Technical Document

- Tools Information
- FAQs
- Application Note

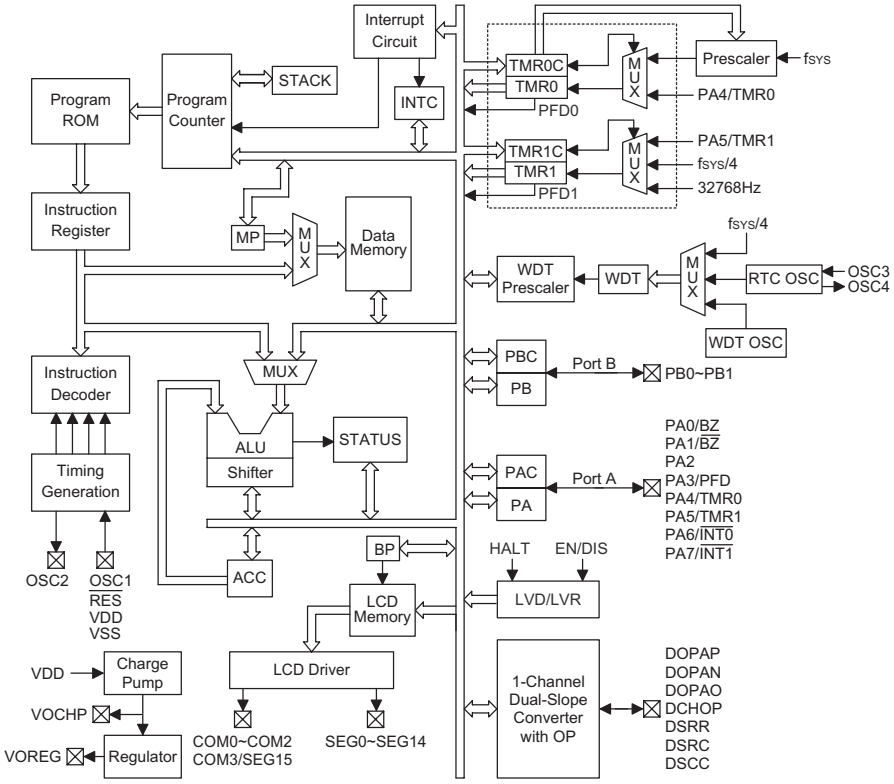
Features

- Operating voltage:
  f_{SYS} = 4MHz: 2.2V~5.5V
  f_{SYS} = 8MHz: 3.3V~5.5V
- 10 bidirectional I/O lines and two ADC inputs
- Two external interrupts inputs shared with I/O lines
- One 8-bit and one 18-bit programmable timer/event counter with overflow interrupt and pre-scaler
- LCD driver with 15×4, 16×3 or 16×2 segments
- 4K×15 program memory with partial lock function
- 96×8 data memory RAM
- Single differential input channel dual slope Analog to Digital Converter with Operational Amplifier.
- Watchdog Timer with regulator power
- Buzzer output
- External 32768Hz RTC oscillator
- Integrated RC or crystal oscillator
- Power-down and wake-up functions reduce power consumption
- Internal 3.3V Voltage regulator and charge pump
- Embedded voltage reference generator - 1.5V
- 6-level subroutine nesting
- Bit manipulation instruction
- 15-bit table read instruction
- Up to 0.5μs instruction cycle with 8MHz system clock at V_{DD}=5V
- 63 powerful instructions
- All instructions in 1 or 2 machine cycles
- Low voltage reset/detector function
- 56-pin SSOP package

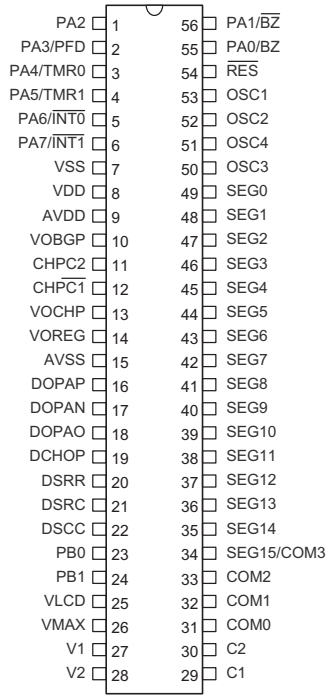
General Description

The HT46R74D-1 is an 8-bit high performance, RISC architecture microcontroller device specifically designed for A/D with LCD applications that interface directly to analog signals, such as those from sensors. The advantages of low power consumption, I/O flexibility, timer functions, oscillator options, Dual slope A/D converter, LCD display, HALT and wake-up functions, watchdog timer, as well as low cost, enhance the versatility of these devices to suit for a wide range of AD with LCD application possibilities such as sensor signal processing, scales, consumer products, subsystem controllers, etc.

Block Diagram



Pin Assignment



HT46R74D-1
— 56 SSOP-A

Pin Description

Pin Name	I/O	Options	Description
PA0/BZ PA1/$\overline{BZ}$ PA2 PA3/PFD PA4/TMR0 PA5/TMR1 PA6/$\overline{INT0}$ PA7/$\overline{INT1}$	I/O	Wake-up Pull-high Buzzer PFD	Bidirectional 8-bit input/output port. Each individual pin on this port can be configured as a wake-up input by a configuration option. Software instructions determine if the pin is a CMOS output or Schmitt trigger input. Configuration options determine which pins on this port have pull-high resistors. The BZ/$\overline{BZ}$, PFD, TMR0/TMR1, $\overline{INT0}$/$\overline{INT1}$ are pin-shared with PA0/1, PA3, PA4/5, and PA6/7, respectively.
PB0~PB1	I/O	Pull-high	Bidirectional 2-bit input/output port. Software instructions determine if the pin is a CMOS output or Schmitt trigger input. Configuration options determine which pins on this port have pull-high resistors.
VLCD	—	—	LCD power supply
VMAX	—	—	IC maximum voltage. Connect to VDD, VLCD or V1
V1, V2, C1, C2	—	—	LCD voltage pump
COM0~COM2 COM3/SEG15	O	1/2, 1/3 or 1/4 Duty	COM0~COM3 are the LCD common outputs. A configuration option selects the LCD duty-cycle. When either 1/3 or 1/2 duty is selected, the COM3/SEG15 pin will be configured as SEG15.
SEG0~SEG14	O	Segment Output	LCD driver outputs for the LCD panel segments.
VOBGP	AO	—	Band gap voltage output pin. (for internal use)
VOREG	O	—	Regulator output - 3.3V
VOCHP	O	—	Charge pump output - requires external capacitor
CHPC1	—	—	Charge pump capacitor, positive
CHPC2	—	—	Charge pump capacitor, negative
DOPAN, DOPAP, DOPAO, DCHOP	AI/AO	—	Dual Slope converter pre-stage OPA related pins. DOPAN is the OPA Negative input pin, DOPAP is the OPA Positive input pin, DOPAO is the OPA output pin and DCHOP is the OPA Chopper pins.
DSRR, DSRC, DSCC	AI/AO	—	Dual slope AD converter main function RC circuit. DSRR is the input or reference signal, DSRC is the Integrator negative input, and DSCC is the comparator negative input.
OSC1 OSC2	I O	Crystal or RC	OSC1, OSC2 are connected to an external RC network or crystal for the internal system clock. The OSC2 pin can be used to monitor the system clock at 1/4 frequency.
OSC3 OSC4	I O	RTC or System Clock	OSC3, OSC4 are connected to a 32768Hz crystal to form a real time clock for timing purposes or to form a system clock.
$\overline{RES}$	I	—	Schmitt trigger reset input, active low
VDD	—	—	Positive power supply
VSS	—	—	Negative power supply, ground
AVDD	—	—	Analog positive power supply
AVSS	—	—	Analog negative power supply, ground

Absolute Maximum Ratings

Supply Voltage .....V_{SS}–0.3V to V_{SS}+6.0V
Input Voltage.....V_{SS}–0.3V to V_{DD}+0.3V
I_{OL} Total 150mA
Total Power Dissipation 500mW

Storage Temperature .....–50°C to 125°C
Operating Temperature.....–40°C to 85°C
I_{OH} Total.....–100mA

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	—	f_{SYS}=4MHz	2.2	—	5.5	V
		—	f_{SYS}=8MHz	3.3	—	5.5	V
I_{DD1}	Operating Current (Crystal OSC)	3V	No load, f_{SYS}=4MHz Analog block off	—	0.6	1.6	mA
		5V		—	2	4	mA
I_{DD2}	Operating Current (RC OSC)	3V	No load, f_{SYS}=4MHz Analog block off	—	0.8	1.5	mA
		5V		—	2.5	4	mA
I_{DD3}	Operating Current (RC OSC)	5V	No load, f_{SYS}=8MHz Analog block off	—	4	8	mA
I_{DD4}	Operating Current (Crystal OSC)	5V	No load, f_{SYS}=8MHz Analog block off	—	4	8	mA
I_{DD5}	Operating Current (RTC OSC)	3V	No load, f_{SYS}=32768Hz	—	0.3	0.6	mA
		5V		—	0.6	1	mA
I_{DD6}	Operating Current (ADC On)	5V	V_{REGO}=3.3V, f_{SYS}=4MHz ADC on, ADCCLK=125kHz (all other analog devices off)	—	3	5	mA
I_{STB1}	Standby Current (*f_S=f_{SYS}/4)	3V	No load, system HALT, Analog block off, LCD off	—	—	1	μA
		5V		—	—	2	μA
I_{STB2}	Standby Current (*f_S=RTC OSC)	3V	No load, system HALT, Analog block off, LCD off	—	2.5	5	μA
		5V		—	10	20	μA
I_{STB3}	Standby Current (*f_S=WDT OSC)	3V	No load, system HALT, Analog block off, LCD off	—	2	5	μA
		5V		—	6	10	μA
I_{STB4}	Standby Current (*f_S=RTC OSC)	3V	No load, system HALT, Analog block off, LCD on 1/2 bias, V_{LCD}=V_{DD} (Low bias current option)	—	17	30	μA
		5V		—	34	60	μA
I_{STB5}	Standby Current (*f_S=RTC OSC)	3V	No load, system HALT, Analog block off, LCD on 1/3 bias, V_{LCD}=V_{DD} (Low bias current option)	—	13	25	μA
		5V		—	28	50	μA
I_{STB6}	Standby Current (*f_S=WDT OSC)	3V	No load, system HALT, Analog block off, LCD on 1/2 bias, V_{LCD}=V_{DD}	—	14	25	μA
		5V		—	26	50	μA
I_{STB7}	Standby Current (*f_S=WDT OSC)	3V	No load, system HALT, Analog block off, LCD on 1/3 bias, V_{LCD}=V_{DD}	—	10	20	μA
		5V		—	19	40	μA

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{IL1}	Input Low Voltage for I/O Ports, TMR and $\overline{INT}$	—	—	0	—	$0.3V_{DD}$	V
V_{IH1}	Input High Voltage for I/O Ports, TMR and $\overline{INT}$	—	—	$0.7V_{DD}$	—	V_{DD}	V
V_{IL2}	Input Low Voltage ($\overline{RES}$)	—	—	0	—	$0.4V_{DD}$	V
V_{IH2}	Input High Voltage ($\overline{RES}$)	—	—	$0.9V_{DD}$	—	V_{DD}	V
V_{LCD}	LCD Highest Voltage	—	—	0	—	V_{DD}	V
V_{LVR1}	Low Voltage Reset 1	—	LVR option= 2.2V	2.1	2.2	2.3	V
V_{LVR2}	Low Voltage Reset 2	—	LVR option= 3.3V	3.15	3.3	3.45	V
V_{LVD1}	Low Voltage Detector 1	—	LVR option= 2.2V, LVD option= LVR+0.2	2.25	2.4	2.55	V
V_{LVD2}	Low Voltage Detector 2	—	LVR option=3.3V LVD option= LVR+0.2	3.45	3.6	3.75	V
I_{OL1}	I/O Port Segment Logic Output Sink Current	3V	$V_{OL}=0.1V_{DD}$	4	8	—	mA
		5V		10	20	—	mA
I_{OH1}	I/O Port Segment Logic Output Source Current	3V	$V_{OH}=0.9V_{DD}$	–2	–4	—	mA
		5V		–5	–10	—	mA
I_{OL2}	LCD Common and Segment Current	3V	$V_{OL}=0.1V_{DD}$	210	420	—	μA
		5V		350	700	—	μA
I_{OH2}	LCD Common and Segment Current	3V	$V_{OH}=0.9V_{DD}$	–80	–160	—	μA
		5V		–180	–360	—	μA
R_{PH}	Pull-high Resistance of I/O Ports and $\overline{INT}$	3V	—	20	60	100	kΩ
		5V	—	10	30	50	kΩ
Charge Pump and Regulator							
V_{CHPI}	Input Voltage	—	Charge pump on	2.2	—	3.6	V
			Charge pump off	3.7	—	5.5	V
V_{REGO}	Output Voltage	—	No load	3	3.3	3.6	V
V_{REGDP1}	Regulator Output Voltage Drop (Compare with No Load)	—	V_{DD}=3.7V~5.5V Charge pump off Current≤10mA	—	100	—	mV
V_{REGDP2}		—	V_{DD}=2.4V~3.6V Charge pump on Current≤6mA	—	100	—	mV
Dual Slope AD, Amplifier and Band Gap							
V_{RFGO}	Reference Generator Output	—	@3.3V	1.45	1.5	1.55	V
V_{RFGTC}	Reference Generator Temperature Coefficient	—	@3.3V	—	50	—	Ppm/C
V_{ADOFF}	Input Offset Range	—	—	—	500	800	μV
V_{ICMR}	Common Mode Input Range	—	Amplifier, no load	0.2	—	V_{REGO}–1	V
		—	Integrator, no load	1	—	V_{REGO}–0.2	V

A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SYS}	System Clock (RC OSC)	—	2.2V~5.5V	400	—	4000	kHz
	System Clock (Crystal OSC)	—	2.2V~5.5V	400	—	4000	kHz
		—	3.3V~5.5V	400	—	8000	kHz
f_{INRC}	Internal RC OSC	3V	—	—	12	—	kHz
		5V		—	15	—	kHz
f_{TIMER}	Timer I/P Frequency (TMR0/TMR1)	—	2.2V~5.5V	0	—	4000	kHz
t_{WDTOSC}	Watchdog Oscillator Period	3V	—	45	90	180	μs
		5V	—	32	65	130	μs
t_{RES}	External Reset Low Pulse Width	—	—	1	—	—	μs
t_{SST}	System Start-up Timer Period	—	Power-up or wake-up from HALT	—	1024	—	t_{SYS}
t_{LVR}	Low Voltage Width to Reset	—	—	0.25	1	2	ms
t_{INT}	Interrupt Pulse Width	—	—	1	—	—	μs

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

Functional Description

Execution Flow

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

Instruction fetching and execution are pipelined in such a way that a fetch takes one instruction cycle while decoding and execution takes the next instruction cycle. The pipelining scheme makes it possible for each instruction to be effectively executed in a cycle. If an instruction changes the value of the program counter, two cycles are required to complete the instruction.

