Outline

- Digital CMOS design
- Arithmetic operators
  - Adders
  - Comparators
  - Shifters
  - Multipliers

Comparators

Comparing a natural number to a constant : =

Let consider a natural number **a** coded on 8 bits using Natural Binary Code

$$a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$$

$$= ?$$

$$0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$$

$$\downarrow$$

$$0\ /\ 1$$

Comparators

Comparing a natural number to zero : =

Boolean function

Null = 1 if

$$\overline{a_7} \cdot \overline{a_6} \cdot \overline{a_5} \cdot \overline{a_4} \cdot \overline{a_3} \cdot \overline{a_2} \cdot \overline{a_1} \cdot \overline{a_0} = 1$$

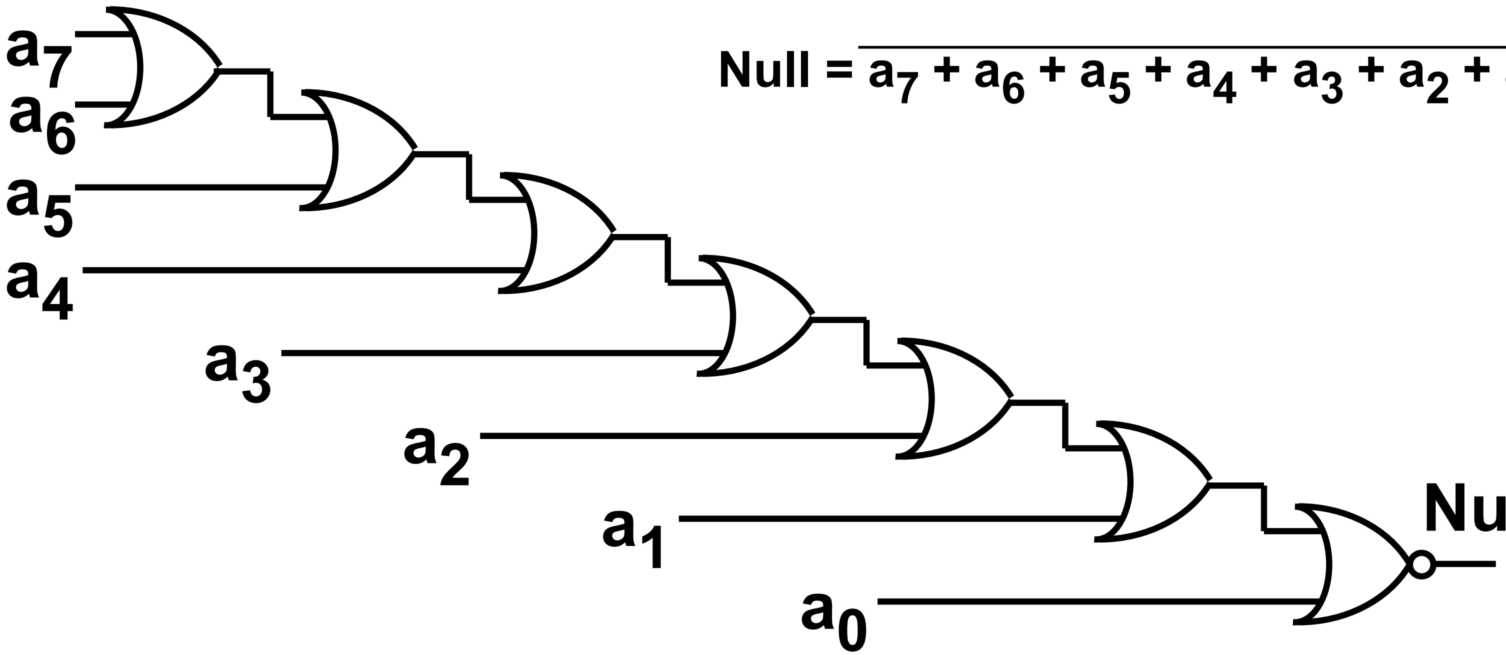
$$\mathrm{Null} = \overline{a_7 + a_6 + a_5 + a_4 + a_3 + a_2 + a_1 + a_0}$$

