Document Number: MC33816ER
Rev. 1.0, 8/2016

MC33816 chip errata

This errata document applies to the MC33816 SMARTMOS product family.

Table 1. Orderable part number identification

Part number	Die ID	Package	Chip marking
MC33816AE MC33816AER2	N27H	98ASA00237D 64-PIN LQFP EXPOSED PAD	(F) MC33816AE AWLYYWWZ

1 Device part number prefixes

Device samples marked with a P pre-fix indicate prototype. These devices have undergone basic testing only and are not considered qualified. Any other device pre-fix indicates the product is in production, has full characterization, qualification and testing has been preformed, unless otherwise noted.

2 Device build information / date code

The marked trace code is the link between the physically marked materials and the manufacturing lot's system genealogy information. Once the connection between the marked material and system genealogy information is made, traceability reports provide the material's manufacturing/shipping history. All devices listed in the Errata are affected unless specific date codes are provided below.

3 Description

The following table provides the general definitions of the errata severity in this document.

Table 2. Definitions of errata severity

Errata severity level	Meaning
High	Failure mode that severely inhibits the use of the device for all or a majority of intended applications
Medium	Failure mode that might restrict or limit the use of the device for all or a majority of intended applications
Low	Unexpected behavior that does not cause significant problems for the intended applications of the device
Enhancement	Improvement made to the device due to previously found issues on the design

4 Component specification errata - analog functions

ER1: Glitch on G_HSx while VBS is ramping up and B_HSx operating voltage limitation

Severity level – low

Description of problem

The MC33816 requires that no HS pre-driver or LS pre-driver be switched on while the device is in the process of starting up or shutting down. The device requires specific operating voltages for S_HSx and B_HSx pre-drivers.
An analog design flaw in the HS pre-driver control circuitry causes several issues:

- The PMOS transistor in the HS pre-driver switches on briefly during device startup. This happens as the HS pre-driver's VBS supply voltage is ramping up to about 1.1 V to 1.5 V. The glitch does not occur at G_HSx during ramp down of the VBS voltage.
- The HS pre-driver can not switch on as long as B_HSx voltage is below 2.0 V with respect to PGND (refer to **Table 3**).
- The HS pre-driver G_HSx switches on briefly when recovering from a -8.0 V glitch at S_HSx.

Table 3. B_HSx operating voltage limitation

Parameter	Symbol	Min.	Typ.	Max.	Unit	Comment
B_HSx operating voltage	V_{B_HSx}	V_{S_HSx}		V_{S_HSx}		V_{S_HSx} has to be 2.0 V above PGND for full functioning (switch on) of the pre driver
		+4.0		+8.0	V	
		2.0		80	V	

Measurement and simulation results

Figure 1 shows measurement results for the glitch at G_HSx during VBS ramp up. The duration of the glitch in these measurements is about 1.8 ms. The maximum voltage at G_HSx is about 1.5 V.
Figure 2 shows simulation results for the same issue with similar results.

