HOLTEK

HT49R30A-1/HT49C30-1/HT49C30L LCD Type 8-Bit MCU

Technical Document

- Tools Information
- FAQs
- Application Note
  - HA0017E Controlling the Read/Write Function of the HT24 Series EEPROM Using the HT49 Series MCUs
  - HA0024E Using the RTC in the HT49 MCU Series
  - HA0025E Using the Time Base in the HT49 MCU Series
  - HA0026E Using the I/O Ports on the HT49 MCU Series
  - HA0027E Using the Timer/Event Counter in the HT49 MCU Series

Features

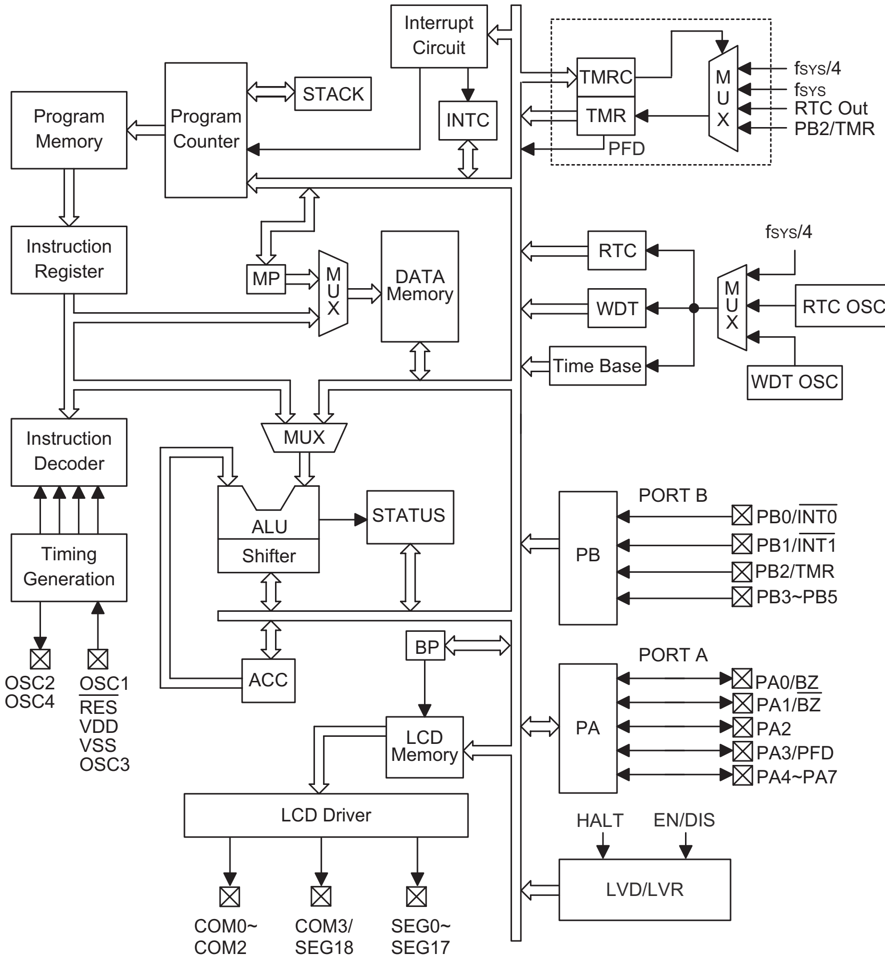
- Operating voltage:
  f_{SYS}= 4MHz: 2.2V~5.5V for HT49R30A-1/HT49C30-1
  f_{SYS}= 8MHz: 3.3V~5.5V for HT49R30A-1/HT49C30-1
  f_{SYS}= 500kHz: 1.2V~2.2V for HT49C30L
- 6 input lines
- 8 bidirectional I/O lines
- Two external interrupt input
- One 8-bit programmable timer/event counter with PFD (programmable frequency divider) function
- LCD driver with 19×2, 19×3 or 18×4 segments
- 2K×14 program memory
- 96×8 data memory RAM
- Real Time Clock (RTC)
- 8-bit prescaler for RTC
- Watchdog Timer
- Buzzer output
- On-chip crystal, RC and 32768Hz crystal oscillator
- HALT function and wake-up feature reduce power consumption
- 4-level subroutine nesting
- Bit manipulation instruction
- 14-bit table read instruction
- Up to 0.5μs instruction cycle with 8MHz system clock for HT49R30A-1/HT49C30-1
- Up to 8μs instruction cycle with 500kHz system clock for HT49C30L
- 63 powerful instructions
- All instructions in 1 or 2 machine cycles
- Low voltage reset/detector for HT49R30A-1/HT49C30-1
- 48-pin SSOP package

General Description

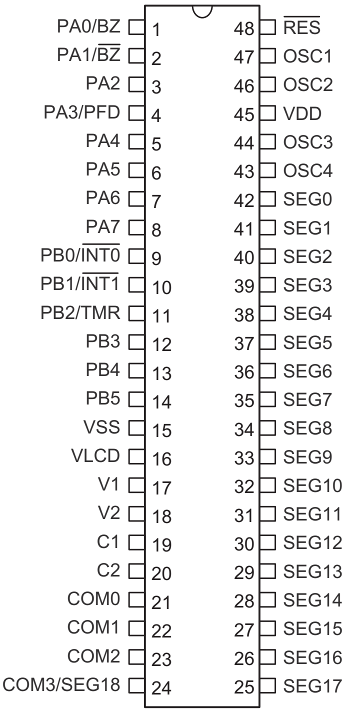
The HT49R30A-1/HT49C30-1/HT49C30L are 8-bit, high performance, RISC architecture microcontroller devices specifically designed for a wide range of LCD applications. The mask version HT49C30-1 and HT49C30L are fully pin and functionally compatible with the OTP version HT49R30A-1 device. The HT49C30L is a low voltage version, with the ability to operate at a minimum power supply of 1.2V, making it suitable for single cell battery applications.

The advantages of low power consumption, I/O flexibility, programmable frequency divider, timer functions, oscillator options, HALT and wake-up functions and buzzer driver in addition to a flexible and configurable LCD interface, enhance the versatility of these devices to control a wide range of LCD-based application possibilities such as measuring scales, electronic multimeters, gas meters, timers, calculators, remote controllers and many other LCD-based industrial and home appliance applications.

Block Diagram



Pin Assignment



HT49R30A-1/HT49C30-1/HT49C30L
— 48 SSOP-A

Pad Description

Pad Name	I/O	Options	Description
PA0/$\overline{\text{BZ}}$ PA1/$\overline{\text{BZ}}$ PA2 PA3/PFD PA4~PA7	I/O	Wake-up Pull-high or None CMOS or NMOS	PA0~PA7 constitute an 8-bit bidirectional input/output port with Schmitt trigger input capability. Each pin on port can be configured as a wake-up input by options. PA0~PA3 can be configured as a CMOS output or NMOS input/output with or without pull-high resistor by options. PA4~PA7 are always pull-high NMOS input/output. Of the eight bits, PA0~PA1 can be set as I/O pins or buzzer outputs by options. PA3 can be set as an I/O pin or as a PFD output also by options.
PB0/$\overline{\text{INT0}}$ PB1/$\overline{\text{INT1}}$ PB2/TMR PB3~PB5	I	—	PB0~PB5 constitute a 6-bit Schmitt trigger input port. Each pin on port are with pull-high resistor. Of the six bits, PB0 and PB1 can be set as input pins or as external interrupt control pins ($\overline{\text{INT0}}$) and ($\overline{\text{INT1}}$) respectively, by software application. PB2 can be set as an input pin or as a timer/event counter input pin TMR also by software application.
VLCD	I	—	LCD power supply for HT49R30A-1/HT49C30-1. Voltage pump for HT49C30L.
V2	I	—	Voltage pump for HT49R30A-1/HT49C30-1. LCD power supply for HT49C30L.
V1,C1,C2	I	—	Voltage pump
COM0~COM2 COM3/SEG18	O	1/2, 1/3 or 1/4 Duty	SEG18 can be set as a segment or as a common output driver for LCD panel by options. COM0~COM2 are outputs for LCD panel plate.
SEG0~SEG17	O	—	LCD driver outputs for LCD panel segments
OSC1 OSC2	I O	Crystal or RC	OSC1 and OSC2 are connected to an RC network or a crystal (by options) for the internal system clock. In the case of RC operation, OSC2 is the output terminal for 1/4 system clock. The system clock may come from the RTC oscillator. If the system clock comes from RTC OSC, these two pins can be floating.
OSC3 OSC4	I O	RTC or System Clock	Real time clock oscillators. OSC3 and OSC4 are connected to a 32768Hz crystal oscillator for timing purposes or to a system clock source (depending on the options).
VSS	—	—	Negative power supply, ground
VDD	—	—	Positive power supply
$\overline{\text{RES}}$	I	—	Schmitt trigger reset input, active low

Absolute Maximum Ratings

Supply Voltage..........................V_{SS}–0.3V to V_{SS}+6.0V*

Storage Temperature ....–50°C to 125°C

Operating Temperature...........................–40°C to 85°C

Supply Voltage ....V_{SS}–0.3V to V_{SS}+2.5V**

Input Voltage..............................V_{SS}–0.3V to V_{DD}+0.3V

Note: These are stress ratings only. Stresses exceeding the range specified under ″Absolute Maximum Ratings″ may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

″*″ For HT49R30A-1/HT49C30-1

″**″ For HT49C30L

D.C. Characteristics

V_{DD}=1.5V for HT49C30L, V_{DD}=3V & V_{DD}=5V for HT49R30A-1 and HT49C30-1 Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	For HT49C30L	1.2	—	2.2	V
			LVR disable, f_{SYS}=4MHz (for HT49R30A-1/HT49C30-1)	2.2	—	5.5	V
			f_{SYS}=8MHz (for HT49R30A-1/HT49C30-1)	3.3	—	5.5	V
V_{LCD}	LCD Power Supply (Note *)	—	For HT49R30A-1/HT49C30-1, VA≤5.5V	2.2	—	5.5	V
I_{DD1}	Operating Current (Crystal OSC)	1.5V	No load, f_{SYS}=455kHz	—	60	100	μA
		3V	No load, f_{SYS}=4MHz	—	1	2	mA
		5V		—	3	5	mA
I_{DD2}	Operating Current (RC OSC)	1.5V	No load, f_{SYS}=400kHz	—	50	100	μA
		3V	No load, f_{SYS}=4MHz	—	1	2	mA
		5V		—	3	5	mA
I_{DD3}	Operating Current (Crystal OSC, RC OSC)	5V	No load, f_{SYS}=8MHz	—	4	8	mA
I_{DD4}	Operating Current (f_{SYS}=RTC OSC)	1.5V	No load	—	2.5	5	μA
		3V		—	0.3	0.6	mA
		5V		—	0.6	1	mA
I_{STB1}	Standby Current (*f_S=T1)	1.5V	No load, system HALT, LCD Off at HALT	—	0.1	0.5	μA
		3V		—	—	1	μA
		5V		—	—	2	μA
I_{STB2}	Standby Current (*f_S=RTC OSC)	1.5V	No load, system HALT, LCD On at HALT, C type	—	1	2	μA
		3V		—	2.5	5	μA
		5V		—	10	20	μA
I_{STB3}	Standby Current (*f_S=WDT RC OSC)	1.5V	No load, system HALT LCD On at HALT, C type	—	0.5	1	μA
		3V		—	2	5	μA
		5V		—	6	10	μA
I_{STB4}	Standby Current (*f_S=RTC OSC)	3V	No load, system HALT, LCD On at HALT, R type, 1/2 bias	—	17	30	μA
		5V		—	34	60	μA
I_{STB5}	Standby Current (*f_S=RTC OSC)	3V	No load, system HALT, LCD On at HALT, R type, 1/3 bias	—	13	25	μA
		5V		—	26	50	μA
I_{STB6}	Standby Current (*f_S=WDT RC OSC)	3V	No load, system HALT, LCD On at HALT, R type, 1/2 bias	—	14	25	μA
		5V		—	28	50	μA
I_{STB7}	Standby Current (*f_S=WDT RC OSC)	3V	No load, system HALT, LCD On at HALT, R type, 1/3 bias	—	10	20	μA
		5V		—	20	40	μA
V_{IL1}	Input Low Voltage for I/O Ports, TMR and $\overline{INT}$	—	—	0	—	$0.3V_{DD}$	V

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{IH1}	Input High Voltage for I/O Ports, TMR and $\overline{\text{INT}}$	—	For HT49C30L	$0.8V_{DD}$	—	V_{DD}	V
			For HT49R30A-1/HT49C30-1	$0.7V_{DD}$	—	V_{DD}	V
V_{IL2}	Input Low Voltage ($\overline{\text{RES}}$)	—	—	0	—	$0.4V_{DD}$	V
V_{IH2}	Input High Voltage ($\overline{\text{RES}}$)	—	—	$0.9V_{DD}$	—	V_{DD}	V
I_{OL1}	I/O Port Sink Current	1.5V	$V_{OL}=0.1V_{DD}$	0.4	0.8	—	mA
		3V		6	12	—	mA
		5V		10	25	—	mA
I_{OH1}	I/O Port Source Current	1.5V	$V_{OH}=0.9V_{DD}$	−0.3	−0.6	—	mA
		3V		−2	-4	—	mA
		5V		−5	−8	—	mA
I_{OL2}	LCD Common and Segment Current	3V	$V_{OL}=0.1V_{DD}$	210	420	—	μA
		5V		350	700	—	μA
I_{OH2}	LCD Common and Segment Current	3V	$V_{OH}=0.9V_{DD}$	−80	−160	—	μA
		5V		−180	−360	—	μA
R_{PH}	Pull-high Resistance	1.5V	—	75	150	300	kΩ
		3V		20	60	100	kΩ
		5V		10	30	50	kΩ
V_{LVR}	Low Voltage Reset Voltage	—	—	2.7	3.2	3.6	V
V_{LVD}	Low Voltage Detector Voltage	—	—	3.0	3.3	3.6	V

