8-Bit 1024 Pixel Dot Matrix LCD MCU Series

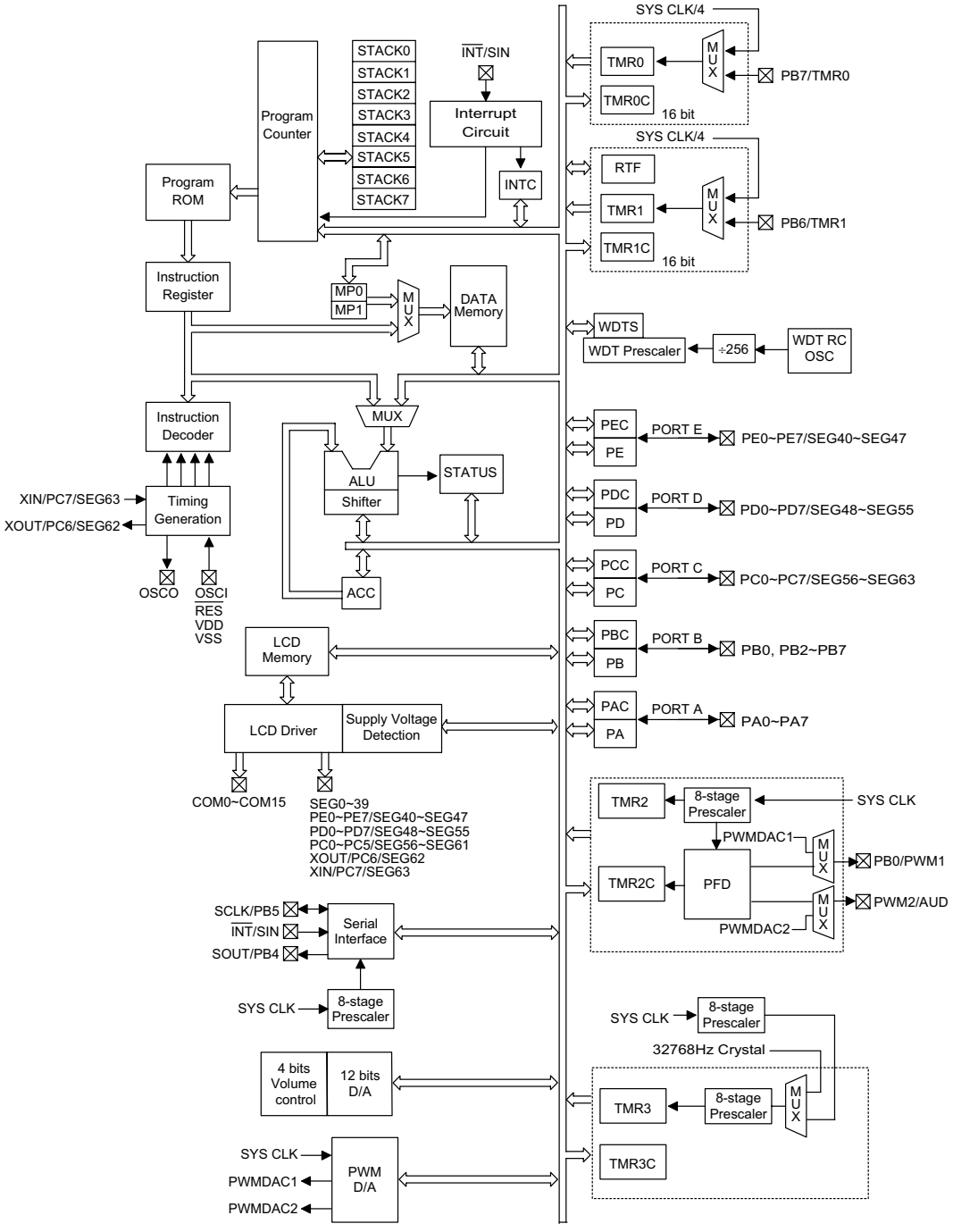
Features

- Operating voltage: 2.4V~3.6V
- 64K×16 bits program ROM
- 2.3K×8 bits data RAM
- 15~39 bidirectional I/O lines
- 16 common×40~64 segment LCD driver
- Two 16-bit programmable timer/event counters with overflow interrupts
- One 8-bit programmable timer with 8-stage prescaler for PFD
- One 8-bit programmable timer with 8-stage prescaler for time base
- One 8-bit PWM audio output to directly drive speaker or buzzer
- One 12-bit current type D/A output with 4-bit volume control
- R to F function for temperature measurement
- Synchronous serial interface
- On-chip RC oscillator for system clock
- 32768Hz crystal oscillator for time base and LCD driver
- Watchdog Timer
- HALT function and wake-up feature reduce power consumption
- Eight-level subroutine nesting
- Bit manipulation instructions
- 63 powerful instructions
- Built-in supply voltage detection circuit
- One interrupt input
- 128-pin QFP package

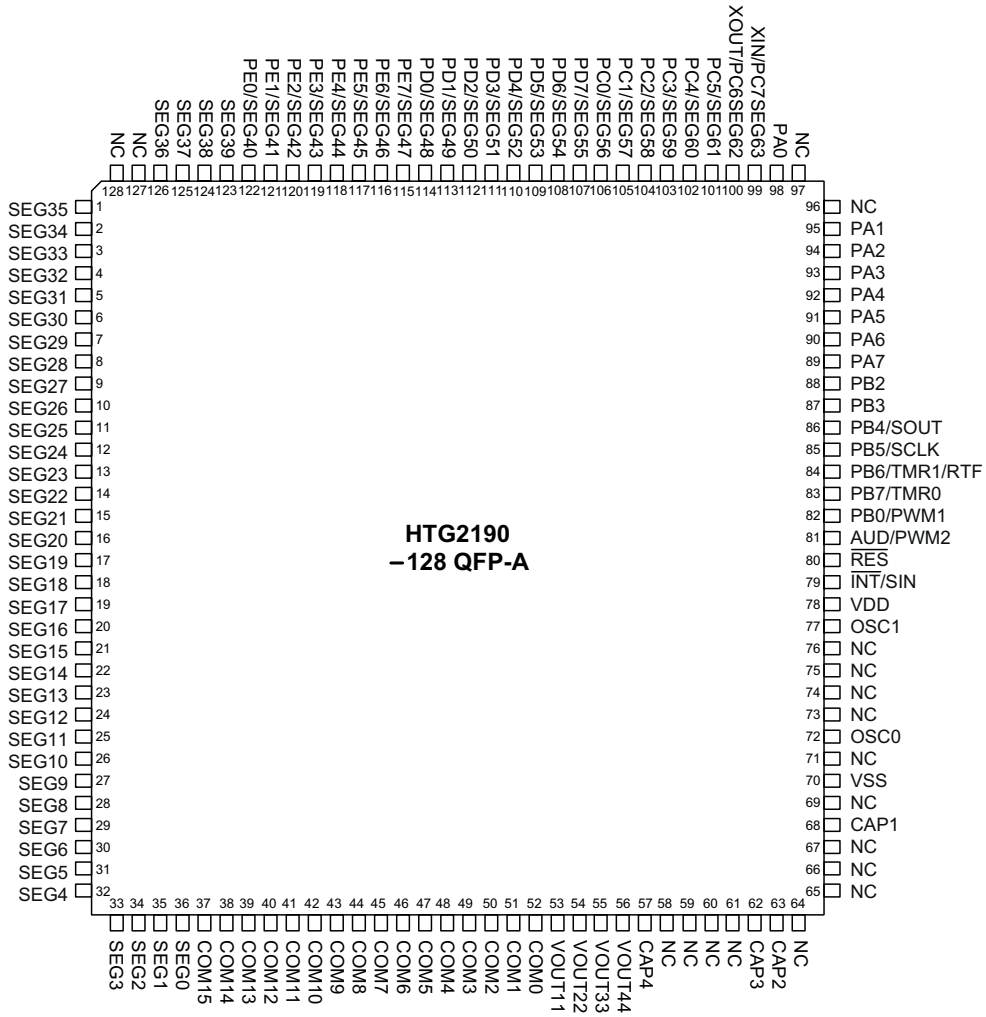
General Description

The HTG2190 is an 8-bit high performance RISC-like microcontroller capable of driving 1024 pixel (max.) LCD display. Its single cycle instruction and two-stage pipeline architecture make it suitable for high speed applications. The device is suited for use in multiple LCD low power applications among which are calculators, clock timer, game, scales, leisure products, other hand held LCD products and battery operated systems in particular.

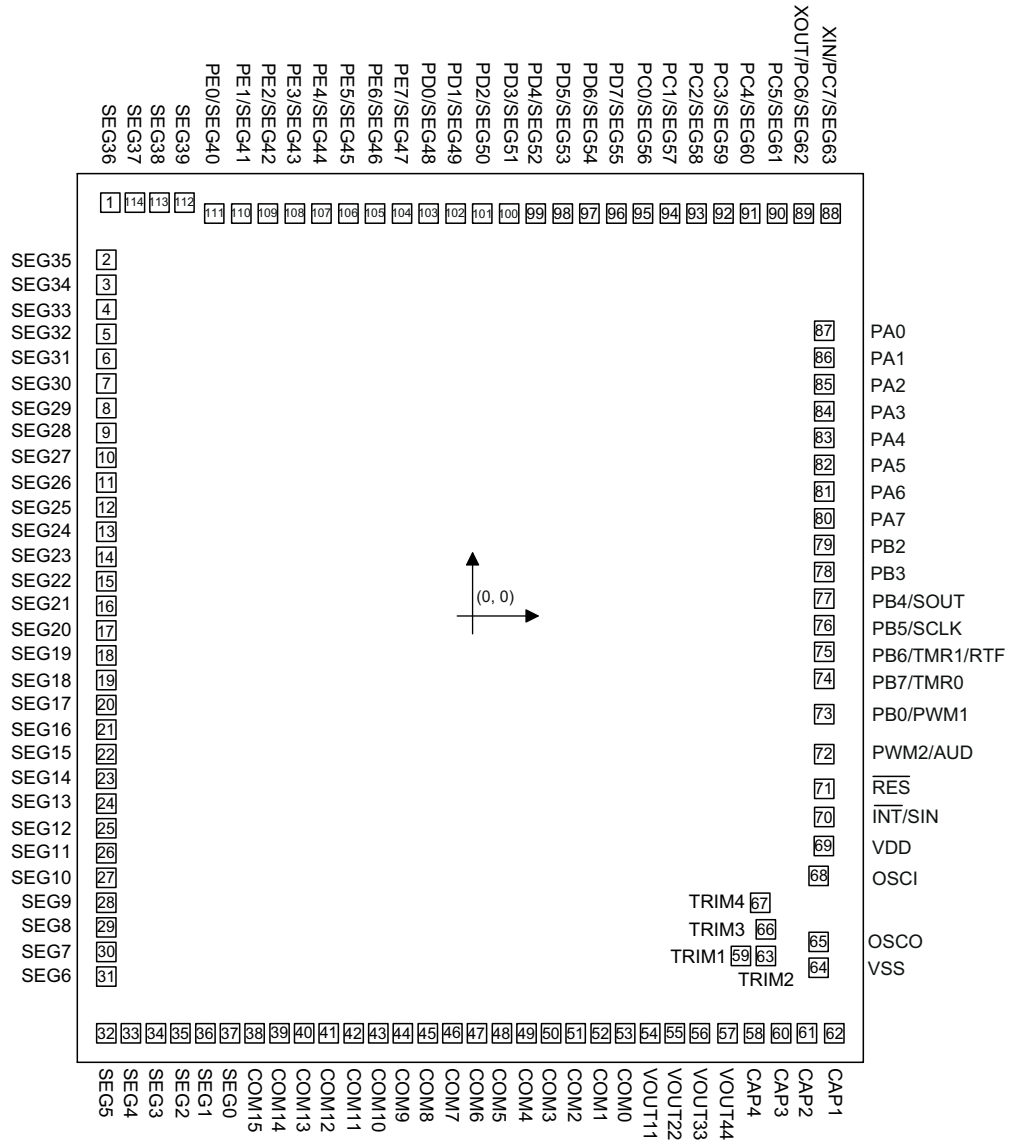
Block Diagram



Pin Assignment



Pad Assignment



Chip size: 3420 × 3855 (μm)^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	−1561.08	1781.81	58	1214.37	−1797.56
2	−1577.85	1535.18	59	1156.72	−1465.33
3	−1577.85	1428.50	60	1331.47	−1797.56
4	−1577.85	1322.07	61	1442.47	−1797.56
5	−1577.85	1215.39	62	1559.56	−1797.56
6	−1577.85	1108.96	63	1264.67	−1465.33
7	−1577.85	1002.28	64	1492.25	−1516.38
8	−1577.85	895.86	65	1492.25	−1404.62

Pad No.	X	Y	Pad No.	X	Y
9	–1577.85	789.18	66	1264.67	–1351.53
10	–1577.85	682.75	67	1240.28	–1235.20
11	–1577.85	576.33	68	1492.25	–1119.38
12	–1577.85	469.65	69	1515.36	–991.36
13	–1577.85	363.22	70	1517.14	–867.16
14	–1577.85	256.54	71	1517.14	–745.49
15	–1577.85	150.11	72	1517.14	–596.65
16	–1577.85	43.43	73	1517.14	–422.91
17	–1577.85	–62.99	74	1517.14	–271.27
18	–1577.85	–169.67	75	1517.14	–154.18
19	–1577.85	–276.10	76	1517.14	–40.13
20	–1577.85	–382.78	77	1517.14	76.45
21	–1577.85	–489.20	78	1517.14	190.50
22	–1577.85	–595.88	79	1517.14	307.34
23	–1577.85	–702.31	80	1517.14	421.39
24	–1577.85	–809.99	81	1517.14	538.23
25	–1577.85	–915.42	82	1517.14	652.27
26	–1577.85	–1022.10	83	1517.14	768.86
27	–1577.85	–1128.52	84	1517.14	882.90
28	–1577.85	–1235.20	85	1517.14	999.74
29	–1577.85	–1341.63	86	1517.14	1113.79
30	–1577.85	–1448.31	87	1517.14	1230.63
31	–1577.85	–1554.73	88	1544.83	1734.82
32	–1577.85	–1797.56	89	1427.99	1734.82
33	–1471.17	–1797.56	90	1313.94	1734.82
34	–1364.74	–1797.56	91	1196.85	1734.82
35	–1258.06	–1797.56	92	1083.06	1734.82
36	–1151.64	–1797.56	93	965.96	1734.82
37	–1044.96	–1797.56	94	851.92	1734.82
38	–938.53	–1797.56	95	735.08	1734.82
39	–831.85	–1797.56	96	621.03	1734.82
40	–725.42	–1797.56	97	503.94	1734.82
41	–618.74	–1797.56	98	389.89	1734.82
42	–512.32	–1797.56	99	273.05	1734.82
43	–405.64	–1797.56	100	159.00	1734.82
44	–299.21	–1797.56	101	41.91	1734.82
45	–192.53	–1797.56	102	–72.14	1734.82
46	–86.11	–1797.56	103	–188.98	1734.82
47	20.57	–1797.56	104	–303.02	1734.82
48	127.00	–1797.56	105	–420.12	1734.82
49	233.45	–1797.56	106	–534.16	1734.82
50	340.11	–1797.56	107	–651.00	1734.82
51	446.53	–1797.56	108	–765.05	1734.82
52	553.21	–1797.56	109	–882.14	1734.82
53	659.64	–1797.56	110	–995.93	1734.82
54	766.83	–1797.56	111	–1113.03	1734.82
55	873.25	–1797.56	112	–1241.55	1781.81
56	979.68	–1797.56	113	–1347.98	1781.81
57	1100.07	–1797.56	114	–1454.66	1781.81

Pad Descriptions

Pad No.	Pad Name	I/O	Mask Option	Description
37~1 114~112	SEG0~SEG39	O	—	LCD segment signal output
53~38	COM0~COM15	O	—	LCD common signal output
54~57	VOUT11, VOUT22 VOUT33, VOUT44	O	—	LCD driving power generated
62, 61, 60, 58	CAP1, CAP2, CAP3, CAP4	—	—	LCD system voltage booster condenser connecting terminal.
59, 63 66, 67	TRIM1~TRIM4	—	—	Test pin only
64	VSS	—	—	Negative power supply, ground
68 65	OSCI OSCO	I O	Crystal or RC	OSCI and OSCO are connected to an RC network or a crystal (by mask option) for the internal system clock. In the case of RC operation,OSCI is connected to the RC network of the internal system clock.
69	VDD	—	—	Positive power supply
70	$\overline{INT}$/SIN	I	$\overline{INT}$ or Serial Data Input	Selectable as external interrupt Schmitt trigger input or serial data input by mask option. External interrupt Schmitt trigger input with pull-high resistor. Edge triggered activated on $\overline{INT}$ or Serial data input a high to low transition. Serial data input with pull-high resistor. $\overline{INT}$ shares pad with SIN.
71	$\overline{RES}$	I	—	Schmitt trigger reset input. Active low without pull-high resistor.
72	PWM2/AUD	O	PWM2 CMOS Output or AUD Output	Selectable as negative PWM CMOS output or audio output by mask option. PWM2 share pad with AUD.
73	PB0/PWM1	I/O or O	I/O or PWM1 CMOS Output	Selectable as bidirectional input/output or positive PWM CMOS output by mask option. On bidirectional input/output mode, software instructions determine whether each pad is a CMOS output or Schmitt trigger input with pull-high resistor. PB0 shares pad with PWM1.
74	PB7/TMR0	I/O or I	I/O or TMR0 Input	Selectable as bidirectional input/output or TMR0 input by mask option. On bidirectional input/output mode, software instructions determine whether it is a CMOS output or Schmitt trigger input with pull-high resistor. On TMR0 input mode, it uses Timer/Event Counter 0. Software can be optioned with or without pull-high resistor. PB7 shares pad with TMR0.
75	PB6/TMR1/RTF	I/O or I	I/O or TMR1 Input or RTF Input	Selectable as bidirectional input/output or TMR1 input or RTF input by mask option. On bidirectional input/output mode, software instructions determine whether it is a CMOS output or Schmitt trigger input with pull-high resistor. On TMR1 and RTF input mode, it uses Timer/Event Counter 1. Software can be optioned with or without pull-high resister and be option the RTF function disable or enable. PB6 and TMR1 share pad with RTF.