Program Counter – PC

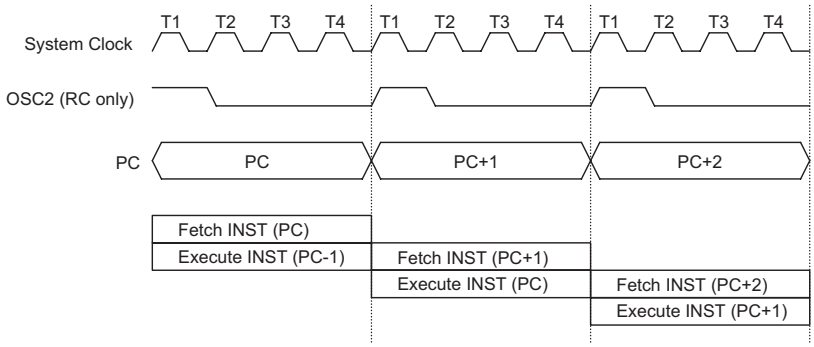
The program counter is 12 bits wide and controls the sequence in which the instructions stored in the program ROM are executed. The contents of the PC can specify a maximum of 4096 addresses.

After accessing a program memory word to fetch an instruction code, the value of the PC is incremented by 1. The PC then points to the memory word containing the next instruction code.

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

The conditional skip is activated by instructions. Once the condition is met, the next instruction, fetched during the current instruction execution, is discarded and a dummy cycle replaces it to get a proper instruction, otherwise the program proceeds with the next instruction.

The lower byte of the Program Counter, PCL, is a readable and writeable register. Moving data into the PCL register performs a short jump. The destination must be within 256 locations.



Execution Flow

Mode	Program Counter											
	*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
External Interrupt 0	0	0	0	0	0	0	0	0	0	1	0	0
External Interrupt 1	0	0	0	0	0	0	0	0	1	0	0	0
Timer/Event Counter 0 Overflow	0	0	0	0	0	0	0	0	1	1	0	0
Timer/Event Counter 1 Overflow	0	0	0	0	0	0	0	1	0	0	0	0
ADC Interrupt	0	0	0	0	0	0	0	1	0	1	0	0
RTC Interrupt	0	0	0	0	0	0	0	1	1	0	0	0
Skip	Program Counter+2											
Loading PCL	*11	*10	*9	*8	@7	@6	@5	@4	@3	@2	@1	@0
Jump, Call Branch	#11	#10	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
Return from Subroutine	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

Program Counter

Note: *11~*0: Program counter bits
#11~#0: Instruction code bits
S11~S0: Stack register bits
@7~@0: PCL bits

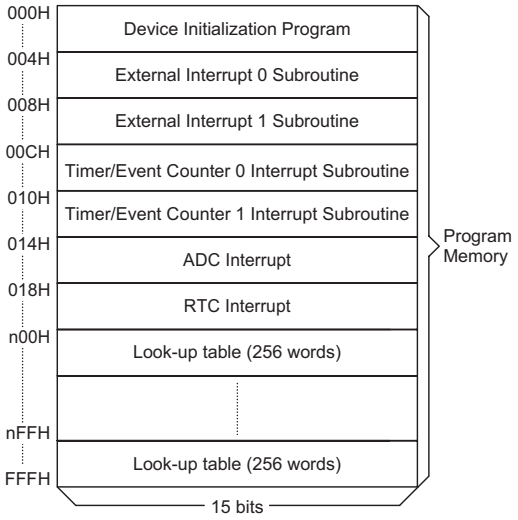
When a control transfer takes place, an additional dummy cycle is required.

Program Memory – ROM

The program memory is used to store the program instructions which are to be executed. It also contains data, table, and interrupt entries, and is organized with a structure of 4096×15 bits which are addressed by the program counter and table pointer.

Certain locations in the ROM are reserved for special usage:

- Location 000H
  Location 000H is reserved for program initialization. After a chip reset, the program always begins execution at this location.
- Location 004H
  Location 004H is reserved for the $\overline{INT0}$ external interrupt service program. If the $\overline{INT0}$ input pin is activated, and the interrupt is enabled, and the stack is not full, the program begins execution at location 004H.
- Location 008H
  Location 008H is reserved for the $\overline{INT1}$ external interrupt service program. If the $\overline{INT1}$ input pin is activated, and the interrupt is enabled, and the stack is not full, the program begins execution at location 008H.



Note: n ranges from 0 to 1F

Program Memory

- Location 00CH
  Location 00CH is reserved for the Timer/Event Counter 0 interrupt service program. If a timer interrupt results from a Timer/Event Counter 0 overflow, and if the interrupt is enabled and the stack is not full, the program begins execution at location 00CH.
- Location 010H
  Location 010H is reserved for the Timer/Event Counter 1 interrupt service program. If a timer interrupt results from a Timer/Event Counter 1 overflow, and if the interrupt is enabled and the stack is not full, the program begins execution at location 010H.
- Location 014H
  Location 014H is reserved for the ADC interrupt service program. If an ADC interrupt occurs, and the interrupt is enabled, and the stack is not full, the program begins execution at location 014H.
- Location 018H
  Location 018H is reserved for the real time clock interrupt service program. If a real time clock interrupt occurs, and the interrupt is enabled, and the stack is not full, the program begins execution at location 018H.
- Table location
  Any location in the Program Memory can be used as a look-up table. The instructions "TABRDC [m]" (the current page, 1 page=256 words) and "TABRDL [m]" (the last page) transfer the contents of the lower-order byte to the specified data memory, and the contents of the higher-order byte to the TBLH register, which is the Table high order byte register. Only the destination of the lower-order byte in the table is well-defined; the other bits of the table word are all transferred to the lower portion of TBLH. The TBLH register is read only, and the table pointer, TBLP, is a read/write register, and is used to indicate the table location. Before accessing the table, the location should be placed into the TBLP register. All the table related instructions require 2 cycles to complete their operation. These areas may function as normal ROM depending upon the user's requirements.

Stack Register – STACK

The stack register is a special part of the memory used to save the contents of the program counter. The stack is organized into 6 levels and is neither part of the data nor part of the program, and is neither readable nor writeable. Its activated level is indexed by a stack pointer, SP, and is neither readable nor writeable. At the

Instruction(s)	Table Location											
	*11	*10	*9	*8	*7	*6	*5	*4	*3	*2	*1	*0
TABRDC [m]	P11	P10	P9	P8	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL [m]	1	1	1	1	@7	@6	@5	@4	@3	@2	@1	@0

Table Location

Note: *11~*0: Table location bits
@7~@0: Table pointer bits
P11~P8: Current program counter bits

start of a subroutine call or an interrupt acknowledgment, the contents of the program counter is pushed onto the stack. At the end of the subroutine or interrupt routine, indicated by a return instruction, RET or RETI, the contents of 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 is recorded but the acknowledgment is still inhibited. Once the SP is decremented, using a RET or RETI instruction, the interrupt is serviced. This feature prevents a stack overflow, allowing the programmer to use the structure easily. Likewise, if the stack is full, and a ″CALL″ is subsequently executed, a stack overflow occurs and the first entry is lost as only the most recent 6 return addresses are stored.

Data Memory – RAM

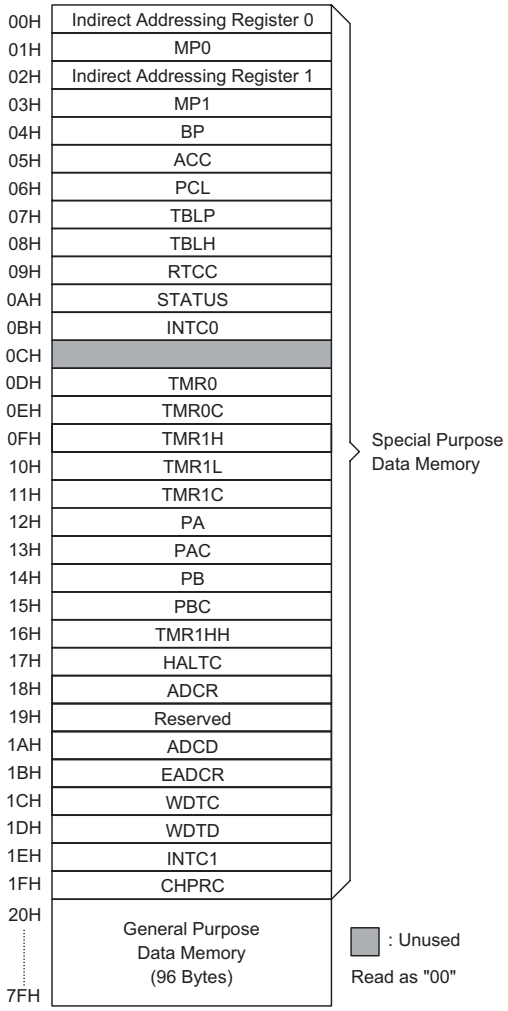
Bank 0 of the data memory has a capacity of 123×8 bits, and is divided into two functional groups, namely the special function registers, which have a 27×8 bit capacity and the general purpose data memory which have a 96×8 bit capacity. Most locations are readable/writable, although some are read only. The special function register are overlapped in all banks.

Any unused locations before 20H will return a zero result if read. The general purpose data memory, addressed from 20H to 7FH , is used for data and control information under instruction commands. All of the 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. They are also indirectly accessible through the memory pointer registers, MP0 and MP1.

Bank 1 contains the LCD Data Memory locations. After first setting up the Bank Pointer, BP, to the value of ″01H″ to access Bank 1, this bank must then be accessed indirectly using Memory Pointer MP1. With BP set to a value of ″01H″, using MP1 to indirectly read or write to the data memory areas with addresses from 40H~4FH will result in operations to Bank 1. Directly addressing the Data Memory will always result in Bank 0 being accessed irrespective of the value of BP.

Indirect Addressing Register

Locations 00H and 02H are indirect addressing registers that are not physically implemented. Any read/write operations on [00H] and [02H] accesses the RAM locations pointed to by MP0 and MP1 respectively. Reading locations 00H or 02H indirectly returns the result 00H. Writing to them indirectly leads to no operation. The function of data movement between two indirect addressing registers is not supported. The memory pointer



RAM Mapping

registers, MP0 and MP1, are both 7-bit registers and are used to access the Data Memory in combination with the indirect addressing registers. MP0 can only be used with the data memory, while MP1 can be used with both the data memory and the LCD display memory.

Accumulator – ACC

The accumulator, ACC, is related to the ALU operations. It is mapped to location 05H of the RAM and is capable of operating with immediate data. The data movement between two data memory locations must pass through the ACC.

Arithmetic and Logic Unit – ALU

This circuit performs 8-bit arithmetic and logic operations and 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 etc.)

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

Status Register – STATUS

The status register is 8 bits wide and contains, a carry flag (C), an auxiliary carry flag (AC), a zero flag (Z), an overflow flag (OV), a power down flag (PDF), and a watchdog time-out flag (TO). It also records the status information and controls the operation sequence.

Except for the TO and PDF flags, the status register bits can be altered by instructions similar to other registers. Data written into the status register does not alter the TO or PDF flags. Operations related to the status register, however, may yield different results from those intended. The TO and PDF flags can only be changed by a Watchdog Timer overflow, a device power-up, or clearing the Watchdog Timer and executing the ″HALT″ instruction. The Z, OV, AC, and C flags reflect the status of the latest operations.

