8x930H*x* (8x930HD/HE & 8x930HF/HG) SPECIFICATION UPDATE

Release Date: February, 1999
Order Number: 272962-013

Notice: The 8x930H*x* may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

Contents

REVISION HISTORY 1

PREFACE 2

SUMMARY TABLE OF CHANGES 4

IDENTIFICATION INFORMATION 8

ERRATA 10

SPECIFICATION CHANGES 24

SPECIFICATION CLARIFICATIONS 27

DOCUMENTATION CHANGES 32

REVISION HISTORY

Rev. Date	Version	Description
10/09/96	001	This is the new Specification Update document.
11/13/96	002	Added errata numbers 9611001, 9611002, and 9611003. Added A-1 stepping information.
12/10/96	003	Deleted Erratum number 9610001. This erratum does not exist in the 8x930H*x* A1 stepping. Added workaround to erratum number 9611002. Added erratum 9612001. Added specification clarification 002. Added documentation changes 001, 002, 003, 004, 005, and 006.
1/8/97	004	Added erratum 9701001. Deleted specification clarification 002. It does not apply to the 8x930H*x*. Added documentation changes 007 and 008. Added information for stepping A3.
2/5/97	005	Added documentation changes 009, 010, 011, and 012.
3/4/97	006	Added erratum number 9702001. Added specification changes for AC and DC characteristics. Added 64-pin SDIP package marking information.
4/15/97	007	Added documentation changes 013, 014, and 015.
5/30/97	008	Added 8x930HF0/HG0 stepping information. Added errata numbers 9705001, 9705002 and 9705003. Added specification clarifications 002, 003, and 004.
7/10/97	009	Added firmware workaround to erratum number 9705003. Added erratum 9706001.
11/4/97	010	Added errata numbers 9711001, 9711002, and 9711003.
4/7/98	011	Added erratum 9804001. Added Vcc and ECAP voltage range to Specification Clarification 002.
6/1/98	012	Added Specification Changes 003, 004 and 005. Added Documentation Change number 016.
2/5/99	013	Added Workaround #2 to erratum 9612001.

PREFACE

As of July, 1996, Intel has consolidated available historical device and documentation errata into this document type called the Specification Update. We have endeavored to include all documented errata in the consolidation process, however, we make no representations or warranties concerning the completeness of the Specification Update.

This document is an update to the specifications contained in the Affected Documents/Related Documents table below. This document is a compilation of device and documentation errata, specification clarifications and changes. It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools.

Information types defined in Nomenclature are consolidated into the specification update and are no longer published in other documents.

This document may also contain information that was not previously published.

Affected Documents/Related Documents

Title	Order
8X930Hx Universal Serial Bus Peripheral Controller Data Sheet	272928-003
8X930Ax, 8X930Hx Universal Serial Bus Microcontroller User's Manual	272949-001

Nomenclature

Errata are design defects or errors. These may cause the published (component, board, system) behavior to deviate from published specifications. Hardware and software designed to be used with any component, board, and system must consider all errata documented.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in any new release of the specification.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in any new release of the specification.

Documentation Changes include typos, errors, or omissions from the current published specifications. These will be incorporated in any new release of the specification.

NOTE:

Errata remain in the specification update throughout the product's lifecycle, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the specification update are archived and available upon request. Specification changes, specification clarifications and documentation changes are removed from the specification update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

SUMMARY TABLE OF CHANGES

The following table indicates the errata, specification changes, specification clarifications, or documentation changes which apply to the 8x930H*x* product. Intel may fix some of the errata in a future stepping of the component, and account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

Codes Used in Summary Table

Steps

X:	Errata exists in the stepping indicated. Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Page

(Page):	Page location of item in this document.

Status

Doc:	Document change or update will be implemented.
Fix:	This erratum is intended to be fixed in a future step of the component.
Fixed:	This erratum has been previously fixed.
NoFix:	There are no plans to fix this erratum.
Eval:	Plans to fix this erratum are under evaluation.

Row

▌	Change bar to left of table row indicates this erratum is either new or modified from the previous version of the document.

Errata

Number	Steppings				Page	Status	ERRATA
	HD1/ HE1	HD3/ HE3	HF0/ HG0	HF2/ HG2			
9611001	X	X			10	Fixed	Non-Stop NoAcknowledging (NAK) with OHCI Systems
9611002	X				10	Fixed	Hub May Accidentally Be Recognized as a Low-speed Device
9611003	X	X	X	X	12	Fix	V_{OH} on Port 1, 2, and 3 are Below Target Specification
9612001	X				12	Fixed	Receive FIFO RXFFRC Error
9701001	X				15	Fixed	Downstream J-K Signal Duty Cycle Not Symmetrical
9702001	X	X			15	Fixed	Transmit FIFO Underrun
9705001	X	X			17	Fixed	Low Speed End of Packet (EOP) Errata
9705002	X	X	X		17	Fixed	Serial Port Auto Address Recognition Errata
9705003	X	X	X		18	Fixed	Isochronous Transfer RXCNT Errata
9706001	X	X	X		20	Fixed	Timer 2 Idle Mode Wake Up Errata
9711001	X	X	X		21	Fixed	Incorrect Response On Receive FIFO Overflow in Low Clock Mode
9711002	X	X	X	X	21	Fix	Timer and Interrupt Flags are Not Set When the TCON SFR is Modified Simultaneously with an External Interrupt
9711003	X	X	X	X	22	Fix	PCA Capture Flag is Not Set when a Capture Occurs with PCA Timer Overflow
9804001	X	X	X	X	22	Fix	Interrupts of 3 or More Priority Levels Occurring at the Same Time

Specification Changes (Sheet 1 of 2)

Number	Steppings				Page	Status	SPECIFICATION CHANGES
	HD1/ HE1	HD3/ HE3	HF0/ HG0	HF2/ HG2			

Specification Changes (Sheet 2 of 2)

001	X	X			24	Doc	Revise datasheet status from “Product Preview” to “Advance Information”
002	X	X			24	Doc	Added 8x930HF0/HG0 Stepping Information to Datasheet
003	X	X	X	X	26	Doc	Series Resistor Requirement for Impedance Matching Changed to 22 Ohms (no longer 27 – 33 ohms)
004	X	X	X		26	Doc	Added 8x930HF2/HG2 Stepping Information to Datasheet
005				X	26	Doc	ECAP Output Voltage Supply Does Not Vary With Vcc

Specification Clarifications

Number	Steppings				Page	Status	SPECIFICATION CLARIFICATIONS
	HD1/ HE1	HD3/ HE3	HF0/ HG0	HF2/ HG2			
001	X	X	X		27	Doc	T_{RHDZ1} Timing
002	X	X	X		30	Doc	ECAP Pin Usage to Supply 3.0 V to 3.6 V Voltage for 1.5K Ohm USB Pullup Terminator
003	X	X	X		31	Doc	TXCNTx Must Be Written With Correct Byte Count
004	X	X	X		31	Doc	Unused D_{PX} and D_{MX} Pins Must Be Pulled Down
005	X	X	X		13	Doc	Receive FIFO RXFFRC Error

