

# HT36M4

## Music Synthesizer 8-Bit MCU

### Technical Document

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

### Features

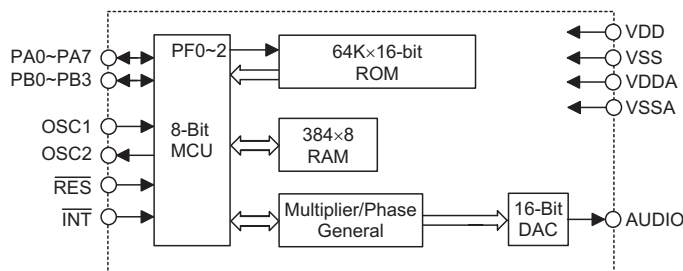
- Operating voltage: 2.6V~5.0V
- Operating frequency:
  - Crystal: 8MHz~12MHz
  - RC: 11.059MHz
- 12 bidirectional I/O lines
- Two 16-bit programmable timer/event counters with overflow interrupts
- Watchdog Timer
- Built-in 8-bit MCU with 384×8 bits RAM
- Built-in 64K×16-bit ROM for program/data shared
- Mono 16-bit DAC
- One external interrupt
- Polyphonic up to 16 notes
- Independent pan and volume mix can be assigned to each sound component
- Sampling rate of 44.1kHz as 11.059MHz for system frequency
- Eight-level subroutine nesting
- HALT function and wake-up feature reduce power consumption
- Bit manipulation instructions
- 16-bit table read instructions
- Low voltage reset 2.2V
- 63 powerful instructions
- All instructions in 1 or 2 machine cycles
- 20-pin DIP/SSOP/TSSOP package

### General Description

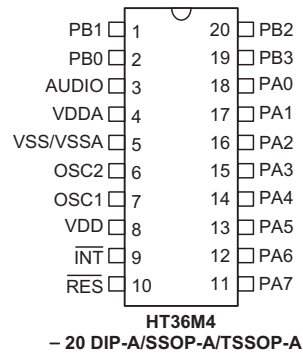
The HT36M4 is an 8-bit high performance RISC microcontroller specifically designed for music applications. It provides an 8-bit MCU and a 16 channel wavetable synthesizer. The program ROM is composed of both program control codes and wavetable voice codes, which can easily be programmed.

The HT36M4 has a built-in 8-bit microprocessor with 64K×16 program ROM, 384×8 data RAM, 12 bidirectional I/O, encapsulated in 20 TSSOP for applications where need tinny package such as ring tone generator for CELLULAR/DECT/CORDLESS PHONES.

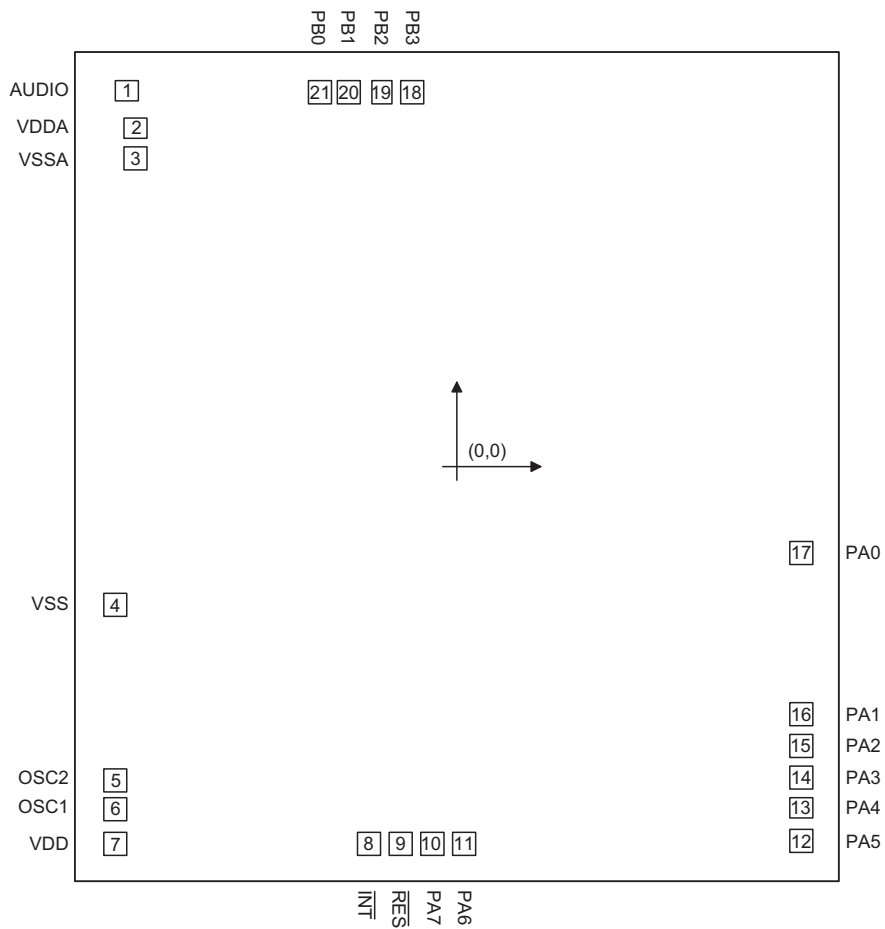
### Block Diagram



**Pin Assignment**



**Pad Assignment**



Chip size: 2595×2815 (μm)<sup>2</sup>

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

**Pad Coordinates**

 Unit:  $\mu\text{m}$ 

Pad No.	X	Y	Pad No.	X	Y
1	-1108.950	1261.050	12	1146.750	-1250.000
2	-1076.600	1135.900	13	1146.750	-1139.400
3	-1076.600	1030.900	14	1146.750	-1039.400
4	-1146.750	-460.600	15	1146.750	-928.800
5	-1146.750	-1044.234	16	1146.750	-828.800
6	-1146.750	-1147.786	17	1146.750	-286.200
7	-1146.750	-1260.200	18	-153.300	1256.550
8	-292.374	-1256.550	19	-253.300	1256.550
9	-187.326	-1256.550	20	-363.900	1256.550
10	-85.550	-1256.550	21	-463.900	1256.550
11	25.050	-1256.550			

**Pad Description**

Pad No.	Pad Name	I/O	Internal Connection	Function
1	AUDIO	O	—	Audio output
2, 3	VDDA, VSSA	—	—	DAC power supply
7, 4	VDD, VSS	—	—	Digital power supply, ground
5	OSC2	O	—	$X_{\text{OUT}}$ or 1/4 system frequency in R mode ( $f_{\text{OSC2}}=f_{\text{OSC}}/8$ )
6	OSC1	I	X'tal/Resistor	$X_{\text{IN}}$ for X'tal or $R_{\text{OSC}}$ in for resistor by mask option
8	$\overline{\text{INT}}$	I	Pull-High	External interrupt
9	$\overline{\text{RES}}$	I	—	Reset input, active low
17~10	PA7~PA0	I/O	Pull-High or None	Bidirectional 8-bit Input/Output port, wake-up by mask option
21~18	PB0~PB3	I/O	Pull-High or None	Bidirectional 8-bit input/output port

