HTG13Q0/HTG13A0
4-bit Microcontroller

Features

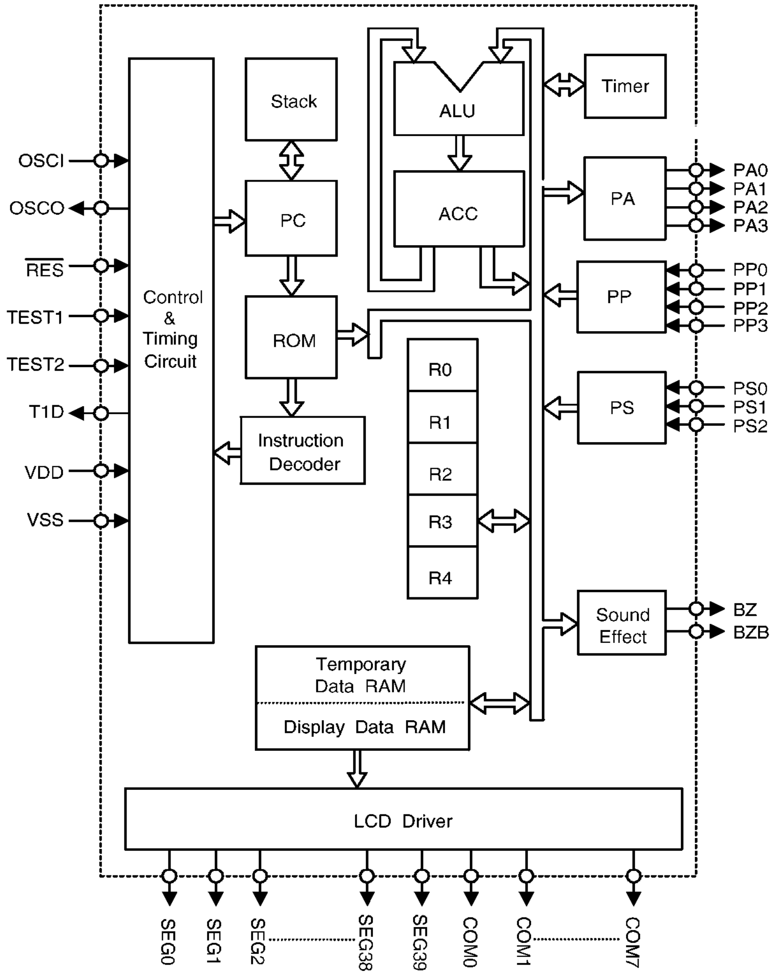
- Operating voltage: 2.4V~3.3V
- 7 input lines
- 4 output lines
- RC oscillator for system clock
- 4K×8 program ROM
- 160×4 data RAM
- 40×8 segment LCD driver, 1/5 bias, 1/8 duty
- 5 working registers
- 8-bit programmable timer with built-in frequency source
- Internal timer overflow interrupt
- 16 kinds of programmable sound effect
- Halt function and wake up reduces power consumption
- 96 powerful instructions
- Up to 4.0μ sec instruction cycle (1.0MHz system clock), at V_{DD}=3V
- One-level subroutine nesting
- 8-bit table read instruction
- Halt instruction

General Description

The HTG13Q0/HTG13A0 are two processors from HOLTEK's 4-bit stand alone single chip microcontroller specially designed for the application of LCD display products. These two devices are similar in most ways apart from chip size and some electronic characteristics.

They are especially suited for applications requiring low power consumption system with many LCD segments, such as calculator, scale, subsystem controller, hand-held LCD products and electronic appliances.

Block Diagram



Note:

