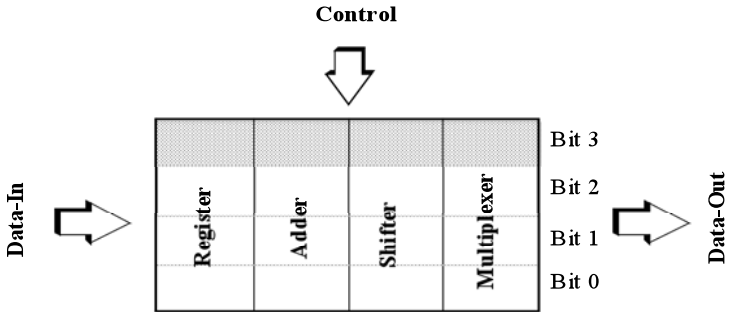
Topic 9
Arithmetic Circuits & Datapaths

Peter Cheung
Department of Electrical & Electronic Engineering
Imperial College London

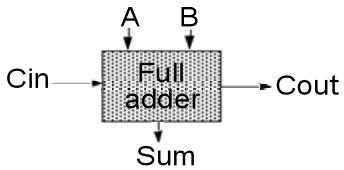
URL: www.ee.ic.ac.uk/pcheung/
E-mail: p.cheung@ic.ac.uk

Bit-Sliced Design



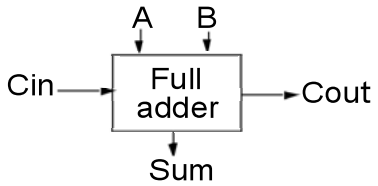
Tile identical processing elements

Full-Adder



A	*B*	C_i	*S*	C_o	*Carry status*
0	0	0	0	0	delete
0	0	1	1	0	delete
0	1	0	1	0	propagate
0	1	1	0	1	propagate
1	0	0	1	0	propagate
1	0	1	0	1	propagate
1	1	0	0	1	generate
1	1	1	1	1	generate

The Binary Adder



$$S = A \oplus B \oplus C_i$$
$$= A\bar{B}\bar{C}_i + \bar{A}B\bar{C}_i + \bar{A}\bar{B}C_i + ABC_i$$
$$C_o = AB + BC_i + AC_i$$

Express Sum and Carry as a function of P, G, D

Define 3 new variable which ONLY depend on A, B

Generate (G) = AB

Propagate (P) = A $\oplus$ B

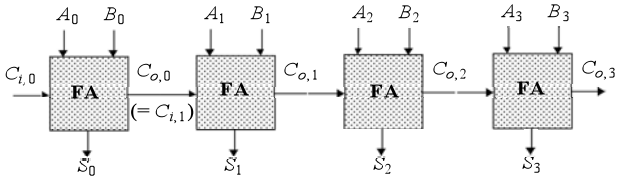
Delete = $\overline{A}\,\overline{B}$

$$C_o(G,P) = G + PC_i$$

$$S(G,P) = P \oplus C_i$$

Can also derive expressions for S and C_o based on D and P

The Ripple-Carry Adder



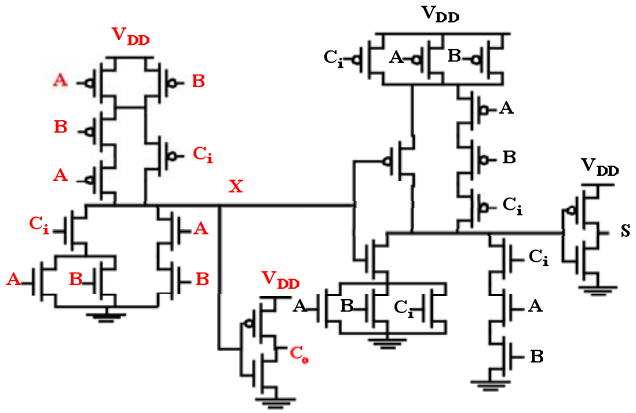
Worst case delay linear with the number of bits

$t_d = O(N)$

$$t_{adder} \approx (N-1)t_{carry} + t_{sum}$$

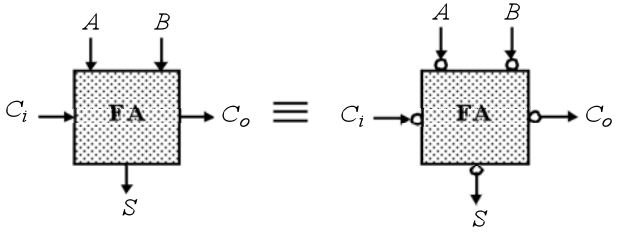
Goal: Make the fastest possible carry path circuit

