Safety Mechanism for Faults in SRAM in Multicore Microcontroller to Check Safe Ram Functionality

Thejaswini k p[1], Neethi M[2], Shaik Mohammed Yaseen Basha [3]
[1]*PG Scholar, Electrical & Electronic Engineering, JSS Science & Technology University, Mysuru*
[2]*Associate professor, Electrical & Electronic Engineering, JSS Science & Technology University, Mysuru*
[3]*SW Architect, Bosch global Software Technologies, Bangalore*

***Abstract*— A transmission control unit (TCU) is one of the types of automotive control unit that is ECU which is used to control the gearing system automatically. This TCU plays a vital position in today's cutting-edge automobile for automating the manual gearing system.**
In modern automatic transmission, the TCU uses sensors to fetch data from the automotive vehicle, and also from the electronic control unit (ECU), in order to calculate, how gear changing control in the vehicle to enhance the overall performance, shift quality and fuel consumption thus improving the sophistication in our daily lifestyles by upgrading to automatic gearing system from manual gearing system and improving the quality of automation. being this important, creating such a highly intelligent and also improved sophisticated control software for the TCU is also becoming challenging. Such system is mainly developed with micro controller which act as sensor, processor, and actuators. If an error or fault is occurring in microcontroller, then it will damage the performance of microcontroller, and which leads to the failure of cascade system which is in contact with the microcontroller. These types of failures can be avoided by adapting the standards created by functional safety ISO (26262). In this project, the different faults that can be occurred in general in ram memory and safety mechanism for those faults has been discussed and implemented.
modern electronic control unit consist of many components. We should ensure the safety of all these components by assigning this with automotive safety integrity level. To make sure the safety, system methodologies followed are partitioning of memory for a different ASIL and developing entire software component according to ASIL depending on applications.
The present paper gives info on configuration and reveal required hardware protection mechanism that are in built and additionally gives software program protection system for memory. For simulation work which we carried out, AURIX TC37X micro controller has been used which is from Infineon technology having 32-bit tri-core architecture.

I.INTRODUCTION

In the era of electronics, manual operating equipment is being replaced by automatic sensors, controllers, and actuators. Automotive gearing vehicles being our future and have been embedded with hundreds of microcontroller units. Controlling and monitoring these controllers bring many challenges that need to be analyzed properly. Due to the nature of the devices, microcontrollers are prone to external disturbances, magnetic interference, and signal glitches, due to these reasons interference in the network communication causes flip of the bit/bits in the memory, these scenario leads to error in the memory that can compromise the data in the memory further safety of the passengers. In order to find out and monitor the errors which usually arise when control units fail, we should implement a safety mechanism feature.

The gear changes are performed by hydraulic controls in passenger cars and also heavy-duty vehicle, these gears are controlled and then monitored via a control unit which is placed in vehicle engine. The control unit receives continuous warnings from sensors, keeps the records for processing and compares the set point information for test. Microcontrollers used in these cases are application specific integrated circuits (ASIC). Any corruption or error in the data processing and storage can lead to drive errors, system crashes and thus system failures. The memory is the part of the protection of the control unit. So, it's very much essential to protect it from unwanted writing and external disturbances in order to achieve safety goals.

International standard ISO 26262 has given standard guidelines under the concept "Functional safety" which has been accepted as general standard to make vehicle electronically at safe state and to protect drivers, pedestrians, and road users from damage caused in the car due to electronic and software fault. Today's vehicles are embedded with more and more software to implement advanced features into the vehicle to enhances the driving features. For example, automotive driving system, cruise controller, automatic speed controller. A fault or error in the software may cause problems, can harm people, and can create hazards for road participants. For example, a fault in the software that lets an uncontrolled acceleration causes traffic tragedy. to reduce these risks in road vehicles due to electronic failure, ISO 26262, Functional safety set certain pre-implementation standards to develop software.

The Memory Test Unit (MTU) screens the test, controls and initialization with the information keenness by checking capacities of the different memories available in the device. Each SRAM in the MTU has a few computerized rationales encompassing it, known as SRAM Support Hardware (SSH). SSH is a hardware square which usually controls the fault Location & Adjustment, Memory Built-In-Self-Test (MBIST) . all SSH square gives an interface in order to control its different actions. There will be many such SSH occurrences, each of these occurrences controls many distinctive memories present internally. The Memory Test Unit (MTU) present in the Aurix gives register interfacing to design and also to control different controllers in the memory.