Note: ″*″ for the value of VA refer to the LCD driver section.
″*f_S″ please refer to the WDT clock option

A.C. Characteristics

V_{DD}=1.5V for HT49C30L, V_{DD}=3V & V_{DD}=5V for HT49R30A-1 and HT49C30-1 Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SYS1}	System Clock (Crystal OSC)	—	1.2V~2.2V (for HT49C30L)	400	—	500	kHz
		—	2.2V~5.5V	400		4000	kHz
		—	3.3V~5.5V	400	—	8000	kHz
f_{SYS2}	System Clock (RC OSC)	—	1.2V~2.2V (for HT49C30L)	400	—	500	kHz
		—	2.2V~5.5V	400	—	4000	kHz
		—	3.3V~5.5V	400	—	8000	kHz
f_{SYS3}	System Clock (32768Hz Crystal OSC)	—	—	—	32768	—	Hz
f_{RTCOSC}	RTC Frequency	—	—	—	32768	—	Hz
f_{TIMER}	Timer I/P Frequency	—	1.2V~2.2V (for HT49C30L)	0	—	500	kHz
		—	2.2V~5.5V	0	—	4000	kHz
		—	3.3V~5.5V	0	—	8000	kHz
t_{WDTOSC}	Watchdog Oscillator Period	1.5V	—	35	70	140	μs
		3V		45	90	180	μs
		5V		32	65	130	μs
t_{RES}	External Reset Low Pulse Width	—	For HT49C30L	10	—	—	μs
			For HT49R30A-1/HT49C30-1	1	—	—	μs
t_{SST}	System Start-up Timer Period	—	Wake-up from HALT	—	1024	—	$^{*}t_{SYS}$
t_{LVR}	Low Voltage Width to Reset	—	—	0.25	1	2	ms
t_{INT}	Interrupt Pulse Width	—	For HT49C30L	10	—	—	μs
			For HT49R30A-1/HT49C30-1	1	—	—	μs

Note: $^{*}t_{SYS}$= $1/f_{SYS1}$, $1/f_{SYS2}$ or $1/f_{SYS3}$

Functional Description

Execution Flow

The system clock is derived from either a crystal or an RC oscillator or a 32768Hz crystal oscillator. It is internally divided into four non-overlapping clocks. One instruction cycle consists of four system clock cycles.

Instruction fetching and execution are pipelined in such a way that a fetch takes one instruction cycle while decoding and execution takes the next instruction cycle. The pipelining scheme causes each instruction to effectively execute in a cycle. If an instruction changes the value of the program counter, two cycles are required to complete the instruction.

Program Counter – PC

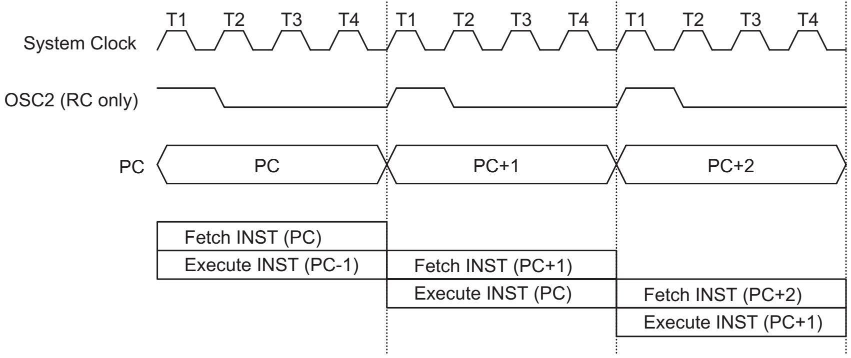
The program counter (PC) is of 11 bits wide and controls the sequence in which the instructions stored in the program ROM are executed. The contents of the PC can specify a maximum of 2048 addresses.

After accessing a program memory word to fetch an instruction code, the value of the PC is incremented by one. The PC then points to the memory word containing the next instruction code.

When executing a jump instruction, conditional skip execution, loading a PCL register, a subroutine call, an initial reset, an internal interrupt, an external interrupt, or returning from a subroutine, the PC manipulates the program transfer by loading the address corresponding to each instruction.

The conditional skip is activated by instructions. Once the condition is met, the next instruction, fetched during the current instruction execution, is discarded and a dummy cycle replaces it to get a proper instruction; otherwise proceed with the next instruction.

The lower byte of the PC (PCL) is a readable and writeable register (06H). Moving data into the PCL performs a short jump. The destination is within 256 locations.



Execution Flow

Mode	Program Counter										
	*10	*9	*8	*7	*6	*5	*4	*3	*2	*1	*0
Initial Reset	0	0	0	0	0	0	0	0	0	0	0
External Interrupt 0	0	0	0	0	0	0	0	0	1	0	0
External Interrupt 1	0	0	0	0	0	0	0	1	0	0	0
Timer/Event Counter Overflow	0	0	0	0	0	0	0	1	1	0	0
Time Base Interrupt	0	0	0	0	0	0	1	0	0	0	0
RTC Interrupt	0	0	0	0	0	0	1	0	1	0	0
Skip	Program Counter + 2										
Loading PCL	*10	*9	*8	@7	@6	@5	@4	@3	@2	@1	@0
Jump, Call Branch	#10	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
Return From Subroutine	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

Program Counter

Note: *10~*0: Program counter bits
#10~#0: Instruction code bits
S10~S0: Stack register bits
@7~@0: PCL bits

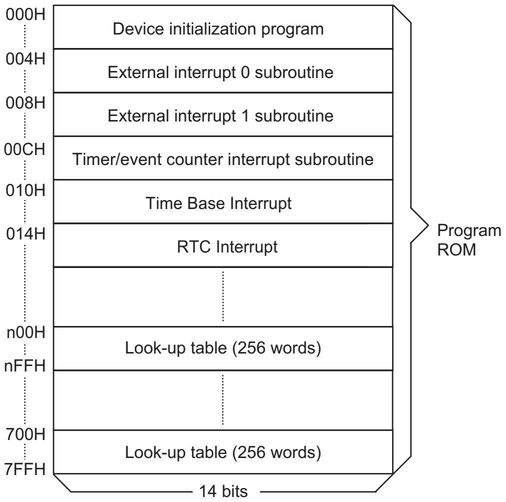
When a control transfer takes place, an additional dummy cycle is required.

Program Memory – ROM

The program memory is used to store the program instructions which are to be executed. It also contains data, table, and interrupt entries, and is organized into 2048×14 bits which are addressed by the program counter and table pointer.

Certain locations in the ROM are reserved for special usage:

- Location 000H
  Location 000H is reserved for program initialization. After chip reset, the program always begins execution at this location.
- Location 004H
  Location 004H is reserved for the external interrupt service program. If the $\overline{\text{INT0}}$ input pin is activated, and the interrupt is enabled, and the stack is not full, the program begins execution at location 004H.
- Location 008H
  Location 008H is reserved for the external interrupt service program also. If the $\overline{\text{INT1}}$ input pin is activated, and the interrupt is enabled, and the stack is not full, the program begins execution at location 008H.
- Location 00CH
  Location 00CH is reserved for the timer/event counter interrupt service program. If a timer interrupt results from a timer/event counter overflow, and if the interrupt is enabled and the stack is not full, the program begins execution at location 00CH.
- Location 010H
  Location 010H is reserved for the Time Base interrupt service program. If a Time Base interrupt occurs, and the interrupt is enabled, and the stack is not full, the program begins execution at location 010H.
- Location 014H
  Location 014H is reserved for the real time clock interrupt service program. If a real time clock interrupt occurs, and the interrupt is enabled, and the stack is not full, the program begins execution at location 014H.
- Table location
  Any location in the ROM can be used as a look-up table. The instructions ″TABRDC [m]″ (the current page, 1 page=256 words) and ″TABRDL [m]″ (the last page) transfer the contents of the lower-order byte to the specified data memory, and the contents of the higher-order byte to TBLH (Table Higher-order byte register) (08H). Only the destination of the lower-order byte in the table is well-defined; the other bits of the table word are all transferred to the lower portion of TBLH, and the remaining 2 bit is read as ″0″. The TBLH is read only, and the table pointer (TBLP) is a read/write register (07H), indicating the table location. Before accessing the table, the location should be placed in TBLP. All the table related instructions require 2 cycles to complete the operation. These areas may function as a normal ROM depending upon the user′s requirements.



Note: n ranges from 0 to 7

Program Memory

Stack Register – STACK

The stack register is a special part of the memory used to save the contents of the program counter. The stack is organized into 4 levels and is neither part of the data nor part of the program, and is neither readable nor writeable. Its activated level is indexed by a stack pointer (SP) and is neither readable nor writeable. At a commencement of a subroutine call or an interrupt acknowledgment, the contents of the program counter is pushed onto the stack. At the end of the subroutine or interrupt routine, signaled by a return instruction (RET or RETI), the contents of the program counter is restored to its previous value from the stack. After chip reset, the SP will point to the top of the stack.

Instruction(s)	Table Location										
	***10**	***9**	***8**	***7**	***6**	***5**	***4**	***3**	***2**	***1**	***0**
TABRDC [m]	P10	P9	P8	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL [m]	1	1	1	@7	@6	@5	@4	@3	@2	@1	@0

Table Location

Note: *10~*0: Table location bits
@7~@0: Table pointer bits
P10~P8: Current program counter bits

If the stack is full and a non-masked interrupt takes place, the interrupt request flag is recorded but the acknowledgment is still inhibited. Once the SP is decremented (by RET or RETI), the interrupt is serviced. This feature prevents stack overflow, allowing the programmer to use the structure easily. Likewise, if the stack is full, and a ″CALL″ is subsequently executed, a stack overflow occurs and the first entry is lost (only the most recent four return addresses are stored).

Data Memory – RAM

The data memory (RAM) is designed with 113×8 bits, and is divided into two functional groups, namely special function registers and general purpose data memory, most of which are readable/writeable, although some are read only.

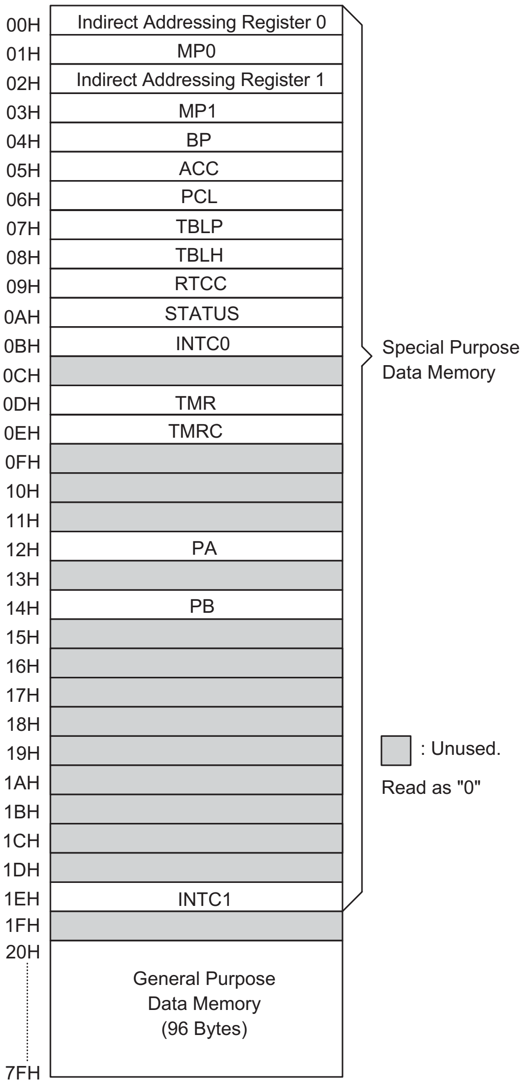
Of the two types of functional groups, the special function registers consist of an Indirect addressing register 0 (00H), a Memory pointer register 0 (MP0;01H), an Indirect addressing register 1 (02H), a Memory pointer register 1 (MP1;03H), a Bank pointer (BP;04H), an Accumulator (ACC;05H), a Program counter lower-order byte register (PCL;06H), a Table pointer (TBLP;07H), a Table higher-order byte register (TBLH;08H), a Real time clock control register (RTCC;09H), a Status register (STATUS;0AH), an Interrupt control register 0 (INTC0;0BH), a timer/event counter (TMR;0DH), a timer/event counter control register (TMRC;0EH), I/O registers (PA;12H, PB;14H), and Interrupt control register 1 (INTC1;1EH). On the other hand, the general purpose data memory, addressed from 20H to 7FH, is used for data and control information under instruction commands.

