HOLTEK

HTG12N0
4-Bit Microcontroller

Features

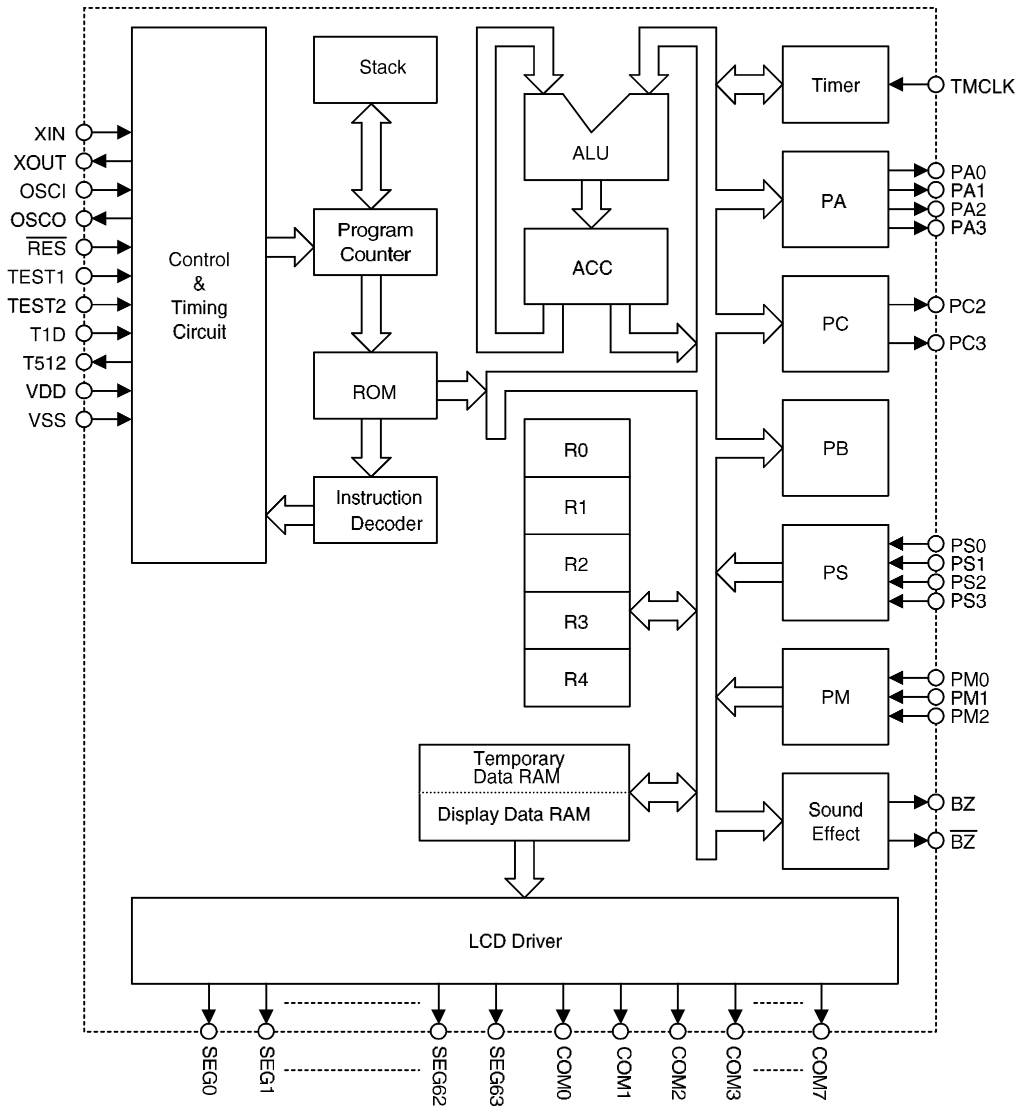
- Operating voltage: 2.4V~3.5V
- Seven input lines
- Six output lines
- Halt feature reduces power consumption
- Up to 4μs instruction cycle with 1MHz system clock
- 4K × 8 × 4 program ROM
- Data memory RAM size 256 × 4 bits
- 64 segments × 8 commons, 1/5 bias LCD driver
- 8-bit table read instruction
- Five working registers
- Internal timer overflow
- One level subroutine nesting
- RC oscillator and 32768Hz crystal oscillator
- 8-bit timer with internal or external clock source
- Sound effect circuit

General Description

The HTG12N0 is the processor from HOLTEK's 4-bit stand alone single chip microcontroller specially designed for LCD display and time piece product applications.

