HOLTEK

Handheld Product Flash MCU with Two Cell Li-Batteries

BP45F1632

Revision: V1.00 Date: December 28, 2020

www.holtek.com

Table of Contents

Features

CPU Features

- Operating voltage
  - f_{SYS}=7.5MHz: 2.6V~5.5V
  - f_{SYS}=15MHz: 4.5V~5.5V
- Up to 0.27µs instruction cycle with 15MHz system clock at V_{DD}=5V
- Power down and wake-up functions to reduce power consumption
- Oscillator types:
  - Internal High Speed 30MHz RC – HIRC
  - Internal Low Speed 32kHz RC – LIRC
- Multi-mode operation: FAST, SLOW, IDLE and SLEEP
- Fully integrated internal oscillators require no external components
- All instructions executed in one or two instruction cycles
- Table read instructions
- 61 powerful instructions
- 4-level subroutine nesting
- Bit manipulation instruction

Peripheral Features

- Flash Program Memory: 2K×16
- Data Memory: 128×8
- Watchdog Timer function
- Up to 8 bidirectional I/O lines
- Four high voltage output (HVO) functions
- Programmable PA port source current for LED applications
- Two external interrupt lines shared with I/O pins
- Multiple Timer Modules for time measurement, compare match output or PWM output or single pulse output function
  - One 10-bit CTM
  - One 10-bit PTM
- Complementary PWM output with dead time
- Over current protection (OCP) with interrupt
- Over voltage protection (OVP) with interrupt
- 5 external channel 12-bit resolution A/D converter with internal reference voltage V_{VR}
- LDO function with multiple voltage divider resistors
- Dual Time-Base functions for generation of fixed time interrupt signals
- Low voltage reset function
- Low voltage detect function
- 1-channel H-Bridge Driver
- Package type: 24-pin SSOP-EP

General Description

The device is a Flash Memory 8-bit high performance RISC architecture microcontroller, specifically designed for handheld products with two cell li-batteries.

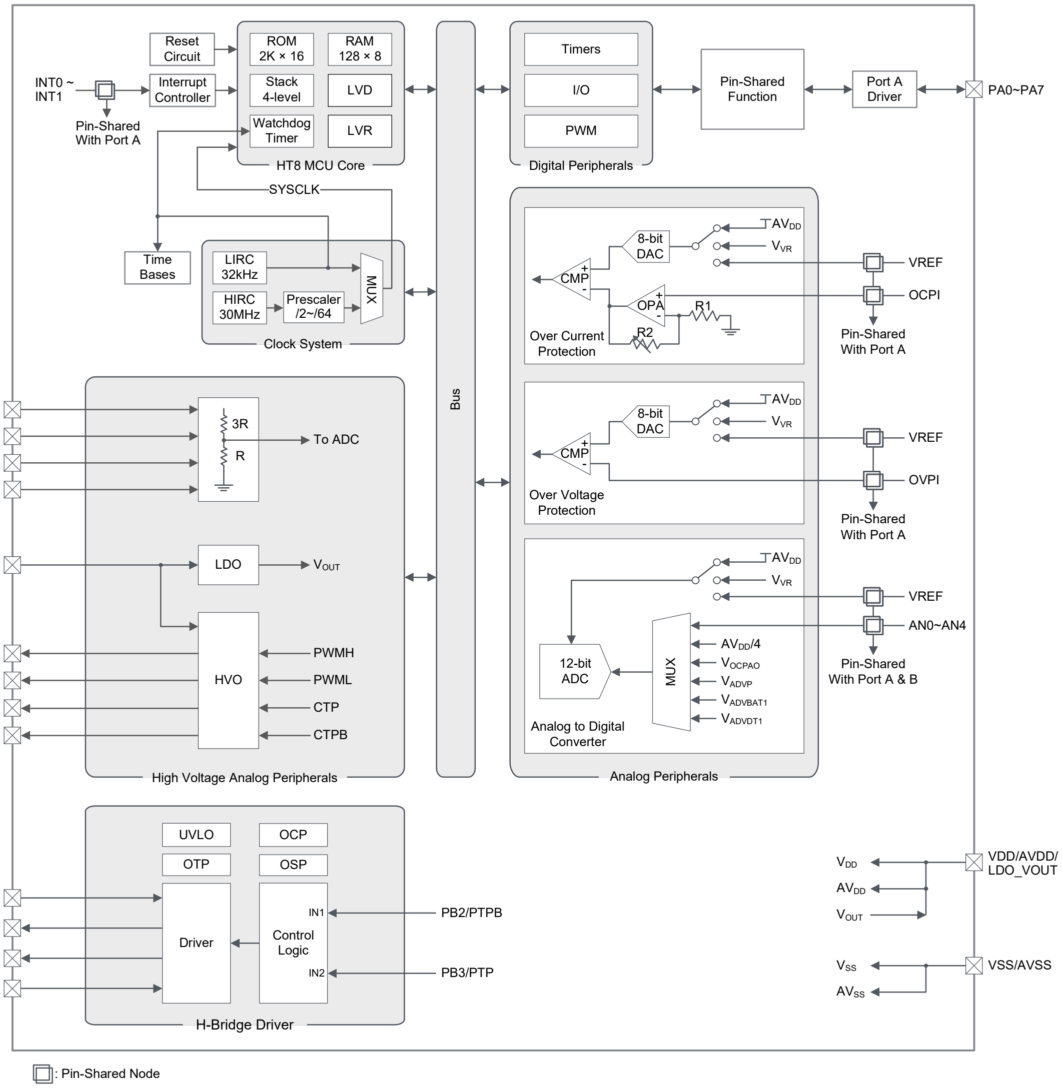
For memory features, the Flash Memory offers users the convenience of multi-programming features. Other memory includes an area of RAM Data Memory. Analog features include a multi-channel 12-bit A/D converter. Multiple extremely flexible Timer Modules provide timing, pulse generation and PWM generation functions. Protective features such as an internal Watchdog Timer, Low Voltage Reset and Low Voltage Detector coupled with excellent noise immunity and ESD protection ensure that reliable operation is maintained in hostile electrical environments.

The device also includes fully integrated high and low speed oscillators which require no external components for their implementation. The ability to operate and switch dynamically between a range of operating modes using different clock sources gives users the ability to optimize microcontroller operation and minimize power consumption.

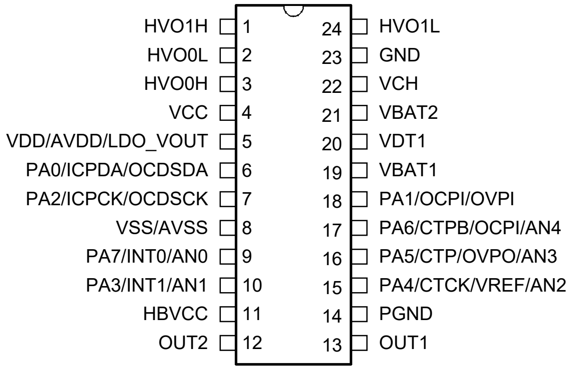
The inclusion of flexible I/O programming features, Time-Base functions along with many other features ensure that the device will find excellent use in different handheld products with two cell li-batteries.

Circuitry specific to handheld products with two cell li-batteries is also fully integrated within the device. These include functions such as LDO, high voltage output, H-bridge driver, over voltage protection and over current protection. These features combine to ensure that a minimum of external components is required to implement handheld products with two cell li-batteries, providing the benefits of reduced component counts and reduced circuit board areas.

Block Diagram



Pin Assignment



BP45F1632/BP45V1632
24 SSOP-EP-A

Note: 1. If the pin-shared pin functions have multiple outputs simultaneously, the desired pin-shared function is determined by the corresponding software control bits.

2. The OCDSDA and OCDSCK pins are supplied for the OCDS dedicated pins and as such only available for the BP45V1632 device which is the OCDS EV chip for the BP45F1632 device.

3. For the unbonded lines, PB0 and PB1, the line status should be properly configured to avoid unwanted power consumption resulting from floating input conditions. Refer to the "Standby Current Considerations" and "Input/Output Ports" sections.

Pin Description

The function of each pin is listed in the following table, however the details behind how each pin is configured is contained in other sections of the datasheet. As each Pin Description table shows the situation for the package with the most pins, not all pins in the tables will be available on smaller package sizes.

Pin Name	Function	OPT	I/T	O/T	Description
PA0/ICPDA/OCDSDA	PA0	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	ICPDA	—	ST	CMOS	ICP data/address pin
	OCDSDA	—	ST	CMOS	OCDS data/address pin, for EV chip only
PA1/OCPI/OVPI	PA1	PAWU PAPU PAS0	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	OCPI	PAS0	AN	—	OCP input
	OVPI	PAS0	AN	—	OVP input
PA2/ICPCK/OCDSCK	PA2	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	ICPCK	—	ST	—	ICP clock pin
	OCDSCK	—	ST	—	OCDS clock pin, for EV chip only
PA3/INT1/AN1	PA3	PAWU PAPU PAS0	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	INT1	INTEG INTC1 PAS0	ST	—	External interrupt 1 input
	AN1	PAS0	AN	—	A/D converter external input channel 1

Pin Name	Function	OPT	I/T	O/T	Description
PA4/CTCK/VREF/AN2	PA4	PAWU PAPU PAS1	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	CTCK	PAS1	ST	—	CTM clock input
	VREF	PAS1	AN	—	ADC/OCP/OVP external reference voltage input
	AN2	PAS1	AN	—	A/D converter external input channel 2
PA5/CTP/OVPO/AN3	PA5	PAWU PAPU PAS1	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	CTP	PAS1	—	CMOS	CTM output
	OVPO	PAS1	—	CMOS	OVP comparator output (after debounce)
	AN3	PAS1	AN	—	A/D converter external input channel 3
PA6/CTPB/OCPI/AN4	PA6	PAWU PAPU PAS1	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	CTPB	PAS1	—	CMOS	CTM inverted output
	OCPI	PAS1	AN	—	OCP input
	AN4	PAS1	AN	—	A/D converter external input channel 4
PA7/INT0/AN0	PA7	PAWU PAPU PAS1	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	INT0	INTEG INTC0 PAS1	ST	—	External interrupt 0 input
	AN0	PAS1	AN	—	A/D converter external input channel 0
VDD/AVDD/ LDO_VOUT	VDD	—	PWR	—	Digital positive power supply
	AVDD	—	PWR	—	Analog positive power supply
	LDO_VOUT	—	—	PWR	LDO output
VSS/AVSS	VSS	—	PWR	—	Digital negative power supply
	AVSS	—	PWR	—	Analog negative power supply
VCC	VCC	—	PWR	—	High voltage positive power supply
GND	GND	—	PWR	—	High voltage negative power supply
VCH	VCH	—	AN	—	Charger voltage input
VDT1	VDT1	—	AN	—	Voltage detect 1 input
VBAT2	VBAT2	—	AN	—	Battery cell 2 positive
VBAT1	VBAT1	—	AN	—	Battery cell 1 positive and Battery cell 2 negative
HVO0H	HVO0H	—	—	CMOS	High voltage output
HVO0L	HVO0L	—	—	CMOS	High voltage output
HVO1H	HVO1H	—	—	CMOS	High voltage output
HVO1L	HVO1L	—	—	CMOS	High voltage output

Legend: I/T: Input type;
OPT: Optional by register option;
ST: Schmitt Trigger input;
AN: Analog signal.
O/T: Output type;
PWR: Power;
CMOS: CMOS output;

H-Bridge Driver Pin Description

Name	Type	Description
HBVCC	P	H-Bridge driver motor power supply
OUT1	O	H-Bridge output 1
PGND	G	Motor current sensing terminal Connect via a sensing resistor to ground. If it is not necessary to sense the motor current, the PGND line should be directly connected to VSS.
OUT2	O	H-Bridge output 2

Legend: I: Input; O: Output; P: Power; G: Ground.

Interconnection Signal Description

Several signals are not connected to external package pins. These signals are interconnection lines between the MCU and the H-Bridge driver and are listed in the following table.

MCU Signal	H-Bridge Signal	Function	Description
PB2/PTPB	IN1	PB2	General purpose I/O. Register enabled pull-high Internally connected to the H-bridge driver input IN1.
		PTPB	PTM inverted output Internally connected to the H-bridge driver input IN1.
		IN1	H-Bridge driver input 1 with internal pull-low Internally connected to the MCU PB2/PTPB line.
PB3/PTP	IN2	PB3	General purpose I/O. Register enabled pull-high Internally connected to the H-bridge driver input IN2.
		PTP	PTM output Internally connected to the H-bridge driver input IN2.
		IN2	H-Bridge driver input 2 with internal pull-low Internally connected to the MCU PB3/PTP line.

Note: The internal signals, PB2/PTPB and PB3/PTP, are internally connected to the H-Bridge driver inputs, IN1 and IN2, respectively which should be properly configured to control the H-Bridge driver. Refer to the “H-Bridge Driver” chapter for more details.

Absolute Maximum Ratings

Supply Voltage	V_{SS}-0.3V to 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_{OH} Total	-90mA
I_{OL}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 the 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

For data in the following tables, note that factors such as oscillator type, operating voltage, operating frequency, pin load conditions, temperature and program instruction type, etc., can all exert an influence on the measured values.

Operating Voltage Characteristics

Ta=-40°C~85°C

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{DD}	Operating Voltage – HIRC	$f_{SYS}=f_{HIRC}/4=7.5MHz$	2.6	—	5.5	V
		$f_{SYS}=f_{HIRC}/2=15MHz$	3.3	—	5.5	
	Operating Voltage – LIRC	$f_{SYS}=32kHz$	2.6	—	5.5	

Operating Current Characteristics

Ta=-40°C~85°C

Symbol	Operating Mode	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{DD}	SLOW Mode – LIRC	2.6V	$f_{SYS}=32kHz$	—	8	16	μA
		3V		—	10	20	
		5V		—	30	50	
	FAST Mode – HIRC	2.6V	$f_{SYS}=f_{HIRC}/4=7.5MHz$	—	1.7	2.1	mA
		3V		—	2.1	2.5	
		5V		—	4.6	5.8	
		3.3V	$f_{SYS}=f_{HIRC}/2=15MHz$	—	3.1	3.7	
		5V		—	5.0	6.3	

Note: When using the characteristic table data, the following notes should be taken into consideration:

1. Any digital inputs are setup in a non-floating condition.
2. All measurements are taken under conditions of no load and with all peripherals in an off state.
3. There are no DC current paths.
4. All Operating Current values are measured using a continuous NOP instruction program loop.

Standby Current Characteristics

Ta=25°C, unless otherwise specified

Symbol	Standby Mode	Test Conditions		Min.	Typ.	Max.	Max. @85°C	Unit
		V_{DD}	Conditions					
I_{STB}	SLEEP Mode	2.6V	WDT off	—	0.08	0.12	1.40	μA
		3V		—	0.08	0.12	1.40	
		5V		—	0.15	0.29	2.20	
		2.6V	WDT on	—	2.4	4.0	4.6	
		3V		—	3.0	5.0	5.7	
		5V		—	5	10	11	
	IDLE0 Mode – LIRC	2.6V	f_{SUB} on	—	2.4	4.0	4.6	μA
		3V		—	3.0	5.0	5.7	
		5V		—	5	10	11	
	IDLE1 Mode – HIRC	2.6V	f_{SUB} on, $f_{SYS}=f_{HIRC}/4=7.5MHz$	—	1.5	2.3	2.3	mA
		3V		—	1.9	2.8	2.8	
		5V		—	3.8	5.6	5.6	
		3.3V	f_{SUB} on, $f_{SYS}=f_{HIRC}/2=15MHz$	—	2.3	2.7	2.7	
		5V		—	4.2	5.0	5.0	

Note: When using the characteristic table data, the following notes should be taken into consideration:

1. Any digital input is setup in a non-floating condition.
2. All measurements are taken under conditions of no load and with all peripherals in an off state.
3. There are no DC current paths.
4. All Standby Current values are taken after a HALT instruction executed thus stopping all instruction execution.

A.C. Characteristics

For data in the following tables, note that factors such as oscillator type, operating voltage, operating frequency and temperature etc., can all exert an influence on the measured values.

High Speed Internal Oscillator – HIRC – Frequency Accuracy

During the program writing operation the writer will trim the HIRC oscillator at a user selected HIRC frequency and user selected voltage of 5V.

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
f_{HIRC}	30MHz Writer Trimmed HIRC Frequency	5V	Ta=25°C, $f_{SYS}=f_{HIRC}/4=7.5MHz$	-2%	30	+2%	MHz
		5V	Ta=-40°C~85°C, $f_{SYS}=f_{HIRC}/4=7.5MHz$	-7%	30	+7%	
		2.6V~5.5V	Ta=-40°C~85°C, $f_{SYS}=f_{HIRC}/4=7.5MHz$	-18%	30	+18%	

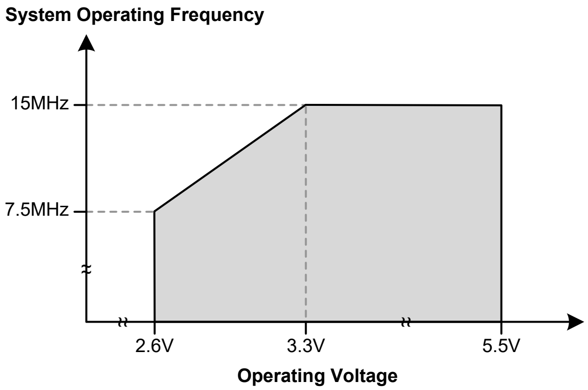
Note: 1. The 5V values for V_{DD} are provided as this is the fixed voltage at which the HIRC frequency is trimmed by the writer.

2. The row below the 5V trim voltage row is provided to show the values for the full V_{DD} range operating voltage.

Low Speed Internal Oscillator Characteristics – LIRC

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Temperature				
f_{LIRC}	LIRC Frequency	5V	25°C	25.6	32	38.4	kHz
		2.6V~5.5V	25°C	12.8	32	41.6	
			-40°C~85°C	8	32	60	
t_{START}	LIRC Start Up Time	—	25°C	—	—	100	μs

Operating Frequency Characteristic Curves



System Start Up Time Characteristics

Ta=-40°C~85°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
t_{SST}	System Start-up Time Wake-up from Condition where f_{SYS} is off	—	$f_{SYS}=f_H/2\sim f_H/64$, $f_H=f_{HIRC}$	—	16	—	t_{SYS}
		—	$f_{SYS}=f_{SUB}=f_{LIRC}$	—	2	—	t_{SYS}
	System Start-up Time Wake-up from condition where f_{SYS} is on	—	$f_{SYS}=f_H/2\sim f_H/64$, $f_H=f_{HIRC}$	—	2	—	t_{SYS}
		—	$f_{SYS}=f_{SUB}=f_{LIRC}$	—	2	—	t_{SYS}
	System Speed Switch Time FAST to SLOW Mode or SLOW to FAST Mode	—	f_{HIRC} switches from off → on	—	16	—	t_{HIRC}
t_{RSTD}	System Reset Delay Time Reset Source from Power-on reset or LVR Hardware Reset	—	RR_{POR}=5V/ms	8.3	16.7	50.0	ms
	System Reset Delay Time LVRC/WDTC Software Reset	—	—				
	System Reset Delay Time Reset Source from WDT Overflow	—	—	8.3	16.7	50.0	ms
t_{SRESET}	Minimum Software Reset Width to Reset	—	—	45	90	375	μs

Note: 1. For the System Start-up time values, whether f_{SYS} is on or off depends upon the mode type and the chosen f_{SYS} system oscillator. Details are provided in the System Operating Modes section.

2. The time units, shown by the symbols t_{HIRC} etc. are the inverse of the corresponding frequency values as provided in the frequency tables. For example $t_{HIRC}=1/f_{HIRC}$, $t_{SYS}=1/f_{SYS}$ etc.

3. If the LIRC is used as the system clock and if it is off when in the SLEEP Mode, then an additional LIRC start up time, t_{START}, as provided in the LIRC frequency table, must be added to the t_{SST} time in the table above.

4. The System Speed Switch Time is effectively the time taken for the newly activated oscillator to start up.

Input/Output Characteristics

Ta=-40°C~85°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{IL}	Input Low Voltage for I/O Ports or Input Pins	5V	—	0	—	1.5	V
		—	—	0	—	$0.2V_{DD}$	
V_{IH}	Input High Voltage for I/O Ports or Input Pins	5V	—	3.5	—	5.0	V
		—	—	$0.8V_{DD}$	—	V_{DD}	
I_{OL}	Sink Current for I/O Ports	3V	$V_{OL}=0.1V_{DD}$	16	32	—	mA
		5V		32	65	—	
I_{OH}	Source Current for PA Port	3V	$V_{OH}=0.9V_{DD}$, SLEDC[m+1:m]=00B (m=0 or 2)	-0.7	-1.5	—	mA
		5V		-1.5	-2.9	—	
		3V	$V_{OH}=0.9V_{DD}$, SLEDC[m+1:m]=01B (m=0 or 2)	-1.3	-2.5	—	
		5V		-2.5	-5.1	—	
		3V	$V_{OH}=0.9V_{DD}$, SLEDC[m+1:m]=10B (m=0 or 2)	-1.8	-3.6	—	
		5V		-3.6	-7.3	—	
		3V	$V_{OH}=0.9V_{DD}$, SLEDC[m+1:m]=11B (m=0 or 2)	-4	-8	—	
		5V		-8	-16	—	
R_{PH}	Pull-high Resistance for I/O Ports	3V	—	20	60	100	kΩ
		5V	—	10	30	50	

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{LEAK}	Input Leakage Current	5V	$V_{IN}=V_{DD}$ or V_{SS}	—	—	±1	μA
t_{TCK}	TM Clock Input Pin Minimum Pulse Width	—	—	0.3	—	—	μs
t_{INT}	External Interrupt Minimum Pulse Width	—	—	10	—	—	μs

Note: The R_{PH} internal pull-high resistance value is calculated by connecting to ground and enabling the input pin with a pull-high resistor and then measuring the pin current at the specified supply voltage level. Dividing the voltage by this measured current provides the R_{PH} value.

Memory Characteristics

Ta=-40°C~85°C, unless otherwise specified

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
Flash Program Memory							
t_{DEW}	Erase/Write Time	—	—	—	2	3	ms
E_P	Cell Endurance	—	—	10K	—	—	E/W
t_{RETD}	ROM Data Retention time	—	Ta=25°C	—	40	—	Year
RAM Data Memory							
V_{DR}	RAM Data Retention voltage	—	Device in SLEEP Mode	1.0	—	—	V

Note: "E/W" means Erase/Write times.

LVR/LVD Electrical Characteristics

Ta=-40°C~85°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{LVR}	Low Voltage Reset Voltage	—	LVR enable, voltage select 2.1V	-5%	2.1	+5%	V
		—	LVR enable, voltage select 2.55V		2.55		
		—	LVR enable, voltage select 3.15V		3.15		
		—	LVR enable, voltage select 3.8V		3.8		
V_{LVD}	Low Voltage Detection Voltage	—	LVD enable, voltage select 2.0V	-5%	2.0	+5%	V
		—	LVD enable, voltage select 2.2V		2.2		
		—	LVD enable, voltage select 2.4V		2.4		
		—	LVD enable, voltage select 2.7V		2.7		
		—	LVD enable, voltage select 3.0V		3.0		
		—	LVD enable, voltage select 3.3V		3.3		
		—	LVD enable, voltage select 3.6V		3.6		
		—	LVD enable, voltage select 4.0V		4.0		
I_{LVRLVD}	Operating Current	3V	LVD enable, LVR enable, $V_{LVR}=2.1V$, $V_{LVD}=2.2V$	—	—	10	μA
		5V		—	8	15	
t_{LVDS}	LVDO Stable Time	—	For LVR enable, LVD off → on	—	—	18	μs
t_{LVR}	Minimum Low Voltage Width to Reset	—	—	140	600	1000	μs
t_{LVD}	Minimum Low voltage Width to Interrupt	—	—	40	150	320	μs
I_{LVR}	Additional Current for LVR Enable	—	LVD disable	—	—	8	μA

A/D Converter Electrical Characteristics

Ta=-40°C~85°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.6	—	5.5	V
V_{ADI}	Input Voltage	—	—	0	—	V_{REF}	V
V_{REF}	Reference Voltage	—	—	2	—	V_{DD}	V
N_R	Resolution	—	—	—	—	12	Bit
DNL	Differential Non-linearity	—	$V_{REF}=V_{DD}$, $t_{ADCK}=0.5\mu s$	-3	—	+3	LSB
INL	Integral Non-linearity	—	$V_{REF}=V_{DD}$, $t_{ADCK}=0.5\mu s$	-4	—	+4	LSB
I_{ADC}	Additional Current for A/D Converter Enable	3V	No load, $t_{ADCK}=0.5\mu s$	—	340	500	μA
		5V		—	500	700	
t_{ADCK}	Clock Period	—	—	0.5	—	10.0	μs
t_{ON2ST}	A/D Converter On-to-start Time	—	—	4	—	—	μs
t_{ADS}	Sampling Time	—	—	—	4	—	t_{ADCK}
t_{ADC}	Conversion Time (Include A/D Sample and Hold Time)	—	—	—	16	—	t_{ADCK}
GERR	A/D Conversion Gain Error	—	$V_{REF}=V_{DD}$	-4	—	4	LSB
OSRR	A/D Conversion Offset Error	—	$V_{REF}=V_{DD}$	-4	—	4	LSB
V_{VR}	OPA Output Voltage	2.6V~ 5.5V	—	-1%	2.4	+1%	V

Over Voltage Protection Electrical Characteristics

Ta=-40°C~85°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{REF}	DAC Reference Voltage	3V	OVPVRS[1:0]=01B	1.8	—	V_{DD}	V
		5V		1.8	—	V_{DD}	
I_{OVP}	Operating Current	3V	OVPEN=1, DAC V_{REF}=2.5V	—	—	350	μA
		5V		—	280	400	
V_{OS}	Input Offset Voltage	3V	With calibration	-2	—	2	mV
		5V		-2	—	2	
V_{HYS}	Hysteresis	3V	—	10	45	70	mV
		5V		10	45	70	
V_{CM}	Common Mode Voltage Range	3V	—	V_{SS}	—	V_{DD}-1.0	V
		5V		V_{SS}	—	V_{DD}-1.0	
DNL	Differential Non-linearity	3V	DAC $V_{REF}=V_{DD}$	-1.5	—	+1.5	LSB
		5V		-1	—	+1	
INL	Integral Non-linearity	3V	DAC $V_{REF}=V_{DD}$	-2	—	+2	LSB
		5V		-1.5	—	+1.5	

Over Current Protection Electrical Characteristics

Ta=-40°C~85°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{REF}	DAC Reference Voltage	3V	OCPVRS[1:0]=01B	1.8	—	V_{DD}	V
		5V		1.8	—	V_{DD}	
V_{OS_CMP}	Comparator Input Offset Voltage	3V	Without calibration (OCPCOF[4:0]=10000B)	-15	—	15	mV
		5V		-15	—	15	
		3V	With calibration	-2	—	2	
		5V		-2	—	2	
V_{HYS}	Hysteresis	3V	—	10	40	60	mV
		5V		10	40	60	
V_{CM_CMP}	Comparator Common Mode Voltage Range	3V	—	V_{SS}	—	V_{DD}-1.0	V
		5V		V_{SS}	—	V_{DD}-1.0	
V_{OS_OPA}	OPA Input Offset Voltage	3V	Without calibration (OCPOOF[5:0]=100000B)	-15	—	15	mV
		5V		-15	—	15	
		3V	With calibration	-2	—	2	
		5V		-2	—	2	
V_{CM_OPA}	OPA Common Mode Voltage Range	3V	—	V_{SS}	—	V_{DD}-1.4	V
		5V		V_{SS}	—	V_{DD}-1.4	
V_{OR}	OPA Maximum Output Voltage Range	3V	—	V_{SS}+0.1	—	V_{DD}-0.1	V
		5V		V_{SS}+0.1	—	V_{DD}-0.1	
Ga	PGA Gain Accuracy	3V	All gain	-5	—	5	%
		5V		-5	—	5	
DNL	Differential Non-linearity	3V	DAC V_{REF}=V_{DD}	-1.5	—	+1.5	LSB
		5V		-1	—	+1	
INL	Integral Non-linearity	3V	DAC V_{REF}=V_{DD}	-2	—	+2	LSB
		5V		-1.5	—	+1.5	

High Voltage Output Electrical Characteristics

Ta=-40°C~85°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{IN}	Input Voltage	—	—	V_{DD}	—	12	V
I_{OH}	Source Current for High Voltage Output Pins	—	V_{OH}=0.9×V_{IN}, V_{IN}=10V	-90	—	—	mA
I_{OL}	Sink Current for High Voltage Output Pins	—	V_{OL}=0.1×V_{IN}, V_{IN}=10V	100	—	—	mA
R_{PH}	Pull-high Resistance for High Voltage Output Pins	—	—	10	20	40	kΩ
R_{PL}	Pull-low Resistance for High Voltage Output Pins	—	—	10	20	40	kΩ

LDO Electrical Characteristics

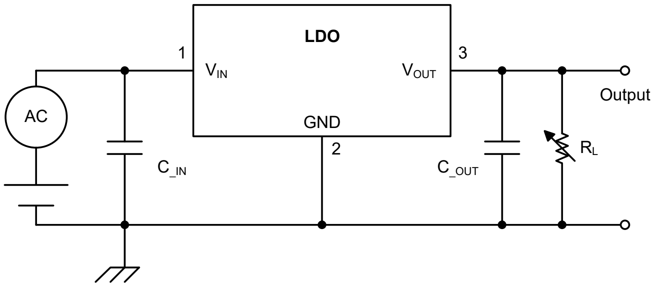
C_{LOAD}=10μF+0.1μF, V_{IN}=V_{OUT}+1V, Ta=-40°C~85°C, unless otherwise specified

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{IN}	Conditions				
V_{IN}	LDO Input Voltage	—	—	6	—	12	V
V_{IN2}	Input Voltage for VBAT2, VCH, VDT1	—	—	0	—	12	V
V_{IN3}	Input Voltage for VBAT1	—	—	0	—	5	V
V_{OUT}	Output Voltage	—	Ta=25°C, I_{LOAD}=1mA, V_{IN}=V_{OUT}+1V	-2%	5.0	2%	V
		—	Ta=-40°C~85°C, I_{LOAD}=1mA, V_{IN}=V_{OUT}+1V	-5%	5.0	5%	
ΔV_{LOAD}	Load Regulation (1)	—	1mA≤I_{LOAD}≤70mA, V_{IN}=V_{OUT}+1V	—	0.015	0.033	%/mA
V_{DROP}	Dropout Voltage (2)	—	ΔV_{OUT}=2%, I_{LOAD}=1mA, V_{IN}=V_{OUT}+1V	—	—	100	mV
I_{OUT}	Output Current	—	V_{IN}=V_{OUT}+1V, ΔV_{OUT}=-3%	70	—	—	mA
		—	V_{IN}=V_{OUT}+2V, ΔV_{OUT}=-3%	150	—	—	mA
I_Q	Quiescent Current	12V	No load	—	3	6	μA
ΔV_{LINE}	Line Regulation	—	6V≤V_{IN}≤12V, I_{LOAD}=1mA	—	—	0.2	%/V
TC	Temperature Coefficient	—	Ta=-40°C~85°C, V_{IN}=V_{OUT}+1V, I_{LOAD}=10mA	—	±1.5	±2	mV/°C
ΔV_{OUT_RIPPLE}	Output Voltage Ripple	6V	V_{IN}=6V, I_{LOAD}=10mA	—	—	40	mV
RR	Ripple Rejection (3)	—	V_{IN}=10V_{DC}+2$V_{P-P(AC)}$, I_{LOAD}≤50mA, f=120Hz	35	—	—	dB
t_{START}	LDO Startup Time	6V	V_{IN}=6V, I_{LOAD}=1mA, V_{OUT}=5V±5%	—	—	10	ms
R_{VIN2}	The Sum of 3R and 1R for VBAT2, VCH, VDT1	6V	V_{IN2}=8.4V	4	8	12	kΩ
R_{VIN3}	The Sum of 3R and 1R for VBAT1	6V	V_{IN3}=4.2V	4	8	12	kΩ
RR_{VIN2}	The Ratio of 3R/1R for VBAT2, VCH, VDT1	6V	V_{IN2}=8.4V	-1%	3	+1%	—
RR_{VIN3}	The Ratio of 3R/1R for VBAT1	6V	V_{IN3}=4.2V	-1%	3	+1%	—
V_{IL}	Input Low Voltage for VCH	—	V_{OUT}=5V	—	—	0.3	V
V_{IH}	Input High Voltage for VCH	—	V_{OUT}=5V	0.9	—	—	V

Note: 1. Load regulation is measured at a constant junction temperature, using pulse testing with a low ON time and is guaranteed up to the maximum power dissipation. Power dissipation is determined by the input/output differential voltage and the output current. Guaranteed maximum power dissipation will not be available over the full input/output range. The maximum allowable power dissipation at any ambient temperature is $P_D=(T_{J(MAX)}-Ta)/\theta_{JA}$.

2. Dropout voltage is defined as the input voltage minus the output voltage that produces a 2% change in the output voltage from the value at appointed V_{IN}.

3. Ripple rejection ratio measurement circuit. $RR=20\times\log(\Delta V_{IN}/\Delta V_{OUT})$.
$C_{\_IN}$=10μF, $C_{\_OUT}$=10μF+0.1μF



4. Application information for LDO load capacitor selection for stability:

Recommended Output Capacitor

$V_{IN}=V_{OUT}+1V$, Ta=-40°C~85°C, unless otherwise specified

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
C_{LOAD}	Output Load Capacitor	—	Capacitor	4.7	10	—	μF

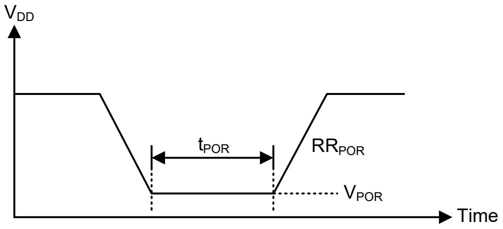
In common with most regulators, the LDO requires external capacitors between V_{OUT} and ground for regulator stability. Capacitor values of 4.7μF or large are acceptable, provided the smaller ESR is less than 10Ω. Aluminum electrolytic capacitor is suitable, provided they meet the requirements described above.

For better load transient response purposes, use a combination of a C_{LOAD} 10μF and extra 0.1μF capacitor on V_{OUT}. Note that the 0.1μF capacitor is always required on V_{OUT} and strong recommended be a multi-layer ceramic capacitor. The internal regulator is designed to be stable with an output filter capacitor C_{LOAD} and ESR as recommended.

