CY7C67200

Errata Revision: *A

June 25, 2004

Errata Document for CY7C67200

This document describes the errata for the CY7C67200. Details include errata trigger conditions, available workarounds, and silicon revision applicability. This document should be used to compare to the datasheet for this device to fully describe the device functionality.

Please contact your local Cypress Sales Representative if you have further questions.

Part Numbers Affected

Part Number	Device Characteristics
CY7C67200	All Packages

CY7C67200 Qualification Status

In Production

CY7C67200 Errata Summary

The following table defines the errata applicability to available CY7C67200 family devices. An "X" indicates that the errata pertains to the selected device.

Note: Errata titles are hyperlinked. Click on table entry to jump to description.

Items	CY7C67200	Rev Letter	Fix Status
1. HPI write to SIE registers.	X	A	No fix is currently planned for future silicon versions. Use workaround.
2. IDE register read when GPIO24 pin is low.	X	A	No fix is currently planned for future silicon versions. Use workaround.
3. UART does not recognize framing errors.	X	A	No fix is currently planned for future silicon versions. Use workaround.
4. UART Does not override GPIO Control Register.	X	A	No fix is currently planned for future silicon versions. Use workaround.
5. VBUS Interrupt (VBUS Valid) Requires Debouncing.	X	A	No fix is currently planned for future silicon versions. Use workaround.
6. Coupled SIE Interrupt Enable Bits.	X	A	No fix is currently planned for future silicon versions. Use workaround.
7. Un-Initialized SIExmsg Registers	X	A	No fix is currently planned for future silicon versions. Use workaround.
8. BIOS USB Peripheral Mode: Descriptor Length.	X	A	No fix is currently planned for future silicon versions. Use workaround.
9. Peripheral short packet issue	X	A	Will be fixed in future silicon revision. Use workaround.
10. Data toggle corruption issue	X	A	Will be fixed in future silicon revision. Use workaround.

1. HPI write to SIE registers.

- **PROBLEM DEFINITION**

  Writing to the SIE2 Control register via HPI can corrupt the SIE1 control register.

  Writing to the SIE1 Control register via HPI can corrupt the SIE2 control register.

- **PARAMETERS AFFECTED**

  SIE control registers

- **TRIGGER CONDITION(S)**

  When an external processor accesses the SIE1 or SIE2 register at the same time the internal CY16 CPU is also accessing the opposite SIE, the SIE accessed by the CY16 CPU will be corrupted.

  For example, the external processor writes a value of 0x80 to the SIE2 register 0xC0B0 while the internal CY16 is doing a read/write to the SIE1 register 0xC08C, the SIE1 register 0xC08C, will be corrupted with the value 0x80.

- **SCOPE OF IMPACT**

  If the internal CPU and external CPU access the SIEs at the same time, contention will occur resulting in incorrect data in one of the SIE registers.

- **WORKAROUND**

  1. Use the LCP COMM_WRITE_CTRL_REG to handle the writing to SIE registers.

  2. Use download code to handle SIE WRITE commands.

  3. Avoid accessing SIE register from the external CPU. For example: Route all the SIE interrupts to the software mailbox interrupt registers 0x144 and 0x148. This requires user to create download code.

- **FIX STATUS**

  Use workaround. No fix is currently planned for future silicon revisions. An implementation example is included in the Cypress Windows CE driver.

2. IDE register read when GPIO24 pin is low.

- **PROBLEM DEFINITION**

  Part does not service USB ISRs when GPIO24 pin (also labeled as HPI_INT and IORDY) is low and any IDE register is read.

- **PARAMETERS AFFECTED**

  USB ISRs do not get serviced.

- **TRIGGER CONDITIONS**

  The IDE registers (0xC050 through 0xC06E) should not be read unless IDE is being used. Debuggers that read all memory locations while single stepping can cause this situation to manifest itself.

- **SCOPE OF IMPACT**

  If debugging and using this pin, your application will appear to hang.

- **WORKAROUND**

  When running in stand-alone mode, avoid using the GPIO24 pin if possible.

- **FIX STATUS**

  Other workarounds being investigated. No fix is currently planned for future silicon revisions.

3. UART Does Not Recognize Framing Errors

- **PROBLEM DEFINITION**

  The UART is not designed to recognize framing errors.

- **PARAMETERS AFFECTED**

  UART serial communications.

- **TRIGGER CONDITIONS**

  Some platforms can cause EZ-OTG to see a string of NULL characters and cause the UART to get out of sync.

- **SCOPE OF IMPACT**

  This can cause the UART to lose connection with the host during serial communications. One example of this is if the UART is used as the debug port to the PC. This problem has occurred on, but is not limited to, Dell machines running WinXP or Win2k.

- **WORKAROUND**

  For general use, there is no workaround. If this problem is experienced while debugging, try running the debugger on a different Host (PC/OS). Otherwise the USB port can be used for the debugging interface.

- **FIX STATUS**

  No fix is currently planned for future silicon versions.