Complimentary Static CMOS Full Adder



28 Transistors

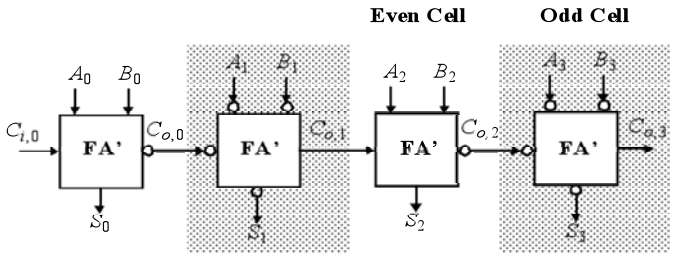
Inversion Property



$$\overline{S}(A,B,C_i) = S(\overline{A},\overline{B},\overline{C_i})$$

$$\overline{C_o}(A,B,C_i) = C_o(\overline{A},\overline{B},\overline{C_i})$$

Minimize Critical Path by Reducing Inverting Stages



Exploit Inversion Property

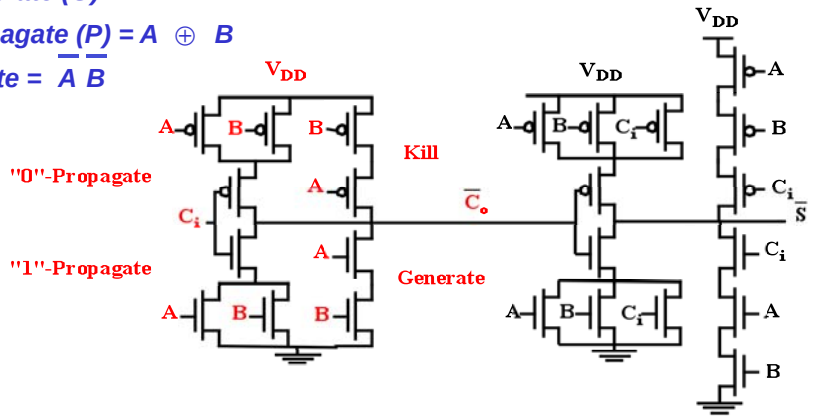
Note: need 2 different types of cells

The better structure: the Mirror Adder

Generate (G) = AB

Propagate (P) = A ⊕ B

Delete = $\overline{A}\,\overline{B}$

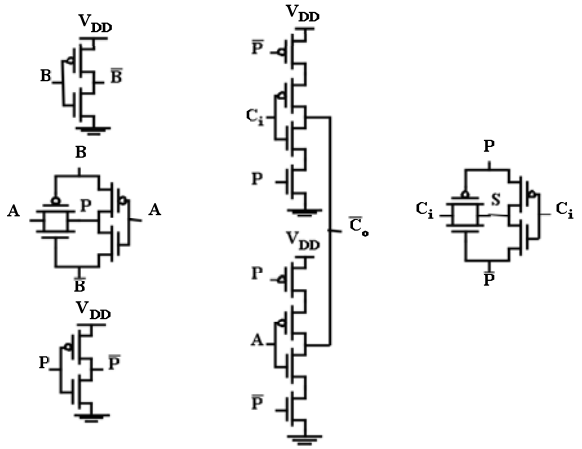


24 transistors

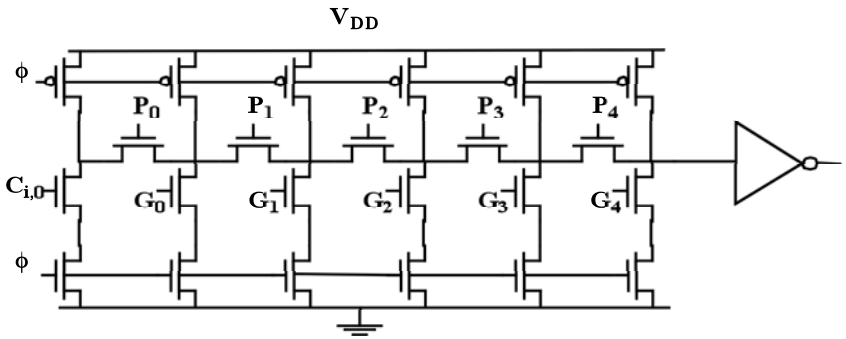
The Mirror Adder

- The NMOS and PMOS chains are completely symmetrical. This guarantees identical rising and falling transitions if the NMOS and PMOS devices are properly sized. A maximum of two series transistors can be observed in the carry-generation circuitry.
- When laying out the cell, the most critical issue is the minimization of the capacitance at node C_o. The reduction of the diffusion capacitances is particularly important.
- The capacitance at node C_o is composed of four diffusion capacitances, two internal gate capacitances, and six gate capacitances in the connecting adder cell .
- The transistors connected to C_i are placed closest to the output.
- Only the transistors in the carry stage have to be optimized for optimal speed. All transistors in the sum stage can be minimal size.

