

HT48RA0-1/HT48CA0-1 Remote Type 8-Bit MCU

Technical Document

- Tools Information
- FAQs
- Application Note
- HA0016E Writing and Reading to the HT24 EEPROM with the HT48 MCU Series
- HA0018E Controlling the HT1621 LCD Controller with the HT48 MCU Series
- HA0041E Using the HT48CA0 to Generate the HT6221 Output Signals
- HA0075E MCU Reset and Oscillator Circuits Application Note
- HA0076E HT48RAx/HT48CAx Software Application Note
- HA0082E HT48xA0-1 and HT48xA0-2 Power-on Reset Timing

Features

- Operating voltage: 2.0V~3.6V
- Ten bidirectional I/O lines
- Six Schmitt trigger input lines
- One carrier output (1/2 or 1/3 duty)
- On-chip crystal and RC oscillator
- Watchdog Timer
- 1K×14 program memory
- 32×8 data RAM
- HALT function and wake-up feature reduce power consumption

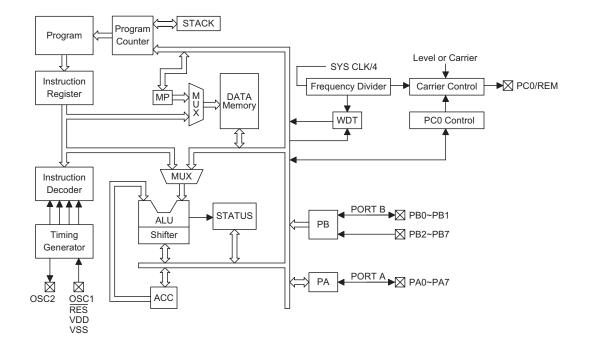
- 62 powerful instructions
- Up to $1\mu s$ instruction cycle with 4MHz system clock
- All instructions in 1 or 2 machine cycles
- 14-bit table read instructions
- One-level subroutine nesting
- Bit manipulation instructions
- Low voltage reset function
- 24-pin SOP/SSOP packages

General Description

The HT48RA0-1/HT48CA0-1 are 8-bit high performance, RISC architecture microcontroller devices specifically designed for multiple I/O control product applications. The mask version HT48CA0-1 is fully pin and functionally compatible with the OTP version HT48RA0-1 device. The advantages of low power consumption, I/O flexibility, timer functions, oscillator options, watchdog timer, HALT and wake-up functions, as well as low cost, enhance the versatility of this device to suit a wide range of application possibilities such as industrial control, consumer products, and particularly suitable for use in products such as infrared remote controllers and various subsystem controllers.



Block Diagram



Pin Assignment

			1	
PA1	1	24	Þ	PA2
PA0	2	23	Þ	PA3
PB1	3	22	Þ	PA4
PB0	4	21	Þ	PA5
PC0/REM	5	20	Þ	PA6
	6	19	Þ	PA7
OSC2	7	18	Þ	PB2
OSC1	8	17	Þ	PB3
VSS	9	16	Þ	PB4
RES	10	15	Þ	PB5
NC	11	14	Þ	PB6
NC	12	13	Þ	PB7
HT48	RA0-1/H	T48C	;A()-1
- 24	SOP-A/	sso	P-/	4



Pin Description

Pin Name	I/O	Code Option	Description
PA0~PA7	I/O	_	Bidirectional 8-bit input/output port with pull-high resistors. Each bit can be de- termined as NMOS output or Schmitt trigger input by software instructions.
PB0, PB1	I/O	Wake-up or None	2-bit bidirectional input/output lines with pull-high resistors. Each bit can be de- termined as NMOS output or Schmitt trigger input by software instructions. Each bit can also be configured as wake-up input by code option.
PB2~PB7	I	Wake-up or None	6-bit Schmitt trigger input lines with pull-high resistors. Each bit can be config- ured as a wake-up input by code option.
PC0/REM	ο	Level or Carrier	Level or carrier output pin PC0 can be set as CMOS output pin or carrier output pin by code option.
VDD	_	_	Positive power supply
VSS	_		Negative power supply, ground
OSC2 OSC1	0 1	Crystal or RC	OSC1, OSC2 are connected to an RC network or a crystal (determined by code option) for the internal system clock. In the case of RC operation, OSC2 is the output terminal for 1/4 system clock (NMOS open drain output).
RES	I		Schmitt trigger reset input. Active low.

Absolute Maximum Ratings

Supply VoltageV_SS=0.3V to V_SS+4.0V	Storage Temperature50°C to 125°C
Input VoltageV _{SS} –0.3V to V_{DD} +0.3V	Operating Temperature40°C to 85°C

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.

D.C. Characteristics

0	Demonster		Test Conditions		Ŧ		11
Symbol	Parameter	V_{DD}	Conditions	Min.	Тур.	Max.	Unit
V_{DD}	Operating Voltage			2.0		3.6	V
I _{DD}	Operating Current	3V	No load, f _{SYS} =4MHz	—	0.7	1.5	mA
I _{STB}	Standby Current	3V	No load, system HALT	_		1	μA
V_{IL1}	Input Low Voltage for I/O Ports	3V		0		$0.3V_{DD}$	V
V_{IH1}	Input High Voltage for I/O Ports	3V	_	$0.7 V_{DD}$		V _{DD}	V
V_{IL2}	Input Low Voltage (RES)	3V		0		$0.4V_{DD}$	V
V_{IH2}	Input High Voltage (RES)	3V		$0.9V_{DD}$		V _{DD}	V
V_{LVR}	Low Voltage Reset Voltage			_	1.9	2.0	V
I _{OL}	I/O Ports Sink Current	3V	V _{OL} =0.1V _{DD}	4	8	_	mA
I _{OH}	PC0/REM Output Source Current	3V	V _{OH} =0.9V _{DD}	-2	-4	_	mA
R _{PH}	Pull-high Resistance	3V	_	20	60	100	kΩ

Ta=25°C



A.C. Characteristics

Ta=25°C

Complete L	Demonster		Test Conditions	Min	T	Marr	11
Symbol	ol Parameter		Conditions	Min.	Тур.	Max.	Unit
f _{SYS}	System Clock	3V		400		4000	kHz
t _{RES}	External Reset Low Pulse Width	_		1		_	μs
t _{SST}	System Start-up Timer Period	_	Power-up, reset or wake-up from HALT		1024		t _{sys}
t _{LVR}	Low Voltage Width to Reset	_		1		_	ms

Note: t_{SYS}=1/f_{SYS}

Functional Description

Execution Flow

The HT48RA0-1/HT48CA0-1 system clock can be derived from a crystal/ceramic resonator 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. However, the pipelining scheme causes each instruction to effectively execute within one cycle. If an instruction changes the program counter, two cycles are required to complete the instruction.

Program Counter – PC

The 10-bit program counter (PC) controls the sequence in which the instructions stored in program ROM are executed and its contents specify a maximum of 1024 addresses.

After accessing a program memory word to fetch an instruction code, the contents of the program counter are incremented by one. The program counter then points to the memory word containing the next instruction code.

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

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

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

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

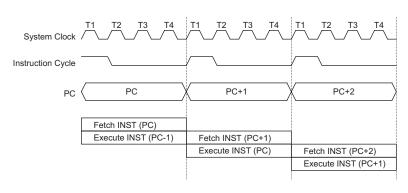
Program Memory – ROM

The program memory is used to store the program instructions which are to be executed. It also contains data and table and is organized into 1024×14 bits, addressed by the program counter and table pointer.

Certain locations in the program memory are reserved for special usage:

Location 000H

This area is reserved for the initialization program. After chip reset, the program always begins execution at location 000H.

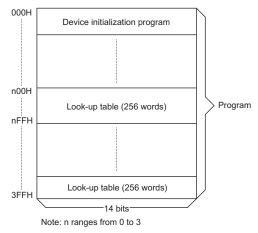


Execution Flow



Table location

Any location in the EPROM space can be used as look-up tables. The instructions TABRDC [m] (the current page, one page=256 words) and TABRDL [m] (the last page) transfer the contents of the lower-order byte to the specified data memory, and the higher-order byte to TBLH (08H). Only the destination of the lower-order byte in the table is well-defined, the other bits of the table word are transferred to the lower portion of TBLH, the remaining 2 bits are read as "0". The Table Higher-order byte register (TBLH) is read only. The table pointer (TBLP) is a read/write register (07H), where P indicates the table location. Before accessing the table, the location must be placed in TBLP. The TBLH is read only and cannot be restored.



Program Memory

All table related instructions need 2 cycles to complete the operation. These areas may function as normal program memory depending upon the requirements.

Stack Register – STACK

This is a special part of the memory used to save the contents of the program counter (PC) only. The stack is organized into one level and is neither part of the data nor part of the program space, and is neither readable nor writeable. The activated level is indexed by the stack pointer (SP) and is neither readable nor writeable. At a subroutine call the contents of the program counter are pushed onto the stack. At the end of a subroutine signaled by a return instruction (RET), the program counter is restored to its previous value from the stack. After a chip reset, the SP will point to the top of the stack.

If the stack is full and a "CALL" is subsequently executed, stack overflow occurs and the first entry will be lost (only the most recent return address is stored).

Data Memory – RAM

The data memory is designed with 42×8 bits. The data memory is divided into two functional groups: special function registers and general purpose data memory (32×8). Most of them are read/write, but some are read only.

The special function registers include the indirect addressing register (00H), the memory pointer register (MP;01H), the accumulator (ACC;05H) the program

Mada	Program Counter									
Mode	*9	*8	*7	*6	*5	*4	*3	*2	*1	*0
Initial reset	0	0	0	0	0	0	0	0	0	0
Skip		Program Counter + 2								
Loading PCL	*9	*8	@7	@6	@5	@4	@3	@2	@1	@0
Jump, call branch	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
Return from subroutine	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

Program Counter

Note: *9~*0: Program counter bits #9~#0: Instruction code bits S9~S0: Stack register bits @7~@0: PCL bits

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

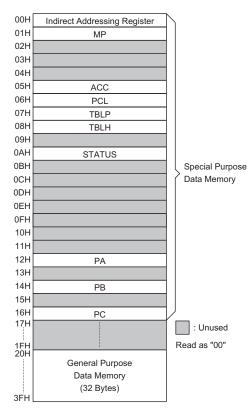
Table Location

Note: *9~*0: Table location bits

P9~P8: Current program counter bits

@7~@0: Table pointer bits





RAM Mapping

counter lower-order byte register (PCL;06H), the table pointer (TBLP;07H), the table higher-order byte register (TBLH;08H), the status register (STATUS;0AH) and the I/O registers (PA;12H, PB;14H, PC;16H). The remaining space before the 20H is reserved for future expanded usage and reading these locations will return the result 00H. The general purpose data memory, addressed from 20H to 3FH, is used for data and control information under instruction command.