Documentation Changes

Number	Document Revision	Page	Status	DOCUMENTATION CHANGES
001	001	32	Doc	Nonvolatile Memory Verification Port Labeled Incorrectly
002	001	32	Doc	Incorrect Address Given for TXSTAT SFR
003	001	32	Doc	Incorrect Signature Byte
004	001	32	Doc	Power-on Reset Capacitor Value Changed
005	001	32	Doc	SCON SFR's REN Bit Description Incorrect
006	001	32	Doc	Extraneous Footnote in RXCON SFR
007	001	33	Doc	Power Off Flag Voltage Values
008	001	33	Doc	W_{CLK} Description Incorrect
009	001	33	Doc	Configuration Byte Misspelled
010	001	33	Doc	RTWCE Description Inaccurate
011	001	33	Doc	RL Instruction Misspelled
012	001	33	Doc	Footnote Incorrect in Data Instructions Table
013	001	34	Doc	Hardware, Not Firmware, Sets FE Bit In SCON Register
014	001	34	Doc	Reset Value of LC Bit (PCON.5) Should Be '1'
015	001	34	Doc	Incorrect OVRI# Signal Description
016	001	34	Doc	Remove SDIP Package Reference Throughout the Datasheet

IDENTIFICATION INFORMATION

Markings

Product	Part Number	Stepping	Marking	Package	Comment
8x930Hx Step 1	N80930HD1	HD1/ HE1	no marking	PLCC	Shipping media = tubes
			Q 831	PLCC	Customer sample No ROM
			SL24D	PLCC	General production, ROMless Shipping media = tape & reel
	N83930HD1		R xxxx	PLCC	8K ROM Customer ROM Code
	N83930HE1		R xxxx	PLCC	16K ROM Customer ROM Code
8x930Hx Step 3	N80930HD3	HD3/ HE3	Q 807	PLCC	General customer sample, ROMless
	N80930HD3		SL26K	PLCC	General production, ROMless, Shipping media = tape & reel
	N80930HD3		no marking	PLCC	General production, ROMless, Shipping media = tubes
	N83930HD3		R xxx	PLCC	General production, 8K ROM
	N83930HE3		R xxx	PLCC	General production, 16K ROM

Markings (Continued)

Product	Part Number	Stepping	Marking	Package	Comment
8x930Hx Step 0	N80930HF0	HF0/ HG0	Q871	68ld PLCC	General customer sample, ROMless
	N8030HF0		SL27F	68ld PLCC	General Production, ROMless, Shipping Media = tape'n reel
	N80930HF0		no marking	68ld PLCC	General Production, ROMless, Shipping media = tubes
	N80930HF0		R xxx	68ld PLCC	General Production, 8K ROM
	N80930HG0		R xxx	68ld PLCC	General Production, 16K ROM
8x930Hx Step 2	N80930HF2	HF2/ HG2	Q871	68ld PLCC	General customer sample, ROMless
	N8030HF2		SL27F	68ld PLCC	General Production, ROMless, Shipping Media = tape'n reel
	N80930HF2		no marking	68ld PLCC	General Production, ROMless, Shipping media = tubes
	N80930HF2		R xxx	68ld PLCC	General Production, 8K ROM
	N80930HG2		R xxx	68ld PLCC	General Production, 16K ROM

ERRATA

9611001. Non-Stop NoAcknowledging (NAK) with OHCI Systems

PROBLEM: While in low-clock mode (LC Bit = 1), clearing the RXSETUP bit while the Transmit FIFO data register is “transmit ready” (transmit ready = data in the Transmit FIFO data register and a byte-count in the TXCNT register) could cause the part to continuously NAK in response to an IN token. This will occur until the Transmit FIFO data register is reset and reloaded. This only happens with OHCI where the bus turnaround is very fast.

IMPLICATION: If this condition occurs, the Serial Bus Interface Engine will continue to NAK even though there is data in the Transmit FIFO data register.

WORKAROUND: Software workaround is available. The user needs to make sure that the RXSETUP bit is cleared before the Transmit FIFO register becomes “transmit ready”.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9611002. Hub May Accidentally Be Recognized as a Low-speed Device

PROBLEM: When a self-powered, suspended hub is reattached to the host, it could be accidentally recognized as a low-speed device. This is caused by the 8x930H*x* A-1 driving K signaling upstream and the host incorrectly sampling the speed of the device immediately after connect instead of after USB reset.

IMPLICATION: Enumeration will fail due to the host sending low-speed packets to a high-speed device.

WORKAROUND: To fix this problem, you must modify the reset circuitry on current 8x930H*x* designs. Figure 1 depicts two circuits, one for a bus-powered hub and another for a self-powered hub. For bus powered devices, the circuit is the same one that is given in Figure 14-1 on page 14-1 of the *8X930Ax, 8X930Hx Universal Serial Bus Microcontroller User’s Manual*, except the capacitor value has been changed from 1.0 µF to 0.3 µF.

For self-powered devices there are two requirements:

- The reset circuit design needs to be activated when the board is first powered-on.
- If the board is already powered-on, then a connection to the host should initiate reset. By connecting another capacitor to V_{BUS} (C2, as shown in Figure 1), a reset pulse is sent to the chip by either a power-up or by a USB connect.

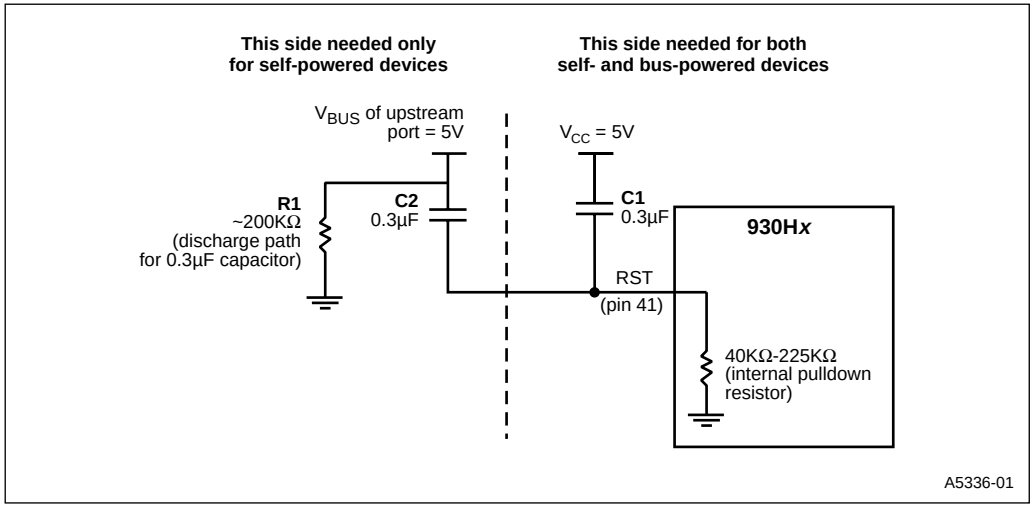

Figure 1. RST Source From V_{CC} and V_{BUS}

The circuit shown in Figure 1 uses the following components:

- C1 — Used in conjunction with the internal pulldown resistor to generate a 20 ms pulse of reset. For bus-powered devices, V_{CC} = V_{BUS}. This creates a 20 ms pulse upon connection to the host. For self-powered devices, a 20-ms pulse will occur regardless of the state of V_{BUS} if the capacitor ratios remain 1:1. The circuit operation and timing relationships rely on C1 = C2.
- C2 — Used only for self-powered devices. The capacitor delivers a 20-ms pulse to the RST pin when the V_{BUS} signal is detected upon device connection.
- R1 — Provides a discharge path for C2. This resistor is mandatory. If R1 is not present, C2 will not discharge the reset pulse and therefore will not allow a proper chip reset during device connection.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9611003. V_{OH} on Port 1, 2, and 3 are Below Target Specification

PROBLEM: When Port 1, 2, and 3 are in quasi bidirectional mode, their V_{OH}s are below the target specification as shown:

$V_{OH} = V_{CC}$ - 0.8 V (instead of V_{CC} - 0.3 V) @10 µA
$V_{OH} = V_{CC}$ - 1.7 V (instead of V_{CC} - 0.7 V) @30 µA
$V_{OH} = V_{CC}$ - 2.7 V (instead of V_{CC} - 1.5 V) @60 µA

IMPLICATION: The fanout of port 1, 2, and 3 pins are reduced.

