Digital Logic Circuits
'Performance Analysis'
ELEC2200
Summer 2009

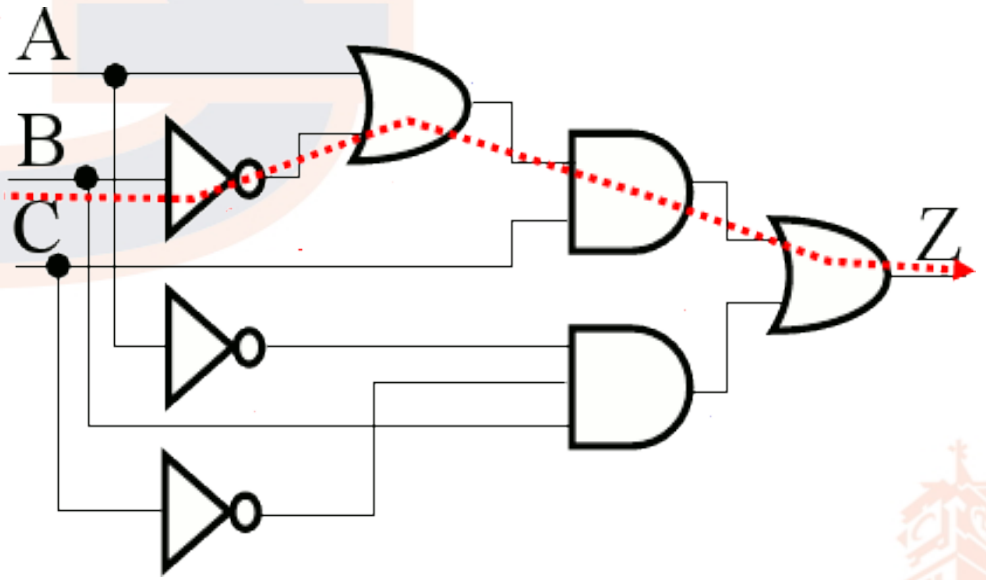
David J. Broderick
brodedj@auburn.edu
http://www.auburn.edu/~brodedj
Office: Broun 360

You Are Here

- We can:
  - Represent real values with binary numbers
  - Use Boolean algebra to manipulate binary expressions (minimize)
  - Use K-maps to find minimal SOP/POS expression
  - Once you have an expression, draw the circuit
  - Reconstruct the truth table and expression given a circuit
- We're left with what's the difference between a good circuit and a great circuit?

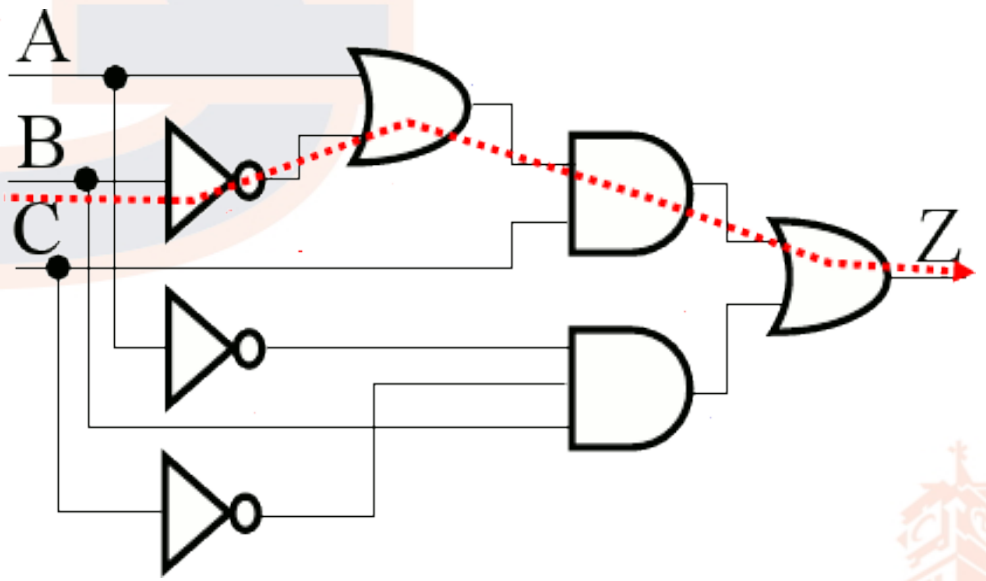
Area Analysis

- How small will the circuit be?
  - # of gates
    - More gates, bigger circuit
  - # of inputs and outputs to all gates
    - Adding inputs make a gate bigger
    - More inputs/outputs, bigger circuit
- Z=(A+B')C+A'BC'
- G=7
- G_{IO}=19



