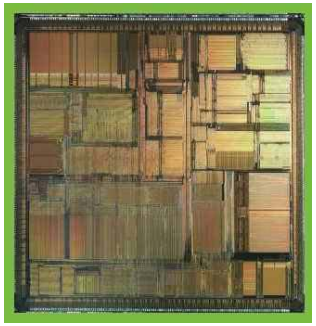
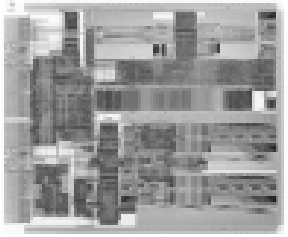
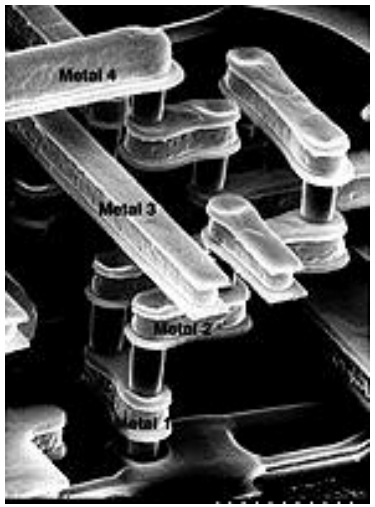




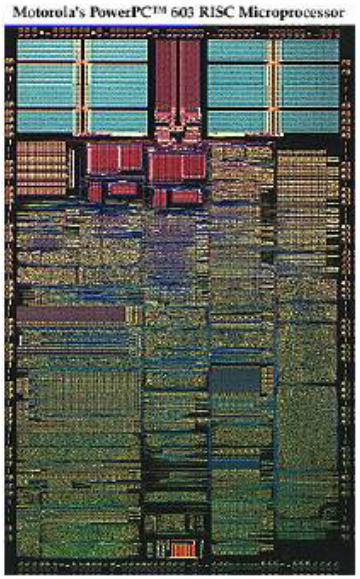
VLSI Arithmetic



Lecture 10:
Multipliers

Prof. Vojin G. Oklobdzija
University of California

http://www.ece.ucdavis.edu/acsel



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

A Method for Generation of Fast Parallel Multipliers

by

Vojin G. Oklobdzija
David Villeger
Simon S. Liu

Electrical and Computer Engineering
University of California
Davis

Fast Parallel Multipliers

Objective

Improved Speed of Parallel Multiplier via:

- Improvements in Partial-Product Bit Reduction Techniques
- Optimization of the Final Adder for the Uneven Signal Arrival Profile from the Multiplier Tree

Multiplication

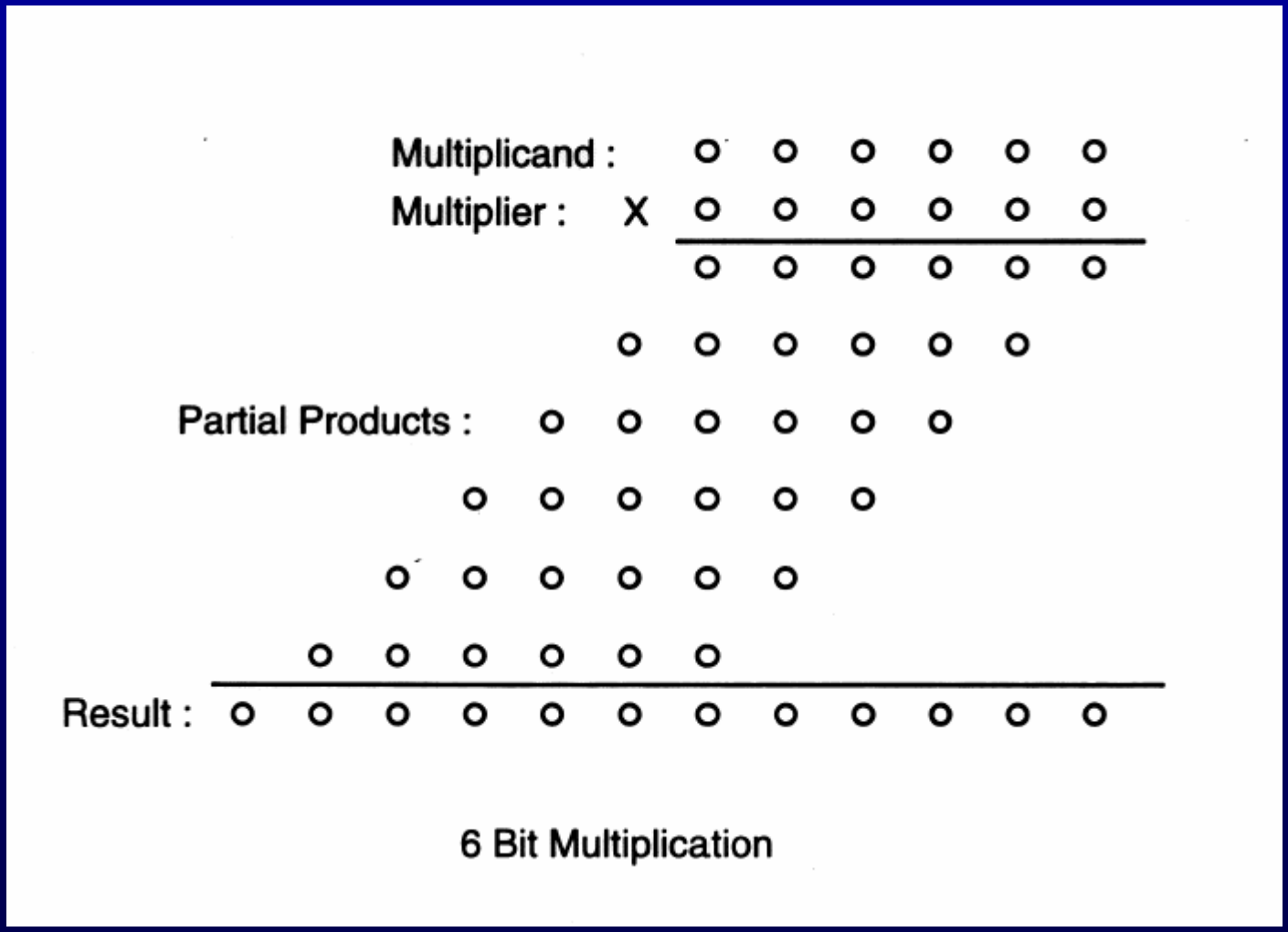
Algorithm:

$$P = XY = X \times \sum_{i=0}^{n-1} y^i r^i = \sum_{i=0}^{n-1} X \times y^i r^i$$

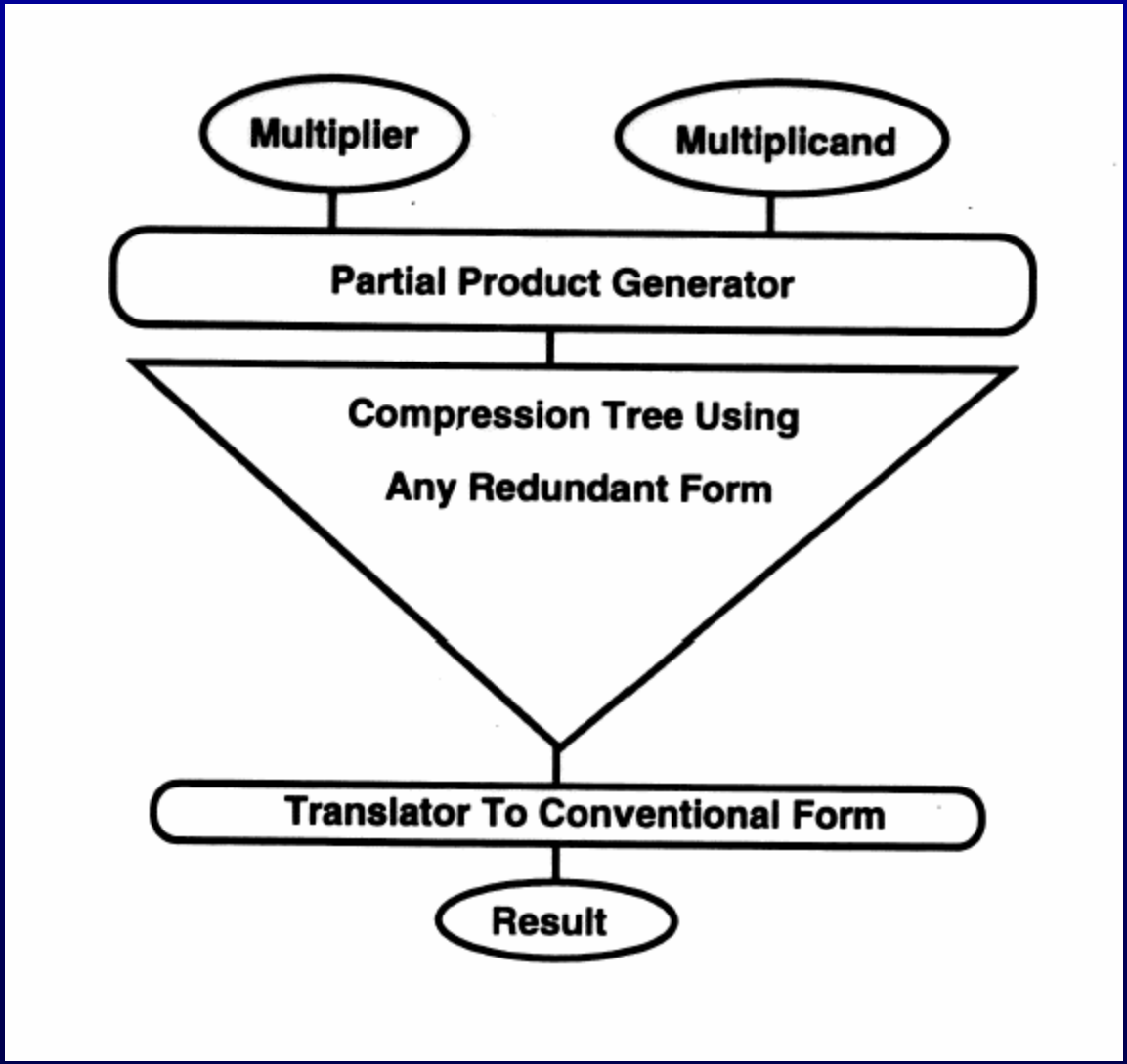
$$p^{(0)} = 0 \qquad \textit{initially}$$

$$p^{(j+1)} = \frac{1}{r}(p^j + r^n X y_j) \quad \textit{for } j=0,\ldots,n\text{-}1$$

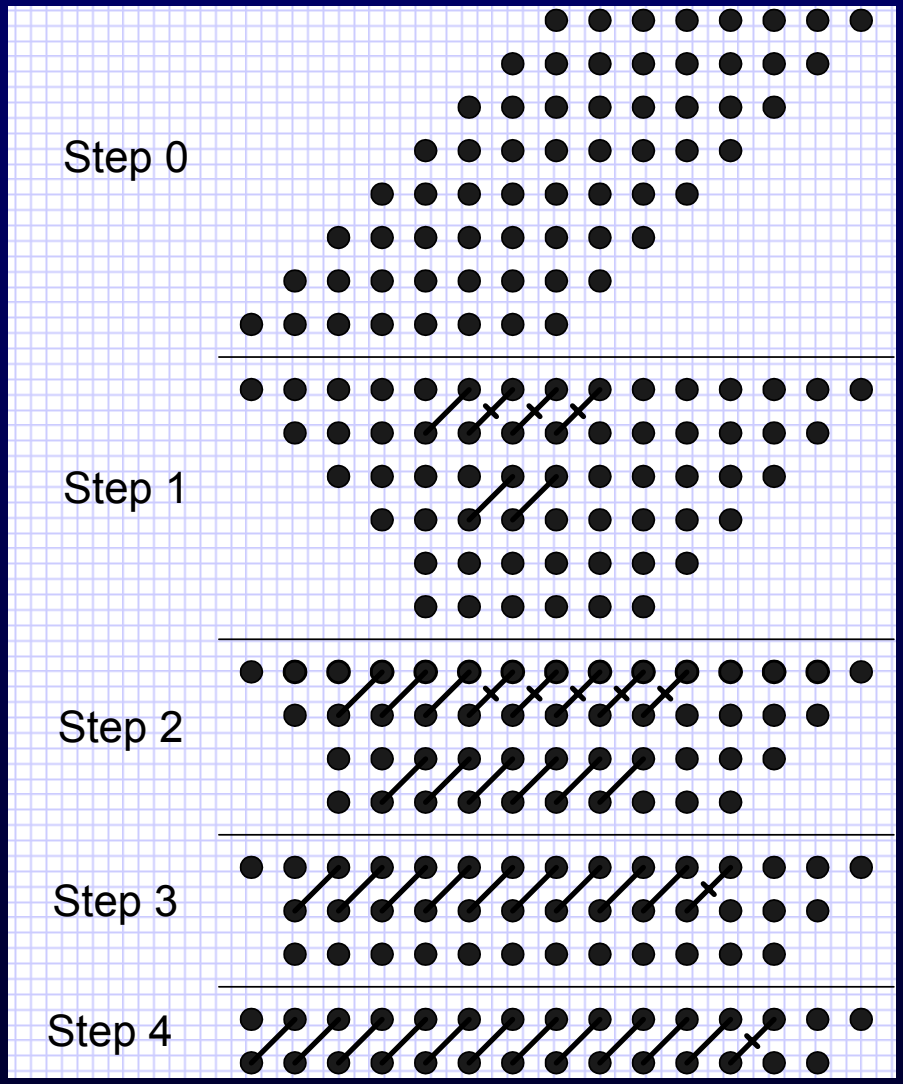
$$p(n) = XY \quad \textit{after } n \textit{ steps}$$



6 Bit Multiplication

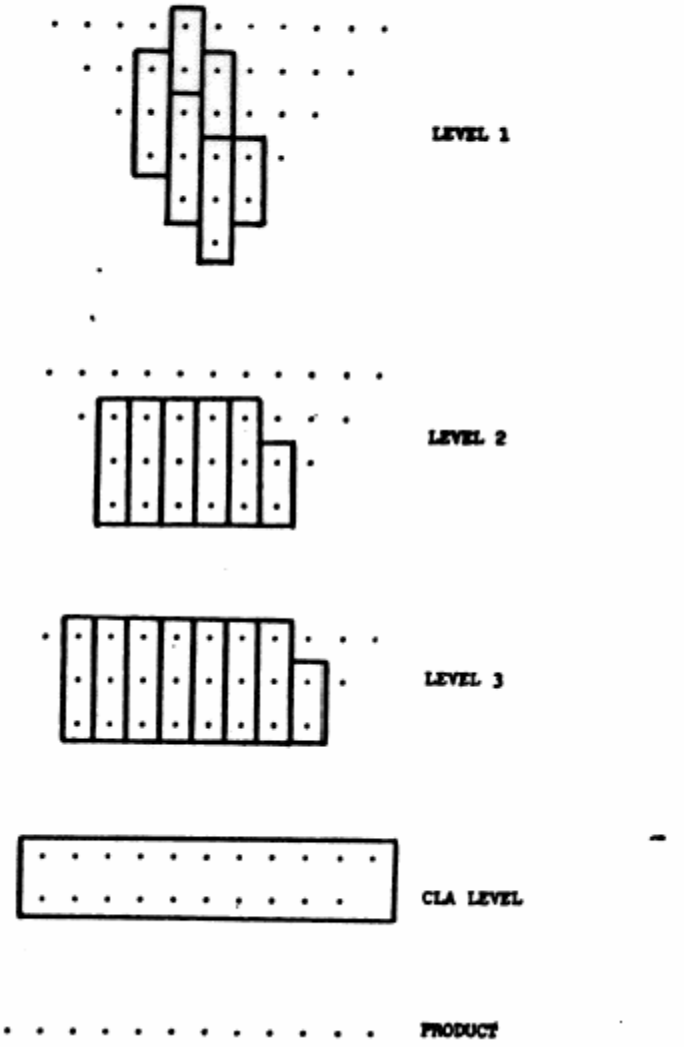

Multiplier
Multiplicand
Partial Product Generator
Compression Tree Using
Any Redundant Form
Translator To Conventional Form
Result

Parallel Multipliers



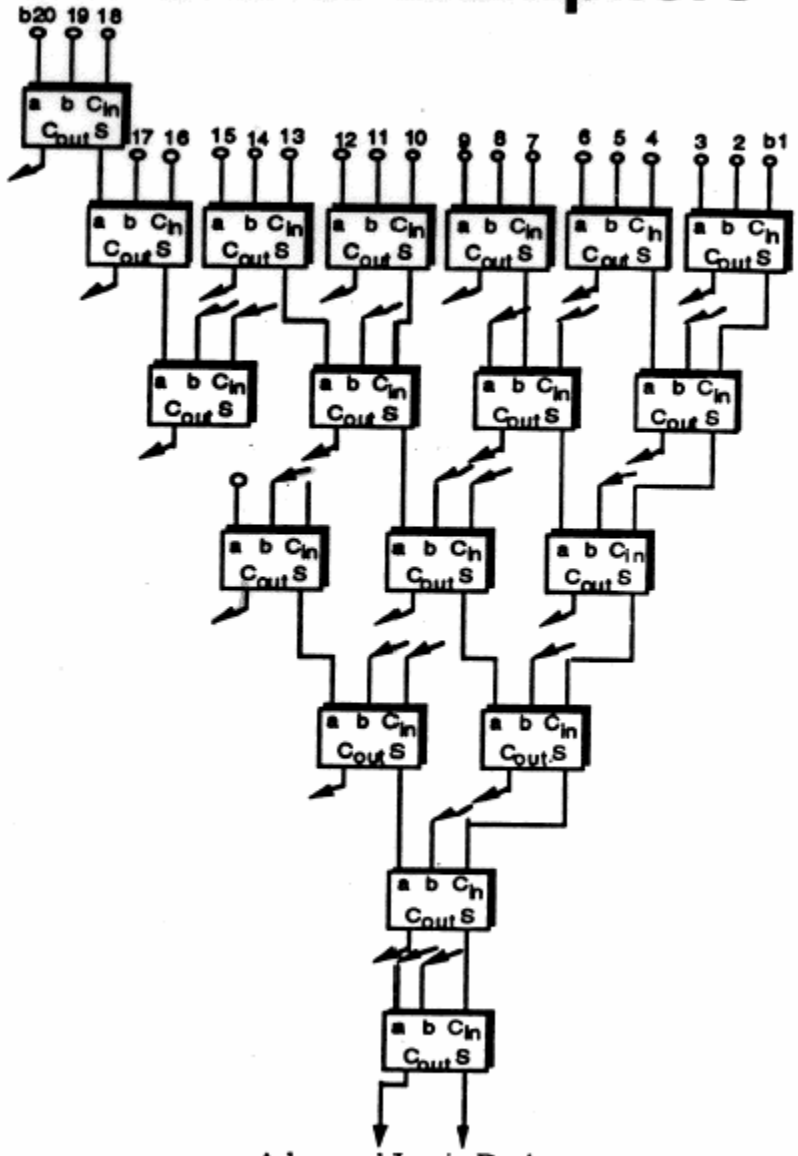
Fast Parallel Multipliers

Wallace:

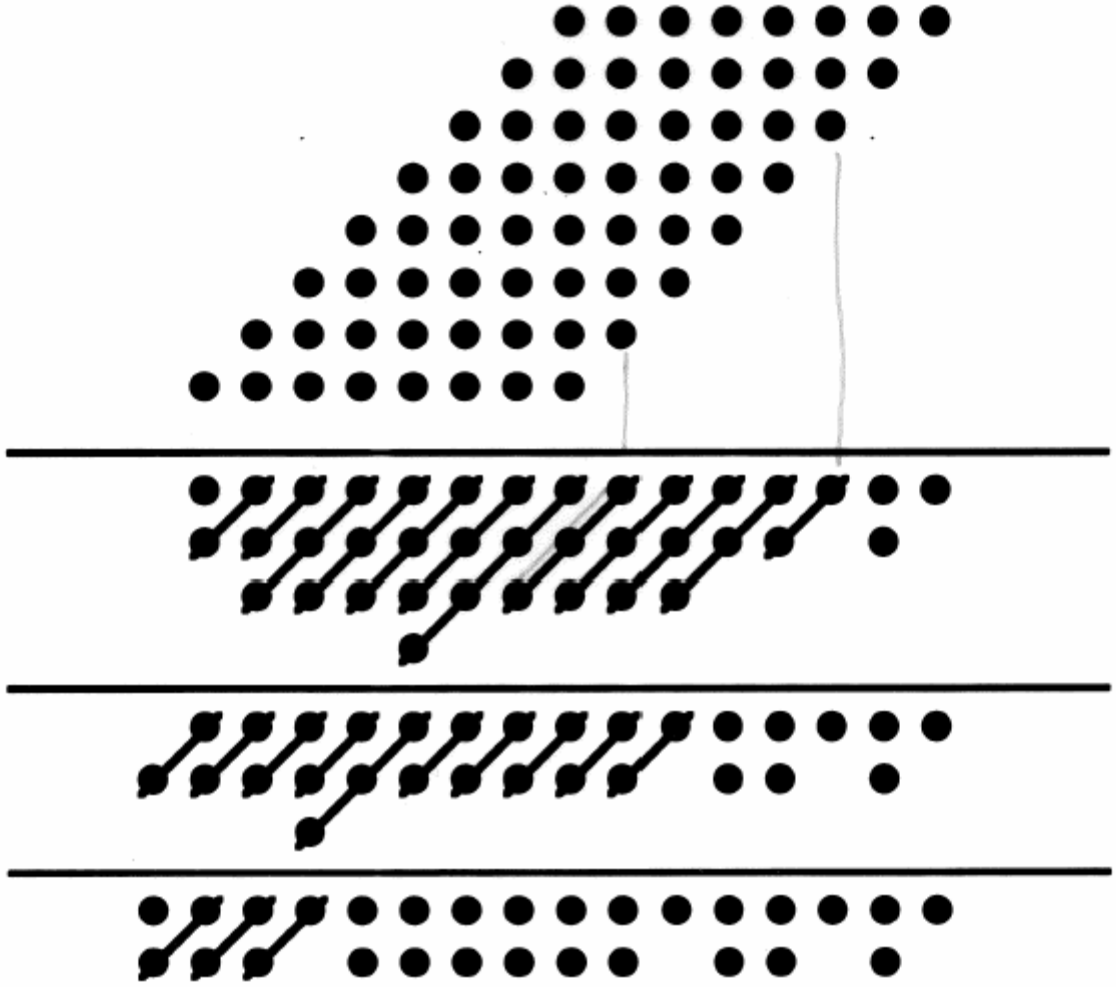


Fast Parallel Multipliers

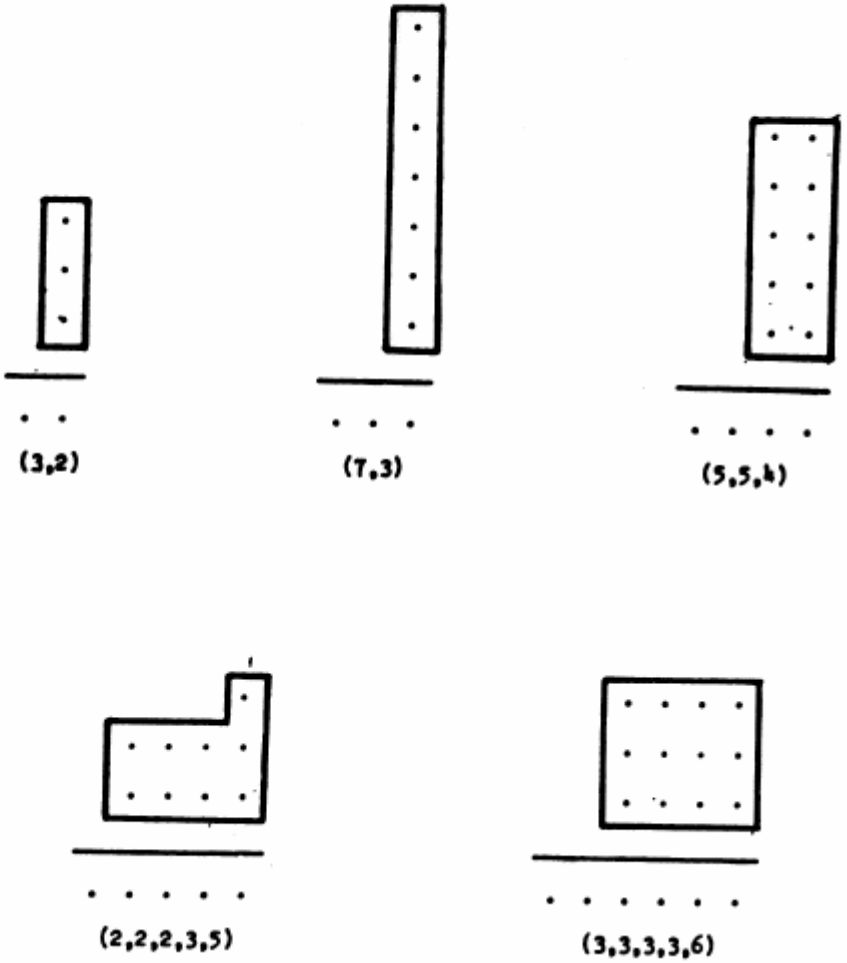
Wallace:



