SILICON LABS

Process Change Notice #1409091

User Registration
Register today to create your account on Silabs.com. Your personalized profile allows you to receive technical document updates, new product announcements, "how-to" and design documents, product change notices (PCN) and other valuable content available only to registered users. http://www.silabs.com/profile

PCN Date: 9/9/2014	**Effective Date:** 10/15/2014	
Title: New supplier for Si3217x-B high voltage die - Addendum to PCN#1409031		
Originator: Rick Bye	**Phone:** 512-532-5740	**Dept:** APS Marketing
Customer Contact: Kathy Haggar	**Phone:** 512-532-5261	**Dept:** Sales

PCN Type:

☐ **Datasheet**	☒ **Foundry**	☐ **Packing**
☐ **Product Revision**	☐ **Assembly**	☐ **Labeling**
☐ **Discontinuance**	☐ **Test**	☐ **Other**

Last Order Date: Not Applicable

PCN Details

Description of Change:
This PCN is issued as an addendum to PCN#1409031 for New supplier for Si3217x-B high voltage die. This addendum updates the white paper in Appendix B to correct the address of the HVIC_STATE register to be 1447 (not 1471 as indicated in the orginal PCN) and comments to code examples for clarity.

In an effort to minimize the effect on operations due to the supply disruption at the Telefunken wafer fab, Silicon Labs will accelerate the availability of products from Vanguard, the 2nd source high voltage wafer fab.

The accelerated schedule requires qualified product volume from Vanguard earlier than originally planned. This effective date is earlier than would allow Silicon Labs to follow its standard Product Change Notification process.

Additional qualification data will be provided upon request as available. Prequalification samples of products are available for order now and early PCN acceptance is encouraged.

Silicon Labs remains very conscious of its commitment to provide the highest quality parts and is not relaxing its stringent qualification processes.

Reason for Change:
Supply disruption at the Telefunken wafer fab.
This addendum corrects a register address and adds comments to code examples for clarity in the white paper in Appendix B.

Impact on Form, Fit, Function, Quality, Reliability:

There is no change to Fit, Quality, or Reliability of the Si3217x-B devices. All devices will comply fully with datasheet parameters and quality levels.

The only change to form is that the new Si3217x-B product uses new high voltage die from Vanguard and a new version of the low voltage die from SMIC (Semiconductor Manufacturing International Corporation), modified for compatibility with the Vanguard HV die.

To achieve functional compatibility, a minor software update is required for users of the Si3217x Revision B device. These software changes address minor differences between suppliers (Telefunken, Vanguard) of the high-voltage linefeed device (HVIC) that is part of the Si3217x. These changes will ensure correct operation of all supplier versions of Si3217x Revision B.

Software changes will be required if any of the following are applicable:

1. Using a software design based on ProSLIC® API Version 6.6.0 or earlier
2. Implementing one or more of the following MLT tests based on the ProSLIC® MLT API
    a. Resistive Faults Test
    b. Capacitance Test
    c. REN Capacitance Test

Please refer to the attached white paper in Appendix B for more details.

Product Identification:

Si32170-B-FM1	Si32170-B-FM1R	Si32170-B-GM1	Si32170-B-GM1R
Si32171-B-FM1	Si32171-B-FM1R	Si32171-B-GM1	Si32171-B-GM1R
Si32176-B-FM1	Si32176-B-FM1R	Si32176-B-GM1	Si32176-B-GM1R
Si32177-B-FM1	Si32177-B-FM1R	Si32177-B-GM1	Si32177-B-GM1R
Si32178-B-FM1	Si32178-B-FM1R	Si32178-B-GM1	Si32178-B-GM1R
Si32179-B-FM1	Si32179-B-FM1R	Si32179-B-GM1	Si32179-B-GM1R

Last Date of Unchanged Product: 10/15/2014

Qualification Samples:

Samples of parts manufactured in Vanguard are available upon request. Please contact your local sales representative for samples. A list of Silicon Labs sales representatives is available at www.silabs.com
Please ensure that your sales representative understands that you are requesting parts specifically from the Vanguard fab.

