Application Report
SPNA128C–April 2011

Basic PBIST Configuration and Influence on Current Consumption

Haixiao Weng

ABSTRACT

The purpose of this application report is to provide help for configuring the PBIST module of TMS570LS series microcontrollers. It also explains the configurations, the different algorithm test durations and the influence on current consumptions.

Project collateral and source code discussed in this application report can be downloaded from the following URL: http://www.ti.com/lit/zip/spna128.

Contents

List of Figures

List of Tables

1 Introduction

TMS570LS series microcontrollers are implemented with programmable built-in self test (PBIST) architecture. The PBIST architecture provides a memory BIST engine for varying levels of coverage across many embedded memory instances.

TMS570LS series microcontrollers can be classified into two categories: 130 nm technology devices (TMS570LS10x and TMS570LS20x series) and 65 nm technology devices (TMS570LS21x and TMS570LS31x series).

The on-chip memories of TMS570LS10x and TMS570LS20x series microcontrollers are classified into fifteen different ROM/RAM groups for the PBIST test, including two ROM groups (STC/PBIST ROM) and thirteen RAM groups. For more details, see the *PBIST RAM Grouping* table in [2]. The clock associated to ROM groups PBIST testing is the STC/PBIST ROM clock; the test clock for eSRAM, DMA and RTP RAM are the HCLK; while the other RAM groups use VCLK as the PBIST testing clock.

The on-chip memories of TMS570LS21x and TMS570LS31x series microcontrollers are classified into twenty-eight different ROM/RAM groups for the PBIST test, including two ROM groups (STC/PBIST ROM) and twenty-six RAM groups. For more details, see the *PBIST RAM Grouping* table in [4]. The clock associated to ROM groups PBIST testing is the STC/PBIST ROM clock; the test clock for eSRAM, DMA and RTP RAM are the HCLK; while the other RAM groups use VCLK as the PBIST testing clock.

The PBIST engine in TMS570LS10x and TMS570LS20x series microcontrollers provides a list of algorithms for memory test. The different algorithms supported in application mode for a certain ROM/RAM group are listed in the *RAM Grouping and Algorithm* section of the *TMS570LS Series Microcontroller Technical Reference Manual* (SPNU489) [1]. Similar information for TMS570LS21x and TMS570LS31x series can be found in the *TMS570LS31/21 16/32-Bit RISC Flash Microcontroller Technical Reference Manual* (SPNU499) [3].

NOTE: The same PBIST test algorithm has been assigned with different number/index in TMS570LS10x and TMS570LS20x series and TMS570LS21x and TMS570LS31x series. For example, the number/index for March13 algorithm on single port RAM is ‘3’ for TMS570LS10x and TMS570LS20x series device while it is ‘4’ for the TMS570LS21x and TMS570LS31x devices.

2 When Should PBIST Be Run?

The PBIST test is designed to test the integrity of the embedded RAMs and ROMs and it is destructive to the data stored in the target RAM. The PBIST test can be treated as:

- Proof testing: Hardware PBIST engine executes multiple algorithms targeted to SRAM physical topology on startup.

  This method is recognized to provide 99% diagnostic coverage (DC) based on TI proven-in-use manufacturing data.
- Online/diagnostic testing (optional): Periodic PBIST can be executed in slices, but is destructive to contents. After test completes, the application code has to re-initialize the target RAM. A reset is not required, but that is the simplest solution that has been chosen to use for this example.

3 PBIST Configuration Steps in TMS570LS Series

This section describes a step-by-step PBIST configuration.

1. Program the HCLK to STC/PBIST ROM clock ratio.

   The PBIST engine can test both ROMs (group 1 - 2) and RAMs (other groups). The ROM_DIV bits in MSTGCR register programs the HCLK to STC/PBIST ROM clock ratio. Therefore, it impacts the test duration and test current consumed during the ROM test. The test duration is proportional to the HCLK to ROM clock ratio. A higher HCLK to ROM clock ratio also means less current consumption during the ROM PBIST test.

   Because all of the other memories (RAMs) are running on HCLK or VCLK, this clock ratio has no influence on the RAM PBIST test (group 3 - x).

   The example below uses the following code to set ROM clock the same as the HCLK frequency:

```
e_SYSTEM_ST.MSTGCR_UN.MSTGCR_ST.ROM_DIV_B2 = 0; //ROM clock = HCLK
```

2. Enable the PBIST controller in the system module.

   The example below uses the following code to enable PBIST controller:

```
e_SYSTEM_ST.MSINENA_UL=0x1;
```

3. Enable the PBIST self test in system module.

   The example below uses the following code to enable PBIST self test:

```
e_SYSTEM_ST.MSTGCR_UN.MSTGCR_ST.MSTGENA_B4=0x0A;
```

4. Wait until PBIST is out of reset.

   Depending on the ROM_DIV ratio, the PBIST will reset for 16~96 VBUS cycles. Please refer to the register MSTGCR (at 0xFFFF_FF58) description in *TMS570LS Series Microcontroller Technical Reference Manual* (SPNU489) [1] for more details.

```
asm(“ nop");
……
asm(" nop");
```

5. Enable the PBIST internal clocks and ROM interface clock.

   The example below uses the following code to enable the PBIST internal clocks and ROM interface clock.

```
e_PBIST_ST.PACT_UN.PACT_UL=3;
```

6. Select the RAM/ROM under test, select the algorithm.

   Choose the RAM-override mode if you want to run the algorithms on all the RAM/ROMs that support these algorithms; otherwise, disable it.

   - RAM-override mode enabled

     In this case, the application code selects the algorithms. The PBIST engine runs on all the valid ROM/RAMs for these algorithms.

     Example:

```
e_PBIST_ST.ALGO_UN.ALGO_UL=0xFFFF; //select all the algorithms
e_PBIST_ST.PBISTOVERRRIDE_UN.PBISTOVERRRIDE_ST.OVER=1; // default setting
```

NOTE: If you are using a subset device, e.g., TMS570LS21x, not all the power domains are turned on. However, the PBIST engine is designed for the superset device with all power domains turned on. Therefore, the application cannot perform the PBIST test in the RAM-override mode, otherwise, it will return a PBIST failure because a few RAM blocks under test are not present in the subset device.

   - RAM-override mode disabled

     In this case, the application code selects the algorithm and runs on the target ROM/RAM through the ALGO register. Then, it selects the ROM/RAM under test through the RINFOL registers and sets the OVER register to be 0. Note that the algorithm selected must be valid for all the ROM/RAMs selected by RINFOL, otherwise, a PBIST test error will be generated.

The example below uses the following code to select the March13 algorithm on the 160 Kbyte eSRAM.