Power-on Reset Characteristics

Ta=-40°C~85°C

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



H-Bridge Driver Electrical Characteristics

$V_{DD}=5V$, $HBV_{CC}=15V$ and Ta=25°C, unless otherwise specified

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Unit
Power Supply						
V_{DD}	Supply Voltage	—	2.5	—	5.5	V
I_{DD}	Supply Operation Current	PWM=25kHz, OUT1 and OUT2 open	—	0.45	1.00	mA
$I_{DD(STB)}$	Supply Standby Current	IN1=IN2="0", Standby mode	—	550	800	μA
$I_{DD(SLP)}$	Supply Sleep Current	IN1=IN2="0" or "1", Sleep Period	—	—	0.1	μA
HBV_{CC}	Motor Power Supply	—	2.5	—	15.0	V
I_M	HBV_{CC} Operation Current	PWM=25kHz, no load	—	0.85	1.00	mA
$I_{M(STB)}$	HBV_{CC} Standby Current	IN1=IN2="0", Standby mode	—	950	1100	μA

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Unit
H-Bridge Driver						
R_{ON}	HS+LS FET On-resistance (1)	V_{DD}=3.3V, I_{OUT}=800mA	—	0.3	0.4	Ω
V_{CLAMP}	Clamp Diode Voltage	I=500mA (HS and LS)	—	0.8	—	V
$I_{HS(OFF)}$	HS MOSFET Leakage Current	IN1=IN2="0", HBV_{CC}=15V, V_{OUT}=0V, measure I (HBV_{CC})	—	—	0.1	μA
$t_{r(OUT)}$	Output Rise Time	R_L=20Ω, 10% to 90% (Figure1)	—	100	—	ns
$t_{f(OUT)}$	Output Fall Time	R_L=20Ω, 90% to 10% (Figure1)	—	100	—	ns
Control Logic						
V_{IL}	Input Logic Low Voltage	V_{DD}=5.0V	—	—	0.8	V
		V_{DD}=2.5V	—	—	0.4	V
V_{IH}	Input Logic High Voltage	V_{DD}=5.0V	2	—	—	V
		V_{DD}=2.5V	1	—	—	V
V_{HYS}	Input Logic Hysteresis	—	—	0.1	—	V
t_{P1}	IN-to-OUT Propagation Delay (Figure1)	R_L=20Ω, INx to OUTx (high-Z to high/low)	—	100	—	ns
t_{P2}		R_L=20Ω, INx to OUTx (high/low to high-Z)	—	100	—	ns
t_{P3}		R_L=20Ω, INx to OUTx	—	100	—	ns
t_{P4}		R_L=20Ω, INx to OUTx	—	100	—	ns
t_{SLPEN}	Sleep Period Entry Time	IN1=IN2="0" or "1" until charge pump switches off (Figure 2)	—	10	—	ms
f_{PWM}	Input PWM Frequency	Internal charge pump activates	—	—	200	kHz
R_{PD}	Input Pull Down Resistance	IN1 and IN2	—	135	—	kΩ
Charge Pump						
t_{CP_ON}	Charge Pump on Time	Charge pump activates time (Figure 2)	—	10	—	ms
Protection						
V_{UVLO+}	V_{DD} Turn on Level	V_{DD} rises	—	—	2.5	V
V_{UVLO-}	V_{DD} Turn off Level	V_{DD} falls	1.8	—	—	V
I_{OCP}	Over Current Threshold	With deglitch time, t_{DEG}	2.5	3.0	3.5	A
t_{DEG}	Over Current Deglitch Time	(Figure 3)	—	2	—	μs
t_{RETRY}	Over Current Retry Time (3)	(Figure 3 & 4 & 5)	—	1	—	ms
I_{OSP}	Short Circuit Threshold (2)	Without deglitch time (Figure 4)	—	4.5	—	A
T_{SHD}	Thermal Shutdown Threshold	—	—	155	—	°C
T_{REC}	Thermal Recovery Temperature	—	—	120	—	°C

Note: 1. HS means High Side and LS means Low Side.

2. The H-bridge driver provides full short circuit protection for the OUTx-to-ground, OUTx-to-power or OUT1-to-OUT2 path.

3. The retry mechanism is only active in Forward and Reverse mode.

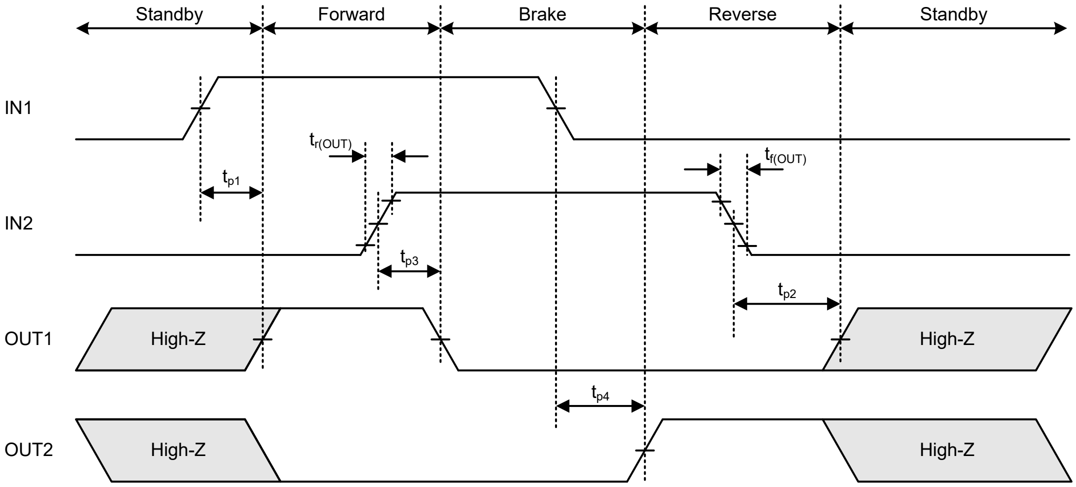


Figure 1. H-Bridge Driver Operation Mode Control Logic in Active Period

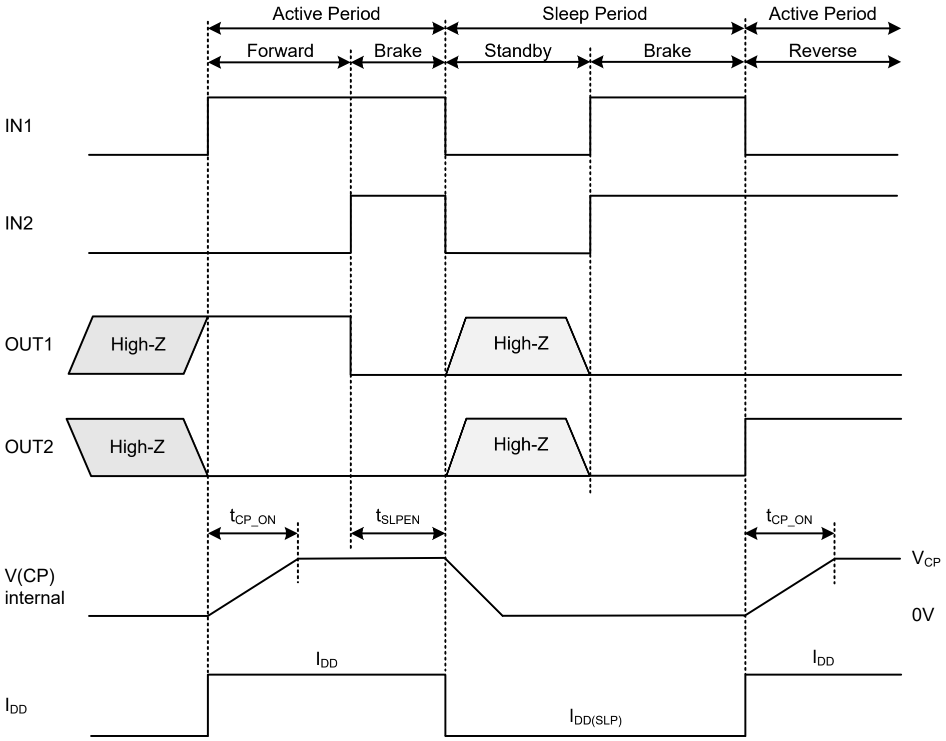


Figure 2. H-Bridge Driver Operation Mode Control Timing Diagram

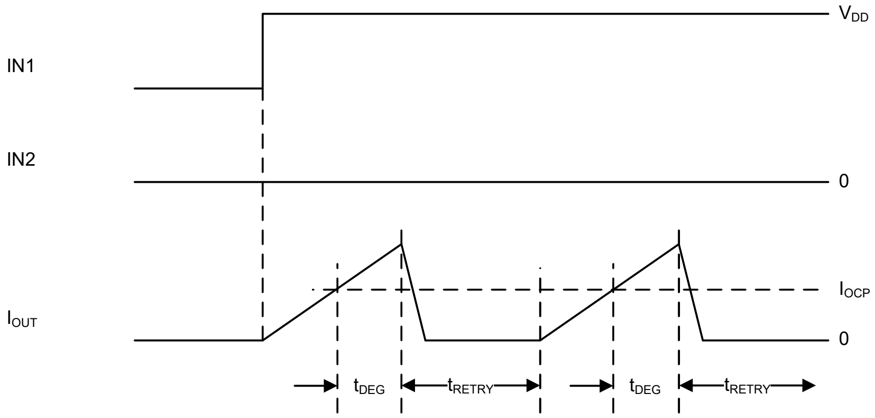


Figure 3. H-Bridge Driver OCP Reaction

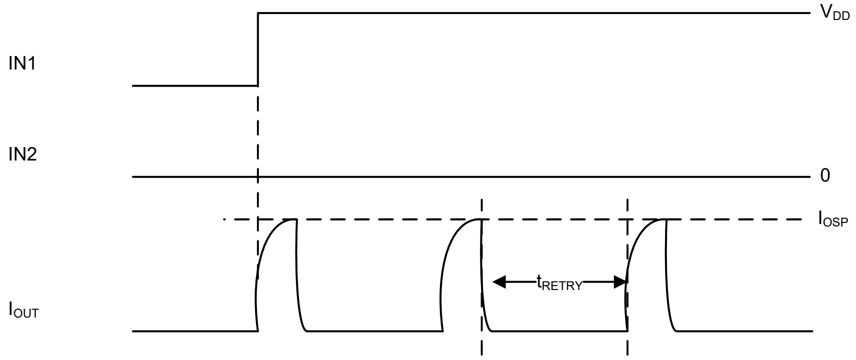


Figure 4. H-Bridge Driver OSP Reaction

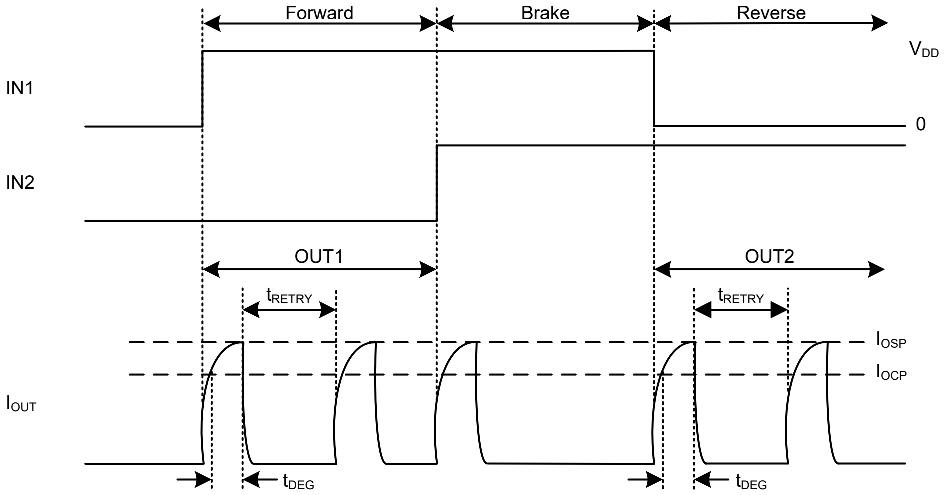


Figure 5. H-Bridge Driver Retry Reaction

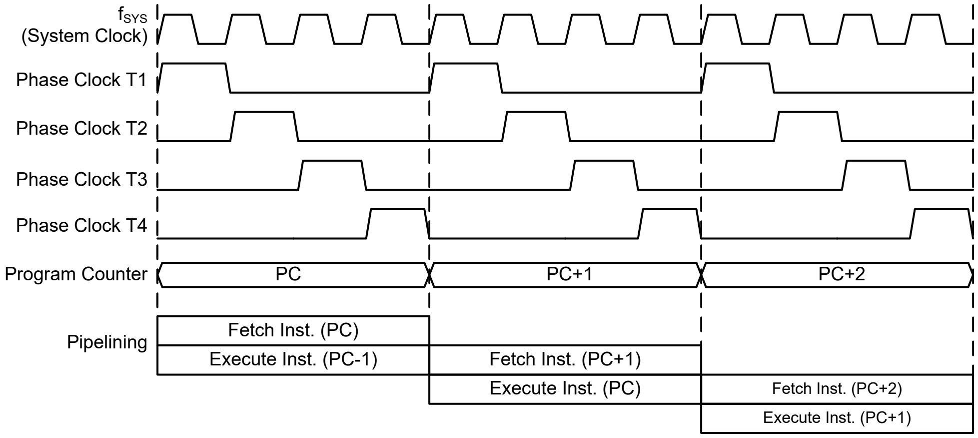
System Architecture

A key factor in the high-performance features of the Holtek range of microcontrollers is attributed to their internal system architecture. The device takes 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 which need one more cycle. An 8-bit wide ALU is used in practically all instruction set operations, which 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 and A/D control system with maximum reliability and flexibility. This makes the device suitable for low-cost, high-volume production for controller applications.

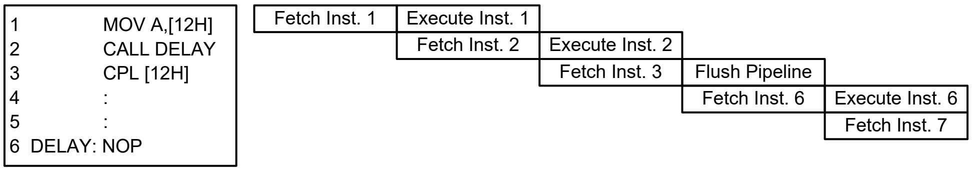
Clocking and Pipelining

The main system clock, derived from either an HIRC or LIRC 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.

For instructions involving branches, such as jump or call instructions, two machine 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.



System Clocking and Pipelining



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. Only the lower 8 bits, known as the Program Counter Low Register, are directly addressable by the application program.

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.

Program Counter	
High Byte	**Low Byte (PCL)**
PC10~PC8	PCL7~PCL0

Program Counter

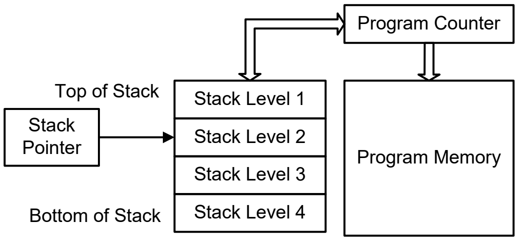
The lower byte of the Program Counter, known as the Program Counter Low register or PCL, is available for program control and is a readable and writeable register. By transferring data directly into this register, a short program jump can be executed directly; however, as only this low byte is available 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. Manipulating the PCL register may cause program branching, so an extra cycle is needed to pre-fetch.

Stack

This is a special part of the memory which is used to save the contents of the Program Counter only. The stack, organized into 4 levels, is neither part of the data nor part of the program space, and is neither readable nor writeable. The activated level is indexed by the Stack Pointer, 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.

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.

If the stack is overflow, the first Program Counter save in the stack will be lost.



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:

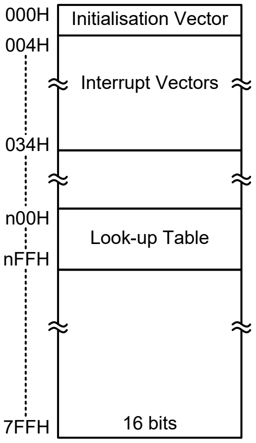
- Arithmetic operations:
  ADD, ADDM, ADC, ADCM, SUB, SUBM, SBC, SBCM, DAA
- Logic operations:
  AND, OR, XOR, ANDM, ORM, XORM, CPL, CPLA
- Rotation:
  RRA, RR, RRCA, RRC, RLA, RL, RLCA, RLC
- Increment and Decrement:
  INCA, INC, DECA, DEC
- Branch decision:
  JMP, SZ, SZA, SNZ, SIZ, SDZ, SIZA, SDZA,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

Flash Program Memory

The Program Memory is the location where the user code or program is stored. For the device the Program Memory is Flash type, which means it can be programmed and re-programmed a large number of times, allowing the user the convenience of code modification on the same device. By using the appropriate programming tools, the Flash device offer users the flexibility to conveniently debug and develop their applications while also offering a means of field programming and updating.

Structure

The Program Memory has a capacity of 2K×16 bits. 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 a separate table pointer register.



Program Memory Structure

Special Vectors

Within the Program Memory, certain locations are reserved for the reset and interrupts. The location 000H 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.

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 address of the look up data to be retrieved in the table pointer registers, TBLP and TBHP. These registers define the total address of the look-up table.

After setting up the table pointer, the table data can be retrieved from the Program Memory using the “TABRD [m]” or “TABRDL[m]” instruction. When the instruction is 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.

The accompanying diagram illustrates the addressing data flow of the look-up table.

