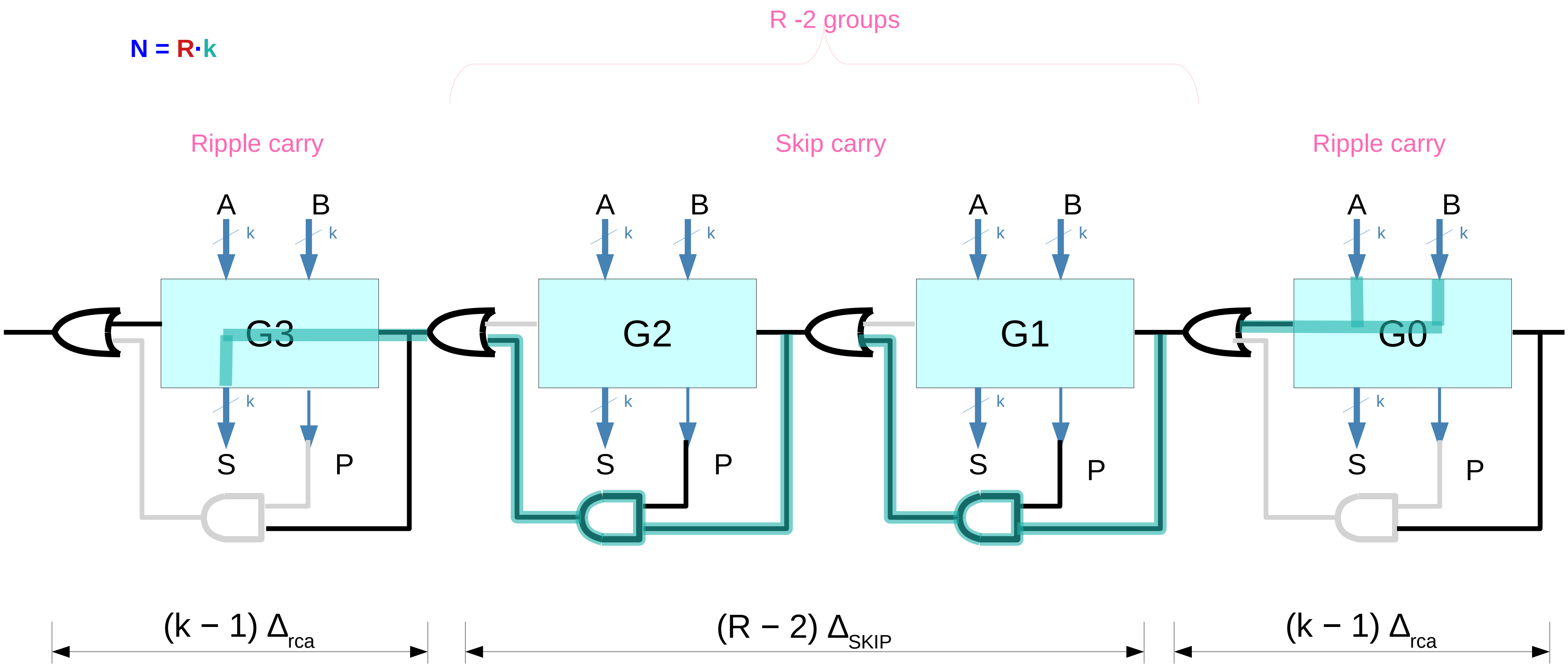
Variable Block Adder (1A)

-
-

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Carry Skip Adder



Any kill or generate condition results in divided (broken) critical paths

All FA's in R-2 groups must have the propagate condition

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Carry Skip Adder

The maximal delay Δ of a Carry Skip Adder is encountered when carry is generated in the least-significant bit position,

- rippling through *k-1* bit positions,
- skipping over *R-2 = N/k-2* groups in the middle,
- rippling to the *k-1* bits of most significant group and
- being assimilated in the *N-th* bit position to produce the sum S_N :

$$\Delta_{CSA} = (k - 1)\,\Delta_{rca} + (R - 2)\,\Delta_{SKIP} + (k - 1)\,\Delta_{rca}$$

$$= 2\,(k - 1)\,\Delta_{rca} + (R - 2)\,\Delta_{SKIP}$$

$$= 2\,(k - 1)\,\Delta_{rca} + (N/k - 2)\,\Delta_{SKIP}$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

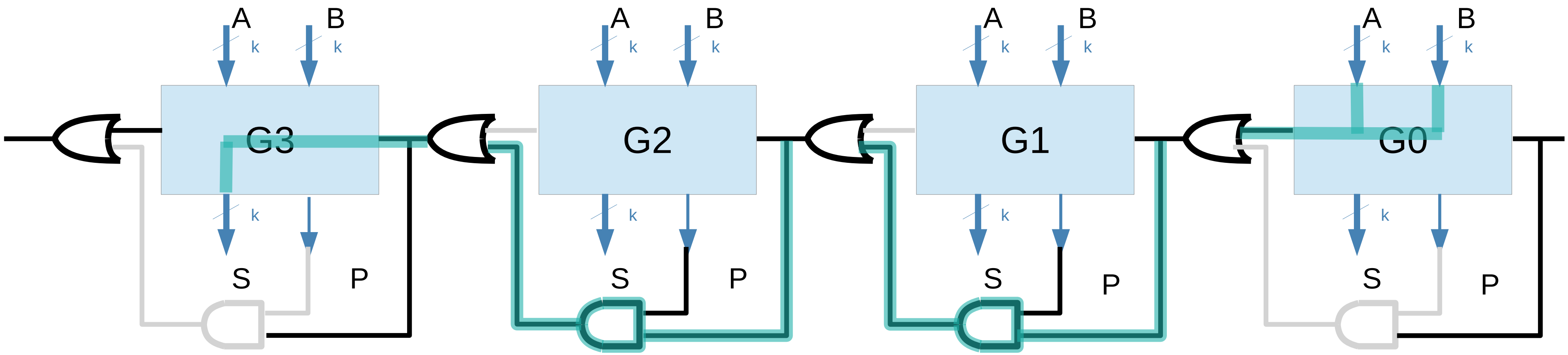
Carry Skip Adder

$$\Delta_{CSA} = (k-1)\,\Delta_{rca} + (R-2)\,\Delta_{SKIP} + (k-1)\,\Delta_{rca}$$
$$= 2\,(k-1)\,\Delta_{rca} + (R-2)\,\Delta_{SKIP}$$
$$= 2\,(k-1)\,\Delta_{rca} + (N/k-2)\,\Delta_{SKIP}$$

Carry Skip Adder is faster than RCA at the expense of a few relatively simple modifications.

N = R·k

The delay is still linearly dependent on the size of the adder N, however this linear dependence is reduced by a factor of *1/k*



Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

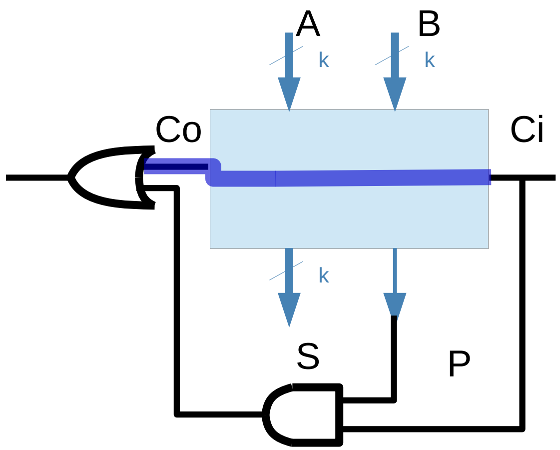
Variable Block

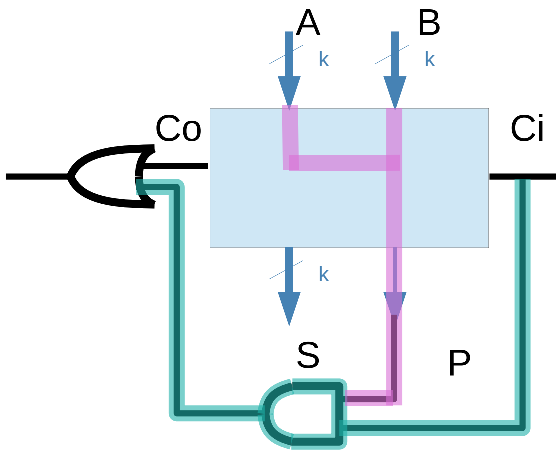
however, unlike the carry select structure,
the variable block adder must also worry about
the delay from the Cin input
through the block's ripple chain

Thus, after the carry chain passes
the midpoint of the logic,
the blocks begin decreasing in length.

This balances the path delays in the system
and improves performance

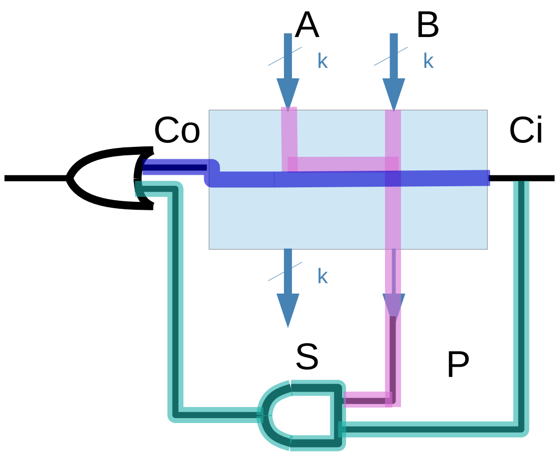
The division of the overall structure into blocks depends on
the details of the logic structure and
the length of the entire computation



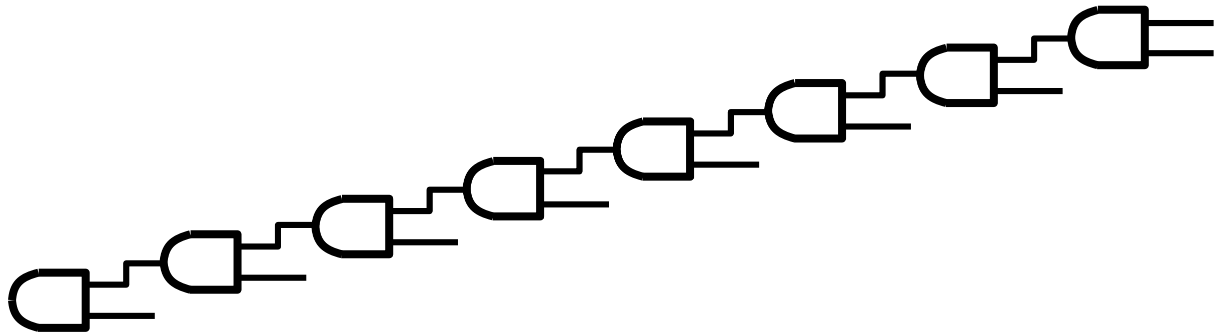


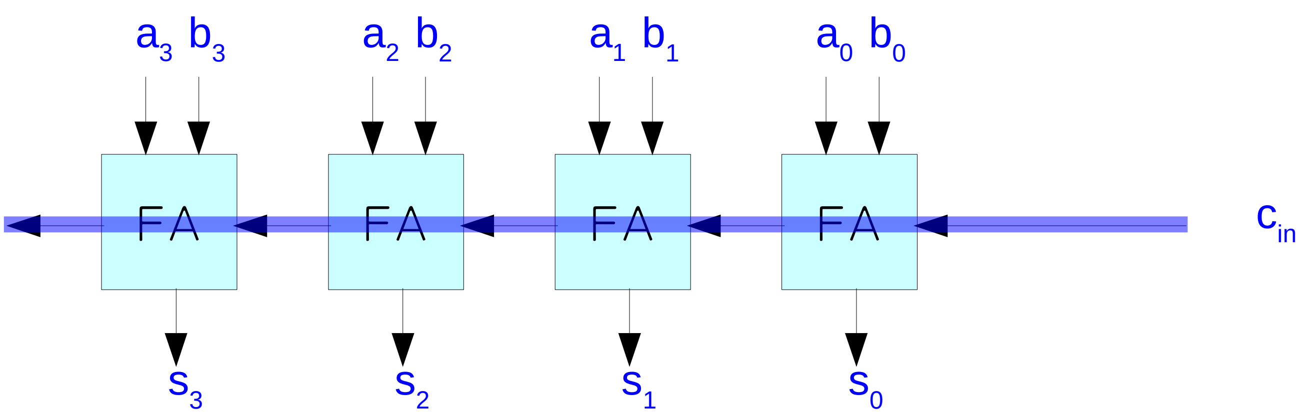
https://en.wikipedia.org/wiki/Carry-lookahead_adder

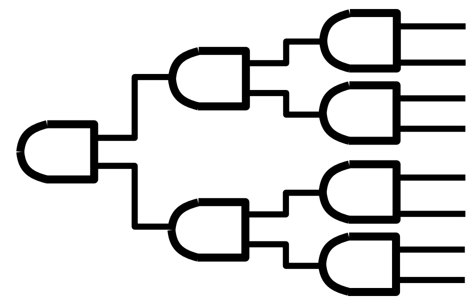
Carry Skip Adder



$P = p_i \bullet p_{i+1} \bullet p_{i+2} \bullet \ldots\ldots \bullet p_{i+k-1}$

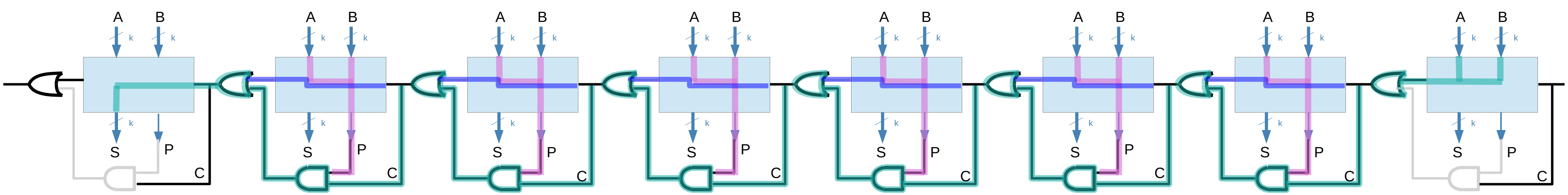






Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

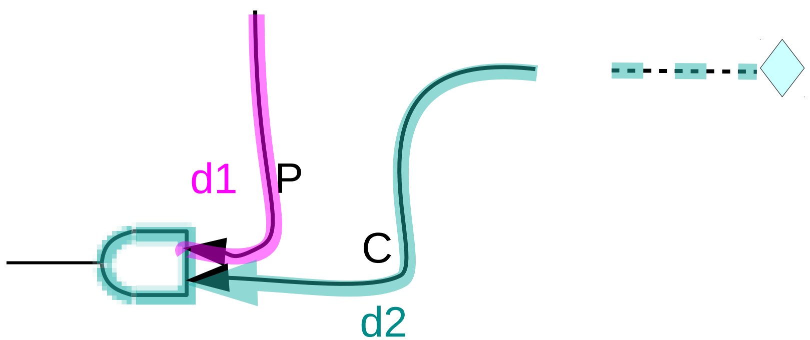
Carry Skip Adder



Delay d1 from A, B to P – parallel, the <u>same</u> delay in each group

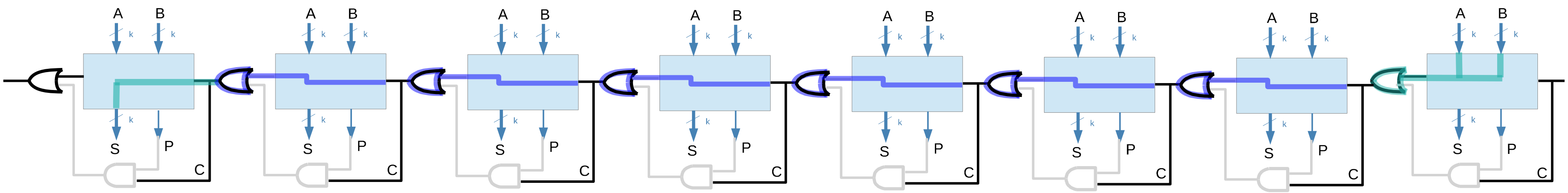
Delay d2 from A, B to C – serial, the <u>accumulated</u> delay from lsb

Delay d3 from A, B, Ci to Co – ripple carry delay

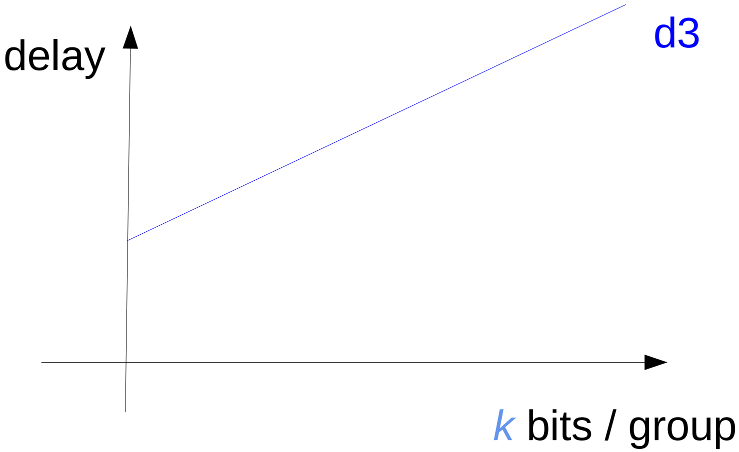


Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Carry Skip Adder

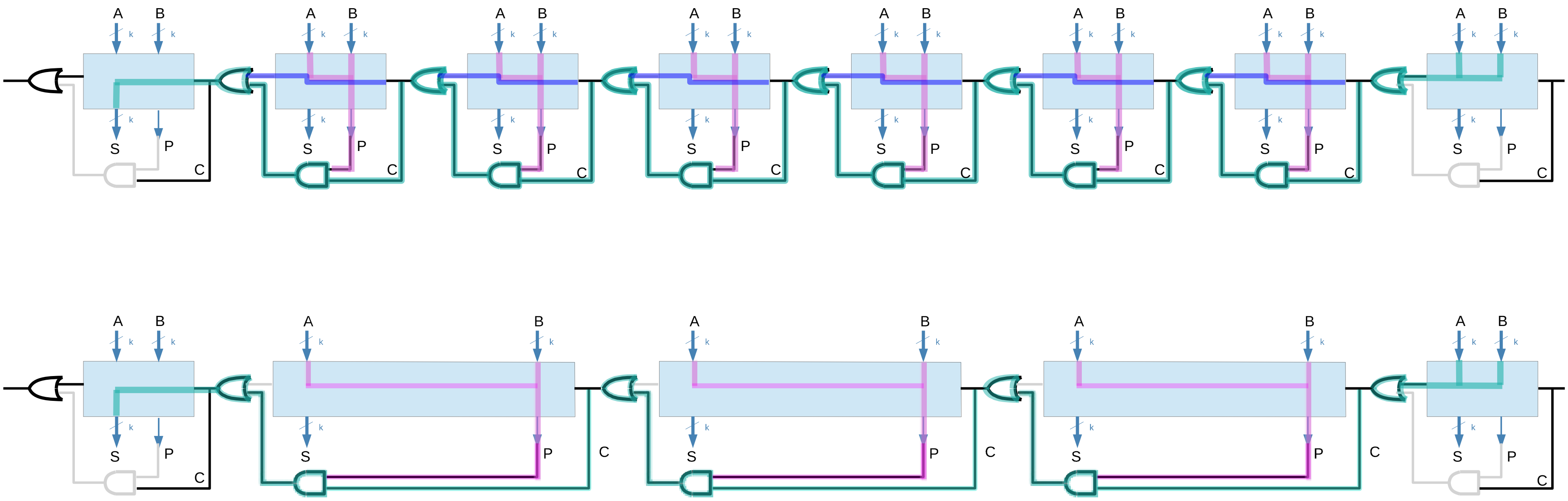


Delay d3 from A, B, Ci to Co – ripple carry delay



Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

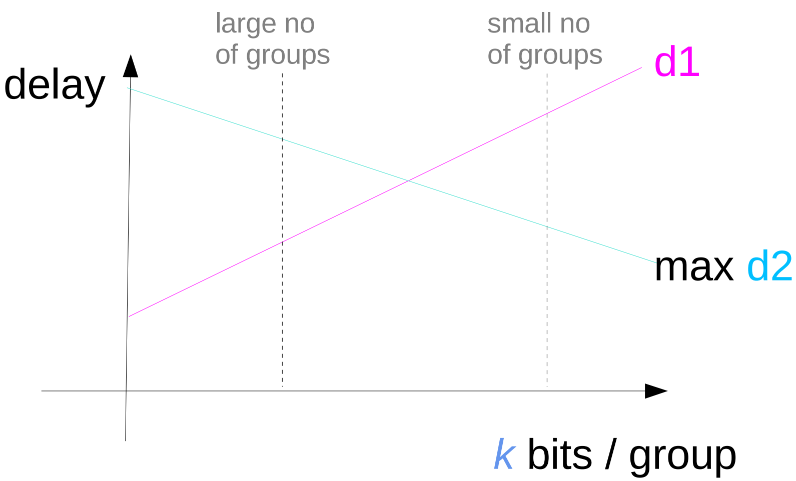
Carry Skip Adder



N = R·k

$$d1 \propto k$$

$$d2 \propto R \left(= \frac{N}{k} \right)$$



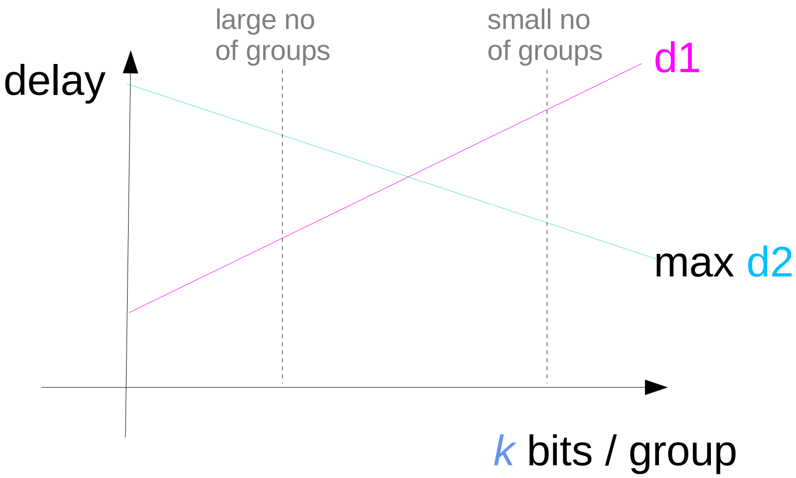
Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

