To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

RENESAS

SH7145 Group

Entering and Exiting the Software Standby Mode

Introduction

Transitions are made to enter and exit software standby mode by using an NMI interrupt. The transition/return status is displayed on a 7-segment LED.

Target Device

SH7145F

Contents

1. Specifications

Transitions are made to enter and exit software standby mode.

During normal program execution, the SH7145 generates an NMI interrupt on detection of a rising edge of the input from a switch that is connected to the NMI pin. After the NMI interrupt processing ends, the switch input detection edge is changed to the falling edge, and a SLEEP instruction is executed to enter software standby mode.

The 7-segment LED indicates "A" during normal program execution. The indication changes to "S" after software standby mode is entered.

In software standby mode, an NMI interrupt is generated on detection of a falling edge of the switch input signal on the NMI pin. This causes the watchdog timer (WDT) to start, and when the WDT overflows, a transition is made from software standby mode to the normal program execution state. After the NMI interrupt processing is complete, the detection edge for the switch input is changed to the rising edge. When a transition is made from software standby mode to normal program execution state, the indication on the 7-segment LED changes from "S" to "A".

In this sample task a bus state controller (BSC) is used for 7-segment LED display. The BSC controls the display on the 7-segment LED, which is connected as an external memory. Figure 1 shows an example connection of a 7-segment LED as an external memory and an NMI switch. Figure 2 shows a 7-segment LED display circuit.

