
EECS 427

Lecture 7: Dynamic Logic and Introduction to Adders

Readings: 6.2, 11.1-11.3.3

Reminders

- CAD3 is due today (tomorrow morning if you like)
- CAD4 is due Wednesday 10/14, 2 weeks away
 - “Soft” deadline: you can submit by Thursday 10/15 at noon
- Looking ahead:
 - HW3 – Project initial proposal: due in a week
 - Quiz1 Wednesday 10/14, 2 weeks away

Last Time

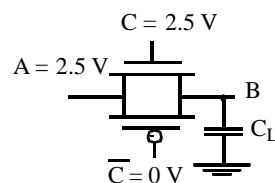
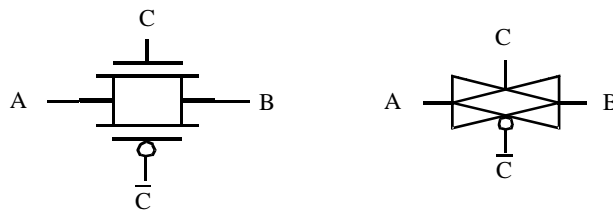
- Skewed gates
 - Up-skewed gates: faster pull-up (lower g), slower pull-down (higher g)
 - Down-skewed gates: faster pull-down, slower pull-up
- Ratioed logic – improved loads
 - Pseudo-NMOS – static current, low noise margin
 - DCVSL – no static current but cross-over current
- Pass-transistor circuits
 - PTL – threshold drop, causing static current in following gate
 - CPL – one side pulls up and the other pulls down, fast switching