Example:

```
e_PBIST_ST.ALGO_UN.ALGO_UL=0x4; // select March13 on Single port RAM
e_PBIST_ST.RINFOL_UN.RINFOL_UL=0x020;  //select the main ESRAM 160 kByte.
e_PBIST_ST.RINFOU_UN.RINFOU_UL=0;//It is always 0 for TMS570LS2x series.
e_PBIST_ST.PBISTOVERRRIDE_UN.PBISTOVERRRIDE_ST.OVER=0;
```

7. Select both Algorithm and ROM/RAM information from on chip PBIST ROM.

   The attached example uses the following code to select algorithm and ROM/RAM information from on chip PBIST ROM.

```
e_PBIST_ST.ROM_UN.ROM_ST.ROM=0x03;
```

8. Configure PBIST to run in ROM Mode and kickoff PBIST test.

   PBIST test will start after writing 0x14 to the DLA register.

   Example:

```
e_PBIST_ST.DLR_UN.DLR_UL =0x14;
```

9. Wait for PBIST test to complete by polling MSTDONE bit in system module.

```
while(e_SYSTEM_ST.MSTCGSTAT_UN.MSTCGSTAT_ST.MSTDONE_B1 != 0x1);
```

10. Once self test is complete, check the fail status registers FSRF0 and FSRF1.

    In case there is a failure (FSRF0 or FSRF1=0x01)

    - Read RAMT register that indicates the RGS and RDS values of the failure ROM/RAM.

      Example:

```
RGS = e_PBIST_ST.RAM_UN.RAM_ST.RAMGROUPSELECT;
RDS = e_PBIST_ST.RAM_UN.RAM_ST.RETURNDATASELECT;
```

      Register group select (RGS) and register data select (RDS) stand for an unique ROM/RAM select id. To find out the failing ROM/RAM, you can check the *PBIST RAM Grouping* table in [2] or the *PBIST RAM Grouping* table in [4].

    - Read FSRC0 and FSRC1 registers that contains the failure count.

      Example:

```
FSCP0=e_PBIST_ST.FSRC0_UN.FSRCO_UL;
FSCP1=e_PBIST_ST.FSRC1_UN.FSRC1_UL;
```

    - Read FSRA0 and FSRA1 registers that contains the address of first failure.

      Example:

```
FSADDRP0 = e_PBIST_ST.FSRA0_UN.FSRA0_UL;
FSADDRP1 = e_PBIST_ST.FSRA1_UN.FSRA1_UL;
```

- Read FSRDL0 and FSRDL1 registers that contain the failure data.

  Example:

```
FSDATAP0 =  e_PBIST_ST.FSRDL0_UN.FSRDL0_UL;
FSDATAP1 =  e_PBIST_ST.FSRDL1_UN.FSRDL1_UL;
```

- Resume the test if required using program control register STR = 2.

  Example:

```
e_SYSTEM_ST.MSTCGSTAT_UN.MSTCGSTAT_ST.MSTDONE_B1 = 0x1;
e_SYSTEM_ST.MSTCGSTAT_UN.MSTCGSTAT_ST.MSTDONE_B1 = 0x1;
```

In case there is no failure (FSRF0 and FSRF1=0x00), the memory self test is complete.

1. Disable the PBIST internal and ROM clocks.

   Example:

```
e_PBIST_ST.PACT_UN.PACT_UL=0x00; /*Disable PBIST internal ROM Clocks*
```

2. Disable the PBIST self test.

   Example:

```
e_SYSTEM_ST.MSTGCR_UN.MSTGCR_ST.MSTGENA_B4=0x05;/*Disable PBIST selftest*/
```

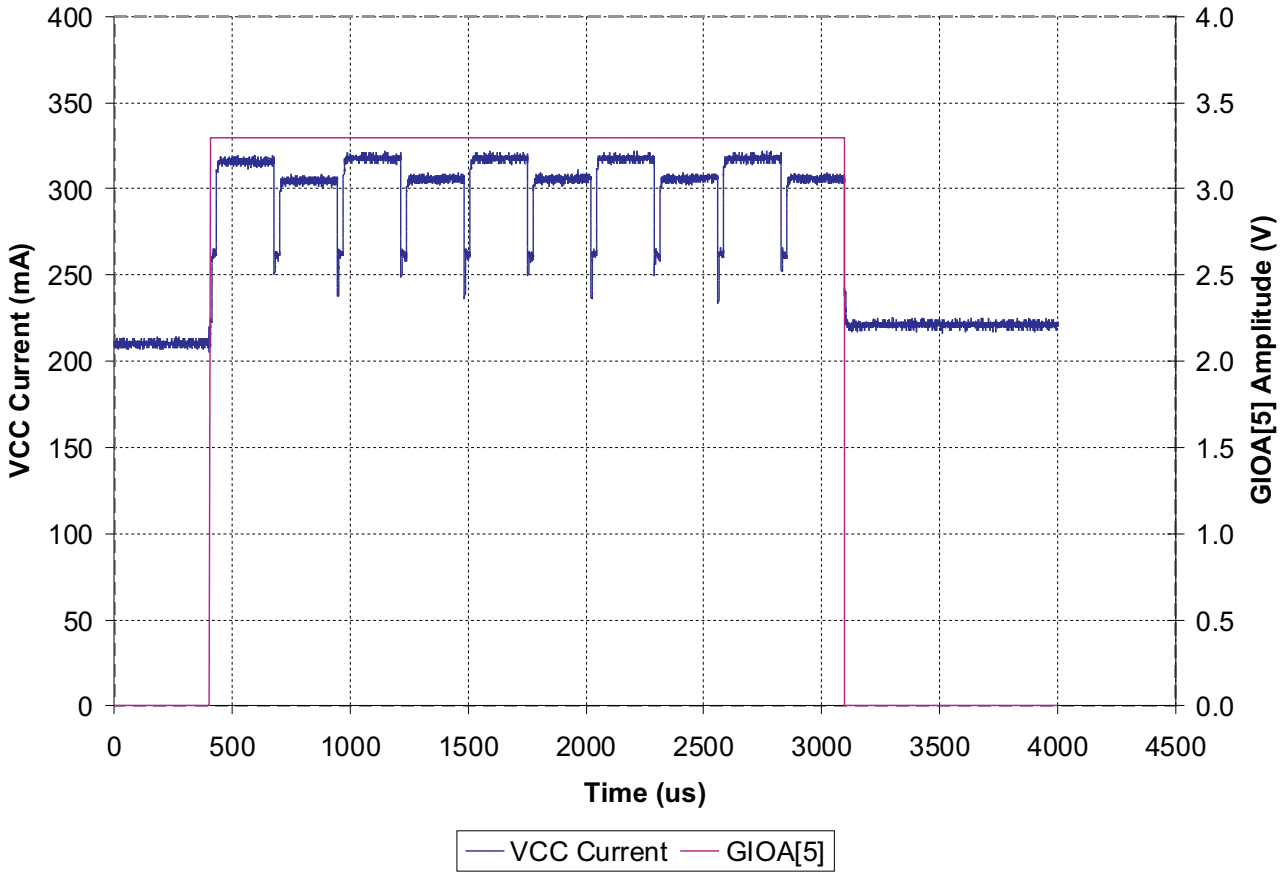
4 PBIST Test Duration and Current Consumption

