G4M-eSRAM to AHB Verification and Validation Report DRAFT

Revision: 0.2

Date: September 30th, 2014

Table of Contents

Table of Figures

Tables

Revision History

Version	Date	Modified by	Changes
0.1	20th July 2012	SM	Initial Version
0.2	30th Sept 2014	SW	Report Version

Glossary

Term	Description
LIN	Local Interconnect Network
Use Case	A specific configuration of a use model.
Use Model	A reference platform designed from configurable h/w and s/w components, which is targeted at a focused application. The use model demonstrates one of several possible ways of integrating IPs to build a system and one of several ways in which it could be used by the customer.
ISR	Interrupt Service Routine – Subroutine executed as a result of an IRQ
MMUART	Multi-Mode Universal Asynchronous Receiver Transmitter
G4	4th Generation Flash chip from Microsemi
Frame	All LIN information is transmitted packed as frames; a frame consists of a header and a response

REFERENCES

1. A4P5000 Chip-Level SAC Spec.pdf
2. G4M-ESRAMTOAHB_SAC
3. G4M_ESRAMTOAHB_PDV_PLAN.docx
4. G4M_ESRAMTOAHB_PDV_Result.docx

1 Introduction

This document explains validation result of various A4P5000 eSRAM to AHB PDV UseCases executed on Validation Board.

1.1 Purpose and Scope

The purpose of this document is to describe the G4 eSRAM to AHB interface as well as its plans and tests that are performed as a part of silicon validation and verification. All UseCases are elaborated upon and reports of issues during verification are documented. This document is to provide a holistic view of the verification and validation process

1.2 Standards and Requirements

The Design team and the Verification and Validation Team have different reporting structures that are independent of one another.

2 G4M eSRAM to AHB background

There are two blocks of 32kB eSRAM present on A4P5000, giving a total of 64kB. Having the eSRAM arranged as two separate blocks allows the user to take advantage of the Harvard architecture of the Cortex-M3 processor. For example, code could be located in one eSRAM, whilst data, such as the stack, could be located in the other.
Another possibility would be that the user may allocate one of the eSRAM blocks for use by a communications peripheral, such as the Ethernet MAC or USB.
The eSRAM allows low latency access by MSS Masters. The eSRAM contains 2-bit error detection and 1-bit error correction. Because the EDAC information is associated with a word of eSRAM, any byte or half-word write to eSRAM involves a read-modify-write cycle, when EDAC is enabled. It is also necessary for the full eSRAM to be initialized to all zeros after power-up, in order to ensure that the EDAC information is correctly initialized. When EDAC is disabled, the SRAM usually used to store EDAC data may be recycled to be used as an extra 16kB of eSRAM. Byte, half-word and word accesses to recycled eSRAM take one, two and four accesses respectively.

The eSRAM to AHB internal architecture is as shown below.

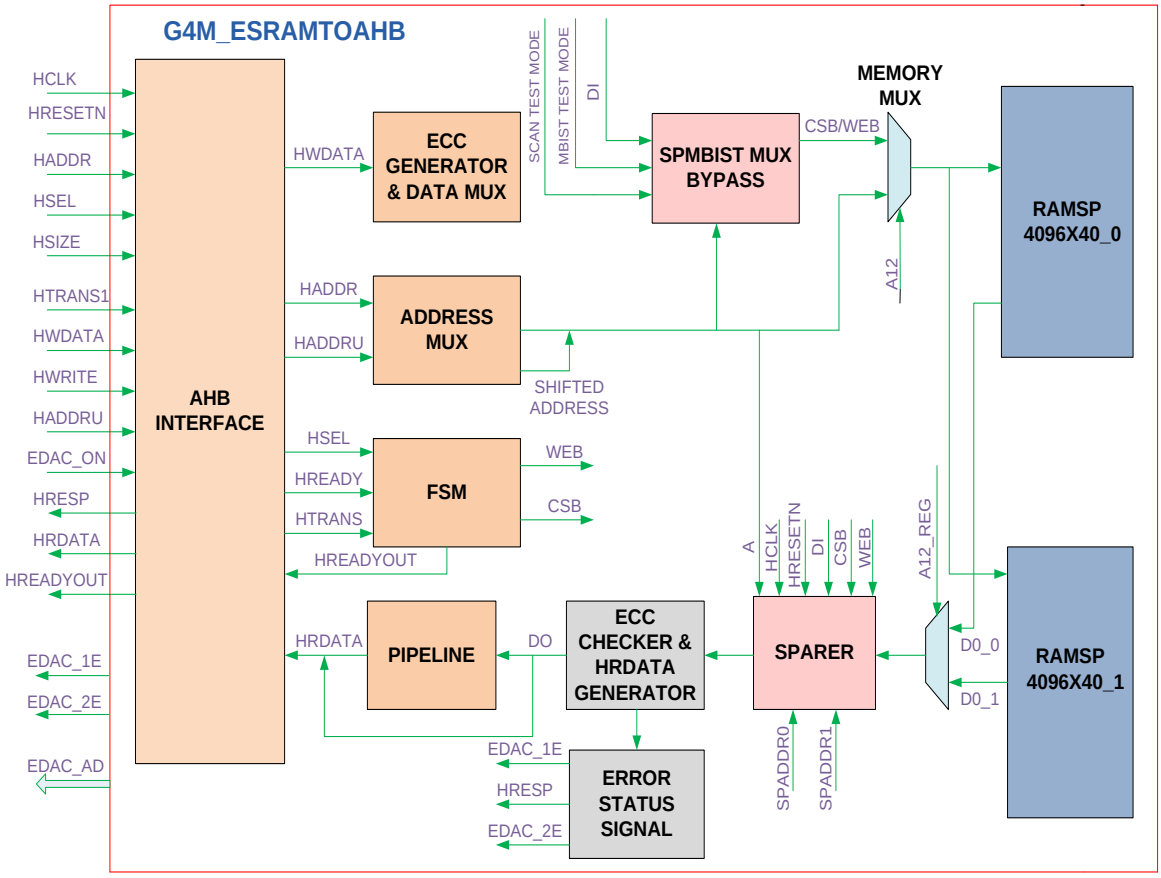


Figure 2-1 : ESRAMTOAHB ARCHITECTURE

2.1 Features of eSRAM to AHB

2.1.1 Modes of Operation

- EDAC Mode
  - It supports one bit error correction.
  - It supports two bit error detection.
  - It has 32K Bytes of EDAC protected RAM.
  - In this mode, read access is zero wait state if not following a write immediately.
  - In this mode, read access is one wait state if it is immediately following a write.
  - The write access is one wait state for byte and half write accesses.
  - Here HRESP is generated for 2-bit error detection.
    - HRESP will cause an exception at the M3 requested the instruction or data fetch.
    - The HRESP event will also be detected in the AHB switch.
  - It has two counters to give Error statistic and the counters are cleared by the clear bits.
- NON-EDAC Mode
  - It has one bank of 32K Bytes of RAM
  - It has second bank of 8K Bytes of RAM
  - 32K Byte Bank has the following features:
    - Zero wait state read access if it is not following a write cycle.
    - One wait state read access if it is immediately following a write cycle.
    - Zero wait state write access.
  - 8K Byte Bank has the following features :
    - Zero wait state read access for a byte if it is not following a write cycle.
    - Zero wait state read access for half word if it is not following a write cycle.
    - One wait state read access for word if it is not following a write cycle.
    - Zero wait state write access for a byte.
    - Zero wait state write access for half word.
    - One wait state write access for word.
- PIPELINE Mode
  - An optional pipeline may be enabled on the Read data bus, this will add a clock cycle to all read cycles. When enabled the FMAX of the system will be higher. This pipeline is assumed to be on for 200/166MHz operation. When turned off system performance is expected to be around 133/100 MHz

