Touch Key Flash Type 8-Bit MCU with LCD/LED Driver

BS85B12-3/BS85C20-3

Revision: V1.00 Date: February 1, 2011

www.holtek.com

Table of Contents

Technical Document

- Application Note
  HA0075E MCU Reset and Oscillator Circuits Application Note

Features

CPU Features

- Operating Voltage:
  f_{SYS}= 8MHz: V_{LVR}~5.5V
  f_{SYS}= 12MHz: 2.7V~5.5V
  f_{SYS}= 16MHz: 4.5V~5.5V
- Power down and wake-up functions to reduce power consumption
- Fully integrated low and high speed internal oscillators
  Low speed -- 32kHz
  High speed -- 8MHz, 12MHz, 16MHz
- Multi-mode operation: NORMAL, SLOW, IDLE and SLEEP
- All instructions executed in one or two instruction cycles
- Table read instructions
- 63 powerful instructions
- Up to 4 subroutine nesting levels
- Bit manipulation instruction

Peripheral Features

- Fully integrated 12 or 20 touch key functions -- require no external components
- Flash Program Memory: 2K×15 or 4K×15
- RAM Data Memory: 256×8 or 384×8
- EEPROM Memory: 64×8 or 128×8
- Watchdog Timer function
- Up to 38 bidirectional I/O lines
- Two or three Timer Modules
- Dual Time-Base functions for generation of fixed time interrupt signals
- I^2C and SPI interfaces
- Low voltage reset function
- Software controlled 4×14 or 4×22 LCD driver with 1/3 bias
- Software controlled 6×8 or 8×14 LED driver

General Description

These devices are a series of Flash Memory type 8-bit high performance RISC architecture microcontrollers with fully integrated touch key functions and LCD/LED drivers. With all touch key functions provided internally and with the convenience of Flash Memory multi-programming features, this device range has all the features to offer designers a reliable and easy means of implementing Touch Keys within their products applications. The touch key functions are fully integrated completely eliminating the need for external components. The inclusion of both LCD and LED driver functions allows for easy and cost effective solutions in applications that require to interface to these display types.

In addition to the flash program memory, other memory includes an area of RAM Data Memory as well as an area of EEPROM memory for storage of non-volatile data such as serial numbers, calibration data etc. Protective features such as an internal Watchdog Timer, Low Voltage Reset and Low Voltage Detector functions coupled with excellent noise immunity and ESD protection ensure that reliable operation is maintained in hostile electrical environments.

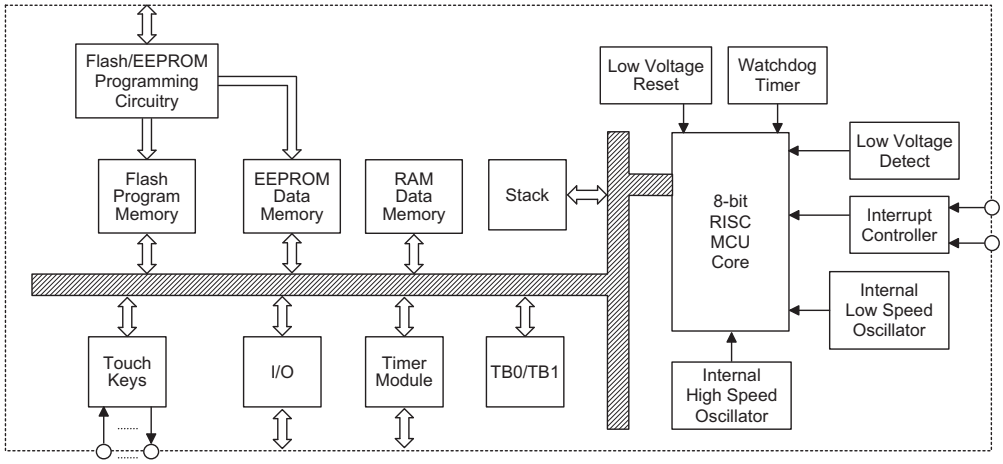
All devices include fully integrated low and high speed oscillators which require no external components for their implementation. The ability to operate and switch dynamically between a range of operating modes using different clock sources gives users the ability to optimise microcontroller operation and minimise power consumption. Easy communication with the outside world is provided using the internal I^2C and SPI interfaces, while the inclusion of flexible I/O programming features, Timer Modules and many other features further enhance device functionality and flexibility.

