# MicroConverter®, Multi-Channel 12-bit ADCs and DACs with Embedded FLASH MCU

# Errata Sheet — ADuC831

**A**. This Errata sheet contains the following known bugs, anomalies and work-arounds for the ADuC831 MicroConverter

**831_S01. NUMBER OF ADC CLOCKS IN PIPLINED MODE**

**831_S02. USE OF TIME INTERVAL COUNTER IN POWER DOWN MODE**

**831_S03. DAC CLEAR BIT FUNCTIONALITY**

**831_S04. EXTENDED (11-BIT) STACK POINTER - PUSH AND POP OPERATION**

**B.** The Errata listed, apply to all ADuC831 packaged material branded as follows:

First Line: **ADuC831XX**

Fourth Line: **BXX**

**C.** Analog Devices Inc. is committed, through future silicon revisions to continuously improve silicon functionality. Analog Devices Inc. will use its best endeavors to ensure that these future silicon revisions remain compatible with your present software/systems that implement the recommended work-arounds outlined in this document.

REV. B.1 November/2002



**One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.**
**Tel: 781/329-4700 World Wide Web Site: http://www.analog.com**
**Fax: 781/326-8703**

### 831_S01. NUMBER OF ADC CLOCKS IN PIPLINED MODE

**Backgound:** An adc conversion takes 16 ADC clocks plus the selected acquisition clocks.

**Issue:** At voltages below 3.5V in piplined mode the number of ADC clocks required for a conversion can vary between 15 and 16 ADC clocks.

**Work-Around :** none.

**Related Issues :** This issue does not affect the performance of the ADC.

---

### 831_S02. USE OF TIME INTERVAL COUNTER IN POWER DOWN MODE

**Backgound:** The Time Interval Counter (TIC) can be used to time longer intervals than the standard 8052 timers are capable of. It can be used to interrupt or 'wake up' the ADuC831 after a specific interval has elapsed.
There are five ways of terminating power down mode on the ADuC831. Three of these ways namely, TIC interrupt, I2C/SPI interrupt or INTO interrupt resume code execution at the point where it was powered down.

**Issue:** When the TIC is activate (i.e. TIMECON.1 = 1)during power down a 'wake up' interrupt from any source may trigger a full Power-On-Reset event instead of resuming code execution at the point where it was powered down. This results in all SFRs being reset to their default values.

**Work-Around :** A bit can be set in dataflash prior to powerdown. By monitoring this bit at the start of code execution the user can determine if the reset was due to a 'wake up' from powerdown.

**Related Issues :** This issue does not affect 'wake up' from powerdown if TIMECON.1=0

---

### 831_S03. DAC CLEAR BIT FUNCTIONALITY

**Backgound:** The DAC outputs are controlled by the SFR DACCON. The CLR1 and CLR0 bits in DACCON can be used to force the output of DAC1 and DAC0 to 0V.

**Issue:** When the DAC is enabled and in buffered mode setting the CLR1 or CLR0 bits may cause a momentary spike on the DAC ouput before the DAC is forced to 0V.

**Work-Around :** To force the output of DAC1 or DAC0 to 0V write 0000H to the corresponding DAC data registers.

**Related Issues :** none.

---

**831_S04. EXTENDED (11-BIT) STACK POINTER - PUSH AND POP OPERATION**

**Backgound:** The ADuC831 offers an extended (11-bit) stack pointer that allows the stack to extend into the 2 KBytes of internal XRAM. This can be very useful where embedded functions are used.

**Issue:** If the extended stack pointer is enabled (CFG831.7=1) and the stack points to the extended stack space (SPH>=1) the the *PUSH direct* or *POP direct* instructions will not operate correctly if the direct address is less than 80H (i.e. not an SFR)

**Work-Around :**

Assembly Programming: By using the Accumulator the extended stack works correctly. e.g.

To PUSH:
```
MOV  A,0
PUSH ACC
```
To POP:
```
POP  ACC
MOV  0,A
```

C Programming (KEIL Compier):

The KEIL Compiler only ever pushes (or pops) and SFR or any of the 32 registers (4 banks of 8 registers) onto the stack. Keil support a compiler directive that disables absolute register addressing (#pragma NOAREGS). Using this directive a PUSH/POP register will be automatically changed as above.

```
#pragma  NOAREGS
int  increment(int);

void  main(void)
{
int  a,b,c;

a=5;
b=6;
c=increment(a)+increment(b);

while(1);
}

int  increment(int  a)
{
return(a+1);
}
```

**Related Issues :** none.

# AMEYA360
Components Supply Platform

## Authorized Distribution Brand :

## Website :

Welcome to visit www.ameya360.com

## Contact Us :

- ### Address :

  401 Building No.5, JiuGe Business Center, Lane 2301, Yishan Rd
  Minhang District, Shanghai , China

- ### Sales :

  Direct +86 (21) 6401-6692

  Email amall@ameya360.com

  QQ 800077892

  Skype ameyasales1 ameyasales2

- ### Customer Service :

  Email service@ameya360.com

- ### Partnership :

  Tel +86 (21) 64016692-8333

  Email mkt@ameya360.com