4. UART does not override GPIO Control Register

- **PROBLEM DEFINITION**

  When the UART is enabled, the GPIO Control Register still has control over GPIO 6 (UART RX pin). When enabled, the UART should override the GPIO Control Register, which defaults to setting the pin as an input.

- **PARAMETERS AFFECTED**

  UART serial communications.

- **TRIGGER CONDITIONS**

  Enabling UART

- **SCOPE OF IMPACT**

  GPIO 6 UART RX pin is controlled by GPIO Control Register and defaults to an input. The UART mode does not override the GPIO control register for this pin and can be inadvertently configured as an output.

- **WORKAROUND**

  Ensure the GPIO Control Register is written appropriately to set GPIO6 as an input when the UART is enabled.

- **FIX STATUS**

  No fix is currently planned for future silicon versions.

5. VBUS Interrupt (VBUS Valid) Requires Debouncing

- **PROBLEM DEFINITION**

  The VBUS interrupt in the Host/Device Status Registers [0xC090 and 0xC0B0] and OTG Control Register [0xC098] triggers multiple times whenever VBUS is turned on. It should only trigger once when VBUS rises above 4.4V and once when VBUS falls from above 4.4V to 0V.

- **PARAMETERS AFFECTED**

  Electrical.

- **TRIGGER CONDITIONS**

  VBUS turned on.

- **SCOPE OF IMPACT**

  Host/Device Registers and OTG Control Register trigger multiple times.

- **WORKAROUND**

  When reading the status of this interrupt, a software debounce should be implemented.

- **FIX STATUS**

  No fix is currently planned for future silicon versions.

6. Coupled SIE Interrupt Enable Bits

- **PROBLEM DEFINITION**

  Host/Device 1 SIE events will still trigger an interrupt when only the Host/Device 2 SIE Interrupt Enable is set and vise versa.

- **PARAMETERS AFFECTED**

  Host/Device SIE Interrupts.

- **TRIGGER CONDITIONS**

  Setting only 1 Host/Device SIE Interrupt Enable.

- **SCOPE OF IMPACT**

  The Host/Device global Interrupt Enable bits can not be used to disable each Host/Device SIE independently. These bits are found in the Interrupt Enable Register (0xC00E).

- **WORKAROUND**

  If an SIE Interrupt is desired, both Host/Device 1 and Host/Device 2 Interrupt Enable bits should be set in the global Interrupt Enable Register (0xC00E). To properly mask an SIE Interrupt to a single SIE, the lower level Host/Device Interrupt Enable Registers (0xC08C and 0xC0AC) must be used. For example, setting the

Host/Device 2 IE Register to 0x0000 will prevent any Host/Device 2 events from generating a Host/Device Interrupt. To disable all SIE interrupts, both Host/Device Interrupt Enable bits in the Interrupt Enable Register should be cleared.

- **FIX STATUS**

  No fix is currently planned for future silicon versions. Examples provided in the Development Kit Software.

7. Un-Initialized SIExmsg Registers

- **PROBLEM DEFINITION**

  The SIE1msg and SIE2msg Registers [0x0144 and 0x0148] are not initialized at power up.

- **PARAMETERS AFFECTED**

  HPI interrupts.

- **TRIGGER CONDITIONS**

  Power up initialization.

- **SCOPE OF IMPACT**

  If using the HPI interface in coprocessor mode, random data will be written to the SIE1msg and SIE2msg Registers [0x0144 and 0x0148] at power up. This will cause two improper HPI interrupts (HPI_INTR) to occur, one for each of the two SIExmsg Registers.

- **WORKAROUND**

  The external processor should clear the SIExmsg Registers [0x0144 and 0x0148] shortly after nRESET is de-asserted and prior to the expected processing of proper HPI interrupts (generally 10ms after nRESET is de-asserted.)

- **FIX STATUS**

  No fix is currently planned for future silicon versions.

8. BIOS USB Peripheral Mode: Descriptor Length

- **PROBLEM DEFINITION**

  The BIOS will not properly return a descriptor or set of descriptors if the length is a multiple of the control endpoint's maximum packet size.

- **PARAMETERS AFFECTED**

  Control Endpoint maximum packet size.

- **TRIGGER CONDITIONS**

  Get Descriptor requests.

- **SCOPE OF IMPACT**

  If the descriptor length is a multiple of the maximum packet size, the BIOS will respond with a STALL instead of a zero-length data packet for the final IN request.

- **WORKAROUND**

  If the requested descriptor length is a multiple of the maximum packet size, then either the maximum packet size or the descriptor length needs to change. A descriptor length can be increased by simply adding a padded byte to the end of a descriptor and increasing the descriptor Length byte by one. Section 9.5 (Descriptor) of the USB2.0 specification allows a descriptor length to be larger than the value defined in the specification.

- **FIX STATUS**

  No fix is currently planned for future silicon versions.

9. Peripheral short packet issue

- **PROBLEM DEFINITION**

  When a SIE is configured as a peripheral, the SUSBx_RECEIVE function does not invoke the callback function when it receives a short packet.

