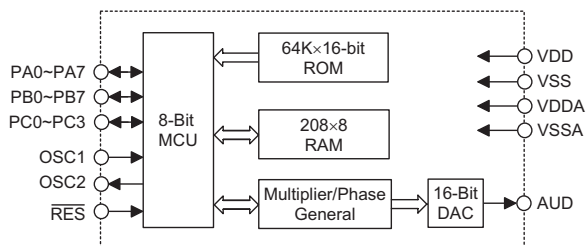# HT36A2
8-Bit Music Synthesizer MCU

Features

- Operating voltage: 2.4V~5.0V
- Operating frequency: 3.58MHz~12MHz (typ. 8MHz)
- 20 bidirectional I/O lines
- Two 8-bit programmable timer with 8 stage prescaler
- Watchdog Timer
- Built-in 8-bit MCU with 208×8 bits RAM
- Built-in 64K×16-bit ROM for program/data shared
- Mono output
- High D/A converter resolution: 16 bits
- Polyphonic up to 8 notes
- Independent volume mix can be assigned to each sound component

- Sampling rate of 25kHz as 6.4MHz for system frequency
- Eight-level subroutine nesting
- HALT function and wake-up feature to reduce power consumption
- Bit manipulation instructions
- 16-bit table read instructions
- 63 powerful instructions
- All instructions in 1 or 2 machine cycles
- 28-pin SOP, 48-pin SSOP package

General Description

The HT36A2 is an 8-bit high performance RISC-like microcontroller specifically designed for music applications. It provides an 8-bit MCU and a 8 channel wavetable synthesizer. The program ROM is composed of both program control codes and wavetable voice codes, and can be easily programmed.

The HT36A2 has a built-in 8-bit microprocessor which programs the synthesizer to generate the melody by setting the special register from 20H~2AH. A HALT feature is provided to reduce power consumption.
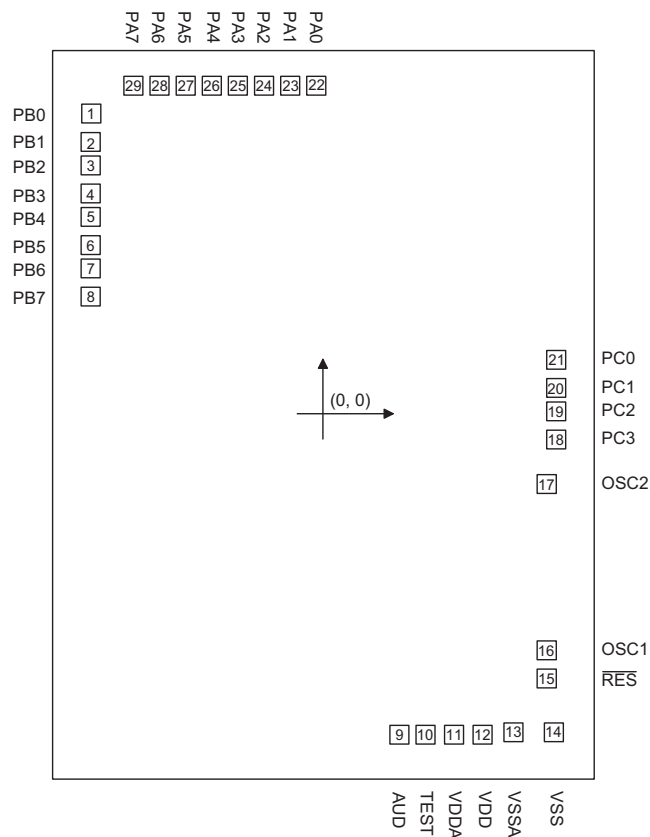
Block Diagram

Pin Assignment

| | HT36A2 − 28 SOP-A |
| | HT36A2 − 48 SSOP-A |

Pad Assignment

Chip size: 87.2 × 118.3 (mil)

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

Pad Coordinates

Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	−939.550	1222.175	16	899.200	−965.126
2	−939.550	1111.575	17	899.200	−287.274
3	−939.550	1011.575	18	938.800	−100.675
4	−939.550	900.975	19	938.800	9.925
5	−939.550	800.975	20	938.800	109.925
6	−939.550	690.375	21	938.800	220.525
7	−939.550	590.375	22	−26.300	1336.775
8	−939.550	479.775	23	−136.900	1336.775
9	301.450	−1303.400	24	−236.900	1336.775
10	413.250	−1303.400	25	−347.500	1336.775
11	526.890	−1303.400	26	−447.500	1336.775
12	636.250	−1303.400	27	−558.100	1336.775
13	761.700	−1296.750	28	−658.100	1336.775
14	924.250	−1296.750	29	−768.700	1336.775
15	899.200	−1078.450			

Pad Description

Pad Name	I/O	Internal Connection	Function
PB0~PB7	I/O	Pull-High or None	Bidirectional 8-bit Input/Output port
AUD	O	—	Audio output for driving a external transistor or for driving HT82V733
TEST	—	—	No connection (open)
VDDA	—	—	DAC power supply
VDD	—	—	Positive power supply
VSSA	—	—	Negative power supply of DAC, ground
VSS	—	—	Negative power supply, ground
$\overline{RES}$	I	—	Reset input, active low
OSC1 OSC2	I O	—	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 1/8 system clock. The system clock may come from the crystal, the two pins cannot be floating.
PC0~PC3	I/O	Pull-High or None	Bidirectional 4-bit Input/Output port
PA0~PA7	I/O	Pull-High or None	Bidirectional 8-bit Input/Output port, wake-up by mask option

