AVR000: Register and Bit-Name Definitions for the AVR Microcontrollers

Introduction

This application note contains files which allows the user to use Register and Bit names from the databook when writing assembly programs. To use the files, simply include them in the top of the source code. The files are named according to the following convention:

```
<Part Number>def.inc
```

As an example, AT90S8515 programs should include the following assembler directive:

```
.include "8515def.inc"
```

In addition, the pointer registers R26 - R31 have been assigned names according to the following table:

Table 1. Pointer Name Definitions

Register	Name
R26	XL
R27	XH
R28	YL
R29	YH
R30	ZL
R31	ZH

Note: For the AT90S1200 or similar, the only defined pointer is R30 - ZL.

For controllers with SRAM, the constant “RAMEND” is defined. For all devices, the constants “FLASHEND” and “EEPROMEND” are defined. This number is useful when initializing the Stack pointer to point at the highest internal SRAM address. Finally, the interrupt addresses have been defined, and can be used together with the “.org” directive in the assembler to position an interrupt vector at the correct memory location. See the file listing for details on this.

To prohibit use of non-implemented instructions, all files contain a “.device” directive for the target MCU.

As new AVR products are released, new files will be made available.

Usage

Bit names in the files are defined as numbers 0-7. The user should be aware of the difference between using bit names with instructions that take bit masks as operands, and instructions that take bit numbers as operands.

Instructions that take bit masks are:

- CBR- Clear Bit in Register
- SBR - Set Bit in Register

Instruction that take bit numbers are:

- CBI - Clear Bit in I/O register
- SBI - Set Bit in I/O register
- SBIC- Skip if Bit in I/O Register Cleared
- SBIS- Skip if Bit in I/O Register Set
- SBRC- Skip if Bit in Register Cleared
- SBRS- Skip if Bit in Register Set
- BLD- Bit LoaD from T-flag
- BST- Bit STore to T-flag

To convert a bit number to a bit mask, use the shift left-operator (“<<”) in the assembler. Observe that the “+” operator has precedence over “<<”. See the following program example:

```
sbr   r16,(1<<SE)+(1<<SM)
out   MCUCR,r16   ;set SE and SM
                  ;in MCUCR
```

8-Bit AVR MCU with Downloadable Flash

Application Note

Rev. 0931B–04/98