Sleep Modes and Achieving Low-power Supply Current on AT94 Series FPSLIC™ Devices

AT94 Sleep Modes and Low-power Supply Current

Application Note

Introduction

The purpose of this application note is to provide designers with an understanding of the 3 available Sleep Modes and how to achieve low-power supply current on AT94 series FPSLIC devices.

Application

To enter any of the Sleep Modes, the SE bit in the MCU Control/Status Register (MCUR) has to be enabled and a SLEEP instruction has to be executed. The MCU Register also contains control bits for general MCU functions and status bits to indicate the source of an MCU Reset. For more information about the MCU Register and its corresponding bits, please refer to the "AVR Core and Peripherals" section in the "*AT94K Series FPSLIC*" datasheet.

MCU Register

Bit	7	6	5	4	3	2	1	0
	JTRF	JTD	SE	SM1	SM0	PORF	WDRF	EXTRF

The SM1 and SM0 bits in the MCUR Register select which Sleep Mode (Idle, Power-save, and Power-down) will be activated by the SLEEP instruction. Table 1 provides you information on how to configure SM1/SM0 bits for your desired mode of SLEEP operation.

Table 1. SLEEP Mode Select

SM1	SM0	Sleep Mode
0	0	Idle
0	1	Reserved
1	0	Power-down
1	1	Power-save

3503A–FPSLI–8/04

The FPGA core's Global clock buffers GCK5 and GCK6 are sourced by either the AVR® core clock (XTAL), the AVR Timer clock (TOSC), or the AVR core's internal Watchdog Oscillator (WDT). The functionality of GCK5 and GCK6 as it relates to the Idle, Power-Save and Power-Down modes is given in Table 2.

Table 2. Clock Activity in Various Modes

Mode	Clock Source	GCK5	GCK6
Idle	XTAL	Active	Active
	TOSC	Not Available	Active
	WDT	Not Available	Active
Power-save	XTAL	Inactive	Inactive
	TOSC	Not Available	Active
	WDT	Not Available	Active
Power-down	XTAL	Inactive	Inactive
	TOSC	Not Available	Inactive
	WDT	Not Available	Active

The following sections explain in detail each mode of the SLEEP operation.

Idle Mode

When the SM1/SM0 bits are set to 00, the SLEEP instruction makes the MCU enter the Idle mode, stopping the CPU but allowing UARTs, Timer/Counters, Watchdog 2-wire Serial and the Interrupt System to continue operating. This enables the MCU to wake-up from externally triggered interrupts as well as internal ones like the Timer Overflow and UART Receive Complete interrupts. When the MCU wakes up from Idle mode, the CPU starts program execution immediately.

Power-down Mode

When the SM1/SM0 bits are set to 10, the SLEEP instruction makes the MCU enter the Power-down mode. In this mode, the external oscillator is stopped, while the external interrupts and the watchdog (if enabled) continue operating. Only an external reset, a watchdog reset (if enabled), or an external level interrupt can wake-up the MCU. In Power-down and Power-save modes, the four external interrupts, EXT_INT0...3, and FPGA interrupts, FPGA_INT0...3, are treated as low-level triggered interrupts.

If a level-triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake-up the MCU. This makes the MCU less sensitive to noise. The changed level is sampled twice by the watchdog oscillator clock, and if the input has the required level during this time, the MCU will wake-up. The period of the watchdog oscillator is 1 µs (nominal) at 3.3V and 25°C. The frequency of the watchdog oscillator is voltage dependent.

When waking up from Power-down mode, there is a delay from the time a wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined by the same time-set bits that define the reset time-out period. The wake-up period is equal to the clock-reset period. If the wake-up condition disappears before the MCU wakes up and starts to execute, the interrupt causing the wake-up will not be executed.

Power-save Mode

When the SM1/SM0 bits are 11, the SLEEP instruction makes the MCU enter the Power-save mode. This mode is identical to power-down, with one exception: If Timer/Counter2 is clocked asynchronously, i.e., the AS2 bit in ASSR is set, Timer/Counter2 will run during sleep. In addition to the power-down wake-up sources, the device can also wake-up from either an Timer Overflow or Output Compare event from Timer/Counter2 if the corresponding Timer/Counter2 interrupt enable bits are set in TIMSK. To ensure that the part executes the Interrupt routine when waking up, also set the global interrupt enable bit in SREG.