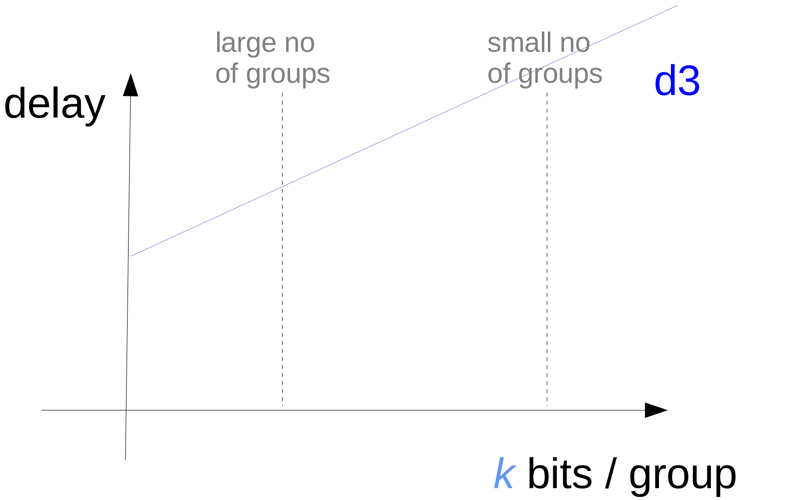
Carry Skip Adder

N = R·k

$$d1 \propto k$$

$$d2 \propto R \left(= \frac{N}{k} \right)$$





Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Variable Block

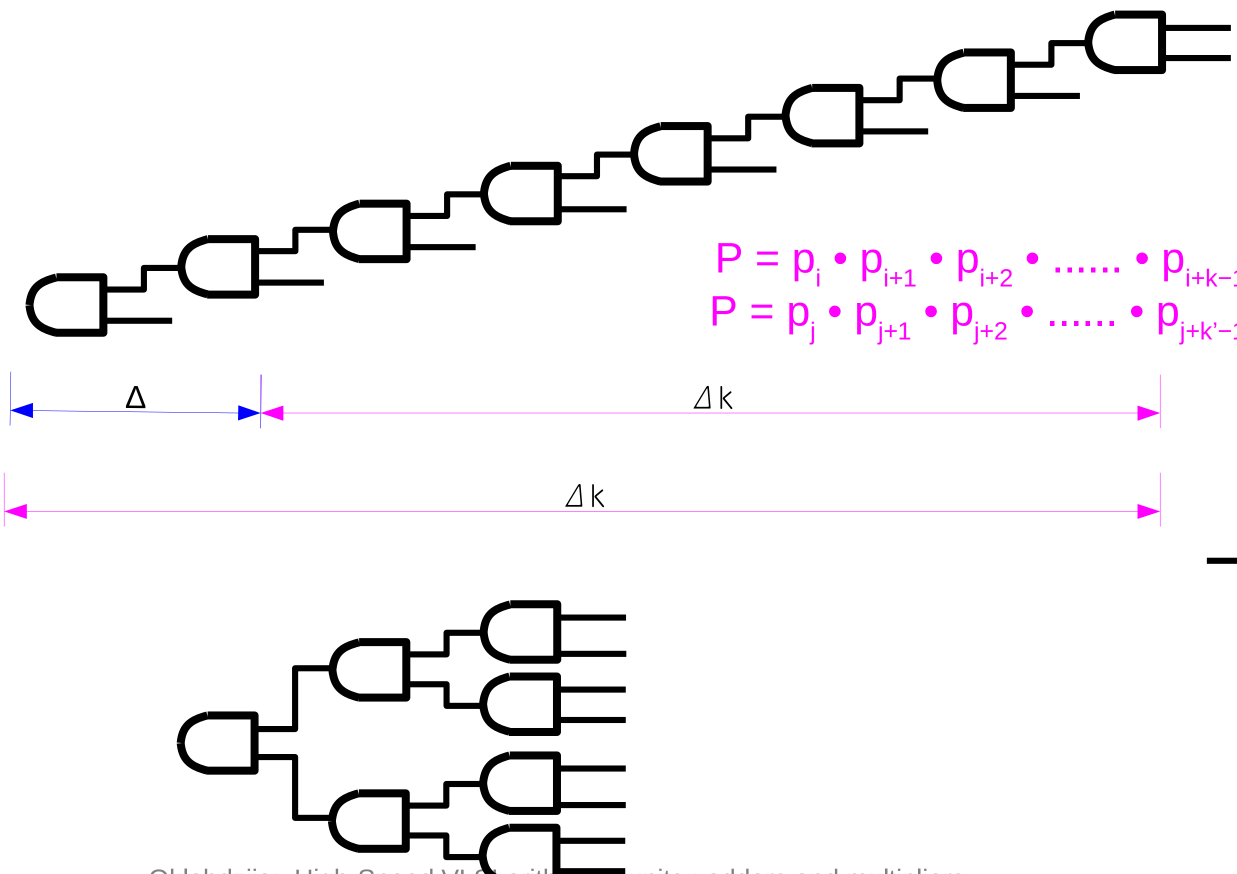
the organization of the blocks in the variable block carry structure
bears some similarity to the carry select structure

the early stages of the structure grow in length,
with short blocks for the low order bits,
building in length further in the chain
in order to equalize the arrival time
of the carry from the block
with that of the previous block

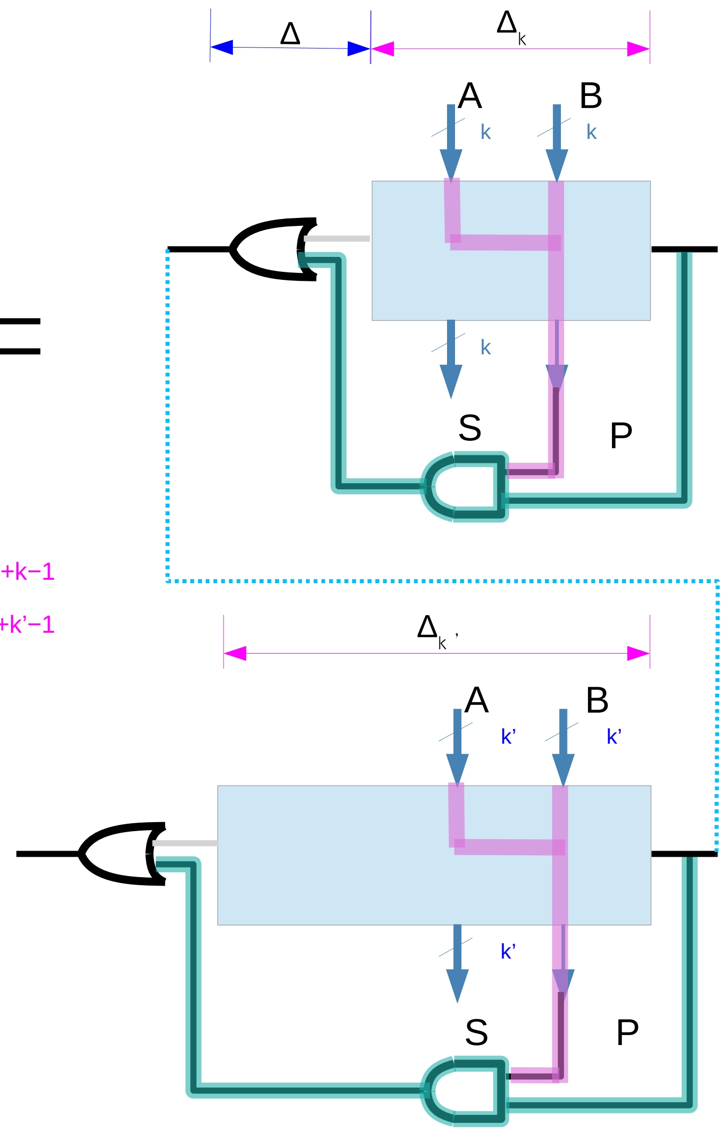
https://en.wikipedia.org/wiki/Carry-lookahead_adder

Carry Skip Adder

to equalize the arrival time
of the carry from the block
with that of the previous block

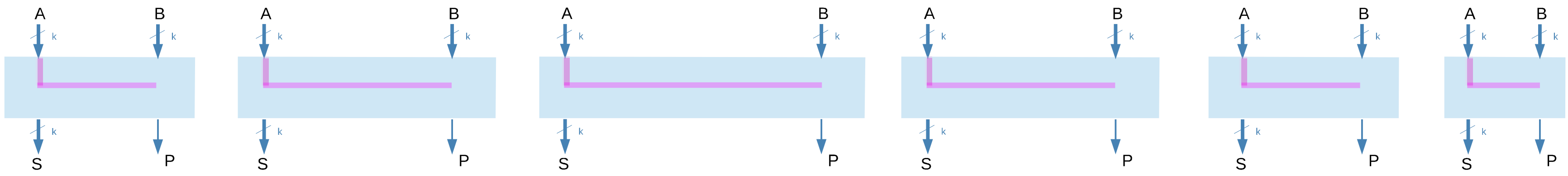


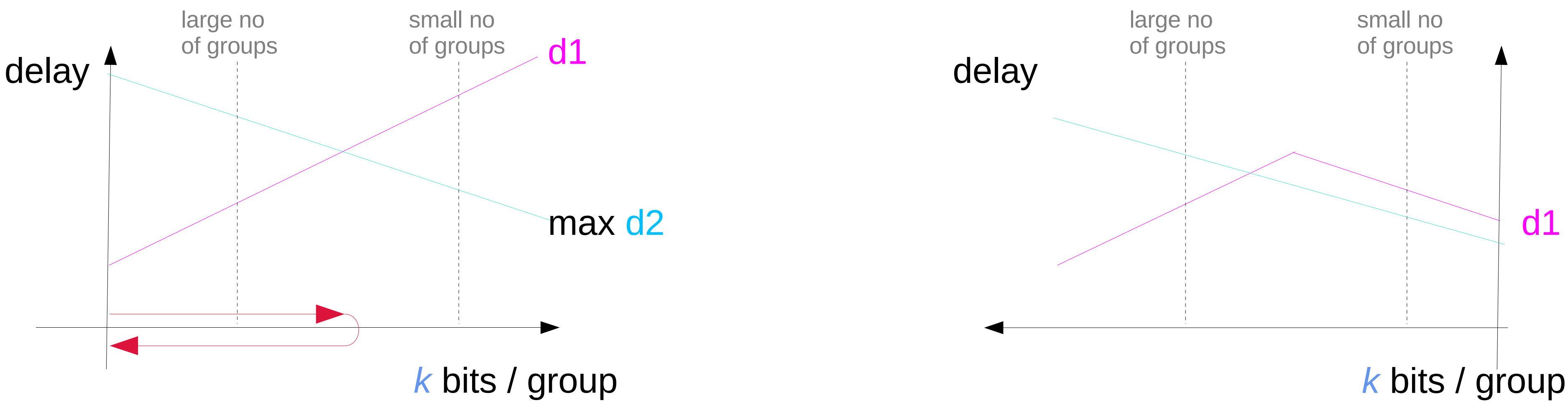
$$P = p_i \bullet p_{i+1} \bullet p_{i+2} \bullet \ldots\ldots \bullet p_{i+k-1}$$
$$P = p_j \bullet p_{j+1} \bullet p_{j+2} \bullet \ldots\ldots \bullet p_{j+k'-1}$$



Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

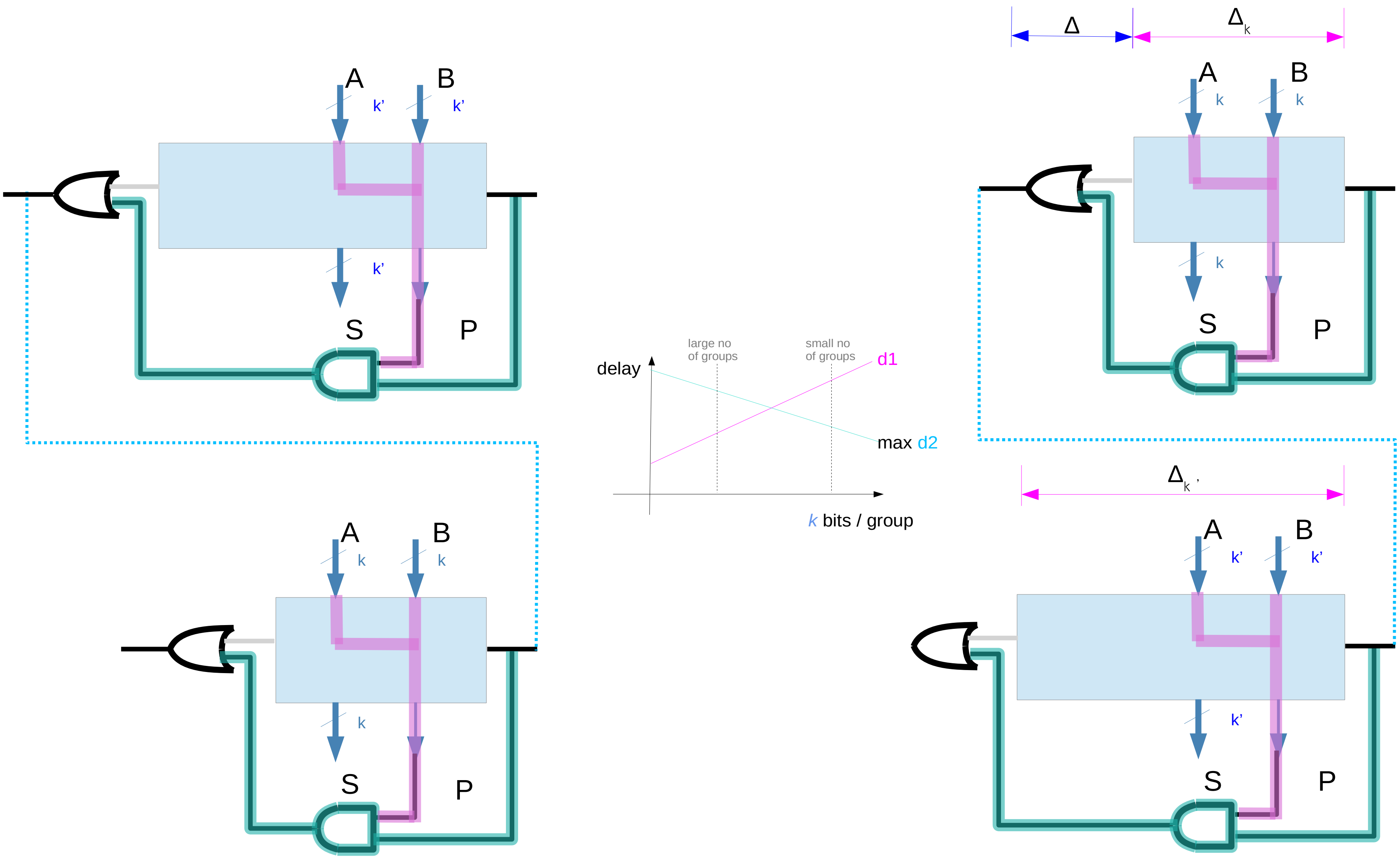
Carry Skip Adder





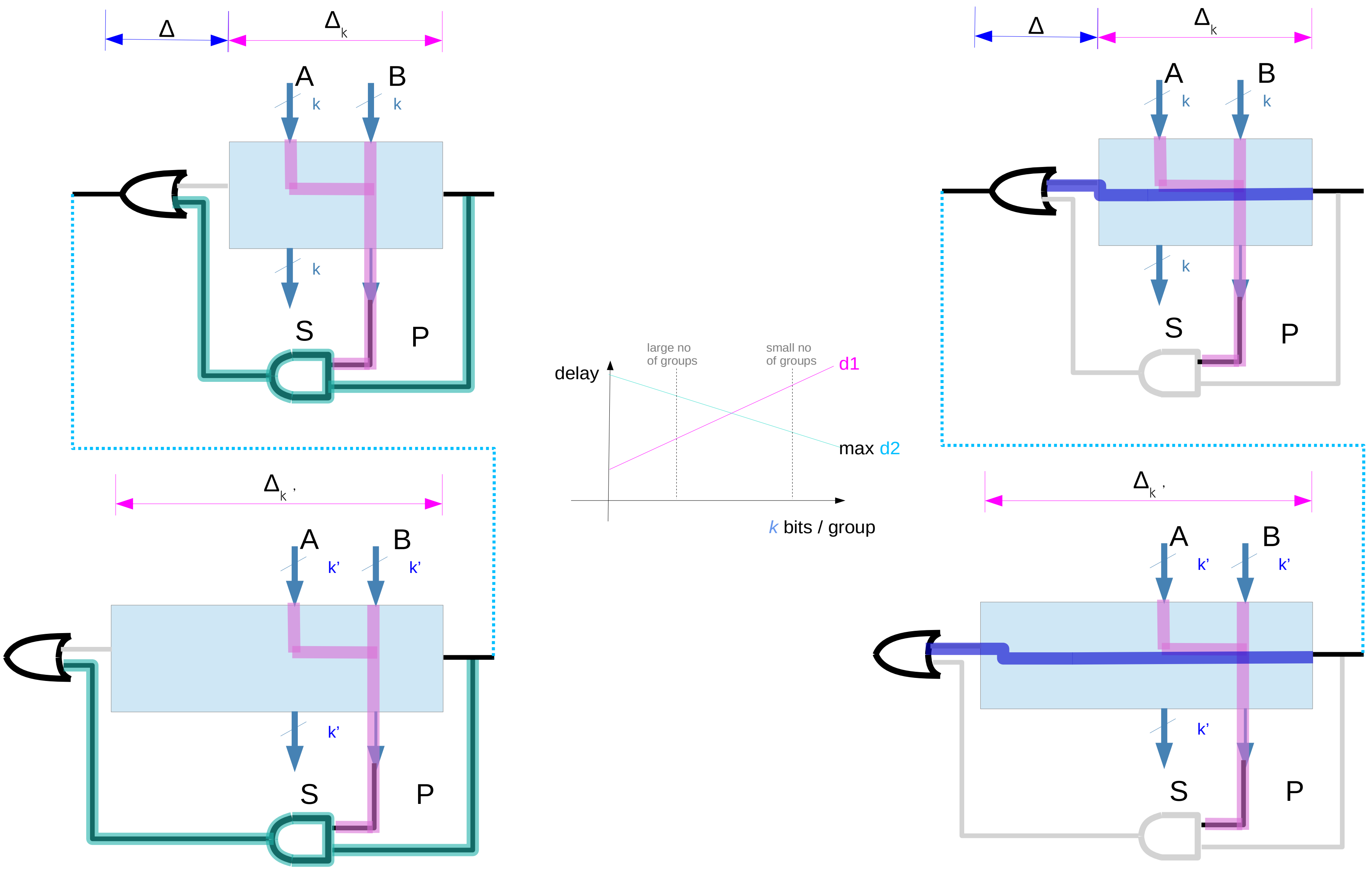
Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Carry Skip Adder