These touch key devices will find excellent use in a huge range of modern Touch Key product applications such as instrumentation, household appliances, electronically controlled tools to name but a few.

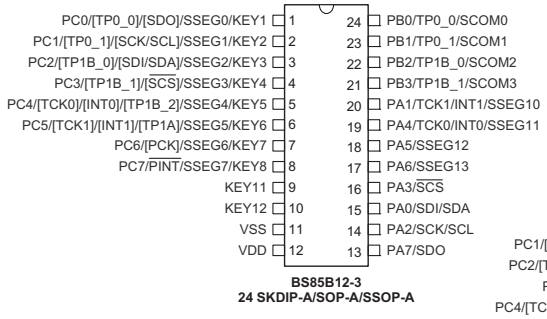
Selection Table

Part No.	Internal Clock	VDD	System Clock	Program Memory	Data Memory	Data EEPROM	I/O	Timer Module	Touch Key	LCD Driver	LED Driver	SPI/ I^2C	Stack	Package
BS85B12-3	8MHz 12MHz 16MHz	V_{LVR}~ 5.5V	8MHz~ 16MHz	2K×15	256×8	64×8	22	2	12	4×14	6×8	1	4	24/28SKDIP/SOP 24/28SSOP
BS85C20-3	8MHz 12MHz 16MHz	V_{LVR}~ 5.5V	8MHz~ 16MHz	4K×15	384×8	128×8	38	3	20	4×22	8×14	1	8	28SKDIP/SOP 28SSOP, 44QFP

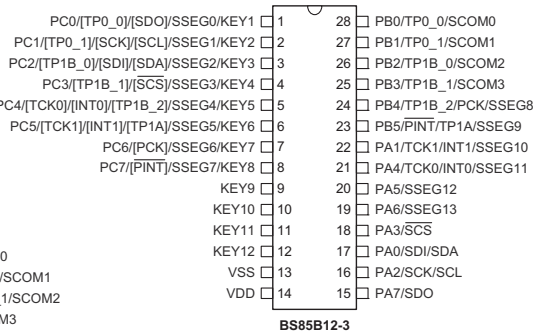
Block Diagram



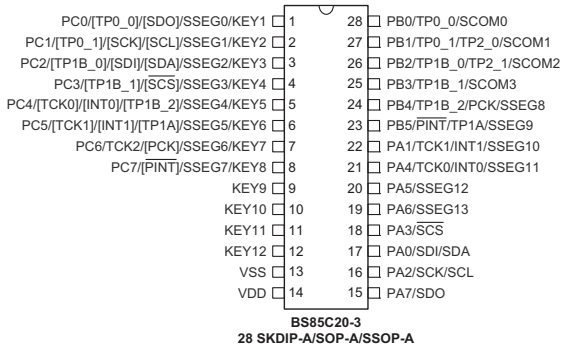
Pin Assignment



BS85B12-3
24 SKDIP-A/SOP-A/SSOP-A



BS85B12-3
28 SKDIP-A/SOP-A/SSOP-A



BS85C20-3
28 SKDIP-A/SOP-A/SSOP-A

BS85C20-3
44 QFP-A

Note: 1. Bracketed pin names indicate non-default pinout remapping locations.

2. If the pin-shared pin functions have multiple outputs simultaneously, its pin names at the right side of the " / " sign can be used for higher priority.

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.

