

演算機能回路

第7週

2006/11/13

泉知論

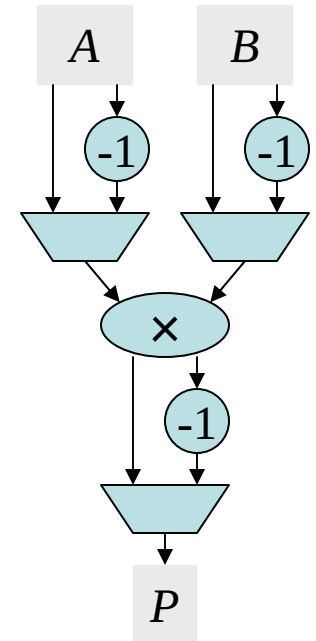
立命館大学 理工学部 電子情報デザイン学科

乘算器

multiplier

乗算器(単純実装)

- $P = A \times B$
- 筆算の手順をそのまま組合せ回路化すれば実現可能
- 符号の扱い
 - P の符号 = A の符号 $\times$ B の符号
 - $|P| = |A| \times |B|$ ※符号無しの値に対する乗算
- 以下では、符号無し整数の乗算器を考える。
固定小数点も同様。小数点の位置に注意。

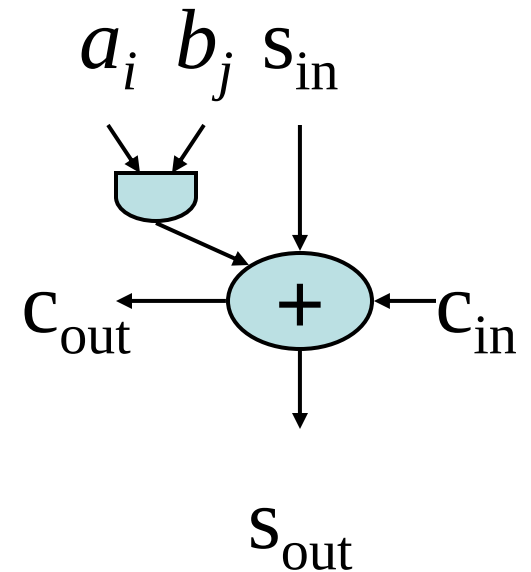


乗算(単純筆算)