4.1 PBIST Test Duration

For TMS570LS10x and TMS570LS20x series, the *PBIST RAM Grouping* table in [2] lists the test duration in clock cycles for any combination of RAM/ROM and algorithm. For STC/PBIST ROM, the unit is ROM clock period. For the 160Kbyte eSRAM, DMA and RTP RAM, the unit is HCLK clock period. For the other peripheral RAMs, the unit is VCLK. Note that the test time in this table does NOT include the time downloading PBIST code from PBIST ROM. Therefore, the actual test time is slightly higher than what is specified here. An empty cell means that an algorithm is not available for such a RAM/ROM. For example, the PBIST and STC ROM can only be tested through *triple slow read* and *triple fast read* algorithm.

In the attached example, the PBIST starts with GIOA[5] high and ends with GIOA[5] low. At the end of the test, GIOA[4] ties to 3.3 V if the algorithm runs successfully. Otherwise, GIOA[4] ties to ground. The test time can be measured as the GIOA[5] high duration.

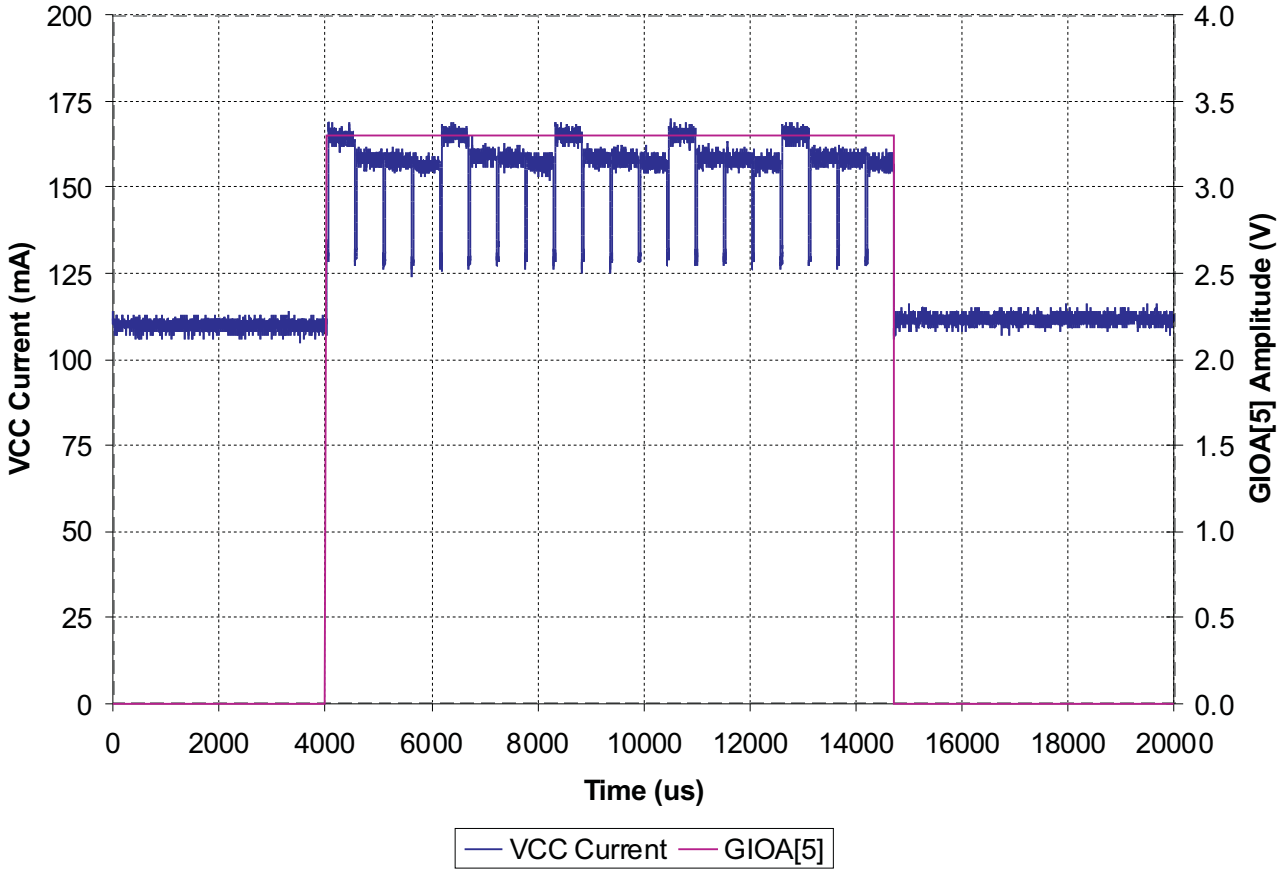
Figure 1 shows the waveform captured by the oscilloscope when the March13 algorithm is running on the main 160 Kbyte eSRAM in TMS570LS20216 (select '3' in the Hyper Terminal when running the attached example). The blue curve is the current measured on V_{CC} supply and the pink curve indicates when the PBIST test starts and stops. The current measured here is the total device current consumption on V_{CC} domain and not just the PBIST portion. Based on the *PBIST RAM Grouping* table in [2], the March13 algorithm on eSRAM takes 266320 HCLK cycles; HCLK is 100 MHz in this case. Therefore, the calculated test duration is 2.66 ms, which matches the measurement in Figure 1 very well.



Note: All the waveforms do not show the absolute worst case current consumption and are just an example.

Figure 1. PBIST Current Waveform Capture (March13 Algorithms on TMS570LS20216 eSRAM)

The same procedure also applies to TMS570LS3137 devices. Figure 2 shows the waveform captured by the oscilloscope when the March13 algorithm is running on the main 256 Kbyte eSRAM in TMS570LS3137 (select '3' in the Hyper Terminal when running the attached example). The blue curve is the current measured on V_{CC} supply and the pink curve indicates when the PBIST test starts and stops. The current measured here is the total device current consumption on V_{CC} domain and not just the PBIST portion. Based on the *PBIST RAM Grouping* table in [4], the March13 algorithm on eSRAM takes 266280 x 4 = 1,065,120 HCLK cycles; HCLK is 100 MHz in this case. Therefore, the calculated test duration is 10.65 ms, which matches the measurement in Figure 2 very well.