Absolute Maximum Ratings

Supply Voltage ....V_{SS}−0.3V to V_{SS}+6V

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

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

Operating Temperature ....−25°C to 70°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.
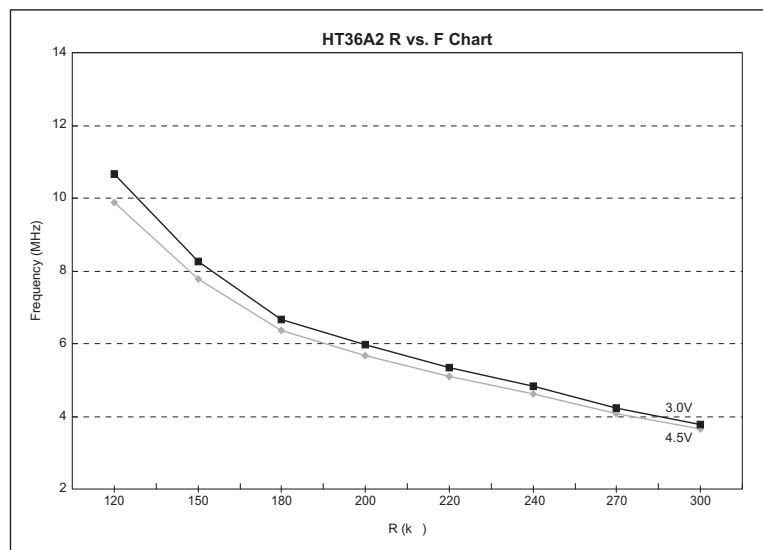
D.C. Characteristics

<div align="right">Ta=25°C</div>

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	3	5	V
I_{DD}	Operating Current	5V	No load, f_{OSC}=8MHz	—	8	16	mA
I_{STB}	Standby Current (WDT Disabled)	5V	No load System HALT	—	1	—	μA
I_{OL}	I/O Ports Sink Current	5V	V_{OL}=0.5V	9.7	16.2	—	mA
I_{OH}	I/O Ports Source Current	5V	V_{OH}=4.5V	−5.2	−8.7	—	mA
R_{PH}	Pull-High Resistance of I/O Ports	5V	V_{IL}=0V	11	22	44	kΩ
V_{IH1}	Input High Voltage for I/O Ports	5V	—	3.5	—	5	V
V_{IL1}	Input Low Voltage for I/O Ports	5V	—	0	—	1.5	V
V_{IH2}	Input High Voltage ($\overline{RES}$)	5V	—	—	4	—	V
V_{IL2}	Input Low Voltage ($\overline{RES}$)	5V	—	—	2.5	—	V

A.C. Characteristics

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
MCU interface							
f_{OSC}	System Frequency	5V	8MHz crystal	—	8	—	MHz
f_{SYS}	System Clock	5V	—	4	—	8	MHz
t_{WDT}	Watchdog Time-Out Period (RC)	—	Without WDT prescaler	9	17	35	ms
t_{RES}	External Reset Low Pulse Width	—	—	1	—	—	μs

Characteristics Curves

R vs F Characteristics Curve

V vs F Characteristics Curve

HT36A2 V vs. F Chart (For 3.0V)

HT36A2 V vs. F Chart (For 4.5V)
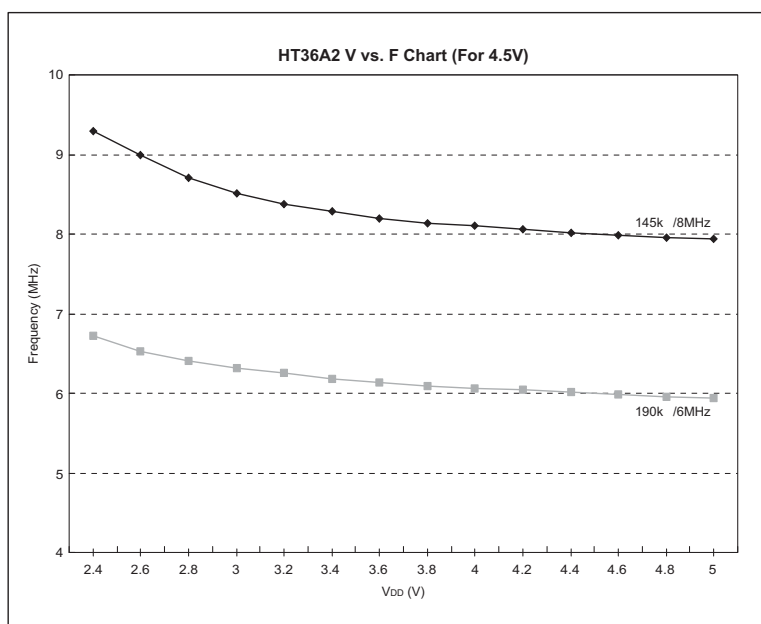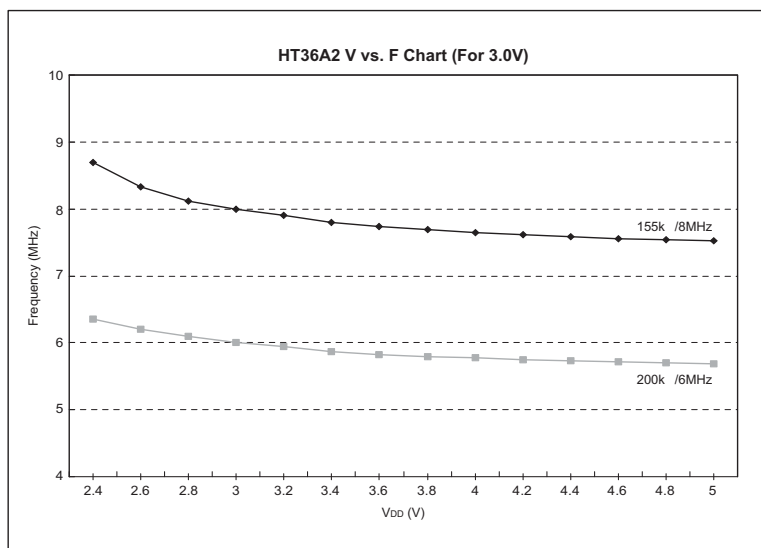
Function Description

Execution Flow

The system clock for the HT36A2 is derived from either a crystal or an RC oscillator. The oscillator frequency divided by 2 is the system clock for the MCU 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.