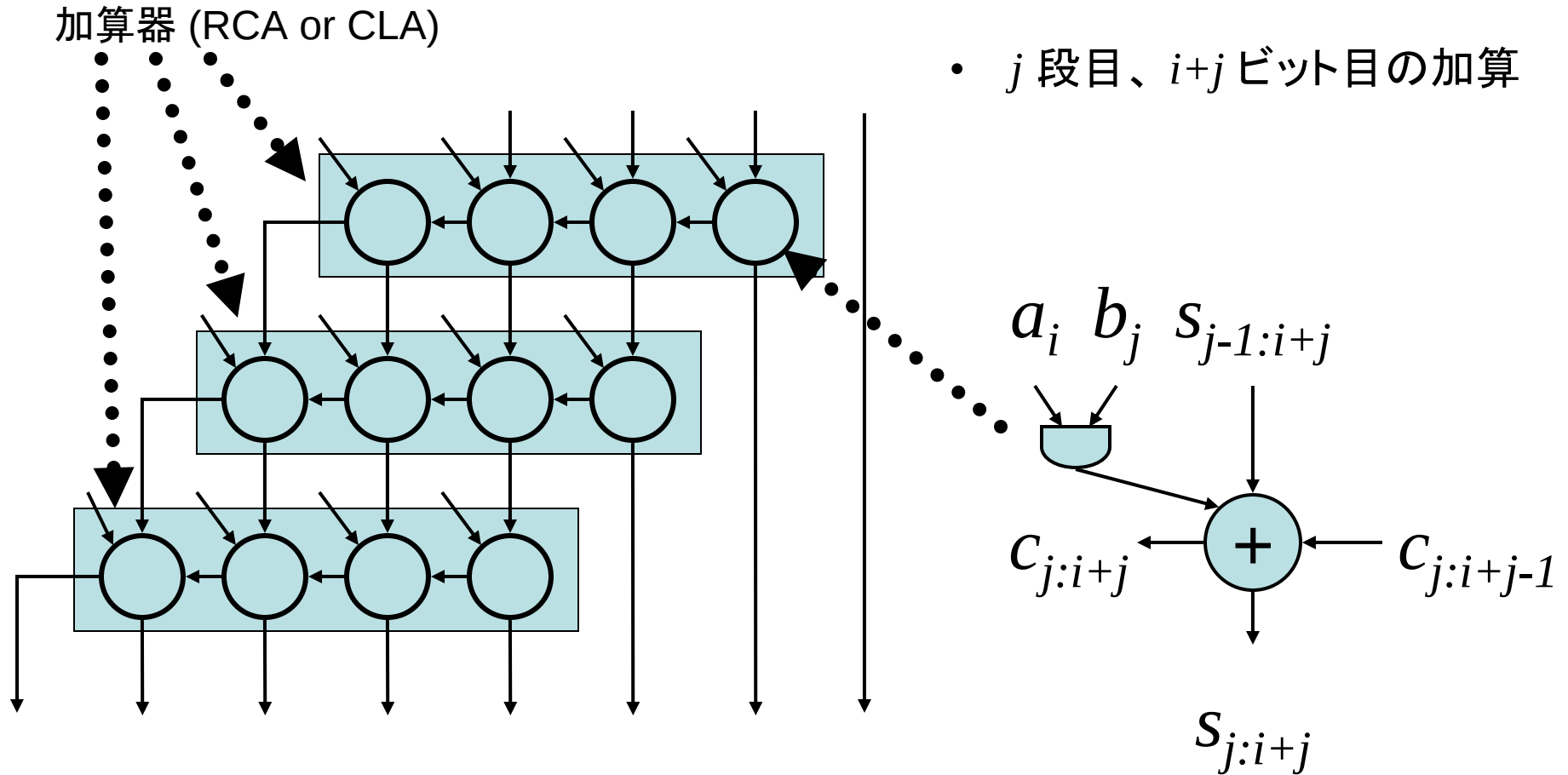
- 4ビット×4ビットの例

$$\begin{array}{r}
 \times \qquad \qquad \qquad a_3 \ a_2 \ a_1 \ a_0 \\
 \hline
 \qquad \qquad \qquad b_3 \ b_2 \ b_1 \ b_0 \\
 \hline
 \qquad \qquad \qquad a_3 \ a_2 \ a_1 \ a_0 \times b_0 \\
 + \qquad \qquad \qquad a_3 \ a_2 \ a_1 \ a_0 \times b_1 \\
 + \qquad \qquad \qquad a_3 \ a_2 \ a_1 \ a_0 \times b_2 \\
 + \qquad \qquad \qquad a_3 \ a_2 \ a_1 \ a_0 \times b_3 \\
 \hline
 p_7 \ p_6 \ p_5 \ p_4 \ p_3 \ p_2 \ p_1 \ p_0
 \end{array}$$

各段、各ビットの加算

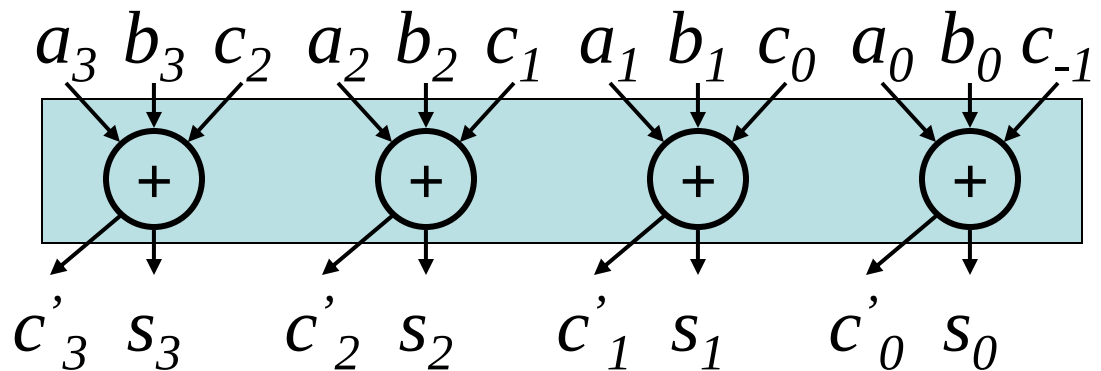


乗算器(単純実装)

