HT44300/443A0 SPECIFICATION

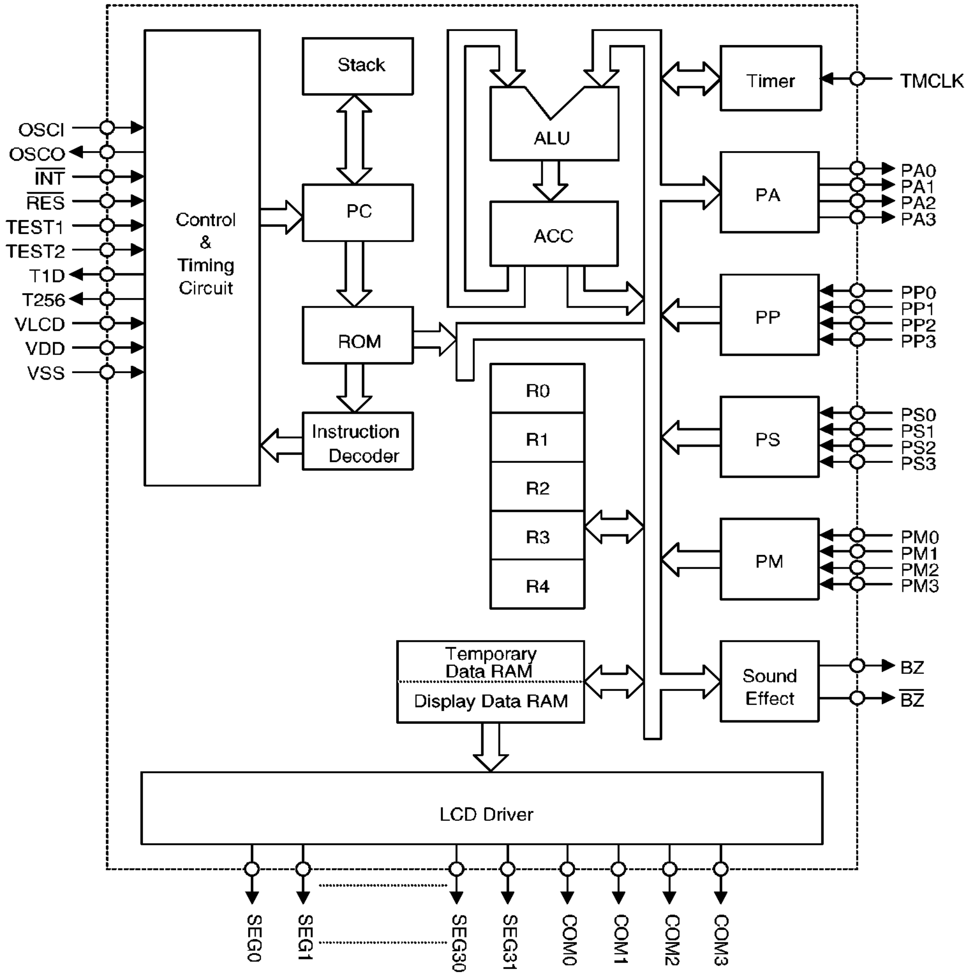
Features

- Operating voltage: 2.4~5.0V
- 12 input lines
- 4 output lines
- Halt feature reduces power consumption
- Up to 1µs instruction cycle with 4MHz system clock at VDD=5V
- All instructions in 1 or 2 machine cycles
- 4K × 8 program ROM
- Data memory RAM size
  160 × 4 bits for HT44300
  240 × 4 bits for HT443A0
- LCD display product driver
- 8-bit table read instruction
- 5 working registers
- External interrupt
- Internal timer overflow
- One level subroutine nesting
- RC oscillator
- 8 bit timer
- Sound effect circuit

General Description

The HT44300/443A0 are two processors from Holtek's 4-bit stand alone single chip microcontroller range specifically designed for LCD product applications. The two devices differ in the number of LCD segments they can drive and in the amount of internal RAM. Otherwise the two processers are identical. Both devices are ideally suited for multiple LCD low power applications among which are calculators, scales, and hand held LCD products.

Block Diagram HT44300



Note:

ACC: Accumulator	PA: Output port
PC: Program counter	PS,PM,PP: Input ports
R0~R4: Working registers	

Pad Description HT44300

Pad No.	Pad Name	I/O	Mask Option	Function
1,2	BZ,$\overline{\text{BZ}}$	O	Note 1	Sound effect outputs
3	VLCD	I	—	(+) LCD bias power supply
4 22 9 10	T256 T1D TEST1 TEST2	O O I I	—	For test mode only TEST1 and TEST2 are left open when the HT44300 is in normal operation (with an internal pull high resistor).
5~8	COM3~COM0	O	Note 2	Output for LCD panel common plate
11~14	PM3~PM0	I	Pull-high or none. Note 3	4-bit port for input only
15~18	PS3~PS0	I	Pull-high or none. Note 3	4-bit port for input only
19	VSS	I	—	Negative power supply, GND
20 21	OSCI OSCO	I O	—	OSCI,OSCO are connected to an external resistor for an internal system clock
23~26	PA3~PA0	O	CMOS or NMOS open drain	4-bit latch port for output only
27~30	PP0~PP3	I	Pull-high or none. Note 3	4-bit port for input only
31	$\overline{\text{INT}}$	I	—	External interrupt input with a pull high resistor Active on edge–triggered high to low transition
32	$\overline{\text{RES}}$	I	—	Input to reset an internal LSI Reset is active on logical low level
33	TMCLK	I	Pull-high or none. Note 4	Input for TIMER clock TIMER can be clocked by an external clock or an internal frequency source.
34~65	SEG0~SEG31	O	—	LCD driver outputs for LCD panel segment
66	VDD	I	—	Positive power supply

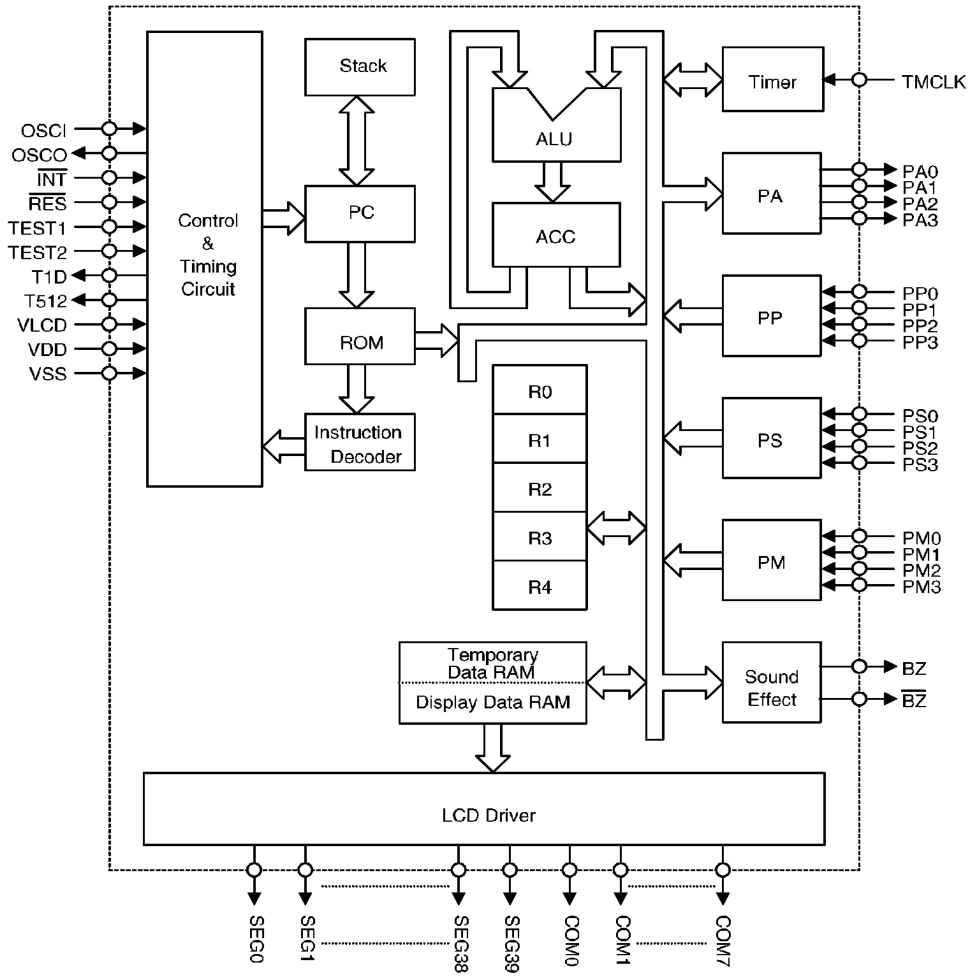
Note 1: The system clock provides 6 different sources selectable by mask option to drive the sound effect clock. If the Holtek sound library is used only 128K and 64K are acceptable.

Note 2: Either (1/4 duty;1/3 bias) or (1/3 duty:1/3 bias) should be specified by mask option.

Note 3: Each bit of ports PM, PS and PP can be a trigger source of the HALT interrupt, selectable by mask option.

Note 4: 13 internal clock sources can be selectable by mask option to drive TMCLK. Note that TMCLK should not be connected to a pull high resistor if an internal source is used.