On entering the interrupt sequence or executing a subroutine call, the status register will not be automatically pushed onto the stack. If the contents of the status register is important, and if the subroutine is likely to corrupt the status register, the programmer should take precautions and save it properly.

Interrupts

The device provides two external interrupts, two internal timer/event counter interrupts and the ADC interrupt. The interrupt control register INTC0, and interrupt control register INTC1, both contain the interrupt control bits that are used to set the enable/ disable status and interrupt request flags.

Once an interrupt subroutine is serviced, other interrupts are all blocked, by clearing the EMI bit. This prevents further interrupt nesting. Other interrupt requests may take place during this interval, but only the interrupt request flag will be recorded. If a certain interrupt requires servicing within the service routine, the EMI bit and the corresponding bit of INTC0 or INTC1 may be set in order to permit interrupt nesting to take place. Once the stack is full, the interrupt request will not be acknowledged, even if the related interrupt is enabled, until the Stack Pointer is decremented. If immediate service is desired, the stack should be prevented from becoming full.

All interrupts will provide a wake-up function. As an interrupt is serviced, a control transfer occurs by pushing the contents of the program counter onto the stack followed by a branch to a subroutine at the specified location in the Program Memory. Only the contents of the program counter is pushed onto the stack. If the contents of the accumulator or of the status register is altered by the interrupt service program which corrupts the desired control sequence, the contents should be saved in advance.

External interrupts are triggered by an edge transition on pin $\overline{\text{INT0}}$ or $\overline{\text{INT1}}$. A configuration option determines the type of edge transition, high to low, low to high, or both low to high and high to low. Their related interrupt request flags are EIF0; bit 4 of INTC0, and EIF1; bit 5 of INTC0, must also be set. After the interrupt is enabled, if the stack is not full and the external interrupt is active, a subroutine call to location 04H or 08H occurs. The interrupt request flag, EIF0 or EIF1, and EMI bits will be cleared to disable other maskable interrupts.

The internal Timer/Event Counter 0 interrupt is generated when the Timer/Event Counter 0 interrupt request flag is set, which is bit T0F; bit 6 of INTC0. This occurs when the timer overflows. After the interrupt is enabled, if the stack is not full, and the T0F bit is set, a subroutine call to location 0CH occurs. The related interrupt request flag, T0F, will be reset, and the EMI bit will be cleared to disable other maskable interrupts. The interrupt for Timer/Event Counter 1 operates in a similar

Bit No.	Label	Function
0	C	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. C is also affected by a rotate through carry instruction.
1	AC	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.
2	Z	Z is set if the result of an arithmetic or logic operation is zero; otherwise Z is cleared.
3	OV	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.
4	PDF	PDF is cleared by either a system power-up or executing the ″CLR WDT″ instruction. PDF is set by executing the ″HALT″ instruction.
5	TO	TO is cleared by a system power-up or executing the ″CLR WDT″ or ″HALT″ instruction. TO is set by a WDT time-out.
6~7	—	Unused bit, read as ″0″

Status (0AH) Register

manner but its related interrupt request flag is T1F, which is bit 4 of INTC1, and its subroutine call location is 10H.

The A/D converter interrupt is generated when the A/D converter interrupt request flag, ADF; bit 5 of INTC1 is set. This occurs when an A/D conversion process has completed. After the interrupt is enabled, if the stack is not full, and the ADF bit is set, a subroutine call to location 14H occurs. The related interrupt request flag, ADF, is reset and the EMI bit is cleared to disable further maskable interrupts.

The real time clock interrupt is generated when the real time clock interrupt request flag, RTF; bit 6 of INTC1, is set. After the interrupt is enabled, if the stack is not full, and the RTF bit is set, a subroutine call to location 18H occurs. The related interrupt request flag, RTF, is reset and the EMI bit is cleared to disable further maskable interrupts.

During the execution of an interrupt subroutine, other maskable interrupt acknowledgments are all held until the ″RETI″ instruction is executed or the EMI bit and the related interrupt control bit are set both to 1 (if the stack is not full). To return from the interrupt subroutine, a ″RET″ or ″RETI″ instruction may be invoked. A RETI instruction sets the EMI bit and enables an interrupt service, but a RET instruction does not.

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

Interrupt Source	Priority	Vector
External interrupt 0	1	04H
External interrupt 1	2	08H
Timer/Event Counter 0 overflow	3	0CH
Timer/Event Counter 1 overflow	4	10H
ADC interrupt	5	14H
Real time clock interrupt	6	18H

Once an interrupt request flag has been set, it remains in the INTC1 or INTC0 register until the interrupt is serviced or cleared by a software instruction.

It is recommended that a program should not use the ″CALL subroutine″ within the interrupt subroutine. This is because interrupts often occur in an unpredictable manner or require to be serviced immediately in some applications. During that period, if only one stack is left, and enabling the interrupts is not well controlled, executing a ″call″ in the interrupt subroutine may damage the original control sequence.

Bit No.	Label	Function
0	EMI	Control the master (global) interrupt (1=enabled; 0=disabled)
1	EEI0	Control the external interrupt 0 (1=enabled; 0=disabled)
2	EEI1	Control the external interrupt 1 (1=enabled; 0=disabled)
3	ET0I	Control the Timer/Event Counter 0 interrupt (1=enabled; 0=disabled)
4	EIF0	External interrupt 0 request flag (1=active; 0=inactive)
5	EIF1	External interrupt 1 request flag (1=active; 0=inactive)
6	T0F	Internal Timer/Event Counter 0 request flag (1=active; 0=inactive)
7	—	For test mode used only. Must be written as ″0″; otherwise may result in unpredictable operation.

INTC0 (0BH) Register

Bit No.	Label	Function
0	ET1I	Control the Timer/Event Counter 1 interrupt (1=enabled; 0=disabled)
1	EADI	Control the ADC interrupt (1=enabled; 0:disabled)
2	ERTI	Control the real time clock interrupt (1=enabled; 0:disabled)
3, 7	—	Unused bit, read as ″0″
4	T1F	Internal Timer/Event Counter 1 request flag (1=active; 0=inactive)
5	ADF	ADC request flag (1=active; 0=inactive)
6	RTF	Real time clock request flag (1=active; 0=inactive)

INTC1 (1EH) Register

Oscillator Configuration

The device provides three oscillator circuits for system clocks, i.e., RC oscillator, crystal oscillator and a 32768Hz crystal oscillator, determined by configuration options. The Power-down mode stops the system oscillator, (RC and crystal oscillator only) and ignores external signals in order to conserve power. The 32768Hz crystal oscillator will continue running even when in the Power-down mode. If the 32768Hz crystal oscillator is selected as the system oscillator, the system oscillator is not stopped; but instruction execution is stopped. Since the 32768Hz oscillator is also designed for timing purposes, the internal timing (RTC, time base, WDT) operation keeps running even if the system enters the Power-down mode.

Of the three oscillators, if the RC oscillator is used, an external resistor between OSC1 and VSS is required, whose resistance should range from 30kΩ to 750kΩ. The system clock, divided by 4, is available on OSC2 with a pull-high resistor added, which can be used to synchronise external logic. The RC oscillator provides the most cost effective solution, however, the frequency of the oscillation may vary with VDD, temperature, and the device itself due to process variations. It is therefore, not suitable for timing sensitive operations where accurate oscillator frequency is desired.

If the crystal oscillator is selected, a crystal across OSC1 and OSC2 is needed to provide the feedback and phase shift required for the oscillator. No other external components are required. A resonator may be connected between OSC1 and OSC2 to replace the crystal and to obtain a frequency reference, but two external capacitors connected between OSC1, OSC2 and ground are required.

Another oscillator circuit is supplied for the real time clock. For this oscillator only a 32.768kHz crystal oscillator can be used, and should be connected between pins OSC3 and OSC4.

The RTC oscillator circuit can be forced to start up quickly by setting the "QOSC" bit, which is bit 4 in the RTCC register. It is recommended to turn on the quick oscillating function when power is first applied, and then turn it off after 2 seconds.

The WDT oscillator is a free running on-chip RC oscillator, and no external components are required. Although when the system enters the power down mode, the system clock stops, the WDT oscillator keeps running with a period of approximately 65μs at 5V. The WDT oscillator can be disabled by a configuration option to conserve power.

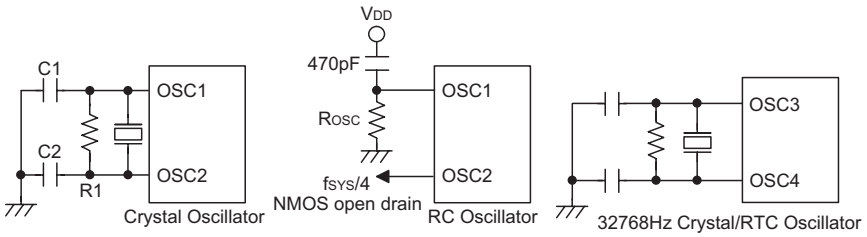
Watchdog Timer – WDT

The WDT clock is sourced from either its dedicated internal RC oscillator or from the instruction clock which is the system clock/4. The WDT is provided to prevent software malfunctions or a sequence from jumping to an unknown location with unpredictable results. The watchdog timer can be disabled by a configuration option. If the watchdog timer is disabled, the WDT timer will have the same operation as if it were enabled except that the timeout signal will not generate a device reset. So when the watchdog timer is disabled, the WDT timer counter can still be read out and can still be cleared. This function is used to permit the application program to access the WDT frequency to obtain the temperature coefficient for analog component adjustment. The WDT oscillator needs to be disabled/enabled using its registers, (WDTC :WDTOSC), to minimise power consumption.

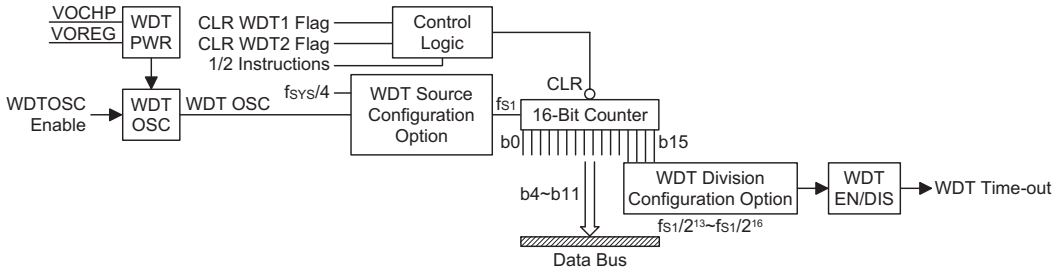
There are 2 registers related to the WDT function, WDTC and WDTD. The WDTC register controls the WDT oscillator enable/disable function and the WDT power source. The WDTD register is the WDT counter readout register.

The WDTPWR bits can be used to choose the WDT power source, the default source is VOCHP. The main purpose of the regulator is to be used for the WDT Temperature-coefficient adjustment. In this case, the application program should enable the regulator before switching to the regulator source. The WDTOSC bits can be used to enable or disable the WDT OSC (12kHz). If the application does not use the WDT OSC, then it needs to disable it in order to reduce power consumption.

If the internal RC oscillator, which has a nominal period of 65μs, is selected, it is first divided by a value which ranges from 2^{12}~2^{15} the exact value of which is determined by a configuration option, to obtain the actual WDT time-out period. The minimum period of the WDT time-out period is about 300ms~600ms. This time-out period may vary with temperature, VDD and process



System Oscillator



Watchdog Timer

variations. By using the related WDT configuration option, longer time-out periods can be .implemented. If the WDT time-out is selected to be 2^{15}, the maximum time-out period is divided by 2^{15}~2^{16}which will give a time-out period of about 2.3s~4.7s.

The WDT clock source may also come from the instruction clock, in which case the WDT will operate in the same manner except that in the Power Down mode the WDT will stop counting and lose its protecting purpose. In this situation the device can only be restarted by external logic. If the device operates in a noisy environment, using the on-chip RC oscillator is strongly recommended, since the HALT instruction will stop the system clock.

Under normal operation, a WDT overflow initialises a device reset and sets the status bit ″TO″. In the HALT or IDLE mode, the overflow initialises a ″warm reset″, and only the PC and SP are reset to zero. There are three methods to clear the contents of the WDT, an external low level on $\overline{RES}$, a software instruction or a ″HALT″ instruction. There are two types of software instructions; the single ″CLR WDT″ instruction, or the pair of instructions — ″CLR WDT1″ and ″CLR WDT2″.

Of these two types of instruction, only one type of instruction can be active at a time depending on the configuration option — ″CLR WDT″ times selection option. If the ″CLR WDT″ is selected (i.e., CLR WDT times equal one), any execution of the ″CLR WDT″ instruction clears the WDT. If the ″CLR WDT1″ and ″CLR WDT2″ option is chosen (i.e., CLR WDT times equal two), these two instructions have to be executed to clear the WDT, otherwise the WDT may reset the device due to a time-out.

