60×11 Pixel Data Bank 8-Bit Mask MCU

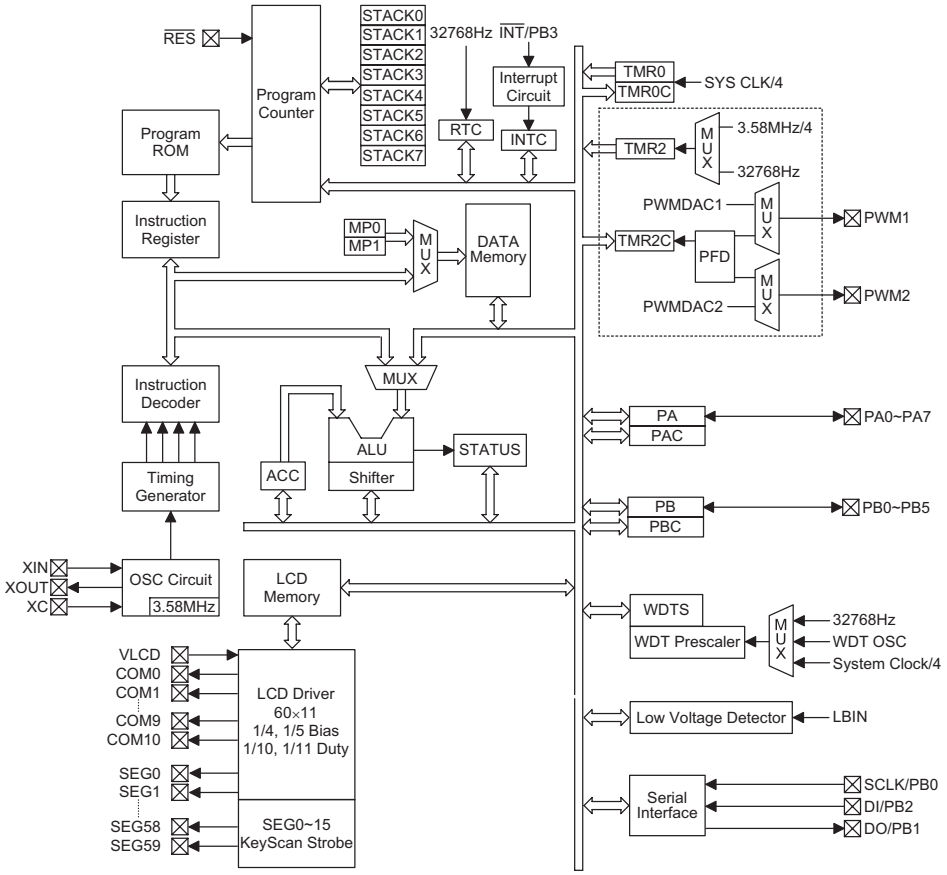
Features

- Operating voltage range: 2.4V~5.5V
- Program ROM: 32K×16 bits
- Data RAM: 2.3K×8 bits
- 16-bit table read instructions
- Eight-level subroutine nesting
- Timer
  - Two 16-bit programmable timer counters
  - Real time clock (RTC)
  - Watchdog Timer (WDT)
- Four operating modes: Idle mode, Sleep mode, Green mode and Normal mode
- Built-in 32768Hz x'tal oscillator circuit
- Build-in circuit dual system clock 32768Hz, 3.58MHz
- Build-in Low Battery detector
- 14 bidirectional I/O lines, 16 bidirectional I/O lines are share pin with segments
- LCD driver:
  - Up to a max. of 60 segments and 11 common
  - 660 dots, 1/4 or 1/5 bias capability, 1/10 or 1/11 duty, R type
  - LCD com/seg driving strength can be adjusted to compromise the display quality and current consumption, adjustable 16-level VLCD
  - Segment 0~15 supports Key Scan function
- Build-in a serial-parallel-interface hardware circuit
- Build-in a 8-bit PWM D/A hardware circuit
- 100-pin QFP package

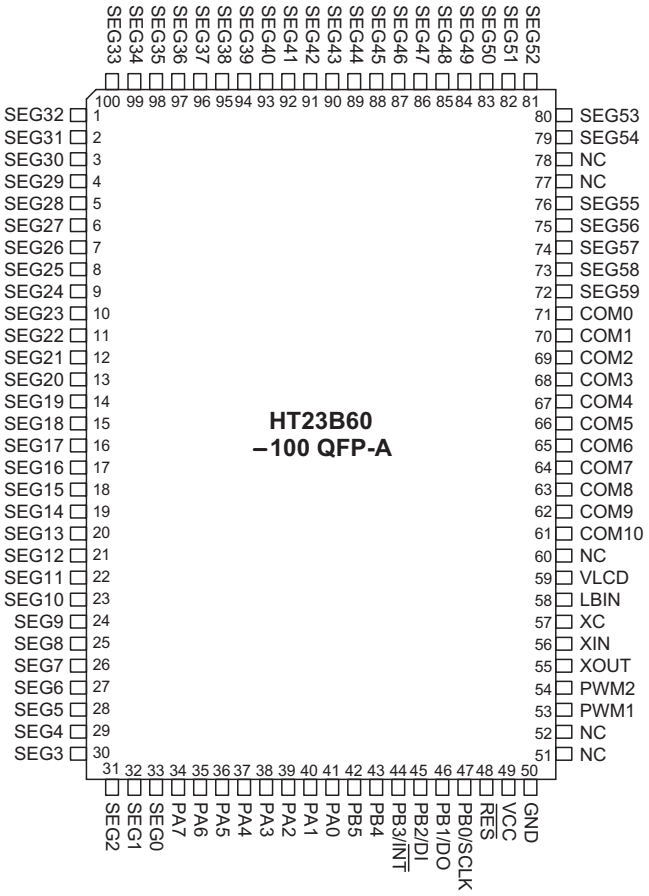
General Description

HT23B60 is an 8-bit CMOS microcontroller with various functionalities in a compact package such as SRAM, ROM I/Os, interrupt controller, timer and LCD controller/driver. It's suitable for use as electrical data bank, LCD game, calendar and speech products.

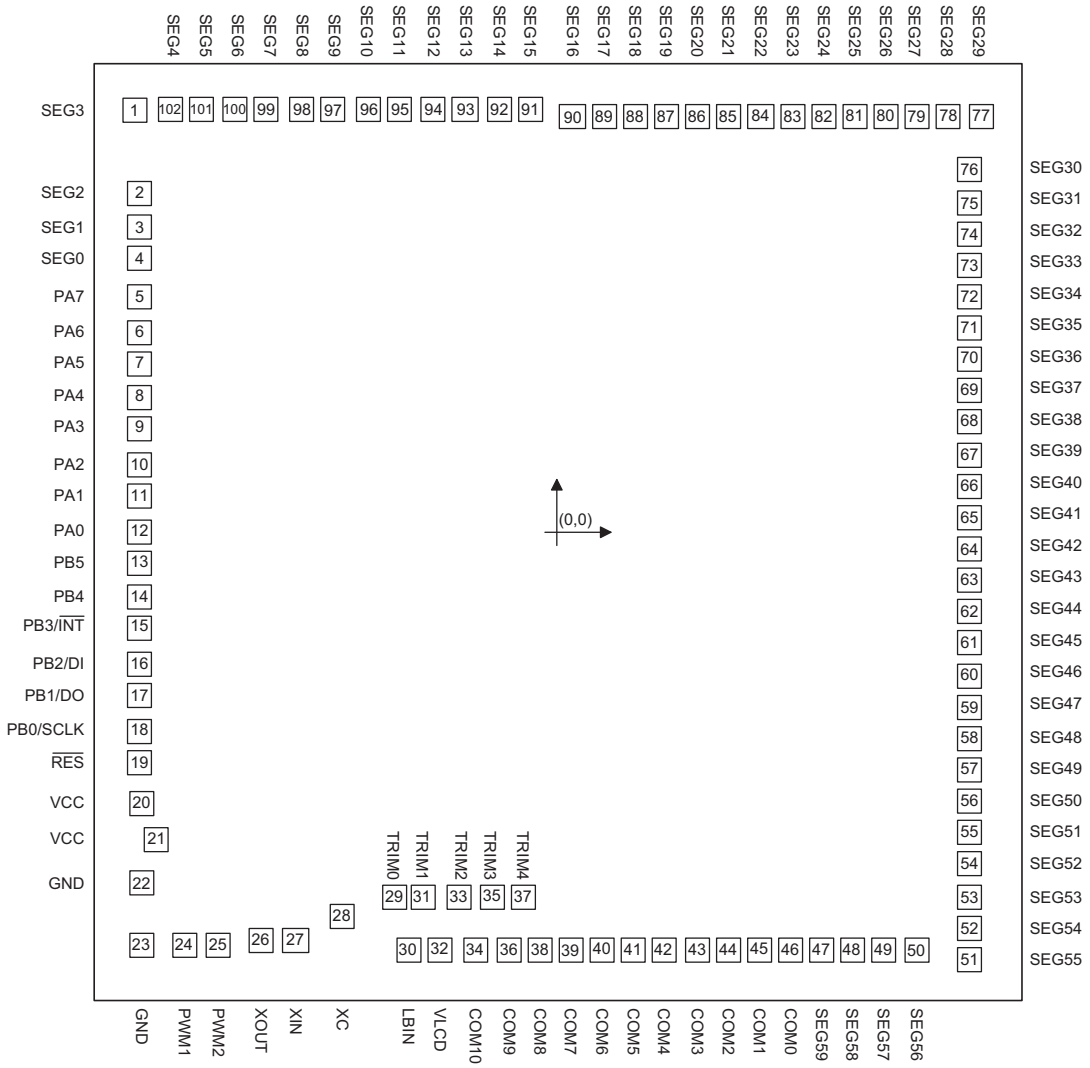
Block Diagram



Pin Assignment



Pad Assignment



* 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	−1348.345	1337.600	52	1314.745	−1252.900
2	−1334.345	1076.850	53	1314.745	−1152.900
3	−1334.345	966.250	54	1314.745	−1052.900
4	−1334.345	866.250	55	1314.745	−952.900
5	−1330.600	746.150	56	1314.745	−852.900
6	−1330.600	635.550	57	1314.745	−752.900
7	−1330.600	535.550	58	1314.745	−652.900
8	−1330.600	424.950	59	1314.745	−552.900
9	−1330.600	324.950	60	1314.745	−452.900
10	−1330.600	214.350	61	1314.745	−352.900
11	−1330.600	114.350	62	1314.745	−252.900
12	−1330.600	3.750	63	1314.745	−152.900
13	−1330.600	−96.250	64	1314.745	−52.900
14	−1330.600	−206.850	65	1314.745	47.100
15	−1330.600	−306.850	66	1314.745	147.100
16	−1330.600	−417.450	67	1314.745	247.100
17	−1330.600	−517.450	68	1314.745	347.100
18	−1330.600	−628.050	69	1314.745	447.100
19	−1330.600	−729.826	70	1314.745	547.100
20	−1323.500	−863.400	71	1314.745	647.100
21	−1282.800	−970.900	72	1314.745	747.100
22	−1323.500	−1110.150	73	1314.745	847.100
23	−1323.500	−1305.950	74	1314.745	947.100
24	−1191.900	−1307.450	75	1314.745	1047.100
25	−1083.800	−1307.450	76	1314.745	1147.100
26	−945.113	−1293.950	77	1346.855	1319.590
27	−835.109	−1293.950	78	1246.855	1319.590
28	−683.945	−1217.385	79	1146.855	1319.590
29	−519.280	−1153.900	80	1046.855	1319.590
30	−477.416	−1321.600	81	946.855	1319.590
31	−425.640	−1153.900	82	846.855	1319.590
32	−373.864	−1321.600	83	746.855	1319.590
33	−312.995	−1153.860	84	646.855	1319.590
34	−257.745	−1320.790	85	546.855	1319.590
35	−207.745	−1153.861	86	446.855	1319.590
36	−157.745	−1320.790	87	346.855	1319.590
37	−107.745	−1153.821	88	246.855	1319.590
38	−57.745	−1320.790	89	146.855	1319.590
39	42.255	−1320.790	90	46.855	1319.590
40	142.255	−1320.790	91	−84.745	1337.600
41	242.255	−1320.790	92	−184.745	1337.600
42	342.255	−1320.790	93	−295.345	1337.600
43	442.255	−1320.790	94	−395.345	1337.600
44	542.255	−1320.790	95	−505.945	1337.600
45	642.255	−1320.790	96	−605.945	1337.600
46	742.255	−1320.790	97	−716.545	1337.600
47	842.255	−1320.790	98	−816.545	1337.600
48	942.255	−1320.790	99	−927.145	1337.600
49	1042.255	−1320.790	100	−1027.145	1337.600
50	1142.255	−1320.790	101	−1137.745	1337.600
51	11314.745	−1352.900	102	−1237.745	1337.600