Block Diagram HT443A0



Note:

ACC: Accumulator	PA: Output port
PC: Program counter	PS,PM,PP: Input ports
R0~R4: Working registers	

Pad Description HT443A0

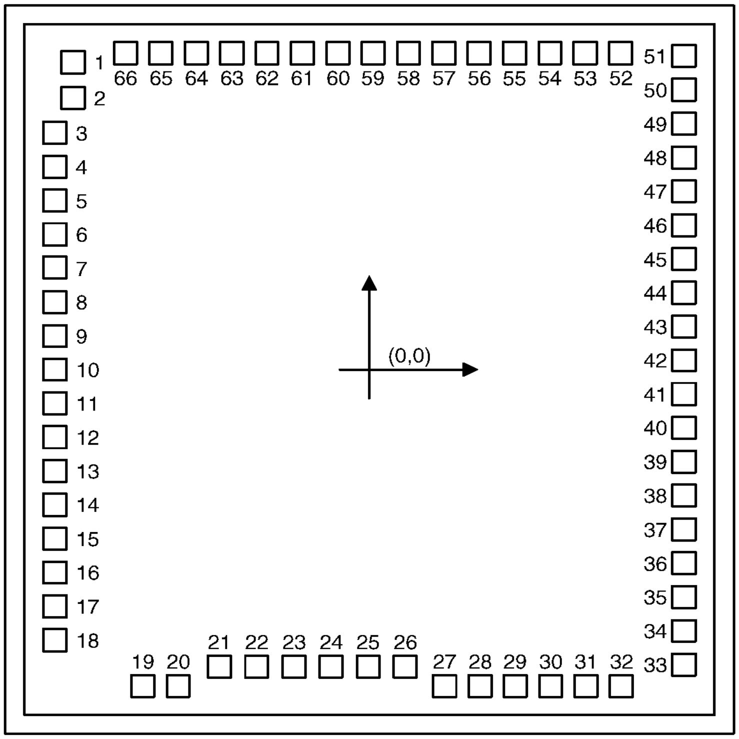
Pad No.	Pad Name	I/O	Mask Option	Function
1	VLCD	I	—	(+) LCD bias power supply
2,3	BZ,$\overline{\text{BZ}}$	O	Note 1	Sound effect outputs
4	VDD	I	—	Positive power supply
5 6	OSCI OSCO	I O	—	OSCI,OSCO connected to an external resistor for an internal system clock
7 27 16 17	T512 T1D TEST1 TEST2	O O I I	—	For test mode only TEST1 and TEST2 are left open when the HT443A0 is in normal operation (with an internal pull high resistor).
8~15	COM7~COM0	O	—	Outputs for LCD panel common plate
18~21	PM3~PM0	I	Pull-high or none. Note 2	4-bit input port
22~25	PS3~PS0	I	Pull-high or none. Note 2	4-bit input port
26	VSS	I	—	Negative power supply, GND
28~31	PA3~PA0	O	CMOS or NMOS open drain	4-bit output latch port
32~35	PP0~PP3	I	Pull-high or none. Note 2	4-bit input port
36	$\overline{\text{INT}}$	I	—	External interrupt, with a pull high resistor, edge triggered on high to low transition
37	$\overline{\text{RES}}$	I	—	Input to reset an internal LSI Reset is active on logical low level
38	TMCLK	I	Pull-high or none. Note 3	Input for TIMER clock TIMER can be clocked by an external clock or by an internal frequency source.
39~78	SEG39~SEG0	O	—	LCD driver outputs for LCD panel segments

Note 1: The system clock provides 6 different sources selectable by mask option to drive the sound effect clock. If the Holtek sound library is used only 128K and 64K are acceptable.

Note 2: Each bit of ports PM, PS and PP can be a trigger source of the HALT interrupt, selectable by mask option.

Note 3: 13 internal clock sources can be selectable by mask option to drive TMCLK. Note that TMCLK should not be connected to a pull high resistor if an internal source is used.

Pad Position HT44300



Chip size: 3560 × 3840 μm

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

Pad Coordinates HT44300

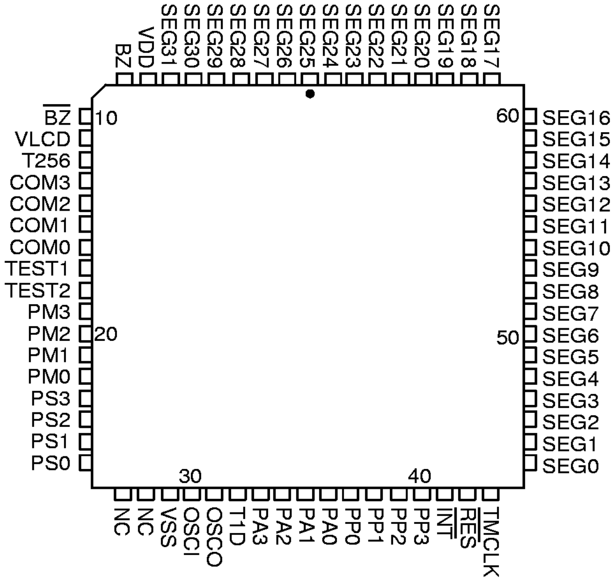
Unit: µm

Pad No.	Pad Name	X	Y	Pad No.	Pad Name	X	Y
1	BZ	–1549.21	1681.46	25	PA1	–6.98	–1625.27
2	$\overline{BZ}$	–1549.25	1486.58	26	PA0	187.91	–1625.14
3*	VLCD	–1645.32	1294.63	27	PP0	393.56	–1731.26
4*	T256	–1645.28	1109.61	28	PP1	578.69	–1731.31
5	COM3	–1645.27	924.51	29	PP2	763.72	–1731.26
6	COM2	–1645.14	739.83	30	PP3	948.90	–1731.31
7	COM1	–1645.35	554.65	31*	$\overline{INT}$	1133.65	–1731.26
8	COM0	–1645.23	369.94	32*	$\overline{RES}$	1318.84	–1731.00
9*	TEST1	–1645.52	184.69	33	TMCLK	1645.16	–1614.12
10*	TEST2	–1645.40	–0.39	34*	SEG0	1645.15	–1429.39
11	PM3	–1645.27	–185.42	35*	SEG1	1645.30	–1244.10
12	PM2	–1645.18	–370.56	36*	SEG2	1645.15	–1059.12
13	PM1	–1645.31	–555.24	37*	SEG3	1645.30	–874.14
14	PM0	–1645.19	–740.39	38*	SEG4	1645.24	–689.15
15	PS3	–1645.31	–925.35	39*	SEG5	1645.29	–504.25
16	PS2	–1645.19	–1110.24	40*	SEG6	1645.52	–319.19
17	PS1	–1645.00	–1295.21	41*	SEG7	1645.38	–134.13
18	PS0	–1645.19	–1480.00	42*	SEG8	1645.15	50.70
19*	VSS	–1185.92	–1731.13	43*	SEG9	1645.38	235.84
20*	OSCI	–1000.84	–1731.35	44*	SEG10	1645.15	420.74
21*	OSCO	–784.95	–1625.36	45*	SEG11	1645.20	605.88
22*	T1D	–590.39	–1625.32	46*	SEG12	1645.33	790.56
23	PA3	–395.78	–1625.27	47*	SEG13	1645.20	975.54
24	PA2	–201.54	–1625.14	48*	SEG14	1645.42	1160.75

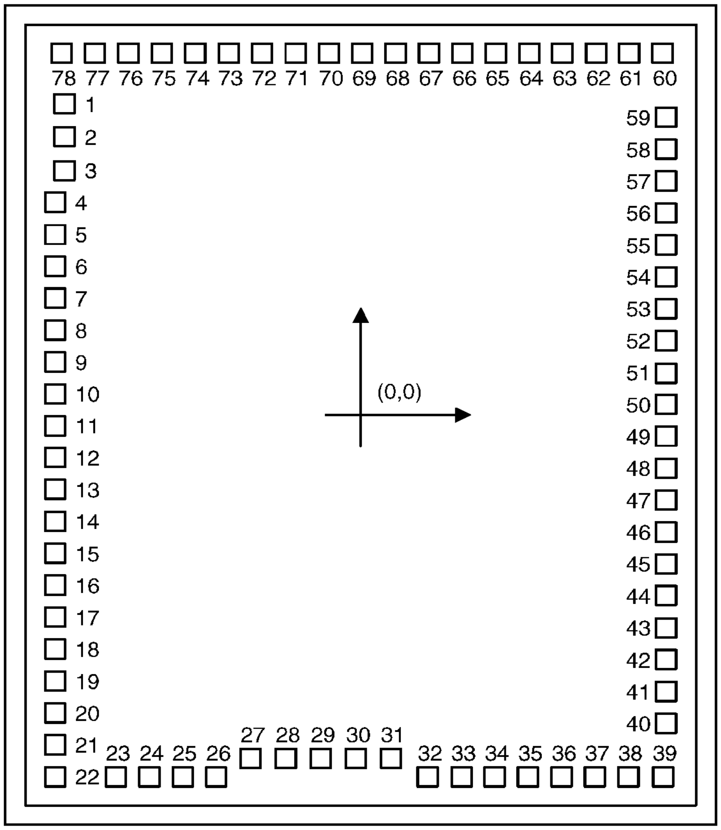
Pad No.	Pad Name	X	Y	Pad No.	Pad Name	X	Y
49*	SEG15	1645.30	1345.89	58*	SEG24	204.33	1731.28
50*	SEG16	1645.06	1530.80	59*	SEG25	19.14	1731.15
51*	SEG17	1645.11	1715.50	60	SEG26	–165.75	1731.36
52*	SEG18	1314.11	1731.28	61	SEG27	–350.89	1731.07
53*	SEG19	1128.97	1730.98	62	SEG28	–535.93	1731.19
54*	SEG20	944.08	1731.28	63	SEG29	–720.62	1731.40
55*	SEG21	759.14	1730.81	64	SEG30	–905.81	1731.19
56*	SEG22	574.30	1731.36	65	SEG31	–1090.70	1731.32
57*	SEG23	389.27	1731.15	66*	VDD	–1275.89	1731.11

* These pins must be bonded out for functional testing.

