Outline

- Digital CMOS design
- Arithmetic operators
- Sequential functions

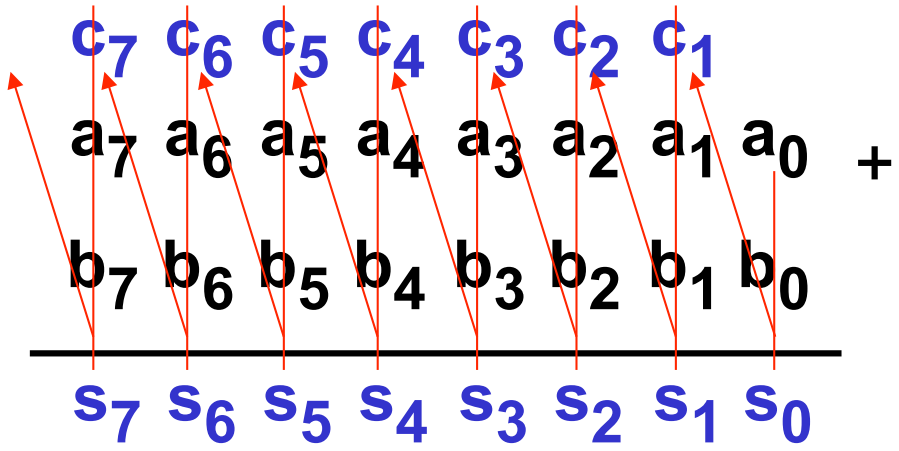
Outline

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

Adders

Adding two natural numbers

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

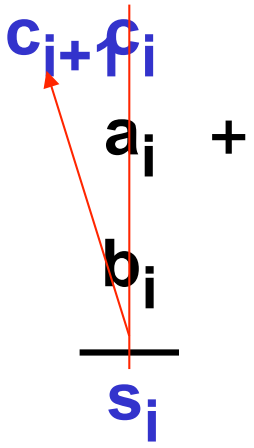


Adders

Adding two natural numbers

At each stage, I need to sum 3 single bit numbers a_i b_i c_i

The carry out of the stage i is the input carry of the next stage



s_i and c_{i+1} are Boolean functions of a_i b_i c_i

Adders

Adding two natural numbers

c_i \ $a_i b_i$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

s_i

$$s_i = a_i \oplus b_i \oplus c_i$$

c_i \ $a_i b_i$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

c_{i+1}

$$c_{i+1} = a_i.b_i + a_i.c_i + b_i.c_i$$

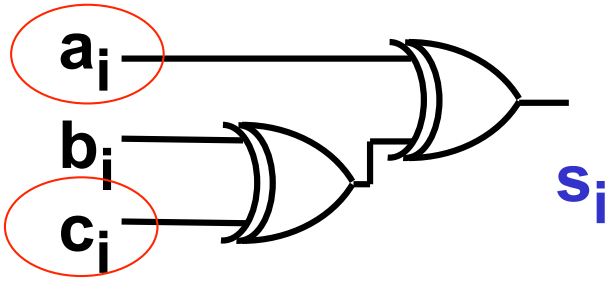
Adders

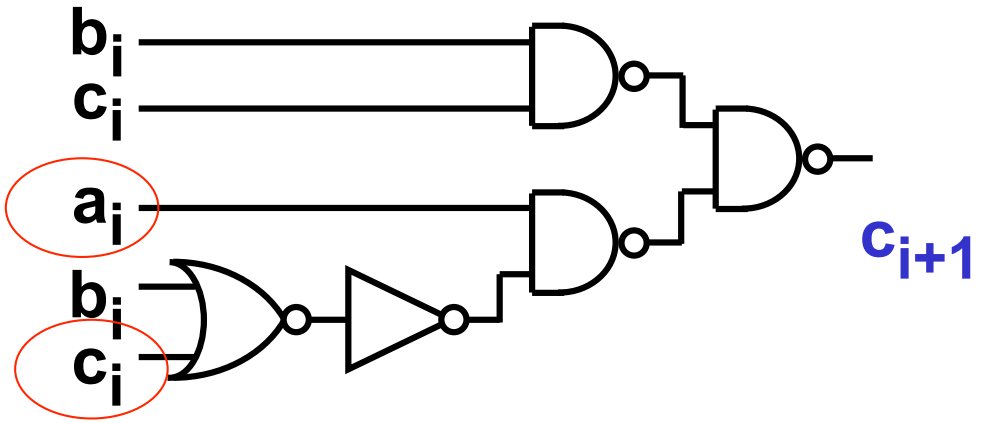
