CS203- Design of Digital System Laboratory

GroupWise Mini Project Assignments

Submission Deadline: December 12, 2021.

A stuck-at fault (SAF) is a logic-level fault that mimics a manufacturing defect on a digital device, circuits, etc. An SAF is two types: stuck-at-0 (SA0) and stuck-at-1 (SA1) fault. Individual input bit or signal on a wire of a logic gate or circuit is assumed to be stuck at Logical '1', or '0' if the input bit (signal) is converted to logic-1 or logic-0 irrespective of its previous or assigned value.

Basic Instructions:

- Each project group needs to design, synthesize a testable circuit for a fault as asked in a problem assigned to the group.
- Each problem number represents a mini-project assignment for the corresponding project group, i.e. Problem-X is assigned to the project Group-X. For example, Probmel-1 is assigned to Group-1, Problem-2 is assigned to Group-2, and so on.
- Implementation Tools: Xilinx or similar HDL simulator.
- Report preparation: Basic HTML Environment.
- Implementation and Presentation Weightage: 20%
- Report Weightage: 20%
- If needed, refer a text book, e.g. Digital Design by Morris Mano, 5th Edn., for theory of the logic circuits considered in the problems.
- Other information, if any, will be communicated whenever necessary.

Problem-1: Detection of both SA0 and SA1 faults on basic logic gates: AND, OR, and NOT.

Problem-2: Detection of both SA0 and SA1 faults on basic universal logic gates: NAND, and NOR.

Problem-3: Detection of both SA0 and SA1 faults on 2–input XOR and XNOR logic gates.

Problem-4: Detection of both SA0 and SA1 faults on the logic circuit (shown below) that represents a 3–input ODD function.

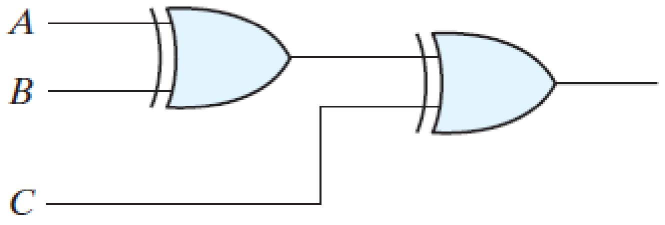


Figure: Logic diagram of a 3-input odd function.

Problem-5: Detection of both SA0 and SA1 faults on the logic circuit (shown below) that represents a 3–input EVEN function.

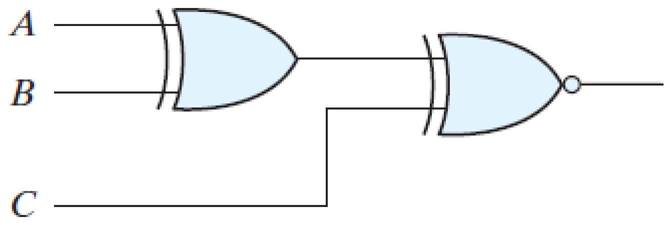


Figure: Logic diagram of a 3-input even function.

Problem-6: Detection of both SA0 and SA1 faults on a 3-bit EVEN parity generator circuit (shown below).

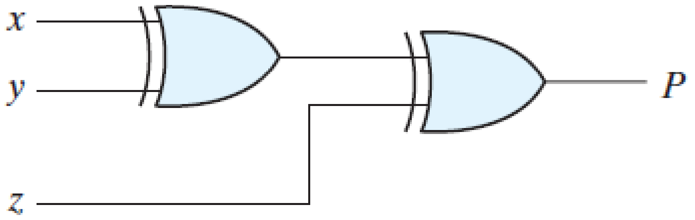


Figure: Logic diagram of a 3-bit even parity generator.

Problem-7: Detection of both SA0 and SA1 faults on a 4-bit EVEN parity checker circuit (shown below).

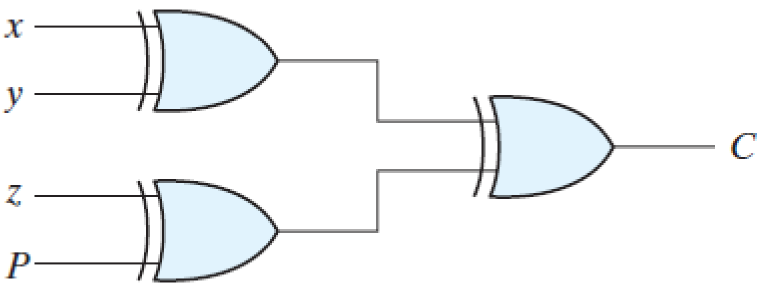


Figure: Logic diagram of a 4-bit even parity checker.