EECS 427 F09

Lecture 7

3

Transmission Gate

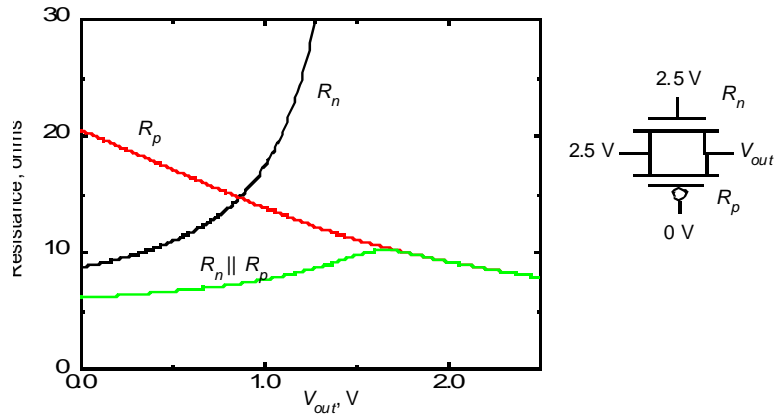


EECS 427 F09

Lecture 7

4

Equivalent Resistance

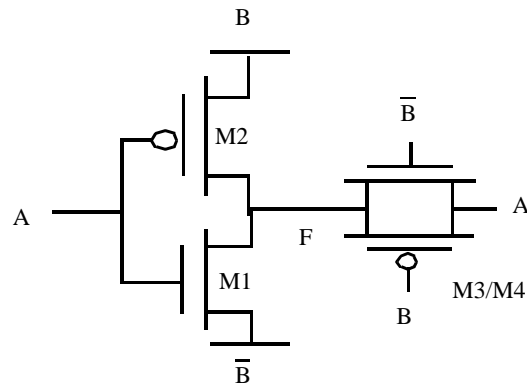


EECS 427 F09

Lecture 7

5

Transmission Gate XOR

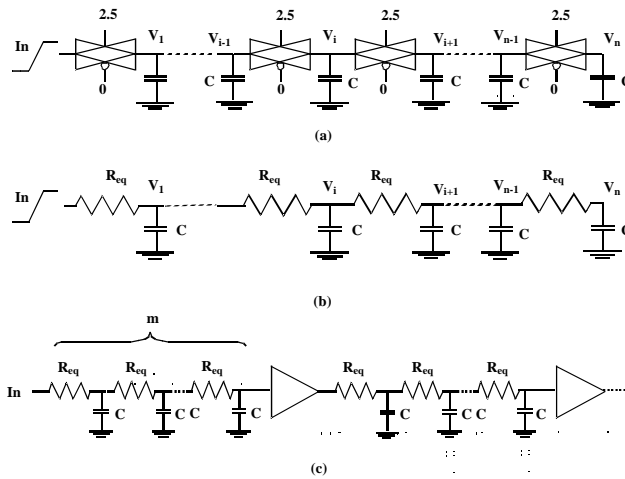


EECS 427 F09

Lecture 7

6

Transmission Gate Network



EECS 427 F09

Lecture 7

7

Dynamic CMOS

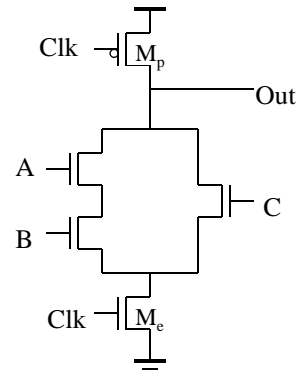
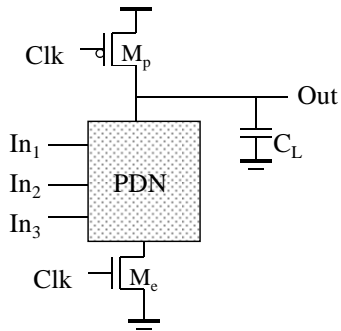
- In **static** circuits at every point in time (except when switching) the output is connected to either GND or V_{DD} via a low resistance path.
 - fan-in of n requires $2n$ (n N-type + n P-type) devices in complementary static CMOS
- **Dynamic** circuits rely on the temporary storage of signal values on the capacitance of high impedance nodes.
 - requires on $n + 2$ ($n+1$ N-type + 1 P-type) transistors

EECS 427 F09

Lecture 7

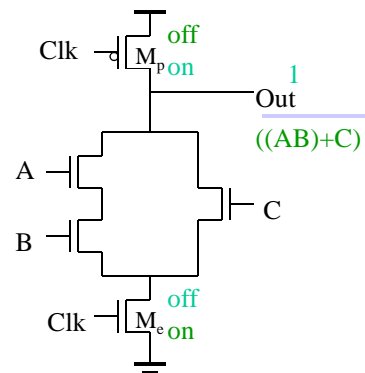
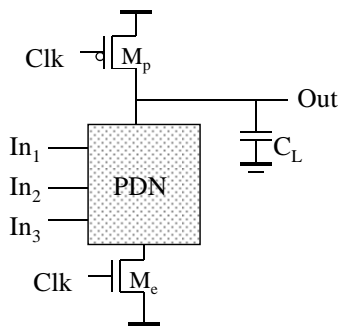
8

Dynamic Gate



Two phase operation
 Precharge (CLK = 0)
 Evaluate (CLK = 1)

Dynamic Gate



Two phase operation
 Precharge (Clk = 0)
 Evaluate (Clk = 1)

Conditions on Output

- Once the output of a dynamic gate is discharged, it cannot be charged again until the next precharge operation.
- Inputs to the gate can make **at most** one transition during evaluation.
- Output can be in the high impedance state during and after evaluation (PDN off), state is stored on C_L

Properties of Dynamic Gates

- Logic function is implemented by the PDN only
 - number of transistors is $N + 2$ (versus $2N$ for static complementary CMOS)
- Full swing outputs ($V_{OL} = \text{GND}$ and $V_{OH} = V_{DD}$)
- Non-ratioed - sizing of the devices does not affect the logic levels
- Faster switching speeds
 - reduced load capacitance due to **lower input** capacitance (C_{in})
 - reduced load capacitance due to smaller output loading (C_{out})
 - no I_{sc} , so all the current provided by PDN goes into discharging C_L

Properties of Dynamic Gates

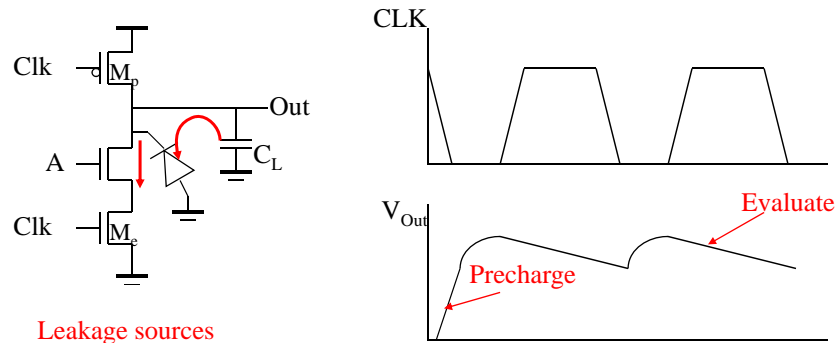
- Overall power dissipation usually **higher** than static CMOS
 - no static current path ever exists between V_{DD} and GND (including P_{sc})
 - no glitching
 - **higher transition probabilities**
 - **extra load on Clk**
- PDN starts to work as soon as the input signals exceed V_{Tn} , so V_M , V_{IH} and V_{IL} equal to V_{Tn}
 - low noise margin (NM_L)
- Needs a precharge/evaluate clock

EECS 427 F09

Lecture 7

13

Charge Leakage



Leakage sources

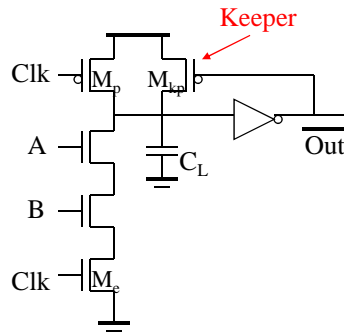
Dominant component is subthreshold current

