Document Number: IMX28CE
Rev. 2, 09/2012

Chip Errata for the i.MX28

This document details all known silicon errata for the i.MX28. Table 1 provides a revision history for this document.

Table 1. Document Revision History

Rev. Number	Date	Substantive Change(s)
2	09/2012	• Added errata ERR002656, ERR002360, 2765, 2858, 2814, 2811, TKT131240, 5837 and TKT140334.
1	09/2010	• Updated Table 2 • Deleted ENGR119940
0	03/2010	Initial release.

Table 2 provides a cross-reference to match the revision code to the revision level marked on the device.

Table 2. Revision Level to Part Marking Cross-Reference

MCIMX28 Revision	Package	Device Marking	Mask Revision
1.2	14 x 14	MCIMX283DVM4B	M06Z
1.2	14 x 14	MCIMX283CVM4B	M06Z
1.2	14 x 14	MCIMX286DVM4B	M06Z
1.2	14 x 14	MCIMX286CVM4B	M06Z
1.2	14 x 14	MCIMX287CVM4B	M06Z
1.2	14 x 14	MCIMX281AVM4B	M06Z
1.2	14 x 14	MCIMX285AVM4B	M06Z

freescale semiconductor

Table 2. Revision Level to Part Marking Cross-Reference

MCIMX28 Revision	Package	Device Marking	Mask Revision
1.2	14 x 14	MCIMX280DVM4B	M06Z
1.2	14 x 14	MCIMX280CVM4B	M06Z

Table 3 summarizes all known errata.

Table 3. Summary of Silicon Errata and Applicable Revision

Errata	Name	Projected Solution
ENGR116296	HSADC: Soft reset causes unexpected request to DMA	No fix scheduled
ENGR116904	PXP: The HW_PXP_CSCCOEFF2_C3 register can not be reset correctly under some PVT corner	No fix scheduled
ENGR119650	USB: USB core INCR8 and INCR16 modes are inoperable	No fix scheduled
ENGR119653	USB: ARM to USB register error issue	No fix scheduled
ENGR119657	PWM: Register write sync issue when HSADC clock frequency is lower than APBX clock frequency	No fix scheduled
ENGR119956	CLKCTRL: ENET 1588 clock (CLK_ENET_TIME) is not under control of ENET disable control bit	No fix scheduled
ENGR121613	ENET: ENET big endian mode not compatible with ARM little endian	No fix scheduled
ENGR121616	DMA: APBH/APBX DMA channel can stall while waiting to access a APBH/APBX bus peripheral when the channel freeze bit is set	No fix scheduled
ERR002656	FlexCAN: Abort request blocks the CODE field	No fix scheduled
ERR002360	FlexCAN: Global Masks misalignment	No fix scheduled
2765	EMI Clock: Switching to synchronous mode error	No fix scheduled
2858	USB controller may access a wrong address for the dTD (endpoint transfer descriptor) and then hangs	No fix scheduled
2814	The DCDC converters unexpectedly turn on when 5 V is removed while the DCDC_XFER bit is clear	No fix scheduled
2811	Unreliability of VDD5V_GT_VDDIO functionality	No fix scheduled
TKT131240	SSP0/1-SD/MMC/eMMC Boot: SSP_SCK polarity setup issue in ROM	No fix scheduled
5837	Setting the ENABLE_DCDC bit in the HW_POWER_DCDC4P2 or HW_POWER_5VCTRL registers can result in false brownout Detection	No fix scheduled
TKT140334	ONFI 3.0 NAND boot-up issue	No fix scheduled

ENGR116296 HSADC: Soft reset causes unexpected request to DMA

Description:

When HSADC is initially powered on and performs a soft reset, the APBH-DMA may receive an unexpected request from the HSADC and enter into an unknown state.

Projected Impact:

HSADC operates incorrectly.

Workaround:

Software workaround 1:

Before completing a normal soft reset, make a timing sequence of CLKGATE and SFTRST by register programming to reset FIFO read and write pointer, and make RD_EMPTY to the known state of 1 before normal operation.

The register programming below can be used to generate the timing sequence of CLKGATE and SFTRST before a normal soft reset.

