

演算機能回路

第4週

2006/10/23

泉知論

回路性能

performance

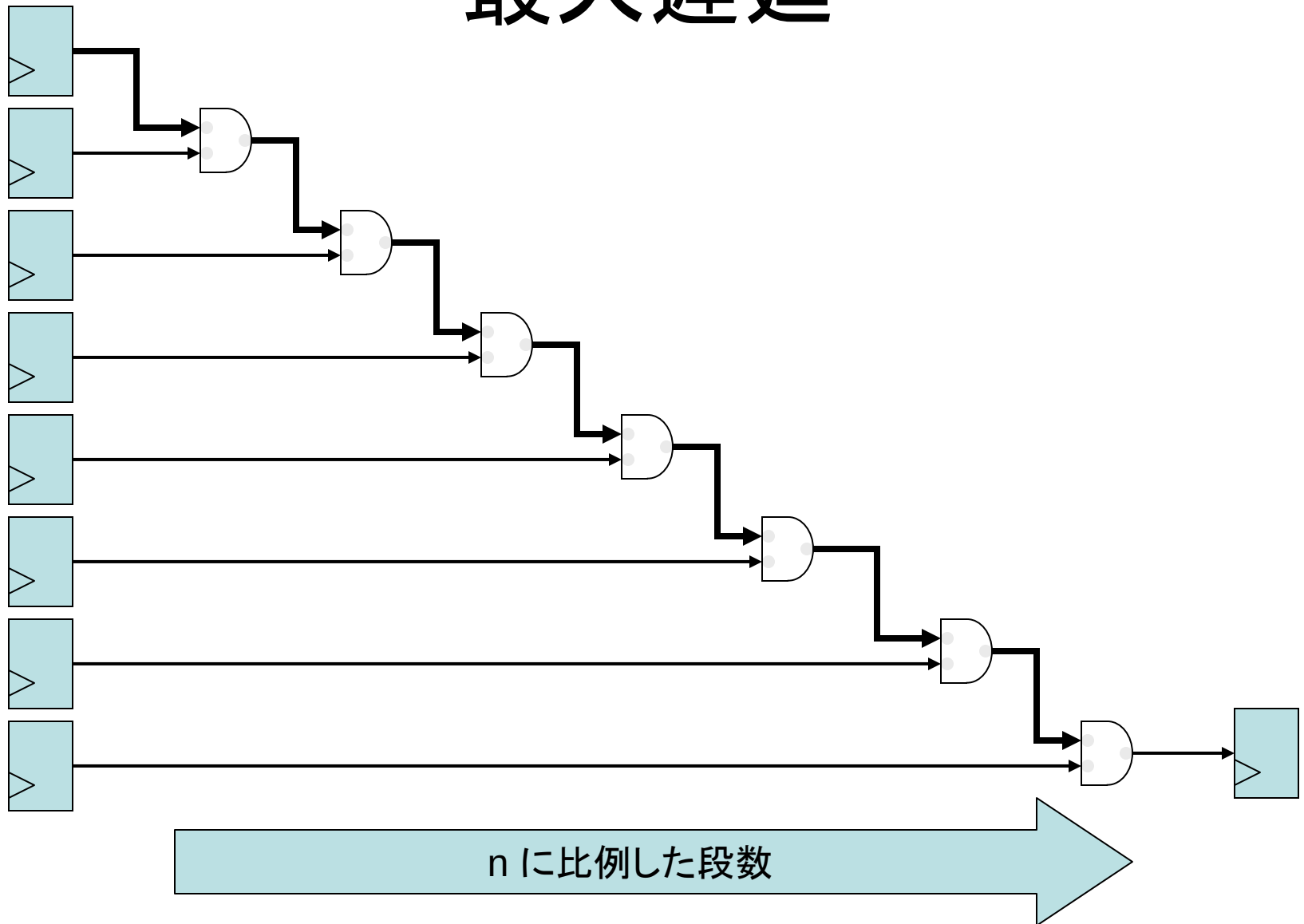
回路の性能

- 回路規模
 - 小さくしたい
- 速度
 - 速くしたい
- トレードオフ
 - tradeoff = 同時に満たしえない幾つかの条件の取捨についての考量; 交換条件, 見返り (英和辞典より)

