WORKING

System Architecture:

• *Level 1:Base Layer*

– Subsystems: Advanced Sensors, Data Acquisition and Analysis, Advanced Materials, Device Failure Physics, Thermal Regulation.

– The system will first focus on diagnosing and recovering these subsystems to ensure the spacecraft's baseline functionality. One subsystem in this level is crucial but uncorrectable; if this subsystem fails, the mission will terminate immediately. Passing Level 1 is mandatory for the mission to proceed to the next level.

• *Level 2: Functional Module Implementation Layer*

– Subsystems: Power Status Monitoring, Attitude Control, Load Monitoring, Measure- ment Control, Communications Integrity.
– If Level 1 is passed, this layer ensures that subsystems crucial for spacecraft stability, control, and communication are fully operational. One subsystem in this layer is uncorrectable, and if it fails, the mission will terminate.

• *Level 3: Task Goal Implementation Layer*

– Subsystems: Fault Detection, Fault Isolation, Fault Prediction, Health Assessment, Repair Planning.

– This final level assesses and ensures fault detection and health management systems are functioning. The same condition applies—one uncorrectable subsystem must pass, and failure will lead to mission termination.

• *Non-Critical Subsystems:*These subsystems, although not essential for the core space- craft operations, can be monitored and toggled based on the user's discretion. In an ideal operational scenario, all non-crucial subsystems are turned off to conserve resources. Fail- ures in non-crucial subsystems are tolerated.

Mission Termination Conditions:

- If any crucial subsystem, particularly the uncorrectable ones in each level, fails to recover, the mission will terminate immediately.
- Progression to the next level occurs only if all crucial subsystems in the current level are functional. Failure to pass any level results in mission termination.
- For the mission to succeed, all levels must be passed.

Truth Tables:

LEVEL1:

Reason for "!!"

In level 1, since the first bit (R1) is 0, the enable signal for R2 also becomes 0, causing the multiplexer (MUX) for R2 not to function, and it outputs an indeterminate signal (x). This "x" signal is mixed, meaning it could be either 0 or 1, which leads to uncertainty and triggers an error (indicated by a red line). As a result, the MUX for the third bit will also fail, and the same logic applies to the fourth bit. This issue carries forward to subsequent levels as well.Hence, mission fails.

Since the system checks each bit individually, any failure to correct bits to match the expected values will prevent further checks on crucial subsystems. Therefore, the process will halt if the corrected bits do not meet the expected conditions.

R1	R2	R3	R4	R5	O1	O2	O3	O4	O5
0	0	0	0	0	0	x	!!	!!	0
0	0	0	0	1	0	x	!!	!!	1
0	0	0	1	0	0	x	!!	!!	0
0	0	0	1	1	0	x	!!	!!	1
0	0	1	0	0	0	x	!!	!!	0
0	0	1	0	1	0	x	!!	!!	1
0	0	1	1	0	0	x	!!	!!	0
0	0	1	1	1	0	x	!!	!!	1
0	1	0	0	0	0	x	!!	!!	0
0	1	0	0	1	0	x	!!	!!	1
0	1	0	1	0	0	x	!!	!!	0
0	1	0	1	1	0	x	!!	!!	1
0	1	1	0	0	0	x	!!	!!	0
0	1	1	0	1	0	x	!!	!!	1
0	1	1	1	0	0	x	!!	!!	0
0	1	1	1	1	0	x	!!	!!	1
1	0	0	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1	1	1
1	0	0	1	0	1	1	1	1	0
1	0	0	1	1	1	1	1	1	1
1	0	1	0	0	1	1	1	1	0
1	0	1	0	1	1	1	1	1	1
1	0	1	1	0	1	1	1	1	0
1	0	1	1	1	1	1	1	1	1
1	1	0	0	0	1	1	1	1	0
1	1	0	0	1	1	1	1	1	1
1	1	0	1	0	1	1	1	1	0
1	1	0	1	1	1	1	1	1	1
1	1	1	0	0	1	1	1	1	0
1	1	1	0	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1

LEVEL 2:

Reason for "!!"

Unlike in Level 1, in Level 2, we give the user the option to decide whether to switch off the non-crucial subsystem (R4). However, this user interaction is only possible if the outputs O1, O2, and O3 match the expected bits. This ensures that only after verifying the crucial subsystems can the user be asked about

the non-crucial subsystem's status.

