Freescale Semiconductor
Chip Errata

MCF5282DE
Rev 8, 02/2015

MCF5282 Chip Errata

Silicon Revision: All

This document identifies implementation differences between the MCF5281/82 processors and the description contained in the *MCF5282 ColdFire® Reference Manual*. Refer to http://www.freescale.com/coldfire for the latest updates.

All current MCF5281/82 devices are marked as L95M mask set. The date code on the marking can be used to determine which errata have been corrected on a particular device as shown in Table 1. The datecode format is XXXYYWW, where YY represents the year and WW represents the work week. The three leading digits can be ignored.

Table 1. Summary of MCF5281/82 Errata

Errata	Module Affected	Date Errata Added	Date Code Affected?		
			<XXX0324	XXX0324 to XXX0326	>XXX0326
SECF035	PLL	3/18/03	Yes	No	No
SECF003	BDM	3/28/03	Yes	Yes	Yes
SECF002	EMAC	3/28/03	Yes	Yes	No
SECF021	Cache	3/31/03	Yes	Yes	No
SECF004	Flash	4/09/03	Yes	Yes	No
SECF005	Cache	7/21/03	Yes	Yes	Yes
SECF001	Cache	7/21/03	Yes	Yes	Yes
SECF029	FlexCAN	7/23/03	Yes	Yes	Yes
SECF009	FEC	4/22/04	Yes	Yes	Yes
SECF007	FEC	4/22/04	Yes	Yes	Yes
SECF036	PLL	8/23/04	Yes	Yes	Yes
SECF010	FEC	9/14/04	Yes	Yes	Yes

Table continues on the next page...

Table 1. Summary of MCF5281/82 Errata (continued)

Errata	Module Affected	Date Errata Added	Date Code Affected?		
			<XXX0324	XXX0324 to XXX0326	>XXX0326
SECF038	QADC	3/15/05	Yes	Yes	Yes
SECF031	GPIO	1/6/06	Yes	Yes	Yes
SECF037	PLL	6/12/06	Yes	Yes	Yes
SECF006	FEC	5/31/07	Yes	Yes	Yes
SECF123	FlexCAN	8/28/08	Yes	Yes	Yes
SECF124	Cache	2/17/09	Yes	Yes	Yes
SECF125	FlexCAN	4/25/09	Yes	Yes	Yes
SECF036A	PLL	6/9/10	Yes	Yes	Yes
SECF015	Flash	11/18/2014	Yes	Yes	Yes

The table below provides a revision history for this document.

Table 2. Document Revision History

Rev. No.	Date	Substantive Changes
3	5/2007	Added errata: SECF037 “Phase Relationship Between CLKOUT and CLKIN Not Preserved When Using PLL” and SECF006 “FEC Duplicate Transmission Bug”
4	8/2008	Added errata: SECF123 “FlexCAN Writing to an Active Receive MB May Corrupt MB Contents"
5	2/2009	Added status headings for each errata. Added errata: SECF124 “Buffered Write May Be Executed Twice”
6	4/2009	Added errata: SECF125 “Any FlexCAN MB access during RX or TX of an extended ID""frame’s CRC and EOF may cause unwanted message reception”
7	6/2010	Added errata: SECF036A “PLL Does Not Lock in Normal PLL Mode with Crystal Reference"
8	02/2015	Added errata: SECF015 “Internal Flash Address Qualification Incomplete"

SECF035: Leakage Current on V_{DDPLL} pin

Errata type: Silicon
Affects: PLL
Description: The device exhibits a 65mA leakage current on the V_{DDPLL} supply, regardless of chip configuration.

Workaround: No workaround.

Fix plan: Fixed in datecodes XX0324 and later.

SECF003: BDM Load of SR Does Not Enable Stack Pointer Exchange

Errata type: Silicon
Affects: BDM
Description: The V2 core used in this device adds support for separate user and supervisor stack pointers. The hardware implements an active stack pointer and an other_stack_pointer. Whenever the operating mode of the processor changes (supervisor to user or user to supervisor), the processor hardware exchanges the active SP and the other SP.

This exchange operation does not work when the processor mode is changed by a write to the SR from the BDM port. The hardware in the processor core required to process the BDM load_SR operation and enable the stack pointer exchange is missing.