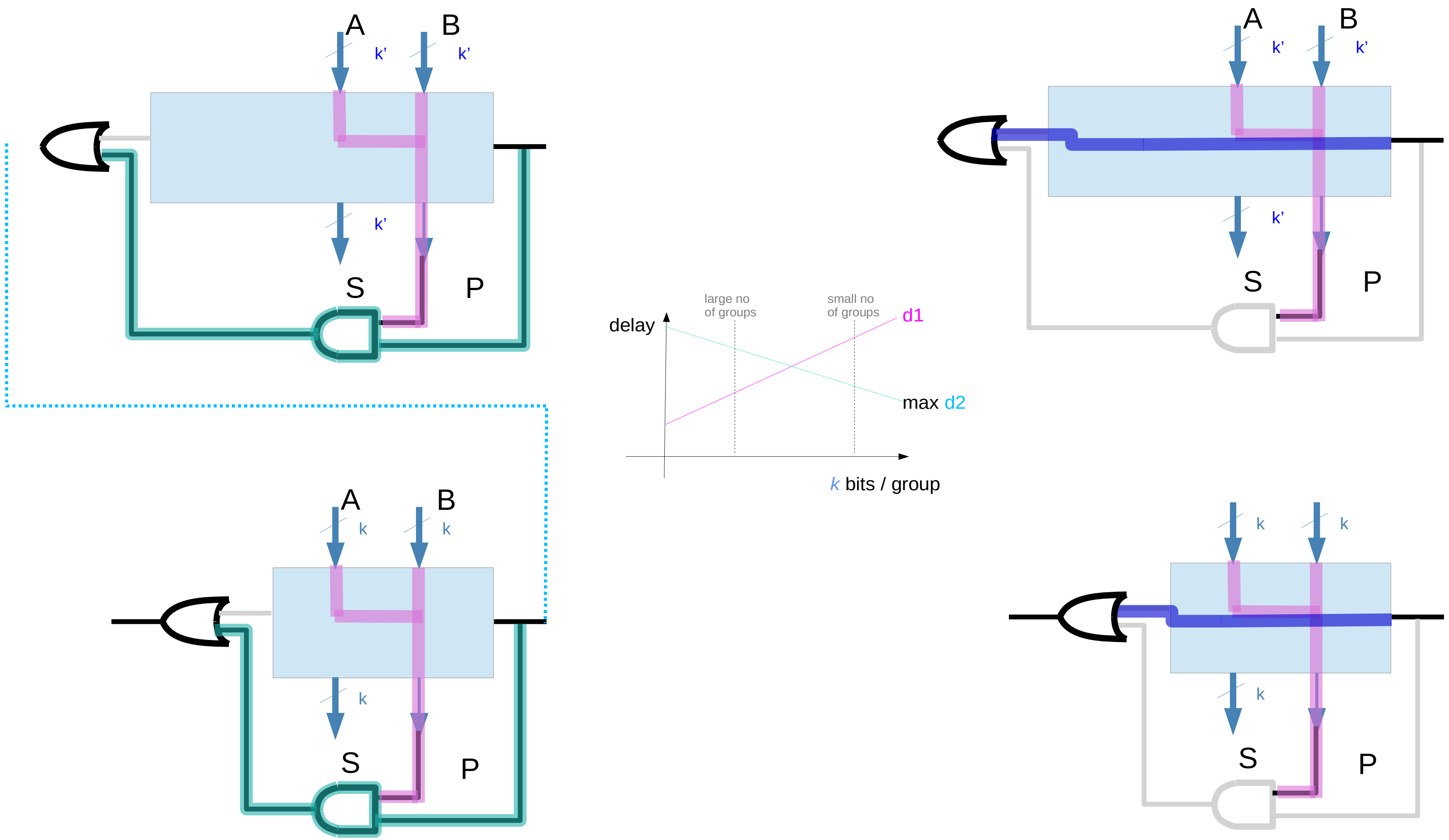
Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Carry Skip Adder



Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Carry Skip Adder



Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

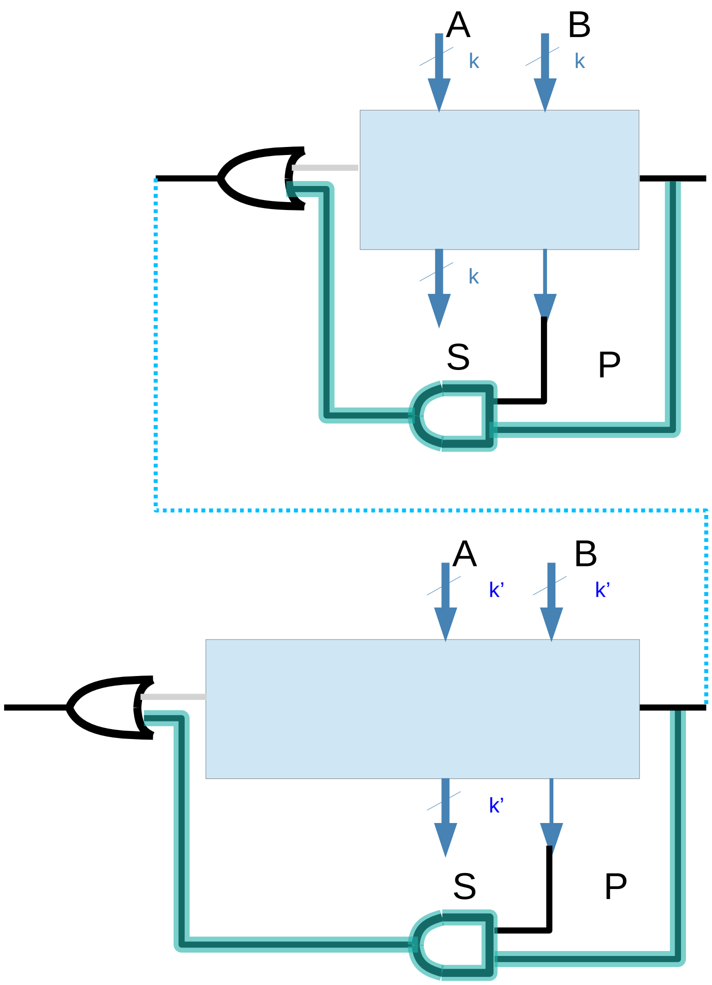
Carry Skip Adder

All carries propagated more quickly
by varying the block sizes
Accordingly the initial blocks of the adder
are made smaller, so as to quickly detect
carry generates that must be propagated

The middle blocks are made larger
because they are not the problem case,

And then the most significant blocks
are made smaller so that the late arriving
carry inputs can be processed quickly

https://en.wikipedia.org/wikik/Carry-skip_adder



Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

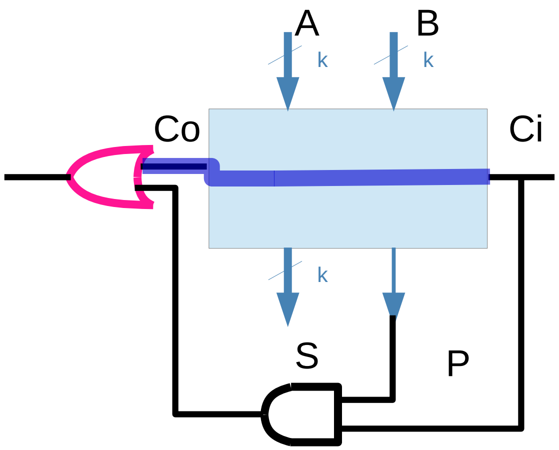
Carry Skip Adder

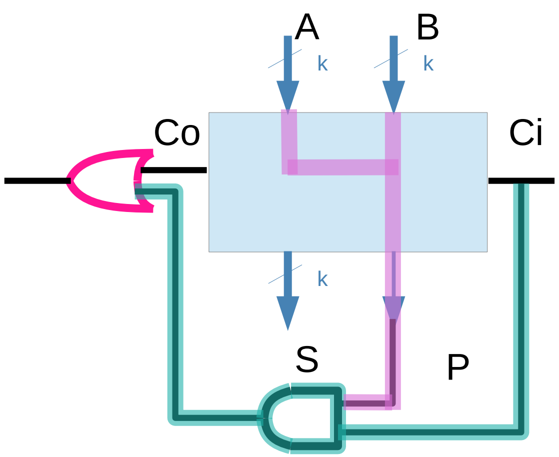
The longest path length through the carry skip block
is potentially much shorter than the path
from carry-in to carry-out
through a ripple carry block.

However, the carry skip block has a slightly longer path
from the least significant <g,t> input to carry output

Hence, this adder will only be faster
when skipping groups makes up
for the extra gate overhead accumulated
by going from generate/transfer to carry-out

The maximum path length through a one block wide
carry skip adder is the same as though a ripple carry adder,
since the bottom block in a skip adder is a ripple carry





Binary Adders, T W Lynch, Master Thesis, University of Texas at Austin 1996

Two separate ripple carry adders

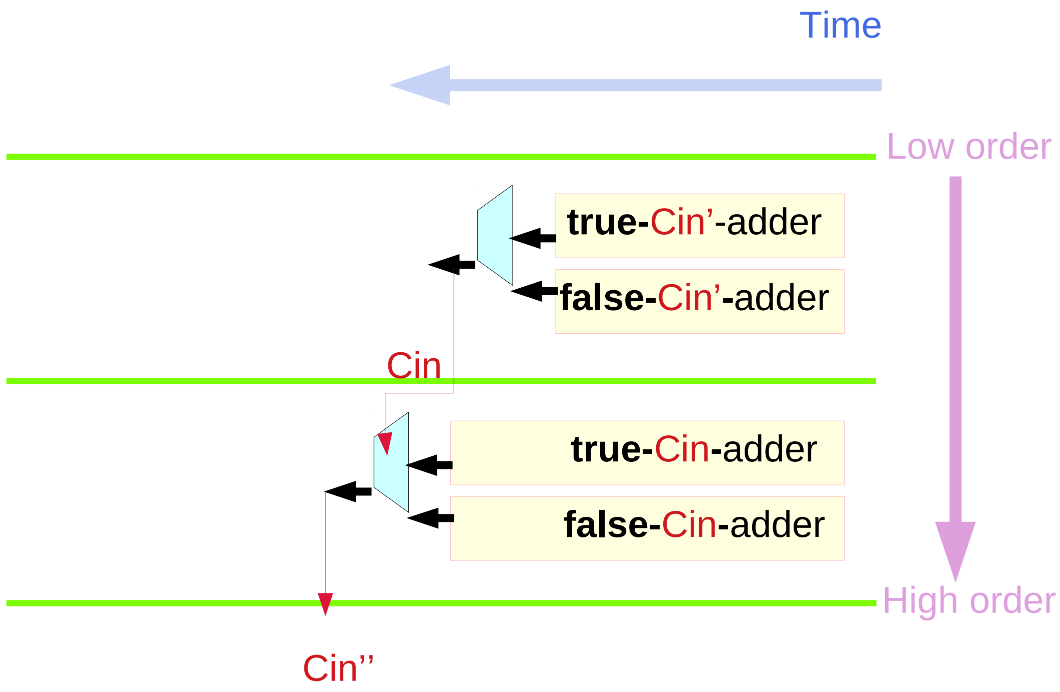
Cin signal is used to determine
which adder's outputs should be used

if the Cin signal is **true**,
the output (carries) are selected from
the **true-Cin-adder**

if the Cin signal is **false**,
the output (carries) are selected from
the **false-Cin-adder**

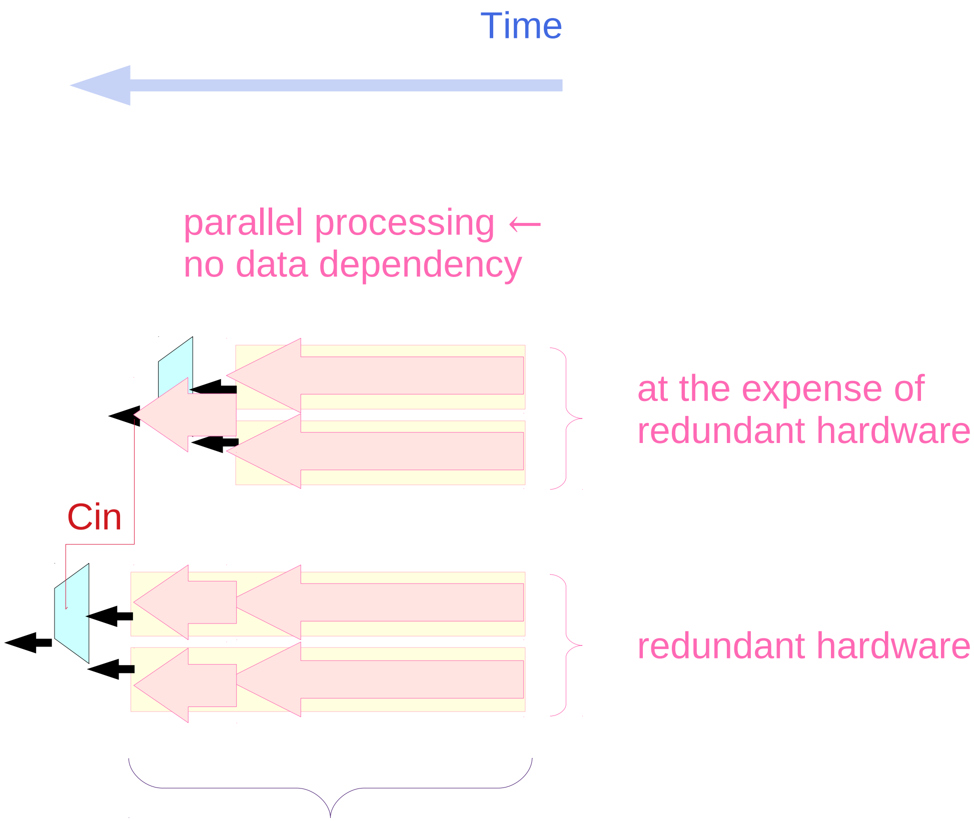
redundant hardware removes
Cin data dependency

first start redundant computation
later select the correct one

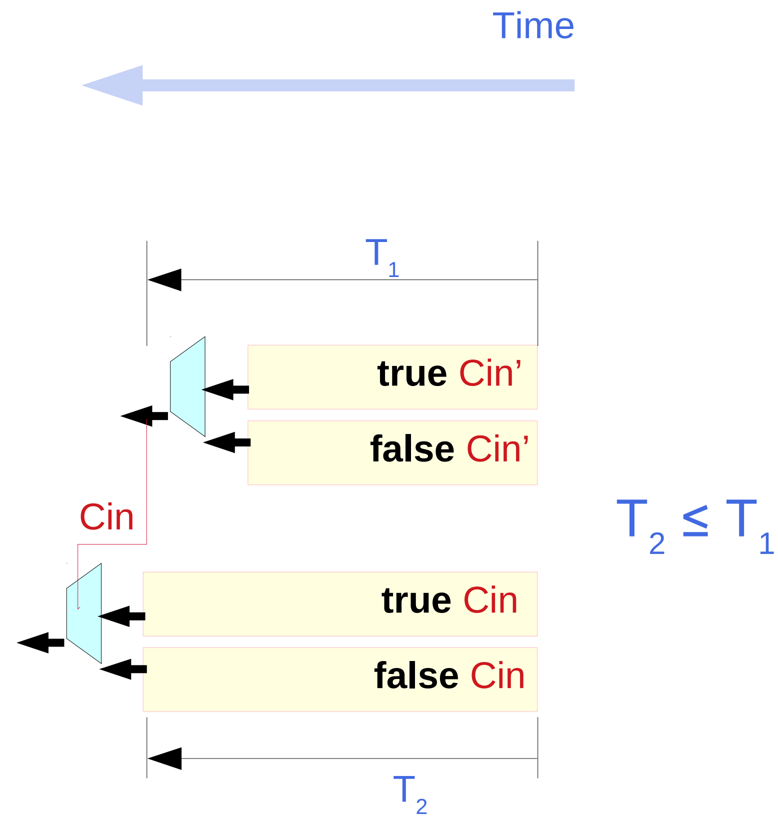


High Performance Carry Chains for FPGAs, S. Hauck, M. M. Hosler, T. W. Fry

Timing in broken carry chains



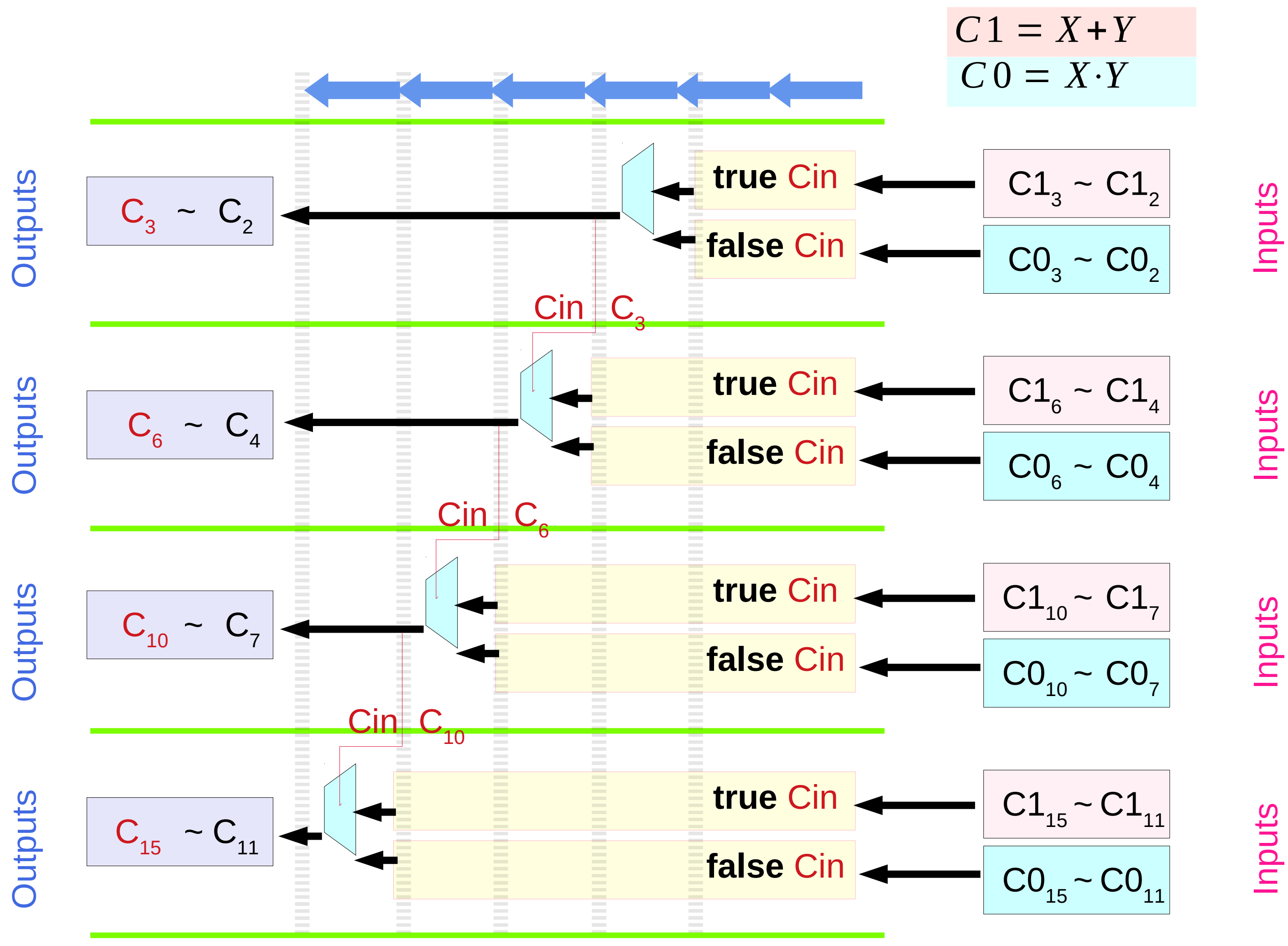
These computations can take place before the completion of the previous columns, since they do not depend on the actual value of the Cin signal



the length of the adders and the breakpoint are carefully chosen such that the **adders** finish computations just as their Cin become available