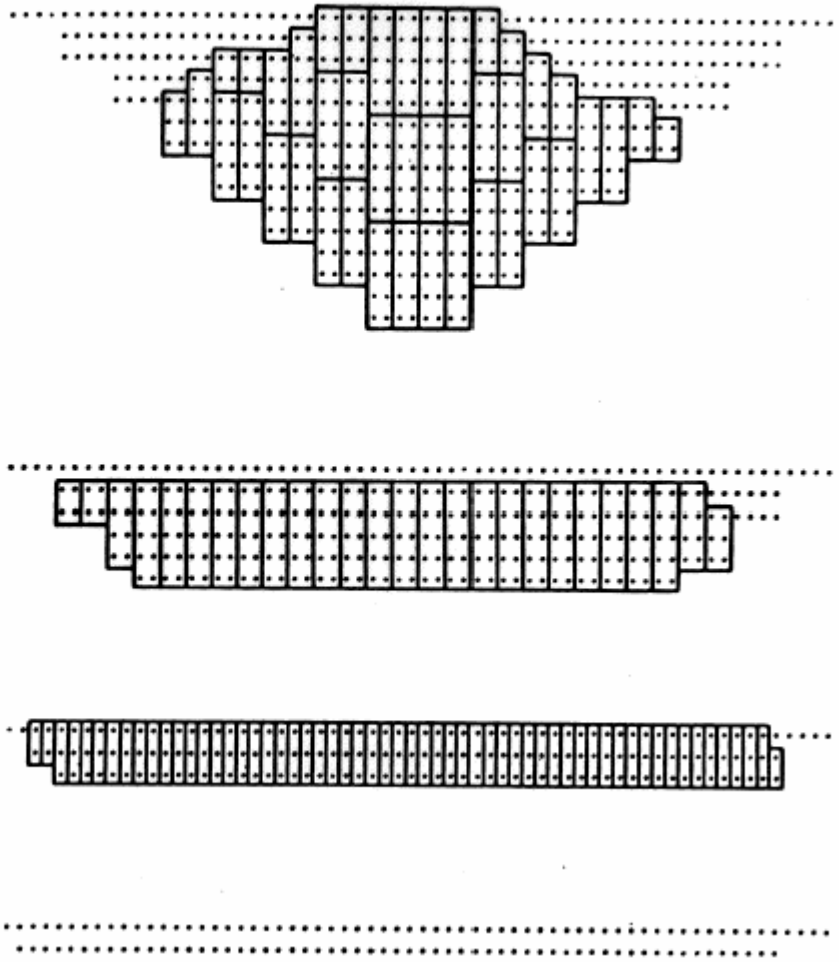
Bit Reduction Using "Dadda" Counters



Generalized Counters (Stenzel):



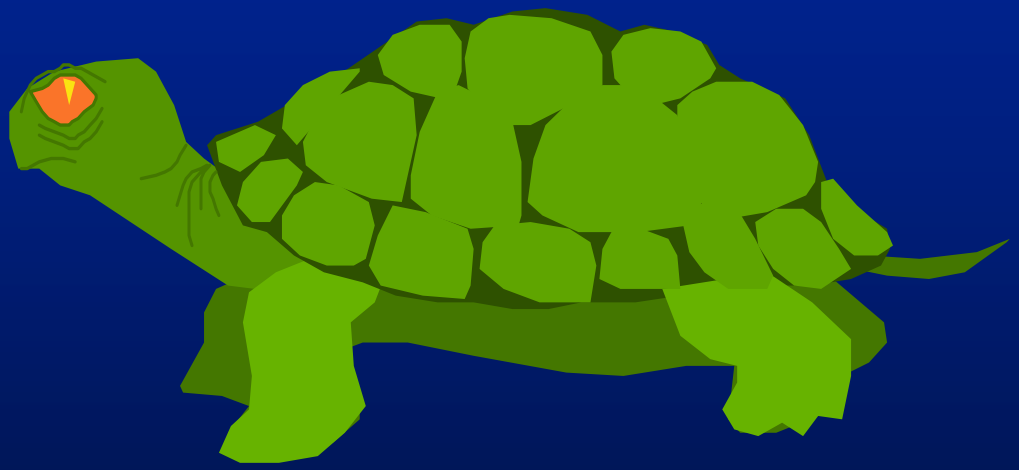
Generalized Counters (Stenzel):



Minimum Number of Stages (Dada's Rule)

Number of bits in the multiplier	Minimum number of stages
3	1
4	2
$4 < n \leq 6$	3
$6 < n \leq 9$	4
$9 < n \leq 13$	5
$13 < n \leq 19$	6
$19 < n \leq 28$	7
$28 < n \leq 42$	8
$42 < n \leq 63$	9

Their Schemes

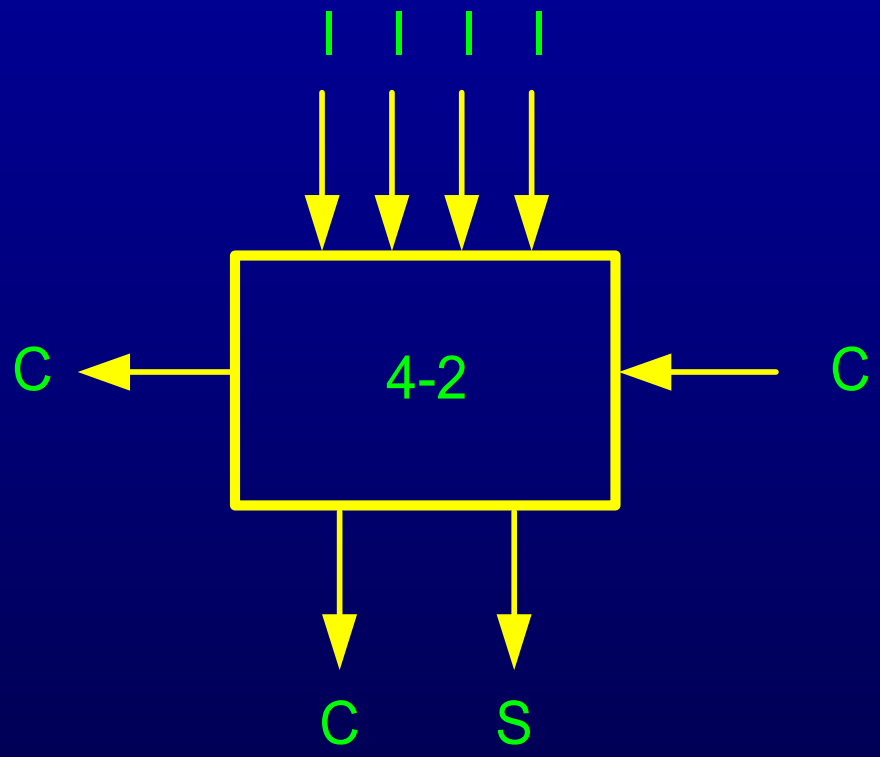


Use of 4:2 Compressors

A. Weinberger 1981

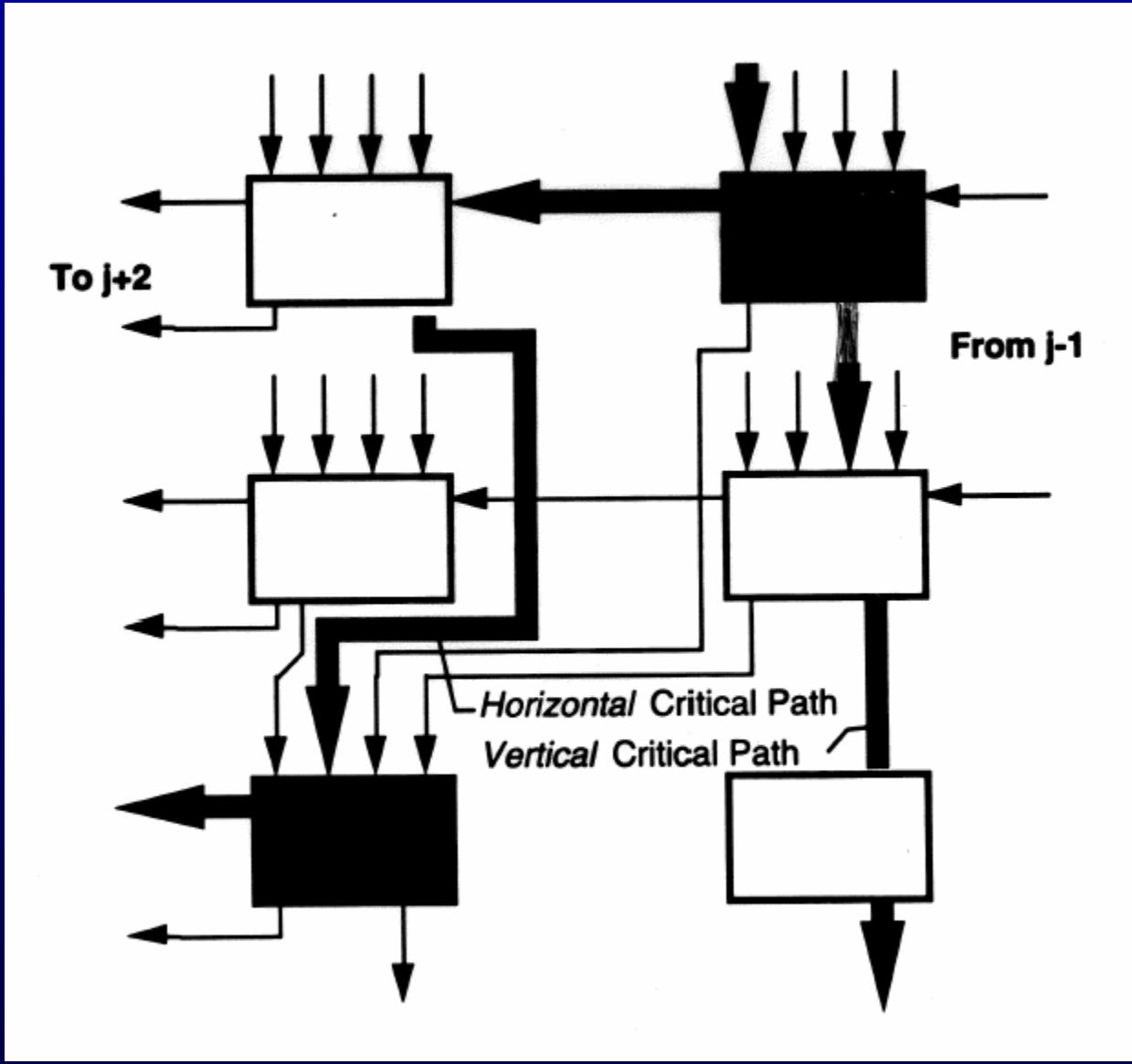
M. Santoro 1988

4:2 Compressor

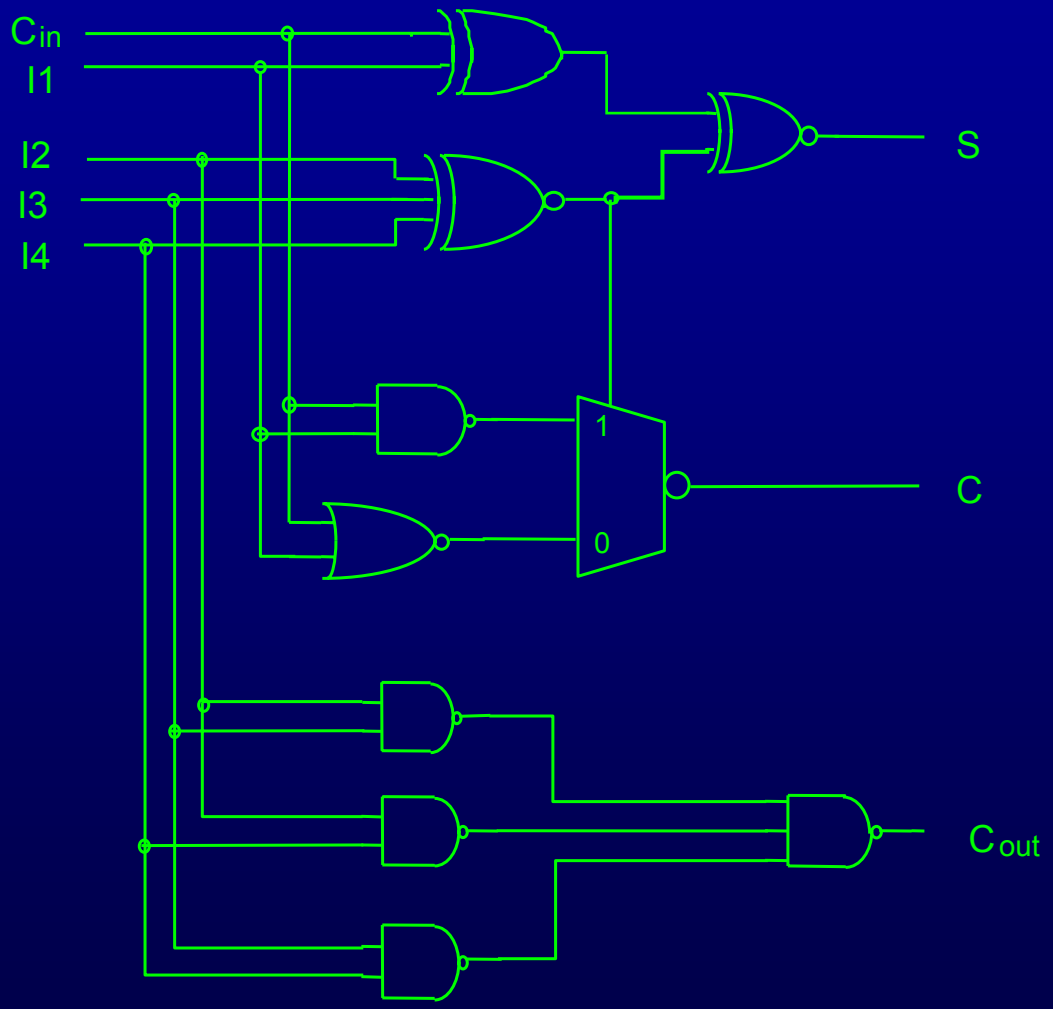


4

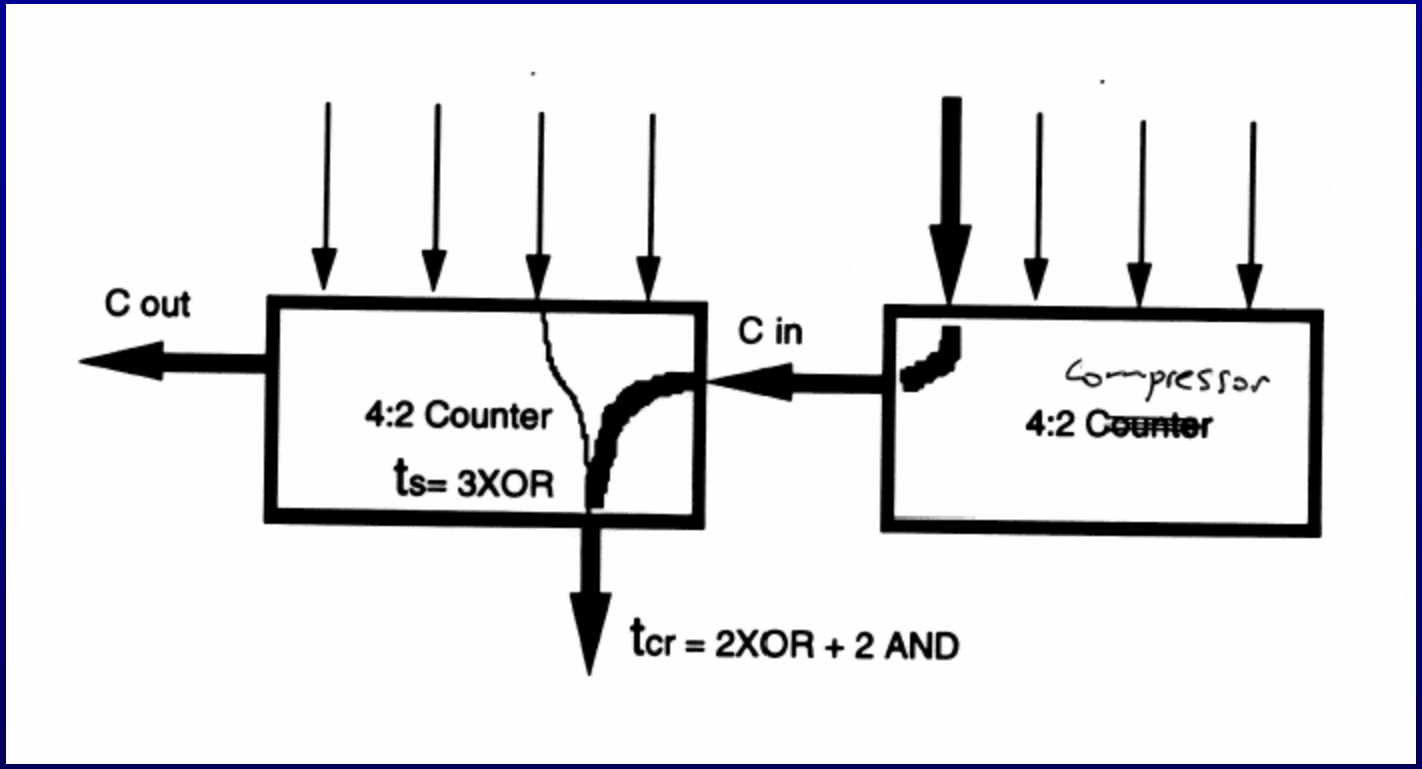
Critical Signal Path in a 4:2 Compressor Tree



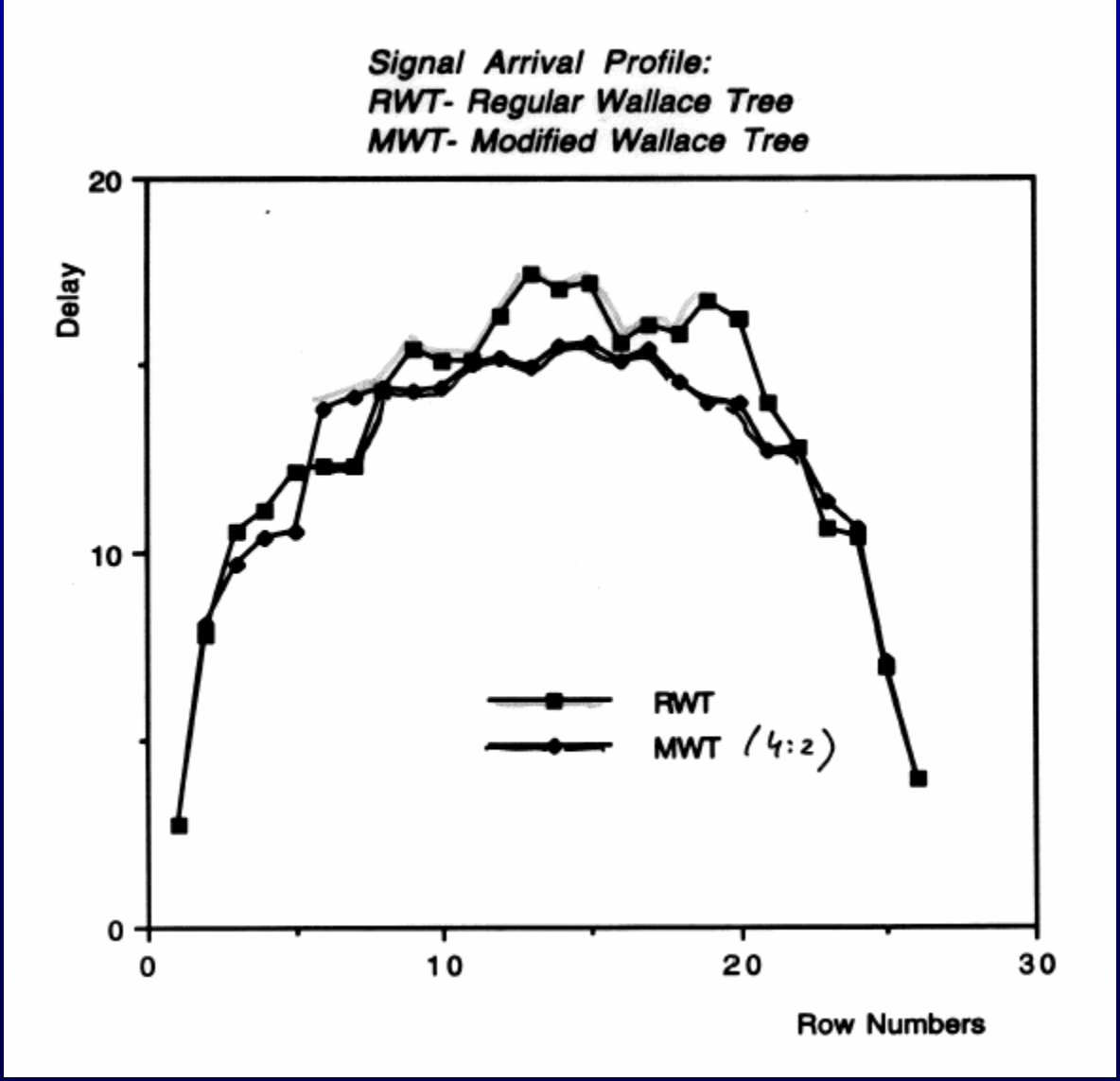
Re-designed 4:2 Compressor with 3 XOR Delay (Nagamatsu, Toshiba)



Critical Path in a 4:2 Compressor



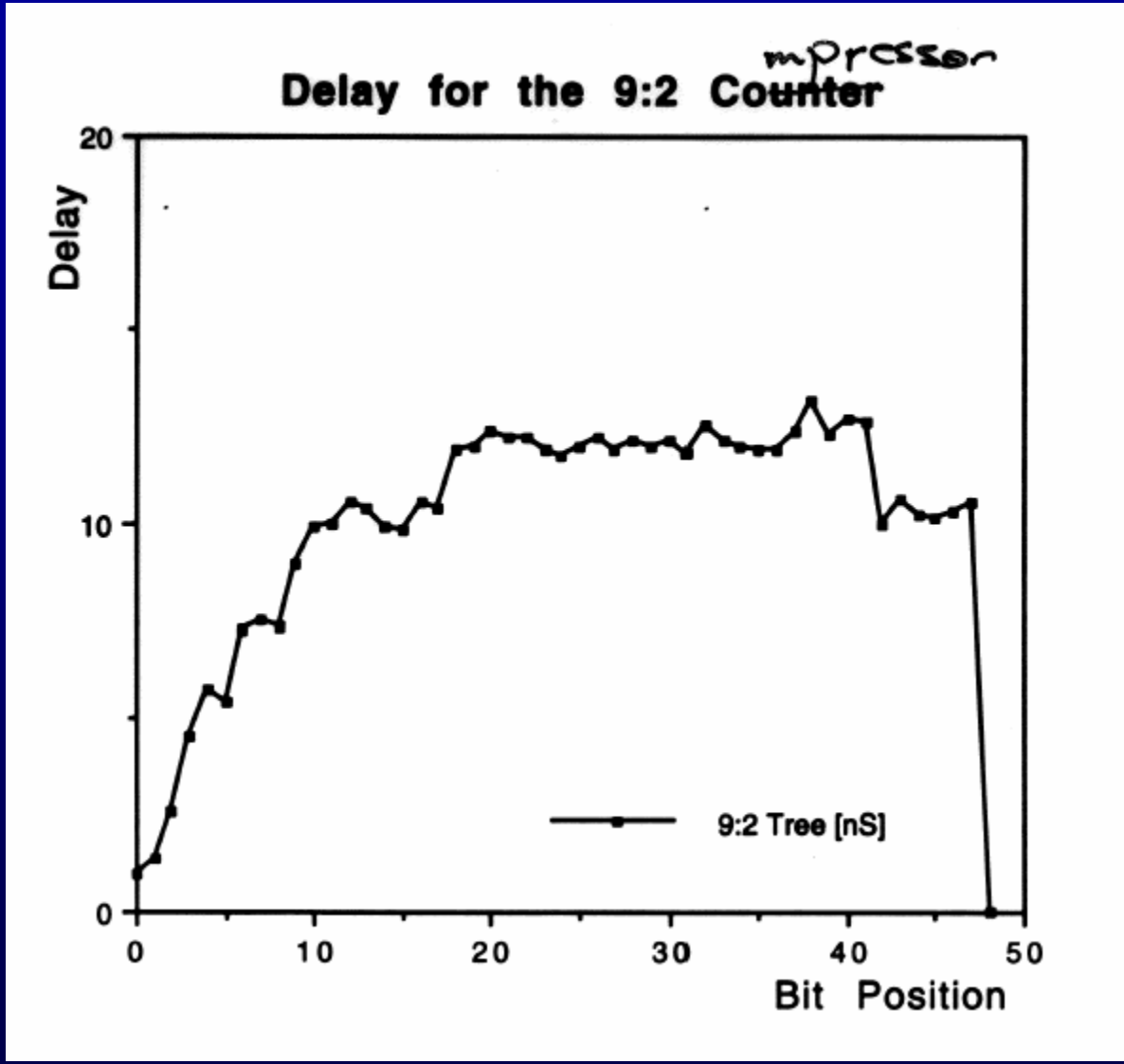
Signal Arrival Profile for RWT (3:2) and MWT (4:2)