The exchange works properly when the SR is changed through software.

Workaround: Use software for any operations that require exchanging the stack pointers.

Fix plan: Currently, there are no plans to fix this.

SECF002: Unexpected Pipeline Stall on EMAC Load/Store Accumulator Instruction

Errata type: Silicon
Affects: EMAC
Description: An unexpected pipeline stall occurs for accumulator load and accumulator store instructions that immediately follow a load accumulator or MAC instruction.

Specifically, the operand execution pipeline (OEP) experiences a 2T pipeline stall when a load/store accumulator instruction enters the pipeline immediately after any load accumulator or MAC instruction. The pipeline is supposed to stall only if there is a store accumulator instruction immediately following a load or MAC instruction that updated the specified accumulator.

A simple example can be created to expose this problem:

```
mac.l ra,rb,acc0
mac.l rc,rd,acc0
mov.l acc1,rx
```

In the above example, the store of

```
acc1 (mov.l acc1,rx)
```

should not experience any stall because that accumulator is not being updated. In the current V2 + EMAC implementation, it incorrectly stalls for two cycles.

NOTE

The operation of the instructions is correct. The problem is that the expected timing is not met.

Workaround: No workaround.

Fix plan: Fixed in datecodes XXX0327 and later.

SECF021: Incorrect Cache Size

Errata type: Silicon
Affects: Cache
Description: The device operates as if it were connected to an 8KB cache; however, the cache size is 2KB. After the 2KB cache is full, the cache controller can have erroneous hits in the cache space resulting in data and/or instruction corruption.

Workaround: Do not enable the cache.

Fix plan: Fixed in datecodes XXX0327 and later.

SECF004: Corrupted Fetches from Flash

Errata type: Silicon
Affects: Flash controller
Description: Leaving bit 6 in the FLASHBAR register cleared can cause corrupted fetches from the device's internal flash. For datecodes after XXX0327, the bit is hardwired high to prevent the corrupted accesses.

Workaround: Set bit 6 in the FLASHBAR. This prevents the corrupted fetches.

Fix plan: Fixed in datecodes XXX0327 and later.

SECF005: Possible Cache Corruption After Clearing Cache (Setting CACR[CINV])

Errata type: Silicon
Affects: Version 2 ColdFire Cache
Description: The cache on the V2 ColdFire core may function as either:

- a unified data and instruction cache
- an instruction cache
- a data cache

The cache function and organization is controlled by the cache control register (CACR). The CACR[CINV] bit causes a cache clear. If the cache is configured as a unified cache and the CINV bit is set, the scope of the cache clear is controlled by two other bits in the CACR:

- CACR[INVI] invalidates instruction cache only
- CACR[INVD] invalidates data cache only

If a write to the CACR is performed to clear the cache (CACR[CINV] = 1) and only a partial clear is done (CACR[INVI] or CACR[INVD] set), then cache corruption may occur.

Workaround: All loads of the CACR that perform a cache clear operation (CACR[CINV] set) should be followed immediately by a NOP instruction. This avoids the cache corruption problem.

Fix plan: Currently, there are no plans to fix this.

SECF001: Incorrect Operation of Cache Freeze (CACR[CFRZ])

Errata type: Silicon
Affects: Version 2 ColdFire Cache
Description: The cache on the V2 ColdFire core is controlled by the cache control register (CACR). When the CACR[CFRZ] bit is set, the cache freeze function is enabled and no valid cache array entry is displaced. However, this feature does not always work as specified, sometimes allowing valid lines to be displaced when CACR[CFRZ] is enabled.

This does not cause any corrupted accesses. However, there could be cache misses for data that was originally loaded into the cache but was subsequently deallocated, even though the CACR[CFRZ] bit was set.

Also, incoherent cache states are possible when a frozen cache is cleared via the CACR[CINV] bit.

Workaround: Unfreeze the cache by clearing CACR[CFRZ] when invalidating the cache using the CACR[CINV] bit

Workaround: Use the internal SRAM to store critical code/data if the system cannot handle a potential cache miss

Fix plan: Currently, there are no plans to fix this.