68 Pin PLCC Package HT44300



Pad Position HT443A0



Chip size: 3660 x 4590 μm

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

Pad Coordinates HT443A0

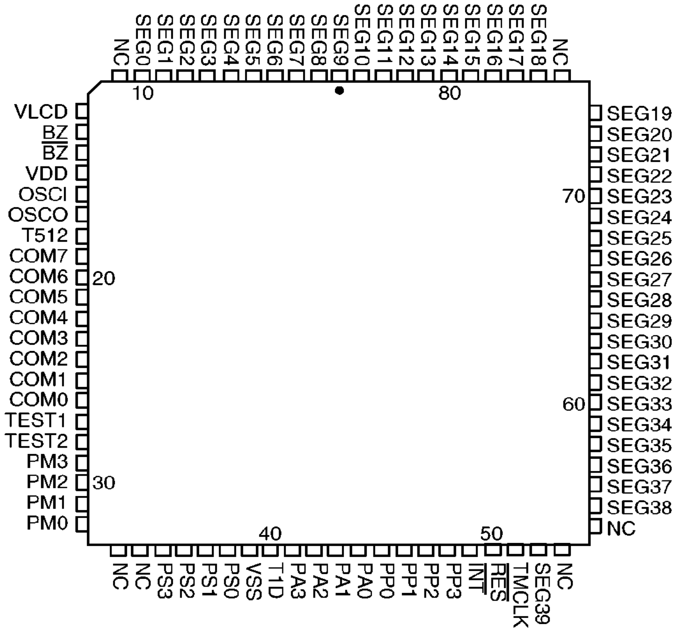
Unit: µm

Pad No.	Pad Name	X	Y	Pad No.	Pad Name	X	Y
1∗	VLCD	–1641.25	1802.25	25	PS0	–985.75	–2097.75
2	BZ	–1641.25	1612.25	26∗	VSS	–800.75	–2097.75
3	$\overline{\text{BZ}}$	–1641.25	1417.25	27∗	T1D	–610.75	–1991.75
4∗	VDD	–1691.25	1232.25	28	PA3	–416.25	–1991.75
5∗	OSCI	–1691.25	1047.25	29	PA2	–221.75	–1991.75
6∗	OSCO	–1691.25	862.25	30	PA1	–27.25	–1991.75
7∗	T512	–1691.25	677.26	31	PA0	167.25	–1991.75
8	COM7	–1691.25	492.25	32	PP0	373.25	–2097.75
9	COM6	–1691.25	307.25	33	PP1	558.25	–2097.75
10	COM5	–1691.25	122.25	34	PP2	743.25	–2097.75
11	COM4	–1691.25	–62.75	35	PP3	928.25	–2097.75
12	COM3	–1691.25	–247.75	36∗	$\overline{\text{INT}}$	1113.25	–2097.75
13	COM2	–1691.25	–432.75	37∗	$\overline{\text{RES}}$	1298.25	–2097.75
14	COM1	–1691.25	–617.75	38	TMCLK	1483.25	–2097.75
15	COM0	–1691.25	–802.75	39∗	SEG39	1675.25	–2097.75
16∗	TEST1	–1691.25	–987.75	40∗	SEG38	1691.25	–1788.75
17∗	TEST2	–1691.25	–1172.75	41∗	SEG37	1691.25	–1603.75
18	PM3	–1691.25	–1357.75	42∗	SEG36	1691.25	–1418.75
19	PM2	–1691.25	–1542.75	43∗	SEG35	1691.25	–1233.75
20	PM1	–1691.25	–1727.75	44∗	SEG34	1691.25	–1048.75
21	PM0	–1691.25	–1912.75	45∗	SEG33	1691.25	–863.75
22	PS3	–1691.25	–2097.75	46∗	SEG32	1691.25	–678.75
23	PS2	–1355.75	–2097.75	47∗	SEG31	1691.25	–493.75
24	PS1	–1170.75	–2097.75	48∗	SEG30	1691.25	–308.75

Pad No.	Pad Name	X	Y	Pad No.	Pad Name	X	Y
49*	SEG29	1691.25	–123.75	64*	SEG14	933.75	2097.75
50*	SEG28	1691.25	61.25	65	SEG13	748.75	2097.75
51*	SEG27	1691.25	246.25	66	SEG12	563.75	2097.75
52*	SEG26	1691.25	431.25	67	SEG11	378.75	2097.75
53*	SEG25	1691.25	616.25	68	SEG10	193.75	2097.75
54*	SEG24	1691.25	801.25	69	SEG9	8.75	2097.75
55*	SEG23	1691.25	986.25	70	SEG8	–176.25	2097.75
56*	SEG22	1691.25	1171.25	71	SEG7	–361.25	2097.75
57*	SEG21	1691.25	1356.25	72	SEG6	–546.25	2097.75
58*	SEG20	1691.25	1541.25	73	SEG5	–731.25	2097.75
59*	SEG19	1691.25	1726.25	74	SEG4	–916.25	2097.75
60*	SEG18	1673.75	2097.75	75	SEG3	–1101.25	2097.75
61*	SEG17	1488.75	2097.75	76	SEG2	–1286.25	2097.75
62*	SEG16	1303.75	2097.75	77	SEG1	–1471.25	2097.75
63*	SEG15	1118.75	2097.75	78	SEG0	–1656.25	2097.75

* These pins must be bonded out for functional testing.

84 Pin PLCC Package HT443A0



Absolute Maximum Ratings HT44300/443A0

Parameter	Symbol	Minimum	Maximum	Unit
Supply Voltage	V_{DD}	–0.3	6	V
Input Voltage	V_I	V_{SS}–0.3	V_{DD}+0.3	V
Storage Temperature	T_{STG}	–50	125	°C
Operating Temperature	T_{OP}	0	70	°C

D.C. Characteristics

(Ta=25°C)

Symbol	Parameter	Test Condition		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
V_{DD}	Operating voltage	—	—	2.4	—	5.0	V
I_{DD}	Operating current	3V	No load, f_{SYS}=500KHz	—	500	—	μA
		5V		—	1000	—	μA
I_{STB}	Stand-by current	3V	No load, HALT mode	—	—	1	μA
		5V		—	—	2	μA
V_{IL}	Input low voltage	3V	—	0	—	0.9	V
		5V	—	0	—	1.5	V
V_{IH}	Input high voltage	3V	—	2.1	—	3.0	V
		5V	—	3.5	—	5.0	V
I_{OL1}	Port A, BZ & $\overline{BZ}$ output sink current	3V	V_{DD}=3V, V_{OL}=0.3V	1.5	3.0	—	mA
		5V	V_{DD}=5V, V_{OL}=0.5V	3.0	8.0	—	mA
I_{OH1}	Port A, BZ & $\overline{BZ}$ output source current	3V	V_{DD}=3V, V_{OH}=2.7V	–0.6	–1.0	—	mA
		5V	V_{DD}=5V, V_{OH}=4.5V	–1.8	–2.5	—	mA
I_{OL2}	Segment output sink current	3V	V_{LCD}=3V, V_{OL}=0.3V	30	55	—	μA
		5V	V_{LCD}=5V, V_{OL}=0.5V	60	110	—	μA
I_{OH2}	Segment output source current	3V	V_{LCD}=3V, V_{OH}=2.7V	–20	–40	—	μA
		5V	V_{LCD}=5V, V_{OH}=4.5V	–40	–90	—	μA
R_{PH}	Pull–high resistance	3V	PS, PP, PM, $\overline{INT}$, $\overline{RES}$, TMCLK	30	—	300	KΩ
		5V		30	—	300	KΩ
V_{LCD}	LCD supply voltage	3V	—	2.5	3.0	3.5	V
		5V	—	4.5	5.0	5.5	V
I_{LCD}	LCD supply current	3V	V_{LCD}=V_{DD} all segments on	—	—	550	μA
		5V		—	—	1000	μA

A.C. Characteristics

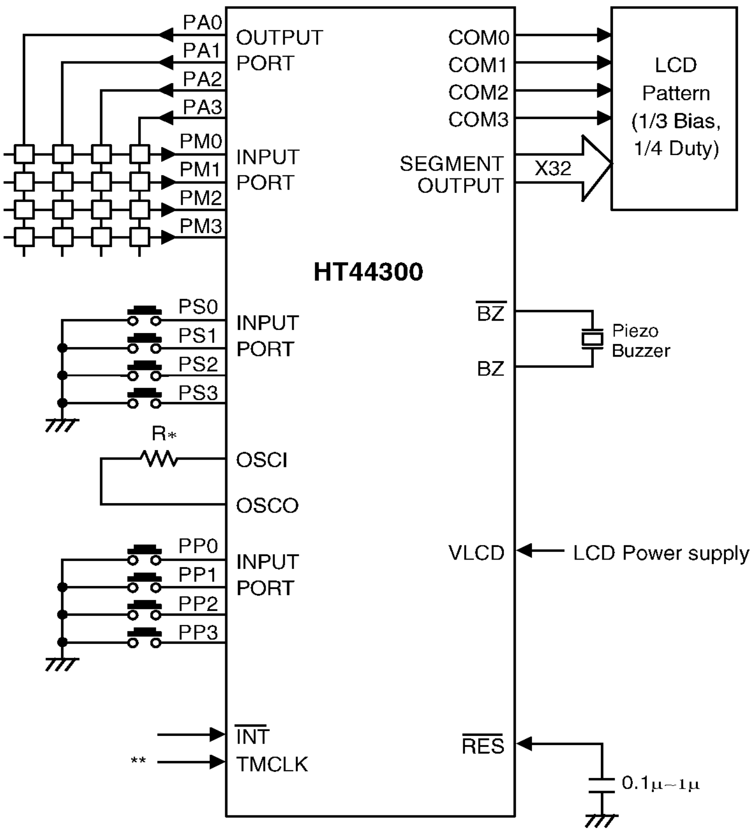
(Ta=25°C)

Symbol	Parameter	Test condition		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
f_{SYS}	System clock	3V	R:620K~36K	32	—	2000	KHz
		5V	R:620K~9K	32	—	4000	KHz
f_{LCD1}	HT44300 LCD clock	3V	—	—	256*	—	Hz
		5V		—	256*	—	Hz
f_{LCD2}	HT443A0 LCD clock	3V	—	—	512*	—	Hz
		5V	—	—	512*	—	Hz
t_{COM1}	HT44300 LCD common period	—	1/3 duty	—	$(1/f_{LCD1})\times3$	—	s
			1/4 duty	—	$(1/f_{LCD1})\times4$	—	s
t_{COM2}	HT443A0 LCD common period	—	1/8 duty	—	$(1/f_{LCD2})\times8$	—	s
t_{CY}	Cycle time	—	f_{SYS}=4.0MHz	—	1.0	—	μs
f_{TIMER}	Timer I/P frequency (TMCLK)	3V	—	0	—	2000	KHz
		5V		0	—	4000	KHz
t_{RES}	Reset pulse width	—	—	5	—	—	ms
t_{INT}	Interrupt pulse width	—	—	1	—	—	μs
f_{SOUND}	Sound effect clock	—	—	—	64 or 128 **	—	KHz

*: In general, f_{LCD} is selected and optimized by Holtek depending upon f_{SYS} and the operating voltage.