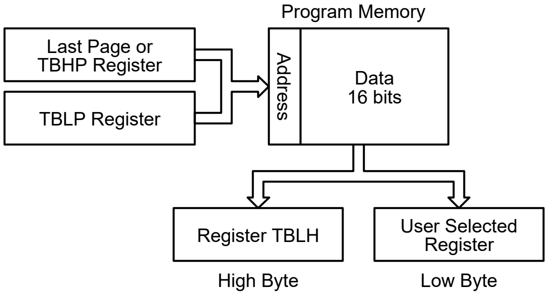


Table Program Example

The following example shows how the table pointer and table data is defined and retrieved from the microcontroller. This example uses raw table data located in the Program Memory which is stored there using the ORG statement. The value at this ORG statement is “0700H” which refers to the start address of the last page within the 2K Program Memory of the device. The table pointer low byte register 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 "0706H" or 6 locations after the start of the last page. Note that the value for the table pointer is referenced to the specific address pointed by the TBHP and TBLP registers if the "TABRD [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 "TABRD [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 table read instructions. If using the table read instructions, the Interrupt Service Routines may change the value of the 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 low table pointer - note that this address
                   ; is referenced
mov tblp, a        ; to the last page or the page that tbhp pointed
mov a, 07h         ; initialise high table pointer
mov tbhp, a
:
:
tabrd tempreg1     ; transfers value in table referenced by table pointer data at program
                   ; memory address "0706H" transferred to tempreg1 and TBLH
dec tblp           ; reduce value of table pointer by one
tabrd tempreg2     ; transfers value in table referenced by table pointer
                   ; data at program memory address "0705H" transferred to
                   ; tempreg2 and TBLH in this example the data "1AH" is
                   ; transferred to tempreg1 and data "0FH" to register tempreg2
:
:
org 0700h          ; sets initial address of program memory
dc 00Ah, 00Bh, 00Ch, 00Dh, 00Eh, 00Fh, 01Ah, 01Bh
:
:
```

In Circuit Programming – ICP

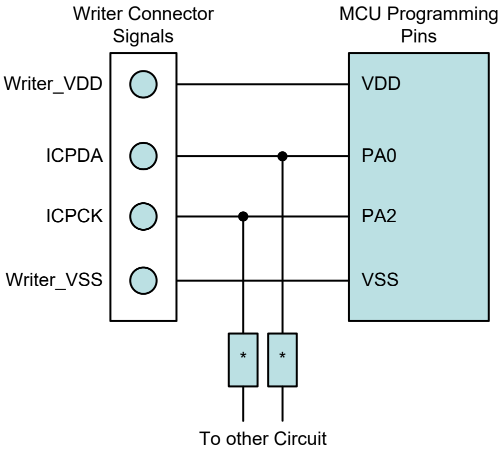
The provision of Flash type Program Memory provides the user with a means of convenient and easy upgrades and modifications to their programs on the same device.

As an additional convenience, Holtek has provided a means of programming the microcontroller in-circuit using a 4-pin interface. This provides manufacturers with the possibility of manufacturing their circuit boards complete with a programmed or un-programmed microcontroller, and then programming or upgrading the program at a later stage. This enables product manufacturers to easily keep their manufactured products supplied with the latest program releases without removal and re-insertion of the device.

Holtek Writer Pins	MCU Programming Pins	Pin Description
ICPDA	PA0	Programming Serial Data/Address
ICPCK	PA2	Programming Clock
VDD	VDD	Power Supply
VSS	VSS	Ground

The Program Memory can be programmed serially in-circuit using this 4-wire interface. Data is downloaded and uploaded serially on a single pin with an additional line for the clock. Two additional lines are required for the power supply. The technical details regarding the in-circuit programming of the device is beyond the scope of this document and will be supplied in supplementary literature.

During the programming process, the user must take care of the ICPDA and ICPCK pins for data and clock programming purposes to ensure that no other outputs are connected to these two pins.



Note: * may be resistor or capacitor. The resistance of * must be greater than 1kΩ or the capacitance of * must be less than 1nF.

On-Chip Debug Support – OCDS

There is an EV chip named BP45V1632 which is used to emulate the BP45F1632 device. The EV chip device also provides an "On-Chip Debug" function to debug the real MCU device during the development process. The EV chip and the real MCU device are almost functionally compatible except for "On-Chip Debug" function. Users can use the EV chip device to emulate the real chip device behavior by connecting the OCDSDA and OCDSCK pins to the Holtek HT-IDE development tools. The OCDSDA pin is the OCDS Data/Address input/output pin while the OCDSCK pin is the OCDS clock input pin. When users use the EV chip device for debugging, other pin functions which are shared with the OCDSDA and OCDSCK pins in the device will have no effect in the EV chip. However, the two OCDS pins which are pin-shared with the ICP programming pins are still used as the Flash Memory programming pins for ICP. For more detailed OCDS information, refer to the corresponding document named "Holtek e-Link for 8-bit MCU OCDS User's Guide".

Holtek e-Link Pins	EV Chip OCDS Pins	Pin Description
OCDSDA	OCDSDA	On-Chip Debug Support Data/Address input/output
OCDSCK	OCDSCK	On-Chip Debug Support Clock input
VDD	VDD	Power Supply
VSS	VSS	Ground

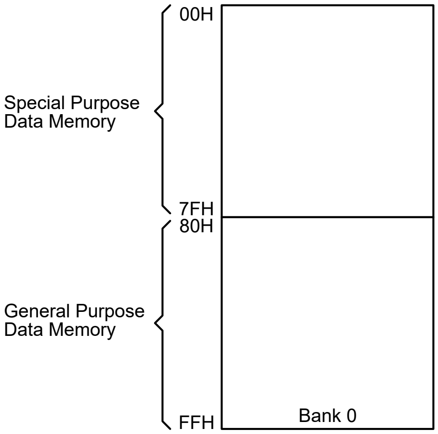
Data Memory

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

Structure

Categorized into two types, 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.

The start address of the Data Memory for the device is 00H. The address range of the Special Purpose Data Memory for the device is from 00H to 7FH while the address range of the General Purpose Data Memory is from 80H to FFH.



Data Memory Structure

General Purpose Data Memory

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 programing for both reading and writing operations. By using the bit operation 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.

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”.

	Bank 0		Bank 0
00H	IAR0	40H	OCPDA
01H	MP0	41H	OCPOCAL
02H	IAR1	42H	OCPCCAL
03H	MP1	43H	▨
04H	▨	44H	▨
05H	ACC	45H	▨
06H	PCL	46H	LDOC
07H	TBLP	47H	HVO0HC
08H	TBLH	48H	HVO0LC
09H	TBHP	49H	HVO1HC
0AH	STATUS	4AH	HVO1LC
0BH	▨	4BH	▨
0CH	▨	4CH	▨
0DH	▨	4DH	▨
0EH	▨	4EH	▨
0FH	RSTFC	4FH	▨
10H	PB	50H	INTC0
11H	PBC	51H	INTC1
12H	PBPU	52H	INTC2
13H	WDTC	53H	INTC3
14H	PA	54H	PAS0
15H	PAC	55H	PAS1
16H	PAPU	56H	▨
17H	PAWU	57H	▨
18H	LVRC	58H	▨
19H	LVDC	59H	INTEG
1AH	SLEDC	5AH	PBS0
1BH	▨	5BH	▨
1CH	ORMC	5CH	▨
1DH	TB0C	5DH	▨
1EH	TB1C	5EH	PTMC0
1FH	CTMC0	5FH	PTMC1
20H	CTMC1	60H	PTMDL
21H	CTMDL	61H	PTMDH
22H	CTMDH	62H	PTMAL
23H	CTMAL	63H	PTMAH
24H	CTMAH	64H	PTMRPL
25H	▨	65H	PTMRPH
26H	▨	66H	PSC0R
27H	▨	67H	PSC1R
28H	▨	68H	▨
29H	CPR	69H	▨
2AH	OCVPC	6AH	▨
2BH	▨	6BH	▨
2CH	▨	6CH	▨
2DH	▨	6DH	▨
2EH	▨	6EH	▨
2FH	SCC	6FH	▨
30H	HIRCC	70H	▨
31H	▨	71H	▨
32H	▨	72H	▨
33H	▨	73H	▨
34H	▨	74H	▨
35H	▨	75H	▨
36H	SADOL	76H	▨
37H	SADOH	77H	▨
38H	SADC0	78H	▨
39H	SADC1	79H	▨
3AH	▨	7AH	▨
3BH	OVPC0	7BH	▨
3CH	OVPC1	7CH	▨
3DH	OVPDA	7DH	▨
3EH	OCPC0	7EH	▨
3FH	OCPC1	7FH	▨

▨ : Unused, read as 00H

Special Purpose Data Memory Structure

Special Function Register Description

Most of the Special Function Register details will be described in the relevant functional section; however several registers require a separate description in this section.

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 Pointers, MP0 or MP1. Acting as a pair, IAR0 and MP0 can together access data only from Bank 0 while the IAR1 register together with the MP1 register can access data from any Data Memory Bank. As the Indirect Addressing Registers are not physically implemented, reading the Indirect Addressing Registers will return a result of “00H” and writing to the registers 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 address and track data. 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. MP0, together with Indirect Addressing Register, IAR0, are used to access data from Bank 0, while MP1 and IAR1 are used to access data from all banks. Direct Addressing can be used in Bank 0, all other banks must be addressed indirectly using MP1 and IAR1.

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 examples 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.

Look-up Table Registers – TBLP, TBHP, TBLH

These three special function registers are used to control operation of the look-up table which is stored in the Program Memory. TBLP and TBHP are the table pointers and indicate the location where the table data is located. Their value must be setup before any table read commands are executed. Their value can be changed, for example using the “INC” or “DEC” instructions, allowing for easy table data pointing and reading. TBLH is the location where the high order byte of the table data is stored after a table read data instruction has been executed. Note that the lower order table data byte is transferred to a user defined location.

Option Memory Mapping Register – ORMC

The ORMC register is used to enable the Option Memory Mapping function. The Option Memory capacity is 32 words. When a specific pattern of 55H and AAH is consecutively written into this register, the Option Memory Mapping function will be enabled and then the Option Memory code can be read by using the table read instruction. The Option Memory addresses 00H~1FH will be mapped to Program Memory last page addresses E0H~FFH.

To successfully enable the Option Memory Mapping function, the specific pattern of 55H and AAH must be written into the ORMC register in two consecutive instruction cycles. It is therefore recommended that the global interrupt bit EMI should first be cleared before writing the specific pattern, and then set high again at a proper time according to users’ requirements after the pattern is successfully written. An internal timer will be activated when the pattern is successfully written. The mapping operation will be automatically finished after a period of $4\times t_{LIRC}$. Therefore, users should read the data in time, otherwise the Option Memory Mapping function needs to be restarted. After the completion of each consecutive write operation to the ORMC register, the timer will recount.

When the table read instructions are used to read the Option Memory code, both “TABRD [m]” and “TABRDL [m]” instructions can be used. However, care must be taken if the “TABRD [m]” instruction is used, the table pointer defined by the TBHP register must be referenced to the last page. Refer to corresponding sections about the table read instruction for more details.

• ORMC Register

Bit	7	6	5	4	3	2	1	0
Name	ORMC7	ORMC6	ORMC5	ORMC4	ORMC3	ORMC2	ORMC1	ORMC0
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 7~0 **ORMC7~ORMC0**: Option Memory Mapping specific pattern
When a specific pattern of 55H and AAH is written into this register, the Option Memory Mapping function will be enabled. Note that the register content will be cleared after the MCU is woken up from the IDLE/SLEEP mode.

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.

- C is set if an operation results in a carry during an addition operation or if a borrow does not take place during a subtraction operation; otherwise C is cleared. C is also affected by a rotate through carry instruction.
- AC is set if an operation results in a carry out of the low nibbles in addition, or no borrow from the high nibble into the low nibble in subtraction; otherwise AC is cleared.
- Z is set if the result of an arithmetic or logical operation is zero; otherwise Z is cleared.
- OV is set if an operation results in a carry into the highest-order bit but not a carry out of the highest-order bit, or vice versa; otherwise OV is cleared.
- PDF is cleared by a system power-up or executing the “CLR WDT” instruction. PDF is set by executing the “HALT” instruction.
- TO is cleared by a system power-up or executing the “CLR WDT” or “HALT” instruction. TO is set by a WDT time-out.

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 subroutine can corrupt the status register, precautions must be taken to correctly save it.

• 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

The “C” flag is also affected by a rotate through carry instruction.

Oscillators

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 operations are selected through the relevant control registers.

Oscillator Overview

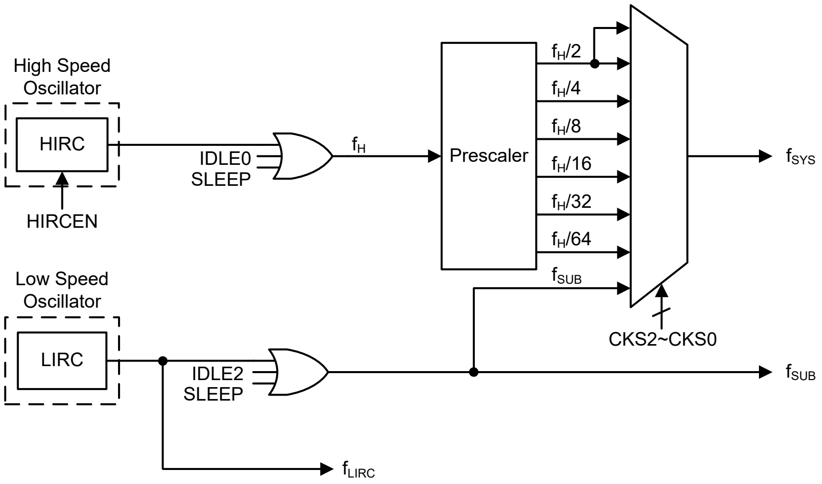
In addition to being the source of the main system clock the oscillators also provide clock sources for the Watchdog Timer and Time Base Interrupt. The fully integrated internal oscillators, requiring no external components, are provided to form a wide range of both fast and slow system oscillators. The higher frequency oscillator provides higher performance but carry with it the disadvantage of higher power requirements, while the opposite is of course true for the lower frequency oscillator. With the capability of dynamically switching between fast and slow system clock, the device has the flexibility to optimize the performance/power ratio, a feature especially important in power sensitive portable applications.

Type	Name	Frequency
Internal High Speed RC	HIRC	30MHz
Internal Low Speed RC	LIRC	32kHz

Oscillator Types

System Clock Configurations

There are two oscillator sources, one high speed oscillator and one low speed oscillator. The high speed system clock is sourced from the internal 30MHz RC oscillator, HIRC. The low speed oscillator is the internal 32kHz RC oscillator, LIRC. Selecting whether the low or high speed oscillator is used as the system oscillator is implemented using the CKS2~CKS0 bits in the SCC register and the system clock can be dynamically selected.



System Clock Configurations

Internal High Speed RC Oscillator – HIRC

The internal RC oscillator is a fully integrated system oscillator requiring no external components. The internal high speed RC oscillator has a fixed frequency of 30MHz. 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.

Internal 32kHz Oscillator – LIRC

The Internal 32kHz System Oscillator is a fully integrated low frequency RC oscillator with a typical frequency of 32kHz, requiring no external components for its implementation.

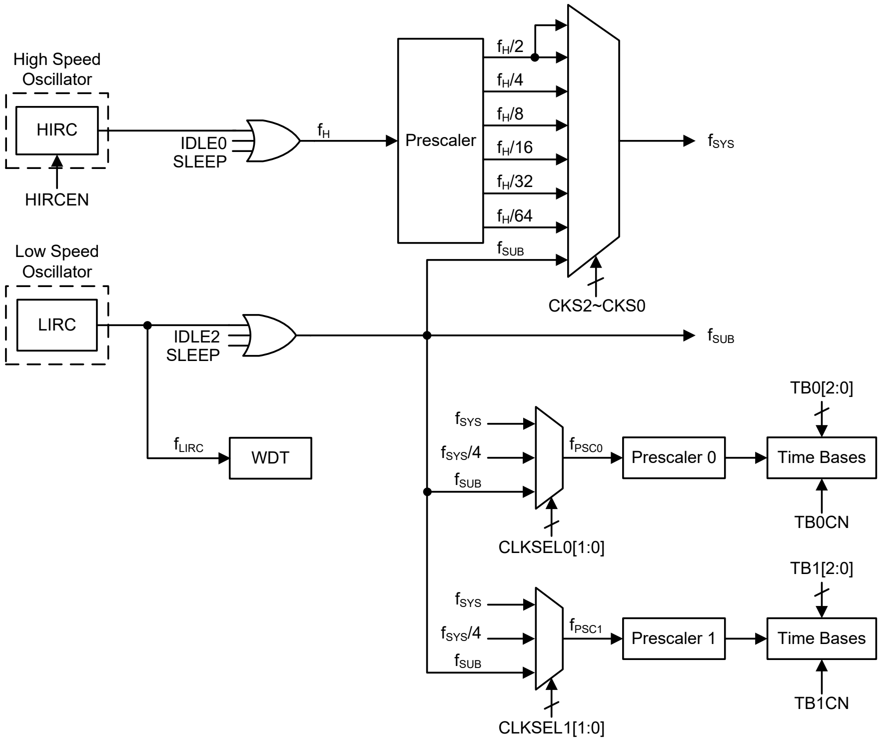
Operating Modes and System Clocks

Present day applications require that their microcontrollers have high performance but often still demand that they consume as little power as possible, conflicting requirements that are especially true in battery powered portable applications. The fast clocks required for high performance will by their nature increase current consumption and of course vice versa, lower speed clocks reduce current consumption. As Holtek has provided the device with both high and low speed clock sources and the means to switch between them dynamically, the user can optimise the operation of their microcontroller to achieve the best performance/power ratio.

System Clocks

The device has many different clock sources for both the CPU and peripheral function operation. By providing the user with a wide range of clock options using register programming, a clock system can be configured to obtain maximum application performance.

The main system clock can come from either a divided version of the high speed system oscillator with a range of $f_H/2$~$f_H/64$ or a low frequency, f_{SUB}, source, and is selected using the CKS2~CKS0 bits in the SCC register. The high speed system clock is sourced from the HIRC oscillator. The low speed system clock source is sourced from the LIRC oscillator.



Device Clock Configurations

Note: When the system clock source f_{SYS} is switched to f_{SUB} from f_H, the high speed oscillator will stop to conserve the power or continue to oscillate to provide the clock source, $f_H/2$~$f_H/64$, for peripheral circuit to use, which is determined by configuring the corresponding high speed oscillator enable control bit.

System Operation Modes

There are six different modes of operation for the microcontrollers, each one with its own special characteristics and which can be chosen according to the specific performance and power requirements of the application. There are two modes allowing normal operation of the microcontroller, the FAST Mode and SLOW Mode. The remaining four modes, the SLEEP, IDLE0, IDLE1 and IDLE2 Mode are used when the microcontroller CPU is switched off to conserve power.

Operation Mode	CPU	Register Setting			f_{SYS}	f_H	f_{SUB}	f_{LIRC}
		FHIDEN	FSIDEN	CKS2~CKS0				
FAST	On	x	x	000~110	$f_H/2$~$f_H/64$	On	On	On
SLOW	On	x	x	111	f_{SUB}	On/Off $^{(1)}$	On	On
IDLE0	Off	0	1	000~110	Off	Off	On	On
				111	On			
IDLE1	Off	1	1	xxx	On	On	On	On
IDLE2	Off	1	0	000~110	On	On	Off	On
				111	Off			
SLEEP	Off	0	0	xxx	Off	Off	Off	On/Off $^{(2)}$

"x": Don't care

Note: 1. The f_H clock will be switched on or off by configuring the corresponding oscillator enable bit in the SLOW mode.

2. The f_{LIRC} clock can be on or off which is controlled by the WDT function being enabled or disabled in the SLEEP mode.

FAST Mode

This is one of the main operating modes where the microcontrollers have all of their functions operational and where the system clock is provided by the high speed oscillator. This mode operates allowing the microcontrollers to operate normally with a clock source which will come from the high speed oscillator, HIRC. The high speed oscillator will however first be divided by a ratio ranging from 2 to 64, the actual ratio being selected by the CKS2~CKS0 bits in the SCC register. Although a high speed oscillator is used, running the microcontrollers at a divided clock ratio reduces the operating current.

SLOW Mode

This is also a mode where the microcontroller operates normally although now with a slower speed clock source. The clock source used will be from f_{SUB}, which is derived from the LIRC oscillator.

SLEEP Mode

The SLEEP Mode is entered when a HALT instruction is executed and when the FHIDEN and FSIDEN bit are low. In the SLEEP mode the CPU will be stopped. The f_{SUB} clock provided to the peripheral function will also be stopped, too. However, the f_{LIRC} clock can continue to operate if the WDT function is enabled.

IDLE0 Mode

The IDLE0 Mode is entered when a HALT instruction is executed and when the FHIDEN bit in the SCC register is low and the FSIDEN bit in the SCC register is high. In the IDLE0 Mode the CPU will be switched off but the low speed oscillator will be turned on to drive some peripheral functions.

IDLE1 Mode

The IDLE1 Mode is entered when a HALT instruction is executed and when the FHIDEN bit in the SCC register is high and the FSIDEN bit in the SCC register is high. In the IDLE1 Mode the CPU will be switched off but both the high and low speed oscillators will be turned on to provide a clock source to keep some peripheral functions operational.

IDLE2 Mode

The IDLE2 Mode is entered when a HALT instruction is executed and when the FHIDEN bit in the SCC register is high and the FSIDEN bit in the SCC register is low. In the IDLE2 Mode the CPU will be switched off but the high speed oscillator will be turned on to provide a clock source to keep some peripheral functions operational.

Control Registers

The SCC and HIRCC registers are used to control the system clock and the corresponding oscillator configurations.

Register Name	Bit							
	7	6	5	4	3	2	1	0
SCC	CKS2	CKS1	CKS0	—	—	—	FHIDEN	FSIDEN
HIRCC	—	—	—	—	—	—	HIRCF	HIRCEN

System Operating Mode Control Register List

- **SCC Register**

Bit	7	6	5	4	3	2	1	0
Name	CKS2	CKS1	CKS0	—	—	—	FHIDEN	FSIDEN
R/W	R/W	R/W	R/W	—	—	—	R/W	R/W
POR	0	1	0	—	—	—	0	0

Bit 7~5 **CKS2~CKS0**: System clock selection
000: $f_H/2$
001: $f_H/2$
010: $f_H/4$
011: $f_H/8$
100: $f_H/16$
101: $f_H/32$
110: $f_H/64$
111: f_{SUB}

These three bits are used to select which clock is used as the system clock source. In addition to the system clock source directly derived from f_{SUB}, a divided version of the high speed system oscillator can also be chosen as the system clock source.

Bit 4~2 Unimplemented, read as “0”

Bit 1 **FHIDEN**: High frequency oscillator control when CPU is switched off
0: Disable
1: Enable

This bit is used to control whether the high speed oscillator is activated or stopped when the CPU is switched off by executing a “HALT” instruction.

Bit 0 **FSIDEN**: Low frequency oscillator control when CPU is switched off
0: Disable
1: Enable

This bit is used to control whether the low speed oscillator is activated or stopped when the CPU is switched off by executing a “HALT” instruction.

- **HIRCC Register**

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

Bit 7~2 Unimplemented, read as “0”

Bit 1 **HIRCF**: HIRC oscillator stable flag
0: HIRC unstable
1: HIRC stable

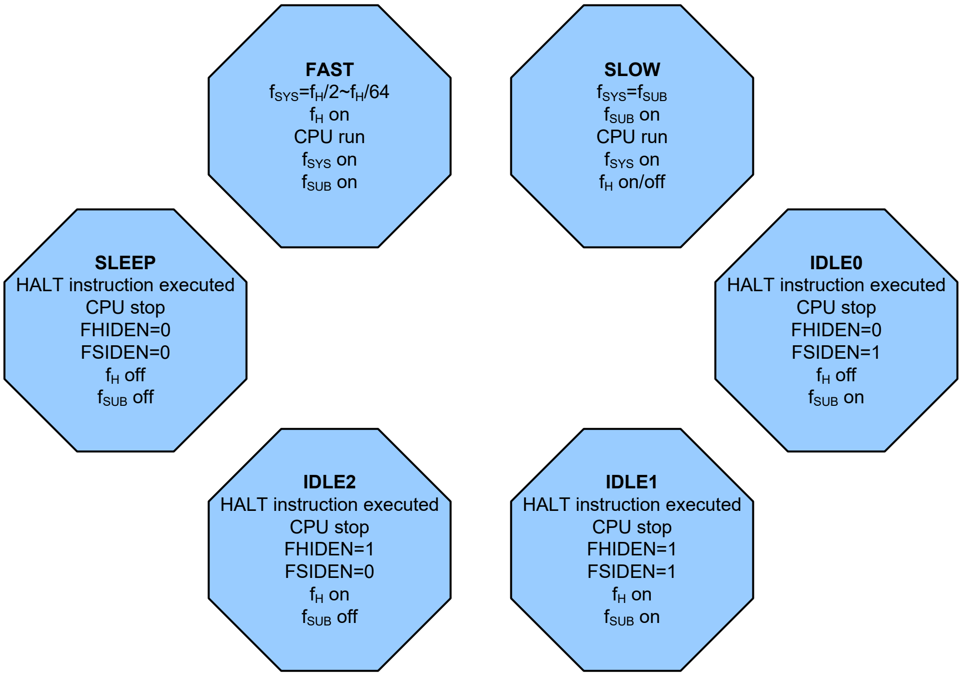
This bit is used to indicate whether the HIRC oscillator is stable or not. When the HIRCEN bit is set to 1 to enable the HIRC oscillator, the HIRCF bit will first be cleared to 0 and then set to 1 after the HIRC oscillator is stable.

Bit 0 **HIRCEN**: HIRC oscillator enable control
0: Disable
1: Enable

Operating Mode Switching

The device can switch between operating modes dynamically allowing the user to select the best performance/power ratio for the present task in hand. In this way microcontroller operations that do not require high performance can be executed using slower clocks thus requiring less operating current and prolonging battery life in portable applications.

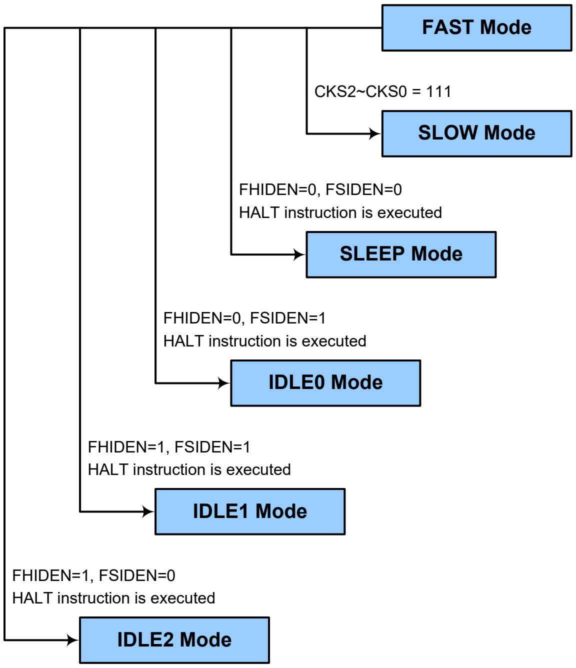
In simple terms, mode switching between the FAST Mode and SLOW Mode is executed using the CKS2~CKS0 bits in the SCC register while mode switching from the FAST/SLOW Modes to the SLEEP/IDLE Modes is executed via the HALT instruction. When a HALT instruction is executed, whether the device enters the IDLE Mode or the SLEEP Mode is determined by the condition of the FHIDEN and FSIDEN bits in the SCC register.



FAST Mode to SLOW Mode Switching

When running in the FAST Mode, which uses the high speed system oscillator, and therefore consumes more power, the system clock can switch to run in the SLOW Mode by setting the CKS2~CKS0 bits to “111” in the SCC register. This will then use the low speed system oscillator which will consume less power. Users may decide to do this for certain operations which do not require high performance and can subsequently reduce power consumption.

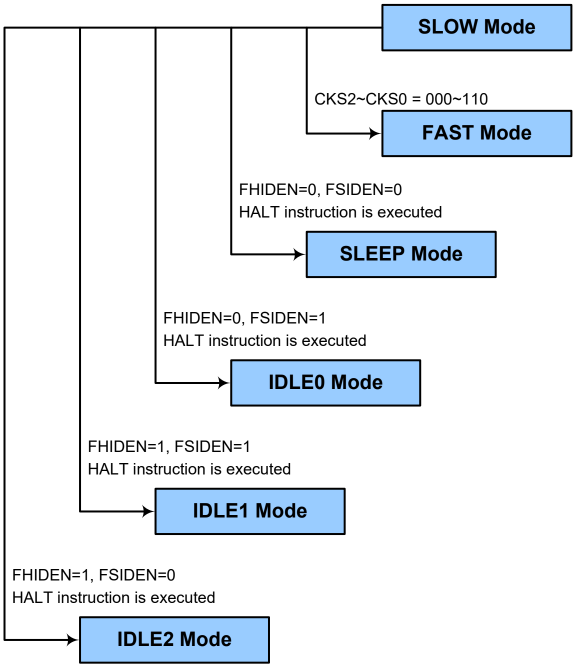
The SLOW Mode system clock is sourced from the LIRC oscillator and therefore requires this oscillator to be stable before full mode switching occurs.



SLOW Mode to FAST Mode Switching

In the SLOW mode the system clock is derived from f_{SUB}. When system clock is switched back to the FAST mode from f_{SUB}, the CKS2~CKS0 bits should be set to "000"~"110" and then the system clock will respectively be switched to $f_H/2$~$f_H/64$.

However, if f_H is not used in the SLOW mode and thus switched off, it will take some time to re-oscillate and stabilise when switching to the FAST mode from the SLOW Mode. This is monitored using the HIRCF bit in the HIRCC register. The time duration required for the high speed system oscillator stabilisation is specified in the System Start Up Time Characteristics.



Entering the SLEEP Mode

There is only one way for the device to enter the SLEEP Mode and that is to execute the "HALT" instruction in the application program with both the FHIDEN and FSIDEN bits in the SCC register equal to "0". In this mode all the clocks and functions will be switched off except the WDT function. When this instruction is executed under the conditions described above, the following will occur:

- The system clock will be stopped and the application program will stop at the "HALT" instruction.
- The Data Memory contents and registers will maintain their present condition.
- The I/O ports will maintain their present conditions.
- In the status register, the Power Down flag, PDF, will be set and the Watchdog time-out flag, TO, will be cleared.
- The WDT will be cleared and resume counting if the WDT function is enabled. If the WDT function is disabled, the WDT will be cleared and then stopped.

Entering the IDLE0 Mode

There is only one way for the device to enter the IDLE0 Mode and that is to execute the "HALT" instruction in the application program with the FHIDEN bit in the SCC register equal to "0" and the FSIDEN bit in the SCC register equal to "1". When this instruction is executed under the conditions described above, the following will occur:

- The f_H clock will be stopped and the application program will stop at the "HALT" instruction, but the f_{SUB} clock will be on.
- The Data Memory contents and registers will maintain their present condition.
- The I/O ports will maintain their present conditions.
- In the status register, the Power Down flag, PDF, will be set and the Watchdog time-out flag, TO, will be cleared.
- The WDT will be cleared and resume counting if the WDT function is enabled. If the WDT function is disabled, the WDT will be cleared and then stopped.

Entering the IDLE1 Mode

There is only one way for the device to enter the IDLE1 Mode and that is to execute the "HALT" instruction in the application program with both the FHIDEN and FSIDEN bits in the SCC register equal to "1". When this instruction is executed under the conditions described above, the following will occur:

- The f_H and f_{SUB} clocks will be on but the application program will stop at the "HALT" instruction.
- The Data Memory contents and registers will maintain their present condition.
- The I/O ports will maintain their present conditions.
- In the status register, the Power Down flag, PDF, will be set and the Watchdog time-out flag, TO, will be cleared.
- The WDT will be cleared and resume counting if the WDT function is enabled. If the WDT function is disabled, the WDT will be cleared and then stopped.

Entering the IDLE2 Mode

There is only one way for the device to enter the IDLE2 Mode and that is to execute the "HALT" instruction in the application program with the FHIDEN bit in the SCC register equal to "1" and the FSIDEN bit in the SCC register equal to "0". When this instruction is executed under the conditions described above, the following will occur:

- The f_H clock will be on but the f_{SUB} clock will be off and the application program will stop at the "HALT" instruction.
- The Data Memory contents and registers will maintain their present condition.
- The I/O ports will maintain their present conditions.
- In the status register, the Power Down flag, PDF, will be set and the Watchdog time-out flag, TO, will be cleared.
- The WDT will be cleared and resume counting if the WDT function is enabled. If the WDT function is disabled, the WDT will be cleared and then stopped.

Standby Current Considerations

As the main reason for entering the SLEEP or IDLE Mode is to keep the current consumption of the device to as low a value as possible, perhaps only in the order of several micro-amps except in the IDLE1 and IDLE2 Mode, 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. This also applies to the device which has different package types, as there may be unbonded pins. These pins must either be set as outputs or if set as inputs must have pull-high resistors connected.

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. Also note that additional standby current will also be required if the LIRC oscillator has enabled.

In the IDLE1 and IDLE2 Mode the high speed oscillator is on, if the peripheral function clock source is derived from the high speed oscillator, the additional standby current will also be perhaps in the order of several hundred micro-amps.

Wake-up

To minimise power consumption the device can enter the SLEEP or any IDLE Mode, where the CPU will be switched off. However, when the device is woken up again, it will take a considerable time for the original system oscillator to restart, stabilise and allow normal operation to resume.

After the system enters the SLEEP or IDLE Mode, it can be woken up from one of various sources listed as follows:

- An external falling edge on Port A
- A system interrupt
- A WDT overflow

When the device executes the "HALT" instruction, it will enter the IDLE or SLEEP mode and the PDF flag will be set to 1. The PDF flag will be cleared to 0 if the device experiences a system power-up or executes the clear Watchdog Timer instruction. If the system is woken up by a WDT overflow, a Watchdog Timer reset will be initiated and the TO flag will be set to 1. 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, other flags remain in their original status.

Each pin on Port A can be set using the PAWU register to permit a negative transition on the pin to wake-up the system. When a pin 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 high before entering the SLEEP or IDLE Mode, the wake-up function of the related interrupt will be disabled.

Watchdog Timer

The Watchdog Timer is provided to prevent program malfunctions or sequences from jumping to unknown locations, due to certain uncontrollable external events such as electrical noise.

Watchdog Timer Clock Source

The Watchdog Timer clock source is provided by the internal clock, f_{LIRC} which is sourced from the LIRC oscillator. The LIRC internal oscillator has an approximate frequency of 32kHz and this specified internal clock period can vary with V_{DD}, temperature and process variations. The Watchdog Timer source clock is then subdivided by a ratio of 2^8 to 2^{18} to give longer timeouts, the actual value being chosen using the WS2~WS0 bits in the WDTC register.

Watchdog Timer Control Register

A single register, WDTC, controls the required time-out period, the WDT enable/disable operation as well as the MCU reset operation.

- **WDTC Register**

Bit	7	6	5	4	3	2	1	0
Name	WE4	WE3	WE2	WE1	WE0	WS2	WS1	WS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	1	0	0	1	1

Bit 7~3 **WE4~WE0**: WDT function enable control
01010: Enable
10101: Disable
Other values: MCU reset

