HT44R70 SPECIFICATION

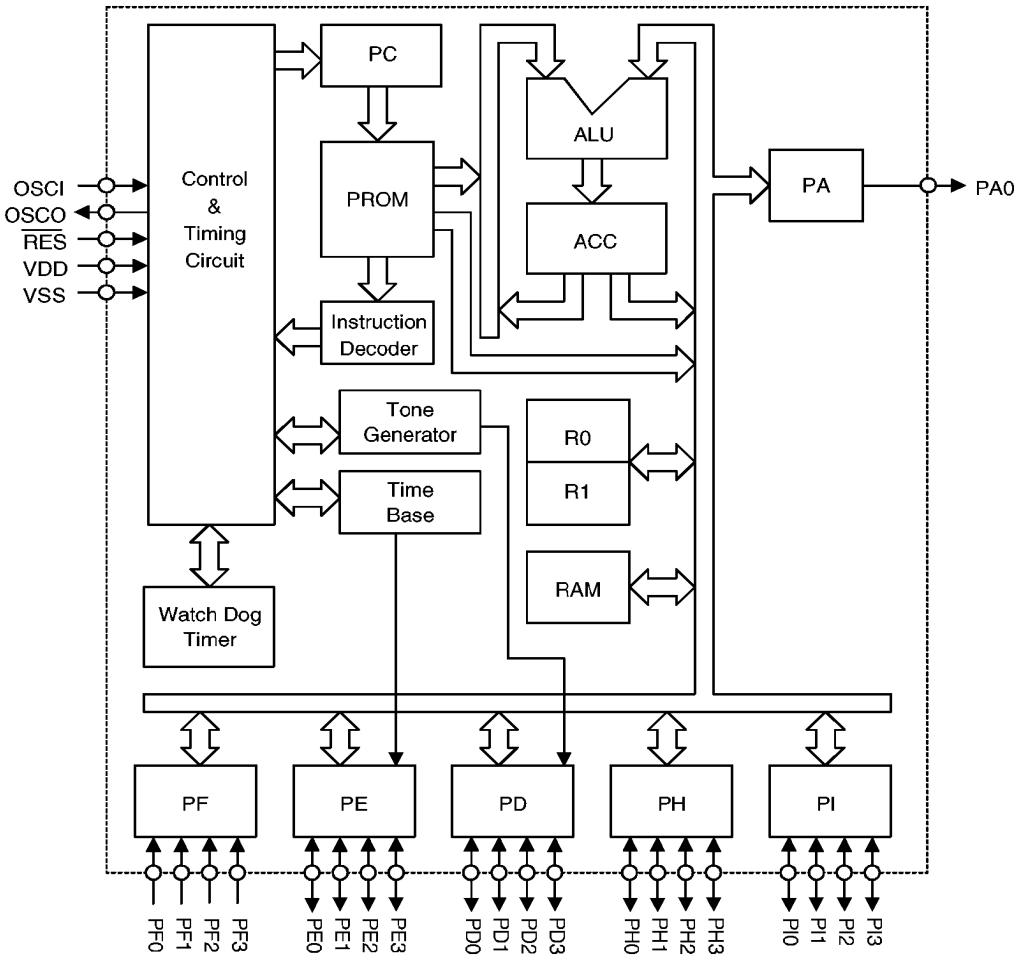
Features

- Operating voltage: 3.3V~5V
- 16 bidirectional I/O lines
- 4 input lines
- 4-bit parallel processor ALU
- 2K × 9 program memory (PROM)
- 14 × 9 option memory (PROM)
- 64 × 4 data memory (RAM)
- 2 working registers
- RC or crystal oscillator
- One high driving output line with or without a carrier
- Input port with latch capability
- Halt function to reduce power consumption, and wake-up feature
- Watch dog timer
- 77 powerful instructions
- Up to 2μs instruction cycle with 2MHz system clock at VDD=5V
- All instructions in 1 or 2 machine cycles
- 8-bit table read instruction
- Bit manipulation instruction
- All options defined by PROM

General Description

The HT44R70 is a 4-bit stand alone single chip microcontroller specifically designed for multiple I/O product applications. The device is particularly suitable for use in products such as remote controllers, fan/light controllers, washing machine controllers, scales, toys and various subsystem controllers. The combination of a program memory PROM allows the device to be used for product development. All hardware options are replaced by ROM code which is stored in PROM. This can also be electrically programmed.

Block Diagram



Note:

ACC: Accumulator

PC: Program counter

R0,R1: Working registers

PA0: Output line with or without a carrier

PD,PE,PH,PI: I/O ports

PF: Input port

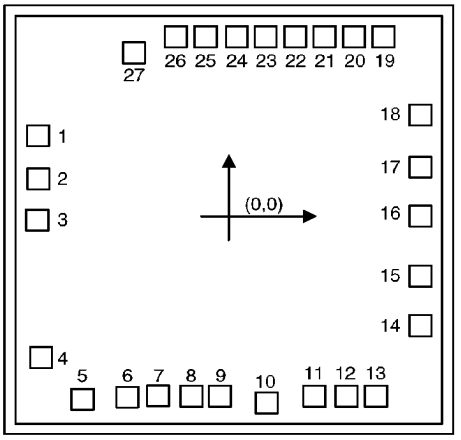
PROM: Program and option memory

Pad Description

Pad No.	Pad Name	I/O	ROM Code Option	Function
1	PA0	O	Level output or 1/2 duty carrier output	1-bit latch for output only, with high driving capacity and carrier output capability The PROM programming identification code is received serially from this pad. During PROM programming and code verification, the signal on PA0 is used to generate the programming control signal (the $\overline{RES}$ pad must be held low).
2	VDD	I	—	Positive power supply
3 12 21 11	PH0 PH1 PH2 PH3	I/O	CMOS open drain pull-high wake-up	4-bit bidirectional I/O port This port can be configurated as a bidirectional I/O by instructions. Also can be used to wake-up the microcomputer from the halt mode. The input configuration is a schmitt trigger input.
4 5	OSCO OSCI	O I	RC crystal	OSCI and OSCO are connected to a resistor or crystal to generate the system clock. The OSCI pad is the clock for receiving the serial identification code on the PA0 pad and combines with the PA0 pad to generate the programming control signal (the $\overline{RES}$ pad must be held low).
6	VSS	I	—	Negative power supply, GND
7	$\overline{RES}$	I	—	Input with a pull-up resistor for resetting the internal LSI It is used to initialize the processor and activated on a low-going edge. During the PROM programming and code verification, the $\overline{RES}$ pad must be held low.
22~23 8~9	PI0~PI3	I/O	CMOS open drain pull-high	4-bit bidirectional I/O port The port can be configurated as a bidirectional I/O by instructions. The input configuration is a schmitt trigger input.
10	NC	—	—	Test pad This pad must be left open when the HT44R70 is in normal operation.

Pad No.	Pad Name	I/O	ROM Code Option	Function
13~16	PD0~PD3 (AD0~AD3)	I/O	CMOS open drain pull-high single tone	4-bit bidirectional I/O port The port can be configurated as a bidirectional I/O by instructions. The input configuration is a schmitt trigger input. PD2,PD3 can drive a pair of inverted signals (by ROM code option). The port is the address/data bus (AD0~AD3) during the PROM programming and code verification
17~20	PE0~PE3 (AD4~AD7)	I/O	CMOS open drain pull-high wake-up timebase (PE3)	4-bit bidirectional I/O port The port can be configurated as a bidirectional I/O by instructions. The input configuration is a schmitt trigger input; also can be used to wake-up the microcomputer from a halt mode. PE3 has an additional time-base option. The port is the address/data bus (AD4~AD7) during the PROM programming and code verification
24~27	PF0~PF3 (AD8~AD10 VPP)	I	Pull-high latch wake-up	4-bit schmitt trigger input port Bit 0 to bit 2 of PF are used for the address bus (AD8~ AD10), bit 0 is used for the data bus (AD8), and bit 3 of PF is the programming supply voltage (VPP) input during the PROM programming and code verification

Pad Coordinates & Position



Chip size: 2790×3010' $(\mu m)^2$

* The IC substrate should be connected to VSS in the PCB layout artwork.

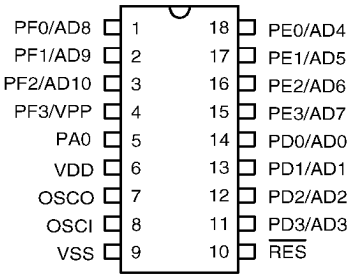
Unit: μm

Pad No.	Pad Name	X	Y	Pad No.	Pad Name	X	Y
1*	PA0	–1190.45	542.45	15*	PD1	1195.05	–402.15
2*	VDD	–1188.75	252.15	16*	PD0	1195.05	4.85
3	PH0	–1195.05	–26.25	17*	PE3	1195.05	334.05
4*	OSCO	–1171.75	–943.55	18*	PE2	1195.05	680.95
5*	OSCI	–913.85	–1224.95	19*	PE1	964.65	1204.45
6*	VSS	–633.25	–1212.85	20*	PE0	782.95	1204.45
7*	$\overline{\text{RES}}$	–443.95	–1200.05	21	PH2	597.95	1204.45
8	PI2	–245.95	–1204.45	22	PI1	416.25	1204.45
9	PI3	–58.75	–1204.45	23	PI0	231.25	1204.45
10	NC	237.65	–1248.85	24*	PF0	50.55	1204.45
11	PH3	534.15	–1204.45	25*	PF1	–145.85	1204.45
12	PH1	733.45	–1204.45	26*	PF2	–333.05	1204.45
13*	PD3	918.45	–1204.45	27*	PF3	–592.35	1097.45
14*	PD2	1195.05	–731.35				

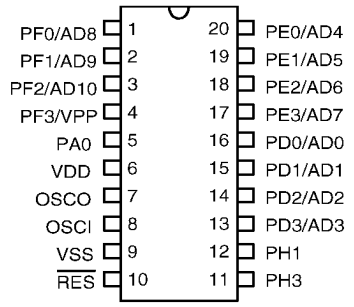
* These pins must be bonded out for functional testing.