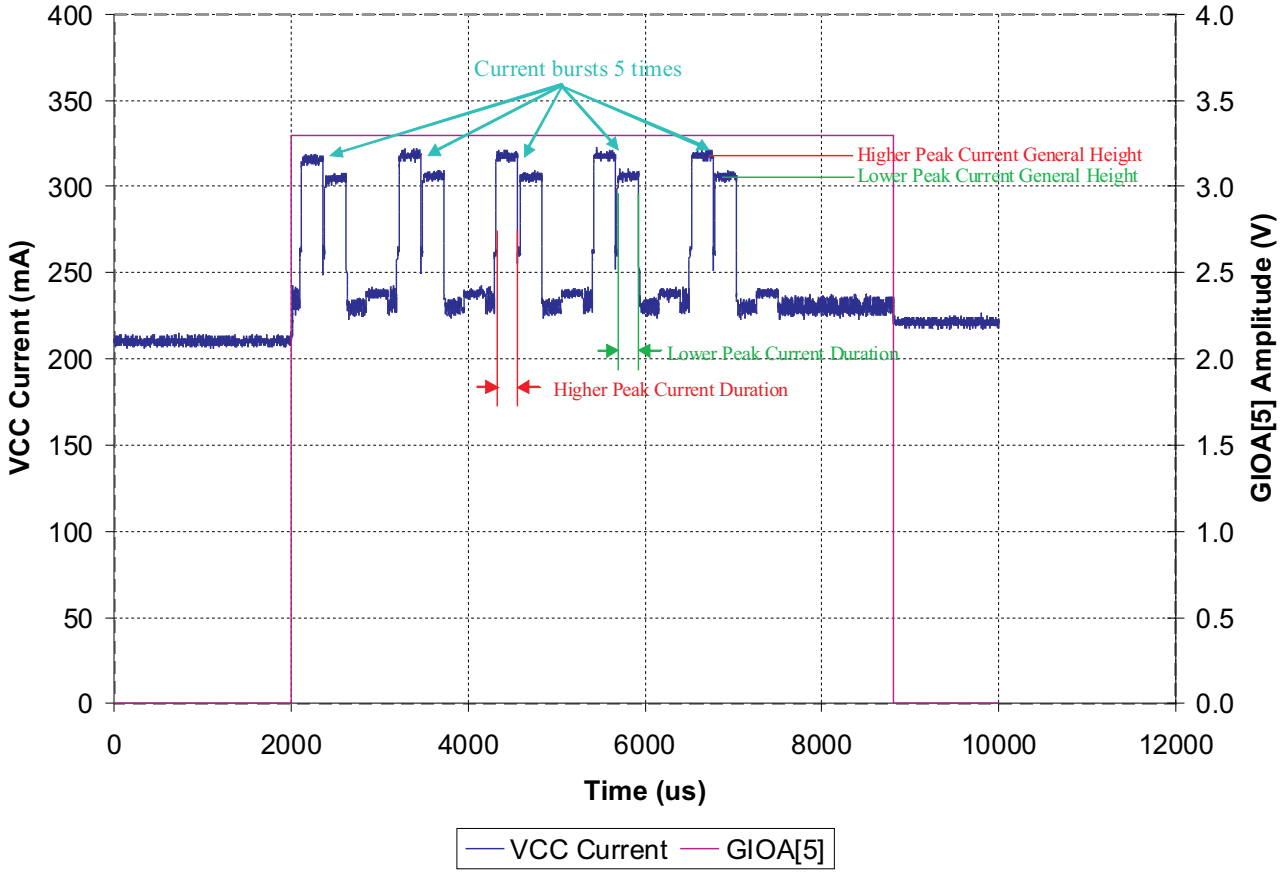


Note: All the waveforms do not show the absolute worst case current consumption and are just an example.

Figure 2. PBIST Current Waveform Capture (March13 Algorithms on TMS570LS3137 eSRAM)

The March13 algorithm is the most recommended algorithm for the memory self test. The waveform in Figure 3 illustrates that when running the March13 algorithm on all the available RAMs in TMS570LS20216 devices (select '2' in the Hyper Terminal when running the attached example). Based on the *PBIST RAM Grouping* table in [2], the March13 algorithm on all RAMs takes (12600 + 12600 + 6360 + 266320 + 50160 + 4200 + 8400 + 18960 + 25440 + 6480 + 37800 + 175040 = 624360) HCLK/VCLK cycles. Considering the 10 ns HCLK/VCLK cycles, the test duration is about 6.24ms. The measured test duration in Figure 3 is slightly longer than this number due to PBIST test microcode loading time.

The current in Figure 3 bursts for 5 times, which is the test pattern number of the March13 algorithm. These burst current peaks are generated by the 160 Kbyte eSRAM test. This also applies to the other algorithms except that the other algorithms (see Figure 5) only burst twice because they only have two test patterns. Each burst (period) is composed by two parts, beginning with a higher current peak and ending with a lower current peak.

