Enhanced I/O Type MCU with OPA

HT48R064G/065G/066G/0662G

Revision: 1.10 Date: October 23, 2012

www.holtek.com

Table of Contents

Features

CPU Features

- Operating voltage:
  f_{SYS}= 4MHz: 2.2V~5.5V
  f_{SYS}= 8MHz: 3.3V~5.5V
  f_{SYS}= 12MHz: 4.5V~5.5V
- Up to 0.33μs instruction cycle with 12MHz system clock at V_{DD}= 5V
- Oscillator types:
  External high freuency Crystal -- HXT
  External RC -- ERC
  Internal RC -- HIRC
  External low frequency crystal -- LXT
- Four operational modes: Normal, Slow, Idle, Sleep
- Fully integrated internal 4MHz, 8MHz and 12MHz oscillator requires no external components
- Watchdog Timer function
- LIRC oscillator function for watchdog timer
- All instructions executed in one or two instruction cycles
- Table read instructions
- 63 powerful instructions
- Up to 6-level subroutine nesting
- Bit manipulation instruction
- Low voltage reset function
- Wide range of available package types

Peripheral Features

- Program Memory: 1K x 14 ~ 4K x 15
- Data Memory: 64 x 8 ~ 224 x 8
- Up to 42 bidirectional I/O lines
- Up to 2 channel 8-bit PWM
- Software controlled 4-SCOM lines LCD driver with 1/2 bias
- External interrupt input shared with an I/O line
- Up to two 8-bit programmable Timer/Event Counter with overflow interrupt and prescaler
- Time-Base function
- Programmable Frequency Divider - PFD
- Two integrated operational amplifiers with interrupt function - one with programmable gain control
- Single comparator with interrupt and low power consumption

General Description

The Enhanced I/O MCU devices are a series of 8-bit high performance, RISC architecture microcontroller specifically designed for a wide range of applications. The usual Holtek microcontroller features of low power consumption, I/O flexibility, timer functions, oscillator options, power down and wake-up functions, watchdog timer and low voltage reset, combine to provide devices with a huge range of functional options while still maintaining a high level of cost effectiveness. The fully integrated system oscillator HIRC, which requires no external components and which has three frequency selections, opens up a huge range of new application possibilities for the device, some of which may include industrial control, consumer products, household appliances subsystem controllers, etc.

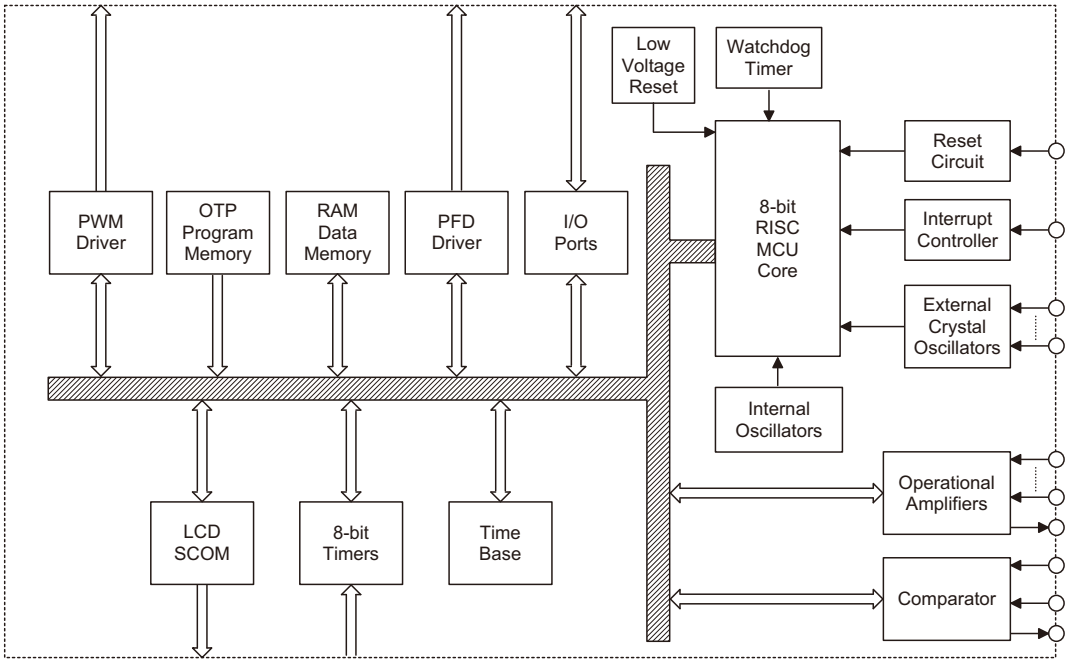
Selection Table

Part No.	Program Memory	Data Memory	I/O	8-bit Timer	HIRC (MHz)	RTC (LXT)	LCD SCOM	PWM	OPA	Comp.	PFD	Stack	Package
HT48R064G	1K×14	64×8	18	1	4/8/12	√	—	—	2	1	√	4	16DIP/NSOP 20DIP/SOP/SSOP
HT48R065G	2K×15	96×8	22	1	4/8/12	√	4	—	2	1	√	4	16DIP/NSOP 20DIP/SOP/SSOP 24SKDIP/SOP/SSOP
HT48R066G	4K×15	128×8	26	2	4/8/12	√	4	8-bit×1	2	1	√	4	20DIP/SOP/SSOP 24/28SKDIP/SOP/SSOP
HT48R0662G	4K×15	224×8	42	2	4/8/12	√(*)	4	8-bit×2	2	1	√	6	24/28SKDIP/SOP/SSOP 44LQFP

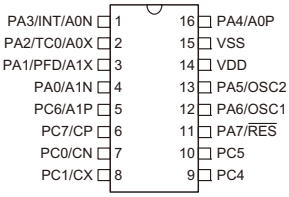
Note: "*" the oscillator is connected to the OSC3/OSC4 pins with TinyPower™ design.

Block Diagram

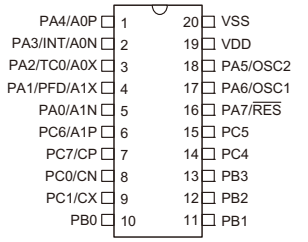
The following block diagram illustrates the main functional blocks.



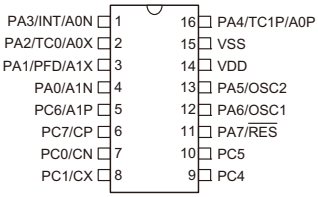
Pin Assignment



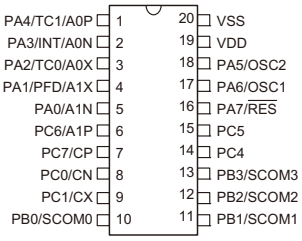
HT48R064G
16 DIP-A/NSOP-A



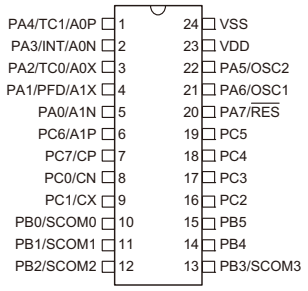
HT48R064G
20 DIP-A/SOP-A/SSOP-A



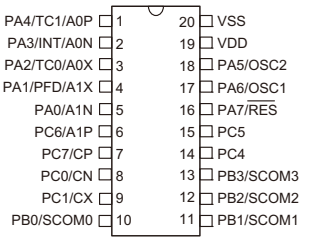
HT48R065G
16 DIP-A/NSOP-A



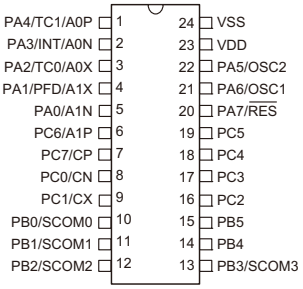
HT46R065G
20 DIP-A/SOP-A/SSOP-A



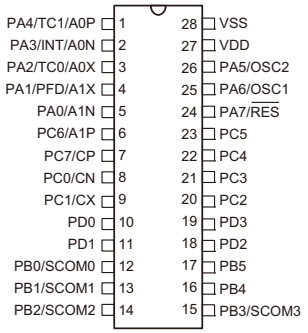
HT46R065G
24 SKDIP-A/SOP-A/SSOP-A



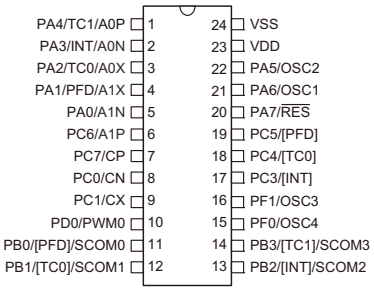
HT48R066G
20 DIP-A/SOP-A/SSOP-A



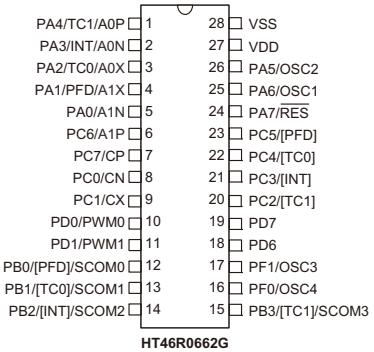
HT48R066G
24 SKDIP-A/SOP-A/SSOP-A



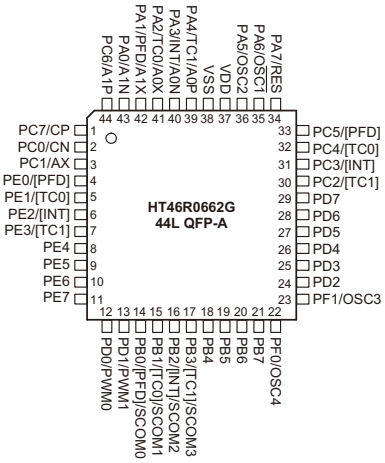
HT48R066G
28 SKDIP-A/SOP-A/SSOP-A



HT46R0662G
24 SKDIP-A/SOP-A/SSOP-A



HT46R0662G
28 SKDIP-A/SOP-A/SSOP-A



HT46R0662G
44L QFP-A

Note: Bracketed pin names indicate non-default pinout remapping locations.

Pin Description

The function of each pin is listed in the following tables, however the details behind how each pin is configured is contained in the other individual peripheral function sections.

HT48R064G

Pin Name	Function	OPT	I/T	O/T	Description
PA0/A1N	PA0	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	A1N	COPA3C	OPAI	—	OPA1 inverting input pin
PA1/PFD/A1X	PA1	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	PFD	CTRL0	—	CMOS	PFD output
	A1X	COPA3C	—	OPAO	OPA1 output pin
PA2/TC0/A0X	PA2	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TC0	TMR0C	ST	—	External Timer 0 clock input
	A0X	COPA3C	—	OPAO	OPA0 output pin
PA3/INT/A0N	PA3	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	INT	INTC0 CTRL1	ST	—	External interrupt input
	A0X	COPA3C	OPAI	—	OPA0 inverting input pin
PA4/A0P	PA4	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	A0P	COPA3C	OPAI	—	OPA0 non-inverting input pin
PA5/OSC2	PA5	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	OSC2	CO	—	OSC	Oscillator pin
PA6/OSC1	PA6	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	OSC1	CO	OSC	—	Oscillator pin
PA7/$\overline{RES}$	PA7	PAWK	ST	NMOS	General purpose I/O. Register enabled wake-up.
	$\overline{RES}$	CO	ST	—	Reset input
PB0~PB3	PBn	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
PC0/CN	PC0	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	CN	COPA3C	CMPI	—	Comparator inverting input pin
PC1/CX	PC1	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	CX	COPA2C	—	CMPO	Comparator output pin
PC4, PC5	PCn	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
PC6/A1P	PC6	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	A1P	COPA3C	OPAI	—	OPA1 non-inverting input pin

Pin Name	Function	OPT	I/T	O/T	Description
PC7/CP	PC7	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	CP	COPA3C	CMPI	—	Comparator non-inverting input pin
VDD	VDD	—	PWR	—	Power supply
VSS	VSS	—	PWR	—	Ground

Note: OPT: Optional by configuration option (CO) or register option
I/T: Input type
O/T: Output type
PWR: Power
CO: Configuration option
ST: Schmitt Trigger input
CMOS: CMOS output
NMOS: NMOS output
OSC: Oscillator pin
OPAI: Operational Amplifier input
OPAO: Operational Amplifier output
CMPI: Comparator input
CMPO: Comparator output

HT48R065G

Pin Name	Function	OPT	I/T	O/T	Description
PA0/A1N	PA0	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	A1N	COPA3C	OPAI	—	OPA1 inverting input pin
PA1/PFD/A1X	PA1	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	PFD	CTRL0	—	CMOS	PFD output
	A1X	COPA3C	—	OPAO	OPA1 output pin
PA2/TC0/A0X	PA2	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TC0	TMR0C	ST	—	External Timer 0 clock input
	A0X	COPA3C	—	OPAO	OPA0 output pin
PA3/INT/A0N	PA3	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	INT	INTC0 CTRL1	ST	—	External interrupt input
	A0X	COPA3C	OPAI	—	OPA0 inverting input pin
PA4/TC1/A0P	PA4	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TC1	TMR1C	ST	—	External Timer 1 clock input
	A0P	COPA3C	OPAI	—	OPA0 non-inverting input pin
PA5/OSC2	PA5	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	OSC2	CO	—	OSC	Oscillator pin

Pin Name	Function	OPT	I/T	O/T	Description
PA6/OSC1	PA6	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	OSC1	CO	OSC	—	Oscillator pin
PA7/$\overline{RES}$	PA7	PAWK	ST	NMOS	General purpose I/O. Register enabled wake-up.
	$\overline{RES}$	CO	ST	—	Reset input
PB0/SCOM0~ PB3/SCOM3	PBn	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	SCOMn	SCOMC	—	SCOM	Software controlled 1/2 bias LCD COM
PB4~PB5	PBn	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
PC0/CN	PC0	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	CN	COPA3C	CMPI	—	Comparator inverting input pin
PC1/CX	PC1	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	CX	COPA2C	—	CMPO	Comparator output pin
PC2~PC5	PCn	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
PC6/A1P	PC6	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	A1P	COPA3C	OPAI	—	OPA1 non-inverting input pin
PC7/CP	PC7	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	CP	COPA3C	CMPI	—	Comparator non-inverting input pin
VDD	VDD	—	PWR	—	Power supply
VSS	VSS	—	PWR	—	Ground

Note: OPT: Optional by configuration option (CO) or register option
I/T: Input type; O/T: Output type; PWR: Power; CO: Configuration option
ST: Schmitt Trigger input; AN: analog input
CMOS: CMOS output; NMOS: NMOS output
OSC: Oscillator pin; SCOM: Software controlled LCD COM
CMPI: Comparator input; CMPO: Comparator output
OPAI: Operational Amplifier input; OPAO: Operational Amplifier output

HT48R066G

Pin Name	Function	OPT	I/T	O/T	Description
PA0/A1N	PA0	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	A1N	COPA3C	OPAI	—	OPA1 inverting input pin
PA1/PFD/A1X	PA1	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	PFD	CTRL0	—	CMOS	PFD output
	A1X	COPA3C	—	OPAO	OPA1 output pin
PA2/TC0/A0X	PA2	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TC0	TMR0C	ST	—	External Timer 0 clock input
	A0X	COPA3C	—	OPAO	OPA0 output pin

Pin Name	Function	OPT	I/T	O/T	Description
PA3/INT/A0N	PA3	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	INT	INTC0 CTRL1	ST	—	External interrupt input
	A0X	COPA3C	OPAI	—	OPA0 inverting input pin
PA4/TC1/A0P	PA4	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TC1	TMR1C	ST	—	External Timer 1 clock input
	A0P	COPA3C	OPAI	—	OPA0 non-inverting input pin
PA5/OSC2	PA5	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	OSC2	CO	—	OSC	Oscillator pin
PA6/OSC1	PA6	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	OSC1	CO	OSC	—	Oscillator pin
PA7/$\overline{\text{RES}}$	PA7	PAWK	ST	NMOS	General purpose I/O. Register enabled wake-up.
	$\overline{\text{RES}}$	CO	ST	—	Reset input
PB0/[PFD]/SCOM0	PB0	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	PFD	CTRL0	—	CMOS	PFD output
	SCOM0	SCOMC	—	SCOM	Software controlled 1/2 bias LCD COM
PB1/[TC0]/SCOM1	PB1	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TC0	TMR0C	ST	—	External Timer 0 clock input
	SCOM1	SCOMC	—	SCOM	Software controlled 1/2 bias LCD COM
PB2/[INT]/SCOM2	PB2	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	INT	INTC0 CTRL1	ST	—	External interrupt input
	SCOM2	SCOMC	—	SCOM	Software controlled 1/2 bias LCD COM
PB3/[TC1]/SCOM3	PB3	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TC1	TMR1C	ST	—	External Timer 1 clock input
	SCOM3	SCOMC	—	SCOM	Software controlled 1/2 bias LCD COM
PB4~PB5	PBn	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
PC0/CN	PC0	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	CN	COPA3C	CMPI	—	Comparator inverting input pin
PC1/CX	PC1	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	CX	COPA2C	—	CMPO	Comparator output pin
PC2~PC5	PCn	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
PC6/A1P	PC6	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	A1P	COPA3C	OPAI	—	OPA1 non-inverting input pin

Pin Name	Function	OPT	I/T	O/T	Description
PC7/CP	PC7	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	CP	COPA3C	CMPI	—	Comparator non-inverting input pin
PD0~PD3	PDn	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
VDD	VDD	—	PWR	—	Power supply
VSS	VSS	—	PWR	—	Ground

Note: OPT: Optional by configuration option (CO) or register option
I/T: Input type
O/T: Output type
PWR: Power
CO: Configuration option
ST: Schmitt Trigger input
CMOS: CMOS output
NMOS: NMOS output
HXT: High frequency crystal oscillator pin
LXT: Low frequency crystal oscillator pin
SCOM: Software controlled LCD COM
CMPI: Comparator input
CMPO: Comparator output
OPAI: Operational Amplifier input
OPAO: Operational Amplifier output

HT48R0662G

Pin Name	Function	OPT	I/T	O/T	Description
PA0/A1N	PA0	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	A1N	COPA3C	OPAI	—	OPA1 inverting input pin
PA1/PFD/A1X	PA1	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	PFD	CTRL0	—	CMOS	PFD output
	A1X	COPA3C	—	OPAO	OPA1 output pin
PA2/TC0/A0X	PA2	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TC0	TMR0C	ST	—	External Timer 0 clock input
	A0X	COPA3C	—	OPAO	OPA0 output pin
PA3/INT/A0N	PA3	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	INT	INTC0 CTRL1	ST	—	External interrupt input
	A0X	COPA3C	OPAI	—	OPA0 inverting input pin
PA4/TC1/A0P	PA4	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TC1	TMR1C	ST	—	External Timer 1 clock input
	A0P	COPA3C	OPAI	—	OPA0 non-inverting input pin

Pin Name	Function	OPT	I/T	O/T	Description
PA5/OSC2	PA5	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	OSC2	CO	—	OSC	Oscillator pin
PA6/OSC1	PA6	PAPU PAWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	OSC1	CO	OSC	—	Oscillator pin
PA7/$\overline{\text{RES}}$	PA7	PAWK	ST	NMOS	General purpose I/O. Register enabled wake-up.
	$\overline{\text{RES}}$	CO	ST	—	Reset input
PB0/[PFD]/SCOM0	PB0	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	PFD	CTRL0	—	CMOS	PFD output
	SCOM0	SCOMC	—	SCOM	Software controlled 1/2 bias LCD COM
PB1/[TC0]/SCOM1	PB1	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TC0	TMR0C	ST	—	External Timer 0 clock input
	SCOM1	SCOMC	—	SCOM	Software controlled 1/2 bias LCD COM
PB2/[INT]/SCOM2	PB2	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	INT	INTC0 CTRL1	ST	—	External interrupt input
	SCOM2	SCOMC	—	SCOM	Software controlled 1/2 bias LCD COM
PB3/[TC1]/SCOM3	PB3	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TC1	TMR1C	ST	—	External Timer 1 clock input
	SCOM3	SCOMC	—	SCOM	Software controlled 1/2 bias LCD COM
PB4~PB7	PBn	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
PC0/CN	PC0	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	CN	COPA3C	CMPI	—	Comparator inverting input pin
PC1/CX	PC1	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	CX	COPA2C	—	CMPO	Comparator output pin
PC2/[TC1]	PC2	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TC1	TMR1C	ST	—	External Timer 1 clock input
PC3/[INT]	PC3	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	INT	INTC0 CTRL1	ST	—	External interrupt input
PC4/[TC0]	PC4	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TC0	TMR0C	ST	—	External Timer 0 clock input
PC5/[PFD]	PC5	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	PFD	CTRL0	—	CMOS	PFD output

Pin Name	Function	OPT	I/T	O/T	Description
PC6/A1P	PC6	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	A1P	COPA3C	OPAI	—	OPA1 non-inverting input pin
PC7/CP	PC7	PCPU PCWK	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	CP	COPA3C	CMPI	—	Comparator non-inverting input pin
PD0/PWM0	PD0	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	PWM0	CTRL0	—	CMOS	PWM 0 output
PD1/PWM1	PD1	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	PWM1	CTRL0	—	CMOS	PWM 1 output
PD2~PD7	PDn	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
PE0/[PFD]	PE0	PEPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	PFD	CTRL0	—	CMOS	PFD output
PE1/[TC0]	PE1	PEPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TC0	TMR0C	—	CMOS	External Timer 0 clock input
PE2/[INT]	PE2	PEPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	INT	INTC0 CTRL1	—	CMOS	External interrupt input
PE3/[TC1]	PE3	PEPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TC1	TMR1C	—	CMOS	External Timer 1 clock input
PE4~PE7	PEn	PEPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
PF0/OSC4	PF0	PFPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	OSC4	CO	—	OSC	LXT Oscillator pin
PF1/OSC3	PF1	PFPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	OSC3	CO	LXT	—	LXT Oscillator pin
VDD	VDD	—	PWR	—	Power supply
VSS	VSS	—	PWR	—	Ground

Note: OPT: Optional by configuration option (CO) or register option
I/T: Input type
O/T: Output type
PWR: Power
CO: Configuration option
ST: Schmitt Trigger input
CMOS: CMOS output
NMOS: NMOS output
HXT: High frequency crystal oscillator pin
LXT: Low frequency crystal oscillator pin
SCOM: Software controlled LCD COM
CMPI: Comparator input
CMPO: Comparator output
OPAI: Operational Amplifier input
OPAO: Operational Amplifier output

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
Storage Temperature ..... –50°C to 125°C
Operating Temperature ..... –40°C to 85°C
I_{OL} Total 100mA
I_{OH} Total ..... –100mA
Total Power Dissipation 500mW

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
			f_{SYS}=12MHz	4.5	—	5.5	V
I_{DD1}	Operating Current (HXT, HIRC, ERC)	3V	No load, f_{SYS}=4MHz	—	0.8	1.2	mA
		5V		—	1.5	2.25	mA
I_{DD2}	Operating Current (HXT, HIRC, ERC)	3V	No load, f_{SYS}=8MHz	—	1.4	2.1	mA
		5V		—	2.8	4.2	mA
I_{DD3}	Operating Current (HXT, HIRC, ERC)	5V	No load, f_{SYS}=12MHz	—	4	6	mA
I_{DD4}	Operating Current (HIRC + LXT, Slow Mode)	3V	No load, f_{SYS}=32768Hz (LXT on OSC1/OSC2, LVR disabled, LXTLP=1)	—	5	10	μA
		5V		—	12	24	μA
		3V	No load, f_{SYS}=32768Hz (LXT on XT1/XT2, LVR disabled, LXTLP=1)	—	5	10	μA
		5V		—	10	20	μA
I_{STB1}	Standby Current (LIRC On, LXT Off)	3V	No load, system HALT	—	—	5	μA
		5V		—	—	10	μA
I_{STB2}	Standby Current (LIRC Off, LXT Off)	3V	No load, system HALT	—	—	1	μA
		5V		—	—	2	μA
I_{STB3}	Standby Current (LIRC Off, LXT On, LXTLP=1)	3V	No load, system HALT (LXT on OSC1/OSC2)	—	—	5	μA
		5V		—	—	10	μA
		3V	No load, system HALT (LXT on XT1/XT2)	—	—	3	μA
		5V		—	—	5	μA
V_{IL1}	Input Low Voltage for I/O, TCn and INT	—	—	0	—	0.3V_{DD}	V

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{IH1}	Input High Voltage for I/O, TCn and 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_{LVR1}	Low Voltage Reset 1	—	VLVR = 4.2V	3.98	4.2	4.42	V
V_{LVR2}	Low Voltage Reset 2	—	VLVR = 3.15V	2.98	3.15	3.32	V
V_{LVR3}	Low Voltage Reset 3	—	VLVR = 2.1V	1.98	2.1	2.22	V
I_{OL1}	I/O Port Sink Current (PA, PB, PC)	3V	$V_{OL}=0.1V_{DD}$	4	8	—	mA
		5V		10	20	—	mA
I_{OH}	I/O Port Source Current	3V	$V_{OH}=0.9V_{DD}$	–2	–4	—	mA
		5V		–5	–10	—	mA
I_{OL2}	PA7 Sink Current	5V	$V_{OL}=0.1V_{DD}$	2	3	—	mA
R_{PH}	Pull-high Resistance	3V	—	20	60	100	kΩ
		5V	—	10	30	50	kΩ
I_{SCOM}	SCOM Operating Current	5V	SCOMC, ISEL[1:0]=00	17.5	25.0	32.5	μA
			SCOMC, ISEL[1:0]=01	35	50	65	μA
			SCOMC, ISEL[1:0]=10	70	100	130	μA
			SCOMC, ISEL[1:0]=11	140	200	260	μA
V_{SCOM}	V_{DD}/2 Voltage for LCD COM	5V	No load	0.475	0.500	0.525	V_{DD}
V_{OPBIAS}	OPA/Comparator bias voltage Deviation (Bias=0.7/0.5/0.1V_{DD} Selected by A1PS[2:0], A0PS[2:0], CPS[2:0] Bits)	3V	No load	0.665	0.700	0.735	V_{DD}
				0.475	0.500	0.525	V_{DD}
				0.995	0.100	0.105	V_{DD}
GOP	OPA1 Gain Deviation (Software Gain Controlled by A1G[2:0]	3V	No load	–5	—	+5	%

Note: The standby current (I_{STB1}~I_{STB3}) and I_{DD4} are measured with all I/O pins in input mode and tied to V_{DD}.

A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SYS}	System Clock	—	2.2V~5.5V	32	—	4000	kHz
			3.0V~5.5V	32	—	8000	kHz
			4.5V~5.5V	32	—	12000	kHz
f_{HIRC}	System Clock (HIRC)	3V/5V	Ta=25°C	–2%	4	+2%	MHz
		3V/5V	Ta=25°C	–2%	8	+2%	MHz
		5V	Ta=25°C	–2%	12	+2%	MHz
		3V/5V	Ta=0~70°C	–5%	4	+5%	MHz
		3V/5V	Ta=0~70°C	–5%	8	+5%	MHz
		5V	Ta=0~70°C	–5%	12	+5%	MHz
		2.2V~3.6V	Ta=0~70°C	–8%	4	+8%	MHz
		3.0V~5.5V	Ta=0~70°C	–8%	4	+8%	MHz
		3.0V~5.5V	Ta=0~70°C	–8%	8	+8%	MHz
		4.5V~5.5V	Ta=0~70°C	–8%	12	+8%	MHz
		2.2V~3.6V	Ta= –40°C~85°C	–12%	4	+12%	MHz
		3.0V~5.5V	Ta= –40°C~85°C	–12%	4	+12%	MHz
		3.0V~5.5V	Ta= –40°C~85°C	–12%	8	+12%	MHz
		4.5V~5.5V	Ta= –40°C~85°C	–12%	12	+12%	MHz
f_{ERC}	System Clock (ERC)	5V	Ta=25°C, R=120kΩ *	–2%	4	+2%	MHz
		5V	Ta=0~70°C, R=120kΩ *	–5%	4	+5%	MHz
		5V	Ta= –40°C~85°C, R=120kΩ *	–7%	4	+7%	MHz
		2.2V~5.5V	Ta= –40°C~85°C, R=120kΩ *	–11%	4	+11%	MHz
f_{LXT}	System Clock (LXT)	—	—	—	32768	—	Hz
f_{TIMER}	Timer Input Frequency (TCn)	—	2.2V~5.5V	0	—	4000	kHz
			3.0V~5.5V	0	—	8000	kHz
			4.5V~5.5V	0	—	12000	kHz
f_{LIRC}	LIRC Oscillator	3V	—	5	10	15	kHz
		5V	—	6.5	13	19.5	kHz
t_{RES}	External Reset Low Pulse Width	—	—	1	—	—	μs

Ta=25°C

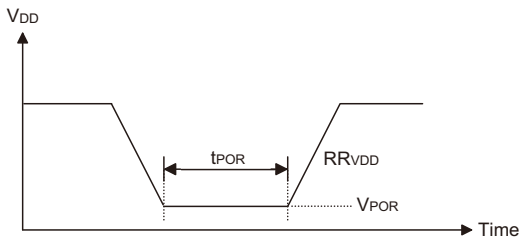
Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
t_{SST}	System Start-up time Period	—	For HXT/LXT	—	128	—	t_{SYS}
			For ERC/IRC	—	2	—	t_{SYS}
t_{INT}	Interrupt Pulse Width	—	—	1	—	—	μs
t_{LVR}	Low Voltage Width to Reset	—	—	0.25	1	2	ms
t_{RSTD}	Reset Delay Time	—	—	—	50	—	ms

Note: 1. $t_{SYS}=1/f_{SYS}$
2. *For f_{ERC}, as the resistor tolerance will influence the frequency a precision resistor is recommended.
3. To maintain the accuracy of the internal HIRC oscillator frequency, a 0.1μF decoupling capacitor should be connected between VDD and VSS and located as close to the device as possible.