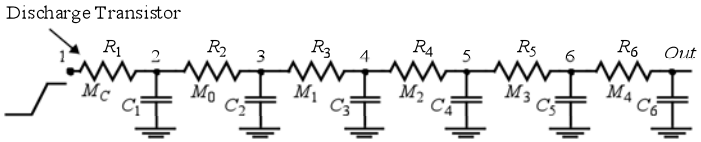
Quasi-Clocked Adder



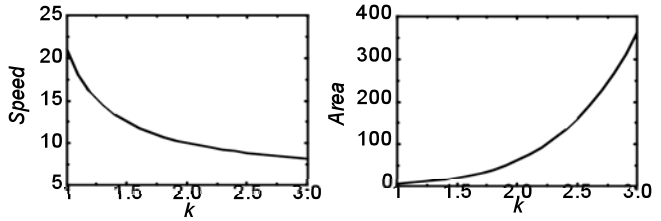
Manchester Carry Chain



Sizing Manchester Carry Chain

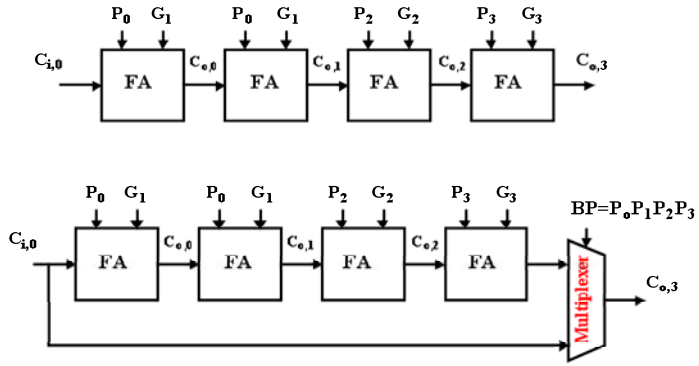


$$t_p = 0.69 \sum_{i=1}^{N} C_i \left(\sum_{j=1}^{i} R_j \right)$$



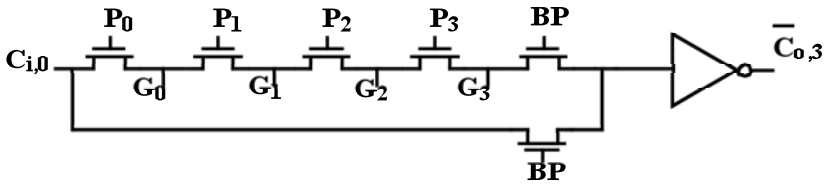
Speed (normalized by $0.69RC$) Area (in minimum size devices)

Carry-Bypass Adder

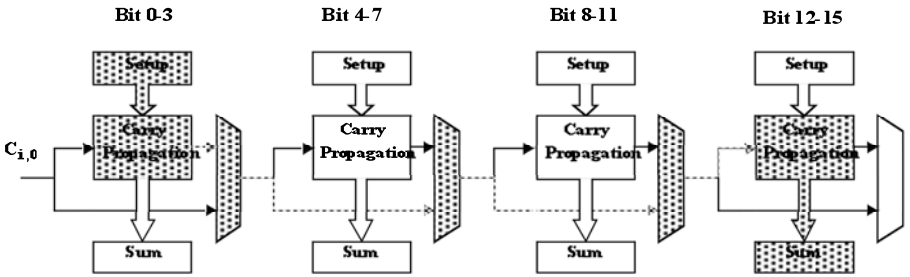


Idea: If (P0 and P1 and P2 and P3 = 1)
then $C_{o3} = C_0$, else "kill" or "generate".

