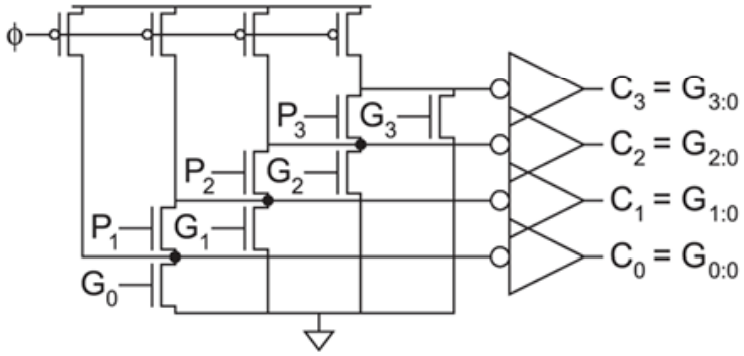
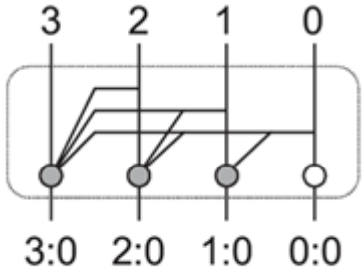
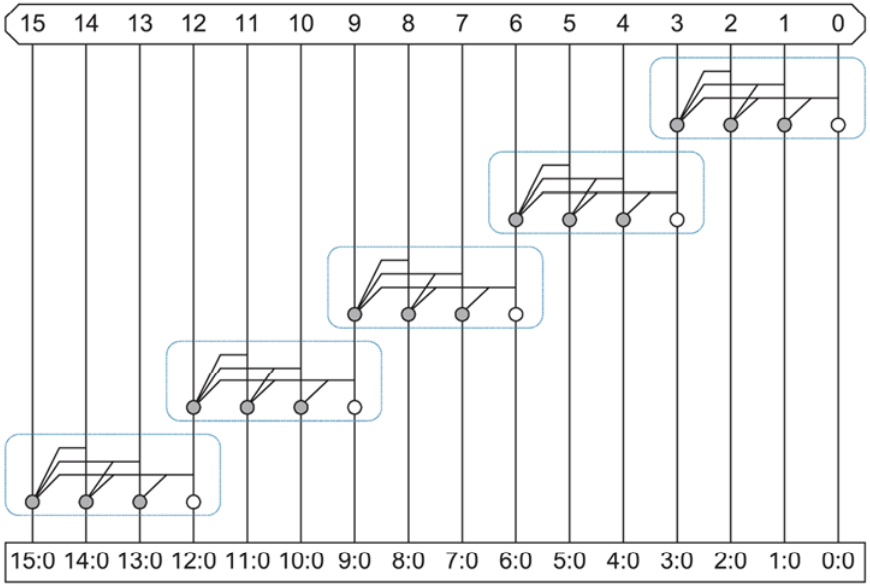
Manchester Carry Chain

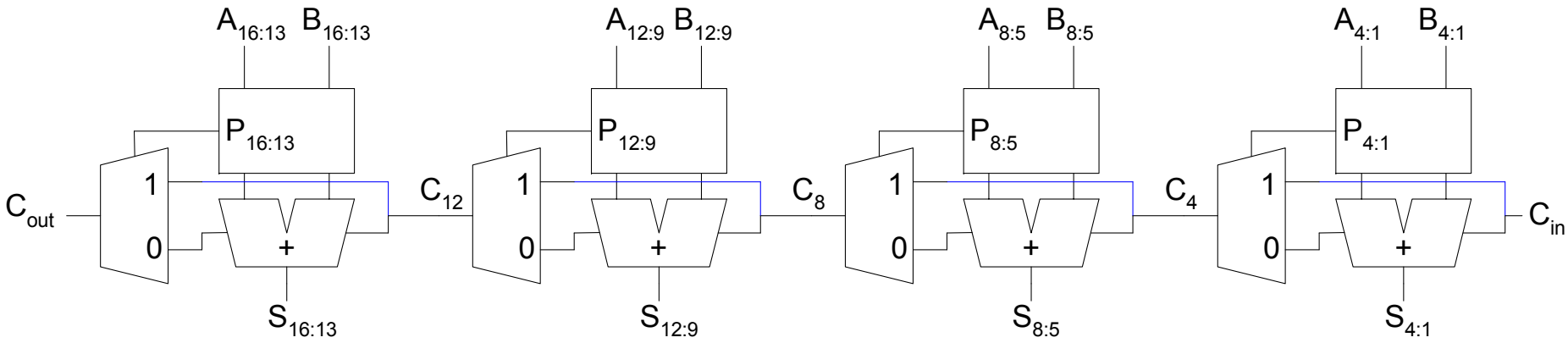




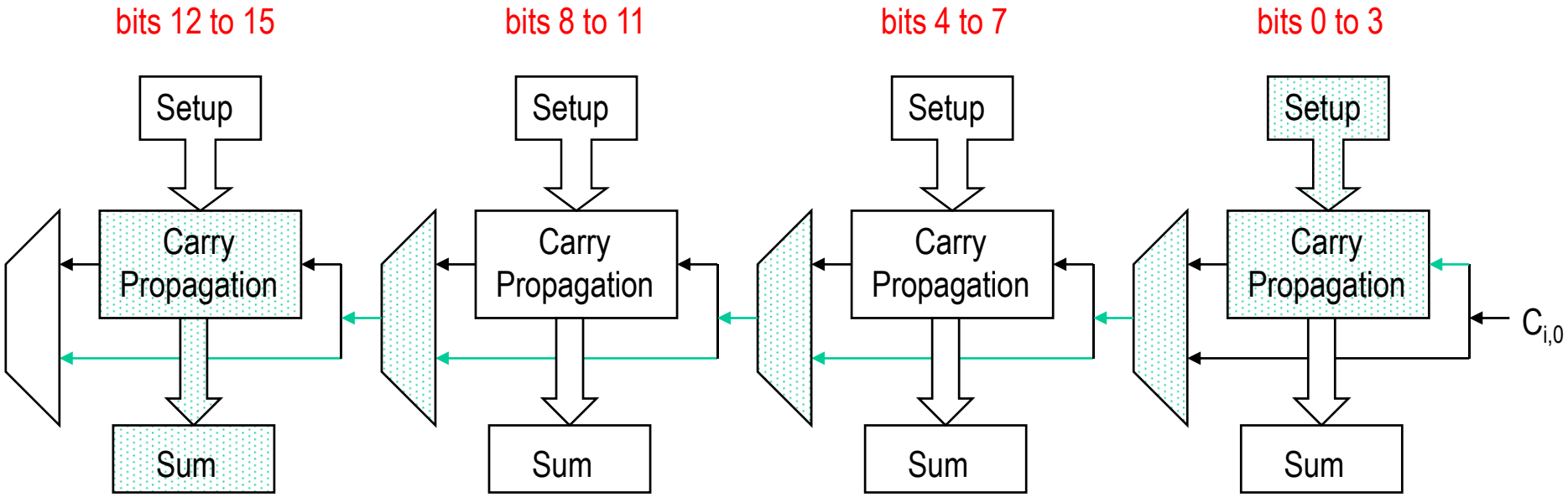


Carry-Skip Adder

- Carry-ripple is slow through all N stages
- Carry-skip allows carry to skip over groups of n bits
  - Decision based on n-bit propagate signal



4/16 bit Block Carry Skip Adder



Worst-case delay → carry from bit 0 to bit 15 = carry generated in bit 0, ripples through bits 1, 2, and 3, skips the middle two groups (B is the group size in bits), ripples in the last group from bit 12 to bit 15

$$T_{add} = t_{setup} + B\,t_{carry} + ((N/B) - 1)\,t_{skip} + (B-1)\,t_{carry} + t_{sum}$$

Optimal Skip Block Size and Add Time

- Assuming one stage of ripple (t_{carry}) has the same delay as one skip logic stage (t_{skip}) and both are 1

$$T_{CSkA} = \underbrace{1}_{t_{setup}} + \underbrace{B}_{\text{ripple in block 0}} + \underbrace{(N/B-1)}_{\text{skips}} + \underbrace{B-1}_{\text{ripple in last block}} + \underbrace{1}_{t_{sum}}$$