3 Use Models & Use Cases

3.1 UseCases

<table>
<tr><th>UseModel</th><th>Feature under validation</th><th>Description</th></tr>
<tr><td>UseCase1</td><td>ScratchPad</td><td>Remapping is done to eNVM so that the application is running from eNVM.
Perform write access to ERAM0 and ESRAM memory
Make read access to the written location and verify.
Execute the usecase for both EDAN ON and OFF case
Note that for EDAC OFF mode, the 16KB memory of EDAC memory is also available and read/write sequence is exercised for this EDAC memory.
<table>
<tr><th><u>Data pattern</u></th><th><u>Access Mode</u></th></tr>
<tr><td>1. Sequential incrementing
2. Walking ones
3. Walking zeros
4. Random Address Random Data</td><td>1. word
2. Half Word
3. Byte</td></tr>
</table></td></tr>
<tr><td>UseCase2</td><td>Image Loading</td><td>Remapping to ESRAM and running the application out of ESRAM
Validation is done by toggling LEDs and printing UART message</td></tr>
<tr><td>UseCase3a</td><td>Dhrystone: ROM and RAM in eSRAM0
ESRAM_PIPELINE=OFF</td><td>In this case ESRAMFWREMAP bit is set to 1 to remap ESRAM0 to 0x00 location (To run the application code from ESRAM0).
Both the application code and RAM region resides in ESRAM0.
G4Main Timer peripheral is used to measure the time taken to execute 50000 Dhrystone runs and DMIPS/MHz shall be calculated for this case
ESRAM_PIPELINE=OFF</td></tr>
<tr><td>UseCase3b</td><td>Dhrystone: ROM in eSRAM0 and RAM in eSRAM1
ESRAM_PIPELINE=OFF</td><td>The above usecase is repeated with the RAM region in ESRAM1.
Calculate the DMIPS/MHz for this case and compare it with UseCase1a
ESRAM_PIPELINE=OFF</td></tr>
<tr><td>UseCase3c</td><td>Dhrystone: ROM and RAM in eSRAM0
ESRAM_PIPEL</td><td>Repeat usecase1a with ESRAM_PIPELINE=ON</td></tr>
</table>

UseModel	Feature under validation	Description
	INE=ON	
UseCase3d	Dhrystone: ROM in eSRAM0 and RAM in eSRAM1 ESRAM_PIPELINE=ON	Repeat usecase1b with ESRAM_PIPELINE=OFF
UseCase4	Read/write access time	1. Make write access to 1000 SRAM sequential address locations. Measure the time taken to execute the test (using timer). 2. Make read access to 1000 SRAM sequential address locations. Measure the time taken to execute the test (using timer). 3. Make write followed immediately by read access to 1000 SRAM sequential address locations. Measure the time taken to execute the test (using timer). Compare the time taken for case (3) with the time taken for case (1) + case(2) Execute the test for all the pipeline & EDAC combinations mentioned in Table 6 F2DSS_TIMER to measure access time: Set the timer in one-shot mode load the timer counter with its maximum value (timer_init). Once the eSRAM access is done read the F2DSS_counter value (timer_cnt). (timer_init)- (timer_cnt) will give the access time value.
UseCase5	EDAC – 1 bit & 2 bit error detection and correction	Set the ESRAM EDAC mode to '1' and verify the single bit and two bit error detection. Read the 1bit and 2bit error counters status registers/ clear counter. Verify the interrupts. TBD: implementation of test