Pad Description

Pad No.	Pad Name	I/O	Mask Option	Description
4~1	SEG0~3	O	—	Selectable as LCD segment signal output or keyscan strobe signal.
12~5	PA0~PA7	I/O	Wake-up or None	Bidirectional 8-bit input/output port. Each bit can be configured as wake-up input by mask option. Software instructions determine the CMOS output or Schmitt trigger input with or without pull-high register by register [35H].
14~13	PB4~PB5	I/O	—	Bidirectional 2-bit input/output port Schmitt trigger input with or without pull-high register by software option or CMOS output
15	PB3/$\overline{INT}$	I/O	—	Software instructions determine the bidirectional input/output pin or external interrupt Schmitt trigger input or CMOS output. When the [INTC0].1 is set to ″1″ the PB3 will used to external interrupt input pin. For I/O pin: Schmitt trigger input with or without pull-high register by software option or CMOS output For $\overline{INT}$: Edge trigger activated on a falling edge.
16	PB2/DI	I/O or I	Serial Data Input	Can be optioned as bidirectional input/output or serial data input. For I/O pin: Schmitt trigger input or CMOS output, see mask option table for pull-high function For serial data input: serial data input without pull-high resistor
17	PB1/DO	I/O or O	Serial Data Output	Can be optioned as bidirectional input/output or serial data output. For I/O pin: Schmitt trigger input or CMOS output, see mask option table for pull-high function For serial data output: SK is a CMOS output
18	PB0/SCLK	I/O	SCLK Signal	Can be optioned as bidirectional input/output or serial interface clock signal. For I/O pin: Schmitt trigger input with or without pull-high resistor by register [36H] or CMOS output For serial interface clock signal: Use as serial I/O interface clock signal SCLK should be set as serial clock output and after 8 clocks from the SCLK terminal, clock output is automatically suspended.
19	$\overline{RES}$	I	—	Schmitt trigger reset input. Active low.
20, 21	VDD	—	—	Positive power supply
22, 23	VSS	—	—	Negative power supply, ground
24	PWM1	O	—	Positive PWM CMOS output
25	PWM2	O	—	Negative PWM CMOS output
26	XOUT	O	—	A 32768Hz crystal (or resonator) should be connected to this pin and XIN
27	XIN	I	—	A 32768Hz crystal (or resonator) should be connected to this pin and XOUT
28	XC	I	—	External low pass filter used for frequency up conversion circuit
29 31 33 35 37	TRIM0 TRIM1 TRIM2 TRIM3 TRIM4	—	—	Test pin only
30	LBIN	I	—	This pin detects battery low through external R1/R2 to determine threshold, when the low voltage detect function is disabled, the ″LBIN″ pin should be connected to VDD.
32	VLCD	I	—	LCD voltage input

Pad No.	Pad Name	I/O	Mask Option	Description
46~38, 36 34	COM0~8 COM9 COM10	O	—	LCD common signal output
102~47	SEG4~59	O	—	LCD segment signal output

Absolute Maximum Ratings

Supply Voltage .....V_{SS}–0.3V to V_{SS}+6.0V

Input Voltage ..... V_{SS}–0.5V to V_{DD}+0.5V

Storage Temperature .....–55°C to 150°C

Operating Temperature .....–10°C to 70°C

Current Drain Per Pin Excluding VDD and VSS 10mA

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	—	3V application	2.4	3.0	5.5	V
I_{DD1}	Operating Current (In Normal Mode)	3V	32768Hz on, 3.58MHz on, CPU on, LCD on, WDT on, no load	—	1	1.5	mA
I_{DD2}	Operating Current (In Green Mode)	3V	32768Hz on, 3.58MHz off CPU on, LCD off, WDT off, no load	—	8	15	μA
I_{STB1}	Standby Current 1 (In Sleep Mode)	3V	32768Hz on, 3.58MHz off, CPU off, LCD off, WDT off, no load	—	2.5	3	μA
I_{STB2}	Standby Current 2 (In Idle Mode)	3V	32768Hz off, 3.58MHz off, CPU off, LCD off, WDT off, no load	—	—	1	μA
V_{IL}	Input Low Voltage for I/O Port	3V	—	0	—	1.0	V
V_{IH}	Input High Voltage for I/O Port	3V	—	2.0	—	V_{DD}	V
V_{OL}	Output Low Voltage	3V	—	—	—	0.4	V
V_{OH}	Output High Voltage	3V	—	2.3	—	V_{DD}	V
I_{OL1}	I/O Port Sink Current	3V	V_{OL}=0.3V	8	13	—	mA
I_{OH1}	I/O Port Source Current	3V	V_{OH}=2.7V	–4	–8	—	mA
I_{OL2}	Segment, Common Output Sink Current	3V	V_{OL}=0.3V	270	480	—	μA
I_{OH2}	Segment, Common Output Source Current	3V	V_{OH}=2.7V	–100	–140	—	μA
I_{OL3}	PWM Sink Current	3V	V_{OL}=0.3V	16	26	—	mA
I_{OH3}	PWM Source Current	3V	V_{OH}=2.7V	–16	–26	—	mA
R_{PH}	Pull-high Resistance of I/O Ports	3V	—	30	60	90	kΩ
LBIN	Low Battery Detection Reference Voltage	3V	—	1.10	1.15	1.20	V

A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SYS1}	Operating System Clock (In Green Mode)	3V	—	—	32	—	kHz
f_{SYS2}	Operating System Clock (In Normal Mode)	3V	—	—	3.58	—	MHz
t_{STB}	Green Mode to Normal Mode System Frequency Stable Time	3V	—	—	—	20	ms
$t_{WDTOSC1}$	Watchdog Oscillator Period	3V	—	45	90	180	μs
$t_{WDTOSC2}$	Watchdog Time-out Period (WDT OSC)	3V	Without WDT prescaler	23	45	90	ms
t_{WDT2}	Watchdog Time-out Period (System Clock)	3V	Without WDT prescaler	—	512	—	t_{SYS}
t_{WDT3}	Watchdog Time-out Period (32kHz OSC)	3V	Without WDT prescaler	—	15.6	—	ms
t_{RESET}	RESET Input Pulse Width	—	—	1	—	—	μs
t_{SST}	System Start-up timer Period	—	—	—	1024	—	t_{SYS}
t_{INT}	Interrupt Pulse Width	—	—	1	—	—	μs

Functional Description

Execution Flow

The system clock for the HT23B60 is derived from a 32768Hz crystal oscillator. A built-in frequency up conversion circuit provides dual system clock, namely 32768Hz and 3.58MHz. The system clock is internally divided into four non-overlapping clocks. One instruction cycle consists of four system clock cycles.

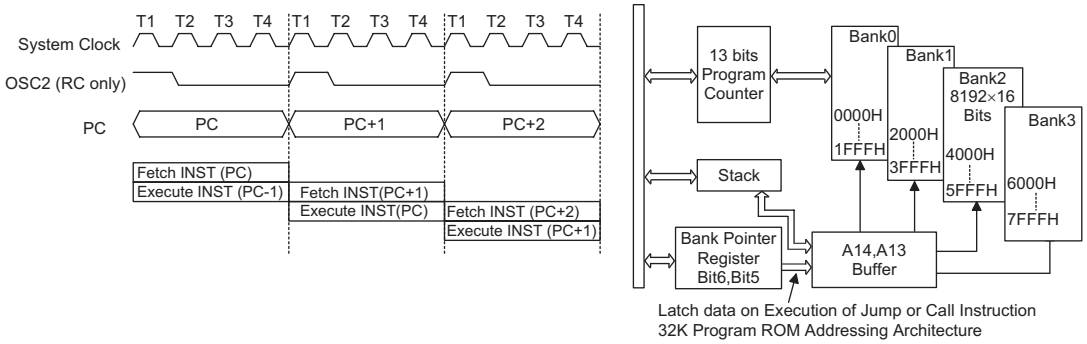
Instruction fetching and execution are pipelined in such a way that a fetch takes one instruction cycle while decoding and execution takes the next instruction cycle. However, the pipelining scheme causes each instruction to effectively execute within one cycle. If an instruction changes the program 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 1. 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.



Execution Flow

Mode	Program ROM Address														
	*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
External or Serial Input Interrupt	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Timer Counter 0 Overflow	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Timer Counter 2 Overflow	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
Keyscan Overflow	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
RTC Overflow	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
PWM Interrupt	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
Skip	PC+2														
Loading PCL	*14	*13	*12	*11	*10	*9	*8	@7	@6	@5	@4	@3	@2	@1	@0
Jump, Call Branch	BP.6	BP.5	#12	#11	#10	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
Return from subroutine	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

Program ROM Address

Note: *14~*0: Program ROM address

@7~@0: PCL bits

#12~#0: Instruction code bits

S14~S0: Stack register bits

BP.5, BP.6: Bit 5, 6 of bank pointer (04H)

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

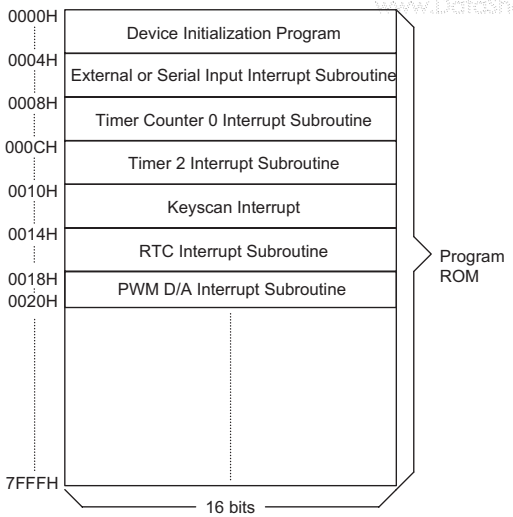
When a control transfer takes place, an additional dummy cycle is required.

Program Memory – ROM

The program memory, which contains executable program instructions, data and table information, is composed of a 32768×16 bit format. However as the PC (program counter) is comprised of only 13 bits, the remaining 2 ROM address bits are managed by dividing the program memory into 4 banks, each bank having a range between 0000H and 1FFFH. To move from the present ROM bank to a different ROM bank, the higher 2 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 15 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 15 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~BP6=00B)
  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
  This area is reserved for the timer counter 0 interrupt service program. If a timer interrupt results from a timer counter 0 overflow, and if the interrupt is enabled and the stack is not full, the program will jump to location 008H and begins execution.



