SINO WEALTH

SH69P848A

OTP 2K 4-bit Micro-controller with 10-bit SAR A/D Converter

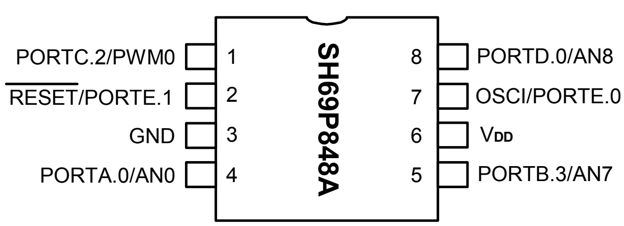
Features

- SH6610D-based single-chip 4-bit micro-controller with 10-bit SAR ADC converter
- OTPROM: 2K X 16bits
- RAM: 253 X 4bits
  - 61 System control register
  - 192 Data memory
- Operation voltage:
  - f_{OSC} = 30kHz - 4MHz, V_{DD} = 2.4V - 5.5V
  - f_{OSC} = 4MHz - 10MHz, V_{DD} = 4.5V - 5.5V
- 5 CMOS bi-directional I/O ports plus 1 CMOS open drain output port (PORTE.1)
- 8-level stack (including interrupts)
- Two 8-bit auto re-loaded timer/counter
- Warm-up timer
- Powerful interrupt sources:
  - ADC interrupt
  - Internal interrupt (Timer1, Timer0)
  - External interrupts: PORTB/D (Falling edge)
- Oscillator (code option)
  - External RC oscillator: 400kHz - 10MHz
  - Internal RC oscillator: 4MHz ± 5%
  - External clock: 30kHz - 10MHz
- Instruction cycle time (4/f_{OSC})
- Two low power operation modes: HALT and STOP
- Reset
  - Built-in watchdog timer (code option)
  - Built-in power-on reset (POR)
  - Built-in low voltage reset (LVR)
- Two-level low voltage reset (LVR) (code option)
- 3 channels 10-bit resolution analog/digital converter (ADC)
- Read ROM table function
- 1 channels (8+2) bits PWM output
- OTP type/code protection
- 8-pin SOP package

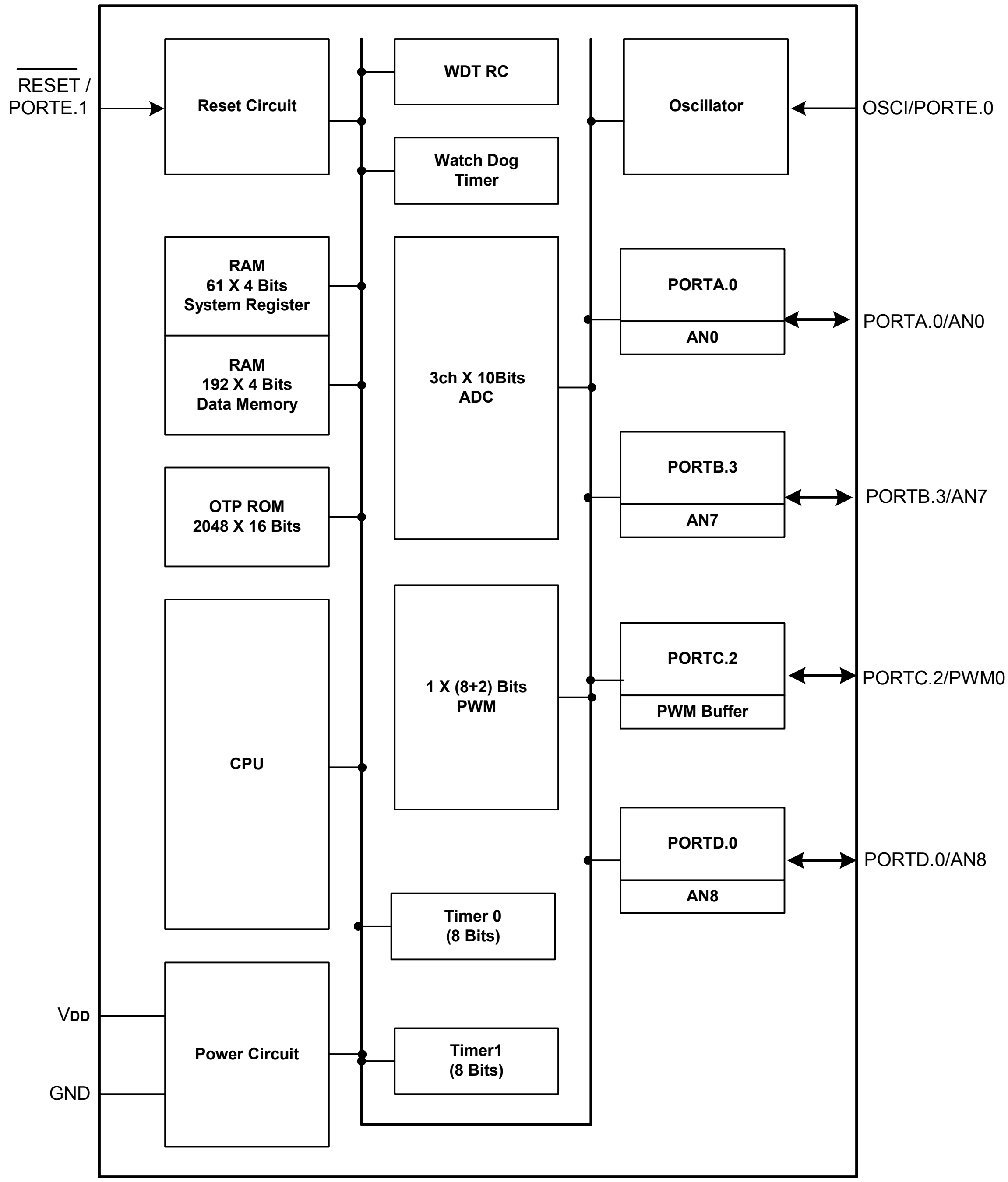
General Description

SH69P848A is a single-chip 4-bit micro-controller. This device integrates a SH6610D CPU core, 253 nibbles of RAM, 2K words of OTP ROM, two 8-bit timer/counter, 3 channels 10-bit ADC, 1 channels (8+2) bits high speed PWM output, on-chip oscillator clock circuitry, on-chip watchdog timer, low voltage reset function and support power saving modes to reduce power consumption. The SH69P848A is suitable for charger application.

Pin Configuration



Block Diagram



Pin Descriptions

Pin No.	Designation	I/O	Description
1	PORTC.2 /PWM0	I/O O	Bit programmable bi-directional I/O port Shared with PWM0 output
2	$\overline{\text{RESET}}$ /PORTE.1	I O	Reset pin input, (low active) Open drain output port (selected by code option)
3	GND	P	Ground pin
4	PORTA.0 /AN0	I/O I	Bit programmable bi-directional I/O port Shared with ADC input channel AN0
5	PORTB.3 - /AN7	I/O I I	Bit programmable bi-directional I/O port Vector port interrupt (falling edge active) Shared with ADC input channel AN7
6	VDD	P	Power supply pin
7	OSCI /PORTE.0	I I/O	Oscillator input pin, connect to external resistor of external RC oscillator. Bi-directional I/O port in the internal RC mode
8	PORTD.0 - /AN8	I/O I I	Bit programmable bi-directional I/O port Vector port interrupt (falling edge active) Shared with ADC input channel AN8

OTP Programming Pin Description (OTP program mode)

Pin No.	Symbol	I/O	Shared by	Description
6	VDD	P	VDD	Programming power supply (+5.5V)
2	VPP	P	$\overline{\text{RESET}}$ /PORTE.1	Programming high voltage power supply (+11V)
3	GND	P	GND	Ground
7	SCK	I	OSCI /PORTE.0	Programming clock input pin
4	SDA	I/O	PORTA.0 /AN0	Programming data pin

Functional Description

1. CPU

The CPU contains the following functional blocks: Program Counter (PC), Arithmetic Logic Unit (ALU), Carry Flag (CY), Accumulator, Table Branch Register, Data Pointer (INX, DPH, DPM, and DPL) and Stacks.

1.1. PC

The PC is used for ROM addressing consisting of 12-bit:
Page Register (PC11), and Ripple Carry Counter (PC10, PC9, PC8, PC7, PC6, PC5, PC4, PC3, PC2, PC1, PC0).

The program counter is loaded with data corresponding to each instruction. The unconditional jump instruction (JMP) can be set at 1-bit page register for higher than 2K.

The program counter cans only 4K program ROM address. (Refer to the ROM description).

1.2. ALU and CY

The ALU performs arithmetic and logic operations. The ALU provides the following functions:
Binary addition/subtraction (ADC, SBC, ADD, SUB, ADI, SBI)
Decimal adjustments for addition/subtraction (DAA, DAS)
Logic operations (AND, EOR, OR, ANDIM, EORIM, ORIM)
Decisions (BA0, BA1, BA2, BA3, BAZ, BNZ, BC, BNC)
Logic Shift (SHR)

The Carry Flag (CY) holds the ALU overflow that the arithmetic operation generates. During an interrupt service or CALL instruction, the carry flag is pushed into the stack and recovered from the stack by the RTNI instruction. It is unaffected by the RTNW instruction.

1.3. Accumulator (AC)

The accumulator is a 4-bit register holding the results of the arithmetic logic unit. In conjunction with the ALU, data is transferred between the accumulator and system register, or data memory can be performed.

1.4. Table Branch Register (TBR)

Table Data can be stored in program memory and can be referenced by using Table Branch (TJMP) and Return Constant (RTNW) instructions. The TBR and AC are placed by an offset address in program ROM. TJMP instruction branch into address ((PC11 - PC8) X (2^8) + (TBR, AC)). The address is determined by RTNW to return look-up value into (TBR, AC). ROM code bit7-bit4 is placed into TBR and bit3-bit0 into AC.

1.5. Data Pointer

The Data Pointer can indirectly address data memory. Pointer address is located in register DPH (3-bit), DPM (3-bit) and DPL (4-bit). The addressing range can have 3FFH locations. Pseudo index address (INX) is used to read or write Data memory, then RAM address bit9 - bit0 which comes from DPH, DPM and DPL.

1.6. Stack

The stack is a group of registers used to save the contents of CY & PC (11-0) sequentially with each subroutine call or interrupt. The MSB is saved for CY and it is organized into 13 bits X 8 levels. The stack is operated on a first-in, last-out basis and returned sequentially to the PC with the return instructions (RTNI/RTNW).

Note:

The stack nesting includes both subroutine calls and interrupts requests. The maximum allowed for subroutine calls and interrupts are 8 levels. If the number of calls and interrupt requests exceeds 8, then the bottom of stack will be shifted out, that program execution may enter an abnormal state.