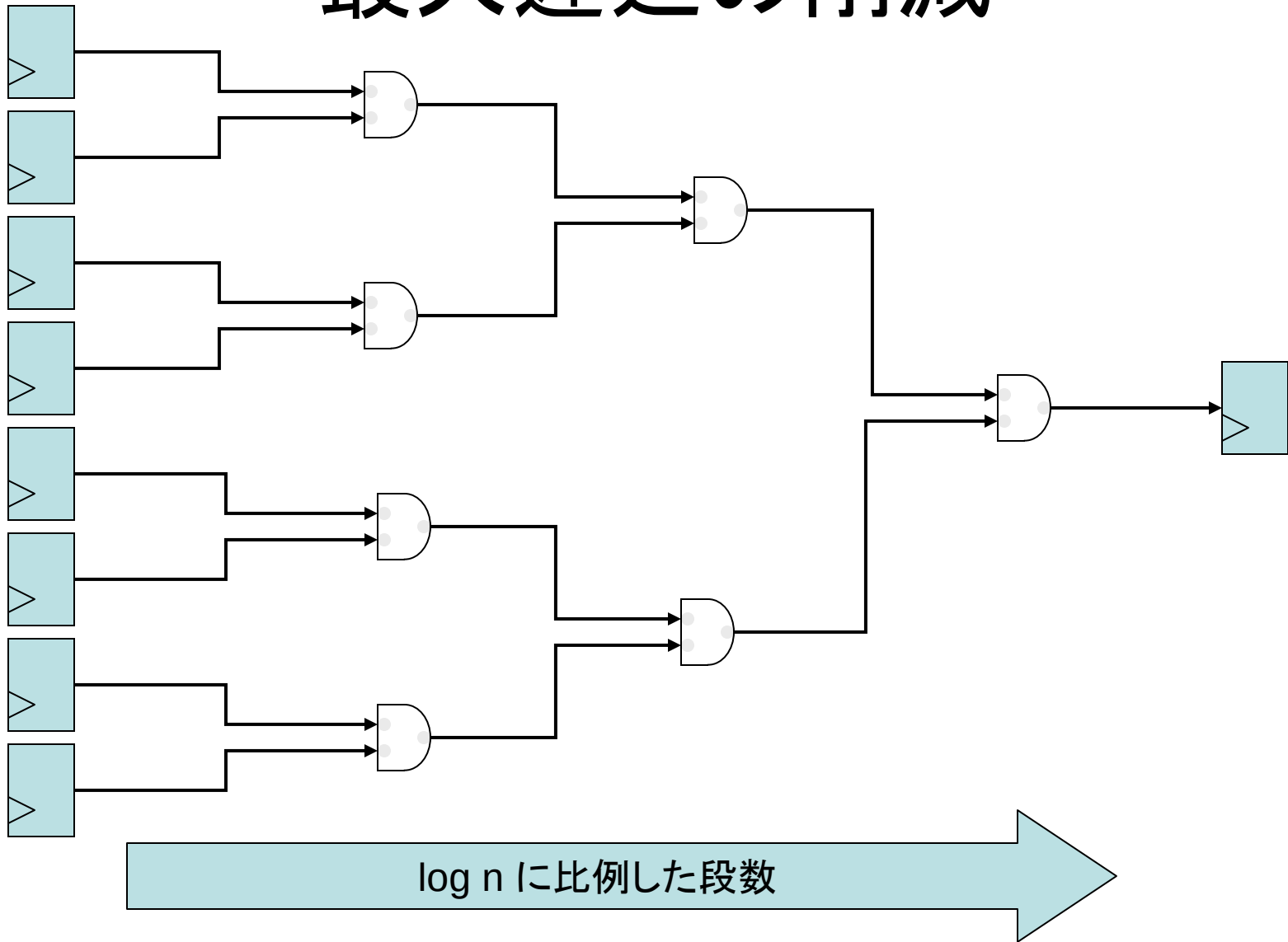
動作周波数

- 回路の性能
 - 高周波数動作→高速
- 動作周波数を決めるもの
 - 記憶素子を出て記憶素子に入る信号の最大遅延
 - 最大遅延の削減→回路の高速化
- 高速化設計の原則
 - 遅延の削減←ゲートの段数 (原則)

最大遅延



最大遅延の削減



桁上先見加算器

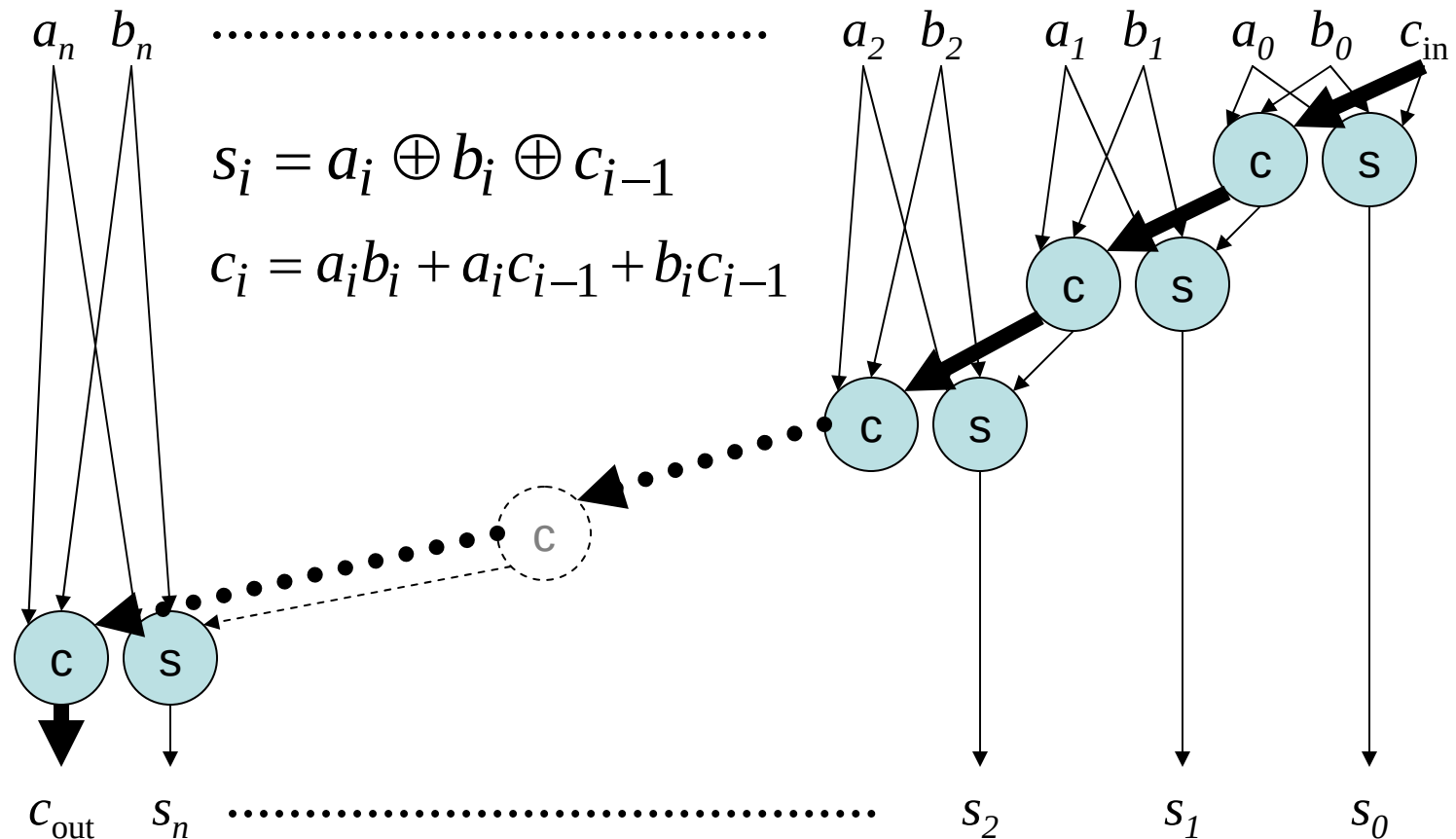
Carry Look-ahead Adder
(CLA)

