



# HT86030/HT86070 Voice Synthesizer 8-Bit MCU

## Technical Document

- [Tools Information](#)
- [FAQs](#)

## Features

- Operating voltage: 2.4V~5.2V
- System clock: 4MHz~8MHz
- Crystal or RC oscillator for system clock
- 16 I/O pins
- 8K×16-bit program ROM
- 208×8-bit RAM
- One external interrupt input
- Two 16-bit programmable timer counter and overflow interrupts
- 12-bit high quality D/A output by transistor or HT82V733
- Built-in voice ROM in various capacity
- One 8-bit counter with 3-bit prescaler
- Watchdog Timer
- 8-level subroutine nesting
- HALT function and wake-up feature reduce power consumption
- Up to 1μs (0.5μs) instruction cycle with 4MHz (8MHz) system clock
- Support 16-bit table read instruction (TBLP, TBHP)
- 63 powerful and efficient instructions
- 28-pin SOP package

## Applications

- Intelligent educational leisure products
- Alert and warning systems
- High end leisure product controllers
- Sound effect generators

## General Description

The HT86030/HT86070 series are 8-bit high performance microcontroller with voice synthesizer and tone generator. The HT86030/HT86070 is designed for applications on multiple I/Os with sound effects, such as voice and melody. It can provide various sampling rates and beats, tone levels, tempos for speech synthesizer and melody generator. It has a single built-in high quality, D/A output. There is an external interrupt which can

be triggered with falling edge pulse or falling/rising edge pulse.

The HT86030/HT86070 is excellent for versatile voice and sound effect product applications. The efficient MCU instructions allow users to program the powerful custom applications. The system frequency of HT86030/HT86070 can be up to 8MHz under 2.4V and include a HALT function to reduce power consumption.

## Selection Table

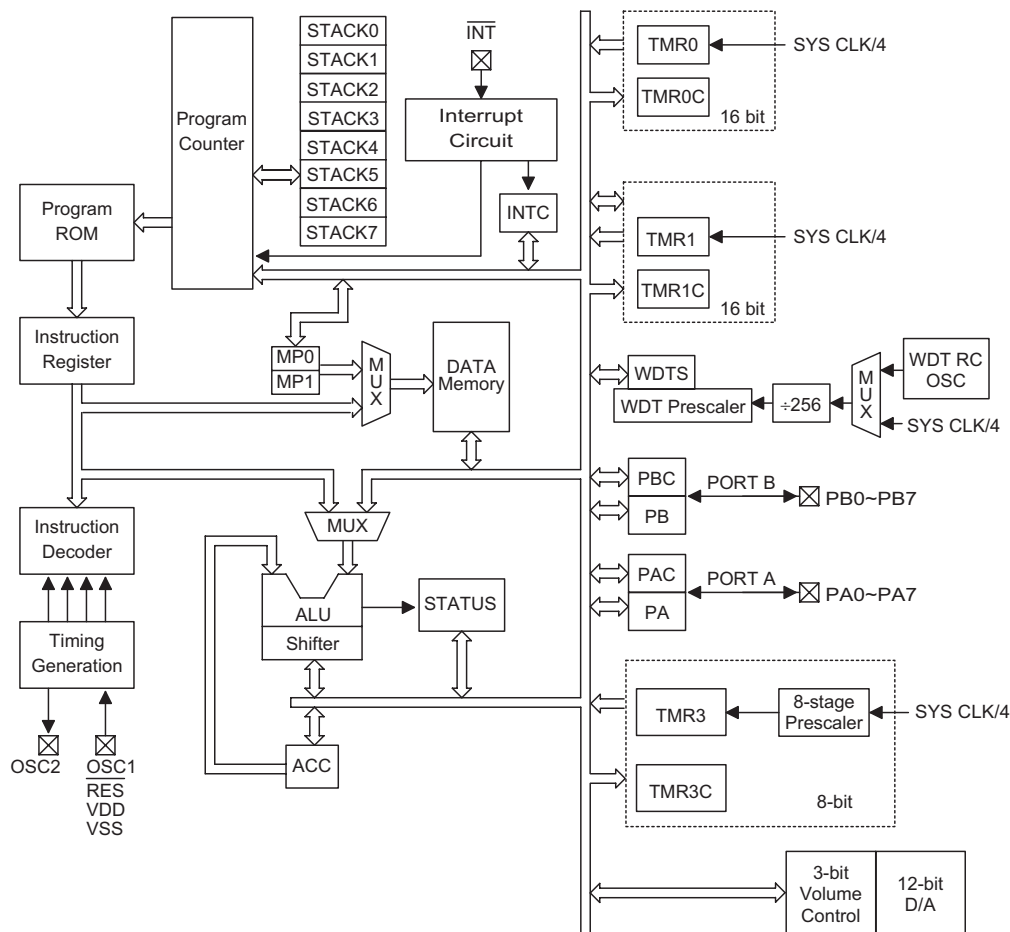
| Body           | HT86030  | HT86070   |
|----------------|----------|-----------|
| Voice ROM size | 768K-bit | 1536K-bit |
| Voice length   | 36 sec   | 72 sec    |

Note: \* Voice length is estimated by 21K-bit data rate





## Block Diagram



## Pin Assignment

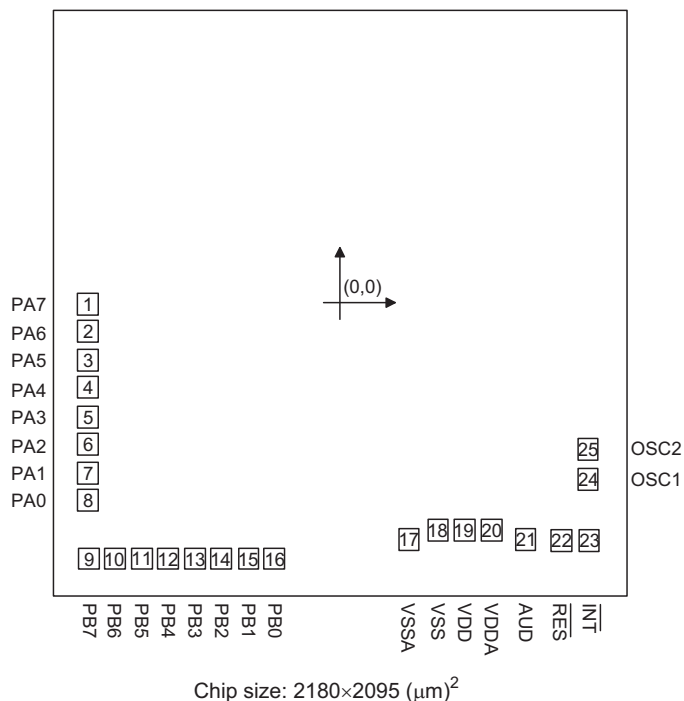
|     |    |    |      |
|-----|----|----|------|
| NC  | 1  | 28 | NC   |
| NC  | 2  | 27 | NC   |
| NC  | 3  | 26 | NC   |
| NC  | 4  | 25 | NC   |
| PA7 | 5  | 24 | NC   |
| PA6 | 6  | 23 | OSC2 |
| PA5 | 7  | 22 | OSC1 |
| PA4 | 8  | 21 | INT  |
| PA3 | 9  | 20 | RES  |
| PA2 | 10 | 19 | AUD  |
| PA1 | 11 | 18 | TEST |
| PA0 | 12 | 17 | VDDA |
| NC  | 13 | 16 | VDD  |
| VSS | 14 | 15 | VSSA |

**HT86030/HT86070**  
- 28 SOP-A



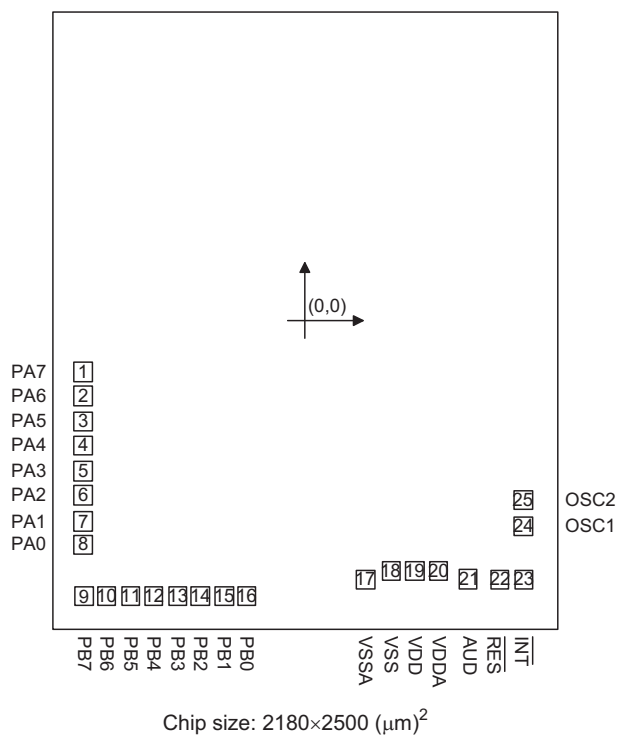
## Pad Assignment

### HT86030



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

### HT86070



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



**Pad Coordinates**
**HT86030**

| Pad No. | X        | Y        | Pad No. | X        | Y        |
|---------|----------|----------|---------|----------|----------|
| 1       | -940.200 | -3.100   | 14      | -444.300 | -897.100 |
| 2       | -940.200 | -98.200  | 15      | -341.300 | -897.100 |
| 3       | -940.200 | -201.200 | 16      | -246.200 | -897.100 |
| 4       | -940.200 | -296.300 | 17      | 255.700  | -830.550 |
| 5       | -940.200 | -399.300 | 18      | 365.000  | -796.050 |
| 6       | -940.200 | -494.400 | 19      | 465.000  | -796.050 |
| 7       | -940.200 | -597.400 | 20      | 565.100  | -796.050 |
| 8       | -940.200 | -692.500 | 21      | 690.250  | -828.350 |
| 9       | -935.600 | -897.100 | 22      | 824.350  | -832.150 |
| 10      | -840.500 | -897.100 | 23      | 927.350  | -832.150 |
| 11      | -737.500 | -897.100 | 24      | 924.100  | -618.924 |
| 12      | -642.400 | -897.100 | 25      | 924.100  | -514.624 |
| 13      | -539.400 | -897.100 |         |          |          |