Pad No.	Pad Name	I/O	Mask Option	Description
76	PB5/SCLK	I/O	I/O or SCLK Signal	Selectable as bidirectional input/output or serial interface clock signal by mask option. On bidirectional input/output mode, software instructions determine whether it is a CMOS output or Schmitt trigger input . Serial I/O interface clock signal. At Master mode: SCLK should be set as serial clock output pin and after 8 clock output from SCLK terminal, clock output is automatically suspended and SCLK terminal is fixed at a high level. At Slave mode: SCLK should be set as serial clock input pin and after 8 clock input from SCLK terminal, subsequent clock input are masked. PB5 shares pad with SLCK
77	PB4/SOUT	I/O or O	I/O or SOUT Output	Selectable as bidirectional input/output or serial data output master by mask option. On bidirectional input/output mode, software instructions determine whether it is a CMOS output or Schmitt trigger input . On serial output mode, SOUT pin is a CMOS output. PB4 shares pad with SOUT.
78 79	PB3 PB2	I/O	—	Software instructions determine whether it is a CMOS output or Schmitt trigger input with pull-high resistor. When the PB6/TMR1/RTF pad is selected to be in RTF mode, the PB3, PB2 pull-high resistor is not present.
87~80	PA0~PA7	I/O	Wake-up or None Wake-up	Bidirectional 8-bit input/output port. Each bit can be configured as a wake-up input by mask option. Software instructions determine whether it is a CMOS output or Schmitt trigger input with pull-high resistor.
88 89	XIN/PC7/SEG63 XOUT/PC6/SEG62	I or I/O or O	Crystal or I/O or Segment Output	Selectable as 32768Hz crystal or bidirectional input/output or segment signal output by mask option. 32768Hz crystal for timer3 and LCD clock source. XOUT and PC6 share with SEG62 Pad,XIN and PC7 share with SEG63 pad.
95~90 103~96 111~104	PC0~5/SEG56~61 PD0~7/SEG48~55 PE0~7/SEG40~47	I/O or O	I/O or Segment Output	Three bidirectional 8-bit input/output ports. Software instructions determine the CMOS output or Schmitt trigger input with pull-high resistor. PC, PD, PE can be individually optioned for segment output by mask option. PC0~PC7 share pads with SEG56~SEG63, PD0~PD7 share pads with SEG48~SEG55, and PE0~PE7 share pads with SEG40~SEG47.

Absolute Maximum Ratings

Supply Voltage .....–0.3V to 3.6V

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

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

Operating Temperature.....0°C to 70°C

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

D.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	—	3.6	V
I_{DD}	Operating Current (RC OSC)	3V	No load, f_{SYS}=4MHz	—	1	1.5	mA
I_{STB1}	Standby Current (RTC ON, LCD ON)	3V	No load, HALT	—	9	30	μA
I_{STB2}	Standby Current (RTC ON, LCD OFF)	3V	No load, HALT	—	3.5	15	μA
V_{IL1}	Input Low Voltage for I/O Ports	3V	—	0	—	0.9	V
V_{IH1}	Input High Voltage for I/O Ports	3V	—	2.1	—	3	V
V_{IL2}	Input Low Voltage (TMR0, TMR1, $\overline{INT}$)	3V	—	0	—	0.7	V
V_{IH2}	Input High Voltage (TMR0, TMR1, $\overline{INT}$)	3V	—	2.3	—	3	V
V_{IL3}	Input Low Voltage ($\overline{RES}$)	3V	—	—	1.5	—	V
V_{IH3}	Input High Voltage ($\overline{RES}$)	3V	—	—	2.4	—	V
I_{OL1}	I/O Ports Sink Current	3V	V_{OL}=0.3V	1.5	4	—	mA
I_{OH1}	I/O Ports Source Current	3V	V_{OH}=2.7V	–1	–2	—	mA
V_{LCD}	LCD Voltage	3V	—	3.96	4.4	4.84	V
I_{OL2}	PWM1/PWM2 Sink Current	3V	V_{OL}=0.3V	12	16	—	mA
I_{OH2}	PWM1/PWM2 Source Current	3V	V_{OH}=2.7V	–8	–10	—	mA
I_{OL3}	Audio Sink Current	3V	V_{OL}=0.3V	1.5	2	—	mA
I_{OH3}	Audio Source Current	3V	V_{OH}=2.7V	–1.5	–2	—	mA
I_{OL4}	Segment 0~39 Output Sink Current	3V	V_{OL}=0.3V	80	130	—	μA
I_{OH4}	Segment 0~39 Output Source Current	3V	V_{OH}=2.7V	–50	–90	—	μA
I_{OL5}	Segment 40~63 Output Sink Current	3V	V_{OL}=0.3V	40	80	—	μA
I_{OH5}	Segment 40~63 Output Source Current	3V	V_{OH}=2.7V	–30	–60	—	μA
R_{PH}	Pull-high Resistance of I/O Ports, TMR0, TMR1 & $\overline{INT}$	3V	—	30	60	100	kΩ

A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SYS}	System Clock (RC OSC)	3V	—	400	—	4000	kHz
f_{TIMER}	Timer Input Frequency (TMR0,TMR1)	3V	—	0	—	4000	kHz
t_{WDTOSC}	Watchdog Oscillator	3V	—	45	90	180	μs
t_{WDT}	Watchdog Time-out Period (RC)	3V	Without WDT prescaler	12	23	45	ms
t_{RES}	External Reset Low Pulse width	—	—	1	—	—	μs
t_{SST}	System Start-up Timer Period	—	Power-up or wake-up from HALT	—	1024	—	t_{SYS}
t_{INT}	Interrupt Pulse Width	—	—	1	—	—	μs

Note: $t_{SYS}=1/f_{SYS}$

Functional Description

Execution flow

The system clock for the HTG2190 is derived from either a crystal or an RC oscillator. It is internally divided into four non-overlapping clocks. One instruction cycle consists of four system clock cycles.

Instruction fetching and execution are pipelined in such a way that a fetch takes one instruction cycle while decoding and execution takes the next instruction cycle. However, the pipelining scheme causes each instruction to effectively execute within one cycle. If an instruction changes the program counter, two cycles are required to complete the instruction.

Program counter – PC

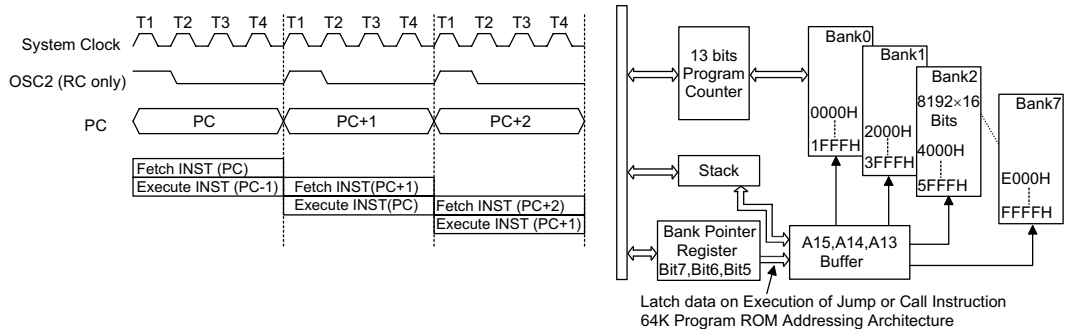
The 13-bit program counter (PC) controls the sequence in which the instructions stored in program ROM are executed.

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

When executing a jump instruction, conditional skip execution, loading PCL register, subroutine call, initial reset, internal interrupt, external interrupt or return from subroutine, the PC manipulates the program transfer by loading the address corresponding to each instruction.

The conditional skip is activated by instruction. Once the condition is met, the next instruction, fetched during the current instruction execution, is discarded and a dummy cycle takes its place while the correct instruction is obtained.

The lower byte of the program counter (PCL) is a read/write register (06H). Moving data into the PCL per-



Execution flow

Mode	Program ROM Address															
	*15	*14	*13	*12	*11	*10	*9	*8	*7	*6	*5	*4	*3	*2	*1	*0
Initial reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
External or serial input interrupt	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Timer/Event Counter 0 overflow	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Timer/Event Counter 1 overflow	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
Timer 2 overflow	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Timer 3 overflow	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
Skip	PC+2															
Loading PCL	*15	*14	*13	*12	*11	*10	*9	*8	@7	@6	@5	@4	@3	@2	@1	@0
Jump, call branch	BP.7	BP.6	BP.5	#12	#11	#10	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
Return from subroutine	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

Program ROM address

Note: *15~*0: Program ROM address
@7~@0: PCL bits
#12~#0: Instruction code bits
S15~S0: Stack register bits
BP.5~BP.7: Bit 5~7 of bank pointer (04H)

forms a short jump. The destination must be within 256 locations.

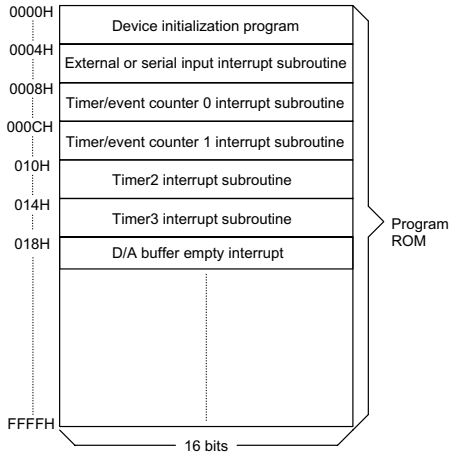
When a control transfer takes place, an additional dummy cycle is required.

Program memory – ROM

The program memory, which contains executable program instructions, data and table information, is composed of a 65536 x 16 bit format. However as the PC (program counter) is comprised of only 13 bits, the remaining 3 ROM address bits are managed by dividing the program memory into 8 banks, each bank having a range between 0000H and 1FFFH. To move from the present ROM bank to a different ROM bank, the higher 3 bits of the ROM address are set by the BP (Bank Pointer), while the remaining 13 bits of the PC are set in the usual way by executing the appropriate jump or call instruction. As the full 16 address bits are latched during the execution of a call or jump instruction, the correct value of the BP must first be setup before a jump or call is executed. When either a software or hardware interrupt is received, note that no matter which ROM bank the program is in the program will always jump to the appropriate interrupt service address in Bank 0. The original full 16 bit address will be stored on the stack and restored when the relevant RET/RETI instruction is executed, automatically returning the program to the original ROM bank. This eliminates the need for programmers to manage the BP when interrupts occur.

Certain locations in Bank 0 of program memory are reserved for special usage:

- ROM Bank 0 (BP5~BP7=000B)
  The ROM bank 0 ranges from 0000H to 1FFFH.
- Location 000H
  This area is reserved for the initialization program. After a chip reset, the program always begins execution at location 000H.
- Location 004H
  This area is reserved for the external interrupt or serial input interrupt service routine. If the $\overline{INT}$ input pin is activated, and the interrupt is enabled and the stack is not full, the program will jump to location 004H and begins execution.
- Location 008H/00CH
  This area is reserved for the Timer/Event Counter 0/1 interrupt service program. If a timer interrupt results from a Timer/Event Counter 0/1 overflow, and if the interrupt is enabled and the stack is not full, the program will jump to location 008H/00CH and begins execution.



Program memory

- Location 010H/014H
  This area is reserved for the timer 2/3 interrupt service program. If a timer interrupt resulting from a timer 2/3 overflow, and if the interrupt is enabled and the stack is not full, the program will jump to location 010H/014H and begins execution.
- Location 018H
  This area is reserved for the PWM D/A buffer empty interrupt service program. After the system latch a D/A code at RAM address 28H, the interrupt is enable, and the stack is not full, the program begins execution at location 018H.
- Location 020H
  For best condition, this location is reserved as the beginning when writing a program.
- ROM Bank 1~7 (BP5~BP7=001B~111)
  The range of the ROM starts from n000H to (n+1)FFFH. (n=2,4,6,8,10,12,14)
- Table location
  Any location in the ROM space can be used as look up table. The instructions TABRDC [m] (use for any bank) and TABRDL [m] (only used for last page of program ROM) transfers the contents of the lower-order byte to the specified data memory, and the higher-order byte to TBLH (08H). Only the destination of the lower-order byte in the table is well-defined, the

Instruction	Table Location															
	*15	*14	*13	*12	*11	*10	*9	*8	*7	*6	*5	*4	*3	*2	*1	*0
TABRDC [m]	#7	#6	#5	#4	#3	#2	#1	#0	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL [m]	1	1	1	1	1	1	1	1	@7	@6	@5	@4	@3	@2	@1	@0

Table location

Note: @7~@0: TBLP register bit7~bit0
*15~*0: Program ROM table address bit15~bit0
#7~#0: TBHP register bit7~bit0

higher-order byte of the table word are transferred to the TBLH. The Table Higher-order byte register (TBLH) is read only. The Table Pointer (TBHP, TBLP) is a read/write register (1FH, 07H), used to indicate the table location. Before accessing the table, the location must be placed in TBLP. The TBLH is read only and cannot be restored. If the main routine and the ISR (Interrupt Service Routine) both employ the table read instruction, the contents of the TBLH in the main routine are likely to be changed by the table read instruction used in the ISR. If this happens errors can occur. In other words, using the table read instruction in the main routine and the ISR simultaneously should be avoided. However, if the table read instruction has to be applied in both the main routine and the ISR, the interrupt(s) should be disabled prior to the table read instruction. It should not be enabled until the TBLH has been backed up. All table related instructions need two cycles to complete the operation. These areas may function as normal program memory depending upon requirements.