R1	R2	R3	R4	R5	SWITCH1	O1	O2	O3	O4	O5
0	0	0	0	0	0	0	x	!!	x	0
0	0	0	0	0	1	0	x	!!	0	0
0	0	0	0	1	0	0	x	!!	x	1
0	0	0	0	1	1	0	x	!!	0	1
0	0	0	1	0	0	0	x	!!	x	0
0	0	0	1	0	1	0	x	!!	!!	0
0	0	0	1	1	0	0	x	!!	x	1
0	0	0	1	1	1	0	x	!!	!!	1
0	0	1	0	0	0	0	x	!!	x	0
0	0	1	0	0	1	0	x	!!	0	0
0	0	1	0	1	0	0	x	!!	x	1
0	0	1	0	1	1	0	x	!!	0	1
0	0	1	1	0	0	0	x	!!	x	0
0	0	1	1	0	1	0	x	!!	!!	0
0	0	1	1	1	0	0	x	!!	x	1
0	0	1	1	1	1	0	x	!!	!!	1
0	1	0	0	0	0	0	x	!!	x	0
0	1	0	0	0	1	0	x	!!	0	0
0	1	0	0	1	0	0	x	!!	x	1
0	1	0	0	1	1	0	x	!!	0	1
0	1	0	1	0	0	0	x	!!	x	0
0	1	0	1	0	1	0	x	!!	!!	0
0	1	0	1	1	0	0	x	!!	x	1
0	1	0	1	1	1	0	x	!!	!!	1
0	1	1	0	0	0	0	x	!!	x	0
0	1	1	0	0	1	0	x	!!	0	0
0	1	1	0	1	0	0	x	!!	x	1
0	1	1	0	1	1	0	x	!!	0	1
0	1	1	1	0	0	0	x	!!	x	0
0	1	1	1	0	1	0	x	!!	!!	0
0	1	1	1	1	0	0	x	!!	x	1
0	1	1	1	1	1	0	x	!!	!!	1
1	0	0	0	0	0	1	1	1	x	0
1	0	0	0	0	1	1	1	1	0	0
1	0	0	0	1	0	1	1	1	x	1
1	0	0	0	1	1	1	1	1	0	1
1	0	0	1	0	0	1	1	1	x	0
1	0	0	1	0	1	1	1	1	0	0
1	0	0	1	1	0	1	1	1	x	1
1	0	0	1	1	1	1	1	1	0	1
1	0	1	0	0	0	1	1	1	x	0
1	0	1	0	0	1	1	1	1	0	0
1	0	1	0	1	0	1	1	1	x	1
1	0	1	0	1	1	1	1	1	0	1
1	0	1	1	0	0	1	1	1	x	0
1	0	1	1	0	1	1	1	1	0	0
1	0	1	1	1	0	1	1	1	x	1
1	0	1	1	1	1	1	1	1	0	1
1	1	0	0	0	0	1	1	1	x	0
1	1	0	0	0	1	1	1	1	0	0
1	1	0	0	1	0	1	1	1	x	1
1	1	0	0	1	1	1	1	1	0	1
1	1	0	1	0	0	1	1	1	x	0
1	1	0	1	0	1	1	1	1	0	0
1	1	0	1	1	0	1	1	1	x	1

LEVEL 3:

Reason for "!!"

In Level 3, there are two non-crucial subsystems (R3 and R4), so the user is asked whether they want to switch off both non-crucial subsystems. The interesting point here is that if the user denies switching off R3 (via switch1) but permits switching off R4 (via switch2), the output O4 still won't change. This is because O4 depends on R3, and since the system checks each bit individually, the behavior of R4 alone won't override the decision for R3.

