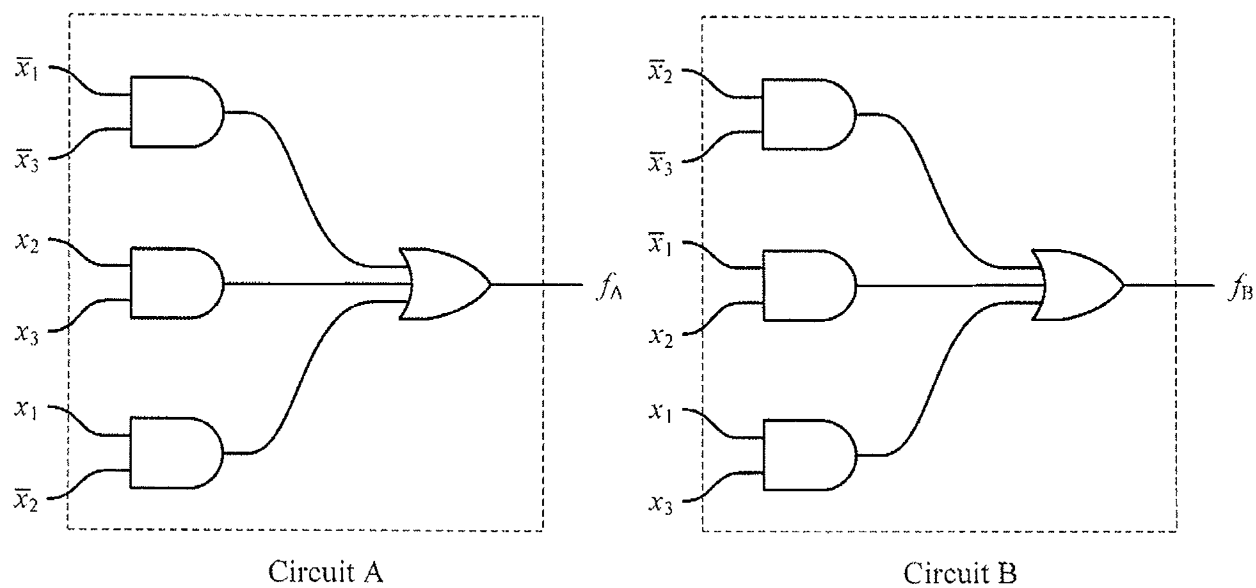
In this problem, you will be asked questions pertaining to CAD techniques for the *verification* of digital circuits. While everyone marvels at the physical scaling of integrated circuits, consider the computational scale. A combinational circuit, such as the arithmetic-logic unit (ALU) of a microprocessor, might have 256 inputs. Conceptually, such a circuit performs a mapping from a space of 2^{256} Boolean input values to Boolean output values. A truth table representing 2^{256} input assignments is inconceivable; verifying the behavior of such a circuit would seem like an intractable problem. Digital circuit designers have succeeded in their endeavor with data structures such as binary decision diagrams and implicit techniques such as Boolean satisfiability (SAT).

Verification Problem

Suppose that you're newly hired at Intel and you think that you've come up with a better design for the ALU of their latest generation of microprocessors. You're convinced that your design is correct but you had better be sure. How can you check?

Call Intel's existing design Circuit A and your new design Circuit B. Suppose, for the purposes of this exam, that Circuits A and B are those shown in Figures 1(a) and 1(b), respectively. Call the Boolean functions computed by these circuits f_A and f_B, respectively.



(a) Intel's existing design. (b) Your new design.

Figure 1: Circuits to be verified.

1. **Explicit Techniques** (*20 points*)

   Show, by evaluating f_A and f_B for all input combinations, that the two functions are equivalent.

   Solution

x_1	x_2	x_3	f_A	f_B
0	0	0	1	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

2. **Algebraic Techniques** (*20 points*)

   Write algebraic expressions for f_A and f_B. Show, algebraically, that the two functions are equivalent.

   Solution

$$\begin{aligned} f_A &= \bar{x}_1\bar{x}_3 + x_2x_3 + x_1\bar{x}_2 \\ &= \bar{x}_1\bar{x}_3 + x_2x_3 + x_1\bar{x}_2 + \bar{x}_2\bar{x}_3 + \bar{x}_1x_2 + x_1x_3 \\ &= \bar{x}_2\bar{x}_3 + \bar{x}_1x_2 + x_1x_3 \\ &= f_B \end{aligned}$$

3. **Implicit Techniques** (*10 points*)

   Explain how, conceptually, you can create a new circuit that will help you with the task of proving that Circuits A and B are equivalent. Specifically, how can you tie the outputs of the circuits together to create a new circuit that computes an output that is identically 0 if and only if the two circuits compute the same Boolean function? Indicate what logical function should replace the question mark in Figure 2. Call the function implemented by the new circuit g.