Stack register – STACK

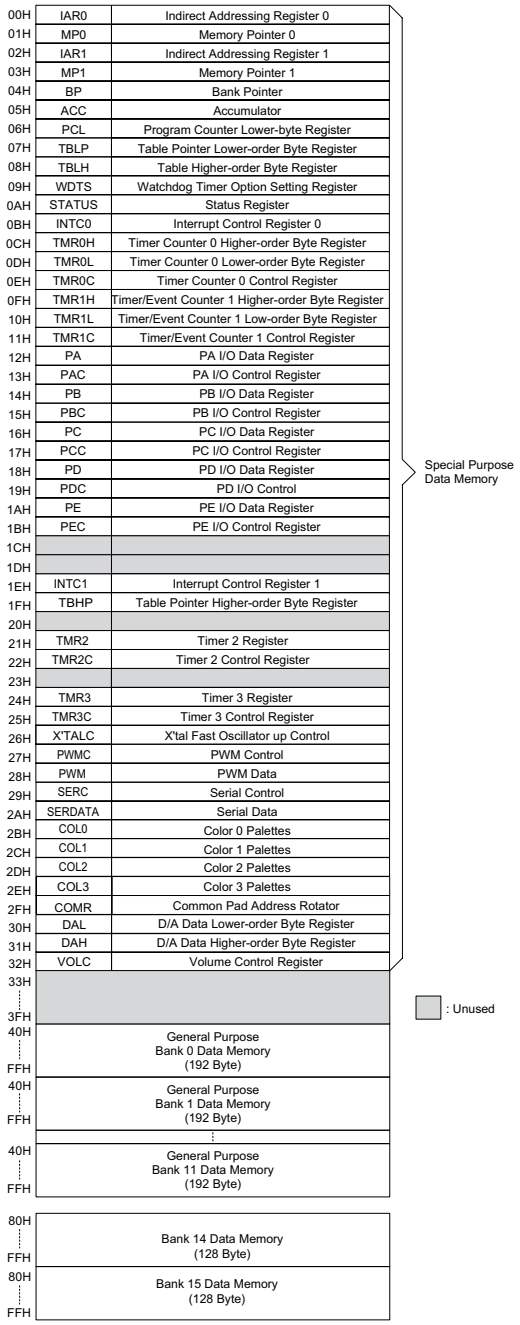
This is a special part of memory which is used to save the contents of the program counter (PC) only. The stack is organized into 8 levels and is neither part of the data nor program space, and is neither readable nor writeable. The activated level is indexed by the stack pointer (SP) and is neither readable nor writeable. At a subroutine call or interrupt acknowledgment, the contents of the program counter are pushed onto the stack. At the end of a subroutine or an interrupt routine, signaled by a return instruction (RET or RETI), the program counter is restored to its previous value from the stack. After a chip reset, the SP will point to the top of the stack.

If the stack is full and a non-masked interrupt takes place, the interrupt request flag will be recorded but the acknowledge will be inhibited. When the stack pointer is decremented (by RET or RETI), the interrupt will be serviced. This feature prevents stack overflow allowing the programmer to use the structure more easily.

In a similar case, if the stack is full and a CALL is subsequently executed, stack overflow occurs and the first entry will be lost (only the most recent eight return address are stored).

Data memory – RAM

- Bank 0 (BP4~BP0=00000)
  The Bank 0 data memory includes special purpose and general purpose memory. The special purpose memory is addressed from 00H to 3FH. All data memory areas can handle arithmetic, logic, increment, decrement and rotate operations directly. Except for some dedicated bits, each bit in the data memory can be set and reset by the SET [m].i and CLR [m].i instructions, respectively. They are also indirectly accessible through the memory pointer registers (MP0;01H, MP1;03H).



RAM mapping

- Bank 1~11 (BP4~BP0=0001B~1011B)
  The range of RAM starting from 40H to FFH are for general purpose. Only MP1 can deal with the memory of this range.

- Bank 14/15 (BP4~BP0=01110B~01111B)
  The range of RAM starts from 80H to FFH. Every bit stands for one dot on the LCD. If the bit is ″1″, the light of the dot on the LCD will be turned on. If the bit is ″0″, then it will be turned off. Only MP1 can deal with the memory of this range.
  The contrast form of RAM location, COMMON, and SEGMENT is as follows.

LCD driver output

The maximum output number of the HTG2190 LCD driver is 16×64. The Common output signal can be selected as 16 com or 8 com by mask option. The LCD driver bias type is ″C″ type, external capacitor is required and the bias voltage is 1/4 bias. Some of the Segment outputs share pins with I/O pins, PE0~PE7 (SEG40~47), PD0~PD7(SEG48~55) and PC0~PC7 (SEG56~63). Whether segment output or I/O pin can be decided by mask option.

LCD driver output can be enabled or disabled by setting COL3 (bit 6 of COL3; 2EH) without the influence of the related memory condition.

On the Color LCD Mode: Every two bits stands for one dot on the ECB. If the both bits is not 0, the light of the dot on the ECB will be turned on. If the both bits is 0, then it will be selected to color 0. Only MP1 can be deal with the memory of this range. The contrast form of RAM location, COMMON, and SEGMENT is as follows.

Address	D7	D6	D5	D4	D3	D2	D1	D0
80H	SEG0~COM3		SEG0~COM2		SEG0~COM1		SEG0~COM0	
81H	SEG0~COM7		SEG0~COM6		SEG0~COM5		SEG0~COM4	
82H	SEG1~COM3		SEG1~COM2		SEG1~COM1		SEG1~COM0	
83H	SEG1~COM7		SEG1~COM6		SEG1~COM5		SEG1~COM4	
84H	SEG2~COM3		SEG2~COM2		SEG2~COM1		SEG2~COM0	
85H	SEG2~COM7		SEG2~COM6		SEG2~COM5		SEG2~COM4	
⋮								
FAH	SEG61~COM3		SEG61~COM2		SEG61~COM1		SEG61~COM0	
FBH	SEG61~COM7		SEG61~COM6		SEG61~COM5		SEG61~COM4	
FCH	SEG62~COM3		SEG62~COM2		SEG62~COM1		SEG62~COM0	
FDH	SEG62~COM7		SEG62~COM6		SEG62~COM5		SEG62~COM4	
FEH	SEG63~COM3		SEG63~COM2		SEG63~COM1		SEG63~COM0	
FFH	SEG63~COM7		SEG63~COM6		SEG63~COM5		SEG63~COM4	

LCD RAM mapping Bank14

Address	D7	D6	D5	D4	D3	D2	D1	D0
80H	SEG0~COM11		SEG0~COM10		SEG0~COM9		SEG0~COM8	
81H	SEG0~COM15		SEG0~COM14		SEG0~COM13		SEG0~COM12	
82H	SEG1~COM11		SEG1~COM10		SEG1~COM9		SEG1~COM8	
83H	SEG1~COM15		SEG1~COM14		SEG1~COM13		SEG1~COM12	
84H	SEG2~COM11		SEG2~COM10		SEG2~COM9		SEG2~COM8	
85H	SEG2~COM15		SEG2~COM14		SEG2~COM13		SEG2~COM12	
⋮								
FAH	SEG61~COM11		SEG61~COM10		SEG61~COM9		SEG61~COM8	
FBH	SEG61~COM15		SEG61~COM14		SEG61~COM13		SEG61~COM12	
FCH	SEG62~COM11		SEG62~COM10		SEG62~COM9		SEG62~COM8	
FDH	SEG62~COM15		SEG62~COM14		SEG62~COM13		SEG62~COM12	
FEH	SEG63~COM11		SEG63~COM10		SEG63~COM9		SEG63~COM8	
FFH	SEG63~COM15		SEG63~COM14		SEG63~COM13		SEG63~COM12	

LCD RAM mapping Bank15

Register	Address	Bit No.	Label	Description
COL0	2BH	0~4	P0~P4	Color0 palette
		5~7	—	Unused bit read only
COL1	2CH	0~4	P0~P4	Color1 palette
		5~7	—	Unused bit read only
COL2	2DH	0~4	P0~P4	Color2 palette
		5~7	—	Unused bit read only

How to select the color:

	Dn+1	Dn
Color 0	0	0
Color 1	0	1
Color 2	1	0
Color 3	1	1
	Segment—Common	

LCDC register

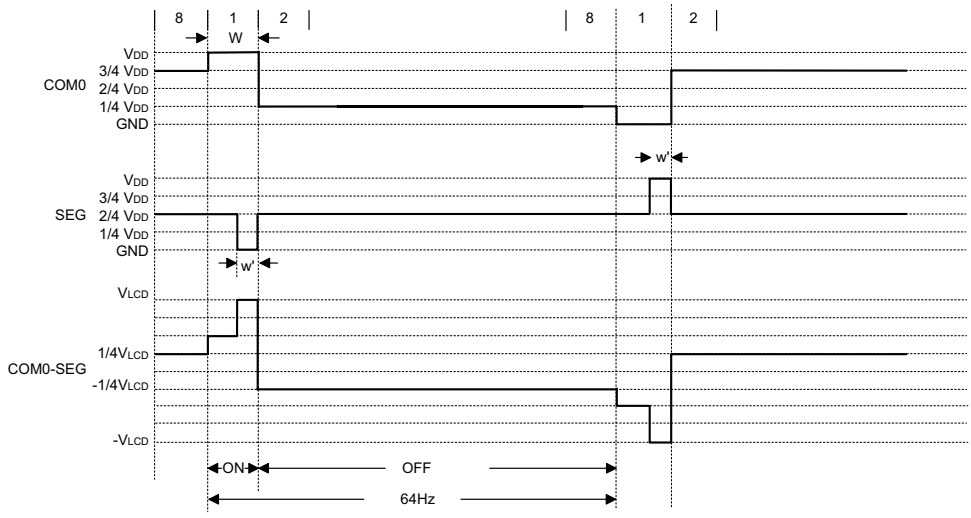
Register	Bit No.	Label	Function
COL3	0~4	P0~P4	Color3 palette
	5	—	Unused bit, read as ″0″
	6	LCD	Controls the LCD output (0=disable, 1=enable) (Default=1)
	7	RC	LCD clock source select (Default=0) 1=32768Hz crystal 0=system clock (note*)

COL3 register

Note: * When the mask option is selected to 32K x′tal disable, user should be set ″0″ to COL3.7.
But the 32K x′tal can′t be disabled in the HT-IDE2000 tools, so user should take care of this difference.

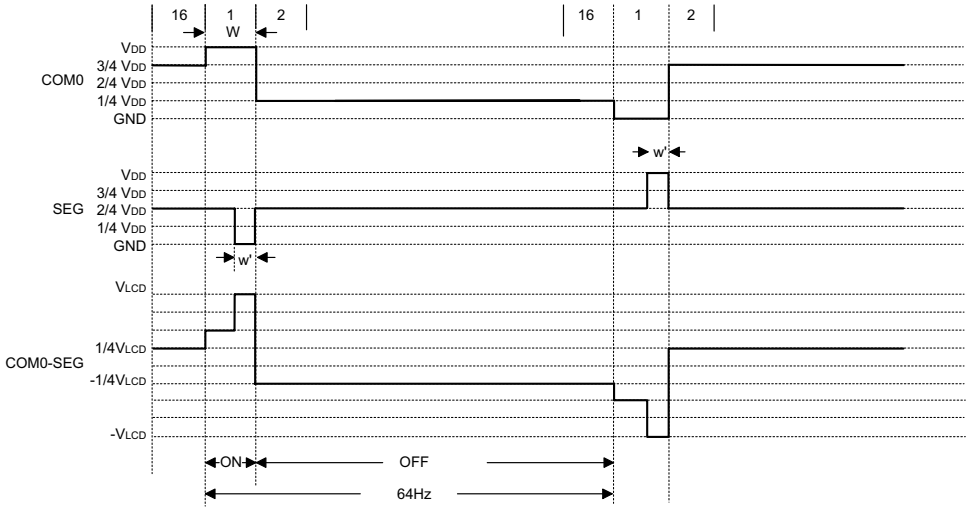
An example of an ECB driving waveform is shown below:

8 COM, 1/4 bias LCD clock source=16384Hz



Note: W' Real active segment signal width $W'=\frac{1W}{32}\sim\frac{31W}{32}$ (adjustable width by RAM Address 2BH,2CH,2DH bit0~bit4)
W: Max. active segment signal width

16COM, 1/4bias LCD clock source=32768Hz



Note: W' Real active segment signal width W'=$\frac{1W}{32}$ ~ $\frac{31W}{32}$ (adjustable width by RAM Address 2BH,2CH,2DH bit0~bit4)
W: Max. active segment signal width

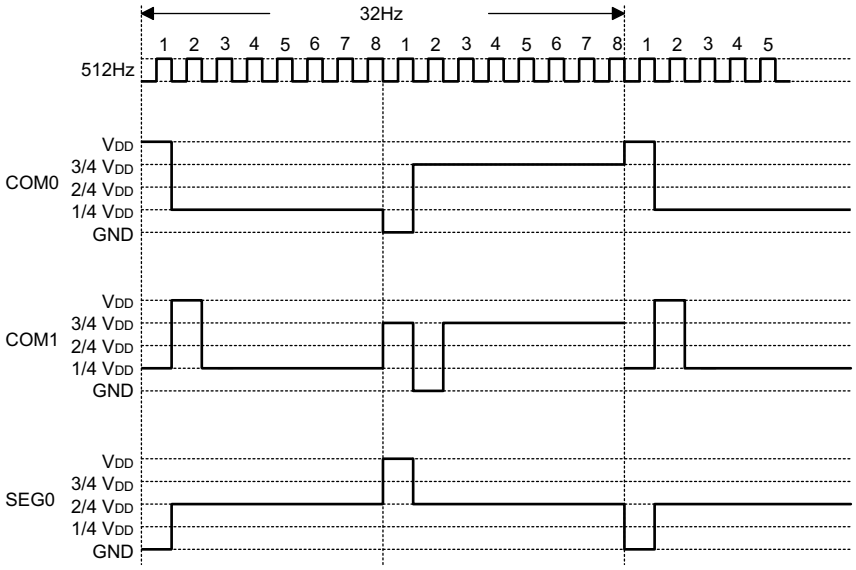
On the BW LCD Mode: Bank14 has a general purpose data RAM and for Bank15, every 1 bit stands for one dot on the LCD. If the bits is not 0, the light of the dot on the LCD will be turned on. If the bits is 0, then it will be turned off. Only MP1 can deal with the memory of this range. The contrast form of RAM location, COMMON and SEGMENT is as follows:

Address	80H	81H	82H	83H	84H	85H----------91H	92H	----	----	----	BFH
COM0	Bit0										
COM1	Bit1										
COM2	Bit2										
COM3	Bit3										
COM4	Bit4										
COM5	Bit5										
COM6	Bit6										
COM7	Bit7										
Address	C0H	C1H	C2H	C3H	C4H	C5H----------	----	----	----	FEH	FFH
COM8	Bit0										
COM9	Bit1										
COM10	Bit2										
COM11	Bit3										
COM12	Bit4										
COM13	Bit5										
COM14	Bit6										
COM15	Bit7										
	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5 ----------SEG17	SEG18	----	----	----	SEG63

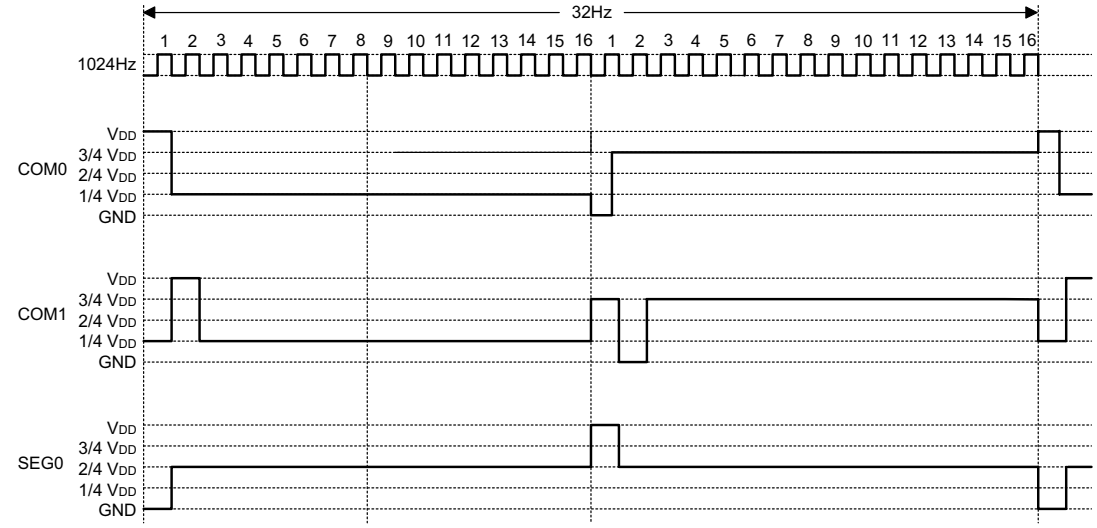
LCD display memory: (Bank15)

An example of an LCD driving waveform is shown below.

8 COM, 1/4 bias



16 COM, 1/4 bias



SSL3	SSL2	SSL1	SSL0	Description
x	0	0	0	The Pad of common 0 is connected to common 0 signal and the Pad of common 1 is connected to common 1 signal and so on.
x	0	0	1	The Pad of common 0 is connected to common 1 signal and the Pad of common 1 is connected to common 2 signal and so on.
x	0	1	0	The Pad of common 0 is connected to common 2 signal and the Pad of common 1 is connected to common 3 signal and so on.
x	0	1	1	The Pad of common 0 is connected to common 3 signal and the Pad of common 1 is connected to common 4 signal and so on.
x	1	0	0	The Pad of common 0 is connected to common 4 signal and the Pad of common 1 is connected to common 5 signal and so on.
x	1	0	1	The Pad of common 0 is connected to common 5 signal and the Pad of common 1 is connected to common 6 signal and so on.
x	1	1	0	The Pad of common 0 is connected to common 6 signal and the Pad of common 1 is connected to common 7 signal and so on.
x	1	1	1	The Pad of common 0 is connected to common 7 signal and the Pad of common 1 is connected to common 0 signal and so on.