Bit No.	Label	Function
0~1	WDTPWR0~ WDTPWR1	WDT Power source selection. 01: WDT power comes from VOCHP 10: WDT power comes from the regulator 00/11: WDT power comes from VOCHP strongly recommend use to use 01 for VOCHP prevent the noise to let the WDT lose the power
2~3	WDTOSC0~ WDTOSC1	The WDT oscillator enable/disable (WDTOSC1:0)= 01: WDT OSC disable10: WDT OSC enable 00/11: WDT OSC enable strongly recommend use to use 10 for WDT OSC enable
4~7	—	Reserved

WDTC (1CH) Register

Note: WDTOSC registers initial value will be set to enable (1,0), if both ″WDT option enable″ and ″WDT clock option set to WDT″, otherwise, it will be set to disable (0,1)

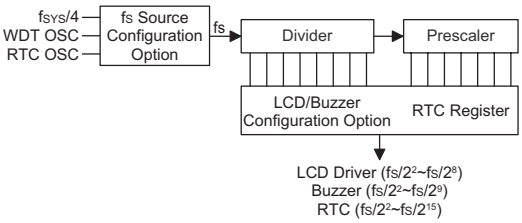
Bit No.	Label	Function
0~7	WDTD0~ WDTD7	The WDT counter data value. This register is read only. It′s used for temperature adjusting.

WDTD (1DH) Register

The WDT clock (f_{S1}) is further divided by an internal counter to give longer watchdog time-outs., In this device, the division ratio can be varied by selecting different configuration options to give 2^{13} to 2^{16} division ration range.

Multi-function Timer

The device provides a multi-function timer for the RTC, LCD and buzzer functions but with different time-out periods. The multi-function timer consists of an 8-stage divider and a 7-bit prescaler, with the clock source coming from the WDT OSC, the RTC OSC or the instruction clock which is the system clock divided by 4. The multi-function timer also provides a selectable frequency signal, which ranges from $f_S/2^2$ to $f_S/2^8$, for the LCD driver circuits, and a selectable frequency signal, ranging from $f_S/2^2$ to $f_S/2^9$, for the buzzer output using configuration options. It is recommended to select a frequency as close as possible to 4kHz signal for the LCD driver circuits to ensure a proper display.



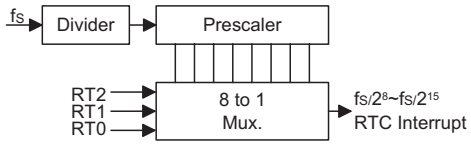
For the Charge Pump and the ADC chopper , the clock is independent of the multi-function timer. The clock always sourced from the system clock (RC or Crystal).

Real Time Clock – RTC

The real time clock, RTC, is operated in the same manner as the time base in that it is used to supply a regular internal interrupt. Its time-out period ranges from $f_S/2^8$ to $f_S/2^{15}$ by software programming . Writing data to RT2, RT1 and RT0, which are bits 2, 1, 0 of the RTCC register, provides various time-out periods. If an RTC time-out occurs, the related interrupt request flag, RTF; bit 6 of INTC1, is set. But if the interrupt is enabled, and if the stack is not full, a subroutine call to location 18H occurs.

RT2	RT1	RT0	RTC Clock Divided Factor
0	0	0	2^8*
0	0	1	2^9*
0	1	0	2^{10}*
0	1	1	2^{11}*
1	0	0	2^{12}
1	0	1	2^{13}
1	1	0	2^{14}
1	1	1	2^{15}

Note: * not recommended for use



Real Time Clock

Buzzer Output

The Buzzer function provides a means of producing a variable frequency output, suitable for applications such as Piezo-buzzer driving or other external circuits that require a precise frequency generator. The BZ and $\overline{BZ}$ pins form a complimentary pair, and are pin-shared with I/O pins, PA0 and PA1. A configuration option is used to select from one of three buzzer options. The first option is for both pins PA0 and PA1 to be used as normal I/Os, the second option is for both pins to be configured as BZ and $\overline{BZ}$ buzzer pins, the third option selects only the PA0 pin to be used as a BZ buzzer pin with the PA1 pin retaining its normal I/O pin function. Note that the $\overline{BZ}$ pin is the inverse of the BZ pin which together generate a differential output which can supply more power to connected interfaces such as buzzers.

The buzzer is driven by the internal clock source, f_S, which then passes through a divider, the division ratio of which is selected by configuration options to provide a range of buzzer frequencies from $f_S/2^2$ to $f_S/2^9$.

The clock source that generates f_S, which in turn controls the buzzer frequency, can originate from two different sources, the Int.RCOSC (Internal RC oscillator) or the System oscillator/4, the choice of which is determined by the f_S clock source configuration option. Note that the buzzer frequency is controlled by configuration options, which select both the source clock for the internal clock f_S and the internal division ratio. There are no internal registers associated with the buzzer frequency.

If the configuration options have selected both pins PA0 and PA1 to function as a BZ and $\overline{BZ}$ complementary pair of buzzer outputs, then for correct buzzer operation it is essential that both pins must be setup as outputs by setting bits PAC0 and PAC1 of the PAC port control register to zero. The PA0 data bit in the PA data register must also be set high to enable the buzzer outputs, if set low, both pins PA0 and PA1 will remain low. In this way the single bit PA0 of the PA register can be used as an on/off control for both the BZ and $\overline{BZ}$ buzzer pin outputs. Note that the PA1 data bit in the PA register has no control over the $\overline{BZ}$ buzzer pin PA1.

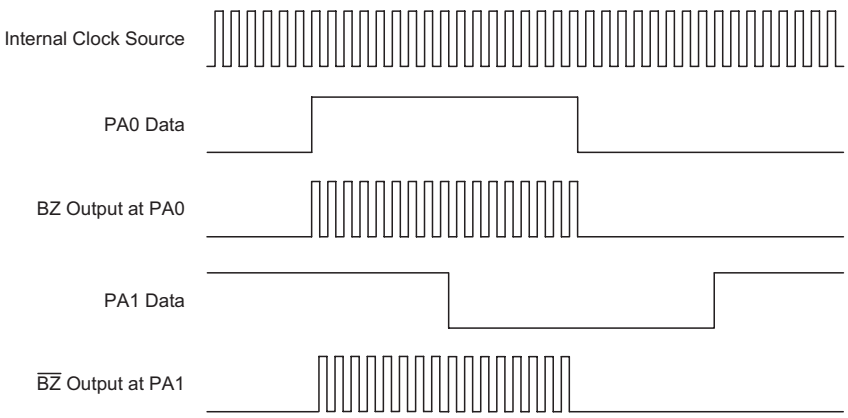
If the configuration options have selected that only the PA0 pin is to function as a BZ buzzer pin, then the PA1 pin can be used as a normal I/O pin. For the PA0 pin to function as a BZ buzzer pin, PA0 must be setup as an output by setting bit PAC0 of the PAC port control register to zero. The PA0 data bit in the PA data register must also be set high to enable the buzzer output, if set low pin PA0 will remain low. In this way the PA0 bit can be used as an on/off control for the BZ buzzer pin PA0. If the PAC0 bit of the PAC port control register is set high, then pin PA0 can still be used as an input even though the configuration option has configured it as a BZ buzzer output.

Note that no matter what configuration option is chosen for the buzzer, if the port control register has setup the pin to function as an input, then this will override the configuration option selection and force the pin to always behave as an input pin. This arrangement enables the pin to be used as both a buzzer pin and as an input pin, so regardless of the configuration option chosen; the actual function of the pin can be changed dynamically by the application program by programming the appropriate port control register bit.

Note:The above drawing shows the situation where both pins PA0 and PA1 are selected by a configuration option to be BZ and $\overline{BZ}$ buzzer pin outputs. The Port Control Register of both pins must have already been setup as outputs. The data setup on pin PA1 has no effect on the buzzer outputs.

PAC Register PAC.0	PAC Register PAC.1	PA data Register PA.0	PA data Register PA.1	Output Function
0	0	0	X	PA0=0, PA1=0
0	0	1	X	PA0=BZ, PA1=$\overline{BZ}$
0	1	0	X	PA0=0, PA1=Input
0	1	1	X	PA0=BZ, PA1=Input
1	0	0	X	PA0=Input, PA1=0
1	1	X	X	PA0=Input, PA1=Input

PA0/PA1 Pin Function Control



Buzzer Output Pin Control

Power Down Operation – HALT

The Power-down mode is initialised by a ″HALT″ instruction and results in the following.

- The system oscillator turns off but the WDT oscillator keeps running if the WDT oscillator or the real time clock is selected.
- The contents of the Data Memory and the registers remain unchanged.
- The WDT is cleared and starts recounting if the WDT clock is sourced from the WDT oscillator or the real time clock oscillator.
- All I/O ports maintain their original status.
- The PDF flag is set but the TO flag is cleared.
- The LCD driver keeps running if the WDT OSC or RTC OSC is selected.

The system leaves the HALT or IDLE mode by means of an external reset, an interrupt, an external falling edge signal on port A, or by a WDT overflow. An external reset causes a device initialisation, while a WDT overflow performs a ″warm reset″. After examining the TO and PDF flags, the reason for the device reset can be determined. The PDF flag is cleared by a system power-up or by executing the ″CLR WDT″ instruction, and is set by executing the ″HALT″ instruction. The TO flag is set if a WDT time-out occurs, and causes a wake-up that only resets the program counter and the SP, and leaves the others in their original state.

A port A wake-up and interrupt methods can be considered as a continuation of normal execution. Each pin of port A can be independently selected to wake-up the device using configuration options. After awakening from an I/O port stimulus, the program will resume execution at the next instruction. However, if awakening from an interrupt, two sequences may occur. 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. But if the interrupt is enabled, and the stack is not full, a regular interrupt response takes place.

When an interrupt request flag is set before entering the Power-down mode, the system cannot be awakened using that interrupt.

If a wake-up events occur, it takes 1024 t_{SYS} (system clock periods) to resume normal operation. In other words, a dummy period is inserted after the wake-up. If the wake-up results from an interrupt acknowledgment, the actual interrupt subroutine execution is delayed by more than one cycle. However, if the wake-up results in the next instruction execution, the execution will be performed immediately after the dummy period is finished.

To minimise power consumption, all the I/O pins should be carefully managed before entering the Power-down mode.

When a HALT instruction is executed, the CPU will stop running, and the related OSC and peripheral clocks will be set by the HALTC register. The HALTC register will only take effect when the system clock (f_{SYS}) is set to OSC.

Note: HALTC has no effect if the 32K oscillator is set as the system clock.

Bit No.	Label	Function
0	LCDON	Specifies the LCD condition in the Power-down mode 1: LCD module remains on (if OSCON=1) and ignores the configuration option setting 0: LCD condition decided by the LCD_ON configuration option
1~6	—	Unused bit, read as ″0″
7	OSCON	System clock oscillator On/off during Power-down mode setting. 0: Oscillator stops running. All related peripherals will lose their clock and stop functioning. (Register bit 0 will be ignored) 1: Oscillator keeps running. (All peripheral keep running, except for the special setting of Bit 0)

HALTC (17H) Register

Reset

There are three ways in which a reset may occur.

- $\overline{\text{RES}}$ is reset during a normal operation
- $\overline{\text{RES}}$ is reset during Power-down
- WDT time-out is reset during normal operation

The WDT time-out during a HALT or IDLE differs from other reset conditions, as it performs a ″warm reset″ that resets only the program counter and SP and leaves the other circuits in their original state. Some registers remain unaffected during any other reset conditions. Most registers are reset to their initial conditions once the reset conditions are met. By examining the PDF and TO flags, the program can distinguish between different reset types.

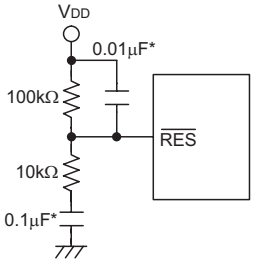
TO	PDF	RESET Conditions
0	0	$\overline{\text{RES}}$ reset during power-up
u	u	$\overline{\text{RES}}$ reset during normal operation
0	1	$\overline{\text{RES}}$ Wake-up HALT
1	u	WDT time-out during normal operation
1	1	WDT Wake-up HALT

Note: ″u″ stands for unchanged

To guarantee that the system oscillator has started and has stabilised, the SST (System Start-up Timer) provides an extra-delay of 1024 system clock pulses when the system awakes from the Power-down mode or during power-up. When awakening from the Power-down mode or during a system power-up, the SST delay is added.

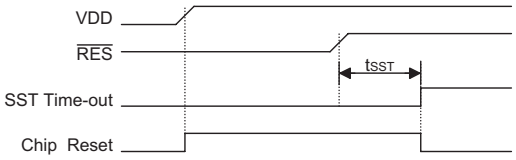
The functional unit chip reset status is shown below.

Program Counter	000H
Interrupt	Disabled
Prescaler, Divider	Cleared
WDT	Cleared. After master reset, WDT starts counting
Timer/Event Counter	Off
Input/output Ports	Input mode
Stack Pointer	Points to the top of the stack

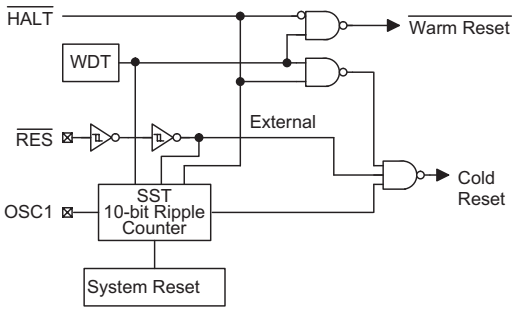


Reset Circuit

Note: ″*″ Make the length of the wiring, which is connected to the $\overline{\text{RES}}$ pin as short as possible, to avoid noise interference.