Program Memory

- Location 00CH
  This area is reserved for the timer 2 interrupt service program. If a timer interrupt resulting from a timer 2 overflow, and if the interrupt is enabled and the stack is not full, the program will jump to location 00CH and begins execution.
- Location 010H
  This area is reserved for the keyscan interrupt
  When the keyscan function is enabled and the stack is not full, the program begins execution a location 010H on each common clock.
- Location 014H
  This location is reserved for real time clock (RTC) interrupt service program. When the RTC generator is enabled and time-out occurs, the RTC interrupt is enabled and the stack is not full, the program begins execution at location 014H.
- 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, if the interrupt is enabled and the stack is not full, the program begins execution at location 018H.
- Location 020H
  For best condition, this location is reserved at the beginning when writing a program.
- ROM Bank 1~3 (BP5~BP6=01B~11B)
  The range of the ROM starts from n000H to (n+1) FFFH. (n=2,4,6)
- 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 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 interrupts 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

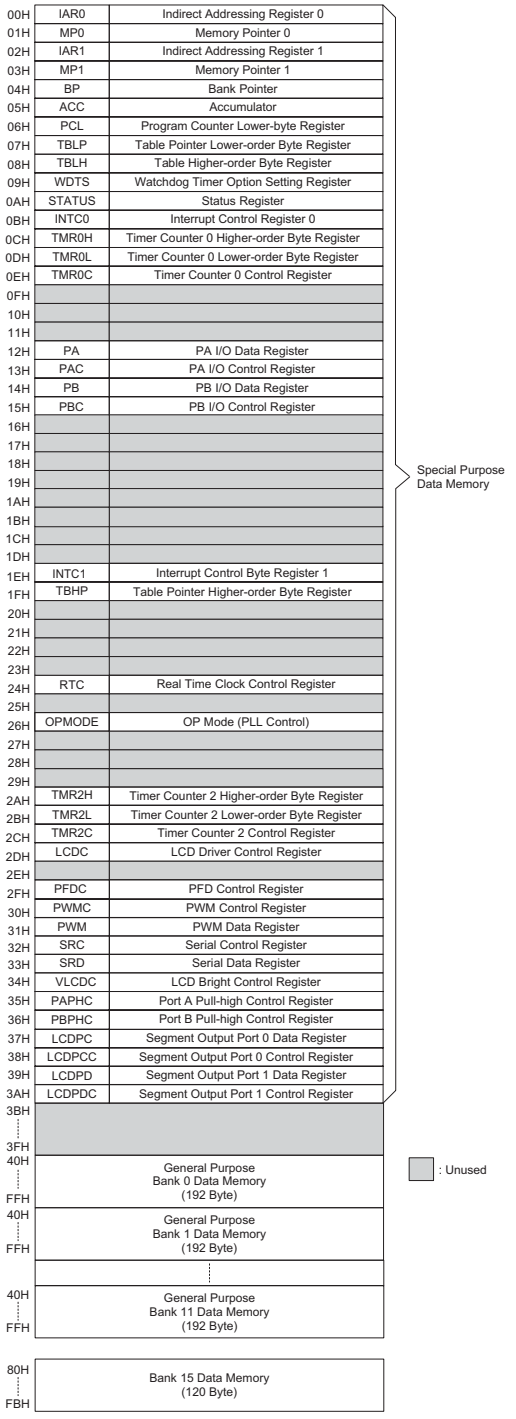
The data memory is designed with (192×12)×8 bits. The data memory is divided into two functional groups: special function registers and general purpose data memory. Most are read/write, but some are read only.

- Bank 0 (BP4~BP0=0000H)
  The Bank 0 data memory includes special purpose and general purpose memory. The special purpose memory is addressed from 00H to 3FH. The special function registers include the indirect addressing registers (IAR0:00H, IAR1:02H), timer counter 0 higher order byte register (TMR0H: 0CH), timer counter 0 lower order byte register (TMR0L: 0DH), timer counter 0 control register (TMR0C: 0EH), timer 2 lower-order byte register (TMR2L:2BH), timer 2 higher-order byte register (TMR2H:2AH), timer 2 control register (TMR2C:2CH), real timer clock control register (RTC: 24H), program counter lower-order byte register (PCL: 06H), memory pointer registers (MP0: 01H, MP1:03H), accumulator (ACC:05H), table pointer lower-order byte register (TBLP: 07H), table pointer higher-order byte register (TBHP:1FH), table higher-order byte register(TBLH:08H), status register (STATUS:0AH), interrupt control register 0 (INTC0:0BH), interrupt control register 1 (INTC1:1EH), watchdog timer option setting register (WDTS:09H), PLL control register (OPMODE:26H), LCD control register (LCDC:2DH), LCD bright control register (VLCDC:34H), LCD segment output port 0 data register (LCDPC: 37H), LCD segment output port 0 control register (LCDPCC: 38H), LCD segment output port 1 data register (LCDPD:39H), LCD segment output port 1 control register (LCDPDC:3AH), PFD control register (PFDC:2FH), PWM data register (PWM:31H), PWM control register (PWMC:30H), serial data register (SRD:33H), serial control register (SRC:32H), I/O registers (PA:12H, PB:14H) , I/O control registers (PAC:13H, PBC:15H) and pull-high control register (PAPHC:35H, PBPHC:36H).
  The general purpose data memory, addressed from 40H to FFH, is used for data and control information under instruction commands. 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

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

Table Location

Note: @7~@0: TBLP register bit7~bit0
#6~#0: TBHP register bit6~bit0
*14~*0: Current program ROM table address bit14~bit0



RAM Mapping

through the memory pointer registers (MP0;01H, MP1;03H).

- Bank 1~11(BP4~PB0=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 15 (BP4~BP0=1111B)
  The range of RAM starts from 80H to FBH (BCH~BFH can't be used). 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 HT23B60 LCD driver is 11×60. The Common output signal can be selected as 11 com or 10 com and 1/4 or 1/5 bias by mask option. The LCD driver used the voltage of VLCD pin to the power source. To adjust the view angle, the programmer can select the real LCD power by the register 34H. Some of the Segment outputs share pins with keyscan outputs (seg0~15). Whether segment output or keyscan outputs can be determined by software option.

LCD driver output can be enabled or disabled by setting the LCD (bit7 of LCDC; 2DH) without the influence of the related memory condition. Only MP1 can deal with the memory of this range. The contrast form of RAM location, COMMON and SEGMENT is as follows:

Register	Label	Bits	R/W	Function
LCDC (2DH)	—	0~1	RO	Unused bit, read as "0"
	LVEN	2	RW	To enable/disable the low voltage detection function (0: disable; 1: enable)
	—	3	R	Unused bit, read as "0"
	LVFG	4	RO	1: LBIN pin voltage is less than low voltage detection level 0: LBIN voltage is not less than low voltage detection level
	—	5, 6	R	Unused bit, read as "0"
	LCDON	7	RW	To enable/disable the LCD output (0: disable; 1: enable)

LCDC Register

VLCDC register (34H)

LCD Level	Bit7	Bit6	Bit5	Bit4	
1	0	0	0	0	0.66×VLCD
2	0	0	0	1	0.68×VLCD
3	0	0	1	0	0.70×VLCD
4	0	0	1	1	0.72×VLCD
5	0	1	0	0	0.74×VLCD
6	0	1	0	1	0.77×VLCD
7	0	1	1	0	0.80×VLCD
8	0	1	1	1	0.83×VLCD
9	1	0	0	0	0.86×VLCD
10	1	0	0	1	0.88×VLCD
11	1	0	1	0	0.90×VLCD
12	1	0	1	1	0.92×VLCD
13	1	1	0	0	0.94×VLCD
14	1	1	0	1	0.96×VLCD
15	1	1	1	0	0.98×VLCD
16	1	1	1	1	1.00×VLCD

Note: VLCD=2.4V~5.5V

The range of RAM starts from 80H to FBH

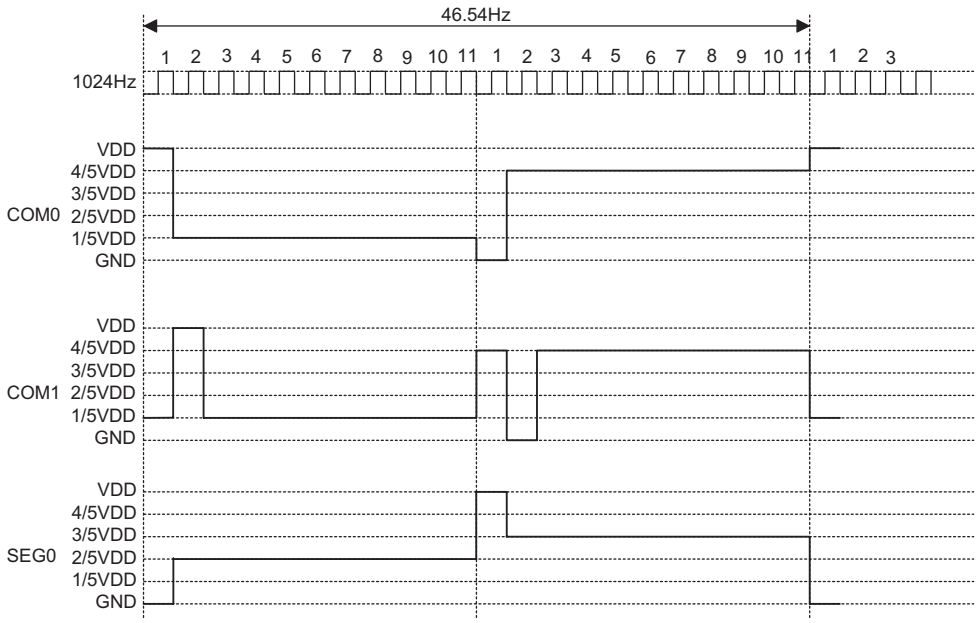
Address	80H	81H	82H	83H	84H	85H	86H- - - - B4H	B5H	B6H	B8H	B9H	BAH	BBH
COM0	Bit0												
COM1	Bit1												
COM2	Bit2												
COM3	Bit3												
COM4	Bit4												
COM5	Bit5												
COM6	Bit6												
COM7	Bit7												
Address	C0H	C1H	C2H	C3H	C4H	C5H	C6H - - - F5H	F6H	F7H	F8H	F9H	FAH	FBH
COM8	Bit0												
COM9	Bit1												
COM10	Bit2												
	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6-SEG53	SEG54	SEG55	SEG56	SEG57	SEG58	SEG59

LCD Display Memory: (Bank15)