Program ROM

HT36A2 provides 16 address lines WA[15:0] 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 PF2~0 and a 13-bit program counter PC 12~0; and the
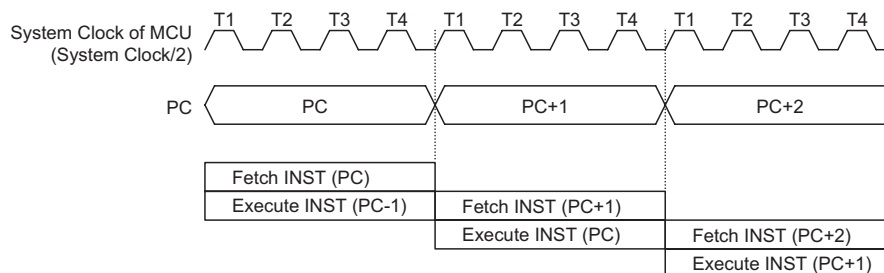
Execution flow

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
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
Skip	PC+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
Return From Subroutine	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

Program counter

Note: *12~*0: Bits of Program Counter

@7~@0: Bits of PCL

#12~#0: Bits of Instruction Code

S12~S0: Bits of Stack Register

@7~@0: Bits of PCL

other type is for wavetable code, which is addressed by the start address ST11~0. On the program type, WA15~0= PF2~0 $\times$ 2^{13}+ PC12~0. On the wave table ROM type, WA15~0=ST11~0 $\times$ 2^5.

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 008H
  This area is reserved for the Timer Counter 0 interrupt service program on each bank. If timer interrupt results from a timer 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 Counter 1 interrupt service program on each bank. If a timer interrupt results from a Timer 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.

- 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 TBLP. The TBLH is read only and cannot be restored. If the main routine and the ISR (Interrupt

Note: n ranges from 00 to 1F.
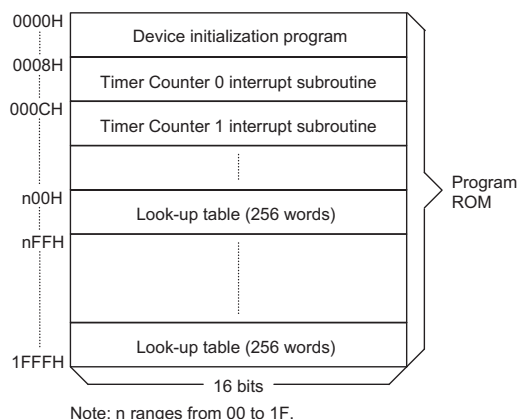
Program memory for each bank

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.

- Bank pointer
  The program memory is organized into 8 banks and each bank into 8192 × 16 bits of program ROM. PF[2~0] 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. 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 has to be cleared before setting to output mode.

Wavetable ROM

The ST[11~0] is used to defined the start address of each sample on the wavetable and read the waveform data from the location. HT36A2 provides 16 output address lines from WA[15~0], the ST[11~0] is used to lo-

Instruction(s)	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: Bits of table location   P12~P8: Bits of current Program Counter
          @7~@0: Bits of table pointer

cate the major 16 bits i.e. WA[15:5] and the undefined data from WA[4~0] is always set to 00000b. So the start address of each sample have to be located at a multiple of 32. 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 (PC) 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 SP will point to the top of the stack.
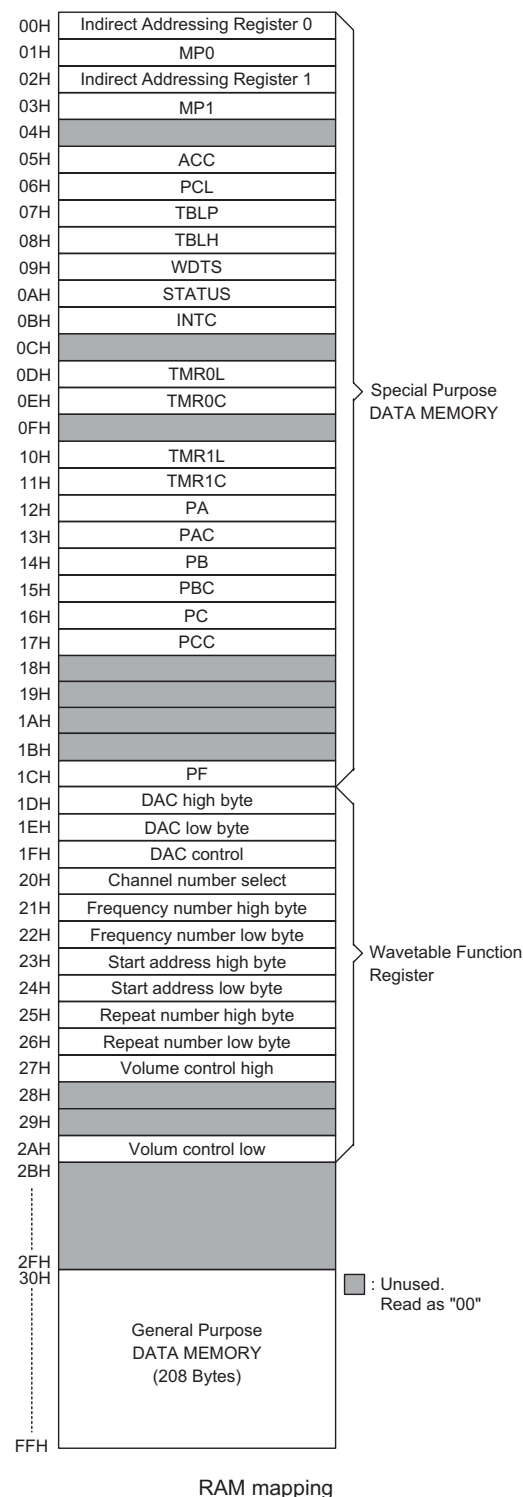
If the stack is full and a non-masked interrupt takes place, the interrupt request flag will be recorded but the acknowledgment will be inhibited. When the stack pointer is decremented (by RET or RETI), the interrupt will be serviced. This feature prevents stack overflow allowing the programmer to use 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 256 × 8 bits. The data memory is divided into three functional groups: special function registers, wavetable function register, and general purpose data memory (208×8). Most of them are read/write, but some are read only.