$$= 2B + N/B$$

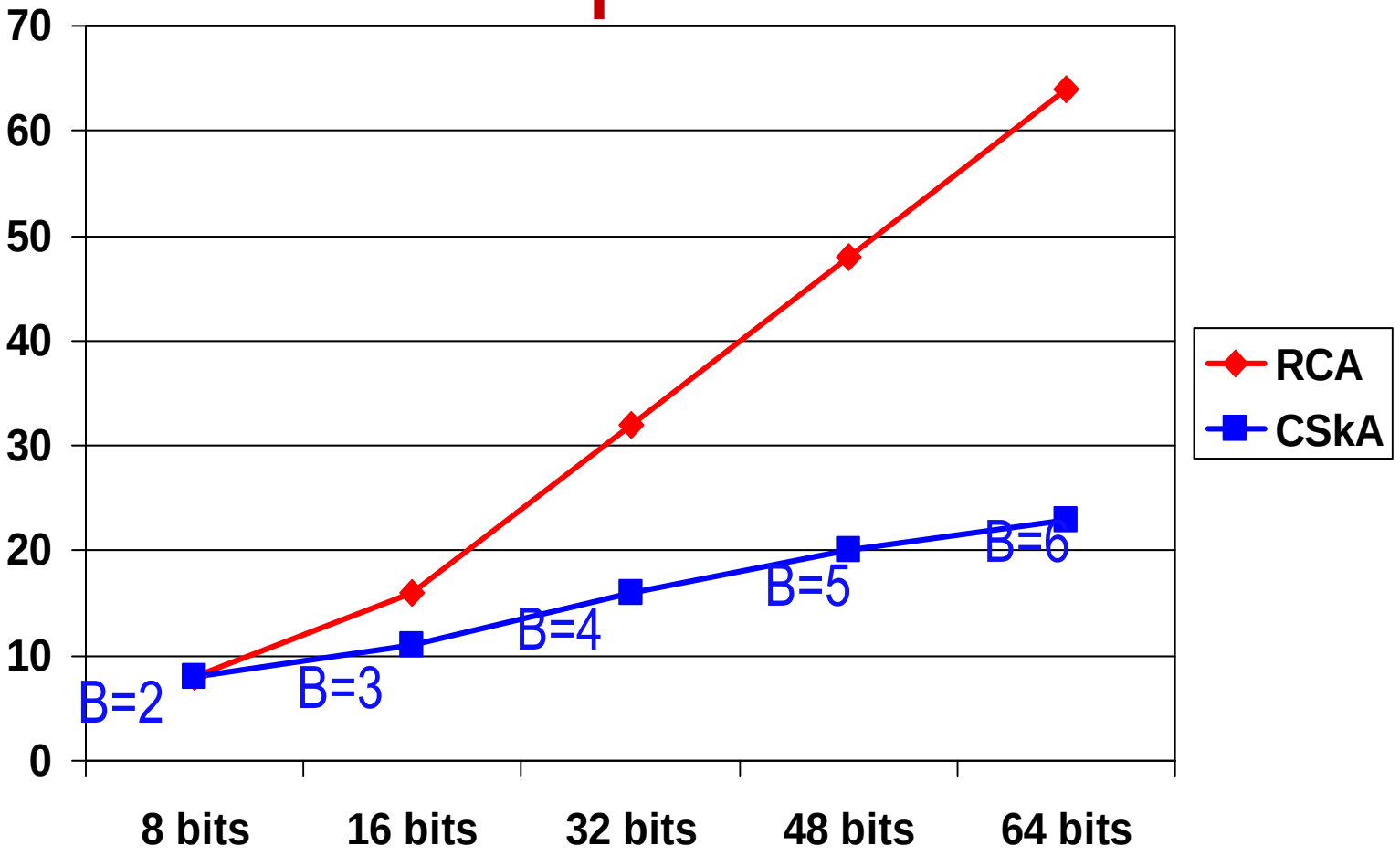
- So the optimal block size, B, is

$$dT_{CSkA}/dB = 0 \Rightarrow \sqrt{(N/2)} = B^{opt}$$

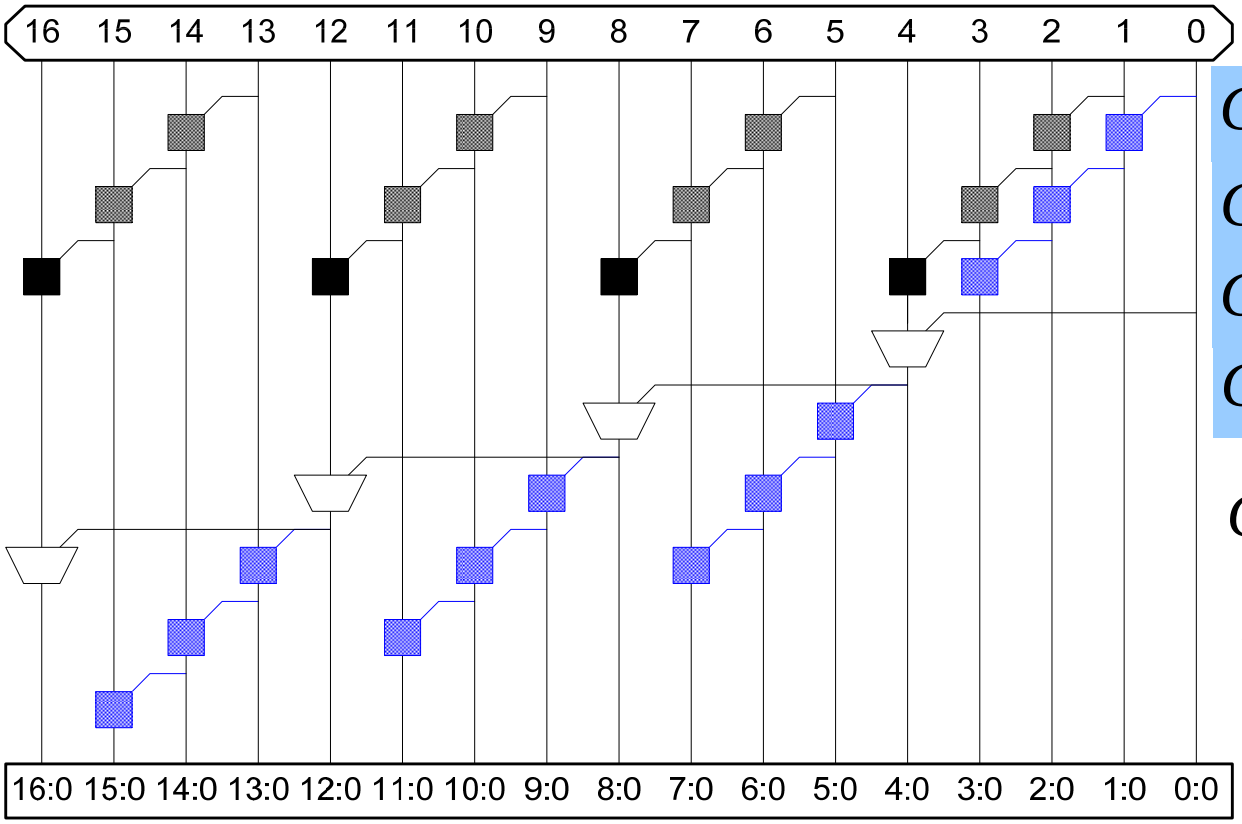
- And the optimal time is

$$\text{Optimal } T_{CSkA} = 4\sqrt{(n/2)} - 1 = 2\sqrt{(2n)} - 1$$

RCA, Carry Skip Adder Comparison



Carry-Skip PG Diagram



$$G_{4:0} = G_{4:1} + P_{4:1}G_{0:0}$$

$$G_{8:0} = G_{8:5} + P_{8:5}G_{4:0}$$

$$G_{12:0} = G_{12:9} + P_{12:9}G_{8:0}$$

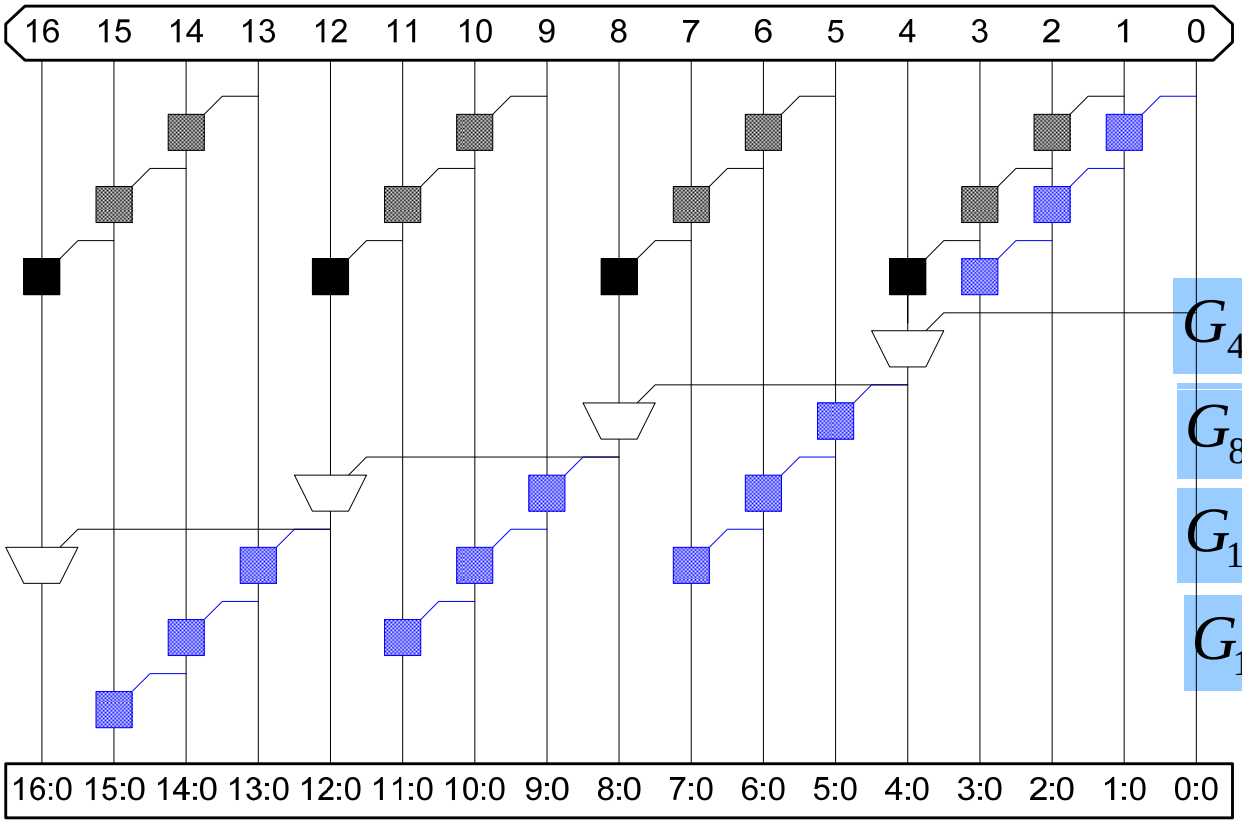
$$G_{16:0} = G_{16:13} + P_{16:13}G_{12:0}$$

$$G_{i:j} = G_{i:k} + P_{i:k} \square G_{k-1:j}$$

$$P_{i:j} = P_{i:k} \square P_{k-1:j}$$

$$t_{\text{skip}} =$$

Carry-Skip PG Diagram



$$G_{4:0} = G_{4:1} + P_{4:1}G_{0:0}$$

$$G_{8:0} = G_{8:5} + P_{8:5}G_{4:0}$$

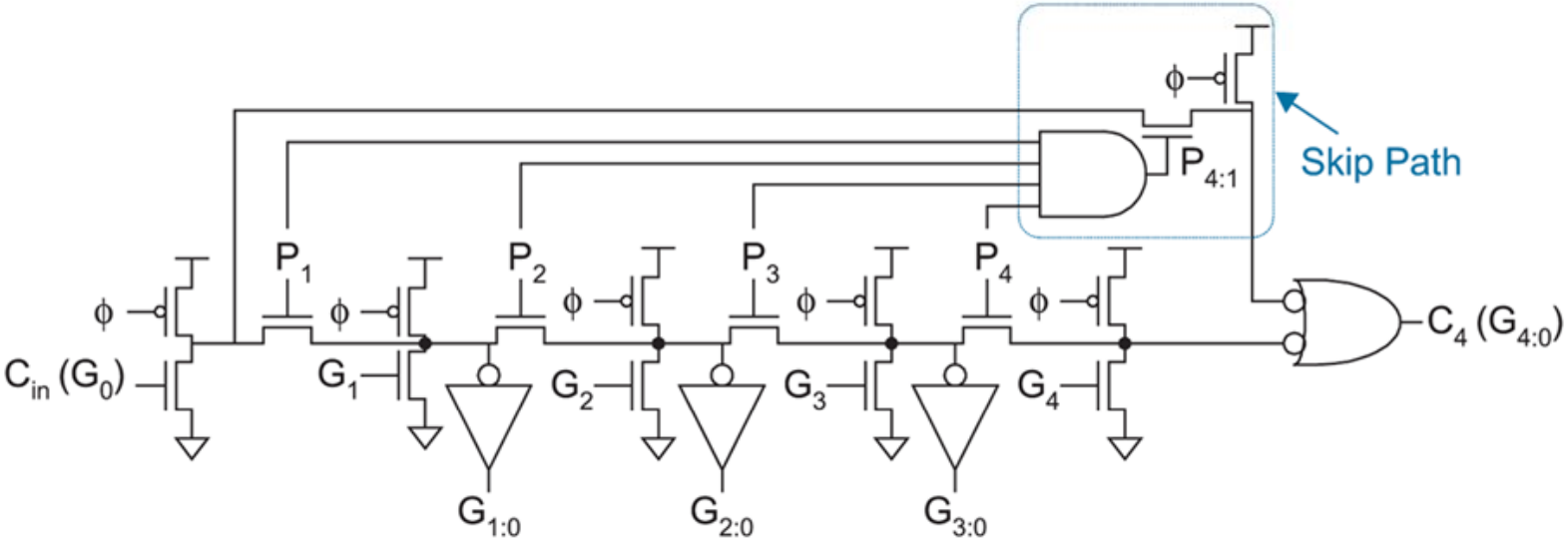
$$G_{12:0} = G_{12:0} + P_{12:9}G_{8:0}$$

$$G_{16:0} = G_{16:13} + P_{16:13}G_{12:0}$$

$$t_{\text{skip}} = t_{pg} + \left[2(n-1) + (k-1)\right]t_{AO} + t_{\text{xor}}$$

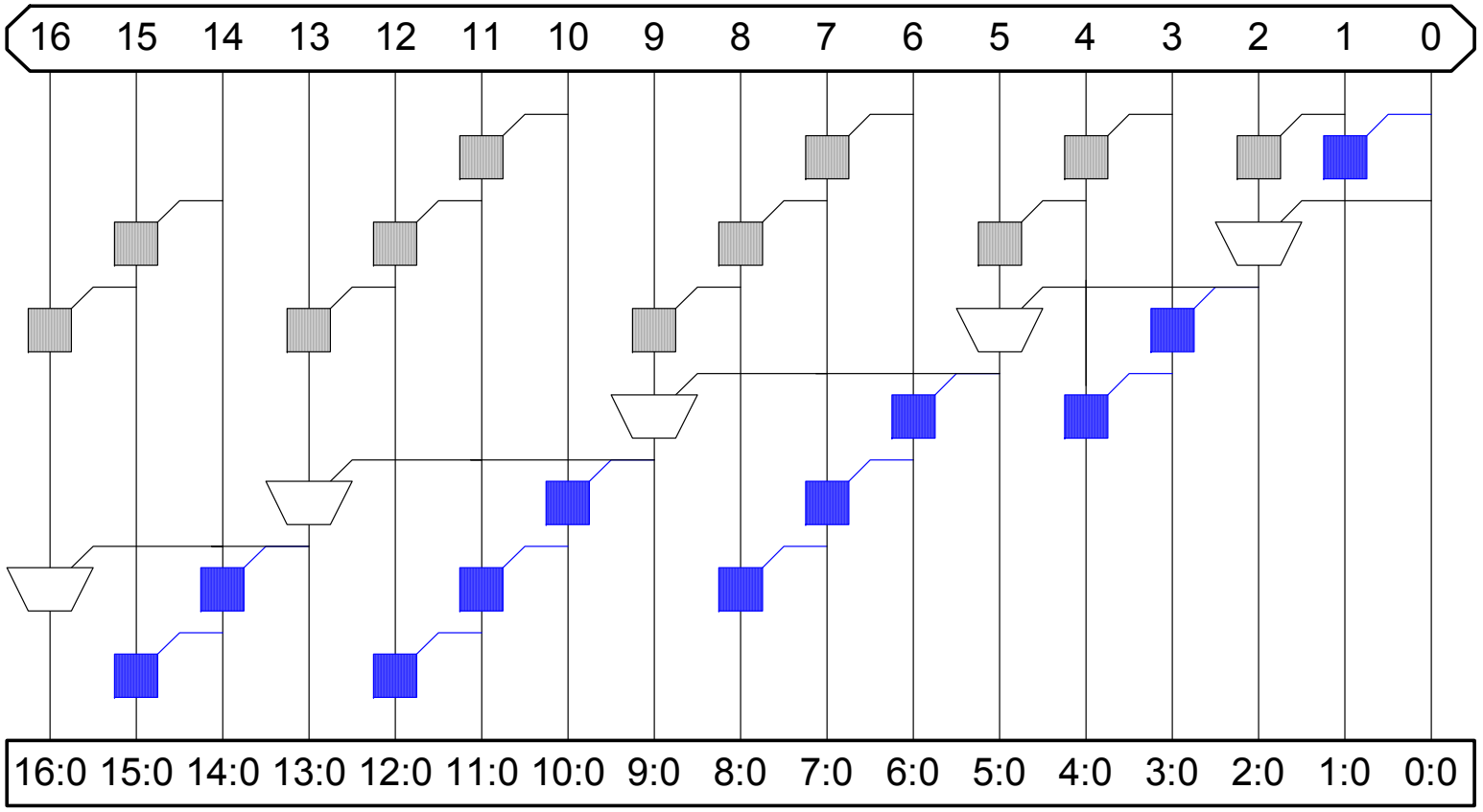
Carry-Skip PG Diagram

P.CHAN and M.Schlag,"Analysis and design of CMOS Manchester adders with variable carry-skip" *IEEE Trans.Computers*,Vol.39,No.8,Aug.1990,pp.983-992



$$G_{4,0} = \overline{G_{4,1}} + P_{4,1}\overline{G_{0,0}}$$

Variable Group Size



Delay grows as O(sqrt(N))

