Application Report
SPRACI7 – October 2018

C2000™ Memory Power-On Self-Test (M-POST)

Prasanth Viswanathan Pillai, Salvatore Pezzino, Peter Ehlig

ABSTRACT

This application report discusses the Memory Power-On Self-Test (M-POST) feature available in select series of C2000 real-time controllers. The M-POST architecture enables parallel testing of multiple memories to reduce test time and is used for power-on testing of the memories on-chip.

Contents

List of Figures

List of Tables

Trademarks

C2000 is a trademark of Texas Instruments.
All other trademarks are the property of their respective owners.

1 Introduction

C2000 devices are powerful 32-bit floating-point microcontroller units (MCU) designed for advanced closed-loop control applications such as motor control and power conversion control in industrial drives and automation, industrial power, solar, and electrical vehicle applications. In addition to the strong control performance offered by the MCU, it supports a host of functional safety features to support customers to design and certify their functionally safe systems. Memory Power-On Self-Test (M-POST) is an important enabler to test the device SRAMs and ROMs during device start-up. Based on customer one-time programmable (OTP) configurations, the test is executed automatically with the help of on-chip hardware during boot-up. When the test is executed, multiple memories are tested in parallel to reduce the impact on boot-time.

1.1 Overview of Memory Test Requirements

With increased deployment of semiconductors in automobile and industrial markets, functional safety is becoming a critical dimension of semiconductor design in addition to the traditional aspects like power, performance, area, and so forth. Functional safety standards (for example, ISO26262 for Automotive, IEC61508 for industrial, IEC60730 for white goods, and so forth) list the requirements to be adhered to during the design and deployment of such systems.

Functional safety requires freedom from unreasonable residual risk originating from faulty behaviour of safety related electronic and electrical systems. Faults can be either systematic or random hardware faults. With regard to random hardware faults, memory (especially SRAM) is one of the most significant components of functional safety in the execution of software in electronics and programmable electronics. This is because SRAM is usually the largest component in the overall device in terms of both area and transistor count. Furthermore, SRAM is very dense and therefore susceptible to subtle defects. Moreover, SRAM operates in reduced voltage ranges vs normal circuit logic and is therefore more susceptible to disturbances. Further discussion of the nature of SRAM and error detection is outside the scope of the application report. For more information, see *Error Detection in SRAM*.

Customers who use the C2000 devices in functionally safe systems may require a specific diagnostic coverage (DC) of the memory in order to meet various functional safety standards or requirements. A particular diagnostic coverage may be required for both single point faults as well as latent faults. Electronics and programmable electronics used in functionally safe applications might require the capability to perform SRAM and ROM test at start-up or periodically during maintenance cycles. Some other class of safety applications require the memories to be tested periodically in parallel with application execution. M-POST on C2000 is designed and enabled to target the former in order to detect latent faults. M-POST assists customers to achieve their functional safety requirements by providing the capability to perform memory tests at start-up.

Additionally, in order to detect random hardware faults in-system, C2000 devices are equipped with EDAC, or error detection and correction logic, on memories. Furthermore, the F28x7x SafeTI Diagnostic Library (SDL) provides software for a RAM March13n algorithm that specifically targets the data, address and EDAC bits for permanent stuck-at faults, as well as some worst-case path timing faults that can occur from system degradation over time. The March13n test is designed for robust in-system testing of memories and can be easily integrated into a system application. For more information, see *C2000™ CPU Memory Built-In Self-Test*.

The processing elements on C2000 MCUs that utilize the on-chip memories are equipped with a Hardware Built-In Self-Test (HWBIST) that targets the C28x CPU logic including the TMU, FPU, and VCU and is able to achieve up to 99% DC. For more details, see *C2000™ CPU Memory Built-In Self-Test*. Furthermore, the Control Law Accelerator Self-Test Library (CLA STL) targets the CLA for 90% DC and enables the use of CLA for processing in functionally safe systems. For more information, see *C2000™ CLA Self-Test Library*.