Problem-8: Detection of both SA0 and SA1 faults on a HALF-ADDER circuit (shown below) implemented by AND, OR gates.

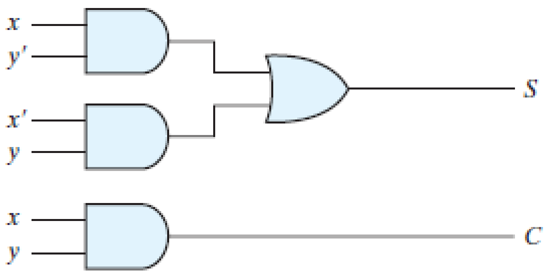


Figure: Logic diagram of a half-adder.

Problem-9: Detection of both SA0 and SA1 faults on a HALF-ADDER circuit (shown below) implemented by AND, XOR gates.

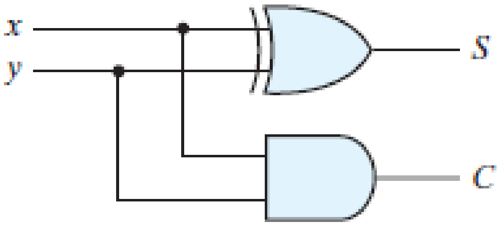


Figure: Logic diagram of a half-adder.

Problem-10: Detection of both SA0 and SA1 faults on a HALF-SUBTRACTOR circuit (shown below).

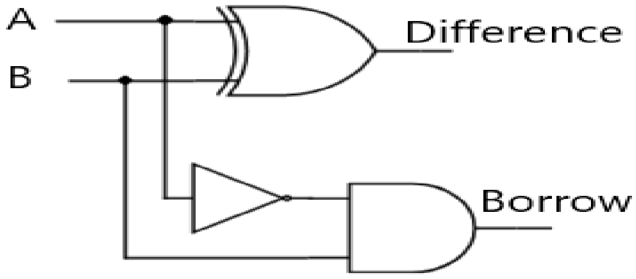


Figure: Logic diagram of a half subtractor.

Problem-11: Detection of both SA0 and SA1 faults on the logic circuit (shown below) that represents the SUM function of a Full-Adder.

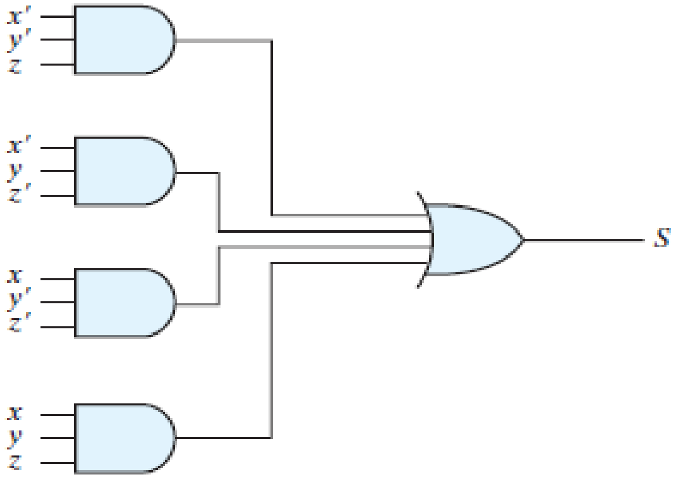


Figure: Logic diagram for the SUM function of a full adder.

Problem-12: Detection of both SA0 and SA1 faults on the logic circuit (shown below) that represents the CARRY function of a Full-Adder.

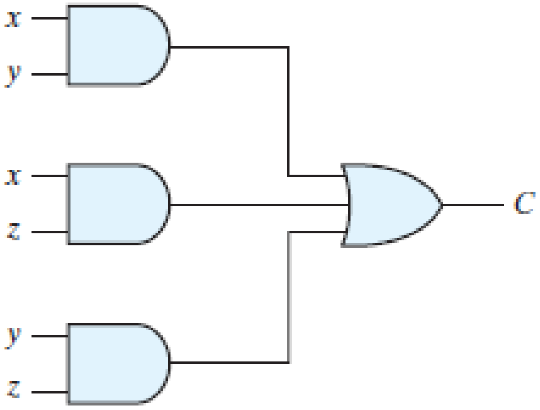


Figure: Logic diagram for the CARRY function of a full adder.