**Absolute Maximum Ratings**

Supply Voltage .....	$V_{\text{SS}}-0.3\text{V}$ to $V_{\text{SS}}+5.5\text{V}$	Storage Temperature .....	$-50^{\circ}\text{C}$ to $125^{\circ}\text{C}$
Input Voltage .....	$V_{\text{SS}}-0.3\text{V}$ to $V_{\text{DD}}+0.3\text{V}$	Operating Temperature .....	$-25^{\circ}\text{C}$ to $70^{\circ}\text{C}$

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	—	—	2.6	3.6	5	V
I <sub>DD</sub>	Operating Current	3.6V	No load (OSC on)	—	8	10	mA
I <sub>STB</sub>	Standby Current	3.6V	—	—	1	3	μA
I <sub>OH</sub>	I/O Ports Source Current	3.6V	—	2	—	—	mA
I <sub>OL</sub>	I/O Ports Sink Current	3.6V	—	3	—	—	mA
V <sub>IH</sub>	Input High Voltage	—	—	0.8V <sub>DD</sub>	—	V <sub>DD</sub>	V
V <sub>IL</sub>	Input Low Voltage	—	—	0	—	0.2V <sub>DD</sub>	V

**A.C. Characteristics**

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V <sub>DD</sub>	Conditions				
f <sub>OSC</sub>	System Frequency	5V	11.059MHz crystal	—	11.059	—	MHz
f <sub>SYS</sub>	System Clock	5V	—	4	—	16	MHz
t <sub>WDT</sub>	Watchdog Time-Out Period (RC)	—	Without WDT prescaler	9	17	35	ms
t <sub>RES</sub>	External Reset Low Pulse Width	—	—	1	—	—	μs

## Functional Description

### Execution Flow

The system clock for the HT36M4 is derived from either a crystal or an RC oscillator. The oscillator frequency divided by 2 is the system clock for the MCU ( $f_{OSC}=2 \times f_{SYS}$ ) and 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 in 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 and its contents specify a maximum of 8192 addresses for each bank.

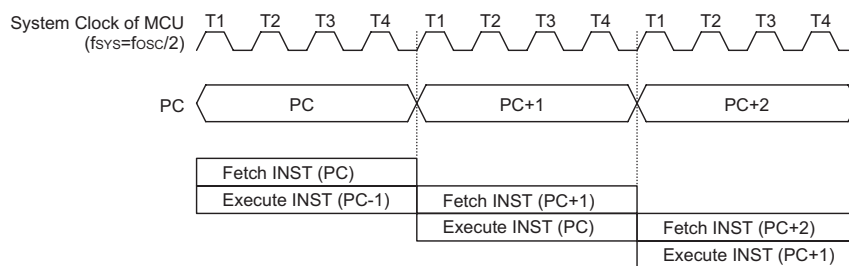
After accessing a program memory word to fetch an instruction code, the contents of the program counter are incremented by one. The program counter then points to the memory word containing the next instruction code.

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.

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

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

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



### Execution Flow

Mode	Program Counter													
	*15~*13	*12	*11	*10	*9	*8	*7	*6	*5	*4	*3	*2	*1	*0
Initial Reset	000	0	0	0	0	0	0	0	0	0	0	0	0	0
Timer/Event Counter 0 Overflow	000	0	0	0	0	0	0	0	0	0	1	0	0	0
Timer/Event Counter 1 Overflow	000	0	0	0	0	0	0	0	0	0	1	1	0	0
Skip	Program Counter+2													
Loading PCL	PF2~PF0	*12	*11	*10	*9	*8	@7	@6	@5	@4	@3	@2	@1	@0
Jump, Call Branch	PF2~PF0	#12	#11	#10	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
Return From Subroutine	S15~S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

### Program Counter

Note: PF2~PF0: Bits of Bank Register  
 \*12~\*0: Bits of Program Counter  
 @7~@0: Bits of PCL

#12~#0: Bits of Instruction Code  
 S15~S0: Bits of Stack Register

### Program ROM

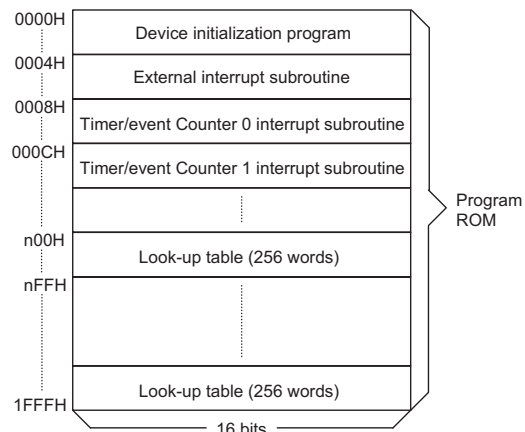
HT36M4 provides 16 address lines WA15~WA0 to read the Program ROM which is up to 1M bits, and is commonly used for the wavetable voice codes and the program memory. It provides two address types, one type is for program ROM, which is addressed by a bank pointer PF2~PF0 and a 13-bit program counter PC12~PC0; and the other type is for wavetable code, which is addressed by the start address ST0~ST11. On the program type, WA15~WA0= PF2~PF0×2<sup>13</sup>+PC12~PC0. On the wave table ROM type, WA16~WA0= ST11~ST0×2<sup>5</sup>×8-bit.

### Program Memory – ROM

The program memory is used to store the program instructions which are to be executed. It also contains data, table, and interrupt entries, and is organized into 8192×16 bits, addressed by the bank pointer, program counter and table pointer.

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

- Location 000H on bank0  
This area is reserved for the initialization program. After chip reset, the program always begins execution at location 000H on bank0.
- Location 004H  
This area is reserved for the external interrupt service program. If the INT input pin is activated, the interrupt is enabled and the stack is not full, the program will jump to location 004H and begins execution.
- Location 008H  
This area is reserved for the Timer/Event Counter 0 interrupt service program on each bank. If timer interrupt results from a Timer/Event Counter 0 overflow, and if the interrupt is enabled and the stack is not full, the program begins execution at location 008H corresponding to its bank.
- Location 00CH  
This area is reserved for the Timer/Event Counter 1 interrupt service program on each bank. 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 begins execution at location 00CH corresponding to its bank.



Note: n ranges from 00 to 1F.

### Program Memory for Each Bank

- Table location  
Any location in the ROM space can be used as look-up tables. The instructions "TABRDC [m]" (the current page, 1 page=256 words) and "TABRDL [m]" (the last page) transfer the contents of the lower-order byte to the specified data memory, and the higher-order byte to TBLH (08H). Only the destination of the lower-order byte in the table is well-defined, the higher-order byte of the table word are transferred to the TBLH. The Table Higher-order byte register (TBLH) is read only. The Table Pointer (TBLP) is a read/write register (07H), which indicates the table location. Before accessing the table, the location must be placed in the TBLP. The TBLH is read only and cannot be restored. If the main routine and the ISR (Interrupt Service Routine) both employ the table read instruction, the contents of the TBLH in the main routine are likely to be changed by the table read instruction used in the ISR. Errors can occur. In this case, using the table read instruction in the main routine and the ISR simultaneously should be avoided. However, if the table read instruction has to be applied in both the main routine and the ISR, the interrupt should be disabled prior to the table read instruction. It will not be enabled until the TBLH has been backed up. All table related instructions need 2 cycles to complete the operation. These areas may function as normal program memory depending upon user requirements.

