

Dynamic 16-bit Carry- Lookahead Adder/Subtractor

Jason Bosko

John Choi

Paul Verheggen

The carry-lookahead is a fast adder designed to minimize the delay caused by carry propagation in basic adders. It utilizes the fact that, at each bit position in the addition, it can be determined if a carry will be generated at that bit, or if a carry will be propagated through that bit. We have used domino logic for improved performance.

Input/Output

Input

- Two 16-bit numbers = 32 inputs
- One input to control whether to add or subtract
- Power, ground
- A clock for precharging dynamic gates

Output

- One 16-bit result
- One bit representing any leftover carry

Basic Idea

Given two 16-bit numbers, X and Y, the carry-bit into any position is calculated by:

$$c_{i+1} = X_i Y_i + X_i c_i + Y_i c_i$$

$$c_{i+1} = X_i Y_i + (X_i + Y_i) c_i$$

$$c_{i+1} = g_i + p_i c_i$$

Where $g_i = X_i Y_i$ and $p_i = X_i + Y_i$

We can pass these along to eliminate unnecessary delay from carry propagation.

Basic Idea

For example: if you want to know if a carry is generated at bit position 3, then:

$$c_3 = G_2 + G_1 P_2 + G_0 P_1 P_2$$

Basic Gates

- AND: 2-input, 3-input, 4-input, 5-input
- OR: 2-input, 3-input, 4-input, 5-input
- XOR: 2-input static gate (used for 1's compliment when subtraction is needed and for basic addition)

Dynamic Circuit

- We want to make these basic gates as fast as possible – use dynamic circuits!
- Eliminates need for pull-up network with big PMOS devices
- Use footed gates

Adding Domino Logic

- We will be using the output of dynamic gates as input to other dynamic gates – but dynamic gates require monotonic inputs and produce non-monotonic outputs!
- Use domino logic – each dynamic gate is followed by a static inverter to restore monotonicity.