Problem-13: Detection of SA0 faults on the logic circuit (shown below) that represents the a Full-Subtractor.

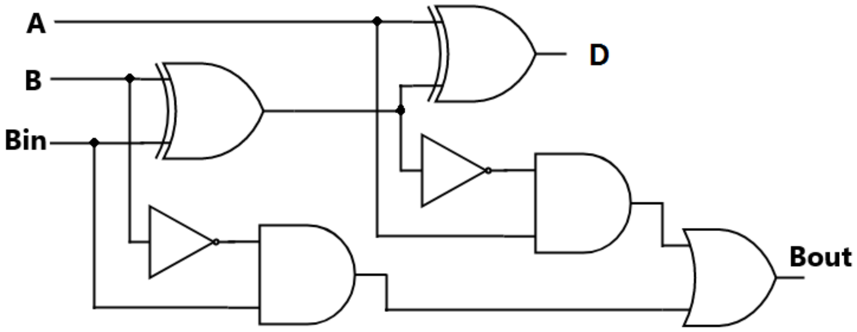


Figure: Logic diagram of full subtractor.

Problem-14: Detection of SA1 faults on the logic circuit (shown below) that represents the a Full-Subtractor.

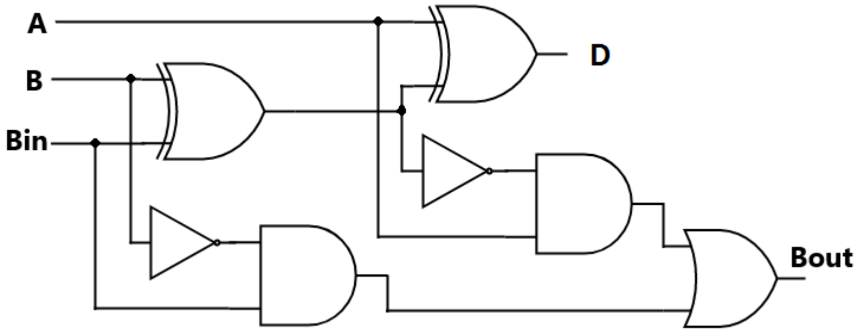


Figure: Logic diagram of full subtractor.

Problem-15: Detection of SA0 faults on selected I/O wires of a BCD-to-excess-3 code converter (shown below). Detect the fault for each output function.