R1	R2	R3	R4	R5	SWITCH2	SWITCH1	O1	O2	O3	O4	O5
0	0	0	0	0	0	0	0	x	x	x	0
0	0	0	0	0	0	1	0	x	x	0	0
0	0	0	0	0	1	0	0	x	0	x	0
0	0	0	0	0	1	1	0	x	0	0	0
0	0	0	0	1	0	0	0	x	x	x	1
0	0	0	0	1	0	1	0	x	x	0	1
0	0	0	0	1	1	0	0	x	0	x	1
0	0	0	0	1	1	1	0	x	0	0	1
0	0	0	1	0	0	0	0	x	x	x	0
0	0	0	1	0	0	1	0	x	x	!!	0
0	0	0	1	0	1	0	0	x	0	x	0
0	0	0	1	0	1	1	0	x	0	0	0
0	0	0	1	1	0	0	0	x	x	x	1
0	0	0	1	1	0	1	0	x	x	!!	1
0	0	0	1	1	1	0	0	x	0	x	1
0	0	0	1	1	1	1	0	x	0	0	1
0	0	1	0	0	0	0	0	x	x	x	0
0	0	1	0	0	0	1	0	x	x	0	0
0	0	1	0	0	1	0	0	x	!!	x	0
0	0	1	0	0	1	1	0	x	!!	0	0
0	0	1	0	1	0	0	0	x	x	x	1
0	0	1	0	1	0	1	0	x	x	0	1
0	0	1	0	1	1	0	0	x	!!	x	1
0	0	1	0	1	1	1	0	x	!!	0	1
0	0	1	1	0	0	0	0	x	x	x	0
0	0	1	1	0	0	1	0	x	x	!!	0
0	0	1	1	0	1	0	0	x	!!	x	0
0	0	1	1	0	1	1	0	x	!!	!!	0
0	0	1	1	1	0	0	0	x	x	x	1
0	0	1	1	1	0	1	0	x	x	!!	1
0	0	1	1	1	1	0	0	x	!!	x	1
0	0	1	1	1	1	1	0	x	!!	!!	1
0	1	0	0	0	0	0	0	x	x	x	0
0	1	0	0	0	0	1	0	x	x	0	0
0	1	0	0	0	1	0	0	x	0	x	0
0	1	0	0	0	1	1	0	x	0	0	0
0	1	0	0	1	0	0	0	x	x	x	1
0	1	0	0	1	0	1	0	x	x	0	1
0	1	0	0	1	1	0	0	x	0	x	1
0	1	0	0	1	1	1	0	x	0	0	1
0	1	0	1	0	0	0	0	x	x	x	0
0	1	0	1	0	0	1	0	x	x	!!	0
0	1	0	1	0	1	0	0	x	0	x	0
0	1	0	1	0	1	1	0	x	0	0	0
0	1	0	1	1	0	0	0	x	x	x	1
0	1	0	1	1	0	1	0	x	x	!!	1
0	1	0	1	1	1	0	0	x	0	x	1
0	1	0	1	1	1	1	0	x	0	0	1
0	1	1	0	0	0	0	0	x	x	x	0

1	1	0	0	1	1	1	1	1	0	0	1
1	1	0	1	0	0	0	1	1	x	x	0
1	1	0	1	0	0	1	1	1	x	!!	0
1	1	0	1	0	1	0	1	1	0	x	0
1	1	0	1	0	1	1	1	1	0	0	0
1	1	0	1	1	0	0	1	1	x	x	1
1	1	0	1	1	0	1	1	1	x	!!	1
1	1	0	1	1	1	0	1	1	0	x	1
1	1	0	1	1	1	1	1	1	0	0	1
1	1	1	0	0	0	0	1	1	x	x	0
1	1	1	0	0	0	1	1	1	x	0	0
1	1	1	0	0	1	0	1	1	0	x	0
1	1	1	0	0	1	1	1	1	0	0	0
1	1	1	0	1	0	0	1	1	x	x	1
1	1	1	0	1	0	1	1	1	x	0	1
1	1	1	0	1	1	0	1	1	0	x	1
1	1	1	0	1	1	1	1	1	0	0	1
1	1	1	1	0	0	0	1	1	x	x	0
1	1	1	1	0	0	1	1	1	x	!!	0
1	1	1	1	0	1	0	1	1	0	x	0
1	1	1	1	0	1	1	1	1	0	0	0
1	1	1	1	1	0	0	1	1	x	x	1
1	1	1	1	1	0	1	1	1	x	!!	1
1	1	1	1	1	1	0	1	1	0	x	1
1	1	1	1	1	1	1	1	1	0	0	1

Note: R5 is a non-essential subsystem and operates independently of the other bits. Therefore, its status does not affect the outcomes or dependencies related to the other subsystems.