COMR register

Indirect addressing register

Locations 00H and 02H are indirect addressing registers that are not physically implemented. Any read/write operation of [00H] and [02H] access data memory pointed to by MP0 (01H) and MP1 (03H) respectively. Reading location 00H or 02H indirectly returns the result 00H, while writing to it indirectly results in no operation.

The data movement function between two indirect addressing registers is not supported. The memory pointer registers MP0 and MP1, are 8-bit registers used to access the data memory by combining corresponding indirect addressing registers, Bank1~Bank11, Bank14 and Bank15 can use MP1 only.

Accumulator

The accumulator closely relates to ALU operations. It is also mapped to location 05H of the data memory and is the one which can operate with immediate data. The data movement between two data memory must pass through the accumulator.

Arithmetic and logic unit – ALU

This circuit performs 8-bit arithmetic and logic operation. The ALU provides the following functions:

- Arithmetic operations (ADD, ADC, SUB, SBC, DAA)
- Logic operations (AND, OR, XOR, CPL)
- Rotation (RL, RR, RLC, RRC)
- Increment and Decrement (INC, DEC)
- Branch decision (SZ, SNZ, SIZ, SDZ ....)

The ALU not only saves the results of a data operation but can also change the status register.

Status register – STATUS

This 8-bit register (0AH) contains the zero flag (Z), carry flag (C), auxiliary carry flag (AC), overflow flag (OV), power down flag (PD) and Watchdog time-out flag (TO). It also records the status information and controls the operation sequence.

With the exception of the TO and PD flags, bits in the status register can be altered by instructions like any other register. Any data written into the status register will not change the TO or PD flags. In addition, operations related to the status register may give different results from those intended. The TO and PD flags can only be changed by a system power up, Watchdog Timer overflow, executing the HALT instruction and clearing the Watchdog Timer.

The Z, OV, AC and C flags generally reflect the status of the latest operations.

In addition, on entering the interrupt sequence or executing the subroutine call, the status register will not be pushed onto the stack automatically. If the contents of the status register are important and the subroutine can corrupt the status register, the programmer must take precautions to save it properly.

Interrupt

The HTG2190 provides external and a PWM D/A interrupt and internal timer/event counter interrupts. The interrupt control register (INTC;0BH, INTCH;1EH) contains the interrupt control bits to set the enable/disable and the interrupt request flags.

Labels	Bits	Function
C	0	C is set if the operation results in a carry during an addition operation or if a borrow does not take place during a subtraction operation; otherwise C is cleared. Also it is affected by a rotate through carry instruction.
AC	1	AC is set if the operation results in a carry out of the low nibbles in addition or no borrow from the high nibble into the low nibble in subtraction; otherwise AC is cleared.
Z	2	Z is set if the result of an arithmetic or logic operation is zero; otherwise Z is cleared.
OV	3	OV is set if the operation results in a carry into the highest-order bit but not a carry out of the high-est-order bit, or vice versa; otherwise OV is cleared.
PD	4	PD is cleared when either a system power-up or executing the CLR WDT instruction. PD is set by executing the HALT instruction.
TO	5	TO is cleared by a system power-up or executing the CLR WDT or HALT instruction. TO is set by a WDT time-out.
—	6,7	Unused bit, read as ″0″

STATUS register

Once an interrupt subroutine is serviced, all other interrupts will be blocked (by clearing the EMI bit). This scheme may prevent any further interrupt nesting. Other interrupt requests may happen during this interval but only the interrupt request flag is recorded. If a certain interrupt needs servicing within the service routine, the EMI bit and the corresponding INTC bit may be set to allow interrupt nesting. If the stack is full, the interrupt request will not be acknowledged, even if the related interrupt is enabled, until the SP is decremented. If immediate service is desired, the stack must be prevented from becoming full.

All these kinds of interrupt have the wake-up capability. As an interrupt is serviced, a control transfer occurs by pushing the program counter and A15~A13 bits onto the stack and then branching to subroutines at specified location(s) in the program memory. Only the program counter are pushed and A15~A13 bits onto the stack. If the contents of the register and Status register (STATUS) are altered by the interrupt service program which corrupt the desired control sequence, the contents must be saved first.

External interrupt is triggered by a high to low transition of $\overline{\text{INT}}$ which sets the related interrupt request flag (EIF; bit 4 of INTC0). When the interrupt is enabled, and the stack is not full and the external interrupt is active, a subroutine call to location 04H will occur. The interrupt request flag (EIF) and EMI bits will be cleared to disable other interrupts.

The internal Timer/Event Counter 0 interrupt is initialized by setting the Timer/Event Counter 0 interrupt request flag (T0F; bit 5 of INTC0), caused by a Timer/Event Counter 0 overflow. When the interrupt is enabled, and the stack is not full and the T0F bit is set, a subroutine call to location 08H will occur. The related interrupt request flag (T0F) will be reset and the bit cleared to disable further interrupts.

The Timer/Event Counter 1 and timer 2/3 interrupt is operated in the same manner as Timer/Event Counter 0. The related interrupt control bits ET1I and T1F of Timer/Event Counter 1 are bit 3 and bit 6 of INTC0 respectively. While ET2I/ET3I and T2F/T3F are the related control bits and the related request flags of TMR2/TMR3, which locate at bit0/bit1 and bit4/bit5 of the INTC1 respectively.

During the execution of an interrupt subroutine, other interrupt acknowledgments are held until the RETI instruction is executed or the EMI bit and the related interrupt control bit are set to 1 (of course, if the stack is not full). To return from the interrupt subroutine, the RET or RETI instruction may be invoked. RETI will set the EMI bit to enable an interrupt service, but RET will not.

Interrupts occurring in the interval between the rising edges of two consecutive T2 pulses, will be serviced on the latter of the two T2 pulses, if the corresponding interrupts are enabled. In the case of simultaneous requests, the following table shows the priority that is applied. These can be masked by resetting the EMI bit.

Interrupt Source	Priority	Vector
External interrupt	1	04H
Timer/Event Counter 0 overflow	2	08H
Timer/Event Counter 1 overflow	3	0CH
Timer 2 overflow	4	10H
Timer 3 overflow	5	14H
PWM D/A interrupt	6	18H

The Timer/Event Counter 0/1 and timer 2/3 interrupt request flag (T0F/T1F/T2F/T3F), external interrupt request flag (EIF), PWM D/A interrupt request flag (PWMF) enable timer/ event counter 0/1/2/3 bit (ET0I/ET1I/ET2I/ ET3I), enable PWM D/A interrupt bit (PWMI), enable external interrupt bit (EEI) and enable master interrupt bit (EMI) form

Register	Bit No.	Label	Function
INTC0	0	EMI	Master (Global) interrupt (1=enable; 0=disable)
	1	EEI	External interrupt (1=enable; 0=disable)
	2	ET0I	Timer/Event Counter 0 interrupt (1=enable; 0=disable)
	3	ET1I	Timer/Event Counter 1 interrupt (1=enable; 0=disable)
	4	EIF	External interrupt request flag (1=active; 0=inactive)
	5	T0F	Internal Timer/Event Counter 0 request flag. (1=active; 0=inactive)
	6	T1F	Internal Timer/Event Counter 1 request flag. (1=active; 0=inactive)
	7	—	Unused bit, read as ″0″

INTC0 register

Register	Bit No.	Label	Function
INTC1	0	ET2I	Controls the Timer 2 interrupt. (1=enable; 0=disable)
	1	ET3I	Controls the Timer 3 interrupt. (1=enable; 0=disable)
	2	PWMI	PWM D/A interrupt (1=enable; 0=disable)
	3	—	Should be set ″0″ always.
	4	T2F	Internal Timer 2 request flag. (1=active; 0=inactive)
	5	T3F	Internal Timer 3 request flag. (1=active; 0=inactive)
	6	PWMF	PWM D/A flag (1=enable; 0=disable)
	7	—	Should be set ″0″ always.

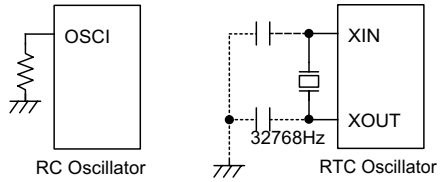
INTC1 register

the interrupt control register (INTC0/INTC1) located at 0BH/1EH in the data memory. EMI, EEI, ET0I, ET1I, ET2I, ET3I, PWMI, are used to control the enabling/disabling of interrupts. These bits prevent the requested interrupt being serviced. Once the interrupt request flags (T0F, T1F, T2F, T3F, EIF, PWMF) are set, they will remain in the INTC0/INTC1 register until the interrupts are serviced or cleared by a software instruction.

It is recommended that application programs do not use CALL subroutines within an interrupt subroutine. Interrupts often occur in an unpredictable manner or need to be serviced immediately in some applications. If only one stack is left and the interrupt enable is not well controlled, once a CALL subroutine if used in the interrupt subroutine will corrupt the original control sequence.

Oscillator configuration

There are two oscillator circuit in the HTG2190.



System and RTC oscillator

The RC oscillator signal provides the internal system clock. The HALT mode stops the system oscillator and ignores any external signal to conserve power. Only the RC oscillator is designed to drive the internal system clock. The RTC oscillator provides the timer3 and LCD driver clock source.

The RC oscillator needs an external resistor connected between OSCI and V_{SS}. The resistance value must range from 50kΩ~400kΩ

However, the frequency of the oscillator may vary with V_{DD}, temperature and the chip itself due to process variations. It is, therefore, not suitable for timing sensitive operations where accurate oscillator frequency is desired.

The RTC oscillator is used to provide clock source for LCD driver and Timer3. It can be enabled or disabled by mask option.

There is another oscillator circuit designed for the real time clock. In this case, only the 32768Hz crystal oscillator can be applied. The crystal should be connected between and XOUT, and two external capacitors along with one external resistor are required for the oscillator circuit in order to get a stable frequency.

The WDT oscillator is a free running on-chip RC oscillator, requiring no external components. Even if the system enters the power down mode, and the system clock is stopped, the WDT oscillator still runs with a period of approximately 78μs. The WDT oscillator can be disabled by mask option to conserve power.

Watchdog Timer – WDT

The WDT clock source is implemented by a dedicated RC oscillator (WDT oscillator). This timer is designed to prevent a software malfunction or sequence jumping to an unknown location with unpredictable results. The Watchdog Timer can be disabled by a mask option. If the Watchdog Timer is disabled, all the executions related to the WDT result in no operation.

Once the internal WDT oscillator (RC oscillator with a nominal period of 78μs) is selected, it is first divided by 256 (8-stages) to get the nominal time-out period of approximately 20 ms. This time-out period may vary with temperature, VDD and process variations. By invoking the WDT prescaler, longer time-out periods can be realized. Writing data to WS2, WS1, WS0 (bit 2,1,0 of the WDTS) can give different time-out periods. If WS2, WS1, WS0 all equal to 1, the division ratio is up to 1:128, and the maximum time-out period is 2.6 seconds.

WS2	WS1	WS0	Division Ratio
0	0	0	1:1
0	0	1	1:2
0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128

WDTS register

If the device operates in a noisy environment, using the on-chip RC oscillator (WDT OSC) is strongly recommended, since the HALT will stop the system clock.

The WDT overflow under normal operation will initialize a "chip reset" and set the status bit "TO". Whereas in the HALT mode, the overflow will initialize a "warm reset" only the PC and SP are reset to zero. To clear the contents of the WDT (including the WDT prescaler), three methods are adopted; external reset (a low level to $\overline{RES}$), software instructions, or a HALT instruction. The software instruction is "CLR WDT" and execution of the "CLR WDT" instruction will clear the WDT.

Power down operation – HALT

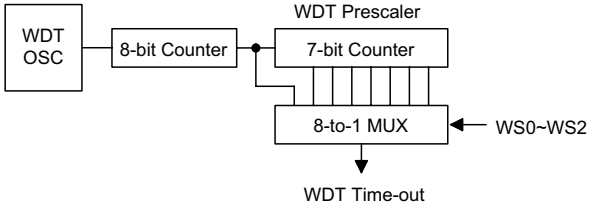
The HALT mode is initialized by a "HALT" instruction and results in the following.

- The system oscillator will turn off but the WDT oscillator keeps running (if the WDT oscillator is selected).
- The contents of the on chip RAM and registers remain unchanged.
- WDT and WDT prescaler will be cleared and recount again.
- All I/O ports maintain their original status.
- The PD flag is set and the TO flag is cleared.

The system can leave the HALT mode by means of an external reset, an interrupt, an external falling edge signal on port A or a WDT overflow. An external reset causes a device initialization and the WDT overflow performs a "warm reset". By examining the TO and PD flags, the reason for the chip reset can be determined. The PD flag is cleared when the system powers-up or executing the "CLR WDT" instruction and is set when the "HALT" instruction is executed. The TO flag is set if a WDT time-out occurs, and causes a wake-up that only resets the PC and SP. The others maintain their original status.

The port A wake-up and interrupt methods can be considered as a continuation of normal execution. Each bit in port A can be independently selected to wake up the device by a mask option. Awakening from an I/O port stimulus, the program will resume execution of the next instruction. If awakening from an interrupt, two sequences may happen. If the related interrupt is disabled or the interrupt is enabled but the stack is full, the program will resume execution at the next instruction. If the interrupt is enabled and the stack is not full, the regular interrupt response takes place.

Once a wake-up event occurs, it takes 1024 t_{SYS} (system clock period) to resume normal operation. In other words, a dummy cycle period will be inserted after a wake-up. If the wake-up results from an interrupt acknowledge, the actual interrupt subroutine will be delayed by one more cycle. If the wake-up results in next instruction execution, this will be executed immediately after a dummy period is finished. If an interrupt request



Watchdog Timer

flag is set to ″1″ before entering the HALT mode, the wake-up function of the related interrupt will be disabled.

To minimize power consumption, all I/O pins should be carefully managed before entering the HALT status.

Reset

There are 3 ways in which a reset can occur:

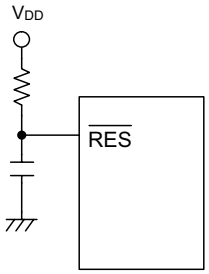
- $\overline{RES}$ reset during normal operation
- $\overline{RES}$ reset during HALT
- WDT time-out reset during normal operation

The WDT time-out during HALT is different from other chip reset conditions, since it can perform a ″warm reset″ that resets only the PC and SP, leaving the other circuits in their original state. Some registers remain unchanged during any other reset conditions. Most registers are reset to their ″initial condition″ when the reset conditions are met. By examining the PD flag and TO flag, the program can distinguish between different ″chip resets″.

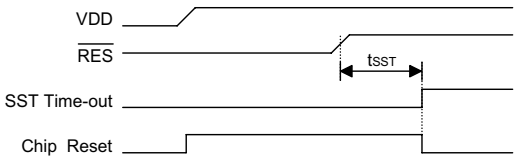
TO	PD	RESET Conditions
0	0	$\overline{RES}$ reset during power-up
u	u	$\overline{RES}$ reset during normal operation
0	1	$\overline{RES}$ wake-up HALT
1	u	WDT time-out during normal operation
1	1	WDT wake-up HALT

Note: ″u″ stands for ″unchange″

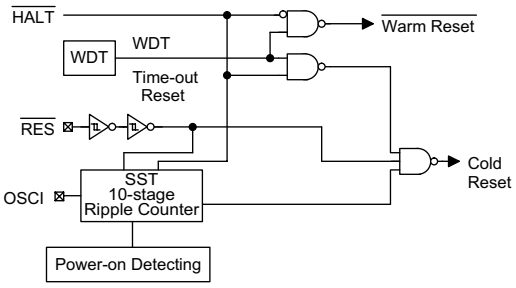
To guarantee that the system oscillator has started and stabilized, the SST (System Start-up Timer) provides an extra-delay of 1024 system clock pulses after a system power up or when awakening from a HALT state.



Reset circuit



Reset timing chart



Reset configuration

When a system power up occurs, the SST delay is added during the reset period. But when the reset comes from the $\overline{RES}$ pin, the SST delay is disabled. Any wake-up from HALT will enable the SST delay.