桁上先見加算器とは

- 高速な加算器
- 演算器の tune-up の古典、基礎、典型
- 実際の回路設計でも多用される有用な回路

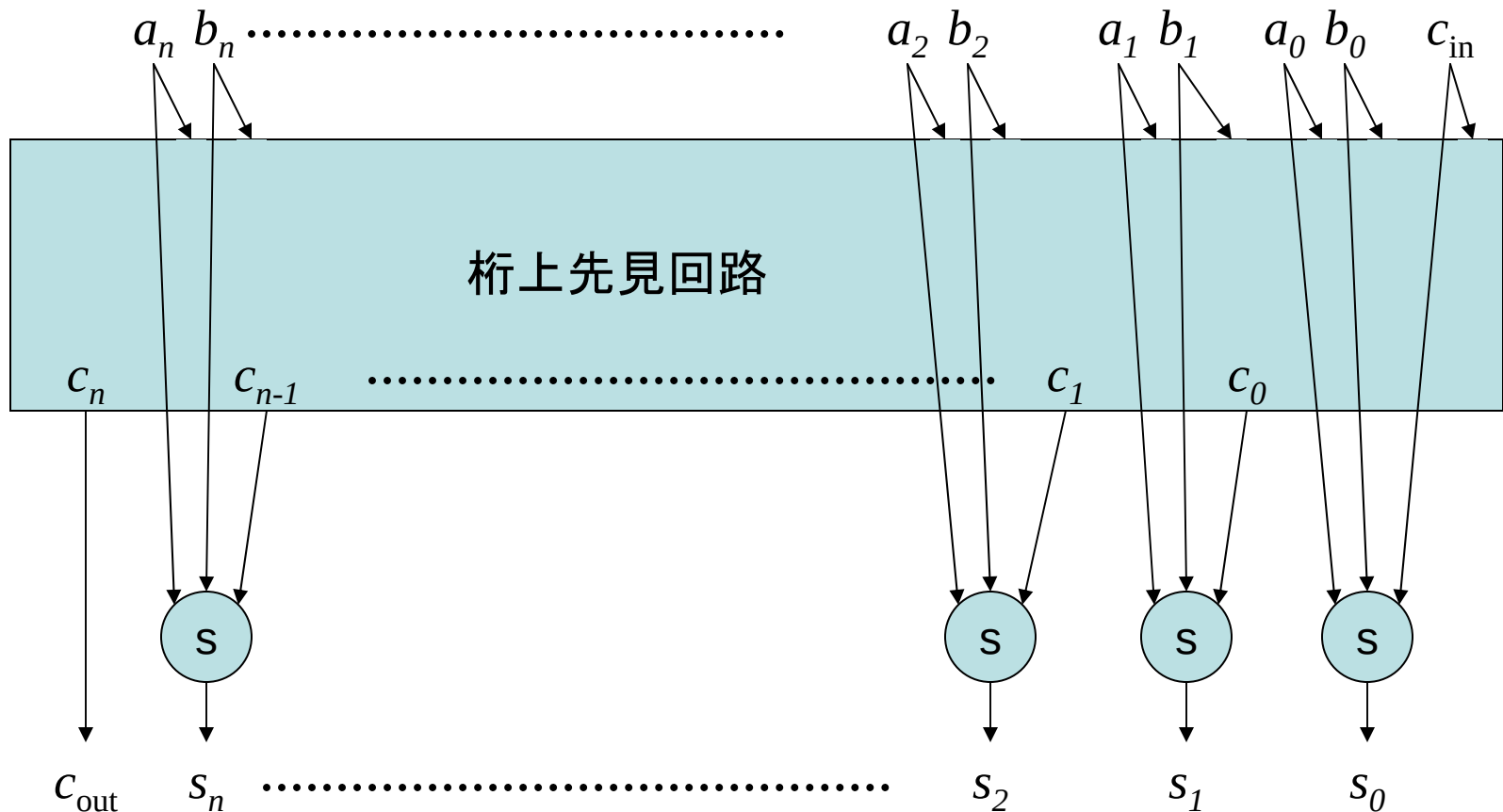
順次桁上加算器の最大遅延

- 桁上げの遅延がビット数に比例して増大



桁上先見加算器

- 桁上信号を先に速く(少ない段数で)計算

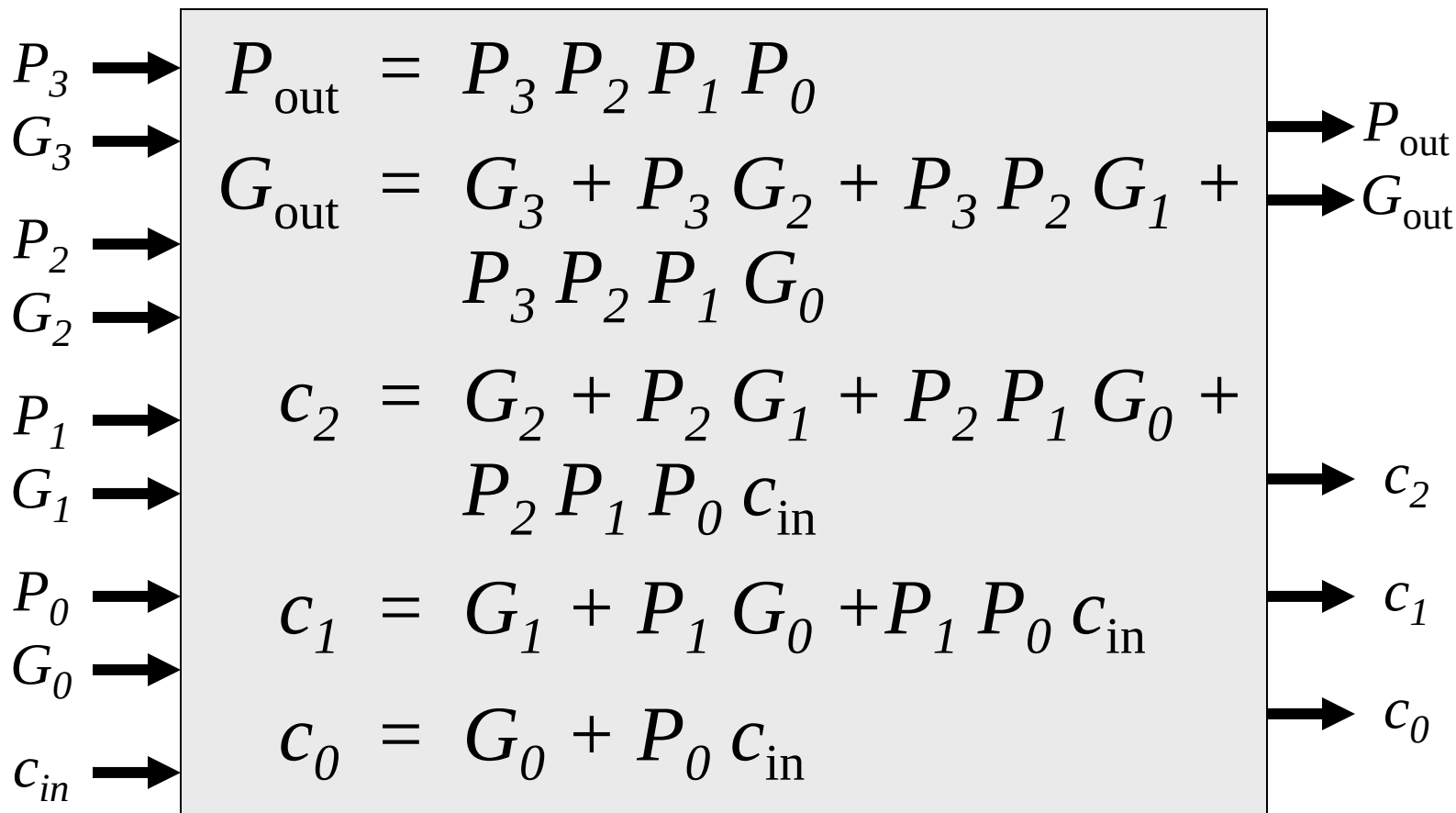


桁上先見回路

- 入力: $a_{n-1}\dots a_1 a_0, b_{n-1}\dots b_1 b_0, c_{\text{in}}$
 - 出力: $c_{\text{out}} c_{n-1}\dots c_1 c_0$
 - 再帰的な木状構成が可能
 - l 段の k 入力の桁上先見回路の木
 - $\rightarrow k^l$ 入力の桁上先見回路
 - 論理代数的解説→別資料
 - $P_{i:j}, G_{i:j}$ の計算 $\rightarrow \log n$ に比例する段数
 - c_i の計算 $\rightarrow \log n$ に比例する段数
 - s_i の計算 $\rightarrow$ 定数段数
- よって、 $\log n$ に比例する段数で加算できる

