Application Note

Deploying DDR Inline ECC on TDA4VE

Linjun Meng, Zekun Bai, Tommy Song *Central FAE*

ABSTRACT

The TDA4VE processor integrates DDR Inline ECC functionality that provides hardware-based single-bit error correction (SEC) and double-bit error detection (DED) to improve DDR memory reliability. This application note describes the recommended deployment process, including ECC configuration, DDR priming, and runtime enablement considerations. The document targets software developers, system architects, and functional safety engineers developing automotive, industrial, robotics, and other reliability-critical systems. Following the guidelines presented in this document helps provide reliable ECC operation, simplifies software integration, and reduces the risk of memory corruption caused by transient DDR errors.

Table of Contents

List of Figures

List of Tables

Trademarks

1 Introduction

Modern automotive and industrial systems increasingly rely on large external DDR memory devices to support advanced applications such as autonomous driving, machine vision, sensor fusion, and edge AI processing. As DDR capacity grows, memory reliability becomes an important system design consideration.

Transient memory faults caused by radiation effects, voltage disturbances, signal integrity issues, or electromagnetic interference can lead to data corruption. In safety-critical applications, undetected memory corruption can result in system malfunction or unexpected behavior.

To improve DDR reliability, the TDA4VE DDR subsystem integrates DDR Inline Error Correction Code (ECC) functionality. The Inline ECC engine operates transparently within the DDR controller and provides:

- Single-bit error correction (SEC)
- Double-bit error detection (DED)
- Automatic ECC generation during memory writes
- Automatic ECC checking during memory reads
- Runtime error reporting through ESM

Although ECC protection is implemented in hardware, successful deployment requires a specific initialization sequence. Before runtime ECC checking can be enabled, all ECC-protected memory locations must first be populated with valid ECC information. This initialization procedure, commonly referred to as DDR priming, is a critical requirement for proper ECC operation.

This application note describes the recommended DDR Inline ECC deployment flow for TDA4VE devices, including ECC region planning, DDR priming, DRU-based implementation, and SBL integration.

In addition to describing the deployment methodology, this application note references a validated software implementation that demonstrates the complete DDR Inline ECC enablement flow on TDA4VE devices. The implementation includes ECC configuration, DRU resource allocation, DDR priming, and SBL integration, and can serve as a reference for customer software development.

2 Detailed Description

2.1 DDR Inline ECC Overview

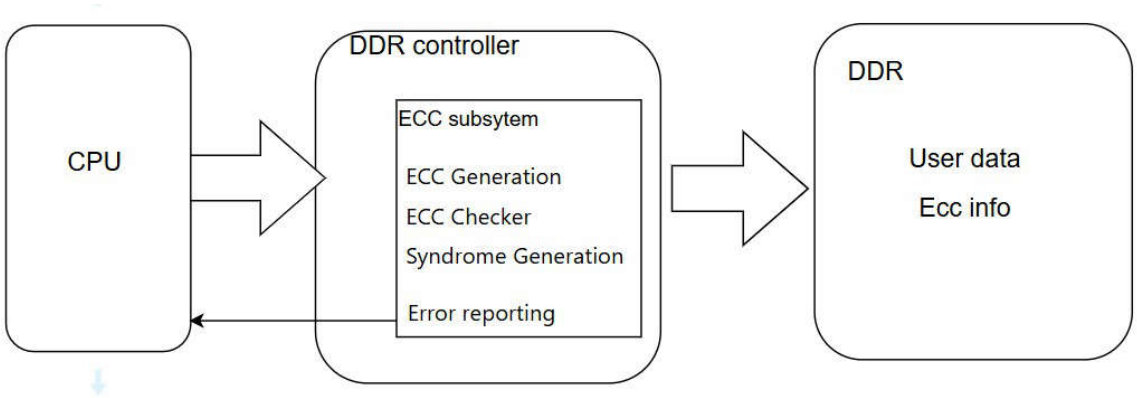

Figure 2-1. TDA4VE DDR Inline ECC Architecture

This architecture provides continuous protection against memory corruption while minimizing software overhead.

DDR Inline ECC operates transparently as part of the DDR subsystem.

During memory write operations:

1. User data is written to DDR.
2. The ECC engine calculates ECC parity information.
3. ECC information is stored automatically alongside protected data.

During memory read operations:

1. Data and ECC information are retrieved.
2. The ECC engine performs syndrome checking.
3. Single-bit errors are corrected automatically.
4. Double-bit errors are detected and reported.

The mechanism provides continuous runtime protection without requiring software involvement during normal operation.