High Performance Carry Chains for FPGAs, S. Hauck, M. M. Hosler, T. W. Fry

Carry Select Fast Carry Logic



High Performance Carry Chains for FPGAs, S. Hauck, M. M. Hosler, T. W. Fry

Variable Block

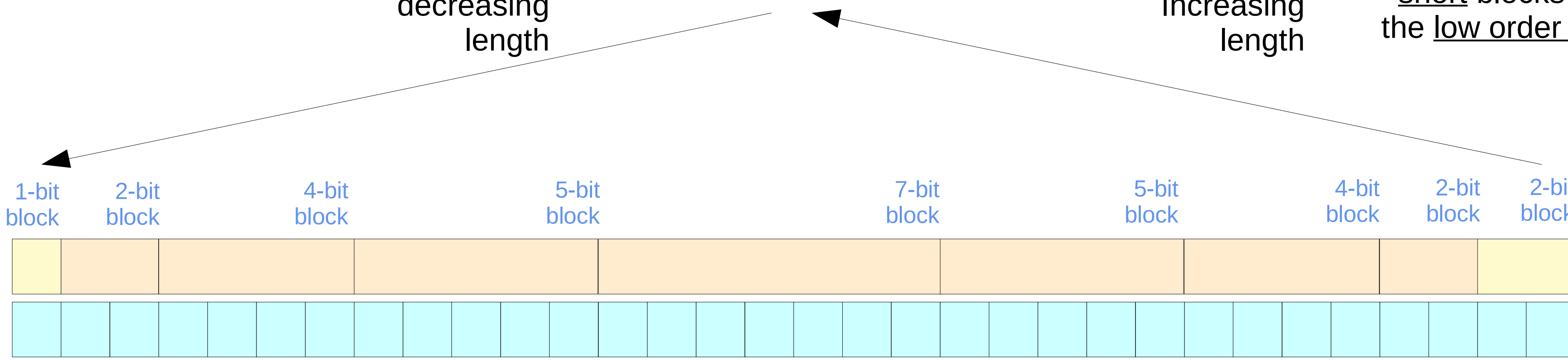
d1 > d2 ➡ d1 < d2

d1 < d2

decreasing length

Increasing length

short blocks for the low order bits

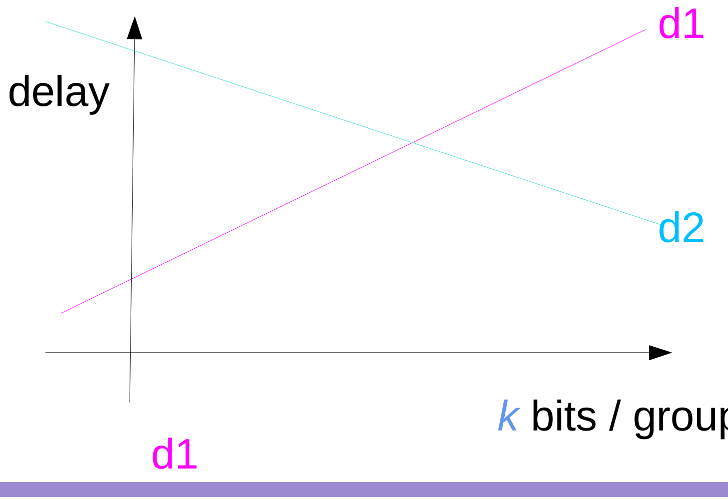


Simple Carry Chain

Variable Block

Simple Carry Chain

the delay from the Cin input through the block's ripple chain



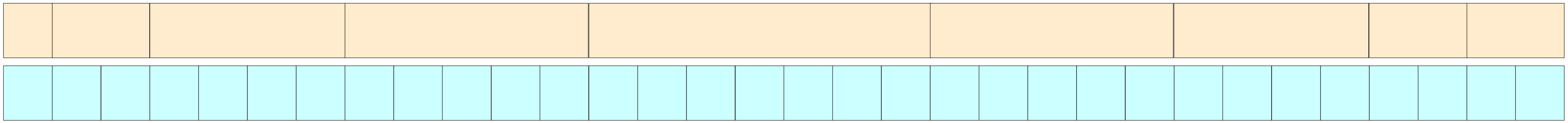
https://en.wikipedia.org/wiki/Carry-lookahead_adder

Variable Block

We use a block length <u>from low order</u> to <u>high order cells</u>
of 2, 2, 4, 5, 7, 5, 4, 2, 1 for a normal 32 bit structure

8+17+7

The first and last block in each adder is
a simple ripple carry chain,
while all other blocks use
the variable block structure.



https://en.wikipedia.org/wiki/Carry-lookahead_adder

Variable Block

Delay values of the variable block carry chain relative to other carry chains

The idea behind Variable Block Adder (VBA) is to minimize the longest critical path in the carry chain of a Carry Skip Adder, while allowing the groups to take different sizes.

Such an optimization in general does not result in an increased complexity as compared to the Carry Skip Adder

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Variable Block

The first and last blocks are smaller,
and the intermediate blocks are larger.

That compensates for the critical paths originating from the ends by shortening the length of the path used for the carry signal to ripple in the end groups, allowing carry to skip over larger groups in the middle.

There are two important consequences of this optimization:

(a) the total delay is reduced as compared to a Carry Skip Adder

(b) the delay dependency is not a linear function of the adder size N as in a Carry Skip Adder.

This dependency follows a square root function of N instead

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Variable Block

For an optimized VBA, it is possible to obtain
a closed form solution expressing this delay dependency

It is also possible to extend this approach
to multi-levels of carry skip as done in
a determination of the optimal sizes of the blocks
on the first and higher levels of skip blocks
is a **linear programming problem**,
which does not yield a closed form solution.

Such types of problems are solved
with the use of **dynamic programming** techniques.

The speed of such a mult-level VBA adder
surpasses single-level VBA and
that of fixed group Carry-Lookahead Adder (CLA).

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Variable Block

For an optimized VBA, it is possible to obtain a closed form solution expressing this delay dependency which is given as:
where: c1 , c2 , c3 are constants.

$$\Delta_{VBA} = c_1 + \sqrt{c_2 N + c_3}$$

It is also possible to extend this approach to multiple levels of carry skip as done.

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Variable Block

(1) the speed of the logic gates
used for CMOS implementation
depends on the output load: fan-out,
as well as the number of inputs: fan-in.

(2) CLA implementation is characterized
with a large fan-in which limits
the available size of the groups.

On the other hand VBA implementation is simple.

Thus, it seems that CLA has passed the point of diminishing returns as far as an efficient implementation is concerned.

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Variable Block

This example also points to the importance of modeling
and incorporating appropriate technology parameters into the algorithm.

Most of the computer arithmetic algorithms developed in the past
use a simple constant gate delay model.

(2.) a fixed-group CLA is not the best way to build an adder.

It is a sub-optimal structure which
after being optimized for speed,
consists of groups that are different in size
yielding a largely irregular structure

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Variable Block

There are other advantages of VBA adder that are direct result of its simplicity and efficient optimization of the critical path.

Those advantages are exhibited in the lower area and power consumption while retaining its speed.

Thus, VBA has the lowest energy-delay product as compared to the other adders in its class.

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

Oklobdzija addition VLSI

On implementing addition in VLSI technology

Delay dependency : Fan-out, Fan-in,
Delay estimates :

D_NAND = 1 +0.3Fo +0.5(Fi-2)
D_NOR = 1 +0.5Fo +0.5(Fi-2)
D_NAND = 0.7+0.3Fo

t denote the time required for a carry signal to ripple across a bit
T denote the time required for the signal to skip over a group of bits
m denotes the optimal number of groups for an n-bit carry chain
m is the smallest positive integer satisfying

$$n \leq m + \frac{1}{2} m T + \frac{1}{4} m^2 T + \left(1 - (-1)^m\right) \frac{1}{8} T$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

n : the number of bits in a carry skip adder
m : the number of groups into which the bits are divided
$x_1, \ldots, x_m$: the sizes of the groups beginning with the most significant bit

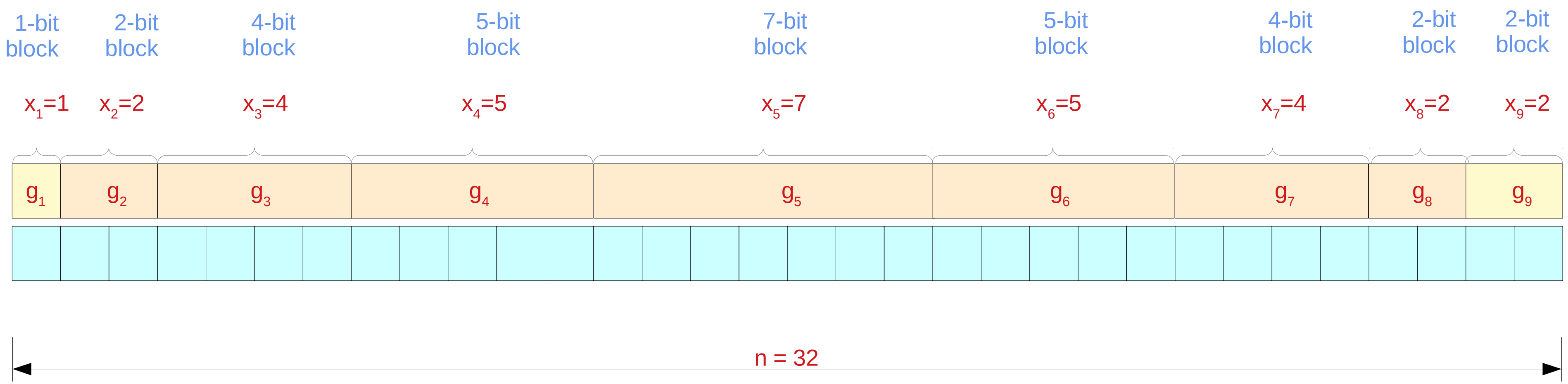
T : the time required for a carry signal to skip over a group of bits

To be precise we should write $T = T(x)$ to indicate that
T depends on the size x of the group over which the carry is skipped
However, T changes very slowly with x over the range of group sizes
So we assume that T is constant

For a given n, the following three step procedure gives
An optimal way of dividing an n bit adder into groups of bits

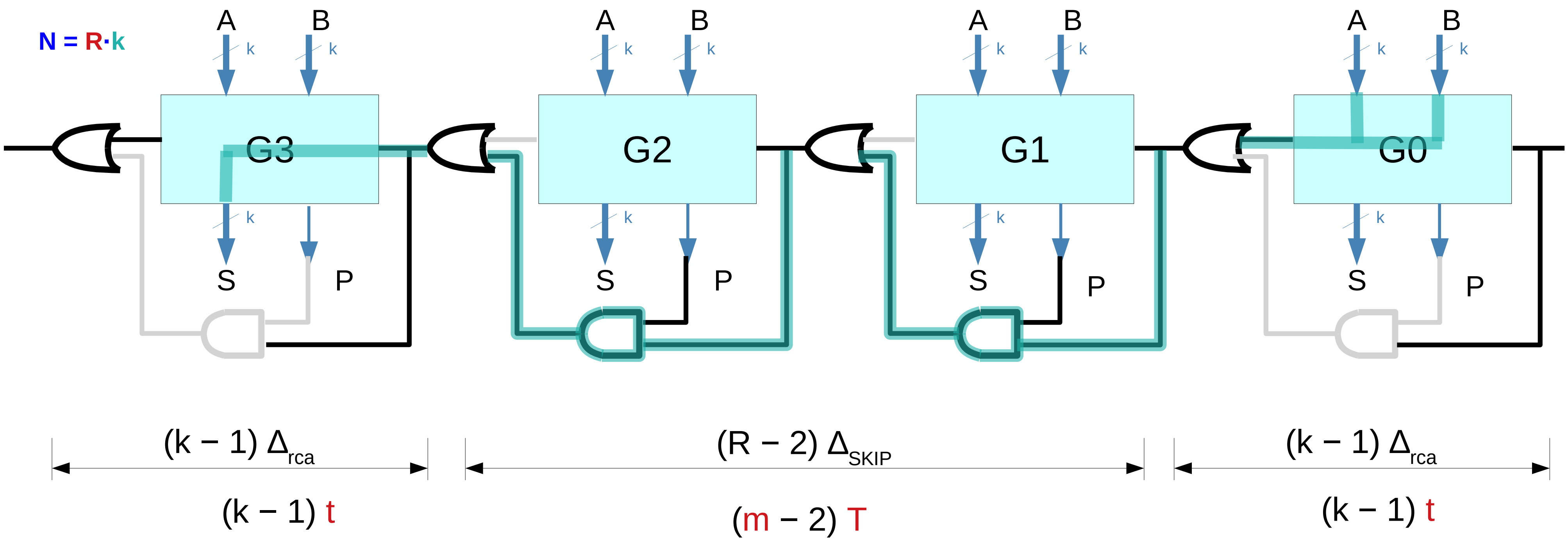
Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Variable Block



- total n = 32 bits
- m = 9 groups
- i-th group has x_i bits (size)
- constant skip delay $T = T(x_i)$

Carry Skip Adder



t denote the time required for a carry signal to ripple across a bit
T denote the time required for the signal to skip over a group of bits
m denotes the optimal number of groups for an n-bit carry chain

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Procedure

- total n =48 bits
- m =7 groups
- i-th group has x_i bits (size)
- constant skip delay $T = T(x_i) = 3$

(I) Let m be the smallest positive integer such that

$$n \le m + \frac{1}{2} m T + \frac{1}{4} m^2 T + \left(1 - (-1)^m\right) \frac{1}{8} T$$

(II) Let

$$y_i = min\{1 + i T, 1 + (m + 1 - i) T\}, \quad i = 1, \dots, m$$

and construct a histogram whose i-*th* column has height y_i

for example, for T=3, and n=48, we have m=7

(III) It is easily verified that the area of the histogram in (II) is

$$m + \frac{1}{2} m T + \frac{1}{4} m^2 T + \left(1 - (-1)^m\right) \frac{1}{8} T \ge n$$

so these are <u>at least</u> n <u>unit squares</u> in the histogram
starting with the first row, shade in n of the squares, row by row
Let x_i denote the number of shaded squares in column i of the histogram,
$i = 1, \dots, m$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Example 1 - (1)

For a 48 bit adder we have, from Figure

$$x_1 = x_7 = 4, x_2 = x_6 = 7, x_3 = 8, x_4 = x_5 = 9$$

The maximum delay is experienced
by a signal generated in the second bit position
and terminating in the 47th bit position

the delay is $mT = 21$

- total $n = 48$ bits
- $m = 7$ groups
- i-th group has x_i bits (size)
- constant skip delay $T = T(x_i) = 3$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Example 1 - (2)

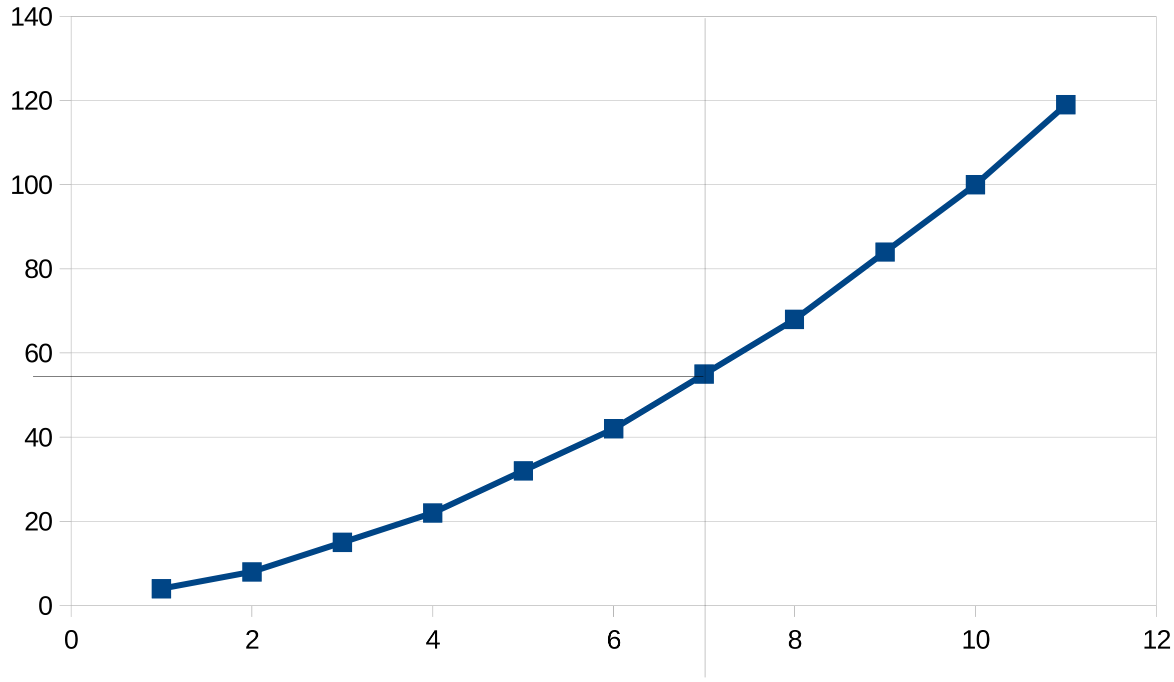
$$n \le m + \frac{1}{2} m T + \frac{1}{4} m^2 T + (1-(-1)^m) \frac{1}{8} T$$

$$48 \le m + \frac{3}{2} m + \frac{3}{4} m^2 + (1-(-1)^m) \frac{3}{8}$$

- total $n = 48$ bits
- $m = 7$ groups
- i-th group has x_i bits (size)
- constant skip delay $T = T(x_i) = 3$

