Adder Architectures

Dr. Shubhajit Roy Chowdhury,

Centre for VLSI and Embedded Systems Technology,
IIIT Hyderabad, India

Email: src.vlsi@iiit.ac.in

Types of Adders

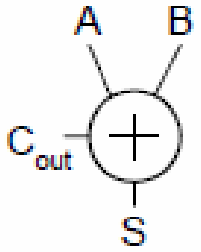
- Single-bit Addition
- Carry-Ripple Adder
- Carry-Skip Adder
- Carry-Lookahead Adder
- Carry-Select Adder
- Tree Adder
- Higher Valency Trees

Single bit addition

Half Adder

$S =$

$C_{out} =$

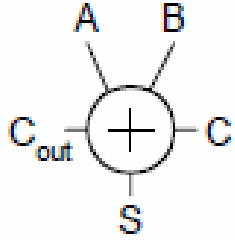


A	B	C_{out}	S
0	0		
0	1		
1	0		
1	1		

Full Adder

$S =$

$C_{out} =$



A	B	C	C_{out}	S
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Propagate, Generate and Kill

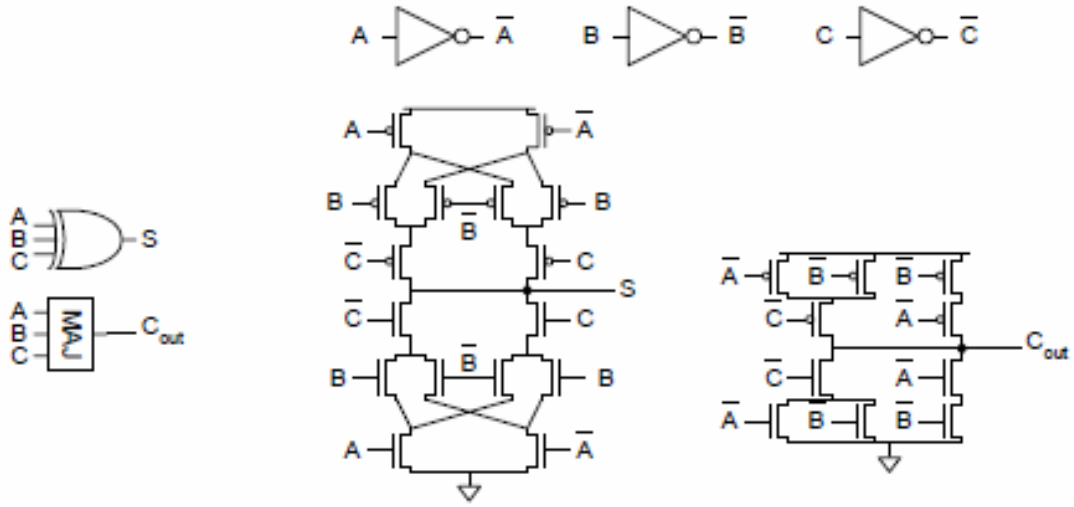
- For a full adder, define what happens to carries
  - Generate: C_{out} = 1 independent of C
    - G =
  - Propagate: C_{out} = C
    - P =
  - Kill: C_{out} = 0 independent of C
    - K =

Full Adder Design I (using 32 transistors)

- Brute force implementation from eqns

$$S = A \oplus B \oplus C$$

$$C_{out} = MAJ(A, B, C)$$

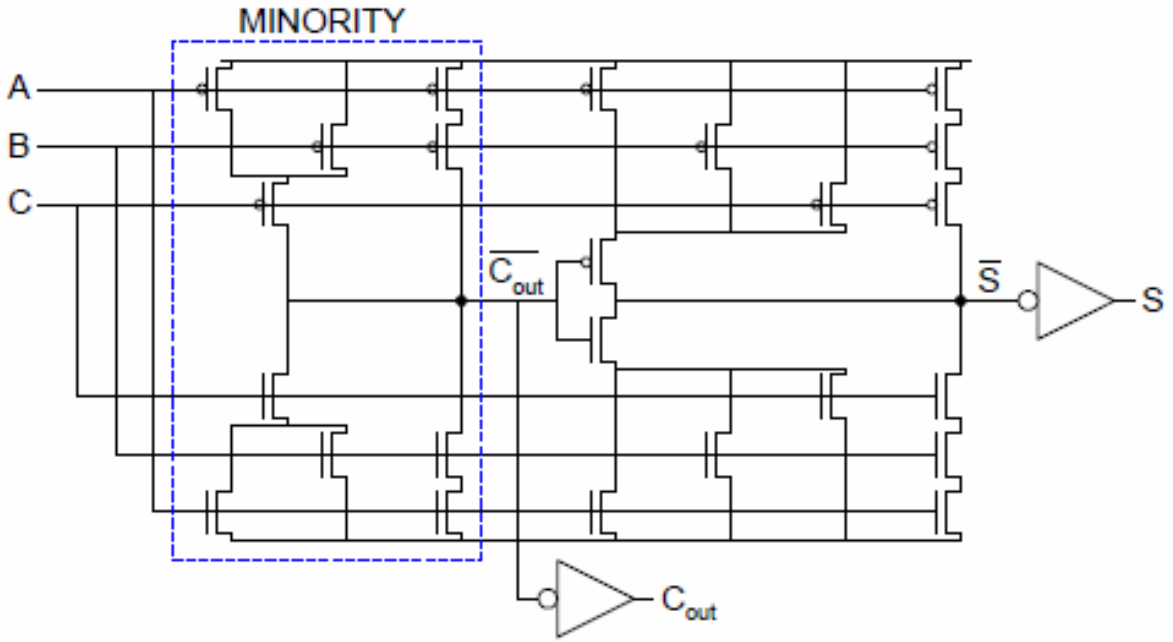


Full Adder Design II (using 28 transistors)