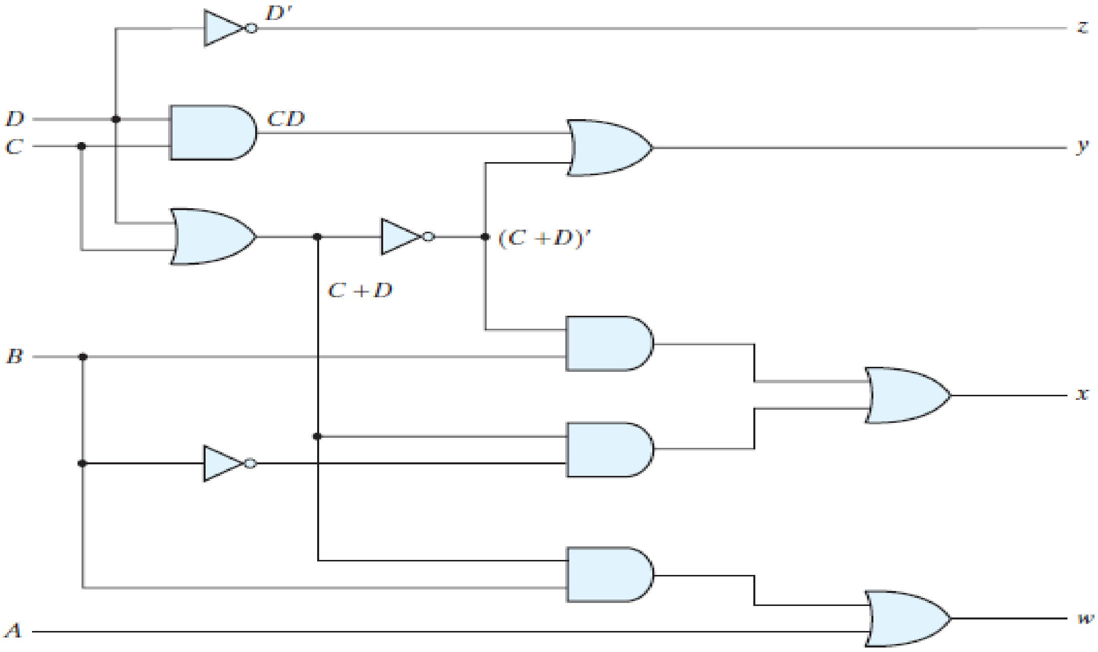


Figure: Logic diagram for BCD-to-excess-3 code converter.

Problem-16: Detection of SA1 faults on selected I/O wires of a BCD-to-excess-3 code converter (shown below). Detect the fault for each output function.

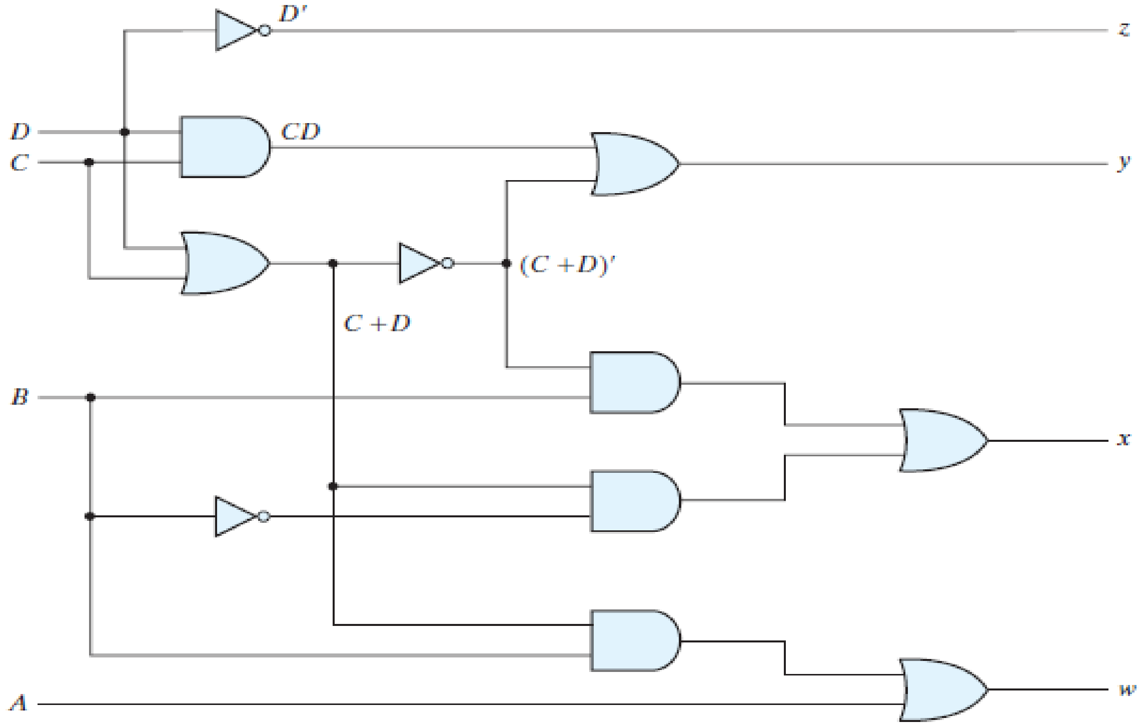


Figure: Logic diagram for BCD-to-excess-3 code converter.

Problem-17: Detection of the SA0 faults on the 4x16 decoder constructed with two 3x8 decoders (See Figure below).