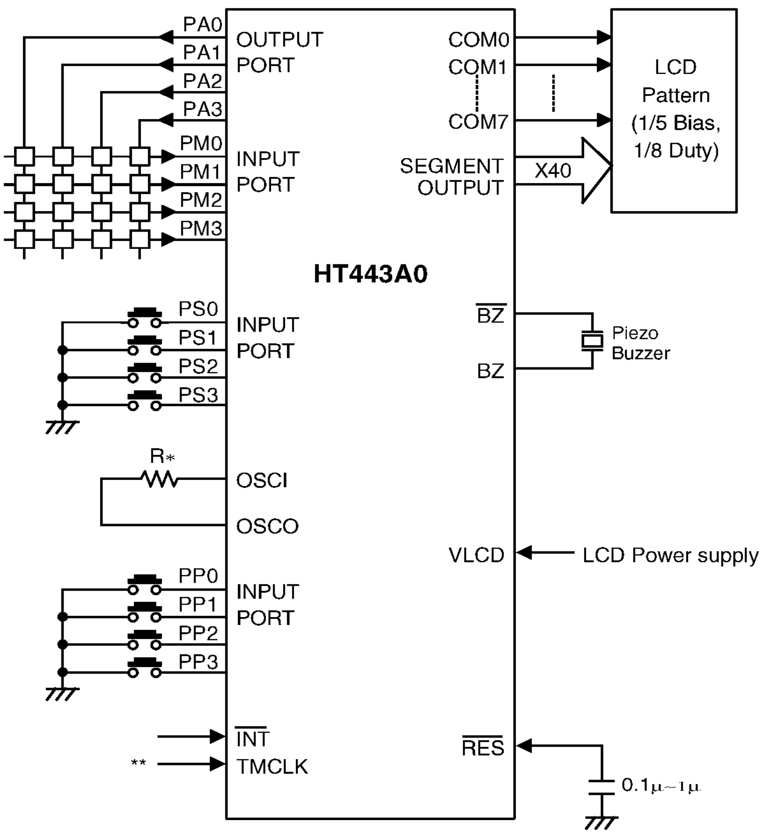
**: Only these two clocking signal frequencies are supported by the Holtek sound library.

Application Diagram HT44300



R*: Depends on the required system clock frequency. (R=36K~620K, at VDD=3V)
**: Timer clock may come from an external or internal frequency source.

Application Diagram HT443A0



R*: Depends on the required system clock frequency. (R=36K~620K, at VDD=3V)
**: Timer clock may come from an external or internal frequency source.

SYSTEM ARCHITECTURE

Program Counter - PC

This counter addresses the program ROM and is arranged as an 12-bit binary counter from PC0 to PC11 whose contents specify a maximum of 4096 addresses. The program counter counts with an increment of 1 or 2 with each execution of an instruction.

When executing the jump instruction (JMP, JNZ, JC, JTMR,...), a subroutine call, initial reset, internal interrupt, external interrupt or returning from a subroutine, the program counter is loaded with the corresponding instruction data as shown in the table.

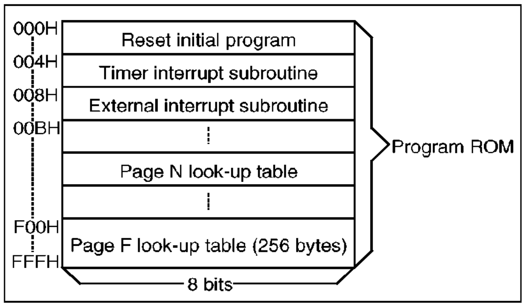
Note: P0~P11: Instruction code
@: PC11 keeps current value
S0~S11: Stack register bits

Program Memory - ROM

The program memory is the executable memory and is arranged in a 4096×8 bit format. The address is specified by the program counter (PC). Four special locations are reserved as described as follows.

Location 0:
Activating the processor $\overline{\text{RES}}$ pin causes the first instruction to be fetched from location 0.

Location 4:
Contains the timer interrupt resulting from a TIMER overflow. If the interrupts are enabled it causes the program to jump to this subroutine.


000H
004H
008H
00BH
F00H
FFFH
Reset initial program
Timer interrupt subroutine
External interrupt subroutine
Page N look-up table
Page F look-up table (256 bytes)
8 bits
Program ROM

Program Memory

Location 8:
Activating the $\overline{\text{INT}}$ input pin of the processor with the interrupts enabled causes the program to jump to this location.

Locations n00H to nFFH:
These are the 256 bytes of each page in program memory. This area from n00H to nFFH and F00H to FFFH can be used as a look–up table. Instructions such as READ R4A, READ MR0A, READF R4A, READF MR0A can read the table and transfer the contents of the table to ACC and R4 or to ACC and a data memory address specified by the register pair R1,R0. However as R1,R0 can only store 8 bits, these instructions cannot fully specify the full 12 bit program memory address. For this reason a jump instruction should be first used to place the program counter in the right page. The above instructions can then be used to read the look up table data.

Mode	Program Counter											
	PC11	**PC10**	**PC9**	**PC8**	**PC7**	**PC6**	**PC5**	**PC4**	**PC3**	**PC2**	**PC1**	**PC0**
Initial reset	0	0	0	0	0	0	0	0	0	0	0	0
Internal interrupt	0	0	0	0	0	0	0	0	0	1	0	0
External interrupt	0	0	0	0	0	0	0	0	1	0	0	0
Jump, call instruction	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
Conditional branch	@	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
Return from subroutine	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

Program Memory

Note that the page number n must be greater than zero as some locations in page 0 are reserved for specific usage as mentioned. This area may function as normal program memory as required.

The program memory mapping is shown in the diagram.

In the execution of an instruction the program counter is added before the execution phase, so careful manipulation of READ MR0A and READ R4A is needed in the page margin.

Stack Register

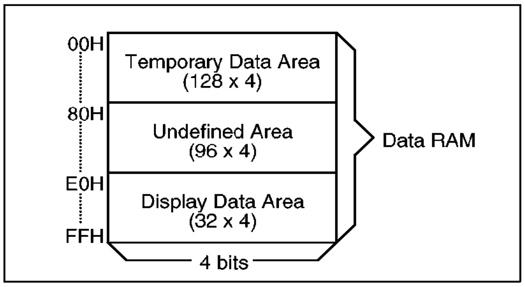
The stack register is a group of registers used to save the contents of the program counter (PC) and is arranged in 13 bits×1 level. One bit is used to store the carry flag. An interrupt will force the contents of the PC and the carry flag onto the stack register. A subroutine call will also cause the PC contents to be pushed onto the stack; however the carry flag will not be stored. At the end of a subroutine or an interrupt (indicated by a return instruction RET), the contents of the stack register are returned to the PC.

Working Registers - R0,R1,R2,R3,R4

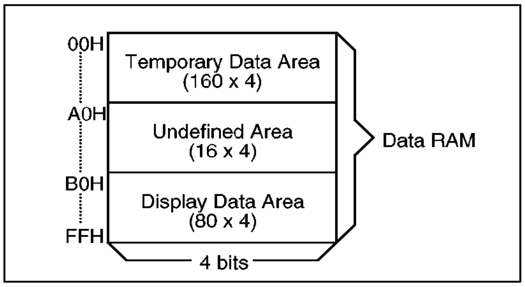
There are 5 working registers (R0,R1,R2,R3, R4) usually used to store the frequently accessed intermediate results. Using the instructions INC Rn and DEC Rn the working registers can increment (+1) or decrement (–1). The JNZ Rn (n=0,1,4) instruction makes efficient use of the working registers as a program loop counter. Also the register pairs R0,R1 and R2,R3 are used as a data memory pointer when the memory transfer instruction is executed.

Data Memory - RAM

The static data memory (RAM) is arranged in 256×4 bit format and is used to store data. All of the data memory locations are indirectly addressable through the register pair R1,R0 or R3,R2; for example MOV A,[R3R2] or MOV [R3R2],A.



HT44300 Data Memory



HT443A0 Data Memory

There are two areas in the data memory, the temporary data area and the display data area. Access to the temporary data area is from 00H to 7FH for the HT44300 and from 00H to 9FH for the HT443A0. Locations E0H to FFH on the HT44300 and B0H to FFH on the HT443A0 represent the display data area. The locations between the temporary and display data areas are undefined and cannot be used.

When data is written into the display data area it is automatically read by the LCD driver which then generates the corresponding LCD driving signals.

The relationship between the data pointer RAM locations is shown in the table.

Accumulator - ACC

The accumulator is the most important data register in the processor. It is one of the sources of input to the ALU and the destination of the results of the operations performed in the ALU. Data to and from the I/O ports and memory also passes through the accumulator.