BS85B12-3

Pin Name	Function	Register Select	I/T	O/T	Description
PA0/SDI/SDA	PA0	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	SDI	SIMC0	ST	—	SPI data input
	SDA	SIMC0	ST	NMOS	I^2C data I/O
PA1/TCK1/ INT1/SSEG10	PA1	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TCK1	TM1C0	ST	—	Timer Module 1 input
	INT1	INTC0	ST	—	External interrupt 1 input
	SSEG10	SLCDCn	—	LCD	Software controlled LCD SEG
PA2/SCK/SCL	PA2	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	SCK	SIMC0	ST	CMOS	SPI serial clock
	SCL	SIMC0	ST	NMOS	I^2C clock
PA3/$\overline{SCS}$	PA3	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	$\overline{SCS}$	SIMC0	ST	CMOS	SPI slave select
PA4/TCK0/ INT0/SSEG11	PA4	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TCK0	TM0C0	ST	—	Timer Module 0 input
	INT0	INTC0	ST	—	External interrupt 0 input
	SSEG11	SLCDCn	—	LCD	Software controlled LCD SEG
PA5/SSEG12	PA5	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	SSEG12	SLCDCn	—	LCD	Software controlled LCD SEG
PA6/SSEG13	PA6	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	SSEG13	SLCDCn	—	LCD	Software controlled LCD SEG
PA7/SDO	PA7	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	SDO	SIMC0	—	CMOS	SPI data output
PB0/TP0_0/ SCOM0	PB0	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP0_0	TMPCn	ST	CMOS	TM0 I/O
	SCOM0	SLCDCn	—	LCD	Software controlled LCD COM
PB1/TP0_1/ SCOM1	PB1	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP0_1	TMPCn	ST	CMOS	TM0 I/O
	SCOM1	SLCDCn	—	LCD	Software controlled LCD COM

Pin Name	Function	Register Select	I/T	O/T	Description
PB2/TP1B_0/SCOM2	PB2	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP1B_0	TMPCn	ST	CMOS	TM1 I/O
	SCOM2	SLCDCn	—	LCD	Software controlled LCD COM
PB3/TP1B_1/SCOM3	PB3	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP1B_1	TMPCn	ST	CMOS	TM1 I/O
	SCOM3	SLCDCn	—	LCD	Software controlled LCD COM
PB4/TP1B_2/PCK/SSEG8	PB4	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP1B_2	TMPCn	ST	CMOS	TM1 I/O
	PCK	SIMC0	—	CMOS	Peripheral clock output
	SSEG8	SLCDCn	—	LCD	Software controlled LCD SEG
PB5/$\overline{\text{PINT}}$/TP1A/SSEG9	PB5	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	$\overline{\text{PINT}}$	MFI3	ST	—	Peripheral interrupt input
	TP1A	TMPCn	ST	CMOS	TM1 I/O
	SSEG9	SLCDCn	—	LCD	Software controlled LCD SEG
PC0/TP0_0/SDO/SSEG0/KEY1	PC0	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP0_0	TMPCn	ST	CMOS	TM0 I/O
	SDO	SIMC0	—	CMOS	SPI data output
	SSEG0	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY1	TKMnC1	NS	—	Touch key input
PC1/TP0_1/SCK/SCL/SSEG1/KEY2	PC1	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP0_1	TMPCn	ST	CMOS	TM0 I/O
	SCK	SIMC0	ST	CMOS	SPI clock
	SCL	SIMC0	ST	NMOS	I^2C clock
	SSEG1	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY2	TKMnC1	NS	—	Touch key input
PC2/TP1B_0/SDI/SDA/SSEG2/KEY3	PC2	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP1B_0	TMPCn	ST	CMOS	TM1 I/O
	SDI	SIMC0	ST	—	SPI data input
	SDA	SIMC0	ST	NMOS	I^2C data I/O
	SSEG2	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY3	TKMnC1	NS	—	Touch key input
PC3/TP1B_1/$\overline{\text{SCS}}$/SSEG3/KEY4	PC3	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP1B_1	TMPCn	ST	CMOS	TM1 I/O
	$\overline{\text{SCS}}$	SIMC0	ST	CMOS	SPI slave select
	SSEG3	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY4	TKMnC1	NS	—	Touch key input