All data memory areas can handle arithmetic, logic, increment, decrement and rotate operations directly. Except for some dedicated bits, each bit in the data memory can be set and reset by the SET [m].i and CLR [m].i instructions, respectively. They are also indirectly accessible through memory pointer register (MP;01H).

Indirect Addressing Register

Location 00H is an indirect addressing register that is not physically implemented. Any read/write operation of [00H] accesses data memory pointed to by MP (01H). Reading location 00H itself indirectly will return the result 00H. Writing indirectly results in no operation. The memory pointer register MP (01H) is a 7-bit register. The bit 7 of MP is undefined and reading will return the result "1". Any writing operation to MP will only transfer the lower 7-bit data to MP.

Accumulator

The accumulator closely relates to ALU operations. It is also mapped to location 05H of the data memory and is capable of carrying out immediate data operations. Data movement between two data memory locations has to pass through the accumulator.

Arithmetic and Logic Unit – ALU

This circuit performs 8-bit arithmetic and logic operation. The ALU 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)

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

Status Register – STATUS

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

With the exception of the TO and PDF flags, bits in the status register can be altered by instructions like most other register. Any data written into the status register will not change the TO or PDF flags. In addition it should be noted that operations related to the status register may give different results from those intended. The TO and PDF flags can only be changed by the Watchdog Timer overflow, chip power-up, clearing the Watchdog Timer and executing the HALT instruction.

The Z, OV, AC and C flags generally reflect the status of the latest operations.

In addition, on executing the subroutine call, the status register will not be automatically pushed onto the stack. If the contents of the status are important and if the sub-routine can corrupt the status register, precautions must be taken to save it properly.

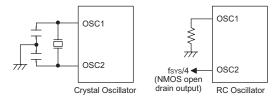


Bit No.	Label	Function
0	С	C is set if the operation results in a carry during an addition operation or if a borrow does not take place during a subtraction operation; otherwise C is cleared. C is also affected by a rotate through carry instruction.
1	AC	AC is set if the operation results in a carry 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 when either a system power-up or executing the CLR WDT instruction. PDF is set by executing the HALT instruction.
5	то	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

Oscillator Configuration

There are two oscillator circuits implemented in the microcontroller.



System Oscillator

Both are designed for system clocks; the RC oscillator and the Crystal oscillator, which are determined by code options. No matter what oscillator type is selected, the signal provides the system clock. The HALT mode stops the system oscillator and ignores the external signal to conserve power.

If an RC oscillator is used, an external resistor between OSC1 and VSS in needed and the resistance must range from 51k Ω to 1M Ω . The system clock, divided by 4, is available on OSC2, 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 V_{DD} , temperature and the chip itself due to process variations. It is, therefore, not suitable for timing sensitive operations where accurate oscillator frequency is desired.

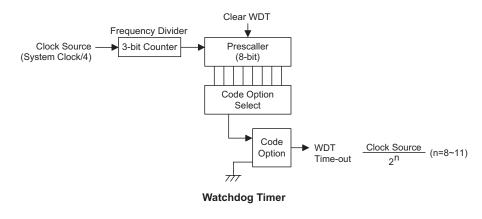
If the Crystal oscillator is used, a crystal across OSC1 and OSC2 is needed to provide the feedback and phase shift for the oscillator. No other external components are needed. Instead of a crystal, the resonator can also be connected between OSC1 and OSC2 to get a frequency reference, but two external capacitors in OSC1 and OSC2 are required.

Watchdog Timer – WDT

The clock source of the WDT is implemented by instruction clock (system clock divided by 4). The clock source is processed by a frequency divider and a prescaller to yield various time out periods.

WDT time out period =
$$\frac{\text{Clock Source}}{2^n}$$

Where n= 8~11 selected by code option.





This timer is designed to prevent a software malfunction or sequence jumping to an unknown location with unpredictable results. The Watchdog Timer can be disabled by code option. If the Watchdog Timer is disabled, all the executions related to the WDT result in no operation and the WDT will lose its protection purpose. In this situation the logic can only be restarted by an external logic.

A WDT overflow under normal operation will initialize "chip reset" and set the status bit "TO". To clear the contents of the WDT prescaler, three methods are adopted; external reset (a low level to RES), software instructions, or a HALT instruction. There are two types of software instructions. One type is the single instruction "CLR WDT", the other type comprises two instructions, "CLR WDT1" and "CLR WDT2". Of these two types of instructions, only one can be active depending on the code option - "CLR WDT times selection option". If the "CLR WDT" is selected (i.e.. CLR WDT times equal one), any execution of the CLR WDT instruction will clear the WDT. In case "CLR WDT1" and "CLR WDT2" are chosen (i.e., CLR WDT times equal two), these two instructions must be executed to clear the WDT; otherwise, the WDT may reset the chip due to a time-out.

Power Down Operation – HALT

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

- The system oscillator turns off and the WDT stops.
- The contents of the on-chip RAM and registers remain unchanged.
- WDT prescaler are cleared.
- · All I/O ports maintain their original status.
- The PDF flag is set and the TO flag is cleared.

The system can quit the HALT mode by means of an external reset or an external falling edge signal on port B. An external reset causes a device initialization. Examining the TO and PDF flags, the reason for chip reset can be determined. The PDF flag is cleared when the system powers up or execute the CLR WDT instruction and is set when the HALT instruction is executed. The TO flag is set if the WDT time-out occurs, and causes a wake-up that only resets the program counter and SP, the others keep their original status.

The port B wake-up can be considered as a continuation of normal execution. Each bit in port B can be independently selected to wake up the device by the code option. Awakening from an I/O port stimulus, the program will resume execution of the next instruction.

Once a wake-up event(s) occurs, it takes 1024 $t_{\mbox{SYS}}$ (system clock period) to resume normal operation. In other words, a dummy cycle period will be inserted after the wake-up.

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

Reset

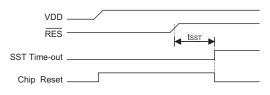
There are three ways in which a reset can occur:

- RES reset during normal operation
- RES reset during HALT
- WDT time-out reset during normal operation

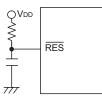
Some registers remain unchanged during reset conditions. Most registers are reset to the "initial condition" when the reset conditions are met. By examining the PDF and TO flags, the program can distinguish between different "chip resets".

то	PDF	RESET Conditions
0	0	RES reset during power-up
u	u	RES reset during normal operation
0	1	RES wake-up HALT
1	u	WDT time-out during normal operation

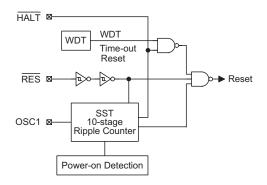
Note: "u" means unchanged.







Reset Circuit



Reset Configuration



Register	Reset (Power On)	WDT Time-out (Normal Operation)	RES Reset (Normal Operation)	RES Reset (HALT)	WDT Time-out (HALT)*
Program Counter	000H	000H	000H	000H	000H
MP	-xxx xxxx	-uuu uuuu	-uuu uuuu	-uuu uuuu	-uuu uuuu
ACC	XXXX XXXX	นนนน นนนน	นนนน นนนน	uuuu uuuu	սսսս սսսս
TBLP	XXXX XXXX	սսսս սսսս	นนนน นนนน	uuuu uuuu	սսսս սսսս
TBLH	xx xxxx	uu uuuu	นน นนนน	uu uuuu	uu uuuu
STATUS	00 xxxx	1u uuuu	uu uuuu	01 uuuu	11 uuuu
PA	1111 1111	1111 1111	1111 1111	1111 1111	սսսս սսսս
РВ	1111 1111	1111 1111	1111 1111	1111 1111	นนนน นนนน
PC	1	1	1	1	u

The chip reset status of the registers is summarized in the following table:

Note: "u" means unchanged

"x" means unknown

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

When a system power up occurs, an SST delay is added during the reset period. But when the reset comes from the $\overline{\text{RES}}$ pin, the SST delay is disabled. Any wake-up from HALT will enable the SST delay.

The functional unit chip reset status is shown below.

Program Counter	000H
WDT Prescaler	Clear
Input/Output ports	Input mode
Stack Pointer	Points to the top of the stack
Carrier output	Low level

Carrier

The HT48RA0-1/HT48CA0-1 provides a carrier output which shares the pin with PC0. It can be selected to be a carrier output (REM) or level output pin (PC0) by code option. If the carrier output option is selected, setting PC0="0" to enable carrier output and setting PC0="1" to disable it at low level output.

The clock source of the carrier is implemented by instruction clock (system clock divided by 4) and processed by a frequency divider to yield various carry frequency.

$$\label{eq:Carry Frequency} \mbox{Clock Source} \ \ \frac{\mbox{Clock Source}}{m \times 2^n}$$

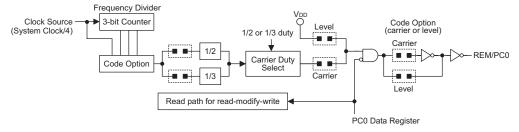
where m=2 or 3 and n=0~3, both are selected by code option. If m=2, the duty cycle of the carrier output is 1/2 duty. If m=3, the duty cycle of the carrier output can be 1/2 duty or 1/3 duty also determined by code option (with the exception of n=0).

Detailed selection of the carrier duty is shown below:

m×2 ⁿ	Duty Cycle
2, 4, 8, 16	1/2
3	1/3
6, 12, 24	1/2 or 1/3

The following table shows examples of carrier frequency selection.

f _{SYS}	f _{CARRIER}	Duty	m×2 ⁿ
	37.92kHz	1/3 only	3
455kHz	56.9kHz	1/2 only	2



Carrier/Level Output



Input/Output Ports

There are an 8-bit bidirectional input/output port, a 6-bit input with 2-bit I/O port and one-bit output port in the HT48RA0-1/HT48CA0-1, labeled PA, PB and PC which are mapped to [12H], [14H], [16H] of the RAM, respectively. Each bit of PA can be selected as NMOS output or Schmitt trigger with pull-high resistor by software instruction. PB0~PB1 have the same structure with PA, while PB2~PB7 can only be used for input operation (Schmitt trigger with pull-high resistors). PC is only one-bit output port shares the pin with carrier output. If the level option is selected, the PC is CMOS output.