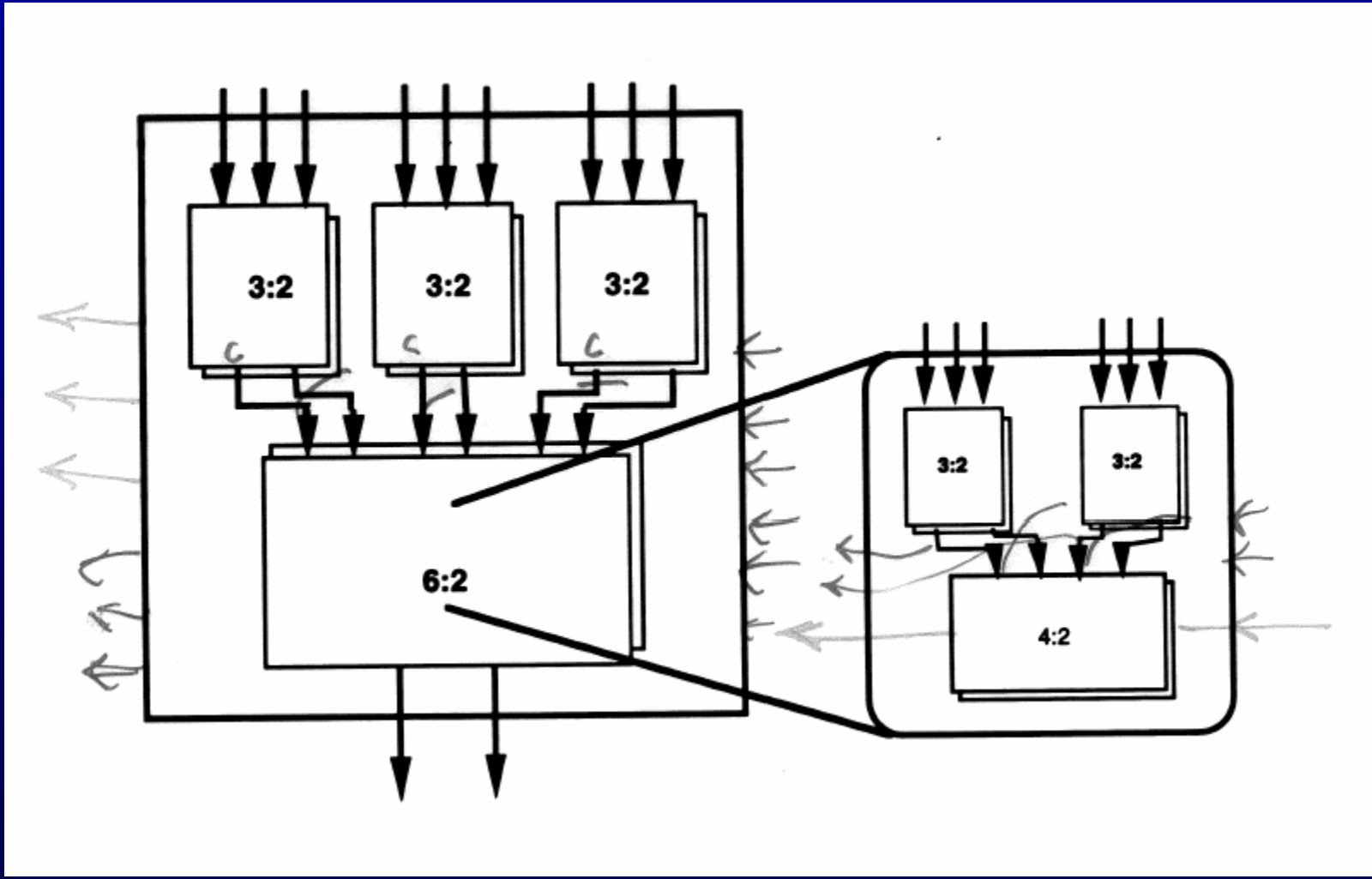
Using 9:2 Compressors

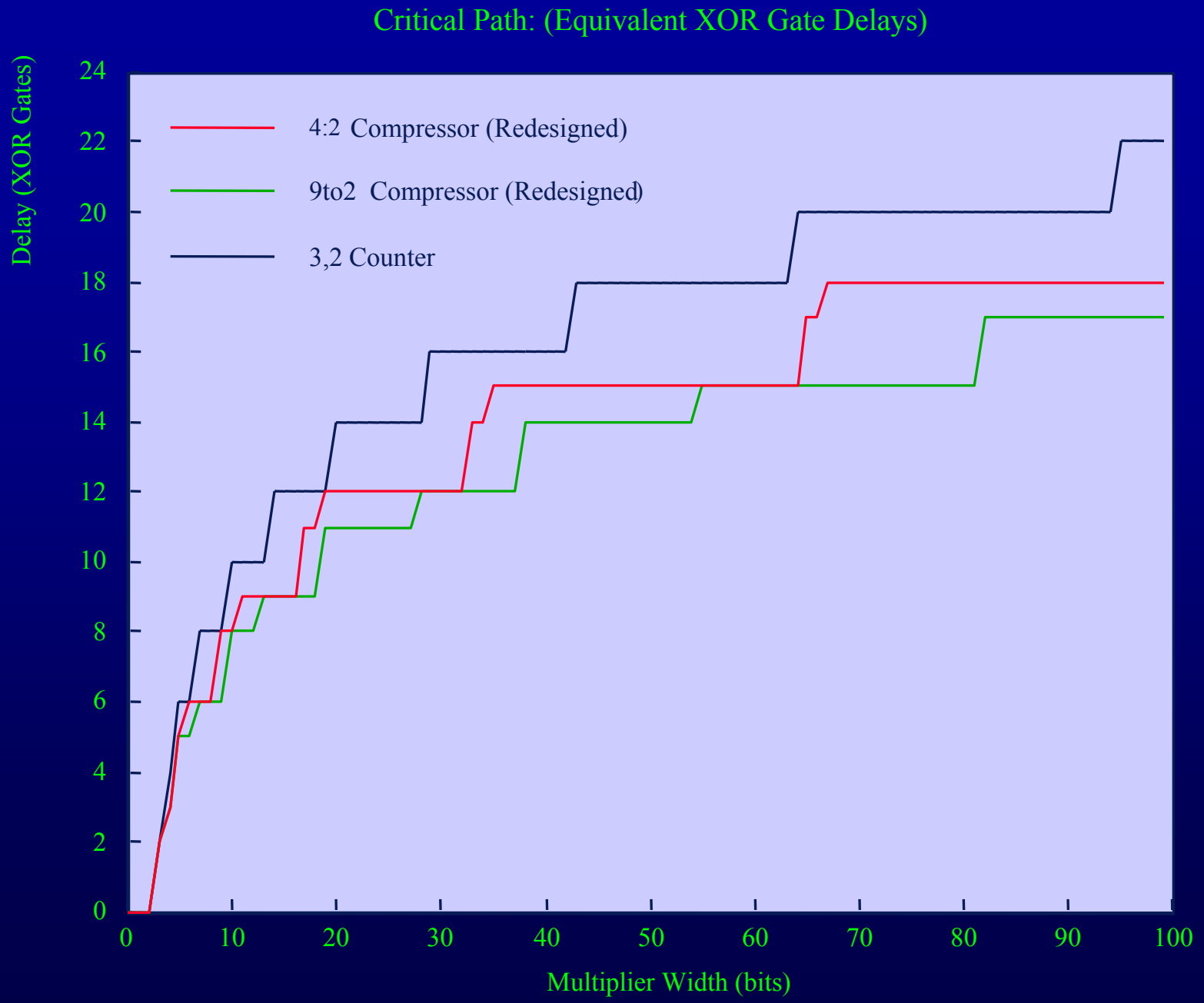
(P. Song, G. De Michelli 1991)

Compressor Tree Implemented with 9:2 Compressors

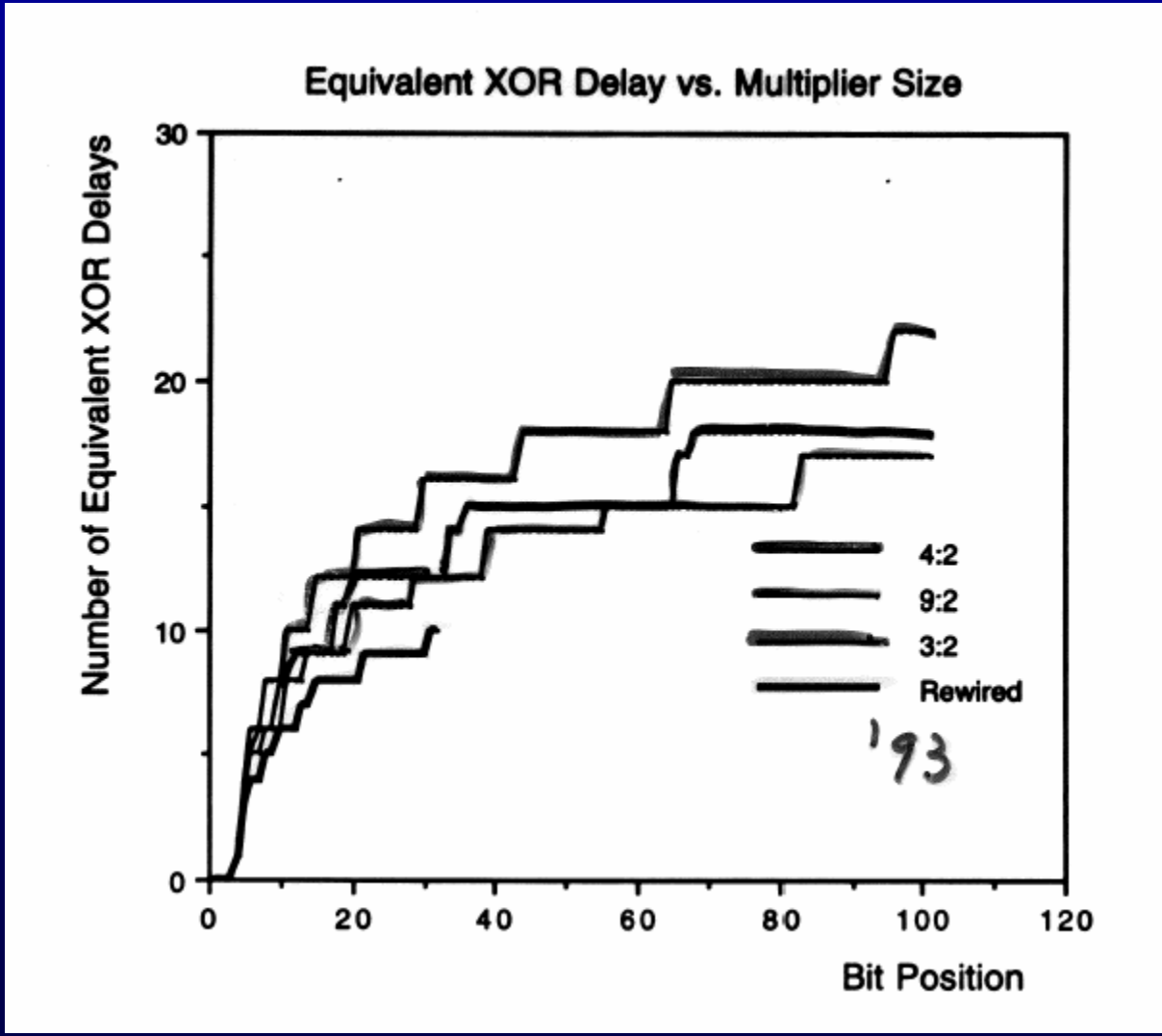


9:2 Compressor Structure




Critical Path: (Equivalent XOR Gate Delays)
Delay (XOR Gates)
24
22
20
18
16
14
12
10
8
6
4
2
0
4:2 Compressor (Redesigned)
9to2 Compressor (Redesigned)
3,2 Counter
0
10
20
30
40
50
60
70
80
90
100
Multiplier Width (bits)

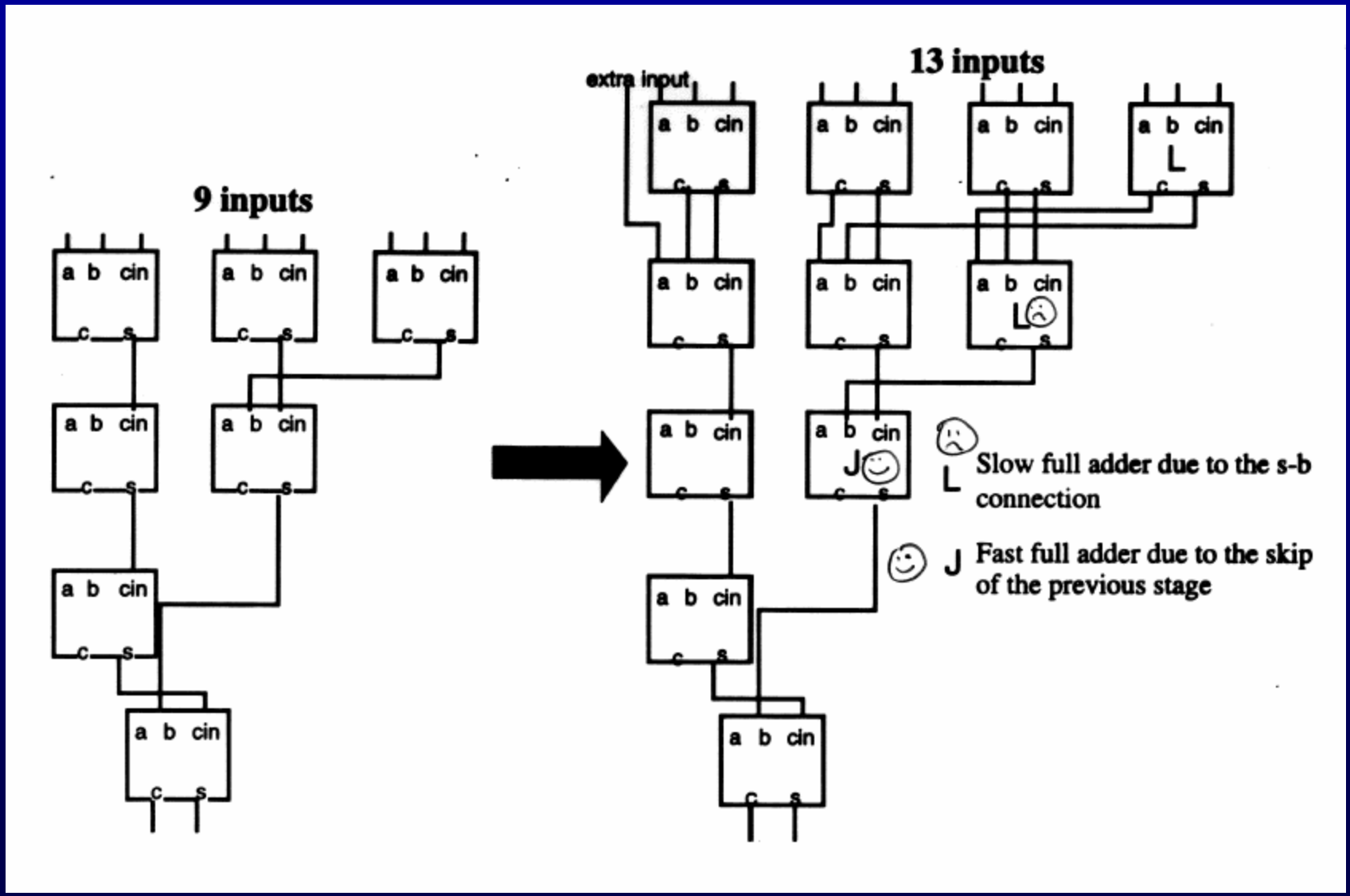
Delay Expressed as No. of XOR Gate Delays



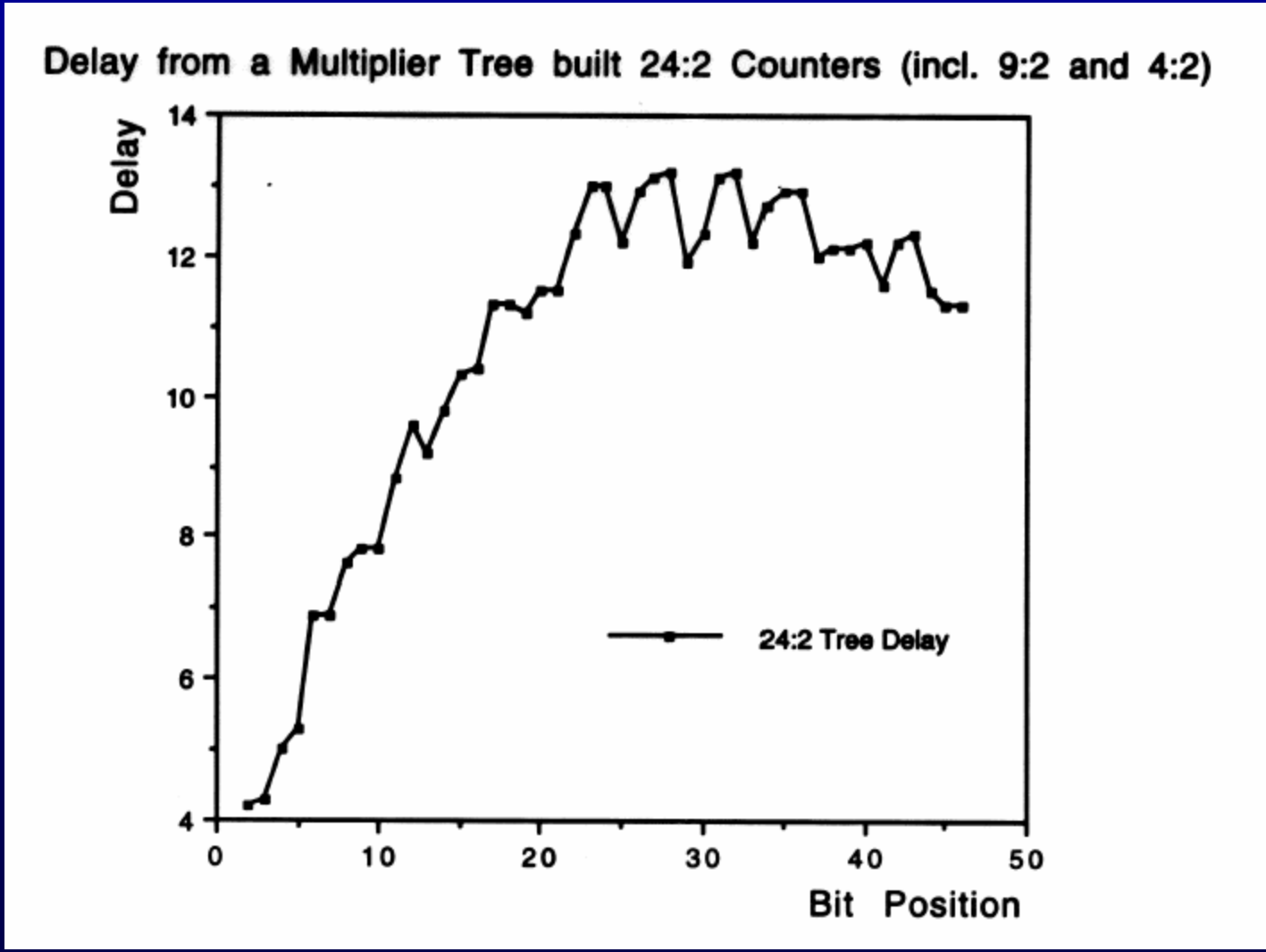
Use of Higher-Order Compressors

D. Villeger, V.G. Oklobdzija 1993

Design of a 13:2 Compressor from a 9:2 Compressor



Delay Profile of a 24:2 Compressor Tree



Compressor Family Characteristics

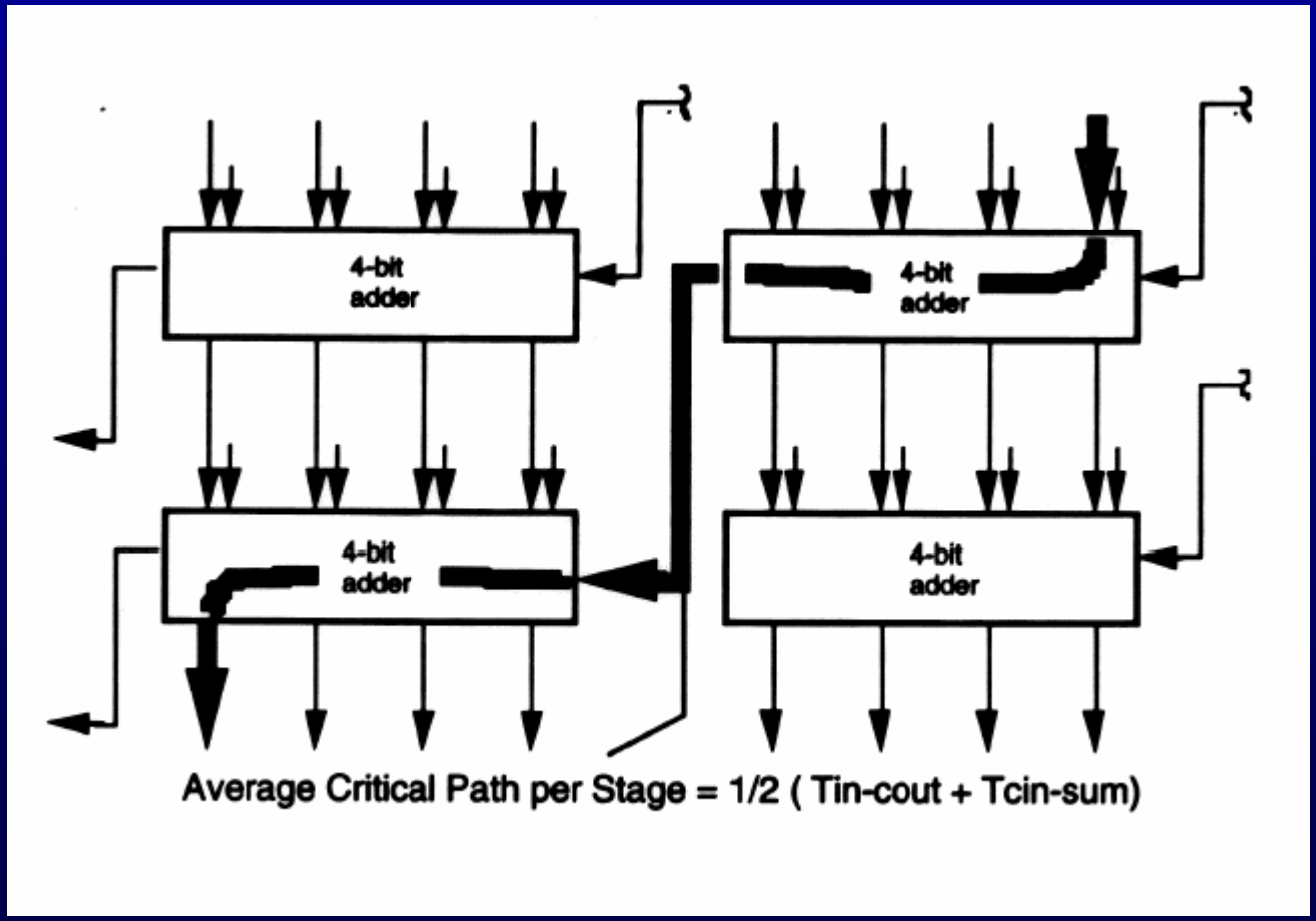
Compressor Counters	No of Full Adder Levels	No XOR Gates
4-2	2	3
6-2	3	5
9-2	4	6
13-2	5	8
18-2	6	9
24-2	7	11
53-2	9	14