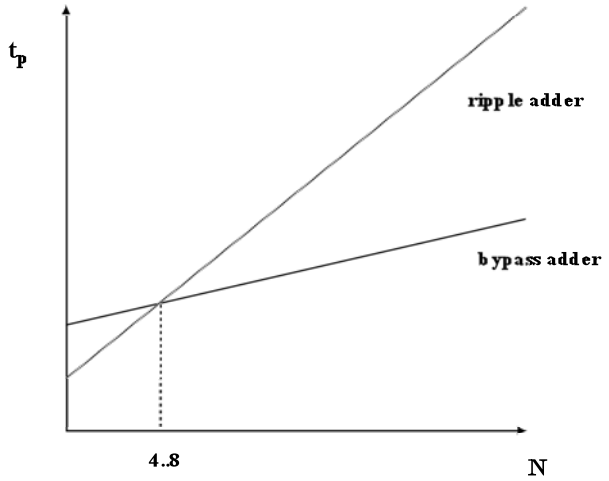
Manchester-Carry Implementation



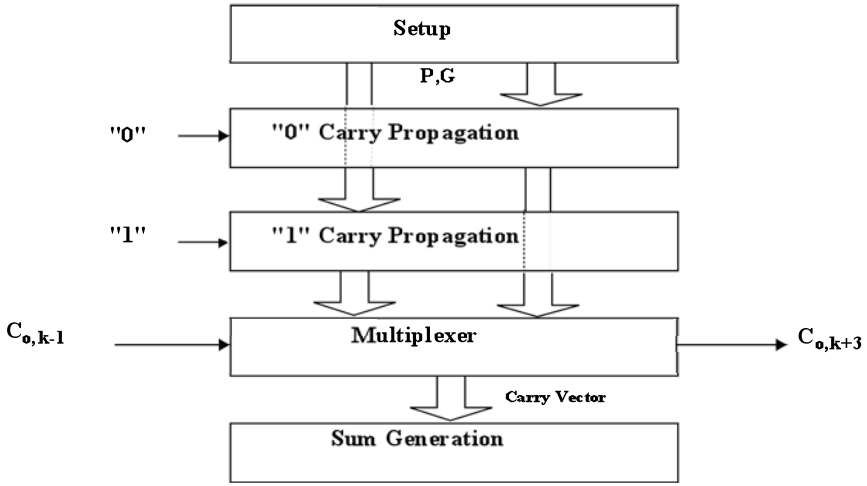
Carry-Bypass Adder (cont.)