Comparators

Comparing a natural number to zero : =

Implementation

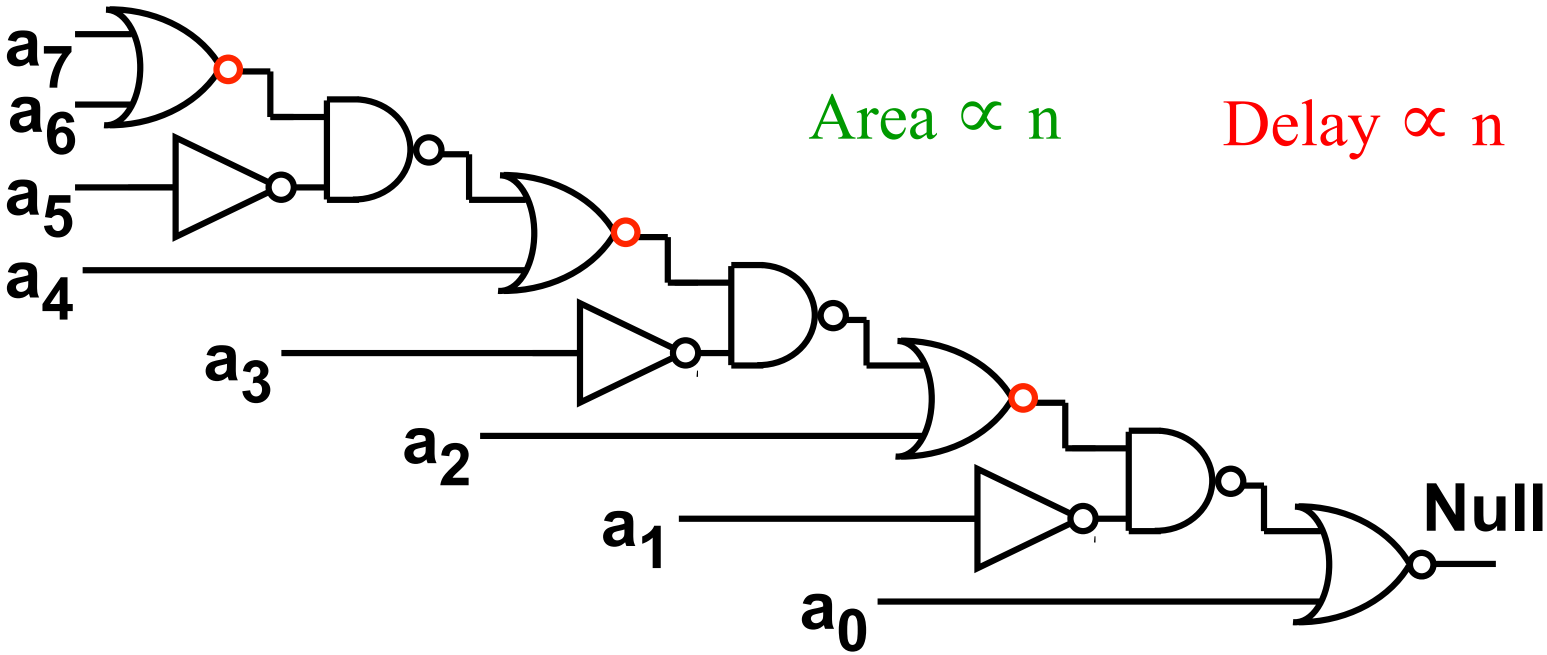
$$\text{Null} = \overline{a_7 + a_6 + a_5 + a_4 + a_3 + a_2 + a_1 + a_0}$$



