2019. 6.24

電子計算機工学

Ibaraki Univ. Dept of Electrical & Electronic Eng.

Keiichi MIYAJIMA

演算アーキテクチャ

-固定小数点数の算術演算回路-

２進加算

前に述べたとおり、コンピュータの内部でのデータは2進数で表現される。

- ２進数の加算のみで減算も表現できた。

↓

実際の回路構成はどうなっているのか？

２進加算

下位からの桁上げがない場合の2進数一桁の加算の真理値表を考える。

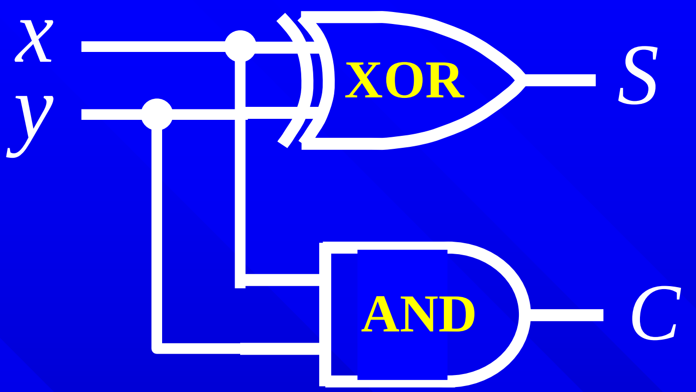
入力		出力	
		和	桁上げ出力
x	y	S	C
０	０	０	０
０	１	１	０
１	０	１	０
１	１	０	１

下位からの桁上げを考慮しない加算

半加算器(half adder : HA)

半加算器

論理記号



論理回路

下位からの桁上げを考慮しない加算

半加算器(half adder : HA)

2進加算

下位からの桁上げを考慮する場合

$$
\begin{array}{r}
11 \\
+)\ 11 \\
\hline
1\boxed{1}0
\end{array}
$$

この場所では下位からの桁上げがあるため、
それを考慮する必要がある

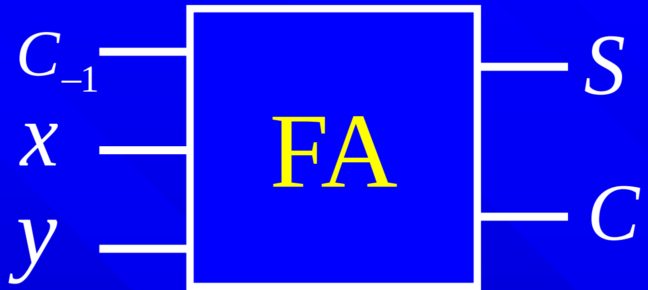
全加算器

下位からの桁上げを考慮する場合

入力			出力	
			和	桁上げ出力
x	y	C_{-1}	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

全加算器

論理記号



半加算機とORゲートによる構成

下位からの桁上げを考慮した加算器

全加算器(full adder：FA)

注

「減算器」についてはここでは述べない。

なぜなら、前に述べたとおり、減算は2の補数を用いることで、加算として取り扱うことができる。

桁上げ伝搬加算機

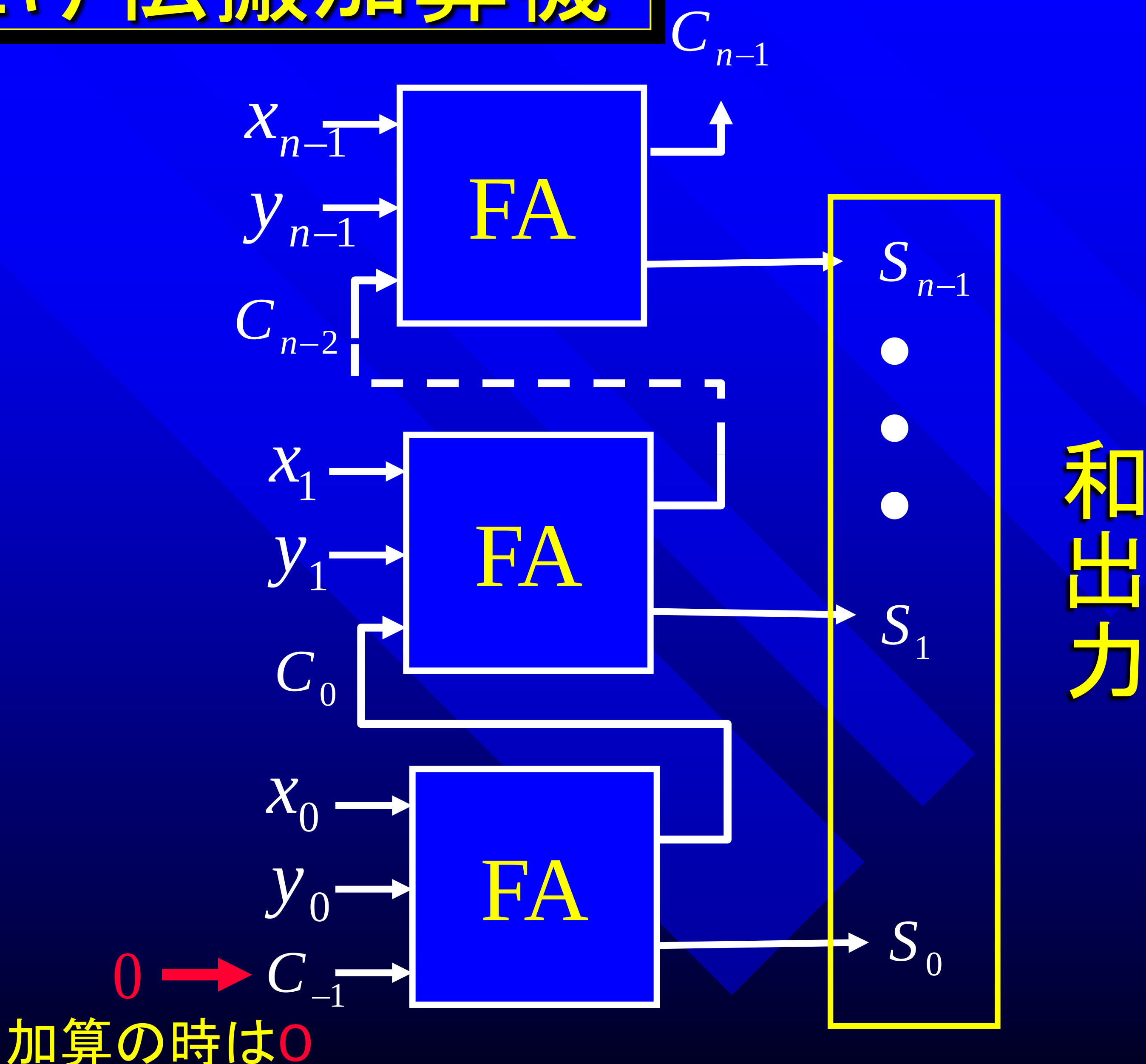
桁上げ伝搬加算器 すべてのビットを同時に加算する
（Carry Ripple Adder）