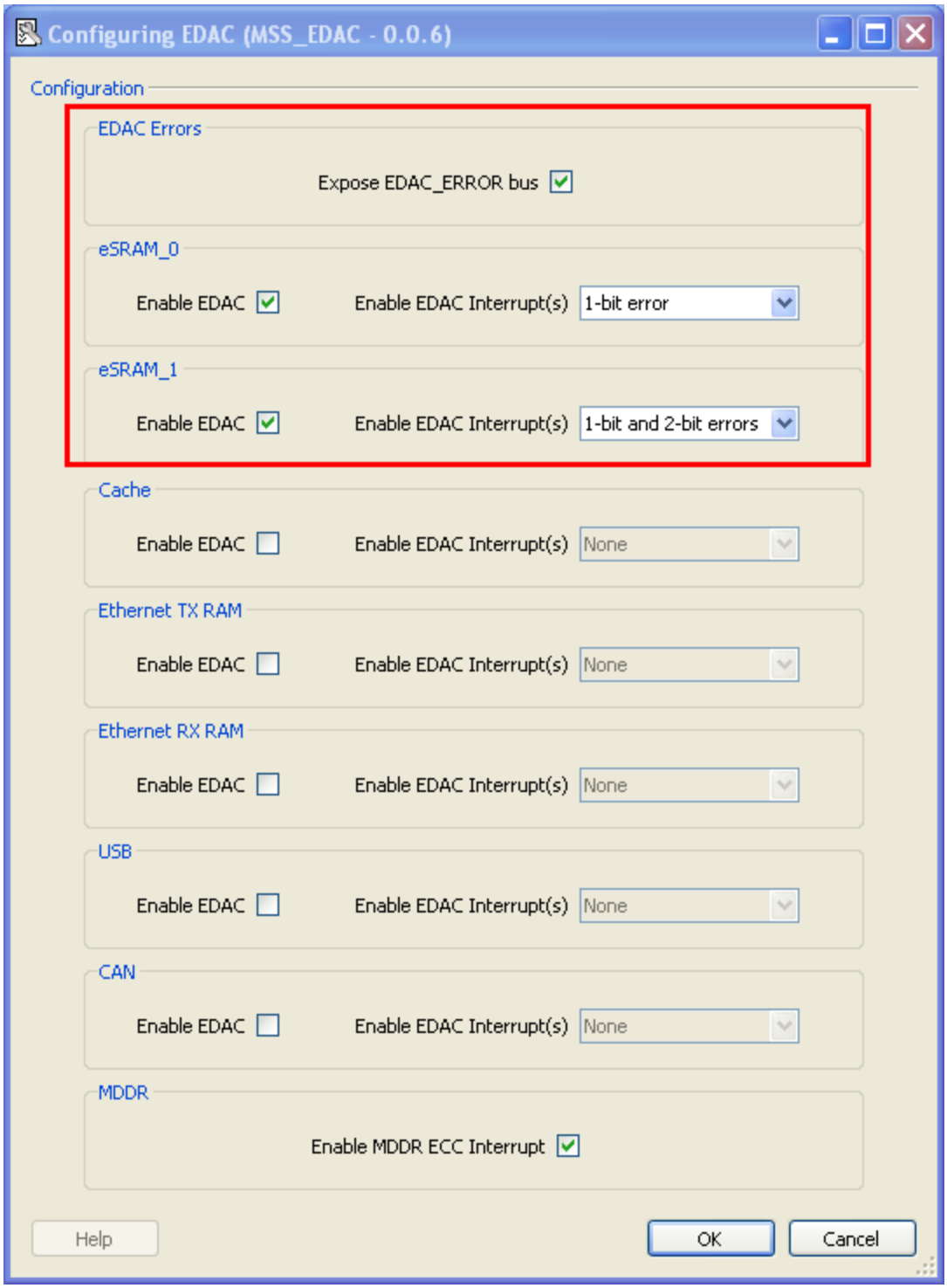
Table 1 List of UseCases

PIPELINE	eSRAM	OPERATION	SIZE	No. of WAIT STATES	No. of WAIT STATES (Write before)
ENABLE	32KB RAM EDAC_ON MODE	WRITE	32-Bit	0	0
			16-Bit	1	2
			8-Bit	1	2
		READ	32-Bit	1	2
			16-Bit	1	2
			8-Bit	1	2
	32KB RAM EDAC_OFF MODE	WRITE	32-Bit	0	0
			16-Bit	0	0
			8-Bit	0	0
		READ	32-Bit	1	2
			16-Bit	1	2
			8-Bit	1	2
	8KB RAM	WRITE	32-Bit	1	1
			16-Bit	0	0
			8-Bit	0	0
		READ	32-Bit	2	3
			16-Bit	1	2
			8-Bit	1	2
DISABLE	32KB RAM EDAC_ON MODE	WRITE	32-Bit	0	0
			16-Bit	1	2
			8-Bit	1	2
		READ	32-Bit	0	1
			16-Bit	0	1
			8-Bit	0	1
	32KB RAM EDAC_OFF MODE	WRITE	32-Bit	0	0
			16-Bit	0	0
			8-Bit	0	0
		READ	32-Bit	0	1
			16-Bit	0	1

			8-Bit	0	1
	8KB RAM	WRITE	32-Bit	1	1
			16-Bit	0	0
			8-Bit	0	0
		READ	32-Bit	1	2
			16-Bit	0	1
			8-Bit	0	1

Table 2 Tabular column for modes of operation in eSRAM

3.2 Tool Flow


Configuring EDAC (MSS_EDAC - 0.0.6)
Configuration
EDAC Errors
Expose EDAC_ERROR bus
eSRAM_0
Enable EDAC
Enable EDAC Interrupt(s)
1-bit error
eSRAM_1
Enable EDAC
Enable EDAC Interrupt(s)
1-bit and 2-bit errors
Cache
Enable EDAC
Enable EDAC Interrupt(s)
None
Ethernet TX RAM
Enable EDAC
Enable EDAC Interrupt(s)
None
Ethernet RX RAM
Enable EDAC
Enable EDAC Interrupt(s)
None
USB
Enable EDAC
Enable EDAC Interrupt(s)
None
CAN
Enable EDAC
Enable EDAC Interrupt(s)
None
MDDR
Enable MDDR ECC Interrupt
Help
OK
Cancel

4 Equipment Used

4.1 Hardware

S.No	Equipment/Board	Qty	Remarks/Serial Number
1	G4M Validation Board	1	DVP-102-000304-001-RevA(labelled-3) (Socket Board) With Silicon labelled 2
2	DDR2 Daughter Board	1	DDR2-DB (labelled- 2)
2	EWARM Debugger	1	S/No:158006667
3	Flashpro4	1	Labelled-3
4	USB to Micro USB cable	1	On G4M Validation board UART0 is connected via fabric to the PC via USB cable.
5	DC Power supply	1	FSP200-50PL-B (12V 5V) is used to power-up the G4M validation board
6	PC	1	Stand-alone PC wxp-svg-09

Table 3 Hardware Equipment used

4.2 Software

S.No	Tools	Version	Remarks
1	Libero Capture	10.9.0.13	
2	MSS versions	0.0.641	
3	EWARM(IAR)	6.0	

Table 4 Software Equipment used

5 Use Models & Use Cases Test Phase

5.1 UseCase1- Scratchpad

5.1.1 Deviation from the Validation Plan

Remapping is done to DDR instead of eNVM

5.1.2 Implementation Guideline

The purpose of this test is to access 32KB ESRAM0 and ESRAM1.

It also verifies that the 8KB EDAC memory is accessible when the EDAC is OFF.

No extra jumper settings are required to execute this usecase. DDR is configured to be in operational mode and remapping is done to DDR as shown in snapshots below.