II. MEMORY TEST UNIT

a.Block diagram

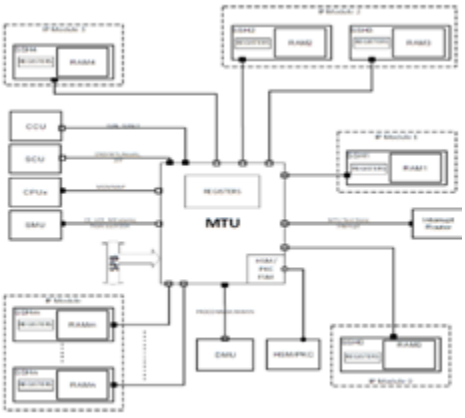


Fig 1: MTU and SSH interface

b. List of features available in MTU

1. interface to all internal available SSH instances

- The MTU gives registers to control the functionality of available SSH instance.

2.Initialization of Data

- In order to prevent data read-out through the MTU, memories which are security sensitive can be auto initialized.
- Through MTU hardware, each SRAM block can be initialized.

3.Memory Error Correction & Detection

- Detecting Address Error
- Detecting Correctable and uncorrectable error
- Detecting Memory error and correcting the same for SRAM in the system can be done through the MTU.

4.Alarm notification to safety management unit: 3 alarms are sent to the MTU from each SRAM/SSH, and then sent to SMU.

c. Overview

Figure 1 "MTU & SSH interface" gives the system level view of the Memory test unit,
SSH Instances, and SRAMs.

IP Modules in the system such as Local memory unit, CPU, etc. have more than one or one SRAM.

These SRAM is surrounded by MBIST logic, SSH Registers and SSH module of its own.

Each SSH is connected to the MTU separately through internal interfaces.

The MTU is in turn connected to the SPB bus, and through this SPB bus every SSH's registers are usually accessed. internally each SPB validation is forwarded back and forth by the MTU with the corresponding SSH.

d. SRAM Support Hardware (SSH)

Depending on the configuration, each internal RAM or multiple RAM in the AurixTC3XX Platform will be having additional logic which is nothing but SSH. In the device, All the SSH instances are connected to an inside bus through MTU-SSH connection. SSH supports direct connection to the memories, even without using CPU.

IV. DESIGN AND HARDWARE ARCHITECTURE

a.Design approach

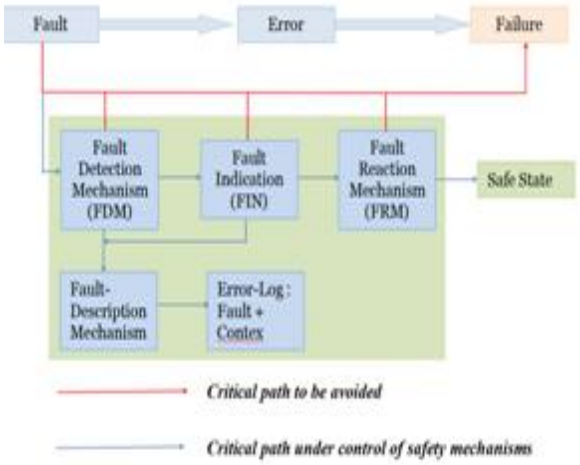


Fig 2: Systematic design approach to handle errors

Any error in the components or in any functionalities leads to the damaging conditions in the car which leads to failure of devices. So, developing proper approach to handle such errors, safety mechanism is very much essential. Fig 2 shows design approach which has been considered systematically to handle errors in which red line indicates the critical path which needs to be avoided. The design should be in such a way where control of the fault should be done using safety mechanism. Here fault detection mechanism (FDM) will detect the fault and indicates the type of fault and based on it the reaction mechanism should get activated and SMU will take care of the activation of safety registers and helps the system to be in safe state. And also, we should maintain the description of the fault and log the error which will assist in handling the fault condition. The steps that are followed in this approach has been mentioned in the fig 3