When waking up from Power-save mode by an external interrupt, two instruction cycles are executed before the interrupt flags are updated. When waking up by the asynchronous timer, three instruction cycles are executed before the flags are updated. During these cycles, the processor executes instructions, but the interrupt condition is not readable, and the interrupt routine has not started yet.

Current Dissipation in Various Modes

To find current dissipation characteristics during each mode of SLEEP operation, please refer to the "*AT94K Series FPSLIC*" datasheet. Note that to achieve minimal current values, the XTAL Pad Bias Resistor should be disabled. Refer to the XTAL bias resistor option under "Device Programming" with-in the System Designer toolset for more details.

AVR Code Examples

Example assembly code is provided for all 3 SLEEP modes in this application note. Upon initial execution of each of the three code examples, PortD will alternate between 55h and AAh ten times before outputting 00h and entering Sleep Mode. The device wakes up from sleep due to the noted wake-up mechanism and starts repeating 55h and AAh patterns.

In the Idle mode code example, a FPSLIC device is put to sleep and wakes upon a Timer/Counter1 Overflow Interrupt.

In the Power-down mode code example, a FPSLIC device is put to sleep and wakes upon an External Interrupt.

In the Power-save mode code example, a FPSLIC device is put to sleep and wakes upon a watchdog reset.