Pin Name	Function	Register Select	I/T	O/T	Description
PC4/TCK0/ INT0/TP1B_2/ SSEG4/KEY5	PC4	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TCK0	TM0C0	ST	—	Timer Module 0 input
	INT0	INTC0	ST	—	External interrupt 0 input
	TP1B_2	TMPCn	ST	CMOS	TM1 I/O
	SSEG4	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY5	TKMnC1	NS	—	Touch key input
PC5/TCK1/ INT1/TP1A/ SSEG5/KEY6	PC5	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TCK1	TM1C0	ST	—	Timer Module 1 input
	INT1	INTC0	ST	—	External interrupt 1 input
	TP1A	TMPCn	ST	CMOS	TM1 I/O
	SSEG5	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY6	TKMnC1	NS	—	Touch key input
PC6/PCK/ SSEG6/KEY7	PC6	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	PCK	SIMC0	—	CMOS	Peripheral clock output
	SSEG6	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY7	TKMnC1	NS	—	Touch key input
PC7/$\overline{\text{PINT}}$/ SSEG7/KEY8	PC7	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	$\overline{\text{PINT}}$	MFI3	ST	—	Peripheral interrupt input
	SSEG7	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY8	TKMnC1	NS	—	Touch key input
KEY9	KEY9	—	NS	—	Touch key input
KEY10	KEY10	—	NS	—	Touch key input
KEY11	KEY11	—	NS	—	Touch key input
KEY12	KEY12	—	NS	—	Touch key input
VDD	VDD	—	PWR	—	Power supply
VSS	VSS	—	PWR	—	Ground

Note: I/T: Input type
O/T: Output type
Register Select: Indicates register which selects alternative function
PWR: Power
ST: Schmitt Trigger input
CMOS: CMOS output
NMOS: NMOS output
LCD: LCD COM or SEG Vbias output
NS: Non-standard input or output
The pins in the table reflect that of the package with the largest number of pins. For this reason not all pins may exist on all package types.

BS85C20-3

Pin Name	Function	Register Select	I/T	O/T	Description
PA0/SDI/SDA	PA0	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	SDI	SIMC0	ST	—	SPI data input
	SDA	SIMC0	ST	NMOS	I^2C data I/O
PA1/TCK1/ INT1/SSEG10	PA1	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TCK1	TM1C0	ST	—	Timer Module 1 input
	INT1	INTC0	ST	—	External interrupt 1 input
	SSEG10	SLCDCn	—	LCD	Software controlled LCD SEG
PA2/SCK/SCL	PA2	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	SCK	SIMC0	ST	CMOS	SPI serial clock
	SCL	SIMC0	ST	NMOS	I^2C clock
PA3/$\overline{SCS}$	PA3	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	$\overline{SCS}$	SIMC0	ST	CMOS	SPI slave select
PA4/TCK0/ INT0/SSEG11	PA4	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	TCK0	TM0C0	ST	—	Timer Module 0 input
	INT0	INTC0	ST	—	External interrupt 0 input
	SSEG11	SLCDCn	—	LCD	Software controlled LCD SEG
PA5/SSEG12	PA5	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	SSEG12	SLCDCn	—	LCD	Software controlled LCD SEG
PA6/SSEG13	PA6	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	SSEG13	SLCDCn	—	LCD	Software controlled LCD SEG
PA7/SDO	PA7	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up.
	SDO	SIMC0	—	CMOS	SPI data output
PB0/TP0_0/ SCOM0	PB0	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP0_0	TMPCn	ST	CMOS	TM0 I/O
	SCOM0	SLCDCn	—	LCD	Software controlled LCD COM
PB1/TP0_1/ TP2_0/SCOM1	PB1	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP0_1	TMPCn	ST	CMOS	TM0 I/O
	TP2_0	TMPCn	ST	CMOS	TM2 I/O
	SCOM1	SLCDCn	—	LCD	Software controlled LCD COM
PB2/TP1B_0/ TP2_1/ SCOM2	PB2	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP1B_0	TMPCn	ST	CMOS	TM1 I/O
	TP2_1	TMPCn	ST	CMOS	TM2 I/O
	SCOM2	SLCDCn	—	LCD	Software controlled LCD COM