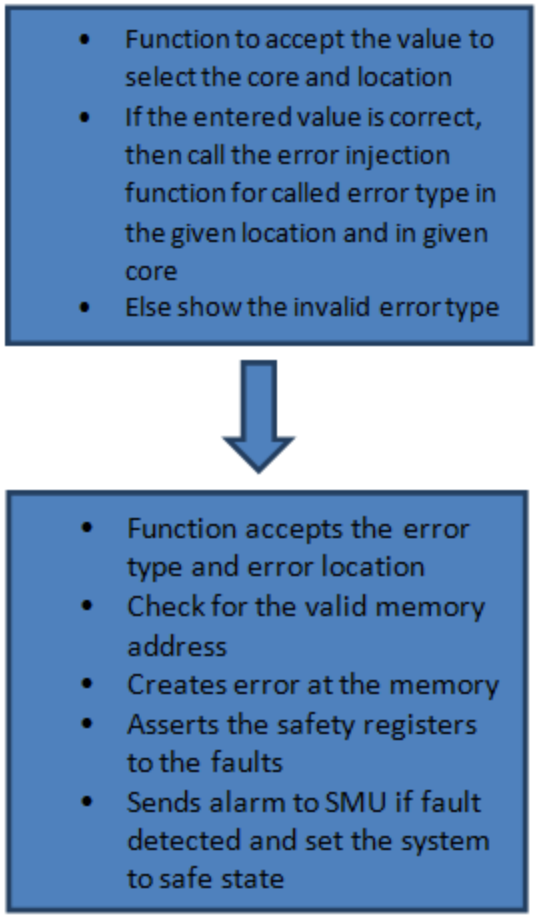


Fig3: Steps for error handling

b.Hardware architecture

The micro controller we have selected is from Aurix TC3xx from the Infineon family, it is 32-bit microcontroller with TriCore Architecture depending on our applications. The AURIX™ TC3xx is known for its high performance with multicore CPUs. It is having DMA controller, bus arbitration, interrupt system, buses, and on chip peripherals. The Aurix series of TC3xx offers many on-chip peripheral units, namely analog-to-digital converters, serial controllers, and timer units. These units are connected to the TriCore CPUs within the Aurix TC3xx Platform through Shared Resource Interconnect (SRI) and also System Peripheral Bus (SPB). on the AURIX™ TC3xx Platform ports, several I/O lines are there for

Feature		TC33x	TC36x	TC37x	TC38x	TC39x
Cache per CPU	program	32kb	32kb	32kb	32kb	32kb
	data	16kb	16kb	16kb	16kb	16kb
SRAM per CPU	PSPR	8kb	32kb	64kb	64kb	64kb
	DSPR	192kb in CPU0	192kb	240kb and 90kb respectively in cpu0, cpu1 & others	240kb and 90kb respectively in cpu0, cpu1 & others	240kb and 90kb respectively in cpu0, cpu1 & others
	DLMU	8kb CPU0	64kb	64kb	64kb	64kb
SRAM global	LMU				128kb	768kb
	DAM			32kb	64kb	128kb

Fig 4: Platform feature overview with memory

all of these peripheral units in order to connect with the external world. Fig 4 shows the memory allocation for different features which we have used as location to inject our faults. To achieve these levels of economy, speed, and power for applications in embedded field, three powerful technologies within one processing unit are combined in the Tricore processor architecture, they are

•Digital Signal Processing (DSP) operations and addressing modes

•On-chip memories and peripherals

•The Processor architecture is Reduced Instruction Set Computing (RISC)

V. IMPLEMENTATION

Now a days we are more adversely relied on electronic control and management comes concern for preferring the safety of the electronic operation. One main concern is the SRAM operation in the system. This is very important and from some perspectives, this is the most important aspect that should be maintained in the execution of electronics.

This is because:

• In the overall device, SRAM is a large component. We can say the largest.

– because its largest even in the area

– and also, in the transistor count

• SRAM is known for dense circuitry and therefore it is susceptible to subtle defects or disturb.

• SRAM are susceptible to disturbances as they operate in normal circuit logic vs low voltage range.

With these reasons, for any new integrated circuit process development, SRAM bit cell and array layouts are the two key components. To find drift and variation in manufacturing this is where integrated circuit fabrication experts will usually prefer. Therefore, in safety related electronics, SRAM Error Detection is important. This is usually done with the help of Memory test unit (MTU) and Safety Mechanism Unit (SMU) to which SSH is connected.

a.Enabling SSH

An enable bit associated with Each SSH in the Aurix Plus Platform in the MTU_MEMTEST register. each SSH control registers are accessible only when particular registers are enabled by the SSH (i.e. ALMSRCS, ETRRx, FAULTSTS, ECCS, ECCD, and ERRINFOx) which can be accessed to modify through the MTU_CLC register. when the SSH is enabled through MEMTEST register, only then in the SSH, all other registers can be accessed if not the MTU gives an error in SPB. For some of the SSHs, through MEMTEST register enabling or disabling the SSH will result in complete or partial memory to be initialized automatically which will be depending on configuration. Hundreds of clock cycles can be taken to complete this auto-initialization. During this time, SPB bus error will occur if register interface to the SSH which is responsible for auto-initialization and no SSH register can be accessed. Note that the module having SRAM needs to be enabled through CCU registers which is having clock input before SSH registers will going to be accessed through the MTU. Module clock need not to be enabled Other than this CCU clock configuration (e.g., to access SSH registers using the CLC register). via a set of registers, each SSH instance operation and functionality can be controlled. These registers can be accessed through MTU as discussed. Below errors can be monitored using SSH instance in the given memory