Performance Analysis

- How fast will the circuit switch states?
- In the worst case, how long does it take for a change in the inputs to cause a change in the output?
- Gate Delay
    - # of gates between input and output in the worst case path
- Z=(A+B')C+A'BC'
- G_{del}=4
- Worst case path=B-->Z

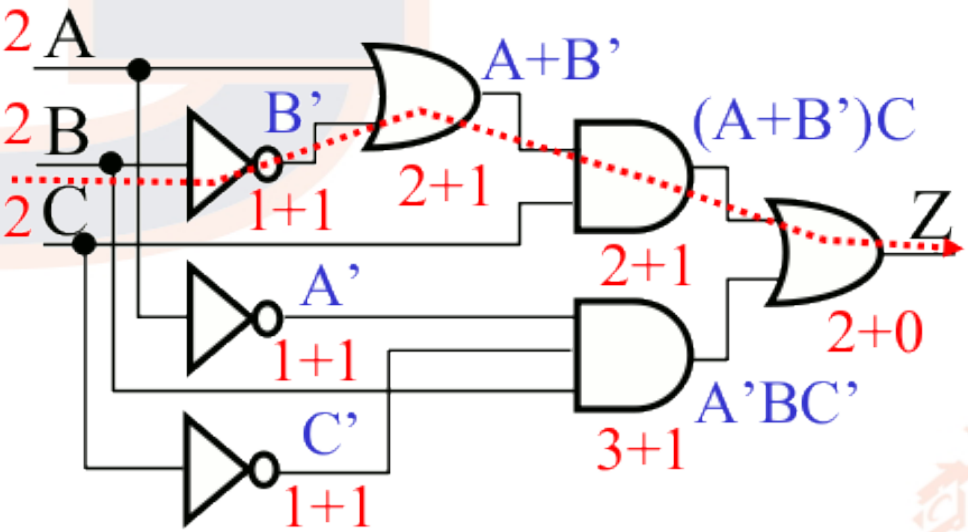


Performance Analysis

- Propagation Delay, a more accurate measure
- Propagation Delay=intrinsic delay+extrinsic delay
- Relative Propagation Delay= # of inputs to gate(intrinsic) + # of loads on gate(extrinsic)
- Assume all gates fabricated with similar design

Performance Analysis

- Still dealing with worst case path, though it may be different
- Add the number of inputs + number of outputs to a gate
- Add these values through the worst case path
- Z=(A+B')C+A'BC'
- P_{del}=12
- Worst case path=B-->Z

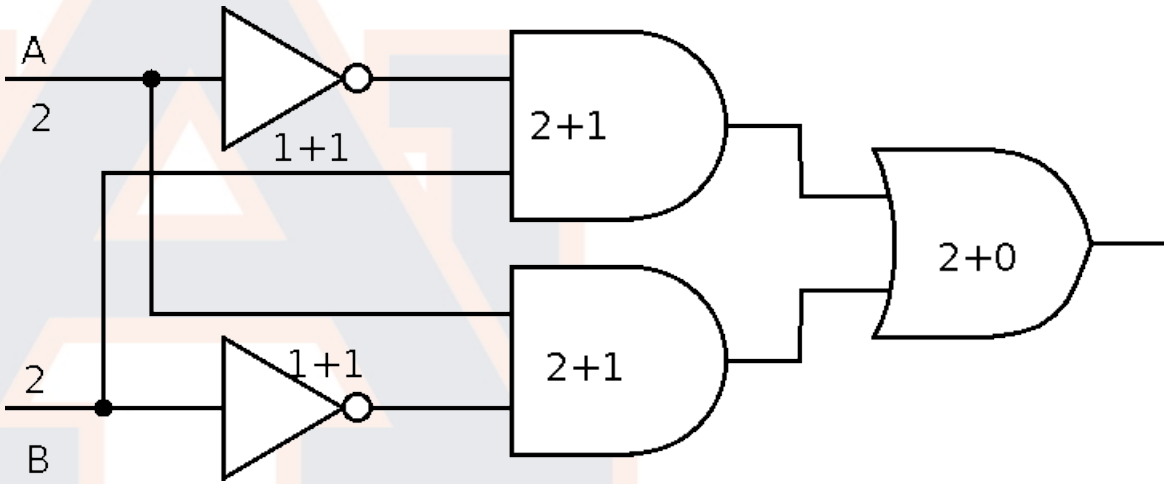


Performance Metrics

- Area Analysis
  - # of gates (G): Most common
  - # of gate input/outputs (G_{IO}): More accurate
- Performance Analysis (worst case path)
  - How quickly does change in inputs propagate to change in output?
  - # of gates in worst case path (G_{del})
  - Propagation Delay: More accurate measure of delay