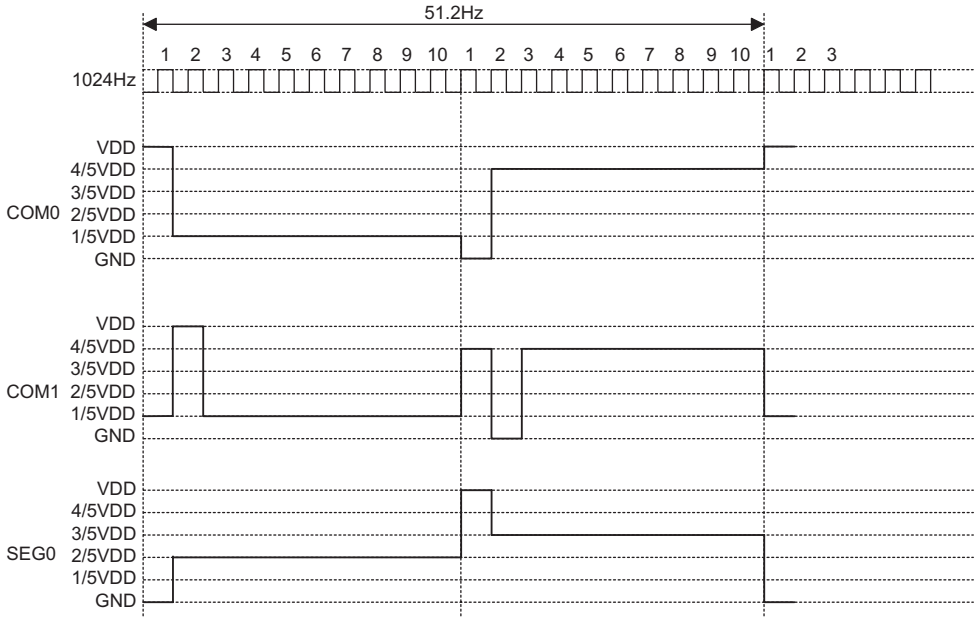
Note: C0~FB, bit3~7, R=0
BCH~BFH, R=0; FC~FFH, R=0
1: LCD pixels on
0: LCD pixels off

An example of an LCD driving waveform is shown below:

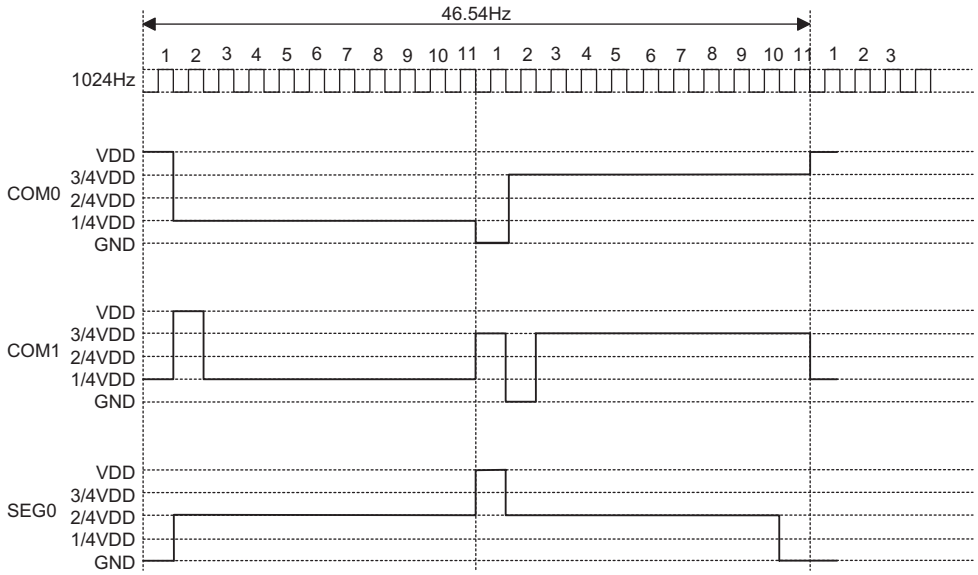
1/11 duty, 1/5 bias



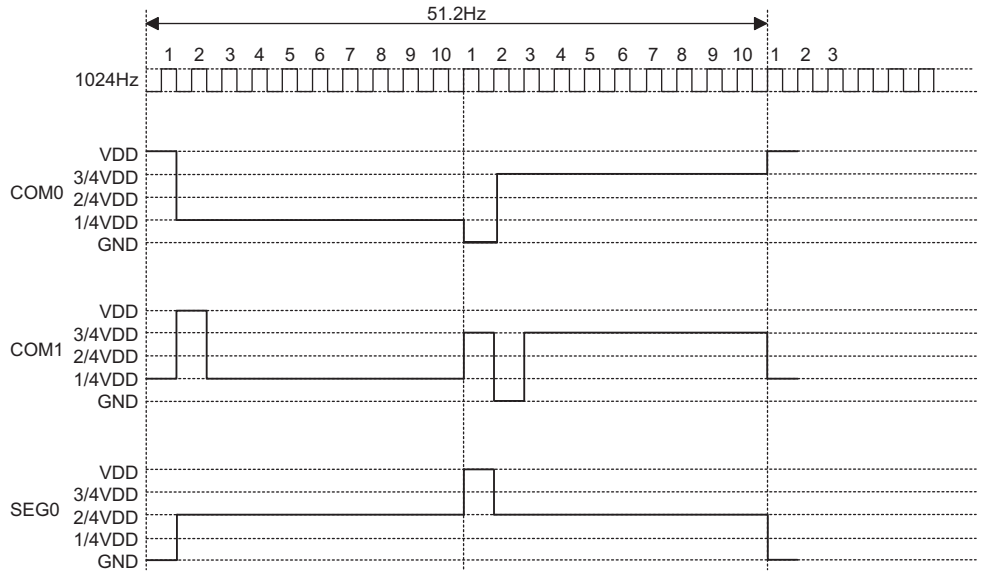
1/10 duty, 1/5 bias



1/11 duty, 1/4 bias



1/10 duty, 1/4 bias



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 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 and Bank15 can use MP1 only.

Accumulator

The accumulator is closely related to ALU operations. It is mapped to location 05H of the data memory and can also 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 (PDF) and Watchdog time-out flag (TO). It also records the status information and controls the operation sequence.

With the exception of the TO and PDF 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 PDF flags. In addition, operations related to the status register may give different results from those intended. The TO and PDF 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 HT23B60 provides external and a D/A interrupt and internal timer 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.

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 occur 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 a wake-up capability. As an interrupt is serviced, a control transfer occurs by pushing the program counter and A14~A13 bits onto the

Register	Labels	Bits	Function
STATUS (0AH)	C	0	C is set if an 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. It is also affected by a rotate through carry instruction.
	AC	1	AC is set if an 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 logical operation is zero; otherwise Z is cleared.
	OV	3	OV is set if an operation results in a carry into the highest-order bit but not a carry out of the highest-order bit, or vice versa; otherwise OV is cleared.
	PDF	4	PDF is cleared by either a system power-up or executing the ″CLR WDT″ instruction. PDF 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″

stack and then branching to subroutines at specified locations in the program memory. Only the program counter are pushed and A14~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 $\overline{INT}$ 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 counter 0 interrupt is initialized by setting the timer counter 0 interrupt request flag (T0F; bit 5 of INTC0), caused by a timer 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 EMI bit cleared to disable further interrupts.

The timer counter 2 interrupt is operated in the same manner as Timer counter 0. The related interrupt control bits ET2I and T2F of timer counter 2 are bit 3 and bit 6 of INTC0 respectively.

The real time clock interrupt is generated by a 2Hz RTC generator. When the RTC time-out occurs, the interrupt request flag RTCF will be set. When the RTC interrupt is enabled, the stack is not full and the RTCF is set, a subroutine call to location 14H will occur. The interrupt request flag RTCF and EMI bits will be cleared to disable other interrupts.

The keyscan interrupt is generated by LCD enable function. When the bit7 of the LCDC (2DH) is set ″1″, for every frame, each have a common signal all of which can generate a single interrupt. And the keyscan function have to be completed in the period of interrupt time.

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 counter 0 overflow	2	08H
Timer counter 2 overflow	3	0CH
Keyscan interrupt	4	10H
RTC interrupt	5	14H
PWM D/A interrupt	6	18H

EMI, EEI, ET0I, ET2I, EKSI, ERTCI, EPWMI are used to control the enabling/disabling of interrupts. These bits prevent the requested interrupt from being serviced. Once the interrupt request flags (EIF, T0F, T2F, KSF, RTCF, PWMF) are set by hardware or software, they will remain in the INTC0 or INTC1 registers 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.

Register	Label	Bit No.	Function
INTC0 (0BH)	EMI	0	Master (Global) interrupt (1=enable; 0=disable)
	EEI	1	External interrupt (1=enable; 0=disable)
	ET0I	2	Timer counter 0 interrupt (1=enable; 0=disable)
	ET2I	3	Timer counter 2 interrupt (1=enable; 0=disable)
	EIF	4	External interrupt request flag (1=active; 0=inactive)
	T0F	5	Internal timer counter 0 request flag. (1=active; 0=inactive)
	T2F	6	Internal timer counter 2 request flag. (1=active; 0=inactive)
	—	7	Unused bit, read as ″0″

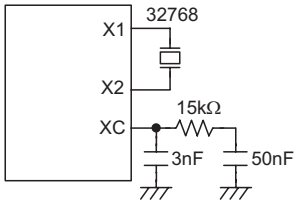
Register	Label	Bit No.	Function
INTC1 (1EH)	EKSI	0	Controls the keyscan interrupt. (1=enable; 0=disable)
	ERTCI	1	Controls the RTC interrupt. (1=enable; 0=disable)
	EPWMI	2	PWM D/A interrupt (1=enable; 0=disable)
	—	3	Should be set ″0″ always.
	KSF	4	Keyscan interrupt request flag. (1=active; 0=inactive)
	RTCF	5	RTC interrupt request flag. (1=active; 0=inactive)
	PWMF	6	PWM D/A flag (1=enable; 0=disable)
	—	7	Should be set ″0″ always.

Oscillator Configuration

There are two oscillator circuits in the controller, the external 32768Hz crystal oscillator and internal WDT RC oscillator.

The 32768Hz crystal oscillator and frequency-up conversion circuit (32768Hz to 3.58MHz) are designed for dual system clock source. It is necessary for the frequency conversion circuit to add external RC components to make up the low pass filter that stabilize the output frequency 3.58MHz (see the oscillator circuit).

The WDT RC oscillator is a free running on-chip RC oscillator, and no external components are required. Even if the system enters the Idle mode (the system clock is stopped), the WDT RC oscillator still works within a period of 65μs~78μs. When the WDT is disabled or the WDT source is not this RC oscillator, the WDT RC oscillator will be disabled.



System Oscillator Circuit

Watchdog Timer – WDT

The WDT clock source is implemented by a WDT OSC or external 32768Hz or an instruction clock (system clock divided by 4), determined by mask option. This timer is designed to prevent software malfunction or protect the sequence from jumping to an unknown location with unpredictable results. The Watchdog Timer can be disabled by mask option. If the Watchdog Timer is disabled, all the executions related to the WDT result in no operation.

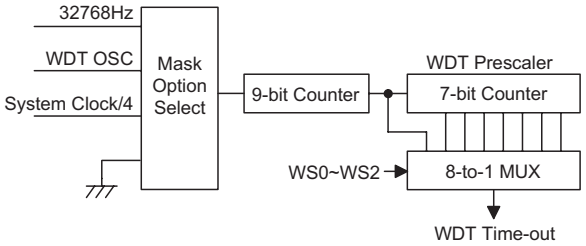
If the device operates in a noisy environment, using the on-chip WDT OSC or 32768Hz crystal oscillator is strongly recommended.

When the WDT clock source is selected, it will be first divided by 512 (9-stage) to get the nominal time-out period. By invoking the WDT prescaler, longer time-out periods can be realized. Writing data to WS2, WS1, WS0 can give different time-out periods.

The WDT OSC period is 78μs. This time-out period may vary with temperature, VDD and process variations. The WDT OSC always works for any operation mode.

If the instruction clock is selected as the WDT clock source, the WDT operates in the same manner except in the Sleep mode or Idle mode. In these two modes, the WDT stops counting and lose its protecting purpose. In this situation the logic can only be re-started by external logic.

