AVR108 Design Note: Setup and Use of the LPM Instruction

Features

- **Use of the LPM (Load Program Memory) Instruction with the AVR Assembler**
- **Load Data Constants from PROGRAM MEMORY**
- **Use of Look-Up Tables**
- **Example Assembly Code Included for AT90S8515**

Introduction

This design note describes how to access data constants saved in the program memory of the AVR microcontrollers. The on-board non-volatile flash memory is the perfect place to store strings, images, and look-up tables that do not change run-time.

Use of the LPM

In the AVR, flash program memory is separate from the data memory. The LPM instruction is included in the AVR instruction set to copy data bytes from the flash program memory into the data memory.

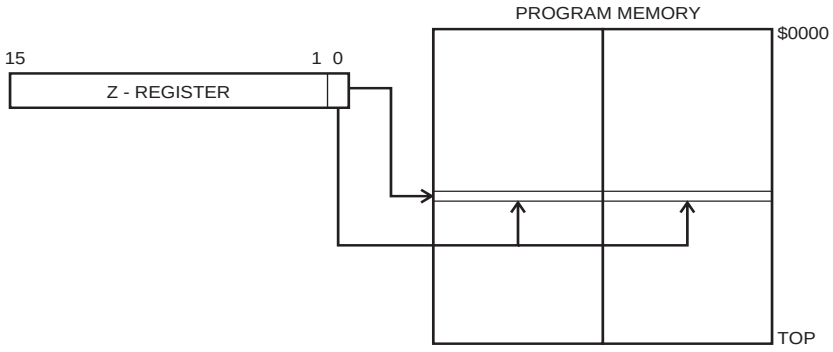
The flash program memory of the AVR microcontroller is organized as 16 bits words. The register file and SRAM data memory are organized as 8-bit bytes. Special consideration must therefore be taken when loading data from program memory to data meory.

The Z-pointer(R31:R30) can be used to point into the program memory. The 15 most significant bits select the word address in program memory

The least significant bit of the Z address register selects either low byte (0) or high byte(1) of the program memory word. To calculate the low (ZL) and high (ZH) part of the address, use the LOW() and HIGH() functions provides in the assembler.

When the Z-pointer is set up properly, the LPM instruction copies the byte pointed to into the R0 register. To load data from random places in program memory, the Z register must be set up with the proper address each time a new address is accessed. If data is stored sequentially, the ADIW(Add Immediate to Word) can be used to increment the Z-pointer.

Figure 1. Z address register



AVR® 8-bit RISC Microcontroller

Application Note

Rev. xxxxA–02/99

Example Program

The program in this application note loads a string of bytes from the program memory, and writes it to Port B. It first initializes Port B so that all the pins are output. It loads the starting address of the string “Hello World” into the Z register, as described above. Then a byte is loaded from program memory using LPM. The program checks whether or not the end of the string is reached (byte was zero). If the end is not reached yet the last read byte is written to Port B, a short delay is made, and the Z register is increased. The program then jumps back to load the next byte.

```
;**** A P P L I C A T I O N   N O T E   A V R 1 0 8 D N ************************
;*
;* Title:        Load Program Memory
;* Version:      1.0
;* Last updated: 98.12.17
;* Target:       All devices with LPM instruction
;*
;* Support E-mail:avr@atmel.com
;*
;* DESCRIPTION
;* This Application note shows how to use the Load Program Memory (LPM)
;* instruction. The App. note loads the string “Hello World” from
;* program memory byte by byte, and puts it onto port B.
;*
;*******************************************************************************

  .include “8515def.inc”
  .device   AT90S8515                          ; Specify device
  .def      temp  =r16                         ; Define temporary variable


start:
    ldi     temp,low(RAMEND)
    out     SPL,temp                           ; Set stack pointer to last internal RAM location
    ldi     temp,high(RAMEND)
    out     SPH,temp

    ldi     temp,$ff
    out     PORTB,temp                         ; Set all pins at port B high
    out     DDRB,temp                          ; Set port B as output

  ; Load the address of ‘message’ into the Z register. Multiplies
  ; word address with 2 to achieve the byte address, and uses the
  ; functions high() and low() to calculate high and low address byte.

    ldi     ZH,high(2*message)                 ; Load high part of byte address into ZH
    ldi     ZL,low(2*message)                  ; Load low part of byte address into ZL

loadbyte:
    lpm                                        ; Load byte from program memory into r0
    tst     r0                                 ; Check if we’ve reached the end of the message
    breq    quit                               ; If so, quit
```

```
    out     PORTB,r0                        ; Put the character onto Port B
    rcall   one_sec_delay                   ; A short delay

    adiw    ZL,1                            ; Increase Z registers
    rjmp    loadbyte

    quit:   rjmp quit                       ; Stop here when done

one_sec_delay:                              ; One second delay at 1 MHz
    ldi     r20, 20
    ldi     r21, 255
    ldi     r22, 255
delay:
    dec     r22
    brne    delay
    dec     r21
    brne    delay
    dec     r20
    brne    delay
    ret

  message:
  .db     “Hello World”, 0
```

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686677
FAX (44) 1276-686697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road
Tsimshatsui East
Kowloon, Hong Kong
TEL (852) 27219778
FAX (852) 27221369

Japan
Atmel Japan K.K.
Tonetsu Shinkawa Bldg., 9F
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4 42 53 60 00
FAX (33) 4 42 53 60 01

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

xxxxA–02/99/xM