EECS 427 F09

Lecture 7

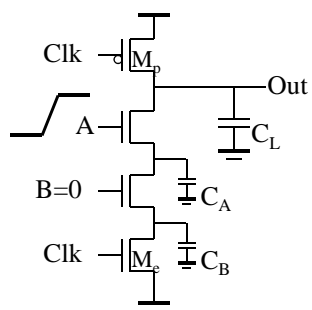
14

Keeper



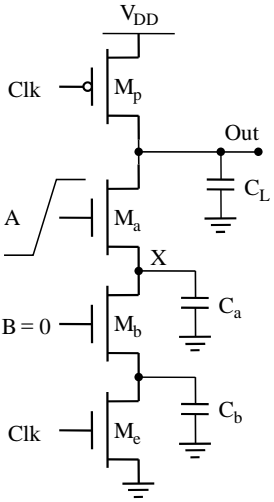
Same approach as level restorer for pass-transistor logic

Charge Sharing



Charge stored originally on C_L is redistributed (shared) over C_L and C_A leading to reduced robustness

Charge Sharing



case 1) if $\Delta V_{out} < V_{Tn}$

$$C_L V_{DD} = C_L V_{out}(t) + C_a (V_{DD} - V_{Tn}(V_X))$$

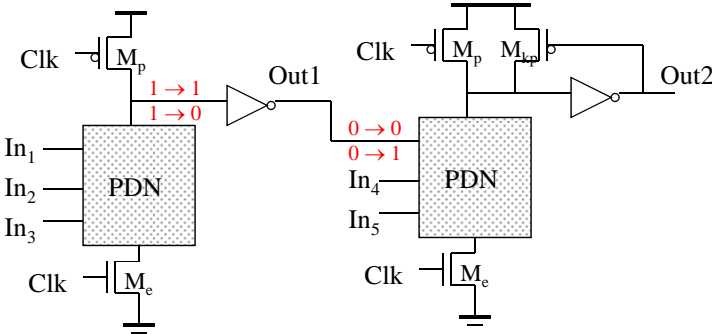
or

$$\Delta V_{out} = V_{out}(t) - V_{DD} = -\frac{C_a}{C_L} (V_{DD} - V_{Tn}(V_X))$$

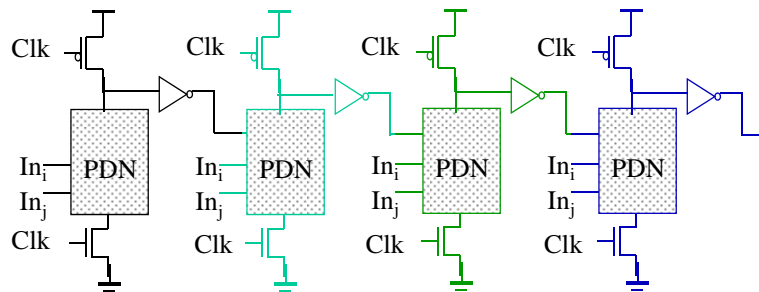
case 2) if $\Delta V_{out} > V_{Tn}$

$$\Delta V_{out} = -V_{DD} \left(\frac{C_a}{C_a + C_L} \right)$$

Domino Logic



Cascading Dominos

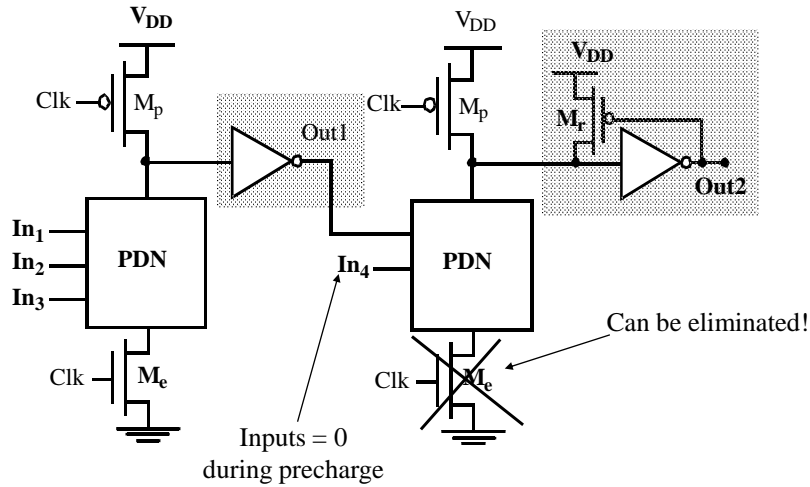


Like falling dominos!