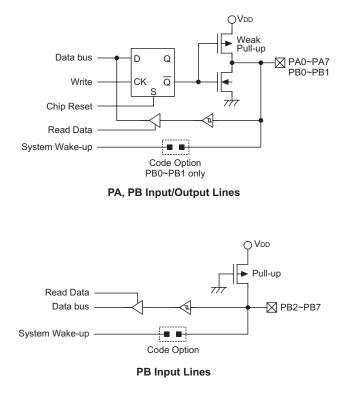
Both PA and PB for the input operation, these ports are non-latched, that is, the inputs should be ready at the T2 rising edge of the instruction "MOV A, [m]" (m=12H or 14H). For PA, PB0~PB1 and PC output operation, all data are latched and remain unchanged until the output latch is rewritten.

When the PA and PB0~PB1 is used for input operation, it should be noted that before reading data from pads, a "1" should be written to the related bits to disable the NMOS device. That is, the instruction "SET [m].i" (i=0~7 for PA, i=0~1 for PB) is executed first to disable related NMOS device, and then "MOV A, [m]" to get stable data. After chip reset, PA and PB remain at a high level input line while PC remain at high level output, if the level option is selected.

Each bit of PA, PB0~PB1 and PC output latches can be set or cleared by the "SET [m].i" and "CLR [m].i" (m=12H, 14H or 16H) instructions respectively.

Some instructions first input data and then follow the output operations. For example, "SET [m].i", "CLR [m]", "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.

Each line of PB has a wake-up capability to the device by code option. The highest seven bits of PC are not physically implemented, on reading them a "0" is returned and writing results in a no-operation.



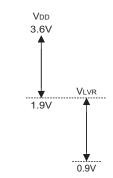


Low Voltage Reset – LVR

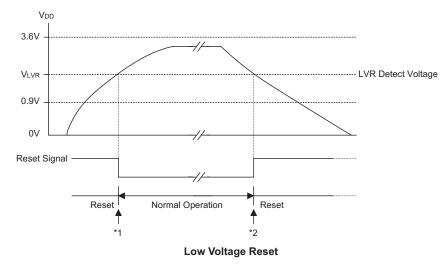
The microcontroller provides low voltage reset circuit in order to monitor the supply voltage of the device. If the supply voltage of the device is within the range $0.9V \sim V_{LVR}$, such as changing a battery, the LVR will automatically reset the device internally.

The LVR includes the following specifications:

- The low voltage (0.9V~V_{LVR}) has to remain in their original state to exceed 1ms. If the low voltage state does not exceed 1ms, the LVR will ignore it and do not perform a reset function.
- The LVR uses the "OR" function with the external $\overrightarrow{\text{RES}}$ signal to perform chip reset.



The relationship between V_{DD} and V_{LVR} is shown below.



- Note: "*1" To make sure that the system oscillator has stabilized, the SST provides an extra delay of 1024 system clock pulses before entering the normal operation.
 - "*2" Since low voltage has to be maintained in its original state and exceed 1ms, therefore 1ms delay enters the reset mode.



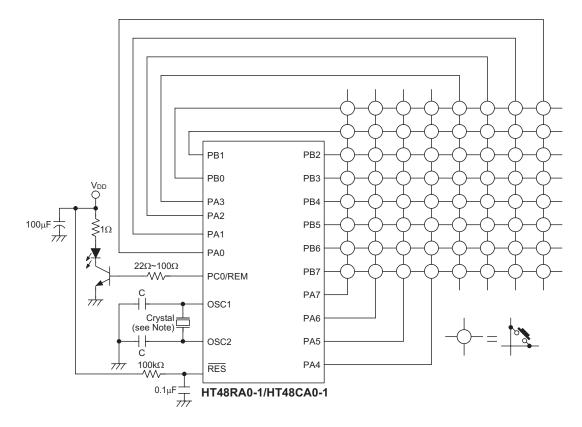
Code Option

The following table shows eight kinds of code option in the HT48RA0-1/HT48CA0-1. All the code options must be defined to ensure proper system functioning.

No.	Code Option
1	WDT time-out period selection Time-out period= $\frac{2^{n}}{\text{Clock Source}}$, where n=8~11.
2	WDT enable/disable selection. This option is to decide whether the WDT timer is enabled or disabled.
3	CLR WDT times selection. This option defines how to clear the WDT by instruction. "One time" means that the CLR WDT instruction can clear the WDT. "Two times" means only if both of the CLR WDT1 and CLR WDT2 instructions have been executed, the WDT can be cleared.
4	Wake-up selection. This option defines the wake-up activity function. External input pins (PB only) all have the capability to wake-up the chip from a HALT.
5	Carrier/level output selection. This option defines the activity of PC0 to be carrier output or level output.
6	Carry frequency selection. Carry frequency= $\frac{\text{Clock Source}}{(2 \text{ or } 3) \times 2^{n}}, \text{ where } n=0~3.$
7	Carrier duty selection. There are two types of selection: 1/2 duty or 1/3 duty. If carrier frequency=Clock Source/(2, 4, 8 or 16), the duty cycle will be 1/2 duty. If carrier frequency=Clock Source/3, the duty cycle will be 1/3 duty. If carrier frequency=Clock Source/(6, 12 or 24), the duty cycle can be 1/2 duty or 1/3 duty.
8	System oscillator selection. RC or crystal oscillator.
9	LVR function: enable or disable



Application Circuits



 Note:
 It is recommended that a 100μF decoupling capacitor is placed between VSS and VDD.

 The resistance and capacitance for reset circuit should be designed to ensure that the VDD is stable and remains in a valid range of the operating voltage before bringing RES to high.

Crystal or Resonator	С
4MHz Crystal	0pF
4MHz Resonator	10pF
3.58MHz Crystal	0pF
3.58MHz Resonator	25pF
2MHz Crystal & Resonator	25pF
1MHz Crystal	35pF
480kHz Resonator	300pF
455kHz Resonator	300pF
429kHz Resonator	300pF

The following table shows the C value according to different crystal values. (For reference only)



Instruction Set Summary

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

 ${\bf \forall}: {\sf Flag} \text{ is affected}$

-: 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

Description		ents of the usly, leavir	•		•		I the carry flag	are ac
Operation	$ACC \leftarrow A$.CC+[m]+C	;					
Affected flag(s)								
	то	PDF	OV	Z	AC	С		
			\checkmark		\checkmark	\checkmark		
ADCM A,[m]	Add the a	ccumulato	r and carr	y to data r	nemory			
Description			•			nulator and ta memor	l the carry flag /.	are a
Operation	[m] ← AC	C+[m]+C						
Affected flag(s)								
	то	PDF	OV	Z	AC	С		
			\checkmark		\checkmark	\checkmark		
ADD A,[m]	Add data	memory to	the accu	nulator				
Description		ents of the the accum		data mem	ory and the	e accumul	ator are added	. The
Operation	$ACC \leftarrow A$	CC+[m]						
Affected flag(s)								
	ТО	PDF	OV	Z	AC	С		
			\checkmark	\checkmark	\checkmark	\checkmark		
ADD A,x	Add imme	ediate data	to the acc	cumulator				
ADD A,x Description		ents of the a			specified	data are ac	lded, leaving th	ne resi
Description	The conte	ents of the a tor.			specified	data are ac	lded, leaving th	ne resi
Description	The conte accumula	ents of the a tor.			specified (data are ac	lded, leaving th	ne resu
Description	The conte accumula	ents of the a tor.			specified of	data are ac	lded, leaving th	ne resu
	The conte accumula ACC ← A	ents of the a tor. CC+x	accumulat	or and the			lded, leaving th	ne resu
Description Operation	The conte accumula ACC ← A TO 	ents of the a tor. CC+x	oV √	z	AC √	С	lded, leaving th	ne resi
Description Operation Affected flag(s)	The conte accumula ACC ← A TO Add the a The conte	ents of the a tor. CC+x PDF 	OV √ r to the da	z √ ta memor	AC √ y	C V	lded, leaving th	
Description Operation Affected flag(s) ADDM A,[m]	The conte accumula ACC ← A TO Add the a The conte	ents of the a tor. CC+x PDF 	OV √ r to the da	z √ ta memor	AC √ y	C V		
Description Operation Affected flag(s) ADDM A,[m] Description	The conte accumula ACC ← A TO Add the a The conte stored in t	ents of the a tor. CC+x PDF 	OV √ r to the da	z √ ta memor	AC √ y	C V		
Description Operation Affected flag(s) ADDM A,[m] Description Operation	The conte accumula ACC ← A TO Add the a The conte stored in t	ents of the a tor. CC+x PDF 	OV √ r to the da	z √ ta memor	AC √ y	C V		