Customer Early Acceptance Sign Off: **Customers may approve early PCN acceptance by completing the information below:** **Early Acceptance: Date: ______________________________** **Name: ____________________________** **Company: _________________________** **Email your early Acceptance approval to: katherine.haggar@silabs.com**
Qualification Data: The preliminary Qualification Report is attached in Appendix A. Additional qualification data will be provided upon request as available. Prequalification samples of products are available for order now and early PCN acceptance is encouraged.

Appendix A

Si3217x-B Vanguard Qualification Report

Part Rev B, Vanguard Fabrication, ASEKR Assembly except as noted							
Test Name	**Test Condition**	**Qualification**	**Lot ID**	**Fail/Pass**	**Notes**	**Summary**	**Status**
Test Group A - Accelerated Environment Stress Tests							
HAST	JA110 110°C, 85%RH Vcc=3.6V, 264 hours	3 lots, N=>25	Q36012 Q35961 Q35884	0/30 0/30 0/30	1 1 1	3 lots 0/90	Pass
Temp Cycle	JA104 Cond C: -65°C to 150°C 500 cycles	3 lots, N=>25	Q34970 Q34969 Q34209	0/29 0/30 0/30	1 1 1	3 lots 0/89	Pass
HTSL	JA103 150°C, 1000hr	3 lots, N=>25	Q34968 Q34967 Q34793	0/29 0/30 0/30	1 1 1	3 lots 0/89	Pass
Test Group B - Accelerated Lifetime Simulation Tests							
HTOL	JA108 125°C, Dynamic Vcc=3.6V, 1000 hours	3 lots, N=>77	Q36149 Q34746 Q34934	f/p 0/77 0/83		2 lots 0/160	Ongoing Due 10/13/14
LTOL	JA108 -10°C, Dynamic Vcc=3.6V, 1000 hours	1 lot, N=>32	Q35041	0/38		1 lot 0/38	Pass
ELFR	JA108 125°C, Dynamic Vcc=3.6V, 48 hours	3 lots, N=>500	Q35670 Q35660 Q35639	0/517 0/520 0/523		3 lots 0/1560	Pass
Test Group E - Electrical Verification							
ESD-HBM	JA114	1 lot, N=>3	Q36114				±2000 V
ESD-CDM	JC101	1 lot, N=>3	Q36115				±1500 V
Latch Up	JESD78 ±200mA	1 lot, N=>6	Q36116	25 °C			Pass

Notes:

1. Parts are Pre-conditioned at MSL3/260°C

Approved by: K Torres   1 of 2   Prepared on: 30-Aug-2014

Si3217x-B Vanguard Qualification Report

Part Rev B, Vanguard Fabrication, ASEKR Assembly except as noted

This report applies to the following part numbers:

Si32170-B-FM1	Si32170-B-FM1R	Si32170-B-GM1	Si32170-B-GM1R
Si32171-B-FM1	Si32171-B-FM1R	Si32171-B-GM1	Si32171-B-GM1R
Si32176-B-FM1	Si32176-B-FM1R	Si32176-B-GM1	Si32176-B-GM1R
Si32177-B-FM1	Si32177-B-FM1R	Si32177-B-GM1	Si32177-B-GM1R
Si32178-B-FM1	Si32178-B-FM1R	Si32178-B-GM1	Si32178-B-GM1R
Si32179-B-FM1	Si32179-B-FM1R	Si32179-B-GM1	Si32179-B-GM1R

Appendix B, White Paper, Revised September 9, 2014

Si3217x Revision B Software Changes for Compatibility with Telefunken and Vanguard HVICs

A minor software update is required for users of the Si3217x Revision B device. These software changes address minor differences between suppliers (Telefunken, Vanguard) of the high-voltage linefeed device (HVIC) that is part of the Si3217x. These changes will ensure correct operation of all supplier versions of Si3217x Revision B.

Software changes will be required if any of the following are applicable

1. Using a software design based on ProSLIC® API Version 6.6.0 or earlier
2. Supporting one or more of the following MLT tests based on the ProSLIC® MLT API
   a. Resistive Faults Test
   b. Capacitance Test
   c. REN Capacitance Test

This document describes the technical details and software changes required.