Register	Label	Bits	R/W	Function
WDTS (09H)	WS0 WS1 WS2	0 1 2	RW	Watchdog Timer division ratio selection bits Bit 2, 1, 0=000, Division ratio=1:1 Bit 2, 1, 0=001, Division ratio=1:2 Bit 2, 1, 0=010, Division ratio=1:4 Bit 2, 1, 0=011, Division ratio=1:8 Bit 2, 1, 0=100, Division ratio=1:16 Bit 2, 1, 0=101, Division ratio=1:32 Bit 2, 1, 0=110, Division ratio=1:64 Bit 2, 1, 0=111, Division ratio=1:128
	—	7~3	RW	Unused bit. These bits are read/write-able.



Watchdog Timer

The high nibble and bit3 of the WDTS are reserved for user defined flags, which can be used to indicate some specified status.

The WDT time-out under Normal mode or Green mode will initialize "chip reset" and set the status bit "TO". But in the Sleep mode or Idle mode, the time-out will initialize a "warm reset" and only the program counter and stack pointer are reset to 0. To clear the WDT contents (including the WDT prescaler), three methods are adopted; external reset (a low level to $\overline{RES}$ pin), software instruction and "HALT" instruction.

The software instruction include "CLR WDT" and the other set "CLR WDT1" and "CLR WDT2". Of these two types of instruction, only one can be active depending on the mask option "WDT instr". If the "CLR WDT" is selected (i.e. One clear instruction), any execution of the "CLR WDT" instruction will clear the WDT. In the case wherein "CLR WDT1" and "CLR WDT2" are chosen (i.e. two clear instructions), these two instructions must be executed to clear the WDT; otherwise, the WDT may reset the chip as a result of time-out.

Controller Operation Mode

Data bank controllers support two system clocks and four operation modes. The system clock could be 32768Hz or 3.58MHz and the operation mode could be Normal, Green, Sleep or Idle mode. There are all selected by the software.

The following conditions will force the operation mode to change to Green mode:

- Any reset condition from any operation mode
- Any interrupt from Sleep mode or Idle mode
- A falling edge on any pin of Port A from Sleep mode or Idle mode

How to change the Operation Mode

- Normal mode to Green mode:

  Step 1: Clear MODE1 to 0

  After step 1, operation mode is changed to Green mode but the PLLEN status has no change.

  However, PLLEN can be cleared by software.

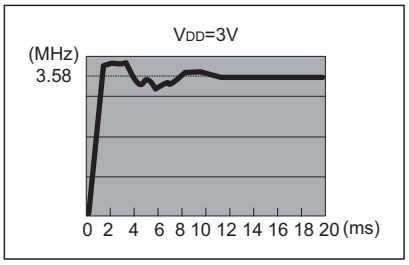
Register	Label	Bits	R/W	Function
OPMODE (26H)	—	4~0	RO	Unused bit, read as "0"
	PLLEN	5	RW	1: Enable the frequency up conversion function to generate 3.58MHz 0: Disable the frequency up conversion function to generate 3.58MHz
	MODE0	6	RW	0: Enable the 32768Hz oscillator while the "HALT" instruction is executed 1: Disable the 32768Hz oscillator while the "HALT" instruction is executed
	MODE1	7	RW	1: Select 3.58MHz as CPU system clock 0: Select 32768Hz as CPU system clock

Operation Mode Description

HALT Instruction	MODE1	MODE0	PLLEN	Operation Mode	32768Hz	3.58MHz	System Clock
Not Execute	1	X	1	Normal	ON	ON	3.58MHz
Not Execute	0	X	0	Green	ON	OFF	32768Hz
Executed	0	0	0	Sleep	ON	OFF	HALT
Executed	0	1	0	Idle	OFF	OFF	HALT

Note: "X" means don't care

- Normal mode or Green mode to Sleep mode:
  Step 1: Clear MODE0 to 0
  Step 2: Execute the "HALT" instruction
  After Step 2, the operation mode is changed to Sleep mode, the PLLEN and MODE1 are cleared to 0 by hardware.
- Normal mode or Green mode to Idle mode:
  Step 1: Set MODE0 to 1
  Step 2: Execute the "HALT" instruction
  After Step 2, the operation mode is changed to Idle mode, the PLLEN and MODE1 are cleared to 0 by hardware.
- Green mode to Normal mode:
  Step 1: Set PLLEN to 1
  Step 2*: Software delay 2ms at least
  Step 3: Set MODE1 to 1
  After Step 3, operation mode is changed to Normal mode.
  Note: * Must delay 20ms at least, if you want to use stable clock source



- Sleep mode or Idle mode to Green mode:
  Method 1: Any reset condition occurred
  Method 2: Any interrupt is active
  Method 3: A falling edge on any pin of Port A
  After any source of the above descriptions, operation mode is changed to Green mode.

The reset conditions include power on reset, external reset, WDT time-out reset. By examining the processor status flags, PDF and TO, the program can distinguish between different "reset conditions". Refer to the Reset function for detailed description.

A falling edge on port A and interrupt can be considered as a continuation of normal execution. Each bit in port A can be independently selected to wake-up the device by mask option. Awakening from Port A stimulus, the program will resume execution of the next instruction.

The interrupts from the Sleep mode or Idle mode may cause two sequences to occur in the controller. One is 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. The other is if the interrupt is enabled and the stack is not full, the regular interrupt response takes place. It is necessary to mention that if an interrupt request flag is set to "1" before entering the Sleep mode or Idle mode, the wake-up function of the related interrupt will be disabled.

Once Idle mode wake-up event occurs, it will take 1024 system clock period or SST delay time to resume to Green mode. In this case, a dummy period is inserted after a wake-up. If the wake-up results from an interrupt acknowledge signal, the actual interrupt subroutine execution will be delayed by one or more cycles. If the wake-up results in the next instruction execution, this will be executed immediately after the dummy period is completed.

To minimize power consumption, all the I/O pins should be carefully managed before entering the Sleep mode or Idle mode.

The Sleep mode or Idle mode is initialized by the "HALT" instruction and results in the following.

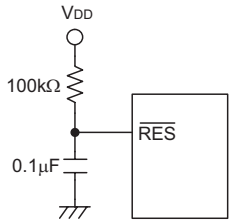
- The system clock will be turned off
- The WDT function will be disabled if the WDT clock source is the instruction clock
- The WDT function will be disabled if the WDT clock source is the 32768Hz in Idle mode
- The WDT will still function if the WDT clock source is the WDT OSC
- If the WDT function is still enabled, the WDT counter and WDT prescaler will be cleared and recounted again
- The contents of the on chip RAM and registers remain unchanged
- All the I/O ports maintain their original status
- The PDF flag is set and the TO flag is cleared by hardware.

Reset

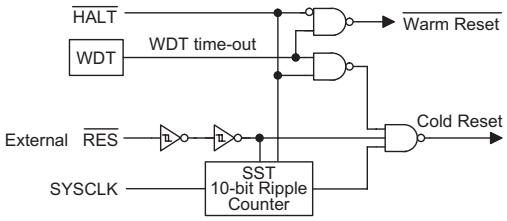
There are three ways in which a reset can occur.

- Power on reset
- A low pulse onto $\overline{RES}$ pin
- time-out

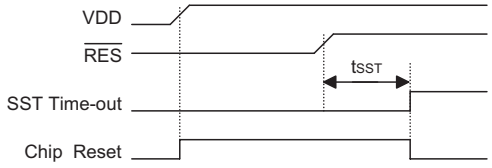
After these reset conditions, the Program Counter and Stack Pointer will be cleared to 0.



Reset Circuit



Reset Configuration



Reset Timing Chart

To guarantee that the system oscillator is started and stabilized, the System Start-up Timer or SST provides an extra-delay of 1024 system clock pulses when the system is reset or awakes from the Sleep or Idle operation mode.

By examining the processor status flags PD and TO, the software program can distinguish between the different "chip resets".

TO	PD	Reset Condition
0	0	Power on reset
u	u	External reset during Normal mode or Green mode
0	1	External reset during Sleep mode or Idle mode
1	u	WDT time-out during Normal mode or Green mode
1	1	WDT time-out during Sleep mode or Idle mode

Note: "u" stands for "unchanged"

The functional unit chip reset status are shown below:

Program Counter	000H
Interrupt	Disabled
Prescaler	Cleared
WDT	Cleared After a master reset, WDT begins counting (if the WDT function is enabled by mask option)
Timer Counter 2	Off
Input/output Port	Input mode
Stack Pointer	Points to the top of the stack
LCD Display	Disable

When the reset conditions occurred, some registers may be changed or unchanged.

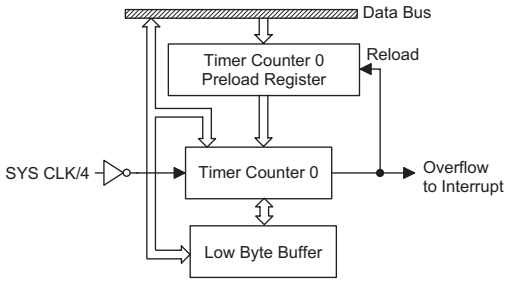
Register	Addr.	Reset Conditions				
		Power On	$\overline{\text{RES}}$ Pin	$\overline{\text{RES}}$ Pin (Sleep/Idle)	WDT	WDT (Sleep/Idle)
IAR0	00H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP0	01H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
IAR1	02H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1	03H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
BP	04H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
ACC	05H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	06H	0000H	0000H	0000H	0000H	0000H
TBLP	07H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	08H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
WDTS	09H	0000 0111	0000 0111	0000 0111	0000 0111	uuuu uuuu
STATUS	0AH	--00 xxxx	--uu uuuu	--01 uuuu	--1u uuuu	--11 uuuu
INTC0	0BH	0000 0000	0000 0000	0000 0000	0000 0000	0uuu uuuu
TMR0H	0CH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR0L	0DH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR0C	0EH	0000 1000	0000 1000	0000 1000	0000 1000	uu0u u000
PA	12H	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	13H	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PB	14H	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBC	15H	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
INTC1	1EH	0000 -000	0000 -000	0000 -000	-000 -000	0uuu 0uuu
TBHP	1FH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
RTC	24H	0000 0000	u0u0 0000	u0u0 0000	u0u0 0000	u0u0 0000
OPMODE	26H	0100 0000	01u0 0000	01u0 0000	01u0 0000	01u0 0000
TMR2H	2AH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR2L	2BH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR2C	2CH	0000 1000	0000 1000	0000 1000	0000 1000	uu0u u000
LCDC	2DH	0000 0000	0000 0000	0000 0000	0000 0000	u00u 0u00
PFDC	2FH	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uu00
PWMC	30H	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PWM	31H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
SRC	32H	0001 0000	0001 0000	0001 0000	0001 0000	uuuu uuuu
SRD	33H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
VLCD	34H	0000 0000	0000 0000	0000 0000	0000 0000	uuuu 0000
PAPHC	35H	1111 1111	1111 1111	1111 1111	1111 1111	00uu uuuu
PBPHC	36H	0011 1111	0011 1111	0011 1111	0011 1111	uuuu uuuu
LCDPC	37H	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
LCDPCC	38H	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
LCDPD	39H	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
LCDPDC	3AH	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
RAM (Data & LCD)		x	u	u	u	u

Note: "u" stands for "unchanged"

"x" stands for "unknown"

"-" stands for "unused"

Timer 0



The timer 0 contains 16-bit programmable count-up counters and the clock source come from the system clock divided by 4.