The areas in the RAM can directly handle arithmetic, logic, increment, decrement, and rotate operations. Except some dedicated bits, Each pin in the RAM can be set and reset by ″SET [m].i″ and ″CLR [m].i″ They are also indirectly accessible through the Memory pointer register 0 (MP0;01H) or the Memory pointer register 1 (MP1;03H).

Indirect Addressing Register

Location 00H and 02H are indirect addressing registers that are not physically implemented. Any read/write operation of [00H] and [02H] accesses the RAM pointed to by MP0 (01H) and MP1(03H) respectively. Reading location 00H or 02H indirectly returns the result 00H. While, writing it indirectly leads to no operation.

The function of data movement between two indirect addressing registers is not supported. The memory pointer registers, MP0 (7-bit) and MP1 (7-bit), used to access the RAM by combining corresponding indirect addressing registers. MP0 can only be applied to data memory, while MP1 can be applied to data memory and LCD display memory.



RAM Mapping

Accumulator – ACC

The accumulator (ACC) is related to the ALU operations. It is also mapped to location 05H of the RAM and is capable of operating with immediate data. The data movement between two data memory locations must pass through the ACC.

Arithmetic and Logic Unit – ALU

This circuit performs 8-bit arithmetic and logic operations and provides the following functions:

- Arithmetic operations (ADD, ADC, SUB, SBC, DAA)
- Logic operations (AND, OR, XOR, CPL)
- Rotation (RL, RR, RLC, RRC)
- Increment and Decrement (INC, DEC)
- Branch decision (SZ, SNZ, SIZ, SDZ etc.)

The ALU not only saves the results of a data operation but also changes the status register.

Status Register – STATUS

The status register (0AH) is of 8 bits wide and contains, a carry flag (C), an auxiliary carry flag (AC), a zero flag (Z), an overflow flag (OV), a power down flag (PDF), and a watchdog time-out flag (TO). It also records the status information and controls the operation sequence.

Except the TO and PDF flags, bits in the status register can be altered by instructions similar to other registers. Data written into the status register does not alter the TO or PDF flags. Operations related to the status register, however, may yield different results from those intended. The TO and PDF flags can only be changed by a Watchdog Timer overflow, chip power-up, or clearing the Watchdog Timer and executing the ″HALT″ instruction. The Z, OV, AC, and C flags reflect the status of the latest operations.

On entering the interrupt sequence or executing the subroutine call, the status register will not be automatically pushed onto the stack. If the contents of the status is important, and if the subroutine is likely to corrupt the status register, the programmer should take precautions and save it properly.

Interrupts

The device provides two external interrupts, an internal timer/event counter interrupt, an internal time base interrupt, and an internal real time clock interrupt. The interrupt control register 0 (INTC0;0BH) and interrupt control register 1 (INTC1;1EH) both contain the interrupt control bits that are used to set the enable/disable status and interrupt request flags.

Once an interrupt subroutine is serviced, other interrupts are all blocked (by clearing the EMI bit). This scheme may prevent any further interrupt nesting. Other interrupt requests may take place during this interval, but only the interrupt request flag will be recorded. If a certain interrupt requires servicing within the service routine, the EMI bit and the corresponding bit of the INTC0 or of INTC1 may be set in order to allow interrupt nesting. Once the stack is full, the interrupt request will not be acknowledged, even if the related interrupt is enabled, until the SP is decremented. If immediate service is desired, the stack should be prevented from becoming full.

All these interrupts can support a wake-up function. As an interrupt is serviced, a control transfer occurs by pushing the contents of the program counter onto the stack followed by a branch to a subroutine at the specified location in the ROM. Only the contents of the program counter is pushed onto the stack. If the contents of the register or of the status register (STATUS) is altered by the interrupt service program which corrupts the desired control sequence, the contents should be saved in advance.

External interrupts are triggered by a high to low transition of $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$, and the related interrupt request flag (EIF0;bit 4 of INTC0, EIF1;bit 5 of INTC0) is set as well. After the interrupt is enabled, the stack is not full, and the external interrupt is active, a subroutine call to location 04H or 08H occurs. The interrupt request flag (EIF0 or EIF1) and EMI bits are all cleared to disable other interrupts.

The internal timer/event counter interrupt is initialized by setting the timer/event counter interrupt request flag (TF;bit 6 of INTC0), which is normally caused by a timer overflow. After the interrupt is enabled, and the stack is not full, and the TF bit is set, a subroutine call to location 0CH occurs. The related interrupt request flag (TF) is reset, and the EMI bit is cleared to disable further interrupts.

Bit No.	Label	Function
0	C	C is set if the operation results in a carry during an addition operation or if a borrow does not take place during a subtraction operation; otherwise C is cleared. C is also affected by a rotate through carry instruction.
1	AC	AC is set if the operation results in a carry out of the low nibbles in addition or no borrow from the high nibble into the low nibble in subtraction; otherwise AC is cleared.
2	Z	Z is set if the result of an arithmetic or logic operation is zero; otherwise Z is cleared.
3	OV	OV is set if the operation results in a carry into the highest-order bit but not a carry out of the highest-order bit, or vice versa; otherwise OV is cleared.
4	PDF	PDF is cleared by either a system power-up or executing the ″CLR WDT″ instruction. PDF is set by executing the ″HALT″ instruction.
5	TO	TO is cleared by a system power-up or executing the ″CLR WDT″ or ″HALT″ instruction. TO is set by a WDT time-out.
6, 7	—	Unused bit, read as ″0″

Status (0AH) Register

Bit No.	Label	Function
0	EMI	Control the master (global) interrupt (1=enabled; 0=disabled)
1	EEI0	Control the external interrupt 0 (1=enabled; 0=disabled)
2	EEI1	Control the external interrupt 1 (1=enabled; 0=disabled)
3	ETI	Control the timer/event counter interrupt (1=enabled; 0=disabled)
4	EIF0	External interrupt 0 request flag (1=active; 0=inactive)
5	EIF1	External interrupt 1 request flag (1=active; 0=inactive)
6	TF	Internal timer/event counter request flag (1=active; 0=inactive)
7	—	Unused bit, read as ″0″

INTC0 (0BH) Register

Bit No.	Label	Function
0	ETBI	Control the time base interrupt (1=enabled; 0:disabled)
1	ERTI	Control the real time clock interrupt (1=enabled; 0:disabled)
2, 3	—	Unused bit, read as ″0″
4	TBF	Time base request flag (1=active; 0=inactive)
5	RTF	Real time clock request flag (1=active; 0=inactive)
6, 7	—	Unused bit, read as ″0″

INTC1 (1EH) Register

The time base interrupt is initialized by setting the time base interrupt request flag (TBF;bit 4 of INTC1), that is caused by a regular time base signal. After the interrupt is enabled, and the stack is not full, and the TBF bit is set, a subroutine call to location 10H occurs. The related interrupt request flag (TBF) is reset and the EMI bit is cleared to disable further interrupts.

The real time clock interrupt is initialized by setting the real time clock interrupt request flag (RTF;bit 5 of INTC1), that is caused by a regular real time clock signal. After the interrupt is enabled, and the stack is not full, and the RTF bit is set, a subroutine call to location 14H occurs. The related interrupt request flag (RTF) is reset and the EMI bit is cleared to disable further interrupts.

During the execution of an interrupt subroutine, other interrupt acknowledgments are all held until the ″RETI″ instruction is executed or the EMI bit and the related interrupt control bit are set both to 1 (if the stack is not full). To return from the interrupt subroutine, ″RET″ or ″RETI″ may be invoked. RETI sets the EMI bit and enables an interrupt service, but RET does not.

Interrupts occurring in the interval between the rising edges of two consecutive T2 pulses are serviced on the latter of the two T2 pulses if the corresponding interrupts are enabled. In the case of simultaneous requests, the priorities in the following table apply. These can be masked by resetting the EMI bit.

Interrupt Source	Priority	Vector
External interrupt 0	1	04H
External interrupt 1	2	08H
Timer/event counter overflow	3	0CH
Time base interrupt	4	10H
Real time clock interrupt	5	14H

The timer/event counter interrupt request flag (TF), external interrupt 1 request flag (EIF1), external interrupt 0 request flag (EIF0), enable timer/event counter interrupt bit (ETI), enable external interrupt 1 bit (EEI1), enable external interrupt 0 bit (EEI0), and enable master interrupt bit (EMI) make up of the Interrupt Control register 0 (INTC0) which is located at 0BH in the RAM. The real time clock interrupt request flag (RTF), time base interrupt request flag (TBF), enable real time clock interrupt bit (ERTI), and enable time base interrupt bit (ETBI), constitute the Interrupt Control register 1 (INTC1) which is located at 1EH in the RAM. EMI, EEI0, EEI1, ETI, ETBI, and ERTI are all used to control the enable/disable status of interrupts. These bits prevent the requested interrupt from being serviced. Once the interrupt request flags (RTF, TBF, TF, EIF1, EIF0) are all set, they remain in the INTC1 or INTC0 respectively until the interrupts are serviced or cleared by a software instruction.

It is recommended that a program not use the ″CALL subroutine″ within the interrupt subroutine. It′s because interrupts often occur in an unpredictable manner or require to be serviced immediately in some applications. At this time, if only one stack is left, and enabling the interrupt is not well controlled, operation of the ″call″ in the interrupt subroutine may damage the original control sequence.

Oscillator Configuration

The device provides three oscillator circuits for system clocks, i.e., RC oscillator, crystal oscillator and 32768Hz crystal oscillator, determined by options. No matter what type of oscillator is selected, the signal is used for the system clock. The HALT mode stops the system oscillator (RC and crystal oscillator only) and ignores external signal to conserve power. The 32768Hz crystal oscillator (system oscillator) still runs at HALT mode. If the 32768Hz crystal oscillator is selected as the system oscillator, the system oscillator is not stopped; but the instruction execution is stopped. Since the (used as system oscillator or oscillator) is also designed for timing purposes, the internal timing (RTC, time base, WDT) operation still runs even if the system enters the HALT mode.

Of the three oscillators, if the RC oscillator is used, an external resistor between OSC1 and VSS is required, and the range of the resistance should be from 24kΩ to 1MΩ for HT49R30A-1/HT49C30-1 and from 560kΩ to 1MΩ for HT49C30L. The system clock, divided by 4, is available on OSC2 with pull-high resistor, which can be used to synchronize external logic. The RC oscillator provides the most cost effective solution. However, the frequency of the oscillation may vary with VDD, temperature, and the chip itself due to process variations. It is therefore, not suitable for timing sensitive operations where accurate oscillator frequency is desired.

On the other hand, if the crystal oscillator is selected, a crystal across OSC1 and OSC2 is needed to provide the feedback and phase shift required for the oscillator, and no other external components are required. A resonator may be connected between OSC1 and OSC2 to replace the crystal and to get a frequency reference, but two external capacitors in OSC1 and OSC2 are required.

There is another oscillator circuit designed for the real time clock. In this case, only the 32.768kHz crystal oscillator can be applied. The crystal should be connected between OSC3 and OSC4.

The RTC oscillator circuit can be controlled to oscillate quickly by setting the ″QOSC″ bit (bit 4 of RTCC). It is recommended to turn on the quick oscillating function upon power on, and then turn it off after 2 seconds.

The WDT oscillator is a free running on-chip RC oscillator, and no external components are required. Although the system enters the power down mode, the system clock stops, and the WDT oscillator still works with a period of approximately 65μs at 5V. The WDT oscillator can be disabled by options to conserve power.

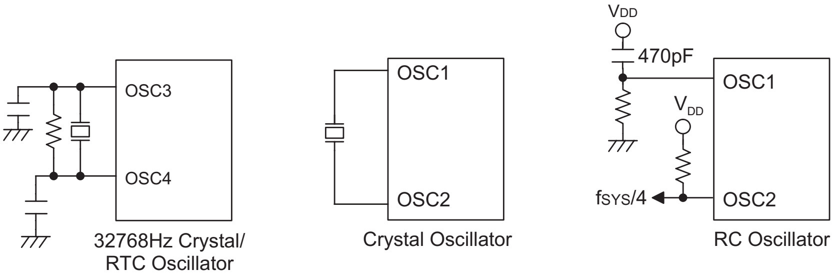
Watchdog Timer − WDT

The WDT clock source is implemented by a dedicated RC oscillator (WDT oscillator) or an instruction clock (system clock/4) or a real time clock oscillator (RTC oscillator). The timer is designed to prevent a software malfunction or sequence from jumping to an unknown location with unpredictable results. The WDT can be disabled by options. But if the WDT is disabled, all executions related to the WDT lead to no operation.

The WDT time-out period is $f_S/2^{15}$~$f_S/2^{16}$. If the WDT clock source chooses the internal WDT oscillator, the time-out period may vary with temperature, VDD, and process variations. On the other hand, if the clock source selects the instruction clock and the ″HALT″ instruction is executed, WDT may stop counting and lose its protecting purpose, and the logic can only be restarted by an external logic.

When the device operates in a noisy environment, using the on-chip RC oscillator (WDT OSC) is strongly recommended, since the HALT can stop the system clock.