- HW_HSADC_CTRL0_CLR(BM_HSADC_CTRL0_SFTRST);
- HW_HSADC_CTRL0_WR((HW_HSADC_CTRL0_RD() | BM_HSADC_CTRL0_SFTRST) & (~BM_HSADC_CTRL0_CLKGATE));
- HW_HSADC_CTRL0_SET(BM_HSADC_CTRL0_CLKGATE);
- HW_HSADC_CTRL0_CLR(BM_HSADC_CTRL0_CLKGATE);
- HW_HSADC_CTRL0_SET(BM_HSADC_CTRL0_CLKGATE);

NOTE

This sequence is only required for the first reset after power up.

Software workaround 2:

Complete a HSADC DMA channel reset after the HSADC module soft reset and before configuring/enabling the HSADC DMA channel as shown below:

- HW_APBH_CHANNEL_CTRL.B.RESET_CHANNEL = 0x1000;
- HSADC_CTRL0_CLKGATE and HSADC_CTRL0_SFTRST address is 0x80002000.

Projected Solution:

No fix scheduled.

ENGR116904 PXP: The HW_PXP_CSCCOEFF2_C3 register can not be reset correctly under some PVT corner

Description:

The HW_PXP_CSCCOEFF2_C3 register does not receive the correct reset value after using the PXP software reset function or after power up.

Projected Impact:

The PXP operates incorrectly with wrong coefficient setting.

Workaround:

Always write the HW_PXP_CSCCOEFF2_C3 register with the expected value before using it. The PXP_CSCCOEFF2 register address is 0x8002A0F0.

Projected Solution:

No fix scheduled.

ENGR119650 USB: USB core INCR8 and INCR16 modes are inoperable

Description:

The USB controller may not operate properly when receiving a packet in INCR8 and INCR16 modes. The packet is completed correctly (ACK is sent) on the USB bus, but cannot be seen by software.

This issue exists when all of following conditions are met:

1. Controller is receiving data (Host Bulk IN or Device Bulk OUT)
2. Primary INCR8/INCR16 mode is selected (SBUSCFG. AHBBRST of the USB register is set to 0b010 or 0b011)
3. Length of data received is less than the total_byte field in TD
4. Data length is not a multiple of the burst size and the remainder is a sub-burst. For example, if the data length is 32n + 16 bytes in INCR8 mode, or 64n + 16/32/48 in INCR16 mode, this errata is triggered.

Projected Impact:

This is a low severity bug because INCR8 and INCR16 are not mandatory modes. Other modes should be used.

Workaround:

Set SBUSCFG.AHBBRST of the USB register to a modes other than 0b010 or 0b011.

Projected Solution:

No fix scheduled.

ENGR119653 USB: ARM to USB register error issue

Description:

The ARM writes a data error to the USB core register unless SRM SWP instruction is used.

The issue occurs when all of the following conditions are met:

1. Last AHB access is to the non-USB AHB slave
2. Current AHB access is to the USB
3. These two accesses are back-to-back
4. The last data phase of the last AHB access has a wait state
5. Only happens when D-cache is enabled

Projected Impact:

The USB register does not get correct data when writing to the USB slave through the AHB bus when D-cache is enabled.

Workaround:

All USB register write operations must use the ARM SWP instruction.

Projected Solution:

No fix scheduled.

ENGR119657 PWM: Register write sync issue when HSADC clock frequency is lower than APBX clock frequency

Description:

The PWM channel might not generate the required output signal when in HSADC driving mode. When in HSADC mode, if the HSADC input clock is much lower than the APBX bus clock (for example APBX Bus clock is 24 MHz and HSADC input clock is 4 MHz) the write signal to the PWM registers is missed. Write access to the following registers has no effect after HSADC mode is enabled:

- PWM Control and Status Register
- PWM Channel Active Register
- PWM Channel Period Register

As a result, dedicated PWM channel is not triggered.

Projected Impact:

HSADC or off chip linear sensor does not receive the required control signals.

Workaround:

If the HSADC input clock is lower than the 24 MHz APBX bus clock the APBX bus clock should be set to a lower frequency before every write to the PWM register. When the write access finishes, the APBX clock can be set back to normal.

Projected Solution:

No fix scheduled.