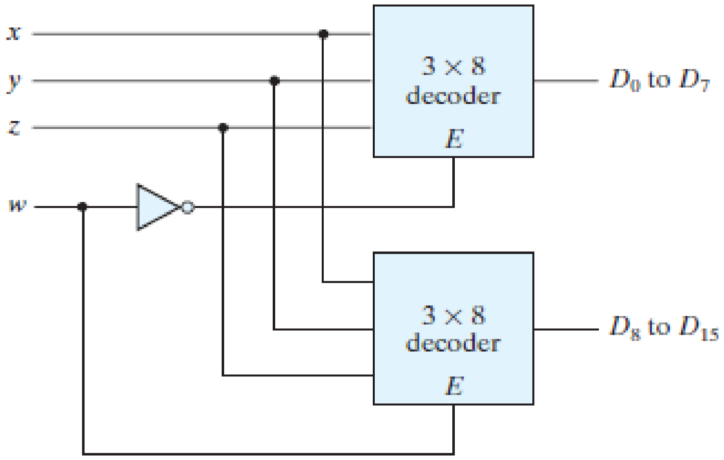


Figure: Logic diagram of a 4x16 decoder.

Problem-18: Detect the SA1 faults on the 4x16 decoder constructed with two 3x8 decoders (See Figure below).

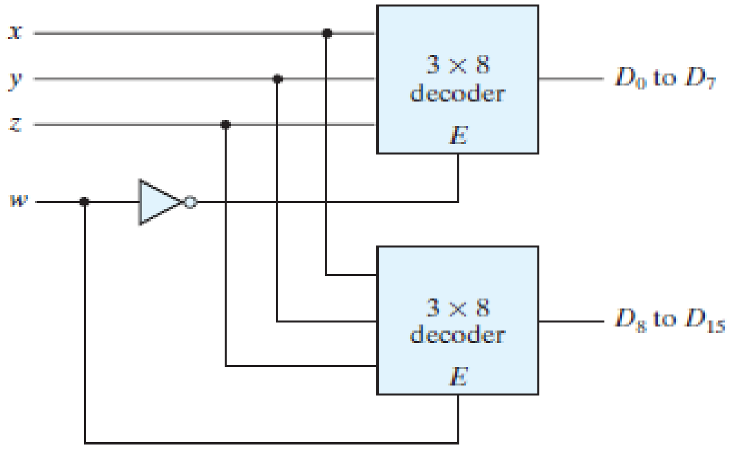
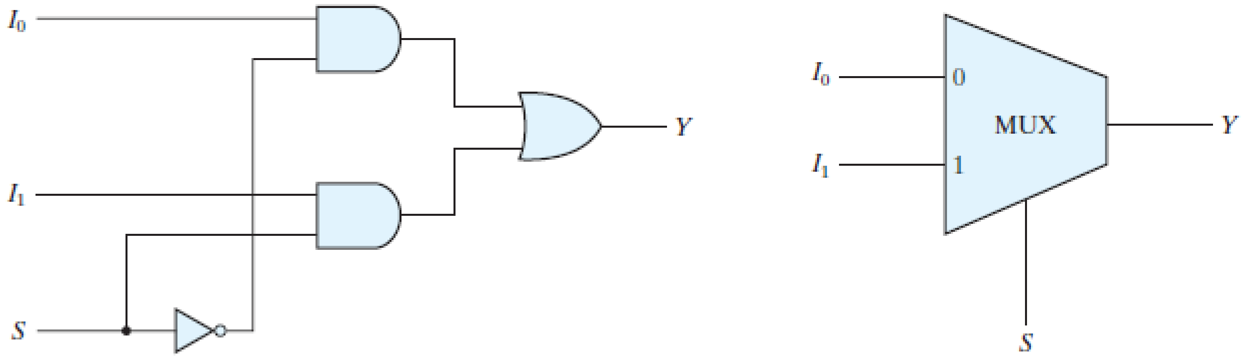


Figure: Logic diagram of a 4x16 decoder.

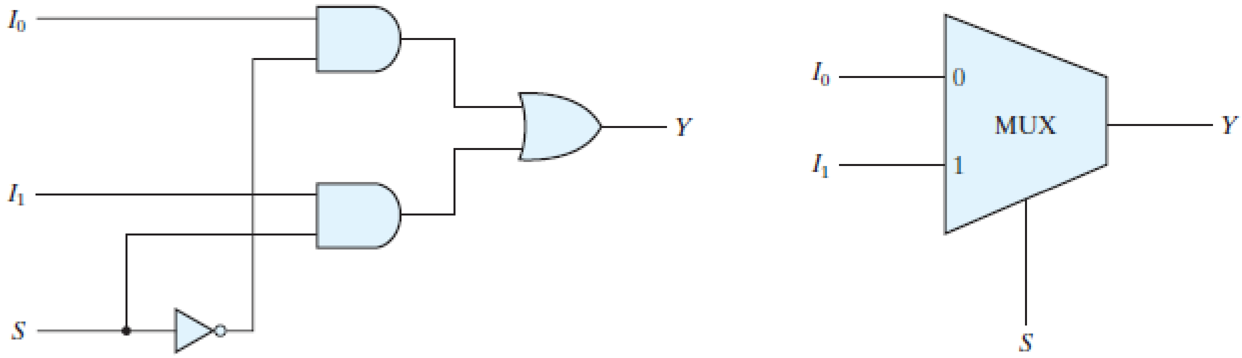
Problem-19: Detection of SA0 faults on a 2x1 multiplexer (mux). Logic and block diagram of a 2x1 mux are shown below.