The WDT overflow under normal operation initializes a ″chip reset″ and sets the status bit ″TO″. In the HALT mode, the overflow initializes a ″warm reset″, and only the program counter and SP are reset to zero. To clear the contents of the WDT, there are three methods to be adopted, i.e., external reset (a low level to $\overline{\text{RES}}$), software instruction, and a ″HALT″ instruction. There are two types of software instructions; ″CLR WDT″ and the other set − ″CLR WDT1″ and ″CLR WDT2″. Of these two types of instruction, only one type of instruction can be active at a time depending on the options − ″CLR WDT″ times selection option. If the ″CLR WDT″ is selected (i.e., CLR WDT times equal one), any execution



32768Hz Crystal/
RTC Oscillator

Crystal Oscillator

RC Oscillator

System Oscillator

of the ″CLR WDT″ instruction clears the WDT. In the case that ″CLR WDT1″ and ″CLR WDT2″ are chosen (i.e., CLR WDT times equal two), these two instructions have to be executed to clear the WDT; otherwise, the WDT may reset the chip due to time-out.

Multi-function Timer

The device provides a multi-function timer for the WDT, time base and RTC but with different time-out periods. The multi-function timer consists of a 8-stage divider and an 7-bit prescaler, with the clock source coming from the WDT OSC or RTC OSC or the instruction clock (i.e., system clock divided by 4). The multi-function timer also provides a selectable frequency signal (ranges from $f_S/2^2$ to $f_S/2^8$) for LCD driver circuits, and a selectable frequency signal (ranges from $f_S/2^2$ to $f_S/2^9$) for the buzzer output by options. It is recommended to select a near 4kHz signal to LCD driver circuits for proper display.

Time Base

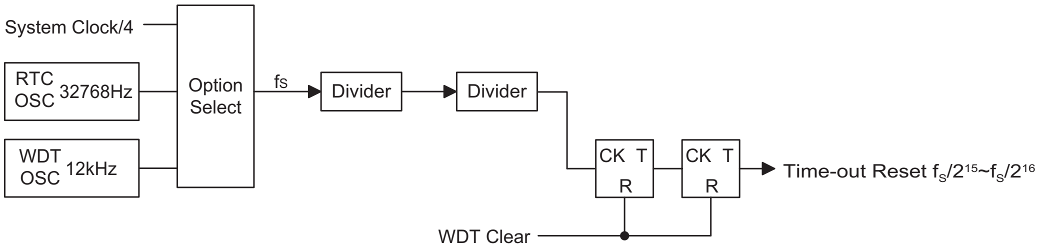
The time base offers a periodic time-out period to generate a regular internal interrupt. Its time-out period ranges from $f_S/2^{12}$ to $f_S/2^{15}$ selected by options. If time base time-out occurs, the related interrupt request flag (TBF; bit 4 of INTC1) is set. But if the interrupt is enabled, and the stack is not full, a subroutine call to location 10H occurs.

Real Time Clock – RTC

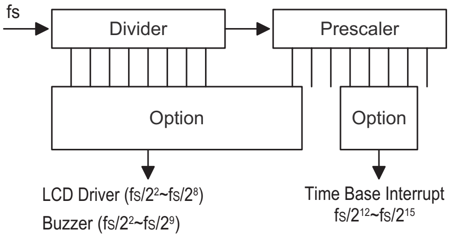
The real time clock (RTC) is operated in the same manner as the time base that is used to supply a regular internal interrupt. Its time-out period ranges from $f_S/2^8$ to $f_S/2^{15}$ by software programming . Writing data to RT2, RT1 and RT0 (bit2, 1, 0 of RTCC;09H) yields various time-out periods. If the RTC time-out occurs, the related interrupt request flag (RTF;bit 5 of INTC1) is set. But if the interrupt is enabled, and the stack is not full, a subroutine call to location 14H occurs. The real time clock time-out signal also can be applied to be a clock source of timer/event counter for getting a longer time-out period.

RT2	RT1	RT0	RTC Clock Divided Factor
0	0	0	2^{8*}
0	0	1	2^{9*}
0	1	0	2^{10*}
0	1	1	2^{11*}
1	0	0	2^{12}
1	0	1	2^{13}
1	1	0	2^{14}
1	1	1	2^{15}

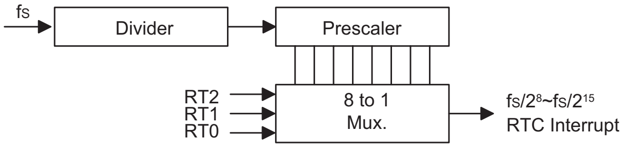
Note: ″*″ not recommended for use



Watchdog Timer



Time Base



Real Time Clock

Power Down Operation − HALT

The HALT mode is initialized by the ″HALT″ instruction and results in the following.

- The system oscillator turns off but the WDT or RTC oscillator keeps running (if the WDT oscillator or the real time clock is selected).
- The contents of the on-chip RAM and of the registers remain unchanged.
- The WDT is cleared and start recounting (if the WDT clock source is from the WDT oscillator or the real time clock oscillator).
- All I/O ports maintain their original status.
- The PDF flag is set but the TO flag is cleared.
- LCD driver is still running (if the WDT OSC or RTC OSC is selected).

The system quits the HALT mode by an external reset, an interrupt, an external falling edge signal on port A, or a WDT overflow. An external reset causes device initialization, and the WDT overflow performs a ″warm reset″. After examining the TO and PDF flags, the reason for chip reset can be determined. The PDF flag is cleared by system power-up or by executing the ″CLR WDT″ instruction, and is set by executing the ″HALT″ instruction. On the other hand, the TO flag is set if WDT time-out occurs, and causes a wake-up that only resets the program counter and SP, and leaves the others at their original state.

The port A wake-up and interrupt methods can be considered as a continuation of normal execution. Each pin in port A can be independently selected to wake up the device by options. Awakening from an I/O port stimulus, the program resumes execution of the next instruction. On the other hand, awakening from an interrupt, two sequences may occur. If the related interrupt is disabled or the interrupt is enabled but the stack is full, the program resumes execution at the next instruction. But if the interrupt is enabled, and the stack is not full, the regular interrupt response takes place.

When an interrupt request flag is set before entering the ″HALT″ status, the system cannot be awaken using that interrupt.

If wake-up events occur, it takes 1024 t_{SYS} (system clock period) to resume normal operation. In other words, a dummy period is inserted after the wake-up. If the wake-up results from an interrupt acknowledgment, the actual interrupt subroutine execution is delayed by more than one cycle. However, if the Wake-up results in the next instruction execution, the execution will be performed immediately after the dummy period is finished.

To minimize power consumption, all the I/O pins should be carefully managed before entering the HALT status.

Reset

There are three ways in which reset may occur.

- $\overline{RES}$ is reset during normal operation
- $\overline{RES}$ is reset during HALT
- WDT time-out is reset during normal operation

The WDT time-out during HALT differs from other chip reset conditions, for it can perform a ″warm reset″ that resets only the program counter and SP and leaves the other circuits at their original state. Some registers remain unaffected during any other reset conditions. Most registers are reset to the ″initial condition″ once the reset conditions are met. Examining the PDF and TO flags, the program can distinguish between different ″chip resets″.

Note: ″*″ Make the length of the wiring, which is connected to the $\overline{RES}$ pin as short as possible, to avoid noise interference.

TO	PDF	RESET Conditions
0	0	$\overline{RES}$ reset during power-up
u	u	$\overline{RES}$ reset during normal operation
0	1	$\overline{RES}$ Wake-up HALT
1	u	WDT time-out during normal operation
1	1	WDT Wake-up HALT

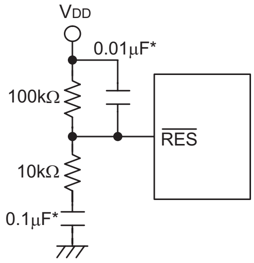
Note: ″u″ stands for unchanged

To guarantee that the system oscillator is started and stabilized, the SST (System Start-up Timer) provides an extra-delay of 1024 system clock pulses when the system awakes from the HALT state. Awaking from the HALT state, the SST delay is added.

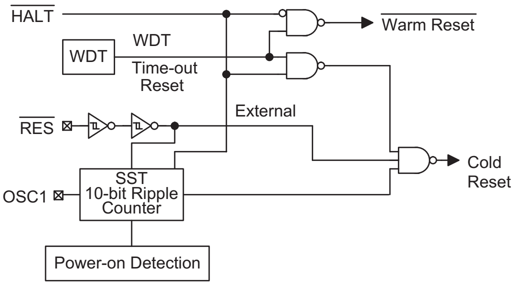
An extra option load time delay is added during reset and power on.

The functional unit chip reset status is shown below.

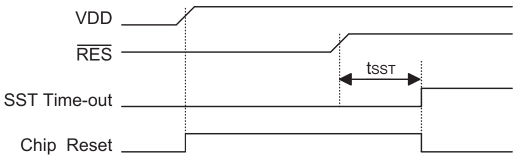
Program Counter	000H
Interrupt	Disabled
Prescaler, Divider	Cleared
WDT, RTC, Time Base	Cleared. After master reset, WDT starts counting
Timer/Event Counter	Off
Input/output Ports	Input mode
SP	Points to the top of the stack



Reset Circuit



Reset Configuration



Reset Timing Chart

The states of the registers are summarized below:

Register	Reset (Power On)	WDT Time-out (Norma Operation)	$\overline{\text{RES}}$ Reset (Normal Operation)	$\overline{\text{RES}}$ Reset (HALT)	WDT Time-out (HALT)*
TMR	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMRC	0000 1---	0000 1---	0000 1---	0000 1---	uuuu u---
Program Counter	0000H	0000H	0000H	0000H	0000H
MP0	-xxx xxxx	-uuu uuuu	-uuu uuuu	-uuu uuuu	-uuu uuuu
MP1	-xxx xxxx	-uuu uuuu	-uuu uuuu	-uuu uuuu	-uuu uuuu
BP	---- ---0	---- ---0	---- ---0	---- ---0	---- ---u
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	--xx xxxx	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu
STATUS	--00 xxxx	--1u uuuu	--uu uuuu	--01 uuuu	--11 uuuu
INTC0	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	--00 --00	--00 --00	--00 --00	--00 --00	--uu --uu
RTCC	--00 0111	--00 0111	--00 0111	--00 0111	--uu uuuu
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PB	--xx xxxx	--xx xxxx	--xx xxxx	--xx xxxx	--uu uuuu

Note: ″*″ stands for warm reset

″u″ stands for unchanged

″x″ stands for unknown

Timer/Event Counter

One timer/event counters is implemented in the device. It contains an 8-bit programmable count-up counter.

The timer/event counter clock source may come from the system clock or system clock/4 or RTC time-out signal or external source. System clock source or system clock/4 is selected by options. Using external clock input allows the user to count external events, measure time internals or pulse widths, or generate an accurate time base. While using the internal clock allows the user to generate an accurate time base.

There are two registers related to the timer/event counter, i.e., TMR ([0DH]) and TMRC ([0EH]). There are also two physical registers which are mapped to TMR location; writing TMR places the starting value in the timer/event counter preload register, while reading it yields the contents of the timer/event counter. TMRC is a timer/event counter control register used to define some options.

The TM0 and TM1 bits define the operation mode. The event count mode is used to count external events, which means that the clock source is from an external TMR pin. The timer mode functions as a normal timer with the clock source coming from the internal selected clock source. Finally, the pulse width measurement mode can be used to count the high or low level duration of the external signal TMR, and the counting is based on the internal selected clock source.

In the event count or timer mode, the timer/event counter starts counting at the current contents in the timer/event counter and ends at FFH. Once an overflow occurs, the counter is reloaded from the timer/event counter preload register, and generates an interrupt request flag (TF; bit 6 of INTC0).

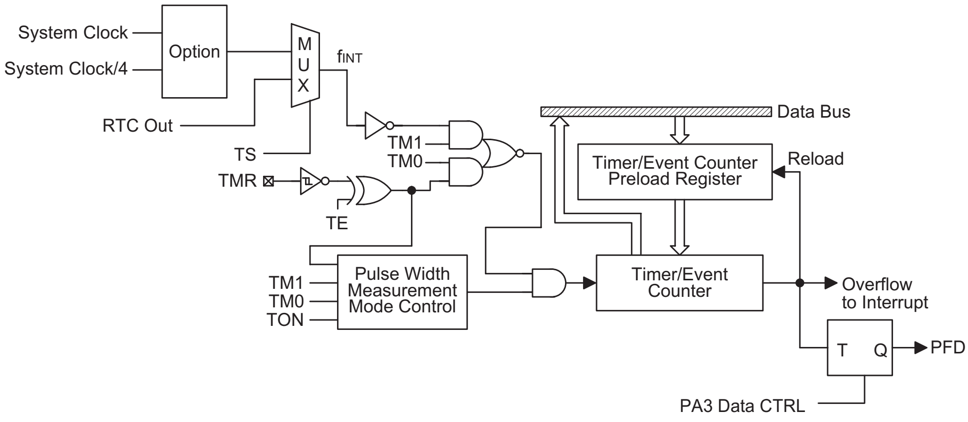
In the pulse width measurement mode with the values of the TON and TE bits equal to one, after the TMR has received a transient from low to high (or high to low if the TE bit is ″0″), it will start counting until the TMR returns to the original level and resets the TON. The measured result remains in the timer/event counter even if the activated transient occurs again. In other words, only one cycle measurement can be made until the TON is set. The cycle measurement will re-function as long as it receives further transient pulse. In this operation mode, the timer/event counter begins counting according not to the logic level but to the transient edges. In the case of counter overflows, the counter is reloaded from the timer/event counter preload register and issues an interrupt request, as in the other two modes, i.e., event and timer modes.