Comparators

Comparing a natural number to zero : =

Implementation



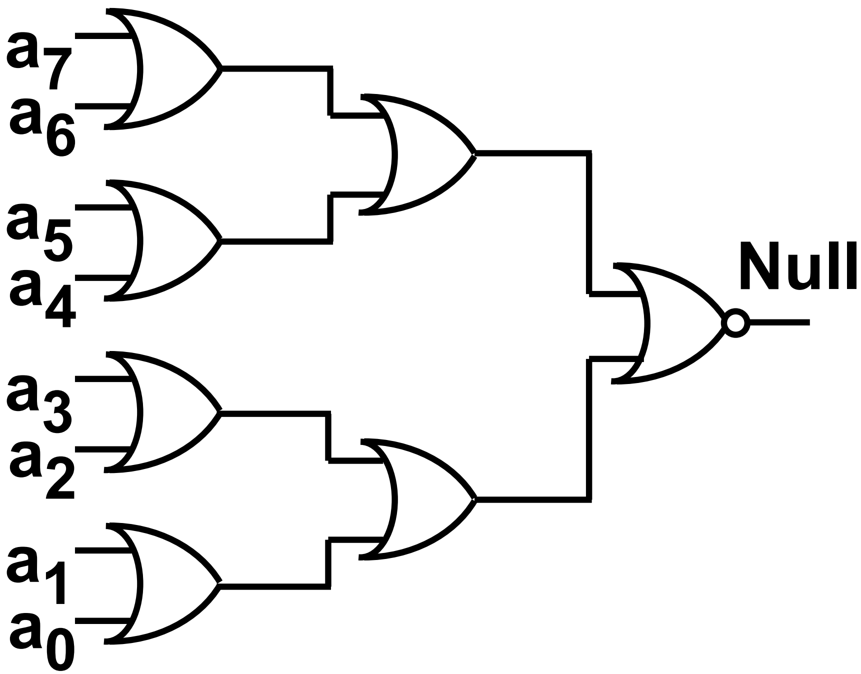
Area ∝ n

Delay ∝ n

Comparators

Comparing a natural number to zero : =

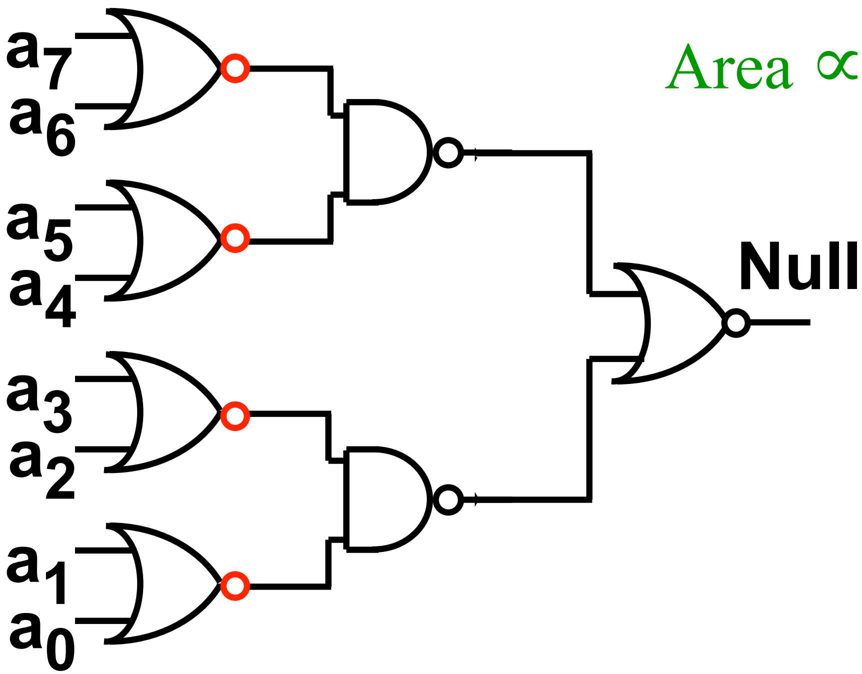
Implementation improvement



Comparators

Comparing a natural number to zero : =

Implementation improvement



Area ∝ n   Delay ∝ *log*(n)