Reset Timing Chart



Reset Configuration

The register states are summarised below:

Register	Reset (Power On)	WDT Time-out (Normal Operation)	$\overline{\text{RES}}$ Reset (Normal Operation)	$\overline{\text{RES}}$ Reset (HALT)	WDT Time-out (HALT)*
MP0	-xxx xxxx	-uuu uuuu	-uuu uuuu	-uuu uuuu	-uuu uuuu
MP1	-xxx xxxx	-uuu uuuu	-uuu uuuu	-uuu uuuu	-uuu uuuu
BP	---- ---0	---- ---0	---- ---0	---- ---0	---- ---u
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
Program Counter	0000H	0000H	0000H	0000H	0000H
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	--xx xxxx	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu
RTCC	--00 0111	--00 0111	--00 0111	--00 0111	--uu uuuu
STATUS	--00 xxxx	--1u uuuu	--uu uuuu	--01 uuuu	--11 uuuu
INTC0	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
TMR0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0C	00-0 1000	00-0 1000	00-0 1000	00-0 1000	uu-u uuuu
TMR1HH	---- --xx	---- --uu	---- --uu	---- --uu	---- --uu
TMR1H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR1L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR1C	0000 1--0	0000 1--0	0000 1--0	0000 1--0	uuuu u--0
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PB	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu
PBC	---- 1111	---- 1111	---- 1111	---- 1111	---- uuuu
ADCR	0000 x000	0000 x000	0000 x000	0000 x000	0000 x000
ADCD	---- -111	---- -111	---- -111	---- -111	---- -uuu
WDTC	---- ss01	---- ss01	---- ss01	---- ss01	---- uuuu
WDTD	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
INTC1	-000 -000	-000 -000	-000 -000	-000 -000	-uuu -uuu
CHPRC	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
HALTC	0--- ---0	0--- ---0	0--- ---0	0--- ---0	u--- ---u
EADCR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu

Note: ″*″ stands for warm reset

″u″ stands for unchanged

″x″ stands for unknown

″s″ for special case, it depends on the option table (please see the WDT chapter for the detail)

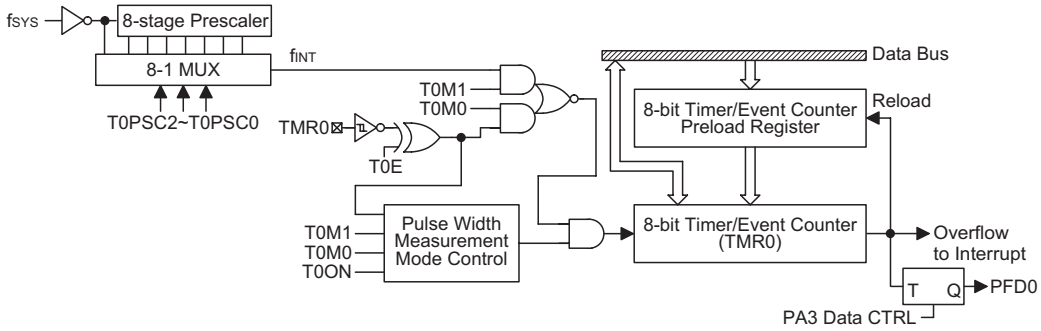
Timer/Event Counter

Two timer/event counters are integrated within the microcontroller. The Timer/Event Counter 0 contains a 8-bit programmable count-up counter whose clock may come from an external source or an internal clock source. The internal clock source comes from f_{SYS}. The Timer/Event Counter 1 contains a 18-bit programmable count-up counter whose clock may come from an external source or an internal clock source. The internal clock source comes from $f_{SYS}/4$ or 32768Hz selected by configuration option. The external clock input allows external events to be counted, time intervals or pulse widths to be measured, or to generate an accurate time base.

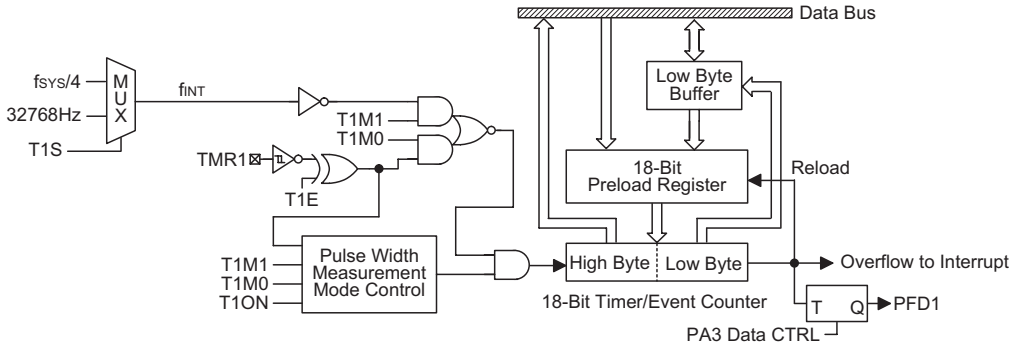
There are two registers related to the Timer/Event Counter 0, TMR0 and TMR0C. Two physical registers are mapped to the TMR0 location. Writing to TMR0 places the start value into the Timer/Event Counter 0 register while reading TMR0 reads directly the contents of the Timer/Event Counter 0. TMR0C is a timer/event counter control register, which defines some options. There are four registers related to Timer/Event Counter 1, TMR1HH, TMR1H, TMR1L and TMR1C. Writing to TMR1L and TMR1H will only put the required data into two internal lower-order byte buffers, each of which is 8-bits. Writing to TMR1HH will transfer the specified data and the contents of the lower-order byte buffers into the TMR1HH, TMR1H and TMR1L registers respectively. The Timer/Event Counter 1 preload register is changed by each write to the TRM1HH register operation. Reading TMR1HH will latch the contents of TMR1HH to the destination and latch the TMR1H and TMR1L counters to the lower-order byte buffers, respectively. Reading the TMR1H and TMR1L registers will read the contents of the lower-order byte buffers. TMR1C is the Timer/ Event Counter 1 control register, which defines the operating mode, counting enable or disable and an active edge.

The T0M0, T0M1 (TMR0C) and T1M0, T1M1 (TMR1C) bits define the operation mode. The event count mode is used to count external events, which means that the clock source comes from an external pin, TMR0 or TMR1. The timer mode functions as a normal timer with the clock source coming from the internally selected clock source. Finally, the pulse width measurement mode can be used to measure a high or low level duration of an external signal on pin TMR0 or TMR1. This measurement uses the internally selected clock source.

To enable a counting operation, the Timer ON bit, (T0ON: bit 4 of TMR0C; T1ON: 4 bit of TMR1C) should be set to 1. In the pulse width measurement mode, the T0ON/T1ON bit is automatically cleared after the measurement cycle is completed. But in the other two modes, the T0ON/T1ON bits can only be reset using instructions. The Timer/Event Counter 0/1 overflow is one of the wake-up sources. The timers and can also be used as the source clock for the PFD (Programmable Frequency Divider) output on PA3. This function is selected by a configuration option. Only one Timer/Event Counter clock source (PFD0 or PFD1) can be used as the PFD clock source, chosen by a configuration option.



Timer/Event Counter 0



Timer/Event Counter 1

Bit No.	Label	Function
0 1 2	T0PSC0 T0PSC1 T0PSC2	To define the prescaler stages. T0PSC2, T0PSC1, T0PSC0= 000: $f_{INT}=f_{SYS}$ 001: $f_{INT}=f_{SYS}/2$ 010: $f_{INT}=f_{SYS}/4$ 011: $f_{INT}=f_{SYS}/8$ 100: $f_{INT}=f_{SYS}/16$ 101: $f_{INT}=f_{SYS}/32$ 110: $f_{INT}=f_{SYS}/64$ 111: $f_{INT}=f_{SYS}/128$
3	T0E	Defines the timer/event counter TMR0 pin active edge: In the Event Counter Mode - T0M1,T0M0 = 0,1: 1:count on falling edge; 0:count on rising edge In the Pulse Width measurement mode - T0M1,T0M0 = 1,1 1: start counting on rising edge, stop on falling edge; 0: start counting on falling edge, stop on rising edge
4	T0ON	Enable/disable timer counting - 0=disabled; 1=enabled
5	—	Unused bit, read as ″0″
6 7	T0M0 T0M1	Operation Mode Definition bits T0M1, T0M0: 01= Event count mode - External clock 10= Timer mode - Internal clock 11= Pulse Width measurement mode - External clock 00= Unused

TMR0C (0EH) Register

Bit No.	Label	Function
0	T132KON	Defines if the 32768 Oscillator is running or not. (See Note) 0: 32768 Oscillator off if no other peripherals are using it 1: 32768 Oscillator starts to run or keeps running.
1~2	—	Unused bit, read as ″0″
3	T1E	Defines the timer/event counter TMR1 active edge: In the Event Counter Mode - T1M1,T1M0=0,1: 1:count on falling edge; 0:count on rising edge In the Pulse Width measurement mode - T1M1,T1M0=1,1: 1: start counting on rising edge, stop on falling edge; 0: start counting on falling edge, stop on rising edge
4	T1ON	Enable/disable timer counting - 0=disabled; 1=enabled
5	T1S	Defines the TMR1 internal clock source - 0=$f_{SYS}/4$; 1=32768Hz
6 7	T1M0 T1M1	Operation Mode Definition bits T1M1, T1M0: 01= Event count mode - External clock 10= Timer mode - Internal clock 11= Pulse Width measurement mode - External clock 00= Unused

TMR1C (11H) Register

Note: The 32768Hz oscillator enable will be logical OR function of the T132KON bit and any configuration option that chooses the 32768Hz oscillator. That is, the 32768Hz OSC will be enabled if any related function enables it, and will be turned off if no function enables it.

If PA3 is selected to be a PFD output, there are two types of selections. One is to use PFD0 as the PFD output, the other is to use PFD1 as the PFD output. PFD0 and PFD1 are the timer overflow signals of the Timer/Event Counter 0 and Timer/Event Counter 1 respectively. No matter what the operation mode is, writing a 0 to ET0I or ET1I disables the related interrupt service. When the PFD function is selected, executing a ″SET [PA].3″ instruction will enable the PFD output while executing a ″CLR [PA].3″ instruction will disable the PFD output.

In the case of timer/event counter OFF condition, writing data to the timer/event counter preload register also reloads that data to the timer/event counter. But if the timer/event counter is running, data written to the timer/event counter will be loaded only to the timer/event counter preload register. The timer/event counter still continues its operation until an overflow occurs.

When the timer/event counter is read, the clock will be blocked to avoid errors, which may result in a counting error, and should be taken into account by the programmer. It is strongly recommended to load a desired value into the TMR0/TMR1 register first, before turning on the related timer/event counter, for proper operation since the initial value of TMR0/TMR1 is unknown. After this procedure, the timer/event function can be operated normally.

Bit0~bit2 of TMR0C can be used to define the pre-scaling stages of the timer/event counter internal clock. The overflow signal of timer/event counter can be used to generate the PFD signal.

Input/Output Ports

There are 10 bidirectional input/output lines in the microcontroller, labeled as PA and PB, which are mapped to the data memory of [12H] and [14H] respectively. All of 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 or 14H). For output operations, all the data is latched and remains unchanged until the output latch is rewritten.

Each I/O line has its own control register, known as PAC and PBC, to control the input/output configuration. With this control register, either CMOS outputs or Schmitt trigger inputs with or without pull-high resistor structures can be reconfigured dynamically under software control. To function as an input, the corresponding latch of the control register must be written with a ″1″. The input source also depends on the control register. If the control register bit is ″1″, the input will read the pad state. To function as an output the the control register bit should be set to ″0″. The latter is possible in the ″read-modify-write″ instruction.

After a device reset, these input/output lines will default to inputs and remain at a high level or in a floating state, depending upon the pull-high configuration options. Each bit of these input/output latches can be set or cleared by a ″SET [m].i″ and ″CLR [m].i″ (m=12H or 14H) instruction.

Some instructions first input data and then follow the output operations. For example, ″SET [m].i″, ″CLR [m].i″, ″CPL [m]″, ″CPLA [m]″ 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 of waking-up the device.

Each I/O port has a pull-high option. Once a pull-high option is selected, the I/O port has a pull-high resistor connected. Take note that a non-pull-high I/O port setup as an input mode will be in a floating condition.

Pins PA0, PA1, PA3, PA4, PA5, PA6 and PA7 are pin-shared with BZ, $\overline{BZ}$, PFD, TMR0, TMR1, $\overline{INT0}$ and $\overline{INT1}$ pins respectively.

