HOLTEK

HTG12G0 Microcontroller

Features

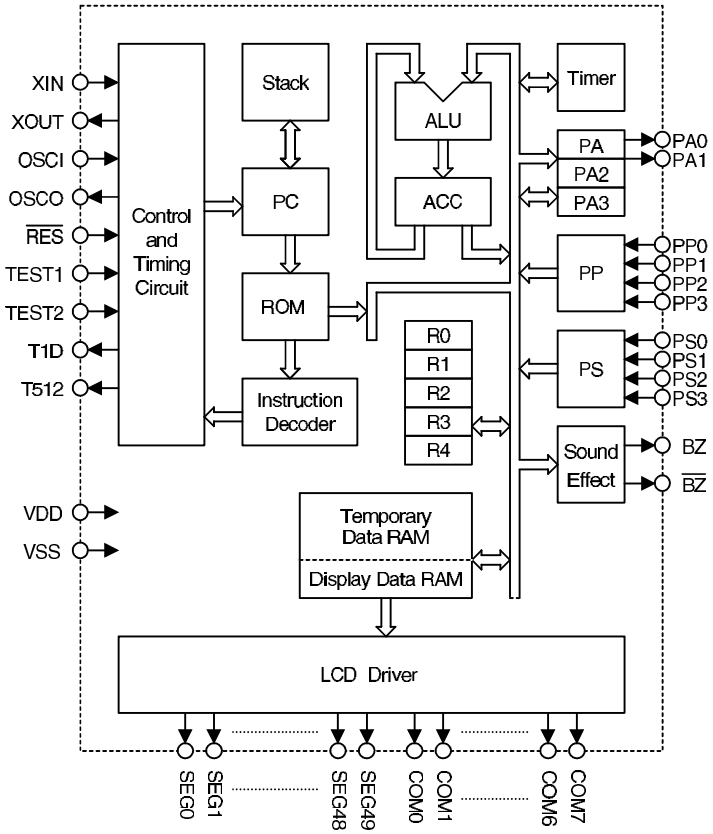
- Operating voltage: 2.4V~3.3V
- Eight input lines
- Two output lines
- Five working registers
- RC oscillator for system clock
- Crystal oscillator for RTC and LCD clock
- 8K×8 program ROM
- 156×4 data RAM
- 50×8 segment LCD driver, 1/5 bias, 1/8 duty
- 8-bit programmable timer with built-in frequency source
- Internal timer overflow and RTC interrupt
- 16 kinds of programmable sound effects
- Halt function and wake-up feature reduce power consumption
- One-level subroutine nesting
- 8-bit table read instruction
- Up to 4.0μs instruction cycle with 1.0MHz system clock at V_{DD}=3V
- 95 powerful instructions

General Description

The HTG12G0 is a 4-bit single chip microcontroller specially designed for LCD product applications.

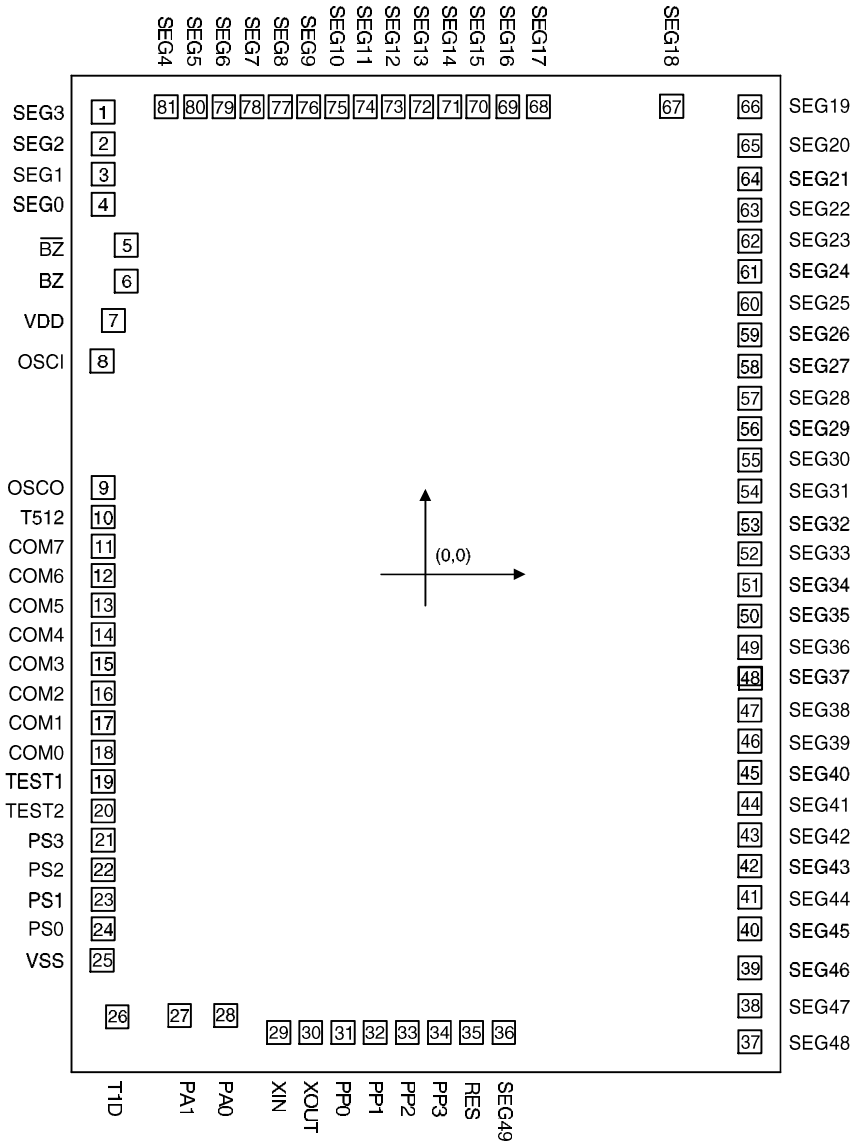
It is ideally suited for applications requiring low power consumption, with many LCD segments such as calculator, scale, subsystem controller, hand-held LCD products and electronic appliances.