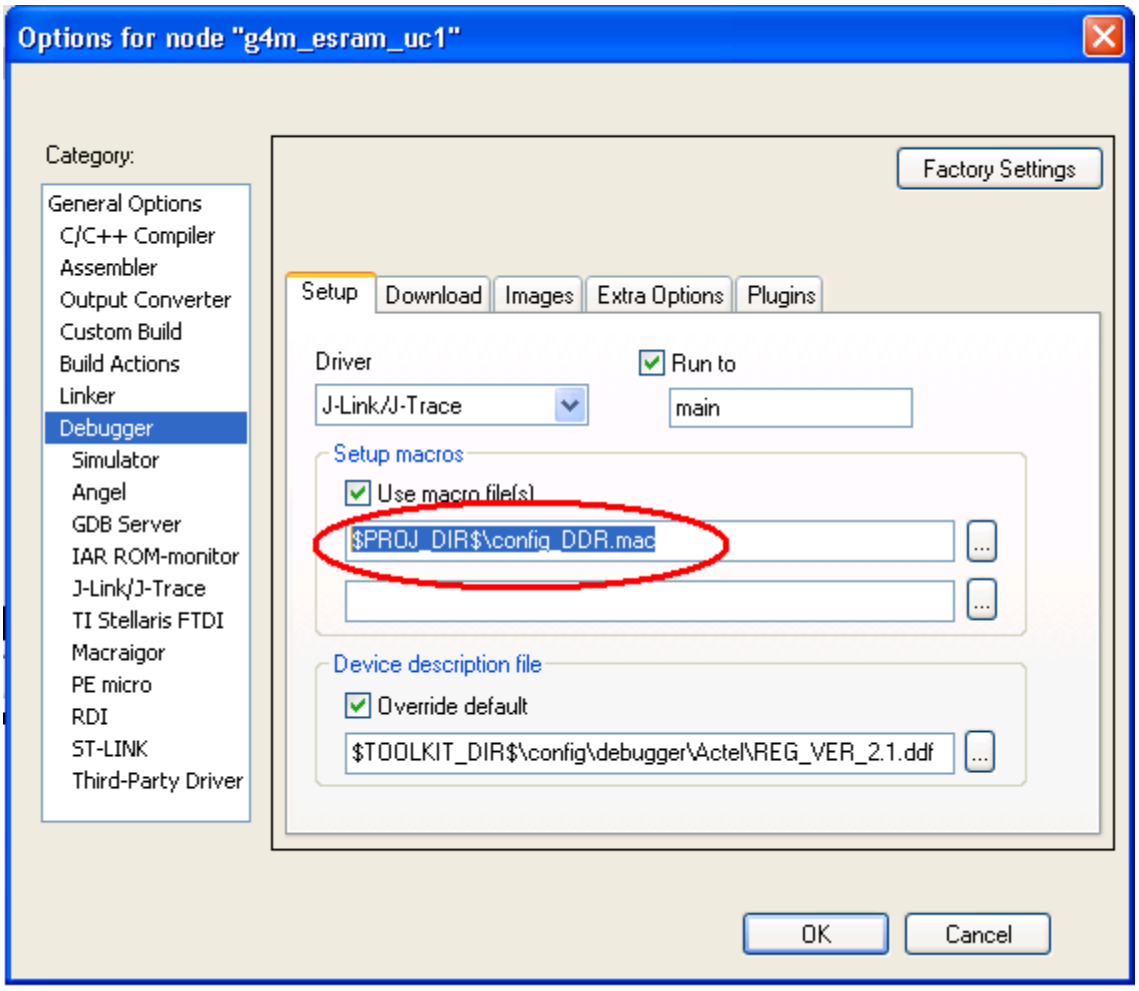


Figure 2 Macro to configure DDR and Remapping DDR to 0x00 location

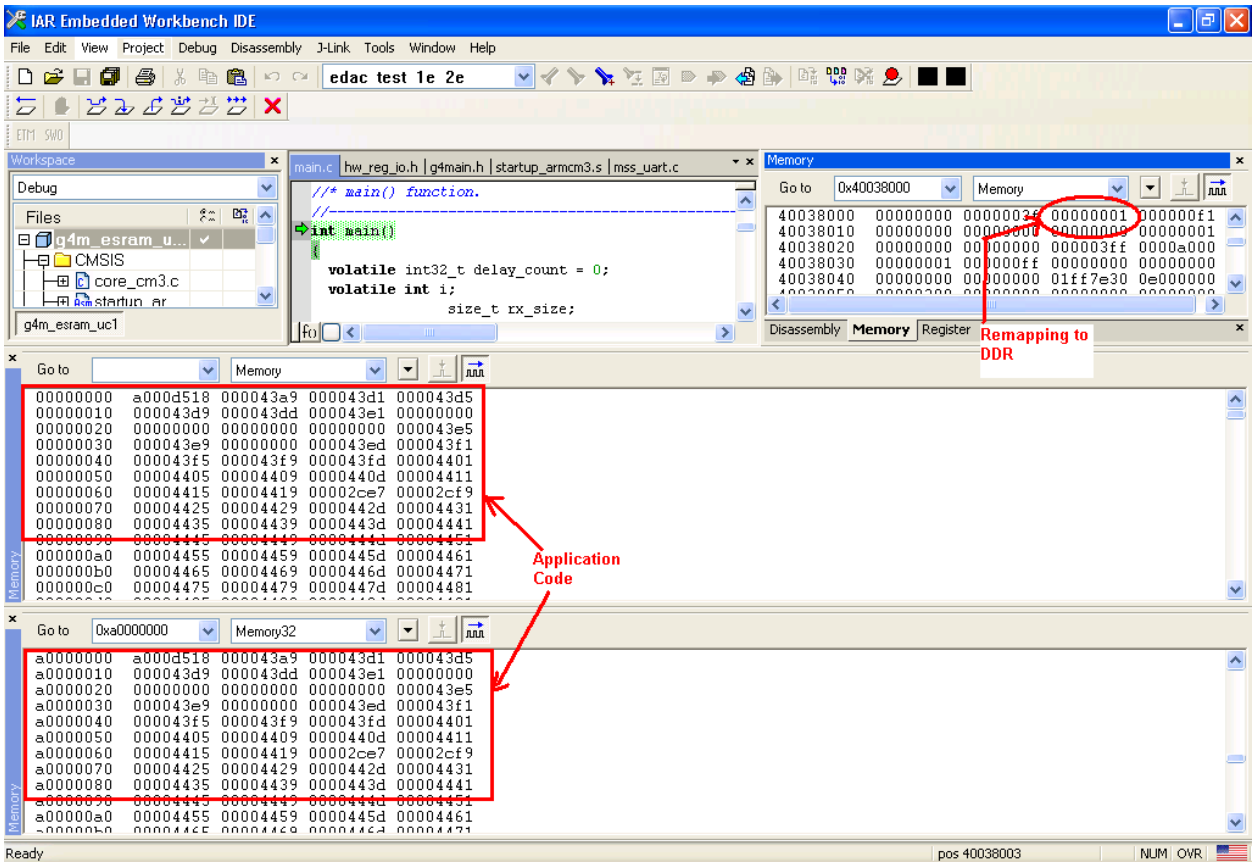


Figure 3 Remapping done to DDR and application code loaded

The firmware used to execute this usecase is menu driven which gives user following option to select:

- ESRAM0 or ESRAM1 selection
- EDAC enable/Disable option
- Byte, Half-Word and Word access selection.
- Data pattern selection (sequential, walking-1, walking-0, random pattern)

5.1.3 Issues

None

5.1.4 PASS or FAIL criterion

Access is verified for ESRAM by reading the DATA written to memory and comparing it against the expected data and the result of the test is displayed on UART terminal.

