DESIGN NOTE
#016

How to Separate ATmega128 from ATmega103 from within the Application code

Introduction

ATmega128 was introduced as an advanced AVR mega device with a complex feature set. In addition it is a drop-in replacement of the ATmega103, meaning that it has a very high degree of compatibility with the ATmega103. As the ATmega128 is far more complex than the ATmega103 it has been equipped with the M103C ("Mega103 compatibility") Fuse. The M103C Fuse "downgrades" the ATmega128 to behave like an ATmega103, to make drop-in replacements easier. However there are still there some minor differences. These differences are described in the application note "AVR080: Replacing ATmega103 by ATmega128".

To get around these differences it can be desirable that the application code is able to recognize if the device is an ATmega103 or an ATmega128 with M103C Fuse set.

This design note shows a method of telling if the device is an ATmega128 or an ATmega103.

Overview

Since it is not possible to read out the device signature from the Application code some of the differences between the ATmega103 and the ATmega128 running in M103C mode have to be exploited; Some AVR I/O Registers of the ATmega103 have unused/reserved bits. These bits are always read as "0", even though "1" is written to the bit in advance to reading it. Not all of these unused bits of the ATmega103 are unused in the ATmega128

To test if the device is an ATmega128 or an ATmega103, write a "1" to a bit available in ATmega128 but not in ATmega103. If it is an ATmega128 the bit will read back "1" while an ATmega103 will read back "0".

Bits that are Available in the ATmega128 are:
- The IVSEL bit in MCUCR
- The REFS1, REFS0, and ADLAR bits in ADMUX.

Note that this method will not be able to separate an ATmega128 from an ATmega128 with the M103C Fuse set. The ATmega128 will respond in the same way in both these two modes.

Since IVSEL can only be set to 1 using a timed sequence with the IVCE bit, the best solution is to use the ADMUX register bits. In any case, the bits used for testing should be restored to their original values.

If, for some reason, the IVCE bit is used, remember to disable interrupts. The IVCE bit will be cleared by hardware after 4 CPU cycles, and an interrupt could invalidate the test.

Code Example

```c
//C-code that determines if an ATmega103 or an ATmega128 is used
bool m128 = false;
ADMUX |= (1<<ADLAR); // Set the ADLAR bit in the ADMUX register
if( ADMUX & (1<<ADLAR) ) // If ADLAR bit is still set
{
  m128 = true;                //...then the device is an ATmega128
}
```