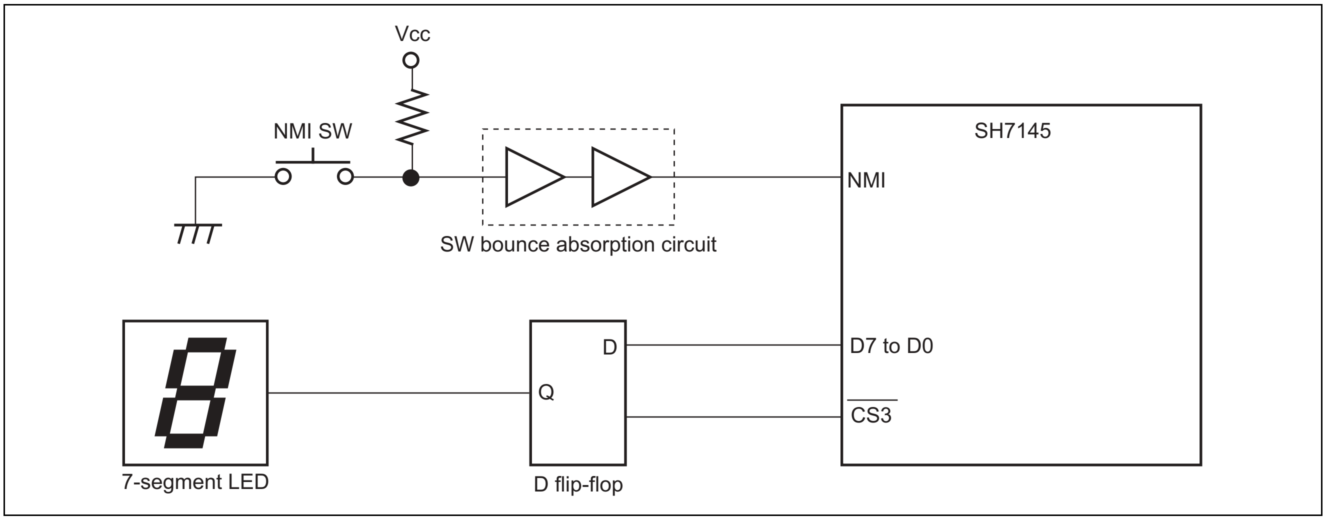


Figure 1 Configuration Diagram

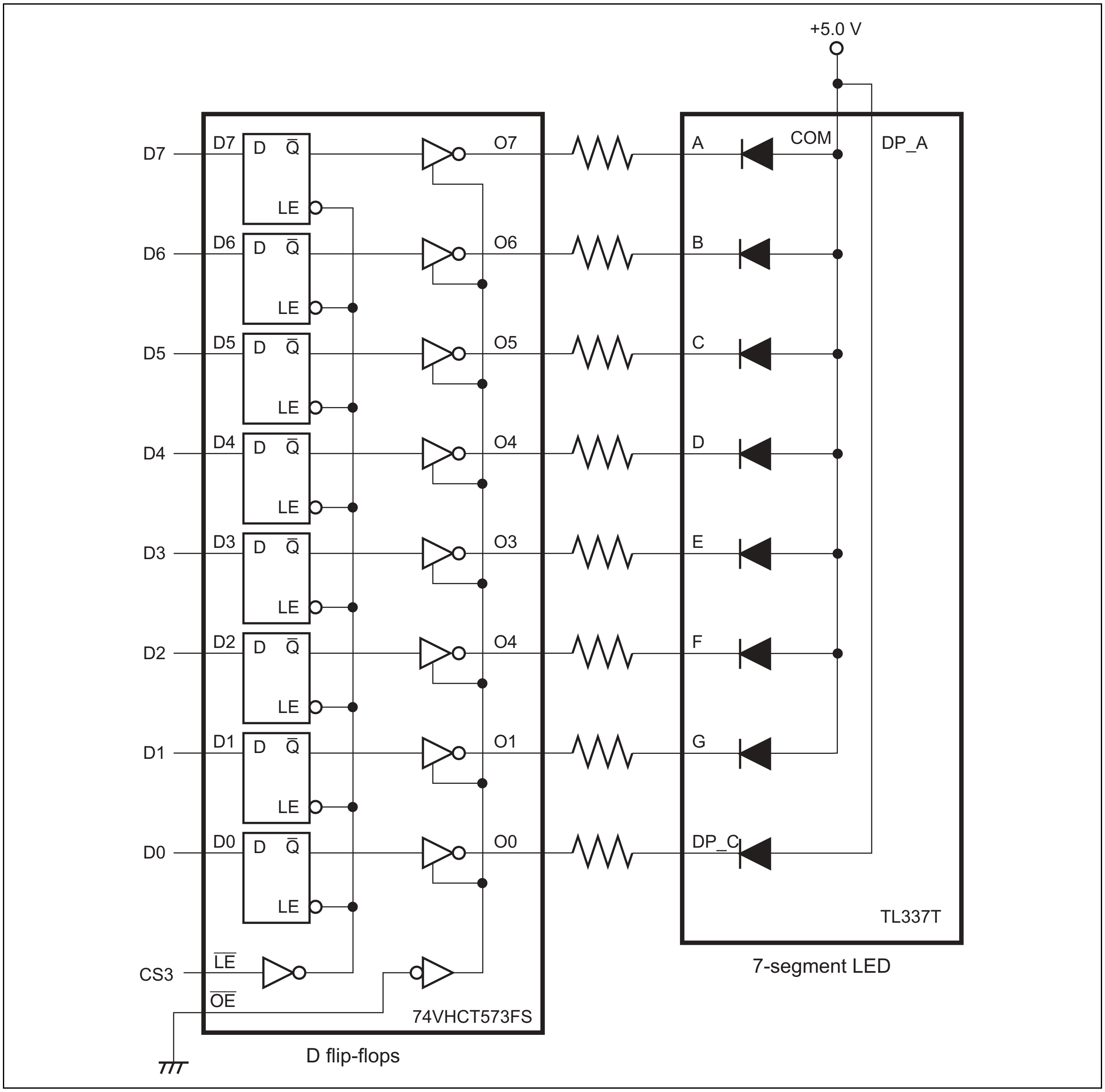


Figure 2 Equivalent Circuit of 7-Segment LED

2. Description of Functions

In this sample task, transitions to and from the software standby mode are made. Figure 3 shows the mode transitions. The following is a description of the functions necessary for transitions to and from the software standby mode.

2.1 Power-Down State

In addition to the normal program execution state, the SH7145 may be in a power-down state, in which the CPU, oscillator, and other functions stop operation to reduce power consumption. The SH7145 can individually control the CPU and the internal peripheral functions to achieve low power consumption. The operating states of the SH7145 include three power-down states: sleep mode, software standby mode, and module standby mode, in addition to the normal operating mode.

A transition to the software standby mode is made when a SLEEP instruction is executed with the SSBY bit in the SBYCR register set to 1. In this mode, all functions are stopped, including the CPU, internal peripheral functions, and the oscillator. However, the contents of CPU registers are retained as long as a prescribed voltage is maintained. This mode, which stops the oscillator, can substantially reduce power consumption.

Operation recovers from software standby mode after the following sequence: when either a falling or rising edge of the NMI pin is detected, oscillation for the clock that is supplied to the watchdog timer (WDT) alone is started; after the WDT overflows, which indicates that the clock has become stable, clocks are supplied to the entire LSI. In this manner, the software standby mode is cancelled, and the SH7145 returns to the normal program execution state. The NMI exception processing then begins.

The standby control register (SBYCR), which is an 8-bit readable/writable register, controls the software standby mode.

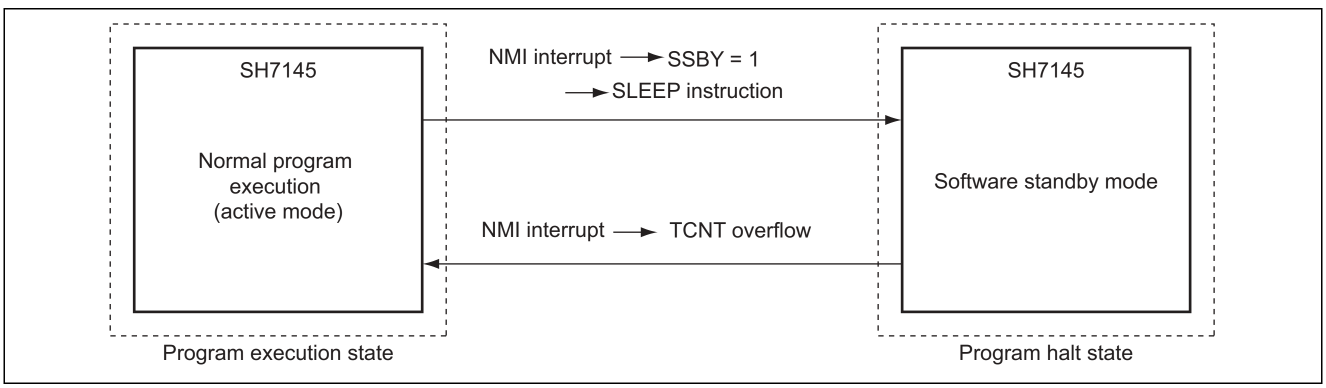
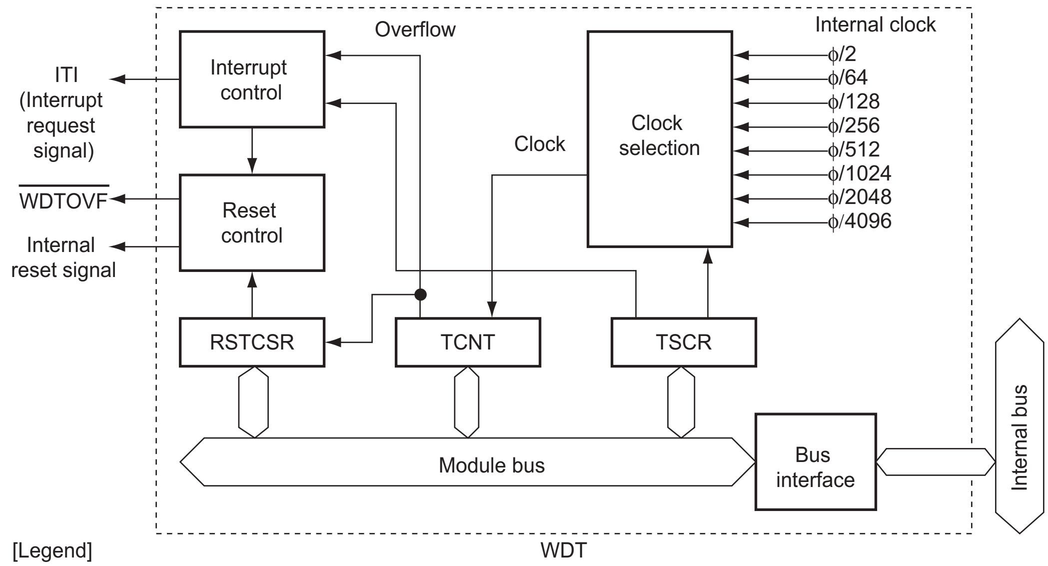


Figure 3 Entering and Exiting Software Standby Mode

2.2 Watchdog Timer (WDT)

The watchdog timer (WDT) is an 8-bit timer; it is used when the software standby mode is cancelled using an NMI interrupt. Figure 4 shows a WDT block diagram.



[Legend]
TCSR: Timer control/status register
TCNT: Timer counter
RSTCSR: Reset control/status register

Note: An internal reset signal can be generated through register settings.
The type of reset is selectable: power-on reset or manual reset.

Figure 4 WDT Block Diagram

Before entering software standby mode, clear the TME bit in the timer control/status register (TCSR) to 0 to stop the WDT. Software standby mode cannot be entered while the TME bit is set to 1. Also, set the CKS2 to CKS0 bits in the TCSR so that the overflow cycle of the timer counter (TCNT) is equal to or greater than the oscillator stabilization time. Table1 shows the oscillator stabilization time.