桁上先見モジュール

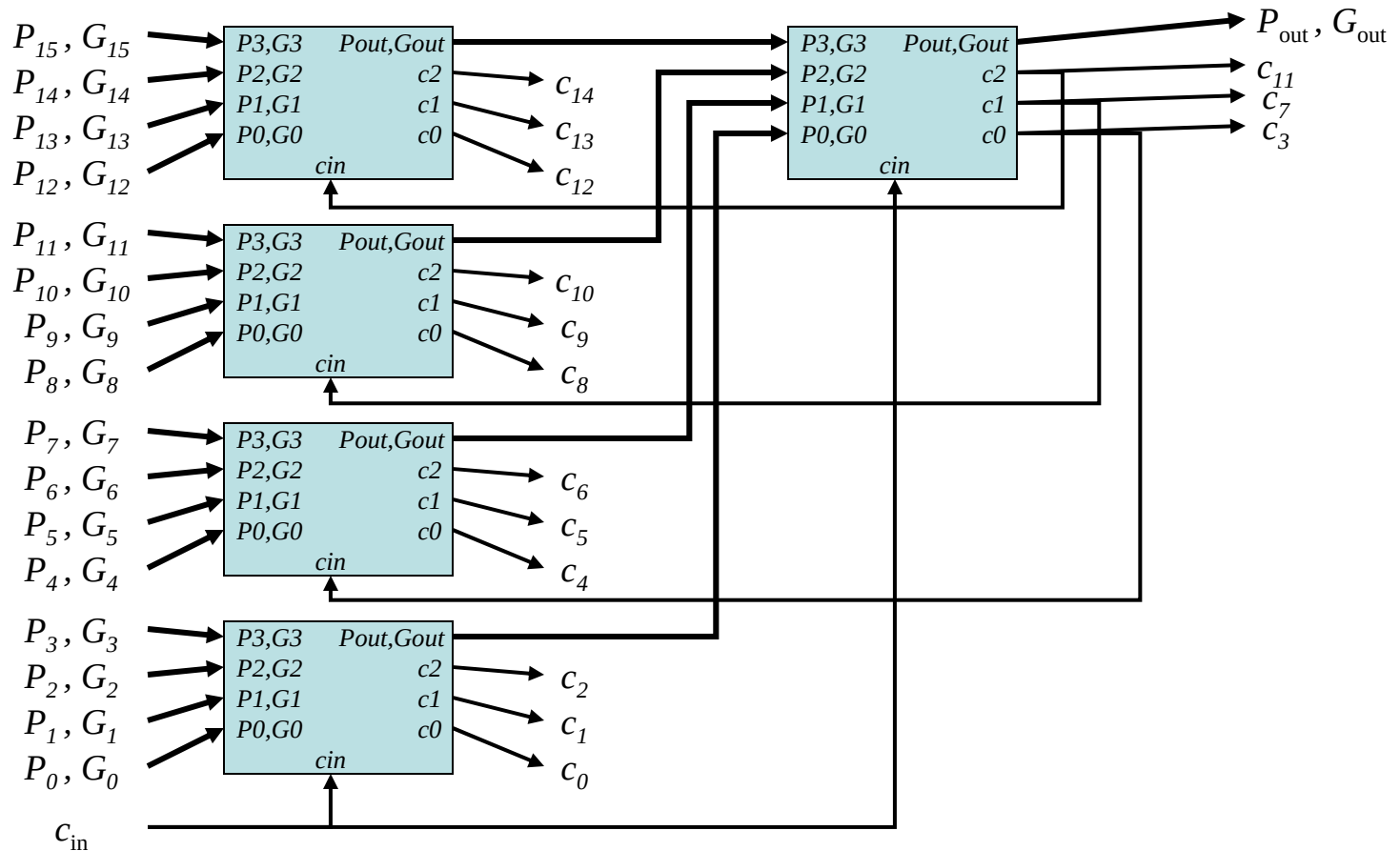
- 4入力桁上先見計算モジュールの例



※実際のモジュールでは負論理の入出力となる場合もある。

桁上先見モジュールの再帰的構成

例: 4bit 桁上先見モジュール2段→16bit 桁上先見モジュール