Package & Pin Assignment

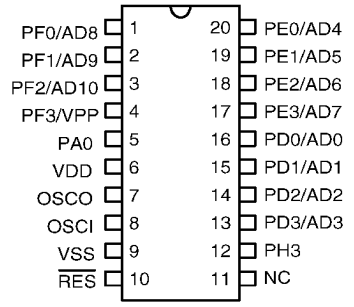
18 Pin DIP package



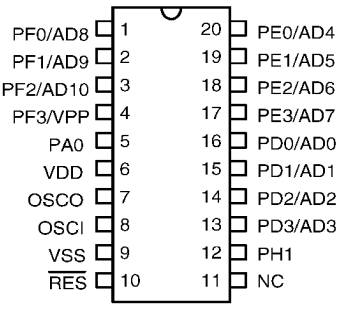
20 Pin DIP/SOP-A package



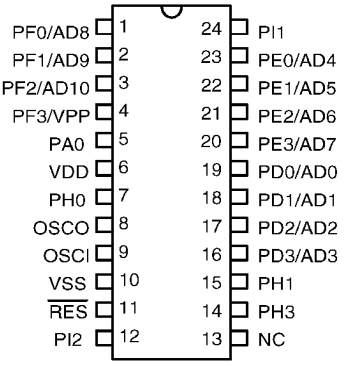
20 Pin DIP/SOP-B package



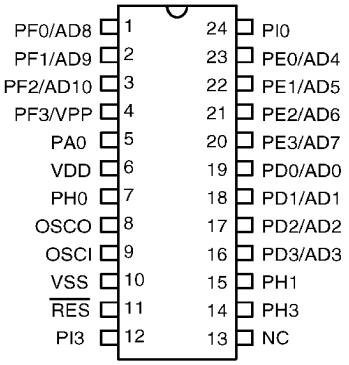
20 Pin DIP/SOP-C package



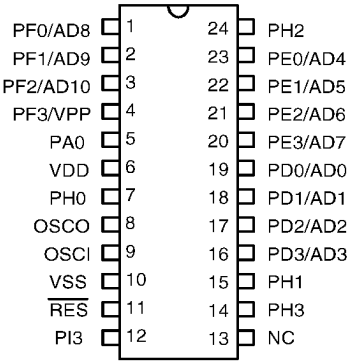
24 Pin SOP/SKINNY-A package



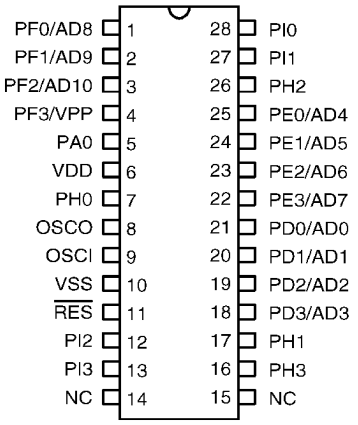
24 Pin SOP/SKINNY-B package



24 Pin SOP/SKINNY-C package



28 Pin SOP/SKINNY package



Absolute Maximum Ratings

Parameter	Symbol	Minimum	Maximum	Unit
Supply Voltage	V_{DD}	–0.3	5.5	V
Input Voltage	V_I	V_{SS}–0.3	V_{DD}+0.3	V
Storage Temperature	T_{STG}	–50	125	°C
Operating Temperature	T_{OP}	–25	75	°C

D.C. Characteristics

(Ta=25°C)

Symbol	Parameter	Test Condition		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
V_{DD}	Operating voltage	—	—	3.3	—	5	V
I_{DD}	Operating current	3.3V	No load, f_{SYS}=2MHz	—	1	2.4	mA
		5V		—	1.5	3	mA
I_{STB}	Stand-by current	3.3V	No load, HALT mode	—	—	1	μA
		5V		—	—	2	μA
V_{IL}	Input low voltage	3.3V	—	0	—	0.2V_{DD}	V
		5V	—	0	—	0.2V_{DD}	V
V_{IH}	Input high voltage	3.3V	—	0.8V_{DD}	—	3.3	V
		5V	—	0.8V_{DD}	—	5.0	V
I_{OL1}	Port PA0 output sink current	3.3V	V_{OL}=0.33V	1	—	—	mA
		5V	V_{OL}=0.5V	2.5	—	—	mA
I_{OH1}	Port PA0 output source current	3.3V	V_{OH}=2.97V	–1	—	—	mA
		5V	V_{OH}=4.5V	–2.5	—	—	mA
I_{OL2}	PD, PE, PH, PI output sink current	3.3V	V_{OL}=0.33V	1.3	—	—	mA
		5V	V_{OL}=0.5V	2.5	—	—	mA
I_{OH2}	PD, PE, PH, PI output source current	3.3V	V_{OH}=2.97V	–0.6	—	—	mA
		5V	V_{OH}=4.5V	–1.3	—	—	mA
R_{PH}	Pull-high resistance	3.3V	PD, PE, PF, $\overline{RES}$, PH, PI	20	—	150	KΩ
		5V		10	—	75	KΩ
V_{PP}	Programming supply voltage	5V	—	11	12.5	13	V

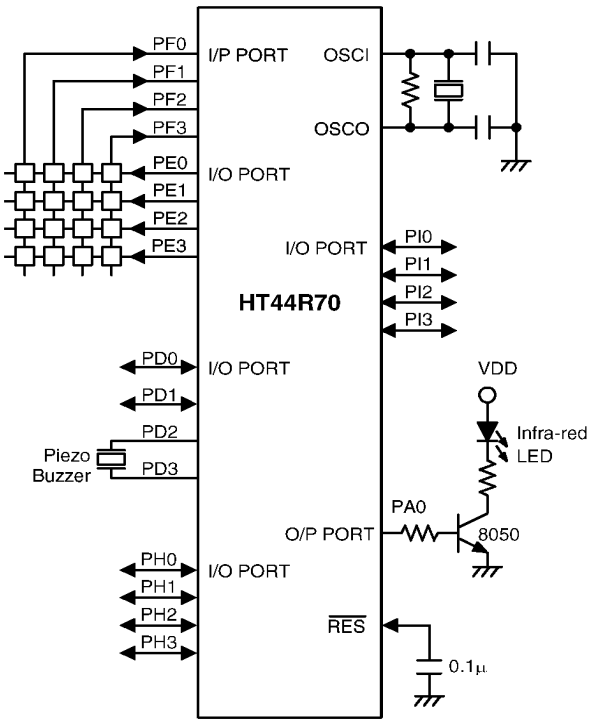
A.C. Characteristics

(Ta=25°C)

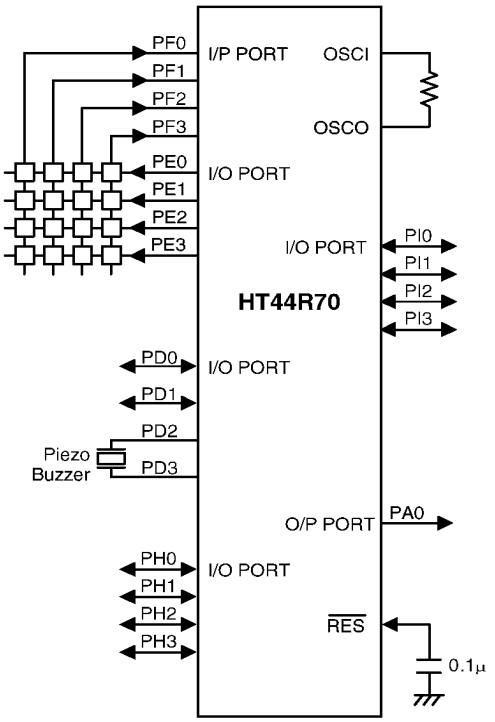
Symbol	Parameter	Test Condition		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
f_{SYS}	System clock	3.3V	RC oscillator	20	—	800	KHz
		5V		20	—	2000	KHz
		3.3V	Crystal oscillator	20	—	2000	KHz
		5V		20	—	2000	KHz
t_{CY}	Cycle time	—	f_{SYS}=2MHz	—	2.0	—	µs
t_{RES}	Reset pulse width	—	—	5	—	—	ms
t_{VPS}	VPP setup time	5V	—	0.2	—	—	µs
t_{AS}	Address setup time to OSCI and PA0	5V	—	0	—	—	µs
t_{DS}	Data setup time to OSCI in write cycle	5V	—	0	—	—	µs
t_{AH}	Address hold time after OSCI and PA0	5V	—	0.02	—	—	µs
t_{DH}	Data hold time after OSCI in write cycle	5V	—	0.02	—	—	µs
t_{DLY}	Delay between OSCI and PA0 in address latch period	5V	—	—	—	0.02	µs
t_{ALP}	Address latch pulse width	5V	—	0.1	—	—	µs
t_{WRP}	Data write pulse width	5V	—	500	—	—	µs
t_{RDP}	Data read pulse width	5V	—	0.1	—	—	µs
t_{BF}	A/D bus floating time in read cycle	5V	—	0	—	—	µs
t_{DD}	Data valid delay from PA0 in read cycle	5V	—	—	—	0.2	µs
t_{DF}	Data floating delay after PA0 in read cycle	5V	—	—	—	0.1	µs
t_{PA0H}	PA0 hold time after OSCI falling in programming mode	5V	—	0.02	—	—	µs