(a) Logic diagram

(b) Block diagram

Problem-20: Detection of SA1 faults on a 2x1 multiplexer (mux). Logic and block diagram of a 2x1 mux are shown below.



(a) Logic diagram

(b) Block diagram

Problem-21: Detection of SA0 faults on a 4-input Priority Encoder circuit shown below.

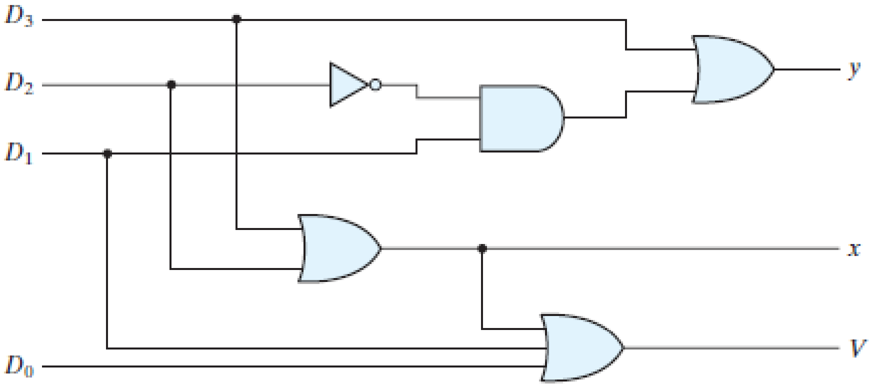


Figure: Logic diagram of a Four-input priority encoder.

Problem-22: Detection of SA1 faults on a 4-input Priority Encoder circuit shown below.

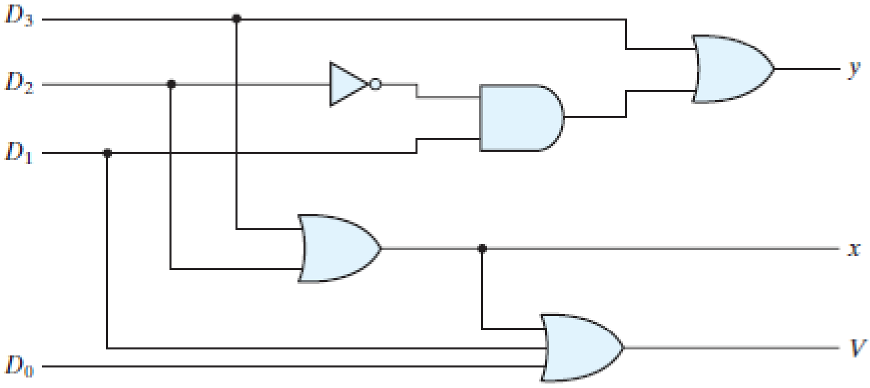


Figure: Logic diagram of a Four-input priority encoder.

Problem-23: Detection of both SA0 and SA1 faults on the SR latch implemented with NOR gates as shown below.

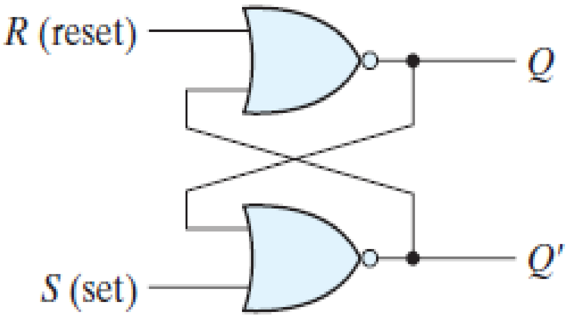


Figure: Logic diagram of an SR latch.