Block Diagram



Notes: ACC: Accumulator
PC: Program counter
R0~R4: Working registers
PA0~PA1: Output ports
PP, PS: Input ports
PA3: ROM bank control bit
PA2: LCD on/off switch

Pad Assignment



Chip size: 3070×4440 $(\mu m)^2$

* 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	–1403.50	2012.75	42	1412.00	–1270.25
2	–1403.50	1872.75	43	1412.00	–1134.25
3	–1403.50	1741.75	44	1412.00	–998.25
4	–1403.50	1610.75	45	1412.00	–862.25
5	–1304.00	1431.75	46	1412.00	–726.25
6	–1304.00	1276.25	47	1412.00	–590.25
7	–1358.50	1104.25	48	1412.00	–454.25
8	–1409.50	929.25	49	1412.00	–318.25
9	–1403.50	377.75	50	1412.00	–182.25
10	–1403.50	249.75	51	1412.00	–46.25
11	–1403.50	121.75	52	1412.00	89.75
12	–1403.50	–6.25	53	1412.00	225.75
13	–1403.50	–134.25	54	1412.00	361.75
14	–1403.50	–262.25	55	1412.00	497.75
15	–1403.50	–390.25	56	1412.00	633.75
16	–1403.50	–518.25	57	1412.00	769.75
17	–1403.50	–646.25	58	1412.00	905.75
18	–1403.50	–774.25	59	1412.00	1041.75
19	–1403.50	–902.25	60	1412.00	1177.75
20	–1403.50	–1030.25	61	1412.00	1313.75
21	–1403.50	–1158.25	62	1412.00	1449.75
22	–1403.50	–1286.25	63	1412.00	1585.25
23	–1403.50	–1414.25	64	1412.00	1720.75
24	–1403.50	–1542.25	65	1412.00	1865.25
25	–1409.50	–1680.75	66	1412.00	2034.75
26	–1342.00	–1926.25	67	1071.50	2037.25
27	–1071.50	–1920.25	68	490.00	2034.75
28	–871.50	–1920.25	69	358.00	2034.75
29	–640.00	–1994.75	70	228.00	2034.75
30	–500.00	–1994.75	71	105.00	2034.75
31	–360.00	–1994.75	72	–18.00	2034.75
32	–220.00	–1994.75	73	–141.00	2034.75
33	–80.00	–1994.75	74	–264.00	2034.75
34	60.00	–1994.75	75	–387.00	2034.75
35	200.00	–1994.75	76	–510.00	2034.75
36	340.00	–1994.75	77	–633.00	2034.75
37	1412.00	–2034.75	78	–756.00	2034.75
38	1412.00	–1877.25	79	–879.00	2034.75
39	1412.00	–1712.25	80	–1002.00	2034.75
40	1412.00	–1542.25	81	–1129.50	2034.75
41	1412.00	–1406.25			