2.1.1 Benefits of DDR Inline ECC

- Improved system reliability
- Enhanced resilience to transient memory errors
- Support for functional safety requirements
- Reduced risk of silent memory corruption
- Hardware-based protection with minimal runtime overhead

2.2 ECC Protection Region Planning

DDR Inline ECC only protects memory regions explicitly configured for ECC operation. Therefore, memory layout planning must be completed early during the software architecture phase.

Typical deployment strategies include:

- Full DDR Protection: All DDR memory is protected by ECC
  - Advantages:
    - Maximum fault coverage
    - Simplified software architecture
    - Recommended for safety-critical systems
  - Disadvantages:
    - Increased memory overhead
    - Longer DDR priming time

- Partial DDR Protection: Only selected memory regions are protected
  - Examples
    - Safety-critical application data
    - Neural-network parameters
    - Control algorithms
    - Diagnostic buffers
  - Advantages
    - Reduced memory overhead
    - Faster initialization
  - Disadvantages
    - Limited fault coverage

2.3 ECC Capacity Considerations

DDR Inline ECC reserves part of the DDR address space for ECC information storage. Consequently, not all physical DDR capacity remains available to applications.

When planning memory usage, system designers must account for:

- ECC metadata storage requirements
- Application memory requirements
- Boot-time priming duration

For systems targeting functional safety certification, full-memory ECC protection is generally recommended unless application requirements dictate a different approach.

2.4 ECC Configuration Guidelines

The DDR Technical Reference Manual (TRM) provides detailed register descriptions and programming information for DDR Inline ECC.

Rather than repeating register definitions, this application note focuses on deployment recommendations and configuration sequencing.

The following guidelines are recommended:

Table 2-1. ECC Configuration Guidelines

Configuration Item	Recommendation	Purpose
ECC Region Configuration	Configure before ECC enablement	Define protected DDR areas
ECC Generation	Enable before priming	Generate valid ECC information
ECC Checking	Enable after priming completes	Avoid false ECC faults
Read-Modify-Write (RMW)	Keep enabled	Maintain ECC consistency during partial writes

2.4.1 ECC Generation Phase

Initially, ECC generation is enabled while ECC checking remains disabled.

During this phase:

- Data writes generate ECC information
- Memory is initialized
- DDR priming is performed

Runtime error checking is not yet active.

2.4.2 ECC Protection Phase

After DDR priming completes:

- ECC checking is enabled
- Runtime SECDED protection is enabled.
- ECC events are reported through ESM

Once the protected memory regions have been initialized, ECC checking can be enabled. The system then enters the ECC protection phase, where all accesses within ECC-protected regions benefit from continuous SECDED protection.

Separating these two phases is essential for reliable system start-up and helps prevent false ECC fault reporting during initialization.

2.5 Why DDR Priming Is Required

Following DDR initialization, memory contents are undefined.

Although memory locations can contain random data, valid ECC information does not yet exist for those locations.

If ECC checking is enabled immediately after DDR initialization, read operations to protected memory locations can trigger false ECC errors because valid ECC information has not yet been generated.

This behavior is often misinterpreted as a DDR hardware issue, when in fact the memory has simply not been initialized with valid ECC data.

DDR priming eliminates this condition by writing known data patterns across all ECC-protected memory locations. During these writes, the DDR controller automatically generates valid ECC information.

2.6 DDR Priming Using DRU

For large DDR devices, CPU-based memory initialization can significantly increase system startup time.

The recommended implementation uses the Data Routing Unit (DRU) to accelerate the priming process.

Benefits of DRU-assisted priming include:

- Minimal CPU involvement
- High memory bandwidth utilization
- Reduced boot time
- Scalable implementation across DDR capacities

The reference implementation uses a zero-filled source buffer located in MSMC memory. DRU transfers repeatedly write this data across the protected DDR address range.

Each write operation automatically causes the DDR controller to generate corresponding ECC information.

As a result:

- DDR contents become initialized
- ECC information becomes valid
- Runtime ECC checking can subsequently be enabled safely

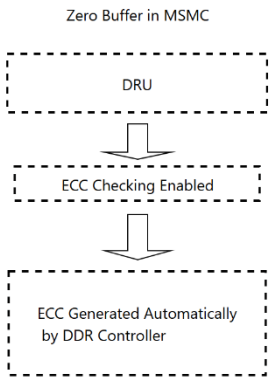

Figure 2-2. DRU-Based DDR Priming

The Data Routing Unit (DRU) transfers a predefined data pattern from a source buffer to the ECC-protected DDR address space. As each destination address is written, the DDR controller automatically generates the corresponding ECC information.

Compared with CPU-based memory initialization methods, DRU-based priming significantly reduces processor involvement and improves start-up performance, particularly when large DDR memories are protected by ECC.

This approach provides an efficient and scalable method for initializing ECC-protected DDR regions.

2.7 DRU Resource Allocation

In the default Processor SDK configuration, MCU1_0 does not own the DRU resources required for DDR priming.