When these bits are changed to any other values due to environmental noise the microcontroller will be reset; this reset operation will be activated after a delay time, t_{SRESET}, and the WRF bit in the RSTFC register will be set high.

Bit 2~0 **WS2~WS0**: WDT time-out period selection
000: $2^8/f_{LIRC}$
001: $2^{10}/f_{LIRC}$
010: $2^{12}/f_{LIRC}$
011: $2^{14}/f_{LIRC}$
100: $2^{15}/f_{LIRC}$
101: $2^{16}/f_{LIRC}$
110: $2^{17}/f_{LIRC}$
111: $2^{18}/f_{LIRC}$

These three bits determine the division ratio of the Watchdog Timer source clock, which in turn determines the timeout period.

- **RSTFC Register**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	LVRF	LRF	WRF
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	x	0	0

"x": Unknown

Bit 7~3	Unimplemented, read as "0"
Bit 2	**LVRF**: LVR function reset flag Refer to the Low Voltage Reset section.
Bit 1	**LRF**: LVR control register software reset flag Refer to the Low Voltage Reset section.
Bit 0	**WRF**: WDT control register software reset flag 0: Not occurred 1: Occurred This bit is set to 1 by the WDT control register software reset and cleared by the application program. Note that this bit can only be cleared to 0 by the application program.

Watchdog Timer Operation

The Watchdog Timer operates by providing a device reset when its timer overflows. This means that in the application program and during normal operation the user has to strategically clear the Watchdog Timer before it overflows to prevent the Watchdog Timer from executing a reset. This is done using the clear watchdog instruction. If the program malfunctions for whatever reason, jumps to an unknown location, or enters an endless loop, the clear instruction will not be executed in the correct manner, in which case the Watchdog Timer will overflow and reset the device. With regard to the Watchdog Timer enable/disable function, there are five bits, WE4~WE0, in the WDTC register to offer the enable/disable control and reset control of the Watchdog Timer. The WDT function will be enabled when the WE4~WE0 bits are set to a value of 01010B while the WDT function will be disabled if the WE4~WE0 bits are equal to 10101B. If the WE4~WE0 bits are set to any other values rather than 01010B and 10101B, it will reset the device after a delay time, t_{SRESET}. After power on these bits will have a value of 01010B.

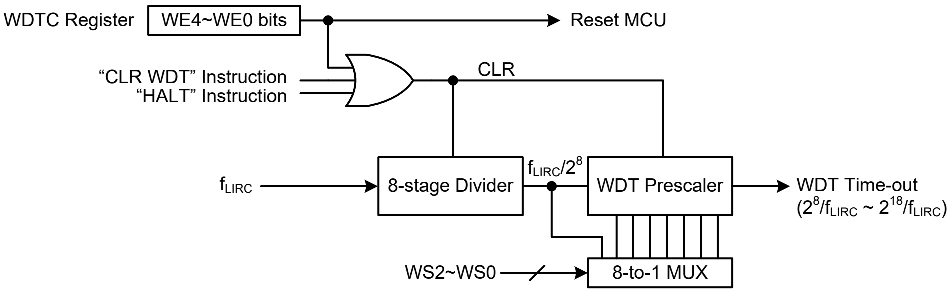
WE4~WE0 Bits	WDT Function
10101B	Disable
01010B	Enable
Any other value	Reset MCU

Watchdog Timer Function Control

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 SLEEP or IDLE Mode, when a Watchdog Timer time-out occurs, 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 a WDTC software reset, which means a certain value except 01010B and 10101B written into the WE4~WE0 bits, the second is using the Watchdog Timer software clear instruction, the third is via a HALT instruction.

There is only one method of using software instruction to clear the Watchdog Timer. That is to use the single "CLR WDT" instruction to clear the WDT.

The maximum time-out period is when the 2^{18} division ratio is selected. As an example, with a 32kHz LIRC oscillator as its source clock, this will give a maximum watchdog period of around 8 seconds for the 2^{18} division ratio, and a minimum timeout of 8ms for the 2^{8} division ration.



Watchdog Timer

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 microcontrollers. In this case, internal circuitry will ensure that the microcontrollers, 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.

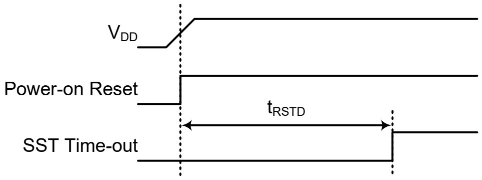
Another reset exists in the form of a Low Voltage Reset, LVR, where a full reset is implemented in situations where the power supply voltage falls below a certain threshold. Another type of reset is when the Watchdog Timer overflows and resets the microcontroller. All types of reset operations result in different register conditions being setup.

Reset Functions

There are several ways in which a microcontroller reset can occur through events occurring internally.

Power-on Reset

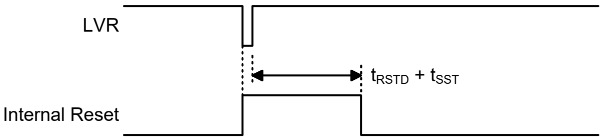
The most fundamental and unavoidable reset is the one that occurs after power is first applied to the microcontrollers. As well as ensuring that the Program Memory begins execution from the first memory address, a power-on reset also ensures that certain other registers are preset to known conditions. 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.



Power-on 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 and provides an MCU reset should the value fall below a certain predefined level. The LVR function is always enabled in the FAST/SLOW mode with a specific LVR voltage V_{LVR}. 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 in battery powered applications, the LVR will automatically reset the device internally and the LVRF bit in the RSTFC register will also be set high. 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 LVR/LVD Electrical 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 by the LVS7~LVS0 bits in the LVRC register. If the LVS7~LVS0 bits are changed to some different values by environmental noise, the LVR will reset the device after a delay time, t_{SRESET}. When this happens, the LRF bit in the RSTFC register will be set high. After power on the register will have the value of 01010101B. Note that the LVR function will be automatically disabled when the device enters the IDLE or SLEEP mode.



Low Voltage Reset Timing Chart

- **LVRC Register**

Bit	7	6	5	4	3	2	1	0
Name	LVS7	LVS6	LVS5	LVS4	LVS3	LVS2	LVS1	LVS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	1	0	1	0	1

Bit 7~0 **LVS7~LVS0**: LVR voltage select control
01010101: 2.1V
00110011: 2.55V
10011001: 3.15V
10101010: 3.8V
Any other value: MCU reset – register is reset to POR value

When an actual low voltage condition occurs, as specified above, an MCU reset will be generated. The reset operation will be activated after the low voltage condition keeps more than a t_{LVR} time. In this situation the register contents will remain the same after such a reset occurs.

Any register value, other than the four defined register values above, will also result in the generation of an MCU reset. The reset operation will be activated after a delay time, t_{SRESET}. However in this situation the register contents will be reset to the POR value.

- **RSTFC Register**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	LVRF	LRF	WRF
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	x	0	0

"x": Unknown

Bit 7~3 Unimplemented, read as "0"

Bit 2 **LVRF**: LVR function reset flag
0: Not occurred
1: Occurred

This bit is set high when a specific low voltage reset situation condition occurs. This bit can only be cleared to zero by the application program.

Bit 1 **LRF**: LVR control register software reset flag
0: Not occurred
1: Occurred

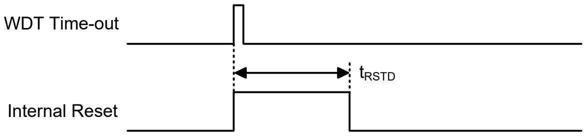
This bit is set high if the LVRC register contains any non-defined LVRC register values. This in effect acts like a software-reset function. This bit can only be cleared to zero by the application program.

Bit 0 **WRF**: WDT control register software reset flag

Refer to the Watchdog Timer Control Register section.

Watchdog Time-out Reset during Normal Operation

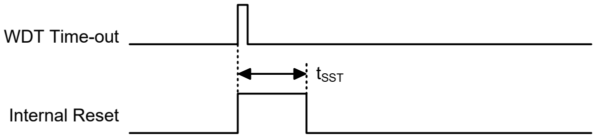
When the Watchdog time-out Reset during normal operations in the FAST or SLOW mode occurs, the Watchdog time-out flag TO will be set to “1”.



WDT Time-out Reset during Normal Operation Timing Chart

Watchdog Time-out Reset during SLEEP or IDLE Mode

The Watchdog time-out Reset during SLEEP or IDLE 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 and PDF flags will be set to “1”. Refer to the System Start Up Time Characteristics for t_{SST} details.



WDT Time-out Reset during SLEEP or IDLE Timing Chart

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 SLEEP or IDLE Mode function or Watchdog Timer. The reset flags are shown in the table:

TO	PDF	Reset Conditions
0	0	Power-on reset
u	u	LVR reset during FAST or SLOW Mode operation
1	u	WDT time-out reset during FAST or SLOW Mode operation
1	1	WDT time-out reset during IDLE or SLEEP Mode operation

“u”: 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, Time Bases	Cleared after reset, WDT begins counting
Timer Modules	All Timer Modules will be turned off
Input/Output Ports	I/O ports will be set 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 execution 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	Power On Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE/SLEEP)
IAR0	xxxx xxxx	uuuu uuuu	uuuu uuuu
MP0	xxxx xxxx	uuuu uuuu	uuuu uuuu
IAR1	xxxx xxxx	uuuu uuuu	uuuu uuuu
MP1	xxxx xxxx	uuuu uuuu	uuuu uuuu
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu
TBHP	---- -xxx	---- -uuu	---- -uuu
STATUS	--00 xxxx	--1u uuuu	--11 uuuu
RSTFC	---- -x00	---- -uuu	---- -uuu
PB	---- 1111	---- 1111	---- uuuu
PBC	---- 1111	---- 1111	---- uuuu
PBPU	---- 0000	---- 0000	---- uuuu
WDTC	0101 0011	0101 0011	uuuu uuuu
PA	1111 1111	1111 1111	uuuu uuuu
PAC	1111 1111	1111 1111	uuuu uuuu
PAPU	0000 0000	0000 0000	uuuu uuuu
PAWU	0000 0000	0000 0000	uuuu uuuu
LVRC	0101 0101	0101 0101	uuuu uuuu
LVDC	--00 0000	--00 0000	--uu uuuu
SLEDC	---- 0000	---- 0000	---- uuuu
ORMC	0000 0000	0000 0000	0000 0000
TB0C	0--- -000	0--- -000	u--- -uuu
TB1C	0--- -000	0--- -000	u--- -uuu
CTMC0	0000 0000	0000 0000	uuuu uuuu
CTMC1	0000 0000	0000 0000	uuuu uuuu
CTMDL	0000 0000	0000 0000	uuuu uuuu
CTMDH	---- --00	---- --00	---- --uu
CTMAL	0000 0000	0000 0000	uuuu uuuu
CTMAH	---- --00	---- --00	---- --uu
CPR	0000 0000	0000 0000	uuuu uuuu
OCVPC	---- 1000	---- 1000	---- uuuu
SCC	010- --00	010- --00	uuu- --uu
HIRCC	---- --01	---- --01	---- --uu
SADOL	xxxx ----	xxxx ----	uuuu ---- (ADRFS=0) uuuu uuuu (ADRFS=1)

Register	Power On Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE/SLEEP)
SADOH	xxxx xxxx	xxxx xxxx	uuuu uuuu (ADRFS=0)
			---- uuuu (ADRFS=1)
SADC0	0000 0000	0000 0000	uuuu uuuu
SADC1	0000 0000	0000 0000	uuuu uuuu
OVPC0	0000 0000	0000 0000	uuuu uuuu
OVPC1	0001 0000	0001 0000	uuuu uuuu
OVPDA	0000 0000	0000 0000	uuuu uuuu
OCPC0	0000 0--0	0000 0--0	uuuu u--u
OCPC1	--00 0000	--00 0000	--uu uuuu
OCPDA	0000 0000	0000 0000	uuuu uuuu
OCPOCAL	0010 0000	0010 0000	uuuu uuuu
OCPCCAL	0001 0000	0001 0000	uuuu uuuu
LDOC	0--- --00	0--- --00	u--- --uu
HVO0HC	0000 0000	0000 0000	uuuu uuuu
HVO0LC	0000 0000	0000 0000	uuuu uuuu
HVO1HC	0000 0000	0000 0000	uuuu uuuu
HVO1LC	0000 0000	0000 0000	uuuu uuuu
INTC0	-000 0000	-000 0000	-uuu uuuu
INTC1	0000 0000	0000 0000	uuuu uuuu
INTC2	0000 0000	0000 0000	uuuu uuuu
INTC3	--00 --00	--00 --00	--uu --uu
PAS0	00-- 00--	00-- 00--	uu-- uu--
PAS1	0000 0000	0000 0000	uuuu uuuu
INTEG	---- 0000	---- 0000	---- uuuu
PBS0	0000 0000	0000 0000	uuuu uuuu
PTMC0	0000 0---	0000 0---	uuuu u---
PTMC1	0000 0000	0000 0000	uuuu uuuu
PTMDL	0000 0000	0000 0000	uuuu uuuu
PTMDH	---- --00	---- --00	---- --uu
PTMAL	0000 0000	0000 0000	uuuu uuuu
PTMAH	---- --00	---- --00	---- --uu
PTMRPL	0000 0000	0000 0000	uuuu uuuu
PTMRPH	---- --00	---- --00	---- --uu
PSC0R	---- --00	---- --00	---- --uu
PSC1R	---- --00	---- --00	---- --uu

Note: “u” stands for unchanged
“x” stands for unknown
“-” stands for unimplemented

Input/Output Ports

Holtek microcontrollers offer considerable flexibility on their I/O ports. With the input or output designation of every pin fully under user program control, pull-high selections for all ports and wake-up selections on certain pins, the user is provided with an I/O structure to meet the needs of a wide range of application possibilities.

The device provides bidirectional input/output lines labeled with port names PA. These I/O ports are mapped to the RAM Data Memory with specific addresses as shown in the Special Purpose Data Memory table. All of these I/O ports can be used for input and output operations. 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.

Register Name	Bit							
	7	6	5	4	3	2	1	0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PB	—	—	—	—	PB3	PB2	PB1	PB0
PBC	—	—	—	—	PBC3	PBC2	PBC1	PBC0
PBPU	—	—	—	—	PBPU3	PBPU2	PBPU1	PBPU0

"—": Unimplemented, read as "0"

Note: For the unbonded lines, PB0 and PB1, as well as the interconnection lines, PB2 and PB3, their input/output, pull-high and other functions are also controlled by the corresponding bits in these registers. They should be properly configured to ensure correct operations. Refer to the relevant register description sections for more details.

I/O Logic Function Register List

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, all I/O pins, when configured as a digital input have the capability of being connected to an internal pull-high resistor. These pull-high resistors are selected using the PxPU registers and are implemented using weak PMOS transistors.

Note that the pull-high resistor can be controlled by the relevant pull-high control register only when the pin-shared functional pin is selected as a digital input or NMOS output. Otherwise, the pull-high resistors cannot be enabled.

- **PxPU Register**

Bit	7	6	5	4	3	2	1	0
Name	PxPU7	PxPU6	PxPU5	PxPU4	PxPU3	PxPU2	PxPU1	PxPU0
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

PxPUn: I/O port x pin pull-high function control
0: Disable
1: Enable

The PxPUn bit is used to control the pin pull-high function. Here the "x" is the Port name which can be A or B. However, the actual available bits for each I/O Port may be different.

Special attention has be paid to the unbonded lines, PB0 and PB1, which should either be set as outputs or if set as inputs must have pull-high resistors connected to avoid unwanted power consumption result from floating input conditions.

For the PB2 and PB3 lines, they are internally connected to the H-bridge driver inputs, IN1 and IN2, respectively which individually have an internal pull-low resistor, if their pull-high functions are enabled, additional power consumption will be required.

Port A Wake-up

The HALT instruction forces the microcontroller into the SLEEP or IDLE Mode which preserves power, 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 Port A pins from high to low. This function is especially suitable for applications that can be woken up via external switches. Each pin on Port A can be selected individually to have this wake-up feature using the PAWU register.

Note that the wake-up function can be controlled by the wake-up control registers only when the pin is selected as a general purpose input and the MCU enters the IDLE or SLEEP mode.

• PAWU Register

Bit	7	6	5	4	3	2	1	0
Name	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
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 7~0 **PAWU7~PAWU0**: PA7~PA0 wake-up function control
0: Disable
1: Enable

I/O Port Control Registers

The I/O port has its own control register known as PAC, to control the input/output configuration. With this control register, each CMOS output or input can be reconfigured dynamically under software control. Each pin of the I/O ports is directly mapped to a bit in its associated port control register. 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.

• PxC Register

Bit	7	6	5	4	3	2	1	0
Name	PxC7	PxC6	PxC5	PxC4	PxC3	PxC2	PxC1	PxC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

PxCn: I/O port x pin type selection
0: Output
1: Input

The PxCn bit is used to control the pin type selection. Here the "x" is the Port name which can be A or B. However, the actual available bits for each I/O Port may be different.

Special attention has be paid to the unbonded lines, PB0 and PB1, which should either be set as outputs or if set as inputs must have pull-high resistors connected to avoid unwanted power consumption result from floating input conditions.

For the PB2 and PB3 lines, they are internally connected to the H-bridge driver inputs, IN1 and IN2, respectively which should be configured as outputs after power on by clearing the corresponding PxCn bit, in order to properly control the H-bridge driver.

PA Port Source Current Selection

The device supports different output source current driving capability for PA port. With the selection register, SLEDC, specific I/O port can support four levels of the source current driving capability. These source current selection bits are available when the corresponding pin is configured as a CMOS output. Otherwise, these select bits have no effect. Users should refer to the Input/Output Characteristics section to select the desired output source current for different applications.

- **SLEDC Register**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	SLEDC3	SLEDC2	SLEDC1	SLEDC0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as “0”

Bit 3~2 **SLEDC3~SLEDC2**: PA7~PA4 source current selection
00: Source current = Level 0 (Min.)
01: Source current = Level 1
10: Source current = Level 2
11: Source current = Level 3 (Max.)

Bit 1~0 **SLEDC1~SLEDC0**: PA3~PA0 source current selection
00: Source current = Level 0 (Min.)
01: Source current = Level 1
10: Source current = Level 2
11: Source current = Level 3 (Max.)

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 these pins, the desired function of the multi-function I/O pins is selected by a series of registers via the application program control.

Pin-shared Function Selection Registers

The limited number of supplied pins in a package can impose restrictions on the amount of functions a certain device can contain. However by allowing the same pins to share several different functions and providing a means of function selection, a wide range of different functions can be incorporated into even relatively small package sizes. The device includes Port “x” output function Selection register “n”, labeled as PxSn, which can select the desired functions of the multi-function pin-shared pins.

The most important point to note is to make sure that the desired pin-shared function is properly selected and also deselected. For most pin-shared functions, to select the desired pin-shared function, the pin-shared function should first be correctly selected using the corresponding pin-shared control register. After that the corresponding peripheral functional setting should be configured and then the peripheral function can be enabled. However, a special point must be noted for digital input pins, such as INTn, CTCK, etc. which share the same pin-shared control configuration with their corresponding general purpose I/O functions when setting the relevant pin-shared control bits. To select these pin functions, in addition to the necessary pin-shared control and peripheral functional setup aforementioned, they must also be set as an input by setting the corresponding bit in the I/O port control register. To correctly deselect the pin-shared function, the peripheral function should first be disabled and then the corresponding pin-shared function control register can be modified to select other pin-shared functions.

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAS0	PAS07	PAS06	—	—	PAS03	PAS02	—	—
PAS1	PAS17	PAS16	PAS15	PAS14	PAS13	PAS12	PAS11	PAS10
PBS0	PBS07	PBS06	PBS05	PBS04	PBS03	PBS02	PBS01	PBS00

Pin-shared Function Selection Register List

- **PAS0 Register**

Bit	7	6	5	4	3	2	1	0
Name	PAS07	PAS06	—	—	PAS03	PAS02	—	—
R/W	R/W	R/W	—	—	R/W	R/W	—	—
POR	0	0	—	—	0	0	—	—

Bit 7~6 **PAS07~PAS06**: PA3 pin-shared function selection
00: PA3/INT1
01: AN1
10: PA3/INT1
11: PA3/INT1

Bit 5~4 Unimplemented, read as "0"

Bit 3~2 **PAS03~PAS02**: PA1 pin-shared function selection
00: PA1
01: OCPI
10: OVPI
11: PA1

Note: 1. If the PA1 pin is setup as the OVPI pin function, then the OVPI will be connected together with the internal A/D converter input signal V_{ADVDT1}, at which piont special consideration should be given to the interaction between internal and external circuits.

2. If the PA1 and PA6 pins are setup as the OCPI pin function simultaneously, then both are avilable, however this will result in short circuit, so this setting must be avoided in applications.

Bit 1~0 Unimplemented, read as "0"

- **PAS1 Register**

Bit	7	6	5	4	3	2	1	0
Name	PAS17	PAS16	PAS15	PAS14	PAS13	PAS12	PAS11	PAS10
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 7~6 **PAS17~PAS16**: PA7 pin-shared function selection
00: PA7/INT0
01: AN0
10: PA7/INT0
11: PA7/INT0

Bit 5~4 **PAS15~PAS14**: PA6 pin-shared function selection
00: PA6
01: CTPB
10: OCPI
11: AN4

Note: If the PA1 and PA6 pins are setup as the OCPI pin function simultaneously, then both are avilable, however this will result in short circuit, so this setting must be avoided in applications.

Bit 3~2 **PAS13~PAS12**: PA5 pin-shared function selection
00: PA5
01: CTP
10: OVPO
11: AN3

Bit 1~0 **PAS11~PAS10**: PA4 pin-shared function selection
00: PA4/CTCK
01: VREF for ADC or OCP/OVP DAC input reference voltage
10: AN2
11: PA4/CTCK

• PBS0 Register

Bit	7	6	5	4	3	2	1	0
Name	PBS07	PBS06	PBS05	PBS04	PBS03	PBS02	PBS01	PBS00
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 7~6 **PBS07~PBS06**: PB3 pin-shared function selection
00: PB3
01: PTP
10: PB3
11: Reserved

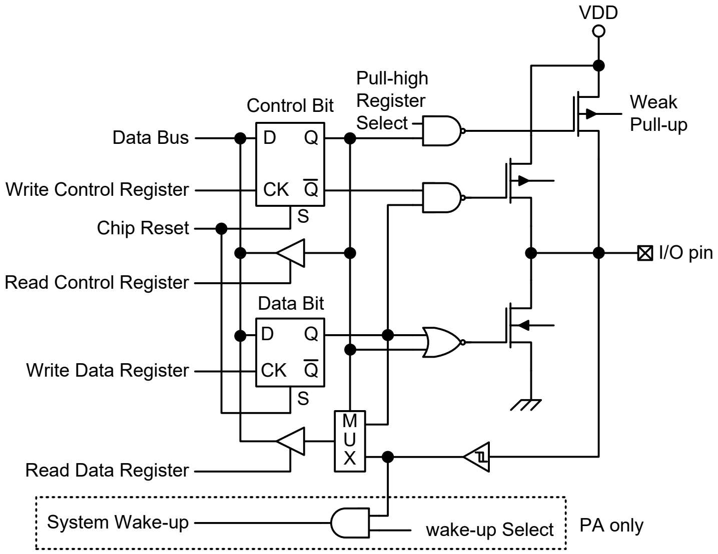
Bit 5~4 **PBS05~PBS04**: PB2 pin-shared function selection
00: PB2
01: PTPB
10: Reserved
11: PB2

Bit 3~2 **PBS03~PBS02**: These bits should be kept unchanged after power on

Bit 1~0 **PBS01~PBS00**: These bits should be kept unchanged after power on

I/O Pin Structure

The accompanying diagram illustrates the internal structure of the I/O logic function. As the exact logical construction of the I/O pin will differ from this drawing, it is supplied as a guide only to assist with the functional understanding of the I/O logic function. The wide range of pin-shared structures does not permit all types to be shown.



Logic Function Input/Output Structure

Programming Considerations

Within the user program, one of the first things to consider is port initialisation. After a reset, all of the I/O data and port control registers 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 selections have been chosen. If the port control registers are then programmed to set some pins as outputs, these output pins will have an initial high output value unless the associated port data registers are first programmed. Selecting which pins are inputs and which are outputs can be achieved byte-wide by loading the correct values into the appropriate 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.

Port A has the additional capability of providing wake-up function. When the device is in the SLEEP or IDLE Mode, various methods are available to wake the device up. One of these is a high to low transition of any of the Port A pins. Single or multiple pins on Port A can be set to have this function.

Timer Modules – TM

One of the most fundamental functions in any microcontroller devices is the ability to control and measure time. To implement time related functions the device includes several Timer Modules, generally abbreviated to the name TM. The TMs are multi-purpose timing units and serve to provide operations such as Timer/Counter, Compare Match Output and Single Pulse Output as well as being the functional unit for the generation of PWM signals. Each of the TMs has two interrupts. The addition of input and output pins for each TM ensures that users are provided with timing units with a wide and flexible range of features.

The common features of the different TM types are described here with more detailed information provided in the individual Compact and Periodic Type TM sections.

Introduction

The device contains two Timer Modules and each individual TM can be categorised as a certain type, namely the Compact Type TM or Periodic Type TM. Although similar in nature, the different TM types vary in their feature complexity. The common features to all of the Compact and Periodic type TMs will be described in this section and the detailed operation regarding each of the TM types will be described in separate sections. The main features and differences between the two types of TMs are summarised in the accompanying table.

Function	CTM	PTM
Timer/Counter	√	√
Compare Match Output	√	√
PWM Output	√	√
Single Pulse Output	—	√
PWM Alignment	Edge	Edge
PWM Adjustment Period & Duty	Duty or Period	Duty or Period

TM Function Summary

TM Operation

The different types of TM offer a diverse range of functions, from simple timing operations to PWM signal generation. The key to understanding how the TM operates is to see it in terms of a free running count-up counter whose value is then compared with the value of pre-programmed internal comparators. When the free running count-up counter has the same value as the pre-programmed comparator, known as a compare match situation, a TM interrupt signal will be generated which can clear the counter and perhaps also change the condition of the TM output pin. The internal TM counter is driven by a user selectable clock source, which can be an internal clock or an external pin.

TM Clock Source

The clock source which drives the main counter in each TM can originate from various sources. The selection of the required clock source is implemented using the xTCK2~xTCK0 bits in the xTM control registers, where "x" stands for C or P type TM. The clock source can be a ratio of the system clock, f_{SYS}, or the internal high clock, f_H, the f_{SUB} clock source or the external xTCK pin. The xTCK pin clock source is used to allow an external signal to drive the TM as an external clock source for event counting.

TM Interrupts

The Compact or Periodic type TM each has two internal interrupt, one for each of the internal comparator A or comparator P, which generate a TM interrupt when a compare match condition occurs. When a TM interrupt is generated, it can be used to clear the counter and also to change the state of the TM output pin.

TM External Pins

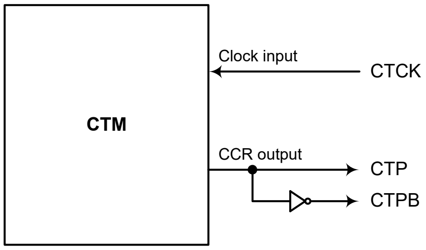
The Periodic type TM has no external pins. The Compact type TM has one input pin with the label CTCK. The CTM input pin, CTCK, is essentially a clock source for the CTM and is selected using the CTCK2~CTCK0 bits in the CTMC0 register. This external TM input pin allows an external clock source to drive the internal TM. The CTCK input pin can be chosen to have either a rising or falling active edge.

The Compact type TM has two output pins, CTP and CTPB, the CTPB is the inverted signal of the CTP output. When the TM is in the Compare Match Output Mode, these pins can be controlled by the TM to switch to a high or low level or to toggle when a compare match situation occurs. The external CTP and CTPB output pins are also the pins where the TM generates the PWM output waveform.

As the TM input and output pins are pin-shared with other functions, the TM input and output function must first be setup using relevant pin-shared function selection register. The details of the pin-shared function selection are described in the pin-shared function section.

CTM		PTM	
Input	Output	Input	Output
CTCK	CTP, CTPB	—	—

TM External Pins

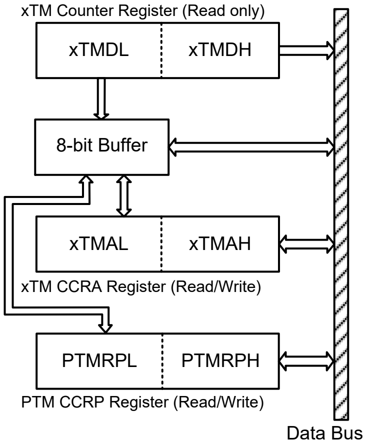


CTM Function Pin Block Diagram

Programming Considerations

The TM Counter Registers and the Compare CCRA and CCRP registers, all have a low and high byte structure. The high bytes can be directly accessed, but as the low bytes can only be accessed via an internal 8-bit buffer, reading or writing to these register pairs must be carried out in a specific way. The important point to note is that data transfer to and from the 8-bit buffer and its related low byte only takes place when a write or read operation to its corresponding high byte is executed.

As the CCRA and CCRP registers are implemented in the way shown in the following diagram and accessing these register pairs is carried out in a specific way as described above, it is recommended to use the "MOV" instruction to access the CCRA and CCRP low byte registers, named xTMAL and PTMRPL, using the following access procedures. Accessing the CCRA or CCRP low byte registers without following these access procedures will result in unpredictable values.

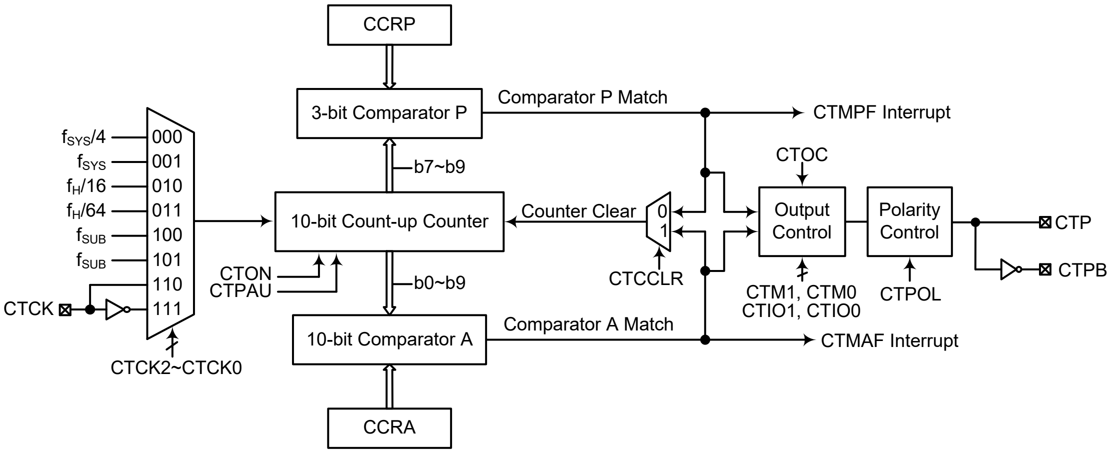


The following steps show the read and write procedures:

- Writing Data to CCRA or CCRP
  - ♦ Step 1. Write data to Low Byte xTMAL or PTMRPL
    – Note that here data is only written to the 8-bit buffer.
  - ♦ Step 2. Write data to High Byte xTMAH or PTMRPH
    – Here data is written directly to the high byte registers and simultaneously data is latched from the 8-bit buffer to the Low Byte registers.
- Reading Data from the Counter Registers, CCRA or CCRP
  - ♦ Step 1. Read data from the High Byte xTMDH, xTMAH or PTMRPH
    – Here data is read directly from the High Byte registers and simultaneously data is latched from the Low Byte register into the 8-bit buffer.
  - ♦ Step 2. Read data from the Low Byte xTMDL, xTMAL or PTMRPL
    – This step reads data from the 8-bit buffer.

Compact Type TM – CTM

The Compact Type TM contains three operating modes, which are Compare Match Output, Timer/Event Counter and PWM Output modes. The Compact Type TM can also be controlled with an external input pin and can drive two external output pins.



Note: The CTM external pins are pin-shared with other functions, so before using the CTM function, the pin-shared function registers must be set properly to enable the CTM pin function. The CTCK pin, if used, must also be set as an input by setting the corresponding bits in the port control register.

10-Bit Compact Type TM Block Diagram

Compact Type TM Operation

