Self-Healing Power Management Unit for IoT Enabled Agricultural Systems

by

Vivek Aswani
202011014

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

May, 2022

Declaration

I hereby declare that

(i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at DA-IICT and has not been submitted elsewhere for a degree,

(ii) due acknowledgement has been made in the text to all the reference material used.

Vivek

Signature of Student

Certificate

This is to certify that the thesis work entitled Self-healing power management unit for IoT enabled agricultural systems has been carried out by Vivek Aswani (202011014) for the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology under my/our supervision.

05 July 2022

Prof. Vinay S Palaparthy
Thesis Supervisor

Acknowledgements

First of all, I would like to extend my regards and gratitude to Prof. Vinay S Palaparthy for giving me the opportunity to work under him and for providing me with constant guidance and freedom of work. His liberal attitude provided me with a conducive environment to work and implement my ideas effectively.

I thank DAIICT for providing industry-level tools, which proved to be the key to my thesis work.

I owe my deep gratitude to my parents and family for their constant support and belief in me.

Contents

Abstract

Self-healing denotes the ability of a system to detect failures or faults and repair them and is progressively becoming a hopeful approach to designing dependable digital systems. Digitally advanced systems having planning for self-healing are likely to rectify errors and failures. Self-healing techniques that are currently being used face hardships such as scalability, reliability, mapping, and area overhead. This thesis explicates and examines the concept of self-healing of a power management unit for IoT-enabled agricultural systems and explores the self-healing methods related to digital design.

List of Principal Symbols and Acronyms

DC	Direct Current
DMR	Dual Modular Redundancy
LDO	Low Dropout Regulator
NoC	Network on Chip
PCB	Printed Circuit Board
SC	Signal Conditioning
SP	Signal Processing

List of Tables

List of Figures

Chapter 1
Introduction

1.1 General

1.1.1 Problem formulation

Hardware systems have become progressively intricate with the use of complex architectures on systems and powerful processors. Hardware systems like this may face failure in any part, which reduces their performance. Failure of hardware can happen anytime during the working of the system when it is performing tasks. Failures like this can arise because of influence from the adjacent environment (e.g., radiation, temperature, etc.) as well as the aging of the hardware. That is why self-healing is being proposed as a key with the potential of repairing or healing the system even without affecting the system' s performance. Hardware components can recover from or heal any harm to the system, in self-healing, from inside without requiring any outside involvement of humans.

The process variations have become more difficult as transistor technology has advanced. Circuit performance is affected by process variations that rapidly degrade technology when it is scaled. Analog and mixed-signal circuits are significantly affected due to the minuteness of such hardware designs. The size of the circuit and the compatibility of different distinct electrical components can critically change the noise features, operating frequency, etc. The study on self-healing hardware systems is accomplished by using actuators, sensors, and analog/digital loops of control which measure the system's unpredictability effects and move the system to the best possible performance point via backup components or parameters controlling the system that make up for the variation effects.

1.1.2 Research objectives

Self-healing can be performed using multiple methods, several of whom are usually dependent on detecting defective components and then repairing them to reintegrate back into the hardware system. The main job of self-healing hardware is to keep the system working with maximum performance after mending the fault in the system.

The difference between self-repairing and self-healing is that self-repairing is the replacement of damaged or faulty components with functioning components in the same area while self-healing is the ability to maintain and reintegration of components into the system. There is also a close relation between self-healing and repairing. Self-healing mends a defective component and is a bottom-up approach, whereas, self-repairing substitutes a defective component and is a top-down approach. A hardware engineer can resort to any way of design he/she wants, bottom-up or top-down while scheming such hardware systems. The term self-healing is often used to specify both self-healing and self-repairing because of the closeness in the meaning of both.

1.2 Literature Review

In Ref [14] the complicated system of NoC (Network on Chip) has billions of transistors, making it subject to failures. As a result, a single transistor failure can bring the entire system down. Faults are defined as data errors or malfunctions, and the sources can include noise, transistor ageing, crosstalk, energetic hits, skin effect, and so on. A router and process element (PE), as well as connectivity links, are the essential components of NoCs. As a result, self-healing NoCs is a hot topic in research to recover errors and maintain system performance.

In Ref [15] because of the high levels of integration and submicron device sizes employed in developing VLSI systems and FPGAs, flaws and operational problems are common. SoCs today are multi-core heterogeneous systems with reconfigurable resources for increased design flexibility. Nanoscale circuits have grown increasingly susceptible to manufacturing flaws, transient faults, and permanent faults as a result of smaller feature sizes and significant reductions in noise margins. Hence, the necessity for fault tolerance and dependability in deployed systems is becoming more important.

In Ref [16] the bulk of biological creatures have a cell-based structure that allows them to develop with fault tolerance and self-repair abilities. Scientific techniques have assisted researchers grasp linked occurrences and associated with principles to build complicated innovative digital systems and increase their capacity by adopting these processes and capacities from nature. Based on these findings, computer-aided modelling, simulation, and experimental research of embryonic systems fault-tolerance and self-healing abilities has been performed, with the goal of developing VLSI hardware structures that can mimic the operation mode of cells or artificial organisms with similar robustness properties as their biological counterparts.

In Ref [17] the goal of deontic logic is to logically encapsulate the concept of norm. It is a kind of modal logic. Database specification, reactive system specification, artificial intelligence, and legal reasoning are all examples of where deontic formalisms have been employed in Computer Science. Deontic logic, according to some academics, is beneficial for thinking about fault-tolerant systems because deontic operators provide a natural means of discriminating between normal (correct, fault-free) and abnormal (faulty) system behaviour.

In Ref [3] during earlier field studies of in-situ measurements, it was realized that there could be many failures in the power management unit mainly due to environmental factors. Regulators used in the power management unit would not get enabled and thereby not provide the regulated voltage required to drive the sensors, signal conditioning, and processing unit. To mitigate the aforementioned issue, we propose a self-healing circuit for the power management unit.

1.3 Novelty of proposed work

The DMR (Dual-Modular Redundancy) technique is used in the power management unit. In this technique, there are two equal occurrences of the component with the same application, present in the circuit (one is main and the other is backup), and the outputs of both of them are connected to a microcontroller that enables the backup whenever a failure occurs in the main. Redundancy is provided by DMR in case one of the components fails. DMR is useful for hardware systems where identical components are working in parallel, especially in self-healing hardware systems. When the design has less area overhead, DMR is preferred to be used for failure tolerant circuits over other redundancy methods.

1.4 Organisation of the Thesis

The organization of thesis is as follows: first of all, the background and technique of self-healing at a hardware level is explained. Post that, the self-healing mechanism for power management unit is designed at a block diagram level. Different integrated circuits are chosen and then made compatible with each other with respect to their input and output characteristics. Simulation on TINA TI software is performed to verify the working of the circuit with the chosen components. Once verified, EAGLE software is used to create schematic of the circuit and then the layout for the same on the printed circuit board. Gerber files are generated and printed circuit board is fabricated for the power management unit.

Chapter 2
Self-healing

2.1 System of self-healing

Self-monitor mode keeps an eye on the system and gives a signal when unbalanced events occur. The following step involves the system changing to a self-diagnosis state where the fault is identified, data is taken out for the erroneous source, and its effect on the hardware system is seen. Upon identification of these, the system will attempt to adjust itself by immediately fixing the faults and switching to the best-expected next state. Figure 2.1 shows the cycle of self-healing of any hardware system.

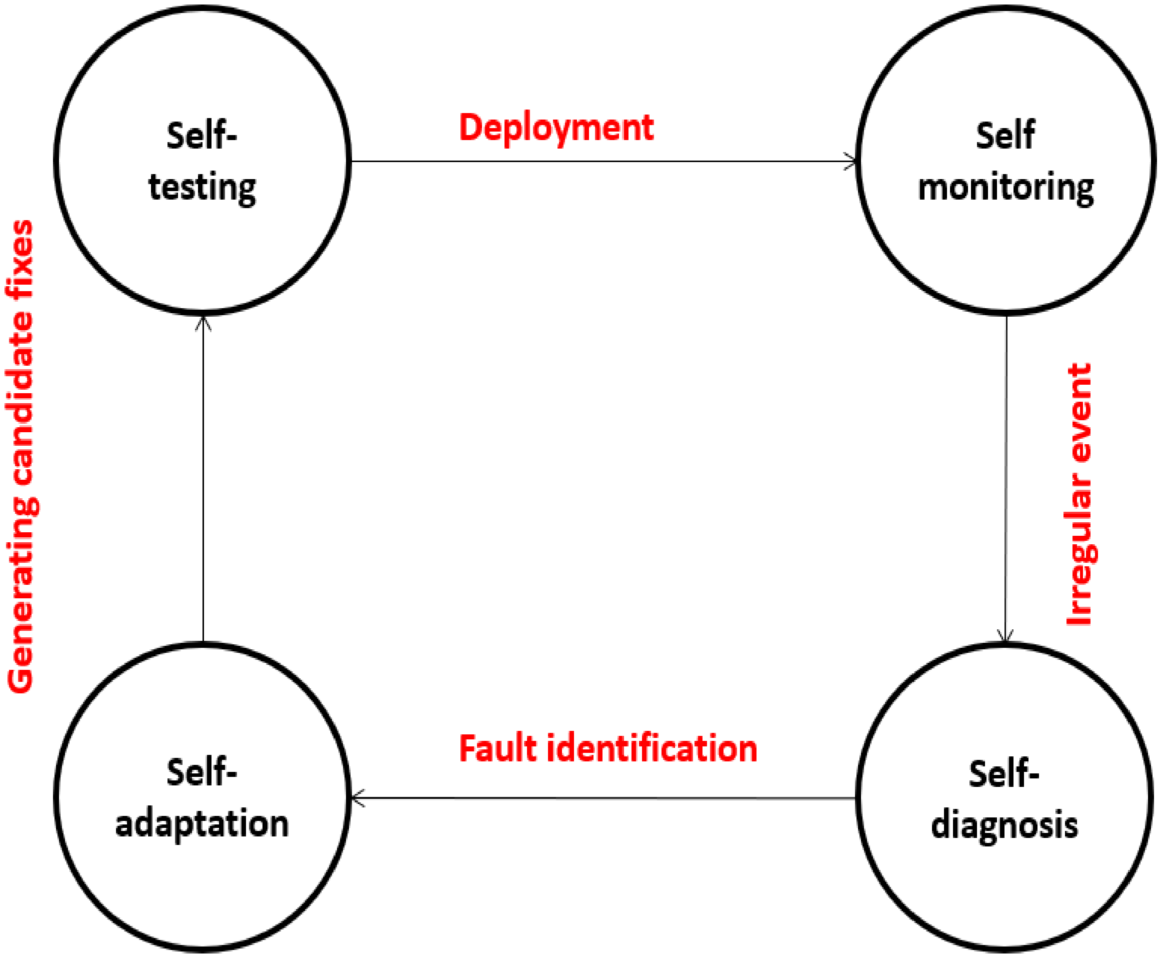


Figure 2.1: The cycle of self-healing of any hardware system [4]