ACC: Accumulator
PC: Program Counter
R0~R4: Working Register

PA: Output Port
PP, PS0~PS2: Input Ports

Pad Description

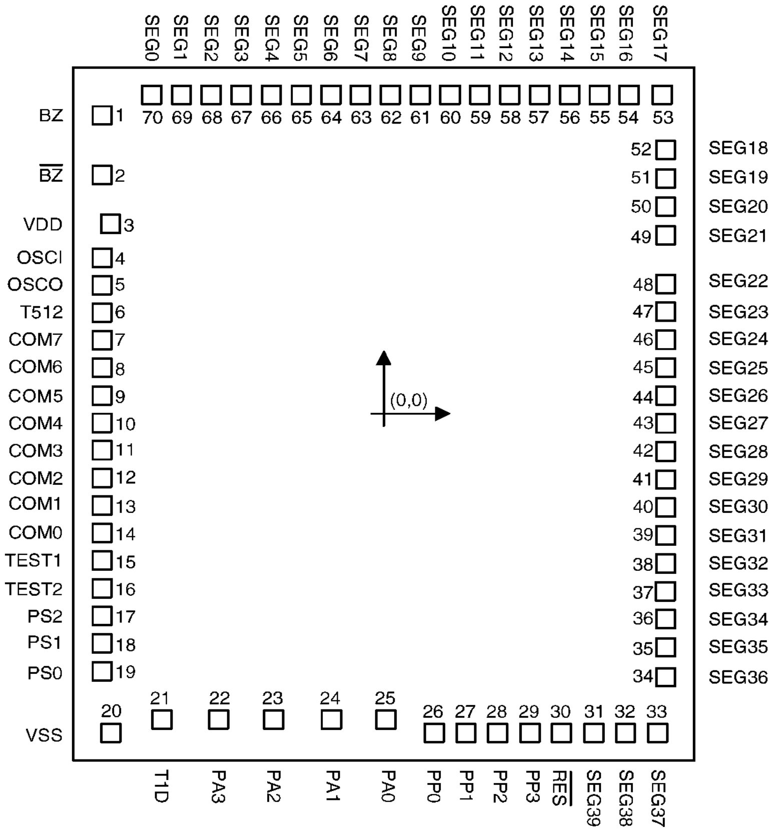
Pad No.	Pad name	I/O	Internal Connection	Description
1 2	BZ $\overline{\text{BZ}}$	O	*	Sound effect output
3	VDD	I	—	Positive power supply
4 5	OSCI OSCO	I O	—	OSCI, OSCO are connected to resistor for internal system clock.
6 15 16 21	T512 TEST1 TEST2 T1D	O I I O	—	For test mode only TEST1 and TEST2 must let it open when the chip is in normal operation (with an internal pull high resistor).
7~14	COM7~COM0	O	—	Output for LCD panel common plate
17~19	PS2~PS0	I	Pull-High or None **	3-bit port for input only
20	VSS	I	—	Negative power supply, GND.
22~25	PA3~PA0	O	CMOS or NMOS Open Drain	4-bit latch port for output only
26~29	PP0~PP3	I	Pull-High or None **	4-bit port for input only
30	$\overline{\text{RES}}$	I	—	Input for reset LSI inside. Reset is active at logical low level.
31~70	SEG39~SEG0	O	—	LCD driver outputs for LCD panel segment

*: 6 internal sources deriving from 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 PS0~PS2, PP can be a trigger source of HALT interrupt. That can be specified by mask option.

