

Customer Process Change Notification Form

PCN-2013-497

Notification Date: 6/4/2013

Part Information

Cirrus Logic Part # / Quantity

CS5484-INZ(R)

PCN Effective Date: 5/29/2013

Lot Effective Date: 5/29/2013

Cirrus P/N Change: ☐ Yes ☒ No

(Contact the sales representative for availability of samples if applicable)

If yes, provide new part number:

Package Mark Change: ☐ Yes ☒ No

If Yes, briefly explain:

[Any Fab, Assembly, or Design changes results in pack mark changes, please provide detail]

Reason for Change:

- | | |
|---|---|
| <input type="checkbox"/> Design/New Rev | <input type="checkbox"/> Fab Process |
| <input type="checkbox"/> Assembly Site | <input type="checkbox"/> Additional Assembly Source |
| <input type="checkbox"/> Fab Site | <input type="checkbox"/> Assembly Process |
| <input checked="" type="checkbox"/> Other (Specify) | <input type="checkbox"/> Additional Fab Source |

Fix Errata

Description of Change:

- | | |
|--|--|
| <input checked="" type="checkbox"/> Fix Errata | <input type="checkbox"/> Yield Enhancement |
| <input type="checkbox"/> Fix Known Bug | <input type="checkbox"/> Performance Improvement |
| <input type="checkbox"/> Other (specify) | |

1. Erratum to use Sample Count Averaging mode and MCU calculated sample count if DSP write protection (DSP_LCK) is required.
2. Erratum to disable an internal circuit to ensure consistent onset reference state during power up.
3. Erratum to reset the serial port when communication has been interrupted.
4. Erratum to enable use of a mask checksum value.

See attached.

Quality and Reliability Impact:

Qualification Data Required? ☐ Yes ☒ No

Data Sheet Change Required? ☐ Yes ☒ No

If Yes, briefly explain:

Software Change Required? ☒ Yes ☐ No

If Yes, briefly explain:

See attached.

Customer Acknowledgement and Agreement

Acknowledged and Agreed as of the Date written below:

Customer Company _____
Name:

Customer Contact _____
Name:
Title: _____

Signature: _____

Date:

☒ Customer Agreed ☐ Customer Reject ☐ Not Applicable

Customer Comments:



Title:

CUSTOMER PROCESS CHANGE NOTIFICATION FORM

Digitally Signed By:

Doc No:

Rev:

4-QUAL-00017

D

Cirrus Logic, 800 W. 6th St., Austin, Texas 78701, TEL: (512) 851-4000

Errata: CS5484 Rev B2 Silicon

(Reference CS5484 data sheet revision DS981F3 dated MAR'13)

Erratum 1 - Using Line-cycle Synchronized Averaging Mode with DSP_LCK[4:0]

Description

Setting the DSP_LCK[4:0] bits in the *RegLock* register to 0x16 enables DSP write-protection mode. When DSP write-protection mode is enabled and the CS5484 is operating in Line-cycle Synchronized Averaging mode, the *SampleCount* register will not be updated by the CS5484 DSP core according to the line frequency.

Workaround

If DSP write-protection mode is used in normal operation, Fixed Number of Samples Averaging mode should be used and the host processor should update the *SampleCount* register according to the line frequency in order to minimize the ripple in all low-rate calculation results.

The host program should include the following steps in a 1-second DRDY loop:

1. Set the HOST_LCK[4:0] bits in the *RegLock* register (page 0, address 34) to 0x09, which disables host write-protection mode.
2. Read the *Epsilon* register (page 16, address 49).
3. Write $50 \times (8388608/\text{Epsilon})$ to the *SampleCount* register (page 16, address 51).
4. Set the HOST_LCK[4:0] bits in the *RegLock* register (page 0, address 34) to 0x16, which enables host write-protection mode.

Note: Steps 1 and 4 are only necessary if host write-protection mode is used in normal operation.

CONTACTING CIRRUS LOGIC SUPPORT

For a complete listing of direct Distributor, Sales, and Sales Representative contacts, visit the Cirrus Logic website at <http://www.cirrus.com>.

Erratum 2 - On Chip Reference Reset State

Description

The on chip voltage reference, $V_{REF\pm}$, may occasionally assume alternate states and apply alternate voltage reference values during device power up. Accuracy performance outside of specification may result when calibration is performed at one state and the device powers up in an alternate state.

Workaround

A register write sequence is used to disable the circuit that creates the alternate states. The following write sequence should be written to the device registers at power up or after any reset event:

1. Write register Page 0, Address 28 with a value of 0x000016
2. Write register Page 0, Address 30 with a value of 0x0C0008
3. Write register Page 0, Address 28 with a value of 0x000000

Note: This write sequence must be followed in the exact order. Periodic reading of the register at Page 0, Address 30 should show a value of 0x0C0008, which confirms proper operation.

Erratum 3 - Serial Peripheral Interface Communication Synchronization Error

Description

This erratum applies only to designs with chip select pin \overline{CS} tied low. If Serial Peripheral Interface (SPI) communication is interrupted, a synchronization error may occur on future SPI communications. A standard reset does not clear the SPI logic and correct this synchronization error.

Workaround

Chip select pin \overline{CS} must be toggled to reset the SPI Interface. A NPN inverter may be added to the \overline{RESET} signal to obtain a suitable \overline{CS} signal to reset the SPI Interface without the need for additional isolators. When the device is reset, using \overline{RESET} , the serial peripheral interface is also reset.

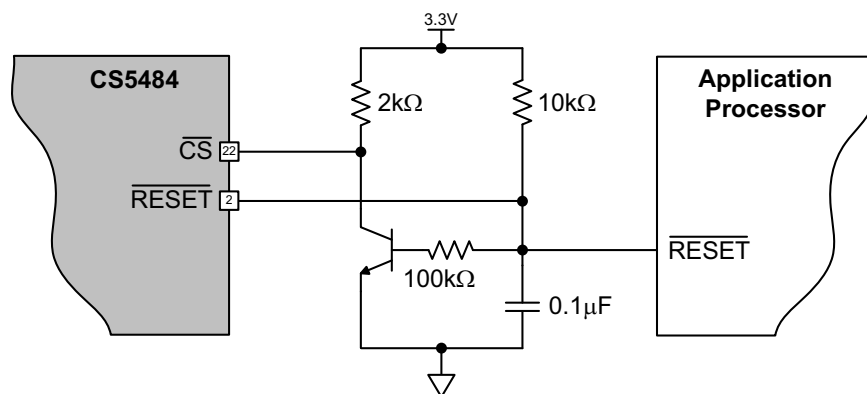


Figure 1. Chip Select Pin \overline{CS} Model

Erratum 4 - Multiple Checksum with Line-cycle Synchronous Averaging Mode

Description

When line-cycle synchronous averaging mode is enabled, the on chip register checksum calculation will produce two checksum values based on reserved bit 16 in register *CONFIG2*: checksum value one, when reserved bit 16 is high; checksum value two, when reserved bit 16 is low.

Workaround

After the CS5484 has been fully configured and loaded with valid calibration values, the application processor should keep a copy of the current checksum masking the reserved bit 16.

After configuration, store the following:

```
ChecksumStored = ChecksumRead & checksumMask;
```

Use the following to check for checksum errors:

```
If ((ChecksumRead & checksumMask) != ChecksumStored) {  
    ReloadConfiguration( );  
}
```

where

checksumMask =

0xFFFF (or not required) with fixed number of samples averaging mode enabled.

0xFEFFFF with line-cycle synchronous average mode enabled.

ChecksumRead is read from the checksum register after full configuration.