Adding two natural numbers

$$s_i = a_i \oplus b_i \oplus c_i$$

$$c_{i+1} = a_i.b_i + a_i.c_i + b_i.c_i$$

$$c_{i+1} = a_i . (b_i + c_i) + b_i.c_i$$





Addition delay depends on the delay of c_i to c_{i+1}

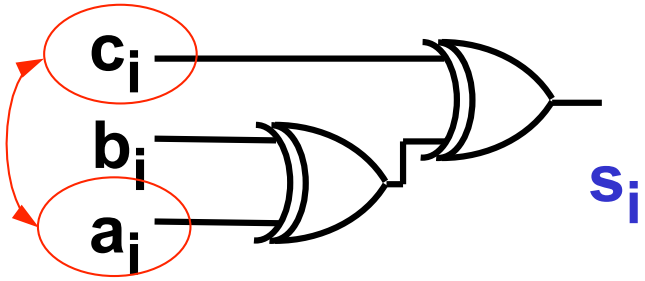
Adders

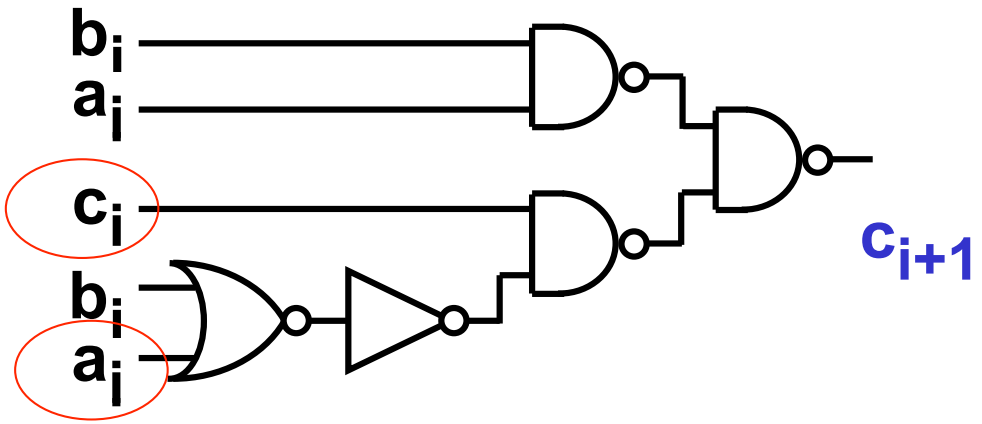
Adding two natural numbers

$s_i = a_i \oplus b_i \oplus c_i$

$c_{i+1} = a_i.b_i + a_i.c_i + b_i.c_i$

$c_{i+1} = a_i.b_i + (a_i+b_i)\ ..c_i$





Adders

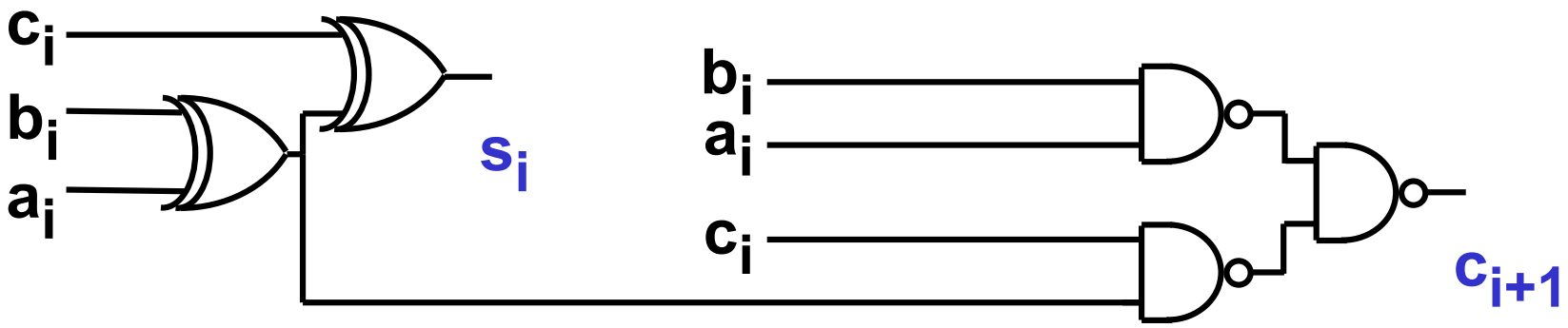
Adding two natural numbers

$s_i = a_i \oplus b_i \oplus c_i$

$c_{i+1} = a_i.b_i + a_i.c_i + b_i.c_i$

$c_{i+1} = a_i.b_i + (a_i+b_i) . c_i$

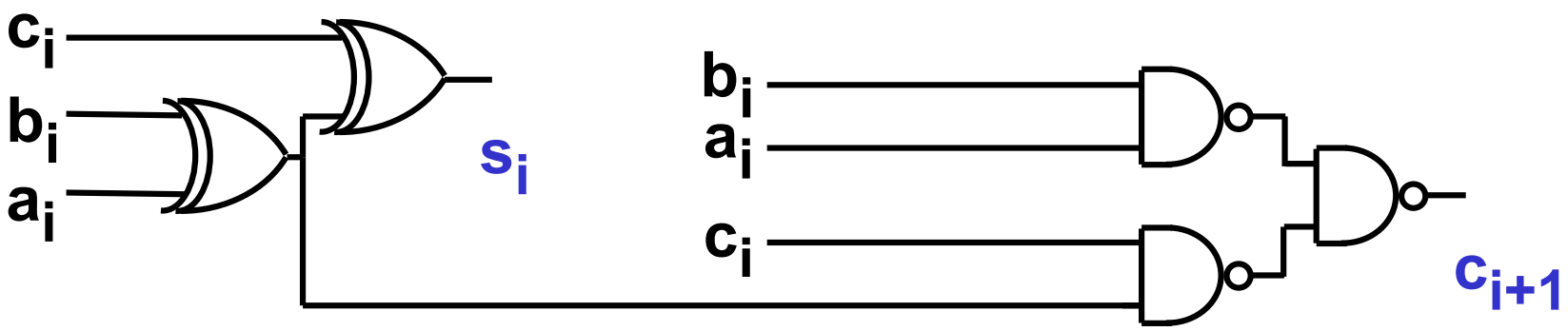
$c_{i+1} = a_i.b_i + (a_i \oplus b_i) . c_i$



Adders

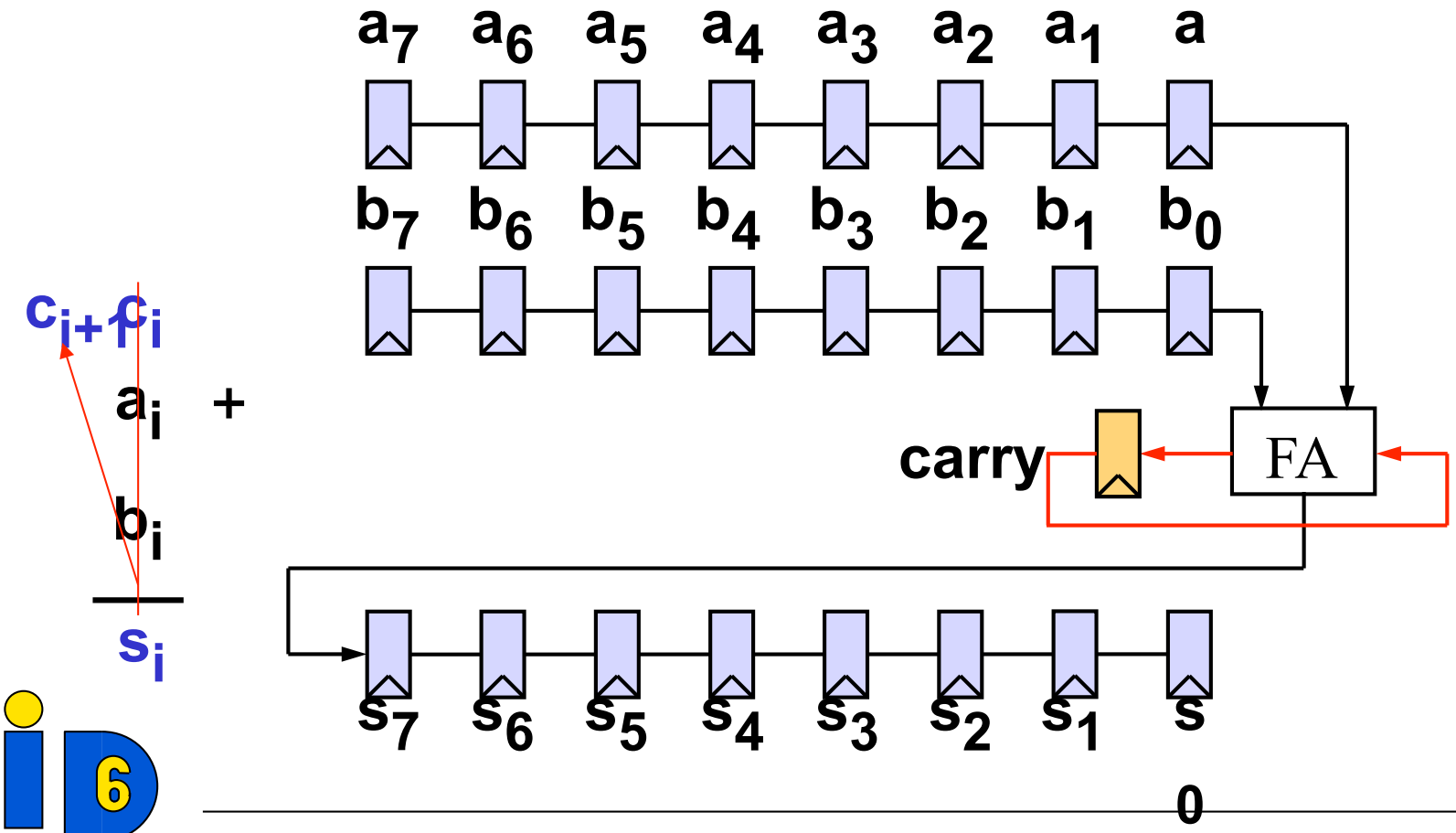
Adding two natural numbers

The circuit generating s_i and c_{i+1} is called a Full Adder (FA)