Pad Position

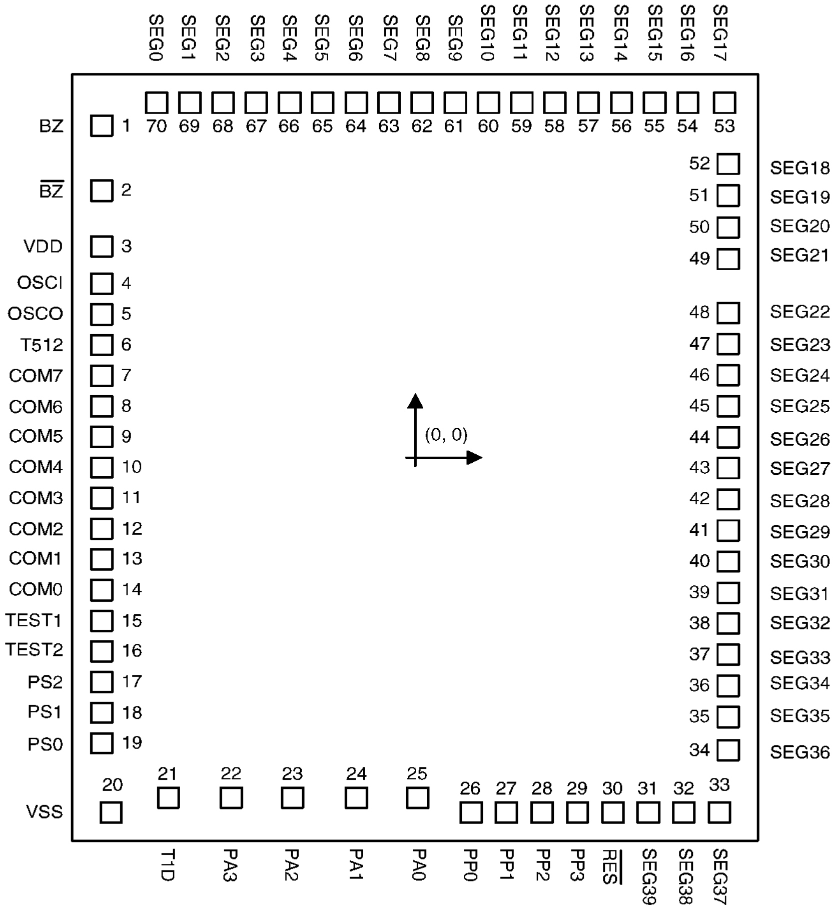
HTG13Q0



Chip size: 2710 × 3228 $(\mu m)^2$

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

HTG13A0



Chip size: 2990 × 3620 (μm)^2

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

Pad Coordinates

HTG13Q0 Unit: μm

Pad No.	X	Y
1	–1232.10	1365.75
2	–1232.10	1093.95
3	–1196.10	868.50
4	–1232.10	715.50
5	–1232.10	589.50
6	–1232.10	463.50
7	–1232.10	337.50
8	–1232.10	211.50
9	–1232.10	85.50
10	–1232.10	–40.50
11	–1232.10	–166.50
12	–1232.10	–292.50
13	–1232.10	–418.50
14	–1232.10	–544.50
15	–1232.10	–670.50
16	–1232.10	–796.50
17	–1232.10	–922.50
18	–1232.10	–1048.50
19	–1232.10	–1174.50
20	–1194.30	–1458.90
21	–970.20	–1398.15
22	–721.80	–1398.15
23	–481.50	–1398.15
24	–229.50	–1398.15
25	10.80	–1398.15
26	220.05	–1458.90
27	359.55	–1458.90
28	499.05	–1458.90
29	638.55	–1458.90
30	778.05	–1458.90
31	917.55	–1458.90
32	1057.05	–1458.90
33	1196.55	–1458.90
34	1232.10	–1202.40
35	1232.10	–1066.05
36	1232.10	–938.25
37	1231.65	–810.45
38	1232.10	–682.65
39	1232.10	–554.85
40	1232.10	–427.05
41	1232.10	–299.25
42	1232.10	–171.45
43	1232.10	–43.65
44	1232.10	84.15
45	1232.10	211.95
46	1232.10	339.75
47	1232.10	467.55
48	1232.10	595.35
49	1232.10	816.75
50	1232.10	947.25
51	1232.10	1077.75
52	1232.10	1208.25
53	1216.80	1459.80
54	1072.80	1459.80
55	942.30	1459.80
56	811.80	1459.80
57	681.30	1459.80
58	550.80	1459.80
59	420.30	1459.80
60	289.80	1459.80
61	159.30	1459.80
62	28.80	1459.80
63	–101.70	1459.80
64	–232.20	1459.80
65	–362.70	1459.80
66	–493.20	1459.80
67	–623.70	1459.80
68	–754.20	1459.80
69	–884.70	1459.80
70	–1015.20	1459.80

HTG13A0 Unit: µm

Pad No.	X	Y	Pad No.	X	Y
1	–1369.00	1517.00	36	1369.00	–1043.00
2	–1369.00	1220.50	37	1368.50	–901.00
3	–1369.00	964.50	38	1369.00	–759.00
4	–1369.00	794.50	39	1369.00	–617.00
5	–1369.00	654.50	40	1369.00	–475.00
6	–1369.00	514.50	41	1369.00	–333.00
7	–1369.00	374.50	42	1369.00	–191.00
8	–1369.00	234.50	43	1369.00	–49.00
9	–1369.00	94.50	44	1369.00	93.00
10	–1369.00	–45.50	45	1369.00	235.00
11	–1369.00	–185.50	46	1369.00	377.00
12	–1369.00	–325.50	47	1369.00	519.00
13	–1369.00	–465.50	48	1369.00	661.00
14	–1369.00	–605.50	49	1369.00	907.00
15	–1369.00	–745.50	50	1369.00	1052.00
16	–1369.00	–885.50	51	1369.00	1197.00
17	–1369.00	–1025.50	52	1369.00	1342.00
18	–1369.00	–1165.50	53	1352.00	1621.50
19	–1369.00	–1305.50	54	1192.00	1621.50
20	–1327.00	–1621.50	55	1047.00	1621.50
21	–1078.00	–1554.00	56	902.00	1621.50
22	–802.00	–1554.00	57	757.00	1621.50
23	–535.00	–1554.00	58	612.00	1621.50
24	–255.00	–1554.00	59	467.00	1621.50
25	12.00	–1554.00	60	322.00	1621.50
26	244.50	–1621.50	61	177.00	1621.50
27	399.50	–1621.50	62	32.00	1621.50
28	554.50	–1621.50	63	–113.00	1621.50
29	709.50	–1621.50	64	–258.00	1621.50
30	864.50	–1621.50	65	–403.00	1621.50
31	1019.50	–1621.50	66	–548.00	1621.50
32	1174.50	–1621.50	67	–693.00	1621.50
33	1329.50	–1621.50	68	–838.00	1621.50
34	1369.00	–1336.50	69	–983.00	1621.50
35	1369.00	–1185.00	70	–1128.00	1621.50

Absolute Maximum Ratings*

Supply Voltage .... –0.3V to 5.5V	Storage Temperature................. –50°C to 125°C
Input Voltage .... V_{SS}–0.3V to V_{DD}+0.3V	Operating Temperature .... 0°C to 70°C

*Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. These are stress ratings only. Functional operation of this device at these or any other conditions above those indicated in the operational sections of this specification is not implied and exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