The size of Compact Type TM is 10-bit wide and its core is a 10-bit count-up counter which is driven by a user selectable internal or external clock source. There are also two internal comparators with the names, Comparator A and Comparator P. These comparators will compare the value in the counter with CCRP and CCRA registers. The CCRP comparator is 3-bit wide whose value is compared with the highest 3 bits in the counter while the CCRA is the 10 bits and therefore compares all counter bits.

The only way of changing the value of the 10-bit counter using the application program, is to clear the counter by changing the CTON bit from low to high. The counter will also be cleared automatically by a counter overflow or a compare match with one of its associated comparators. When these conditions occur, a CTM interrupt signal will also usually be generated. The Compact Type TM can operate in a number of different operational modes, can be driven by different clock sources including an input pin and can also control two output pins. All operating setup conditions are selected using relevant internal registers.

Compact Type TM Register Description

Overall operation of the Compact Type TM is controlled using a series of registers. A read only register pair exists to store the internal counter 10-bit value, while a read/write register pair exists to store the internal 10-bit CCRA value. The remaining two registers are control registers which setup the different operating and control modes as well as three CCRP bits.

Register Name	Bit							
	7	6	5	4	3	2	1	0
CTMC0	CTPAU	CTCK2	CTCK1	CTCK0	CTON	CTRP2	CTRP1	CTRP0
CTMC1	CTM1	CTM0	CTIO1	CTIO0	CTOC	CTPOL	CTDPX	CTCCLR
CTMDL	D7	D6	D5	D4	D3	D2	D1	D0
CTMDH	—	—	—	—	—	—	D9	D8
CTMAL	D7	D6	D5	D4	D3	D2	D1	D0
CTMAH	—	—	—	—	—	—	D9	D8

10-Bit Compact Type TM Register List

• CTMC0 Register

Bit	7	6	5	4	3	2	1	0
Name	CTPAU	CTCK2	CTCK1	CTCK0	CTON	CTRP2	CTRP1	CTRP0
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 **CTPAU**: CTM counter pause control
0: Run
1: Pause

The counter can be paused by setting this bit high. Clearing the bit to zero restores normal counter operation. When in a Pause condition the CTM will remain powered up and continue to consume power. The counter will retain its residual value when this bit changes from low to high and resume counting from this value when the bit changes to a low value again.

Bit 6~4 **CTCK2~CTCK0**: CTM Counter clock selection
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{SUB}
101: f_{SUB}
110: CTCK rising edge clock
111: CTCK falling edge clock

These three bits are used to select the clock source for the CTM. The external pin clock source can be chosen to be active on the rising or falling edge. The clock source f_{SYS} is the system clock, while f_H and f_{SUB} are other internal clocks, the details of which can be found in the Operating Modes and System Clocks section.

Bit 3 **CTON**: CTM counter on/off control
0: Off
1: On

This bit controls the overall on/off function of the CTM. Setting the bit high enables the counter to run while clearing the bit disables the CTM. Clearing this bit to zero will stop the counter from counting and turn off the CTM which will reduce its power consumption. When the bit changes state from low to high the internal counter value will be reset to zero, however when the bit changes from high to low, the internal counter will retain its residual value until the bit returns high again. If the CTM is in the Compare Match Output Mode or the PWM Output Mode then the CTM output pin will be reset to its initial condition, as specified by the CTOC bit, when the CTON bit changes from low to high.

Bit 2~0 **CTRP2~CTRP0**: CTM CCRP 3-bit register, compared with the CTM counter bit 9 ~ bit 7

Comparator P match period =
- 000: 1024 CTM clocks
- 001: 128 CTM clocks
- 010: 256 CTM clocks
- 011: 384 CTM clocks
- 100: 512 CTM clocks
- 101: 640 CTM clocks
- 110: 768 CTM clocks
- 111: 896 CTM clocks

These three bits are used to setup the value on the internal CCRP 3-bit register, which are then compared with the internal counter's highest three bits. The result of this comparison can be selected to clear the internal counter if the CTCCLR bit is set to zero. Setting the CTCCLR bit to zero ensures that a compare match with the CCRP values will reset the internal counter. As the CCRP bits are only compared with the highest three counter bits, the compare values exist in 128 clock cycle multiples. Clearing all three bits to zero is in effect allowing the counter to overflow at its maximum value.

• CTMC1 Register

Bit	7	6	5	4	3	2	1	0
Name	CTM1	CTM0	CTIO1	CTIO0	CTOC	CTPOL	CTDPX	CTCCLR
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 7~6 **CTM1~CTM0**: CTM operating mode selection
- 00: Compare Match Output Mode
- 01: Undefined
- 10: PWM Output Mode
- 11: Timer/Counter Mode

These bits setup the required operating mode for the CTM. To ensure reliable operation the CTM should be switched off before any changes are made to the CTM1 and CTM0 bits. In the Timer/Counter Mode, the CTM output pin state is undefined.

Bit 5~4 **CTIO1~CTIO0**: CTM external pin function selection

Compare Match Output Mode
- 00: No change
- 01: Output low
- 10: Output high
- 11: Toggle output

PWM Output Mode
- 00: PWM output inactive state
- 01: PWM output active state
- 10: PWM output
- 11: Undefined

Timer/Counter Mode
- Unused

These two bits are used to determine how the CTM external pin changes state when a certain condition is reached. The function that these bits select depends upon in which mode the CTM is running.

In the Compare Match Output Mode, the CTIO1 and CTIO0 bits determine how the CTM output pin changes state when a compare match occurs from the Comparator A. The CTM output pin can be setup to switch high, switch low or to toggle its present state when a compare match occurs from the Comparator A. When the bits are both zero, then no change will take place on the output. The initial value of the CTM output pin should be setup using the CTOC bit in the CTMC1 register. Note that the output level requested by the CTIO1 and CTIO0 bits must be different from the initial value setup using the CTOC bit otherwise no change will occur on the CTM output pin when a compare match occurs. After the CTM output pin changes state, it can be reset to its initial level by changing the level of the CTON bit from low to high.

In the PWM Output Mode, the CTIO1 and CTIO0 bits determine how the CTM output pin changes state when a certain compare match condition occurs. The PWM output function is modified by changing these two bits. It is necessary to only change the values of the CTIO1 and CTIO0 bits only after the CTM has been switched off. Unpredictable PWM outputs will occur if the CTIO1 and CTIO0 bits are changed when the CTM is running.

Bit 3 **CTOC**: CTM CTP output control

Compare Match Output Mode
  0: Initial low
  1: Initial high

PWM Output Mode/Single Pulse Output Mode
  0: Active low
  1: Active high

This is the output control bit for the CTM output pin. Its operation depends upon whether CTM is being used in the Compare Match Output Mode or in the PWM Output Mode. It has no effect if the CTM is in the Timer/Counter Mode. In the Compare Match Output Mode it determines the logic level of the CTM output pin before a compare match occurs. In the PWM Output Mode it determines if the PWM signal is active high or active low.

Bit 2 **CTPOL**: CTM CTP output polarity control
  0: Non-invert
  1: Invert

This bit controls the polarity of the CTP output pin. When the bit is set high the CTM output pin will be inverted and not inverted when the bit is zero. It has no effect if the CTM is in the Timer/Counter Mode.

Bit 1 **CTDPX**: CTM PWM duty/period control
  0: CCRP – period; CCRA – duty
  1: CCRP – duty; CCRA – period

This bit determines which of the CCRA and CCRP registers are used for period and duty control of the PWM waveform.

Bit 0 **CTCCLR**: CTM counter clear condition selection
  0: CTM Comparator P match
  1: CTM Comparator A match

This bit is used to select the method which clears the counter. Remember that the Compact TM contains two comparators, Comparator A and Comparator P, either of which can be selected to clear the internal counter. With the CTCCLR bit set high, the counter will be cleared when a compare match occurs from the Comparator A. When the bit is low, the counter will be cleared when a compare match occurs from the Comparator P or with a counter overflow. A counter overflow clearing method can only be implemented if the CCRP bits are all cleared to zero. The CTCCLR bit is not used in the PWM Output mode.

- **CTMDL Register**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**: CTM counter low byte register bit 7 ~ bit 0
CTM 10-bit counter bit 7 ~ bit 0

- **CTMDH Register**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as “0”

Bit 1~0 **D9~D8**: CTM counter high byte register bit 1 ~ bit 0
CTM 10-bit counter bit 9 ~ bit 8

- **CTMAL Register**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
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 7~0 **D7~D0**: CTM CCRA low byte register bit 7 ~ bit 0
CTM 10-bit CCRA bit 7 ~ bit 0

- **CTMAH Register**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as “0”

Bit 1~0 **D9~D8**: CTM CCRA high byte register bit 7 ~ bit 0
CTM 10-bit CCRA bit 9 ~ bit 8

Compact Type TM Operation Modes

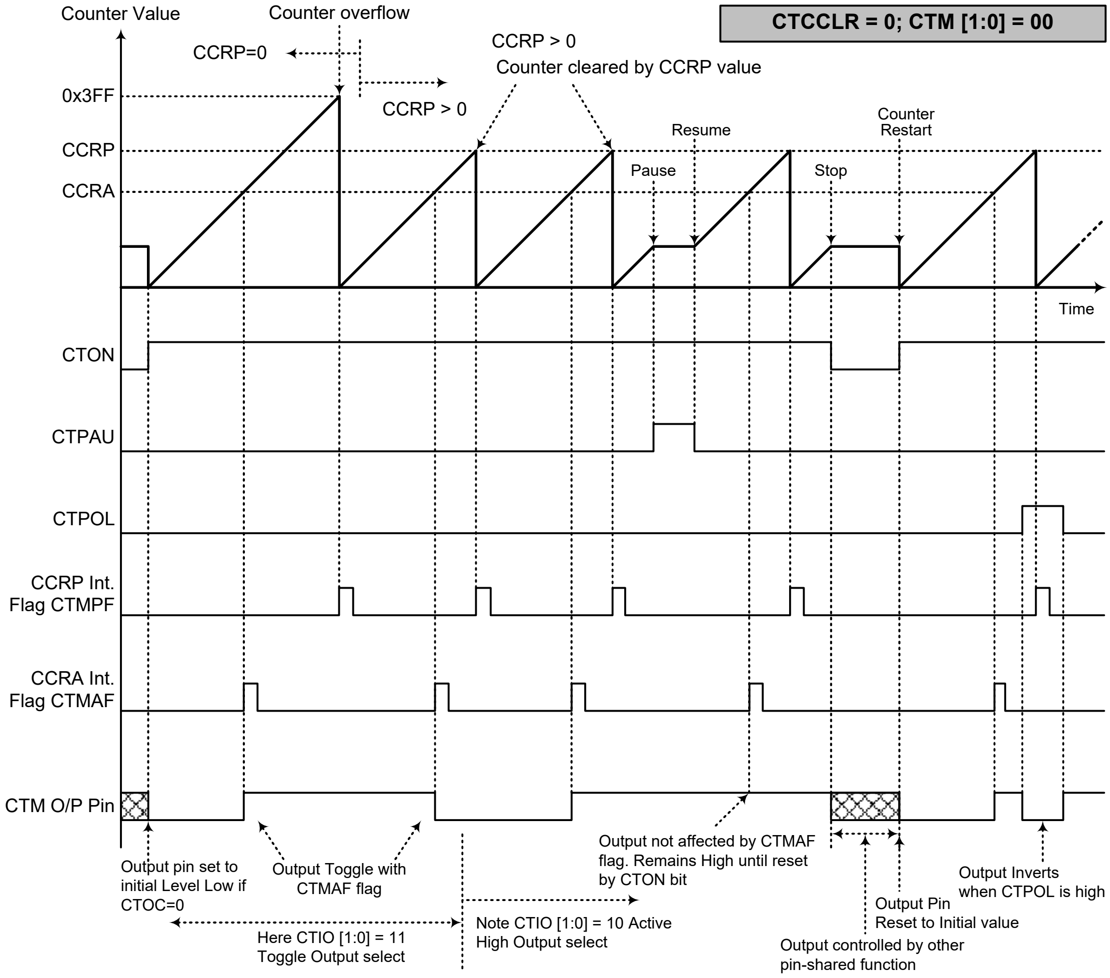
The Compact Type TM can operate in one of three operating modes, Compare Match Output Mode, PWM Output Mode or Timer/Counter Mode. The operating mode is selected using the CTM1 and CTM0 bits in the CTMC1 register.

Compare Match Output Mode

To select this mode, bits CTM1 and CTM0 in the CTMC1 register, should be set to 00 respectively. In this mode once the counter is enabled and running it can be cleared by three methods. These are a counter overflow, a compare match from Comparator A and a compare match from Comparator P. When the CTCCLR bit is low, there are two ways in which the counter can be cleared. One is when a compare match from Comparator P, the other is when the CCRP bits are all zero which allows the counter to overflow. Here both CTMAF and CTMPF interrupt request flags for Comparator A and Comparator P respectively, will both be generated.

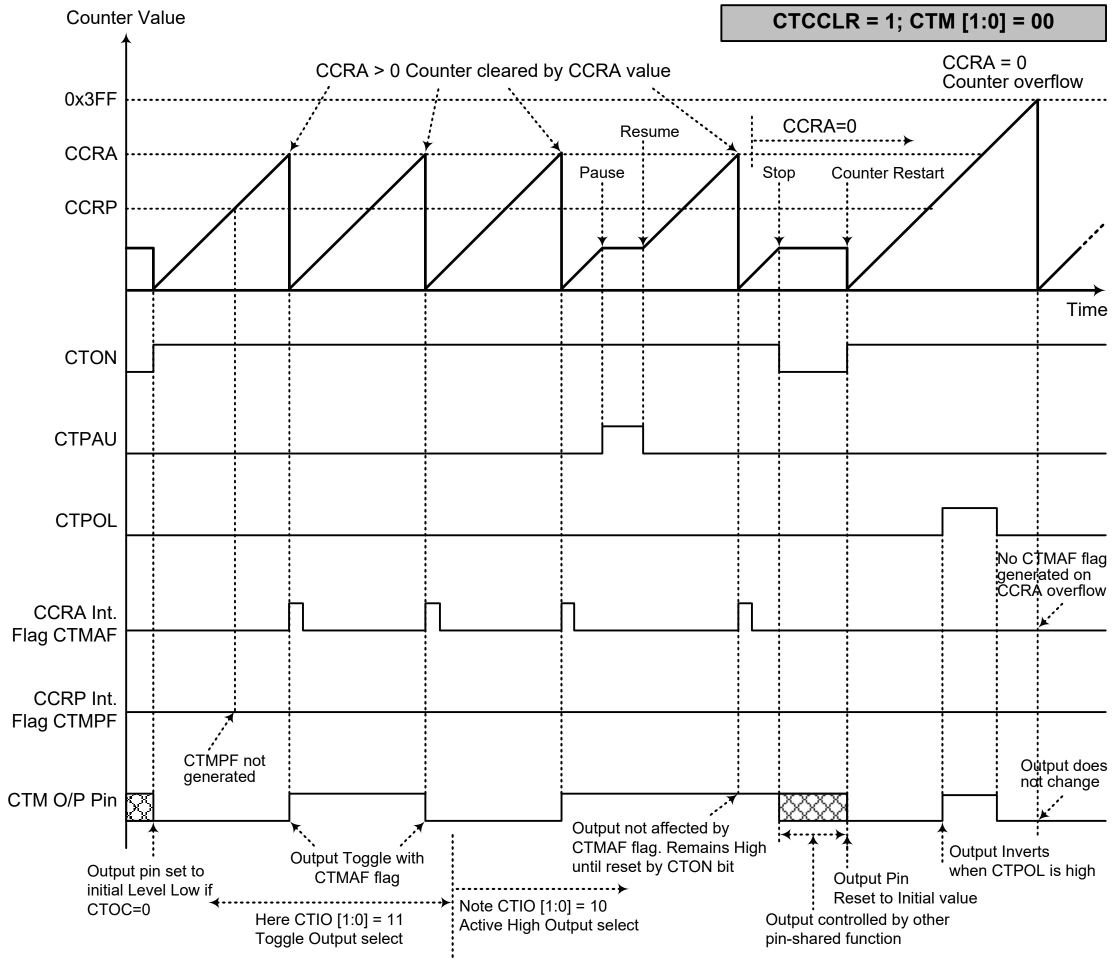
If the CTCCLR bit in the CTMC1 register is high then the counter will be cleared when a compare match occurs from Comparator A. However, here only the CTMAF interrupt request flag will be generated even if the value of the CCRP bits is less than that of the CCRA registers. Therefore when CTCCLR is high no CTMPF interrupt request flag will be generated. If the CCRA bits are all zero, the counter will overflow when it reaches its maximum 10-bit, 3FF Hex, value. However, here the CTMAF interrupt request flag will not be generated.

As the name of the mode suggests, after a comparison is made, the CTM output pin, will change state. The CTM output pin condition however only changes state when a CTMAF interrupt request flag is generated after a compare match occurs from Comparator A. The CTMPF interrupt request flag, generated from a compare match occurs from Comparator P, will have no effect on the CTM output pin. The way in which the CTM output pin changes state are determined by the condition of the CTIO1 and CTIO0 bits in the CTMC1 register. The CTM output pin can be selected using the CTIO1 and CTIO0 bits to go high, to go low or to toggle from its present condition when a compare match occurs from Comparator A. The initial condition of the CTM output pin, which is setup after the CTON bit changes from low to high, is setup using the CTOC bit. Note that if the CTIO1 and CTIO0 bits are zero then no pin change will take place.



Compare Match Output Mode – CTCCLR=0

Note: 1. With CTCCLR=0 a Comparator P match will clear the counter
2. The CTM output pin is controlled only by the CTMAF flag
3. The output pin is reset to its initial state by a CTON bit rising edge



Compare Match Output Mode – CTCCLR=1

Note: 1. With CTCCLR=1 a Comparator A match will clear the counter
2. The CTM output pin is controlled only by the CTMAF flag
3. The output pin is reset to its initial state by a CTON bit rising edge
4. A CTMPF flag is not generated when CTCCLR=1

Timer/Counter Mode

To select this mode, bits CTM1 and CTM0 in the CTMC1 register should be set to 11 respectively. The Timer/Counter Mode operates in an identical way to the Compare Match Output Mode generating the same interrupt flags. The exception is that in the Timer/Counter Mode the CTM output pin is not used. Therefore the above description and Timing Diagrams for the Compare Match Output Mode can be used to understand its function. As the CTM output pin is not used in this mode, the pin can be used as a normal I/O pin or other pin-shared function.

PWM Output Mode

To select this mode, bits CTM1 and CTM0 in the CTMC1 register should be set to 10 respectively. The PWM function within the CTM is useful for applications which require functions such as motor control, heating control, illumination control, etc. By providing a signal of fixed frequency but of varying duty cycle on the CTM output pin, a square wave AC waveform can be generated with varying equivalent DC RMS values.

As both the period and duty cycle of the PWM waveform can be controlled, the choice of generated waveform is extremely flexible. In the PWM Output Mode, the CTCCLR bit has no effect as the PWM period. Both of the CCRA and CCRP registers are used to generate the PWM waveform, the CCRP is used to clear the internal counter and thus control the PWM waveform frequency, while the CCRA is used to control the duty cycle. Which register is used to control either frequency or duty cycle is determined using the CTDPX bit in the CTMC1 register. The PWM waveform frequency and duty cycle can therefore be controlled by the values in the CCRA and CCRP registers.

An interrupt flag, one for each of the CCRA and CCRP, will be generated when a compare match occurs from either Comparator A or Comparator P. The CTOC bit in the CTMC1 register is used to select the required polarity of the PWM waveform while the two CTIO1 and CTIO0 bits are used to enable the PWM output or to force the CTM output pin to a fixed high or low level. The CTPOL bit is used to reverse the polarity of the PWM output waveform.

- **10-bit CTM, PWM Output Mode, Edge-aligned Mode, CTDPX=0**

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	128	256	384	512	640	768	896	1024
Duty	CCRA							

If f_{SYS}=7.5MHz, CTM clock source is $f_{SYS}/4$, CCRP=100b, CCRA=128,

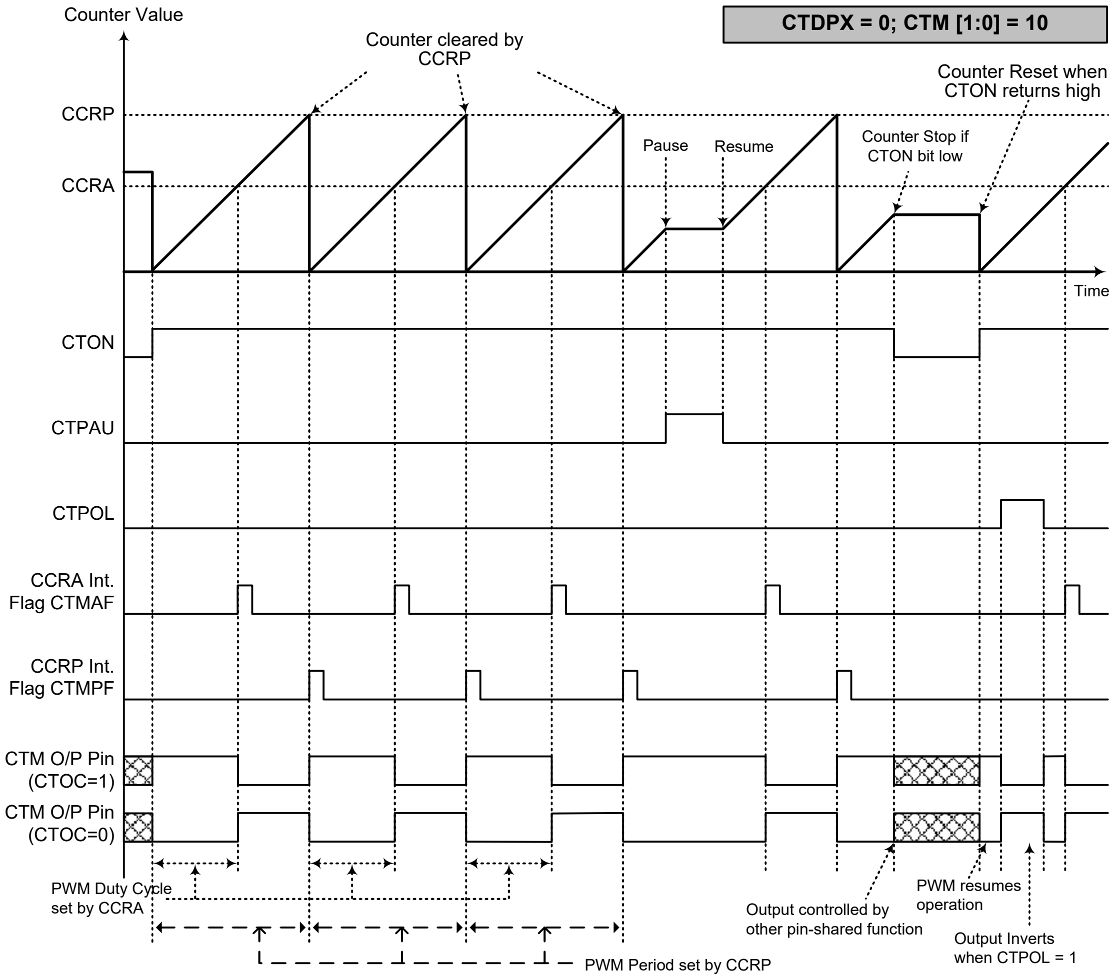
The CTM PWM output frequency=$(f_{SYS}/4)/512=f_{SYS}/2048$=3.66kHz, duty=128/512=25%.

If the Duty value defined by the CCRA register is equal to or greater than the Period value, then the PWM output duty is 100%.

- **10-bit CTM, PWM Output Mode, Edge-aligned Mode, CTDPX=1**

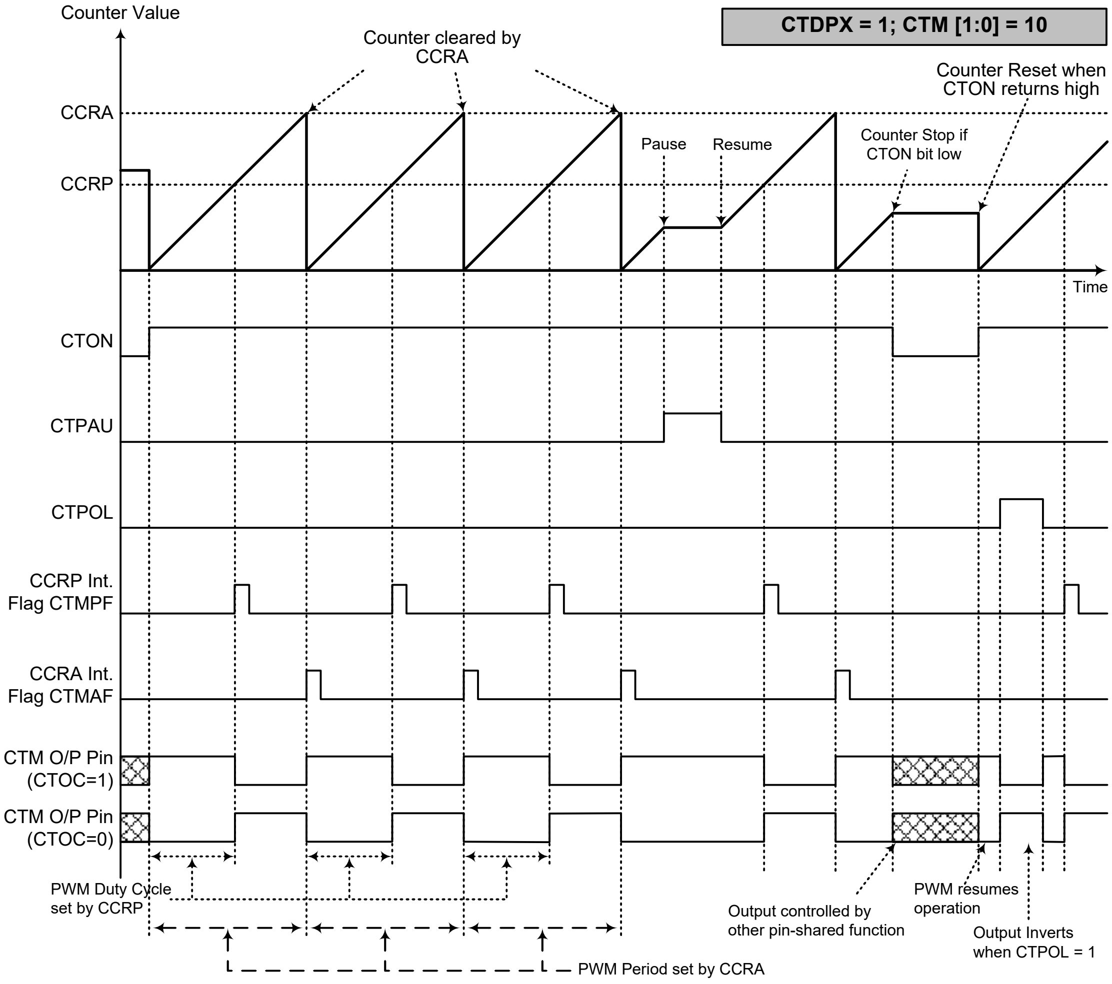
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	CCRA							
Duty	128	256	384	512	640	768	896	1024

The PWM output period is determined by the CCRA register value together with the CTM clock while the PWM duty cycle is defined by the CCRP register value except when the CCRP value is equal to 0.



PWM Output Mode – CTDPX=0

Note: 1. Here CTDPX=0 – Counter cleared by CCRP
2. A counter clear sets the PWM Period
3. The internal PWM function continues running even when CTIO[1:0]=00 or 01
4. The CTCCLR bit has no influence on PWM operation

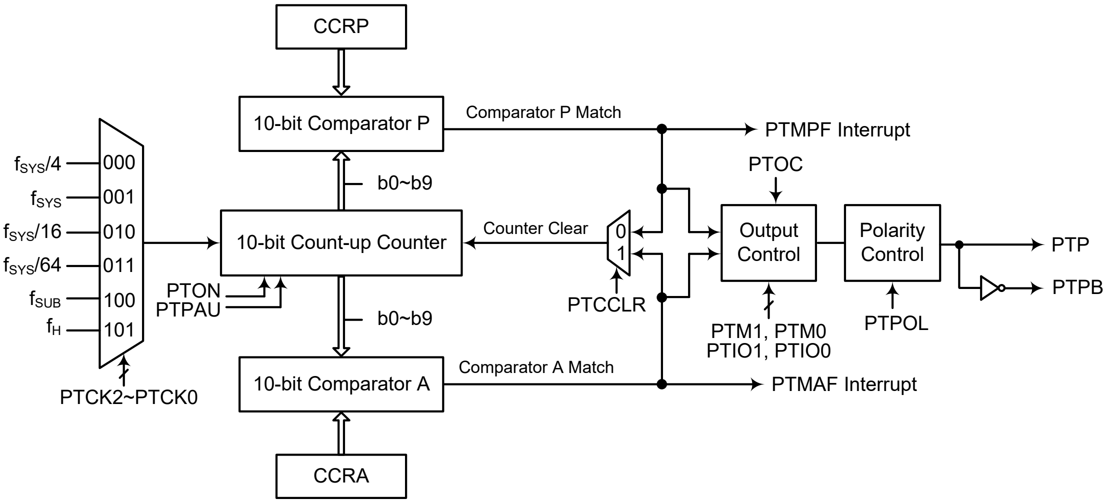


PWM Output Mode – CTDPX=1

Note: 1. Here CTDPX=1 – Counter cleared by CCRA
2. A counter clear sets the PWM Period
3. The internal PWM function continues even when CTIO[1:0]=00 or 01
4. The CTCCLR bit has no influence on PWM operation

Periodic Type TM – PTM

The Periodic Type TM contains four operating modes, which are Compare Match Output, Timer/Event Counter, Single Pulse Output and PWM Output modes.



10-Bit Periodic Type TM Block Diagram

Periodic Type TM Operation

The size of Periodic Type TM is 10-bit wide and its core is a 10-bit count-up counter which is driven by a user selectable internal or external clock source. There are also two internal comparators with the names, Comparator A and Comparator P. These comparators will compare the value in the counter with CCRP and CCRA registers. The CCRP and CCRA comparators are 10-bit wide whose value is respectively compared with all counter bits.

The only way of changing the value of the 10-bit counter using the application program is to clear the counter by changing the PTON bit from low to high. The counter will also be cleared automatically by a counter overflow or a compare match with one of its associated comparators. When these conditions occur, a PTM interrupt signal will also usually be generated. The Periodic Type TM can operate in a number of different operational modes, can be driven by different clock sources and can also control the outputs. All operating setup conditions are selected using relevant internal registers.

Periodic Type TM Register Description

Overall operation of the Periodic Type TM is controlled using a series of registers. A read only register pair exists to store the internal counter 10-bit value, while two read/write register pairs exist to store the internal 10-bit CCRA and CCRP value. The remaining two registers are control registers which setup the different operating and control modes.

Register Name	Bit							
	7	6	5	4	3	2	1	0
PTMC0	PTPAU	PTCK2	PTCK1	PTCK0	PTON	—	—	—
PTMC1	PTM1	PTM0	PTIO1	PTIO0	PTOC	PTPOL	D1	PTCCLR
PTMDL	D7	D6	D5	D4	D3	D2	D1	D0
PTMDH	—	—	—	—	—	—	D9	D8
PTMAL	D7	D6	D5	D4	D3	D2	D1	D0
PTMAH	—	—	—	—	—	—	D9	D8
PTMRPL	D7	D6	D5	D4	D3	D2	D1	D0
PTMRPH	—	—	—	—	—	—	D9	D8

10-Bit Periodic Type TM Register List

- **PTMC0 Register**

Bit	7	6	5	4	3	2	1	0
Name	PTPAU	PTCK2	PTCK1	PTCK0	PTON	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—
POR	0	0	0	0	0	—	—	—

Bit 7 **PTPAU**: PTM counter pause control
0: Run
1: Pause

The counter can be paused by setting this bit high. Clearing the bit to zero restores normal counter operation. When in a Pause condition the PTM will remain powered up and continue to consume power. The counter will retain its residual value when this bit changes from low to high and resume counting from this value when the bit changes to a low value again.

Bit 6~4 **PTCK2~PTCK0**: PTM counter clock selection
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{SUB}
101: f_H
110: Reserved
111: Reserved

These three bits are used to select the clock source for the PTM. The clock source f_{SYS} is the system clock, while f_H and f_{SUB} are other internal clocks, the details of which can be found in the Operating Modes and System Clocks section.

Bit 3 **PTON**: PTM counter on/off control
0: Off
1: On

This bit controls the overall on/off function of the PTM. Setting the bit high enables the counter to run while clearing the bit disables the PTM. Clearing this bit to zero will stop the counter from counting and turn off the PTM which will reduce its power consumption. When the bit changes state from low to high the internal counter value will be reset to zero, however when the bit changes from high to low, the internal counter will retain its residual value until the bit returns high again. If the PTM is in the Compare Match Output Mode, PWM Output Mode or Single Pulse Output Mode then the PTM output will be reset to its initial condition, as specified by the PTOC bit, when the PTON bit changes from low to high.

Bit 2~0 Unimplemented, read as “0”

- **PTMC1 Register**

Bit	7	6	5	4	3	2	1	0
Name	PTM1	PTM0	PTIO1	PTIO0	PTOC	PTPOL	D1	PTCCLR
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 7~6 **PTM1~PTM0**: PTM operating mode selection
00: Compare Match Output Mode
01: Undefined
10: PWM Output Mode or Single Pulse Output Mode
11: Timer/Counter Mode

These bits setup the required operating mode for the PTM. To ensure reliable operation the PTM should be switched off before any changes are made to the PTM1 and PTM0 bits. In the Timer/Counter Mode, the PTM output state is undefined.

Bit 5~4 **PTIO1~PTIO0**: PTM output function selection

Compare Match Output Mode
00: No change
01: Output low
10: Output high
11: Toggle output

PWM Output Mode/Single Pulse Output Mode
00: PWM output inactive state
01: PWM output active state
10: PWM output
11: Single Pulse Output

Timer/Counter Mode
Unused

These two bits are used to determine how the PTM output changes state when a certain condition is reached. The function that these bits select depends upon in which mode the PTM is running.

In the Compare Match Output Mode, the PTIO1 and PTIO0 bits determine how the PTM output changes state when a compare match occurs from the Comparator A. The PTM output can be setup to switch high, switch low or to toggle its present state when a compare match occurs from the Comparator A. When the bits are both zero, then no change will take place on the output. The initial value of the PTM output should be setup using the PTOC bit in the PTMC1 register. Note that the output level requested by the PTIO1 and PTIO0 bits must be different from the initial value setup using the PTOC bit otherwise no change will occur on the PTM output when a compare match occurs. After the PTM output changes state, it can be reset to its initial level by changing the level of the PTON bit from low to high.

In the PWM Output Mode, the PTIO1 and PTIO0 bits determine how the TM output changes state when a certain compare match condition occurs. The PTM output function is modified by changing these two bits. It is necessary to only change the values of the PTIO1 and PTIO0 bits only after the PTM has been switched off. Unpredictable PWM outputs will occur if the PTIO1 and PTIO0 bits are changed when the PTM is running.

Bit 3 **PTOC**: PTM PTP output control

Compare Match Output Mode
0: Initial low
1: Initial high

PWM Output Mode/Single Pulse Output Mode
0: Active low
1: Active high

This is the output control bit for the PTM output. Its operation depends upon whether PTM is being used in the Compare Match Output Mode or in the PWM Output Mode/Single Pulse Output Mode. It has no effect if the PTM is in the Timer/Counter Mode. In the Compare Match Output Mode it determines the logic level of the PTM output before a compare match occurs. In the PWM Output Mode it determines if the PWM signal is active high or active low. In the Single Pulse Output Mode it determines the logic level of the PTM output when PTON bit changes from low to high.

Bit 2 **PTPOL**: PTM PTP output polarity control
0: Non-invert
1: Invert

This bit controls the polarity of the PTP output. When the bit is set high the PTM output will be inverted and not inverted when the bit is zero. It has no effect if the PTM is in the Timer/Counter Mode.

Bit 1 **D1**: Reserved, must be fixed at “0”

Bit 0 **PTCCLR**: PTM counter clear condition selection
0: Comparator P match
1: Comparator A match

This bit is used to select the method which clears the counter. Remember that the Periodic type TM contains two comparators, Comparator A and Comparator P, either of which can be selected to clear the internal counter. With the PTCCLR bit set high, the counter will be cleared when a compare match occurs from the Comparator A. When the bit is low, the counter will be cleared when a compare match occurs from the Comparator P or with a counter overflow. A counter overflow clearing method can only be implemented if the CCRP bits are all cleared to zero. The PTCCLR bit is not used in the PWM Output Mode or Single Pulse Output Mode.

• PTMDL Register

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **D7~D0**: PTM counter low byte register bit 7 ~ bit 0
PTM 10-bit counter bit 7 ~ bit 0

• PTMDH Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as “0”

Bit 1~0 **D9~D8**: PTM counter high byte register bit 1 ~ bit 0
PTM 10-bit counter bit 9 ~ bit 8

• PTMAL Register

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
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 7~0 **D7~D0**: PTM CCRA low byte register bit 7 ~ bit 0
PTM 10-bit CCRA bit 7 ~ bit 0

- **PTMAH Register**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **D9~D8**: PTM CCRA high byte register bit 1 ~ bit 0
PTM 10-bit CCRA bit 9 ~ bit 8

- **PTMRPL Register**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
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 7~0 **D7~D0**: PTM CCRP low byte register bit 7 ~ bit 0
PTM 10-bit CCRP bit 7 ~ bit 0

- **PTMRPH Register**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **D9~D8**: PTM CCRP high byte register bit 1 ~ bit 0
PTM 10-bit CCRP bit 9 ~ bit 8

Periodic Type TM Operation Modes

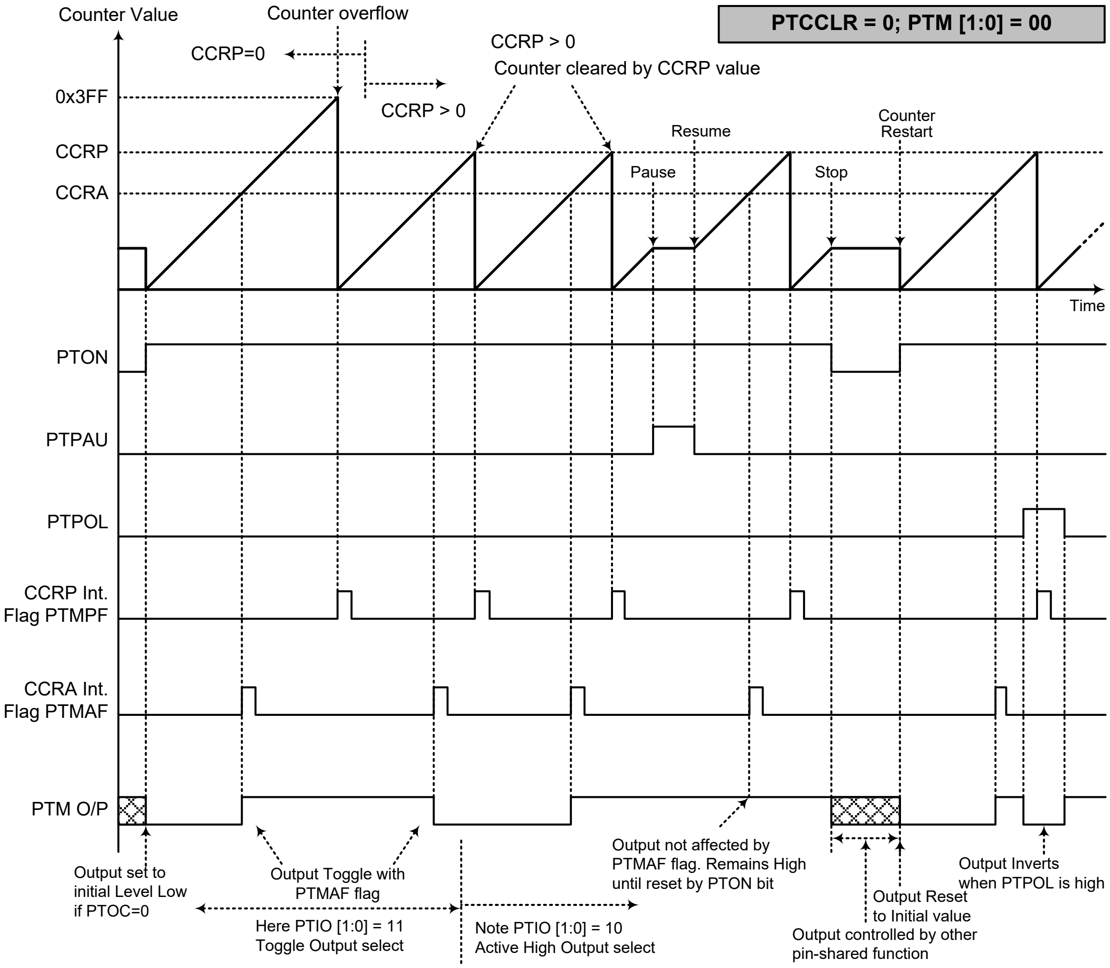
The Periodic Type TM can operate in one of four operating modes, Compare Match Output Mode, PWM Output Mode, Single Pulse Output Mode or Timer/Counter Mode. The operating mode is selected using the PTM1 and PTM0 bits in the PTMC1 register.

Compare Match Output Mode

To select this mode, bits PTM1 and PTM0 in the PTMC1 register, should be set to 00 respectively. In this mode once the counter is enabled and running it can be cleared by three methods. These are a counter overflow, a compare match from Comparator A and a compare match from Comparator P. When the PTCCLR bit is low, there are two ways in which the counter can be cleared. One is when a compare match from Comparator P, the other is when the CCRP bits are all zero which allows the counter to overflow. Here both PTMAF and PTMPF interrupt request flags for Comparator A and Comparator P respectively, will both be generated.

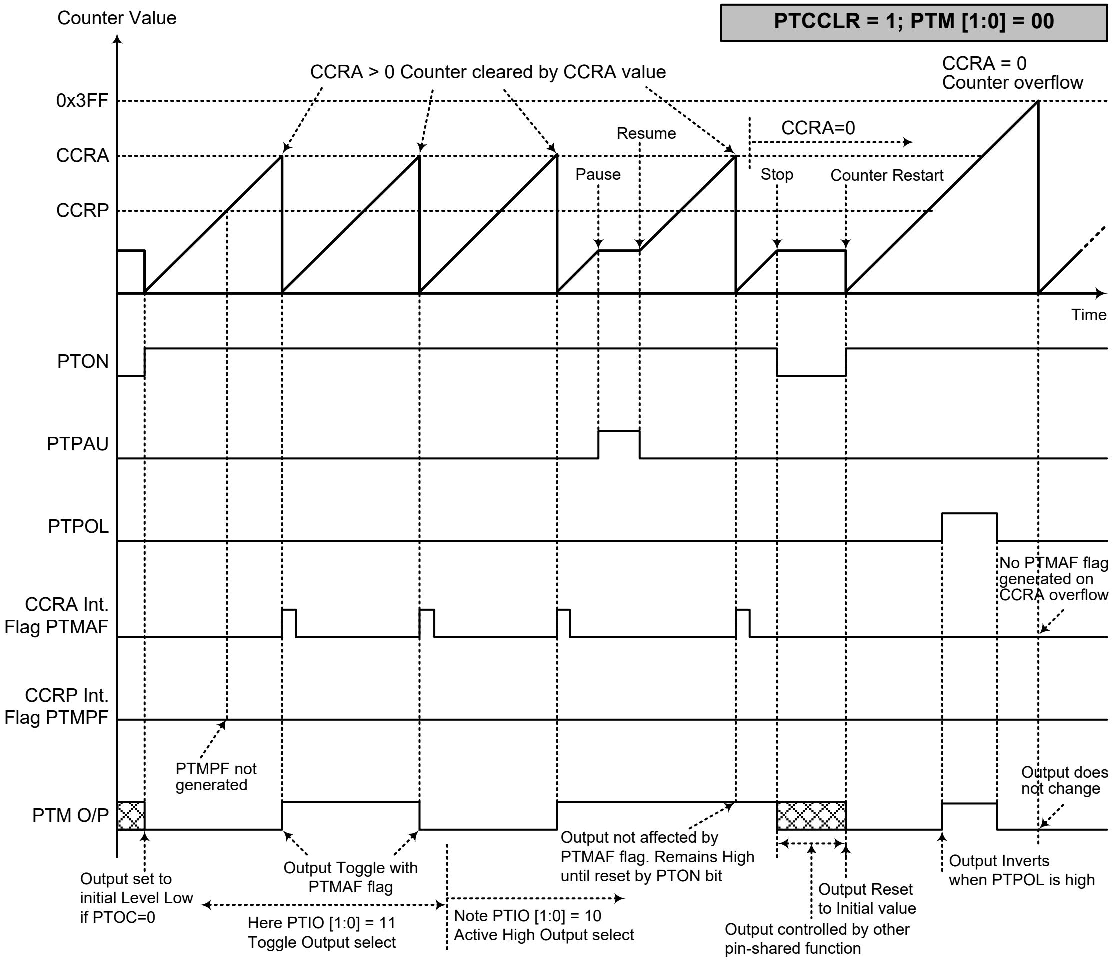
If the PTCCLR bit in the PTMC1 register is high then the counter will be cleared when a compare match occurs from Comparator A. However, here only the PTMAF interrupt request flag will be generated even if the value of the CCRP bits is less than that of the CCRA registers. Therefore when PTCCLR is high no PTMPF interrupt request flag will be generated. In the Compare Match Output Mode, the CCRA can not be cleared to "0". If the CCRA bits are all zero, the counter will overflow when its reaches its maximum 10-bit, 3FF Hex, value, however here the PTMAF interrupt request flag will not be generated.

As the name of the mode suggests, after a comparison is made, the PTM output will change state. The PTM output condition however only changes state when a PTMAF interrupt request flag is generated after a compare match occurs from Comparator A. The PTMPF interrupt request flag, generated from a compare match occurs from Comparator P, will have no effect on the PTM output. The way in which the PTM output changes state are determined by the condition of the PTIO1 and PTIO0 bits in the PTMC1 register. The PTM output can be selected using the PTIO1 and PTIO0 bits to go high, to go low or to toggle from its present condition when a compare match occurs from Comparator A. The initial condition of the PTM output, which is setup after the PTON bit changes from low to high, is setup using the PTOC bit. Note that if the PTIO1 and PTIO0 bits are zero then no output change will take place.



Compare Match Output Mode – PTCCLR=0

Note: 1. With PTCCLR=0, a Comparator P match will clear the counter
2. The PTM output is controlled only by the PTMAF flag
3. The output is reset to its initial state by a PTON bit rising edge



Compare Match Output Mode – PTCCLR=1

Note: 1. With PTCCLR=1, a Comparator A match will clear the counter
2. The PTM output is controlled only by the PTMAF flag
3. The output is reset to its initial state by a PTON bit rising edge
4. A PTMPF flag is not generated when PTCCLR=1

Timer/Counter Mode

To select this mode, bits PTM1 and PTM0 in the PTMC1 register should be set to 11 respectively. The Timer/Counter Mode operates in an identical way to the Compare Match Output Mode generating the same interrupt flags. The exception is that in the Timer/Counter Mode the PTM output is not used. Therefore the above description and Timing Diagrams for the Compare Match Output Mode can be used to understand its function.

PWM Output Mode

To select this mode, bits PTM1 and PTM0 in the PTMC1 register should be set to 10 respectively and also the PTIO1 and PTIO0 bits should be set to 10 respectively. The PWM function within the PTM is useful for applications which require functions such as motor control, heating control, illumination control, etc. By providing a signal of fixed frequency but of varying duty cycle on the PTM output, a square wave AC waveform can be generated with varying equivalent DC RMS values.

As both the period and duty cycle of the PWM waveform can be controlled, the choice of generated waveform is extremely flexible. In the PWM Output mode, the PTCCLR bit has no effect as the PWM period. Both of the CCRP and CCRA registers are used to generate the PWM waveform, the CCRP is used to clear the internal counter and thus control the PWM waveform frequency, while the CCRA is used to control the duty cycle. The PWM waveform frequency and duty cycle can therefore be controlled by the values in the CCRA and CCRP registers.

An interrupt flag, one for each of the CCRA and CCRP, will be generated when a compare match occurs from either Comparator A or Comparator P. The PTOC bit in the PTMC1 register is used to select the required polarity of the PWM waveform while the two PTIO1 and PTIO0 bits are used to enable the PWM output or to force the PTM output to a fixed high or low level. The PTPOL bit is used to reverse the polarity of the PWM output waveform.

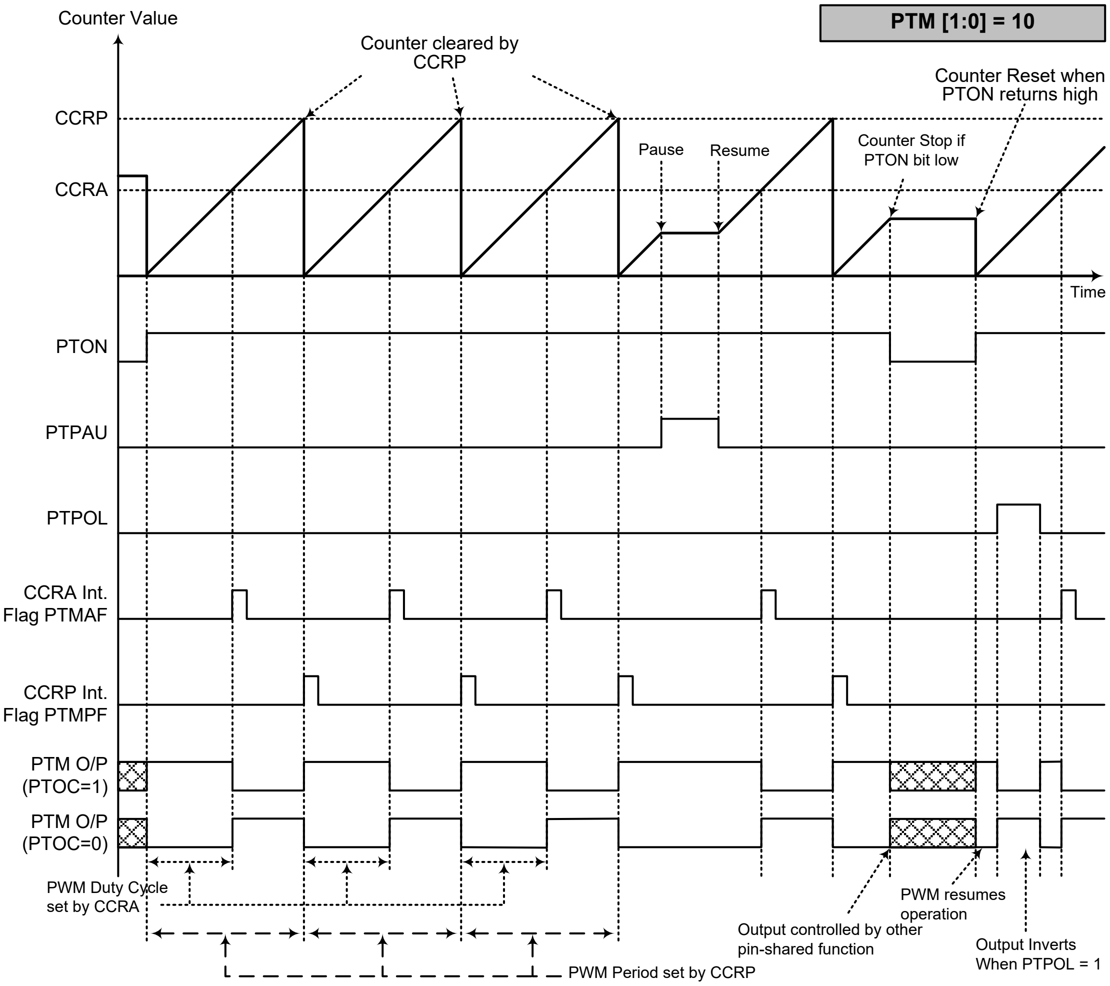
- **10-bit PTM, PWM Output Mode, Edge-aligned Mode**

CCRP	1~1023	0
Period	1~1023	1024