Application Diagram

Crystal system clock for remote controller applications



RC system clock for multiple I/O applications

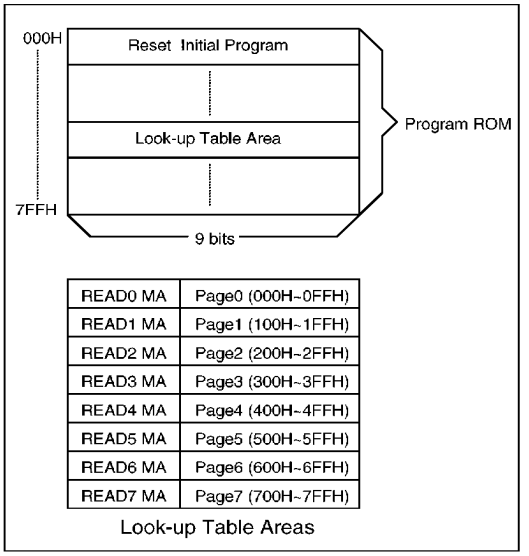


SYSTEM ARCHITECTURE

Program Counter - PC

This 11 bit binary counter which addresses the program memory (PROM) is organized from PC0 to PC10 and can specify a maximum of 2048 addresses. The program counter (PC) is incremented by 1 or 2 with each execution of an instruction. When executing the jump instructions (JMP, JNZ, JC, JZ...) or an initial reset, the program counter (PC) will be loaded with the corresponding address data. For jump instructions and branch instructions the address space is capable of specifying 2048 addresses directly (jump bit instructions excluded).

Note: PC10~PC0: Bits of instruction code
P10~P0: Program location
* is the current page number.



Program Memory

Program Memory - PROM

The program memory is used to store the executed program and non-volatile data. It is organized into 2048×9 bits and is addressed by the program counter. The PROM can be electrically programmed making it a very powerful tool for product development. There are some special locations in program memory described as follows:

Location 0:
Activating the $\overline{RES}$ pin of the processor causes the first instruction to be fetched from location 0. Care must be taken when preparing the program.

Table location:
The look-up table can be located in any ROM location. The instruction READn MA is available for reading the table and transferring the table data to the ACC and data memory addressed by register pair R1,R0. Bit 8 of the table data cannot be accessed by the table read instruction.

Working Registers - R1,R0

The working registers consist of register R0 which is of 4-bits wide and register R1 which is of 2-bits wide. They are usually used to store the frequently accessed intermediate results. Register R0 can increment (+1) or decrement (-1). The register pair R1, R0 can be used as data memory, or effectively used as a data memory pointer when the data memory transfer instructrion is executed.

Mode	Program Counter										
	PC10	**PC9**	**PC8**	**PC7**	**PC6**	**PC5**	**PC4**	**PC3**	**PC2**	**PC1**	**PC0**
Initial reset	0	0	0	0	0	0	0	0	0	0	0
Jump, Jump carry, Jump zero	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
Jump bit	*	*	*	P7	P6	P5	P4	P3	P2	P1	P0

Program Counter

Data Memory - RAM

The static data memory (RAM) is organized with 64×4 bits and is used to store data. The data memory can be directly accessed by MOV A,[XXH], MOV [XXH],A and be indirectly addressed through the working register pair R1,R0. Each bit of data memory can be set or reset by instructons which are useful during data manipulation. The data memory may be affected by binary addition, logical operations, increment and decrement operations, data memory movements, and bit manipulations.

Accumulator - ACC

The accumulator is the most important data register in data operation and control. It is one of the sources of input to the ALU and the destination of the result of operations performed in the ALU. Data transfers between I/O ports and memory may also pass through the accumulator.

Arithmetic and Logic Unit - ALU

This circuit performs arithmetic and logical operation. The ALU provides the following functions ... add with or without carry flag, AND, OR, Exclusive OR, Complement, Rotate, Increment, Decrement, Data transfer, Branch decision. The ALU not only outputs the results of data operations but also sets the status of the carry flag (CF) and zero flag (ZF) in some instructions.

Initial Reset

The HT44R70 provides an $\overline{RES}$ pin for system initialization. This reset pin has an internal pull-high resistor. In order to guarantee all circuits are reset it is combined with an external 0.1uF capacitor to provide an internal reset pulse. If the reset pulse is generated externally the $\overline{RES}$ pin must be held low for at least 5ms. The $\overline{RES}$ pin must be held low during PROM programming and code verification. The reset performs the following functions...

- Sets the program counter (PC) to 000H.
- Set all bidirectional pins to tri-state (disable output).
- PA0 set to high (level option) or low without a carrier output (carrier option)
- Watch dog timer starts counting.

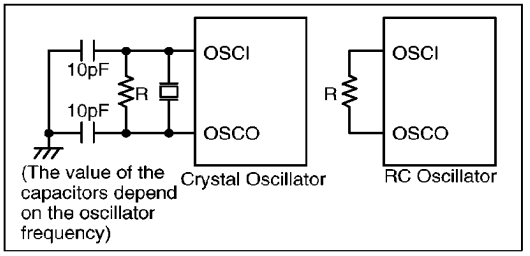
Note: During a reset PA0 will be inhibited and a pull-low resistor is added.

Oscillator Circuit

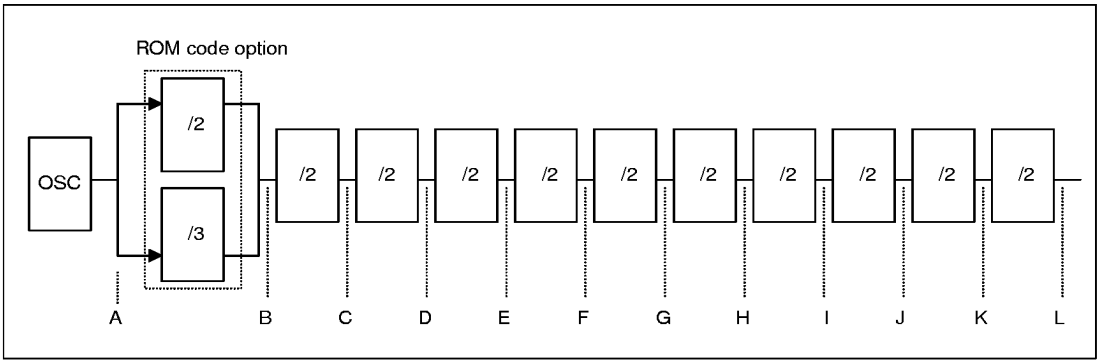
The HT44R70 system clock oscillation circuit can be a crystal or RC oscillator by ROM code option. For the crystal oscillator a crystal or ceramic resonator connected across OSCI and OSCO provides the feedback and phase shift required for oscillation. If an accurate frequency is not required a ceramic resonator may be used in place of the crystal but the external capacitors must be changed. For the RC oscillator only a resistor across OSCI and OSCO is necessary because the IC contains an internal capacitor. The system clock can be obtained from the oscillator circuit. Behind the oscillator circuit there is a 6 stage counter. All the output can be used as a system clock. In these stages the first stage is different from others which has a divide-by-2 or divide-by-3 option. If the system clock is not obtained from the first stage different first stages will get various system clocks. The system clock can be calculated by the following equation:

$$\text{System clock} = \frac{\text{XTAL (RC)}}{2^{(6-n)} * k}$$

Where n ranges from 0 to 6 and k can be 1 or 1.5 by ROM code option. If n is selected as "6", k can only be 1. XTAL(RC) is the oscillator frequency. The HT44R70's machine cycle consists of a sequence of 4 states numbered T1 to T4. Each state lasts for one oscillator period. The system



Oscillator



Prescaler

oscillator frequency ranges from 20KHz to 2MHz. The machine cycle is 2μs if the system clock is 2MHz. During PROM programming and code verification the programming control signal is produced by combining the signal on the OSCI and PA0 pins.

Prescaler

There is one prescaler for the HT44R70 to generate the system clock, the carrier signal, WDT's clock and the single tone signal. The first stage of the prescaler can be selected /2 or /3. The system clock may come from A to G. The clock of WDT is derived from J to L. The single tone signal may come from H to L.

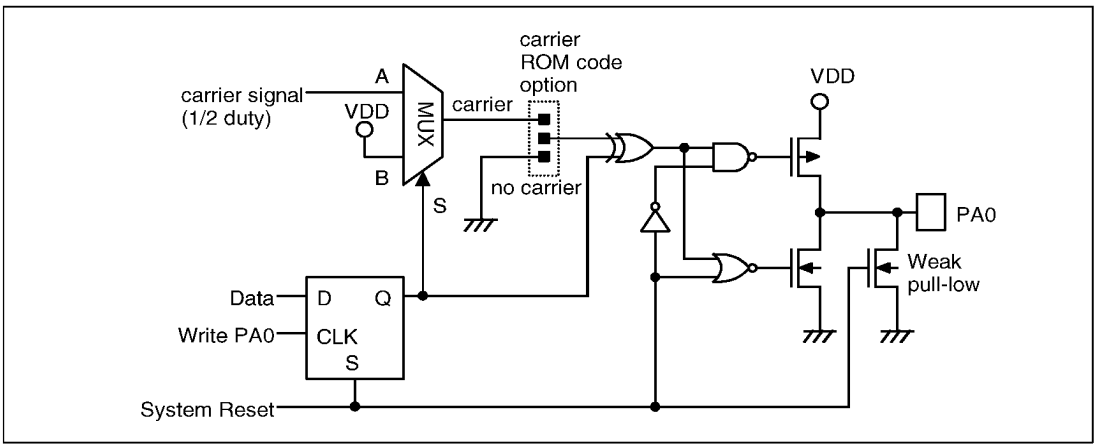
Output Line - PA0

The output line is configured as shown.