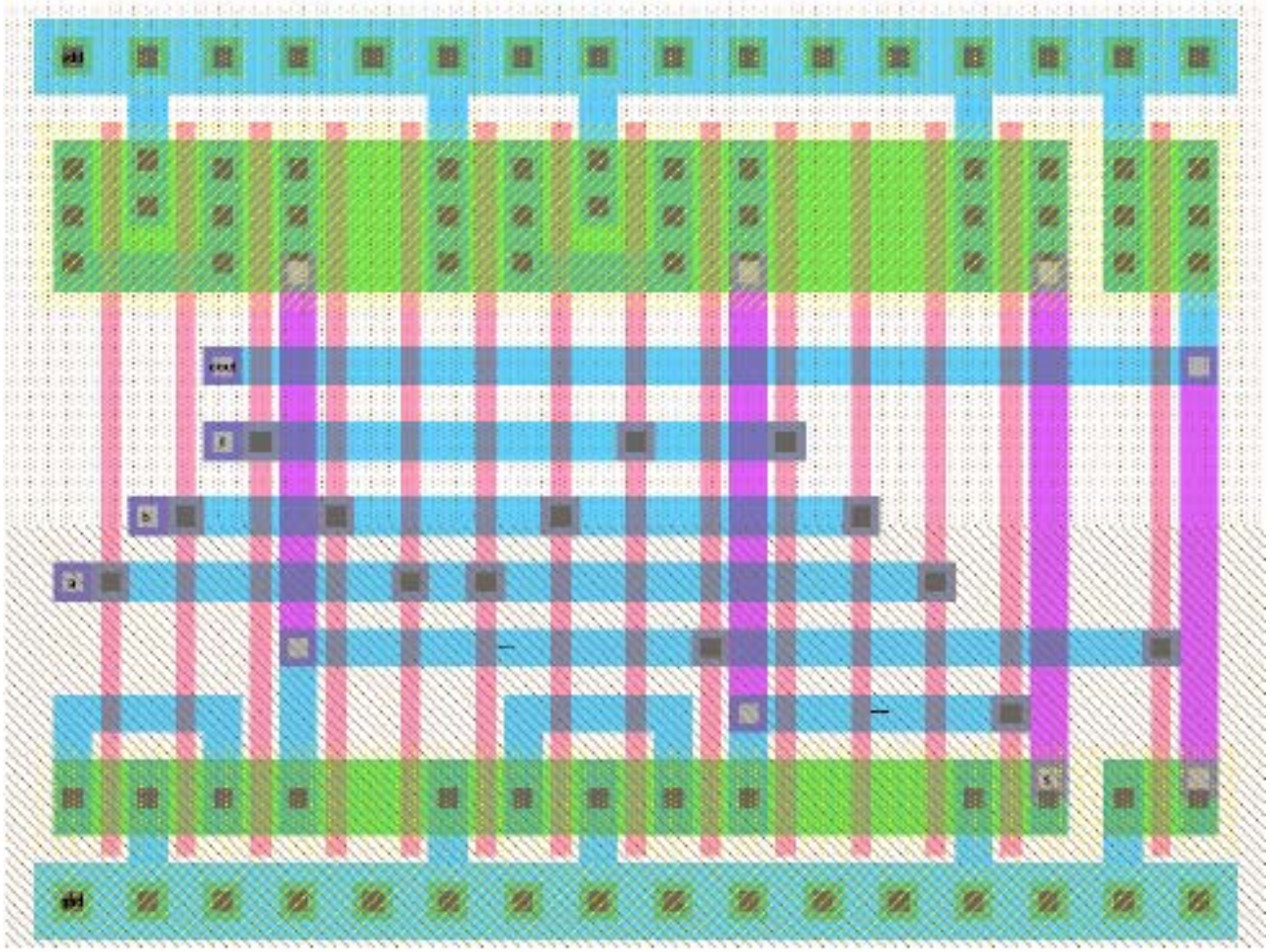
- Factor S in terms of C_{out}

  $S = ABC + (A + B + C)(\sim C_{out})$

- Critical path is usually C to C_{out} in ripple adder

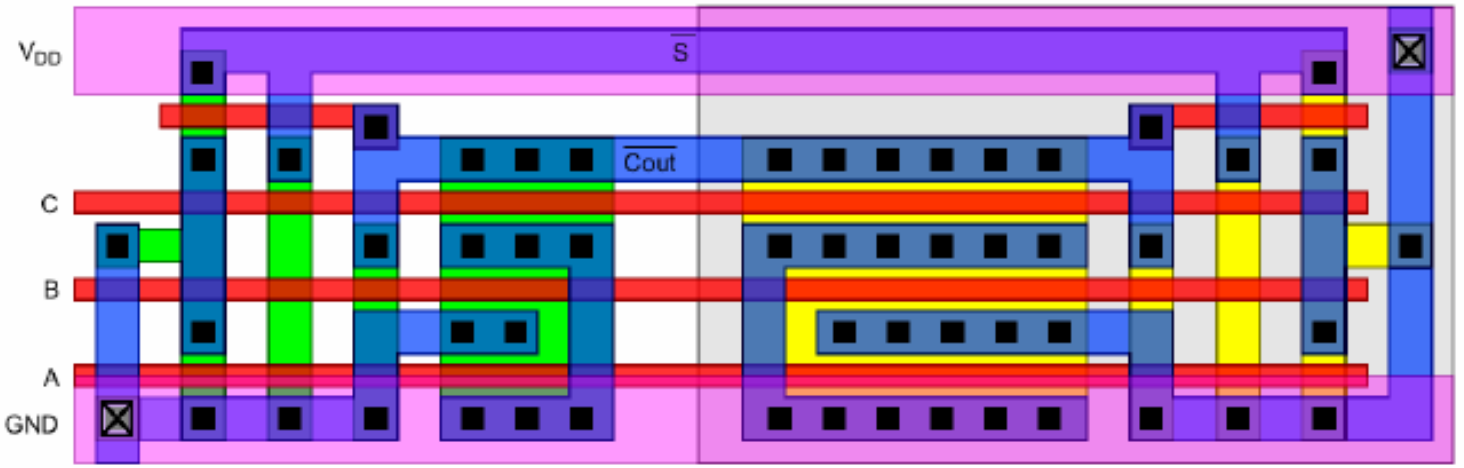


Layout of 28 transistor Full Adder



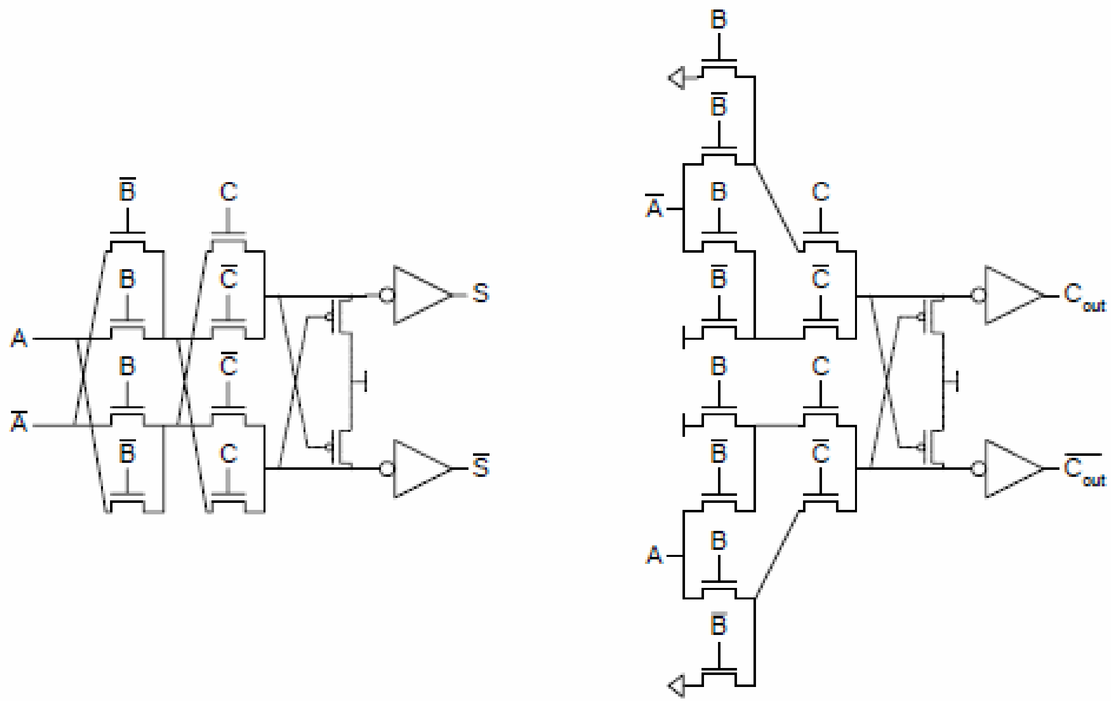
Modified Layout of 28T full adder

- Clever layout circumvents usual line of diffusion
  - Use wide transistors on critical path
  - Eliminate output inverters