Power-on Reset Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{POR}	VDD Start Voltage to Ensure Power-on Reset	—	—	—	—	100	mV
RR_{VDD}	VDD Rise Rate to Ensure Power-on Reset	—	—	0.035	—	—	V/ms
t_{POR}	Minimum Time for VDD to remain at V_{POR} to Ensure Power-on Reset	—	—	1	—	—	ms



Comparator Amplifier Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{COMP}	Comparator Operating Current	3V	CPCS[1:0]=00B	—	200	300	μA
			CPCS[1:0]=01B	—	5	10	μA
			CPCS[1:0]=10B	—	1	2	μA
V_{OS}	Comparator Input Offset Voltage	3V	—	–10	—	10	mV
V_{CM}	Comparator Common Mode Voltage Range	—	—	0	—	V_{DD}-1.4V	V
t_{PD}	Comparator Response Time (With 10mV overdrive)	3V	CPCS[1:0]=00B	—	—	2	μs
		3V	CPCS[1:0]=01B	—	—	60	μs
		—	CPCS[1:0]=10B	—	—	400	μs

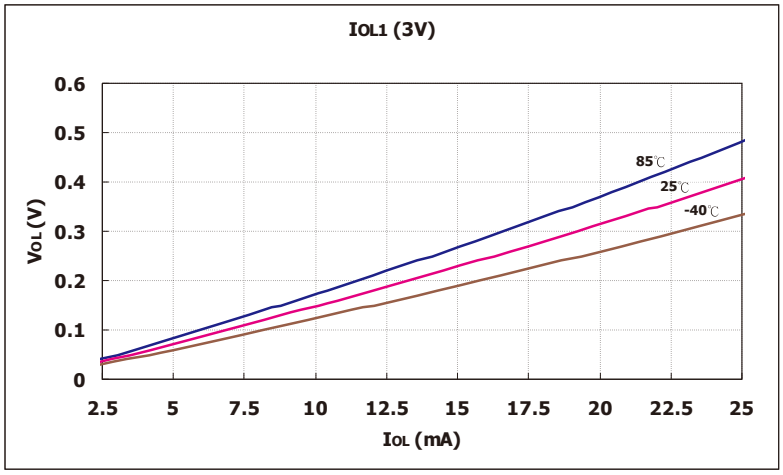
Operational Amplifier Characteristics

Ta=25°C

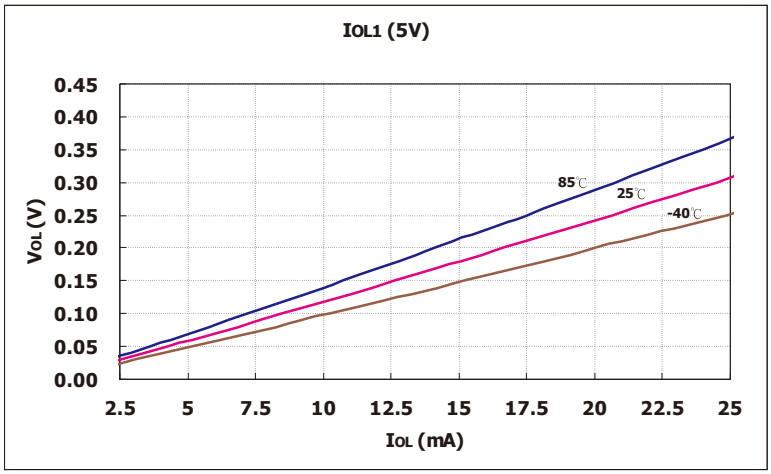
Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
	Power Down Current	3V	—	—	—	0.1	μA
V_{OPOS1}	Input Offset Voltage	3V	Without calibration, OPOF[3:0]=1000B	–15	—	15	mV
V_{OPOS2}	Input Offset Voltage	3V	By Calibration	–4	—	4	mV
V_{CM}	Common Mode Voltage Range	—	—	V_{SS}	—	V_{DD}-1.4V	V
PSRR	Power Supply Rejection Ratio	3V	—	60	80	—	dB
CMRR	Common Mode Rejection Ratio	3V	V_{CM}=0~V_{DD}–1.4V	60	80	—	dB
SR	Slew Rate +, Slew Rate -	3V	No load	1.8	2.5	—	V/μs
GBW	Gain Band Width	3V	R_L=1M, C_L=100p	500	—	—	kHz

Characteristic Curve

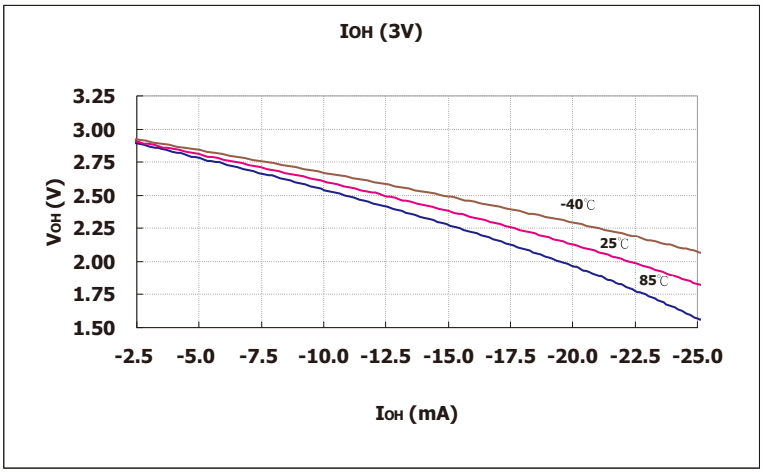
V_{OL} vs. I_{OL1} Over Temperature (V_{DD}=3.0V)



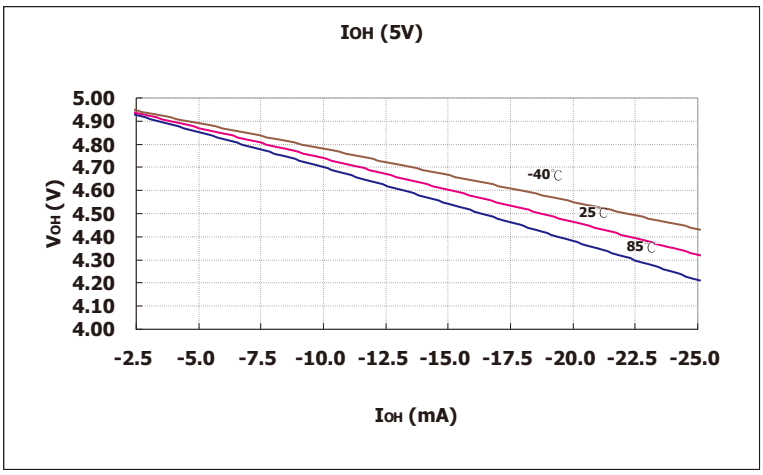
V_{OL} vs. I_{OL1} Over Temperature (V_{DD}=5.0V)



V_{OH} vs. I_{OH} Over Temperature (V_{DD}=3.0V)



V_{OH} vs. I_{OH} Over Temperature (V_{DD}=5.0V)

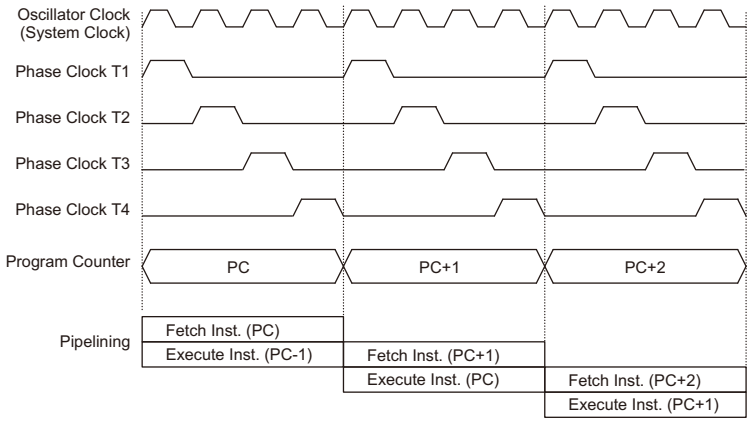


System Architecture

A key factor in the high-performance features of the Holtek range of microcontrollers is attributed to the internal system architecture. The range of devices take advantage of the usual features found within RISC microcontrollers providing increased speed of operation and enhanced performance. The pipelining scheme is implemented in such a way that instruction fetching and instruction execution are overlapped, hence instructions are effectively executed in one cycle, with the exception of branch or call instructions. An 8-bit wide ALU is used in practically all operations of the instruction set. It carries out arithmetic operations, logic operations, rotation, increment, decrement, branch decisions, etc. The internal data path is simplified by moving data through the Accumulator and the ALU. Certain internal registers are implemented in the Data Memory and can be directly or indirectly addressed. The simple addressing methods of these registers along with additional architectural features ensure that a minimum of external components is required to provide a functional I/O control system with maximum reliability and flexibility.

Clocking and Pipelining

The main system clock, derived from either a Crystal/Resonator or RC oscillator is subdivided into four internally generated non-overlapping clocks, T1~T4. The Program Counter is incremented at the beginning of the T1 clock during which time a new instruction is fetched. The remaining T2~T4 clocks carry out the decoding and execution functions. In this way, one T1~T4 clock cycle forms one instruction cycle. Although the fetching and execution of instructions takes place in consecutive instruction cycles, the pipelining structure of the microcontroller ensures that instructions are effectively executed in one instruction cycle. The exception to this are instructions where the contents of the Program Counter are changed, such as subroutine calls or jumps, in which case the instruction will take one more instruction cycle to execute.

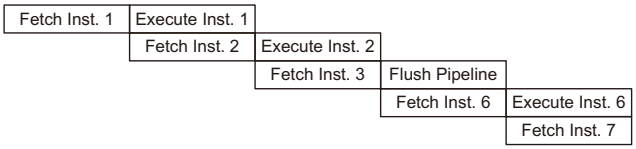


System Clocking and Pipelining

For instructions involving branches, such as jump or call instructions, two instruction cycles are required to complete instruction execution. An extra cycle is required as the program takes one cycle to first obtain the actual jump or call address and then another cycle to actually execute the branch. The requirement for this extra cycle should be taken into account by programmers in timing sensitive applications.

```
1          MOV A,[12H]
2          CALL DELAY
3          CPL [12H]
4          :
5          :
6          NOP

   DELAY:
```



Instruction Fetching

Program Counter

During program execution, the Program Counter is used to keep track of the address of the next instruction to be executed. It is automatically incremented by one each time an instruction is executed except for instructions, such as "JMP" or "CALL" that demand a jump to a non-consecutive Program Memory address. Note that the Program Counter width varies with the Program Memory capacity depending upon which device is selected. However, it must be noted that only the lower 8 bits, known as the Program Counter Low Register, are directly addressable by user.

When executing instructions requiring jumps to non-consecutive addresses such as a jump instruction, a subroutine call, interrupt or reset, etc., the microcontroller manages program control by loading the required address into the Program Counter. For conditional skip instructions, once the condition has been met, the next instruction, which has already been fetched during the present instruction execution, is discarded and a dummy cycle takes its place while the correct instruction is obtained.

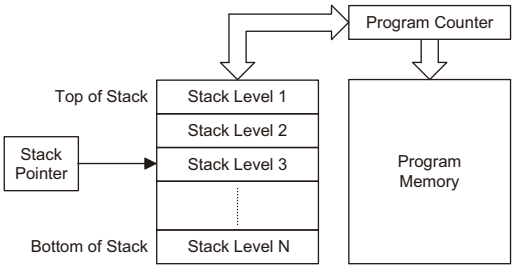
Device	Program Counter	
	Program Counter High Byte	PCL Register
HT48R064G	PC9, PC8	PCL7~PCL0
HT48R065G	PC10~PC8	
HT48R066G HT48R0662G	PC11~PC8	

The lower byte of the Program Counter, known as the Program Counter Low register or PCL, is available for program control and is a readable and writeable register. By transferring data directly into this register, a short program jump can be executed directly, however, as only this low byte is available for manipulation, the jumps are limited to the present page of memory, that is 256 locations. When such program jumps are executed it should also be noted that a dummy cycle will be inserted.

The lower byte of the Program Counter is fully accessible under program control. Manipulating the PCL might cause program branching, so an extra cycle is needed to pre-fetch. Further information on the PCL register can be found in the Special Function Register section.

Stack

This is a special part of the memory which is used to save the contents of the Program Counter only. The stack is neither part of the Data or Program Memory space, and is neither readable nor writeable. The activated level is indexed by the Stack Pointer, SP, and is neither readable nor writeable. At a subroutine call or interrupt acknowledge signal, the contents of the Program Counter are pushed onto the stack. At the end of a subroutine or an interrupt routine, signaled by a return instruction, RET or RETI, the Program Counter is restored to its previous value from the stack. After a device reset, the Stack Pointer will point to the top of the stack.



Device	Stack Levels
HT48R064G/065G/066G	4
HT48R0662G	6

If the stack is full and an enabled interrupt takes place, the interrupt request flag will be recorded but the acknowledge signal will be inhibited. When the Stack Pointer is decremented, by RET or RETI, the interrupt will be serviced. This feature prevents stack overflow allowing the programmer to use the structure more easily. However, when the stack is full, a CALL subroutine instruction can still be executed which will result in a stack overflow. Precautions should be taken to avoid such cases which might cause unpredictable program branching.

Arithmetic and Logic Unit – ALU

The arithmetic-logic unit or ALU is a critical area of the microcontroller that carries out arithmetic and logic operations of the instruction set. Connected to the main microcontroller data bus, the ALU receives related instruction codes and performs the required arithmetic or logical operations after which the result will be placed in the specified register. As these ALU calculation or operations may result in carry, borrow or other status changes, the status register will be correspondingly updated to reflect these changes. The ALU supports the following functions:

- Arithmetic operations: ADD, ADDM, ADC, ADCM, SUB, SUBM, SBC, SBCM, DAA
- Logic operations: AND, OR, XOR, ANDM, ORM, XORM, CPL, CPLA
- Rotation RRA, RR, RRCA, RRC, RLA, RL, RLCA, RLC
- Increment and Decrement INCA, INC, DECA, DEC
- Branch decision, JMP, SZ, SZA, SNZ, SIZ, SDZ, SIZA, SDZA, CALL, RET, RETI

Program Memory

The Program Memory is the location where the user code or program is stored. The device is supplied with One-Time Programmable, OTP, memory where users can program their application code into the device. By using the appropriate programming tools, OTP devices offer users the flexibility to freely develop their applications which may be useful during debug or for products requiring frequent upgrades or program changes.

Structure

The Program Memory has a capacity of 1K×14 to 4K×15. The Program Memory is addressed by the Program Counter and also contains data, table information and interrupt entries. Table data, which can be setup in any location within the Program Memory, is addressed by separate table pointer registers.

Device	Capacity
HT48R064G	1K×14
HT48R065G	2K×15
HT48R066G/0662G	4K×15

Special Vectors

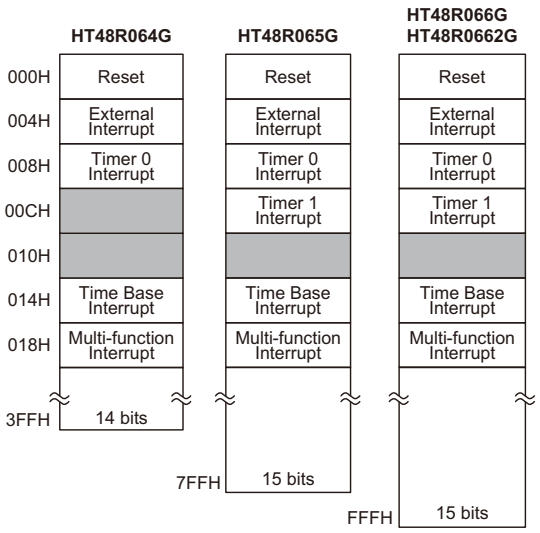
Within the Program Memory, certain locations are reserved for special usage such as reset and interrupts.

Reset Vector

This vector is reserved for use by the device reset for program initialisation. After a device reset is initiated, the program will jump to this location and begin execution.

External Interrupt Vector

This vector is used by the external interrupt. If the external interrupt pin on the device receives an edge transition, the program will jump to this location and begin execution if the external interrupt is enabled and the stack is not full. The external interrupt active edge transition type, whether high to low, low to high or both is specified in the CTRL1 register.



Program Memory Structure

Timer/Event 0/1 Counter Interrupt Vector

This internal vector is used by the Timer/Event Counters. If a Timer/Event Counter overflow occurs, the program will jump to its respective location and begin execution if the associated Timer/Event Counter interrupt is enabled and the stack is not full.

Time Base Interrupt Vector

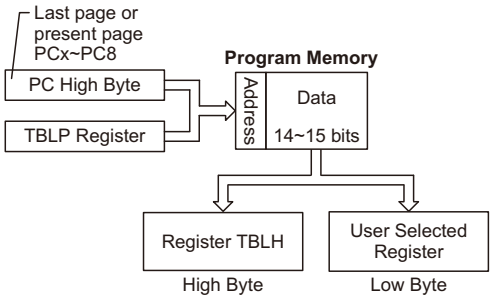
This vector is used by the OPA0, OPA1 and Comparator. When either an OPA or Comparator, dependent upon which one is selected, requires interrupt servicing, the program will jump to this location and begin execution if the output interrupt is enabled and the stack is not full.

Look-up Table

Any location within the Program Memory can be defined as a look-up table where programmers can store fixed data. To use the look-up table, the table pointer must first be setup by placing the lower order address of the look up data to be retrieved in the table pointer register, TBLP. This register defines the lower 8-bit address of the look-up table.

After setting up the table pointer, the table data can be retrieved from the current Program Memory page or last Program Memory page using the "TABRDC[m]" or "TABRDL [m]" instructions, respectively. When these instructions are executed, the lower order table byte from the Program Memory will be transferred to the user defined Data Memory register [m] as specified in the instruction. The higher order table data byte from the Program Memory will be transferred to the TBLH special register. Any unused bits in this transferred higher order byte will be read as "0".

The following diagram illustrates the addressing/data flow of the look-up table:



Instruction	Table Location Bits											
	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TABRDC [m]	PC11	PC10	PC9	PC8	@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: PC11~PC8: Current Program Counter bits
@7~@0: Table Pointer TBLP bits
For the HT48R064G, the Table address location is 10 bits, i.e. from b9~b0
For the HT48R065G, the Table address location is 11 bits, i.e. from b10~b0
For the HT48R066G/HT48R0662G, the Table address location is 12 bits, i.e. from b11~b0

Table Program Example

The accompanying example shows how the table pointer and table data is defined and retrieved from the device. This example uses raw table data located in the last page which is stored there using the ORG statement. The value at this ORG statement is "300H" which refers to the start address of the last page within the 1K Program Memory of the device. The table pointer is setup here to have an initial value of "06H". This will ensure that the first data read from the data table will be at the Program Memory address "306H" or 6 locations after the start of the last page. Note that the value for the table pointer is referenced to the first address of the present page if the "TABRDC [m]" instruction is being used. The high byte of the table data which in this case is equal to zero will be transferred to the TBLH register automatically when the "TABRDL [m]" instruction is executed.

Because the TBLH register is a read-only register and cannot be restored, care should be taken to ensure its protection if both the main routine and Interrupt Service Routine use the table read instructions. If using the table read instructions, the Interrupt Service Routines may change the value of TBLH and subsequently cause errors if used again by the main routine. As a rule it is recommended that simultaneous use of the table read instructions should be avoided. However, in situations where simultaneous use cannot be avoided, the interrupts should be disabled prior to the execution of any main routine table-read instructions. Note that all table related instructions require two instruction cycles to complete their operation.

Table Read Program Example

```
tempreg1  db ?     ; temporary register #1
tempreg2  db ?     ; temporary register #2
  :
  :

mov   a,06h        ; initialise table pointer - note that this address is referenced

mov   tblp,a       ; to the last page or present page
  :
  :

tabrdl  tempreg1   ; transfers value in table referenced by table pointer
                   ; to tempreg1
                   ; data at prog. memory address "306H" transferred to
                   ; tempreg1 and TBLH

dec   tblp         ; reduce value of table pointer by one

Tabrdl  tempreg2   ; transfers value in table referenced by table pointer
                   ; to tempreg2
                   ; data at prog.memory address "305H" transferred to
                   ; tempreg2 and TBLH
                   ; in this example the data "1AH" is transferred to
                   ; tempreg1 and data "0FH" to register tempreg2
                   ; the value "00H" will be transferred to the high byte
                   ; register TBLH
  :
  :

org   300h         ; sets initial address of last page

dc    00Ah, 00Bh, 00Ch, 00Dh, 00Eh, 00Fh, 01Ah, 01Bh
  :
  :
```

Data Memory

The Data Memory is a volatile area of 8-bit wide RAM internal memory and is the location where temporary information is stored.

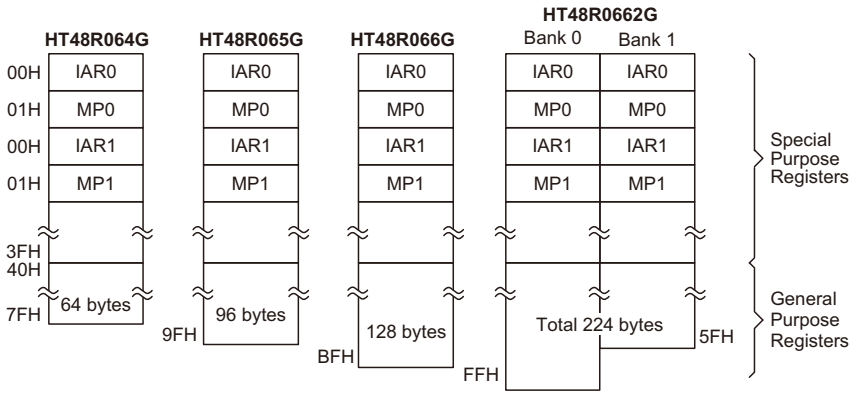
Structure

Divided into two sections, the first of these is an area of RAM where special function registers are located. These registers have fixed locations and are necessary for correct operation of the device. Many of these registers can be read from and written to directly under program control, however, some remain protected from user manipulation. The second area of Data Memory is reserved for general purpose use. All locations within this area are read and write accessible under program control.

Device	Capacity	Banks
HT48R064G	64×8	—
HT48R065G	96×8	—
HT48R066G	128×8	—
HT48R0662G	224×8	0, 1

The two sections of Data Memory, the Special Purpose and General Purpose Data Memory are located at consecutive locations. All are implemented in RAM and are 8 bits wide but the length of each memory section is dictated by the type of microcontroller chosen. The start address of the Data Memory for all devices is the address ″00H″.

All microcontroller programs require an area of read/write memory where temporary data can be stored and retrieved for use later. It is this area of RAM memory that is known as General Purpose Data Memory. This area of Data Memory is fully accessible by the user program for both read and write operations. By using the ″SET [m].i″ and ″CLR [m].i″ instructions individual bits can be set or reset under program control giving the user a large range of flexibility for bit manipulation in the Data Memory.



Data Memory Structure

Note: Most of the Data Memory bits can be directly manipulated using the ″SET [m].i″ and ″CLR [m].i″ with the exception of a few dedicated bits. The Data Memory can also be accessed through the memory pointer registers.

Special Purpose Data Memory

This area of Data Memory is where registers, necessary for the correct operation of the microcontroller, are stored. Most of the registers are both readable and writeable but some are protected and are readable only, the details of which are located under the relevant Special Function Register section. Note that for locations that are unused, any read instruction to these addresses will return the value ″00H″.

	HT48R064G	HT48R065G	HT48R066G	HT48R0662G
00H	IAR0	IAR0	IAR0	IAR0
01H	MP0	MP0	MP0	MP0
02H	IAR1	IAR1	IAR1	IAR1
03H	MP1	MP1	MP1	MP1
04H			BP	BP
05H	ACC	ACC	ACC	ACC
06H	PCL	PCL	PCL	PCL
07H	TBLP	TBLP	TBLP	TBLP
08H	TBLH	TBLH	TBLH	TBLH
09H	WDTS	WDTS	WDTS	WDTS
0AH	STATUS	STATUS	STATUS	STATUS
0BH	INTC0	INTC0	INTC0	INTC0
0CH	TMR0	TMR0	TMR0	TMR0
0DH	TMR0C	TMR0C	TMR0C	TMR0C
0EH		TMR1	TMR1	TMR1
0FH		TMR1C	TMR1C	TMR1C
10H	PA	PA	PA	PA
11H	PAC	PAC	PAC	PAC
12H	PAPU	PAPU	PAPU	PAPU
13H	PAWK	PAWK	PAWK	PAWK
14H	PB	PB	PB	PB
15H	PBC	PBC	PBC	PBC
16H	PBPU	PBPU	PBPU	PBPU
17H	PC	PC	PC	PC
18H	PCC	PCC	PCC	PCC
19H	PCPU	PCPU	PCPU	PCPU
1AH	CTRL0	CTRL0	CTRL0	CTRL0
1BH	CTRL1	CTRL1	CTRL1	CTRL1
1CH		LCDC	LCDC	LCDC
1DH		MFIC		PWM1
1EH	INTC1	INTC1	INTC1	INTC1
1FH				PWM0
20H				
21H				
22H				
23H				
24H			MFIC	MFIC
25H			PD	PD
26H			PDC	PDC
27H			PDPU	PDPU
28H	CMP0C	CMP0C		PE
29H	CMP1C	CMP1C		PEC
2AH	COPA0C	COPA0C		PEPU
2BH	COPA1C	COPA1C		PF
2CH	COPA2C	COPA2C		PFC
2DH	COPA3C	COPA3C		PFPU
2EH	OPA0OC	OPA0OC		
2FH	OPA1OC	OPA1OC		
30H				
31H			CTRL2	CTRL2
32H			CMP0C	CMP0C
33H			CMP1C	CMP1C
34H			COPA0C	COPA0C
35H			COPA1C	COPA1C
36H			COPA2C	COPA2C
37H			COPA3C	COPA3C
38H			OPA0OC	OPA0OC
39H			OPA1OC	OPA1OC
3AH				
≈	≈	≈	≈	≈
3EH				
3FH			PCWK	PCWK

: unused, read as 00H

Special Purpose Data Memory

Special Function Registers

To ensure successful operation of the microcontroller, certain internal registers are implemented in the Data Memory area. These registers ensure correct operation of internal functions such as timers, interrupts, etc., as well as external functions such as I/O data control. The location of these registers within the Data Memory begins at the address "00H" and are mapped into both Bank 0 and Bank 1. Any unused Data Memory locations between these special function registers and the point where the General Purpose Memory begins is reserved and attempting to read data from these locations will return a value of "00H".

Indirect Addressing Registers – IAR0, IAR1

The Indirect Addressing Registers, IAR0 and IAR1, although having their locations in normal RAM register space, do not actually physically exist as normal registers. The method of indirect addressing for RAM data manipulation uses these Indirect Addressing Registers and Memory Pointers, in contrast to direct memory addressing, where the actual memory address is specified. Actions on the IAR0 and IAR1 registers will result in no actual read or write operation to these registers but rather to the memory location specified by their corresponding Memory Pointer, MP0 or MP1. Acting as a pair, IAR0 with MP0 and IAR1 with MP1 can together access data from the Data Memory. As the Indirect Addressing Registers are not physically implemented, reading the Indirect Addressing Registers indirectly will return a result of "00H" and writing to the registers indirectly will result in no operation.

Memory Pointers – MP0, MP1

Two Memory Pointers, known as MP0 and MP1 are provided. These Memory Pointers are physically implemented in the Data Memory and can be manipulated in the same way as normal registers providing a convenient way with which to indirectly address and track data. MP0 can only be used to indirectly address data in Bank 0 while MP1 can be used to address data in Bank 0 and Bank1. When any operation to the relevant Indirect Addressing Registers is carried out, the actual address that the microcontroller is directed to, is the address specified by the related Memory Pointer. Note that for the HT48R064G device, bit 7 of the Memory Pointers is not required to address the full memory space. When bit 7 of the Memory Pointers for these devices is read, a value of "1" will be returned. Note that indirect addressing using MP1 and IAR1 must be used to access any data in Bank 1. The following example shows how to clear a section of four Data Memory locations already defined as locations adres1 to adres4.

• **Indirect Addressing Program Example**

```
data .section 'data'
adres1 db ?
adres2 db ?
adres3 db ?
adres4 db ?
block  db ?
code .section at 0 code
org  00h

start:
      mov a,04h              ; setup size of block
      mov block,a
      mov a,offset adres1    ; Accumulator loaded with first RAM address
      mov mp0,a              ; setup memory pointer with first RAM address

loop:
      clr IAR0               ; clear the data at address defined by MP0
      inc mp0                ; increment memory pointer
      sdz block              ; check if last memory location has been cleared
      jmp loop

      continue:
```

The important point to note here is that in the example shown above, no reference is made to specific Data Memory addresses.

Accumulator – ACC

The Accumulator is central to the operation of any microcontroller and is closely related with operations carried out by the ALU. The Accumulator is the place where all intermediate results from the ALU are stored. Without the Accumulator it would be necessary to write the result of each calculation or logical operation such as addition, subtraction, shift, etc., to the Data Memory resulting in higher programming and timing overheads. Data transfer operations usually involve the temporary storage function of the Accumulator; for example, when transferring data between one user defined register and another, it is necessary to do this by passing the data through the Accumulator as no direct transfer between two registers is permitted.

Program Counter Low Register – PCL

To provide additional program control functions, the low byte of the Program Counter is made accessible to programmers by locating it within the Special Purpose area of the Data Memory. By manipulating this register, direct jumps to other program locations are easily implemented. Loading a value directly into this PCL register will cause a jump to the specified Program Memory location, however, as the register is only 8-bit wide, only jumps within the current Program Memory page are permitted. When such operations are used, note that a dummy cycle will be inserted.

Bank Pointer – BP

In the HT48R0662G device, the Data Memory is divided into two Banks, known as Bank 0 and Bank 1. A Bank Pointer, which is bit 0 of the Bank Pointer register is used to select the required Data Memory bank. Only data in Bank 0 can be directly addressed as data in Bank 1 must be indirectly addressed using Memory Pointer MP1 and Indirect Addressing Register IAR1. Using Memory Pointer MP0 and Indirect Addressing Register IAR0 will always access data from Bank 0, irrespective of the value of the Bank Pointer. Memory Pointer MP1 and Indirect Addressing Register IAR1 can indirectly address data in either Bank 0 or Bank 1 depending upon the value of the Bank Pointer.

The Data Memory is initialised to Bank 0 after a reset, except for the WDT time-out reset in the Idle/Sleep Mode, in which case, the Data Memory bank remains unaffected. It should be noted that Special Function Data Memory is not affected by the bank selection, which means that the Special Function Registers can be accessed from within either Bank 0 or Bank 1. Directly addressing the Data Memory will always result in Bank 0 being accessed irrespective of the value of the Bank Pointer.

- **BP Register – HT48R0662G**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	DMBP0
R/W	—	—	—	—	—	—	—	R/W
POR	—	—	—	—	—	—	—	0

Bit 7~1 : unimplemented, read as ″0″

Bit 0 **DMBP0**: Data Memory bank point
0: Bank 0
1: Bank 1

Status Register – STATUS

This 8-bit register contains the zero flag (Z), carry flag (C), auxiliary carry flag (AC), overflow flag (OV), power down flag (PDF), and watchdog time-out flag (TO). These arithmetic/logical operation and system management flags are used to record the status and operation of the microcontroller.

With the exception of the TO and PDF flags, bits in the status register can be altered by instructions like most other registers. Any data written into the status register will not change the TO or PDF flag. In addition, operations related to the status register may give different results due to the different instruction operations. The TO flag can be affected only by a system power-up, a WDT time-out or by executing the ″CLR WDT″ or ″HALT″ instruction. The PDF flag is affected only by executing the ″HALT″ or ″CLR WDT″ instruction or during a system power-up.

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

In addition, on entering an interrupt sequence or executing a subroutine call, the status register will not be pushed onto the stack automatically. If the contents of the status registers are important and if the interrupt routine can change the status register, precautions must be taken to correctly save it. Note that bits 0~3 of the STATUS register are both readable and writeable bits.

• **STATUS Register**

Bit	7	6	5	4	3	2	1	0
Name	—	—	TO	PDF	OV	Z	AC	C
R/W	—	—	R	R	R/W	R/W	R/W	R/W
POR	—	—	0	0	x	x	x	x

″x″ unknown

Bit 7, 6 unimplemented, read as ″0″

Bit 5 **TO**: Watchdog Time-Out flag
0: After power up or executing the ″CLR WDT″ or ″HALT″ instruction
1: A watchdog time-out occurred.

Bit 4 **PDF**: Power down flag
0: After power up or executing the ″CLR WDT″ instruction
1: By executing the ″HALT″ instruction

Bit 3 **OV**: Overflow flag
0: no overflow
1: an operation results in a carry into the highest-order bit but not a carry out of the highest-order bit or vice versa.

Bit 2 **Z**: Zero flag
0: The result of an arithmetic or logical operation is not zero
1: The result of an arithmetic or logical operation is zero

Bit 1 **AC**: Auxiliary flag
0: no auxiliary carry
1: 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

Bit 0 C: Carry flag
0: no carry-out
1: an operation results in a carry during an addition operation or if a borrow does not take place during a subtraction operation
C is also affected by a rotate through carry instruction.

Input/Output Ports and Control Registers

Within the area of Special Function Registers, the port PA, PB, etc data I/O registers and their associated control register PAC, PBC, etc play a prominent role. These registers are mapped to specific addresses within the Data Memory as shown in the Data Memory table. The data I/O registers, are used to transfer the appropriate output or input data on the port. The control registers specifies which pins of the port are set as inputs and which are set as outputs. To setup a pin as an input, the corresponding bit of the control register must be set high, for an output it must be set low. During program initialisation, it is important to first setup the control registers to specify which pins are outputs and which are inputs before reading data from or writing data to the I/O ports. One flexible feature of these registers is the ability to directly program single bits using the "SET [m].i" and "CLR [m].i" instructions. The ability to change I/O pins from output to input and vice versa by manipulating specific bits of the I/O control registers during normal program operation is a useful feature of these devices.