PA0 is bit 0 of port PA. It can be configured as a CMOS output with or without carrier driving capacity. The CMOS output is designed specialy for high current driving application. The carrier option is easy to interface with an infrared diode. The carrier frequency can be calculated by the following equation.

$$\text{Carrier frequency} = \frac{\text{XTAL (RC)}}{2^{(6-n)} * k}$$

where n ranges from 0 to 6 and k can be 1 or 1.5 by ROM code option. If n is selected as 6, k can only be 1. XTAL(RC) is the oscillator frequency.



Output Line PA0

In addition 1/2 duty cycle can be selected. Writing 0 to the PA0 latch (OUT PA,A, where bit 0 of ACC=0), results in a carrier output. Writing 1 to the PA0 latch (OUT PA,A, where bit 0 of ACC=1), keeps the state of PA0 at a normal low level. If the no carrier option is selected it functions as a normal output line. Writing "0" to the PA0 latch results in a normal low output. Otherwise writing "1" to the PA0 latch will keep a normal high output. No matter what option has been selected, during a system reset PA0 output will be inhibited and a pull-low resistor is visible. After a system reset, if the carrier option is selected then PA0 will stay at a low level. On the contrary, if the no carrier option is chosen, the PA0 will stay at a high level. In the PROM programming and code verification mode PA0 is used as an input pin. When the $\overline{RES}$ pin is forced low the HT44R70 is ready to receive serial data on the PA0 pin. The serial data on the PA0 pin is clocked by the OSCI input and used to activate the PROM programming and code verification mode. The serial data 1011 and 1001 will force the HT44R70 to program/verify the program PROM and option PROM respectively. After the PROM programming mode is activated, the PA0 pin is combined with the OSCI pin to produce the programming control signal.

Input/Output Ports - PD,PE,PH,PI

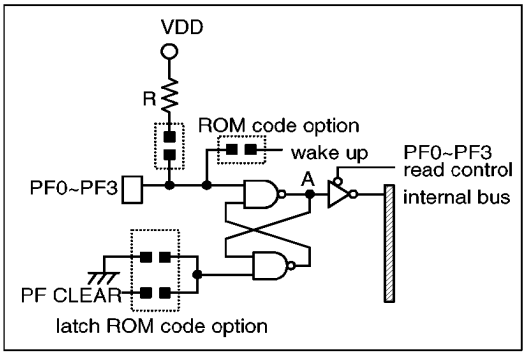
There are four physical I/O ports (PD,PE,PH,PI) with different configuration and ROM code options. These ports all have the same basic structure option, namely open drain NMOS output, CMOS output, and internal pull-high resistor.

With this basic configuration, if a line performs an input function, a TRI instruction can be invoked (writing "1" to the corresponding tri-state control latch). This makes the output part of the input/output port exhibit a floating state (in the case of no pull-high resistor option) to minimize the loading effect. After a power on reset, the tri-state control latch will be 1. It implies that all I/O lines are floating or high level if they have a pull-high resistor option. In the output application, three configurations can be selected, namely CMOS, NMOS or open drain output with or without pull-high resistors. They can also drive open collector or open drain outputs without the need for additional external pull-high resistors. In the situation where pull-high resistors are selected and interfaced with an external output circuit, a "1" must be written to the data latch to turn off the NMOS. This is to avoid a logical conflict. Bits 0~2 of the PE port are normal type configurations. PE3 has the time-base option and if selected, the input function will differ from other I/O line(s). The SET PE.3 and CLR PE.3 are read-modify-write instructions whose operation may get different results from those intended if the time-base is selected. PE also has a wake-up option (by forcing the terminal low) which may make the HT44R70 escape the HALT state and resume operation. The CPU can be used as an S/W timer by polling the time-base signal to measure its width or period.

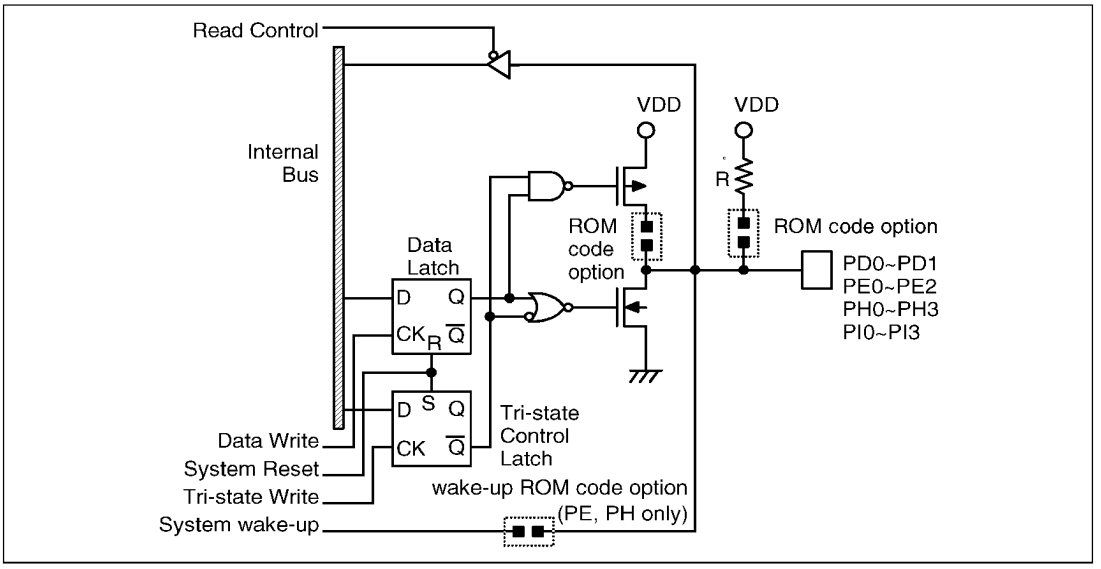
Bit 2 and bit 3 of PD have single tone options. If the single tone is selected, writing a "1" to the related latch will drive the output stage according to the internal single tone signal. The single tone in bit 2 is the inverse of bit 3, which is a useful feature in differential drive applications. Reading these two bits will get 0 if the single tone is selected. The set-bit or clear-bit instructions will lose their original specific functions. They can still be executed but cannot obtain results.

PD and PE are also used as the address/data (AD0~AD7) bus for PROM programming and code verification.