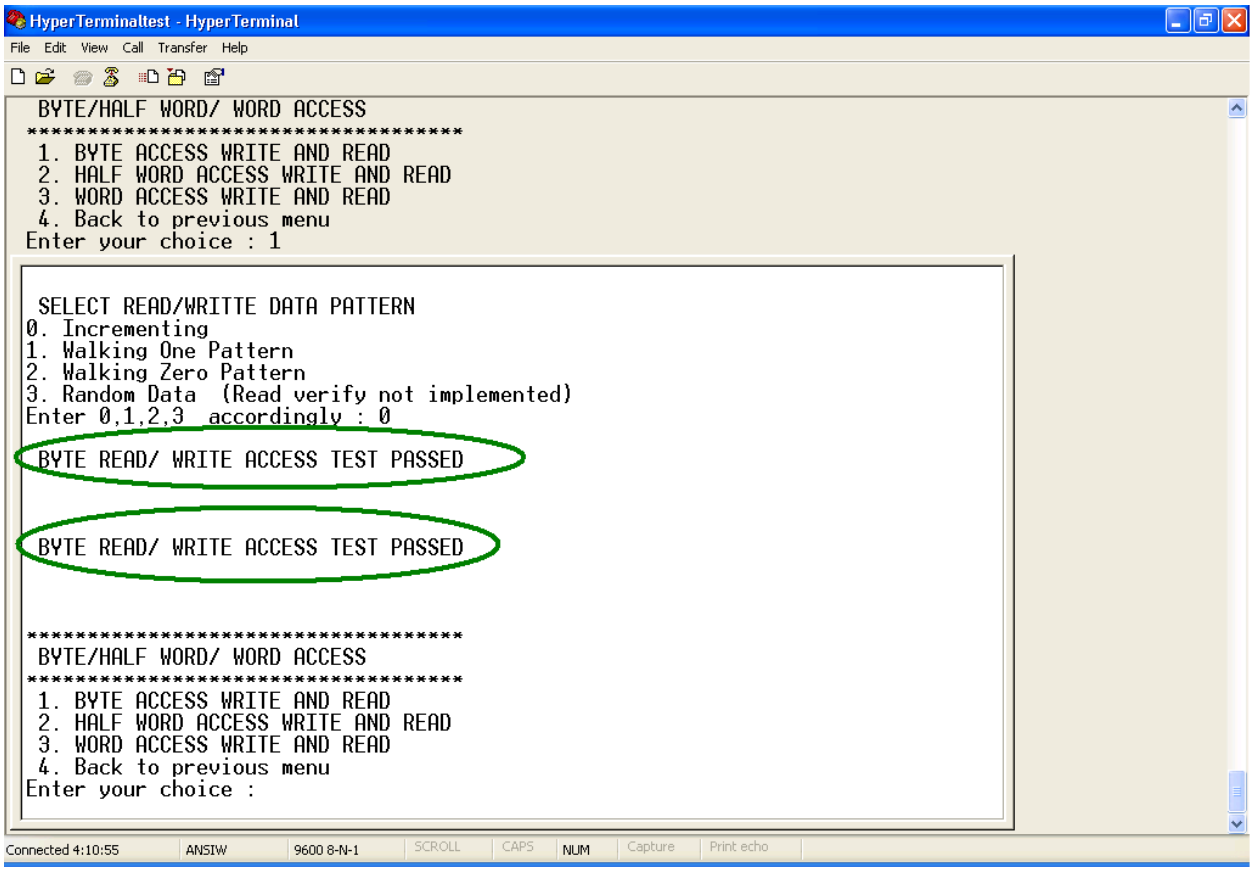


Figure 4 Hyper terminal snapshot for usecase1

5.1.5 Observation

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/results/G4M_ESRAM_UC1.TXT

5.1.6 SVN Database

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/design/hw/G4M_ESRAM_UC1.zip

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/design/fw/G4M_ESRAM_UC1.zip

5.2 UseCase2- Image Loading

5.2.1 Deviation from the Validation Plan

None

5.2.2 Implementation Guideline

Remapping to both ESRAM0 and ESRAM1 are validated by loading the application code to the respective memory and printing the UART display message and the displaying the remapping registers values.

Remapping is done to ESRAM using the mmuart.mac initialization file as shown in IAR option below:

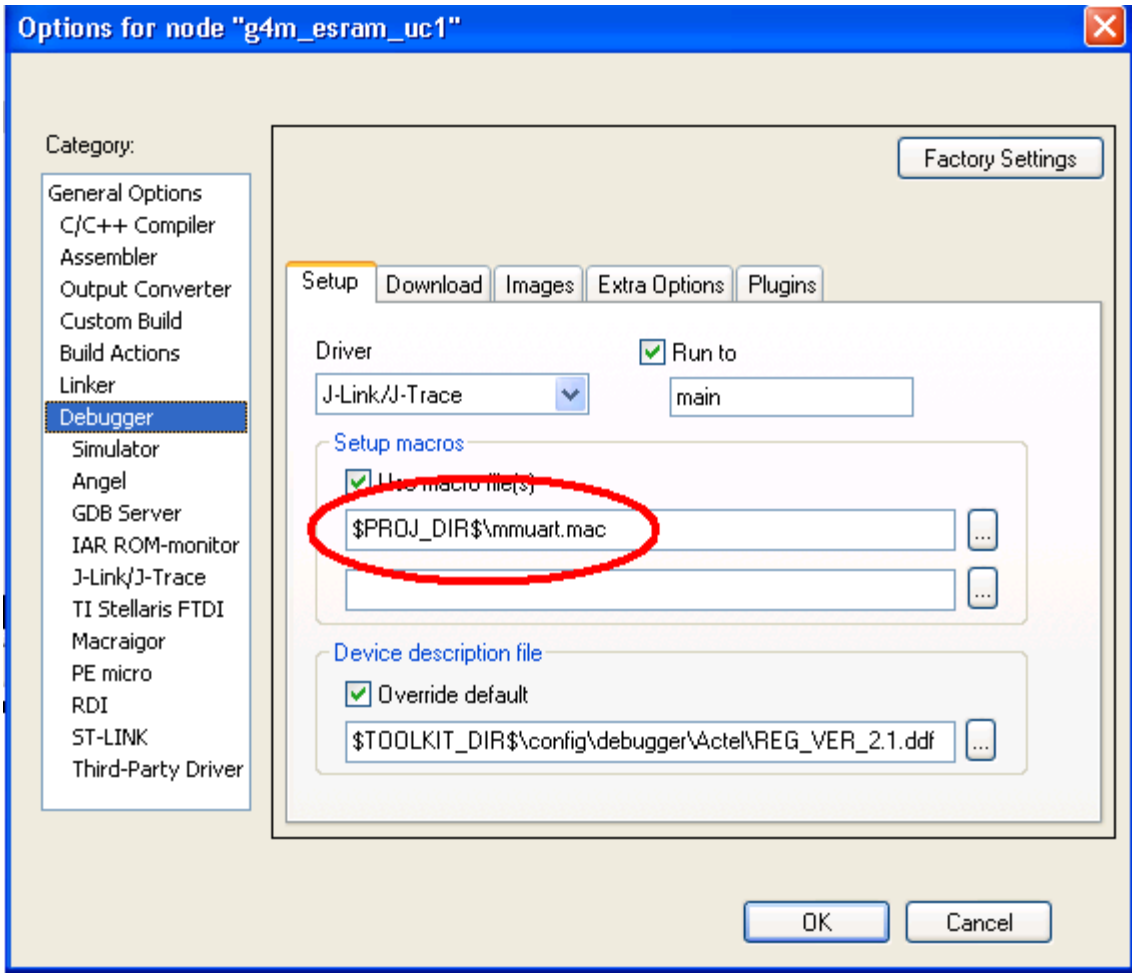


Figure 5 Remapping to ESRAM

ESRAM0 Remapping:

Mmuart.mac contents →

```
execUserPreload()
{
   __message "Remapping ESRAM to address 0x00000000\n";

__writeMemory32( 0x0000, 0x4003806C, "Memory"); //Disable WDOG
__writeMemory32( 0x0000, 0x40038010, "Memory"); //Disable ENVM REMAP
__writeMemory32( 0x0001, 0x40038000, "Memory"); //Enable ESRAM
__writeMemory32( 0x0000, 0x40038008, "Memory");//Disable DDR
}
```

ESRAM1 Remapping:

Mmuart.mac contents →

```
execUserPreload()
```

```
{
   __message "Remapping ESRAM to address 0x00000000\n";

__writeMemory32( 0x0000, 0x4003806C, "Memory"); //Disable WDOG
__writeMemory32( 0x0000, 0x40038010, "Memory"); //Disable ENVM REMAP
__writeMemory32( 0x0003, 0x40038000, "Memory"); //Enable ESRAM
__writeMemory32( 0x0000, 0x40038008, "Memory");//Disable DDR
}
```