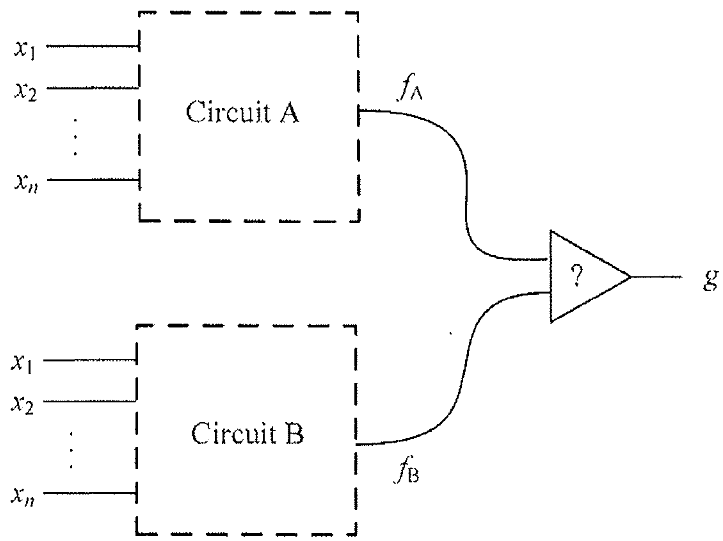


Figure 2: A conceptual circuit for verifying whether Circuits A and B are equivalent.

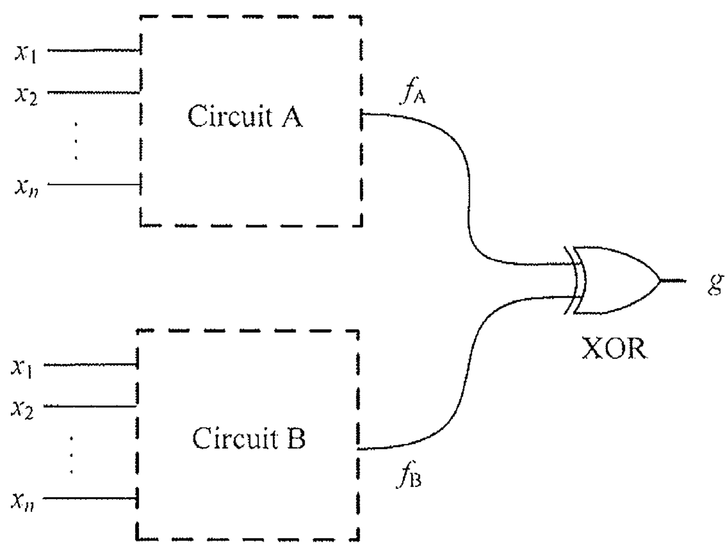


Figure 3: **Solution for Part 3.**

4. **Binary Decision Diagrams** (*25 points*)

First proposed in 1959 by Lee, binary decision diagrams (BDDs) were popularized in 1986 through a seminal paper by Bryant. A BDD consists of a directed graph in which nodes either have associated input variables or else are designated as a constant nodes ("0" or "1"). To evaluate a function one begins at a designated source node and follows a path dictated by the values of the variables until one arrives at one of the two constant nodes. The value of this constant node specifies the value of the function. An example is shown in Figure 5. The BDD in the figure represents the function

$$f = x_1(x_2 + x_3).$$

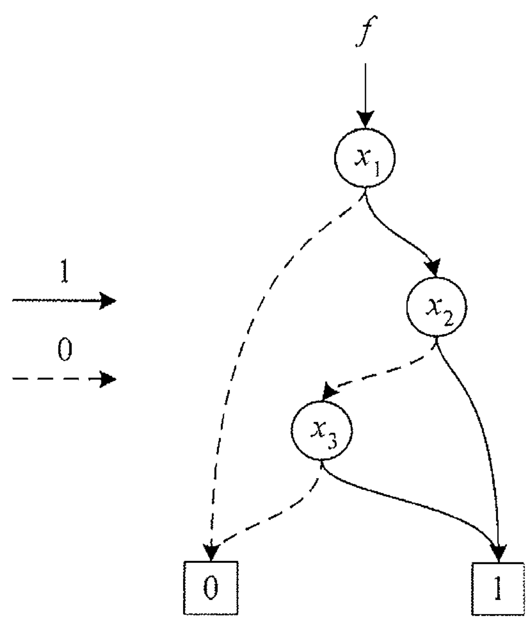


Figure 4: A binary decision diagram (BDD).