WORKAROUND: External buffers can be used to provide the required drive capability needed for interfaces that require more drive than the 8x930H*x* can support.

STATUS: Fix. Refer to Summary Table of Changes for affected steppings.

9612001. Receive FIFO RXFFRC Error

PROBLEM: After the RXFFRC bit of the RXCON register is set, the RXFIF1:0 bits in the RXFLG register remain "11". According to specifications, RXFIF1:0 should immediately decrement when RXFFRC is set. This problem occurs when the following conditions are simultaneously met:

1. The function interface unit (FIU) and serial bus interface engine (SIE) write a byte count to the RXCNTL SFR of endpoint 1's receive FIFO.
2. The CPU sets the RXFFRC bit for any other endpoint's receive FIFO (not endpoint 1).
3. The receive FIFO for the endpoint in (2) has RXFIF1:0 bits = "11".

This problem is most likely to occur in low-clock mode when the device has a high data receive rate (bulk mode) on endpoint 1 and one or more of the other receive FIFO endpoints.

IMPLICATION: When the problem occurs, having RXFIF1:0 remain as "11" will cause firmware to incorrectly assume that there are two packets left in the receive FIFO (in dual packet mode), when in reality there is only one packet left. If firmware attempts to read the non-existent second packet, hardware will set the RXURF bit. When the RXURF bit gets set, the 8x930H*x* continues to NAK all OUT packets on the affected FIFO. At this point, the FIFO will be in an unknown state, requiring firmware to reset/clear that FIFO.

WORKAROUND: #1 (if you experience 1/3 data loss with this workaround, try workaround #2) Additional code must be added to the firmware location(s) where a non-endpoint 1 receive FIFO is released. This code must determine if the RXFIF1:0 bits are "11" before and after setting the RXFFRC bit. If this is true, and if the RXSEQ bit has not been toggled by hardware during this time, then the error has occurred. At this point,

firmware must attempt to release the receive FIFO again by re-setting the RXFFRC bit. This process must be repeated until the receive FIFO is successfully released.

Insert a firmware routine similar to the example shown in Figure in your code at the point where you release the receive FIFO for all endpoints except endpoint 1. The code must be duplicated for each endpoint (except endpoint 1), replacing the “x” in the example’s code labels with the endpoint number.

```
;*****************************************************
;
; RXFFRC Firmware Workaround
;*****************************************************
;
; Note: Registers 11 through 14 are utilized in this example.
; Please be sure to save and restore these registers as needed.

RELEASE_FIFO_x:
            mov     A,      RXFLG
            mov     R12,    RXFLG         ; before
            mov     R13,    RXSTAT

            setb    RXFFRC

            ; if (RXFLG_BEFORE = RXFLG_AFTER)
            ;  then { continue and check if RXFIF="11" }
            ;  else { setting RXFFRC was successful }
            cjne    A,    RXFLG, REL_FIFOx_OK

            ; if we get here R11 has RXFLG before and after - no
            ; change
            mov     R14,    RXSTAT

            ; if (RXFLG_AFTER = "11")
            ;  then { continue and check RXSEQ data toggle }
            ;  else { jump to REL_FIFOx_OK}
            anl     R11, #11000000b         ; check RXFIF after
            cjne    A,   #11000000b,  REL_FIFOx_OK

            ; RXFIF bits are "11"    RXFLG="C0"
            ; if (RXSEQ_BEFORE = RXSEQ_AFTER)
            ;  then
            ;   { no data toggle - set RXFFRC again }
            ;  else
            ;   { data toggle - OK jump to REL_FIFOx_OK}
            anl     R13, #10000000b
            anl     R14, #10000000b
            cmp     R13,   R14
            jne     REL_FIFOx_OK
```

Figure 2. RXFFRC Firmware Workaround

```
            ; FIFO errata condition: RXFIF was "11" before & after; &
            ; RXSTAT didn't change
RELEASE_FIFO_x_AGAIN:
            ljmp    RELEASE_FIFO_x

REL_FIFOx_OK:
```

Figure 2. RXFFRC Firmware Workaround (Continued)

WORKAROUND: #2 Make sure that if workaround #1 was attempted, that you backout of the workaround. Then, set RXFFRC as usual but ignore RXCNT entirely. Use a read routine of RXDAT that reads it until RXURF is set and then clear RXURF. An example is provided below that can be a subroutine from within the SOF or can be put in-line as well. The following example is a code suggestion only and responsibility of verification of the workaround lies with the customer as each implementation and application will be different. Also note that this workaround may cause issues with performance sensitive applications and some fine tuning or simliar algorithms may need to be engineered.

```
READ_RXDATA_ROUTINE:
; push registers here as required by your application

moredata:
; read RXDAT in scratch buffers
mov R3, RXDAT
mov R2, RXDAT

; if RXURF then we've read past the end of the FIFO
jb RXURF, nomoredata

; good data, move to R5 and R4 (or wherever you want the data)
mov R5, R3
mov R4, R2

; try to read more data
sjmp moredata

nomoredata:
clr RXURF
; with this setup you will always read until RXURF so it always needs to be
; cleared every time

; now pop whatever registers you pushed and return

ret
```

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9701001 Downstream J-K Signal Duty Cycle Not Symmetrical

PROBLEM: Customers cannot cascade 8x930H*x* hubs up to the maximum allowable five tiers as specified by the *Universal Serial Bus Specification*, version 1.0.

This problem occurs because the J-K duty cycle of the 8x930H*x* USB differential signal is not symmetrical. Therefore, the signaling marginality worsens when more 8x930H*x* hubs are cascaded. Timeouts may occur on the host and/or on the device.

If an end function device using an A3-stepping 8*x*930A*x* is connected to the 8x930H*x* hubs, it works well only up to four tiers of 8x930H*x* hubs. If an end function device using an A2-stepping 8*x*930A*x* is connected to the 8x930H*x* hubs, it works will only up to two tiers of 8x930H*x* hubs.

if an end function device using an A3-stepping 8*x*930A*x* (or any other USB device with a symmetrical duty cycle for the J-K signaling) is connected to the hubs, it works well up to four tiers of 8x930H*x* cascaded with another tier using a hub with a symmetrical duty cycle for the J-K signaling.

The maximum allowable cascading of hubs specified by the *Universal Serial Bus Specification*, Rev. 1.0, is:

Host/root hub-->1st tier hub-->2nd tier hub-->3rd tier hub-->4th tier hub-->5th tier hub-->end function device.

IMPLICATION: If 8x930H*x* hubs are cascaded more than the number of tiers described above, the USB device attached to the last tier hub will not function correctly.

