H8/3867 Series

Application Note

HITACHI

ADE-502-068
Rev. 1.0
8/30/1999
Hitachi, Ltd.

Cautions

Preface

The H8/300L Series of single-chip microcomputers are based on the high-speed H8/300L CPU, and integrate all peripheral functions necessary for system configuration.

The H8/300L CPU uses an instruction set which is compatible with the H8/300 CPU.

The H8/3867 Series and H8/3827 Series are provided with such peripheral functions for system configuration as an LCD controller/driver, six different timers, a 14-bit pulse width modulator (PWM), a two-channel serial communication interface, and an A/D converter. These models can be used as microcomputers for embedded systems where LCD display is required.

The H8/3867 Series models are equipped with a booster constant-voltage (5 V) power supply as an LCD driver power supply, providing a constant 5 V regardless of V_{CC}.

These H8/3867 Series application notes include a “Basic Operation” section with operation examples when using the built-in peripheral functions of the H8/3867 Series independently. They are provided in the hope that they will be of use for software and hardware design.

Operation of the programs and circuits described in these application notes has been verified, but their operation should be confirmed by the user as well before actually being used.

Contents

Section 1 Guide to Using the H8/3867 Series Application Notes

These application notes consist of two sections, as follows.

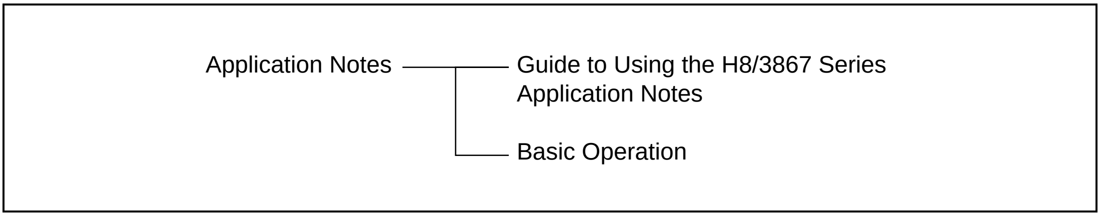


Figure 1 Contents of these Application Notes

Guide to Using the H8/3867 Series Application Notes

Explains how to use the H8/3867 Series application notes.

Basic Operation

Explains how to use the built-in peripheral functions of the H8/3867 Series through simple task examples.

1.1 Contents of Basic Operation

Basic Operation includes the sections shown below, explaining use of the built-in peripheral functions.

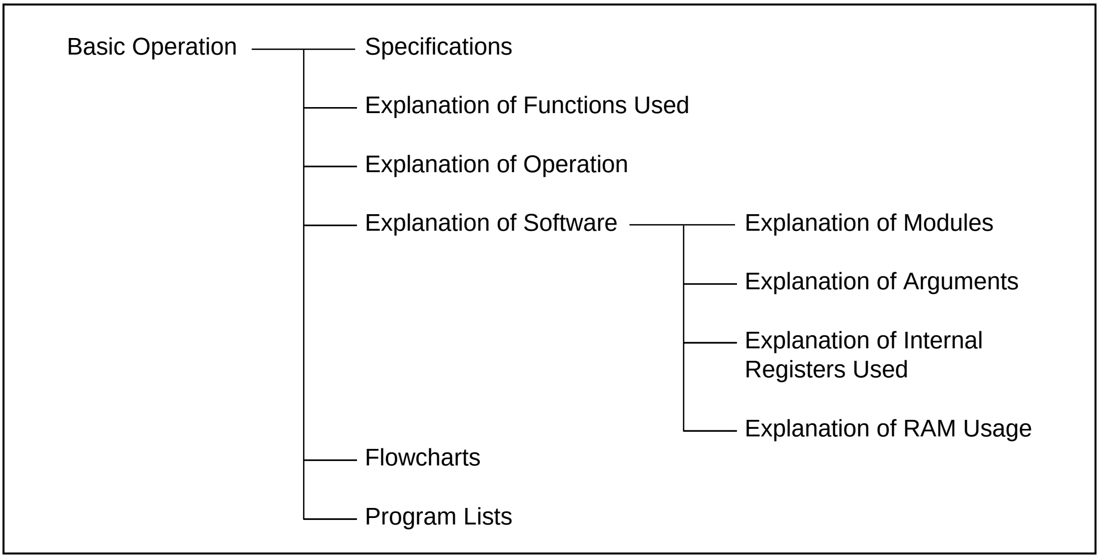


Figure 2 Contents of Basic Operation

Specifications

Explains system specifications for task examples.

Explanation of Functions Used

Explains the features of peripheral functions used in task examples, and allocation of the peripheral functions.

Explanation of Operation

Explains operation of task examples using timing charts.

Explanation of Software

1. Explanation of Modules
   Explains the software modules used for operation in task examples.

2. Explanation of Arguments
   Explains input arguments necessary for module execution, and arguments output following execution.

3. Explanation of Internal Registers Used
   Explains internal registers such as a timer control register and serial mode register of peripheral functions used in modules.

4. Explanation of RAM Usage
   Explains RAM label names and functions used in modules.

Flowcharts

Uses flowcharts to explain the software executed in task examples.

Program Lists

Gives program lists for software executed in task examples.

Section 2 Basic Operation

2.1 Internal Power Supply Step-Down Circuit Settings

Internal Power Supply Step-Down Circuit Settings	MCU: H8/3867 Series	Functions Used: Internal Power Supply Step-Down Circuit

Usage

The H8/3867 Series incorporates an internal power supply step-down circuit. Below the features and usage of the internal power supply step-down circuit are explained, together with important notes and the power supply voltage and operating range.

Features of the internal power supply step-down circuit

1. By using the internal power supply step-down circuit, the internal power supply voltage can be held constant at approximately 1.5 V without depending on the voltage of the power supply connected to the external V_{CC} pin.

2. Current consumed when an external power supply at 1.8 V or higher is used can be held to approximately the same low current as at 1.5 V.

3. It is also possible to use the same level of an external power supply voltage and internal power supply voltage, without using the internal power supply step-down circuit.

Power supply connection when using the internal power supply step-down circuit

An external power supply is connected to the V_{CC} pin as shown in figure 1, and a capacitance of approximately 1 µF is inserted between CV_{CC} and V_{SS}. By adding this external circuit, the internal step-down circuit becomes operative.

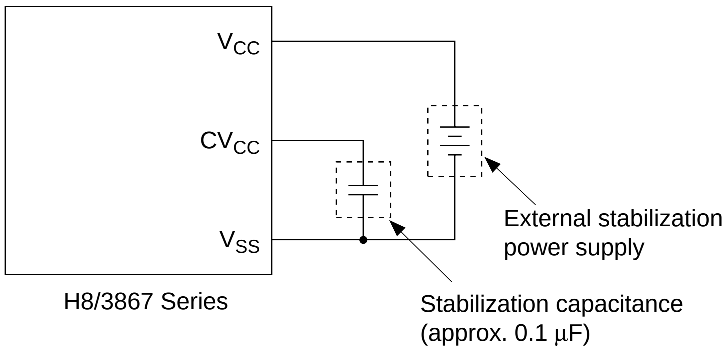


Figure 1 Power Supply Connection When Using the Internal Power Supply Step-down Circuit

Notes on operation using the internal power supply step-down circuit

1. The interface to the external circuit uses as reference levels the voltage of the power supply connected to the V_{CC} pin and the level of the ground connected to the V_{SS} pin. For example, the high and low port input/output levels become the V_{CC} level and the V_{SS} level, respectively.

2. When the internal power supply step-down circuit is used, the operating frequency f_{osc} range is, for a V_{CC} of 2.2 to 5.5 V, f_{osc} = 0.4 MHz to 2 MHz; otherwise, it is f_{osc} = 0.4 MHz to 1 MHz.

3. The LCD power supply and A/D converter analog power supply are not affected by internal step-down processing.

Power supply connection when not using the internal power supply step-down circuit

The external power supply is connected across the V_{CC} and CV_{CC} pins, as shown in figure 2. The external power supply is input directly to the internal power supply circuit.

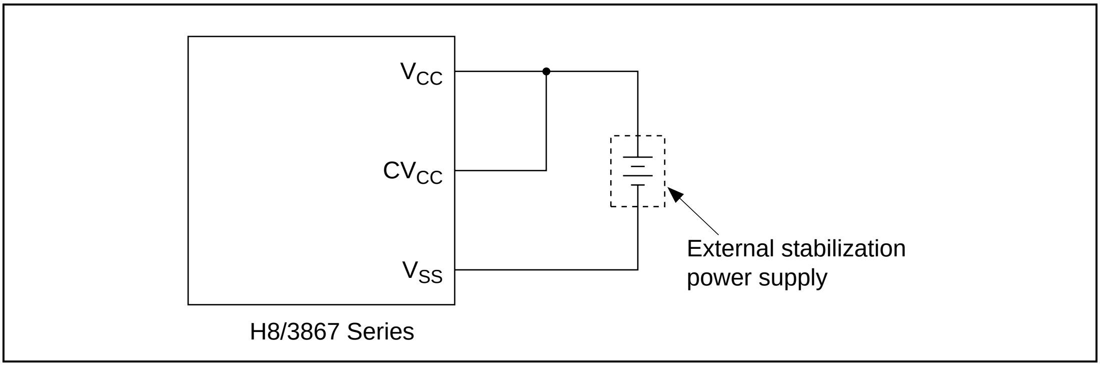


Figure 2 Power Supply Connection When Not Using the Internal Power Supply Step-down Circuit

Note on operation not using the internal power supply step-down circuit

Power supply voltages between 1.8 V and 5.5 V can be used. Operation cannot be guaranteed if a voltage outside this range (less than 1.8 V or more than 5.5 V) is input.

Power supply voltage and oscillator frequency ranges

Figure 3 shows the ranges of the power supply voltage and the oscillator frequency (shaded regions).

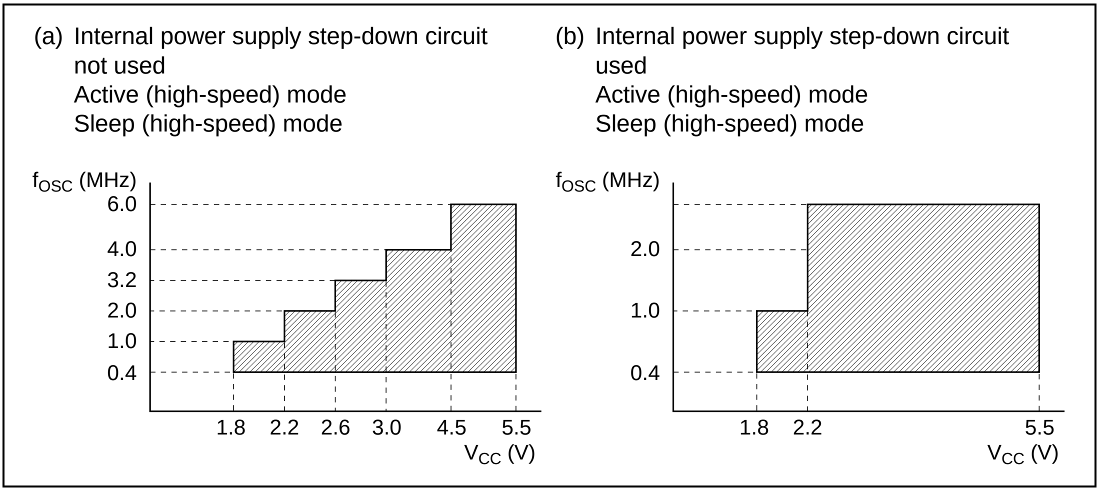


Figure 3 Power Supply Voltage and Oscillator Frequency Ranges

Power supply voltage and operating frequency ranges

Figure 4 shows the ranges of the power supply voltage and operating frequency (shaded regions).

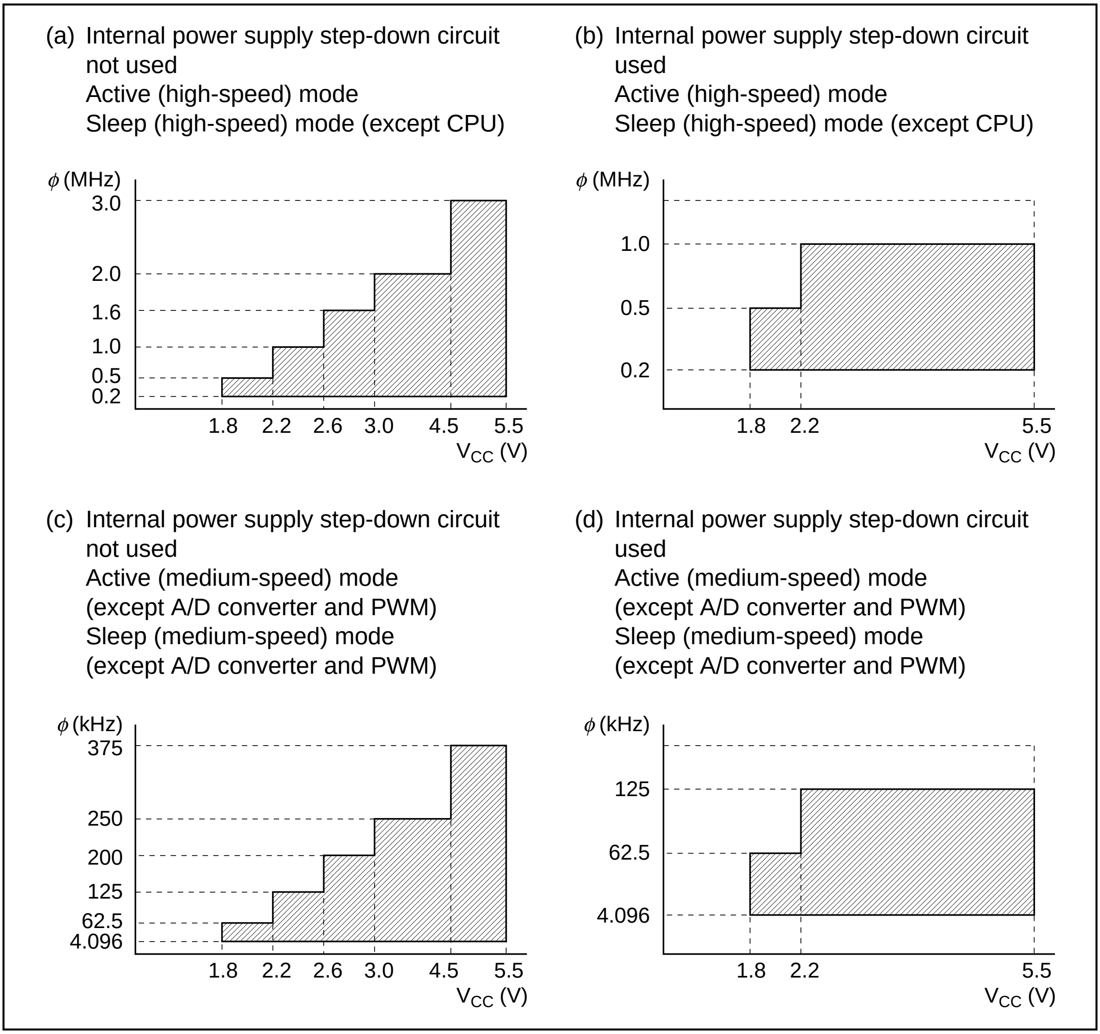


Figure 4 Power Supply Voltage and Operating Frequency Ranges

2.2 Asynchronous Event Counter Operation

Asynchronous Event Counter Operation	MCU: H8/3867 Series	Functions Used: Asynchronous Event Counter (AEC)

Specifications

1. Using an asynchronous event counter, once every 524.288 ms there is a transition from subactive mode to active (high-speed) mode, reversal of the port output in active (high-speed) mode, and a transition back to subactive mode.

2. The 2-MHz event input is applied to the asynchronous event input L pin (AEVL).

3. In this task example, the circuit is used as a 16-bit asynchronous event counter.

Explanation of Functions Used

1. In this task example, an asynchronous event counter (AEC) is used to induce transitions between subactive and active modes and to invert the port output. The features of the AEC are as follows.

   - Input external events can be counted asynchronously, independently of basic clock operation.
   - The counter has a 16-bit configuration, and can count up to 65,536 events.
   - The circuit can also be used as two independent 8-bit event counter channels.
   - The counter can be reset or halted under software control.
   - Event counter overflow can be detected to automatically generate an interrupt.
   - A module standby mode can be employed to set standby mode in module units when not in use.

2. Figure 1 is a block diagram of the 16-bit asynchronous event counter used in this task example.

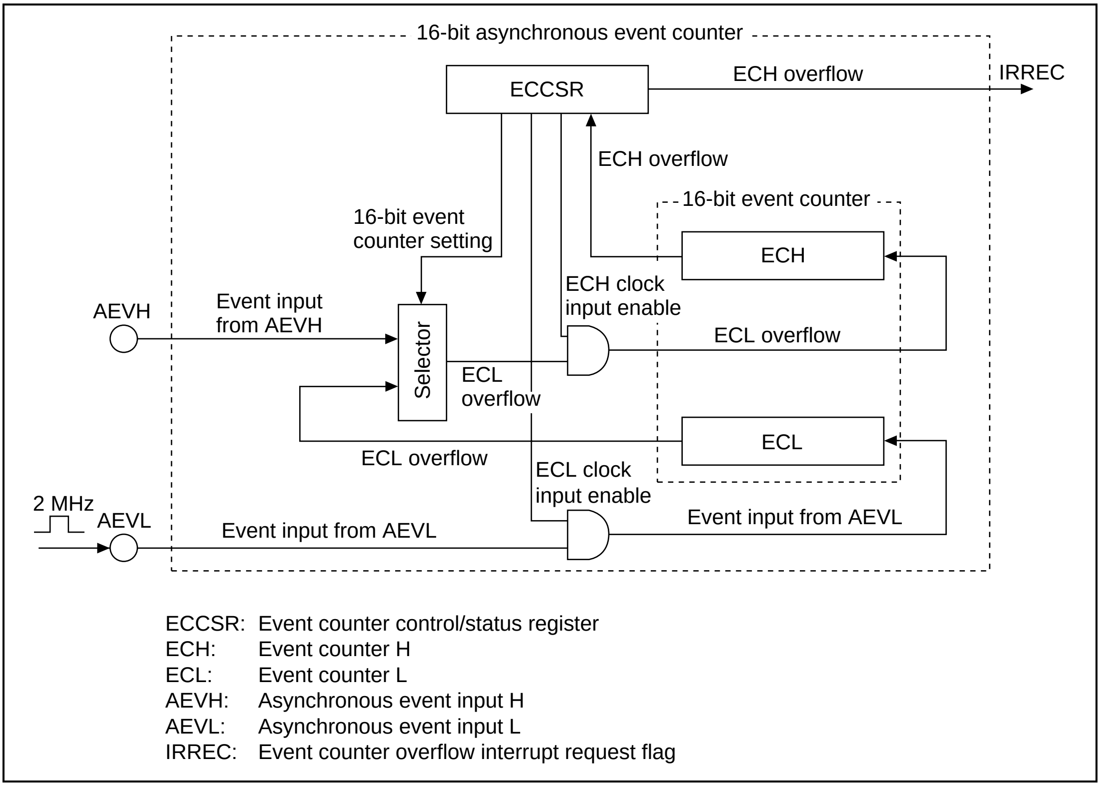


Figure 1 Block Diagram of Asynchronous Event Counter

3. Functions of the 16-bit asynchronous event counter are explained in table 1 below.

Table 1 Functions of the 16-bit Asynchronous Event Counter

Event counter control/status register (ECCSR)	
Function	ECCSR is an 8-bit read/write register which is used to detect counter overflow, reset the counter, and halt counting-up operation. Upon reset, ECCSR is initialized to H'00.
Event counter H (ECH)	
Function	ECH is an 8-bit readable up-counter which operates either as an independent 8-bit event counter, or, in combination with ECL, as the counter for the upper eight bits of a 16-bit event counter. As the input clock signal, either the external asynchronous event AEVH pin, or the overflow signal from the lower 8-bit counter ECH can be selected by the CH2 bit of ECCSR. ECH can be cleared to H'00 by software. Upon reset, ECH is initialized to H'00.
Event counter L (ECL)	
Function	ECL is an 8-bit readable up-counter which operates either as an independent 8-bit event counter, or, in combination with ECH, as the counter for the lower eight bits of a 16-bit event counter. As the input clock signal, the event clock from the external asynchronous event AEVL pin is used by the CH2 bit of ECCSR. ECL can be cleared to H'00 by software. Upon reset, ECL is initialized to H'00.
Asynchronous event input H (AEVH)	
Function	AEVH is the event input pin for input to the event counter H (ECH).
Asynchronous event input L (AEVL)	
Function	AEVL is the event input pin for input to the event counter L (ECL).
Asynchronous event counter interrupt request flag (IRREC)	
Function	When an asynchronous event counter interrupt request occurs, IRREC is set to 1. Even when the interrupt is accepted, IRREC is not automatically cleared. To clear IRREC, use software to write 0.
Asynchronous event counter interrupt enable (IENEC)	
Function	Enables or disables asynchronous event counter interrupt requests.

4. Figure 2 shows an example of settings when using the circuit as a 16-bit asynchronous event counter.

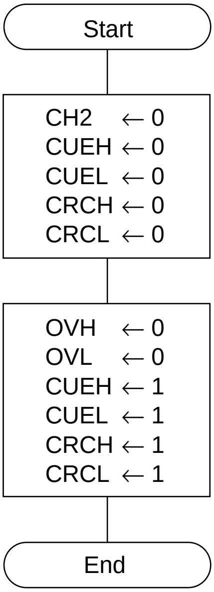


Figure 2 Example of Settings for 16-bit Asynchronous Event Counter

Upon reset, CH2 is cleared to 0, so that after reset ECH and ECL operate as a 16-bit event counter. In addition, the circuit will also operate as a 16-bit event counter by using the settings shown in figure 2. The operating clock source is the asynchronous event input from the AEVL pin. When the next clock pulse is input after the count values for both ECH and ECL reach H'FF, ECH and ECL overflow, the OVH flag of ESSSR is set to 1, the count values of ECH and ECL are both returned to H'00, and counting-up is restarted. Upon occurrence of overflow, the IRREC bit of IRR2 is set to 1. At this time, if the IENEC bit of IENR2 is 1, an interrupt request is sent to the CPU.

5. Asynchronous event counter operating modes are indicated in table 2.

Table 2 Asynchronous Event Counter Operating Modes

Operating Mode	Reset	Active	Sleep	Watch	Subactive	Subsleep	Standby	Module Standby
ECCSR	Reset	Functions	Functions	Held*	Functions	Functions	Held*	Held
ECH	Reset	Functions	Functions*	Functions*	Functions	Functions	Functions*	Halted
ECL	Reset	Functions	Functions*	Functions*	Functions	Functions	Functions*	Halted

Note: * When an asynchronous external event is input, the counter is incremented, but the count overflow H/L flags are not affected.

6. Notes on the 16-bit asynchronous event counter

   a. Before reading the values of ECH and ECL, the CUEH and CUEL bits of ECCSR are cleared to 0, to prevent asynchronous events from being input to the counter. If the counter is incremented during reading, the correct value cannot be read. When clearing the CUEH and CUEL bits of ECCSR to 0, ECH and ECL may each be incremented by one.

   b. When the internal power supply step-down circuit is not being used, the maximum clock frequency for input to the AEVH and AEVL pins is 6 MHz when V_{CC} is 4.5 to 5.5 V, is 4 MHz when V_{CC} is 3.0 to 5.5 V, and is 3.2 MHz when V_{CC} is 2.6 to 5.5 V. When the internal power supply step-down circuit is being used or not being used, the maximum clock frequency is 2 MHz when V_{CC} is 2.2 to 5.5 V, and otherwise is 1 MHz. In addition, the clock high and low widths should be a minimum of 83 ns.

   c. When the AEC is used in 16-bit mode, either the CUEH bit in ECCSR should be set to 1 and then CRCH set to 1, or else after CUEH and CRCH are set simultaneously the clock pulse should be input. Thereafter, the value of CUEH should not be modified during use in 16-bit mode. If, while in 16-bit mode, CUEH is changed, ECH may be erroneously incremented.

   d. Table 3 shows operating modes and event input frequencies.

Table 3 Relation between Operating Modes and AEVH/AEVL Pin Event Input Frequencies

Mode		Maximum AEVH/AEVL Pin Input Clock Frequency
16-bit mode		Internal step-down circuit not used:
8-bit mode Active (high-speed), Sleep (high-speed)		V_{CC} = 4.5 to 5.5 V/6 MHz V_{CC} = 3.0 to 5.5 V/8 MHz V_{CC} = 2.6 to 5.5 V/3.2 MHz V_{CC} = 2.2 to 5.5 V/2 MHz Other than above/1 MHz Internal step-down circuit used: V_{CC} = 2.2 to 5.5 V/2 MHz Other than above/1 MHz
8-bit mode Active (medium-speed), Sleep (medium-speed) f_{osc} = 400 kHz to 4 MHz	(ϕ/16) (ϕ/32) (ϕ/64) (ϕ/128)	$2 \cdot f_{osc}$ f_{osc} $1/2 \cdot f_{osc}$ $1/4 \cdot f_{osc}$
8-bit mode Watch, Subactive, Subsleep, Standby f_{osc} = 32.768 kHz or 38.4 kHz	(ϕ_w/2) (ϕ_w/4) (ϕ_w/8)	1000 kHz 500 kHz 250 kHz

7. Table 4 indicates function allocation in this task example.

Table 4 Function Allocation

Function	Function Allocation
ECCSR	Sets 16-bit asynchronous event counter functions, detects counter overflow, enables/disables input to ECH, ECL of the event clock.
ECH	Functions as the upper 8-bit up-counter of a 16-bit event counter, taking the ECL overflow signal as the input clock.
ECL	Functions as the lower 8-bit up-counter of a 16-bit event counter, taking the external asynchronous event AEVL pin as the input clock.
AVEL	Functions as the input pin for 2-MHz external asynchronous event input.
IRREC	Indicates whether there has been an asynchronous event counter interrupt request.
IENEC	Enables/disables asynchronous event counter interrupt requests.

Explanation of Operation

1. Figure 3 illustrates the principle of operation. Asynchronous event counter operation is based on the hardware and software processing indicated in the figure.

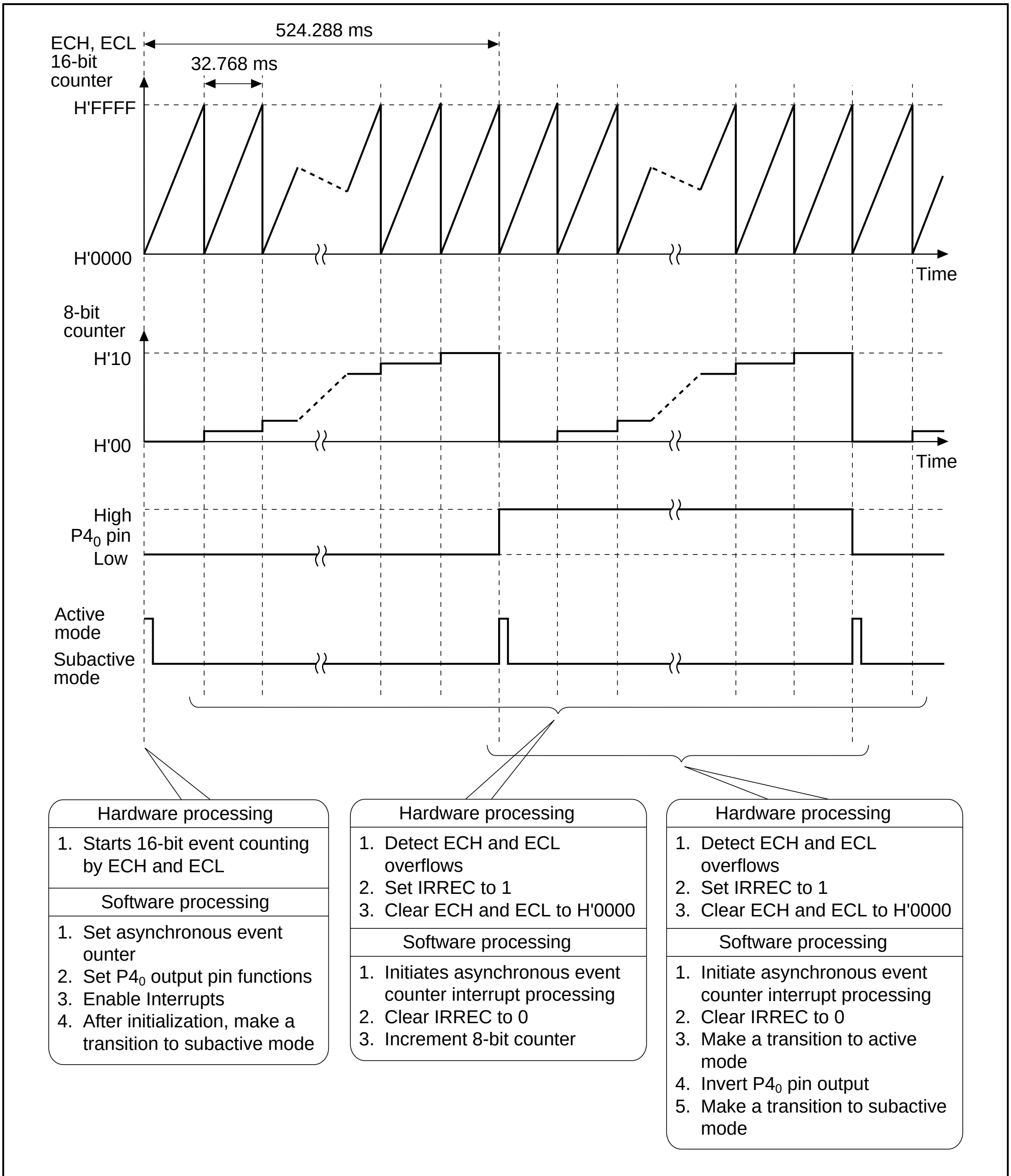


Figure 3 Principle of Operation of Asynchronous Event Counter

Explanation of Software

1. Explanation of Modules

   Table 5 explains the modules in this task example.

Table 5 Explanation of Modules

Module Name	Label Name	Function
Main routine	MAIN	Initializes the stack pointer, RAM, port 4_0, asynchronous event counter, and system control register; enables interrupts; executes direct transitions to subactive mode; after 524.288 ms, controls port output and executes direct transitions to active (high-speed) mode and to subactive mode.
Asynchronous event counter interrupt processing routine	AECINT	By routine for processing asynchronous event counter interrupts, clears an interrupt request flag, increments and initializes an 8-bit counter, and after 524.288 ms, sets a flag in RAM.
Direct transition interrupt processing routine	DTINT	By routine for processing direct transition interrupts, clears the interrupt request flag

2. Explanation of Arguments

   In this task example, no arguments are used.

3. Explanation of Internal Registers Used

Table 6 gives explanations of the internal registers used in this task example.