Description Data in the accumulator and the specified data memory performeration. The result is stored in the accumulator. Operation ACC \leftarrow ACC "AND" [m] Affected flag(s) \overline{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 bite The result is stored in the accumulator. Operation ACC \leftarrow ACC "AND" x Affected flag(s) \overline{TO} PDF OV Z AC C $ -$ AND A,m Logical AND data memory with the accumulator. Operation ACC \leftarrow ACC "AND" x Affected flag(s) \overline{TO} PDF OV Z AC C $ -$ Operation Data in the specified data memory and the accumulator performeration. The result is stored in the data memory. Operation [m] \leftarrow ACC "AND" [m] Affected flag(s) \overline{TO} PDF OV Z AC C $ -$ Operation [m] \leftarrow ACC "AND" [m] Affected flag(s) \overline{TO} PDF OV Z AC C $ -$ Operation The instruction unconditionally calls a subroutine located at program counter increments once to obtain the address of the this onto the stack. The indicated address is t	AND A,[m]	Logical AND accumulator with data memory
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 bit The result is stored in the accumulator. Operation ACC \leftarrow ACC "AND" x Affected flag(s) TO PDF OV Z AC C AND A,[m] Logical AND data memory with the accumulator Description Data in the specified data memory and the accumulator performeration. The result is stored in the data memory. Operation [m] \leftarrow ACC "AND" [m] Affected flag(s) TO PDF OV Z AC C Operation [m] \leftarrow ACC "AND" [m] Image: ACC "AND" [m] Image: ACC "AND" [m] Image: ACC "AND" [m] Affected flag(s) TO PDF OV Z AC C Operation [m] \leftarrow ACC "AND" [m] Image: ACC "AND" [m] Image: ACC "AND" [m] Affected flag(s) TO PDF OV Z AC C Operation The instruction unconditionally calls a subroutine located at program counter increments once to obtain the address of the this address. Image: ACC </td <td>Description</td> <td></td>	Description	
TOPDFOVZACCImage: constraint of the section of the section of the section of the section of the sectified data perform a bit the result is stored in the accumulator.AND A,xLogical AND immediate data to the accumulatorDescriptionData in the accumulator and the specified data perform a bit the result is stored in the accumulator.OperationACC \leftarrow ACC "AND" xAffected flag(s)TOPDFOVZACCImage: constraint of the specified data memory with the accumulatorDescriptionData in the specified data memory and the accumulator performeration. The result is stored in the data memory.Operation[m] \leftarrow ACC "AND" [m]Affected flag(s)TOPDFOVZACCImage: constraint of the specified data memory and the accumulator performeration. The result is stored in the data memory.Operation[m] \leftarrow ACC "AND" [m]Affected flag(s)TOPDFOVZACCImage: constraint of the stack. The indicated address is then loaded. PPWith the instruction unconditionally calls a subroutine located at program counter (- addrPAffected flag(s)TOPDFOVZACCImage: constraint of the stack. The indicated address is then loaded. PPImage: constraint of the specified data memory are cleared to 0.OperationStack (- Program Counter+1Program Counter+1PImage: constraint of the specified data memory are cleared to 0.OperationThe contents of the specified data memory are cl	Operation	ACC ← ACC "AND" [m]
Image: Second secon	Affected flag(s)	
AND A,x Logical AND immediate data to the accumulator Description Data in the accumulator and the specified data perform a bit The result is stored in the accumulator. Operation ACC \leftarrow 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 performent of the action. The result is stored in the data memory. Operation [m] \leftarrow ACC "AND" [m] Affected flag(s) TO PDF OV Z AC C Operation [m] \leftarrow ACC "AND" [m] Affected flag(s) TO PDF OV Z AC C CALL addr Subroutine call TO PDF OV Z AC C Description The instruction unconditionally calls a subroutine located at program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. P with the instruction at this address. Operation Stack \leftarrow Program Counter+1 Program Counter+1 Program Counter \leftarrow addr Affected flag(s) TO PDF OV Z AC C CLR [m] <td></td> <td>TO PDF OV Z AC C</td>		TO PDF OV Z AC C
Description Data in the accumulator and the specified data perform a bit The result is stored in the accumulator. Operation ACC \leftarrow ACC "AND" x Affected flag(s) \overline{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 performant in the specified data memory. Operation [m] \leftarrow ACC "AND" [m] Affected flag(s) \overline{TO} PDF OV Z AC C Operation [m] \leftarrow ACC "AND" [m] Affected flag(s) \overline{TO} PDF OV Z AC C CALL addr Subroutine call \overline{TO} PDF OV Z AC C Description The instruction unconditionally calls a subroutine located at program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. P with the instruction at this address. $\overline{POProgram Counter+1}$ $\overline{Program Counter \leftarrow addr}$ Affected flag(s) \overline{TO} \overline{PDF} \overline{OV} \overline{AC} \overline{C} \overline{DP} \overline{OV} \overline{Z} \overline{AC} \overline{C} $ \overline{-$		
The result is stored in the accumulator. Operation ACC \leftarrow 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 performeration. The result is stored in the data memory. Operation [m] \leftarrow ACC "AND" [m] Affected flag(s) TO PDF OV Z AC C Operation [m] \leftarrow 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 program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. P with the instruction at this address. Operation Stack \leftarrow Program Counter+1 Program Counter+1 Program Counter \leftarrow addr Affected flag(s) TO PDF OV Z AC C CLR [m] Clear data memory Clear data memory Clear data memory are cleared to 0. Operation The contents of the specified data memory are cleared to 0. Operation [m] \leftarrow OOH AC C C	AND A,x	Logical AND immediate data to the accumulator
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 performeration. The result is stored in the data memory. Operation [m] \leftarrow ACC "AND" [m] Affected flag(s) TO PDF OV Z AC C CALL addr Subroutine call Operation The instruction unconditionally calls a subroutine located at program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. Find the instruction at this address. Operation Stack \leftarrow Program Counter+1 Program Counter \leftarrow addr Affected flag(s) TO PDF OV Z AC C CLR [m] Clear data memory OV Z AC C Description To entents of the specified data memory are cleared to 0. Operation The contents of the specified data memory are cleared to 0. Operation [m] \leftarrow 00H Affected flag(s) M AC C	Description	
TOPDFOVZACC $ -$ ANDM A,[m]Logical AND data memory with the accumulatorDescriptionData in the specified data memory and the accumulator perforeration. The result is stored in the data memory.Operation[m] \leftarrow ACC "AND" [m]Affected flag(s)TOPDFOVZACCCALL addrSubroutine callDescriptionThe instruction unconditionally calls a subroutine located a program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. FOperationStack \leftarrow Program Counter+1 Program Counter+1 Program Counter \leftarrow addrACCCAffected flag(s)TOPDFOVZACCCLR [m]Clear data memoryClear data memoryThe contents of the specified data memory are cleared to 0.OperationIntercontents of the specified data memory are cleared to 0.OperationIntercontents of the specified data memory are cleared to 0.	Operation	$ACC \gets ACC "AND" x$
ANDM A,[m] Logical AND data memory with the accumulator Description Data in the specified data memory and the accumulator performation. The result is stored in the data memory. Operation $[m] \leftarrow ACC$ "AND" [m] Affected flag(s) \overline{TO} PDF OV Z AC C CALL addr Subroutine call \overline{V} <td>Affected flag(s)</td> <td></td>	Affected flag(s)	
ANDM A,[m] Logical AND data memory with the accumulator Description Data in the specified data memory and the accumulator performeration. The result is stored in the data memory. Operation [m] \leftarrow ACC "AND" [m] Affected flag(s) \overline{TO} PDF OV Z AC C CALL addr Subroutine call \overline{V} <		
DescriptionData in the specified data memory and the accumulator performeration. The result is stored in the data memory.Operation $[m] \leftarrow ACC$ "AND" $[m]$ Affected flag(s) $\boxed{TO PDF OV Z AC C}{\ - }$ CALL addrSubroutine callDescriptionThe instruction unconditionally calls a subroutine located a program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. F with the instruction at this address.OperationStack \leftarrow Program Counter+1 Program Counter \leftarrow addrAffected flag(s) $\boxed{TO PDF OV Z AC C}{\ - - - - - - - - - $		
operation[m] \leftarrow ACC "AND" [m]Affected flag(s) \overline{TO} PDFOVZACC $ $ $ -$ CALL addrSubroutine callDescriptionThe instruction unconditionally calls a subroutine located at program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. F with the instruction at this address.OperationStack \leftarrow Program Counter+1 Program Counter \leftarrow addrAffected flag(s) \overline{TO} PDFOVZACC $ -$ CLR [m]Clear data memory The contents of the specified data memory are cleared to 0. Operation[m] \leftarrow 00HAffected flag(s)	ANDM A,[m]	Logical AND data memory with the accumulator
Affected flag(s) TO PDF OV Z AC C $ -$ CALL addr Subroutine call The instruction unconditionally calls a subroutine located at program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. P with the instruction at this address. Operation Stack \leftarrow Program Counter+1 Program Counter+1 Program Counter \leftarrow addr Affected flag(s) TO PDF OV Z AC C $ -$ CLR [m] Clear data memory The contents of the specified data memory are cleared to 0. Operation Operation [m] \leftarrow 00H Affected flag(s) $ -$	Description	
TOPDFOVZACC $ -$ CALL addrSubroutine callDescriptionThe instruction unconditionally calls a subroutine located a program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. F with the instruction at this address.OperationStack \leftarrow Program Counter+1 Program Counter \leftarrow addrAffected flag(s)TOPDFOVZACC $ -$ CLR [m]Clear data memory The contents of the specified data memory are cleared to 0. OperationClear data memory Image: OperationClear data memory Image: OperationAffected flag(s)Image: OperationImage: OperationImage: OperationAffected flag(s)Image: OperationImage: OperationImage: OperationAffected flag(s)Image: OperationImage: OperationImage: OperationAffected flag(s)Image: OperationImage: OperationImage: Operation	Operation	[m] ← ACC ″AND″ [m]
CALL addr Subroutine call Description The instruction unconditionally calls a subroutine located at program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. F with the instruction at this address. Operation Stack \leftarrow Program Counter+1 Program Counter \leftarrow addr Affected flag(s) TO PDF OV Z AC C $ -$ CLR [m] Clear data memory Clear data memory are cleared to 0. Operation [m] \leftarrow 00H Affected flag(s) $ -$	Affected flag(s)	
CALL addr Subroutine call Description The instruction unconditionally calls a subroutine located at program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. F with the instruction at this address. Operation Stack \leftarrow Program Counter+1 Program Counter \leftarrow addr Affected flag(s) TO PDF OV Z AC C $ -$ CLR [m] Clear data memory Clear data memory The contents of the specified data memory are cleared to 0. Operation Affected flag(s) $(m) \leftarrow 00H$ Affected flag(s) $(m) \leftarrow 00H$		TO PDF OV Z AC C
Description The instruction unconditionally calls a subroutine located at program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. P with the instruction at this address. Operation Stack \leftarrow Program Counter+1 Program Counter \leftarrow 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] \leftarrow 00H Affected flag(s) $ -$		
program counter increments once to obtain the address of the this onto the stack. The indicated address is then loaded. P with the instruction at this address. Operation Stack \leftarrow Program Counter+1 Program Counter \leftarrow addr Affected flag(s) TO PDF OV Z AC C $ -$ CLR [m] Clear data memory Clear data memory The contents of the specified data memory are cleared to 0. Operation [m] \leftarrow 00H Affected flag(s) $ -$	CALL addr	Subroutine call
$\begin{tabular}{lllllllllllllllllllllllllllllllllll$	Description	program counter increments once to obtain the address of the
Affected flag(s) TO PDF OV Z AC C $ -$ CLR [m] Clear data memory Clear data memory The contents of the specified data memory are cleared to 0. Operation [m] \leftarrow 00H Affected flag(s)		
TOPDFOVZACCCLR [m]Clear data memoryClear data memoryDescriptionThe contents of the specified data memory are cleared to 0.Operation[m] \leftarrow 00HAffected flag(s)	Operation	0
	Affected flag(s)	
DescriptionThe contents of the specified data memory are cleared to 0.Operation $[m] \leftarrow 00H$ Affected flag(s) \square		TO PDF OV Z AC C
DescriptionThe contents of the specified data memory are cleared to 0.Operation $[m] \leftarrow 00H$ Affected flag(s) \square		
DescriptionThe contents of the specified data memory are cleared to 0.Operation $[m] \leftarrow 00H$ Affected flag(s) $[m] \leftarrow 00H$	CLR [m]	Clear data memory
Affected flag(s)		The contents of the specified data memory are cleared to 0.
	Operation	[m] ← 00H
TO PDF OV Z AC C	Affected flag(s)	
		TO PDF OV Z AC C