ENGR119956 CLKCTRL: ENET 1588 clock (CLK_ENET_TIME) is not under control of ENET disable control bit

Description:

The Ethernet 1588 clock (CLK_ENET_TIME) continues to toggle when the Ethernet module is disabled by setting ENET disable control bit in the HW_CLKCTRL_ENET register. The ethernet controller consumes 30 µA on the 4.2 V power supply.

Projected Impact:

The Ethernet 1588 clock consumes 30 µA on the 4.2 V power supply when the Ethernet module is disabled.

Workaround:

The Ethernet 1588 clock can be gated off by clearing the HW_CLKCTRL_ENET_DIV_TIME register. The HW_CLKCTRL_ENET_DIV_TIME register address is 0x80040140.

Projected Solution:

No fix scheduled.

ENGR121613 ENET: ENET big endian mode not compatible with ARM little endian

Description:

The endian mode of the Ethernet controller is designed to be big-endian mode which is not compatible with the ARM core and reset sections of the device.

Projected Impact:

The ARM core cannot establish data communication correctly to/from the Ethernet controller without software endian conversion.

Workaround:

When communicating with the Ethernet controller, an additional byte-swap routine has to be called by the ARM core.

Projected Solution:

No fix scheduled.

ENGR121616 DMA: APBH/APBX DMA channel can stall while waiting to access a APBH/APBX bus peripheral when the channel freeze bit is set

Description:

When the channel freeze bit is set, the APBH/APBX DMA channel can stall while waiting to access a peripheral on the APBH/APBX bus. This occurs if the channel freeze bit is set exactly at the same time as when the channel internal state machine changes from the PIO_REQ state to the REQ_WAIT state.

Projected Impact:

The data communication with the APBH/APBX DMA channel associated peripheral is stalled.

Workaround:

Do not use DMA PIO operation to configure the associated peripheral when using channel freeze function. Use ARM PIO operation instead.

Projected Solution:

No fix scheduled

No fix scheduled.

ERR002656 FlexCAN: Abort request blocks the CODE field

Description:

An Abort request to a transmit message buffer (TxMB) can block any write operation into its CODE field. Therefore, the TxMB cannot be aborted or deactivated until it completes a valid transmission (by winning the CAN bus arbitration and transmitting the contents of the TxMB).

Projected Impact:

The TxMB cannot be aborted or deactivated until it completes a valid transmission.

Workaround:

Instead of aborting the transmission, use deactivation instead.

Note that there is a chance the deactivated TxMB can be transmitted without setting IFLAG and updating the CODE field if it is deactivated.

Projected Solution:

No fix scheduled.

ERR002360 FlexCAN: Global Masks misalignment

Description:

During CAN message reception by FlexCAN, the RXGMASK (Rx Global Mask) is used as an acceptance mask for most of the Rx message buffers (MB). When the FIFO Enable bit in the FlexCAN Module Configuration Register (CANx_MCR[FEN], bit 29) is set, the RXGMASK also applies to most of the elements of the ID filter table. However, there is a misalignment between the position of the ID field in the RxMB and that in the RXIDA, RXIDB and RXIDC fields of the ID tables. In fact, the RXIDA filter in the ID tables is shifted one bit to the left from the RxMBs ID position, as shown below:

RxMB ID = bits 28-0 of ID word corresponding to message ID bits 28-0

RXIDA = bits 29-1 of ID Table corresponding to message ID bits 28-0

Note that the mask bits align to the ID filter bits, not to the incoming ID bits. For example, the bit 4 in RXGMASK masks bit 4 in RxMB ID, but it does not mask bit 4 in incoming message ID. This misalignment leads the RXGMASK to affect RxMB and Rx FIFO filtering in different ways.

For example, if the user intends to mask out bit 4 of the ID filter of message buffer then the RXGMASK will be configured as 0xffff_ffef. As a result, bit 4 of the ID field of the incoming message is ignored during the filtering process for message buffers. This very same configuration of RXGMASK, which would lead bit 4 of RXIDA to be “do not care” and thus bit 3 of the ID field of the incoming message would be ignored during the filtering process for the Rx FIFO.

Similarly, both RXIDB and RXIDC filters have multiple misalignments with regards to position of the ID field in the RxMBs, which can lead to erroneous masking during the filtering process for either Rx FIFO or MBs.

