## Comments on Domino Logic Simulation

- When giving a power figure, you need to specify under what conditions the power figure was measured!
  - For problem #1 (size a domino inverter in an 11 stage circuit for power and speed), students gave power figures and implied that this was the power consumed when running at maximum speed when in fact it was measured for some slow clock speed, and hence the power figure was very low
  - To credit, some students either specified what clock speed power was measured at, or measured at maximum clock speed.
- The 2nd part asked to locate charge sharing in a domino logic gate
  - You MUST have spice waveforms in your report to back up any statements about circuit malfunction such as charge sharing!!!

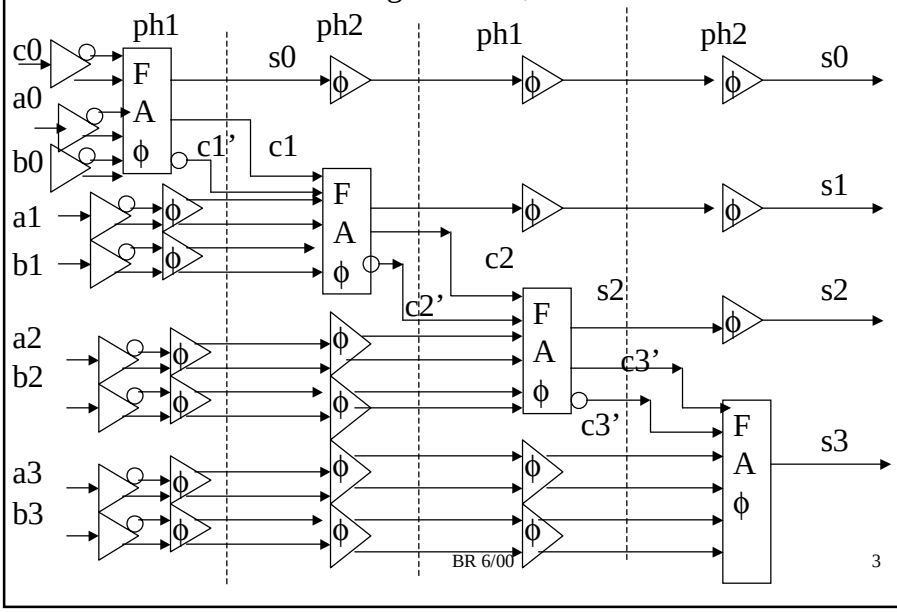


## Comments on Pipelined Adder Simulation

- Top speed (500 ps) performer used a clever architecture technique to increase performance
  - was also close to being best PDP
- Top power performer used a dual rail domino circuit
- Most working designs used 600-700 ps as period
- At least two designs used static inverters to provide complemented inputs to full adder circuits
  - It turns out the circuits still worked because of the logic used for the full adder, but this will fail for arbitrary logic!!!!