CLR [m].i	Clear bit o	of data me	mory			
Description	The bit i c	f the spec	ified data r	nemory is	cleared to	0.
Operation	[m].i ← 0					
Affected flag(s)	[
	ТО	PDF	OV	Z	AC	С
				_	—	
CLR WDT	Clear Wat	chdog Tin	ner			
Description	The WDT cleared.	is cleared	(clears the	WDT). Tł	ne power d	lown bit (F
Operation	WDT \leftarrow 0 PDF and					
Affected flag(s)						
	ТО	PDF	OV	Z	AC	С
	0	0	—		—	_
CLR WDT1	Preclear \	Vatchdog	Timer			
Description	of this inst	ruction wit	NDT2, clea hout the ot has been	her precle	ar instruct	ion just se
Operation	WDT $\leftarrow 0$ PDF and					
Affected flag(s)						
Affected flag(s)	ТО	PDF	OV	Z	AC	С
Affected flag(s)	TO 0*	PDF 0*	OV	Z 	AC	C
Affected flag(s)		0*		Z 	AC	C
	0* Preclear V Together of this ins	0* Watchdog with CLR V truction w		ars the WI	 DT. PDF ar	
CLR WDT2	0* Preclear V Together of this ins	0* Watchdog with CLR V truction w nstruction 0H*	Timer NDT1, clea	ars the WI	 DT. PDF ar	
CLR WDT2 Description	0* Preclear \ Together v of this ins plies this i WDT ← 0	0* Watchdog with CLR V truction w nstruction 0H*	Timer NDT1, clea	ars the WI	 DT. PDF ar	
CLR WDT2 Description Operation	0* Preclear \ Together v of this ins plies this i WDT ← 0	0* Watchdog with CLR V truction w nstruction 0H*	Timer NDT1, clea	ars the WI	 DT. PDF ar	
CLR WDT2 Description Operation	0* Preclear \ Together v of this ins plies this i WDT ← 0 PDF and	0^* Watchdog with CLR V truction w nstruction 0H [*] TO $\leftarrow 0^*$	Timer WDT1, clea ithout the o has been	ars the WE other prec executed	DT. PDF ar lear instru and the To	nd TO are ction, set O and PD
CLR WDT2 Description Operation	0^* Preclear V of this ins plies this is WDT $\leftarrow 0$ PDF and TO	0^* Watchdog with CLR V truction w nstruction 0H* TO $\leftarrow 0^*$ PDF 0^*	Timer NDT1, clea ithout the o has been OV	ars the WE other prec executed	DT. PDF ar lear instru and the To	nd TO are ction, set O and PD
CLR WDT2 Description Operation Affected flag(s)	0^* Preclear V of this ins plies this is WDT $\leftarrow 0$ PDF and TO 0^* Complem Each bit of	0^* Watchdog with CLR \ truction w nstruction 0H* TO ← 0* PDF 0* ent data n of the spece	Timer NDT1, clea ithout the o has been OV	ars the WI other prec executed Z memory is	DT. PDF ar lear instru- and the To AC 	nd TO are ction, set O and PD C C complem
CLR WDT2 Description Operation Affected flag(s)	0^* Preclear V of this ins plies this is WDT $\leftarrow 0$ PDF and TO 0^* Complem Each bit of	0^* Watchdog with CLR \ truction w nstruction 0H* TO ← 0* PDF 0* ent data n of the spection viously co	Timer WDT1, clea ithout the o has been OV OV	ars the WI other prec executed Z memory is	DT. PDF ar lear instru- and the To AC 	nd TO are ction, set O and PD C C complem
CLR WDT2 Description Operation Affected flag(s) CPL [m] Description	0^* Preclear N of this ins plies this is WDT ← 0 PDF and TO 0^* Complem Each bit of which pre	0^* Watchdog with CLR \ truction w nstruction 0H* TO ← 0* PDF 0* ent data n of the spection viously co	Timer WDT1, clea ithout the o has been OV OV	ars the WI other prec executed Z memory is	DT. PDF ar lear instru- and the To AC 	nd TO are ction, set O and PD C C complem
CLR WDT2 Description Operation Affected flag(s) CPL [m] Description Operation	0^* Preclear N of this ins plies this is WDT ← 0 PDF and TO 0^* Complem Each bit of which pre	0^* Watchdog with CLR \ truction w nstruction 0H* TO ← 0* PDF 0* ent data n of the spection viously co	Timer WDT1, clea ithout the o has been OV OV	ars the WI other prec executed Z memory is	DT. PDF ar lear instru- and the To AC 	nd TO are ction, set O and PD C C complem



CPLA [m]	Complemer	nt data m	emory and	1 place res	sult in the	accumulat	tor
Description	Each bit of which previo	the spec ously con	ified data tained a 1	memory is are chang	s logically jed to 0 and	compleme d vice-vers	ented (1's complement) sa. The complemented mory remain unchange
Operation	$ACC \leftarrow [\overline{m}]$						
Affected flag(s)							1
	то	PDF	OV	Z	AC	С	
		_		\checkmark			
DAA [m]	Decimal-Ad	just accu	imulator fo	or addition			
Description	lator is divid carry (AC1) justment is d	led into t will be do done by a C) is set	wo nibbles one if the lo adding 6 to ; otherwise	s. Each nib ow nibble o o the origin e the origin	oble is adju of the accu nal value if nal value re	usted to th mulator is the origina emains un	Decimal) code. The acc ne BCD code and an inf greater than 9. The BC al value is greater than changed. The result is s red.
Operation	If ACC.3~A0 then [m].3~[else [m].3~[and	[m].0 ← ([m].0 ← ((ACC.3~A) ACC.3~A(CC.0), AC			
	If ACC.7~A0 then [m].7~[else [m].7~[[m].4 ← /	ACC.7~AC	C.4+6+A			
Affected flag(s)	then [m].7~[else [m].7~[[m].4 ← / [m].4 ← /	ACC.7~AC	C.4+6+A C.4+AC1	,C=C		1
Affected flag(s)	then [m].7~[[m].4 ← /	ACC.7~AC	C.4+6+A		С	
Affected flag(s)	then [m].7~[else [m].7~[[m].4 ← / [m].4 ← /	ACC.7~AC	C.4+6+A C.4+AC1	,C=C	С √	
Affected flag(s) DEC [m]	then [m].7~[else [m].7~[[m].4 ← / [m].4 ← / PDF 	ACC.7~AC ACC.7~AC	C.4+6+A C.4+AC1	,C=C		
	then [m].7~[else [m].7~[TO	[m].4 ← / m].4 ← / PDF 	ACC.7~AC	2C.4+6+A 2C.4+AC1 Z	,C=C AC —	\checkmark	
DEC [m]	then [m].7~[else [m].7~[TO Decrement	[m].4 ← / m].4 ← / PDF 	ACC.7~AC	2C.4+6+A 2C.4+AC1 Z	,C=C AC —	\checkmark	
DEC [m] Description	then [m].7~[else [m].7~[TO Decrement Data in the	[m].4 ← / m].4 ← / PDF 	ACC.7~AC	2C.4+6+A 2C.4+AC1 Z	,C=C AC —	\checkmark	
DEC [m] Description Operation	then [m].7~[else [m].7~[TO Decrement Data in the	[m].4 ← / m].4 ← / PDF 	ACC.7~AC	2C.4+6+A 2C.4+AC1 Z	,C=C AC —	\checkmark	
DEC [m] Description Operation	then [m].7~[else [m].7~[TO Decrement Data in the [m] \leftarrow [m]-1	[m].4 ← / m].4 ← / PDF data men specified	ACC.7~AC ACC.7~AC OV — mory data men	C.4+6+Ad C.4+AC1 Z 	AC AC —	√ by 1.	
DEC [m] Description Operation	then [m].7~[else [m].7~[TO Decrement Data in the [m] \leftarrow [m]-1	[m].4 ← / m].4 ← / PDF 	ACC.7~AC ACC.7~AC OV — mory data men OV —	$\frac{2}{z}$	AC AC cremented AC AC	√ by 1.]]
DEC [m] Description Operation Affected flag(s)	then [m].7~[else [m].7~[TO Decrement Data in the $[m] \leftarrow [m]-1$ TO Decrement	[m].4 ← / m].4 ← / PDF 	ACC.7~AC ACC.7~AC OV — mory data men OV — mory and p data mem	C.4+6+Ar C.4+AC1 Z 	AC A	√ by 1. C — ccumulator by 1, leavir	r ng the result in the accur
DEC [m] Description Operation Affected flag(s)	then [m].7~[else [m].7~[TO Decrement Data in the s $[m] \leftarrow [m]-1$ TO Decrement Data in the s	[m].4 ← / m].4 ← / PDF 	ACC.7~AC ACC.7~AC OV — mory data men OV — mory and p data mem	C.4+6+Ar C.4+AC1 Z 	AC A	√ by 1. C — ccumulator by 1, leavir	
DEC [m] Description Operation Affected flag(s) DECA [m] Description	then [m].7~[else [m].7~[TO Decrement Data in the s [m] \leftarrow [m]-1 TO Decrement Data in the s tor. The con	[m].4 ← / m].4 ← / PDF 	ACC.7~AC ACC.7~AC OV — mory data men OV — mory and p data mem	C.4+6+Ar C.4+AC1 Z 	AC A	√ by 1. C — ccumulator by 1, leavir	
DEC [m] Description Operation Affected flag(s) DECA [m] Description Operation	then [m].7~[else [m].7~[TO Decrement Data in the s [m] \leftarrow [m]-1 TO Decrement Data in the s tor. The con	[m].4 ← / m].4 ← / PDF 	ACC.7~AC ACC.7~AC OV — mory data men OV — mory and p data mem	C.4+6+Ar C.4+AC1 Z 	AC A	√ by 1. C — ccumulator by 1, leavir	