SECF029: Incorrect 32-bit Accesses to FlexCAN Registers

Errata type: Silicon
Affects: FlexCAN
Description: Because the FlexCAN was originally designed for 16-bit architectures, all 32-bit register accesses are broken down into two back-to-back 16-bit accesses. However, the timing for the back-to-back accesses is incorrect and leads to corruption of the second 16-bit read or write.

Workaround: When reading or writing to the 32-bit RxMASK registers, use two 16-bit accesses instead of a single 32-bit access.

Fix plan: Currently, there are no plans to fix this.

SECF009: FEC Receive Buffer Overrun in 10BaseT Mode

Errata type: Silicon
Affects: FEC
Description: When the FEC is connected to a 10BaseT network, if length of the data stored in a descriptor is not evenly divisible by 16 (not line-aligned), the FEC writes extra lines at the end of the buffer. The entire line that contains the last valid data is written and at least one extra line, but

up to four lines after the end of the valid data can also be written. In most cases, this is not a problem because the extra lines of data continue falling within the limits of the buffer. However, if the valid data ends near the end of the buffer, the extra lines written by the FEC might be outside of the data buffer. This leads to corruption of the next buffer, descriptor, data, or code stored in the adjacent memory.

For example, as shown in the figure below, if the max buffer size is programmed to 0x600 and a frame that is 0x5F8 bytes long is received, a line is written starting at buffer start + 0x5F0. The first half of the line at buffer start + 0x5F0 is valid frame data that should be processed by the FEC driver; the second half of the line is additional data that is written because the FEC only writes complete lines. This data should be ignored by the FEC driver. So far, this is correct FEC behavior as originally specified. However, the FEC repeats the last line of valid data a number of times. The line at buffer start + 0x600 is written, and as many as three additional lines beyond the end of the data buffer could be written.

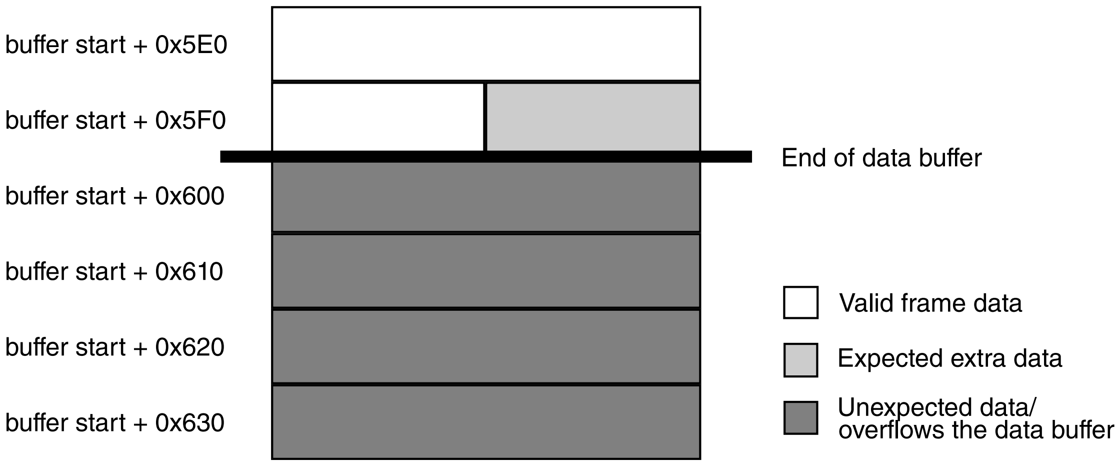

Figure 1. Buffer Overrun Example

Workaround: Only use 100BaseT.

Workaround: Allocate extra lines for the receive data buffers. The actual allocated memory for each buffer should be equal to the receive buffer size programmed in the FEC's EMRBR register plus four lines (16 byte-sized lines).

Workaround: Program the data buffer size one line larger than the max packet size (data buffer size = EMRBR + 0x40).

Fix plan: Currently, there are no plans to fix this.

SECF007: Concatenation of Received Frames in 10BaseT Mode

Errata type: Silicon
Affects: FEC
Description: When the FEC is connected to a 10BaseT network, sometimes the FEC combines the data from multiple frames to generate a single frame. The data from the frames is received correctly, but the frame boundary is not reported correctly. This causes the descriptor to report the length as the data length for all of the concantenated frames added together. The incorrect data length might exceed the max frame length programmed in the RCR[MAX_FL] field.