Technical Overview

This first change relates to the DCDC converter startup. On the Vanguard version of the HVIC, one additional RAM parameter must be updated prior to starting the converter if ProSLIC® API Version 6.6.0 or earlier is being used. For those users that do not use the ProSLIC® API, please contact Silicon Laboratories to see if your code requires this change.

The second change relates to test features used during the MLT testing. The ProSLIC® MLT API test suite utilizes a test feature that allows the software to manually control the state of on-chip test switches that are located on the high voltage linefeed device (HVIC). Specific to MLT testing, switches are present that allow the open lead in the TIP-OPEN or RING-OPEN linefeed states to be connected to GND. This feature is utilized in the following tests

- `ProSLIC_mlt_resistive_faults()`
- `ProSLIC_mlt_capacitance()`
- `ProSLIC_mlt_ren_capacitance()`

Although the functional and electrical behavior of the switches from each HVIC suppliers (Telefunken, Vanguard) is equivalent, there is a small difference in how these switches are manually controlled using the HVIC_STATE (1447) RAM location, thus requiring a minor software change.

The following table details the values written to HVIC_STATE (1447) that are different between HVIC suppliers

Test Switch Connection	HVIC_STATE (Telefunken)	HVIC_STATE (Vanguard)
TIP to GND	0x2F0000 or 0x6F0000	0x02F000
RING to GND	0x1F0000	0x82F000

Software Modification Options

There is a single change required to address the additional parameter update prior to starting the DCDC converter, however, the user has several options for implementing the required changes in their MLT software design. Each option is behaviorally equivalent and ensures their software design supports both Telefunken and Vanguard variants of the HVIC on Si3217x Revision B.

DCDC Converter Startup Code Change

For all Si3217xB applications using ProSLIC® API Version 6.6.0 or earlier, the following code change should be made

File: *si3217x_intf.c*
Function: *Si3217x_PowerUpConverter()*

Original Code:

```
WriteRAM(pProHW,pProslic->channel,PD_DCDC,0x700000L);
WriteRAM(pProHW,pProslic->channel,PD_DCDC,0x600000L);
Delay(pProTimer,50);
```

New Code:

```
WriteRAM(pProHW,pProslic->channel,1430,0x200000L);
WriteRAM(pProHW,pProslic->channel,PD_DCDC,0x700000L);
WriteRAM(pProHW,pProslic->channel,PD_DCDC,0x600000L);
Delay(pProTimer,50);
```

For all Si3217xB applications that do not use the ProSLIC® API, please contact Silicon Laboratories to see if your code will require this change.

MLT Code Changes

The user may use one of the following methods for implementing the necessary changes to their software

1. Upgrade to ProSLIC® MLT API version 2.4.0
2. Explicitly modify existing MLT code
3. Embed MLT software changes in user's WriteRAM wrapper function

MLT Code Change Option 1: Upgrade to Latest ProSLIC® MLT API Version

The user may upgrade their design to use ProSLIC® MLT API version 2.4.0 to address all necessary changes.

MLT Code Change Option 2: Explicit Code Modifications

The user may make explicit changes to their existing MLT code base to implement all necessary changes, as enumerated below

MLT Code Change #1:
File: *si3217x_mlt.c*
Function: *gndOpenTerm()*

Original Code:

```
WriteRAM(pProHW,pProslic->channel,MLT_COMM_RAM_HVIC_STATE, 0x6F0000L);
```

New Code:

```
uInt8 rev_test; /* Declare at top of function */
rev_test = (uInt8)((ReadRAM(pProHW,pProslic->channel,1791)>>20)&0x000000FFL);
if(rev_test == 0x10)
{
    WriteRAM(pProHW,pProslic->channel,MLT_COMM_RAM_HVIC_STATE, 0x2F000L);
}
else
{
     WriteRAM(pProHW,pProslic->channel,MLT_COMM_RAM_HVIC_STATE, 0x6F0000L);
}
```