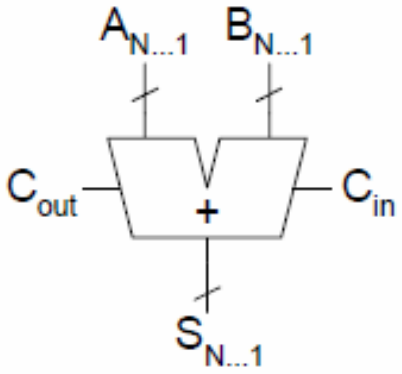
Full Adder Design III

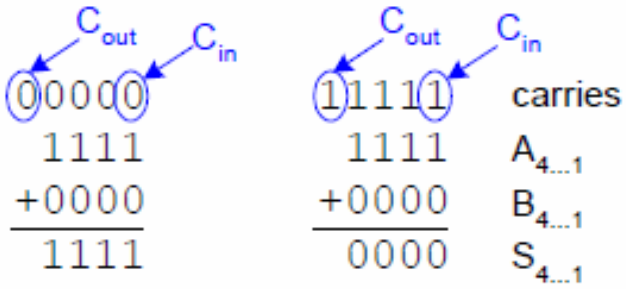
- Complementary Pass Transistor Logic (CPL)
  - Slightly faster, but more area



Carry Propagate Adder

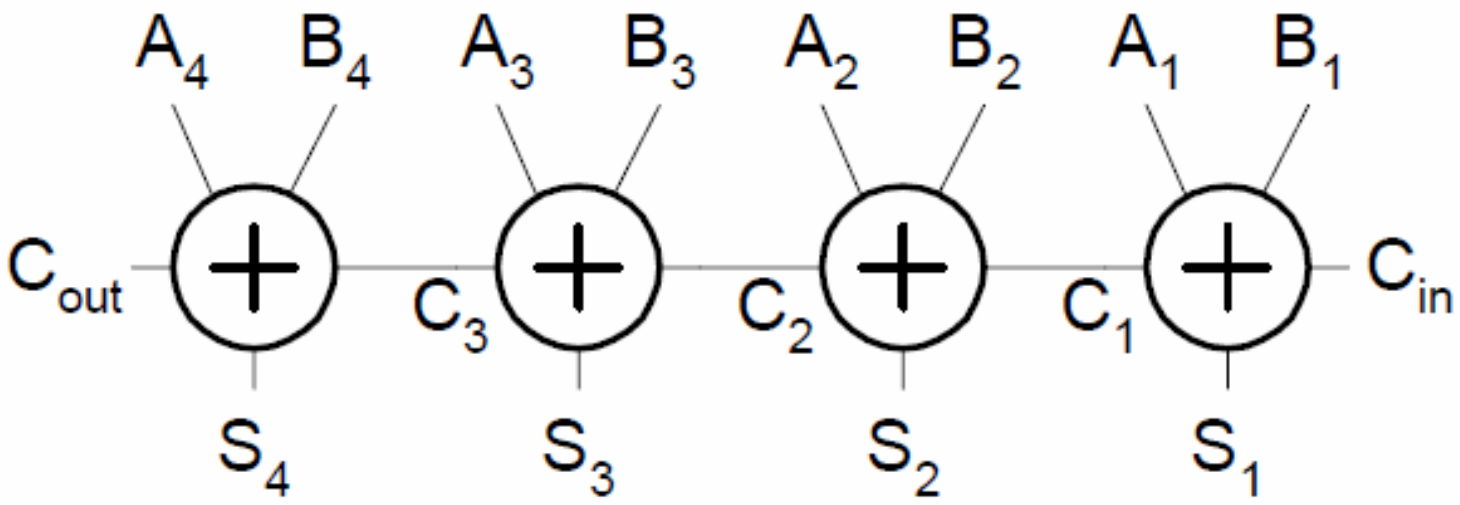
- N-bit adder called CPA
  - Each sum bit depends on all previous carries
  - How do we compute all these carries quickly?





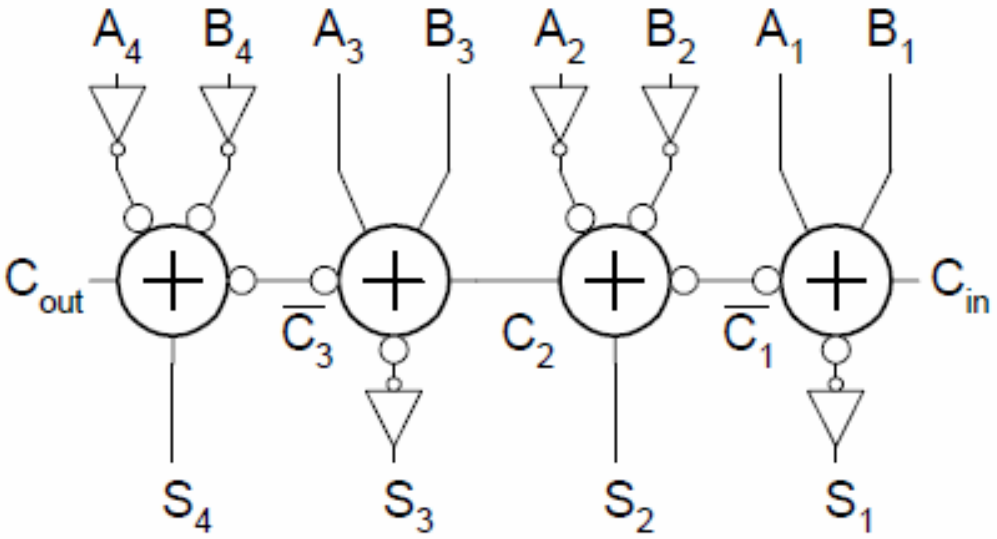
Ripple Carry Adder

- Simplest design: cascade full adders
  - Critical path goes from C_{in} to C_{out}
  - Design full adder to have fast carry delay



Inversions

- Critical path passes through majority gate
  - Built from minority + inverter
  - Eliminate inverter and use inverting full adder



Propagate and Generate Logic

- Equations often factored into G and P
- Generate and propagate for groups spanning i:j