**HT86070**

| Pad No. | X        | Y         | Pad No. | X        | Y         |
|---------|----------|-----------|---------|----------|-----------|
| 1       | -940.200 | -205.600  | 14      | -444.300 | -1099.600 |
| 2       | -940.200 | -300.700  | 15      | -341.300 | -1099.600 |
| 3       | -940.200 | -403.700  | 16      | -246.200 | -1099.600 |
| 4       | -940.200 | -498.800  | 17      | 255.700  | -1033.050 |
| 5       | -940.200 | -601.800  | 18      | 365.000  | -998.550  |
| 6       | -940.200 | -696.900  | 19      | 465.000  | -998.550  |
| 7       | -940.200 | -799.900  | 20      | 565.100  | -998.550  |
| 8       | -940.200 | -895.000  | 21      | 690.250  | -1030.850 |
| 9       | -935.600 | -1099.600 | 22      | 824.350  | -1034.650 |
| 10      | -840.500 | -1099.600 | 23      | 927.350  | -1034.650 |
| 11      | -737.500 | -1099.600 | 24      | 924.100  | -821.424  |
| 12      | -642.400 | -1099.600 | 25      | 924.100  | -717.124  |
| 13      | -539.400 | -1099.600 |         |          |           |



**Pad Description**

| Pad Name                | I/O | Mask Option   | Description  |
|-------------------------|-----|---|--|
| PA0~PA7                 | I/O | Wake-up,<br>Pull-high<br>or None                          | Bidirectional 8-bit I/O port. Each bit can be configured as a wake-up input by mask option. Software instructions determine the CMOS output or Schmitt trigger input with or without pull-high resistor (mask option).   |
| PB0~PB7                 | I/O | Pull-high<br>or None                                      | Bidirectional 8-bit I/O port. Software instructions determine the CMOS output or Schmitt trigger input (pull-high resistor depending on mask option).  |
| VSS                     | —   | —   | Negative power supply, ground  |
| VDD                     | —   | —   | Positive power supply  |
| VDDA                    | —   | —   | DAC power supply   |
| VSSA                    | —   | —   | DAC negative power supply, ground  |
| $\overline{\text{RES}}$ | I   | —   | Schmitt trigger reset input, active low  |
| $\overline{\text{INT}}$ | I   | Falling Edge Trigger<br>or Falling/Rising Edge<br>Trigger | External interrupt Schmitt trigger input without pull-high resistor. Choice falling edge trigger or falling/rising edge trigger by mask option. Falling edge triggered active on a high to low transition. Rising edge triggered active on a low to high transition.     |
| OSC1<br>OSC2            | —   | RC or Crystal   | OSC1 and OSC2 are connected to an RC network or a crystal (by mask option) for the internal system clock. In the case of RC operation, OSC2 is the output terminal for 1/4 system clock.<br>The system clock may come from the crystal, the two pins cannot be floating. |
| AUD                     | O   | —   | Audio output for driving a external transistor or for driving HT82V733   |

**Absolute Maximum Ratings**

|                      |                                |                             |                                  |
|----------------------|--------------------------------|-----------------------------|----------------------------------|
| Supply Voltage ..... | $V_{SS}-0.3V$ to $V_{SS}+5.5V$ | Storage Temperature .....   | $-50^{\circ}C$ to $125^{\circ}C$ |
| Input Voltage .....  | $V_{SS}-0.3V$ to $V_{DD}+0.3V$ | Operating Temperature ..... | $-20^{\circ}C$ to $70^{\circ}C$  |

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**

Ta=25°C

| Symbol            | Parameter                                      | Test Conditions |                                     | Min. | Typ. | Max. | Unit |
|-------------------|--|-----------------|-------------------------------------|------|------|------|------|
|                   |  | V <sub>DD</sub> | Conditions                          |      |      |      |      |
| V <sub>DD</sub>   | Operating Voltage                              | —               | f <sub>SYS</sub> =4MHz/8MHz         | 2.4  | —    | 5.2  | V    |
| I <sub>STB1</sub> | Standby Current (Watchdog Off)                 | 3V              | No load, system HALT                | —    | —    | 1    | μA   |
|                   |  | 5V              |                                     | —    | —    | 2    |      |
| I <sub>STB2</sub> | Standby Current(Watchdog On)                   | 3V              | No load, system HALT                | —    | —    | 7    | μA   |
|                   |  | 5V              |                                     | —    | —    | 10   |      |
| I <sub>DD</sub>   | Operating Current (RC OSC)                     | 3V              | No load, f <sub>SYS</sub> =8MHz     | —    | —    | 3.5  | mA   |
|                   |  | 5V              |                                     | —    | —    | 8    |      |
| I <sub>OL</sub>   | I/O Port Sink Current                          | 3V              | V <sub>OL</sub> =0.1V <sub>DD</sub> | —    | 11   | —    | mA   |
|                   |  | 5V              |                                     | —    | 25   | —    |      |
| I <sub>OH</sub>   | I/O Port Source Current                        | 3V              | V <sub>OH</sub> =0.9V <sub>DD</sub> | —    | -6   | —    | mA   |
|                   |  | 5V              |                                     | —    | -15  | —    |      |
| I <sub>O</sub>    | AUD Source Current                             | 3V              | V <sub>OH</sub> =0.9V <sub>DD</sub> | —    | -3   | —    | mA   |
|                   |  | 5V              |                                     | —    | -7   | —    |      |
| V <sub>IL1</sub>  | Input Low Voltage for I/O Ports                | 3V              | —                                   | —    | 1.3  | —    | V    |
|                   |  | 5V              |                                     | —    | 2.3  | —    |      |
| V <sub>IH1</sub>  | Input High Voltage for I/O Ports               | 3V              | —                                   | —    | 1.8  | —    | V    |
|                   |  | 5V              |                                     | —    | 2.9  | —    |      |
| V <sub>IL2</sub>  | Reset Low Voltage ( $\overline{\text{RES}}$ )  | 3V              | —                                   | —    | 1.4  | —    | V    |
|                   |  | 5V              |                                     | —    | 3    | —    |      |
| V <sub>IH2</sub>  | Reset High Voltage ( $\overline{\text{RES}}$ ) | 3V              | —                                   | —    | 2.4  | —    | V    |
|                   |  | 5V              |                                     | —    | 3.9  | —    |      |
| R <sub>PH</sub>   | Pull-high Resistance                           | 3V              | —                                   | 20   | 60   | 100  | kΩ   |
|                   |  | 5V              |                                     | 10   | 30   | 50   |      |

**A.C. Characteristics**

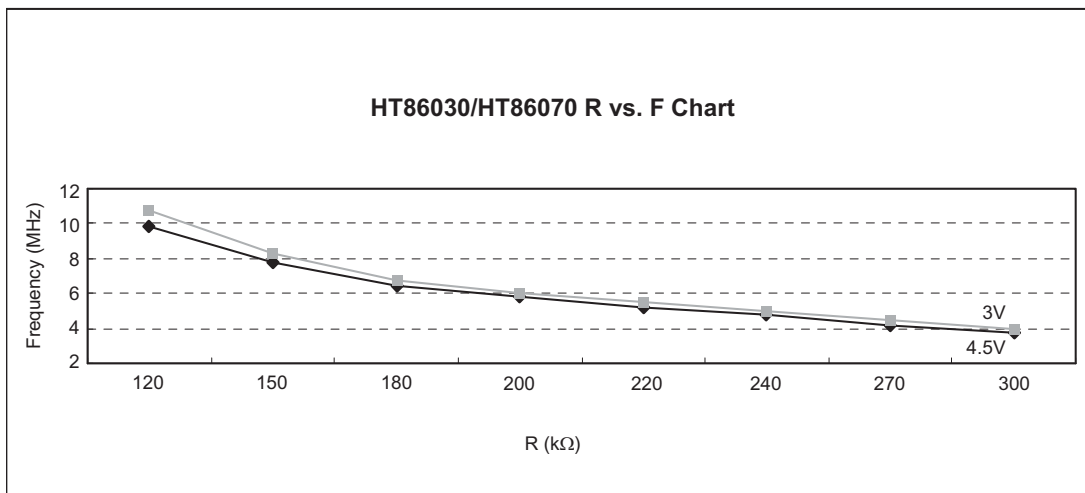
Ta=25°C

| Symbol              | Parameter                               | Test Conditions |                       | Min. | Typ. | Max. | Unit             |
|---------------------|---|-----------------|-----------------------|------|------|------|------------------|
|                     |   | V <sub>DD</sub> | Conditions            |      |      |      |                  |
| f <sub>SYS1</sub>   | System Clock (RC OSC)                   | —               | 2.4V~5.2V             | 4    | —    | 8    | MHz              |
| f <sub>SYS2</sub>   | System Clock (Crystal OSC)              | —               | 2.4V~5.2V             | 4    | —    | 8    | MHz              |
| f <sub>TIMER</sub>  | Timer Input Frequency                   | —               | 2.4V~5.2V             | 0    | —    | 8    | MHz              |
| t <sub>WDTOSC</sub> | Watchdog Oscillator Period              | 3V              | —                     | 45   | 90   | 180  | μs               |
|                     |   | 5V              | —                     | 32   | 65   | 130  | μs               |
| t <sub>WDT1</sub>   | Watchdog Time-out Period (WDT OSC)      | 3V              | Without WDT prescaler | 11   | 23   | 46   | ms               |
|                     |   | 5V              |                       | 8    | 17   | 33   | ms               |
| t <sub>WDT2</sub>   | Watchdog Time-out Period (System Clock) | —               | Without WDT prescaler | —    | 1024 | —    | t <sub>SYS</sub> |
| t <sub>RES</sub>    | External Reset Low Pulse Width          | —               | —                     | 1    | —    | —    | μs               |
| t <sub>SST</sub>    | System Start-up Timer Period            | —               | Wake-up from HALT     | —    | 1024 | —    | t <sub>SYS</sub> |
| t <sub>INT</sub>    | Interrupt Pulse Width                   | —               | —                     | 1    | —    | —    | μs               |