Using Carry-Propagate Adders

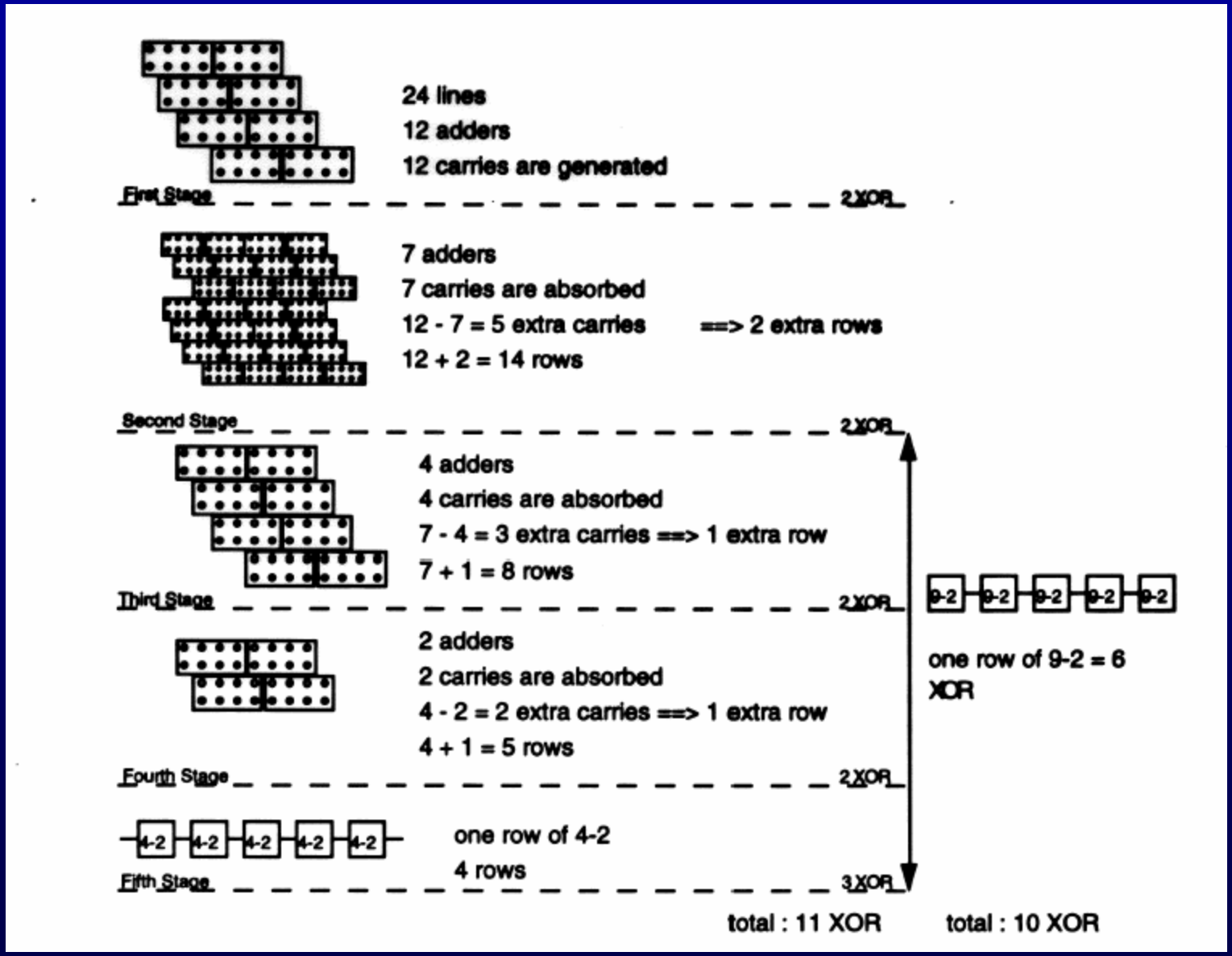
(G. Bewick 1993)

(D. Villeger & V. G. Oklobdzija 1993)

Column Compression Tree Consisting of 4-bit Adders



Bit Reduction Using 4-bit Adders (24X24)

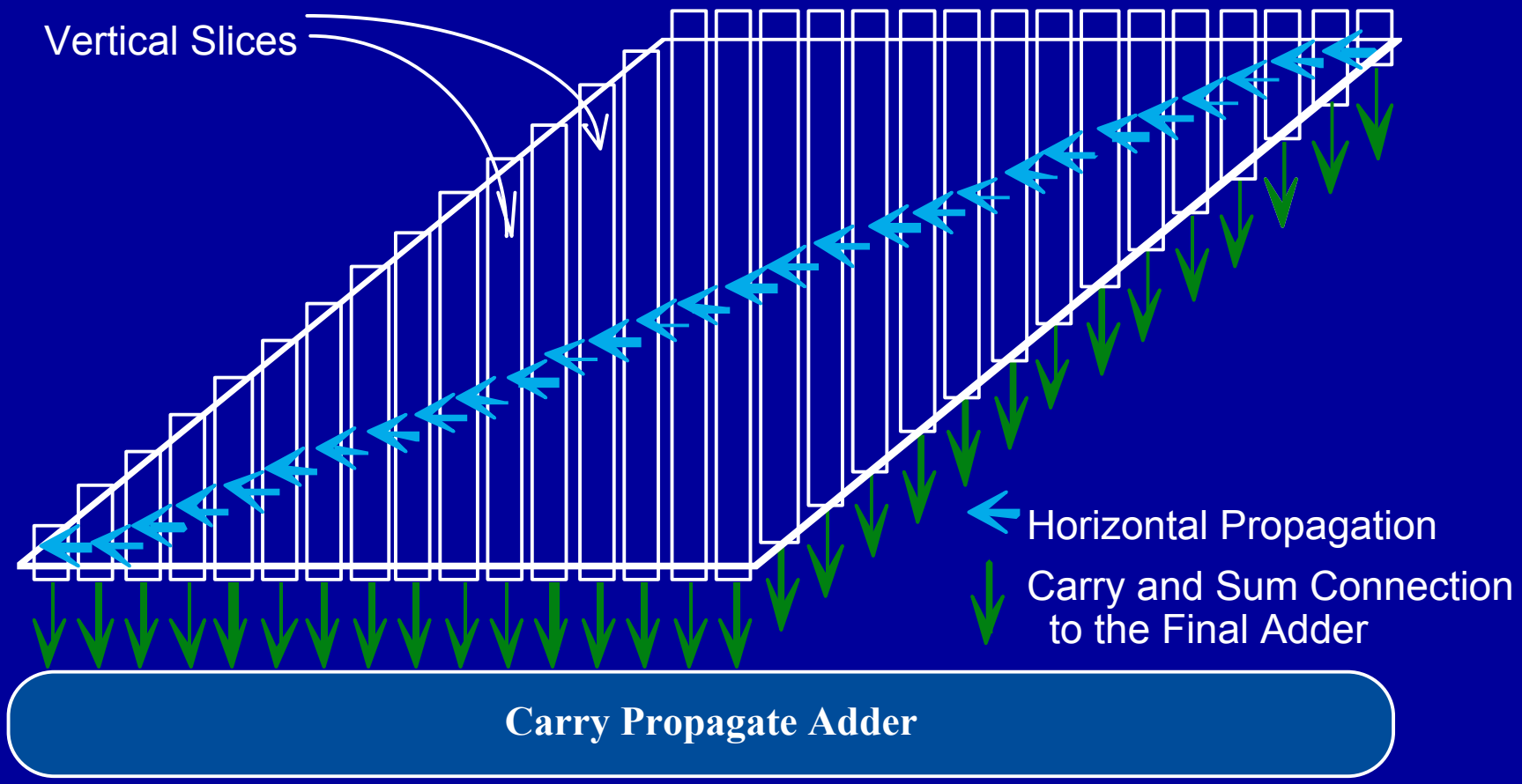



Idea !!!!!

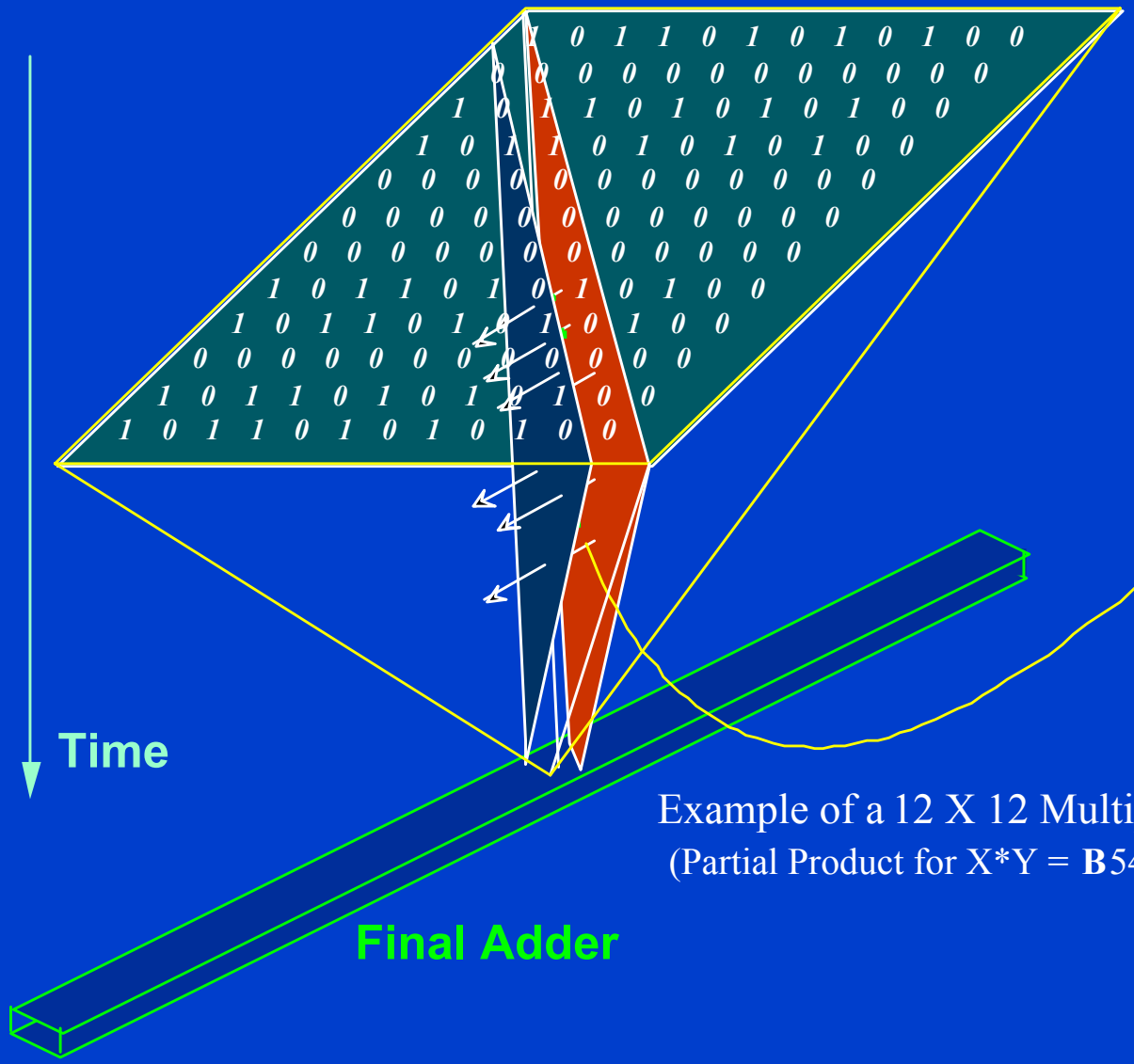
A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach – TDM

(Oklobdzija, Villeger, Liu, 1994)

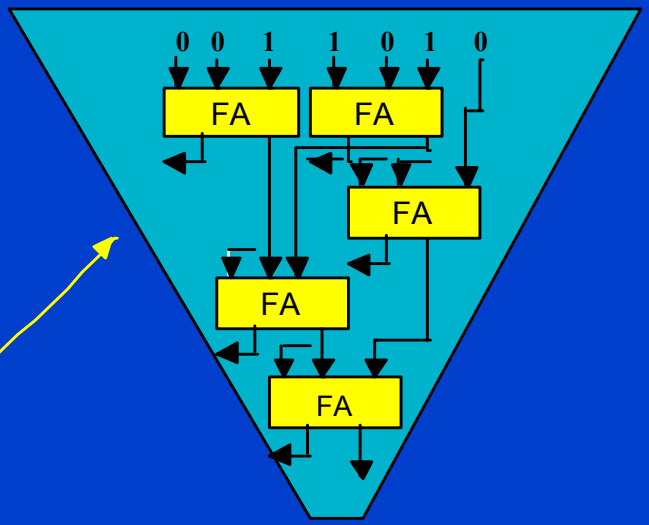
Partial Product Martix Divided into Vertical Compressor Slices



3-Dimensional View of Partial Product Reduction



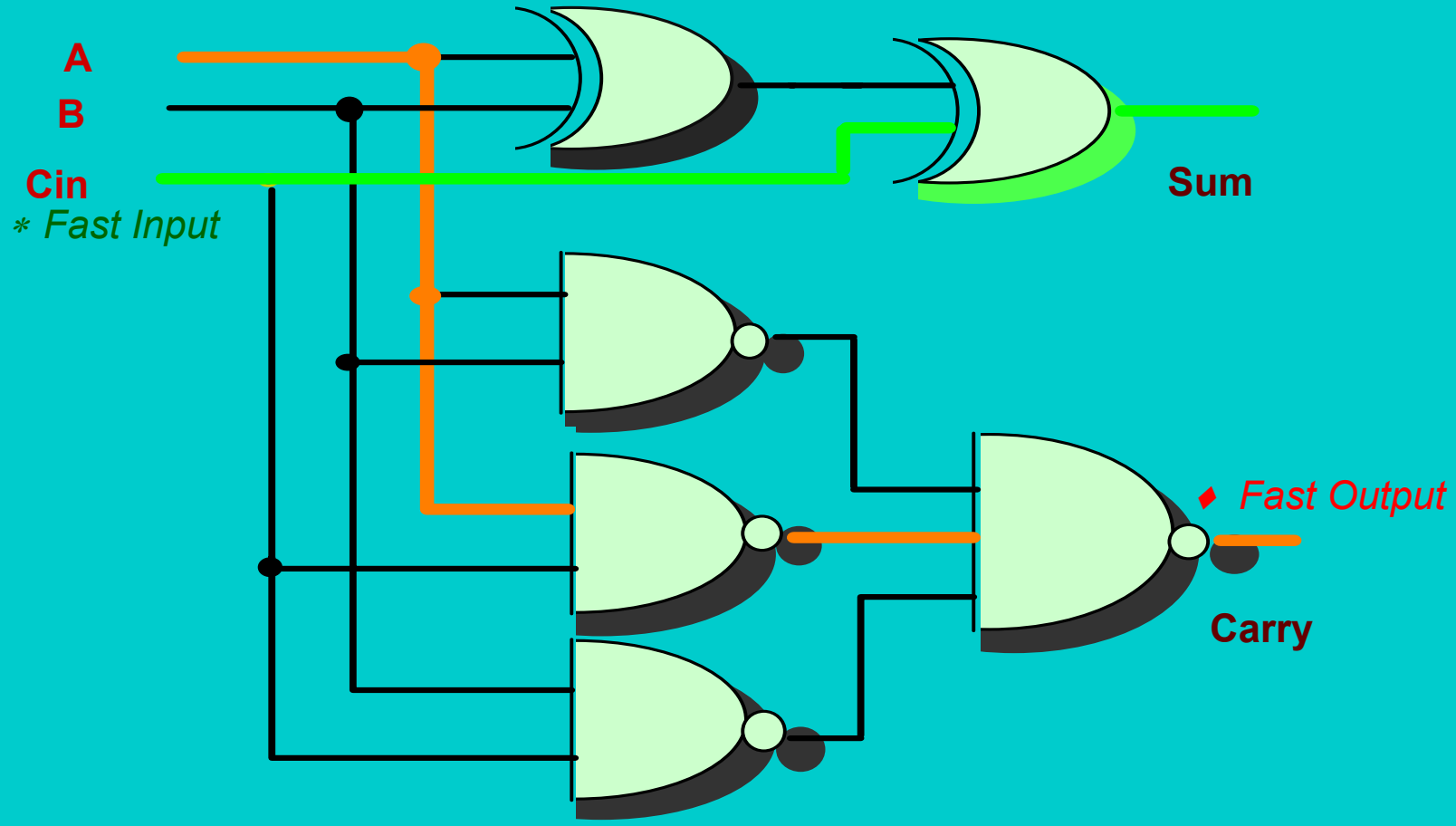
Vertical Compressor Slice - VCS



Example of a 12 X 12 Multiplication
(Partial Product for X*Y = **B**54 * **B**1**B**)

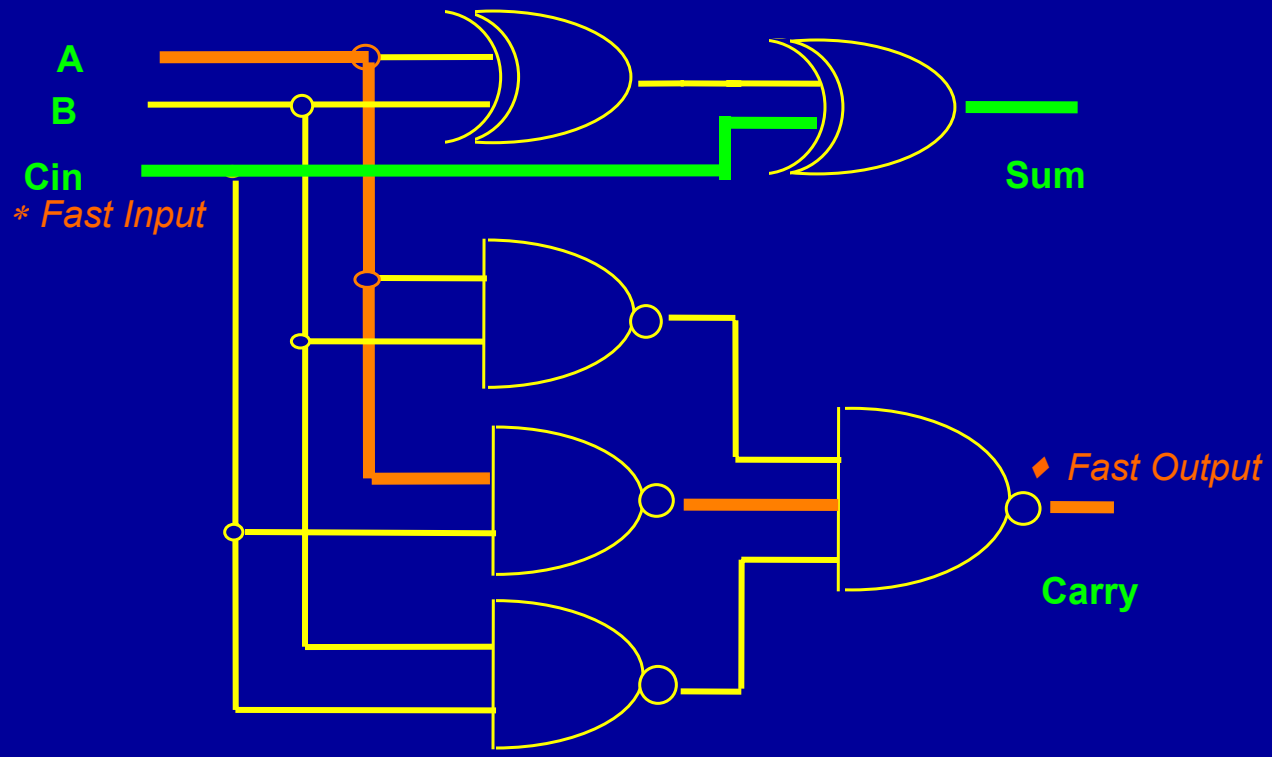
Signal Delays in a Full Adder

(3,2) Counter

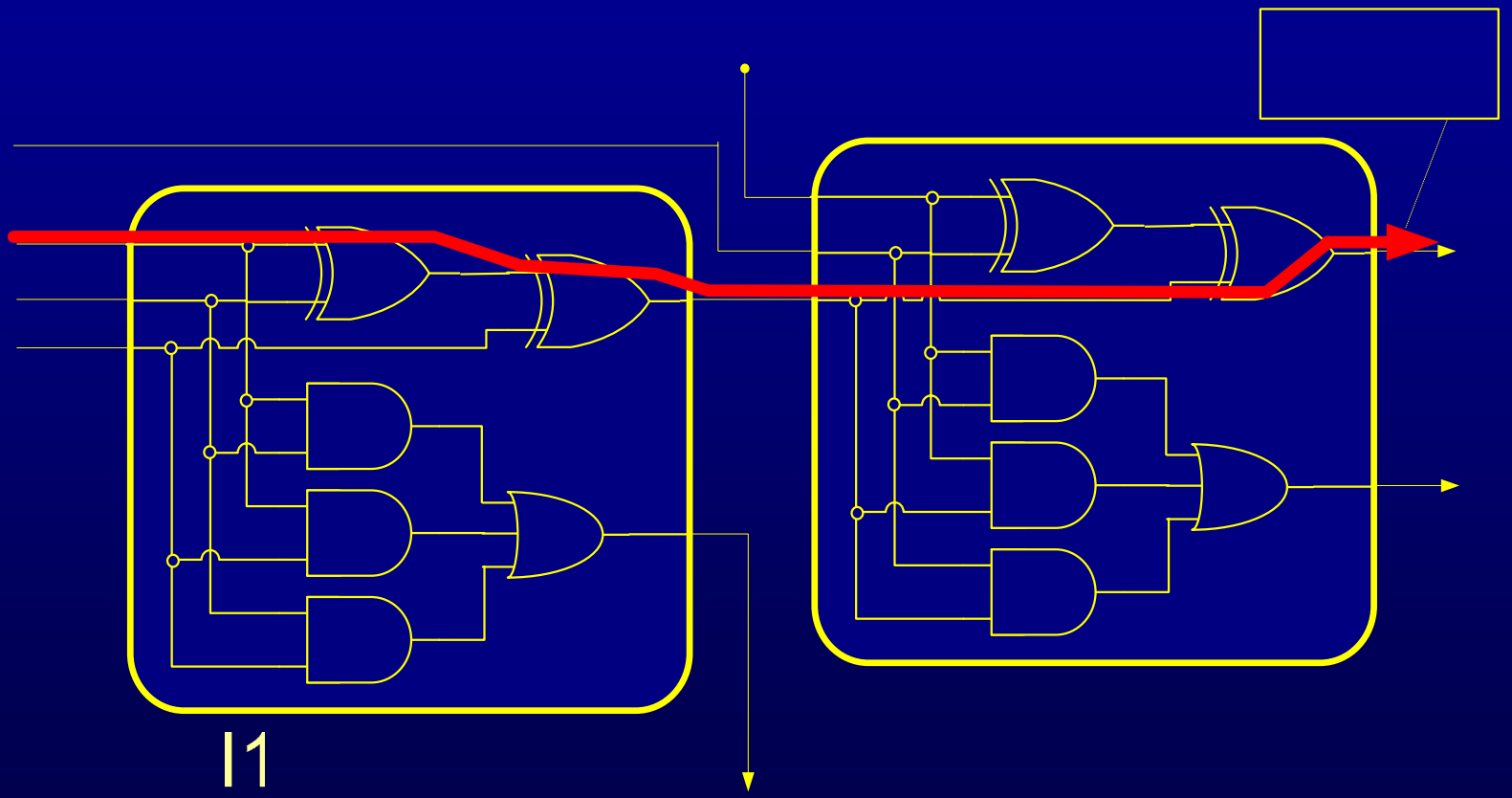


Signal Delays in a Full Adder

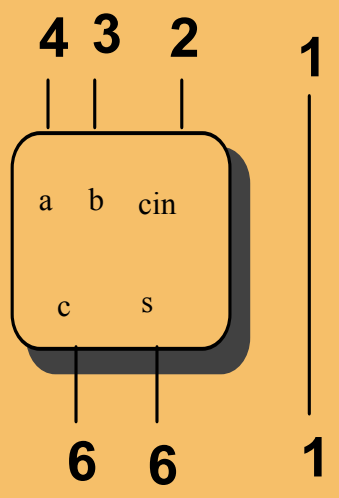
(3,2) Counter



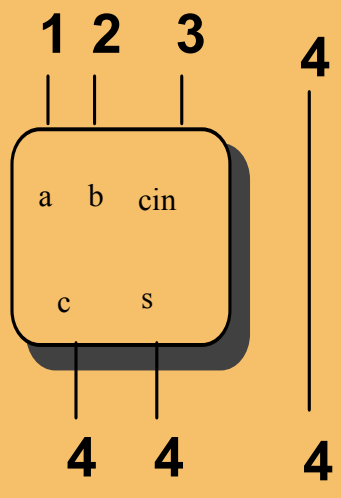
Three-Dimensional optimization Method: TDM (Oklobdzija, Villeger, Liu, 1996)