The functional unit chip reset status is shown in the table.

Program Counter	000H
Interrupt	Disable
Prescaler	Clear
WDT	Clear. After master reset, WDT begins counting
Timer/Event Counter (0/1/2/3)	Off
LCD Display	Enable
Pull-high of TMR0/TMR1	With pull-high resistor
Input/output Ports	Input mode
SP	Points to the top of the stack

The state of the registers is summarized in the following table:

Register	Reset (power on)	WDT time-out (normal operation)	$\overline{RES}$ reset (normal operation)	$\overline{RES}$ reset (HALT)	WDT time-out (HALT)*
TMR1H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR1C	00-0 1---	00-0 1---	00-0 1---	00-0 1---	uu-u u---
TMR0H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu

Register	Reset (power on)	WDT time-out (normal operation)	$\overline{RES}$ reset (normal operation)	$\overline{RES}$ reset (HALT)	WDT time-out (HALT)*
TMR0L	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR0C	00-0 1---	00-0 1---	00-0 1---	00-0 1---	uu-u u---
TMR2	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR2C	00-0 1000	00-0 1000	00-0 1000	00-0 1000	uu-u uuuu
TMR3	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR3C	00-0 1000	00-0 1000	00-0 1000	00-0 1000	uu-u uuuu
INTCH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
TBHP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
Program Counter	0000H	0000H	0000H	0000H	0000H
MP0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
STATUS	--00 xxxx	--1u uuuu	--uu uuuu	--01 uuuu	--11 uuuu
BP	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
INTCL	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
WDTS	0000 0111	0000 0111	0000 0111	0000 0111	uuuu uuuu
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PB	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PWMC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PWM	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
SERC	-111 1111	-111 1111	-111 1111	-111 1111	-111 1111
SERDATA	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
COL0	---0 0000	---0 0000	---0 0000	---0 0000	---0 0000
COL1	---0 0000	---0 0000	---0 0000	---0 0000	---0 0000
COL2	---0 0000	---0 0000	---0 0000	---0 0000	---0 0000
COL3	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
DAL	0000 ----	0000 ----	0000 ----	0000 ----	0000 ----
DAH	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
VOLC	0000 ----	0000 ----	0000 ----	0000 ----	0000 ----
PCC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PD	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PDC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PE	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
X′TAL	0000 1100	0000 1100	0000 1100	0000 1100	0000 1100

Note: ″*″ stands for ″warm reset″

″u″ stands for ″unchange″

″x″ stands for ″unknown″

″–″ stands for ″unknown″

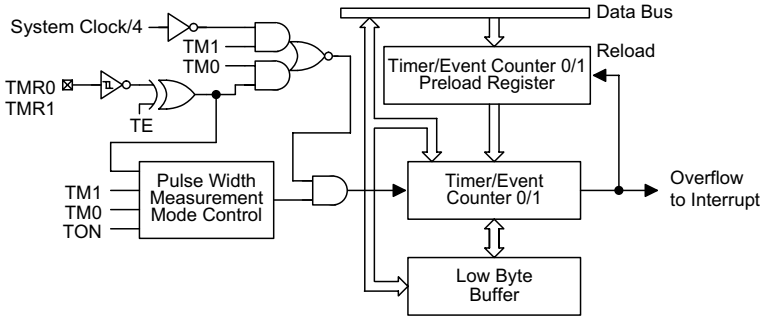
Timer/Event Counter 0/1

Two timer/event counters are implemented in the HTG2190. The Timer/Event Counter 0 and 1 contain 16-bit programmable count-up counters whose clock may come from an external source or the system clock divided by 4. It should be noted that, if the Timer/Event Counter 0 is selected, the TMR0 pin should be enabled by mask option. The pull-high resistor can be disabled or enabled by software instruction.

Using the internal instruction clock, there is only one reference time-base. The external clock input allows the user to count external events, measure time intervals or pulse width, or to generate an accurate time base.

There are three registers related to Timer/Event Counter 0; TMR0H (0CH), TMR0L (0DH), TMR0C (0EH). Writing to TMR0L only writes the data into a low byte buffer. Writing to TMR0H will write the data and the contents of the low byte buffer into the Timer/Event Counter 0 preload register (16-bit) simultaneously. The Timer/Event Counter 0 preload register is changed only by a write to TMR0H operation. Writing to TMR0L will keep the Timer/Event Counter 0 preload register unchanged.

Reading TMR0H will also latch the TMR0L into the low byte buffer to avoid false timing problems. Reading the TMR0L only returns the value from the low byte buffer which may be a previously loaded value. In other words, the low byte of Timer/Event Counter 0 cannot be read di-



Timer/Event Counter 0/1

Label	Bits	Function
—	0~2	Unused bit, read as ″0″
TE	3	Defines the TMR0/TMR1 active edge of timer/event counter (0=active on low to high; 1=active on high to low)
TON	4	Enable/disable timer counting (0=disabled; 1=enabled)
—	5	Unused bit, read as ″0″
TM1, TM0	7, 6	Define the operating mode (TM1, TM0) 01=Event count mode (external clock) 10=Timer mode (internal clock) 11=Unused 00=Unused

TMR0C/TMR1C register

Label	Bits	Function
FAST	0	0=speed-up 32K x'tal (default) 1=Non-speed-up 32K x'tal
RTF	1	Select the R-to-F function 0=disable R-to-F (default), 1=enable R-to-F
RTMR0	2	Select the TMR0 pull-high resistor (1=with pull-high; 0=without pull-high)
RTMR1	3	Select the TMR1 pull-high resistor (1=with pull-high; 0=without pull-high)
VDET	4	Supply voltage detection circuit (1=enable VDET, 0=disable VDET)
—	5~7	Unused bit, read as ″0″

X'TALC register

rectly. It must read the TMR0H first to ensure that the low byte contents of Timer/Event Counter 0 are latched into the buffer.

There are three registers related to the Timer/Event Counter 1; TMR1H (0FH), TMR1L (10H), TMR1C (11H). The Timer/Event Counter 1 operates in the same manner as Timer/Event Counter 0.

The TMR0C is the Timer/Event Counter 0 control register, which defines the Timer/Event Counter 0 options. The Timer/Event Counter 1 has the same options as the Timer/Event Counter 0 and is defined by TMR1C.

The timer/event counter control registers define the operating mode, counting enable or disable and active edge.

The TM0, TM1 bits define the operating mode. The event count mode is used to count external events, which implies that the clock source comes from an external (TMR0/TMR1) pin. The timer mode functions as a normal timer with the clock source coming from the instruction clock. The pulse width measurement mode can be used to count the high or low level duration of an external signal (TMR0/TMR1). The counting method is based on the instruction clock.

In the event count or timer mode, once the timer/event counter starts counting, it will count from the current contents in the timer/event counter to FFFFH. Once overflow occurs, the counter is reloaded from the timer/event counter preload register and generates a corresponding interrupt request flag (T0F/T1F; bit 5/6 of INTC) at the same time.

To enable the counting operation, the Timer ON bit (TON; bit 4 of TMR0C/TMR1C) should be set to 1. In the pulse width measurement mode, TON will be cleared automatically after the measurement cycle is complete. But in the other two modes TON can only be reset by instruction. The overflow of the timer/event counter is one of the wake-up sources. No matter what the operation mode is, writing a 0 to ET0I/ET1I can disable the corresponding interrupt service.

In the case of a timer/event counter OFF condition, writing data to the timer/event counter preload register will also reload that data to the timer/event counter. But if the timer/event counter is turned on, data written to the timer/event counter will only be kept in the timer/event counter preload register. The timer/event counter will continue to operate until an overflow occurs.

When the timer/event counter (reading TMR0H/TMR1H) is read, the clock will be blocked to avoid errors. As this may result in a counting error, this must be taken into consideration by the programmer.

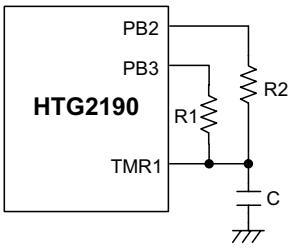
Low voltage detect function (LVD)

The HTG2190 provides low voltage detect to monitor the supply voltage of devices. The user can use the x′talc.4 to enable/disable (1/0) the LVD function and read the LVD detect status (0/1) from x′talc.4 otherwise, the LVD function is disabled. If the x′talc.4 is ″1″, means low battery otherwise if still work (V_D=(V22+150mV)± 100mV, V_D means voltage detect value).

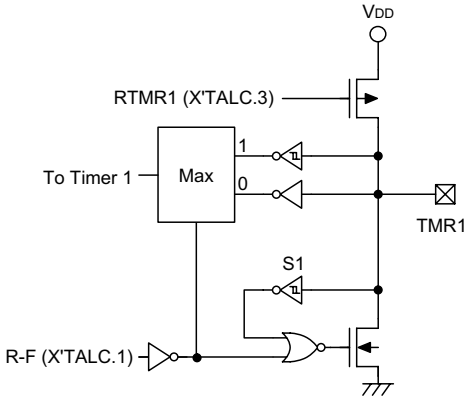
R to F function

The HTG2190 provides an ″R to F″ (Resistor to Frequency) function for temperature measurement and so on.

The application circuit is shown below.



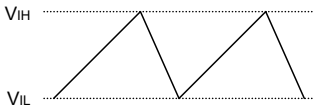
R to F application circuit



R to F structure

R1 is a fixed resistor about 10kΩ for reference, while R2 is a thermistor whose resistance is variable according to the temperature. C is a capacitance about 2200pF for charge and discharge purposes. It should be noted that, if the R to F function is selected, the pull-high resistors of TMR1. The TMR1 pull-high resistor can be disabled by software instruction. The related frequency of R1 is measured by setting PB3 to be an output pin and PB2 to be an input pin. Then set the PB3 to be a source current to charge the capacitance C. If the charged voltage ar-

rives as V_{IH} (input high voltage) of S1 gate, the NMOS will be turned on for discharge. The same manner can be used for measuring the related frequency of R2, but the roles of PB2 and PB3 should be exchanged.



Charge/discharge waveform

During execution of R to F function, the TMR1 should be set as event counter to count the charge/discharge waveform. Since the falling edge of the charge/discharge waveform is sharper than the rising edge, it is recommended that the active edge of the event counter on high to low be defined by setting TE (bit 3 of TMR1C).

For other detailed operation about Timer/Event Counter 1, refer to a related section of the Timer/Event Counter 0/1.

An example of a software program written for the purpose of measuring related frequency of R2 is shown below.

```
SET   PBC.3      ;  set PB.3 to be input pin
CLR   PBC.2      ;  set PB.2 to be output pin
SET   PB.2       ;  output source current at PB.2
CLR   X'TALC.3   ;  disable TMR1 pull-high resistor
Mov   A, 48H     ;  set TMR1 to be event counter
Mov   TMR1C, A   ;  and active edge on high to low
CLR   TMR1L      ;  set TMR1L and TMR1H to be "00"
CLR   TMR1H      ;
SET   TMR1C.4    ;  turn on TMR1
SET   XT'ALC.1   ;  turn on R to F function
```

Timer 2/3

Timer 2 is an 8-bit counter, and its clock source come from the system clock divided by an 8-stage prescaler. There are two registers related to timer 2 ; TMR2 (21H) and TMR2C (22H). Two physical registers are mapped to TMR2 location; writing to TMR2 makes the starting value be placed in the Timer 2 preload register and reading the TMR2 obtains the contents of the timer 2 counter. The TMR2C is a control register, which defines the division ratio of the prescaler and counting enable or disable.

Writing data to B2, B1 and B0 (bit 2, 1, 0 of TMR2C) can generate various clock source.

Once the timer 2 starts counting, it will count from the current contents in the counter to FFH. Once an overflow occurs, the counter is reloaded from the preload register, and generates an interrupt request flag (T2F; bit 4 of the INTCH). To enable the counting operation, the timer on bit (TON ; bit 4 of TMR2C) should be set to "1".

For proper operation, bit 6 of TMR2C should be set to "1", and bit3, bit7 should be set to "0".

Timer 2 can also be used as a buzzer output by setting PB.0 and AUD to be PWM1 and PWM2 output respectively by mask option. When the PWM1/PWM2 is selected, setting 2FH.6/2FH.7 to "1" will select PFD/PFDB output and setting 2FH.6/2FH.7 to "0" will select PWM1/PWM2 output. When the PFD/PFDB function is selected, setting 2FH.4/2FH.5 to "1" will enable PFD/PFDB output and setting 2FH.4/2FH.5 to "0" will disable PFD/PFDB output. PFD Frequency= T2f/[(256-TMR2)×2].

TMR2C			T2f
B2	B1	B0	
0	0	0	SYS CLK/2
0	0	1	SYS CLK/4
0	1	0	SYS CLK/8
0	1	1	SYS CLK/16
1	0	0	SYS CLK/32
1	0	1	SYS CLK/64
1	1	0	SYS CLK/128
1	1	1	SYS CLK/256

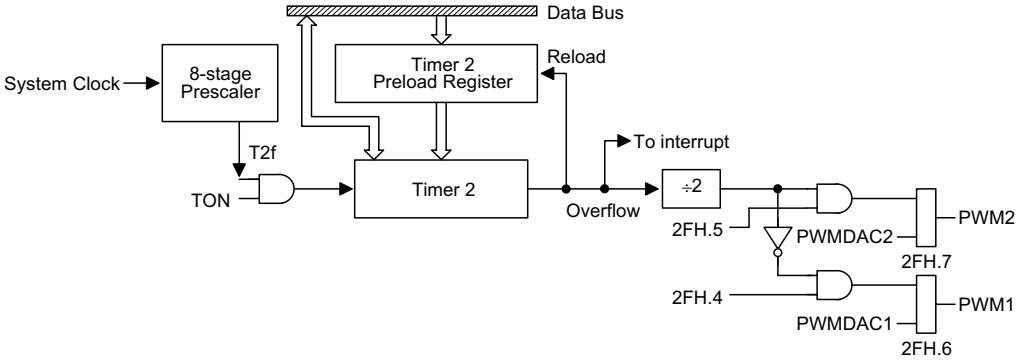
TMR2C Bit4 to enable/disable timer counting (0=disable;1=enable)

TMR2C Bit3, always write "0"

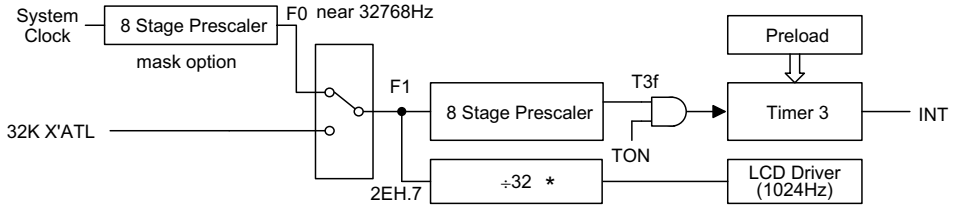
TMR2C Bit5, always write "0"

TMR2C Bit6, always write "1"

TMR2C Bit7, always write "0"



Timer 2



* If the 8 COM B/W mode is selected, the value of the divider is 64.
If the 16 COM B/W mode is selected, the value of the divider is 32.
If the 8 COM ECB mode is selected, the value of the divider is 2.
If the 16 COM ECB mode is selected, the value of the divider is 1.

Timer 3

Label	Bits	Function
SSL3~0	3~0	LCD common used
PFD	4	Enable/disable PFD output (0=disable, 1=enable)
PFDB	5	Enable/disable PFDB output (0=disable, 1=enable)
PWM1	6	Select PFDB/PWM1 output (0=PWM1, 1=PFDB)
PWM2	7	Select PFD/PWM2 output (0=PWM2, 1=PFD)

COMR register

TMR3C			T3f
B2	B1	B0	
0	0	0	F1/2
0	0	1	F1/4
0	1	0	F1/8
0	1	1	F1/16
1	0	0	F1/32
1	0	1	F1/64
1	1	0	F1/128
1	1	1	F1/256

F1 can select 4 frequencies by mask option

Auto Mask Option	F0
System Clock near 512kHz	$f_{SYS}/16$
System Clock near 1024kHz	$f_{SYS}/32$
System Clock near 2048kHz	$f_{SYS}/64$
System Clock near 4096kHz	$f_{SYS}/128$

Time base frequency = T3f / (256 − TMR3)

TMR3C Bit 4 to enable/disable timer counting (0=disable; 1=enable)

TMR3C Bit6, always write ″1″

TMR3C Bit7, always write ″0″

TMR3C Bit 3, always write ″0″

Timer 3 has the same structure and operating manner with timer 2, except for clock source and PFD function. The timer 3 can be used as a time base to generate a regular internal interrupt. The clock source of timer 3 can come from RTC OSC (X'tal 32kHz) or system clock divided by an 8-stage prescaler. If the RTC mask option is enabled, a 32kHz crystal is needed to connect across XIN and XOUT pins. The 32kHz signal is processed by an 8-stage prescaler to generate various counting clock for timer 3. There are two registers related to timer 3; TMR3 (24H) and TMR3C (25H). Writing data to B2, B1, B0 (bit 2, 1, 0 of TMR3C) can generate various counting clock.