HTG13Q0 (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	—	200	500	µA
I_{STB}	Standby Current	3V	System halt	—	—	1	µA
V_{IL1}	Input Low Voltage PS, PP	3V	—	0	—	0.6	V
V_{IH1}	Input High Voltage PS, PP	3V	—	2.1	—	3.0	V
V_{IL2}	Input Low Voltage $\overline{RES}$	3V	—	0	—	0.6	V
V_{IH2}	Input High Voltage $\overline{RES}$	3V	—	2.6	—	3.0	V
I_{OL1}	Port A & BZ & $\overline{BZ}$ Output Sink Current	3V	V_{DD}=3V, V_{OL}=0.3V	1.5	3.0	—	mA
I_{OH1}	Port A & BZ & $\overline{BZ}$ Output Source Current	3V	V_{DD}=3V, V_{OH}=2.7V	–0.8	–1.5	—	mA
I_{OL2}	Segment 0~17 Output Sink Current	3V	V_{LCD}=3V, V_{OL}=0.3V	60	120	—	µA
I_{OH2}	Segment 0~17 Output Source Current	3V	V_{LCD}=3V, V_{OH}=2.7V	–40	–60	—	µA
I_{OL3}	Segment 18~39 Output Sink Current	3V	V_{LCD}=3V, V_{OL}=0.3V	40	60	—	µA
I_{OH3}	Segment 18~39 Output Source Current	3V	V_{LCD}=3V, V_{OH}=2.7V	–20	–30	—	µA
I_{OL4}	Common Sink Current	3V	V_{LCD}=3V, V_{OL}=0.3V	60	120	—	µA
I_{OH4}	Common Source Current	3V	V_{LCD}=3V, V_{OH}=2.7V	–60	–120	—	µA
R_{PH}	Pull-High Resistance	3V	PS0~PS2, PP, $\overline{RES}$	50	—	300	kΩ

HTG13A0 (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	—	300	500	µA
I_{STB}	Standby Current	3V	System halt	—	—	1	µA
V_{IL1}	Input Low Voltage PS, PP	3V	—	0	—	0.6	V
V_{IH1}	Input High Voltage PS, PP	3V	—	2.1	—	3.0	V
V_{IL2}	Input Low Voltage $\overline{RES}$	3V	—	0	—	0.6	V
V_{IH2}	Input High Voltage $\overline{RES}$	3V	—	2.6	—	3.0	V
I_{OL1}	Port A & BZ & $\overline{BZ}$ Output Sink Current	3V	V_{DD}=3V, V_{OL}=0.3V	1.5	3.0	—	mA
I_{OH1}	Port A & BZ & $\overline{BZ}$ Output Source Current	3V	V_{DD}=3V, V_{OH}=2.7V	–0.8	–1.5	—	mA
I_{OL2}	Segment 0~17 Output Sink Current	3V	V_{LCD}=3V, V_{OL}=0.3V	60	120	—	µA
I_{OH2}	Segment 0~17 Output Source Current	3V	V_{LCD}=3V, V_{OH}=2.7V	–40	–60	—	µA
I_{OL3}	Segment 18~39 Output Sink Current	3V	V_{LCD}=3V, V_{OL}=0.3V	40	60	—	µA
I_{OH3}	Segment 18~39 Output Source Current	3V	V_{LCD}=3V, V_{OH}=2.7V	–20	–40	—	µA
I_{OL4}	Common Sink Current	3V	V_{LCD}=3V, V_{OL}=0.3V	60	120	—	µA
I_{OH4}	Common Source Current	3V	V_{LCD}=3V, V_{OH}=2.7V	–60	–120	—	µA
R_{PH}	Pull-High Resistance	3V	PS0~PS2, PP, $\overline{RES}$	50	—	300	kΩ

A.C. Characteristics

HTG13Q0

(Ta=25°C)

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SYS}	System Clock	3V	R:680k~15kΩ	32	—	1000	kHz
f_{LCD}	LCD Clock	3V	—	—	512*	—	Hz
t_{COM}	LCD Common Period	—	1/8 duty	—	$(1/f_{LCD})\times 8$	—	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

*: In general,, f_{LCD} is selected and optimized by HOLTEK according to f_{SYS} and operating voltage.