2. RAM

Built-in RAM contains general-purpose data memory and system register. Because of its static nature, the RAM can keep data after the CPU enters STOP or HALT.

2.1. RAM Addressing

Data memory and system register can be accessed in one instruction by direct addressing. The following is the memory allocation map:

System register and I/O: \$000 - \$02F, \$380 - \$38C
Data memory: \$030 - \$0EF

2.2. Configuration of System Register

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$00	IEAD	IET0	IET1	IEP	R/W	Interrupt enable flags register
$01	IRQAD	IRQT0	IRQT1	IRQP	R/W	Interrupt request flags register
$02	REV0	T0M.2	T0M.1	T0M.0	R/W	Bit0-2: Timer0 mode register Bit3 is reserved, always keep it to “0” in the user’s program
$03	REV0	T1M.2	T1M.1	T1M.0	R/W	Timer1 mode register Bit3 is reserved, always keep it to “0” in the user’s program
$04	T0L.3	T0L.2	T0L.1	T0L.0	R/W	Timer0 load/counter low nibble register
$05	T0H.3	T0H.2	T0H.1	T0H.0	R/W	Timer0 load/counter high nibble register
$06	T1L.3	T1L.2	T1L.1	T1L.0	R/W	Timer1 load/counter low nibble register
$07	T1H.3	T1H.2	T1H.2	T1H.0	R/W	Timer1 load/counter high nibble register
$08	REV0	REV0	REV0	PA.0	R/W	PORTA data register Bit1-3 are reserved, always keep it to “0” in the user’s program Refer to I/O notice
$09	PB.3	REV0	REV0	REV0	R/W	PORTB data register Bit0-2 are reserved, always keep it to “0” in the user’s program Refer to I/O notice
$0A	REV0	PC.2	REV0	REV0	R/W	PORTC data register Bit0, bit1 and bit3 are reserved, always keep it to “0” in the user’s program. Refer to I/O notice
$0B	REV0	REV0	REV0	PD.0	R/W	PORTD data register Bit1-3 are reserved, always keep it to “0” in the user’s program Refer to I/O notice
$0C	-	-	PE.1	PE.0	R/W	PORTE data register
$0D	-	-	-	-	-	Reserved
$0E	TBR.3	TBR.2	TBR.1	TBR.0	R/W	Table branch register
$0F	INX.3	INX.2	INX.1	INX.0	R/W	Pseudo index register
$10	DPL.3	DPL.2	DPL.1	DPL.0	R/W	Data pointer for INX low nibble register
$11	-	DPM.2	DPM.1	DPM.0	R/W	Data pointer for INX middle nibble register
$12	-	DPH.2	DPH.1	DPH.0	R/W	Data pointer for INX high nibble register
$13	T1GO	REV0	-	REV0	R/W	Bit3: Timer1 Start/stop register Bit0 and bit2 are reserved, always keep it to “0” in the user’s program
$14	REV0	-	-	ADCON	R/W	Bit0: ADC module on/off register Bit3 is reserved, always keep it to “0” in the user’s program
$15	GO/$\overline{\text{DONE}}$	TADC1	TADC0	ADCS	R/W	Bit0: ADC conversion time select register Bit1-2: ADC clock period select register Bit3: ADC status flag register
$16	ACR3	ACR2	ACR1	ACR0	R/W	ADC port configuration control register
$17	CH3	CH2	CH1	CH0	R/W	ADC channel control register
$18	REV1	REV1	REV1	PACR.0	R/W	PORTA input/output control register Bit1-3 are reserved, always keep it to “1” in the user’s program Refer to I/O notice
$19	PBCR.3	REV1	REV1	REV1	R/W	PORTB input/output control register Bit0-2 are reserved, always keep it to “1” in the user’s program Refer to I/O notice

2.2. Configuration of System Register (continued)

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$1A	REV1	PCCR.2	REV1	REV1	R/W	PORTC input/output control register Bit0, bit1 and bit3 are reserved, always keep it to “1” in the user’s program. Refer to I/O notice.
$1B	REV1	REV1	REV1	PDCR.0	R/W	PORTD input/output control register Bit1-3 are reserved, always keep it to “1” in the user’s program Refer to I/O notice
$1C	-	-	-	PECR.0	R/W	PORTE input/output control register
$1D	-	-	-	-	-	Reserved
$1E	- WDT	WDT.2 -	WDT.1 -	WDT.0 -	R/W R	Bit2-0: Watchdog timer control register Bit3: Watchdog timer overflow flag register
$1F	-	-	-	-	-	Reserved
$20	PWM0S	T0CK1	T0CK0	PWM0_EN	R/W	Bit0: PWM0 output enable Bit2, Bit1: Select PWM0 clock Bit3: Set PWM0 output mode of duty cycle
$21	REV0	REV0	REV0	REV0	R/W	Bit0-3: Always keep it to “0” in the user’s program Refer to I/O notice
$22	PP0.3	PP0.2	PP0.1	PP0.0	R/W	PWM0 period low nibble register
$23	PP0.7	PP0.6	PP0.5	PP0.4	R/W	PWM0 period high nibble register
$24	-	-	PDF0.1	PDF0.0	R/W	PWM0 duty fine tune nibble register
$25	PD0.3	PD0.2	PD0.1	PD0.0	R/W	PWM0 duty low nibble register
$26	PD0.7	PD0.6	PD0.5	PD0.4	R/W	PWM0 duty high nibble register
$27 - 2C	-	-	-	-	-	Reserved
$2D	-	-	A1	A0	R	ADC data low nibble (Read only)
$2E	A5	A4	A3	A2	R	ADC data medium nibble (Read only)
$2F	A9	A8	A7	A6	R	ADC data high nibble (Read only)
$380	RDT.3	RDT.2	RDT.1	RDT.0	R/W	ROM data table address/data register
$381	RDT.7	RDT.6	RDT.5	RDT.4	R/W	ROM data table address/data register
$382	RDT.11	RDT.10	RDT.9	RDT.8	R/W	ROM data table address/data register
$383	RDT.15	RDT.14	RDT.13	RDT.12	R/W	ROM data table address/data register
$384	REV0	REV0	REV0	PDIEN.0	R/W	PORTD.0 interrupt enable flag Bit1-3 are reserved, always keep it to “0” in the user’s program
$385	-	-	-	PDIF.0	R/W	PORTD.0 interrupt request flag
$386	PBIEN.3	REV0	REV0	REV0	R/W	PORTB.3 interrupt enable flag Bit0-2 are reserved, always keep it to “0” in the user’s program
$387	PBIF.3	-	-	-	R/W	PORTB.3 interrupt request flag
$388	-	-	-	PPACR.0	R/W	PORTA.0 pull-high control
$389	PPBCR.3	-	-	-	R/W	PORTB.3 pull-high control
$38A	-	PPCCR.2	-	-	R/W	PORTC.2 pull-high control
$38B	-	-	-	PPDCR.0	R/W	PORTD.0 pull-high control
$38C	-	-	-	PPECR.0	R/W	PORTE.0 pull-high control

3. ROM

The ROM can address 2048 X 16 bits of program area from $0000 to $07FF.

3.1. Vector Address Area ($000 to $004)

The program is sequentially executed. There is an area address $000 through $004 that is reserved for a special interrupt service routine such as starting vector address

Address	Instruction	Remarks
$000	JMP*	Jump to Reset service routine
$001	JMP*	Jump to ADC interrupt service routine
$002	JMP*	Jump to Timer0 interrupt service routine
$003	JMP*	Jump to Timer1 interrupt service routine
$004	JMP*	Jump to PORTB/D interrupt service routine

*JMP instruction can be replaced by any instruction.

3.2. ROM Data Table

System Register:

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$380	RDT.3	RDT.2	RDT.1	RDT.0	R/W	ROM data table address/data register
$381	RDT.7	RDT.6	RDT.5	RDT.4	R/W	ROM data table address/data register
$382	RDT.11	RDT.10	RDT.9	RDT.8	R/W	ROM data table address/data register
$383	RDT.15	RDT.14	RDT.13	RDT.12	R/W	ROM data table address/data register

The RDT register consists of a 11-bit write-only PC address load register (RDT.10 - RDT.0) and a 16-bit read-only ROM table data read-out register (RDT.15 - RDT.0).

To read out the ROM table data, users should write the ROM table address to RDT register first (high nibble first then low nibble), then after one instruction, the right data will put into RDT register automatically (write lowest nibble of address into register will start the data read-out action).

4. Initial State

4.1. System Register State

Address	Bit 3	Bit 2	Bit 1	Bit 0	Power-On Reset /Pin Reset /Low Voltage Reset	WDT Reset
$00	IEAD	IET0	IET1	IEP	0000	0000
$01	IRQAD	IRQT0	IRQT1	IRQP	0000	0000
$02	REV0	T0M.2	T0M.1	T0M.0	0000	uuuu
$03	REV0	T1M.2	T1M.1	T1M.0	0000	uuuu
$04	T0L.3	T0L.2	T0L.1	T0L.0	xxxx	xxxx
$05	T0H.3	T0H.2	T0H.1	T0H.0	xxxx	xxxx
$06	T1L.3	T1L.2	T1L.1	T1L.0	xxxx	xxxx
$07	T1H.3	T1H.2	T1H.1	T1H.0	xxxx	xxxx
$08	REV0	REV0	REV0	PA.0	0000	0000
$09	PB.3	REV0	REV0	REV0	0000	0000
$0A	REV0	PC.2	REV0	REV0	0000	0000
$0B	REV0	REV0	REV0	PD.0	0000	0000
$0C	-	-	PE.1	PE.0	--10	--10
$0D	-	-	-	-	----	----
$0E	TBR.3	TBR.2	TBR.1	TBR.0	xxxx	uuuu
$0F	INX.3	INX.2	INX.1	INX.0	xxxx	uuuu
$10	DPL.3	DPL.2	DPL.1	DPL.0	xxxx	uuuu
$11	-	DPM.2	DPM.1	DPM.0	-xxx	-uuu
$12	-	DPH.2	DPH.1	DPH.0	-xxx	-uuu
$13	T1GO	REV0	-	REV0	00-0	0u-u
$14	REV0	-	-	ADCON	0--0	u--0
$15	GO/$\overline{\text{DONE}}$	TADC1	TADC0	ADCS	0000	0uuu
$16	ACR3	ACR2	ACR1	ACR0	0000	uuuu
$17	CH3	CH2	CH1	CH0	0000	uuuu
$18	REV1	REV1	REV1	PACR.0	0000	0000
$19	PBCR.3	REV1	REV1	REV1	0000	0000
$1A	REV1	PCCR.2	REV1	REV1	0000	0000
$1B	REV1	REV1	REV1	PDCR.0	0000	0000
$1C	-	-	-	PECR.0	---0	---0
$1D	-	-	-	-	----	----
$1E	WDT	WDT.2	WDT.1	WDT.0	0000	1000
$1F	-	-	-	-	----	----
$20	PWM0S	T0CK1	T0CK0	PWM0_EN	0000	uuu0

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'.