RX14MASK (Rx 14 Mask) and RX15MASK (Rx 15 Mask) have the same structure as the RXGMASK. This includes the misalignment problem between the position of the ID field in the RxMBs and in the RXIDA, RXIDB and RXIDC fields of the ID Tables.

Projected Impact:

Leading to mask misalignment between RxMB and Rx FIFO filtering.

Workaround:

It is recommended that one of the following actions be taken to avoid problems:

- Do not enable the RxFIFO. If CANx_MCR[FEN]=0 then the Rx FIFO is disabled and thus the masks RXGMASK, RX14MASK and RX15MASK do not affect it.

- Enable Rx Individual Mask Registers. If the Backwards Compatibility Configuration bit in the FlexCAN Module Configuration Register (CANx_MCR[BCC], bit 16) is set then the Rx Individual Mask Registers (RXIMR0-63) are enabled and thus the masks RXGMASK, RX14MASK and RX15MASK are not used.

- Do not use the masks RXGMASK, RX14MASK and RX15MASK (leave them in reset value which is 0xffff_ffff) when CANx_MCR[FEN]=1 and CANx_MCR[BCC]=0. In this case, filtering processes for both RxMBs and Rx FIFO are not affected by those masks.

- Do not configure any MB as Rx (leave all MBs as either Tx or inactive) when CANx_MCR[FEN]=1 and CANx_MCR[BCC]=0. In this case, the masks RXGMASK, RX14MASK and RX15MASK can be used to affect ID tables without affecting filtering process for RxMBs.

Projected Solution:

No fix scheduled.

2765 EMI Clock: Switching to synchronous mode error

Description:

In order to switch the EMI clock from asynchronous to synchronous mode, both the xtal_ref and the cpu_ref must be active, even if both BYPASS_CPU and BYPASS_EMI are set in the HW_CLKCTRL_CLKSEQ register.

Projected Impact:

The i.MX28 locks up.

Workaround:

If cpu_ref is inactive, then activate the cpu_ref before attempting to switch to synchronous mode.

Projected Solution:

No fix scheduled

2858 USB controller may access a wrong address for the dTD (endpoint transfer descriptor) and then hangs

Description:

Currently, software checks the active bit in dTD to see whether it is finished. If the Active bit is 0, then software frees the allocated memory for the dTD.

The hardware sequence after all data of a dTD is transferred is as follows:

1. Update the dTD. This includes an AHB write access of three DWords. The active bit is cleared in the first DW write.
2. Update the qHead (this includes an AHB write access of three DWords).
3. Read the dTD again to check if software added a new dTD (this is a SINGLE AHB read). At the same time, send out an interrupt if needed.

After step 1, if software finds the Active bit is cleared, then the dTD memory space is freed and may be allocated for another thread's use. In step 3, hardware may get a wrong dTD.

This issue does not occur if some delay is added before freeing the dTD memory space.

This issue only occurs in USB INCR8 mode, because steps 1 and 2 have 6 SINGLE AHB transfers in INC8 mode, but only two burst AHB transfers in INCR mode.

This issue only occurs when the dTD list is used; because if only one dTD is used, the software only checks the Active bit after an interrupt is received (step 3). However, when the dTD list is used, the software may check the entire list after the interrupt for the first dTD is received, when the hardware has just finished the transfer of the second dTD.

Figure 1 illustrates this issue.

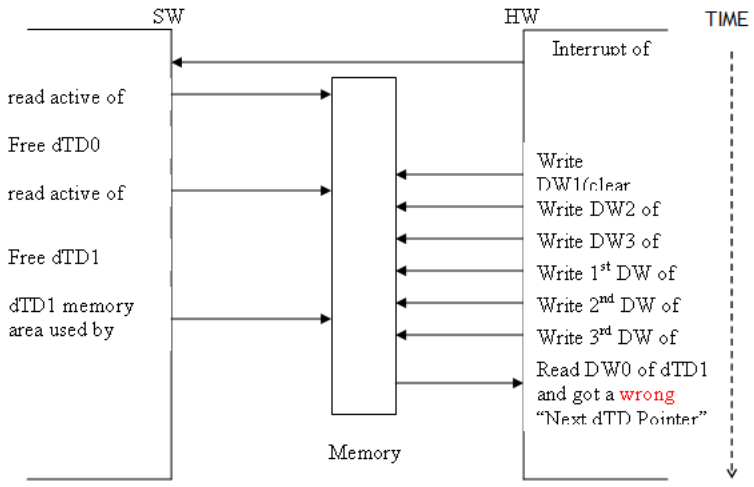

Figure 1. dTD Issue Sequence of Events

Projected Impact:

USB Controller may hang if dTD is freed too quickly.