The following code is available on the Atmel web site.

```
;*********************************************************************
;* File Name:           avr_idle.asm
;* Title:               FPSLIC Sample Design 1 (Assembly)
;* Version:             1.1
;* Last Updated:        May 24, 2004
;* Target Device:       AT94K05/10/40
;* Purpose:            To put the FPSLIC AVR Core to Sleep in Idle Mode
;*                     and wake it upon a Timer/Counter1 Overflow
;*                     Interrupt
;* Hardware used:      ATSTK94 FPSLIC Starter Kit. The Starter Board
;*                     can be used to observe the behavior of the
;*                     FPSLIC as its LED's go from flashing 55/AAh in
;*                     the active state to 00h in the sleep state
;*********************************************************************

;***** Include Files  *****
.include "AT94KDEF.INC"

;***** Global Registers *****
;* Define Symbolic Names for Registers
;* Remember immediate operations (i.e. ldi r16, 0x05) available on r16
;* through r31

.def    temp      = r17     ;Temporary Storage Register
.def    counter1  = r20     ;Counter used for delay
.def    counter2  = r21     ;Counter used for delay
.def    counter3  = r22     ;Counter used for delay
.def    counter4  = r23     ;Register to set 55/AAh toggle count
.def    pattern   = r24     ;Register used to load the pattern

;***** Interrupt Vectors *****
.org RESETaddr
   rjmp RESET               ;Jump to Reset Interrupt

.org TIM1_OVFaddr
   rjmp TIM1_OVF            ;Jump to Timer1 Overflow Interrupt
```

```
; ----------------------------------------------------------------------
; Interrupt Service Routine
; ----------------------------------------------------------------------
TIM1_OVF:                     ;Timer1 Overflow Interrupt Service Routine

delay_ISR:
   inc      counter1         ;After waking from sleep mode as a result
   cpi      counter1, 0xAA   ;of Timer/Counter 1 interrupt, this
   brne     delay_ISR        ;routine is used to delay the indication
   clr      counter1         ;of FPSLIC being in active mode by keeping
   inc      counter2         ;LED's OFF(on ATSTK94 board).
   cpi      counter2, 0xAA   ;This loop is strictly used for display
   brne     delay_ISR        ;purpose, in case of measuring idle mode
   clr      counter2         ;power consumption comment this code.

   reti

; ----------------------------------------------------------------------
; Initialization Routine
; ----------------------------------------------------------------------
RESET:                        ;Initialize Stack Pointer and DDRD
   ldi      temp,0x0f
   out      SPH,temp          ;Initialize Stack Pointer-> 0fff
   ldi      temp,0xff
   out      SPL,temp
   out      DDRD,temp         ;Configure PortD pins as output pins
   ldi      pattern, 0x55

   ldi      temp, (1<<TOIE1)  ;Enable Timer/Counter1 Overflow interrupt
   out      TIMSK,temp
   clr      Temp
   out      TCNT1H,temp       ;Count value is at 0000
   out      TCNT1L,temp
   ldi      Temp,(1<<CS12)+(0<<CS11)+(1<<CS10)
   out      TCCR1B,temp       ;Timer Pre-scalar set to ck/1024

   ldi      Temp, (1<<SE)+(0<<SM1)+(0<<SM0)
   out      MCUR, temp        ;Enable Sleep in Idle Mode

   sei                        ;Enable global interrupts
```

```
; ----------------------------------------------------------------------
; Main Routine
; ----------------------------------------------------------------------
MAIN:
   clr      counter1          ;Initialize Counters
   clr      counter2
   clr      counter3
   clr      counter4
   out      PORTD,pattern     ;Display 55h on LEDs

delay:                        ;Embedded delay loop to slow the display
   inc      counter3          ;of AAh & 55h pattern, making it
   cpi      counter3, 0xff    ;observable on STK94 starter board
   brne     delay
   clr      counter3
   inc      counter2
   cpi      counter2, 0xff
   brne     delay
   clr      counter2
   inc      counter1
   cpi      counter1, 0x0f
   brne     delay
   clr      counter1
   com      pattern           ;One's complements the pattern(55h-> AAh)
   out      PORTD,pattern
   inc      counter4
   cpi      counter4, 0x0A    ;55/AAh toggles 10 times
   brne     delay
   clr      counter4

   clr      temp              ;LEDs are switched off to infer idle sleep
   out      PORTD,temp        ;mode on STK94 starter board
   sleep                      ;Go to sleep (wait for Timer/Counter1 int)
   rjmp     MAIN              ;Jump to MAIN and repeat displaying 55/AAh
                              ;pattern after waking-up from sleep
```

```
;***********************************************************************
;* File Name:            avr_powerdown.asm
;* Title:                FPSLIC Sample Design 1 (Assembly)
;* Version:              1.1
;* Last Updated:         May 26, 2004
;* Target Device:        AT94K05/10/40
;* Purpose:             To put the FPSLIC AVR Core to Sleep in Power-
;*                      Down Mode and wake it upon an External Interrupt
;* Hardware used:       ATSTK94 FPSLIC Starter Kit. The Starter Board
;*                      can be used to observe the behavior of the
;*                      FPSLIC as its LED's go from flashing 55/AAh in
;*                      the active state to 00h in the sleep state
;***********************************************************************

;***** Include Files  *****
.include "AT94KDEF.INC"

;***** Global Registers *****
;* Define Symbolic Names for Registers
;* Remember immediate operations (i.e. ldi r16, 0x05) available on r16
;* through r31

.def     temp      = r16        ;Temporary Storage Register
.def     counter1  = r21        ;Registers used for delay
.def     counter2  = r22        ;Registers used for delay
.def     counter3  = r23        ;Registers used for delay
.def     counter4  = r24        ;Register to set 55/AAh toggle count
.def     pattern   = r25        ;Register that holds the LED pattern

;***** Program Start - Execution Starts Here *****
;
;***** Interrupt Vectors
    .org RESETaddr
        rjmp RESET              ;Jump to Reset Interrupt
    .org EXT_INT0addr
        rjmp EXT_INT0           ;Jump to External Interrupt
