



Errata: CS5480 Rev B2 Silicon

(Reference CS5480 data sheet revision DS980F3 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 CS5480 is operating in Line-cycle Synchronized Averaging mode, the *SampleCount* register will not be updated by the CS5480 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.

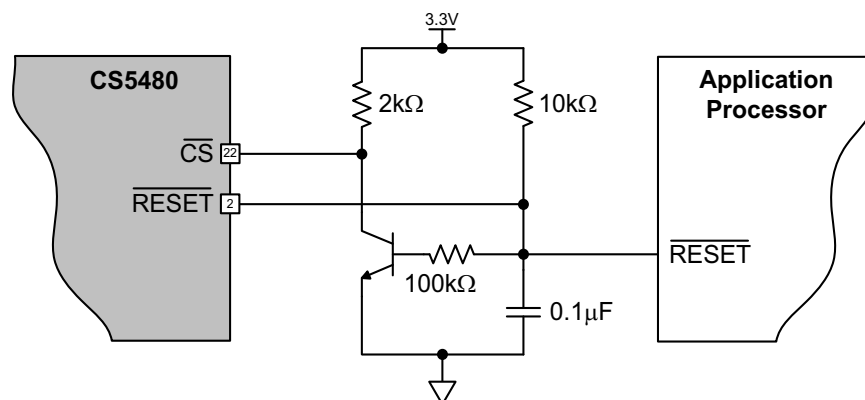


Figure 1. Chip Select Pin \overline{CS} Model

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

Description

When automatic channel selection is enabled, the on-chip register checksum calculation will produce two checksum values based on bit ICHAN in register *CONFIG2*: checksum value one, when the highest load is on channel 1 (ICHAN = 0); checksum value two, when the highest load is on channel 2 (ICHAN = 1).

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 CS5480 has been fully configured and loaded with valid calibration values, the application processor should keep a copy of the current masked checksum.

After configuration, store the following:

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

Use the following to check for checksum errors:

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

where

checksumMask =

0xFFFFFFFF (or not required) with fixed number of samples averaging mode enabled and automatic channel selection disabled.

0xDFFFFFFF with fixed number of samples average mode and automatic channel selection enabled.

0xDEFFFFFF with line-cycle synchronous average mode and automatic channel selection enabled

0xFEFFFFFF with line-cycle synchronous average mode enabled and automatic channel selection disabled.

ChecksumRead is read from the checksum register after full configuration.