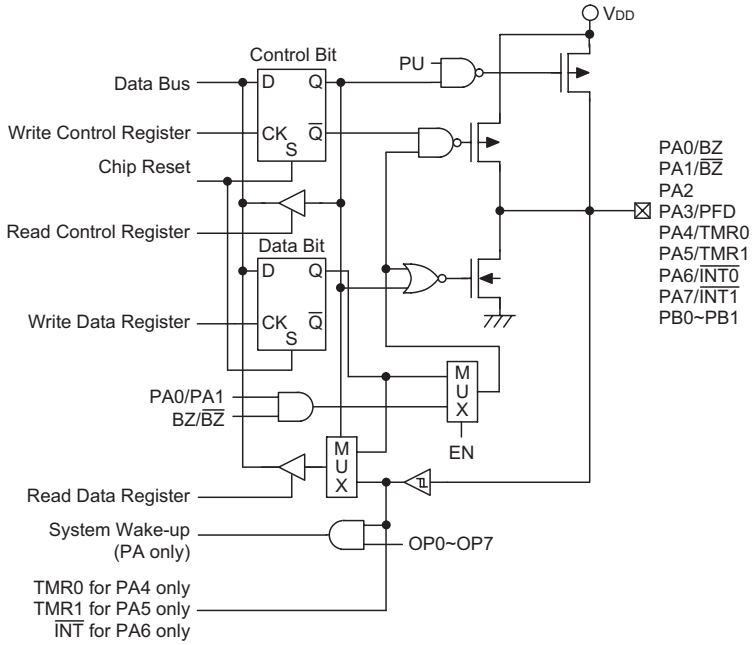
PA0 and PA1 are pin-shared with BZ and $\overline{BZ}$ signal, respectively. If the BZ/$\overline{BZ}$ configuration option is selected, the output signals in the output mode of PA0/PA1 will be the buzzer signal. The input mode always retains its original function. Once the BZ/$\overline{BZ}$ configuration option is selected, the buzzer output signals are controlled by the PA0 data register.

The PA0/PA1 I/O function is shown below.

PA0 I/O	I	I	O	O	O	O	O	O	O	O
PA1 I/O	I	O	I	I	I	O	O	O	O	O
PA0 Mode	X	X	C	B	B	C	B	B	B	B
PA1 Mode	X	C	X	X	X	C	C	C	B	B
PA0 Data	X	X	D	0	1	D_0	0	1	0	1
PA1 Data	X	D	X	X	X	D1	D	D	X	X
PA0 Pad Status	I	I	D	0	B	D_0	0	B	0	B
PA1 Pad Status	I	D	I	I	I	D_1	D	D	0	B

Note: ″I″ input; ″O″ output
″D, D0, D1″ Data
″B″ buzzer option, BZ or $\overline{BZ}$
″X″ don′t care
″C″ CMOS output

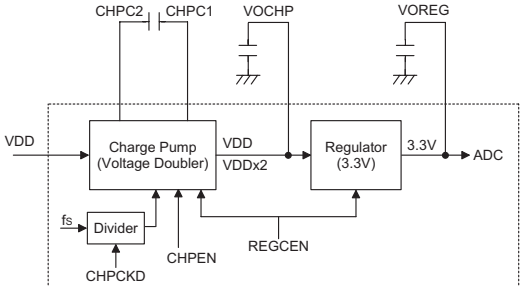
It is recommended that if there are unused or not bonded out I/O lines then they should be setup as output pins using software instructions to avoid consuming power. If setup as inputs and left floating this may result in unnecessary increased power consumption.



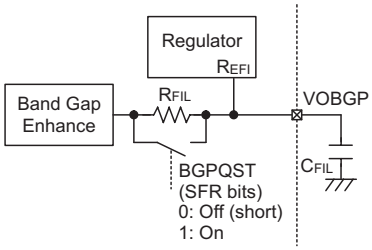
Input/Output Ports

Charge Pump and Voltage Regulator

A charge pump and voltage regulator are integrated within the device. The charge pump can be enabled/disabled by the application program. The charge pump uses VDD as its input, and has the function of doubling the V_{DD} voltage, therefore the charge pump output voltage will be $V_{DD}\times2$. The regulator can generate a stable voltage of 3.3V for the ADC and also can provide an external bridge sensor excitation voltage or supply a reference voltage for other applications. The user needs to ensure that the charge pump output voltage is greater than 3.6V to ensure that the regulator is capable of generating the required 3.3V voltage output.



Additionally, the device also includes a band gap voltage generator for the 1.5V low temperature sensitive reference voltage. This reference voltage is used as the zero adjustment and for a single end type reference voltage.



R_{FIL} is about 100kΩ and the recommend value for C_{FIL} is 10μF.

Note: The VOBGP signal is an internal signal only and only the recommend C_{FIL} should be connected.

There is a single register associated with this module named CHPRC. The CHPRC is the Charge Pump/Regulator Control register, which controls the charge pump on/off function, the regulator on/off functions as well as setting the clock divider value to generate the clock for the charge pump.

The CHPCKD4~CHPCKD0 bits are used to set the clock divider to generate the desired clock frequency for proper charge pump operation. The actual frequency is determined by the following formula.

Actual Charge Pump Clock= (f_{SYS}/16)/(CHPCKD +1).

Bit No.	Label	Function
0	REGCEN	Enable/disable Regulator/Charge-Pump module. (1=enable; 0=disable)
1	CHPEN	Charge Pump Enable/disable setting. (1=enable; 0=disable) Note: this bit will be ignored if the REGCEN bit is disabled
2	BGPQST	Bandgap quick start-up function 0: R short, quick start up 1: R connected, normal RC filter mode Each time REGCEN changes from 0 to 1, that is when the regulator turns on, this bit should be set to 0 and then set to 1 to ensure a quick start up. The minimum time to keep the bit low should be about 2ms.
3~7	CHPCKD0~ CHPCKD4	Charge pump clock divider. These 5 bits form a clock divider with a division ratio range of 1 to 32. Charge Pump clock = (f_{SYS}/16) / (CHPCKD+1)

CHPRC (1FH) Register

REGCEN	CHPEN	Charge Pump	VOCHP Pin	Regulator	VOREG Pin	OPA ADC	Description
0	X	OFF	V_{DD}	OFF	Hi-Impedance	Disable	Complete module is disabled, OPA/ADC will have no Power
1	0	OFF	V_{DD}	ON	3.3V	Active	Used when V_{DD} is greater than 3.6V
1	1	ON	$2\times V_{DD}$	ON	3.3V	Active	Use whefor V_{DD} is less than 3.6V (V_{DD}=2.2V~3.6V)

The suggested charge pump clock frequency is 20kHz. The application needs to set the correct value to get the desired clock frequency. For a 4MHz application, the CHPCKD bits should be set to the decimal value 11, and for a 2MHz application, the bits should be set to 5.

The REGCEN bit in the CHPRC register is the Regulator/Charge-pump module enable/disable control bit. If this bit is disabled, then the regulator will be disabled and the charge pump will be also be disabled to save power. When REGCEN = 0, the module will enter the Power Down Mode ignoring the CHPEN setting. The ADC and OPA will also be disabled to reduce power.

If REGCEN is set to ″1″, the regulator will be enabled. If the CHPEN bit is enabled, the charge pump will be active and will use VDD as its input to generate the double voltage output. This double voltage will then be used as the input voltage for the regulator. If CHPEN is set to ″0″, the charge pump is disabled and the charge pump output will be equal to the charge pump input, VDD.

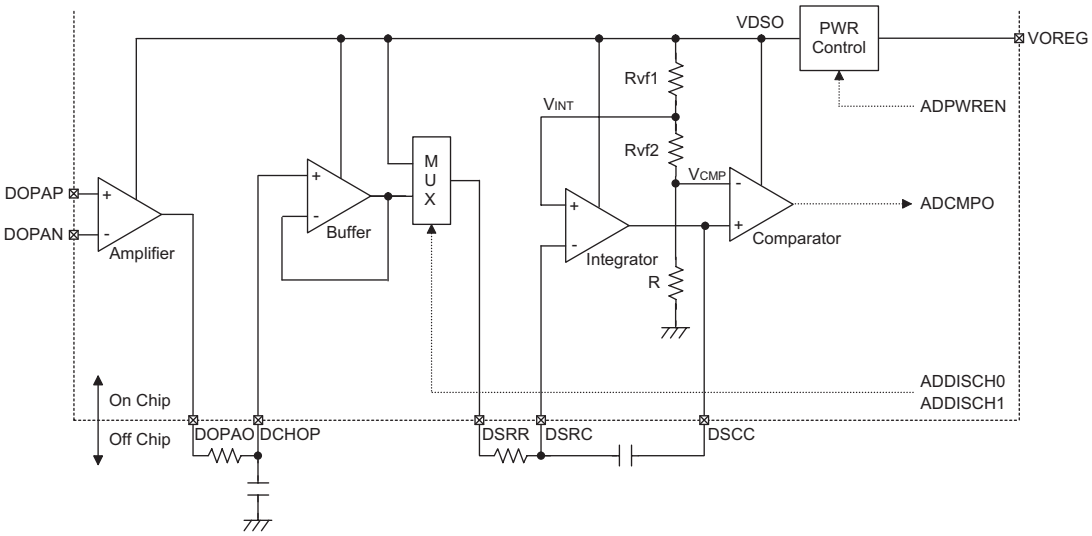
It is necessary to carefully manage the V_{DD} voltage. If the voltage is less than 3.6V, then CHPEN should be set to 1 to enable the charge pump, otherwise CHPEN should be set to zero. If the Charge pump is disabled and V_{DD} is less than 3.6V then the output voltage of the regulator will not be guaranteed.

ADC – Dual Slope

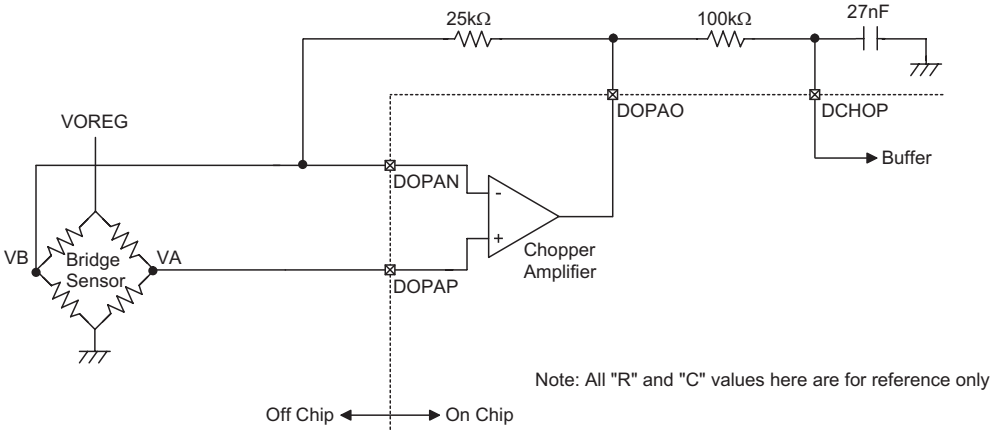
A Dual Slope A/D converter is implemented within the microcontroller. The dual slope module includes an Operational Amplifier and a buffer for the amplification of differential signals and an Integrator and comparator for the main dual slope AD converter.

There are 3 special function registers related to the ADC function known as ADCR, ADCD and EADCR. The ADCR register is the A/D control register, which controls the ADC block power on/off, the chopper clock on/off, the charge/discharge control and is also used to read out the comparator output status. The ADCD register is the A/D Chopper clock divider register, which defines the chopper clock to the ADC module. The EADCR register is the enhanced A/D control register, which defines the Auto Mode Dual Slope ADC function.

The ADPWREN bit in the ADCR register, is used to control the ADC module on/off function. The ADCCKEN bit in the ADCR register is used to control the chopper clock on/off function. When the ADCCKEN bit is set to ″1″ it will enable the Chopper clock, with the clock frequency defined by the ADCD registers. The ADC module includes the OPA, buffer, integrator and comparator, however the Bandgap voltage generator is independent of this module. It will be automatically enabled when the

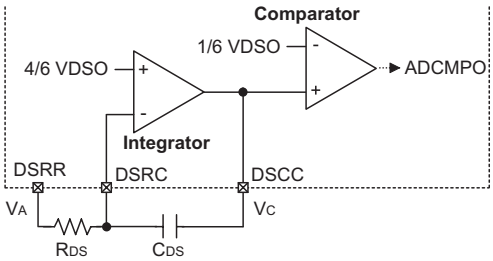


Note: The V_{INT}, V_{CMP} signals can come from different R groups which are selected by software registers.



regulator is enabled, and also be disabled when the regulator is disabled. The application program should enable the related power to permit them to function and disable them when idle to conserve power. The charge/discharge control bits, ADDISCH1~ ADDISCH0, are used to control the Dual slope circuit charging and discharging behavior. The ADCMPO bit is read only for the comparator output, while the ADINTM bits can set the ADCMPO trigger mode for interrupt generation.

The following descriptions are based on the fact that ADRR0=0



The amplifier and buffer combination, form a differential input pre-amplifier which amplifies the sensor input signal.

The combination of the Integrator, the comparator, the resistor Rds, between DSRR and DSRC and the capacitor Cds, between DSRC and DSCC form the main body of the Dual slope ADC.

The Integrator integrates the output voltage increase or decrease and is controlled by the ″Switch Circuit″ - refer to the block diagram. The charge and discharge curves are illustrated by the following.