WORKAROUND: No workaround.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9702001. Transmit FIFO Underrun

PROBLEM: When setting the TXCLR coincides with an IN token received, the Transmit FIFO (TXFIFO) will be underrun. This will happen when all the following conditions are met:

A. The RXSETUP bit is set
B. An IN token is received
C. Setting the TXCLR bit

IMPLICATION: When the problem occurs, the TxFIFO will be in underrun error condition (TXURF bit set). If the TXURF bit is not checked by the firmware and is not cleared properly, the 8x930H*x* will continue to NAK all the IN tokens on the affected TXFIFO.

WORKAROUND: Additional code must be added to the firmware wherever it is applicable. This code must determine if the RXSETUP bit is set, then set the TXCLR bit twice or more if necessary to make sure the TXFIFO is cleared. When the TXFIFO is successfully cleared, all the flags (especially TXFIFO underrun flag, TXURF) in the TXFLG register are cleared except for the empty flag (TXEMP) which will be set.

Insert a firmware routine similar to the example shown in Figure 3 in your code wherever it is necessary. The code must be duplicated for each transmit endpoint, replacing the “x” in the example code labels and the EPINDEX register with the proper endpoint number for hub or for embedded function.

STATUS: Fixed. Refer to Summary Table of Change for affected steppings.

```
Clear_TXFIFO_xx:

     mov EPINDEX, #x0h        ; hub or embedded function endpoint 0
     mov A, RXSTAT
     jb ACC.6 Clear_Again_xx  ; jump if RXSETUP bit is set
     mov EPINDEX, #xxh        ; related hub or embedded function endpoint number
     orl TXCON, #10000000b    ; set TXCLR bit
     sjmp End_Clear_TXFIFO_xx:

   Clear_Again_xx:
     mov EPINDEX, #xxh        ;  related hub or embedded function endpoint number
     orl TXCON, #10000000b    ; set TXCLR bit
     mov A, TXFLG
     jb ACC.1, Clear_Again_xx ; jump if TXURF is set
     sjmp End_Clear_TXFIFO_xx

   End_Clear_TXFIFO_xx:
```

Figure 3. Transmit FIFO Underrun Firmware Workaround

9705001 Low Speed End of Packet (EOP) Errata

PROBLEM: Low speed devices connected to downstream ports of the 8x930HDx hub peripheral controller may lose connectivity after an undetermined period of time due to noise glitches on the USB input pins. These glitches on the input pins result in the 8x930HDx recognizing a valid signal transition during a EOP signal, causing the EOP signal to be detected incorrectly by the hub. This problem occurs when **all** of the following conditions occur simultaneously:

1. The glitch happens at the end of the EOP signal from the downstream low speed device.
2. The amplitude of the glitch is more than 100mV due to the hysteresis margin on the input pins.
3. The asynchronous glitch happens at a critical internal clock sampling node.
4. The glitch happens within 20ns of the signal reaching the rising trip threshold (~1V @ Vcc=5V).

IMPLICATION: When this problem occurs, the hub fails to properly detect a valid EOP from the downstream low speed device. The hub then detects a “babble” condition at the end of frame on the offending port and disables the port. The low speed device connected to that port will stop communicating with the host. The user will need to unplug and re-plug in the low speed device again to recover from this condition.

WORKAROUND: As with any design, it is very important to design the USB circuit board to suppress noise. Noise usually originates from the power plane or the USB cable. For the 8x930HDx, analysis has shown that if noise glitches are limited to less than 100mV on USB pins, this problem will not occur. Also note that if the 8x930HDx device is run in low clock mode (PCON.5=1), the possibility of this failure is dramatically reduced.

Customers should migrate their designs to the new 8x930HFx 4 external downstream port hub peripheral controller. A document titled ***Design Considerations when Migrating between Intel’s 8x930HD3 and 8x930HF0 USB Hub Controllers*** has been posted on the web ***at: developer.intel.com/design/usb/papers***.

STATUS: Fixed. Please refer to Summary Table of Changes for a list of affected steppings.

9705002 Serial Port Auto Address Recognition Errata

PROBLEM: When the 8x930Hx serial port is configured to use auto address recognition feature in multiprocessor communication by setting the SM2 bit in SCON register, the 8x930H*x* cannot automatically recognize the sent slave (given) address and broadcast address. This applies to serial port modes 1, 2, and 3.

IMPLICATION: When configured to use this feature, the serial port cannot recognize automatically the given address and broadcast address sent to it. As such, the serial port receive interrupt, RI flag in SCON register will not be set, and no interrupt will be generated. This errata is not applicable to Serial port mode 0 as it does not support this feature.

WORKAROUND: Manually check the address if multiprocessor communication, using serial port, is needed. To avoid enabling the auto address recognition feature, do not set the SM2 bit in SCON register. The RI flag is set and interrupt is generated with every serial data received. In the serial port receive interrupt service routine, check the RB8 bit in SCON register to determine if the serial data received is an address byte (RB8=1) or a data byte (RB8=0). If it is the address byte and the address matches with a pre-assigned address, then read the serial data bytes on the subsequent receive interrupts until next address byte is received. If the address byte does not match with the pre-assigned address, then ignore the subsequent serial data bytes received until next address byte is received.

STATUS: Fixed. Please refer to Summary Table of Changes for a list of affected steppings.

9705003 Isochronous Transfer RXCNT Errata

PROBLEM: When the 8x930H*x* is configured to use dual packet mode in isochronous transfer, the receive FIFO count register (RXCNTx) of the new data packet is corrupted if the ***read completion*** of the previous data packet (setting RXFFRC bit) coincides with ***receive done*** of the new data packet.

IMPLICATION: When this problem occurs, the 8x930H*x* will read an incorrect value from the RXCNTx register on the new data packet and hence read an incorrect number of bytes from the receive FIFO. With that, the data read will be either shorter or longer than actually received, the data read will not be correct, and the receive FIFO will overflow or underrun.

WORKAROUND: At the end of an isochronous data read in the start of frame (SOF) ISR, do **not** release the RXFIFO by setting the RXFFRC bit. In the very beginning of the next SOF interrupt service routine, before the next OUT token arrives, set the RXFFRC bit to release the RXFIFO that was read in the previous frame. This will prevent the ***read completion*** of the previous data packet (setting RXFFRC bit) from coinciding with the ***receive done*** of the new data packet. An example of a firmware workaround is shown in Figure STATUS:.

STATUS: Fixed. Please refer to Summary Table of Changes for a list of affected steppings.

```
;*****************************************************************************************************
; Isochronous Transfer Endpoint 1 Initialization Code
; > Use a direct address RAM location (eg. 30h) to store a value for setting or not setting RXFFRC bit:
; > [30h] = 0Ch ->not to set RXFFRC bit,
; > [30h] = 1Ch ->to set RXFFRC bit,
;*****************************************************************************************************
ISO_EP1_INIT:                                   ; example endpoint 1 initialization routine
   mov                                          ; do not set RXFFRC bit when entering SOF ISR for the
   30h, #0Ch                                    ;first time

   ;other Endpoint 1 initialization code
   ret
;*****************************************************************************************************
; START OF FRAME (SOF) Interrupt Service Routine
;  > RXFFRC bit is set to release the isochronous data packet that was read in previous frame
;*****************************************************************************************************

;
SOF_ISR:                                        ; example endpoint 1 isochronous transfer ISR
    push   EPINDEX
   mov    EPINDEX, #01h                         ; select iso endpoint 1
   mov    RXCON, 30h                            ; set or do not set RXFFRC bit depends on value ;in 30

   ; do not add additional instructions before this line; RXFFRC bit must be set before OUT token
   arrives
   push    PSW
   push    PSW1
   push    ACC
   push    R3
   clr     ASOF                                  ;clear SOF interrupt flag

;
EP1_RX_ISOC:                                    ;example ISO receive processing routine

   jb      RXFIF0, EP1_RX_DATA_AVAIL
   jb      RXFIF1, EP1_RX_DATA_AVAIL

   mov     30h, #0Ch                             ; no iso data packet received, don't
                                                 ;setRXFFRC bit in next frame
   ljmp    EXIT_SOF_ISR

;
EP1_RX_DATA_AVAIL:
   mov     30h, #1Ch                             ; to set RXFFRC bit in next frame as it is read in
                                                 this ;frame
   jnb     RXERR, EP1_NO_RX_ERROR
   ljmp    HANDLER_RX1_ERROR_X                   ; jump to iso receive error handling routine

(Continued)
```