The special function registers include the Indirect Addressing register 0 (00H), the Memory Pointer register 0 (MP0;01H), the Indirect Addressing register 1 (02H), the Memory Pointer register 1 (MP1;03H), the Accumulator (ACC;05H), the Program Counter Lower-byte register (PCL;06H), the Table Pointer (TBLP;07H), the Table Higher-order byte register (TBLH;08H), the Watchdog Timer option Setting register (WDTS;09H), the Status register (STATUS;0AH), the Interrupt Control register (INTC;0BH), the Timer Counter 0 Lower-order byte register (TMR0L;0DH), the Timer Counter 0 Control register (TMR0C;0EH), the Timer Counter 1 Lower-order byte register (TMR1L;10H), the Timer Counter 1 Control register (TMR1C;11H), the I/O registers (PA;12H, PB;14H, PC;16H) and the I/O control registers (PAC;13H, PBC;15H, PCC;17H). The program ROM bank select (PF;1CH). The DAC High byte (DAH;1DH). The DAC low byte (DAL;1EH). The DAC control (DAC;1FH). The wavetable function registers is defined

between 20H~2AH. The remaining space before the 30H is reserved for future expanded usage and reading these locations will return the result 00H. The general purpose data memory, addressed from 30H to FFH, is used for data and control information under instruction command.

Address	Register	Group
00H	Indirect Addressing Register 0	
01H	MP0	
02H	Indirect Addressing Register 1	
03H	MP1	
04H		
05H	ACC	
06H	PCL	
07H	TBLP	
08H	TBLH	
09H	WDTS	
0AH	STATUS	
0BH	INTC	
0CH		
0DH	TMR0L	
0EH	TMR0C	Special Purpose DATA MEMORY
0FH		
10H	TMR1L	
11H	TMR1C	
12H	PA	
13H	PAC	
14H	PB	
15H	PBC	
16H	PC	
17H	PCC	
18H		
19H		
1AH		
1BH		
1CH	PF	
1DH	DAC high byte	
1EH	DAC low byte	
1FH	DAC control	
20H	Channel number select	
21H	Frequency number high byte	
22H	Frequency number low byte	
23H	Start address high byte	
24H	Start address low byte	Wavetable Function Register
25H	Repeat number high byte	
26H	Repeat number low byte	
27H	Volume control high	
28H		
29H		
2AH	Volum control low	
2BH		
2FH		
30H		
	General Purpose DATA MEMORY (208 Bytes)	
FFH		

: Unused. Read as "00"

RAM mapping

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 ....)

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 (PD) and Watchdog time-out flag (TO). It also records the status information and controls the operation sequence.

With the exception of the TO and PD 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 PD flags. In addition it should be noted that operations related to the status register may give different results from those intended. The TO and PD 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.

Interrupt

The HT36A2 provides two internal timer 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,

Labels	Bits	Function
C	0	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.
AC	1	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.
Z	2	Z is set if the result of an arithmetic or logical operation is zero; otherwise Z is cleared.
OV	3	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.
PD	4	PD is cleared by either a system power-up or executing the CLR WDT instruction. PD is set by executing the HALT instruction.
TO	5	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 register

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.

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 Counter 0 interrupt is initialized by setting the Timer Counter 0 interrupt request flag (T0F; bit 5 of INTC), caused by a Timer 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 Counter 1 interrupt is operated in the same manner as Timer Counter 0. The related interrupt control bits ET1I and T1F of the Timer 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 inter-

rupts 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 Counter 0 overflow	1	08H
Timer Counter 1 overflow	2	0CH

The Timer Counter 0/1 interrupt request flag (T0F/T1F), Enable Timer Counter 0/1 bit (ET0I/ET1I) and Enable Master Interrupt bit (EMI) constitute an interrupt control register (INTC) which is located at 0BH in the data memory. EMI, ET0I, ET1I are used to control the enabling/disabling of interrupts. These bits prevent the requested interrupt from being serviced. 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.