Table 1 Clock Timing

Conditions: VCC = PLLVcc = 3.3 V ± 0.3 V, VSS = PLVSS = 0 V, Ta = –20 to +75°C (standard)

Item	Minimum Time	Unit
Oscillator stabilization time necessary to return from the standby mode	10	ms

The software standby mode is cancelled in the following sequence: when the NMI signal is input in software standby mode, the oscillator starts operation and the TCNT starts counting up using the clock that was selected by the CKS2 to CKS0 bits before the software standby mode was entered. When the TCNT overflows (H'FF → H'00), it is recognized that the clock has stabilized for use, and clocks are then supplied to the entire LSI. This cancels the software standby mode.

The timer counter (TCNT) is a readable/writable 8-bit up counter. When the timer enable bit (TME) in the timer control/status register (TCSR) is set to 1, the TCNT starts counting up using the internal clock that was selected by the CKS2 to CKS0 bits in the TCSR. The initial value of the TCNT is H'00.

The timer control/status register (TCSR) is an 8-bit readable/writable register that selects the input clock for TCNT, timer mode, etc.

2.3 Interrupt Source

An NMI interrupt is a level-16 interrupt and can always be accepted. The input from the NMI pin is detected by edge, and the detection edge, rising or falling edge, can be selected by setting the NMI edge select bit (NMIE) in the interrupt control register 1 (ICR1) of the interrupt controller (INTC).

The interrupt control register 1 (ICR1), which is a 16-bit register, selects the detection edge for the NMI external interrupt input pin, and indicates the input level on the NMI pin.

Table 2 shows the interrupt operation in transitions to and from the software standby mode in this sample task.

Table 2 NMI Interrupt Operation

State Transition	Interrupt Source	Interrupt Detection	Clock Supply State	Transition Condition
Transition to software standby mode	NMI interrupt	Falling edge of the NMI pin	After transition, the supply of clocks to the entire LSI is stopped.	The WDT must be stopped.
Transition to normal program execution state	NMI interrupt	Rising edge of the NMI pin	Clock is first supplied to the WDT only. After transition, clocks are supplied to the entire LSI.	TCNT of the WDT overflows.

2.4 Bus State Controller (BSC)

The bus state controller (BSC) divedes up the address space and outputs control signals according to the memory type. The use of this controller enables the direct connection of SRAM and ROM to the LSI without requiring additional circuitry. Figure 5 shows a BSC block diagram. In this sample task, the BSC accesses the 7-segment LED as an external memory to control the display on the LED.

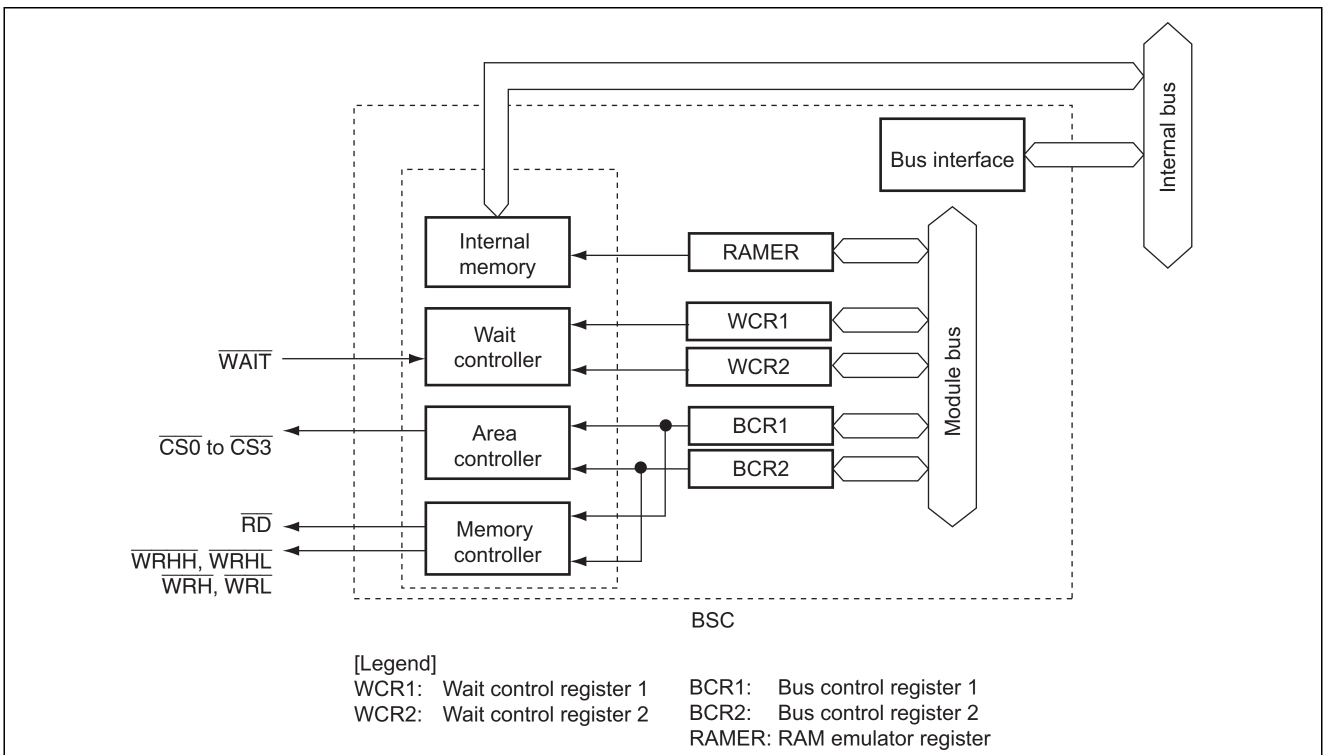


Figure 5 BSC Block diagram

Bus cycles for external space access are performed in two states. Figure 6 shows the basic timing for external space access. For reading, all bits equal to the data bus width of the space (address) to be accessed are latched into the LSI with the timing of the RD signal, irrespective of the size of the operand, and required bytes are internally selected and used. For writing, the byte position at which data is actually written to is specified by the following signals: $\overline{\text{WRHH}}$ (bits 31 to 34), $\overline{\text{WRHL}}$ (bits 23 to 16), $\overline{\text{WRH}}$ (bits 15 to 08), and $\overline{\text{WRL}}$ (bits 7 to 0).