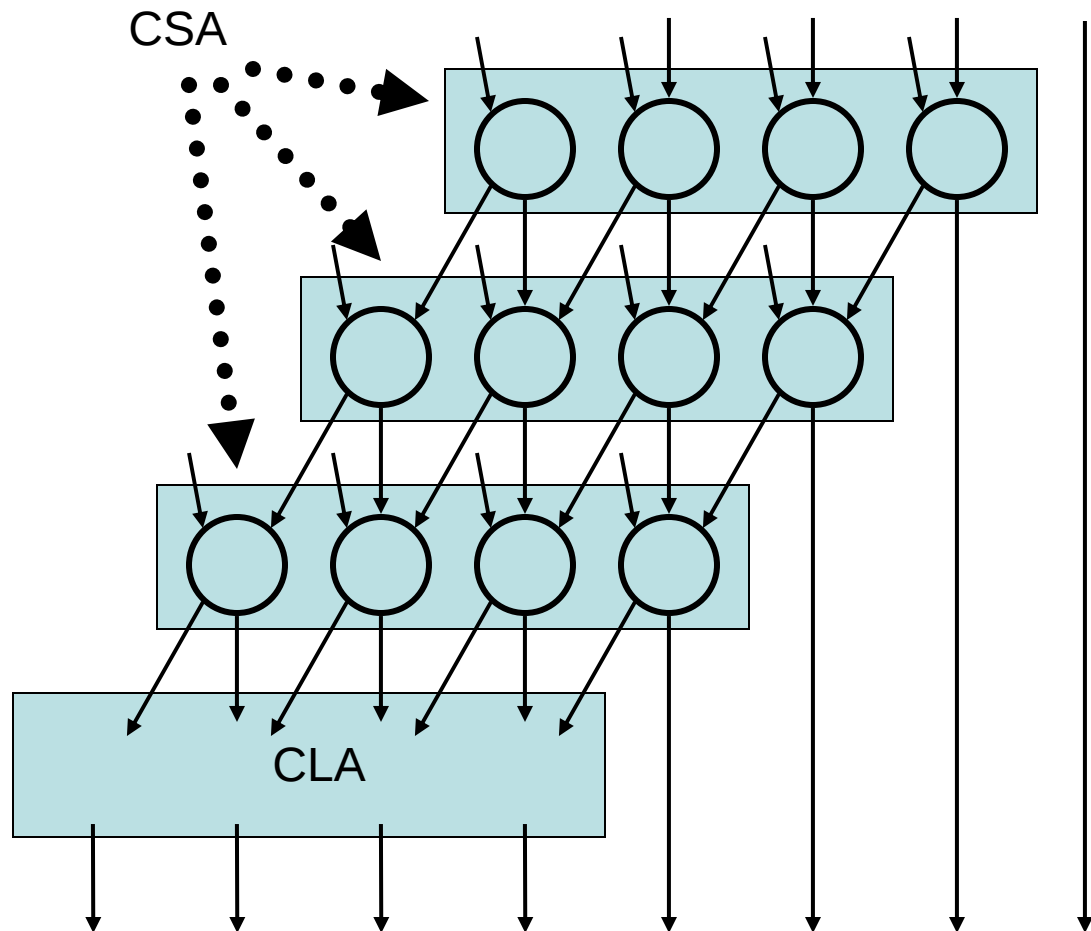


桁上保存加算器

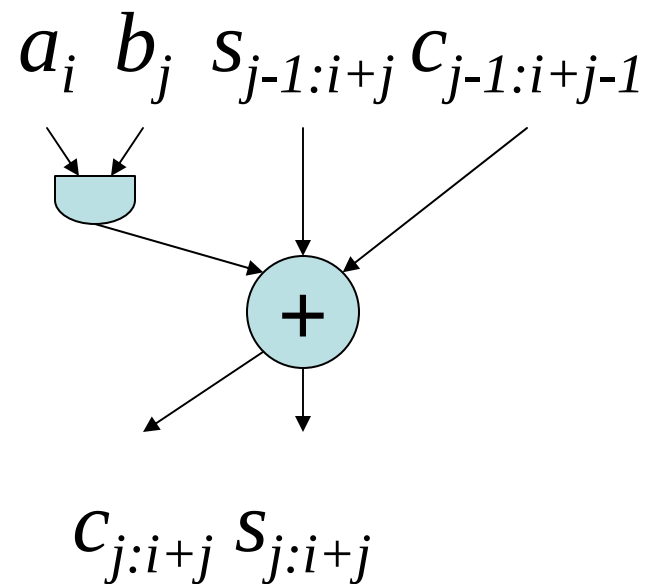
- 桁上げを伝播させる加算器 を用いた実装
 - RCA を使うと遅い
 - CLA を使うと大きい
- 桁上保存加算器(Carry Save Adder, CSA)
 - 桁上を伝播させずにそのまま出力する加算器
 - 最終的には桁上伝播加算器が必要
 - その分、小さくて速い
 - 中間値計算に有用