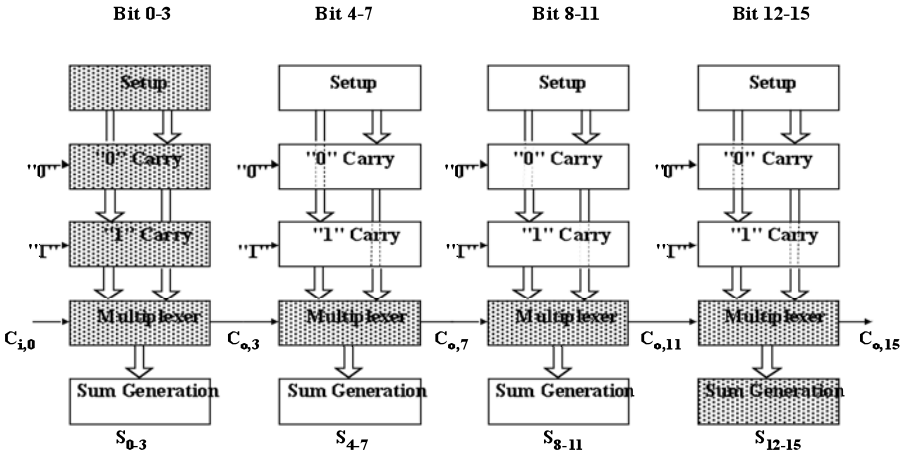
Carry Ripple versus Carry Bypass



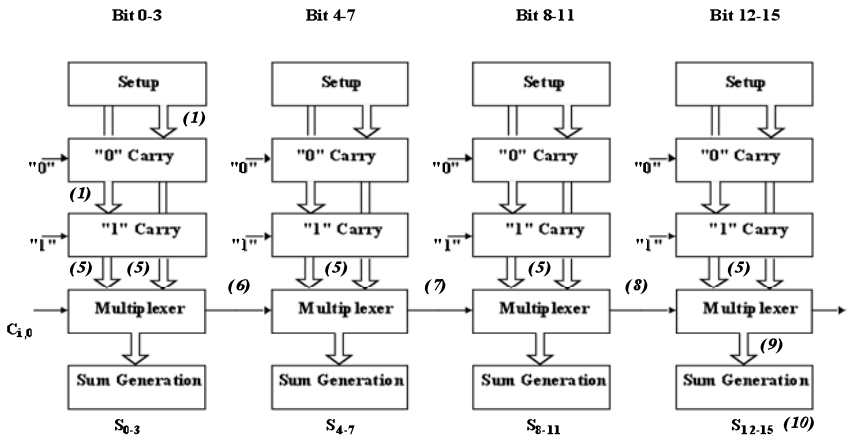
Carry-Select Adder



Carry Select Adder: Critical Path

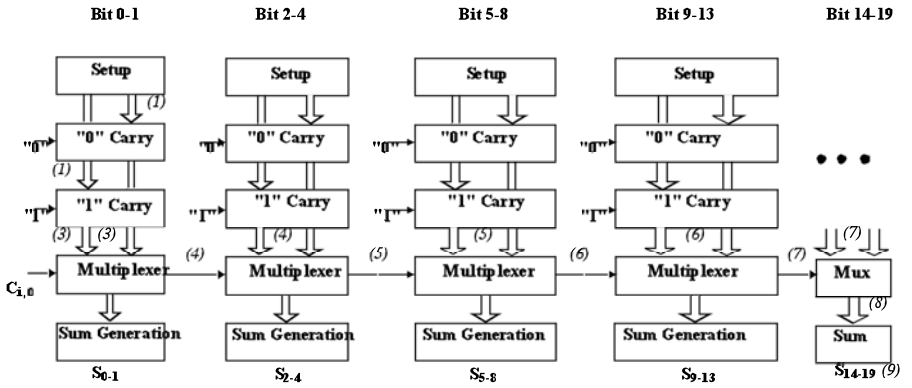


Linear Carry Select



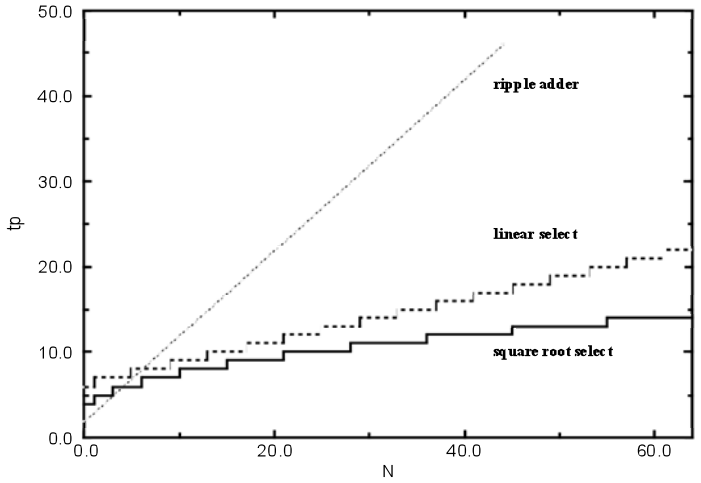
$$t_{add} = t_{setup} + \left(\frac{N}{M}\right) t_{carry} + M t_{mux} + t_{sum}$$