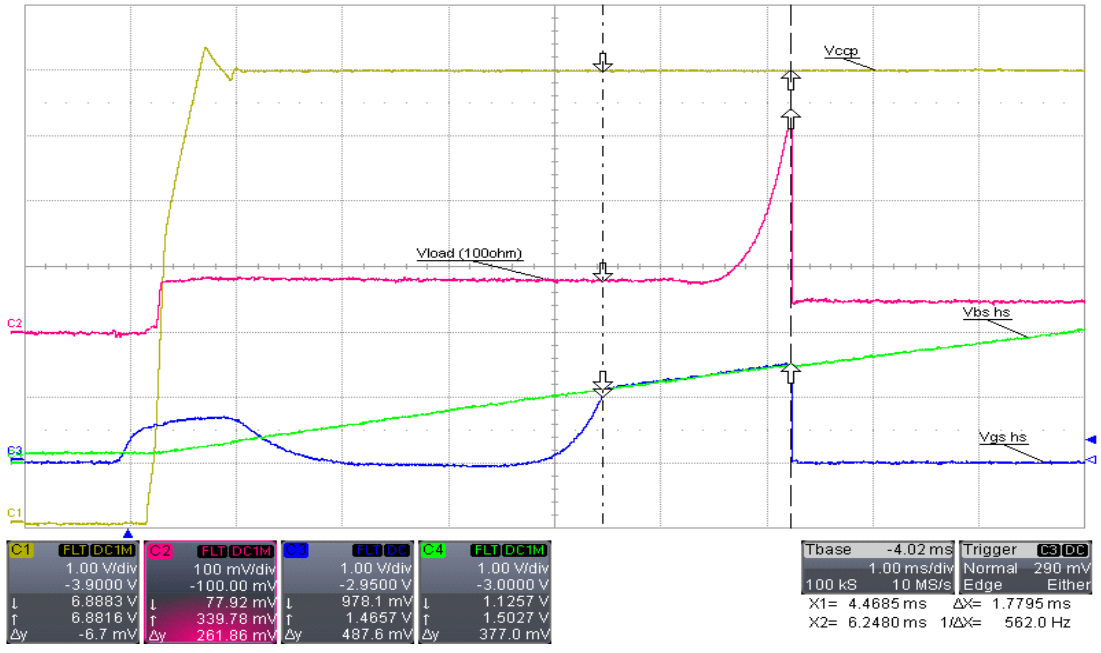

Figure 1. Measurement of glitch on G_HSx during V_{BS} Ramp-up at 25 °C

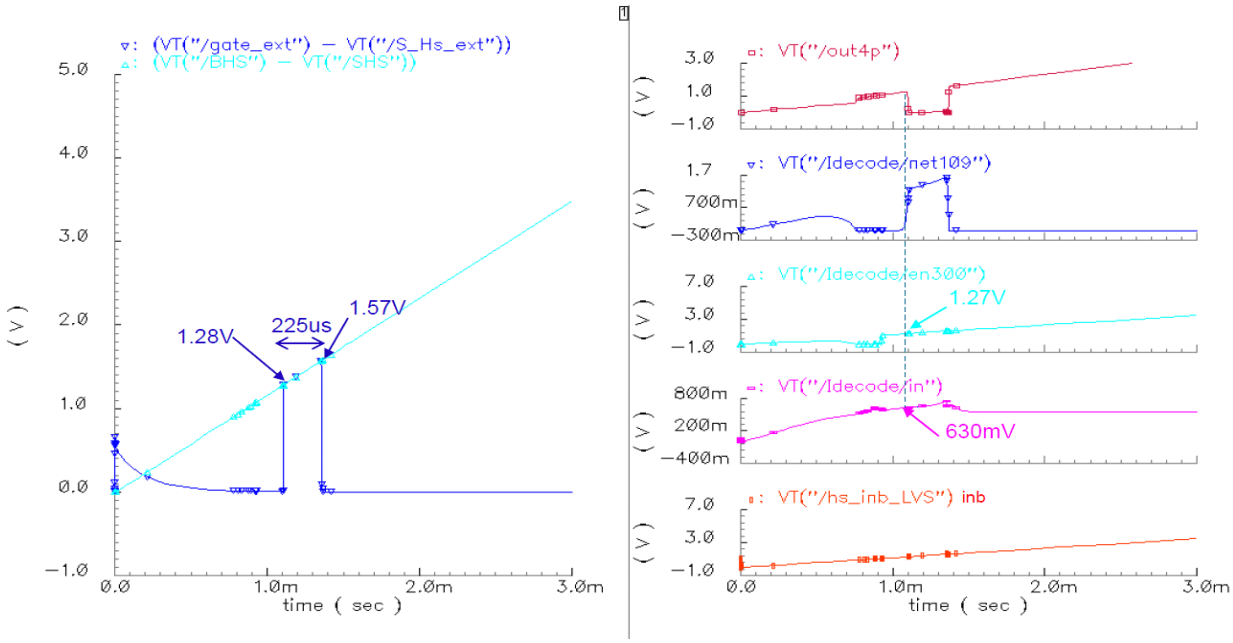

Figure 2. Simulation of glitch on G_HSx during V_{BS} ramp-up

Figure 3 shows an active G_HSx signal after recovery from a -8.0 V transient at S_HSx. The G_HSx is switched on for about 2.5 µs. Figure 4 shows that the glitch is no longer present when the negative transient is limited to -7.0 V.