1	4
2	8
3	15
4	22
5	32
6	42
7	55
8	68
9	84
10	100

$48 < 55$

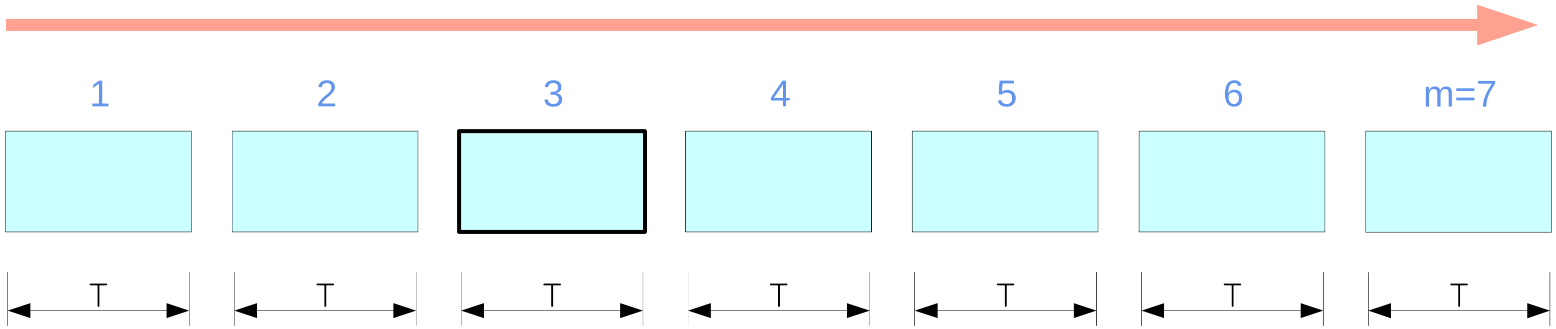


$m = 7$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Example 1 - (3)

$n = 48$
$m = 7$
$T = 3$



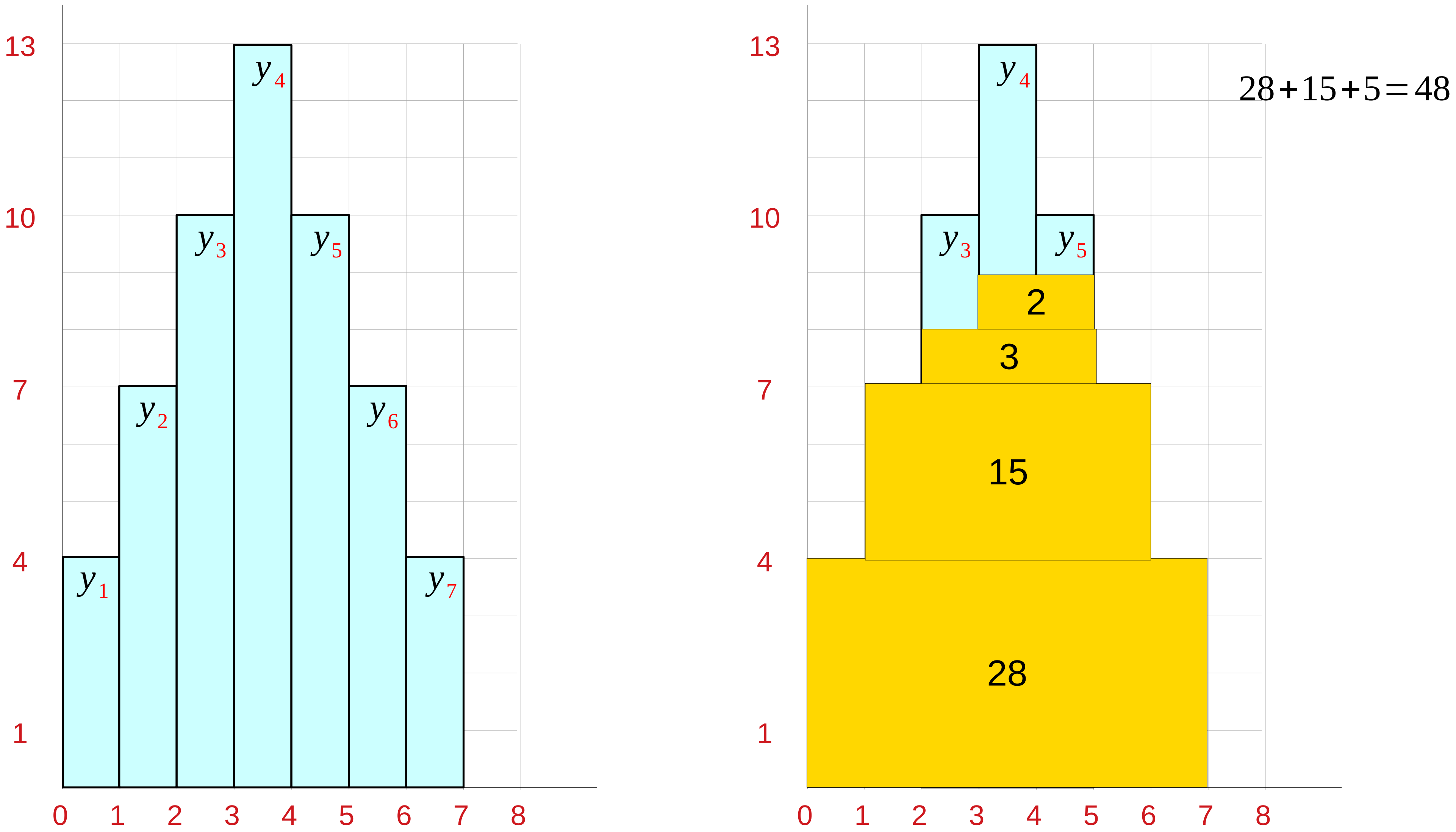
$$y_i = \min\{1+iT, 1+(m+1-i)T\}, \quad i = 1,\ldots,m$$

$$y_1 = \min\{1+1\cdot T, 1+7\cdot T\} = 1+1\cdot T = 4 \qquad 0 \le x_1 \le 1+1\cdot T = 4$$
$$y_2 = \min\{1+2\cdot T, 1+6\cdot T\} = 1+2\cdot T = 7 \qquad 0 \le x_2 \le 1+2\cdot T = 7$$
$$y_3 = \min\{1+3\cdot T, 1+5\cdot T\} = 1+3\cdot T = 10 \qquad 0 \le x_3 \le 1+3\cdot T = 10$$
$$y_4 = \min\{1+4\cdot T, 1+4\cdot T\} = 1+4\cdot T = 13 \qquad 0 \le x_4 \le 1+4\cdot T = 13$$
$$y_5 = \min\{1+5\cdot T, 1+3\cdot T\} = 1+3\cdot T = 10 \qquad 0 \le x_5 \le 1+3\cdot T = 10$$
$$y_6 = \min\{1+6\cdot T, 1+2\cdot T\} = 1+2\cdot T = 7 \qquad 0 \le x_6 \le 1+2\cdot T = 7$$
$$y_7 = \min\{1+7\cdot T, 1+1\cdot T\} = 1+1\cdot T = 4 \qquad 0 \le x_7 \le 1+1\cdot T = 4$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

$$0 \le x_i \le y_i, i = 1,\ldots,m$$

Example 1 - (4)



Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Example 2 - (1)

consider a 54 bit adder

From 2(i), we see that again m=7.

If we shade 54 squares in Figure, we see that

$$x_1 = x_7 = 4, x_2 = x_6 = 7, x_3 = x_5 = 10, x_4 = 12$$

Yields an optimal division of the adder.
Again the maximum delay is mT = 21

- total $n = 54$ bits
- $m = 7$ groups
- i-th group has x_i bits (size)
- constant skip delay $T = T(x_i) = 3$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Example 2 - (2)

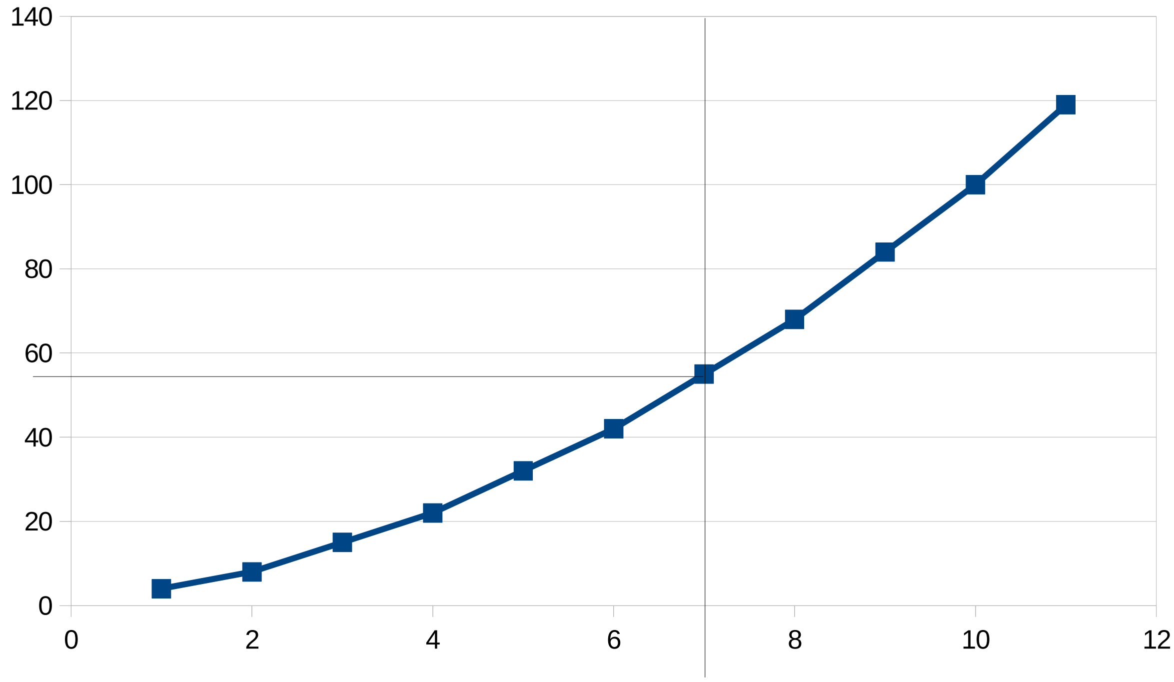
$$n \le m + \frac{1}{2} m T + \frac{1}{4} m^2 T + \left(1-(-1)^m\right)\frac{1}{8} T$$

$$54 \le m + \frac{3}{2} m + \frac{3}{4} m^2 + \left(1-(-1)^m\right)\frac{3}{8}$$

- total n =54 bits
- m =7 groups
- i-th group has x_i bits (size)
- constant skip delay $T = T(x_i)$ =3

1	4
2	8
3	15
4	22
5	32
6	42
7	55
8	68
9	84
10	100

$54 < 55$

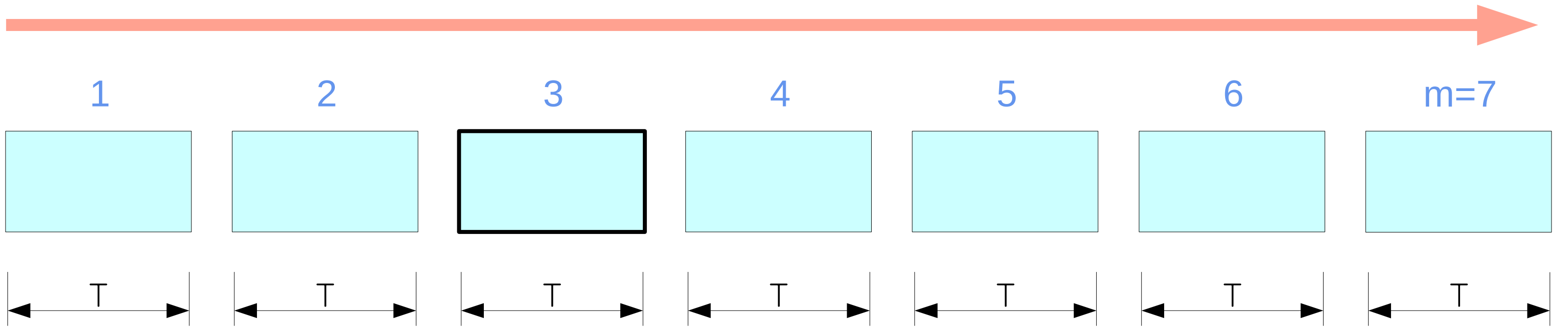


$m = 7$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Example 2 - (3)

n =54
m =7
T =3



$$y_i = \min\{1+iT, 1+(m+1-i)T\}, \quad i = 1, \ldots, m$$

$$y_1 = \min\{1+1\cdot T, 1+7\cdot T\} = 1+1\cdot T = 4 \qquad 0 \le x_1 \le 1+1\cdot T = 4$$

$$y_2 = \min\{1+2\cdot T, 1+6\cdot T\} = 1+2\cdot T = 7 \qquad 0 \le x_2 \le 1+2\cdot T = 7$$

$$y_3 = \min\{1+3\cdot T, 1+5\cdot T\} = 1+3\cdot T = 10 \qquad 0 \le x_3 \le 1+3\cdot T = 10$$

$$y_4 = \min\{1+4\cdot T, 1+4\cdot T\} = 1+4\cdot T = 13 \qquad 0 \le x_4 \le 1+4\cdot T = 13$$

$$y_5 = \min\{1+5\cdot T, 1+3\cdot T\} = 1+3\cdot T = 10 \qquad 0 \le x_5 \le 1+3\cdot T = 10$$

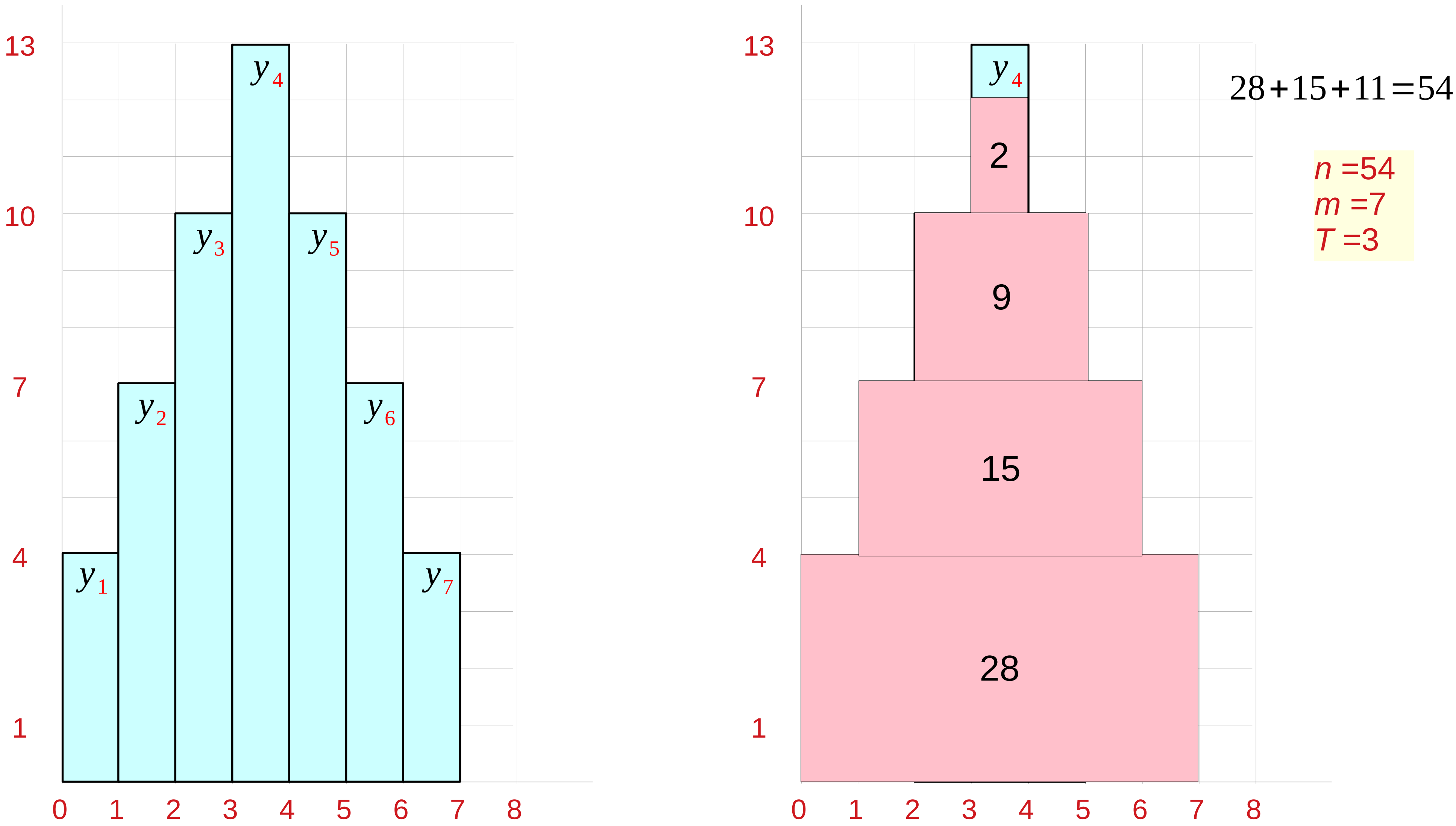
$$y_6 = \min\{1+6\cdot T, 1+2\cdot T\} = 1+2\cdot T = 7 \qquad 0 \le x_6 \le 1+2\cdot T = 7$$

$$y_7 = \min\{1+7\cdot T, 1+1\cdot T\} = 1+1\cdot T = 4 \qquad 0 \le x_7 \le 1+1\cdot T = 4$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

$$0 \le x_i \le y_i, i = 1, \ldots, m$$

Example 2 - (4)



Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Example 3 - (1)

Consider a 64 bit adder
From 2(i) we compute m=8.

the corresponding histogram is shown in Figure

The optimal gorup sizes are:

$$x_1 = x_8 = 4, x_2 = x_7 = 7, x_3 = x_6 = 10, x_4 = x_5 = 11$$

The delay of the longest signal is mT = 24

- total n =64 bits
- m =8 groups
- i-th group has x_i bits (size)
- constant skip delay $T = T(x_i) = 3$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Example 3 - (2)

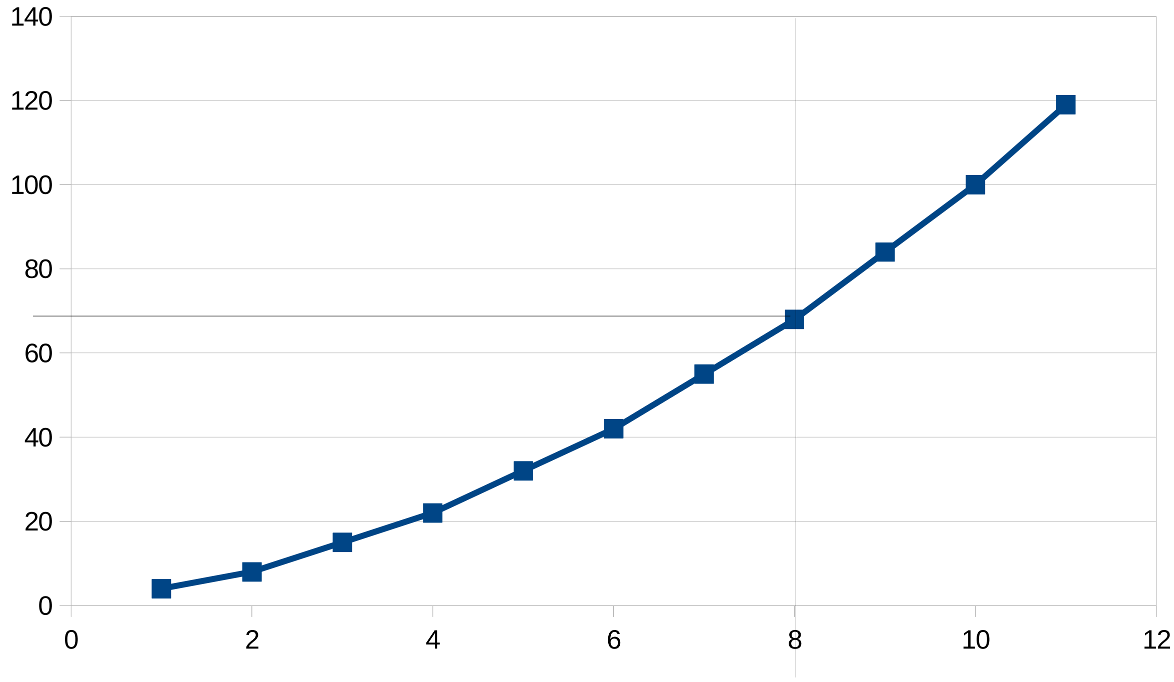
$$n \le m + \frac{1}{2} m T + \frac{1}{4} m^2 T + \left(1-(-1)^m\right)\frac{1}{8} T$$

$$64 \le m + \frac{3}{2} m + \frac{3}{4} m^2 + \left(1-(-1)^m\right)\frac{3}{8}$$