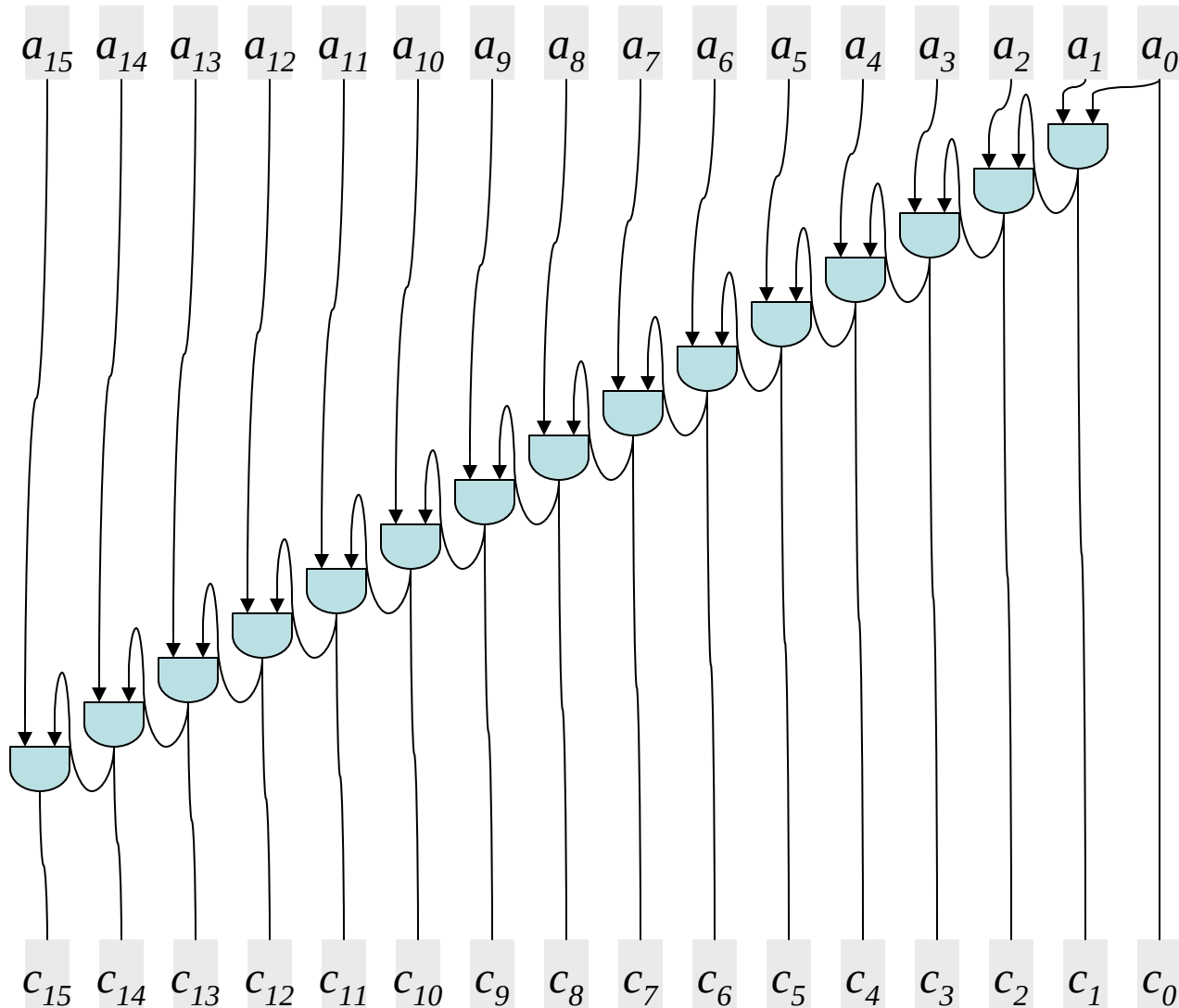


例題：インクリメンタの桁上先見

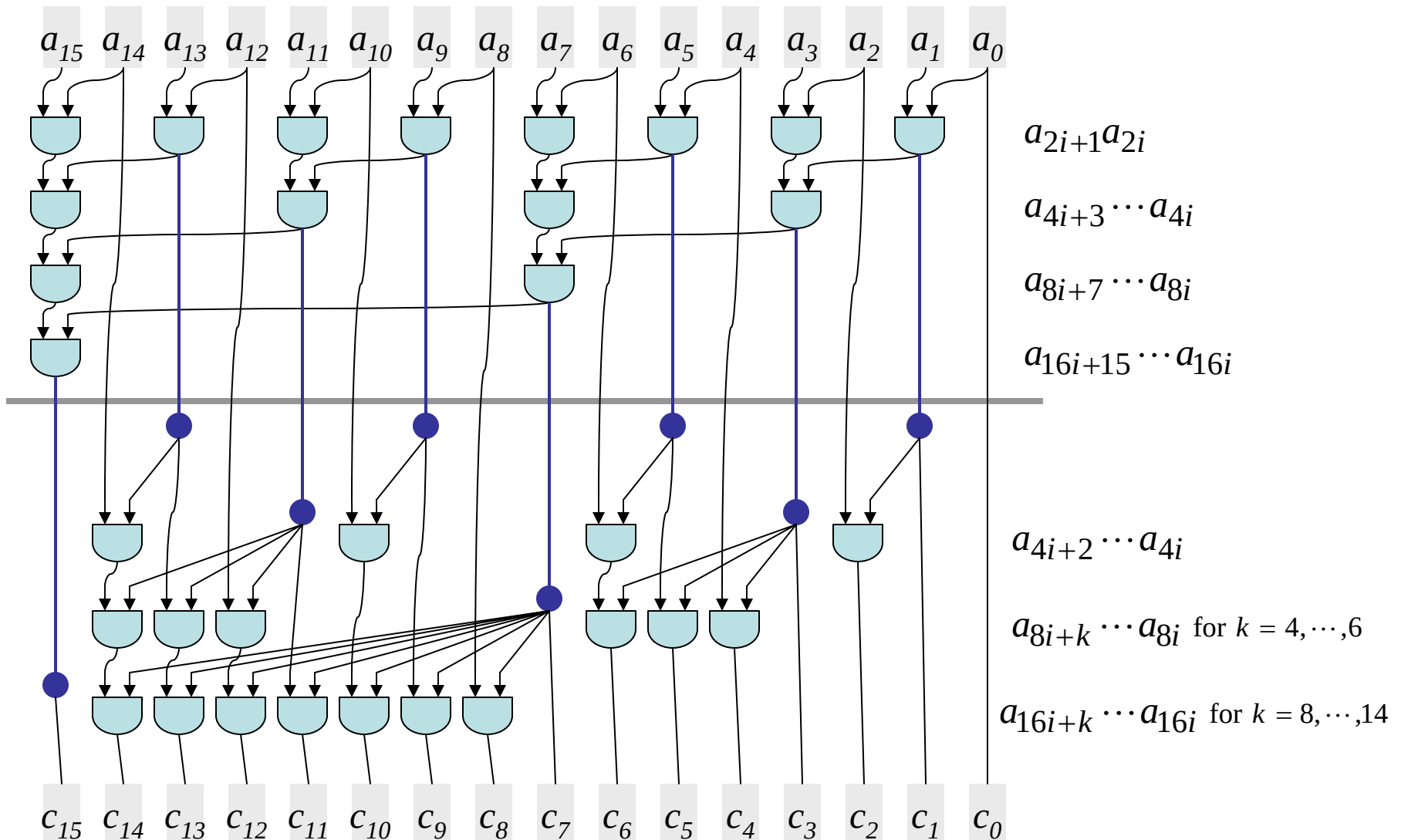
- インクリメンタ、デクリメンタ
- $B=000\dots001$ あるいは $B=111\dots111$
- 桁上先見がより簡単になる。