Comparators

Comparing two natural numbers : =

Let consider two natural numbers **a** and **b** coded on 8 bits using Natural Binary Code

$$\mathbf{a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0}$$
$$= ?$$
$$\mathbf{b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0}$$
$$\downarrow$$

0 / 1

Comparators

Comparing two natural numbers : =

Boolean function

a Equal b if :$a_7=b_7$ and $a_6=b_6$ and **...** and $a_0=b_0$

a Equal b if :$\overline{(a_7 \oplus b_7)} \ . \ \ldots \ . \ \overline{(a_0 \oplus b_0)} = 1$

$$\text{Equal} = \overline{(a_7 \oplus b_7) + \ \ldots \ + (a_0 \oplus b_0)}$$

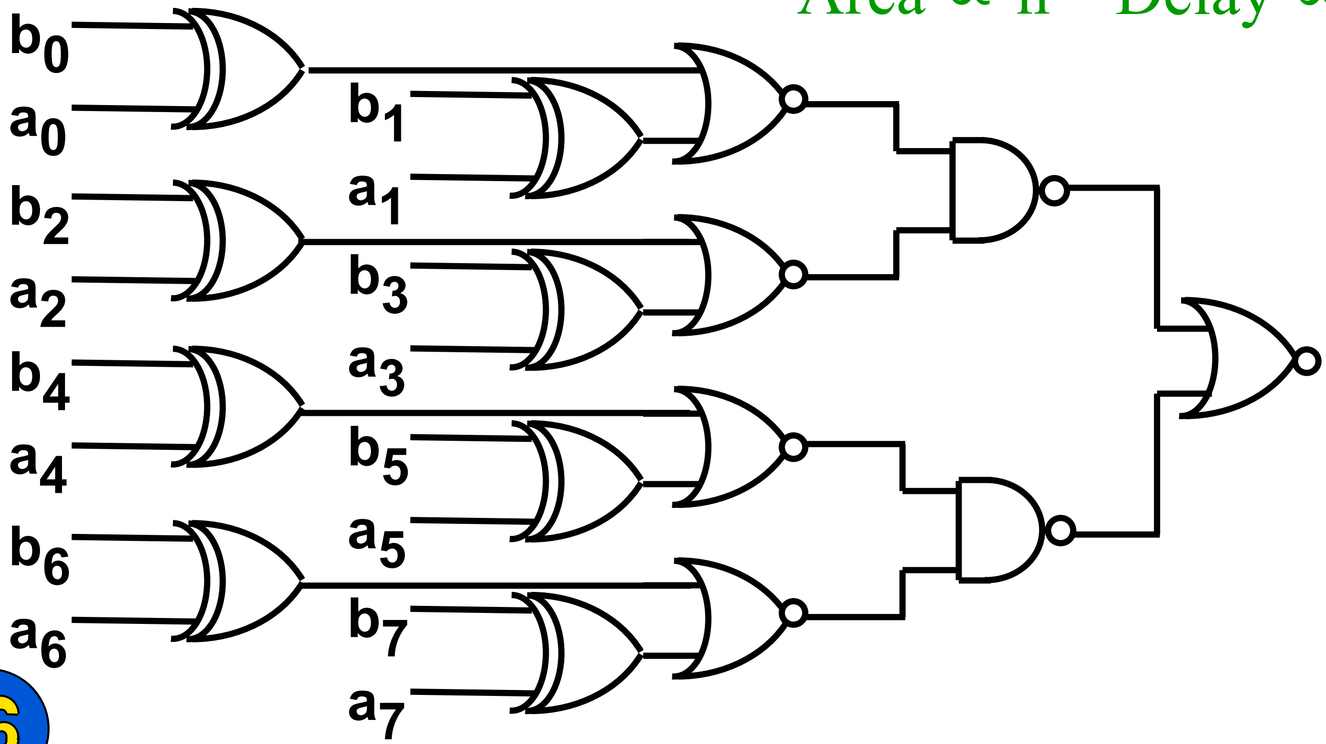
$$\text{Equal} = (\ e_7\ ) + \ \ldots \ + (\ e_0\ )$$