乗算器 (CSA による単純実装)



- j 段目、 $i+j$ ビット目の加算

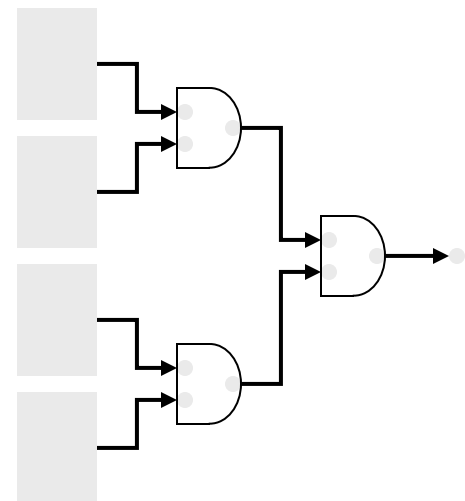
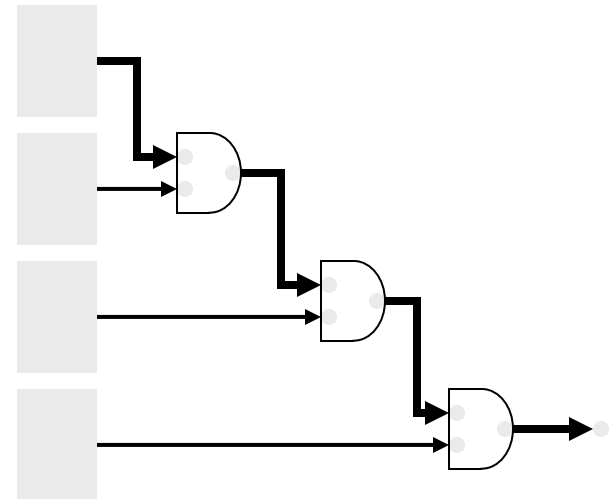


CSA 部分の改良

- CLA 部分の段数が $O(\log n)$ でも、このままでは CSA の部分の段数が $O(n)$
- CSA の部分を組み替えてみる
 - 加算なので**組合せ**も**順番**も**自由**に決めてよい。