Hardware systems performing self-healing have three important states in the state diagram; normal state, degraded state, and defective state. In the normal state, the system retains its normal functioning to make sure there is no failure. The system changes its state to a degraded state if there is any failure. The system goes back to the normal state from the degraded state if the failure is rectified. The system goes into a defective state if the hardware fails to function. The system keeps trying to fix the failure, in the defective state, and after repairing the failure, the system is restored to the normal original state.

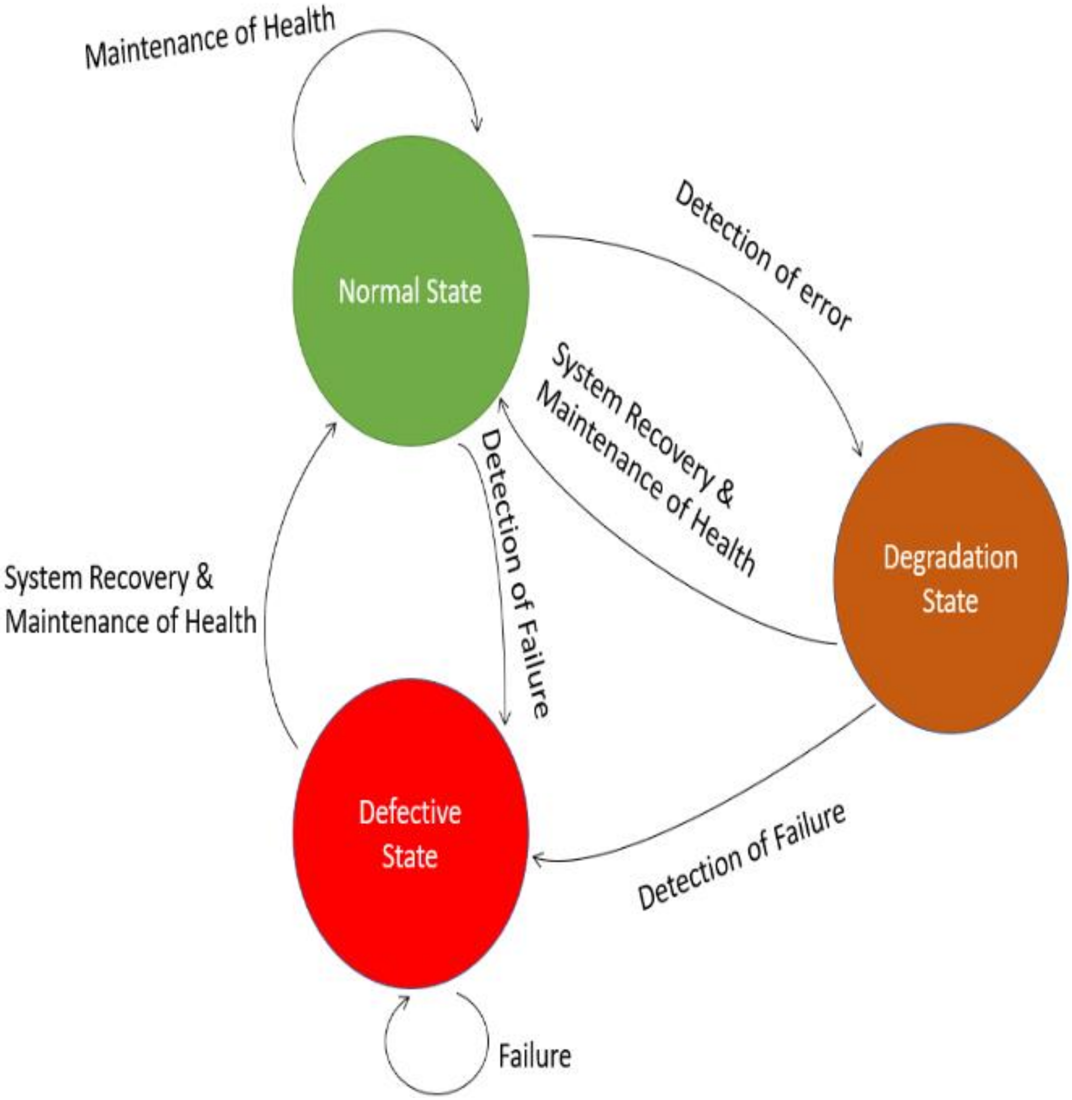


Figure 2.2: Self-healing state diagram [4]

Self-healing properties are shown in Figure 2.3. The vision behind self-healing involves trying to make the hardware accessible and operative always. Some vital attributes of systems performing self-healing are diagnosing, recovering, detecting, isolating, reconfiguration and synchronization. The main aims of self-healing are its ability to survive and system health maintenance.

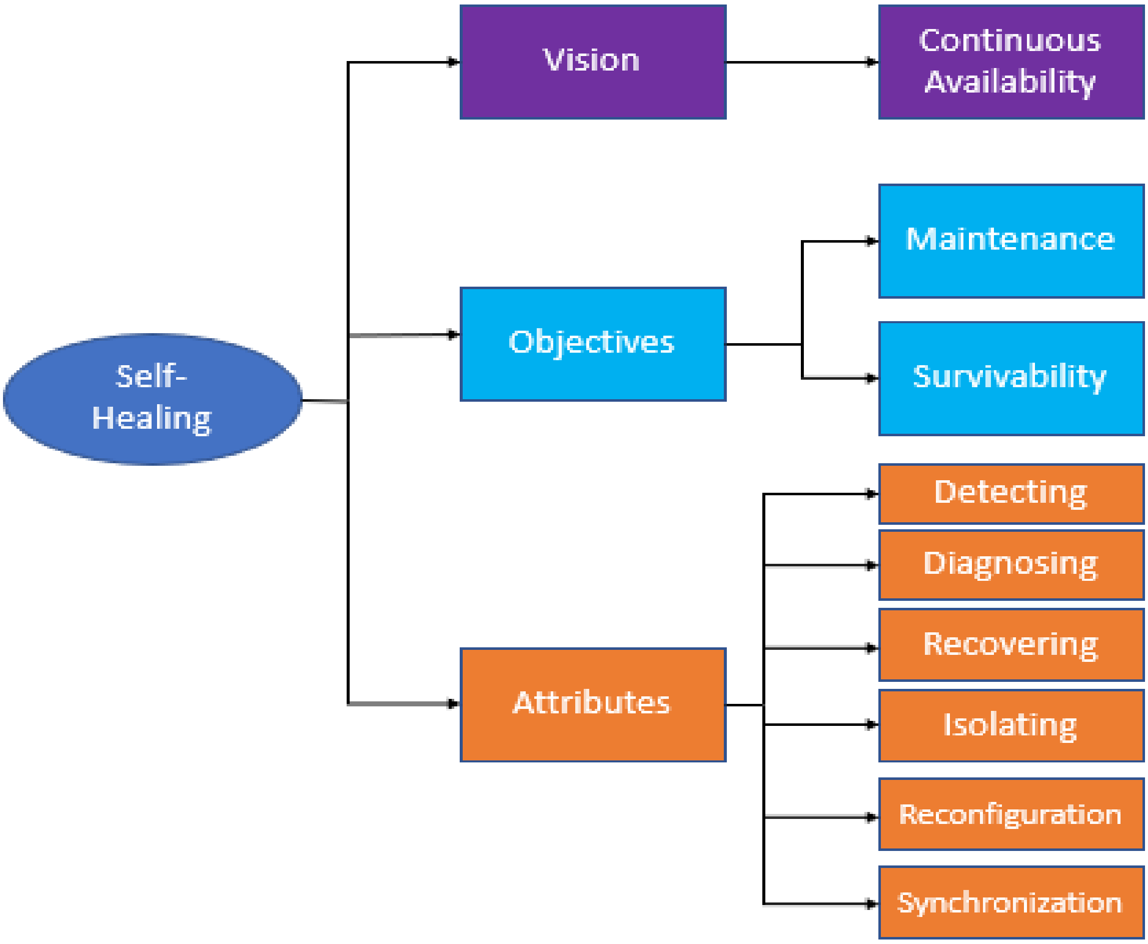


Figure 2.3: Self-healing properties [4]

2.2 Approaches of self-healing

A hardware system must self-heal to maintain the performance of a system under the given conditions. There are many methods concerning self-healing and every method has its advantages and disadvantages. Failures in technical systems are due to miscellaneous reasons which contain chemical, mechanical, electromagnetic, thermal, electrical, or software. Also, the features of failure are everlasting, which says that they continue until amended, or temporary which might ascend repeatedly.

Solutions for self-healing or self-repair have mechanisms that are built on multiple ideologies like exposure to radiation, cold, heat, bringing the system back to its normal state, reshaping, and self-healing the system.