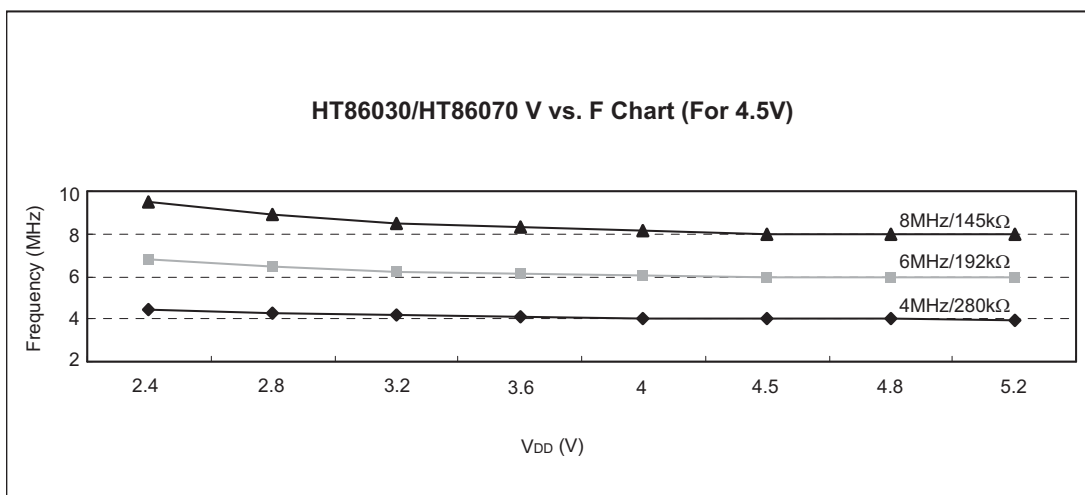
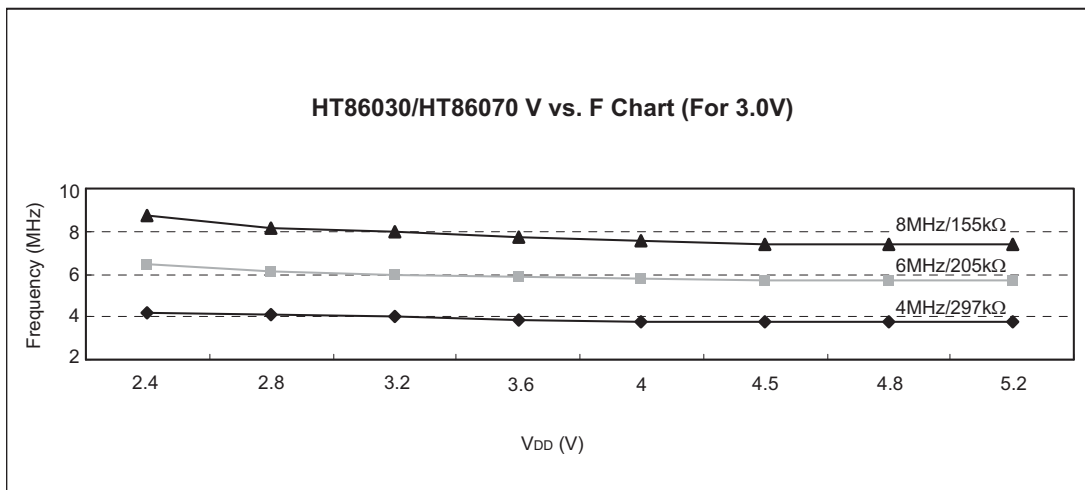


## Characteristics Curves

### HT86030/HT86070 R vs. F Characteristics Curve



### HT86030/HT86070 V vs. F Characteristics Curve





## Functional Description

### Execution Flow

The system clock for the HT86030/HT86070 series is derived from either a crystal or an RC oscillator. It is internally divided into four non-overlapping clocks. One instruction cycle consists of four system clock cycles.

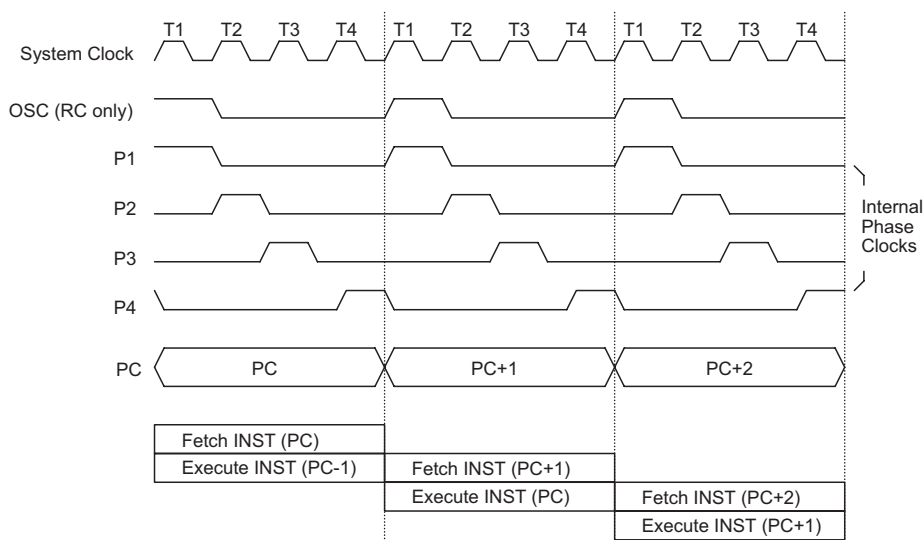
Instruction fetching and execution are pipelined in such a way that a fetch takes one instruction cycle while decoding and execution takes the next instruction cycle. However, the pipelining scheme causes each instruction to effectively execute within one cycle. If an instruction changes the program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

### Program Counter – PC

The 13-bit program counter (PC) controls the sequence in which the instructions stored in program ROM are executed.

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, internal interrupt, external interrupt or return from subroutine, the PC manipulates the program transfer by loading the address corresponding to each instruction.



**Execution Flow**

| Mode                               | Program Counter   |     |     |    |    |    |    |    |    |    |    |    |    |
|------------------------------------|-------------------|-----|-----|----|----|----|----|----|----|----|----|----|----|
|                                    | *12               | *11 | *10 | *9 | *8 | *7 | *6 | *5 | *4 | *3 | *2 | *1 | *0 |
| Initial Reset                      | 0                 | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| External or Serial Input Interrupt | 0                 | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  |
| Timer/Event Counter 0 Overflow     | 0                 | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  |
| Timer/Event Counter 1 Overflow     | 0                 | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 0  | 0  |
| Timer Counter 3 Overflow           | 0                 | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 1  | 0  | 0  |
| Skip                               | Program Counter+2 |     |     |    |    |    |    |    |    |    |    |    |    |
| Loading PCL                        | *12               | *11 | *10 | *9 | *8 | @7 | @6 | @5 | @4 | @3 | @2 | @1 | @0 |
| Jump, Call Branch                  | #12               | #11 | #10 | #9 | #8 | #7 | #6 | #5 | #4 | #3 | #2 | #1 | #0 |

**Program Counter**

Note: \*12~\*0: Program counter bits

#12~#0: Instruction code bits

S12~S0: Stack register bits

@7~@0: PCL bits



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 takes its place while the correct instruction is obtained.

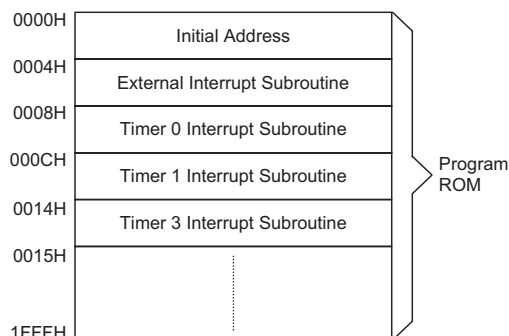
The lower byte of the program counter (PCL) is a read/write register (06H). Moving data into the PCL performs a short jump. The destination must be within 256 locations.

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

### Program Memory – ROM

The program memory stores the program instructions that are to be executed. It also includes data, table and interrupt entries, addressed by the program counter along with the table pointer. The program memory size for HT86030/HT86070 is 8192×16 bits. Certain locations in the program memory are reserved for special usage:

- Location 000H  
This area is reserved for program initialization. The program always begins execution at location 000H each time the system is reset.
- Location 004H  
This area is reserved for the external interrupt service program. If the  $\overline{\text{INT}}$  input pin is activated, and the interrupt is enabled and the stack is not full, the program will jump to location 004H and begins execution.



**Program Memory**

- Location 008H

This area is reserved for the 16-bit Timer/Event Counter 0 interrupt service program. If a timer interrupt results from a Timer/Event Counter 0 overflow, and if the interrupt is enabled and the stack is not full, the program will jump to location 008H and begins execution.

- Location 00CH

This area is reserved for the 16-bit Timer/Event Counter 1 interrupt service program. If a timer interrupt results from a Timer/Event Counter 1 overflow, and if the interrupt is enabled and the stack is not full, the program will jump to location 00CH and begins execution.

- Location 014H

This area is reserved for the 8-bit Timer Counter 3 interrupt service program. If a timer interrupt results from a Timer Counter 3 overflow, and if the interrupt is enabled and the stack is not full, the program will jump to location 014H and begins execution.