When TCP is used as a transport mechanism, this errata manifests itself as lost packets and reduced throughput. Data continues to be received correctly because TCP requests retransmission of bad packets. However, UDP does not include any mechanism for packet retransmission, as it is a send and forget protocol. Consequently, while UDP should be able to identify an incorrectly received packet (because its checksum will fail), higher level software in the protocol stack must be capable of requesting retransmission to work around this errata.

Workaround: Higher level Ethernet layer code should compare the length reported by the descriptor to the length included in its header. If the lengths do not match, the packet should be truncated or discarded as needed. The protocol stack must be responsible for requesting retransmission of any frames that are discarded due to the data length mismatch.

Fix plan: Currently, there are no plans to fix this.

SECF036: PLL Does Not Lock in Normal PLL Mode with External Clock Reference

Errata type: Silicon
Affects: PLL
Description: During a power on reset, if the CLKMOD[1:0] equals 10 (normal PLL mode with external clock reference), the PLL does not lock and the device never comes out of reset.

Workaround: When configuring the PLL for normal PLL mode with external clock reference, tie CLKMOD1 to $\overline{\text{RSTI}}$ and not straight to 3.3V. This allows the PLL to correctly detect the desired operating mode and lock.

Fix plan: Currently, there are no plans to fix this.

SECF010: FEC Interrupts will not Trigger on Consecutive Transmit Frames

Errata type: Silicon
Affects: FEC
Description: The late collision (LC), retry limit (RL), and underrun (UN) interrupts do not trigger on consecutive transmit frames. For example, if back-to-back frames cause a transmit underrun, only the first frame generates an underrun interrupt. No other underrun interrupts are generated until a frame is transmitted that does not underrun or the FEC is reset.

Workaround: Because late collision, retry limit, and underrun errors are not directly correlated to a specific transmit frame, in most cases a workaround for this problem is not needed. If a workaround is required, there are two independent workarounds:

- Ensure that a correct frame is transmitted after a late collision, retry limit, or underrun errors are detected.
- Perform a soft reset of the FEC by setting ECR[RESET] when a late collision, retry limit, or underrun errors are detected.

Fix plan: Currently, there are no plans to fix this.

SECF038: Possible QADC Command Conversion Word (CCW) Table Corruption

Errata type: Silicon
Affects: QADC

Description: A CCW table location may be corrupted by writing any other CCW or results table location while any queue is active. If a CCW table or result table write occurs while either queue is active, it is possible for another CCW location to be corrupted. This bug only occurs if the write cycle is simultaneous with the queue state machine reading the next CCW location. The odds of this happening are one in the number of clocks in a conversion.

Workaround: Make sure that both queues have completed or are paused before updating a CCW table or result table location.

Workaround: If workaround one is not possible, the application code can monitor the CWP bits in the QASR0. After it changes, it is safe to write a CCW table or result register location. The safe time is equal to the input sample time of the next conversion (4-18 QCLKs).

Workaround: If workarounds one and two are not possible, it is possible to update a CCW or result register location while a queue is active by using the following sequence:

- Read status register 0 and save the CWP value
- Perform the write
- Read CCW locations pointed to by CWP and CWP+1 to check if they are corrupted
- Fix any of the possibly corrupted locations

The above sequence should be safe. If a CCW location is corrupted, it is not used until a queue wraps around back to this CCW. The user has one conversion time to perform the corruption checks and fixes. There should be plenty of time to do this without worrying about another CCW corruption.

Fix plan: Currently, there are no plans to fix this.

SECF031: GPIO Inputs Behave Inappropriately with 10k Ohm Pull Downs

Errata type: Silicon
Affects: Ports
Description: GPIO inputs that shouldn't have internal pull-ups, behave as if internal pull-ups are enabled when pull-down resistors larger than 10k ohm are used.

To achieve 5V tolerance for the I/O pads, a pull-up device is used to latch the input value of the pads while protecting internal circuitry to direct exposure to potentials above 3.6V. These pull-up devices are not disabled after stimulus is removed and a pull-down resistor value larger than 10k ohm is used.

Workaround: To disable the pull-up, a pull-down resistor value of 10k Ohm or less is needed.

Fix plan: Currently, there are no plans to fix this.