$$c_i = \prod_{k=0}^{i-1} a_k$$

インクリメンタの桁上(先見なし)



インクリメンタの桁上先見



1 adder

Consider a m bit + n bit adder. Hereinafter, assume $m = n$. Here is an example of a 4bit adder:

$$\begin{array}{rcccccc}
 & & & & c_{\text{in}} & \\
 & & a_3 & a_2 & a_1 & a_0 \\
 +) & & b_3 & b_2 & b_1 & b_0 \\
 \hline
 c_{\text{out}} & s_3 & s_2 & s_1 & s_0 &
 \end{array}$$

1.1 ripple-carry adder

(skip)

1.2 carry-lookahead adder

Let

$$G_i = a_i b_i$$

and

$$P_i = a_i \oplus b_i$$

and c_{-1} be c_{in} , then we have

$$s_i = P_i \oplus c_{i-1}$$

and

$$\begin{aligned}
 c_i &= G_i + P_i c_{i-1} \\
 &= G_i + \\
 &\quad P_i G_{i-1} + \\
 &\quad P_i P_{i-1} G_{i-2} + \\
 &\quad P_i P_{i-1} P_{i-2} G_{i-3} + \\
 &\quad \cdots + \\
 &\quad P_i P_{i-1} P_{i-2} \cdots P_1 G_0 + \\
 &\quad P_i P_{i-1} P_{i-2} \cdots P_1 P_0 c_{-1} \\
 &= G_i + \sum_{k=0}^{i-1} \prod_{\ell=k+1}^i P_\ell G_k + \prod_{\ell=0}^i P_\ell c_{-1}.
 \end{aligned}$$

1.2.1 recursive division of carry-lookahead

For $0 \leq j \leq i \leq n-1$, let

$$\begin{aligned}
 P_{\langle i:j \rangle} &= P_i P_{i-1} \cdots P_j \\
 &= \prod_{\ell=j}^i P_\ell
 \end{aligned}$$

and

$$\begin{aligned}
G_{\langle i:j \rangle} &= G_i + \\
&\quad P_i G_{i-1} + \\
&\quad P_i P_{i-1} G_{i-2} + \\
&\quad \cdots + \\
&\quad P_i P_{i-1} \cdots P_{j+1} G_j \\
&= G_i + \sum_{k=j}^{i-1} \prod_{\ell=k+1}^i P_\ell G_j.
\end{aligned}$$

Consider to divide $P_{\langle i:j \rangle}$ and $G_{\langle i:j \rangle}$ into λ subparts where the positions to divide are $j-1 = x(-1) < x(0) < \cdots < x(\lambda-1) = i$. Fig. 1 shows an example of positions to a division where $i = 31$, $j = 16$, $\lambda = 4$, $x(3) = 31$, $x(2) = 27$, $x(1) = 23$, $x(0) = 19$, and $x(-1) = 15$.

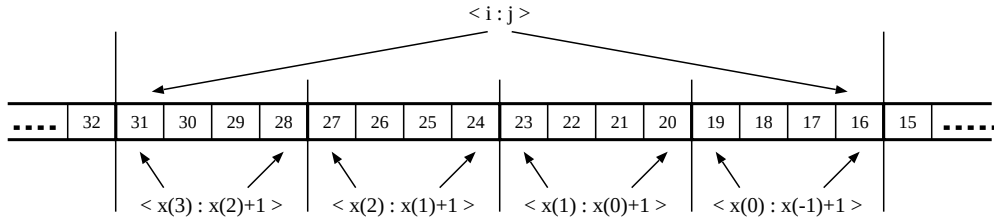


Figure 1: An example of positions to a division

We have

$$\begin{aligned}
P_{\langle i:j \rangle} &= P_{\langle x(\lambda-1):x(\lambda-2)+1 \rangle} P_{\langle x(\lambda-2):x(\lambda-3)+1 \rangle} \cdots P_{\langle x(0):x(-1)+1 \rangle} \\
&= \prod_{\ell=0}^{\lambda-1} P_{\langle x(\ell):x(\ell-1)+1 \rangle}
\end{aligned}$$

and

$$\begin{aligned}
G_{\langle i:j \rangle} &= G_{\langle x(\lambda-1):x(\lambda-2)+1 \rangle} + \\
&\quad P_{\langle x(\lambda-1):x(\lambda-2)+1 \rangle} G_{\langle x(\lambda-2):x(\lambda-3)+1 \rangle} + \\
&\quad P_{\langle x(\lambda-1):x(\lambda-2)+1 \rangle} P_{\langle x(\lambda-2):x(\lambda-3)+1 \rangle} G_{\langle x(\lambda-3):x(\lambda-4)+1 \rangle} + \\
&\quad \cdots + \\
&\quad P_{\langle x(\lambda-1):x(\lambda-2)+1 \rangle} P_{\langle x(\lambda-2):x(\lambda-3)+1 \rangle} P_{\langle x(\lambda-3):x(\lambda-4)+1 \rangle} \cdots P_{\langle x(1):x(0)+1 \rangle} G_{\langle x(0):x(-1)+1 \rangle} \\
&\quad G_{\langle x(\lambda-1):x(\lambda-2)+1 \rangle} + \sum_{k=0}^{\lambda-2} \prod_{\ell=k+1}^{\lambda-1} P_{\langle x(\ell):x(\ell-1)+1 \rangle} G_{\langle x(k):x(k-1)+1 \rangle}.
\end{aligned}$$