HALT	Enter pov	ver down r	mode					
Description	This instruction stops program execution and turns off the system clock. The conte the RAM and registers are retained. The WDT and prescaler are cleared. The power bit (PDF) is set and the WDT time-out bit (TO) is cleared.							
Operation	Program Counter \leftarrow Program Counter+1 PDF \leftarrow 1 TO \leftarrow 0							
Affected flag(s)								
	ТО	PDF	OV	Z	AC	С		
	0	1	_					
INC [m]	Incremen	t data mer	mory					
Description	Data in th	ne specifie	d data mer	mory is inc	remented	by 1		
Operation	[m] ← [m]+1						
Affected flag(s)	[
	ТО	PDF	OV	Z	AC	С		
		_	_	V		_		
INCA [m]	Incremen	t data mer	mory and p	lace resul	t in the ac	cumulator		
Description			d data men					
			the data n	nemory rei	main unch	anged.		
Operation	ACC ← [m]+1						
Affected flag(s)	то	PDF	OV	Z	AC	С		
	10			∠ √	AC			
				N				
JMP addr	Directly ju	ump						
Description			er are repla this destir		he directly	-specified		
Operation	Program	Counter ←	-addr					
Affected flag(s)	[
	ТО	PDF	OV	Z	AC	С		
		_	_	_		—		
MOV A,[m]	Move dat	a memory	to the acc	umulator				
Description	The conte	ents of the	specified	data mem	ory are co	pied to the		
Operation	ACC ← [m]						
Affected flag(s)								
	то	PDF	OV	Z	AC	С		
			_	_		—		



Description	The 8-bit of	data sneci					nulator.
Operation	ACC \leftarrow x						
Affected flag(s)							
	ТО	PDF	OV	Z	AC	С]
	_	_	_				-
							L
MOV [m],A			tor to data	-	ind to the	oposified	data mamany (an
Description	memories		accumulat	or are cop	led to the	specified	data memory (one
Operation	[m] ←ACC	2					
Affected flag(s)							
	то	PDF	OV	Z	AC	С	
		—	_	—	_	—	
NOP	No operat	ion					
Description	•		ormed Fx	ecution co	ontinues w	ith the ne	xt instruction.
Operation			- Program				
Affected flag(s)	riografii e		rogram	Countor '			
(neoted hag(s)	ТО	PDF	OV	Z	AC	С]
							-
					1		
OR A,[m]	Logical Of	R accumu	lator with o	data mem	ory		
DR A,[m] Description	Data in the	e accumu	lator and t	he specifie	ed data me		e of the data mer
Description	Data in the form a bit	e accumu vise logica	lator and tl al_OR ope	he specifie	ed data me		e of the data mer
Description Operation	Data in the	e accumu vise logica	lator and tl al_OR ope	he specifie	ed data me		
Description	Data in the form a bits $ACC \leftarrow AC$	e accumu wise logica CC "OR"	lator and tl al_OR ope [m]	he specifie ration. Th	ed data mo e result is	stored in	
Description Operation	Data in the form a bit	e accumu vise logica	lator and tl al_OR ope	he specifie ration. Th Z	ed data me		
Description Operation	Data in the form a bits $ACC \leftarrow AC$	e accumu wise logica CC "OR"	lator and tl al_OR ope [m]	he specifie ration. Th	ed data mo e result is	stored in	
Description Operation	Data in the form a bitw ACC ← A TO 	e accumu vise logica CC "OR" PDF 	lator and tl al_OR ope [m]	he specifi∉ ration. Th Z √	AC	stored in	
Description Operation Affected flag(s)	Data in the form a bitw ACC ← At TO Logical Of Data in the	e accumu wise logica CC "OR" PDF — R immedia e accumu	lator and ti al_OR ope [m] OV 	he specifie ration. Th Z √ the accur he specifi	AC	C	
Description Operation Affected flag(s) DR A,x Description	Data in the form a bitw ACC ← A TO Logical Of Data in the The result	e accumu wise logica CC "OR" PDF — R immedia e accumu is stored	lator and ti al_OR ope [m] OV ate data to ate data to alator and t in the accu	he specifie ration. Th Z √ the accur he specifi	AC	C	the accumulator.
Description Operation Affected flag(s) DR A,x Description Operation	Data in the form a bitw ACC ← At TO Logical Of Data in the	e accumu wise logica CC "OR" PDF — R immedia e accumu is stored	lator and ti al_OR ope [m] OV ate data to ate data to alator and t in the accu	he specifie ration. Th Z √ the accur he specifi	AC	C	the accumulator.
Description Operation Affected flag(s) DR A,x Description	Data in the form a bitw $ACC \leftarrow At$ TO Logical Of Data in the The result $ACC \leftarrow At$	e accumu wise logica CC "OR" PDF — R immedia e accumu is stored CC "OR"	lator and ti al_OR ope [m] OV ate data to ate data to ate data to in the accu x	he specifie ration. Th Z √ the accur he specifi umulator.	AC A	C C erform a b	the accumulator.
Description Operation Affected flag(s) DR A,x Description Operation	Data in the form a bitw ACC ← A TO Logical Of Data in the The result	e accumu wise logica CC "OR" PDF — R immedia e accumu is stored	lator and ti al_OR ope [m] OV ate data to ate data to alator and t in the accu	he specific ration. Th Z √ the accur he specifi umulator. Z	AC	C	the accumulator.
Description Operation Affected flag(s) DR A,x Description Operation	Data in the form a bitw $ACC \leftarrow At$ TO Logical Of Data in the The result $ACC \leftarrow At$	e accumu wise logica CC "OR" PDF — R immedia e accumu is stored CC "OR"	lator and ti al_OR ope [m] OV ate data to ate data to ate data to in the accu x	he specifie ration. Th Z √ the accur he specifi umulator.	AC A	C C erform a b	the accumulator.
Description Operation Affected flag(s) DR A,x Description Operation	Data in the form a bitw $ACC \leftarrow At$ TO Logical Of Data in the The result $ACC \leftarrow At$ TO TO TO	e accumu wise logica CC "OR" PDF — R immedia e accumu is stored CC "OR" PDF —	lator and ti al_OR ope [m] OV ate data to ate data to ate data to in the accu x	he specific ration. Th Z the accur he specific unulator. Z 	AC AC AC AC AC AC	C C erform a b	the accumulator.
Description Operation Affected flag(s) DR A,x Description Operation Affected flag(s)	Data in the form a bitw $ACC \leftarrow Ad$ TO Logical OF Data in the The result $ACC \leftarrow Ad$ TO Logical OF Logical OF	e accumu wise logica CC "OR" PDF 	lator and the al_OR operation of the algorithm of the accurate data to a second the accurate dat	the accur z the accur he specific unulator. z the accur	AC A	C C erform a b C	the accumulator.
Description Operation Affected flag(s) DR A,x Description Operation Affected flag(s)	Data in the form a bitw $ACC \leftarrow AC$ TO Logical OF Data in the The result $ACC \leftarrow AC$ TO Logical OF Data in the	e accumu wise logica CC "OR" PDF 	lator and the al_OR operation of the algorithm of the accurate data to a second the accurate dat	the accur z the accur he specific unulator. z the accur e of the of	AC A	C C erform a b C C ories) and	the accumulator.
Description Operation Affected flag(s) DR A,x Description Operation Affected flag(s)	Data in the form a bitw $ACC \leftarrow AC$ TO Logical OF Data in the The result $ACC \leftarrow AC$ TO Logical OF Data in the	e accumu wise logica CC "OR" PDF — R immedia e accumu is stored CC "OR" PDF — R data me ie data me jical_OR o	lator and ti al_OR ope [m] OV ate data to ate data to alator and t in the accu x OV OV emory with emory (on operation.	the accur z the accur he specific unulator. z the accur e of the of	AC A	C C erform a b C C ories) and	the accumulator.
Description Operation Affected flag(s) OR A,x Description Operation Affected flag(s) ORM A,[m] Description	Data in the form a bitw $ACC \leftarrow Ad$ TO Logical OF Data in the The result $ACC \leftarrow Ad$ TO Logical OF Data in the bitwise log [m] $\leftarrow ACC$	e accumu vise logica CC "OR" PDF —— R immedia e accumu is stored CC "OR" PDF —— R data me jical_OR o C "OR" [m	lator and ti al_OR ope [m] OV ate data to lator and t in the accu x OV OV emory with emory (on operation.]	the accur the accur he specific unulator. Z the accur e of the of The result	AC A	C C erform a b C C ories) and in the data	the accumulator.
Description Deration Affected flag(s) DR A,x Description Description Affected flag(s) DRM A,[m] Description Dperation	Data in the form a bitw ACC \leftarrow Ad TO Data in the Data in the The result ACC \leftarrow Ad TO Data in the Logical Of Data in the bitwise log	e accumu wise logica CC "OR" PDF — R immedia e accumu is stored CC "OR" PDF — R data me ie data me jical_OR o	lator and ti al_OR ope [m] OV ate data to ate data to alator and t in the accu x OV OV emory with emory (on operation.	the accur z the accur he specific unulator. z the accur e of the of	AC A	C C erform a b C C ories) and	the accumulator.