1.2 Terms and Definitions

Table 1. Acronyms Used in This Document

Term	Definition
ATE	Automated Test Equipment
CPU	Central Processing Unit
DC	Diagnostic Coverage
DCSM	Dual Zone Code Security Module
FTTI	Fault Tolerant Time Interval
MDP	Memory Data Path (Path interfacing the PBIST controller and the memory)
MPOST	Memory Power on Self-Test
NMI	Non-Maskable Interrupt
OTP	One-Time Programmable
PBIST	Programmable Built In Self-Test (Programmable engine required for self-test)
ROM	Read-Only Memory
SRAM	Static Random Access Memory

2 System Challenges to Memory Validation

2.1 Memory Test Flow

M-POST is enabled with the help of a Programmable Built in Self-Test (PBIST) solution, which is used for the manufacturing test of the device. PBIST has a CPU configurable interface used for field self-test in addition to the Automated Test Equipment (ATE) interface used during manufacturing test.

The PBIST architecture consists of a controller which is specifically designed toward efficient memory testing. The controller is designed with a dedicated register set and a highly specialized pipeline and instruction set targeted specifically toward testing memories. Furthermore, the PBIST engine is equipped with multiple read and multiple write memory ports, or buses, which enable it to efficiently test multiple memory instances in parallel. The PBIST controller also has access to the PBIST ROM. The PBIST ROM is where test routines are stored for the PBIST engine to fetch and execute. These test routines are fetched by the PBIST controller and executed on multiple on-chip memory instances in parallel. Because of the specialized architecture and test routines, PBIST provides very high diagnostic coverage on the implemented SRAMs and ROMs at a transistor level in a very efficient manner. Because PBIST controller has direct access to the ROMs and SRAMs, it is even able to test secure memory instances on devices equipped with DCSM (Dual Zone Code Security Module). For details on leveraging the DCSM on F28004x, see *Achieving Coexistence of Safety Functions for EV/HEV Using C2000™ MCUs*.

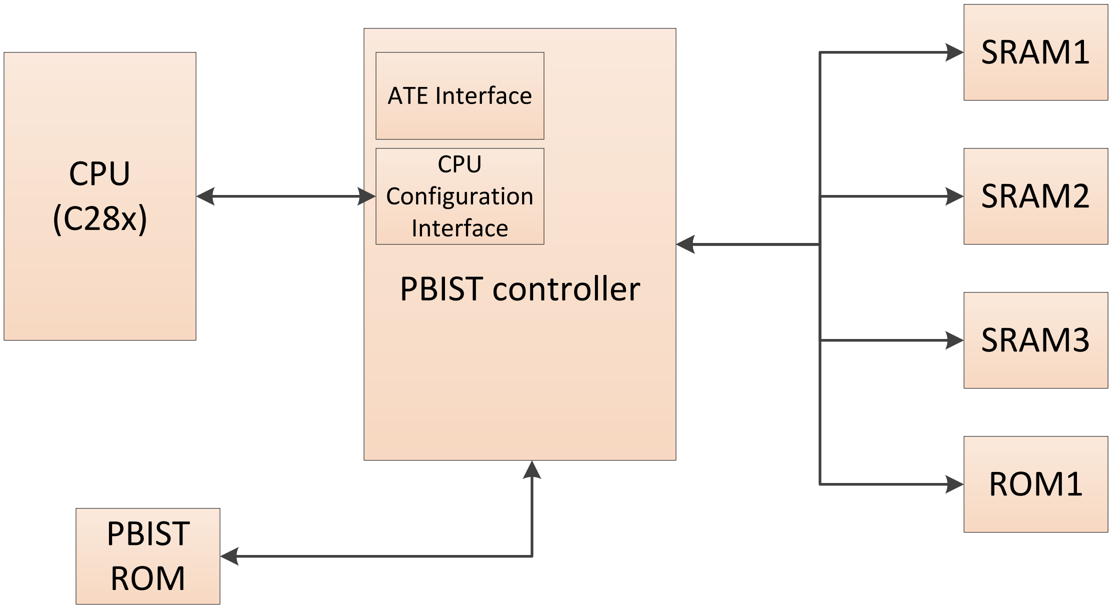

Figure 1. PBIST Architecture

The steps involved in the execution of memory self-test are described in Figure 2.