Comparators

Comparing two natural numbers : =

Implementation

Area ∝ n   Delay ∝ *log*(n)



Comparators

Comparing two natural numbers : <

Let consider two natural numbers **a** and **b** coded on 8 bits using Natural Binary Code

$$\mathbf{a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0}$$

$$< ?$$

$$\mathbf{b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0}$$

$$\downarrow$$

0 / 1

Comparators

Comparing two natural numbers : <

Boolean function

a < b if : $a_7 < b_7$ or ($a_7 = b_7$ and ($a_6 < b_6$ or ($a_6 = b_6$ and **...**)))

$$a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$$

< ?

$$b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$$

$\downarrow$

0 / 1

Comparators

Comparing two natural numbers : <

Boolean function

$$\mathbf{a < b} \text{ if : } \mathbf{a_7 < b_7} \text{ or } (\mathbf{a_7 = b_7} \text{ and } (\mathbf{a_6 < b_6} \text{ or } (\mathbf{a_6 = b_6} \text{ and } \mathbf{\ldots}\ )))$$

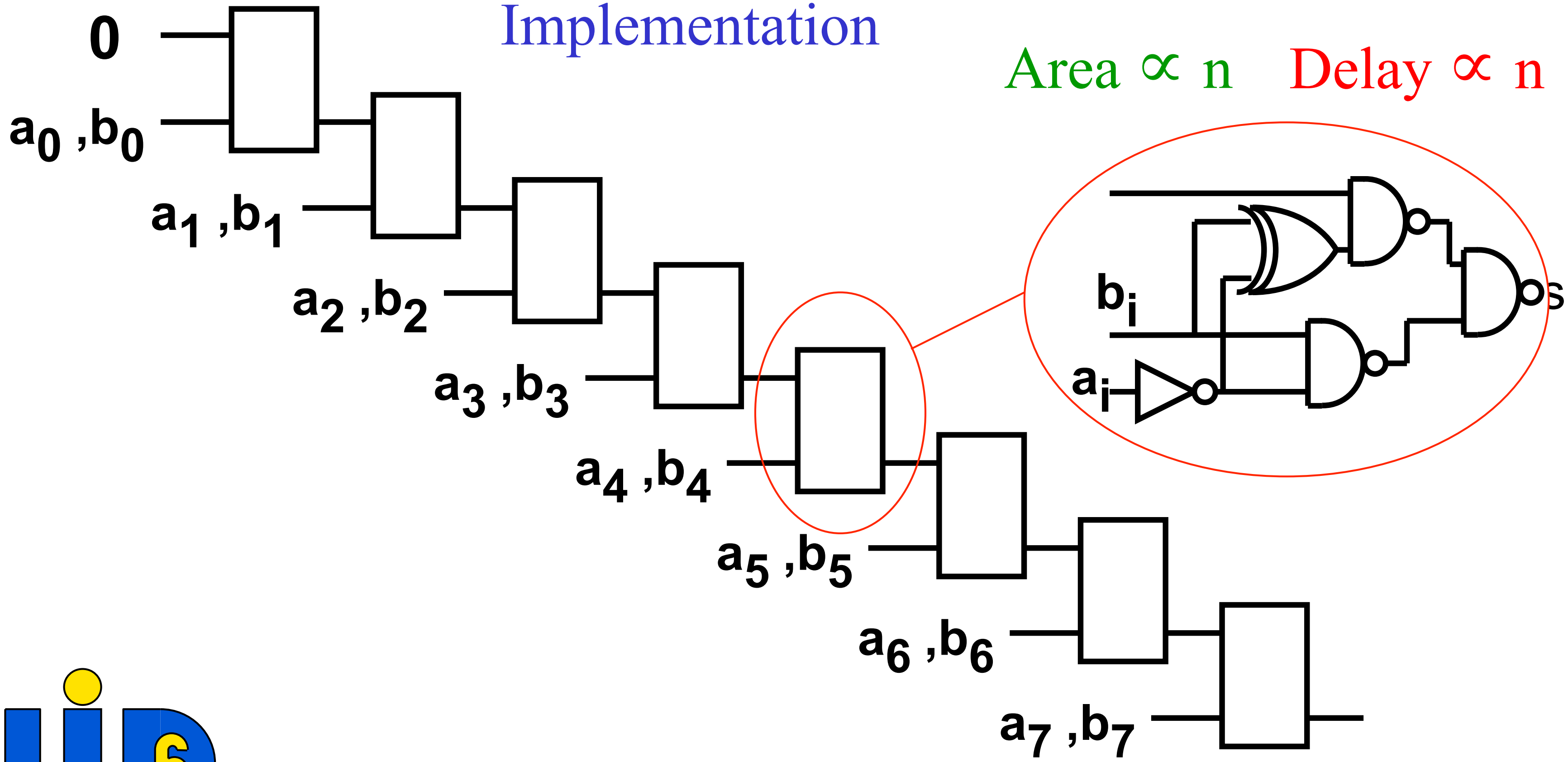
$$\mathbf{a < b} \text{ if : } \overline{\mathbf{a_7}}\mathbf{b_7} + (\overline{(\mathbf{a_7 \oplus b_7})} \cdot (\overline{\mathbf{a_6}}\mathbf{b_6} + (\overline{(\mathbf{a_6 \oplus b_6})} \cdot \mathbf{\ldots}\ )))$$