Pin Name	Function	Register Select	I/T	O/T	Description
PB3/TP1B_1/SCOM3	PB3	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP1B_1	TMPCn	ST	CMOS	TM1 I/O
	SCOM3	SLCDCn	—	LCD	Software controlled LCD COM
PB4/TP1B_2/PCK/SSEG8	PB4	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP1B_2	TMPCn	ST	CMOS	TM1 I/O
	PCK	SIMC0	—	CMOS	Peripheral clock output
	SSEG8	SLCDCn	—	LCD	Software controlled LCD SEG
PB5/$\overline{\text{PINT}}$/TP1A/SSEG9	PB5	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	$\overline{\text{PINT}}$	MFI3	ST	—	Peripheral interrupt input
	TP1A	TMPCn	ST	CMOS	TM1 I/O
	SSEG9	SLCDCn	—	LCD	Software controlled LCD SEG
PB6/SSEG20	PB6	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	SSEG20	SLCDCn	—	LCD	Software controlled LCD SEG
PB7/SSEG21	PB7	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	SSEG21	SLCDCn	—	LCD	Software controlled LCD SEG
PC0/TP0_0/SDO/SSEG0/KEY1	PC0	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP0_0	TMPCn	ST	CMOS	TM0 I/O
	SDO	SIMC0	—	CMOS	SPI data output
	SSEG0	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY1	TKMnC1	NS	—	Touch key input
PC1/TP0_1/SCK/SCL/SSEG1/KEY2	PC1	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP0_1	TMPCn	ST	CMOS	TM0 I/O
	SCK	SIMC0	ST	CMOS	SPI clock
	SCL	SIMC0	ST	NMOS	I^2C clock
	SSEG1	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY2	TKMnC1	NS	—	Touch key input
PC2/TP1B_0/SDI/SDA/SSEG2/KEY3	PC2	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP1B_0	TMPCn	ST	CMOS	TM1 I/O
	SDI	SIMC0	ST	—	SPI data input
	SDA	SIMC0	ST	NMOS	I^2C data I/O
	SSEG2	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY3	TKMnC1	NS	—	Touch key input
PC3/TP1B_1/$\overline{\text{SCS}}$/SSEG3/KEY4	PC3	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP1B_1	TMPCn	ST	CMOS	TM1 I/O
	$\overline{\text{SCS}}$	SIMC0	ST	CMOS	SPI slave select
	SSEG3	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY4	TKMnC1	NS	—	Touch key input

Pin Name	Function	Register Select	I/T	O/T	Description
PC4/TCK0/ INT0/TP1B_2/ SSEG4/KEY5	PC4	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TCK0	TM0C0	ST	—	Timer Module 0 input
	INT0	INTC0	ST	—	External interrupt 0 input
	TP1B_2	TMPCn	ST	CMOS	TM1 I/O
	SSEG4	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY5	TKMnC1	NS	—	Touch key input
PC5/TCK1/ INT1/TP1A/ SSEG5/KEY6	PC5	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TCK1	TM1C0	ST	—	Timer Module 1 input
	INT1	INTC0	ST	—	External interrupt 1 input
	TP1A	TMPCn	ST	CMOS	TM1 I/O
	SSEG5	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY6	TKMnC1	NS	—	Touch key input
PC6/TCK2/ PCK/SSEG6/ KEY7	PC6	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TCK2	TM2C0	ST	—	Timer Module 2 input
	PCK	SIMC0	—	CMOS	Peripheral clock output
	SSEG6	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY7	TKMnC1	NS	—	Touch key input
PC7/$\overline{\text{PINT}}$/ SSEG7/KEY8	PC7	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	$\overline{\text{PINT}}$	MFI3	ST	—	Peripheral interrupt input
	SSEG7	SLCDCn	—	LCD	Software controlled LCD SEG
	KEY8	TKMnC1	NS	—	Touch key input
PD0/TCK2/ KEY13	PD0	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TCK2	TM2C0	ST	—	Timer Module 2 input
	KEY13	TKMnC1	NS	—	Touch key input
PD1/KEY14	PD1	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	KEY14	TKMnC1	NS	—	Touch key input
PD2/KEY15	PD2	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	KEY15	TKMnC1	NS	—	Touch key input
PD3/KEY16	PD3	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	KEY16	TKMnC1	NS	—	Touch key input
PD4/KEY17	PD4	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	KEY17	TKMnC1	NS	—	Touch key input
PD5/KEY18	PD5	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	KEY18	TKMnC1	NS	—	Touch key input
PD6/KEY19	PD6	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	KEY19	TKMnC1	NS	—	Touch key input
PD7/KEY20	PD7	PDPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	KEY20	TKMnC1	NS	—	Touch key input