Table 6 Explanation of Internal Registers Used

Register Name		Description	RAM Address	Setting
ECCSR	OVH	Event counter control/status register (Counter overflow H) A status flag indicating overflow of ECH. • When OVH = 0, indicates no overflow of ECH • When OVH = 1, indicates ECH overflow	H'FF95 Bit 7	0
ECCSR	OVL	Event counter control/status register (Counter overflow L) A status flag indicating overflow of ECL. • When OVL = 0, indicates no overflow of ECL • When OVL = 1, indicates ECL overflow	H'FF95 Bit 6	0
ECCSR	CH2	Event counter control/status register (Channel selection) Selects whether to use ECH and ECL as a single-channel 16-bit event counter, or as two independent 8-bit event counter channels. • When CH2 = 0, ECH and ECL function as a single concatenated 16-bit event counter • When CH2 = 1, ECH and ECL function as two independent 8-bit event counter channels	H'FF95 Bit 4	0
ECCSR	CUEH	Event counter control/status register (Count-up enable H) Enables or disables the event clock input to ECH. • When CUEH = 0, disables the event clock input to ECH • When CUEH = 1, enables the event clock input to ECH	H'FF95 Bit 3	0

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
ECCSR	CUEL	Event counter control/status register (Count-up enable L) Enables or disables the event clock input to ECL. • When CUEL = 0, disables the event clock input to ECL • When CUEL = 1, enables the event clock input to ECL	H'FF95 Bit 2	0
ECCSR	CRCH	Event counter control/status register (Counter reset control H) Controls ECH reset. • When CRCH = 0, ECH is reset • When CRCH = 1, ECH reset is canceled and count-up function is enabled	H'FF95 Bit 1	0
ECCSR	CRCL	Event counter control/status register (Counter reset control L) Controls ECL reset. • When CRCL = 0, ECL is reset • When CRCL = 1, ECL reset is canceled and count-up function is enabled	H'FF95 Bit 0	0
ECH		Event counter H An 8-bit readable up-counter; by combining it with ECL, it can operate as the upper 8 bits of a 16-bit event counter.	H'FF96	H'00
ECL		Event counter L An 8-bit readable up-counter; by combining it with ECH, it can operate as the lower 8 bits of a 16-bit event counter.	H'FF97	H'00

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
TMA	TMA3	Timer mode register A (Internal clock selector 3) Selects the clock input to TCA. • When TMA3 = 0, PSS is selected as the TCA input clock source, and an interval timer function is selected for timer A • When TMA3 = 1, PSW is selected as the TCA input clock source, and a clock time base function is selected for timer A	H'FFB0 Bit 3	1
PMR3	AVEL	Port mode register 3 ($P3_7$/AEVL pin function switch) Determines whether the $P3_7$/AEVL pin is to be used as the $P3_7$ pin, or as the AEVL pin. • When AEVL = 0, the $P3_7$/AEVL pin functions as the $P3_7$ pin • When AEVL = 1, the $P3_7$/AEVL pin functions as the AEVL pin	H'FFCA Bit 7	1
PDR4	$P4_0$	Port data register 4 ($P4_0$) Stores the $P4_0$ pin data. • When $P4_0$ = 0, the $P4_0$ pin output level is low • When $P4_0$ = 1, the $P4_0$ pin output level is high	H'FFD7 Bit 0	0
PCR4	$PCR4_0$	Port control register 4 (Port control register 4_0) Controls the $P4_0$ pin input/output. • When $PCR4_0$ = 0, the $P4_0$ pin functions as an input pin • When $PCR4_0$ = 1, the $P4_0$ pin functions as an output pin	H'FFE7 Bit 0	1

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
SYSCR1	SSBY	System control register 1 (Software standby) Carries out transition to standby mode or watch mode. • When SSBY = 0, after executing a SLEEP instruction in active mode, causes a transition to sleep mode, or after executing a SLEEP instruction in subactive mode, causes a transition to subsleep mode • When SSBY = 1, after executing a SLEEP instruction in active mode, causes a transition to standby mode or to watch mode, or after executing a SLEEP instruction in subactive mode, causes a transition to watch mode	H'FFF0 Bit 7	1
SYSCR1	STS2 STS1 STS0	System control register 1 (Standby timer select 2 to 0) Specify the time for the CPU and peripheral functions to wait until the clock stabilizes when standby mode or watch mode is canceled and a transition is made to active mode due to a specific interrupt. • When STS2 to STS1 = 000, standby time is 8,192 states • When STS2 to STS1 = 001, standby time is 16,384 states • When STS2 to STS1 = 010, standby time is 32,768 states • When STS2 to STS1 = 011, standby time is 65,536 states • When STS2 to STS1 = 100, standby time is 131,072 states • When STS2 to STS1 = 101, standby time is 2 states • When STS2 to STS1 = 110, standby time is 8 states • When STS2 to STS1 = 111, standby time is 16 states	H'FFF0 Bit 6 to bit 4	STS2 = 0 STS1 = 0 STS0 = 0

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
SYSCR1	LSON	System control register 1 (Low speed on flag) When watch mode is canceled, selects either the system clock (ϕ) or the subclock (ϕ_{sub}) as the CPU operating clock. • When LSON = 0, selects the system clock (ϕ) as the CPU operating clock • When LSON = 1, selects the subclock (ϕ_{sub}) as the CPU operating clock	H'FFF0 Bit 3	1
SYSCR2	NESEL	System control register 2 (Noise elimination sampling frequency selection) Selects the frequency at which the watch clock signal (ϕ_w) generated by the subclock oscillator is sampled relative to the oscillator clock (ϕ_{osc}) generated by the system clock oscillator. • When NESEL = 0, sampling rate is ϕ_{osc} /16 • When NESEL = 1, sampling rate is ϕ_{osc} /4	H'FFF1 Bit 4	1

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
SYSCR2	DTON	System control register 2 (Direct transfer on flag) Specifies whether or not to make direct transitions among active (high-speed) mode, active (medium-speed) mode, and subactive mode when a SLEEP instruction is executed. • When DTON = 0, if a SLEEP instruction is executed in active mode, a transition to standby mode, watch mode or sleep mode occurs; if a SLEEP instruction is executed in subactive mode, a transition to watch mode or subsleep mode occurs • When DTON = 1, if a SLEEP instruction is executed in active (high-speed) mode, a direct transition occurs to active (medium-speed) mode (when SSBY = 1, MSON = 1, LSON = 0) or to subactive mode (when SSBY = 1, TMA3 = 1, LSON = 1); if a SLEEP instruction is executed in active (medium-speed) mode, a direct transition occurs to active (high-speed) mode (when SSBY = 0, MSON = 0, LSON = 0) or to subactive mode (when SSBY = 1, TMA3 = 1, LSON = 1); and if a SLEEP instruction is executed in subactive mode, a direct transition occurs to active (high-speed) mode (when SSBY = 1, TMA3 = 1, LSON = 0, MSON = 0) or to active (medium-speed) mode (when SSBY = 1, TMA3 = 1, LSON = 0, MSON = 1)	H'FFF1 Bit 3	1

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
SYSCR2	MSON	System control register 2 (Medium speed on flag) Selects whether to operate in active (high-speed) mode or in active (medium-speed) mode after cancellation of standby mode, watch mode, or sleep mode. • When MSON = 0, operates in active (high-speed) mode • When MSON = 1, operates in active (medium- speed) mode	H'FFF1 Bit 2	0
SYSCR2	SA1 SA0	System control register 1 (Subactive mode clock select 1, 0) Select the CPU clock rate (ϕ_w/8, ϕ_w/4, ϕ_w/2) in subactive mode. • When SA1 = 0 and SA0 = 0, ϕ_w/8 is selected • When SA1 = 0 and SA0 = 1, ϕ_w/4 is selected • When SA1 = 1 and SA0 = *, ϕ_w/2 is selected *: Don't care	H'FFF0 Bit 1, bit 0	1
IRR2	IRRDT	Interrupt request register 2 (Direct transition interrupt request flag) Indicates whether there has been a direct transition interrupt request. • When IRRDT = 0, indicates that no direct transition interrupt has been requested • When IRRDT = 1, indicates that a direct transition interrupt has been requested	H'FFF7 Bit 7	0

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
IRR2	IRREC	Interrupt request register 2 (Asynchronous event counter interrupt request flag) Indicates whether there has been an asynchronous event counter interrupt request. • When IRREC = 0, indicates that no asynchronous event counter interrupt has been requested • When IRREC = 1, indicates that an asynchronous event counter interrupt has been requested	H'FFF7 Bit 0	0
IENR2	IENDT	Interrupt enable register 2 (Direct transition interrupt enable) Enables or disables direct transition interrupt requests. • When IENDT = 0, disables direct transition interrupt requests • When IENDT = 1, enables direct transition interrupt requests	H'FFF4 Bit 7	1
IENR2	IENEC	Interrupt enable register 2 (Asynchronous event counter interrupt enable) Enables or disables asynchronous event counter interrupt requests. • When IENEC = 0, disables asynchronous event counter interrupt requests • When IENEC = 1, enables asynchronous event counter interrupt request	H'FFF4 Bit 0	1

4. Explanation of RAM Usage

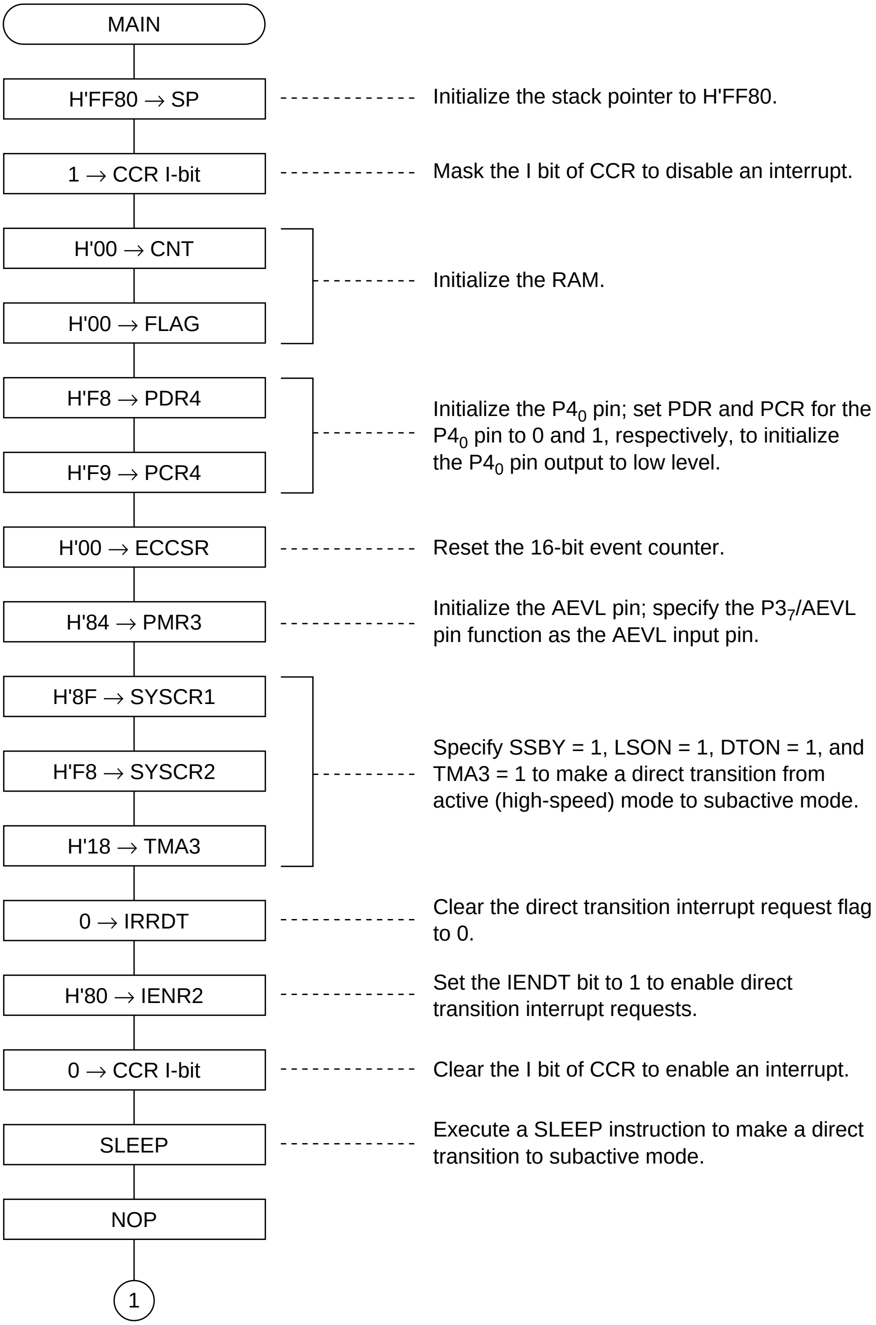
Table 7 explains RAM usage for this task example.

Table 7 Explanation of RAM Usage

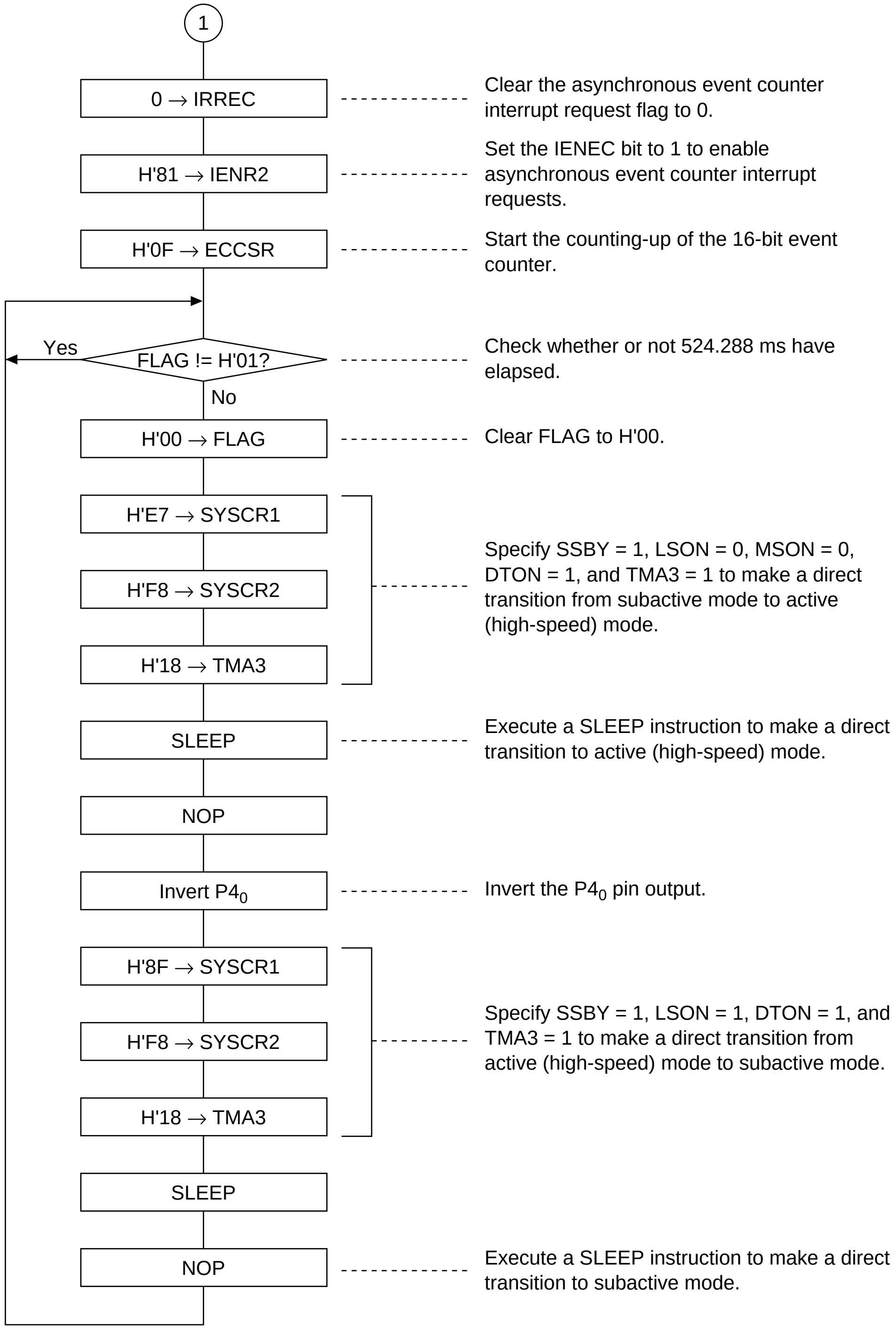
Label Name	Function	RAM Address	Modules Used
FLAG	Flag indicating 524.288 ms have elapsed.	H'F780	MAIN, AECINT
CNT	8-bit counter to count the number of occurrences of a timer F interrupt request.	H'F781	MAIN, AECINT

Flowchart

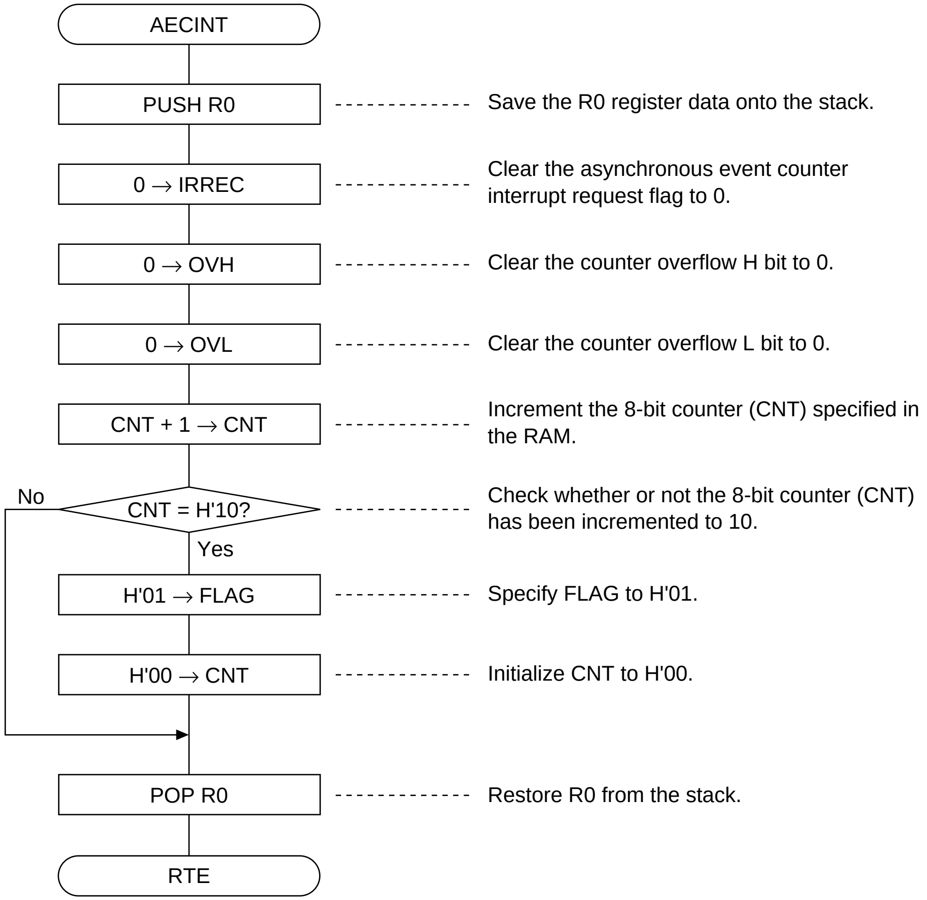
1. Main routine



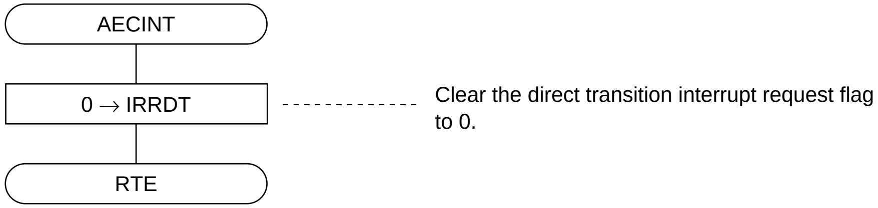
1. Main routine (cont)



2. Asynchronous event counter interrupt processing routine



3. Direct transition interrupt processing routine



Program Lists

```
;*********************************************************
;*   H8/3867 Application Note
;*
;*     'Asynchronous Event Counter Control'
;*
;*             Function : AEC(Asynvhronous Event Counter)  *
;*
;*             External Clock : 6MHz
;*             Internal Clock : 3MHz
;*             Sub Clock      : 32.768kHz
;*********************************************************
;
          .cpu     300l
;
;*********************************************************
;*         Symbol Defnition
;*********************************************************
;
ECCSR      .equ     h'ff95      ;Event Counter Control/Status Register
ECH        .equ     h'ff96      ;Event Counter H
ECL        .equ     h'ff97      ;Event Counter L
TMA        .equ     h'ffb0      ;Timer Mode Register A
PMR3       .equ     h'ffca      ;Port Mode Register 3
PDR4       .equ     h'ffd7      ;Port Data Register 4
PCR4       .equ     h'ffe7      ;Port Control Register 4
SYSCR1     .equ     h'fff0      ;System Control Register 1
SYSCR2     .equ     h'fff1      ;System Control Register 2
IENR2      .equ     h'fff4      ;Interrupt Enable Register 2
IRR2       .equ     h'fff7      ;Interrupt Request Register 2
;
;*********************************************************
;*         RAM Allocation
;*********************************************************
;
FLAG       .equ     h'f780      ;Bit0 : Event Flag
CNT        .equ     h'f781      ;8-bit Counter
```

```
;
;*********************************************************
;*         Vector Address
;*********************************************************
;
           .org     h'0000
           .data.w  MAIN        ;No.0 Reset Interrupt(H'0000-H'0001)
;
           .org     h'0008
           .data.w  MAIN        ;No.4 _IRQ0 Interrupt(H'0008-H'0009)
           .data.w  MAIN        ;No.5 _IRQ1 Interrupt(H'000A-H'000B)
           .data.w  MAIN        ;No.6 _IRQ2 Interrupt(H'000C-H'000D)
           .data.w  MAIN        ;No.7 _IRQ3 Interrupt(H'000E-H'000F)
           .data.w  MAIN        ;No.8 _IRQ4 Interrupt(H'0010-H'0011)
           .data.w  MAIN        ;No.9 _WKP0-_WKP7 Interrupt(H'0012-H'0013)
;
           .org     h'0016
           .data.w  MAIN        ;No.11 Timer A Interrupt(H'0016-H'0017)
           .data.w  AECINT      ;No.12 AEC Interrupt(H'0018-H'0019)
           .data.w  MAIN        ;No.13 Timer C Interrupt(H'001A-H'001B)
           .data.w  MAIN        ;No.14 Timer FL Interrupt(H'001C-H'001D)
           .data.w  MAIN        ;No.15 Timer FH Interrupt(H'001E-H'001F)
           .data.w  MAIN        ;No.16 Timer G Interrupt(H'0020-H'0021)
           .data.w  MAIN        ;No.17 SCI31 Interrupt(H'0022-H'0023)
           .data.w  MAIN        ;No.18 SCI32 Interrupt(H'0024-H'0025)
           .data.w  MAIN        ;No.19 A/D Converter Interrupt(H'0026-H'0028)
           .data.w  DTINT       ;No.20 Direct Transfer Interrupt(H'0028-H'0029)
;
;*********************************************************
;*         MAIN : Main Routine
;*********************************************************
;
           .org     h'1000
;
MAIN:      .equ     $
           mov.w    #h'ff80,sp  ;Initialize Stack Pointer
           orc      #h'80,ccr   ;Interrupt Disable
```

```
;
            mov.b    #h'00,r0l
            mov.b    r0l,@CNT     ;Initialize 8-bit Counter
            mov.b    r0l,@FLAG    ;Initialize Event Flag
;
            mov.w    #h'f8f9,r0
            mov.b    r0h,@PDR4    ;Initialize P40 PDR
            mov.b    r0l,@PCR4    ;Initialize P40 Terminall Function
;
            mov.b    #h'00,r0l
            mov.b    r0l,@ECCSR   ;Reset 16-bit Event Counter
            mov.b    #h'84,r0l
            mov.b    r0l,@PMR3    ;Initialize AEVL Terminal Function
;
            mov.b    #h'8f,r0l    ;SSBY="1", LSON="1"
            mov.b    r0l,@SYSCR1 ;DTON="1", TMA3="1"
            mov.b    #h'F8,r0l
            mov.b    r0l,@SYSCR2
            mov.b    #H'18,r0l
            mov.b    r0l,@TMA
;
            bclr     #7,@IRR2     ;Clear IRRDT
            mov.b    #h'80,r0l
            mov.b    r0l,@IENR2   ;Direct Transfer Interrupt Enable
;
            andc     #h'7f,ccr    ;Interrupt Enable
;
            sleep                 ;Direct Transfer to Subactive Mode
            nop
;
            bclr     #0,@IRR2     ;Clear IRREC
            mov.b    #h'81,r0l
            mov.b    r0l,@IENR2   ;Asynchronous Event Counter Interrupt Enable
;
            mov.b    #h'0f,r0l
            mov.b    r0l,@ECCSR   ;16-bit Event counter count-up start
;
```

```
EVTMN:    mov.b    @FLAG,r0l
          btst     #0,r0l        ;Event Flag = "1" ?
          beq      EVTMN         ;No.
;
          mov.b    #h'00,r0l
          mov.b    r0l,@FLAG     ;Clear Event Flag
;
          mov.w    #h'e7f8,r0    ;SSBY="1", LSON="0"
          mov.b    r0h,@SYSCR1   ;MSON="0", DTON="1"
          mov.b    r0l,@SYSCR2   ;TMA3="1"
          mov.b    #h'18,r0l     ;STS2-0="000"
          mov.b    r0l,@TMA
;
          sleep                  ;Direct Transfer to Active Mode
          nop
;
          mov.b    @PDR4,r0l     ;Load PDR4
          bnot     #0,r0l        ;Invert P40 PDR
          mov.b    r0l,@PDR4     ;Store PDR4
;
          mov.w    #h'8ff8,r0    ;SSBY="1", LSON="1"
          mov.b    r0h,@SYSCR1   ;DTON="1", TMA3="1"
          mov.b    r0l,@SYSCR2
          mov.b    #h'18,r0l
          mov.b    r0l,@TMA
;
          sleep                  ;Direct Transfer to Subavtive Mode
          nop
;
          bra      EVTMN
;
;************************************************************
;*        AECINT : AEC Interrupt Routine
;************************************************************
;
AECINT:   .equ     $
          push     r0            ;Store r0
```

```
;
            bclr    #0,@IRR2     ;Clear IRREC
            bclr    #7,@ECCSR    ;Clear OVH
            bclr    #6,@ECCSR    ;Clear OVL
;
            mov.b   @CNT,r0l     ;Load CNT
            inc     r0l          ;Increment CNT
            cmp.b   #h'10,r0l    ;CNT = h'10 ?
            beq     EVNT         ;Yes. CNT Initialize
            mov.b   r0l,@CNT     ;Store CNT
            bra     RNFI
;
EVNT:       mov.b   #h'01,r0l
            mov.b   r0l,@FLAG    ;Set Event Flag
            mov.b   #h'00,r0l
            mov.b   r0l,@CNT     ;Initialize 8-bit Counter
;
RNFI        pop     r0           ;Restore r0
            rte
;
;***********************************************************
;*          DTINT : Direct Transfer Interrupt Routine
;***********************************************************
;
DTINT:      .equ    $
            bclr    #7,@IRR2     ;Clear IRRDT
            rte
;
            .end
```

2.3 LCD Display with Static Duty

LCD Display with Static Duty	MCU: H8/3867 Series	Functions Used: LCD Controller/Driver

Specifications

1. LCD display is performed using the segment-type LCD controller circuit, LCD driver, and power supply circuit of the H8/3867 Series.

2. A single common signal and 32 segment signals are used for LCD display with static duty.

3. As the power supply driving the LCD, a step-up constant-voltage power supply (5 V) is used.

4. An example of LCD module connection and an LCD display example for this task example appear in figure 1.

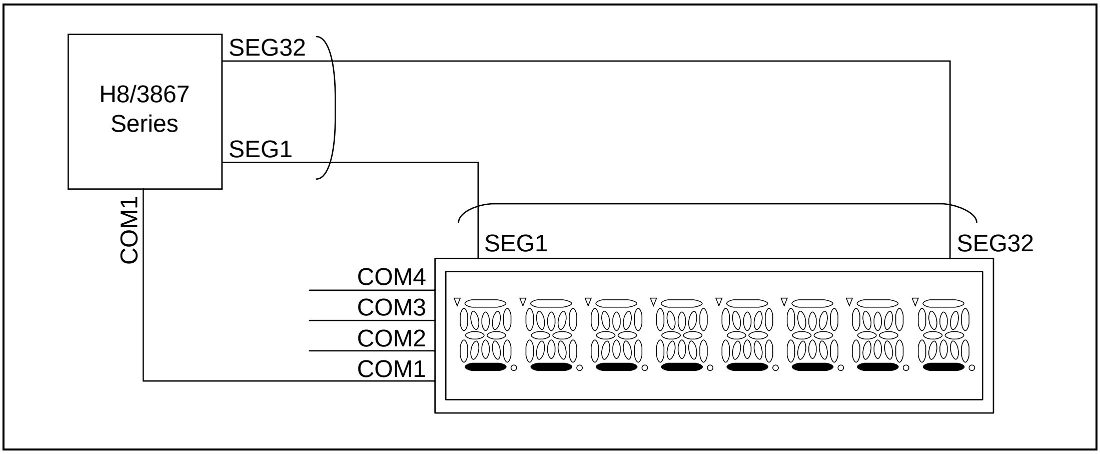


Figure 1 LCD Display Example

Explanation of Functions Used

1. In this task example, the LCD controller/driver is used for LCD display. The features of the LCD controller/driver are described below.

    - Display capacity
        a. Duty cycle: static
           Internal driver: 32 segments
           Segment external-expansion driver: 256 segments
        b. Duty cycle: 1/2
           Internal driver: 32 segment
           Segment external-expansion driver: 128 segments
        c. Duty cycle: 1/3
           Internal driver: 32 segment
           Segment external-expansion driver: 64 segments
        d. Duty cycle: 1/4
           Internal driver: 32 segment
           Segment external-expansion driver: 64 segments
    - LCD RAM capacity: 8 bits × 32 bytes (256 bits)
    - LCD RAM is word-accessible.
    - All segment output pins can be used as port pins in eight-pin units.
    - Depending on the duty cycle, the common output pins not used can be used for a common double-buffer (parallel connection).
    - Display is possible in all operating modes other than standby mode.
    - Frame frequency can be selected from among 11 values.
    - A power supply split-resistance is built-in, for supply of LCD driver power.
    - Use of module standby mode enables a module to be placed in standby mode independently when not used.
    - An internal step-up constant-voltage (5 V) power supply enables LCD display even at low voltages.
    - A or B waveform can be selected by software.

2. Figure 2 is a block diagram of the LCD controller/driver used in this task examples.

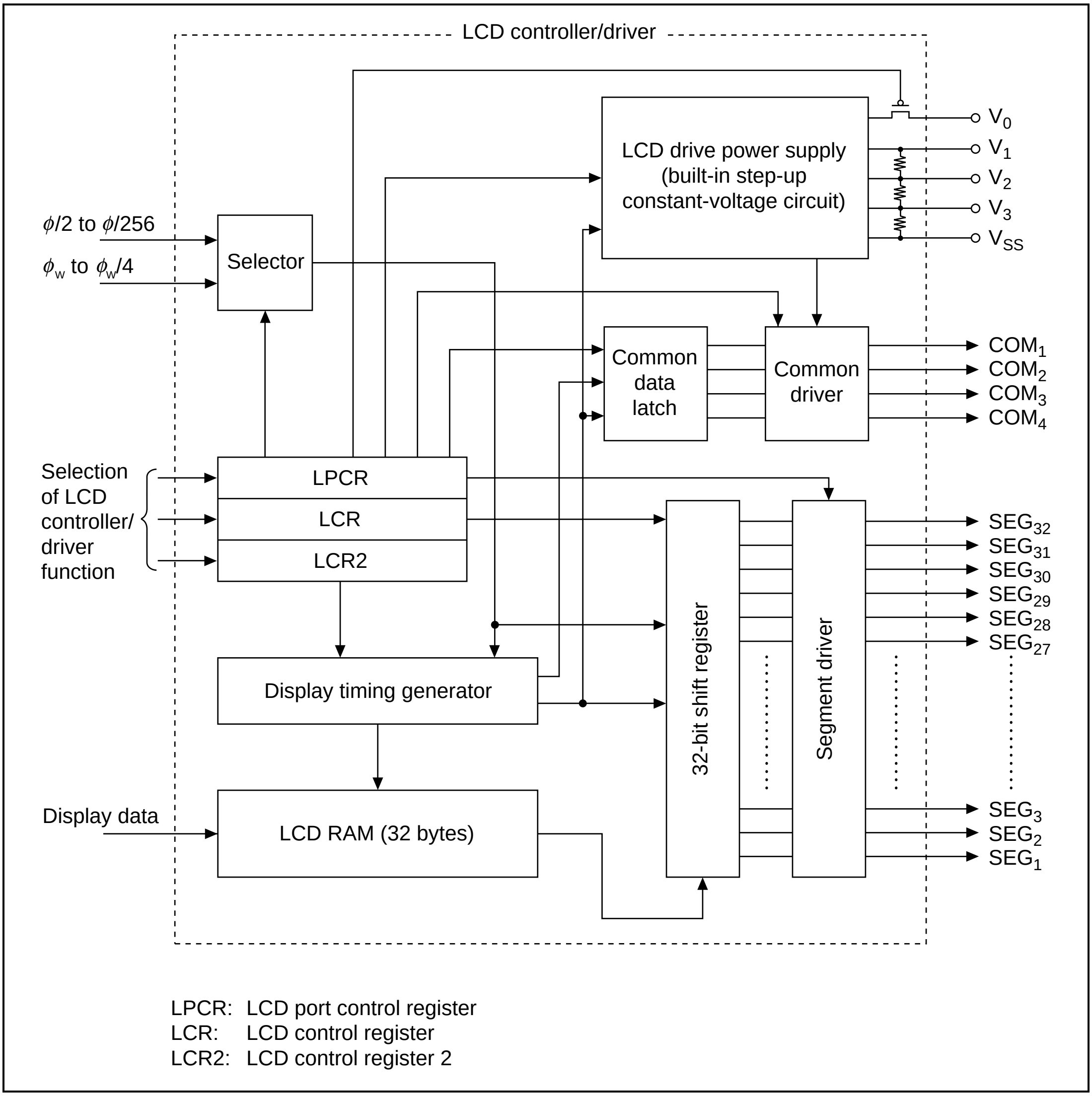


Figure 2 Block Diagram of LCD Controller/Driver (LCD Display with Static Duty)

3. Functions of the LCD controller/driver are explained in table 1.

Table 1 LCD Controller/Driver Functions

LCD port control register (LPCR)	
Function	LPCR is an 8-bit read/write register which selects the duty cycle, the LCD driver and pin functions. LPCR is initialized to H'00 upon reset.
LCD control register (LCR)	
Function	LCR is an 8-bit read/write register which turns the LCD drive power supply on and off, controls display data, and selects the frame frequency. LCR is initialized to H'80 upon reset.
LCD control register 2 (LCR2)	
Function	LCR2 is an 8-bit read/write register which controls switching between A and B waveforms, selects the driver power supply, controls the step-up constant-voltage (5 V) power supply, and selects the duty cycle for charge/discharge pulses controlling disconnection of the power supply split-resistance from power supply circuit. LCR2 is initialized to H'60 upon reset.
Segment output pins (SEG_{32} to SEG_1)	
Function	These are pins used for LCD segment driving; all these pins are multiplexed as port pins, and their functions can be selected programmably.
Common output pins (COM_4 to COM_1)	
Function	These are LCD common driving output pins; under static or 1/2-duty driving, they can be configured in parallel.
Segment external expansion signal pin (CL_1)	
Function	This is a display data latch clock pin which is multiplexed as SEG_{32}.
Segment external expansion signal pin (CL_2)	
Function	This is a display data shift clock pin which is multiplexed as SEG_{31}.
Segment external expansion signal pin (M)	
Function	This is an LCD alternation signal pin which is multiplexed as SEG_{29}.
Segment external expansion signal pin (DO)	
Function	This is a serial display data signal pin which is multiplexed as SEG_{30}.
LCD power supply pins (V_0 to V_3)	
Function	These pins are used when connecting an external bypass capacitor or when using an external power supply circuit.
LCD RAM	
Function	Sets the display data. The relation between the LCD RAM and the display segments differs depending on the duty cycle. After the registers necessary for display have been set, instructions similar to the instructions for normal RAM are used to write data corresponding to the duty, and when the display is turned on, display is started automatically. Word/byte access instructions can be used to set data in the LCD RAM.

4. In this task example, a 16-line 8-character segment LCD is used for display under static driving. Figure 3 is a diagram showing connections for segment signals and common signals of 16-line 8-character segment LCD used in this task example.