System Control Registers – CTRL0, CTRL1, CTRL2

These registers are used to provide control over several internal functions. These functions include the external Interrupt edge trigger type, the PWM function control, Time Base period selection and LXT oscillator low power control,etc.

- **CTRL0 Register – HT48R064G**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	PFDC	LXTLP	CLKMOD
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	0	0	0

- **CTRL0 Register – HT48R065G**

Bit	7	6	5	4	3	2	1	0
Name	—	PFDCS	—	—	—	PFDC	LXTLP	CLKMOD
R/W	—	R/W	—	—	—	R/W	R/W	R/W
POR	—	0	—	—	—	0	0	0

- **CTRL0 Register – HT48R0662G**

Bit	7	6	5	4	3	2	1	0
Name	—	PFDCS	PWMSEL	PWMC1	PWMC0	PFDC	LXTLP	CLKMOD
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

unimplemented, read as "0"

PFDCS: PFD clock source selection
0: timer0
1: timer1
For HT48R064G device, this bit is read as 0 and the PFD clock source always comes from the timer.

PWMSEL: PWM type selection
0: 6+2 type
1: 7+1 type

PWMC1: I/O or PWM1 selection
0: I/O
1: PWM1

PWMC0: I/O or PWM0 selection
0: I/O or other pin-shared functions
1: PWM0

PFDC: I/O or PFD selection
0: I/O
1: PFD

LXTLP: LXT oscillator low power control function
0: LXT Oscillator quick start-up mode
1: LXT Oscillator Low Power Mode

CLKMOD: system clock mode selection
0: High speed - HIRC oscillator used as system clock
1: Low speed - LXT oscillator used as system clock, HIRC oscillator stopped

For HT48R064G/HT48R065G devices, this bit is available if the oscillator configuration options have selected the HIRC+LXT.

Note: If the PWMn output is selected by the PWMCn bit, the PWM clock source f_{TP} always comes from the system clock source f_{SYS}. The f_{TP} clock is the clock source for timer0, timer 1, time base and PWM.

- **CTRL0 Register – HT48R066G**

Bit	7	6	5	4	3	2	1	0
Name	PCFG	PFDCS	—	—	PFDEN1	PFDEN0	LXTLP	CLKMOD
R/W	R/W	R/W	—	—	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **PCFG**: I/O configuration
0: INTB/TC0/PFD pin-shared with PA3/PA2/PA0
1: INTB/TC0/PFD pin-shared with PC5/PC4/PC3

Bit 6 **PFDCS**: PFD clock source
0: Timer 0
1: Timer 1

Bit 5~4 unimplemented, read as ″0″

Bit 3~2: **PFDEN[1:0]**: PFD/PFDB enable/ disable
00: Both disable
01: unimplemented, read as ″0″
10: PFD enable
11: PFD and PFDB enable

Bit 1 **LXTLP**: LXT oscillator low power control function
0: LXT oscillator quick start-up mode
1: LXT oscillator Low Power Mode

Bit 0 **CLKMOD**: System clock mode selection
0: High speed - HIRC used as system clock
1: Low speed - LXT used as system clock, HIRC oscillator stopped

This clock mode selection is only valid if the oscillator configuration option has selected the HIRC+LXT.

• **CTRL1 Register**

Bit	7	6	5	4	3	2	1	0
Name	INTEG1	INTEG0	TBSEL1	TBSEL0	WDTEN3	WDTEN2	WDTEN1	WDTEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	0	0	0	1	0	1	0

Bit 7, 6 **INTEG1, INTEG0**: External interrupt edge type
00: disable
01: rising edge trigger
10: falling edge trigger
11: dual edge trigger

Bit 5, 4 **TBSEL1, TBSEL0**: Time base period selection
00: $2^{10}/f_{TP}$
01: $2^{11}/f_{TP}$
10: $2^{12}1/f_{TP}$
11: $2^{13}1/f_{TP}$

Bit 3~0 **WDTEN3, WDTEN2, WDTEN1, WDTEN0**: WDT function enable
1010: WDT function disabled
Other values: WDT function enabled - Recommended value is 0101
If the "watchdog timer enable configuration option" is selected, then the watchdog timer will always be enabled and the WDTEN3~WDTEN0 control bits will have no effect.

Note: The WDT is only disabled when both the WDT configuration option is disabled and when bits WDTEN3~WDTEN0 is set to 1010. The WDT is enabled when either the WDT configuration option is enabled or when bits WDTEN3~WDTEN0≠1010.

• **CTRL2 Register – HT48R0662G**

Bit	7	6	5	4	3	2	1	0
Name	PCFG1	PCFG0	—	—	—	—	—	LXTEN
R/W	R/W	R/W	—	—	—	—	—	R/W
POR	0	0	—	—	—	—	—	1

Bit 7~6 **PCFG1**, **PCFG0**: Pin configuration
00: PFD/TC0/INT/TC1 pin-shared with PA1/PA2/PA3/PA4
01: PFD/TC0/INT/TC1 pin-shared with PC5/PC4/PC3/PC2
10: PFD/TC0/INT/TC1 pin-shared with PB0/PB1/PB2/PB3
11: PFD/TC0/INT/TC1 pin-shared with PE0/PE1/PE2/PE3

Bit 5~1 Unimplemented, read as "0"

Bit 0 **LXTEN**: LXT Oscillator on/off control after execution of HALT instruction
0: LXT oscillator off after HALT instruction
1: LXT oscillator on after HALT instruction

Wake-up Function Register – PAWK, PCWK

When the microcontroller enters the Idle/Sleep Mode, various methods exist to wake the device up and continue with normal operation. One method is to allow a falling edge on the I/O pins to have a wake-up function. These register are used to selected which pins on I/O Port A or Port C are used to have this wake-up function.

Pull-high Registers – PAPU, PBPU, PCPU, PDPU, PEPU, PFPU

The I/O pins, if configured as inputs, can have internal pull-high resistors connected, which eliminates the need for external pull-high resistors. This register selects which I/O pins are connected to internal pull-high resistors.

Software COM Register – SCOMC

For HT48R065G, HT48R066G and HT48R0662G devices, the pins PB0~PB3 on Port B can be used as SCOM lines to drive an external LCD panel. To implement this function, the SCOMC register is used to setup the correct bias voltages on these pins.

Oscillator

Various oscillator options offer the user a wide range of functions according to their various application requirements. The flexible features of the oscillator functions ensure that the best optimisation can be achieved in terms of speed and power saving. Oscillator selections and operation are selected through a combination of configuration options and registers.

System Oscillator Overview

In addition to being the source of the main system clock the oscillators also provide clock sources for the Watchdog Timer and Time Base functions. External oscillators requiring some external components as well as a two fully integrated internal oscillators, requiring no external components, are provided to form a wide range of both fast and slow system oscillators.

Type	Name	Freq.	Pins
External Crystal	HXT	400kHz~12MHz	OSC1/OSC2
External RC	ERC	400kHz~12MHz	OSC1
Internal High Speed RC	HIRC	4, 8 or 12MHz	—
External Low Speed Crystal	LXT	32768Hz	OSC3/OSC4*
Internal Low Speed RC	LIRC	13kHz	—

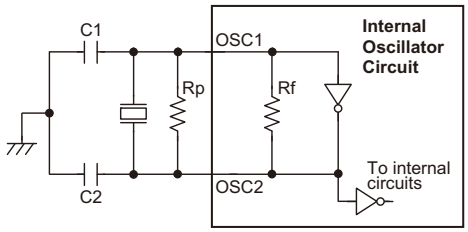
"*" For HT48R0662G only

System Clock Configurations

There are four system oscillators implemented in this device, three high speed oscillators and one low speed oscillator. The high speed oscillators are the external crystal/ceramic oscillator -- HXT, the external RC oscillator -- ERC and the internal RC oscillator -- HIRC. The low speed oscillator is the external 32.768kHz crystal oscillator -- LXT. The LXT oscillator can be used as the system oscillator only when the HIRC oscillator is selected as the high speed system oscillator for the HT48R0662G device. Also there is an internal 13kHz RC oscillator named LIRC oscillator used as the clock source for the WDT function. More details are described in the accompanying sections.

External Crystal/Resonator Oscillator – HXT

The simple connection of a crystal across OSC1 and OSC2 will create the necessary phase shift and feedback for oscillation. However, for some crystals and most resonator types, to ensure oscillation and accurate frequency generation, it is necessary to add two small value external capacitors, C1 and C2. The exact values of C1 and C2 should be selected in consultation with the crystal or resonator manufacturer′s specification.



Note: 1. Rp is normally not required. C1 and C2 are required.
2. Although not shown OSC1/OSC2 pins have a parasitic capacitance of around 7pF.

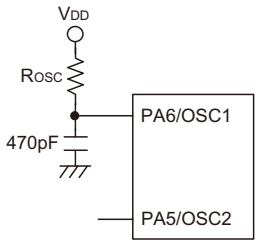
Crystal/Resonator Oscillator – HXT

Crystal Oscillator C1 and C2 Values		
Crystal Frequency	**C1**	**C2**
12MHz	—	—
8MHz	—	—
4MHz	—	—
1MHz	100pF	100pF
Note: C1 and C2 values are for guidance only.		

Crystal Recommended Capacitor Values

External RC Oscillator – ERC

Using the ERC oscillator only requires that a resistor, with a value between 24kΩ and 1.5MΩ, is connected between OSC1 and VDD, and a capacitor is connected between OSC and ground, providing a low cost oscillator configuration. It is only the external resistor that determines the oscillation frequency; the external capacitor has no influence over the frequency and is connected for stability purposes only. Device trimming during the manufacturing process and the inclusion of internal frequency compensation circuits are used to ensure that the influence of the power supply voltage, temperature and process variations on the oscillation frequency are minimised. Here only the OSC1 pin is used, which is shared with I/O pin PA6, leaving pin PA5 free for use as a normal I/O pin.



External RC Oscillator – ERC

Internal RC Oscillator – HIRC

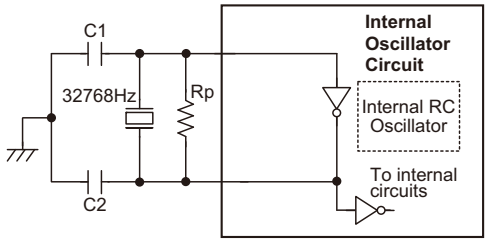
The internal RC oscillator is a fully integrated system oscillator requiring no external components. The internal RC oscillator has three fixed frequencies of either 4MHz, 8MHz or 12MHz. Device trimming during the manufacturing process and the inclusion of internal frequency compensation circuits are used to ensure that the influence of the power supply voltage, temperature and process variations on the oscillation frequency are minimised. Refer to the A.C. Characteristics for more frequency accuracy details. Note that if this internal system clock option is selected, as it requires no external pins for its operation, I/O pins PA5 and PA6 are free for use as normal I/O pins or the LXT oscillator pins depending upon the selected device.

Note: PA5/PA6 used as normal I/Os

Internal RC Oscillator – HIRC

External 32768Hz Crystal Oscillator – LXT

When the microcontroller enters the Idle/Sleep Mode, the system clock is switched off to stop microcontroller activity and to conserve power. However, in many microcontroller applications it may be necessary to keep the internal timers operational even when the microcontroller is in the Power-down Mode. To do this, another clock, independent of the system clock, must be provided. To do this a configuration option exists to allow a high speed oscillator to be used in conjunction with a low speed oscillator, known as the LXT oscillator. The LXT oscillator is implemented using a 32768Hz crystal connected to pins OSC1/OSC2 for the HT48R064G/HT48R065G or connected to pins OSC3/OSC4 for the HT48R0662G. However, for some crystals, to ensure oscillation and accurate frequency generation, it is necessary to add two small value external capacitors, C1 and C2. The exact values of C1 and C2 should be selected in consultation with the crystal or resonator manufacturers specification. The external parallel feedback resistor, Rp, is required. For the HT48R064G/HT48R065G devices the LXT oscillator must be used together with the HIRC oscillator. For the HT48R0662G device the LXT oscillator must be used together with the HXT, ERC or HIRC register.



Note: 1. Rp, C1 and C2 are required.
2. Although not shown pins have a parasitic capacitance of around 7pF.

External LXT Oscillator – LXT

LXT Oscillator C1 and C2 Values		
Crystal Frequency	C1	C2
32768Hz	10pF	10pF
Note: 1. C1 and C2 values are for guidance only. 2. R_P=5M~10MΩ is recommended.		

32768 Hz Crystal Recommended Capacitor Values

For the HT48R0662G device, a configuration option determines if the OSC3/OSC4 pins are used for the LXT oscillator or as I/O pins.

- If the I/O option is selected then the OSC3/OSC4 pins can be used as normal I/O pins.
- If the LXT oscillator is selected, then the 32.768kHz crystal should be connected to the OSC3/OSC4 pins.

LXT Oscillator Low Power Function

The LXT oscillator can function in one of two modes, the Quick Start Mode and the Low Power Mode. The mode selection is executed using the LXTLP bit in the CTRL0 register.

LXTLP Bit	LXT Mode
0	Quick Start
1	Low-power

After power on the LXTLP bit will be automatically cleared to zero ensuring that the LXT oscillator is in the Quick Start operating mode. In the Quick Start Mode the LXT oscillator will power up and stabilise quickly. However, after the LXT oscillator has fully powered up it can be placed into the Low-power mode by setting the LXTLP bit high. The oscillator will continue to run but with reduced current consumption, as the higher current consumption is only required during the LXT oscillator start-up. In power sensitive applications, such as battery applications, where power consumption must be kept to a minimum, it is therefore recommended that the application program sets the LXTLP bit high about 2 seconds after power-on.

It should be noted that, no matter what condition the LXTLP bit is set to, the LXT oscillator will always function normally, the only difference is that it will take more time to start up if in the Low-power mode.

Internal Low Speed Oscillator – LIRC

The LIRC is a fully self-contained free running on-chip RC oscillator with a typical frequency of 13kHz at 5V requiring no external components. When the device enters the Idle/Sleep Mode, the system clock will stop running but the LIRC oscillator continues to free-run and to keep the watchdog active. However, to preserve power in certain applications the LIRC can be disabled via a configuration option.

Operating Modes

By using the LXT low frequency oscillator in combination with a high frequency oscillator, the system can be selected to operate in a number of different modes. These Modes are Normal, Slow, Idle and Sleep.

Mode Types and Selection

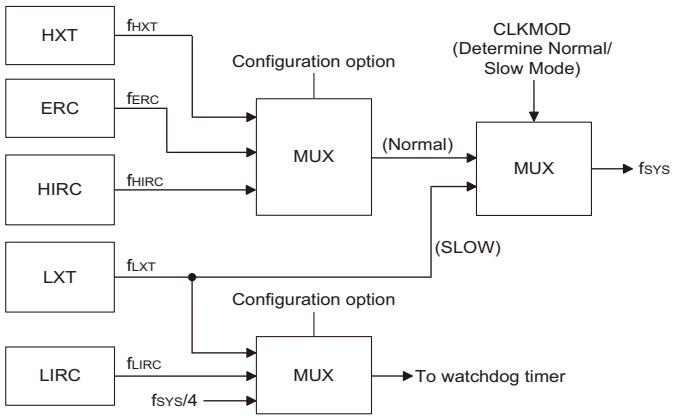
HT48R064G/HT48R065G/HT48R066G

For these devices, if the LXT oscillator is used then the internal RC oscillator, HIRC, must be used as the high frequency oscillator. If the HXT or the ERC oscillator is chosen as the high frequency system clock then the LXT oscillator cannot be used as they share the same oscillator pins. The CLKMOD bit in the CTRL0 register can be used to switch the system clock from the high speed HIRC oscillator to the low speed LXT oscillator. When the HALT instruction is executed and the device enters the Idle/Sleep Mode the LXT oscillator will always continue to run. For these devices the LXT crystal is connected to the OSC1/OSC2 pins and LXT will always run (the LXTEN bit is not used). Note that CLKMOD is only valid in HIRC+LXT oscillator configuration for HT48R064G/HT48R065G/ HT48R066G.

HT48R0662G

For the device the LXT oscillator can run together with any of the high speed oscillators, namely the HXT, ERC or the HIRC. The CLKMOD bit in the CTRL0 register can be used to switch the system clock from the selected high speed oscillator to the low speed LXT oscillator. When the HALT instruction is executed the LXT oscillator can be chosen to run or not using the LXTEN bit in the CTRL2 register.

Note that CLKMOD is only valid in HIRC+LXT oscillator configuration.



System Clock Configurations

For all devices, when the system enters the Sleep or Idle Mode, the high frequency system clock will always stop running. The accompanying tables shows the relationship between the CLKMOD bit, the HALT instruction and the high/low frequency oscillators. The CLMOD bit can change normal or Slow Mode.

Operating Mode Control

- HT48R064G/HT48R065G/HT48R066G

Operating Mode	OSC1/OSC2 Configuration				
	HXT	ERC	HIRC	HIRC + LXT	
				HIRC	LXT
Normal	Run	Run	Run	Run	Run
Slow	—	—	—	Stop	Run
Sleep	Stop	Stop	Stop	Stop	Run

"—" unimplemented

- HT48R0662G

Operating Mode	OSC1/OSC2 Configuration			OSC3/OSC4 Configuration	
	HXT	ERC	HIRC	LXT	
				LXTEN=0	LXTEN=1
Normal	Run	Run	Run	Run	Run
Slow	Stop	Stop	Stop	Run	Run
Idle	Stop	Stop	Stop	Stop	Run
Sleep	Stop	Stop	Stop	Stop	Stop

Mode Switching

The devices are switched between one mode and another using a combination of the CLKMOD bit in the CTRL0 register and the HALT instruction. The CLKMOD bit chooses whether the system runs in either the Normal or Slow Mode by selecting the system clock to be sourced from either a high or low frequency oscillator. The HALT instruction forces the system into either the Idle or Sleep Mode, depending upon whether the LXT oscillator is running or not. The HALT instruction operates independently of the CLKMOD bit condition.

When a HALT instruction is executed and the LXT oscillator is not running, the system enters the Sleep mode the following conditions exist:

- The system oscillator will stop running and the application program will stop at the "HALT" instruction.
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and resume counting if the WDT clock source is selected to come from the LIRC or LXT oscillator. The WDT will stop if its clock source originates from the system clock.
- The I/O ports will maintain their present condition.
- In the status register, the Power Down flag, PDF, will be set and the Watchdog time-out flag, TO, will be cleared.

Standby Current Considerations

As the main reason for entering the Idle/Sleep Mode is to keep the current consumption of the MCU to as low a value as possible, perhaps only in the order of several micro-amps, there are other considerations which must also be taken into account by the circuit designer if the power consumption is to be minimised.

Special attention must be made to the I/O pins on the device. All high-impedance input pins must be connected to either a fixed high or low level as any floating input pins could create internal oscillations and result in increased current consumption. Care must also be taken with the loads, which are connected to I/O pins, which are setup as outputs. These should be placed in a condition in which minimum current is drawn or connected only to external circuits that do not draw current, such as other CMOS inputs.

If the configuration options have enabled the LIRC oscillator, then this will continue to run when in the Idle/Sleep Mode and will thus consume some power. For power sensitive applications it may be therefore preferable to use the system clock source for the Watchdog Timer. The LXT, if configured for use, will also consume a limited amount of power, as it continues to run when the device enters the Idle/Sleep Mode. To keep the LXT power consumption to a minimum level the LXTLP bit in the CTRL0 register, which controls the low power function, should be set high.

Wake-up

After the system enters the Idle/Sleep Mode, it can be woken up from one of various sources listed as follows:

- An external reset
- An external falling edge on PA0~PA7 or PC0~PC7 (HT48R0662G only)
- A system interrupt
- A WDT overflow

If the system is woken up by an external reset, the device will experience a full system reset, however, if the device is woken up by a WDT overflow, a Watchdog Timer reset will be initiated. Although both of these wake-up methods will initiate a reset operation, the actual source of the wake-up can be determined by examining the TO and PDF flags. The PDF flag is cleared by a system power-up or executing the clear Watchdog Timer instructions and is set when 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 Stack Pointer, the other flags remain in their original status.

Pins PA0 to PA7 or PC0 to PC7 can be setup via the PAWK or PCWK register to permit a negative transition on the pin to wake-up the system. When a pin on PA0~PA7 or PC0~PC7 wake-up occurs, the program will resume execution at the instruction following the ″HALT″ instruction.

If the system is woken up by an interrupt, then two possible situations may occur. The first is where the related interrupt is disabled or the interrupt is enabled but the stack is full, in which case the program will resume execution at the instruction following the ″HALT″ instruction. In this situation, the interrupt which woke-up the device will not be immediately serviced, but will rather be serviced later when the related interrupt is finally enabled or when a stack level becomes free. The other situation is where the related interrupt is enabled and the stack is not full, in which case the regular interrupt response takes place. If an interrupt request flag is set to ″1″ before entering the Idle/Sleep Mode, then any future interrupt requests will not generate a wake-up function and the related interrupt will be ignored.

No matter what the source of the wake-up event is, once a wake-up event occurs, there will be a time delay before normal program execution resumes. Consult the table for the related time.

Wake-up Source	Oscillator Type	
	ERC, IRC	Crystal
External $\overline{RES}$	t_{RSDT} + t_{SST2}	t_{RSDT} + t_{SST2}
PA or PC* Port	t_{SST1}	t_{SST2}
Interrupt		
WDT Overflow		

"*" Port C pin wake-up is only available for the HT48R0662G device.

Note: 1. t_{RSTD} (reset delay time), t_{SYS} (system clock)
2. t_{RSTD} is power-on delay, typical time=50ms
3. t_{SST1}= 2 t_{SYS}
4. t_{SST2}= 128 t_{SYS}

Wake-up Delay Time

Watchdog Timer

The Watchdog Timer, also known as the WDT, is provided to inhibit program malfunctions caused by the program jumping to unknown locations due to certain uncontrollable external events such as electrical noise.

Watchdog Timer Operation

It operates by providing a device reset when the Watchdog Timer counter overflows. Note that if the Watchdog Timer function is not enabled, then any instructions related to the Watchdog Timer will result in no operation.

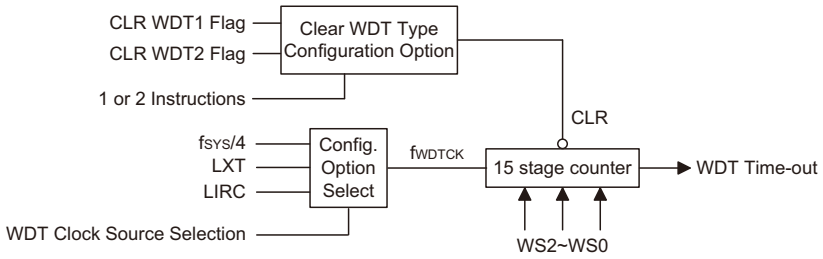
Setting up the various Watchdog Timer options are controlled via the configuration options and two internal registers WDTS and CTRL1. Enabling the Watchdog Timer can be controlled by both a configuration option and the WDTEN bits in the CTRL1 internal register in the Data Memory.

Configuration Option	CTRL1 Register	WDT Function
Disable	Disable	OFF
Disable	Enable	ON
Enable	x	ON

Watchdog Timer On/Off Control

The Watchdog Timer will be disabled if bits WDTEN3~WDTEN0 in the CTRL1 register are written with the binary value 1010B and WDT configuration option is disable. This will be the condition when the device is powered up. Although any other data written to WDTEN3~WDTEN0 will ensure that the Watchdog Timer is enabled, for maximum protection it is recommended that the value 0101B is written to these bits.

The Watchdog Timer clock can emanate from three different sources, selected by configuration option. These are LXT, $f_{SYS}/4$, or LIRC. It is important to note that when the system enters the Idle/Sleep Mode the instruction clock is stopped, therefore if the configuration options have selected $f_{SYS}/4$ as the Watchdog Timer clock source, the Watchdog Timer will cease to function. For systems that operate in noisy environments, using the LIRC or the LXT as the clock source is therefore the recommended choice. The division ratio of the prescaler is determined by bits 0, 1 and 2 of the WDTS register, known as WS0, WS1 and WS2. If the Watchdog Timer internal clock source is selected and with the WS0, WS1 and WS2 bits of the WDTS register all set high, the prescaler division ratio will give a maximum time-out period.



Watchdog Timer

Under normal program operation, a Watchdog Timer time-out will initialise a device reset and set the status bit TO. However, if the system is in the Idle/Sleep Mode, when a Watchdog Timer time-out occurs, the device will be woken up, the TO bit in the status register will be set and only the Program Counter and Stack Pointer will be reset. Three methods can be adopted to clear the contents of the Watchdog Timer. The first is an external hardware reset, which means a low level on the external reset pin, the second is using the Clear Watchdog Timer software instructions and the third is when a HALT instruction is executed. There are two methods of using software instructions to clear the Watchdog Timer, one of which must be chosen by configuration option. The first option is to use the single ″CLR WDT″ instruction while the second is to use the two commands ″CLR WDT1″ and ″CLR WDT2″. For the first option, a simple execution of ″CLR WDT″ will clear the Watchdog Timer while for the second option, both ″CLR WDT1″ and ″CLR WDT2″ must both be executed to successfully clear the Watchdog Timer. Note that for this second option, if ″CLR WDT1″ is used to clear the Watchdog Timer, successive executions of this instruction will have no effect, only the execution of a ″CLR WDT2″ instruction will clear the Watchdog Timer. Similarly after the ″CLR WDT2″ instruction has been executed, only a successive ″CLR WDT1″ instruction can clear the Watchdog Timer.

WDTS Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	WS2	WS1	WS0
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	1	1	1

Bit 7~3 : unimplemented, read as ″0″

Bit 2~0 **WS2, WS1, WS0**: WDT time-out period selection

000: 2^8 t_{WDTCK}
001: 2^9 t_{WDTCK}
010: 2^{10} t_{WDTCK}
011: 2^{11} t_{WDTCK}
100: 2^{12} t_{WDTCK}
101: 2^{13} t_{WDTCK}
110: 2^{14} t_{WDTCK}
111: 2^{15} t_{WDTCK}

Reset and Initialisation

A reset function is a fundamental part of any microcontroller ensuring that the device can be set to some predetermined condition irrespective of outside parameters. The most important reset condition is after power is first applied to the microcontroller. In this case, internal circuitry will ensure that the microcontroller, after a short delay, will be in a well defined state and ready to execute the first program instruction. After this power-on reset, certain important internal registers will be set to defined states before the program commences. One of these registers is the Program Counter, which will be reset to zero forcing the microcontroller to begin program execution from the lowest Program Memory address.

In addition to the power-on reset, situations may arise where it is necessary to forcefully apply a reset condition when the microcontroller is running. One example of this is where after power has been applied and the microcontroller is already running, the $\overline{\text{RES}}$ line is forcefully pulled low. In such a case, known as a normal operation reset, some of the microcontroller registers remain unchanged allowing the microcontroller to proceed with normal operation after the reset line is allowed to return high. Another type of reset is when the Watchdog Timer overflows and resets the microcontroller. All types of reset operations result in different register conditions being setup.

Another reset exists in the form of a Low Voltage Reset, LVR, where a full reset, similar to the $\overline{\text{RES}}$ reset is implemented in situations where the power supply voltage falls below a certain threshold.

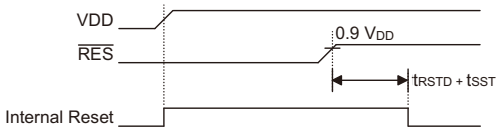
Reset Functions

There are five ways in which a microcontroller reset can occur, through events occurring both internally and externally:

Power-on Reset

The most fundamental and unavoidable reset is the one that occurs after power is first applied to the microcontroller. As well as ensuring that the Program Memory begins execution from the first memory address, a power-on reset also ensures that certain other registers are preset to known conditions. All the I/O port and port control registers will power up in a high condition ensuring that all pins will be first set to inputs.

Although the microcontroller has an internal RC reset function, if the VDD power supply rise time is not fast enough or does not stabilise quickly at power-on, the internal reset function may be incapable of providing proper reset operation. For this reason it is recommended that an external RC network is connected to the $\overline{\text{RES}}$ pin, whose additional time delay will ensure that the $\overline{\text{RES}}$ pin remains low for an extended period to allow the power supply to stabilise. During this time delay, normal operation of the microcontroller will be inhibited. After the $\overline{\text{RES}}$ line reaches a certain voltage value, the reset delay time t_{RSTD} is invoked to provide an extra delay time after which the microcontroller will begin normal operation. The abbreviation SST in the figures stands for System Start-up Timer.

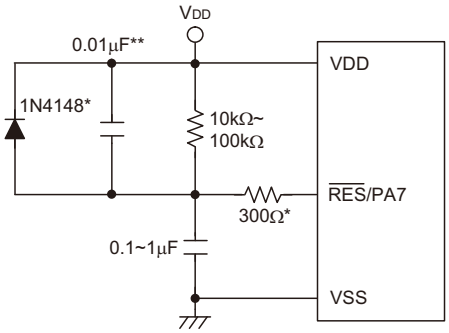


Note: t_{RSTD} is power-on delay, typical time=50ms

Power-On Reset Timing Chart

For most applications a resistor connected between VDD and the $\overline{RES}$ pin and a capacitor connected between VSS and the $\overline{RES}$ pin will provide a suitable external reset circuit. Any wiring connected to the $\overline{RES}$ pin should be kept as short as possible to minimise any stray noise interference.

For applications that operate within an environment where more noise is present the Reset Circuit shown is recommended.



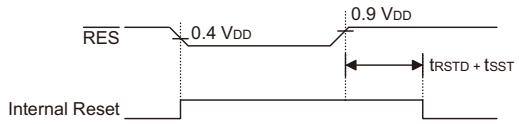
Note: "*" It is recommended that this component is added for added ESD protection
"**" It is recommended that this component is added in environments where power line noise is significant

External $\overline{RES}$ Circuit

More information regarding external reset circuits is located in Application Note HA0075E on the Holtek website.

$\overline{RES}$ Pin Reset

This type of reset occurs when the microcontroller is already running and the $\overline{RES}$ pin is forcefully pulled low by external hardware such as an external switch. In this case as in the case of other reset, the Program Counter will reset to zero and program execution initiated from this point.

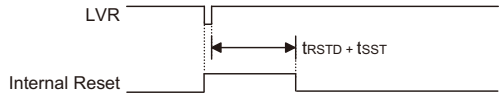


Note: t_{RSTD} is power-on delay, typical time=50ms

$\overline{RES}$ Reset Timing Chart

Low Voltage Reset – LVR

The microcontroller contains a low voltage reset circuit in order to monitor the supply voltage of the device. The LVR function is selected via a configuration option. If the supply voltage of the device drops to within a range of 0.9V~V_{LVR} such as might occur when changing the battery, the LVR will automatically reset the device internally. For a valid LVR signal, a low supply voltage, i.e., a voltage in the range between 0.9V~V_{LVR} must exist for a time greater than that specified by t_{LVR} in the A.C. characteristics. If the low supply voltage state does not exceed this value, the LVR will ignore the low supply voltage and will not perform a reset function. The actual V_{LVR} value can be selected via configuration options.

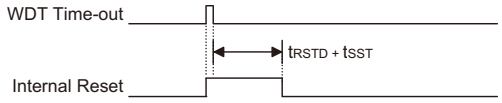


Note: t_{RSTD} is power-on delay, typical time=50ms

Low Voltage Reset Timing Chart

Watchdog Time-out Reset during Normal Operation

The Watchdog time-out Reset during normal operation is the same as a hardware $\overline{RES}$ pin reset except that the Watchdog time-out flag TO will be set to "1".