### Table location

Any location in the ROM space can be used as look up tables. The instructions TABRDC [m] (used for any bank) and TABRDL [m] (only used for last page of program ROM) transfer the contents of the lower-order byte to the specified data memory [m], and the higher-order byte to TBLH (08H). Only the destination of the lower-order byte in the table is well-defined. The higher-order bytes of the table word are transferred to the TBLH. The table higher-order byte register (TBLH) is read only.

The table pointer (TBHP, TBLP) is a read/write register, which indicates the table location. Because TBHP is unknown after power on reset, TBHP must be set specified.

### Stack Register – Stack

The stack register is a special part of the memory used to save the contents of the program counter. This stack is organized into eight levels. It is neither part of the data nor part of the program space, and cannot be read or written to. Its activated level is indexed by a stack pointer (SP) and cannot be read or written to. At a subroutine call or interrupt acknowledgment, the contents of the program counter are pushed onto the stack.

| Instruction | Table Location |     |     |    |    |    |    |    |    |    |    |    |    |
|-------------|----------------|-----|-----|----|----|----|----|----|----|----|----|----|----|
|             | *12            | *11 | *10 | *9 | *8 | *7 | *6 | *5 | *4 | *3 | *2 | *1 | *0 |
| TABRDC [m]  | P12            | P11 | P10 | P9 | P8 | @7 | @6 | @5 | @4 | @3 | @2 | @1 | @0 |
| TABRDL [m]  | 1              | 1   | 1   | 1  | 1  | @7 | @6 | @5 | @4 | @3 | @2 | @1 | @0 |

**Table Location**

Note: \*12~\*0: Current program ROM table

@7~@0: Write @7~@0 to TBLP pointer register

P12~P8: Write P12~P8 to TBHP pointer register



The program counter is restored to its previous value from the stack at the end of subroutine or interrupt routine, which is signaled by return instruction (RET or RETI). After a chip resets, SP will point to the top of the stack.

The interrupt request flag will be recorded but the acknowledgment will be inhibited when the stack is full and a non-masked interrupt takes place. After the stack pointer is decremented (by RET or RETI), the interrupt request will be serviced. This feature prevents stack overflow and allows programmers to use the structure more easily. In a similar case, if the stack is full and a "CALL" is subsequently executed, stack overflow occurs and the first entry is lost.

#### Data Memory – RAM

The data memory is designed with 208×8 bits. The data memory is further divided into two functional groups, namely, special function registers (00H~2AH) and general purpose user data memory (30H~FFH). Although most of them can be read or be written to, some are read only.

The special function registers include an indirect addressing register (R0:00H), memory pointer register (MP0:01H), accumulator (ACC:05H), program counter lower-order byte register (PCL:06H), table pointer (TBLP:07H), table higher-order byte register (TBLH:08H), status register (STATUS:0AH), interrupt

control register 0 (INTC:0BH), Timer/Event Counter 0 (TMR0H:0CH,TMR0L:0DH), Timer/Event Counter 0 control register (TMR0C:0EH), Timer/Event Counter 1 (TMR1H:0FH, TMR1L:10H), Timer/Event Counter 1 control register (TMR1C:11H), I/O registers (PA:12H,PB:14H), I/O control registers (PAC:13H,PBC:15H), voice ROM address latch0[23:0] (LATCH0H:18H, LATCH0M:19H, LATCH0L:1AH), voice ROM address latch1[23:0] (LATCH1H:1BH, LATCH1M:1CH, LATCH1L:1DH), interrupt control register 1 (INTCH:1EH), table pointer higher-order byte register (TBHP:1FH), Timer Counter 3 (TMR3L:24H), Timer Counter 3 control register (TMR3C:25H), voice control register (VOICEC:26H), DAC output (DAH:27H,DAL:28H), volume control register (VOL:29H), voice ROM latch data register (LATCHD:2AH).

The general purpose data memory, addressed from 30H~FFH, is used for data and control information under instruction commands.

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

| Address | RAM Mapping | Read/Write | Description                                |
|---------|-------------|------------|--|
| 00H     | R0          | R/W        | Indirect addressing register 0             |
| 01H     | MP0         | R/W        | Memory pointer 0                           |
| 02H     | R1          | R/W        | Indirect addressing register 1             |
| 03H     | MP1         | R/W        | Memory pointer 1                           |
| 04H     | Unused      |            |  |
| 05H     | ACC         | R/W        | Accumulator                                |
| 06H     | PCL         | R/W        | Program counter lower-order byte address   |
| 07H     | TBLP        | R/W        | Table pointer lower-order byte address     |
| 08H     | TBLH        | R          | Table higher-order byte content register   |
| 09H     | WDTS        | R/W        | Watchdog Timer option setting register     |
| 0AH     | STATUS      | R/W        | Status register                            |
| 0BH     | INTC        | R/W        | Interrupt control register 0               |
| 0CH     | TMR0H       | R/W        | Timer/Event counter 0 higher-byte register |
| 0DH     | TMR0L       | R/W        | Timer/Event counter 0 lower-byte register  |
| 0EH     | TMR0C       | R/W        | Timer/Event counter 0 control register     |
| 0FH     | TMR1H       | R/W        | Timer/Event counter 1 higher-byte register |
| 10H     | TMR1L       | R/W        | Timer/Event counter 1 lower-byte register  |
| 11H     | TMR1C       | R/W        | Timer/Event counter 1 control register     |
| 12H     | PA          | R/W        | Port A I/O data register                   |



| Address | RAM Mapping   | Read/Write                        | Description   |
|---------|---------------|-----------------------------------|---|
| 13H     | PAC           | R/W                               | Port A I/O control register                                 |
| 14H     | PB            | R/W                               | Port B I/O data register                                    |
| 15H     | PBC           | R/W                               | Port B I/O control register                                 |
| 18H     | LATCH0H       | R/W                               | Voice ROM address latch 0 [A23~A16]                         |
| 19H     | LATCH0M       | R/W                               | Voice ROM address latch 0 [A15~A8]                          |
| 1AH     | LATCH0L       | R/W                               | Voice ROM address latch 0 [A7~A0]                           |
| 1BH     | LATCH1H       | R/W                               | Voice ROM address latch 1 [A23~A16]                         |
| 1CH     | LATCH1M       | R/W                               | Voice ROM address latch 1 [A15~A8]                          |
| 1DH     | LATCH1L       | R/W                               | Voice ROM address latch 1 [A7~A0]                           |
| 1EH     | INTCH         | R/W                               | Interrupt control register 1                                |
| 1FH     | TBHP          | R/W                               | Table pointer higher-order byte register                    |
| 23H     | Unused        |                                   |   |
| 24H     | TMR3L         | R/W                               | Timer Counter 3 lower-byte register                         |
| 25H     | TMR3C         | R/W                               | Timer Counter 3 control register                            |
| 26H     | VOICEEC       | R/W                               | Voice control register                                      |
| 27H     | DAL           | R/W, higher-nibble available only | DAC output data D3~D0 to DAL7~DAL4                          |
| 28H     | DAH           | R/W                               | DAC output data D11~D4 to DAH7~DAH0                         |
| 29H     | VOL           | R/W, higher-nibble available only | Volume control register, and volume controlled by VOL7~VOL5 |
| 2AH     | LATCHD        | R                                 | Voice ROM data register                                     |
| 2BH~2FH | Unused        |                                   |   |
| 30H~FFH | User data RAM | R/W                               | User data RAM   |

### Indirect Addressing Register

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

The function of data movement between two indirect addressing registers is not supported. The memory pointer registers, MP0 and MP1, are both 8-bit registers used to access the RAM by combining the corresponding indirect addressing registers.

### Accumulator – ACC (05H)

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

### Arithmetic and Logic Unit – ALU

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

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

### Status Register – STATUS

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

Except the TO and PDF flags, bits in the status register can be altered by instructions similar to other registers. Data written into the status register does not alter the TO or PDF flags. Operations related to the status register,



however, may yield different results from those intended. The TO and PDF flags can only be changed by a Watchdog Timer overflow, chip power-up, or clearing the Watchdog Timer and executing the "HALT" instruction. The Z, OV, AC, and C flags reflect the status of the latest operations.

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

### Interrupts

The HT86030/HT86070 provides an external interrupt, three 16-bit programmable timer interrupts, and an 8-bit programmable timer interrupt. The Interrupt Control registers (INTC:0BH, INTCH:1EH) contain the interrupt control bits to set to enable/disable and the interrupt request flags.

Once an interrupt subroutine is serviced, all other interrupts will be blocked (by clearing the EMI bit). This scheme may prevent any further interrupt nesting. Other interrupt requests may happen during this interval but only the interrupt request flag is recorded. If a certain interrupt needs servicing within the service routine, the EMI bit and the corresponding INTC/INTCH bit may be set to allow interrupt nesting. If the stack is full, the interrupt request will not be acknowledged, even if the related interrupt is enabled, until the SP is decremented. If immediate service is desired, the stack must be prevented from becoming full.

As an interrupt is serviced, a control transfer occurs by pushing the program counter onto the stack and then branching to subroutines at the specified location(s) in the program memory. Only the program counter is

pushed onto the stack. The programmer must save the contents of the register or status register (STATUS) in advance if they are altered by an interrupt service program which corrupts the desired control sequence.

External interrupt is triggered by a high-to-low/low-to-high transition of  $\overline{\text{INT}}$  pin which sets the related interrupt request flag (EIF:bit 4 of INTC). When the interrupt is enabled, and the stack is not full and the external interrupt is active, a subroutine call to location 04H will occur. The interrupt request flag (EIF) and EMI bits will be cleared to disable other interrupts.

The internal Timer/Event Counter 0 interrupt is initialized by setting the Timer/Event counter 0 interrupt request flag (T0F:bit 5 of INTC), caused by a Timer/Event Counter 0 overflow. When the interrupt is enabled, and the stack is not full and the T0F bit is set, a subroutine call to location 08H will occur. The related interrupt request flag (T0F) will be reset and the EMI bit cleared to disable further interrupts.