- total n =64 bits
- m =7 groups
- i-th group has x_i bits (size)
- constant skip delay $T = T(x_i)$ =3

1	4
2	8
3	15
4	22
5	32
6	42
7	55
8	68
9	84
10	100

$64 < 68$

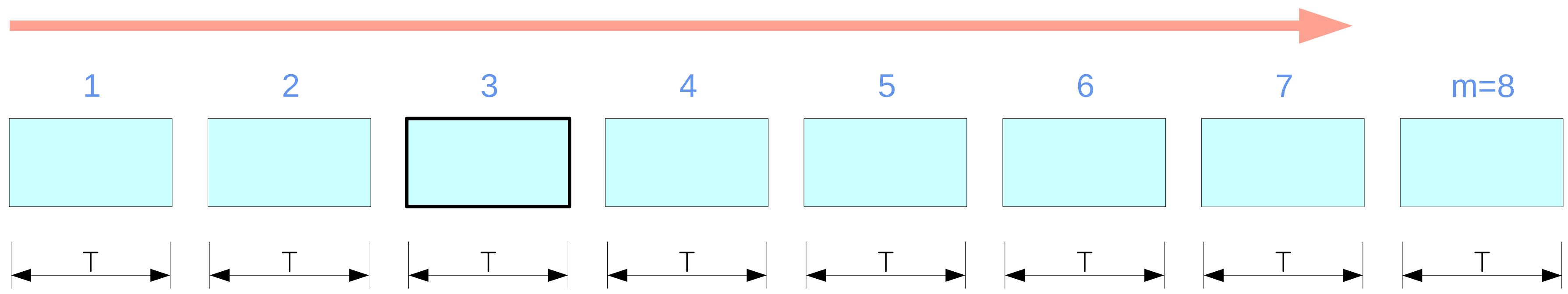


$m = 8$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Example 3 - (3)

n =64
m =8
T =3



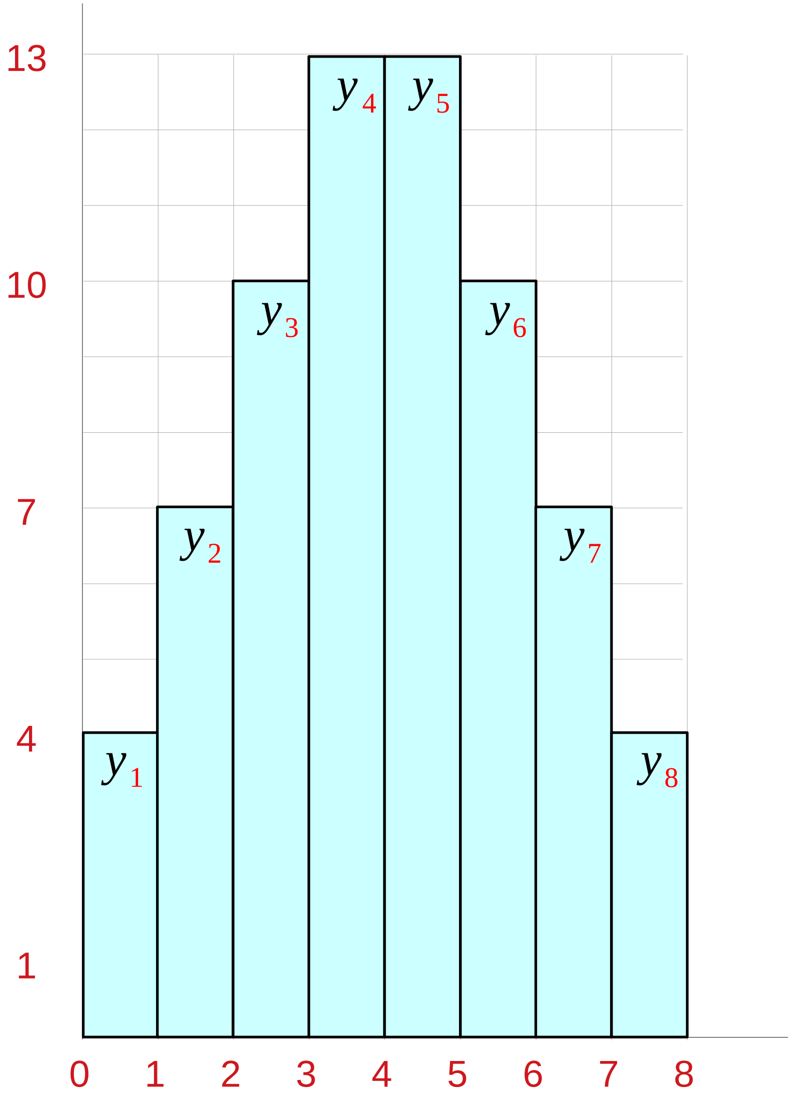
$$y_i = min\{1+iT, 1+(m+1-i)T\}, \quad i = 1,\dots,m$$

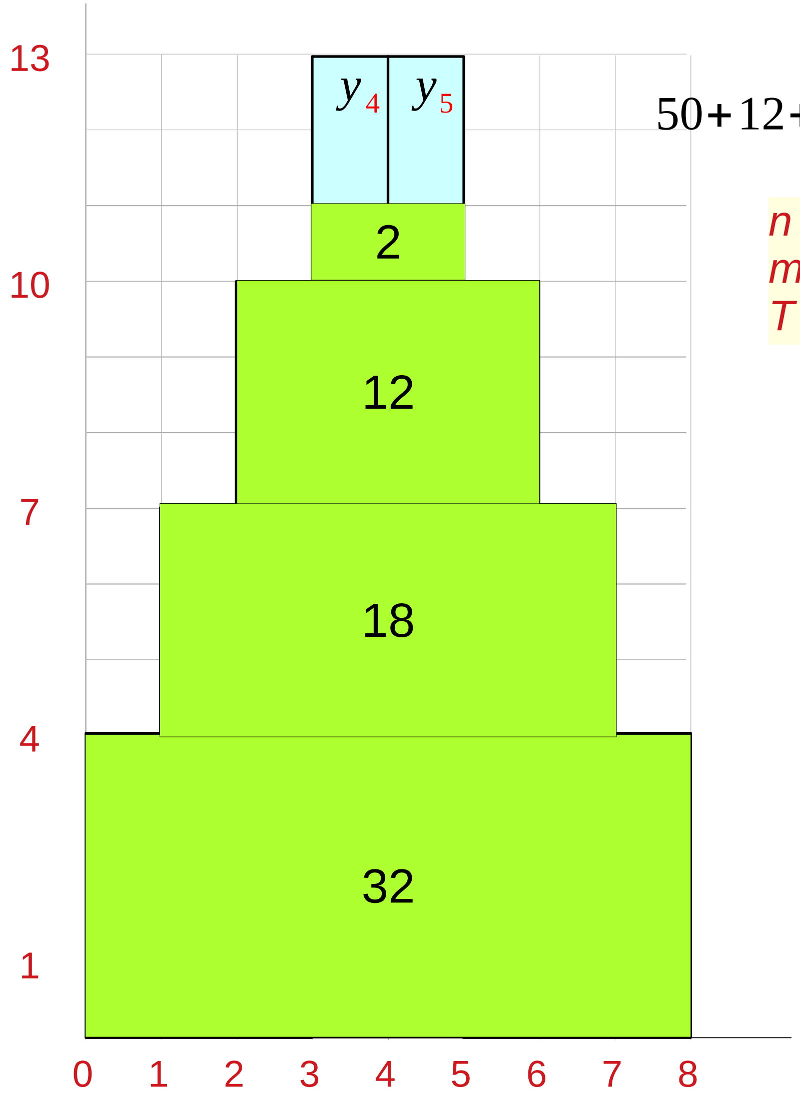
$$y_1 = min\{1+1\cdot T, 1+8\cdot T\} = 1+1\cdot T = 4 \qquad 0 \le x_1 \le 1+1\cdot T = 4$$
$$y_2 = min\{1+2\cdot T, 1+7\cdot T\} = 1+2\cdot T = 7 \qquad 0 \le x_2 \le 1+2\cdot T = 7$$
$$y_3 = min\{1+3\cdot T, 1+6\cdot T\} = 1+3\cdot T = 10 \qquad 0 \le x_3 \le 1+3\cdot T = 10$$
$$y_4 = min\{1+4\cdot T, 1+5\cdot T\} = 1+4\cdot T = 13 \qquad 0 \le x_4 \le 1+4\cdot T = 13$$
$$y_5 = min\{1+5\cdot T, 1+4\cdot T\} = 1+4\cdot T = 13 \qquad 0 \le x_5 \le 1+4\cdot T = 13$$
$$y_6 = min\{1+6\cdot T, 1+3\cdot T\} = 1+3\cdot T = 10 \qquad 0 \le x_6 \le 1+3\cdot T = 10$$
$$y_7 = min\{1+7\cdot T, 1+2\cdot T\} = 1+2\cdot T = 7 \qquad 0 \le x_7 \le 1+2\cdot T = 7$$
$$y_8 = min\{1+8\cdot T, 1+1\cdot T\} = 1+1\cdot T = 4 \qquad 0 \le x_8 \le 1+1\cdot T = 4$$

$$0 \le x_i \le y_i, i=1,\dots,m$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Example 3 - (4)





$50+12+2=64$

$n=64$
$m=8$
$T=3$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Ripple delay and skip delay

For a 32-bit adder, and the VBA scheme,
we divided the Carry chain into blocks of sizes
1, ,3, 5, 7, 7, 5, 3, 1.

Why this division is optimal?

Let *t* denote the time required for a carry signal
to ripple across a one bit in the carry chain,
and let *T* denote the time required for the signal
to skip over a group of bits.

By simulation of the blocks, we have found that
t = 0.8 ns and *T*=165ns.
To simplify our analysis, we normalize them
So that *t* = 1 and *T* = 2.

Then we apply the theory developed
for finding the optimal division of a carry chain

$\Delta_{rca} = 1$ ripple delay over a bit

$\Delta_{SKIP} = T$ skip delay over a group

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Maximum propagation time P

- ***n*** bits
- ***m*** groups

Lemma 1 When the bits of a carry skip adder are grouped according to the scheme (i)-(iii), the maximum propagation time of a carry signal is *mT*

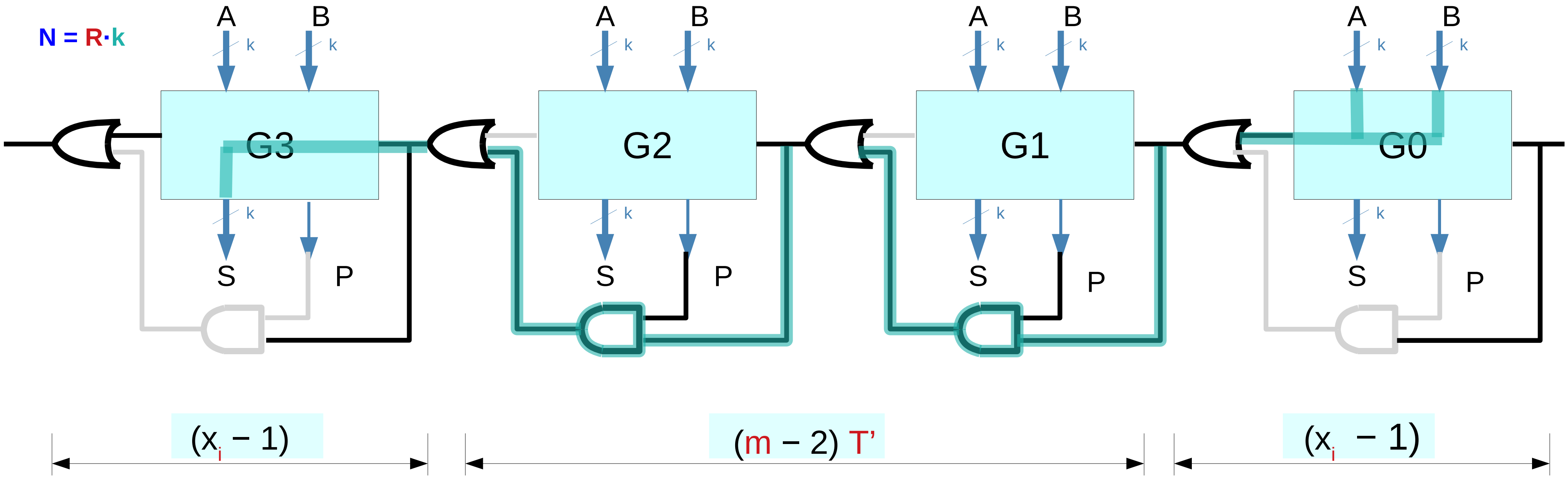
The carry generated at the 2nd bit position and terminating at the (n-1) clearly has propagation time *mT*. We must show that any other signal has propagation time smaller than or equal to *mT*.

Consider a signal originating in the *i-th* group and terminating in the *j-th*, i < j. Denote its propagation time by P. Clearly

$$P \le (x_i - 1) + (j - i - 1)T + (x_j - 1) \le mT$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Propagation delay P



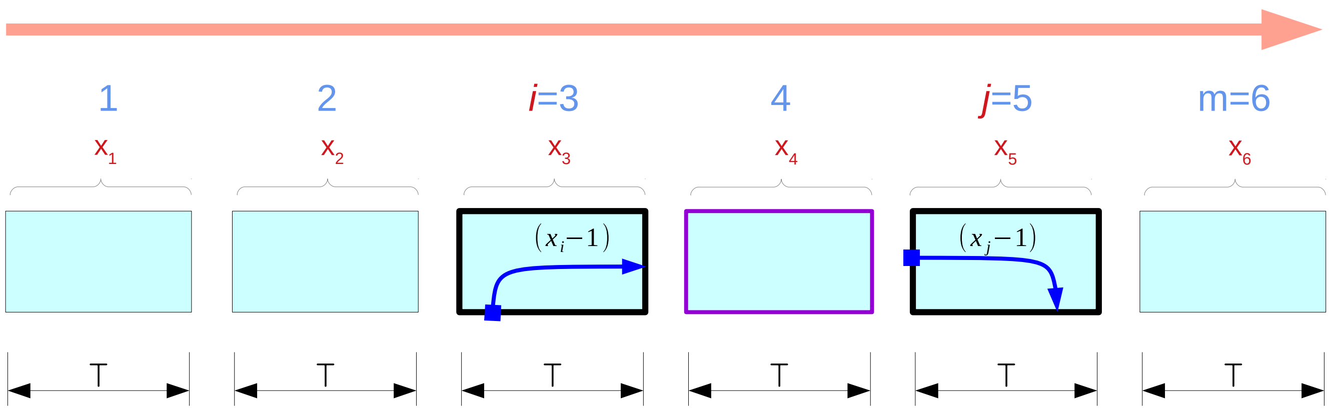
$\Delta_{rca} = 1$ ripple delay over a bit

$\Delta_{SKIP} = T$ skip delay over a group

$$P \le (x_i - 1) + (j - i - 1)T + (x_j - 1) \le mT$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Propagation delay P



- ***n*** bits
- ***m*** groups

j-i-1 = 5-3-1 = 1

$\Delta_{rca} = 1$

$\Delta_{SKIP} = T$

$$P \le (x_i - 1) + (j-i-1)T + (x_j - 1) \le mT$$

$$x_i \le min\{1 + iT, 1 + (m+1-i)T\}$$

$$x_j \le min\{1 + jT, 1 + (m+1-j)T\}$$

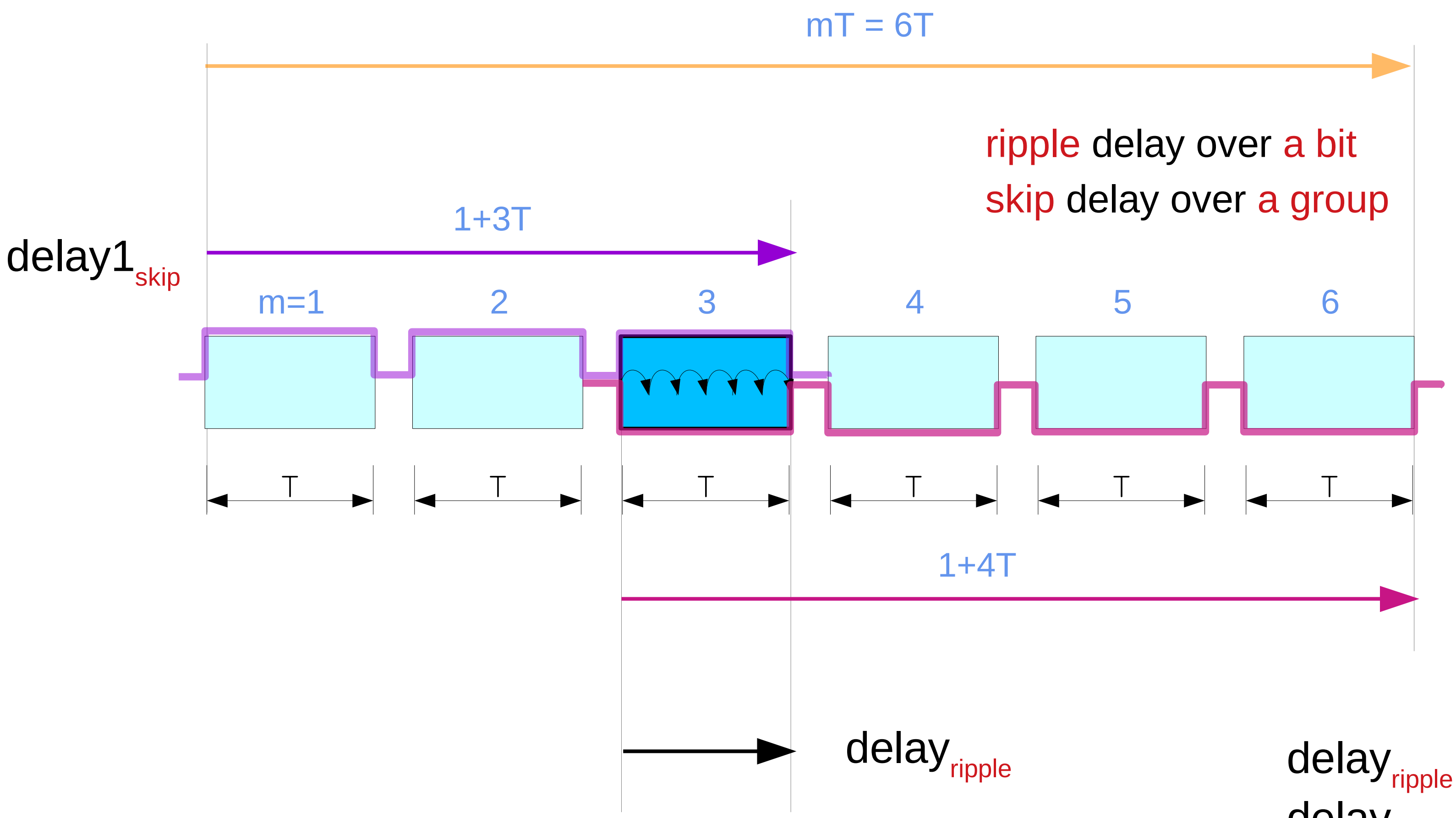
Assume a carry signal is generated in the *i-th* group and terminated in the *j-th*, i < j.