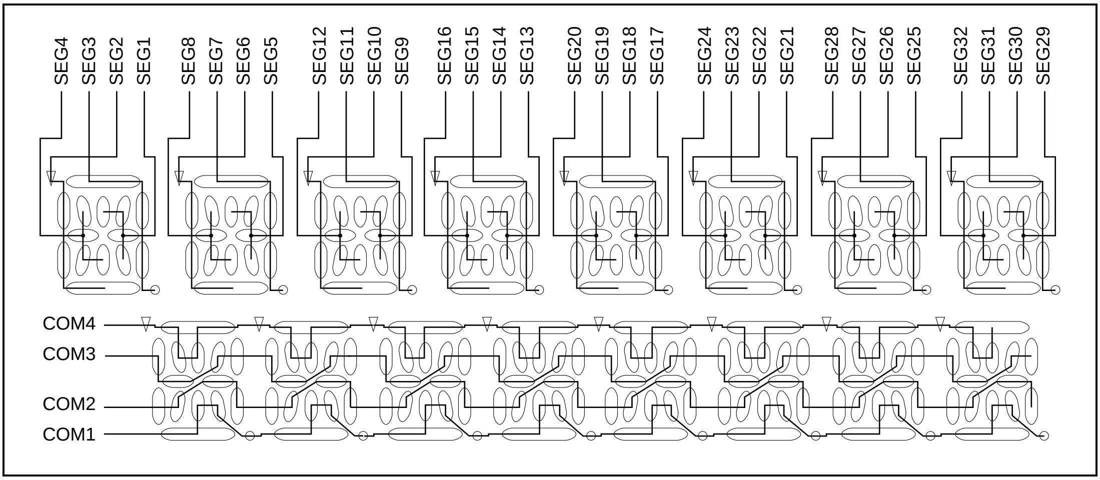


Figure 3 Connections of Segment Signals and Common Signals of the 16-Line 8-Character Segment LCD Used in this Task Example

5. Figure 4 shows the LCD RAM mapping under static driving without segment external expansion.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
H'F740	SEG_8	SEG_7	SEG_6	SEG_5	SEG_4	SEG_3	SEG_2	SEG_1
H'F741	SEG_{16}	SEG_{15}	SEG_{14}	SEG_{13}	SEG_{12}	SEG_{11}	SEG_{10}	SEG_9
H'F742	SEG_{24}	SEG_{23}	SEG_{22}	SEG_{21}	SEG_{20}	SEG_{19}	SEG_{18}	SEG_{17}
H'F743	SEG_{32}	SEG_{31}	SEG_{30}	SEG_{29}	SEG_{28}	SEG_{27}	SEG_{26}	SEG_{25}
H'F744								
H'F745								
H'F746								
H'F747								
H'F748								
H'F749								
H'F74A								
H'F74B								
H'F74C								
H'F74D								
H'F74E								
H'F74F								
	COM_1	COM_1	COM_1	COM_1	COM_1	COM_1	COM_1	COM_1

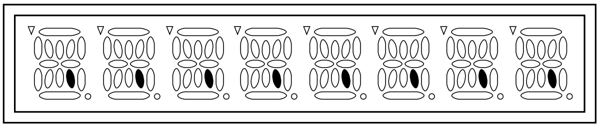
H'F740 to H'F743: Display area

H'F744 to H'F74F: Area not used for display

Figure 4 LCD RAM Mapping under Static Driving without Segment External Expansion

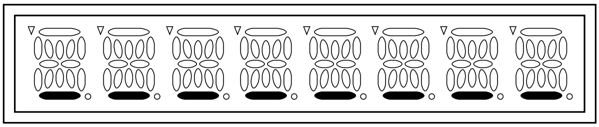
6. Figure 5 shows the relation between the 16-line 8-character segment LCD display and LCD RAM settings used in this task example. In this example, the LCD display is cycled through the series pattern 1 → pattern 2 → pattern 3 → pattern 4 → pattern 1 →.

1. Pattern 1



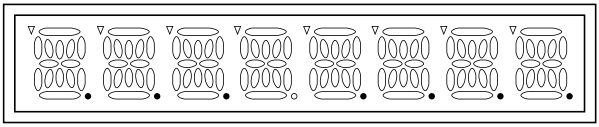
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
H'F740	0	0	0	1	0	0	0	1
H'F741	0	0	0	1	0	0	0	1
H'F742	0	0	0	1	0	0	0	1
H'F743	0	0	0	1	0	0	0	1

2. Pattern 2



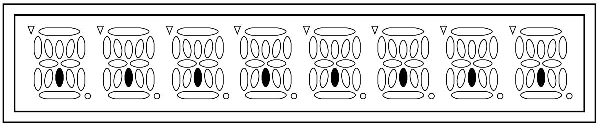
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
H'F740	0	0	1	0	0	0	1	0
H'F741	0	0	1	0	0	0	1	0
H'F742	0	0	1	0	0	0	1	0
H'F743	0	0	1	0	0	0	1	0

3. Pattern 3



	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
H'F740	0	1	0	0	0	1	0	0
H'F741	0	1	0	0	0	1	0	0
H'F742	0	1	0	0	0	1	0	0
H'F743	0	1	0	0	0	1	0	0

4. Pattern 4



	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
H'F740	1	0	0	0	1	0	0	0
H'F741	1	0	0	0	1	0	0	0
H'F742	1	0	0	0	1	0	0	0
H'F743	1	0	0	0	1	0	0	0

Figure 5 Relation between LCD Display and LCD RAM Settings

7. Figure 6 shows the relation between the LCD RAM addresses and the segments SEG_1 through SEG_8 of the 16-line 8-character segment LCD. As the figure indicates, when the LCD RAM bits corresponding to 0 through 7 are set to 1 the corresponding LCD areas are lit, and when cleared to 0 the corresponding areas are unlit.

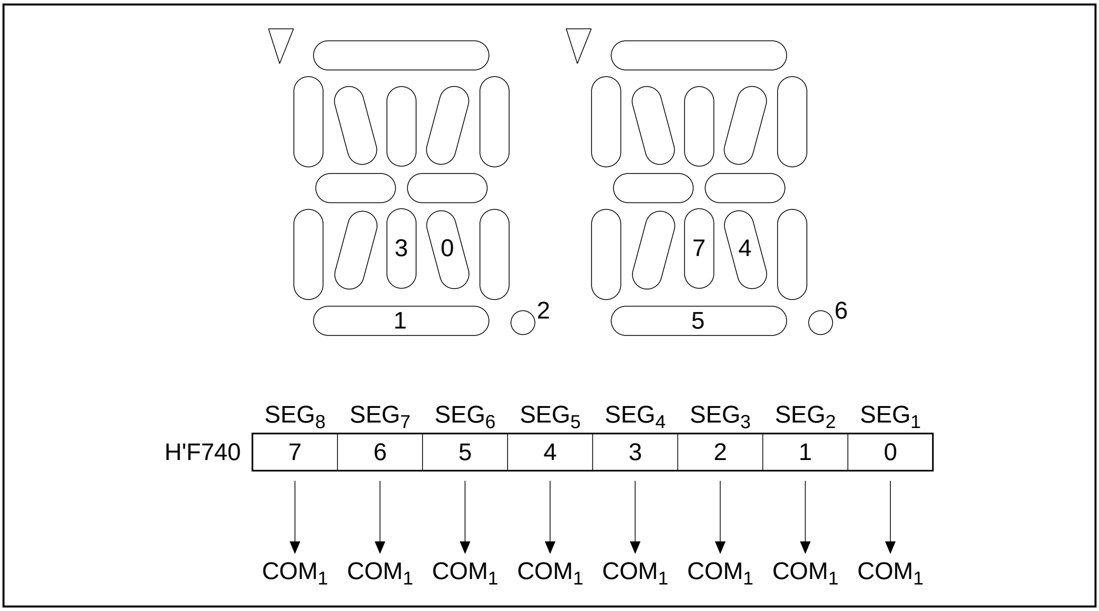


Figure 6 Relation between LCD Lit/Unlit States and LCD RAM Settings

8. Table 2 indicates function allocations in this task example.

Table 2 Function Allocations

Function	Function Allocation
LPCR	Selects duty cycle, LCD driver, and pin functions.
LCR	Turns LCD drive power supply on and off, controls display data, and selects frame frequency.
LCR2	Switches between A and B waveforms, selects drive power supply, controls step-up constant-voltage (5 V) power supply, selects duty cycle for charge/discharge pulses to control disconnection of power supply split-resistance from power supply circuit.
SEG_{32} to SEG_1	Used as segment drivers.
COM_1	Used as a common driver.
V_0, V_1	The V_0 and V_1 pins are shorted in order to use the step-up constant-voltage (5 V) power supply as the LCD drive power supply.
LCD RAM	Sets the LCD display data.

Explanation of Functions Used

1. Hardware settings for LCD display are explained below.

   a. LCD drive power supply settings

      The H8/3867 Series can use either the internal power supply circuit or an external power supply circuit as the LCD drive power supply. In addition, either the power supply voltage (V_{CC}) or the step-up constant-voltage (5 V) can be selected for the internal power supply circuit.

      When using the internal power supply circuit to drive the LCD, the V_0 and V_1 pins are connected externally, as illustrated in figure 7.

      In this task example, the step-up constant-voltage power supply is used as the LCD drive power supply.

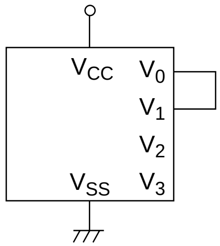


Figure 7 Example of Connection of LCD Power Supply Pins When Using Internal Power Supply Circuit

b. Contrast control function

A block diagram of the LCD drive power supply circuit appears in figure 8. Either V_{CC} or a 5 V output from the step-up constant-voltage power supply circuit is output to pin V_0. When these voltages are used directly to drive the LCD, the V_0 and V_1 pins should be shorted. By inserting a variable resistance R between the V_0 and V_1 pins, the voltage applied to the V_1 pin can be adjusted, and the LCD panel contrast can be controlled.

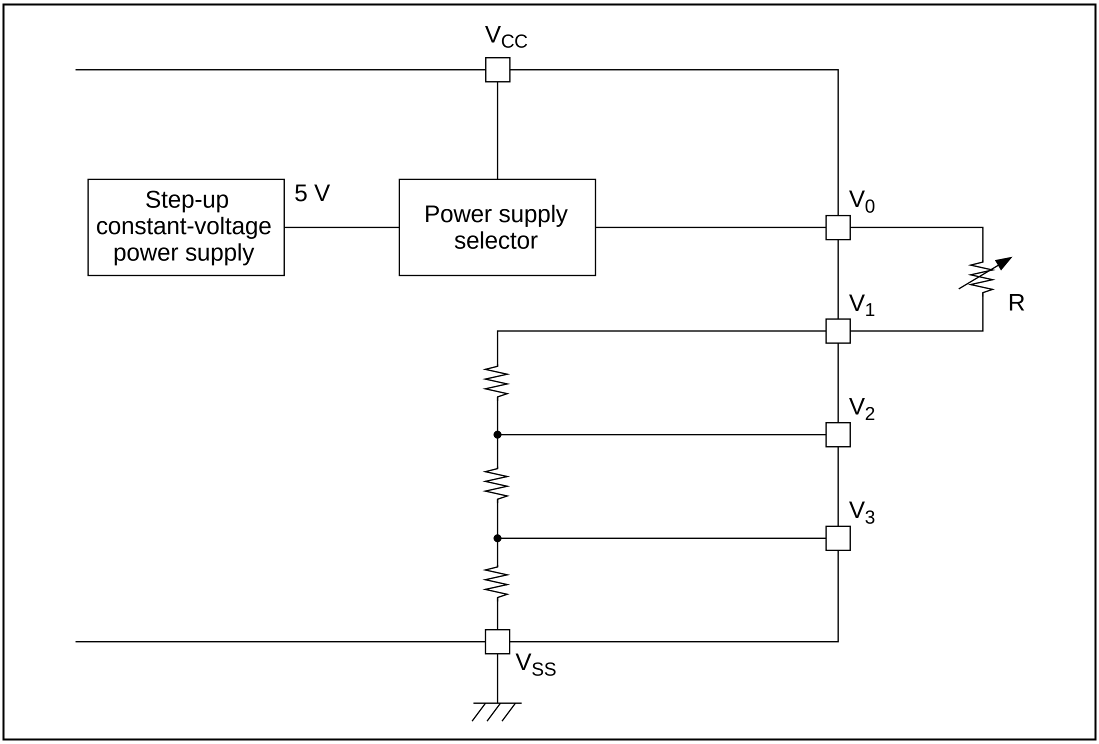


Figure 8 Block Diagram of LCD Drive Power Supply Circuit

c. Step-up constant-voltage (5 V) power supply

The H8/3867 Series has an internal step-up constant-voltage (5 V) power supply, supplying a constant 5 V independent of V_{CC}.

By setting SUPS of the LCD control register 2 (LCR2) to 1, the step-up constant-voltage (5 V) power supply is activated, and a constant 5 V is output to the V_0 pin. This can be used either with pins V_0 and V_1 short-circuited, or with a resistance inserted to divide the voltage.

Note: The step-up constant-voltage (5 V) power supply must not be used for purposes other than to drive the LCD. In addition, when driving a large panel, the power supply capacity may be insufficient. In such cases, either V_{CC} or an external power supply circuit can be used as the power supply.

2. Software settings for LCD display are explained below.

a. Duty selection

DTS1 and DTS0 are used to select from among static, 1/2 duty, 1/3 duty, and 1/4 duty.

b. Segment driver selection

SGS3 to SGS0 are used to select the segment drivers to be used.

c. Frame frequency selection

By setting CKS3 to CKS0, the frame frequency can be selected. The frame frequency should be selected according to the LCD panel.

d. Selection of A and B waveforms

LCDAB can be used to select either the A or the B waveform for use as the LCD waveform.

e. Selection of LCD drive power supply

When using the internal power supply circuit, SUPS can be used to select the power supply to be used. When using an external power supply circuit, SUPS is used to select V_{CC}, and PSW should be used to turn off the LCD drive power supply.

3. Figure 9 shows the operation principle of this task example.

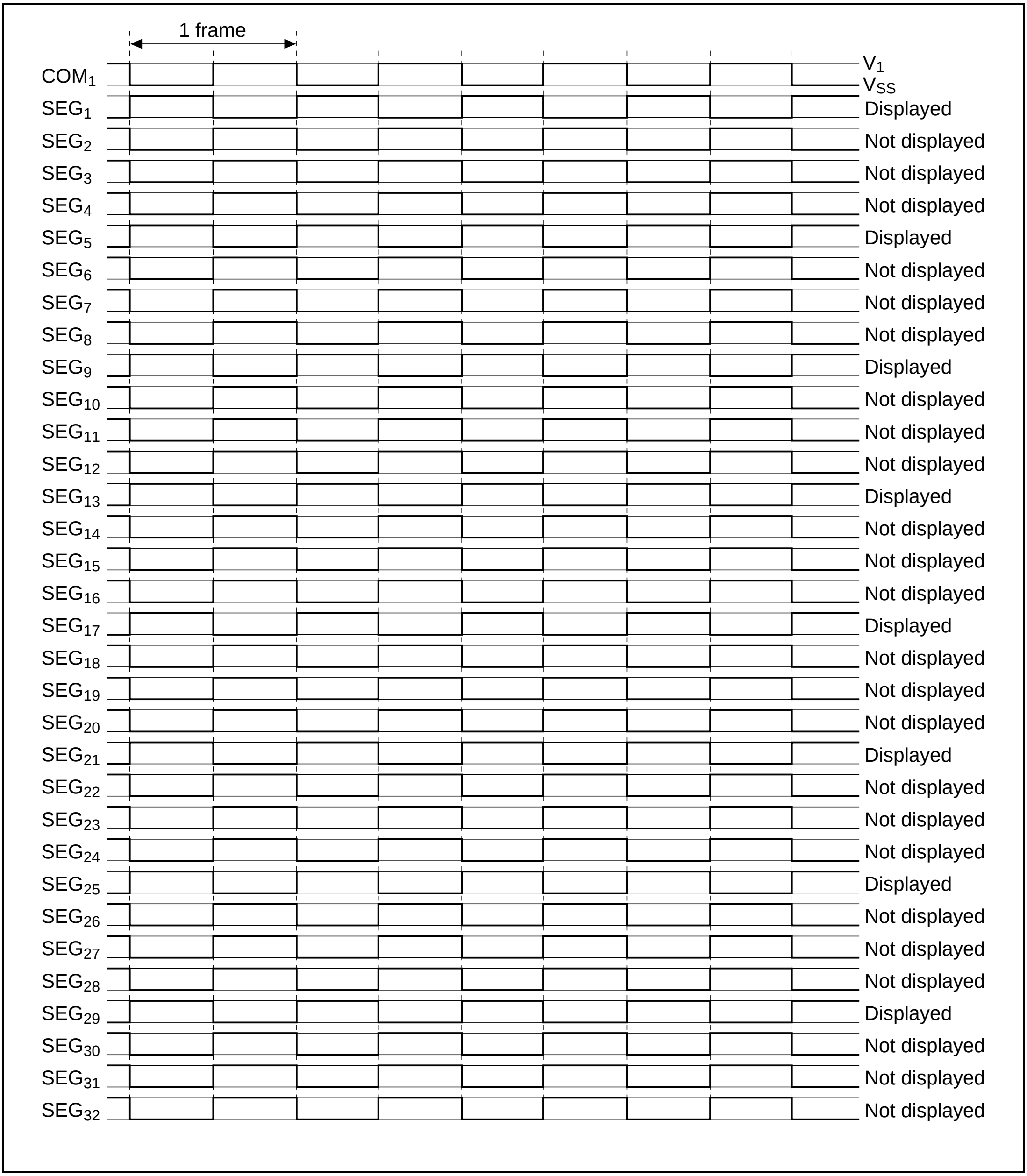


Figure 9 Operation Principle

Explanation of Software

1. Explanation of Modules

   Table 3 explains the modules in this task example.

Table 3 Module Explanation

Module Name	Label Name	Function
Main routine	MAIN	Initializes the stack pointer, LCD RAM and LCD controller/driver, enables interrupts, and controls LCD display.

2. Explanation of Arguments

   In this task example, no arguments are used.

3. Explanation of Internal Registers Used

   Table 4 gives explanation of the internal registers used in this task example.

Table 4 Explanation of Internal Registers Used

Register Name		Description	RAM Address	Settings
LPCR	DTS1, DTS0	LCD port control register (Duty cycle selection 1, 0) Select duty from among static, 1/2 duty, 1/3 duty, and 1/4 duty. • When DTS1 = 0 and DTS0 = 0, static duty is selected • When DTS1 = 0 and DTS0 = 1, 1/2 duty is selected • When DTS1 = 1 and DTS0 = 0, 1/3 duty is selected • When DTS1 = 1 and DTS0 = 1, 1/4 duty is selected	H'FFC0 Bit 7, bit 6	DTS1 = 0 DTS0 = 0

Table 4 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Settings
LPCR	CMX	LCD port control register (Common function selection) Selects whether the same waveform is output from several pins in order to increase the common driving capacity, when common pins are not selected for a given duty cycle. • When CMX = 0, the same waveform is not output from multiple common pins not used at that duty cycle • When CMX = 1, the same waveform is output from multiple common pins not used at that duty cycle	H'FFC0 Bit 5	0
LPCR	SGX	LCD port control register (Expansion signal select) Selects whether the SEG_{32}/CL_1, SEG_{31}/CL_2, SEG_{30}/D0, and SEG_{29}/M pins are used as segment pins (SEG_{32} through SEG_{29}), or as segment external expansion signal pins (CL_1, CL_2, D0, M). • When SGX = 0, they are used as segment pins (SEG_{32} through SEG_{29}) • When SGX = 1, they are used as segment external expansion signal pins (CL_1, CL_2, D0, M)	H'FFC0 Bit 4	0

Table 4 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Settings
LPCR	SGS3 to SGS0	LCD port control register (Segment driver selection) Select the segment driver to be used. • When SGX = 0, SGS3 = 0, SGS2 = 0, SGS1 = 0 and SGS0 = 0, pins SEG_{32} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 0, SGS1 = 0 and SGS0 = 1, pins SEG_{32} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 0, SGS1 = 1 and SGS0 = *, pins SEG_{32} through SEG_{25} function as segment drivers and pins SEG_{24} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 1, SGS1 = 0 and SGS0 = *, pins SEG_{32} through SEG_{17} function as segment drivers and pins SEG_{16} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 1, SGS1 = 1 and SGS0 = *, pins SEG_{32} through SEG_9 function as segment drivers and pins SEG_8 through SEG_1 function as ports • When SGX = 0, SGS3 = 1, SGS2 = *, SGS1 = * and SGS0 = *, pins SEG_{32} through SEG_1 function as segment drivers • When SGX = 1, SGS3 = 0, SGS2 = 0, SGS1 = 0 and SGS0 = 0, pins SEG_{32} through SEG_{29} function as external expansion pins and pins SEG_{28} through SEG_1 function as ports • SGX = 1, SGS3 = *, SGS2 = *, SGS1 = * and SGS0 = * cannot be specified *: Don’t care	H'FFC0 Bit 3 to bit 0	SGS3 = 1 SGS2 = 0 SGS1 = 0 SGS0 = 0

Table 4 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Settings
LCR	PSW	LCD control register (LCD drive power supply on/off control) Turns the LCD drive power supply off when LCD display is not used in power-down mode, or when an external power supply is used. When the ACT bit is cleared to 0, or when in standby mode, the LCD drive power supply is turned off regardless of this bit setting. • When PSW = 0, the LCD drive power supply is turned off • When PSW = 1, the LCD drive power supply is turned on	H'FFC1 Bit 6	1
LCR	ACT	LCD control register (Display function activate) Selects whether the LCD controller/driver is to be used or not. By clearing this bit to 0, LCD controller/driver operation is halted. Also, regardless of the value of PSW, the LCD drive power supply is turned off. However, the register contents are maintained. • When ACT = 0, LCD controller/driver operation is halted • When ACT = 1, LCD controller/driver operations	H'FFC1 Bit 5	1
LCR	DISP	LCD control register (Display data control) DISP selects whether the LCD RAM contents or blank data are to be displayed. • When DISP = 0, blank data is displayed • When DISP = 1, LCD RAM data is displayed	H'FFC1 Bit 4	1

Table 4 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Settings
LCR	CKS3 to CKS0	LCD control register (Frame frequency select 3 to 0) Select the operating clock and the frame frequency. • When CKS3 = 0, CKS2 = *, CKS1 = 0 and CKS0 = 0, ϕ_w is selected as operating clock • When CKS3 = 0, CKS2 = *, CKS1 = 0 and CKS0 = 1, $\phi_w/2$ is selected as operating clock • When CKS3 = 0, CKS2 = *, CKS1 = 1 and CKS0 = *, $\phi_w/4$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 0 and CKS0 = 0, $\phi/2$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 0 and CKS0 = 1, $\phi/4$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 1 and CKS0 = 0, $\phi/8$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 1 and CKS0 = 1, $\phi/16$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 0 and CKS0 = 0, $\phi/32$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 0 and CKS0 = 1, $\phi/64$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 1 and CKS0 = 0, $\phi/128$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 1 and CKS0 = 1, $\phi/256$ is selected as operating clock *: Don’t care	H'FFC1 Bit 3 to bit 0	CKS3 = 1 CKS2 = 1 CKS1 = 1 CKS0 = 0

Table 4 Explanation of Internal Registers Used (cont)

Register	Name	Description	RAM Address	Settings
LCR2	LCDAB	LCD control register 2 (A waveform/B waveform switching control) Selects whether the A or B waveform is to be used for LCD driving. • When LCDAB = 0, the LCD is driven using the A waveform • When LCDAB = 1, the LCD is driven using the B waveform	H'FFC2 Bit 7	0
LCR2	SUPS	LCD control register 2 (Drive power supply select, step-up constant-voltage (5 V) power supply control) When V_{CC} is selected as the drive power supply, the step-up constant-voltage (5 V) power supply operation is halted; when 5 V is selected as the drive power supply, the step-up constant-voltage (5 V) power supply operates. • When SUPS = 0, the drive power supply is V_{CC}, and the step-up constant-voltage (5 V) power supply operation is halted • When SUPS = 1, the drive power supply is 5 V, and the step-up constant-voltage (5 V) power supply operates	H'FFC2 Bit 4	1

Table 4 Explanation of Internal Registers Used (cont)

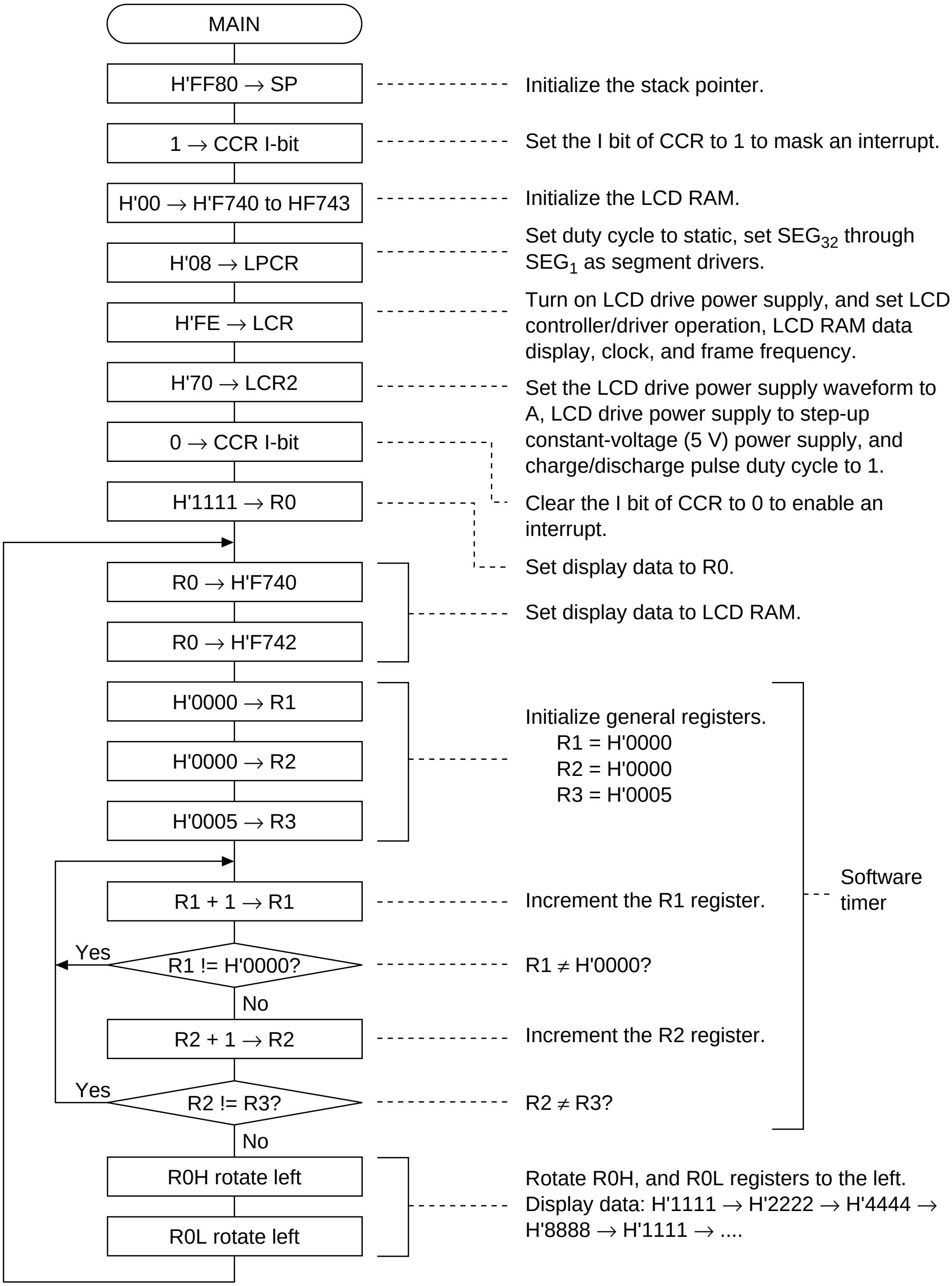
Register Name		Description	RAM Address	Settings
LCR2	CDS3 to CDS0	LCD control register 2 (Charge/discharge pulse duty cycle select 3 to 0) Select the duty cycle while the power supply split-resistance is connected to the power supply circuit. When CDS3 = 0, CDS2 = 0, CDS1 = 0 and CDS0 = 0, the duty cycle is 1. • When CDS3 = 0, CDS2 = 0, CDS1 = 0 and CDS0 = 1, the duty cycle is 1/8 • When CDS3 = 0, CDS2 = 0, CDS1 = 1 and CDS0 = 0, the duty cycle is 2/8 • When CDS3 = 0, CDS2 = 0, CDS1 = 1 and CDS0 = 1, the duty cycle is 3/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 0 and CDS0 = 0, the duty cycle is 4/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 0 and CDS0 = 1, the duty cycle is 5/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 1 and CDS0 = 0, the duty cycle is 6/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 1 and CDS0 = 1, the duty cycle is 0 • When CDS3 = 1, CDS2 = 0, CDS1 = * and CDS0 = *, the duty cycle is 1/16 • When CDS3 = 1, CDS2 = 1, CDS1 = * and CDS0 = *, the duty cycle is 1/32 *: Don’t care	H'FFC2 Bit 3 to bit 0	CDS3 = 0 CDS2 = 0 CDS1 = 0 CDS0 = 0

4. Explanation of RAM Usage

In this task example, RAM is not used.

Flowchart

1. Main routine



Program Lists

```
;*********************************************************************
;*   H8/3867 Application Note
;*
;*     'Liquid Crystal Display
;*     -Static Drive, Internal Driver-'
;*
;*            Function : LCD Controller / Driver
;*
;*            External Clock : 6MHz
;*            Internal Clock : 3MHz
;*            Sub Clock      : 32.768kHz
;*********************************************************************
;
        .cpu      300l
;
;*********************************************************************
;*      Symbol Defnition
;*********************************************************************
;
LPCR    .equ      h'ffc0      ;LCD Port Control Register
LCR     .equ      h'ffc1      ;LCD Control Register
LCR2    .equ      h'ffc2      ;LCD Control Register 2
;
;*********************************************************************
;*      Vector Address
;*********************************************************************
;
        .org      h'0000
        .data.w   MAIN        ;No.0 Reset Interrupt(H'0000-H'0001)
;
        .org      h'0008
        .data.w   MAIN        ;No.4 _IRQ0 Interrupt(H'0008-H'0009)
        .data.w   MAIN        ;No.5 _IRQ1 Interrupt(H'000A-H'000B)
        .data.w   MAIN        ;No.6 _IRQ2 Interrupt(H'000C-H'000D)
        .data.w   MAIN        ;No.7 _IRQ3 Interrupt(H'000E-H'000F)
        .data.w   MAIN        ;No.8 _IRQ4 Interrupt(H'0010-H'0011)
```

```
        .data.w   MAIN         ;No.9 _WKP0-_WKP7 Interrupt(H'0012-H'0013)
;
        .org      h'0016
        .data.w   MAIN         ;No.11 Timer A Interrupt(H'0016-H'0017)
        .data.w   MAIN         ;No.12 AEC Interrupt(H'0018-H'0019)
        .data.w   MAIN         ;No.13 Timer C Interrupt(H'001A-H'001B)
        .data.w   MAIN         ;No.14 Timer FL Interrupt(H'001C-H'001D)
        .data.w   MAIN         ;No.15 Timer FH Interrupt(H'001E-H'001F)
        .data.w   MAIN         ;No.16 Timer G Interrupt(H'0020-H'0021)
        .data.w   MAIN         ;No.17 SCI31 Interrupt(H'0022-H'0023)
        .data.w   MAIN         ;No.18 SCI32 Interrupt(H'0024-H'0025)
        .data.w   MAIN         ;No.19 A/D Converter Interrupt(H'0026-H'0028)
        .data.w   MAIN         ;No.20 Direct Transfer Interrupt(H'0028-H'0029)
;
;***********************************************************
;*      MAIN : Main Routine
;***********************************************************
;
        .org      h'1000
;
MAIN:   .equ      $
        mov.w     #h'ff80,sp   ;Initialize Stack Pointer
        orc       #h'80,ccr    ;Interrupt Disable
;
        mov.w     #h'0000,r0   ;Initialize LCD RAM
        mov.w     r0,@(h'f740)
        mov.w     r0,@(h'f742)
;
        mov.b     #h'08,r0l    ;Initialize LCD Controller/Driver
        mov.b     r0l,@LPCR
        mov.b     #h'fe,r0l
        mov.b     r0l,@LCR
        mov.b     #h'70,r0l
        mov.b     r0l,@LCR2
;
        andc      #h'7f,ccr    ;Interrupt REnable
;
```

```
        mov.w     #h'1111,r0   ;Set LCD RAM
MAIN99: mov.w     r0,@(h'f740)
        mov.w     r0,@(h'f742)
;
        sub.w     r1,r1        ;Set Software Timer
        sub.w     r2,r2
        mov.w     #h'0005,r3
INC:    adds      #1,r1
        mov.w     r1,r1
        bne       INC
;
        adds      #1,r2
        cmp.w     r2,r3
        bne       INC
;
        rotl      r0h          ;Display Data Rotate Left
        rotl      r0l
;
        bra       MAIN99
;
        .end
```

2.4 LCD Display with 1/4 Duty

LCD Display with 1/4 Duty	MCU: H8/3867 Series	Functions Used: LCD Controller/Driver

Specifications

1. LCD display is performed using the segment-type LCD controller circuit, LCD driver, and power supply circuit of the H8/3867 Series.
2. Four common signals and 32 segment signals are used for LCD display with 1/4 duty.
3. As the power supply driving the LCD, a step-up constant-voltage power supply (5 V) is used.
4. An example of LCD module connection and an LCD display example for this task example appear in figure 1.

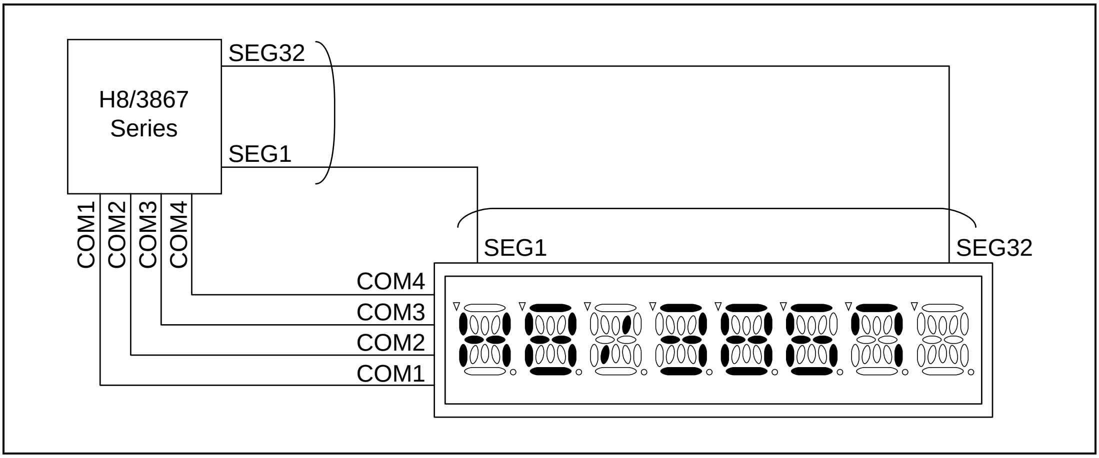


Figure 1 LCD Display Example

Explanation of Functions Used

1. In this task example, the LCD controller/driver is used for LCD display. The features of the LCD controller/driver are described below.