The internal Timer/Event Counter 1 interrupt is initialized by setting the Timer/Event Counter 1 interrupt request flag (T1F:bit 6 of INTC), caused by a Timer/Event Counter 1 overflow. When the interrupt is enabled, and the stack is not full and the T1F bit is set, a subroutine call to location 0CH will occur. The related interrupt request flag (T1F) will be reset and the EMI bit cleared to disable further interrupts.

The internal Timer Counter 3 interrupt is initialized by setting the Timer Counter 3 interrupt request flag (T3F:bit 1 of INTCH), caused by a Timer Counter 3 overflow. When the interrupt is enabled, and the stack is not full and the T3F bit is set, a subroutine call to location 14H will occur. The related interrupt request flag (T3F) will be reset and the EMI bit cleared to disable further interrupts.

| Bit No. | Label | Function  |
|---------|-------|---|
| 0       | C     | C is set if an 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 an 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 logical operation is zero; otherwise Z is cleared.   |
| 3       | OV    | OV is set if an operation results in a carry into the highest-order bit but not a carry out of the highest-order bit, or vice versa; otherwise OV is cleared.   |
| 4       | PDF   | PDF is cleared by system power-up or executing the "CLR WDT" instruction.<br>PDF is set by executing the "HALT" instruction.  |
| 5       | TO    | TO is cleared by system power-up or executing the "CLR WDT" or "HALT" instruction.<br>TO is set by a WDT time-out.  |
| 6, 7    | —     | Unused bit, read as "0"   |

**Status (0AH) Register**



During the execution of an interrupt subroutine, other interrupt acknowledges are held until the RETI instruction is executed or the EMI bit and the related interrupt control bit are set to 1 (of course, if the stack is not full). To return from the interrupt subroutine, the RET or RETI instruction may be invoked. RETI will set the EMI bit to enable an interrupt service, but RET will not.

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

The Timer/Event Counter 0/1 interrupt request flag (T0F/T1F) which enables Timer/Event Counter 0/1 control bit (ET0I/ET1I), the Timer Counter 3 interrupt request flag (T3F) which enables Timer Counter 3 control bit (ET3I), and external interrupt request flag (EIF) which enables external interrupt control bit (EEI) form the interrupt control register (INTC:0BH and INTCH:1EH). EMI, EEI, ET0I, ET1I and ET3I are used to control the enabling/disabling of interrupts. These bits prevent the requested interrupt begin serviced. Once the interrupt request flags (T0F, T1F, T3F, EIF) are set, they will remain in the INTC/INTCH register until the interrupts are serviced or cleared by a software instruction.

It is recommended that application programs do not use CALL subroutines within an interrupt subroutine. Interrupts often occur in an unpredictable manner or need to be serviced immediately in some applications. If only one stack is left and the interrupt enable is not well controlled, once a CALL subroutine is used in the interrupt subroutine will corrupt the original control sequence.

| Bit No. | Label | Function   |
|---------|-------|--|
| 0       | EMI   | Controls the master (global) interrupt (1= enabled; 0= disabled) |
| 1       | EEI   | Controls the external interrupt (1= enabled; 0= disabled)        |
| 2       | ET0I  | Controls the timer 0 interrupt (1= enabled; 0= disabled)         |
| 3       | ET1I  | Controls the timer 1 interrupt (1= enabled; 0= disabled)         |
| 4       | EIF   | External interrupt request flag (1= active; 0= inactive)         |
| 5       | T0F   | Timer 0 request flag (1= active; 0= inactive)                    |
| 6       | T1F   | Timer 1 request flag (1= active; 0= inactive)                    |
| 7       | —     | Unused bit, read as "0"  |

**INTC (0BH) Register**

| Bit No.     | Label | Function   |
|-------------|-------|--|
| 0, 2~4, 6~7 | —     | Unused bit, read as "0"                                  |
| 1           | ET3I  | Controls the timer 3 interrupt (1= enabled; 0= disabled) |
| 5           | T3F   | Timer 3 interrupt request flag (1= active; 0= inactive)  |

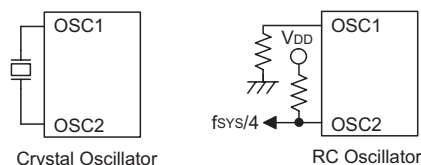
**INTCH (1EH) 1 Register**

| Interrupt Source               | Priority | Vector |
|--------------------------------|----------|--------|
| External Interrupt             | 1        | 04H    |
| Timer/Event Counter 0 Overflow | 2        | 08H    |
| Timer/Event Counter 1 Overflow | 3        | 0CH    |
| Timer Counter 3 Overflow       | 4        | 14H    |

### Oscillator Configuration

The HT86030/HT86070 provides two types of oscillator circuit for the system clock, i.e., RC oscillator and crystal oscillator. No matter what type of oscillator, the signal is used for the system clock. The HALT mode stops the system oscillator and ignores external signal to conserve power. If the RC oscillator is used, an external resistor between OSC1 and VDD is required, and the range of the resistance should be from 30kΩ to 680kΩ. The system clock, divided by 4, is available on OSC2 with pull-high resistor, which can be used to synchronize external logic. The RC oscillator provides the most cost effective solution. However, the frequency of the oscillation may vary with VDD, temperature, and the chip itself due to process variations. It is therefore not suitable for timing sensitive operations where accurate oscillator frequency is desired.

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



**System Oscillator**



### Watchdog Timer – WDT

The WDT clock source is implemented by a dedicated RC oscillator (WDT oscillator) or instruction clock (system clock divided by 4), decided by mask options. 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 mask option. If the Watchdog Timer is disabled, all the executions related to the WDT result in no operation.

Once the internal WDT oscillator (RC oscillator with period 78 $\mu$ s normally) is selected, it is first divided by 256 (8-stages) to get the nominal time-out period of approximately 20 ms. This time-out period may vary with temperature, VDD and process variations. By invoking the WDT prescaler, longer time-out period can be realized. Writing data to WS2, WS1, WS0 (bit 2,1,0 of WDTS(09H)) can give different time-out period.

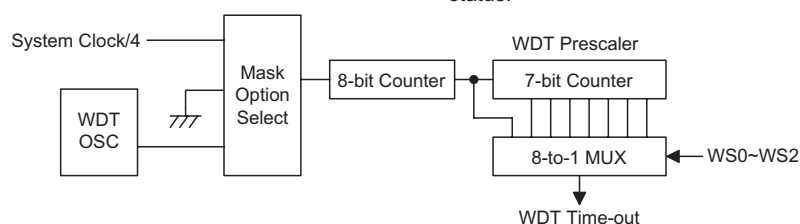
If WS2, WS1, WS0 all equal to 1, the division ratio is up to 1:128, and the maximum time-out period is 2.6 seconds.

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

The WDT overflow under normal operation will initialize a "chip reset" and set the status bit "TO". Whereas in the HALT mode, the overflow will initialize a "warm reset" only the Program Counter and SP are reset to zero. To clear the contents of the WDT (including the WDT prescaler), three methods are adopted; external reset (external reset (a low level to RES), software instructions, or a HALT instruction. The software instruction is "CLR WDT" and execution of the "CLR WDT" instruction will clear the WDT.

| WS2 | WS1 | WS0 | Division Ratio |
|-----|-----|-----|----------------|
| 0   | 0   | 0   | 1:1            |
| 0   | 0   | 1   | 1:2            |
| 0   | 1   | 0   | 1:4            |
| 0   | 1   | 1   | 1:8            |
| 1   | 0   | 0   | 1:16           |
| 1   | 0   | 1   | 1:32           |
| 1   | 1   | 0   | 1:64           |
| 1   | 1   | 1   | 1:128          |

**WDTS (09H) Register**



**Watchdog Timer**

### Power Down – HALT

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

The system oscillator will be turned off but the WDT oscillator keeps running (if the WDT oscillator is selected).

- The contents of the on chip RAM and registers remain unchanged.
- WDT and WDT prescaler will be cleared and recount again.
- All I/O ports maintain their original status.
- The PDF flag is set and the TO flag is cleared.

The system can leave the HALT mode by means of an external reset, an interrupt, an external falling edge signal on port A or a WDT overflow. An external reset causes a device initialization and the WDT overflow performs a "warm reset". By examining the TO and PDF flags, the reason for the chip reset can be determined. The PDF flag is cleared when the system powers-up or executes the "CLR WDT" instruction, and is set when the "HALT" instruction is executed. The TO flag is set if a WDT time-out occurs, and causes a wake-up that only resets the Program Counter and SP. The other maintain their original status.

The port A wake-up and interrupt methods can be considered as a continuation of normal execution. Each bit in port A can be independently selected to wake up the device by a mask option. Awakening from an I/O port stimulus, the program will resume execution of the next instruction. If awakening from an interrupt, two sequences may happen. If the related interrupt is disabled or the interrupt is enabled by the stack is full, the program will resume execution at the next instruction. If the interrupt is enabled and the stack is not full, the regular interrupt response takes place.

Once a wake-up event occurs, it takes 1024 system clock period to resume normal operation. In other words, a dummy cycle period will be inserted after a wake-up. If the wake-up results from an interrupt acknowledge, the actual interrupt subroutine will be delayed by one more cycle. If the wake-up results in next instruction execution, this will be executed immediately after a dummy period is finished. If an interrupt request flag is set to "1" before entering the HALT mode, the wake-up function of the related interrupt will be disabled. To minimize power consumption, all I/O pins should be carefully managed before entering the HALT status.



## Reset

There are 3 ways in which a reset can occur:

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

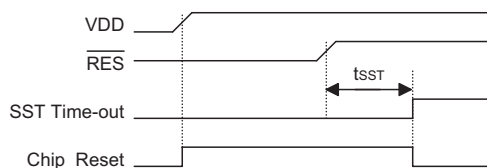
The WDT time-out during HALT is different from other chip reset conditions, since it can perform a "warm re-set" that resets only the Program Counter and SP, leaving the other circuits in their original state. Some registers remain unchanged during any other reset conditions. Most registers are reset to their "initial condition" when the reset conditions are met. By examining the PDF flag and TO flag, the program can distinguish between different "chip resets".