5.2.3 Issues

None

5.2.4 PASS or FAIL criterion

Copied from UART terminal Display:

ESRAM0 Remapping

```
************************************
G4M-ESRAM UseModel:
************************************
UseCase1: ESRAM READ/WRITE ACCESS
UseCase2: REMAPPING & IMAGE LOADING TO ESRAM
UseCase4: READ/Write Access Time
UseCase5: EDAC feature test
Enter your choice : 2

************************************
REMAPPIND IS DONE TO ESRAM


ESRAM_CR = 1

DDR_CR = 0

ENVM_REMAP_BASE_CR = 0

************************************
```

ESRAM1 Remapping:

```
************************************
G4M-ESRAM UseModel:
```

****

UseCase1: ESRAM READ/WRITE ACCESS

UseCase2: REMAPPING & IMAGE LOADING TO ESRAM

UseCase4: READ/Write Access Time

UseCase5: EDAC feature test

Enter your choice : 2

****

REMAPPIND IS DONE TO ESRAM

ESRAM_CR = 3

DDR_CR = 0

ENVM_REMAP_BASE_CR = 0

****

5.2.5 Observation

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/results/G4M_ESRAM_UC2.txt

5.2.6 SVN Database

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/design/hw/G4M_ESRAM_UC1.zip

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/design/fw/G4M_ESRAM_UC1.zip

5.3 UseCase3- Dhrystone

5.3.1 Deviation from the Validation Plan

None

5.3.2 Implementation Guideline

All the below mentioned usecases will be executed as part of Dhrystone PDV G4M_Dhrystone_PDV_PLAN.docx.

UseCase3a	Dhrystone: ROM and RAM in eSRAM0 ESRAM_PIPELINE=OFF
UseCase3b	Dhrystone: ROM in eSRAM0 and RAM in eSRAM1 ESRAM_PIPELINE=OFF

UseCase3c	Dhrystone: ROM and RAM in eSRAM0 ESRAM_PIPELINE=ON
UseCase3d	Dhrystone: ROM in eSRAM0 and RAM in eSRAM1 ESRAM_PIPELINE=ON

5.4 UseCase4- Read/write access time

5.4.1 Deviation from the Validation Plan

None

5.4.2 Implementation Guideline

Firmware is used to perform Byte, Half-word and word access and perform write, read and alternate read write access.

User is given the option to enable or disable pipelining and EDAC.

Remapping is done to ESRAM1 (both RAM and ROM are in ESRAM1) and the read write access is done to ESRAM0. G4M_Timer is used to measure the time of execution.

5.4.3 Issues

None

5.4.4 PASS or FAIL criterion

5.4.5 Observation

When EDAC is OFF write access execution time is same for Word, Half-word and byte access.

When EDAC is ON, write access execution time for Half-word and Byte access is more than word access

Word write access execution time is always same irrespective of PIPELINE ON/OFF or EDAC ON/OFF.

Half-word and byte write access time is more when EDAC is ON when compared with EDAC OFF case.

Byte, Half-word and word read execution time is less when PIPELINE is OFF when compared with PIPELINE ON case. It has no impact of EDAC mode

PIPELINING	EDAC	Word Access			Half-Word Access			Byte Access		
		write	read	Alternate read/write	write	read	Alternate read/write	write	read	Alternate read/write
OFF	OFF	0xc5	0x13e	0x1f6	0xc5	0x13e	0x233	0xc5	0x13e	0x1f6
	ON	0xc5	0x13e	0x1f6	0x100	0x13e	0x270	0x100	0x13e	0x233
ON	OFF	0xca	0x17f	0x237	0xca	0x17f	0x275	0xca	0x17f	0x237
	ON	0xca	0x17f	0x237	0x105	0x17f	0x2b2	0x105	0x17f	0x274

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/results/G4M_ESRAM_UC4.TXT

5.4.6 SVN Database

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/design/hw/G4M_ESRAM_UC1.zip

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/design/fw/G4M_ESRAM_UC1.zip

5.5 UseCase5- EDAC Test

5.5.1 Deviation from the Validation Plan

None

5.5.2 Implementation Guideline

This test will validate 1-bit error detection and correction and two bit error detection feature of ESRAM.

Firmware is used to generate 1-bit EDAC error as follows.

First EDAC is enabled and walking-1 pattern is written to ESRAM.

Then EDAC is disabled and the ESRAM contents are modified such that single bit of ESRAM data will change.

In order to check that this single bit gets corrected, EDAC is enabled and the ESRAM contents are read.

5.5.3 Issues

None

5.5.4 PASS or FAIL criterion

Verification is done that the single bit error is corrected and the respective error counter is incremented.

Two bit error is detected and the respective EDAC counter is incremented, which is observed in debugger window because the two-bit error will result in HRESP error and further CM3 access will not be active