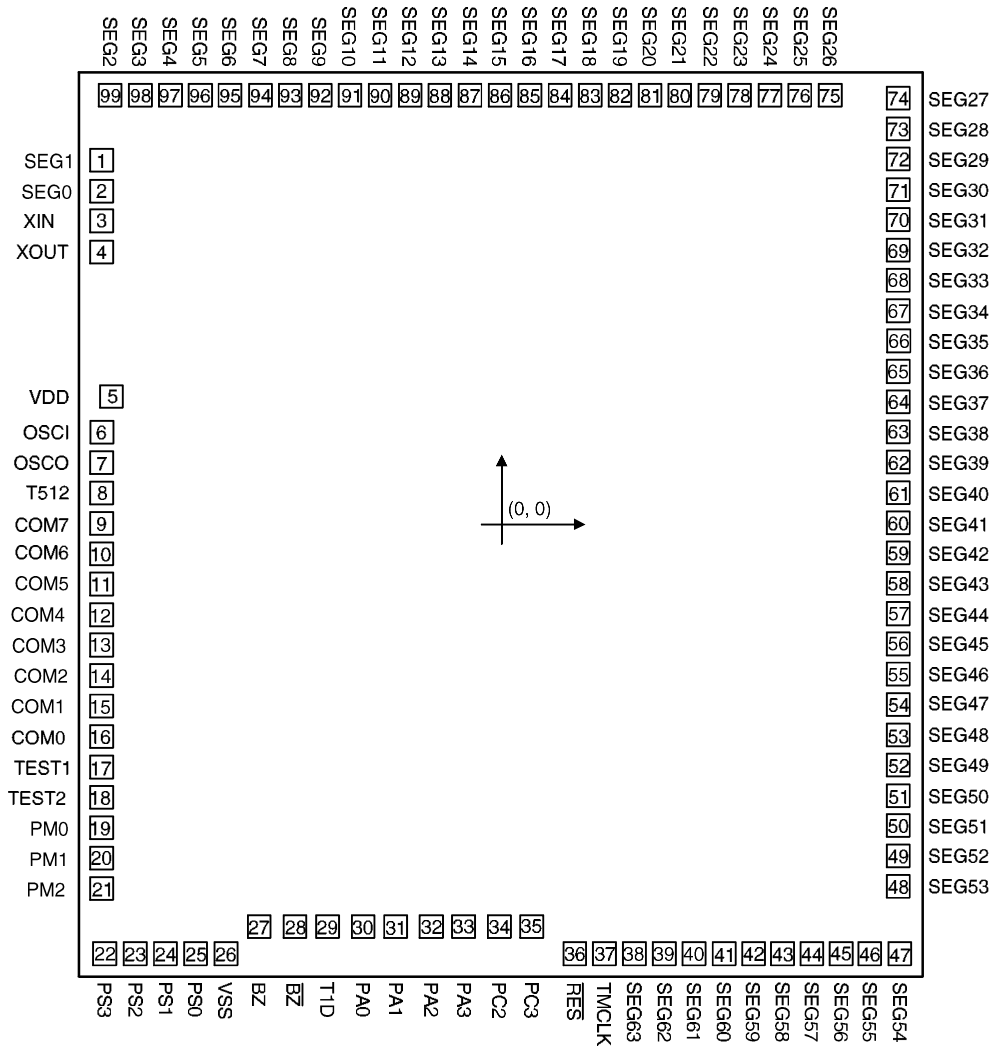
It is ideally suited for multiple LCD time piece low power applications among which are calculators, scales, calendar and hand held LCD products.