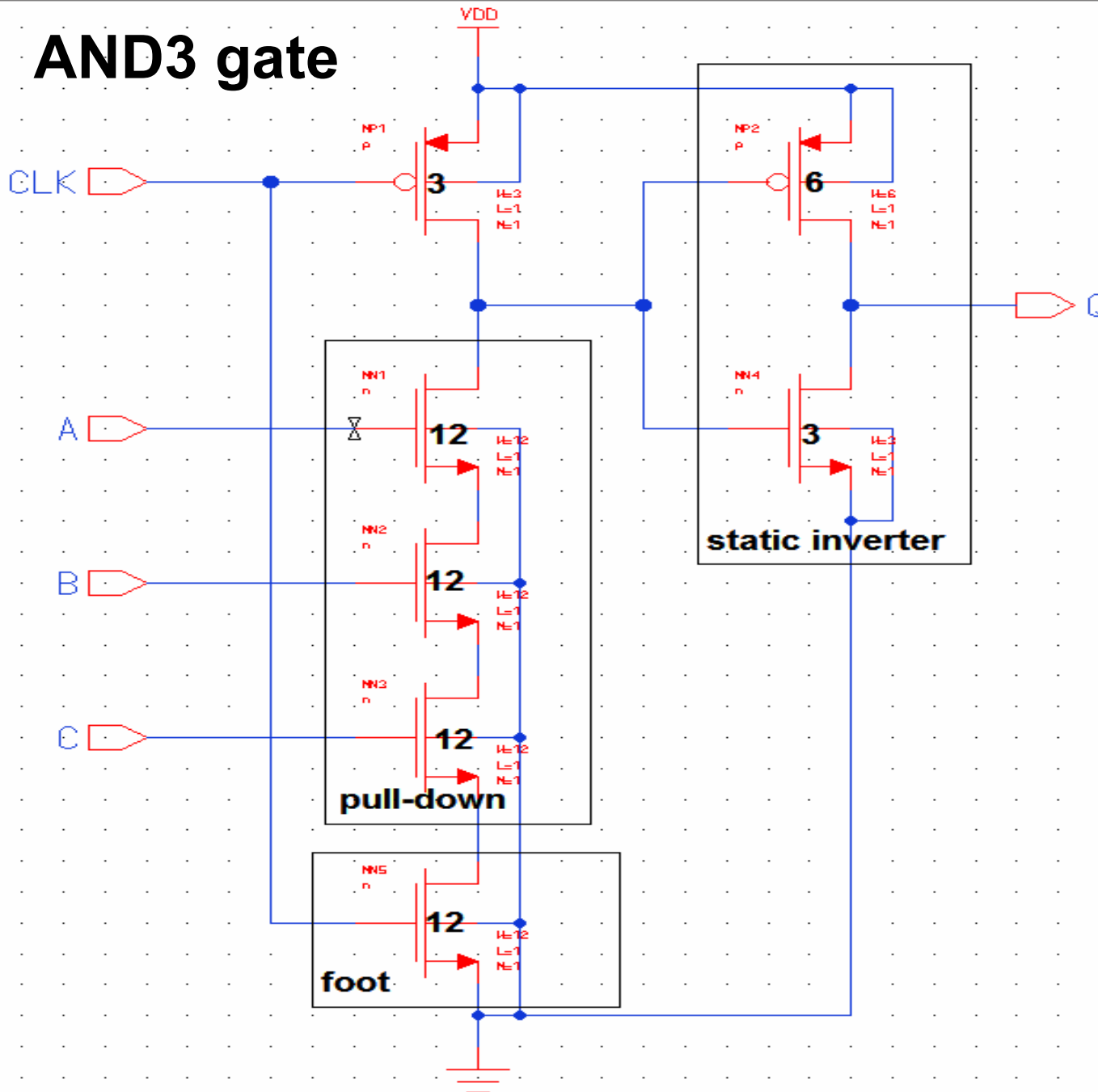
Dynamic + Domino

- Using domino logic results in naturally noninverting gates.
- CLA is comprised of lots of ANDs and ORs – noninverting logic is good!