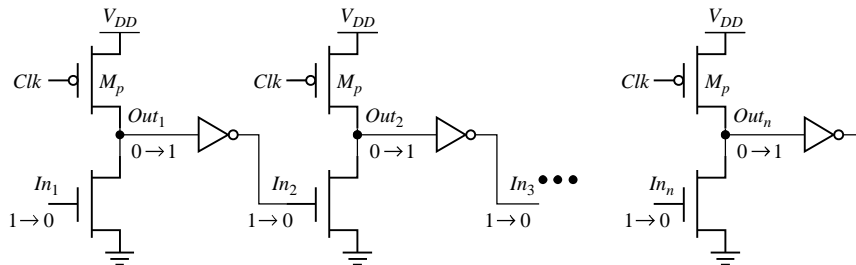
Properties of Domino Logic

- Only non-inverting logic can be implemented
- Very high speed
 - static inverter can be skewed, only L-H transition
 - Input capacitance reduced – smaller logical effort

Design with Domino Logic

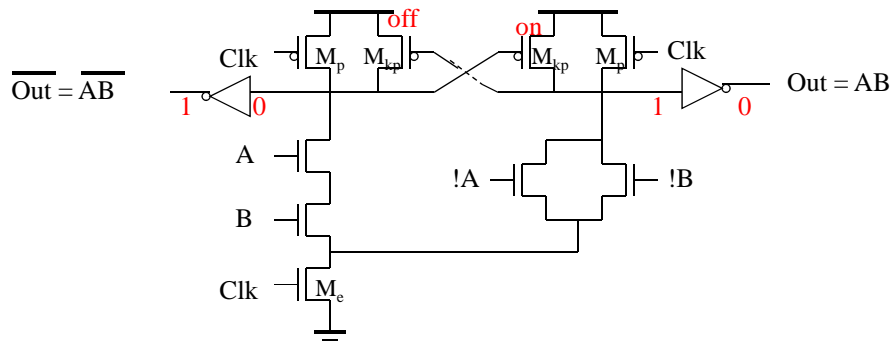


Footless Domino



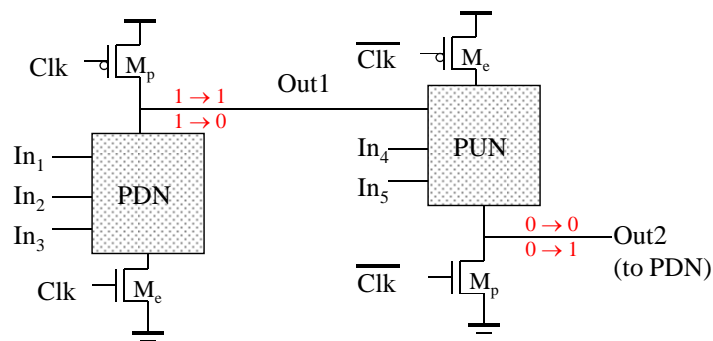
The first gate in the chain needs a foot switch
 Precharge is rippling – short-circuit current
 A solution is to delay the clock for each stage

Dual-Rail Domino



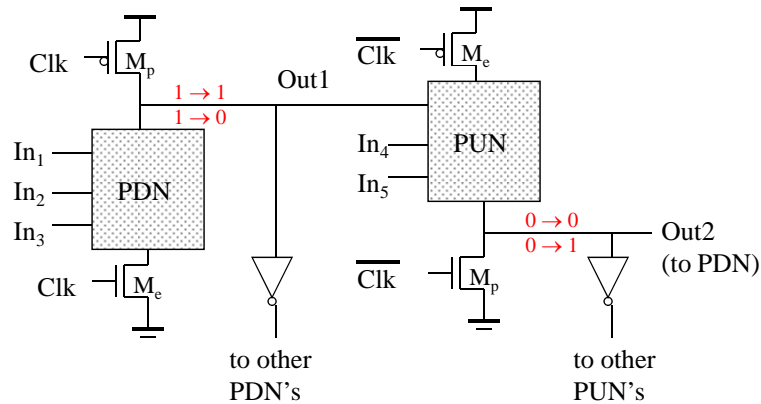
Solves the problem of non-inverting logic

np-CMOS



Only 0 → 1 transitions allowed at inputs of PDN
 Only 1 → 0 transitions allowed at inputs of PUN

NORA Logic

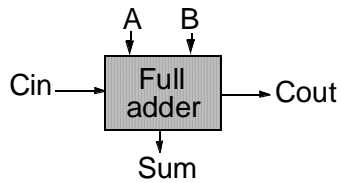


WARNING: Very sensitive to noise!

Short Summary

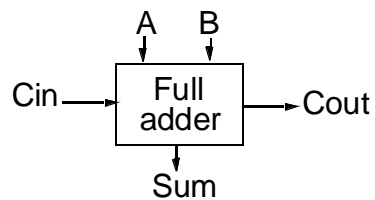
- Dynamic circuits – high performance
 - Dynamic logic – non-ratioed, dynamic power only, no static current, higher activity, low noise margin
 - Domino logic – can be safely cascaded, only non-inverting logic
 - Footless domino – ripple precharge, delayed clock, extra power
 - Dual-rail domino
 - NP CMOS

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



$$\begin{aligned}
 S &= A \oplus B \oplus C_i \\
 &= \overline{A}\overline{B}C_i + \overline{A}B\overline{C}_i + A\overline{B}\overline{C}_i + ABC_i \\
 C_o &= AB + BC_i + AC_i
 \end{aligned}$$

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

Define 3 new variables which ONLY depend on A, B WHY?