To enable the counting operation, the Timer ON bit (TON;bit 4 of TMRC) should be set to 1. In the pulse width measurement mode, the TON is automatically cleared after the measurement cycle is completed. But in the other two modes, the TON can only be reset by instructions. The overflow of the Timer/Event Counter is one of the wake-up sources and can also be applied to a PFD (Programmable Frequency Divider) output at PA3 by options. No matter what the operation mode is, writing a 0 to ETI disables the related interrupt service. When the PFD function is selected, executing ″CLR [PA].3″ instruction to enable PFD output and executing ″SET [PA].3″ instruction to disable PFD output.

In the case of timer/event counter OFF condition, writing data to the timer/event counter preload register also reloads that data to the timer/event counter. But if the timer/event counter is turn on, data written to the timer/event counter is kept only in the timer/event counter preload register. The timer/event counter still continues its operation until an overflow occurs.

When the timer/event counter (reading TMR) is read, the clock is blocked to avoid errors. As this may results in a counting error, blocking of the clock should be taken into account by the programmer.

It is strongly recommended to load a desired value into the TMR register first, then turn on the related timer/event counter for proper operation, because the initial value of TMR is unknown.



Timer/Event Counter

Bit No.	Label	Function
0~2	—	Unused bit, read as ″0″
3	TE	Defines the TMR active edge of the timer/event counter: In Event Counter Mode (TM1,TM0)=(0,1): 1:count on falling edge; 0:count on rising edge In Pulse Width measurement mode (TM1,TM0)=(1,1): 1: start counting on the rising edge, stop on the falling edge; 0: start counting on the falling edge, stop on the rising edge
4	TON	To enable/disable timer counting (0=disabled; 1=enabled)
5	TS	2 to 1 multiplexer control inputs to select the timer/event counter clock source (0=RTC outputs; 1= system clock or system clock/4)
6 7	TM0 TM1	To define the operating mode (TM1, TM0) 01= Event count mode (External clock) 10= Timer mode (Internal clock) 11= Pulse Width measurement mode (External clock) 00= Unused

TMRC (0EH) Register

Due to the timer/event scheme, the programmer should pay special attention on the instruction to enable then disable the timer for the first time, whenever there is a need to use the timer/event function, to avoid unpredictable result. After this procedure, the timer/event function can be operated normally.

Input/Output Ports

There are a 8-bit bidirectional input/output port, an 6-bit input port in the device, labeled PA, PB which are mapped to [12H], [14H] of the RAM, respectively. PA0~PA3 can be configured as CMOS (output) or NMOS (input/output) with or without pull-high resistor by options. PA4~PA7 are always pull-high and NMOS (input/output).

If you choose NMOS (input), Each pin on the port (PA0~PA7) can be configured as a wake-up input. PB can only be used for input operation. All the ports for the input operation (PA, PB), are non-latched, that is, the inputs should be ready at the T2 rising edge of the instruction ″MOV A, [m]″ (m=12H or 14H).

For PA output operation, all data are latched and remain unchanged until the output latch is rewritten.

When the PA structures are open drain NMOS type, it should be noted that, before reading data from the pads, a ″1″ should be written to the related bits to disable the NMOS device. That is executing first the instruction ″SET [m].i″ (i=0~7 for PA) to disable related NMOS device, and then ″MOV A, [m]″ to get stable data.

After chip reset, these input lines remain at the high level or are left floating (by options). Each pin of these output latches can be set or cleared by the ″MOV [m], A″ (m=12H) instruction.

Some instructions first input data and then follow the output operations. For example, ″SET [m].i″, ″CLR [m].i″, ″CPL [m]″, ″CPLA [m]″ read the entire port states into the CPU, execute the defined operations (bit-operation), and then write the results back to the latches or to the accumulator. When a PA line is used as an I/O line, the related PA line options should be configured as NMOS with or without pull-high resistor. Once a PA line is selected as a CMOS output, the I/O function cannot be used.

The input state of a PA line is read from the related PA pad. When the PA is configured as NMOS with or without pull-high resistor, one should be careful when applying a read-modify-write instruction to PA. Since the read-modify-write will read the entire port state (pads state) firstly, execute the specified instruction and then write the result to the port data register. When the read operation is executed, a fault pad state (caused by the load effect or floating state) may be read. Errors will then occur.

There are three function pins that share with the PA port: PA0/BZ, PA1/$\overline{BZ}$ and PA3/PFD.

The BZ and $\overline{BZ}$ are buzzer driving output pair and the PFD is a programmable frequency divider output. If the user wants to use the BZ/$\overline{BZ}$ or PFD function, the related PA port should be set as a CMOS output. The buzzer output signals are controlled by PA0 and PA1 data registers and defined in the following table.

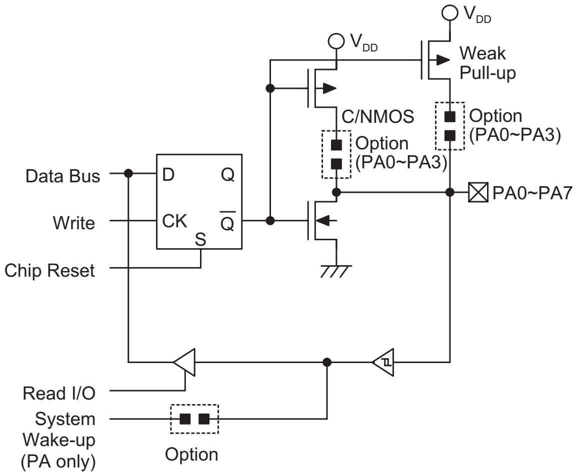
PA1 Data Register	PA0 Data Register	PA0/PA1 Pad State
0	0	PA0=BZ, PA1=$\overline{BZ}$
1	0	PA0=BZ, PA1=0
X	1	PA0=0, PA1=0

Note: ″X″ stands for unused

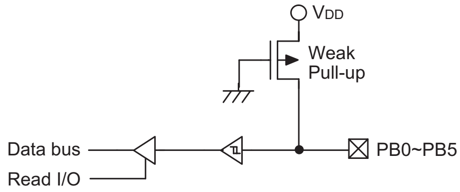
The PFD output signal function is controlled by the PA3 data register and the timer/event counter state. The PFD output signal frequency is also dependent on the timer/event counter overflow period. The definitions of PFD control signal and PFD output frequency are listed in the following table.

Timer	Timer Preload Value	PA3 Data Register	PA3 Pad State	PFD Frequency
OFF	X	0	U	X
OFF	X	1	0	X
ON	N	0	PFD	$f_{INT}/[2\times(256-N)]$
ON	N	1	0	X

Note: ″X″ stands for unused
″U″ stands for unknown



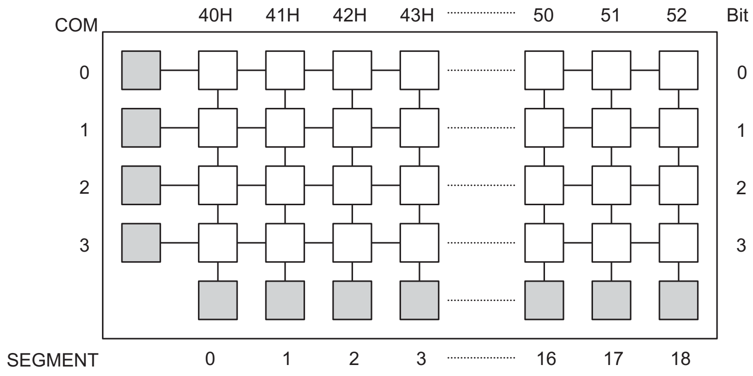
PA Input/Output Ports



PB Input Ports

LCD Display Memory

The device provides an area of embedded data memory for LCD display. This area is located from 40H to 52H of the RAM at Bank 1. Bank pointer (BP; located at 04H of the RAM) is the switch between the RAM and the LCD display memory. When the BP is set as ″01H″, any data written into 40H~52H will effect the LCD display. When the BP is cleared to ″00H″, any data written into 40H~52H means to access the general purpose data memory. The LCD display memory can be read and written to only by indirect addressing mode using MP1. When data is written into the display data area, it is automatically read by the LCD driver which then generates the corresponding LCD driving signals. To turn the display on or off, a ″1″ or a ″0″ is written to the corresponding bit of the display memory, respectively. The figure illustrates the mapping between the display memory and LCD pattern for the device.



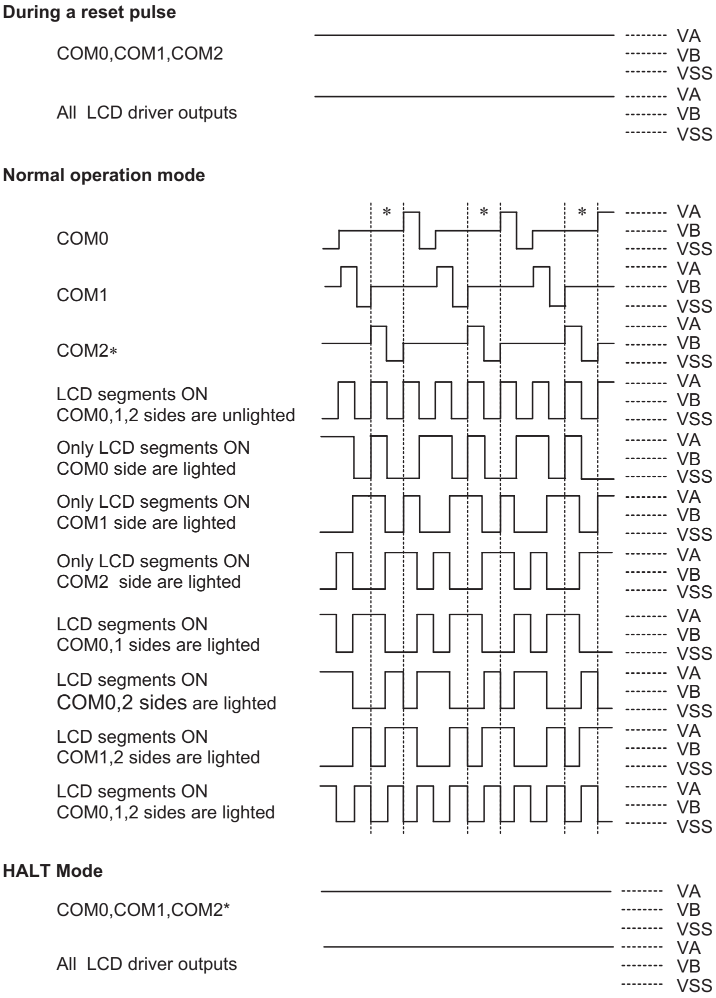
Display Memory

LCD Driver Output

The output number of the LCD driver device can be 19×2, 19×3 or 18×4 by option (i.e., 1/2 duty, 1/3 duty or 1/4 duty). The bias type LCD driver can be ″R″ type or ″C″ type for HT49R30A-1/HT49C30-1 while the bias type LCD driver can only be ″C″ type for HT49C30L. If the ″R″ bias type is selected, no external capacitor is required. If the ″C″ bias type is selected, a capacitor mounted between C1 and C2 pins is needed. The LCD driver bias voltage for HT49R30A-1/HT49C30-1 can be 1/2 bias or 1/3 bias by option, while the LCD driver bias voltage for HT49C30L can only be 1/2 bias. If 1/2 bias is selected, a capacitor mounted between V2 pin and ground is required. If 1/3 bias is selected, two capacitors are needed for V1 and V2 pins.

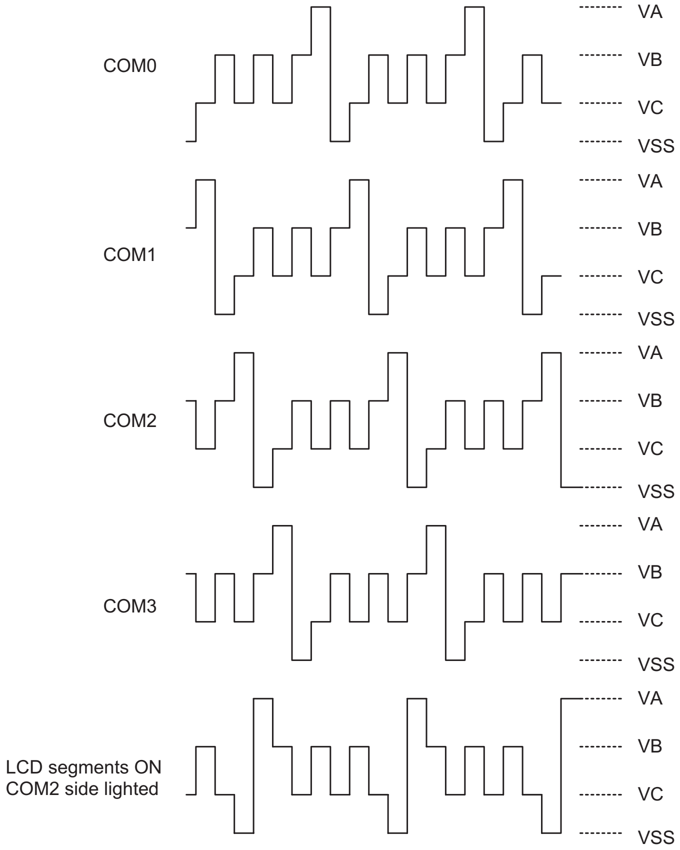
LCD bias power supply selection for HT49R30A-1/HT49C30-1: There are two types of selections: 1/2 bias or 1/3 bias.