System Register State: (continued)

Address	Bit 3	Bit 2	Bit 1	Bit 0	Power-On Reset /Pin Reset /Low Voltage Reset	WDT Reset
$21	REV0	REV0	REV0	REV0	0000	uuu0
$22	PP0.3	PP0.2	PP0.1	PP0.0	xxxx	uuuu
$23	PP0.7	PP0.6	PP0.5	PP0.4	xxxx	uuuu
$24	-	-	PDF0.1	PDF0.0	--xx	--uu
$25	PD0.3	PD0.2	PD0.1	PD0.0	xxxx	uuuu
$26	PD0.7	PD0.6	PD0.5	PD0.4	xxxx	uuuu
$27 - 2C	-	-	-	-	----	----
$2D	-	-	A1	A0	--xx	--uu
$2E	A5	A4	A3	A2	xxxx	uuuu
$2F	A9	A8	A7	A6	xxxx	uuuu
$380	RDT.3	RDT.2	RDT.1	RDT.0	xxxx	uuuu
$381	RDT.7	RDT.6	RDT.5	RDT.4	xxxx	uuuu
$382	RDT.11	RDT.10	RDT.9	RDT.8	xxxx	uuuu
$383	RDT.15	RDT.14	RDT.13	RDT.12	xxxx	uuuu
$384	REV0	REV0	REV0	PDIEN.0	0000	0000
$385	-	-	-	PDIF.0	---0	---0
$386	PBIEN.3	REV0	REV0	REV0	0000	0000
$387	PBIF.3	-	-	-	0---	0---
$388	-	-	-	PPACR.0	---0	---0
$389	PPBCR.3	-	-	-	0---	0---
$38A	-	PPCCR.2	-	-	-0--	-0--
$38B	-	-	-	PPDCR.0	---0	---0
$38C	-	-	-	PPECR.0	---0	---0

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'.

4.2. Others Initial States

Others	After any Reset
Program Counter (PC)	$000
CY	Undefined
Accumulator (AC)	Undefined
Data Memory	Undefined

5. System Clock and Oscillator

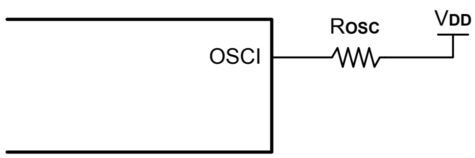
The oscillator generates the basic clock pulses that provide the system clock to supply CPU and on-chip peripherals.
System clock = fosc/4

5.1. Instruction Cycle Time

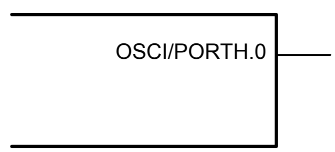
(1)4/4MHz (= 1.0μs) for 4MHz oscillator.
(2)4/10MHz (= 0.4μs) for 10MHz oscillator.

5.2 Oscillator Type

(1) RC Oscillator: 400kHz - 10MHz

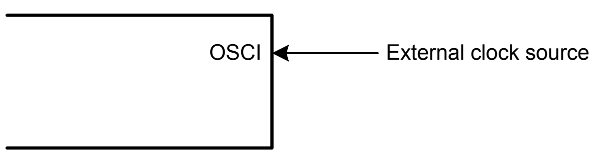


External Rosc RC



Internal Rosc RC (fosc = 4MHz)

(2) External input clock: 30kHz - 10MHz



Note:
- If the internal RC oscillator is selected, OSCI pin is used as the PORTE.0.

6. I/O Ports

The MCU provides 5 bi-directional I/O ports plus 1 CMOS open drain output port (PORTE.1). The PORT data put in register ($08 - $0C). The PORT control register ($18 - $1C) controls the PORT as input or output. Each I/O port (excluding those open drain output ports) contains a pull-high resistor, which is controlled by the value of the corresponding bit in the port pull-high control register ($388 - $38C), independently.

- When the port is selected as an input port, write “1” to the relevant bit in the port pull-high control register ($388 - $38C), could turn on the pull-high resistor and write “0” could turn off the pull-high resistor.
- When the port is selected as an output port, the pull-high resistor will be turned off automatically, regardless the value of the corresponding bit in the port pull-high control register ($388 - $38C).
- When PORTB/D are selected as the digital input direction, they can active port interrupt by falling edge (if port interrupt is enabled).

System Register $08 - $0C: Port Data Register

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$08	REV0	REV0	REV0	PA.0	R/W	PORTA
$09	PB.3	REV0	REV0	REV0	R/W	PORTB
$0A	REV0	PC.2	REV0	REV0	R/W	PORTC
$0B	REV0	REV0	REV0	PD.0	R/W	PORTD
$0C	-	-	PE.1	PE.0	R/W	PORTE

System Register $21:

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$21	REV0	REV0	REV0	REV0	R/W	Bit0-3: Always keep it to “0” in the user’s program

Note: All the REV0 bits are reserved, always keep them to “0” in the user’s program.

System Register $18 - $1C: Port Control Register

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$18	REV1	REV1	REV1	PACR.0	R/W	PORTA input/output control
$19	PBCR.3	REV1	REV1	REV1	R/W	PORTB input/output control
$1A	REV1	PCCR.2	REV1	REV1	R/W	PORTC input/output control
$1B	REV1	REV1	REV1	PDCR.0	R/W	PORTD input/output control
$1C	-	-	-	PECR.0	R/W	PORTE input/output control

PA (/B/C/D/E) CR.n, (n = 0, 1, 2, 3)
0: Set I/O as an input direction. (Default)
1: Set I/O as an output direction.

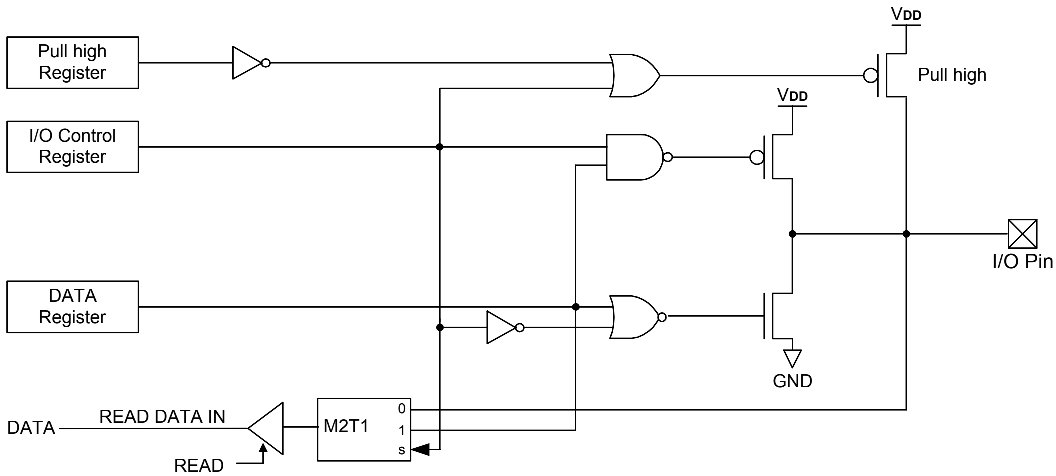
Note: All the REV1 bits are reserved, always keep them to “1” in the user’s program.

System Register $388 - $38C: Port Pull-high Control Register

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$388	-	-	-	PPACR.0	R/W	PORTA pull-high control
$389	PPBCR.3	-	-	-	R/W	PORTB pull-high control
$38A	-	PPCCR.2	-	-	R/W	PORTC pull-high control
$38B	-	-	-	PPDCR.0	R/W	PORTD pull-high control
$38C	-	-	-	PPECR.0	R/W	PORTE pull-high control

PPA (/B/C/D/E) CR.n, (n = 0, 1, 2, 3)
0: Disable internal pull-high resistor. (Default)
1: Enable internal pull-high resistor.

Equivalent Circuit for a Single I/O Pin



- In SH69P848A, each output port contains a latch, which can hold the output data. Writing the port data register (PDR) under the output mode can directly transfer data to the corresponding pad.
  All input ports do not have latches, so the external input data should be held externally until the input data is read from outside or reading the port data register (PDR) under the input mode.
  When a digital I/O port is selected to be an output, the reading of the associated port bit actually represents the value of the output data latch, not the voltage on the pad. When a digital I/O port is selected to be an input, the reading of the associated port bit represents the status on the corresponding pad.
- PORTA.0 can be shared with ADC input channel (AN0).
- PORTB.3 can be shared with ADC input channel (AN7).
- PORTD.0 can be shared with ADC input channel (AN8).
- The OSCI pin can be shared with PORTE.0, if the SH69P848A uses the internal RC oscillator as the system oscillation. (Refer to the Code option (OP_OSC [2:0]).)
- The RESET pin can be shared with PORTE.1 for open drain output port. (Refer to the Code option (OP_RST).)

Port Interrupt

The PORTB and PORTD are used as external port interrupt sources. Since PORTB and PORTD are bit programmable I/Os, only the voltage transition from V_{DD} to GND applying to the digital input port can generate a port interrupt. The analog input cannot generate any interrupt request.

The interrupt control registers are mapped on $385 - $387 of the system register. They can be accessed or tested by the read/write operation. Those flags are clear to “0” at the initialization by the chip reset.

Port Interrupts (including other external interrupt sources) can be used to wake up the CPU from the HALT or the STOP mode.

System Register $384, $386: Port Interrupt Enable Flags Register

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$384	REV0	REV0	REV0	PDIEN.0	R/W	PORTD.0 interrupt enable flag Bit1-3 are reserved, always keep it to “0” in the user’s program
$386	PBIEN.3	REV0	REV0	REV0	R/W	PORTB.3 interrupt enable flag Bit0-2 are reserved, always keep it to “0” in the user’s program

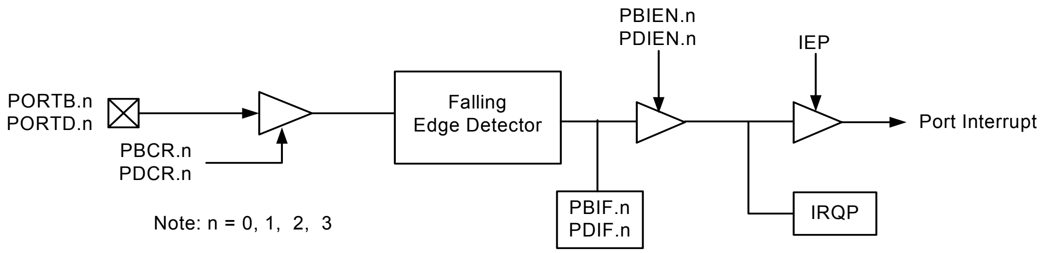
PDIEN.n, PBIEN.n (n = 0, 1, 2, 3)
0: Disable port interrupt. (Default)
1: Enable port interrupt.