Block Diagram



Notes: ACC: Accumulator
PB0, PB1: ROM bank switch
PC1: LCD On/Off switch
PS, PM0~PM2: Input ports
R0~R4: Working registers
PC0: RAM bank switch
PA, PC2~PC3: Output ports

Pad Assignment



Chip size: 3430 × 3730 $(\mu m)^2$

* 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	Pad No.	X	Y
1	–1592.40	1448.48	34	–10.08	–1598.48	67	1578.80	850.32
2	–1592.40	1324.64	35	119.04	–1598.48	68	1578.80	970.96
3	–1592.40	1207.36	36	290.48	–1706.56	69	1578.80	1091.28
4	–1592.40	1083.52	37	409.20	–1706.56	70	1578.80	1211.92
5	–1553.16	508.96	38	527.92	–1706.56	71	1578.80	1332.24
6	–1592.40	367.52	39	646.64	–1706.56	72	1578.80	1452.88
7	–1592.40	246.48	40	765.36	–1706.56	73	1578.80	1573.20
8	–1592.40	125.44	41	884.48	–1706.56	74	1578.80	1695.12
9	–1592.40	4.40	42	1001.44	–1706.56	75	1306.40	1706.56
10	–1592.40	–116.64	43	1117.60	–1706.56	76	1185.65	1706.56
11	–1592.40	–237.68	44	1233.44	–1706.56	77	1066.56	1706.56
12	–1592.40	–358.72	45	1349.60	–1706.56	78	947.12	1706.56
13	–1592.40	–479.76	46	1465.44	–1706.56	79	828.00	1706.56
14	–1592.40	–600.80	47	1584.16	–1706.56	80	708.56	1706.56
15	–1592.40	–721.84	48	1578.80	–1438.64	81	589.44	1706.56
16	–1592.40	–842.88	49	1578.80	–1318.32	82	470.00	1706.56
17	–1592.40	–963.92	50	1578.80	–1197.68	83	350.88	1706.56
18	–1592.40	–1084.96	51	1578.80	–1077.36	84	231.44	1706.56
19	–1592.40	–1206.00	52	1578.80	–956.72	85	112.32	1706.56
20	–1592.40	–1327.04	53	1578.80	–836.40	86	–7.12	1706.56
21	–1592.40	–1448.08	54	1578.80	–715.76	87	–126.24	1706.56
22	–1579.60	–1706.56	55	1578.80	–595.44	88	–245.68	1706.56
23	–1459.36	–1706.56	56	1578.80	–474.80	89	–364.80	1706.56
24	–1338.80	–1706.56	57	1578.80	–354.48	90	–484.24	1706.56
25	–1218.56	–1706.56	58	1578.80	–233.84	91	–603.36	1706.56
26	–1097.52	–1706.56	59	1578.80	–113.52	92	–722.80	1706.56
27	–965.12	–1598.48	60	1578.80	7.12	93	–841.92	1706.56
28	–823.20	–1598.48	61	1578.80	127.44	94	–961.36	1706.56
29	–694.08	–1598.48	62	1578.80	248.08	95	–1080.48	1706.56
30	–552.16	–1598.48	63	1578.80	368.40	96	–1199.92	1706.56
31	–423.04	–1598.48	64	1578.80	489.04	97	–1319.04	1706.56
32	–281.12	–1598.48	65	1578.80	609.36	98	–1438.48	1706.56
33	–152.00	–1598.48	66	1578.80	730.00	99	–1557.60	1706.56

Pad Description

Pad No.	Pad Name	I/O	Mask Option	Description
38~99 1~2	SEG63~SEG2 SEG1~SEG0	O	—	LCD driver outputs for LCD panel segment
3 4	XIN XOUT	I O	—	32768Hz crystal oscillator for time base, LCD clock
5	VDD	I	—	Positive power supply
6 7	OSCI OSCO	I O	—	An external resistor between OSCI and OSC0 is needed for the internal system clock.
8 29 17 18	T512 T1D TEST1 TEST2	O O I I		For test mode only TEST1 and TEST2 are left open when the chip is in normal operation (with an internal pull-high resistor).
9~16	COM7~COM0	O	—	Output for LCD panel common plate
22~25 21~19	PS3~PS0 PM2~PM0	I	Pull-high or None, Note 2	Input pins for input only
26	VSS	I	—	Negative power supply, GND
27, 28	BZ, $\overline{\text{BZ}}$	O	Note 1	Sound effect outputs
33~30 35~34	PA3~PA0 PC3~PC2	O	CMOS or NMOS Open Drain	Output latch pins for output only
36	$\overline{\text{RES}}$	I	—	Input to reset an internal LSI Reset is active on logical low level
37	TMCLK	I	Pull-high or None, Note 3	Input for TIMER clock TIMER can be clocked by an external clock or an internal frequency source.