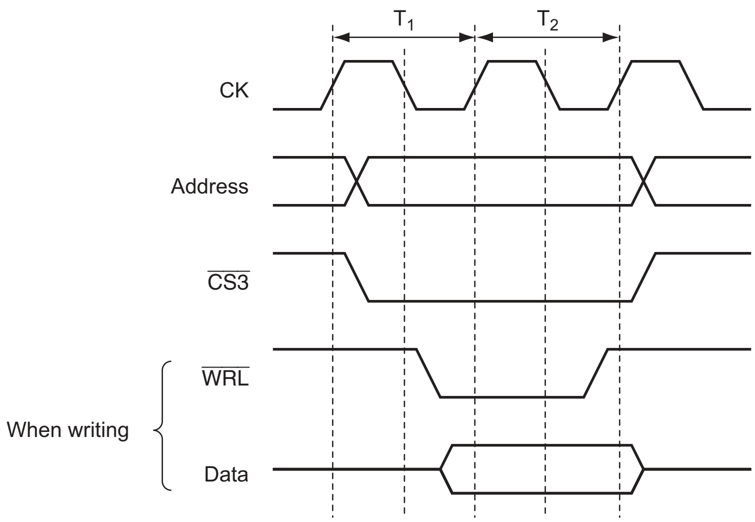


Figure 6 Basic Timing for External Space Access

Table 3 shows memory allocation for the 7-segment LED.

Table 3 Memory Allocation

Area	Address of Allocation	Device Type	Bus Size	Access Wait
CS3	From H'00C00000	7-segment LED	8 bits	None

Figure 7 identifies each segment of the 7-segment LED display. Table 4 is a segment correspondence table. Each segment is lit by negative logic.

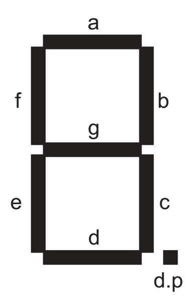


Figure 7 Correspondence of 7-Segment LED Display

Table 4 Segment Correspondence

CS3 Data Bus Name	D7	D6	D5	D4	D3	D2	D1	D0
Segment	dp	g	f	e	d	c	b	a
Character "A" display	1	0	0	0	1	0	0	0
Character "S" display	1	0	0	1	0	0	1	0
Character "." display	0	1	1	1	1	1	1	1

Note: Negative logic lighting (0 = on, 1 = off)

The bus control register 1 (BCR1), which is a 16-bit readable/writable register, is used to enable writing to the MTU control register and to specify a bus size for each CS space. Bits 7 to 0 of the BCR1 should be written to during initialization after a power-on reset, and must not be modified after that. In on-chip ROM enabled mode, the CS spaces should not be accessed until the initialization of the register is complete. In on-chip ROM disabled mode, CS spaces other than space CS0 should not be accessed until the initialization of the register is complete.

Note: In this sample task, the system operates in on-chip ROM enabled mode.

The wait control register 1 (WCR1), which is a 16-bit readable/writable register, specifies the number of wait cycles (0 to15) for each CS space.

2.5 Pin Function Controller (PFC)

The pin function controller (PFC) is comprised of registers for selecting functions of multiplexed pins and their I/O directions.

The port D control registers L1 and L2 (PDCRL1and PDCRL2), which are 16-bit readable/writable registers, select the functions of multiplexed pins of Port D. This sample task selects data bus functions (D0 to D7) to control display on the 7-segment LED.

2.6 Function Assignment

Table 5 shows the assignment of functions in this sample task.

Table 5 Assignment of Functions

Register	Description
SBYCR	Used to make a transition to software standby mode after execution of a SLEEP instruction.
TCSR	Controls the timer counter of the WDT.
ICR1	Sets the input signal detection mode for the external interrupt input pin (NMI).
BCR1	Specifies the bus size for the CS space.
WCR1	Specifies the number of wait cycles for the CS space.
PDCRL1	Selects the functions of port D’s multiplexed pins (port D or data bus).

3. Description of Operation

Figure 8 shows the operation. The LSI enters and exits software standby mode through the hardware and software processing as illustrated in the figure.

