



Solution House  
**GAIO**  
TECHNOLOGY

Copyright (c) 2007 GAIO TECHNOLOGY CO., LTD. All rights reserved.

The screenshot displays the CoverageMaster winAMS interface. It includes a menu bar with options like 'Convert OMP', 'Launch Simulator', 'Create CSV', 'Search Signal', and 'CaseFile'. Below the menu is a toolbar with icons for 'Converter Settings' and 'Target Settings'. The main area is divided into several sections: 'Test Project Title' (Out2006-12-29(22'20'00)'), 'Date Tested' (2007/03/28 13:11:03), and 'Result' (NG). A 'Variable Name' table lists variables like @code, gb\_a, gb\_b, gb\_c, gb\_out, and FuncName. A 'CONDITION' table shows coverage data for various conditions. The bottom section features the text 'CoverageMaster winAMS GAIO Unit Test Simulator for Embedded Software'.

Variable Name	Test Data	I/O	St
@code		Input	
gb_a		Input	
gb_b	20,21	Input	20
gb_c	30,31	Input	30
gb_out		Output	
FuncName		Output	

CONDITION	CONDID	1	2	3	4	5
CONDID						
CONDID	CONDID	CONDID	CONDID	CONDID	CONDID	CONDID

**CoverageMaster winAMS**  
GAIO Unit Test Simulator for Embedded Software

## Solving "Memory not assigned" Errors

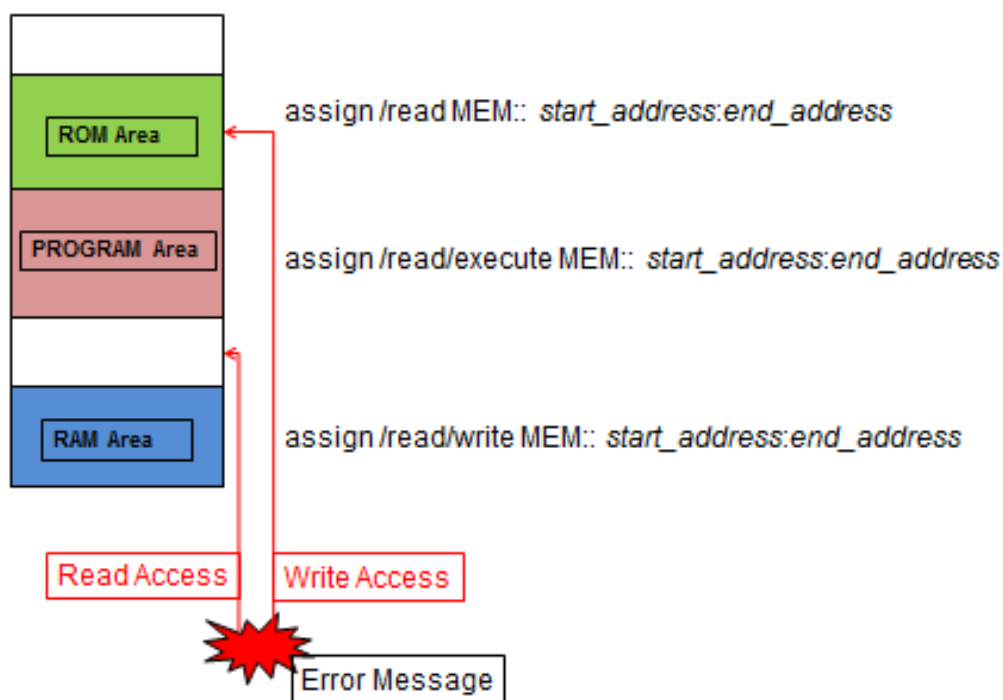
Ver. 1.0  
August 2014

# CoverageMaster: Solving “Memory not assigned” Errors

## 1. About Memory Assignment

Before running a simulation with GAIO’s MPU Simulator System-G, the memory of the simulated target must be assigned, with access permissions set like in the real MPU. This allows the simulation to detect memory related errors like unauthorized or unexpected access to certain areas.

To “assign” memory means to define areas (=range of addresses) and set the access permissions for each of these areas. The 3 possible access levels are READ, WRITE and EXEC (=Execute). The permissions are set with the “ASSIGN” command in a command file or directly from the console in the Simulator window.



## 2. Using the *assign* command

**assign[/read][/write][/exec] *start\_address:end\_address***

Grants the following permission to the memory space between *start\_address* and *end\_address*:

/read: Grant Read Access permission

/write: Grant Write Access permission

/exec: Grant Execution Access permission

Parameters in brackets [] are optional.

When no parameter is specified, “/read/write” is assumed.

### Example:

In this example, we assume that we want to assign the Memory Map below:

Data area:

0x00000 - 0x0ffff ROM

0x20000 - 0x2ffff RAM

Program area:

0x10000 - 0x1ffff ROM

0x30000 - 0x3ffff RAM

In this case the following commands are used:

<b>Memory Area</b>	<b>Memory Type</b>	<b>Command</b>
0x00000 - 0x0ffff	ROM	assign/read 0:0ffffh
0x10000 - 0x1ffff	ROM	assign/read/exec 10000h:1ffffh
0x20000 - 0x2ffff	RAM	assign 20000h:2ffffh
0x30000 - 0x3ffff	RAM	assign/read/write/exec 30000h:3ffffh

### 3. When to use the *assign* command

In a standard program, memory areas for constants, variables and program instructions are automatically assigned during the linking of the object file. Manually assigning areas is only required when a “Memory is not assigned” error occurs because an area not defined in the code is accessed (ex: Stack, I/O Registers, etc...) or because of an error in the code.