System register $385, $387: Port Interrupt Request Flags Register

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$385	-	-	-	PDIF.0	R/W	PORTD.0 interrupt enable flag
$387	PBIF.3	-	-	-	R/W	PORTB.3 interrupt enable flag

PDIF.n, PBIF.n (n = 0, 1, 2, 3)
0: Port interrupt is not presented. (Default)
1: Port interrupt is presented.
Only writing these bits to 0 is available.

Following is the port interrupt function block-diagram for reference.



Port Interrupt function block-diagram

Port Interrupt Programming Notes:

When the Port falling edge is active, any one of I/O input pin transitions from V_{DD} to GND would set PIF.n to "1". Together, if the PIEN.n = 1, the port would generate an interrupt request (IRQP = 1).

Port Interrupt can wake the CPU from the HALT or STOP mode.

ADC Converter Enable Register $14:

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$14	REV0	-	-	ADCON	R/W	Bit0: ADC module on/off register Bit3 is reserved, always keep it to "0" in the user's program
	X	-	-	0	R/W	ADC converter module not operating
	X	-	-	1	R/W	ADC converter module operating

When ADC converter is disabled, PORTA.0, PORTB.3 and PORTD.0 are used as normal I/O port. When ADC converter is enabled, set ADC port configuration register ($16) to select anyone of PORTA.0, PORTB.3 and PORTD.0 as normal I/O port or ADC port. For detail, refer to ADC description.

PORTC.2 can be shared with the PWM0 output (PWM0).

PWM0 Control Register $20:

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$20	PWM0S	T0CK1	T0CK0	PWM0_EN	R/W	Bit0: PWM0 output enable Bit2, Bit1: Select PWM0 clock Bit3: Set PWM0 output mode of duty cycle
	X	X	X	0	R/W	Set PORTC.2 as normal I/O port and disable PWM0 (Default)
	X	X	X	1	R/W	Set PORTC.2 as PWM0 output and enable PWM0

7. Timer

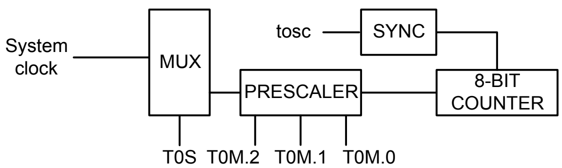
7.1. Timer0

SH69P848A has two 8-bit timers.

The Timer0 has the following features:

- 8-bit up-counting timer/counter.
- Automatic re-loads counter.
- 8-level pre-scale.
- Interrupt on overflow from $FF to $00.

The following is a simplified Timer0 block diagram.



The Timer0 provides the following functions:

- Programmable interval timer function.
- Read counter value.

7.1.1. Timer0 Configuration and Operation

The Timer0 consist of an 8-bit write-only timer load register (TL0L, TL0H) and an 8-bit read-only timer counter (TC0L, TC0H). Each of them has low order digits and high order digits. Writing data into the timer load register (TL0L, TL0H) can initialize the timer counter.

The low-order digit should be written first, and then the high-order digit. The timer counter is automatically loaded with the contents of the load register when the high order digit is written or counter counts overflow from $FF to $00.

Timer Load Register: Since the register H controls the physical READ and WRITE operations.

Please follow these steps:

Write Operation:

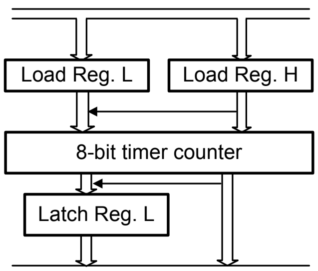
Low nibble first

High nibble to update the counter

Read Operation:

High nibble first

Low nibble followed.



7.1.2. Timer0 Mode Register

The Timer0 can be programmed in several different prescalers by setting Timer0 Mode register (TM0).

The 8-bit counter prescaler overflow output pulses. The Timer0 Mode register is 4-bit registers used for the timer control as shown in bellow. These mode registers select the input pulse sources into the timer.

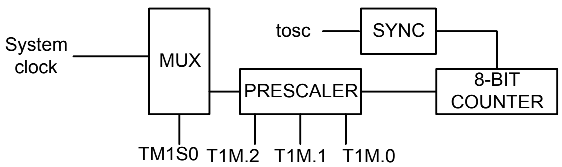
Timer0 Mode Register: $02

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$02	REV0	T0M.2	T0M.1	T0M.0	R/W	Bit0-2: Timer0 mode register Bit3 is reserved, always keep it to “0” in the user’s program
	X	0	0	0	R/W	Timer0 clock: System clock/2^{11} (Default)
	X	0	0	1	R/W	Timer0 clock: System clock/2^{9}
	X	0	1	0	R/W	Timer0 clock: System clock/2^{7}
	X	0	1	1	R/W	Timer0 clock: System clock/2^{5}
	X	1	0	0	R/W	Timer0 clock: System clock/2^{3}
	X	1	0	1	R/W	Timer0 clock: System clock/2^{2}
	X	1	1	0	R/W	Timer0 clock: System clock/2^{1}
	X	1	1	1	R/W	Timer0 clock: System clock/2^{0}

7.2. Timer1

The Timer1 has the following features:
- 8-bit up-counting timer/counter.
- Automatic re-loads counter.
- 8-level pre-scale.
- Interrupt on overflow from $FF to $00.

The following is a simplified Timer1 block diagram.



The Timer1 provides the following functions:
- Programmable interval timer function.
- Read counter value.

7.2.1. Timer1 Configuration and Operation

Timer1 consists of a 8-bit write-only timer load register (TL1L, TL1H) and a 8-bit read-only timer counter (TC1L, TC1H). Each of them has low order digits and high order digits. Writing data into the timer load register (TL1L, TL1H) can initialize the timer counter.

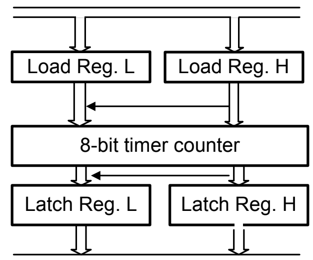
The low-order digit should be written first, and then the high-order digit. The timer counter is automatically loaded with the contents of the load register when the high order digit is written or the counter counts overflow from $FF to $00.

Timer Load Register: Since the register H controls the physical READ and WRITE operations.

Please follow these steps:

Write Operation:
  Low nibble first
  High nibble to update the counter

Read Operation:
  High Nibble first
  Low nibble followed.



7.2.2. Timer1 Control Register

The counting of Timer1 is controlled by TM1S0 (bit0) and T1GO (bit3) in the Timer1 control register ($13). To start the Timer1 counter, the TM1S0 should be cleared to 0 and the T1GO should be set to 1.

After the T1GO has been set to 1, writing the Timer1 counter register ($06 - $07) can't affect the up-counter operating anymore. Only when the T1GO has been cleared to 0, the revised contents of the Timer1 counter register will be loaded into the up-counter while the highest nibble ($07) is written.

Timer1 Control Register: $13

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$13	T1GO	REV0	-	REV0	R/W	Bit3: Timer1 Start/stop register Bit0 and bit3 are reserved, always keep it to “0” in the user’s program
	0	X	-	X	R/W	Timer/counter stops (Read: status; Write: command) (Default)
	1	X	-	X	R/W	Timer/counter starts (Read: status; Write: command)

The Timer1 can be programmed in several different prescalers by setting Timer1 Mode register (TM1).

The 8-bit counter prescaler overflow output pulses. The Timer0 Mode register is 3-bit registers used for the timer control as shown in bellow. These mode registers select the input pulse sources into the timer.

Timer1 Mode Register: $03

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$03	REV0	T1M.2	T1M.1	T1M.0	R/W	Timer1 mode register Bit3 is reserved, always keep it to “0” in the user’s program
	X	0	0	0	R/W	Timer1 clock: System clock/2^{11} (Default)
	X	0	0	1	R/W	Timer1 clock: System clock/2^{9}
	X	0	1	0	R/W	Timer1 clock: System clock/2^{7}
	X	0	1	1	R/W	Timer1 clock: System clock/2^{5}
	X	1	0	0	R/W	Timer1 clock: System clock/2^{3}
	X	1	0	1	R/W	Timer1 clock: System clock/2^{2}
	X	1	1	0	R/W	Timer1 clock: System clock/2^{1}
	X	1	1	1	R/W	Timer1 clock: System clock/2^{0}

Timer1 Counter Register: $06 - $07

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$06	T1L.3	T1L.2	T1L.1	T1L.0	R/W	Timer1 load/counter register (low nibble)
$07	T1H.3	T1H.2	T1H.2	T1H.0	R/W	Timer1 load/counter register (high nibble)

8. Interrupt

Four interrupt sources are available on SH69P848A:
- ADC interrupt
- Timer0 interrupt
- Timer1 interrupt
- PORTB/D interrupts (Falling edge)

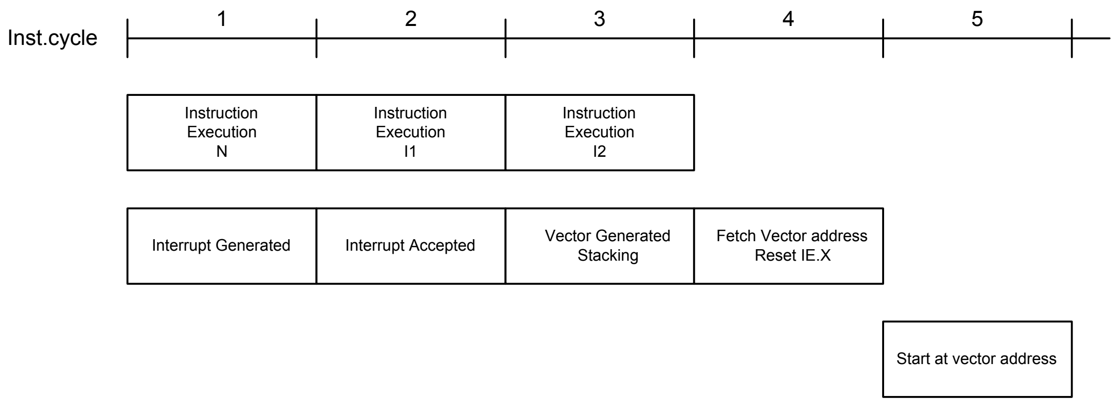
Interrupt Control Bits and Interrupt Service

The interrupt control flags are mapped on $00 and $01 of the system register. They can be accessed or tested by the program. Those flags are clear to “0” at initialization by the chip reset.