Look-Ahead: Topology

$$C_{o,k} = f(A_k, B_k, C_{o,k-1}) = G_k + P_k C_{o,k-1}$$

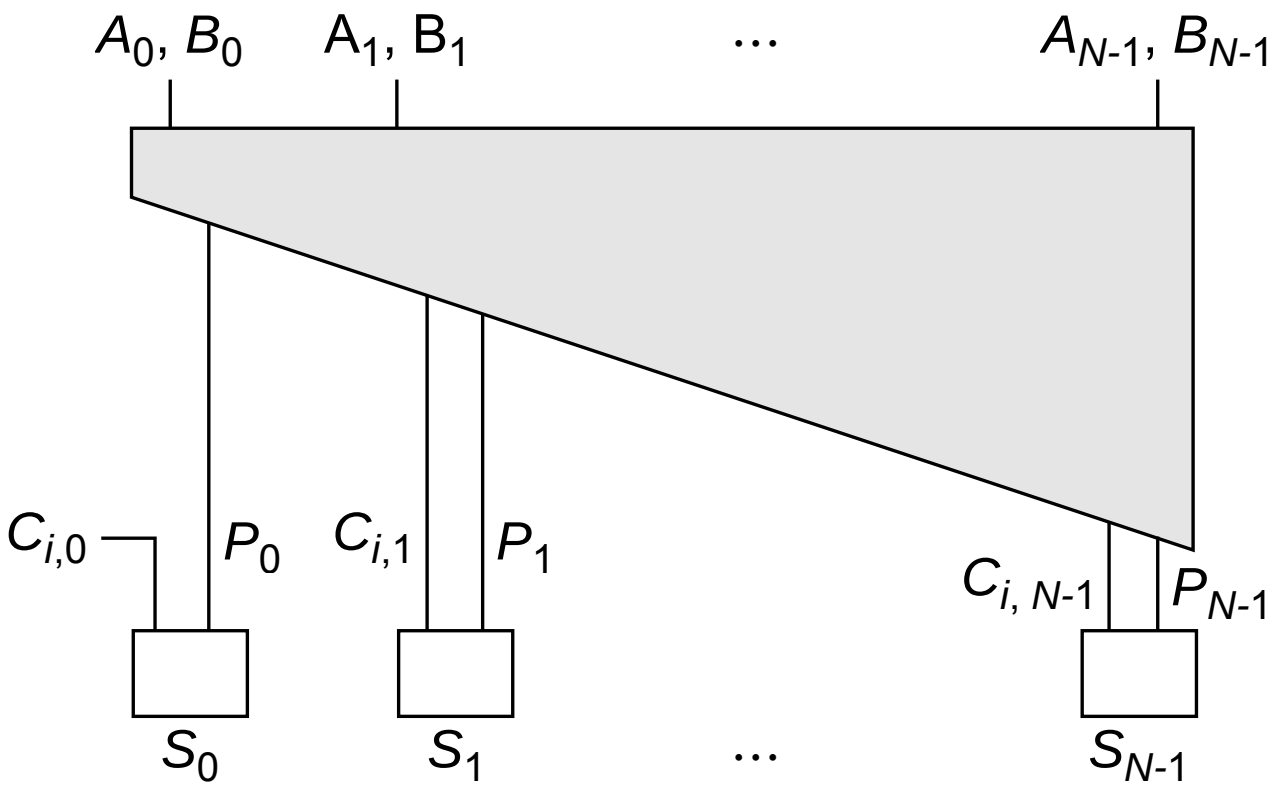
Expanding Lookahead equations:

$$C_{o,k} = G_k + P_k(G_{k-1} + P_{k-1} C_{o,k-2})$$

All the way:

$$C_{o,k} = G_k + P_k(G_{k-1} + P_{k-1}(\ldots + P_1(G_0 + P_0 C_{i,0})))$$

LookAhead - Basic Idea



$$C_{o,k} = f(A_k, B_k, C_{o,k-1}) = G_k + P_k C_{o,k-1}$$

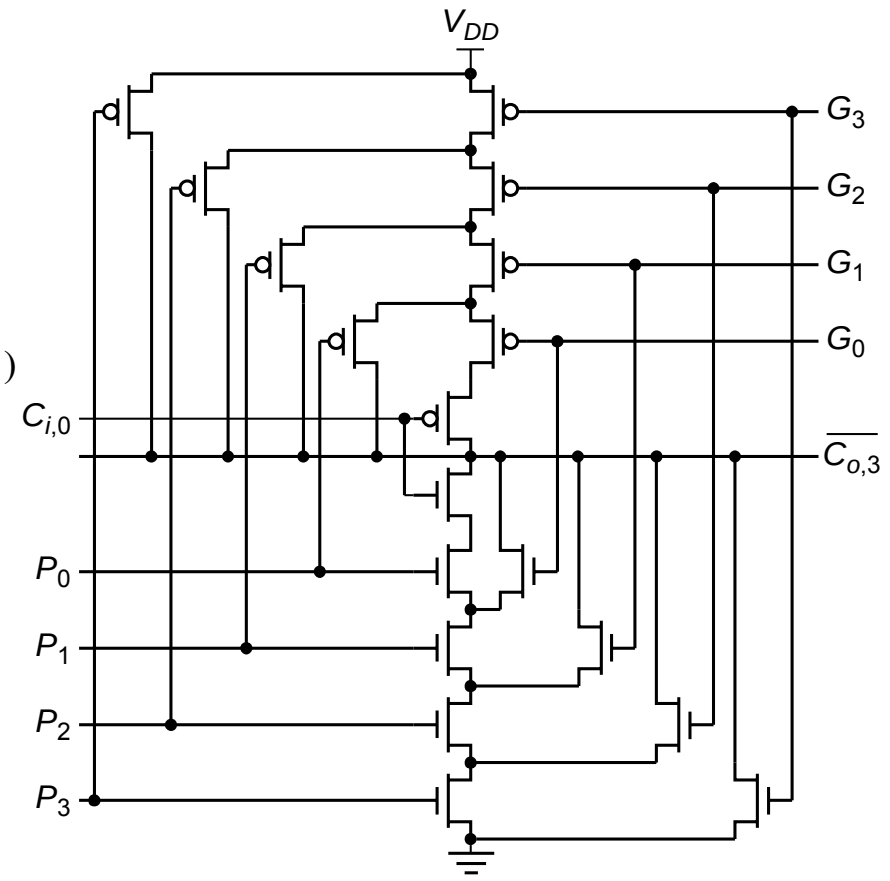
Look-Ahead: Topology

Expanding Lookahead equations:

$$C_{o,k} = G_k + P_k(G_{k-1} + P_{k-1}C_{o,k-2})$$

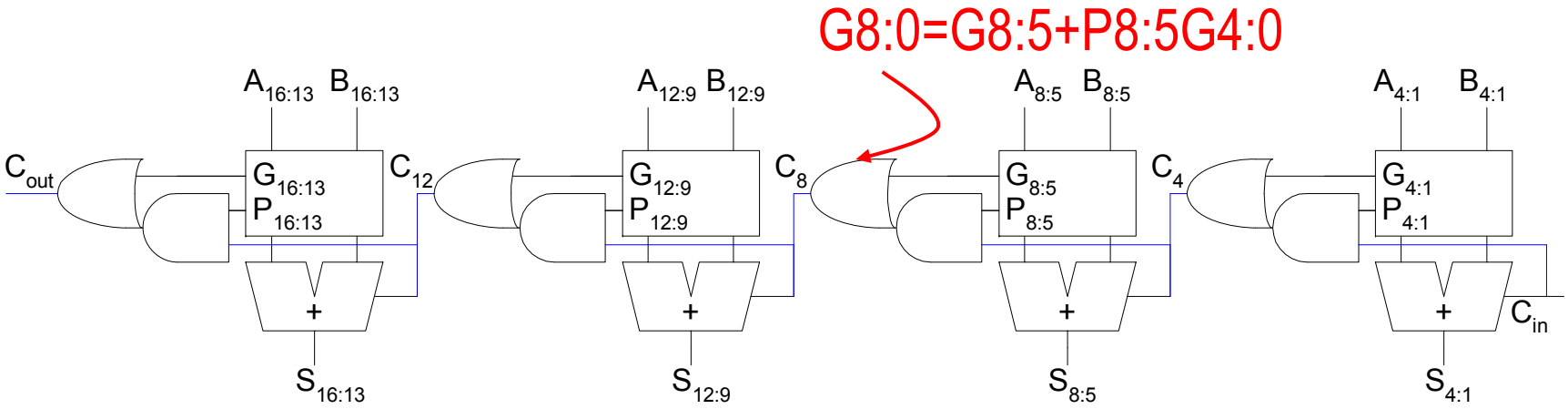
All the way:

$$C_{o,k} = G_k + P_k(G_{k-1} + P_{k-1}(\ldots + P_1(G_0 + P_0 C_{i,0})))$$



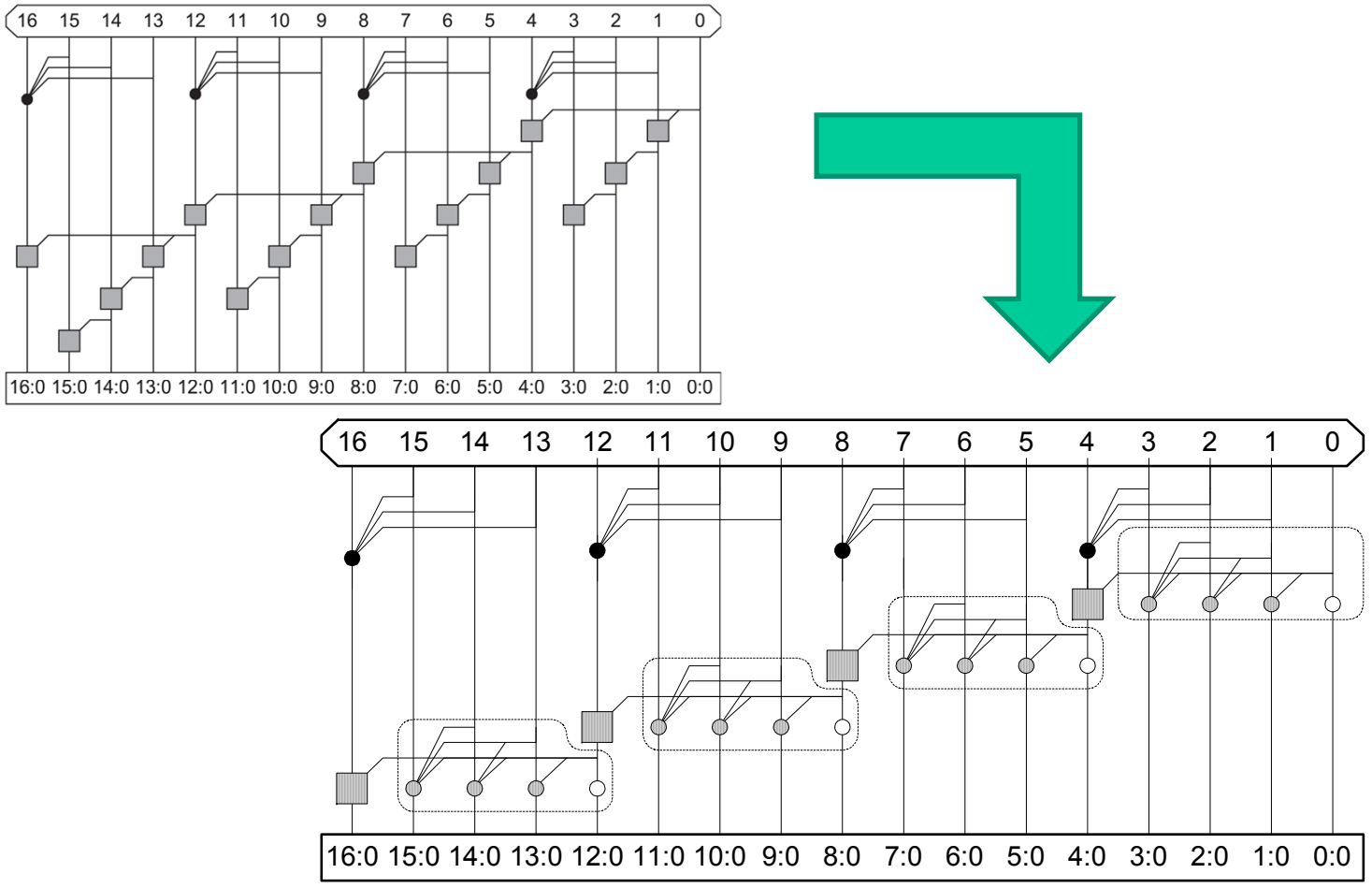
Carry-Lookahead Adder

- Carry-lookahead adder computes $G_{i:0}$ for many bits in parallel.
- Uses higher-valency cells with more than two inputs.