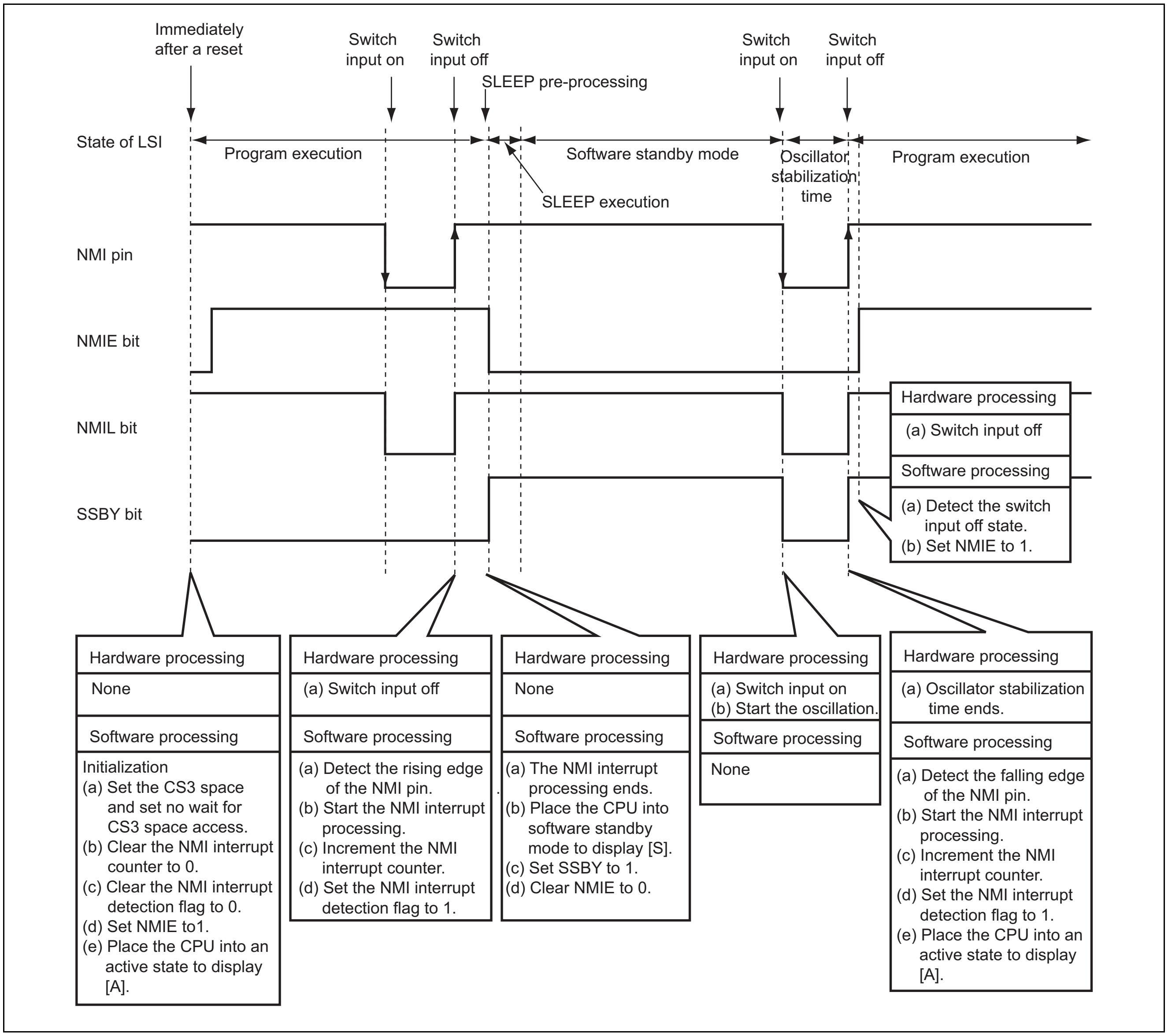


Figure 8 Entering and Exiting Software Standby Mode

4. Description of Software

4.1 Modules

Table 6 describes the modules used in this sample task.

Table 6 Description of Modules

Module Name	Label Name	Functions
Main routine	main	Calls initialization routine, displays the transition state on the 7-segment LED, stops the WDT counter, sets the standby control register for entering software standby mode, sets an NMI interrupt detection edge, makes a transition to software standby mode by SLEEP instruction, and performs initialization after the transition to the normal program execution state.
7-segment LED display processing	led7	Displays on the 7-segment LED.
7-segment LED display data table setting	tblset	Sets the 7-segment LED display data table during initialization.
Initialization	inisub	Sets the access width and wait cycles for the BSC, specifies data bus pin functions by the PFC, initializes the NMI interrupt detection flag and the NMI interrupt counter, selects the NMI interrupt detection edge, and calls the routine for initialization of the 7-segment LED display.
Write processing	wrl	Writes long-type data to a given address.
Wait processing	wait	Performs wait operation
NMI interrupt processing	nmisub	Sets the NMI interrupt detection flag and increments the NMI interrupt counter.

4.2 Arguments

Table 7 shows the arguments used in this sample task.

Table 7 Description of Arguments

Argument Name	Function	Used in	Size	Input/ Output
data	Specifies 7-segment LED display data	7-segment LED display routine	1 byte	Input
dot	Specifies whether to display a dot on the 7-segment LED.	7-segment LED display routine	1 byte	Input

4.3 Internal Registers

Table 8 describes the internal registers used in this sample task.

Table 8 Description of Internal Registers

Register Name		Function	Address	Setting
TCSR	WT/$\overline{IT}$	Timer control/status register (Timer mode select) When WT/$\overline{IT}$ = "0", selects interval timer mode.	H'FFFF8610 Bit 6	0
	TME	Timer control/status register (Timer enable) When TME = "0", the timer is disabled.	H'FFFF8610 Bit 5	0
	CKS2 to CKS0	Timer control/status register (Clock select 2 to 0) When CKS2 = "1", CKS1 = "1", and CKS0 = "0", selects the clock 1/4096 (period: 26.2 ms).	H'FFFF8610 Bit 2 Bit 1, Bit 0	CKS2 = 1 CKS1 = 1 CKS0 = 0
ADCR_0	TRGE	A/D control register_0 (Trigger enable) When TRGE = "0", initiation by trigger is disabled.	H'FFFF8488 Bit 7	0
SBYCR	SSBY	Standby control register (Software standby) When SSBY = "1", software standby mode is entered after execution of a SLEEP instruction.	H'FFFF8614 Bit 7	1
	HIZ	Standby control register (Port high-impedance) When HIZ = "0", pin states are retained during software standby mode.	H'FFFF8614 Bit 6	0
	IRQEH	Standby control register (IRQ7 to IRQ4 enable) When IRQEH = "1", software standby mode cancellation by IRQ7 to IRQ4 is disabled.	H'FFFF8614 Bit 1	1
	IRQEL	Standby control register (IRQ3 to IRQ0 enable) When IRQEL = "1", software standby mode cancellation by IRQ3 to IRQ0 is disabled.	H'FFFF8614 Bit 0	1
ICR1	NMIL	Interrupt control register 1 (NMI input level) When NMIL = "0", a low level is input to the NMI pin. When NMIL = "1", a high level is input to the NMI pin.	H'FFF8358 Bit 15	—
	NMIE	Interrupt control register 1 (NMI edge select) When NMIE = "0", interrupt requests are detected on the falling edge of the NMI input. When NMIE = "1", interrupt requests are detected on the rising edge of the NMI input.	H'FFF8358 Bit 8	Falling edge: 0 Rising edge: 1

Register Name		Function	Address	Setting
BCR1	A3LG	Bus control register 1 (CS3 space long size specification) When A3LG = "0", the bus size is according to the value of the A3SZ bit of this register.	H'FFF8620 Bit 7	0
	A3SZ	Bus control register 1 (CS3 space size specification) When A3SZ = "0", byte size (8 bits) is selected.	H'FFF8620 Bit 3	0
WCR1	W33 W32 W31 W30	Wait control register 1 (CS3 and CS7 space wait specification) When W33 = "0", W32 = "0", W31 = "0", and W30 = "0", no wait cycle is inserted (external wait input disabled).	H'FFF8624 Bit 15 Bit 14 Bit 13 Bit 12	W33 = 0 W32 = 0 W31 = 0 W30 = 0
PDCRL1	PD7MD	Port D control register L1 (PD7 mode bit) When PD7MD = "1", D7 I/O pin function (BSC) is selected.	H'FFFF83AD Bit 7	PD7MD = 1
PDCRL1	PD6MD	Port D control register L1 (PD6 mode bit) When PD6MD = "1", D6 I/O pin function (BSC) is selected.	H'FFFF83AD Bit 6	PD6MD = 1
PDCRL1	PD5MD	Port D control register L1 (PD5 mode bit) When PD5MD = "1", D5 I/O pin function (BSC) is selected.	H'FFFF83AD Bit 5	PD5MD = 1
PDCRL1	PD4MD	Port D control register L1 (PD4 mode bit) When PD4MD = "1", D4 I/O pin function (BSC) is selected.	H'FFFF83AD Bit 4	PD4MD = 1
PDCRL1	PD3MD	Port D control register L1 (PD3 mode bit) When PD3MD = "1", D3 I/O pin function (BSC) is selected.	H'FFFF83AD Bit 3	PD3MD = 1
PDCRL1	PD2MD	Port D control register L1 (PD2 mode bit) When PD2MD = "1", D2 I/O pin function (BSC) is selected.	H'FFFF83AD Bit 2	PD2MD = 1
PDCRL1	PD1MD	Port D control register L1 (PD1 mode bit) When PD1MD = "1", D1 I/O pin function (BSC) is selected.	H'FFFF83AD Bit 1	PD1MD = 1
PDCRL1	PD0MD	Port D control register L1 (PD0 mode bit) When PD0MD = "1", D0 I/O pin function (BSC) is selected.	H'FFFF83AD Bit 0	PD0MD = 1