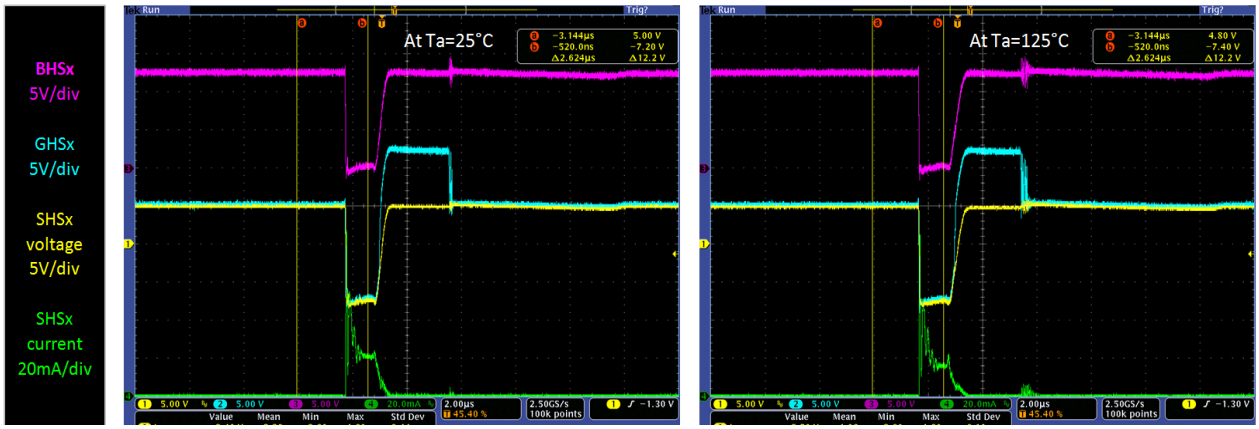

Figure 3. Measurement of glitch on G_HSx when -8.0 V transient is removed

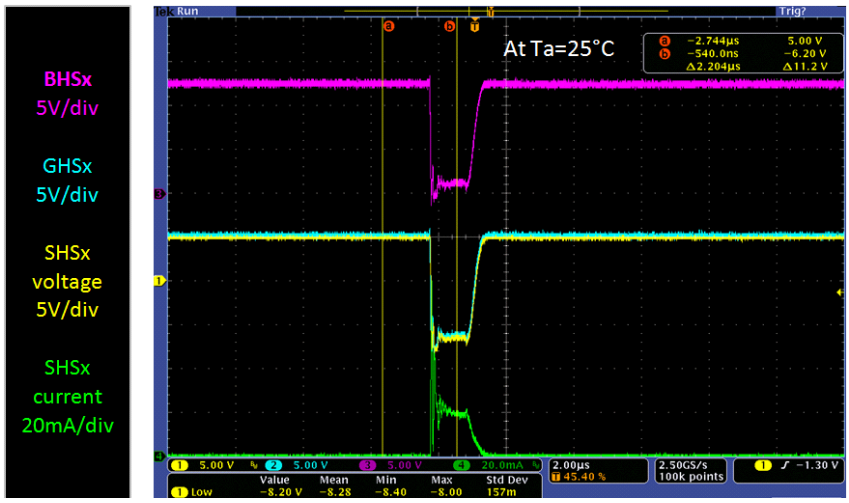

Figure 4. No glitch on G_HSx when -7.0 V transient is removed

Customer impact or symptoms

Unexpected behavior of the HS pre-driver or LS pre-driver while the MC33816 is in the process of starting up or shutting down.