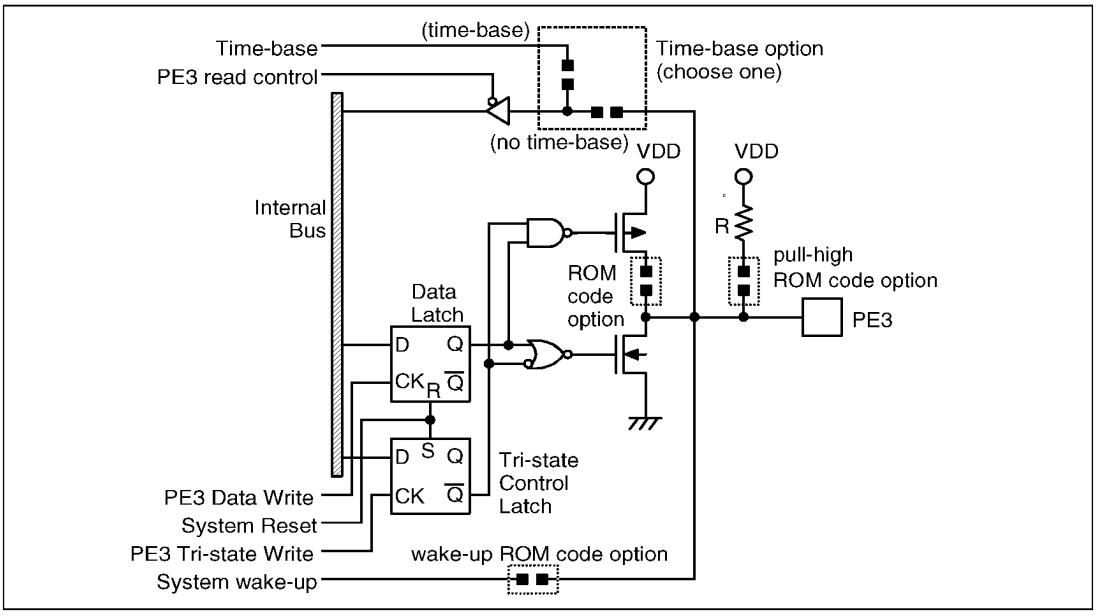
Input Port - PF



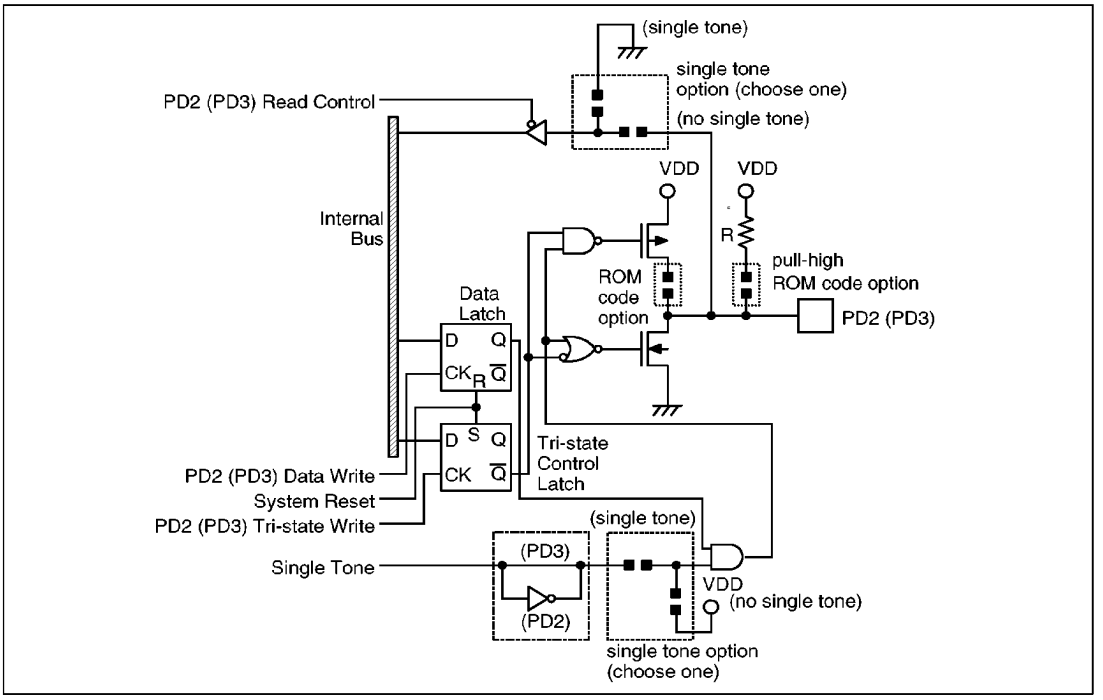
Input Port PF



I/O Ports Basic Structure



Port PE3

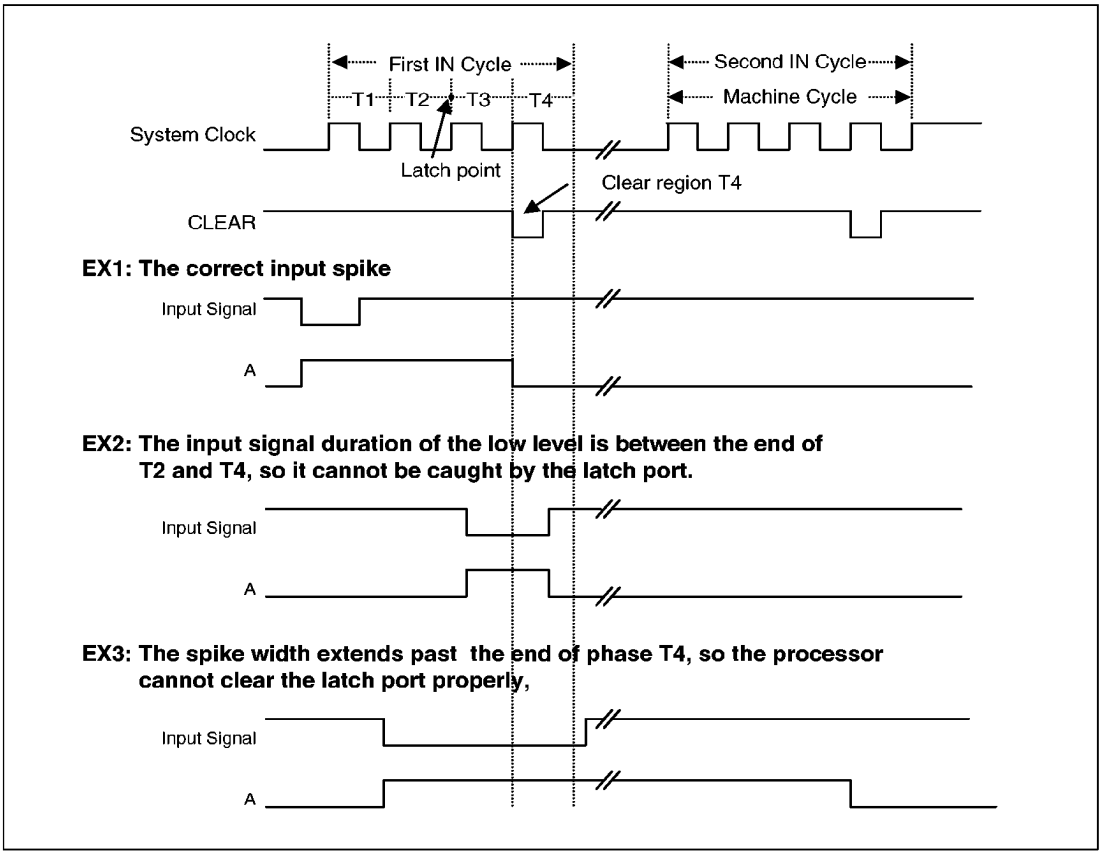


Port PD2(PD3)

The input port (PF) is configured as shown.

PF is an input port with 3 options: pull-high, wake-up and latch type inputs. If the ROM code option invokes the wake-up, a low level input will force the HT44R70 to resume operation and leave the HALT mode. In the input function the HT44R70 offers a normal type input or latch type input. When the normal type input is desired the input type control option must select VSS. If the latch type input is used the PF CLEAR option must be selected. Normal type input is the same as most input ports, which polls the state of the input port by using a CPU instruction. A latch input stores the port data by using a hardware latch. When the input is stimulated by a low-going pulse, the internal bit will be set to "0". After an IN A,PF instruction is executed and the input returned to "1", the internal bit will be reset to "1". A latch type input may reduce the CPU loading even if the expected spike will be caught which is impossible using software instructions. There are four phases in a machine cycle. If the port is configured as a latch type the processor will read the data in the latch at the second phase (T2) by executing the instruction IN A,PF. At the fourth phase (T4) of the same cycle, the processor will clear the latch port by setting the PF CLEAR to 0. Attention must be paid to two improper operations under latch type input configurations. Firstly if the low-going input has just occured in the executing input cycle and produced after phase T2 and ended before the end of the fourth phase, the input will be lost. Secondly if the low level duration of the input signal extends to the end of the input instruction's machine cycle the processor does not clear the latch port properly and the unexpected data will still be held in the latch producing a read error in the second input instruction. The figure shows examples of reading the data from the latch port.

During PROM programming and code verification, bits 0~2 of PF are used as the address



PF Timing Diagram

(AD8~AD10) buses, with bit 0 of PF used as the data bus (AD8) and the programming supply voltage (VPP) as the input from bit 3 of PF.

Tone Generator

The HT44R70 provides a single tone generator. The frequency of the tone signal is given by the following equation:

$$F_{tone} = \frac{XTAL\ (RC)}{2^{(11-n)} * k}$$

where n ranges from 0 to 4 by ROM code option. The value of k is determined by the prescaler option. XTAL (RC) is the oscillator frequency. The output terminal(s) of the tone generator depend upon the ROM code option. Bit 3 of port PD (PD3) can be optioned as an output of the tone generator and bit 2 of port PD (PD2) can be optioned as an inverting output of the tone generator. These can be chosen only by PD3 or both PD2 and PD3. Once PD3 or PD2 are configured as tone outputs, the input/output and set/clear operation in PD will not function as in the original spec. When optioned as a tone output reading the corresponding bit will get a "0".

If PD2 and PD3 are configured as tone outputs, writing "1" to PD3 will enable the tone outputs and writing "0" to PD3 will disable the tone outputs, and writing "1" or "0" to PD2 will not affect the tone outputs.

Watch Dog Timer

This timer is composed of an 8-stage count-up counter designed to prevent the program from jumping to an unknown or unwanted location. This prevents the application circuit from losing control with unpredictable results. The timer clock comes from the prescaler; also the carrier frequency option affects the clock. It can be summarised in the following equation:

$$\text{WDT clock frequency} = \frac{\text{XTAL (RC)}}{2^{(11-n)} * k}$$

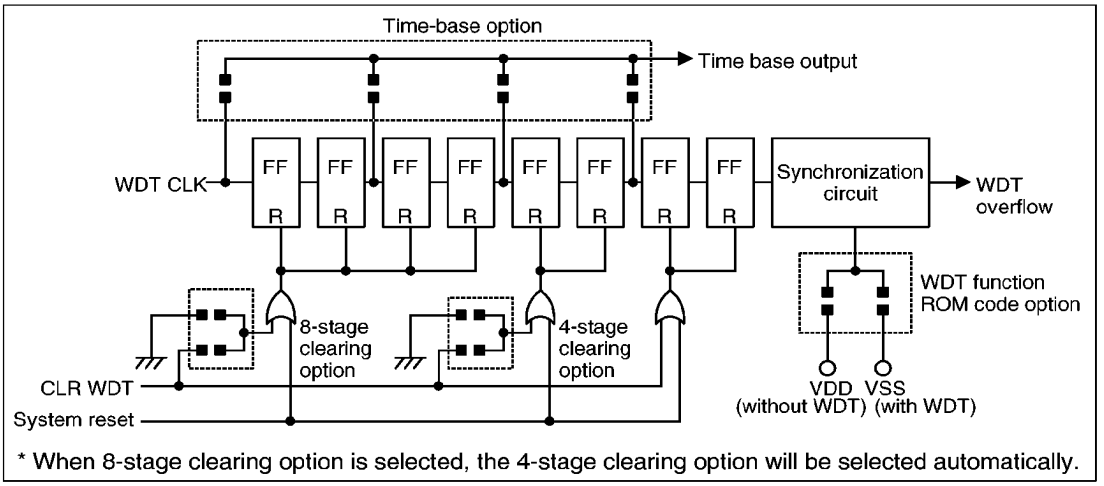
where n ranges from 0 to 2 by ROM code option, and XTAL (RC) is the oscillator frequency. The value of k is determined by the prescaler option. In normal operation, the application program will reset this timer by issuing a CLEAR WDT instruction before the timer overflows. If timer overflow occurs it implies that the operation is not under control. The HT44R70 will then perform a system reset to initialize the system. It has another option regarding the CLEAR WDT instruction — the clearing stage. The two or four stages in the MSB side or all eight stages can be optioned to be cleared whenever the CLEAR WDT is performed. In other word, the remaining stage(s) will keep its content(s) even if CLEAR WDT is executed. During the power on reset period the WDT will be cleared no matter what the clearing stage option is. After that the WDT will start counting to function as a watch dog. The user may disable the WDT by a ROM code option. The time base uses the same stage of the WDT, so restraining the WDT function occurs in the synchronization stage.