Instruction (s)	Table Location													
	*15~*13	*12	*11	*10	*9	*8	*7	*6	*5	*4	*3	*2	*1	*0
TABRDC [m]	P15~P13	P12	P11	P10	P9	P8	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL [m]	111	1	1	1	1	1	@7	@6	@5	@4	@3	@2	@1	@0

### Table Location

Note: \*12~\*0: Bits of table location  
@7~@0: Bits of table pointer

P15~P8: Bits of current Program Counter

- Bank pointer

The program memory is organized into 8 banks and each bank into 8192×16 of bits program ROM. PF2~PF0 is used as the bank pointer. After an instruction has been executed to write data to the PF register to select a different bank, note that the new bank will not be selected immediately. The new bank is only selected after an instruction cycle is executed. When the PF register is used to select the bank, the PF register is write only. It is not until the following instruction has completed execution that the bank will be actually selected. It should be note that the PF register is write only.

### Wavetable ROM

The ST11~ST0 are used to defined the start address of each sample on the wavetable and read the waveform data from the location. HT36M4 provides 17 output address lines from WA16~WA0, the ST11~ST0 are used to locate the major 12 bits i.e. WA16~WA5 and the undefined data from WA4~WA0 are always set to 00000b. So the start address of each sample have to be located at a multiple of 32 bytes. Otherwise, the sample will not be read out correctly because it has a wrong starting code.

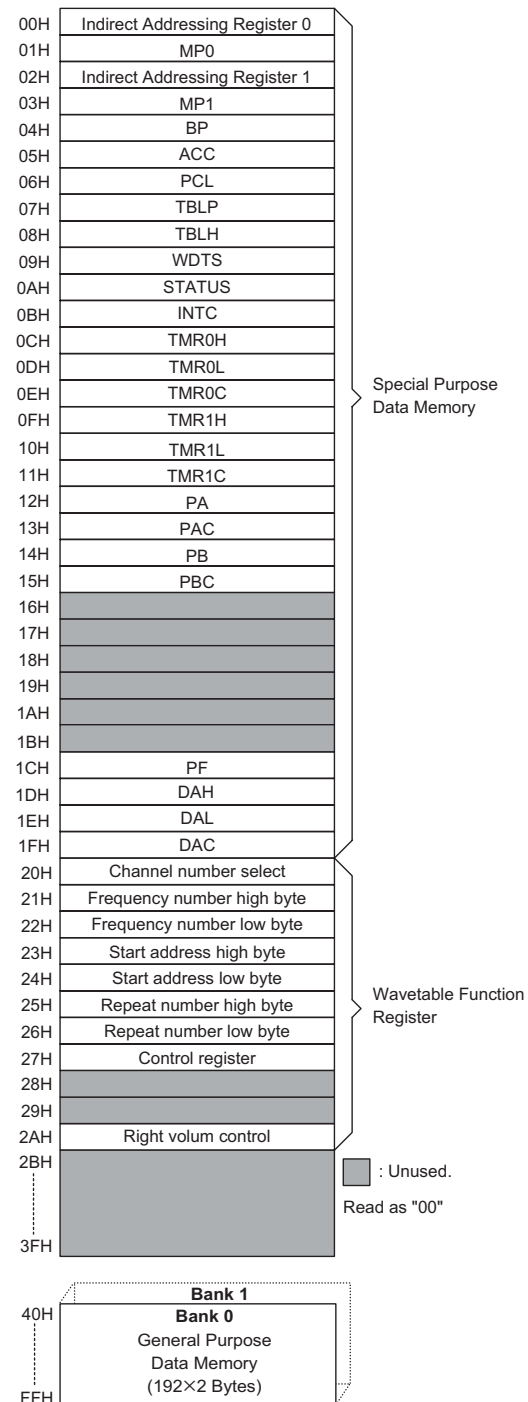
### Stack Register – Stack

This is a special part of the memory which is used to save the contents of the program counter only. The stack is organized into 8 levels and is neither part of the data nor part of the program space, and is neither readable nor writeable. The activated level is indexed by the stack pointer (SP) and is neither readable nor writeable. At a subroutine call or interrupt acknowledgment, the contents of the program counter are pushed onto the stack. At the end of a subroutine or an interrupt routine, signaled by a return instruction (RET or RETI), the program counter is restored to its previous value from the stack. After a chip reset, the Stack Pointer will point to the top of the stack.

If the stack is full and a non-masked interrupt takes place, the interrupt request flag will be recorded but the acknowledgment will be inhibited. When the stack pointer is decremented (by RET or RETI), the interrupt will be serviced. This feature prevents stack overflow allowing the programmer to use the structure more easily. In a similar case, if the stack is full and a CALL is subsequently executed, a stack overflow occurs and the first entry will be lost (only the most recent eight return address are stored).

### Data Memory – RAM

The data memory is designed with 2×256×8 bits. The data memory is divided into three functional groups: special function registers, wavetable function register, and general purpose data memory (2×192×8). Most of them are read/write, but some are read only.



**RAM Mapping**

The wavetable function registers are defined between 20H~2AH. The unused space before 40H is reserved for future expanded usage and reading these locations will return the result 00H. The general purpose data memory, addressed from 40H to FFH, is used for data and control information under instruction command.

All data memory areas can handle arithmetic, logic, increment, decrement and rotate operations directly. Except for some dedicated bits, each bit in the data memory can be set and reset by the "SET [m].i" and "CLR [m].i" instructions, respectively. They are also indirectly accessible through Memory pointer registers (MP0;01H, MP1;03H).

#### Indirect Addressing Register

Location 00H and 02H are indirect addressing registers that are not physically implemented. Any read/write operation of [00H] and [02H] access data memory pointed to by MP0 (01H) and MP1 (03H) respectively. Reading location 00H or 02H directly will return the result 00H. And writing directly results in no operation.

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

#### Accumulator

The accumulator closely relates to ALU operations. It is mapped to location 05H of the data memory and it can operate with immediate data. The data movement between two data memory locations must pass through the accumulator.

#### Arithmetic and Logic Unit – ALU

This circuit performs 8-bit arithmetic and logic operation. The ALU provides the following functions:

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

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

#### Status Register – STATUS

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

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

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

In addition, on entering the interrupt sequence or executing a subroutine call, the status register will not be automatically pushed onto the stack. If the contents of status are important and the subroutine can corrupt the status register, the programmer must take precautions to save it properly.

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. Also it is 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 either a system power-up or executing the "CLR WDT" instruction. PDF is set by executing the "HALT" instruction.
5	TO	TO is cleared by a system power-up or executing the "CLR WDT" or "HALT" instruction. TO is set by a WDT time-out.
6~7	—	Unused bit, read as "0"

**STATUS (0AH) Register**



### Interrupt

The HT36M4 provides two internal timer/event counter interrupts on each bank. The Interrupt Control register (INTC;0BH) contains the interrupt control bits that sets the 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 occur during this interval but only the interrupt request flag is recorded. If a certain interrupt needs servicing within the service routine, the programmer may set the EMI bit and the corresponding bit of the INTC 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 stack pointer is decremented. If immediate service is desired, the stack must be prevented from becoming full.