**: Only these two frequency of clock signal is supported by HOLTEK sound library.

HTG13A0

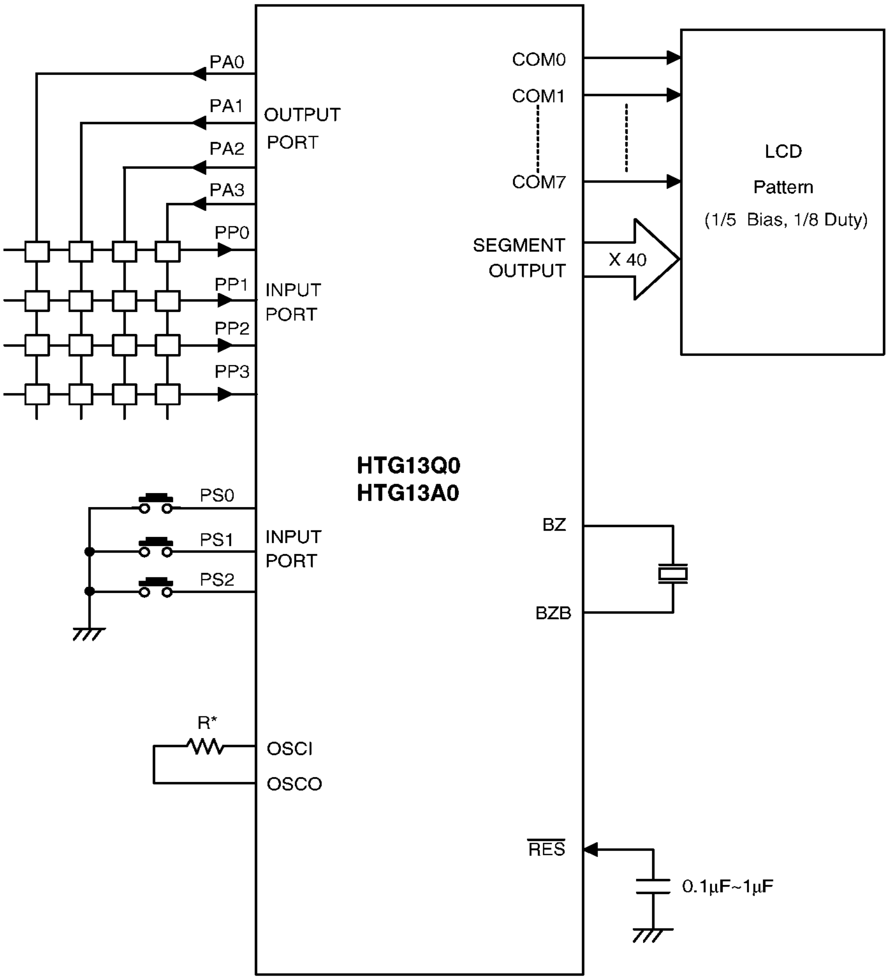
(Ta=25°C)

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SYS}	System Clock	3V	R:680k~15kΩ	32	—	1000	kHz
f_{LCD}	LCD Clock	3V	—	—	512*	—	Hz
t_{COM}	LCD Common Period	—	1/8 duty	—	$(1/f_{LCD})\times 8$	—	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

*: In general,, f_{LCD} is selected and optimized by HOLTEK according to f_{SYS} and operating voltage.

**: Only these two frequency of clock signal is supported by HOLTEK sound library.

Application Circuit



R*: Depends on the required frequency of system clock. (R=680k~15kΩ, at V_{DD}=3V)

System Architecture

Program counter – PC

The 12-bit program counter (PC) controls the sequence in which the instructions stored in program ROM are executed and its contents specify maximum 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, external 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 with 4096 × 8 bits

- Location 000H
  This area are reserved for the initialization program. After reset, the CPU always begins execution at location 000H.
- Location 0004H
  This area are reserved for TIMER interrupt service program. A timer interrupt resulting from TIMER overflow, if interrupt is enabled, the CPU begins execution at location 0004H.

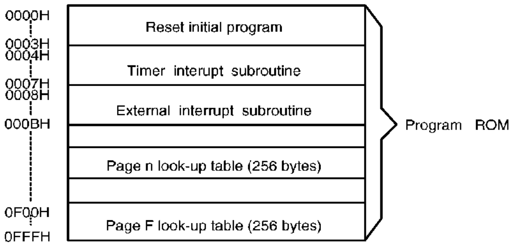
Mode	Program Counter											
	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial reset	0	0	0	0	0	0	0	0	0	0	0	0
Internal interrupt	0	0	0	0	0	0	0	0	0	1	0	0
External interrupt	0	0	0	0	0	0	0	0	1	0	0	0
Jump, call instruction	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Conditional branch	@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

Notes:

PC11~PC0	:	Bits of Instruction Code.
@	:	PC11 Keeps Current Value.
S11~S0	:	Bits of Stack Register.

- Location 0008H
  Activating the PS or PP input pins of the processor with the interrupts enabled during Halt mode causes the program to jump to this location.
- Location 0n00H~0nFFH (n=current number) and 0F00H~0FFFH.
  The last 256 bytes of each page in 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 area may function as normal program memory depending on user's requirement. Note that the page number n must be greater than zero, some locations in page 0 are reserved for specific usage as mentioned.

The program memory (ROM) mapping is shown below:


0000H
0003H
0004H
0007H
0008H
000BH
0F00H
0FFFH
Reset initial program
Timer interupt subroutine
External interrupt subroutine
Page n look-up table (256 bytes)
Page F look-up table (256 bytes)
Program ROM

Program memory

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 or RETI), the contents of the stack register are returned to the PC.