Workaround

No workaround available, needs to be taken in account during ECU design

Fix plan

No fix scheduled

5 Component specification errata — digital functions

ER2: SPI error detection when command word is less than 16-bit

Severity level – low

Description of problem

When an SPI command word is transferred using mode A, the SPI slave select is set to inactive immediately after the 16 bits of the command word are transferred. A command word of less than 16 bits might be detected at the same time that slave select is set to inactive. Because of an implementation bug, the MC33816 does not detect the SPI word error until the transmission of the next word (which would be a data word in this case.)

Customer impact or symptoms

There is a delay in the detection of SPI word errors.

Workaround

There is no workaround. The delay in the detection of the SPI error is not critical.

Fix plan

No fix scheduled

ER3: Auto iret for loss of clock ISR

Severity level – low

Description of problem

The driver disabled interrupt routine (ISR) can be terminated (if configured in this way by the iconf instruction) by reading the related diagnosis register through SPI (not through the SPI back door) or at the moment when the drivers are enabled again.

Customer impact or symptoms

Based on the fact that the loss of clock interrupt uses the same ISR as the driver disabled interrupt, any automatic iret (if enabled) which would end the driver disabled ISR will also end the loss of clock ISR.
Loss of clock ISR when automatic iret is enabled.

Workaround

Only use the iconf instruction to enable automatic iret when you are sure that the ISR was called due to a driver disabled interrupt.

Fix plan

No fix scheduled

ER4: cksys out of spec

Severity level – low

Description of problem

Simulations of the MC33816 cksys frequency indicate that, under certain circumstances, cksys exceeds 24 MHz while it is still propagated to the digital core.
The specified maximum clock frequency for the digital core and memory is 25.725 MHz (nominal 24 MHz). Clock frequencies beyond 24 MHz may cause the digital core to malfunction.

Failure event in CLK signal

Test description

A single event like a phase shift, a pulse that is too short or a missing pulse is created in the CLK signal.
The trace unit is set to a mode that continuously outputs trace info on the Dbg pin. The trace data is made of a 0101 sequence. this configuration allows you to measure the device's cksys clock via the bit rate of the Dbg output.
In summary, the set up is:

- MC33816 device mounted in a test board socket
- External CLK set to 1.00 MHz (with one single phase shift event)
- PLL set to 24 MHz
- PLL spread on or off

Results

All measurements indicate that the PLL reacts very quickly to a disturbance in the CLK clock reference. In the process it may occasionally generate a cksys output frequency beyond the 24 MHz limit. The clock monitor doesn't always switch the PLL input to the backup clock when the cksys frequency overshoot occurs. But in some cases it does, which can result in the generation of additional cksys frequencies beyond the 24 MHz limit.