To enable DRU-based priming, BoardCfg Resource Management configuration must be updated so that DRU resources are assigned to MCU1_0.

The reference implementation includes:

- DRU channel allocation
- BoardCfg modifications
- DRU initialization code
- DDR priming implementation

The required modifications are typically made within:

- sciclient_defaultBoardcfg_rm.c
- sciclient_defaultBoardcfg_tifs_rm.c

2.8 SBL Integration

The recommended deployment strategy integrates DDR Inline ECC initialization into the Secondary Boot Loader (SBL). Performing ECC initialization during system boot verifies that all ECC-protected memory regions contain valid ECC information before application software begins execution.

Integrating the initialization process into the SBL provides several advantages:

- Consistent ECC configuration for all applications.
- Centralized management of DDR initialization and ECC enablement.
- Elimination of application-specific ECC initialization code.
- Simplified system integration and validation.

Within the SBL, DDR priming is performed immediately after DDR initialization and ECC region configuration. Once priming is complete, ECC checking is enabled before control is transferred to the application image. This approach makes sure that application software only accesses DDR memory that already contains valid ECC information.

2.9 Recommended Initialization Flow

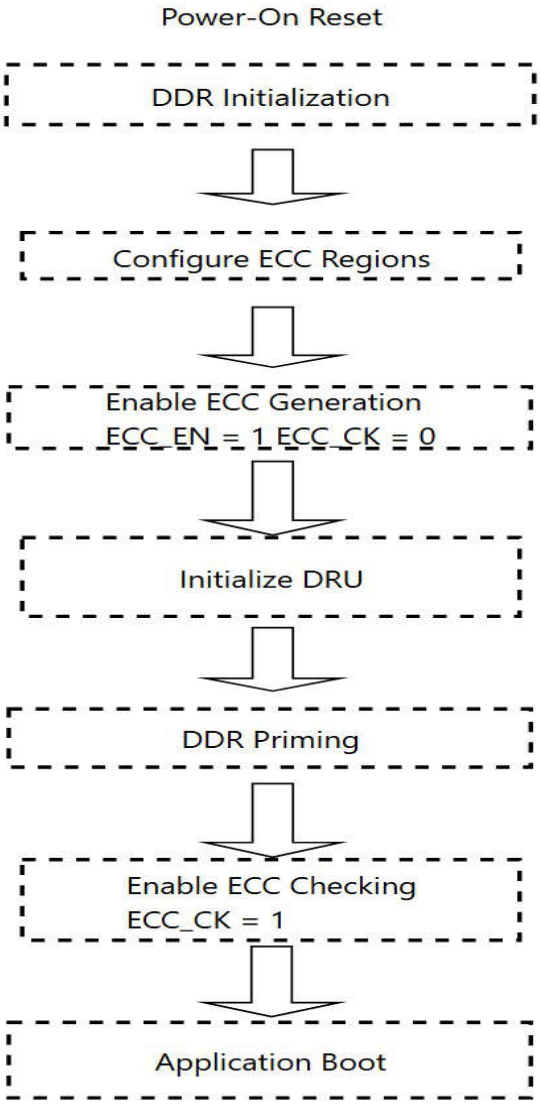

Figure 2-3. Recommended TDA4VE DDR Inline ECC Deployment Flow

Figure 2-3 illustrates the recommended initialization sequence for deploying DDR Inline ECC on TDA4VE devices.

The recommended initialization flow consists of the following steps:

1. Initialize the DDR subsystem.
2. Configure ECC-protected memory regions.
3. Enable ECC generation.
4. Keep ECC checking disabled.
5. Initialize DRU resources.
6. Perform DDR priming across all ECC-protected memory regions.
7. Verify completion of all DRU transfer operations.
8. Enable ECC checking.
9. Configure ESM reporting and system-level fault handling as required.
10. Transfer control to the application software.

This sequence separates ECC information generation from runtime ECC validation. By verifying that valid ECC information exists throughout all protected memory regions before ECC checking is enabled, the system avoids false ECC fault reports during start-up and enters runtime operation with full SECDED protection enabled.

2.10 Performance Evaluation

The reference implementation was evaluated using a 4GB DDR configuration.

DDR Capacity	Priming Time
4GB	Approximately 147ms

The result demonstrates that DRU-based initialization can efficiently generate ECC information across the entire protected DDR address space while maintaining acceptable start-up overhead.