Duty	CCRA	

If f_{SYS}=7.5MHz, PTM clock source select $f_{SYS}/4$, CCRP=512 and CCRA=128,

The PTM PWM output frequency=$(f_{SYS}/4)/512=f_{SYS}/2048$=3.66kHz, duty=128/512=25%.

If the Duty value defined by the CCRA register is equal to or greater than the Period value, then the PWM output duty is 100%.



PWM Output Mode

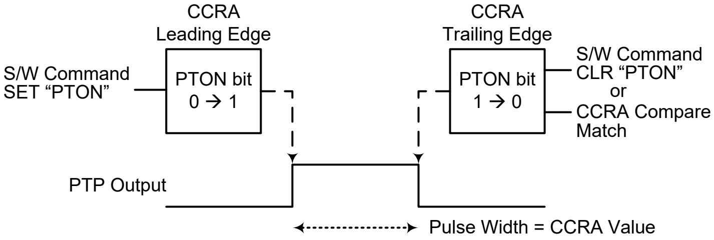
Note: 1. The counter is cleared by CCRP
2. A counter clear sets the PWM Period
3. The internal PWM function continues running even when PTIO[1:0]=00 or 01
4. The PTCCLR bit has no influence on PWM operation

Single Pulse Output Mode

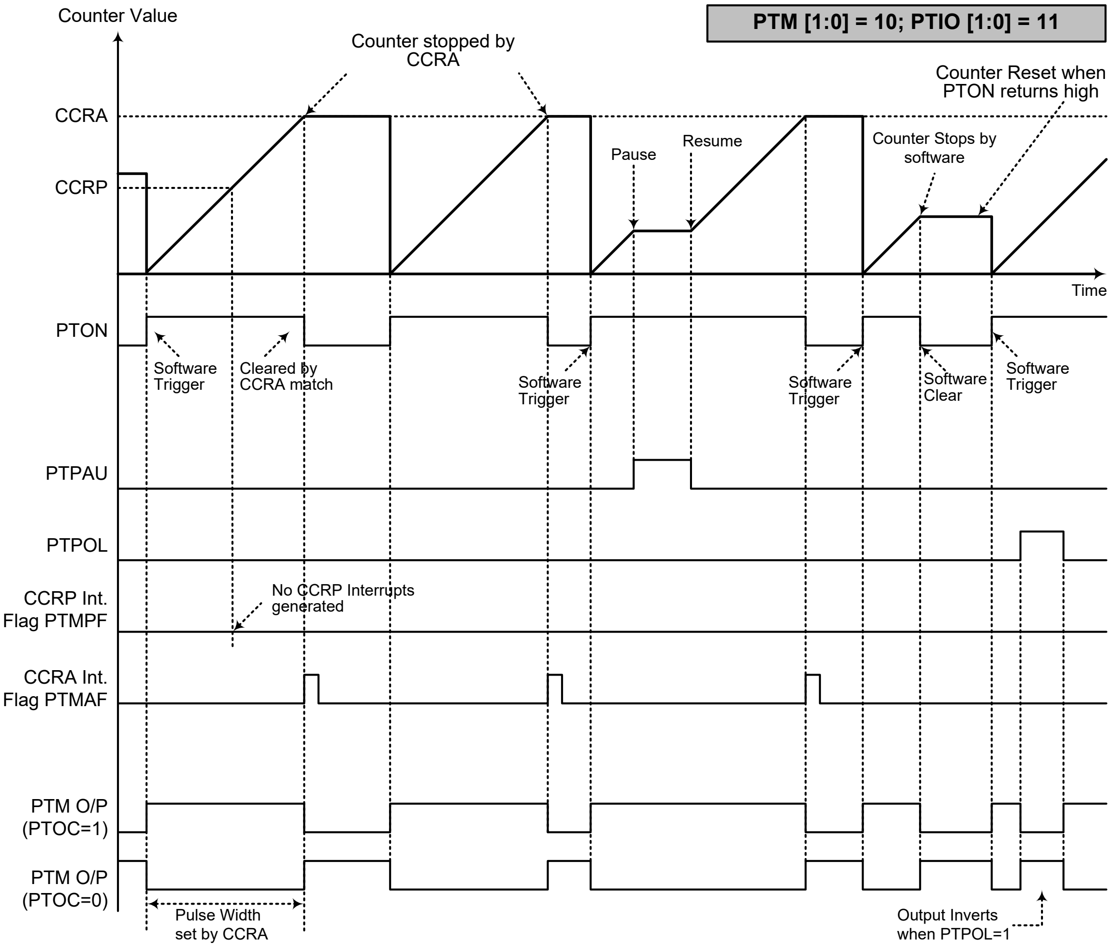
To select this mode, bits PTM1 and PTM0 in the PTMC1 register should be set to 10 respectively and also the PTIO1 and PTIO0 bits should be set to 11 respectively. The Single Pulse Output Mode, as the name suggests, will generate a single shot pulse on the PTM output.

The trigger for the pulse output leading edge is a low to high transition of the PTON bit, which can be implemented using the application program. When the PTON bit transitions to a high level, the counter will start running and the pulse leading edge will be generated. The PTON bit should remain high when the pulse is in its active state. The generated pulse trailing edge will be generated when the PTON bit is cleared to zero, which can be implemented using the application program or when a compare match occurs from Comparator A.

However a compare match from Comparator A will also automatically clear the PTON bit and thus generate the Single Pulse output trailing edge. In this way the CCRA value can be used to control the pulse width. A compare match from Comparator A will also generate a PTM interrupt. The counter can only be reset back to zero when the PTON bit changes from low to high when the counter restarts. In the Single Pulse Output Mode CCRP is not used. The PTCCLR is not used in this Mode.



Single Pulse Generation

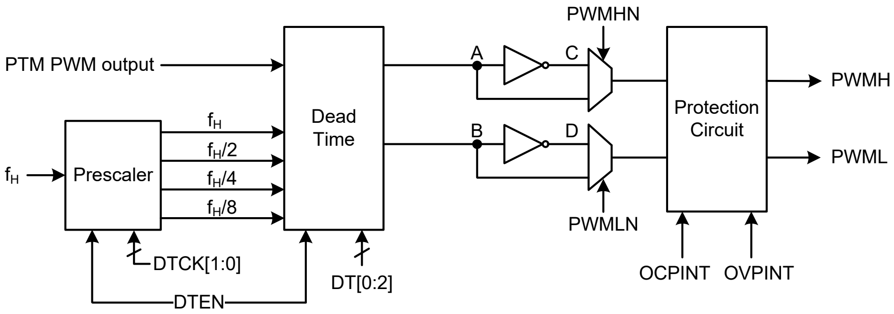


Single Pulse Output Mode

Note: 1. Counter stopped by CCRA
2. CCRP is not used
3. The pulse triggered by setting the PTON bit high
4. In the Single Pulse Output Mode, PTIO [1:0] must be set to “11” and can not be changed

Complementary PWM Output with Dead Time

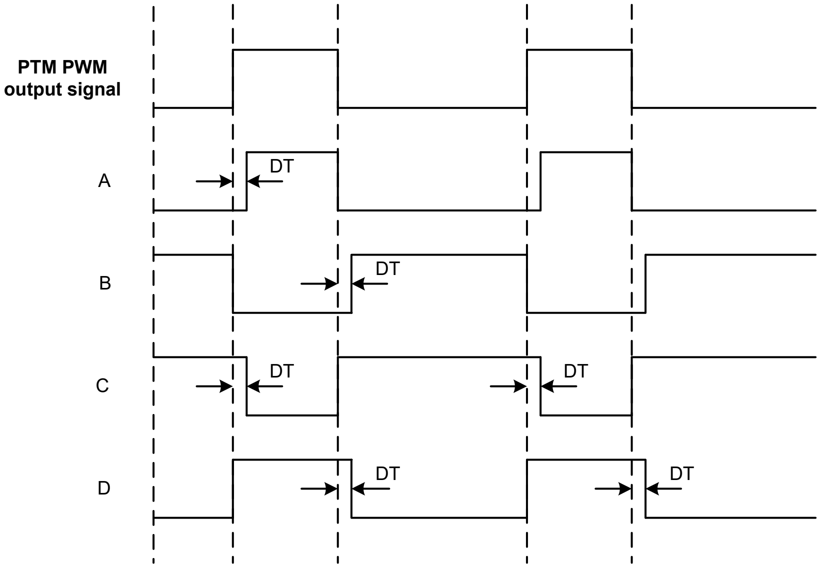
The device provides a complementary output pair of signals which can be used as a PWM driver signal. The PWM signal is sourced from the PTM PWM output which is an active high signal. A dead time will be inserted into the PTM PWM output signals to prevent excessive DC currents. In addition to register configuration, the complementary PWM output can also be stopped by an OCP or OVP condition occurrence, when such condition occurs and the corresponding control bit in the OCVPC register is enabled, the PWM output will stop and the PWM output pair status will be forced to a certain level determined by the PWMHOPS and PWMLOPS bits.



Complementary PWM Output with Dead Time Block Diagram

Dead Time Insertion

The complementary PWM output circuit provides a dead time insertion function. By setting the DTEN bit in the CPR register, the dead time generator and prescaler will be enabled. The clock source of the prescaler originates from the internal clock f_H and the division ratio is determined by the DTCK[1:0] bits. When the related register bits are properly configured, a dead time, which is programmable using the DT[2:0] bits in the CPR register, will be inserted to prevent excessive DC currents. The dead time will be inserted whenever the rising edge of the dead time generator input signal, namely the PTM PWM output signal, occurs.



Complementary PWM Output with Dead Time Control

Complementary PWM Registers

The complementary PWM output function can be controlled using internal registers. The CPR register is used to control the dead time function enable/disable, PWMH/PWML inverse signal selection, dead time prescaler selection and dead time selection. The OCVPC register is used to control the protection circuit and determine the PWM output pair status when the complementary PWM output circuit is stopped.

- **CPR Register**

Bit	7	6	5	4	3	2	1	0
Name	DTEN	PWMHN	PWMLN	DTCK1	DTCK0	DT2	DT1	DT0
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 **DTEN**: Dead Time on/off control
0: Dead time & prescaler off
1: Dead time & Prescaler on
When this bit is cleared to zero, the PWMH and PWML status are determined by the PWMHOPS and PWMLOPS bits respectively.

Bit 6 **PWMHN**: PWMH inverse signal selection
0: PWMH=A
1: PWMH=C

Bit 5 **PWMLN**: PWML inverse signal selection
0: PWML=B
1: PWML=D

Bit 4~3 **DTCK1~DTCK0**: Dead time prescaler selection
00: $f_D=f_H$
01: $f_D=f_H/2$
10: $f_D=f_H/4$
11: $f_D=f_H/8$

Bit 2~0 **DT2~DT0**: Dead time selection
000: $[(1/f_D)-(1/f_H)] \sim (1/f_D)$
001: $[(2/f_D)-(1/f_H)] \sim (2/f_D)$
010: $[(3/f_D)-(1/f_H)] \sim (3/f_D)$
011: $[(4/f_D)-(1/f_H)] \sim (4/f_D)$
100: $[(5/f_D)-(1/f_H)] \sim (5/f_D)$
101: $[(6/f_D)-(1/f_H)] \sim (6/f_D)$
110: $[(7/f_D)-(1/f_H)] \sim (7/f_D)$
111: $[(8/f_D)-(1/f_H)] \sim (8/f_D)$
Note: $t_D=1/f_D$.

- **OCVPC Register**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PWMHOPS	PWMLOPS	PWMOCEN	PWMOVEN
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	1	0	0	0

Bit 7~4 Unimplemented, read as “0”

Bit 3 **PWMHOPS**: PWMH output status when complementary PWM output is stopped
0: Output 0
1: Output 1
When the complementary PWM output circuit is stopped by clearing the DTEN bit to zero, or by an OVP or OCP condition occurrence, the PWMH output status will be forced to output 1 if the PWMHOPS bit is set to “1”, otherwise the output status will be forced to output 0 if this bit is cleared to “0”. Note that configuring this bit has no effect when the complementary PWM output circuit is in normal operation.

Bit 2 **PWMLOPS**: PWML output status when complementary PWM output is stopped
0: Output 0
1: Output 1

When the complementary PWM output circuit is stopped by clearing the DTEN bit to zero, or by an OVP or OCP condition occurrence, the PWML output status will be forced to output 1 if the PWMLOPS bit is set, otherwise the output status will be forced to output 0 if this bit is cleared to zero. Note that configuring this bit has no effect when the complementary PWM output circuit is in normal operation.

Bit 1 **PWMOCEN**: PWM over current protection enable control
0: Disable
1: Enable

This bit is used to determine if an OCP condition occurrence will affect the PWM output circuit. If an OCP condition occurs and this bit is set high, the DTEN will be automatically cleared to zero by hardware to stop the complementary PWM output circuit. In this case, the PWMH and PWML status will be forced to a fixed high or low level determined by the PWMHOPS and PWMLOPS bits respectively.

Bit 0 **PWMOVEN**: PWM over voltage protection enable control
0: Disable
1: Enable

This bit is used to determine if an OVP condition occurrence will affect the PWM output circuit. If an OVP condition occurs and this bit is set high, the DTEN will be automatically cleared to zero by hardware to stop the complementary PWM output circuit. In this case, the PWMH and PWML status will be forced to a fixed high or low level determined by the PWMHOPS and PWMLOPS bits respectively.

Analog to Digital Converter

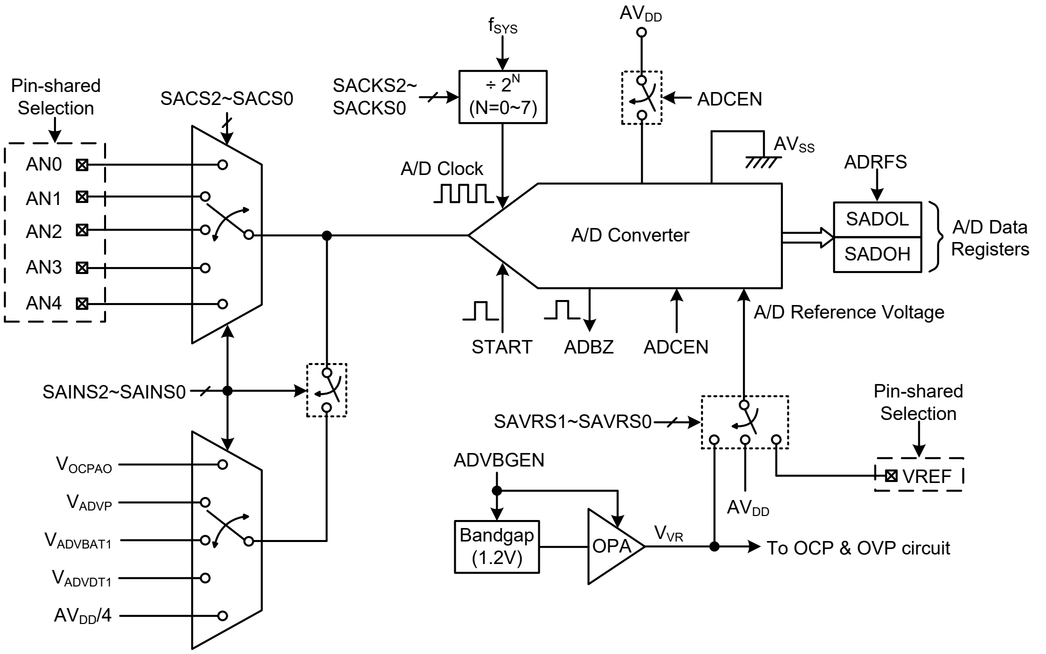
The need to interface to real world analog signals is a common requirement for many electronic systems. However, to properly process these signals by a microcontroller, they must first be converted into digital signals by A/D converters. By integrating the A/D conversion electronic circuitry into the microcontroller, the need for external components is reduced significantly with the corresponding follow-on benefits of lower costs and reduced component space requirements.

A/D Converter Overview

The device contains a multi-channel analog to digital converter which can directly interface to external analog signals, such as that from sensors or other control signals and convert these signals directly into a 12-bit digital value. It also can convert the internal signals, such as the OCP OPA output signal, into a 12-bit digital value. The external or internal analog signal to be converted is determined by the SAINS2~SAINS0 bits together with the SACS2~SACS0 bits. More detailed information about the A/D input signal is described in the “A/D Converter Control Registers” and “A/D Converter Input Signals” sections respectively.

External Input Channels	Internal Signals	A/D Channel Selection Bits
5: AN0~AN4	5: V_{OCPAO}, V_{ADVP}, $V_{ADVBAT1}$, V_{ADVDT1}, $AV_{DD}/4$	SAINS2~SAINS0, SACS2~SACS0

The accompanying block diagram shows the overall internal structure of the A/D converter, together with its associated registers.



A/D Converter Structure

Note: 1. When the V_{ADVP} or $V_{ADVBAT1}$ is selected to be converted, the ADVDS1~ADVBATS0 bits in the LDOC register must be set properly to enable the corresponding input signal.

2. When the V_{ADVDT1} is selected to be converted, if the PA1 pin is setup as the OVPI pin function, then the OVPI will be connected together with the internal A/D converter input signal V_{ADVDT1}, at which piont special consideration should be given to the interaction between internal and external circuits.

A/D Converter Register Description

Overall operation of the A/D converter is controlled using a series of registers. A read only register pair exists to store the A/D converter data 12-bit value. The remaining two registers are control registers which setup the operating and control function of the A/D converter.

Register Name	Bit							
	7	6	5	4	3	2	1	0
SADOL (ADRFS=0)	D3	D2	D1	D0	—	—	—	—
SADOL (ADRFS=1)	D7	D6	D5	D4	D3	D2	D1	D0
SADOH (ADRFS=0)	D11	D10	D9	D8	D7	D6	D5	D4
SADOH (ADRFS=1)	—	—	—	—	D11	D10	D9	D8
SADC0	START	ADBZ	ADCEN	ADRFS	ADVBGEN	SACS2	SACS1	SACS0
SADC1	SAINS2	SAINS1	SAINS0	SAVRS1	SAVRS0	SACKS2	SACKS1	SACKS0

A/D Converter Register List

A/D Converter Data Registers – SADOL, SADOH

As the device contains an internal 12-bit A/D converter, it requires two data registers to store the converted value. These are a high byte register, known as SADOH, and a low byte register, known as SADOL. After the conversion process takes place, these registers can be directly read by the microcontroller to obtain the digitised conversion value. As only 12 bits of the 16-bit register space is utilised, the format in which the data is stored is controlled by the ADRFS bit in the SADC0 register as shown in the accompanying table. D0~D11 are the A/D conversion result data bits. Any unused bits will be read as zero. Note that A/D Converter data register contents will be cleared if the A/D converter is disabled.

ADRFS	SADOH								SADOL							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0
1	0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

A/D Converter Data Registers

A/D Converter Control Registers – SADC0, SADC1

To control the function and operation of the A/D converter, two control registers known as SADC0 and SADC1 are provided. These 8-bit registers define functions such as the selection of which analog channel is connected to the internal A/D converter, the digitised data format, the A/D clock source as well as controlling the start function and monitoring the A/D converter busy status, etc. As the device contains only one actual analog to digital converter hardware circuit, each of the external or internal analog signal inputs must be routed to the converter. The SACS2~SACS0 bits in the SADC0 register are used to determine which external channel input is selected to be converted. The SAINS2~SAINS0 bits in the SADC1 register are used to determine that the analog signal to be converted comes from the internal analog signal or external analog channel input.

The relevant pin-shared function selection bits determine which pins on I/O Ports are used as analog inputs for the A/D converter input and which pins are not to be used as the A/D converter input. When the pin is selected to be an A/D input, its original function whether it is an I/O or other pin-shared function will be removed. In addition, any internal pull-high resistor connected to the pin will be automatically removed if the pin is selected to be an A/D converter input.

- **SADC0 Register**

Bit	7	6	5	4	3	2	1	0
Name	START	ADBZ	ADCEN	ADRFS	ADVBGEN	SACS2	SACS1	SACS0
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 **START**: Start the A/D conversion
0→1→0: Start
This bit is used to initiate an A/D conversion process. The bit is normally low but if set high and then cleared low again, the A/D converter will initiate a conversion process.

Bit 6 **ADBZ**: A/D converter busy flag
0: No A/D conversion is in progress
1: A/D conversion is in progress
This read only flag is used to indicate whether the A/D conversion is in progress or not. When the START bit is set from low to high and then to low again, the ADBZ flag will be set to 1 to indicate that the A/D conversion is initiated. The ADBZ flag will be cleared to 0 after the A/D conversion is complete.

Bit 5 **ADCEN**: A/D converter function enable control
0: Disable
1: Enable
This bit controls the A/D internal function. This bit should be set to one to enable the A/D converter. If the bit is cleared to zero, then the A/D converter will be switched off reducing the device power consumption. When the A/D converter function is disabled, the contents of the A/D data register pair known as SADOH and SADOL will be cleared to zero.

Bit 4 **ADRFS**: A/D converter data format selection
0: A/D converter data format → SADOH = D[11:4]; SADOL = D[3:0]
1: A/D converter data format → SADOH = D[11:8]; SADOL = D[7:0]

This bit controls the format of the 12-bit converted A/D value in the two A/D data registers. Details are provided in the A/D data register section.

Bit 3 **ADVBGEN**: A/D converter internal 1.2V bandgap and OPA (Gain=2) enable control
0: Disable
1: Enable

Bit 2~0 **SACS2~SACS0**: A/D converter external analog channel input selection
000: AN0
001: AN1
010: AN2
011: AN3
100: AN4
101~111: Non-existed channel, the input will be floating

- **SADC1 Register**

Bit	7	6	5	4	3	2	1	0
Name	SAINS2	SAINS1	SAINS0	SAVRS1	SAVRS0	SACKS2	SACKS1	SACKS0
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 **SAINS2~SAINS0**: A/D converter input signal selection
000: External input – External analog channel input
001: Internal input – Internal OCP OPA output, V_{OCPAO}
010: Internal input – Battery cell 2 power supply or Charger voltage input divided voltage, V_{ADVP}
011: Internal input – Battery cell 1 power supply divided voltage, $V_{ADVBAT1}$
100: Internal input – Voltage detect 1 input divided voltage, V_{ADVDT1}
101: Internal input – A/D converter power supply divided by 4, $AV_{DD}/4$
110~111: External input – External analog channel input

When the internal analog signal is selected to be converted, the external channel signal input will automatically be switched off regardless of the SACS bit field value. It will prevent the external channel input from being connected together with the internal analog signal.

Bit 4~3 **SAVRS1~SAVRS0**: A/D converter reference voltage selection
00: From internal A/D converter power supply, AV_{DD}
01: From external VREF pin
10: From internal A/D converter OPA output voltage, V_{VR}
11: From internal A/D converter power supply, AV_{DD}

These bits are used to select the A/D converter reference voltage. When the internal reference voltage source is selected, the reference voltage derived from the external VREF pin will automatically be switched off.

Bit 2~0 **SACKS2~SACKS0**: A/D conversion clock source selection
000: f_{SYS}
001: $f_{SYS}/2$
010: $f_{SYS}/4$
011: $f_{SYS}/8$
100: $f_{SYS}/16$
101: $f_{SYS}/32$
110: $f_{SYS}/64$
111: $f_{SYS}/128$

These three bits are used to select the clock source for the A/D converter.

A/D Converter Reference Voltage

The reference voltage supply to the A/D converter can be supplied from the internal A/D converter power supply, AV_{DD}, or from the A/D converter OPA output, V_{VR}, or from an external reference source supplied on pin VREF. The desired selection is made using the SAVRS1~SAVRS0 bits. When the SAVRS1~SAVRS0 bits are set to "00" or "11", the A/D converter reference voltage will come from the AV_{DD}. When the SAVRS1~SAVRS0 bits are set to "10", the A/D converter reference voltage will come from the V_{VR}. Otherwise, if the SAVRS1~SAVRS0 bits are set to "01", the A/D converter reference voltage will come from the VREF pin. As the VREF pin is pin-shared with other functions, when the VREF pin is selected as the reference voltage supply pin, the VREF pin-shared function control bit should be properly configured to disable other pin function. If the internal A/D converter power AV_{DD} or the OPA output V_{VR} is selected as the reference voltage, the external reference input from the VREF pin will automatically be switched off by hardware.

The analog input values must not be allowed to exceed the value of the selected A/D reference voltage.

SAVRS[1:0]	Reference Source	Description
00, 11	AV_{DD}	Internal A/D converter power supply voltage
01	VREF pin	External A/D converter reference voltage pin
10	V_{VR}	Internal A/D converter OPA output voltage

A/D Converter Reference Voltage Selection

A/D Converter Input Signals

All the external A/D converter analog channel input pins are pin-shared with the I/O pins as well as other functions. The corresponding control bits for each A/D converter external input pin in the PxS0 and PxS1 registers determine whether the input pins are setup as A/D converter analog input channel or whether they have other functions. If the pin is setup to be as an A/D converter analog channel input, the original pin functions will be disabled. In this way, pins can be changed under program control to change their function between A/D inputs and other functions. All pull-high resistors, which are setup through register programming, will be automatically disconnected if the pins are setup as A/D inputs. Note that it is not necessary to first setup the A/D pin as an input in the port control register to enable the A/D input as when the pin-shared function control bits enable an A/D input, the status of the port control register will be overridden.

There are five internal analog signals derived from V_{OCPAO}, V_{ADVP}, $V_{ADVBAT1}$, V_{ADVDT1} or $AV_{DD}/4$, which can be connected to the A/D converter as the analog input signal by configuring the SAINS2~SAINS0 bits. If the external channel input is selected to be converted, the SAINS2~SAINS0 bits should be set to "000" or "110~111" and the SACS2~SACS0 bits can determine which external channel is selected. If the internal analog signal is selected to be converted, the external channel signal input will automatically be switched off regardless of the SACS bit field value. It will prevent the external channel input from being connected together with the internal analog signal.

SAINS[2:0]	SACS[2:0]	Input Signals	Description
000, 110~111	000~100	AN0~AN4	External channel analog input
	101~111	—	Non-existed channel, input is floating
001	xxx	V_{OPAO}	OCP OPA output
010	xxx	V_{ADVP}	Battery cell 2 power supply or Charger voltage input divided voltage
011	xxx	$V_{ADVBAT1}$	Battery cell 1 power supply divided voltage
100	xxx	V_{ADVDT1}	Voltage detect 1 input divided voltage
101	xxx	$AV_{DD}/4$	A/D converter power supply divided by 4

A/D Converter Input Signal Selection

A/D Converter Operation

The START bit in the SADC0 register is used to start the AD conversion. When the microcontroller sets this bit from low to high and then low again, an analog to digital conversion cycle will be initiated.

The ADBZ bit in the SADC0 register is used to indicate whether the analog to digital conversion process is in progress or not. This bit will be automatically set to 1 by the microcontroller after an A/D conversion is successfully initiated. When the A/D conversion is complete, the ADBZ will be cleared to 0. In addition, the corresponding A/D interrupt request flag will be set in the interrupt control register, and if the associated interrupts are enabled, an appropriate internal interrupt signal will be generated. This A/D internal interrupt signal will direct the program flow to the associated A/D internal interrupt address for processing. If the A/D internal interrupt is disabled, the microcontroller can poll the ADBZ bit in the SADC0 register to check whether it has been cleared as an alternative method of detecting the end of an A/D conversion cycle.

The clock source for the A/D converter, which originates from the system clock f_{SYS}, can be chosen to be either f_{SYS} or a subdivided version of f_{SYS}. The division ratio value is determined by the SACKS2~SACKS0 bits in the SADC1 register. Although the A/D clock source is determined by the system clock f_{SYS} and by bits SACKS2~SACKS0, there are some limitations on the maximum A/D clock source speed that can be selected. As the recommended range of permissible A/D clock period, t_{ADCK}, is from 0.5μs to 10μs, care must be taken for system clock frequencies. For example, as the system clock operates at a frequency of 8MHz, the SACKS2~SACKS0 bits should not be set to 000, 001 or 111. Doing so will give A/D clock periods that are less or larger than the minimum or maximum A/D clock period which may result in inaccurate A/D conversion values. Refer to the following table for examples, special care must be taken to values marked with an asterisk *, as these values may be beyond the specified A/D clock period range.

f_{SYS}	A/D Clock Period (t_{ADCK})							
	SACKS[2:0] = 000 (f_{SYS})	SACKS[2:0] = 001 ($f_{SYS}/2$)	SACKS[2:0] = 010 ($f_{SYS}/4$)	SACKS[2:0] = 011 ($f_{SYS}/8$)	SACKS[2:0] = 100 ($f_{SYS}/16$)	SACKS[2:0] = 101 ($f_{SYS}/32$)	SACKS[2:0] = 110 ($f_{SYS}/64$)	SACKS[2:0] = 111 ($f_{SYS}/128$)
1MHz	1μs	2μs	4μs	8μs	16μs *	32μs *	64μs *	128μs *
2MHz	500ns	1μs	2μs	4μs	8μs	16μs *	32μs *	64μs *
4MHz	250ns *	500ns	1μs	2μs	4μs	8μs	16μs *	32μs *
8MHz	125ns *	250ns *	500ns	1μs	2μs	4μs	8μs	16μs *
12MHz	83ns *	167ns *	333ns *	667ns	1.33μs	2.67μs	5.33μs	10.67μs *

A/D Clock Period Examples

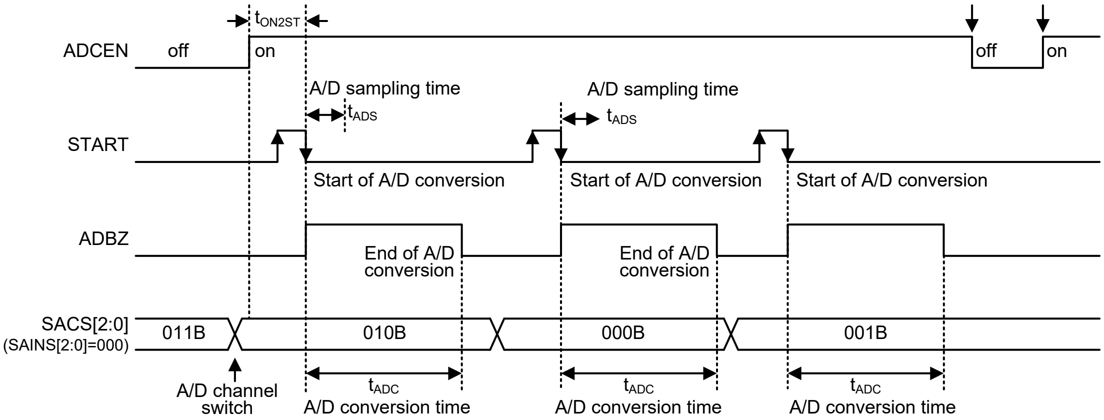
Controlling the power on/off function of the A/D converter circuitry is implemented using the ADCEN bit in the SADC0 register. This bit must be set high to power on the A/D converter. When the ADCEN bit is set high to power on the A/D converter internal circuitry a certain delay, as indicated in the timing diagram, must be allowed before an A/D conversion is initiated. Even if no pins are selected for use as A/D inputs, if the ADCEN bit is high, then some power will still be consumed. In power conscious applications it is therefore recommended that the ADCEN is cleared to zero to reduce power consumption when the A/D converter function is not being used.

A/D Conversion Rate and Timing Diagram

A complete A/D conversion contains two parts, data sampling and data conversion. The data sampling which is defined as t_{ADS} takes 4 A/D clock cycles and the data conversion takes 12 A/D clock cycles. Therefore, a total of 16 A/D clock cycles for an external input A/D conversion which is defined as t_{ADC} are necessary.

$$\text{Maximum single A/D conversion rate} = \text{A/D clock period} \div 16$$

The accompanying diagram shows graphically the various stages involved in an analog to digital conversion process and its associated timing. After an A/D conversion process has been initiated by the application program, the microcontroller internal hardware will begin to carry out the conversion, during which time the program can continue with other functions. The time taken for the A/D conversion is 16 t_{ADCK} clock cycles where t_{ADCK} is equal to the A/D clock period.



A/D Conversion Timing – External Channel Input

Summary of A/D Conversion Steps

The following summarises the individual steps that should be executed in order to implement an A/D conversion process.

- Step 1
  Select the required A/D conversion clock by correctly programming bits SACKS2~SACKS0 in the SADC1 register.
- Step 2
  Enable the A/D converter by setting the ADCEN bit in the SADC0 register to 1.
- Step 3
  Select which signal is to be connected to the internal A/D converter by correctly configuring the SAINS2~SAINS0 bits.
  Select the external channel input to be converted, go to Step 4.
  Select the internal analog signal to be converted, go to Step 5.
- Step 4
  If the A/D input signal comes from the external channel input selected by configuring the SAINS2~SAINS0 bit field, the corresponding pins should be configured as A/D input function by configuring the relevant pin-shared function control bits. The desired analog channel then should be selected by configuring the SACS2~SACS0 bits. After this step, go to Step 6.
- Step 5
  If the SAINS2~SAINS0 bits are set to "001"~"101", the relevant internal analog signal will be selected. When the internal analog signal is selected to be converted, the external channel analog input will automatically be disconnected. After this step, go to Step 6.

- Step 6
  Select the reference voltage source by configuring the SAVRS1~SAVRS0 bits in the SADC1 register. If the internal reference voltage source is selected, the external reference voltage supplied on the VREF pin will be automatically disconnected. Set the ADVBGEN bit high if the OPA output voltage, V_{VR}, is selected as the A/D converter reference voltage.
- Step 7
  Select A/D converter output data format by setting the ADRFS bit in the SADC0 register.
- Step 8
  If A/D conversion interrupt is used, the interrupt control registers must be correctly configured to ensure the A/D interrupt function is active. The master interrupt control bit, EMI, and the A/D conversion interrupt control bit, ADE, must both be set high in advance.
- Step 9
  The A/D conversion procedure can now be initialized by setting the START bit from low to high and then low again.
- Step 10
  If A/D conversion is in progress, the ADBZ flag will be set high. After the A/D conversion process is complete, the ADBZ flag will go low and then the output data can be read from SADOH and SADOL registers.

Note: When checking for the end of the conversion process, if the method of polling the ADBZ bit in the SADC0 register is used, the interrupt enable step above can be omitted.

Programming Considerations

During microcontroller operations where the A/D converter is not being used, the A/D internal circuitry can be switched off to reduce power consumption, by clearing bit ADCEN to 0 in the SADC0 register. When this happens, the internal A/D converter circuits will not consume power irrespective of what analog voltage is applied to their input lines. If the A/D converter input lines are used as normal I/O pins, then care must be taken as if the input voltage is not at a valid logic level, then this may lead to some increase in power consumption.

A/D Conversion Function

As the device contains a 12-bit A/D converter, its full-scale converted digitised value is equal to FFFH. Since the full-scale analog input value is equal to the actual A/D converter reference voltage, V_{REF}, this gives a single bit analog input value of V_{REF} divided by 4096.

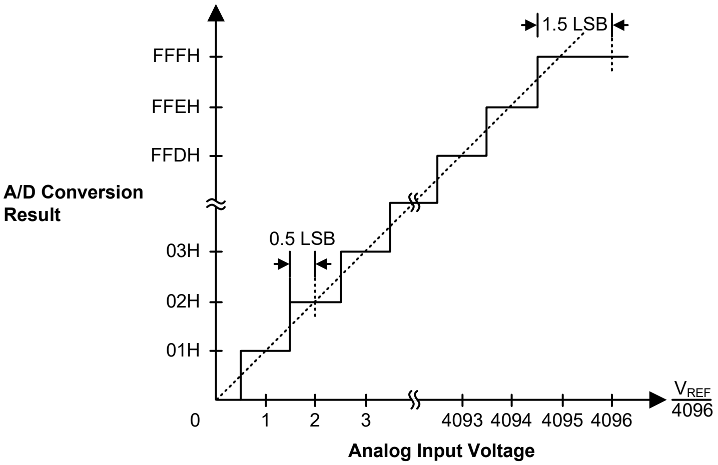
$$1\ \text{LSB} = V_{REF} \div 4096$$

The A/D Converter input voltage value can be calculated using the following equation:

$$\text{A/D input voltage} = \text{A/D output digital value} \times (V_{REF} \div 4096)$$

The diagram shows the ideal transfer function between the analog input value and the digitised output value for the A/D converter. Except for the digitised zero value, the subsequent digitised values will change at a point 0.5 LSB below where they would change without the offset, and the last full scale digitised value will change at a point 1.5 LSB below the V_{REF} level.

Note that here the V_{REF} voltage is the actual A/D converter reference voltage determined by the SAVRS bit field.



Ideal A/D Transfer Function

A/D Conversion Programming Examples

The following two programming examples illustrate how to setup and implement an A/D conversion. In the first example, the method of polling the ADBZ bit in the SADC0 register is used to detect when the conversion cycle is complete, whereas in the second example, the A/D interrupt is used to determine when the conversion is complete.

Example: using an ADBZ polling method to detect the end of conversion

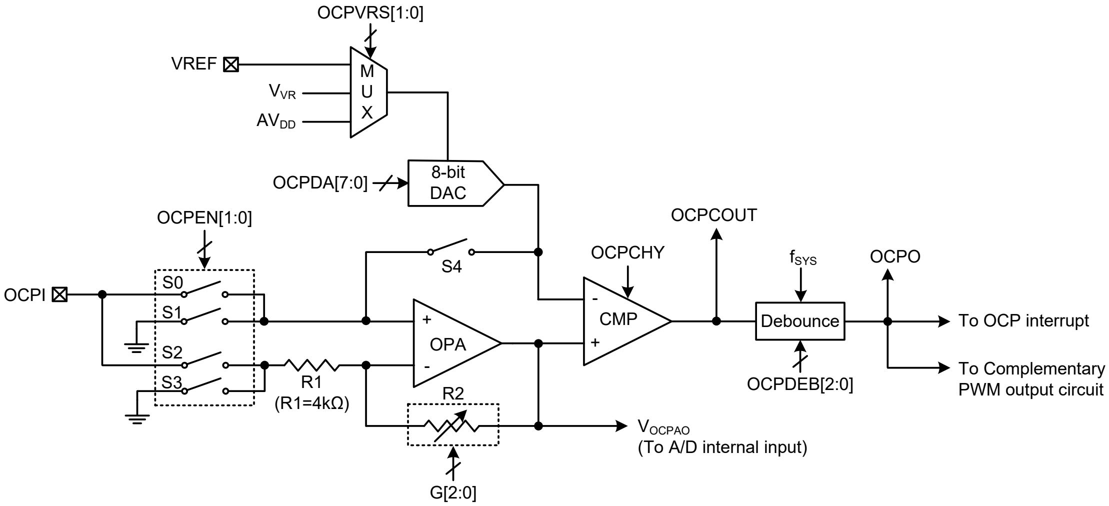
```
clr ADE               ; disable ADC interrupt
mov a, 03H            ; select input signal from external channel input,
mov SADC1, a          ; reference voltage from AVDD and fSYS/8 as A/D clock
mov a, 40h            ; setup PAS1 register to configure pin AN0
mov PAS1, a
mov a, 20h
mov SADC0, a          ; enable A/D converter and connect AN0 channel to A/D converter
:
:
start_conversion:
clr START             ; high pulse on start bit to initiate conversion
set START             ; reset A/D
clr START             ; start A/D
polling_EOC:
sz  ADBZ              ; poll the SADC0 register ADBZ bit to detect end of A/D conversion
jmp polling_EOC       ; continue polling
mov a, SADOL          ; read low byte conversion result value
mov SADOL_buffer, a   ; save result to user defined register
mov a, SADOH          ; read high byte conversion result value
mov SADOH_buffer, a   ; save result to user defined register
:
:
jmp start_conversion  ; start next A/D conversion
```

Example: using the interrupt method to detect the end of conversion

