Problem 1  **FPGA and ASIC Cost Analysis (Easy)**

Ben Bitdiddle has established a start up called Herp Derp Incorporated which specializes in making FPGA and ASIC solutions. He is debating whether to use an FPGA or ASIC solution for the launch of his first product. He goes to Alyssa P. Hacker who is now a consultant at L337 Associates who tells him that he should expect to sell 100,000 chips. She also tells him that the NRE costs associated with an ASIC solution and FPGA solution for his design are $1,000,000 and $100,000 respectively, and that the cost per unit for an FPGA solution and ASIC solution are $20 and $2 respectively. Since Alyssa is doing Ben a favor and not charging him for her time, she doesn’t tell him which one he should use. Given the above information, should Ben choose to use an FPGA or ASIC solution? Why?

Problem 2  **Finite State Machines (Easy)**

Consider the design of a simple edge detector circuit. The circuit has a one bit input $\text{In}$ and a one bit output $\text{Out}$. $\text{Out}$ should only be asserted for one cycle when the circuit detects a positive edge in the signal $\text{In}$ (i.e. a transition from zero to one). The $\text{Out}$ signal should be asserted the same cycle that the positive edge is detected (i.e. the cycle where the signal is high for the first time). Draw the Mealy style FSM
for this circuit and the corresponding circuit implementation using only registers and simple logic gates. (Make sure to include a reset signal \texttt{Reset}).

Problem 3  \textit{Finite State Machines}

Ben Bitdiddle and Alyssa P. Hacker are arguing over whether the following two finite state machines are functionally equivalent. Ben argues that the circuit are functionally equivalent because they will both assert the signal \texttt{Out} after the input \texttt{In} is asserted for three consecutive cycles. Alyssa disagrees and argues that the FSMs are not functionally equivalent. Who is correct and why?

Problem 4  \textit{Verilog Practice}

(a) Write a verilog module that implements the edge detector module \texttt{edge\_detect} specified in problem 2. Your module should take three inputs: the input signal
Problem 7

In, the clock signal clk, and the reset signal Reset. Your module should produce a single output Out which is asserted when a positive edge is detected. If a Reset signal is asserted, Out should be zero.

(b) Write a verilog module odd_pair that takes in a serial input In, clock signal clk, and a reset signal Rst and asserts a signal Odd if the signal In has been asserted for an odd number of cycles and an output signal Pair if the value of In in the previous cycle is the same as the current cycle. If a Reset signal is asserted, Pair and Odd should be zero.

Problem 5  Blast from the Past (Review)

Draw the CMOS transistor implementation of the following Boolean expressions. Minimize the number of transistors used:

(a) \((A + B)(DC + EC)\)

(b) \(\overline{A}B + \overline{C} + \overline{D}\)

Problem 6  Finite State Machines (Interview Question)

Design a Mealy style FSM that takes a one bit input in and asserts a one bit signal out if the sequence 11011 has been detected. The out signal should be asserted the same cycle that the last 1 in the sequence is detected. Sequence can also overlap, for instance if the sequence 11011011 is input, the signal out should be asserted on the 5th cycle, and the 8th cycle. Minimize the number of states in your FSM. Make sure to also designate what the initial state of the machine will be on a reset.

Problem 7  Bonus Question: The Logic Gate Pokemon (Optional)

Make up and draw a new pokemon that we can call the "Logic Gate Pokemon".