Notes:

1. The system clock provides six 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.
2. Each bit of ports PM0~PM2, PS can be a trigger source of the HALT interrupt, selectable by mask option.
3. 14 internal clock sources can be selected by mask option to drive TMCLK. Note that TMCLK should not be connected to a pull high resistor if an internal source is used.

Absolute Maximum Ratings

Supply Voltage ....–0.3V~5.5V

Input Voltage....................V_{SS}–0.3V~V_{DD}+0.3V

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

Operating Temperature..................... 0°C~70 °C

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

D.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	3	3.5	V
I_{DD}	Operating Current (LCD ON)	3V	No load, f_{SYS}=512kHz	—	100	200	μA
I_{STB1}	Standby Current (LCD OFF)	3V	HALT mode	—	2	5	μA
I_{STB2}	Standby Current (LCD ON)	3V	HALT mode	—	10	20	μA
V_{IL}	Input Low Voltage	3V	—	0	—	0.2V_{DD}	V
V_{IH}	Input High Voltage	3V	—	0.8V_{DD}	—	V_{DD}	V
I_{OL1}	PA, PC, BZ and $\overline{BZ}$ Output Sink Current	3V	V_{OL}=0.3V	1.5	3	—	mA
I_{OH1}	PA, PC, BZ and $\overline{BZ}$ Output Source Current	3V	V_{OH}=2.7V	–0.5	–1	—	mA
I_{OL2}	Segment Output Sink Current	3V	V_{OL}=0.3V	30	60	—	μA
I_{OH2}	Segment Output Source Current	3V	V_{OH}=2.7V	–50	–100	—	μA
R_{PH}	Pull-high Resistor	3V	PS, PM, $\overline{RES}$, TMCLK	15	100	200	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~36kΩ	128	—	1000	kHz
f_{LCD}	LCD Clock	3V	—	—	256	—	Hz
t_{COM}	LCD Common Period	—	1/8 duty	—	$(1/f_{LCD})\times8$	—	s
t_{CY}	Cycle Time	3V	f_{SYS}=1MHz	—	4	—	μs
f_{TIMER}	Timer I/P Frequency (TMCLK)	3V	—	0	—	1000	kHz
t_{RES}	Reset Pulse Width	—	—	5	—	—	ms
f_{SOUND}	Sound Effect Clock	—	—	—	*64 or 128	—	kHz

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

Functional Description

Program counter – PC

This counter addresses the program ROM and is arranged as a 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, RTC interrupt or returning from a subroutine, the program counter is loaded with the corresponding instruction data as shown in the table.

Notes: P0~P11: Instruction code

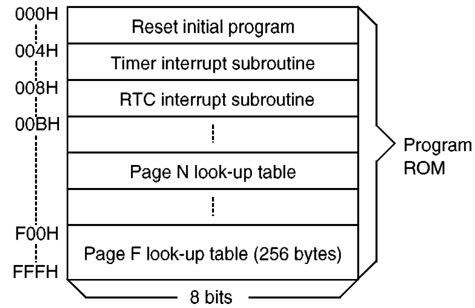
@: PC11 keeps the current value

S0~S11: Stack register bits

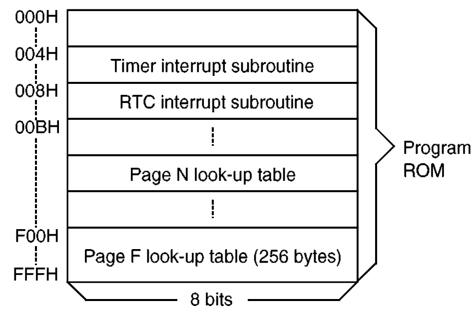
PB0 and PB1 are set to 0 at power on reset.

Program memory – ROM

The program memory is the executable memory and is arranged in a 4096×8 bit format. There are four banks for the program memory in HTG12N0, each bank shown in the figure can be switched by the assignment of PB0 and PB1. The address is specified by the program counter (PC). Four special locations are reserved as shown below.



Program memory PB0=0, PB1=0



Program memory PB0=1, PB1=0

• Location 0

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

Mode	Program Counter													
	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial reset	PB1	PB0	0	0	0	0	0	0	0	0	0	0	0	0
Internal interrupt	PB1	PB0	0	0	0	0	0	0	0	0	0	1	0	0
External interrupt	PB1	PB0	0	0	0	0	0	0	0	0	1	0	0	0
Jump, call instruction	PB1	PB0	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
Conditional branch	PB1	PB0	@	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
Return from subroutine	PB1	PB0	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

Program memory

• 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.