Generate (G) = AB

Propagate (P) = A ⊕ B

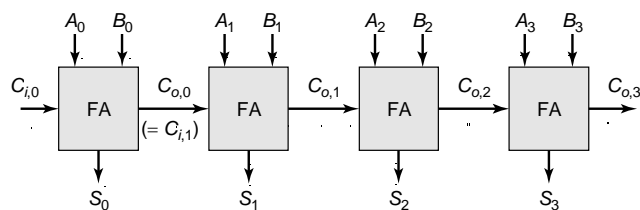
Delete = $\overline{A} \overline{B}$

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

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

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

The Ripple-Carry Adder



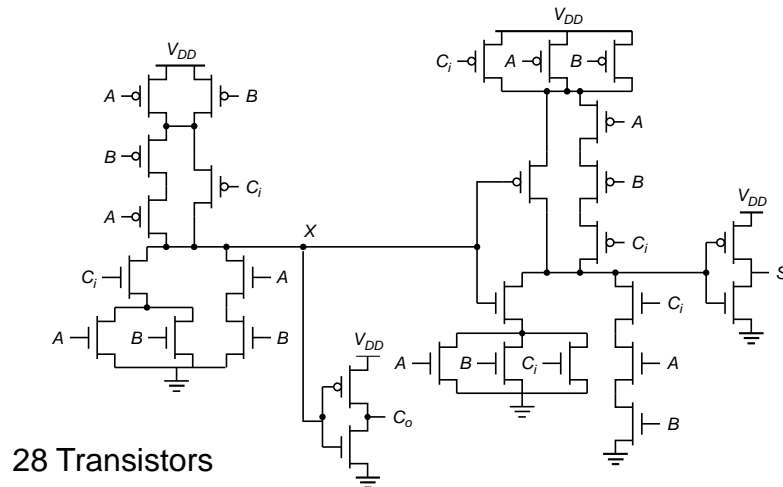
Worst case delay linear with the number of bits

$$t_d = O(N)$$

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

Goal: Make the fastest possible carry path circuit

Complementary Static CMOS Full Adder

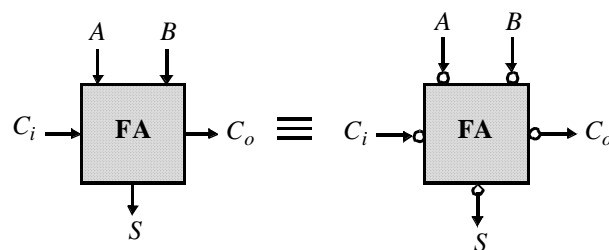


EECS 427 F08

Lecture 8

31

Inversion Property



$$\bar{S}(A, B, C_i) = S(\bar{A}, \bar{B}, \bar{C}_i)$$

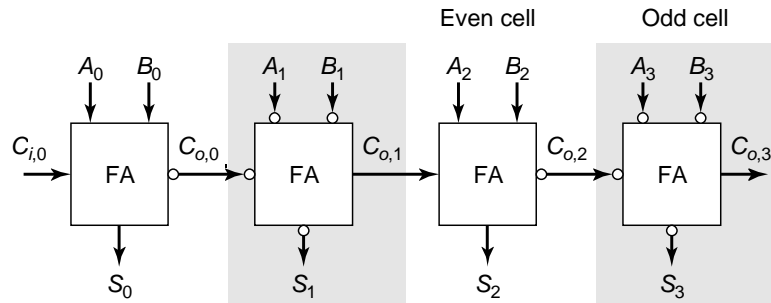
$$\bar{C}_o(A, B, C_i) = C_o(\bar{A}, \bar{B}, \bar{C}_i)$$

EECS 427 F08

Lecture 9

32

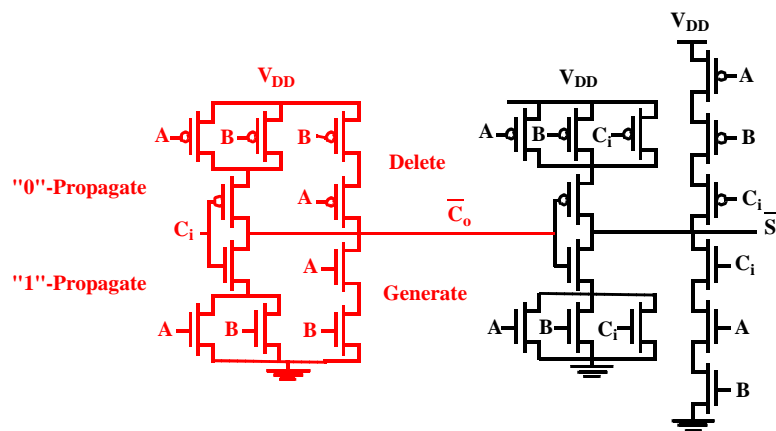
Minimize Critical Path by Reducing Inverting Stages Along Carry Path



Exploit Inversion Property

Allows us to remove inverter in carry chain → at what cost?

