



2進数補数サマリ

負の数を正の数で処理する（減算を実行するとき）

人間： 大脳の場合、コスト・空間は不要

マシン： 「[加算器](#)」に加えた「[減算器](#)」のコスト、デバイス内空間が必要

⇒加算器に減算をさせる工夫

⇒4,000年間続く数学のメインストリームとは異質





10進数の補数 一桁電卓を想定

- 定義された値 0~9
- $6 + 3 = 9$ …ある
- $6 + (-6) = 0$ …ある…電卓は 0 を表記 ①
- $6 + 4 = 10$ …ない…電卓は 0 を表記 ②
- ①②より (-6) と (+4) は同等に扱える
- ポイント
 - |-6| に 3 を加えると 9 定義された最大値
 - 3 に 1 を加えて「桁上がり発生」となり、4 が (-6) 補数になる



2進数の補数 四桁電卓を想定

- 定義された値 0~1
- 10進数 $(-6)_{10}$ の場合 $|-6| = (6)_{10} \Rightarrow (0110)_2$
- **0, 1 逆転** $\Rightarrow 1001$ $\cdots (0110)$ に加えると 1111  +1 で桁上がり
- なので、逆転した値に 1 を加える
- **$1001 + 1 = 1010$** $\cdots (0110)$ に加えると 10000  5桁目表記不能
電卓表記は **0000**
- 従って $(-6)_{10}$ の2進数は **$(1010)_2$**





2進数補数の表記

- $(-6)_{10} \Rightarrow (1010)_2$
- 「先頭ビットが 1 の場合に負の数」宣言がある場合、補数と考える
- 確認 $(1010)_2$ 先頭ビットが 1 なので負が確定、次は絶対値
- 0, 1 逆転 $\Rightarrow 0101$
- 逆転した値に 1 を加える
- $0101 + 1 = (0110)_2$ $\hookrightarrow (6)_{10}$ である
- $(1010)_2$ は負の値 6 即ち -6



2進数減算を加算で実行

- $(-6)_{10} \Rightarrow (1010)_2$
- $8 - 6 \Rightarrow (1000)_2 + (1010)_2$
- 1000
- + 1010
- 10010  5桁目表記不能 $(0010)_2$  $(2)_{10}$



2進数減算を加算で実行

- $(-6)_{10} \Rightarrow (1010)_2$
- $4 - 6 \Rightarrow (0100)_2 + (1010)_2$
- 0100
- $+ \underline{1010}$
- 1110 ☞先頭ビットが **1** なので負が確定 次は絶対値
- $0, 1$ 逆転 $\Rightarrow 0001$
- 逆転した値に 1 を加える
- $0001 + 1 = (0010)_2$ ☞ $(2)_{10}$ である
- $(1110)_2$ は負の値 2 即ち -2