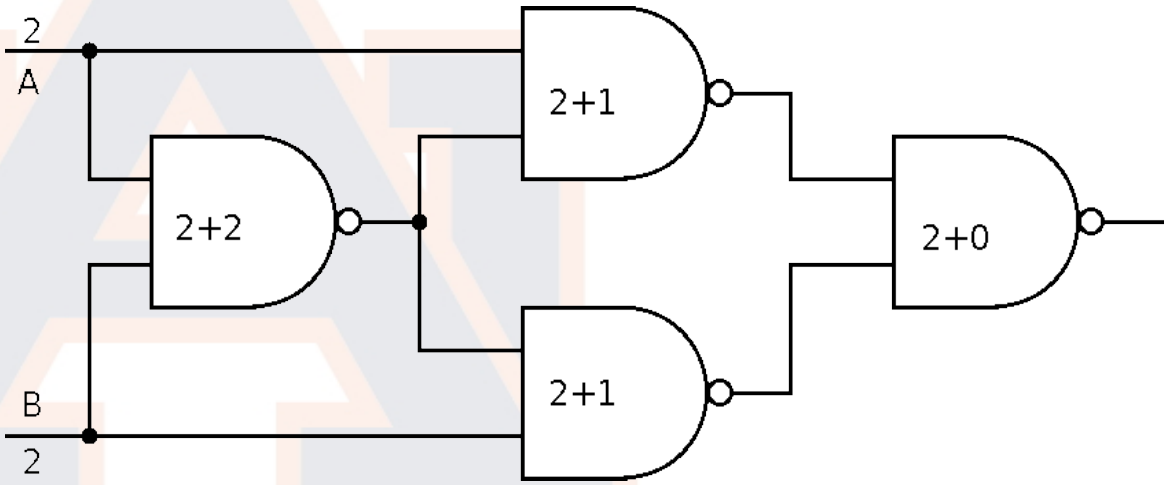
Performance Metrics

- A look a XOR circuits
- MSOP Form
  - G=5
  - G_{IO}=13
- Worst case:
  - Either path
  - G_{del}=3
  - P_{del}=9



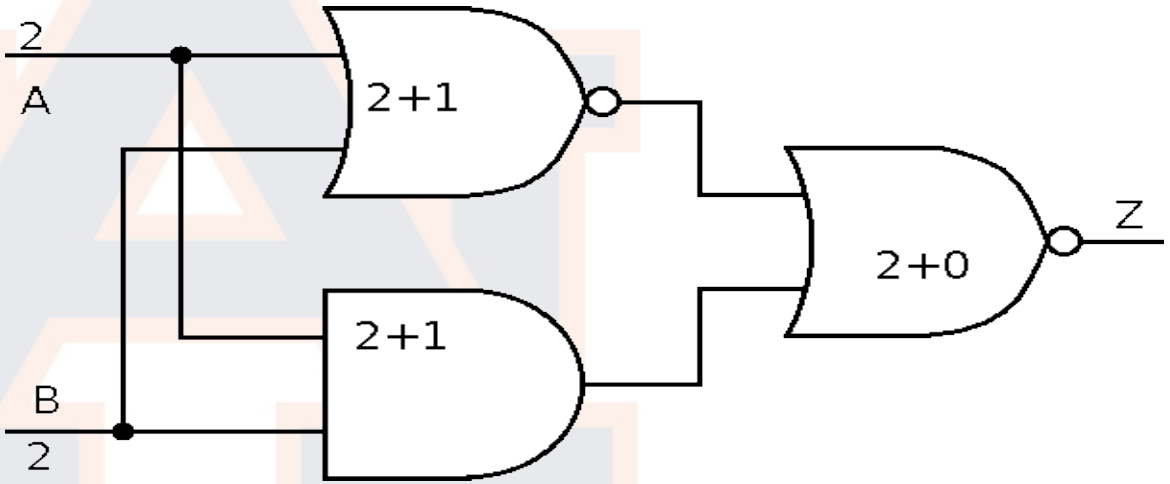
Performance Metrics

- A look a XOR circuits
- NAND-NAND
  - G=4
  - G_{IO}=12
- Worst case:
  - Either path
  - G_{del}=3
  - P_{del}=11



Performance Metrics

- A look a XOR circuits
  - G=3
  - G_{IO}=9
- Worst case:
  - Either path
  - G_{del}=2
  - P_{del}=7



XOR Performance

- MSOP is a good start
- NAND-NAND makes circuit smaller at coist of performance
- 3rd circuit minimizes both area and performance
- **Don't assume the MSOP is an optimized circuit!**

Circuit Optimization

- We want smallest, fastest circuit
- Basic Goals
    - Minimize the # of product terms
    - Minimize the # of literals
- K-maps a very effective tool
- Further optimization possible with Boolean algebra