$$G_{i:j} =$$

$$P_{i:j} =$$

- Base case

$$G_{i:i} \equiv G_i =$$

$$P_{i:i} \equiv P_i =$$

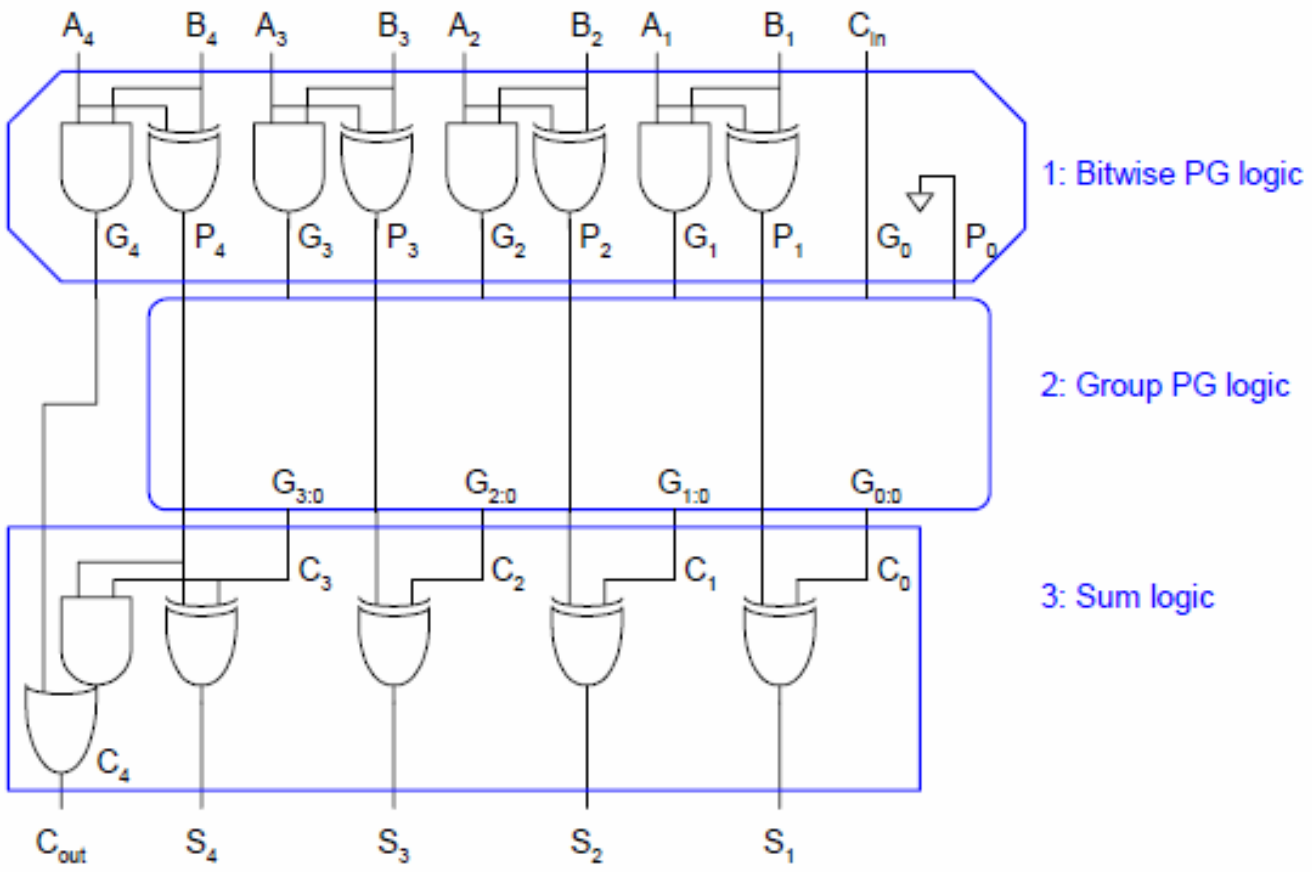
$$G_{0:0} \equiv G_0 =$$

$$P_{0:0} \equiv P_0 =$$

- Sum:

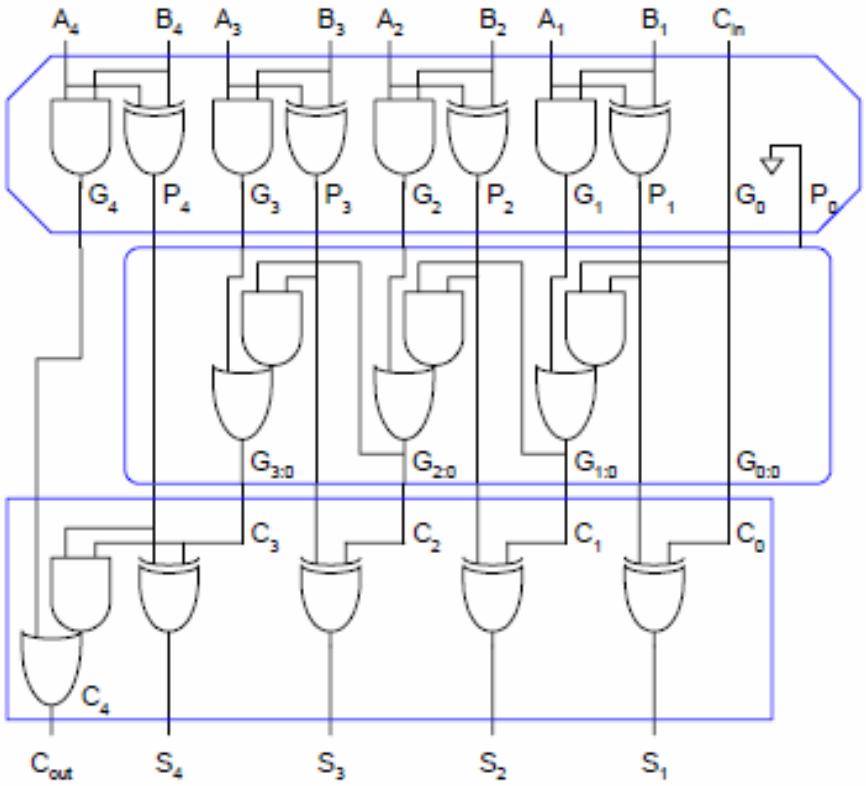
$$S_i =$$

PG Logic


A4
B4
A3
B3
A2
B2
A1
B1
Cin
G4
P4
G3
P3
G2
P2
G1
P1
G0
P0
1: Bitwise PG logic
G3:0
G2:0
G1:0
G0:0
2: Group PG logic
C3
C2
C1
C0
C4
3: Sum logic
Cout
S4
S3
S2
S1