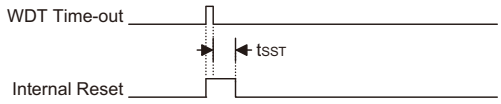


Note: t_{RSTD} is power-on delay, typical time=50ms

WDT Time-out Reset during Normal Operation Timing Chart

Watchdog Time-out Reset during Idle/Sleep Mode

The Watchdog time-out Reset during Idle/Sleep mode is a little different from other kinds of reset. Most of the conditions remain unchanged except that the Program Counter and the Stack Pointer will be cleared to "0" and the TO flag will be set to "1". Refer to the A.C. Characteristics for t_{SST} details.



WDT Time-out Reset during Idle/Sleep Timing Chart

Note: The t_{SST} can be chosen to be either 128 or 2 clock cycles via configuration option if the system clock source is provided by ERC or HIRC. The SST is 128 for HXT or LXT.

Reset Initial Conditions

The different types of reset described affect the reset flags in different ways. These flags, known as PDF and TO are located in the status register and are controlled by various microcontroller operations, such as the Idle/Sleep function or Watchdog Timer. The reset flags are shown in the table:

TO	PDF	RESET Conditions
0	0	Power-on reset
u	u	$\overline{RES}$ or LVR reset during Normal or Slow Mode operation
1	u	WDT time-out reset during Normal or Slow Mode operation
1	1	WDT time-out reset during Idle or Sleep Mode operation

Note: "u" stands for unchanged

The following table indicates the way in which the various components of the microcontroller are affected after a power-on reset occurs.

Item	Condition After RESET
Program Counter	Reset to zero
Interrupts	All interrupts will be disabled
WDT	Clear after reset, WDT begins counting
Timer/Event Counter	Timer Counter will be turned off
Prescaler	The Timer Counter Prescaler will be cleared
Input/Output Ports	I/O ports will be setup as inputs
Stack Pointer	Stack Pointer will point to the top of the stack

The different kinds of resets all affect the internal registers of the microcontroller in different ways. To ensure reliable continuation of normal program executio n after a reset occurs, it is important to know what condition the microcontroller is in after a particular reset occurs. The following table describes how each type of reset affects each of the microcontroller internal registers.

Register	HT48R064G	HT48R065G	HT46R066G	HT46R0662G	Power-on Reset	$\overline{\text{RES}}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (Idle/Sleep)
PCL	•	•	•	•	0000 0000	0000 0000	0000 0000	0000 0000
MP0	•				1xxx xxxx	1xxx xxxx	1xxx xxxx	1uuu uuuu
		•	•	•	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
MP1	•				1xxx xxxx	1xxx xxxx	1xxx xxxx	1uuu uuuu
		•	•	•	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
BP				•	---- ---0	---- ---0	---- ---0	---- ---u
ACC	•	•	•	•	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLP	•	•	•	•	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	•				--xx xxxx	--uu uuuu	--uu uuuu	--uu uuuu
		•	•	•	-xxx xxxx	-uuu uuuu	-uuu uuuu	-uuu uuuu
WDTS	•	•	•	•	---- -111	---- -111	---- -111	---- -uuu
STATUS	•	•	•	•	--00 xxxx	--uu uuuu	--1u uuuu	--11 uuuu
INTC0	•				--00 -000	--00 -000	--00 -000	--uu -uuu
		•	•	•	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	•	•		•	-000 -000	-000 -000	-000 -000	-uuu -uuu
			•		-00- -00-	-00- -00-	-00- -00-	-uu- -uu-
MFIC	•	•	•	•	-000 -000	-000 -000	-000 -000	-uuu -uuu
TMR0	•	•	•	•	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0C	•	•	•	•	0000 1000	0000 1000	0000 1000	uuuu uuuu
TMR1		•	•	•	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR1C		•	•	•	0000 1---	0000 1---	0000 1---	uuuu u---
PA	•	•	•	•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	•	•	•	•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAWK	•	•	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PAPU	•	•	•	•	-000 0000	-000 0000	-000 0000	-uuu uuuu
PB	•				---- 1111	---- 1111	---- 1111	---- uuuu
		•	•		--11 1111	--11 1111	--11 1111	--uu uuuu
				•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBC	•				---- 1111	---- 1111	---- 1111	---- uuuu
		•	•		--11 1111	--11 1111	--11 1111	--uu uuuu
				•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBPU	•				---- 0000	---- 0000	---- 0000	---- uuuu
		•	•		--00 0000	--00 0000	--00 0000	--uu uuuu
				•	0000 0000	0000 0000	0000 0000	uuuu uuuu

Register	HT48R064G	HT48R065G	HT46R066G	HT48R0662G	Power-on Reset	$\overline{\text{RES}}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (Idle/Sleep)
PC	•				1111 --11	1111 --11	1111 --11	uuuu --uu
		•	•	•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PCC	•				1111 --11	1111 --11	1111 --11	uuuu --uu
		•	•	•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PCPU	•				0000 --00	0000 --00	0000 --00	uuuu --uu
		•	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PCWK			•		0000 0000	0000 0000	0000 0000	uuuu uuuu
PD			•		---- 1111	---- 1111	---- 1111	---- uuuu
				•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PDC			•		---- 1111	---- 1111	---- 1111	---- uuuu
				•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PDPU			•		---- 0000	---- 0000	---- 0000	---- uuuu
				•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PE				•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEC				•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEPU				•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PF				•	----- --11	----- --11	----- --11	----- --uu
PFC				•	----- --11	----- --11	----- --11	----- --uu
PFPU				•	----- --00	----- --00	----- --00	----- --uu
CTRL0	•		•		--0- 0000	--0- 0000	--0- 0000	--u- uuuu
		•			-00- 0000	-00- 0000	-00- 0000	-uu- uuuu
				•	-000 0000	-000 0000	-000 0000	-uuu uuuu
CTRL1	•	•	•	•	1000 1010	1000 1010	1000 1010	uuuu uuuu
CTRL2				•	00-- ---1	00-- ---1	00-- ---1	uu-- ---u
SCOMC		•	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PWM0				•	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
PWM1				•	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
CMP0C	•	•	•	•	-000 0000	-000 0000	-000 0000	-uuu uuuu
CMP1C	•	•	•	•	000- 0-00	000- 0-00	000- 0-00	uuu- u-uu
COPA0C	•	•	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
COPA1C	•	•	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
COPA2C	•	•	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
COPA3C	•	•	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
OPA0OC	•	•	•	•	0x00 1000	0x00 1000	0x00 1000	uuuu uuuu
OPA1OC	•	•	•	•	0x00 1000	0x00 1000	0x00 1000	uuuu uuuu

Note: "-" not implemented
"u" means "unchanged"
"x" means "unknown"

Input/Output Ports

Holtek microcontrollers offer considerable flexibility on their I/O ports. Most pins can have either an input or output designation under user program control. Additionally, as there are pull-high resistors and wake-up software configurations, the user is provided with an I/O structure to meet the needs of a wide range of application possibilities.

For input operation, these ports are non-latching, which means the inputs must be ready at the T2 rising edge of instruction "MOV A,[m]", where m denotes the port address. For output operation, all the data is latched and remains unchanged until the output latch is rewritten.

Pull-high Resistors

Many product applications require pull-high resistors for their switch inputs usually requiring the use of an external resistor. To eliminate the need for these external resistors, when configured as an input have the capability of being connected to an internal pull-high resistor. These pull-high resistors are selectable via a register known as PAPU, PBPU, PCPU, PDPU, PEPU and PFPU located in the Data Memory. The pull-high resistors are implemented using weak PMOS transistors. Note that pin PA7 does not have a pull-high resistor selection.

I/O Port Wake-up

If the HALT instruction is executed, the device will enter the Idle/Sleep Mode, where the system clock will stop resulting in power being conserved, a feature that is important for battery and other low-power applications. Various methods exist to wake-up the microcontroller, one of which is to change the logic condition on one of the PA0~PA7 pins from high to low. For the HT48R0662G device, a logic transition from high to low on one of the PC0~PC7 pins can also wake up the microcontroller if the corresponding wake-up function control is enabled. After a HALT instruction forces the microcontroller into entering the Idle/Sleep Mode, the processor will remain idle or in a low-power state until the logic condition of the selected wake-up pin on Port A or Port C changes from high to low. This function is especially suitable for applications that can be woken up via external switches. Note that pins PA0~PA7 or PC0~PC7 can be selected individually to have this wake-up feature using an internal register known as PAWK or PCWK, located in the Data Memory.

PAWK, PAC~PCC, PAPU~PCPU Registers – HT48R064G

Register Name	POR	Bit							
		7	6	5	4	3	2	1	0
PAWK	00H	PAWK7	PAWK6	PAWK5	PAWK4	PAWK3	PAWK2	PAWK1	PAWK0
PAC	FFH	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PAPU	00H	—	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PBC	0FH	—	—	—	—	PBC3	PBC2	PBC1	PBC0
PBPU	00H	—	—	—	—	PBPU3	PBPU2	PBPU1	PBPU0
PCC	F3H	PCC7	PCC6	PCC5	PCC4	—	—	PCC1	PCC0
PCPU	00H	PCPU7	PCPU6	PCPU5	PCPU4	—	—	PCPU1	PCPU0

"—" Unimplemented, read as "0"

PAWKn: PA wake-up function enable
0: disable
1: enable

PACn/PBCn/PCCn: I/O type selection
0: output
1: input

PAPUn/PBPUn/PCPUn: Pull-high function enable
0: disable
1: enable

PAWK, PAC~PCC, PAPU~PCPU Registers – HT48R065G

Register Name	POR	Bit							
		7	6	5	4	3	2	1	0
PAWK	00H	PAWK7	PAWK6	PAWK5	PAWK4	PAWK3	PAWK2	PAWK1	PAWK0
PAC	FFH	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PAPU	00H	—	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PBC	3FH	—	—	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PBPU	00H	—	—	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PCC	FFH	PCC7	PCC6	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0
PCPU	00H	PCPU7	PCPU6	PCPU5	PCPU4	PCPU3	PCPU2	PCPU1	PCPU0

"—" Unimplemented, read as "0"

PAWKn: PA wake-up function enable
0: disable
1: enable

PACn/PBCn/PCCn: I/O type selection
0: output
1: input

PAPUn/PBPUn/PCPUn: Pull-high function enable
0: disable
1: enable

PAWK, PAC~PDC, PAPU~PDPU, PCWK Registers – HT48R066G

Register Name	POR	Bit							
		7	6	5	4	3	2	1	0
PAWK	00H	PAWK7	PAWK6	PAWK5	PAWK4	PAWK3	PAWK2	PAWK1	PAWK0
PAC	FFH	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PAPU	00H	—	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PBC	3FH	—	—	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PBPU	00H	—	—	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PCWK	00H	PCWK7	PCWK6	PCWK5	PCWK4	PCWK3	PCWK2	PCWK1	PCWK0
PCC	FFH	PCC7	PCC6	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0
PCPU	00H	PCPU7	PCPU6	PCPU5	PCPU4	PCPU3	PCPU2	PCPU1	PCPU0
PDC	0FH	—	—	—	—	PDC3	PDC2	PDC1	PDC0
PDPU	00H	—	—	—	—	PDPU3	PDPU2	PDPU1	PDPU0

"—" Unimplemented, read as "0"

PAWKn, PCWKn: PA, PC wake-up function enable
0: disable
1: enable

PACn/PBCn/PCCn/PDCn: I/O type selection
0: output
1: input

PAPUn/PBPUn/PCPUn/PDPUn: Pull-high function enable
0: disable
1: enable

PAWK, PCWK, PAC~PFC, PAPU~PFPU Registers – HT48R0662G

Register Name	POR	Bit							
		7	6	5	4	3	2	1	0
PAWK	00H	PAWK7	PAWK6	PAWK5	PAWK4	PAWK3	PAWK2	PAWK1	PAWK0
PAC	FFH	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PAPU	00H	—	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PBC	FFH	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PBPU	00H	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PCWK	00H	PCWK7	PCWK6	PCWK5	PCWK4	PCWK3	PCWK2	PCWK1	PCWK0
PCC	FFH	PCC7	PCC6	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0
PCPU	00H	PCPU7	PCPU6	PCPU5	PCPU4	PCPU3	PCPU2	PCPU1	PCPU0
PDC	FFH	PDC7	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
PDPU	00H	PDPU7	PDPU6	PDPU5	PDPU4	PDPU3	PDPU2	PDPU1	PDPU0
PEC	FFH	PEC7	PEC6	PEC5	PEC4	PEC3	PEC2	PEC1	PEC0
PEPU	00H	PEPU7	PEPU6	PEPU5	PEPU4	PEPU3	PEPU2	PEPU1	PEPU0
PFC	03H	—	—	—	—	—	—	PFC1	PFC0
PFPU	00H	—	—	—	—	—	—	PFPU1	PFPU0

"—" Unimplemented, read as "0"

PAWKn/PCWKn: PA/PC wake-up function enable
0: disable
1: enable

PACn/PBCn/PCCn/PDCn/PECn/PFCn: I/O type selection
0: output
1: input

PAPUn/PBPUn/PCPUn/PDPUn/PEPUn/PFPUn: Pull-high function enable
0: disable
1: enable

I/O Port Control Registers

Each Port has its own control register, known as PAC, PBC, PCC, PDC, PEC, PFC which controls the input/output configuration. With this control register, each I/O pin with or without pull-high resistors can be reconfigured dynamically under software control. For the I/O pin to function as an input, the corresponding bit of the control register must be written as a "1". This will then allow the logic state of the input pin to be directly read by instructions. When the corresponding bit of the control register is written as a "0", the I/O pin will be setup as a CMOS output. If the pin is currently setup as an output, instructions can still be used to read the output register. However, it should be noted that the program will in fact only read the status of the output data latch and not the actual logic status of the output pin.

Pin-shared Functions

The flexibility of the microcontroller range is greatly enhanced by the use of pins that have more than one function. Limited numbers of pins can force serious design constraints on designers but by supplying pins with multi-functions, many of these difficulties can be overcome. For some pins, the chosen function of the multi-function I/O pins is set by configuration options while for others the function is set by application program control.

External Interrupt Input

The external interrupt pin, INT, is pin-shared with an I/O pin. To use the pin as an external interrupt input the correct bits in the INTC register must be programmed. The pin must also be setup as an input by setting the PAC3 bit in the Port Control Register. An internal pull-high resistor can be selected to be connected to this pin by the corresponding pull-high function enable control bit. Note that even if the pin is setup as an external interrupt input the I/O function still remains.

External Timer/Event Counter Input

The Timer/Event Counter pin TCn is pin-shared with I/O pins. For the shared pin to be used as the Timer/Event Counter input, the Timer/Event Counter n must be configured to be in the Event Counter or Pulse Width Capture Mode. This is achieved by setting the appropriate bits in the Timer/Event Counter Control Register. The pins must also be setup as inputs by setting the appropriate bit in the Port Control Register. Pull-high resistor function for the TCn pin can also be selected using the port pull-high resistor register. Note that even if the pin is setup as an external timer input the I/O function still remains.

PFD Output

The PFD function output is pin-shared with an I/O pin. The output function of this pin is chosen using the CTRL0 register. Note that the corresponding bit of the port control register, must setup the pin as an output to enable the PFD output. If the port control register has setup the pin as an input, then the pin will function as a normal logic input with the usual pull-high selection, even if the PFD function has been selected.

PWM Outputs

The PWM function whose outputs are pin-shared with I/O pins. The PWM output functions are chosen using the CTRL0 register. Note that the corresponding bit of the port control registers, for the output pin, must setup the pin as an output to enable the PWM output. If the pins are setup as inputs, then the pin will function as a normal logic input with the usual pull-high selections, even if the PWM registers have enabled the PWM function.

SCOM Driver Pins

Pins PB0~PB3 on Port B for the HT48R065G, HT48R066G and HT48R0662G devices can be used as LCD COM driver pins. This function is controlled using the register which will generate the necessary 1/2 bias signals on these four pins.

Pin Remapping Configuration – HT48R0662G

The pin remapping function for the HT48R0662G device enables the function pins INT, TC0, TC1 and PFD to be located on different port pins. It is important not to confuse the Pin Remapping function with the Pin-shared function; these two functions have no interdependence.

The PCFG1 and PCFG0 bits in the CTRL2 register allow the four function pins INT, TC0, TC1 and PFD to be remapped to different port pins. After power up, this bit will be reset to zero, which will define the default port pins to which these functions will be mapped. Changing these bits will move the functions to other port pins.

Examination of the pin names on the package diagrams will reveal that some pin function names are repeated, this indicates a function pin that can be remapped to other port pins. If the pin name is bracketed, then this indicates its alternative location. Pin name without brackets indicates its default location which is the condition after Power-on.

PCFG [1:0] Bits Status				
PCFG [1:0] Bit	00	01	10	11
Pin Mapping	PFD/PA1 TC0/PA2 INT/PA3 TC1/PA4	PFD/PC5 TC0/PC4 INT/PC3 TC1/PC2	PFD/PB0 TC0/PB1 INT/PB2 TC1/PB3	PFD/PE0 TC0/PE1 INT/PE2 TC1/PE3

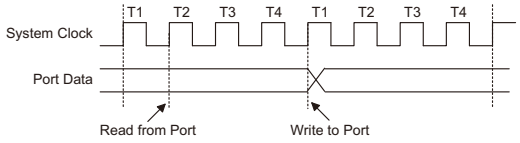
Pin Remapping

I/O Pin Structures

The diagrams illustrate the I/O pin internal structures. As the exact logical construction of the I/O pin may differ from these drawings, they are supplied as a guide only to assist with the functional understanding of the I/O pins.

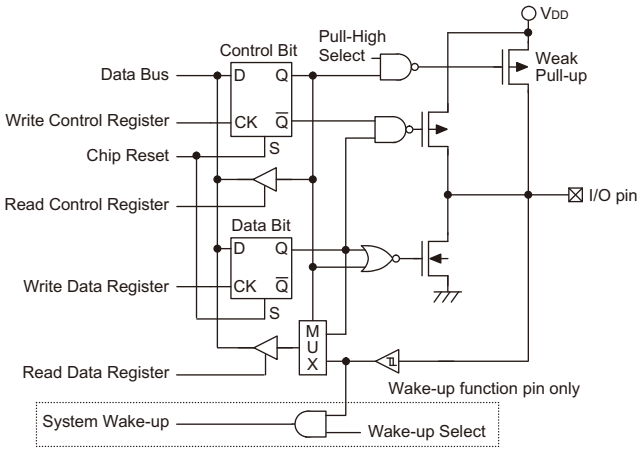
Programming Considerations

Within the user program, one of the first things to consider is port initialisation. After a reset, the I/O data register and I/O port control register will be set high. This means that all I/O pins will default to an input state, the level of which depends on the other connected circuitry and whether pull-high options have been selected. If the port control registers, are then programmed to setup some pins as outputs, these output pins will have an initial high output value unless the associated port data register is first programmed. Selecting which pins are inputs and which are outputs can be achieved byte-wide by loading the correct value into the port control register or by programming individual bits in the port control register using the "SET [m].i" and "CLR [m].i" instructions. Note that when using these bit control instructions, a read-modify-write operation takes place. The microcontroller must first read in the data on the entire port, modify it to the required new bit values and then rewrite this data back to the output ports.

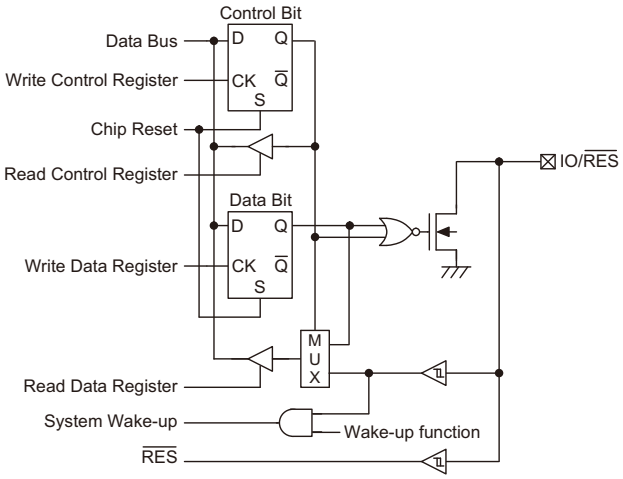


Read Modify Write Timing

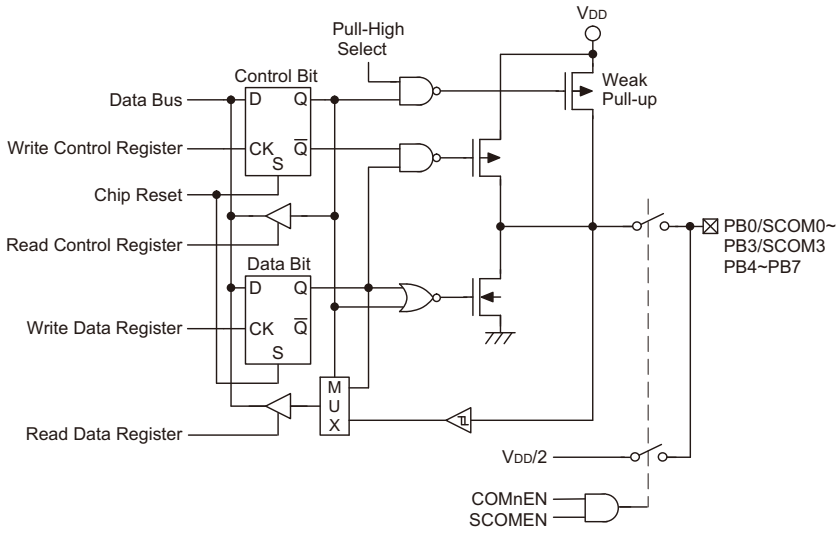
Pins on PA0 to PA7 for all the devices or PC0 to PC7 for only the HT48R0662G device each have a wake-up function, selected via the PAWK or PCWK register. When the device is in the Idle/Sleep Mode, various methods are available to wake the device up. One of these is a high to low transition of any of these pins. Single or multiple pins on Port A or Port C can be setup to have this function.



Generic Input/Output Ports



$\overline{\text{RES}}$ NMOS Input/Output Port



PB Input/Output Port

Timer/Event Counters

The provision of timers form an important part of any microcontroller, giving the designer a means of carrying out time related functions. The devices contain from one to three count-up timer of 8-bit capacity. As the timers have three different operating modes, they can be configured to operate as a general timer, an external event counter or as a pulse width capture device. The provision of an internal prescaler to the clock circuitry on gives added range to the timers.

There are two types of registers related to the Timer/Event Counters. The first is the register that contains the actual value of the timer and into which an initial value can be preloaded. Reading from this register retrieves the contents of the Timer/Event Counter. The second type of associated register is the Timer Control Register which defines the timer options and determines how the timer is to be used. The device can have the timer clock configured to come from the internal clock source. In addition, the timer clock source can also be configured to come from an external timer pin.

Configuring the Timer/Event Counter Input Clock Source

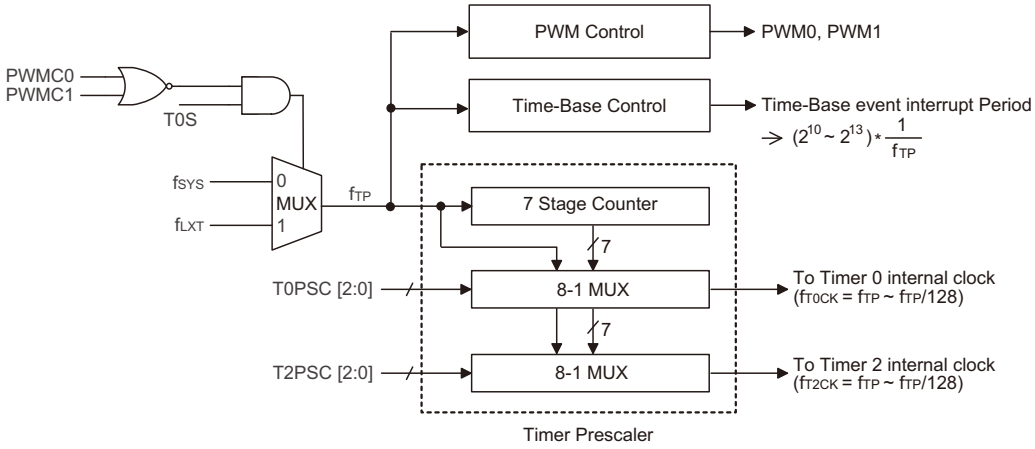
The Timer/Event Counter clock source can originate from various sources, an internal clock or an external pin. The internal clock source is used when the timer is in the timer mode or in the pulse width capture mode. For the Timer/Event Counter 0, this internal clock source is first divided by a prescaler, the division ratio of which is conditioned by the Timer Control Register bits T0PSC0~T0PSC2. The internal clock source can be derived from the system clock f_{SYS} or the LXT oscillator for Timer/Event Counter 0 or from the instruction clock $f_{SYS}/4$ or the LXT oscillator for Timer/Event Counter 1 selected by the clock selection bit TnS in the control register TMRnC.

An external clock source is used when the Timer/Event Counter is in the event counting mode, the clock source being provided on an external timer pin TCn. Depending upon the condition of the TnEG bit, each high to low, or low to high transition on the external timer pin will increment the counter by one.

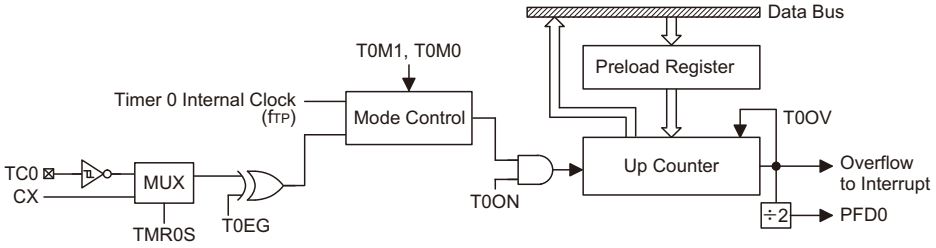
Timer Registers – TMR0, TMR1

The timer register is a special function register located in the Special Purpose Data Memory and is the place where the actual timer value is stored and the register is known as TMRn. The value in the timer register increases by one each time an internal clock pulse is received or an external transition occurs on the external timer pin. The timer will count from the initial value loaded by the preload register to the full count of FFH at which point the timer overflows and an internal interrupt signal is generated. The timer value will be reset with the initial preload register value and continue counting.

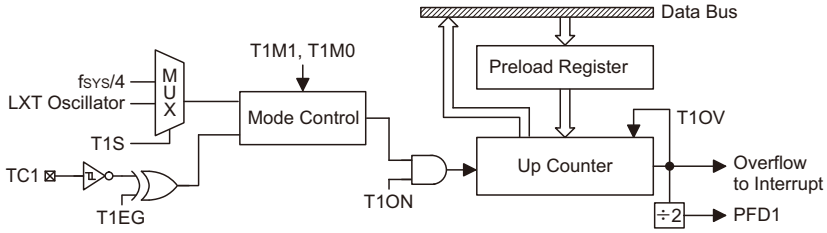
To achieve a maximum full range count of FFH, the preload register must first be cleared to all zeros. It should be noted that after power-on, the preload register will be in an unknown condition. Note that if the Timer/Event Counter is switched off and data is written to its preload register, this data will be immediately written into the actual timer register. However, if the Timer/Event Counter is enabled and counting, any new data written into the preload data register during this period will remain in the preload register and will only be written into the timer register the next time an overflow occurs.



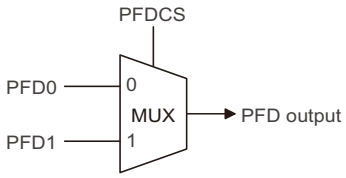
Clock Structure for Timer/PWM/Time Base



8-bit Timer/Event Counter 0 Structure



8-bit Timer/Event Counter 1 Structure



HT48R0662G PFD Clock Source

Note: If PWM0/PWM1 is enabled, then f_{TP} comes from f_{SYS} and the T0S bit will have no effect.

Timer Control Registers – TMR0C, TMR1C

The flexible features of the Holtek microcontroller Timer/Event Counters enable them to operate in three different modes, the options of which are determined by the contents of their respective control register.

The Timer Control Register is known as TMRnC. It is the Timer Control Register together with its corresponding timer register that control the full operation of the Timer/Event Counter. Before the timer can be used, it is essential that the Timer Control Register is fully programmed with the right data to ensure its correct operation, a process that is normally carried out during program initialisation.

To choose which of the three modes the timer is to operate in, either in the timer mode, the event counting mode or the pulse width capture mode, bits 7 and 6 of the Timer Control Register, which are known as the bit pair TnM1/TnM0, must be set to the required logic levels. The timer-on bit, which is bit 4 of the Timer Control Register and known as TnON, provides the basic on/off control of the respective timer. Setting the bit high allows the counter to run, clearing the bit stops the counter. Bits 0~2 of the Timer Control Register determine the division ratio of the input clock prescaler. The prescaler bit settings have no effect if an external clock source is used. If the timer is in the event count or pulse width capture mode, the active transition edge level type is selected by the logic level of bit 3 of the Timer Control Register which is known as TnEG. The TnS bit selects the internal clock source if used.

TMR0C Register

Bit	7	6	5	4	3	2	1	0
Name	T0M1	T0M0	T0S	T0ON	T0EG	T0PSC2	T0PSC1	T0PSC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	1	0	0	0

Bit 7,6 **T0M1, T0M0**: Timer0 operation mode selection
00: no mode available
01: event counter mode
10: timer mode
11: pulse width capture mode

Bit 5 **T0S**: timer clock source
0: f_{SYS}
1: LXT oscillator
T0S selects the clock source for f_{TP} which is provided for Timer 0, the Time-Base and the PWM. If the PWM is enabled, then f_{SYS} will be selected, overriding the T0S selection.

Bit 4 **T0ON**: Timer/event counter counting enable
0: disable
1: enable

Bit 3 **T0EG**:
Event counter active edge selection
0: count on raising edge
1: count on falling edge
Pulse Width Capture active edge selection
0: start counting on falling edge, stop on rasing edge
1: start counting on raising edge, stop on falling edge

Bit 2~0 **T0PSC2, T0PSC1, T0PSC0**: Timer prescaler rate selection
Timer internal clock=
000: f_{TP}
001: $f_{TP}/2$
010: $f_{TP}/4$
011: $f_{TP}/8$
100: $f_{TP}/16$
101: $f_{TP}/32$
110: $f_{TP}/64$
111: $f_{TP}/128$

TMR1C Register

Bit	7	6	5	4	3	2	1	0
Name	T1M1	T1M0	T1S	T1ON	T1EG	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—
POR	0	0	0	0	1	—	—	—

Bit 7,6 **T1M1, T1M0**: Timer 1 Operation mode selection
00: no mode available
01: event counter mode
10: timer mode
11: pulse width capture mode

Bit 5 **T1S**: timer clock source
0: $f_{SYS}/4$
1: LXT oscillator

Bit 4 **T1ON**: Timer/event counter counting enable
0: disable
1: enable

Bit 3 **T1EG**:
Event counter active edge selection
0: count on raising edge
1: count on falling edge
Pulse Width Capture active edge selection
0: start counting on falling edge, stop on rasing edge
1: start counting on raising edge, stop on falling edge

Bit 2~0 unimplemented, read as ″0″

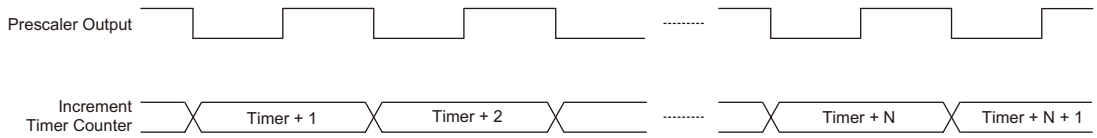
Timer Mode

In this mode, the Timer/Event Counter can be utilised to measure fixed time intervals, providing an internal interrupt signal each time the Timer/Event Counter overflows. To operate in this mode, the Operating Mode Select bit pair, TnM1/TnM0, in the Timer Control Register must be set to the correct value as shown.