SECF037: Phase Relationship Between CLKOUT and CLKIN Not Preserved When Using PLL

Errata type: Silicon
Affects: PLL
Description: In all PLL modes, the CLKOUT phase relationship to the input clock drifts by 25-33%.

Workaround: Use bypass mode, by setting CLKMOD[1:0] to 00. The CLKOUT to CLKIN phase relationship is maintained.

Fix plan: Currently, there are no plans to fix this.

SECF006: FEC Duplicate Transmission

Errata type: Silicon
Affects: FEC
Description: In some cases, the FEC transmits single frames more than once. The FEC fetches transmit buffer descriptors (TxBDs) and the corresponding Tx data continuously until the Tx FIFO is full. It does not determine whether the TxBD to be fetched is already being processed internally (as a result of a wrap). As the FEC nears the end of the transmission of one frame, it begins to DMA the data for the next frame. To remain one BD ahead of the DMA, it also fetches the TxBD for the next frame. It is possible that the FEC fetches from memory a BD that has already been processed but not yet written back (it is read a second time with the R bit set). In this case, the data is fetched and transmitted again.

Workaround: Using at least three TxBDs fixes this problem for large frames, but not for small frames. To ensure correct operation for large or small frames, one of the following must be true:

- The FEC software driver ensures that there is always at least one TxBD with the Ready bit cleared.
- Every frame uses more than one TxBD and every TxBD, but the last is written back immediately after the data is fetched.

The FEC software driver ensures a minimum frame size, n. The minimum number of TxBDs is then rounded up to the nearest integer (though the result cannot be less than 3). The default Tx FIFO size is 192 Bytes; this size is programmable.

Fix plan: Currently, there are no plans to fix this.

SECF123: FlexCAN Writing to an Active Receive MB May Corrupt MB Contents

Errata type: Silicon
Affects: FlexCAN
Description: Deactivating a FlexCAN receive message buffer (MB) may cause corruption of another active receive MB, including the ID field, if the following sequence occurs.

1. A receive MB is locked via reading the control/status word, and has a pending frame in the temporary receive serial message buffer (SMB).
2. A second frame is received that matches a second receive MB, and is queued in the second SMB.
3. The first MB is unlocked during the time between receiving the CRC field and the sixth bit of end of frame (EOF) of the second frame.
4. The second MB is deactivated within nine bus clock cycles of the sixth bit of EOF, resulting in corruption of the first MB.

During standard use of the FlexCAN hardware, the errata can appear during heavy communications with several Rx MBs at a low baudrate and while using Rx extended MB’s IDs. This can be easily observed by checking ID value overwrite. In all cases, CAN transmissions from the processor are not affected at any moment.

Workaround: Do not write to the control/status word after initializing a receive MB. If a write (deactivation) is required to the control/status field of an active receive MB, either freeze the FlexCAN module or insert a delay of at least 27 CAN bit times plus 10 bus clock cycles between unlocking one MB and deactivating another MB. This avoids MB corruption; however, frames may still be lost.

Workaround: The FlexCAN software driver ensures IDs are not changed during each reception. As soon as it has changed, return to original value.

Fix plan: Currently, there are no plans to fix this.

SECF124: Buffered Write May Be Executed Twice

Errata type: Silicon
Affects: Cache
Description: If buffered writes are enabled using the CACR or ACR registers, the imprecise write transaction generated by a buffered write may be executed twice.

Workaround: Do not enable buffered writes in the CACR or ACR registers:

- CACR[8] = DBWE (default buffered write enable) must be 0
- ACRn[5] = BUFW (buffered write enable) must be 0

Fix plan: Currently, there are no plans to fix this.

SECF125: Any FlexCAN MB access during RX or TX of an extended ID frame’s CRC and EOF may cause unwanted message reception

Errata type: Silicon
Affects: FlexCAN
Description: With extended ID frames, a frame may be received into a receive message buffer (MB) if a match occurs for the ID_HIGH part of the frame's extended ID, irrespective of the ID_LOW and the acceptance mask bits MID[14:0]. So unwanted messages, which should be filtered by the hardware mask register, may be received and so the ID field of the receive MB may be overwritten by the received frame's ID.

This issue only happens to the messages with the extended ID when the match occurs for the extended ID bits 28 to 15.