Input/output ports

There are 39 bidirectional input/output lines in the HTG2190, labeled from PA to PE, which are mapped to the data memory of [12H], [14H], [16H], [18H], [1AH], respectively. All these I/O ports can be used for input and output operations. For input operation, these ports are non-latching, that is, the inputs must be ready at the T2 rising edge of instruction MOV A,[m] (m=12H, 14H, 16H, 18H, 1AH). For output operation, all data is latched and remain unchanged until the output latch is rewritten.

Each I/O line has its own control register (PAC, PCC, PDC, PEC) which controls the input/output configuration. With this control register, CMOS output or Schmitt trigger input with or without pull-high resistor structures can be reconfigured dynamically (i.e., on-the-fly) under software control. To function as an input, the corresponding latch of the control register must write ″1″. The pull-high resistance will be exhibited automatically if the pull-high option is selected. The input source also depends on the control register. If the control register bit is ″1″ input will read the pad state. If the control register bit is ″0″ the contents of the latches will move to the internal bus. The latter is possible in ″read-modify-write″ instruction. For output function, CMOS is the only configuration. These control registers are mapped to locations 13H, 15H, 17H, 19H, 1BH.

After a chip reset, these input/output lines stay at Schmitt trigger input with pull-high resistor. Each bit of these input/output latches can be set or cleared by the SET [m].i or CLR [m].i (m=12H, 14H, 16H, 18H, 1AH) instruction.

Some instructions first input data and then follow the output operations. For example, the SET [m].i, CLR [m].i, CPL [m] and CPLA [m] instructions read the entire port states into the CPU, execute the defined operations (bit-operation), and then write the results back to the latches or the accumulator.

Each line of port A has a wake-up capability. PC, PD and PE can be selected as segment output by mask option. If the segment output is selected, the related I/O register (PC, PD and PE) cannot be used as a general purpose register. Reading the register will result to an unknown state.

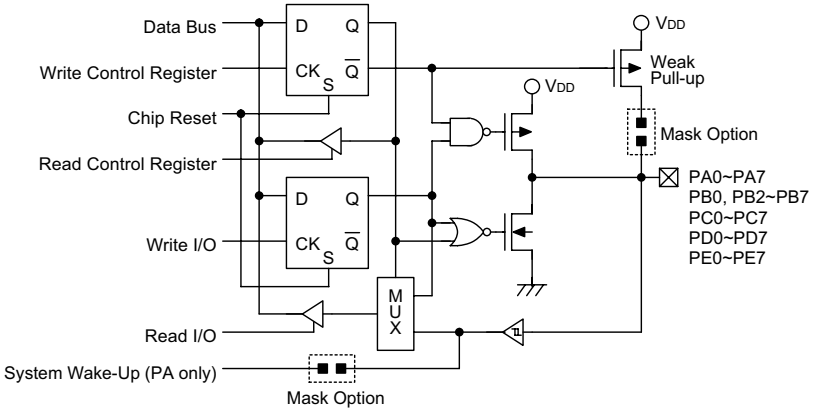
PWM interface

The HTG2190 provides an 8 bit (bit7 is a sign bit) PWM D/A interface, which is good for speech synthesis. The user can record or synthesize the sound and digitize it into the program ROM.

These sound could be played back in sequence of the function as designed by the internal program ROM. There are several algorithms that can be used in the HTG2190, they are PCM, μLAW, DPCM, ADPCM.....

The PWM circuit provides two pad outputs: PWM2, PWM1 which can directly drive a piezo or a 32Ω speaker without adding any external element. Refer to the Application Circuits.

The PWM clock source comes from the system clock divided by a 3-bit prescaler. Setting data to P0, P1 and P2 (bit 3, 4, 5 of 27H) can generate various clock sources. The clock source are used for PWM modulating clock and sampling clock. After setting the start bit (bit 0 of 27H) and the next falling edge coming from the prescaler, the ″DIV″ will generate a serial clock to PWM counter for modulation and PWMI for interrupt. The PWM counter latch data at the first ″F1″ clock falling edge and the start counter at ″F1″ rising edge. The PWM base frequency is 32kHz. For every 32kHz latch data once. The ″F2″ clock is synchronous with the first ″F1″ clock and it is also connected to the PWM output latch. In setting the ″start bit″ initial status, the ″PWM1 DAC″ outputs a ″low″ level and change the output status to ″high″ while the ″7-bit counter″ overflows.



Input/output ports

BZ/SP	6/7 Bit	F1	F2 (Sampling Rate)	Device
0	0	F0	F0/64	32Ω speaker
0	1	F0	F0/128	32Ω speaker
1	0	F0	F0/64	Buzzer/8Ω speaker
1	1	F0	F0/128	Buzzer/8Ω speaker

Note: ″F1″ for PWM modulation clock and F2 for sampling clock
″F0″ $f_{SYS}/(n+1)$ n=0~7 (n:3 bits preload counter)

On the above table, we can easily see that the sampling rate is dependent on the system clock. If start bit is set to ″1″, the PWM2 and PWM1 will output a GND level voltage.

Label	Bits	Function
PWM dis/EN	0	Enable/disable PWM output 0: enable, 1: disable
BZ/SP	1	Output driver select 1:buzzer ; 0:speaker
6/7 Bits	2	PWM counter bit select 1:7 bits ; 0:6 bits
P0~P2	3~5	3 bits preload counter Bit543: 000B~111B (0~7) Bit3: LSB
D0, D1	6, 7	PWMI

PWMC register

D1	D0	PWM Interrupt
0	0	1
0	1	2
1	0	4
1	1	8

The ratio of latch to interrupt

	Bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
7-bit	D7	D6	D5	D4	D3	D2	D1	D0
6-bit	D7	D6	D5	D4	D3	D2	D1	X

PWM data buffer

Note: ″X″ stands for don′t care
bit7: Sign bit

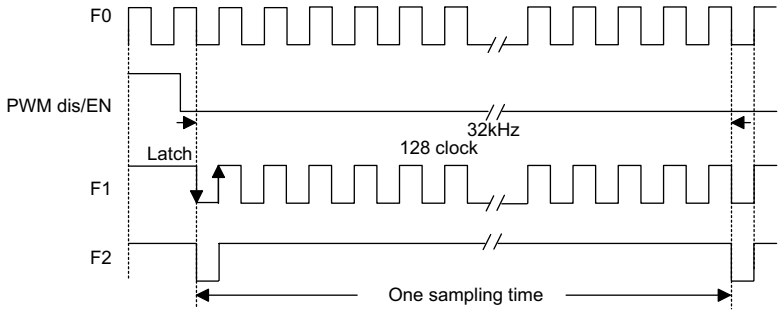
Serial I/O interface function

The serial interface of the HTG2190 has two types of operation mode: master mode and slave mode.

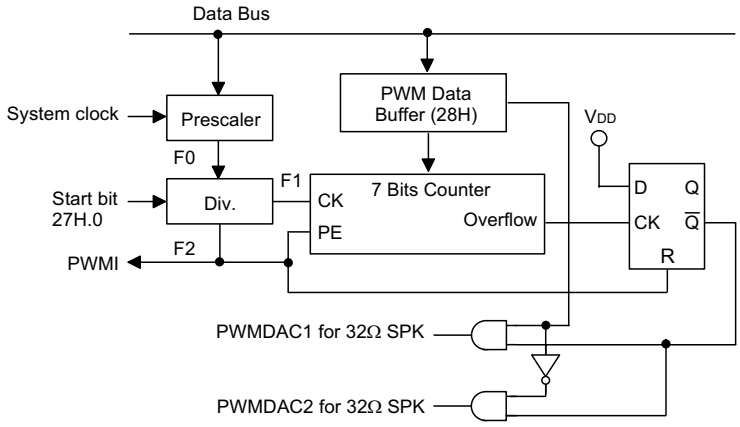
In the master mode, it uses an internal clock as synchronous clock. In the slave mode, the synchronous output from the external (master side) serial device is input.

The master mode and slave mode are selected through registers SERC.2 and SERC.3; when the master mode is selected, a synchronous clock may be selected from among 2 types as shown in table.

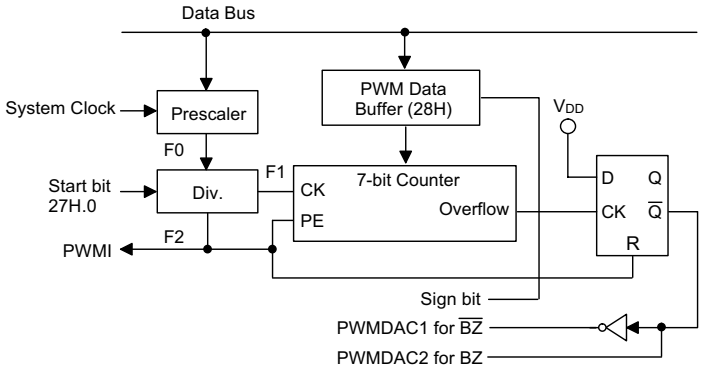
SERC.3	SERC.2	Mode	Synchronous Clock
1	1	Slave mode	External clock
1	0	Master mode	SCLKS
0	1	Master mode	SCLKX
0	0	No used	



7 bits PWM counter bit



27H.1=0 speaker



27H.1=1 buzzer

SOUT is latch data at H to L by fixed mode

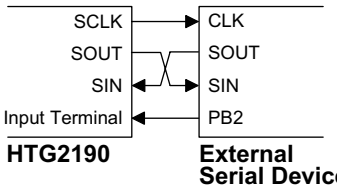
SERC			SCLKX	SCLKS
b6	b5	b4		
0	0	0	16K	$f_{SYS}/4$
0	0	1	8K	$f_{SYS}/8$
0	1	0	4K	$f_{SYS}/16$
0	1	1	2K	$f_{SYS}/32$
1	0	0	1K	$f_{SYS}/64$
1	0	1	512	$f_{SYS}/128$
1	1	0	256	$f_{SYS}/256$
1	1	1	128	$f_{SYS}/512$

At initial reset, the slave mode (external clock mode) is selected. Moreover, the synchronous clock, along with the input/output of the 8 bits serial data, is controlled as follows:

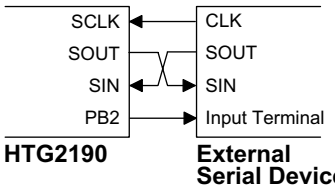
- At master mode, after output of 8 clock from the SCLK terminal, clock output is automatically suspended and SCLK terminal is fixed at high level.
- At slave mode, after input of 8 clocks to the terminal, subsequent clock inputs are masked.

Serial data output

By setting the parallel data to serial data registers SERDATA individually and writing ″0″ to SERC.0, it synchronizes with the synchronous clock and serial data is output at the SOUT terminal.



(a) Master mode



(b) Slave mode

When the output of the 8 bits data from D0~D7 is completed, the interrupt factor flag is set to ″1″ and interrupt is generated. Moreover, the interrupt can be masked by the interrupt mask register INTC.

Note, however, that regardless of the setting of the interrupt mask register, the interrupt factor flag is set to ″1″ after output of the 8 bits data.

Serial data input

By writing ″0″ to SERC.0, the serial data is input from the SIN terminal, synchronizes with the synchronous clock, and is sequentially read in the 8 bits shift register.

The input data will be fetched at the rising edge (.1=″0″) of SCLK. When the input of the 8 bits data from D0~D7 is completed, the interrupt factor flag EIF is set to ″1″ and interrupt is generated. Moreover, the interrupt can be masked by the interrupt mask register SERDATA. Note, however, that regardless of the setting of the interrupt mask register, the interrupt factor flag is set to ″1″ after input of the 8 bits data.

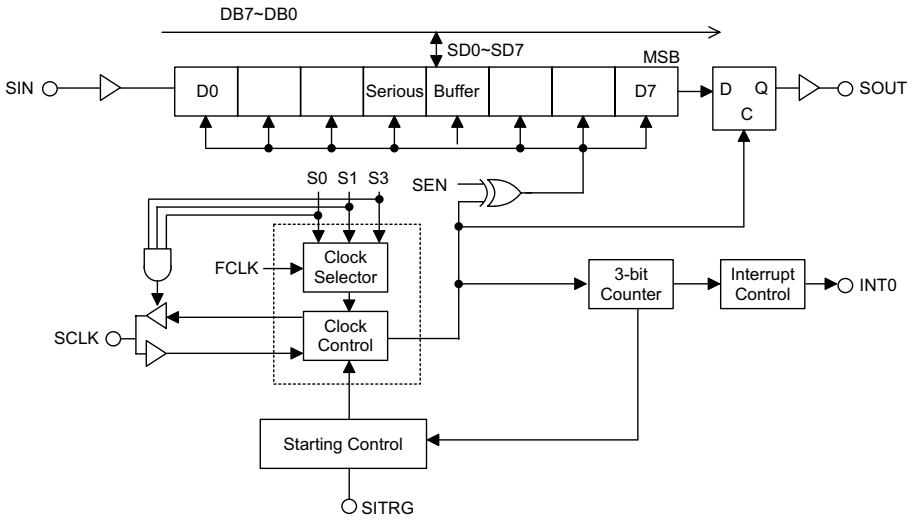
```
;* use master mode
;* serial clock=F1/2
;*

serial      equ     2Ah             ;serialdatabuffer
serc        equ     29h             ;serial control register
data        equ     51h             ;
            org             004h
            jmp             serial_op   ;serial
;
;
serialap:                           ;serial function testing (PA=16h)
            mov             a,55h
            mov             data,a
            mov             serial,a    ;write data to serial register
            set             pbc         ;set port B I/P
serail_I:
            clr             pbc.5       ;serial clock output
            mov             a,00000111b ;SCLKX=F1/2
            mov             serc,a      ;latch data (h->1)
wait:
            sz              pb.2        ;
            jmp             wait
            set             EEI
            clr             serc.0      ;trigger the serial interface start
            ret
serial_op:
            mov             a,data      ;reload data to register
            mov             serial,a
            clr             serc.0      ;enable serial
            ret
```

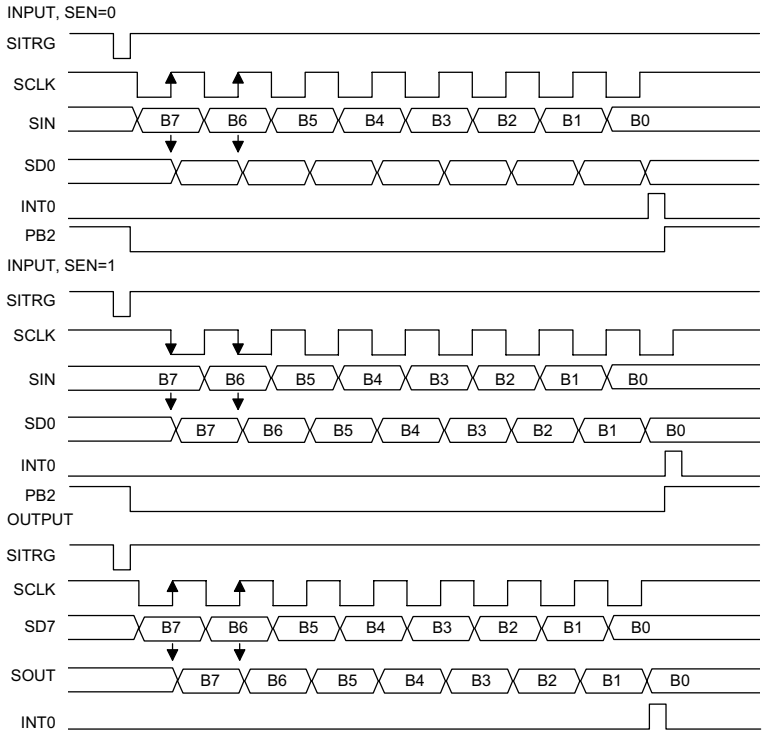
```
;* use slave mode
;* serial clock=external clock
;*

            org         004h
            jmp         serial_ip   ;serial interrupt

;
serialap:                           ;serial function testing (PA=16h)
            clr         pac         ;set port A O/P
            mov         a,serial    ;show initial data from serial buffer
            mov         pa,a
serial_i:
            set         pbc.5       ;serial clock input
            set         serc.2
            set         serc.3      ;select salve mode
            set         serc.1      ;latch data (h->1)
            clr         serc.0      ;enable trigger
            clr         pbc.2       ;set pb.2 output pin
            set         EEI
            set         pb.2
wait:
            sz          pb.2        ;
            jmp         wait
            mov         a,data      ;trigger the serial interface start
            mov         pa,a        ;show register data to port A
            ret
serial_ip:
            mov         a,serial
            mov         data,a
            clr         pb.2
            reti
```



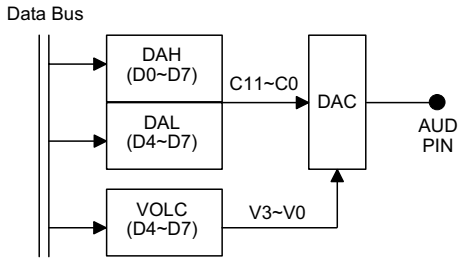
Serial I/O



Note: * B0~B7 means 8 bits
** The serial data transfer from Hi bit to Low bit.