Self-healing methods have certain characteristics which are determined by the following features:

- The redundancy rate and its percentage.
- The type of method used for detection by the hardware performing self-healing.
- The rate at which states of the system are checked and it also includes the sequence at which the state of the system is checked such as arbitrary checking and periodic checking.
- The method of support for retentive system performance.
- Localization of the fault which denotes the technique to identify the location of the failure.
- The process of self-healing which describes the behaviour of the self-healing method.
- The support by which the system reorganizes refers to the system's ability to function properly after self-healing to maintain normal functioning.
- The status of the technique to recover failures whether it is offline or online.

The most common failure tolerant technique is the DMR (Dual-Modular Redundancy) technique which is used in the power management unit. In this technique, there are two equal occurrences of the component with the same application, present in the circuit (one is main and the other is backup), and the outputs of both of them are connected to a microcontroller that enables the backup whenever a failure occurs in the main. Redundancy is provided by DMR in case one of the components fails. DMR is useful for hardware systems where identical components are working in parallel, especially in self-healing hardware systems. When the design has less area overhead, DMR is preferred to be used for failure tolerant circuits over other redundancy methods.

2.3 Levels of Self-healing

2.3.1 Hardware level

The main two sub-levels of a self-healing hardware system are the circuit and the architecture level. The architecture level which has been used for the power management unit delivers self-healing through the use of spare components to substitute defective components. The circuit-level involves circuit duplication, and if the main circuit does not function properly, the backup circuit stands in for the defective circuit. Duplicate elements are utilized to accomplish an increase in power saving or performance and any other desirable parameters, at both the circuit and the architecture level.

Self-healing at an architectural level involves an architecture with extra components and the planning of extra components considering other functional components. One such method describes a technique, in digital circuits, for self-healing, where every functional component is bounded by one extra component. If any of the surrounding working components become damaged, the spare component should automatically turn on. There are two layers to this approach; the upper layer contains practical units such as spare components, working components, and architecture of routing, which is defined as the functional layer. The lowest layer which controls the complete operation of the functional layer is called the control layer. The control layer consists of an ATMEGA328P microcontroller, which wheels the correct duty assignment of extra components for faulty component replacement. The microcontroller can control one working component and its neighbour component (the neighbour component is a spare). Therefore, the microcontroller is aware of the right replacement of the component, and the microcontroller chooses which spare component will mend the defective component.

2.3.2 System level

A program is used to design the self-healing at a system level. This program is responsible for all the functions and tasks the system performs including detecting faults and conducting the repair. The whole system can be planned according to this way of self-healing. Response time and component failure are usually observed features to monitor system performance.

Chapter 3

The power management unit

3.1 Design of the power management unit

Dual modular redundancy is used to design the power management unit shown in Figure 3.1. which is powered by 3.7 V and 2200 mAh Li-ion batteries. This unit supplies power to both the signal conditioning and the signal processing unit. The signal conditioning unit receives a 3.3V regulated DC supply from the power management unit more specifically from the TPS73733 low dropout regulator. The signal processing unit receives a 3.3V regulated DC supply from the LDO MIC 5504-3.3.

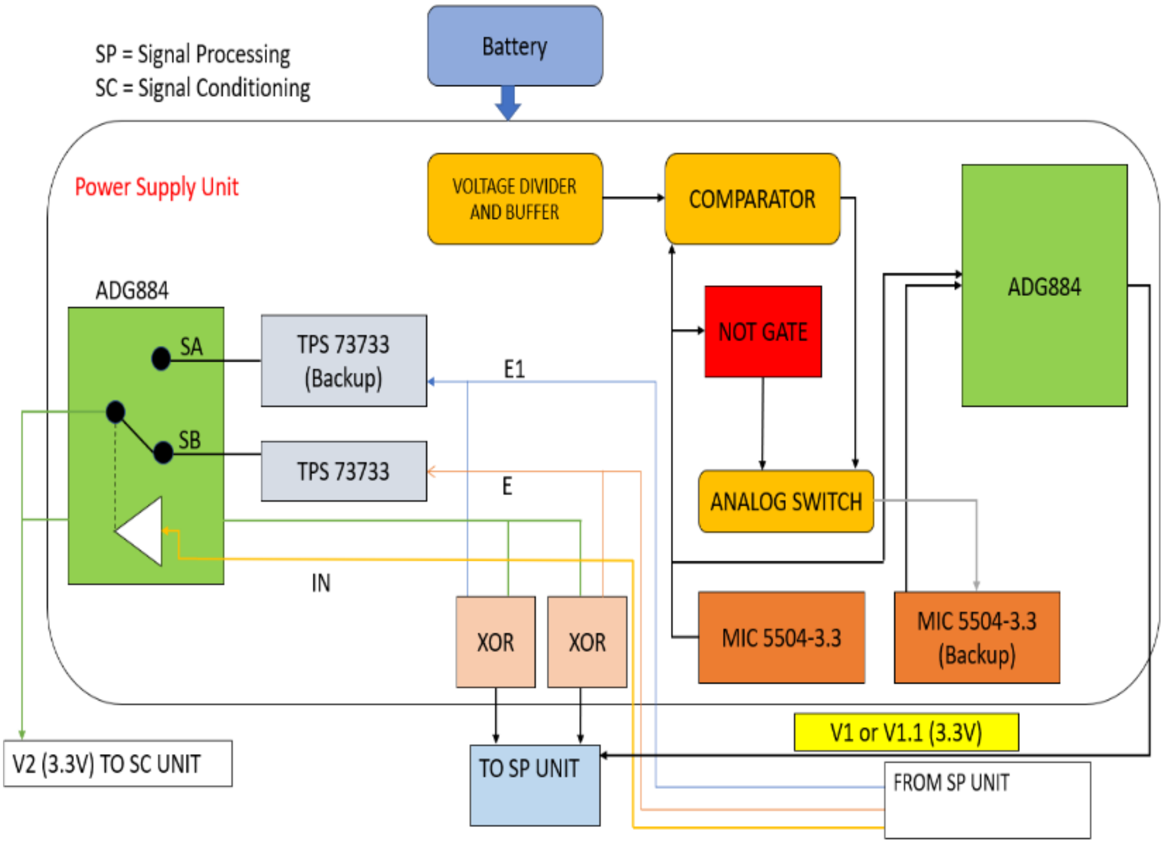


Figure 3.1: Block Diagram of the power management unit

There are two TPS73733 and two MIC 5504-3.3 for backup or self-healing. In case of failure of one, the other takes its place and ensures smooth and continuous operation of the unit. The two ADG884 switches play an important role in the selection between the main and backup LDO for power supply to both the signal conditioning and the signal processing unit.

To enable the backup, for the TPS73733, the enable signal is provided by the ATMEGA328P microcontroller which contains the embedded C code to detect failure or abnormal functioning of the unit. For the MIC 5504-3.3, the backup is enabled via a comparator, not a gate and switch combination. If the primary MIC 5504-3.3 functions, then the output of the comparator is equal in voltage to the output of the MIC 5504-3.3. Hence, when the output of the primary MIC 5504-3.3 is fed to the not gate, it is converted to logic low. The not gate output is fed to the analog switch which in turn does not enable the backup MIC 5504-3.3. The backup is only enabled only if the primary MIC 5504-3.3 fails and the not gate output feeds a logic high to the input of the analog switch. Figure 3.2 shows the circuit diagram of the power management unit.

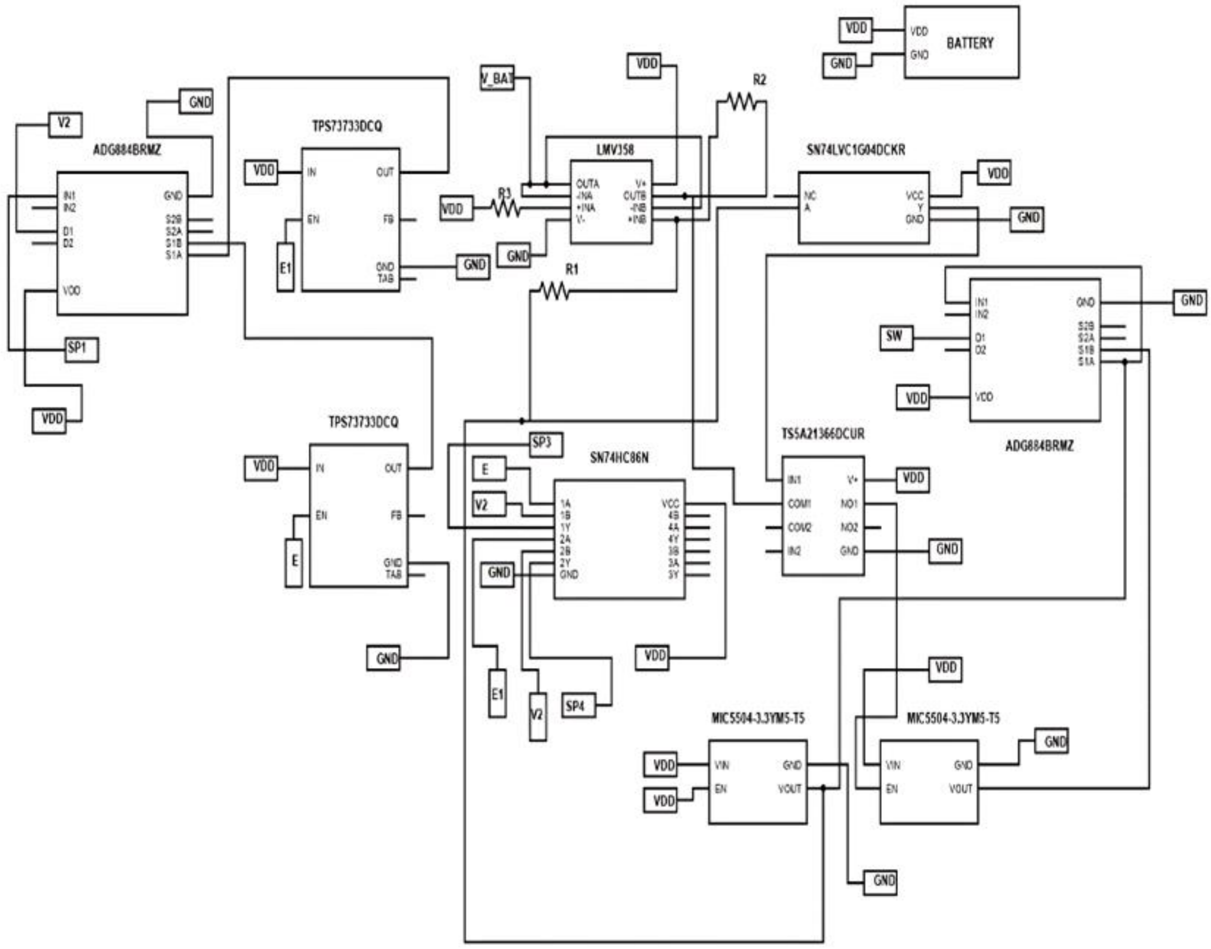


Figure 3.2: Circuit diagram of the power management unit

3.2 Simulation and Fabrication

The proposed power management unit has been simulated on TINA TI software in two parts.