• Location 8

Activating the RTC of the processor with the interrupts enabled causes the program to jump to this location.

• Locations n00H to nFFH

Each page in the program memory consists of 256 bytes. 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 used first to place the program counter in the right page. The above instructions can then be used to read the look up table data.

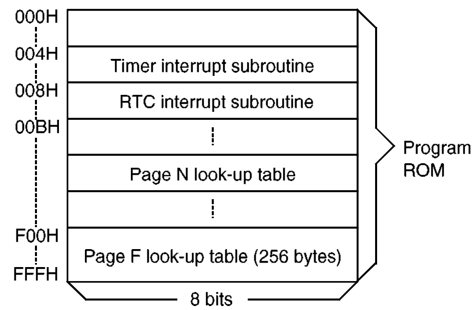
Note that the page number n must be greater than zero as some locations in page 0 are reserved for specific usage as mentioned. This area may function as normal program memory as required.

The program memory mapping is shown in the diagram.

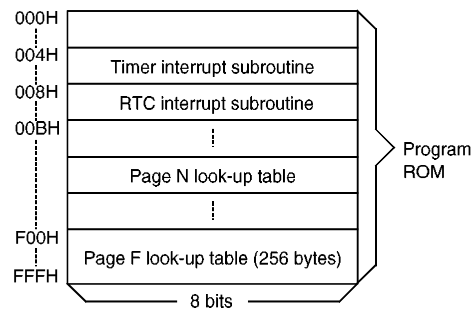
In the execution of an instruction the program counter is added before the execution phase, so careful manipulation of READ MR0A and READ R4A is required in the page margin.

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 $\times$ 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



Program memory PB0=0, PB1=1



Program memory PB0=1, PB1=1

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 routine which is signaled by a return instruction, RET or RETI restore the program counter to its previous value from the stack register. Executing “RETI” instruction will restore the carry flag from the stack register, but “RET” will not.

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

There are five 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

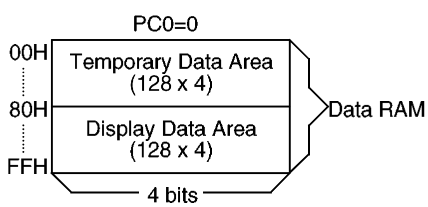
The static data memory (RAM) is arranged in 256×4 bit format and is used to store data. All of the data memory locations are indirectly addressable through the register pair R1,R0 or R3,R2; for example MOV A,[R3R2] or MOV [R3R2],A.

There are two banks for data memory in HTG12N0, each bank shown in the figure can be switched by the assignment of PC0. Each bank maps to different area of the data memory.

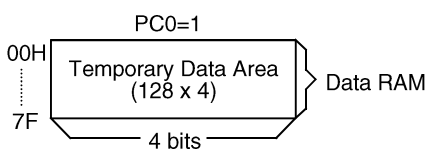
There are two areas in the data memory, the temporary data area and the display data area. Access to the temporary data area is from 00H to 7FH of bank 0 and 00H to 7FH of bank 1, Locations 80H to FFH (don't care the bank pointer) represent the display data area.

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 are shown in the table.



Data memory



Data memory

Display data area (80H~FFH) don't care about the PC0.

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 HTG12N0 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, 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 it is 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 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 instructions. If the interrupt is enabled the timer overflow will cause a subroutine call to location 4. The state of the timer flag can also be tested with the conditional jump instruction JTMR. The timer flag is cleared after the interrupt or the JTMR instruction is executed.

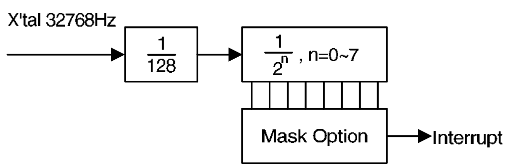
If an internal source is used, the frequency is determined by the system clock and the parameter n as defined in the equation. The frequency of the internal frequency source can be selected by mask option.

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

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

RTC

There is a real time clock (RTC) function implemented on the HTG12N0. The RTC function is used to generate an accurate time period. The clock source of RTC circuit comes from the 32768Hz crystal oscillator. The block diagram is shown as follows.