Problem-24: Detection of SA0 faults on the D latch as shown below.

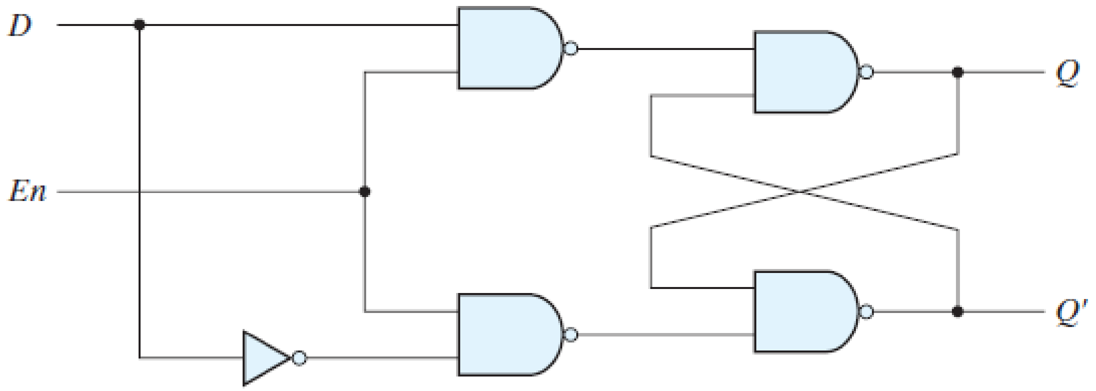


Figure: Logic diagram of a D latch.

Problem-25: Detection of SA1 faults on the D latch as shown below.

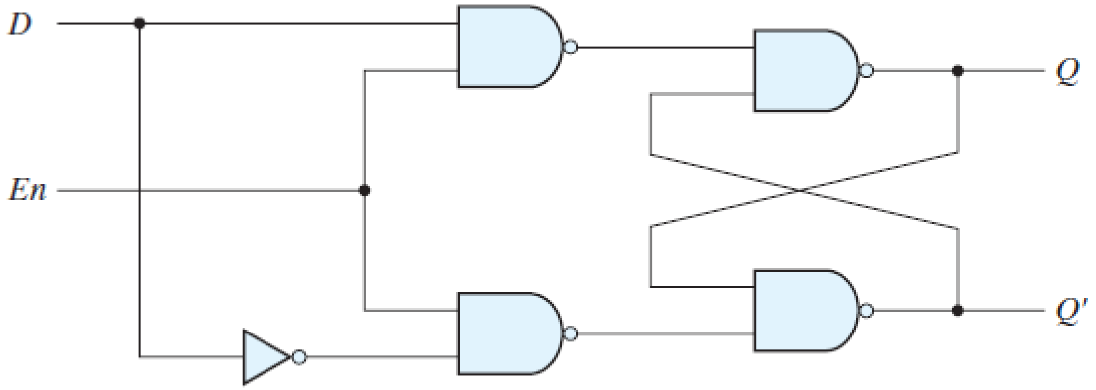


Figure: Logic diagram of a D latch.

Problem-26: Detection of SA0 faults on the JK Flip-Flop as shown below.

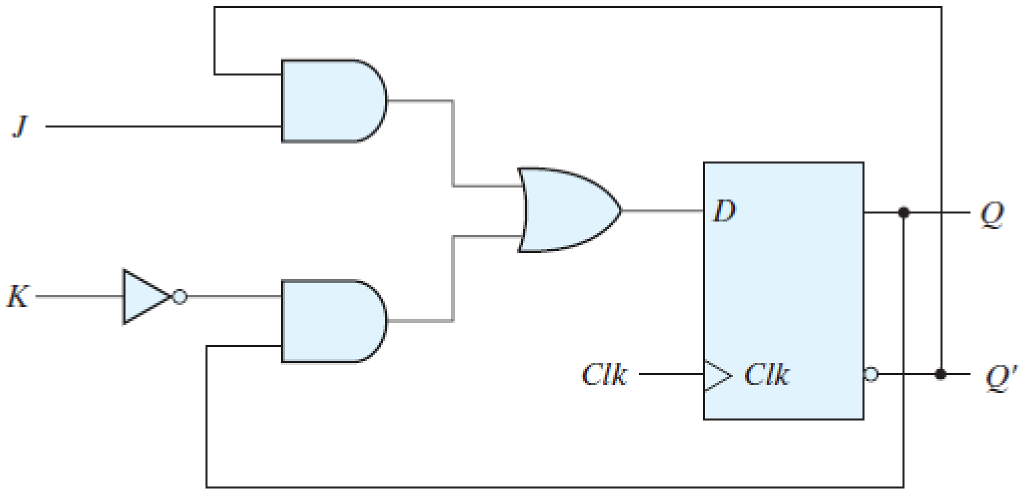


Figure: Logic diagram of a JK Flip-Flop.