MLT Code Change #2:
File: *si3217x_mlt.c*
Function: *measRrg(), case 7*

Original Code:

```
pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE, pState->hvic_state_save | 0x2f0000L);
```

New Code:

```
uInt8 rev_test; /* Declare at top of function */
rev_test = (uInt8)((pProSLICMLT->ReadRAM(pMLT_HW,pMLT_CHAN, 1791)>>20)&0x000000FFL);
if(rev_test == 0x10)
{
    pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE, pState->hvic_state_save | 0x2f000L);
}
else
{
    pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE, pState->hvic_state_save | 0x2f0000L);
}
```


Original Code:

```
pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE,pState->hvic_state_save | 0x1f0000L);
```

New Code:

```
uInt8 rev_test;  /* Declare at top of function */
rev_test = (uInt8)((pProSLICMLT->ReadRAM(pMLT_HW,pMLT_CHAN, 1791)>>20)&0x000000FFL);
if(rev_test == 0x10)
{
    pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE,pState->hvic_state_save | 0x82f000L);
}
else
{
    pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE,pState->hvic_state_save | 0x1f0000L);
}
```

MLT Code Change #4:
File: *si3217x_mlt.c*
Function: *si3217x_mlt_capacitance(), case 4*

Original Code:

```
pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE,pState->ram1447 | 0x2f0000L);
```

New Code:

```
uInt8 rev_test;  /* Declare at top of function */
rev_test = (uInt8)((pProSLICMLT->ReadRAM(pMLT_HW, pMLT_CHAN, 1791)>>20)&0x000000FFL);
if(rev_test == 0x10)
{
    pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE,pState->ram1447 | 0x2f000L);
}
else
{
    pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE,pState->ram1447 | 0x2f0000L);
}
```

MLT Code Change #5:
File: *si3217x_mlt.c*
Function: *si3217x_mlt_capacitance(), case 8*

Original Code:

```
pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE,pState->ram1447 | 0x1f0000L);
```

New Code:

```
uInt8 rev_test; /* Declare at top of function */
rev_test = (uInt8)((pProSLICMLT->ReadRAM(pMLT_HW, pMLT_CHAN, 1791)>>20)&0x000000FFL);
if(rev_test == 0x10)
{
    pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE,pState->ram1447 | 0x82f000L);
}
else
{
    pProSLICMLT->WriteRAM(pMLT_HW,pMLT_CHAN,MLT_COMM_RAM_HVIC_STATE,pState->ram1447 | 0x1f0000L);
}
```

MLT Code Change Option 3: Embedded Changes in WriteRAM Wrapper Function

A third, and possibly the easiest modification is to embed the changes to the MLT portion of the code in the user's WriteRAM Wrapper function.

The embedded MLT code change will monitor the RAM address written to and only take action if

1. Address 1447 (HVIC_STATE) is being written and
2. The Vanguard HVIC is identified

Since the content of the function pointed to by `ctrl_WriteRAMWrapper()` is user dependent, below is one example of how this functionality may be added. In this example, `spiWriteRAM()` and `spiReadRAM()` are the user's functions for performing the RAM read/write tasks.

Original Code:

```
int ctrl_WriteRAMWrapper (ctrl_S * hctrl, uInt8 channel, uInt16 ramAddr, ramData data)
{
    spiWriteRAM(ramAddr,data,channel);
    return 0;
}
```

New Code:

```
int ctrl_WriteRAMWrapper (ctrl_S * hctrl, uInt8 channel, uInt16 ramAddr, ramData data)
{
uInt8 rev_test; /* Declare at top of function */

    if(ramAddr == 1447)
    {
      rev_test = (uInt8)((spiReadRAM(1791,channel)>>20)&0x000000FF);
       if(rev_test == 0x10)
       {
           if((data&0x2F0000)==0x2F0000)        data = 0x2F000L;
           else if((data&0x1F0000)==0x1F0000)   data = 0x82F000L;
       }
    }
    spiWriteRAM(ramAddr,data,channel);
    return 0;
}
```


