Review Problem 7

* Simplify the following Boolean Equation

\[ AB + AC + \overline{AB} \]

\[ = AB + \overline{AB} + AC \quad \text{Comm} \]

\[ = B(A + \overline{A}) + AC \quad \text{Distrib} \]

\[ = B + AC \quad \text{Identity} \]
DeMorgan’s Law example

- If \( F = (XY+Z)(\bar{Y}+\bar{X}Z)(XY+\bar{Z}) \),

\[
\bar{F} = \left( (XY + z) (\bar{Y} + (\bar{x} + z)) (\bar{x}\bar{Y} + \bar{z}) \right) \\
= \left( (\bar{x} + \bar{Y} + z) + (\bar{y} + (x + z)) + (x + y + z) \right)
\]
Boolean Equations to Circuit Diagrams

\[ F = \overline{XYZ} + \overline{X}Y + XYZ \]

\[ F = XY + X(WZ + \overline{WZ}) \]
Circuit Timing Behavior

- Simple model: gates react after fixed delay

![Diagram of circuit timing behavior]

Stable for all time before this

1 gate delay
Hazards/Glitches

- Circuit can temporarily go to incorrect states

**Circuit Diagram**

- Copilot Autopilot Request
- Pilot in Charge?
- Autopilot Engaged
- Pilot Autopilot Request

**Timing Diagram**

- CAR
- PIC
- PAR
- A
- B
- C
- AE
Field Programmable Gate Arrays (FPGAs)

Logic cells imbedded in a general routing structure

Logic cells usually contain:
- 6-input Boolean function calculator
- Flip-flop (1-bit memory)

All features electronically (re)programmable
Using an FPGA

Verilog

FPGA CAD Tools

Simulation

Bitstream
Verilog

- Programming language for describing hardware
  - Simulate behavior before (wasting time) implementing
  - Find bugs early
  - Enable tools to automatically create implementation

- Similar to C/C++/Java
  - VHDL similar to ADA

- Modern version is “System Verilog”
  - Superset of previous; cleaner and more efficient
// Verilog code for AND-OR-INVERT gate

module AOI (F, A, B, C, D);
  output logic F;
  input logic A, B, C, D;
  assign F = ~(A & B) | (C & D);
endmodule

// end of Verilog code