    - Display capacity
        a. Duty cycle: static
        Internal driver: 32 segments
        Segment external-expansion driver: 256 segments
        b. Duty cycle: 1/2
        Internal driver: 32 segment
        Segment external-expansion driver: 128 segments
        c. Duty cycle: 1/3
        Internal driver: 32 segment
        Segment external-expansion driver: 64 segments
        d. Duty cycle: 1/4
        Internal driver: 32 segment
        Segment external-expansion driver: 64 segments
    - LCD RAM capacity: 8 bits × 32 bytes (256 bits)
    - LCD RAM is word-accessible.
    - All segment output pins can be used as port pins in eight-pin units.
    - Depending on the duty cycle, the common output pins not used can be used for a common double-buffer (parallel connection).
    - Display is possible in all operating modes other than standby mode.
    - Frame frequency can be selected from among 11 values.
    - A power supply split-resistance is built-in, for supply of LCD driver power.
    - Use of module standby mode enables a module to be placed in standby mode independently when not used.
    - An internal step-up constant-voltage (5 V) power supply enables LCD display even at low voltages.
    - A or B waveform can be selected by software.

2. Figure 2 is a block diagram of the LCD controller/driver used in this task examples.

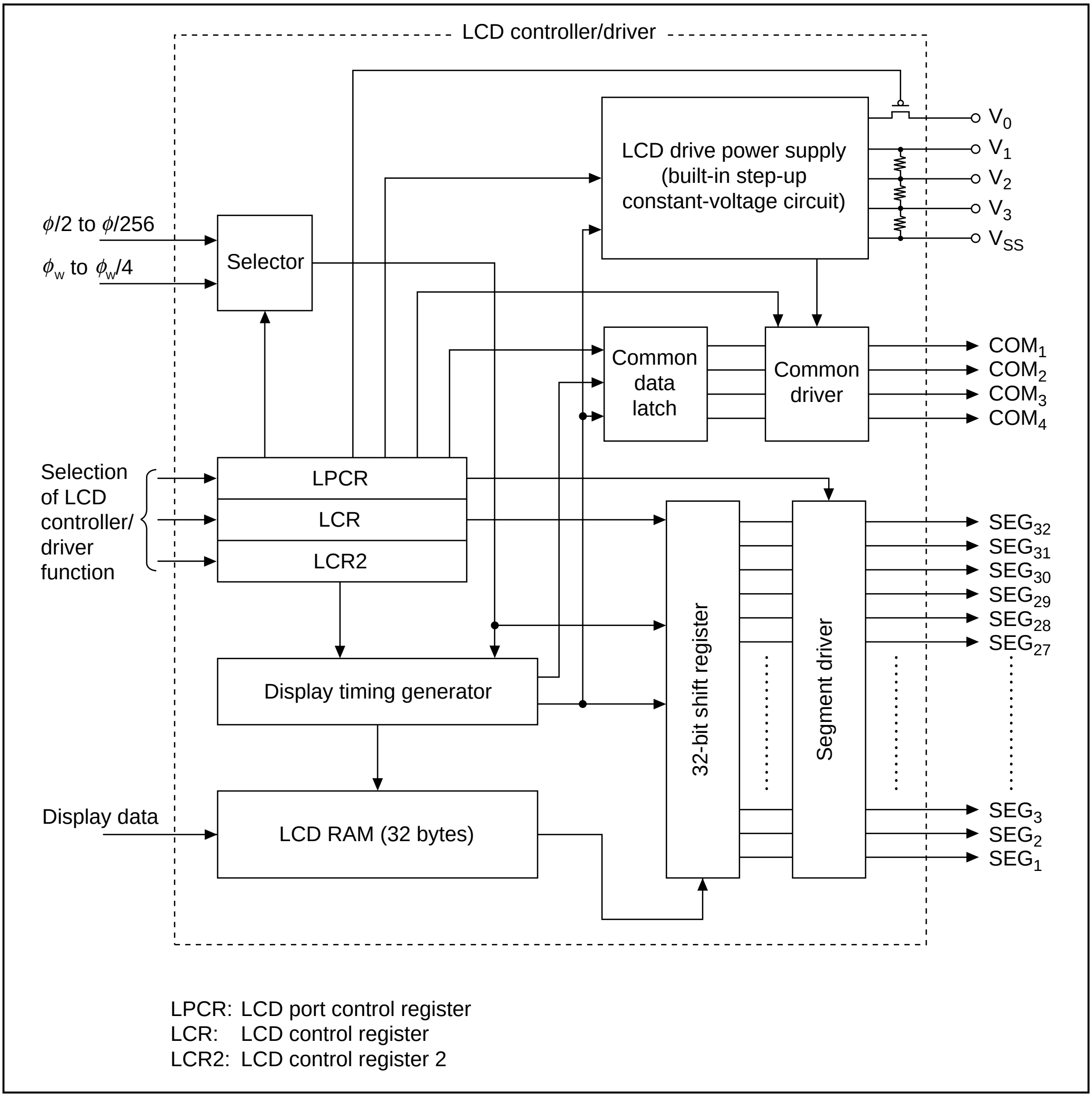


Figure 2 Block Diagram of LCD Controller/Driver (LCD Display with 1/4 Duty)

3. Functions of the LCD controller/driver are explained in table 1.

Table 1 LCD Controller/Driver Functions

LCD port control register (LPCR)	
Function	LPCR is an 8-bit read/write register which selects the duty cycle, the LCD driver and pin functions. LPCR is initialized to H'00 upon reset.
LCD control register (LCR)	
Function	LCR is an 8-bit read/write register which turns the LCD drive power supply on and off, controls display data, and selects the frame frequency. LCR is initialized to H'80 upon reset.
LCD control register 2 (LCR2)	
Function	LCR2 is an 8-bit read/write register which controls switching between A and B waveforms, selects the driver power supply, controls the step-up constant-voltage (5 V) power supply, and selects the duty cycle for charge/discharge pulses controlling disconnection of the power supply split-resistance from power supply circuit. LCR2 is initialized to H'60 upon reset.
Segment output pins (SEG_{32} to SEG_1)	
Function	These are pins used for LCD segment driving; all these pins are multiplexed as port pins, and their functions can be selected programmably.
Common output pins (COM_4 to COM_1)	
Function	These are LCD common driving output pins; under static or 1/2-duty driving, they can be configured in parallel.
Segment external expansion signal pin (CL_1)	
Function	This is a display data latch clock pin which is multiplexed as SEG_{32}.
Segment external expansion signal pin (CL_2)	
Function	This is a display data shift clock pin which is multiplexed as SEG_{31}.
Segment external expansion signal pin (M)	
Function	This is an LCD alternation signal pin which is multiplexed as SEG_{29}.
Segment external expansion signal pin (DO)	
Function	This is a serial display data signal pin which is multiplexed as SEG_{30}.
LCD power supply pins (V_0 to V_3)	
Function	These pins are used when connecting an external bypass capacitor or when using an external power supply circuit.
LCD RAM	
Function	Sets the display data. The relation between the LCD RAM and the display segments differs depending on the duty cycle. After the registers necessary for display have been set, instructions similar to the instructions for normal RAM are used to write data corresponding to the duty, and when the display is turned on, display is started automatically. Word/byte access instructions can be used to set data in the LCD RAM.

4. In this task example, a 16-line 8-character segment LCD is used for display with 1/4 duty driving. Figure 3 is a diagram showing connections for segment signals and common signals of the 16-line 8-character segment LCD used in this task example.

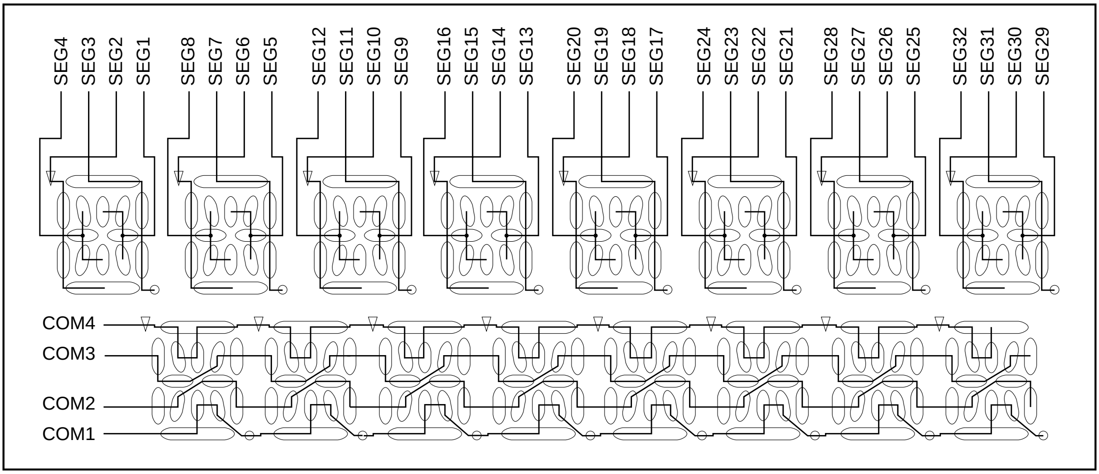


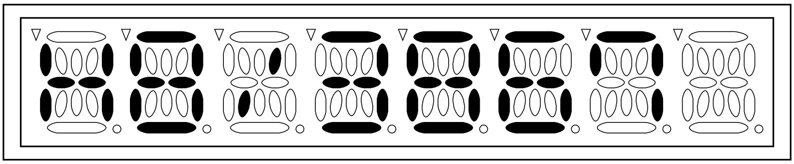
Figure 3 Connections of Segment Signals and Common Signals of the 16-Line 8-Character Segment LCD Used in this Task Example

5. Figure 4 shows the LCD RAM mapping under 1/4 duty driving without segment external expansion.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
H'F740	SEG_2	SEG_2	SEG_2	SEG_2	SEG_1	SEG_1	SEG_1	SEG_1
H'F741	SEG_4	SEG_4	SEG_4	SEG_4	SEG_3	SEG_3	SEG_3	SEG_3
H'F742	SEG_6	SEG_6	SEG_6	SEG_6	SEG_5	SEG_5	SEG_5	SEG_5
H'F743	SEG_8	SEG_8	SEG_8	SEG_8	SEG_7	SEG_7	SEG_7	SEG_7
H'F744	SEG_{10}	SEG_{10}	SEG_{10}	SEG_{10}	SEG_9	SEG_9	SEG_9	SEG_9
H'F745	SEG_{12}	SEG_{12}	SEG_{12}	SEG_{12}	SEG_{11}	SEG_{11}	SEG_{11}	SEG_{11}
H'F746	SEG_{14}	SEG_{14}	SEG_{14}	SEG_{14}	SEG_{13}	SEG_{13}	SEG_{13}	SEG_{13}
H'F747	SEG_{16}	SEG_{16}	SEG_{16}	SEG_{16}	SEG_{15}	SEG_{15}	SEG_{15}	SEG_{15}
H'F748	SEG_{18}	SEG_{18}	SEG_{18}	SEG_{18}	SEG_{17}	SEG_{17}	SEG_{17}	SEG_{17}
H'F749	SEG_{20}	SEG_{20}	SEG_{20}	SEG_{20}	SEG_{19}	SEG_{19}	SEG_{19}	SEG_{19}
H'F74A	SEG_{22}	SEG_{22}	SEG_{22}	SEG_{22}	SEG_{21}	SEG_{21}	SEG_{21}	SEG_{21}
H'F74B	SEG_{24}	SEG_{24}	SEG_{24}	SEG_{24}	SEG_{23}	SEG_{23}	SEG_{23}	SEG_{23}
H'F74C	SEG_{26}	SEG_{26}	SEG_{26}	SEG_{26}	SEG_{25}	SEG_{25}	SEG_{25}	SEG_{25}
H'F74D	SEG_{28}	SEG_{28}	SEG_{28}	SEG_{28}	SEG_{27}	SEG_{27}	SEG_{27}	SEG_{27}
H'F74E	SEG_{30}	SEG_{30}	SEG_{30}	SEG_{30}	SEG_{29}	SEG_{29}	SEG_{29}	SEG_{29}
H'F74F	SEG_{32}	SEG_{32}	SEG_{32}	SEG_{32}	SEG_{31}	SEG_{31}	SEG_{31}	SEG_{31}
	↓ COM_4	↓ COM_3	↓ COM_2	↓ COM_1	↓ COM_4	↓ COM_3	↓ COM_2	↓ COM_1

Figure 4 LCD RAM Mapping under 1/4 Duty Driving without Segment External Expansion

6. Figure 5 shows the relation between the 16-line 8-character segment LCD display and LCD RAM settings used in this task example. As shown in the figure, by setting the LCD RAM appropriately, “H8/3867” is displayed on the 16-line 8-character segment LCD.

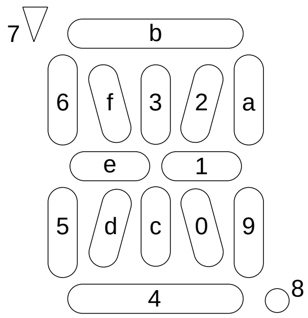


	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'F740	0	1	1	0	0	0	1	0	Display data for “H”
H'F741	0	1	0	0	0	1	1	0	
H'F742	0	1	1	1	0	0	1	0	Display data for “8”
H'F743	0	1	0	0	1	1	1	0	
H'F744	0	0	0	0	0	1	0	0	Display data for “/”
H'F745	0	0	1	0	0	0	0	0	
H'F746	0	0	0	1	0	0	1	0	Display data for “3”
H'F747	0	1	0	0	1	1	1	0	
H'F748	0	1	1	1	0	0	1	0	Display data for “8”
H'F749	0	1	0	0	1	1	1	0	
H'F74A	0	1	1	1	0	0	1	0	Display data for “6”
H'F74B	0	1	0	0	1	0	1	0	
H'F74C	0	1	0	0	0	0	0	0	Display data for “7”
H'F74D	0	0	0	0	1	1	1	0	
H'F74E	0	0	0	0	0	0	0	0	Display data for “ ” *
H'F74F	0	0	0	0	0	0	0	0	

Note: *: “ ” denotes a blank (nothing displayed)

Figure 5 Relation between LCD Display and LCD RAM Settings

7. Figure 6 shows the relation between the display of the eighth column from the right in the 16-line 8-character segment LCD and the LCD RAM corresponding to SEG_1 through SEG_4. As indicated in figure 6, when the LCD RAM bits corresponding to 0 through f are set to 1, the LCD regions are lit; when they are cleared to 0, the LCD regions are unlit.



	SEG_2				SEG_1			
H'F740	7	6	5	4	3	2	1	0
H'F741	f	e	d	c	b	a	9	8
	SEG_4				SEG_3			
	COM_4	COM_3	COM_2	COM_1	COM_4	COM_3	COM_2	COM_1

Figure 6 Relation between LCD Lit/Unlit States and Corresponding LCD RAM Settings

8. Table 2 shows examples of SEG_1 through SEG_4 display and display data for a 16-line 8-character segment LCD.

Table 2 Display Data Examples

Symbol	Display	Address	Display Data								Symbol	Display	Address	Display Data							
		H'F740	0	0	0	0	0	0	0	0	G		H'F740	0	1	1	1	0	0	1	0
		H'F741	0	0	0	0	0	0	0	0			H'F741	0	0	0	0	1	0	1	0
*		H'F740	0	0	0	0	1	1	1	1	H		H'F740	0	1	1	0	0	0	1	0
		H'F741	1	1	1	1	0	0	0	0			H'F741	0	1	0	0	0	1	1	0
–		H'F740	0	0	0	0	0	0	1	0	I		H'F740	0	0	0	1	1	0	0	0
		H'F741	0	1	0	0	0	0	0	0			H'F741	0	0	0	1	1	0	0	0
/		H'F740	0	0	0	0	0	1	0	0	J		H'F740	0	0	1	1	1	0	0	0
		H'F741	0	0	1	0	0	0	0	0			H'F741	0	0	0	1	1	0	0	0
0		H'F740	0	1	1	1	0	1	0	0	K		H'F740	0	1	1	0	0	1	0	1
		H'F741	0	0	1	0	1	1	1	0			H'F741	0	1	0	0	0	0	0	0
1		H'F740	0	0	0	0	0	0	0	0	L		H'F740	0	1	1	1	0	0	0	0
		H'F741	0	0	0	0	0	1	1	0			H'F741	0	0	0	0	0	0	0	0
2		H'F740	0	0	1	1	0	0	1	0	M		H'F740	0	1	1	0	0	1	0	0
		H'F741	0	1	0	0	1	1	0	0			H'F741	1	0	0	0	0	1	1	0
3		H'F740	0	0	0	1	0	0	1	0	N		H'F740	0	1	1	0	0	0	0	1
		H'F741	0	1	0	0	1	1	0	0			H'F741	1	0	0	0	0	1	1	0
4		H'F740	0	1	0	0	0	0	1	0	O		H'F740	0	1	1	1	0	0	0	0
		H'F741	0	1	0	0	0	1	1	0			H'F741	0	0	0	0	1	1	1	0
5		H'F740	0	1	0	1	0	0	1	0	P		H'F740	0	1	1	0	0	0	1	0
		H'F741	0	1	0	0	1	0	1	0			H'F741	0	1	0	0	1	1	9	0
6		H'F740	0	1	1	1	0	0	1	0	Q		H'F740	0	1	1	1	0	0	9	1
		H'F741	0	1	0	0	0	0	1	0			H'F741	0	0	0	0	1	1	1	0
7		H'F740	0	1	0	0	0	0	0	0	R		H'F740	0	1	1	0	0	0	1	1
		H'F741	0	0	0	0	1	1	1	0			H'F741	0	1	0	0	1	1	0	0
8		H'F740	0	1	1	1	0	0	1	0	S		H'F740	0	0	0	1	0	0	0	1
		H'F741	0	1	0	0	1	1	1	0			H'F741	1	0	0	0	1	0	0	0
9		H'F740	0	1	0	0	0	0	1	0	T		H'F740	0	0	0	0	1	0	0	0
		H'F741	0	1	0	0	1	1	1	0			H'F741	0	0	0	1	1	0	0	0
A		H'F740	0	1	1	0	0	0	1	0	U		H'F740	0	1	1	1	0	0	0	0
		H'F741	0	1	0	0	1	1	1	0			H'F741	0	0	0	0	0	1	1	0
B		H'F740	0	0	0	1	1	1	0	0	V		H'F740	0	1	1	0	0	1	0	0
		H'F741	0	0	0	1	1	1	1	0			H'F741	0	0	1	0	0	0	0	0
C		H'F740	0	1	1	1	0	0	0	0	W		H'F740	0	1	1	0	0	0	0	1
		H'F741	0	0	0	0	1	0	0	0			H'F741	0	0	1	0	0	1	1	0
D		H'F740	0	0	0	1	1	0	0	0	X		H'F740	0	0	0	0	0	1	0	1
		H'F741	0	0	0	1	1	1	1	0			H'F741	1	0	1	0	0	0	0	0
E		H'F740	0	1	1	1	0	0	1	0	Y		H'F740	0	0	0	0	0	1	0	0
		H'F741	0	1	0	0	1	0	0	0			H'F741	1	0	0	1	0	0	0	0
F		H'F740	0	1	1	0	0	0	1	0	Z		H'F740	0	0	0	1	0	1	0	0
		H'F741	0	1	0	0	1	0	0	0			H'F741	0	0	1	0	1	0	0	0

9. Table 3 indicates function allocations in this task example.

Table 3 Function Allocations

Function	Function Allocation
LPCR	Selects duty cycle, LCD driver, and pin functions.
LCR	Turns LCD drive power supply on and off, controls display data, and selects frame frequency.
LCR2	Switches between A and B waveforms, selects drive power supply, controls step-up constant-voltage (5 V) power supply, selects duty cycle for charge/discharge pulses to control disconnection of power supply split-resistance from power supply circuit.
SEG_{32} to SEG_1	Used as segment drivers.
COM_4 to COM_1	Used as common drivers.
V_0, V_1	The V_0 and V_1 pins are shorted in order to use the step-up constant-voltage (5 V) power supply as the LCD drive power supply.
LCD RAM	Sets the LCD display data.

Explanation of Functions Used

1. Hardware settings for LCD display are explained below.

   a. LCD drive power supply settings

      The H8/3867 Series can use either the internal power supply circuit or can use an external power supply circuit as the LCD drive power supply. In addition, either the power supply voltage (V_{CC}) or the step-up constant-voltage (5 V) can be selected for the internal power supply circuit.

      When using the internal power supply circuit to drive the LCD, the V_0 and V_1 pins are connected externally, as illustrated in figure 7.

      In this task example, the step-up constant-voltage power supply is used as the LCD drive power supply.

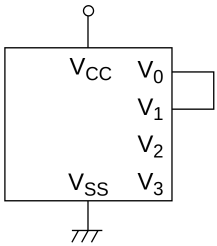


Figure 7 Example of Connection of LCD Power Supply Pins When Using Internal Power Supply Circuit

b. Contrast control function

A block diagram of the LCD drive power supply circuit appears in figure 8. Either V_{CC} or a 5 V output from the step-up constant-voltage power supply circuit is output to pin V_0. When these voltages are used directly to drive the LCD, the V_0 and V_1 pins should be shorted. By inserting a variable resistance R between the V_0 and V_1 pins, the voltage applied to the V_1 pin can be adjusted, and the LCD panel contrast can be controlled.

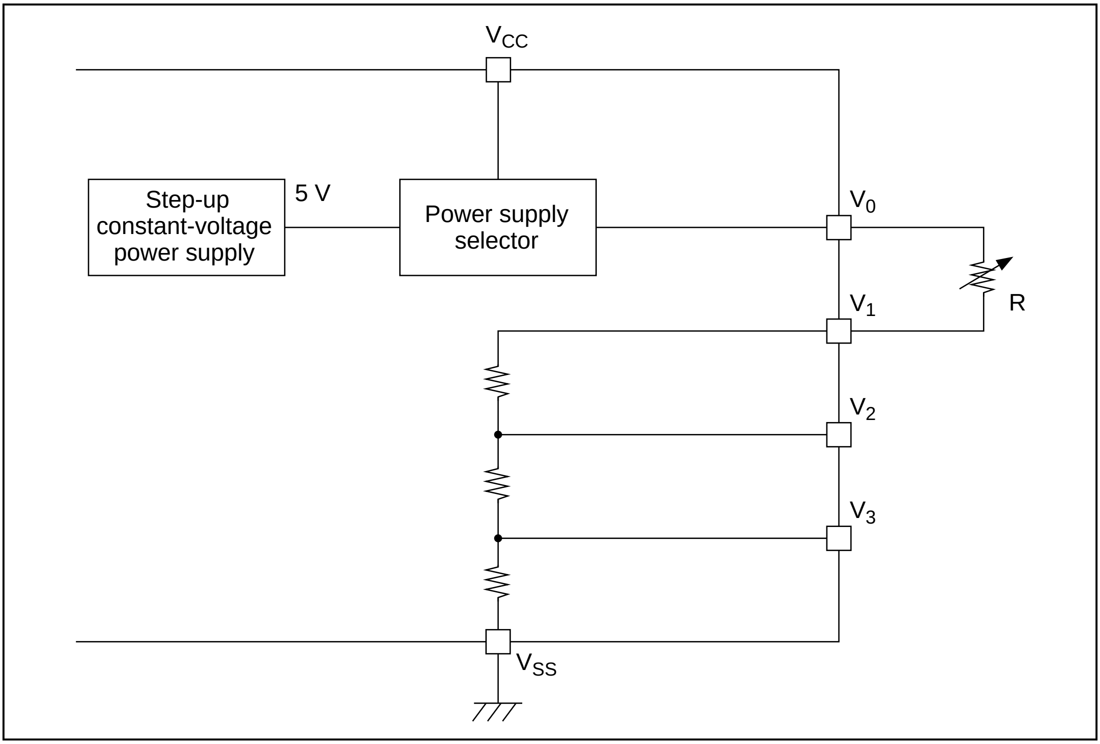


Figure 8 Block Diagram of LCD Drive Power Supply Circuit

c. Step-up constant-voltage (5 V) power supply

The H8/3867 Series has an internal step-up constant-voltage (5 V) power supply, supplying a constant 5 V independent of V_{CC}.

By setting SUPS of the LCD control register 2 (LCR2) to 1, the step-up constant-voltage (5 V) power supply is activated, and a constant 5 V is output to the V_0 pin. This can be used either with pins V_0 and V_1 short-circuited, or with a resistance inserted to divide the voltage.

Note: The step-up constant-voltage (5 V) power supply must not be used for purposes other than to drive the LCD. In addition, when driving a large panel, the power supply capacity may be insufficient. In such cases, either V_{CC} or an external power supply circuit can be used as the power supply.

2. Software settings for LCD display are explained below.

a. Duty selection

DTS1 and DTS0 are used to select from among static, 1/2 duty, 1/3 duty, and 1/4 duty.

b. Segment driver selection

SGS3 to SGS0 are used to select the segment drivers to be used.

c. Frame frequency selection

By setting CKS3 to CKS0, the frame frequency can be selected. The frame frequency should be selected according to the LCD panel.

d. Selection of A and B waveforms

LCDAB can be used to select either the A or the B waveform for use as the LCD waveform.

e. Selection of LCD drive power supply

When using the internal power supply circuit, SUPS can be used to select the power supply to be used. When using an external power supply circuit, SUPS is used to select V_{CC}, and PSW should be used to turn off the LCD drive power supply.

3. Figure 9 shows the operation principle of this task example.

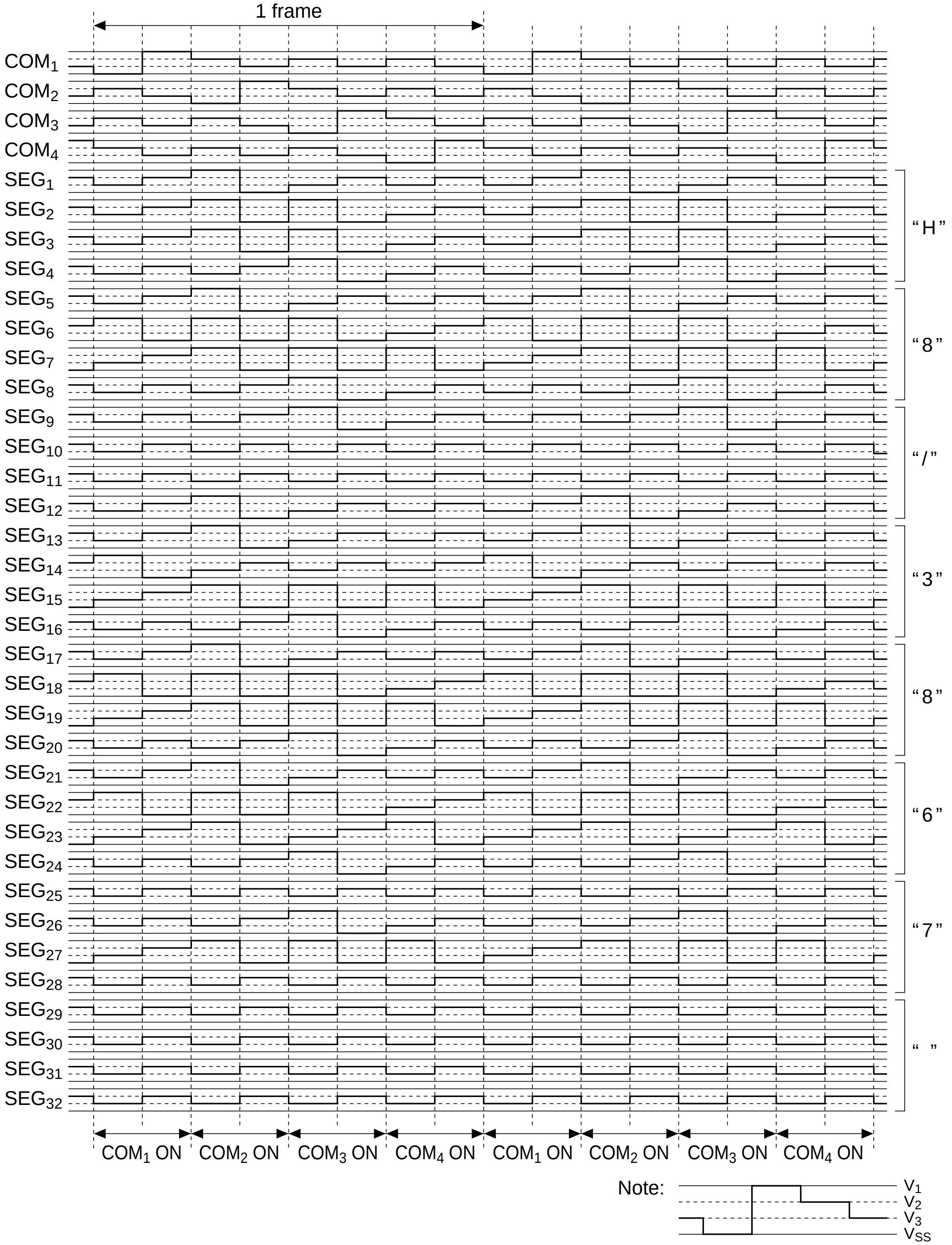


Figure 9 Operation Principle

Explanation of Software

1. Explanation of Modules

   Table 4 explains the modules in this task example.

Table 4 Module Explanation

Module Name	Label Name	Function
Main routine	MAIN	Initializes the stack pointer, LCD RAM, and LCD controller/driver, and enables interrupts.

2. Explanation of Arguments

   In this task example, no arguments are used.

3. Explanation of Internal Registers Used

   Table 5 gives explanation of the internal registers used in this task example.

Table 5 Explanation of Internal Registers Used

Register Name		Description	RAM Address	Settings
LPCR	DTS1, DTS0	LCD port control register (Duty cycle selection 1, 0) Select duty from among static, 1/2 duty, 1/3 duty, and 1/4 duty. • When DTS1 = 0 and DTS0 = 0, static duty is selected • When DTS1 = 0 and DTS0 = 1, 1/2 duty is selected • When DTS1 = 1 and DTS0 = 0, 1/3 duty is selected • When DTS1 = 1 and DTS0 = 1, 1/4 duty is selected	H'FFC0 Bit 7 and bit 6	DTS1 = 1 DTS0 = 1

Table 5 Explanation of Internal Registers Used (cont)

Register	Name	Description	RAM Address	Settings
LPCR	CMX	LCD port control register (Common function selection) Selects whether the same waveform is output from several pins in order to increase the common driving capacity, when common pins are not selected for a given duty cycle. • When CMX = 0, the same waveform is not output from multiple common pins not used at that duty cycle • When CMX = 1, the same waveform is output from multiple common pins not used at that duty cycle	H'FFC0 Bit 5	0
LPCR	SGX	LCD port control register (Expansion signal select) Selects whether the SEG_{32}/CL_1, SEG_{31}/CL_2, $SEG_{30}/D0$, and SEG_{29}/M pins are used as segment pins (SEG_{32} through SEG_{29}), or as segment external expansion signal pins (CL_1, CL_2, D0, M). • When SGX = 0, they are used as segment pins (SEG_{32} through SEG_{29}) • When SGX = 1, they are used as segment external expansion signal pins (CL_1, CL_2, D0, M)	H'FFC0 Bit 4	0

Table 5 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Settings
LPCR	SGS3 to SGS0	LCD port control register (Segment driver selection) Select the segment driver to be used. • When SGX = 0, SGS3 = 0, SGS2 = 0, SGS1 = 0 and SGS0 = 0, pins SEG_{32} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 0, SGS1 = 0 and SGS0 = 1, pins SEG_{32} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 0, SGS1 = 1 and SGS0 = *, pins SEG_{32} through SEG_{25} function as segment drivers and pins SEG_{24} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 1, SGS1 = 0 and SGS0 = *, pins SEG_{32} through SEG_{17} function as segment drivers and pins SEG_{16} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 1, SGS1 = 1 and SGS0 = *, pins SEG_{32} through SEG_9 function as segment drivers and pins SEG_8 through SEG_1 function as ports • When SGX = 0, SGS3 = 1, SGS2 = *, SGS1 = * and SGS0 = *, pins SEG_{32} through SEG_1 function as segment drivers • When SGX = 1, SGS3 = 0, SGS2 = 0, SGS1 = 0 and SGS0 = 0, pins SEG_{32} through SEG_{29} function as external expansion pins and pins SEG_{28} through SEG_1 function as ports • SGX = 1, SGS3 = *, SGS2 = *, SGS1 = * and SGS0 = * cannot be specified *: Don’t care	H'FFC0 Bit 3 to bit 0	SGS3 = 1 SGS2 = 0 SGS1 = 0 SGS0 = 0

Table 5 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Settings
LCR	PSW	LCD control register (LCD drive power supply on/off control) Turns the LCD drive power supply off when LCD display is not used in power-down mode, or when an external power supply is used. When the ACT bit is cleared to 0, or when in standby mode, the LCD drive power supply is turned off regardless of this bit setting. • When PSW = 0, the LCD drive power supply is turned off • When PSW = 1, the LCD drive power supply is turned on	H'FFC1 Bit 6	1
LCR	ACT	LCD control register (Display function activate) Selects whether the LCD controller/driver is to be used or not. By clearing this bit to 0, LCD controller/driver operation is halted. Also, regardless of the value of PSW, the LCD drive power supply is turned off. However, the register contents are maintained. • When ACT = 0, LCD controller/driver operation is halted • When ACT = 1, LCD controller/driver operations	H'FFC1 Bit 5	1
LCR	DISP	LCD control register (Display data control) DISP selects whether the LCD RAM contents or blank data are to be displayed. • When DISP = 0, blank data is displayed • When DISP = 1, LCD RAM data is displayed	H'FFC1 Bit 4	1

Table 5 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Settings
LCR	CKS3 to CKS0	LCD control register (Frame frequency select 3 to 0) Select the operating clock and the frame frequency. • When CKS3 = 0, CKS2 = *, CKS1 = 0 and CKS0 = 0, ϕ_w is selected as operating clock • When CKS3 = 0, CKS2 = *, CKS1 = 0 and CKS0 = 1, $\phi_w/2$ is selected as operating clock • When CKS3 = 0, CKS2 = *, CKS1 = 1 and CKS0 = *, $\phi_w/4$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 0 and CKS0 = 0, $\phi/2$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 0 and CKS0 = 1, $\phi/4$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 1 and CKS0 = 0, $\phi/8$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 1 and CKS0 = 1, $\phi/16$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 0 and CKS0 = 0, $\phi/32$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 0 and CKS0 = 1, $\phi/64$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 1 and CKS0 = 0, $\phi/128$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 1 and CKS0 = 1, $\phi/256$ is selected as operating clock *: Don’t care	H'FFC1 Bit 3 to bit 0	CKS3 = 1 CKS2 = 1 CKS1 = 1 CKS0 = 0

Table 5 Explanation of Internal Registers Used (cont)