Pad Description

Pad No.	Pad name	I/O	Mask Option	Description
1~4, 36~81	SEG3~SEG0 SEG49~SEG4	O	—	LCD driver outputs for LCD panel segment
5 6	$\overline{BZ}$ BZ	O	*	Sound effect output
7	VDD	I	—	Positive power supply
8 9	OSCI OSCO	I O	—	An external resistor between OSCI and OSCO is needed for internal system clock.
10 19 20 26	T512 TEST1 TEST2 T1D	O I I O	—	For test mode only TEST1 and TEST2 must be open when the HTG12G0 is in normal operation (with an internal pull high resistor)
11~18	COM7~COM0	O	—	Output for LCD panel common plate
21~24	PS3~PS0	I	Pull-high or None **	4-bit port for input only
25	VSS	I	—	Negative power supply, GND
27 28	PA1 PA0	O	CMOS or NMOS Open Drain	2-bit latch port for output only
29 30	XIN XOUT	I O	—	32768Hz crystal oscillator for time base
31~34	PP0~PP3	I	Pull-high or None **	4-bit port for input only
35	$\overline{RES}$	I	—	Input for reset LSI inside Reset is active at logical low level

*: 6 internal sources deriving from the system clock can be selected as sound effect clock by mask option. If Holtek's sound library is invoked, only 128K and 64K is accepted.

**: Each bit of input ports PS, PP can be a trigger source of HALT interrupt. That can be specified by mask option.

Absolute Maximum Ratings

Supply Voltage .....V_{SS}–0.3V to 13V

Input Voltage.....V_{SS}–0.3 to V_{DD}+0.3

Operating Temperature..... 0°C to 70°C

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

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

D.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Uint
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	—	3.3	V
I_{DD}	Operating Current	3V	No load, f_{SYS}=500kHz	—	100	500	μA
I_{STB}	Standby Current (RTC OSC ON) and LCD ON	3V	System halt	—	10	20	μA
V_{IL1}	Input Low Voltage	3V	PS, PP	0	—	0.6	V
V_{IH1}	Input High Voltage	3V	PS, PP	2.1	—	3.0	V
V_{IL2}	Input Low Voltage	3V	$\overline{RES}$	0	—	0.6	V
V_{IH2}	Input High Voltage	3V	$\overline{RES}$	2.5	—	3.0	V
I_{OL1}	Port A, BZ and $\overline{BZ}$ Output Sink Current	3V	V_{DD}=3V, V_{OL}=0.3V	1.5	3.0	—	mA
I_{OH1}	Port A, BZ and $\overline{BZ}$ Output Source Current	3V	V_{DD}=3V, V_{OH}=2.7V	–0.5	–1.5	—	mA
I_{OL2}	Segment 0~7 Output Sink Current	3V	V_{LCD}=3V, V_{OL}=0.3V	80	100	—	μA
I_{OH2}	Segment 0~7 Output Source Current	3V	V_{LCD}=3V, V_{OH}=2.7V	–50	–70	—	μA
I_{OL3}	Segment 8~49 Output Sink Current	3V	V_{LCD}=3V, V_{OL}=0.3V	40	60	—	μA
I_{OH3}	Segment 8~49 Output Source Current	3V	V_{LCD}=3V, V_{OH}=2.7V	–20	–40	—	μA
I_{OL4}	Common Output Sink Current	3V	V_{LCD}=3V, V_{OL}=0.3V	100	120	—	μA
I_{OH4}	Common Output Source Current	3V	V_{LCD}=3V, V_{OH}=2.7V	–100	–130	—	μA
R_{PH1}	Pull-high Resistance	3V	PS, PP	15	—	200	kΩ
R_{PH2}	Pull-high Resistance	3V	$\overline{RES}$	100	—	300	kΩ

A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SYS}	System Clock	3V	R:620kΩ~51kΩ	100	—	1000	kHz
f_{LCD}	LCD Clock	3V	—	—	512	—	Hz
t_{COM}	LCD Common Period	—	1/8 duty	—	$(1/f_{LCD})$x8	—	sec
t_{CY}	Cycle Time	—	f_{SYS}=1.0MHz	—	4.0	—	μs
t_{RES}	Reset Pulse Width	—	—	5	—	—	ms
f_{SOUND}	Sound Effect Clock	—	—	—	64 or 128*	—	kHz

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

Functional Description

Program counter – PC

The 12-bit program counter is controlled by PA3 which can change the ROM bank of the program memory. There are two program memory banks which are selected by PA3, each bank is 4KB ROM. The instruction"OUT PA, A" is used to change the value of PA3. Then, low or high 4K ROM is selected accordingly. All instructions are not effective on the crossing bank, unless the value of PA3 is changed in advance.