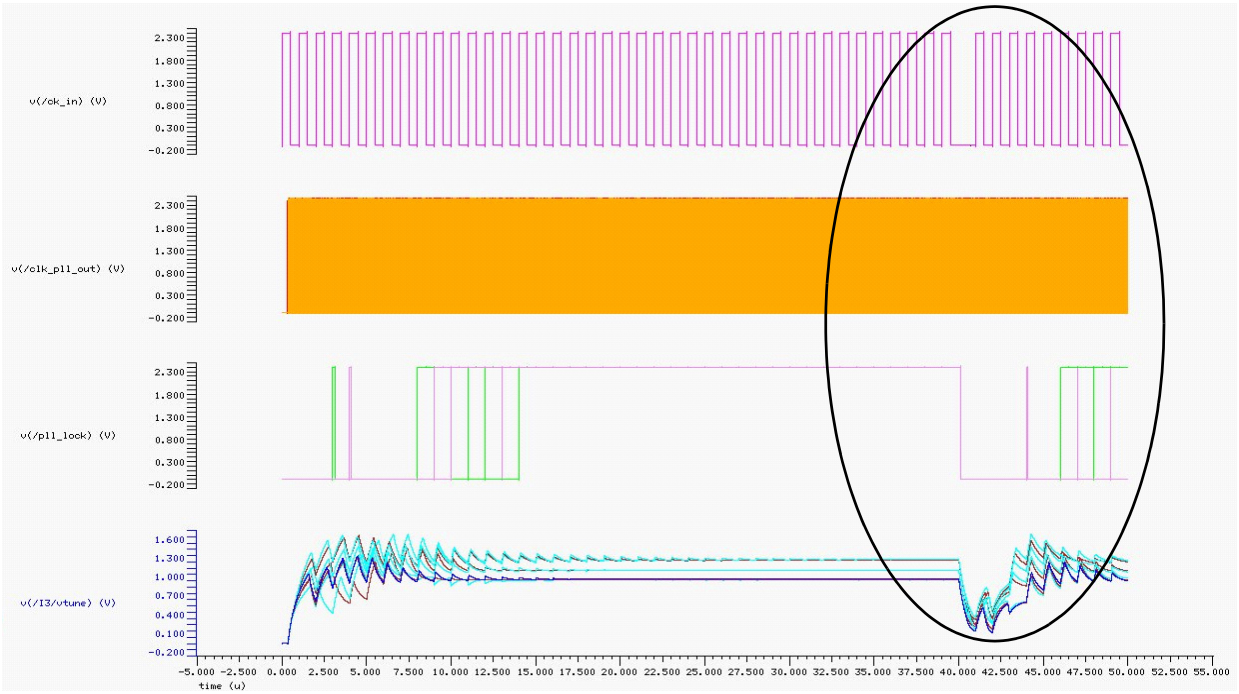

Figure 5. Simulation CLK on edge missing

Figure 5 shows the effect of one missing clock edge.

Customer impact or symptoms

Table 4 shows the situation where a critical cksys frequency can occur.

Table 4. List of cksys frequency issues

Situation	Fault(s)
Disturbance on CLK reference (when device is running on CLK reference)	(1) cksys >30 MHz for some clock pulses
	(2) PLL is unlocked

Workaround

To avoid a malfunction of the digital core and the power stage when ext. CLK fails, do the following:

- Configure ASIC to assert IrqB when a loss of clock is detected
- Set IrqB to be detected by µC
- Read ASIC status via SPI (e.g. backup_clock_status register)
- Trigger SW reset of ECU if a loss of clock is detected

Simulations have demonstrated that, with this configuration, the IrqB pin function and the SPI interface and registers are reliable up to a PLL output frequency of 83.3 MHz (41.65 MHz on cksys.)

Fix plan

No fix scheduled

ER5: SPI mode B: one word more written

Severity level – low

Description of problem

SPI mode B allows you to specify the number of data words transmitted in the control word. While specifying the number of data words is not mandatory, doing so improves SPI error detection.

When the specified number parameter is greater than zero, the SPI interface goes into an error state (frame error) if:

- the chip select is de-asserted and the number of words transferred is lower than the number parameter required by the command word
- the number of words transferred is equal to the number parameter + 1

A design flaw causes the MC33816 to incorrectly process a mode B write access when the number parameter is used. When the number parameter is set to *n* and the SPI master transmits more than *n* words, the MC33816 erroneously writes one additional data word (*n* + 1) to memory.

Customer impact or symptoms

The MC33816 exhibits the following behavior during a mode B SPI write access with the data word number parameter set:

- the parameter number is set to 3 (for example) and the SPI master transmits three data words
  - No error is issued
- the parameter number is set to 3 (for example) and the SPI master transmits four data words
  - An SPI frame error is issued after the 4th data word but 4th data word is written to RAM or register.
- the parameter number is set to 3 (for example) and the SPI master transmits six data words
  - An SPI frame error is issued after 4th data word but 4th data word is written to RAM or register. The 5th and 6th data words are not written to memory.

Workaround

There is no software workaround. The SPI master must not transmit additional data words during an SPI mode B write access.

Fix plan

No fix scheduled

ER6: Digital flag/start anti-glitch filter

Severity level – low

Description of problem

All the digital input pins (flag/start) have an anti-glitch filter whose time constant is 3 clock cycles (125 ns@24 MHz).
The anti-glitch filters are preceded by a sampling stage using the ck clock, whose period can be longer than filter time constant. When this occurs, the filters are useless.