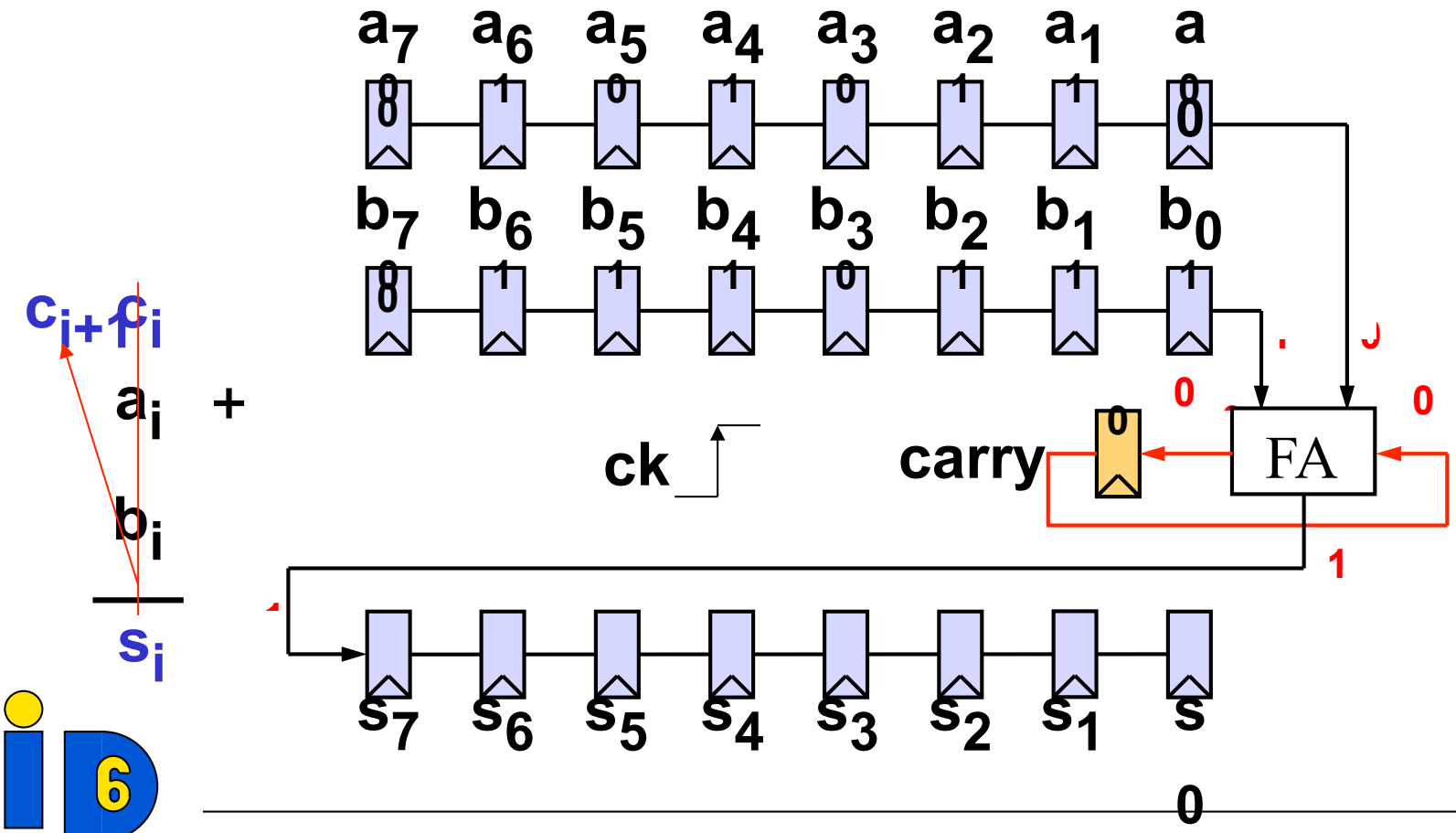
Adders

Adding two natural numbers



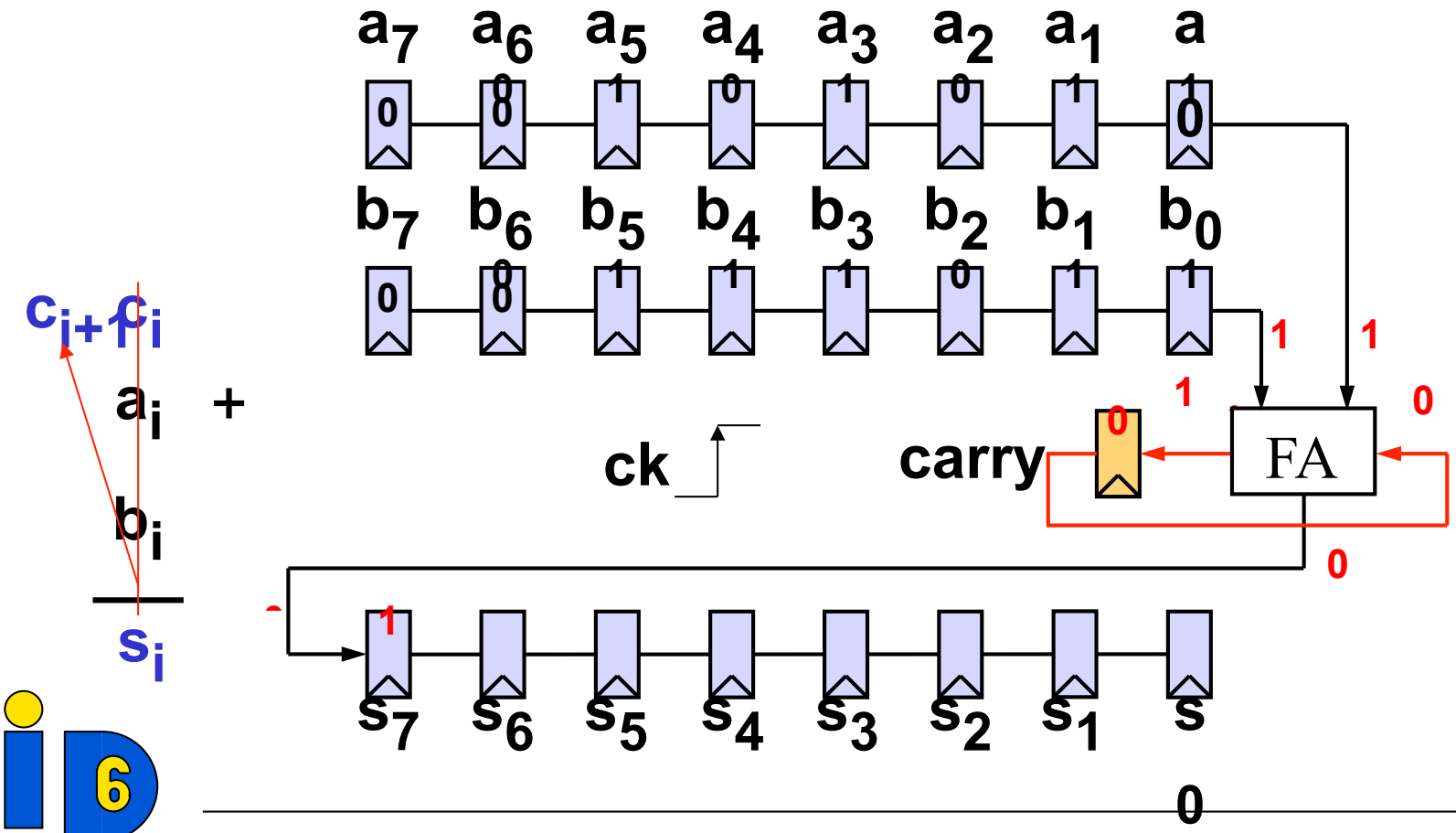
Adders

Adding two natural numbers



Adders

Adding two natural numbers



Adders

Adding two natural numbers

Sequential Adder

Area $\propto$ n Delay $\propto$ n cycles

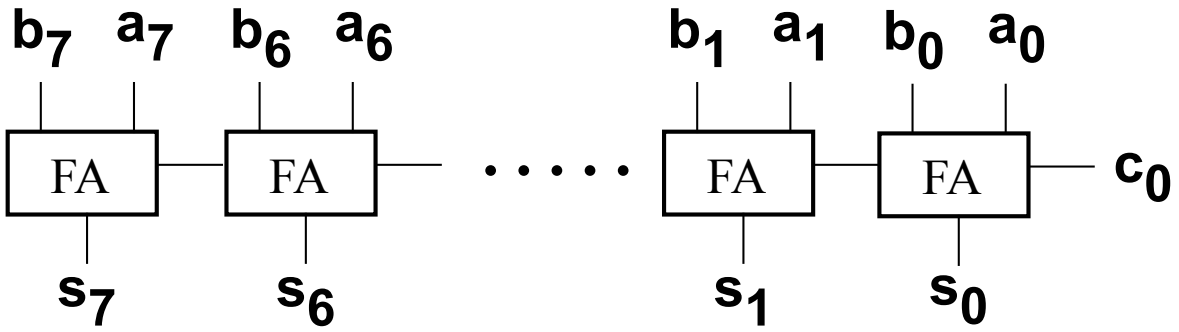
Timing should be improved

Adders

Adding two natural numbers

At each stage, I need to sum 3 single bit numbers a_i b_i c_i

The carry out of the stage i is the input carry of the next stage



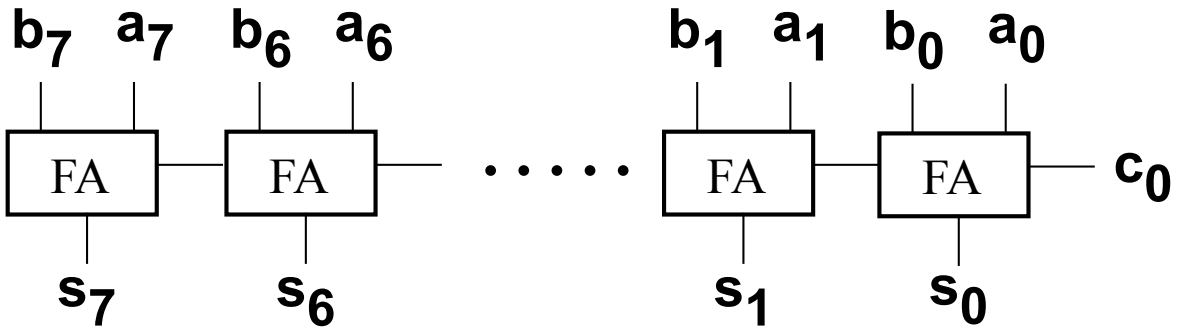
Ripple Carry Adder (RCA)

Adders

Adding two natural numbers

Ripple Carry Adder (RCA)