The 12-bit program counter (PC) controls the sequence in which the instructions stored in the program ROM are executed and its contents specify a max. of 4096 address.

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

When executing the jump instruction (JMP, JNZ, JC, JTMR...), subroutine call, internal interrupt, RTC interrupt or return from subroutine, the PC manipulates the program transfer by loading the address corresponding to each instruction.

Program memory – ROM

The program memory is used to store program instruction which is to be executed. It is organized into 8192×8 bits and addressed by the program counter and PA3.

Certain locations in bank 0 of the program memory are reserved for specific usage:

- Location 0004H

  This area is reserved for TIMER interrupt service program. A timer interrupt results from TIMER overflow, if interrupt is enabled, the CPU begins execution at location 0004H.

Mode	Program Counter												
	PA3	**PC11**	**PC10**	**PC9**	**PC8**	**PC7**	**PC6**	**PC5**	**PC4**	**PC3**	**PC2**	**PC1**	**PC0**
Initial reset	1	0	0	0	0	0	0	0	0	0	0	0	0
Internal interrupt	PA3	0	0	0	0	0	0	0	0	0	1	0	0
External interrupt	PA3	0	0	0	0	0	0	0	0	1	0	0	0
Jump, call instruction	PA3	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Conditional branch	PA3	@	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Return from subroutine	PA3	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

Notes: PC11~PC0: Instruction code bits
S11~S0: Stack register bits
@: PC11 keeps current value
PA3: Bank value bits

- Location 0008H

  This area is reserved for RTC interrupt service program.

- Location 0n00H~0nFFH (n=current number) and 0F00H~0FFFH.

  The last 256 bytes of each page in the program memory, addressed from 0n00H to 0nFFH and 0F00H to 0FFFH can be used as a look–up table. The instructions READ R4A, READ MR0A, READF R4A, READF MR0A can read the table and transfer the contents of the table to ACC and R4 or transfer to ACC and data memory addressed by register pair "R1, R0". These areas may function as a normal program memory depending on the requirements.

  Certain locations in bank 1 of the program memory are reserved for specific usage:

- Location 1000H

  This area is reserved for the initialization program. After reset, the CPU always begins execution at location 1000H.

- Location 1004H

  This area is reserved for TIMER interrupt service program. A timer interrupt results from TIMER overflow, if interrupt is enabled, the CPU begins execution at location 1004H.

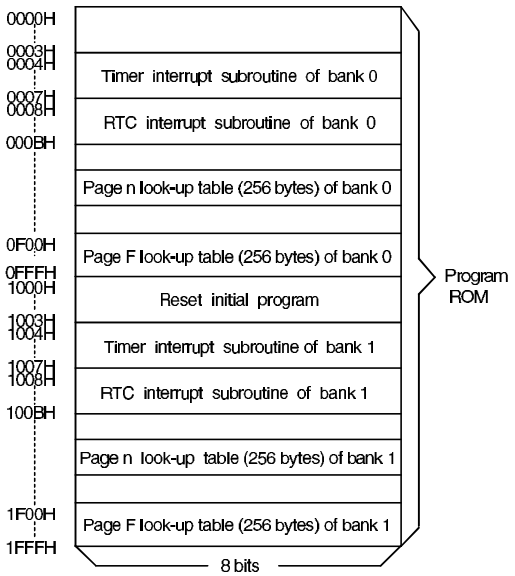
- Location 1008H

  This area is reserved for RTC interrupt service program.

- Location 1n00H~1nFFH (n=current number) and 1F00H~1FFFH.

  The last 256 bytes of each page in the program memory, addressed from 1n00H to 1nFFH and 1F00H to 1FFFH can be used as a look–up table. The instructions READ R4A, READ MR0A, READF R4A, READF MR0A can read the table and transfer the contents of the table to ACC and R4 or transfer to ACC and data memory addressed by register pair "R1, R0". These areas may function as a normal program memory depending on the requirements.

  The program memory (ROM) mapping is shown below:



Program memory

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

Stack register

This is a special group of register which is used to save the contents of the program counter (PC) and is organized with 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 interrupt routine which is signaled by a return instruction, RET or RETI restores the program counter to its previous value from stack register.

Executing "RETI" instruction will restore the carry flag from the stack register, but "RET" does not.