Oscillator Configuration

The HT36A2 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. 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, is available on OSC2

Register	Bit No.	Label	Function
INTC (0BH)	0	EMI	Controls the Master (Global) interrupt (1=enabled; 0=disabled)
	1	—	Unused bit, read as ″0″
	2	ET0I	Controls the Timer Counter 0 interrupt (1=enabled; 0=disabled)
	3	ET1I	Controls the Timer Counter 1 interrupt (1=enabled; 0=disabled)
	4	—	Unused bit, read as ″0″
	5	T0F	Internal Timer Counter 0 request flag (1=active; 0=inactive)
	6	T1F	Internal Timer Counter 1 request flag (1=active; 0=inactive)
	7	—	Unused bit, read as ″0″

INTC register

Crystal Oscillator   RC Oscillator
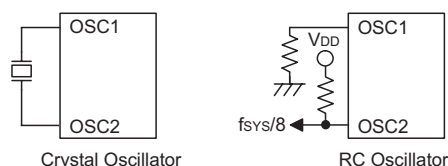
System oscillator

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 $78\mu s$. The WDT oscillator can be disabled by mask option to conserve power.

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 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 a period of $78\mu s$ 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 (bit 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 PC and SP are reset to zero. To clear the WDT contents (including the WDT prescaler), 3 methods are implemented; external reset (a low level to $\overline{\text{RES}}$), 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 because of time-out.
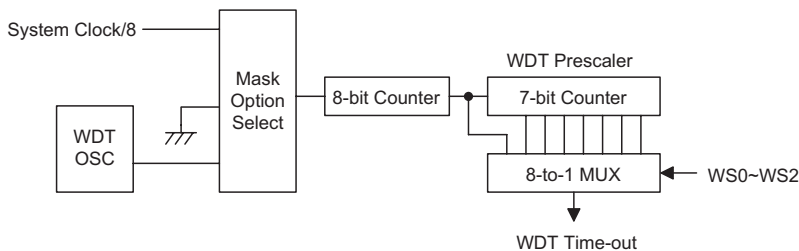
Watchdog timer

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 PD 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 PD flags, the cause for a chip reset can be determined. The PD 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 the WDT time-out occurs, and causes a wake-up that only resets the PC and SP, 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 the 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 execute 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.
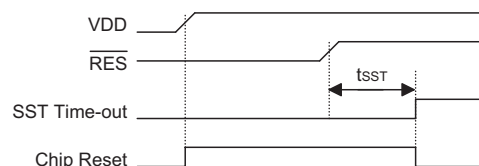
Reset

There are 3 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 reset" that just resets the PC and SP, leaving the other circuits to maintain their state. Some registers remain unchanged during any other reset conditions. Most registers are reset to the "initial condition" when the reset conditions are met. By examining the PD and TO flags, the program can distinguish between different "chip resets".

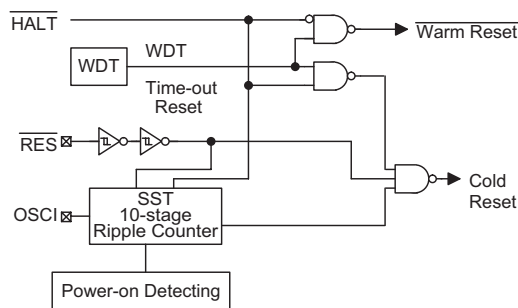| TO	PD	RESET Conditions
0	0	$\overline{RES}$ reset during power-up
u	u	$\overline{RES}$ reset during normal operation
0	1	$\overline{RES}$ wake-up HALT
1	u	WDT time-out during normal operation
1	1	WDT wake-up HALT

Note: "u" stands for "unchanged"

Reset timing chart
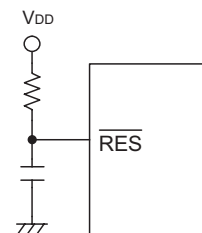
Reset circuit

Reset configuration

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 $\overline{RES}$ 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 Counter (0/1)	Off
Input/output ports	Input mode
SP	Points to the top of stack

The registers status is summarized in the following table:

Register	Reset (Power On)	WDT Time-out (Normal Operation)	$\overline{RES}$ Reset (Normal Operation)	$\overline{RES}$ 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
TMR0L	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR0C	00-0 1000	00-0 1000	00-0 1000	00-0 1000	uu-u 1uuu
TMR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR1C	00-0 1000	00-0 1000	00-0 1000	00-0 1000	uu-u 1uuu
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
PC	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu
PCC	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu
PF	---- -000	---- -000	---- -000	---- -000	---- -uuu
CHAN	00-- -000	uu-- -uuu	uu-- -uuu	uu-- -uuu	uu-- -uuu
FreqNH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
FreqNL	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
AddrH	---- xxxx	---- uuuu	---- uuuu	---- uuuu	---- uuuu
AddrL	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
ReH	---- --xx	---- --uu	---- --uu	---- --uu	---- --uu
ReL	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
VolH	---- xxxx	---- uuuu	---- uuuu	---- uuuu	---- uuuu
VolL	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
"−" stands for unused

Timer 0/1

Timer 0 is an 8-bit counter, and its clock source comes from the system clock divided by an 8-stage prescaler. There are two registers related to Timer 0; TMR0L(0DH) and TMR0C(0EH). One physical registers are mapped to TMR0L location; writing TMR0L makes the starting value be placed in the Timer 0 preload register and reading the TMR0 gets the contents of the Timer 0 counter. The TMR0C is a control register, which defines the division ration of the prescaler and counting enable or disable.

Writing data to B2, B1 and B0 (bits 2, 1, 0 of TMR0C) can yield various clock sources.

One the Timer 0 starts counting, it will count from the current contents in the counter to FFH. Once an overflow occurs, the counter is reloaded from a preload register, and generates an interrupt request flag (T0F; bit 2 of INTCH). To enable the counting operation, the timer On bit (TON; bit 4 of TMR0C) should be set to ″1″. For proper operation, bit 7 of TMR0C should be set to ″1″ and bit 3, bit 6 should be set to ″0″.