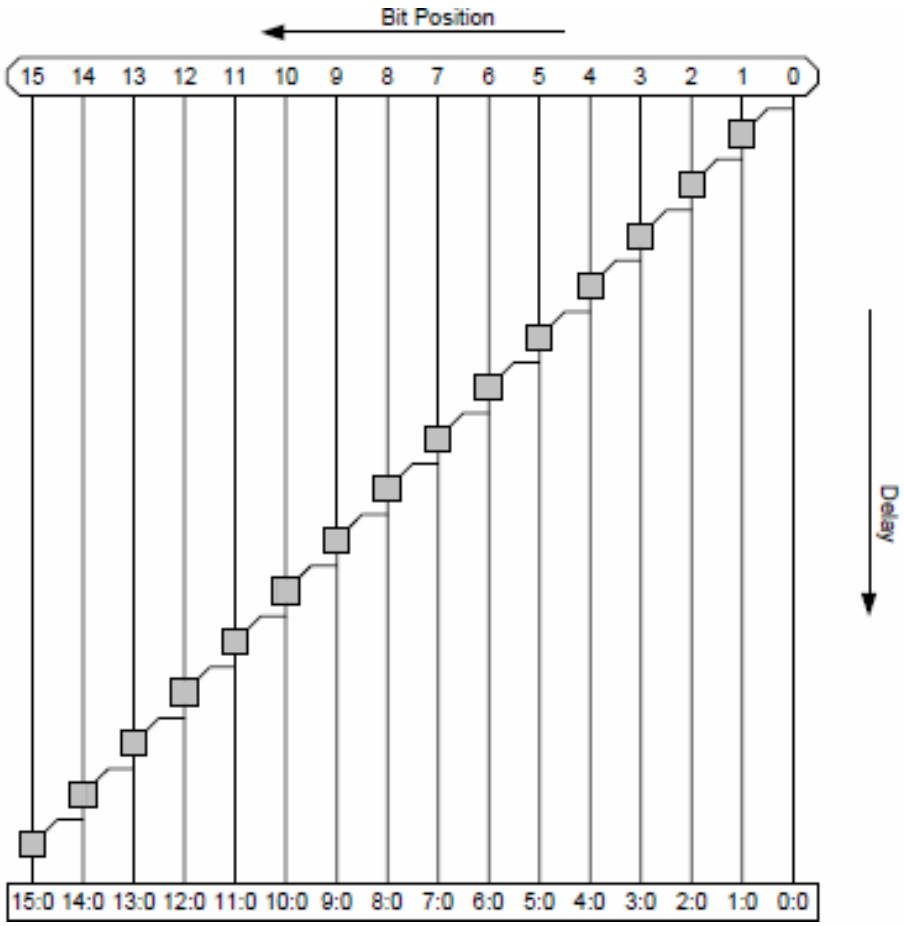
Carry Ripple Revisited

$$G_{i:0} = G_i + P_i \bullet G_{i-1:0}$$

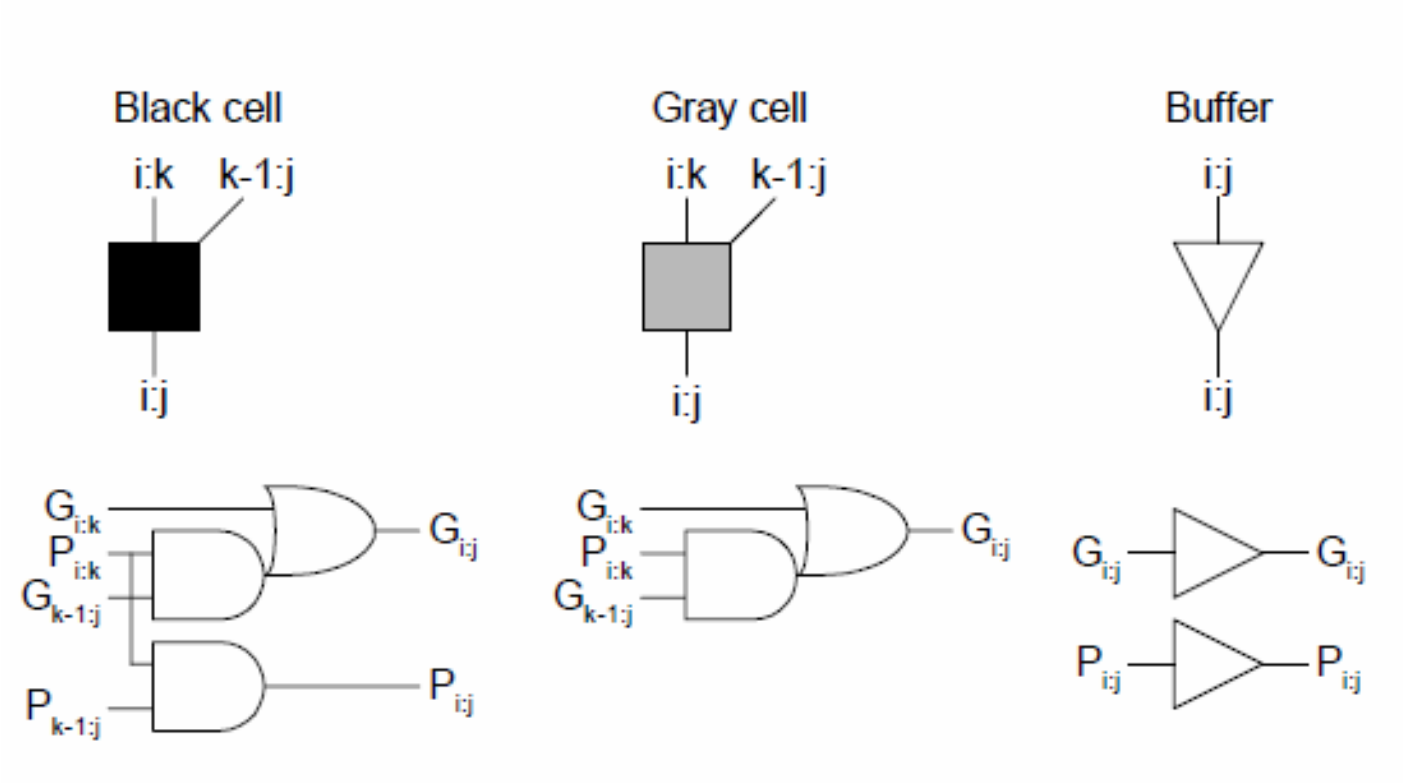


Carry Ripple PG Diagram

$t_{ripple} =$

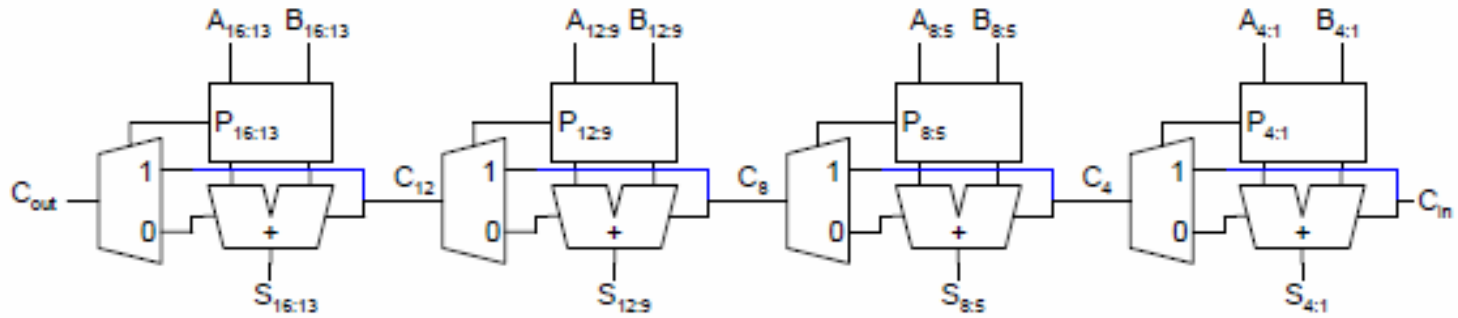


Notion of Gray Cell, Black Cell and Buffer

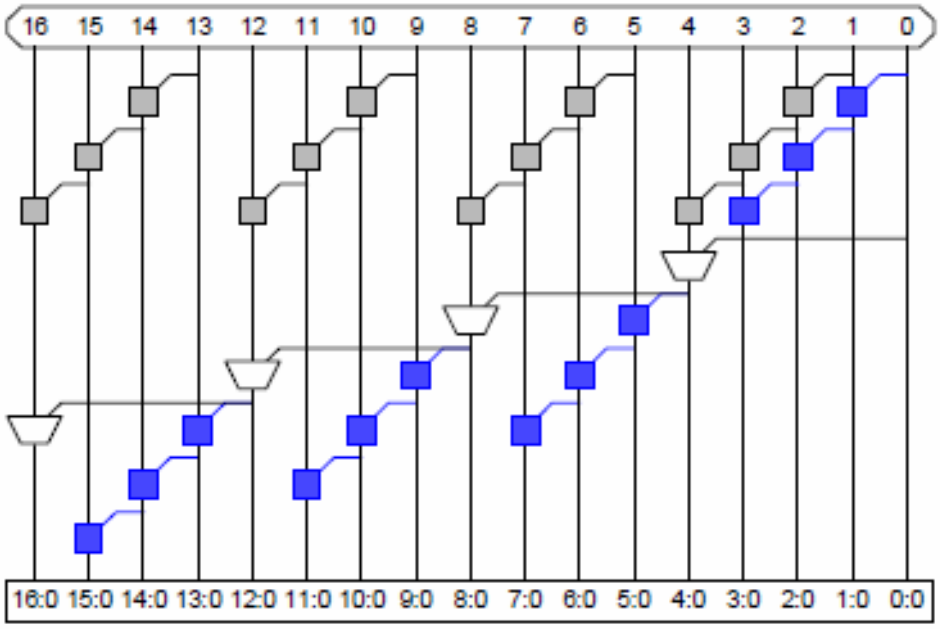


Carry Skip Adder

- 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