Working registers – R0, R1, R2, R3, R4

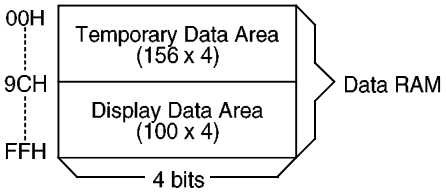
These five registers are usually used to store the frequently accessed data. The working register can be incremented (+1) or decremented (–1). The JNZ Rn,address (n=0,1,4) instruction makes very efficient use of the working register as program loop counter. The register pairs of R1, R0 and R3, R2 can also be used as the data memory pointer, when the data memory transfer instruction is executed.

Data memory – RAM

The data memory is a static RAM organized with 256×4 bits and is used to store temporary data and display data. All of the data memory locations are indirectly addressable through the register pair "R1, R0" or "R3, R2".

There are two areas in the data memory, temporary data area and display data area. Access to temporary data memory is made through 00H~9BH address, and access to display data memory is made in 9CH~FFH address.

When data is written in the display area, the LCD driver automatically reads it and generates an LCD driving signal.



Data memory

Accumulator – ACC

The register ACC plays the most important role in data manipulation and data transfer. It is not only one of the sources of input to the ALU but also the destination of the result due to ALU. Data transfer can be performed between ACC and other registers, data memory or I/O ports.

Arithmetic and logic unit – ALU

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

- Arithmetic operation
  (ADD, ADC, SUB, SBC, DAA)
- Logic operation (AND, OR, XOR)
- Rotation (RL, RR, RLC, RRC)
- Increment and Decrement (INC, DEC)
- Branch decision (JZ, JNZ, JC, JNC...)

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

Timer

This is a programmable 8–bit count-up counter, internal frequency sources used to aid the user in counting and generating accurate time base.

The Timer can be pre-set and read with software instructions. "TIMER XXH", "MOV TMRL, A" and "MOV TMRH, A" preload TIMER value. "MOV A, TMRL" and "MOV A, TMRH" read the contents of TIMER to ACC.

The Timer is stopped by a hardware reset or "TIMER OFF" instruction and started by a TIMER ON instruction.

Once the Timer is started, it will increment to its maximum count (FFH) and overflows to zero (00H). It will not stop until there is a "TIMER OFF" instruction or reset. When an overflow occurs, it will set the Timer Flag (TF) simultaneously. If interrupt is enabled, the Timer circuit supports TF for internal interrupt. The state of the TF can be tested with the conditional instruction JTMR.

The Timer flag is cleared after the interrupt or JTMR instruction is executed.

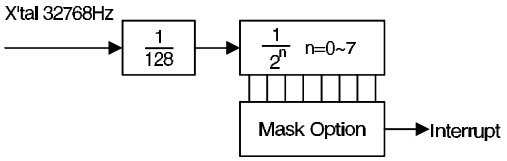
The frequency of the internal frequency source can be selected by mask option.

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

Where n=0,1,2.....13 except 6, by mask option (the sixth stage is reserved for internal use).

RTC

There is a real time clock (RTC) function implemented on the HTG12G0. The RTC function is used to generate an accurate time period. The clock source of the RTC circuit comes from the 32768Hz crystal oscillator. The block diagram is shown as follows:



The output of the RTC can be selected by mask option.

Frequency of the RTC output = $\frac{256}{2^n}$ n=0~7

The RTC output is used to generate an interrupt signal.

Interrupt

The HTG12G0 provides both TIMER and RTC interrupt modes. The DI and EI instructions are used to disable and enable the interrupts. When the RTC is activated during enable interrupt mode and the program is not within a CALL subroutine, this causes a subroutine call to location 8 and reset 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 TIMER interrupt is activated. This cause a subroutine call to location 4 and resets the timer flag. If both TIMER and RTC interrupts arrive at the same time, the RTC 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 behaviors may occur. If within a CALL subroutine both TIMER and RTC interrupt occur, no matter what order they arrive in, the RTC 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.

Initial reset

The HTG12G0 provides a $\overline{\text{RES}}$ pin for system initialization. Since the $\overline{\text{RES}}$ pin has internal pull high resistor, only an external 0.1μ~1μ capacitor is needed. If the reset pulse is generated externally, it must be held low for at least 5 ms.