Figure 4. Isochronous Transfer RXCNTx Errata Firmware Workaround Example

```
;*******************************************************************************************************
; Start of Frame Interrupt Service Routine, Continued
;*******************************************************************************************************

;
EP1_NO_RX_ERROR:
   mov    A,  RXCNTL                          ; get the receive count
   jz   EXIT_SOF_ISR
   mov    R3, A                               ; temp storage of RXCNTL

;
RX_Loop_x:
   mov     A,   RXDAT                         ; read the iso data received

  ; process the data here, such as save to memory etc
   djnz    R3, RX_LOOP_X

;
EXIT_SOF_ISR:                                 ; restore any registers used
  pop  R3
   pop    ACC
   pop    PSW1
   pop    PSW
   pop    EPINDEX
  reti
```

Figure 4. Isochronous Transfer RXCNTx Errata Firmware Workaround Example (continued)

9706001. Timer 2 Idle Mode Wake Up Errata

PROBLEM: When the 8x930H*x* is in idle mode, the interrupt generated by the Timer 2 is not able to wake up the CPU. The Timer 2 interrupt can be generated in various modes by timer or counter overflow, or by capturing an external high-to-low transition signal on T2EX pin.

IMPLICATION: When this problem occurs, the 8x930H*x* will remain in the idle mode. The CPU will not wake up, and the Timer 2 interrupt service routine will not execute.

Note that this erratum is not applicable to powerdown mode operation as all peripherals (including Timer 2) and the CPU stop running in powerdown mode. Powerdown mode is required when the 8x930H*x* is suspended by USB to meet the suspend current specification. Idle mode is not.

WORKAROUND: There is no workaround when using Timer 2 to wake up the 8x930H*x* CPU from idle mode. If Timer 2 service is needed, use another unused interrupt such as Timer 0 or 1, or External Interrupt 0 or 1. Synchronize it with Timer 2 to wake up the CPU in idle mode and then jump to Timer 2 service routine.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9711001 Incorrect Response On Receive FIFO Overflow in Low Clock Mode

PROBLEM: Intermittently, the device will respond incorrectly to an Out token from the host, which causes an RxFIFO overflow. Instead of timing out and setting the appropriate flags, the device will occasionally ACK the data, signalling a valid reception of data.

1. Device is running in Low Clock Mode (PCON.5 is set).
2. Host incorrectly sends one additional byte to the RxFIFO, that is, the total length sent has to be equal to 1 + physical maximum endpoint size. 17 bytes for Endpoint 0 and 5, 33 bytes for Endpoint 2, 3, and 4, and 1025 bytes for Endpoint 1.

The correct response is a USB timeout.The RxEMP and RxOVF bits in the RXFLG SFR should be set. The RxACK should be cleared and RxSEQ should remain unchanged.

IMPLICATION: The host will see an ACK instead of a timeout and the FIFO is in an unknown state. The host and the device will be out of sequence.

WORKAROUND: No workaround is available. Under normal operation, Endpoints 0 and 5 are configured as an 8-byte FIFO single/dual packet mode, Endpoints 2, 3, and 4 are configured as a 16-byte FIFO single/dual packet mode, and Endpoint 1 should be configured as a 512 byte dual-packet mode if used in ISOC transfer type.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9711002 Timer and Interrupt Flags are Not Set When the TCON SFR is Modified Simultaneously with an External Interrupt

PROBLEM: When the control bits (IT0, IT1, TR0 and TR1) in the TCON SFR are set or cleared at the same instance as an edge-triggered external interrupt(s) is generated, the IE0 and/or the IE1 flag(s) will not be set. Therefore, no external interrupt will be generated. For a level-sensitive triggered external interrupt, the interrupt flag will continue to be set until the external source is removed from the INT0# or INT1# pins.

Similarly, when the control bits (IT0, IT1, TR0, and TR1) in the TCON SFR are set or cleared at the same instance the Timer0 or Timer1 overflows, the IT0 and/or IT1 flag(s) will not be set. Therefore, no Timer0/Timer1 interrupt will be generated.

IMPLICATION: Application will not be able to detect that an external event has happened. As for the timer, it must wait for another 256/65535 machine cycles before another timer overflow interrupt is generated.

WORKAROUND: Do not modify the TCON control bits after the timer has been started or the external interrupt bits are enabled. If the control bits require modification when a timer is enabled, Timer2 or PCA timer can be used instead. If two timers are required, and the control bits require modification, a combination of Timer0 and Timer1 or Timer1 and

Timer2 can also be used for IT0 and IT1 to remain set if Timer2 control bits in T2CON SFR are modified. If external interrupts are required and the control bits in the TCON SFR require modifying, use a level-sensitive type triggered interrupt to set the IE0 and IE1 flags upon clearing/setting control bits in the TCON SFR.

STATUS: Fix. Refer to Summary Table of Changes for affected steppings.

9711003 PCA Capture Flag is Not Set when a Capture Occurs with PCA Timer Overflow

PROBLEM: When the PCA is used in capture mode, the capture flag (CCFx) in the CCON SFR will not be set upon a positive/negative edge signal on the CEXx pin. This only happens when the PCA timer overflows at exactly the same instance as the positive/negative edge signal on the CEXx pin.

IMPLICATION: The device may not detect a capture event.

WORKAROUND: The firmware can be coded so that the PCA timer never overflows, thus setting the PCA capture flag.

Another option would be to enable the PCA timer overflow (CF) in the CCON SFR. To do this, clear the CCAPxH and CCAPxL registers to values other than FF00h. Upon an overflow of the PCA timer, the firmware checks the CCAPxH and CCAPxL registers for any new values. If there is a value of FF00h in the SFR pair, CCAPxH/CCAPxL, a capture event has occurred. Clear both registers to values other than FFh and 00h, respectively, for the next capture event.

STATUS: Fix. Refer to Summary Table of Changes for affected steppings.

9804001 Interrupts of 3 or More Priority Levels Occurring at the Same Time

PROBLEM: When three interrupts of different priorities occur at approximately the same time (~40 nsec. window) on the 8x930 peripheral controllers, the lowest priority interrupt will not be serviced correctly. As the lower priority interrupt is interrupted by a higher priority interrupt, the internal interrupt pending bit remains set and the device believes it is still servicing the lowest priority interrupt. This disables any further lowest priority interrupts from being serviced.

Any interrupt source can be affected by this if it is the lowest priority of the three enabled interrupts. For example, if external interrupt 0 (INT0) is priority 3, start of frame (SOF) interrupt is priority 2, and serial port (SIO) interrupt is priority 0, and they all occur at the same time, then the SIO interrupt would stop being serviced.