System Register:

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$00	IEAD	IET0	IET1	IEP	R/W	Interrupt enable flags
$01	IRQAD	IRQT0	IRQT1	IRQP	R/W	Interrupt request flags

When IEx is set to “1” and the interrupt request is generated (IRQx is 1), the interrupt will be activated and vector address will be generated from the priority PLA corresponding to the interrupt sources. When an interrupt occurs, the PC and CY flag will be saved into stack memory and jump to interrupt service vector address. After the interrupt occurs, all interrupt enable flags (IEx) are clear to “0” automatically, so when IRQx is 1 and IEx is set to “1” again, the interrupt will be activated and vector address will be generated from the priority PLA corresponding to the interrupt sources.



Interrupt Servicing Sequence Diagram

Interrupt Nesting

During the CPU interrupt service, the user can enable any interrupt enable flag before returning from the interrupt. The servicing sequence diagram shows the next interrupt and the next nesting interrupt occurrences. If the interrupt request is ready and the instruction of execution N is IE enabled, then the interrupt will start immediately after the next two instruction executions. However, if instruction I1 or instruction I2 disables the interrupt request or enable flag, then the interrupt service will be terminated.

ADC Interrupt

Bit3 (IEAD) of system register $00 is the ADC interrupt enable flag. When the ADC conversion is complete, it will generate an interrupt request (IRQAD = 1), if the ADC interrupt is enabled (IEAD = 1), an ADC interrupt service routine will start. The ADC interrupt can be used to wake the CPU from HALT mode.

Timer (Timer0, Timer1) Interrupt

The input clock of Timer0 and Timer1 are based on the system clock. The timer overflow from $FF to $00 will generate an internal interrupt request (IRQT0 or IRQT1 = 1), If the interrupt enable flag is enabled (IET0 or IET1 = 1), a timer interrupt service routine will start. Timer interrupt can also be used to wake the CPU from HALT mode.

Port Falling Edge Interrupt

Only the digital input port can generate a port interrupt. The analog input cannot generate an interrupt request.
Port Interrupt can be used to wake the CPU from HALT or STOP mode.

Port Interrupts by Bit

Only the digital input port can generate a port interrupt. The analog input cannot generate an interrupt request.

System Register $384, $386: Port Interrupt Enable Flags Register

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$384	REV0	REV0	REV0	PDIEN.0	R/W	PORTD.0 interrupt enable flag Bit1-3 are reserved, always keep it to “0” in the user’s program
$386	PBIEN.3	REV0	REV0	REV0	R/W	PORTB.3 interrupt enable flag Bit0-2 are reserved, always keep it to “0” in the user’s program

System Register $385, $387: Port Interrupt Request Flags Register

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$385	-	-	-	PDIF.0	R/W	PORTD.0 interrupt request flag
$387	PBIF.3	-	-	-	R/W	PORTB.3 interrupt request flag

9. Analog/Digital Converter (ADC)

The 3 channels and 10-bit resolution ADC converter are implemented in this micro-controller.

The ADC converter control registers can be used to define the ADC channel number, select analog channel and conversion clock, start ADC conversion, and set the end of ADC conversion flag. The ADC conversion result register byte is read-only.

The approach for ADC conversion:

- Set analog channel. (Keep in mind that any analog input voltage must not exceed V_{DD})
- Operating ADC converter module and select the converted analog channel.
- Set ADC conversion clock source.
- GO/$\overline{DONE}$ = 1, start ADC conversion.

Systems Register $14

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$14	REV0	-	-	ADCON	R/W	Bit0: ADC module on/off register Bit3 is reserved, always keep it to “0” in the user’s program
	X	-	-	0	R/W	ADC converter module not operating (Default)
	X	-	-	1	R/W	ADC converter module operating

Systems Register $16

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$16	ACR3	ACR2	ACR1	ACR0	R/W	ADC port configuration control register
	0	0	0	0	R/W	See the following table (Default)

Set Analog Channels

ACR3	ACR2	ACR1	ACR0	8	7	0
0	0	0	0	PD.0	PB3	PA.0
0	0	0	1	PD.0	PB.3	AN0
0	0	1	0	PD.0	PB.3	AN0
0	0	1	1	PD.0	PB.3	AN0
0	1	0	0	PD.0	PB.3	AN0
0	1	0	1	PD.0	PB.3	AN0
0	1	1	0	PD.0	PB.3	AN0
0	1	1	1	PD.0	PB.3	AN0
1	X	0	0	PD.0	AN7	AN0
1	X	0	1	AN8	AN7	AN0
1	X	1	X	AN8	AN7	AN0

System Register $17 for ADC Channel Selection

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$17	CH3	CH2	CH1	CH0	R/W	ADC channel control register
	0	0	0	0	R/W	ADC channel AN0 (Default)
	0	1	1	1	R/W	ADC channel AN7
	1	X	X	0	R/W	ADC channel AN8

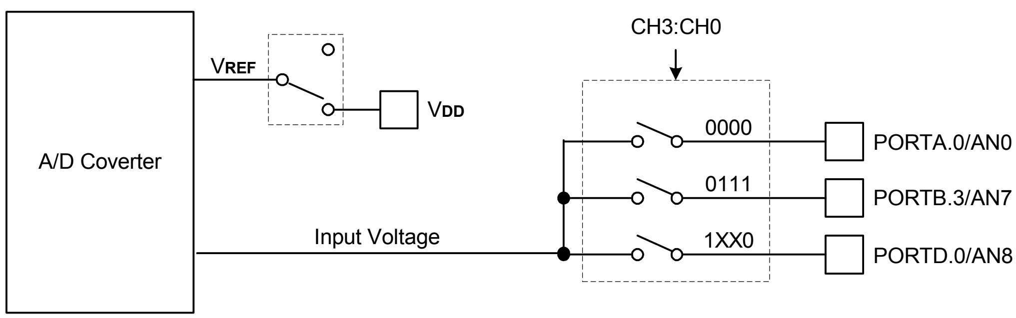
Note: The value not included in the above table should not be written to Bit0 - 3 of the system register $17.

System Register $2D - $2F for ADC Data

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$2D	X	X	A1	A0	R	ADC data low nibble (Read only)
$2E	A5	A4	A3	A2	R	ADC data medium nibble (Read only)
$2F	A9	A8	A7	A6	R	ADC data high nibble (Read only)

Systems Register $15

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$15	$GO/\overline{DONE}$	TADC1	TADC0	ADCS	R/W	Bit0: ADC conversion time select register Bit1-2: ADC clock period select register Bit3: ADC status flag register
	X	X	X	0	R/W	ADC conversion time = 204t_{AD} (Default)
	X	X	X	1	R/W	ADC conversion time = 780t_{AD}
	X	0	0	X	R/W	ADC clock period t_{AD} = t_{OSC} (Default)
	X	0	1	X	R/W	ADC clock period t_{AD} = 4t_{OSC}
	X	1	0	X	R/W	ADC clock period t_{AD} = 8t_{OSC}
	X	1	1	X	R/W	ADC clock period t_{AD} = 16t_{OSC}
	0	X	X	X	R/W	ADC conversion not in progress (Default)
	1	X	X	X	R/W	ADC conversion in progress, when ADCON = 1



ADC Converter Block Diagram

Notes:

- Select ADC clock period t_{AD}, make sure that $1\mu s \le t_{AD} \le 33.4\mu s$.
- When the ADC conversion is complete, an ADC converter interrupt occurs (if the ADC converter interrupt is enabled).
- The analog input channels must have their corresponding PXCR (X = A, B, D) bits selected as inputs.
- If select I/O port as analog input, the I/O functions and pull-high resistor are disabled.
- Bit $GO/\overline{DONE}$ is automatically cleared by hardware when the ADC conversion is complete.
- Clearing the $GO/\overline{DONE}$ bit during a conversion will abort the current conversion.
- The ADC result register will NOT be updated with the partially completed ADC conversion sample.
- 4t_{OSC} wait is required before the next acquisition is started.
- ADC converter could keep on working in HALT mode, and would stop automatic while executing a “STOP” instruction.
- ADC converter could wake-up the device from HALT mode (if the ADC converter interrupt is enabled).

10. Pulse Width Modulation (PWM)

The SH69P848A consists of one 8+2 PWM modules. The PWM module can provide the pulse width modulation waveform with the period and the duty being controlled, individually. The PWMC is used to control the PWM module operation with proper clocks. The PWMP is used to control the period cycle of the PWM module output. And the PWMD is used to control the duty in the waveform of the PWM module output.

System Register $20: PWM Control Register (PWMC)

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$20	PWM0S	T0CK1	T0CK0	PWM0_EN	R/W	Bit0: PWM0 output enable Bit2, Bit1: Select PWM0 clock Bit3: Set PWM0 output mode of duty cycle
	X	X	X	0	R/W	Shared with I/O port (Default)
	X	X	X	1	R/W	Shared with PWM0
	X	0	0	X	R/W	PWM0 clock = tosc (Default)
	X	0	1	X	R/W	PWM0 clock = 2tosc
	X	1	0	X	R/W	PWM0 clock = 4tosc
	X	1	1	X	R/W	PWM0 clock = 8tosc
	0	X	X	X	R/W	PWM0 output normal mode of duty cycle (high active) (Default)
	1	X	X	X	R/W	PWM0 output negative mode of duty cycle (low active)

The PWM0 output pin is shared with PORTC.2.

System Register $22 - $23: PWM Period Control Register (PWMP)

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$22	PP0.3	PP0.2	PP0.1	PP0.0	R/W	PWM0 period low nibble
$23	PP0.7	PP0.6	PP0.5	PP0.4	R/W	PWM0 period high nibble

PWM output period cycle = [PP0.7, PP0.0] X PWM0 clock.
When [PP0.7, PP0.0] = 00H, PWM0 outputs GND if the PWM0S bit is set to “0”.
When [PP0.7, PP0.0] = 00H, PWM0 outputs high level if the PWM0S bit is set to “1”.