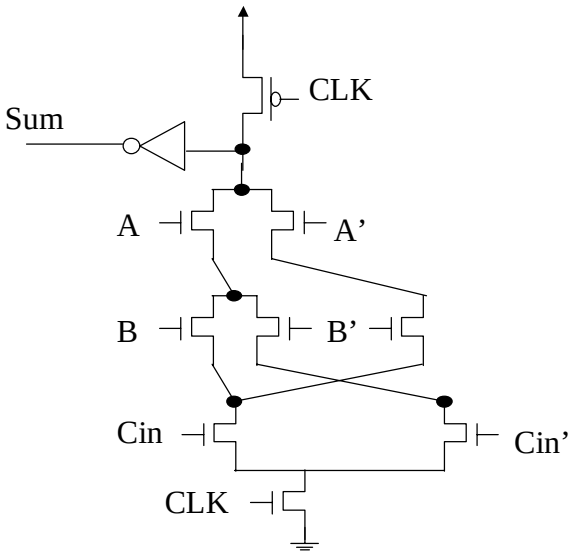
## Most common approach – 600,700 ps

Dual rail domino used to generate C, C’ from full adder

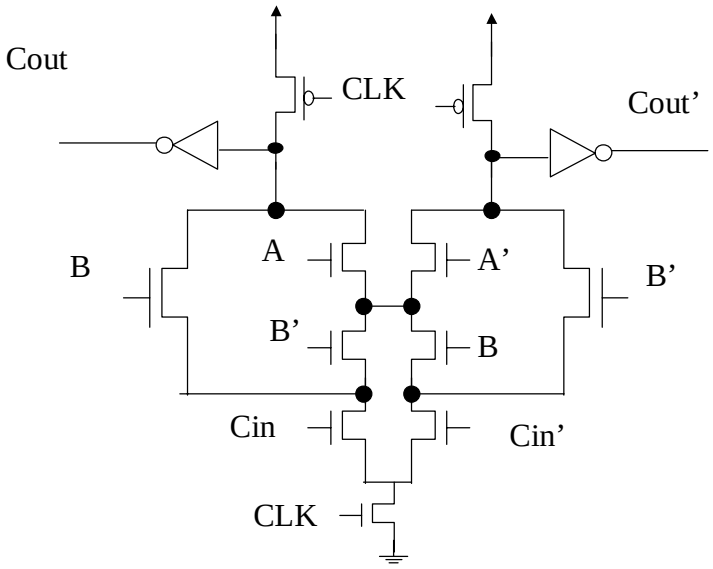





## Full Adder: Sum logic



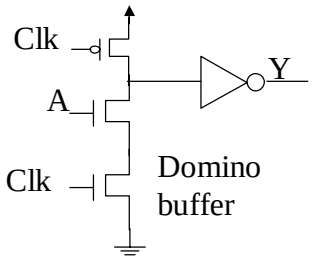
## Full Adder, Cout logic

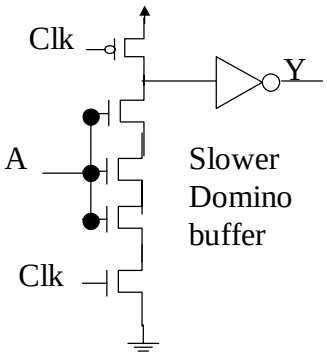





## Race Through

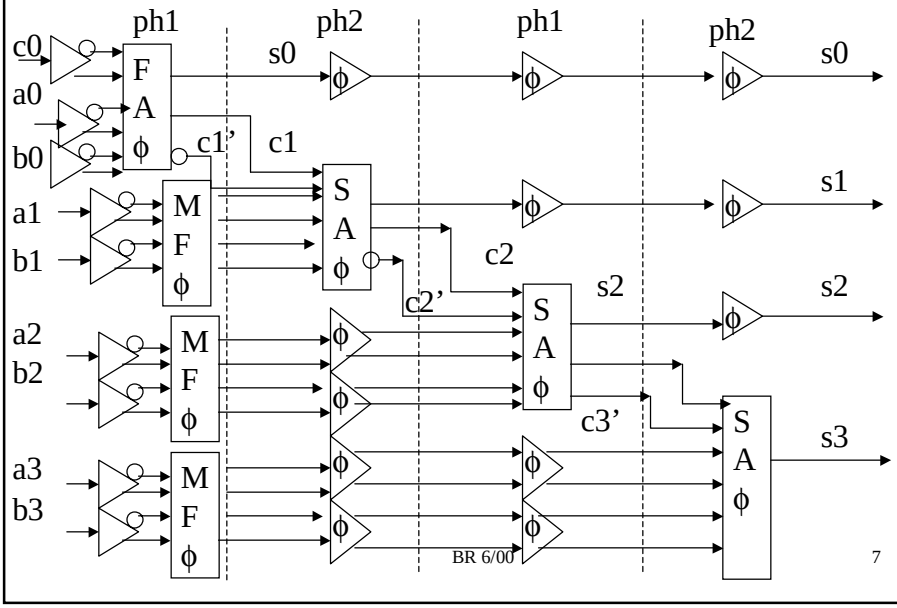
- If the overlap period was too long, race through from one stage to the other could be a problem. The delay of the domino buffers needed to match the delay of the FA circuit
  - Could use multiple domino buffers, but this consumed power
  - Can 'slow' down the domino buffer by placing multiple N-transistors in series