If four different interrupt priority levels are used, then both priority 0 and 1 interrupts would stop being serviced.

IMPLICATION: This will cause the lowest priority interrupt to no longer be serviced and the interrupt will not recover unless a device reset occurs.

WORKAROUND 1: To insure this interrupt priority conflict does not occur, make sure only two interrupt priority levels are enabled at any given time.

WORKAROUND 2: Once the device is in the failing condition and the lower priority interrupt(s) are no longer being serviced, there is a possible software workaround for some applications. This workaround involves detecting the fail event, manually pushing a return address on the stack, and executing a RETI instruction. The user will have to evaluate if this workaround will apply to their application.

Example:

```
main_code:
      ljmp int_fail_fix
org FF1500H
      Continue main code
      .
      .
      .
int_fail_fix:
      push #00H
      push #15H
      reti
```

How it Works:

The users main program code (main_code) detects the failing condition. One method of doing this would be reading the appropriate interrupt pending bit and starting a timer when this bit was set. If the interrupt is not serviced in a specified amount of time, the program would jump to the int_fail_fix routine. The int_fail_fix routine pushes the address of the instruction after the ljmp to int_fail_fix,1500 in this case.The RETI instruction will return the program counter to 1500. This will pop the internal interrupt stack so that it will handle future lowest priority interrupt(s).

STATUS: Fix. Refer to Summary Table of Changes for affected steppings.

SPECIFICATION CHANGES

001. Revise datasheet status from "Product Preview" to "Advance Information"

The *8X930Hx Universal Serial Bus Peripheral Controller* data sheet, Product Preview version (order number 272928-001) has been revised to Advance Information datasheet (order number 272928-002). Refer to the SUMMARY TABLE OF CHANGES for affected steppings.

The changed specifications are listed in the following tables:

Table 1. DC Characteristics

Symbol	Parameter	Min	Typical (1)	Max	Units	Test Conditions
I_{PD}	Powerdown Current Normal powerdown USB suspend		 25 145	 75 (was 50) 175	µA	
I_{DL}	Idle Mode I_{CC}			60 (was 40)	mA	Full speed (in low clock mode) PLLSEL2:0 = 110 F_{CLK} = 3 MHz
				110 (was 100)		Full speed (not in low clock mode) PLLSEL2:0 = 110 F_{CLK} = 12 MHz
I_{CC}	Active I_{CC}			75 (was 70)	mA	Full speed (in low clock mode) PLLSEL2:0 = 110 F_{CLK} = 3 MHz
				170		Full speed (not in low clock mode) PLLSEL2:0 = 110 F_{CLK} = 12 MHz

NOTE:
1. Typical values are obtained using V_{CC} = 5.0 V, T_A = 25° C and are not guaranteed.

Table 2. AC Characteristics

Symbol	Parameter	CPU Frequency @ 12 MHz (M, N = 0)	CPU Frequency (F_{CLK}) Variable		Units
			Min	Max	
T_{AVLL}	Address Valid to ALE Low	21.66 (was 28.66)	(0.5+M)T_{CLK} – 20 (was -13)		ns
T_{LLAX}	Address Hold after ALE Low	4 (was 10)	4 (was 10)		ns
T_{WLWH}	WR# Pulse Width	71.33 (was 73.33)	(1+N)T_{CLK} – 12 (was -10)		ns
T_{LLRL}	ALE Low to RD# or PSEN# Low	4 (was 5)	4 (was 5)		ns
T_{LHAX}	ALE High to Address Hold	40.33 (was 56.33)	(1+M)T_{CLK} – 43 (was -27)		ns
T_{RHDZ2}	Data Float After RD# or PSEN# High	83.33 (was 93.33)		T_{CLK} (was + 10)	ns
T_{RHLH2}	RD# or PSEN# High to ALE High (data)	83.33 (was 93.33)	T_{CLK} (was + 10)		ns
T_{WHLH}	WR# High to ALE High	88.33 (was 93.33)	T_{CLK} + 5 (was +10)		ns
T_{AVDV1}	Address (Port 0) Valid to Valid Data/Instruction In	98.66 (was 106.66)		(2+M+N)T_{CLK} – 68 (was -60)	ns
T_{AVDV3}	Address (Port 2) Valid to Valid Instruction In	23.33 (was 35.33)		(1+N)T_{CLK} – 60 (was -48)	ns
T_{AVRL}	Address Valid to RD# or PSEN# Low	37.33 (was 56.33)	(1+M)T_{CLK} – 46 (was -27)		ns
T_{AVWL1}	Address (Port 0) Valid to WR# Low	37.33 (was 56.33)	(1+M)T_{CLK} – 46 (was -27)		ns
T_{AVWL2}	Address (Port 2) Valid to WR# Low	66.33 (was 83.33)	(1+M)T_{CLK} –17 (was -0)		ns

002. Added 8x930HF0/HG0 Stepping Information to Datasheet

The *8X930Hx Universal Serial Bus Peripheral Controller* data sheet, Advance Information version (Order Number 272928-003) has been revised to include 8x930HF0/HG0 information.

003. Series Resistor Requirement for Impedance Matching Changed to 22 Ohms (no longer 27 – 33 ohms)

Section 8.4 of the *8X930Hx Universal Serial Bus Peripheral Controller* data sheet has been changed. To better match the output driver impedance, the recommended resistance is 22 ohms.

004. Added 8x930HF2/HG2 Stepping Information to Datasheet

The *8X930Hx Universal Serial Bus Peripheral Controller* data sheet, Advance Information version (Order Number 272928-004) has been revised to include 8x930HF2/HG2 information.

005. ECAP Output Voltage Supply Does Not Vary With Vcc

The voltage at the ECAP pin can be used to provide the pullup voltage for the 1.5Kohm resistor. The voltage required for the pullup is between 3.0v to 3.6v per the USB Specification, revision 1.0. ECAP voltage of the 8x930HF2/HG2 does not change when Vcc changes and the typical voltage at the ECAP pin is 3.1v (unlike the 8x930HF0/HG0 devices where the output voltage at the ECAP pin varies with Vcc).

SPECIFICATION CLARIFICATIONS

001. T_{RHDZ1} Timing

PROBLEM: The TRHDZ1 (Instruction Float After RD#/PSEN# High) specification on the 8x930H*x* when operating at 12 MHz is 10 ns. However, a slow memory device such as an EPROM typically takes approximately 30 – 90 ns to float its output. (This specification, generically called TPHZ, may vary depending on the memory type and manufacturer.) Figures 5 and 6 illustrate the TRHDZ1 timing.

The difference between the TRHDZ1 and TPHZ specifications causes contention on the data bus (P0 in nonpage page, P2 in page mode). The 8X930H***x*** begins to drive the address for the next bus cycle while the memory device is still driving data from the previous bus cycle.