Carry Skip PG Diagram

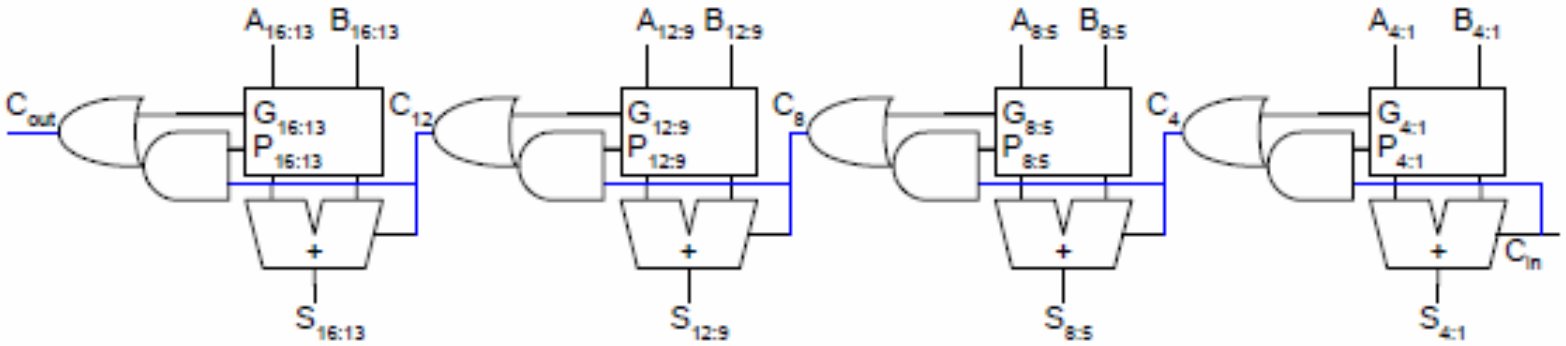


For k n-bit groups (N = nk)

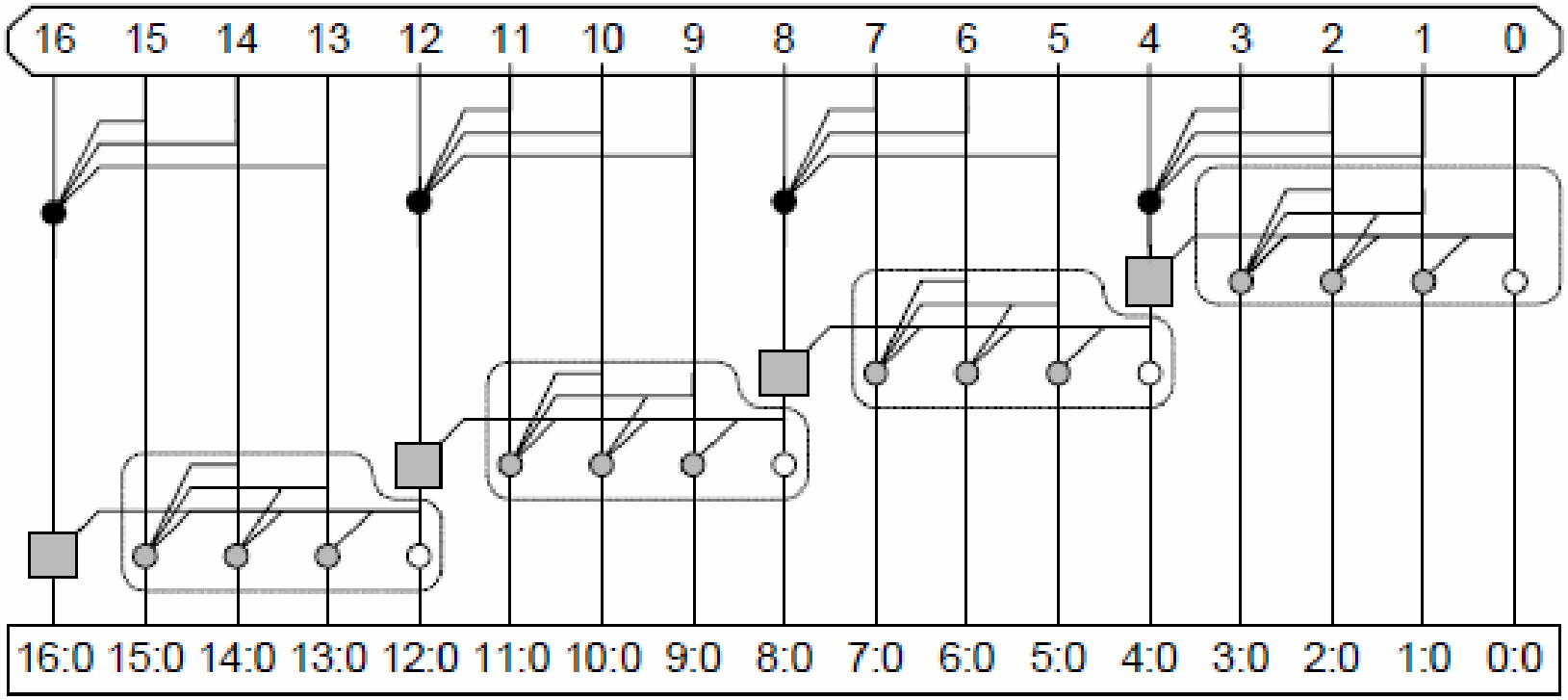
t_{skip} =

Carry Look Ahead Adder

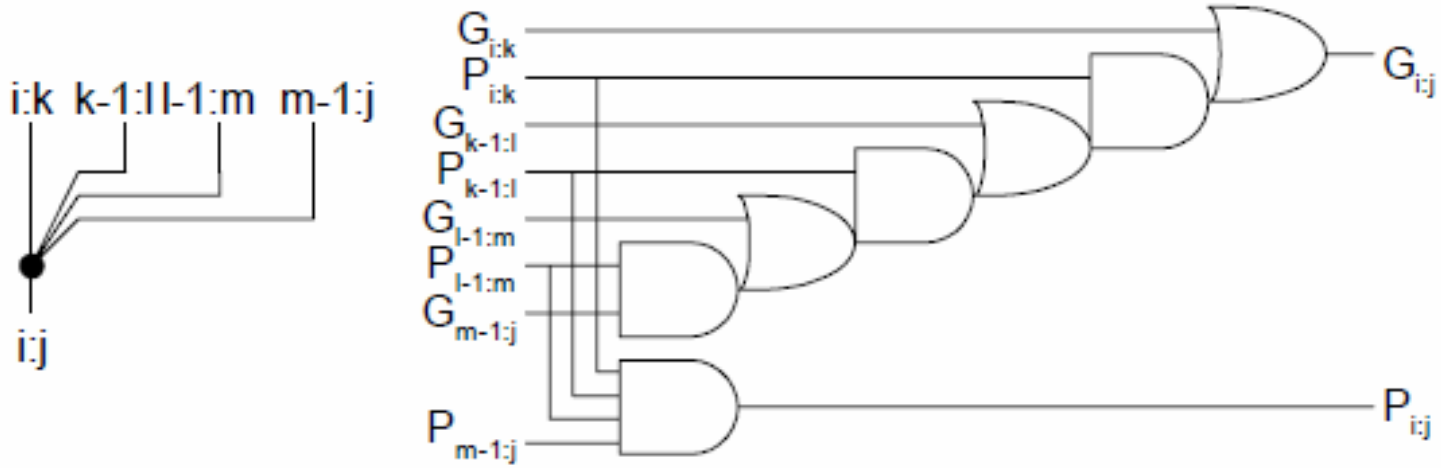
- Carry-lookahead adder computes $G_{i:0}$ for many bits in parallel.
- Uses higher-valency cells with more than two inputs.