↓

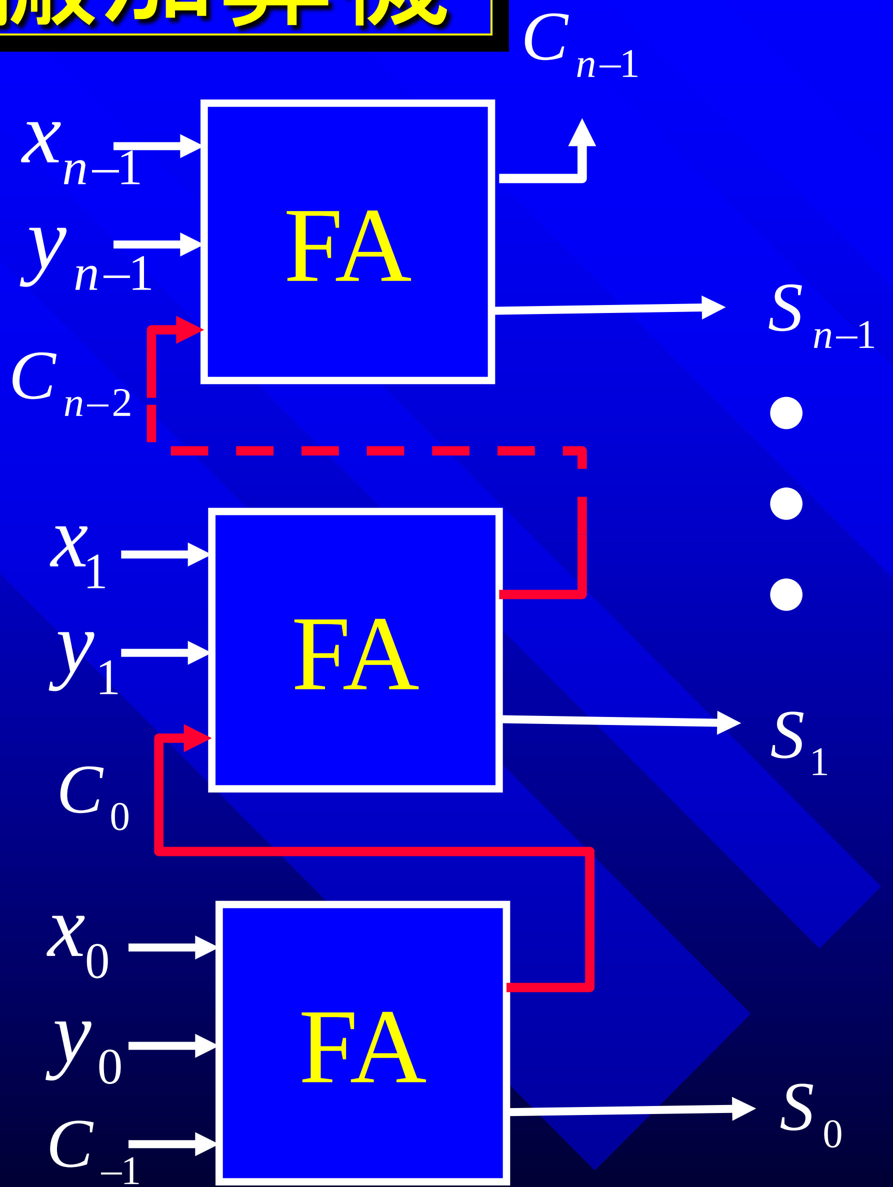
ハードウェア量は増加するが、

加算の高速化が構成可能

桁上げ伝搬加算機



桁上げ伝搬加算機



桁上げが伝達されるのに時間がかかる。

桁上げ先見加算器 （Carry Look-Ahead Adder）

$$G_k = x_k \bullet y_k \qquad P_k = x_k + y_k$$ とおくと

$$C_0 = G_0 + P_0 C_{-1}$$

$$C_1 = G_1 + P_1 G_0 + P_1 P_0 C_{-1}$$

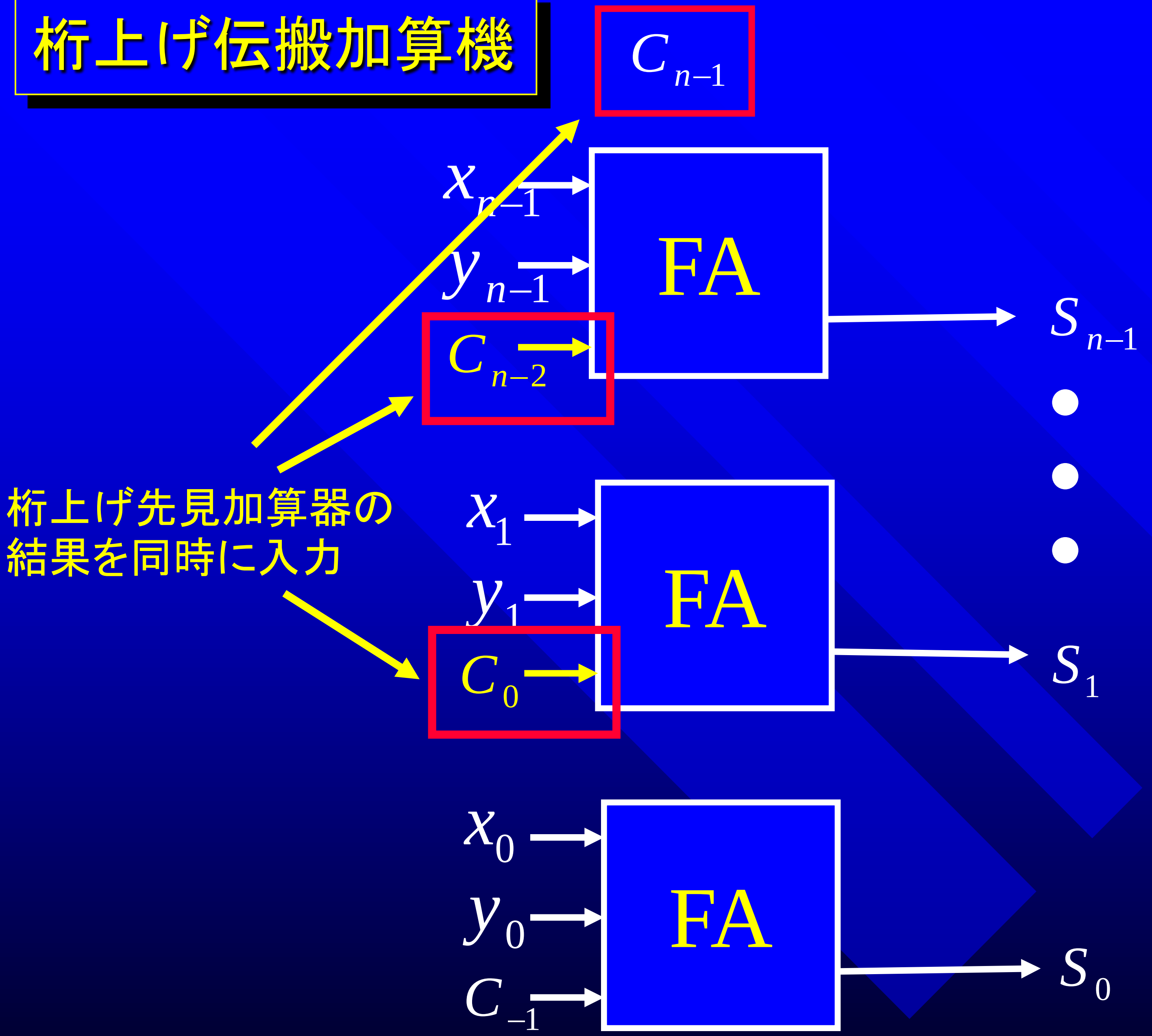
$$C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{-1}$$