Arithmetic and Logic Unit - ALU

This circuit performs the following arithmetic and logical operations ...

- Add with or without carry
- Subtract with or without carry
- AND, OR, Exclusive-OR
- Rotate right, left through carry
- BCD decimal adjust for addition
- Increment, decrement
- Data transfers
- Branch decisions

The ALU not only outputs the results of data operations, but also sets the status of the carry flag (CF) in some instructions.

Timer/Counter

The HT44300/443A0 contains a programmable 8-bit count-up counter which can be used to count external events or as a clock to generate an accurate time base.

If the 8-bit timer clock is supplied by an external source from pin TMCLK then synchronization problems may occur when reading the data from the timer. It is therefore suggested that the timer is stopped before retrieving the data. The 8 bit counter will increment on the rising edge of the clock whether internally or externally generated.

The Timer/Counter may be set and read with software instructions and stopped by a hardware reset or a TIMER OFF instruction. To restart the timer load the counter with the value XXH and then issue a TIMER ON instruction. Note that XX is the desired start count immediate value of the 8 bits. Once the Timer/Counter is started it increments to a maximum count of FFH and then overflows to zero (00H). It then continues to count until stopped by a TIMER OFF instruction or a reset.

The increment from the maximum count of FFH to a zero (00H) triggers a timer flag TF and an internal interrupt request. The interrupt may be enabled or disabled by executing the EI and DI instruction. If the interrupt is enabled the timer overflow will cause a subroutine call to location 4. The state of the timer flag is also testable with the conditional jump instruction JTMR. The timer flag is cleared after the interrupt or the JTMR instruction is executed.

If an internal source is used the frequency is determined by the system clock and the parameter n as defined in the equation. The frequency of the internal frequency source can be selected by mask option.

$$\text{Frequency of TIMER clock} = \frac{\text{system clock}}{2^n}$$

where n=0,1,2 ...13 selectable by mask option.

Note that for both devices n cannot have the value of 6, which is reserved for internal use.

Interrupt

The HT44300/443A0 provides both internal and external interrupt modes. The DI and EI instructions are used to disable and enable the interrupts. When the $\overline{\text{INT}}$ pin is triggered on a high to low transition in the enable interrupt mode and the program is not within a CALL subroutine, the external interrupt is activated. This causes a subroutine call to location 8 and resets the interrupt latch.

Likewise when the timer flag is set in the enable interrupt mode and the program is not within a CALL subroutine the internal interrupt is activated. This causes a subroutine call to location 4 and resets the timer flag. If both external and internal interrupts arrive at the same time then the external one will be serviced first.

When running under a CALL subroutine or DI the interrupt acknowledge is on hold until the RET or EI instruction is invoked. The CALL instruction should not be used within an interrupt routine as unpredictable behaviour may occur. If within a CALL subroutine both internal and external interrupts occur, no matter what order they arrive in the external interrupt will be serviced first after leaving the CALL subroutine. This also applies if the two interrupts arrive at the same time.

The interrupts are disabled by a hardware reset or a DI instruction. They remain disabled until the EI instruction is executed.

Each input port bit can be programmed by mask option to have an external interrupt function in the HALT mode.

Initial Reset

The HT44300/443A0 provides an $\overline{\text{RES}}$ pin for system initialization. This pin is equipped with an internal pull high resistor and in combination with an external 0.1μ~1μF capacitor, provides an internal reset pulse of sufficient length to guarantee a reset to all internal circuits. If the reset pulse is generated externally, the $\overline{\text{RES}}$ pin must be held low for at least 5ms. Normal circuit operation will not commence until the $\overline{\text{RES}}$ pin returns high.

The reset performs the following functions:

- Sets the program counter PC to 000H
- Disables the interrupt mode
- Stops the timer
- Resets the timer and carry flag
- Clears the carry flag
- Sets the sound off and one sing mode
- Sets port A high or floating

Halt

This is a special feature of the HT44300/443A0 to interrupt the chip's normal operation and reduce power consumption. When a HALT is executed the following happens ...

- The system clock will be stopped
- The contents of the on-chip RAM and registers remain unchanged
- All of the LCD segments and commons will have the VLCD voltage so the LCD becomes blank

The halt status can be terminated by an external interrupt or a hardware reset.

In the HALT mode any bit of ports PP, PS, PM can be used as external interrupts set by mask option to wake-up the system. This signal is active on a low-going transition.

When the halt status is terminated by an external interrupt, the following procedure takes place ...

Case 1: If the system is in an interrupt-disable state before entering the halt state:

- The instruction HALT is executed and the system enters a halt state
- A falling edge transition on $\overline{\text{INT}}$, or on any of the wake up pins on ports PP,PS or PM, will awaken the system and return to the main program instruction following the HALT command.
- An interrupt signal, whether caused by $\overline{\text{INT}}$ or the ports PP,PS or PM, will be held until the system receives an enable interrupt command at which point the held interrupt will be serviced.

Case 2: If the system is in an interrupt enable state:

- The instruction HALT is executed and the system enters a halt state
- A falling edge transition on $\overline{\text{INT}}$, or on any of the wake up pins on ports PP,PS or PM, will awaken the system and execute the external interrupt subroutine

Sound Effects

The HT44300/443A0 includes sound effect circuitry which offers up to 16 sounds with 3 tone, boom and noise effects. Holtek supports a sound library which has melodies, alarms, machine guns etc..

Whenever the instruction "SOUND n" or "SOUND A" is executed, the specified sound will begin. Whenever "SOUND OFF" is executed, it terminates the singing sound immediately.

There are two singing modes, SONE mode and SLOOP mode activated by SOUND ONE and SOUND LOOP. In SONE mode the specified sound plays just once. In SLOOP mode the specified sound keeps re-playing.

Since sounds 0~11 contain 32 notes and sounds

12~15 contain 64 notes the latter possesses better sound than the former.

The frequency of the sound effect circuit can be selected by mask option.

$$\text{Frequency of sound effect circuit} = \frac{\text{system clock}}{2^m}$$

...where m=0,1,2,3,4,5.

Holtek's sound library supports only sound clock frequencies of 128K or 64K. To use Holtek's sound library the proper system clock and mask option should be selected.

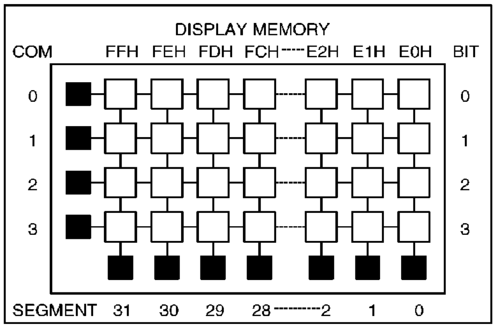
LCD Display Memory

As mentioned in the data memory section the LCD display memory is embedded in the data memory. It can be read and written to in the same way as normal data memory.

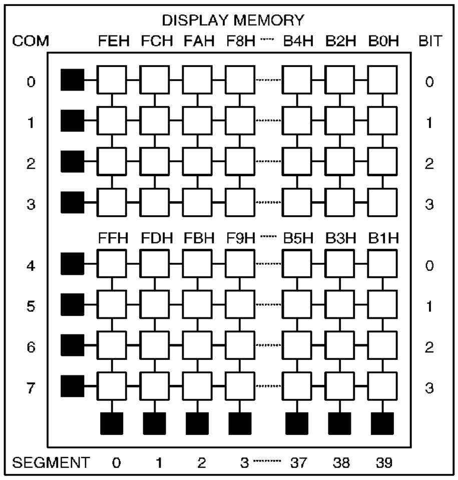
The figures show the mapping between the display memory and LCD pattern for both the HT44300 and the HT443A0.

To turn the display on or off a 1/0 is written to the corresponding bit of the display memory.

The LCD display module may have any form as long as the number of commons does not exceed 4 and the number of segments does not exceed 32 for the HT44300 and does not exceed 8 commons and 40 segments for the HT443A0.



HT44300 LCD Display Memory



HT443A0 LCD Display Memory

LCD Driver Output

All LCD segments are random after an initial clear. The bias voltage circuits of the LCD display is built-in and no external resistor is needed.

HT44300

The output number of the HT44300 LCD driver is 32×4 which can directly drive an LCD with 1/4 (or 1/3 by mask option) duty cycle and 1/3 bias.

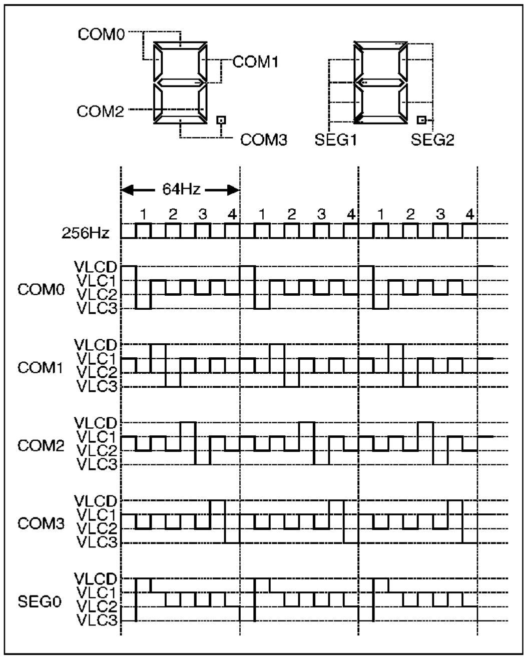
The frequency of the LCD driving clock is fixed at about 256Hz. This is set by Holtek according to the application and cannot be changed.

An example of an 8-segment digit display for the HT44300 is shown. The example shows the waveform of a "5" displayed with 1/4 duty, 1/3 bias.

