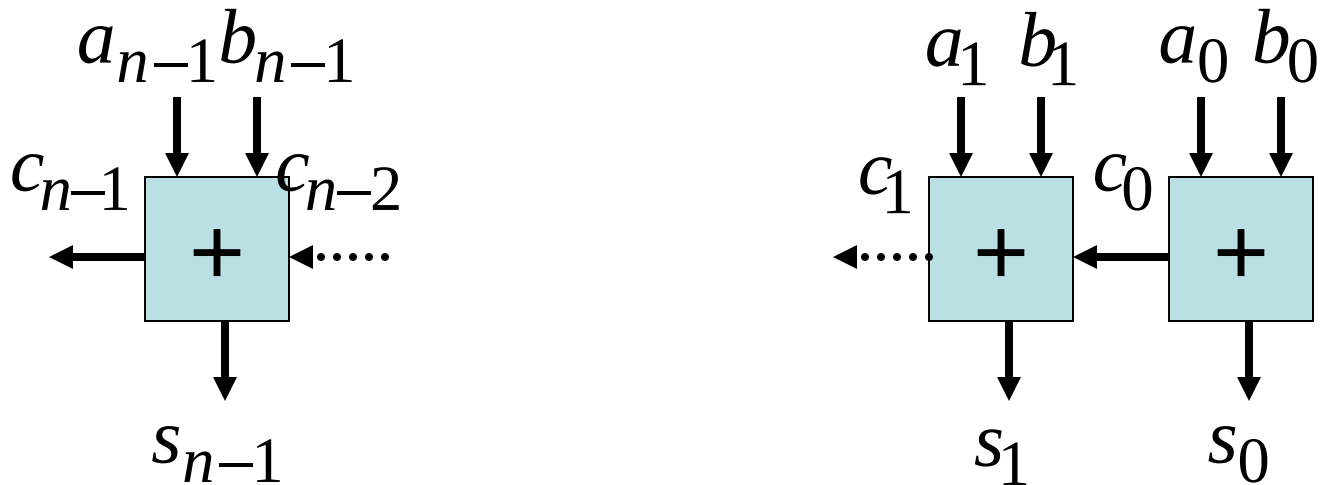


演習2B

- 1bit 全加算器(和と桁上げ)をAND, OR, NOT からなる論理回路として設計せよ。

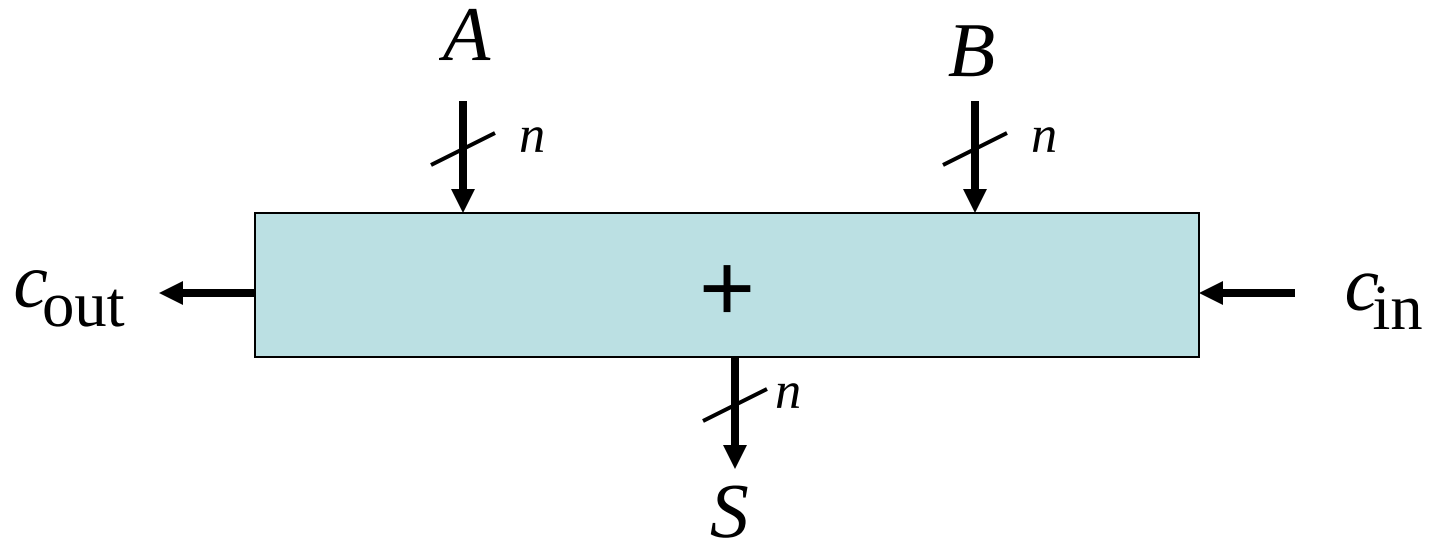
順次桁上加算回路

- nビット順次桁上加算器
- Ripple Carry Adder と呼ばれる



順次桁上加算回路

- n ビット桁上伝播回路



- 複数接続することでビット幅を拡張できる
- 最下位の c_{in} には 0 を入れる
- 最上位の c_{out} が 1 $\rightarrow$ 桁あふれ (over flow)