HT48RA0-1/HT48CA0-1

RET	Return fro	om subrou	tine						
Description	The program counter is restored from the stack. This is a 2-cycle instruction.								
Operation	Program Counter ← Stack								
Affected flag(s)									
	то	PDF	OV	Z	AC	С			
RET A,x	Return ar	id place in	nmediate d	lata in the	accumula	tor			
Description	The program counter is restored from the stack and the accumulator loaded with the fied 8-bit immediate data.								
Operation	Program ϕ ACC \leftarrow x	Counter ←	- Stack						
Affected flag(s)									
	ТО	PDF	OV	Z	AC	С			
RL [m]	Rotate da	ta memor	y left						
Description	The conte	ents of the s	specified d	ata memo	ry are rota	ted 1 bit le			
Operation	[m].(i+1) ∢ [m].0 ← [ɪ].i:bit i of t	he data m	emory (i=0)~6)			
Affected flag(s)									
	ТО	PDF	OV	Z	AC	С			
		_							
RLA [m]	Rotate da	ta memor	y left and p	blace resul	It in the ac	cumulator			
Description			l data merr accumula						
Operation	ACC.(i+1) ACC.0 ←		m].i:bit i of	the data r	memory (i	=0~6)			
Affected flag(s)									
	ТО	PDF	OV	Z	AC	С			
		_	_	—					
RLC [m]	Rotate da	ta memor	y left throu	gh carry					
Description			specified of the origination of						
Operation	[m].(i+1) ↔ [m].0 ← C C ← [m].	;].i:bit i of t	he data m	emory (i=0)~6)			
Affected flag(s)	[
	ТО	PDF	OV	Z	AC	С			



RLCA [m]	Rotate lef	t through	carry and p	place resu	It in the ac	cumulator
Description	Data in the specified data memory and the carry flag are rotated 1 bit left. Bit 7 replaces carry bit and the original carry flag is rotated into bit 0 position. The rotated result is sto in the accumulator but the contents of the data memory remain unchanged.					
Operation	ACC.(i+1) ACC.0 ← C ← [m].	С	[m].i:bit i of	f the data r	memory (i	=0~6)
Affected flag(s)						
	ТО	PDF	OV	Z	AC	С
						\checkmark
RR [m]	Rotate da	ta memor	y right			
Description	The conte	nts of the	specified d	ata memo	ry are rota	ted 1 bit rig
Operation	[m].i ← [n [m].7 ← [ı		n].i:bit i of t	he data m	emory (i=(0~6)
Affected flag(s)						
	ТО	PDF	OV	Z	AC	С
			_			
RRA [m]	Rotate rig	ht and pla	ice result i	n the accu	mulator	
RRA [m] Description	Data in th	e specifie	d data mer	n the accu nory is rota ulator. The	ated 1 bit i	0
	Data in th the rotate	e specified d result in t - [m].(i+1)	d data mer the accum	nory is rota	ated 1 bit i contents o	of the data
Description	Data in th the rotate ACC.(i) ←	e specified d result in t - [m].(i+1)	d data mer the accum	mory is rota ulator. The	ated 1 bit i contents o	of the data
Description Operation	Data in th the rotate ACC.(i) ←	e specified d result in t - [m].(i+1)	d data mer the accum	mory is rota ulator. The	ated 1 bit i contents o	of the data
Description Operation	Data in th the rotate ACC.(i) ← ACC.7 ←	e specified d result in t - [m].(i+1); [m].0	d data mer the accum ; [m].i:bit i	nory is rota ulator. The of the data	ated 1 bit i contents o memory	of the data (i=0~6)
Description Operation	Data in th the rotate ACC.(i) ← ACC.7 ←	e specified d result in t - [m].(i+1); [m].0 PDF	d data mer the accum ; [m].i:bit i	nory is rota ulator. The of the data Z	Ac	of the data (i=0~6)
Description Operation Affected flag(s)	Data in th the rotate ACC.(i) ← ACC.7 ← TO 	e specified d result in f - [m].(i+1); [m].0 PDF 	d data mer the accum ; [m].i:bit i OV y right thro specified	nory is rota ulator. The of the data Z	AC	(i=0~6) C
Description Operation Affected flag(s)	Data in th the rotate ACC.(i) ← ACC.7 ← TO 	e specified d result in f - [m].(i+1); [m].0 PDF ta memor ents of the preplaces n].(i+1); [m	d data mer the accum ; [m].i:bit i OV y right thro specified the carry h	nory is rota ulator. The of the data Z ugh carry data mem	Ac	(i=0~6) C
Description Operation Affected flag(s) RRC [m] Description	Data in the the rotated ACC.(i) \leftarrow ACC.7 \leftarrow TO TO Rotate da The conte right. Bit ([m].i \leftarrow [m [m].7 \leftarrow (n	e specified d result in f - [m].(i+1); [m].0 PDF ta memor ents of the preplaces n].(i+1); [m	d data mer the accum ; [m].i:bit i OV y right thro specified the carry h	nory is rota ulator. The of the data Z ugh carry data mem bit; the orig	Ac	(i=0~6) C
Description Operation Affected flag(s) RRC [m] Description Operation	Data in the the rotated ACC.(i) \leftarrow ACC.7 \leftarrow TO TO Rotate da The conte right. Bit ([m].i \leftarrow [m [m].7 \leftarrow (n	e specified d result in f - [m].(i+1); [m].0 PDF ta memor ents of the preplaces n].(i+1); [m	d data mer the accum ; [m].i:bit i OV y right thro specified the carry h	nory is rota ulator. The of the data Z ugh carry data mem bit; the orig	Ac	(i=0~6) 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 replace the carry bit and the original carry flag is rotated into the bit 7 position. The rotated result i stored in the accumulator. The contents of the data memory remain unchanged.						
Operation	ACC.i \leftarrow [m].(i+1); [m].i:bit i of the data memory (i=0~6) ACC.7 \leftarrow C C \leftarrow [m].0						
Affected flag(s)	[
	ТО	PDF	OV	Z	AC	С	
			—			\checkmark	
SBC A,[m]	Subtract of	lata memo	ory and car	ry from th	e accumul	ator	
Description			•		ory and the e result in t		ent of the carry flag are su ulator.
Operation	$ACC \leftarrow A$	CC+[m]+C	;				
Affected flag(s)	[
	ТО	PDF	OV	Z	AC	С	
		_		\checkmark	\checkmark	\checkmark	
SBCM A,[m]	Subtract of	lata memo	ory and car	ry from th	e accumul	ator	
Description	The conte	Subtract data memory and carry from the accumulator The contents of the specified data memory and the complement of the carry flag are sub- tracted from the accumulator, leaving the result in the data memory.					
Operation	[m] ← AC	C+[m]+C		-			
Affected flag(s)							
0()	ТО	PDF	OV	Z	AC	С	
	_	_		\checkmark	\checkmark		
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 nex instruction is skipped. If the result is 0, the following instruction, fetched during the curren 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]–1)=0, [m	ı] ← ([m]–1				
]–1)=0, [m	ı] ← ([m]–1				
]–1)=0, [m PDF	l] ← ([m]−1		AC	C	
	Skip if ([m)	AC		
Affected flag(s)	Skip if ([m	PDF	OV —	l) Z —		C	
Affected flag(s) SDZA [m]	Skip if ([m TO Decremen	PDF —	OV — mory and	Z	Ilt in ACC,	C — skip if 0	by 1. If the result is 0, the ne
Affected flag(s) SDZA [m]	Skip if ([m TO — Decremen The conte instruction unchange execution	PDF — at data me nts of the s is skipped d. If the real is discard	OV — mory and p specified da d. The resu sult is 0, the led and a c	Z place resu ata memo ilt is stored e following dummy cy	Ilt in ACC, ry are decr d in the acc g instruction	C 	by 1. If the result is 0, the ne but the data memory remair during the current instructio t the proper instruction (2 c
Affected flag(s) SDZA [m] Description	Skip if ([m TO Decremer The conte instructior unchange execution cles). Oth	PDF 	OV — mory and p specified da d. The resu sult is 0, the led and a c	Z Dace resu ata memo It is stored e following dummy cy the next in	It in ACC, ry are decr in the acc g instruction cle is repla	C 	but the data memory remain during the current instruction
Affected flag(s) SDZA [m] Description Operation	Skip if ([m TO Decremer The conte instructior unchange execution cles). Oth	PDF 	OV — specified da d. The resu sult is 0, the led and a c ceed with	Z Dace resu ata memo It is stored e following dummy cy the next in	It in ACC, ry are decr in the acc g instruction cle is repla	C 	but the data memory remain during the current instruction
Operation Affected flag(s) SDZA [m] Description Operation Affected flag(s)	Skip if ([m TO Decremer The conte instructior unchange execution cles). Oth	PDF 	OV — specified da d. The resu sult is 0, the led and a c ceed with	Z Dace resu ata memo It is stored e following dummy cy the next in	It in ACC, ry are decr in the acc g instruction cle is repla	C 	but the data memory remain during the current instruction



SET [m]	Set data memo	ry				
Description	Each bit of the	specified data	memory is	set to 1.		
Operation	[m] ← FFH					
Affected flag(s)						
	TO PE	F OV	Z	AC	С	
			_			
SET [m]. i	Set bit of data i	nemorv				
Description	Bit i of the spec		nory is set	to 1.		
Operation	[m].i ← 1					
Affected flag(s)						
3(1)	TO PE	F OV	Z	AC	С	
				1		I
SIZ [m]	Skip if increme	nt data memor	y is 0			
Description		•		•		by 1. If the result is 0, the fol-
	-		-			ecution, is discarded and a les). Otherwise proceed with
	the next instruc					.,
Operation	Skip if ([m]+1)=	0, [m] ← ([m]+	-1)			
Affected flag(s)						1
	TO PE	F OV	Z	AC	С	
SIZA [m]	Increment data	memory and	nlace resul	t in ACC.	skin if ()	
Description					-	by 1. If the result is 0, the next
Description		•		•		ulator. The data memory re-
	-			-		fetched during the current in-
	instruction exect					replaced to get the proper
Operation	Skip if ([m]+1)=					
Affected flag(s)		0, 7100 (([iii	1. 1)			
	TO PE	F OV	Z	AC	С	
0.177				1		1
SNZ [m].i	Skip if bit i of th		-	A (1)		
Description	•		•			n is skipped. If bit i of the data current instruction execution,
						instruction (2 cycles). Other-
	wise proceed w	ith the next in	struction (1	cycle).		
Operation	Skip if [m].i≠0					
Affected flag(s)]
	TO PE	F OV	Z	AC	С	
		-			—	



SUB A,[m]	Subtract	data mami	ory from the	0.000	latar	
Description	The spec		nemory is s			ontents
Operation	$ACC \leftarrow A$.CC+[m]+1	ł			
Affected flag(s)						
	то	PDF	OV	Z	AC	С
		_	\checkmark	\checkmark	\checkmark	\checkmark
SUBM A,[m]	Subtract	data memo	ory from the	e accumu	lator	
Description		ified data n he data me	nemory is s emory.	subtracted	l from the c	ontents
Operation	$[m] \leftarrow AC$	C+[m]+1				
Affected flag(s)						
	ТО	PDF	OV	Z	AC	С
	_		\checkmark	\checkmark	\checkmark	\checkmark
SUB A,x	Subtract	mmediate	data from	the accun	nulator	
Description			specified b It in the ac	•		ted from
Operation	$ACC \leftarrow A$	CC+x+1				
Affected flag(s)						
	ТО	PDF	OV	Z	AC	С
			\checkmark		\checkmark	
SWAP [m]	Swap nib	bles within	the data n	nemory		
Description		order and h nterchang	nigh-order i ed.	nibbles of	the specifi	ed data
Operation	[m].3~[m]	.0 ↔ [m].7	'~[m].4			
Affected flag(s)						
	ТО	PDF	OV	Z	AC	С
		_	—	—	—	—
SWAPA [m]	Swap dat	a memory	and place	result in t	he accumu	lator
Description						
Description			nigh-order r accumulat		•	ed data n
Operation	ing the re ACC.3~A		accumulat n].7~[m].4		•	ed data n
·	ing the re ACC.3~A	sult to the $CC.0 \leftarrow [n]$	accumulat n].7~[m].4		•	ed data n
Operation	ing the re ACC.3~A	sult to the $CC.0 \leftarrow [n]$	accumulat n].7~[m].4		•	ed data n