Control Register Operating Mode Select Bits for the Timer Mode

Bit7	Bit6
1	0

In this mode the internal clock is used as the timer clock. The timer input clock source is f_{SYS}, $f_{SYS}/4$ or the LXT oscillator depending upon whether the Timer/Event Counter 0 or Timer/Event Counter 1 is selected. For Timer/Event Counter 0, the timer clock source is further divided by a prescaler, the value of which is determined by the bits T0PSC2~T0PSC0 in the Timer Control Register TMR0C. The timer-on bit, TnON must be set high to enable the timer to run. Each time an internal clock high to low transition occurs, the timer increments by one; when the timer is full and overflows, an interrupt signal is generated and the timer will reload the value already loaded into the preload register and continue counting. A timer overflow condition and corresponding internal interrupt is one of the wake-up sources, however, the internal interrupts can be disabled by ensuring that the TnE bits of the INTC0 register are reset to zero.



Timer Mode Timing Chart

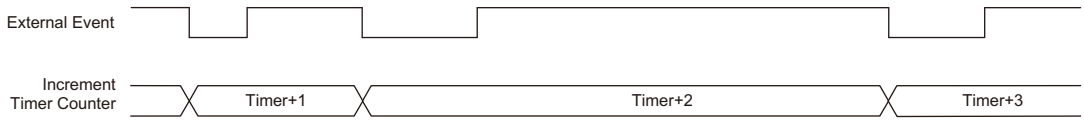
Event Counter Mode

In this mode, a number of externally changing logic events, occurring on the external timer TCn pin, can be recorded by the Timer/Event Counter. To operate in this mode, the Operating Mode Select bit pair, TnM1/TnM0, in the Timer Control Register must be set to the correct value as shown.

Control Register Operating Mode Select Bits for the Event Counter Mode	Bit7	Bit6
	0	1

In this mode, the external timer TCn pin, is used as the Timer/Event Counter clock source, however it is not divided by the internal prescaler. After the other bits in the Timer Control Register have been setup, the enable bit TnON, which is bit 4 of the Timer Control Register, can be set high to enable the Timer/Event Counter to run. If the Active Edge Select bit, TnEG, which is bit 3 of the Timer Control Register, is low, the Timer/Event Counter will increment each time the external timer pin receives a low to high transition. If the TnEG is high, the counter will increment each time the external timer pin receives a high to low transition. When it is full and overflows, an interrupt signal is generated and the Timer/Event Counter will reload the value already loaded into the preload register and continue counting. The interrupt can be disabled by ensuring that the Timer/Event Counter Interrupt Enable bit in the corresponding Interrupt Control Register, is reset to zero.

As the external timer pin is shared with an I/O pin, to ensure that the pin is configured to operate as an event counter input pin, two things have to happen. The first is to ensure that the Operating Mode Select bits in the Timer Control Register place the Timer/Event Counter in the Event Counting Mode, the second is to ensure that the port control register configures the pin as an input. It should be noted that in the event counting mode, even if the microcontroller is in the Idle/Sleep Mode, the Timer/Event Counter will continue to record externally changing logic events on the timer input TCn pin. As a result when the timer overflows it will generate a timer interrupt and corresponding wake-up source.



Event Counter Mode Timing Chart (TnEG=1)

Pulse Width Capture Mode

In this mode, the Timer/Event Counter can be utilised to measure the width of external pulses applied to the external timer pin. To operate in this mode, the Operating Mode Select bit pair, TnM1/TnM0, in the Timer Control Register must be set to the correct value as shown.

Control Register Operating Mode Select Bits for the Pulse Width Capture Mode	Bit7	Bit6
	1	1

In this mode the internal clock, f_{SYS}, $f_{SYS}/4$ or the LXT oscillator, is used as the internal clock determined by which Timer/Event Counter is selected to be used. The internal clock source for the Timer/Event Counter 0 is further divided by a prescaler, the value of which is determined by the Prescaler Rate Select bits named T0PSC2~T0PSC0, which are bits 2~0 in the Timer Control Register. After other bits in the Timer Control Register have been setup, the enable bit TnON, which is bit 4 of the Timer Control Register, can be set high to enable the Timer/Event Counter, however it will not actually start counting until an active edge is received on the external timer pin.

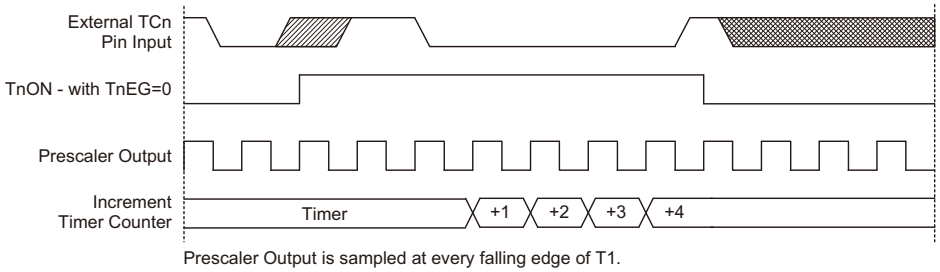
If the Active Edge Select bit TnEG, which is bit 3 of the Timer Control Register, is low, once a high to low transition has been received on the external timer pin, the Timer/Event Counter will start counting until the external timer pin returns to its original high level. At this point the enable bit will be automatically reset to zero and the Timer/Event Counter will stop counting. If the Active Edge Select bit is high, the Timer/Event Counter will begin counting once a low to high transition has been received on the external timer pin and stop counting when the external timer pin returns to its original

low level. As before, the enable bit will be automatically reset to zero and the Timer/Event Counter will stop counting. It is important to note that in the pulse width capture Mode, the enable bit is automatically reset to zero when the external control signal on the external timer pin returns to its original level, whereas in the other two modes the enable bit can only be reset to zero under program control.

The residual value in the Timer/Event Counter, which can now be read by the program, therefore represents the length of the pulse received on the TCn pin. As the enable bit has now been reset, any further transitions on the external timer pin will be ignored. The timer cannot begin further pulse width capture until the enable bit is set high again by the program. In this way, single shot pulse measurements can be easily made.

It should be noted that in this mode the Timer/Event Counter is controlled by logical transitions on the external timer pin and not by the logic level. When the Timer/Event Counter is full and overflows, an interrupt signal is generated and the Timer/Event Counter will reload the value already loaded into the preload register and continue counting. The interrupt can be disabled by ensuring that the Timer/Event Counter Interrupt Enable bit in the corresponding Interrupt Control Register, is reset to zero.

As the TCn pin is shared with an I/O pin, to ensure that the pin is configured to operate as a pulse width capture pin, two things have to happen. The first is to ensure that the Operating Mode Select bits in the Timer Control Register place the Timer/Event Counter in the pulse width capture Mode, the second is to ensure that the port control register configures the pin as an input.



Pulse Width Capture Mode Timing Chart (TnEG=0)

Prescaler

Bits T0PSC0~T0PSC2 of the TMR0C register can be used to define a division ratio for the internal clock source of the Timer/Event Counter enabling longer time out periods to be setup.

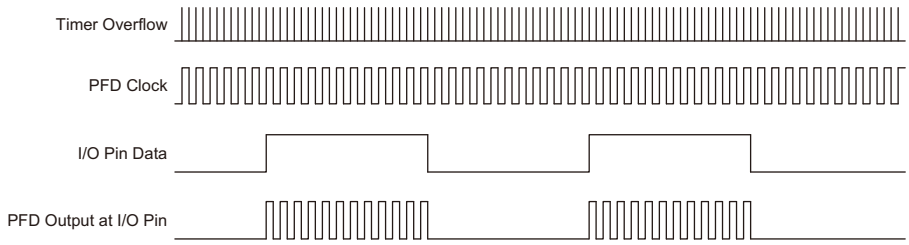
PFD Function

The Programmable Frequency Divider provides a means of producing a variable frequency output suitable for applications, such as piezo-buzzer driving or other interfaces requiring a precise frequency generator.

The Timer/Event Counter overflow signal is the clock source for the PFD function, which is controlled by PFDCS bit in CTRL0. For applicable devices the clock source can come from either Timer/Event Counter 0 or Timer/Event Counter 1. The output frequency is controlled by loading the required values into the timer prescaler and timer registers to give the required division ratio. The counter will begin to count-up from this preload register value until full, at which point an overflow signal is generated, causing both the PFD outputs to change state. The counter will then be automatically reloaded with the preload register value and continue counting-up.

If the CTRL0 register has selected the PFD function, then for PFD output to operate, it is essential for the corresponding Port control register, to setup the PFD pins as outputs. The corresponding I/O pin data bit must be set high to activate the PFD. The output data bits can be used as the on/off control bit for the PFD outputs. Note that the PFD outputs will all be low if the output data bit is cleared to zero.

Using this method of frequency generation, and if a crystal oscillator is used for the system clock, very precise values of frequency can be generated.



PFD Function

I/O Interfacing

The Timer/Event Counter, when configured to run in the event counter or pulse width capture mode, requires the use of an external timer pin for its operation. As this pin is a shared pin it must be configured correctly to ensure that it is setup for use as a Timer/Event Counter input pin. This is achieved by ensuring that the mode select bits in the Timer/Event Counter control register, select either the event counter or pulse width capture mode. Additionally the corresponding Port Control Register bit must be set high to ensure that the pin is setup as an input. Any pull-high resistor connected to this pin will remain valid even if the pin is used as a Timer/Event Counter input.
Programming Considerations

When configured to run in the timer mode, the internal system clock is used as the timer clock source and is therefore synchronised with the overall operation of the microcontroller. In this mode when the appropriate timer register is full, the microcontroller will generate an internal interrupt signal directing the program flow to the respective internal interrupt vector. For the pulse width capture mode, the internal system clock is also used as the timer clock source but the timer will only run when the correct logic condition appears on the external timer input pin. As this is an external event and not synchronised with the internal timer clock, the microcontroller will only see this external event when the next timer clock pulse arrives. As a result, there may be small differences in measured values requiring programmers to take this into account during programming. The same applies if the timer is configured to be in the event counting mode, which again is an external event and not synchronised with the internal system or timer clock.

When the Timer/Event Counter is read, or if data is written to the preload register, the clock is inhibited to avoid errors, however as this may result in a counting error, this should be taken into account by the programmer. Care must be taken to ensure that the timers are properly initialised before using them for the first time. The associated timer enable bits in the interrupt control register must be properly set otherwise the internal interrupt associated with the timer will remain inactive. The edge select, timer mode and clock source control bits in timer control register must also be correctly set to ensure the timer is properly configured for the required application. It is also important to ensure that an initial value is first loaded into the timer registers before the timer is switched on; this is because after power-on the initial values of the timer registers are unknown. After the timer has been initialised the timer can be turned on and off by controlling the enable bit in the timer control register.

When the Timer/Event Counter overflows, its corresponding interrupt request flag in the interrupt control register will be set. If the Timer/Event Counter interrupt is enabled this will in turn generate an interrupt signal. However irrespective of whether the interrupts are enabled or not, a Timer/Event Counter overflow will also generate a wake-up signal if the device is in a Power-down condition. This situation may occur if the Timer/Event Counter is in the Event Counting Mode and if the external signal continues to change state. In such a case, the Timer/Event Counter will continue to count these external events and if an overflow occurs the device will be woken up from its Power-down condition. To prevent such a wake-up from occurring, the timer interrupt request flag should first be set high before issuing the ″HALT″ instruction to enter the Idle/Sleep Mode.

Timer Program Example

The program shows how the Timer/Event Counter registers are setup along with how the interrupts are enabled and managed. Note how the Timer/Event Counter is turned on, by setting bit 4 of the Timer Control Register. The Timer/Event Counter can be turned off in a similar way by clearing the same bit. This example program sets the Timer/Event Counters to be in the timer mode, which uses the internal system clock as their clock source.

Timer Programming Example

```
org     04h                 ; external interrupt vector

org     08h                 ; Timer Counter 0 interrupt vector
jmp     tmr0int             ; jump here when Timer 0 overflows
:       :
org     20h                 ; main program
:       :
;internal Timer 0 interrupt routine
tmr0int:
:
; Timer 0 main program placed here
:

:
begin:
;setup Timer 0 registers
mov     a,09bh              ; setup Timer 0 preload value
mov     tmr0,a
mov     a,081h              ; setup Timer 0 control register
mov     tmr0c,a             ; timer mode and prescaler set to /2
;setup interrupt register
mov     a,00dh              ; enable master interrupt and both timer interrupts
mov     intc0,a
:       :
set tmr0c.4                 ; start Timer 0
:       :
```

Time Base

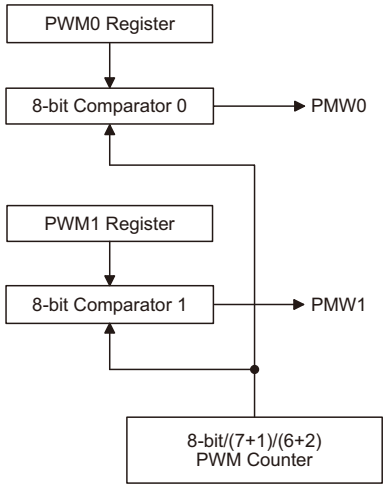
The device includes a Time Base function which is used to generate a regular time interval signal.

The Time Base time interval magnitude is determined using an internal 13 stage counter sets the division ratio of the clock source. This division ratio is controlled by both the TBSEL0 and TBSEL1 bits in the CTRL1 register. The clock source is selected using the T0S bit in the TMR0C register.

When the Time Base time out, a Time Base interrupt signal will be generated. It should be noted that as the Time Base clock source is the same as the Timer/Event Counter clock source, care should be taken when programming.

Pulse Width Modulator

The series of devices includes up to 2 8-bit PWM outputs. Useful for the applications such as motor speed control, the PWM function provides outputs with a fixed frequency but with a duty cycle that can be varied by setting particular values into the corresponding PWM register.



PWM Block Diagram

Device	Channels	Mode	Pins	Registers
HT48R0662G	2	6+2 7+1	PD0 PD1	PWM0 PWM1

PWM Operation

The register, known as PWMn and located in the Data Memory, is assigned to each Pulse Width Modulator channel. It is here that the 8-bit value, which represents the overall duty cycle of one modulation cycle of the output waveform, should be placed. To increase the PWM modulation frequency, each modulation cycle is subdivided into two or four individual modulation subsections, known as the 7+1 mode or 6+2 mode respectively. The required mode and the on/off control for each PWM channel is selected using the CTRL0 register. Note that when using the PWM, it is only necessary to write the required value into the PWMn register and select the required mode setup and on/off control using the CTRL0 register, the subdivision of the waveform into its sub-modulation cycles is implemented automatically within the microcontroller hardware. The PWM clock source is the system clock f_{SYS}. This method of dividing the original modulation cycle into a further 2 or 4 sub-cycles enable the generation of higher PWM frequencies which allow a wider range of applications to be served. The difference between what is known as the PWM cycle frequency and the PWM modulation frequency should be understood. As the PWM clock is the system clock, f_{SYS}, and as the PWM value is 8-bits wide, the overall PWM cycle frequency is $f_{SYS}/256$. However, when in the 7+1 mode of operation the PWM modulation frequency will be $f_{SYS}/128$, while the PWM modulation frequency for the 6+2 mode of operation will be $f_{SYS}/64$.

PWM Modulation	PWM Cycle Frequency	PWM Cycle Duty
$f_{SYS}/64$ for (6+2) bits mode $f_{SYS}/128$for (7+1) bits mode	$f_{SYS}/256$	[PWM]/256

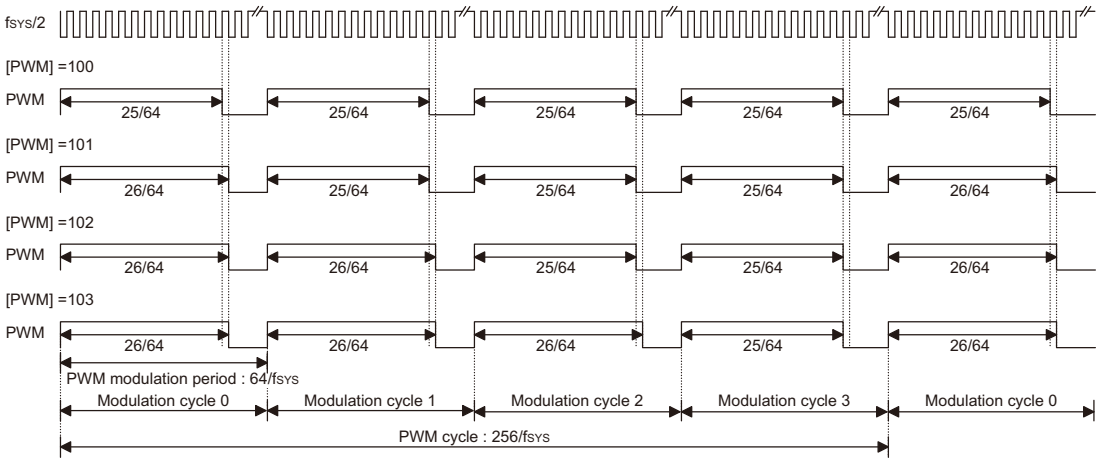
6+2 PWM Mode

Each full PWM cycle, as it is controlled by an 8-bit PWM register, has 256 clock periods. However, in the 6+2 PWM mode, each PWM cycle is subdivided into four individual sub-cycles known as modulation cycle 0 ~ modulation cycle 3, denoted as i in the table. Each one of these four sub-cycles contains 64 clock cycles. In this mode, a modulation frequency increase of four is achieved. The 8-bit PWM register value, which represents the overall duty cycle of the PWM waveform, is divided into two groups. The first group which consists of bit2~bit7 is denoted here as the DC value. The second group which consists of bit0~bit1 is known as the AC value. In the 6+2 PWM mode, the duty cycle value of each of the four modulation sub-cycles is shown in the following table.

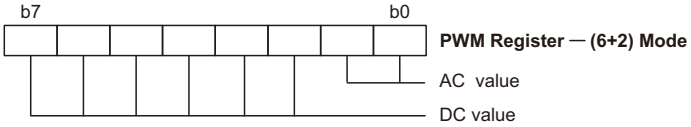
Parameter	AC (0~3)	DC (Duty Cycle)
Modulation cycle i (i=0~3)	i<AC	$\frac{DC+1}{64}$
	i≥AC	$\frac{DC}{64}$

6+2 Mode Modulation Cycle Values

The following diagram illustrates the waveforms associated with the 6+2 mode of PWM operation. It is important to note how the single PWM cycle is subdivided into 4 individual modulation cycles, numbered from 0~3 and how the AC value is related to the PWM value.



6+2 PWM Mode



PWM Register for 6+2 Mode

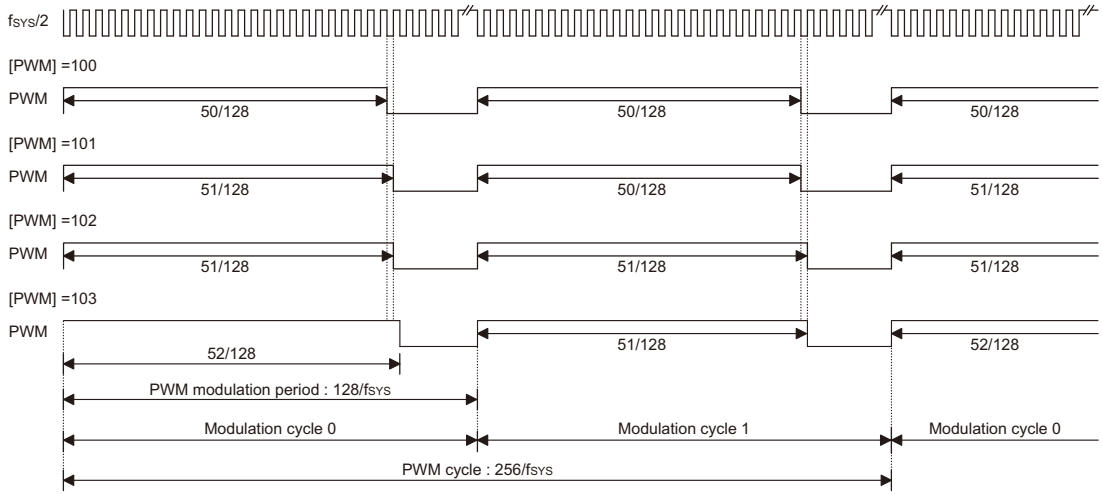
7+1 PWM Mode

Each full PWM cycle, as it is controlled by an 8-bit PWM register, has 256 clock periods. However, in the 7+1 PWM mode, each PWM cycle is subdivided into two individual sub-cycles known as modulation cycle 0 ~ modulation cycle 1, denoted as i in the table. Each one of these two sub-cycles contains 128 clock cycles. In this mode, a modulation frequency increase of two is achieved. The 8-bit PWM register value, which represents the overall duty cycle of the PWM waveform, is divided into two groups. The first group which consists of bit1~bit7 is denoted here as the DC value. The second group which consists of bit0 is known as the AC value. In the 7+1 PWM mode, the duty cycle value of each of the two modulation sub-cycles is shown in the following table.

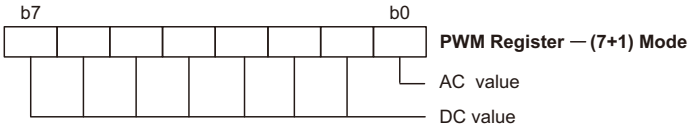
Parameter	AC (0~1)	DC (Duty Cycle)
Modulation cycle i (i=0~1)	i<AC	$\frac{DC+1}{128}$
	i≥AC	$\frac{DC}{128}$

7+1 Mode Modulation Cycle Values

The following diagram illustrates the waveforms associated with the 7+1 mode PWM operation. It is important to note how the single PWM cycle is subdivided into 2 individual modulation cycles, numbered 0 and 1 and how the AC value is related to the PWM value.



7+1 PWM Mode



PWM Register for 7+1 Mode

PWM Output Control

The PWM outputs are pin-shared with the I/O pins PA4, PD0 and PD3 respectively depending upon the selected device. To operate as a PWM output and not as an I/O pin, the correct bits must be set in the CTRL0 register. A zero value must also be written to the corresponding I/O Port Control bit to ensure that the corresponding PWM output pin is setup as an output. After these two initial steps have been carried out, and of course after the required PWM value has been written into the PWMn register, writing a high value to the corresponding I/O Output Data bit will enable the PWM data to appear on the pin. Writing a zero value will disable the PWM output function and force the output low. In this way, the Port data output registers can be used as an on/off control for the PWM function. Note that if the CTRL0 register has selected the PWM function, but a high value has been written to its corresponding I/O Port Control bit to configure the pin as an input, then the pin can still function as a normal input line, with pull-high resistor options.

PWM Programming Example

The following sample program shows how the PWM0 output is setup and controlled.

```
mov    a,64h              ; setup PWM value of decimal 100
mov    pwm0,a
set    ctrl0.5            ; select the 7+1 PWM mode
set    ctrl0.3            ; select pin PA7 to have a PWM function
clr    pac.7              ; setup pin PA7 as an output
set    pa.7               ; enable the PWM output
:      :
clr    pa.7               ; disable the PWM output_ pin
                          ; PA7 forced low
```

Operational Amplifiers

There are two fully integrated Operational Amplifiers in these devices, OPA0 and OPA1. These OPAs can be used for user specified analog signal processing. The OPAs can be disabled or enabled entirely under software control using internal registers. With specific control registers, some OPA related applications can be easily implemented, such as Unity Gain Buffer, Non-Inverting Amplifier, Inverting Amplifier and various kinds of filters, etc.

Comparator & Operational Amplifier Registers

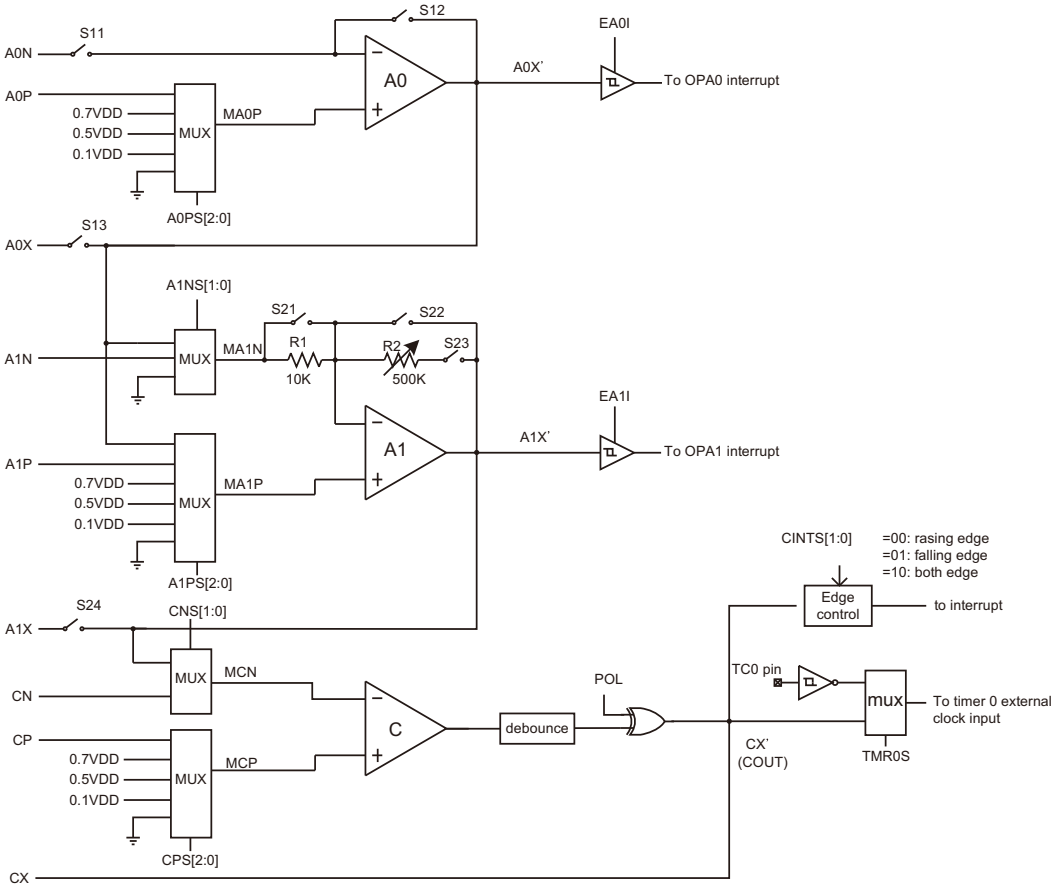
The internal Operational Amplifiers are fully under the control of internal registers, COPA0C, COPA1C, COPA2C, COPA3C, OPA0OC and OPA1OC. These registers control the enable/disable function, input path selection, gain control, polarity and calibration function.

Operational Amplifier Operation

The advantages of multiple switches and input path options, various reference voltage selection, up to 8 kinds of internal software gain control, output with interrupt function, offset reference voltage calibration function and power down control for low power consumption enhance the flexibility of these two OPAs to suit a wide range of application possibilities.

Note that the EA0I, EA1I interrupt control bits should be set to ″0″ before entering halt mode for power saving.

The following block diagram illustrates the main functional blocks of the OPAs and Comparator in this device.



COPA0C Register

Bit	7	6	5	4	3	2	1	0
Name	A0PS2	A0PS1	A0PS0	CPS2	CPS1	CPS0	CNS1	CNS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~5 **A0PS2~A0PS0**: OPA0 Non-inverting input signal selection bits
000: A0P pin
001: $0.7V_{DD}$
010: $0.5V_{DD}$
011: $0.1V_{DD}$
100: V_{SS}
101~111: undefined

Bit 4~2 **CPS2~CPS0**: Comparator Non-inverting input signal selection bits
000: CP pin
001: $0.7V_{DD}$
010: $0.5V_{DD}$
011: $0.1V_{DD}$
100: V_{SS}
101~111: undefined

Bit 1~0 **CNS1~CNS0**: Comparator Inverting input signal selection bits
00: CN pin
01: A1X
10: V_{SS}
11: undefined

COPA1C Register

Bit	7	6	5	4	3	2	1	0
Name	A1G2	A1G1	A1G0	A1PS2	A1PS1	A1PS0	A1NS1	A1NS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~5 **A1G2~A1G0**: OPA1 Gain control bits
000: 6.25
001: 12.50
010: 18.75
011: 25.00
100: 31.25
101: 37.50
110: 43.75
111: 50.00

Bit 4~2 **A1PS2~A1PS0**: OPA1 Non-inverting input signal selection bits
000: A1P pin
001: $0.7V_{DD}$
010: $0.5V_{DD}$
011: $0.1V_{DD}$
100: V_{SS}
101: A0X, the OPA0 internal output
110~111: undefined

Bit 1~0 **A1NS1~A1NS0**: OPA1 Inverting input signal selection bits
00: A1N pin
01: A0X, the OPA0 internal output
10: V_{SS}
11: undefined

COPA2C Register

Bit	7	6	5	4	3	2	1	0
Name	S24	S23	S22	S21	S13	S12	S11	CXC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **S24**: Switch S24 on/off control bit
0: Off
1: On

Bit 6 **S23**: Switch S23 on/off control bit
0: Off
1: On

Bit 5 **S22**: Switch S22 on/off control bit
0: Off
1: On

Bit 4 **S21**: Switch S21 on/off control bit
0: Off
1: On

Bit 3 **S13**: Switch S13 on/off control bit
0: Off
1: On

Bit 2 **S12**: Switch S12 on/off control bit
0: Off
1: On

Bit 1 **S11**: Switch S11 on/off control bit
0: Off
1: On

Bit 0 **CXC**: Comparator output pin CX enable control bit
0: I/O pin or other pin-shared functional pin
1: CX output pin (I/O pull-high disabled)

COPA3C Register

Bit	7	6	5	4	3	2	1	0
Name	A1XC	A1PC	A1NC	A0XC	A0PC	A0NC	CPC	CNC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **A1XC**: OPA1 output pin A1X enable control bit
0: I/O pin or other pin-shared functional pin
1: A1X output pin (I/O pull-high disabled)

Bit 6 **A1PC**: OPA1 non-inverting input pin A1P enable control bit
0: I/O pin or other pin-shared functional pin
1: A1P input pin (I/O pull-high disabled)

Bit 5 **A1NC**: OPA1 inverting input pin A1N enable control bit
0: I/O pin or other pin-shared functional pin
1: A1N input pin (I/O pull-high disabled)

Bit 4 **A0XC**: OPA0 output pin A0X enable control bit
0: I/O pin or other pin-shared functional pin
1: A0X output pin (I/O pull-high disabled)

Bit 3 **A0PC**: OPA0 non-inverting input pin A0P enable control bit
0: I/O pin or other pin-shared functional pin
1: A0P input pin (I/O pull-high disabled)

Bit 2 **A0NC**: OPA0 inverting input pin A0N enable control bit
0: I/O pin or other pin-shared functional pin
1: A0N input pin (I/O pull-high disabled)

Bit 1 **CPC**: Comparator non-inverting input pin CP enable control bit
0: I/O pin or other pin-shared functional pin
1: CP input pin (I/O pull-high disabled)

Bit 0 **CNC**: Comparator inverting input pin CN enable control bit
0: I/O pin or other pin-shared functional pin
1: CN input pin (I/O pull-high disabled)

OPA0OC Register

Bit	7	6	5	4	3	2	1	0
Name	A0EN	A0OP	A0OFM	A0RS	A0OF3	A0OF2	A0OF1	A0OF0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **A0EN**: Operational Amplifier 0 enable control bit
0: disable
1: enable

Bit 6 **A0OP**: Operational Amplifier 0 output; positive logic. This bit is read only bit.