Area ∝ n Delay ∝ n

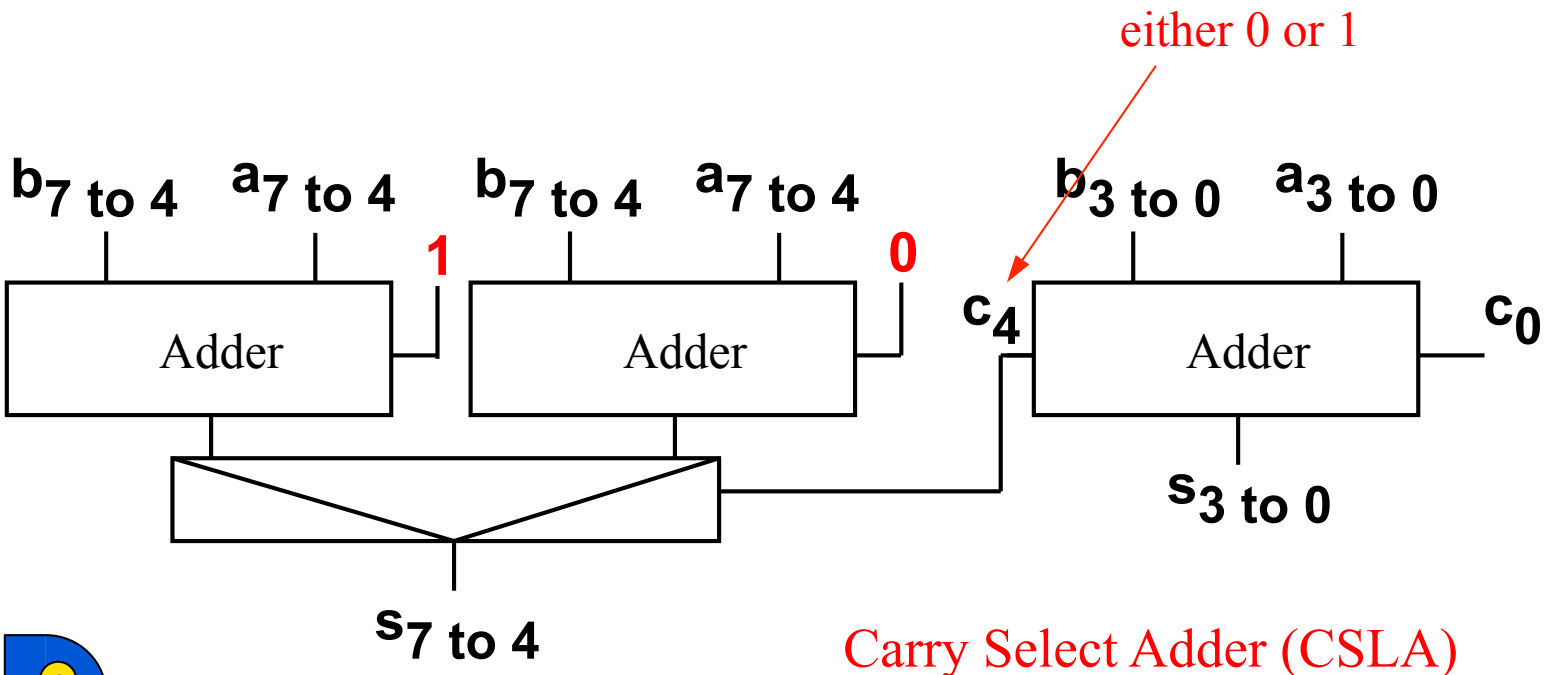


Timing should be improved

Adders

Adding two natural numbers

Acceleration techniques



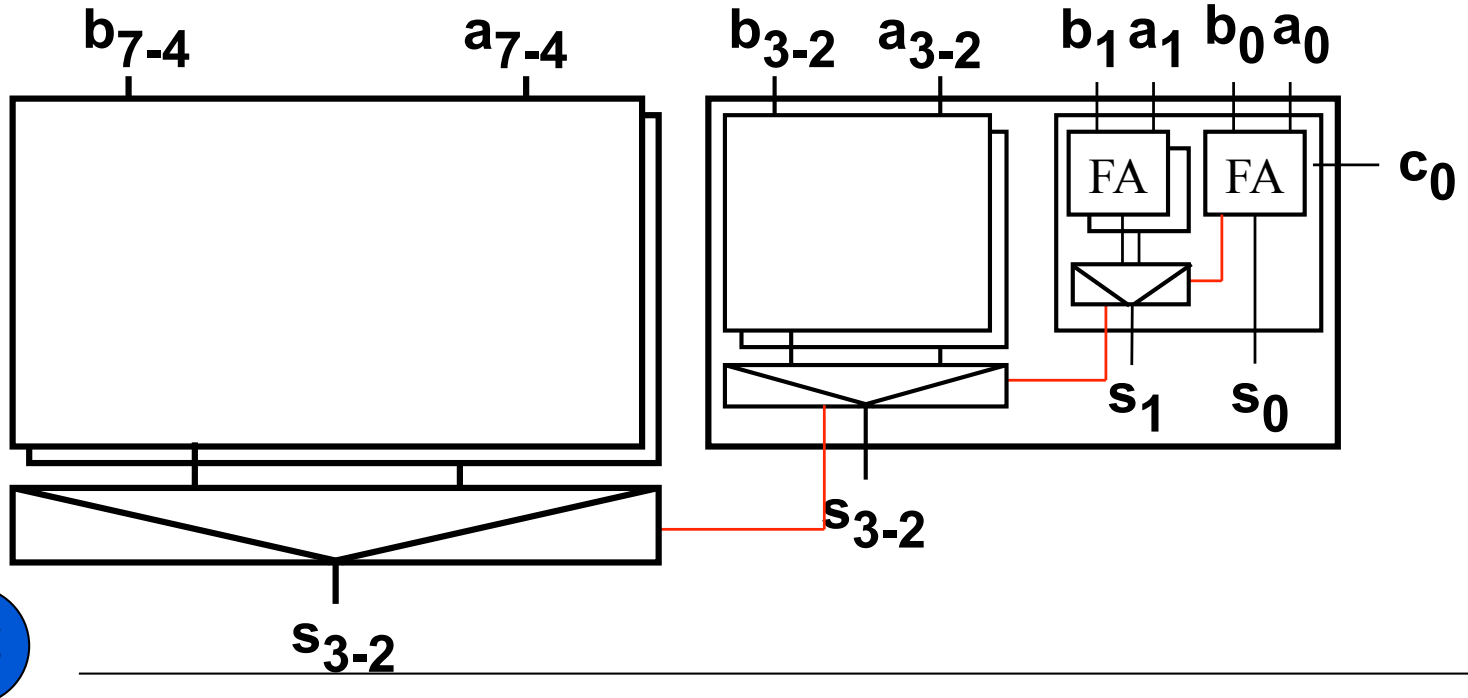
Carry Select Adder (CSLA)

Adders

Adding two natural numbers

Carry Select Adder (CSLA)