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

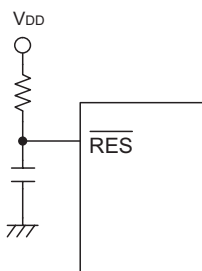
Note: "u" stands for "unchanged"

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

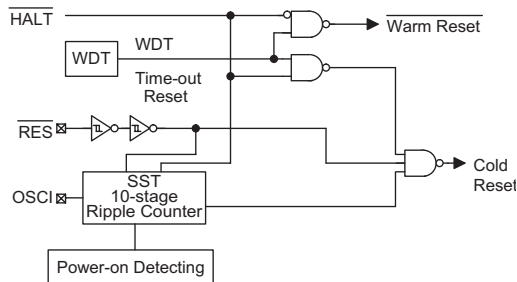
When a system power up occurs, the SST delay is added during the reset period. But when the reset comes from the  $\overline{\text{RES}}$  pin, the SST delay is disabled. Any wake-up from HALT will enable the SST delay.



**Reset Timing Chart**



**Reset Circuit**



## Reset Configuration

The function unit chip reset status are shown below.

|                     |  |
|---------------------|--|
| Program Counter     | 000H   |
| Interrupt           | Disable  |
| Prescaler           | Clear  |
| WDT                 | Clear. After master reset, WDT begins counting |
| Timer/event counter | Off  |
| Input/output ports  | Input mode                                     |
| Stack Pointer       | Points to the top of the stack                 |

## Timer/Event Counter 0/1

There are four timer counters are implemented in the HT86030/HT86070. The Timer/Event Counter 0 and 1 contain 16-bit programmable count-up counters whose clock may come from an external source or the system clock divided by 4 ( $T_1$ ). Using the internal instruction clock ( $T_1$ ), there is only one reference time base. The external clock input allows the user to count external events, measure time intervals or pulse width, or to generate an accurate time base.

There are three registers related to Timer/Event Counter 0; TMR0H (0CH), TMR0L (0DH), TMR0C (0EH). Writing to TMR0L only writes the data into a low byte buffer. Writing to TMR0H will write the data and the contents of the low byte buffer into the Timer/Event Counter 0 preload register (16-bit) simultaneously. The Timer/Event Counter 0 preload register is changed only by a write to TMR0H operation. Writing to TMR0L will keep the Timer/Event Counter 0 preload register unchanged.

Reading TMR0H will also latch the TMR0L into the low byte buffer to avoid false timing problems. Reading the TMR0L only returns the value from the low byte buffer which may be a previously loaded value. In other words, the low byte of Timer/Event Counter 0 cannot be read directly. It must read the TMR0H first to ensure that the low byte contents of Timer/Event Counter 0 are latched into the buffer.



There are three registers related to the Timer/Event Counter 1; TMR1H (0FH), TMR1L (10H), TMR1C (11H). The Timer/Event Counter 1 operates in the same manner as Timer/Event Counter 0.

| Bit No. | Label | Function  |
|---------|-------|---|
| 0~2, 5  | —     | Unused bit, read as "0"   |
| 3       | TE    | To define the TMR0/TMR1 active edge of timer/event counter (0=active on low to high; 1=active on high to low)   |
| 4       | TON   | To enable/disable timer counting (0=disabled; 1=enabled)  |
| 6       | TM0,  | To define the operating mode (TMR1, TMR0)<br>01=Event count mode (external clock)<br>10=Timer mode (internal clock)<br>11=Pulse width measurement mode<br>00=Unused |
| 7       | TM1   |   |

#### TMR0C (0EH)/TMR1C (11H) Register

The TMR0C is the Timer/Event Counter 0 control register, which defines the Timer/Event Counter 0 options. The Timer/Event Counter 1 has the same options as the Timer/Event Counter 0 and is defined by TMR1C.

The timer/event counter control registers define the operating mode, counting enable or disable and active edge.

The TM0, TM1 bits define the operating mode. The event count mode is used to count external events, which implies that the clock source comes from an external pin. The timer mode functions as a normal timer with the clock source coming from the instruction clock. The pulse width measurement mode can be used to count the high or low level duration of an external signal (TMR0/TMR1). The counting method is based on the instruction clock.

In the event count or timer mode, once the timer/event counter starts counting, it will count from the current contents in the timer/event counter to FFFFH. Once an

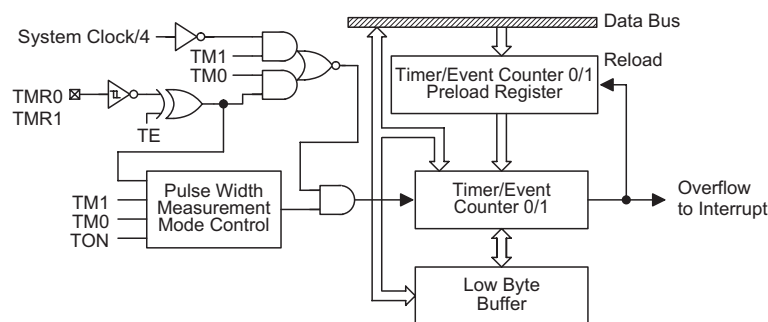
overflow occurs, the counter is reloaded from the timer/event counter preload register and generates a corresponding interrupt request flag (TOF/T1F; bit 5/6 of INTC) at the same time.

In the pulse width measurement mode with the TON and TE bits equal to one, once the TMR0/TMR1 has received a transient from low to high (or high to low; if the TE bit is 0) it will start counting until the TMR0/TMR1 returns to the original level and resets TON. The measured result will remain in the timer/event counter even if the activated transient occurs again. In other words, only one cycle measurement can be done. When TON is set again, the cycle measurement will function again as long as it receives further transient pulses. Note that, in this operating mode, the timer/event counter starts counting not according to the logic level but according to the transient edges. In the case of counter overflows, the counter is reloaded from the timer/event counter preload register and issues the interrupt request just like in the other two modes.

To enable the counting operation, the Timer ON bit (TON; bit 4 of TMR0C/TMR1C) should be set to 1. In the pulse width measurement mode, TON will be cleared automatically after the measurement cycle is complete. But in the other two modes TON can only be reset by instruction. The overflow of the timer/event counter is one of the wake-up sources. No matter what the operation mode is, writing a 0 to ET0I/ET1I can disable the corresponding interrupt service.

In the case of a Timer/Event Counter OFF condition, writing data to the timer/event counter preload register will also reload that data to the timer/event counter. But if the timer/event counter is turned on, data written to the timer/event counter will only be kept in the timer/event counter preload register. The timer/event counter will continue to operate until an overflow occurs.

When the timer/event counter (reading TMR0H/TMR1H) is read, the clock will be blocked to avoid errors. As this may result in a counting error, this must be taken into consideration by the programmer.



Timer/Event Counter 0/1



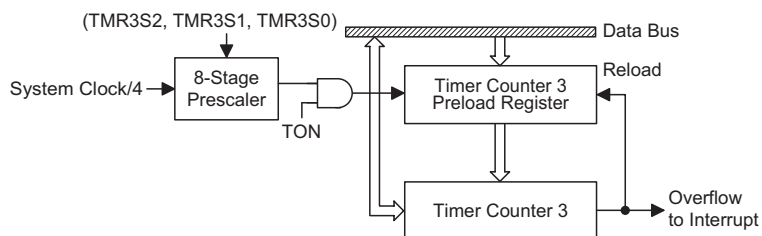
### Timer Counter 3

The timer counter TMR3 is an 8-bit programmable count-up counter. Its counting is as the same manner as Timer Event Counter 0/1, but the clock source of TMR3 can be from internal instruction cycle (T1).

The TMR3 is internal clock source only, i.e. (TM1, TM0)=(1,0). There is a 3-bit prescaler (TMR3S2, TMR3S1, TMR3S0) which defines different division ratio of TMR3's clock source.

| Bit No. | Label                        | Function  |
|---------|------------------------------|---|
| 0~2     | TMR3S2,<br>TMR3S1,<br>TMR3S0 | To define the operating clock source<br>(TMR3S2, TMR3S1, TMR3S0)<br>000: clock source/2<br>001: clock source/4<br>010: clock source/8<br>011: clock source/16<br>100: clock source/32<br>101: clock source/64<br>110: clock source/128<br>111: clock source/256 |
| 3       | TE                           | To define the TMR3 active edge of timer/event counter<br>(0=active on low to high; 1=active on high to low)   |
| 4       | TON                          | To enable/disable timer counting (0=disabled; 1=enabled)  |
| 5       | —                            | Unused bit, read as "0"   |
| 6<br>7  | TM0,<br>TM1                  | To define the operating mode<br>(TM1, TM0)<br>01=Unused<br>10=Timer mode (internal clock)<br>11=Unused<br>00=Unused   |

**TMR3C (25H) Register**



**Timer Counter 3**



The registers states are summarized in the following table.