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

Reading TMR0H will also latch the TMR0L into the low byte buffer to avoid any false timing problem. Reading TMR0L returns the contents of the low byte buffer. In this case, the low byte of the timer counter 0 cannot be read directly. It must read the TMR0H first to make the low byte contents of the Timer 0 be latched into the buffer.

The TMR0C is the Timer 0 control register, which defines the Timer 0 options. The timer counter control registers define the operating mode, counting enable or disable and active edge.

If the timer counter starts counting, it will count from the current contents in the timer counter to FFFFH. Once an overflow occurs, the counter is reloaded from the timer counter preload register and at the same time generates the corresponding interrupt request flag (T0F; bit of the INTC0).

To enable the counting operation, the Timer ON bit (TON; bit 4 of the TMR0C) should be set to 1. The overflow of the timer counter is one of the wake-up sources. No matter what the operation mode is, writing a 0 to ET0I can disable the corresponding interrupt service.

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

When the timer counter (reading TMR0H) 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.

Timer 2

The Timer 2 contains 16-bit programmable count-up counters whose clock may come from the 32768 Hz oscillator or the clock source come from the system clock divided by 4.

There are three registers related to the timer counter 2; TMR2H (2AH), TMR2L (2BH), TMR2C(2CH). Writing TMR2L only writes the data into a low byte buffer, and writing TMR2H will simultaneously write the data and the contents of the low byte buffer into the timer 2 preload register (16-bit). The timer 2 preload register is changed by writing TMR2H operations and writing

Register	Label	Bits	R/W	Function
TMR0C (0EH)	—	0~2	RO	Unused bit, read as "0"
	—	3	—	Unused bit, read as "0"
	TON	4	RW	Enable/disable the timer counting (0=disabled; 1=enabled)
	—	5	—	Unused bit, read as "0"
	TM0 TM1	6 7	RW	Fixed bit 7, 6=10, internal timer mode

Register	Label	Bits	R/W	Function
TMR2C (2AH)	—	0~3	—	Unused bit, read as "0"
	TON	4	RW	Enable/disable the timer counting (0=disabled; 1=enabled)
	—	5	—	Unused bit, read as "0"
	TM0 TM1	6 7	RW	Fixed bit 7, 6=10, internal timer mode

TMR2L will keep the timer 2 preload register unchanged.

Reading TMR2H will also latch the TMR2L into the low byte buffer to avoid any false timing problem. Reading TMR2L returns the contents of the low byte buffer. In other words, the low byte of the timer counter 2 cannot be read directly. It must read the TMR2H first to make the low byte contents of Timer 2 be latched into the buffer.

The TMR2C is the Timer 2 control register, which defines the Timer 2 options. The timer counter control registers define the operating mode, counting enable or disable and active edge.

If the timer counter starts counting, it will count from the current contents in the timer counter to FFFFH. Once an overflow occurs, the counter is reloaded from the timer counter preload register and generates the corresponding interrupt request flag (T2F; bit of INTC0) at the same time.

To enable the counting operation, the Timer ON bit (TON; bit 4 of TMR2C) should be set to 1. The overflow of the timer counter is one of the wake-up sources. No matter what the operation mode is, writing a 0 to ET0I can disable the corresponding interrupt service.

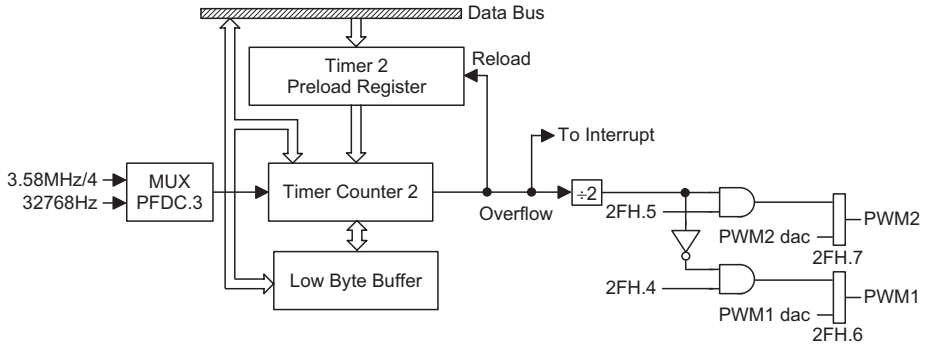
In the case of timer counter OFF condition, writing data to the timer counter preload register will also reload that data to the timer counter. But if the timer counter is turned on, data written to the timer counter will only be kept in the timer counter preload register. The timer counter will still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

When the timer counter (reading 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.

The Timer 2 can also be used as PFD output by setting PWM1 and PWM2 to be PFD and PFDB output respectively by 2FH.7 and 2FH.6. When the PFD/PFDB function is selected, setting 2FH.4/2FH.5 to ″1″ will enable the PFD/PFDB output and setting 2FH.4/2FH.5 to ″0″ will disable the PFD/PFDB output.

PFDC

Register	Label	Bits	R/W	Function
PFDC (2FH)	—	2~0	R	Unused bit, read as ″0″
	TIM2	3	RW	1: The timer 2 frequency source is 3.58MHz/4 0: The timer 2 frequency source is 32768Hz
	PFDB	4	RW	1: Enable PFDB 0: Disable PFDB
	PFD	5	RW	1: Enable PFD 0: Disable PFD
	PFDB/PWM1	6	RW	1: Enable PFDB 0: Enable PWM1
	PFD/PWM2	7	RW	1: Enable PFD 0: Enable PWM2

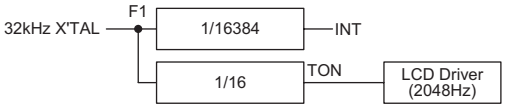


RTC & WDT & LCD Clock

- RTC function

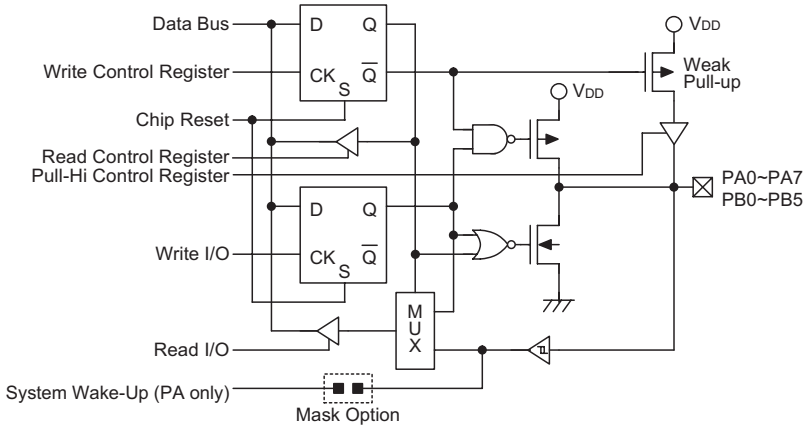
Register	Label	Bits	R/W	Function
RTC (24H)	—	6,4 ~0	RO	Unused bit, read as ″0″
	RTCEN	5	RW	Enable/disable the RTC counting (0: disable; 1: enable)
	RTCSET	7	RW	RTC time-out flag (1: active; 0: inactive)

The real time clock (RTC) is used to supply a regular internal interrupt. Its time-out period is 2Hz. If the RTC time-out occurs, the interrupt request flag RTCF and the RTCSET flag will be set to 1. The interrupt vector for the RTC is 14H. When the interrupt subroutine is serviced, the interrupt request flag (RTCF) will be cleared to 0, but the flag RTCSET maintain its original value. If RTC is time-out, the flag RTCSET and RTCF will be set to 1. The flag RTCSET can be cleared to 0 by software.



Input/Output Ports

There are 14 bidirectional input/output lines in the HT23B60, labeled PA and PB, which are mapped to the data memory of [12H], [14H], 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). For output operation, all data is latched and remains unchanged until the output latch is rewritten. Each I/O line has its own control register (PAC, PBC) to control the input/output configuration. With this control register, CMOS output or Schmitt trigger input with or without pull-high resistor (software option 35H, 36H) structures can be reconfigured dynamically under software control. To function as an input, the corresponding latch of the control register must be written a ″1″. The pull-high resistance will exhibit automatically if the pull-high option is selected. The input source also depends on the control register. If the control register bit is ″1″, the 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. After a chip reset, these input/output lines remain at high levels or floating (mask option). Each bit of these input/output latches can be set or cleared by the ″SET [m].i″ or ″CLR [m].i″ (m=12H, 14H) 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 the capability to wake-up the device. Port B are share pad, each pin function are defined by mask option, when the PB3 be used as a normal I/O port, INT function must be disable. (Set [0BH].4 to ″0″). The PB2, PB1 and PB0 share with serial data input, serial data output and serial clock. If the serial function is selected, the related I/O register (PB) cannot be used as general purpose register. Reading the register will result to an unknown state.



Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1: pull-high	1: pull-high	1: pull-high	1: pull-high	1: pull-high	1: pull-high	1: pull-high	1: pull-high
0: No pull	0: No pull	0: No pull	0: No pull	0: No pull	0: No pull	0: No pull	0: No pull

PA Pull-High Resistor

Bit7~Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Unused bit	1: pull-high	1: pull-high	1: pull-high	1: pull-high	1: pull-high	1: pull-high
	0: No pull	0: No pull	0: No pull	0: No pull	0: No pull	0: No pull

PB Pull-High Resistor

PWM Interface

The HT23B60 provides an 8 bit (bit 7 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. This sound could be played back in sequence of the functions as designed by the internal program ROM. There are several algorithms that can be used in the HT23B60, namely, PCM, μ_LAW, DPCM, ADPCM...etc.

The PWM circuit consists of seven counters. When initialized, QB goes high and when an overflow occurs, QB goes low. When the PWM controller bits 0 of the 30H are set as ″0″, each of the 128 clock will initialize the counter and load the value that come from PWM data buffer to counter. The PWM modulation can be controlled by using a different value of the PWM data buffer. A single bit can control the signal changes from the PWM1 or PWM2 output. The PWM clock source comes from the system clock divided by a 3-bit prescaler. Setting data to P0, P1 and P2 (bit3, 4, 5 of 30H) can yield various clock sources. Setting PWM controller bits D0, D1 (bit6, 7 of 30H) can control the interrupt as to how many times the counter overflows.

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

Note: F1: for PWM modulation clock and F2 for sampling clock

F0: system/[n+1], n=0~7 (n: 3 bits preload counter)

″X″ stands for don't care

On the sampling rate table, we can easily see that the sampling rate is dependent on the system clock. If start bit of the 30H.0 is set as ″1″, the PWM2 and PWM1 will output a GND level voltage.