Messages with standard ID have no such issue.

Workaround: Use only the Standard ID format for all messages, not the extended format.

Workaround: If extended IDs are used, ensure that only ID bits 28 to 15 are used as the filter criteria, so that other ID bits (ID bits 14 to 0) are not used to filter messages. ID bits 14 to 0 may contain information not used for message filtering purposes.

Fix plan: Currently, there are no plans to fix this.

SECF036A: PLL Does Not Lock in Normal PLL Mode with Crystal Reference

Errata type: Silicon

Affects: PLL

Description: During a power on reset, if the CLKMOD[1:0] equals 11 (normal PLL mode with crystal reference), the PLL does not lock and the device never comes out of reset.

Workaround: When configuring the PLL for normal PLL mode with crystal reference, tie CLKMOD1 to RSTI and not straight to 3.3V. This allows the PLL to correctly detect the desired operating mode and lock.

Fix plan: Currently, there are no plans to fix this.

SECF015: Internal Flash Speculation Address Qualification Incomplete

Errata type: Silicon

Affects: Flash controller

Description: The flash controller uses a variety of advanced techniques, including two-way 32-bit bank interleaving, address speculation, and pipelining to improve performance. An issue involving a complex series of interactions between the local flash controller and other memory accesses (internal SRAM, EIM, or SDRAM) has been uncovered. In rare instances, the interaction between a non-flash memory access and a flash access can result in incorrect data usage for a read operation. This may produce unexpected exceptions, incorrect execution, or silent data corruption.

The problem requires two accesses where the modulo (flash size) address and address mask configuration are the same for both a flash access and a non-flash access that occur close in time.

Workaround: **Workaround Step 1 (Always do this)**: Use FLASHBAR[6] to disable the address speculation mechanisms of the flash controller. The default configuration (FLASHBAR[6] = 0) enables the address speculation. If FLASHBAR[6] equals 1, address speculation is disabled. Core performance may be degraded from 4% – 9%, depending heavily on application code.

NOTE

FLASHBAR[6] is user accessible via the movec instruction. FLASHBAR[6] always reads back as 0.

NOTE

On MCF528x and MCF521x devices FLASHBAR[6] is already set to 1 for datecodes XXX0327 and later. The bit still reads back as 0.

Workaround Step 2a (Select one of the step 2 options to use): Construct the device memory map so the flash and SRAM spaces are disjoint within the modulo-(flash_size) addresses. In some cases if this approach is selected, the upper portion of the flash memory might be unused and the SRAM be mapped to this unused flash space.

Consider an example where the flash memory size is 256 Kbytes and the on-chip SRAM size is 32 Kbytes. If 224 Kbytes or less of flash are used, the SRAM can be based at the upper 32 Kbytes (within the modulo-256 Kbyte address) of the flash address space:

```
Flash: size = 0x40000, base = 0x0000_0000
RAM: size = 0x08000, base = 0x8003_8000 = RAM_BASE+(256-32) Kbytes
```

where the flash and SRAM base addresses are unique BA[31:16].

In summary, this approach can be applied if the combined size of the used flash and used SRAM is less than the total flash size, with the flash contents justified to the lower address range and the SRAM contents justified to the upper address range.

Workaround Step 2b (Select one of the step 2 options to use): Separate the contents of the SRAM and the flash memory into exclusive categories and use the address space mask bits in FLASHBAR and RAMBAR to restrict accesses. For example, if the flash contains only instructions and the SRAM contains only operands (all data), the appropriate address space mask fields are specified to prevent flash and SRAM accesses from overlapping.

Workaround Step 3a (Select one of the step 3 options to use if external parallel memory is used in the system): Do not enable caching of external memories. With caching disabled the timing requirements for an issue to occur will not be met, so this will prevent conflicts between flash and external parallel memory accesses through the EIM or SDRAMC.

Workaround Step 3b (Select one of the step 3 options to use if external parallel memory is used in the system): Separate the contents of the EIM and/or SDRAM and the flash memory into exclusive categories and use the address space mask bits in FLASHBAR, CSMRn, and DMRn to restrict accesses. For example, if the flash contains only instructions and the SDRAM contains only operands (all data), the appropriate address space mask fields are specified to prevent flash and SRAM accesses from overlapping.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.