Workaround:

Postpone freeing the current dTD; free it when its next dTD can be freed, so the last completed dTD (followed by an ACTIVE dTD) is always freed when the next IOC irq comes.

Projected Solution:

No fix scheduled

2814 The DCDC converters unexpectedly turn on when 5 V is removed while the DCDC_XFER bit is clear

Description:

If the DCDC_XFER bit is clear, the DCDC converter should not automatically turn on when 5 V is removed. Instead, a power-down should occur if 5 V is removed, when DCDC_XFER and ENABLE_DCDC are zero.

Projected Impact:

DCDC converter input source may switch to DCDC_BATT pin but no power source is present there. This may latch-up the DCDC converter circuit.

Workaround:

In order to power down the system properly when 5 V is removed, set PWDN_5VBRNOUT bit in Register HW_POWER_5VCTRL.

Projected Solution:

No fix scheduled

2811 Unreliability of VDD5V_GT_VDDIO functionality

Description:

Due to unreliability of the VDD5V_GT_VDDIO functionality, the power supply should never be configured to be used as the 5-V plug/unplug detection method.

Projected Impact:

VDD5V_GT_VDDIO output may not change to “0” (and VDD5V_GT_VDDIO_IRQ may not be triggered) when 5 V is unplugged.

Workaround:

Use the VBUSVALID comparator for 5V plug/unplug detection. Actually, the VBUSVALID comparator is recommended in the reference manual, not VDD5V_GT_VDDIO.

The VBUSVALID_5VDETECT bit in HW_POWER_5VCTRL should be set to “1” (it’s default value) and never cleared. The detection threshold can be changed in the VBUSVALID_TRSH bit field in HW_POWER_5VCTRL.

Projected Solution:

No silicon fix scheduled.

TKT131240 SSP0/1-SD/MMC/eMMC Boot: SSP_SCK polarity setup issue in ROM

Description:

When boot mode is set as boot from SD/MMC/eMMC on SSP0/1, the SSP_SCK polarity is not correctly set up in ROM. The POLARITY bit in HW_SSP_CTRL1 register should be set to “1” (command and data output on falling edge of clock) according to SD/MMC/eMMC specification. However, the POLARITY bit is set to “0” in ROM in the existing silicon (TO1.2). As a result, input setup time (tISU) at SD/MMC/eMMC input may not be met.

Projected Impact:

Write command error may occur when booting from the SD/MMC/eMMC on SSP0/1 and result in boot failure.

Workaround:

If tISU at SD/MMC/eMMC input is violated and write command error occurs during boot from SD/MMC/eMMC, a ROM patch of 1kByte size loaded from the EEPROM is required to fix this issue. Boot mode should be set to [0001] for the EEPROM on I2C0 or [1000] for the EEPROM on SPI3. The patch executes from the EEPROM, patches the ROM SSP driver code, and switches boot to either SSP0 or SSP1. There are separate patch binaries to boot from SSP0 and SSP1.

Projected Solution:

No silicon fix scheduled.

5837 Setting the ENABLE_DCDC bit in the HW_POWER_DCDC4P2 or HW_POWER_5VCTRL registers can result in false brownout Detection

Description:

When the ENABLE_DCDC bit in HW_POWER_DCDC4P2 or HW_POWER_5V_CTRL is set, a glitch is propagated through the brownout comparators. If the glitch is sufficiently large, it can cause a false brownout detection. The VDDD, VDDA, VDDIO, and VBUSVALID comparators are all susceptible to the glitch.

Projected Impact:

Can result in a false brownout detection.