P denotes propagation time

$$P \le min\{1 + iT, 1 + (m+1-i)T\} + min\{1 + jT, 1 + (m+1-j)T\} + (j-i-1)T - 2$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model



- ***n*** bits
- ***m*** groups

ripple delay over a bit

skip delay over a group

$\text{delay}_{\text{ripple}} \leq \text{delay1}_{\text{skip}}$

$\text{delay}_{\text{ripple}} \leq \text{delay2}_{\text{skip}}$

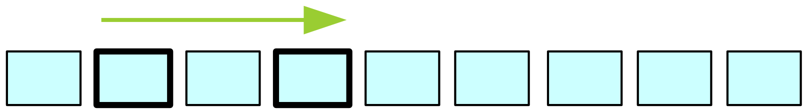
Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Three cases

Case 1

$$min\{1+iT, 1+(m+1-i)T\} = 1+iT$$

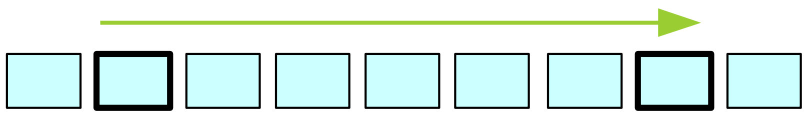
$$min\{1+jT, 1+(m+1-j)T\} = 1+jT$$



Case 2

$$min\{1+iT, 1+(m+1-i)T\} = 1+iT$$

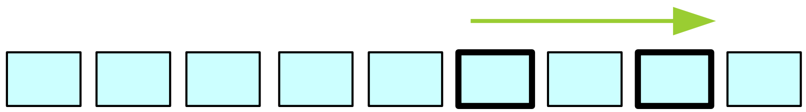
$$min\{1+jT, 1+(m+1-j)T\} = 1+(m+1-j)T$$



Case 3

$$min\{1+iT, 1+(m+1-i)T\} = 1+(m+1-i)T$$

$$min\{1+jT, 1+(m+1-j)T\} = 1+(m+1-j)T$$

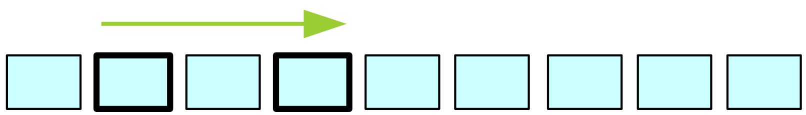


Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Case 1

1. First, assume

$$min\{1+iT, 1+(m+1-i)T\} = 1+iT$$
$$min\{1+jT, 1+(m+1-j)T\} = 1+jT$$



$$1+jT \le 1+(m+1-j)T \Rightarrow 2jT \le (m+1)T \Rightarrow 2jT - T \le mT$$

$$P < 1+iT + 1+(m+1-i)T + (j-i-1)T - 2 = 2jT - T \le mT$$

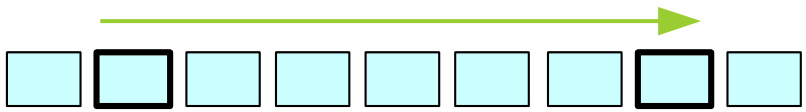
$$x_i \le min\{1+iT, 1+(m+1-i)T\} = 1+iT$$
$$x_j \le min\{1+jT, 1+(m+1-j)T\} = 1+jT$$

$$P \le min\{1+iT, 1+(m+1-i)T\} + min\{1+jT, 1+(m+1-j)T\} + (j-i-1)T - 2$$
$$= 1+iT + 1+jT + (j-i-1)T - 2$$
$$= 2jT - T \le mT$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Case 2

2. Now, assume



$$min\{1+iT, 1+(m+1-i)T\} = 1+iT$$

$$min\{1+jT, 1+(m+1-j)T\} = 1+(m+1-j)T$$

$$P \le 1+iT + 1 + (m+1-i)T + (j-i-1)T - 2 = mT$$

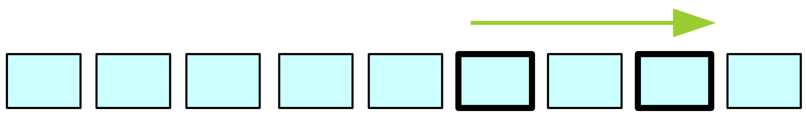
$$P \le min\{1+iT, 1+(m+1-i)T\} + min\{1+jT, 1+(m+1-j)T\} + (j-i-1)T - 2$$

$$P \le 1+iT + 1+(m+1-j)T + (j-i-1)T - 2 = mT$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Case 3

3. Finally, assume



$$min\{1+iT, 1+(m+1-i)T\}=1+(m+1-i)T$$

$$min\{1+jT, 1+(m+1-j)T\}=1+(m+1-j)T$$

$$P\le 1+(m+1-i)T+1+(m+1-j)T+(j-i-1)T-2=2mT-(2iT-T)\le 2mT-mT=mT$$

$$P \le min\{1+iT, 1+(m+1-i)T\} + min\{1+jT, 1+(m+1-j)T\} + (j-i-1)T - 2$$

$$P \le 1+(m+1-i)T + 1+(m+1-j)T + (j-i-1)T - 2 = 2(m+1-i)T-T$$

$$\le 2(m+1-i)T-T = 2mT-(2iT-T) = mT$$

$$1+iT \ge 1+(m+1-i)T \Rightarrow 2iT \ge (m+1)T \Rightarrow 2iT-T\ge mT$$

$$-(2iT-T)\le -mT$$

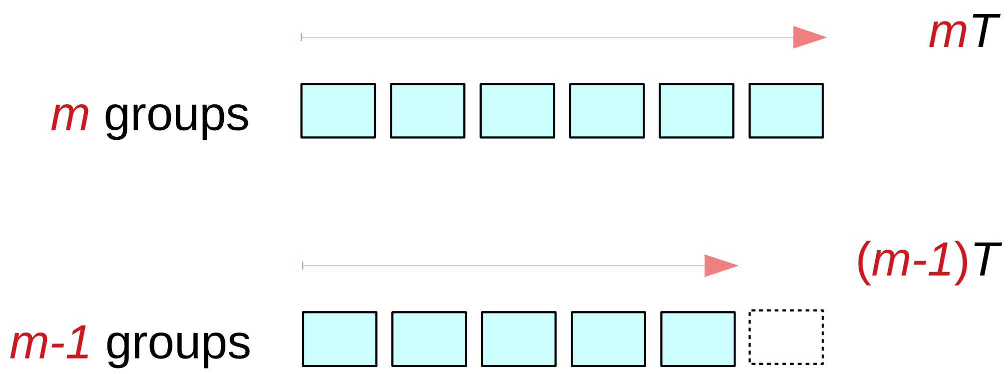
Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Maximum delay of a carry signal

Lemma 2 Let *D* denote the maximum delay of a carry signal in a *n* bit carry skip adder with group sizes chosen optimally. Then

$$(m-1)T \leq D \leq mT$$

Since we have exhibited a division of the carry chain into groups
In such a way that the maximum delay of a carry signal is *mT*
We clearly have $D \leq mT$



- ***n*** bits
- ***m*** groups

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Even number **r = 2k** of groups (1)

Let $x_{1,} x_{2,} \ldots , x_r$ denote the optimal group sizes corresponding to *D.*
For the moment assume that $r = 2k$ is even.
By considering carries originating in group *i*
and terminating in group , $r - i + 1,\ i = 1, \ldots , k$
we deduce the following system inequalities

the maximum delay of a carry signal is *mT*

$$D \leq mT$$

- ***n*** bits
- ***m*** groups

- ***n*** bits
- ***r*** groups – optimal

$$1,\quad 2k \qquad = r - (1-1),\quad i = 1$$

$$2,\quad 2k-1 = r - (2-1),\quad i = 2$$

$$k,\quad k+1 \qquad = r - (k-1),\quad i = k$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

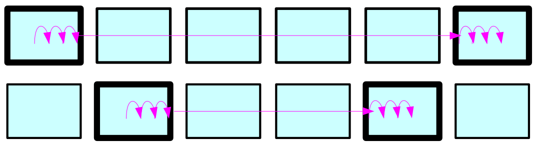
Even number **r = 2k** of groups (2)

$r = 2k$

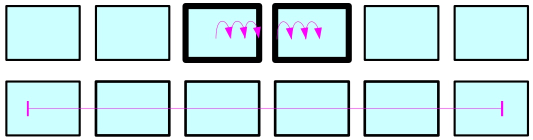
$$P \le (x_i - 1) + (j - i - 1)T + (x_j - 1) \le mT$$

- ***n*** bits
- ***r*** groups

$$(x_1 - 1) + (r-2)T + (x_r - 1) \le D$$
$$(x_2 - 1) + (r-4)T + (x_{r-1} - 1) \le D$$
$$(x_3 - 1) + (r-6)T + (x_{r-2} - 1) \le D$$



$$(x_k - 1) + (r-2k)T + (x_{k+1} - 1) \le D$$
$$mT \le D$$



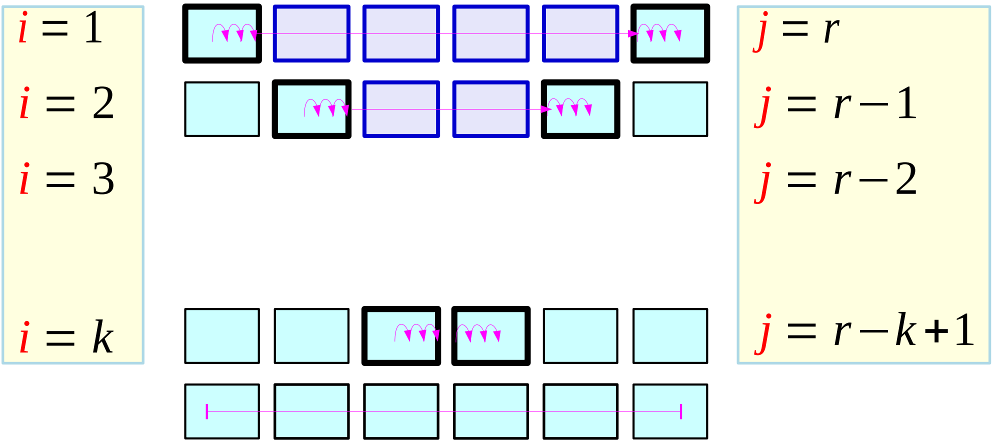
Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Even number **r = 2k** of groups (3)

$r = 2k$

$$P \le (x_i - 1) + (j - i - 1)T + (x_j - 1) \le mT$$

- ***n*** bits
- ***r*** groups



$(j - i - 1) = r - 2 \quad = 2(k-1)$

$(j - i - 1) = r - 4 \quad = 2(k-2)$

$(j - i - 1) = r - 6 \quad = 2(k-3)$

$(j - i - 1) = r - 2k \quad = 2(k-k)$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Even number **r = 2k** of groups (4)

$r = 2k$

$$P \le (x_i - 1) + (j - i - 1)T + (x_j - 1) \le mT$$

- ***n*** bits
- ***r*** groups

$$\begin{aligned}
(x_1 - 1) + 2(k-1)T + (x_{2k} - 1) &\le D \quad i = 1 \\
(x_2 - 1) + 2(k-2)T + (x_{2k-1} - 1) &\le D \quad i = 2 \\
(x_3 - 1) + 2(k-3)T + (x_{2k-2} - 1) &\le D \quad i = 3 \\
(x_k - 1) + 2(k-k)T + (x_{k+1} - 1) &\le D \quad i = k \\
rT \quad &\le D
\end{aligned}$$

$$\sum_{i=1}^{r} x_i = n$$

$$\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$$

$$2k = r$$

$$n - 2k + (k+1)rT - k(k+1)T \le (k+1)D$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Even number **r = 2k** of groups (5)

$r = 2k$

$$n - 2k + (k+1)rT - k(k+1)T \le (k+1)D$$

$$\frac{n-2k}{k+1} + rT - kT \le D$$

$$\frac{n-2k}{k+1} + 2kT - kT \le D$$

$$\frac{n-2(k+1)+2}{k+1} + kT \le D$$

$$\frac{n+2}{k+1} + (k+1)T - (T+2) \le D$$

$$2\sqrt{(n+2)T} - (T+2) \le D$$

$$\sqrt{4nT+8T} - (T+2) \le D$$

arith mean $\ge$ *geo mean*

$$\frac{n+2}{k+1} + (k+1)T \ge 2 \cdot \sqrt{\frac{n+2}{k+1} \cdot (k+1)T}$$

$$\frac{n+2}{k+1} + (k+1)T \ge 2\sqrt{(n+2)T}$$

min when $\frac{n+2}{k+1} = (k+1)T$

$$\frac{n+2}{T} = (k+1)^2$$

$$(k+1) = \sqrt{\frac{n+2}{T}}$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Odd number **r = 2k+1** of groups (1)

Let $x_1, x_2, \ldots, x_r$ denote the optimal group sizes corresponding to *D.*
For the moment assume that $r=2k+1$ is odd.
By considering carries originating in group *i*
and terminating in group , $r-i+1, \; i = 1,\ldots,k$
we deduce the following system inequalities

the maximum delay of a carry signal is *mT*

$$D \le mT$$

- ***n*** bits
- ***m*** groups

- ***n*** bits
- ***r*** groups – optimal

$$1, \quad 2k+1 \quad = r-(1-1), \quad i = 1$$

$$2, \quad 2k \quad = r-(2-1), \quad i = 2$$

$$k, \quad k+2 \quad = r-(k-1), \quad i = k$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

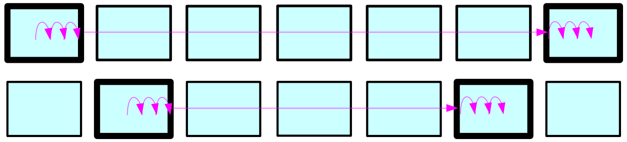
Odd number **r = 2k+1** of groups (2)

$r = 2k + 1$

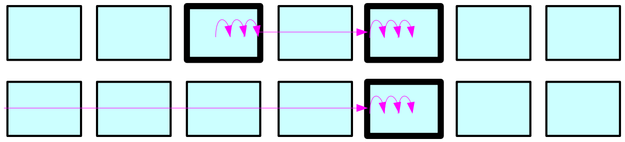
$$P \le (x_i - 1) + (j - i - 1)T + (x_j - 1) \le mT$$

- ***n*** bits
- ***r*** groups

$$(x_1 - 1) + (r - 2)T + (x_r - 1) \le D$$
$$(x_2 - 1) + (r - 4)T + (x_{r-1} - 1) \le D$$
$$(x_3 - 1) + (r - 6)T + (x_{r-2} - 1) \le D$$



$$(x_k - 1) + (r - 2k)T + (x_{r-k+1} - 1) \le D$$
$$kT + (x_{k+1} - 1) \le D$$



Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Odd number **r = 2k+1** of groups (3)

$r = 2k+1$

$$P \le (x_i - 1) + (j - i - 1)T + (x_j - 1) \le mT$$

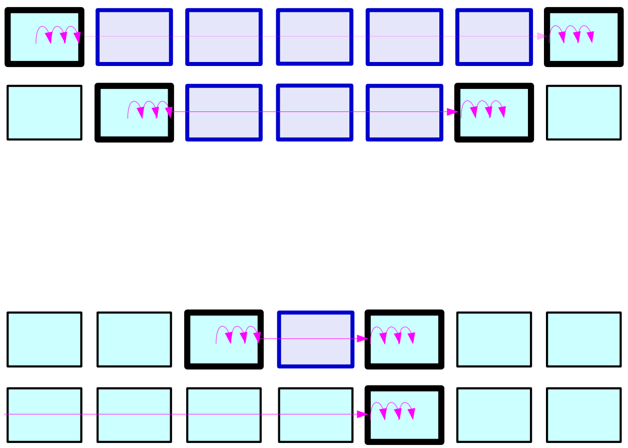
- ***n*** bits
- ***r*** groups

$i = 1$

$i = 2$

$i = 3$

$i = k$



$j = r$

$j = r-1$

$j = r-2$

$j = r-k+1$

$(j - i - 1) = r-2 \quad = 2(k-1)$

$(j - i - 1) = r-4 \quad = 2(k-2)+1$

$(j - i - 1) = r-6 \quad = 2(k-3)+1$

$(j - i - 1) = r-2k \quad = 2(k-k)+1$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Odd number **r = 2k+1** of groups (4)

$r = 2k+1$

$$P \le (x_i - 1) + (j - i - 1)T + (x_j - 1) \le mT$$

- ***n*** bits
- ***r*** groups

$$(x_1 - 1) + 2(k - 1)T + T + (x_{2k+1} - 1) \le D$$
$$(x_2 - 1) + 2(k - 2)T + T + (x_{2k} - 1) \le D$$
$$(x_3 - 1) + 2(k - 3)T + T + (x_{2k-1} - 1) \le D$$
$$(x_k - 1) + 2(k - k)T + T + (x_{k+2} - 1) \le D$$
$$kT + (x_{k+1} - 1) \le D$$

$$\sum_{i=1}^{r} x_i = n$$

$$\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$$

$$2k+1 = r$$

$$n - 2k - 1 + (r+1)kT - k(k+1)T \le (k+1)D$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Odd number **r = 2k+1** of groups (5)

$r = 2k+1$

$$n - 2k - 1 + (r+1)kT - k(k+1)T \le (k+1)D$$

$$\frac{n-2k-1}{(k+1)} + \frac{(r+1)kT}{(k+1)} - kT \le D$$

$$\frac{n-2k-1}{(k+1)} + \frac{(2(k+1))kT}{(k+1)} - kT \le D$$

$$\frac{n-2k-1}{(k+1)} + kT \le D$$

$$\frac{n-2(k+1)+1}{(k+1)} + (k+1)T - T \le D$$

$$\frac{(n+1)}{(k+1)} + (k+1)T - (T+2) \le D$$

$$2 \cdot \sqrt{(n+1)T} - (T+2) \le D$$

$$\sqrt{4nT+4T} - (T+2) \le D$$

arith mean $\ge$ *geo mean*

$$\frac{(n+1)}{(k+1)} + (k+1)T \ge 2 \cdot \sqrt{(n+1)T}$$

min when $\frac{(n+1)}{(k+1)} = (k+1)T$

$$\frac{n+1}{T} = (k+1)^2$$

$$(k+1) = \sqrt{\frac{n+1}{T}}$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

$r=2k$ $\qquad \sqrt{4nT+8T} - (T+2) \le D$

$r=2k+1$ $\qquad \sqrt{4nT+4T} - (T+2) \le D$

$$\sqrt{4nT+4T} - (T+2) \le \sqrt{4nT+8T} - (T+2)$$

$$\sqrt{4nT+4T} - (T+2) \le D$$

We will not produce an upper bound on mT.