Although comparable in size to a truth table in the worst case, BDDs are surprisingly compact for many of the Boolean functions encountered in practice. This is due to the fact that BDDs can often be reduced in size by *collapsing* redundant nodes and *merging* equivalent nodes.

Draw reduced BDDs for the functions f_A, f_B and g in Part 3.

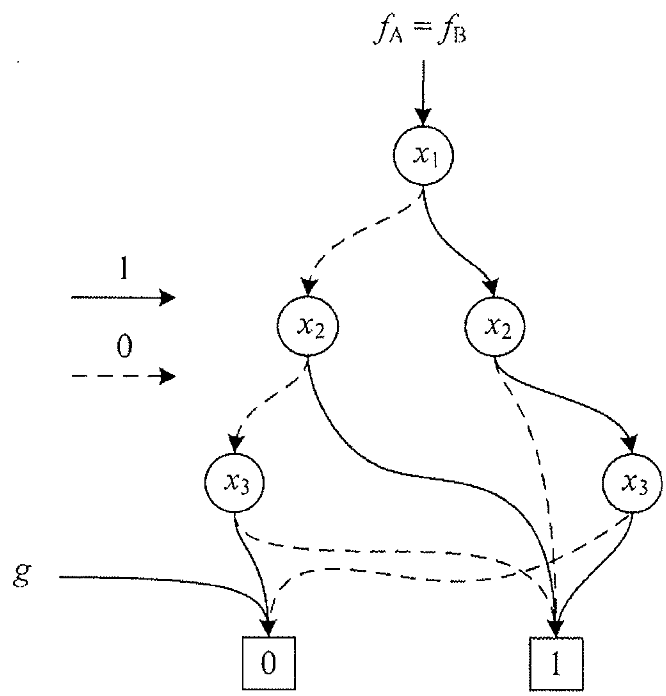


Figure 5: **Solution for Part 4.**

5. **Boolean Satisfiability** (*25 points*)

In our verification problem, the question that we are trying to answer – namely, whether the circuits are equivalent – has an affirmative answer if equivalence holds for *all* possible input assignments. It has a negative answer if equivalence does not hold for *any* input assignment. So-called SAT-based techniques, based on heuristic solutions to the Boolean satisfiability problem, can be used to answer questions that fit this mold. In theory, such algorithms can take time that is exponential in the number of variables to complete. In practice, they have shown themselves to be remarkably efficient.

SAT-based analysis begins with a circuit structure and proceeds by packaging the Boolean function that it computes in conjunctive normal form (CNF). This is passed to heuristic algorithms known as SAT solvers. If the solver returns "UNSAT," this means that there is no satisfying assignment to the formula. Otherwise, the solver returns "SAT" along with a satisfying assignment. For instance, consider the circuit in Figure 6. The corresponding CNF formula is:

$$(\bar{x}_2 + y)(\bar{x}_3 + y)(x_2 + x_3 + \bar{y})(x_1 + \bar{h})(y + \bar{h})(\bar{x}_1 + \bar{y} + h)(h).$$

The solver would return SAT. A satisfying assignment for this formula is $x_1 = x_2 = x_3 = y = h = 1$.

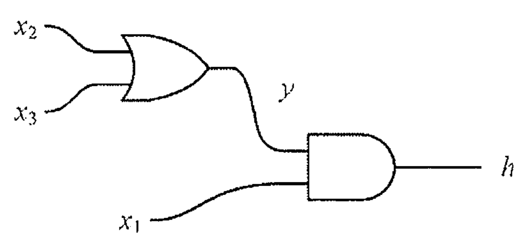


Figure 6: A circuit illustrating Boolean satisfiability.

Based on the construct in Figure 2, write a CNF formula for the question: are Circuits A and B equivalent?