All these kinds of interrupt have a wake-up capability. As an interrupt is serviced, a control transfer occurs by pushing the program counter onto the stack and then branching to subroutines at specified locations in the program memory. Only the program counter is pushed onto the stack. If the contents of the register and Status register (STATUS) are altered by the interrupt service program which may corrupt the desired control sequence, then the programmer must save the contents first.

The internal Timer/Event Counter 0 interrupt is initialized by setting the Timer/Event Counter 0 interrupt request flag (T0F; bit 5 of the 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 Timer/Event Counter 1 interrupt is operated in the same manner as Timer/Event Counter 0. The related interrupt control bits ET1I and T1F of the Timer/Event Counter 1 are bit 3 and bit 6 of the INTC respectively.

During the execution of an interrupt subroutine, other interrupt acknowledgments are held until the RETI instruction is executed or the EMI bit and the related interrupt control bit are set to 1 (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 priorities in the following table apply. These can be masked by resetting the EMI bit.

Interrupt Source	Priority	Vector
Timer/Event Counter 0 overflow	1	08H
Timer/Event Counter 1 overflow	2	0CH

Once the interrupt request flags (T0F, T1F) are set, they will remain in the INTC register until the interrupts are serviced or cleared by a software instruction. It is recommended that a program does not use the "CALL subroutine" within the interrupt subroutine. Because interrupts often occur in an unpredictable manner or need to be serviced immediately in some applications, if only one stack is left and enabling the interrupt is not well controlled, once the "CALL subroutine" operates in the interrupt subroutine, it may damage the original control sequence.

Bit No.	Label	Function
0	EMI	Controls the Master (Global) interrupt (1=enable; 0=disable)
1	E EI	Control the external interrupt (1=enable; 0=disable)
2	ET0I	Controls the Timer/Event Counter 0 interrupt (1=enable; 0=disable)
3	ET1I	Controls the Timer/Event Counter 1 interrupt (1=enable; 0=disable)
4	EEO	External interrupt request flag (1=active; 0=inactive)
5	T0F	Internal Timer/Event Counter 0 request flag (1=active; 0=inactive)
6	T1F	Internal Timer/Event Counter 1 request flag (1=active; 0=inactive)
7	—	Unused bit, read as "0"

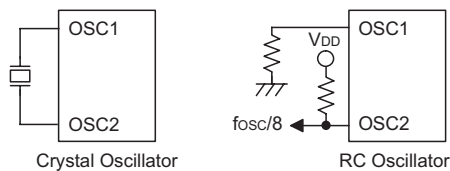
### INTC (0BH) Register

**Oscillator Configuration**

The HT36M4 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 divided by 2 is used for the system clock ( $f_{SYS}=f_{OSC}/2$ ). 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 VSS is required, and the range of the resistance should be from 30kΩ to 680kΩ. The system clock, divided by 4 ( $f_{OSC2}=f_{SYS}/4=f_{OSC}/8$ ), 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.

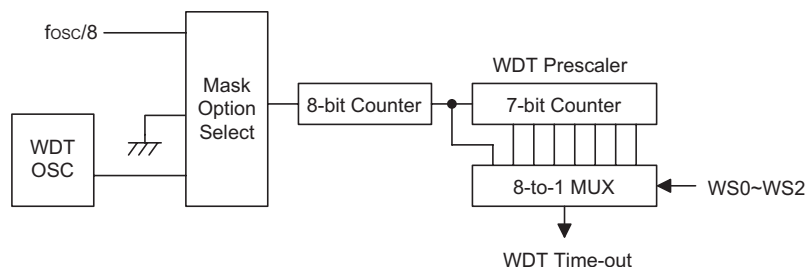
The WDT oscillator is a free running on-chip RC oscillator, and no external components are required. Even if the system enters the power down mode, the system clock is stopped, but the WDT oscillator still works with a period of approximately 78μs. The WDT oscillator can be disabled by mask option to conserve power.



**System Oscillator**

**Watchdog Timer – WDT**

The WDT clock source is implemented by a dedicated RC oscillator (WDT oscillator) or instruction clock (system clock of the MCU divided by 4), determined by mask options. This timer is designed to prevent a software



**Watchdog Timer**

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 executions related to the WDT result in no operation.

Once the internal WDT oscillator (RC oscillator with a period of 78μs normally) is selected, it is first divided by 256 (8-stages) to get the nominal time-out period of approximately 20ms. This time-out period may vary with temperature, VDD and process variations. By invoking the WDT prescaler, longer time-out periods can be realized. Writing data to WS2, WS1, WS0 (bits 2, 1, 0 of the WDTS) can give different time-out periods. 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 WDT oscillator is disabled, the WDT clock may still come from the instruction clock and operate in the same manner except that in the HALT state the WDT may stop counting and lose its protecting purpose. In this situation the logic can only be restarted by external logic. The high nibble and bit 3 of the WDTS are reserved for user defined flags, and the programmer may use these flags to indicate some specified status.

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

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 Stack Pointer are reset to zero. To clear the WDT contents (including the WDT prescaler), three methods are implemented; external reset (a low level to RES), software instructions, or a HALT

instruction. The software instructions include "CLR WDT" and the other set – "CLR WDT1" and "CLR WDT2". Of these two types of instructions, only one can be active depending on the mask option – "CLR WDT times selection option". If the "CLR WDT" is selected (i.e. CLRWDT times equal one), any execution of the "CLR WDT" instruction will clear the WDT. In case "CLR WDT1" and "CLR WDT2" are chosen (i.e. CLRWDT times equal two), these two instructions must be executed to clear the WDT; otherwise, the WDT may reset the chip due to time-out.

**Power Down Operation – HALT**

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

- The system oscillator will turn off but the WDT oscillator keeps running (if the WDT oscillator is selected). Watchdog Timer – WDT
- The contents of the on-chip RAM and registers remain unchanged.
- The WDT and WDT prescaler will be cleared and starts to count again (if the clock comes from the WDT oscillator).
- All I/O ports maintain their original status.
- The PDF flag is set and the TO flag is cleared.
- The HALT pin will output a high level signal to disable the external ROM.

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 cause for a chip reset can be determined. The PDF flag is cleared when there is a system power-up or by executing the "CLR WDT" instruction and it is set when a 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 Stack Pointer, the others remain in 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 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 occur. If the related interrupts is disabled or the interrupts is enabled but the stack is full, the program will resume execution at the next instruction. If the interrupt is enabled and the stack is not full, a regular interrupt response takes place.

Once a wake-up event occurs, it takes  $1024 t_{SYS}$  (system clock period) to resume to 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 de-

layed by one more cycle. If the wake-up results in next instruction execution, this will be executed immediately after a dummy period has 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.

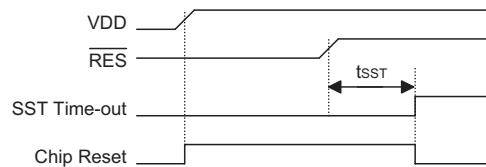
To minimize power consumption, all I/O pins should be carefully managed before entering the HALT status.