Serial data

The audio output



The HTG2190 series provide a 12-bits current type DAC devices for driving an external 8Ω speaker through an external NPN transistor. The programmer must write the voice data to the register DAL (30H) and DAH (31H). Only 12 bits which include the high nibble of DAL and the whole byte of DAH are used. The correct procedure for DAC output as shown below, high nibble data of DAL must be written next at first, and then the DAH data is written.

There are 16 scales of volume controllable level that are provided for the current type DAC output. The programmer only writes the volume control data to the VOLC register (32H). Only the high nibble of VOLC are used. Note that writing 0H to the high nibble of VOLC does not denote mute output, this means there is still leakage from AUD pin through external NPN transistor, and also external 8Ω speaker leak current. Only load 00h to DAH-DAL will turn off DAC and prevent leakage.

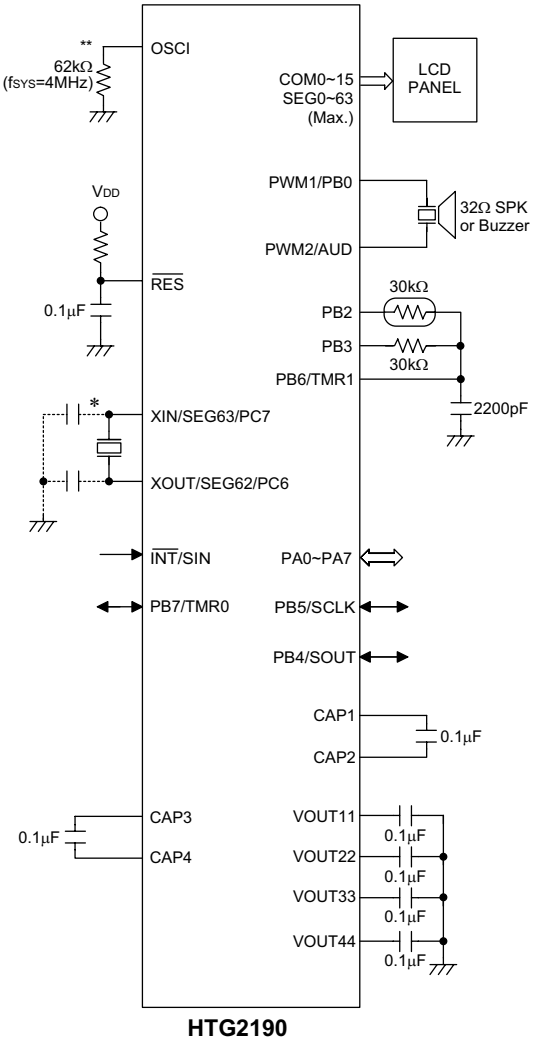
Mask option

The following shows many kinds of mask options in the HTG2190. All these option should be defined on order to ensure proper system functioning.

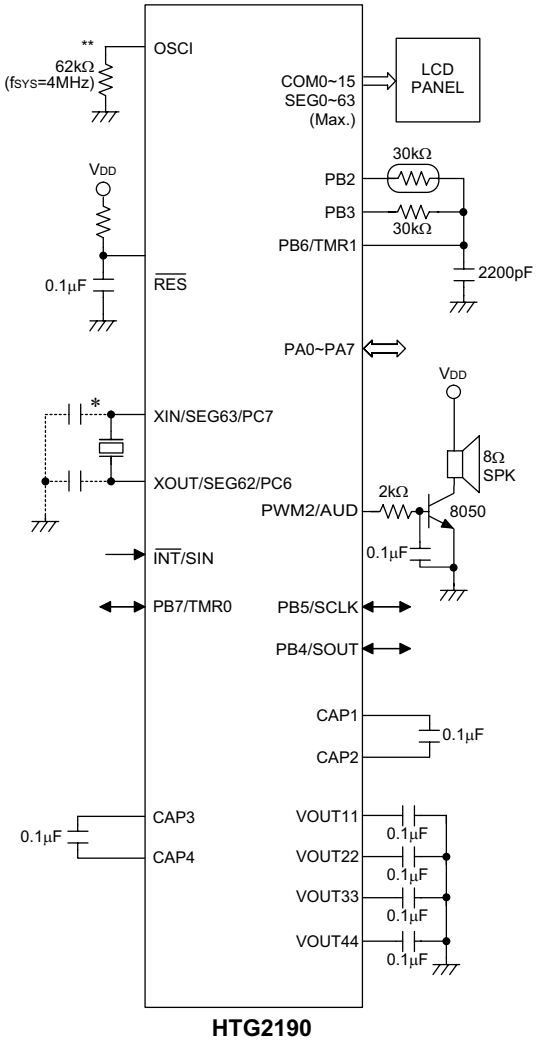
No.	Mask Option
1	WDT enable or disable selection. WDT can be enable or disable by mask option.
2	Wake-up selection. This option defines the wake-up function activity. External I/O pins (PA only) all have the capability to wake-up the chip from a HALT mode by a following edge.
3	External interrupt input pin share with other function selection. $\overline{INT}$/SEG37: $\overline{INT}$ can be set as an external interrupt input pin or LCD segment output pin.
4	I/O pins share with other functions selection. PB0/PWM1: PB0 can be set as I/O pin or positive audio PWM output pin. PWM2/AUD: PWM2 can be set as negative PWM output pin or current type D/A output pin. PB4/SOUT: PB4 can be set as I/O pin or serial data output pin. PB5/SCLK: PB5 can be set as I/O pin or serial driving clock pin. PB6/TMR1/RTF: PB6 can be set as I/O pin, Timer/Event Counter 1 input pin or resistor to frequency input pin. PB7/TMR0: PB7 can be set as I/O pin or Timer/Event Counter 0 input pin.
5	I/O pins share with other function selection. PC0/SEG56, PC1/SEG57, PC2/SEG58, PC3/SEG59, PC4/SEG60, PC5/SEG61: PC0~PC5 can be set as I/O pins or LCD segment output pins. PC6/SEG63/XIN, PC7/SEG62/XOUT: PC6, PC7 can be set as I/O pins, LCD segment output pins or XIN, XOUT pins are connect to 32768Hz crystal.
6	Segment output pins share with other function selection. SEG55~SEG48/PD7~PD0: SEG55~SEG48 can be set as LCD segment output pins or I/O pins. SEG47~SEG40/PE7~PE0: SEG47~SEG40 can be set as LCD segment output pins or I/O pins
7	LCD common selection. There are two types of selection: 8-common or 16-common
8	LCD type selection. There are two types of selection: black/white LCD or color ECB

Application Circuits

32Ω speaker/buzzer application



8Ω speaker application



Note: * Optional capacitors can be added to get a more accurate frequency.
Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate value of external capacitors.

** R=100kΩ, f_{SYS}=2MHz
R=200kΩ, f_{SYS}=1MHz

Instruction Set Summary

Mnemonic	Description	Instruction Cycle	Flag Affected
Arithmetic			
ADD A,[m]	Add data memory to ACC	1	Z,C,AC,OV
ADDM A,[m]	Add ACC to data memory	1(1)	Z,C,AC,OV
ADD A,x	Add immediate data to ACC	1	Z,C,AC,OV
ADC A,[m]	Add data memory to ACC with carry	1	Z,C,AC,OV
ADCM A,[m]	Add ACC to data memory with carry	1(1)	Z,C,AC,OV
SUB A,x	Subtract immediate data from ACC	1	Z,C,AC,OV
SUB A,[m]	Subtract data memory from ACC	1	Z,C,AC,OV
SUBM A,[m]	Subtract data memory from ACC with result in data memory	1(1)	Z,C,AC,OV
SBC A,[m]	Subtract data memory from ACC with carry	1	Z,C,AC,OV
SBCM A,[m]	Subtract data memory from ACC with carry and result in data memory	1(1)	Z,C,AC,OV
DAA [m]	Decimal adjust ACC for addition with result in data memory	1(1)	C
Logic Operation			
AND A,[m]	AND data memory to ACC	1	Z
OR A,[m]	OR data memory to ACC	1	Z
XOR A,[m]	Exclusive-OR data memory to ACC	1	Z
ANDM A,[m]	AND ACC to data memory	1(1)	Z
ORM A,[m]	OR ACC to data memory	1(1)	Z
XORM A,[m]	Exclusive-OR ACC to data memory	1(1)	Z
AND A,x	AND immediate data to ACC	1	Z
OR A,x	OR immediate data to ACC	1	Z
XOR A,x	Exclusive-OR immediate data to ACC	1	Z
CPL [m]	Complement data memory	1(1)	Z
CPLA [m]	Complement data memory with result in ACC	1	Z
Increment & Decrement			
INCA [m]	Increment data memory with result in ACC	1	Z
INC [m]	Increment data memory	1(1)	Z
DECA [m]	Decrement data memory with result in ACC	1	Z
DEC [m]	Decrement data memory	1(1)	Z
Rotate			
RRA [m]	Rotate data memory right with result in ACC	1	None
RR [m]	Rotate data memory right	1(1)	None
RRCA [m]	Rotate data memory right through carry with result in ACC	1	C
RRC [m]	Rotate data memory right through carry	1(1)	C
RLA [m]	Rotate data memory left with result in ACC	1	None
RL [m]	Rotate data memory left	1(1)	None
RLCA [m]	Rotate data memory left through carry with result in ACC	1	C
RLC [m]	Rotate data memory left through carry	1(1)	C
Data Move			
MOV A,[m]	Move data memory to ACC	1	None
MOV [m],A	Move ACC to data memory	1(1)	None
MOV A,x	Move immediate data to ACC	1	None
Bit Operation			
CLR [m].i	Clear bit of data memory	1(1)	None
SET [m].i	Set bit of data memory	1(1)	None

Mnemonic	Description	Instruction Cycle	Flag Affected
Branch			
JMP addr	Jump unconditionally	2	None
SZ [m]	Skip if data memory is zero	1(2)	None
SZA [m]	Skip if data memory is zero with data movement to ACC	1(2)	None
SZ [m].i	Skip if bit i of data memory is zero	1(2)	None
SNZ [m].i	Skip if bit i of data memory is not zero	1(2)	None
SIZ [m]	Skip if increment data memory is zero	1(3)	None
SDZ [m]	Skip if decrement data memory is zero	1(3)	None
SIZA [m]	Skip if increment data memory is zero with result in ACC	1(2)	None
SDZA [m]	Skip if decrement data memory is zero with result in ACC	1(2)	None
CALL addr	Subroutine call	2	None
RET	Return from subroutine	2	None
RET A,x	Return from subroutine and load immediate data to ACC	2	None
RETI	Return from interrupt	2	None
Table Read			
TABRDC [m]	Read ROM code (current page) to data memory and TBLH	2(1)	None
TABRDL [m]	Read ROM code (last page) to data memory and TBLH	2(1)	None
Miscellaneous			
NOP	No operation	1	None
CLR [m]	Clear data memory	1(1)	None
SET [m]	Set data memory	1(1)	None
CLR WDT	Clear Watchdog Timer	1	TO,PD
CLR WDT1	Pre-clear Watchdog Timer	1	TO(4),PD(4)
CLR WDT2	Pre-clear Watchdog Timer	1	TO(4),PD(4)
SWAP [m]	Swap nibbles of data memory	1(1)	None
SWAPA [m]	Swap nibbles of data memory with result in ACC	1	None
HALT	Enter power down mode	1	TO,PD

Note: x: Immediate data

m: Data memory address

A: Accumulator

i: 0~7 number of bits

addr: Program memory address

√: Flag is affected

–: Flag is not affected

(1): If a loading to the PCL register occurs, the execution cycle of instructions will be delayed for one more cycle (four system clocks).

(2): If a skipping to the next instruction occurs, the execution cycle of instructions will be delayed for one more cycle (four system clocks). Otherwise the original instruction cycle is unchanged.

(3): (1) and (2)

(4): The flags may be affected by the execution status. If the Watchdog Timer is cleared by executing the CLR WDT1 or CLR WDT2 instruction, the TO and PD are cleared.
Otherwise the TO and PD flags remain unchanged.

Instruction Definition

ADC A,[m] Add data memory and carry to the accumulator

Description The contents of the specified data memory, accumulator and the carry flag are added simultaneously, leaving the result in the accumulator.

Operation ACC ← ACC+[m]+C

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

ADCM A,[m] Add the accumulator and carry to data memory

Description The contents of the specified data memory, accumulator and the carry flag are added simultaneously, leaving the result in the specified data memory.

Operation [m] ← ACC+[m]+C

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

ADD A,[m] Add data memory to the accumulator

Description The contents of the specified data memory and the accumulator are added. The result is stored in the accumulator.

Operation ACC ← ACC+[m]

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

ADD A,x Add immediate data to the accumulator

Description The contents of the accumulator and the specified data are added, leaving the result in the accumulator.

Operation ACC ← ACC+x

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

ADDM A,[m] Add the accumulator to the data memory

Description The contents of the specified data memory and the accumulator are added. The result is stored in the data memory.

Operation [m] ← ACC+[m]

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

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

Description Data in the accumulator and the specified data memory perform a bitwise logical_AND operation. The result is stored in the accumulator.

Operation ACC ← ACC ″AND″ [m]

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

AND A,x Logical AND immediate data to the accumulator

Description Data in the accumulator and the specified data perform a bitwise logical_AND operation. The result is stored in the accumulator.

Operation ACC ← ACC ″AND″ x

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

ANDM A,[m] Logical AND data memory with the accumulator

Description Data in the specified data memory and the accumulator perform a bitwise logical_AND operation. The result is stored in the data memory.

Operation [m] ← ACC ″AND″ [m]

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

CALL addr Subroutine call

Description The instruction unconditionally calls a subroutine located at the indicated address. The program counter increments once to obtain the address of the next instruction, and pushes this onto the stack. The indicated address is then loaded. Program execution continues with the instruction at this address.

Operation Stack ← PC+1
PC ← addr

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

CLR [m] Clear data memory

Description The contents of the specified data memory are cleared to 0.

Operation [m] ← 00H

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

CLR [m].i Clear bit of data memory

Description The bit i of the specified data memory is cleared to 0.

Operation [m].i ← 0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

CLR WDT Clear Watchdog Timer

Description The WDT is cleared (clears the WDT). The power down bit (PD) and time-out bit (TO) are cleared.

Operation WDT ← 00H
PD and TO ← 0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	0	0	—	—	—	—

CLR WDT1 Preclear Watchdog Timer

Description Together with CLR WDT2, clears the WDT. PD and TO are also cleared. Only execution of this instruction without the other preclear instruction just sets the indicated flag which implies this instruction has been executed and the TO and PD flags remain unchanged.

Operation WDT ← 00H*
PD and TO ← 0*

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	0*	0*	—	—	—	—

CLR WDT2 Preclear Watchdog Timer

Description Together with CLR WDT1, clears the WDT. PD and TO are also cleared. Only execution of this instruction without the other preclear instruction, sets the indicated flag which implies this instruction has been executed and the TO and PD flags remain unchanged.

Operation WDT ← 00H*
PD and TO ← 0*

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	0*	0*	—	—	—	—

CPL [m] Complement data memory

Description Each bit of the specified data memory is logically complemented (1′s complement). Bits which previously contained a 1 are changed to 0 and vice-versa.