4.4 RAM Usage

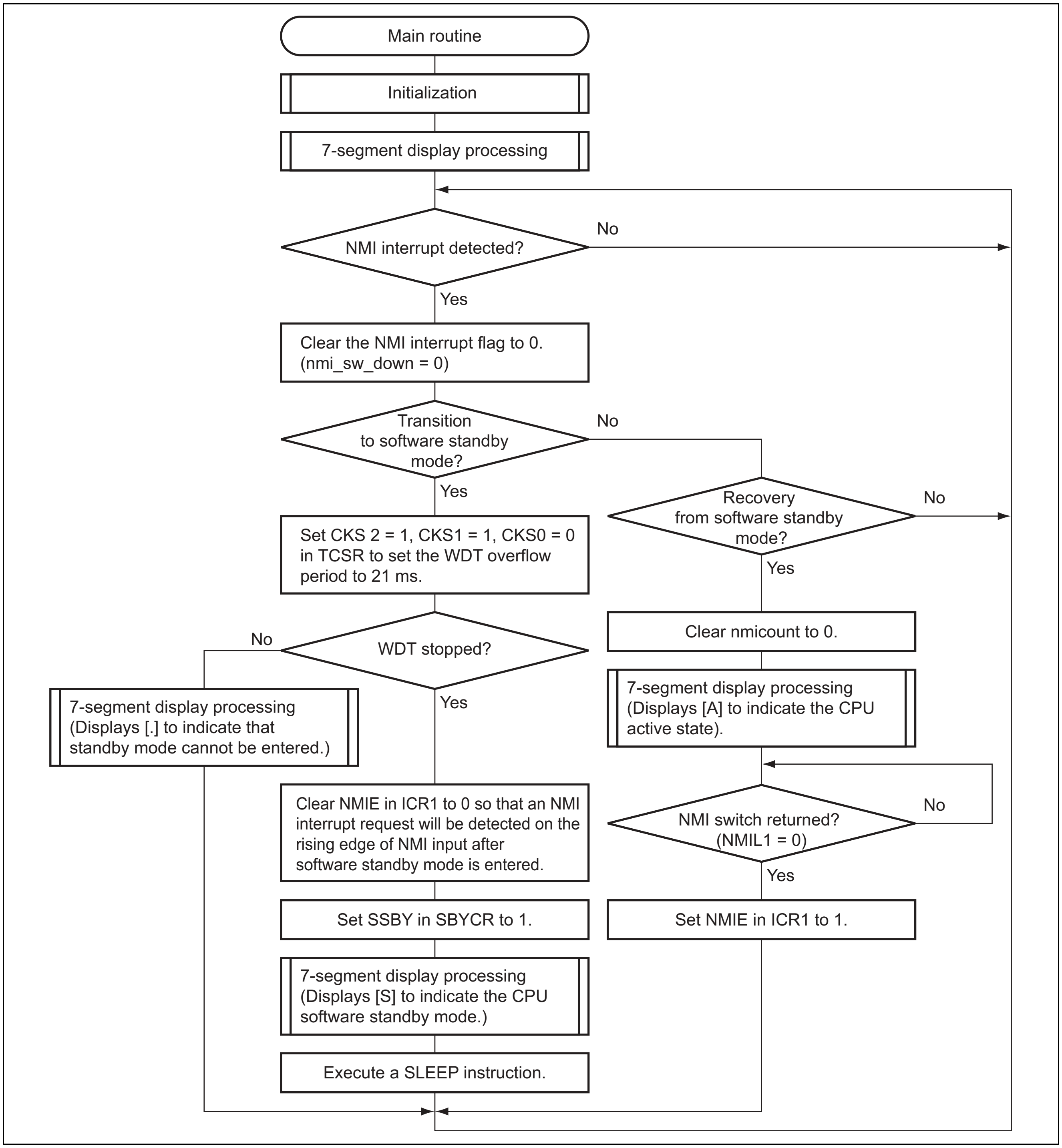
Table 9 describes the RAM usage in this sample task.

Table 9 Description of RAM

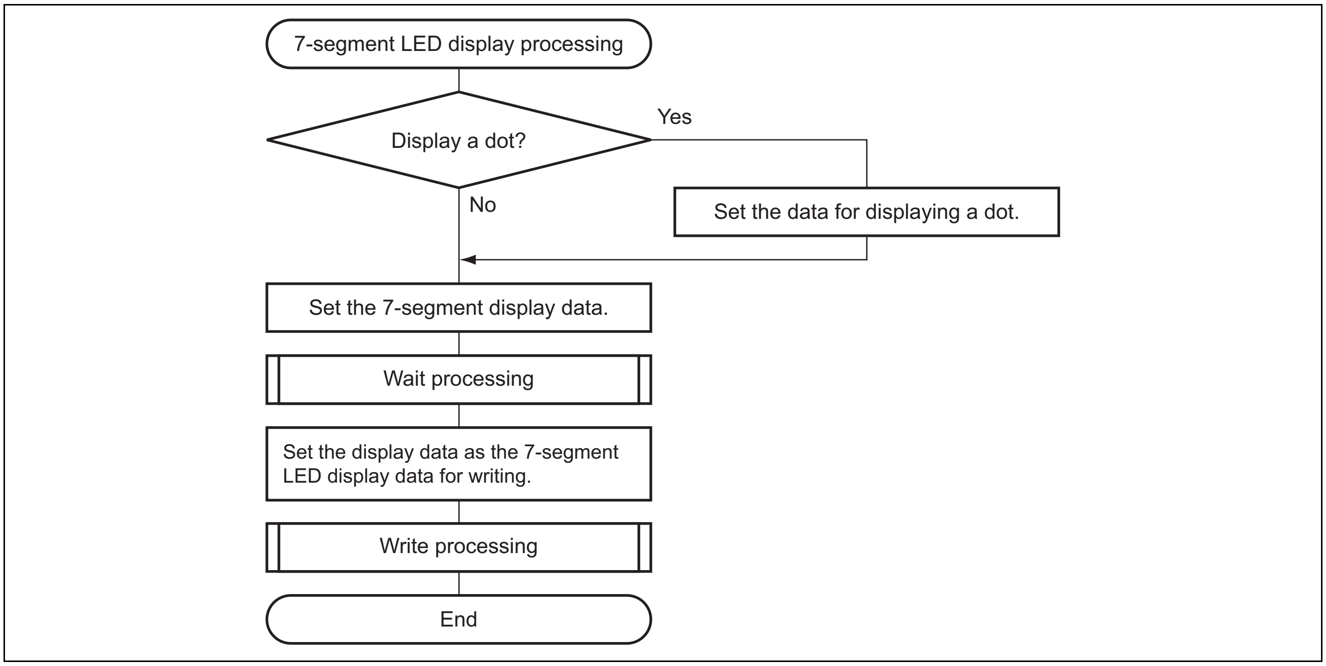
Label Name	Description	Address	Used in
table[0]	Stores the data for displaying "A".	H'00407005	7-segment LED display routine, 7-segment LED display data table setting routine
table[1]	Stores the data for displaying "S".	H'00407006	7-segment LED display routine, 7-segment LED display data table setting routine
table[2]	Stores the data for displaying ". ".	H'00407007	7-segment LED display routine, 7-segment LED display data table setting routine
nmicount	Stores the NMI interrupt count data.	H'00407000	Main routine, initialization routine
nmi_sw_down	Stores the NMI interrupt detection flag data.	H'00407004	Main routine, initialization routine

