Design of Fault Tolerance in Spacecraft Using Logisim Circuit

Using digital logic concepts, this design simulates a fault-tolerant control system for spacecraft subsystems. Each level, representing groups of spacecraft subsystems, is structured to ensure fault detection, recovery, and progression through a multi-level system.

Error Detection and Prioritization:

- Each level receives a 5-bit input representing the current state of subsystems and compares it with the **expected 5-bit output**.
- For **Level 1**, the expected 5-bit output is constant. For subsequent levels, the expected output is obtained by **left-shifting** the expected output from the previous level.
- If the input bits do not match the expected output, a fault is detected.
    - For **crucial subsystems**, the system attempts **automatic recovery**, as these subsystems are critical for the mission. However, note that the **first bit** of each level is uncorrectable, while the next three bits are correctable.
    - For **non-crucial subsystems**, the system relies on **user input** via small switches to decide whether or not to attempt recovery. In this context, recovery means:
        - For **crucial subsystems**: Restoring the bit to `1` (working).
        - For **non-crucial subsystems**: Setting the bit to `0` (non-working).
    - **Note:** The meaning of recovery differs for crucial and non-crucial subsystems.

Fault Recovery Logic:

- The circuit uses a combination of **XOR** and **AND gates** to compare the input bits to the expected output at each level.
- **Automatic recovery** is applied to crucial subsystems whenever possible, while user decisions (through switches) govern recovery for non-crucial subsystems.
- The system can only proceed to the next level if the crucial subsystem bits match the expected values after recovery.
- **Multiplexers (MUX)** are used to switch between faulty and corrected bits based on the system's recovery algorithm, allowing the system to adjust the state of subsystems based on the user’s decision and recovery rules.

Subsystem Classification:

- Subsystems are classified into **crucial** and **non-crucial** categories:
    - **Crucial subsystems** are prioritized for correction. If any crucial subsystem remains faulty after recovery attempts, the **mission is terminated**.
    - **Non-crucial subsystems** can be left unrecovered based on user preference, as they are not vital for mission success.

State Monitoring:

- **LED indicators** provide real-time feedback on the status of each subsystem:
    - LEDs show whether subsystems are functioning (`1`) or faulty (`0`).
    - Different LEDs signal the status of recovery, mission progression, and completion.
- If **Level 1** fails to meet the expected conditions (i.e., any crucial subsystem cannot be recovered), the mission terminates immediately.
- If all levels pass successfully, the **Mission Success LED** lights up to confirm that the mission has been completed without critical failures.

Summary of Fault Tolerance Mechanism:

- The design ensures that only **successful recovery** of all **crucial subsystems** leads to mission success.
- If any crucial subsystem remains faulty, the mission fails, and the system terminates.
- This fault tolerance mechanism leverages automatic recovery for critical subsystems and user input for non-crucial ones, providing a robust, flexible method for handling errors in spacecraft systems.