**Reset**

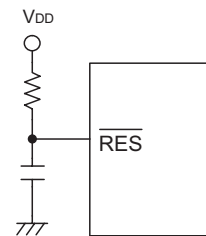
There are three ways in which a reset can occur:

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

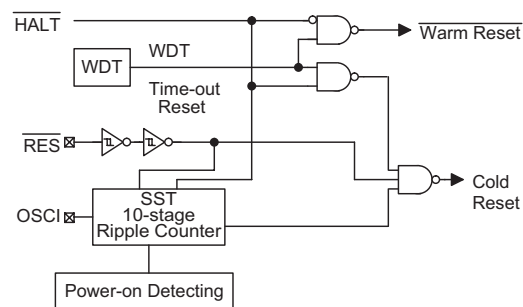
The WDT time-out during HALT is different from other chip reset conditions, since it can perform a "warm re-set" that resets the Program Counter and Stack Pointer, leaving the other circuits in their original state. Some registers remain unchanged during other reset conditions. Most registers are reset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



**Reset Timing Chart**



**Reset Circuit**



**Reset Configuration**

The registers' status is summarized in the following table:

Register	Reset (Power On)	WDT Time-out (Normal Operation)	RES Reset (Normal Operation)	RES Reset (HALT)	WDT Time-out (HALT)*
Program Counter	0000H	0000H	0000H	0000H	0000H
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
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	uuuu uuuu
TMR0L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0C	00-0 1---	00-0 1---	00-0 1---	00-0 1---	uu-u u---
TMR1H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR1L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR1C	00-0 1---	00-0 1---	00-0 1---	00-0 1---	uu-u u---
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	---- uuuu
PBC	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu
PF	---- -000	---- -000	---- -000	---- -000	---- -uuu
CHAN	00-- 0000	uu-- uuuu	uu-- uuuu	uu-- uuuu	uu-- uuuu
FreqNH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
FreqNL	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
AddrH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
AddrL	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
ReH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
ReL	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
ENV	x-xx xxxx	u-uu uuuu	u-uu uuuu	u-uu uuuu	u-uu uuuu
RVC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
DAH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
DAL	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
DAC	---- -00	---- -00	---- -00	---- -00	---- -uu

Note: "\*" stands for warm reset  
 "u" stands for unchanged  
 "x" stands for unknown

TO		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 during system power up or when the system awakes 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.

The functional units chip reset status are shown below.

Program Counter	000H
Interrupt	Disable
Prescaler	Clear
WDT	Clear. After master reset, WDT begins counting
Timer/Event Counter (0/1)	Off
Input/output ports	Input mode
Stack Pointer	Points to the top of the stack

### Timer/Event Counter

Two timer/event counters are implemented in the HT36M4. The Timer/Event Counter 0 and Timer/Event Counter 1 contain 16-bit programmable count-up counters and the clock comes from the system clock divided by 4.

There are three registers related to Timer/Event Counter 0; TMR0H (0CH), TMR0L (0DH), TMR0C (0EH).

Bit No.	Label	Function
0~2	—	Unused bit, read as "0"
3	TE	Defines the TMR active edge of the Timer/Event Counter 0 (0=active on low to high; 1=active on high to low)
4	TON	Enable/disable timer counting (0=disable; 1=enable)
5	—	Unused bit, read as "0"
6 7	TM0 TM1	Defines the operating mode 01=Event count mode (External clock) 10=Timer mode (Internal clock) 11=Pulse width measurement mode 00=Unused

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

Writing TMR0L only writes the data into a low byte buffer, and writing 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 by writing TMR0H operations and writing 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 a false timing problem. Reading TMR0L returns the contents of the low byte buffer. In other words, the low byte of the Timer/Event Counter 0 cannot be read directly. It must read the TMR0H first to make the low byte contents of the Timer/Event Counter 0 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.

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 with 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 means the clock source comes from an external (TMR) 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 the external signal (TMR). The counting 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 overflow occurs, the counter is reloaded from the Timer/Event Counter Preload register and simultaneously generates the corresponding interrupt request flag (T0F/T1F; bit 5/6 of INTC).

In pulse width measurement mode with the TON and TE bits equal to one, once the TMR has received a transient from low to high (or high to low; if the TE bit is 0) it will start counting until the TMR returns to the original level and resets the TON. The measured result will remain in the timer/event counter even if the activated transient occurs again. In other words, only one cycle measurements can be done. Until setting the TON, the cycle measurement will function again as long as it receives further transient pulse. 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 overflow, the counter is reloaded from the timer/event counter preload register and issues the interrupt request just like the other two modes.

To enable the counting operation, the Timer ON bit (TON; bit 4 of the TMR0C/TMR1C) should be set to 1. In the pulse width measurement mode, the TON will be cleared automatically after the measurement cycle is completed. But in the other two modes the 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 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 still operate until 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.

The two timer counters of the HT36M4 are internal clock mode only, so only Timer mode can be selected. Therefore the (TM1, TM0) bits can only be set to (TM1, TM0) = (1, 0), and the other clock modes are invalid.

**Input/Output Ports**

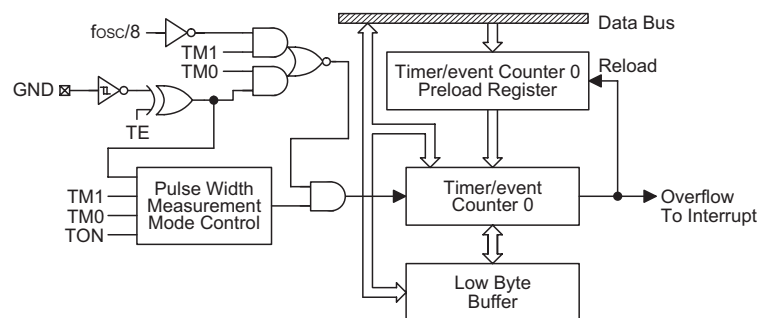
There are 12 bidirectional input/output lines labeled from PA, which are mapped to the data memory of [12H], [14H] respectively. All these I/O ports can be used for input and output operations. For input operation, these ports are non-latching, that is, the inputs must be ready at the T2 rising edge of instruction "MOV A,[m]" (m=12H, 14H). For output operation, all 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 (mask option) structures can be reconfigured dynamically under software control. To function as an input, the corresponding latch of the control register must write a "1". The pull-high resistance will exhibit automatically if the pull-high option is selected. The input source also depends on the control register. If the control register bit is "1", 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 "read-modify-write" instruction. For output function, CMOS is the only configuration. These control registers are mapped to locations 13H and 15H).