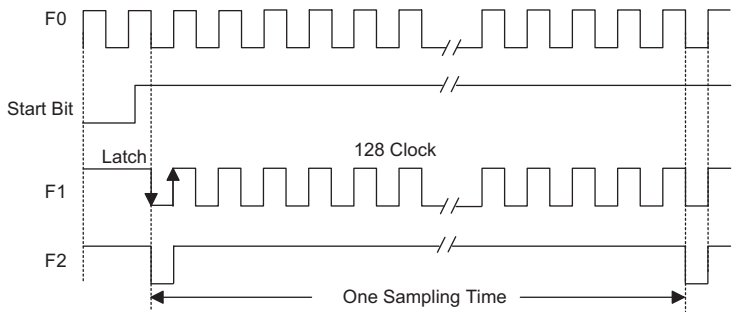
Label	Bits	Function
PWM Dis/En	0	Enable/disable PWM output 0: enable; 1: disable
BZ/SP	1	Output driver select 1: buzzer; 0: speaker
6/7 Bits	2	PWM counter bit select 1: 7 bits; 0: 6 bits
P0~P2	3~5	3 bits preload counter Bit5~3: 000B~111B (0~7); Bit3: LSB
D0, D1	6, 7	PWMI

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

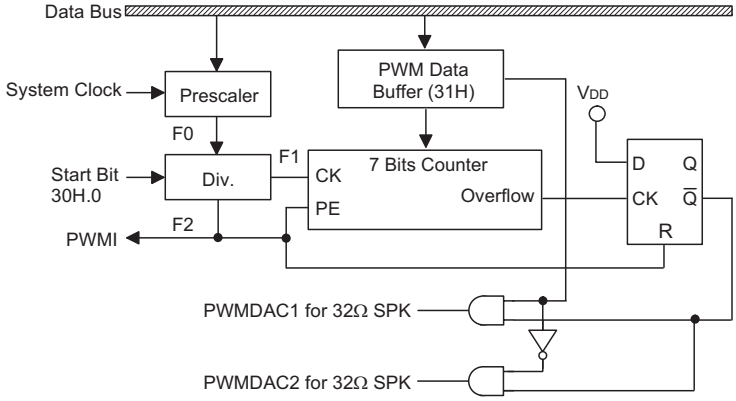
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit	D7	D6	D5	D4	D3	D2	D1	D0
6-bit	D7	D6	D5	D4	D3	D2	D1	X

Note: ″X″ stands for don't care

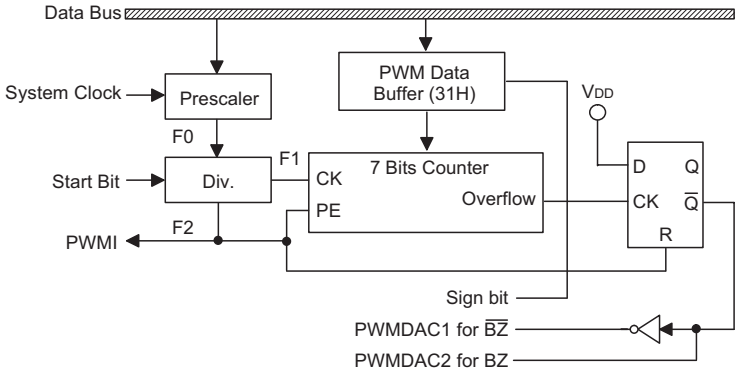
Bit7: Sign bit



PWM



30H.1=0 Speaker



30H.1=1 Buzzer

Serial Interface Protocol (SPI)

3 wire SPI format, support 32KBytes/64KBytes /128K.Bytes/256KBytes

Rising edge latch data, falling edge output data

Serial RAM Control Register

Register	Label	Bits	Function
SRC (32H)	—	0	Unused bit, read as ″0″
	Busy	1	0: The data register is full (readable) or empty (writeable) (Default=0) 1: Serial RAM interface is busy, cannot write/read the data register
	—	2	Unused bit, read as ″0″
	Sread	3	1: Series read; 0: step read
	W/R	4	W/R=0: read mode; To read data from the external serial RAM W/R=1: write mode; To write data to the external serial RAM
	SMODE	5	SPI mode setting 0: Mode 0; 1: Mode 3
	Sclk0	6	Serial RAM interface clock, Default=0
	Sclk1	7	Serial RAM interface clock, Default=0

Sclk1	Sclk0	Serial RAM interface clock selector
0	0	Serial RAM interface clock=system clock/2
0	1	Serial RAM interface clock=system clock/4
1	1	Serial RAM interface clock=system clock/8
1	1	Serial RAM interface clock=system clock/16

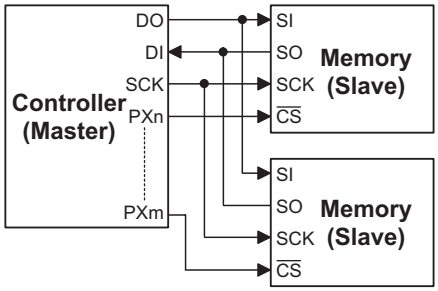
- Data are read from or written to the Data Register which is transmitted through the Serial RAM interface
- After the next 8-bit data are written to, it is transmitted to the 8 Serial RAM interface clock
- Within the 8 serial RAM interface clock, while transmission occurs the busy flag goes 1, after which, when the transmission is completed, the busy flag goes 0

Serial RAM Data Register

Address	Register	7	6	5	4	3	2	1	0	R/W	Default
33H	SRD	D7	D6	D5	D4	D3	D2	D1	D0	R/W	00000000 (1sb)

SPI Interface Information

- SPI interface connection



Note: Controller (master): DO/DI/SCK is SPI interface pin, PXn~PXm are generic I/O port.
Memory (slave): SI/SO/SCK is SPI interface pin, CS is chip select.

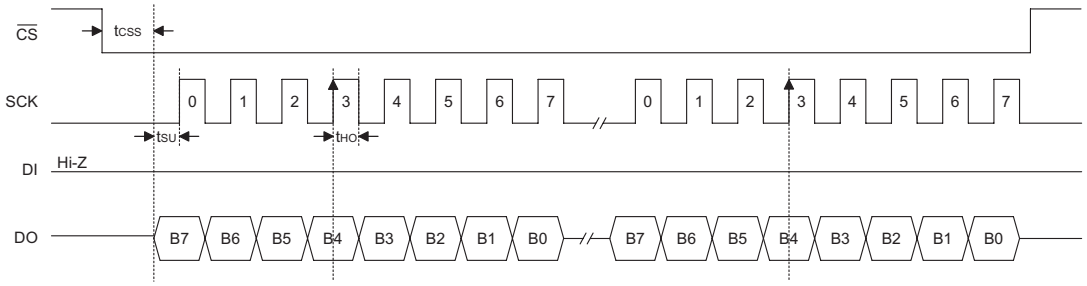
- SPI register description

The SPI is used two register, one is data register, the other one is control register. And for the data register is used for data transfer/receiver register, the control register is used to control and display the status of the SPI.

Control Register Bit	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
	SCK Setting	SCK Setting	SPI Mode Setting	Read/write Mode Setting	SPI Series Read	Reserved	SPI busy	Reserved
	B7,B6=0,0→SCK=f_{SYS}/2 B7,B6=0,1→SCK=f_{SYS}/4 B7,B6=1,0→SCK=f_{SYS}/8 B7,B6=1,1→SCK=f_{SYS}/16		B5=1 →Mode3 B5=0 →Mode0	B4=1 →Write B4=0 →Read	B3=1 →Series read B3=0 →Step read	RO="0"	B1=1 →Busy, data serial Out. B1=0 →Ready, can access data.	RO="0"

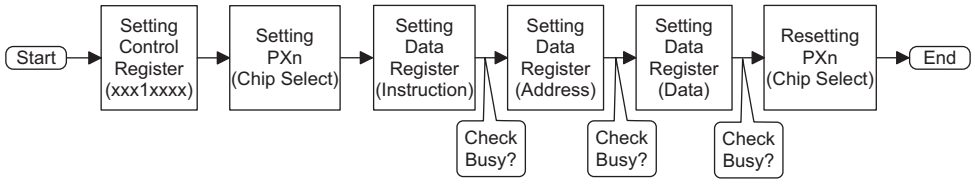
Note: The SPI mode0 & mode3 is changed by software option.

- Write data into memory (write mode) timing chart (Mode 0)



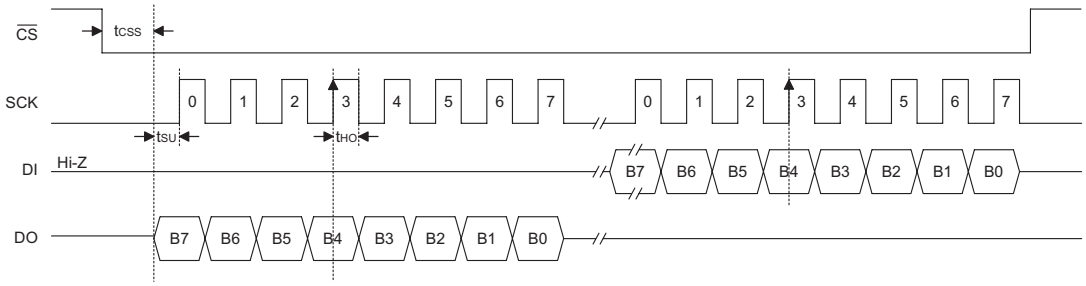
Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{CSS}	CS Setup Time	0		—	ns
t_{SU}	Data in Setup Time	—	$\frac{1}{2}$ SCK	—	—
t_{HO}	Data in HOLD Time	—	$\frac{1}{2}$ SCK	—	—

- The following will show how to write data to memory by the way of the flowchart



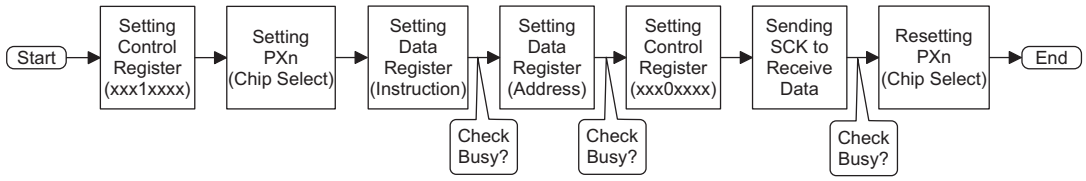
Note: After to write the data register, the serial out is executed automatically, and the busy bit of the SPI will be set to "1", when the serial out is completed, the busy bit of the SPI will be set to "0", and can be readable/writeable in this moment.

- Read data from memory (read mode) timing chart (Mode 0)



Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{CSS}	CS Setup Time	0		—	ns
t_{SU}	Data in Setup Time	—	$\frac{1}{2 \quad SCK}$	—	—
t_{HO}	Data in HOLD Time	—	$\frac{1}{2 \quad SCK}$	—	—

- The following will show how to read data from memory by the way of the flowchart



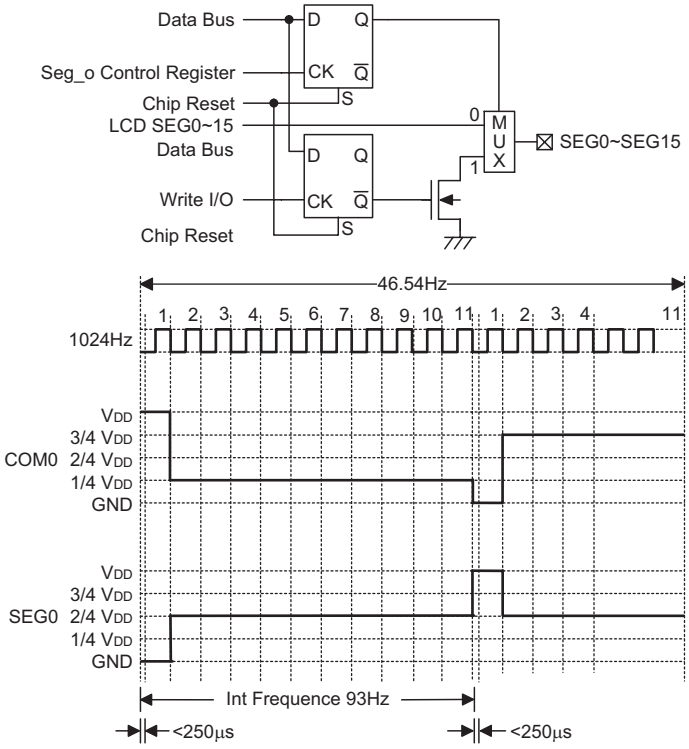
Low Voltage Detected

The Controller provides a circuit that detects the VLCD pin voltage level. To enable this detection function, the LVEN should be written as 1. Once this function is enabled, the detection circuit needs 100μs to be stable. After that, user could read the result from the LVFG. The low voltage detect function will consume power. For power saving, write 0 to LVEN if the low voltage detection function is unnecessary.