改良のヒント

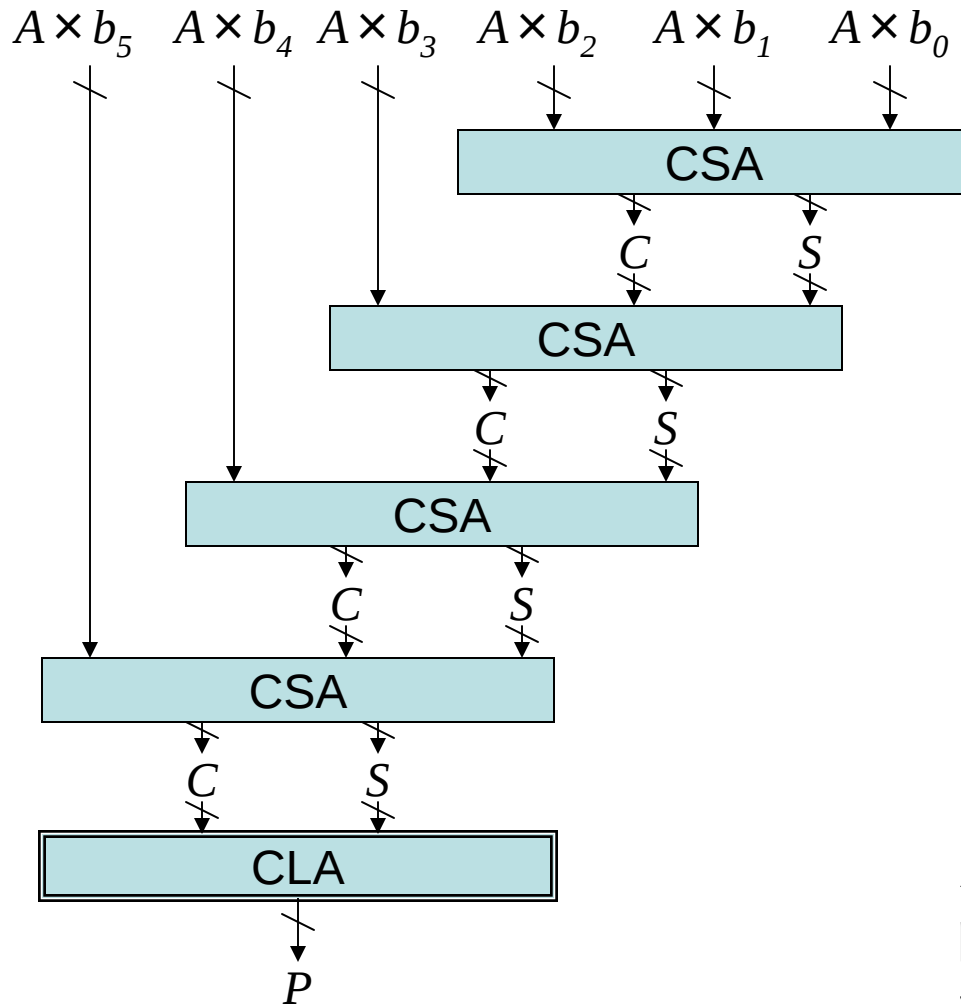
- 一列に計算すると
各段でひとつずつ減って
 $O(n)$
- 各段で2個ずつまとめていくと
各段で半分ずつに減って
 $O(\log_2 n)$



Wallace Tree

- 各段でできるだけ3つずつの組にする
- 3つの入力→2つの出力
- 中間結果の個数が1段でおおよそ $2/3$ に減る
- 段数は $O(\log_{1.5} n)$