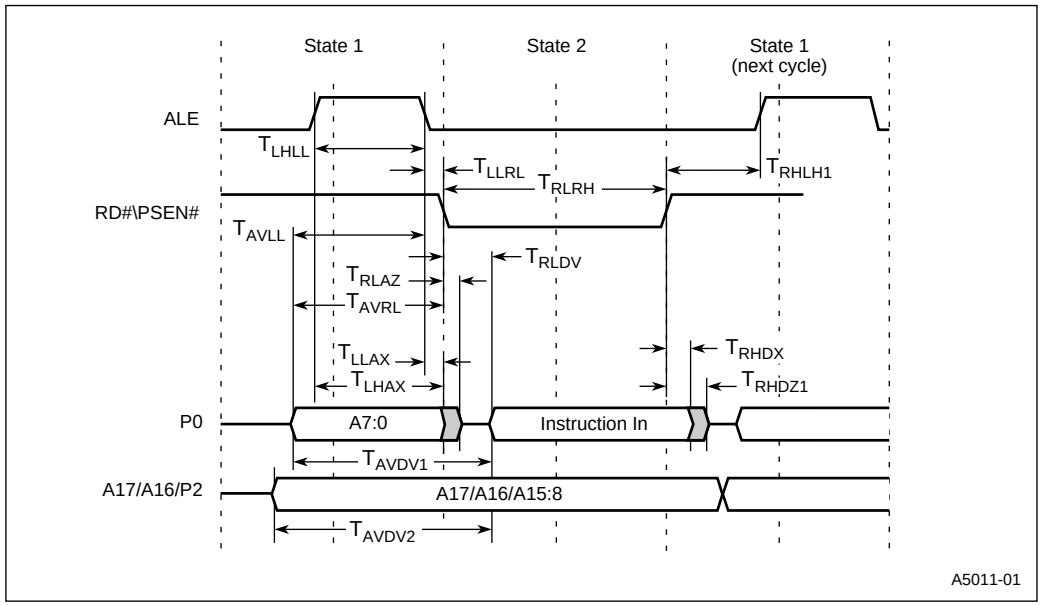

Figure 5. External Code Fetch, Nonpage Mode

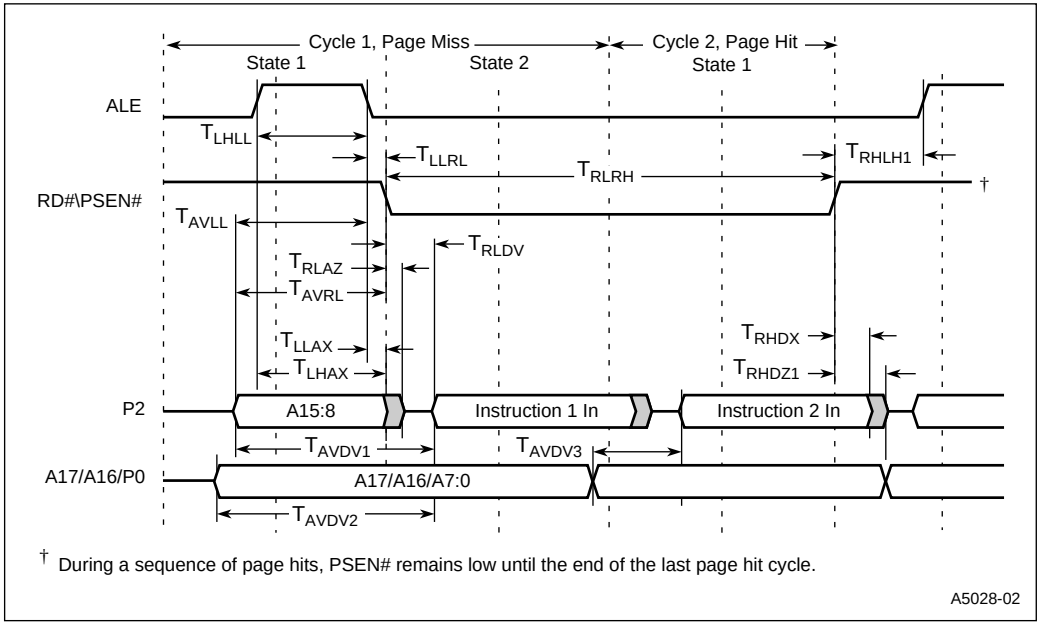

Figure 6. External Code Fetch, Page Mode

To prevent this contention, designers can use a buffer to isolate the output of the memory device from the data bus (port 0 or port 2) of the 8x930H*x*. This will prevent the memory device from driving the data bus during the critical period after T_{RHDZ1} expires. We suggest a buffer such as the 74F541 octal, three-state line driver shown in Figure 7.

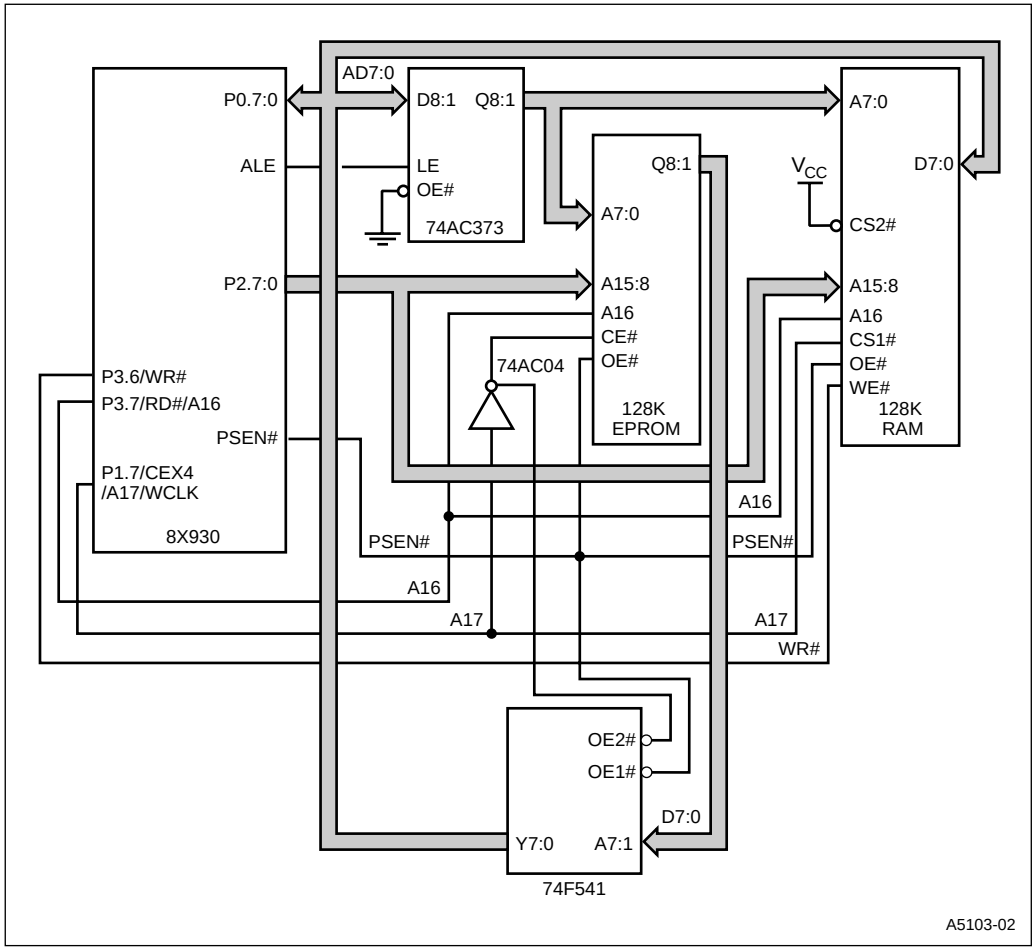

Figure 7. Example Bus Contention Solution for 8X930Hx (Nonpage Mode)

Figure 5 illustrates the connections of an 8x930H*x* configured for nonpage mode with EPROM and RAM in external memory. If your system uses a different configuration, your circuit will be different from the example. The 74F541 is enabled to pass data from the EPROM to the 8x930H*x* (port 0) when OE1# and OE2# are active. Table 3 is a truth table for the 74F541.