Figure 3.3 shows the first part which is the one providing power to the signal processing unit more specifically the circuit consisting of the LDO MIC 5504-3.3.

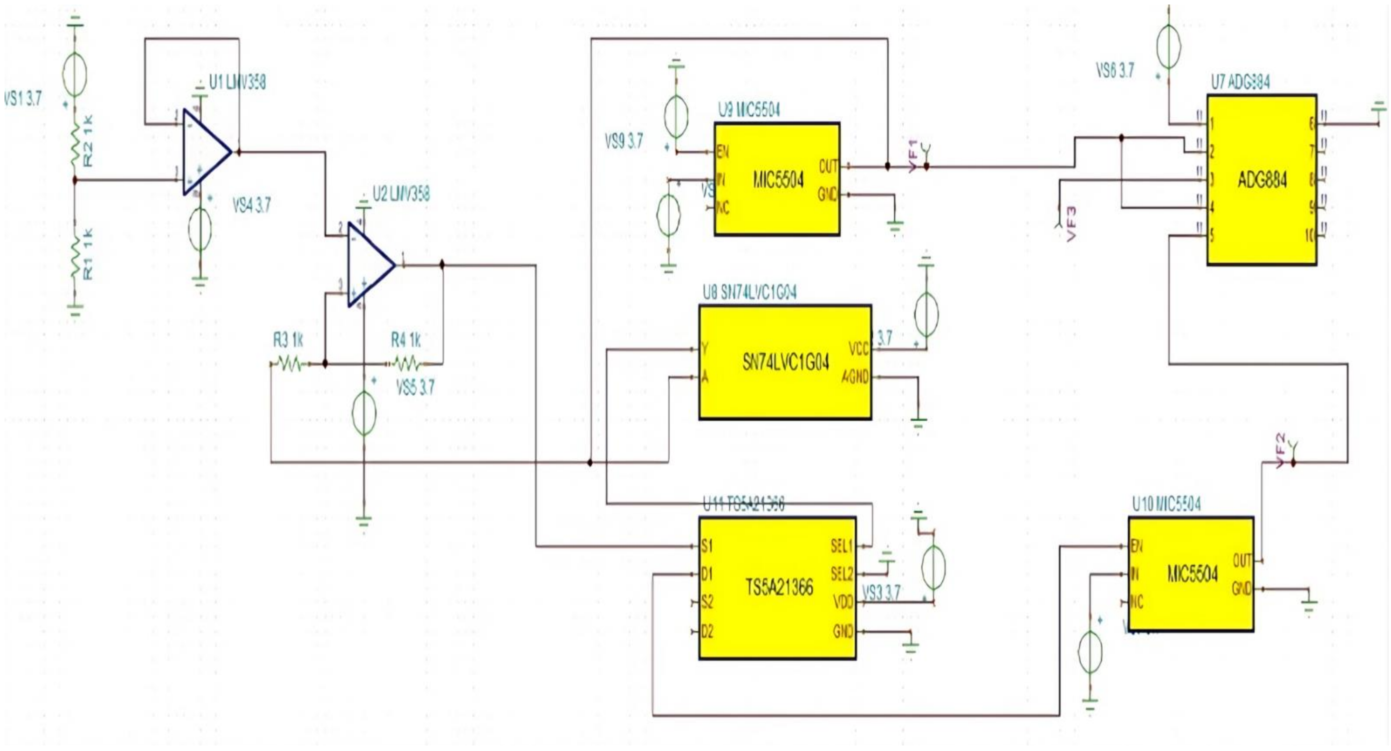


Figure 3.3: Schematic of power unit providing power to signal processing unit

Figure 3.4 shows the output waveform which proves the working of the main MIC 5504-3.3 and the backup is dormant. VF1 is the output of the main MIC 5504-3.3 and VF2 is the output of the backup MIC 5504-3.3. When the main is working fine, 3.3V output is obtained at VF1, and since the backup is not needed so 0V output is obtained at VF2. VF3 is the output of the ADG884 switch which is fed to the signal processing unit. Since ADG884 is the switch that decides between backup and main, its output will always be 3.3V unless both backup and main MIC 5504-3.3 fail.

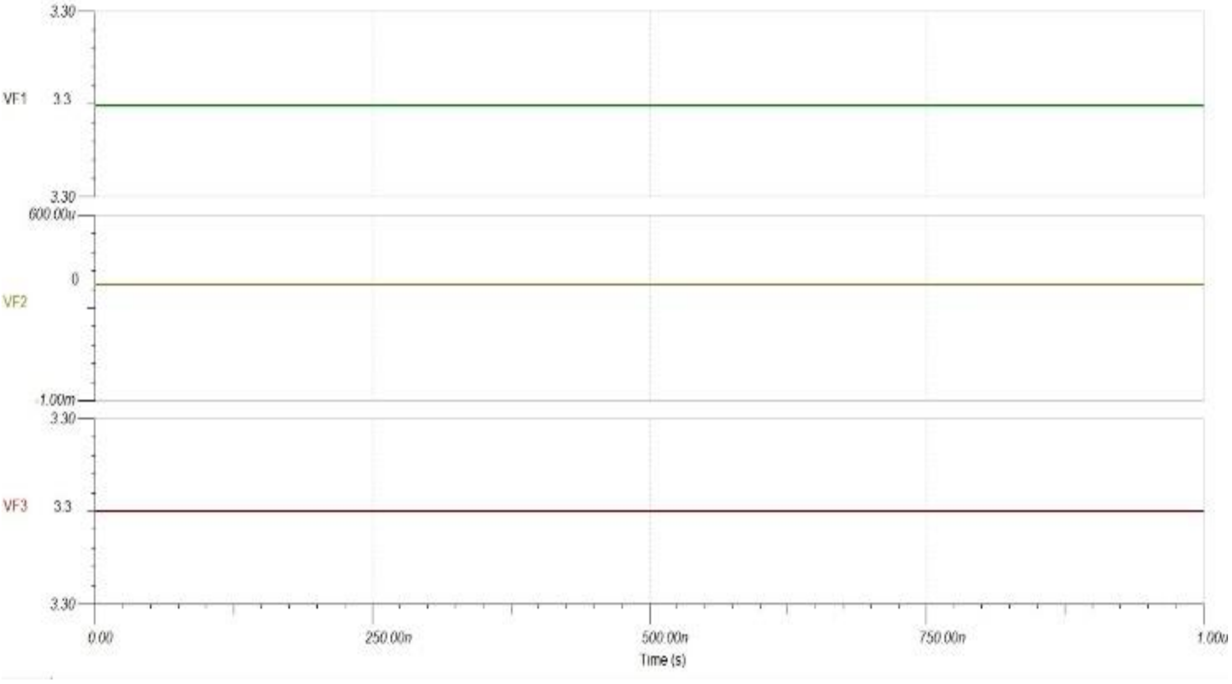


Figure 3.4: Waveform showing working of main MIC 5504-3.3

Figure 3.5 shows the failure of main MIC 5504-3.3 and activation of backup MIC 5504-3.3. In such a case VF1 which the output of the main MIC 5504-3.3 is 0V while VF2 which is the output of the backup MIC 5504-3.3 is 3.3V and VF3 which is the output of the ADG884 switch is also 3.3V.