$$C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{-1}$$

$$\vdots$$

$$C_k = G_k + P_k C_{k-1}$$

桁上げ伝搬加算機

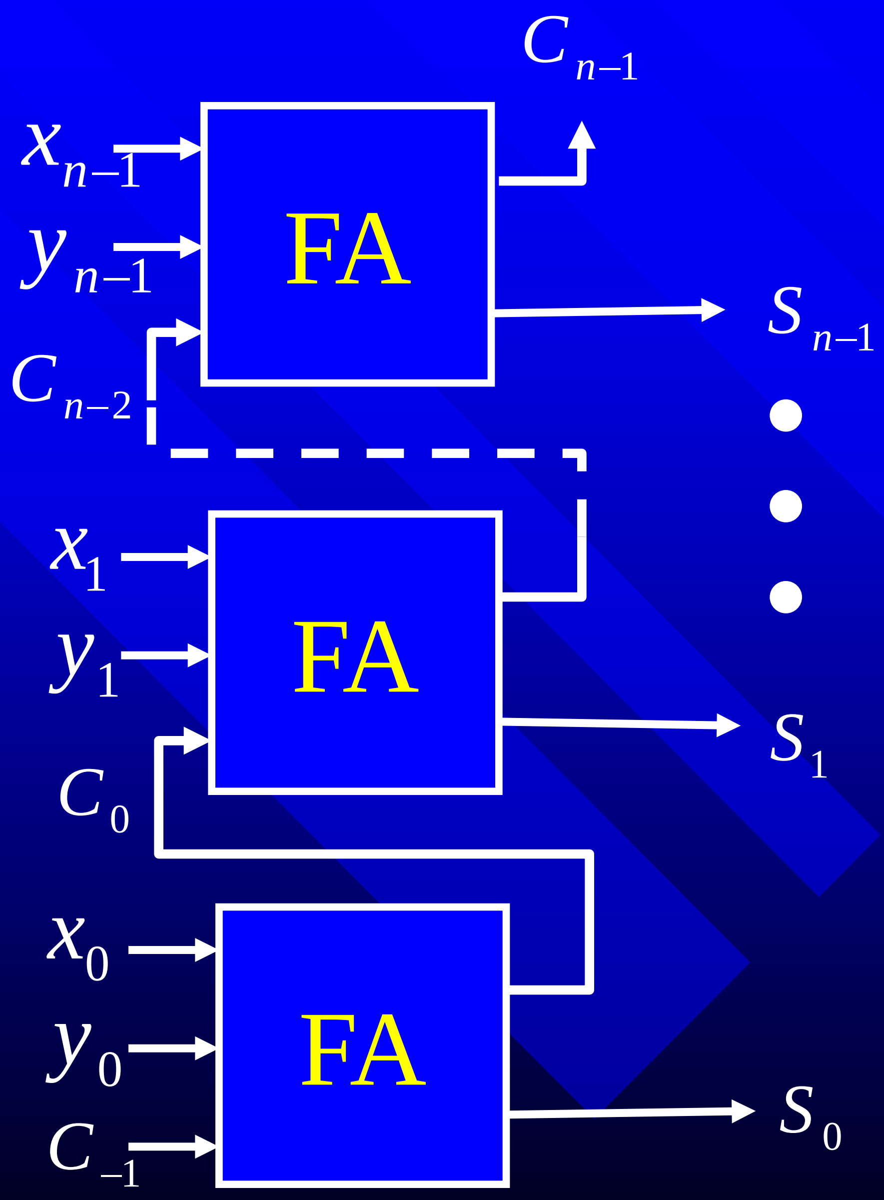


C_{n-1}
x_{n-1}
y_{n-1}
C_{n-2}
FA
S_{n-1}
桁上げ先見加算器の
結果を同時に入力
x_1
y_1
C_0
FA
S_1
x_0
y_0
C_{-1}
FA
S_0