Comparators

Comparing two natural numbers : <

Implementation

Area ∝ n   Delay ∝ n



Comparators

Comparing two natural numbers : <

Implementation Improvement

$$\mathbf{a < b} \text{ if : } \mathbf{a_7 < b_7} \text{ or } (\mathbf{a_7 = b_7} \text{ and } (\mathbf{a_6 < b_6} \text{ or } (\mathbf{a_6 = b_6} \text{ and } \ldots)))$$

$$\mathbf{a < b} \text{ if : } \overline{\mathbf{a_7}}\mathbf{b_7} + ((\overline{\mathbf{a_7} \oplus \mathbf{b_7}}) \,.\, (\overline{\mathbf{a_6}}\mathbf{b_6} + ((\overline{\mathbf{a_6} \oplus \mathbf{b_6}}) . \ldots)))$$

$$\overline{\mathbf{a_i}}\mathbf{b_i} + \overline{(\mathbf{a_i} \oplus \mathbf{b_i})} \text{ .previous}$$

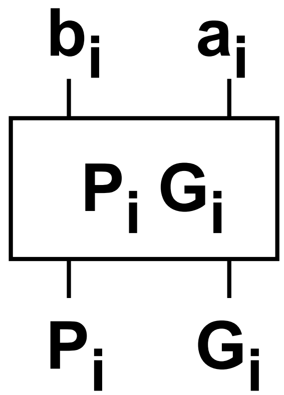
Propagation

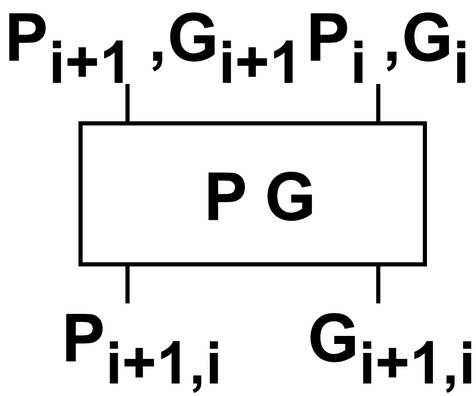
Generation

Comparators

Comparing two natural numbers : <

Implementation Improvement





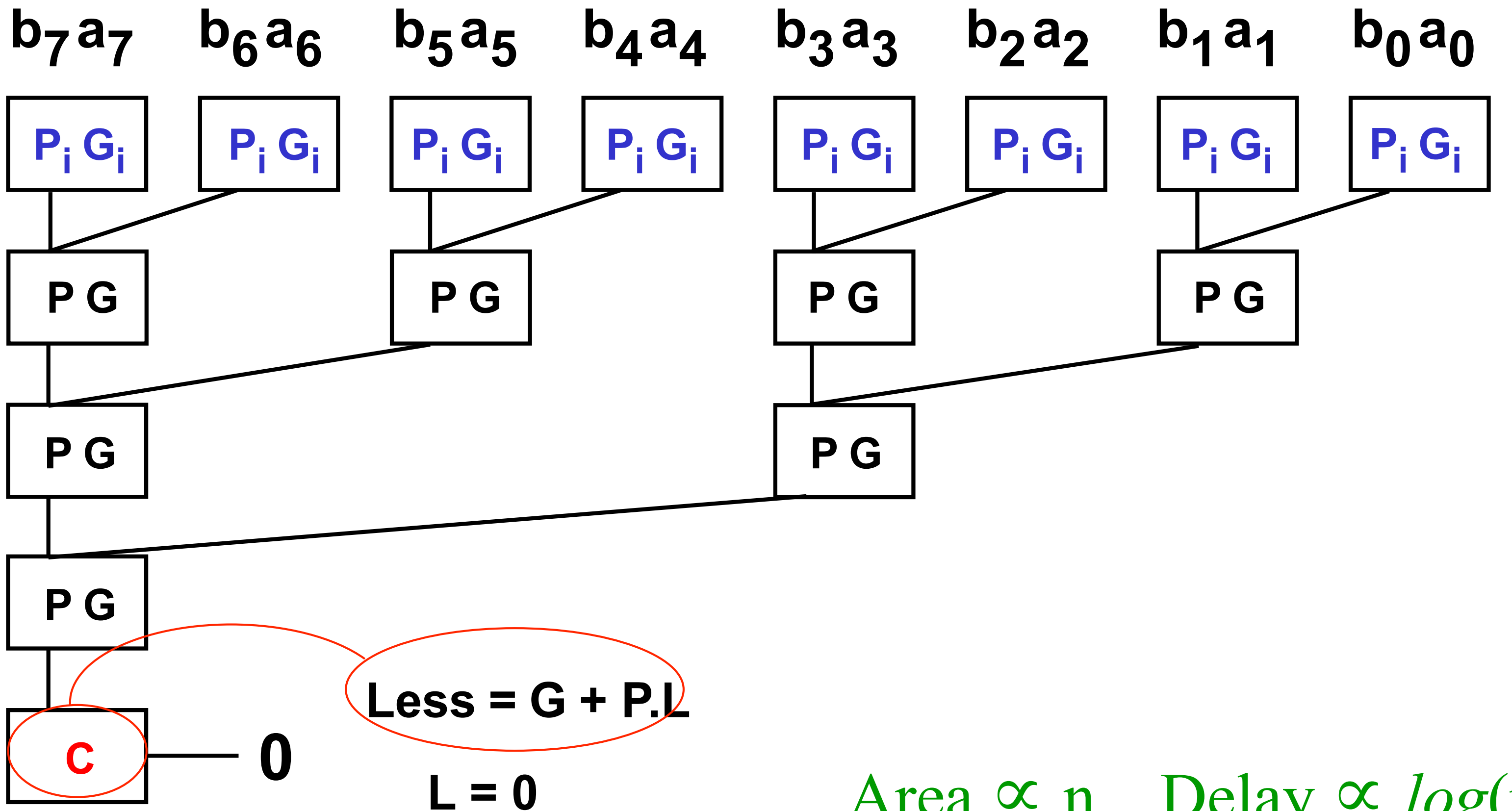
$$G_i = \bar{a}_i\, b_i$$

$$P_i = \bar{a}_i \oplus b_i$$

$$G_{i+1,\, i} = G_{i+1} + G_i . P_{i+1}$$

$$P_{i+1,\, i} = P_i \,.\, P_{i+1}$$

Comparators



Area ∝ n   Delay ∝ *log*(n)

Comparators