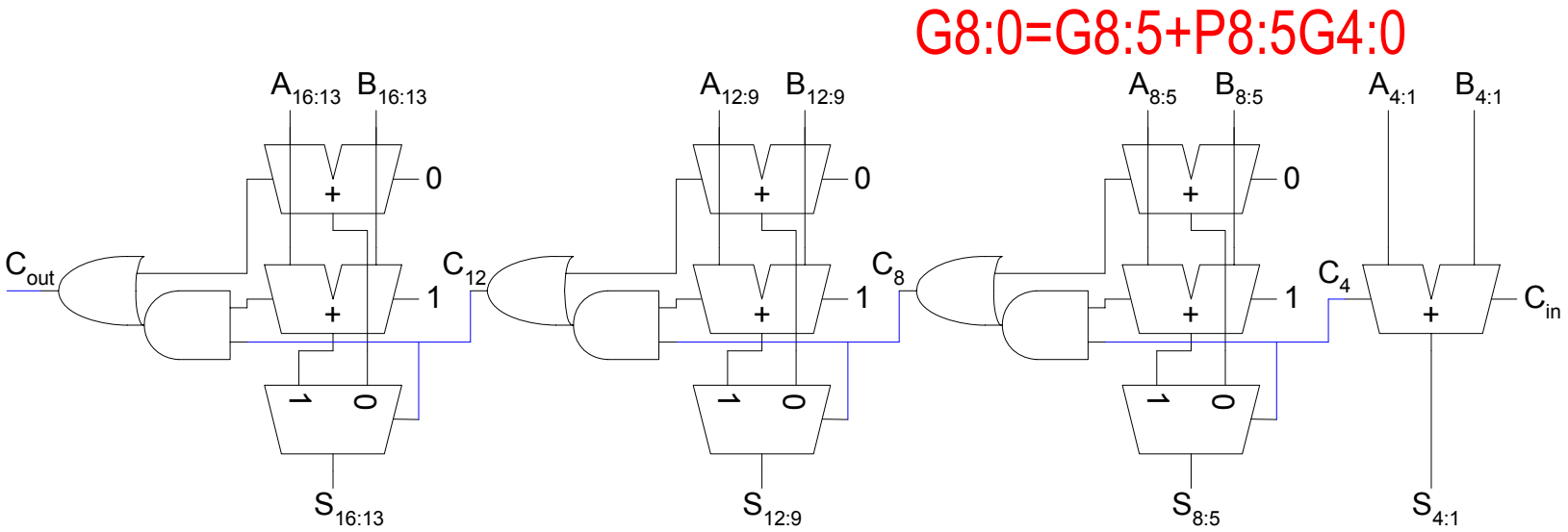
$$t_{cla} = t_{pg} + t_{pg(n)} + [(n-1) + (k-1)]t_{AO} + t_{xor}$$

CLA PG Diagram

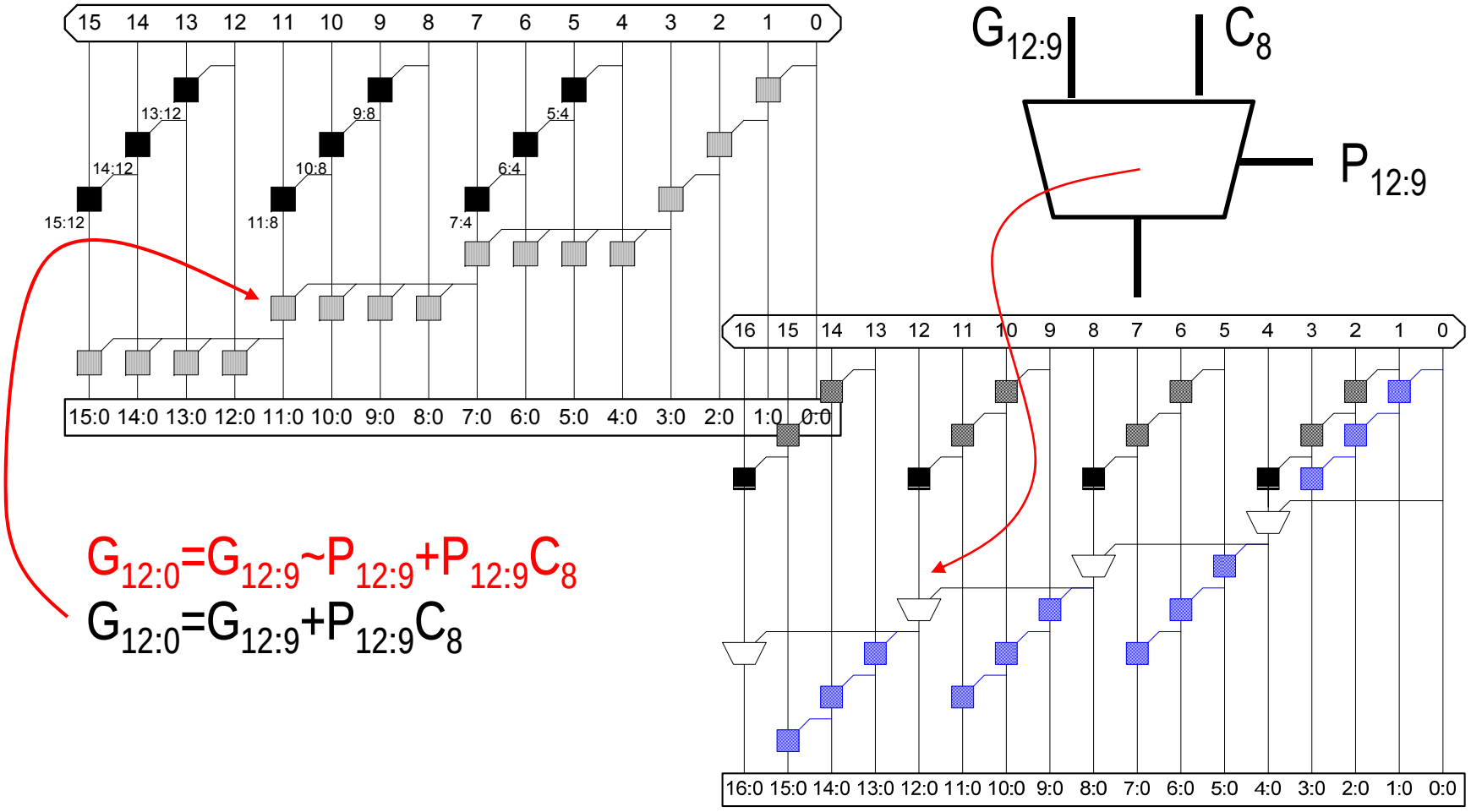


Carry-Select Adder

- Trick for critical paths dependent on late input X
  - Precompute two possible outputs for X = 0, 1
  - Select proper output when X arrives
- Carry-select adder precomputes n-bit sums
  - For both possible carries into n-bit group



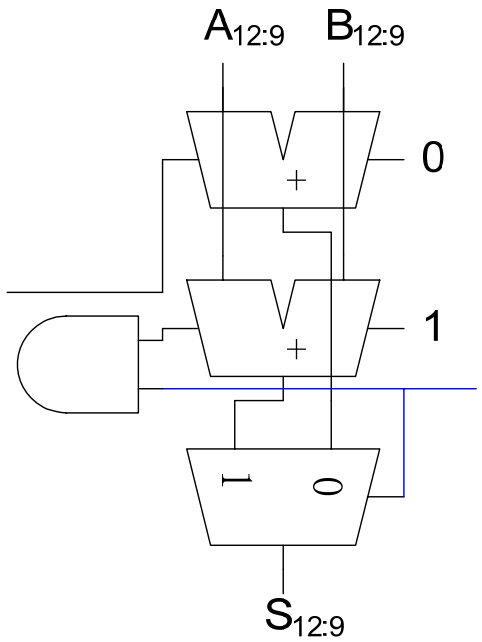
Carry-Increment Adder



$$G_{12:0}=G_{12:9}\sim P_{12:9}+P_{12:9}C_8$$

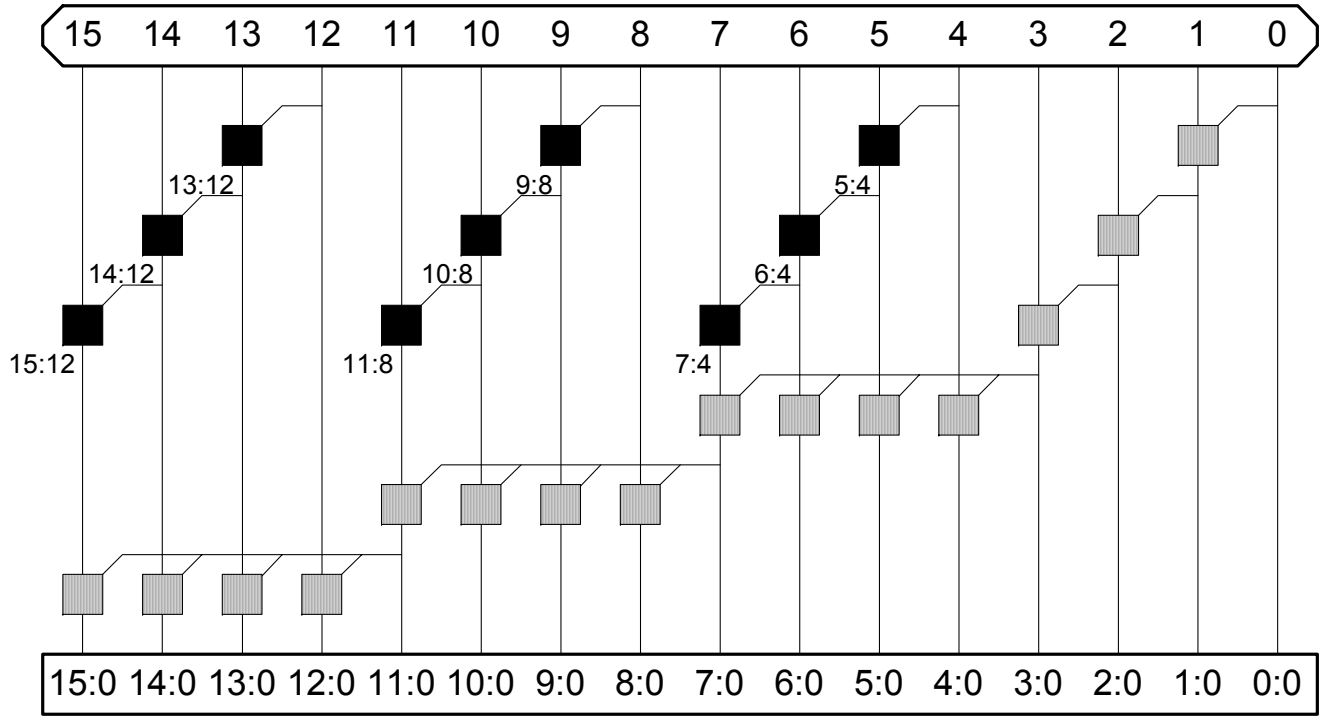
$$G_{12:0}=G_{12:9}+P_{12:9}C_8$$

$$G_{12:0}=G_{12:9}\sim C_8+(G_{12:9}+P_{12:9})C_8=G_{12:9}+P_{12:9}C_8$$



Carry-Increment Adder

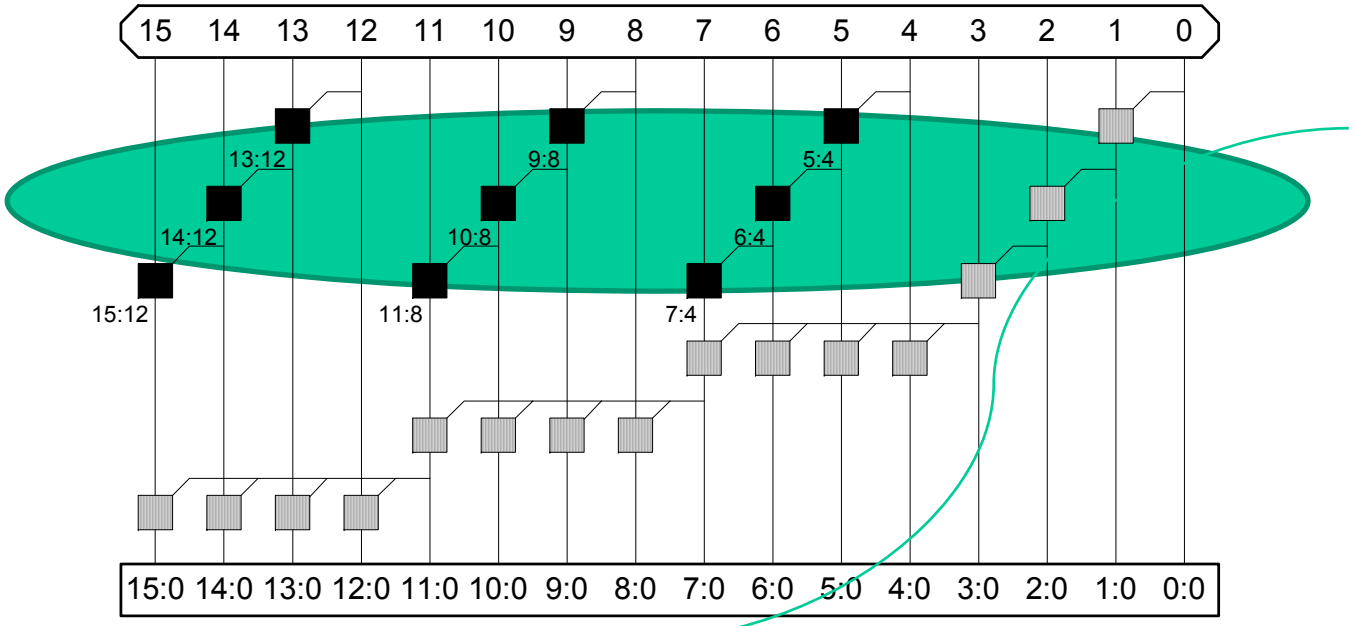
- Factor initial PG and final XOR out of carry-select