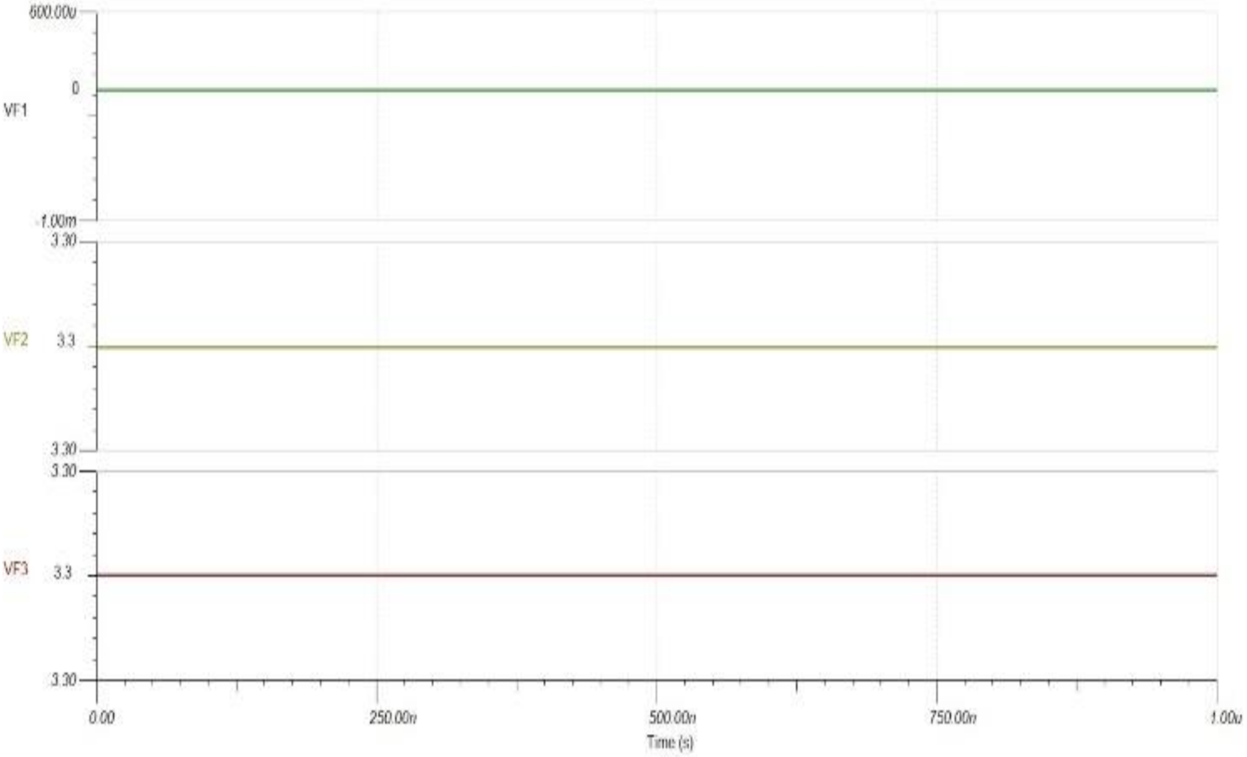


Figure 3.5: Waveform showing working of backup MIC5504-3.3

Figure 3.6 shows the failure of both main and backup MIC 5504-3.3. 0V is obtained at VF1, VF2, and VF3 which are the outputs of main MIC 5504-3.3, backup MIC 5504-3.3 and ADG884 switch respectively.

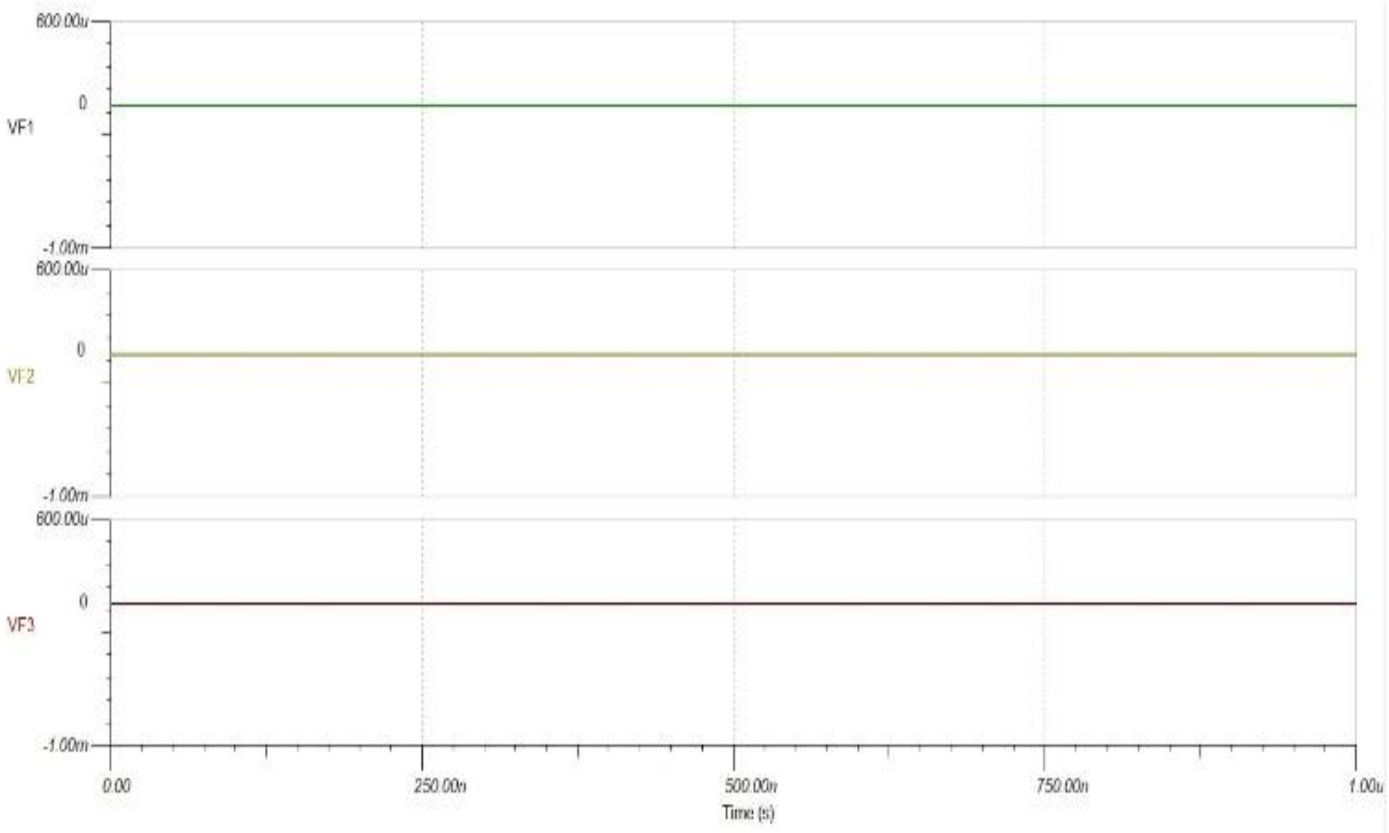


Figure 3.6: Failure of both main and backup MIC 5504-3.3

The truth table 3.1 shows the different possible outcomes at the output of ADG884 switch (VF3) due to main (VF1) and backup (VF2) MIC 5504-3.3. 0 represents 0V whereas 1 represents 3.3V. When main (VF1) and backup (VF2) MIC 5504-3.3 both fail, output at ADG884 switch (VF3) is 0V. When main MIC 5504-3.3 (VF1) fails and the backup MIC 5504-3.3 (VF2) gets activated, output at ADG884 switch (VF3) is 3.3V. When main MIC 5504-3.3 (VF1) is working fine and the backup MIC 5504-3.3 (VF2) is dormant, output at ADG884 switch (VF3) is 3.3V. There is no condition for both main (VF1) and backup (VF2) MIC 5504-3.3 to be activated simultaneously because activation of backup is only possible when main MIC 5504-3.3 fails.

VF1	VF2	VF3
0	0	0
0	1	1
1	0	1
1	1	Not Exist

Table 3.1: Truth table showing different combinations for the working of main (VF1) and backup (VF2) MIC 5504-3.3 and the respective output obtained at the ADG884 switch (VF3)

Figure 3.7 shows the second part of the power management unit which is the one providing power to the signal conditioning unit more specifically the circuit consisting of the LDO TPS73733.

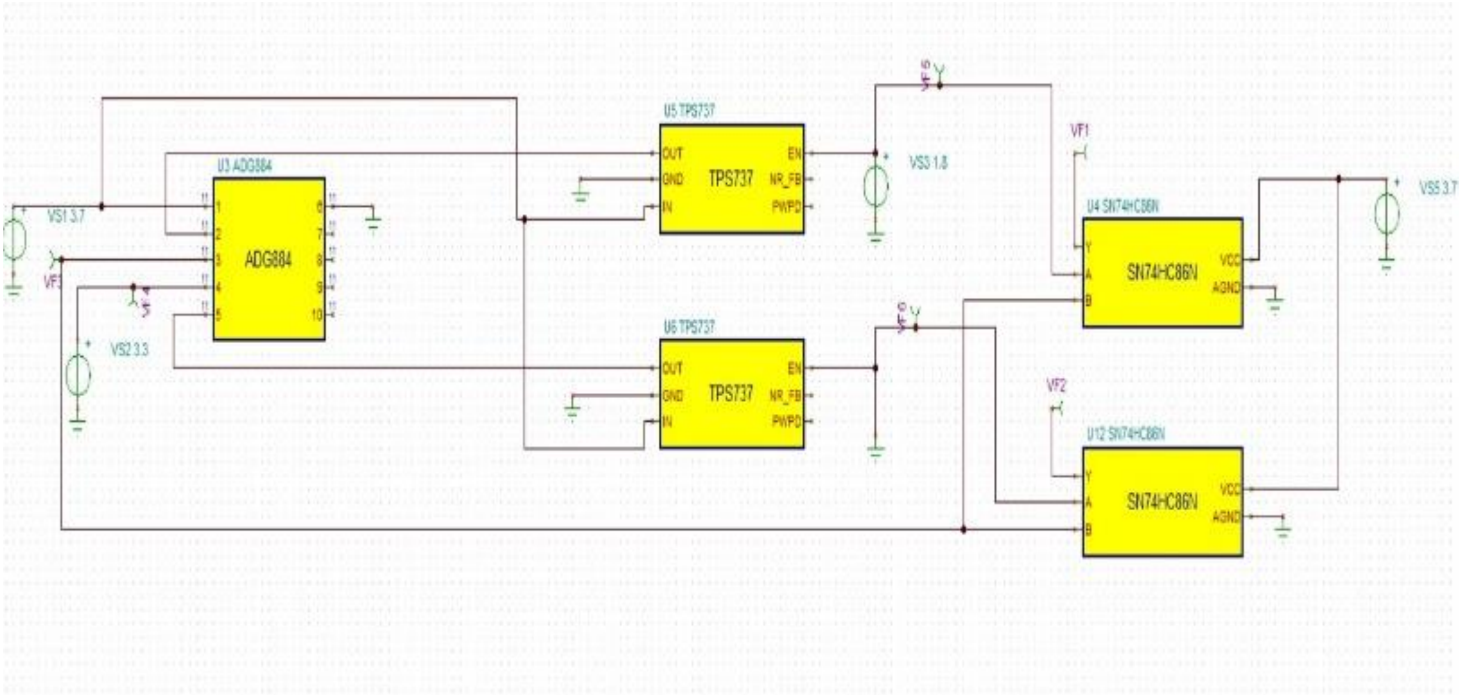


Figure 3.7: Schematic of power unit providing power to signal conditioning unit

Figure 3.8 shows the output waveform which proves the working of the main TPS73733 and the backup is dormant. VF1 is the output of the first XOR gate (SN74HC86N) to which enable pin of main TPS73733 is one input and the output of the ADG884 switch is the other. VF2 is the output of the second XOR gate to which enable pin of backup TPS73733 is one input and the output of the ADG884 switch is the other. VF3 is the output of the ADG884 switch which is fed to the signal conditioning unit. VF4 is the probe connected at the select pin of ADG884 to check the switch between main and backup TPS73733. VF5 and VF6 are probes at the enable pin of the main and backup TPS73733 respectively. When the main is working fine, VF4 and VF5 are high and 3.3V output is obtained at VF3. VF1 is low because both VF5 and VF3 are high, and since the backup is not needed so VF6 is low and high output is obtained at VF2.