演習2C

- 次の8bit符号無整数の加算をせよ
 1. 00000001+00000001
 2. 00001101+00000011
 3. 11111111+00000001

2の補数符号付整数

signed integer

2の補数符号付整数

- signed int と呼ばれる。
- n-bit ワードの第i番目のビットの値を x_i とすると

$$\text{値} = -x_{n-1} \times 2^{n-1} + \sum_{i=0}^{n-2} x_i \times 2^i$$

X7	X6	X5	X4	X3	X2	X1	X0
0 なら正 1 なら負	$\times 2^6$	$\times 2^5$	$\times 2^4$	$\times 2^3$	$\times 2^2$	$\times 2^1$	$\times 2^0$

- 最上位($i=n-1$)ビットを符号ビットと呼ぶ。

演習2D

- 次の8ビット2の補数符号付整数の表す値を10進数で答えよ。
 1. 00000000
 2. 00001000
 3. 10000000
 4. 11111111

2の補数符号付整数(2)

- n bit で表現できる値は $-2^{n-1} \dots + 2^{n-1} - 1$ の 2^n とおり
※ただし -2^{n-1} は使用禁止にするほうが無難。
符号反転ができない。
- 逆に $\pm N-1$ 未満の値を表現するには
 $1 + \lceil \log_2 N \rceil$ ビット必要
- 正の数の個数より負の数の個数がひとつ多いことに注意

減算器

(加減算器)

subtractor (SUB)

2の補数符号付整数(3)

- 符号反転=ビット反転+1

$$\bar{X} + 1 = -\bar{x}_{n-1} \times 2^{n-1} + \sum_{i=0}^{n-2} \bar{x}_i \times 2^i + 1$$

$$= -(1 - x_{n-1}) \times 2^{n-1} + \sum_{i=0}^{n-2} (1 - x_i) \times 2^i + 1$$

$$= -2^{n-1} - x_{n-1} \times 2^{n-1} - \sum_{i=0}^{n-2} x_i \times 2^i + \sum_{i=0}^{n-2} 2^i + 1$$

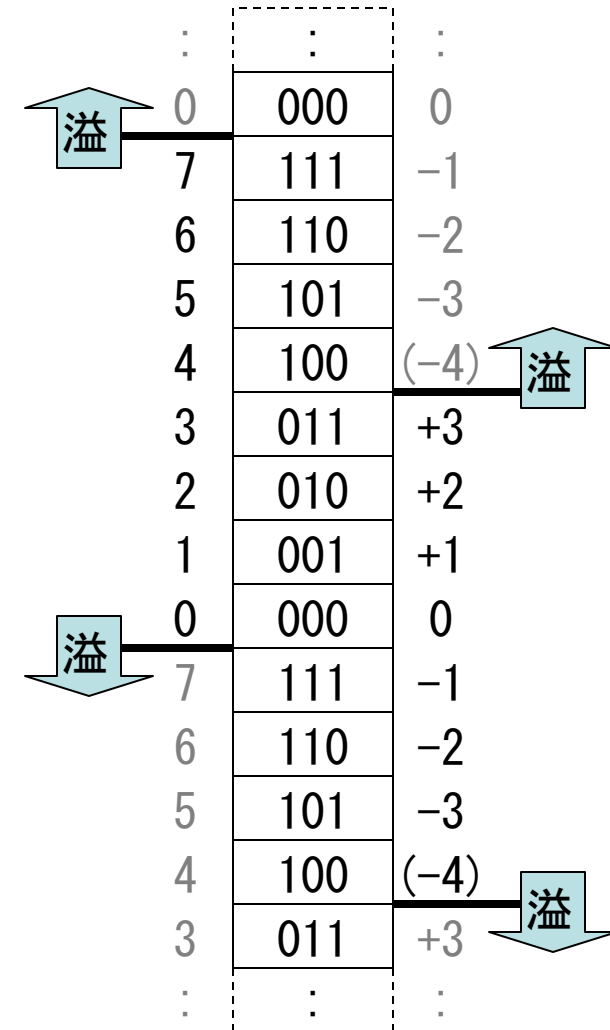
$$= - \left(-x_{n-1} \times 2^{n-1} + \sum_{i=0}^{n-2} x_i \times 2^i + 1 \right)$$

$$= -X$$

2の補数符号付整数(4)

