

## Mask Set Errata for Mask 2N40C

### Introduction

This report applies to mask 2N40C for these products:

- S08PT60

Errata ID	Errata Title
4044	BDC: a breakpoint may be missed in some cases
3955	DBG: the code does not stop at the DBG TAG breakpoint address, instead, it may stop a few instructions later following the breakpoint address.
3868	KBI: Port: When a port pin is enabled as a KBI pin, the port data register read is always 0 even if the corresponding PTxIE bit is set (i.e., the port pin input is enabled).
4063	NVM: 2nd flash read returns invalid data and error flags may not be set when flash is busy programming or erasing a block
3957	PMC: The STOP IDD with LVD enabled is high and not stable, oscillating around 530uA
4138	PMC: The part may be damaged at hot temperature
3869	SCI: When SCI is configured in 1 STOP bit mode, a false RDRF flag will be generated after the correct RDRF flag is set.
3996	SCI: the LSB bit will be lost in the receiver side when using the frame format of 2 STOP bits plus 9 data bits
3958	SOC: MCU can not be woken up from STOP mode when an interrupt happens within 6 bus clock cycles after CPU starts to execute the STOP instruction
3959	SOC: SYS_ILLAH and SYS_ILLAL registers can't contain illegal opcode address of ILOP reset
4137	SOC: UUID registers may not contain the real UUID for PC samples
4592	PMC: Unstable to wake up MCU from stop3 mode with LVD disabled when the MCU power supply VDD is in the low range

### e4044: BDC: a breakpoint may be missed in some cases

**Errata type:** Errata

**Description:** The instruction address match may be masked if the breakpoint is set within two address change cycles from PC, and hence no breakpoint hit is generated in this case.

This issue will not affect code running, but affect code debugging.

**Workaround:** Do not set breakpoint within two address change cycles from PC.

This issue will be fixed in the next release.

**e3955: DBG: the code does not stop at the DBG TAG breakpoint address, instead, it may stop a few instructions later following the breakpoint address.**

**Errata type:** Errata

**Description:** If a DBG TAG breakpoint is configured and enabled, the CPU can't break before this tagged instruction is executed, but may break after several clock cycles. This does not affect code running, but affect code debugging.

**Workaround:** Use the BDC breakpoint instead of DBG breakpoints and set only one breakpoint in the code. Before performing "Step Into", "Step Over" and "Step Return" debugger functions, remove all breakpoints.

This issue will be fixed in the next release.

**e3868: KBI: Port: When a port pin is enabled as a KBI pin, the port data register read is always 0 even if the corresponding PTxIE bit is set (i.e., the port pin input is enabled).**

**Errata type:** Errata

**Description:** When a port pin is enabled as a KBI pin, the port data register read is always 0 even if the corresponding PTxIE bit is set (i.e., the port pin input is enabled). This issue applies to all port pins that are shared with higher priority peripherals. When any higher priority peripheral controls the I/O pin, the port data register always get value of 0.

**Workaround:** Disable the peripheral control on the I/O port pins followed by reading port data register to a variable, then re-enable the peripheral control on the I/O pins, and at last check the variable bit value for the pin. This limits the frequency of external signals. Below is an example workaround code for checking KBI1 pin 5 and pin 4 states:

```
byte buttonStates;
// NOTE: disable KBI1 pin function before reading PORT data register
KBI1_PE = 0;
buttonStates = PORT_PTDD;
// NOTE: restore KBI pin function after PORT data register
KBI1_PE = KBI1_PE_KBIPE4_MASK | KBI1_PE_KBIPE5_MASK; /* enable KBI1 pin 5 and 4
*/
KBI1_SC_KBACK = 1; // clear KBI1 interrupt flag
// Check pin states
if( !(buttonStates & (1<<5)))
{
// handle KBI1 pin 5 pressed event
...
}
```

```

}
if( !(buttonStates & (1<<4)))
{
// handle KBI1 pin 4 pressed event
...
}

```

### **e4063: NVM: 2nd flash read returns invalid data and error flags may not be set when flash is busy programming or erasing a block**

**Errata type:** Errata

**Description:** A corner case appears with the following sequence of events:

- The Flash is busy programming/erasing a block (CCIF=0)
- the application tries to read from PFlash, causing a collision error (not all PFlash reads would cause such an error)
- a read collision error is reported (SFIF and DFIF flags set together)
- the application clears the error flags
- the application reads from the same 32-bit longword that originally caused the error (could be the same address that originated the issue, but not necessarily)
- this second read does not return meaningful data, error flags may not set to indicate such situation.