LCD bias type selection for HT49R30A-1/HT49C30-1: This option is to determine what kind of bias is selected, R type or C type.



Note: "*" Omit the COM2 signal, if the 1/2 duty LCD is used.
VA=VLCD, VB=1/2 VLCD for HT49R30A-1/HT49C30-1
VA=2V2, VB=V2, C type for HT49C30L

LCD Driver Output (1/3 Duty, 1/2 Bias, R/C Type)



Note: 1/4 duty, 1/3 bias, C type: "VA" 3/2 VLCD, "VB" VLCD, "VC" 1/2 VLCD
1/4 duty, 1/3 bias, R type: "VA" VLCD, "VB" 2/3 VLCD, "VC" 1/3 VLCD
1/3 bias only for HT49R30A-1/HT49C30-1

LCD Driver Output

Low Voltage Reset/Detector Functions for HT49R30A-1/HT49C30-1

There is a low voltage detector (LVD) and a low voltage reset circuit (LVR) implemented in the microcontroller. These two functions can be enabled/disabled by options. Once the options of LVD is enabled, the user can use the RTCC.3 to enable/disable (1/0) the LVD circuit and read the LVD detector status (0/1) from RTCC.5; otherwise, the LVD function is disabled.

The LVR has the same effect or function with the external $\overline{\text{RES}}$ signal which performs chip reset. During HALT state, LVR is disabled.

The definitions of RTCC register are listed in the following table.

Bit No.	Label	Read/Write	Reset	Function
0~2	RT0~RT2	R/W	111B	8 to 1 multiplexer control inputs to select the real clock prescaler output
3	LVDC*	R/W	0	LVD enable/disable (1/0)
4	QOSC	R/W	0	32768Hz OSC quick start-up oscillating 0/1: quickly/slowly start
5	LVDO*	R	0	LVD detection output (1/0) 1: low voltage detected
6~7	—	—	—	Unused bit, read as ″0″

Note: ″*″ For HT49R30A-1/HT49C30-1

RTCC (09H) Register

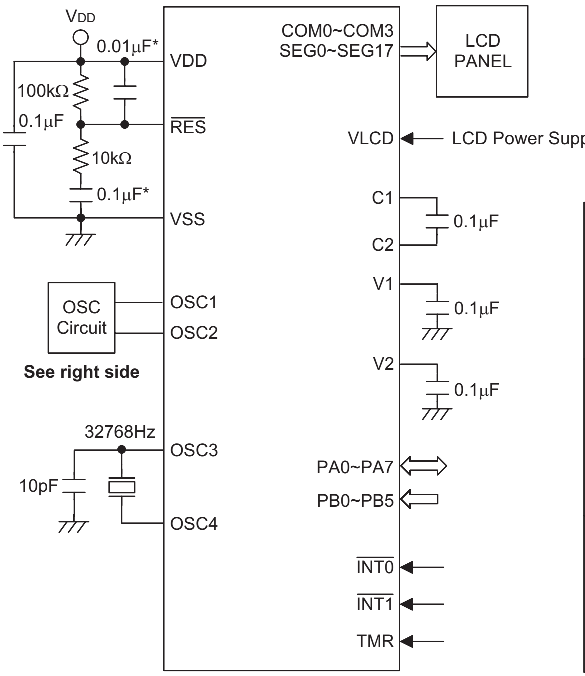
Options

The following shows the options in the device. All these options should be defined in order to ensure proper functioning system.

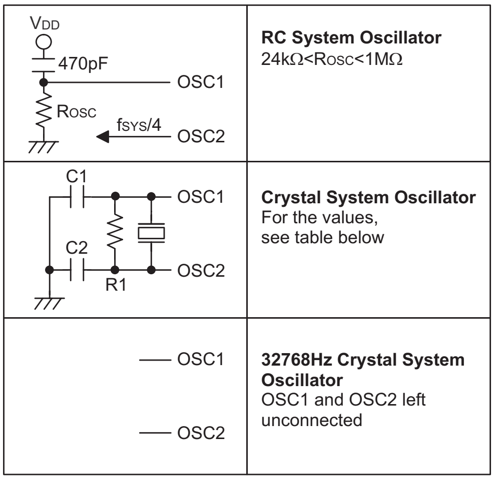
Options
OSC type selection. This option is to determine whether an RC or crystal or 32768Hz crystal oscillator is chosen as system clock.
WDT Clock source selection. RTC and Time Base. There are three types of selection: system clock/4 or RTC OSC or WDT OSC.
WDT enable/disable selection. WDT can be enabled or disabled by options.
CLR WDT times selection. This option defines how to clear the WDT by instruction. ″One time″ means that the ″CLR WDT″ can clear the WDT. ″Two times″ means that if both of the ″CLR WDT1″ and ″CLR WDT2″ have been executed, only then will the WDT be cleared.
Time Base time-out period selection. The Time Base time-out period ranges from clock/2^{12} to clock/2^{15} ″Clock″ means the clock source selected by options.
Buzzer output frequency selection. There are eight types of frequency signals for buzzer output: Clock/2^2~Clock/2^9. ″Clock″ means the clock source selected by options.
Wake-up selection. This option defines the wake-up capability. External I/O pins (PA only) all have the capability to wake-up the chip from a HALT by a falling edge.
Pull-high selection. This option is to decide whether the pull-high resistance is visible or not on the PA0~PA3. (PB and PA4~PA7 are always pull-high)
PA0~PA3 CMOS or NMOS selection. The structure of PA0~PA3 4 bits can be selected as CMOS or NMOS individually. When the CMOS is selected, the related pins only can be used for output operations. When the NMOS is selected, the related pins can be used for input or output operations. (PA4~PA7 are always NMOS)
Clock source selection of timer/event counter. There are two types of selection: system clock or system clock/4.
I/O pins share with other functions selection. PA0/$\overline{BZ}$, PA1/BZ: PA0 and PA1 can be set as I/O pins or buzzer outputs. PA3/PFD: PA3 can be set as I/O pins or PFD output.
LCD common selection. There are three types of selection: 2 common (1/2 duty) or 3 common (1/3 duty) or 4 common (1/4 duty). If the 4 common is selected, the segment output pin ″SEG18″ will be set as a common output.
LCD bias power supply selection. There are two types of selection: 1/2 bias or 1/3 bias for HT49R30A-1/HT49C30-1.
LCD bias type selection. This option is to determine what kind of bias is selected, R type or C type for HT49R30A-1/HT49C30-1, C type for HT49C30L.
LCD driver clock selection. There are seven types of frequency signals for the LCD driver circuits: $f_S/2^2$~$f_S/2^8$. ″f_S″ means the clock source selection by options.
LCD ON/OFF at HALT selection
LVR selection. LVR has enable or disable options for HT49R30A-1/HT49C30-1
LVD selection. LVD has enable or disable options for HT49R30A-1/HT49C30-1

Application Circuits

For HT49R30A-1/HT49C30-1 Application Circuit



HT49R30A-1/HT49C30-1



OSC Circuit

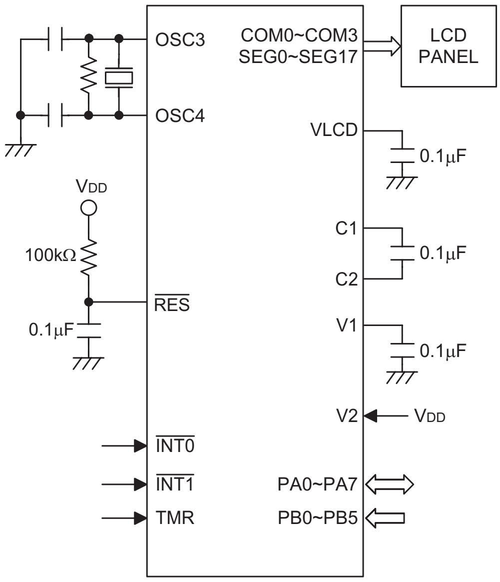
The following table shows the C1, C2 and R1 values corresponding to the different crystal values. (For reference only)

Crystal or Resonator	C1, C2	R1
4MHz Crystal	0pF	10kΩ
4MHz Resonator	10pF	12kΩ
3.58MHz Crystal	0pF	10kΩ
3.58MHz Resonator	25pF	10kΩ
2MHz Crystal & Resonator	25pF	10kΩ
1MHz Crystal	35pF	27kΩ
480kHz Resonator	300pF	9.1kΩ
455kHz Resonator	300pF	10kΩ
429kHz Resonator	300pF	10kΩ
The function of the resistor R1 is to ensure that the oscillator will switch off should low voltage conditions occur. Such a low voltage, as mentioned here, is one which is less than the lowest value of the MCU operating voltage. Note however that if the LVR is enabled then R1 can be removed.		

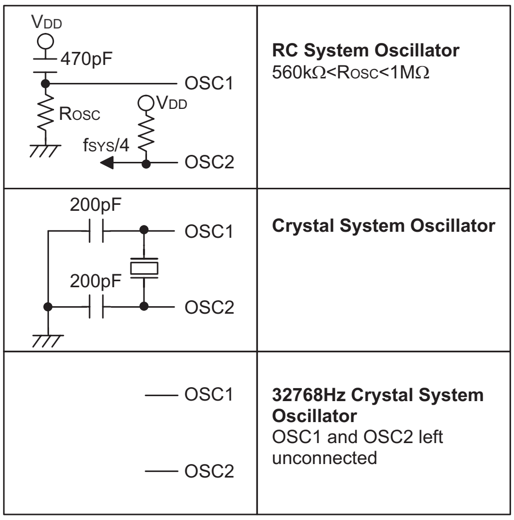
Note: The resistance and capacitance for reset circuit should be designed in such a way as to ensure that the VDD is stable and remains within a valid operating voltage range before bringing $\overline{\text{RES}}$ to high.

"*" Make the length of the wiring, which is connected to the $\overline{\text{RES}}$ pin as short as possible, to avoid noise interference.

For HT49C30L Application Circuit



HT49C30L



OSC Circuit

Note: The resistance and capacitance for reset circuit should be designed in such a way as to ensure that the VDD is stable and remains within a valid operating voltage range before bringing $\overline{\text{RES}}$ to high.

Instruction Set Summary

Mnemonic	Description	Instruction Cycle	Flag Affected
Arithmetic			
ADD A,[m]	Add data memory to ACC	1	Z,C,AC,OV
ADDM A,[m]	Add ACC to data memory	1(1)	Z,C,AC,OV
ADD A,x	Add immediate data to ACC	1	Z,C,AC,OV
ADC A,[m]	Add data memory to ACC with carry	1	Z,C,AC,OV
ADCM A,[m]	Add ACC to data memory with carry	1(1)	Z,C,AC,OV
SUB A,x	Subtract immediate data from ACC	1	Z,C,AC,OV
SUB A,[m]	Subtract data memory from ACC	1	Z,C,AC,OV
SUBM A,[m]	Subtract data memory from ACC with result in data memory	1(1)	Z,C,AC,OV
SBC A,[m]	Subtract data memory from ACC with carry	1	Z,C,AC,OV
SBCM A,[m]	Subtract data memory from ACC with carry and result in data memory	1(1)	Z,C,AC,OV
DAA [m]	Decimal adjust ACC for addition with result in data memory	1(1)	C
Logic Operation			
AND A,[m]	AND data memory to ACC	1	Z
OR A,[m]	OR data memory to ACC	1	Z
XOR A,[m]	Exclusive-OR data memory to ACC	1	Z
ANDM A,[m]	AND ACC to data memory	1(1)	Z
ORM A,[m]	OR ACC to data memory	1(1)	Z
XORM A,[m]	Exclusive-OR ACC to data memory	1(1)	Z
AND A,x	AND immediate data to ACC	1	Z
OR A,x	OR immediate data to ACC	1	Z
XOR A,x	Exclusive-OR immediate data to ACC	1	Z
CPL [m]	Complement data memory	1(1)	Z
CPLA [m]	Complement data memory with result in ACC	1	Z
Increment & Decrement			
INCA [m]	Increment data memory with result in ACC	1	Z
INC [m]	Increment data memory	1(1)	Z
DECA [m]	Decrement data memory with result in ACC	1	Z
DEC [m]	Decrement data memory	1(1)	Z
Rotate			
RRA [m]	Rotate data memory right with result in ACC	1	None
RR [m]	Rotate data memory right	1(1)	None
RRCA [m]	Rotate data memory right through carry with result in ACC	1	C
RRC [m]	Rotate data memory right through carry	1(1)	C
RLA [m]	Rotate data memory left with result in ACC	1	None
RL [m]	Rotate data memory left	1(1)	None
RLCA [m]	Rotate data memory left through carry with result in ACC	1	C
RLC [m]	Rotate data memory left through carry	1(1)	C
Data Move			
MOV A,[m]	Move data memory to ACC	1	None
MOV [m],A	Move ACC to data memory	1(1)	None
MOV A,x	Move immediate data to ACC	1	None
Bit Operation			
CLR [m].i	Clear bit of data memory	1(1)	None
SET [m].i	Set bit of data memory	1(1)	None