Workaround:

The sequence below is needed to work around this issue prior to setting the ENABLE_DCDC bit in HW_POWER_DCDC4P2:

1. Disable the power rail brownout interrupts (clear HW_POWER_CTRL VDDA, VDDD, VDDIO ENIRQ bits).
2. Set the HW_POWER_5VCTRL PWRUP_VBUS_CMPS bit.
3. Set the HW_POWER_5VCTRL VBUSVALID_TRSH to 0x0 (2.9 V).
4. Set the HW_POWER_5VCTRL VBUSVALID_5VDETECT bit to 1.
5. Disable VBUSDROOP status and interrupts (clear VDD5V_DROOP_IRQ).
6. Set the ENABLE_DCDC bit in HW_POWER_DCDC4P2.
7. Wait 100 µs
8. Check VBUSVALID_IRQ bit. If it is set, then set and clear the PWD_CHARGE_4P2 bit to repower on the 4P2 regulator because it is automatically shut off on a VBUSVALID false condition. It may be helpful to ramp up the CHARGE_4P2_ILIMIT value at this point to gradually draw power from 5 V rail. If HW_POWER_5VCTRL ENABLE_DCDC is already set, the DCDC will draw current from VDD4P2 as soon as PWD_CHARGE_4P2 is cleared.
9. Clear VBUSDROOP, VBUSVALID, and the output rails IRQ bits as needed.
10. Restore the output rail ENIRQ bits, the VBUSVALID_TRSH level, VBUSVALID_5VDETECT value, ENIRQ_VBUS_VALID, and ENIRQ_VDD5V_DROOP to their original values.

The sequence below is needed to work-around this issue prior to setting the ENABLE_DCDC bit in HW_POWER_5VCTRL and HW_POWER_DCDC4P2. Note that the below workaround assumes the usual requirements for setting the HW_POWER_5VCTRL ENABLE_DCDC bit are met (that is, the hardware and/or software battery brownout protection mechanism is enabled to properly protect the system against the DCDC sourcing from the battery if the battery voltage is too low).

1. Disable the power rail brownout interrupts (clear HW_POWER_CTRL VDDA, VDDD, VDDIO ENIRQ bits).

2. Set the HW_POWER_5VCTRL PWRUP_VBUS_CMPS bit.
3. Set the HW_POWER_5VCTRL VBUSVALID_TRSH to 0x0 (2.9 V).
4. Set the HW_POWER_5VCTRL VBUSVALID_5VDETECT bit to 1.
5. Disable VBUSDROOP status and interrupts (clear VDD5V_DROOP_IRQ).
6. Set the ENABLE_DCDC bit in HW_POWER_5VCTRL.
7. Wait 100 µs.
8. Check VBUSVALID_IRQ bit. If it is set, and 5 V is present and the VDD4P2 rail was enabled, then repeat the sequence for enabling the 4P2 regulator and DCDC from VDD4P2. This bit indicates that the DCDC has tried to source from the battery, even if 4P2 sourcing is enabled. This is because a VBUSVALID false condition automatically disables the 4P2 regulator, so the DCDC then falls back to battery sourcing.
9. Clear VBUSDROOP, VBUSVALID, and the output rails IRQ bits as needed.
10. Restore the output rail ENIRQ bits, the VBUSVALID_TRSH level, VBUSVALID_5VDETECT value, ENIRQ_VBUS_VALID, and ENIRQ_VDD5V_DROOP to their original values.

Projected Solution:

No silicon fix scheduled.

TKT140334 ONFI 3.0 NAND boot-up issue

Description:

ROM in existing silicon (TO1.2) supports ONFI BA-NAND boot-up. During boot-up it reads Bit 7 of Byte 6-7 (1=supports Block Abstracted access mode) in ONFI NAND device's parameter page to determine if the NAND device is ONFI BA-NAND. BA-NAND memory devices are no longer part of the ONFI spec and memory vendors do not support these devices anymore. However in ONFI 3.0 Spec, that bit has been re-used to specify whether the NAND device supports extended parameter page.

When a system is mounted with ONFI 3.0 NAND device with "supports extended parameter page" bit set to "1", the i.MX28 boot ROM will see it as BA-NAND and will use BA-NAND commands to access the NAND device. As a result, the system will fail to boot-up.

Projected Impact:

System mounted with ONFI 3.0 NAND device will fail to boot-up.

Workaround:

Contact ONFI NAND vendor to supply NAND device with the "supports extended parameter page" bit set to "0", so that the i.MX28 boot ROM will not treat it as BA-NAND.

Projected Solution:

No silicon fix scheduled.

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.

Document Number: IMX28CE
Rev. 2
09/2012