1.Correctable error-single bit

2.Uncorrectable error- double bit

3.Uncorrectable critical error- address buffer overflow

4.Miscellaneous error-Ram operations monitor
5.Miscellaneous error-Ram safety mechanism
Each of these errors will be monitored with the help of special interface which is been provided by SSH. they are error correction code safety register (ECC safety register) which will be enabled by SMU.As a safety feature, alarm will be sent to safety management unit, if there is any error in the given memory section. then every bit in the ECCS register will be enabled through MTU and SSH registers. In this case, there will be three-alarm which are generated in ECC, they are correctable alarm, which is usually single bit type of errors, uncorrectable alarm which are generally double bit type of errors, and the last one is miscellaneous alarm to the SMU.
When SMU gets these signals then it will analyze the type of fault and it will bring the system to the safe state by resetting the system and then the trap will be generated.

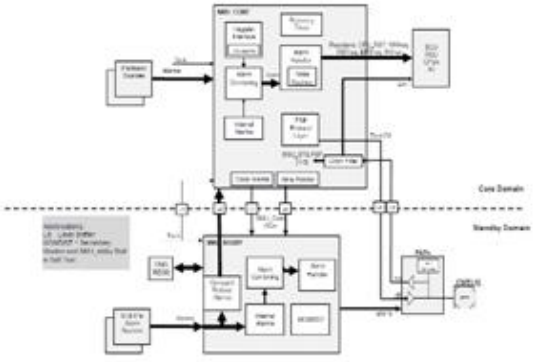
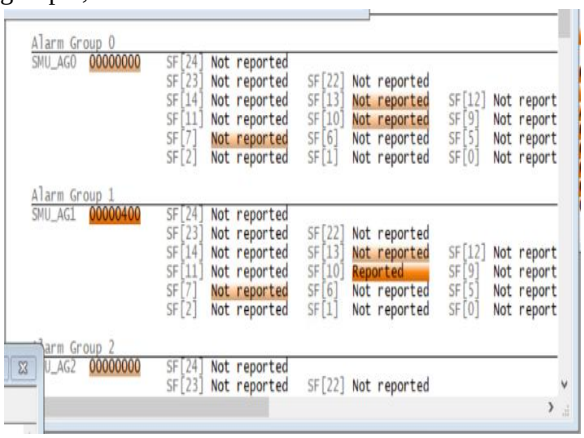


Fig 5: shows SMU unit architecture

VI. RESULT AND DISCUSSION

These test cases are to verify all the mentioned error scenarios.When an error happen, SMU alarms will be set corresponding to error bit and error pin is also active to bring system to safety state.
Case 1: single bit error and double bit error
INPUT:
Error injected at DSPR in CORE 1
Type of error injected = double bit error
Alarm should be generated at group, 10th bit

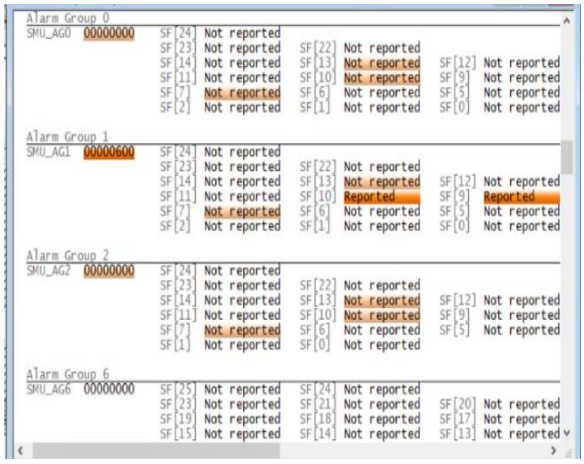
OUTPUT:
alarm status = ALMx [10] = here x indicates the core group in this case, core is 1 hence alarm is generated in group 1, 10th bit.



case 2: address buffer overflow

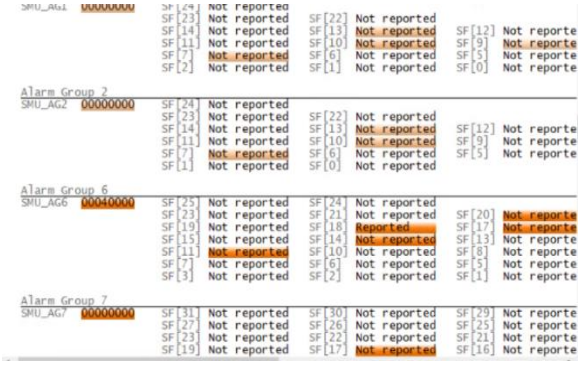
INPUT:
Error injected at DLMU in CORE 1
Type of error injected = address buffer overflow
Alarm should be generated at group 1, 9th & 10th bit