CLA PG Diagram

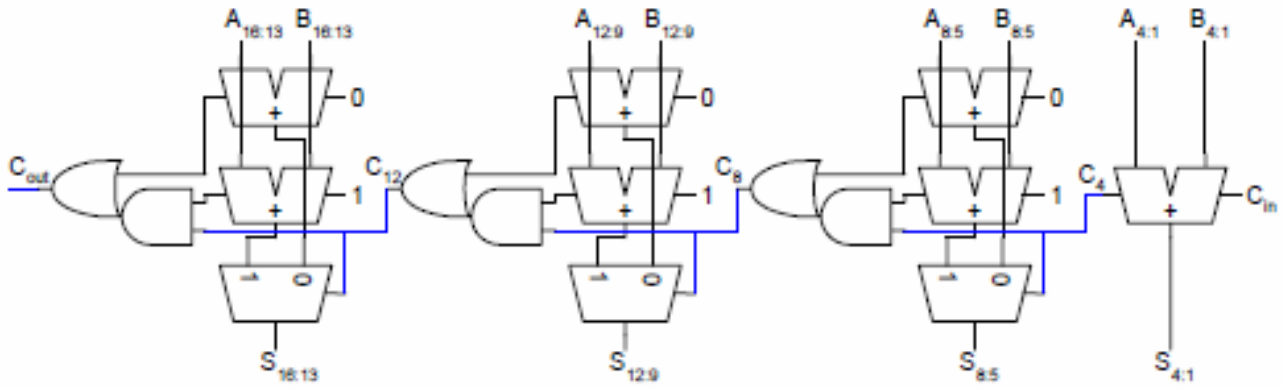


Higher Valency Cells



Carry Select Adder

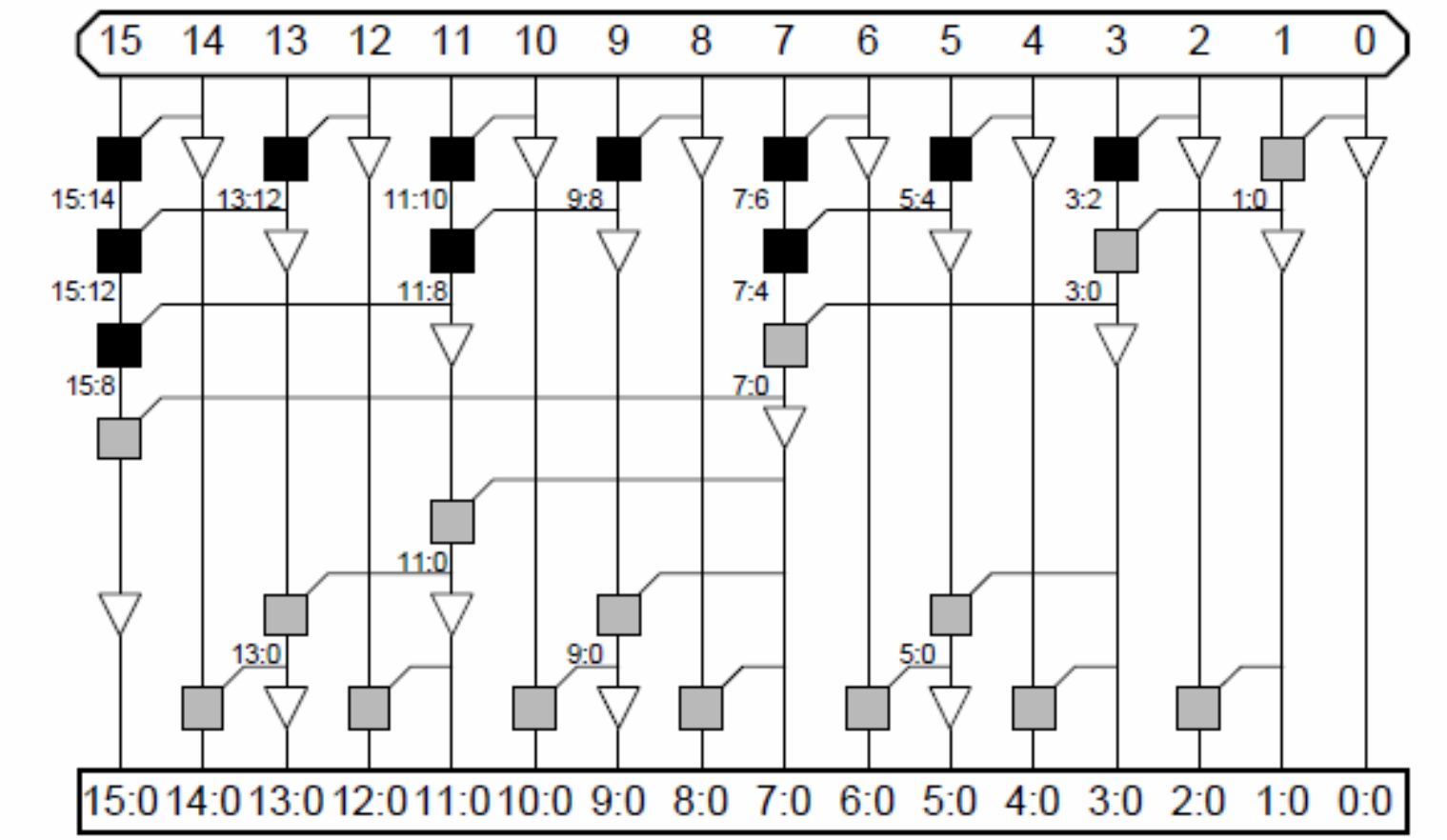
- Trick for critical paths dependent on late input X
  - Precompute two possible outputs for X = 0, 1
  - Select proper output when X arrives
- Carry-select adder precomputes n-bit sums
  - For both possible carries into n-bit group