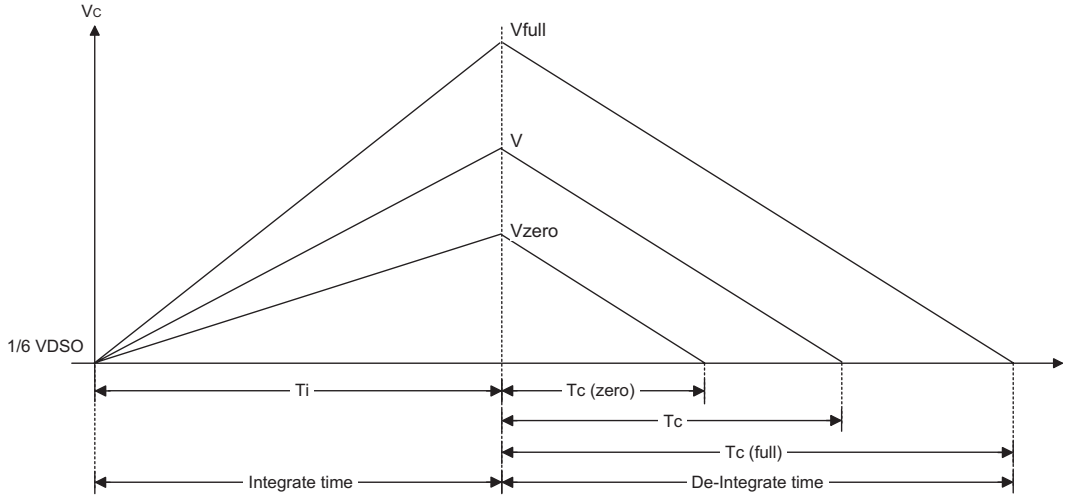
The ″comparator″ will switch the state from high to low when VC, which is the DSCC pin voltage,drops to less than 1/6 VDSO.

In general applications, the application program will switch the ADC to the charging mode for a fixed time called Ti, which is the integrating time. It will then switch to the dis-charging mode and wait for V_c to drop to less than 1/6VDSO. At this point the comparator will change

state and store the time taken, Tc, which is the de-integrating time. The following formula 1 can then be used to calculate the input voltage V_A.

formula 1: $V_A = (1/3) \times VDSO \times (2 - Tc/Ti)$.
(Based on ADRR0=0)

In user applications, it is required to choose the correct value of R_{DS} and C_{DS} to determine the Ti value, to allow the V_C value to operate between 5/6VDSO and 1/6VDSO. Vfull cannot be greater than 5/6VDSO and Vzero cannot be less than 1/6VDSO



Bit No.	Label	Function
0	ADPWREN	Dual slope block (including input OP) power on/off switch. 0: disable Power 1: Power source comes from the regulator.
1~2	ADDISCH0~ ADDISCH1	Defines the ADC discharge/charge. 00: reserved 01: charging - Integrator input connect to buffer output 10: discharging - Integrator input connect to VDSO) 11: reserved
3	ADCMPO	Dual Slope ADC - last stage comparator output. Read only bit, write data instructions will be ignored. During the discharging state, when the integrator output is less than the reference voltage, the ADCMPO will change from high to low.
4~5	ADINTM0~ ADINTM1	ADC integrator interrupt mode definition. These two bits define the ADCMPO data interrupt trigger mode: 00: no interrupt 01: rising edge 10: falling edge 11: both edge
6	ADCCKEN	ADC OP chopper clock source on/off switch. 0: disable 1: enable (clock value is defined by ADCD register)
7	ADRR0	ADC resistor selection 0: (V_{INT}, V_{CMP})= (4/6 VOREG, 1/6 VOREG) 1: (V_{INT}, V_{CMP})= (4.4/6 VOREG, 1/6 VOREG)

ADCR (18H) Register

Bit No.	Label	Function
0 1 2	ADCD0 ADCD1 ADCD2	Defines the chopper clock. ADCCKEN should be enabled. The suggested clock value should be around 10kHz. The chopper clock definitions are: 0: clock= (f_{SYS}/32)/1 1: clock= (f_{SYS}/32)/2 2: clock= (f_{SYS}/32)/4 3: clock= (f_{SYS}/32)/8 4: clock= (f_{SYS}/32)/16 5: clock= (f_{SYS}/32)/32 6: clock= (f_{SYS}/32)/64 7: clock= (f_{SYS}/32)/128
3~7	—	Reserved

ADCD (1AH) Register

Bit No.	Label	Function
0~1	—	Unused bit, read as ″0″
2	CHGTS	Select the ADC CHARGE state timer 0: timer 0 1: timer1
3	DISTS	Select the ADC discharge state timer 0: timer 0 1: timer 1
4	CHGCMP	Select if the charge timer (note 1) will auto-start by the ADCMPO result or not. ASTEN should be enabled. 0: immediately auto start timer counting (note 3) when charging begins 1: Wait for the ADCMPO rising edge to auto start timer counting (note 3) when charging begins
5	ASTEN	Dual slope auto start enable. When this function is enabled, the charging timer (note 1) will auto start (note 3) when the user sets ADDISCH to the charging mode. The start method is determined by CHGCMP. 0: disable 1: enable this auto function
6	ADISEN	Dual slope auto discharge enable. When this function enable and ADISCH is set to the charging mode and the charging timer (note 1)) run overflow, the charging timer will auto stop (note 3) and ADDISCH will be auto set to the discharging mode (note 4) and the discharging timer (note 2) will auto start counting (note 3). 0: disable 1: enable this auto function
7	AENDEN	Dual slope Auto End Enable. When this function is enabled and ADDISCH is set to the discharge mode and detects the ADCMPO falling edge, the discharging timer will auto stop (note 2). 0: disable 1: enable this auto function

Note: 1: charge timer means the timer0 or timer 1 that select in the CHGTS registers.

2: discharge timer means the timer0 or timer 1 that select in the DISTS registers.

3: timer auto start only set the TxS to 1, and auto stop only set the TxON to 0

4: auto set the discharge mode only set the ADISCH1/0 to 1/0.

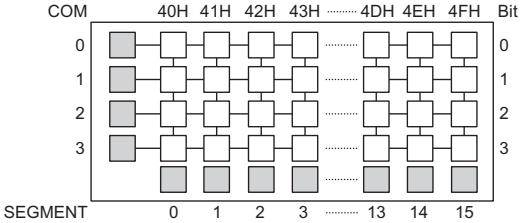
EADCR (1BH) Register

LCD Display Memory

The device provides an area of embedded data memory for the LCD display. This area is located at 40H to 4FH in Bank 1 of the Data Memory. The bank pointer BP, enables either the General Purpose Data Memory or LCD Memory to be chosen. When BP is set to ″1″, any data written into location range 40H~4FH will affect the LCD display. When the BP is cleared to ″0″, any data written into 40H~4FH will access the general purpose data memory. The LCD display memory can be read and written to only indirectly using MP1. When data is written into the display data area, it is automatically read by the LCD driver which then generates the corresponding LCD driving signals. To turn the display on or off, a ″1″ or a ″0″ is written to the corresponding bit of the display memory, respectively. The figure illustrates the mapping between the display memory and LCD pattern for the device.

The LCD clock frequency is determined by configuration options, and has a division ratio range of $fs/2^2$~$fs/2^8$. The LCD clock source frequency should be chosen to be as close as possible to 4kHz.

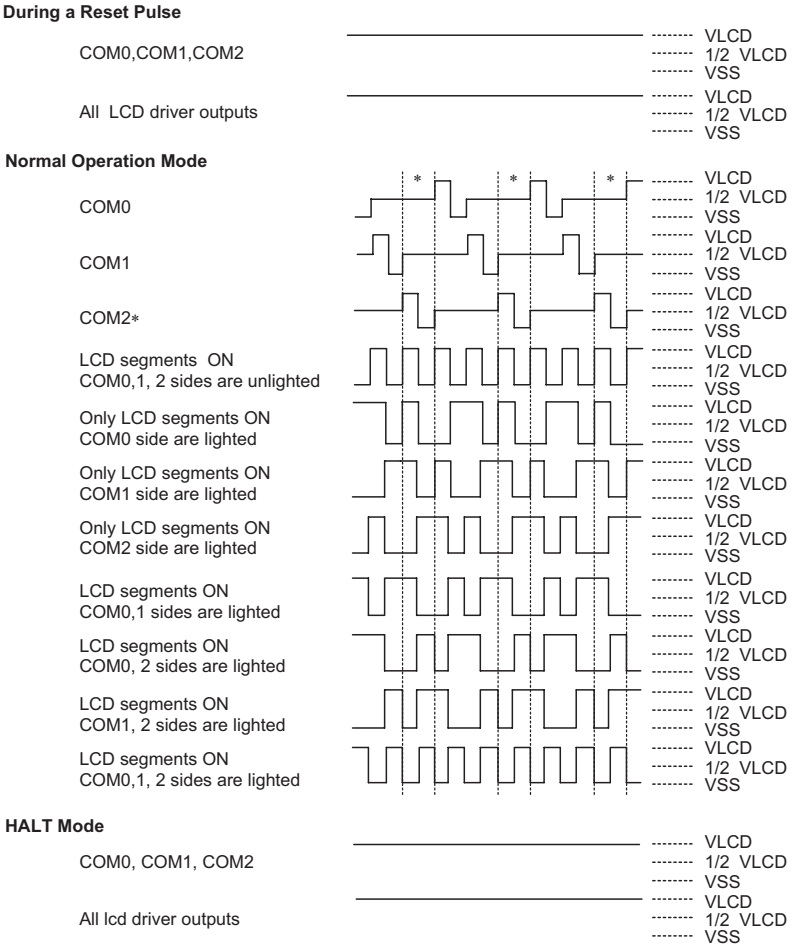
Note that the LCD frequency is controlled by configuration options, which select the internal division ratio.



Display Memory

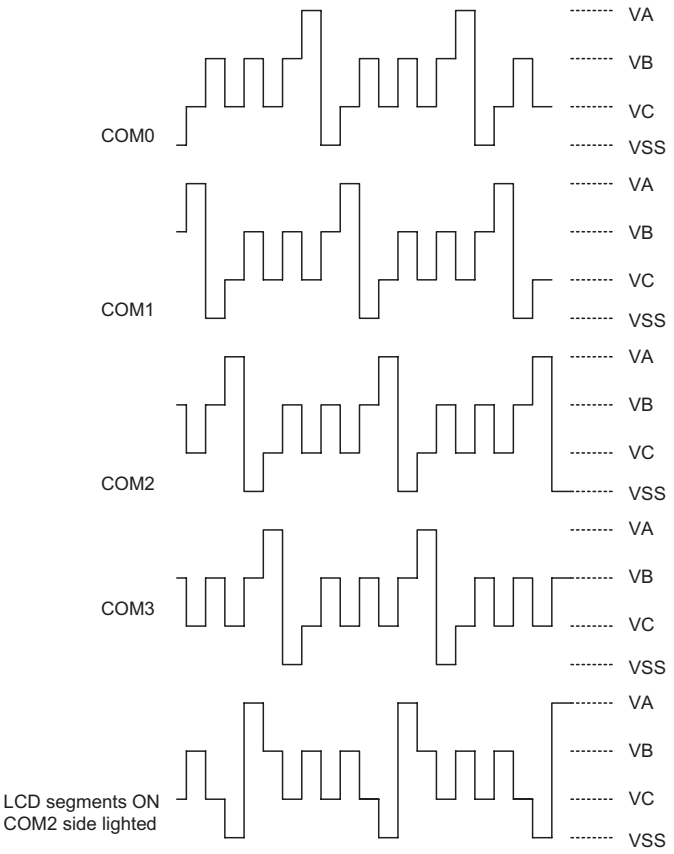
LCD Driver Output

The output number of the device LCD driver can be 16×2, 16×3 or 15×4 by configuration option (i.e., 1/2 duty, 1/3 duty or 1/4 duty). The bias type LCD driver can be ″R″ type or ″C″ type. If the ″R″ bias type is selected, no external capacitor is required. If the ″C″ bias type is



Note: "*" Omit the COM2 signal, if the 1/2 duty LCD is used.

LCD Driver Output (1/3 Duty, 1/2 Duty, R Type)



Note: 1/4 duty, 1/3 bias, R type: "VA" VLCD, "VB" 2/3 VLCD, "VC" 1/3 VLCD

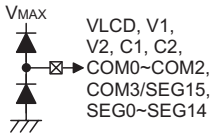
LCD Driver Output (1/4 Duty)

selected, a capacitor mounted between C1 and C2 pins is needed. The LCD driver bias voltage can be 1/2 bias or 1/3 bias by option. If 1/2 bias is selected, a capacitor mounted between V2 pin and ground is required. If 1/3 bias is selected, two capacitors are needed for V1 and V2 pins. Refer to application diagram.

LCD Type	R Type		C Type	
LCD Bias Type	1/2 bias	1/3 bias	1/2 bias	1/3 bias
V_{MAX}	If $V_{DD}>V_{LCD}$, then V_{MAX} connect to V_{DD}, else V_{MAX} connect to V_{LCD}			If $V_{DD}>\frac{3}{2}V_{LCD}$, then V_{MAX} connect to V_{DD}, else V_{MAX} connect to V1

LCD Segment pins used as Logic Outputs

The SEG0~SEG7 pins also can be setup for use logic outputs using configuration options. Once an LCD segment is selected for use as a logic output, the content of bit 0 of the related segment address in the LCD RAM will appear on the segment. SEG0~SEG7 are bits that can be individually optioned as logical outputs.