```
clr ADE                ; disable ADC interrupt
mov a, 03H             ; select input signal from external channel input,
mov SADC1, a           ; reference voltage from AVDD and fSYS/8 as A/D clock
mov a, 40h             ; setup PAS1 register to configure pin AN0
mov PAS1, a
mov a, 20h
mov SADC0, a           ; enable A/D converter and connect AN0 channel to A/D converter
:
:
start_conversion:
clr START              ; high pulse on START bit to initiate conversion
set START              ; reset A/D
clr START              ; start A/D
clr ADF                ; clear ADC interrupt request flag
set ADE                ; enable ADC interrupt
set EMI                ; enable global interrupt
:
:
; ADC interrupt service routine
ADC_ISR:
mov acc_stack, a       ; save ACC to user defined memory
mov a, STATUS
mov status_stack, a    ; save STATUS to user defined memory
:
:
mov a, SADOL           ; read low byte conversion result value
mov SADOL_buffer, a    ; save result to user defined register
mov a, SADOH           ; read high byte conversion result value
mov SADOH_buffer, a    ; save result to user defined register
:
:
EXIT_INT_ISR:
mov a, status_stack
mov STATUS, a          ; restore STATUS from user defined memory
mov a, acc_stack       ; restore ACC from user defined memory
reti
```

Over Current Protection – OCP

The device includes an over current protection function which provides a protection mechanism for applications. To prevent the battery charge or load current from exceeding a specific level, the current on the OCPI pin is converted to a relevant voltage level according to the current value using the OCP operational amplifier. It is then compared with a reference voltage generated by an 8-bit D/A converter. When an over current event occurs, an OCP interrupt will be generated if the corresponding interrupt control bit is enabled.



Note: The V_{VR} is from the A/D converter OPA output and the V_{OCPAO} can be selected as the A/D converter input signal.

Over Current Protection Circuit

Over Current Protection Operation

The illustrated OCP circuit is used to prevent the input current from exceeding a reference level. The current on the OCPI pin is converted to a voltage and then amplified by the OCP operational amplifier with a programmable gain from 1 to 50 selected by the G2~G0 bits in the OCPC1 register. This is known as a Programmable Gain Amplifier or PGA. This PGA can also be configured to operate in the non-invert, invert or input offset calibration mode determined by the OCPEN1 and OCPEN0 bits in the OCPC0 register. After the current is converted and amplified to a specific voltage level, it will be compared with a reference voltage provided by an 8-bit D/A converter. The 8-bit D/A converter reference voltage can be supplied from the internal power supply voltage, AV_{DD}, or A/D converter internal operational amplifer output voltage, V_{VR}, or from an external reference source supplied on pin VREF, selected by the OCPVRS[1:0] bits in the OCPC0 register. The comparator output, OCPCOUT, will first be filtered with a certain de-bounce time period selected by the OCPDEB2~OCPDEB0 bits in the OCPC1 register. Then a filtered OCP digital comparator output, OCPO, is obtained to indicate whether an over current condition occurs or not. The OCPO bit will be set to 1 if an over current condition occurs. Otherwise, the OCPO bit is zero. Once an over current event occurs, i.e., the converted voltage of the OCP input current is greater than the reference voltage, the corresponding interrupt will be generated if the relevant interrupt control bit is enabled.

Over Current Protection Registers

Overall operation of the over current protection is controlled using several registers. The OCPDA register is used to provide the reference voltages for the over current protection circuit. The OCPOCAL and OCPCCAL registers are used to cancel out the operational amplifier and comparator input offset. The OCPC0 and OCPC1 registers are control registers which control the OCP function, D/A converter reference voltage selection, PGA gain selection, comparator de-bounce time together with the hysteresis function.

Register Name	Bit							
	7	6	5	4	3	2	1	0
OCPC0	OCPEN1	OCPEN0	OCPVRS1	OCPVRS0	OCPCHY	—	—	OCPO
OCPC1	—	—	G2	G1	G0	OCPDEB2	OCPDEB1	OCPDEB0
OCPDA	D7	D6	D5	D4	D3	D2	D1	D0
OCPOCAL	OCPOOFM	OCPORSP	OCPOOF5	OCPOOF4	OCPOOF3	OCPOOF2	OCPOOF1	OCPOOF0
OCPCCAL	OCPCOUT	OCPCOFM	OCPCRSP	OCPCOF4	OCPCOF3	OCPCOF2	OCPCOF1	OCPCOF0

OCP Register List

• OCPC0 Register

Bit	7	6	5	4	3	2	1	0
Name	OCPEN1	OCPEN0	OCPVRS1	OCPVRS0	OCPCHY	—	—	OCPO
R/W	R/W	R/W	R/W	R/W	R/W	—	—	R
POR	0	0	0	0	0	—	—	0

Bit 7~6 **OCPEN1~OCPEN0**: OCP function operating mode selection
00: OCP function is disabled; S1and S3 on, S0 and S2 off
01: Non-invert mode; S0 and S3 on, S1 and S2 off
10: Invert mode; S1 and S2 on, S0 and S3 off
11: Calibration mode; S1 and S3 on, S0 and S2 off

Bit 5~4 **OCPVRS1~OCPVRS0**: OCP D/A converter reference voltage selection
00/11: From AV_{DD}
01: From external VREF pin
10: From V_{VR}

When setting these bits to “10” to select the V_{VR} as the OCP D/A converter reference voltage, care must be taken that as the V_{VR} signal is from the A/D converter OPA output, so the OPA must first be enabled by setting the ADVBGEN bit high.

Bit 3 **OCPCHY**: OCP comparator hysteresis function control
0: Disable
1: Enable

Bit 2~1 Unimplemented, read as “0”

Bit 0 **OCPO**: OCP digital output bit
0: No over current condition occurs in the monitored source current
1: Over current condition occurs in the monitored source current

• OCPC1 Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	G2	G1	G0	OCPDEB2	OCPDEB1	OCPDEB0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

Bit 7~6 Unimplemented, read as “0”

Bit 5~3 **G2~G0**: R2/R1 ratio selection
000: Unity gain buffer (non-invert mode) or R2/R1=1(invert mode)
001: R2/R1=5
010: R2/R1=10
011: R2/R1=15
100: R2/R1=20
101: R2/R1=30
110: R2/R1=40
111: R2/R1=50

These bits are used to select the R2/R1 ratio to obtain various gain values for invert and non-invert mode. The calculating formula of the OCP PGA gain for the invert and non-invert mode is described in the "Input Voltage Range" section.

Bit 2~0 **OCPDEB2~OCPDEB0**: OCP output filter debounce time selection
000: Bypass, without debounce
001: (1~2) × t_{DEB}
010: (3~4) × t_{DEB}
011: (7~8) × t_{DEB}
100: (15~16) × t_{DEB}
101: (31~32) × t_{DEB}
110: (63~64) × t_{DEB}
111: (127~128) × t_{DEB}

Note: t_{DEB}=1/f_{SYS}.

• OCPDA Register

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
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 7~0 **D7~D0**: OCP D/A converter output voltage control bits
OCP D/A converter output V_{OUT}=(D/A converter reference voltage/256)×D[7:0]

• OCPOCAL Register

Bit	7	6	5	4	3	2	1	0
Name	OCPOOFM	OCPORSP	OCPOOF5	OCPOOF4	OCPOOF3	OCPOOF2	OCPOOF1	OCPOOF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	1	0	0	0	0	0

Bit 7 **OCPOOFM**: OCP operational amplifier operating mode selection
0: Normal operation mode
1: Input Offset Calibration Mode

This bit is used to control the OCP operational amplifier input offset calibration function. The OCPEN1 and OCPEN0 bits must first be set to "11" and then the OCPOOFM bit must be set to 1 followed by the OCPCOFM bit being cleared to 0, then the operational amplifier input offset calibration mode will be enabled. Refer to the "Operational Amplifier Input Offset Calibration" section for the detailed offset calibration procedures.

Bit 6 **OCPORSP**: OCP operational amplifier input offset voltage calibration reference selection
0: Select negative input as the reference input
1: Select positive input as the reference input

Bit 5~0 **OCPOOF5~OCPOOF0**: OCP operational amplifier input offset voltage calibration value
This 6-bit field is used to perform the operational amplifier input offset calibration operation and the value for the OCP operational amplifier input offset calibration can be restored into this bit field. More detailed information is described in the "Operational Amplifier Input Offset Calibration" section.

- **OCPCCAL Register**

Bit	7	6	5	4	3	2	1	0
Name	OCPCOUT	OCPCOFM	OCPCRSP	OCPCOF4	OCPCOF3	OCPCOF2	OCPCOF1	OCPCOF0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	1	0	0	0	0

Bit 7 **OCPOUT**: OCP comparator output bit, positive logic (read only)
0: Positive input voltage < Negative input voltage
1: Positive input voltage > Negative input voltage

This bit is used to indicate whether the positive input voltage is greater than the negative input voltage when the OCP operates in the input offset calibration mode. If the OCPCOUT is set to 1, the positive input voltage is greater than the negative input voltage. Otherwise, the positive input voltage is less than the negative input voltage.

Bit 6 **OCPCOFM**: OCP comparator operating mode selection
0: Normal operation
1: Input Offset Calibration Mode

This bit is used to control the OCP comparator input offset calibration function. The OCPEN1 and OCPEN0 bits must first be set to “11” and then the OCPCOFM bit must be set to 1 followed by the OCPOOFM bit being cleared to 0, then the comparator input offset calibration mode will be enabled. Refer to the “Comparator Input Offset Calibration” section for the detailed offset calibration procedures.

Bit 5 **OCPCRSP**: OCP comparator input offset calibration reference input selection
0: Select negative input as the reference input
1: Select positive input as the reference input

Bit 4~0 **OCPCOF4~OCPCOF0**: OCP comparator input offset calibration value

This 5-bit field is used to perform the comparator input offset calibration operation and the value for the OCP comparator input offset calibration can be restored into this bit field. More detailed information is described in the “Comparator Input Offset Calibration” section.

Input Voltage Range

Together with different PGA operating modes, the input voltage on the OCPI pin can be positive or negative for flexible operation. The PGA output for the positive or negative input voltage is calculated based on different formulas and described by the following.

- For input voltages $V_{IN} > 0$, the PGA operates in the non-invert mode and the PGA output is obtained using the formula below:

$$V_{OUT} = (1+R2/R1) \times V_{IN}$$

- When the PGA operates in the non-invert mode by setting the OCPEN[1:0] to “01” with unity gain select by setting the G[2:0] to “000”, the PGA will act as a unit-gain buffer whose output is equal to V_{IN}.

$$V_{OUT} = V_{IN}$$

- For input voltages $0 > V_{IN} > -0.2V$, the PGA operates in the invert mode and the PGA output is obtained using the formula below. Note that if the input voltage is negative, it cannot be lower than -0.2V which will result in current leakage.

$$V_{OUT} = -(R2/R1) \times V_{IN}$$

OCP OPA and Comparator Offset Calibration

The OCP circuit has four operating modes controlled by OCPEN[1:0], one of them is calibration mode. In calibration mode, Operational amplifier and comparator offset can be calibrated. The procedures and settings of the operational amplifier and comparator input offset calibration are shown as follows.

Operational Amplifier Input Offset Calibration

Step 1. Set OCPEN[1:0]=11, OCPOOFM=1, OCPCOFM=0 and OCPORSP=1, the OCP will operate in the operational amplifier input offset calibration mode. In this mode operation, the S4 is off, the OPA output to the OCPCOUT will bypass the comparator.

Step 2. Set OCPOOF[5:0]=000000 and then read the OCPCOUT bit.

Step 3. Increase the OCPOOF[5:0] value by 1 and then read the OCPCOUT bit.

If the OCPCOUT bit state has not changed, then repeat Step 3 until the OCPCOUT bit state has changed.

If the OCPCOUT bit state has changed, record the OCPOOF value as V_{OOS1} and then go to Step 4.

Step 4. Set OCPOOF[5:0]=111111 and read the OCPCOUT bit.

Step 5. Decrease the OCPOOF[5:0] value by 1 and then read the OCPCOUT bit.

If the OCPCOUT bit state has not changed, then repeat Step 5 until the OCPCOUT bit state has changed.

If the OCPCOUT bit state has changed, record the OCPOOF value as V_{OOS2} and then go to Step 6.

Step 6. Restore the operational amplifier input offset calibration value V_{OOS} into the OCPOOF[5:0] bit field. The offset Calibration procedure is now finished.

Where $V_{OOS} = (V_{OOS1}+V_{OOS2})/2$

Comparator Input Offset Calibration

Step 1. Set OCPEN[1:0]=11, OCPCOFM=1 and OCPOOFM=0, the OCP is now in the comparator input offset calibration mode in which the S4 is on and the D/A converter is off (S4 is used only for comparator calibration mode, in other operation modes, it is off).

Step 2. Set OCPCOF[4:0]=00000 and read the OCPCOUT bit.

Step 3. Increase the OCPCOF[4:0] value by 1 and then read the OCPCOUT bit.

If the OCPCOUT bit state has not changed, then repeat Step 3 until the OCPCOUT bit state has changed.

If the OCPCOUT bit state has changed, record the OCPCOF value as V_{COS1} and then go to Step 4.

Step 4. Set OCPCOF[4:0]=11111 and then read the OCPCOUT bit.

Step 5. Decrease the OCPCOF[4:0] value by 1 and then read the OCPCOUT bit.

If the OCPCOUT bit state has not changed, then repeat Step 5 until the OCPCOUT bit state has changed.

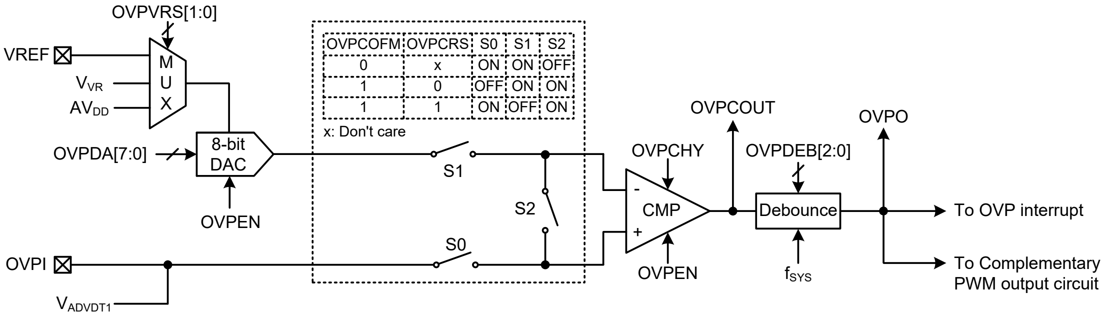
If the OCPCOUT bit state has changed, record the OCPCOF value as V_{COS2} and then go to Step 6.

Step 6. Restore the comparator input offset calibration value V_{COS} into the OCPCOF[4:0] bit field. The offset Calibration procedure is now finished.

Where $V_{COS} = (V_{COS1}+V_{COS2})/2$

Over Voltage Protection – OVP

The device includes an over voltage protection function which provides a protection mechanism for applications. To prevent the operating voltage from exceeding a specific level, the voltage on the OVPI pin is compared with a reference voltage generated by an 8-bit D/A converter. When an over voltage event occurs, an OVP interrupt will be generated if the corresponding interrupt control is enabled.



OVPCOFM	OVPCRS	S0	S1	S2
0	x	ON	ON	OFF
1	0	OFF	ON	ON
1	1	ON	OFF	ON

x: Don't care

Over Voltage Protection Circuit

Note: 1. The V_{VR} is from the A/D converter OPA output.

2. As the OVPI pin is connected together with the internal A/D converter input signal V_{ADVDT1}, special consideration should be given to the interaction between internal and external circuits.

Over Voltage Protection Operation

The OVP circuit is used to prevent the input voltage from being in an unexpected level range. The input voltage will be compared with a reference voltage provided by the 8-bit D/A converter. The 8-bit D/A converter reference voltage can be supplied from the internal power supply voltage, AV_{DD}, or A/D converter internal operational amplifer output voltage, V_{VR}, or from an external reference source supplied on pin VREF, selected by the OVPVRS[1:0] bits in the OVPC0 register. The comparator output, OVPCOUT, will first be filtered with a certain de-bounce time period selected by the OVPDEB2~OVPDEB0 bits in the OVPC0 register. Then a filtered OVP digital comparator output, OVPO, is obtained to indicate whether a user-defined voltage condition occurs or not. The OVPO bit will be set to 1 if an over voltage condition occurs. Otherwise, the OVPO bit is zero. Once an over voltage event occurs, i.e., the input voltage on the OVPI pin is greater than the reference voltage, the corresponding interrupt will be generated if the relevant interrupt control bit is enabled. The comparator in the OVP circuit also has hysteresis function controlled by OVPCHY bit.

Over Voltage Protection Registers

Overall operation of the OVP function is controlled using several registers. The OVPC0 control register is used to control the OVP function, switches on/off control, D/A converter reference voltage selection, comparator de-bounce time together hysteresis function, etc. The OVPC1 register is used to cancel out the comparator input offset. The OVPDA register is used to provide the reference voltage for the OVP circuit.

Register Name	Bit							
	7	6	5	4	3	2	1	0
OVPC0	OVPO	OVPEN	OVPCHY	OVPVRS1	OVPVRS0	OVPDEB2	OVPDEB1	OVPDEB0
OVPC1	OVPCOUT	OVPCOFM	OVPCRS	OVPCOF4	OVPCOF3	OVPCOF2	OVPCOF1	OVPCOF0
OVPDA	D7	D6	D5	D4	D3	D2	D1	D0

OVP Register List

- **OVPC0 Register**

Bit	7	6	5	4	3	2	1	0
Name	OVPO	OVPEN	OVPCHY	OVPVRS1	OVPVRS0	OVPDEB2	OVPDEB1	OVPDEB0
R/W	R	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 **OVPO**: OVP comparator output bit after debounce
0: Positive input voltage < negative input voltage
1: Positive input voltage > negative input voltage

Bit 6 **OVPEN**: OVP function enable control
0: Disable
1: Enable

If the OVPEN bit is cleared to 0, the over voltage protection function is disabled and no power will be consumed. This results in the comparator and D/A converter of OVP both being switched off.

Bit 5 **OVPCHY**: OVP comparator hysteresis function control
0: Disable
1: Enable

Bit 4~3 **OVPVRS1~OVPVRS0**: OVP D/A converter reference voltage selection
00/11: From AV_{DD}
01: From external VREF pin
10: From V_{VR}

When setting these bits to “10” to select the V_{VR} as the OVP D/A converter reference voltage, care must be taken that as the V_{VR} signal is from the A/D converter OPA output, so the OPA must first be enabled by setting the ADVBGEN bit high.

Bit 2~0 **OVPDEB2~OVPDEB0**: OVP comparator output debounce time selection
000: Bypass, without debounce
001: (1~2) × t_{DEB}
010: (3~4) × t_{DEB}
011: (7~8) × t_{DEB}
100: (15~16) × t_{DEB}
101: (31~32) × t_{DEB}
110: (63~64) × t_{DEB}
111: (127~128) × t_{DEB}

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

- **OVPC1 Register**

Bit	7	6	5	4	3	2	1	0
Name	OVPCOUT	OVPCOFM	OVPCRS	OVPCOF4	OVPCOF3	OVPCOF2	OVPCOF1	OVPCOF0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	1	0	0	0	0

Bit 7 **OVPCOUT**: OVP comparator output bit
0: Positive input voltage < negative input voltage
1: Positive input voltage > negative input voltage

This bit is used to indicate whether the positive input voltage is greater than the negative input voltage when the OVP operates in the input offset calibration mode. If the OVPCOUT is set to 1, the positive input voltage is greater than the negative input voltage. Otherwise, the positive input voltage is less than the negative input voltage.

Bit 6 **OVPCOFM**: OVP comparator operating mode selection
0: Normal operating mode
1: Input offset voltage calibration mode

This bit is used to select the OVP comparator operating mode. To select the comparator input offset voltage calibration mode, the OVPCOFM bit must be set to 1. Refer to the "Comparator Input Offset Calibration" section for the detailed offset calibration procedures.

Bit 5 **OVPCRS**: OVP comparator input offset voltage calibration reference selection
0: Select negative input as the reference input
1: Select positive input as the reference input

Bit 4~0 **OVPCOF4~OVPCOF0**: OVP comparator input offset voltage calibration value

This 5-bit field is used to perform the comparator input offset voltage calibration operation and the value for the OVP comparator input offset voltage calibration can be restored into this bit field. More detailed information is described in the "Comparator Input Offset Calibration" section.

- **OVPDA Register**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
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 7~0 **D7~D0**: OVP D/A converter output voltage control bits
OVP D/A converter output V_{OUT} = (D/A converter reference voltage/256)×D[7:0]

Comparator Input Offset Calibration

Before offset calibration, the hysteresis function should be zero by clearing the OVPCHY bit to 0. As the OVPI is pin-shared with I/O or other pin functions, it should first be configured as the OVP input using the corresponding pin-share function control bits. The procedures and settings of the comparator input offset calibration are shown as follows.

Step 1. Set OVPCOFM=1 and OVPCRS=1, the OVP will now operate in the comparator input offset voltage calibration mode (S0 and S2 on). To make sure the V_{OS} is as minimized as possible after calibration, the input reference voltage in calibration should be the same as input DC operating voltage in normal mode operation.

Step 2. Set OVPCOF[4:0]=00000 then read the OVPCOUT bit.

Step 3. Increase the OVPCOF[4:0] value by 1 and then read the OVPCOUT bit.

If the OVPCOUT bit state has not changed, then repeat Step 3 until the OVPCOUT bit state has changed.

If the OVPCOUT bit state has changed, record the OVPCOF value as V_{OS1} and then go to Step 4.

Step 4. Set OVPCOF[4:0]=11111 and read the OVPCOUT bit.

Step 5. Decrease the OVPCOF[4:0] value by 1 and then read the OVPCOUT bit.

If the OVPCOUT bit state has not changed, then repeat Step 5 until the OVPCOUT bit state has changed.

If the OVPCOUT bit state has changed, record the OVPCOF value as V_{OS2} and then go to Step 6.

Step 6. Restore the comparator input offset voltage calibration value V_{OS} into the OVPCOF[4:0] bit field. The offset calibration procedure is now finished.

Where $V_{OS}=(V_{OS1}+V_{OS2})/2$, if the result is not integral, discard the decimal.

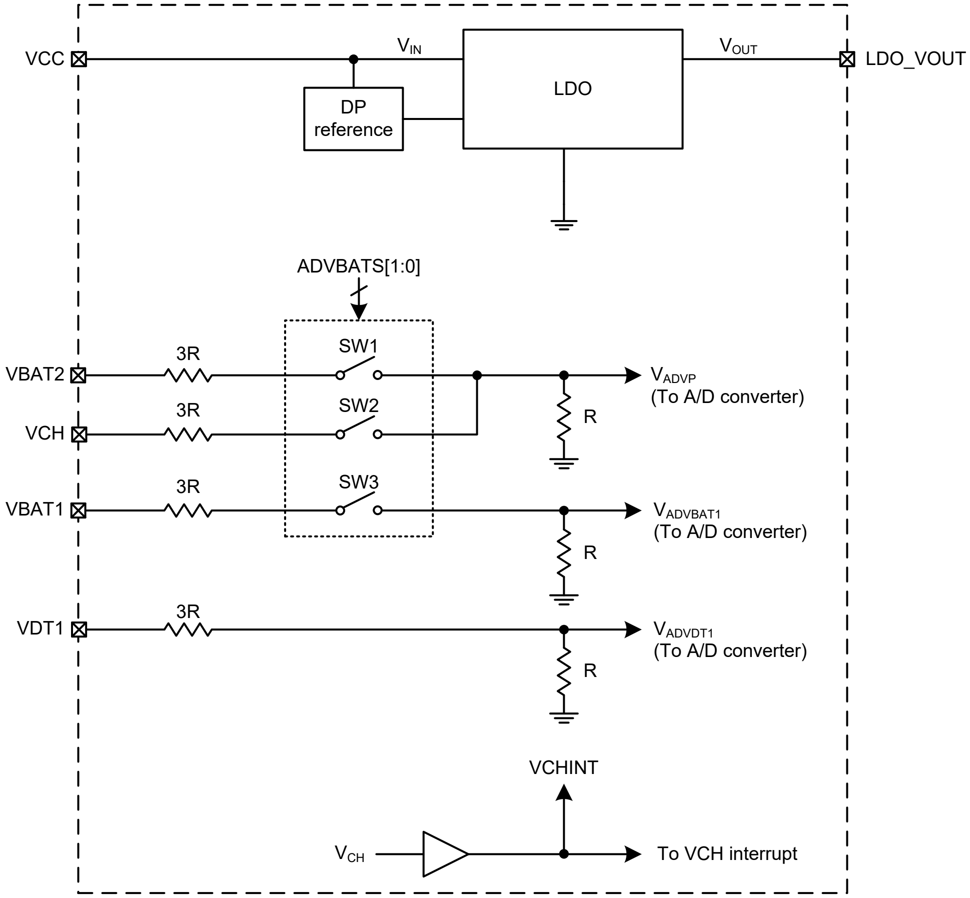
Residue $V_{OS}=V_{OUT}-V_{IN}$.

Low Dropout Regulator – LDO

The device contains an internal Low Dropout Regulator, known as LDO. The LDO regulator can reduce a higher voltage approximately 6V to 12V on input pin VCC to a 5V level supplied on output pin LDO_VOUT. This lower voltage level can provide a fixed power supply for internal or external circuits.

The LDO circuit includes multiple integrated voltage divider resistors. Each input voltage on pins VCH, VDT1, VBAT2 and VBAT1 after voltage division can be measured by connecting it to the corresponding A/D converter internal input channel. The ADVDS1~ADVBATS0 bits in the LDOC register can be used to control the switches SW1~SW3 on/off state to avoid additional power consumption generated by voltage division circuits.

When a voltage is input on the VCH pin, the VCHINT flag will change from 0 to 1 to generate an interrupt signal to wake up the MCU. However, an interrupt may also be triggered on a falling edge because the output is not debounce processed. Therefore, when an interrupt occurs, the VCHINT flag status must be read by the software and then debounce processed to ensure that the interrupt is valid. If the method of polling the VCHINT flag status is used, the debounce processing is also required to ensure that the status is valid.



LDO Block Diagram

- **LDOC Register**

Bit	7	6	5	4	3	2	1	0
Name	VCHINT	—	—	—	—	—	ADVBATS1	ADVBATS0
R/W	R	—	—	—	—	—	R/W	R/W
POR	0	—	—	—	—	—	0	0

Bit 7 **VCHINT**: VCH input voltage status
0: No input voltage
1: Input voltage

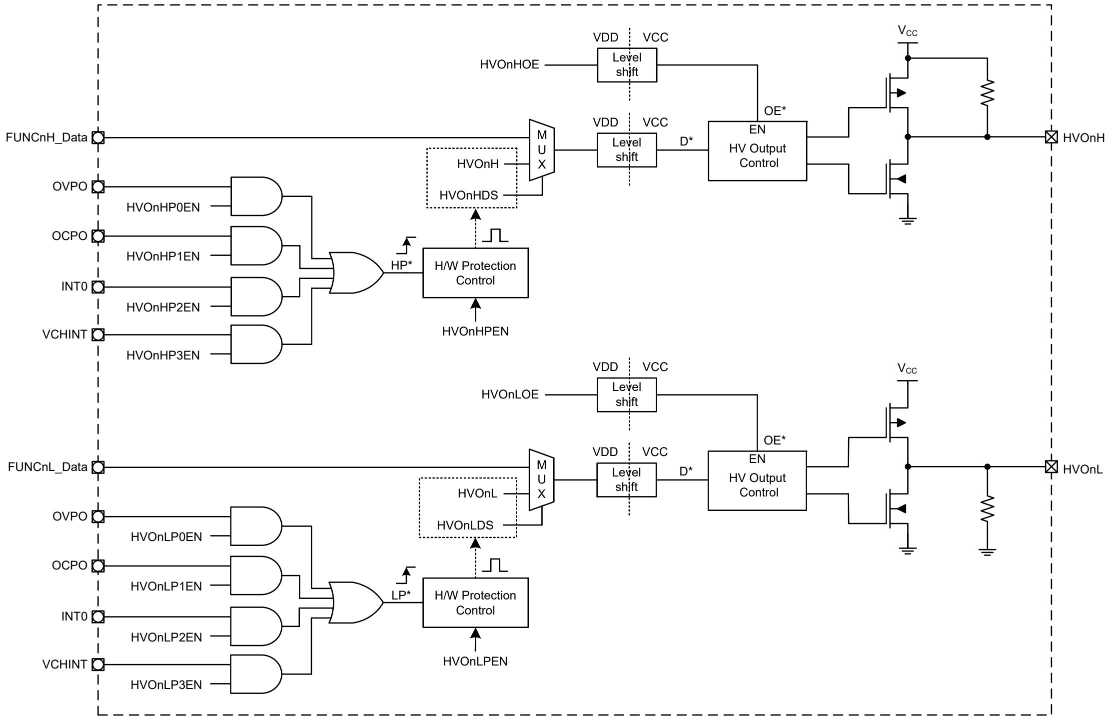
The VCHINT bit only takes effect when the ADVBATS[1:0] bits are set to 10, otherwise this bit is unknown.

Bit 6~2 Unimplemented, read as "0"

Bit 1~0 **ADVBATS1~ADVBATS10**: Switches SW1~SW3 on/off control
00: SW1, SW2, SW3 off
01: SW1 on and SW2, SW3 off
10: SW2 on and SW1, SW3 off
11: SW3 on and SW1, SW2 off

High Voltage Output Ports

The device provides several high voltage output lines, known as HVOnH and HVOnL, where "n" stands for the specific High Voltage Output number. These high voltage ports can be used to drive the external Power MOS.



High Voltage Output Block Diagram (n=0~1)

Note: 1. "*" is the circuit node name and not the Special Function Register bit.
2. FUNCnH_Data is sourced from PWMH for n=0 and from CTP for n=1.
FUNCnL_Data is sourced from PWML for n=0 and from CTPB for n=1.

3. The HVOnH Truth Table is shown below:

HVOnHOE	D	HVOnH
0	0	Pull high
0	1	
1	0	V_{SS}
1	1	V_{CC}

4. The HVOnL Truth Table is shown below:

HVOnLOE	D	HVOnL
0	0	Pull low
0	1	
1	0	V_{SS}
1	1	V_{CC}

Functional Description

The high voltage output on the HVOnH/HVOnL pin is used to directly drive the external power MOS and is controlled by the HVOnH/HVOnL bit high/low or FUNCnH_Data/FUNCnL_Data output, which can be selected by the HVOnHDS/HVOnLDS bit.

Protection Mechanism

There is a protection mechanism provided for high voltage output.

- If HVOnHPEN/HVOnLPEN=0, the HVOnHOE/HVOnLOE bit is used to determine whether the HVOnH/HVOnL pin is pulled high/low or normally outputs (HVOnH/HVOnL data or FUNCnH_Data/FUNCnL_Data).
- If HVOnHPEN/HVOnLPEN=1, when the control signal P trigger from low to high occurs, the HVOnH/HVOnL and HVOnHDS/HVOnLDS bits will be forced to zero by the hardware to implement the purposes as follows.
  - When OE=1, the HVOnH/HVOnL pin output status is low.
  - When OE=0, the HVOnH/HVOnL pin output status is pulled high/low, where the OE value is decided by the HVOnHOE/HVOnLOE bit.

  Then the subsequent action of the HVOnHC/HVOnLC register is determined by the software.
- If HVOnHPEN/HVOnLPEN=1, when the control signal P trigger does not occur, the HVOnH/HVOnL and HVOnHDS/HVOnLDS bits will not be affected.

Control Registers

The overall operation of high voltage output ports is controlled using a series of registers. Each high voltage output port has an identical register, HVOnHC or HVOnLC.

• HVOnHC Register (n=0~1)

Bit	7	6	5	4	3	2	1	0
Name	HVOnHPEN	HVOnHOE	HVOnHDS	HVOnH	HVOnHP3EN	HVOnHP2EN	HVOnHP1EN	HVOnHP0EN
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 **HVOnHPEN**: HVOnH pin hardware protection enable control
0: Disable
1: Enable

Bit 6 **HVOnHOE**: HVOnH pin output enable control
0: Output disable
1: Output enable

Bit 5 **HVOnHDS**: HVOnH pin output data selection
0: Output decided by HVOnH bit
1: Output decided by FUNCnH_Data

Bit 4 **HVOnH**: HVOnH pin output control
0: Output low
1: Output high

Bit 3 **HVOnHP3EN**: HVOnHP3 hardware protection enable control
0: Disable
1: Enable

Bit 2 **HVOnHP2EN**: HVOnHP2 hardware protection enable control
0: Disable
1: Enable

Bit 1 **HVOnHP1EN**: HVOnHP1 hardware protection enable control
0: Disable
1: Enable

Bit 0 **HVOnHP0EN**: HVOnHP0 hardware protection enable control
0: Disable
1: Enable

- **HVOnLC Register (n=0~1)**

Bit	7	6	5	4	3	2	1	0
Name	HVOnLPEN	HVOnLOE	HVOnLDS	HVOnL	HVOnLP3EN	HVOnLP2EN	HVOnLP1EN	HVOnLP0EN
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 **HVOnLPEN**: HVOnL pin hardware protection enable control
0: Disable
1: Enable

Bit 6 **HVOnLOE**: HVOnL pin output enable control
0: Output disable
1: Output enable

Bit 5 **HVOnLDS**: HVOnL pin output data selection
0: Output decided by HVOnL bit
1: Output decided by FUNCnL_Data

Bit 4 **HVOnL**: HVOnL pin output control
0: Output low
1: Output high

Bit 3 **HVOnLP3EN**: HVOnLP3 hardware protection enable control
0: Disable
1: Enable

Bit 2 **HVOnLP2EN**: HVOnLP2 hardware protection enable control
0: Disable
1: Enable

Bit 1 **HVOnLP1EN**: HVOnLP1 hardware protection enable control
0: Disable
1: Enable

Bit 0 **HVOnLP0EN**: HVOnLP0 hardware protection enable control
0: Disable
1: Enable

Low Voltage Detector – LVD

The device has a Low Voltage Detector function, also known as LVD. This enables the device to monitor the power supply voltage, V_{DD}, and provides a warning signal should it fall below a certain level. This function may be especially useful in battery applications where the supply voltage will gradually reduce as the battery ages, as it allows an early warning battery low signal to be generated. The Low Voltage Detector also has the capability of generating an interrupt signal.

LVD Register

The Low Voltage Detector function is controlled using a single register with the name LVDC. Three bits in this register, VLVD2~VLVD0, are used to select one of eight fixed voltages below which a low voltage condition will be determined. A low voltage condition is indicated when the LVDO bit is set. If the LVDO bit is low, this indicates that the V_{DD} voltage is above the preset low voltage value. The LVDEN bit is used to control the overall on/off function of the low voltage detector. Setting the bit high will enable the low voltage detector. Clearing the bit to zero will switch off the internal low voltage detector circuits. As the low voltage detector will consume a certain amount of power, it may be desirable to switch off the circuit when not in use, an important consideration in power sensitive battery powered applications.

- **LVDC Register**

Bit	7	6	5	4	3	2	1	0
Name	—	—	LVDO	LVDEN	VBGEN	VLVD2	VLVD1	VLVD0
R/W	—	—	R	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

Bit 7~6 Unimplemented, read as “0”

Bit 5 **LVDO**: LVD output flag
0: No Low Voltage Detected
1: Low Voltage Detected

Bit 4 **LVDEN**: Low voltage detector enable control
0: Disable
1: Enable

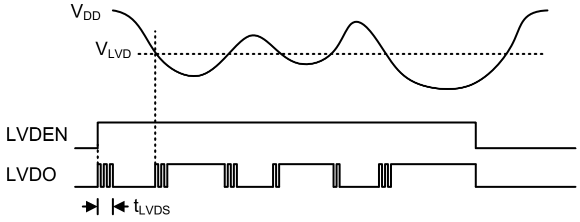
Bit 3 **VBGEN**: Bandgap voltage output enable control
0: Disable
1: Enable

Note that the Bandgap circuit is enabled when the LVD or LVR function is enabled or when the VBGEN bit is set to 1.

Bit 2~0 **VLVD2~VLVD0**: LVD voltage selection
000: 2.0V
001: 2.2V
010: 2.4V
011: 2.7V
100: 3.0V
101: 3.3V
110: 3.6V
111: 4.0V

LVD Operation

The Low Voltage Detector function operates by comparing the power supply voltage, V_{DD}, with a pre-specified voltage level stored in the LVDC register. This has a range of between 2.0V and 4.0V. When the power supply voltage, V_{DD}, falls below this pre-determined value, the LVDO bit will be set high indicating a low power supply voltage condition. When the device enters the SLEEP mode, the low voltage detector will be automatically disabled even if the LVDEN bit is set high. After enabling the Low Voltage Detector, a time delay t_{LVDS} should be allowed for the circuitry to stabilise before reading the LVDO bit. Note also that as the V_{DD} voltage may rise and fall rather slowly, at the voltage nears that of V_{LVD}, there may be multiple bit LVDO transitions.



LVD Operation

The Low Voltage Detector also has its own interrupt, providing an alternative means of low voltage detection, in addition to polling the LVDO bit. The interrupt will only be generated after a delay of t_{LVD} after the LVDO bit has been set high by a low voltage condition. In this case, the LVF interrupt request flag will be set, causing an interrupt to be generated if V_{DD} falls below the preset LVD voltage. This will cause the device to wake-up from the IDLE Mode, however if the Low Voltage Detector wake up function is not required then the LVF flag should be first set high before the device enters the IDLE Mode.

Interrupts