- **PARAMETERS AFFECTED**

  SIEx Endpoint x Interrupt (Interrupt 32-47).

- **TRIGGER CONDITIONS**

  This issue is seen when a SIE is configured as a peripheral during an OUT data transfer when the host sends a zero length or short packet. If this occurs the BIOS will behave as if a full packet was received and will continue to accept data until the Device n Endpoint n Count Register value is satisfied.

- **SCOPE OF IMPACT**
  All peripheral functions are susceptible to this as it is a normal occurrence with USB traffic.
- **WORKAROUND**
  To fix this problem the SIEx Endpoint x Interrupt must be replaced for any peripheral endpoint that is configured as an OUT endpoint.
  1. Acquire the file called susb1.s from Cypress Support or by downloading a newer version of the frameworks that has had this fix applied and includes susb1.s.
  2. Modify fwxcfg.h in your project to have the following flags and define/undef the fix for the endpoints you are using:

```
#define FIX_USB1_EP1
#define FIX_USB1_EP2
#undef  FIX_USB1_EP3
#undef  FIX_USB1_EP4
#undef  FIX_USB1_EP5
#undef  FIX_USB1_EP6
#undef  FIX_USB1_EP7

#undef  FIX_USB2_EP1
#undef  FIX_USB2_EP2
#undef  FIX_USB2_EP3
#undef  FIX_USB2_EP4
#undef  FIX_USB2_EP5
#undef  FIX_USB2_EP6
#undef  FIX_USB2_EP7
```

  3. Add the new susb1.s to the included assembly source files in the make file.

     For example : ASM_SRC := startup.s isrs.s susb1.s

  4. Add usb_init somewhere in the startup code. This will likely be in fwxmain.c as demonstrated below:

```
void fwx_program_init(void)
{
void usb_init();          /* define the prototype */
usb_init();
fwx_init();               /* Initialize everything in the base framework. */
}
```

  5. Build the project using the modified make file.
- **FIX STATUS**
  The ROM version of the BIOS will be updated during any future silicon rolls. No current roll of the part currently exists.

10. Data toggle corruption issue

- **PROBLEM DEFINITION**
  When a SIE is configured as a peripheral, data toggle corruption as specified in the USB 2.0 spec, section 8.6.4, does not work as specified.
- **PARAMETERS AFFECTED**
  SIEx Endpoint x Interrupt (Interrupt 32-47).
- **TRIGGER CONDITIONS**
  This issue is seen when a SIE is configured as a peripheral and the host sends an incorrect data toggle. According to the USB specification, when an incorrect data toggle is seen from the host the peripheral should

throw away the data but increment the data toggle bit to re-synchronize the data toggle bits. In the current ROM BIOS the SIEx Endpoint x Interrupt will ignore the data toggle error and accept the data.

- **SCOPE OF IMPACT**

  All peripheral functions are susceptible to this as it is a normal occurrence with USB traffic.

- **WORKAROUND**

  To fix this problem the SIEx Endpoint x Interrupt must be replaced for any endpoint that is configured as an OUT endpoint.

  1. Acquire the file called susb1.s from Cypress Support or by downloading a newer version of the frameworks that has this included.
  2. Modify fwxcfg.h in your project to have the following flags and define/undef the fix for the endpoints you are using:

```
#define FIX_USB1_EP1
#define FIX_USB1_EP2
#undef  FIX_USB1_EP3
#undef  FIX_USB1_EP4
#undef  FIX_USB1_EP5
#undef  FIX_USB1_EP6
#undef  FIX_USB1_EP7

#undef  FIX_USB2_EP1
#undef  FIX_USB2_EP2
#undef  FIX_USB2_EP3
#undef  FIX_USB2_EP4
#undef  FIX_USB2_EP5
#undef  FIX_USB2_EP6
#undef  FIX_USB2_EP7
```

  3. Add the new susb1.s to the included assembly source files in the make file.
     For example : ASM_SRC := startup.s isrs.s susb1.s
  4. Add usb_init somewhere in the startup code. This will likely be in fwxmain.c as demonstrated below:

```
void fwx_program_init(void)
{
void usb_init(); /* define the prototype */

usb_init();

fwx_init();    /* Initialize everything in the base framework. */
}
```

  5. Build the project using the modified make file.

- **FIX STATUS**

  The ROM version of the BIOS will be updated during any future silicon rolls. No current roll of the part currently exists.

References

1 Document # 38-08014, CY7C67200 EZ-OTG™Programmable USB On-The-Go Host/Peripheral Controller

Document History Page

Document Title: CY7C67200 EZ-OTG™ Programmable USB on-The-Go Host/Peripheral Controller Errata Document Number: 38-17021				
REV.	**ECN NO.**	**Issue Date**	**Orig. of Change**	**Description of Change**
**	227148	See ECN	bha	Initial release
*A	237769	See ECN	ari	Adding items 9 and 10