A Better Structure: Mirror Adder

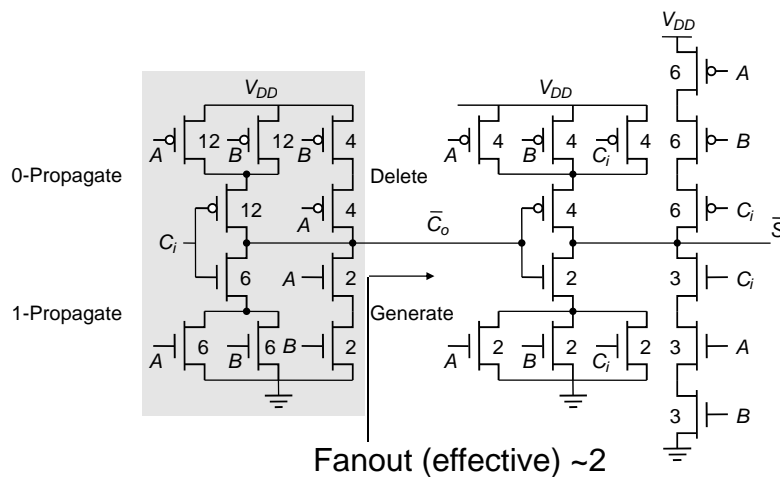


24 transistors

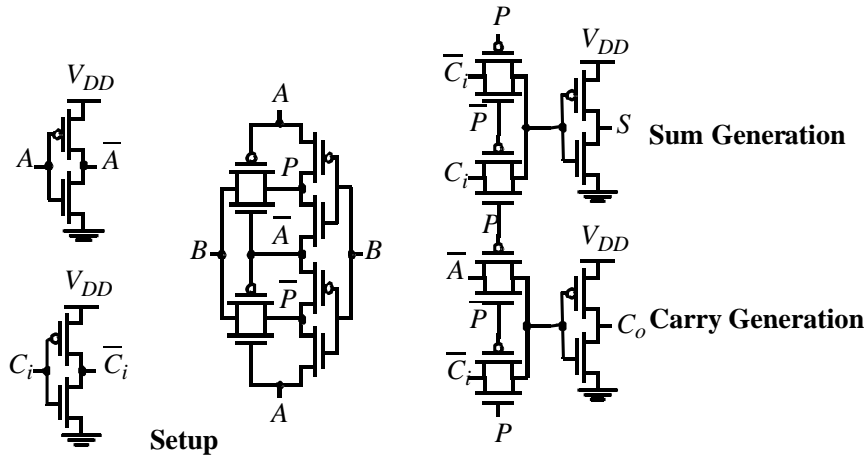
Mirror Adder Details

- NMOS and PMOS chains are **completely symmetric**.
Maximum of 2 series transistors in carry generation
- In layout → critical to minimize capacitance at node C_o .
Reduction of junction capacitances is particularly important
- Capacitance at node C_o is composed of 4 junction capacitances, 2 internal gate capacitances, and 6 gate capacitances in connecting adder cell
- Transistors connected to C_i are closest to output
- Only optimize transistors in carry stage for speed
Transistors in sum stage can be small