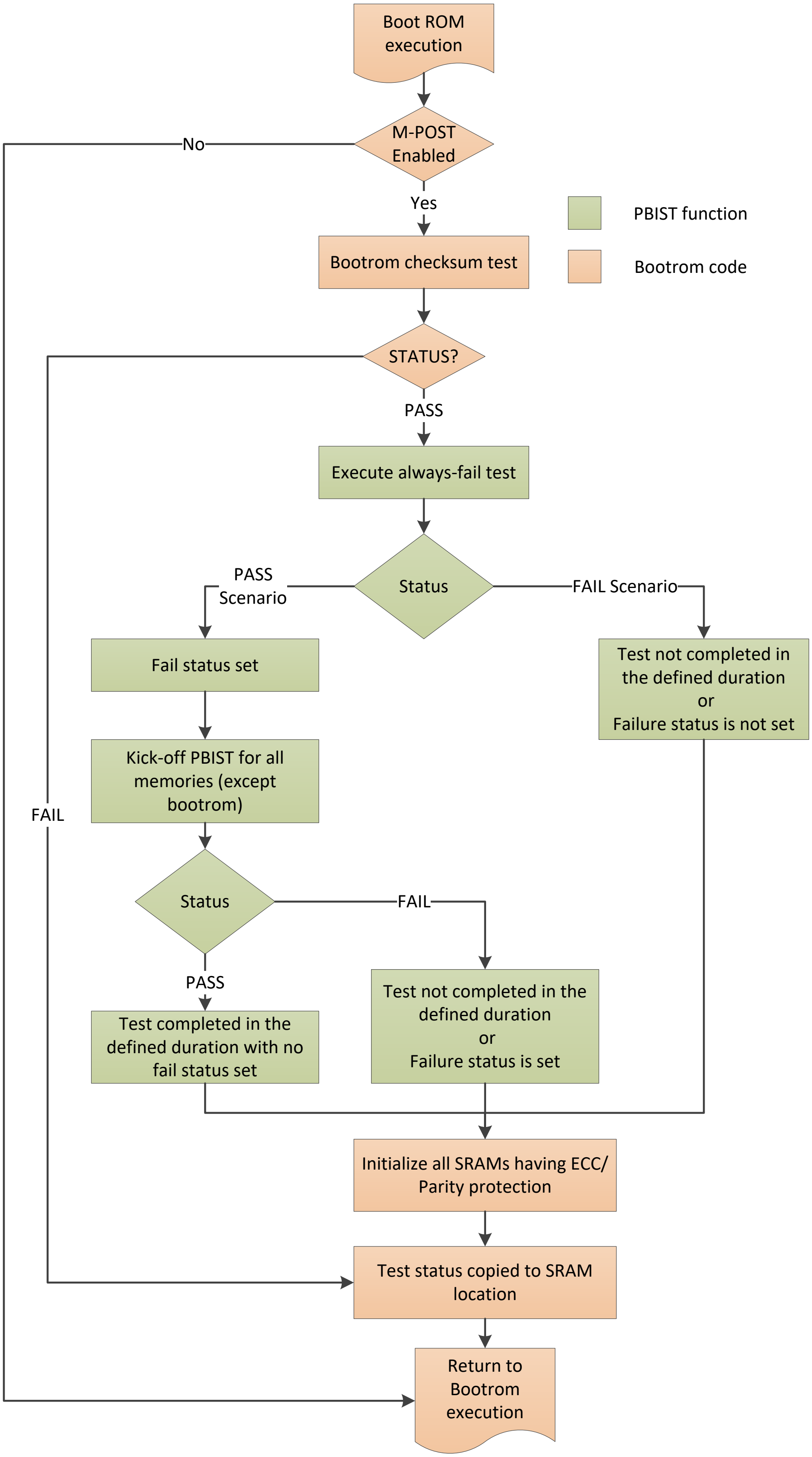
Boot ROM execution
M-POST Enabled
No
Yes
PBIST function
Bootrom code
Bootrom checksum test
STATUS?
PASS
FAIL
Execute always-fail test
Status
PASS Scenario
FAIL Scenario
Fail status set
Test not completed in the defined duration or Failure status is not set
Kick-off PBIST for all memories (except bootrom)
Status
FAIL
PASS
Test completed in the defined duration with no fail status set
Test not completed in the defined duration or Failure status is set
Initialize all SRAMs having ECC/ Parity protection
Test status copied to SRAM location
Return to Bootrom execution