The RTC output can be selected by mask option.

$$\text{Frequency of the RTC output} = \frac{256}{2^n},\ n=0\sim7$$

The RTC output is used to generate an interrupt signal.

Interrupt

The HTG12N0 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 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 a 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 interrupt arrive at the same time.

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

Initial reset

The HTG12N0 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, it 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 at least 5ms.

When $\overline{\text{RES}}$ is active, the internal block will be initialized as shown 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)
LCD output	Disabled
BZ and $\overline{\text{BZ}}$ output	High level

Halt

This is a special feature of the HTG12N0 used to interrupt the chip's normal operation and reduce the 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
- RTC oscillator still keeps running
- BZ and $\overline{\text{BZ}}$ keep high level output

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

Initial reset: 00H

Wake-up: next address of the HALT instruction

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

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

- The system will awake 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 PM, 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 effects

The HTG12N0 includes sound effect circuitry which offers up to 16 sounds with 3 tones, boom and noise effects. Holtek supports a sound library including melodies, alarms, machine guns etc..

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

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

Since sounds 0~11 contain 32 notes and sounds 12~15 include 64 notes, the latter possesses better sound than the former.

The sound effect circuit frequency 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 frequency 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 figure illustrates the mapping between the display memory and LCD pattern for the HTG12N0.

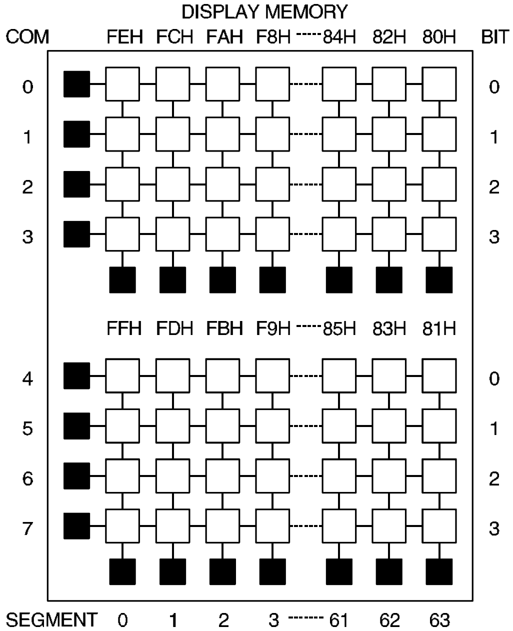
There is an ON/OFF switch for display controlled by bit 1 of port PC (PC1). The corresponding bit of the PC1 represents "ON" or "OFF" of the LCD display memory.

The LCD display module may have any form as long as the number of commons does not exceed 8 and the number of segments is not over 64.

LCD driver output

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

The output number of the HTG12N0 LCD driver is 64×8 which can directly drive an LCD with 1/8 duty cycle and 1/5 bias.



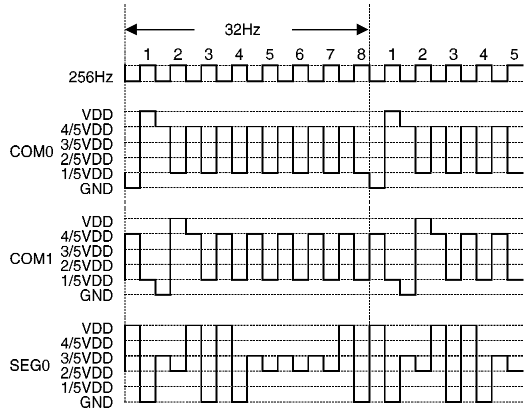
LCD display memory

The LCD driving clock frequency is fixed at 256Hz. This is set by the RTC OSC (32.768kHz).

LCD driver output can be enabled or disabled by setting PC1 without the influence of the related memory condition.

LCD driver output is enabled by setting PC1 as "1", and disabled by setting PC1 as "0".

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



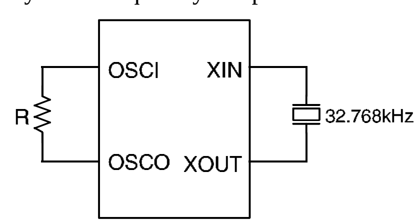
Oscillator

Only one external resistor is required for the HTG12N0 system clock.