Solution

Label gate outputs y_1, y_2, y_3, and z_1, z_2, z_3, as shown in Figure 7. Define

$$
\begin{aligned}
c_1 &= (\bar{x}_1 + \bar{y}_1)(\bar{x}_3 + \bar{y}_1)(x_1 + x_3 + y_1) \\
c_2 &= (x_2 + \bar{y}_2)(x_3 + \bar{y}_2)(\bar{x}_2 + \bar{x}_3 + y_2) \\
c_3 &= (x_1 + \bar{y}_3)(\bar{x}_2 + \bar{y}_3)(x_1 + \bar{x}_2 + y_3) \\
c &= (\bar{y}_1 + f_A)(\bar{y}_2 + f_A)(\bar{y}_3 + f_A)(y_1 + y_2 + y_3 + \bar{f}_A) \\
d_1 &= (\bar{x}_2 + \bar{z}_1)(\bar{x}_3 + \bar{z}_1)(x_2 + x_3 + z_1) \\
d_2 &= (\bar{x}_1 + \bar{z}_2)(x_2 + \bar{z}_2)(x_1 + x_2 + z_2) \\
d_3 &= (x_1 + \bar{z}_3)(x_3 + \bar{z}_3)(\bar{x}_1 + \bar{x}_3 + z_3) \\
d &= (\bar{z}_1 + f_B)(\bar{z}_2 + f_B)(\bar{z}_3 + f_B)(z_1 + z_2 + z_3 + \bar{f}_b) \\
e &= (\bar{f}_A + f_B + g)(f_A + \bar{f}_B + g)(f_A + f_B + \bar{g})(\bar{f}_A + \bar{f}_B + \bar{g})
\end{aligned}
$$

The CNF formula is

$$(c_1)\,(c_2)\,(c_3)\,(c)\,(d_1)\,(d_2)\,(d_3)\,(d)\,(e)\,(g)$$

You can verify that it is UNSAT.

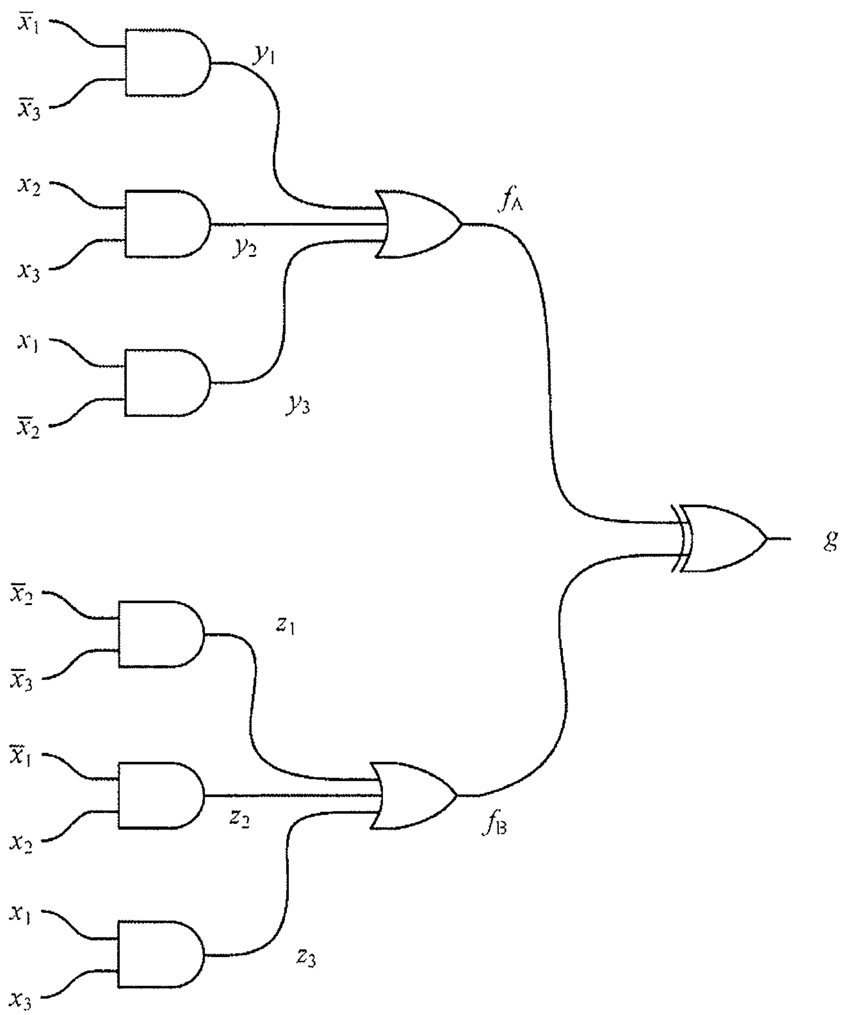


Figure 7: **Circuit illustrating solution for Part 5.**