When $\overline{\text{RES}}$ is active, the internal block will be initialized as shown below:

PA3 and PC	1000H
TIMER	Stop
Timer flag	Reset (low)
SOUND	Sound off and one sing mode
Output port A	High (or floating state)
LCD output	Disabled
BZ and $\overline{\text{BZ}}$ output	High level

HALT

This is a special feature of HTG12G0. It will stop the chip's normal operation and reduces power consumption. When the instruction "HALT" is executed, then

- System oscillator will be stopped
- The contents of the on-chip RAM and registers remain unchanged
- RTC oscillator still keeps running

The system can escape HALT mode by ways of initial reset or RTC interrupt or wake-up from the following entry of program counter value.

- Initial reset: 1000H

  Wake-up: next address of the HALT instruction

  When the halt status is terminated by the RTC interrupt, the following procedure takes place:

  * Case1: If the system is in an interrupt-disable state before entering the halt state:

- The system will be awakened and returns to the main program instruction following the HALT command.
- The RTC interrupt will be held until the system receives an enable interrupt command by which the RTC interrupt will be serviced.

* Case 2: If the system is in an interrupt enable state:
  - The RTC interrupt will awake the system and execute the RTC interrupt subroutine.
  - In the HALT mode, each bit of ports PP, PS, can be used as wake-up signal by mask option to wake-up the system. This signal is active in low-going transition.

Sound effect

HTG12G0 provides sound effect circuit which offers up to 16 sounds with 3 effects of tone, boom and noise. Holtek supports a sound library which have melody, alarm, machine gun shooting etc.

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

There are two singing modes, SONE mode and SLOOP mode, which is activated by "SOUND ONE" and "SOUND LOOP". In SONE mode, the sound that has been specified plays just once. In SLOOP mode, the sound being specified keeps playing repeatedly.

Since sound 0~11 contain 32 notes, sound 12~15 contain 64 notes, the later possess 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 only supports sound clock frequency 128K or 64K. To utilize Holtek's sound library, select the proper system clock and mask option.

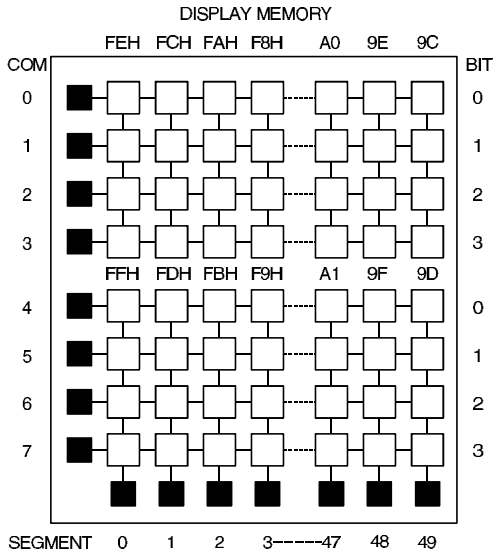
LCD display memory

The LCD display memory is embedded in the data memory. It can be read and written to as normal data memory.

The figure shows the mapping between display memory and LCD pattern.

To turn on/off the display, the programmer just writes 1/0 to the corresponding bit of the display memory.

The LCD display module may have any form as long as the number of the common is no more than 8 and the segment is no more than 50.



LCD display memory

LCD driver output

The output number of the LCD driver is 50×8, directly driving a 1/8 duty cycle and 1/5 bias LCD. All LCD segments are at random during initial clear mode.

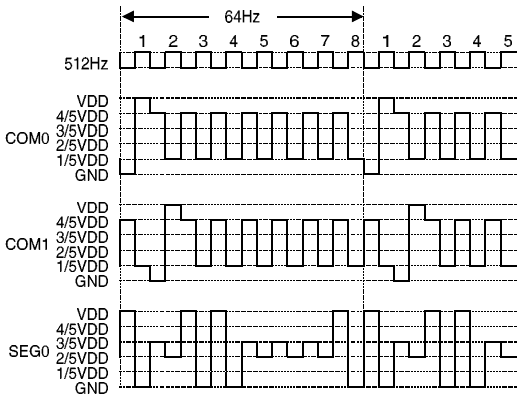
The LCD has built-in bias voltage circuit. No external resistor is needed.

The frequency of the LCD driving clock is fixed at about 512Hz. This is set by RTC OSC (32.768kHz).

LCD driver output can be enabled or disabled by setting PA2 without the influence of the related memory condition.

LCD driver output is enabled by setting PA2 as "1", and disabled by setting PA2 as "0".