$$t_{\text{increment}} =$$

Carry-Increment Adder

- Factor initial PG and final XOR out of carry-select



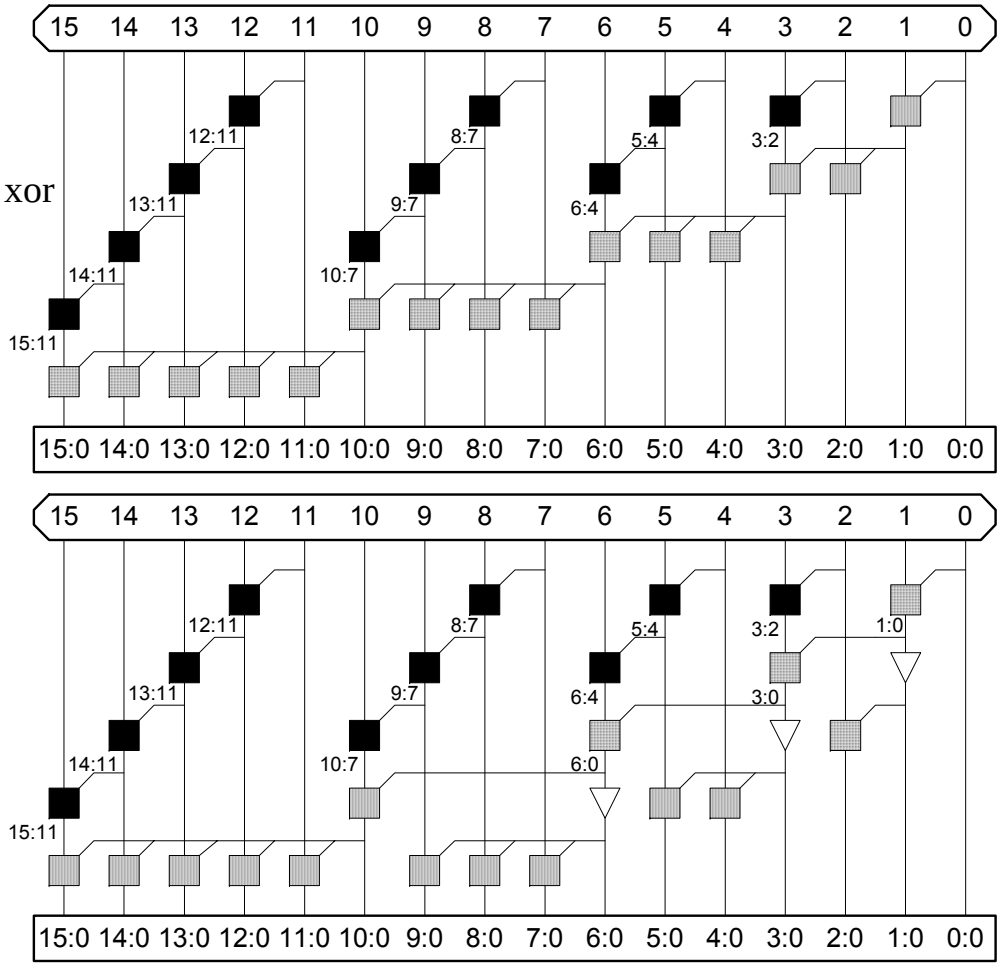
$$t_{\text{increment}} = t_{pg} + \left[(n-1) + (k-1)\right] t_{AO} + t_{\text{xor}}$$

$$t_{\text{increment}} = t_{\text{pg}} + [t_{\text{pg(n)}} + (k-1)]t_{\text{AO}} + t_{\text{xor}}$$

Variable Group Size

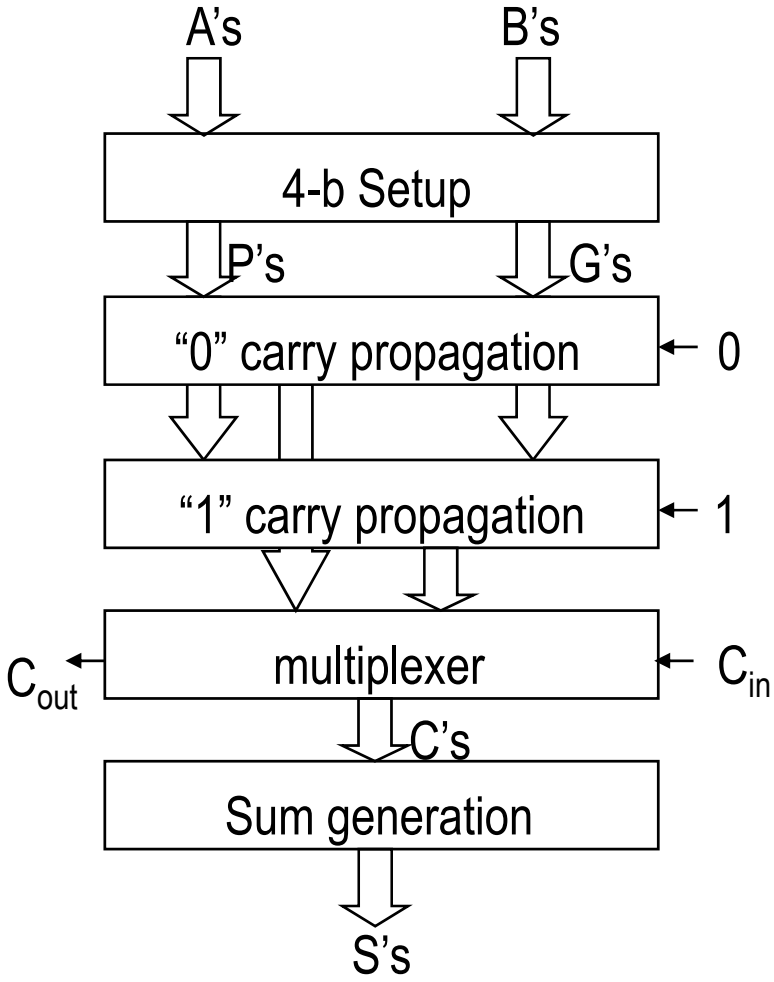
$$t_{increment} = t_{pg} + \sqrt{2N}\,t_{AO} + t_{xor}$$

Also buffer noncritical signals

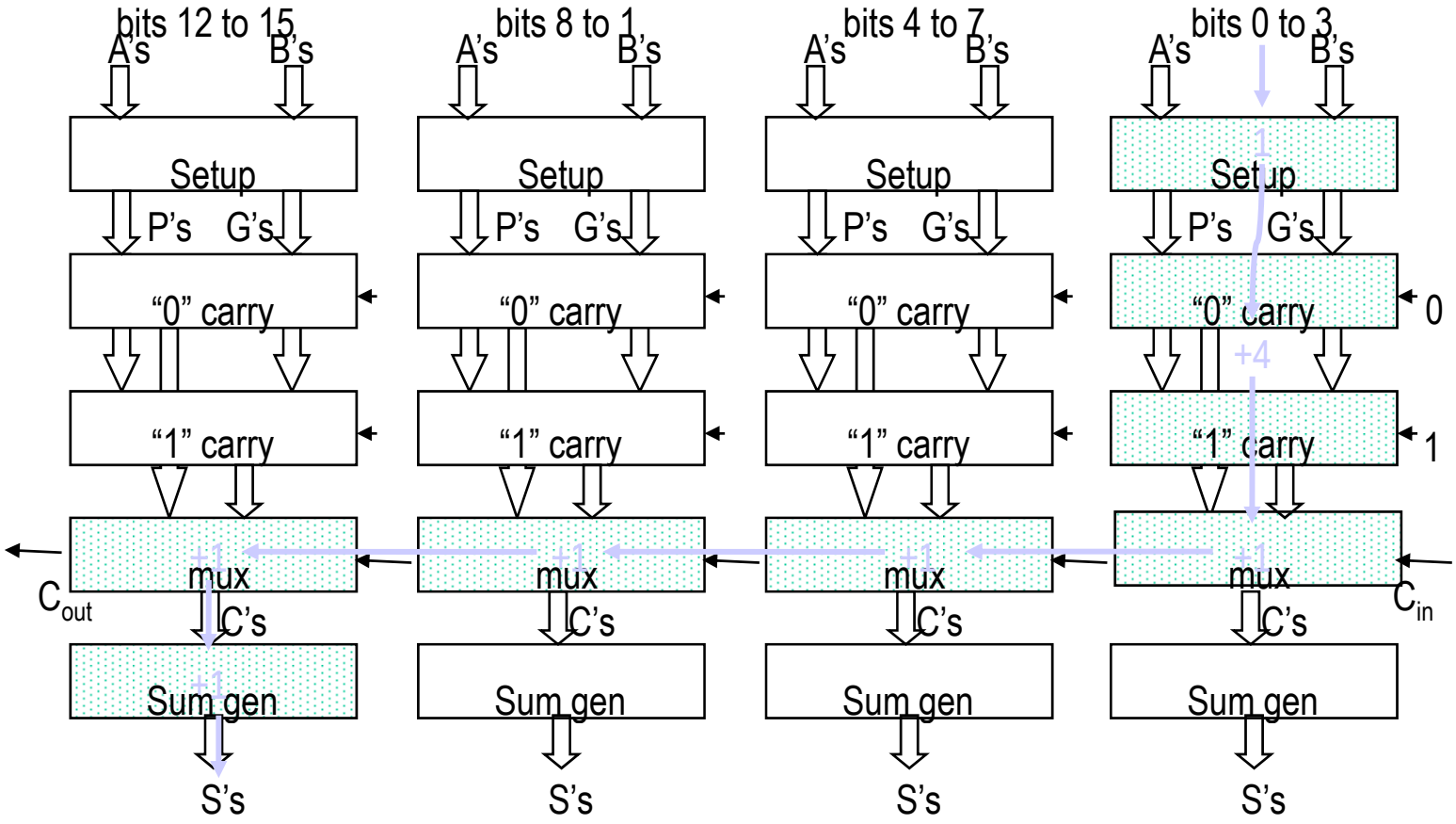


Carry Select Adder

- Precompute the carry out of each block for both carry_in = 0 and carry_in = 1 (can be done for all blocks in parallel) and then select the correct one