Bit 5 **A0OFM**: Operational Amplifier 0 normal mode or input offset voltage cancellation mode selection bit
0: Operational Amplifier 0 normal mode
1: input offset voltage cancellation mode

Bit 4 **A0RS**: Operational Amplifier 0 input offset voltage cancellation reference input selection bit
0: Operational Amplifier A0N as the reference input
1: Operational Amplifier A0P as the reference input

Bit 3~0 **A0OF3~A0OF0**: Operational Amplifier 0 input offset voltage cancellation control bits

OPA1OC Register

Bit	7	6	5	4	3	2	1	0
Name	A1EN	A1OP	A1OFM	A1RS	A1OF3	A1OF2	A1OF1	A1OF0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **A1EN**: Operational Amplifier 1 enable control bit
0: disable
1: enable

Bit 6 **A1OP**: Operational Amplifier 1 output; positive logic. This bit is read only bit.

Bit 5 **A1OFM**: Operational Amplifier 1 normal mode or input offset voltage cancellation mode selection bit
0: Operational Amplifier 1 normal mode
1: input offset voltage cancellation mode

Bit 4 **A1RS**: Operational Amplifier 1 input offset voltage cancellation reference input selection bit
0: Operational Amplifier A1N as the reference input
1: Operational Amplifier A1P as the reference input

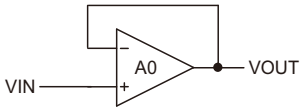
Bit 3~0 **A1OF3~A1OF0**: Operational Amplifier 1 input offset voltage cancellation control bits

Operational Amplifier Application Example

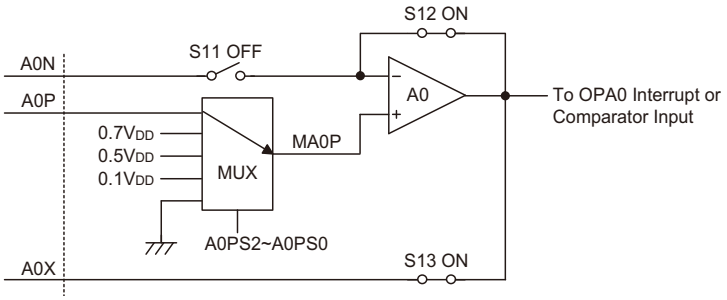
The OPAs can be connected to work with each other or standalone as shown in the block diagram. With the software controlled Switch and MUX, the OPAs can be connected to form various OPA related applications, such as, Unity Gain Buffer, Non-Inverting Amplifier, Inverting Amplifier, Integrators, Differential Amplifier, Low-Pass filter, High-Pass filter and Band-Pass filter,etc. The following diagrams show the interconnection and settings between the OPAs to implement these applications. The following examples are however only for reference.

Unity Gain Buffer

- Example



- Implementation connection



- Unity Gain Buffer Switch Setup

Bit	7	6	5	4	3	2	1	0
OPA2C	S24	S23	S22	S21	S13	S12	S11	CXC
Setup value	x	x	x	x	1	1	0	x

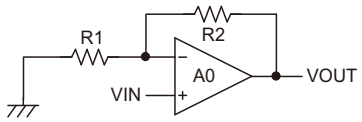
"x" don't care

Bit	7	6	5	4	3	2	1	0
OPA0C	A0PS2	A0PS1	A0PS0	CPS2	CPS1	CPS0	CNS1	CNS0
Setup value	0	0	0	0	0	0	0	0

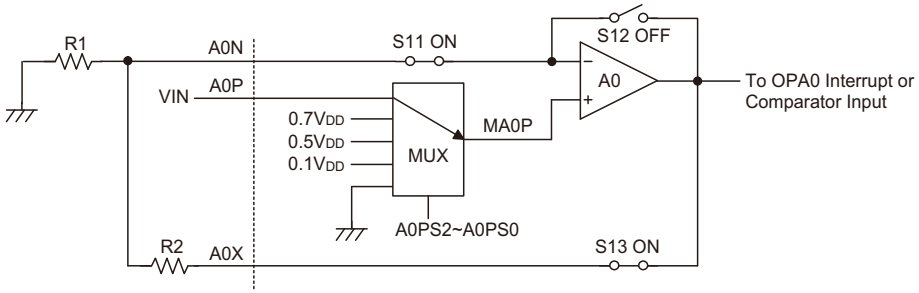
Switch control bits options:
S11: OFF
S12: ON
S13: ON
A0PS[2:0]: 000

Non-Inverting Amplifier

- Example



- Implementation connection



- Non-Inverting Amplifier Switch Setup

Bit	7	6	5	4	3	2	1	0
OPA2C	S24	S23	S22	S21	S13	S12	S11	CXC
Setup value	x	x	x	x	1	0	1	x

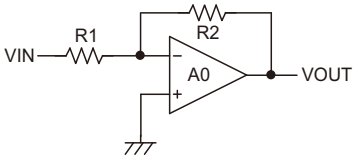
″x″ don't care

Bit	7	6	5	4	3	2	1	0
OPA0C	A0PS2	A0PS1	A0PS0	CPS2	CPS1	CPS0	CNS1	CNS0
Setup value	0	0	0	0	0	0	0	0

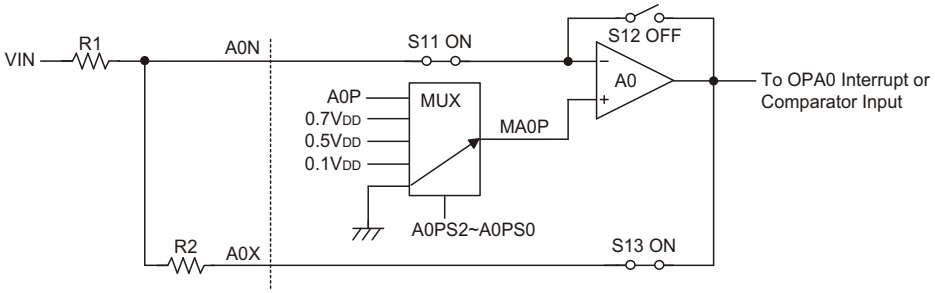
Switch control bits options:
S11: ON
S12: OFF
S13: ON
A0PS[2:0]: 000

Inverting Amplifier

- Example



- Implementation connection



- Inverting Amplifier Switch Setup

Bit	7	6	5	4	3	2	1	0
OPA2C	S24	S23	S22	S21	S13	S12	S11	CXC
Setup value	x	x	x	x	1	0	1	x

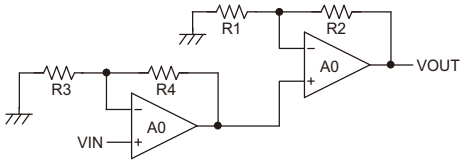
″x″ don't care

Bit	7	6	5	4	3	2	1	0
OPA0C	A0PS2	A0PS1	A0PS0	CPS2	CPS1	CPS0	CNS1	CNS0
Setup value	1	0	0	0	0	0	0	0

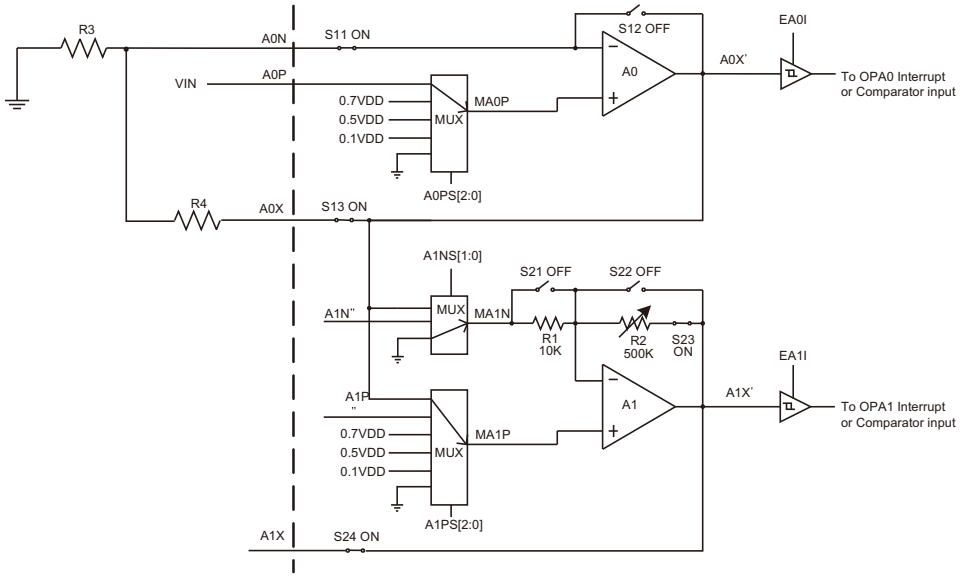
Switch control bits options:
S11: ON
S12: OFF
S13: ON
A0PS[2:0]: 100

Two-Stage Non-Inverting Amplifier

- Example



- Implementation connection



- Two-Stage Non-Inverting Amplifier Switch Setup

Bit	7	6	5	4	3	2	1	0
OPA2C	S24	S23	S22	S21	S13	S12	S11	CXC
Setup value	1	1	0	0	1	0	1	x

"x" don't care

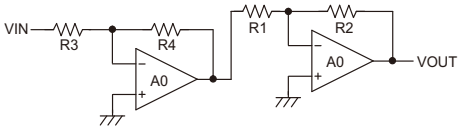
Bit	7	6	5	4	3	2	1	0
OPA0C	A0PS2	A0PS1	A0PS0	CPS2	CPS1	CPS0	CNS1	CNS0
Setup value	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
OPA1C	A1G2	A1G1	A1G0	A1PS2	A1PS1	A1PS0	A1NS1	A1NS0
Setup value	0	0	0	1	0	1	1	0

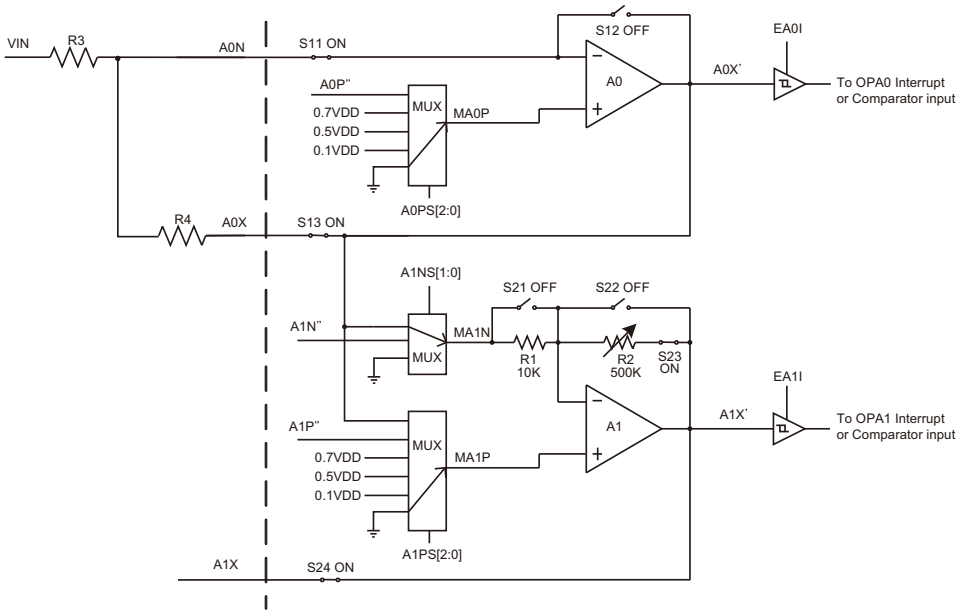
Switch control bits options:
S11: ON
S12: OFF
S13: ON
S21: OFF
S22: OFF
S23: ON
S24: ON
A0PS[2:0]: 000
A1PS[2:0]: 101
A1NS[1:0]: 10
A1G[2:0]: User define OPA1 Gain control

Two-Stage Inverting Amplifier

- Example



- Implementation connection



- Two-Stage Inverting Amplifier Switch Setup

Bit	7	6	5	4	3	2	1	0
OPA2C	S24	S23	S22	S21	S13	S12	S11	CXC
Setup value	1	1	0	0	1	0	1	x

"x" don't care

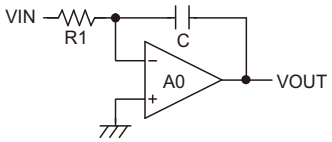
Bit	7	6	5	4	3	2	1	0
OPA0C	A0PS2	A0PS1	A0PS0	CPS2	CPS1	CPS0	CNS1	CNS0
Setup value	1	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
OPA1C	A1G2	A1G1	A1G0	A1PS2	A1PS1	A1PS0	A1NS1	A1NS0
Setup value	0	0	0	1	0	0	0	1

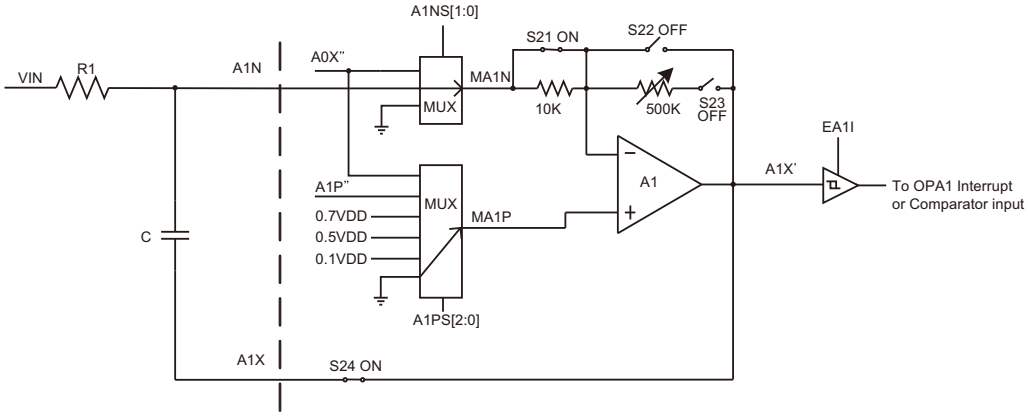
Switch control bits options:
S11: ON
S12: OFF
S13: ON
S21: OFF
S22: OFF
S23: ON
S24: ON
A0PS[2:0]: 100
A1PS[2:0]: 100
A1NS[1:0]: 01
A1G[2:0]: User define OPA1 Gain control

Integrator

- Example



- Implementation connection



- Integrator Switch Setup

Bit	7	6	5	4	3	2	1	0
OPA2C	S24	S23	S22	S21	S13	S12	S11	CXC
Setup value	1	0	0	1	x	x	x	x

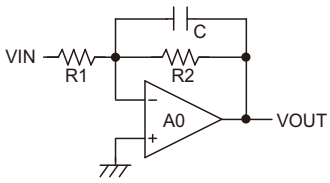
"x" don't care

Bit	7	6	5	4	3	2	1	0
OPA0C	A0PS2	A0PS1	A0PS0	CPS2	CPS1	CPS0	CNS1	CNS0
Setup value	0	0	0	1	0	0	0	0

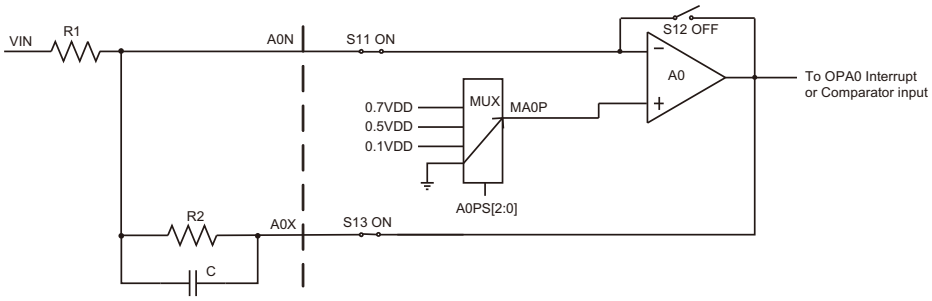
Switch control bits options:
S21: ON
S22: OFF
S23: OFF
S24: ON
A1PS[2:0]: 100
A1NS[1:0]: 00
A1G[2:0]: User define OPA1 Gain control

Low Pass Filter

- Example



- Implementation connection



- Low Pass Filter Switch Setup

Bit	7	6	5	4	3	2	1	0
OPA2C	S24	S23	S22	S21	S13	S12	S11	CXC
Setup value	x	x	x	x	1	0	1	x

"x" don't care

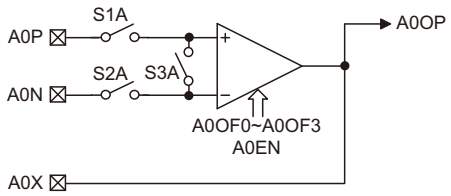
Bit	7	6	5	4	3	2	1	0
OPA0C	A0PS2	A0PS1	A0PS0	CPS2	CPS1	CPS0	CNS1	CNS0
Setup value	1	0	0	0	0	0	0	0

Switch control bits options:
S11: ON
S12: OFF
S13: ON
A0PS[2:0]: 100

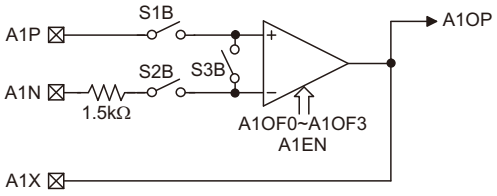
Operational Amplifier Offset Cancellation Function

Each of the internal OPAs allows for a common mode adjustment method of its input offset voltage.

A0RS	A0OFM	S1A	S2A	S3A
0	0	ON	ON	OFF
0	1	OFF	ON	ON
1	0	ON	ON	OFF
1	1	ON	OFF	ON



A1RS	A1OFM	S1B	S2B	S3B
0	0	ON	ON	OFF
0	1	OFF	ON	ON
1	0	ON	ON	OFF
1	1	ON	OFF	ON



The calibration steps are as following:

1. Set A0OFM=1 to setup the offset cancellation mode, here S3A is closed.
2. Set A0RS to select which input pin is to be used as the reference voltage - S1 or S2 is closed
3. Adjust A0OF0~A0OF3 until the output status changes
4. Set A0OFM = 0 to restore the normal OPA mode
5. Repeat the same procedure from steps 1 to 4 for OPA1.

Comparator

These devices contain a fully integrated Comparator whose operation is controlled by the Comparator control registers, known as the CMP0C, CMP1C, COPA0C, COPA2C and COPA3C registers. The CEN bit within CMP0C register is used as the enable or disable bit for the comparator function. The advantages of multiple input resources, multiple reference voltage options, output polarity control, output to Timer counter, multiple output interrupt triggers, comparator output wakeup MCU function, comparator output with de-bounce options, comparator operating current selection and power down control for low power consumption enhance the flexibility of this comparator to suit a wide range of application possibilities.

Comparator Functions

The Comparator can work with OPAs or standalone as shown in the main functional blocks of the OPAs and Comparator in this device. This comparator provides three operating current options, which are 200μA, 5μA and 1μA. The purpose of this design is to provide the suitable comparator power consumption for different operating modes of the device. The higher the operating current, the shorter the comparator response time, therefore, the designer can select the higher operating current for the device working at normal mode and a lower one for the device entering power down mode. By this way, this comparator can operate under very low power consumption and perform as a wakeup resource when the device enters power down mode. In addition, this device provides different comparator output de-bounce time options for different input signal. If the input signal is noise sensitive, then the better choice will be the longer de-bounce time. The designer could select the suitable de-bounce time according to the input signal.

CMP0C Register

Bit	7	6	5	4	3	2	1	0
Name	—	CEN	CPOL	COUT	DBC1	DBC0	CPCS1	CPCS0
R/W	—	R/W	R/W	R	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 unimplemented, read as ″0″

Bit 6 **CEN**: comparator on/off bit
0: off
1: on
Note that the designer should enable the comparator first before enabling the comparator interrupt, in order to prevent an unexpected interrupt.

Bit 5 **CPOL**: comparator output polarity control bit
0: not inverted
1: inverted

Bit 4 **COUT**: comparator output bit.
CPOL=0: If the CP pin input voltage is less than CN pin, then the COUT is ″0″.
If the CP pin input voltage is greater than CN pin, then the COUT is ″1″.
CPOL=1: If the CP pin input voltage is less than CN pin, then the COUT is ″1″.
If the CP pin input voltage is greater than CN pin, then the COUT is ″0″.

Bit 3~2 **DBC1, DBC0**: De-bounce time selection, up to application signal
00: no de-bounce
01: de-bounce time= 1 system clock
10: de-bounce time= 4 system clock
11: de-bounce time= 16 system clock

Bit 1~0 **CPCS1, CPCS0]**: Comparator operating current selection for low power consumption
00: 200μA
01: 5μA
10: 1μA
11: not implemented

CMP1C Register

Bit	7	6	5	4	3	2	1	0
Name	A0VRC	A1VRC	CPVRC	—	TMR0S	—	CINTS1	CINTS0
R/W	R/W	R/W	R/W	—	R/W	—	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit7 **A0VRC**: OPA0 non-inverting input connection control bit
0: connected to internal reference voltage only
1: connected to both internal reference voltage and external I/O (A0P) pin

Bit6 **A1VRC**: OPA1 non-inverting input connection control bit
0: connected to internal reference voltage only
1: connected to both internal reference voltage and external I/O (A1P) pin

Bit5 **CPVRC**: Comparator non-inverting input connection control bit
0: connected to internal reference voltage only
1: connected to both internal reference voltage and external I/O (CP) pin
Note that the above setting of these three bits, which are A0VRC, A1VRC and CPVRC, is valid when the non inverting input pins are selected to be connected to the internal reference voltage by A0PS[2:0],A1PS[2:0] and CPS[2:0] control bits respectively.

Bit 4, 2 unimplemented, read as ″0″

Bit 3 **TMR0S**: signal input path selection for Timer 0 Event counter
0: from TC0 pin
1: from comparator output

Bit 1~0 **CINTS1, CINTS0**: comparator interrupt trigger type selection
00: falling edge
01: rising edge
10: both edge
11: reserved

Interrupts

Interrupts are an important part of any microcontroller system. When an external event or an internal function such as a Timer/Event Counter or Time Base requires microcontroller attention, their corresponding interrupt will enforce a temporary suspension of the main program allowing the microcontroller to direct attention to their respective needs.

The devices contain a single external interrupt and multiple internal interrupts. The external interrupt is controlled by the action of the external interrupt pin, while the internal interrupts are controlled by the various functions such as Timer/Event Counters and Time Base overflow, etc.

Interrupt Register

Overall interrupt control, which means interrupt enabling and request flag setting, is controlled by using the registers, INTC0 and INTC1. By controlling the appropriate enable bits in the registers each individual interrupt can be enabled or disabled. Also when an interrupt occurs, the corresponding request flag will be set by the microcontroller. The global enable control bit if cleared to zero will disable all interrupts.

INTC0 Register – HT48R064G

Bit	7	6	5	4	3	2	1	0
Name	—	—	T0F	INTF	—	T0E	INTE	EMI
R/W	—	—	R/W	R/W	—	R/W	R/W	R/W
POR	—	—	0	0	—	0	0	0

Bit 7~6 unimplemented, read as ″0″

Bit 5 **T0F**: Timer/Event Counter 0 interrupt request flag
0: inactive
1: active

Bit 4 **INTF**: External interrupt request flag
0: inactive
1: active

Bit 3 unimplemented, read as ″0″

Bit 2 **T0E**: Timer/Event Counter 0 interrupt enable
0: disable
1: enable

Bit 1 **INTE**: external interrupt enable
0: disable
1: enable

Bit 0 **EMI**: Master interrupt global enable
0: disable
1: enable

INTC0 Register – HT48R065G/HT48R0662G

Bit	7	6	5	4	3	2	1	0
Name	—	T1F	T0F	INTF	T1E	T0E	INTE	EMI
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

Bit 7 unimplemented, read as "0"
Bit 6 **T1F**: Timer/Event Counter 1 interrupt request flag
0: inactive
1: active
Bit 5 **T0F**: Timer/Event Counter 0 interrupt request flag
0: inactive
1: active
Bit 4 **INTF**: External interrupt request flag
0: inactive
1: active
Bit 3 **T1E**: Timer/Event Counter 1 interrupt enable
0: disable
1: enable
Bit 2 **T0E**: Timer/Event Counter 0 interrupt enable
0: disable
1: enable
Bit 1 **INTE**: external interrupt enable
0: disable
1: enable
Bit 0 **EMI**: Master interrupt global enable
0: disable
1: enable

INTC0 Register – HT48R066G

Bit	7	6	5	4	3	2	1	0
Name	—	T1F	T0F	EIF	ET1I	ET0I	EEI	EMI
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

Bit 7 unimplemented, read as "0"
Bit 6 **T1F**: Timer/Event Counter 1 interrupt request flag
0: inactive
1: active
Bit 5 **T0F**: Timer/Event Counter 0 interrupt request flag
0: inactive
1: active
Bit 4 **EIF**: External interrupt request flag
0: inactive
1: active
Bit 3 **ET1I**: Timer/Event Counter 1 interrupt enable
0: disable
1: enable
Bit 2 **ET0I**: Timer/Event Counter 0 interrupt enable
0: disable
1: enable
Bit 1 **EEI**: External interrupt enable
0: disable
1: enable
Bit 0 **EMI**: Master interrupt global enable
0: disable
1: enable

INTC1 Register – All devices

Bit	7	6	5	4	3	2	1	0
Name	—	MFF	TBF	—	—	MFE	TBE	—
R/W	—	R/W	R/W	—	—	R/W	R/W	—
POR	—	0	0	—	—	0	0	—

Bit 7 unimplemented, read as ″0″

Bit 6 **MFF**: Multi-function interrupt request flag
0: inactive
1: active

Bit 5 **TBF**: Time Base event interrupt request flag
0: inactive
1: active

Bit 4~3 unimplemented, read as ″0″

Bit 2 **MFE**: Multi-function interrupt enable
0: disable
1: enable

Bit 1 **TBE**: Time base event interrupt enable
0: disable
1: enable

Bit 0 unimplemented, read as ″0″

MFIC Register – All devices

Bit	7	6	5	4	3	2	1	0
Name	—	A1F	A0F	CF	—	EA1I	EA0I	ECI
R/W	—	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	—	0	0	0	—	0	0	0

Bit 7 unimplemented, read as ″0″

Bit 6 **A1F**: OPA1 interrupt request flag
0: inactive
1: active

Bit 5 **A0F**: OPA0 interrupt request flag
0: inactive
1: active

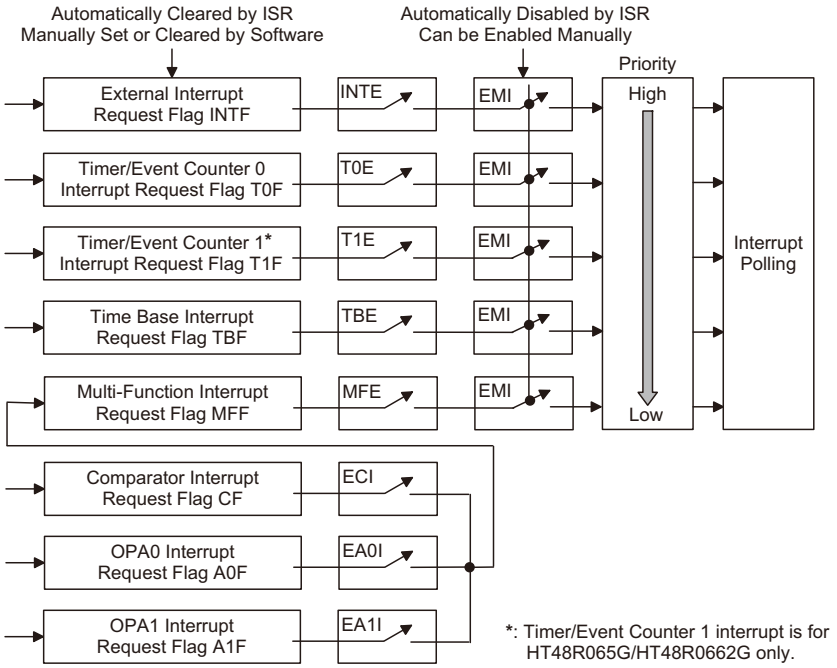
Bit 4 **CF**: Comparator interrupt request flag
0: inactive
1: active

Bit 3 unimplemented, read as ″0″

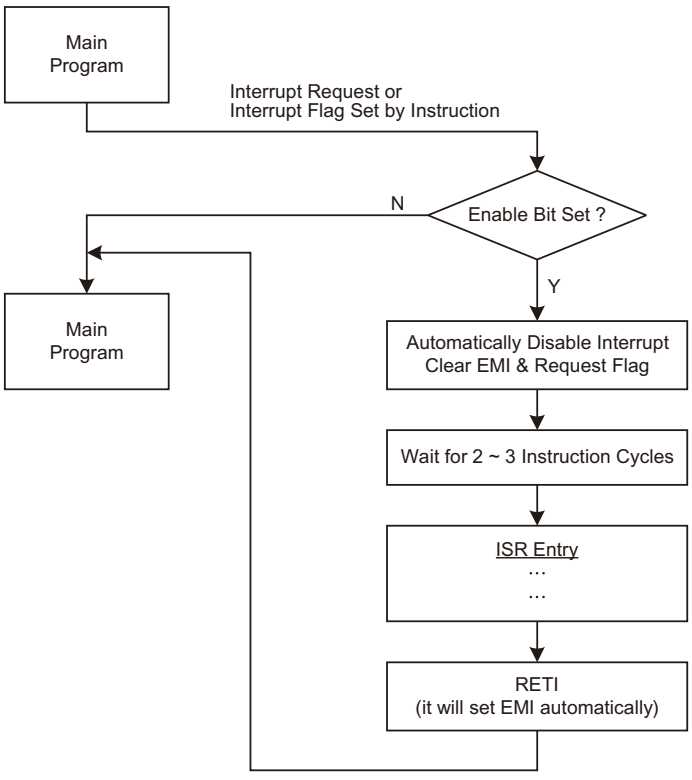
Bit 2 **EA1I**: OPA1 interrupt enable
0: disable
1: enable

Bit 1 **EA0I**: OPA0 interrupt enable
0: disable
1: enable

Bit 0 **ECI**: Comparator interrupt enable
0: disable
1: enable



Interrupt Scheme



Interrupt Flow

Interrupt Operation

A Timer/Event Counter overflow, an active edge on the external interrupt pin, a comparator output transition, an OPA output falling edge or a Time Base event will all generate an interrupt request by setting their corresponding request flag. When this happens, the Program Counter, which stores the address of the next instruction to be executed, will be transferred onto the stack. The Program Counter will then be loaded with a new address which will be the value of the corresponding interrupt vector. The microcontroller will then fetch its next instruction from this interrupt vector. The instruction at this vector will usually be a JMP statement which will jump to another section of program which is known as the interrupt service routine. Here is located the code to control the appropriate interrupt. The interrupt service routine must be terminated with a RETI instruction, which retrieves the original Program Counter address from the stack and allows the microcontroller to continue with normal execution at the point where the interrupt occurred.

The various interrupt enable bits, together with their associated request flags, are shown in the following diagram with their order of priority.

Once an interrupt subroutine is serviced, all the other interrupts will be blocked, as the EMI bit will be cleared automatically. This will prevent any further interrupt nesting from occurring. However, if other interrupt requests occur during this interval, although the interrupt will not be immediately serviced, the request flag will still be recorded. If an interrupt requires immediate servicing while the program is already in another interrupt service routine, the EMI bit should be set after entering the routine, to allow interrupt nesting. If the stack is full, the interrupt request will not be acknowledged, even if the related interrupt is enabled, until the Stack Pointer is decremented. If immediate service is desired, the stack must be prevented from becoming full.

When an interrupt request is generated it takes 2 or 3 instruction cycle before the program jumps to the interrupt vector. If the device is in the Sleep or Idle Mode and is woken up by an interrupt request then it will take 3 cycles before the program jumps to the interrupt vector.