Silicon implementation

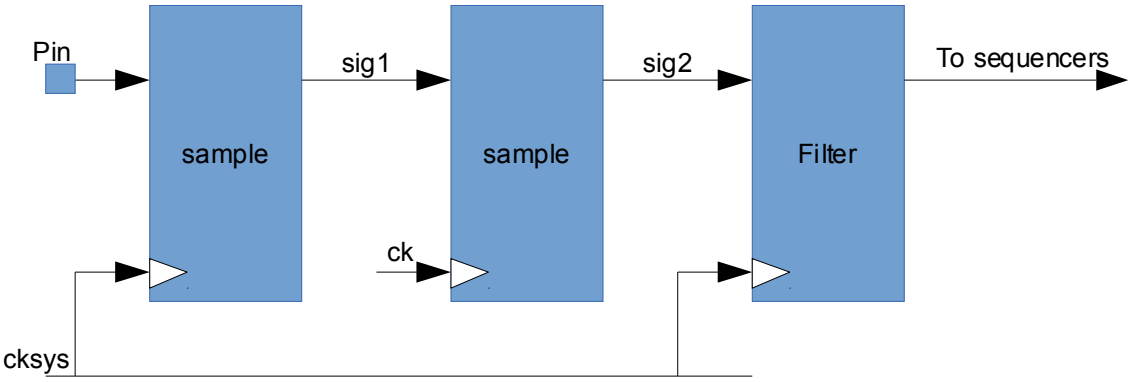

Figure 6. Anti-glitch filters current implementation

In the current implementation (Figure 6):

- Sig1 maximum bandwidth is the cksys frequency (24 MHz). Because this is a sampling stage, a glitch can bypass the first stage, as long as the pin value changes during the cksys clock edge.
- Sig2 maximum bandwidth is the ck frequency (12 MHz or lower). Because this is also a sampling stage, a glitch can bypass the second stage, as long as the pin value changes during ck clock edge. Because ck is slower than cksys, a glitch that bypasses the first stage could be caught in the second sampling stage. However, the glitch might also bypass both stages.
- The filter cutoff frequency is 8 MHz, while its input has a bandwidth of 12 MHz or lower. If the "ck frequency"/bandwidth is 6.0 MHz (very common in applications), the filter is useless (filtering an 6.0 MHz signal with an 8.0 MHz period). Under these circumstances, any filter that bypass the second sampling stage will also bypass the filters.

If ck frequency is lower than 8.0 MHz, no anti-glitch filter function is available on flag/start pins.

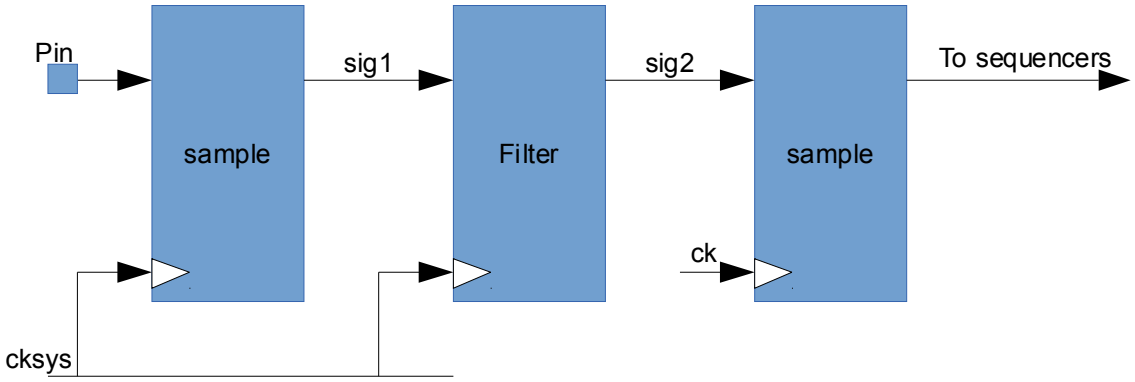

Figure 7. Anti-glitch filters correct implementation

In a correct implementation, the filter would have been placed before the ck sampling stage so that its input would always have a 24 MHz bandwidth.