Method

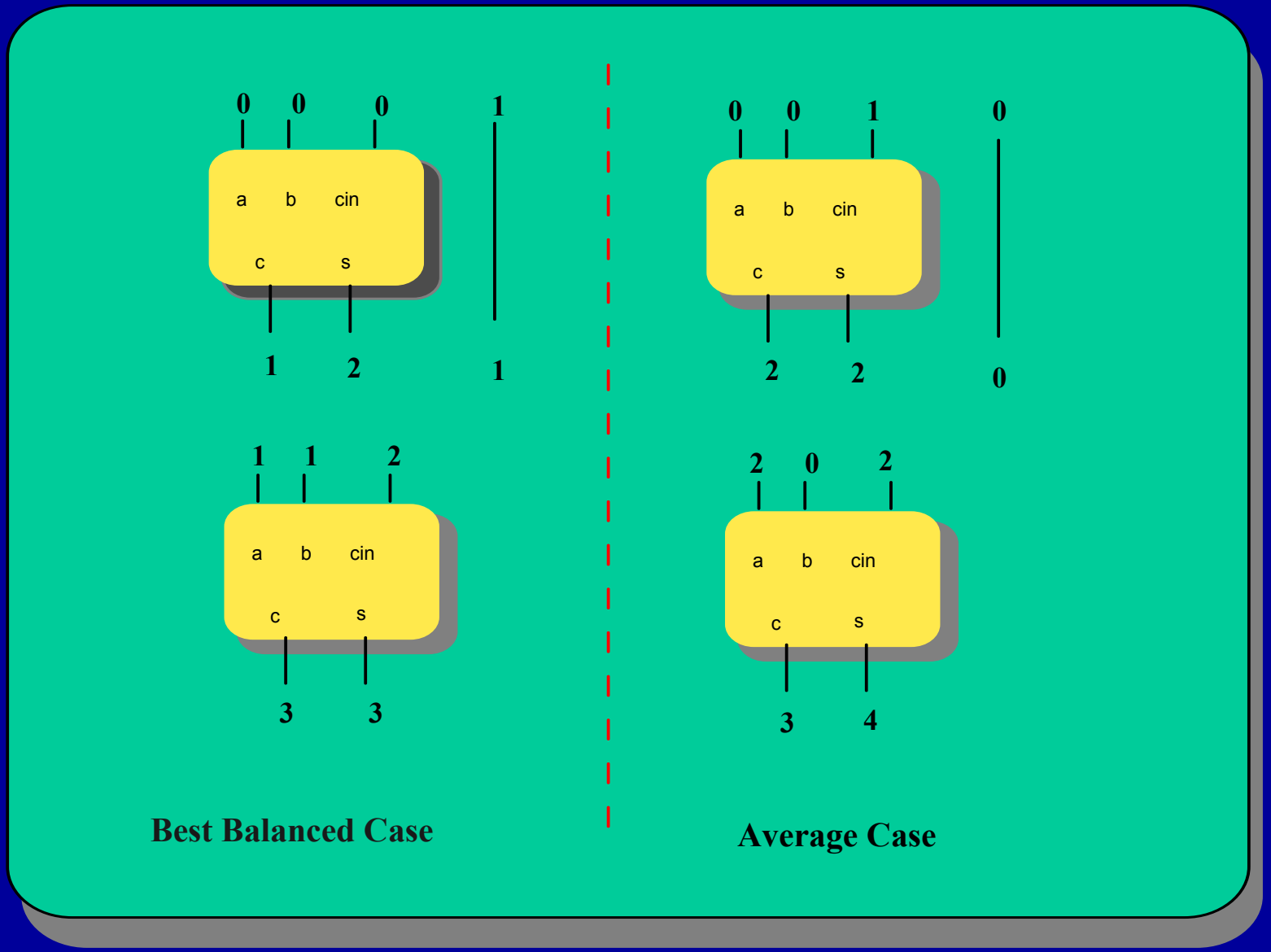


Worst Case



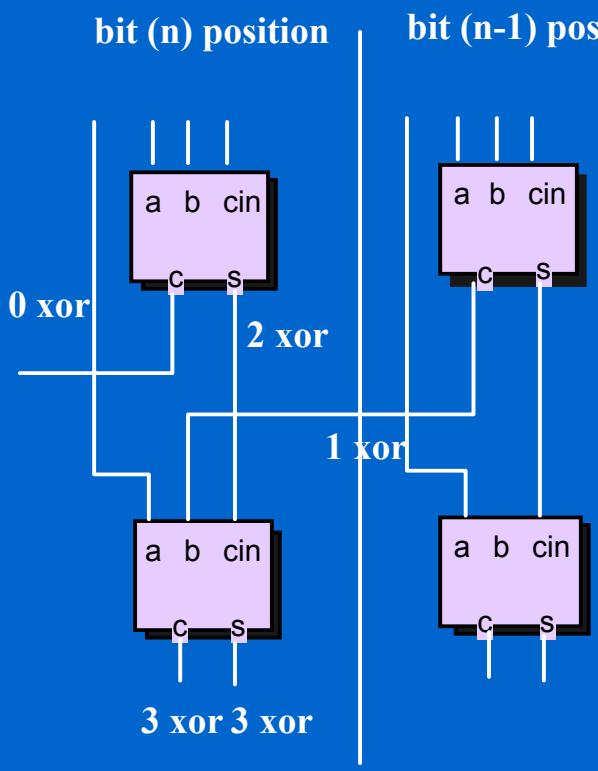
TDM Arrangement

Two cases of signals passing through the next level

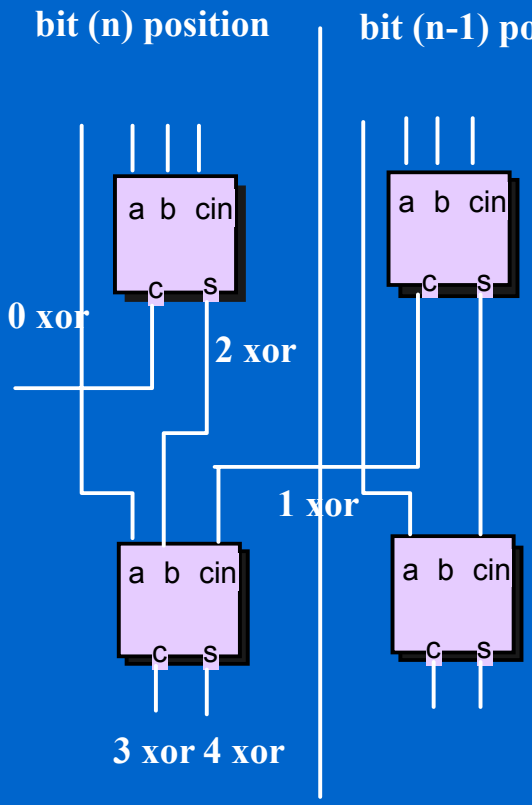


Example of Delay Optimization

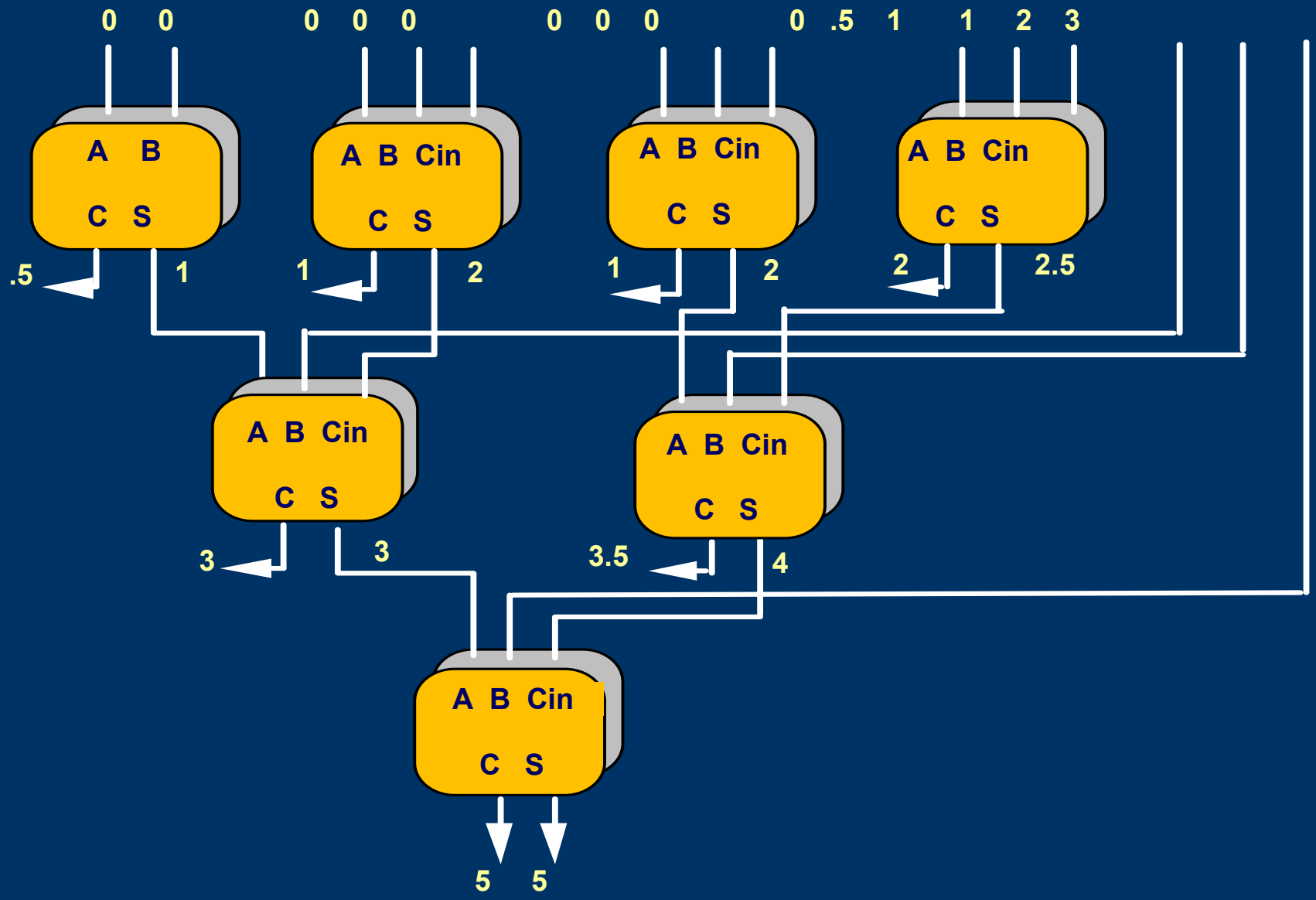
Example of a Optimized Interconnection

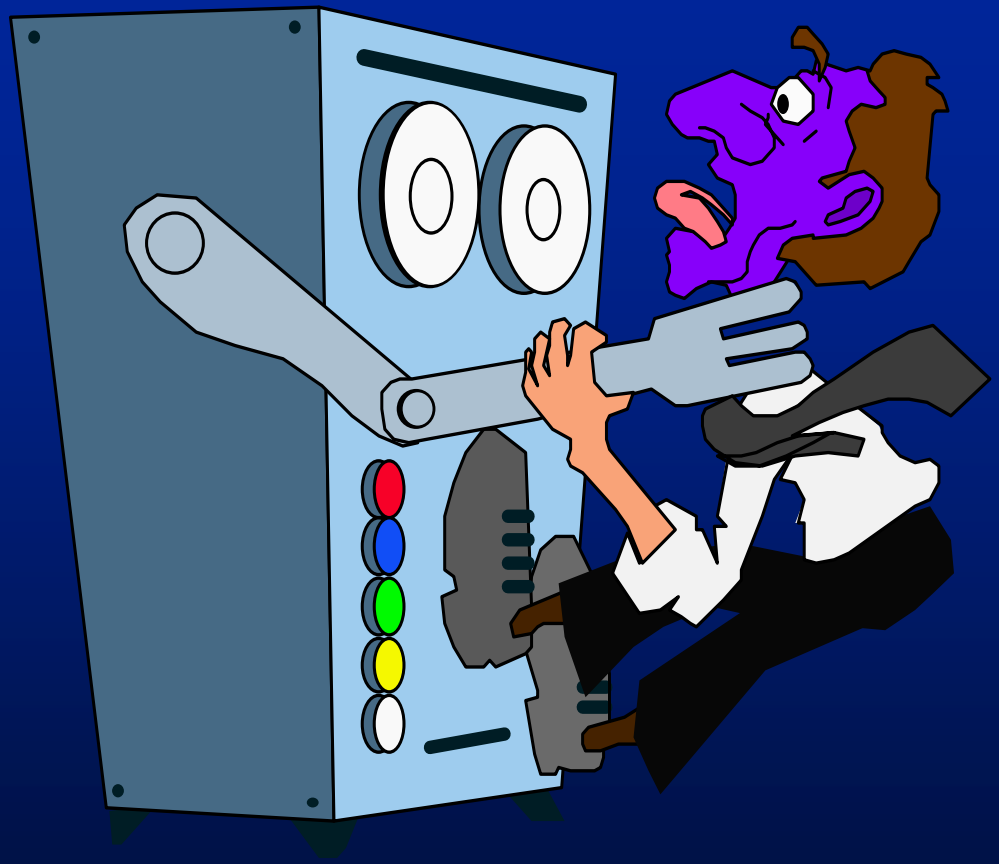


Example of a not Optimized Interconnection



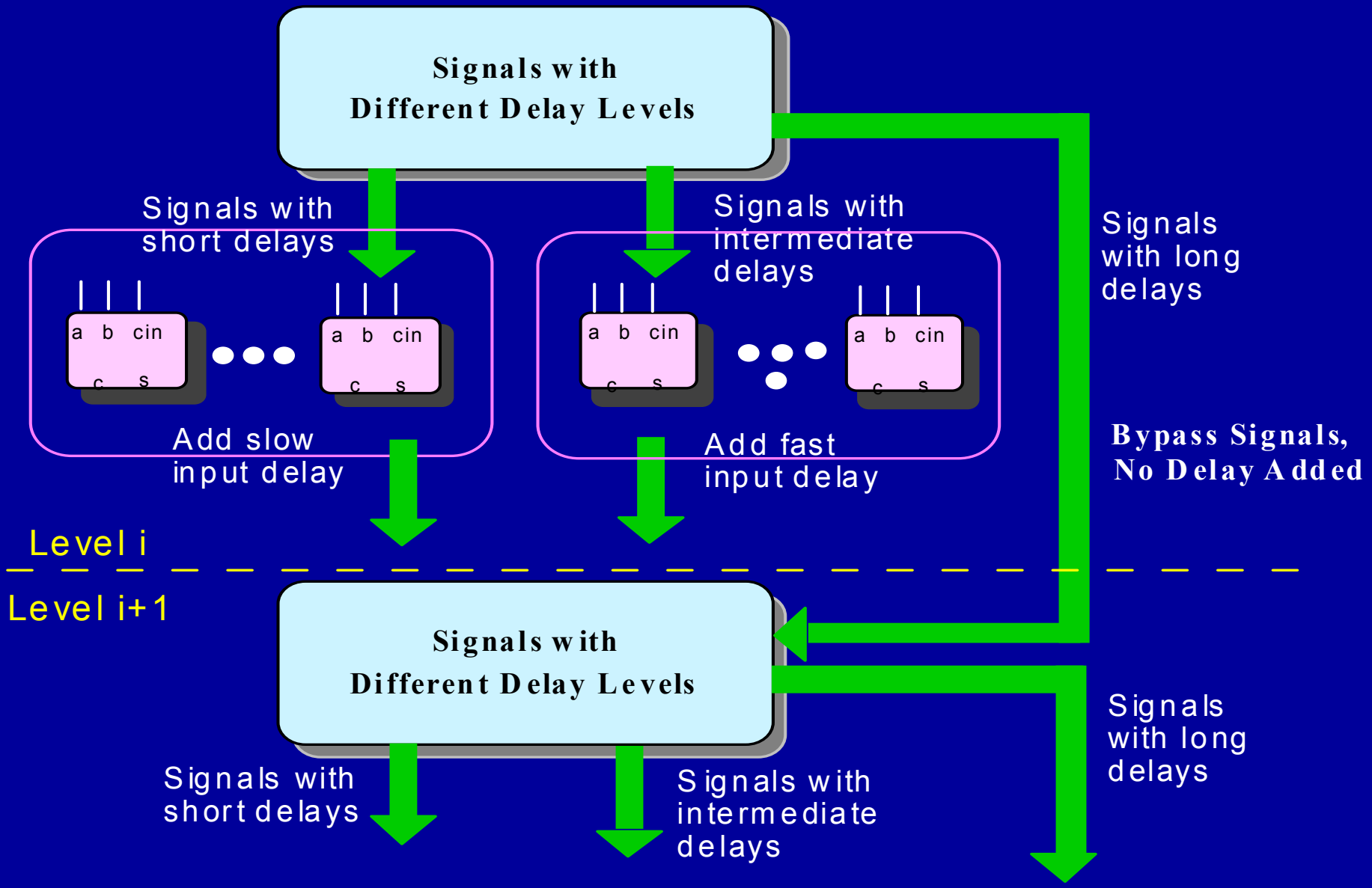
The 9th Vertical Compressor Slice of a Multiplier





Computer Tools

Method for Optimal Interconnection



Design of Parallel Multipliers

Algorithm for Automatic Generation of Partial Product Array.

Initialize:

Form 2N-1 lists Li (i = 0, 2N-2) each consisting of p_i elements where:

$p_i = i+1$ for $i \leq N-1$ and $p_i = 2N-1-i$ for $i \geq N$

An element of a list L_i ($j = 0,\ldots,p_{i-1}$) is a pair: $\langle n_j, D_j \rangle i$ where:

n_j : is a unique node identifying name

D_j : is a delay associated with that node representing a delay of a signal arriving to the node nj with respect to some reference point.

For i = 0,1 and 2N-2: connect nodes from the corresponding lists L_i directly to the CPA.

Delays

$$\text{Delay}(S) = \text{MAX}\{\text{Delay}(A) + D_{A\text{-}S}, \text{Delay}(B) + D_{B\text{-}S}, \text{Delay}(C_{in}) + D_{Cin\text{-}S}\}$$

$$\text{Delay}(C) = \text{MAX}\{\text{Delay}(A) + D_{A\text{-}C}, \text{Delay}(B) + D_{B\text{-}C}, \text{Delay}(C_{in}) + D_{Cin\text{-}C}\}$$

In our case the delays in a FA are :

$$FA_{A\to S} = FA_{B\to S} = 2 \text{ XOR delays}$$

$$FA_{Cin\to S} = FA_{A\to C} = FA_{B\to C} = FA_{Cin\to C} = 1 \text{ XOR delay.}$$

In a HA:

$$HA_{A\to S} = HA_{B\to S} = 1 \text{ XOR delay while } HA_{A\to C} = HA_{B\to C} = 0.5 \text{ XOR delay.}$$

For i=2 to i=2N-3 {Partial Product Array Generation}

Begin For

if length of Li is even Then

Begin If

sort the elements of Li in ascending order by the values of delay δ_j

connect an HA to the first 2 elements of Li starting with the slowest input

$Ds = \max \{\delta_A+\delta_{A-S}, \delta_B+\delta_{B-S}\}$

$Dc = \max \{\delta_A+\delta_{A-C}, \delta_B+\delta_{B-C}\}$

remove 2 elements from L_i

insert the pair <Ds,NetName> into L_i

insert the pair <Dc,NetName> into L_{i+1}

decrement the length of L_i

increment the length of L_{i+1}

End If;

while length of Li > 3
Begin While
sort the elements of Li in ascending order by the values of delay δj
connect an FA to the first 3 elements of Li starting with the slowest input of the FA:

$Ds = \max \{\delta c_A + \delta c_{A-S}, \delta c_B + \delta c_{B-S}, \delta c_{Ci} + \delta c_{Ci-S}\}$
$Dc = \max \{\delta c_A + \delta c_{A-C}, \delta c_B + \delta c_{B-C}, \delta c_{Ci} + \delta c_{Ci-C}\}$

remove 3 elements from Li
insert the pair <Ds,NetName> into Li
insert the pair <Dc,NetName> into Li+1
subtract 2 from the length of Li
increment the length of Li+1

End While;

sort the elements of Li
connect an FA to the last 3 nodes of Li
connect the S and C to the bit i and i+1 of the CPA

End For;
End Method;

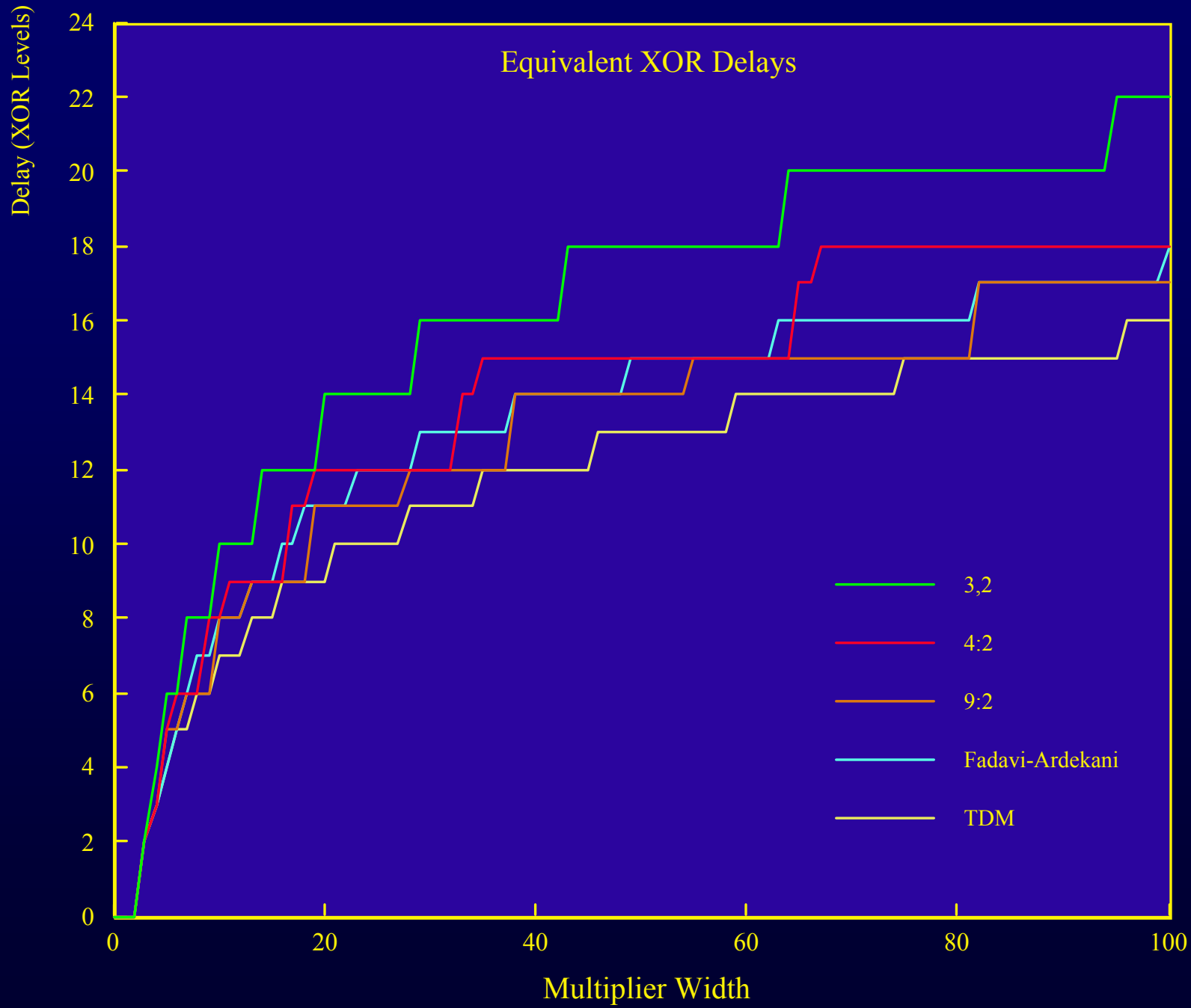
Competing Approaches

Comparison between TDM and other representative schemes, in XOR levels.

Multiplier Word-length	Wallace Tree [7]	4:2 Tree [11]	Fadavi - Ardekani [16]	TDM
3	2	2	2	2
4	4	3	3	3
6	6	6	5	5
8	8	6	7	5
9	8	8	7	6
11	10	9	8	7
12	10	9	8	7
16	12	9	10	8
19	12	12	11	9
24	14	12	12	10
32	16	12	13	11
42	16	15	14	12
53	18	15	15	13
64	20	15	16	14
95	20	18	17	15

Critical Path Delay [CMOS: Leff=1 μ, T=25°C, V_{cc}=5V]

N = 24-bits	4:2 Design	9:2 Design	Fadavi-Ardekani	TDM Design
Delay [nS]	14.0	13.0	11.7	10.5


Equivalent XOR Delays
Delay (XOR Levels)
0
2
4
6
8
10
12
14
16
18
20
22
24
0
20
40
60
80
100
Multiplier Width
3,2
4:2
9:2
Fadavi-Ardekani
TDM

Algorithm for Implementation of Fast Parallel Multipliers

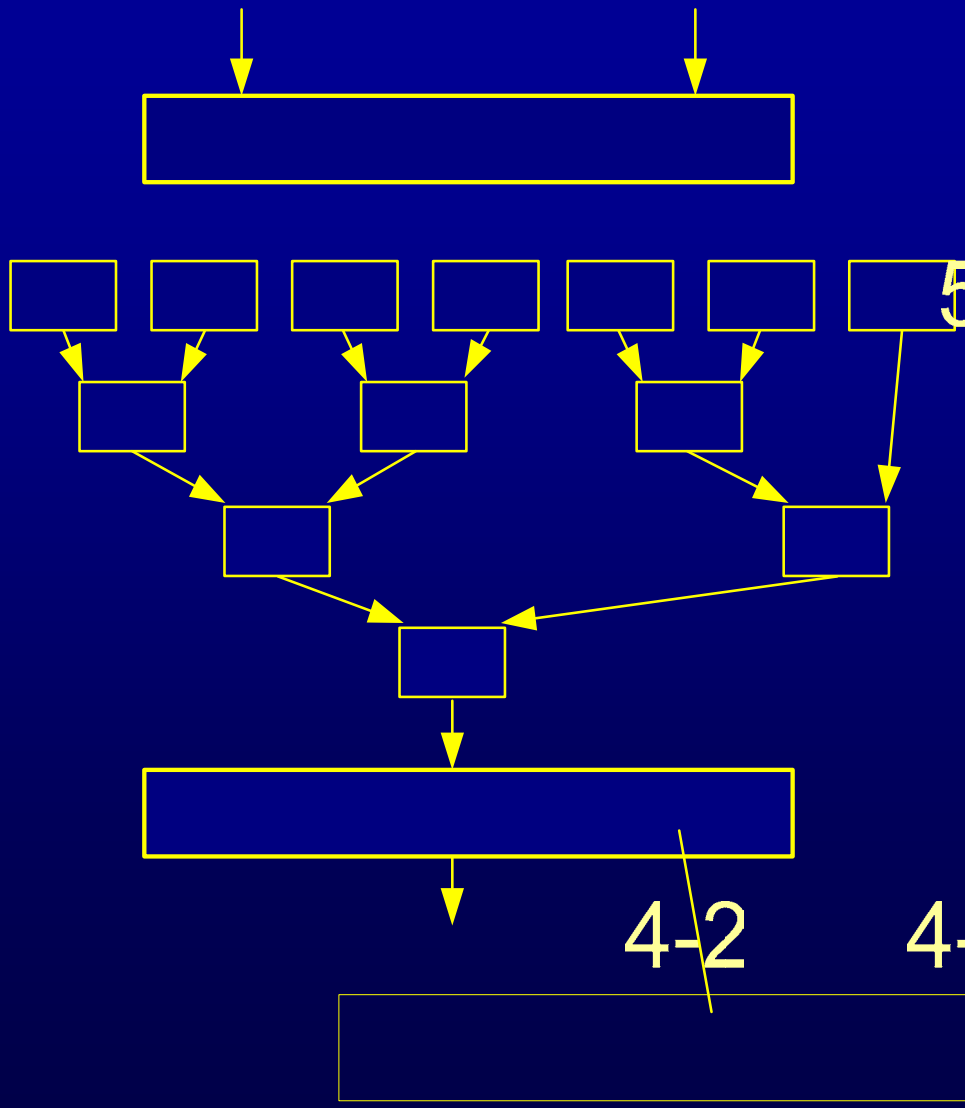
[1] V. G. Oklobdzija, D. Villeger, and S. S. Liu, "A Method For Speed Optimized Partial Product Reduction And Generation Of Fast Parallel Multipliers Using An Algorithmic Approach," IEEE Transactions on Computers, Vol 45, No.3, March, 1996.