Table 3. Truth Table for 74F541

Inputs			Outputs
OE1#	**OE2#**	**A7:1**	**Y7:1**
L	L	L	L
L	L	H	H
X	H	X	Z
H	X	X	Z

During a read, PSEN# turns the buffer on (OE1# and OE2# are active), connecting the EPROM's output to the 8x930H*x*'s port 0.

When PSEN# goes high after a read, OE1# and OE2# are deasserted, and the buffer's output switches to the high-impedance state in approximately 9.5 ns (T_{PHZ} for a typical 74F541; the value may vary from one manufacturer to another). Thus, contention on the data bus is prevented. This or a similar hardware solution is recommended for 8X930Hx designs in which memory devices do not meet the $T_{RHDZ}1$ timing specification.

002. ECAP Pin Usage to Supply 3.0 V to 3.6 V Voltage for 1.5K Ohm USB Pullup Terminator

ITEM: For a self-powered or bus-powered device, when the voltage at the V_{CC} pins are at 5.25 V to 4.15V, the voltage at ECAP pin will be at approximately 3.6 V to 3.0V. If the V_{CC} pin is at 4.65 V [Min, Vbus Powered (host or hub) Port specification], the voltage at the ECAP pin will be at approximately 3.2 V (refer to Table 4 below). The capability for this pin to supply the 3.0 V to 3.6 V voltage to the 1.5 K Ohm USB pullup terminator depends upon the V_{CC} voltage level.

For a bus-powered device that is connected to a bus-powered hub, when the voltage at the V_{CC} pins (in the bus-powered devices) are at 4.28 V, the voltage at ECAP pin will be at approximately 3.0 V. If the V_{CC} voltage drops below 4.28 V, the ECAP pin cannot supply voltage above 3.0 V for the 1.5 K Ohm USB pullup terminator.

NOTE:
The typical ECAP values, listed in the table below, reflect a 1 µF capacitor connection between the ECAP pin and ground.

Table 4. Vcc and Typical ECAP Voltages

V_{CC}	ECAP Pin
5.25v	3.6v
5.00v	3.5v
4.65v	3.2v
4.40v	3.1v
4.28v	3.0v

003. TXCNTx Must Be Written With Correct Byte Count

PROBLEM: Both the transmit count registers, TXCNTH and TXCNTL of the 8x930H*x* embedded function transmit endpoint 1 must be written with the correct byte count after moving data into the transmit FIFO data register (TXDAT). When transmitting less than 256 bytes (FFh), the user must initially write TXCNTH with '00h', and then stage TXCNTL with the correct transmit byte count.

The 8x930Ax, 8x930Hx User's Manual (order number 272949-001) page numbers 7-17, 7-19 and C-69 contain incorrect descriptions for the reset state of the TXCNTH register, Endpoint 1. The reset state of both TXCNTH and TXCNTL registers are indeterminate.

004. Unused Dpx and Dmx Pins Must Be Pulled Down

PROBLEM: The unused USB downstream ports pins must be pulled low so that the input pins are not floated. Connect both unused D_{PX} and D_{MX} pins on the unused downstream port with a 15KΩ resistor to ground respectively to simulate a disconnect state.

DOCUMENTATION CHANGES

001. Nonvolatile Memory Verification Port Labeled Incorrectly

ITEM: The illustration “Setup for Verifying Nonvolatile Memory” (Figure 17-1 of the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual***) incorrectly depicts P1 as the Verify Modes port. The correct port for Verify Modes is P0.

002. Incorrect Address Given for TXSTAT SFR

ITEM: The USB Function SFR tables (Tables 3-11 and C-7) in the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual*** give an incorrect address for the TXSTAT SFR. The correct address is S:F2H.

003. Incorrect Signature Byte

ITEM: Section 17-6 of the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual*** mentions a signature byte at 61H. There is no signature byte at 61H.

004. Power-on Reset Capacitor Value Changed

ITEM: Figure 14-1 of the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual*** depicts a 1 µF power-on reset capacitor. The correct value for this capacitor is 0.3 µF.

005. SCON SFR’s REN Bit Description Incorrect

ITEM: The description for the Serial Port Control SFR’s REN bit (SCON.4), as given in Figure 13-2 of the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual***, is incorrect. The text should say: ”To enable reception, set this bit. To disable reception, clear this bit.”

The SCON SFR also appears in Appendix C of the same manual.

006. Extraneous Footnote in RXCON SFR

ITEM: The dagger footnote (†) does not apply to the RXFFRC and RXISO bits in the RXCON SFR, as shown in Figure 7-15 of the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual***. The RXCON SFR also appears in Appendix C. Note that the dagger footnote ***does*** apply to the SFR’s ADVWM and REVWP bits.

007. Power Off Flag Voltage Values

ITEM: Section 15.2.2 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual* states that "the hardware sets the Power Off Flag (POF) in PCON when V_{CC} rises from < 3 V to > 3 V to indicate that on-chip volatile memory is indeterminate...since for V_{CC} < 3 V data may have been lost or some logic may have malfunctioned." The voltage value should be 3.5 V for all references, not 3 V.

008. W_{CLK} Description Incorrect

ITEM: The description for the wait clock output (W_{CLK}) given in Table 16-1 and Table B-2 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual* is incorrect. Instead of "When enabled, the W_{CLK} output produces a square wave signal with a period of one-half the oscillator frequency," the final sentence should read "When enabled, the W_{CLK} output produces a square wave signal with a period of T_{CLK}."

009. Configuration Byte Misspelled

ITEM: On page 16-11 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual*, in the third paragraph of the note, UNCONFIG0 should be UCONFIG0.

010. RTWCE Description Inaccurate

ITEM: On page 16-11 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual*, in Figure 16-11, the description of RTWCE should say "with RTWE set, setting RTWCE will enable the WAIT clock.....". In other words, setting RTWCE alone will not enable the wait clock, both bits must be set.

011. RL Instruction Misspelled

ITEM: On page A-4 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual*, the 'A' should be moved to the second line for RLA and RLCA. The instruction is RL A not RLA.

012. Footnote Incorrect in Data Instructions Table

ITEM: On page A-6 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual*, Note 2 for Table A-9 is not correct. ORL, ANL, and XRL all have one instruction that uses DRk (see page A-38 for an example).

013. Hardware, Not Firmware, Sets FE Bit In SCON Register

ITEM: On page 13-7 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual*, under section 13.3 in the first paragraph, in the last sentence it should read: "If a valid stop bit is not found, the hardware sets the FE bit in the SCON register".

014. Reset Value of LC Bit (PCON.5) Should Be '1'

ITEM: On pages 15-3 (Figure 15-1), and C-43, the reset value for the LC bit in the PCON register (PCON.5) reads 'x' but should read '1' since low clock mode is automatically set after a reset.

015. Incorrect OVRI# Signal Description

ITEM: On page B-5 (Table B-2), the OVRI# description should read: "Sense input to indicate an overcurrent condition for a self-powered USB device on an external downstream port. This pin is active low with an internal pullup".

016. Remove SDIP Package Reference Throughout the Datasheet

ITEM: The following SDIP package information has been removed from the *8X930Hx Universal Serial Bus Peripheral Controller* data sheet, Advance Information version (Order Number 272928-004): Figure 5 on page 7, Table 7 on page 9, and Table 9 on page 11.