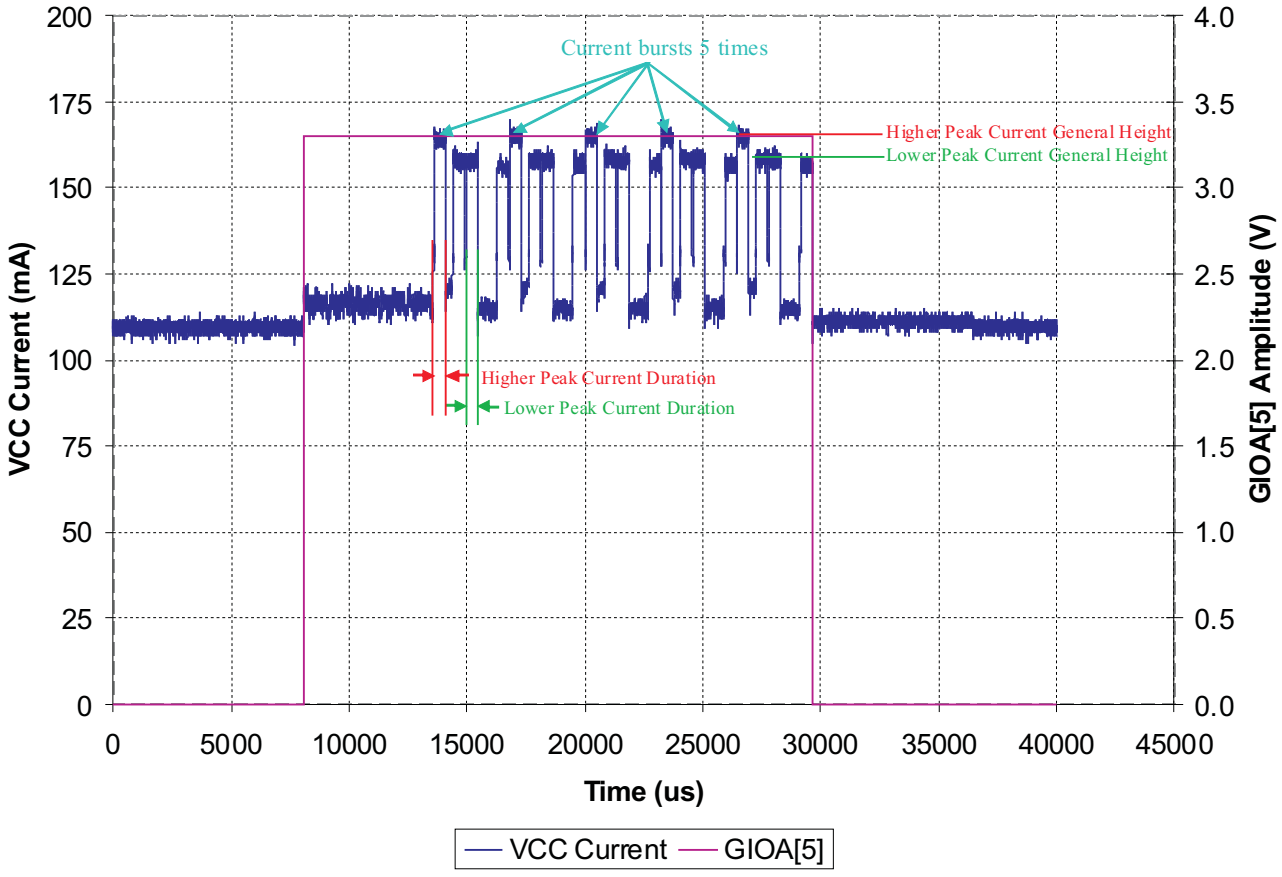


Note: All the waveforms do not show the absolute worst case current consumption and are just an example.

Figure 3. PBIST Current Waveform Capture (March13 Algorithms on All TMS570LS20216RAMs)

The same procedure also applies to TMS570LS3137 devices. The waveform in Figure 4 illustrates that when running the March13 algorithm on all the available RAMs in TMS570LS3137 devices (select '2' in the Hyper Terminal when running the attached example). Based on the *Timing Specifications for Flash* table in [4], the March13 algorithm on all RAMs takes (25200x3 + 266280x4 + 33440x3 + 12560 + 4200 + 18960 + 31680 + 6480 + 37800 + 75400 + 133160 + 4200 + 31680 + 6480 + 8700 + 6360 + 133160 +70840 (reserved RAM for other peripherals) = 1,822,700) HCLK/VCLK cycles. Considering the 10 ns HCLK/VCLK cycles, the test duration is about 18.23 ms. The measured test duration in Figure 4 is slightly longer than this number due to PBIST test microcode loading time.

The current in Figure 4 bursts for 5 times, which is the test pattern number of the March13 algorithm. These burst current peaks are generated by the 256 Kbyte eSRAM test. This also applies to the other algorithms except that the other algorithms (see Figure 6) only burst twice because they only have two test patterns. Each burst (period) is composed by two parts, beginning with a higher current peak and ending with three lower current peaks.



Note: All the waveforms do not show the absolute worst case current consumption and are just an example.

Figure 4. PBIST Current Waveform Capture (March13 Algorithms on All TMS570LS3137 RAMs)