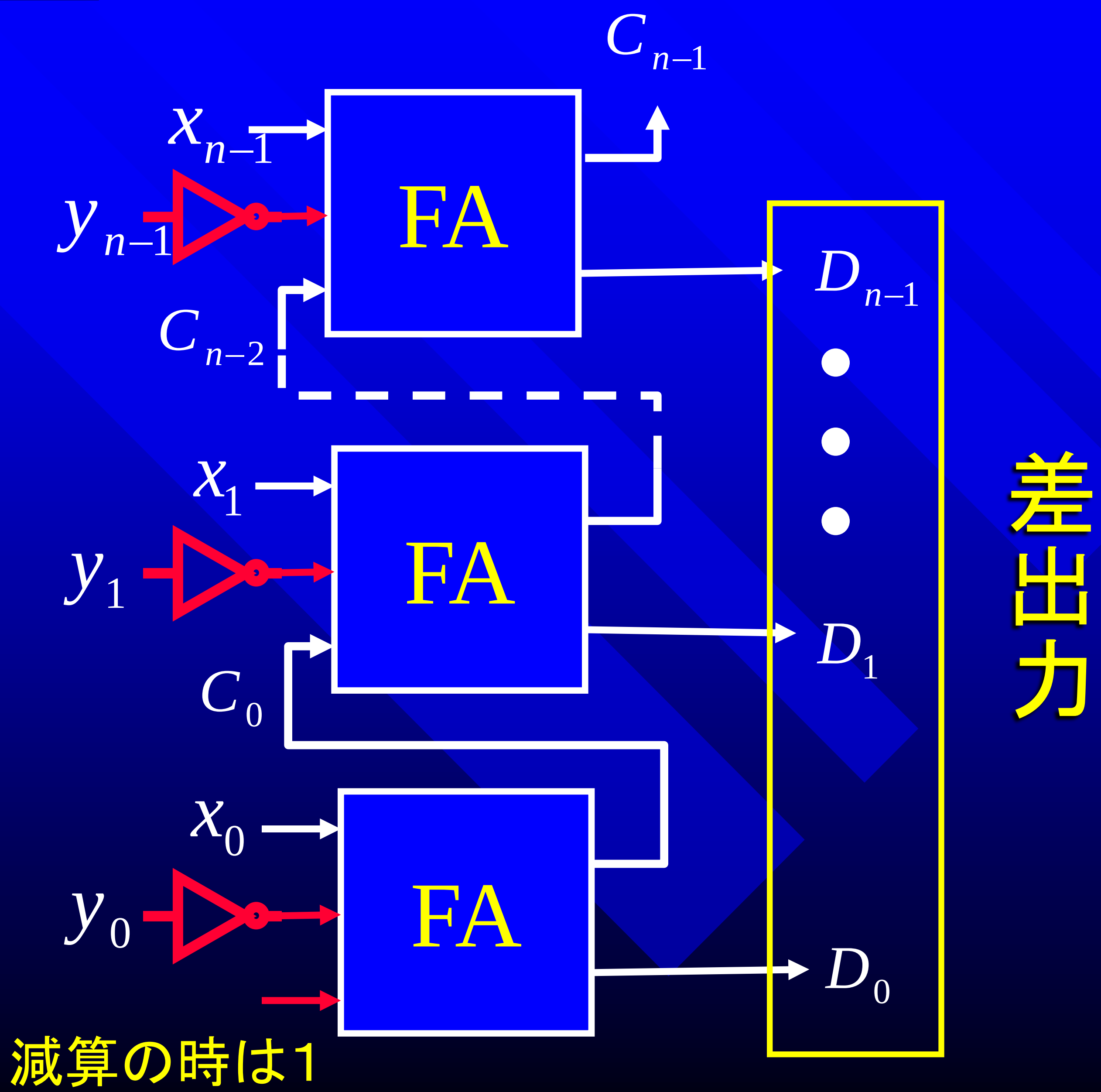
加算器を用いた減算

補数を用いた加算器による減算はどうなるのか？

補数器



補数器



オーバーフロー

コンピュータのハードウェア（容量）は有限なので、オーバーフロー（桁あふれ）が起こりえる

$-(19)_{10}-(13)_{10}=-(32)_{10}$ の例

$(19)_{10}=(0\ 1\ 0\ 0\ 1\ 1)_2$

$(13)_{10}=(0\ 0\ 1\ 1\ 0\ 1)_2$

（6ビット演算）

$$
\begin{array}{rl}
1\ 0\ 1\ 1\ 0\ 0 & \\
1\ 1\ 0\ 0\ 1\ 0 & (+ \\
\hline
1\ \ 0\ 1\ 1\ 1\ 1\ 0 & \\
1 & (+ \\
\hline
(0\ 1\ 1\ 1\ 1\ 1)_2=(\cancel{31})_{10} &
\end{array}
$$

オーバフロー
誤り

（7ビット演算）

$$
\begin{array}{rl}
1\ 1\ 0\ 1\ 1\ 0\ 0 & \\
1\ 1\ 1\ 0\ 0\ 1\ 0 & (+ \\
\hline
1\ \ 1\ 0\ 1\ 1\ 1\ 1\ 0 & \\
1 & (+ \\
\hline
(1\ 0\ 1\ 1\ 1\ 1\ 1)_2 &
\end{array}
$$

$=-(0\ 1\ 0\ 0\ 0\ 0\ 0)_2=-(32)_{10}$

正しい

基本乗算機構　かけ算の場合

$$
\begin{array}{cccccccccll}
 & & & \text{(被乗数)} & X_3 & X_2 & X_1 & X_0 & & & \\
 & & & \text{(乗\ \ 数)} & Y_3 & Y_2 & Y_1 & Y_0 & \times & & \\
\cline{4-9}
 & & & & X_3 \times Y_0 & X_2 \times Y_0 & X_1 \times Y_0 & X_0 \times Y_0 & \longrightarrow & PP_0 & \\
 & & & X_3 \times Y_1 & X_2 \times Y_1 & X_1 \times Y_1 & X_0 \times Y_1 & & \longrightarrow & PP_1 & \\
 & & X_3 \times Y_2 & X_2 \times Y_2 & X_1 \times Y_2 & X_0 \times Y_2 & & & \longrightarrow & PP_2 & \text{部分積} \\
 & X_3 \times Y_3 & X_2 \times Y_3 & X_1 \times Y_3 & X_0 \times Y_3 & & & & \longrightarrow & PP_3 & + \\
\hline
P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0 & & \text{(積)} &
\end{array}
$$

基本乗算機構　具体的な例

$(9)_{10} \times (11)_{10} = (99)_{10}$の例

$(9)_{10} = (1001)_2$　　$(11)_{10} = (1011)_2$

ⓐ 1 0 0 1

ⓑ 1 0 0 1 0

ⓒ 0 0 0 0 0 0

ⓓ 1 0 0 1 0 0 0　+　(4)

$(1\ 1\ 0\ 0\ 0\ 1\ 1)_2$

$= (99)_{10}$

1 0 0 1　　(1)

1 0 1 ①　×　(2) ①

ⓐ 1 0 0 1

1 0 0 1

1 0 ① 1　×　(2) ①

ⓑ 1 0 0 1

1 0 0 1

1 ⓪ 1 1　×　(2) ②

ⓒ 0 0 0 0

1 0 0 1

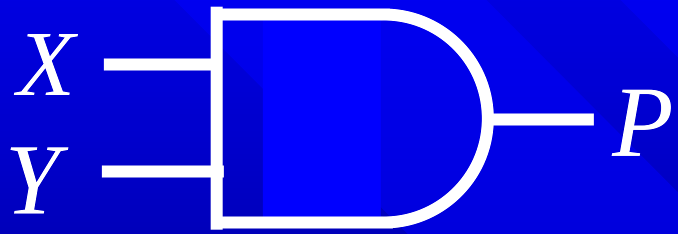
① 0 1 1　×　(2) ①

ⓓ 1 0 0 1

基本乗算機構

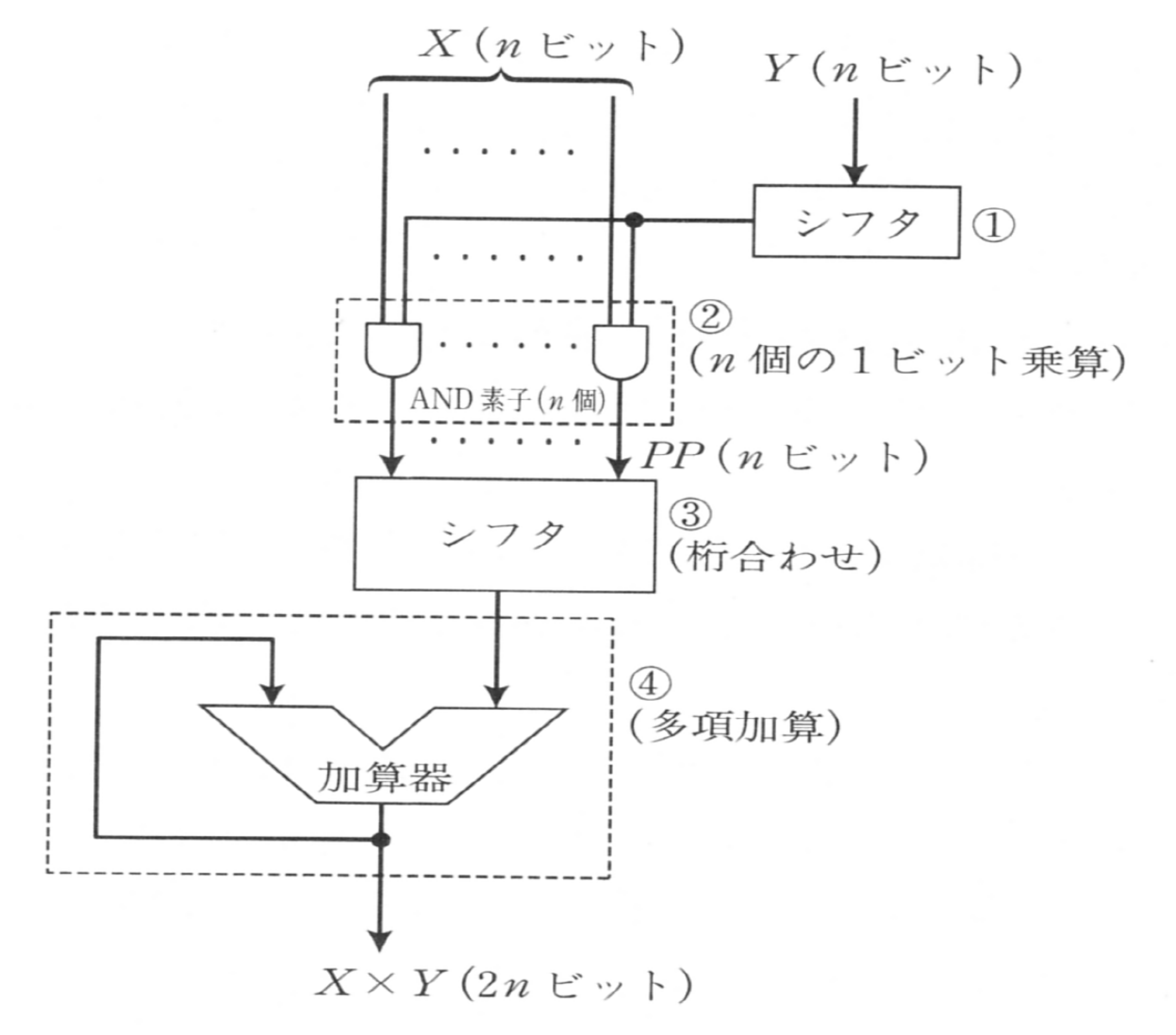
AND素子と1ビット演算

論理積(AND)