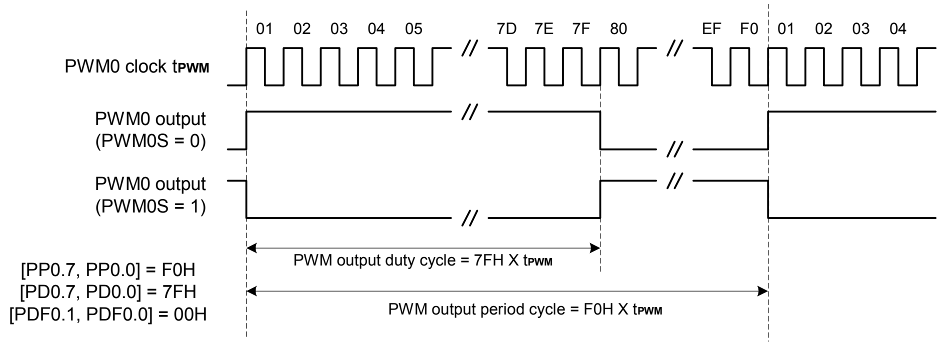
System Register $24 - $26: PWM Duty Control Register (PWMD)

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$24	-	-	PDF0.1	PDF0.0	R/W	PWM0 duty fine tune nibble
$25	PD0.3	PD0.2	PD0.1	PD0.0	R/W	PWM0 duty low nibble
$26	PD0.7	PD0.6	PD0.5	PD0.4	R/W	PWM0 duty high nibble

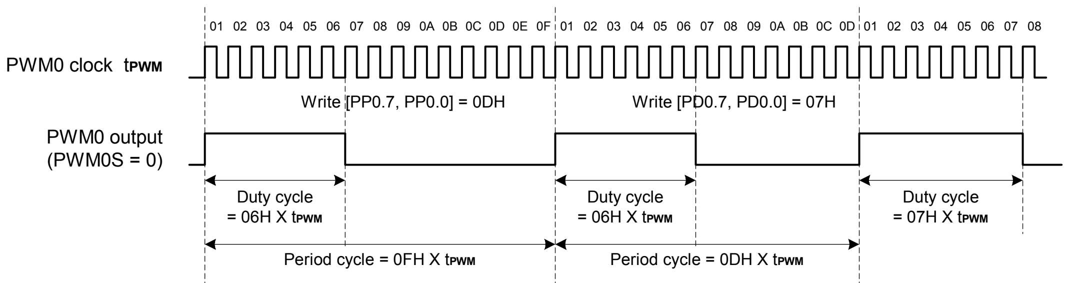
Average PWM0 output duty cycle = ([PD0.7, PD0.0] + [PDF0.1, PDF0.0]/4) X PWM0 clock.
If [PP0.7, PP0.0] $\leq$ [PD0.7, PD0.0], PWM0 outputs high when the PWM0S bit is set to “0”.
If [PP0.7, PP0.0] $\leq$ [PD0.7, PD0.0], PWM0 outputs GND level when the PWM0S bit is set to “1”.

Note:

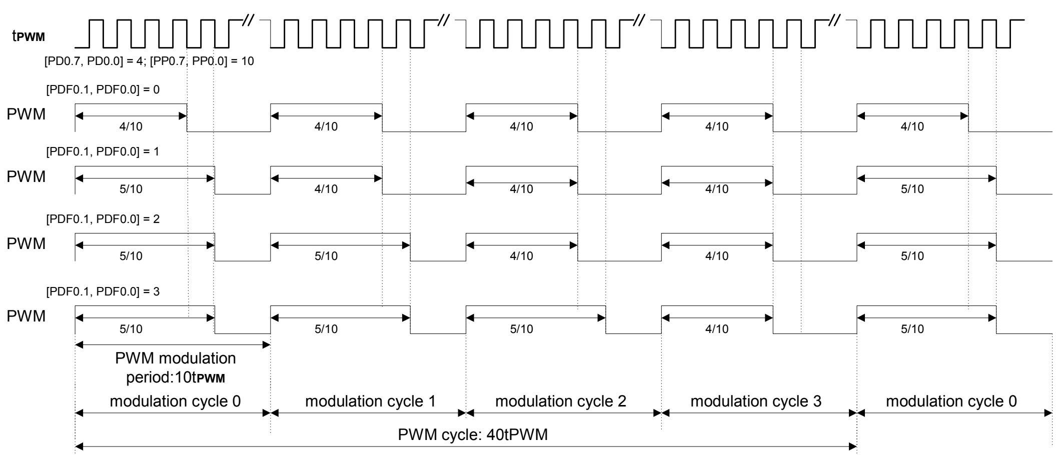
- If the I/O port is selected as the PWM output, the I/O functions and pull-high resistor are disabled.
- The writing flow of the PWM0 duty control register is described as follows. First set the fine tune nibble, then the low nibble and set the high nibble at last.
- The writing flow of the PWM0 period control register is described as follows. First set the low nibble, then set the high nibble.
- After the high nibble of the PWM0 period or duty control register is written, the data are loaded into the re-load counter and start counting at next period.
- The reading flow of the PWM0 period or duty control register is at the reverse direction with that described above. First read the high nibble, then read the low nibble.
- PWM could keep on working in the HALT mode, and would stop automatic when the "STOP" instruction is executed.



PWM Output Example



PWM Output Period or Duty Cycle Changing Example



(8+2) bits PWM Waveform

In the (8+2) bits PWM waveform, A PWM cycle is divided into 4 modulation cycles (cycle 0 - cycle 3), each modulation cycle has certain period decided by period cycle registers (PWMP). The contents of duty cycle register (PWMD) is divided into two parts. The basic part of PWMD is PD0.7 - PD0.0. The extended part is PDF0.1 - PDF0.0. In a PWM cycle, the duty cycle of each modulation cycle is shown in the table.

Parameter	[PDF0.1, PDF0.0] (0-3)	Duty Cycle
Modulation Cycle I (I = 0-3)	I< [PDF0.1, PDF0.0]	([PD0.7, PD0.0] + 1)/[PP0.7, PP0.0]
	I≥ [PDF0.1, PDF0.0]	[PD0.7, PD0.0]/[PP0.7, PP0.0]

The modulation period, cycle period and cycle duty of the PWM output signal are summarized in the following table.

PWM modulation period	PWM cycle period	PWM cycle duty
[PP0.7, PP0.0] * **tpwm**	4*[PP0.7, PP0.0] * **tpwm**	(4* [PD0.7, PD0.0] + [PDF0.1, PDF0.0])/(4*[PP0.7, PP0.0])

11. Low Voltage Reset (LVR)

The LVR function is to monitor the supply voltage and generate an internal reset in the device. It is typically used in AC line applications or large battery where heavy loads may be switched in and cause the device voltage to temporarily fall below the specified operating minimum.

The LVR function is selected by the Code option.

The LVR circuit has the following functions when the LVR function is enabled:

- Generates a system reset when $V_{DD} \leq V_{LVR}$.
- Cancels the system reset when $V_{DD} > V_{LVR}$.

12. Watchdog Timer (WDT)

The watchdog timer is a count-down counter, and its clock source is an independent built-in RC oscillator, so that it will always run even in the STOP mode. The watchdog timer automatically generates a device reset when it overflows. It can be enabled or disabled permanently by using the code option.

The watchdog timer control bits ($1E bit2 - bit0) are used to select different overflow frequency. The watchdog timer overflow flag ($1E bit3) will be automatically set to "1" by hardware when the watchdog timer overflows. By reading or writing the system register $1E, the watchdog timer should re-count before the overflow happens.

System Register $1E: Watchdog Timer (WDT)

Address	Bit 3	Bit 2	Bit 1	Bit 0	R/W	Remarks
$1E	- WDT	WDT.2 -	WDT.1 -	WDT.0 -	R/W R	Bit2-0: Watchdog timer control register Bit3: Watchdog timer overflow flag register
	X	0	0	0	R/W	Watchdog timer-out period = 4096ms (Default)
	X	0	0	1	R/W	Watchdog timer-out period = 1024ms
	X	0	1	0	R/W	Watchdog timer-out period = 256ms
	X	0	1	1	R/W	Watchdog timer-out period = 128ms
	X	1	0	0	R/W	Watchdog timer-out period = 64ms
	X	1	0	1	R/W	Watchdog timer-out period = 16ms
	X	1	1	0	R/W	Watchdog timer-out period = 4ms
	X	1	1	1	R/W	Watchdog timer-out period = 1ms
	0	X	X	X	R	No watchdog timer overflow reset (Default)
	1	X	X	X	R	Watchdog timer overflow, WDT reset happens

Note:

Watchdog timer-out period valid for V_{DD} = 5V.

13. HALT and STOP Mode

After the execution of HALT instruction, SH69P848A will enter the HALT mode. In the HALT mode, CPU will STOP operating. But peripheral circuit (Timer0, Timer1, ADC and watchdog timer) will keep status.

After the execution of STOP instruction, SH69P848A will enter the STOP mode. The whole chip (including oscillator) will STOP operating. But watchdog is still enabled.

In the HALT mode, SH69P848A can be waked up if any interrupt occurs.

In the STOP mode, SH69P848A can be waked up if port interrupt occurs or watchdog timer overflow (WDT is enabled).

When CPU is awaked from the HALT/STOP by any interrupt source, it will execute the relevant interrupt serve subroutine at first. Then the instruction next to HALT/STOP is executed.

14. Warm-up Timer

The device has a built-in warm-up timer to eliminate unstable state of initial oscillation when oscillator starts oscillating in the following conditions:

Power-on Reset

Warm-up time interval:

(1) When oscillator range is 30kHz - 2MHz, the warm-up counter prescaler divide ratio is 2^{12} (4096)
(2) When oscillator range is 2MHz -10MHz, the warm-up counter prescaler divide ratio is 2^{14} (16384).

Wake up from Stop Mode

Warm-up time interval:

In RC oscillator or external clock mode, the warm-up counter prescaler divide ratio is 2^{7} (128).

15. Code option

Oscillator Type:

OP_OSC [2:0]:

000 = External clock (Default)
001 = Internal RC oscillator (4MHz) (Select OSCI pin as PORTE.0 for normal I/O ports)
010 = Internal RC oscillator (4MHz) (Select OSCI pin as PORTE.0 for normal I/O ports)
011 = Internal RC oscillator (4MHz) (Select OSCI pin as PORTE.0 for normal I/O ports)
100 = External RC oscillator (400kHz - 10MHz)

Oscillator Range:

OP_OSC 3:

0 = 2MHz - 10MHz (Default)
1 = 30kHz - 2MHz

Watchdog Timer:

OP_WDT:

0 = Enable (Default)
1 = Disable

Low Voltage Reset:

OP_LVR:

0 = Disable (Default)
1 = Enable

LVR Voltage Range:

OP_LVR0:

0 = High LVR voltage (Default)
1 = Low LVR voltage

Chip Pin Reset:

OP_RST:

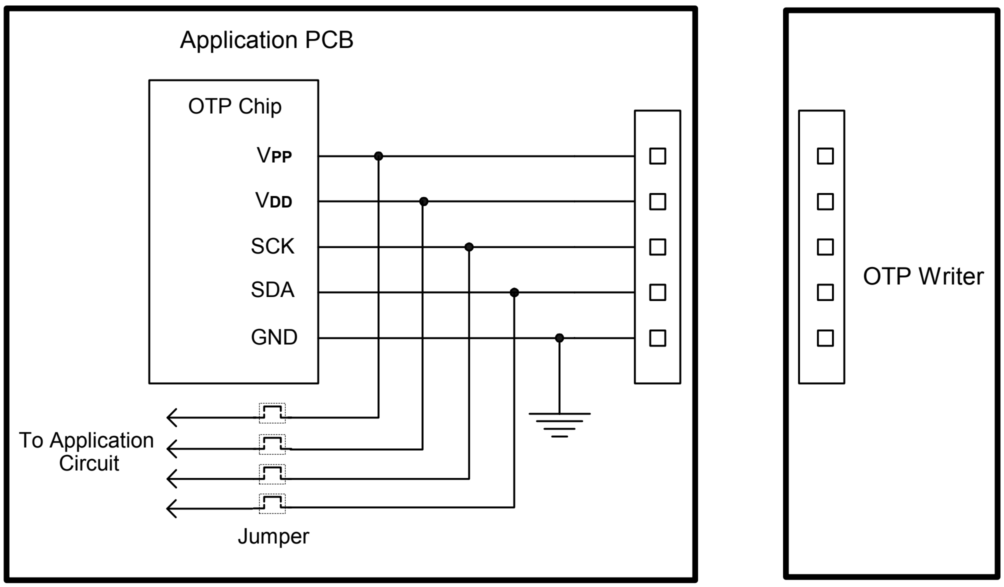
0 = Enable chip pin reset (Default)
1 = Disable chip pin reset (Select RESET pin as PORTE.1 for an open drain output)

In System Programming Notice for OTP

The In System Programming technology is valid for OTP chip.