Interrupts are an important part of any microcontroller system. When an external event or an internal function such as a Timer Module or an A/D converter requires microcontroller attention, its corresponding interrupt will enforce a temporary suspension of the main program allowing the microcontroller to direct attention to its needs. The device contains several external interrupt and internal interrupt functions. The external interrupts are generated by the action of the external INT0~INT1 pins, while the internal interrupts are generated by various internal functions such as the TMs, Time Bases, LVD and the A/D converter, etc.

Interrupt Registers

Overall interrupt control, which basically means the setting of request flags when certain microcontroller conditions occur and the setting of interrupt enable bits by the application program, is controlled by a series of registers, located in the Special Purpose Data Memory, as shown in the accompanying table. The number of registers falls into two categories. The first is the INTC0~INTC3 registers which set the primary interrupts, the second is an INTEG register to setup the external interrupts trigger edge type.

Each register contains a number of enable bits to enable or disable individual registers as well as interrupt flags to indicate the presence of an interrupt request. The naming convention of these follows a specific pattern. First is listed an abbreviated interrupt type, then the (optional) number of that interrupt followed by either an "E" for enable/disable bit or "F" for request flag.

Function	Enable Bit	Request Flag	Notes
Global	EMI	—	—
OCP	OCPE	OCPF	—
OVP	OVPE	OVPF	—
INTn Pin	INTnE	INTnF	n=0~1
A/D Converter	ADE	ADF	—
CTM	CTMPE	CTMPF	—
	CTMAE	CTMAF	—
PTM	PTMPE	PTMPF	—
	PTMAE	PTMAF	—
Time Bases	TBnE	TBnF	n=0~1
LVD	LVE	LVF	—
VCH	VCHE	VCHF	—

Interrupt Register Bit Naming Conventions

Register Name	Bit							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	INT0F	OVPF	OCPF	INT0E	OVPE	OCPE	EMI
INTC1	CTMAF	CTMPF	ADF	INT1F	CTMAE	CTMPE	ADE	INT1E
INTC2	TB1F	TB0F	PTMAF	PTMPF	TB1E	TB0E	PTMAE	PTMPE
INTC3	—	—	VCHF	LVF	—	—	VCHE	LVE

Interrupt Register List

• INTEG Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as “0”

Bit 3~2 **INT1S1~INT1S0**: Interrupt edge control for INT1 pin
00: Disable
01: Rising edge
10: Falling edge
11: Rising and falling edges

Bit 1~0 **INT0S1~INT0S0**: Interrupt edge control for INT0 pin
00: Disable
01: Rising edge
10: Falling edge
11: Rising and falling edges

- **INTC0 Register**

Bit	7	6	5	4	3	2	1	0
Name	—	INT0F	OVPF	OCPF	INT0E	OVPE	OCPE	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 **INT0F**: INT0 interrupt request flag
0: No request
1: Interrupt request

Bit 5 **OVPF**: OVP interrupt request flag
0: No request
1: Interrupt request

Bit 4 **OCPF**: OCP interrupt request flag
0: No request
1: Interrupt request

Bit 3 **INT0E**: INT0 interrupt control
0: Disable
1: Enable

Bit 2 **OVPE**: OVP interrupt control
0: Disable
1: Enable

Bit 1 **OCPE**: OCP interrupt control
0: Disable
1: Enable

Bit 0 **EMI**: Global interrupt control
0: Disable
1: Enable

- **INTC1 Register**

Bit	7	6	5	4	3	2	1	0
Name	CTMAF	CTMPF	ADF	INT1F	CTMAE	CTMPE	ADE	INT1E
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 **CTMAF**: CTM comparator A match interrupt request flag
0: No request
1: Interrupt request

Bit 6 **CTMPF**: CTM comparator P match interrupt request flag
0: No request
1: Interrupt request

Bit 5 **ADF**: A/D converter interrupt request flag
0: No request
1: Interrupt request

Bit 4 **INT1F**: INT1 interrupt request flag
0: No request
1: Interrupt request

Bit 3 **CTMAE**: CTM comparator A match interrupt control
0: Disable
1: Enable

Bit 2 **CTMPE**: CTM comparator P match interrupt control
0: Disable
1: Enable

Bit 1 **ADE**: A/D converter interrupt control
0: Disable
1: Enable

Bit 0 **INT1E**: INT1 interrupt control
0: Disable
1: Enable

• INTC2 Register

Bit	7	6	5	4	3	2	1	0
Name	TB1F	TB0F	PTMAF	PTMPF	TB1E	TB0E	PTMAE	PTMPE
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 **TB1F**: Time Base 1 interrupt request flag
0: No request
1: Interrupt request

Bit 6 **TB0F**: Time Base 0 interrupt request flag
0: No request
1: Interrupt request

Bit 5 **PTMAF**: PTM comparator A match interrupt request flag
0: No request
1: Interrupt request

Bit 4 **PTMPF**: PTM comparator P match interrupt request flag
0: No request
1: Interrupt request

Bit 3 **TB1E**: Time Base 1 interrupt control
0: Disable
1: Enable

Bit 2 **TB0E**: Time Base 0 interrupt control
0: Disable
1: Enable

Bit 1 **PTMAE**: PTM comparator A match interrupt control
0: Disable
1: Enable

Bit 0 **PTMPE**: PTM comparator P match interrupt control
0: Disable
1: Enable

• INTC3 Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	VCHF	LVF	—	—	VCHE	LVE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

Bit 7~6 Unimplemented, read as “0”

Bit 5 **VCHF**: VCH interrupt request flag
0: No request
1: Interrupt request

Bit 4 **LVF**: LVD interrupt request flag
0: No request
1: Interrupt request

Bit 3~2 Unimplemented, read as “0”

Bit 1	**VCHE**: VCH interrupt control 0: Disable 1: Enable
Bit 0	**LVE**: LVD interrupt control 0: Disable 1: Enable

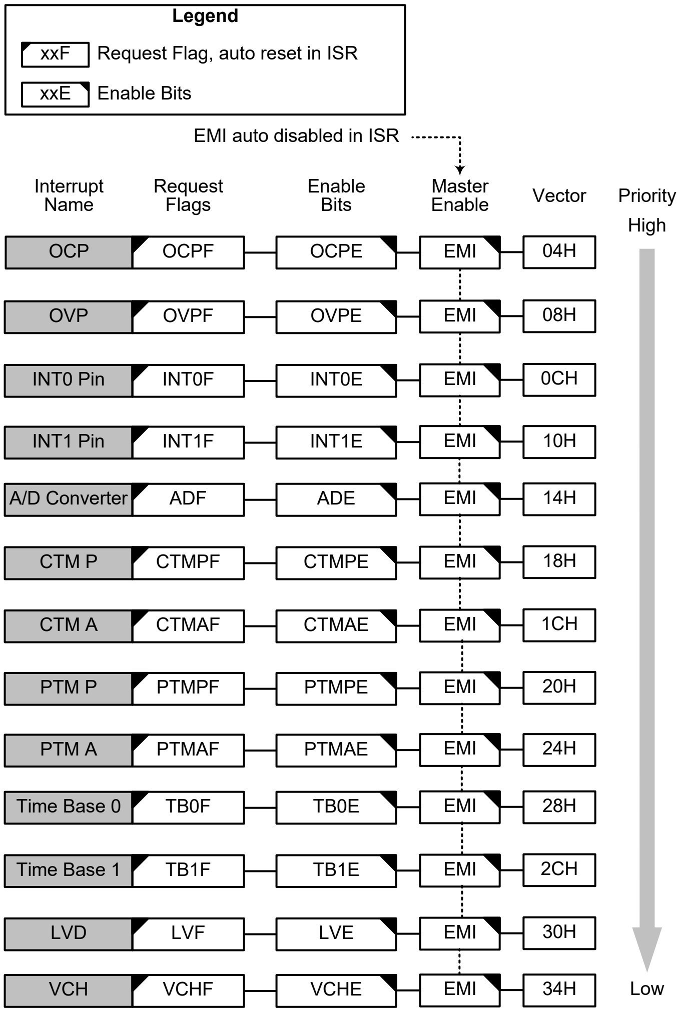
Interrupt Operation

When the conditions for an interrupt event occur, such as a TM Comparator P, Comparator A match, the relevant interrupt request flag will be set. Whether the request flag actually generates a program jump to the relevant interrupt vector is determined by the condition of the interrupt enable bit. If the enable bit is set high then the program will jump to its relevant vector; if the enable bit is zero then although the interrupt request flag is set an actual interrupt will not be generated and the program will not jump to the relevant interrupt vector. The global interrupt enable bit, if cleared to zero, will disable all interrupts.

When an interrupt is generated, 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” 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 an “RETI”, 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 accompanying diagram with their order of priority. All of the interrupt sources have their own individual vector. Once an interrupt subroutine is serviced, all the other interrupts will be blocked, as the global interrupt enable bit, 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. In case of simultaneous requests, the accompanying diagram shows the priority that is applied. All of the interrupt request flags when set will wake-up the device if it is in SLEEP or IDLE Mode, however to prevent a wake-up from occurring the corresponding flag should be set before the device is in SLEEP or IDLE Mode.



Interrupt Structure

Over Current Protection Interrupt

The OCP interrupt is controlled by detecting the OCP input current. An OCP interrupt request will take place when the OCP interrupt request flag, OCPF, is set, which occurs when an over current condition is detected. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and OCP interrupt enable bit, OCPE, must first be set. When the interrupt is enabled, the stack is not full and an over current condition is detected, a subroutine call to the OCP interrupt vector, will take place. When the interrupt is serviced, the OCP interrupt flag, OCPF, will be automatically cleared. The EMI bit will also be automatically cleared to disable other interrupts.

Over Voltage Protection Interrupt

The OVP interrupt is controlled by detecting the OVP input voltage. An OVP interrupt request will take place when the OVP interrupt request flag, OVPF, is set, which occurs when the over voltage protection circuit detects an over voltage condition. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and OVP interrupt enable bit, OVPE, must first be set. When the interrupt is enabled, the stack is not full and an over voltage condition is detected, a subroutine call to the OVP interrupt vector, will take place. When the interrupt is serviced, the OVP interrupt flag, OVPF, will be automatically cleared. The EMI bit will also be automatically cleared to disable other interrupts.

External Interrupts

The external interrupts are controlled by signal transitions on the INTn pins. An external interrupt request will take place when the external interrupt request flag, INTnF, is set, which will occur when a transition, whose type is chosen by the edge select bits, appears on the external interrupt pins. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and the external interrupt enable bit, INTnE, must first be set. Additionally the correct interrupt edge type must be selected using the INTEG register to enable the external interrupt function and to choose the trigger edge type. As the external interrupt pins are pin-shared with I/O pins, they can only be configured as external interrupt pins if their external interrupt enable bits in the corresponding interrupt registers has been set and the external interrupt pins are selected by the corresponding pin-shared function selection bits. The pins must also be set as an input by setting the corresponding bit in the port control register. When the interrupt is enabled, the stack is not full and the correct transition type appears on the external interrupt pins, a subroutine call to the external interrupt vector, will take place. When the interrupt is serviced, the external interrupt request flag, INTnF, will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts. Note that any pull-high resistor selection on the external interrupt pin will remain valid even if the pin is used as an external interrupt input.

The INTEG register is used to select the type of active edge that will trigger the external interrupt. A choice of either rising or falling or both edge types can be chosen to trigger an external interrupt. Note that the INTEG register can also be used to disable the external interrupt function.

A/D Converter Interrupt

An A/D converter interrupt request will take place when the A/D converter interrupt request flag, ADF, is set, which occurs when the A/D conversion process finishes. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and A/D converter interrupt enable bit, ADE, must first be set. When the interrupt is enabled, the stack is not full and the A/D conversion process has ended, a subroutine call to the A/D converter interrupt vector, will take place. When the A/D converter interrupt is serviced, the A/D converter interrupt flag, ADF, will be automatically cleared. The EMI bit will also be automatically cleared to disable other interrupts.

Timer Module Interrupts

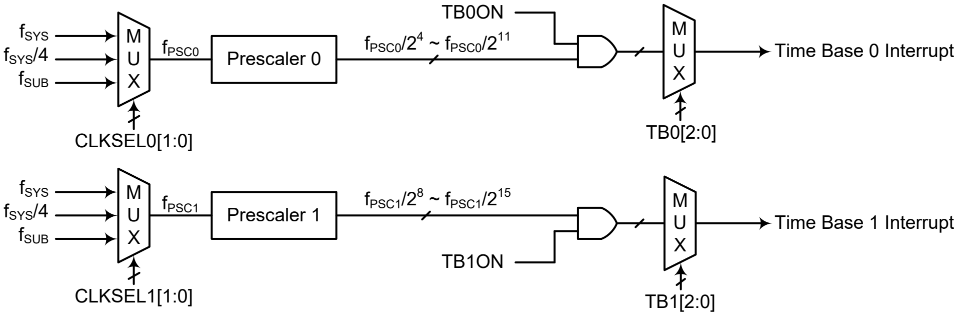
The Compact and Periodic type TMs each has two interrupts, one comes from the comparator A match situation and the other comes from the comparator P match situation. For all of the TM types there are two interrupt request flags and two enable control bits. A TM interrupt request will take place when any of the TM request flags are set, a situation which occurs when a TM comparator P or A match situation happens.

To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and the respective TM interrupt enable bit, must first be set. When the interrupt is enabled, the stack is not full and a TM comparator match situation occurs, a subroutine call to the relevant TM interrupt vector locations, will take place. When the TM interrupt is serviced, the TM interrupt request flags will be automatically cleared, the EMI bit will also be automatically cleared to disable other interrupts.

Time Base Interrupts

The function of the Time Base interrupts is to provide regular time signals in the form of an internal interrupt. It is controlled by the overflow signals from the timer function. When this happens its interrupt request flag TBnF will be set. To allow the program to branch to its interrupt vector address, the global interrupt enable bit, EMI and Time Base enable bit, TBnE, must first be set. When the interrupt is enabled, the stack is not full and the Time Base overflows, a subroutine call to its interrupt vector location will take place. When the interrupt is serviced, the interrupt request flag, TBnF, will be automatically reset and the EMI bit will be cleared to disable other interrupts.

The purpose of the Time Base interrupts is to provide an interrupt signal at fixed time periods. Its clock source, f_{PSCn}, originates from the internal clock source f_{SYS}, $f_{SYS}/4$ or f_{SUB} and then passes through a divider, the division ratio of which is selected by programming the appropriate bits in the TBnC register to obtain longer interrupt periods whose value ranges. The clock source that generates f_{PSCn}, which in turn controls the Time Base interrupt period is selected using the CLKSELn[1:0] bits in the PSCnR register.



Time Base Interrupts

- **PSCnR Register (n=0~1)**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	CLKSELn1	CLKSELn0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as “0”

Bit 1~0 **CLKSELn1~CLKSELn0**: Prescaler n clock source f_{PSCn} selection
00: f_{SYS}
01: $f_{SYS}/4$
1x: f_{SUB}

- **TB0C Register**

Bit	7	6	5	4	3	2	1	0
Name	TB0ON	—	—	—	—	TB02	TB01	TB00
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	0	0	0

Bit 7 **TB0ON**: Time Base 0 control
0: Disable
1: Enable

Bit 6~3 Unimplemented, read as “0”

Bit 2~0 **TB02~TB00**: Time Base 0 time-out period selection
000: $2^4/f_{PSC0}$
001: $2^5/f_{PSC0}$
010: $2^6/f_{PSC0}$
011: $2^7/f_{PSC0}$
100: $2^8/f_{PSC0}$
101: $2^9/f_{PSC0}$
110: $2^{10}/f_{PSC0}$
111: $2^{11}/f_{PSC0}$

- **TB1C Register**

Bit	7	6	5	4	3	2	1	0
Name	TB1ON	—	—	—	—	TB12	TB11	TB10
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	0	0	0

Bit 7 **TB1ON**: Time Base 1 control
0: Disable
1: Enable

Bit 6~3 Unimplemented, read as “0”

Bit 2~0 **TB12~TB10**: Time Base 1 time-out period selection
000: $2^8/f_{PSC1}$
001: $2^9/f_{PSC1}$
010: $2^{10}/f_{PSC1}$
011: $2^{11}/f_{PSC1}$
100: $2^{12}/f_{PSC1}$
101: $2^{13}/f_{PSC1}$
110: $2^{14}/f_{PSC1}$
111: $2^{15}/f_{PSC1}$

LVD Interrupt

An LVD interrupt request will take place when the LVD interrupt request flag, LVF, is set, which occurs when the Low Voltage Detector function detects a low power supply voltage. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and the Low Voltage interrupt enable bit, LVE, must first be set. When the interrupt is enabled, the stack is not full and a low voltage condition occurs, a subroutine call to the LVD interrupt vector, will take place. When the LVD interrupt is serviced, the LVF flag will be automatically cleared. The EMI bit will also be automatically cleared to disable other interrupts.

VCH Interrupt

A VCH interrupt request will take place when the VCH interrupt request flag, VCHF, is set, which occurs when a voltage is input on the VCH pin. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and VCH interrupt enable bit, VCHE, must first be set. When the interrupt is enabled, the stack is not full and a voltage is input on the VCH pin, a subroutine call to the VCH interrupt vector, will take place. When the interrupt is serviced, the VCH interrupt flag, VCHF, will be automatically cleared. The EMI bit will also be automatically cleared to disable other interrupts.

Interrupt Wake-up Function

Each of the interrupt functions has the capability of waking up the microcontroller when in the SLEEP or IDLE Mode. A wake-up is generated when an interrupt request flag changes from low to high and is independent of whether the interrupt is enabled or not. Therefore, even though the device is in the SLEEP or IDLE Mode and its system oscillator stopped, situations such as external edge transitions on the external interrupt pin or a low power supply voltage may cause their respective interrupt flag to be set high and consequently generate an interrupt. Care must therefore be taken if spurious wake-up situations are to be avoided. If an interrupt wake-up function is to be disabled then the corresponding interrupt request flag should be set high before the device enters the SLEEP or IDLE Mode. The interrupt enable bits have no effect on the interrupt wake-up function.

Programming Considerations

By disabling the relevant 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 the application program.

It is recommended that programs do not use the “CALL” instruction within the interrupt service subroutine. Interrupts often occur in an unpredictable manner or need to be serviced immediately. 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.

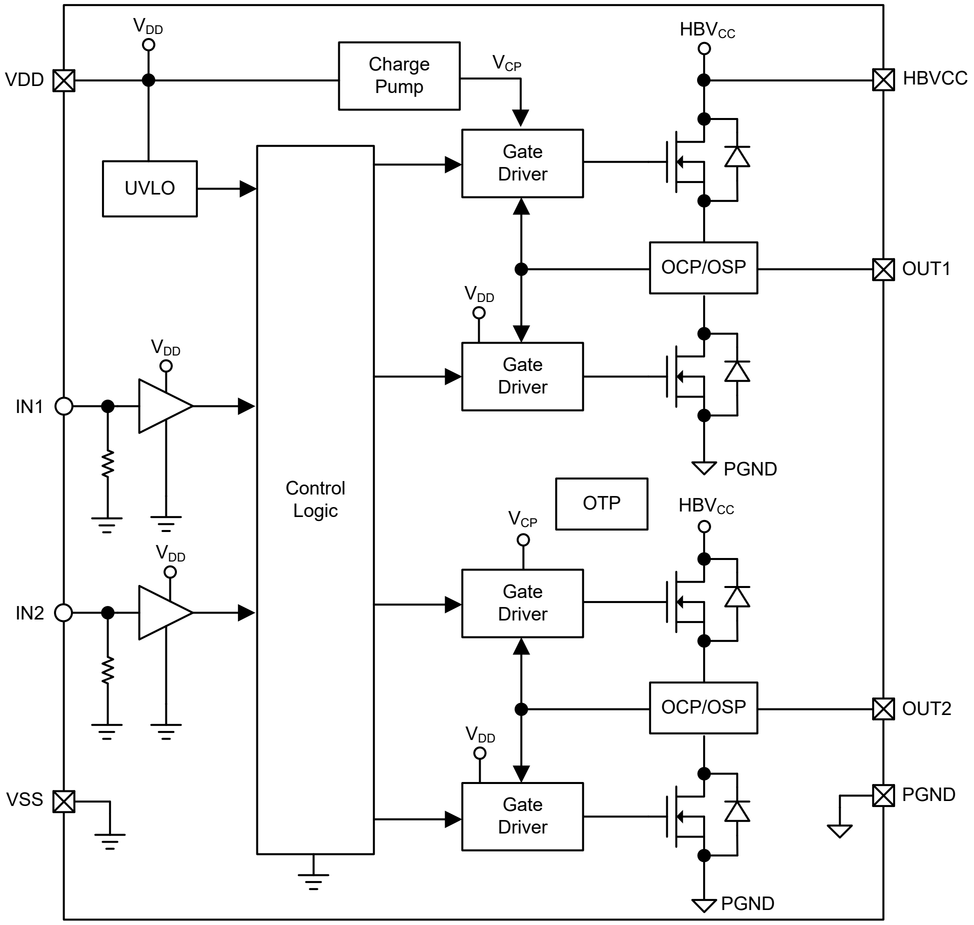
Every interrupt has the capability of waking up the microcontroller when it is in the SLEEP or IDLE Mode, the wake up being generated when the interrupt request flag changes from low to high. If it is required to prevent a certain interrupt from waking up the microcontroller then its respective request flag should be first set high before enter SLEEP or IDLE Mode.

As only the Program Counter is pushed onto the stack, then when the interrupt is serviced, if the contents of the accumulator, status register or other registers are altered by the interrupt service program, their contents should be saved to the memory at the beginning of the interrupt service routine.

To return from an interrupt subroutine, either an RET or RETI instruction may be executed. The RETI instruction in addition to executing a return to the main program also automatically sets the EMI bit high to allow further interrupts. The RET instruction however only executes a return to the main program leaving the EMI bit in its present zero state and therefore disabling the execution of further interrupts.

H-Bridge Driver

The device includes a 1-channel H-bridge driver which can drive DC brush motors or solenoids. Due to the 4 internal very low on-resistance power MOSFETs which have parallel spark killer diodes, the H-bridge motor driver has a high efficiency motor driving capability, reduced external components and outstanding thermal performance. Separate controller and motor power supplies allow for simplified system power domain design. The isolated motor current sensing pin, PGND, is designed to detect the motor current by connecting a resistor from this pin to ground. The H-bridge driver also includes a full range of protection functions including over-current and over-temperature to prevent the possibility of burn-out occurring even if the motor stalls or if the output pins are shorted to each other.



Note: The IN1 and IN2 lines are internally connected to the MCU PB2/PTPB and PB3/PTP lines respectively.

H-Bridge Driver Block Diagram

H-Bridge Control

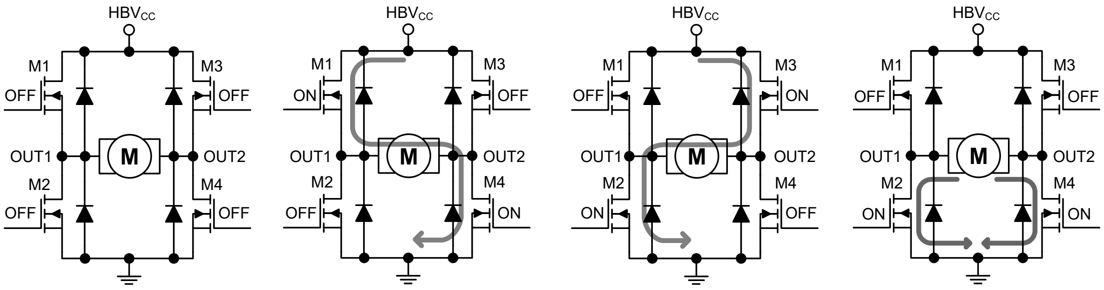
According to the IN1 and IN2 line states the H-bridge driver will generate four H-bridge output states: Standby, Forward, Reverse and Brake. The H-bridge driver operation truth tables in Active Period and in Sleep Period are shown in the following tables. Note that the IN1 and IN2 lines are internally connected to the MCU PB2/PTPB and PB3/PTP lines respectively. The PB2 and PB3 lines, if selected, should be configured as outputs by clearing the relevant I/O port control bits, in order to properly control the H-bridge driver function.

IN1	IN2	OUT1	OUT2	Operation Mode	H-Bridge Status			
					M1	M2	M3	M4
0	0	Z	Z	Standby	OFF	OFF	OFF	OFF
0	1	L	H	Reverse	OFF	ON	ON	OFF
1	0	H	L	Forward	ON	OFF	OFF	ON
1	1	L	L	Brake	OFF	ON	OFF	ON

H-Bridge Driver Operation True Table in Active Period

IN1	IN2	OUT1	OUT2	Operation Mode	H-Bridge Status			
					M1	M2	M3	M4
0	0	Z	Z	Standby	OFF	OFF	OFF	OFF
1	1	L	L	Brake	OFF	ON	OFF	ON

H-Bridge Driver Operation True Table in Sleep Period



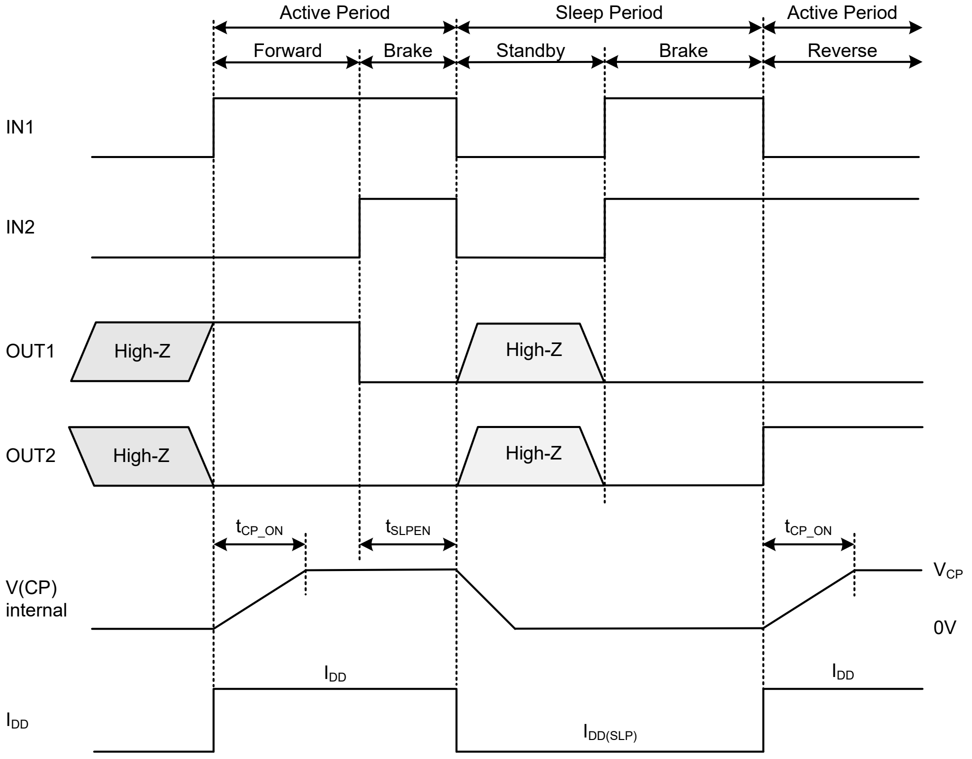
Standby Forward Reverse Brake

H-Bridge Operation Modes

Active Period and Sleep Period

When the Standby or Brake mode continuously exceeds over 10ms, the H-bridge driver will enter the Sleep Period. At this time, the Standby or Brake mode still works in the Sleep Period as shown in the table above. Change the operation mode to Forward or Reverse will go back to the Active Period.

In the Sleep Period, all functional blocks are turned off to reduce the current consumption to an ultra-low value of less than 0.1μA (max). At this time, switching IN1/IN2 to Brake or Standby configuration only affects the output – OUT1/OUT2, the driver remains in the Sleep Period as shown in the following diagram. Since all functional blocks are turned off, the Standby and Brake mode outputs are not protected. When an IN1 or IN2 line is set to “High”, the H-bridge driver will exit from the Sleep Period.



H-Bridge Driver Timing Diagram

V_{DD} Under Voltage Lock-Out

In order to avoid an H-bridge metastable output condition when powered-on or with a low battery voltage, an under voltage lockout function is integrated within the H-bridge driver. During the power-on period, the H-bridge outputs will remain in high impedance states and the control inputs are ignored when V_{DD} is lower than V_{UVLO+}. The H-bridge outputs are only controlled by inputs when V_{DD} is higher than V_{UVLO+}. The H-bridge driver will be locked again when V_{DD} falls to a voltage level lower than V_{UVLO-}

Over Current Protection – OCP

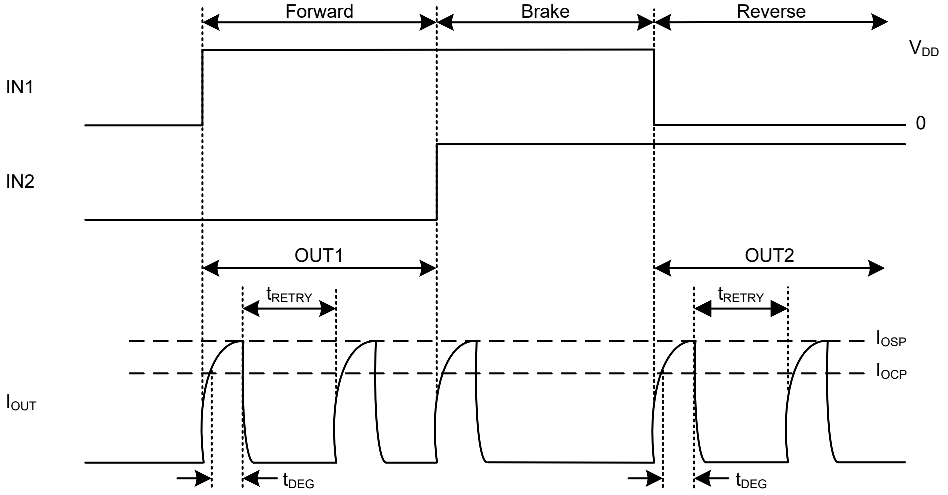
The H-bridge driver includes a fully integrated over current protection function within each of the internal power MOSFETs. When the motor current exceeds the over current protection threshold, I_{OCP}, exceeding a de-glitch time, t_{DEG}, all power MOSFETs will be turned off immediately. After the retry time times out, the H-bridge driver will release the protection activation and allow normal operation to resume. The retry mechanism is only available in Forward and Reverse modes.

Output Short-Circuit Protection – OSP

The H-bridge driver provides full output protection for conditions such as an output pin short to ground, to the motor supply or to each other. The H-bridge driver detects the current through each power MOSFETs and compares it with the output short circuit protection threshold, I_{OSP}, without a de-glitch time. The current threshold I_{OSP} is internally set to 1.5 times the I_{OCP}. When an OSP condition occurs, the H-bridge driver will turn off all power MOSFETs and keep checking the output status every retry time, t_{RETRY}, until the fault is removed. The retry mechanism is only available in Forward and Reverse modes.

Over Temperature Protection – OTP

If the die temperature exceeds the internal limit threshold, T_{SHD}, the H-bridge driver will turn off all power MOSFETs until the temperature decreases to a specific level less than the recovery temperature, T_{REC}.



Retry Mechanism

The retry mechanism entry and release conditions are shown as follows.

Protection Type	Retry Entry Condition	Functional Mode				Retry Release Condition
		Forward/Reverse	Brake	Standby	Sleep	
OCP	I_{OCP}>3.0A	O	—	—	—	I_{OCP}<3.0A
OSP	OUTx-to-ground, OUTx-to-power or OUT1-to-OUT2 path	O	—	—	—	Short circuit fault is removed

Retry Mechanism Conditions

The protection function entry and release conditions are shown as follows.

Protection Type	Protection Entry Condition	Functional Mode				Protection Release Condition
		Forward/Reverse	Brake	Standby	Sleep	
UVLO	V_{IN}<1.8V	—	O	—	—	V_{IN}>2.5V
OCP	I_{OCP}>3.0A	O	O	—	—	I_{OCP}<3.0A
OSP	OUTx-to-ground, OUTx-to-power or OUT1-to-OUT2 path	O	O	—	—	Short circuit fault is removed
OTP	T_J>155°C	O	O	O	—	T_J<120°C

Protection Function Conditions

Motor Current Sensing

The H-bridge driver can be used to implement a motor current sensing function by connecting an external resistor from PGND to ground. The PGND voltage is recommended to be kept lower than 0.5V to avoid turning on the protection diodes on the input pin such as the MCU ADC input. The current sensing resistor, R_S, is also recommended to be less than $0.5V/I_{M(max)}$, where $I_{M(max)}$ stands for the maximum motor current (motor stall current typical).

Power Dissipation

The main power dissipation in the H-bridge driver is determined by the on-resistance of internal power MOSFETs. The average power dissipation can be estimated using the following equation:

$$P_{AVG} = R_{ON}\times(I_{OUT(RMS)})^2$$

Where P_{AVG} is the average power dissipation of the H-bridge driver, R_{ON} is the total on-resistance of HS and LS MOSFETs and $I_{OUT(RMS)}$ is the RMS or DC output current through the load. Note that the R_{ON} value will vary with the die temperature. The higher the die temperature is, the higher will be the R_{ON} value. When the ambient temperature increases or as the H-bridge driver heats up, the power dissipation of the H-bridge driver will also increase.

Component/Motor Selection Guide

Motor Consideration

The appropriate motor voltage depends upon the desired RPM and power supply source. Higher motor voltages also increase the motor current rate. Note that the motor stall current must be less than the internal limit output current, I_{OCP}, to avoid failures when the motor starts up.

Motor Supply Capacitor

It is suggested to use at least a 10μF value capacitor connected between HBVCC and ground. There are two main functions for this capacitor. Firstly, it absorbs the energy released by the motor to reduce any overshoot voltage damage. Secondly, it provides a transient power source to the motor to compensate for the battery response time or for long connecting wire effects when the motor starts up or for fast control switching between forward and reverse modes.

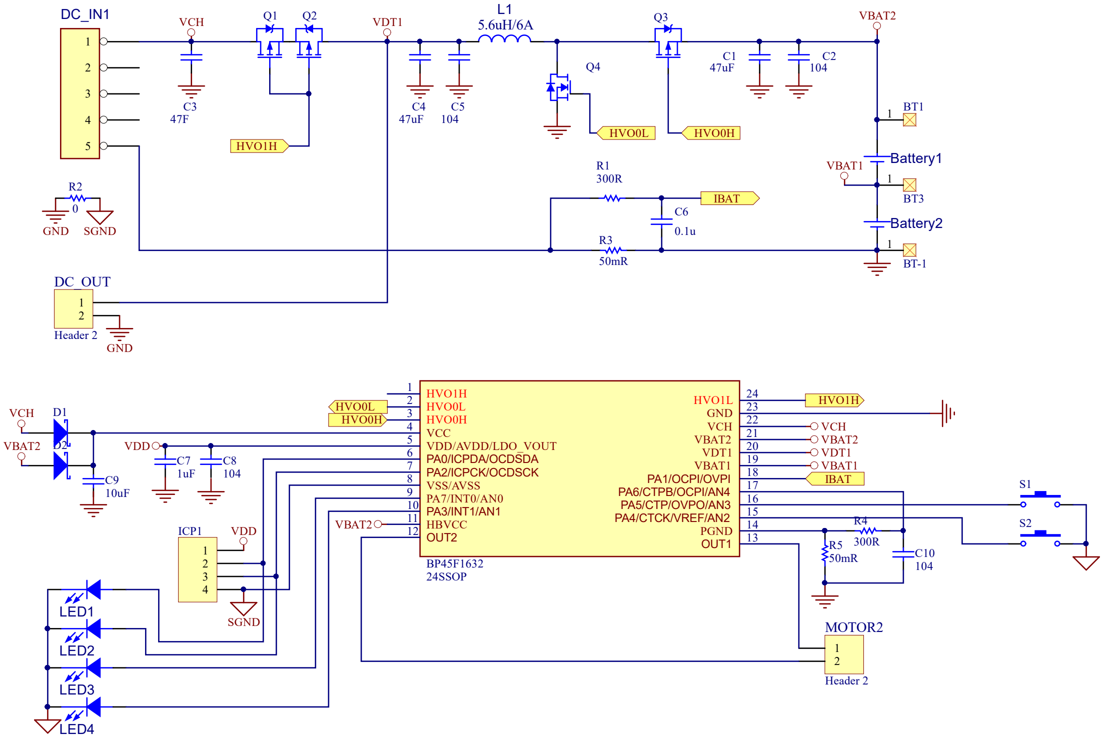
Motor Bypass Capacitor

The motor bypass capacitor connected between OUT1 and OUT2, provides the fast flywheel path to release the inductive energy of the motor. In most applications, the capacitance value is set to a value of 0.01μF to 0.1μF. Usually this capacitor is internally contained within the motor and not required externally. In some applications, especially in low speed motors, the large internal motor resistor connected with the bypass capacitor in parallel may result in an instantaneous large current when the motor starts up. It may however trigger a faulty OCP/OSP reaction which will fail to start up the motor. There are two ways to solve this phenomenon: decrease the bypass capacitor value or add a 47Ω to 100Ω resistor in series with the bypass capacitor.

Motor Current Sensing Resistor

The power dissipation of the selected motor current sensing resistor should be considered carefully. As described before the PGND maximum voltage should be lower than 0.5V. For a selected maximum motor current $I_{M(max)}$, the maximum power dissipation of current sensing resistor can be calculated by $0.5V\times I_{M(max)}$. For instance, if the $I_{M(max)}=1A$, the rated power of the selected current sensing resistor should be greater than 0.5W.

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 microcontroller, 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 Operation

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 which 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 set 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

Mnemonic	Description	Cycles	Flag Affected
Data Move			
MOV A,[m]	Move Data Memory to ACC	1	None
MOV [m],A	Move ACC to Data Memory	1^{Note}	None
MOV A,x	Move immediate data to ACC	1	None
Bit Operation			
CLR [m].i	Clear bit of Data Memory	1^{Note}	None
SET [m].i	Set bit of Data Memory	1^{Note}	None
Branch Operation			
JMP addr	Jump unconditionally	2	None
SZ [m]	Skip if Data Memory is zero	1^{Note}	None
SZA [m]	Skip if Data Memory is zero with data movement to ACC	1^{Note}	None
SZ [m].i	Skip if bit i of Data Memory is zero	1^{Note}	None
SNZ [m].i	Skip if bit i of Data Memory is not zero	1^{Note}	None
SIZ [m]	Skip if increment Data Memory is zero	1^{Note}	None
SDZ [m]	Skip if decrement Data Memory is zero	1^{Note}	None
SIZA [m]	Skip if increment Data Memory is zero with result in ACC	1^{Note}	None
SDZA [m]	Skip if decrement Data Memory is zero with result in ACC	1^{Note}	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 Operation			
TABRD [m]	Read table (specific page or current page) to TBLH and Data Memory	2^{Note}	None
TABRDL [m]	Read table (last page) to TBLH and Data Memory	2^{Note}	None
Miscellaneous			
NOP	No operation	1	None
CLR [m]	Clear Data Memory	1^{Note}	None
SET [m]	Set Data Memory	1^{Note}	None
CLR WDT	Clear Watchdog Timer	1	TO, PDF
SWAP [m]	Swap nibbles of Data Memory	1^{Note}	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.

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 bit wise 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

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 is 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 ← 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] ← 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] ← [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 ← [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] ← 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 ← 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

TABRD [m]	Read table (specific page or current page) to TBLH and Data Memory
Description	The low byte of the program code addressed by the table pointer (TBHP and TBLP or only TBLP if no TBHP) 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 for the latest version of the Package/Carton Information.

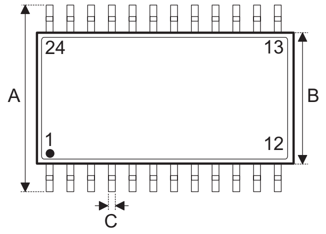
Additional supplementary information with regard to packaging is listed below. Click on the relevant section to be transferred to the relevant website page.

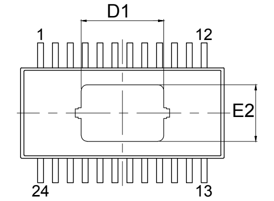
- Package Information (include Outline Dimensions, Product Tape and Reel Specifications)

- The Operation Instruction of Packing Materials

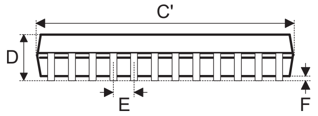
- Carton information

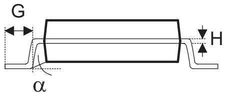
24-pin SSOP-EP (150mil) Outline Dimensions





THERMALLY ENHANCED VERIATIONS ONLY





Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.236 BSC	—
B	—	0.154 BSC	—
C	0.008	—	0.012
C’	—	0.341 BSC	—
D	—	—	0.069
D1	—	0.140	—
E	—	0.025 BSC	—
E2	—	0.096	—
F	0.000	—	0.004
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

Symbol	Dimensions mm		
	Min.	Nom.	Max.
A	—	6.00 BSC	—
B	—	3.90 BSC	—
C	0.20	—	0.30
C’	—	8.66 BSC	—
D	—	—	1.75
D1	—	3.56	—
E	—	0.635 BSC	—
E2	—	2.44	—
F	0.00	—	0.10
G	0.41	—	1.27
H	0.10	—	0.25
α	0°	—	8°

Copyright© 2020 by HOLTEK SEMICONDUCTOR INC.

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