Area $\propto n^{log(3)} = n^{1.585}$ Delay $\propto log\,(n)$



Adders

Adding two natural numbers

Acceleration techniques

	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$a_i\ b_i$

c_i

c_{i+1}

carry out depends on carry in

carry out does not depend on carry in

00	01	11	10
absorption	propagation	generation	propagation

$a_i\ b_i$

Adders

Adding two natural numbers

Acceleration techniques

c_i \ $a_i b_i$	00	01	11	10
0	0	0	1	0
1	0	1	1	1

c_{i+1}

00	01	11	10	$a_i b_i$
absorption	propagation	generation	propagation	

$$G_i = a_i b_i$$

$$P_i = a_i \oplus b_i$$

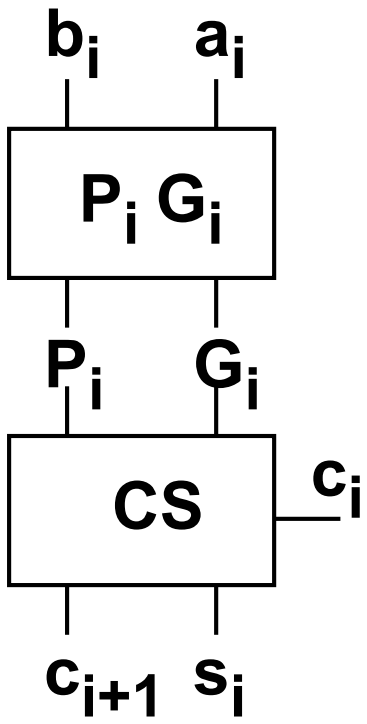
$$c_{i+1} = G_i + P_i c_i$$

$$s_i = P_i \oplus c_i$$

Adders

Adding two natural numbers

Acceleration techniques



$G_i = a_i\, b_i$

$P_i = a_i \oplus b_i$

$c_{i+1} = G_i + P_i\, c_i$

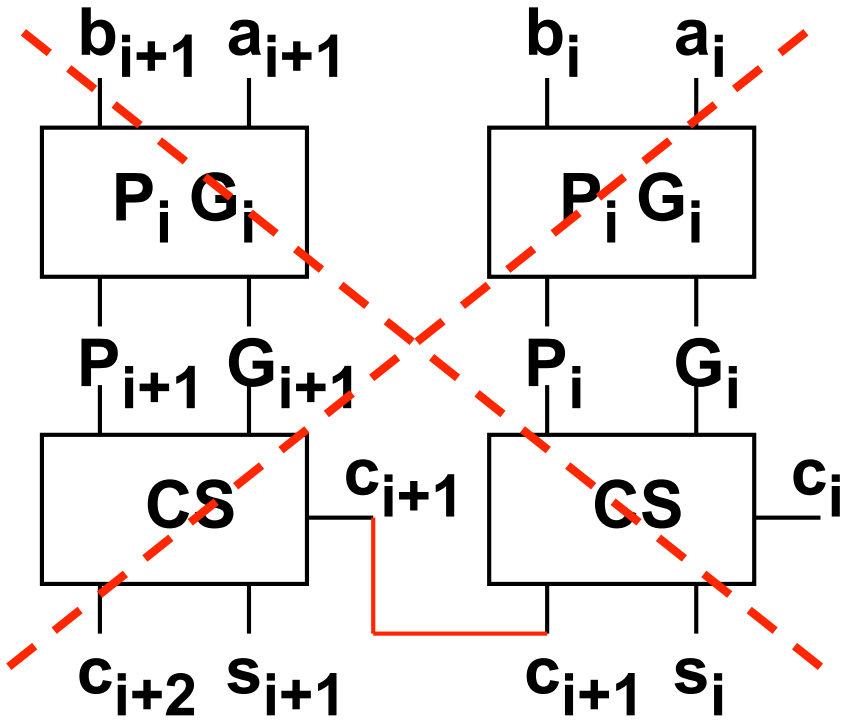
$s_i = P_i \oplus c_i$

$a_i\, b_i$	00	01	11	10
	absorption	propagation	generation	propagation

Adders

Adding two natural numbers

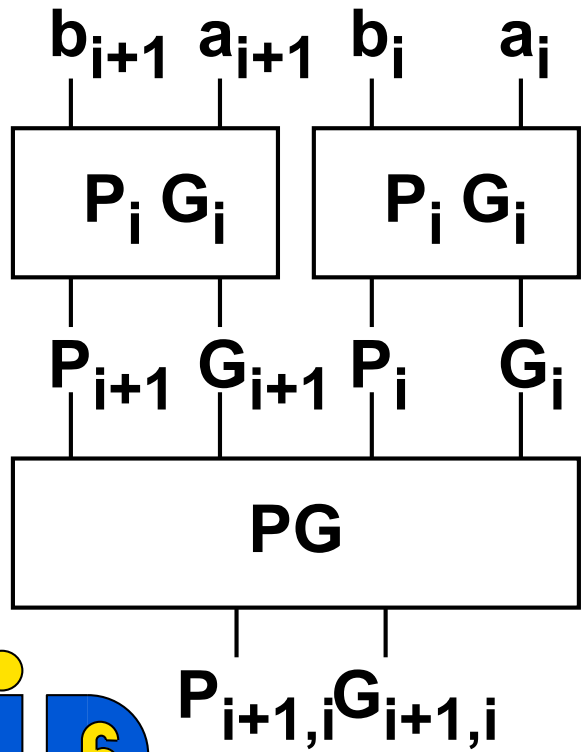
Acceleration techniques



Adders

Adding two natural numbers

Acceleration techniques



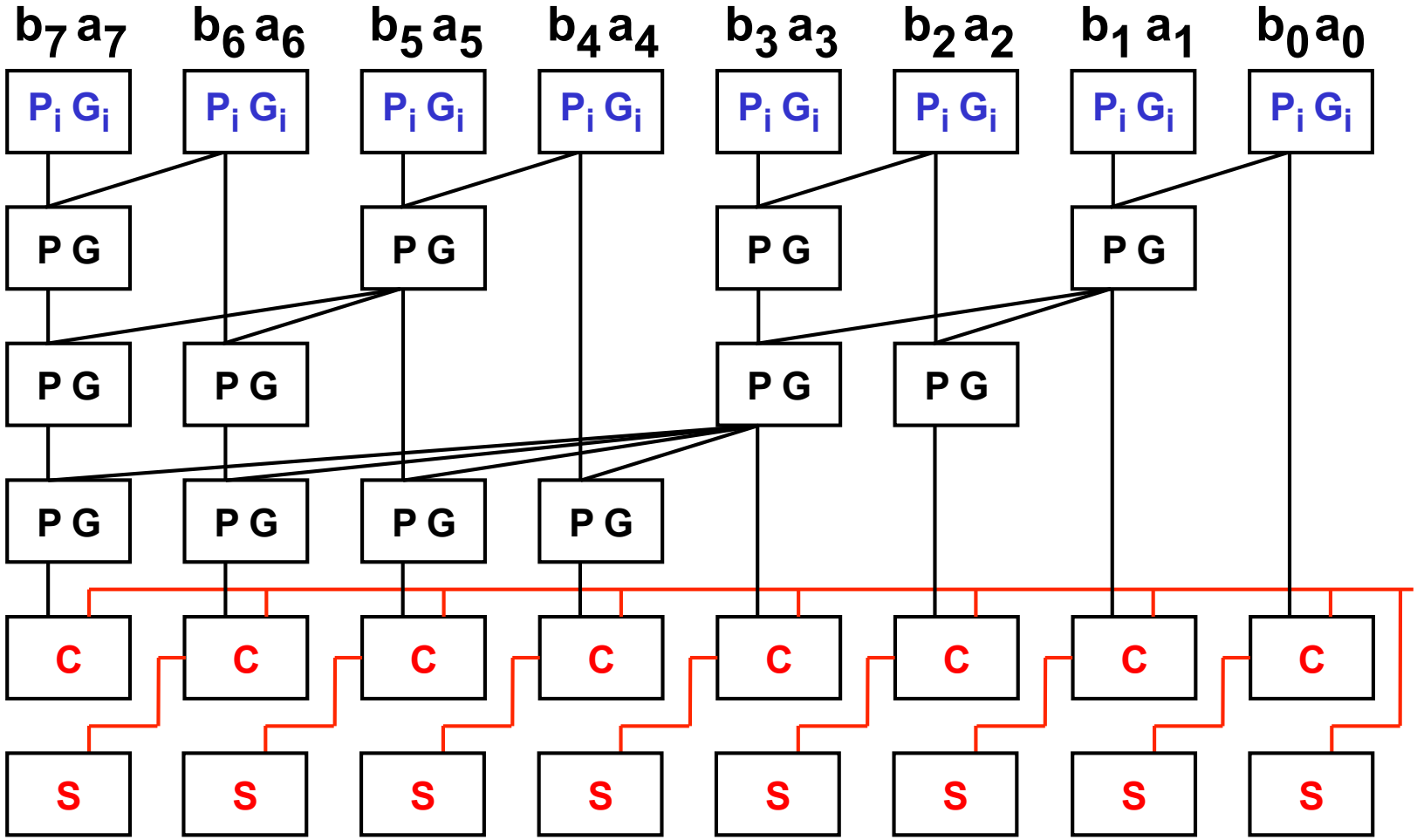
$$G_{i+1} = a_{i+1}\, b_{i+1} \quad G_i = a_i\, b_i$$