Finally, the $x(\mu)$ -th carry for $0 \leq \mu < \lambda - 1$ is obtained by

$$c_{x(\mu)} = G_{\langle x(\mu):x(\mu-1)+1 \rangle} + \sum_{k=0}^{\mu-1} \prod_{\ell=k+1}^{\mu} P_{\langle x(\ell):x(\ell-1)+1 \rangle} G_{\langle x(k):x(k-1)+1 \rangle} + \prod_{\ell=0}^{\mu} P_{\langle x(\ell):x(\ell-1)+1 \rangle} c_{x(-1)}.$$

1.2.2 example

The following is an example of the CLA computation for a 16bit adder dividing P and G into four-parts, recursively.

$$\begin{aligned}
P_{\langle 3:0 \rangle} &= P_{\langle 3:3 \rangle} P_{\langle 2:2 \rangle} P_{\langle 1:1 \rangle} P_{\langle 0:0 \rangle} \\
G_{\langle 3:0 \rangle} &= G_{\langle 3:3 \rangle} + P_{\langle 3:3 \rangle} G_{\langle 2:2 \rangle} + P_{\langle 3:3 \rangle} P_{\langle 2:2 \rangle} G_{\langle 1:1 \rangle} + P_{\langle 3:3 \rangle} P_{\langle 2:2 \rangle} P_{\langle 1:1 \rangle} G_{\langle 0:0 \rangle} \\
P_{\langle 7:4 \rangle} &= P_{\langle 7:7 \rangle} P_{\langle 6:6 \rangle} P_{\langle 5:5 \rangle} P_{\langle 4:4 \rangle} \\
G_{\langle 7:4 \rangle} &= G_{\langle 7:7 \rangle} + P_{\langle 7:7 \rangle} G_{\langle 6:6 \rangle} + P_{\langle 7:7 \rangle} P_{\langle 6:6 \rangle} G_{\langle 5:5 \rangle} + P_{\langle 7:7 \rangle} P_{\langle 6:6 \rangle} P_{\langle 5:5 \rangle} G_{\langle 4:4 \rangle} \\
P_{\langle 11:8 \rangle} &= P_{\langle 11:11 \rangle} P_{\langle 10:10 \rangle} P_{\langle 9:9 \rangle} P_{\langle 8:8 \rangle} \\
G_{\langle 11:8 \rangle} &= G_{\langle 11:11 \rangle} + P_{\langle 11:11 \rangle} G_{\langle 10:10 \rangle} + P_{\langle 11:11 \rangle} P_{\langle 10:10 \rangle} G_{\langle 9:9 \rangle} + P_{\langle 11:11 \rangle} P_{\langle 10:10 \rangle} P_{\langle 9:9 \rangle} G_{\langle 8:8 \rangle} \\
P_{\langle 15:12 \rangle} &= P_{\langle 15:15 \rangle} P_{\langle 14:14 \rangle} P_{\langle 13:13 \rangle} P_{\langle 12:12 \rangle} \\
G_{\langle 15:12 \rangle} &= G_{\langle 15:15 \rangle} + P_{\langle 15:15 \rangle} G_{\langle 14:14 \rangle} + P_{\langle 15:15 \rangle} P_{\langle 14:14 \rangle} G_{\langle 13:13 \rangle} + P_{\langle 15:15 \rangle} P_{\langle 14:14 \rangle} P_{\langle 13:13 \rangle} G_{\langle 12:12 \rangle} \\
P_{\langle 15:0 \rangle} &= P_{\langle 15:12 \rangle} P_{\langle 11:8 \rangle} P_{\langle 7:4 \rangle} P_{\langle 3:0 \rangle} \\
G_{\langle 15:0 \rangle} &= G_{\langle 15:12 \rangle} + P_{\langle 15:12 \rangle} G_{\langle 11:8 \rangle} + P_{\langle 15:12 \rangle} P_{\langle 11:8 \rangle} G_{\langle 7:4 \rangle} + P_{\langle 15:12 \rangle} P_{\langle 11:8 \rangle} P_{\langle 7:4 \rangle} G_{\langle 3:0 \rangle} \\
c_{15} &= G_{\langle 15:0 \rangle} + P_{\langle 15:0 \rangle} c_{-1} \\
c_3 &= G_{\langle 3:0 \rangle} + P_{\langle 3:0 \rangle} c_{-1} \\
c_7 &= G_{\langle 7:4 \rangle} + P_{\langle 7:4 \rangle} G_{\langle 3:0 \rangle} + P_{\langle 7:4 \rangle} P_{\langle 3:0 \rangle} c_{-1} \\
c_{11} &= G_{\langle 11:8 \rangle} + P_{\langle 11:8 \rangle} G_{\langle 7:4 \rangle} + P_{\langle 11:8 \rangle} P_{\langle 7:4 \rangle} G_{\langle 3:0 \rangle} + P_{\langle 11:8 \rangle} P_{\langle 7:4 \rangle} P_{\langle 3:0 \rangle} c_{-1} \\
c_0 &= G_{\langle 0:0 \rangle} + P_{\langle 0:0 \rangle} c_{-1} \\
c_1 &= G_{\langle 1:1 \rangle} + P_{\langle 1:1 \rangle} G_{\langle 0:0 \rangle} + P_{\langle 1:1 \rangle} P_{\langle 0:0 \rangle} c_{-1} \\
c_2 &= G_{\langle 2:2 \rangle} + P_{\langle 2:2 \rangle} G_{\langle 1:1 \rangle} + P_{\langle 2:2 \rangle} P_{\langle 1:1 \rangle} G_{\langle 0:0 \rangle} + P_{\langle 2:2 \rangle} P_{\langle 1:1 \rangle} P_{\langle 0:0 \rangle} c_{-1} \\
c_4 &= G_{\langle 4:4 \rangle} + P_{\langle 4:4 \rangle} c_3 \\
c_5 &= G_{\langle 5:5 \rangle} + P_{\langle 5:5 \rangle} G_{\langle 4:4 \rangle} + P_{\langle 5:5 \rangle} P_{\langle 4:4 \rangle} c_3 \\
c_6 &= G_{\langle 6:6 \rangle} + P_{\langle 6:6 \rangle} G_{\langle 5:5 \rangle} + P_{\langle 6:6 \rangle} P_{\langle 5:5 \rangle} G_{\langle 4:4 \rangle} + P_{\langle 6:6 \rangle} P_{\langle 5:5 \rangle} P_{\langle 4:4 \rangle} c_3 \\
c_8 &= G_{\langle 8:8 \rangle} + P_{\langle 8:8 \rangle} c_7 \\
c_9 &= G_{\langle 9:9 \rangle} + P_{\langle 9:9 \rangle} G_{\langle 8:8 \rangle} + P_{\langle 9:9 \rangle} P_{\langle 8:8 \rangle} c_7 \\
c_{10} &= G_{\langle 10:10 \rangle} + P_{\langle 10:10 \rangle} G_{\langle 9:9 \rangle} + P_{\langle 10:10 \rangle} P_{\langle 9:9 \rangle} G_{\langle 8:8 \rangle} + P_{\langle 10:10 \rangle} P_{\langle 9:9 \rangle} P_{\langle 8:8 \rangle} c_7 \\
c_{12} &= G_{\langle 12:12 \rangle} + P_{\langle 12:12 \rangle} c_{11} \\
c_{13} &= G_{\langle 13:13 \rangle} + P_{\langle 13:13 \rangle} G_{\langle 12:12 \rangle} + P_{\langle 13:13 \rangle} P_{\langle 12:12 \rangle} c_{11} \\
c_{14} &= G_{\langle 14:14 \rangle} + P_{\langle 14:14 \rangle} G_{\langle 13:13 \rangle} + P_{\langle 14:14 \rangle} P_{\langle 13:13 \rangle} G_{\langle 12:12 \rangle} + P_{\langle 14:14 \rangle} P_{\langle 13:13 \rangle} P_{\langle 12:12 \rangle} c_{11}
\end{aligned}$$