SZ [m]	SKID II DAI	a memory	is 0			
Description	If the cont the currer	ents of the nt instructi	especified	on, is disc	arded and	l a dumm
Operation	Skip if [m]	=0				
Affected flag(s)						
	ТО	PDF	OV	Z	AC	С
			—		—	
SZA [m]	Move data	a memory	to ACC, s	kip if 0		
Description	0, the follo	owing inst nmy cycle	specified d ruction, fei is replaced ction (1 cyc	ched duri d to get the	ng the cur	rent instru
Operation	Skip if [m]	=0				
Affected flag(s)	TO			7	40	0
	ТО	PDF	OV	Z	AC	С
SZ [m].i	Skip if bit	i of the da	ta memory	/ is 0		
Description Operation	If bit i of th instructior tion (2 cyc Skip if [m]	n execution cles). Othe	n, is discar	ded and a	dummy cy	cle is repl
Affected flag(s)	ТО	PDF	OV	Z	AC	С
	10	FDF	00	2	AC	
					_	Ū
		_				_
TABRDC [m]	 Move the	 ROM cod	e (current	 page) to T		
TABRDC [m] Description		yte of ROI	A code (cu	rrent page	BLH and	 data mem ed by the t
	The low by to the spe [m] ← RO	yte of RON cified data M code (N	A code (cu a memory	rrent page and the hi	BLH and	 data mem ed by the t
Description	The low by to the spe $[m] \leftarrow RO$ TBLH $\leftarrow F$	yte of ROI cified data M code (I ROM code	M code (cu a memory ow byte) e (high byte	rrent page and the hig	BLH and of addresse	data mem ed by the t
Description Operation	The low by to the spe [m] ← RO	yte of RON cified data M code (N	M code (cu a memory ow byte)	rrent page and the hi	BLH and	 data mem ed by the t
Description Operation	The low by to the spe $[m] \leftarrow RO$ TBLH $\leftarrow F$	yte of ROI cified data M code (I ROM code	M code (cu a memory ow byte) e (high byte	rrent page and the hig	BLH and of addresse	data mem ed by the t
Description Operation	The low by to the spe [m] ← RO TBLH ← I TO 	yte of ROI cified data M code (I ROM code PDF	M code (cu a memory ow byte) e (high byte	rrent page and the hi e) Z	BLH and of addresse gh byte tra	data mem ed by the t insferred C
Description Operation Affected flag(s)	The low by to the specific to the specific the specific term of the specific term of the specific term of the specific term of the low between terms and the low between terms are specific terms and the low between terms are specific terms are specifis	yte of ROI cified data M code (I ROM code PDF 	M code (cu a memory ow byte) e (high byte OV 	rrent page and the hi e) Z () e) to TBLI st page) a	BLH and of addresse gh byte tra	C C memory c
Description Operation Affected flag(s)	The low by to the spe $[m] \leftarrow RO$ TBLH $\leftarrow F$ TO — Move the The low b the data n $[m] \leftarrow RO$	yte of ROI cified data M code (I ROM code PDF 	A code (cu a memory ow byte) e (high byte OV e (last pag M code (la nd the high	rrent page and the hid e) Z (e) to TBLI st page) a byte trans	BLH and of addresse gh byte tra	C C memory c
Description Operation Affected flag(s) TABRDL [m] Description	The low by to the spe $[m] \leftarrow RO$ TBLH $\leftarrow F$ TO — Move the The low b the data n $[m] \leftarrow RO$	yte of ROI cified data M code (I ROM code PDF 	M code (cu a memory ow byte) e (high byte OV e (last pag M code (la nd the high ow byte)	rrent page and the hid e) Z (e) to TBLI st page) a byte trans	BLH and of addresse gh byte tra	C C memory c
Description Operation Affected flag(s) TABRDL [m] Description Operation	The low by to the spe $[m] \leftarrow RO$ TBLH $\leftarrow F$ TO — Move the The low b the data n $[m] \leftarrow RO$	yte of ROI cified data M code (I ROM code PDF 	M code (cu a memory ow byte) e (high byte OV e (last pag M code (la nd the high ow byte)	rrent page and the hid e) Z (e) to TBLI st page) a byte trans	BLH and of addresse gh byte tra	C C memory c

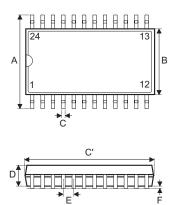


XOR A,[m]	Logical X	OR accum	ulator with	n data mer	nory	
Description	Data in the accumulator and the indicated data memory p sive_OR operation and the result is stored in the accumula					
Operation	$ACC \leftarrow A$	CC "XOR	" [m]			
Affected flag(s)						
	ТО	PDF	OV	Z	AC	С
				\checkmark		
XORM A,[m]	Logical X	OR data n	nemory wi	th the accu	umulator	
Description				mory and t is stored		•
Operation	$[m] \leftarrow AC$	C "XOR"	[m]			
Affected flag(s)						
	ТО	PDF	OV	Z	AC	С
		_	_	\checkmark	_	
XOR A,x	Logical X	OR immed	diate data	to the accu	umulator	
Description	Data in the accumulator and the specified data perform a bitwise logical Exclusive_OR eration. The result is stored in the accumulator. The 0 flag is affected.					
Operation	$ACC \leftarrow A$	CC "XOR	″ x			
Affected flag(s)						
	то	PDF	OV	Z	AC	С
		_		\checkmark		



Package Information

24-pin SOP (300mil) Outline Dimensions

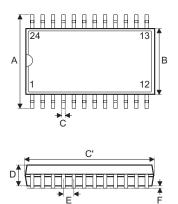




Cumula al	Dimensions in mil					
Symbol	Min.	Nom.	Max.			
A	394		419			
В	290		300			
С	14		20			
C'	590		614			
D	92		104			
E		50	_			
F	4					
G	32		38			
н	4		12			
α	0°		10°			



24-pin SSOP (150mil) Outline Dimensions



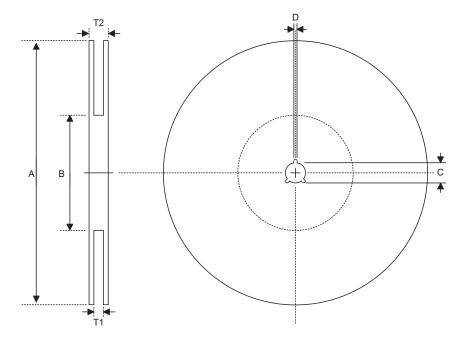


Counch al	Dimensions in mil					
Symbol	Min.	Nom.	Max.			
А	228		244			
В	150	_	157			
С	8		12			
C'	335		346			
D	54		60			
E	_	25	_			
F	4		10			
G	22		28			
Н	7	_	10			
α	0°		8°			



Product Tape and Reel Specifications

Reel Dimensions



SOP 24W

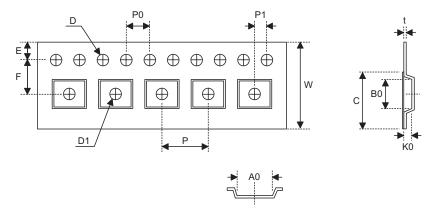
Symbol	Description	Dimensions in mm
А	Reel Outer Diameter	330±1.0
В	Reel Inner Diameter	62±1.5
С	Spindle Hole Diameter	13.0+0.5 _0.2
D	Key Slit Width	2.0±0.5
T1	Space Between Flange	24.8+0.3 0.2
T2	Reel Thickness	30.2±0.2

SSOP 24S (150mil)

Symbol	Description	Dimensions in mm
А	Reel Outer Diameter	330±1.0
В	Reel Inner Diameter	62±1.5
с	Spindle Hole Diameter	13.0+0.5 _0.2
D	Key Slit Width	2.0±0.5
T1	Space Between Flange	16.8+0.3 0.2
T2	Reel Thickness	22.2±0.2



Carrier Tape Dimensions



SOP 24W

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	24.0±0.3
Р	Cavity Pitch	12.0±0.1
E	Perforation Position	1.75±0.1
F	Cavity to Perforation (Width Direction)	11.5±0.1
D	Perforation Diameter	1.55+0.1
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	10.9±0.1
B0	Cavity Width	15.9±0.1
K0	Cavity Depth	3.1±0.1
t	Carrier Tape Thickness	0.35±0.05
С	Cover Tape Width	21.3

SSOP 24S (150mil)

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	16.0+0.3 _0.1
Р	Cavity Pitch	8.0±0.1
Е	Perforation Position	1.75±0.1
F	Cavity to Perforation (Width Direction)	7.5±0.1
D	Perforation Diameter	1.5+0.1
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	6.5±0.1
B0	Cavity Width	9.5±0.1
K0	Cavity Depth	2.1±0.1
t	Carrier Tape Thickness	0.3±0.05
С	Cover Tape Width	13.3



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)

43F, SEG Plaza, Shen Nan Zhong Road, Shenzhen, China 518031 Tel: 0755-8346-5589 Fax: 0755-8346-5590 ISDN: 0755-8346-5591

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

Holmate Semiconductor, Inc. (North America Sales Office)

46712 Fremont Blvd., Fremont, CA 94538 Tel: 510-252-9880 Fax: 510-252-9885 http://www.holmate.com

Copyright © 2005 by HOLTEK SEMICONDUCTOR INC.

The information appearing in this Data Sheet is believed to be accurate at the time of publication. However, Holtek assumes no responsibility arising from the use of the specifications described. The applications mentioned herein are used solely for the purpose of illustration and Holtek makes no warranty or representation that such applications will be suitable without further modification, nor recommends the use of its products for application that may present a risk to human life due to malfunction or otherwise. Holtek's products are not authorized for use as critical components in life support devices or systems. Holtek reserves the right to alter its products without prior notification. For the most up-to-date information, please visit our web site at http://www.holtek.com.tw.