Since *m* is the smallest positive integer satisfying

$$(m-1) + \frac{1}{2}(m-1)T + \frac{1}{4}(m-1)^2 T + (1-(-1)^{m-1})\frac{T}{8} < n$$

$$n \le m + \frac{1}{2}mT + \frac{1}{4}m^2 T + (1-(-1)^m)\frac{1}{8}T$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

$$(m-1) + \frac{1}{2}(m-1)T + \frac{1}{4}(m-1)^2 T + (1-(-1)^{m-1})\frac{T}{8} < n$$

$$(m-1) + \frac{1}{2}(m-1)T + \frac{1}{4}(m-1)^2 T + (1-(-1)^{m-1})\frac{T}{8} + 1 \leq n$$

$$(m) + \frac{1}{2}(mT-T) + \frac{1}{4}(m^2 T - 2mT + T) + (1-(-1)^{m-1})\frac{T}{8} \leq n$$

$$m - \frac{1}{4}T + \frac{1}{4}m^2 T + (1-(-1)^{m-1})\frac{T}{8} \leq n$$

$$\frac{1}{4}m^2 T \leq n - m + \frac{1}{4}T - (1-(-1)^{m-1})\frac{T}{8}$$

$$\left(\frac{1}{4}m^2 T\right)\cdot 4T \leq \left(n - m + \frac{1}{4}T - (1-(-1)^{m-1})\frac{T}{8}\right)\cdot 4T$$

$$m^2 T^2 \leq 4nT - 4mT + T^2 - (1-(-1)^{m-1})\frac{T^2}{2}$$

$$m^2 T^2 + 4mT \leq 4nT + T^2 - (1-(-1)^{m-1})\frac{T^2}{2}$$

$$m^2 T^2 + 4mT + 4 \leq 4 + 4nT + T^2 - (1-(-1)^{m-1})\frac{T^2}{2}$$

$$(mT + 2)^2 \leq 4 + 4nT + T^2 - (1-(-1)^{m-1})\frac{T^2}{2}$$

$$(mT + 2) \leq \sqrt{4 + 4nT + T^2 - (1-(-1)^{m-1})\frac{T^2}{2}}$$

$$mT \leq -2 + \sqrt{4 + 4nT + T^2 - (1-(-1)^{m-1})\frac{T^2}{2}}$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

$r=2k$ $\quad\quad \sqrt{4nT+8T} - (T+2) \le D$

$r=2k+1$ $\quad\quad \sqrt{4nT+4T} - (T+2) \le D$

$$mT \le -2 + \sqrt{4 + 4nT + T^2 - (1-(-1)^{m-1})\frac{T^2}{2}}$$

$$mT - D \le T + \frac{T^2 - 8T + 4 - (1-(-1)^{m-1})\frac{T^2}{2}}{\sqrt{4nT+8T} + \sqrt{4nT+4+T^2-(1-(-1)^{m-1})\frac{T^2}{2}}} \quad \text{even } r$$

$$mT - D \le T + \frac{T^2 - 4T + 4 - (1-(-1)^{m-1})\frac{T^2}{2}}{\sqrt{4nT+4T} + \sqrt{4nT+4+T^2-(1-(-1)^{m-1})\frac{T^2}{2}}} \quad \text{odd } r$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

$$mT \le -2 + \sqrt{4nT + T^2 + 4 - (1-(-1)^{m-1})\frac{T^2}{2}}$$

$r = 2k$

$$\sqrt{4nT + 8T} - (T+2) \le D$$

$$mT - D \le T + \frac{T^2 - 8T + 4 - (1-(-1)^{m-1})\frac{T^2}{2}}{\sqrt{4nT+8T} + \sqrt{4nT + T^2 + 4 - (1-(-1)^{m-1})\frac{T^2}{2}}}$$

$r = 2k+1$

$$\sqrt{4nT + 4T} - (T+2) \le D$$

$$mT - D \le T + \frac{T^2 - 4T + 4 - (1-(-1)^{m-1})\frac{T^2}{2}}{\sqrt{4nT+4T} + \sqrt{4nT + T^2 + 4 - (1-(-1)^{m-1})\frac{T^2}{2}}}$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

r=2k even r

$r=2k$

$$mT \le -2 + \sqrt{4nT + T^2 + 4 - \left(1-(-1)^{m-1}\right)\frac{T^2}{2}}$$

$$-D \le -\sqrt{4nT+8T} + (T+2)$$

$$mT-D \le T - \sqrt{4nT+8T} + \sqrt{4nT+T^2+4-\left(1-(-1)^{m-1}\right)\frac{T^2}{2}}$$

$$mT-D \le T + \frac{T^2-8T+4-\left(1-(-1)^{m-1}\right)\frac{T^2}{2}}{\sqrt{4nT+8T} + \sqrt{4nT+T^2+4-\left(1-(-1)^{m-1}\right)\frac{T^2}{2}}}$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

r=2k even r

$r = 2k$

$X \overset{\text{def}}{=} 4 - (1 - (-1)^{m-1}) \frac{T^2}{2}$

$$mT \le -2 + \sqrt{4nT + T^2 + X}$$

$$-D \le -\sqrt{4nT + 8T} + (T + 2)$$

$$mT - D \le T - \sqrt{4nT + 8T} + \sqrt{4nT + T^2 + X}$$

$$(-\sqrt{a} + \sqrt{b}) \cdot \frac{(+\sqrt{a} + \sqrt{b})}{(+\sqrt{a} + \sqrt{b})} = \frac{(-a + b)}{(+\sqrt{a} + \sqrt{b})}$$

$$\left(-\sqrt{4nT + 8T} + \sqrt{4nT + T^2 + X}\right) \cdot \frac{\left(\sqrt{4nT + 8T} + \sqrt{4nT + T^2 + X}\right)}{\left(\sqrt{4nT + 8T} + \sqrt{4nT + T^2 + X}\right)}$$

$$\{-\sqrt{4nT + 8T} + \sqrt{4nT + T^2 + X}\} \cdot \{+\sqrt{4nT + 8T} + \sqrt{4nT + T^2 + X}\}$$

$$= -(4nT + 8T) + (4nT + T^2 + X) = T^2 - 8T + X$$

$$mT - D \le T + \frac{T^2 - 8T + X}{\sqrt{4nT + 8T} + \sqrt{4nT + T^2 + X}}$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

r=2k+1 odd r

$r = 2k + 1$

$\sqrt{4nT + 4T} - (T + 2) \leq D$

$-\sqrt{4nT + 4T} + (T + 2) \geq -D$

$mT \leq -2 + \sqrt{4nT + T^2 + 4 - (1 - (-1)^{m-1})\frac{T^2}{2}}$

$-D \leq -\sqrt{4nT + 4T} + (T + 2)$

$mT - D \leq T - \sqrt{4nT + 8T} + \sqrt{4nT + T^2 + 4 - (1 - (-1)^{m-1})\frac{T^2}{2}}$

$$mT - D \leq T + \frac{T^2 - 4T + 4 - (1 - (-1)^{m-1})\frac{T^2}{2}}{\sqrt{4nT + 4T} + \sqrt{4nT + T^2 + 4 - (1 - (-1)^{m-1})\frac{T^2}{2}}}$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

r=2k+1 odd r

$r = 2k + 1$

$X \stackrel{\text{def}}{=} 4 - (1 - (-1)^{m-1}) \dfrac{T^2}{2}$

$$mT \leq -2 + \sqrt{4nT + T^2 + X}$$

$$-D \leq -\sqrt{4nT + 4T} + (T + 2)$$

$$mT - D \leq T - \sqrt{4nT + 4T} + \sqrt{4nT + T^2 + X}$$

$$(-\sqrt{a} + \sqrt{b}) \cdot \frac{(+\sqrt{a} + \sqrt{b})}{(+\sqrt{a} + \sqrt{b})} = \frac{(-a + b)}{(+\sqrt{a} + \sqrt{b})}$$

$$\left(-\sqrt{4nT + 4T} + \sqrt{4nT + T^2 + X}\right) \cdot \frac{\left(\sqrt{4nT + 4T} + \sqrt{4nT + T^2 + X}\right)}{\left(\sqrt{4nT + 4T} + \sqrt{4nT + T^2 + X}\right)}$$

$$\left\{-\sqrt{4nT + 4T} + \sqrt{4nT + T^2 + X}\right\} \cdot \left\{+\sqrt{4nT + 4T} + \sqrt{4nT + T^2 + X}\right\}$$

$$= -(4nT + 4T) + (4nT + T^2 + X) = T^2 - 4T + X$$

$$mT - D \leq T + \frac{T^2 - 4T + X}{\sqrt{4nT + 4T} + \sqrt{4nT + T^2 + X}}$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

r=2k even r

$$X \stackrel{\text{def}}{=} 4-\left(1-(-1)^{m-1}\right)\frac{T^2}{2}$$

- odd m $\quad X = 4$
- even m $\quad X = 4 - T^2$

$$mT - D \le T + \frac{T^2 - 8T + X}{\sqrt{4nT + 8T} + \sqrt{4nT + T^2 + X}}$$

$$\le T + \frac{T^2 - 8T + 4}{\sqrt{4nT + 8T} + \sqrt{4nT + T^2 + 4}}$$

$$\le T + \frac{-8T + 4}{\sqrt{4nT + 8T} + \sqrt{4nT + 4}}$$

$$\sqrt{4nT + T^2 + 4 - \left(1-(-1)^{m-1}\right)\frac{T^2}{2}} = \sqrt{4nT + T^2 + X}$$

- odd m

$$(-1)^{m-1} = 1$$

$$\left(1-(-1)^{m-1}\right) = 0$$

$$\sqrt{4nT + 4 + T^2}$$

- even m

$$(-1)^{m-1} = -1$$

$$\left(1-(-1)^{m-1}\right) = 2$$

$$\sqrt{4nT + T^2 + 4 - T^2}$$

$$= \sqrt{4nT + 4}$$

$$mT - D \le T + \frac{T^2 - 4T + X}{\sqrt{4nT + 4T} + \sqrt{4nT + T^2 + X}}$$

$$\le T + \frac{T^2 - 4T + 4}{\sqrt{4nT + 4T} + \sqrt{4nT + T^2 + 4}}$$

$$\le T + \frac{-4T + 4}{\sqrt{4nT + 4T} + \sqrt{4nT + 4}}$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

r=2k even r

$$mT - D \le T + \frac{T^2 - 8T + X}{\sqrt{4nT + 8T} + \sqrt{4nT + T^2 + X}}$$

$$X \overset{\text{def}}{=} 4 - \left(1 - (-1)^{m-1}\right)\frac{T^2}{2}$$

$$mT - D \le T + \frac{T^2 - 4T + X}{\sqrt{4nT + 4T} + \sqrt{4nT + T^2 + X}}$$

For n sufficiently large, we have $mT - \Delta < T + 1$
And since $mT - \Delta$ is an integer, $mT - \Delta \le T$
This completes the proof of the lemma

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

The scheme 2(i)- 2(iii) given above
for dividing the bits of a carry skip adder
Into groups is optimal for $2 \le T \le 7$

Assume the scheme is not optimal and
let D be the maximum delay
Corresponding to an optimal division of the bits into groups
Assume there are r groups in the optimal division.
Since a carry in signal to the least significant bit group can skip over
Each group we have $rT \le D \le mT$, so $r \le m$
If $r = m$ then $D = mT$ and the theorem holds by Lemma 1

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

If *r* = *m-1* m and *r* have different parities
and it follows from (5) that
$mT - D < T$ for $2 \le T \le 7$
so that $D > (m-1)T = rT$

This means that a signal
which skips over each of the *r* groups
has delay less than the maximum %DELTA

Similarly, if r < *m-1*, $D > (m-1)T > rT$
So that a signal which skips over each group
has delay < D

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

It follows that a signal with delay D
must start in a group *i*,

ripple to the end of this group,
then skip over $s < r$ groups and either terminate,
or ripple through the first few bits of a group $j > i$

Let x_i and x_j denote the lengths
of the *i*-th and *j*-th groups respectively.

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

Assume that *i* is chosen as small as possible
and *j* as large as possible.

A signal originating in group *i*,
rippling to the end of this group
and then skipping over the next *s* group has delay

$$D \le (x_i - 1) + sT \le (x_i - 1) + (r-1)T \le (x_i - 1) + (m-2)T\,.$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

Since %DELTA >= (m-1)T this implies that x_i >= T=1
Divide group I into two groups such that the group containing the most significant Bits has size T.
Since the i-th group is the first group in which a signal having maximum delay can Originate, this subdivision does not increase the delay of any carry signal of maximum delay
However, it increases the number of groups by 1

Suppose now that a carry signal originates in a group I, ripples to its end, Skips over s <= r-2 groups and finally ripples through the first few bits of a group J and terminates. We then have
%DELTA <= (x_i-1) + sT + (x_j-1) <= x_i + x_j – 2 + (m – 3)T

pp

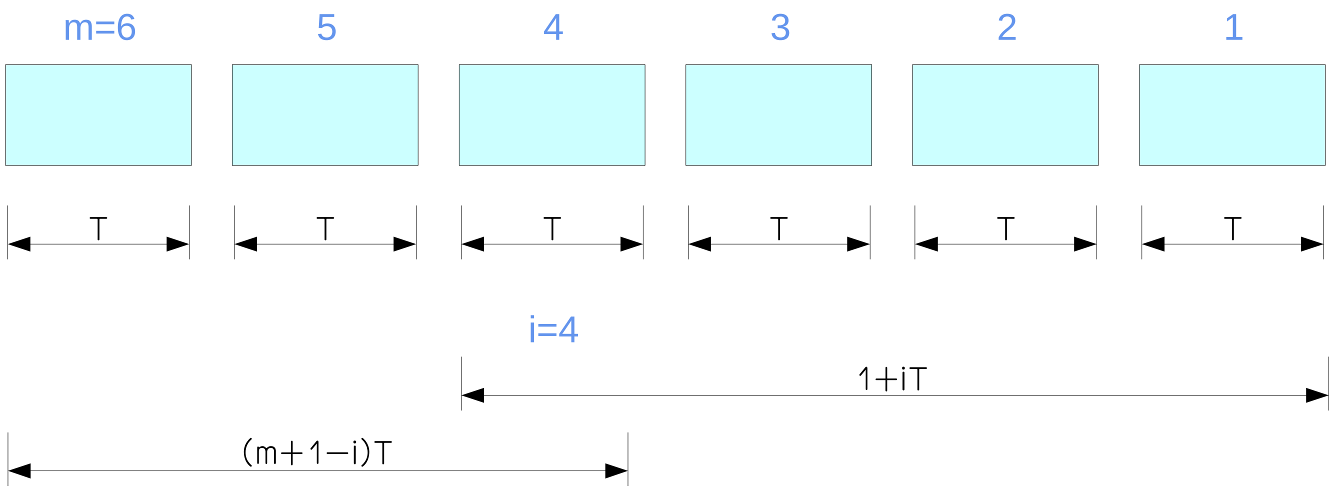
Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

So that either $x_i >= T+1$ or $x_j >= T+1$.
This means that we can subdivide one of the groups I,j without increasing %DELTA
Continuing in this way, we can always increase the number r of group
In an optimal division of a carry chain by 1 without increasing %DELTA
If $r < m$
This means that we can arrive at an optimal division of the carry chain
Into m groups.
We must then have %DELTA >= mT which, together with Lemma 2,
Implies %DELTA = mT
This completes the proof of the theorem

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model



$$y_i = min\{1 + iT, 1 + (m+1-i)T\}$$

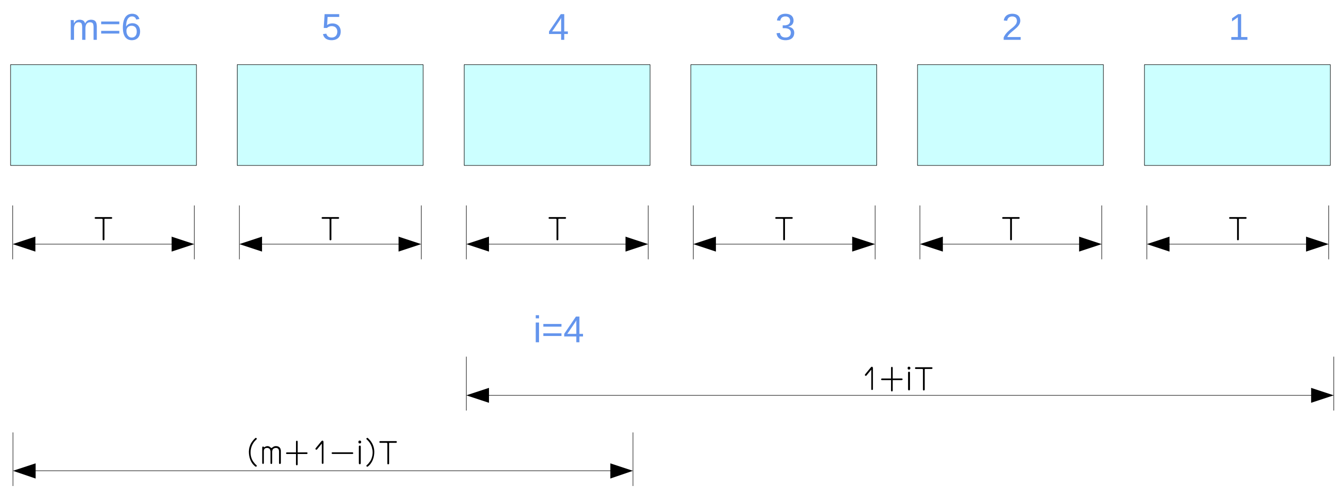
$$y_{1,} \dots, y_m$$

$$0 \le x_i \le y_i, i = 1, \dots, m$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

$$\sum_{i=1}^{m} x_i = n$$

Delay model



$$y_i = min\{1 + iT, 1 + (m+1-i)T\}$$

$$y_{1,} \ldots, y_m$$

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

Given *m*, an optimal division of the carry chain into groups
Can be obtained as follows
Let

$$y_i = min\{1+iT, 1+(m+1-i)T\}$$

Given $y_{1,} \ldots, y_m$ solve the minimization problem

$$\min_x max\{x_{1,} \ldots, x_n\}$$

Subject to

$$0 \le x_i \le y_i, i=1, \ldots, m$$

And

$$\sum_{i=1}^{m} x_i = n$$

Any solution $x_{1,} \ldots, x_m$ gives optimal group sizes
for a division of the carry chain

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

The x's can be computed iteratively as follows:

Initially take $x_1 = x_m = 0$

At each iteration, increase as many of the x's as possible by one unit, without violating the constraints

$$0 \le x_i \le y_i, i = 1, \ldots, m \qquad \sum_{i=1}^{m} x_i \le n$$

An easy calculation shows that

$$\sum_{i=1}^{m} y_i = m + \frac{1}{2} mT + \frac{1}{4} m^2 T + \left(1 - (-1)^m\right) \frac{1}{8} T \ge n$$

Thus, at some iteration, we have $\sum_{i=1}^{m} x_i = n$ and
The algorithm terminates

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

For n=32, we have m=7, (y1, y2, y3, y4, y5, y6, y7) = (3,5,7,9,7,5,3)
The above algorithm gives (x1, x2, x3, x4, x5, x6, x7) = (3,5,5,6,5,5,3)

A carry chain divided in this way has maximum delay D = mT =14
Since one unit of delay is 0.8ns, the maximum delay for 32-bit carry chain
is D = 14*0.8ns = 11.2ns
This time involves only the delay in the carry chain

It is easy to check that this is also the delay for a chain divided into groups of
sizes 1,3,5,7,7,5,3,1.
Thus this is also an optimal subdivision

The worst case delay includes the time needed to generate p_i and g_i signals
Delay of the carry chain, and the time for producing last sum bit s_n

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

Implement it with a string of multiplexers

The multiplexer cell is designed as very fast

Multiplexers are designed
as very fast structures using buffered pass gates and
in this sense are similar to the Manchester carry chain
which has been shown to be
the most effective implementation of a carry chain

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

The implementation of a single carry block is done
by mixing a 4 to 1 multiplexer (actually used as a 3 to 1)

In the last stage with a string of 2 to 1 multiplexers

a carry bypass is connected to inputs 3 and 4
of the 4:1 multiplexer (group carry multiplexer)
and the selection of the carry bypass is activated
by the NAND gate singaling when the condition
for group propagate is reached and
activating the group multiplexer in turn.

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers

Delay model

The32-bit implementation of the VBA adder is obtained
By connecting the groups of the sizes calculated
For the full length of n=32 bits

To increase the speed further we used a faster inverting version
Of the multiplexer, alternating between Ci and Cb_i signals

Oklobdzija: High-Speed VLSI arithmetic units : adders and multipliers