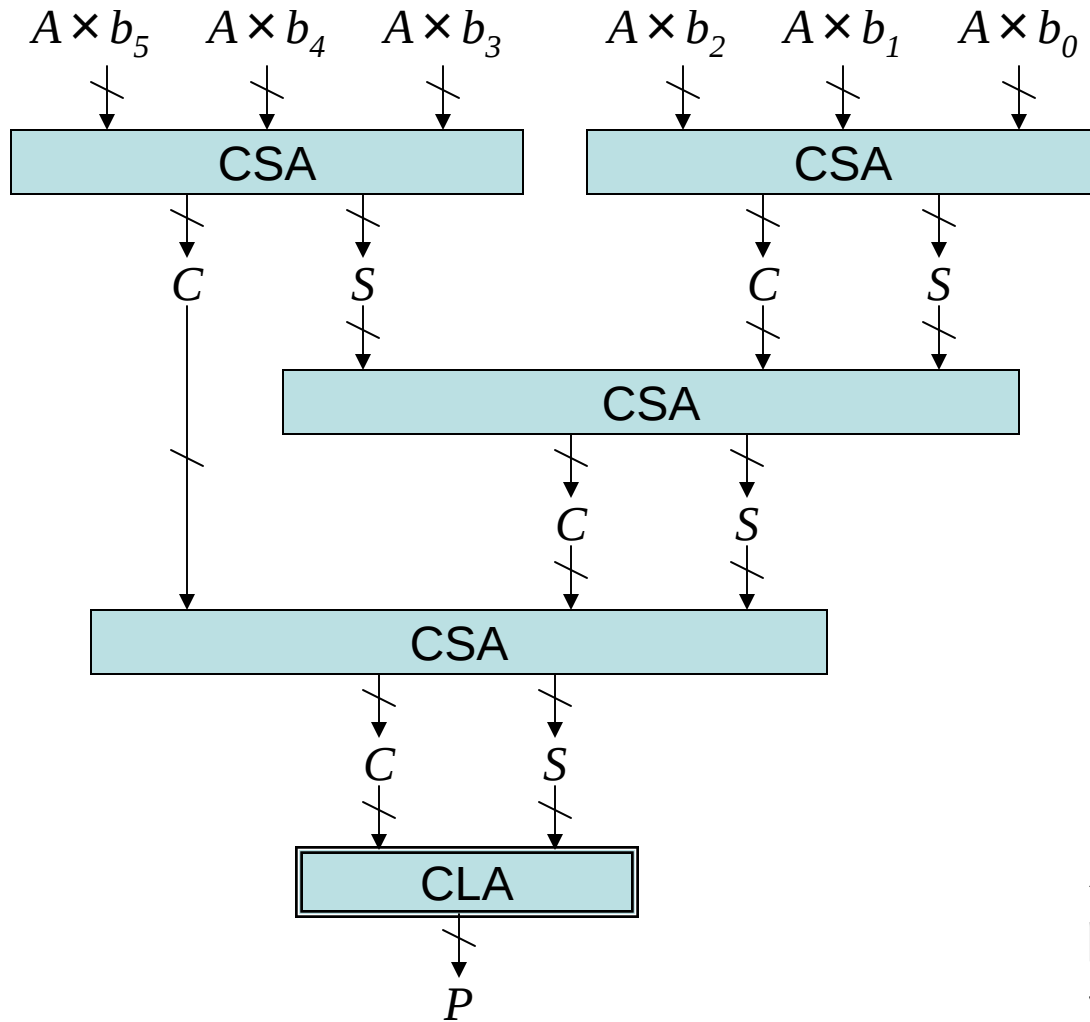
一列に加算する状態



6bit × 6bit 乗算器の例

注: 桁あわせが必要。下位 0 padding。上位符号拡張。最終的には縮退する。

Wallace Tree による乗算器



6bit × 6bit 乗算器の例

注: 桁あわせが必要。下位 0 padding。上位符号拡張。最終的には縮退する。

定数乗算器

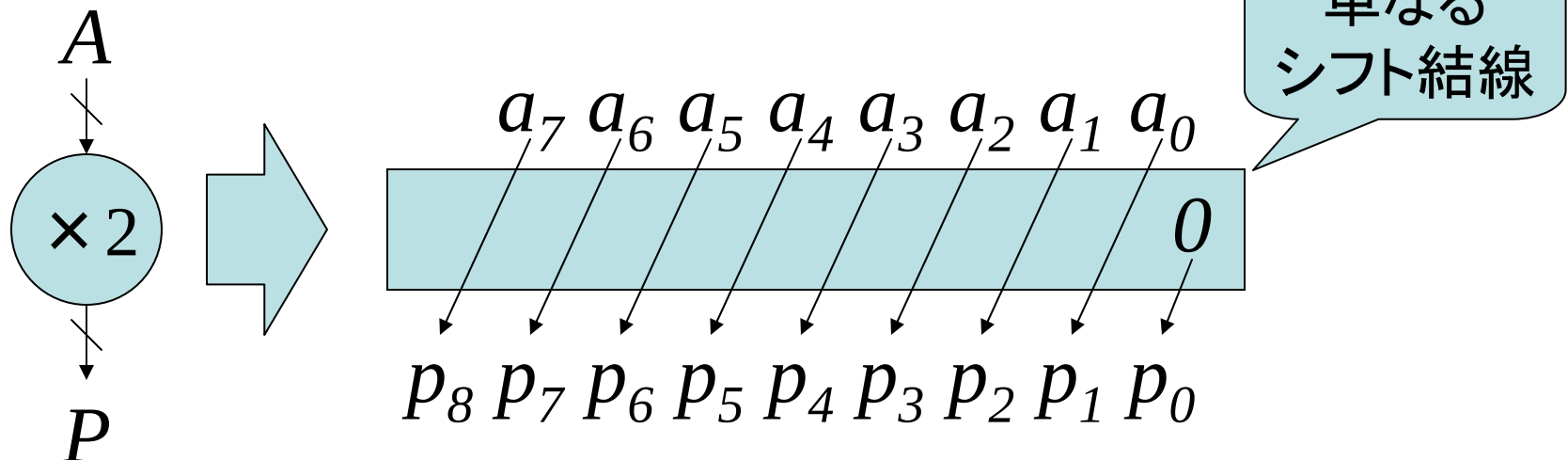
- 定数係数の乗算：専用回路では需要がある

例 $\frac{1}{4}A + \frac{3}{4}B, \sqrt{2}X, \frac{X}{\sqrt{2}}, 2\pi R, \dots$

※定数除算は逆数の定数乗算

- 乗算器は大きくて遅いが...

－例えば、2倍の定数乗算器は？



定数乗算

- $b_j=1 \rightarrow A$ を j ビット右シフトしたものを加算
- B のなかで 1 であるビットに対応するもののみを加算すればよい。
- 定数中に 1 の数が少なければ効果的
例: $\times 2$ 、 $\times 4$ 、 $\times 8$ 、... $\rightarrow$ 単なる結線
例: $\times 100 \rightarrow$ 二進では 1100100 で加算2回
- しかし、1 の数が多いと...
例: $\times 1000 \rightarrow$ 二進では 1111101000 で加算5回？

定数乗算の工夫

- 連続する 1 は減算 1 回で表現可能

例: $\times 15 \rightarrow$ 二進では 1111