Pin Name	Function	Register Select	I/T	O/T	Description
PE0/$\overline{\text{PINT}}$/SSEG14	PE0	PEPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	$\overline{\text{PINT}}$	MFI3	ST	—	Peripheral interrupt input
	SSEG14	SLCDCn	—	LCD	Software controlled LCD SEG
PE1/TP2_0/PCK/SSEG15	PE1	PEPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP2_0	TMPCn	ST	CMOS	TM2 I/O
	PCK	SIMC0	—	CMOS	Peripheral Clock Output
	SSEG15	SLCDCn	—	LCD	Software controlled LCD SEG
PE2/TP2_1/$\overline{\text{SCS}}$/SSEG16	PE2	PEPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	TP2_1	TMPCn	ST	CMOS	TM2 I/O
	$\overline{\text{SCS}}$	SIMC0	ST	CMOS	SPI select
	SSEG16	SLCDCn	—	LCD	Software controlled LCD SEG
PE3/SCK/SCL/SSEG17	PE3	PEPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	SCK	SIMC0	ST	CMOS	SPI serial clock
	SCL	SIMC0	ST	NMOS	I^2C clock
	SSEG17	SLCDCn	—	LCD	Software controlled LCD SEG
PE4/SDI/SDA/SSEG18	PE4	PEPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	SDI	SIMC0	ST	—	SPI data input
	SDA	SIMC0	ST	NMOS	I^2C data I/O
	SSEG18	SLCDCn	—	LCD	Software controlled LCD SEG
PE5/SDO/SSEG19	PE5	PEPU	ST	CMOS	General purpose I/O. Register enabled pull-up.
	SDO	SIMC0	ST	CMOS	SPI data output
	SSEG19	SLCDCn	—	LCD	Software controlled LCD SEG
KEY9	KEY9	—	NS	—	Touch key input
KEY10	KEY10	—	NS	—	Touch key input
KEY11	KEY11	—	NS	—	Touch key input
KEY12	KEY12	—	NS	—	Touch key input
VDD	VDD	—	PWR	—	Power supply
VSS	VSS	—	PWR	—	Ground

Note: I/T: Input type
O/T: Output type
Register Select: Indicates register which selects alternative function
PWR: Power
ST: Schmitt Trigger input
CMOS: CMOS output
NMOS: NMOS output
LCD: LCD COM or SEG Vbias output
NS: Non-standard input or output
The pins in the table reflect that of the package with the largest number of pins. For this reason not all pins may exist on all package types.

Absolute Maximum Ratings

Supply Voltage ..... V_{SS}–0.3V to V_{SS}+6.0V
Storage Temperature ..... –50°C to 125°C
Input Voltage ..... V_{SS}–0.3V to V_{DD}+0.3V
Operating Temperature ..... –40°C to 85°
CI_{OL} Total 80mA
I_{OH} Total ..... –80mA
Total Power Dissipation 500mW