$$P_{i+1} = a_{i+1} \oplus b_{i+1} \quad P_i = a_i \oplus b_i$$

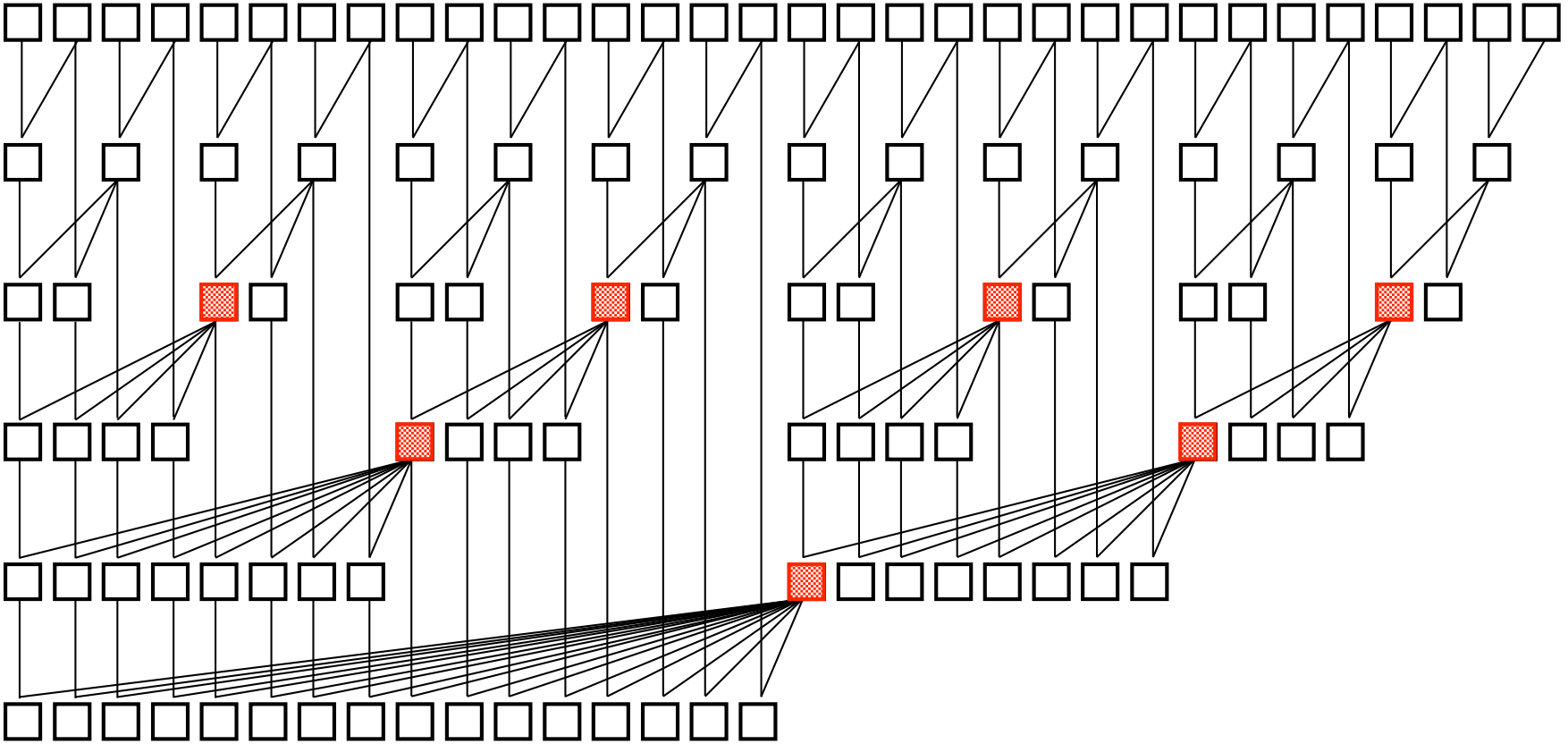
$$G_{i+1,\,i} = G_{i+1} + G_i . P_{i+1}$$

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

Adders

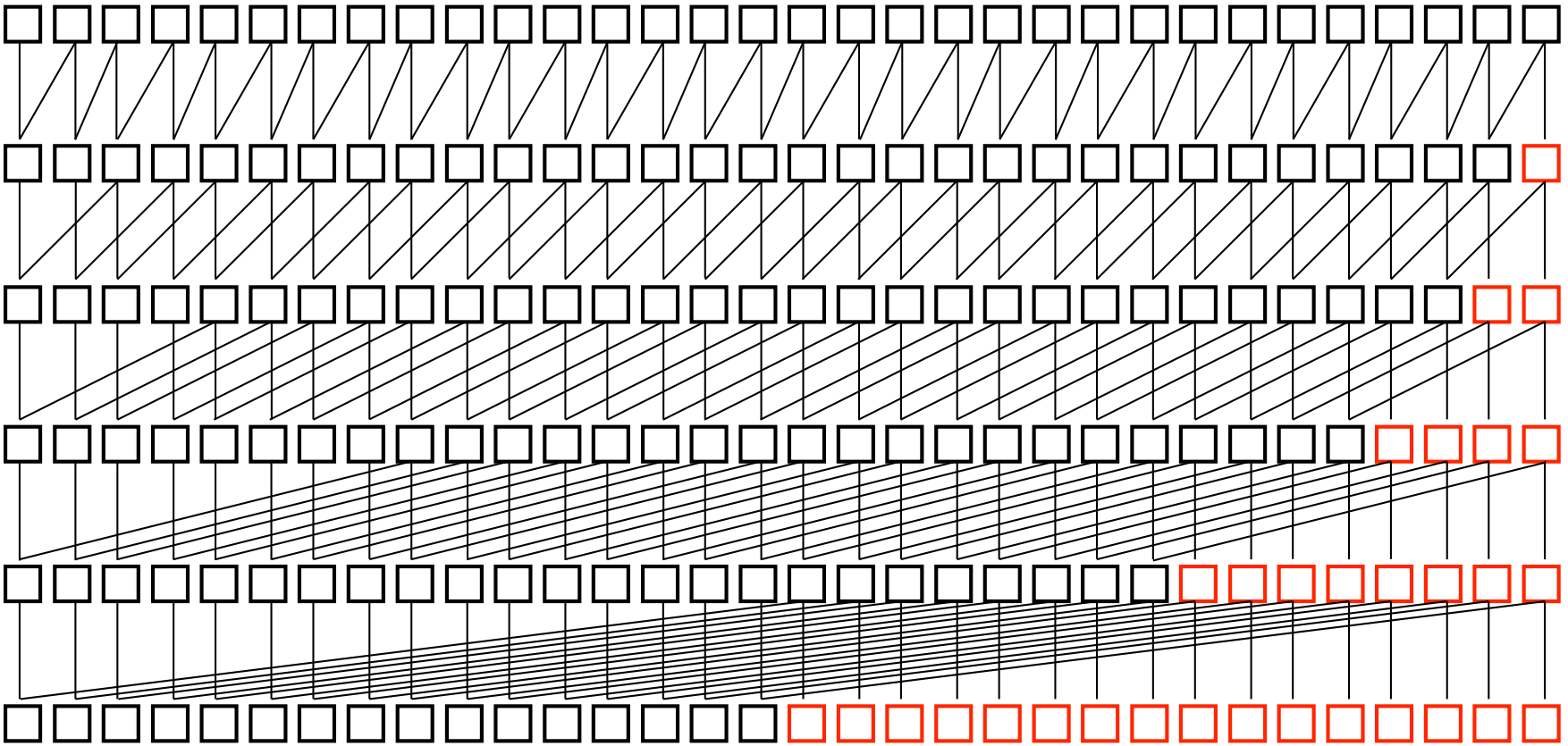


Adders



Slansky Adder

Adders



Kogge-Stone Adder

Adders

Adding two natural numbers (summary)

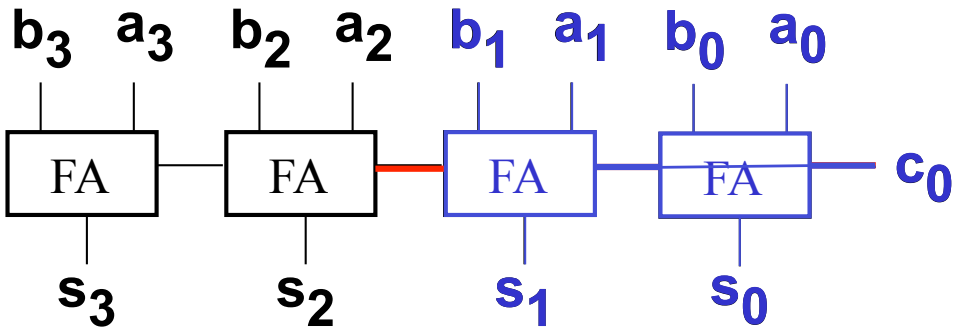
	Area	Delay
Ripple Carry (RCA)	n	n
Carry Select (CSLA)	$n^{log(3)}$	log (n)
Carry Lookahead (CLA)	n log (n)	log (n)
Magic Adder	n	Cste

More improvements ?