The Programming Interface of the OTP chip must be set on user's application PCB, and users can assemble all components including the OTP chip in the application PCB before programming the OTP chip. Of course, it's accessible bonding OTP chip only first, and then programming code and finally assembling other components.

Since the programming timing of Programming Interface is very sensitive, therefore four jumpers are needed (V_{DD}, V_{PP}, SDA, SCK) to separate the programming pins from the application circuit as shown in the following diagram.



The recommended steps are the followings:

(1) The jumpers are open to separate the programming pins from the application circuit before programming the chip.
(2) Connect the programming interface with OTP writer and begin programming.
(3) Disconnect OTP writer and shorten these jumpers when programming is completed.

For more detail information, please refer to the OTP writer user manual.

Instruction Set

All instructions are one cycle and one-word instructions. The characteristic is memory-oriented operation.

1. Arithmetic and Logical Instruction

1.1. Accumulator Type

Mnemonic	Instruction Code	Function	Flag Change
ADC X (, B)	00000 0bbb xxx xxxx	AC <- Mx + AC + CY	CY
ADCM X (, B)	00000 1bbb xxx xxxx	AC, Mx <- Mx + AC + CY	CY
ADD X (, B)	00001 0bbb xxx xxxx	AC <- Mx + AC	CY
ADDM X (, B)	00001 1bbb xxx xxxx	AC, Mx <- Mx + AC	CY
SBC X (, B)	00010 0bbb xxx xxxx	AC <- Mx + -AC + CY	CY
SBCM X (, B)	00010 1bbb xxx xxxx	AC, Mx <- Mx + -AC + CY	CY
SUB X (, B)	00011 0bbb xxx xxxx	AC <- Mx + -AC +1	CY
SUBM X (, B)	00011 1bbb xxx xxxx	AC, Mx <- Mx + -AC +1	CY
EOR X (, B)	00100 0bbb xxx xxxx	AC <- Mx ⊕ AC	
EORM X (, B)	00100 1bbb xxx xxxx	AC, Mx <- Mx ⊕ AC	
OR X (, B)	00101 0bbb xxx xxxx	AC <- Mx | AC	
ORM X (, B)	00101 1bbb xxx xxxx	AC, Mx <- Mx | AC	
AND X (, B)	00110 0bbb xxx xxxx	AC <- Mx & AC	
ANDM X (, B)	00110 1bbb xxx xxxx	AC, Mx <- Mx & AC	
SHR	11110 0000 000 0000	0 -> AC[3], AC[0] -> CY; AC shift right one bit	CY

1.2. Immediate Type

Mnemonic	Instruction Code	Function	Flag Change
ADI X, I	01000 iiii xxx xxxx	AC <- Mx + I	CY
ADIM X, I	01001 iiii xxx xxxx	AC, Mx <- Mx + I	CY
SBI X, I	01010 iiii xxx xxxx	AC <- Mx + -I +1	CY
SBIM X, I	01011 iiii xxx xxxx	AC, Mx <- Mx + -I +1	CY
EORIM X, I	01100 iiii xxx xxxx	AC, Mx <- Mx ⊕ I	
ORIM X, I	01101 iiii xxx xxxx	AC, Mx <- Mx | I	
ANDIM X, I	01110 iiii xxx xxxx	AC, Mx <- Mx & I	

1.3. Decimal Adjustment

Mnemonic	Instruction Code	Function	Flag Change
DAA X	11001 0110 xxx xxxx	AC, Mx <- Decimal adjust for add	CY
DAS X	11001 1010 xxx xxxx	AC, Mx <- Decimal adjust for sub	CY

2. Transfer Instruction

Mnemonic	Instruction Code	Function	Flag Change
LDA X (, B)	00111 0bbb xxx xxxx	AC <- Mx	
STA X (, B)	00111 1bbb xxx xxxx	Mx <- AC	
LDI X, I	01111 iiii xxx xxxx	AC, Mx <- I	

3. Control Instruction

Mnemonic	Instruction Code	Function	Flag Change
BAZ X	10010 xxxx xxx xxxx	PC <- X, if AC = 0	
BNZ X	10000 xxxx xxx xxxx	PC <- X, if AC ≠ 0	
BC X	10011 xxxx xxx xxxx	PC <- X, if CY = 1	
BNC X	10001 xxxx xxx xxxx	PC <- X, if CY ≠ 1	
BA0 X	10100 xxxx xxx xxxx	PC <- X, if AC (0) = 1	
BA1 X	10101 xxxx xxx xxxx	PC <- X, if AC (1) = 1	
BA2 X	10110 xxxx xxx xxxx	PC <- X, if AC (2) = 1	
BA3 X	10111 xxxx xxx xxxx	PC <- X, if AC (3) = 1	
CALL X	11000 xxxx xxx xxxx	ST <- CY, PC +1 PC <- X (Not include p)	
RTNW H, L	11010 000h hhh llll	PC <- ST; TBR <- hhhh, AC <- lll	
RTNI	11010 1000 000 0000	CY, PC <- ST	CY
HALT	11011 0000 000 0000		
STOP	11011 1000 000 0000		
JMP X	1110p xxxx xxx xxxx	PC <- X (Include p)	
TJMP	11110 1111 111 1111	PC <- (PC11-PC8) (TBR) (AC)	
NOP	11111 1111 111 1111	No Operation	

Where,

PC	Program counter	I	Immediate data
AC	Accumulator	⊕	Logical exclusive OR
-AC	Complement of accumulator	|	Logical OR
CY	Carry flag	&	Logical AND
Mx	Data memory	bbb	RAM bank
p	ROM page		
ST	Stack	TBR	Table Branch Register

Electrical Characteristics

Absolute Maximum Ratings*

DC Supply Voltage . . . . -0.3V to +7.0V

Input/Output Voltage . . . . .GND-0.3V to V_{DD}+0.3V

Operating Ambient Temperature . . . .-40℃ to +85℃

Storage Temperature . . . . -55℃ to +125℃

***Comments**

Stresses exceed those listed under "**Absolute Maximum Ratings**" may cause permanent damage to this device. These are stress ratings only. Functional operation of this device at these or any other conditions above those indicated in the operational sections of this specification is not implied or intended. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

DC Electrical Characteristics (V_{DD} = 2.4-5.5V, GND = 0V, T_A = -40°C to +85°C, unless otherwise specified)

Parameter	Symbol	Min.	Typ. ∗	Max.	Unit	Condition
Operating Voltage	V_{DD}	4.5	5.0	5.5	V	30kHz ≤ fosc ≤ 10MHz
Operating Voltage	V_{DD}	2.4	5.0	5.5	V	30kHz ≤ fosc ≤ 4MHz
Operating Current	I_{OP}	-	3	4.5	mA	fosc = 10MHz All output pins unload, execute NOP instruction, WDT off, ADC disable, LVR off. V_{DD} = 5.0V
		-	2	3	mA	fosc = 4MHz All output pins unload, execute NOP instruction, WDT off, ADC disable, LVR off. V_{DD} = 5.0V
Stand by Current (HALT)	I_{SB1}	-	-	1.5	mA	fosc = 10MHz All output pins unload (HALT mode), WDT off, ADC disable, LVR off. V_{DD} = 5.0V
		-	-	1	mA	fosc = 4MHz All output pins unload (HALT mode), WDT off, ADC disable, LVR off. V_{DD} = 5.0V
Stand by Current (STOP)	I_{SB2}	-	-	1	μA	All output pins unload (STOP mode), WDT off, ADC disable, LVR off. V_{DD} = 5.0V
WDT Current	I_{WDT}	-	-	20	μA	All output pins unload (STOP mode), WDT on, ADC disable, LVR off, V_{DD} = 5.0V
Input Low Voltage	V_{IL1}	GND	-	0.3 X V_{DD}	V	I/O Ports
Input Low Voltage	V_{IL2}	GND	-	0.2 X V_{DD}	V	$\overline{RESET}$, OSCI
Input High Voltage	V_{IH1}	0.7 X V_{DD}	-	V_{DD}	V	I/O Ports
Input High Voltage	V_{IH2}	0.8 X V_{DD}	-	V_{DD}	V	$\overline{RESET}$, OSCI
Input Leakage Current	I_{IL}	-1	-	1	μA	Input pad, V_{IN} = V_{DD} or GND
Pull-high Resistor	R_{PH}	10	30	50	kΩ	V_{DD} = 5.0V, V_{IN} = GND
Output Leakage Current	I_{OL}	-1	-	1	μA	Open drain output, V_{DD} = 5.0V V_{OUT} = V_{DD} or GND
Output High Voltage	V_{OH}	V_{DD} - 0.7	-	-	V	I/O Ports, PWM0, I_{OH} = -10mA, V_{DD} = 5.0V
Output Low Voltage	V_{OL}	-	-	GND + 0.6	V	I/O Ports, PWM0, I_{OL} = 20mA, V_{DD} = 5.0V

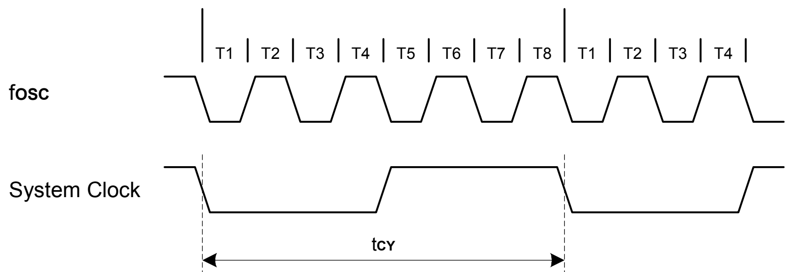
∗: Data in “Typ.” column is at 5.0V, 25℃,unless otherwise specified.
Maximum value of the supply current to V_{DD} is 100mA.
Maximum value of the output current from GND is 150mA.