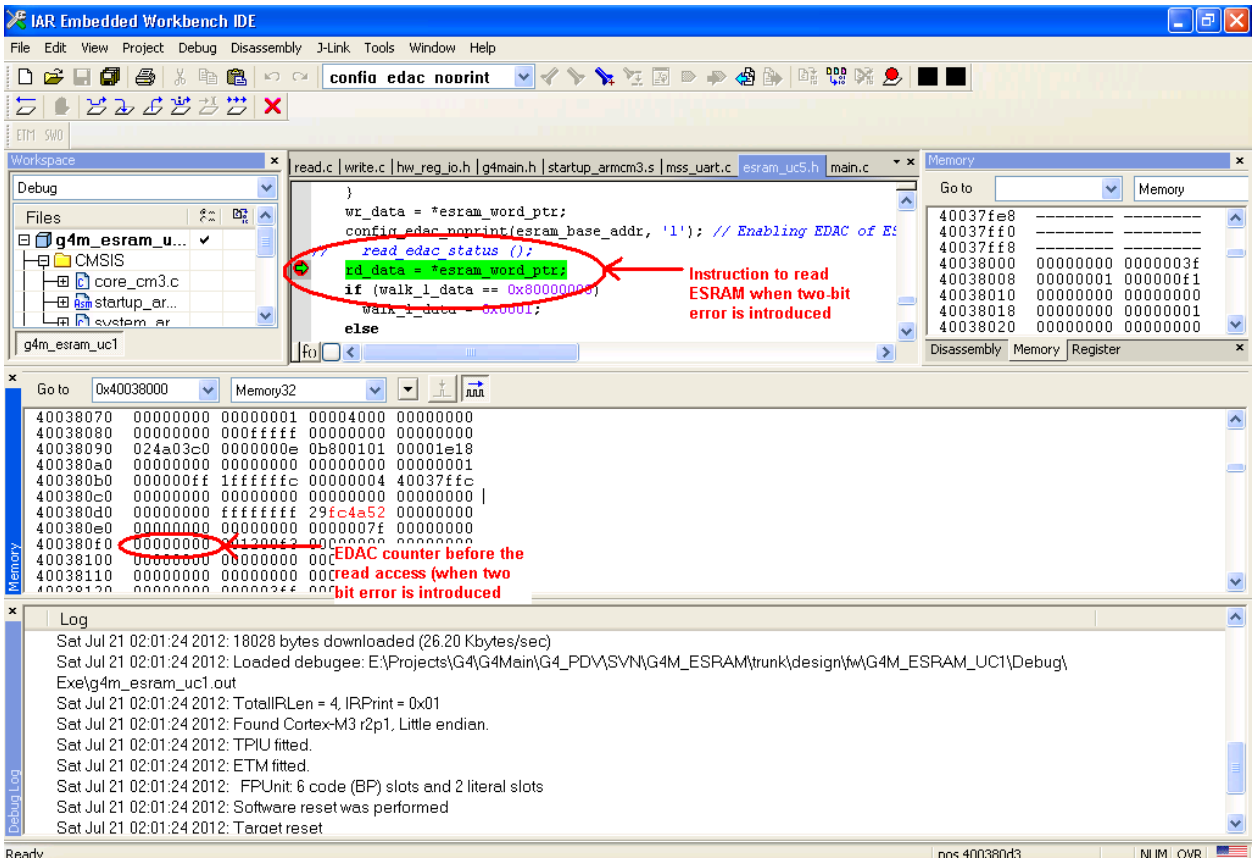


Figure 6 EDAC error status before reading ESRAM contents (with two-bit error)