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

Another crystal oscillator is needed for use as the reference signal of LCD driving clock and RTC interrupt clock source.

A machine cycle consists of a sequence of four states numbered T1 to T4. Each state lasts for one oscillator period. The machine cycle is 4µs if the system frequency is up to 1MHz.



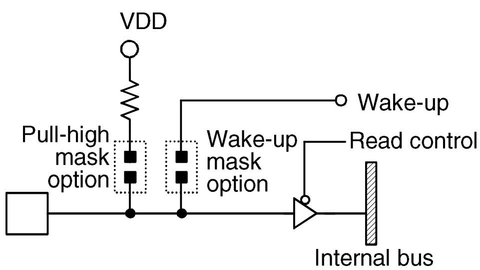
RC and RTC oscillator

Interfacing

The HTG12N0 microcontroller communicates with the outside world through 7-bit input pins PS and PM0~PM2 and 6-bit output pins PA and PC2~PC3.

Input ports – PS, PM0~PM2

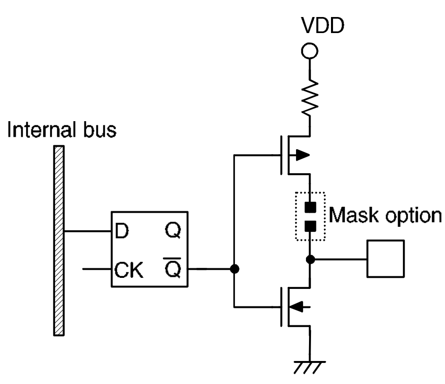
All of the ports can have internal pull high resistors determined by mask option. Every bit of the input ports PS and PM0~PM2 can be specified to be a trigger source for waking up the HALT interrupt by mask option. A high to low transition on one of these pins will wake up the device from a HALT status.



Input ports PS, PM0~PM2

Output port – PA, PC2~PC3

A mask option is available to select whether the output is of a CMOS or open drain NMOS type. After an initial clear the output port PA and PC2~PC3 defaults to be high for CMOS or floating for NMOS.



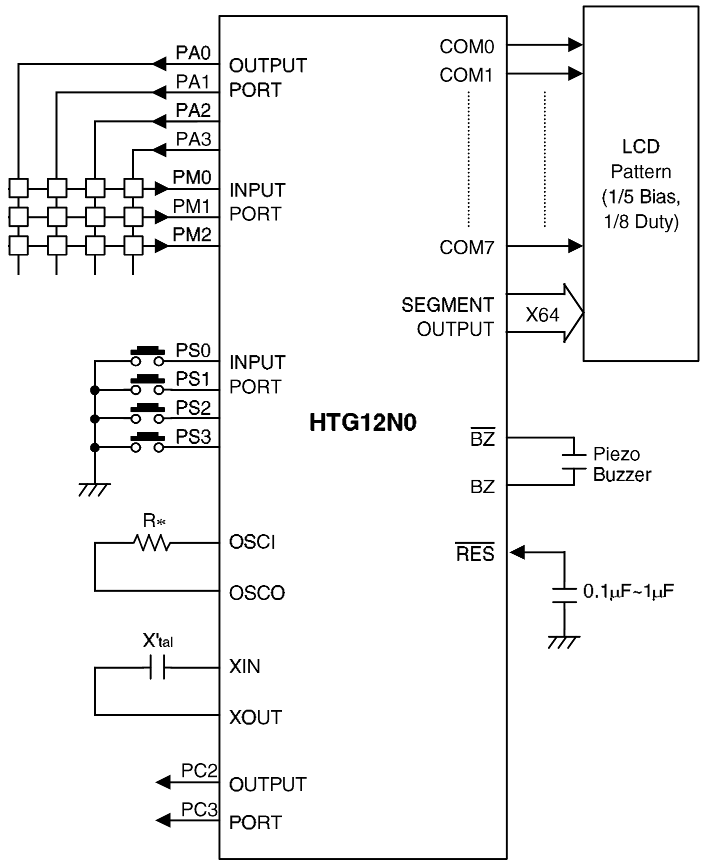
Output port PA and PC2~PC3

Mask options

HTG12N0 provides seven kinds of mask option for different applications.