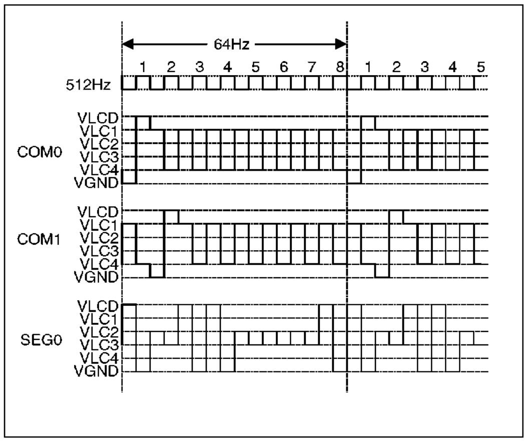
HT443A0

The output number of the HT443A0 LCD driver is 40×8 which can directly drive an LCD with 1/8 duty cycle and 1/4 or 1/5 bias.

The frequency of the LCD driving clock is fixed at about 512 Hz. This is set by Holtek according to the application and cannot be changed.



HT44300 LCD Driver Output

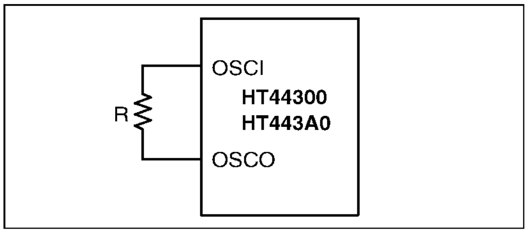


HT443A0 LCD Driver Output

An example of an LCD driving waveform with 1/8 duty and 1/5 bias is shown for the HT443A0.

Oscillator

Only one external resistor is needed for the HT44300/443A0 oscillator circuit.



RC Oscillator

The system clock is also used as the reference signal of the LCD driving clock, sound effect clock, or internal frequency source of TIMER.

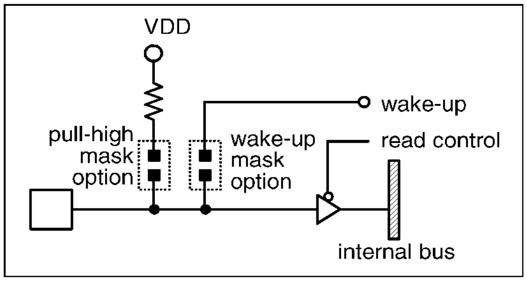
One HT44300/443A0 machine cycle consists of a sequence of 4 states numbered T1 to T4. Each state lasts for one oscillator period. The machine cycle is 1.0μs if the system frequency is up to 4.0MHz.

Interfacing

The HT44300/443A0 microcontrollers communicate with the outside world through three 4-bit input ports PP, PS and PM and one 4-bit output port PA.

Input Ports - PP, PS, PM

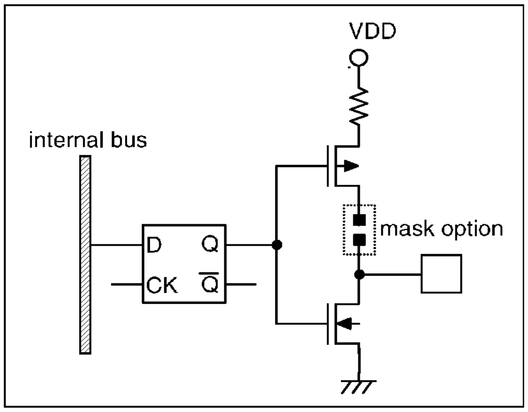
All ports can have internal pull high resistors determined by mask option. Every bit of the input ports PP, PS and PM can be specified to be a trigger source to wake up the HALT interrupt by mask option . A high to low transition on one of these pins will wake up the device from a HALT status.



Input Ports PP, PS, PM

Output port PA

A mask option is available to select whether the output is a CMOS or open drain NMOS type.



Output Port PA

After an initial clear the output port PA defaults to be high for CMOS or floating for NMOS.

Mask Options

The following either/or options are available by mask option which the user must select prior to manufacture.

- 4-bit input ports PP, PS and PM with or without pull high resistors
- Each bit of PP, PS and PM can wake up the processor from a HALT state
- Output Port PA to be CMOS or open drain NMOS
- 8-bit programmable timer with external clock or internal frequency source. Thirteen internal frequency sources are available to provide an internal clock. Note that a value of n=6 cannot be used for the devices. If the internal frequency sources are used as a clocking signal then TMCLK cannot be connected to a pull-high resistor
- Two sound effects clock frequencies 128K or 64K
- Two kinds of LCD applications: 1/4 duty 1/3 bias or 1/3 duty 1/3 bias - HT44300 only
- 1/8 duty 1/4 bias for VDD = 3V or 1/8 duty 1/5 bias for VDD = 5V - HT443A0 only

Software Tools

To ease the programming task and to reduce development time Holtek supplies a development system for the HT44300/443A0. The system runs under an IBM PC-XT/AT environment and consists of both a hardware emulation board and a suite of programs including powerful debug functions. The user can download the code from the PC to the emulation board for verification. The main features of the system are as follows.

- Can incorporate the user's text editor or word processor with Holtek's cross assembler to form an integrated development system
- Supports mouse functions with its window based human interface
- Performs stand-alone operation for demonstration purposes
- Auto-executes self test function at every power on reset
- Provides symbolic debugging capabilities
- User defined mask options
- RC with variable resistor
- Displays and modifies registers, carry flag, timer, port output level and internal RAM
- Single instruction stepping
- Jumps unconditionally to any address and halts anytime during execution
- Provides up to 8 breakpoint settings
- Real time 255 forward step or 256 backward step trace

After program verification on the emulation board the customer supplies Holtek with the verified code prior to manufacture.

INSTRUCTION SET

Instruction Set Summary

Mnemonic	Description	Byte	Cycle	CF
Arithmetic				
ADD A,[R1R0]	Add data memory to ACC	1	1	√
ADC A,[R1R0]	Add data memory with carry to ACC	1	1	√
SUB A,[R1R0]	Subtract data memory from ACC	1	1	√
SBC A,[R1R0]	Subtract data memory from ACC with borrow	1	1	√
ADD A,XH	Add immediate data to ACC	2	2	√
SUB A,XH	Subtract immediate data from ACC	2	2	√
DAA	Decimal adjust ACC for addition	1	1	√
Logic Operation				
AND A,[R1R0]	AND data memory to ACC	1	1	—
OR A,[R1R0]	OR data memory to ACC	1	1	—
XOR A,[R1R0]	Exclusive-OR data memory to ACC	1	1	—
AND [R1R0],A	AND ACC to data memory	1	1	—
OR [R1R0],A	OR ACC to data memory	1	1	—
XOR [R1R0],A	Exclusive-OR ACC to data memory	1	1	—
AND A,XH	AND immediate data to ACC	2	2	—
OR A,XH	OR immediate data to ACC	2	2	—
XOR A,XH	Exclusive-OR immediate data to ACC	2	2	—
Increment & Decrement				
INC A	Increment ACC	1	1	—
INC Rn	Increment register, n=0~4	1	1	—
INC [R1R0]	Increment data memory	1	1	—
INC [R3R2]	Increment data memory	1	1	—
DEC A	Decrement ACC	1	1	—
DEC Rn	Decrement register, n=0~4	1	1	—
DEC [R1R0]	Decrement data memory	1	1	—
DEC [R3R2]	Decrement data memory	1	1	—

Mnemonic	Description	Byte	Cycle	CF
Data Move				
MOV A,Rn	Move register to ACC, n=0~4	1	1	—
MOV Rn,A	Move ACC to register, n=0~4	1	1	—
MOV A,[R1R0]	Move data memory to ACC	1	1	—
MOV A,[R3R2]	Move data memory to ACC	1	1	—
MOV [R1R0],A	Move ACC to data memory	1	1	—
MOV [R3R2],A	Move ACC to data memory	1	1	—
MOV A,XH	Move immediate data to ACC	1	1	—
MOV R1R0,XXH	Move immediate data to R1 and R0	2	2	—
MOV R3R2,XXH	Move immediate data to R3 and R2	2	2	—
MOV R4,XH	Move immediate data to R4	2	2	—
Rotate				
RL A	Rotate ACC left	1	1	√
RLC A	Rotate ACC left through the carry	1	1	√
RR A	Rotate ACC right	1	1	√
RRC A	Rotate ACC right through the carry	1	1	√
Input & Output				
IN A,Pi	Input port-i to ACC ,port–i=PM,PS,PP	1	1	—
OUT PA,A	Output ACC to port-A	1	1	—
Branch				
JMP addr	Jump unconditionally	2	2	—
JC addr	Jump on carry=1	2	2	—
JNC addr	Jump on carry=0	2	2	—
JTMR addr	Jump on timer overflow	2	2	—
JAn addr	Jump on ACC bit n=1	2	2	—
JZ A,addr	Jump on ACC is zero	2	2	—
JNZ A,addr	Jump on ACC is not zero	2	2	—
JNZ Rn,addr	Jump on register Rn not zero, n=0,1,4	2	2	—
Subroutine				
CALL addr	Subroutine call	2	2	—
RET	Return from subroutine or interrupt	1	1	—
RETI	Return from interrupt service routine	1	1	√

Mnemonic	Description	Byte	Cycle	CF
Flag				
CLC	Clear carry flag	1	1	0
STC	Set carry flag	1	1	1
EI	Enable interrupt	1	1	—
DI	Disable interrupt	1	1	—
NOP	No operation	1	1	—
Timer				
TIMER XXH	Set 8 bits immediate data to TIMER	2	2	—
TIMER ON	Set TIMER start counting	1	1	—
TIMER OFF	Set TIMER stop counting	1	1	—
MOV A,TMRL	Move low nibble of TIMER to ACC	1	1	—
MOV A,TMRH	Move high nibble of TIMER to ACC	1	1	—
MOV TMRL,A	Move ACC to low nibble of TIMER	1	1	—
MOV TMRH,A	Move ACC to hight nibble of TIMER	1	1	—
Table Read				
READ R4A	Read ROM code of current page to R4 & ACC	1	2	—
READ MR0A	Read ROM code of current page to M(R1,R0), ACC	1	2	—
READF R4A	Read ROM code of page F to R4 & ACC	1	2	—
READF MR0A	Read ROM code of page F to M(R1,R0),ACC	1	2	—
Sound Control				
SOUND n	Activate SOUND channel n	2	2	—
SOUND A	Activate SOUND channel with ACC	1	1	—
SOUND ONE	Turn on SOUND one cycle	1	1	—
SOUND LOOP	Turn on SOUND repeat cycle	1	1	—
SOUND OFF	Turn off SOUND	1	1	—
Miscellaneous				
HALT	Enter power down mode	2	2	—

Instruction Definitions

ADC A,[R1R0] Add data memory contents and carry to accumulator

Machine code 0 0 0 0 1 0 0 0

Description The contents of the data memory addressed by the register pair “R1,R0” and the carry are added to the accumulator. Carry is affected.