Adders

Adding two natural numbers

Ripple Carry Adder (RCA)

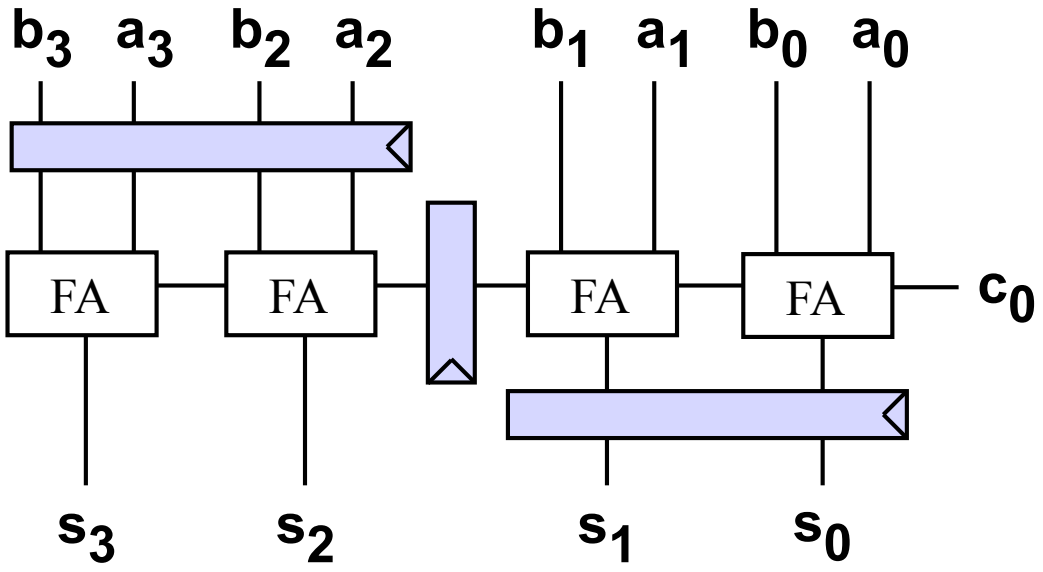


Adders

Adding two natural numbers

Ripple Carry Adder (RCA)

pipelining

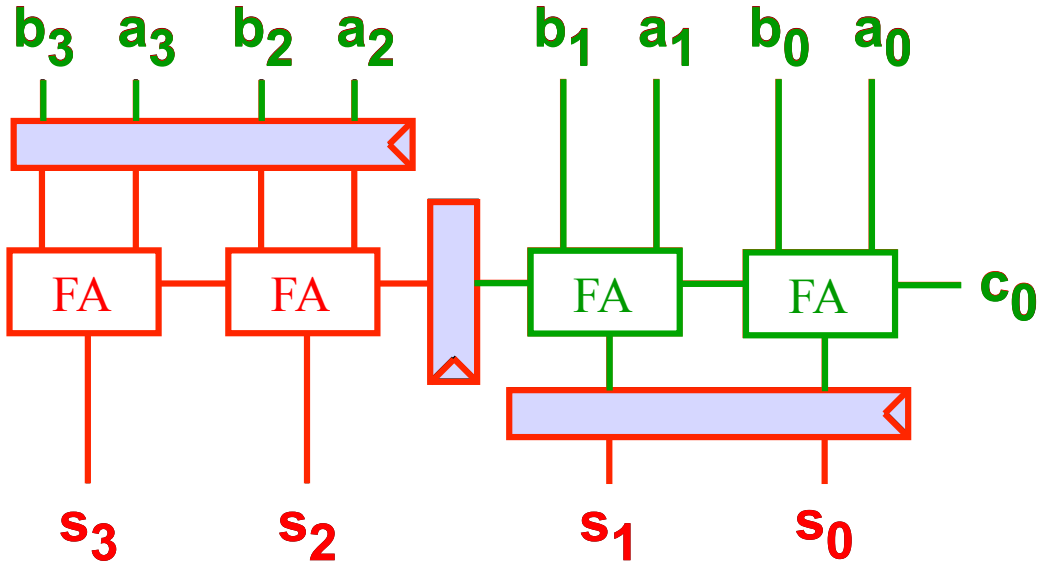


Adders

Adding two natural numbers

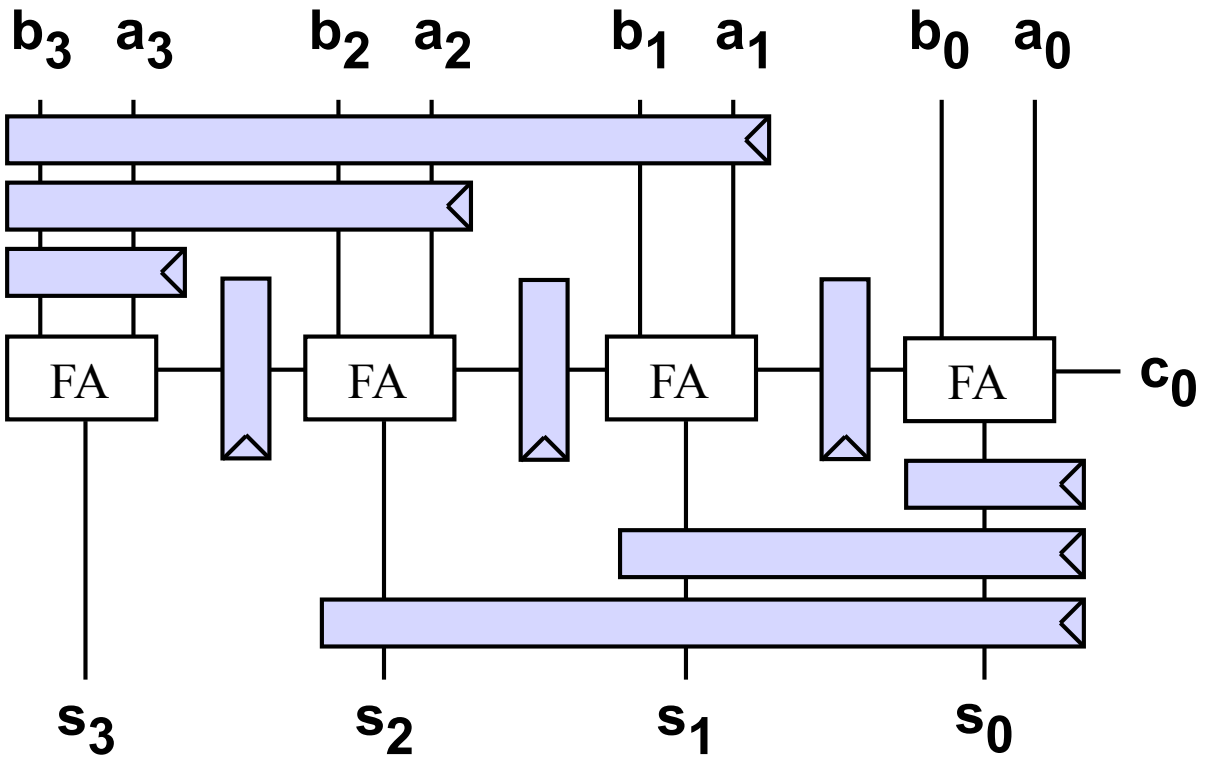
Ripple Carry Adder (RCA)

pipelining



Adders

Adding two natural numbers



Adders

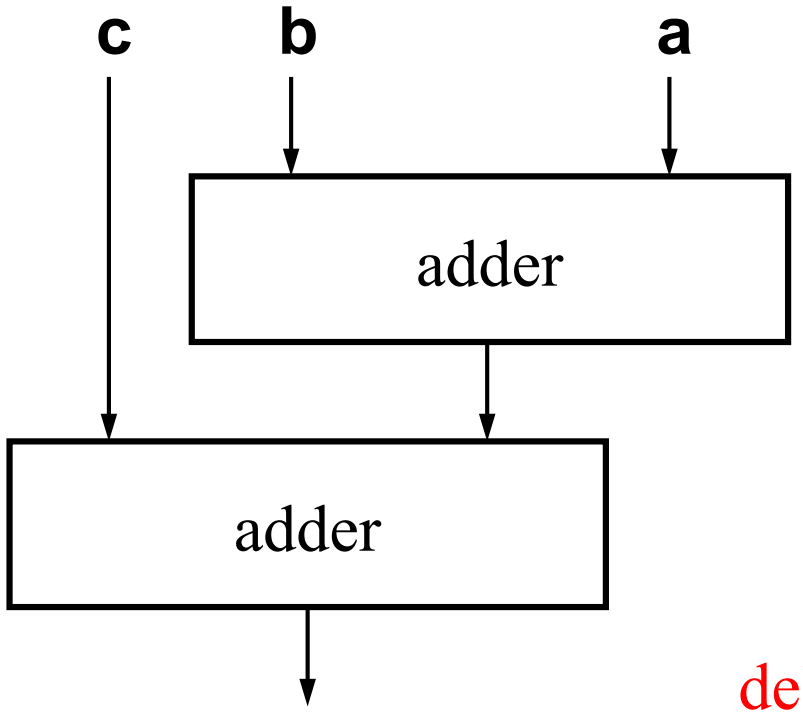
Adding two natural numbers (summary)