5. Flowchart

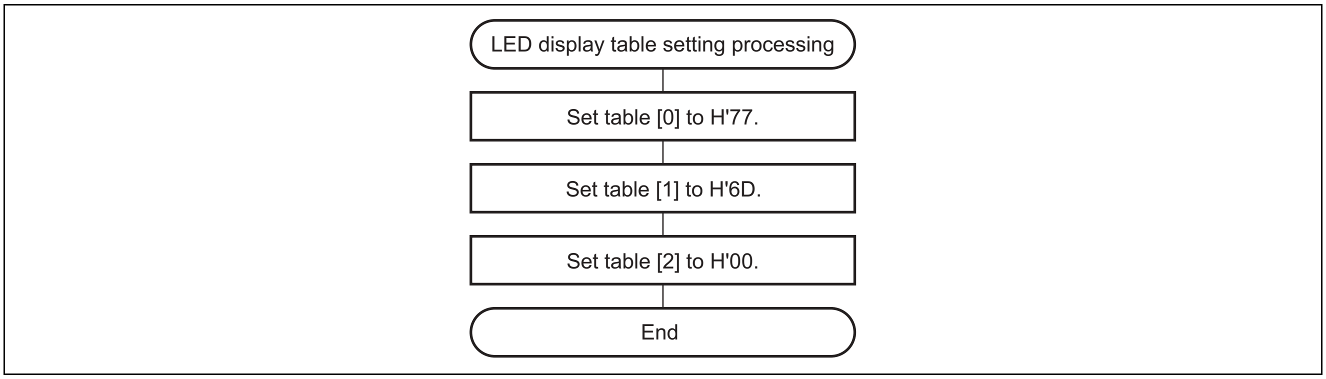
5.1 Main Routine



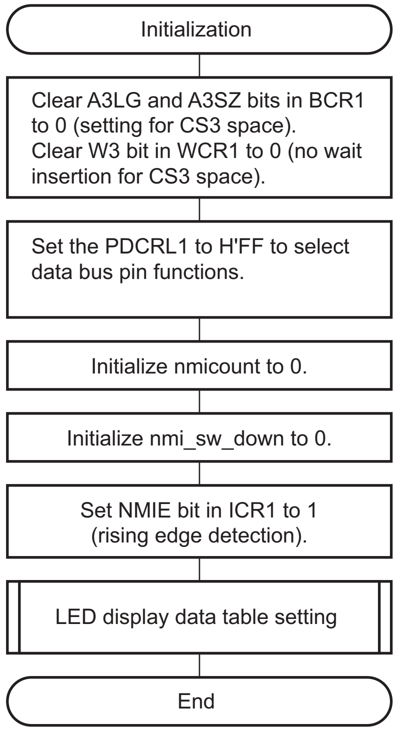
5.2 7-Segment LED Display Processing



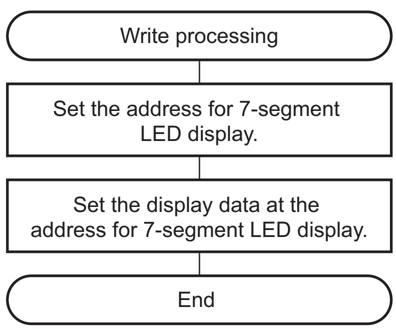
5.3 7-Segment LED Display Data Table Setting Processing



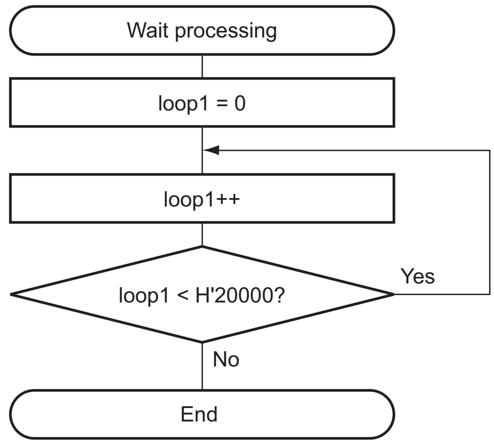
5.4 Initialization Processing



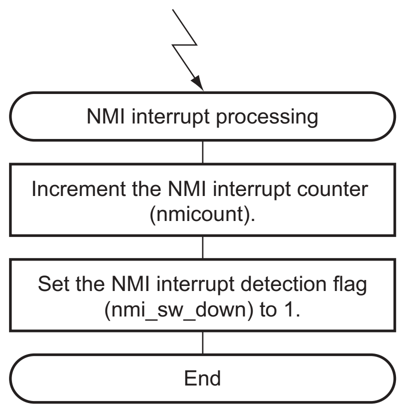
5.5 Write Processing



5.6 Wait Processing



5.7 NMI Interrupt Processing



6. Program Listing

INIT.C (program listing)

```
extern  _INITSCT(void) ;
extern  void main(void) ;

void    _INIT()
{
    _INITSCT() ;
    main() ;
    for(;;) ;
}
```

VEC_TBL.SRC (program listing)

```
.SECTION   VECT,DATA,LOCATE=H'0000

.IMPORT    __INIT
.IMPORT    __STACK
.IMPORT    _nmisub

.ORG        H'00000000
.DATA.L    __INIT
.DATA.L    __STACK
.DATA.L    __INIT
.DATA.L    __STACK

.ORG        H'0000002c
.DATA.L    _nmisub
.END
```