Low Voltage Reset/Detector Functions

There is a low voltage detector, LVD, and a low voltage reset circuit, LVR, implemented within the microcontroller. These two functions can be enabled/disabled via configuration options. Once the LVD option is enabled, the user can use the RTCC.3 bit to enable/disable the LVD circuit and read the LVD detector status from the RTCC.5 bit. Otherwise the LVD function is disabled.

When the LVD function configuration option is set to the enable state, the LVD voltage option will decide the detecting voltage. When the LVD configuration option is set to LVR+0.2, the actual LVD voltage depends upon the LVR options and will detect the VDD voltage. When LVD option is set to Regulator+ 0.2, the actual LVD voltage will detect the regulator input voltage.

The RTCC register definitions are listed below.

Bit No.	Label	Function
0~2	RT0~RT2	8 to 1 multiplexer control inputs to select the real clock prescaler output
3	LVDC	LVD enable/disable (1/0)
4	QOSC	32768Hz OSC quick start-up function 0/1: quick/slow start
5	LVDO	LVD detect output (1/0) 1: low voltage detect, read only
6~7	—	Unused bit, read as ″0″

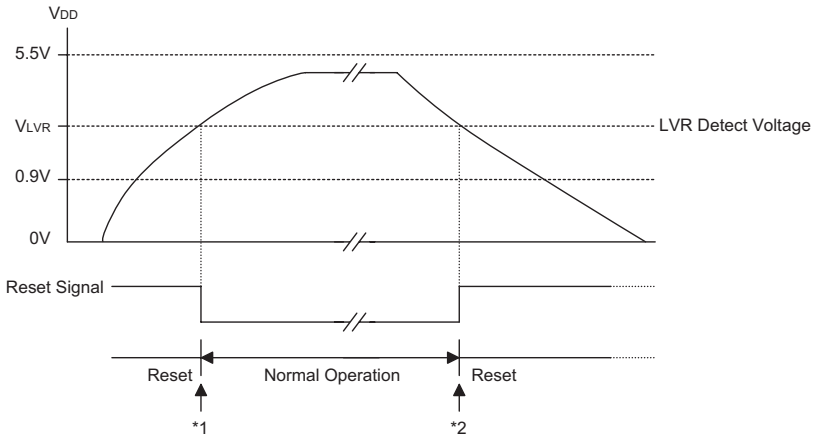
RTCC (09H) Register

The LVR has the same effect or function as the external RES signal which performs a device reset. When in the Power Down Mode, both the LVR and LVD are disabled.

The microcontroller provides a low voltage reset circuit in order to monitor the supply voltage of the device. If the supply voltage of the device is within the range 0.9V~V_{LVR}, such as what might happen when changing a battery, the LVR will automatically reset the device internally.

The LVR includes the following specifications:

- The low voltage, which is specified as 0.9V~V_{LVR}, has to remain within this range for a period of time greater than 1ms. If the low voltage state does not exceed 1ms, the LVR will ignore it will not perform a reset function.
- The LVR has an ″OR″ function with the external $\overline{RES}$ signal to perform a device reset.



Low Voltage Reset

Note: *1: To make sure that the system oscillator has stabilised, the SST provides an extra delay of 1024 system clock pulses before entering normal operation.

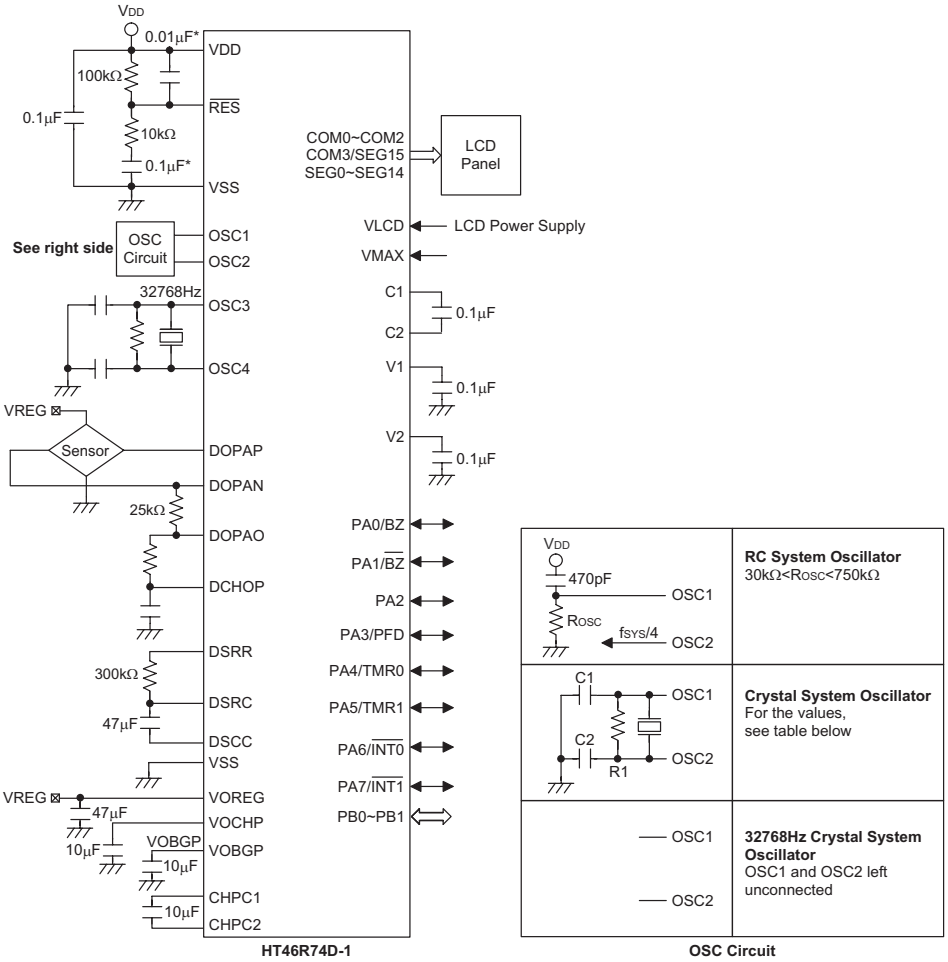
*2: Since a low voltage state has to be maintained in its original state for over 1ms, therefore after the 1ms delay, the device enters the reset mode.

Options

The following shows the options in the device. All these options should be defined in order to ensure a proper functioning system.

Options
OSC type selection. There are two types selection: Crystal OSC or RC OSC
System clock selection: OSC or RTC
f_S clock source. There are three types of selections: $f_{SYS}/4$, WDT or RTC
WDT clock source selection. There are various types of selections: system clock/4, WDT or RTC
WDT enable/disable selection. The WDT function can be enabled or disabled by this configuration option.
WDT time-out period selection. There are four types of selection: WDT clock source divided by $2^{16}/f_S$, $2^{15}/f_S$, $2^{14}/f_S$ or $2^{13}/f_S$
CLR WDT times selection. This option selects the instruction method of clearing the WDT. ″One time″ means that the ″CLR WDT″ instruction can clear the WDT. ″Two times″ means only if both the ″CLR WDT1″ and ″CLR WDT2″ instructions have been executed, can the WDT be cleared.
Buzzer output frequency selection. There are eight types of frequency signals for the buzzer output: $f_S/2^2$~$f_S/2^9$. ″f_S″
Wake-up selection. This option defines the wake-up capability. A falling edge on each external pin on PA has the capability to wake-up the device from a Power Down condition. Bit option.
Pull-high selection. Selects pull-high resistors when the I/O in in the input mode. Bit options.
I/O pins shared with other function selections. PA0/$\overline{BZ}$, PA1/BZ: PA0 and PA1 can be setup as I/O pins or buzzer outputs. PA3 can be setup as an I/O pin or as a PFD output.
PFD clock source selection: Timer/Event Counter 0 or Timer/Event Counter 1
LCD common selection. There are three types of selections: 2 commons (1/2 duty), 3 commons (1/3 duty) or 4 commons (1/4 duty).
LCD bias selection. This option is to determine what kind of bias is selected, 1/2 bias or 1/3 bias.
LCD segment logic output This option is to determine if pins SEG0~SEG7 are to be setup as logic outputs or setup as segment outputs - bit option
LCD driver clock frequency selection. There are a range of frequency signals for the LCD driver circuits: $f_S/2^2$~$f_S/2^8$
LCD ON/OFF when in Power Down Mode selection
LVR selection. LVR enable or disable option
LVD selection. LVD enable or disable option
LVR voltage selection: 2.2V or 3.3V
LVD voltage selection: LVR+0.2 or regulator+0.2
$\overline{INT}$ trigger edge selection: disable; high to low; low to high; low to high or high to low
Partial-lock selection: Page0~3, Page4~6, Page7.

Application Circuits



The following table shows the C1, C2 and R1 values corresponding to the different crystal values. For reference only.

Crystal or Resonator	C1, C2	R1
4MHz Crystal	0pF	10kΩ
4MHz Resonator	10pF	12kΩ
3.58MHz Crystal	0pF	10kΩ
3.58MHz Resonator	25pF	10kΩ
2MHz Crystal & Resonator	25pF	10kΩ
1MHz Crystal	35pF	27kΩ
480kHz Resonator	300pF	9.1kΩ
455kHz Resonator	300pF	10kΩ
429kHz Resonator	300pF	10kΩ
The function of the resistor R1 is to ensure that the oscillator will switch off should low voltage conditions occur. Such a low voltage, as mentioned here, is one which is less than the lowest value of the MCU operating voltage. Note however that if the LVR is enabled then R1 can be removed.		

Note: The resistance and capacitance for the reset circuit should be designed in such a way as to ensure that the VDD is stable and remains within a valid operating voltage range before bringing $\overline{RES}$ high.

"*" Make the length of the wiring, which is connected to the $\overline{RES}$ pin as short as possible, to avoid noise interference.

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 ← Program Counter+1
Program Counter ← 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 Program Counter ← Program Counter+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 Program Counter ←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 Program Counter ← Program Counter+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 Program Counter ← 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 Program Counter ← 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 Program Counter ← 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.

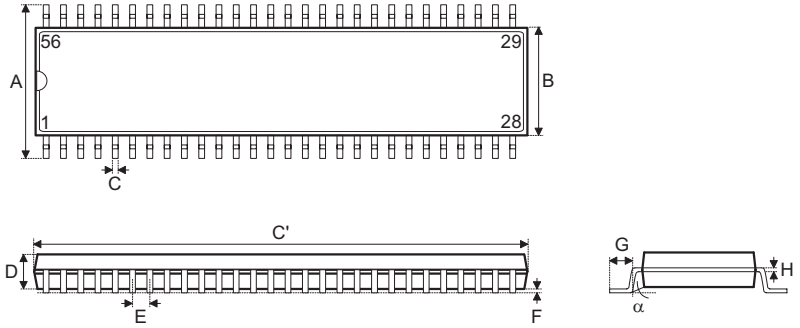
Operation ACC ← ACC ″XOR″ x

Affected flag(s)

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

Package Information

56-pin SSOP (300mil) Outline Dimensions



Symbol	Dimensions in mil		
	Min.	Nom.	Max.
A	395	—	420
B	291	—	299
C	8	—	12
C′	720	—	730
D	89	—	99
E	—	25	—
F	4	—	10
G	25	—	35
H	4	—	12
α	0°	—	8°

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: 86-21-6485-5560
Fax: 86-21-6485-0313
http://www.holtek.com.cn

Holtek Semiconductor Inc. (Shenzhen Sales Office)
5/F, Unit A, Productivity Building, Cross of Science M 3rd Road and Gaoxin M 2nd Road, Science Park, Nanshan District, Shenzhen, China 518057
Tel: 86-755-8616-9908, 86-755-8616-9308
Fax: 86-755-8616-9722

Holtek Semiconductor Inc. (Beijing Sales Office)
Suite 1721, Jinyu Tower, A129 West Xuan Wu Men Street, Xicheng District, Beijing, China 100031
Tel: 86-10-6641-0030, 86-10-6641-7751, 86-10-6641-7752
Fax: 86-10-6641-0125

Holtek Semiconductor Inc. (Chengdu Sales Office)
709, Building 3, Champagne Plaza, No.97 Dongda Street, Chengdu, Sichuan, China 610016
Tel: 86-28-6653-6590
Fax: 86-28-6653-6591

Holtek Semiconductor (USA), Inc. (North America Sales Office)
46729 Fremont Blvd., Fremont, CA 94538
Tel: 1-510-252-9880
Fax: 1-510-252-9885
http://www.holtek.com

Copyright © 2007 by HOLTEK SEMICONDUCTOR INC.

The information appearing in this Data Sheet is believed to be accurate at the time of publication. However, Holtek assumes no responsibility arising from the use of the specifications described. The applications mentioned herein are used solely for the purpose of illustration and Holtek makes no warranty or representation that such applications will be suitable without further modification, nor recommends the use of its products for application that may present a risk to human life due to malfunction or otherwise. Holtek's products are not authorized for use as critical components in life support devices or systems. Holtek reserves the right to alter its products without prior notification. For the most up-to-date information, please visit our web site at http://www.holtek.com.tw.