



**ANOMALIES****1. SPI Interface [er006]**


---

Background:	The SPI can either be used on the standard pins or can be moved to P3.3, P3.4, and P3.5 by setting the MSPI bit in CFG841/CFG842. When the MSPI bit is set, P3.3 should be MISO, P3.4 MOSI, and P3.5 SCLOCK.
Issue A:	By setting the MSPI bit, the P3.3, P3.4, and P3.5 have the following configuration: P3.3 = MISO, P3.4 = SCLOCK, P3.5 = MOSI
Workaround A:	None.
Issue B:	When the ADuC841/ADuC842/ADuC843 is set up as an SPI slave, the device may receive or transmit bytes incorrectly.
Workaround B:	Incorporate checksums into all communication with the ADuC841/ADuC842/ADuC843 slave. This allows the master devices to retransmit if an error occurs.
Related Issues:	None.

---

**2. Interrupts During Reading/Writing to Data FLASH/EE [er007]**


---

Background:	There are 4 kB of DATAFLASH/EE that can be used for nonvolatile data storage.
Issue:	If an interrupt occurs during a DATAFLASH/EE read or write operation, code execution following the ISR may resume at a random program memory address.
Workaround:	Disable all interrupts prior to a read or write operation. This can be done by setting the EA bit to 0.
Related Issues:	None.

---

**3. PWM Operation [er008]**


---

Background:	The PWM output rate is determined by the PWMxH and PWMxL registers for the PWM0 and PWM1 outputs.
Issue:	Modifying RAM Address 0x2E causes the PWM timer to be reset.
Workaround:	For Assembly code: Do not use memory location 0x2E. For C code: Assign a dummy variable to location 0x2E using the following code: <code>idata unsigned int ui32Dummy[2] _at_ 0x2E;</code>
Related Issues:	None.

---

**4. Watchdog Timer [er009]**


---

Background:	The ADuC841, ADuC842, and ADuC843 incorporate a Watchdog Timer. The purpose of the WDT is to ensure the part is never stuck in an endless loop by generating either a hardware reset or an interrupt event that vectors to the WDT ISR.
Issue:	If the WDT generates an interrupt as opposed to a hardware reset, and if the ISR subsequently sets up the WDT to time out to a hardware reset, the reset is ignored.
Workaround:	Ensure that a double write to the WDCON is executed inside the ISR with the first write being a reset of the WDT. For example: <pre>void isr_wdt( void ) interrupt 11 {     WDWR = 1;    // This first WDT write is required to get the WDT to work inside the ISR.     WDCON = 0x60; // Reset WDT.     WDWR = 1;    // Now set the WDT to the required 1s timeout     WDCON = 0x62; // select reset after 1000mS     while(1); } void main(void) {     EA = 0;     WDWR = 1;    // Allow write to WDCON     WDCON = 0x6A; // timeout=1000mS, WDT enable, WDT ISR Interrupt     while (1); }</pre>
Related Issues:	None.

---

**5. Level Triggered Interrupt Operation [er010]**

Background:	The ADuC841/ADuC842/ADuC843 incorporate two external interrupt sources (INT0 and INT1) that can be configured to respond to either an edge event or a level event.
Issue:	If an interrupt occurs on the INT0 or INT1 pins and is then removed within one core instruction cycle, the interrupt vector address that is generated may be incorrect resulting in a vector to 0000H. This effectively restarts code execution.
Workaround:	To ensure that this does not occur, the level triggered interrupt source must be kept low for a minimum of 9 core clock cycles.
Related Issues:	None.

**ADuC841/ADuC842/ADuC843 SILICON ANOMALIES**

<b>Anomaly No.</b>	<b>Description</b>	<b>Status</b>
er001	Mode 0 UART operation	Fixed
er002	Use of the extended stack pointer	Fixed
er003	Use of I <sup>2</sup> C in slave mode with stop interrupt enabled	Fixed
er004	Use of I <sup>2</sup> C in slave mode with stop interrupt disabled	Fixed
er005	I <sup>2</sup> C data transfer	Fixed
er006	SPI interface	Pending
er007	Interrupts during reading/writing to data FLASH/EE	Pending
er008	PWM operation	Pending
er009	Watchdog timer	Pending
er010	Level triggered interrupt operation	Pending
er011	Stack pointer in ULOAD mode	Fixed

**NOTES**

Purchase of licensed I<sup>2</sup>C components of Analog Devices or one of its sublicensed Associated Companies conveys a license for the purchaser under the Philips I<sup>2</sup>C Patent Rights to use these components in an I<sup>2</sup>C system, provided that the system conforms to the I<sup>2</sup>C Standard Specification as defined by Philips.