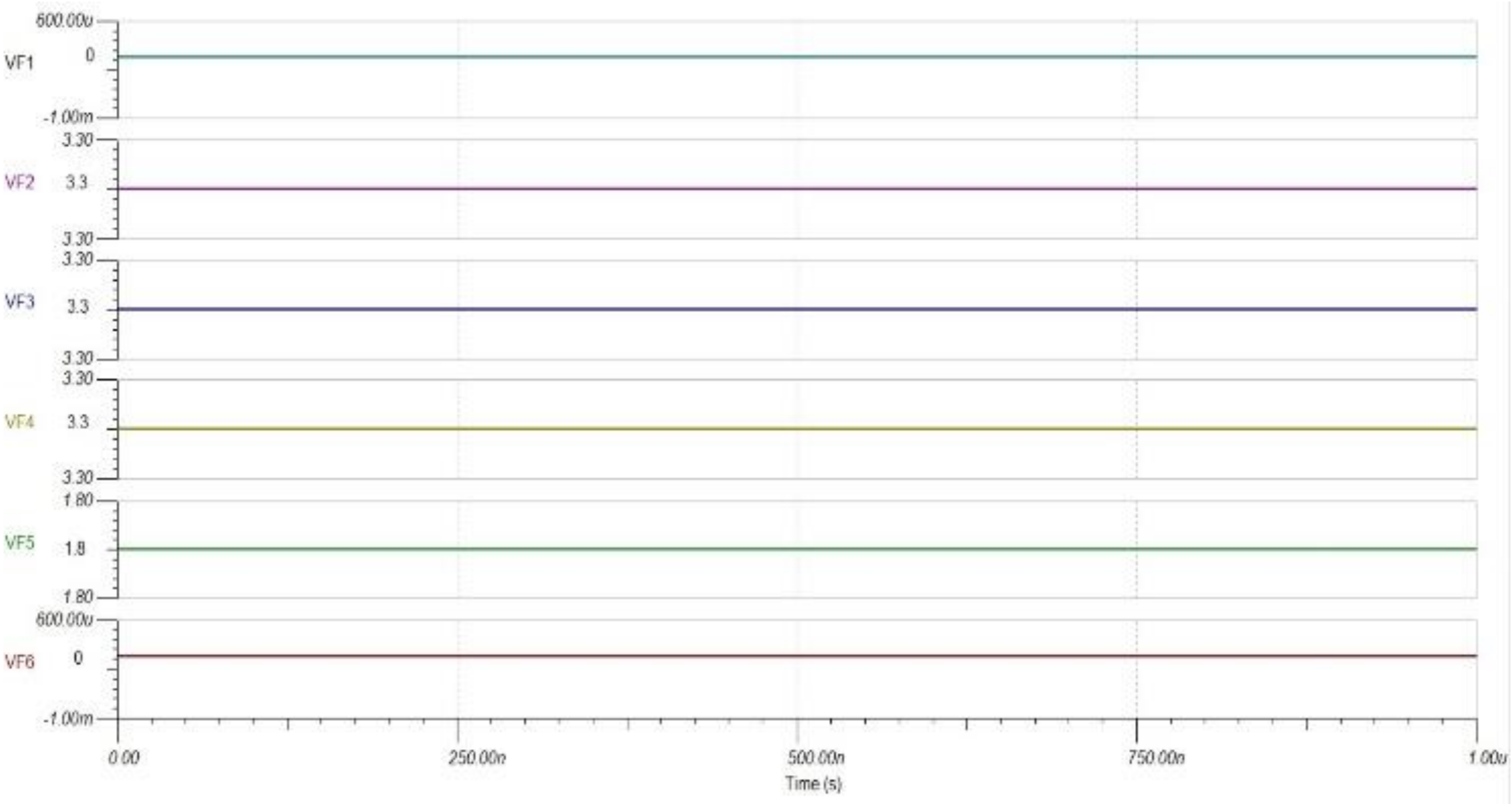


Figure 3.8: Waveform showing working of main TPS73733

Figure 3.9 shows waveform for working of backup TPS73733 and failure of the main one. When the backup starts working, VF4 and VF5 are low and 3.3V output is obtained at VF3. VF1 is high because VF5 is low and VF3 is high, and since the backup is activated so VF6 is high and low output is obtained at VF2.

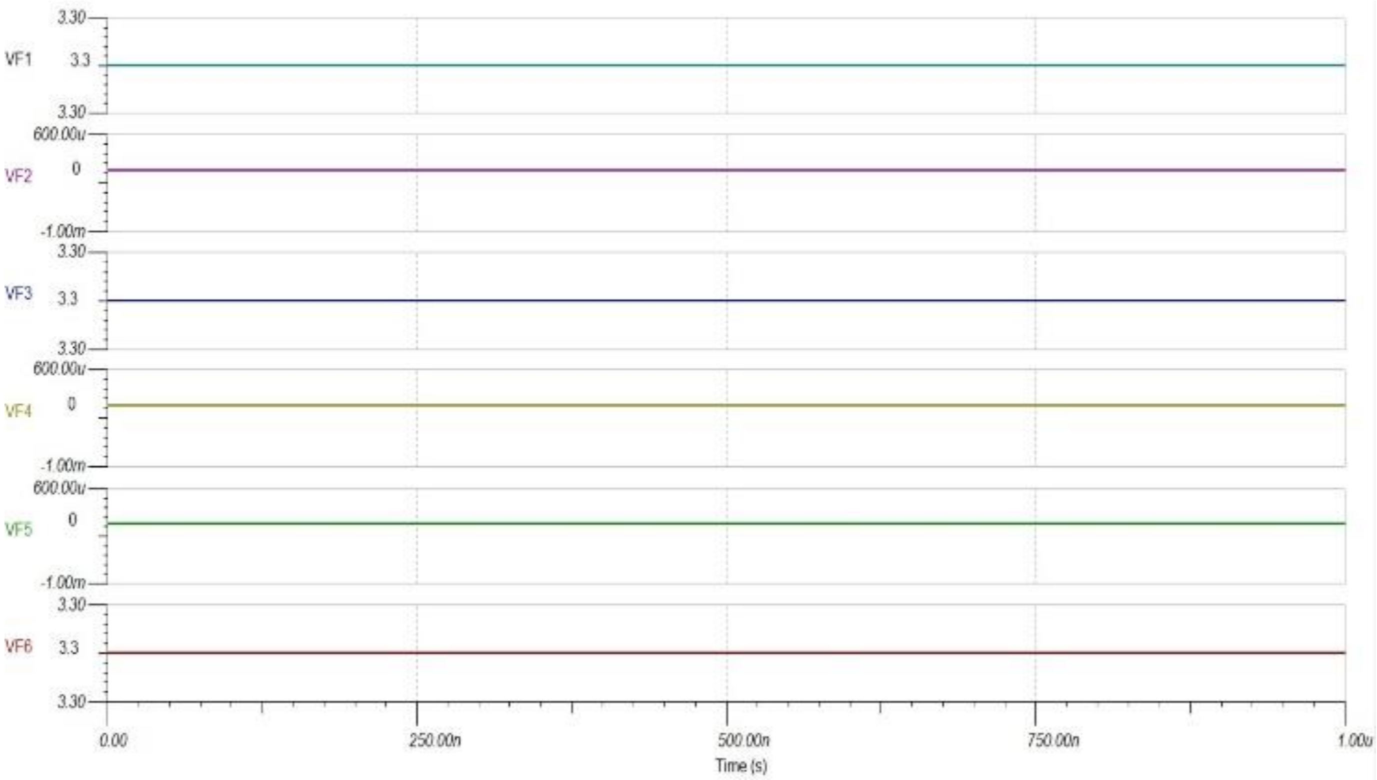


Figure 3.9: Waveform showing working of backup TPS73733

The truth table 3.2 shows different combinations for the main TPS73733 enable (VF5) and output of the ADG884 switch (VF3) and their corresponding impact on the output of the XOR gate (VF1). 0 represents 0V whereas 1 represents 3.3V. When main TPS73733 (VF5) is enabled, the circuit works fine, the output at ADGG84 switch (VF3) is 3.3V and the output of XOR gate (VF1) is 0V. When main TPS73733 (VF5) is enabled, but due to some malfunction in the ADG884 switch, the output at ADGG84 switch (VF3) is 0V then the output of XOR gate (VF1) is 3.3V indicating failure. When main TPS73733 (VF5) fails, backup TPS73733 (VF6) also fails, the output at ADGG84 switch (VF3) is 0V and the output of XOR gate (VF1) is also 0V. When main TPS73733 (VF5) fails, the backup TPS73733 (VF6) is enabled, the output at ADGG84 switch (VF3) is 3.3V and the output of XOR gate (VF1) is 3.3V.