Standby.c (program listing)

```
/*******************************************************************************************/
/* SH7145 Series -SH7145- Application note                                                 */
/*                                                                                         */
/* Application Note                                                                        */
/* Transition To Software Standby Mode And Return From Software                            */
/* Standby Mode By NMI Interrupt                                                           */
/*                                                                                         */
/* Function : Transition To Software Standby Mode                                          */
/*            And Return From Software Standby Mode By                                     */
/*            NMI Interrupt                                                                */
/*                                                                                         */
/* Peripheral Clock     : 25MHz                                                            */
/* Internal Clock       : 50MHz                                                            */
/*******************************************************************************************/

/*******************************************************************************************/
/*  Include File                                                                           */
/*******************************************************************************************/
#include <machine.h>
#include "IODEFINE.H"
```

```
/*************************************************************************************/
/*  Function Prototype                                                               */
/*************************************************************************************/
void inisub(void);                                  /* Initialize configuration      */
void wrl(unsigned long addr, unsigned long data);   /* Write Function (Long)         */
void led7(unsigned char data,unsigned char dot);    /* 7 Segment LED Display Function */
void tblset(void);                                  /* Set 7 Segment LED Display Table Function */
void wait(void);                                    /* Wait Function                 */
void main(void);                                    /* Control Software Standby Mode Function */
#pragma interrupt(nmisub)                           /* NMI Interrupt Handler Function */

/*************************************************************************************/
/*  RAM Allocation                                                                   */
/*************************************************************************************/
unsigned long nmicount;                             /* NMI Interrupt Counter         */
unsigned char nmi_sw_down;                          /* NMI Switch                    */
unsigned char table[4];                             /* 7 Segment LED Display Data Table */

/*************************************************************************************/
/*  I/O Definition                                                                   */
/*************************************************************************************/
#define LED    0x00C00000                           /* LED Port                      */

/*************************************************************************************/
/*  Function Definition (Main Program)                                               */
/*************************************************************************************/
void main(void)
{

    inisub();                                       /* Initialize Configuration      */
    led7(0,0);                                      /* Output "A" on 7 Segment LED Display */

    while(1){                                       /* Check Software Standby Mode Status */
        if(nmi_sw_down == 1){                       /* NMI Switch is Down?           */
           nmi_sw_down = 0;                         /* Clear NMI Switch Down Flag    */
           if(nmicount == 1){                       /* NMI Interrupt Counter is 1?   */

                /*****************************************************************/
                /* Set WDT Overflow Cycles by TCSR Register                       */
                /* Cycles = (CPU Clock/4096) * 256(TCNT Clock) = 21[ms]           */
                /*****************************************************************/
                WDT.WRITE.TCSR = 0x1e;
                                                    /* WDT Timer Counter is Stopped, */
                                                    /*           Set WDT Overflow Cycles */

                if(WDT.READ.TCSR.BIT.TME == 0){     /* WDT Timer Counter is Stopped? */
                    INTC.ICR1.BIT.NMIE = 0x0;       /* NMI Falling Edge Interrupt    */
                    SBYCR.BYTE = 0x9f;              /* Set SSBY Bit                  */
                    led7(1,0);                      /* Output "S" on 7 Segment LED Display */
                    sleep();                        /* Transition To Software Standby Mode */
                }else {                             /* WDT Timer Counter is Working  */
                    led7(2,1);                      /* Output "." on 7 Segment LED Display */
                }
        }else if(nmicount == 2){                    /* NMI Interrupt Counter is 2?   */
```

```
                nmicount = 0;                     /* Clear NMI Interrupt Counter          */
                led7(0,0);                        /* Output "A" on 7 Segment LED Display      */
                while( INTC.ICR1.BIT.NMIL == 0x0); /* Check NMI Switch Release          */
                INTC.ICR1.BIT.NMIE = 0x1;
                                                  /* NMI Rising Edge Interrupt,           */
                                                  /*        Return From Software Standby Mode */
            }
        }
    }
}
/*************************************************************************************************/
/*  Initial Configuration                                                                        */
/*************************************************************************************************/
void inisub(void)
{
    BSC.BCR1.BIT.A3LG = 0x0;                      /* Clear A3LG Bit for A3SZ Bit           */
    BSC.BCR1.BIT.A3SZ = 0x0;                      /* Byte Size CS3 Memory Area             */
    BSC.WCR1.BIT.W3   = 0x0;                      /* No Wait CS3 Memory Area               */

    PFC.PDCRL1.BYTE.L = 0xff;                     /* Use From D0-7 Input And Output(BSC) Port */
    PFC.PDCRL2.BYTE.L = 0x0;                      /* Use From D0-7 Input And Output(BSC) Port */

    nmicount = 0;                                 /* Initialize NMI Interrupt Counter      */
    nmi_sw_down = 0;                              /* Initialize NMI Switch                 */
    INTC.ICR1.BIT.NMIE = 0x1;                     /* Set NMI Rising Edge Interrupt         */
    tblset();                                     /* Set 7 Segment LED Display Table       */

}
/*************************************************************************************************/
/*  Write Function                                                                               */
/*************************************************************************************************/
/* Write Function (Long) */
void wrl(unsigned long addr, unsigned long data)
{
    unsigned long *paddr = ((unsigned long *) addr);
                                                  /* Set paddr as LED Port                 */
    *paddr = data;                                /* Save data                             */
}
```

```
/*****************************************************************************************/
/*  7 Segment LED Display                                                               */
/*****************************************************************************************/
void led7(unsigned char data,unsigned char dot)
{
    unsigned char leddata;
    unsigned long wleddata;

    if(dot!=0){                                 /* Output "." on 7 Segment LED Display?     */
        leddata=(table[data]|0x80);             /* Output "."                               */
    } else {
        leddata=(table[data]&0x7f);             /* 7 Segment LED Display Output Value       */
    }

    wait();                                     /* Wait                                     */

    wleddata = leddata;                         /* Save 7 Segment LED Display Output Value  */
    wrl(LED,~wleddata);                         /* Output 7 Seg LED Display                 */
}


/*****************************************************************************************/
/*  Set 7 Segment LED Display Table                                                     */
/*****************************************************************************************/
void tblset(void)
{
    table[0]=0x77;                              /* 7 Segment LED Display Data 'A'           */
    table[1]=0x6d;                              /* 7 Segment LED Display Data 'S'           */
    table[2]=0x00;                              /* 7 Segment LED Display Data ' '           */
}

/*****************************************************************************************/
/*  NMI Interrupt Function                                                              */
/*****************************************************************************************/
void nmisub(void)
{
    nmicount++;                                 /* Increment NMI Interrupt Counter          */
    nmi_sw_down = 1;                            /* Set NMI Switch Down                      */
}

/*****************************************************************************************/
/*  Wait Function                                                                       */
/*****************************************************************************************/
void wait(void)
{
    unsigned long loop1;
    for(loop1=0;loop1<0x20000;loop1++){         /* Wait                                     */
    }
}
```

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Sep.16.04	—	First edition issued