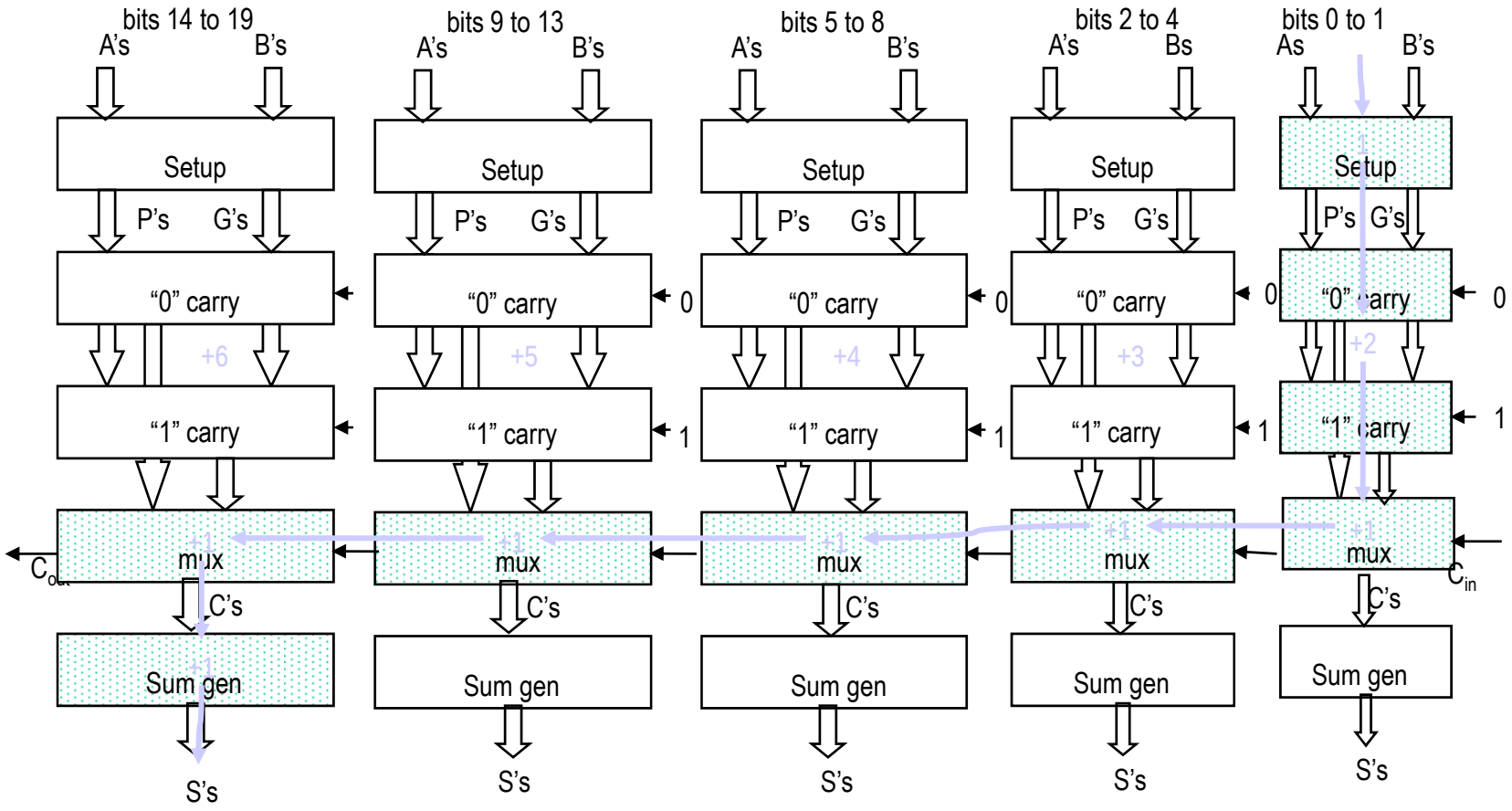


Carry Select Adder: Critical Path



$$T_{add} = t_{setup} + B\,t_{carry} + N/B\,t_{mux} + t_{sum}$$

Square Root Carry Select Adder



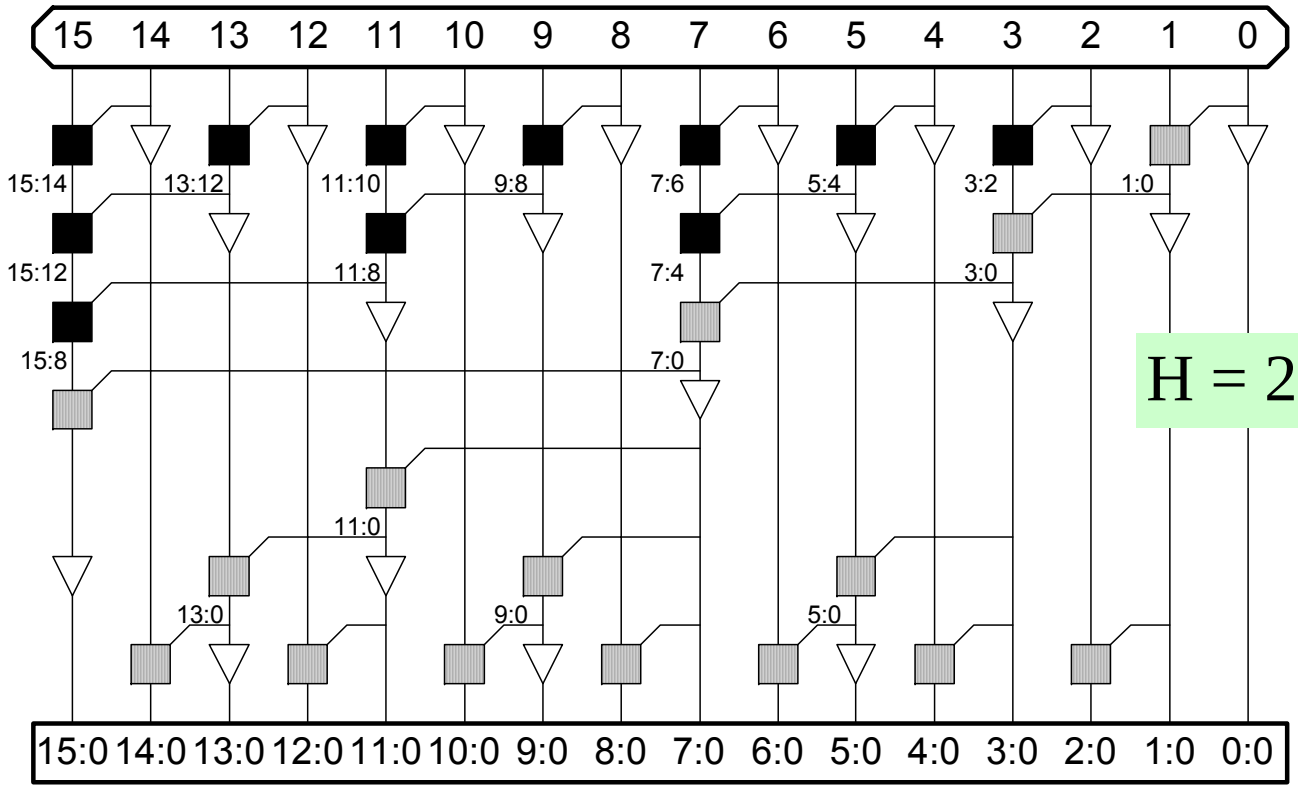
$$T_{add} = t_{setup} + 2\,t_{carry} + \sqrt{2N}\,t_{mux} + t_{sum}$$

Tree Adder

- If lookahead is good, lookahead across lookahead!
  - Recursive lookahead gives O(log N) delay
- Many variations on tree adders

Brent-Kung*

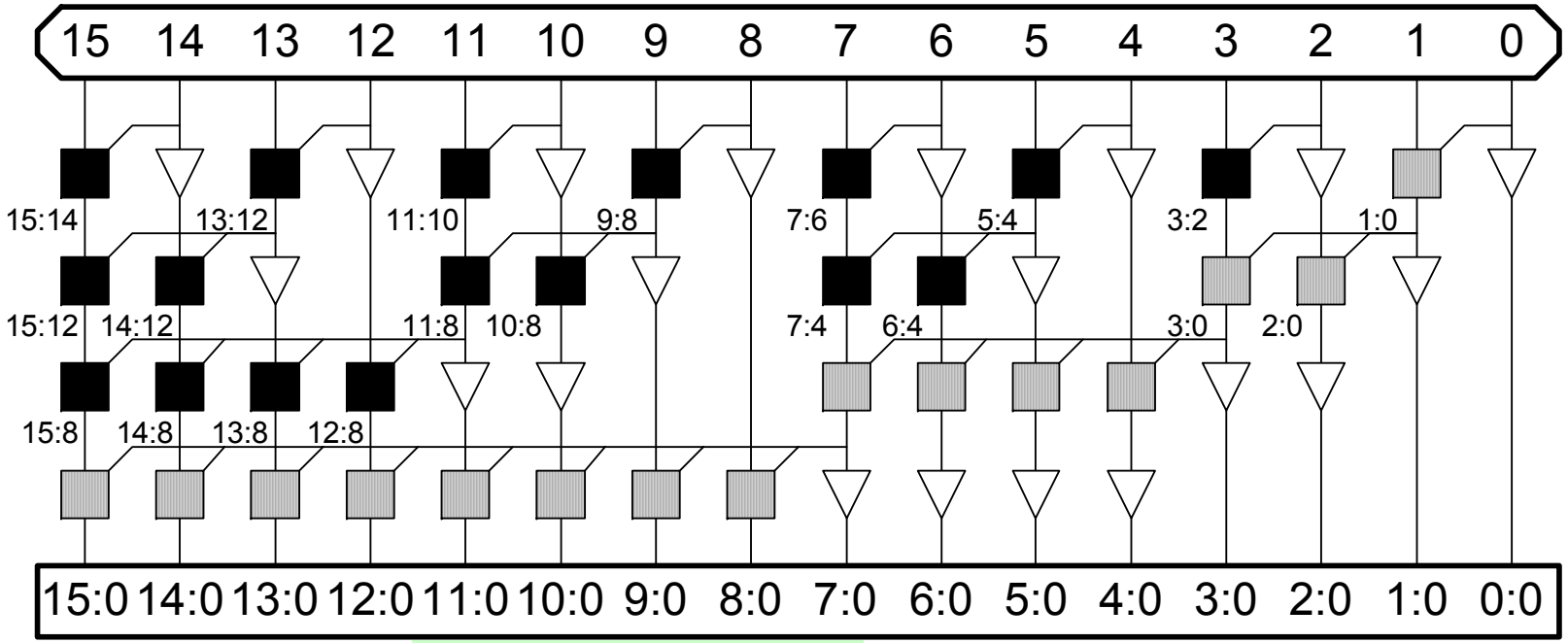
*R.Brent and H.Kung,"a regular layout for parallel adders" IEEE Trans. Computer, vol. C-31,No.3,March 1982,pp.260-264



$$H = 2\log_2 N - 1$$

Sklansky*

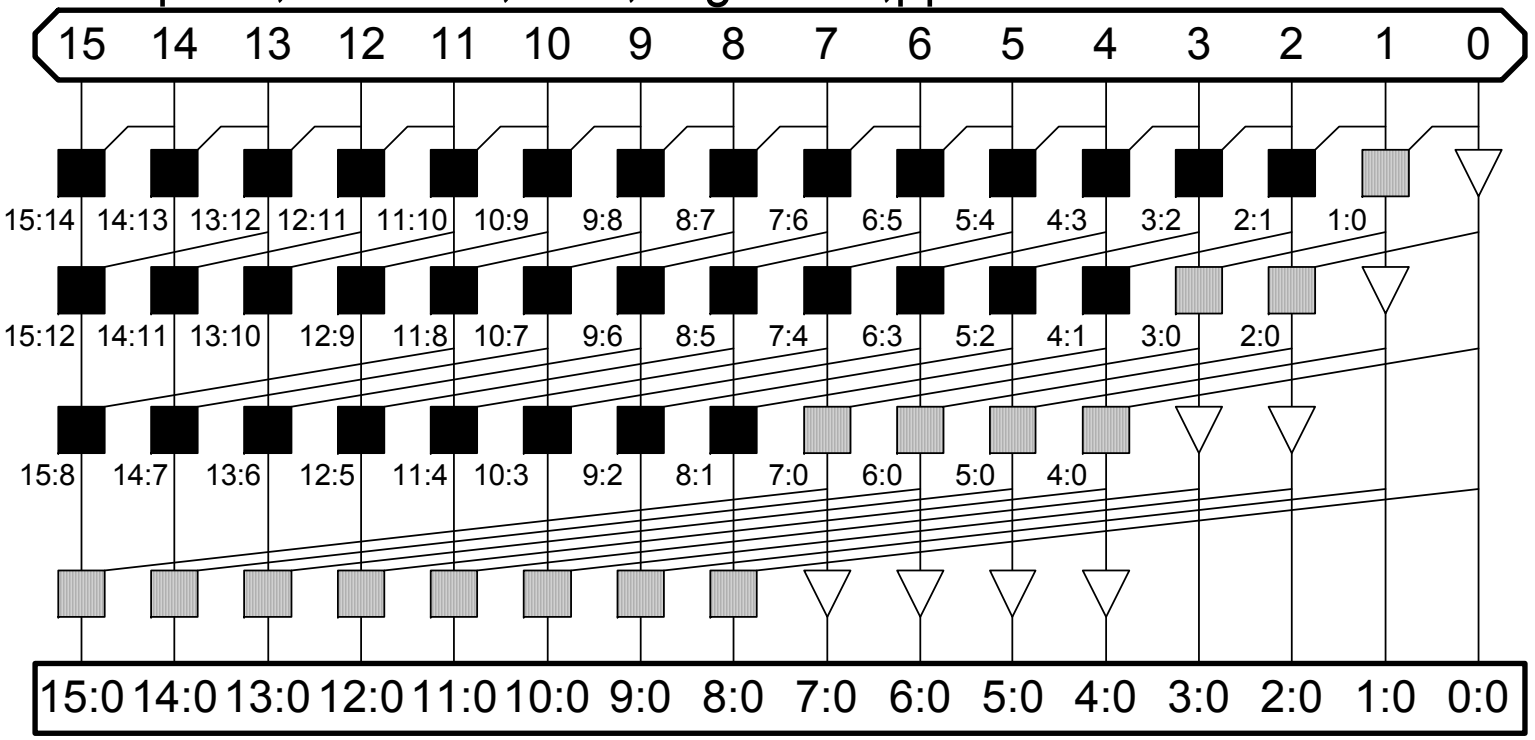
*J.Sklansky ”conditional-sum addition logic” IER Trans. Electronic computers, vol.EC-9,June 1960,pp.226-231



fanout = [8,4,2,1]

Kogge-Stone*

*P.Kogge and H.Stone,"a parallel algorithm for the efficient solution of a general class of recurrence equations" IEEE Trans. Computer, vol.C-22,No.8,Aug. 1973,pp.786-793