1.2.3 carry-lookahead generator

Examples, a unit of a four-input CLA generator and a construction of a 16-bit adder, are shown in slides.

1.2.4 Pseudo Code to make CLA-tree

Generate a n bit carry-lookahead adder with CLA-generators each of which has λ or less inputs.

```

make-CLA-Tree {
  let  $U$  be a list  $(\langle n-1 : n-1 \rangle, \langle n-2 : n-2 \rangle, \dots, \langle 0 : 0 \rangle)$ 
  let  $V \leftarrow U$ 

```

```

while  $|U| > 1$  {
  let  $U' \leftarrow ()$ 
  while  $|U| > \lambda$  {
    take the first  $\lambda$  items from  $U$  out
    let the items be  $\langle i_{\lambda-1} : j_{\lambda-1} \rangle, \langle i_{\lambda-2} : j_{\lambda-2} \rangle, \dots, \langle i_0 : j_0 \rangle$ 
    create a CLA-generator  $\text{clagen}_{\langle i_{\lambda-1} : j_0 \rangle}$  of size  $\lambda$ 
    connect  $P_{\langle i_y : j_y \rangle}$  and  $G_{\langle i_y : j_y \rangle}$  to the  $y$ -th input of the CLA-generator for  $\lambda - 1 \geq y \geq 0$ 
    connect the  $(j_0 - 1)$ -th carry, which will be created later, to the carry-in of the CLA-generator.
    Note that we have  $P_{\langle i_{\lambda-1} : j_0 \rangle}$ ,  $G_{\langle i_{\lambda-1} : j_0 \rangle}$ , and  $c_{i_y}$  for  $\mu - 2 \geq y \geq 0$ 
    put  $\langle i_{\lambda-1} : j_0 \rangle$  at the end of list  $U'$ 
  }
  append the remained list  $U$  to the end of  $U'$ 
  let  $U \leftarrow U'$ 
}
create  $c_{n-1}$  from the inputs of the last CLA-generator
}

```

```

make-CLA-Tree-fully-balanced {
  let  $U$  be a list  $(\langle n-1 : n-1 \rangle, \langle n-2 : n-2 \rangle, \dots, \langle 0 : 0 \rangle)$ 
  let  $V \leftarrow U$ 
  while  $|U| > 1$  {
    let  $m \leftarrow \lceil |U|/\lambda \rceil$ , the number of CLA-generators in the current level
    let  $\lambda' \leftarrow \lceil |U|/m \rceil$ , the size of CLA-generators in the current level
    let  $m_0 \leftarrow m \times \lambda' - |U|$ , the number of CLA-generators of the size  $\lambda' - 1$ 
    let  $m_1 \leftarrow m - m_0$ , the number of CLA-generators of the size  $\lambda'$ 
    let  $U' \leftarrow ()$ 
    for  $x \leftarrow m$  downto 1 {
      let  $\mu$  be  $\lambda'$  if  $x \leq m_1$ , o.w.  $\lambda' - 1$ 
      take the first  $\mu$  items from  $U$  out
      let the items be  $\langle i_{\mu-1} : j_{\mu-1} \rangle, \langle i_{\mu-2} : j_{\mu-2} \rangle, \dots, \langle i_0 : j_0 \rangle$ 
      create a CLA-generator  $\text{clagen}_{\langle i_{\mu-1} : j_0 \rangle}$  of size  $\mu$ 
      connect  $P_{\langle i_y : j_y \rangle}$  and  $G_{\langle i_y : j_y \rangle}$  to the  $y$ -th input of the CLA-generator for  $\mu - 1 \geq y \geq 0$ 
      connect the  $(j_0 - 1)$ -th carry, which will be created later, to the carry-in of the CLA-generator.
      Note that we have  $P_{\langle i_{\mu-1} : j_0 \rangle}$ ,  $G_{\langle i_{\mu-1} : j_0 \rangle}$ , and  $c_{i_y}$  for  $\mu - 2 \geq y \geq 0$ 
      put  $\langle i_{\mu-1} : j_0 \rangle$  at the end of list  $U'$ 
    }
    let  $U \leftarrow U'$ 
  }
  create  $c_{n-1}$  from the inputs of the last CLA-generator
}

```