Register	Name	Description	RAM Address	Settings
LCR2	LCDAB	LCD control register 2 (A waveform/B waveform switching control) Selects whether the A or B waveform is to be used for LCD driving. • When LCDAB = 0, the LCD is driven using the A waveform • When LCDAB = 1, the LCD is driven using the B waveform	H'FFC2 Bit 7	0
LCR2	SUPS	LCD control register 2 (Drive power supply select, step-up constant-voltage (5 V) power supply control) When V_{CC} is selected as the drive power supply, the step-up constant-voltage (5 V) power supply operation is halted; when 5 V is selected as the drive power supply, the step-up constant-voltage (5 V) power supply operates. • When SUPS = 0, the drive power supply is V_{CC}, and the step-up constant-voltage (5 V) power supply operation is halted • When SUPS = 1, the drive power supply is 5 V, and the step-up constant-voltage (5 V) power supply operates.	H'FFC2 Bit 4	1

Table 5 Explanation of Internal Registers Used (cont)

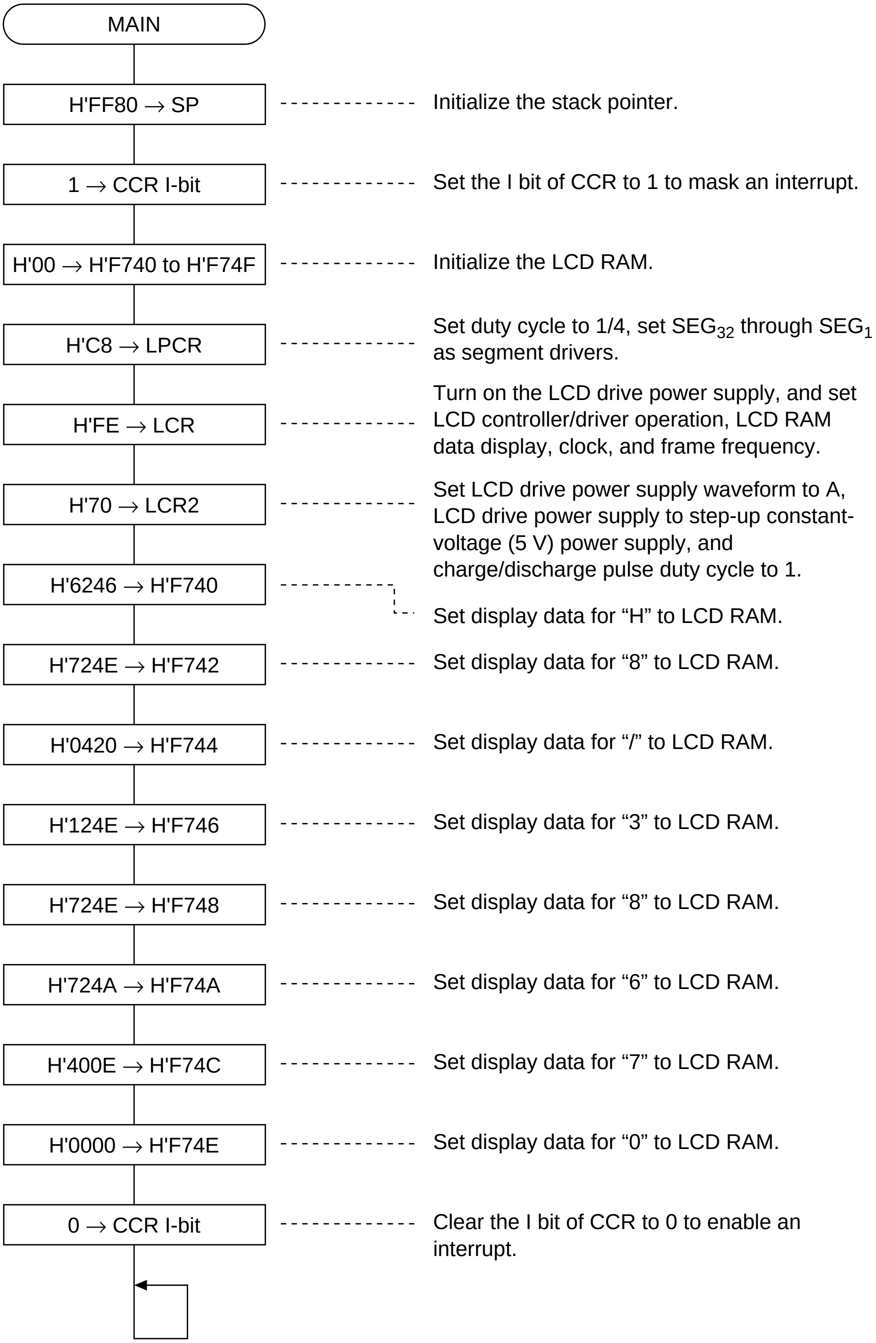
Register Name		Description	RAM Address	Settings
LCR2	CDS3 to CDS0	LCD control register 2 (Charge/discharge pulse duty cycle selection 3 to 0) Select the duty cycle while the power supply split-resistance is connected to the power supply circuit. • When CDS3 = 0, CDS2 = 0, CDS1 = 0 and CDS0 = 0, the duty cycle is 1 • When CDS3 = 0, CDS2 = 0, CDS1 = 0 and CDS0 = 1, the duty cycle is 1/8 • When CDS3 = 0, CDS2 = 0, CDS1 = 1 and CDS0 = 0, the duty cycle is 2/8 • When CDS3 = 0, CDS2 = 0, CDS1 = 1 and CDS0 = 1, the duty cycle is 3/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 0 and CDS0 = 0, the duty cycle is 4/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 0 and CDS0 = 1, the duty cycle is 5/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 1 and CDS0 = 0, the duty cycle is 6/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 1 and CDS0 = 1, the duty cycle is 0 • When CDS3 = 1, CDS2 = 0, CDS1 = * and CDS0 = *, the duty cycle is 1/16 • When CDS3 = 1, CDS2 = 1, CDS1 = * and CDS0 = *, the duty cycle is 1/32 *: Don’t care	H'FFC2 Bit 3 to bit 0	CDS3 = 0 CDS2 = 0 CDS1 = 0 CDS0 = 0

4. Explanation of RAM Usage

In this task example, RAM is not used.

Flowchart

1. Main routine



Program Lists

```
;*************************************************************
;*   H8/3867 Application Note
;*
;*       'Liquid Crystal Display
;*        -1/4 Duty Drive, Internal Driver-'
;*
;*              Function : LCD Controller / Driver
;*
;*              External Clock : 6MHz
;*              Internal Clock : 3MHz
;*              Sub Clock      : 32.768kHz
;*************************************************************
;
        .cpu      300l
;
;*************************************************************
;*      Symbol Defnition
;*************************************************************
;
LPCR    .equ      h'ffc0       ;LCD Port Control Register
LCR     .equ      h'ffc1       ;LCD Control Register
LCR2    .equ      h'ffc2       ;LCD Control Register 2
;
;*************************************************************
;*      Vector Address
;*************************************************************
;
        .org      h'0000
        .data.w   MAIN         ;No.0 Reset Interrupt(H'0000-H'0001)
;
        .org      h'0008
        .data.w   MAIN         ;No.4 _IRQ0 Interrupt(H'0008-H'0009)
        .data.w   MAIN         ;No.5 _IRQ1 Interrupt(H'000A-H'000B)
        .data.w   MAIN         ;No.6 _IRQ2 Interrupt(H'000C-H'000D)
        .data.w   MAIN         ;No.7 _IRQ3 Interrupt(H'000E-H'000F)
        .data.w   MAIN         ;No.8 _IRQ4 Interrupt(H'0010-H'0011)
```

```
        .data.w   MAIN          ;No.9 _WKP0-_WKP7 Interrupt(H'0012-H'0013)
;
        .org      h'0016
        .data.w   MAIN          ;No.11 Timer A Interrupt(H'0016-H'0017)
        .data.w   MAIN          ;No.12 AEC Interrupt(H'0018-H'0019)
        .data.w   MAIN          ;No.13 Timer C Interrupt(H'001A-H'001B)
        .data.w   MAIN          ;No.14 Timer FL Interrupt(H'001C-H'001D)
        .data.w   MAIN          ;No.15 Timer FH Interrupt(H'001E-H'001F)
        .data.w   MAIN          ;No.16 Timer G Interrupt(H'0020-H'0021)
        .data.w   MAIN          ;No.17 SCI31 Interrupt(H'0022-H'0023)
        .data.w   MAIN          ;No.18 SCI32 Interrupt(H'0024-H'0025)
        .data.w   MAIN          ;No.19 A/D Converter Interrupt(H'0026-H'0028)
        .data.w   MAIN          ;No.20 Direct Transfer Interrupt(H'0028-H'0029)
;
;************************************************************
;*      MAIN : Main Routine
;************************************************************
;
        .org      h'1000
;
MAIN:   .equ      $
        mov.w     #h'ff80,sp    ;Initialize Stack Pointer
        orc       #h'80,ccr     ;Interrupt Disable
;
        sub.b     r0l,r0l       ;Initialize LCD RAM
        mov.w     #h'f740,r1
        mov.w     #h'f750,r2
INIT:   mov.b     r0l,@r1
        adds      #1,r1
        cmp.w     r2,r1
        bne       INIT
;
        mov.b     #h'c8,r0l
        mov.b     r0l,@LPCR     ;Initialize LCD Port Control
        mov.b     #h'fe,r0l
        mov.b     r0l,@LCR      ;Initialize LCD Control
        mov.b     #h'70,r0l
```

```
        mov.b     r0l,@LCR2    ;Initialize LCD Control 2
;
        mov.w     #h'f740,r1   ;Set LCD RAM Start Address
        mov.w     #h'f750,r2   ;Set LCD RAM End Address
        mov.w     #h'1500,r3   ;Set LCD Data Address
DISP:   mov.w     @r3,r0       ;Load LCD Data
        mov.w     r0,@r1       ;Store LCD Data to LCD RAM
        adds      #2,r3        ;Increment LCD Data Address
        adds      #2,r1        ;Increment LCD RAM Address
        cmp.w     r2,r1        ;LCD RAM Address = LCD RAM End Address ?
        bne       DISP         ;No.
;
        andc      #h'7f,ccr    ;Interrupt Enable
EXIT:   bra       EXIT         ;Yes.
;
;************************************************************
;*      LCD Data Table
;************************************************************
;
        .org      h'1500
;
        .data.w   h'6246       ;"H"
        .data.w   h'724e       ;"8"
        .data.w   h'0420       ;"/"
        .data.w   h'124e       ;"3"
        .data.w   h'724e       ;"8"
        .data.w   h'724a       ;"6"
        .data.w   h'400e       ;"7"
        .data.w   h'0000       ;" "
;
        .end
```

2.5 LCD Display with Segment External Expansion

LCD Display with Segment External Expansion	MCU: H8/3867 Series	Functions Used: LCD Controller/Driver

Specifications

1. LCD display is performed using the segment-type LCD controller circuit, LCD driver, and power supply circuit of the H8/3867 Series.

2. By connecting an HD66100 to an H8/3867 Series for segment external expansion, data is displayed on the LCD.

3. Data is displayed on the 16-line 8-character segment LCD with 1/4 duty.

4. As the power supply driving the LCD, Vcc is used.

5. An example of LCD module to the HD66100 connection and an LCD display example for this task example appear in figure 1.

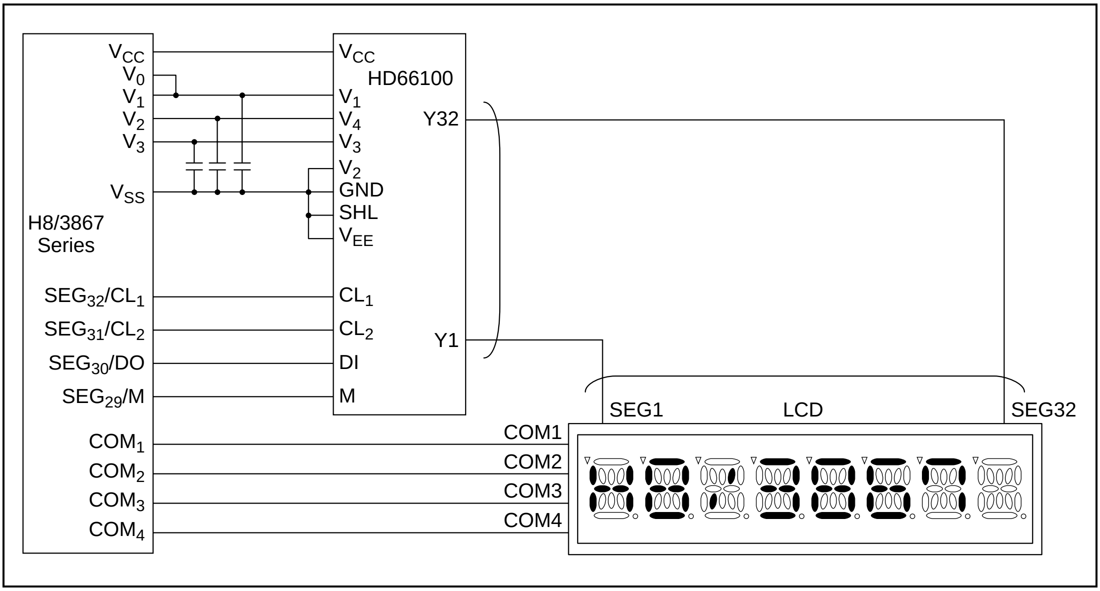


Figure 1 LCD Module to HD66100 Connection and LCD Display Example

Explanation of Functions Used

1. In this task example, the LCD controller/driver is used for LCD display. The features of the LCD controller/driver are described below.

   - Display capacity
     a. Duty cycle: static
        Internal driver: 32 segments
        Segment external-expansion driver: 256 segments
     b. Duty cycle: 1/2
        Internal driver: 32 segment
        Segment external-expansion driver: 128 segments
     c. Duty cycle: 1/3
        Internal driver: 32 segment
        Segment external-expansion driver: 64 segments
     d. Duty cycle: 1/4
        Internal driver: 32 segment
        Segment external-expansion driver: 64 segments
   - LCD RAM capacity: 8 bits × 32 bytes (256 bits)
   - LCD RAM is word-accessible.
   - All segment output pins can be used as port pins in eight-pin units.
   - Depending on the duty cycle, the common output pins not used can be used for a common double-buffer (parallel connection).
   - Display is possible in all operating modes other than standby mode.
   - Frame frequency can be selected from among 11 values.
   - A power supply split-resistance is built-in, for supply of LCD driver power.
   - Use of module standby mode enables a module to be placed in standby mode independently when not used.
   - An internal step-up constant-voltage (5 V) power supply enables LCD display even at low voltages.
   - A or B waveform can be selected by software.

2. Figure 2 is a block diagram of the LCD controller/driver used in this task examples.

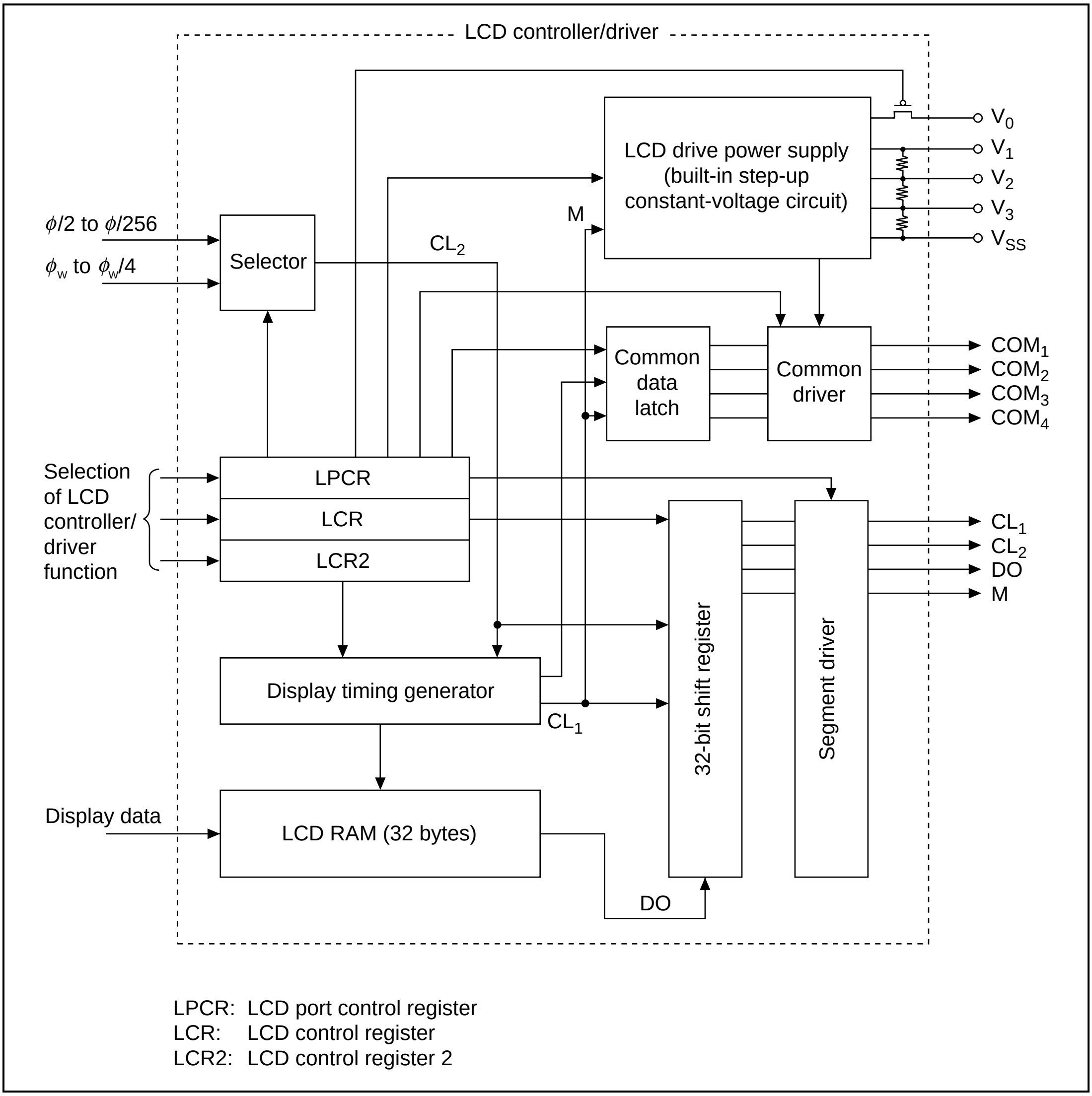


Figure 2 Block Diagram of LCD Controller/Driver
(LCD Display with Segment External Expansion)

3. Functions of the LCD controller/driver are explained in table 1.

Table 1 LCD Controller/Driver Functions

LCD port control register (LPCR)	
Function	LPCR is an 8-bit read/write register which selects the duty cycle, the LCD driver and pin functions. LPCR is initialized to H'00 upon reset.
LCD control register (LCR)	
Function	LCR is an 8-bit read/write register which turns the LCD drive power supply on and off, controls display data, and selects the frame frequency. LCR is initialized to H'80 upon reset.
LCD control register 2 (LCR2)	
Function	LCR2 is an 8-bit read/write register which controls switching between A and B waveforms, selects the driver power supply, controls the step-up constant-voltage (5 V) power supply, and selects the duty cycle for charge/discharge pulses controlling disconnection of the power supply split-resistance from power supply circuit. LCR2 is initialized to H'60 upon reset.
Common output pins (COM_4 to COM_1)	
Function	These are LCD common driving output pins; under static or 1/2-duty driving, they can be configured in parallel.
Segment external expansion signal pin (CL_1)	
Function	This is a display data latch clock pin which is multiplexed as SEG_{32}.
Segment external expansion signal pin (CL_2)	
Function	This is a display data shift clock pin which is multiplexed as SEG_{31}.
Segment external expansion signal pin (M)	
Function	This is an LCD alternation signal pin which is multiplexed as SEG_{29}.
Segment external expansion signal pin (DO)	
Function	This is a serial display data signal pin which is multiplexed as SEG_{30}.
LCD power supply pins (V_0 to V_3)	
Function	These pins are used when connecting an external bypass capacitor or when using an external power supply circuit.
LCD RAM	
Function	Sets the display data. The relation between the LCD RAM and the display segments differs depending on the duty cycle. After the registers necessary for display have been set, instructions similar to the instructions for normal RAM are used to write data corresponding to the duty, and when the display is turned on, display is started automatically. Word/byte access instructions can be used to set data in RAM.

4. In this task example, a 16-line 8-character segment LCD is used for display with segment external expansion with 1/4 duty. Figure 3 is a diagram showing connections for segment signals and common signals of the 16-line 8-character segment LCD used in this task example.

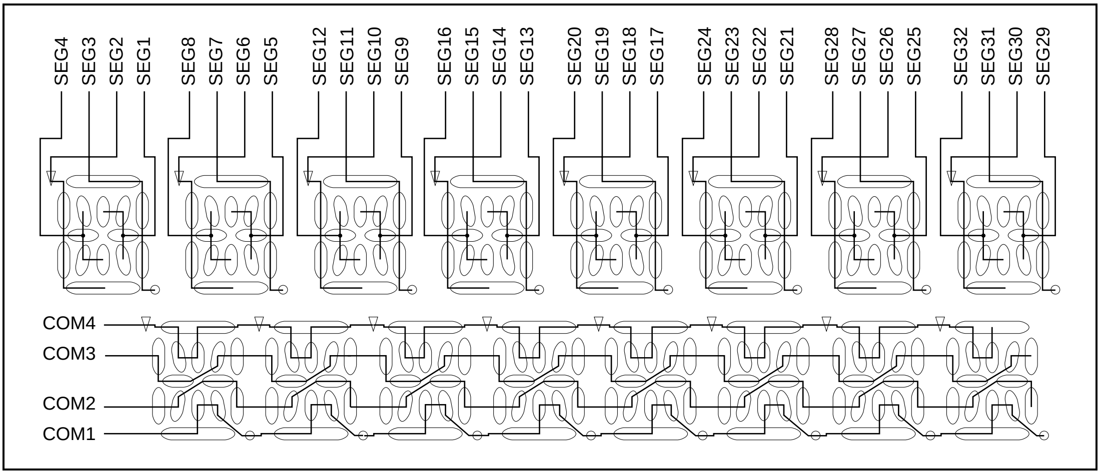


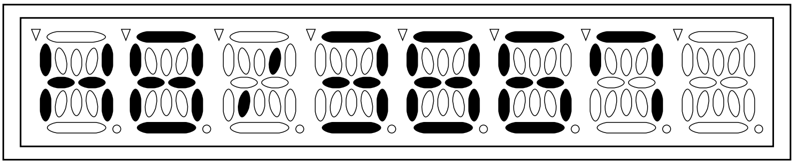
Figure 3 Connections of Segment Signals and Common Signals of the 16-Line 8-Character Segment LCD Used in this Task Example

5. Figure 4 shows the LCD RAM mapping under 1/4 duty driving when using segment external expansion.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
H'F740	SEG_{2}	SEG_{2}	SEG_{2}	SEG_{2}	SEG_{1}	SEG_{1}	SEG_{1}	SEG_{1}
H'F741	SEG_{4}	SEG_{4}	SEG_{4}	SEG_{4}	SEG_{3}	SEG_{3}	SEG_{3}	SEG_{3}
H'F742	SEG_{6}	SEG_{6}	SEG_{6}	SEG_{6}	SEG_{5}	SEG_{5}	SEG_{5}	SEG_{5}
H'F743	SEG_{8}	SEG_{8}	SEG_{8}	SEG_{8}	SEG_{7}	SEG_{7}	SEG_{7}	SEG_{7}
H'F744	SEG_{10}	SEG_{10}	SEG_{10}	SEG_{10}	SEG_{9}	SEG_{9}	SEG_{9}	SEG_{9}
H'F745	SEG_{12}	SEG_{12}	SEG_{12}	SEG_{12}	SEG_{11}	SEG_{11}	SEG_{11}	SEG_{11}
H'F746	SEG_{14}	SEG_{14}	SEG_{14}	SEG_{14}	SEG_{13}	SEG_{13}	SEG_{13}	SEG_{13}
H'F747	SEG_{16}	SEG_{16}	SEG_{16}	SEG_{16}	SEG_{15}	SEG_{15}	SEG_{15}	SEG_{15}
H'F748	SEG_{18}	SEG_{18}	SEG_{18}	SEG_{18}	SEG_{17}	SEG_{17}	SEG_{17}	SEG_{17}
H'F749	SEG_{20}	SEG_{20}	SEG_{20}	SEG_{20}	SEG_{19}	SEG_{19}	SEG_{19}	SEG_{19}
H'F74A	SEG_{22}	SEG_{22}	SEG_{22}	SEG_{22}	SEG_{21}	SEG_{21}	SEG_{21}	SEG_{21}
H'F74B	SEG_{24}	SEG_{24}	SEG_{24}	SEG_{24}	SEG_{23}	SEG_{23}	SEG_{23}	SEG_{23}
H'F74C	SEG_{26}	SEG_{26}	SEG_{26}	SEG_{26}	SEG_{25}	SEG_{25}	SEG_{25}	SEG_{25}
H'F74D	SEG_{28}	SEG_{28}	SEG_{28}	SEG_{28}	SEG_{27}	SEG_{27}	SEG_{27}	SEG_{27}
H'F74E	SEG_{30}	SEG_{30}	SEG_{30}	SEG_{30}	SEG_{29}	SEG_{29}	SEG_{29}	SEG_{29}
H'F74F	SEG_{32}	SEG_{32}	SEG_{32}	SEG_{32}	SEG_{31}	SEG_{31}	SEG_{31}	SEG_{31}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
H'F75F	SEG_{64}	SEG_{64}	SEG_{64}	SEG_{64}	SEG_{63}	SEG_{63}	SEG_{63}	SEG_{63}
	↓ COM_{4}	↓ COM_{3}	↓ COM_{2}	↓ COM_{1}	↓ COM_{4}	↓ COM_{3}	↓ COM_{2}	↓ COM_{1}

Figure 4 LCD RAM Mapping under 1/4 Duty Driving with Segment External Expansion

6. Figure 5 shows the relation between the 16-line 8-character segment LCD display and LCD RAM settings used in this task example. As shown in the figure, by setting the LCD RAM appropriately, “H8/3867” is displayed on the 16-line 8-character segment LCD.



	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'F740	0	1	1	0	0	0	1	0	Display data for “H”
H'F741	0	1	0	0	0	1	1	0	
H'F742	0	1	1	1	0	0	1	0	Display data for “8”
H'F743	0	1	0	0	1	1	1	0	
H'F744	0	0	0	0	0	1	0	0	Display data for “/”
H'F745	0	0	1	0	0	0	0	0	
H'F746	0	0	0	1	0	0	1	0	Display data for “3”
H'F747	0	1	0	0	1	1	1	0	
H'F748	0	1	1	1	0	0	1	0	Display data for “8”
H'F749	0	1	0	0	1	1	1	0	
H'F74A	0	1	1	1	0	0	1	0	Display data for “6”
H'F74B	0	1	0	0	1	0	1	0	
H'F74C	0	1	0	0	0	0	0	0	Display data for “7”
H'F74D	0	0	0	0	1	1	1	0	
H'F74E	0	0	0	0	0	0	0	0	Display data for “ ” *
H'F74F	0	0	0	0	0	0	0	0	

Note: *: “ ” denotes a blank (nothing displayed)

Figure 5 Relation between LCD Display and LCD RAM Settings

7. Figure 6 shows the relation between the display of the eighth column from the right in the 16-line 8-character segment LCD and the LCD RAM corresponding to SEG_1 through SEG_4. As indicated in figure 6, when the LCD RAM bits corresponding to 0 through f are set to 1, the LCD regions are lit; when they are cleared to 0, the LCD regions are unlit.

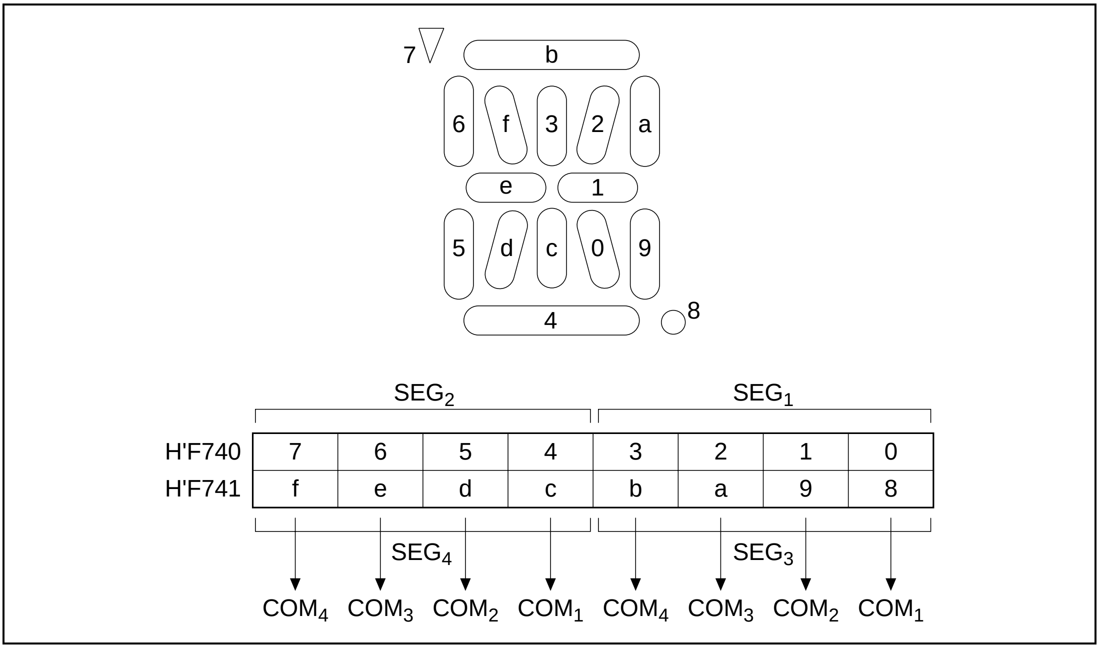


Figure 6 Relation between LCD Lit/Unlit States and Corresponding LCD RAM Settings

8. Table 2 shows examples of SEG_1 through SEG_4 display and display data for a 16-line 8-character segment LCD.

Table 2 Display Data Examples

Symbol	Display	Address	Display Data								Symbol	Display	Address	Display Data							
		H'F740	0	0	0	0	0	0	0	0	G		H'F740	0	1	1	1	0	0	1	0
		H'F741	0	0	0	0	0	0	0	0			H'F741	0	0	0	0	1	0	1	0
*		H'F740	0	0	0	0	1	1	1	1	H		H'F740	0	1	1	0	0	0	1	0
		H'F741	1	1	1	1	0	0	0	0			H'F741	0	1	0	0	0	1	1	0
–		H'F740	0	0	0	0	0	0	1	0	I		H'F740	0	0	0	1	1	0	0	0
		H'F741	0	1	0	0	0	0	0	0			H'F741	0	0	0	1	1	0	0	0
/		H'F740	0	0	0	0	0	1	0	0	J		H'F740	0	0	1	1	1	0	0	0
		H'F741	0	0	1	0	0	0	0	0			H'F741	0	0	0	1	1	0	0	0
0		H'F740	0	1	1	1	0	1	0	0	K		H'F740	0	1	1	0	0	1	0	1
		H'F741	0	0	1	0	1	1	1	0			H'F741	0	1	0	0	0	0	0	0
1		H'F740	0	0	0	0	0	0	0	0	L		H'F740	0	1	1	1	0	0	0	0
		H'F741	0	0	0	0	0	1	1	0			H'F741	0	0	0	0	0	0	0	0
2		H'F740	0	0	1	1	0	0	1	0	M		H'F740	0	1	1	0	0	1	0	0
		H'F741	0	1	0	0	1	1	0	0			H'F741	1	0	0	0	0	1	1	0
3		H'F740	0	0	0	1	0	0	1	0	N		H'F740	0	1	1	0	0	0	0	1
		H'F741	0	1	0	0	1	1	0	0			H'F741	1	0	0	0	0	1	1	0
4		H'F740	0	1	0	0	0	0	1	0	O		H'F740	0	1	1	1	0	0	0	0
		H'F741	0	1	0	0	0	1	1	0			H'F741	0	0	0	0	1	1	1	0
5		H'F740	0	1	0	1	0	0	1	0	P		H'F740	0	1	1	0	0	0	1	0
		H'F741	0	1	0	0	1	0	1	0			H'F741	0	1	0	0	1	1	9	0
6		H'F740	0	1	1	1	0	0	1	0	Q		H'F740	0	1	1	1	0	0	9	1
		H'F741	0	1	0	0	0	0	1	0			H'F741	0	0	0	0	1	1	1	0
7		H'F740	0	1	0	0	0	0	0	0	R		H'F740	0	1	1	0	0	0	1	1
		H'F741	0	0	0	0	1	1	1	0			H'F741	0	1	0	0	1	1	0	0
8		H'F740	0	1	1	1	0	0	1	0	S		H'F740	0	0	0	1	0	0	0	1
		H'F741	0	1	0	0	1	1	1	0			H'F741	1	0	0	0	1	0	0	0
9		H'F740	0	1	0	0	0	0	1	0	T		H'F740	0	0	0	0	1	0	0	0
		H'F741	0	1	0	0	1	1	1	0			H'F741	0	0	0	1	1	0	0	0
A		H'F740	0	1	1	0	0	0	1	0	U		H'F740	0	1	1	1	0	0	0	0
		H'F741	0	1	0	0	1	1	1	0			H'F741	0	0	0	0	0	1	1	0
B		H'F740	0	0	0	1	1	1	0	0	V		H'F740	0	1	1	0	0	1	0	0
		H'F741	0	0	0	1	1	1	1	0			H'F741	0	0	1	0	0	0	0	0
C		H'F740	0	1	1	1	0	0	0	0	W		H'F740	0	1	1	0	0	0	0	1
		H'F741	0	0	0	0	1	0	0	0			H'F741	0	0	1	0	0	1	1	0
D		H'F740	0	0	0	1	1	0	0	0	X		H'F740	0	0	0	0	0	1	0	1
		H'F741	0	0	0	1	1	1	1	0			H'F741	1	0	1	0	0	0	0	0
E		H'F740	0	1	1	1	0	0	1	0	Y		H'F740	0	0	0	0	0	1	0	0
		H'F741	0	1	0	0	1	0	0	0			H'F741	1	0	0	1	0	0	0	0
F		H'F740	0	1	1	0	0	0	1	0	Z		H'F740	0	0	0	1	0	1	0	0
		H'F741	0	1	0	0	1	0	0	0			H'F741	0	0	1	0	1	0	0	0

9. Table 3 indicates function allocations in this task example.

Table 3 Function Allocations

Function	Function Allocation
LPCR	Selects duty cycle, LCD driver, and pin functions.
LCR	Turns LCD drive power supply on and off, controls display data, and selects frame frequency.
LCR2	Switches between A and B waveforms, selects drive power supply, controls step-up constant-voltage (5 V) power supply, selects duty cycle for charge/discharge pulses to control disconnection of power supply split-resistance from power supply circuit.
COM_4 to COM_1	Used as common drivers.
V_0 to V_3	Used as the LCD drive power supply pins to be connected to the HD66100.
CL_1	Functions as a display data latch clock output to be connected to the HD66100.
CL_2	Functions as a display data latch clock output to be connected to the HD66100.
M	Functions as an LCD alternation signal output to be connected to the HD66100.
DO	Functions as a serial display data output to be connected to the HD66100.
LCD RAM	Sets the LCD display data.