## Winning Approach (speed) – 500 ps

Dual rail domino used to generate C, C' from full adder

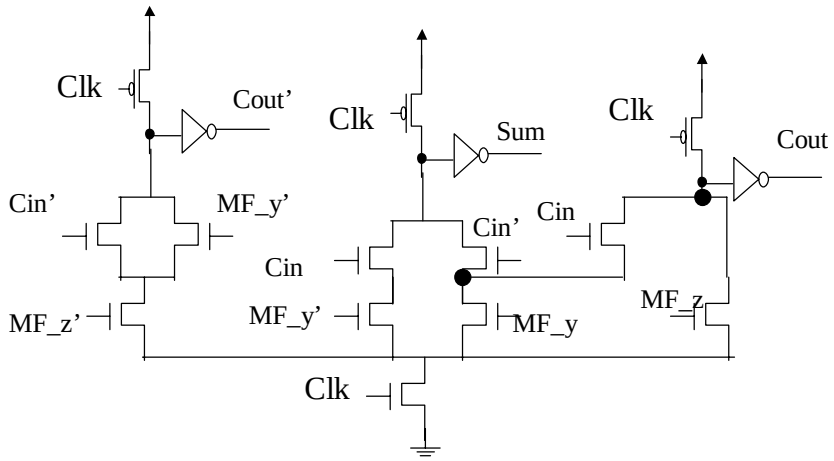





## Comments on Winning Circuit

- The MF blocks in first stage computed some terms needed by the FAs in stages 2,3, 4
  - MF_y = A xor B, MF_y' = A xnor B
  - MF_z = A and B, MF_z' = A' or B'
- This allowed the evaluation blocks of the SA (simplified Adder circuits) to only have *three transistors* (2 in N-tree + clock eval transistor) in series instead of four
  - SUM = Cin xor (MF_Y)
  - Cout = MF_z or (Cin and Mf_y)
  - Cout' = MF_z' and (Cin' or MF_y')
- The first FA (in stage 1) still had *4 transistors in series.* This FA was sized to be very high performance so as to match delay of the MF blocks. This cost some power, but this FA was only used once.