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

D.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage (HIRC)	—	f_{SYS}=8MHz	V_{LVR}	—	5.5	V
			f_{SYS}=12MHz	2.7	—	5.5	V
			f_{SYS}=16MHz	4.5	—	5.5	V
I_{DD1}	Operating Current (HIRC), (f_{SYS}=f_H)	3V	No load, f_H=8MHz, WDT enable	—	1.2	1.8	mA
		5V		—	2.7	4.1	mA
		3V	No load, f_H=12MHz, WDT enable	—	1.9	2.9	mA
		5V		—	4.2	6.3	mA
		5V	No load, f_H=16MHz, WDT enable	—	5.6	8.4	mA
I_{DD2}	Operating Current (LIRC), (f_{SYS}=f_L)	3V	No load, f_L=32kHz, WDT enable	—	15	30	μA
		5V		—	30	50	μA
I_{IDLE0}	IDLE0 Mode Standby Current	3V	No load, LVR disable	—	1.5	3.0	μA
		5V		—	3.0	6.0	μA
I_{IDLE1}	IDLE1 Mode Standby Current	3V	No load, LVR disable, f_{SYS}=12MHz on	—	0.9	1.4	mA
		5V		—	1.6	2.4	mA
I_{SLEEP}	SLEEP1 Mode Standby Current	3V	No load, LVR disable	—	1.5	3.0	μA
		5V		—	2.5	5.0	μA
V_{IL}	Input Low Voltage for I/O Ports or Input Pins	5V	—	0	—	1.5	V
		—		0	—	$0.2V_{DD}$	V
V_{IH}	Input High Voltage for I/O Ports or Input Pins	5V	—	3.5	—	5.0	V
		—		$0.8V_{DD}$	—	V_{DD}	V
V_{LVR}	LVR Voltage Level	—	LVR Enable	–5%	2.55	+5%	V
V_{OL1}	Output Low Voltage I/O Port	3V	I_{OL}=9mA	—	—	0.3	V
		5V	I_{OL}=20mA	—	—	0.5	V

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{OL2}	Output Low Voltage I/O Port (PB) (High Current Enable)	3V	I_{OL}=18mA	—	—	0.3	V
		5V	I_{OL}=40mA	—	—	0.5	V
V_{OH1}	Output High Voltage I/O Port	3V	I_{OH}=−3.2mA	2.7	—	—	V
		5V	I_{OH}=−7.4mA	4.5	—	—	V
V_{OH2}	Output High Voltage I/O Port (PA, PE) (High Current Enable)	3V	I_{OH}=−6.4mA	2.7	—	—	V
		5V	I_{OH}=−15.0mA	4.5	—	—	V
V_{LVR}	LVR Voltage Level	—	LVR Enable, 2.55V option	−5%	2.55	+5%	V
V_{LVD}	LVD Voltage Level	—	LVDEN=1, V_{LVD}=2.0V	−5%	2.00	+5%	V
			LVDEN=1, V_{LVD}=2.2V	−5%	2.20	+5%	V
			LVDEN=1, V_{LVD}=2.4V	−5%	2.40	+5%	V
			LVDEN=1, V_{LVD}=2.7V	−5%	2.70	+5%	V
			LVDEN=1, V_{LVD}=3.0V	−5%	3.00	+5%	V
			LVDEN=1, V_{LVD}=3.3V	−5%	3.30	+5%	V
			LVDEN=1, V_{LVD}=3.6V	−5%	3.60	+5%	V
			LVDEN=1, V_{LVD}=4.2V	−5%	4.20	+5%	V
I_{LV}	Additional Power Consumption if LVD is Used	—	NORMAL or SLOW Mode	—	2	5	μA
			IDLE or SLEEP Mode	—	15	30	μA
ISCOM	SCOM Operating Current	5V	ISEL[1:0]=00	17.5	25.0	32.5	μA
			ISEL[1:0]=01	35	50	65	μA
			ISEL[1:0]=10	70	100	130	μA
			ISEL[1:0]=11	140	200	260	μA
ISSEG	SSEG Operating Current	5V	ISEL[1:0]=00	17.5	25.0	32.5	μA
			ISEL[1:0]=01	35	50	65	μA
			ISEL[1:0]=10	70	100	130	μA
			ISEL[1:0]=11	140	200	260	μA
VSSOM	Voltage for LCD SCOM	5V	1/3 V_{DD}	−3%	0.33	+3%	V_{DD}
			2/3 V_{DD}	−3%	0.67	+3%	V_{DD}
VSSEG	Voltage for LCD SSEG	5V	1/3 V_{DD}	−3%	0.33	+3%	V_{DD}
			2/3 V_{DD}	−3%	0.67	+3%	V_{DD}
R_{PH}	Pull-high Resistance for I/O Ports	3V	—	20	60	100	kΩ
		5V		10	30	50	kΩ