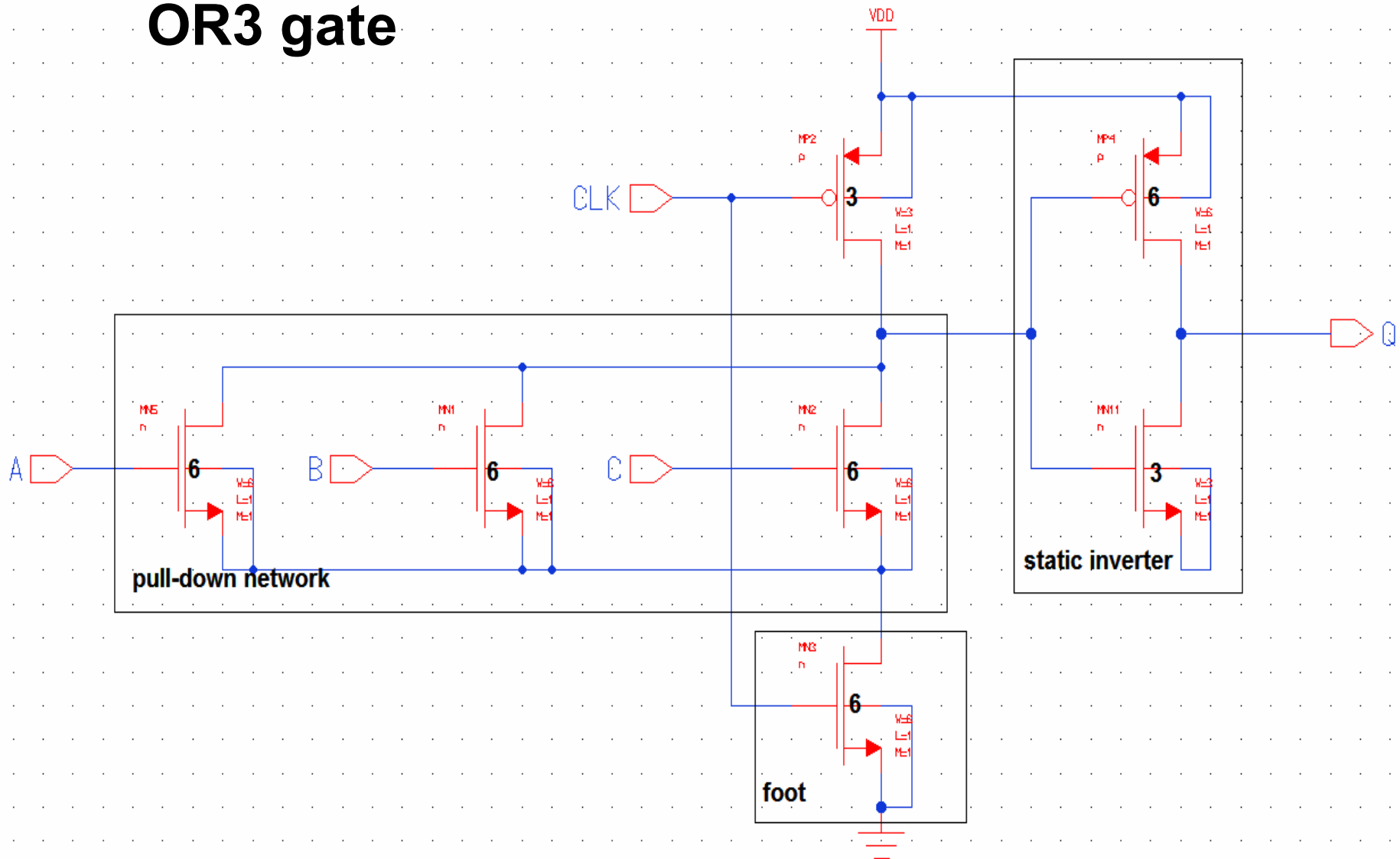
Downside to Domino Logic

- Larger dynamic power consumption.
- Issues of charge leakage and charge sharing (solved with keepers and secondary precharging).
- Lower tolerance for noisy signals.
- Requires very careful design!