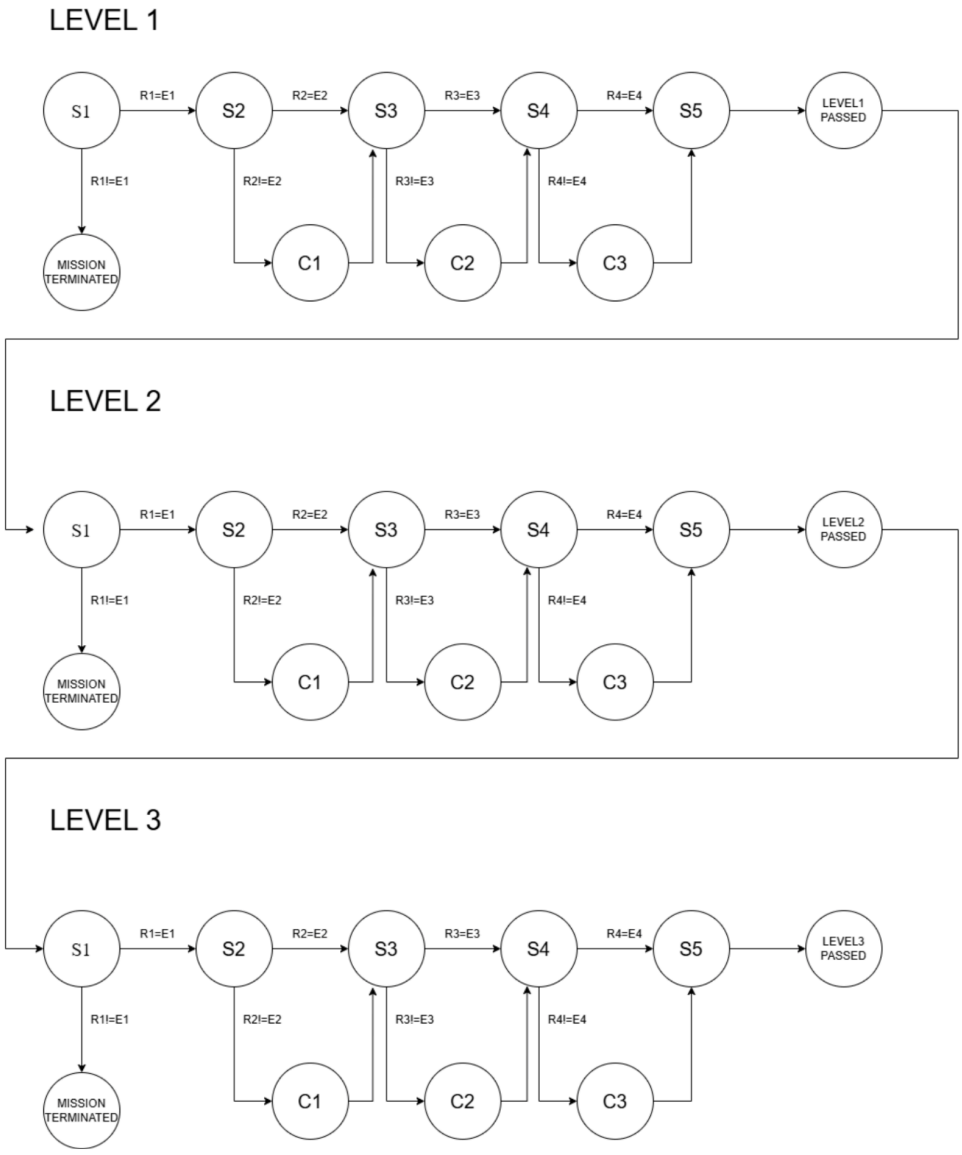
MULTIPLEXER(2 X 1)

I1	S0	I0	E	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

E: Enable
S0: Select line
I0 : Input 0
I1 : Input 1
Y : Output

STATE DIAGRAM:

S:Before correction
C:After correction



STATE EQUATIONS:

– Multiplexer:

Y = E.(S0'.I0+S0.I1)

– Level 1:

Level1_passed = O1.O2.O3.O4

– Level 2:

Level2_passed = O1.O2.O3

– Level 3:

Level3_passed = O1.O2