[2] V. G. Oklobdzija and D. Villeger, "Improving Multiplier Design By Using Improved Column Compression Tree And Optimized Final Adder In CMOS Technology," IEEE Transactions on VLSI Systems, Vol.3, No.2, June, 1995, 25 pages.

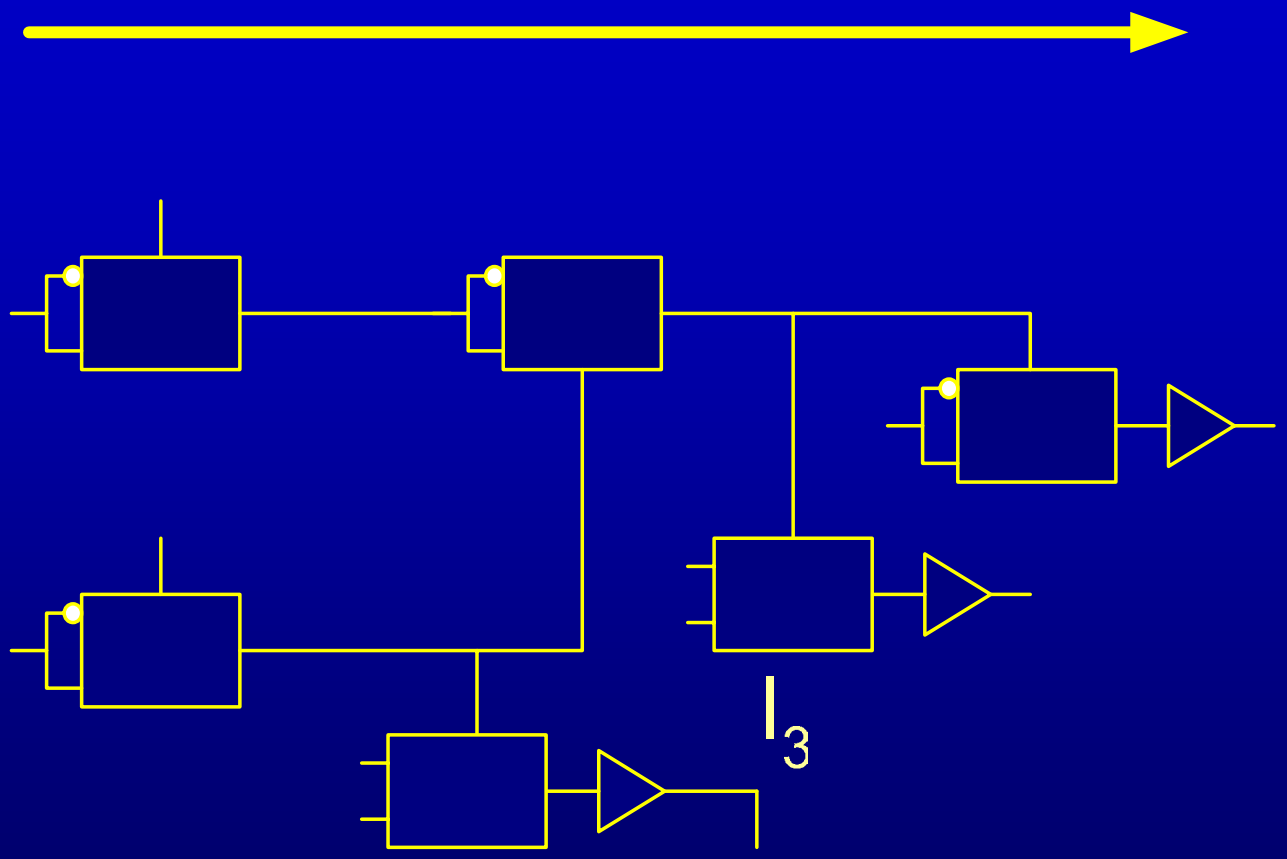
[3] V. G. Oklobdzija and D. Villeger, "Multiplier Design Utilizing Improved Column Compression Tree And Optimized Final Adder In CMOS Technology," Proceedings of the 1993 International Symposium on VLSI Technology, Systems and Applications, pp. 209-212, 1993.

[4] P. Stelling, C. Martel, V. G. Oklobdzija, R. Ravi, "Optimal Circuits for Parallel Multipliers," IEEE Transaction on Computers, Vol. 47, No.3, pp. 273-285, March, 1998.

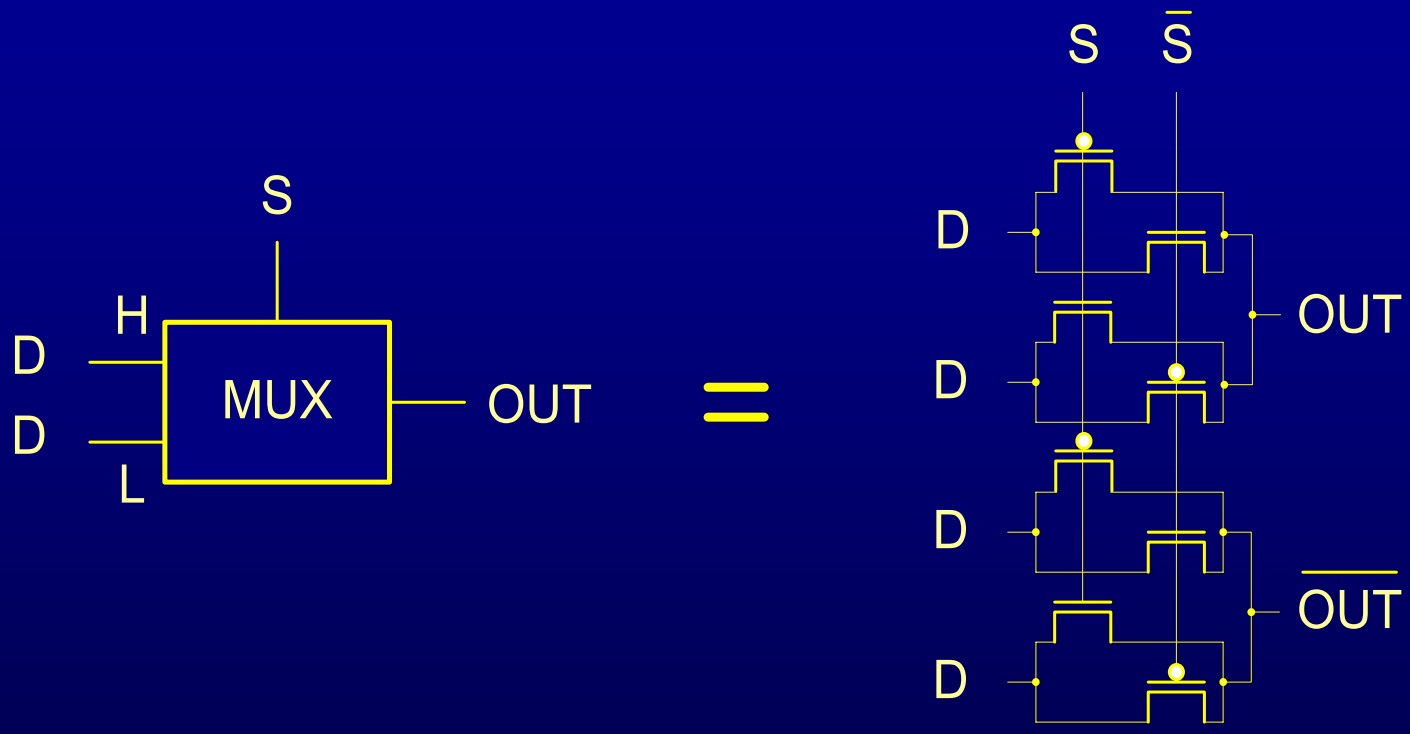
Organization of Hitachi's DPL multiplier



Hitachi's 4:2 compressor structure



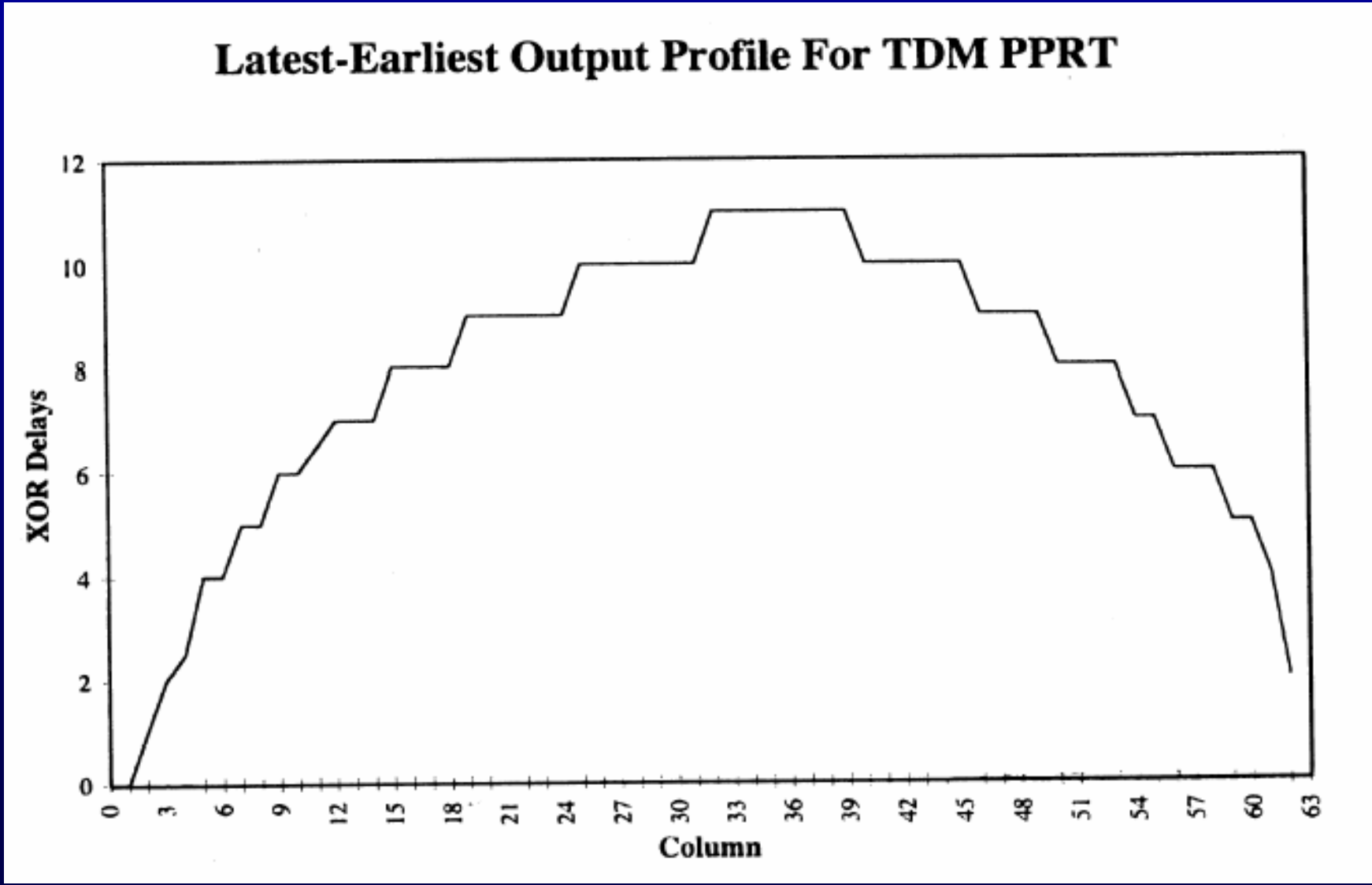
DPL multiplexer circuit



Addition Under Non-equal Signal Arrival Profile Assumption

P. Stelling , V. G. Oklobdzija, "Design Strategies for Optimal Hybrid Final Adders in a Parallel Multiplier", *special issue on VLSI Arithmetic*, Journal of VLSI Signal Processing, Kluwer Academic Publishers, Vol.14, No.3, December 1996

Signal Arrival Profile form the Parallel Multiplier Partial-Product Recuction Tree



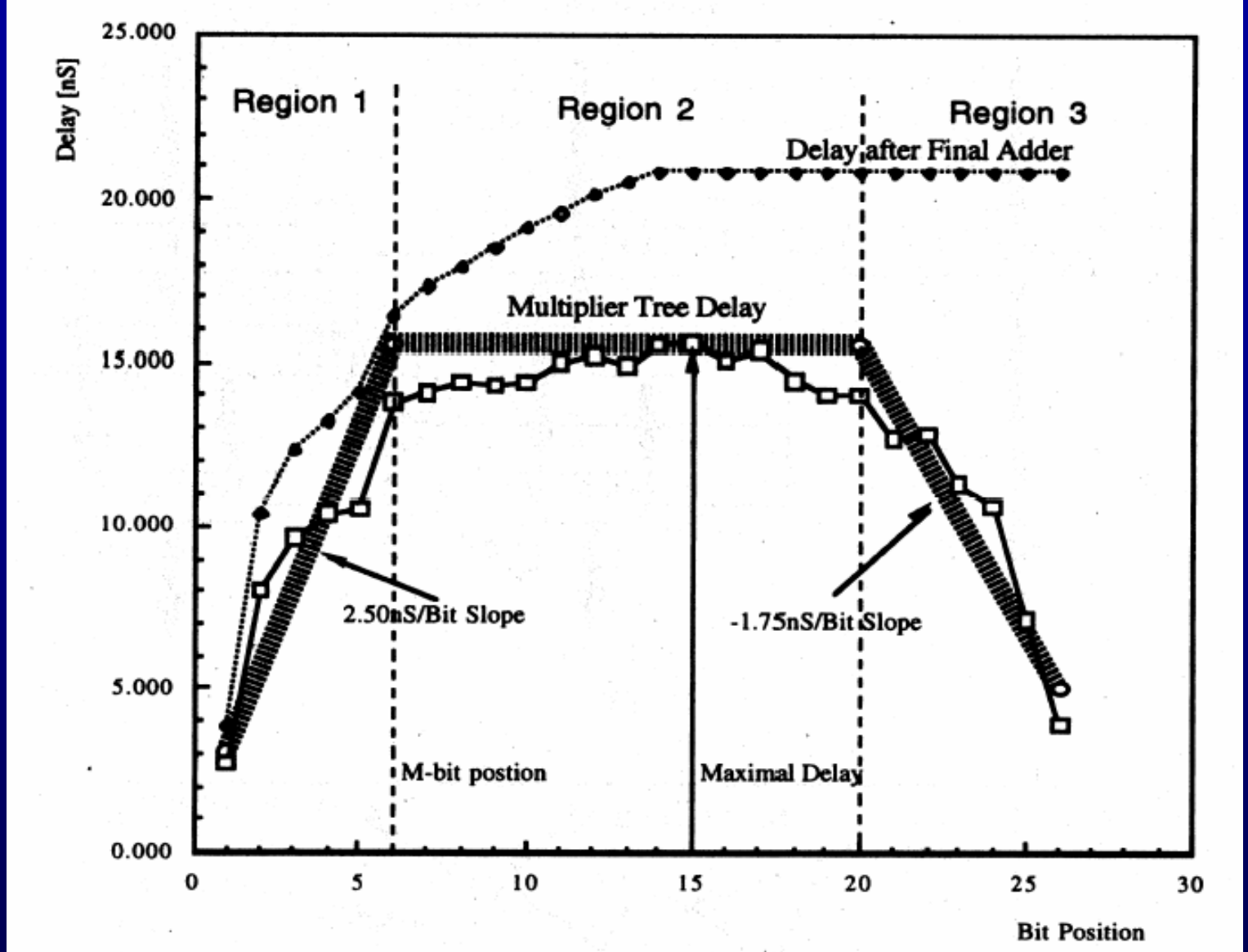


Fig. 15. Selection of the adder types in the three regions of the multiplier.