入力		出力
X	Y	P
0	0	0
0	1	0
1	0	0
1	1	1

繰り返し乗算器


X (n ビット)
Y (n ビット)
シフタ
①
②
(n 個の 1 ビット乗算)
AND 素子 (n 個)
PP (n ビット)
シフタ
③
(桁合わせ)
④
(多項加算)
加算器
X × Y (2n ビット)

ブースの方法

ブースの方法ついては自分でやること

（レポート課題）

並列乗算器

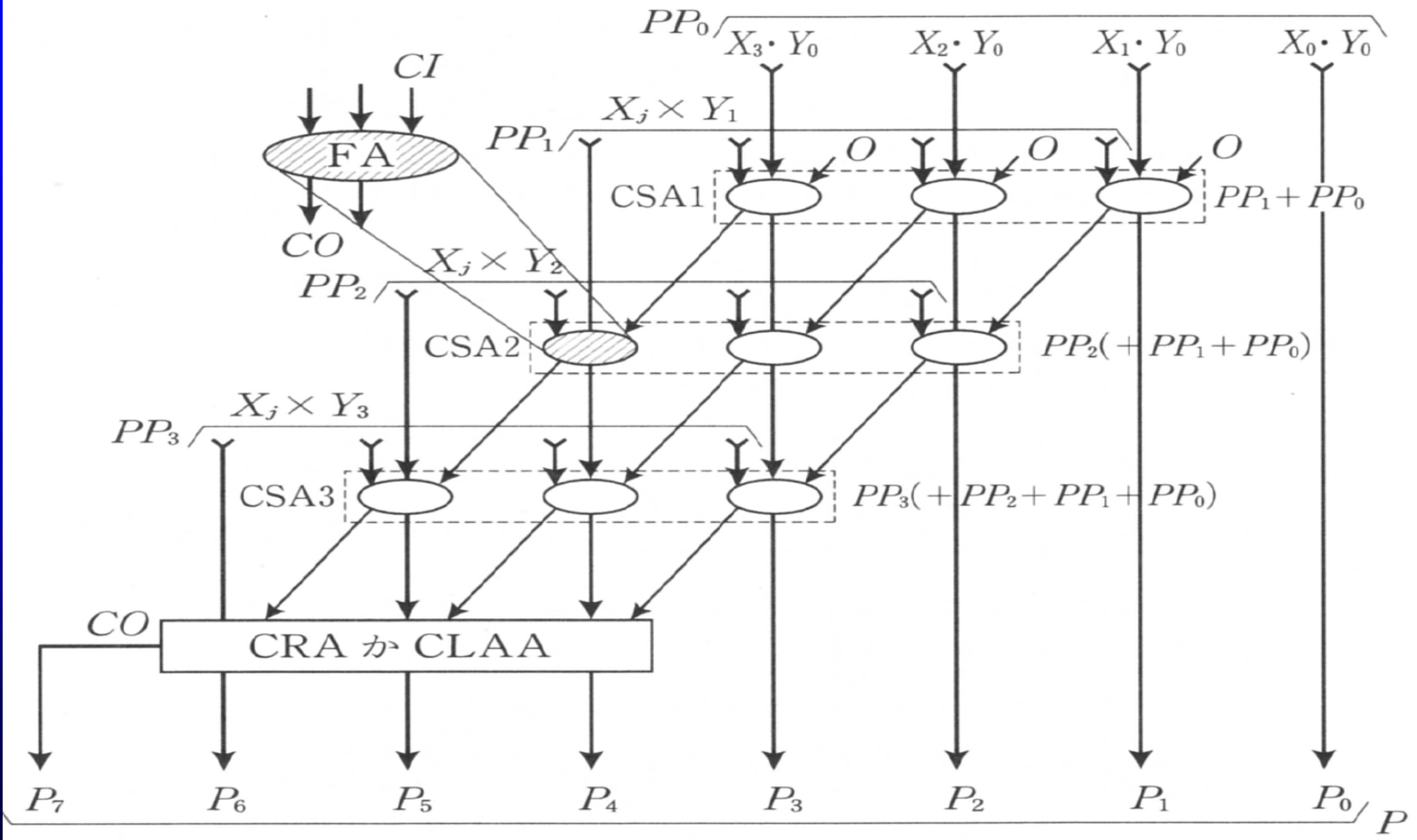
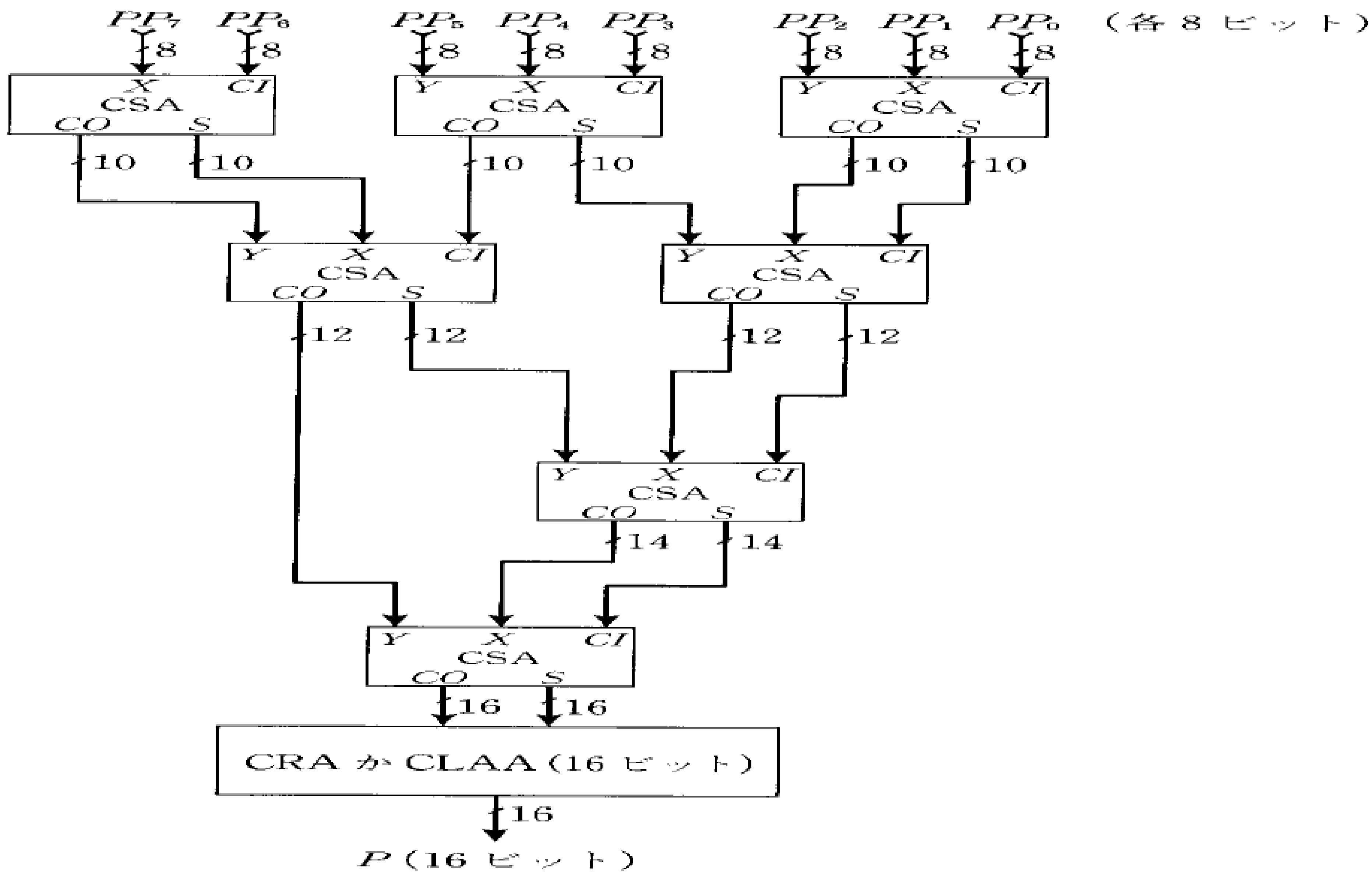