Actual priming time can vary depending on:

- DDR frequency
- DRU configuration
- Protected memory size
- System initialization sequence

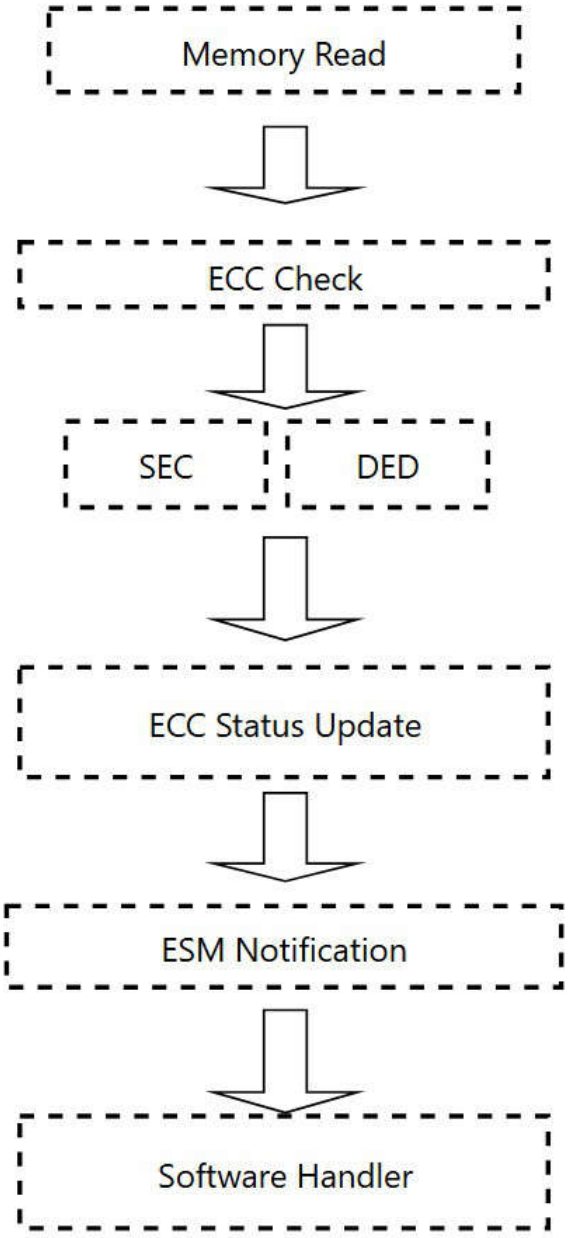

Figure 2-4. ECC Error Reporting Flow

Figure 2-4 illustrates the ECC error reporting path.

Correctable errors (CE) are automatically corrected by the DDR subsystem, while uncorrectable errors (UE) are reported through the ESM for software handling.

2.11 Troubleshooting Common DDR Inline ECC Deployment Issues

2.11.1 ECC Checking Enabled Before Priming

Symptoms:

- ECC faults immediately after boot
- Unexpected UECC reports

Root cause:

- Memory locations contain invalid ECC information

Solution:

- Complete DDR priming before enabling ECC checking

2.11.2 Incorrect ECC Region Configuration

Symptoms:

- Memory not protected as expected
- Missing ECC event reporting

Root cause:

- ECC region boundaries configured incorrectly

Solution:

- Verify region configuration and memory map consistency

2.11.3 Read-Modify-Write Disabled

Symptoms:

- ECC mismatches following partial writes
- Intermittent ECC faults

Root cause:

- ECC information is not regenerated correctly for sub-word write operations

Solution:

- Verify RMW functionality remains enabled whenever DDR Inline ECC is enabled

2.12 Validation and Debug Considerations

After deployment, system-level validation is recommended to confirm proper ECC operation.

Typical validation activities include:

- Single-bit error correction verification
- Double-bit error detection verification
- ECC fault injection testing
- ESM event validation

For detailed ECC verification procedures, refer to the *DDR Inline ECC Error Injection and Validation Guide for Jacinto™ 7 and Sitara™ Processors* application note.

For runtime error analysis and syndrome interpretation, refer to the *DDR Inline ECC Error Address Decoding on Jacinto™ 7 and Sitara™ Processors* application note.

3 Summary

DDR Inline ECC is a key reliability feature of the TDA4VE platform, providing hardware-based single-bit error correction and double-bit error detection for external DDR memory. Successful deployment requires proper ECC protection planning, controlled ECC enablement, DDR priming, and software integration.

This application note presented a recommended deployment methodology based on SBL and DRU, enabling valid ECC information to be generated during system start-up before runtime ECC checking is activated. Following the initialization sequence described in this document helps provide reliable ECC operation while minimizing start-up overhead and integration effort. The approach can be adapted to different DDR capacities, memory layouts, and application requirements while providing a solid foundation for subsequent ECC validation and debug activities.

4 References

1. Texas Instruments, J721S2, TDA4AL, TDA4VL, TDA4VE, and AM68A Technical Reference Manual (TRM), technical reference manual.
2. Texas Instruments, Processor SDK RTOS Documentation and DDR Inline ECC Reference Patch

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025