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

Working registers – R0,R1,R2,R3,R4

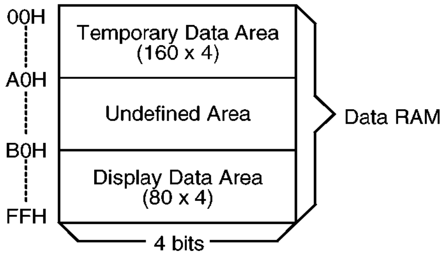
These five registers are usually used to store the frequently accessed data. The working register can be incremented (+1) or decremented (–1). The JNZ Rn,address (n=0,1,4) instruction makes very efficient use of the working register as program loop counter. Also the register pairs of R1, R0 and R3, R2 can 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 bit format and is used to store temporary data & 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 done 00H–9FH address, and access to display data memory is done in B0H–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 & 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 to aid the user in counting and generate accurate time base.

The Timer is presettable and readable 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 counter 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 overflow to zero (00H) and will not stop until a “TIMER OFF” instruction or reset. When the 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 is also testable with conditional instruction JTMR.

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

The frequency of 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).

Interrupt

The HTG13Q0/HTG13A0 provide both internal and external interrupt modes. The DI and EI instructions are used to disable and enable the interrupts. During halt mode, if the PP or PS input 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.

When running under a CALL subroutine or DI the inerrupt 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 behaviours may occur. If within a CALL subroutine interrupt occurs, the interrupt will be serviced after leaving the CALL subroutine.

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

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

Initial reset

The HTG13Q0/HTG13A0 provide 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 below:

PC	000H
TIMER	Stop
Timer flag, Carry flag	Reset (low)
SOUND	Sound off and One sing mode
Output Port A	high (or floating state)
Interrupt	Disable
BZ and $\overline{\text{BZ}}$ output	Low level

Halt

This is a special feature of HTG13Q0 /HTG13A0. It will stop the chip's normal operation and reduce power consumption. When the instruction "HALT" is executed, then

- The system clock will be stopped
- The contents of the on-chip RAM and registers remain unchanged
- LCD segments and commons keep VDD voltage (i.e. LCD becomes blank)

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

Initial reset: 000H.
Interrupt (enabled): 008H
Interrupt (disabled): next address of HALT instruction.

In HALT mode, each bit of port PP, PS0~PS2, can be used as external interrupt by mask option to wake-up system. This signal is active in low-going transition.

Sound effect

HTG13Q0/HTG13A0 provide 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, shooting of machine gun etc. That can meet user's requirement.

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

There are two singing mode, SONE mode and SLOOP mode, this 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 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.

The HOLTEK'S sound library only supports sound clock frequency 128K or 64K. If user wants to utilize HOLTEK'S sound library, please select the proper system clock and mask option.

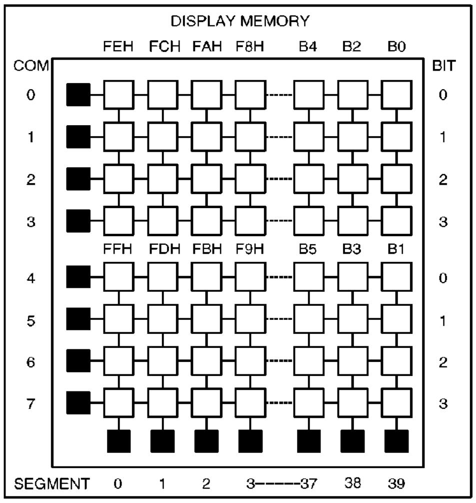
LCD display memory

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

The following 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 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 40.



LCD display memory

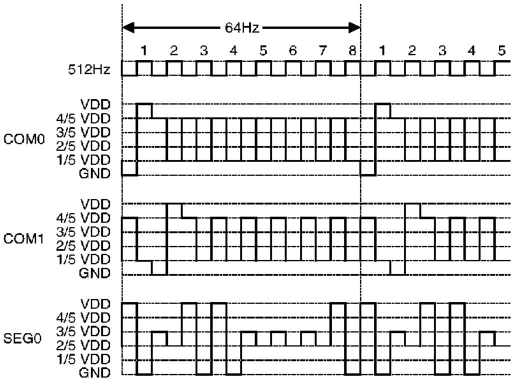
LCD driver output

The output number of the LCD driver is 40 × 8. That can directly drive a LCD being 1/8 duty cycle and 1/5 bias. All LCD segments are random at the initial clear mode.

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

The frequency of LCD driving clock shall be fixed in about 512Hz. That can not be selected by the user, and HOLTEK will set it according to the application.

An example of a LCD driving waveform (1/8 duty & 1/5 bias) is shown below.



Oscillator circuit

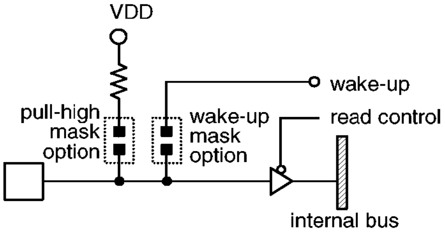
Only one external resistor is needed for HTG13Q0/HTG13A0 oscillator circuit.