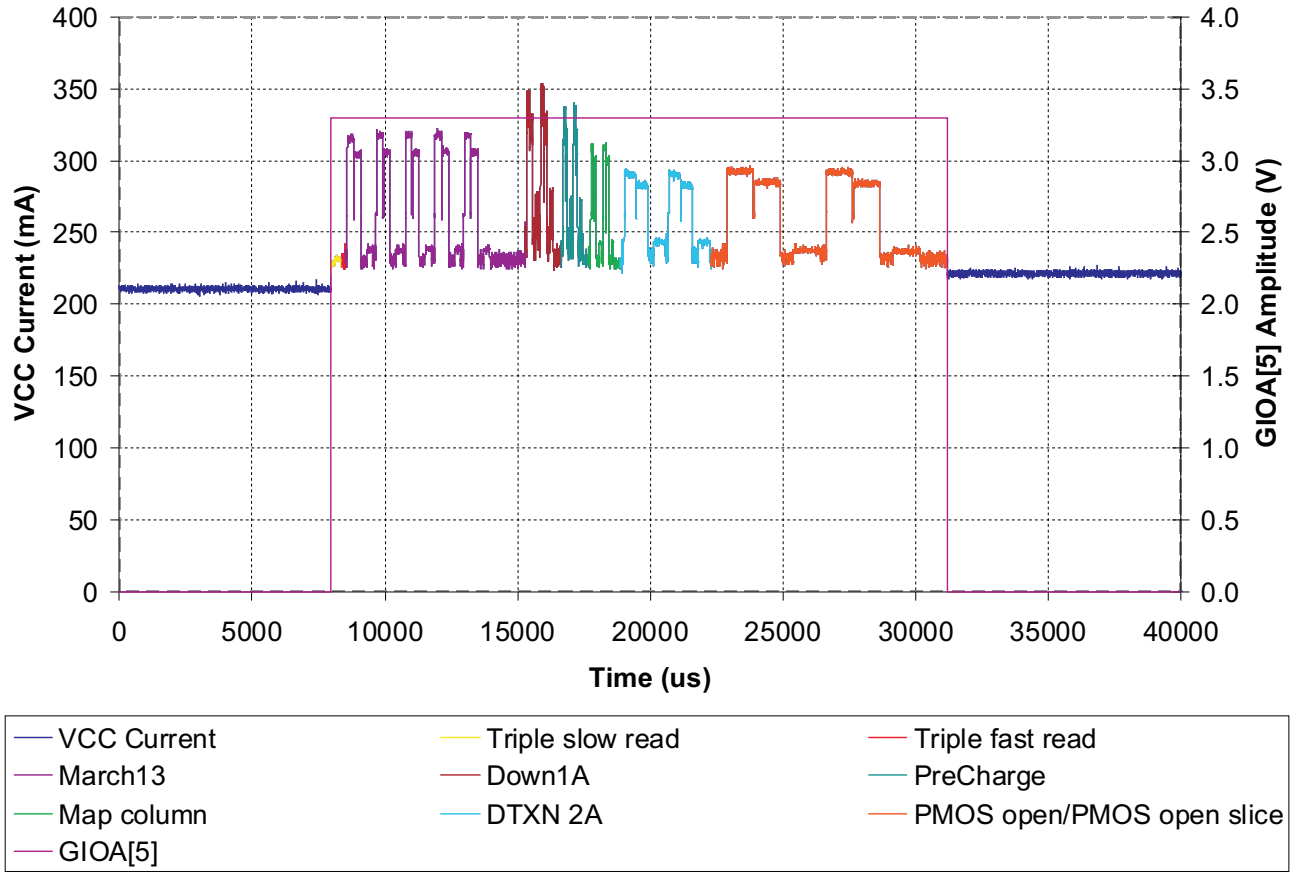
4.2 PBIST Test Current Consumption

Usually, running PBIST consumes more current in V_{CC} domain than running in normal mode at the same HCLK/VCLK frequency. The PBIST controller executes each selected algorithm on each valid memory group sequentially until all the algorithms are executed. eSRAM banks are run in parallel by default doing this the PBIST test time is reduced but the current is increased.

Across different ROM/RAM groups, the eSRAM consumes the highest current because it has the highest number of banks running in parallel – 10 for ROM/RAM group six and 20 for ROM/RAM group 15 in TMS570LS20216 devices. Note that group 6 and group 15 (see the *PBIST RAM Grouping* table in [2]) are duplicated. Running either of them will test the entire 160 Kbyte eSRAM. Group 15 takes only half the time of group 6 but consumes 25% more current. Group 15 is only available in RAM Override-Mode Disabled.

Figure 5 shows the waveform captured by the oscilloscope when all algorithms are running on all available RAM/ROMs in TMS570LS20216 (select '1' in the Hyper Terminal when running the attached example). The current amplitude and duration of different algorithm are represented by different colors. Across different algorithms, Down1A algorithm consumes the highest current.

In each algorithm, the current peaks are generated by the 160 Kbyte eSRAM test. Table 1 illustrates the duration of each peak and general height in Figure 5. For the definition of duration and general height, see Figure 3.



Note: All the waveforms do not show the absolute worst case current consumption and are just an example.

Figure 5. PBIST Current Waveform Capture (All Algorithms on All TMS570LS20216 ROM/RAMs)

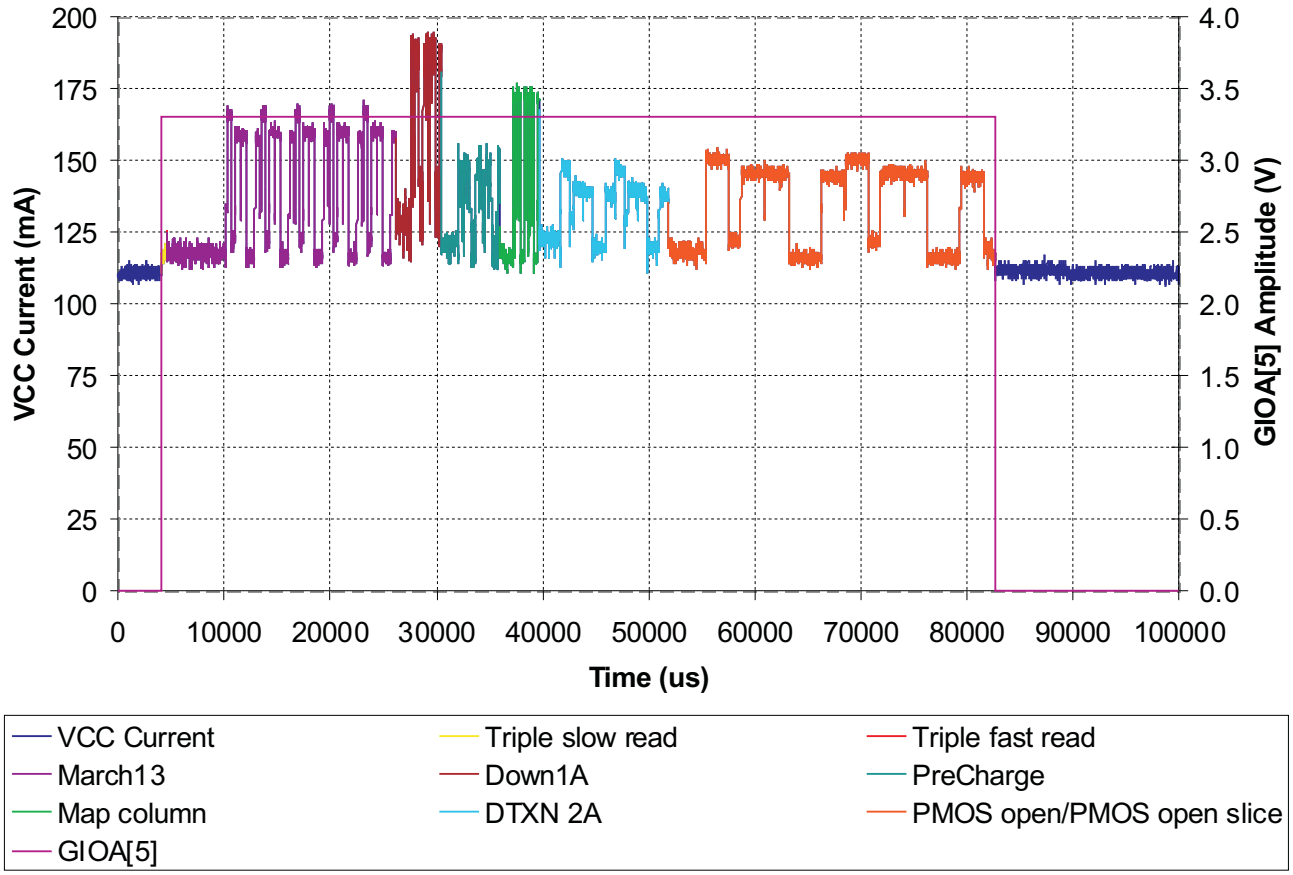
Table 1. Current Peak Height and Duration of TMS570LS20216[(1)]

Algorithm	NO of Bursts	Peak Current Duration		Peak Current General Height	
		Higher Peak	Lower Peak	Higher Peak	Lower Peak
March 13	5	2.66 ms	2.66 ms	315 mA	304 mA
Down1A	2	130 us	130 us	350 mA	330 mA
PreCharge	2	103 us	103 us	340 mA	325 mA
Map column	2	83 us	83 us	310 mA	300 mA
DTXN 2A	2	450 us	450 us	293 mA	285 mA
PMOS open	2	1.03 ms	1.03 ms	294 mA	285 mA

(1) All the waveforms do not show the absolute worst case current consumption and are just an example.