The tolerance value for the 3 Conditions (Min, Typ. Max) are within 5%.

Keyscan

LCD out port structure



Int,
Clear LCD output PC & PD
Turn on PA Pull-Hi Resister
Turn on LCD output PC
Input PA
Turn off LCD output PC
Turn off PA Pull-Hi Resister
Turn on PA Pull-Hi Resister
Turn on LCD output PD
Input PA
Turn off LCD output PD
Turn off PA Pull-Hi Resister

Return Int.

For every Frame, each have a Common signal, all of which can generate a single interrupt. The Scan Key performs the following:

- The keyscan loop with 32kHz system frequency
- The keyscan loop executes every 2 keyscan Int,.
- After an Interrupt occurs, clear LCD segment output port
- Turn on the PA Port's Pull-hi resistance
- Turn on the LCD segment output port.
- Use the LCD segment Output port and input port PA to implement the Key Scan
- Take the Scan value and store into the Memory
- Turn off the LCD segment output port transition to LCD segment output.
- Turn off the PA pull-hi
- Return to the main routine and change system frequency
- The keyscan function have to be completed in the period of interrupt time.
- The keyscan function is generated by common signal. When the LCD function is enable, the keyscan function can be used.

Example:

```
;*keyscan loop executes every 2 keyscan Int.

clr     lcdpc       ;clear LCD output (seg0~seg7)
clr     lcdpd       ;clear LCD output (seg8~seg15)
set     paphc       ;enable PA Pull-hi
clr     lcdpcc      ;seg0~seg7 output port
mov     a,pa        ;write PA to ACC
set     lcdpcc      ;turn on seg0~seg7
clr     paphc       ;disable PA Pull-hi
cpla    acc         ;complement ACC
sz      acc         ;check if stroke down
jmp     speed_up    ;then change from Normal mode to Green mode

set     paphc       ;enable PA Pull-hi
clr     lcdpdc      ;seg8~seg15 output port
mov     a,pa        ;write PA to ACC
set     lcdpdc      ;turn on seg8~seg15
clr     paphc       ;disable PA Pull-hi
cpla    acc         ;complement ACC
sz      acc         ;check if stroke down
jmp     speed_up    ;then change from Normal mode to Green mode
reti
```

LCD output control (38H)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1: seg7	1: seg6	1: seg5	1: seg4	1: seg3	1: seg2	1: seg1	1: seg0
0: output	0: output	0: output	0: output	0: output	0: output	0: output	0: output

LCD output (37H)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (Seg0)
D7	D6	D5	D4	D3	D2	D1	D0

LCD output control (3AH)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1: seg15	1: seg14	1: seg13	1: seg12	1: seg11	1: seg10	1: seg9	1: seg8
0: output	0: output	0: output	0: output	0: output	0: output	0: output	0: output

LCD output (39H)

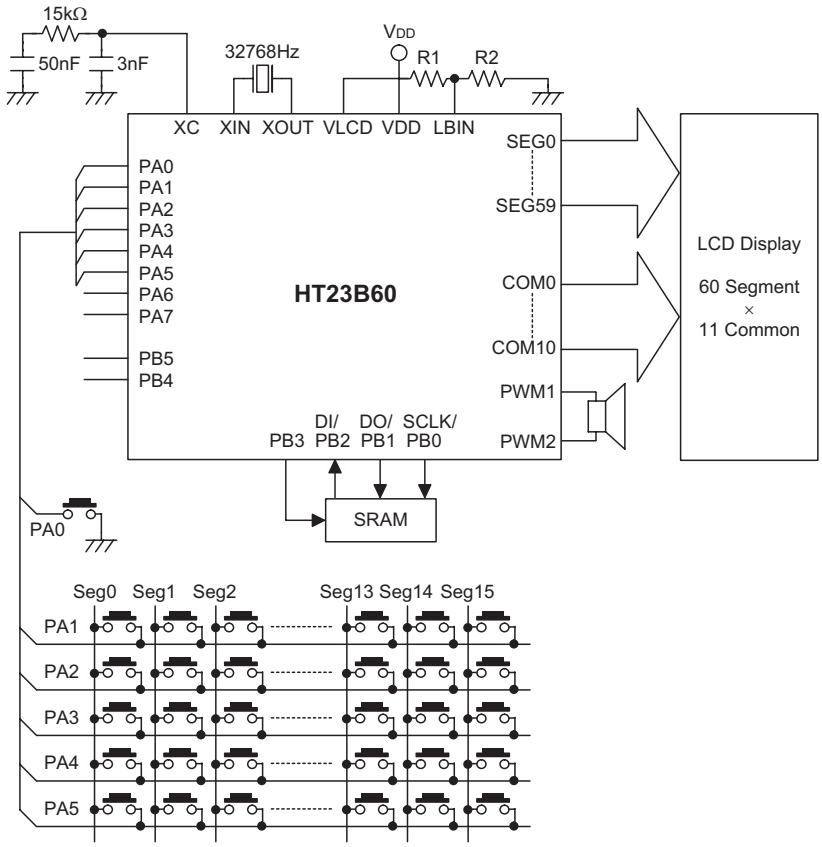
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0 (Seg8)
D7	D2	D5	D4	D3	D2	D1	D7

Mask Option

The following table shows many kinds of mask option in the Databank Controller. All these options should be defined in order to ensure prober system functions

Name	Mask Option
WDT	WDT source selection RC→Select the WDT OSC to be the WDT source T1→Select the instruction clock to be the WDT source 32kHz→Select the external 32768Hz to be the WDT source Disable→Disable the WDT function
WDTinstr	This option defines how to clear the WDT by instruction One clear instruction→The ″CLR WDT″ can clear the WDT Two clear instructions→Only when both of the ″CLR WDT1″ and ″CLR WDT2″ have been executed, then the WDT can be cleared
HALT option	HALT function selection Defines the HALT function either disabled or enabled
Wake-up PA	Port A wake-up selection. Defines the wake-up function activity All port A have the capability to wake-up the chip from HALT This wake-up function is selected per bit
LCD bias register selection	This option describes the LCD bias current. There are three types of selection *_Selectable as small, middle or large current Small current: 660K Middle current: 330K Large current: 66K
PB0~2 share pad option	Defines the pad ″PB0~PB2″ whether normal I/O pad or serial RAM interface pad
LCD duty option	Defines the LCD duty whether 1/10 or 1/11 duty
LCD bias option	Defines the LCD bias whether 1/4 or 1/5 bias
PB3 share pad option	Defines the pad ″PB3″ whether INT interrupt input pad or normal I/O pad

Application Circuits



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,PDF
CLR WDT1	Pre-clear Watchdog Timer	1	TO(4),PDF(4)
CLR WDT2	Pre-clear Watchdog Timer	1	TO(4),PDF(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,PDF

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 PDF are cleared.
Otherwise the TO and PDF 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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

CLR WDT Clear Watchdog Timer

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

Operation WDT ← 00H
PDF and TO ← 0

Affected flag(s)

TO	PDF	OV	Z	AC	C
0	0	—	—	—	—

CLR WDT1 Preclear Watchdog Timer

Description Together with CLR WDT2, clears the WDT. PDF 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 PDF flags remain unchanged.

Operation WDT ← 00H*
PDF and TO ← 0*

Affected flag(s)

TO	PDF	OV	Z	AC	C
0*	0*	—	—	—	—

CLR WDT2 Preclear Watchdog Timer

Description Together with CLR WDT1, clears the WDT. PDF 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 PDF flags remain unchanged.

Operation WDT ← 00H*
PDF and TO ← 0*

Affected flag(s)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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 (PDF) is set and the WDT time-out bit (TO) is cleared.

Operation PC ← PC+1
PDF ← 1
TO ← 0

Affected flag(s)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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 ← ROM code (high byte)

Affected flag(s)

TO	PDF	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)

TO	PDF	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)

TO	PDF	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.

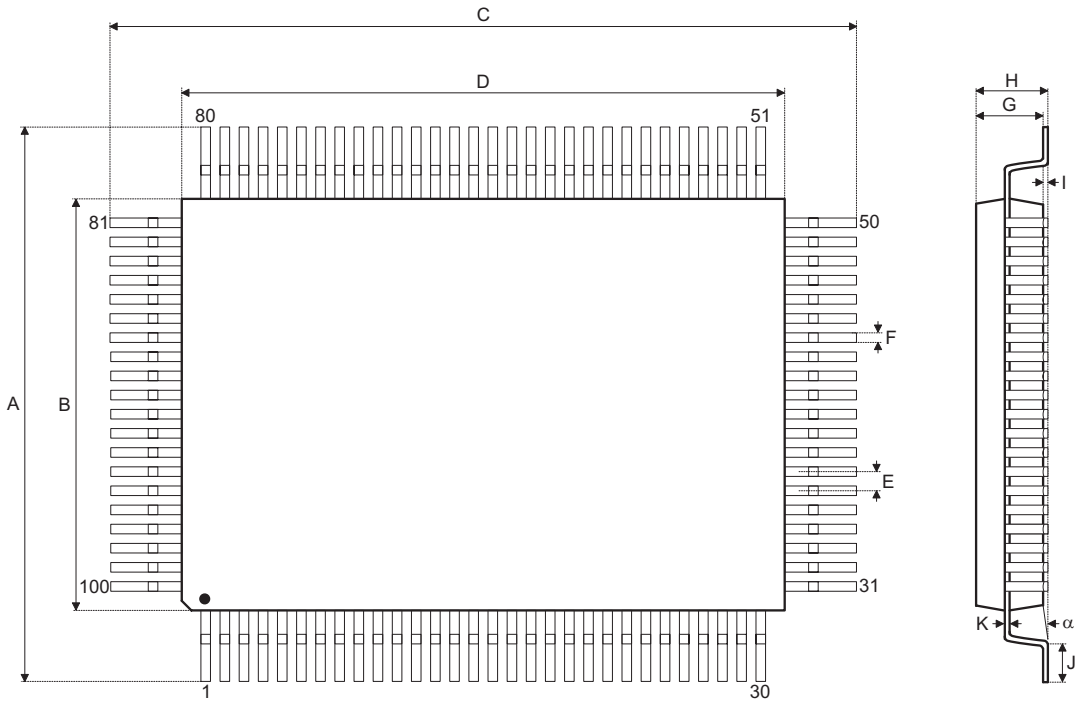
Operation ACC ← ACC ″XOR″ x

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

Package Information

100-pin QFP (14×20) Outline Dimensions



Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	18.50	—	19.20
B	13.90	—	14.10
C	24.50	—	25.20
D	19.90	—	20.10
E	—	0.65	—
F	—	0.30	—
G	2.50	—	3.10
H	—	—	3.40
I	—	0.10	—
J	1	—	1.40
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)
43F, SEG Plaza, Shen Nan Zhong Road, Shenzhen, China 518031
Tel: 0755-8346-5589
Fax: 0755-8346-5590
ISDN: 0755-8346-5591

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

Holmate Semiconductor, Inc. (North America Sales Office)
46712 Fremont Blvd., Fremont, CA 94538
Tel: 510-252-9880
Fax: 510-252-9885
http://www.holmate.com