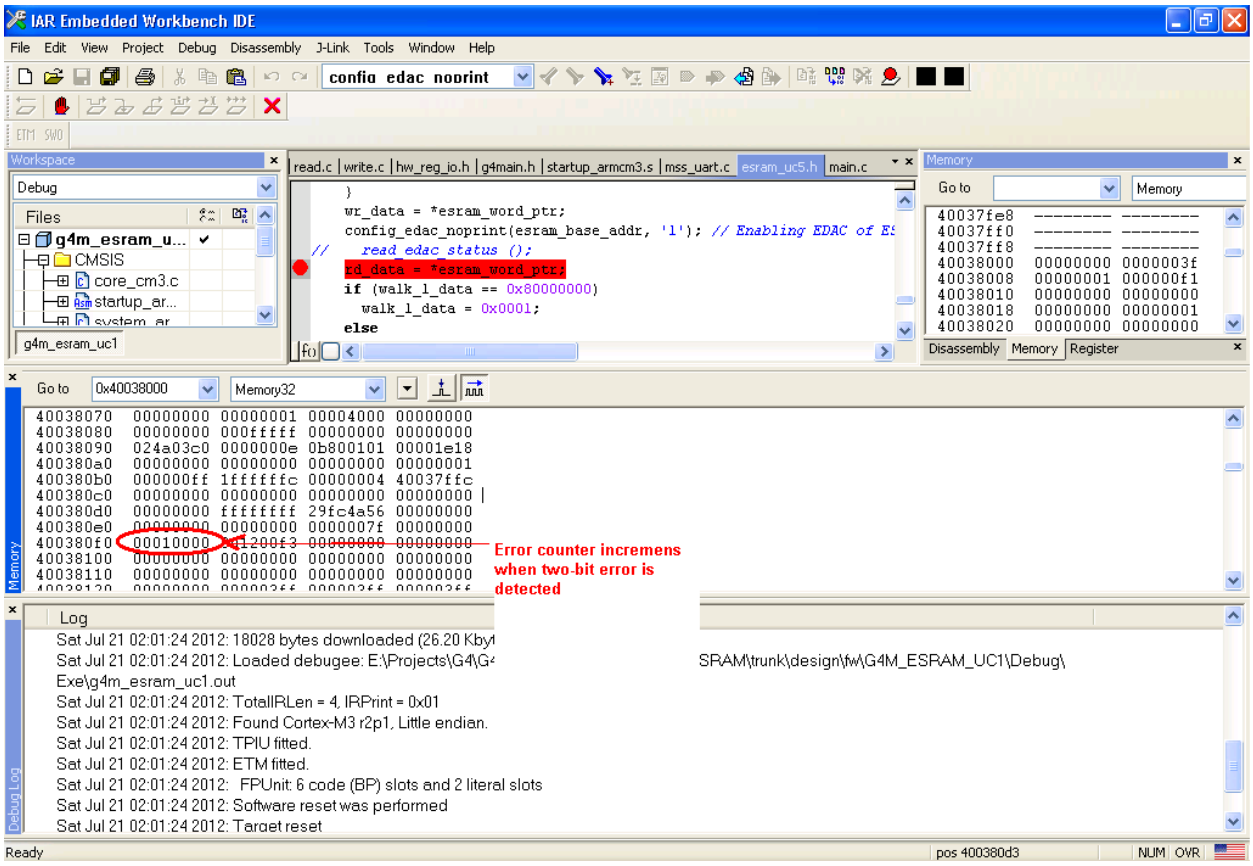


Figure 7 Error count increments after two-bit error detection

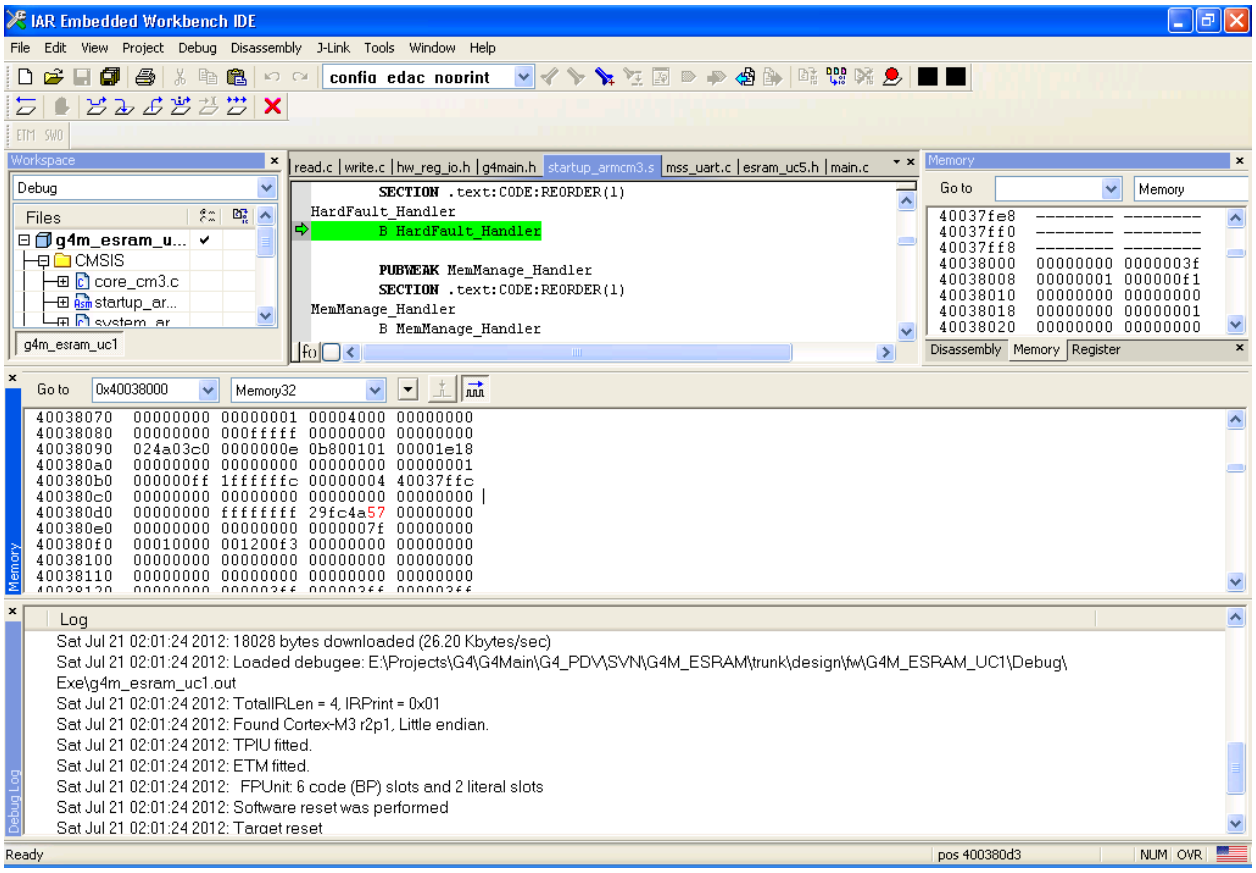


Figure 8 CM3 status after two-bit error detection

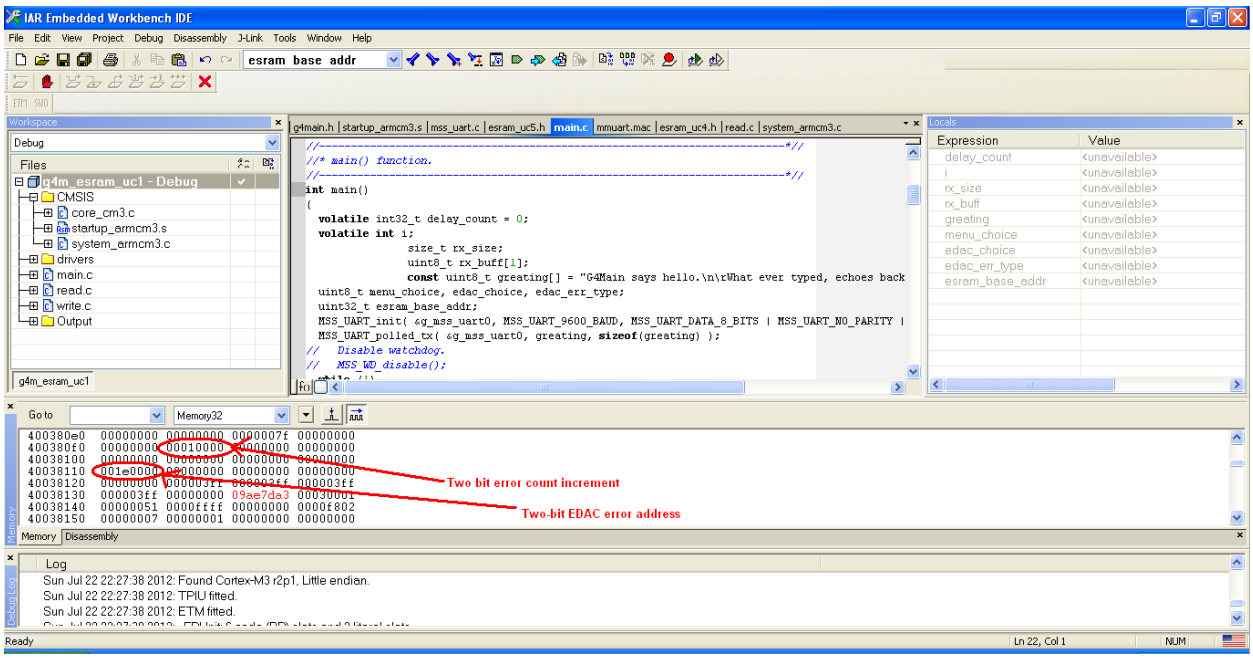


Figure 9 2-Bit error counter and EDAC error address

5.5.5 Observation

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/results/G4M_ESRAM_UC5.TXT

5.5.6 SVN Database

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/design/hw/G4M_ESRAM_UC1.zip

svn://hoppin/IP/PDV/G4_MAIN/G4M_ESRAM/tags/1.0.100/design/fw/G4M_ESRAM_UC1.zip

5.6 A2F5000 eSRAM to AHB Validation Summary

UseModel	Feature under validation	Result	Issues/Comments
UseCase1	Scratchpad	Passed	None
UseCase2	Image Loading	Pass	None
UseCase3a	Dhrystone: ROM and RAM in eSRAM0 ESRAM_PIPELINE=OFF	Will be executed as part of Dhrystone usecase	NA
UseCase3b	Dhrystone: ROM in eSRAM0 and RAM in eSRAM1 ESRAM_PIPELINE=OFF	Will be executed as part of Dhrystone usecase	NA
UseCase3c	Dhrystone: ROM and RAM in eSRAM0 ESRAM_PIPELINE=ON	Will be executed as part of Dhrystone usecase	NA
UseCase3d	Dhrystone: ROM in eSRAM0 and RAM in eSRAM1 ESRAM_PIPELINE=ON	Will be executed as part of Dhrystone usecase	NA
UseCase4	Read/write access time	Passed	None
UseCase5	EDAC – 1 bit & 2 bit error detection and correction	Passed	None

Table 5 List of UseCases

6 Appendix

6.1 Access delay

PIPELINE	ESRAM	OPERATION	SIZE	No. of WAIT STATES	No. of WAIT STATES (Write before)
ENABLE	32KB RAM EDAC_ON MODE	WRITE	32-Bit	0	0
			16-Bit	2	3
			8-Bit	2	3
		READ	32-Bit	1	2
			16-Bit	1	2
			8-Bit	1	2
	32KB RAM EDAC_OFF MODE	WRITE	32-Bit	0	0
			16-Bit	0	0
			8-Bit	0	0
		READ	32-Bit	1	2
			16-Bit	1	2
			8-Bit	1	2
	8KB RAM	WRITE	32-Bit	1	1
			16-Bit	0	0
			8-Bit	0	0
		READ	32-Bit	2	3
			16-Bit	1	2
			8-Bit	1	2
	32KB RAM EDAC_ON MODE	WRITE	32-Bit	0	0
			16-Bit	2	3
			8-Bit	2	3
		READ	32-Bit	0	1
			16-Bit	0	1
			8-Bit	0	1

DISABLE					
	32KB RAM EDAC_OFF MODE	WRITE	32-Bit	0	0
			16-Bit	0	0
			8-Bit	0	0
		READ	32-Bit	0	1
			16-Bit	0	1
			8-Bit	0	1
	8KB RAM	WRITE	32-Bit	1	1
			16-Bit	0	0
			8-Bit	0	0
		READ	32-Bit	1	2
			16-Bit	0	1
			8-Bit	0	1

Table 6 Tabular column for modes of operation in ESRAM