VF3	VF5	VF1
1	1	0
1	0	1
0	0	0
0	1	1

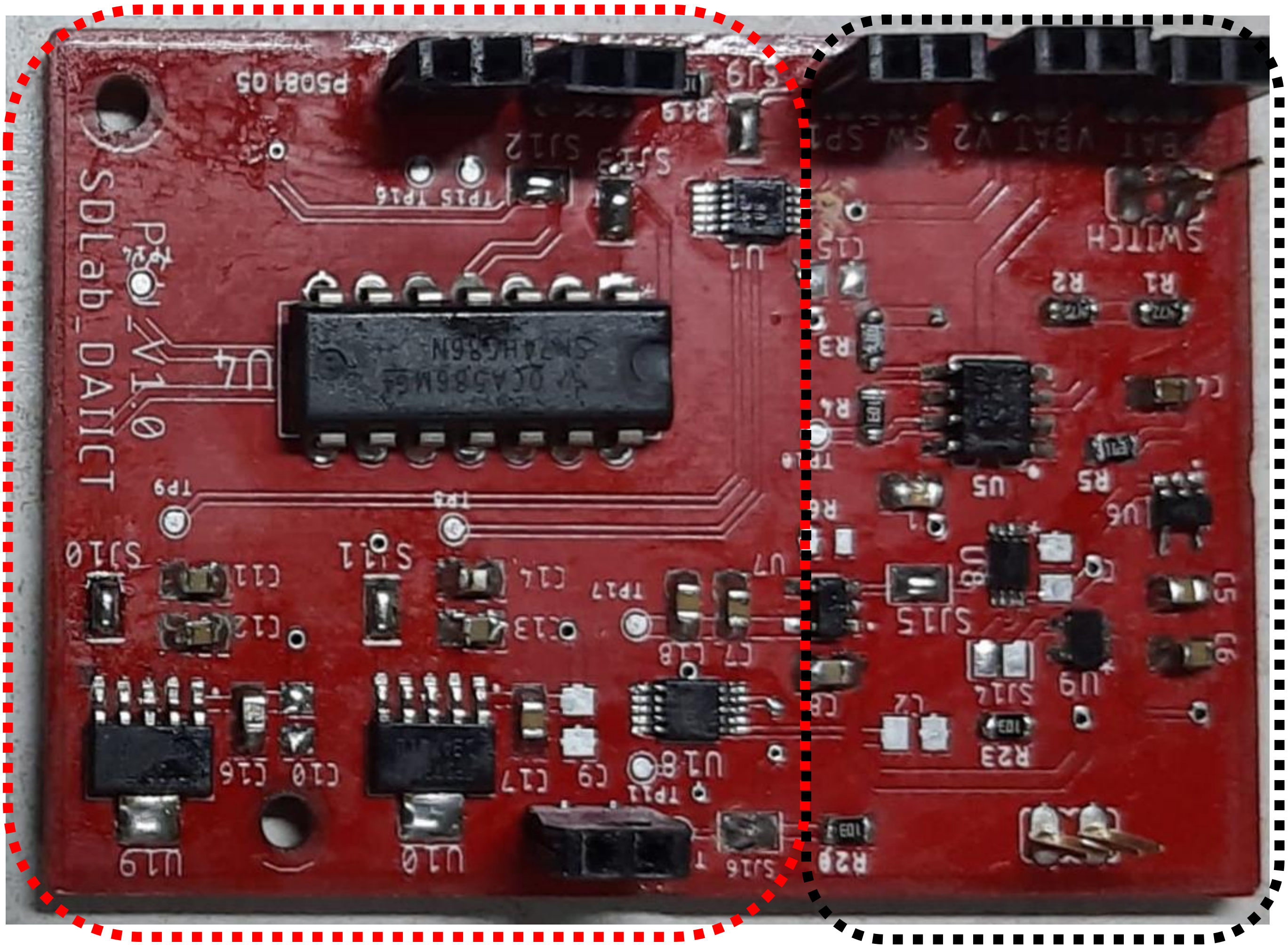
Table 3.2: Truth table showing different combinations for the main TPS73733 enable (VF5) and output of the ADG884 switch (VF3) and their corresponding impact on the output of the XOR gate (VF1)

The truth table 3.3 shows different combinations for the backup TPS73733 enable (VF6) and output of the ADG884 switch (VF3) and their corresponding impact on the output of the XOR gate (VF2). 0 represents 0V whereas 1 represents 3.3V. When backup TPS73733 (VF6) is enabled, the circuit works fine, the output at ADGG84 switch (VF3) is 3.3V and the output of XOR gate (VF2) is 0V. When backup TPS73733 (VF6) is enabled, but due to some malfunction in the ADG884 switch, the output at ADGG84 switch (VF3) is 0V then the output of XOR gate (VF2) is 3.3V indicating failure. When backup TPS73733 (VF6) fails, main TPS73733 (VF5) also fails, the output at ADGG84 switch (VF3) is 0V and the output of XOR gate (VF2) is also 0V. When backup TPS73733 (VF6) is dormant, the main TPS73733 (VF5) is enabled, the output at ADGG84 switch (VF3) is 3.3V and the output of XOR gate (VF2) is 3.3V.

VF3	VF6	VF2
1	1	0
1	0	1
0	0	0
0	1	1

Table 3.3: Truth table showing different combinations for the backup TPS73733 enable (VF6) and output of the ADG884 switch (VF3) and their corresponding impact on the output of the XOR gate (VF2)

Post-simulation EAGLE software was used to create a schematic and PCB layout for the power management unit. PCB layout was confirmed and fabricated too. Components were soldered and hardware testing was performed for the proposed power management unit. Figure 3.10 shows the final PCB of the power management unit. The part highlighted in red shows the self-healing mechanism providing power to signal conditioning unit. The part highlighted in black shows self-healing mechanism providing power to signal processing unit.



Self-healing mechanism for providing power to signal conditioning unit

Self-healing mechanism for providing for power to signal processing unit

Figure 3.10: Final PCB of the power management unit

3.3 Proposed test flow

The proposed power management unit is tested in two parts. The first part that is tested is the self-healing mechanism for providing power to signal conditioning unit. The test flow is to check whether the main and backup TPS73733 is getting enabled properly and the switch is able to select the correct TPS73733 and produce 3.3V output to be fed to signal conditioning unit.

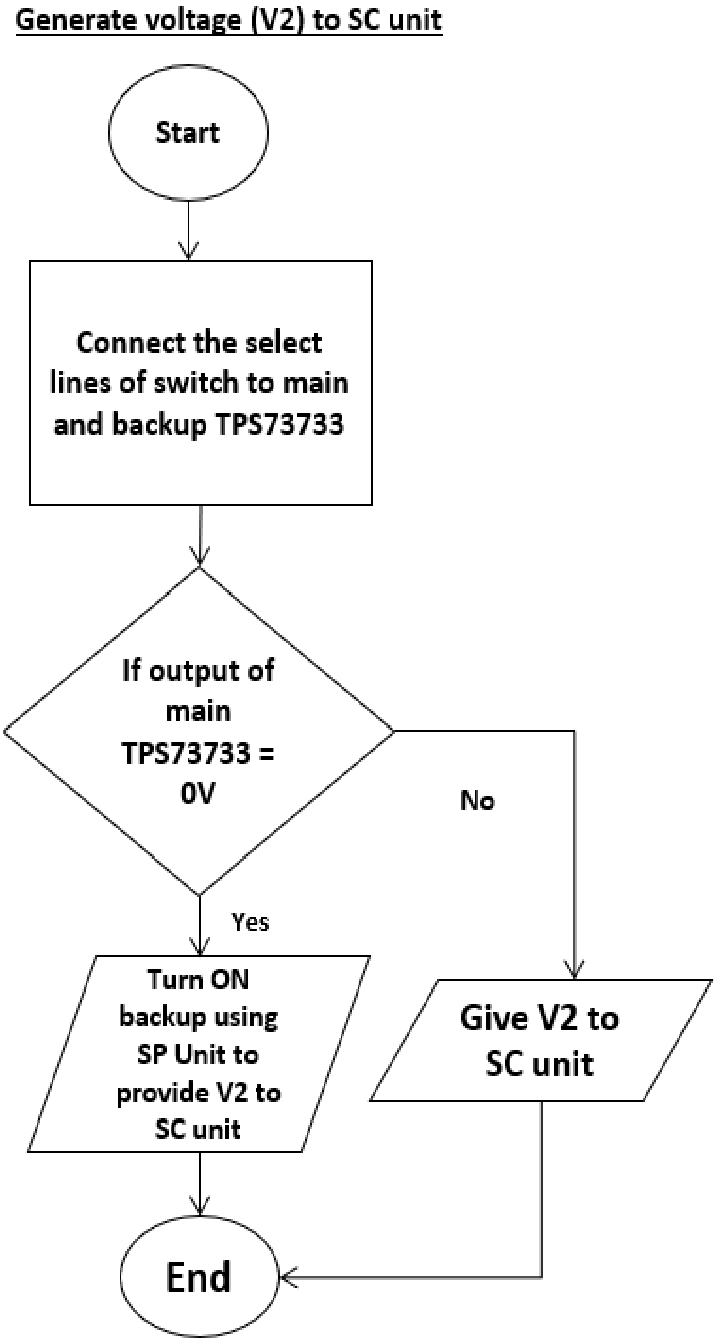

Generate voltage (V2) to SC unit
Start
Connect the select lines of switch to main and backup TPS73733
If output of main TPS73733 = 0V
No
Yes
Turn ON backup using SP Unit to provide V2 to SC unit
Give V2 to SC unit
End

Figure 3.11: Flowchart of the working of the power management unit (SC)

The second part that is tested is the self-healing mechanism for providing power to signal processing unit. The test flow is to check whether the main and backup MIC 5504-3.3 is getting enabled properly via the comparator, not a gate and switch combination, and the ADG884 switch is able to select the correct MIC 5504-3.3 and produce 3.3V output to be fed to signal processing unit.