There are two registers related to the Timer Counter1; TMR1L(10H), TMR1C(11H). The Timer Counter 1 operates in the same manner as Timer Counter 0.

TMR0C/TMR1C			T0F
B2	B1	B0	
0	0	0	SYS CLK/16
0	0	1	SYS CLK/32
0	1	0	SYS CLK/64
0	1	1	SYS CLK/128
1	0	0	SYS CLK/256
1	0	1	SYS CLK/512
1	1	0	SYS CLK/1024
1	1	1	SYS CLK/2048

TMR0C Bit 4 to enable/disable timer counting (1=enable; 0=disable)

TMR0C Bit 3, always write ″0″.
TMR0C Bit 5, always write ″0″.
TMR0C Bit 6, always write ″0″.
TMR0C Bit 7, always write ″1″.

Input/Output Ports

There are 20 bidirectional input/output lines labeled from PA to PC0~3, which are mapped to the data memory of [12H], [14H], [16H] 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 or 16H). 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, PCC0~3) 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, 15H and 17H.

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, 14H or 16H) 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 0/1

Input/output ports
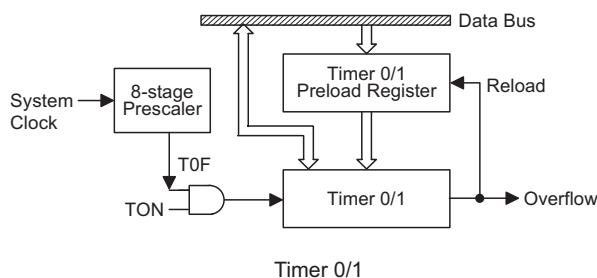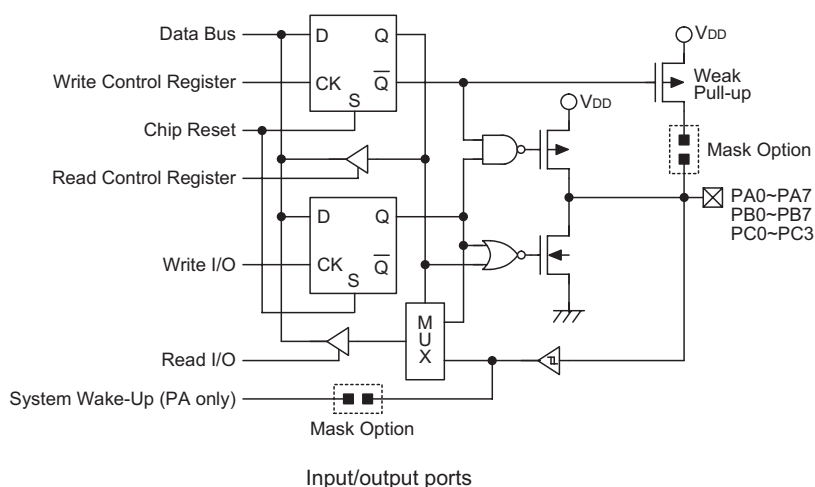
8 Channel Wavetable Synthesizer

Memory Map Register Table (1DH~FFH)

Name	Function	D7	D6	D5	D4	D3	D2	D1	D0
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 DAON=0: DAC OFF (default) SELW=1: DAC data from wavetable SELW=0: DAC data from MCU	—	—	—	—	—	—	DAON	SELW
20H	Channel number selection	VM	FR				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					ST11	ST10	ST9	ST8
24H	Low byte start address	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
25H	Wave bit select, High byte repeat number	WBS						RE9	RE8
26H	Low byte repeat number	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0
27H	Envelope control, Volume control	A_R				ENV1	ENV0	VR9	VR8
28H~29H	Unused								
2AH	Volume control	VR7	VR6	VR5	VR4	VR3	VR2	VR1	VR0
2BH~2FH	Unused								
30H~FFH	Data memory (RAM)	General purpose data memory (same as 8-Bit MCU)							

Note: "—" No function, read only, read as "0".

Unused: No function, read only, read as "0".

- CH[2~0] channel number selection

  The HT36A2 has a built-in 8 output channels and CH[2~0] 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 BL[3~0] and FR[11~0] are used to define the output speed of the PCM file, i.e. it can be used to generate the tone scale. When the FR[11:0] is 800H and BL[3:0] is 6H, each sample data of the PCM code will be sent out sequentially.

  When the f_{OSC} is 6.4MHz, the formula of a tone frequency is:

  $$f_{OUT} = f_{RECORD} \times \frac{25kHz}{SR} \times \frac{FR[11\sim0]}{2^{(17-BL[3\sim0])}}$$

  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}=98Hz.

  Can be obtained by using the formula:

  $$98Hz = 261Hz \times \frac{25kHz}{11025Hz} \times \frac{FR[11\sim0]}{2^{(17-BL[3\sim0])}}$$

  A pair of the values FR[11~0] and BL[3~0] can be determined when the f_{OSC} is 6.4MHz.
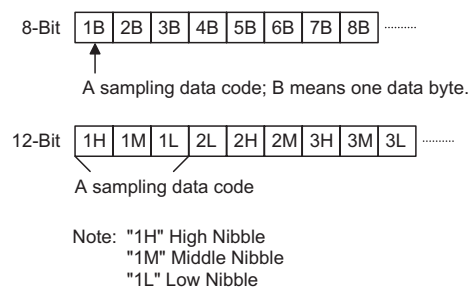
- Start address definition

  The HT36A2 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 ST[11~0] is used to define the start address of each PCM code and reads the waveform data from this location. The HT36A2 provides 16 input data lines from WA[15~0], the ST[11~0] is used to locate the ma-

jor 12 bits i.e. WA[15~5] and the undefined data from WA[4~0] is always set as 00000b. In other words, the WA[15~0]=ST[11~0]×2^5. 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 HT36A2 accepts two waveform formats to ensure a more economical data space. WBS is used to define the sample format of each PCM code.