Explanation of Functions Used

1. Hardware settings for LCD display are explained below.

   a. LCD drive power supply settings

      The H8/3867 Series can use either the internal power supply circuit or an external power supply circuit as the LCD drive power supply. In addition, either the power supply voltage (V_{CC}) or the step-up constant-voltage (5 V) can be selected for the internal power supply circuit.

      When using the internal power supply circuit to drive the LCD, the V_0 and V_1 pins are connected externally, as illustrated in figure 7.

      In this task example, the step-up constant-voltage power supply is used as the LCD drive power supply.

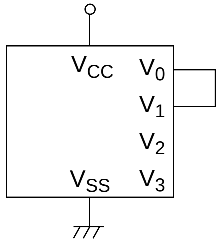


Figure 7 Example of Connection of LCD Power Supply Pins When Using Internal Power Supply Circuit

b. Contrast control function

A block diagram of the LCD drive power supply circuit appears in figure 8. Either V_{CC} or a 5 V output from the step-up constant-voltage power supply circuit is output to pin V_0. When these voltages are used directly to drive the LCD, the V_0 and V_1 pins should be shorted. By inserting a variable resistance R between the V_0 and V_1 pins, the voltage applied to the V_1 pin can be adjusted, and the LCD panel contrast can be controlled.

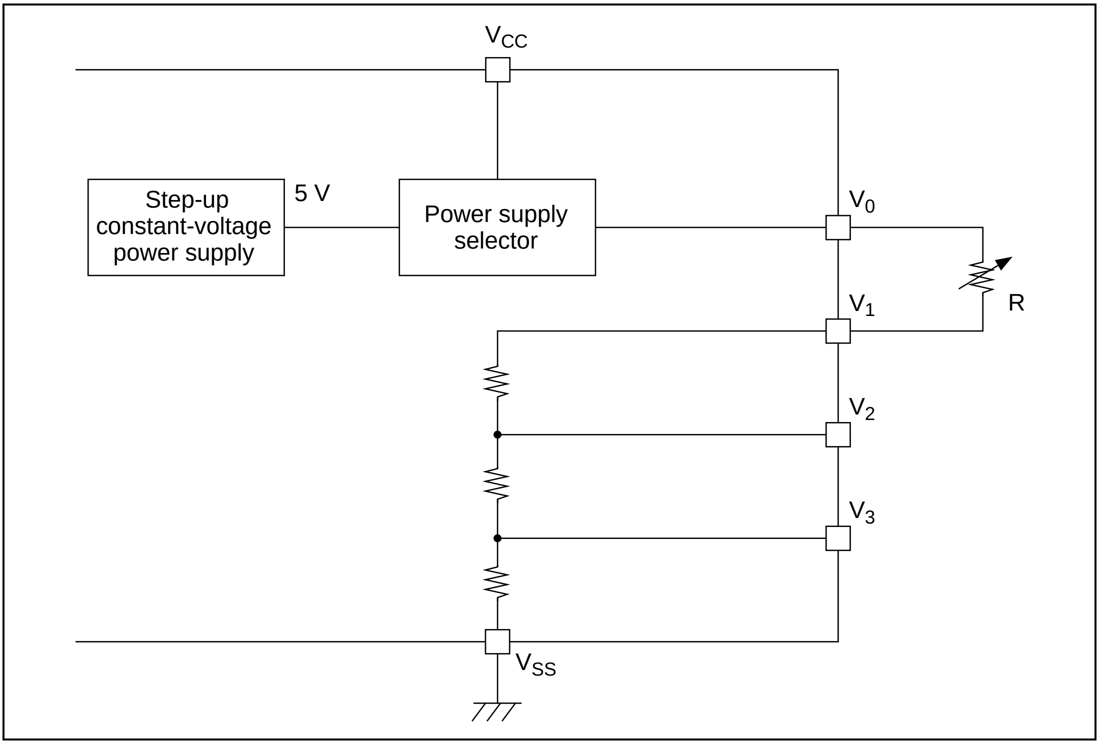


Figure 8 Block Diagram of LCD Drive Power Supply Circuit

2. Software settings for LCD display are explained below.

   a. Duty selection

      DTS1 and DTS0 are used to select from among static, 1/2 duty, 1/3 duty, and 1/4 duty.

   b. Segment driver selection

      SGS3 to SGS0 are used to select the segment drivers to be used.

   c. Frame frequency selection

      By setting CKS3 to CKS0, the frame frequency can be selected. The frame frequency should be selected according to the LCD panel.

   d. Selection of A and B waveforms

      LCDAB can be used to select either the A or the B waveform for use as the LCD waveform.

   e. Selection of LCD drive power supply

      When using the internal power supply circuit, SUPS can be used to select the power supply to be used. When using an external power supply circuit, SUPS is used to select V_{CC}, and PSW should be used to turn off the LCD drive power supply.

3. Figure 9 shows the operation principle of this task example.

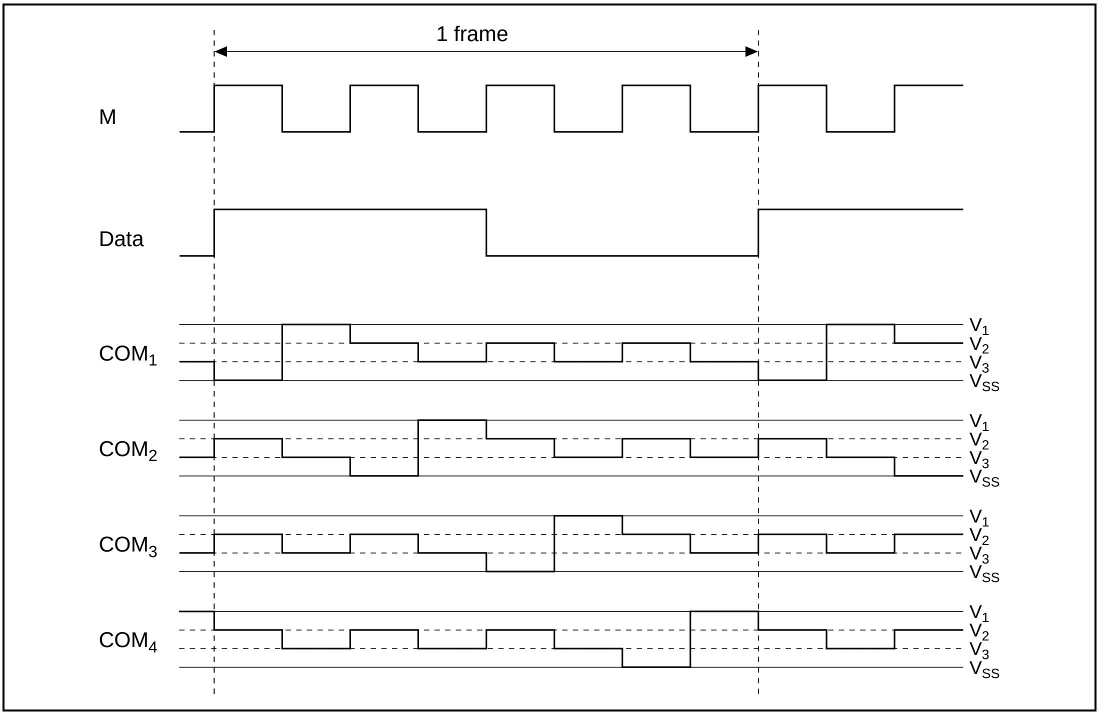


Figure 9 Operation Principle

Explanation of Software

1. Explanation of Modules

   Table 4 explains the modules in this task example.

Table 4 Module Explanation

Module Name	Label Name	Function
Main routine	MAIN	Initializes the stack pointer, LCD RAM, and LCD controller/driver, and enables interrupts.

2. Explanation of Arguments

   In this task example, no arguments are used.

3. Explanation of Internal Registers Used

   Table 5 gives explanation of the internal registers used in this task example.

Table 5 Explanation of Internal Registers Used

Register Name		Description	RAM Address	Settings
LPCR	DTS1, DTS0	LCD port control register (Duty cycle selection 1, 0) Select duty from among static, 1/2 duty, 1/3 duty, and 1/4 duty. • When DTS1 = 0 and DTS0 = 0, static duty is selected • When DTS1 = 0 and DTS0 = 1, 1/2 duty is selected • When DTS1 = 1 and DTS0 = 0, 1/3 duty is selected • When DTS1 = 1 and DTS0 = 1, 1/4 duty is selected	H'FFC0 Bit 7 and bit 6	DTS1 = 1 DTS0 = 1

Table 5 Explanation of Internal Registers Used (cont)

Register	Name	Description	RAM Address	Settings
LPCR	CMX	LCD port control register (Common function selection) Selects whether the same waveform is output from several pins in order to increase the common driving capacity, when common pins are not selected for a given duty cycle. • When CMX = 0, the same waveform is not output from multiple common pins not used at that duty cycle • When CMX = 1, the same waveform is output from multiple common pins not used at that duty cycle	H'FFC0 Bit 5	0
LPCR	SGX	LCD port control register (Expansion signal select) Selects whether the SEG_{32}/CL_1, SEG_{31}/CL_2, SEG_{30}/D0, and SEG_{29}/M pins are used as segment pins (SEG_{32} through SEG_{29}), or as segment external expansion signal pins (CL_1, CL_2, D0, M). • When SGX = 0, they are used as segment pins (SEG_{32} through SEG_{29}) • When SGX = 1, they are used as segment external expansion signal pins (CL_1, CL_2, D0, M)	H'FFC0 Bit 4	1

Table 5 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Settings
LPCR	SGS3 to SGS0	LCD port control register (Segment driver selection) Select the segment driver to be used. • When SGX = 0, SGS3 = 0, SGS2 = 0, SGS1 = 0 and SGS0 = 0, pins SEG_{32} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 0, SGS1 = 0 and SGS0 = 1, pins SEG_{32} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 0, SGS1 = 1 and SGS0 = *, pins SEG_{32} through SEG_{25} function as segment drivers and pins SEG_{24} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 1, SGS1 = 0 and SGS0 = *, pins SEG_{32} through SEG_{17} function as segment drivers and pins SEG_{16} through SEG_1 function as ports • When SGX = 0, SGS3 = 0, SGS2 = 1, SGS1 = 1 and SGS0 = *, pins SEG_{32} through SEG_9 function as segment drivers and pins SEG_8 through SEG_1 function as ports • When SGX = 0, SGS3 = 1, SGS2 = *, SGS1 = * and SGS0 = *, pins SEG_{32} through SEG_1 function as segment drivers • When SGX = 1, SGS3 = 0, SGS2 = 0, SGS1 = 0 and SGS0 = 0, pins SEG_{32} through SEG_{29} function as external expansion pins and pins SEG_{28} through SEG_1 function as ports • SGX = 1, SGS3 = *, SGS2 = *, SGS1 = * and SGS0 = * cannot be specified *: Don’t care	H'FFC0 Bit 3 to bit 0	SGS3 = 0 SGS2 = 0 SGS1 = 0 SGS0 = 0

Table 5 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Settings
LCR	PSW	LCD control register (LCD drive power supply on/off control) Turns the LCD drive power supply off when LCD display is not used in power-down mode, or when an external power supply is used. When the ACT bit is cleared to 0, or when in standby mode, the LCD drive power supply is turned off regardless of this bit setting. • When PSW = 0, the LCD drive power supply is turned off • When PSW = 1, the LCD drive power supply is turned on	H'FFC1 Bit 6	1
LCR	ACT	LCD control register (Display function activate) Selects whether the LCD controller/driver is to be used or not. By clearing this bit to 0, LCD controller/driver operation is halted. Also, regardless of the value of PSW, the LCD drive power supply is turned off. However, the register contents are maintained. • When ACT = 0, LCD controller/driver operation is halted • When ACT = 1, LCD controller/driver operations	H'FFC1 Bit 5	1
LCR	DISP	LCD control register (Display data control) DISP selects whether the LCD RAM contents or blank data are to be displayed. • When DISP = 0, blank data is displayed • When DISP = 1, LCD RAM data is displayed	H'FFC1 Bit 4	1

Table 5 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Settings
LCR	CKS3 to CKS0	LCD control register (Frame frequency select 3 to 0) Select the operating clock and the frame frequency. • When CKS3 = 0, CKS2 = *, CKS1 = 0 and CKS0 = 0, ϕ_w is selected as operating clock • When CKS3 = 0, CKS2 = *, CKS1 = 0 and CKS0 = 1, $\phi_w/2$ is selected as operating clock • When CKS3 = 0, CKS2 = *, CKS1 = 1 and CKS0 = *, $\phi_w/4$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 0 and CKS0 = 0, $\phi/2$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 0 and CKS0 = 1, $\phi/4$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 1 and CKS0 = 0, $\phi/8$ is selected as operating clock • When CKS3 = 1, CKS2 = 0, CKS1 = 1 and CKS0 = 1, $\phi/16$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 0 and CKS0 = 0, $\phi/32$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 0 and CKS0 = 1, $\phi/64$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 1 and CKS0 = 0, $\phi/128$ is selected as operating clock • When CKS3 = 1, CKS2 = 1, CKS1 = 1 and CKS0 = 1, $\phi/256$ is selected as operating clock *: Don’t care	H'FFC1 Bit 3 to bit 0	CKS3 = 1 CKS2 = 1 CKS1 = 1 CKS0 = 0

Table 5 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Settings
LCR2	LCDAB	LCD control register 2 (A waveform/B waveform switching control) Selects whether the A or B waveform is to be used for LCD driving. • When LCDAB = 0, the LCD is driven using the A waveform • When LCDAB = 1, the LCD is driven using the B waveform	H'FFC2 Bit 7	0
LCR2	SUPS	LCD control register 2 (Drive power supply select, step-up constant-voltage (5 V) power supply control) When V_{CC} is selected as the drive power supply, the step-up constant-voltage (5 V) power supply operation is halted; when 5 V is selected as the drive power supply, the step-up constant-voltage (5 V) power supply operates. • When SUPS = 0, the drive power supply is V_{CC}, and the step-up constant-voltage (5 V) power supply operation is halted • When SUPS = 1, the drive power supply is 5 V, and the step-up constant-voltage (5 V) power supply operates	H'FFC2 Bit 4	0

Table 5 Explanation of Internal Registers Used (cont)

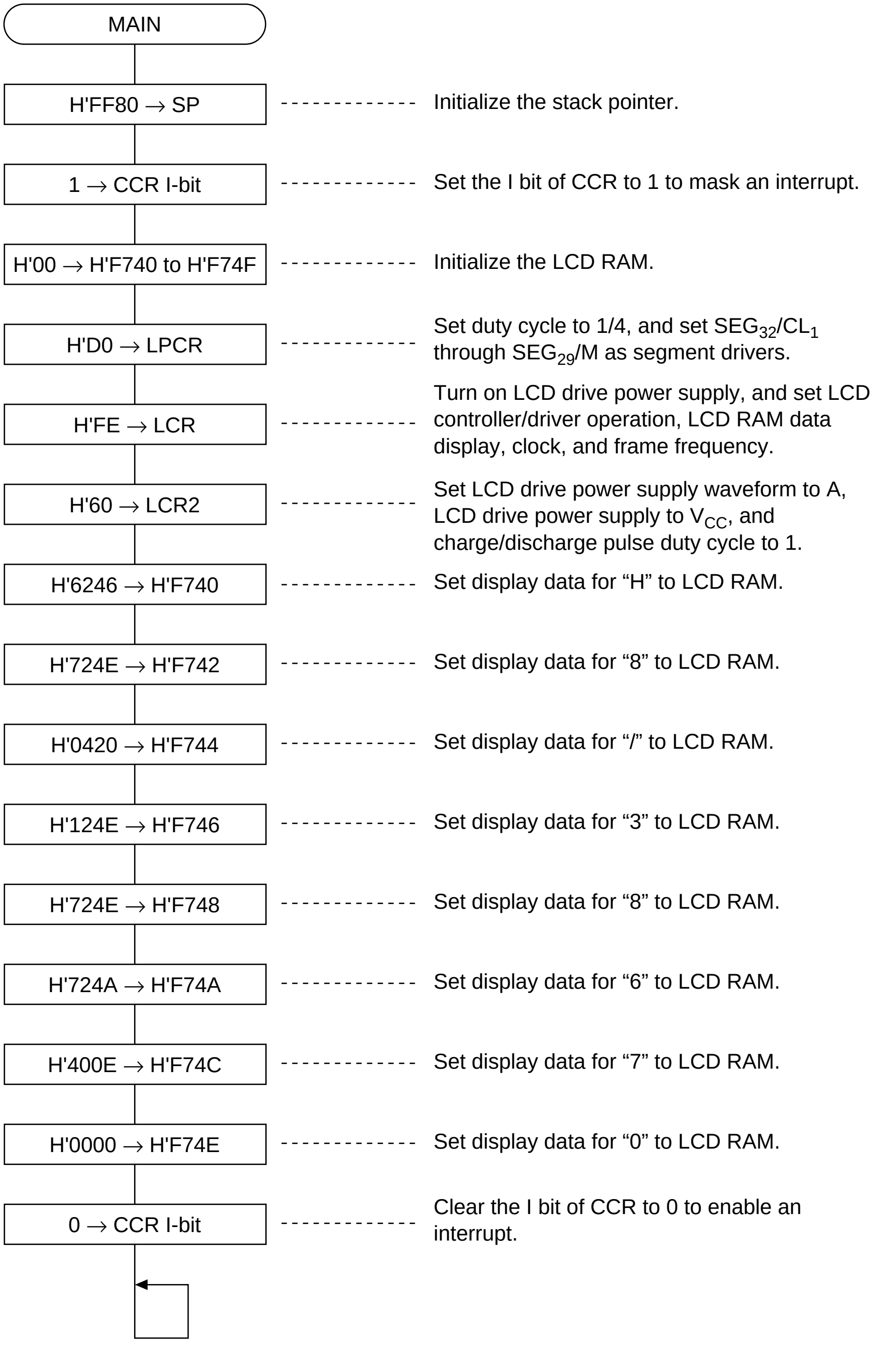
Register Name		Description	RAM Address	Settings
LCR2	CDS3 to CDS0	LCD control register 2 (Charge/discharge pulse duty cycle selection 3 to 0) Select the duty cycle while the power supply split-resistance is connected to the power supply circuit. • When CDS3 = 0, CDS2 = 0, CDS1 = 0 and CDS0 = 0, the duty cycle is 1 • When CDS3 = 0, CDS2 = 0, CDS1 = 0 and CDS0 = 1, the duty cycle is 1/8 • When CDS3 = 0, CDS2 = 0, CDS1 = 1 and CDS0 = 0, the duty cycle is 2/8 • When CDS3 = 0, CDS2 = 0, CDS1 = 1 and CDS0 = 1, the duty cycle is 3/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 0 and CDS0 = 0, the duty cycle is 4/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 0 and CDS0 = 1, the duty cycle is 5/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 1 and CDS0 = 0, the duty cycle is 6/8 • When CDS3 = 0, CDS2 = 1, CDS1 = 1 and CDS0 = 1, the duty cycle is 0 • When CDS3 = 1, CDS2 = 0, CDS1 = * and CDS0 = *, the duty cycle is 1/16 • When CDS3 = 1, CDS2 = 1, CDS1 = * and CDS0 = *, the duty cycle is 1/32 *: Don’t care	H'FFC2 Bit 3 to bit 0	CDS3 = 0 CDS2 = 0 CDS1 = 0 CDS0 = 0

4. Explanation of RAM Usage

In this task example, RAM is not used.

Flowchart

1. Main routine



```
;***********************************************************
;*   H8/3867 Application Note
;*
;*     'Liquid Crystal Display
;*     -Using Segment External Expansion,
;*      Using HD66100, 1/4 Duty Drive'
;*
;*            Function : LCD Controller / Driver
;*
;*            External Clock : 6MHz
;*            Internal Clock : 3MHz
;*            Sub Clock      : 32.768kHz
;***********************************************************
;
        .cpu      300l
;
;***********************************************************
;*      Symbol Defnition
;***********************************************************
;
LPCR    .equ      h'ffc0      ;LCD Port Control Register
LCR     .equ      h'ffc1      ;LCD Control Register
LCR2    .equ      h'ffc2      ;LCD Control Register 2
;
;***********************************************************
;*      Vector Address
;***********************************************************
;
        .org      h'0000
        .data.w   MAIN        ;No.0 Reset Interrupt(H'0000-H'0001)
;
        .org      h'0008
        .data.w   MAIN        ;No.4 _IRQ0 Interrupt(H'0008-H'0009)
        .data.w   MAIN        ;No.5 _IRQ1 Interrupt(H'000A-H'000B)
        .data.w   MAIN        ;No.6 _IRQ2 Interrupt(H'000C-H'000D)
        .data.w   MAIN        ;No.7 _IRQ3 Interrupt(H'000E-H'000F)
```

```
        .data.w   MAIN        ;No.8 _IRQ4 Interrupt(H'0010-H'0011)
        .data.w   MAIN        ;No.9 _WKP0-_WKP7 Interrupt(H'0012-H'0013)
;
        .org      h'0016
        .data.w   MAIN        ;No.11 Timer A Interrupt(H'0016-H'0017)
        .data.w   MAIN        ;No.12 AEC Interrupt(H'0018-H'0019)
        .data.w   MAIN        ;No.13 Timer C Interrupt(H'001A-H'001B)
        .data.w   MAIN        ;No.14 Timer FL Interrupt(H'001C-H'001D)
        .data.w   MAIN        ;No.15 Timer FH Interrupt(H'001E-H'001F)
        .data.w   MAIN        ;No.16 Timer G Interrupt(H'0020-H'0021)
        .data.w   MAIN        ;No.17 SCI31 Interrupt(H'0022-H'0023)
        .data.w   MAIN        ;No.18 SCI32 Interrupt(H'0024-H'0025)
        .data.w   MAIN        ;No.19 A/D Converter Interrupt(H'0026-H'0028)
        .data.w   MAIN        ;No.20 Direct Transfer Interrupt(H'0028-H'0029)
;
;***********************************************************
;*      MAIN : Main Routine
;***********************************************************
        .org      h'1000
;
MAIN:   .equ      $
        mov.w     #h'ff80,sp  ;Initialize Stack Pointer
        orc       #h'80,ccr   ;Interrupt Disable
;
        sub.b     r0l,r0l     ;Initialize LCD RAM
        mov.w     #h'f740,r1
        mov.w     #h'f750,r2
INIT:   mov.b     r0l,@r1
        adds      #1,r1
        cmp.w     r2,r1
        bne       INIT
;
        mov.b     #h'd0,r0l   ;Initialize LCD Port Control
        mov.b     r0l,@LPCR
        mov.b     #h'fe,r0l   ;Initialize LCD Control
        mov.b     r0l,@LCR
        mov.b     #h'60,r0l   ;Initialize LCD Control 2
```

```
        mov.b     r0l,@LCR2
;
        mov.w     #h'f740,r1   ;Set LCD RAM Start Address
        mov.w     #h'f750,r2   ;Set LCD RAM End Address
        mov.w     #h'1500,r3   ;Set LCD Data Address
DISP:   mov.w     @r3,r0       ;Load LCD Data
        mov.w     r0,@r1       ;Store LCD Data to LCD RAM
        adds      #2,r3        ;Increment LCD Data Address
        adds      #2,r1        ;Increment LCD RAM Address
        cmp.w     r2,r1        ;LCD RAM Address = LCD RAM End Address ?
        bne       DISP         ;No.
;
EXIT:   bra       EXIT         ;Yes.
;
;***********************************************************
;*      LCD Data Table
;***********************************************************
        .org      h'1500
;
        .data.w   h'6246       ;"H"
        .data.w   h'724e       ;"8"
        .data.w   h'0420       ;"/"
        .data.w   h'124e       ;"3"
        .data.w   h'724e       ;"8"
        .data.w   h'724a       ;"6"
        .data.w   h'400e       ;"7"
        .data.w   h'0000       ;" "
;
        .end
```

2.6 Oscillation Stabilization Time Settings

Oscillation Stabilization Time Settings	MCU: H8/3867 Series	Functions Used: Power-Down Mode

Settings

The time for which the CPU and peripheral functions must wait until the clock stabilizes when, by means of specific interrupts, standby or watch mode is canceled and there is a transition to active mode, is set. This standby time must be set to be longer than the time for oscillation stabilization, in accordance with the operating frequency.

Setting the standby time

The standby time is set by setting the standby timer selects 2 to 0 bits (STS2 to STS0) of the system control register 1 (SYSCR1).

Explanation of the STS2 to STS0 bits

Table 1 explains the STS2 to STS0 bits of the SYSCR1 register.

Table 1 STS2 to STS0 Settings and Standby Time

SYSCR1			
Bit 6	**Bit 5**	**Bit 4**	
STS2	**STS1**	**STS0**	**Description**
0	0	0	Standby time = 8,192 states (Initial value)
		1	Standby time = 16,384 states
	1	0	Standby time = 32,768 states
		1	Standby time = 65,536 states
1	0	0	Standby time = 131,072 states
		1	Standby time = 2 states (External clock mode)
	1	0	Standby time = 8 states
		1	Standby time = 16 states

Note: When an external clock signal is to be input, the standby timer select pins should be set to external clock mode prior to execution of the mode transition. When an external clock is not used, the external clock input mode should not be set.

Operating frequency and oscillation stabilization time when a crystal oscillator is used

Table 2 shows the standby times resulting for different operating frequencies and STS2 to STS0 settings when a crystal oscillator is used. STS2 to STS0 are set so that the standby time is longer than the time required for oscillation stabilization.

Table 2 Operating Frequency and Oscillation Stabilization Times for Crystal Oscillators

Standby Timer Select Bit Settings				Operating Frequency		
STS2	**STS1**	**STS0**	**Standby Time**	**2 MHz**	**1 MHz**	**0.5 MHz**
0	0	0	8,192 states	4.1	8.2	16.4
		1	16,384 states	8.2	16.4	32.8
	1	0	32,768 states	16.4	32.8	65.5
		1	65,536 states	32.8	65.5	131.1
1	0	0	131,072 states	65.5	131.1	262.1
		1	2 states (Use prohibited)	0.001	0.002	0.004
	1	0	8 states	0.004	0.008	0.016
		1	16 states	0.008	0.016	0.032

Unit: ms

When an external clock is used

It is recommended that the circuit be used with STS2 = 1, STS1 = 0, and STS0 = 1. Use at other settings is also possible, but operation may begin before the standby time has completed.

Oscillation stabilization times

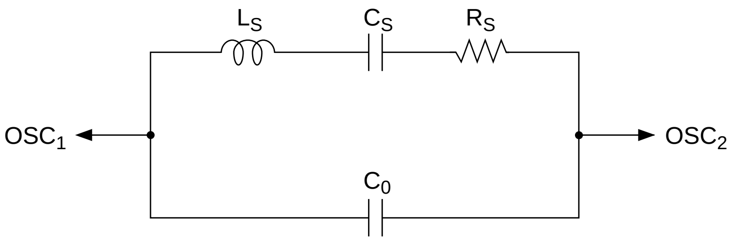
Table 3 shows the AC characteristics of oscillation stabilization times.

Table 3 AC Characteristics of Oscillation Stabilization Time

Item	Symbol	Applicable Pins	Test Conditions	Values			Unit	Reference Figure
				Min	Typ	Max		
Oscillation stabilization time	t_{rc}	OSC_1, OSC_2	V_{CC} = 2.2 V to 5.5 V (as shown in figure 1)	—	20	45	us	Figure 1*
			V_{CC} = 2.2 V to 5.5 V (as shown in figure 1)		0.1	8	ms	Figure 1
			Other than the above	—	—	50	ms	Figure 1
Oscillation stabilization time	t_{rc}	X_1, X_2	—	—	—	2.0	s	—

(V_{CC} = 1.8 to 5.5 V, AV_{CC} = 1.8 to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20 to +75°C, including subactive mode)

Note: *: Internal power supply step-down circuit not used.



Crystal oscillator parameters

Frequency (MHz)	1	4.193
R_S (max)	40 Ω	100 Ω
C_0 (max)	3.5 pF	16 pF

Ceramic oscillator parameters

Frequency (MHz)	0.4	4
R_S (max)	8.6 Ω	8.8 Ω
C_0 (max)	326 pF	36 pF

Figure 1 Oscillator Equivalent Circuit

Example of oscillation stabilization time settings

1. Functions

   A transition from active (high-speed) to watch mode is induced, and after 250 ms watch mode is canceled by a timer A interrupt, with a transition back to active (high-speed) mode. In making the transition from watch mode to active (high-speed) mode, the time for the CPU and peripheral functions to wait until the clock stabilizes is set to eight states.

2. Notes

   In these settings, when the watch mode is canceled by a timer A interrupt, part of the timer A interrupt processing includes prohibition of timer A interrupt requests. Hence when there is a transition from active (high-speed) mode to watch mode, and watch mode is then canceled by a timer A interrupt with a transition to active (high-speed) mode, processing is completed.

3. Watch mode

   a. Transition to watch mode

      When, in active mode or subactive mode, the software standby bit (SSBY) of the system control register 1 (SYSCR1) is 1 and the internal clock selector 3 bit (TMA3) of the timer mode register A (TMA) is 1, if a sleep instruction is executed, there is a transition to watch mode. In watch mode, operation of all built-in peripheral functions other than timer A, timer F, timer G, the asynchronous event counter, and the LCD (operation/halted selectable), is halted. So long as the standard voltage is supplied, the CPU, the internal registers for part of the built-in peripheral functions, and internal RAM are maintained, and the I/O ports are held at their states prior to transition.

   b. Watch mode cancellation

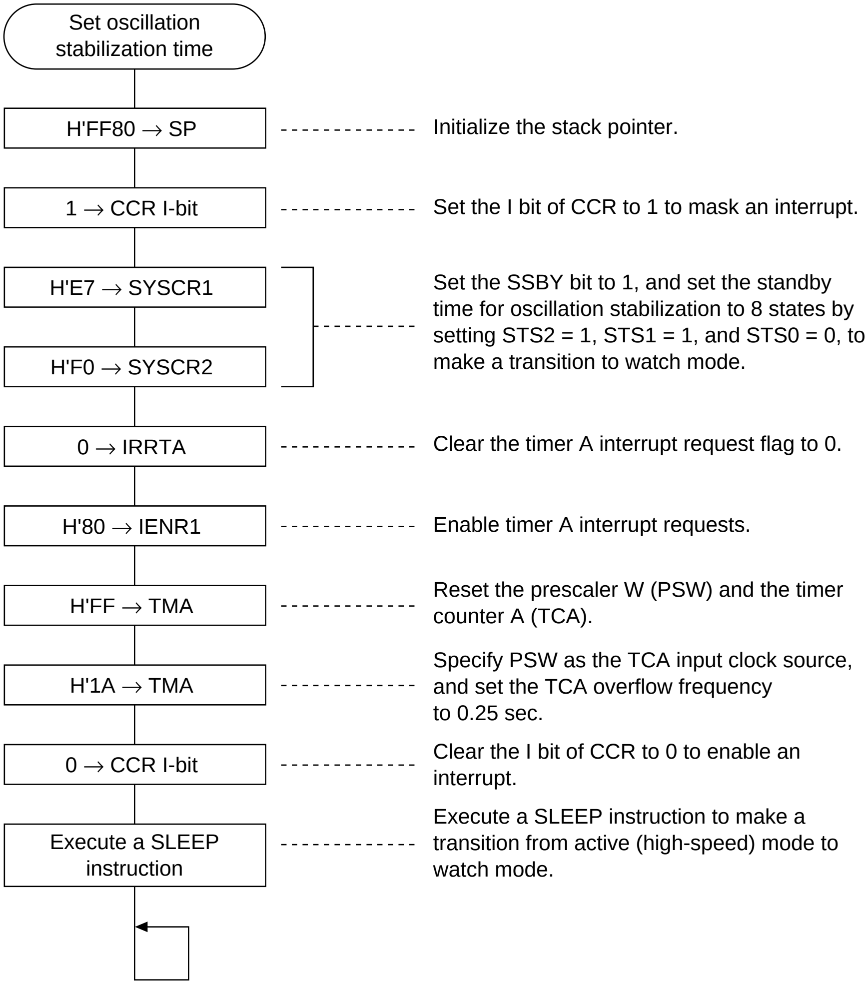
      Watch mode is canceled by an interrupt (IRQ_0, WKP_7 to WKP_0, timer A, timer F, timer G) or by RES pin input.

      In cancellation by an interrupt, when the interrupt occurs the watch mode is canceled, and if the low-speed on-flag (LSON) of SYSCR1 is 0 and the medium-speed on-flag (MSON) of the system control register 2 (SYSCR2) is 0, there is a transition to active (high-speed) mode. If LSON = 0 and MSON = 1, there is a transition to active (medium-speed) mode, and if LSON = 1 there is a transition to subactive mode. On transitions to active modes, after the time set by the STS2 to STS0 bits of SYSCR1 has elapsed, the stabilized clock pulse is supplied to the entire LSI, and interrupt exception processing begins. When the I bit of CCR is 1 or when acceptance of the relevant interrupt by the interrupt enable register is disabled, watch mode is not canceled.

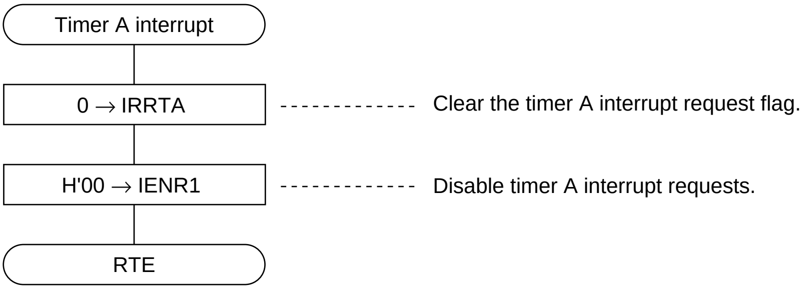
      In cancellation by input to the RES pin, if the RES pin is forced low, system clock oscillation is started. After the time for oscillation stabilization has elapsed, if the RES pin is forced high, the CPU initiates reset exception processing. The system clock is supplied to the entire LSI at the time, system clock oscillation is started. The RES pin should always be held low until the system clock oscillation stabilizes.