| Register        | Reset<br>(Power On) | WDT Time-out<br>(Normal Operation) | RES Reset<br>(Normal Operation) | RES Reset<br>(HALT) | WDT Time-out<br>(HALT) |
|-----------------|---------------------|------------------------------------|---------------------------------|---------------------|------------------------|
| MP0             | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| MP1             | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| ACC             | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| Program Counter | 0000H               | 0000H                              | 0000H                           | 0000H               | 0000H                  |
| TBLP            | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| TBLH            | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| WDTS            | 0000 0111           | 0000 0111                          | 0000 0111                       | 0000 0111           | uuuu uuuu              |
| STATUS          | --00 xxxx           | --1u uuuu                          | --uu uuuu                       | --01 uuuu           | --11 uuuu              |
| INTC            | -000 0000           | -000 0000                          | -000 0000                       | -000 0000           | -uuu uuuu              |
| TMR0H           | xxxx xxxx           | xxxx xxxx                          | xxxx xxxx                       | xxxx xxxx           | xxxx xxxx              |
| TMR0L           | xxxx xxxx           | xxxx xxxx                          | xxxx xxxx                       | xxxx xxxx           | xxxx xxxx              |
| TMR0C           | xxxx xxxx           | xxxx xxxx                          | xxxx xxxx                       | xxxx xxxx           | xxxx xxxx              |
| TMR1H           | xxxx xxxx           | xxxx xxxx                          | xxxx xxxx                       | xxxx xxxx           | xxxx xxxx              |
| TMR1L           | xxxx xxxx           | xxxx xxxx                          | xxxx xxxx                       | xxxx xxxx           | xxxx xxxx              |
| TMR1C           | xxxx xxxx           | xxxx xxxx                          | xxxx xxxx                       | xxxx xxxx           | xxxx xxxx              |
| PA              | 1111 1111           | 1111 1111                          | 1111 1111                       | 1111 1111           | uuuu uuuu              |
| PAC             | 1111 1111           | 1111 1111                          | 1111 1111                       | 1111 1111           | uuuu uuuu              |
| PB              | 1111 1111           | 1111 1111                          | 1111 1111                       | 1111 1111           | uuuu uuuu              |
| PBC             | 1111 1111           | 1111 1111                          | 1111 1111                       | 1111 1111           | uuuu uuuu              |
| LATCH0H         | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| LATCH0M         | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| LATCH0L         | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| LATCH1H         | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| LATCH1M         | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| LATCH1L         | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| INTCH           | -000 ---0           | -000 ---0                          | -000 ---0                       | -000 ---0           | -uuu ---u              |
| TBHP            | ---x xxxx           | ---u uuuu                          | ---u uuuu                       | ---u uuuu           | ---u uuuu              |
| TMR3L           | xxxx xxxx           | xxxx xxxx                          | xxxx xxxx                       | xxxx xxxx           | xxxx xxxx              |
| TMR3C           | xxxx xxxx           | xxxx xxxx                          | xxxx xxxx                       | xxxx xxxx           | xxxx xxxx              |
| VOICEC          | 0--0 -00-           | u--u -uu-                          | u--u -uu-                       | u--u -uu-           | u--u -uu-              |
| DAL             | xxxx ----           | uuuu ----                          | uuuu ----                       | uuuu ----           | uuuu ----              |
| DAH             | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |
| VOL             | xxx- ----           | uuu- ----                          | uuu- ----                       | uuu- ----           | uuu- ----              |
| LATCHD          | xxxx xxxx           | uuuu uuuu                          | uuuu uuuu                       | uuuu uuuu           | uuuu uuuu              |

Note: "u" means "unchanged"

"x" means "unknown"

"-" means "undefined"



### Input/Output Ports

There are 16 bidirectional input/output lines in the microcontroller, labeled from PA to PB, which are mapped to the data memory of [12H] and [14H] respectively. All of these I/O ports can be used for input and output operations. For input operation, these ports are non-latching, that is, the inputs must be ready at the T2 rising edge of instruction "MOV A, [m]" (m=12H or 14H). For output operation, all the data is latched and remains unchanged until the output latch is rewritten.

Each I/O line has its own control register (PAC, PBC) to control the input/output configuration. With this control register, CMOS output or Schmitt trigger input with or without pull-high resistor structures can be reconfigured dynamically (i.e. on-the-fly) under software control. To function as an input, the corresponding latch of the control register must write "1". The input source also depends on the control register. If the control register bit is "1", the input will read the pad state. If the control register bit is "0", the contents of the latches will move to the internal bus. The latter is possible in the "read-modify-write" instruction.

For output function, CMOS is the only configuration. These control registers are mapped to locations 13H and 15H.

After a chip reset, these input/output lines remain at high levels or floating state (dependent on pull-high options). Each bit of these input/output latches can be set or cleared by "SET [m].i" and "CLR [m].i" (m=12H, 14H) instructions.

Some instructions first input data and then follow the output operations. For example, "SET [m].i", "CLR [m].i", "CPL [m]", "CPLA [m]" read the entire port states into the CPU, execute the defined operations (bit-operation), and then write the results back to the latches or the accumulator.

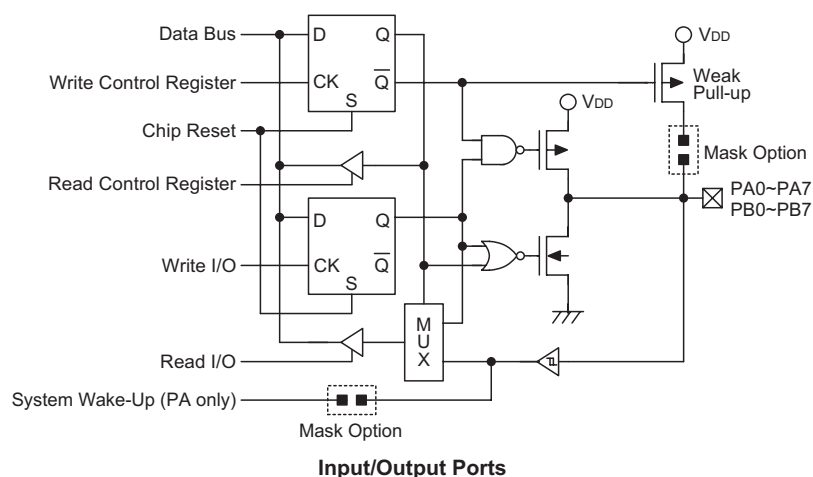
Each line of port A has the capability of waking-up the device. The wake-up capability of port A is determined by mask option. There is a pull-high option available for all I/O lines. Once the pull-high option is selected, all I/O lines have pull-high resistors. Otherwise, the pull-high resistors are absent. It should be noted that a non-pull-high I/O line operating in input mode will cause a floating state.

### Audio Output and Volume Control – DAL, DAH, VOL

The HT86030/HT86070 provides one 12-bit voltage type DAC device for driving external 8Ω speaker through an external NPN transistor. The programmer must write the voice data to register DAL (27H) and DAH (28H). The 12-bit audio output will be written to the higher nibble of DAL and the whole byte of DAH, and the DAL3~0 is always read as 0H. There are 8 scales of volume controllable level that are provided for the voltage type DAC output. The programmer can change the volume by only writing the volume control data to the higher-nibble of the VOL (29H), and the lower-nibble of VOL (29H) is always read as 0H.

### Voice Control Register

The voice control register controls the voice ROM circuit and DAC circuit, selects voice ROM latch counter. If the DAC circuit is not enabled, any DAH/DAL output is invalid. Writing a "1" to DAC bit is to enable DAC circuit, and writing a "0" to DAC bit is to disable DAC circuit. If the voice ROM circuit is not enabled, then voice ROM data cannot be accessed at all. Writing a "1" to VROMC bit is to enable the voice ROM circuit, and writing a "0" to VROMC bit is to disable the voice ROM circuit. The bit 4 (LATCHC) is to determine what voice ROM address latch counter will be adopted as voice ROM address latch counter.





| Bit No.   | Label  | Function  |
|-----------|--------|---|
| 0, 3, 5~6 | —      | Unused bit, read as "0"   |
| 1         | DAC    | Enable/disable DAC circuit (0= disable DAC circuit; 1= enable DAC circuit)<br>The DAC circuit is not affected by the HALT instruction.<br>The software controls bit DAC (VoiceC.1) whether to enable/disable. |
| 2         | VROMC  | Enable/disable voice ROM circuit (0= disable voice ROM circuit; 1= enable voice ROM circuit)  |
| 4         | LATCHC | Select voice ROM counter (0= voice ROM address latch 0; 1= voice ROM address latch 1)   |

#### VOICEC (26H) Register

##### Voice ROM Data Address Latch Counter

LATCH0H(18H)/LATCH0M(19H)/LATCH0L(1AH),  
LATCH1H(1BH)/LATCH1M(1CH)/LATCH1L(1DH) and  
voice ROM data register(2AH)

The voice ROM data address latch counter is the hand-shaking between the microcontroller and voice ROM, where the voice codes are stored. One 8-bit of voice ROM data will be addressed by setting 24-bit address latch counter LATCH0H/LATCH0M/LATCH0L or LATCH1H/LATCH1M/LATCH1L. After the 8-bit voice ROM data is addressed, a few instruction cycles (4 $\mu$ s at least) will be cost to latch the voice ROM data, then the microcontroller can read the voice data from LATCHD(2AH).