Figure 2. M-POST Execution Flow

1. If enabled by configuring customer-OTP, M-POST is executed during every Power-ON reset sequence. Test is executed only during Power-ON reset and not during other resets (XRSn, WDRSn, Debugger reset, and so forth).
2. Since the code for testing of the memories resides in boot rom, it is not be possible to test the boot rom using PBIST. Hence a separate boot-rom checksum test will be done prior to PBIST. If the boot-rom checksum test fails, rest of PBIST test is not executed. The test status is copied to SRAM.
3. Prior to performing any test using PBIST, an always fail test case is executed. This is to validate the proper functioning of the PBIST controller and its ability to indicate failure. The test status bits are registered on completion of the test. If the test fails (i.e. always fail test passes), the entire test sequence is bypassed and test status is returned to the application code. If the test passes, the complete set of device memories (except boot-rom) is tested in the next phase.
4. All the SRAMs are tested using March13n and all the ROMs other than boot-rom will be tested using ROM signature computation algorithm.
5. Once the test is complete, all the CPU RAMs are initialized.
6. Timeout feature is implemented during each of the test sequence to identify whether the test is completing during the stipulated time or getting delayed due to a fault.
7. The status of the PBIST test (test not executed due to failure of checksum test, always fail test status, memory self-test status and timeout error) are communicated to the application.
8. The device will not be fit to run any functional safety application if the memory test is failing. Boot-rom will let the application code determine the future course. The application may fire NMI (using software) and let the external devices know about the failure using ERROR_STS pin.
9. If the start-up memory test is failing, application will not have enough information to identify the failing memory. Application may have to run diagnostics for this. The diagnostics sequence may be to execute tests on individual memory instances and identify the failing memory.

2.2 SRAM test Algorithmic Coverage

Device SRAMs are tested using March13n algorithm. This algorithm ensures that:

- The bit cell can be written and read as both a 1 and a 0
- Bridging defects between adjacent bit cells are detected
- The row and column decode is such that the targeted bits (and only the targeted bits) are affected on a write
- The row and column decode is such that only the targeted bits are presented to the boundary of the memory on a specified read

The sequence of operations included during the test of an SRAM is given in .

Table 2. March13n Test Sequence

Address	Initialization	March Element 1	March Element 2	March Element 3	March Element 4
0	Wr(0)	Rd(0), Wr(1), Rd(1)	Rd(1), Wr(0), Rd(0)	Rd(0), Wr(1), Rd(1)	Rd(1), Wr(0), Rd(0)
1	Wr(0)	Rd(0), Wr(1), Rd(1)	Rd(1), Wr(0), Rd(0)	/	/
2	Wr(0)	Rd(0), Wr(1), Rd(1)	Rd(1), Wr(0), Rd(0)	Rd(0), Wr(1), Rd(1)	Rd(1), Wr(0), Rd(0)
|	\	\	\	Rd(0), Wr(1), Rd(1)	Rd(1), Wr(0), Rd(0)
N-1	Wr(0)	Rd(0), Wr(1), Rd(1)	Rd(1), Wr(0), Rd(0)	Rd(0), Wr(1), Rd(1)	Rd(1), Wr(0), Rd(0)