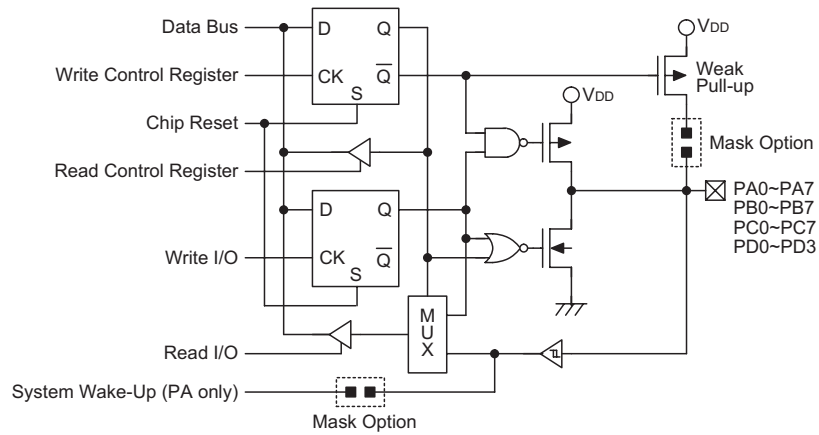
After a chip reset, these input/output lines remain at high levels or floating (mask option). Each bit of these input/output latches can be set or cleared by the "SET [m].i" or "CLR [m].i" (m=12H or 14H) instruction.

Some instructions first input data and then follow the output operations. For example, the "SET [m].i", "CLR [m].i", "CPL [m]" and "CPLA [m]" instructions 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 to wake-up the device.



Timer/Event Counter 0/1



Input/Output Ports

16 Channel Wavetable Synthesizer

Register Name	Register Function	B7	B6	B5	B4	B3	B2	B1	B0
1DH	DAC High Byte (No Default Value)	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
1EH	DAC Low Byte (No Default Value)	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
1FH	DAON=1: DAC ON DANO=0: DAC OFF (Default) SELW=1: DAC Data from Wavetable SELW=0: DAC Data from MCU (Default)	—	—	—	—	—	—	DAON	SELW — Right
20H	Channel Number Select	VM	FR	—	—	CH3	CH2	CH1	CH0
21H	High Byte Frequency Number	BL3	BL2	BL1	BL0	FR11	FR10	FR9	FR8
22H	Low Byte Frequency Number	FR7	FR6	FR5	FR4	FR3	FR2	FR1	FR0
23H	High Byte Start Address Selection	—	—	—	—	ST11	ST10	ST9	ST8
24H	Low Byte Start Address Selection	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
25H	Waveform Bit Selection High Byte Repeat Number Selection	WBS	RE14	RE13	RE12	RE11	RE10	RE9	RE8
26H	Low Byte Repeat Number Selection	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0
27H	Envelope Control Selection Left/Right Volume Control	A_R	—	—	—	—	—	VR9	VR8
28H		—							
29H		—							
2AH	Right Volume Controller	VR7	VR6	VR5	VR4	VR3	VR2	VR1	VR0
2B~2FH		—							
30H~1FFH	Data Memory (RAM)	General Purpose Data Memory (Same As 8-bit MCU)							

Wavetable Function Register Table

Note: 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 the DAC circuit.

- CH3~CH0 channel number selection

The HT36M4 has a built-in 16 output channels and CH3~CH0 is used to define which channel is selected. When this register is written to, the wavetable synthesizer will automatically output the dedicated PCM code. So this register is also used as a start playing key and it has to be written to after all the other wavetable function registers are already defined.

- Change parameter selection

These two bits, VM and FR, are used to define which register will be updated on this selected channel. There are two modes that can be selected to reduce the process of setting the register. Please refer to the statements of the following table:

VM	FR	Function
0	0	Update all the parameter
0	1	Only update the frequency number
1	0	Only update the volume

- Output frequency definition

The data on BL3~BL0 and FR11~FR0 are used to define the output speed of the PCM file, i.e. it can be used to generate the tone scale. When the FR11~FR0 is 800H and BL3~BL0 is 6H, each sample data of the PCM code will be sent out sequentially.

When the  $f_{OSC}$  is 12.8MHz, the formula of a tone frequency is:

$$f_{OUT} = f_{RECORD} \times \frac{50kHz}{SR} \times \frac{FR11 \sim FR0}{2^{(17 - BL3 - BL0)}}$$

where  $f_{OUT}$  is the output signal frequency,  $f_{RECORD}$  and SR is the frequency and sampling rate on the sample code, respectively.

So if a voice code of C3 has been recorded which has the  $f_{RECORD}$  of 261Hz and the SR of 11025Hz, the tone frequency ( $f_{OUT}$ ) of G3:  $f_{OUT} = 196Hz$ .

Can be obtained by using the formula:

$$196Hz = 261Hz \times \frac{50kHz}{11025Hz} \times \frac{FR11 \sim FR0}{2^{(17 - BL3 - BL0)}}$$

A pair of the values FR11~FR0 and BL3~BL0 can be determined when the  $f_{OSC}$  is 12.8MHz.

- Start address definition

The HT36M4 provides two address types for extended use, one is the program ROM address which is program counter corresponding with PF value, the other is the start address of the PCM code.

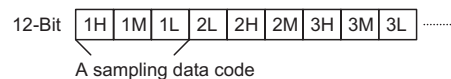
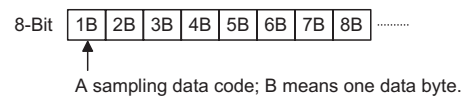
The ST11~ST0 is used to define the start address of each PCM code and reads the waveform data from this location. The HT36M4 provides 17 input data lines from WA16~WA0, the ST11~ST0 is used to locate the major 16 bits i.e. WA16~WA5 and the undefined data from WA4~WA0 is always set as 00000b. In other words, the  $WA16 \sim WA0 = ST11 \sim ST0 \times 2^5 \times 8$ -bit. So each PCM code has to be located at a multiple of 32. Otherwise, the PCM code will not be read out correctly because it has a wrong start code.

- Waveform format definition

The HT36M4 accepts two waveform formats to ensure a more economical data space. WBS is used to define the sample format of each PCM code.

- ♦ WBS=0 means the sample format is 8-bit
- ♦ WBS=1 means the sample format is 12-bit

The 12-bit sample format allocates location to each sample data. Please refer to the waveform format statement as shown below.



Note: "1H" High Nibble  
"1M" Middle Nibble  
"1L" Low Nibble

### Waveform Format

- Repeat number definition

The repeat number is used to define the address which is the repeat point of the sample. When the repeat number is defined, it will be output from the start code to the end code once and always output the range between the repeat address to the end code (80H) until the volume becomes close.

The RE14~RE0 is used to calculate the repeat address of the PCM code. The process for setting the RE14~RE0 is to write the 2's complement of the repeat length to RE14~RE0, with the highest carry ignored. The HT36M4 will get the repeat address by adding the RE14~RE0 to the address of the end code, then jump to the address to repeat this range.

- Left and right volume control

The HT36M4 provides the left and right volume control independently. The left and right volume are controlled by VL9~VL0 and VR9~VR0 respectively. The chip provides 1024 levels of controllable volume, the 000H is the maximum and 3FFH is the minimum output volume.

- Envelope type definition

The HT36M4 provides a function to easily program the envelope by setting the data of ENV1~ENV0 and A\_R. It forms a vibrato effect by a change of the volume to attach and release alternately.