Time Base

The time base is used to generate the regular time base signal. Since it utilizes the same counter with WDT, the clearing stage option of WDT will affect the time base option selection. Normally, the frequency of the time base is the following:

$$\text{Time base frequency} = \frac{\text{CLOCK of WDT}}{2^{(6-n)}}$$

If 2 clearing stages of the WDT option are selected, the lower 6 stages still operate and n can be any value of 0,2,4 or 6. If 4 clearing stages of WDT option are adopted, the lower 4 stages operate and n can be 0,2, or 4. In the case of 8 clearing stages being chosen then n can only be 0. The time base signal can be polled in bit 3 of port PE by ROM code option. With the importing PE operation the user can check the status of the time base signal. In that way bit 3 of port PE can no longer reflect the state of PE3. Only the output function is



Watch Dog Timer

available and bit set/clear will be useless. For I/O ports operating in output mode, a reading operation will return the pad state.

Halt

When the HALT instruction is executed the system clock will be stopped and the system driven into a low power consumption state. The contents of the on-chip RAM and registers remain unchanged. The halt state can be terminated by a low level input to PE, PF, or PH (ROM code option) or a hardware reset.

It should be noted that when the halt state is terminated by a low level input of PE, PF, or PH the system will resume and execute the instruction right after the HALT instruction.

Options

Oscillator	Crystal or RC
	The system clock equation: System clock = $\frac{XTAL(RC)}{2^{(6-n)} * ko}$ where n ranges from 0 to 6 by mask option ko is the k value of the prescaler; if n=6 then ko=1, and XTAL(RC)=oscillator frequency
PD, PE, PH, PI	CMOS or NMOS open drain
	Pull-high resistor or no pull-high resistor
	Wake-up (PE,PH); each line of PE and PH can be selected to wake-up the microcomputer from the HALT state.
	Time-base input (PE3); the time-base signal will substitute the input line.
	Single tone output (PD2,PD3); the single tone signal will drive the output stage.
PF	Pull-high resistor or no pull-high resistor latch type
	Latch type
	Wake-up; each line of PF can be selected to wake-up the microcomputer from the HALT mode.
PA0	Carrier frequency eauation: Carrier frequency = $\frac{XTAL(RC)}{2^{(6-n)} * k_C}$ where n ranges from 0 to 6 by mask option k_C is the k value of the prescaler; if n=6 then k_C=1, and XTAL (RC)= oscillator frequency.
	With or without carrier option (1/2 duty cycle)

Watch dog timer	No watch dog timer
	The frequency of WDT's clock = $\frac{XTAL(RC)}{2^{(11-n)} * k}$ where n ranges from 0 to 2 by mask option XTAL(RC)= oscillator frequency
	Clearing stage option; the 2,4,8 can be selected.
Time-base	The frequency of time base = $\frac{CLOCK\ of\ WDT}{2^{(6-n)}}$ where n can be 0,2,4 or 6, which is also determined by the clearing stage option of WDT.
	PE3 input option; the time-base signal can be monitored in the PE3 input.
Tone	The frequency of tone = $\frac{XTAL(RC)}{2^{(11-n)} * k}$ where n ranges from 0 to 4 by mask option XTAL (RC)= oscillator frequency
	PD2 and PD3 output option; the tone signal can be exhibited in PD2 and/or PD3 with different phases.

Software Tools

The HT44R70 is similar to the HT447K0 and HT447P0 in most ways apart from size of ROM and RAM, number of I/O ports, the architecture of I/O ports and carrier option for PA0. The HT44R70 can be programmed and verified by using the HT44R70 writer provided by Holtek. The HT44R70 writer can perform the programming, verification, blank checking, ROM code check sum and option code reading tasks for the HT44R70. All of the ROM code files developed by the HT447K0 and HT447P0 can be converted into the HT44R70 version by using the utility files of the HT44R70 writer. Note that the 1/2 duty carrier option is available for the HT44R70, HT447K0 and HT447P0. The 1/4 duty carrier option is available only for the HT447K0 and HT447P0. For I/O ports operating in output mode, a reading operation will return the value in the output register and pad state for the HT447K0/HT447P0 and HT44R70 respectively. The maximum system clock frequency of the HT44R70 is 2MHz, and that of the HT447K0/447P0 is 4MHz.

INSTRUCTION SET

Instruction Set Summary

Mnemonic	Description	Word	Cycle	CF	ZF
Arithmetic					
ADD A,M	Add data memory to ACC	1	1	√	√
ADD M,A	Add ACC to data memory	1	1	√	√
ADC A,M	Add data memory with carry to ACC	1	1	√	√
ADC M,A	Add ACC with carry to data memory	1	1	√	√
ADD A,XH	Add immediate data to ACC	1	1	√	√
ANC A,XH	Add immediate data to ACC with CF not affected	2	2	—	√
CPL A	Complement ACC	1	1	—	√
CPL R1	Complement R1	1	1	—	√
Logic operation					
AND A,M	AND data memory to ACC	1	1	—	√
AND M,A	AND ACC to data memory	1	1	—	√
AND A,XH	AND immediate data to ACC	2	2	—	√
OR A,M	OR data memory to ACC	1	1	—	√
OR M,A	OR ACC to data memory	1	1	—	√
OR A,XH	OR immediate data to ACC	2	2	—	√
XOR A,M	Exclusive–OR data memory to ACC	1	1	—	√
XOR M,A	Exclusive–OR ACC to data memory	1	1	—	√
XOR A,XH	Exclusive–OR immediate data to ACC	2	2	—	√
Increment & Decrement					
INC R0	Increment register R0	1	1	—	√
INC M	Increment data memory	1	1	√	√
DEC R0	Decrement register R0	1	1	—	√
DEC M	Decrement data memory	1	1	√	√
Rotate					
RLC A	Rotate ACC left through the carry	1	1	√	—
RRC A	Rotate ACC right through the carry	1	1	√	—
Input & Output					
IN A,Pi	Input port–i to ACC, port–i=PD,PE,PF,PH,PI	1	1	—	√
OUT Po,A	Output ACC to port–o, port–o=PD,PE,PH,PI	1	1	—	—
TRI Pn,A	Output ACC to tri-state latch of port-n, port-n= PD,PE,PH,PI	1	1	—	—
OUT PA,A	Output ACC0 to port A	1	1	—	—

Mnemonic	Description	Word	Cycle	CF	ZF
Data Move					
MOV A,R0	Move R0 to ACC	1	1	—	√
MOV R0,A	Move ACC to R0	1	1	—	—
MOV A,R1	Move R1 to ACC	1	1	—	√
MOV R1,A	Move ACC to R1	1	1	—	—
MOV A,M	Move data memory to ACC	1	1	—	√
MOV M,A	Move ACC to data memory	1	1	—	—
MOV A,XH	Move immediate data to ACC	1	1	—	—
MOV R1R0,XXH	Move immediate data to R1 and R0	1	1	—	—
MOV R0,M	Move data memory to R0	1	1	—	—
MOV A,[XXH]	Move data memory to ACC directly	1	1	—	√
MOV [XXH],A	Move ACC to data memory directly	1	1	—	—
Branch					
JMP addr	Jump unconditionally	2	2	—	—
JC addr	Jump on carry=1	2	2	—	—
JNC addr	Jump on carry=0	2	2	—	—
JZ addr	Jump on zero flag=1	2	2	—	—
JB A.i,addr	Jump on A.i=1	2	2	—	—
JB Pm.i,addr	Jump on Pm.i=1, Pm=PE	2	2	—	—
JB M.i,addr	Jump on M(R1,R0). i=1	2	2	—	—
JNZ addr	Jump on zero flag=0	2	2	—	—
JNB A.i,addr	Jump on A.i=0	2	2	—	—
JNB M.i,addr	Jump on M(R1,R0). i=0	2	2	—	—
Miscellaneous					
HALT	Enter power down mode	1	2	—	—
NOP	No operation	1	1	—	—
Flag					
CLR C	Clear carry flag	1	1	0	—
SET C	Set carry flag	1	1	1	—
Table Read					
READn MA	Read page 0~7 of ROM code to M(R1,R0) & ACC	1	2	—	√
Bit Set/Reset					
SET M.i	Set bit of data memory	1	1	—	—
CLR M.i	Clear bit of data memory	1	1	—	—
SET Pn.i	Set bit of Pn.i, Pn=PD,PE,PI	1	1	—	—
CLR Pn.i	Clear bit of Pn.i, Pn=PD,PE,PI	1	1	—	—
Watch Dog					
CLEAR WDT	Clear watch dog timer	1	1	—	—

Instruction Definitions

ADC A,M Add data memory content and carry to accumulator

Machine code 0 0 0 1 0 0 0 0 0

Description The content of the data memory addressed by the register pair "R1,R0", the carry flag and the accumulator are added simultaneously. The result is stored in the accumulator. The carry and zero flags are affected.