```

```
; ----------------------------------------------------------------------
;Interrupt Service Routine
; ----------------------------------------------------------------------
EXT_INT0:                     ;External Interrupt Service Routine
   reti                       ;Return from interrupt after waking up
                              ;from Power-Down mode

; ----------------------------------------------------------------------
; Initialization Routine
; ----------------------------------------------------------------------
RESET:                        ;Sets up Stack Pointer and PORTD
   cli                        ;Disable Global Interrupts
   ldi     temp,0x0f
   out     SPH,temp          ;Initialize Stack Pointer
   ldi     temp,0xff
   out     SPL,temp          ;Initialize Stack Pointer
   out     DDRD,temp         ;Configure PortD pins as outputs
   ldi     pattern, 0x55     ;Pattern displayed is 55

   ldi     temp, 0x01
   out     EIMF, temp        ;Enable External Interrupt0

   ldi     temp, (1<<SE)+(1<<SM1)+(0<<SM0)
   out     MCUR, temp        ;Enable Sleep in Power Save Mode

   sei                        ;Enable global interrupts
```

```
; ----------------------------------------------------------------------
; Main Routine
; ----------------------------------------------------------------------
MAIN:
   clr    counter1            ;Initialize Counters
   clr    counter2
   clr    counter3
   clr    counter4
   out    PORTD, pattern      ;Display 55h on LEDs

delay:                        ;Embedded delay loop to slow the display
   inc    counter3            ;of AAh & 55h pattern, making it
   cpi    counter3, 0xff      ;observable on STK94 starter board
   brne   delay
   clr    counter3
   inc    counter2
   cpi    counter2, 0xff
   brne   delay
   clr    counter2
   inc    counter1
   cpi    counter1, 0x0f
   brne   delay
   clr    counter1
   com    pattern             ;One's complement the pattern(55h->AAh)
   out    PORTD, pattern
   inc    counter4
   cpi    counter4, 0x0A      ;55/AAh toggles 10 times;
   brne   delay
   clr    counter4

   clr    temp                ;LEDs are switched off to infer idle
   out    PORTD,temp          ;sleep mode on STK94 starter board
   sleep                      ;Go to sleep (wait for External Interrupt)
   rjmp   MAIN                ;Jump to MAIN and repeat displaying 55/AAh
                              ;pattern after waking-up from sleep
```

```
;***********************************************************************
;* File Name:           avr_powersave.asm
;* Title:               FPSLIC Sample Design 1 (Assembly)
;* Version:             1.1
;* Last Updated:        May 26, 2004
;* Target Device:       AT94K05/10/40
;* Purpose:             To put the FPSLIC AVR Core to Sleep in Power-
;*                      Save Mode and wake it upon an External Interrupt
;* Hardware used:       ATSTK94 FPSLIC Starter Kit. The Starter Board
;*                      can be used to observe the behavior of the
;*                      FPSLIC as its LED's go from flashing 55/AAh in
;*                      the active state to 00h in the sleep state
;***********************************************************************

;***** Include Files *****
.include "AT94KDEF.INC"

;***** Global Registers *****
;* Define Symbolic Names for Registers
;* Remember immediate operations (i.e. ldi r16, 0x05) available on r16 ; ;* through r31

.def    temp      = r16       ;Temporary Storage Register
.def    counter1  = r21       ;Register used for delay
.def    counter2  = r22       ;Register used for delay
.def    counter3  = r23       ;Register used for delay
.def    counter4  = r24       ;Register to set 55/AAh toggle count
.def    pattern   = r25       ;Register that holds the LED pattern

;***** Program Start - Execution Starts Here *****
;***** Interrupt Vectors *****
   .org RESETaddr
     rjmp RESET                ;Jump to Reset Interrupt
```

```
; ----------------------------------------------------------------------
; Initialization Routine
; ----------------------------------------------------------------------
RESET:                        ;Sets up Stack Pointer and PORTD
  ldi     temp,0x0f
  out     SPH,temp            ;Initialize Stack Pointer
  ldi     temp,0xff
  out     SPL,temp            ;Initialize Stack Pointer
  out     DDRD,temp           ;Configure PortD pins as outputs
  ldi     pattern, 0x55       ;Displays 55 on LEDs

  ldi     temp, (0<<WDE)+(1<<WDP2)+(1<<WDP1)+(1<<WDP0)
  out     WDTCR, temp         ;Set Watchdog Timer Time-out to 1.9s

  ldi     temp, (1<<SE)+(1<<SM1)+(1<<SM0);
  out     MCUR, Temp          ;Enable Sleep in Power Save Mode
```

```
; ----------------------------------------------------------------------
; Main Routine
; ----------------------------------------------------------------------
MAIN:
  clr     counter1              ;Initialize counters
  clr     counter2
  clr     counter3
  clr     counter4
  out     PORTD, pattern        ;Display 55h on LEDs

delay:                          ;Embedded delay loop to slow the display
  inc     counter3              ;of AAh & 55h pattern, making it
  cpi     counter3, 0xff        ;observable on STK94 starter board
  brne    Delay
  clr     counter3
  inc     counter2
  cpi     counter2, 0xff
  brne    Delay
  clr     counter2
  inc     counter1
  cpi     counter1, 0x0f
  brne    Delay
  clr     counter1
  com     pattern               ;One's complement the pattern(55h->AAh)
  out     PORTD, pattern
  inc     counter4
  cpi     counter4, 0x0A        ;55/AAh toggles 10 times
  brne    Delay
  clr     counter4

  clr     temp                  ;LEDs are switched off to infer idle
  out     PORTD,temp            ;sleep mode on STK94 starter board
  WDR
  in      Temp, WDTCR           ;In WDTCR value into Temp
  ori     Temp, 0x08            ;Set WDE masking WDP2, WDP1 & WDP0
  out     WDTCR, Temp           ;Enable Watchdog Timer
  sleep                         ;Go to sleep (wait for External Interrupt)
```

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

Disclaimer: Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Printed on recycled paper.