The A\_R signal is used to define the volume change in attach mode or release mode and ENV1~ENV0 is used to define which volume control bit will be changeable. On the attach mode, the control bits will be sequentially signaled down to 0. On the release mode, the control bits will be sequentially signaled up to 1. The relationship is shown in the following table.



A_R	ENV1	ENV0	Volume Control Bit	Control Bit Final Value	Mode
0	0	0	VL2~VL0, VR2~VR0	111b	Release mode
0	0	1	VL1~VL0, VR1~VR0	11b	
0	1	0	VL0, VR0	1b	
x	1	1	No Bit	Unchanged	No change mode
1	0	0	VL2~VL0, VR2~VR0	000b	Attach mode
1	0	1	VL1~VL0, VR1~VR0	00b	
1	1	0	VL0, VR0	0b	

#### Envelope Type Definition

- The PCM code definition

The HT36M4 can only solve the voice format of the signed 8-bit raw PCM. And the MCU will take the voice code 80H as the end code.

So each PCM code section must be ended with the end code 80H.

#### Digital to Analog Converter (DAC)

The HT36M4 provides one 16-bit voltage type DAC device controlled by the MCU or wavetable synthesizer for driving an external speaker through an external NPN transistor. It is in fact an optional object used for Wavetable Synthesizer DAC or general DAC, this is chosen by DAC control register. If general DAC is chosen for application, then Wavetable synthesizer is disabled because the DAC is taken up and controlled by the MCU. If general DAC is selected, the programmer must write the voice data to register DAL and DAH to get the corresponding analog data. If Mask Option enables the DAC register and enables the SELW, then the following table comes useful.

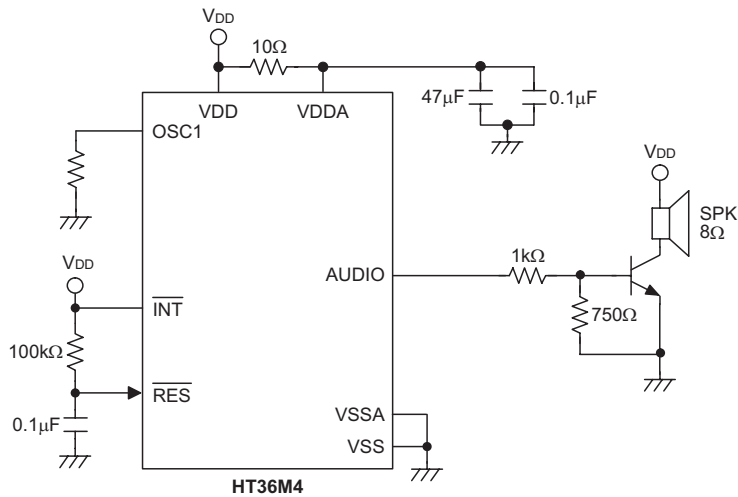
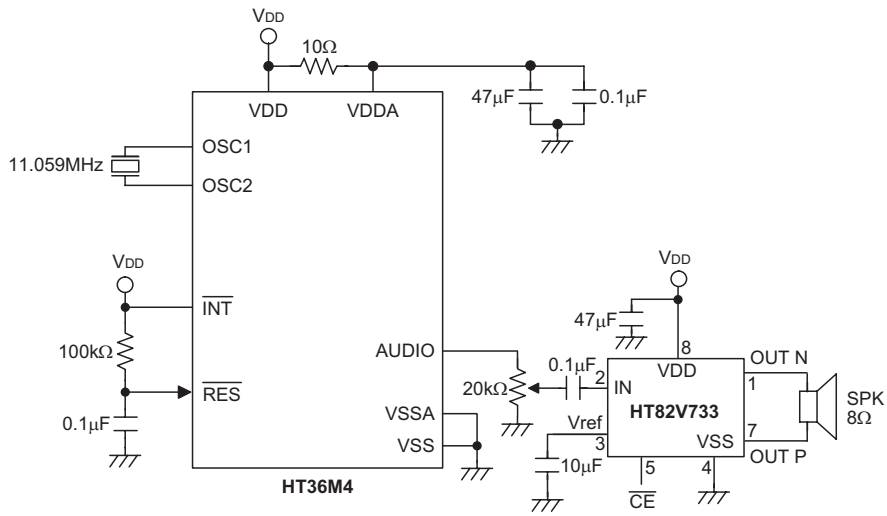
Bit No.	Label	Function
Bit7~Bit3	—	Unused
Bit2	—	Unused
Bit1	DANO	DAON=1: DAC ON DAON=0: DAC OFF (Default)
Bit0	SELWR	SELWR=1: Right Channel DAC data from Wavetable SELWR=0: Right Channel DAC data from MCU (default)

#### DAC (1FH) Control Regulation

#### Mask Option

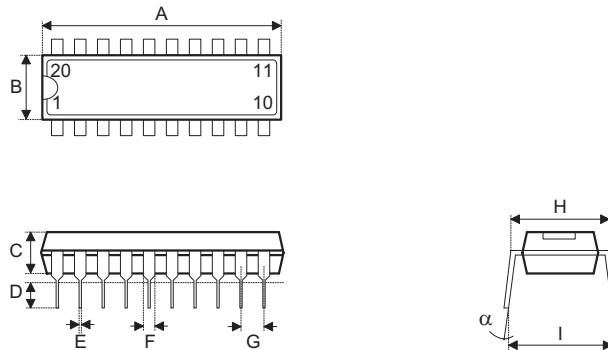
No	Mask Option	Description
1	WDT Source	On-chip RC/instruction clock/disable WDT
2	CLRWDT Times	One time, two times (CLR WDT1/WDT2)
3	Wake-up	PA only
4	Pull-high	PA, PB0~PB3 input
5	OSC Mode	Crystal or resistor type
6	LVR	Enable or disable

Application Circuit



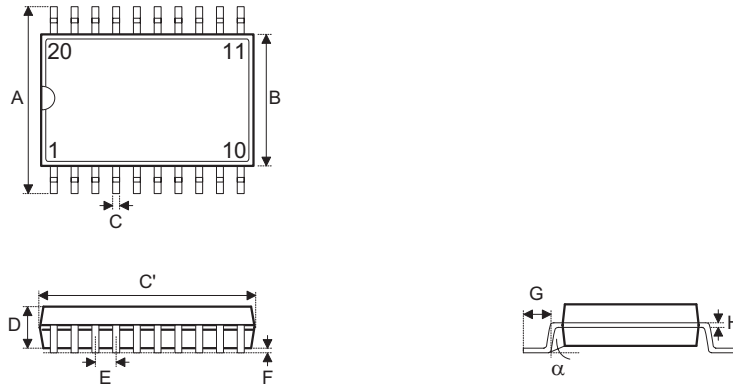
**Package Information**

**20-pin DIP (300mil) Outline Dimensions**



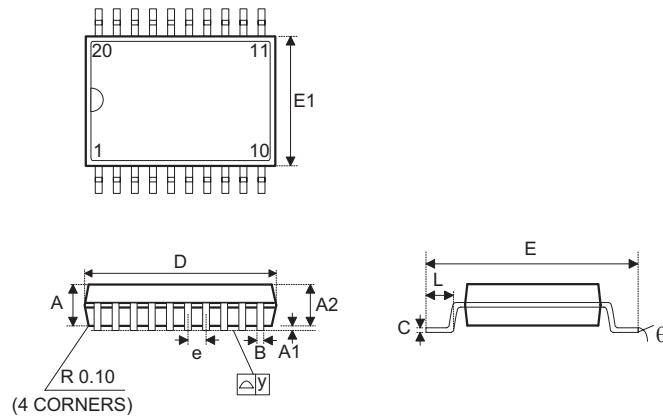
Symbol	Dimensions in mil		
	Min.	Nom.	Max.
A	1020	—	1045
B	240	—	260
C	125	—	135
D	125	—	145
E	16	—	20
F	50	—	70
G	—	100	—
H	295	—	315
I	335	—	375
$\alpha$	0°	—	15°