Operation ACC ← ACC+M(R1,R0)+CF

ADC M,A Add accumulator and carry to data memory

Machine code 0 0 0 0 0 0 0 0 0

Description The content of the data memory addressed by the register pair "R1,R0", the carry flag and the accumulator are added simultaneously. The result is stored in the data memory. The carry and zero flags are affected.

Operation M(R1,R0) ← ACC+M(R1,R0)+CF

ADD A,M Add data memory to accumulator

Machine code 0 0 0 1 1 0 0 0 0

Description The content of the data memory addressed by the register pair "R1,R0" and the accumulator are added. The result is stored in the accumulator. The carry and zero flags are affected.

Operation ACC ← ACC+M(R1,R0)

ADD A,XH Add immediate data to accumulator

Machine code 0 0 1 1 1 d d d d

Description The immediate data and the accumulator content are added. The result is stored in the accumulator. The carry and zero flags are affected.

Operation ACC ← ACC+XH

ADD M,A Add accumulator to data memory

Machine code 0 0 0 0 1 0 0 0 0

Description The content of the data memory addressed by the register pair "R1,R0", and the accumulator content are added. The result is stored in the data memory. The carry and zero flags are affected.

Operation M(R1,R0) ← ACC+M(R1,R0)

ANC A,XH Add immediate data to ACC with CF not affected

Machine code 0 0 0 0 0 d d d d 0 0 0 0 0 0 0 1 1

Description The immediate data and the accumulator content are added. The result is stored in the accumulator. The zero flag is affected.

Operation ACC ← ACC+XH

AND A,M Logical AND data memory to accumulator

Machine code 0 0 0 0 1 0 0 1 1

Description Data in the accumulator and the data memory addressed by the register pair "R1,R0" performs the bitwise logical-AND operation and the result is stored in the accumulator. The zero flag is affected.

Operation ACC ← ACC "AND" M(R1,R0)

AND A,XH Logical AND accumulator with immediate data

Machine code 0 0 0 0 0 0 0 1 1 0 0 0 1 0 d d d d

Description Data in the accumulator and the specified data perform the bitwise logical–AND operation and the result is stored in the accumulator. The zero flag is affected.

Operation ACC ← ACC "AND" XH

AND M,A Logical AND accumulator to data memory

Machine code 0 1 1 0 0 0 0 0 0

Description Data in the accumulator and the data memory addressed by the register pair "R1,R0" performs the bitwise logical-AND operation and the result is stored in the data memory. The zero flag is affected.

Operation M(R1,R0) ← ACC "AND" M(R1,R0)

CLEAR WDT Clear watch dog timer

Machine code 0 1 1 1 0 0 0 0 1

Description The watch dog timer is cleared. The zero flag is not affected.

CLR C Clear carry flag

Machine code 0 1 1 1 0 0 0 1 0

Description The carry is reset to zero.

Operation CF ← 0

CLR M.i Clear bit of data memory

Machine code 0 0 0 1 1 i3 i2 i1 i0

i0~i3 are determined by operand i. The corresponding bit will be "0" if i is reset. Otherwise the bit is set to 1. For example, if i=0 then i3~i0=1110.

Description The specified bit of data memory addressed by register pair "R1,R0" is reset to zero.

Operation M(R1,R0).i ← 0

CLR Pn.i Clear bit of port

Machine code PD 0 0 0 1 0 i3 i2 i1 i0

PE 0 0 0 0 0 i3 i2 i1 i0

PI 1 0 0 1 0 i3 i2 i1 i0

i0~i3 are determined by operand i. The corresponding bit will be "0" if i is reset. Otherwise the bit is set to 1. For example, if i=0 then i3~i0=1110.

Description The specified bit of port "Pn" is reset to zero. Pn can be PD,PE,PI.

Operation Pn.i ← 0; Pn=PD,PE,PI

CPL A Complement accumulator

Machine code 0 0 0 1 1 1 1 1 1

Description Each bit of the accumulator is logically complemented. The zero flag is affected.

Operation ACC ← $\overline{\text{ACC}}$

CPL R1 Complement R1

Machine code 0 1 1 1 1 0 0 0 1

Description Each bit of the register R1 is logically complemented. The zero flag is affected.

Operation R1 ← $\overline{\text{R1}}$

DEC M Decrement data memory

Machine code 0 0 0 0 0 1 1 1 1

Description Data in the data memory specified by the register pair "R1,R0" is decremented by one. The carry and zero flags are affected. Carry is set if a borrow does not take place in DEC M operation; otherwise carry is cleared.

Operation M(R1,R0) ← M(R1,R0)–1

DEC R0 Decrement register R0

Machine code 0 0 0 0 1 1 1 1 1

Description Data in the working register R0 is decremented by one. Only the zero flag is affected.

Operation R0 ← R0–1

HALT Enter halt state

Machine code 0 0 0 1 1 0 1 0 1

Description HALT stops instruction execution and places the controller in power down mode. Reset or an active signal in the "PE,PF,PH" ports (by ROM code option) will resume execution. No flags are affected.

IN A,Pi Input port to accumulator

Machine code

PD	0 0 0 1 0 1 0 0 1
PE	0 0 0 0 0 1 0 0 1
PF	0 0 0 1 0 1 1 0 0
PH	0 0 0 1 0 0 1 0 1
PI	1 0 0 1 0 1 0 0 1

Description The data on port "Pi" is transferred to the accumulator. The zero flag is affected.

Operation ACC ← Pi; Pi=PD,PE,PF,PH,PI

INC M Increment data memory

Machine code 0 1 1 0 0 0 0 0 1

Description Data in the data memory specified by the register pair "R1,R0" is incremented by one. The carry and zero flags are affected. Carry is set if the operation results in a carry out; otherwise the carry is cleared.

Operation M(R1,R0) ← M(R1,R0)+1

INC R0 Increment register R0

Machine code 0 1 1 0 1 0 0 0 1

Description Data in the working register "R0" is incremented by one. The zero flag is affected.

Operation R0 ← R0+1

JB A.i,address Jump if bit of accumulator is set

Machine code 0 1 1 0 0 0 1 i1 i0 0 a a a a a a a a a

i0,i1 indicate which bit of accumulator will be detected. For example, i0=i1=0 means that if bit 0 of accumulator=1, the jump will be executed.

Description If the indicated bit of accumulator is set to 1, control passes to the specified address; otherwise proceed with the next instruction. Note that the branch destination is available only in the current page. (only bits 0~7 of the program counter will be replaced by the destination address.)

Operation PC ← address, if bit i of ACC=1
PC ← PC+2, if bit i of ACC=0

JB Pm.i,address Jump if bit of I/O port is set

Machine code PE 0 1 1 0 0 1 1 i1 i0 0 a a a a a a a a a

i0,i1 indicate which bit of port PE will be detected. For example, i0=i1=0 means that if bit 0 of port Pm=1, the jump will be executed.

Description If the indicated bit of port PE is set to 1, control passes to the specified address; otherwise proceed with the next instruction. Note that the branch destination is available only in the current page. (Only bits 0~7 of the program counter will be replaced by the destination address.)

Operation PC ← address, if bit i of Pm=1, Pm=PE
PC ← PC+2, if bit i of Pm=0, Pm=PE

JB M.i,address Jump if bit of data memory is set

Machine code 0 1 1 1 0 0 1 i1 i0 0 a a a a a a a a a

i0,i1 indicate which bit of data memory will be detected. For example, i0=i1=0 means that if bit 0 of data memory is equal to 1, the jump will be executed.

Description If the indicated bit of data memory addressed by register pair "R1,R0" is set to 1, control passes to the specified address; otherwise proceed with the next instruction. Note that the branch destination is available only in the current page. (Only bits 0~7 of program counter is replaced by the destination address.)

Operation PC ← address, if bit i of M(R1,R0)=1
PC ← PC+2, if bit i of M(R1,R0)=0

JC address Jump if carry flag is set

Machine code a 1 1 1 0 1 0 a a 0 a a a a a a a a a

Description If the carry flag is set to one, control passes to the specified address; otherwise proceed with the next instruction.

Operation PC ← address, if CF=1
PC ← PC+2, CF=0

JMP address Direct jump

Machine code a 1 1 1 1 1 1 a a 0 a a a a a a a a a

Description Bits 0~10 of the program counter are replaced with the directly specified address, and control passes to the destination.

Operation PC ← address

JNB A.i,address Jump if bit of accumulator is not set

Machine code 0 1 1 0 1 0 1 i1 i0 0 a a a a a a a a a

i0,i1 indicate which bit of accumulator will be detected. For example, i0=i1=0 means that if bit 0 of accumulator =0, the jump will be executed.

Description If the indicated bit of accumulator is reset to 0, control passes to the specified address; otherwise proceed with the next instruction. Note that the branch destination is available only in the current page. (Only bits 0~7 of program counter is replaced by the destination address.)

Operation PC ← address, if bit i of ACC=0
PC ← PC+2, if bit i of ACC=1

JNB M.i,address Jump if bit of data memory is not set

Machine code 0 1 1 1 1 0 1 i1 i0 0 a a a a a a a a a

i0,i1 indicate which bit of data memory will be detected. For example, i0=i1=0 means that if bit 0 of data memory =0, the jump will be executed.

Description If the indicated bit of data memory addressed by register pair "R1,R0" is reset to 0, control passes to the specified address; otherwise proceed with the next instruction. Note that the branch destination is available only in the current page. (Only bits 0~7 of program counter is replaced by the destination address.)