Operation [m] ← $[\overline{m}]$

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

CPLA [m] Complement data memory and place result in the accumulator

Description Each bit of the specified data memory is logically complemented (1′s complement). Bits which previously contained a 1 are changed to 0 and vice-versa. The complemented result is stored in the accumulator and the contents of the data memory remain unchanged.

Operation ACC ← $[\overline{m}]$

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

DAA [m] Decimal-Adjust accumulator for addition

Description The accumulator value is adjusted to the BCD (Binary Coded Decimal) code. The accumulator is divided into two nibbles. Each nibble is adjusted to the BCD code and an internal carry (AC1) will be done if the low nibble of the accumulator is greater than 9. The BCD adjustment is done by adding 6 to the original value if the original value is greater than 9 or a carry (AC or C) is set; otherwise the original value remains unchanged. The result is stored in the data memory and only the carry flag (C) may be affected.

Operation If ACC.3~ACC.0 >9 or AC=1
then [m].3~[m].0 ← (ACC.3~ACC.0)+6, AC1=$\overline{AC}$
else [m].3~[m].0 ← (ACC.3~ACC.0), AC1=0
and
If ACC.7~ACC.4+AC1 >9 or C=1
then [m].7~[m].4 ← ACC.7~ACC.4+6+AC1,C=1
else [m].7~[m].4 ← ACC.7~ACC.4+AC1,C=C

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	√

DEC [m] Decrement data memory

Description Data in the specified data memory is decremented by 1.

Operation [m] ← [m]–1

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

DECA [m] Decrement data memory and place result in the accumulator

Description Data in the specified data memory is decremented by 1, leaving the result in the accumulator. The contents of the data memory remain unchanged.

Operation ACC ← [m]–1

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

HALT Enter power down mode

Description This instruction stops program execution and turns off the system clock. The contents of the RAM and registers are retained. The WDT and prescaler are cleared. The power down bit (PD) is set and the WDT time-out bit (TO) is cleared.

Operation PC ← PC+1
PD ← 1
TO ← 0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	0	1	—	—	—	—

INC [m] Increment data memory

Description Data in the specified data memory is incremented by 1

Operation [m] ← [m]+1

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

INCA [m] Increment data memory and place result in the accumulator

Description Data in the specified data memory is incremented by 1, leaving the result in the accumulator. The contents of the data memory remain unchanged.

Operation ACC ← [m]+1

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

JMP addr Directly jump

Description The program counter are replaced with the directly-specified address unconditionally, and control is passed to this destination.

Operation PC ←addr

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

MOV A,[m] Move data memory to the accumulator

Description The contents of the specified data memory are copied to the accumulator.

Operation ACC ← [m]

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

MOV A,x Move immediate data to the accumulator

Description The 8-bit data specified by the code is loaded into the accumulator.

Operation ACC ← x

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

MOV [m],A Move the accumulator to data memory

Description The contents of the accumulator are copied to the specified data memory (one of the data memories).

Operation [m] ←ACC

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

NOP No operation

Description No operation is performed. Execution continues with the next instruction.

Operation PC ← PC+1

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

OR A,[m] Logical OR accumulator with data memory

Description Data in the accumulator and the specified data memory (one of the data memories) perform a bitwise logical_OR operation. The result is stored in the accumulator.

Operation ACC ← ACC ″OR″ [m]

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

OR A,x Logical OR immediate data to the accumulator

Description Data in the accumulator and the specified data perform a bitwise logical_OR operation. The result is stored in the accumulator.

Operation ACC ← ACC ″OR″ x

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

ORM A,[m] Logical OR data memory with the accumulator

Description Data in the data memory (one of the data memories) and the accumulator perform a bitwise logical_OR operation. The result is stored in the data memory.

Operation [m] ←ACC ″OR″ [m]

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

RET Return from subroutine

Description The program counter is restored from the stack. This is a 2-cycle instruction.

Operation PC ← Stack

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RET A,x Return and place immediate data in the accumulator

Description The program counter is restored from the stack and the accumulator loaded with the specified 8-bit immediate data.

Operation PC ← Stack
ACC ← x

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RETI Return from interrupt

Description The program counter is restored from the stack, and interrupts are enabled by setting the EMI bit. EMI is the enable master (global) interrupt bit.

Operation PC ← Stack
EMI ← 1

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RL [m] Rotate data memory left

Description The contents of the specified data memory are rotated 1 bit left with bit 7 rotated into bit 0.

Operation [m].(i+1) ← [m].i; [m].i:bit i of the data memory (i=0~6)
[m].0 ← [m].7

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RLA [m] Rotate data memory left and place result in the accumulator

Description Data in the specified data memory is rotated 1 bit left with bit 7 rotated into bit 0, leaving the rotated result in the accumulator. The contents of the data memory remain unchanged.

Operation ACC.(i+1) ← [m].i; [m].i:bit i of the data memory (i=0~6)
ACC.0 ← [m].7

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RLC [m] Rotate data memory left through carry

Description The contents of the specified data memory and the carry flag are rotated 1 bit left. Bit 7 replaces the carry bit; the original carry flag is rotated into the bit 0 position.

Operation [m].(i+1) ← [m].i; [m].i:bit i of the data memory (i=0~6)
[m].0 ← C
C ← [m].7

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	√

RLCA [m] Rotate left through carry and place result in the accumulator

Description Data in the specified data memory and the carry flag are rotated 1 bit left. Bit 7 replaces the carry bit and the original carry flag is rotated into bit 0 position. The rotated result is stored in the accumulator but the contents of the data memory remain unchanged.

Operation ACC.(i+1) ← [m].i; [m].i:bit i of the data memory (i=0~6)
ACC.0 ← C
C ← [m].7

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	√

RR [m] Rotate data memory right

Description The contents of the specified data memory are rotated 1 bit right with bit 0 rotated to bit 7.

Operation [m].i ← [m].(i+1); [m].i:bit i of the data memory (i=0~6)
[m].7 ← [m].0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RRA [m] Rotate right and place result in the accumulator

Description Data in the specified data memory is rotated 1 bit right with bit 0 rotated into bit 7, leaving the rotated result in the accumulator. The contents of the data memory remain unchanged.

Operation ACC.(i) ← [m].(i+1); [m].i:bit i of the data memory (i=0~6)
ACC.7 ← [m].0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

RRC [m] Rotate data memory right through carry

Description The contents of the specified data memory and the carry flag are together rotated 1 bit right. Bit 0 replaces the carry bit; the original carry flag is rotated into the bit 7 position.

Operation [m].i ← [m].(i+1); [m].i:bit i of the data memory (i=0~6)
[m].7 ← C
C ← [m].0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	√

RRCA [m] Rotate right through carry and place result in the accumulator

Description Data of the specified data memory and the carry flag are rotated 1 bit right. Bit 0 replaces the carry bit and the original carry flag is rotated into the bit 7 position. The rotated result is stored in the accumulator. The contents of the data memory remain unchanged.

Operation ACC.i ← [m].(i+1); [m].i:bit i of the data memory (i=0~6)
ACC.7 ← C
C ← [m].0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	√

SBC A,[m] Subtract data memory and carry from the accumulator

Description The contents of the specified data memory and the complement of the carry flag are subtracted from the accumulator, leaving the result in the accumulator.

Operation ACC ← ACC+$\overline{[m]}$+C

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

SBCM A,[m] Subtract data memory and carry from the accumulator

Description The contents of the specified data memory and the complement of the carry flag are subtracted from the accumulator, leaving the result in the data memory.

Operation [m] ← ACC+$\overline{[m]}$+C

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

SDZ [m] Skip if decrement data memory is 0

Description The contents of the specified data memory are decremented by 1. If the result is 0, the next instruction is skipped. If the result is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).

Operation Skip if ([m]−1)=0, [m] ← ([m]−1)

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SDZA [m] Decrement data memory and place result in ACC, skip if 0

Description The contents of the specified data memory are decremented by 1. If the result is 0, the next instruction is skipped. The result is stored in the accumulator but the data memory remains unchanged. If the result is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).

Operation Skip if ([m]−1)=0, ACC ← ([m]−1)

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SET [m] Set data memory

Description Each bit of the specified data memory is set to 1.

Operation [m] ← FFH

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SET [m]. i Set bit of data memory

Description Bit i of the specified data memory is set to 1.

Operation [m].i ← 1

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SIZ [m] Skip if increment data memory is 0

Description The contents of the specified data memory are incremented by 1. If the result is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).

Operation Skip if ([m]+1)=0, [m] ← ([m]+1)

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SIZA [m] Increment data memory and place result in ACC, skip if 0

Description The contents of the specified data memory are incremented by 1. If the result is 0, the next instruction is skipped and the result is stored in the accumulator. The data memory remains unchanged. If the result is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).

Operation Skip if ([m]+1)=0, ACC ← ([m]+1)

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SNZ [m].i Skip if bit i of the data memory is not 0

Description If bit i of the specified data memory is not 0, the next instruction is skipped. If bit i of the data memory is not 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).

Operation Skip if [m].i≠0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SUB A,[m] Subtract data memory from the accumulator

Description The specified data memory is subtracted from the contents of the accumulator, leaving the result in the accumulator.

Operation ACC ← ACC+[$\overline{m}$]+1

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

SUBM A,[m] Subtract data memory from the accumulator

Description The specified data memory is subtracted from the contents of the accumulator, leaving the result in the data memory.

Operation [m] ← ACC+[$\overline{m}$]+1

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

SUB A,x Subtract immediate data from the accumulator

Description The immediate data specified by the code is subtracted from the contents of the accumulator, leaving the result in the accumulator.

Operation ACC ← ACC+$\overline{x}$+1

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	√	√	√	√

SWAP [m] Swap nibbles within the data memory

Description The low-order and high-order nibbles of the specified data memory (1 of the data memories) are interchanged.

Operation [m].3~[m].0 ↔ [m].7~[m].4

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SWAPA [m] Swap data memory and place result in the accumulator

Description The low-order and high-order nibbles of the specified data memory are interchanged, writing the result to the accumulator. The contents of the data memory remain unchanged.

Operation ACC.3~ACC.0 ← [m].7~[m].4
ACC.7~ACC.4 ← [m].3~[m].0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SZ [m] Skip if data memory is 0

Description If the contents of the specified data memory are 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).

Operation Skip if [m]=0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SZA [m] Move data memory to ACC, skip if 0

Description The contents of the specified data memory are copied to the accumulator. If the contents is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).

Operation Skip if [m]=0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

SZ [m].i Skip if bit i of the data memory is 0

Description If bit i of the specified data memory is 0, the following instruction, fetched during the current instruction execution, is discarded and a dummy cycle is replaced to get the proper instruction (2 cycles). Otherwise proceed with the next instruction (1 cycle).

Operation Skip if [m].i=0

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

TABRDC [m] Move the ROM code (current page) to TBLH and data memory

Description The low byte of ROM code (current page) addressed by the table pointer (TBLP) is moved to the specified data memory and the high byte transferred to TBLH directly.

Operation [m] ← ROM code (low byte)
TBLH ← ROM code (high byte)

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

TABRDL [m] Move the ROM code (last page) to TBLH and data memory

Description The low byte of ROM code (last page) addressed by the table pointer (TBLP) is moved to the data memory and the high byte transferred to TBLH directly.

Operation [m] ← ROM code (low byte)
TBLH ← POM code (high byte)

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	—	—	—

XOR A,[m] Logical XOR accumulator with data memory

Description Data in the accumulator and the indicated data memory perform a bitwise logical Exclusive_OR operation and the result is stored in the accumulator.

Operation ACC ← ACC ″XOR″ [m]

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

XORM A,[m] Logical XOR data memory with the accumulator

Description Data in the indicated data memory and the accumulator perform a bitwise logical Exclusive_OR operation. The result is stored in the data memory. The 0 flag is affected.

Operation [m] ← ACC ″XOR″ [m]

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

XOR A,x Logical XOR immediate data to the accumulator

Description Data in the accumulator and the specified data perform a bitwise logical Exclusive_OR operation. The result is stored in the accumulator. The 0 flag is affected.

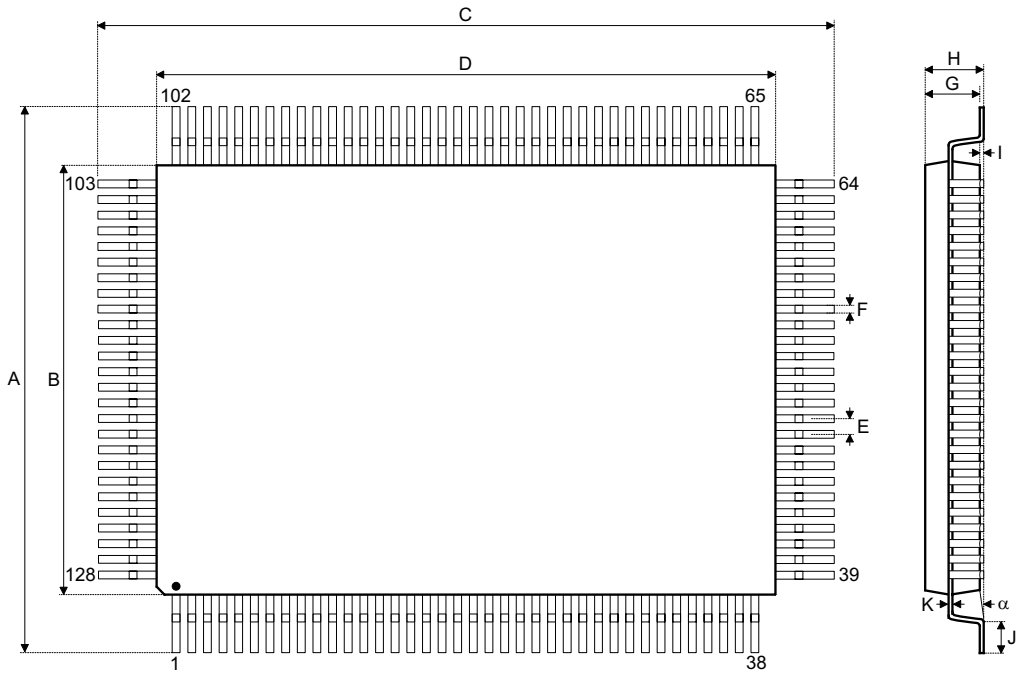
Operation ACC ← ACC ″XOR″ x

Affected flag(s)

TC2	TC1	TO	PD	OV	Z	AC	C
—	—	—	—	—	√	—	—

Package Information

128-pin QFP (14×20) outline dimensions



Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	17.00	—	17.50
B	13.90	—	14.10
C	23.00	—	23.50
D	19.90	—	20.10
E	—	0.50	—
F	—	0.20	—
G	2.50	—	3.10
H	—	—	3.40
I	—	0.10	—
J	0.65	—	0.95
K	0.10	—	0.20
α	0°	—	7°

Holtek Semiconductor Inc. (Headquarters)
No.3, Creation Rd. II, Science Park, Hsinchu, Taiwan
Tel: 886-3-563-1999
Fax: 886-3-563-1189
http://www.holtek.com.tw

Holtek Semiconductor Inc. (Taipei Sales Office)
4F-2, No. 3-2, YuanQu St., Nankang Software Park, Taipei 115, Taiwan
Tel: 886-2-2655-7070
Fax: 886-2-2655-7373
Fax: 886-2-2655-7383 (International sales hotline)

Holtek Semiconductor Inc. (Shanghai Sales Office)
7th Floor, Building 2, No.889, Yi Shan Rd., Shanghai, China 200233
Tel: 021-6485-5560
Fax: 021-6485-0313
http://www.holtek.com.cn

Holtek Semiconductor Inc. (Shenzhen Sales Office)
5/F, Unit A, Productivity Building, Cross of Science M 3rd Road and Gaoxin M 2nd Road, Science Park, Nanshan District, Shenzhen, China 518057
Tel: 0755-8616-9908, 8616-9308
Fax: 0755-8616-9533

Holtek Semiconductor Inc. (Beijing Sales Office)
Suite 1721, Jinyu Tower, A129 West Xuan Wu Men Street, Xicheng District, Beijing, China 100031
Tel: 010-6641-0030, 6641-7751, 6641-7752
Fax: 010-6641-0125

Holtek Semiconductor Inc. (Chengdu Sales Office)
709, Building 3, Champagne Plaza, No.97 Dongda Street, Chengdu, Sichuan, China 610016
Tel: 028-6653-6590
Fax: 028-6653-6591

Holmate Semiconductor, Inc. (North America Sales Office)
46729 Fremont Blvd., Fremont, CA 94538
Tel: 510-252-9880
Fax: 510-252-9885
http://www.holmate.com