20-pin SSOP (150mil) Outline Dimensions



Symbol	Dimensions in mil		
	Min.	Nom.	Max.
A	228	—	244
B	150	—	158
C	8	—	12
C'	335	—	347
D	49	—	65
E	—	25	—
F	4	—	10
G	15	—	50
H	7	—	10
$\alpha$	0°	—	8°

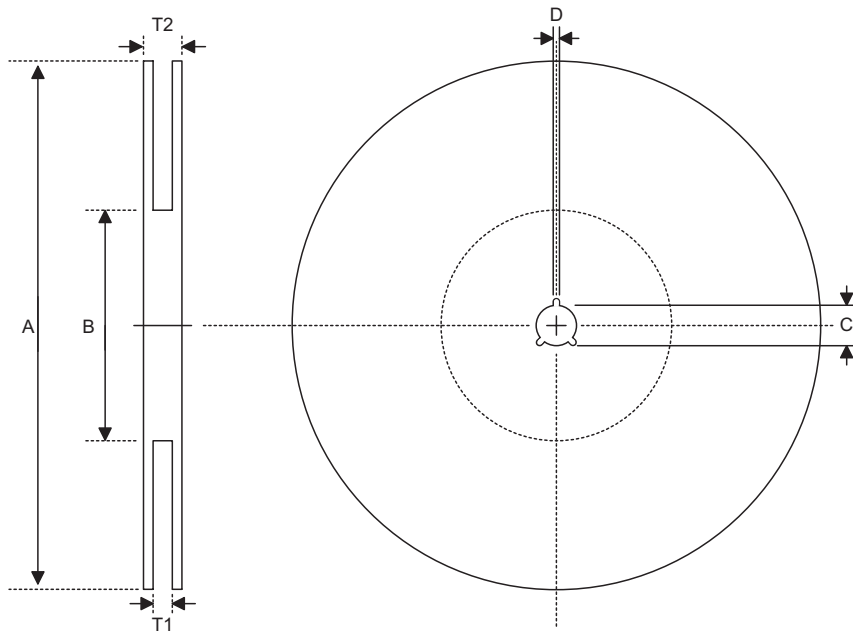
20-pin TSSOP Outline Dimensions



Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	1.05	—	1.2
A1	0.05	—	0.15
A2	0.95	—	1.05
B	—	0.22	—
C	0.13	—	0.17
D	6.4	—	6.6
E	6.3	—	6.5
E1	4.3	—	4.5
e	—	0.65	—
L	0.45	—	0.75
y	—	—	0.1
$\theta$	0°	—	8°

**Product Tape and Reel Specifications**

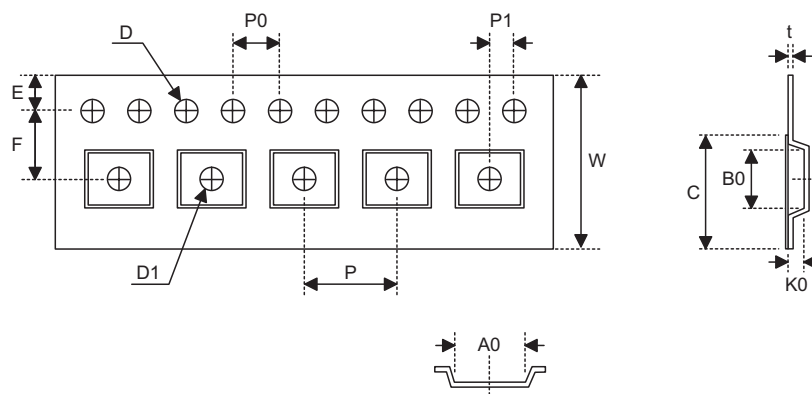
**Reel Dimensions**



SSOP 20S (150mil)

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 -0.2
D	Key Slit Width	2±0.5
T1	Space Between Flange	16.8+0.3 -0.2
T2	Reel Thickness	22.2±0.2

**Carrier Tape Dimensions**



SSOP 20S (150mil)

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	16+0.3 -0.1
P	Cavity Pitch	8±0.1
E	Perforation Position	1.75±0.1
F	Cavity to Perforation (Width Direction)	7.5±0.1
D	Perforation Diameter	1.5+0.1
D1	Cavity Hole Diameter	1.5+0.25
P0	Perforation Pitch	4±0.1
P1	Cavity to Perforation (Length Direction)	2±0.1
A0	Cavity Length	6.5±0.1
B0	Cavity Width	9±0.1
K0	Cavity Depth	2.3±0.1
t	Carrier Tape Thickness	0.3±0.05
C	Cover Tape Width	13.3

**Holtek Semiconductor Inc. (Headquarters)**

No.3, Creation Rd. II, Science Park, Hsinchu, Taiwan  
Tel: 886-3-563-1999  
Fax: 886-3-563-1189  
<http://www.holtek.com.tw>

**Holtek Semiconductor Inc. (Taipei Sales Office)**

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

**Holtek Semiconductor Inc. (Shanghai Sales Office)**

7th Floor, Building 2, No.889, Yi Shan Rd., Shanghai, China 200233  
Tel: 86-21-6485-5560  
Fax: 86-21-6485-0313  
<http://www.holtek.com.cn>

**Holtek Semiconductor Inc. (Shenzhen Sales Office)**

5/F, Unit A, Productivity Building, Cross of Science M 3rd Road and Gaoxin M 2nd Road, Science Park, Nanshan District, Shenzhen, China 518057  
Tel: 86-755-8616-9908, 86-755-8616-9308  
Fax: 86-755-8616-9722

**Holtek Semiconductor Inc. (Beijing Sales Office)**

Suite 1721, Jinyu Tower, A129 West Xuan Wu Men Street, Xicheng District, Beijing, China 100031  
Tel: 86-10-6641-0030, 86-10-6641-7751, 86-10-6641-7752  
Fax: 86-10-6641-0125

**Holtek Semiconductor Inc. (Chengdu Sales Office)**

709, Building 3, Champagne Plaza, No.97 Dongda Street, Chengdu, Sichuan, China 610016  
Tel: 86-28-6653-6590  
Fax: 86-28-6653-6591

**Holtek Semiconductor (USA), Inc. (North America Sales Office)**

46729 Fremont Blvd., Fremont, CA 94538  
Tel: 1-510-252-9880  
Fax: 1-510-252-9885  
<http://www.holmate.com>

Copyright © 2007 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>.