Operation PC ← address, if bit i of M(R1,R0)=0
PC ← PC+2, if bit i of M(R1,R0)=1

JNC address Jump if carry flag is not set

Machine code a 1 1 1 1 1 0 a a 0 a a a a a a a a a

Description If the carry flag is reset to zero, control passes to the specified address; otherwise proceed with the next instruction.

Operation PC ← address, if CF=0
PC ← PC+2, if CF=1

JNZ address Jump if zero flag is not set

Machine code a 1 1 0 1 1 0 a a 0 a a a a a a a a a

Description If the zero flag is reset to zero, control passes to the specified address; otherwise proceed with the next instruction.

Operation PC ← address, if ZF=0
PC ← PC+2, if ZF=1

JZ address Jump if zero flag is set

Machine code a 1 1 0 0 1 0 a a 0 a a a a a a a a a

Description If the zero flag is set to one, control passes to the specified address; otherwise proceed with the next instruction.

Operation PC ← address, if ZF=1
PC ← PC+2, if ZF=0

MOV A,M Move data memory to accumulator

Machine code 0 0 0 1 1 1 0 0 1

Description The content of the data memory addressed by the register pair "R1,R0" is moved to the accumulator. If the contents of data memory is zero, the zero flag will be set.

Operation ACC ← M(R1,R0)

MOV A,R0 Move register R0 content to accumulator

Machine code 0 0 0 0 0 0 1 0 1

Description The content of register R0 is moved into the accumulator. If the content of register R0 is zero, the zero flag will be set.

Operation ACC ← R0

MOV A,R1 Move register R1 content to accumulator

Machine code 0 0 0 0 0 0 1 1 0

Description The content of register R1 is moved into the bit 0 and 1 of the accumulator. The bit 2 and 3 of the accumulator are reset to 0. If the content of register R1 is zero, the zero flag will be set.

Operation ACC ← R1

MOV A,XH Move immediate data to accumulator

Machine code 0 0 1 1 0 d d d d

Description The 4-bit data specified by code is loaded in the accumulator. No flags are affected.

Operation ACC ← XH

MOV A,[XXH] Move data memory to accumulator directly

Machine code m5 1 0 0 m4 m3 m2 m1 m0

m5~m0: address of data memory

Description The content of the data memory directly addressed by code is moved to the accumulator. The zero flags is affected. If the content of the data memory is zero, the zero flag will be set.

Operation ACC ← M(m5~m0)

MOV M,A Move accumulator to data memory

Machine code 0 0 0 1 1 1 0 1 0

Description The content of the accumulator is moved to the data memory addressed by register pair "R1,R0".

Operation M(R1,R0) ← ACC

MOV R0,A Move accumulator to the register R0

Machine code 0 0 0 0 1 0 1 0 1

Description The content of accumulator is moved into the register "R0".

Operation R0 ← ACC

MOV R1,A Move accumulator to the register R1

Machine code 0 0 0 0 1 0 1 1 0

Description The bit 0 and 1 of the accumulator are moved into the register "R1".

Operation R1 ← ACC

MOV R0,M Move data memory to register R0

Machine code 0 0 0 1 0 0 1 1 0

Description The content of the data memory addressed by the register pair "R1,R0" is moved to the register R0.

Operation R0 ← M(R1,R0)

MOV R1R0,XXH Move immediate data to register R0 and R1

Machine code d 0 1 0 d d d d d

Description The 6-bit data specified by code is loaded in the register pair "R1,R0". No flags are affected.

Operation R1 ← XH (high nibble)
R0 ← XH (low nibble)

MOV [XXH],A Move accumulator to data memory directly

Machine code m5 1 0 1 m4 m3 m2 m1 m0

m5~m0:address of data memory

Description The content of accumulator is directly moved to the data memory addressed by code. No flags are affected.

Operation M(m5~m0) ← ACC

NOP No operation

Machine code 0 1 1 1 1 0 0 0 0

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

Operation PC ← PC+1

OR A,M Logical OR data memory to accumulator

Machine code 0 0 0 1 0 0 0 1 1

Description Data in the accumulator is logically ORed with the data memory addressed by register pair "R1,R0". The result is stored in the accumulator. The zero flag is affected.

Operation ACC ← ACC "OR" M(R1,R0)

OR A,XH Logical OR accumulator with immediate data

Machine code 0 0 0 0 0 0 0 1 1 0 0 0 0 1 d d d d

Description Data in the accumulator is logically ORed with the immediate data. The result is stored in the accumulator and the zero flag is affected.

Operation ACC ← ACC "OR" XH

OR M,A Logical OR accumulator to data memory

Machine code 0 1 1 0 1 0 0 0 0

Description Data in the accumulator is logically ORed with the data memory addressed by register pair "R1,R0". The result is stored in the data memory. The zero flag is affected.

Operation M(R1,R0) ← ACC "OR" M(R1,R0)

OUT PA,A Output accumulator data to port A

Machine code 0 0 0 1 1 1 1 0 0

Description The bit 0 on the accumulator is transferred to the output port PA0 for no carrier option. If carrier output option is selected, writing "0" to the PA0 will result in a carrier output, and writing "1" to the PA0 will keep the state of PA0 at normal low level.

Operation PA0 ← ACC0 (no carrier option).
ACC=0, PA0: carrier (carrier output option).
ACC=1, PA0: 0 (carrier output option).

OUT Po,A Output accumulator data to port

Machine code PD 0 0 0 1 0 1 0 1 0

PE 0 0 0 0 0 1 0 1 0

PH 0 0 0 1 1 0 1 1 0

PI 1 0 0 1 0 1 0 1 0

Description The data on the accumulator is transferred to the "Po" port.

Operation Po ← ACC; Po=PD,PE,PH,PI

READn MA Read ROM code to data memory and accumulator

Machine code n 1 1 1 0 1 1 n n

nnn: page number (0~7)

Description The 8 bits of ROM code addressed by ACC and M(R1,R0) in page n are moved to the data memory addressed by register pair "R1,R0" and the accumulator. The high nibble of the ROM code is loaded to the data memory and the low nibble of the ROM code is loaded to the accumulator. If the ACC is zero, the zero flag will be set. The address of the ROM code are specified by the following description:
ROM code address bit 10~8 ← Page "nnn"
ROM code address bit 7~4 ← ACC
ROM code address bit 3~0 ← M(R1,R0)

Operation M(R1,R0) ← ROM code (high nibble)
ACC ← ROM code (low nibble)

RLC A Rotate accumulator left through carry

Machine Code 0 1 1 1 1 0 0 1 1

Description The contents of the accumulator are rotated left one bit. Bit 3 replaces the carry bit; the carry bit is rotated into the bit 0 position.

Operation An+1 ← An; An: Accumulator bit n (n=0,1,2)
A0 ← CF
CF ← A3

RRC A Rotate accumulator right through carry

Machine Code 0 1 1 1 1 0 0 1 0

Description The contents of the accumulator are rotated right one bit. Bit 0 replaces the carry bit; the carry bit is rotated into the bit 3 position.

Operation An ← An+1; An: Accumulator bit n (n=0,1,2)
A3 ← CF
CF ← A0

SET C Set carry flag

Machine code 0 1 1 1 0 0 0 1 1

Description The carry flag is set to one.

Operation CF ← 1

SET M.i Set bit of data memory

Machine code 0 0 0 1 1 i3 i2 i1 i0

i0~i3 are determined by operand i. The corresponding bit will be "1" if the bit i of the memory is set to 1. For example, if i=0 then i3~i0=0001.

Description The bit of memory addressed by "R1,R0" is set to one.

Operation M(R1,R0).i ← 1

SET Pn.i Set bit of I/O port

Machine code
PD 0 0 0 1 0 i3 i2 i1 i0
PE 0 0 0 0 0 i3 i2 i1 i0
PI 1 0 0 1 0 i3 i2 i1 i0

i0~i3 are determined by operand i. The correponding bit will be "1" if the bit i of port Pn is set to 1. For example, if i=0 then i3~i0=0001

Description The specified bit i of port "Pn" is set to one. Pn=PD,PE,PI.

Operation Pn.i ← 1; Pn=PD,PE,PI

TRI Po,A Output accumulator to tri-state latch

Machine code
PD 0 0 0 0 0 1 1 0 0
PE 0 0 0 0 1 1 1 0 0
PH 0 0 0 1 0 1 1 1 1
PI 1 0 0 0 0 1 1 0 0

Description Data in the accumulator is transferred to the tri-state latch of port "Po". The "1" written to the tri-state latch makes the corresponding output part become floating. Writing "0" to the tri-state latch will force the related I/O bit to operate in output mode.

Operation Po ← ACC ; Po=PD,PE,PH,PI

XOR A,M Logical Exclusive–OR data memory to accumulator

Machine code 0 0 0 1 1 0 0 1 1

Description Data in the accumulator is Exclusive–ORed with the data memory addressed by register pair "R1,R0". The result is stored in the accumulator. The zero flag is affected.

Operation ACC ← ACC "XOR" M(R1,R0)

XOR A,XH Logical Exclusive–OR accumulator with immediate data

Machine code 0 0 0 0 0 0 0 1 1 0 0 0 1 1 d d d d

Description Data in the accumulator is Exclusive-ORed with the immediate data specified by code. The result is stored in the accumulator. The zero flag is affected.

Operation ACC ← ACC "XOR" XH

XOR M,A Logical Exclusive-OR accumulator to data memory

Machine code 0 1 1 1 0 0 0 0 0

Description Data in the accumulator is Exclusive-ORed with the data memory addressed by register pair "R1,R0". The result is stored in the data memory. The zero flag is affected.

Operation M(R1,R0) ← ACC "XOR" M(R1,R0)