Oklobdzija, Villeger, *IEEE Transactions on VLSI Systems*, June, 1995

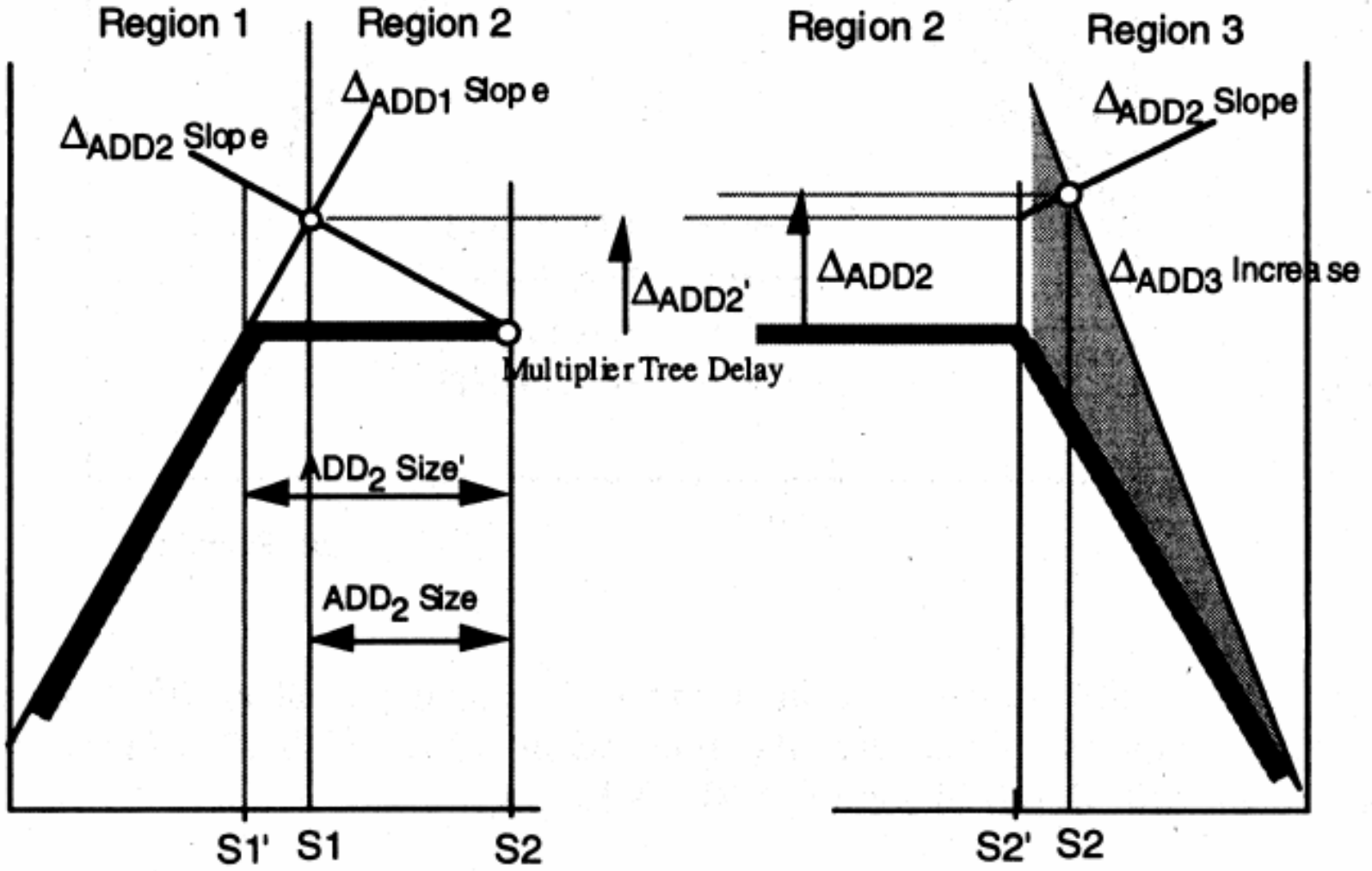
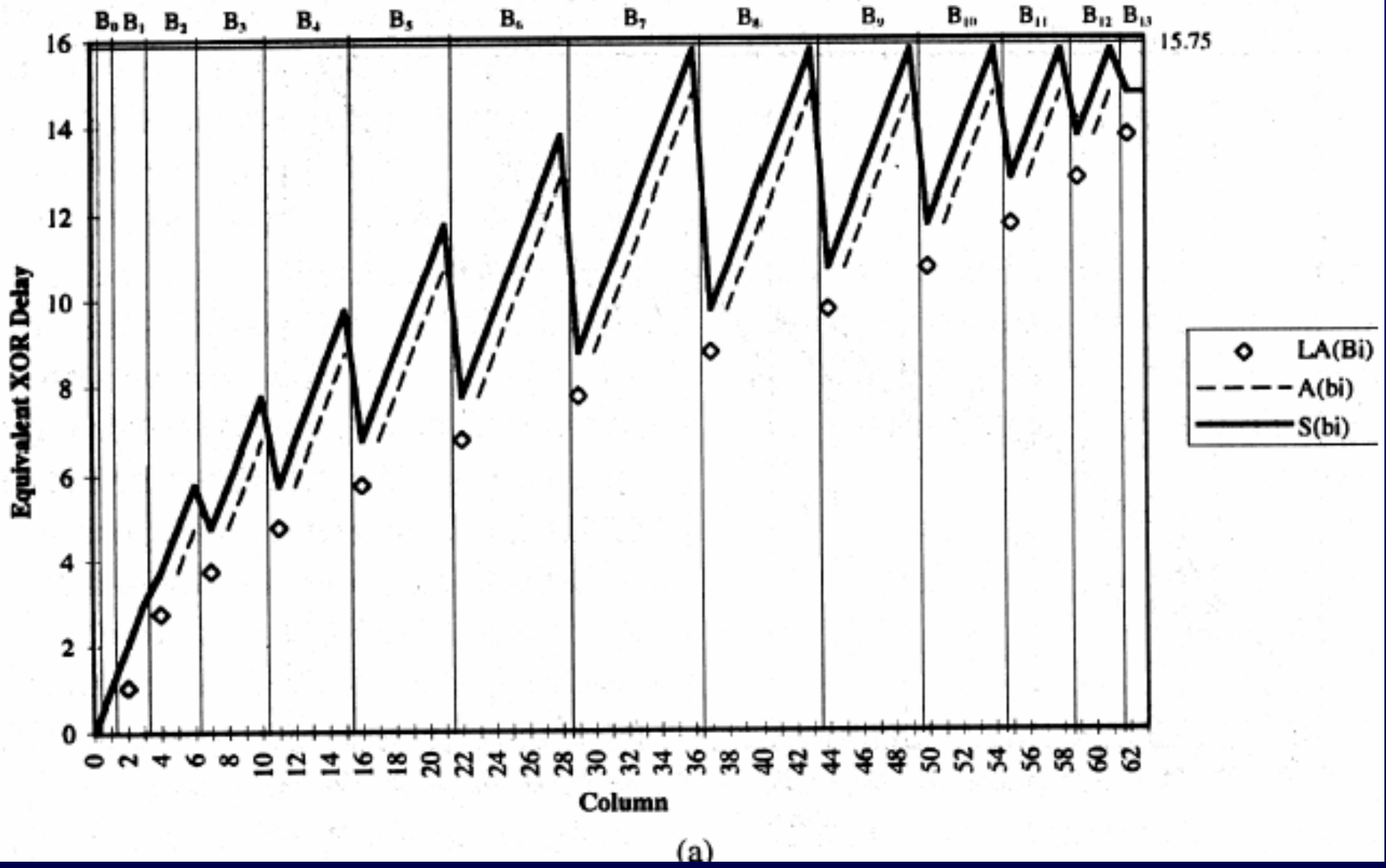
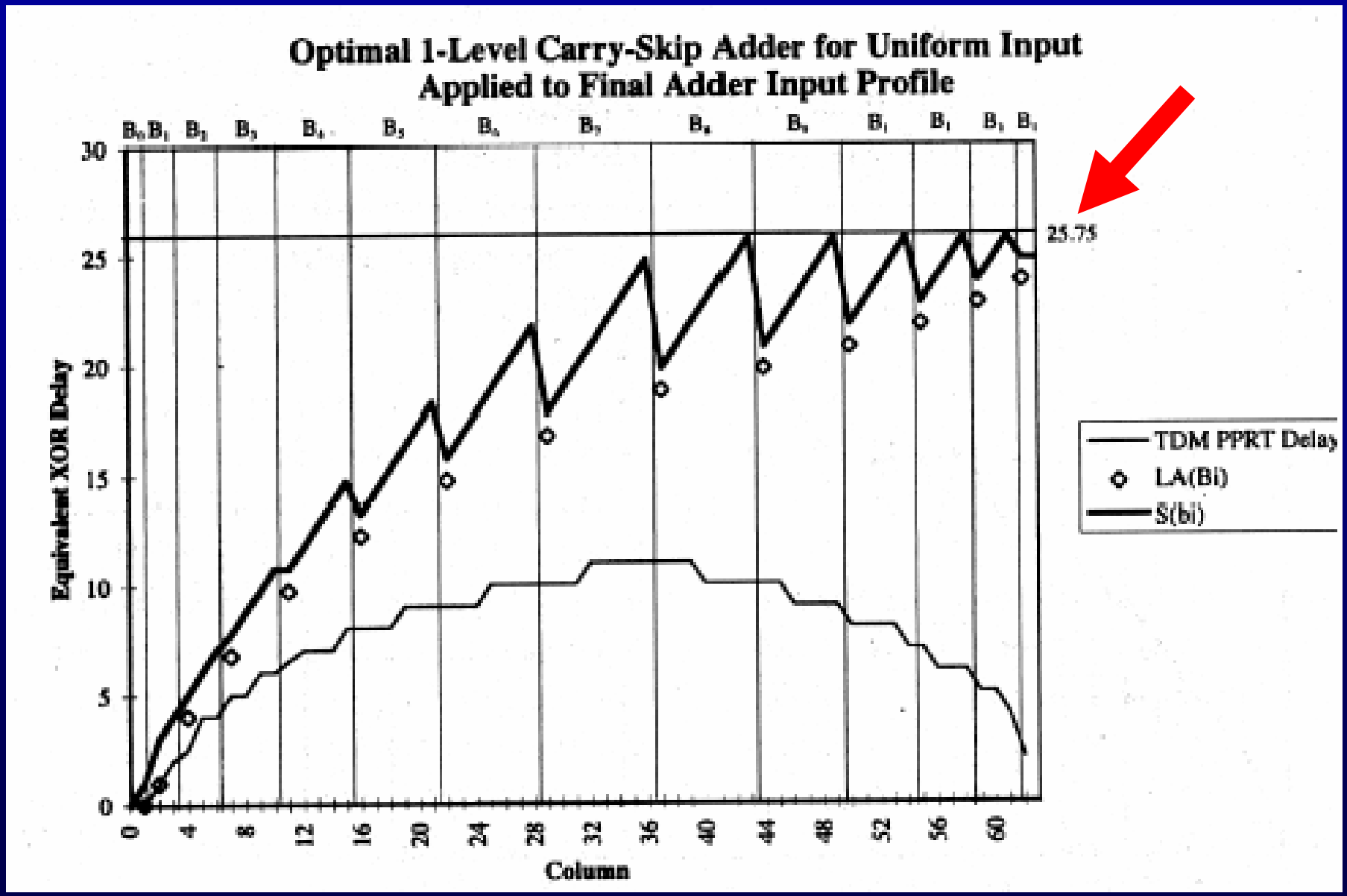
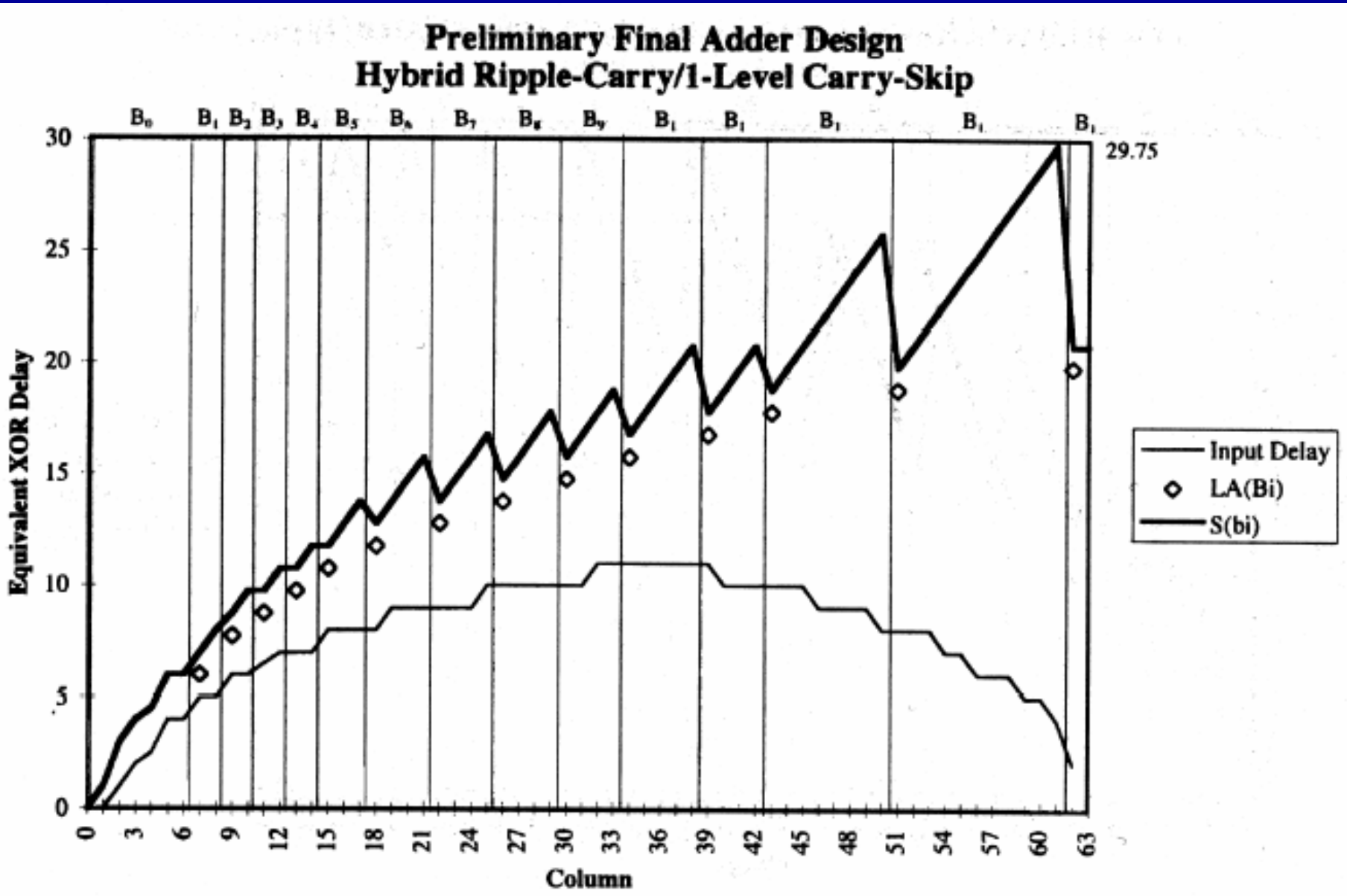
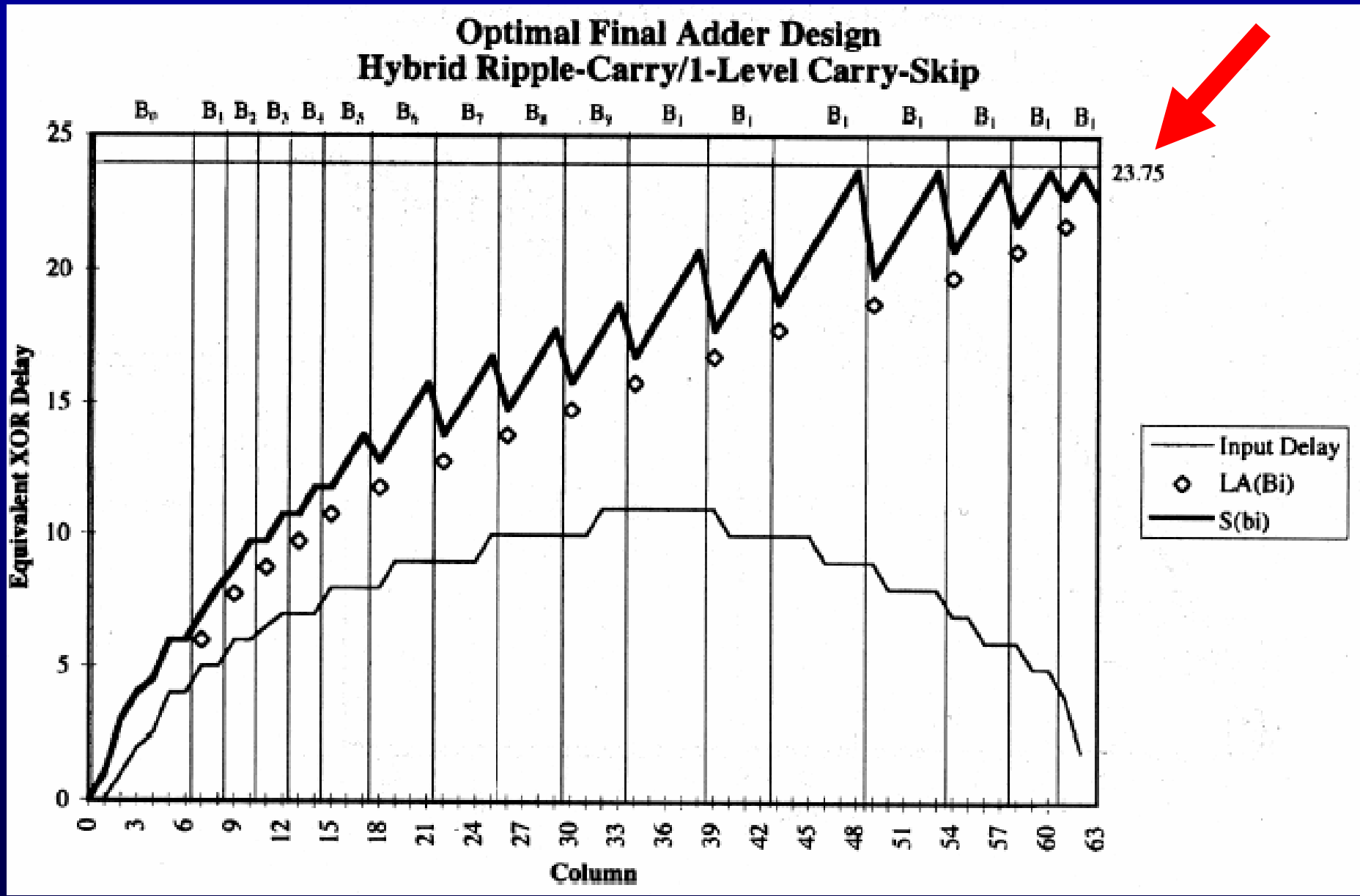


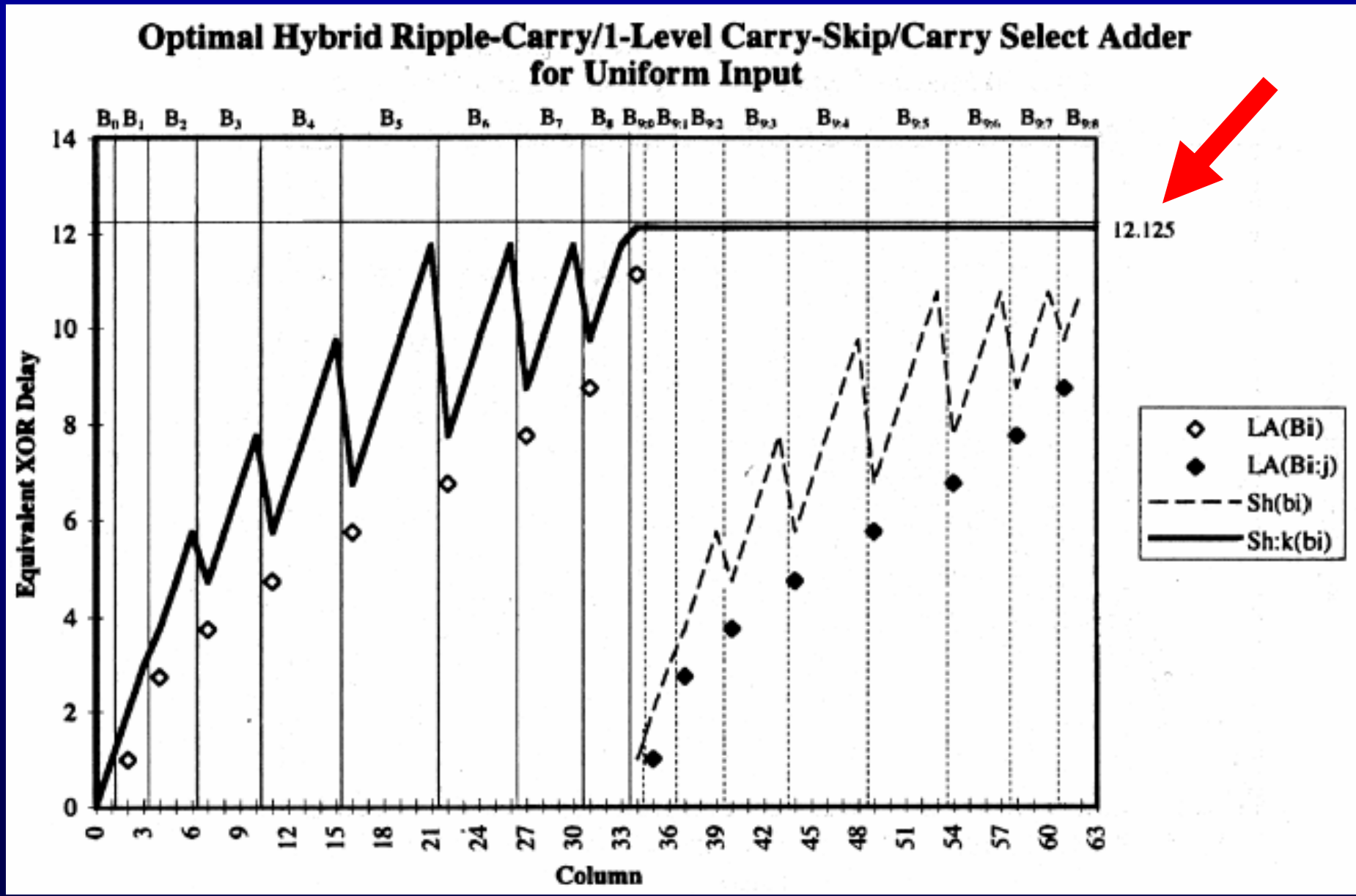
Fig. 16. Determination of bit positions S_1 and S_2 determining the size of the adders used.

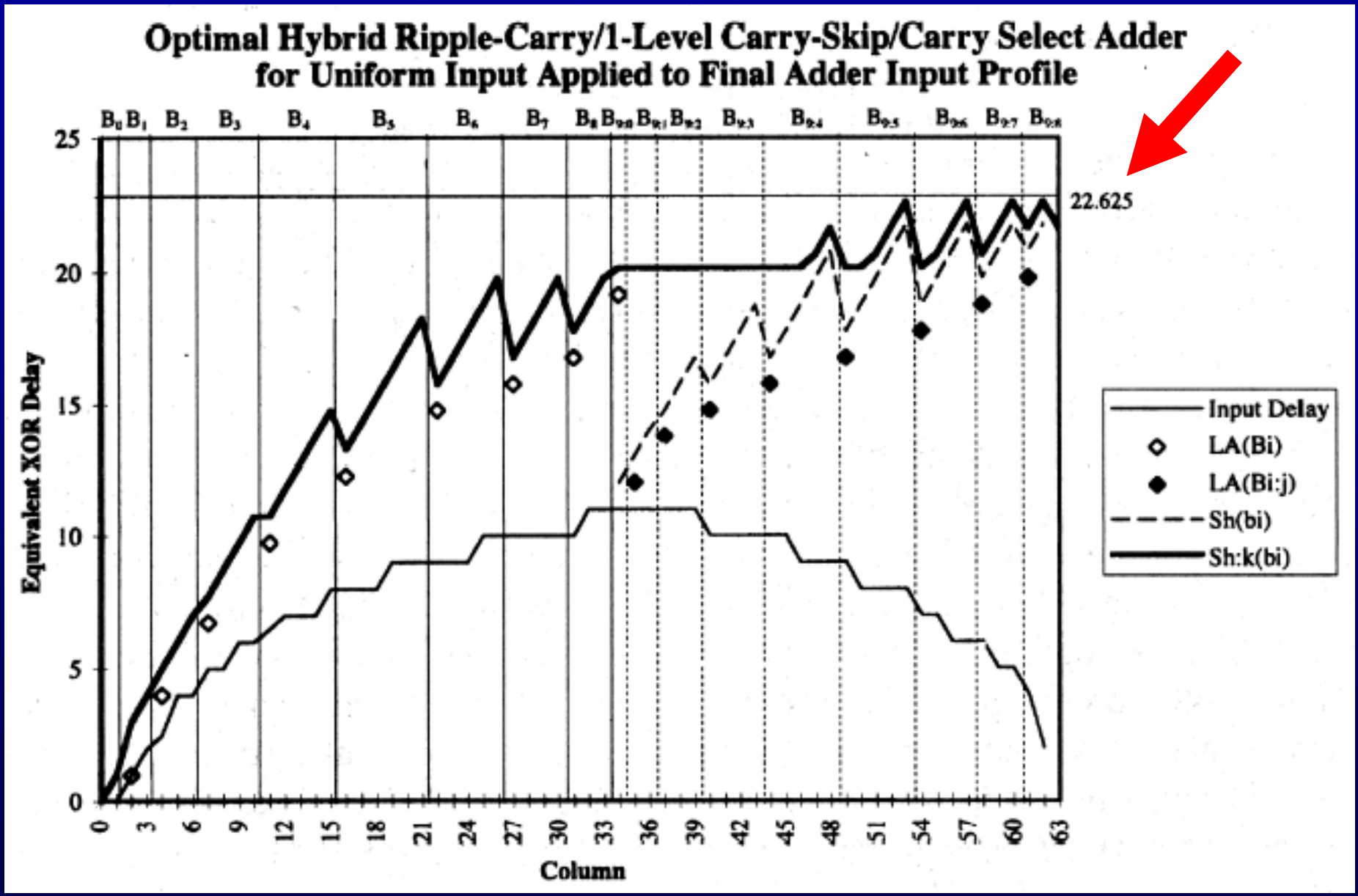

Optimal 1-Level Carry-Skip Adder for Uniform (All 0) Input
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 B11 B12 B13
15.75
Equivalent XOR Delay
Column
LA(Bi)
A(bi)
S(bi)
(a)

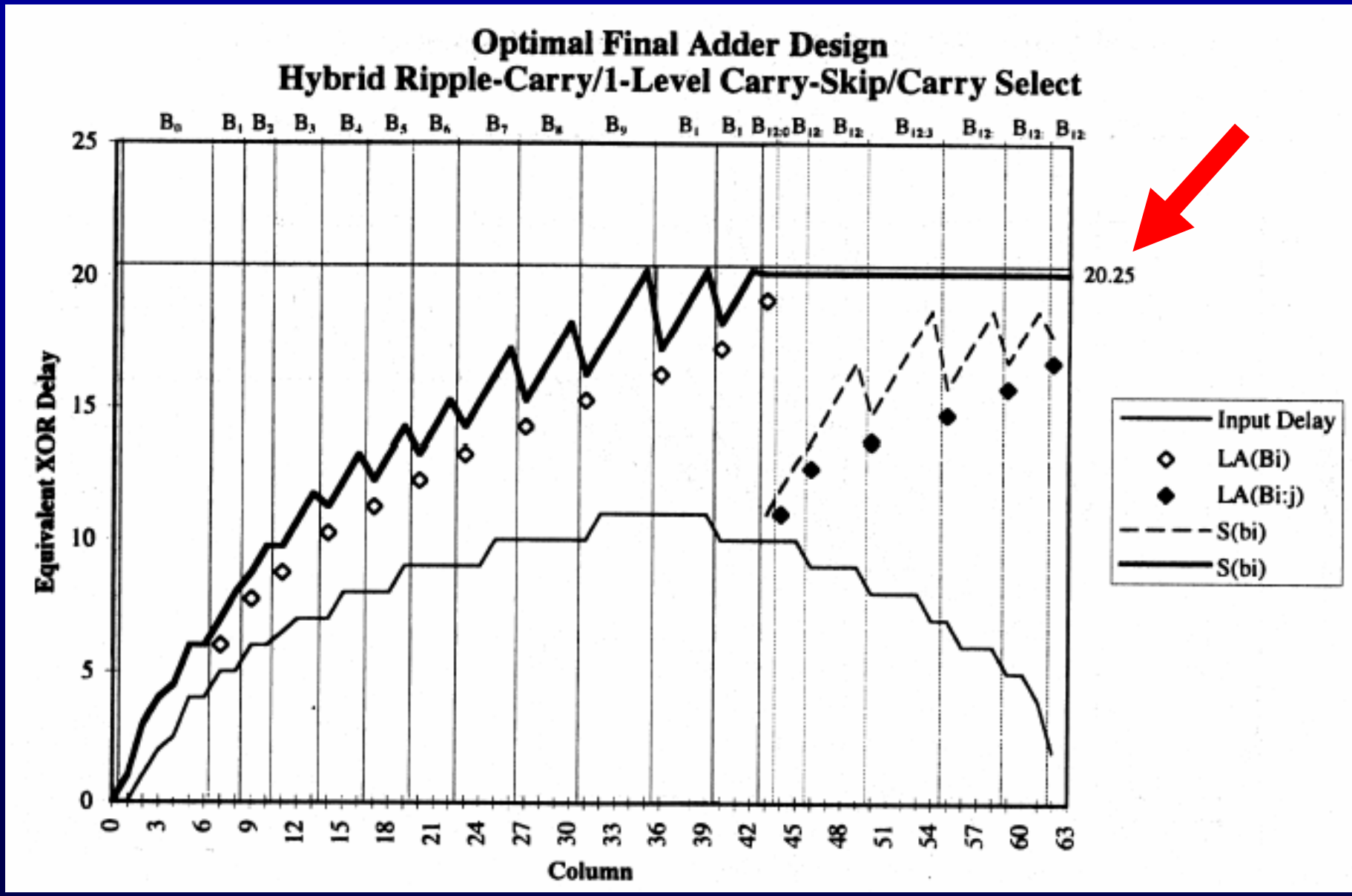

Optimal 1-Level Carry-Skip Adder for Uniform Input
Applied to Final Adder Input Profile
Equivalent XOR Delay
Column
25.75
TDM PPRT Delay
LA(Bi)
S(bi)

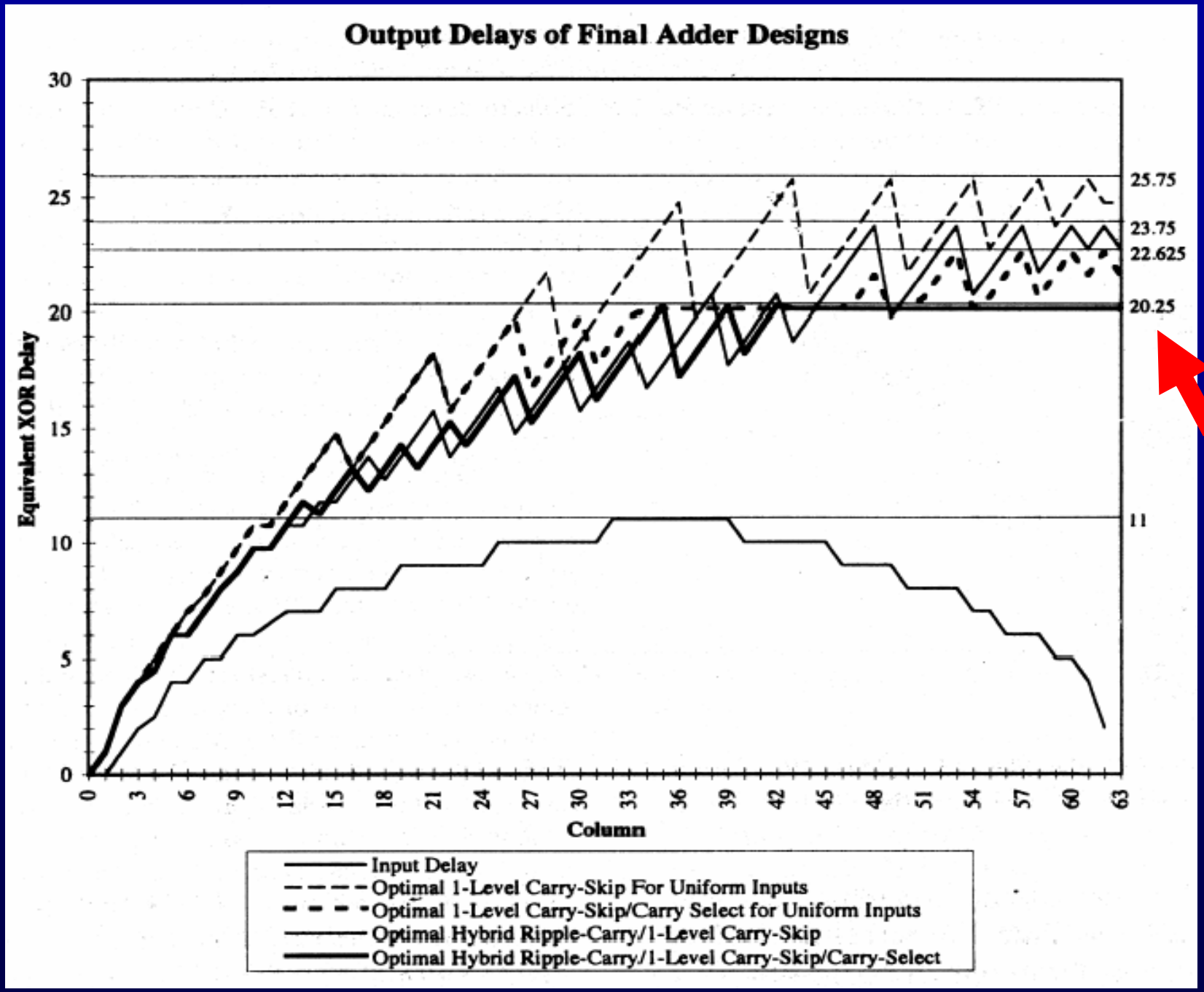

Preliminary Final Adder Design
Hybrid Ripple-Carry/1-Level Carry-Skip
Equivalent XOR Delay
Column
29.75
Input Delay
LA(Bi)
S(bi)