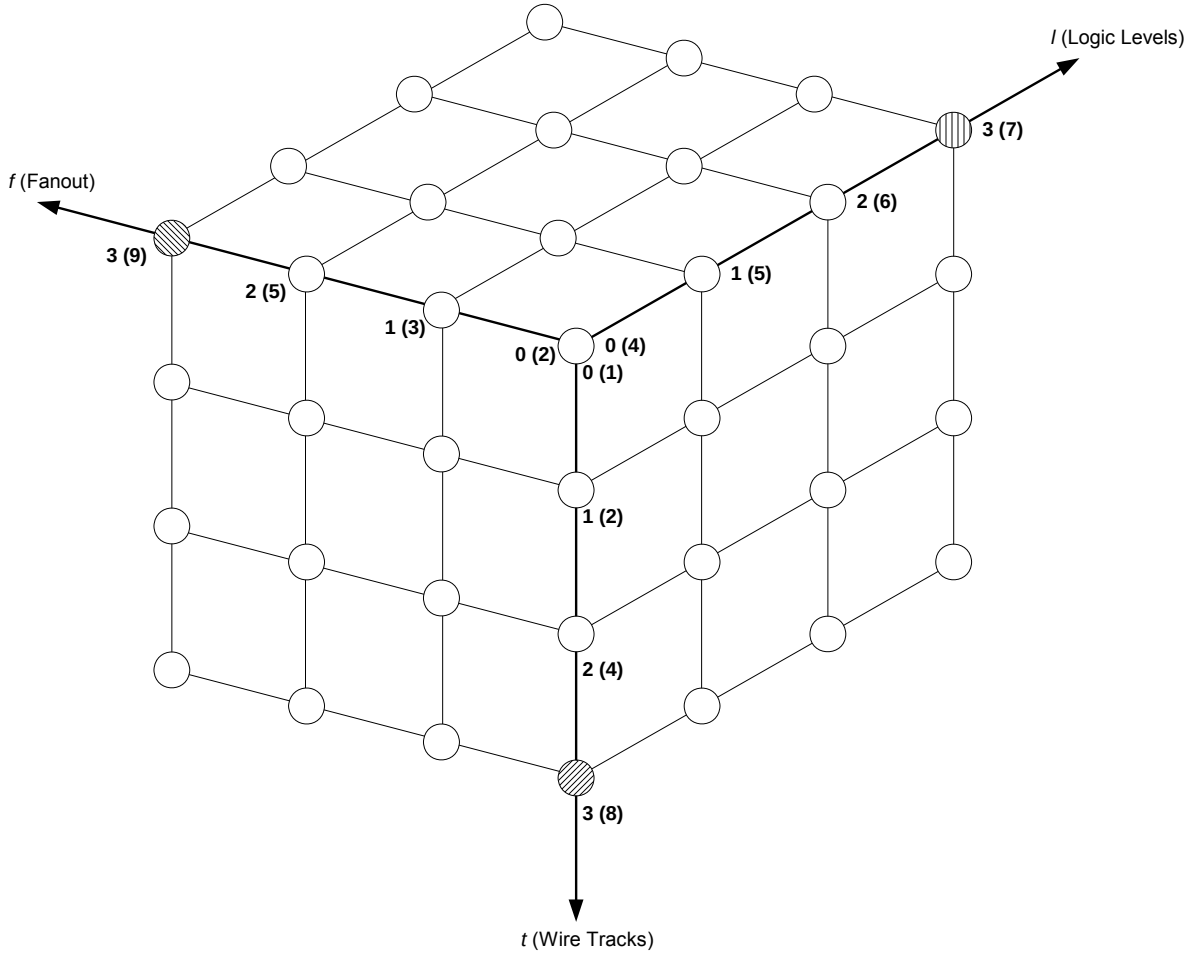
Tree Adder Taxonomy

- Describe adder with 3-D taxonomy (l, f, t)
  - Logic levels: $L + l$
  - Fanout: $2^f + 1$
  - Wiring tracks: 2^t
- Known tree adders sit on plane defined by

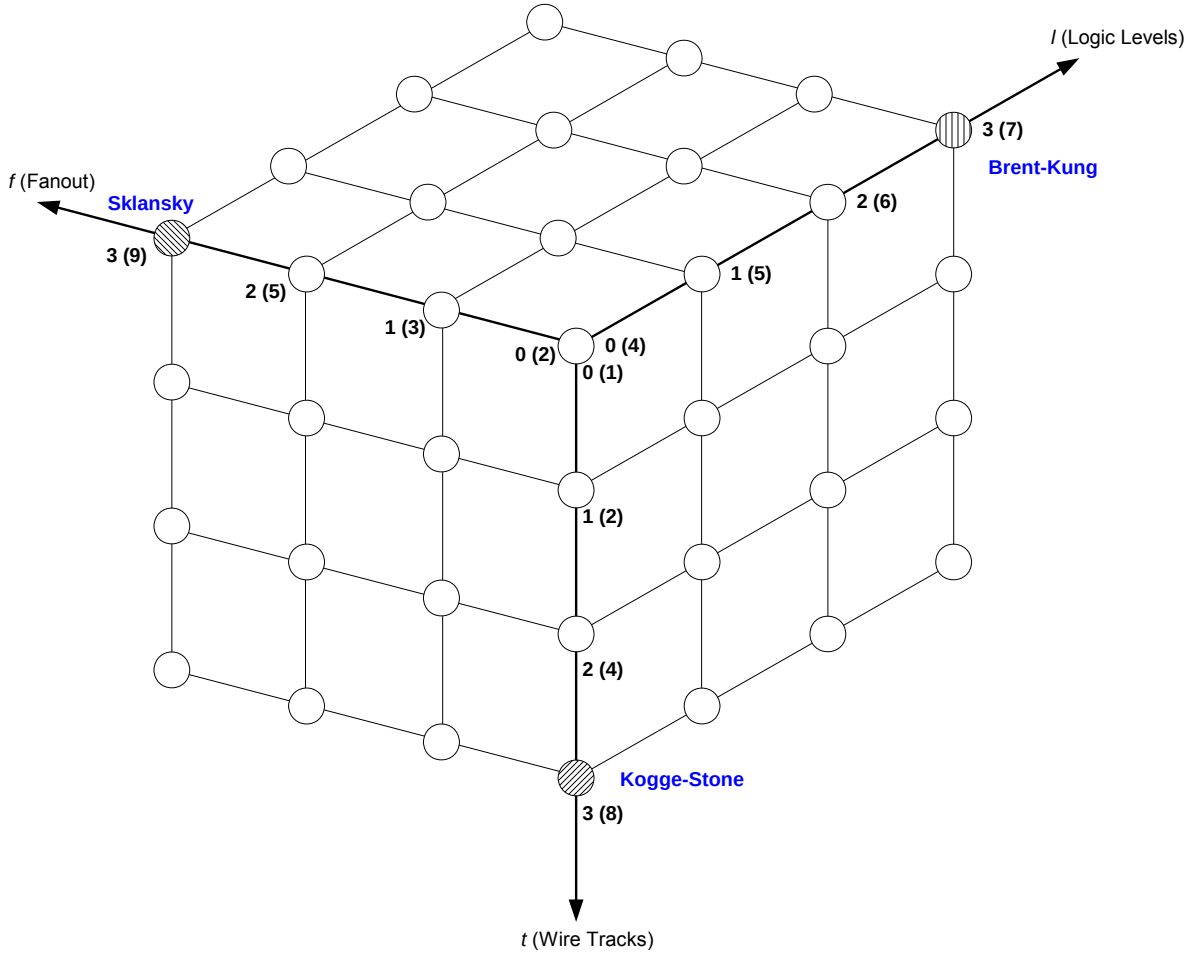
$$l + f + t = L\text{-}1$$

L = log N logic levels
Fanout never exceeding 2
No more than one wiring track between levels