図 6. 26　4 ビット並列乗算器（8 ビット積）

ウォリスの木



PP_7 PP_6 PP_5 PP_4 PP_3 PP_2 PP_1 PP_0
（各 8 ビット）
8
X CI CSA CO S
Y X CI CSA CO S
10
12
14
16
CRA か CLAA（16 ビット）
16
P（16 ビット）

乗算幅の拡張

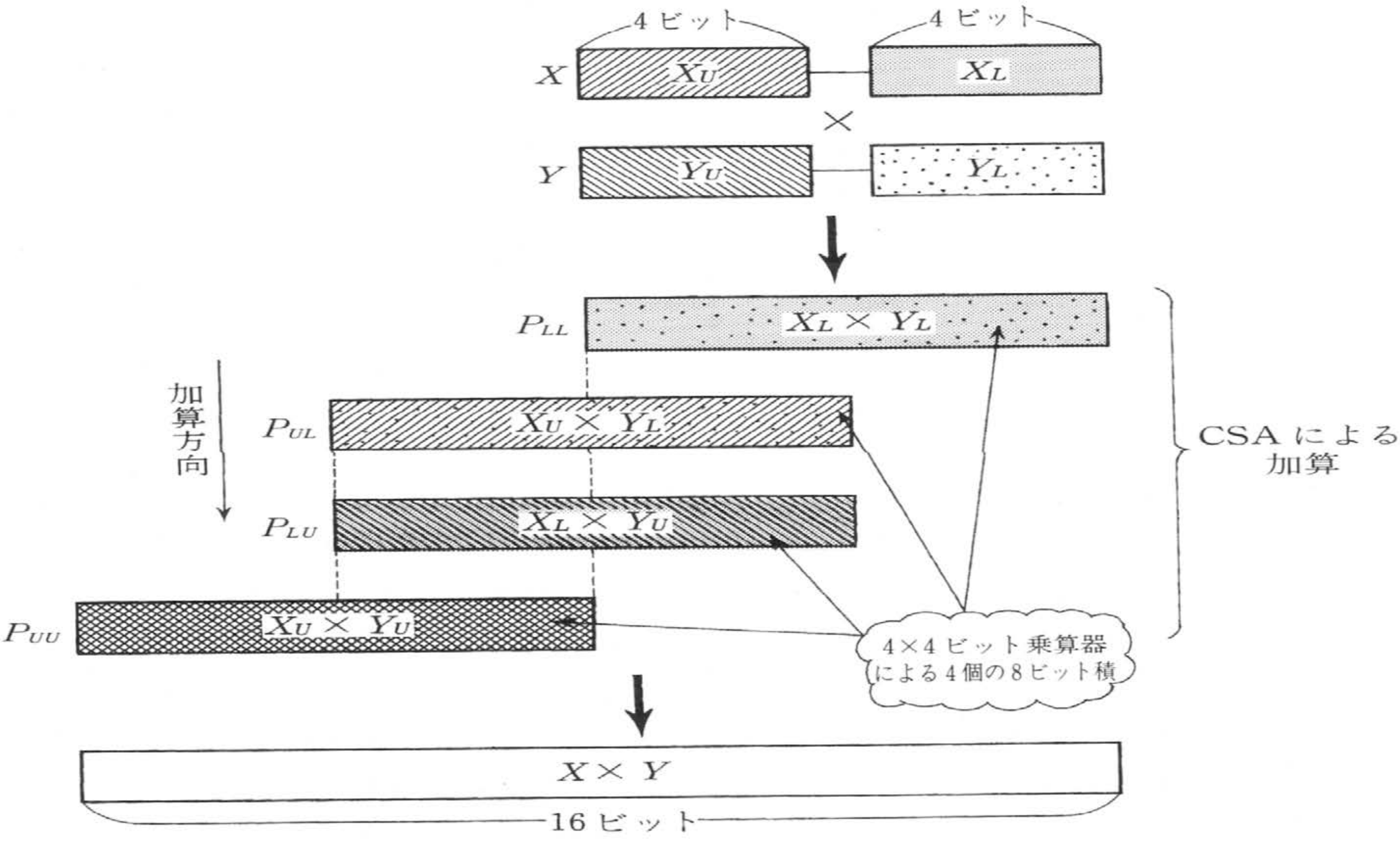


図 6.28　乗算幅の拡張例（8 ビット積 → 16 ビット積）

基本除算機構

割り算の場合 $98_{10} \div 9_{10} = 10_{10} \cdots 8_{10}$

$98_{10} = 0110\ 0010_2$　　$9_{10} = 1001_2$

１）　1001　＞　0110 (0010)　　PQ（商）

←へシフト

２）　1001　＜　1100 (010) ⟶

1001　（－

0011 (010)

３）　1001　＞　0110 (10) ⟶ 0

←へシフト

1001　＜　1101 (0)