An example of an LCD driving waveform (1/8 duty and 1/5 bias) is shown below:

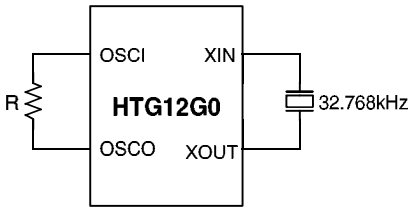


Oscillator circuit

Only one external resistor is required for the HTG12G0 system clock.

The system clock is also used as the sound effect clock, or internal frequency source of TIMER. Another crystal oscillator is needed to be used as the reference signal of LCD driving clock and RTC interrupt clock source.

The HTG12G0 machine cycle consists of a sequence of 4 states numbered T1 to T4. Each state lasts for one oscillator period. The machine cycle is 4.0μs, if the system frequency is up to 1.0MHz.



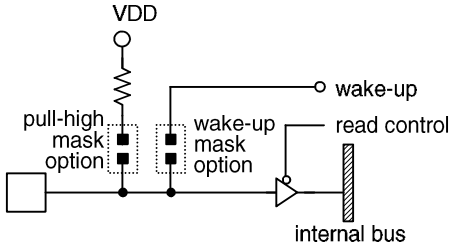
RC and RTC oscillator

Input ports – PS, PP

Ports PS, PP are 4-bit input ports. These input ports are configured as shown below:

All of these ports have internal pull-high resistor determined by mask option.

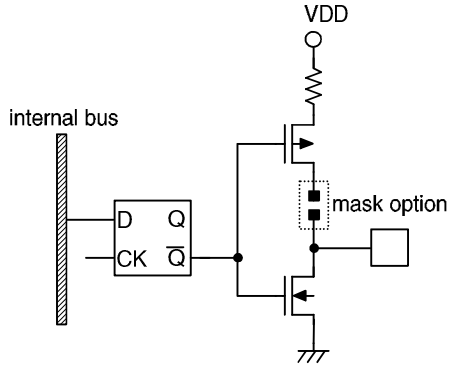
Each bit of input ports PS, PP can be a trigger source of HALT interrupt. That is also specified by mask option. A transition from high to low will make HTG12G0 wake-up.



Input ports PS, PP

Output port – PA

Port A is a 2-bit output port (PA0~PA1), and configured as shown below:



Output port PA

The mask option available for selecting the output configuration is either normal CMOS output type or open drain NMOS output type. At the initial clear mode, the output ports are at high state (in CMOS output type) or at floating state (in NMOS output type).

Note:

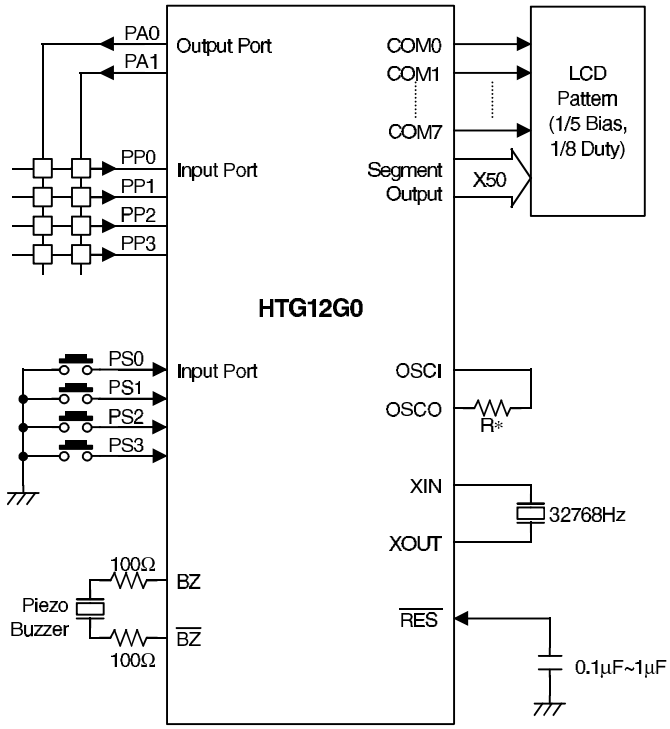
PA3 controls the bit 12 of the program counter. When the instruction "OUT PA, A" is operated, port A is changed as well. PA2 controls the ON/OFF state of the LCD. Without the influence of the memory condition, "1" turns the LCD on, and "0" off.

Mask option

HTG12G0 provides six kinds of mask option for different applications.