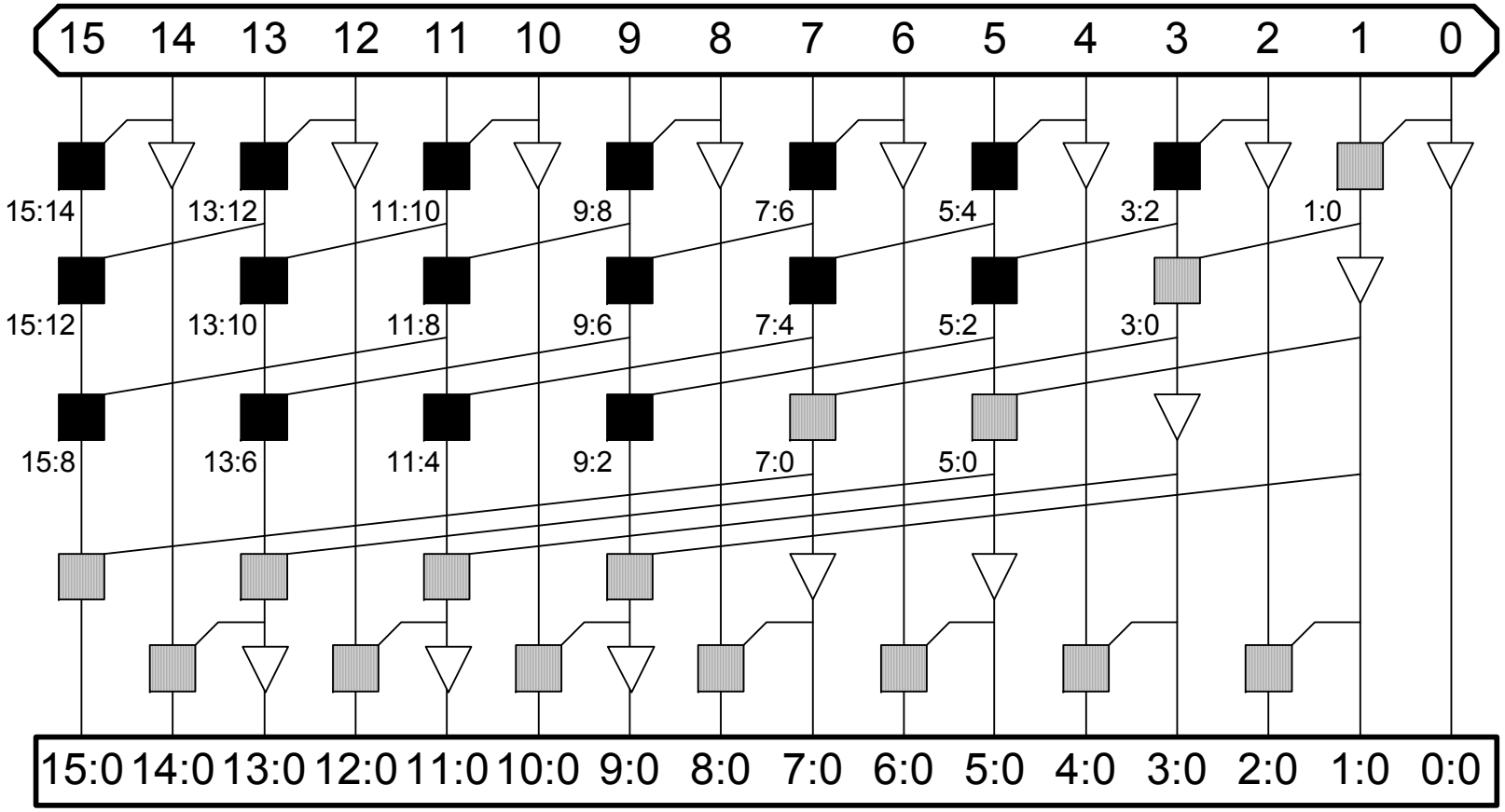
Tree Adder Taxonomy



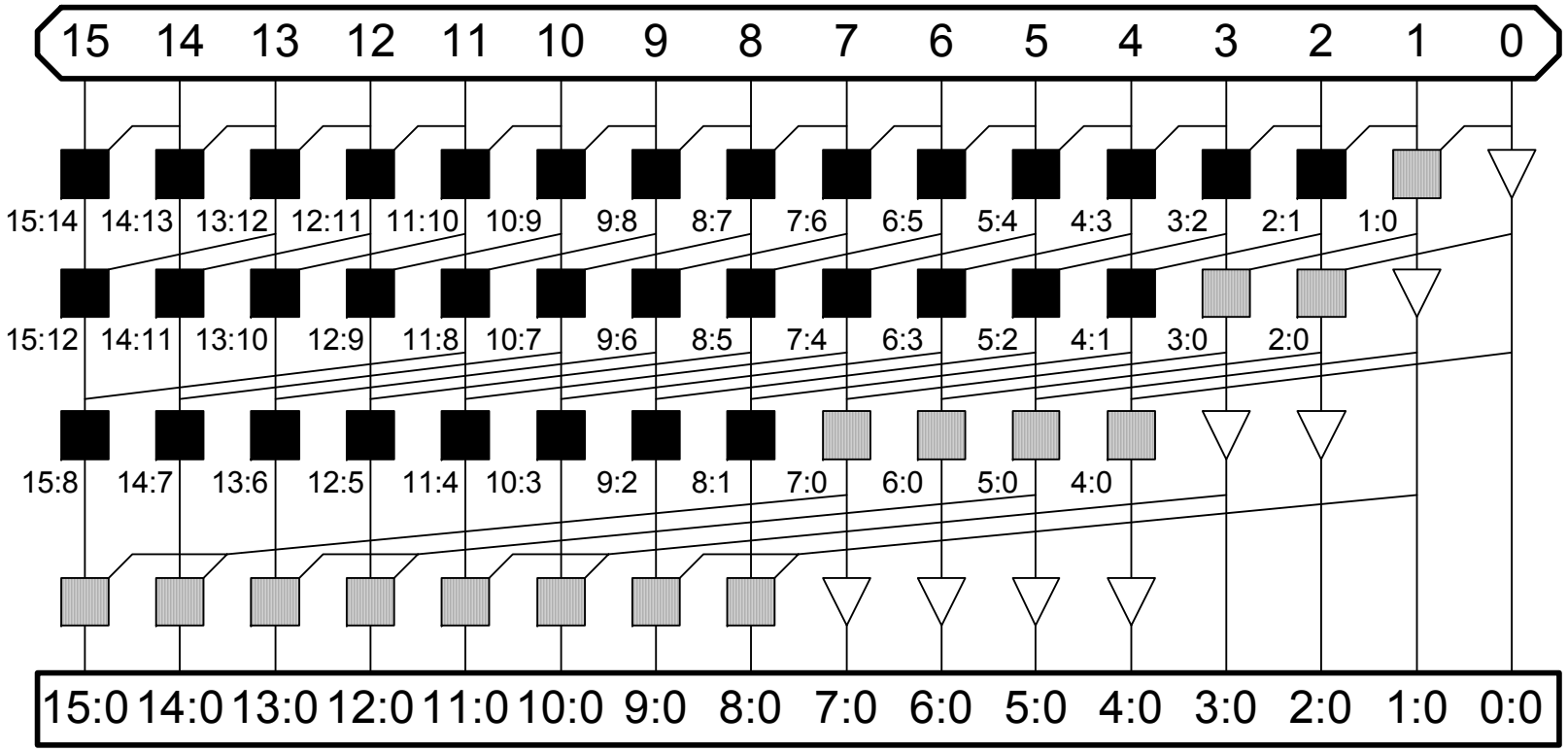
Tree Adder Taxonomy



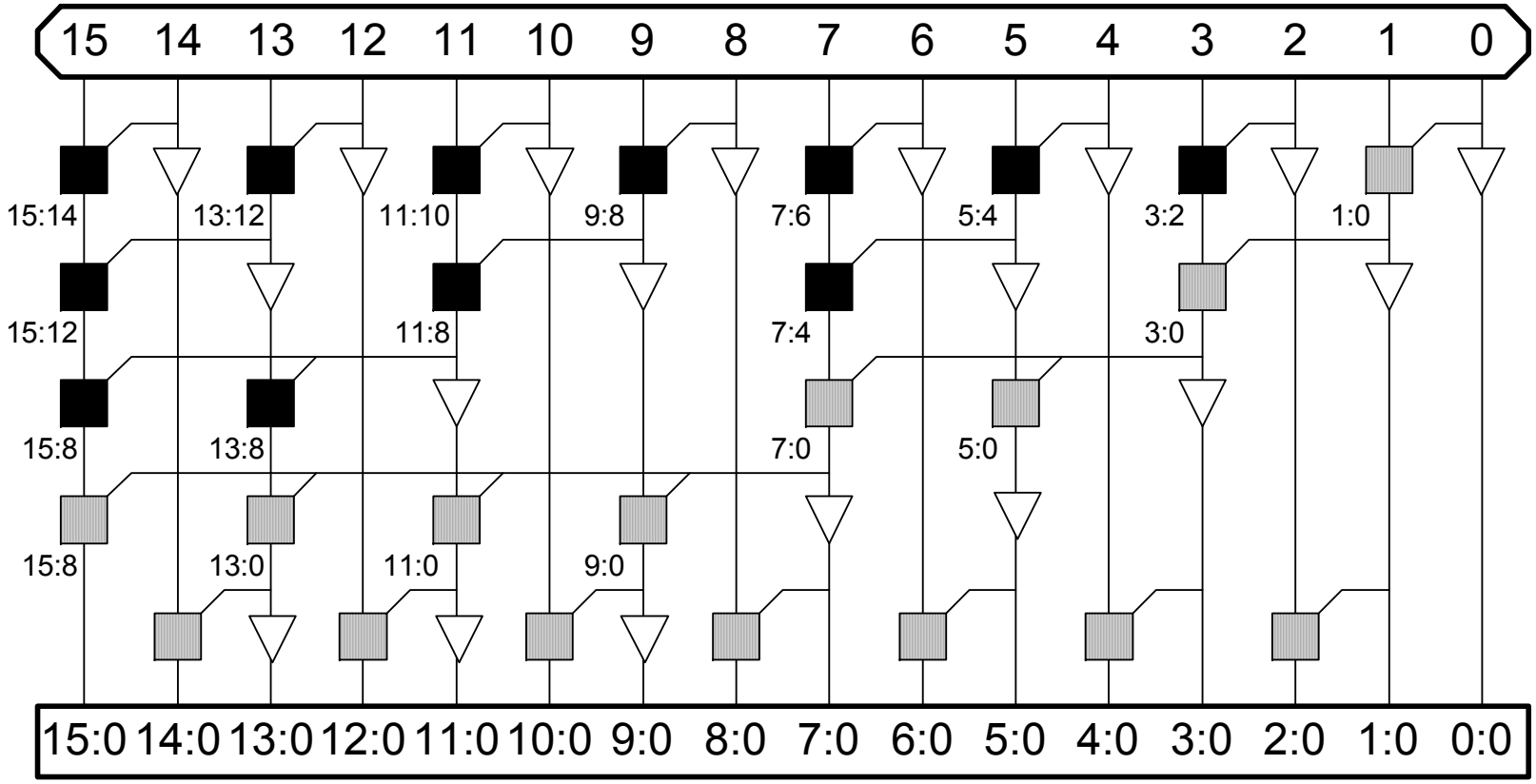
Han-Carlson


15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
15:14 13:12 11:10 9:8 7:6 5:4 3:2 1:0
15:12 13:10 11:8 9:6 7:4 5:2 3:0
15:8 13:6 11:4 9:2 7:0 5:0
15:0 14:0 13:0 12:0 11:0 10:0 9:0 8:0 7:0 6:0 5:0 4:0 3:0 2:0 1:0 0:0

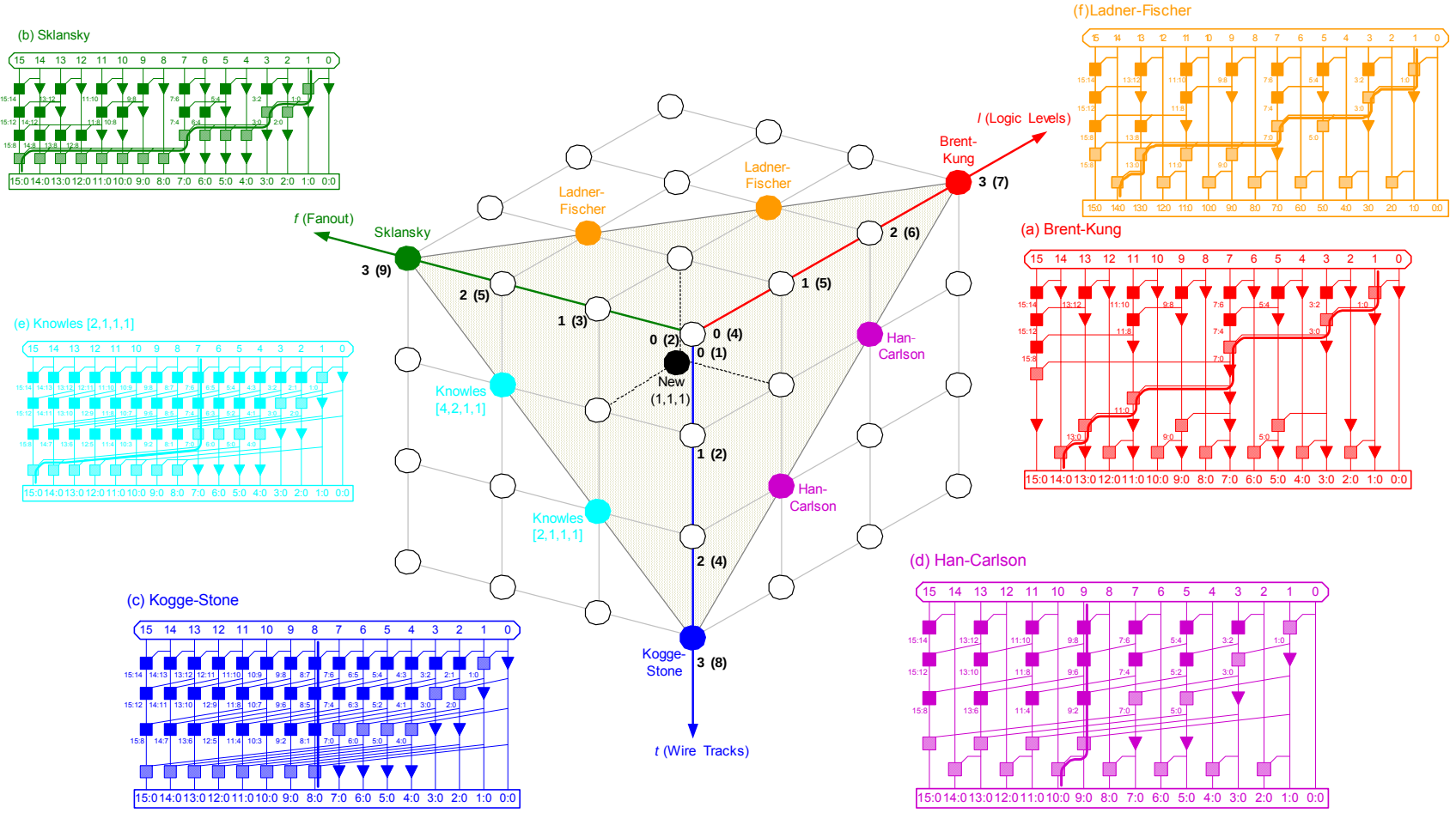
Knowles [2, 1, 1, 1]



Ladner-Fischer



Taxonomy Revisited



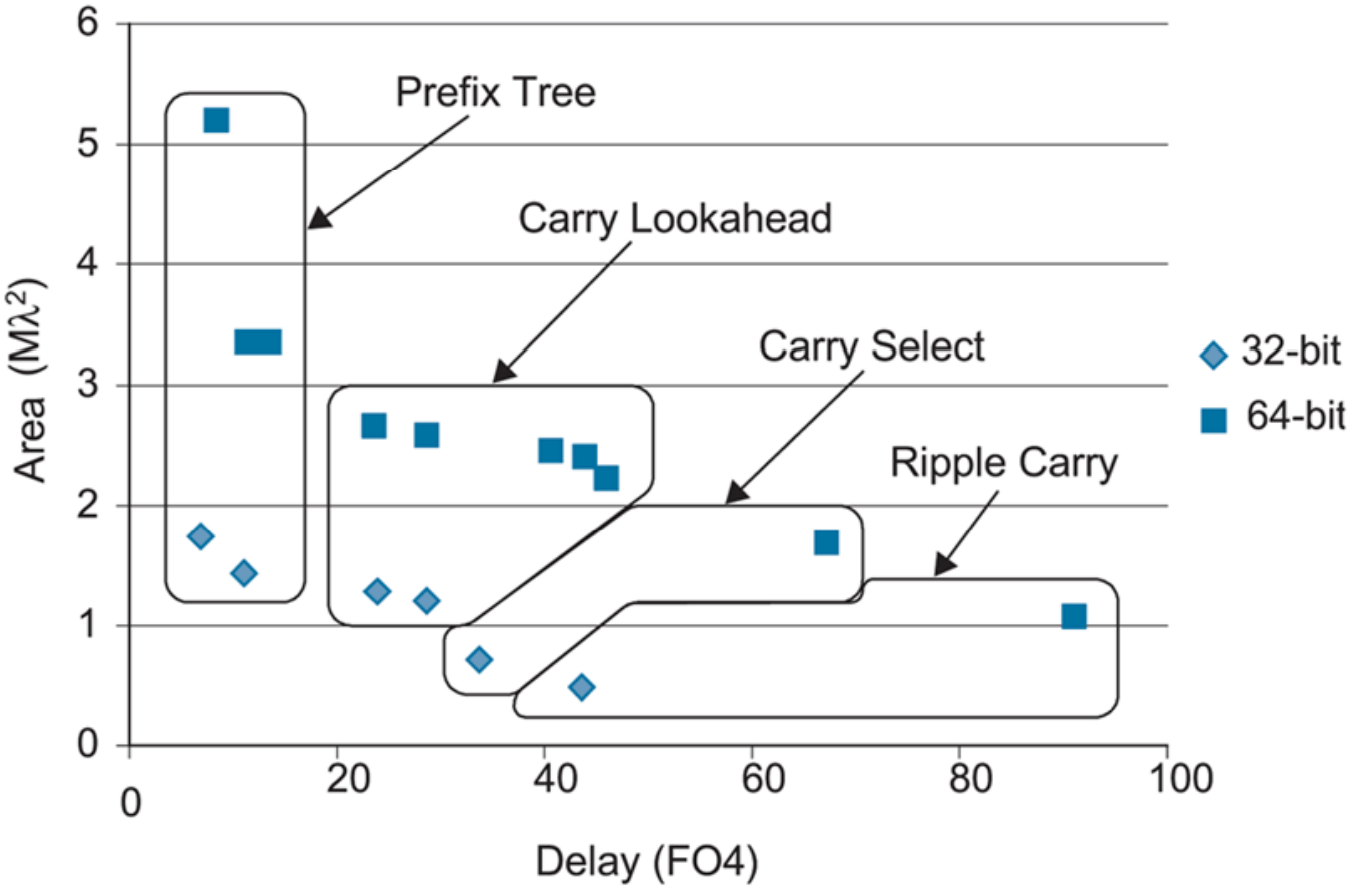
Summary

Adder architectures offer area / power / delay tradeoffs.

Choose the best one for your application.

Architecture	Classification	Logic Levels	Max Fanout	Tracks	Cells
Carry-Ripple		N-1	1	1	N
Carry-Skip n=4		N/4 + 5	2	1	1.25N
Carry-Inc. n=4		N/4 + 2	4	1	2N
Brent-Kung	(L-1, 0, 0)	$2\log_2 N - 1$	2	1	2N
Sklansky	(0, L-1, 0)	$\log_2 N$	N/2 + 1	1	$0.5\ N\log_2 N$
Kogge-Stone	(0, 0, L-1)	$\log_2 N$	2	N/2	$N\log_2 N$

summary


Prefix Tree
Carry Lookahead
Carry Select
Ripple Carry
32-bit
64-bit
Area (Mλ²)
Delay (FO4)
0
1
2
3
4
5
6
0
20
40
60
80
100