Sizing Mirror Adder



Transmission Gate Full Adder

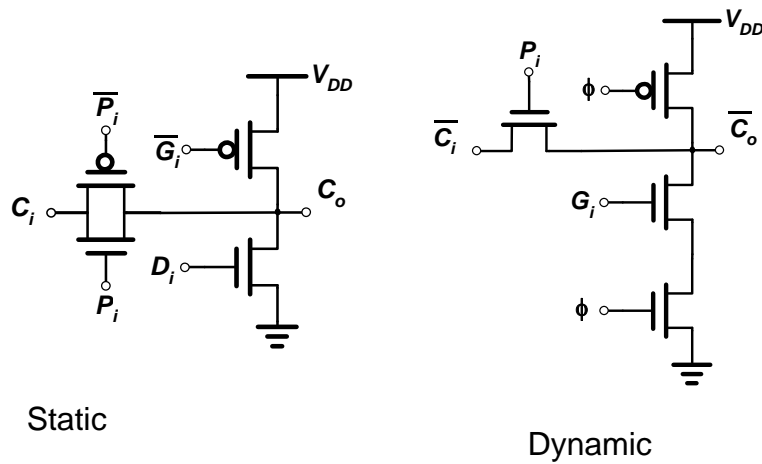


EECS 427 F08

Lecture 9

37

Manchester Carry Chain



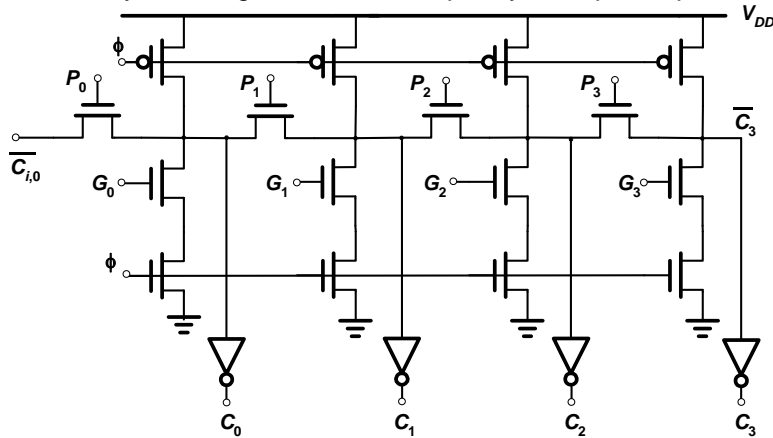
EECS 427 F08

Lecture 9

38

Manchester Carry Chain

- Implement P with pass-transistors
- Implement G with pull-up OR kill (delete) with pull-down (note inversion)
- Use dynamic logic to reduce complexity and speed up

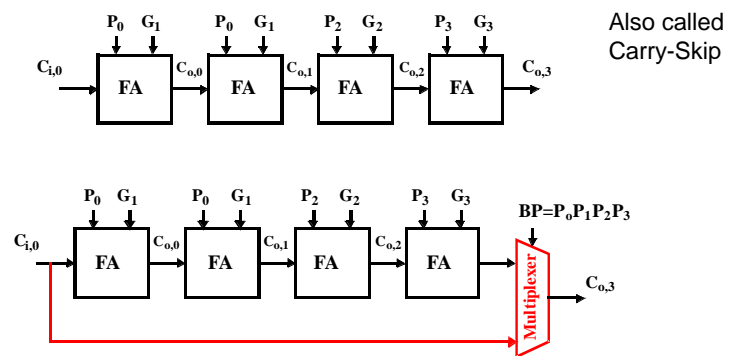


EECS 427 F08

Lecture 9

39

Carry-Bypass Adder



Also called
Carry-Skip

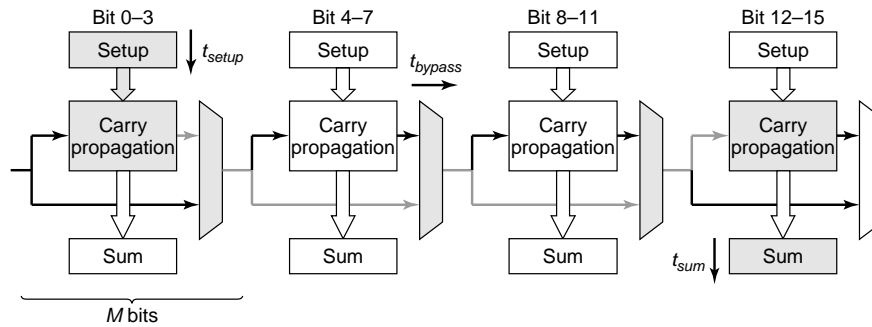
Idea: If $(P_0 \text{ and } P_1 \text{ and } P_2 \text{ and } P_3 = 1)$
then $C_{o3} = C_{i,0}$ else "delete" or "generate"

EECS 427 F08

Lecture 9

40

Carry-Bypass Adder (cont.)

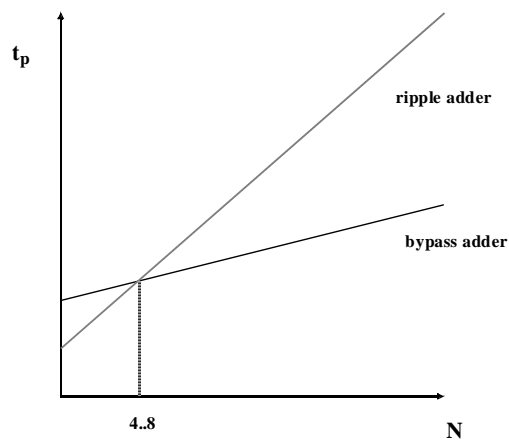


$$t_{adder} = t_{setup} + M t_{carry} + (N/M - 1) t_{bypass} + (M - 1) t_{carry} + t_{sum}$$