基本除算機構

割り算の場合 $98_{10} \div 9_{10} = 10_{10} \cdots 8_{10}$

$98_{10} = 0110\ 0010_2$　　$9_{10} = 1001_2$

３）　1001　＞　0110 (10)　→　0

←へシフト

２）　1001　＜　1101 (0)　→

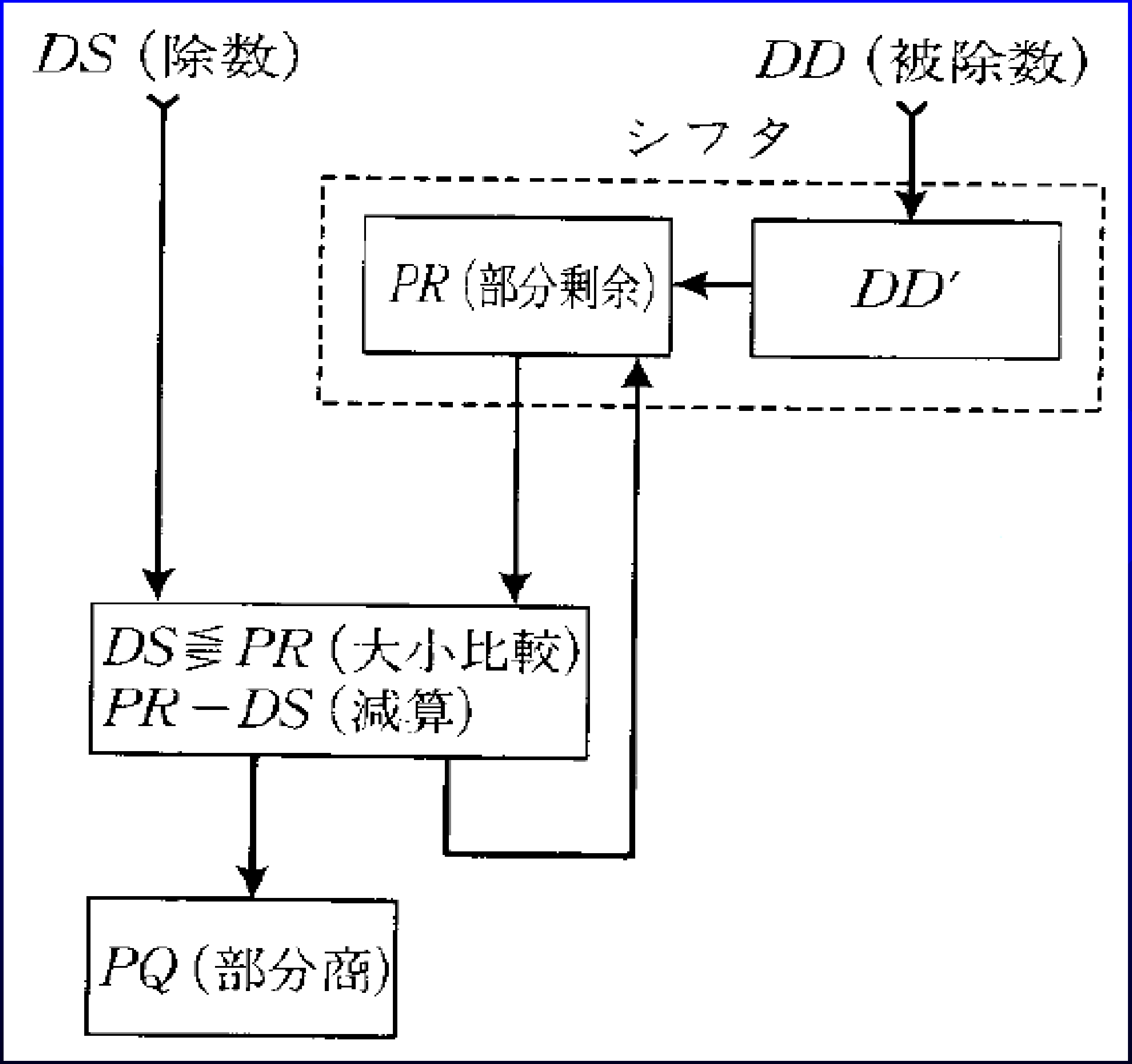
　　　　　　1001　（－

　　　　　　0100 (0)

　　1001　＞　1000　→　0

余り　　　　　　　　　商

繰り返し除算器



除算法の改良

繰り返し除算法の短所
・PRとDSの大小比較と減算を別々のタイミングで行う

理由
・比較の結果により減算を行うか否かが決定される

大小比較と減算を演算器としてだけでなく手順としても
共用する方式

⇩

引き戻し法・引き放し法

引き戻し法

$47_{10} \div 5_{10} = 9_{10} \cdots 2_{10}$ の場合

$47_{10} = 101111_2$ $5_{10} = 101_2$

PQ（商）

1） $101 = 101(111)$

101 （−

$000(111)$

2） $101 > 001(11)$ 0

101 （−

1100

101 （＋

$001(11)$

マイナスに
なったら、＋
に引き戻す

引き戻し法

$47_{10} \div 5_{10} = 9_{10} \cdots 2_{10}$ の場合

PQ（商）

$47_{10} = 101111_2$ $\quad$ $5_{10} = 101_2$

0

101 $\quad$ 001 (11)

101 $>$ 011 (1) $\quad$ 0

$\quad$ 101 （－

1110

$\quad$ 101 （＋

011 (1)

111

引き戻し法

$47_{10} \div 5_{10} = 9_{10} \cdots 2_{10}$ の場合

PQ（商）

$47_{10} = 101111_2$ $5_{10} = 101_2$

0
0

$$
\begin{array}{lcr}
101 & < & 111 \\
 & & 101 \quad (- \\
\hline
 & & 010
\end{array}
$$