Problem-27: Detection of SA1 faults on the JK Flip-Flop as shown below.

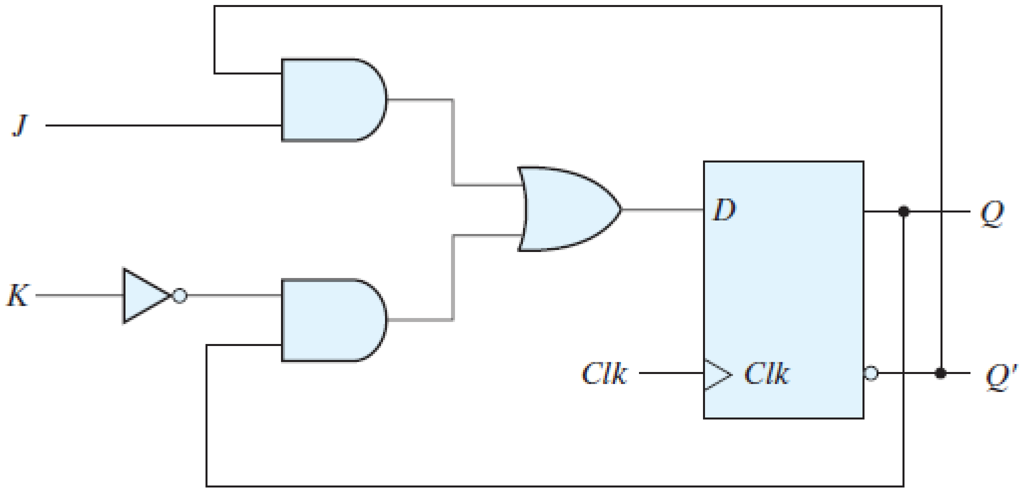


Figure: Logic diagram of a JK Flip-Flop.

Problem-28: Detection of SA0 faults on the 4-bit SHIFT REGISTER as shown below.

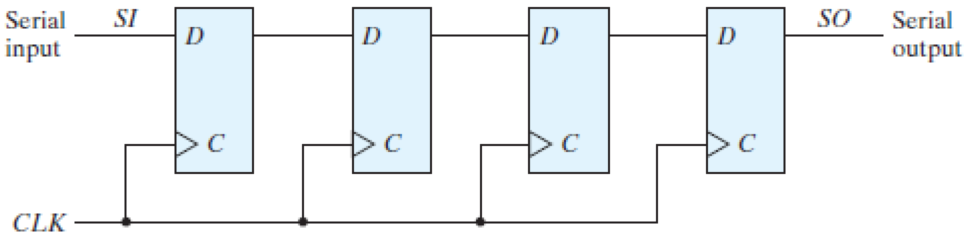


Figure: Logic diagram of a 4-bit shift register.

Problem-29: Detection of SA1 faults on the 4-bit SHIFT REGISTER as shown below.

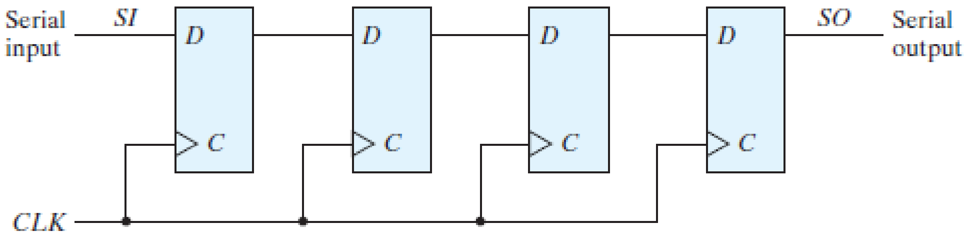


Figure: Logic diagram of a 4-bit shift register.