Square Root Carry Select

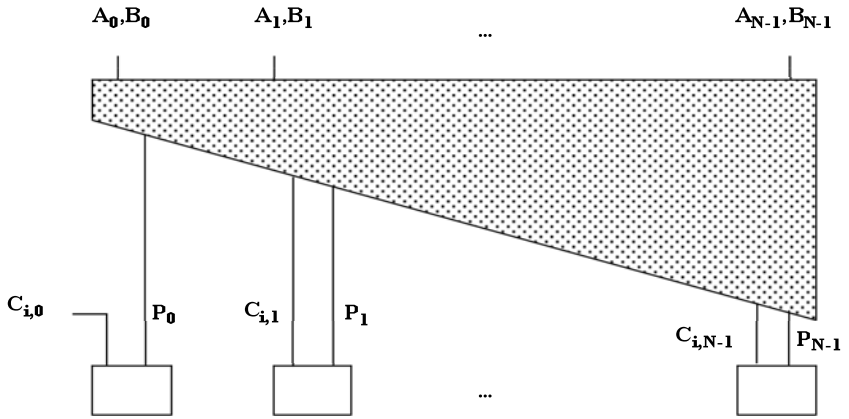


$$t_{add} = t_{setup} + P \cdot t_{carry} + (\sqrt{2N}) t_{mux} + t_{sum}$$

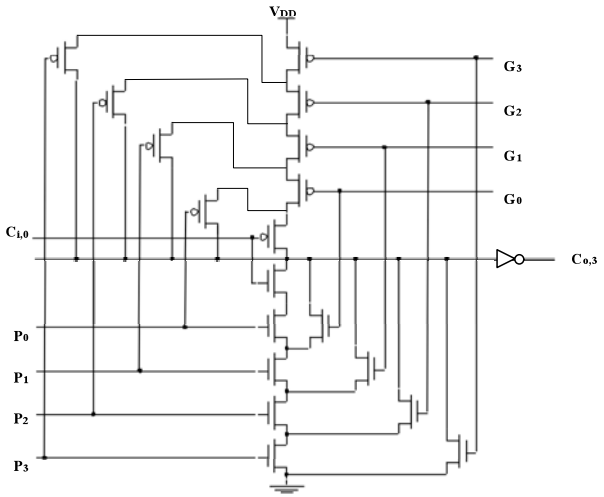
Adder Delays - Comparison



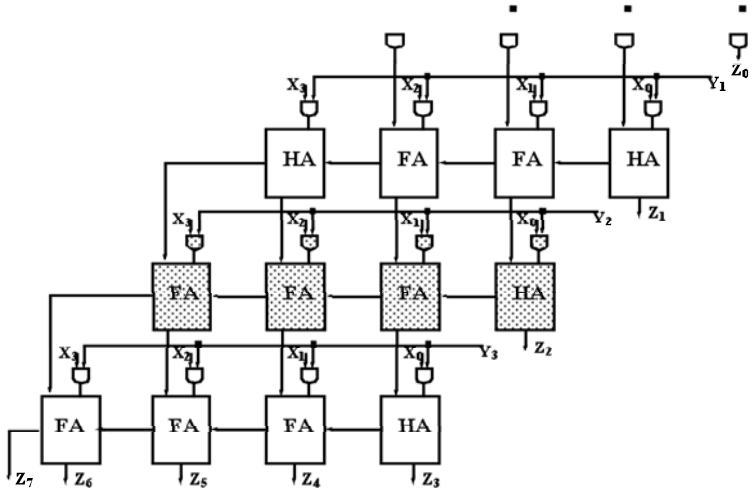
LookAhead - Basic Idea



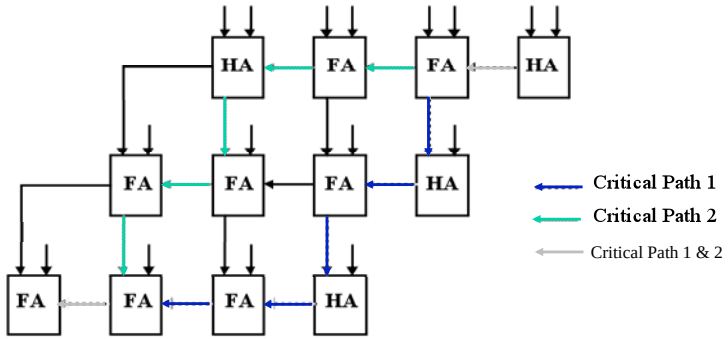
Look-Ahead: Topology



The Array Multiplier

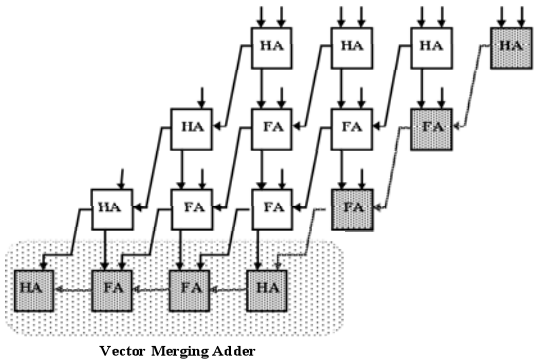


The MxN Array Multiplier — Critical Path



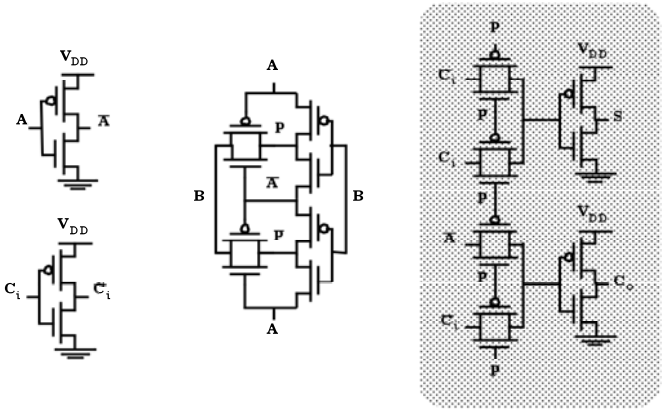
$$t_{mult} \approx [(M-1)+(N-2)]t_{carry}+(N-1)t_{sum}+(N-1)t_{and}$$

Carry-Save Multiplier



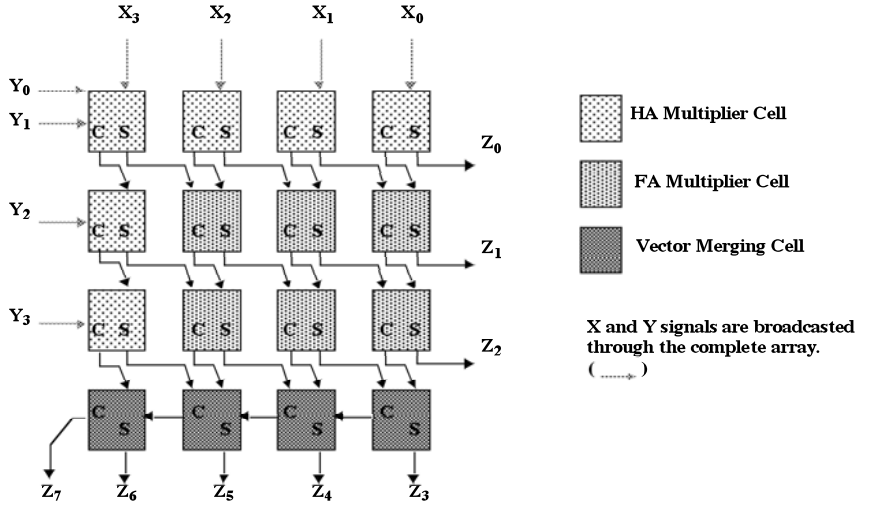
$$t_{mult} = (N-1)t_{carry}+(N-1)t_{and}+t_{merge}]$$