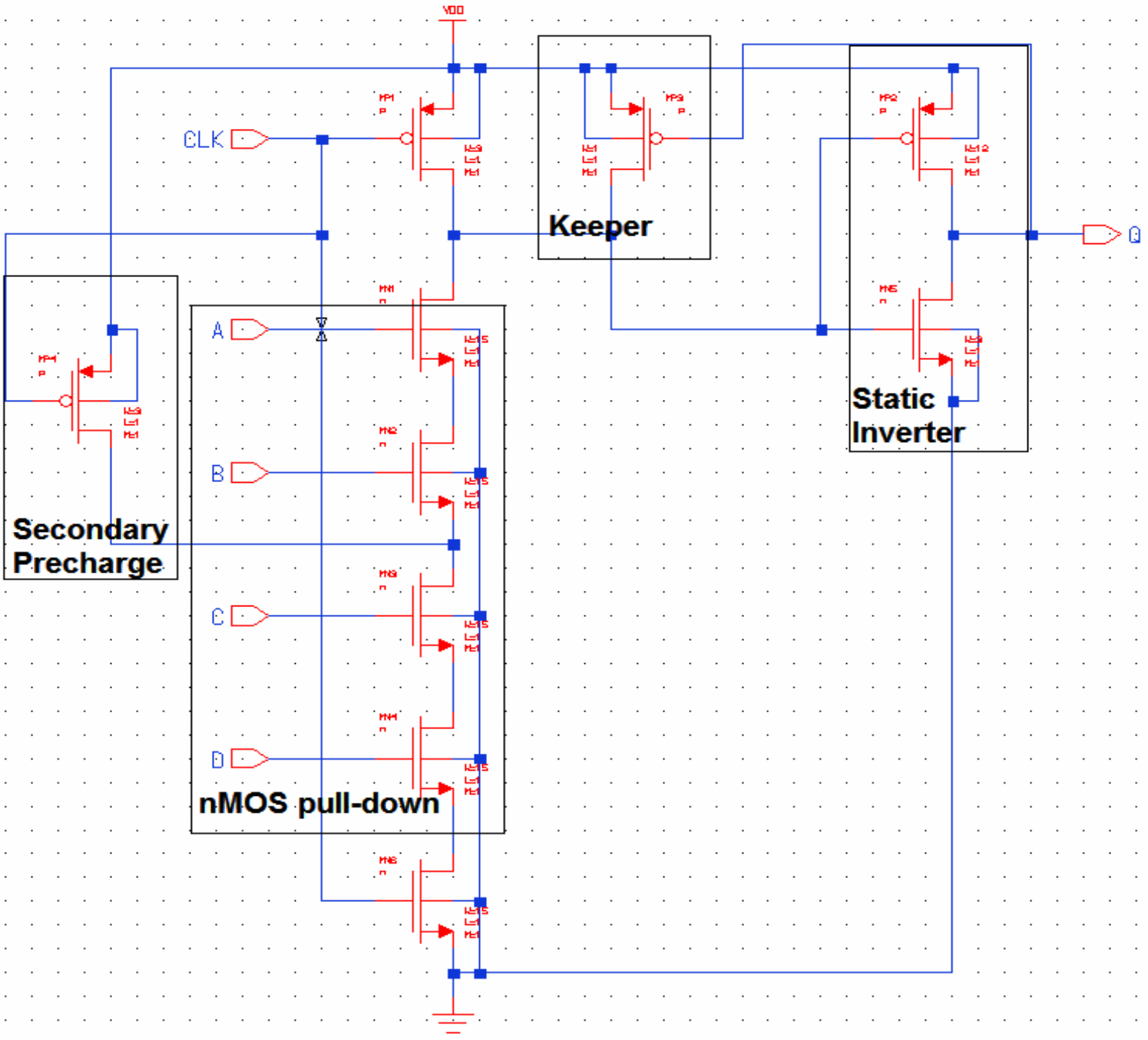
AND3 gate

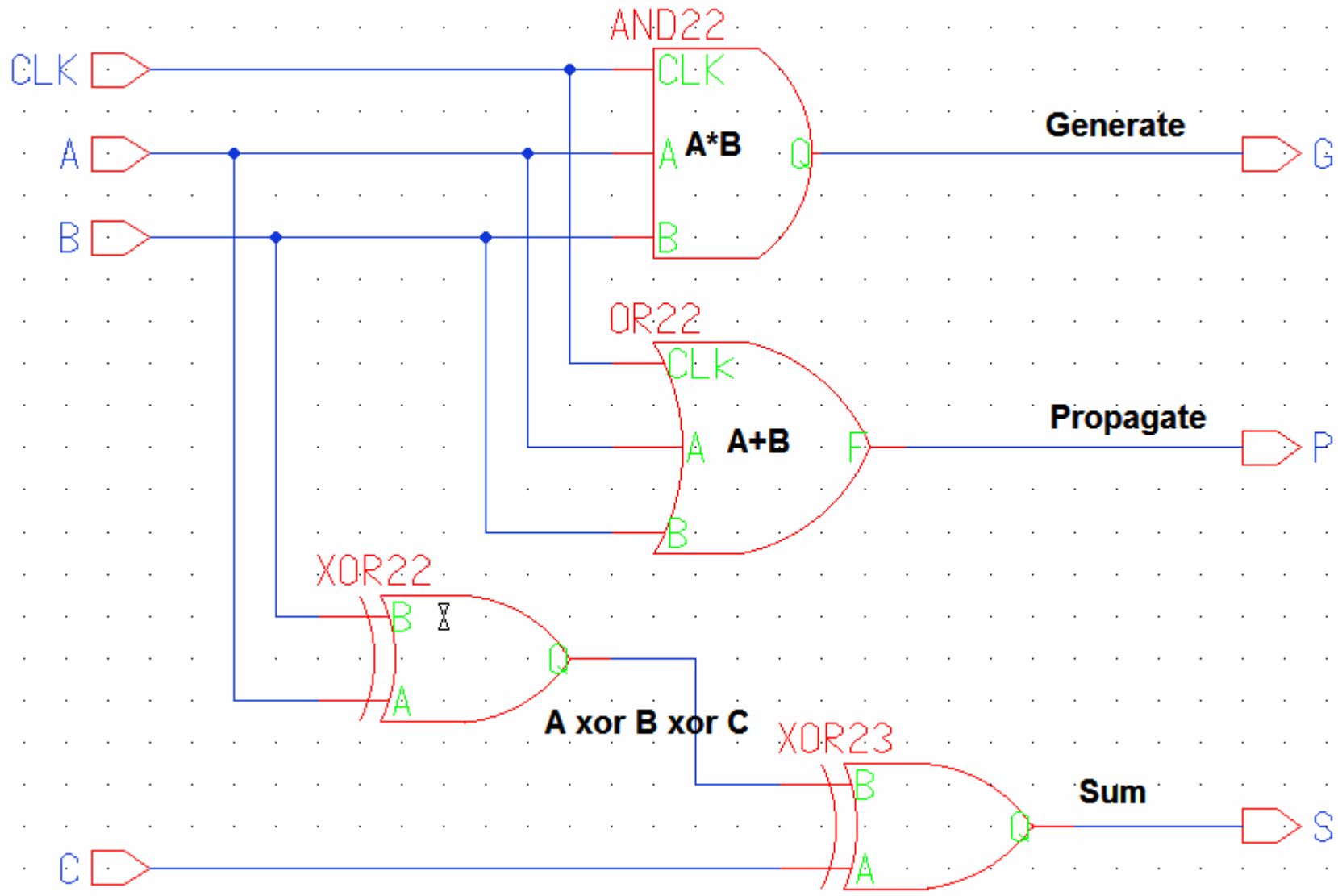


OR3 gate

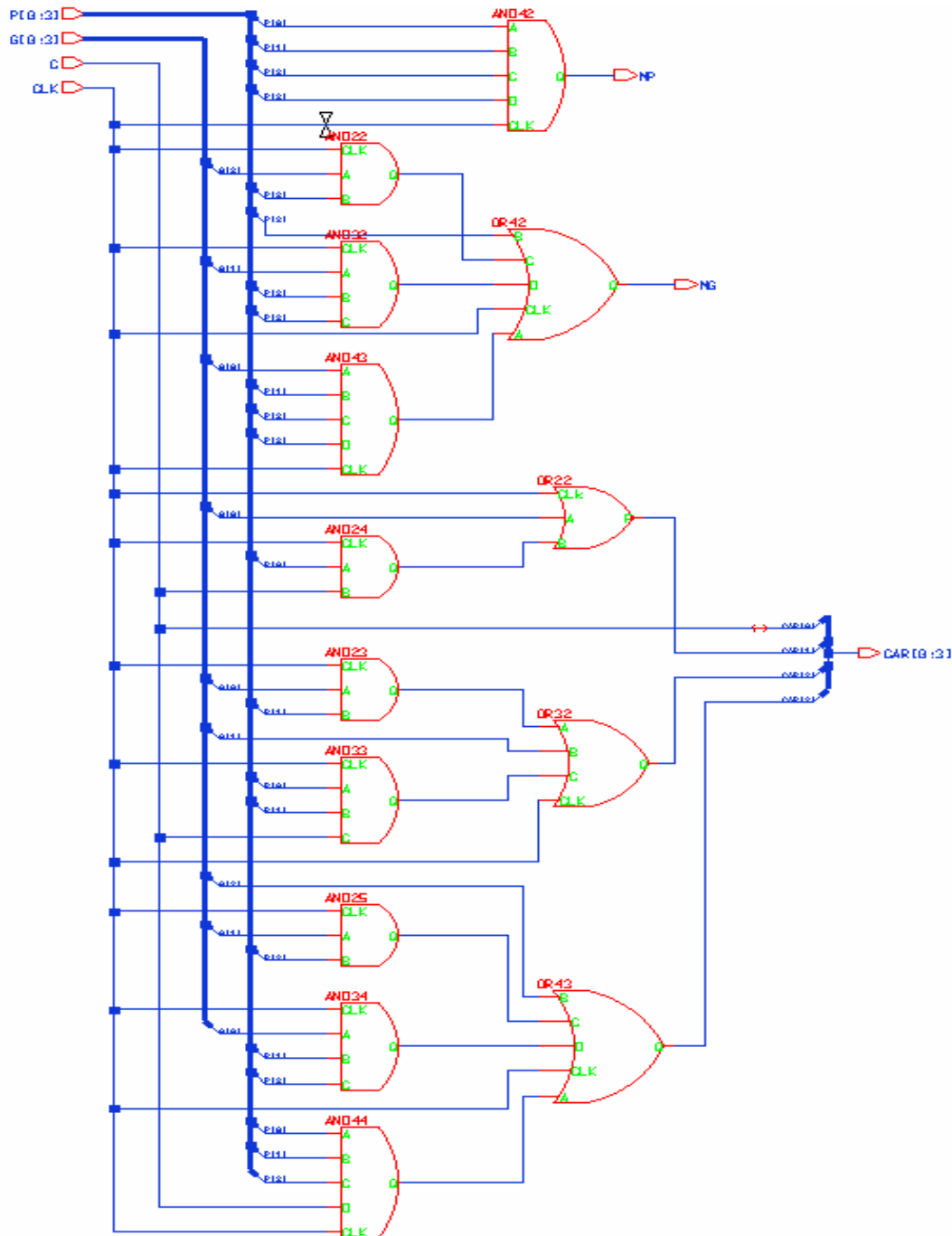