Interrupt Priority

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

HT48R064G		
Interrupt Source	**Priority**	**Vector**
External Interrupt	1	04H
Timer/Event Counter 0 Overflow	2	08H
Time Base Overflow	3	14H
Multi-function interrupt (Comparator, OPA0, OPA1)	4	18H

HT48R065G/HT48R066G/HT48R0662G		
Interrupt Source	**Priority**	**Vector**
External Interrupt	1	04H
Timer/Event Counter 0 Overflow	2	08H
Timer/Event Counter 1 Overflow	3	0CH
Time Base Overflow	4	14H
Multi-function interrupt (Comparator, OPA0, OPA1)	5	18H

In cases where both external and internal interrupts are enabled and where an external and internal interrupt occurs simultaneously, the external interrupt will always have priority and will therefore be serviced first. Suitable masking of the individual interrupts using the interrupt registers can prevent simultaneous occurrences.

External Interrupt

For an external interrupt to occur, the global interrupt enable bit, EMI, and external interrupt enable bit, INTE, must first be set. An actual external interrupt will take place when the external interrupt request flag, INTF, is set, a situation that will occur when an edge transition appears on the external INT line. The type of transition that will trigger an external interrupt, whether high to low, low to high or both is determined by the INTEG0 and INTEG1 bits, which are bits 6 and 7 respectively, in the CTRL1 control register. These two bits can also disable the external interrupt function.

INTEG1	INTEG0	Edge Trigger Type
0	0	External interrupt disable
0	1	Rising edge Trigger
1	0	Falling edge Trigger
1	1	Both edge Trigger

The external interrupt pin is pin-shared with the I/O pin PA3 and can only be configured as an external interrupt pin if the corresponding external interrupt enable bit in the INTC register has been set and the edge trigger type has been selected using the CTRL1 register. The pin must also be setup as an input by setting the corresponding PAC.3 bit in the port control register. When the interrupt is enabled, the stack is not full and an active transition appears on the external interrupt pin, a subroutine call to the external interrupt vector at location 04H, will take place. When the interrupt is serviced, the external interrupt request flag, INTF, will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts. Note that any pull-high resistor connections on this pin will remain valid even if the pin is used as an external interrupt input.

Timer/Event Counter Interrupt

For a Timer/Event Counter interrupt to occur, the global interrupt enable bit, EMI, and the corresponding timer interrupt enable bit, TnE, must first be set. An actual Timer/Event Counter interrupt will take place when the Timer/Event Counter request flag, TnF, is set, a situation that will occur when the relevant Timer/Event Counter overflows. When the interrupt is enabled, the stack is not full and a Timer/Event Counter n overflow occurs, a subroutine call to the relevant timer interrupt vector, will take place. When the interrupt is serviced, the timer interrupt request flag, TnF, will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts.

Time Base Interrupt

For a time base interrupt to occur the global interrupt enable bit EMI and the corresponding interrupt enable bit TBE, must first be set. An actual Time Base interrupt will take place when the time base request flag TBF is set, a situation that will occur when the Time Base overflows. When the interrupt is enabled, the stack is not full and a time base overflow occurs a subroutine call to time base vector will take place. When the interrupt is serviced, the time base interrupt flag. TBF will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts.

Multi-function Interrupt

For a Multi-function interrupt to occur, the global interrupt enable bit, EMI, and the corresponding multi-function interrupt enable bit, MFE, must first be set. An actual Multi-function interrupt will take place when the Multi-function interrupt request flag, MFF, is set, a situation that will occur when OPA0 or OPA1 output has a falling edge, or a Comparator output transition occurs. When the interrupt is enabled, the stack is not full and a Multi-function interrupt request occurs, a subroutine call to the Multi-function interrupt vector at location 18H, will take place. When the interrupt is serviced, the Multi-function interrupt request flag, MFF, will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts. After the Multi-function took place, the programmer can check what the interrupt source was by interrogating the request flags, A0F, A1F or CF within the MFIC register.

Programming Considerations

By disabling the interrupt enable bits, a requested interrupt can be prevented from being serviced, however, once an interrupt request flag is set, it will remain in this condition in the interrupt register until the corresponding interrupt is serviced or until the request flag is cleared by a software instruction.

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

All of these interrupts have the capability of waking up the processor when in the Idle/Sleep Mode.

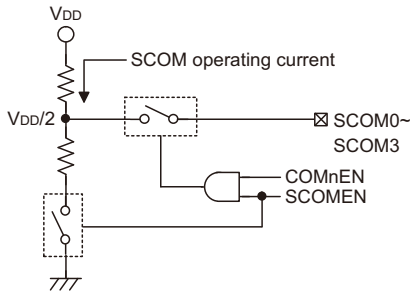
Only the Program Counter is pushed onto the stack. If the contents of the register or status register are altered by the interrupt service program, which may corrupt the desired control sequence, then the contents should be saved in advance.

SCOM Function for LCD

The HT48R065G, HT48R066G and HT48R0662G devices have the capability of driving external LCD panels. The common pins for LCD driving, SCOM0~SCOM3, are pin shared with certain pin on the PB0~PB3 port. The LCD signals (COM and SEG) are generated using the application program.

LCD Operation

An external LCD panel can be driven using this device by configuring the PB0~PB3 pins as common pins and using other output ports lines as segment pins. The LCD driver function is controlled using the SCOMC register which in addition to controlling the overall on/off function also controls the bias voltage setup function. This enables the LCD COM driver to generate the necessary $V_{DD}/2$ voltage levels for LCD 1/2 bias operation.



LCD COM Bias

The SCOMEN bit in the SCOMC register is the overall master control for the LCD Driver, however this bit is used in conjunction with the COMnEN bits to select which Port B and Port C pins are used for LCD driving. Note that the Port Control register does not need to first setup the pins as outputs to enable the LCD driver operation.

SCOMEN	COMnEN	Pin Function	O/P Level
0	X	I/O	0 or 1
1	0	I/O	0 or 1
1	1	SCOMN	$V_{DD}/2$

Output Control

LCD Bias Control

The LCD COM driver enables a range of selections to be provided to suit the requirement of the LCD panel which is being used. The bias resistor choice is implemented using the ISEL1 and ISEL0 bits in the SCOMC register.

SCOMC Register

Bit	7	6	5	4	3	2	1	0
Name	—	ISEL1	ISEL0	SCOMEN	COM3EN	COM2EN	COM1EN	COM0EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 Reserved Bit
1: Unpredictable operation - bit must NOT be set high
0: Correct level - bit must be reset to zero for correct operation

Bit 6,5 **ISEL1, ISEL0**: SCOM operating current selection (V_{DD}=5V)
00: 25μA
01: 50μA
10: 100μA
11: 200μA

Bit 4 **SCOMEN**: SCOM module on/off control
0: disable
1: enable
SCOMn can be enable by COMnEN if SCOMEN=1

Bit 3 **COM3EN**: PC6 or SCOM3 selection
0: GPIO
1: SCOM3

Bit 2 **COM2EN**: PC7 or SCOM2 selection
0: GPIO
1: SCOM2

Bit 1 **COM1EN**: PB7 or SCOM1 selection
0: GPIO
1: SCOM1

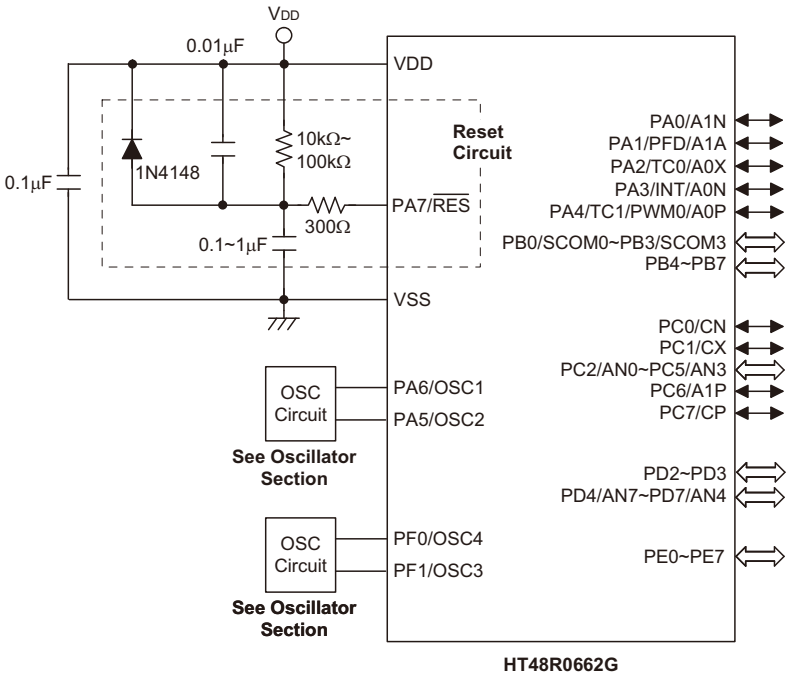
Bit 0 **COM0EN**: PB6 or SCOM0 selection
0: GPIO
1: SCOM0

Configuration Options

Configuration options refer to certain options within the MCU that are programmed into the OTP Program Memory device during the programming process. During the development process, these options are selected using the HT-IDE software development tools. As these options are programmed into the device using the hardware programming tools, once they are selected they cannot be changed later by the application software. All options must be defined for proper system function, the details of which are shown in the table.

No.	Options
1	Watchdog Timer: enable or disable
2	Watchdog Timer clock source: LXT, LIRC or $f_{SYS}/4$ Note: LXT oscillator must be selected by OSC configuration option if WDT clock source is from LXT.
3	CLRWDT instructions: 1 or 2 instructions
4	For HT48R064G/HT48R065G/HT48R066G System oscillator configuration: HXT, HIRC, ERC, HIRC+LXT For HT48R0662G System oscillator configuration: HXT, HIRC, ERC, HXT+LXT, HIRC+LXT, ERC+LXT
5	LVR function: enable or disable
6	LVR voltage: 2.1V, 3.15V or 4.2V
7	$\overline{RES}$ or PA7 pin function
8	HIRC oscillator frequency: 4MHz, 8MHz or 12MHz

Application Circuits



Instruction Set

Introduction

Central to the successful operation of any microcontroller is its instruction set, which is a set of program instruction codes that directs the microcontroller to perform certain operations. In the case of Holtek microcontrollers, a comprehensive and flexible set of over 60 instructions is provided to enable programmers to implement their application with the minimum of programming overheads.

For easier understanding of the various instruction codes, they have been subdivided into several functional groupings.

Instruction Timing

Most instructions are implemented within one instruction cycle. The exceptions to this are branch, call, or table read instructions where two instruction cycles are required. One instruction cycle is equal to 4 system clock cycles, therefore in the case of an 8MHz system oscillator, most instructions would be implemented within 0.5μs and branch or call instructions would be implemented within 1μs. Although instructions which require one more cycle to implement are generally limited to the JMP, CALL, RET, RETI and table read instructions, it is important to realize that any other instructions which involve manipulation of the Program Counter Low register or PCL will also take one more cycle to implement. As instructions which change the contents of the PCL will imply a direct jump to that new address, one more cycle will be required. Examples of such instructions would be ″CLR PCL″ or ″MOV PCL, A″. For the case of skip instructions, it must be noted that if the result of the comparison involves a skip operation then this will also take one more cycle, if no skip is involved then only one cycle is required.

Moving and Transferring Data

The transfer of data within the microcontroller program is one of the most frequently used operations. Making use of three kinds of MOV instructions, data can be transferred from registers to the Accumulator and vice-versa as well as being able to move specific immediate data directly into the Accumulator. One of the most important data transfer applications is to receive data from the input ports and transfer data to the output ports.

Arithmetic Operations

The ability to perform certain arithmetic operations and data manipulation is a necessary feature of most microcontroller applications. Within the Holtek microcontroller instruction set are a range of add and subtract instruction mnemonics to enable the necessary arithmetic to be carried out. Care must be taken to ensure correct handling of carry and borrow data when results exceed 255 for addition and less than 0 for subtraction. The increment and decrement instructions INC, INCA, DEC and DECA provide a simple means of increasing or decreasing by a value of one of the values in the destination specified.

Logical and Rotate Operations

The standard logical operations such as AND, OR, XOR and CPL all have their own instruction within the Holtek microcontroller instruction set. As with the case of most instructions involving data manipulation, data must pass through the Accumulator which may involve additional programming steps. In all logical data operations, the zero flag may be set if the result of the operation is zero. Another form of logical data manipulation comes from the rotate instructions such as RR, RL, RRC and RLC which provide a simple means of rotating one bit right or left. Different rotate instructions exist depending on program requirements. Rotate instructions are useful for serial port programming applications where data can be rotated from an internal register into the Carry bit from where it can be examined and the necessary serial bit set high or low. Another application where rotate data operations are used is to implement multiplication and division calculations.

Branches and Control Transfer

Program branching takes the form of either jumps to specified locations using the JMP instruction or to a subroutine using the CALL instruction. They differ in the sense that in the case of a subroutine call, the program must return to the instruction immediately when the subroutine has been carried out. This is done by placing a return instruction RET in the subroutine which will cause the program to jump back to the address right after the CALL instruction. In the case of a JMP instruction, the program simply jumps to the desired location. There is no requirement to jump back to the original jumping off point as in the case of the CALL instruction. One special and extremely useful set of branch instructions are the conditional branches. Here a decision is first made regarding the condition of a certain data memory or individual bits. Depending upon the conditions, the program will continue with the next instruction or skip over it and jump to the following instruction. These instructions are the key to decision making and branching within the program perhaps determined by the condition of certain input switches or by the condition of internal data bits.

Bit Operations

The ability to provide single bit operations on Data Memory is an extremely flexible feature of all Holtek microcontrollers. This feature is especially useful for output port bit programming where individual bits or port pins can be directly set high or low using either the "SET [m].i" or "CLR [m].i" instructions respectively. The feature removes the need for programmers to first read the 8-bit output port, manipulate the input data to ensure that other bits are not changed and then output the port with the correct new data. This read-modify-write process is taken care of automatically when these bit operation instructions are used.

Table Read Operations

Data storage is normally implemented by using registers. However, when working with large amounts of fixed data, the volume involved often makes it inconvenient to store the fixed data in the Data Memory. To overcome this problem, Holtek microcontrollers allow an area of Program Memory to be setup as a table where data can be directly stored. A set of easy to use instructions provides the means by which this fixed data can be referenced and retrieved from the Program Memory.

Other Operations

In addition to the above functional instructions, a range of other instructions also exist such as the "HALT" instruction for Power-down operations and instructions to control the operation of the Watchdog Timer for reliable program operations under extreme electric or electromagnetic environments. For their relevant operations, refer to the functional related sections.

Instruction Set Summary

The following table depicts a summary of the instruction set categorised according to function and can be consulted as a basic instruction reference using the following listed conventions.

Table conventions:
x: Bits immediate data
m: Data Memory address
A: Accumulator
i: 0~7 number of bits
addr: Program memory address

Mnemonic	Description	Cycles	Flag Affected
Arithmetic			
ADD A,[m]	Add Data Memory to ACC	1	Z, C, AC, OV
ADDM A,[m]	Add ACC to Data Memory	1Note	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	1Note	Z, C, AC, OV
SUB A,x	Subtract immediate data from the 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	1Note	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, result in Data Memory	1Note	Z, C, AC, OV
DAA [m]	Decimal adjust ACC for Addition with result in Data Memory	1Note	C
Logic Operation			
AND A,[m]	Logical AND Data Memory to ACC	1	Z
OR A,[m]	Logical OR Data Memory to ACC	1	Z
XOR A,[m]	Logical XOR Data Memory to ACC	1	Z
ANDM A,[m]	Logical AND ACC to Data Memory	1Note	Z
ORM A,[m]	Logical OR ACC to Data Memory	1Note	Z
XORM A,[m]	Logical XOR ACC to Data Memory	1Note	Z
AND A,x	Logical AND immediate Data to ACC	1	Z
OR A,x	Logical OR immediate Data to ACC	1	Z
XOR A,x	Logical XOR immediate Data to ACC	1	Z
CPL [m]	Complement Data Memory	1Note	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	1Note	Z
DECA [m]	Decrement Data Memory with result in ACC	1	Z
DEC [m]	Decrement Data Memory	1Note	Z
Rotate			
RRA [m]	Rotate Data Memory right with result in ACC	1	None
RR [m]	Rotate Data Memory right	1Note	None
RRCA [m]	Rotate Data Memory right through Carry with result in ACC	1	C
RRC [m]	Rotate Data Memory right through Carry	1Note	C
RLA [m]	Rotate Data Memory left with result in ACC	1	None
RL [m]	Rotate Data Memory left	1Note	None
RLCA [m]	Rotate Data Memory left through Carry with result in ACC	1	C
RLC [m]	Rotate Data Memory left through Carry	1Note	C
Data Move			
MOV A,[m]	Move Data Memory to ACC	1	None
MOV [m],A	Move ACC to Data Memory	1Note	None
MOV A,x	Move immediate data to ACC	1	None

Mnemonic	Description	Cycles	Flag Affected
Bit Operation			
CLR [m].i	Clear bit of Data Memory	1Note	None
SET [m].i	Set bit of Data Memory	1Note	None
Branch			
JMP addr	Jump unconditionally	2	None
SZ [m]	Skip if Data Memory is zero	1Note	None
SZA [m]	Skip if Data Memory is zero with data movement to ACC	1note	None
SZ [m].i	Skip if bit i of Data Memory is zero	1Note	None
SNZ [m].i	Skip if bit i of Data Memory is not zero	1Note	None
SIZ [m]	Skip if increment Data Memory is zero	1Note	None
SDZ [m]	Skip if decrement Data Memory is zero	1Note	None
SIZA [m]	Skip if increment Data Memory is zero with result in ACC	1Note	None
SDZA [m]	Skip if decrement Data Memory is zero with result in ACC	1Note	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 table (current page) to TBLH and Data Memory	2Note	None
TABRDL [m]	Read table (last page) to TBLH and Data Memory	2Note	None
Miscellaneous			
NOP	No operation	1	None
CLR [m]	Clear Data Memory	1Note	None
SET [m]	Set Data Memory	1Note	None
CLR WDT	Clear Watchdog Timer	1	TO, PDF
CLR WDT1	Pre-clear Watchdog Timer	1	TO, PDF
CLR WDT2	Pre-clear Watchdog Timer	1	TO, PDF
SWAP [m]	Swap nibbles of Data Memory	1Note	None
SWAPA [m]	Swap nibbles of Data Memory with result in ACC	1	None
HALT	Enter power down mode	1	TO, PDF

Note: 1. For skip instructions, if the result of the comparison involves a skip then two cycles are required, if no skip takes place only one cycle is required.
2. Any instruction which changes the contents of the PCL will also require 2 cycles for execution.
3. For the ″CLR WDT1″ and ″CLR WDT2″ instructions the TO and PDF flags may be affected by the execution status. The TO and PDF flags are cleared after both ″CLR WDT1″ and ″CLR WDT2″ instructions are consecutively executed. Otherwise the TO and PDF flags remain unchanged.

Instruction Definition

ADC A,[m] Add Data Memory to ACC with Carry

Description: The contents of the specified Data Memory, Accumulator and the carry flag are added. The result is stored in the Accumulator.

Operation: ACC ← ACC + [m] + C

Affected flag(s): OV, Z, AC, C

ADCM A,[m] Add ACC to Data Memory with Carry

Description: The contents of the specified Data Memory, Accumulator and the carry flag are added. The result is stored in the specified Data Memory.

Operation: [m] ← ACC + [m] + C

Affected flag(s): OV, Z, AC, C

ADD A,[m] Add Data Memory to ACC

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): OV, Z, AC, C

ADD A,x Add immediate data to ACC

Description: The contents of the Accumulator and the specified immediate data are added. The result is stored in the Accumulator.

Operation: ACC ← ACC + x

Affected flag(s): OV, Z, AC, C

ADDM A,[m] Add ACC to Data Memory

Description: The contents of the specified Data Memory and the Accumulator are added. The result is stored in the specified Data Memory.

Operation: [m] ← ACC + [m]

Affected flag(s): OV, Z, AC, C

AND A,[m] Logical AND Data Memory to ACC

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): Z

AND A,x Logical AND immediate data to ACC

Description: Data in the Accumulator and the specified immediate data perform a bitwise logical AND operation. The result is stored in the Accumulator.

Operation: ACC ← ACC ″AND″ x

Affected flag(s): Z

ANDM A,[m] Logical AND ACC to Data Memory

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): Z

CALL addr	Subroutine call
Description	Unconditionally calls a subroutine at the specified address. The Program Counter then increments by 1 to obtain the address of the next instruction which is then pushed onto the stack. The specified address is then loaded and the program continues execution from this new address. As this instruction requires an additional operation, it is a two cycle instruction.
Operation	Stack ← Program Counter + 1 Program Counter ← addr
Affected flag(s)	None
CLR [m]	Clear Data Memory
Description	Each bit of the specified Data Memory is cleared to 0.
Operation	[m] ← 00H
Affected flag(s)	None
CLR [m].i	Clear bit of Data Memory
Description	Bit i of the specified Data Memory is cleared to 0.
Operation	[m].i ← 0
Affected flag(s)	None
CLR WDT	Clear Watchdog Timer
Description	The TO, PDF flags and the WDT are all cleared.
Operation	WDT cleared TO ← 0 PDF ← 0
Affected flag(s)	TO, PDF
CLR WDT1	Pre-clear Watchdog Timer
Description	The TO, PDF flags and the WDT are all cleared. Note that this instruction works in conjunction with CLR WDT2 and must be executed alternately with CLR WDT2 to have effect. Repetitively executing this instruction without alternately executing CLR WDT2 will have no effect.
Operation	WDT cleared TO ← 0 PDF ← 0
Affected flag(s)	TO, PDF
CLR WDT2	Pre-clear Watchdog Timer
Description	The TO, PDF flags and the WDT are all cleared. Note that this instruction works in conjunction with CLR WDT1 and must be executed alternately with CLR WDT1 to have effect. Repetitively executing this instruction without alternately executing CLR WDT1 will have no effect.
Operation	WDT cleared TO ← 0 PDF ← 0
Affected flag(s)	TO, PDF

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): Z

CPLA [m] Complement Data Memory with result in ACC

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): Z

DAA [m] Decimal-Adjust ACC for addition with result in Data Memory

Description: Convert the contents of the Accumulator value to a BCD (Binary Coded Decimal) value resulting from the previous addition of two BCD variables. If the low nibble is greater than 9 or if AC flag is set, then a value of 6 will be added to the low nibble. Otherwise the low nibble remains unchanged. If the high nibble is greater than 9 or if the C flag is set, then a value of 6 will be added to the high nibble. Essentially, the decimal conversion is performed by adding 00H, 06H, 60H or 66H depending on the Accumulator and flag conditions. Only the C flag may be affected by this instruction which indicates that if the original BCD sum is greater than 100, it allows multiple precision decimal addition.

Operation:
[m] ← ACC + 00H or
[m] ← ACC + 06H or
[m] ← ACC + 60H or
[m] ← ACC + 66H

Affected flag(s): C

DEC [m] Decrement Data Memory

Description: Data in the specified Data Memory is decremented by 1.

Operation: [m] ← [m] − 1

Affected flag(s): Z

DECA [m] Decrement Data Memory with result in ACC

Description: Data in the specified Data Memory is decremented by 1. The result is stored in the Accumulator. The contents of the Data Memory remain unchanged.

Operation: ACC ← [m] − 1

Affected flag(s): Z

HALT Enter power down mode

Description: This instruction stops the program execution and turns off the system clock. The contents of the Data Memory and registers are retained. The WDT and prescaler are cleared. The power down flag PDF is set and the WDT time-out flag TO is cleared.

Operation:
TO ← 0
PDF ← 1

Affected flag(s): TO, PDF

INC [m]	Increment Data Memory
Description	Data in the specified Data Memory is incremented by 1.
Operation	[m] ← [m] + 1
Affected flag(s)	Z
INCA [m]	Increment Data Memory with result in ACC
Description	Data in the specified Data Memory is incremented by 1. The result is stored in the Accumulator. The contents of the Data Memory remain unchanged.
Operation	ACC ← [m] + 1
Affected flag(s)	Z
JMP addr	Jump unconditionally
Description	The contents of the Program Counter are replaced with the specified address. Program execution then continues from this new address. As this requires the insertion of a dummy instruction while the new address is loaded, it is a two cycle instruction.
Operation	Program Counter ← addr
Affected flag(s)	None
MOV A,[m]	Move Data Memory to ACC
Description	The contents of the specified Data Memory are copied to the Accumulator.
Operation	ACC ← [m]
Affected flag(s)	None
MOV A,x	Move immediate data to ACC
Description	The immediate data specified is loaded into the Accumulator.
Operation	ACC ← x
Affected flag(s)	None
MOV [m],A	Move ACC to Data Memory
Description	The contents of the Accumulator are copied to the specified Data Memory.
Operation	[m] ← ACC
Affected flag(s)	None
NOP	No operation
Description	No operation is performed. Execution continues with the next instruction.
Operation	No operation
Affected flag(s)	None
OR A,[m]	Logical OR Data Memory to ACC
Description	Data in the Accumulator and the specified Data Memory perform a bitwise logical OR operation. The result is stored in the Accumulator.
Operation	ACC ← ACC ″OR″ [m]
Affected flag(s)	Z

OR A,x Logical OR immediate data to ACC

Description: Data in the Accumulator and the specified immediate data perform a bitwise logical OR operation. The result is stored in the Accumulator.

Operation: ACC ← ACC "OR" x

Affected flag(s): Z

ORM A,[m] Logical OR ACC to Data Memory

Description: Data in the specified Data Memory 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): Z

RET Return from subroutine

Description: The Program Counter is restored from the stack. Program execution continues at the restored address.

Operation: Program Counter ← Stack

Affected flag(s): None

RET A,x Return from subroutine and load immediate data to ACC

Description: The Program Counter is restored from the stack and the Accumulator loaded with the specified immediate data. Program execution continues at the restored address.

Operation: Program Counter ← Stack
ACC ← x

Affected flag(s): None

RETI Return from interrupt

Description: The Program Counter is restored from the stack and the interrupts are re-enabled by setting the EMI bit. EMI is the master interrupt global enable bit. If an interrupt was pending when the RETI instruction is executed, the pending Interrupt routine will be processed before returning to the main program.

Operation: Program Counter ← Stack
EMI ← 1

Affected flag(s): None

RL [m] Rotate Data Memory left

Description: The contents of the specified Data Memory are rotated left by 1 bit with bit 7 rotated into bit 0.

Operation: [m].(i+1) ← [m].i; (i = 0~6)
[m].0 ← [m].7

Affected flag(s): None

RLA [m] Rotate Data Memory left with result in ACC

Description: The contents of the specified Data Memory are rotated left by 1 bit with bit 7 rotated into bit 0. The rotated result is stored in the Accumulator and the contents of the Data Memory remain unchanged.

Operation: ACC.(i+1) ← [m].i; (i = 0~6)
ACC.0 ← [m].7

Affected flag(s): None

RLC [m] Rotate Data Memory left through Carry

Description: The contents of the specified Data Memory and the carry flag are rotated left by 1 bit. Bit 7 replaces the Carry bit and the original carry flag is rotated into bit 0.

Operation:
[m].(i+1) ← [m].i; (i = 0~6)
[m].0 ← C
C ← [m].7

Affected flag(s): C

RLCA [m] Rotate Data Memory left through Carry with result in ACC

Description: Data in the specified Data Memory and the carry flag are rotated left by 1 bit. Bit 7 replaces the Carry bit and the original carry flag is rotated into the bit 0. The rotated result is stored in the Accumulator and the contents of the Data Memory remain unchanged.

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

Affected flag(s): C

RR [m] Rotate Data Memory right

Description: The contents of the specified Data Memory are rotated right by 1 bit with bit 0 rotated into bit 7.

Operation:
[m].i ← [m].(i+1); (i = 0~6)
[m].7 ← [m].0

Affected flag(s): None

RRA [m] Rotate Data Memory right with result in ACC

Description: Data in the specified Data Memory and the carry flag are rotated right by 1 bit with bit 0 rotated into bit 7. The rotated result is stored in the Accumulator and the contents of the Data Memory remain unchanged.

Operation:
ACC.i ← [m].(i+1); (i = 0~6)
ACC.7 ← [m].0

Affected flag(s): None

RRC [m] Rotate Data Memory right through Carry

Description: The contents of the specified Data Memory and the carry flag are rotated right by 1 bit. Bit 0 replaces the Carry bit and the original carry flag is rotated into bit 7.

Operation:
[m].i ← [m].(i+1); (i = 0~6)
[m].7 ← C
C ← [m].0

Affected flag(s): C

RRCA [m] Rotate Data Memory right through Carry with result in ACC

Description: Data in the specified Data Memory and the carry flag are rotated right by 1 bit. Bit 0 replaces the Carry bit and the original carry flag is rotated into bit 7. The rotated result is stored in the Accumulator and the contents of the Data Memory remain unchanged.

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

Affected flag(s): C

SBC A,[m] Subtract Data Memory from ACC with Carry

Description The contents of the specified Data Memory and the complement of the carry flag are subtracted from the Accumulator. The result is stored in the Accumulator. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.

Operation $ACC \leftarrow ACC - [m] - \overline{C}$

Affected flag(s) OV, Z, AC, C

SBCM A,[m] Subtract Data Memory from ACC with Carry and result in Data Memory

Description The contents of the specified Data Memory and the complement of the carry flag are subtracted from the Accumulator. The result is stored in the Data Memory. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.

Operation $[m] \leftarrow ACC - [m] - \overline{C}$

Affected flag(s) OV, Z, AC, C

SDZ [m] Skip if decrement Data Memory is 0

Description The contents of the specified Data Memory are first decremented by 1. If the result is 0 the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.

Operation $[m] \leftarrow [m] - 1$
Skip if $[m] = 0$

Affected flag(s) None

SDZA [m] Skip if decrement Data Memory is zero with result in ACC

Description The contents of the specified Data Memory are first decremented by 1. If the result is 0, the following instruction is skipped. The result is stored in the Accumulator but the specified Data Memory contents remain unchanged. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0, the program proceeds with the following instruction.

Operation $ACC \leftarrow [m] - 1$
Skip if $ACC = 0$

Affected flag(s) None

SET [m] Set Data Memory

Description Each bit of the specified Data Memory is set to 1.

Operation $[m] \leftarrow FFH$

Affected flag(s) None

SET [m].i Set bit of Data Memory

Description Bit i of the specified Data Memory is set to 1.

Operation $[m].i \leftarrow 1$

Affected flag(s) None

SIZ [m] Skip if increment Data Memory is 0

Description The contents of the specified Data Memory are first incremented by 1. If the result is 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.

Operation [m] ← [m] + 1
Skip if [m] = 0

Affected flag(s) None

SIZA [m] Skip if increment Data Memory is zero with result in ACC

Description The contents of the specified Data Memory are first incremented by 1. If the result is 0, the following instruction is skipped. The result is stored in the Accumulator but the specified Data Memory contents remain unchanged. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.

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

Affected flag(s) None

SNZ [m].i Skip if bit i of Data Memory is not 0

Description If bit i of the specified Data Memory is not 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is 0 the program proceeds with the following instruction.

Operation Skip if [m].i ≠ 0

Affected flag(s) None

SUB A,[m] Subtract Data Memory from ACC

Description The specified Data Memory is subtracted from the contents of the Accumulator. The result is stored in the Accumulator. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.

Operation ACC ← ACC − [m]

Affected flag(s) OV, Z, AC, C

SUBM A,[m] Subtract Data Memory from ACC with result in Data Memory

Description The specified Data Memory is subtracted from the contents of the Accumulator. The result is stored in the Data Memory. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.

Operation [m] ← ACC − [m]

Affected flag(s) OV, Z, AC, C

SUB A,x Subtract immediate data from ACC

Description The immediate data specified by the code is subtracted from the contents of the Accumulator. The result is stored in the Accumulator. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.