2.3 ROM Test Algorithmic Coverage

ROM test algorithm reads the contents of the ROM and computes a 32-/64-bit signature that is compared against the golden signature.

3 Summary

C2000 PBIST provides another diagnostic safety mechanism to the customer that can be used to further enable functionally safe systems. It can be used as an efficient means to detect latent faults in the ROMs and SRAMs of C2000 devices at start-up.

4 References

- *Error Detection in SRAM*
- *C2000™ CPU Memory Built-In Self-Test*
- *C2000™ CLA Self-Test Library*
- *Achieving Coexistence of Safety Functions for EV/HEV Using C2000™ MCUs*
- *TMS320F28004x Piccolo™ Microcontrollers Data Manual*
- *TMS320F28004x Piccolo Microcontrollers Technical Reference Manual*
- *C2000™ Hardware Built-In Self-Test*
- *Testing semiconductor memories : theory and practice*, A.J. van de Goor

Appendix A

M-POST Working in F28004x

The implementation of the M-POST feature on F28004x is described in this section. To confirm the feature availability and details pertaining to other devices, see the device-specific TRM and data sheet.

A.1 Enabling of Test

Memory power on self test can be enabled by configuring the “Z1-OTP-BOOT-GPREG2” register as given in Table 3. All the requirements as mentioned in the *ROM Code and Peripheral Booting* section of the TRM needs to be adhered to for configuring these USER OTP bits.

Table 3. Z1-OTP-BOOT-GPREG2 Register Description

Bit	Name	Description	Boot ROM Action
31-24	Key	Write 0x5A to these 8-bits to tell the boot ROM code that the bits in this register are valid	If set to 0x5A, boot ROM uses the values in this register. If set to any other value, boot ROM ignores the values in this register.
23-8	Reserved	Reserved	No action
7-6	Reserved	Configures the memory power on self-test 0x0 - Execute self-test 0x1 - No action (Reserved) 0x2 - No action (Reserved) 0x3 - No action (Reserved)	
5-4	Error Status Pin	Sets the GPIO pin to be used as the ERRORSTS 0x0 – GPIO24 0x1 – GPIO28 0x2 – GPIO29 0x3 – ERRORSTS disabled (Default)	No action Boot ROM configures the appropriate mux for the selected GPIO pin
3-0	Reserved	Reserved	No action

A.2 M-POST Duration

Once the test is enabled, power-on self-test is executed in PLL bypass mode. Depending upon the device configuration the test will take a approximately 150,000 INTOSC2 cycles for completion of the test. This will be observed as an increase in the boot duration.

A.3 M-POST Result

The status of the execution of M-POST can be understood by reading 10th bit of BROM_STATUS located at 0x0000_0002. The details regarding the test execution can be understood from return value of M-POST test function is stored at “0x0000_0008”. The values can be decoded as given in Table 4.

Table 4. M-POST Status

M-POST Status	Return Value
Test not executed	0x00000000
Always fail test timeout error	0xFF00FF00
Always fail test incorrect operation	0xFF11FF00
Test timeout error	0xFF22FF00
Test timeout error	0xFF33FF00
Test fail	0xFF44FF00
Test pass	0xFFFFFFFF

A.4 Periodic Self-Test

The capability described in the document will help support power-on self-test of the memories. If your application needs to support periodic self-test, see *C2000™ CPU Memory Built-In Self-Test*.

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, 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 will 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 (www.ti.com/legal/termsofsale.html) 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.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265