AND4 gate with keeper and secondary precharge

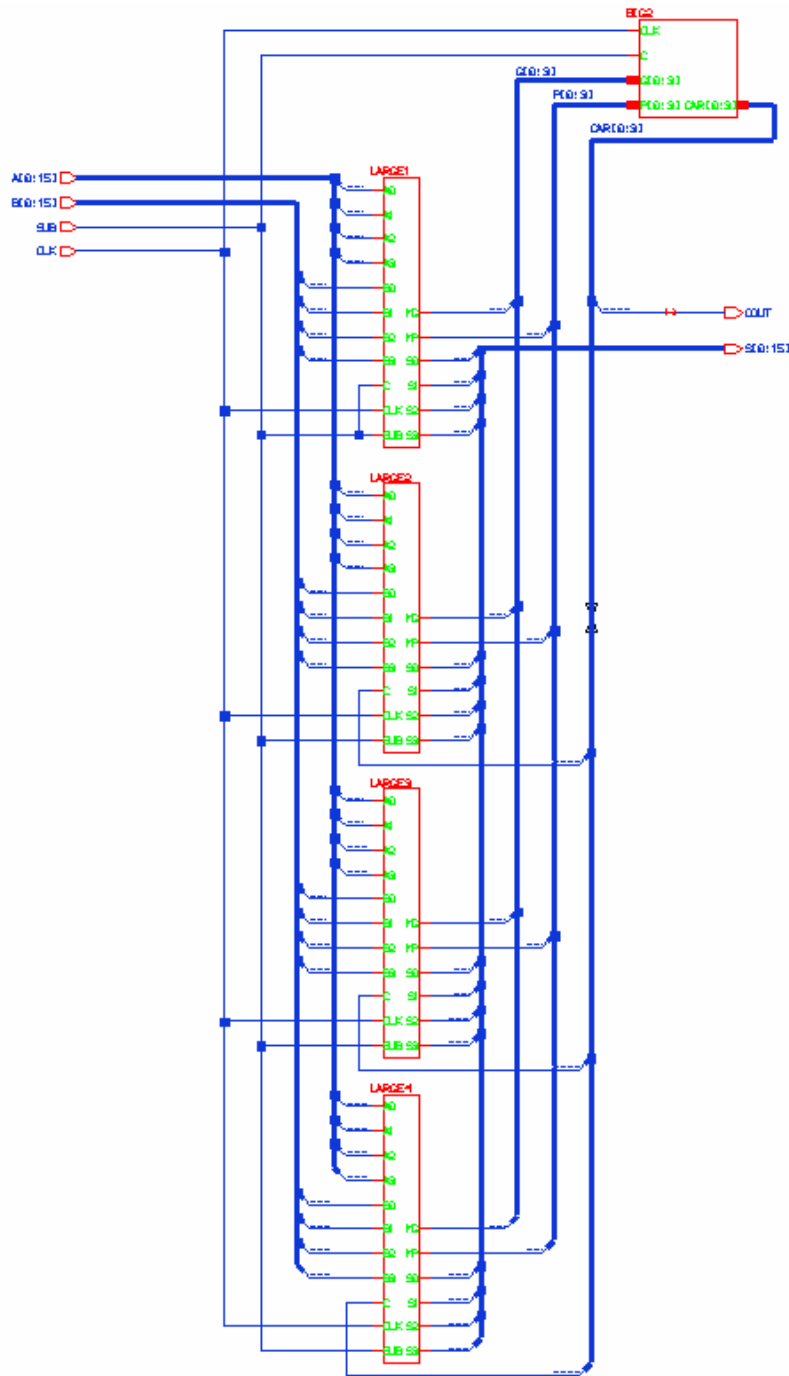




This generates the “generate carry” and “propagate carry” bits at a given position, and also calculates the sum at the current position



This performs the logic steps for combining generate and propagate bits into actual carry bits to send back to the adders.