**Workaround:** The recommended operation for the Flash is to avoid reading from a block (flash or EEPROM) while the Flash is busy programming or erasing that same block. To get an indication if the Flash is busy the application can check the status of flag CCIF: while CCIF=0 the Flash is busy and the application will not read meaningful data from the block being erased or programmed, and should avoid reading.

Another workaround is as below which is not recommended:

After the collision was detected and the error flags were cleared by the user, the application should execute a dummy read from another 32-bits longword (in a different address than the one that caused the collision error), before reading the Flash block again. If the collision error sets again, should be cleared accordingly.

This issue will be fixed in the next release.

### **e3957: PMC: The STOP IDD with LVD enabled is high and not stable, oscillating around 530uA**

**Errata type:** Errata

**Description:** When LVD is enabled in stop mode, the voltage reference refresh logic will stay off. This results in several issues: the elevated IDD, the loss of currents to the 32khz low power oscillator, the 1khz clock will lock, and ACMP. In general, this condition is unsafe to operate in for more than 1 second at room temperature, 50ms at hot temperature and around 5-10 seconds at cold temperature.

**Workaround:** Do not enable LVD when entering STOP mode. This issue will be fixed in the next release.

## e4138: PMC: The part may be damaged at hot temperature

**Errata type:** Errata

**Description:** At hot temperature above 85°C, the latchup may be triggered with rapid VDD ramp time, causing EOS and resulting in elevated IDD. This may damage the device.

**Workaround:** Add a large capacitor on VDD to slow the VDD ramp time. It will be fixed in the next release.

## e3869: SCI: When SCI is configured in 1 STOP bit mode, a false RDRF flag will be generated after the correct RDRF flag is set.

**Errata type:** Errata

**Description:** The SCI receiver sets Receive Data Register Full (RDRF) bit twice when 1 STOP bit frame format is used. In this case, the first RDRF bit is correct, but the second RDRF bit is a false flag. The false RDRF is generated after one bit time of the first RDRF gets set.

So if there is no delay or the delay is shorter than 1 bit time in between two data transmit frames from the transmitter, the S08PT60 SCI receiver will generate a frame error in addition to the RDRF flag.

**Workaround:** 1 bit time delay is required before reading SCI Data Register when RDRF bit is set. In addition, in order for S08PT60 to receive the right data bytes in the data stream, the transmitter must delay at least one bit time in between two consecutive data frames.

Example code snippet for SCI1 Rx polling mode is as below:

```
if(SCI1_S1 & SCI1_S1_RDRF_MASK) /* is the first RDRF set? */
{
    /* delay 1 bit time to allow false flag to come*/
    delay_1_bit_time();
    dummy = SCI1_S1; /* read S1 to clear the false RDRF and other flags */
    ucRxBuff[i++] = SCI1_D; /* read true data and store it to a buffer */
}
```

Example code snippet for SCI1 Rx interrupt mode:

```
interrupt VectorNumber_Vsci1rx void SCI1_RxISR( void )
{
    if(SCI1_S1 & SCI1_S1_RDRF_MASK) /* is the first RDRF set? */
    {
        /* delay 1 bit time to allow false flag to come*/
        delay_1_bit_time();
        dummy = SCI1_S1; /* read S1 to clear false RDRF and other flags */
        ucRxBuff[i++] = SCI1_D; /* read true data and store it to a buffer */
        /* Place your other code here */
    }
}
```

**e3996: SCI: the LSB bit will be lost in the receiver side when using the frame format of 2 STOP bits plus 9 data bits**

**Errata type:** Errata

**Description:** The LSB bit will be lost in the receiver side when using the frame format of 2 STOP bits plus 9 data bits: the data shifter will shift 1 STOP bit as a MSB of the data into the data register and the LSB of the data is lost when the parity is disabled. In case of parity enabled, the data shifter will take 1 STOP bit as a parity bit and shift the LSB out. So the LSB of the data will be lost too in this case.

**Workaround:** There is no workaround for this issue. Do not use the format of 2 STOP bits plus 9 data bits. This issue will be fixed in the next release.