Customer impact or symptoms

The anti-glitch filter on the flag/start pins does not function when ck frequency is lower than 8.0 MHz.

Workaround

No workaround is possible: if ck frequency is lower than 8.0 MHz, no anti-glitch filter function is available on flag/start pins.

Fix plan

No fix scheduled

ER7: Load [DRAM] spi_data instruction causes corruption to spi_data register

Severity level – low

Description of problem

Each microcore has a private "spi_data" register in its internal memory map. The microcore can write to its spi_data register using the load/cp instruction. When triggered by the rdspi instruction, the SPI back door module can also write to this register. When the microcore uses the load instruction to write to the spi_data register (that is, when it copies a value from the Data RAM), the data written into the target register may be corrupted if the external SPI interface is accessing the DRAM at the same time.
Testing confirms that:

- The issue does not affect the other means used to write to the spi_data register, such as the cp and wrspi instructions.
- The issue does not affect any of the other entries in the microcore's internal memory map.
- DRAM accesses by other microcores in the same channel will not corrupt the spi_data register value.

Customer impact or symptoms

Data corruption occurs in a microcore's internal spi_data register when the microcore loads the register with data from DRAM.

Workaround

The workaround consist in splitting the instruction

```
load xxx spi_data ofs/_ofs;
```

into two instructions

```
load xxx tempReg ofs/_ofs;
cp tempReg spi_data;
```

where *tempReg* is any register of the internal memory map whose value can be overwritten (for example, the register ir)

Fix plan

No fix scheduled

ER8: HS5_LS36 override interaction with diagnosis_option

Severity level – low

Description of problem

The MC33816 exhibits an anomaly related to the interaction of the Diagnosis_option register with the Driver_config register.
When the Diagnosis_option register's Diag_option bit (bit 0) and the Driver_config register's hs5_ls36_ovr bit (bit 15) are both set, the specification requires the following:

- HS5, LS3 and LS6 will be driven even if the DrvEn pin is low.
- Diagnosis will be disabled if the pre-drivers are disabled. Since HS5, LS3 and LS6 are not disabled, their automatic diagnosis should still be active.

As implemented in silicon, when DrvEn is low, diagnosis is disabled on all outputs regardless of the override bit setting. When the Diag_option and the hs5_ls36_ovr bits are set and DrvEn is low, HS5, LS3 and LS6 can be driven and are not protected by automatic diagnosis.
The Table 5 summarize the status of Hs5, Ls3 and Ls6 outputs in all the above conditions.

Table 5. Diag_option, hs5_ls36_ovr and DrvEn results

Diag_option bit	hs5_ls36_ovr bit	DrvEn	Hs5 Ls3 Ls6	
			PreDriver	Diagnosis
0	0	High	Enabled	Enabled
0	1	High	Enabled	Enabled
1	0	High	Enabled	Enabled
1	1	High	Enabled	Enabled
0	0	Low	Disabled	Disabled
0	1	Low	Enabled	Enabled
1	0	Low	Disabled	Disabled
1	1	Low	Enabled	Disabled

The issue does not affect the other outputs.

Customer impact or symptoms

On Hs5, LS3, and LS6, diagnosis is disabled when the Diag_option bit (Diagnosis_option register) and the hs5_ls36_over bit (Driver_config register) are both set and DrvEn is low.

Workaround

There are two workarounds:

- Do not allow the Diag_option bit (Diagnosis_Option register) and the hs5_ls36_ovr bit (Driver_config register) bit to be set at the same time.
- Implement the protection in microcode, by continuously checking the value of the Hs5/Ls3/Ls6 Vds feedbacks

Fix plan

No fix scheduled

6 Revision history

Revision	Date	Description of changes
1.0	9/2015	• Initial release (PB # 16936)
	10/2015	• Corrected errors in Table 1
	8/2016	• Updated to NXP document form and style

How to Reach Us:

Home Page:
NXP.com

Web Support:
http://www.nxp.com/support

Document Number: MC33816ER
Rev. 1.0
8/2016