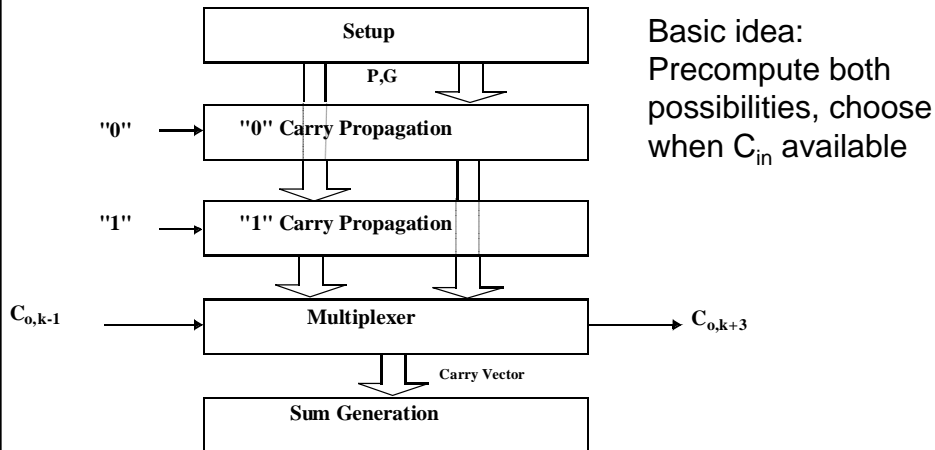
Inner blocks do not contribute to worst-case delay since they have time to compute while bits 0-3 are propagating (assuming they have a generate or delete)

Block sizes can be made non-uniform (HOW?)

Carry Ripple versus Carry Bypass



Carry-Select Adder

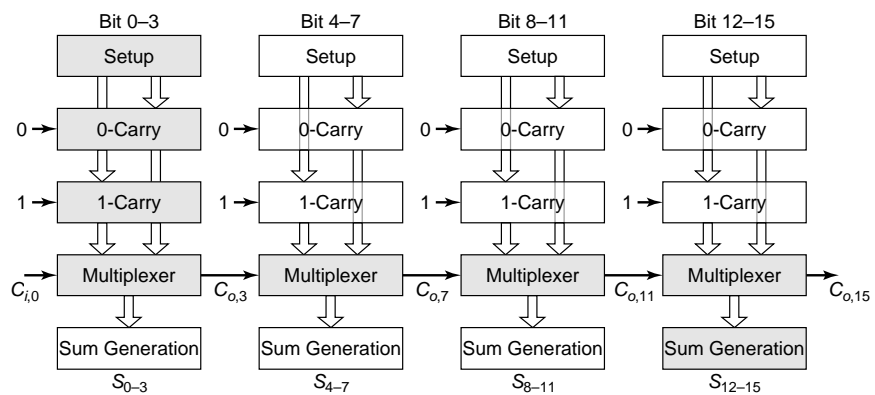


EECS 427 F08

Lecture 9

43

Carry Select Adder: Critical Path



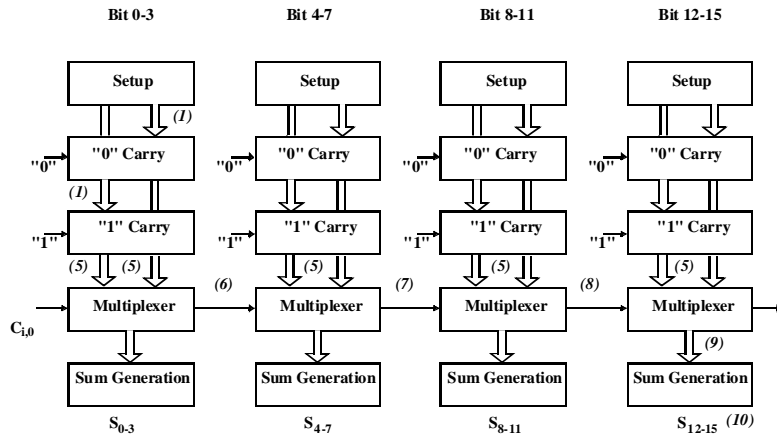
$$t_{add} = t_{setup} + Mt_{carry} + (N/M)t_{mux} + t_{sum}$$

EECS 427 F08

Lecture 9

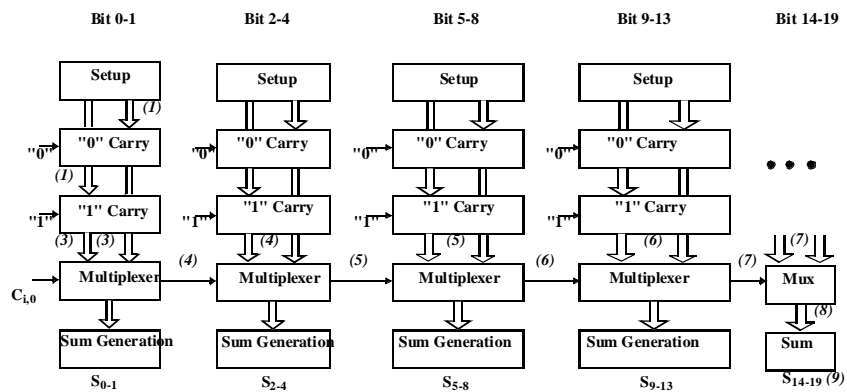
44

Linear Carry Select



$$t_{add} = t_{setup} + M \cdot t_{carry} + (N/M) t_{mux} + t_{sum}$$

Square Root Carry Select



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

Adder Delays - Comparison