Flowchart

1. Main routine



2. Timer A interrupt processing routine



Program Lists

```
;**********************************************************
;*   H8/3867 Application Note
;*
;*     'Oscillator Settling Time -8 States'
;*
;*            Function : Oscillator Settling Time
;*
;*            External Clock : 6MHz
;*            Internal Clock : 3MHz
;*            Sub Clock      : 32.768kHz
;**********************************************************
;
        .cpu      300l
;
;**********************************************************
;*      Symbol Defnition
;**********************************************************
;
TMA      .equ      h'ffb0      ;Timer Mode Register A
SYSCR1   .equ      h'fff0      ;System Control Register 1
SYSCR2   .equ      h'fff1      ;System Control Register 2
IENR1    .equ      h'fff3      ;Interrupt Enable Register 1
IRR1     .equ      h'fff6      ;Interrupt Request Register 1
;
;**********************************************************
;*      Vector Address
;**********************************************************
;
        .org      h'0000
        .data.w   MAIN        ;No.0 Reset Interrupt(H'0000-H'0001)
;
        .org      h'0008
        .data.w   MAIN        ;No.4 _IRQ0 Interrupt(H'0008-H'0009)
        .data.w   MAIN        ;No.5 _IRQ1 Interrupt(H'000A-H'000B)
        .data.w   MAIN        ;No.6 _IRQ2 Interrupt(H'000C-H'000D)
        .data.w   MAIN        ;No.7 _IRQ3 Interrupt(H'000E-H'000F)
```

```
        .data.w   MAIN         ;No.8 _IRQ4 Interrupt(H'0010-H'0011)
        .data.w   MAIN         ;No.9 _WKP0-_WKP7 Interrupt(H'0012-H'0013)
;
        .org      h'0016
        .data.w   TAINT        ;No.11 Timer A Interrupt(H'0016-H'0017)
        .data.w   MAIN         ;No.12 AEC Interrupt(H'0018-H'0019)
        .data.w   MAIN         ;No.13 Timer C Interrupt(H'001A-H'001B)
        .data.w   MAIN         ;No.14 Timer FL Interrupt(H'001C-H'001D)
        .data.w   MAIN         ;No.15 Timer FH Interrupt(H'001E-H'001F)
        .data.w   MAIN         ;No.16 Timer G Interrupt(H'0020-H'0021)
        .data.w   MAIN         ;No.17 SCI31 Interrupt(H'0022-H'0023)
        .data.w   MAIN         ;No.18 SCI32 Interrupt(H'0024-H'0025)
        .data.w   MAIN         ;No.19 A/D Converter Interrupt(H'0026-H'0028)
        .data.w   MAIN         ;No.20 Direct Transfer Interrupt(H'0028-H'0029)
;
;**********************************************************
;*       MAIN : Main Routine
;**********************************************************
;
        .org      h'1000
;
MAIN:   .equ      $
        mov.w     #h'ff80,sp   ;Initialize Stack Pointer
        orc       #h'80,ccr    ;Interrupt Disable
;
        mov.b     #h'e7,r0l    ;Initialize System Control Regsiter
        mov.b     r0l,@SYSCR1
        mov.b     #h'f0,r0l
        mov.b     r0l,@SYSCR2
;
        bclr      #7,@IRR1
        mov.b     #h'80,r0l
        mov.b     r0l,@IENR1

;
        mov.b     #h'ff,r0l
        mov.b     r0l,@TMA
```

```
        mov.b   #h'1a,r0l
        mov.b   r0l,@TMA
;
        andc    #h'7f,ccr
;
        sleep
;
        nop
;
EXIT:   bra     EXIT
;
;***********************************************************
;*      TMAINT : Timer A Interrupt Routine
;***********************************************************
;
TAINT:  .equ    $
        bclr    #7,@IRR1
;
        mov.b   #h'00,r0l
        mov.b   r0l,@IENR1
;
        rte

;
        .end
```

2.7 Module Standby Mode Settings

Module Standby Mode Settings	**MCU:** **H8/3867 Series**	**Functions Used:** **Module Standby Mode**

Settings

Module standby halts the supply of the system clock to the module and stops module functions. Module standby can be set for individual peripheral functions. All built-in peripheral modules can be set to module standby mode.

Setting module standby mode

Module standby mode can be set for a particular module by clearing the corresponding bits of the clock stop register 1 (CKSTPR1) and clock stop register 2 (CKSTPR2) to 0.

Canceling module standby mode

Module standby mode can be cancelled for a particular module by setting the corresponding bits of the clock stop register 1 (CKSTPR1) and clock stop register 2 (CKSTPR2) to 1.

After reset, CKSTPR1 and CKSTPR2 are both initialized to H'FF.

Explanation of CKSTPR1 and CKSTPR2 registers

Table 1 gives explanations of the CKSTPR1 and CKSTPR2 registers.

Table 1 Explanations of CKSTPR1 and CKSTPR2 Registers

Register Name	Bit Number	Bit Name	Set Value	Description
CKSTPR1	Bit 6	S31CKSTP	0	Sets SCI3-1 to module standby mode
			1	Cancels SCI3-1 module standby mode
	Bit 5	S32CKSTP	0	Sets SCI3-2 to module standby mode
			1	Cancels SCI3-2 module standby mode
	Bit 4	ADCKSTP	0	Sets A/D converter to module standby mode
			1	Cancels A/D converter module standby mode
	Bit 3	TGCKSTP	0	Sets timer G to module standby mode
			1	Cancels timer G module standby mode
	Bit 2	TFCKSTP	0	Sets timer F to module standby mode
			1	Cancels timer F module standby mode
	Bit 1	TCCKSTP	0	Sets timer C to module standby mode
			1	Cancels timer C module standby mode
	Bit 0	TACCKSTP	0	Sets timer A to module standby mode
			1	Cancels timer A module standby mode
CKSTPR2	Bit 3	AECKSTP	0	Sets AEC to module standby mode
			1	Cancels AEC module standby mode
	Bit 2	WDCKSTP	0	Sets WDT to module standby mode
			1	Cancels WDT module standby mode
	Bit 1	PWCKSTP	0	Sets PWM to module standby mode
			1	Cancels PWM module standby mode
	Bit 0	LDCKSTP	0	Sets LCD to module standby mode
			1	Cancels LCD module standby mode

Example of module standby mode settings

1. Function

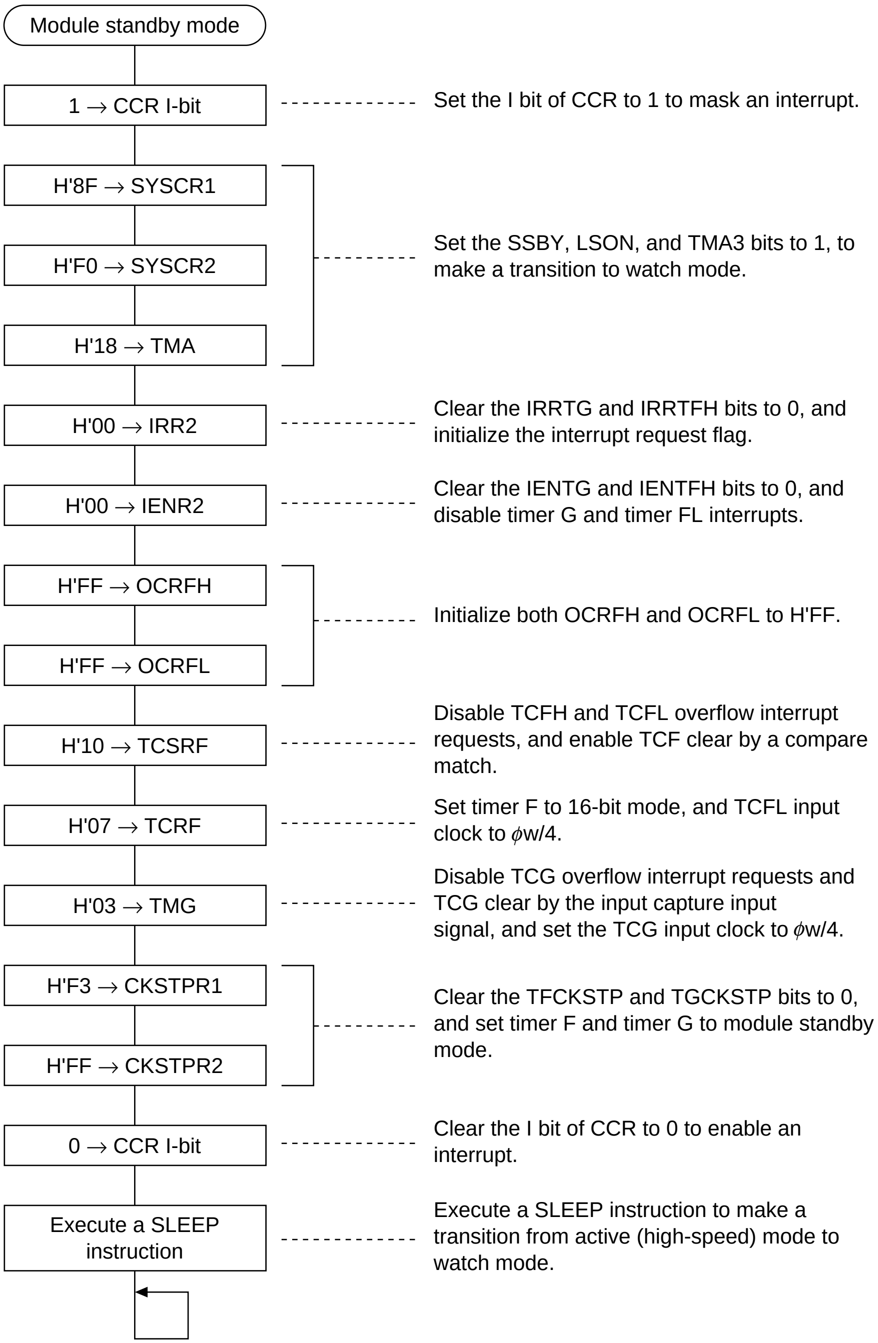
   In this example, while in active (high-speed) mode, timer F and timer G are set to module standby mode, and then make a transition to watch mode.

2. Notes

   a. Operation continues in watch mode only when an external clock is input as the timer F and timer G input clock, or when $\phi_w/4$ is selected as the internal clock. If any other clock is selected, operation is stopped while in watch mode. Hence under these settings, after setting the timer F, timer G input clock to $\phi_w/4$, timer F and timer G should be set to module standby mode before the transition to watch mode.

   b. For the settings of this example, all interrupt requests are disabled, so that if there is a transition to watch mode, watch mode can only be canceled by input from the RES pin.

Flowchart


Module standby mode
1 → CCR I-bit
Set the I bit of CCR to 1 to mask an interrupt.
H'8F → SYSCR1
H'F0 → SYSCR2
H'18 → TMA
Set the SSBY, LSON, and TMA3 bits to 1, to make a transition to watch mode.
H'00 → IRR2
Clear the IRRTG and IRRTFH bits to 0, and initialize the interrupt request flag.
H'00 → IENR2
Clear the IENTG and IENTFH bits to 0, and disable timer G and timer FL interrupts.
H'FF → OCRFH
H'FF → OCRFL
Initialize both OCRFH and OCRFL to H'FF.
H'10 → TCSRF
Disable TCFH and TCFL overflow interrupt requests, and enable TCF clear by a compare match.
H'07 → TCRF
Set timer F to 16-bit mode, and TCFL input clock to φw/4.
H'03 → TMG
Disable TCG overflow interrupt requests and TCG clear by the input capture input signal, and set the TCG input clock to φw/4.
H'F3 → CKSTPR1
H'FF → CKSTPR2
Clear the TFCKSTP and TGCKSTP bits to 0, and set timer F and timer G to module standby mode.
0 → CCR I-bit
Clear the I bit of CCR to 0 to enable an interrupt.
Execute a SLEEP instruction
Execute a SLEEP instruction to make a transition from active (high-speed) mode to watch mode.

```
;************************************************************
;*   H8/3867 Application Note
;*
;*     'Module Standby Mode
;*      -In Watch Mode, Timer F&G Module Stsndby Mode Set
;*
;*           Function : Module Standby Mode
;*
;*           External Clock : 6MHz
;*           Internal Clock : 3MHz
;*           Sub Clock      : 32.768kHz
;************************************************************
;
        .cpu      300l
;
;************************************************************
;*      Symbol Defnition
;************************************************************
;
TMA      .equ      h'ffb0      ;Timer Mode Register A
TCRF     .equ      h'ffb6      ;Timer Control Register F
TCSRF    .equ      h'ffb7      ;Timer Control/Status Register F
OCRFH    .equ      h'ffba      ;Output Compare Register FH
OCRFL    .equ      h'ffbb      ;Output Compare Register FL
TMG      .equ      h'ffbc      ;Timer Mode Register G
SYSCR1   .equ      h'fff0      ;System Control Register 1
SYSCR2   .equ      h'fff1      ;System Control Register 2
IENR2    .equ      h'fff4      ;Interrupt Enable Register 2
IRR2     .equ      h'fff7      ;Interrupt Request Register 2
CKSTPR1  .equ      h'fffa      ;Clock Stop Register 1
CKSTPR2  .equ      h'fffb      ;Clock Stop Register 2
;
;************************************************************
;*      Vector Address
;************************************************************
;
```

```
        .org      h'0000
        .data.w   MAIN         ;No.0 Reset Interrupt(H'0000-H'0001)
;
        .org      h'0008
        .data.w   MAIN         ;No.4 _IRQ0 Interrupt(H'0008-H'0009)
        .data.w   MAIN         ;No.5 _IRQ1 Interrupt(H'000A-H'000B)
        .data.w   MAIN         ;No.6 _IRQ2 Interrupt(H'000C-H'000D)
        .data.w   MAIN         ;No.7 _IRQ3 Interrupt(H'000E-H'000F)
        .data.w   MAIN         ;No.8 _IRQ4 Interrupt(H'0010-H'0011)
        .data.w   MAIN         ;No.9 _WKP0-_WKP7 Interrupt(H'0012-H'0013)
;
        .org      h'0016
        .data.w   MAIN         ;No.11 Timer A Interrupt(H'0016-H'0017)
        .data.w   MAIN         ;No.12 AEC Interrupt(H'0018-H'0019)
        .data.w   MAIN         ;No.13 Timer C Interrupt(H'001A-H'001B)
        .data.w   MAIN         ;No.14 Timer FL Interrupt(H'001C-H'001D)
        .data.w   MAIN         ;No.15 Timer FH Interrupt(H'001E-H'001F)
        .data.w   MAIN         ;No.16 Timer G Interrupt(H'0020-H'0021)
        .data.w   MAIN         ;No.17 SCI31 Interrupt(H'0022-H'0023)
        .data.w   MAIN         ;No.18 SCI32 Interrupt(H'0024-H'0025)
        .data.w   MAIN         ;No.19 A/D Converter Interrupt(H'0026-H'0028)
        .data.w   MAIN         ;No.20 Direct Transfer Interrupt(H'0028-H'0029)
;
;************************************************************
;*      MAIN : Main Routine
;************************************************************
;
        .org      h'1000
;
MAIN:   .equ      $
        mov.w     #h'ff80,sp   ;Initialize Stack Pointer
        orc       #h'80,ccr    ;Interrupt Disable
;
        mov.w     #h'8ff0,r0   ;Initialize System Control Regsiter
        mov.b     r0h,@SYSCR1
        mov.b     r0l,@SYSCR2
        mov.b     #h'18,r0l    ;Initialize Timer Mode Register
```

```
        mov.b     r0l,@TMA
;
        sub.b     r0l,r0l     ;Initialize Timer F
        mov.b     r0l,@IRR2
        mov.b     r0l,@IENR2
        mov.b     #h'ff,r0l
        mov.b     r0l,@OCRFH
        mov.b     r0l,@OCRFL
        mov.b     #h'10,r0l
        mov.b     r0l,@TCSRF
        mov.b     #h'07,r0l
        mov.b     r0l,@TCRF
;
        mov.b     #h'03,r0l   ;Initialize Timer G
        mov.b     r0l,@TMG
;
        mov.w     #h'f3ff,r0  ;Timer F & G Module Standby Mode ON
        mov.b     r0h,@CKSTPR1
        mov.b     r0l,@CKSTPR2
;
        andc      #h'7f,ccr   ;Interrupt Enable
;
        sleep                 ;Transfer to Watch Mode
        nop
;
EXIT:   bra       EXIT
;
        .end
```

2.8 Clock Operation Using Timer F

Clock Operation Using Timer F	MCU: H8/3867 Series	Functions Used: Timer F

Specifications

1. A 38.4-kHz subclock is used for clock operation employing timer F.

2. Timer F interrupts are issued every 1 sec, and a counter provided for clock use in RAM is incremented.

3. The clock counter provided in RAM has eight bits for counting seconds and eight bits for counting minutes; it begins counting from 00 min, 00 sec, and after counting up to 59 min, 59 sec, in the next cycle it is initialized to 00 min, 00 sec and continues counting.

4. After completion of initialization, there is a transition from active (high-speed) mode to watch mode, a timer F interrupt request causes a transition to subactive mode, the counter provided in RAM is incremented, and there is another transition to watch mode.

5. The mode transition diagram for this task example is shown in figure 1.

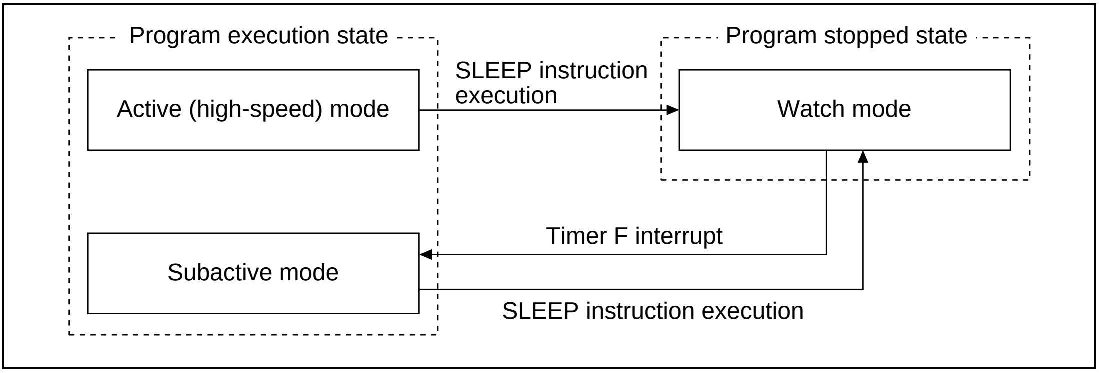


Figure 1 Diagram of Mode Transitions for this Task Example

Explanation of Functions Used

1. In this task example, clock operation is performed in which timer F is used to increment a counter provided in RAM every second. The features of timer F are as follows.

   - Four different internal clocks (ϕ/32, ϕ/16, ϕ/4, ϕ_w/4) or an external clock can be selected (external event counting is possible).
   - A single compare-match signal can be used for a toggle output to the TMOFH pin (the initial value of the toggle output can be set).
   - The counter can be reset by a compare-match signal.
   - There are a total of two interrupt factors: one compare-match, and one overflow.
   - Operation as two independent 8-bit timers (timer FH and timer FL) is also possible (in 8-bit mode).
   - When ϕ_w/4 is selected as the internal clock, operation in watch mode, subactive mode, and sleep mode is possible.
   - Using the module standby mode, it is possible to set standby mode in module units when not in use.

2. Figure 2 shows a block diagram of the timer F 16-bit compare match function used in this task example.

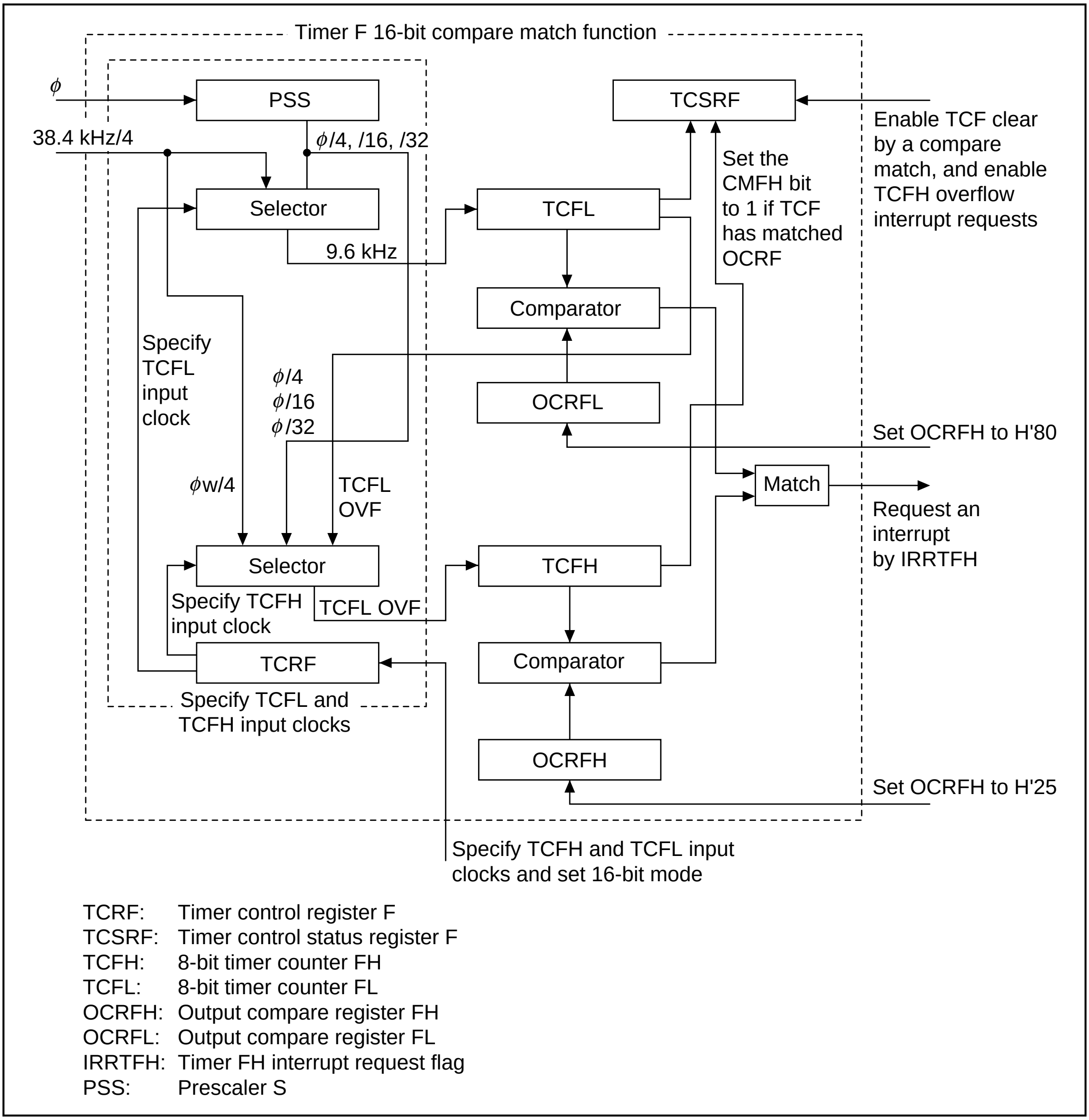


Figure 2 Block Diagram of Timer F 16-Bit Compare Match Function

3. Table 1 explains timer F functions.

Table 1 Timer F Functions

Timer control register F (TCRF)	
Functions	TCRF is an 8-bit write-only register. It switches between 16-bit mode and 8-bit mode, selects between four types of internal clocks and an external event, and sets the output levels of the TMOFH and TMOFL pins. On reset, TCRF is initialized to H'00.
Timer control status register F (TCSRF)	
Functions	TCSRF is an 8-bit read/write register. It selects counter clear, sets the overflow flag, sets the compare match flag, and enables interrupt requests due to overflows.
16-bit timer counter F (TCF)	
Functions	TCF is a 16-bit read/write up-counter. It consists of cascade connections of the 8-bit timer counters (TCFH and TCFL). In addition to use as a 16-bit counter employing TCFH for the upper eight bits and TCFL for the lower eight bits, TCFH and TCFL can also be used as independent 8-bit counters. TCFH and TCFL can be read and written from the CPU, but when used in 16-bit mode, data transfer with the CPU is via a temporary register (TEMP). On reset, TCFH and TCFL are both initialized to H'00. If the CKSH2 bit of TCRF is clear to 0, then TCF functions as a 16-bit counter. The TCF input clock can be selected using the CKSL2 to CKSL0 bits of TCRF. The CCLRH bit of TCSRF can be used to clear TCF on compare match. When TCF overflows, the OVFH bit of TCSRF is set to 1, and if the IENTFH bit of IENR2 is 1, an interrupt request is sent to the CPU.
8-bit timer counter FH (TCFH)	
Functions	TCFH can be made to operate as an independent 8-bit counter by setting the CKSH2 bit of TCRF to 1. The TCFH input clock is selected using the CKSH2 to CKSH0 bits of TCRF. The CCLRH bit of TCSRF can be used to clear TCFH on a compare match. When there is an overflow of TCFH, the OVFH bit of TCSRF is set to 1. At this time, if the OVIEH bit of TCSRF is 1, the IRRTFH bit of IRR2 is set to 1, and if the IENTFH bit of IENR2 is 1, an interrupt request is sent to the CPU.
8-bit timer counter FL (TCFL)	
Functions	TCFL can be made to operate as an independent 8-bit counter by setting the CKSH2 bit of TCRF to 1. The TCFL input clock is selected using the CKSL2 to CKSL0 bits of TCRF. The CCLRL bit of TCSRF can be used to clear TCFL on a compare match. When there is an overflow of TCFL, the OVFL bit of TCSRF is set to 1. At this time, if the OVIEL bit of TCSRF is 1, the IRRTFL bit of IRR2 is set to 1, and if the IENTFL bit of IENR2 is 1, an interrupt request is sent to the CPU.

Table 1 Timer F Functions (cont)

16-bit output compare register F (OCRF)	
Functions	OCRF consists of two 8-bit read/write registers (OCRFH and OCRFL). In addition to being used as a 16-bit register of which OCRFH is the upper eight bits and OCRFL is the lower eight bits, OCRFH and OCRFL can also be used as independent 8-bit registers. OCRFH and OCRFL can be read and written from the CPU, but when used in 16-bit mode, data transfer with the CPU is via TEMP. On reset, OCRFH and OCRFL are both initialized to H'FF. On clearing the CKSH2 bit of TCRF to 0, OCRF operates as a 16-bit register. The contents of OCRF are constantly compared with TCF, and if the values of the two match, the CMFH bit of TCSRF is set to 1. At the same time, the IRRTFH bit of IRR2 is also set to 1. At this time if the IENTFH bit of IENR2 is 1, an interrupt request is sent to the CPU. The toggle output of a compare match can be output from the TMOFH pin. In addition, the TOLH bit of TCRF can be used to select the output level (high/low).
8-bit output compare register FH (OCRFH)	
Functions	When the CKSH2 bit of TCRF is set to 1, OCRF operates as two 8-bit registers (OCRFH and OCRFL). The contents of OCRFH are compared with TCFH, and the contents of OCRFL are compared with TCFL. If the values of OCRFH and TCFH match, the CMFH bit of TCSRF is set to 1. At the same time, the IRRTFH bit of IRR2 is also set to 1. At this time if the IENTFH bit of IENR2 is 1, an interrupt request is sent to the CPU. The toggle output of a compare match can be output from the TMOFH pin. In addition, the TOLH pin of TCRF can be used to select the output level (high/low).
8-bit output compare register FL (OCRFL)	
Functions	When the CKSH2 bit of TCRF is set to 1, OCRF operates as two 8-bit registers (OCRFH and OCRFL). The contents of OCRFH are compared with TCFH, and the contents of OCRFL are compared with TCFL. If the values of OCRFL and TCFL match, the CMFL bit of TCSRF is set to 1. At the same time, the IRRTFL bit of IRR2 is also set to 1. At this time if the IENTFL bit of IENR2 is 1, an interrupt request is sent to the CPU. The toggle output of a compare match can be output from the TMOFL pin. In addition, the TOLL pin of TCRF can be used to select the output level (high/low).
Timer FH interrupt request flag (IRRTFH)	
Functions	IRRTFH of the OCRFH register is set to 1, if TCF matches OCRF in 16-bit mode, if TCFH matches OCRFH in 8-bit mode, or if TCF and TCFH overflow when IENTFH is set to 1. IRRTFH is cleared to 0, if IRRTFH is written to 1 when IRRTFH is set to 1.
Timer FL interrupt request flag (IRRTFL)	
Functions	IRRTFL of the OCRFL register is set to 1, if TCFL matches OCRFL in 8-bit mode or if TCFL overflows when IENTFL is set to 1. IRRTFL is cleared to 0, if IRRTFL is written to 1 when IRRTFL is set to 1.

Table 1 Timer F Functions (cont)

Timer FH interrupt enable (IENTFH)	
Functions	IENTFH enables or disables interrupt requests caused by timer FH compare matches or overflows.
Timer FL interrupt enable (IENTFL)	
Functions	IENTFL enables or disables interrupt requests caused by timer FL compare matches or overflows.
Timer F event input (TMIF)	
Functions	TIMF is used as an event input pin to be input to the TCFL.
Timer FH output (TMOFH)	
Functions	TMOFH is a timer FH toggle output pin.
Timer FL output (TMOFL)	
Functions	TMOFL is a timer FL toggle output pin.

4. The method for setting the timer FH interrupt cycle is explained below.

   In this task example, 38.4 kHz is used as the subclock, and the timer F operates as a clock time base.

   By setting TCRF CKSL2 to 1, CKSL1 to 1 and CKSL0 to 1, the TCF input clock is set to $\phi_w/4$. Here $\phi_w/4$ is given by

   $$\phi_w/4 = 38.4\ \text{kHz}/4 = 9.6\ \text{kHz}$$

   Hence the TCF input clock cycle is

   $$1/9.6\ \text{kHz} \cong 104.167\ \mu\text{s}$$

   Here if OCRF is set to H'2580, then the time until the values of TCF and OCRF match is calculated as

   $$\text{H'2580} \times (1/9.6\ \text{kHz}) = 9600 \times 104.167\ \mu\text{s} = 1\ \text{sec}$$

   Hence the settings for OCRF used to set the timer FH interrupt cycle T_{FH} are calculated by the following equation.

   $$\text{OCRF setting} = T_{FH}/(1/9.6\ \text{kHz}) = T_{FH} \times 9.6\ \text{kHz}$$

   Table 2 shows timer FH interrupt cycle T_{FH} values and OCRA setting examples.

Table 2 Examples of Timer FH Interrupt Cycles and OCRF Settings

T_{FH} (sec)	Calculation	OCRA Setting
0.125	0.125 sec × 9.6 kHz = 1200	H'04B0
0.25	0.25 sec × 9.6 kHz = 2400	H'0960
0.5	0.5 sec × 9.6 kHz = 4800	H'12C0
1	1 sec × 9.6 kHz = 9600	H'2580
0.125	2 sec × 9.6 kHz = 19200	H'4B00

5. The interface with the CPU is explained below.

   TCF and OCRF are 16-bit read/write registers. On the other hand, the data bus between the CPU and internal peripheral modules has an 8-bit data width. Hence when the CPU accesses TCF or OCRF, it must do so via the 8-bit temporary register (TEMP).

   When reading or writing TCF or writing OCRF in 16-bit mode, operations must always be performed in 16-bit units (with byte-size MOV instructions executed twice in succession), in the order of the upper byte and lower byte. If only the upper byte or only the lower byte is accessed, the data is not transferred correctly.

   In 8-bit mode, there are no restraints on access order.

a. Write operation

By writing the upper byte, the upper byte data is transferred to TEMP. Next the lower byte is written; the data in TEMP is written to the upper byte register, and the lower-byte data is written directly to the lower byte register.

The TCF write operation when H'AA55 is written to TCF is illustrated in figure 3.

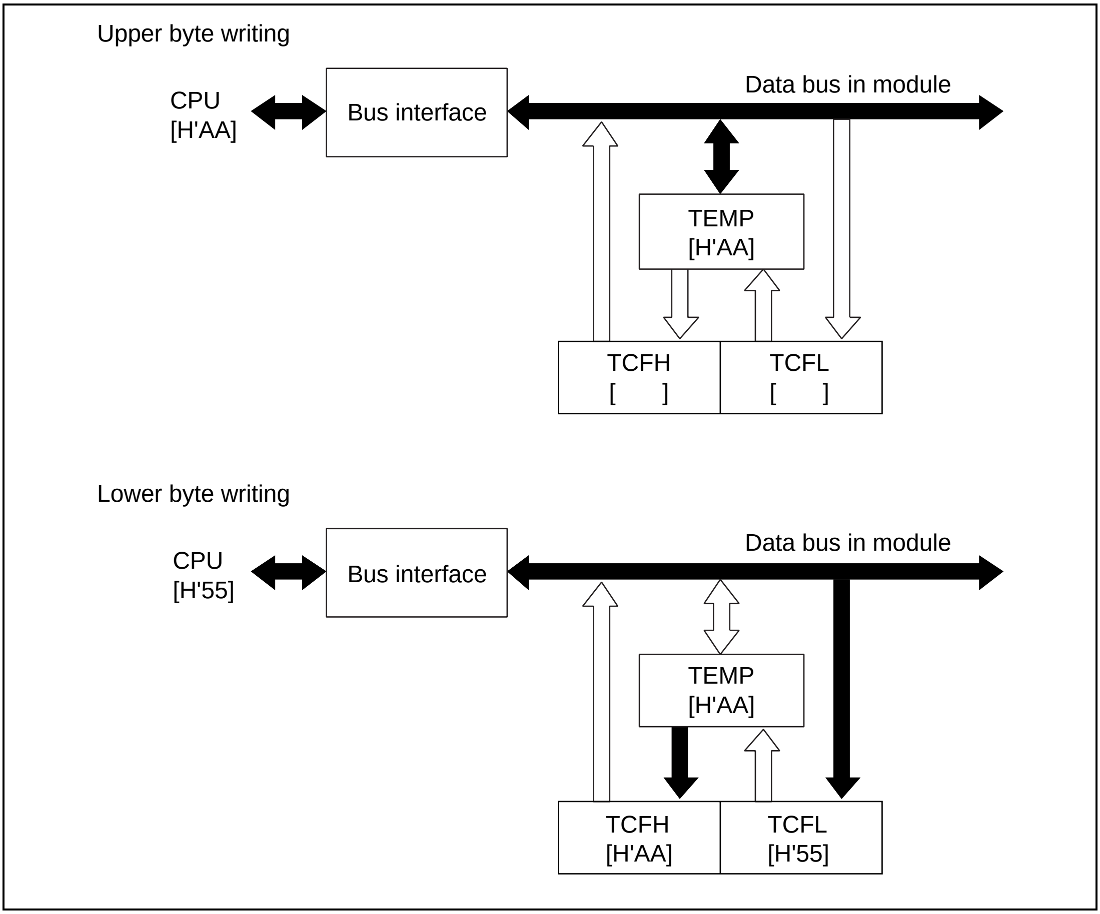


Figure 3 TCF Write Operation

b. Read operation

In the case of TCF, when the upper byte is read, the upper byte data is transferred directly to the CPU, and the lower byte data is transferred to TEMP. Next the lower byte data is read; the lower byte data in TEMP is transferred to the CPU.