- Each bit of input ports PS, PP with pull-high resistor
- Each bit of input ports PS, PP function as HALT wake-up trigger
- Each bit of output port PA0~PA1 with CMOS or open drain NMOS
- 8 bit programmable TIMER with internal frequency sources. There are 13 (the sixth stage is reserved for internal use) internal frequency sources which can be selected as a clocking signal.
- Six kinds of sound clock frequencies: $f_{SYS}/2^m$, m= 0, 1, 2, 3, 4, 5
- There are eight kinds of RTC interrupt frequencies. RTC interrupt frequency= $256/2^n$Hz, n=0~7

Application Circuits



R*: depends on the required system clock frequency (R=51kΩ~620kΩ, at V_{DD}=3V)

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 and Decrement				
INC A	Increment ACC	1	1	—
INC Rn	Increment register	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	1	1	—
DEC [R1R0]	Decrement data memory	1	1	—
DEC [R3R2]	Decrement data memory	1	1	—
Data Move				
MOV A,Rn	Move register to ACC	1	1	—
MOV Rn,A	Move ACC to register	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	—

Mnemonic	Description	Byte	Cycle	CF
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=PS, PP	1	1	—
OUT PA,A	Output ACC to port-A	1	1	—
Branch				
JMP addr	Jump unconditional	2	2	—
JC addr	Jump on carry=1	2	2	—
JNC addr	Jump on carry=0	2	2	—
JTMR addr	Jump on timer out	2	2	—
JAn addr	Jump on ACC bit n=1, n=0,1,2,3	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	√
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 to start counting	1	1	—
TIMER OFF	Set TIMER to 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 high 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	—

Mnemonic	Description	Byte	Cycle	CF
Sound Control				
SOUND n	Active SOUND channel n	2	2	—
SOUND A	Active SOUND channel with Accumulator	1	1	—
SOUND ONE	Turn on SOUND one mode	1	1	—
SOUND LOOP	Turn on SOUND repeat mode	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 logical AND 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 logical AND 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 logical AND 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 0
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 1.

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 1. 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 1. 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 1. 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 1. 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 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 0 0 1 1 0 0 1 1 PS
0 0 1 1 0 1 0 0 PP

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

Operation ACC ← Pi; Pi=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 1. 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 1. 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 1. 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 1. Carry flag is not affected.
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
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 1.
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
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 1.
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
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
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 0.
Operation	PC (bit 0~10) ← address, if CF=0 PC ← PC+2, if CF=1
JNZ A,address	Jump if accumulator is not 0
Machine code	1 0 1 1 1 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 0.
Operation	PC (bit 0~10) ← address, if ACC≠0 PC ← PC+2, if ACC=0

JNZ Rn,address Jump if register is not 0

Machine code
1 0 1 0 0 a a a a a a a a a a a R0
1 0 1 0 1 a a a a a a a a a a a R1
1 1 0 1 1 a a a a a a a a a a a R4

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 0.

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 1.

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

JZ A,address Jump if accumulator is 0

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 0.

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 logical OR 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 logical OR 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 logical OR 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 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 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

Description The program counter bits 0~11 are restored from the stack. The carry flag is restored before entering the interrupt service routine.

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 by 1 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 by 1 bit. Bit 3 replaces the carry bit, which is rotated into the bit 0 position.

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

RR A Rotate accumulator right

Machine code 0 0 0 0 0 0 0 0

Description The contents of the accumulator are rotated right by 1 bit. Bit 0 is rotated to both bit 3 and the carry flag.

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

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 by 1 bit. Bit 0 replaces the carry bit, which is rotated into the bit 3 position.

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

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 ← 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+$\overline{M(R1,R0)}$+1

TIMER OFF Set timer to 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 to 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-OR 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-OR 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 logical Exclusive-OR with the accumulator.
Operation	M(R1,R0) ← M(R1,R0) "XOR" ACC

Holtek Semiconductor Inc. (Headquarters)
No.3 Creation Rd. II, Science-based Industrial Park, Hsinchu, Taiwan, R.O.C.
Tel: 886-3-563-1999
Fax: 886-3-563-1189

Holtek Semiconductor Inc. (Taipei Office)
5F, No.576, Sec.7 Chung Hsiao E. Rd., Taipei, Taiwan, R.O.C.
Tel: 886-2-2782-9635
Fax: 886-2-2782-9636
Fax: 886-2-2782-7128 (International sales hotline)

Holtek Microelectronics Enterprises Ltd.
RM.711, Tower 2, Cheung Sha Wan Plaza, 833 Cheung Sha Wan Rd., Kowloon, Hong Kong
Tel: 852-2-745-8288
Fax: 852-2-742-8657