The same procedure also applies to TMS570LS3137 devices. Figure 6 shows the waveform captured by the oscilloscope when all algorithms are running on all available RAM/ROMs in TMS570LS3137 devices (select '1' in the Hyper Terminal when running the attached example). The current amplitude and duration of different algorithm are represented by different colors. Across different algorithms, Down1A algorithm consumes the highest current.

In each algorithm, the current peaks are generated by the 256 Kbyte eSRAM test. Table 2 illustrates the duration of each peak and general height in Figure 6. For the definition of duration and general height, see Figure 4.



Note: All the waveforms do not show the absolute worst case current consumption and are just an example.

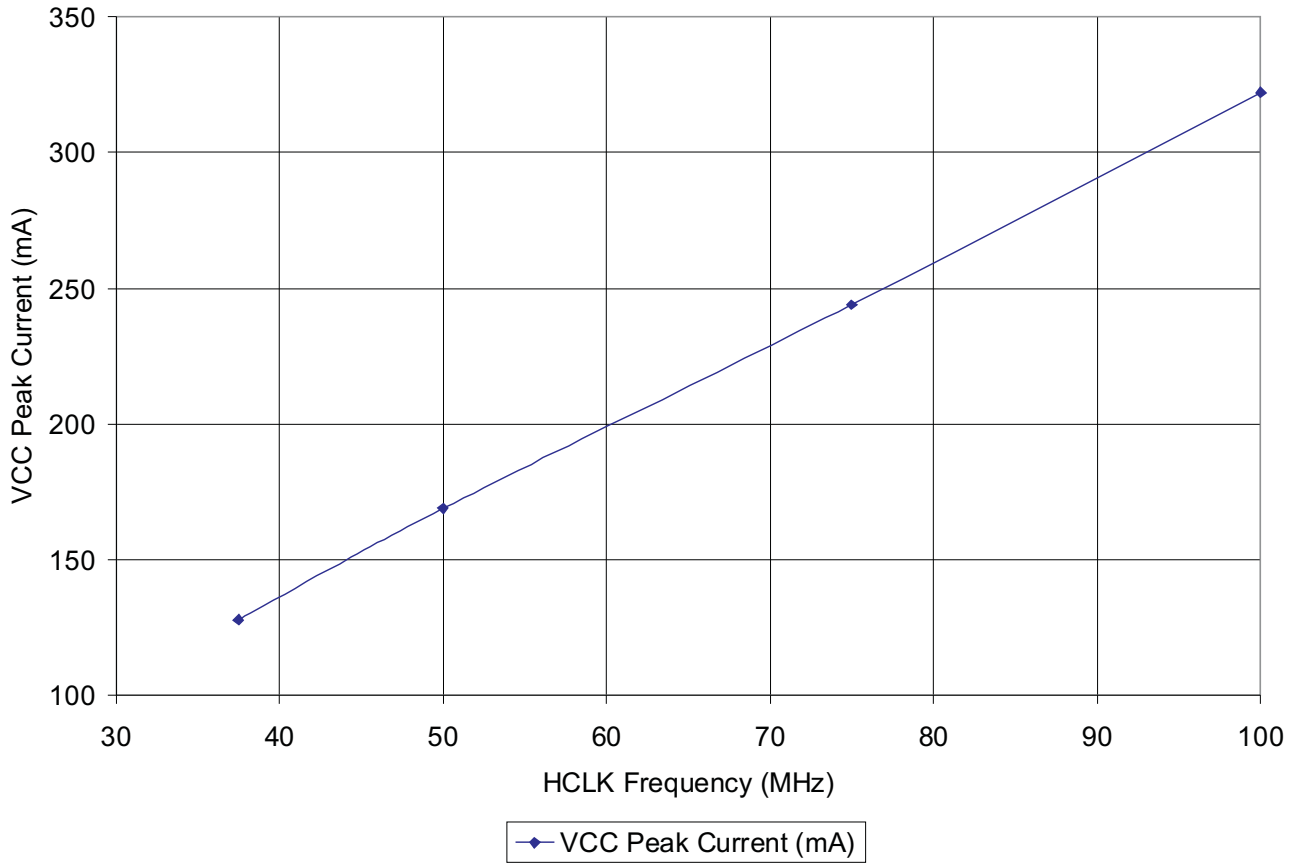
Figure 6. PBIST Current Waveform Capture (All Algorithms on All TMS570LS3137 ROM/RAMs)

Table 2. Current Peak Height and Duration of TMS570LS3137[(1)]

		Peak Current Duration		Peak Current General Height	
Algorithm	**NO of Bursts**	**Higher Peak**	**Lower Peak (3 times)**	**Higher Peak**	**Lower Peak**
March 13	5	4.9 ms	4.9 ms	167 mA	155 mA
Down1A	2	270 µs	270 us	197 mA	187 mA
PreCharge	2	vague to measure	vague to measure	154 mA	154 mA
Map column	2	160 µs	160 us	175 mA	175 mA
DTXN 2A	2	860 µs	860 µs	147 mA	139 mA
PMOS open	2	2.21 ms	2.21 ms	150 mA	145 mA

(1) All the waveforms do not show the absolute worst case current consumption and are just an example.