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

The one HTG13Q0/HTG13A0 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.

Input ports – PS0~PS2, PP

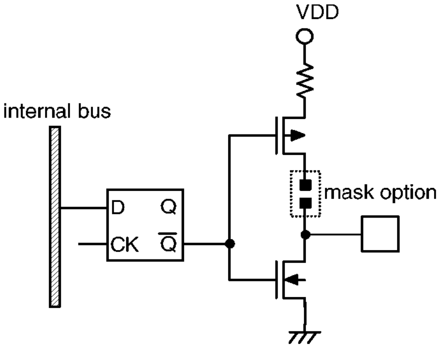
All ports can have internal pull high resistors determined by mask option. Every bit of the input ports PP and PS can be specified to be a trigger source to wake up the HALT interrupt by mask option. A high to low transition on one of these pins will wake up the device from a HALT status.



Input ports – PS0~PS2, PP

Output port – PA

A mask option is available to select whether the output is a CMOS or open drain NMOS type. After an initial clear the output port PA defaults to be high for CMOS or floating for NMOS.



Output port – PA

Mask option

HTG13Q0/HTG13A0 provide 6 kinds of mask option to different applications.

Each bit of input ports PS0~PS2, PP with pull-high resistor

Each bit of input ports PS0~PS2, PP function as HALT interrupt trigger

Each bit of output port PA 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 clocking signal.

Six kinds of sound clock frequency: $f_{SYS}/2^{m}$, m=0, 1, 2, 3, 4, 5

Instruction Set Summary

Mnemonic	Description	Byte	Cycle	CF
Arithmetic				
ADD A,[R1R0] ADC A,[R1R0] SUB A,[R1R0] SBC A,[R1R0] ADD A,XH SUB A,XH DAA	Add data memory to ACC Add data memory with carry to ACC Subtract data memory from ACC Subtract data memory from ACC with borrow Add immediate data to ACC Subtract immediate data from ACC Decimal adjust ACC for addition	1 1 1 1 2 2 1	1 1 1 1 2 2 1	√ √ √ √ √ √ √
Logic Operation				
AND A,[R1R0] OR A,[R1R0] XOR A,[R1R0] AND [R1R0],A OR [R1R0],A XOR [R1R0],A AND A,XH OR A,XH XOR A,XH	AND data memory to ACC OR data memory to ACC Exclusive-OR data memory to ACC AND ACC to data memory OR ACC to data memory Exclusive–OR ACC to data memory AND immediate data to ACC OR immediate data to ACC Exclusive–OR immediate data to ACC	1 1 1 1 1 1 2 2 2	1 1 1 1 1 1 2 2 2	— — — — — — — — —
Increment & Decrement				
INC A INC Rn INC [R1R0] INC [R3R2] DEC A DEC Rn DEC [R1R0] DEC [R3R2]	Increment ACC Increment register Increment data memory Increment data memory Decrement ACC Decrement register Decrement data memory Decrement data memory	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	— — — — — — — —
Data Move				
MOV A,Rn MOV Rn,A MOV A,[R1R0] MOV A,[R3R2] MOV [R1R0],A MOV [R3R2],A MOV A,XH MOV R1R0,XXH MOV R3R2,XXH MOV R4,XH	Move register to ACC Move ACC to register Move data memory to ACC Move data memory to ACC Move ACC to data memory Move ACC to data memory Move immediate data to ACC Move immediate data to R1 and R0 Move immediate data to R3 and R2 Move immediate data to R4	1 1 1 1 1 1 1 2 2 2	1 1 1 1 1 1 1 2 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=PP, PS	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 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 high nibble of TIMER	1	1	—

Mnemonic	Description	Byte	Cycle	CF
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	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 content and carry to accumulator

Machine Code 0 0 0 0 1 0 0 0

Description The content 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)+C

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 content to accumulator

Machine Code 0 0 0 0 1 0 0 1

Description The content 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 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. The program counter is then loaded from the directly-specified address.

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 C ← 0

DAA Decimal–Adjust accumulator

Machine Code 0 0 1 1 0 1 1 0

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

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

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 registor 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)+1

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 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,R1,R2,R3,R4 for nnn=0,1,2,3,4

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 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 a

Description Bits 0–10 of the program counter are replaced with the directly–specified address, bit 11 of the program counter and PA3 of memory bank remain, if accumulator bit n is set to one.

Operation PC (bit 0–10) ← address, if ACC bit n=1(n=0,1,2,3,)
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, bit 11 of the program counter and PA3 of memory bank remain, if the C (Carry flag) is set to one.

Operation PC (bit 0–10) ← address, if C=1
PC ← PC+2, if C=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, bit 11 of the program counter and PA3 of memory bank remain, if the C (Carry flag) is set to zero.