AC Electrical Characteristics
(V_{DD} = 2.4V - 5.5V, GND = 0V, T_A = 25°C, f_{OSC} = 30kHz -10MHz, unless otherwise specified.)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
RESET pulse width	t_{RESET}	10	-	-	μs	V_{DD} = 5.0V
WDT Period	t_{WDT}	1	-	-	ms	V_{DD} = 5.0V
Frequency Variation	|ΔF|/F	-	-	15	%	External RC Oscillator |F(5.0V)-F(2.4V)|/F(5.0V)
Internal RC Frequency Variation	f_{OSC}	3.80	4.00	4.20	MHz	V_{DD} = 5.0V, T_A = 5°C - 45°C
Instruction cycle time	t_{CY}	0.4	-	133.4	μs	f_{OSC} = 30kHz - 10MHz

Timing Waveform

(a) System Clock Timing Waveform



ADC Converter Electrical Characteristics

(V_{DD} = 2.4V - 5.5V, GND = 0V, T_A = 25°C, f_{OSC} = 30kHz - 10MHz, unless otherwise specified.)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Resolution	NR	-	-	10	bit	GND ≤ V_{AIN} ≤ V_{REF}
Reference Voltage	V_{REF}	2.4	-	V_{DD}	V	
ADC Input Voltage	V_{AIN}	GND	-	V_{REF}	V	
ADC Input Resistor	R_{AIN}	2000	-	-	kΩ	V_{IN} = 5.0V
ADC conversion current	I_{AD}	-	500	1000	μA	ADC converter module operating, V_{DD}=5.0V
Nonlinear Error	E_{NL}	-	-	±2	LSB	V_{REF} = V_{DD} = 5.0V
Full scale error	E_F	-	-	±1	LSB	V_{REF} = V_{DD} = 5.0V
Offset error	E_Z	-	-	±1	LSB	V_{REF} = V_{DD} = 5.0V
Total Absolute error	E_{AD}	-	±1	±2	LSB	V_{REF} = V_{DD} = 5.0V
ADC Clock Period	t_{AD}	1	-	33.4	μs	f_{OSC} = 30kHz - 10MHz
ADC Conversion Time	t_{CNV1}	-	204	-	t_{AD}	Set ADCS = 0
ADC Conversion Time	t_{CNV2}	-	780	-	t_{AD}	Set ADCS = 1

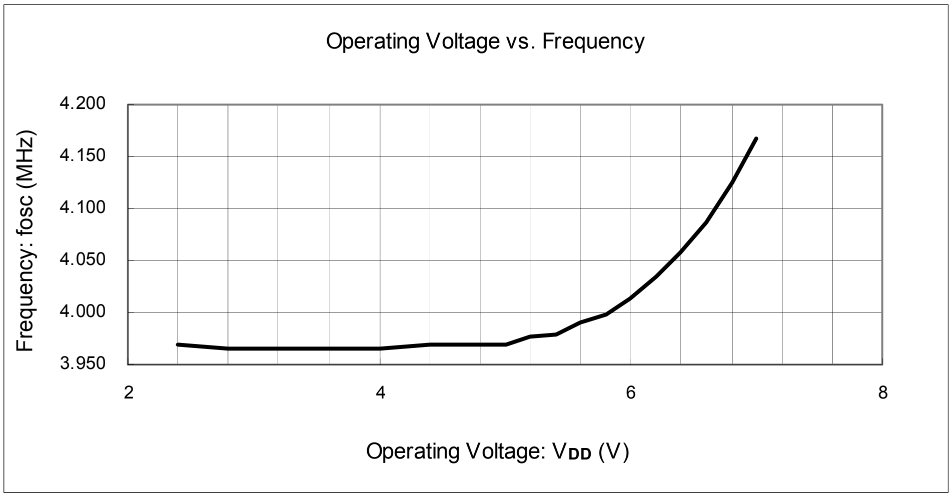
Low Voltage Reset Electrical Characteristics

(GND = 0V, T_A = 25°C, f_{OSC} = 30kHz - 10MHz, unless otherwise specified.)

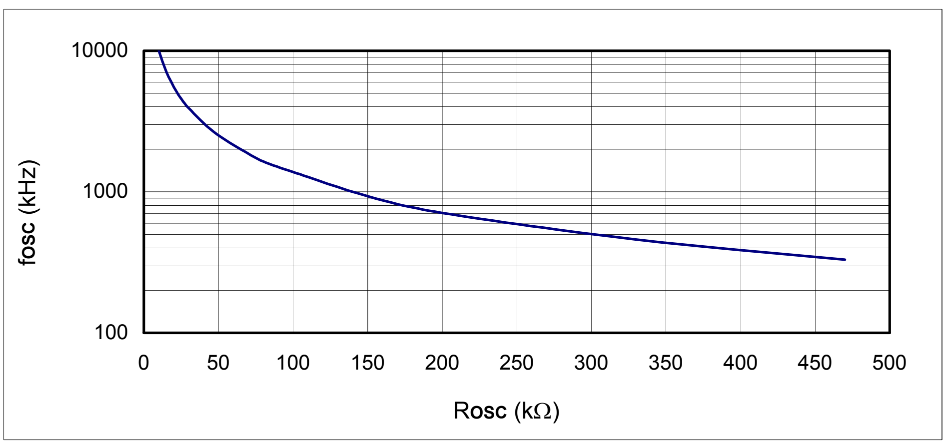
Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
LVR Voltage (1)	V_{LVR1}	3.8	-	4.2	V	LVR enable
LVR Voltage (2)	V_{LVR2}	2.3	-	2.7	V	LVR enable
LVR Voltage Pulse Width	t_{LVR}	500	-	-	μs	V_{DD} ≤ V_{LVR}

RC oscillator Characteristics Graphs (for reference only)

Internal RC Oscillator Characteristics Graphs (Operating Voltage vs. Frequency):

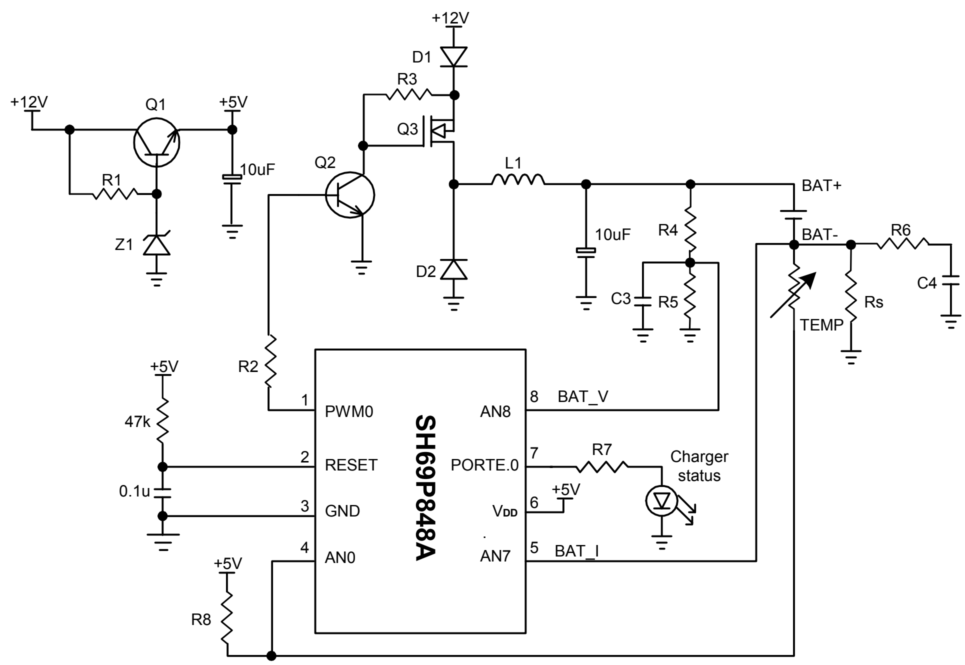


External RC Oscillator Characteristics Graphs (External Resistor vs. Frequency):



Application Circuits (For Reference Only):

(1) Operating voltage: 5.0V
(2) Oscillator: Internal RC 4MHz
(3) AN0: Temperature analog input;
AN7: Current analog input;
AN8: Battery voltage analog input;
PWM0: PWM output;
PORTE.0: Charging status output

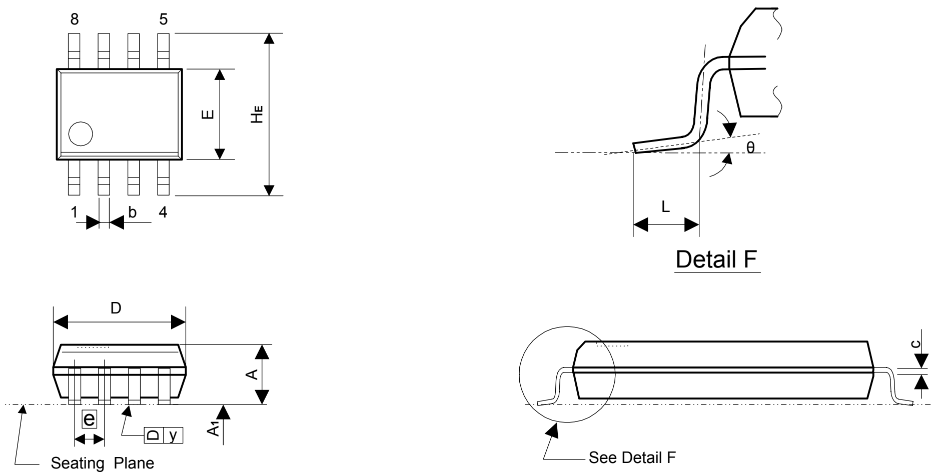


Ordering Information

Part No.	Package
SH69P848AM/008MU	8L SOP

Package Information

SOP 8L Outline Dimensions unit: inches/mm



Symbol	Dimensions in inches	Dimensions in mm
A	0.069 Max.	1.75 Max.
	0.053 Min.	1.35 Min.
A_1	0.010 Max.	0.25 Max.
	0.004 Min.	0.10 Min.
b	0.016 Typ.	0.41 Typ.
c	0.008 Typ.	0.20 Typ.
D	0.196 Max.	4.98 Max.
	0.189 Min.	4.80 Min.
E	0.157 Max.	3.99 Max.
	0.150 Min.	3.81 Min.
e	0.050 Typ.	1.27 Typ.
H_E	0.244 Max.	6.20 Max.
	0.228 Min.	5.79 Min.
L	0.050 Max.	1.27 Max.
	0.016 Min.	0.41 Min.
y	0.004 Max.	0.10 Max.
θ	0° ~ 8°	0° ~ 8°

Notes:

(1) The maximum value of dimension D includes end flash.

(2) Dimension E does not include resin fins.

Data Sheet Revision History

Version	Content	Date
2.0	Ordering information update	Dec. 2008
1.0	Original	Feb. 2008