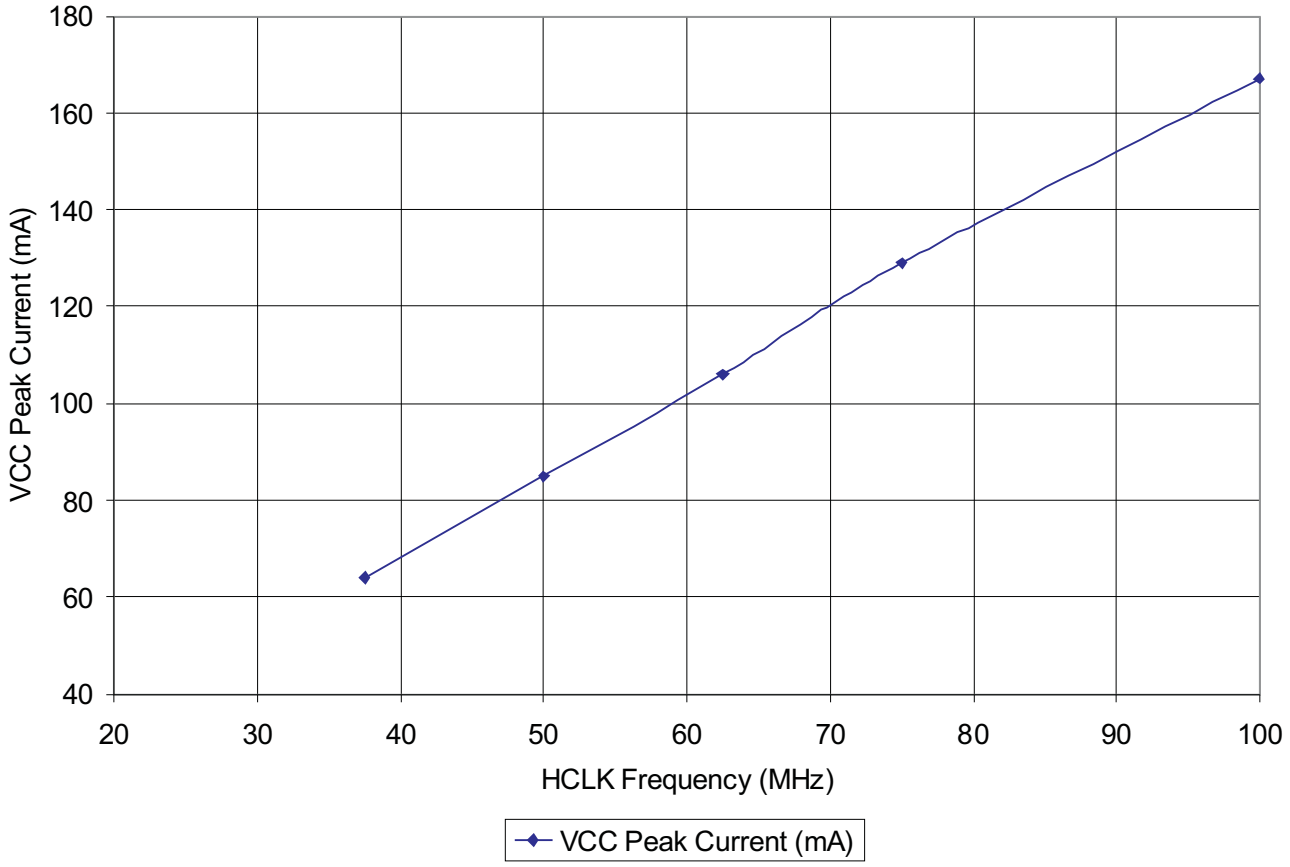
Running at a lower HCLK frequency is a feasible method to reduce the PBIST current. One caveat is the longer test time, which is inverse proportional to the HCLK/ROM clock frequency. Another caveat is that the memory is not tested at speed. Figure 7 shows the V_{CC} PBIST peak current versus HCLK frequency measured on one TMS570LS20216 unit.



Note: All the waveforms do not show the absolute worst case current consumption and are just an example.

Figure 7. PBIST Peak Current versus HCLK Frequency (March13 Algorithms on TMS570LS20216 eSRAM)

Figure 8 shows the V_{CC} PBIST peak current versus HCLK frequency measured on one TMS570LS3137 unit.



Note: All the waveforms do not show the absolute worst case current consumption and are just an example.

Figure 8. PBIST Peak Current versus HCLK Frequency (March13 Algorithms on TMS570LS3137 eSRAM)

5 Example

The attached example communicates with the PC through SCI using RS232 protocols. You can type the command in Hyper Terminal to use PBIST to test all the RAM/ROM with all the available algorithms or test the main 160/256 Kbyte SRAM with the March13 algorithm.

It includes the following source files:

Table 3. Source Files in Project

File Name	Description
Intvecs.asm	Interrupt vectors setup file.
Boot.asm	Initialize stack, call _init and _main.
Swi_util.asm	SWI interrupt service routine.
System_Init	Initialize the system, enable peripherals.
SCI1_RS232.c	RS232 functions.
TMS570_PBIST_Test.c	Demonstrate the PBIST.
TMS570_PBIST_Test.pjt	Project file.

To build the demo with TMS570LS10x and TMS570LS20x series, include this line in the main file:

```
#define Tech_130nm
```

Otherwise, to build the demo with TMS570LS21x and TMS570LS31x series, remove or comment the same line.

The following line in boot.asm and the function associated with this line are for the ESM Errata of TMS570LS21x and TMS570LS31x Rev0 devices. It will be fixed in RevA silicon. Therefore, this line should be removed/commented once the RevA silicon is released.

```
; to clear the ESM error flags
BL _ESM_CCM_ERROR_CLR
```

To run the demo:

- Hardware

  Connect the TI EVM board USB port to a PC; open a HyperTerminal; configure the right COM port and the baud rate to be 19.2 kHz, no parity, one stop bit.

  NOTE: SCI1/LIN1 receive/transmit pins are mapped to different position of the 337 pin BGA in TMS570LS10x and TMS570LS20x series (W12, V12) and TMS570LS21x and TMS570LS31x series (A7, B7).

- Software

  Program TMS570_PBIST.out to the device flash using nowFlash tool; reset the device.

  Then, the following menu prompts out from the HyperTerminal:

```
-Please input a character to start PBIST testing:-
-Run all algorithms on all RAM/ROMs     ------------ 1
-Run March13 on all Single/Dual Port RAM------------ 2
-Run March13 on 160K/256K ESRAM only    ------------ 3
```

After that, you should input the function index in the HyperTerminal to run the function. The PBIST starts with GIOA[5] high and ends with GIOA[5] low. At the end of the test, GIOA[4] ties to 3.3 V if no error is detected. Otherwise, GIOA[4] ties to ground. Meanwhile, the HyperTerminal also outputs information about the PBIST test status.

6 References

1. *TMS570LS Series Microcontroller Technical Reference Manual* (SPNU489)
2. *TMS570LS20216, TMS570LS20206, TMS570LS10216, TMS570LS10206, TMS570LS10116, TMS570LS10106 Data Sheet* (SPNS141)
3. *TMS570LS31/21 16/32-Bit RISC Flash Microcontroller Technical Reference Manual* (SPNU499)
4. *TMS570LS12004, TMS570LS30376U, RM5H3071 Data Sheet* (SPNS160)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Mobile Processors	www.ti.com/omap		
Wireless Connectivity	www.ti.com/wirelessconnectivity		
	TI E2E Community Home Page		e2e.ti.com

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