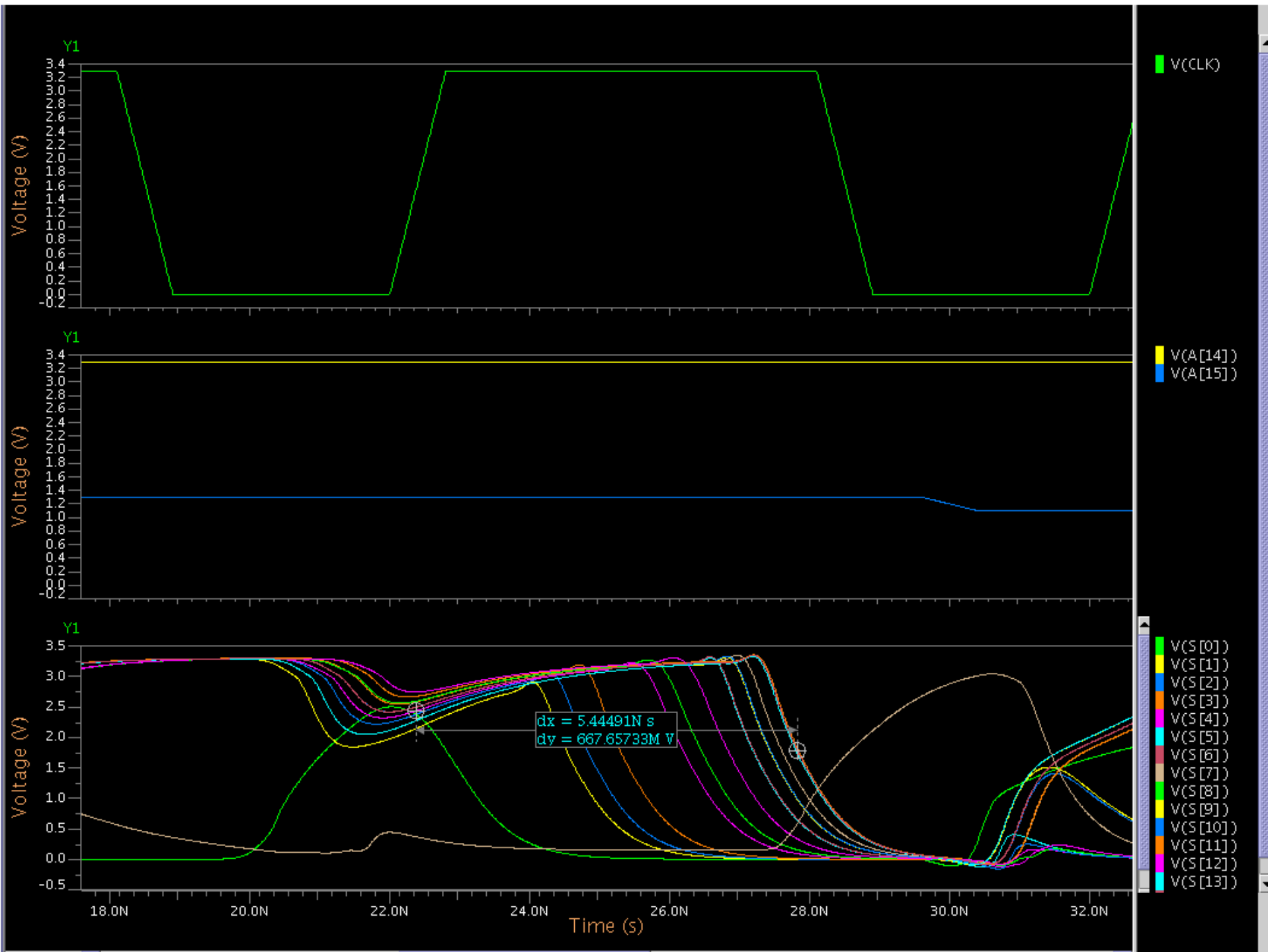
This is the top-level schematic. There are four copies of the previous schematic, which feed their “master generate” and “master propagate” signals into another logic block, which then feeds the carry bits back into each of the four instances.

Clock

- Fastest clock achieved was 100 MHz.
- How can we do better? Use pipelined stages – but this would require registers that would significantly increase area

Analog Simulation Results

- Noise sensitivity
 - noise margins: $NM_H = 3.2 - 1.35 = 1.85V$, $NM_L = 1.1 - 0.6 = 0.5V$
 - Power supply noise $< 1.15V$ P-P tolerated.
- $t_{pd} = 5.4ns$ (corner case)



Area

- Total area was $1100\lambda \times 650\lambda$. In $0.5\mu\text{m}$ technology, this is $\sim 0.17875\text{mm}^2$
- Original estimate was 2.5mm^2 . What happened?
 - Use of dynamic logic replaces big pMOS networks with one small pMOS transistor. Even with domino logic (static inverter), keepers, and secondary precharge, dynamic gates save space!