Mnemonic	Description	Instruction Cycle	Flag Affected
Branch			
JMP addr	Jump unconditionally	2	None
SZ [m]	Skip if data memory is zero	$1^{(2)}$	None
SZA [m]	Skip if data memory is zero with data movement to ACC	$1^{(2)}$	None
SZ [m].i	Skip if bit i of data memory is zero	$1^{(2)}$	None
SNZ [m].i	Skip if bit i of data memory is not zero	$1^{(2)}$	None
SIZ [m]	Skip if increment data memory is zero	$1^{(3)}$	None
SDZ [m]	Skip if decrement data memory is zero	$1^{(3)}$	None
SIZA [m]	Skip if increment data memory is zero with result in ACC	$1^{(2)}$	None
SDZA [m]	Skip if decrement data memory is zero with result in ACC	$1^{(2)}$	None
CALL addr	Subroutine call	2	None
RET	Return from subroutine	2	None
RET A,x	Return from subroutine and load immediate data to ACC	2	None
RETI	Return from interrupt	2	None
Table Read			
TABRDC [m]	Read ROM code (current page) to data memory and TBLH	$2^{(1)}$	None
TABRDL [m]	Read ROM code (last page) to data memory and TBLH	$2^{(1)}$	None
Miscellaneous			
NOP	No operation	1	None
CLR [m]	Clear data memory	$1^{(1)}$	None
SET [m]	Set data memory	$1^{(1)}$	None
CLR WDT	Clear Watchdog Timer	1	TO,PDF
CLR WDT1	Pre-clear Watchdog Timer	1	$TO^{(4)}$,$PDF^{(4)}$
CLR WDT2	Pre-clear Watchdog Timer	1	$TO^{(4)}$,$PDF^{(4)}$
SWAP [m]	Swap nibbles of data memory	$1^{(1)}$	None
SWAPA [m]	Swap nibbles of data memory with result in ACC	1	None
HALT	Enter power down mode	1	TO,PDF

Note: x: Immediate data

m: Data memory address

A: Accumulator

i: 0~7 number of bits

addr: Program memory address

√: Flag is affected

–: Flag is not affected

(1): If a loading to the PCL register occurs, the execution cycle of instructions will be delayed for one more cycle (four system clocks).

(2): If a skipping to the next instruction occurs, the execution cycle of instructions will be delayed for one more cycle (four system clocks). Otherwise the original instruction cycle is unchanged.

(3): (1) and (2)

(4): The flags may be affected by the execution status. If the Watchdog Timer is cleared by executing the CLR WDT1 or CLR WDT2 instruction, the TO and PDF are cleared.
Otherwise the TO and PDF flags remain unchanged.

Instruction Definition

ADC A,[m]	Add data memory and carry to the accumulator
Description	The contents of the specified data memory, accumulator and the carry flag are added simultaneously, leaving the result in the accumulator.
Operation	ACC ← ACC+[m]+C
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

ADCM A,[m]	Add the accumulator and carry to data memory
Description	The contents of the specified data memory, accumulator and the carry flag are added simultaneously, leaving the result in the specified data memory.
Operation	[m] ← ACC+[m]+C
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

ADD A,[m]	Add data memory to the accumulator
Description	The contents of the specified data memory and the accumulator are added. The result is stored in the accumulator.
Operation	ACC ← ACC+[m]
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

ADD A,x	Add immediate data to the accumulator
Description	The contents of the accumulator and the specified data are added, leaving the result in the accumulator.
Operation	ACC ← ACC+x
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

ADDM A,[m]	Add the accumulator to the data memory
Description	The contents of the specified data memory and the accumulator are added. The result is stored in the data memory.
Operation	[m] ← ACC+[m]
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

AND A,[m]	Logical AND accumulator with data memory
Description	Data in the accumulator and the specified data memory perform a bitwise logical_AND operation. The result is stored in the accumulator.
Operation	ACC ← ACC ″AND″ [m]
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

AND A,x	Logical AND immediate data to the accumulator
Description	Data in the accumulator and the specified data perform a bitwise logical_AND operation. The result is stored in the accumulator.
Operation	ACC ← ACC ″AND″ x
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

ANDM A,[m]	Logical AND data memory with the accumulator
Description	Data in the specified data memory and the accumulator perform a bitwise logical_AND operation. The result is stored in the data memory.
Operation	[m] ← ACC ″AND″ [m]
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

CALL addr	Subroutine call
Description	The instruction unconditionally calls a subroutine located at the indicated address. The program counter increments once to obtain the address of the next instruction, and pushes this onto the stack. The indicated address is then loaded. Program execution continues with the instruction at this address.
Operation	Stack ← Program Counter+1 Program Counter ← addr
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

CLR [m]	Clear data memory
Description	The contents of the specified data memory are cleared to 0.
Operation	[m] ← 00H
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

CLR [m].i	Clear bit of data memory
Description	The bit i of the specified data memory is cleared to 0.
Operation	[m].i ← 0
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

CLR WDT	Clear Watchdog Timer
Description	The WDT is cleared (clears the WDT). The power down bit (PDF) and time-out bit (TO) are cleared.
Operation	WDT ← 00H PDF and TO ← 0
Affected flag(s)	

TO	PDF	OV	Z	AC	C
0	0	—	—	—	—

CLR WDT1	Preclear Watchdog Timer
Description	Together with CLR WDT2, clears the WDT. PDF and TO are also cleared. Only execution of this instruction without the other preclear instruction just sets the indicated flag which implies this instruction has been executed and the TO and PDF flags remain unchanged.
Operation	WDT ← 00H* PDF and TO ← 0*
Affected flag(s)	

TO	PDF	OV	Z	AC	C
0*	0*	—	—	—	—

CLR WDT2	Preclear Watchdog Timer
Description	Together with CLR WDT1, clears the WDT. PDF and TO are also cleared. Only execution of this instruction without the other preclear instruction, sets the indicated flag which implies this instruction has been executed and the TO and PDF flags remain unchanged.
Operation	WDT ← 00H* PDF and TO ← 0*
Affected flag(s)	

TO	PDF	OV	Z	AC	C
0*	0*	—	—	—	—

CPL [m]	Complement data memory
Description	Each pin of the specified data memory is logically complemented (1′s complement). Bits which previously contained a 1 are changed to 0 and vice-versa.
Operation	[m] ← $\overline{[m]}$
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

CPLA [m] Complement data memory and place result in the accumulator

Description: Each pin of the specified data memory is logically complemented (1′s complement). Bits which previously contained a 1 are changed to 0 and vice-versa. The complemented result is stored in the accumulator and the contents of the data memory remain unchanged.

Operation: ACC ← $\overline{[m]}$

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

DAA [m] Decimal-Adjust accumulator for addition

Description: The accumulator value is adjusted to the BCD (Binary Coded Decimal) code. The accumulator is divided into two nibbles. Each nibble is adjusted to the BCD code and an internal carry (AC1) will be done if the low nibble of the accumulator is greater than 9. The BCD adjustment is done by adding 6 to the original value if the original value is greater than 9 or a carry (AC or C) is set; otherwise the original value remains unchanged. The result is stored in the data memory and only the carry flag (C) may be affected.

Operation:

If ACC.3~ACC.0 >9 or AC=1
then [m].3~[m].0 ← (ACC.3~ACC.0)+6, AC1=$\overline{AC}$
else [m].3~[m].0 ← (ACC.3~ACC.0), AC1=0
and
If ACC.7~ACC.4+AC1 >9 or C=1
then [m].7~[m].4 ← ACC.7~ACC.4+6+AC1,C=1
else [m].7~[m].4 ← ACC.7~ACC.4+AC1,C=C

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

DEC [m] Decrement data memory

Description: Data in the specified data memory is decremented by 1.

Operation: [m] ← [m]–1

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

DECA [m] Decrement data memory and place result in the accumulator

Description: Data in the specified data memory is decremented by 1, leaving the result in the accumulator. The contents of the data memory remain unchanged.

Operation: ACC ← [m]–1

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

HALT Enter power down mode

Description This instruction stops program execution and turns off the system clock. The contents of the RAM and registers are retained. The WDT and prescaler are cleared. The power down bit (PDF) is set and the WDT time-out bit (TO) is cleared.

Operation Program Counter ← Program Counter+1
PDF ← 1
TO ← 0

Affected flag(s)

TO	PDF	OV	Z	AC	C
0	1	—	—	—	—

INC [m] Increment data memory

Description Data in the specified data memory is incremented by 1

Operation [m] ← [m]+1

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

INCA [m] Increment data memory and place result in the accumulator

Description Data in the specified data memory is incremented by 1, leaving the result in the accumulator. The contents of the data memory remain unchanged.

Operation ACC ← [m]+1

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

JMP addr Directly jump

Description The program counter are replaced with the directly-specified address unconditionally, and control is passed to this destination.

Operation Program Counter ←addr

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

MOV A,[m] Move data memory to the accumulator

Description The contents of the specified data memory are copied to the accumulator.

Operation ACC ← [m]

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

MOV A,x Move immediate data to the accumulator

Description The 8-bit data specified by the code is loaded into the accumulator.

Operation ACC ← x

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

MOV [m],A Move the accumulator to data memory

Description The contents of the accumulator are copied to the specified data memory (one of the data memories).

Operation [m] ←ACC

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

NOP No operation

Description No operation is performed. Execution continues with the next instruction.

Operation Program Counter ← Program Counter+1

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

OR A,[m] Logical OR accumulator with data memory

Description Data in the accumulator and the specified data memory (one of the data memories) perform a bitwise logical_OR operation. The result is stored in the accumulator.

Operation ACC ← ACC ″OR″ [m]

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

OR A,x Logical OR immediate data to the accumulator

Description Data in the accumulator and the specified data perform a bitwise logical_OR operation. The result is stored in the accumulator.

Operation ACC ← ACC ″OR″ x

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

ORM A,[m] Logical OR data memory with the accumulator

Description Data in the data memory (one of the data memories) and the accumulator perform a bitwise logical_OR operation. The result is stored in the data memory.

Operation [m] ←ACC ″OR″ [m]

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

RET Return from subroutine

Description The program counter is restored from the stack. This is a 2-cycle instruction.

Operation Program Counter ← Stack

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RET A,x Return and place immediate data in the accumulator

Description The program counter is restored from the stack and the accumulator loaded with the specified 8-bit immediate data.

Operation Program Counter ← Stack
ACC ← x

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RETI Return from interrupt

Description The program counter is restored from the stack, and interrupts are enabled by setting the EMI bit. EMI is the enable master (global) interrupt bit.

Operation Program Counter ← Stack
EMI ← 1

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RL [m] Rotate data memory left

Description The contents of the specified data memory are rotated 1 bit left with bit 7 rotated into bit 0.

Operation [m].(i+1) ← [m].i; [m].i:bit i of the data memory (i=0~6)
[m].0 ← [m].7

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RLA [m] Rotate data memory left and place result in the accumulator

Description Data in the specified data memory is rotated 1 bit left with bit 7 rotated into bit 0, leaving the rotated result in the accumulator. The contents of the data memory remain unchanged.

Operation ACC.(i+1) ← [m].i; [m].i:bit i of the data memory (i=0~6)
ACC.0 ← [m].7

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RLC [m] Rotate data memory left through carry

Description: The contents of the specified data memory and the carry flag are rotated 1 bit left. Bit 7 replaces the carry bit; the original carry flag is rotated into the bit 0 position.

Operation:
[m].(i+1) ← [m].i; [m].i:bit i of the data memory (i=0~6)
[m].0 ← C
C ← [m].7

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

RLCA [m] Rotate left through carry and place result in the accumulator

Description: Data in the specified data memory and the carry flag are rotated 1 bit left. Bit 7 replaces the carry bit and the original carry flag is rotated into bit 0 position. The rotated result is stored in the accumulator but the contents of the data memory remain unchanged.

Operation:
ACC.(i+1) ← [m].i; [m].i:bit i of the data memory (i=0~6)
ACC.0 ← C
C ← [m].7

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

RR [m] Rotate data memory right

Description: The contents of the specified data memory are rotated 1 bit right with bit 0 rotated to bit 7.

Operation:
[m].i ← [m].(i+1); [m].i:bit i of the data memory (i=0~6)
[m].7 ← [m].0

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RRA [m] Rotate right and place result in the accumulator

Description: Data in the specified data memory is rotated 1 bit right with bit 0 rotated into bit 7, leaving the rotated result in the accumulator. The contents of the data memory remain unchanged.

Operation:
ACC.(i) ← [m].(i+1); [m].i:bit i of the data memory (i=0~6)
ACC.7 ← [m].0

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RRC [m] Rotate data memory right through carry

Description: The contents of the specified data memory and the carry flag are together rotated 1 bit right. Bit 0 replaces the carry bit; the original carry flag is rotated into the bit 7 position.

Operation:
[m].i ← [m].(i+1); [m].i:bit i of the data memory (i=0~6)
[m].7 ← C
C ← [m].0

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

RRCA [m] Rotate right through carry and place result in the accumulator

Description: Data of the specified data memory and the carry flag are rotated 1 bit right. Bit 0 replaces the carry bit and the original carry flag is rotated into the bit 7 position. The rotated result is stored in the accumulator. The contents of the data memory remain unchanged.