  A sampling data code; B means one data byte.

  A sampling data code

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

  Waveform format

  ◆ 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.

- 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 become close.

  The RE[9~0] is used to calculate the repeat address of the PCM code. The process for setting the RE[9~0] is to write the 2′s complement of the repeat length to RE[9~0], with the highest carry ignored. The HT36A2 will get the repeat address by adding the RE[9~0] to the address of the end code, then jump to the address to repeat this range.

- Volume control

  The HT36A2 provides the volume control independently. The volume are controlled by VR[9~0] 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 HT36A2 provides a function to easily program the envelope by setting the data of ENV[1~0] 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 ENV[1~0] 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.

- The PCM code definition

  The HT36A2 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.

Mask Option

No.	Mask Option	Function
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, PB, PC0~3 input
5	OSC mode	Crystal or Resistor type

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

Envelope type definition

Application Circuit

Note: R1 > R2
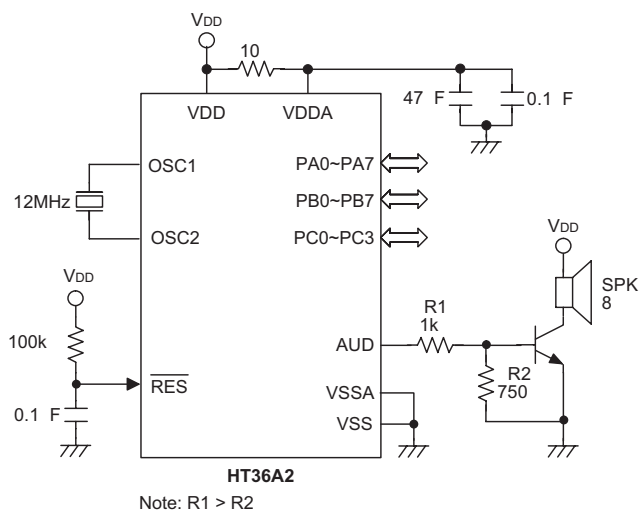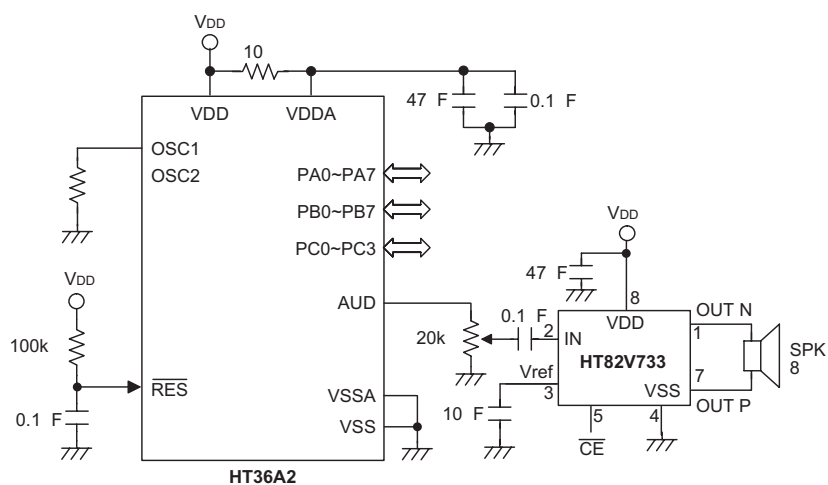
Package Information

28-pin SOP (300mil) Outline Dimensions
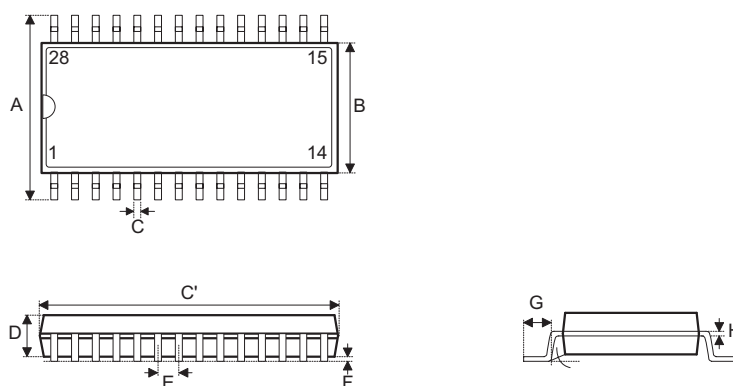
Symbol	Dimensions in mil		
	Min.	**Nom.**	**Max.**
A	394	—	419
B	290	—	300
C	14	—	20
C′	697	—	713
D	92	—	104
E	—	50	—
F	4	—	—
G	32	—	38
H	4	—	12
α	0°	—	10°

48-pin SSOP (300mil) Outline Dimensions
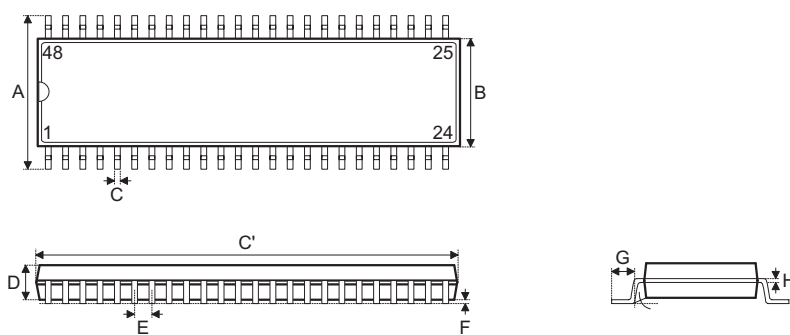
Symbol	Dimensions in mil		
	Min.	**Nom.**	**Max.**
A	395	—	420
B	291	—	299
C	8	—	12
C′	613	—	637
D	85	—	99
E	—	25	—
F	4	—	10
G	25	—	35
H	4	—	12
α	0°	—	8°