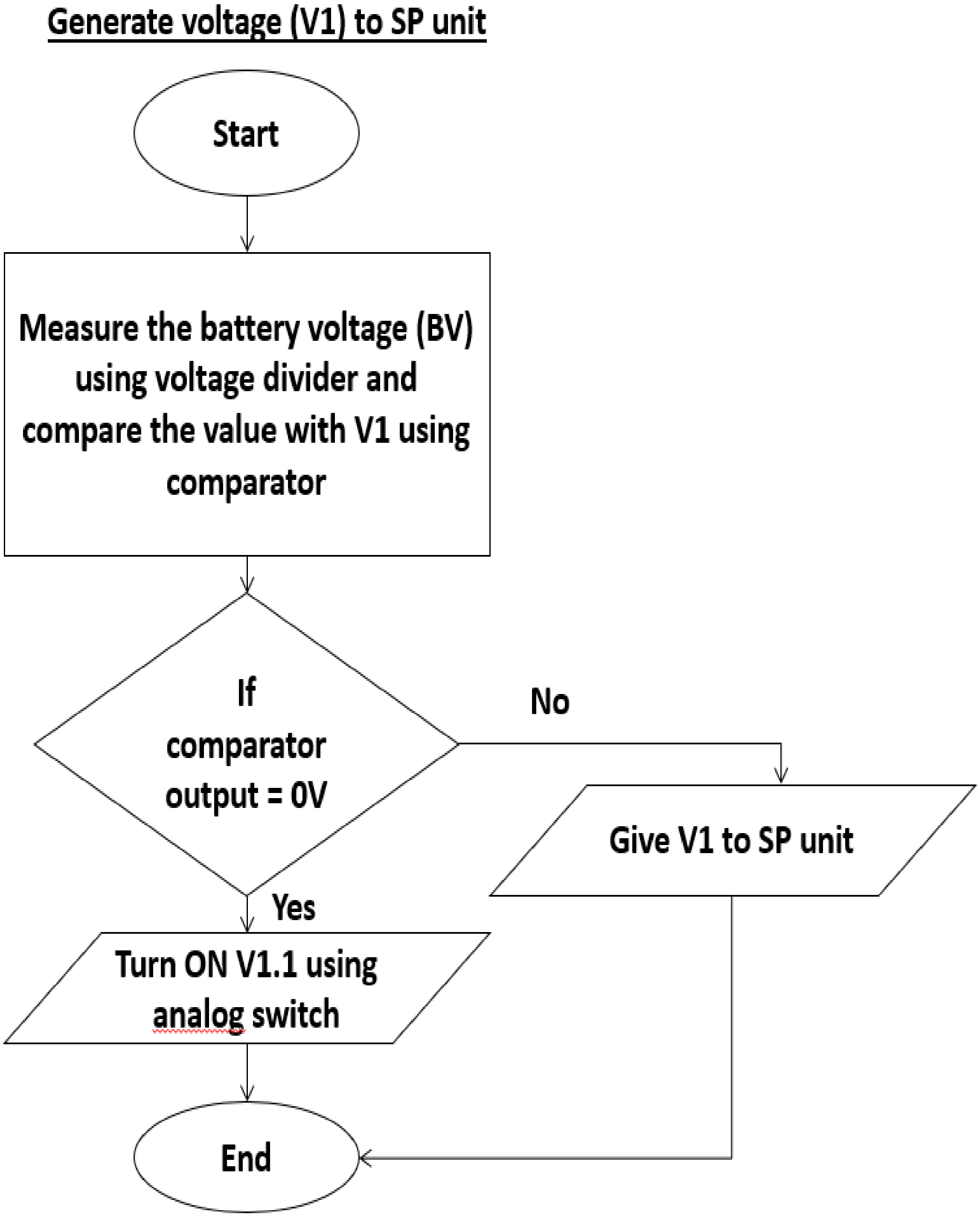

Generate voltage (V1) to SP unit
Start
Measure the battery voltage (BV) using voltage divider and compare the value with V1 using comparator
If comparator output = 0V
No
Give V1 to SP unit
Yes
Turn ON V1.1 using analog switch
End

Figure 3.12: Flowchart of the working of the power management unit (SP)

Chapter 4
Discussions and Conclusion

4.1 Discussion of Results

The working of the self-healing mechanism providing power to signal conditioning unit has been verified by comparing the truth table 3.1 and the simulation results on TINA TI. The conditions in the truth table 3.1 provided a basis to check the working of the unit.

The working of the self-healing mechanism providing power to signal processing unit has been verified by comparing the truth table 3.2 and 3.3 and the simulation results on TINA TI. The conditions in the truth table 3.2 and 3.3 provided a basis to check the working of the unit.

4.2 Conclusion

We have developed a smart and adapting architecture for the power management unit which automatically adjusts the structure of its system and rearranges the tasks and processes of the system. This smart architecture makes sure that the human touch is only required if the backup scenario or plan fails.

4.3 Future Work

1. Develop artificially intelligent code to recognize faults in the system and start its preservation. Real-time data from the field will be required to describe the estimation of failure from system data. The system will be monitored over time and failure will be caught if there is any lapse in data collected. Cataloguing and forecasting the behaviour of the system based on real-time data will be done by using a machine-learning algorithm.

2. Analogue multiplexing technique can be explored for activation of dynamically reconfigurable modules during system failure. By allowing several sensors/components to use a same analogue-to-digital converter, analogue multiplexers provide significant cost, space, and power savings. They also provide a lot of versatility when it comes to modifying circuit connections under computer

control, such as sharing communications buses or changing transducer connections. This thereby will help in selecting main/backup module to ensure continuous system performance and availability.

3. Testing and field deployment of the proposed self-healing power management unit for any IoT-enabled agricultural system.

References

[1] M. Kumar, A. Kumar, and V. S. Palaparthy, “Soil sensors-based prediction system for plant diseases using exploratory data analysis and machine learning,” IEEE Sensors J., early access, Dec. 21, 2020, doi: 10.1109/JSEN.2020.3046295.

[2] V. S. Palaparthy, H. Kalita, S. G. Surya, M. S. Baghini, and M. Aslam, “Graphene oxide-based soil moisture microsensor for in situ agriculture applications,” Sens. Actuators B, Chem., vol. 273, pp. 1660-1669, Nov. 2018.

[3] K. S. Patle, R. Saini, A. Kumar, S. G. Surya, V. S. Palaparthy and K. N. Salama, ″IoT Enabled, Leaf Wetness Sensor on the Flexible Substrates for In-Situ Plant Disease Management,″ in IEEE Sensors Journal, vol. 21, no. 17, pp. 19481-9491,1Sept.1,2021, doi:10.1109/JSEN.2021.3089722.

[4] Khalil, Kasem & Eddash, Omar & Kumar, Ashok & Bayoumi, Magdy. (2019). Self-healing hardware systems: A review. Microelectronics Journal. 93. 104620. 10.1016/j.mejo.2019.104620.

[5] D. Ghosh, R. Sharman, H.R. Rao, S. Upadhyaya, Self-healing systems - survey and synthesis, Decis. Support Syst. 42 (4) (2007) 2164-2185.

[6] K. Khalil, O. Eldash, M. Bayoumi, A cost-effective self-healing approach for reliable hardware systems, in: Circuits and Systems (ISCAS), 2018 IEEE International Symposium on, IEEE, 2018, pp. 1-5.

[7] G. Martinovi ́c, I. Novak, A combined architecture of biologically inspired approaches to self-healing in embedded systems, in: Smart Systems and Technologies (SST), 2017 International Conference on, IEEE, 2017, pp. 17-22.

[8] H. Psaier, S. Dustdar, A survey on self-healing systems: approaches and systems, Computing 91 (1) (2011) 43-73

[9] E. Benkhelifa, A. Pipe, A. Tiwari, Evolvable embryonics: 2-in-1 approach to self-healing systems, Procedia CIRP 11 (2013) 394-399.

[10] S. Narasimhan, S. Paul, R.S. Chakraborty, F. Wolff, C. Papachristou, D.J. Weyer, S. Bhunia, System level self-healing for parametric yield and reliability improvement under power bound, in: Adaptive Hardware and Systems (AHS), 2010 NASA/ESA Conference on, IEEE, 2010, pp. 52-58.

[11] A. Weiss, D. L. Lukens, and J. R. Steadman, “A sensor for the direct measurement of leaf wetness: Construction techniques and testing under controlled conditions,” Agricult. Forest Meteorol., vol. 43, nos. 3-4, pp. 241-249, Sep. 1988.

[12] A. Weiss and D. L. Lukens, “Electronic circuit for detecting leaf wetness and comparison of two sensors,” Plant Diseases, vol. 65, no. 1, pp. 41-43, 1981.

[13] G. Hornero, J. E. Gaitán-Pitre, E. Serrano-Finetti, O. Casas, and R. Pallas-Areny, “A novel low-cost smart leaf wetness sensor,” Comput. Electron. Agricult., vol. 143, pp. 286-292, Dec. 2017.

[14] K. Khalil, O. Eldash and M. Bayoumi, "Self-healing router architecture for reliable network-on-chips," 2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2017, pp. 330-333, doi: 10.1109/ICECS.2017.8292030.

[15] D. Jose, P. N. Kumar and S. Ramkumar, "Reliability aware self-healing FFT system employing partial reconfiguration for reduced power consumption," Proceedings of the 2014 IEEE Students' Technology Symposium, 2014, pp. 31-36, doi: 10.1109/TechSym.2014.6807909.

[16] C. Szasz and V. Chindris, "Fault-tolerance properties and self-healing abilities implementation in FPGA-based embryonic hardware systems," 2009 7th IEEE International Conference on Industrial Informatics, 2009, pp. 155-160, doi: 10.1109/INDIN.2009.5195795.

[17] R. Demasi, "Synthesizing fault-tolerant programs from deontic logic specifications," 2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2013, pp. 750-753, doi: 10.1109/ASE.2013.6693149.