Operation PC (bit 0–10) ← address, if C=0
PC ← PC+2, if C=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, bit 11 of the program counter and PA3 of memory bank remain, 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 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, bit 11 of the program counter and PA3 of memory bank remain, 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, bit 11 of the program counter and PA3 of memory bank remain, 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, bit 11 of the program counter and PA3 of memory bank remain, 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,R1,R2,R3,R4, for nnn=0,1,2,3,4

MOV A,TMRH	Move timer to accumulator
Machine Code	0 0 1 1 1 0 1 1
Description	The high nibble data of Timer counter is loaded to the accumulator.
Operation	ACC ← TIMER (high nibble)

MOV A,TMRL	Move timer to accumulator
Machine Code	0 0 1 1 1 0 1 0
Description	The low nibble data of 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 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 code are loaded to the working registers R1 and R0, the high nibble of the data is loaded to the R1, and the low nibble of the data is loaded to the 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 code are loaded to the working register R3 and R2, the high nibble of the data is loaded to the R3, and the low nibble of the data is loaded to the 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 code are 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,R1,R2,R3,R4, for nnn=0,1,2,3,4

MOV TMRH,A Move accumulator to timer

Machine Code 0 0 1 1 1 1 0 1

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

Operation TIMER (high nibble) ← ACC

MOV TMRL,A Move accumulator to timer

Machine Code 0 0 1 1 1 1 0 0

Description The contents of accumulator is loaded to the low nibble of 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 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 Logical 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 PA

Description The data in the accumulator is transferred to the port-A 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–bits of ROM code (current page) addressed by ACC and R4 are moved to the data memory M(R1,R0) and 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 accumulator. The address of ROM code are specified as below :
Current page → ROM code address bit 12–8
ACC → ROM code address bit 7–4
R4 → ROM code address bit 3–0

Operation M(R1R0) ← 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–bits of ROM code (current page) addressed by ACC and M(R1,R0) are moved to the working register R4 and accumulator. The high nibble of the ROM code is loaded to R4 and the low nibble of the ROM code is loaded to accumulator. The address of ROM code are specified below:
Current page → ROM code address bit 12–8
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 of ROM code (page F) addressed by ACC and R4 are moved to the data memory M(R1,R0) and accumulator. The high nibble of the ROM code is loaded to M(R1,R0) and the low nibble of the ROM coed is loaded to accumulator.
page F → ROM code address bit 12–8 are "PA3 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 of ROM code (page F) addressed by ACC and M(R1,R0) are moved to the working register R4 and accumulator. The high nibble of the ROM code is loaded to R4 and the low nibble of the ROM code is loaded to accumulator.
page F → ROM code address bit 12–8 are "PA3 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 interrupt service routine is restored.

Operation PC ← Stack
C ← C (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 bit 0 and carry flag.

Operation An+1 ← An; An: accumulator bit n (n=0,1,2)
A0 ← A3
C ← 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; the carry bit is rotated into the bit 0 position.

Operation An+1 ← An; An: Accumulator bit n (n=0,1,2)
A0 ← C
C ← 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 one bit. Bit 0 is rotated to bit 3 and carry flag.

Operation An ← An+1; An: Accumulator bit n (n=0,1,2)
A3 ← A0
C ← 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 one bit. Bit 0 replaces the carry bit; the carry bit is rotated into the bit 3 position.

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

SBC A,[R1R0] Subtract data memory content and carry from ACC

Machine Code 0 0 0 0 1 0 1 0

Description The content of the data memory addressed by the register pair "R1,R0" and the carry are subtracted from the accumulator. Carry is affected.

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

SOUND A Active 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 mode

Machine Code 0 1 0 0 1 0 0 1

Description The activated sound plays repeatly.

SOUND OFF Turn off sound

Machine Code 0 1 0 0 1 0 1 0

Description The singing sound will terminate immediately.

SOUND ONE Turn on sound one mode

Machine Code 0 1 0 0 1 0 0 0

Description The activated sound plays only one time.

SOUND n Active SOUND Channel n

Machine Code 0 0 0 0 n n n n 0 1 0 0 0 1 0 1

Description The specified sound begins playing and overwriting the previous singing 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 C ← 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 affected.

Operation $ACC \leftarrow ACC+\overline{XH}+1$

SUB A,[R1R0] Subtract data memory content from accumulator

Machine Code 0 0 0 0 1 0 1 1

Description The content of the data memory addressed by the register pair "R1,R0" is subtracted from the accumulator. Carry is affected.

Operation $ACC \leftarrow ACC+\overline{M(R1,R0)}+1$

TIMER OFF Set timer stop counting

Machine Code 0 0 1 1 1 0 0 1

Description The Timer stop 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 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 Exclussive–ORed with the immediate data specified by 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 Exclussive–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 Exclussive–ORed with the accumulator.

Operation M(R1,R0) ← M(R1,R0) "XOR" ACC