Product Tape and Reel Specifications

Reel Dimensions
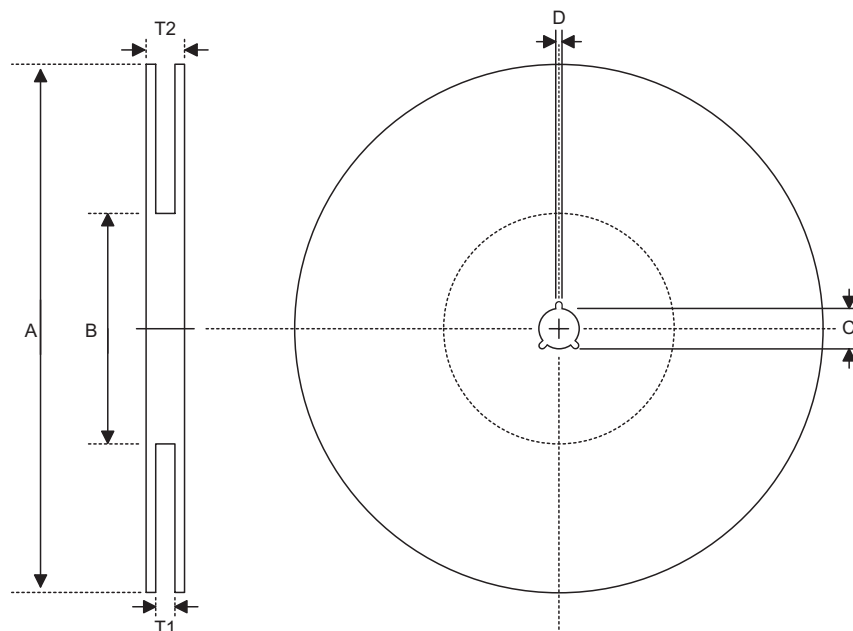
SOP 28W (300mil)

Symbol	Description	Dimensions in mm
A	Reel Outer Diameter	330±1.0
B	Reel Inner Diameter	62±1.5
C	Spindle Hole Diameter	13.0+0.5 −0.2
D	Key Slit Width	2.0±0.5
T1	Space Between Flange	24.8+0.3 −0.2
T2	Reel Thickness	30.2±0.2

SSOP 48W

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

Carrier Tape Dimensions
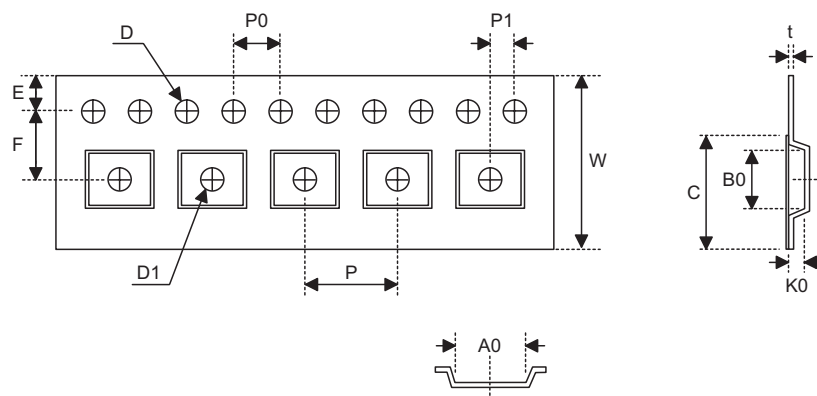
SOP 28W (300mil)

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	24.0±0.3
P	Cavity Pitch	12.0±0.1
E	Perforation Position	1.75±0.1
F	Cavity to Perforation (Width Direction)	11.5±0.1
D	Perforation Diameter	1.5+0.1
D1	Cavity Hole Diameter	1.5+0.25
P0	Perforation Pitch	4.0±0.1
P1	Cavity to Perforation (Length Direction)	2.0±0.1
A0	Cavity Length	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
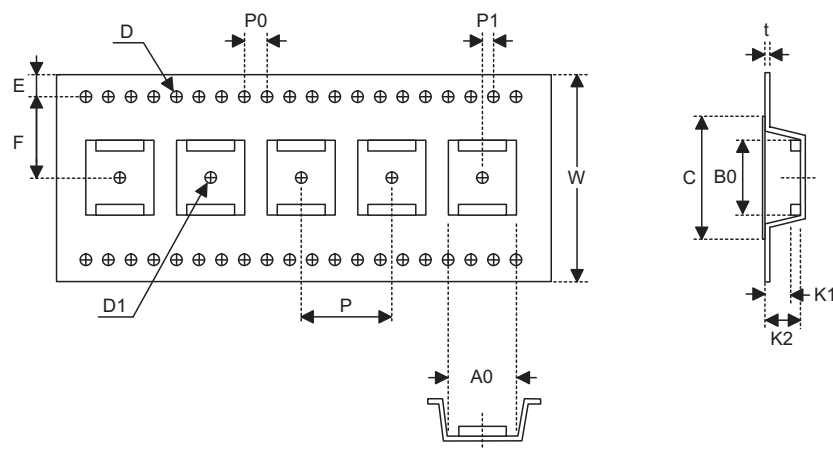
SSOP 48W

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