**e3958: SOC: MCU can not be woken up from STOP mode when an interrupt happens within 6 bus clock cycles after CPU starts to execute the STOP instruction**

**Errata type:** Errata

**Description:** The MCU has some problem when handling the stop abort feature. When an interrupt is generated within 6 bus clock cycles after CPU starts to execute the STOP instruction, the CPU will respond to the interrupt and abort the STOP. This action will mask all the asynchronous interrupts. However, the MCU system doesn't abort STOP which causes the system clock to be stopped. Synchronous interrupts are disabled due to no clock to use. As a result, no interrupt can wake up the MCU from STOP state.

**Workaround:** Avoid generating interrupt in the time frame of 0 ~ 6 bus clock cycles after STOP instruction is executed by CPU. This issue will be fixed in the next release.

**e3959: SOC: SYS\_ILLAH and SYS\_ILLAL registers can't contain illegal opcode address of ILOP reset**

**Errata type:** Errata

**Description:** When the illegal opcode reset occurs, the MCU can not capture the illegal opcode address into SYS\_ILLAH and SYS\_ILLAL registers.

**Workaround:** Do not use the address in SYS\_ILLAH and SYS\_ILLAL registers as illegal opcode address.

**e4137: SOC: UUID registers may not contain the real UUID for PC samples**

**Errata type:** Errata

**Description:** For PC samples, the real UUID may not be programmed to the universally unique identification (UUID) registers.

**Workaround:** The UUID will be programmed to the UUID registers during mass production.

**e4592: PMC: Unstable to wake up MCU from stop3 mode with LVD disabled when the MCU power supply VDD is in the low range**

**Errata type:** Errata

**Description:** The PMC does not provide enough current while waking up MCU from stop3 mode if the MCU power supply VDD is in the low range. If LVD enabled with LVDRE bit set and LVDSE bit clear, the LVD reset may occur with high MCU power supply VDD. The observed VDD can be as high as 4V when LVD reset on wakeup happens.

**Workaround:** The low power stop3 mode can only be used at VDD from 5V down to the high LVD trip (VLVDH) safely (4.2V to 5.5V). The low power stop3 mode cannot be used from 2.7V to 4.2V. High power stop3 mode (LVDE and LVDSE fields set) can still be used for low voltage applications.

It'll be fixed in future silicon revision.

## **How to Reach Us:**

### **Home Page:**

[www.freescale.com](http://www.freescale.com)

### **Web Support:**

<http://www.freescale.com/support>

### **USA/Europe or Locations Not Listed:**

Freescale Semiconductor  
 Technical Information Center, EL516  
 2100 East Elliot Road  
 Tempe, Arizona 85284  
 +1-800-521-6274 or +1-480-768-2130  
[www.freescale.com/support](http://www.freescale.com/support)

### **Europe, Middle East, and Africa:**

Freescale Halbleiter Deutschland GmbH  
 Technical Information Center  
 Schatzbogen 7  
 81829 Muenchen, Germany  
 +44 1296 380 456 (English)  
 +46 8 52200080 (English)  
 +49 89 92103 559 (German)  
 +33 1 69 35 48 48 (French)  
[www.freescale.com/support](http://www.freescale.com/support)

### **Japan:**

Freescale Semiconductor Japan Ltd.  
 Headquarters  
 ARCO Tower 15F  
 1-8-1, Shimo-Meguro, Meguro-ku,  
 Tokyo 153-0064  
 Japan  
 0120 191014 or +81 3 5437 9125  
[support.japan@freescale.com](mailto:support.japan@freescale.com)

### **Asia/Pacific:**

Freescale Semiconductor China Ltd.  
 Exchange Building 23F  
 No. 118 Jianguo Road  
 Chaoyang District  
 Beijing 100022  
 China  
 +86 10 5879 8000  
[support.asia@freescale.com](mailto:support.asia@freescale.com)

### **For Literature Requests Only:**

Freescale Semiconductor Literature Distribution Center  
 1-800-441-2447 or +1-303-675-2140  
 Fax: +1-303-675-2150  
[LDCForFreescaleSemiconductor@hibbertgroup.com](mailto:LDCForFreescaleSemiconductor@hibbertgroup.com)

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

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor 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 Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2011-2012 Freescale Semiconductor, Inc.