Operation ACC ← ACC+M(R1,R0)+CF

ADD A,XH Add immediate data to accumulator

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

Description The specified data is added to the accumulator. Carry is affected.

Operation ACC ← ACC+XH

ADD A,[R1R0] Add data memory contents to accumulator

Machine code 0 0 0 0 1 0 0 1

Description The contents of the data memory addressed by the register pair “R1,R0” is added to the accumulator. Carry is affected.

Operation ACC ← ACC+M(R1,R0)

AND A,XH Logical AND immediate data to accumulator

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

Description Data in the accumulator is logically ANDed with the immediate data specified by the code.

Operation ACC ← ACC “AND” XH

AND A,[R1R0] Logical AND accumulator with data memory

Machine code 0 0 0 1 1 0 1 0

Description Data in the accumulator is logically ANDed with the data memory addressed by the register pair “R1,R0”.

Operation ACC ← ACC “AND” M(R1,R0)

AND [R1R0],A

Logical AND data memory with accumulator

Machine code 0 0 0 1 1 1 0 1

Description Data in the data memory addressed by the register pair “R1,R0” is logically ANDed with the accumulator

Operation M(R1,R0) ← M(R1,R0) “AND” ACC

CALL address

Subroutine call

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

Description The program counter bits 0~11 are saved in the stack and the specified address loaded into the program counter.

Operation Stack ← PC+2
PC ← address

CLC

Clear carry flag

Machine code 0 0 1 0 1 0 1 0

Description The carry flag is reset to zero.

Operation CF ← 0

DAA

Decimal-Adjust accumulator

Machine code 0 0 1 1 0 1 1 0

Description The accumulator value is adjusted to BCD (Binary Code Decimal), if the contents of the accumulator is greater than 9 or CF (Carry flag) is one.

Operation If ACC>9 or CF=1 then
ACC ← ACC+6, CF ← 1
else
ACC ← ACC, CF ← CF

DEC A

Decrement accumulator

Machine code 0 0 1 1 1 1 1 1

Description Data in the accumulator is decremented by one. Carry flag is not affected.

Operation ACC ← ACC–1

DEC Rn

Decrement register

Machine code 0 0 0 1 n n n 1

Description Data in the working register “Rn” is decremented by one. Carry flag is not affected.

Operation Rn ← Rn–1; Rn=R0,R1,R2,R3,R4, for nnn=0,1,2,3,4

DEC [R1R0]

Decrement data memory

Machine code 0 0 0 0 1 1 0 1

Description Data in the data memory specified by the register pair “R1,R0” is decremented by one. Carry flag is not affected.

Operation M(R1,R0) ← M(R1,R0)–1

DEC [R3R2]

Decrement data memory

Machine code 0 0 0 0 1 1 1 1

Description Data in the data memory specified by the register pair “R3,R2” is decremented by one. Carry flag is not affected.

Operation M(R3,R2) ← M(R3,R2)–1

DI

Disable interrupt

Machine code 0 0 1 0 1 1 0 1

Description Internal time-out interrupt and external interrupt are disabled.

EI

Enable interrupt

Machine code 0 0 1 0 1 1 0 0

Description Internal time-out interrupt and external interrupt are enabled.

HALT

Halt system clock

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

Description Turn off system clock, and enter power down mode.

Operation PC ← PC+2

IN A,Pi

Input port to accumulator

Machine code PM 0 0 1 1 0 0 1 0

PS 0 0 1 1 0 0 1 1

PP 0 0 1 1 0 1 0 0

Description The data on port “Pi” is transferred to the accumulator.

Operation ACC ← Pi; Pi=PM, PS or PP

INC A

Increment accumulator

Machine code 0 0 1 1 0 0 0 1

Description Data in the accumulator is incremented by one. Carry flag is not affected.

Operation ACC ← ACC+1

INC Rn

Increment register

Machine code 0 0 0 1 n n n 0

Description Data in the working register “Rn” is incremented by one. Carry flag is not affected.

Operation Rn ← Rn+1; Rn=R0~R4 for nnn=0~4

INC [R1R0]

Increment data memory

Machine code 0 0 0 0 1 1 0 0

Description Data in the data memory specified by the register pair “R1,R0” is incremented by one. Carry flag is not affected.

Operation M(R1,R0) ← M(R1,R0)+1

INC [R3R2]

Increment data memory

Machine code 0 0 0 0 1 1 1 0

Description Data memory specified by the register pair “R3,R2” is incremented by one. Carry flag is not affacted.

Operation M(R3,R2) ← M(R3,R2)+1

JAn address

Jump if accumulator bit n is set

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

Description Bits 0~10 of the program counter are replaced with the directly-specified address but bit 11 of the program counter is unaffected, if accumulator bit n is set to one.

Operation PC (bit 0~10) ← address, if ACC bit n=1(n=0~3)
PC ← PC+2, if ACC bit n=0

JC address

Jump if carry is set

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

Description Bits 0~10 of the program counter are replaced with the directly-specified address but bit 11 of the program counter is unaffected, if the CF (Carry flag) is set to one.

Operation PC (bit 0~10) ← address, if CF=1
PC ← PC+2, if CF=0

JMP address

Direct jump

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

Description Bits 0~11 of the program counter are replaced with the directly-specified address.

Operation PC ← address

JNC address

Jump if carry is not set

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

Description Bits 0~10 of the program counter are replaced with the directly-specified address and bit 11 of the program counter is unaffected, if the CF (Carry flag) is set to zero.

Operation PC (bit 0~10) ← address, if CF=0
PC ← PC+2, if CF=1

JNZ A,address

Jump if accumulator is not zero

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

Description Bits 0~10 of the program counter are replaced with the directly-specified address but bit 11 of the program counter is unaffected, if the accumulator is not zero.

Operation PC (bit 0~10) ← address, if ACC≠0
PC ← PC+2, if ACC=0

JNZ Rn,address

Jump if register is not zero

Machine code

R0 1 0 1 0 0 a a a a a a a a a a a

R1 1 0 1 0 1 a a a a a a a a a a a

R4 1 1 0 1 1 a a a a a a a a a a a

Description

Bits 0~10 of the program counter are replaced with the directly-specified address but bit 11 of the program counter is unaffected, if the register is not zero.

Operation

PC (bit 0~10) ← address, if Rn≠0; Rn=R0,R1,R4
PC ← PC+2, if Rn=0

JTMR address

Jump if time-out

Machine code

1 1 0 1 0 a a a a a a a a a a a

Description

Bits 0~10 of the program counter are replaced with the directly-specified address but bit 11 of the program counter is unaffected, if the TF (Timer flag) is set to one.

Operation

PC (bit 0~10) ← address, if TF=1
PC ← PC+2, if TF=0

JZ A,address

Jump if accumulator is zero

Machine code

1 0 1 1 0 a a a a a a a a a a a

Description

Bits 0~10 of the program counter are replaced with the directly-specified address but bit 11 of the program counter is unaffected, if the accumulator is zero.

Operation

PC (bit 0~10) ← address, if ACC=0
PC ← PC+2, if ACC≠0

MOV A,Rn

Move register to accumulator

Machine code

0 0 1 0 n n n 1

Description

Data in the working register “Rn” is moved to the accumulator.

Operation

ACC ← Rn; Rn=R0~R4, for nnn=0~4

MOV A,TMRH

Move timer high nibble to accumulator

Machine code

0 0 1 1 1 0 1 1

Description

The high nibble data of the timer counter is loaded to the accumulator.

Operation

ACC ← TIMER (high nibble)

MOV A,TMRL	Move timer low nibble to accumulator
Machine code	0 0 1 1 1 0 1 0
Description	The low nibble data of the timer counter is loaded to the accumulator.
Operation	ACC ← TIMER (low nibble)

MOV A,XH	Move immediate data to accumulator
Machine code	0 1 1 1 d d d d
Description	The 4-bit data specified by the code is loaded to the accumulator.
Operation	ACC ← XH

MOV A,[R1R0]	Move data memory to accumulator
Machine code	0 0 0 0 0 1 0 0
Description	Data in the data memory specified by the register pair “R1,R0” is moved to the accumulator.
Operation	ACC ← M(R1,R0)

MOV A,[R3R2]	Move data memory to accumulator
Machine code	0 0 0 0 0 1 1 0
Description	Data in the data memory specified by the register pair “R3,R2” is moved to the accumulator.
Operation	ACC ← M(R3,R2)

MOV R1R0,XXH	Move immediate data to R1 and R0
Machine code	0 1 0 1 d d d d 0 0 0 0 d d d d
Description	The 8-bit data specified by the code is loaded to the working registers R1 and R0, the high nibble of the data is loaded to R1, and the low nibble to R0.