$$A \times 15 = A \times (16 - 1) = A \times 16 - A$$

- 連続する 1 の間の 0 も減算 1 回で表現可能

例: $\times 1000 \rightarrow$ 二進では 1111101000

$$10000000000 - 0000001000 = 1111111000$$

$$1111111000 - 0000010000 = 1111101000$$

※減算の際の符号拡張に注意

※減算は符号反転と +1 であり、Wallace Tree
では c_{in} が使えないので加算が 1 回増える

演習7A

- 8bit 符号なし整数に対する100倍の定数乗算器を設計せよ。出力は符号なし 16bit とせよ。部品として CSA と CLA を部品として用い、Wallace Tree を用いて段数を削減せよ。

演習7B

- $\sqrt{3}$ を二進数で表すと約 1.1011101110 である。
- $\frac{\sqrt{3}}{2}$ を整数部 0bit、小数部 8bit の固定小数点値で表せ。小数点以下第9位で0捨1入せよ。
- 60度回転 $\frac{1}{2}A + \frac{\sqrt{3}}{2}B$ の計算には加減算が何回必要か？係数はそれぞれ整数部 0bit、小数部 8bit の0捨1入で表現するものとする。

演習7C

- 符号 1bit、整数部 0bit、小数部10bit の固定小数点値に対する1000倍の定数乗算器を設計せよ。出力は符号 1bit、整数部10bit の固定小数点値とし、下位ビットは単純切捨てとする。CSA と CLA を部品として用い、Wallace Tree を用いて段数を削減せよ。減算は反転と+1で表現すること。符号拡張部分の記載は省略してよい。