引き戻し法

$47_{10} \div 5_{10} = 9_{10} \cdots 2_{10}$ の場合

PQ（商）

$47_{10} = 101111_2$　　$5_{10} = 101_2$

0
0

$101 < 111 \longrightarrow$

101　（−

010

余り

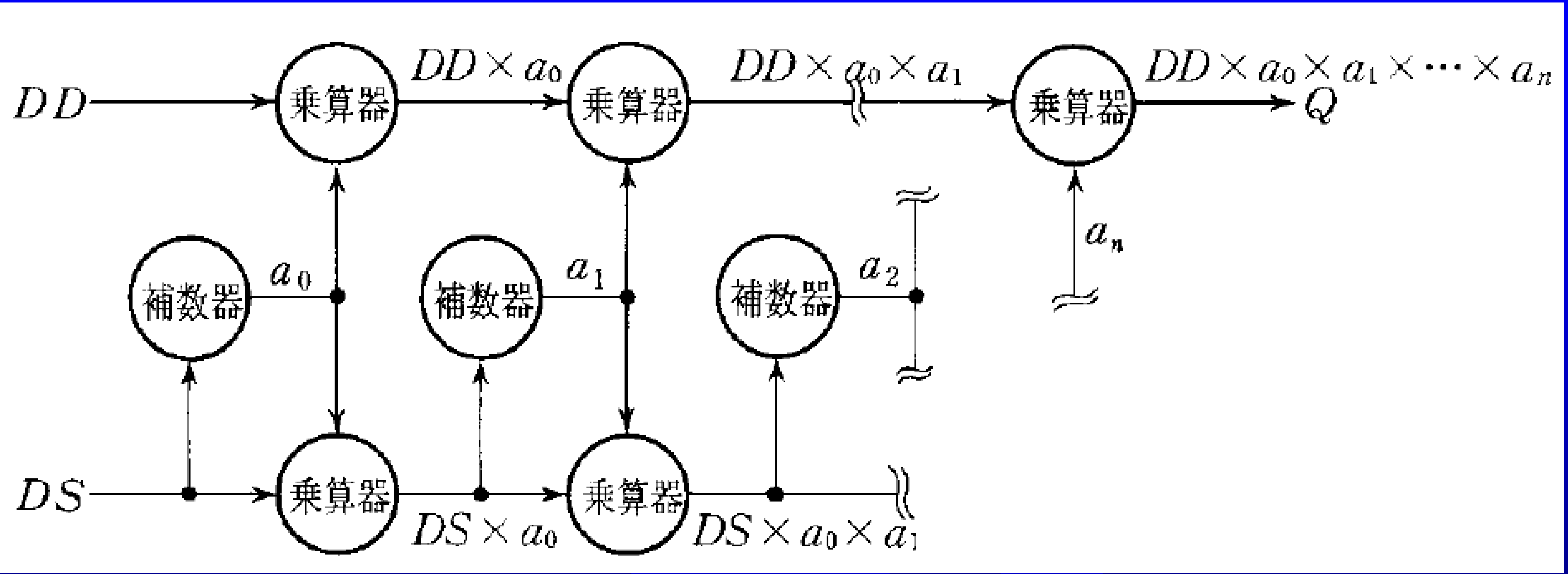
商

引き放し法

引き放し法については自分でやること

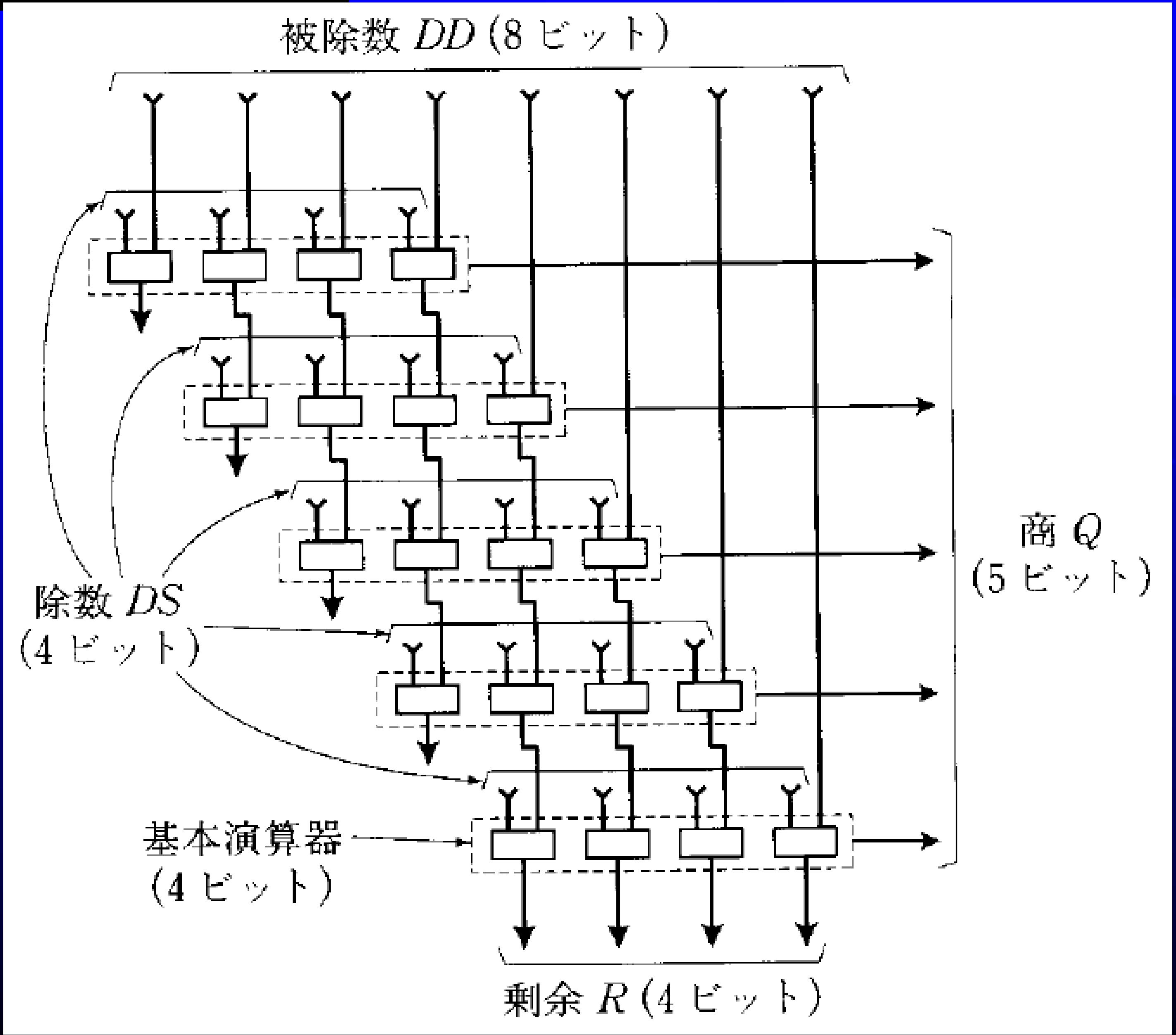
（レポート課題）

乗算収束型除算機構



除算より乗算の法が数倍早いことを利用して、除算を乗算によって計算する方法

配列型除算器



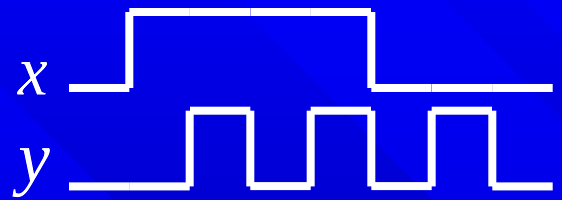
本日のまとめ

演算アーキテクチャー
-固定小数点の算術演算装置-

1. 半加算回路と全加算回路
2. 加算機
3. 乗算機構
4. 除算機構

本日の課題

１．図の半加算機において、以下の入力パルス列に対する和出力と桁上げ出力を求めよ



２．ブースの方法で　$9_{10} \times (-13_{10})$ を計算しなさい。計算の手順も教科書（図6.25）の例にならって示すこと。

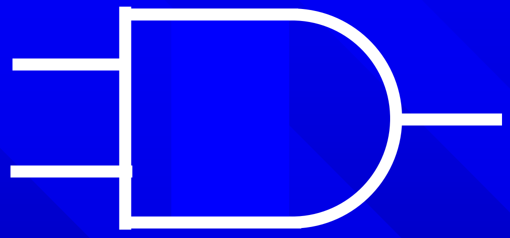
３．　$88_{10} \div (6_{10})$　の計算を

①繰り返し除算法　②引き戻し法　③引き放し法

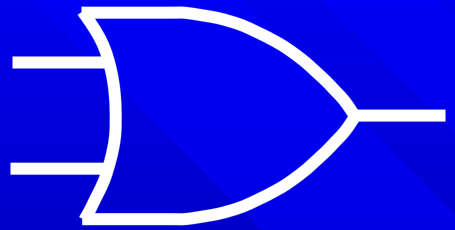
で、計算しなさい。
計算の手順も教科書の例にならって示すこと。

補足

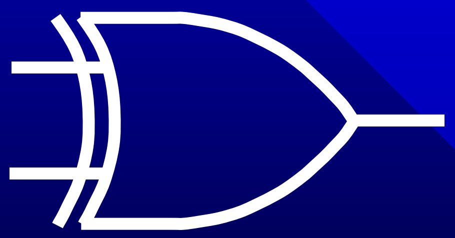
論理積(AND)



論理和(OR)



排他的論理和(XOR)



[Exclusive OR]

否定(NOT)