Example: Read an 8-bit voice ROM data which is located at address 000007H by address latch 0

```

set    [26H].2    ; Enable voice ROM circuit
clr    [26H].4    ; Select voice ROM address
                    ; latch counter 0
mov     A, 07H    ;
mov     LATCH0L, A ; Set LATCH0L to 07H
mov     A, 00H    ;
mov     LATCH0M, A ; Set LATCH0M to 00H
mov     A, 00H    ;
mov     LATCH0H, A ; Set LATCH0H to 00H
call    Delay Time ; Delay a short period of time
mov     A, LATCHD ; Get voice data at 000007H

```

##### Mask Option

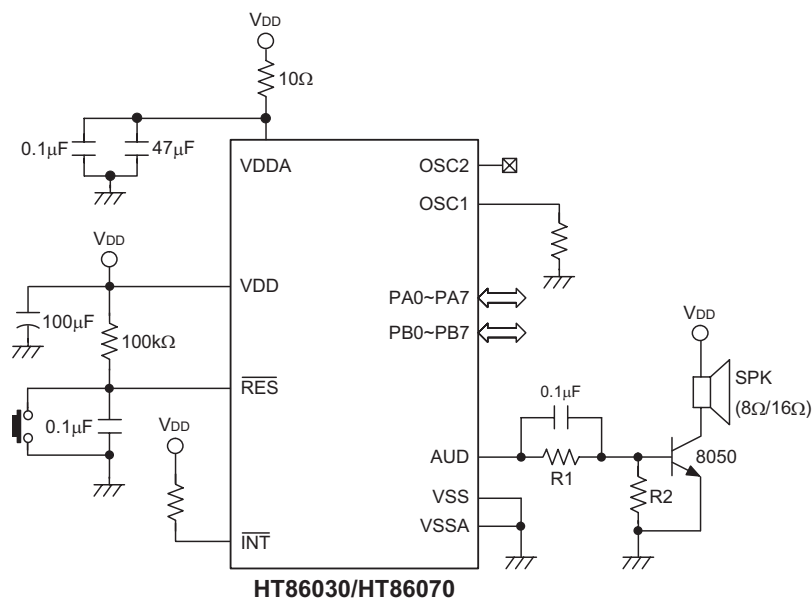
| Mask Option                     | Description  |
|---------------------------------|--|
| PA Wake-up                      | Enable/disable PA wake-up function   |
| Watchdog Timer (WDT)            | Enable/disable WDT function<br>One or two CLR instruction<br>WDT clock source is from WDTOSC or T1 |
| External INT Trigger Edge       | External INT is triggered on falling edge only, or is triggered on falling and rising edge.        |
| External Timer 0/1 Clock Source | Enable/disable external timer of timer 0 and timer 1.  |
| PA Pull-high                    | Enable/disable PA pull-high  |
| PB Pull-high                    | Enable/disable PB pull-high  |

##### fosc – R<sub>osc</sub> Table (V<sub>DD</sub>=3V)

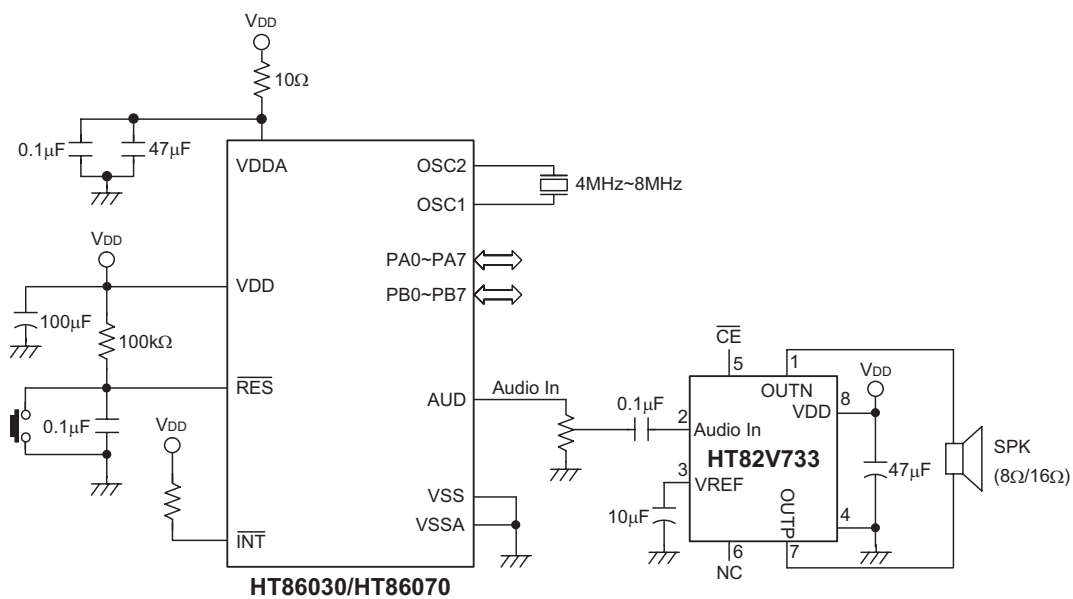
| fosc           | R <sub>osc</sub> |
|----------------|------------------|
| 4MHz $\pm$ 10% | 300k $\Omega$    |
| 6MHz $\pm$ 10% | 200k $\Omega$    |
| 8MHz $\pm$ 10% | 150k $\Omega$    |



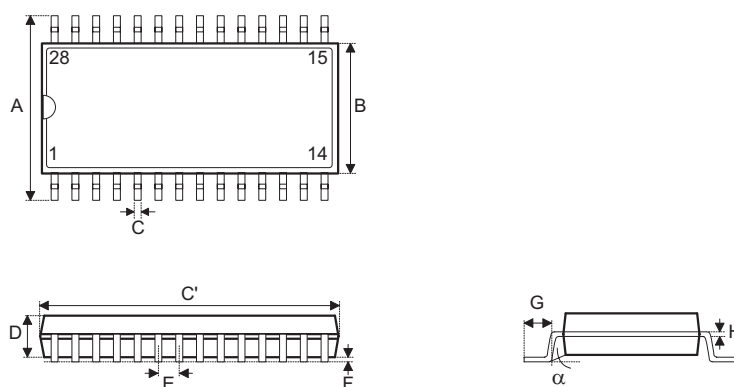
**Application Circuits**



Note: R1 > R2

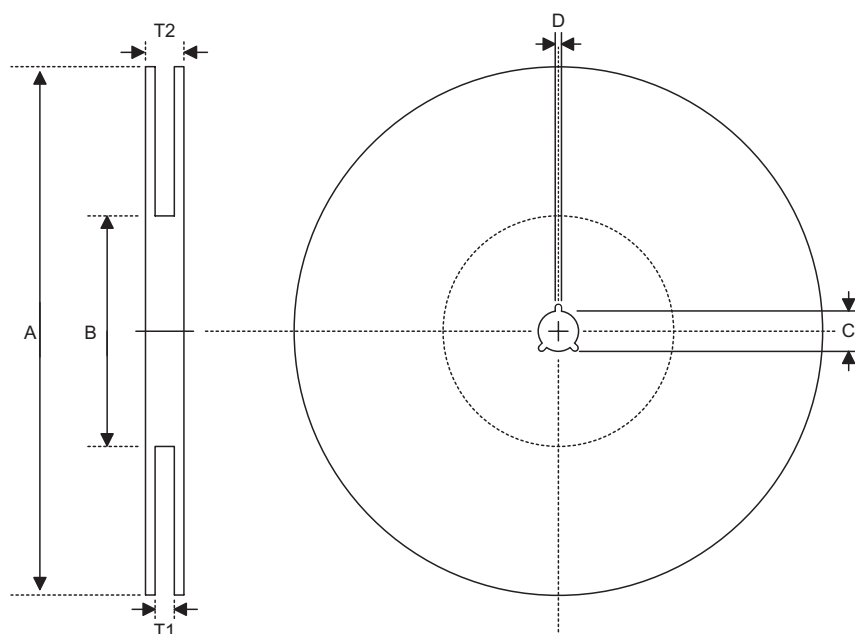




**Package Information**
**28-pin SOP (300mil) Outline Dimensions**


| Symbol   | Dimensions in mil |      |      |
|----------|-------------------|------|------|
|          | Min.              | Nom. | Max. |
| A        | 394               | —    | 419  |
| B        | 290               | —    | 300  |
| C        | 14                | —    | 20   |
| C'       | 697               | —    | 713  |
| D        | 92                | —    | 104  |
| E        | —                 | 50   | —    |
| F        | 4                 | —    | —    |
| G        | 32                | —    | 38   |
| H        | 4                 | —    | 12   |
| $\alpha$ | 0°                | —    | 10°  |

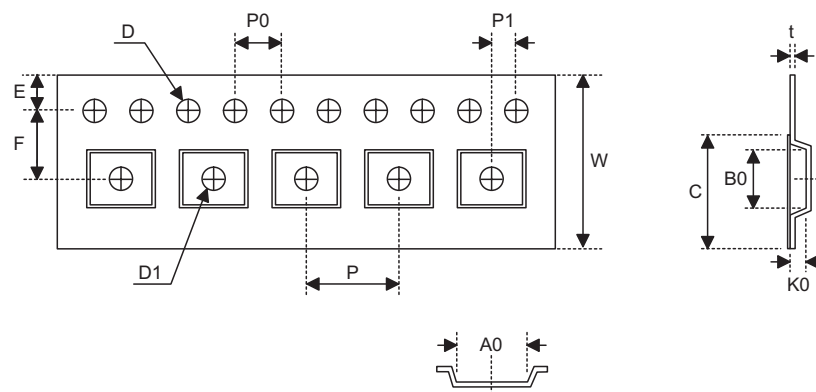


**Product Tape and Reel Specifications**
**Reel Dimensions**


SOP 28W (300mil)

| Symbol | Description           | Dimensions in mm |
|--------|-----------------------|------------------|
| A      | Reel Outer Diameter   | 330±1            |
| B      | Reel Inner Diameter   | 62±1.5           |
| C      | Spindle Hole Diameter | 13+0.5<br>-0.2   |
| D      | Key Slit Width        | 2±0.5            |
| T1     | Space Between Flange  | 24.8+0.3<br>-0.2 |
| T2     | Reel Thickness        | 30.2±0.2         |



**Carrier Tape Dimensions**


SOP 28W (300mil)

| Symbol | Description                              | Dimensions in mm |
|--------|--|------------------|
| W      | Carrier Tape Width                       | 24±0.3           |
| P      | Cavity Pitch                             | 12±0.1           |
| E      | Perforation Position                     | 1.75±0.1         |
| F      | Cavity to Perforation (Width Direction)  | 11.5±0.1         |
| D      | Perforation Diameter                     | 1.5±0.1          |
| D1     | Cavity Hole Diameter                     | 1.5±0.25         |
| P0     | Perforation Pitch                        | 4±0.1            |
| P1     | Cavity to Perforation (Length Direction) | 2±0.1            |
| A0     | Cavity Length                            | 10.85±0.1        |
| B0     | Cavity Width                             | 18.34±0.1        |
| K0     | Cavity Depth                             | 2.97±0.1         |
| t      | Carrier Tape Thickness                   | 0.35±0.01        |
| C      | Cover Tape Width                         | 21.3             |



**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 © 2006 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>.