OUTPUT:
Alarm status = ALMx [9] & ALMx[10] = alarm is generated at group 1, 9th bit & 10th bit



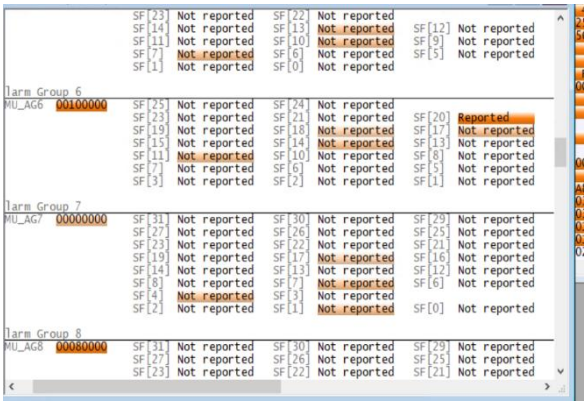
Case 3: safety mechanism
INPUT:
Error injected at CAN in CORE 1
Type of error injected = Safety mechanism
Alarm should be generated at 6th group, 18th bit

OUTPUT:
Alarm status = ALM6[18]- alarm is generated at group 6, 18th bit



Case 4: ram operation monitor
INPUT:
Error injected at DMA in CORE 1
Type of error injected = Ram operation monitor
Alarm should be generated at 6th group, 20th bit

OUTPUT:
Alarm status = ALM6[20]- alarm is generated at group 6, 20th bit



VII. CONCLUSION

Every software developer will be having concern to make system safer when some faults occur in that system. In order to give such safety mechanism for the mentioned type of errors in the SRAM, the mentioned design approach has been implemented.
With this discussion, we can say when the error or the fault appears in SRAM section then which will be monitored with microcontroller and soon the memory section will be detached from the fault section with the help of software component that won'taffect on other components. By handling this fault, overall performance of the system will also be improved as per the functional safety goals by handling the system faults.

REFERENCES

[1] 25th International Symposium on On-Line Testing and Robust SystemDesign (IOLTS)Felipe Augusto da Silva; Ahmet CagriBagbaba; Said Hamdioui;Christian Sauer, 2019
[2] "Mapping the Software Errors and Effects Analysis Method to ISO26262 Requirements for Software Architecture Analysis" by Paulo Victor Carvalho in IEEE International Symposium on Software Reliability Engineering Workshops, 2014
[3] Error Detection in SRAM -webpage
[4] Functional Safety-road vehicles: Supporting processes, International Standardization Organization Std, Nov. 2011.
[5] A. Nardi and A. Armato, "Functional safety methodologies, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), nov 2017.
[6] PRASANNKUMARY SHIVAYOGI" Safety Mechanism Implementation for Fault in the Memory" IEEE Mysore Sub Section International Conference (MysuruCon-2021) in association with IEEE Bangalore,2021
[7] "Effective Use of Fault Injection for AUTOSAR Safety Mechanisms", 11th European Dependable Computing Conference by Thorsten Piper, Stefan Winter, Neeraj Suri, 2018
[8] "Framework For Real- time Automotive Applications to Multicore Platform", 4th International Conference on Recent Trends on Electronics, Information, Communication Technology (RTEICT-2019), by N.P. Singh, Geetha Srinivasan, Priyanshi Gupta, MAY 17th 18th 2019
[9] Kogan, T., Abotbol, Y., Boschi, G., Harutyunyan, G., Kroul, I., Shaheen, H., Zorian, Y. (2017). Advanced functional safety mechanisms for embedded memories and IPs in automotive SoCs. In 2017 IEEE International Test Conference (ITC) (pp. 1-6). IEEE.
[10] AUTOSAR Administration, "AUTomotive Open System ARchitecture," http://www.autosar.org, 2014.
[11] R. Schneider, A. Kohn, K. Schmidt, S. Schoenberg, U. Dannebaum, J. Harnisch, and Q. Zhou, "Efficient Virtualization for Functional

Integration on Modern Microcontrollers in Safety-Relevant Domains," in SAE 2014 World Congress Exhibition, Apr. 2014.

[12] Tae-Wook Kim, Gun-Min Lee, and Jong-Chan Kim3 "AUTOSAR Runnable Scheduling for Optimal Tradeoff between Control Performance and CPU Utilization" 2018 18th International Conference on Control, Automation and Systems (ICCAS 2018).