In the case of OCRF, in upper byte reading the upper byte data is transferred directly to the CPU. In lower byte reading the lower byte data is also transferred directly to the CPU.

Figure 4 shows a TCF read operation when TCF contains H'AAFF.

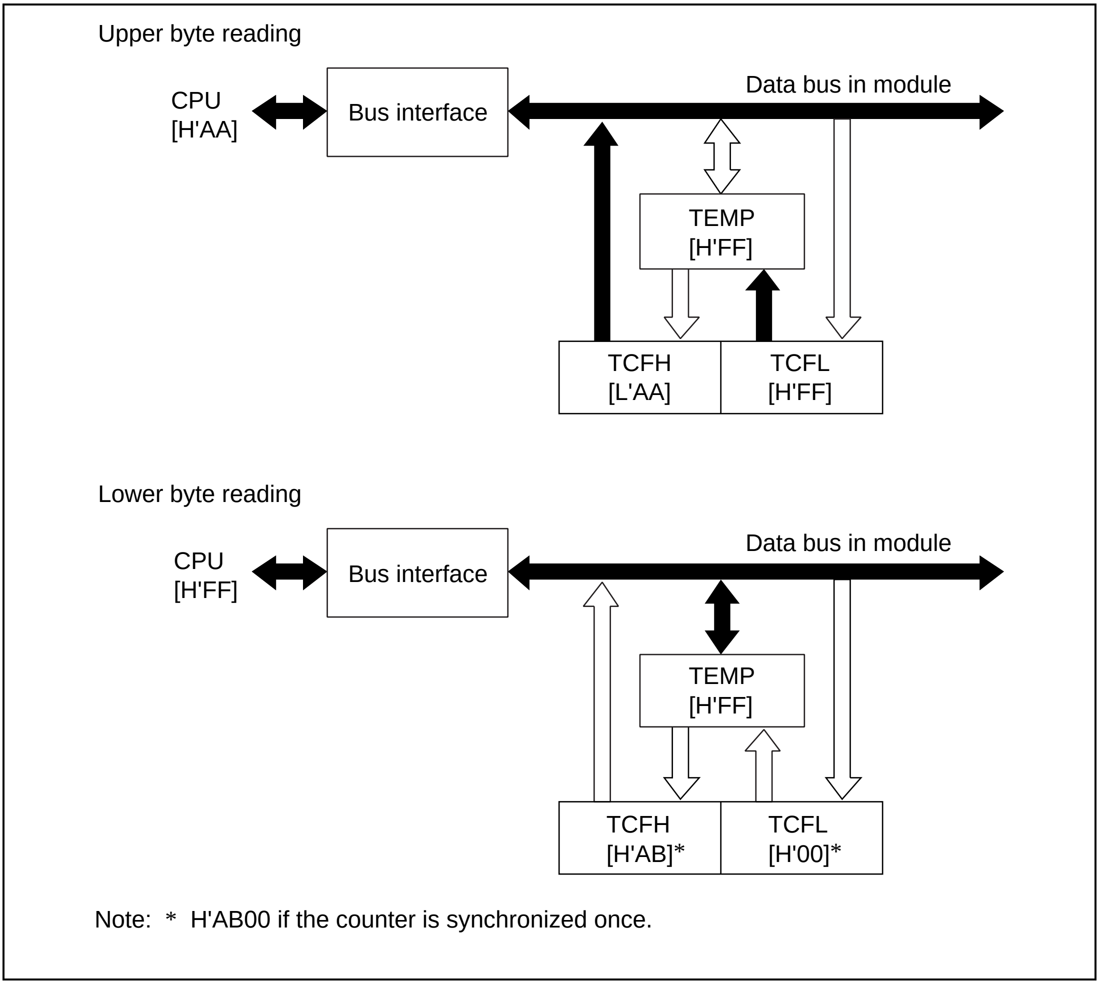


Figure 4 TCF Read Operation

6. Notes on use of the timer F

While the timer F is operating in 16-bit timer mode, the following conflicts and operations may occur.

a. When all bits match and a compare match signal is generated, the TMOFH pin output is toggled. When TCRF writing by a MOV instruction and a compare match signal occur simultaneously, TOLH data resulting from TCRF writing is output to pin TMOFH. In 16-bit mode, the TMOFL pin output is undefined, and should not be used. Use it as a port instead.

b. When OCRFL writing and compare match signal generation occur simultaneously, the compare match signal is invalid. However, when the data to be written matches the counter value, at that point a compare match signal is generated. The compare match signal is output in synchronization with the TCFL clock, so that if the clock is stopped, no compare match signal is generated even if a compare match occurs.

c. When all 16 bits match and a compare match signal is generated, the compare match flag CMFH is set. Similarly, if conditions for the lower 8 bits are satisfied, CMFL is set.

d. When there is a TCF overflow, OVFH is set; but when the lower 8 bits of OVFL overflow, if set conditions are satisfied, OVFL is set. If TCFL writing and overflow signal output occur simultaneously, the overflow signal is not output.

e. When, in active mode and sleep mode, $\phi_w/4$ is selected as the TCF internal clock, the system clock and internal clock are out of synchronization, and so synchronization is secured by an internal synchronization circuit. This results in a maximum count cycle error of $1/\phi$ (sec). In order to prevent this error from occurring, the system must be operated in subactive mode, subsleep mode, or watch mode.

7. Table 3 indicates function allocation in this task example.

Table 3 Function Allocation

Function	Function Allocation
TCRF	Sets 16-bit mode and selects TCFL input clock.
TCSRF	Selects counter clear, sets the overflow flag, sets the compare match flag, and enables/disables interrupt requests due to overflows.
TCF	Functions as a 16-bit up-counter configured by the connection of 8-bit timer counters TCFH and TCFL. TCF counts internal clock ($\phi_W/4$), and sets the IRTFH and CMFH bits to 1 if a compare match occurs.
OCRF	16-bit register configured by the connection of 8-bit registers OCRFH and OCRFL. If OCRF matches TCF, a compare match signal is generated.
IRRTFH	Indicates if a timer FH interrupt is requested or not.
IENTFH	Enables or disables timer FH interrupt requests.
ϕ_w	Subclock frequency, 38.4 kHz in this task example.

Explanation of Operation

1. Operation of the timer F 16-bit timer mode is explained below.

   Timer F is a 16-bit counter which is incremented each time a clock pulse is input. The value of the timer counter F is continuously compared with the value set in the output compare register F; when they match, the counter is cleared, an interrupt request is issued, and port toggle output is possible. The timer can also operate as two independent 8-bit timers.

   When the CKSH2 bit of the timer control register F (TCRF) is set to 0, timer F operates as a 16-bit timer.

   Immediately after reset, the timer counter F (TCF) is set to H'0000, the output compare register F (OCRF) is set to H'FFFF, and the timer control register F (TCRF) and timer control status register F (TCSRF) are both initialized to H'00. The counter begins to be incremented by input from an external event (TMIF). The external event edge is selected through the IEG3 bit of the IRQ edge select register (IEGR).

   As the operating clock for timer F, the CKSL2 through CKSL0 bits of TCRF can be used to select from three kinds of internal clock output by prescaler S, an internal clock which is 1/4 the subclock, or an external clock.

   The contents of TCF and OCRF are continuously compared; when the two match, the CMFH bit of TCSRF is set to 1. At this time if the IENTFH bit of IENR2 is 1 an interrupt request is sent to the CPU, and at the same time the TMOFH pin output is toggled. Also, if the CCLRH bit of TCSRF is 1, TCF is cleared. The output from pin TMOFH can be set by the TOLH bit of TCRF.

   When TCF overflows (H'FFFF → H'0000), the OVFH bit of TCSRF is set to 1. At the time, if both the OVIEH bit of TCSRF and the IENTFH bit of IENR2 are 1, an interrupt request is sent to the CPU.

2. Timer F operating modes are indicated in table 4.

Table 4 Timer F Operating Modes

Operating Mode	TCF	OCRF	TCRF	TCSRF
Reset	Reset	Reset	Reset	Reset
Active	Functions	Functions	Functions	Functions
Sleep	Functions	Held	Held	Held
Watch	Functions/ Halted*	Held	Held	Held
Subactive	Functions/ Halted*	Functions	Functions	Functions
Subsleep	Functions/ Halted*	Held	Held	Held
Standby	Halted	Held	Held	Held
Module standby	Halted	Held	Held	Held

Note: * If $\phi_w/4$ is selected as the TCF's internal clock in active mode or sleep mode, the system clock and internal clock are mutually asynchronous, and so synchronization is established by a synchronization circuit. This results in a maximum error of $1/\phi(s)$ in the count period.

When the counter is operated in subactive mode, watch mode, or subsleep mode, $\phi_w/4$ must always be selected as the internal clock. The counter will not operate if any other internal clock is selected.

3. Figure 5 illustrates the principle of operation in this task example.

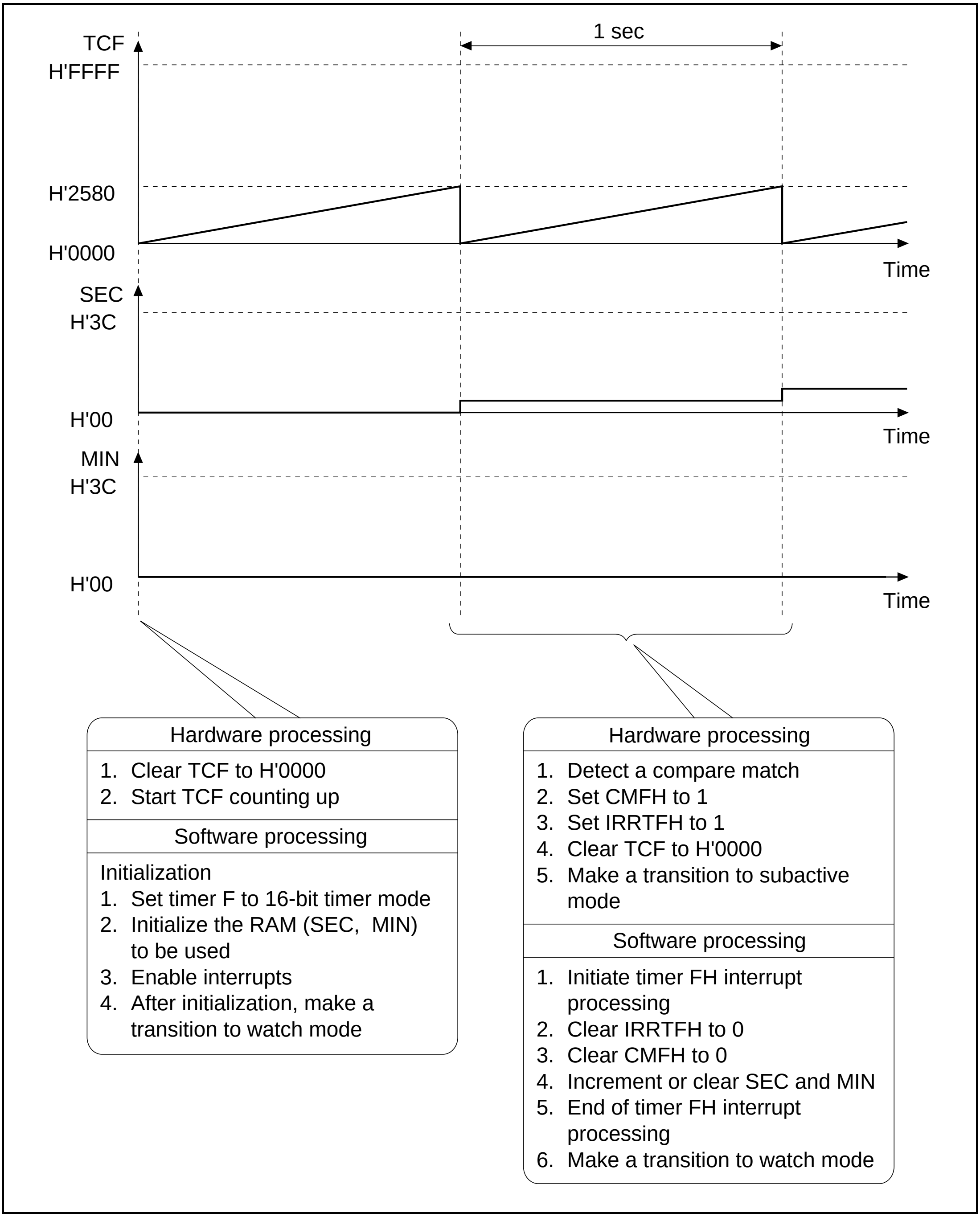


Figure 5 Operation Principle

Explanation of Software

1. Explanation of Modules

   Table 5 explains the modules in this task example.

Table 5 Explanation of Modules

Module Name	Label Name	Function
Main routine	MAIN	Initializes the stack pointer, RAM and timer F, enables interrupts, and executes a transition to watch mode.
Timer F interrupt processing routine	TFINT	Clears IRRTFH and CMFH to 0, save register data, increment and clear SEC and MIN defined in RAM, restores register data.

2. Explanation of Arguments

   In this task example, no arguments are used.

3. Explanation of Internal Registers Used

Table 6 gives explanations of the internal registers used in this task example.

Table 6 Explanation of Internal Registers Used

Register Name		Description	RAM Address	Setting
TCRF	CKSH2 to CKSH0	Timer control register F (Clock select H 2 to 0) Select the clock input to TCFH from among four internal clock source or TCFL overflow. • When CKSH2 = 0, CKSH1 = 0 and CKSH0 = 0, TCFL overflow is selected • When CKSH2 = 0, CKSH1 = 0 and CKSH0 = 1, TCFL overflow is selected • When CKSH2 = 0, CKSH1 = 1 and CKSH0 = 0, TCFL overflow is selected • When CKSH2 = 1, CKSH1 = 0 and CKSH0 = 0, internal clock ϕ/32 is selected • When CKSH2 = 1, CKSH1 = 0 and CKSH0 = 1, internal clock ϕ/16 is selected • When CKSH2 = 1, CKSH1 = 1 and CKSH0 = 0, internal clock ϕ/4 is selected • When CKSH2 = 1, CKSH1 = 1 and CKSH0 = 1, internal clock ϕw/4 is selected • Note that CKSH2 = 0, CKSH1 = 1 and CKSH0 = 1 cannot be specified	H'FFB6 Bit 6 to bit 4	CKSH2 = 0 CKSH1 = 0 CKSH0 = 0

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
TCRF	CKSL2 to CKSL0	Timer control register F (Clock select L 2 to 0) Select the clock input to TCFL from among four internal clock source or an external event. • When CKSL2 = 0, CKSL1 = 0 and CKSL0 = 0, an external event is selected • When CKSL2 = 0, CKSL1 = 0 and CKSL0 = 1, an external event is selected • When CKSL2 = 0, CKSL1 = 1 and CKSL0 = 0, an external event is selected • When CKSL2 = 1, CKSL1 = 0 and CKSL0 = 0, internal clock ϕ/32 is selected • When CKSL2 = 1, CKSL1 = 0 and CKSL0 = 1, internal clock ϕ/16 is selected • When CKSL2 = 1, CKSL1 = 1 and CKSL0 = 0, internal clock ϕ/4 is selected • When CKSL2 = 1, CKSL1 = 1 and CKSL0 = 1, internal clock ϕw/4 is selected • Note that CKSL2 = 0, CKSL1 = 1 and CKSL0 = 1 cannot be specified	H'FFB6 Bit 2 to bit 0	CKSL2 = 1 CKSL1 = 1 CKSL0 = 1
TCSRF	OVFH	Timer control/status register F (Timer overflow flag H) A status flag indicating overflow of TCF. • When OVFH = 0, indicates no overflow of TCF • When OVFH = 1, indicates TCF overflow	H'FFB7 Bit 7	0
TCSRF	CMFH	Timer control/status register F (Compare match flag H) A status flag indicating that TCF has matched OCRF. • When CMFH = 0, indicates no compare match between TCF and OCRF • When CMFH = 1, indicates TCF has matched OCRF	H'FFB7 Bit 6	0

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
TCSRF	OVIEH	Timer control/status register F (Timer overflow interrupt enable H) Enables or disables interrupt generation when TCF overflows. • When OVIEH = 0, disables TCF overflow interrupt requests • When OVIEH = 1, enables TCF overflow interrupt requests	H'FFB7 Bit 5	1
TCSRF	CCLRH	Timer control/status register F (Counter clear H) Selects whether or not TCF is cleared when TCF has matched OCRF. • When CCLRH = 0, disables TCF clear by compare match • When CCLRH = 1, enables TCF clear by compare match	H'FFB7 Bit 4	1
TCFH		Timer counter FH Upper 8 bits of 16-bit timer counter F (TCF); functions as an 8-bit up-counter using a TCFL overflow signal as an input clock.	H'FFB8	H'00
TCFL		Timer counter FL Lower 8 bits of 16-bit timer counter F (TCF); functions as an 8-bit up-counter using ϕw/4 of internal clock as an input clock.	H'FFB9	H'00
OCRFH		Output compare register FH Upper 8 bits of 16-bit output compare register (OCRF); generates a compare match signal when OCRF has matched TCF.	H'FFBA	H'25
OCRFL		Output compare register FL Lower 8 bits of 16-bit output compare register (OCRF); generates a compare match signal when OCRF has matched TCF.	H'FFBB	H'80

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
IENR2	IENTFH	Interrupt enable register 2 (Timer FH interrupt enable) Enables or disables timer FH interrupt requests. • When IENTFH = 0, disables timer FH interrupt requests • When IENTFH = 1, enables timer FH interrupt requests	H'FFF4 Bit 3	1
IRR2	IRRTFH	Interrupt request register 2 (Timer FH interrupt request flag) Indicates whether there has been a timer FH interrupt request. • When IRRTFH = 0, indicates that no timer FH interrupt has been requested • When IRRTFH = 1, indicates that a timer FH interrupt has been requested	H'FFF7 Bit 3	0
SYSCR1	SSBY	System control register 1 (Software standby) Carries out transitions to standby mode or watch mode. • When SSBY = 0, after executing a SLEEP instruction in active mode, causes a transition to sleep mode, or after executing a SLEEP instruction in subactive mode, causes a transition to subsleep mode. • When SSBY = 1, after executing a SLEEP instruction in active mode, causes a transition to standby mode or to watch mode, or after executing a SLEEP instruction in subactive mode, causes a transition to watch mode.	H'FFF0 Bit 7	1

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
SYSCR1	STS2 STS1 STS0	System control register 1 (Standby timer select 2 to 0) Specify the time for the CPU and peripheral functions to wait until the clock stabilizes when standby mode or watch mode is canceled and a transition is made to active mode due to a specific interrupt. Note that the standby time must be specified to be equal to or longer than the oscillation stabilization time according to the operating frequency. • When STS2 to STS1 = 000, standby time is 8,192 states • When STS2 to STS1 = 001, standby time is 16,384 states • When STS2 to STS1 = 010, standby time is 32,768 states • When STS2 to STS1 = 011, standby time is 65,536 states • When STS2 to STS1 = 100, standby time is 131,072 states • When STS2 to STS1 = 101, standby time is 2 states • When STS2 to STS1 = 110, standby time is 8 states • When STS2 to STS1 = 111, standby time is 16 states	H’FFF0 Bit 6 to bit 4	STS2 = 0 STS1 = 0 STS0 = 0
SYSCR1	LSON	System control register 1 (Low speed on flag) When watch mode is canceled, selects either the system clock (ϕ) or the subclock (ϕ_{sub}) as the CPU operating clock. • When LSON = 0, selects the system clock (ϕ) as the CPU operating clock • When LSON = 1, selects the subclock (ϕ_{sub}) as the CPU operating clock	H'FFF0 Bit 3	1

Table 6 Explanation of Internal Registers Used (cont)

Register Name		Description	RAM Address	Setting
SYSCR2	DTON	System control register 2 (Direct transfer on flag) Specifies whether or not to make direct transitions among active (high-speed) mode, active (medium-speed) mode, and subactive mode when a SLEEP instruction is executed. • When DTON = 0, if a SLEEP instruction is executed in active mode, a transition to standby mode, watch mode or sleep mode occurs • When DTON = 1, if a SLEEP instruction is executed in active (high-speed) mode, a direct transition occurs to active (medium-speed) mode (when SSBY = 1, MSON = 1, LSON = 0) or to subactive mode (when SSBY = 1, TMA = 1, LSON = 1)	H'FFF1 Bit 3	0
SYSCR2	MSON	System control register 2 (Medium speed on flag) Selects whether to operate in active (high-speed) mode or in active (medium-speed) mode after cancellation of standby mode, watch mode, or sleep mode. • When MSON = 0, operates in active (high-speed) mode • When MSON = 1, operates in active (medium- speed) mode	H'FFF1 Bit 2	0
TMA	TMA3	Timer mode register A (Internal clock select 3) Selects the clock input to TCA. • When TMA3 = 0, PSS is selected as the TCA input clock source • When TMA3 = 1, PSW is selected as the TCA input clock source	H'FFB0 Bit 3	1

4. Explanation of RAM Usage

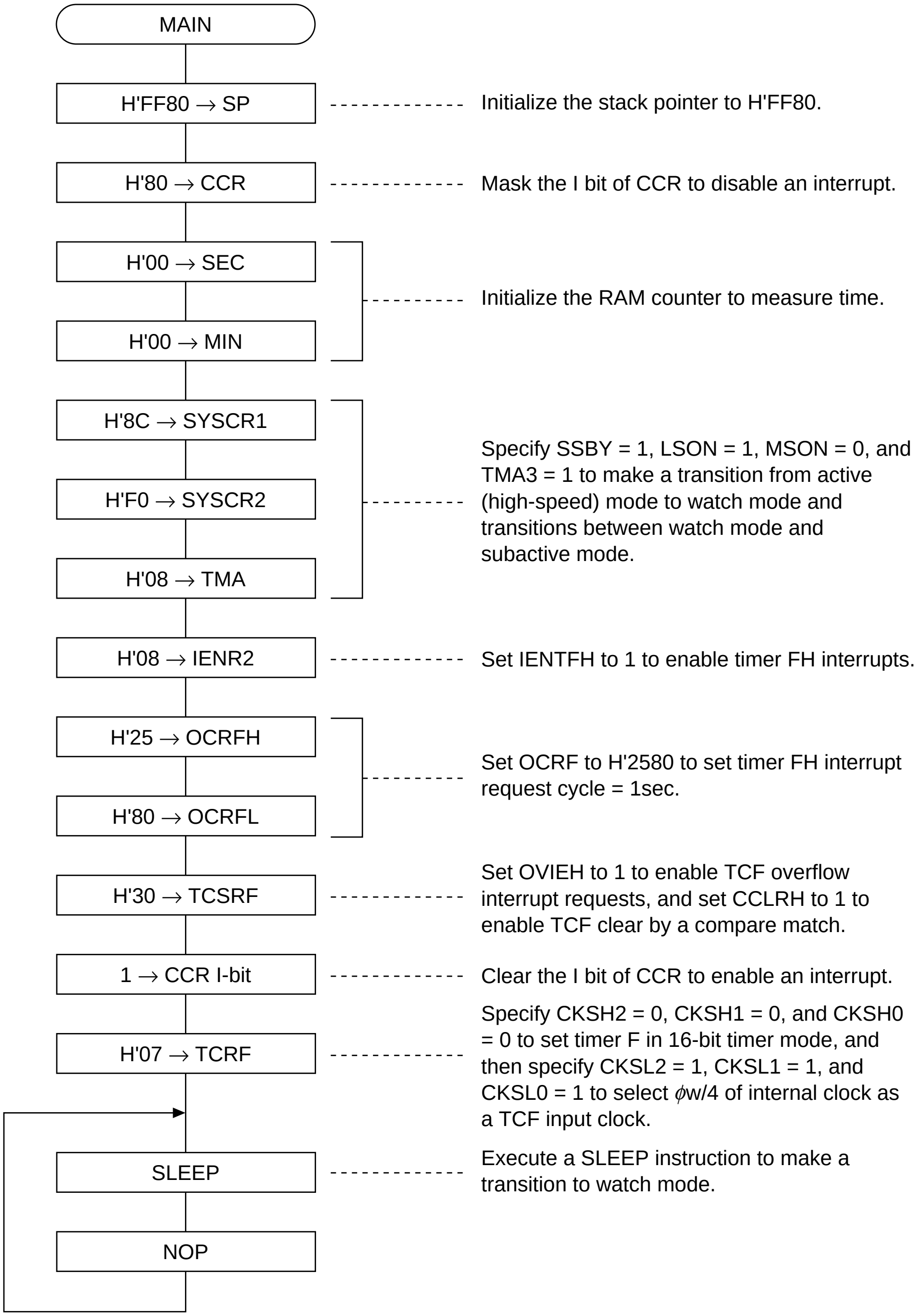
Table 7 explains RAM usage for this task example.

Table 7 Explanation of RAM Usage

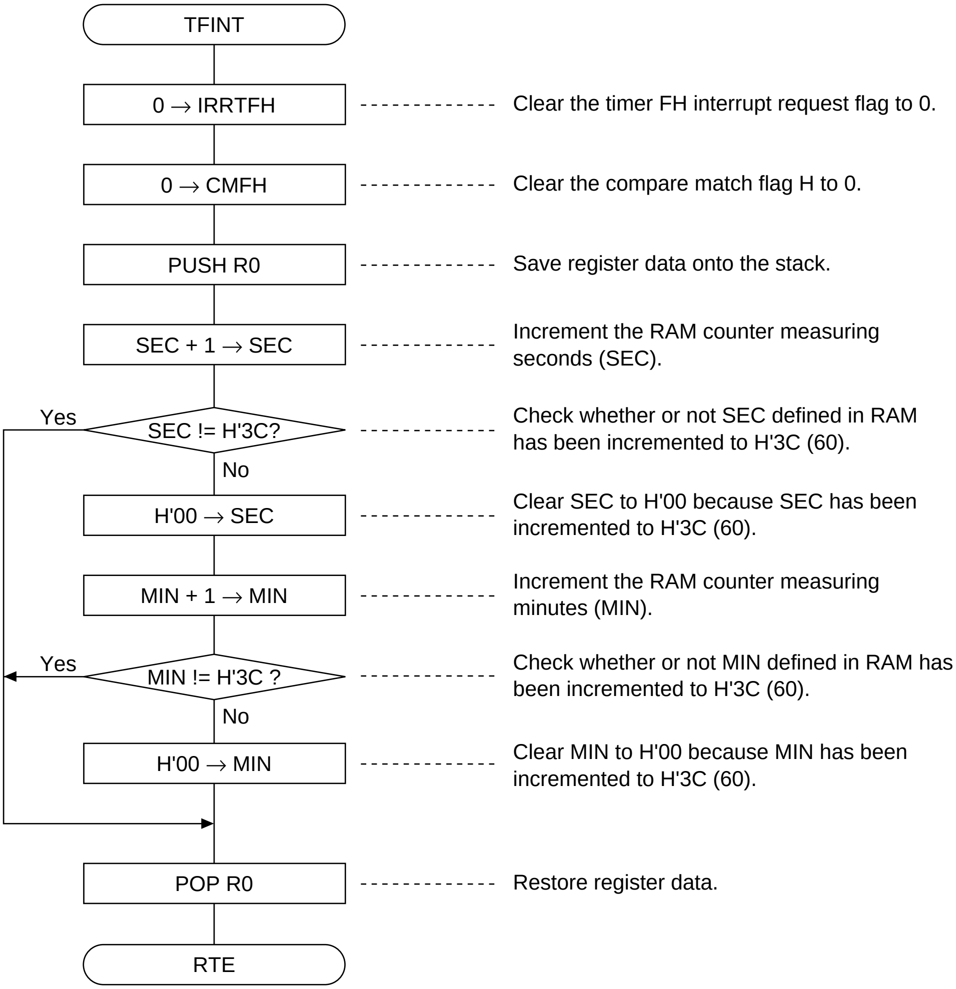
Label Name	Function	RAM Address	Modules Used
SEC	Counter used as a clock, counting the minutes	H'F780	MAIN, TFINT
MIN	Counter used as a clock, counting the seconds	H'F781	MAIN, TFINT

Flowchart

1. Main routine



2. Timer F interrupt processing routine



Program Lists

```
;*************************************************************
;*   H8/3867 Application Note
;*
;*     'Timer F -Clock Time Base-'
;*
;*              Function : Timer F
;*
;*              External Clock : 6MHz
;*              Internal Clock : 3MHz
;*              Sub Clock      : 38.4kHz
;*************************************************************
;
        .cpu      300l
;
;*************************************************************
;*       Symbol Defnition
;*************************************************************
;
TMA      .equ      h'ffb0      ;Timer Mode Register A
TCRF     .equ      h'ffb6      ;Timer Control Register F
TCSRF    .equ      h'ffb7      ;Timer Control/Status Register F
TCFH     .equ      h'ffb8      ;8-bit Timer Counter FH
TCFL     .equ      h'ffb9      ;8-bit Timer Counter FL
OCRFH    .equ      h'ffba      ;Output Compare Register FH
OCRFL    .equ      h'ffbb      ;Output Compare Register FL
SYSCR1   .equ      h'fff0      ;System Control Register 1
SYSCR2   .equ      h'fff1      ;System Control Register 2
IENR2    .equ      h'fff4      ;Interrupt Enable Register 2
IRR2     .equ      h'fff7      ;Interrupt Request Register 2
;
;*************************************************************
;*       RAM Allocation
;*************************************************************
;
SEC      .equ      h'f780      ;Second Counter
MIN      .equ      h'f781      ;Minute Counter
```

```
;
;*********************************************************
;*       Vector Address
;*********************************************************
;
        .org      h'0000
        .data.w   MAIN        ;No.0 Reset Interrupt(H'0000-H'0001)
;
        .org      h'0008
        .data.w   MAIN        ;No.4 _IRQ0 Interrupt(H'0008-H'0009)
        .data.w   MAIN        ;No.5 _IRQ1 Interrupt(H'000A-H'000B)
        .data.w   MAIN        ;No.6 _IRQ2 Interrupt(H'000C-H'000D)
        .data.w   MAIN        ;No.7 _IRQ3 Interrupt(H'000E-H'000F)
        .data.w   MAIN        ;No.8 _IRQ4 Interrupt(H'0010-H'0011)
        .data.w   MAIN        ;No.9 _WKP0-_WKP7 Interrupt(H'0012-H'0013)
;
        .org      h'0016
        .data.w   MAIN        ;No.11 Timer A Interrupt(H'0016-H'0017)
        .data.w   MAIN        ;No.12 AEC Interrupt(H'0018-H'0019)
        .data.w   MAIN        ;No.13 Timer C Interrupt(H'001A-H'001B)
        .data.w   MAIN        ;No.14 Timer FL Interrupt(H'001C-H'001D)
        .data.w   TFINT       ;No.15 Timer FH Interrupt(H'001E-H'001F)
        .data.w   MAIN        ;No.16 Timer G Interrupt(H'0020-H'0021)
        .data.w   MAIN        ;No.17 SCI31 Interrupt(H'0022-H'0023)
        .data.w   MAIN        ;No.18 SCI32 Interrupt(H'0024-H'0025)
        .data.w   MAIN        ;No.19 A/D Converter Interrupt(H'0026-H'0028)
        .data.w   MAIN        ;No.20 Direct Transfer Interrupt(H'0028-H'0029)
;
;*********************************************************
;*       MAIN : Main Routine
;*********************************************************
;
        .org      h'1000
;
MAIN:   .equ      $
        mov.w     #H'ff80,sp  ;Initialize Stack Pointer
        orc       #h'80,ccr   ;Interrupt Disable
```

```
;
        sub.b     r0l,r0l      ;Initialize RAM
        mov.b     r0l,@SEC
        mov.b     r0l,@MIN
;
        mov.b     #h'8c,r0l    ;Initialize System Control
        mov.b     r0l,@SYSCR1
        mov.b     #h'f0,r0l
        mov.b     r0l,@SYSCR2
        mov.b     #h'08,r0l
        mov.b     r0l,@TMA
;
        mov.b     #h'08,r0l    ;Timer F Interrupt Enable
        mov.b     r0l,@IENR2
;
        mov.b     #h'25,r0h    ;Initialize Timer F
        mov.b     #h'80,r0l
        mov.b     r0h,@OCRFH
        mov.b     r0l,@OCRFL
        mov.b     #h'30,r0l
        mov.b     r0l,@TCSRF
;
        andc      #h'7f,ccr    ;Interrupt Enable
;
        mov.b     #h'07,r0l    ;Initialize TCFL Input Clock
        mov.b     r0l,@TCRF
;
LOOP:   sleep                  ;Transfer to Watch Mode
        nop
        bra       LOOP
;
;*************************************************************
;*      TFINT : Timer F Interrupt Routine
;*************************************************************
;
TFINT:  .equ      $
        bclr      #3,@IRR2     ;Clear Timer F Interrupt Request Flag
```

```
        bclr      #6,@TCSRF    ;Clear Compare Match Flag H
;
        push      r0           ;Store r0
;
        mov.b     @SEC,r0l     ;Load Second Counter
        mov.b     @MIN,r0h     ;Load Minute Counter
        inc       r0l          ;Increment Second Counter
        cmp.b     #h'3c,r0l    ;@SEC = d'60 ?
        bne       INTEXT       ;No. Exit
        mov.b     #h'00,r0l    ;Yes. Initialize Second Counter
        inc       r0h          ;Increment Minute Counter
        cmp.b     #h'3c,r0h    ;@MIN = d'60 ?
        bne       INTEXT       ;No. Exit
        mov.b     #h'00,r0h    ;Yes. Initialize Minute Counter
;
INTEXT: mov.b     r0h,@MIN     ;Store Minute Counter
        mov.b     r0l,@SEC     ;Store Second Counter
;
        pop       r0           ;Restore r0
;
        rte
;
        .end
```

H8/3867 Series Application Note

Publication Date: 1st Edition, August 1999
Published by: Electronic Devices Sales & Marketing Group
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
UL Media Co., Ltd.