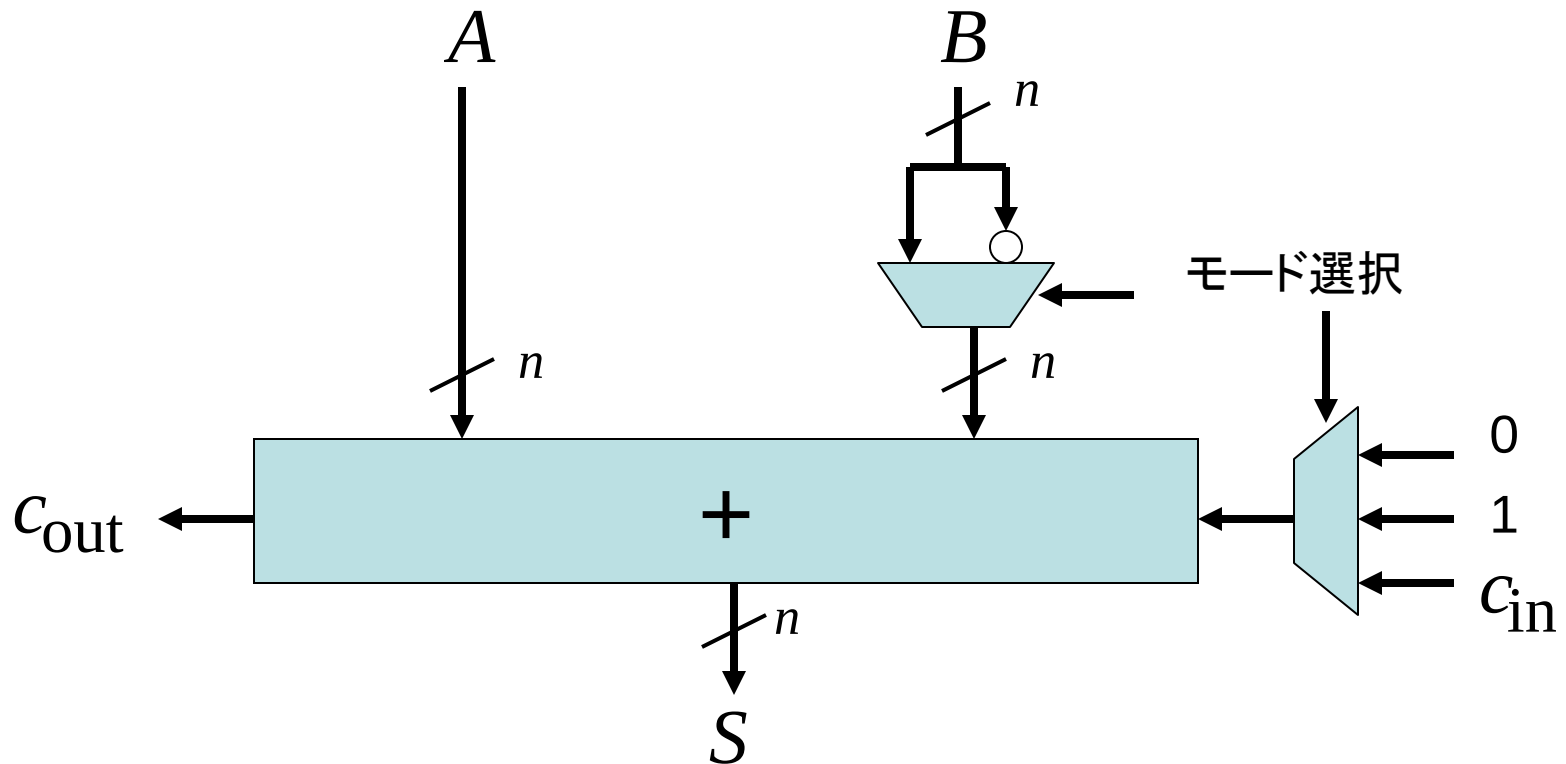
- なぜ、素直に符号無整数に符号を表す1ビットを加えるだけ(1の補数)にしないのか？
- 2の補数は 2^n を法とする剰余に基づく
 → $-x$ と $+(2^n - x)$ は等価
 → 符号無整数の加算がそのままつかえる！

以後、2の補数符号付整数を単に符号付整数あるいは整数と呼ぶ



加減算器

- nビット加減算器



- 選択器(セレクタ)で B 入力と c_{in} 入力を選択する

演習2E

- 次の8bit符号付整数の加算、減算をせよ
 1. 00000001, 00000001
 2. 00001101, 00000011
 3. 11111111, 00000001

加減算結果の判定

- ゼロ(一致)

$$\text{zero} = s_{n-1} + s_{n-2} + \cdots + s_1 + s_0$$

- 符号(大小比較)

$$\text{sgn} = s_{n-1}$$

- 桁あふれ(無効)

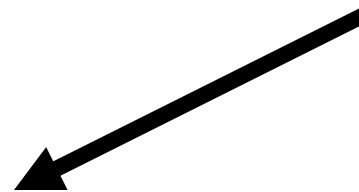
- 符号無

$$\text{overflow} = c_{out}$$

- 符号付

$$\text{overflow} = a_{n-1}b_{n-1}\bar{s}_{n-1} + \bar{a}_{n-1}\bar{b}_{n-1}s_{n-1}$$

A	B	S	結果
正	正	正	○
負	負	負	○
正	負	*	○
負	正	*	○
正	正	負	溢
負	負	正	溢



演習2F

- zero, sgn, overflow を計算する論理回路を AND, OR, NOT で設計せよ。