Silicon Errata Sheet   ADuC836

A. This Errata sheet represents the known bugs, anomalies and work-arounds for the ADuC836 MicroConverter.

B. The Errata listed, apply to all ADuC836 packaged material branded as follows:

> First Line: **ADuC836BS or ADuC836BCP**
> Fourth Line: **F23**

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 implementing the recommended work-arounds outlined in this document.

D. ADuC836 Silicon Errata Sheet Revision History :

Revision	Date	Relevance	Silicon Status	# of Bugs Reported
F	April 2009	All Silicon branded ADUC836BS or ADuC836BCP Fourth Line: F23	Release	1 Errata

REV. F

2 EXTENDED (11-BIT) STACK POINTER – PUSH AND POP OPERATION

Background: The ADuC836 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 (CFG836.7=1) and the stack points to the extended stack space (SPH>=1) then *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:
A fix for the bug is to PUSH or POP the data through the accumulator. i.e. instead of

```
      PUSH   00h
```
Use
```
      MOV    A, 00H    ;(or MOV A, R0 if register bank 0 selected)
      PUSH   ACC
```

instead of
```
      POP    00h
```
use
```
      POP    ACC
      MOV    00H, A   ;(or MOV R0, A if register bank 0 selected)
```

C Programming (KEIL Compiler)
The Keil Compiler only ever pushes (or pops) an 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

ADuC836 Silicon Errata Revision History

Errata #	Description	Status
er1	WAKEUP FROM POWERDOWN	Fixed
er2	EXTENDED (11-BIT) STACK POINTER – PUSH AND POP OPERATION	New

ANALOG
DEVICES

www.analog.com