Adder Cells in Array Multiplier



Identical Delays for Carry and Sum

Multiplier Floorplan

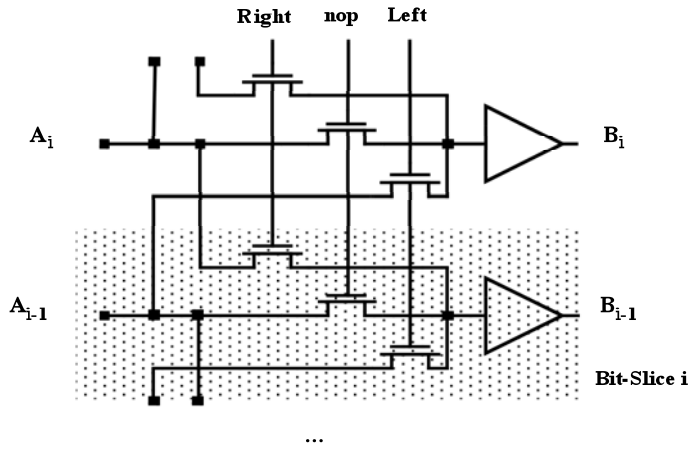


Multipliers —Summary

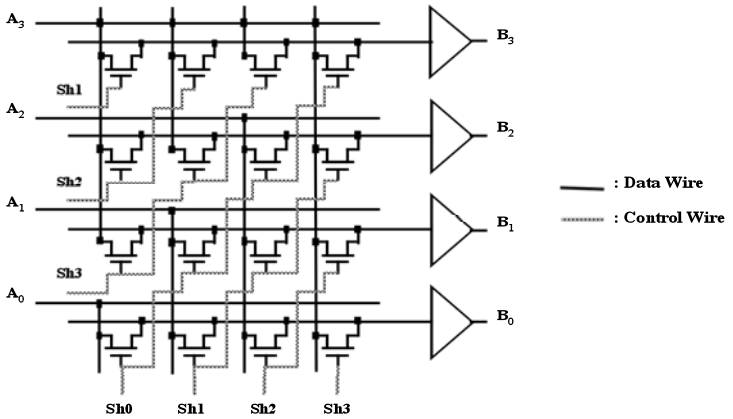
- Optimization Goals Different Vs Binary Adder
- Once Again: Identify Critical Path
- Other possible techniques
  - Logarithmic versus Linear (Wallace Tree Mult)
  - Data encoding (Booth)
  - Pipelining