	Area	Delay
Ripple Carry (RCA)	n	n
Carry Select (CSLA)	$n^{log(3)}$	*log* (n)
Carry Lookahead (CLA)	n *log* (n)	*log* (n)
Pipeline Adder	n^2	Cste (1 cycle)
Magic Adder	n	Cste

When there is no door to escape break the wall

Adders

Adding three natural numbers



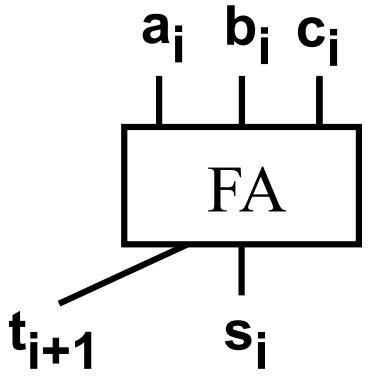
delay

Adders

Adding three natural numbers

$s_i = a_i \oplus b_i \oplus c_i$ $\quad$ $c_{i+1} = a_i.b_i + a_i.c_i + b_i.c_i$

the expressions are symmetrical in regard of **a, b** and **c**

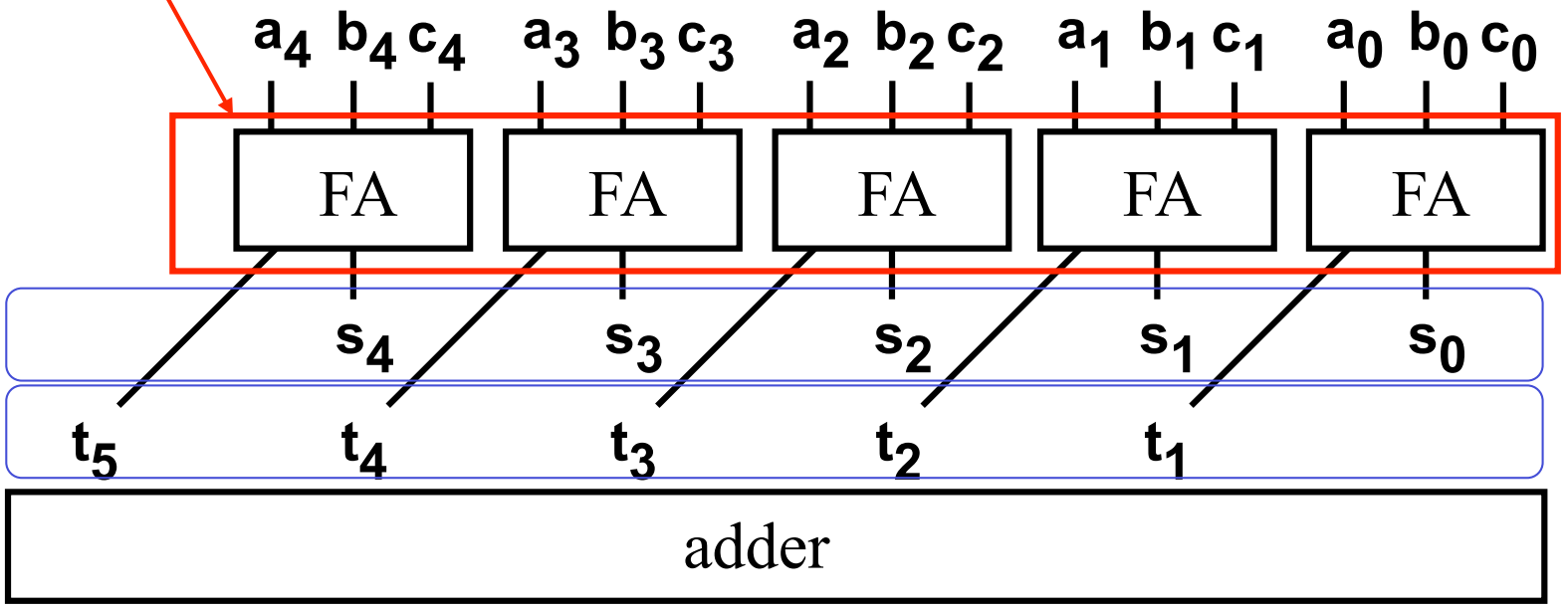


A full adder creates 2 numbers from 3

Adders

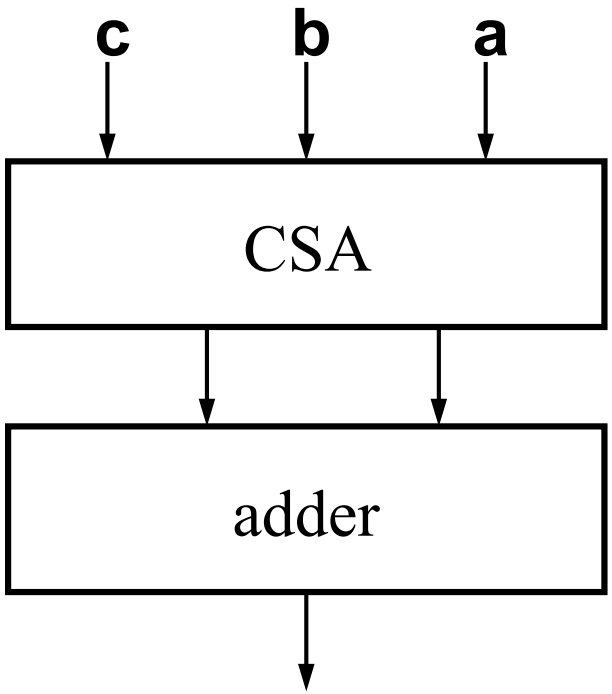
Adding three natural numbers

Carry Save Adder (CSA)



Adders

Adding three natural numbers



Delay = cste

Area $\propto$ n

Adders

Adding two natural numbers

Change the representation of numbers

Given a natural number **a** : **a** is coded using 2n bits

a = a0 + a1 Redundant Binary Code

Example : the number 5 can be coded on 4 bits as

0000 + 0101
0001 + 0100
0010 + 0011

Adders

Adding two natural numbers

Changing the representation of numbers

a = a0 + a1

b = b0 + b1

Adding **a** and **b** in Redundant Binary Code is finding **c**

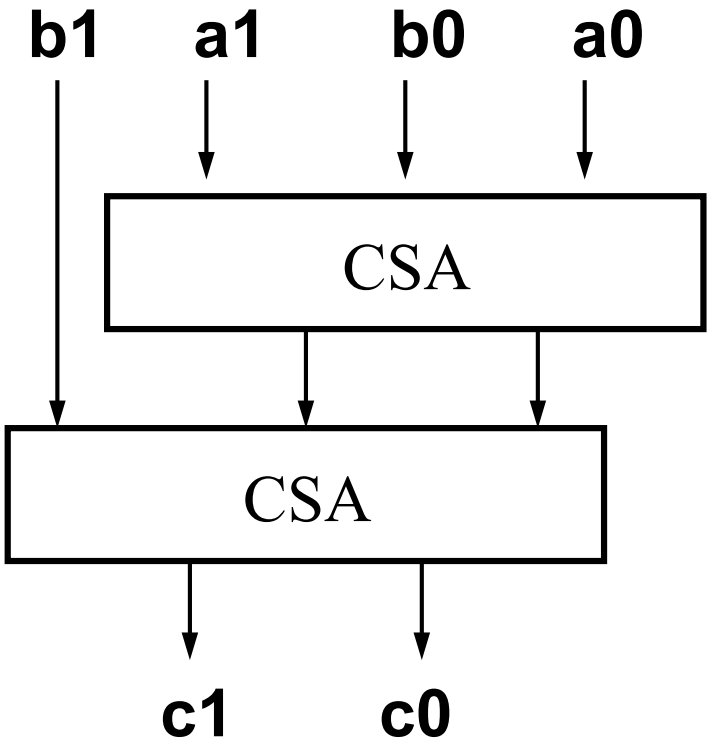
c = c0 + c1 such as

c0 + c1 = a0 + a1 + b0 + b1

Adding 4 numbers to generate 2

Adders

Adding two natural numbers



Delay = cste

Area $\propto$ n