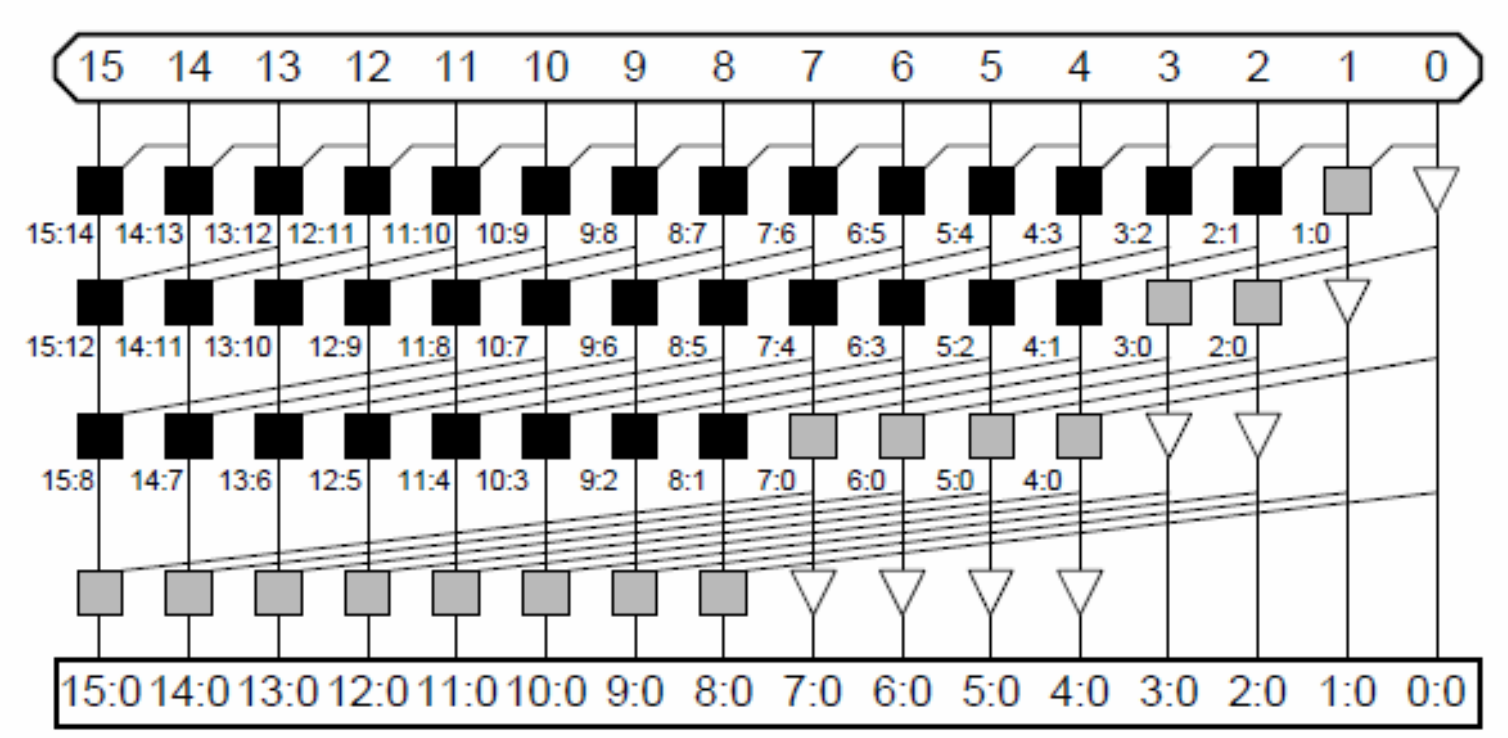
Tree Adders

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

Brent Kung Adder



Kogge Stone Adder

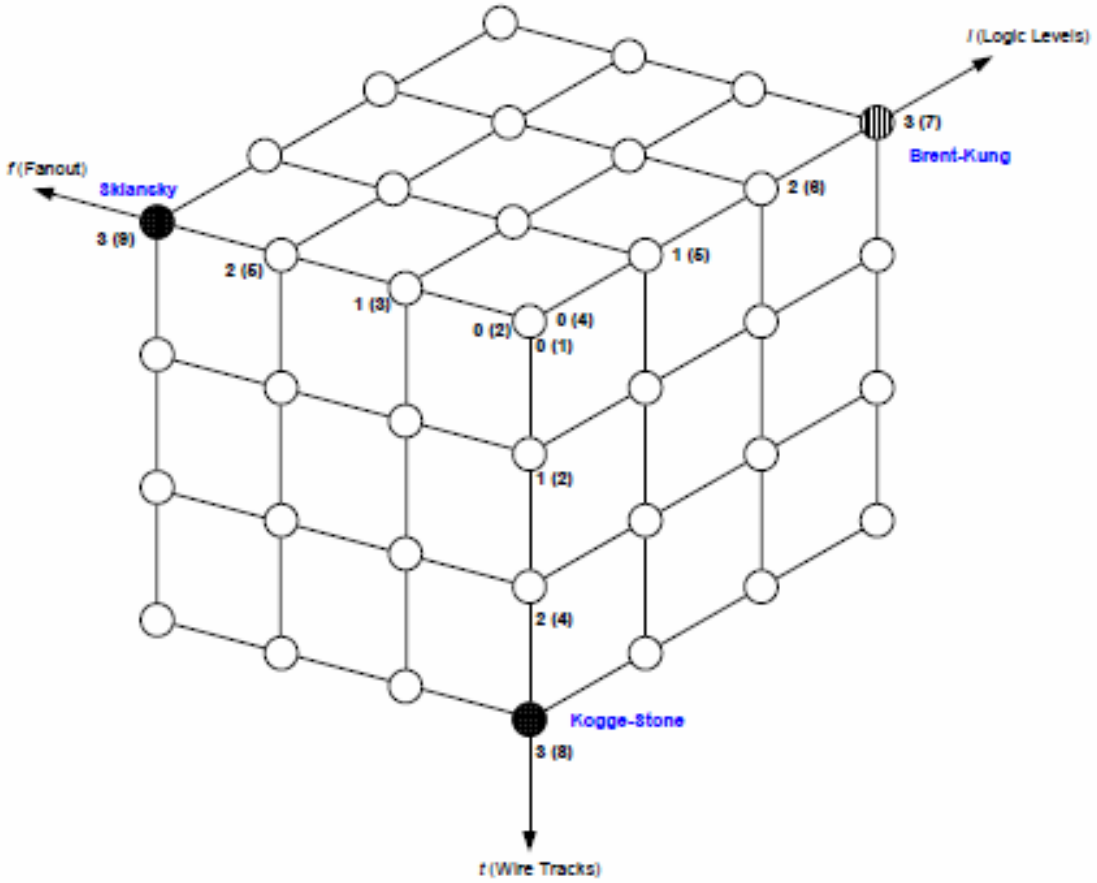


Tree Adder Taxonomy

- Ideal N-bit tree adder would have
  - $L = \log N$ logic levels
  - Fanout never exceeding 2
  - No more than one wiring track between levels
- 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

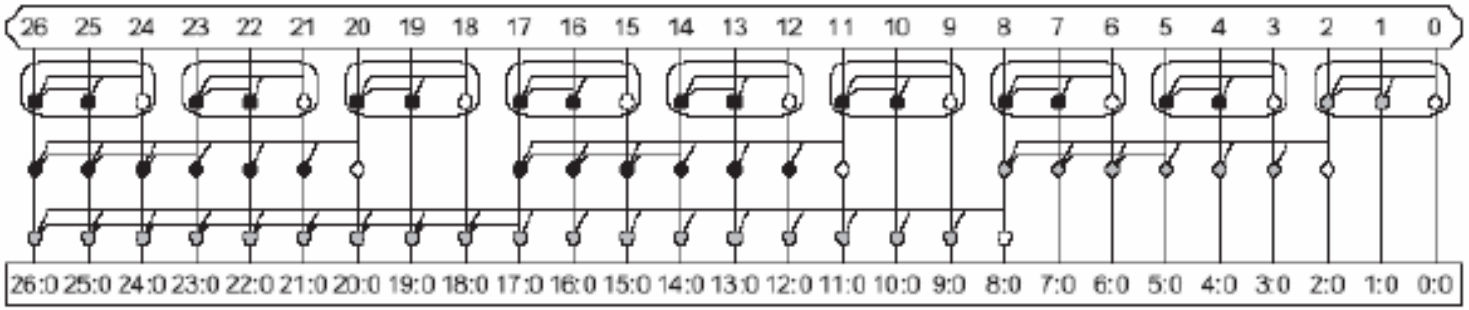
$$l + f + t = L\text{-}1$$

Tree Adder Taxonomy



Higher Valency Trees

- Combine 3 or 4 groups at each level
- High fan-in gates better suited to domino circuits



Thank You