Optimal Final Adder Design
Hybrid Ripple-Carry/1-Level Carry-Skip
Equivalent XOR Delay
Column
23.75
Input Delay
LA(Bi)
S(bi)


Optimal Hybrid Ripple-Carry/1-Level Carry-Skip/Carry Select Adder
for Uniform Input
Equivalent XOR Delay
Column
12.125
LA(Bi)
LA(Bi:j)
Sh(bi)
Sh:k(bi)

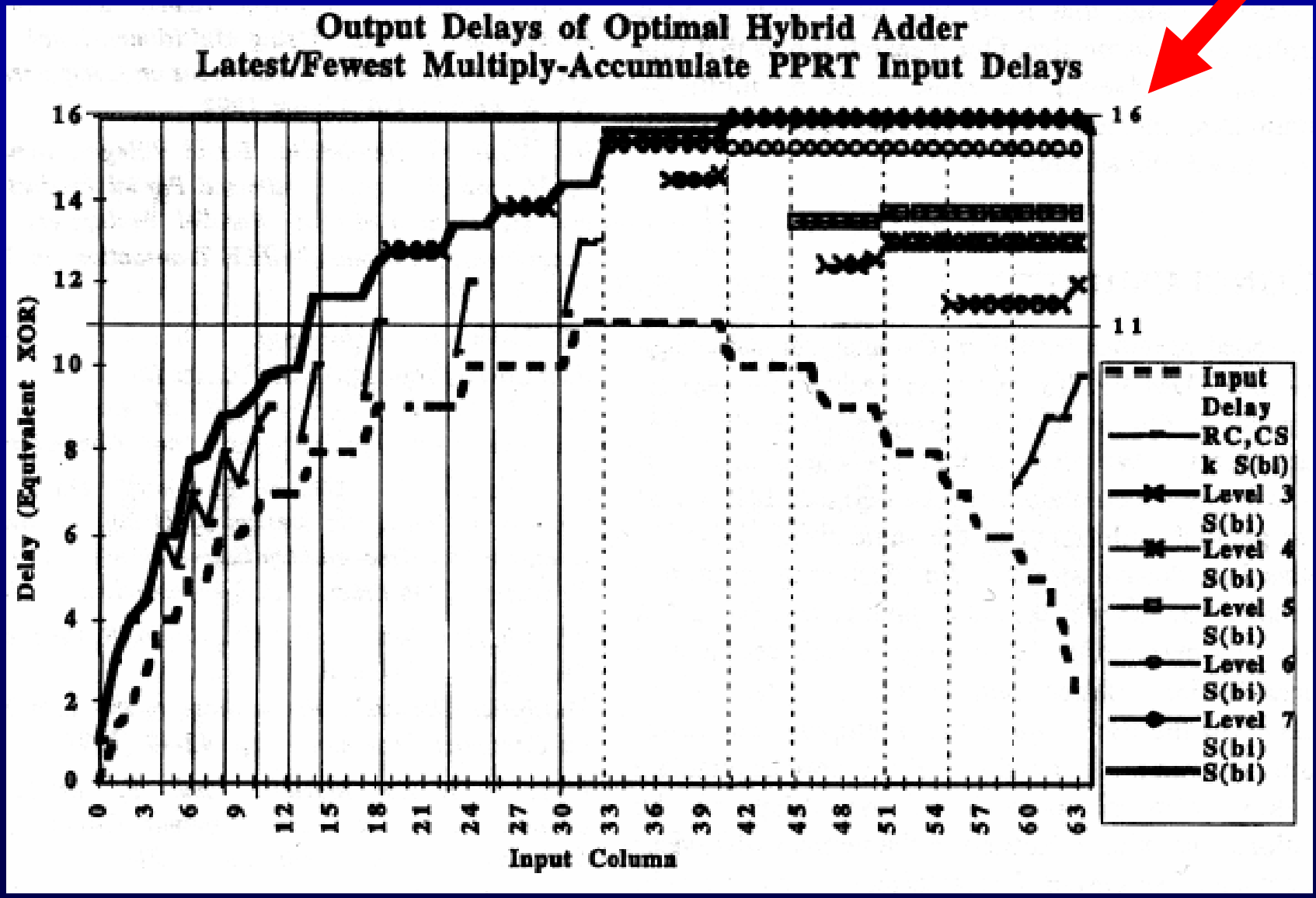

Optimal Hybrid Ripple-Carry/1-Level Carry-Skip/Carry Select Adder
for Uniform Input Applied to Final Adder Input Profile
Equivalent XOR Delay
Column
22.625
Input Delay
LA(Bi)
LA(Bi:j)
Sh(bi)
Sh:k(bi)


Optimal Final Adder Design
Hybrid Ripple-Carry/1-Level Carry-Skip/Carry Select
Equivalent XOR Delay
Column
20.25
Input Delay
LA(Bi)
LA(Bi:j)
S(bi)
S(bi)

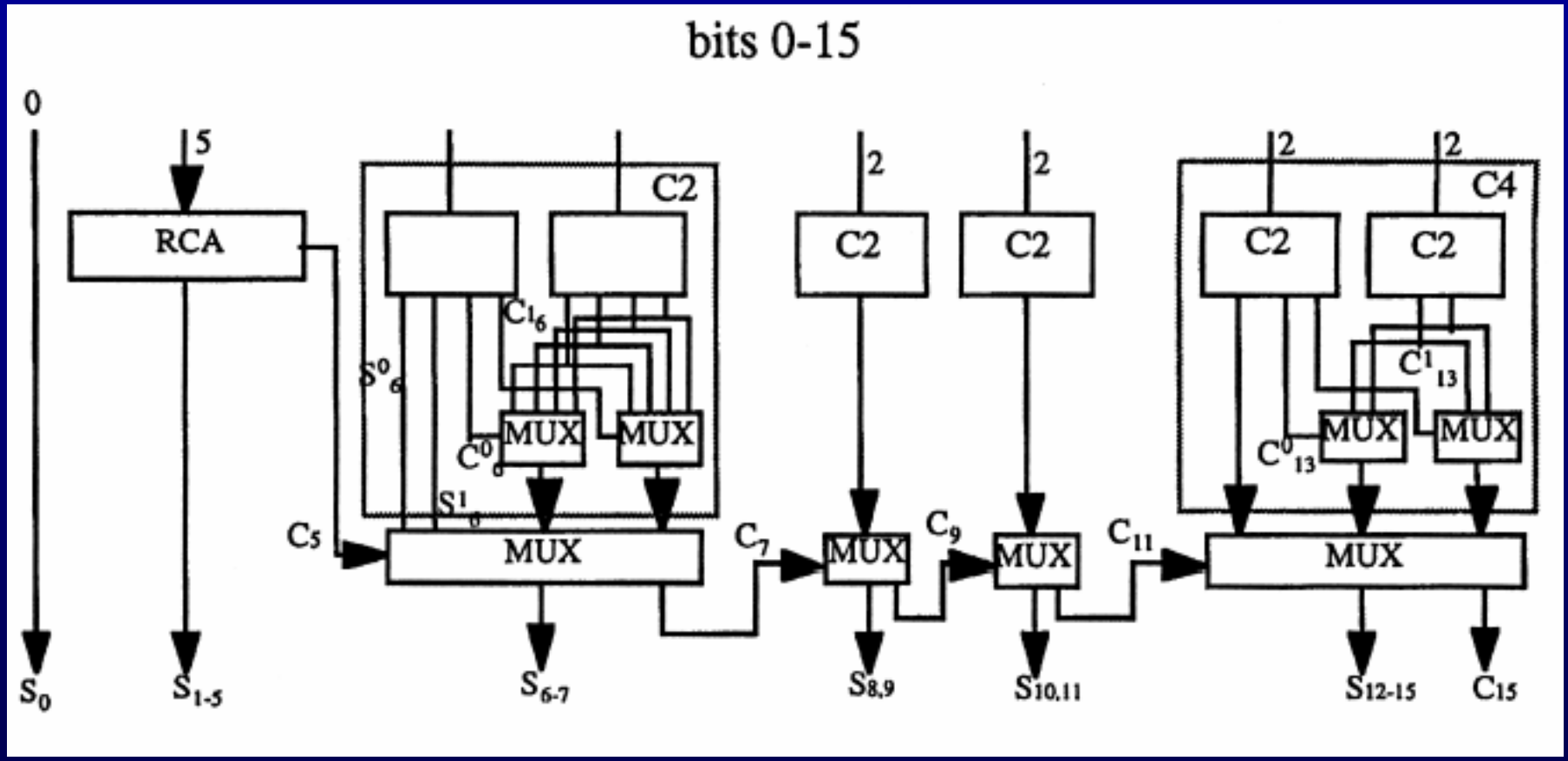

Output Delays of Final Adder Designs
Equivalent XOR Delay
Column
0
5
10
15
20
25
30
0
3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
48
51
54
57
60
63
25.75
23.75
22.625
20.25
11
Input Delay
Optimal 1-Level Carry-Skip For Uniform Inputs
Optimal 1-Level Carry-Skip/Carry Select for Uniform Inputs
Optimal Hybrid Ripple-Carry/1-Level Carry-Skip
Optimal Hybrid Ripple-Carry/1-Level Carry-Skip/Carry-Select

Performing Multiply-Add Operation in the Multiply Time

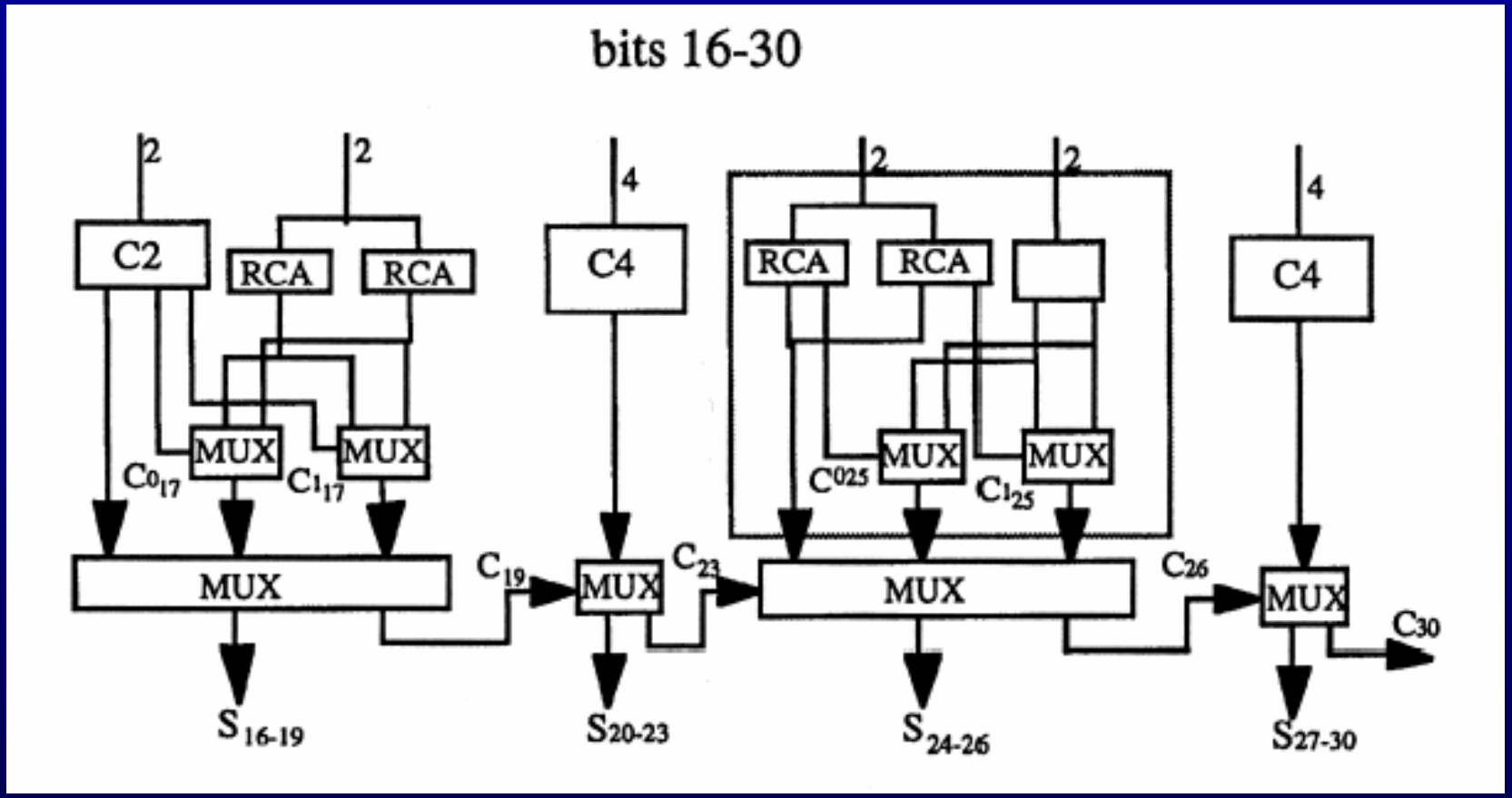
P. Stelling, V. G. Oklobdzija, " Achieving Multiply-Accumulate Operation in the Multiply Time", Thirteenth International Symposium on Computer Arithmetic, Pacific Grove, California, July 5 - 9, 1997.


Output Delays of Optimal Hybrid Adder
Latest/Fewest Multiply-Accumulate PPRT Input Delays
Delay (Equivalent XOR)
Input Column
16
11
Input Delay
RC,CSk S(bl)
Level 3 S(bl)
Level 4 S(bl)
Level 5 S(bl)
Level 6 S(bi)
Level 7 S(bi)
S(bl)

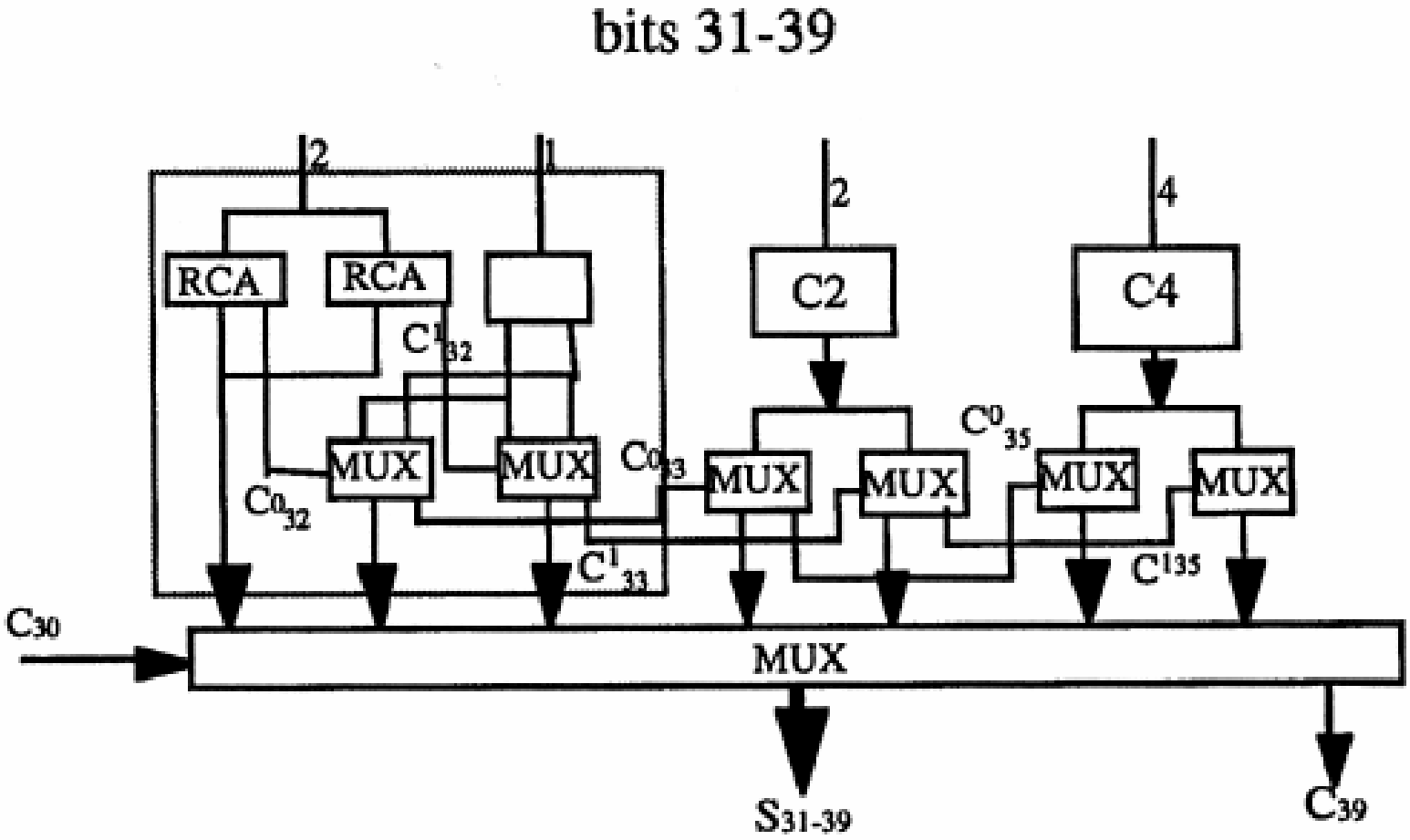
Final Adder: Implementation

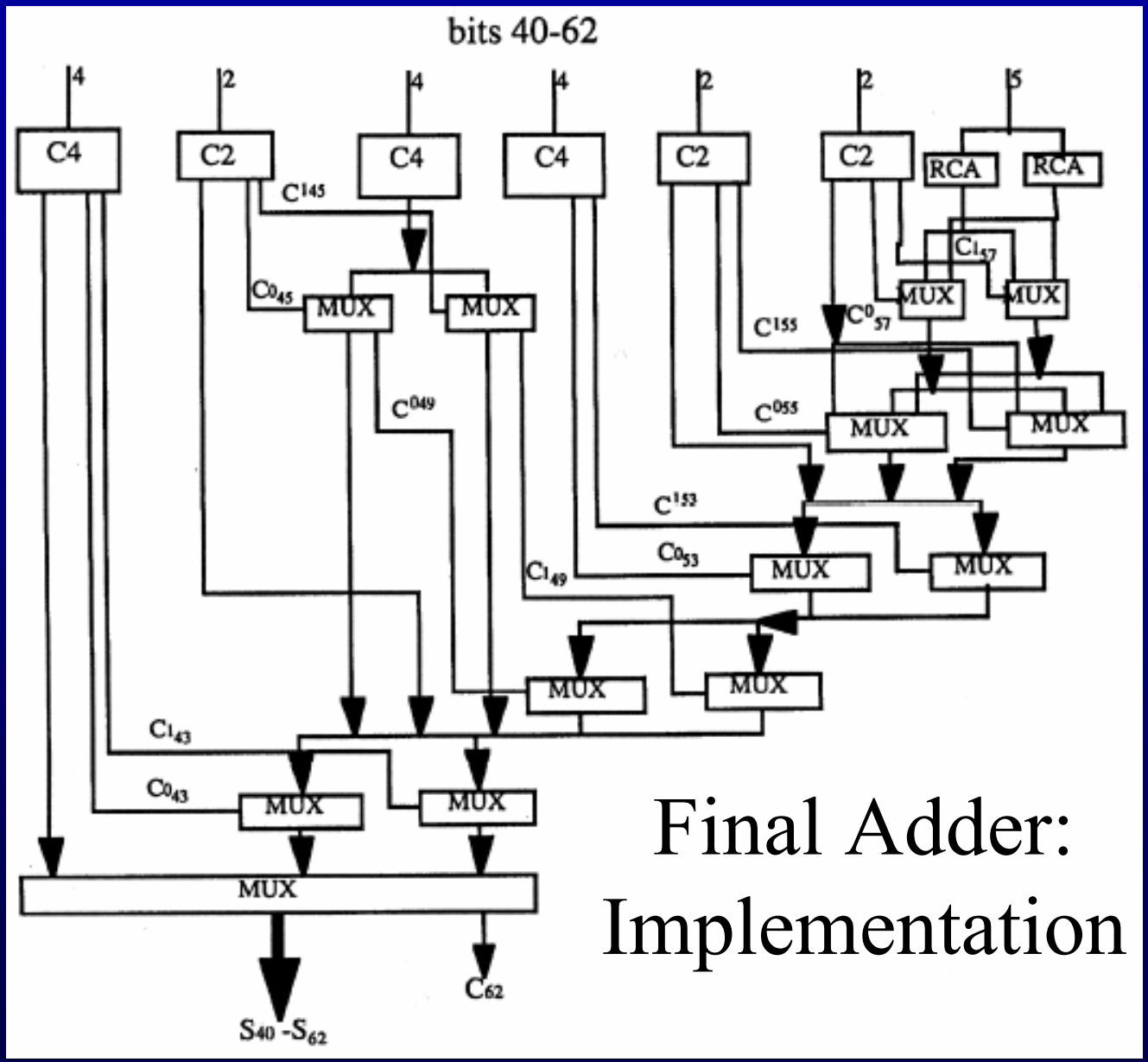


Final Adder: Implementation



Final Adder: Implementation



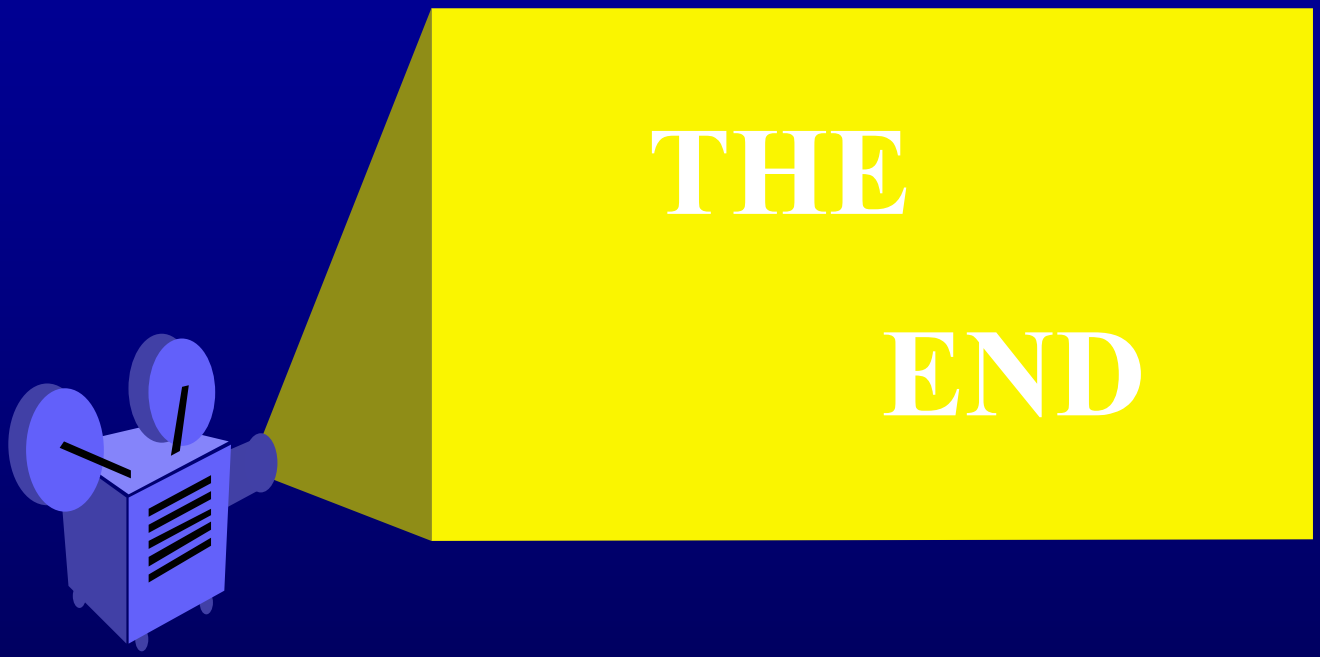


Final Adder: Implementation


RECOMENDATIONS

Fast Parallel Multipliers

- Different Counter and Compressor Families were compared. The best way is to build a compressor of the maximal size (i.e. the entire size of the multiplier)
- The Essence of the optimal tree is optimal wiring and NOT the use of counter/compressor family
- The use of Carry-Propagate Adders is advantageous for larger size multipliers in the first stage and for particular technology
- Tuning of the Final Adder into the signal arrival profile is more important than the speed of the Final Adder.


THE
END