Operation:
ACC.i ← [m].(i+1); [m].i:bit i of the data memory (i=0~6)
ACC.7 ← C
C ← [m].0

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

SBC A,[m] Subtract data memory and carry from the accumulator

Description: The contents of the specified data memory and the complement of the carry flag are subtracted from the accumulator, leaving the result in the accumulator.

Operation: ACC ← ACC+$[\overline{m}]$+C

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SBCM A,[m] Subtract data memory and carry from the accumulator

Description: The contents of the specified data memory and the complement of the carry flag are subtracted from the accumulator, leaving the result in the data memory.

Operation: [m] ← ACC+$[\overline{m}]$+C

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SDZ [m] Skip if decrement data memory is 0

Description: The contents of the specified data memory are decremented by 1. If the result is 0, the next instruction is skipped. If the result is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).

Operation: Skip if ([m]–1)=0, [m] ← ([m]–1)

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SDZA [m] Decrement data memory and place result in ACC, skip if 0

Description: The contents of the specified data memory are decremented by 1. If the result is 0, the next instruction is skipped. The result is stored in the accumulator but the data memory remains unchanged. If the result is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).

Operation: Skip if ([m]–1)=0, ACC ← ([m]–1)

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SET [m]	Set data memory
Description	Each pin of the specified data memory is set to 1.
Operation	[m] ← FFH
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SET [m]. i	Set bit of data memory
Description	Bit i of the specified data memory is set to 1.
Operation	[m].i ← 1
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SIZ [m]	Skip if increment data memory is 0
Description	The contents of the specified data memory are incremented by 1. If the result is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).
Operation	Skip if ([m]+1)=0, [m] ← ([m]+1)
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SIZA [m]	Increment data memory and place result in ACC, skip if 0
Description	The contents of the specified data memory are incremented by 1. If the result is 0, the next instruction is skipped and the result is stored in the accumulator. The data memory remains unchanged. If the result is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).
Operation	Skip if ([m]+1)=0, ACC ← ([m]+1)
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SNZ [m].i	Skip if bit i of the data memory is not 0
Description	If bit i of the specified data memory is not 0, the next instruction is skipped. If bit i of the data memory is not 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).
Operation	Skip if [m].i≠0
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SUB A,[m]	Subtract data memory from the accumulator
Description	The specified data memory is subtracted from the contents of the accumulator, leaving the result in the accumulator.
Operation	ACC ← ACC+$\overline{[m]}$+1
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SUBM A,[m]	Subtract data memory from the accumulator
Description	The specified data memory is subtracted from the contents of the accumulator, leaving the result in the data memory.
Operation	[m] ← ACC+$\overline{[m]}$+1
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SUB A,x	Subtract immediate data from the accumulator
Description	The immediate data specified by the code is subtracted from the contents of the accumulator, leaving the result in the accumulator.
Operation	ACC ← ACC+$\overline{x}$+1
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SWAP [m]	Swap nibbles within the data memory
Description	The low-order and high-order nibbles of the specified data memory (1 of the data memories) are interchanged.
Operation	[m].3~[m].0 ↔ [m].7~[m].4
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SWAPA [m]	Swap data memory and place result in the accumulator
Description	The low-order and high-order nibbles of the specified data memory are interchanged, writing the result to the accumulator. The contents of the data memory remain unchanged.
Operation	ACC.3~ACC.0 ← [m].7~[m].4 ACC.7~ACC.4 ← [m].3~[m].0
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SZ [m]	Skip if data memory is 0
Description	If the contents of the specified data memory are 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).
Operation	Skip if [m]=0
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SZA [m]	Move data memory to ACC, skip if 0
Description	The contents of the specified data memory are copied to the accumulator. If the contents is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).
Operation	Skip if [m]=0
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SZ [m].i	Skip if bit i of the data memory is 0
Description	If bit i of the specified data memory is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).
Operation	Skip if [m].i=0
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

TABRDC [m]	Move the ROM code (current page) to TBLH and data memory
Description	The low byte of ROM code (current page) addressed by the table pointer (TBLP) is moved to the specified data memory and the high byte transferred to TBLH directly.
Operation	[m] ← ROM code (low byte) TBLH ← ROM code (high byte)
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

TABRDL [m]	Move the ROM code (last page) to TBLH and data memory
Description	The low byte of ROM code (last page) addressed by the table pointer (TBLP) is moved to the data memory and the high byte transferred to TBLH directly.
Operation	[m] ← ROM code (low byte) TBLH ← ROM code (high byte)
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

XOR A,[m]	Logical XOR accumulator with data memory
Description	Data in the accumulator and the indicated data memory perform a bitwise logical Exclusive_OR operation and the result is stored in the accumulator.
Operation	ACC ← ACC ″XOR″ [m]
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

XORM A,[m]	Logical XOR data memory with the accumulator
Description	Data in the indicated data memory and the accumulator perform a bitwise logical Exclusive_OR operation. The result is stored in the data memory. The 0 flag is affected.
Operation	[m] ← ACC ″XOR″ [m]
Affected flag(s)	

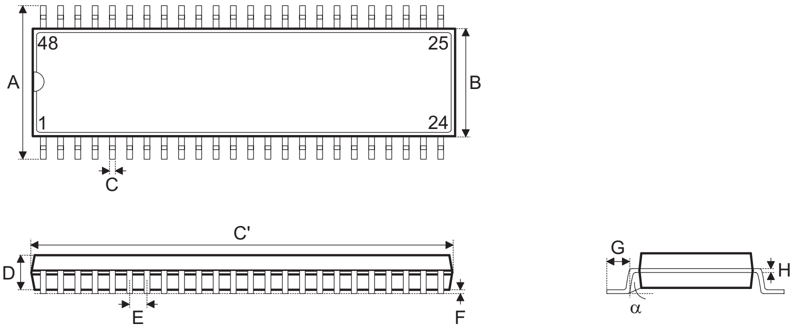
TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

XOR A,x	Logical XOR immediate data to the accumulator
Description	Data in the accumulator and the specified data perform a bitwise logical Exclusive_OR operation. The result is stored in the accumulator. The 0 flag is affected.
Operation	ACC ← ACC ″XOR″ x
Affected flag(s)	

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

Package Information

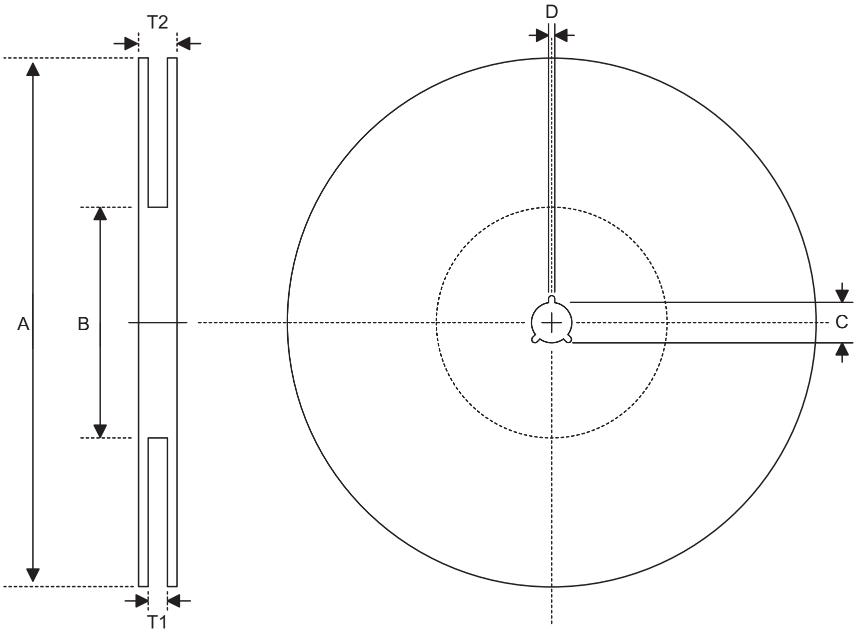
48-pin SSOP (300mil) Outline Dimensions



<table>
<tr><th rowspan="2">Symbol</th><th colspan="3">Dimensions in mil</th></tr>
<tr><th>Min.</th><th>Nom.</th><th>Max.</th></tr>
<tr><td>A</td><td>395</td><td>—</td><td>420</td></tr>
<tr><td>B</td><td>291</td><td>—</td><td>299</td></tr>
<tr><td>C</td><td>8</td><td>—</td><td>12</td></tr>
<tr><td>C′</td><td>613</td><td>—</td><td>637</td></tr>
<tr><td>D</td><td>85</td><td>—</td><td>99</td></tr>
<tr><td>E</td><td>—</td><td>25</td><td>—</td></tr>
<tr><td>F</td><td>4</td><td>—</td><td>10</td></tr>
<tr><td>G</td><td>25</td><td>—</td><td>35</td></tr>
<tr><td>H</td><td>4</td><td>—</td><td>12</td></tr>
<tr><td>α</td><td>0°</td><td>—</td><td>8°</td></tr>
</table>

Product Tape and Reel Specifications

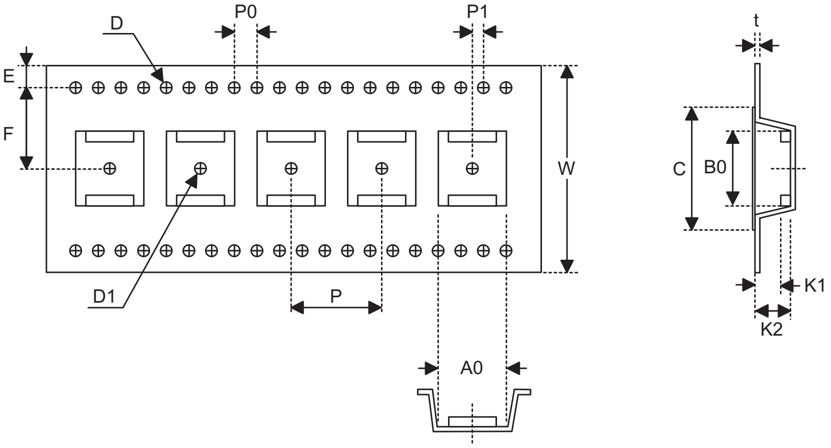
Reel Dimensions



SSOP 48W

Symbol	Description	Dimensions in mm
A	Reel Outer Diameter	330±1.0
B	Reel Inner Diameter	100±0.1
C	Spindle Hole Diameter	13.0+0.5 −0.2
D	Key Slit Width	2.0±0.5
T1	Space Between Flange	32.2+0.3 −0.2
T2	Reel Thickness	38.2±0.2

Carrier Tape Dimensions



SSOP 48W

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	32.0±0.3
P	Cavity Pitch	16.0±0.1
E	Perforation Position	1.75±0.1
F	Cavity to Perforation (Width Direction)	14.2±0.1
D	Perforation Diameter	2.0 Min.
D1	Cavity Hole Diameter	1.5+0.25
P0	Perforation Pitch	4.0±0.1
P1	Cavity to Perforation (Length Direction)	2.0±0.1
A0	Cavity Length	12.0±0.1
B0	Cavity Width	16.20±0.1
K1	Cavity Depth	2.4±0.1
K2	Cavity Depth	3.2±0.1
t	Carrier Tape Thickness	0.35±0.05
C	Cover Tape Width	25.5

Holtek Semiconductor Inc. (Headquarters)
No.3, Creation Rd. II, Science Park, Hsinchu, Taiwan
Tel: 886-3-563-1999
Fax: 886-3-563-1189
http://www.holtek.com.tw

Holtek Semiconductor Inc. (Taipei Sales Office)
4F-2, No. 3-2, YuanQu St., Nankang Software Park, Taipei 115, Taiwan
Tel: 886-2-2655-7070
Fax: 886-2-2655-7373
Fax: 886-2-2655-7383 (International sales hotline)

Holtek Semiconductor Inc. (Shanghai Sales Office)
7th Floor, Building 2, No.889, Yi Shan Rd., Shanghai, China 200233
Tel: 021-6485-5560
Fax: 021-6485-0313
http://www.holtek.com.cn

Holtek Semiconductor Inc. (Shenzhen Sales Office)
5/F, Unit A, Productivity Building, Cross of Science M 3rd Road and Gaoxin M 2nd Road, Science Park, Nanshan District,
Shenzhen, China 518057
Tel: 0755-8616-9908, 8616-9308
Fax: 0755-8616-9533

Holtek Semiconductor Inc. (Beijing Sales Office)
Suite 1721, Jinyu Tower, A129 West Xuan Wu Men Street, Xicheng District, Beijing, China 100031
Tel: 010-6641-0030, 6641-7751, 6641-7752
Fax: 010-6641-0125

Holtek Semiconductor Inc. (Chengdu Sales Office)
709, Building 3, Champagne Plaza, No.97 Dongda Street, Chengdu, Sichuan, China 610016
Tel: 028-6653-6590
Fax: 028-6653-6591

Holmate Semiconductor, Inc. (North America Sales Office)
46729 Fremont Blvd., Fremont, CA 94538
Tel: 510-252-9880
Fax: 510-252-9885
http://www.holmate.com