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

MOV R3R2,XXH Move immediate data to R3 and R2

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

Description The 8-bit data specified by the code is loaded to the working registers R3 and R2, the high nibble of the data is loaded to R3, and the low nibble to R2.

Operation R3 ← XH (high nibble)
R2 ← XH (low nibble)

MOV R4,XH Move immediate data to R4

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

Description The 4-bit data specified by the code is loaded to the working register R4.

Operation R4 ← XH

MOV Rn,A Move accumulator to register

Machine code 0 0 1 0 n n n 0

Description Data in the accumulator is moved to the working register "Rn".

Operation Rn ← ACC; Rn=R0~R4, for nnn=0~4

MOV TMRH,A Move accumulator to timer high nibble

Machine code 0 0 1 1 1 1 0 1

Description The contents of the accumulator is loaded to the high nibble of the timer counter.

Operation TIMER(high nibble) ← ACC

MOV TMRL,A Move accumulator to timer low nibble

Machine code 0 0 1 1 1 1 0 0

Description The contents of the accumulator is loaded to the low nibble of the timer counter.

Operation TIMER(low nibble) ← ACC

MOV [R1R0],A Move accumulator to data memory

Machine code 0 0 0 0 0 1 0 1

Description Data in the accumulator is moved to the data memory specified by the register pair "R1,R0".

Operation M(R1,R0) ← ACC

MOV [R3R2],A

Move accumulator to data memory

Machine code 0 0 0 0 0 1 1 1

Description Data in the accumulator is moved to the data memory specified by the register pair “R3,R2”.

Operation M(R3,R2) ← ACC

NOP

No operation

Machine code 0 0 1 1 1 1 1 0

Description Do nothing, but one instruction cycle is delayed.

OR A,XH

Logical OR immediate data to accumulator

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

Description Data in the accumulator is logically ORed with the immediate data specified by the code.

Operation ACC ← ACC “OR” XH

OR A,[R1R0]

Logical OR accumulator with data memory

Machine code 0 0 0 1 1 1 0 0

Description Data in the accumulator is logically ORed with the data memory addressed by the register pair “R1,R0”.

Operation ACC ← ACC “OR” M(R1,R0)

OR [R1R0],A

Logically OR data memory with accumulator

Machine code 0 0 0 1 1 1 1 1

Description Data in the data memory addressed by the register pair “R1,R0” is logically ORed with the accumulator.

Operation M(R1,R0) ← M(R1,R0) “OR” ACC

OUT PA,A

Output accumulator data to port A

Machine code 0 0 1 1 0 0 0 0

Description The data in the accumulator is transferred to port PA and latched.

Operation PA ← ACC

READ MR0A

Read ROM code of current page to M(R1,R0) and ACC

Machine code

0 1 0 0 1 1 1 0

Description

The 8-bit ROM code (current page) addressed by ACC and R4 is moved to the data memory M(R1,R0) and the accumulator. The high nibble of the ROM code is loaded to M(R1,R0) and the low nibble of the ROM code is loaded to the accumulator. The address of the ROM code is specified as below:
Current page → ROM code address bit 11~8
ACC → ROM code address bit 7~4
R4 → ROM code address bit 3~0

Operation

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

READ R4A

Read ROM code of current page to R4 and accumulator

Machine code

0 1 0 0 1 1 0 0

Description

The 8-bit ROM code (current page) addressed by ACC and M(R1,R0) is moved to the working register R4 and the accumulator. The high nibble of the ROM code is loaded to R4 and the low nibble of the ROM code is loaded to the accumulator. The address of the ROM code is specified as below:
Current page → ROM code address bit 11~8
ACC → ROM code address bit 7~4
M(R1,R0) → ROM code address bit 3~0

Operation

R4 ← ROM code (high nibble)
ACC ← ROM code (low nibble)

READF MR0A

Read ROM Code of page F to M(R1,R0) and ACC

Machine code

0 1 0 0 1 1 1 1

Description

The 8-bit ROM code (page F) addressed by ACC and R4 is moved to the data memory M(R1,R0) and the accumulator. The high nibble of the ROM code is loaded to M(R1,R0) and the low nibble of the ROM code is loaded to the accumulator.
Page F → ROM code address bit 11~8 are “1111”
ACC → ROM code address bit 7~4
R4 → ROM code address bit 3~0

Operation

M(R1,R0) ← high nibble of ROM code (page F)
ACC ← low nibble of ROM code (page F)

READF R4A	Read ROM code of page F to R4 and accumulator
Machine code	0 1 0 0 1 1 0 1
Description	The 8-bit ROM code (page F) addressed by ACC and M(R1,R0) is moved to the working register R4 and the accumulator. The high nibble of the ROM code is loaded to R4 and the low nibble of the ROM code is loaded to the accumulator. Page F → ROM code address bit 11~8 are "1111" ACC → ROM code address bit 7~4 M(R1,R0) → ROM code address bit 3~0

Operation	R4 ← high nibble of ROM code (page F)
	ACC ← low nibble of ROM code (page F)

RET	Return from subroutine or interrupt
Machine code	0 0 1 0 1 1 1 0
Description	The program counter bits 0~11 are restored from the stack.
Operation	PC ← Stack

RETI	Return from interrupt subroutine
Machine code	0 0 1 0 1 1 1 1
Decription	The program counter bits 0~11 are restored from the stack. The carry flag before entering the interrupt service routine is restored.
Operation	PC ← Stack CF ← CF (before interrupt service routine)

RL A	Rotate accumulator left
Machine code	0 0 0 0 0 0 0 1
Description	The contents of the accumulator are rotated left one bit. Bit 3 is rotated to both bit 0 and the carry flag.

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

RLC A

Rotate accumulator left through carry

Machine code: 0 0 0 0 0 0 1 1

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

Operation:
$A_{n+1} \leftarrow A_n$, A_n: Accumulator bit n (n=0,1,2)
$A_0 \leftarrow CF$
$CF \leftarrow A_3$

RR A

Rotate accumulator right

Machine code: 0 0 0 0 0 0 0 0

Description: The contents of the accumulator are rotated right one bit. Bit 0 is rotated to both bit 3 and the carry flag.

Operation:
$A_n \leftarrow A_{n+1}$, A_n: Accumulator bit n (n=0,1,2)
$A_3 \leftarrow A_0$
$CF \leftarrow A_0$

RRC A

Rotate accumulator right through carry

Machine code: 0 0 0 0 0 0 1 0

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

Operation:
$A_n \leftarrow A_{n+1}$, A_n: Accumulator bit n (n=0,1,2)
$A_3 \leftarrow CF$
$CF \leftarrow A_0$

SBC A,[R1R0]

Subtract data memory contents and carry from ACC

Machine code: 0 0 0 0 1 0 1 0

Description: The contents of the data memory addressed by the register pair "R1,R0" and the complement of the carry are subtracted from the accumulator. Carry is set if a borrow does not take place in subtraction; otherwise carry is cleared.

Operation: $ACC \leftarrow ACC+\overline{M(R1,R0)}+CF$

SOUND A

Activate SOUND channel with accumulator

Machine code: 0 1 0 0 1 0 1 1

Description: The activated sound begins playing in accordance with the contents of accumulator when the specified sound channel is matched.

SOUND LOOP	Turn on sound repeat cycle
Machine code	0 1 0 0 1 0 0 1
Description	The activated sound plays repeatedly.

SOUND OFF	Turn off sound
Machine code	0 1 0 0 1 0 1 0
Description	The activated sound will terminate immediately.

SOUND ONE	Turn on sound one cycle
Machine code	0 1 0 0 1 0 0 0
Description	The activated sound plays once.

SOUND n	Activate SOUND channel n
Machine code	0 1 0 0 0 1 0 1 0 0 0 0 n n n n
Description	The specified sound begins playing and overwrites the previous activated sound. (nnnn=0~15)

STC	Set carry flag
Machine code	0 0 1 0 1 0 1 1
Description	The carry flag is set to one.
Operation	CF ← 1

SUB A,XH	Subtract immediate data from accumulator
Machine code	0 1 0 0 0 0 0 1 0 0 0 0 d d d d
Description	The specified data is subtracted from the accumulator. Carry is set if a borrow does not take place in subtraction; otherwise carry is cleared.
Operation	ACC ← ACC+$\overline{XH}$+1

SUB A,[R1R0]

Subtract data memory contents from accumulator

Machine code: 0 0 0 0 1 0 1 1

Description: The contents of the data memory addressed by the register pair “R1,R0” is subtracted from the accumulator. Carry is set if a borrow does not take place in subtraction; otherwise carry is cleared.

Operation: ACC ← ACC+$\overline{\text{M(R1,R0)}}$+1

TIMER OFF

Set timer stop counting

Machine code: 0 0 1 1 1 0 0 1

Description: The timer stops counting, when the “TIMER OFF” instruction is executed.

TIMER ON

Set timer start counting

Machine code: 0 0 1 1 1 0 0 0

Description: The timer starts counting, when the “TIMER ON” instruction is executed.

TIMER XXH

Set immediate data to timer counter

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

Description: The 8-bit data specified by the code is loaded to the timer counter.

Operation: TIMER ← XXH

XOR A,XH

Logical XOR immediate data to accumulator

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

Description: Data in the accumulator is Exclusive-ORed with the immediate data specified by the code.

Operation: ACC ← ACC “XOR” XH

XOR A,[R1R0]

Logical XOR accumulator with data memory

Machine code: 0 0 0 1 1 0 1 1

Description: Data in the accumulator is Exclusive-ORed with the data memory addressed by the register pair “R1,R0”.

Operation: ACC ← ACC “XOR” M(R1,R0)

XOR [R1R0],A	Logical XOR data memory with accumulator
Machine code	0 0 0 1 1 1 1 0
Description	Data in the data memory addressed by the register pair “R1,R0” is logically Exclusive-ORed with the accumulator.
Operation	M(R1,R0) ← M(R1,R0) “XOR” ACC