Operation ACC ← ACC − x

Affected flag(s) OV, Z, AC, C

SWAP [m] Swap nibbles of Data Memory

Description The low-order and high-order nibbles of the specified Data Memory are interchanged.

Operation [m].3~[m].0 ↔ [m].7 ~ [m].4

Affected flag(s) None

SWAPA [m] Swap nibbles of Data Memory with result in ACC

Description The low-order and high-order nibbles of the specified Data Memory are interchanged. The result is stored in 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) None

SZ [m] Skip if Data Memory is 0

Description If the contents of the specified Data Memory is 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.

Operation Skip if [m] = 0

Affected flag(s) None

SZA [m] Skip if Data Memory is 0 with data movement to ACC

Description The contents of the specified Data Memory are copied to the Accumulator. If the value is zero, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.

Operation ACC ← [m]
Skip if [m] = 0

Affected flag(s) None

SZ [m].i Skip if bit i of Data Memory is 0

Description If bit i of the specified Data Memory is 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0, the program proceeds with the following instruction.

Operation Skip if [m].i = 0

Affected flag(s) None

TABRDC [m] Read table (current page) to TBLH and Data Memory

Description The low byte of the program code (current page) addressed by the table pointer (TBLP) is moved to the specified Data Memory and the high byte moved to TBLH.

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

Affected flag(s) None

TABRDL [m]	Read table (last page) to TBLH and Data Memory
Description	The low byte of the program code (last page) addressed by the table pointer (TBLP) is moved to the specified Data Memory and the high byte moved to TBLH.
Operation	[m] ← program code (low byte) TBLH ← program code (high byte)
Affected flag(s)	None

XOR A,[m]	Logical XOR Data Memory to ACC
Description	Data in the Accumulator and the specified Data Memory perform a bitwise logical XOR operation. The result is stored in the Accumulator.
Operation	ACC ← ACC ″XOR″ [m]
Affected flag(s)	Z

XORM A,[m]	Logical XOR ACC to Data Memory
Description	Data in the specified Data Memory and the Accumulator perform a bitwise logical XOR operation. The result is stored in the Data Memory.
Operation	[m] ← ACC ″XOR″ [m]
Affected flag(s)	Z

XOR A,x	Logical XOR immediate data to ACC
Description	Data in the Accumulator and the specified immediate data perform a bitwise logical XOR operation. The result is stored in the Accumulator.
Operation	ACC ← ACC ″XOR″ x
Affected flag(s)	Z

Package Information

Note that the package information provided here is for consultation purposes only. As this information may be updated at regular intervals users are reminded to consult the Holtek website (http://www.holtek.com.tw/english/literature/package.pdf) for the latest version of the package information.

16-pin DIP (300mil) Outline Dimensions

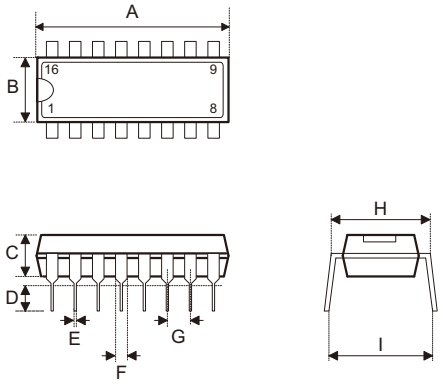


Fig1. Full Lead Packages

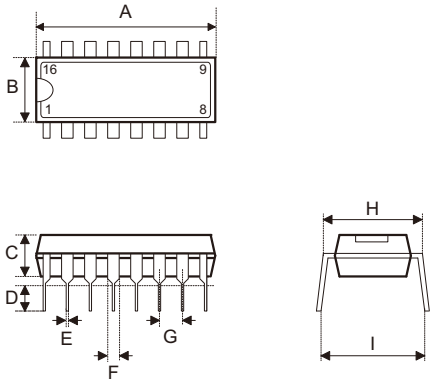


Fig2. 1/2 Lead Packages

MS-001d (see fig1)

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.780	—	0.880
B	0.240	—	0.280
C	0.115	—	0.195
D	0.115	—	0.150
E	0.014	—	0.022
F	0.045	—	0.070
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	19.81	—	22.35
B	6.10	—	7.11
C	2.92	—	4.95
D	2.92	—	3.81
E	0.36	—	0.56
F	1.14	—	1.78
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

MS-001d (see fig2)

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.735	—	0.775
B	0.240	—	0.280
C	0.115	—	0.195
D	0.115	—	0.150
E	0.014	—	0.022
F	0.045	—	0.070
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

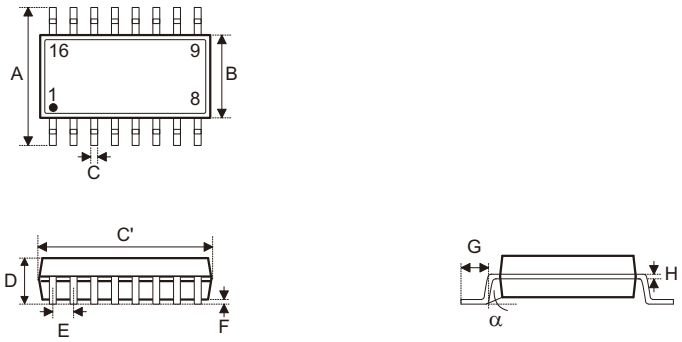
Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	18.67	—	19.69
B	6.10	—	7.11
C	2.92	—	4.95
D	2.92	—	3.81
E	0.36	—	0.56
F	1.14	—	1.78
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

MO-095a (see fig2)

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.745	—	0.785
B	0.275	—	0.295
C	0.120	—	0.150
D	0.110	—	0.150
E	0.014	—	0.022
F	0.045	—	0.060
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	18.92	—	19.94
B	6.99	—	7.49
C	3.05	—	3.81
D	2.79	—	3.81
E	0.36	—	0.56
F	1.14	—	1.52
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

16-pin NSOP (150mil) Outline Dimensions



MS-012

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.228	—	0.244
B	0.150	—	0.157
C	0.012	—	0.020
C′	0.386	—	0.402
D	—	—	0.069
E	—	0.050	—
F	0.004	—	0.010
G	0.016	—	0.050
H	0.007	—	0.010
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	5.79	—	6.20
B	3.81	—	3.99
C	0.30	—	0.51
C′	9.80	—	10.21
D	—	—	1.75
E	—	1.27	—
F	0.10	—	0.25
G	0.41	—	1.27
H	0.18	—	0.25
α	0°	—	8°

20-pin DIP (300mil) Outline Dimensions

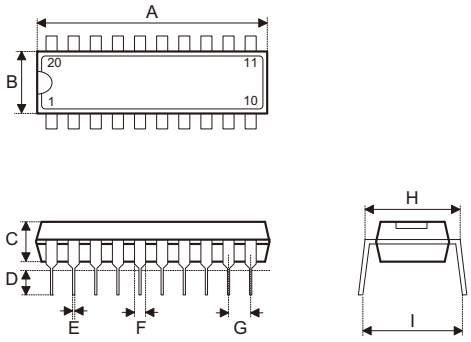


Fig1. Full Lead Packages

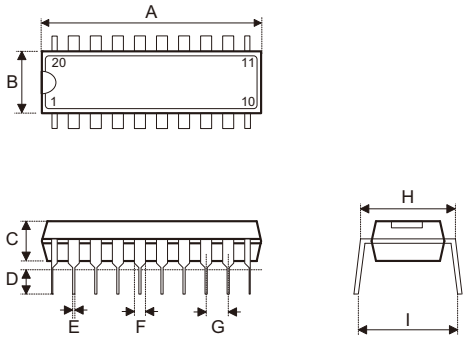


Fig2. 1/2 Lead Packages

MS-001d (see fig1)

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.980	—	1.060
B	0.240	—	0.280
C	0.115	—	0.195
D	0.115	—	0.150
E	0.014	—	0.022
F	0.045	—	0.070
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

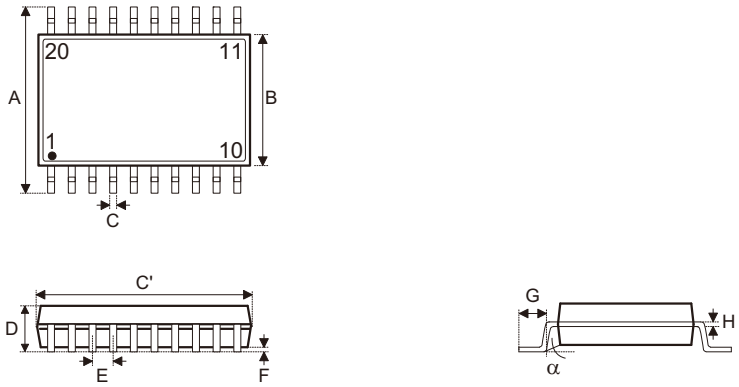
Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	24.89	—	26.92
B	6.10	—	7.11
C	2.92	—	4.95
D	2.92	—	3.81
E	0.36	—	0.56
F	1.14	—	1.78
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

MO-095a (see fig2)

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.945	—	0.985
B	0.275	—	0.295
C	0.120	—	0.150
D	0.110	—	0.150
E	0.014	—	0.022
F	0.045	—	0.060
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	24.00	—	25.02
B	6.99	—	7.49
C	3.05	—	3.81
D	2.79	—	3.81
E	0.36	—	0.56
F	1.14	—	1.52
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

20-pin SOP (300mil) Outline Dimensions

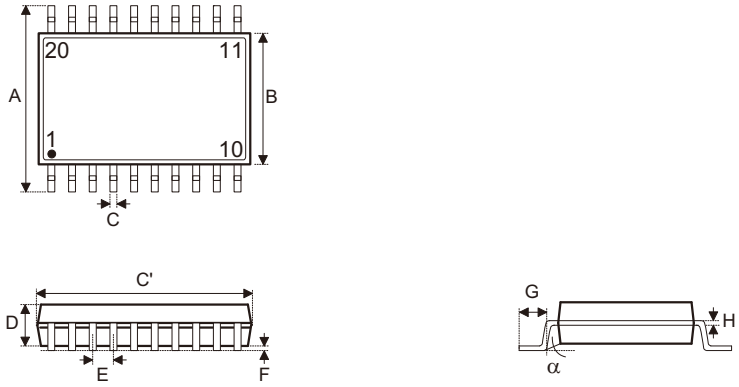


MS-013

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.393	—	0.419
B	0.256	—	0.300
C	0.012	—	0.020
C′	0.496	—	0.512
D	—	—	0.104
E	—	0.050	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	9.98	—	10.64
B	6.50	—	7.62
C	0.30	—	0.51
C′	12.60	—	13.00
D	—	—	2.64
E	—	1.27	—
F	0.10	—	0.30
G	0.41	—	1.27
H	0.20	—	0.33
α	0°	—	8°

20-pin SSOP (150mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.228	—	0.244
B	0.150	—	0.158
C	0.008	—	0.012
C′	0.335	—	0.347
D	0.049	—	0.065
E	—	0.025	—
F	0.004	—	0.010
G	0.015	—	0.050
H	0.007	—	0.010
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	5.79	—	6.20
B	3.81	—	4.01
C	0.20	—	0.30
C′	8.51	—	8.81
D	1.24	—	1.65
E	—	0.64	—
F	0.10	—	0.25
G	0.38	—	1.27
H	0.18	—	0.25
α	0°	—	8°

24-pin SKDIP (300mil) Outline Dimensions

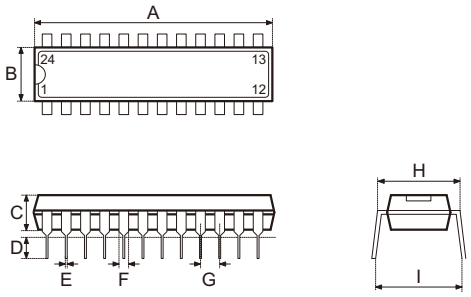


Fig1. Full Lead Packages

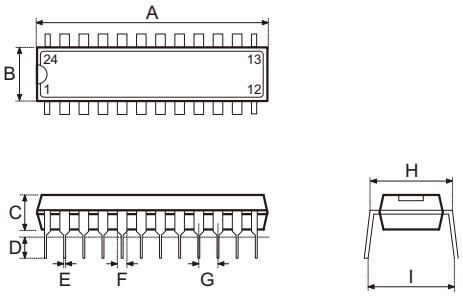


Fig2. 1/2 Lead Packages

MS-001d (see fig1)

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	1.230	—	1.280
B	0.240	—	0.280
C	0.115	—	0.195
D	0.115	—	0.150
E	0.014	—	0.022
F	0.045	—	0.070
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	31.24	—	32.51
B	6.10	—	7.11
C	2.92	—	4.95
D	2.92	—	3.81
E	0.36	—	0.56
F	1.14	—	1.78
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

MS-001d (see fig2)

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	1.160	—	1.195
B	0.240	—	0.280
C	0.115	—	0.195
D	0.115	—	0.150
E	0.014	—	0.022
F	0.045	—	0.070
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

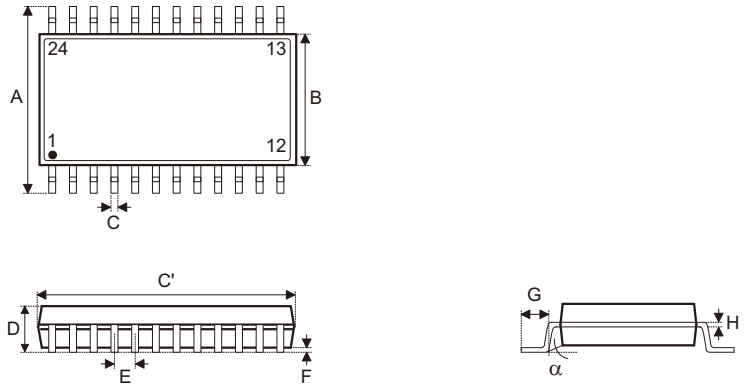
Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	29.46	—	30.35
B	6.10	—	7.11
C	2.92	—	4.95
D	2.92	—	3.81
E	0.36	—	0.56
F	1.14	—	1.78
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

MO-095a (see fig2)

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	1.145	—	1.185
B	0.275	—	0.295
C	0.120	—	0.150
D	0.110	—	0.150
E	0.014	—	0.022
F	0.045	—	0.060
G	—	0.100	—
H	0.300	—	0.325
I	—	0.430	—

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	29.08	—	30.10
B	6.99	—	7.49
C	3.05	—	3.81
D	2.79	—	3.81
E	0.36	—	0.56
F	1.14	—	1.52
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

24-pin SOP (300mil) Outline Dimensions

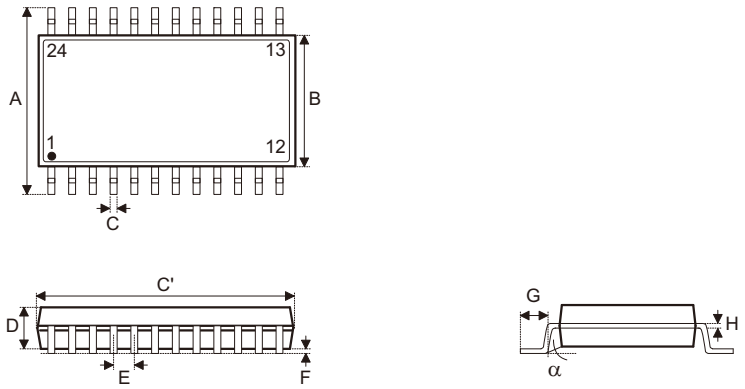


MS-013

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.393	—	0.419
B	0.256	—	0.300
C	0.012	—	0.020
C′	0.598	—	0.613
D	—	—	0.104
E	—	0.050	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	9.98	—	10.64
B	6.50	—	7.62
C	0.30	—	0.51
C′	15.19	—	15.57
D	—	—	2.64
E	—	1.27	—
F	0.10	—	0.30
G	0.41	—	1.27
H	0.20	—	0.33
α	0°	—	8°

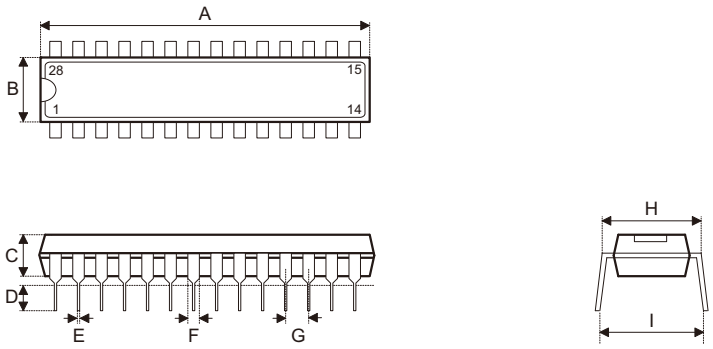
24-pin SSOP (150mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.228	—	0.244
B	0.150	—	0.157
C	0.008	—	0.012
C′	0.335	—	0.346
D	0.054	—	0.060
E	—	0.025	—
F	0.004	—	0.010
G	0.022	—	0.028
H	0.007	—	0.010
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	5.79	—	6.20
B	3.81	—	3.99
C	0.20	—	0.30
C′	8.51	—	8.79
D	1.37	—	1.52
E	—	0.64	—
F	0.10	—	0.25
G	0.56	—	0.71
H	0.18	—	0.25
α	0°	—	8°

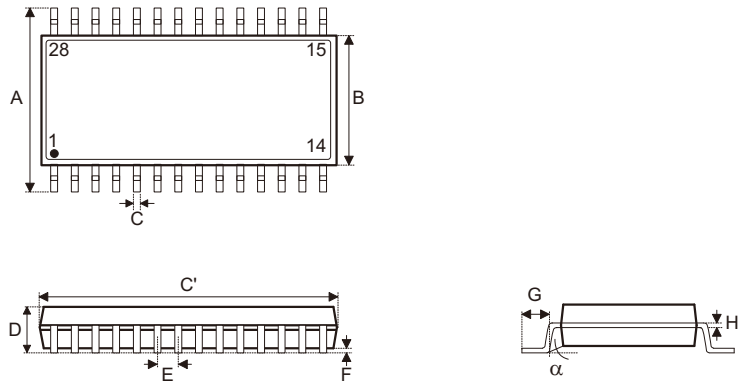
28-pin SKDIP (300mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	1.375	—	1.395
B	0.278	—	0.298
C	0.125	—	0.135
D	0.125	—	0.145
E	0.016	—	0.020
F	0.050	—	0.070
G	—	0.100	—
H	0.295	—	0.315
I	—	0.375	—

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	34.93	—	35.43
B	7.06	—	7.57
C	3.18	—	3.43
D	3.18	—	3.68
E	0.41	—	0.51
F	1.27	—	1.78
G	—	2.54	—
H	7.49	—	8.00
I	—	9.53	—

28-pin SOP (300mil) Outline Dimensions

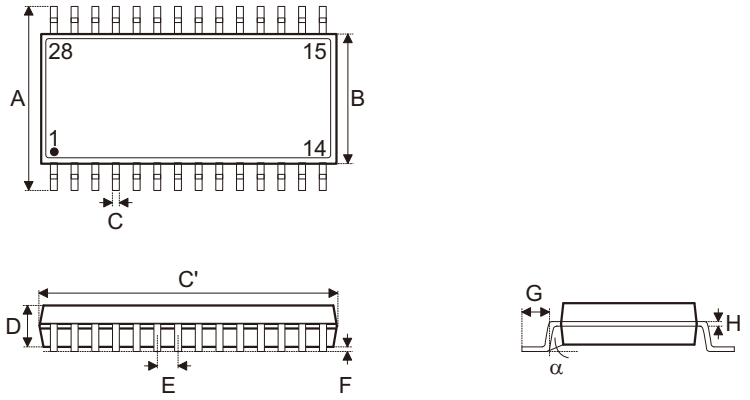


MS-013

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.393	—	0.419
B	0.256	—	0.300
C	0.012	—	0.020
C′	0.697	—	0.713
D	—	—	0.104
E	—	0.050	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	9.98	—	10.64
B	6.50	—	7.62
C	0.30	—	0.51
C′	17.70	—	18.11
D	—	—	2.64
E	—	1.27	—
F	0.10	—	0.30
G	0.41	—	1.27
H	0.20	—	0.33
α	0°	—	8°

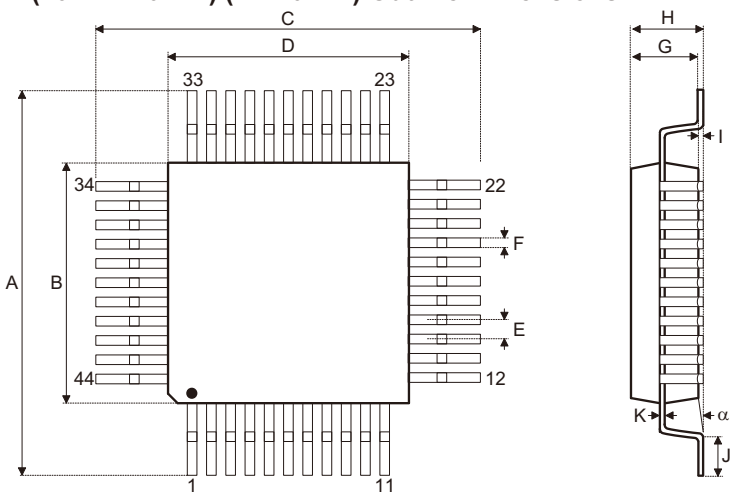
28-pin SSOP (150mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.228	—	0.244
B	0.150	—	0.157
C	0.008	—	0.012
C′	0.386	—	0.394
D	0.054	—	0.060
E	—	0.025	—
F	0.004	—	0.010
G	0.022	—	0.028
H	0.007	—	0.010
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	5.79	—	6.20
B	3.81	—	3.99
C	0.20	—	0.30
C′	9.80	—	10.01
D	1.37	—	1.52
E	—	0.64	—
F	0.10	—	0.25
G	0.56	—	0.71
H	0.18	—	0.25
α	0°	—	8°

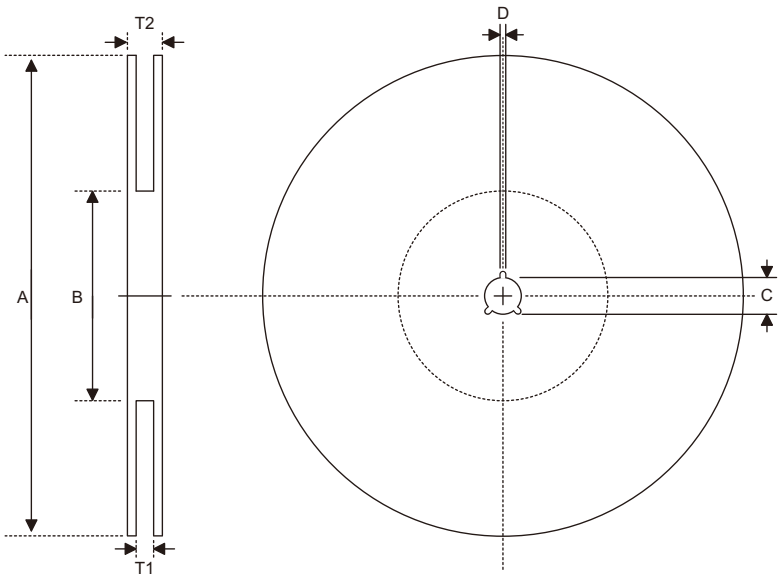
44-pin LQFP (10mm×10mm) (FP2.0mm) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.469	—	0.476
B	0.390	—	0.398
C	0.469	—	0.476
D	0.390	—	0.398
E	—	0.031	—
F	—	0.012	—
G	0.053	—	0.057
H	—	—	0.063
I	—	0.004	—
J	0.018	—	0.030
K	0.004	—	0.008
α	0°	—	7°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	11.90	—	12.10
B	9.90	—	10.10
C	11.90	—	12.10
D	9.90	—	10.10
E	—	0.80	—
F	—	0.30	—
G	1.35	—	1.45
H	—	—	1.60
I	—	0.10	—
J	0.45	—	0.75
K	0.10	—	0.20
α	0°	—	7°

Reel Dimensions



SOP 16N (150mil)

Symbol	Description	Dimensions in mm
A	Reel Outer Diameter	330.0±1.0
B	Reel Inner Diameter	100.0±1.5
C	Spindle Hole Diameter	13.0 $^{+0.5/-0.2}$
D	Key Slit Width	2.0±0.5
T1	Space Between Flange	16.8 $^{+0.3/-0.2}$
T2	Reel Thickness	22.2±0.2

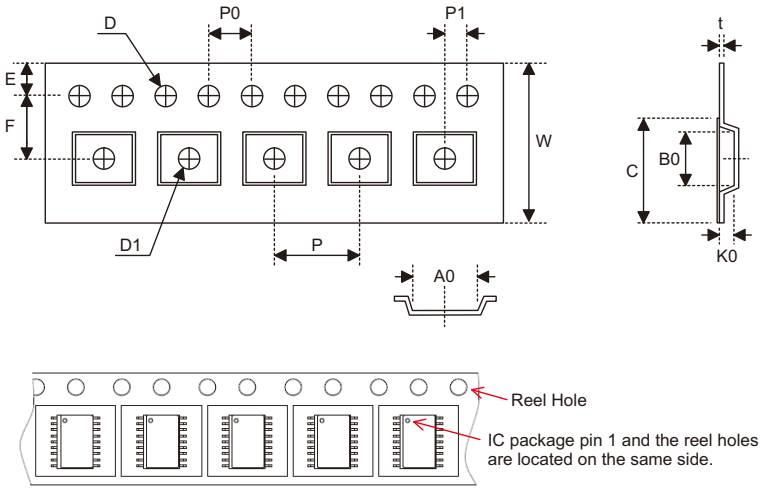
SOP 20W, SOP 24W, SOP 28W (300mil)

Symbol	Description	Dimensions in mm
A	Reel Outer Diameter	330.0±1.0
B	Reel Inner Diameter	100.0±1.5
C	Spindle Hole Diameter	13.0 $^{+0.5/-0.2}$
D	Key Slit Width	2.0±0.5
T1	Space Between Flange	24.8 $^{+0.3/-0.2}$
T2	Reel Thickness	30.2±0.2

SSOP 20S (150mil), SSOP 24S (150mil), SSOP 28S (150mil)

Symbol	Description	Dimensions in mm
A	Reel Outer Diameter	330.0±1.0
B	Reel Inner Diameter	100.0±1.5
C	Spindle Hole Diameter	13.0 $^{+0.5/-0.2}$
D	Key Slit Width	2.0±0.5
T1	Space Between Flange	16.8 $^{+0.3/-0.2}$
T2	Reel Thickness	22.2±0.2

Carrier Tape Dimensions



SOP 16N (150mil)

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	16.0±0.3
P	Cavity Pitch	8.0±0.1
E	Perforation Position	1.75±0.1
F	Cavity to Perforation (Width Direction)	7.5±0.1
D	Perforation Diameter	$1.55^{+0.10/-0.00}$
D1	Cavity Hole Diameter	$1.50^{+0.25/-0.00}$
P0	Perforation Pitch	4.0±0.1
P1	Cavity to Perforation (Length Direction)	2.0±0.1
A0	Cavity Length	6.5±0.1
B0	Cavity Width	10.3±0.1
K0	Cavity Depth	2.1±0.1
t	Carrier Tape Thickness	0.30±0.05
C	Cover Tape Width	13.3±0.1

SOP 20W

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	24.0 $^{+0.3/-0.1}$
P	Cavity Pitch	12.0±0.1
E	Perforation Position	1.75±0.10
F	Cavity to Perforation (Width Direction)	11.5±0.1
D	Perforation Diameter	1.5 $^{+0.1/-0.0}$
D1	Cavity Hole Diameter	1.50 $^{+0.25/-0.00}$
P0	Perforation Pitch	4.0±0.1
P1	Cavity to Perforation (Length Direction)	2.0±0.1
A0	Cavity Length	10.8±0.1
B0	Cavity Width	13.3±0.1
K0	Cavity Depth	3.2±0.1
t	Carrier Tape Thickness	0.30±0.05
C	Cover Tape Width	21.3±0.1

SOP 24W

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	24.0±0.3
P	Cavity Pitch	12.0±0.1
E	Perforation Position	1.75±0.1
F	Cavity to Perforation (Width Direction)	11.5±0.1
D	Perforation Diameter	1.55 $^{+0.10/-0.00}$
D1	Cavity Hole Diameter	1.50 $^{+0.25/-0.00}$
P0	Perforation Pitch	4.0±0.1
P1	Cavity to Perforation (Length Direction)	2.0±0.1
A0	Cavity Length	10.9±0.1
B0	Cavity Width	15.9±0.1
K0	Cavity Depth	3.1±0.1
t	Carrier Tape Thickness	0.35±0.05
C	Cover Tape Width	21.3±0.1

SOP 28W (300mil)

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	24.0±0.3
P	Cavity Pitch	12.0±0.1
E	Perforation Position	1.75±0.10
F	Cavity to Perforation (Width Direction)	11.5±0.1
D	Perforation Diameter	$1.5^{+0.1/-0.0}$
D1	Cavity Hole Diameter	$1.50^{+0.25/-0.00}$
P0	Perforation Pitch	4.0±0.1
P1	Cavity to Perforation (Length Direction)	2.0±0.1
A0	Cavity Length	10.85±0.10
B0	Cavity Width	18.34±0.10
K0	Cavity Depth	2.97±0.10
t	Carrier Tape Thickness	0.35±0.01
C	Cover Tape Width	21.3±0.1

SSOP 20S (150mil)

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	$16.0^{+0.3/-0.1}$
P	Cavity Pitch	8.0±0.1
E	Perforation Position	1.75±0.10
F	Cavity to Perforation (Width Direction)	7.5±0.1
D	Perforation Diameter	$1.5^{+0.1/-0.0}$
D1	Cavity Hole Diameter	$1.50^{+0.25/-0.00}$
P0	Perforation Pitch	4.0±0.1
P1	Cavity to Perforation (Length Direction)	2.0±0.1
A0	Cavity Length	6.5±0.1
B0	Cavity Width	9.0±0.1
K0	Cavity Depth	2.3±0.1
t	Carrier Tape Thickness	0.30±0.05
C	Cover Tape Width	13.3±0.1

SSOP 24S (150mil)

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	16.0 $^{+0.3/-0.1}$
P	Cavity Pitch	8.0±0.1
E	Perforation Position	1.75±0.10
F	Cavity to Perforation (Width Direction)	7.5±0.1
D	Perforation Diameter	1.5 $^{+0.1/-0.0}$
D1	Cavity Hole Diameter	1.50 $^{+0.25/-0.00}$
P0	Perforation Pitch	4.0±0.1
P1	Cavity to Perforation (Length Direction)	2.0±0.1
A0	Cavity Length	6.5±0.1
B0	Cavity Width	9.5±0.1
K0	Cavity Depth	2.1±0.1
t	Carrier Tape Thickness	0.30±0.05
C	Cover Tape Width	13.3±0.1

SSOP 28S (150mil)

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	16.0±0.3
P	Cavity Pitch	8.0±0.1
E	Perforation Position	1.75±0.1
F	Cavity to Perforation (Width Direction)	7.5±0.1
D	Perforation Diameter	1.55 $^{+0.10/-0.00}$
D1	Cavity Hole Diameter	1.50 $^{+0.25/-0.00}$
P0	Perforation Pitch	4.0±0.1
P1	Cavity to Perforation (Length Direction)	2.0±0.1
A0	Cavity Length	6.5±0.1
B0	Cavity Width	10.3±0.1
K0	Cavity Depth	2.1±0.1
t	Carrier Tape Thickness	0.30±0.05
C	Cover Tape Width	13.3±0.1

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 (China) Inc.
Building No. 10, Xinzhu Court, (No. 1 Headquarters), 4 Cuizhu Road, Songshan Lake, Dongguan, China 523808
Tel: 86-769-2626-1300
Fax: 86-769-2626-1311

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