When an unauthorized access happens, an error message will be displayed in the System Simulator console, and the simulation will stop. The error message depends on the MPU, two examples are show below.

#### Ex1: SH Series

```
Invalid address found [code = 18] (pc=*****)  
%SYSTEMG-E-SEM, stopped simulation (status code = 12h)  
'MEM::*****'
```

Explanation: An unauthorized access occurred when executing the instruction at the address \*\*\*\*\*

#### Ex2: Generic Example

```
%SYSTEMG-E-SEM, memory is not assigned on the address  
'MEM::*****'
```

Explanation: An unauthorized access occurred on the address \*\*\*\*\* , which has not been assigned.

In the error message, either the address that is being accessed or the address of the instruction performing the unauthorized access is displayed. Using the “Register” and “Memory” views of the System Simulator interface, confirm that the address being accessed is the intended one.

If the address is correct, then the Memory not being properly assigned is the issue. In this case the solution is to assign the related area with the correct permissions using an *assign* command in the Startup Command File.

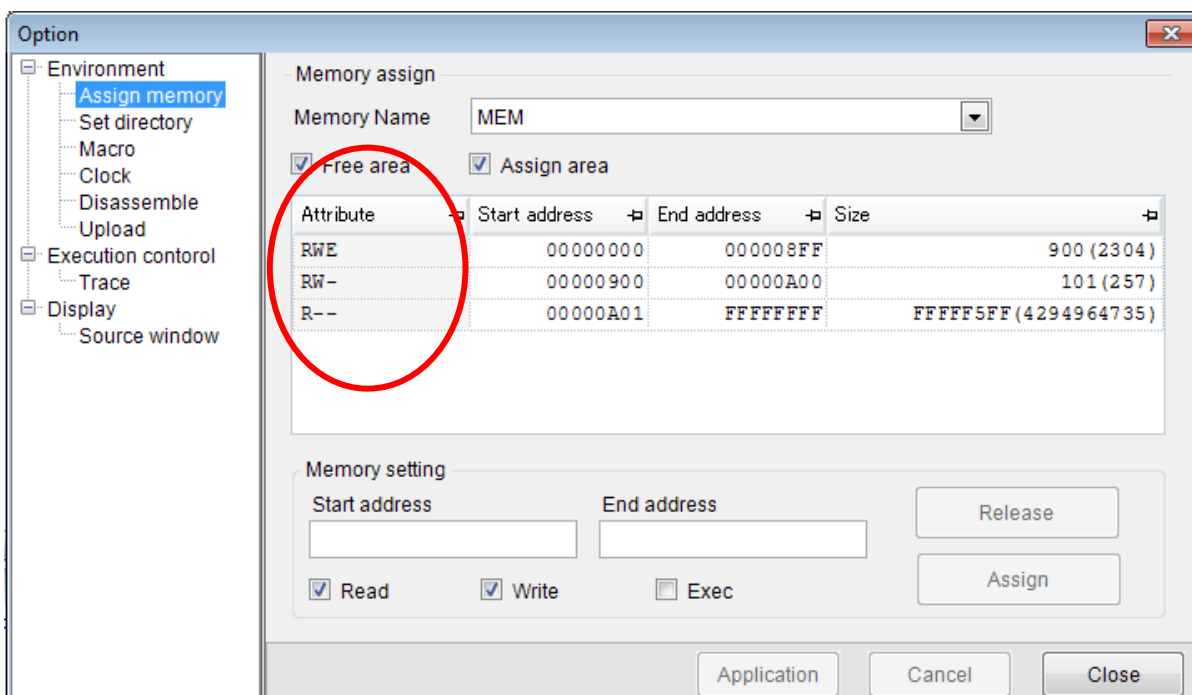
If the address is incorrect, then the problem is in the code, and the program must be corrected using the debugger features of the System Simulator (step-by-step execution, breakpoints, etc...) to locate the cause.

#### 4. How to check the Memory Mapping

The current status of memory allocation can be checked in the “Assign memory” screen of the Simulator’s (Debugger) Option menu. To access it, select the [Program] menu and click on [Memory Allocation].

In the “Attribute” column, the permissions for Read, Write and Execute are displayed: [R], [W] or [E] if corresponding access is allowed, and [-] if it is not.

Example:



### Copyright Notice and Disclaimer

**GAIO TECHNOLOGY CO., LTD.**

Tennouzu First Tower 25F  
2-2-4 Higashi-Shinagawa, Shinagawa-ku,  
Tokyo 140-0002 Japan

Tel: +81-3-4455-4767  
Email: user@gaio.co.jp

\* Company names and product names that appear in this presentation are trademarks of their respective company.

\* Unauthorized distribution or duplication of this presentation material is prohibited.

This document is copyright of GAIO TECHNOLOGY CO., LTD. All rights reserved.  
The content of this document is subject to change without notice.

We assume no responsibility for any losses or damages that may occur from errors in this document.