- Each bit of input ports PS, PM0~PM2 with pull-high resistor
- Each bit of input ports PS, PM0~PM2 function as HALT wake-up trigger
- Each bit of output port PA, PC2~PC3 with CMOS or open drain NMOS
- 8-bit programmable TIMER with internal or external frequency sources. There are 14 internal frequency sources which can be selected as a clocking signal.
  If using internal frequency sources as clocking signal TMCLK cannot connect with a pull-high resistor.
- 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.
- Three kinds of LCD bias current, 6μA, 15μA and 60μA for suitable size of LCD panel.

Application Circuits



R*: Depends on the required system clock frequency (R=36kΩ~620kΩ, at V_{DD}=3V)
X'tal: Realtime clock frequency (X'tal=32768Hz)

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, 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	—
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	—

Mnemonic	Description	Byte	Cycle	CF
Rotate				
RL A RLC A RR A RRC A	Rotate ACC left Rotate ACC left through the carry Rotate ACC right Rotate ACC right through the carry	1 1 1 1	1 1 1 1	√ √ √ √
Input & Output				
IN A,Pi OUT Pi,A	Input port-i to ACC, port-i=PM0~PM2,PS Output ACC to port-i, port-i=PC2~PC3, PA	1 1	1 1	— —
Branch				
JMP addr JC addr JNC addr JTMR addr JAn addr JZ A,addr JNZ A,addr JNZ Rn,addr	Jump unconditionally Jump on carry=1 Jump on carry=0 Jump on timer overflow Jump on ACC bit n=1 Jump on ACC is zero Jump on ACC is not zero Jump on register Rn not zero, n=0,1,4	2 2 2 2 2 2 2 2	2 2 2 2 2 2 2 2	— — — — — — — —
Subroutine				
CALL addr RET RETI	Subroutine call Return from subroutine or interrupt Return from interrupt service routine	2 1 1	2 1 1	— — √
Flag				
CLC STC EI DI NOP	Clear carry flag Set carry flag Enable interrupt Disable interrupt No operation	1 1 1 1 1	1 1 1 1 1	0 1 — — —
Timer				

TIMER XXH
TIMER ON
TIMER OFF
MOV A,TMRL
MOV A,TMRH
MOV TMRL,A
MOV TMRH,A
	Set 8 bits immediate data to TIMER
	Set TIMER start counting
	Set TIMER stop counting
	Move low nibble of TIMER to ACC
	Move high nibble of TIMER to ACC
	Move ACC to low nibble of TIMER
	Move ACC to high nibble of TIMER
		2
		1
		1
		1
		1
		1
		1
			2
			1
			1
			1
			1
			1
			1
				—
				—
				—
				—
				—
				—
				—

Table Read				
READ R4A READ MR0A READF R4A READF MR0A	Read ROM code of current page to R4 and ACC Read ROM code of current page to M(R1,R0), ACC Read ROM code of page F to R4 and ACC Read ROM code of page F to M(R1,R0),ACC	1 1 1 1	2 2 2 2	— — — —

Mnemonic	Description	Byte	Cycle	CF
Sound Control				
SOUND n SOUND A SOUND ONE SOUND LOOP SOUND OFF	Activate SOUND channel n Activate SOUND channel with ACC Turn on SOUND one cycle Turn on SOUND repeat cycle Turn off SOUND	2 1 1 1 1	2 1 1 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)+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 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 a 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. 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 than 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 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)+1

IN A,Pi Input port to accumulator

Machine Code 0 0 1 1 0 0 1 0 PM 0 0 1 1 0 0 1 1 PS

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

Operation ACC ← Pi; Pi=PM or PS

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 the memory bank remain, if the 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 the 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 the 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 the 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 the 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 the 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 the 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 the 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 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 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 the 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 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 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 OR 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 logical OR with the accumulator.
Operation	M(R1,R0) ← M(R1,R0) “OR” ACC

OUT Pi,A	Output accumulator data to port-i
Machine Code	0 0 1 1 0 0 0 0 PA 0 0 1 1 0 1 0 0 PC
Description	The data in the accumulator is transferred to the port-i and latched.
Operation	Pi ← ACC; Pi=PA or PC

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 the ROM code are specified 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 the accumulator. The address of the 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 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 code is loaded to the 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 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 the 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

Description 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
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 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 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 the 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 repeatedly.
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. (nnn=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 ← 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 affected.	
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 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 Exclusive-OR 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 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 logically Exclusive-OR with the accumulator.

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