## SA Design

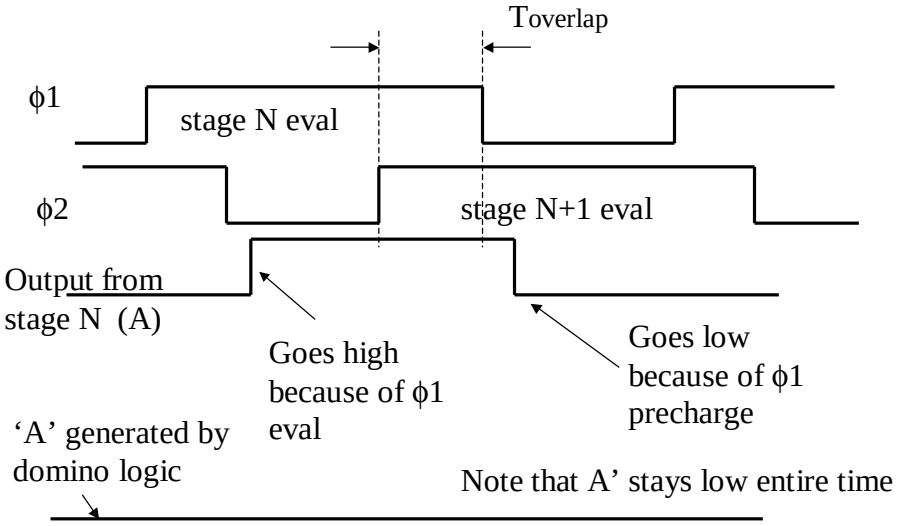





## Static Inverters

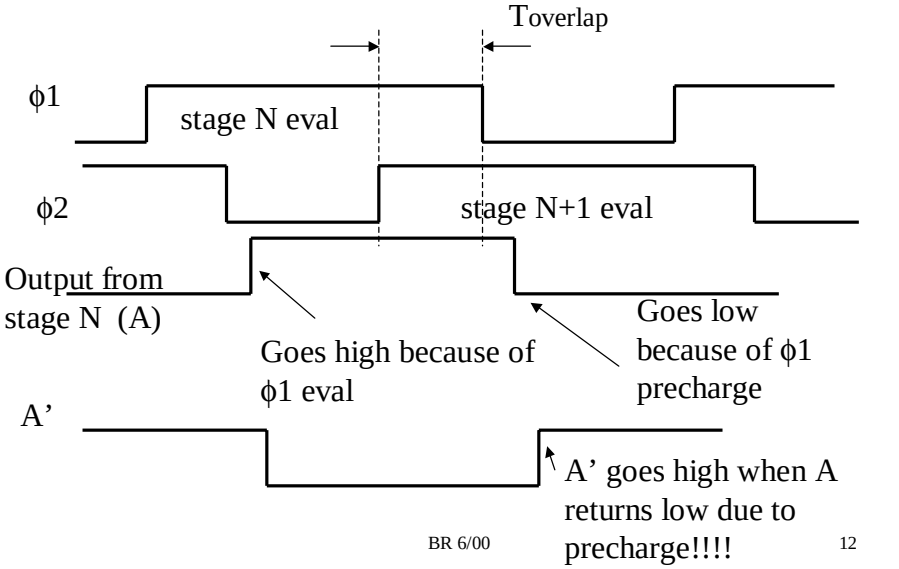
- At least two designs did not generate both Cout and Cout' bar in their designs and did not carry both A, A' and B, B' through the stages
- Instead, used static inverters inside of full adder circuits
- I mentioned in the notes that this could cause a problem
  - However, in their designs, static inverters worked ok. Why?

## *A* and *A'* generated by Domino Logic from Stage N





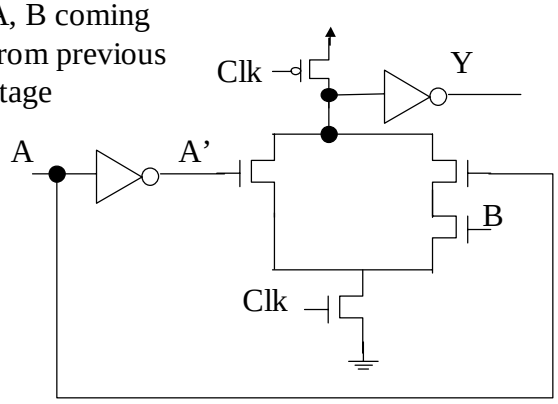

## *A* generated by Domino Logic, *A'* generated by static inverter



## What type of logic will fail?

When A' goes high because of A returning low due to precharge, circuit can be upset due to A' going high!!!!!

A, B coming from previous stage

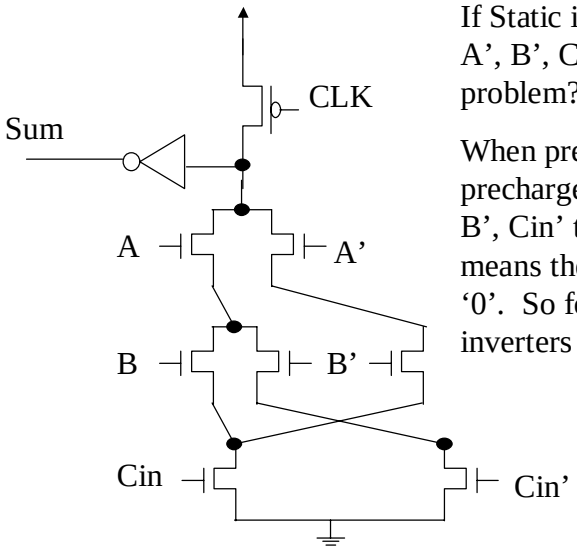



If A = 1 and B=0, then Y = 0.

However, when A returns to 0 during precharge, A' becomes 1, so Y goes to 1!!!!!



## Full Adder: Sum logic




If Static inverters used for A', B', Cin', will this cause a problem?

When previous stage begins precharge, this will cause A', B', Cin' to become '1'. This means the sum output will be '0'. So for this logic, static inverters are ok.