FIRST GLIMPSE AT SYSTEM LEVEL OPTIMIZATION

The Binary Shifter

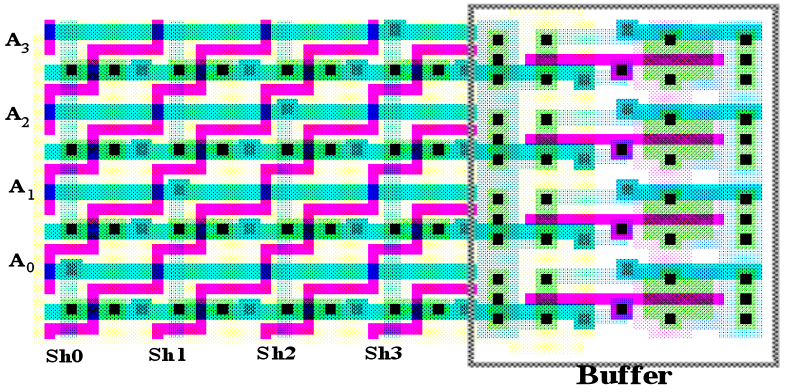


The Barrel Shifter



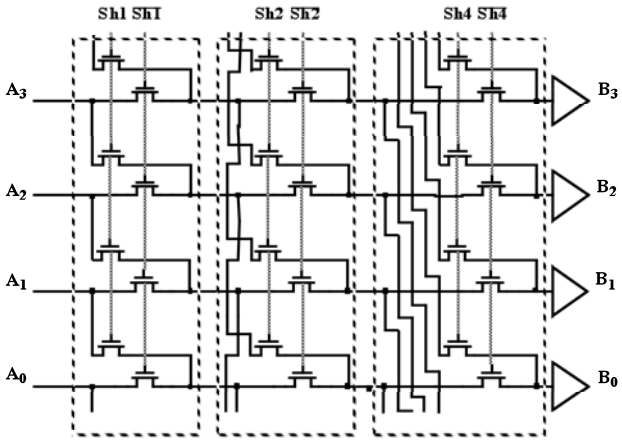
Area Dominated by Wiring

4x4 barrel shifter

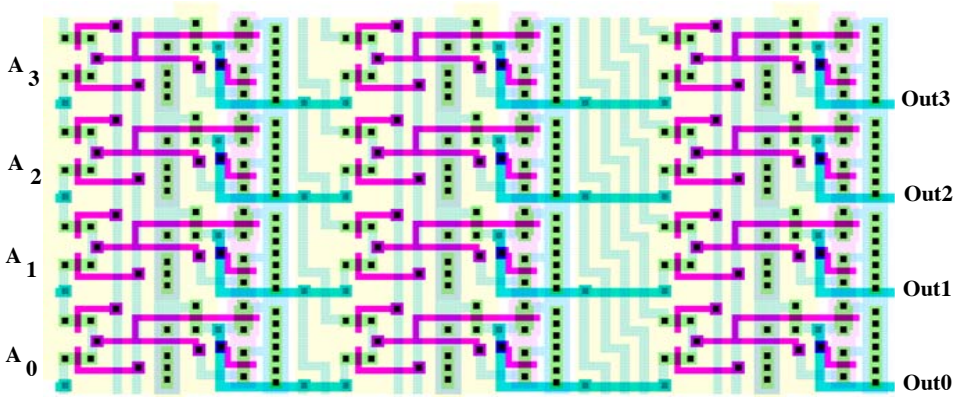


$$Width_{barrel} \sim 2\, p_m\, M$$

Logarithmic Shifter

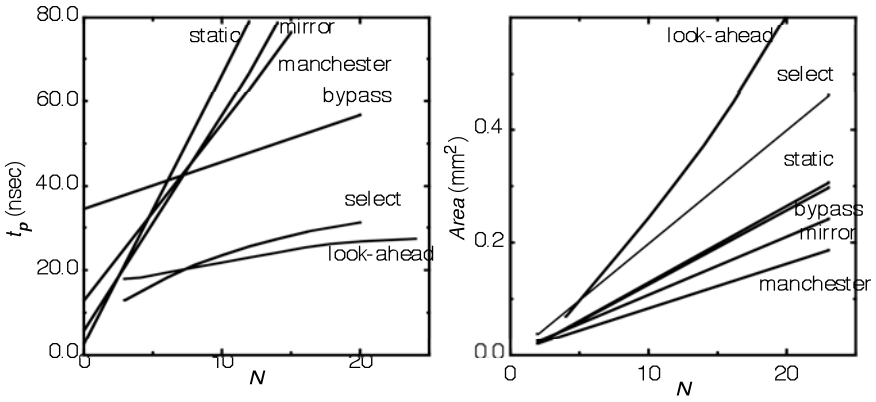


0-7 bit Logarithmic Shifter

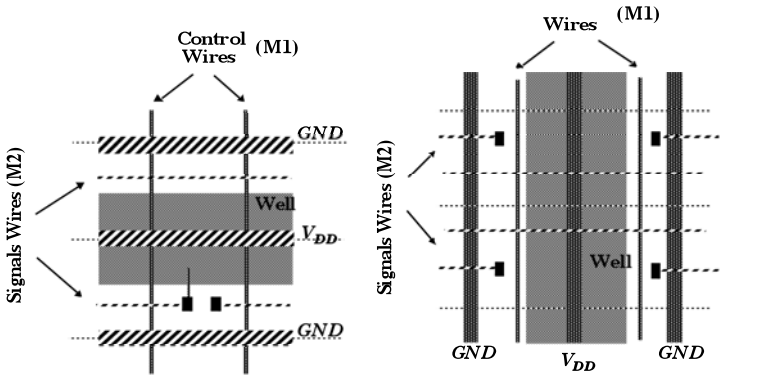


$$width_{\log} \approx p_m\left(2K + \left(1 + 2 + \ldots + 2^{K-1}\right)\right) = p_m\left(2^K + 2K - 1\right)$$

Design as a Trade-Off



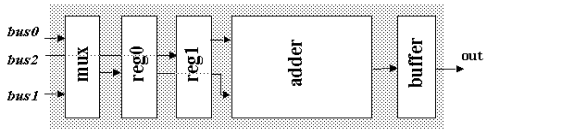
Layout Strategies for Bit-Sliced Datapaths

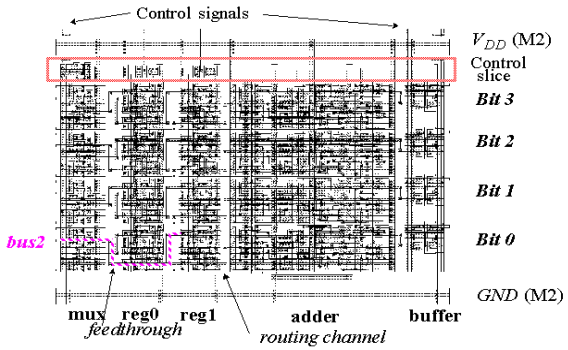


Approach I — Signal and power lines parallel

Approach II — Signal and power lines perpendicular

Layout of Bit-sliced Datapaths





Layout of Bit-sliced Datapaths

(a) Datapath without feedthroughs and without pitch matching (area = 4.2 mm²).

(b) Adding feedthroughs (area = 3.2 mm²)

(c) Equalizing the cell height reduces the area to 2.2 mm².