A.C. Characteristics

Ta=25°C

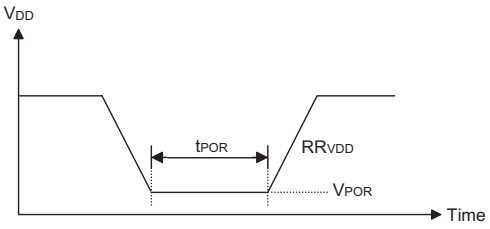
Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{CPU}	Operating Clock	—	V_{LVR}~5.5V	DC	—	8	MHz
			2.7V~5.5V	DC	—	12	MHz
			4.5V~5.5V	DC	—	16	MHz
f_{HIRC}	System Clock (HIRC)	3V/5V	Ta=25°C	–2%	8	+2%	MHz
		3V/5V	Ta=25°C	–2%	12	+2%	MHz
		5V	Ta=25°C	–2%	16	+2%	MHz
		3V/5V	Ta=0~70°C	–4%	8	+3%	MHz
		3V/5V	Ta=0~70°C	–4%	12	+3%	MHz
		5V	Ta=0~70°C	–4%	16	+3%	MHz
		2.5V~4.0V	Ta=0~70°C	–9%	8	+6%	MHz
		3.0V~5.5V	Ta=0~70°C	–5%	8	+12%	MHz
		2.7V~4.0V	Ta=0~70°C	–9%	12	+5%	MHz
		3.0V~5.5V	Ta=0~70°C	–5%	12	+11%	MHz
		4.5V~5.5V	Ta=0~70°C	–5%	16	+5%	MHz
		2.5V~4.0V	Ta= –40°C~85°C	–12%	8	+6%	MHz
		3.0V~5.5V	Ta= –40°C~85°C	–8%	8	+12%	MHz
		2.7V~4.0V	Ta= –40°C~85°C	–13%	12	+5%	MHz
		3.0V~5.5V	Ta= –40°C~85°C	–8%	12	+11%	MHz
		4.5V~5.5V	Ta= –40°C~85°C	–7%	16	+5%	MHz
f_{LIRC}	System Clock (LIRC)	5V	—	–10%	32	+10%	kHz
		2.2V~5.5V	Ta=–40°C~+85°C	–50%	32	+60%	kHz
f_{TIMER}	Timer Input Pin Frequency	—	—	—	—	1	f_{SYS}
t_{INT}	Interrupt Pulse Width	—	—	1	—	—	μs
t_{LVR}	Low Voltage Width to Reset	—	—	60	120	240	μs
t_{LVD}	Low Voltage Width to Interrupt	—	—	180	240	360	μs
t_{LVDS}	LVDO Stable Time	—	—	15	—	—	μs
t_{EERD}	EEPROM Read Time	—	—	—	2	4	t_{SYS}
t_{EEWR}	EEPROM Write Time	—	—	—	2	4	ms
t_{SST}	System Start-up Timer Period (Wake-up from HALT)	—	f_{SYS}=HIRC	—	15~16	—	t_{SYS}
			f_{SYS}=LIRC	—	1~2	—	

Note: 1. $t_{SYS}=1/f_{SYS}$
2. To maintain the accuracy of the internal HIRC oscillator frequency, a 0.1μF decoupling capacitor should be connected between VDD and VSS and located as close to the device as possible.

Power-on Reset Characteristics

Ta=25°C

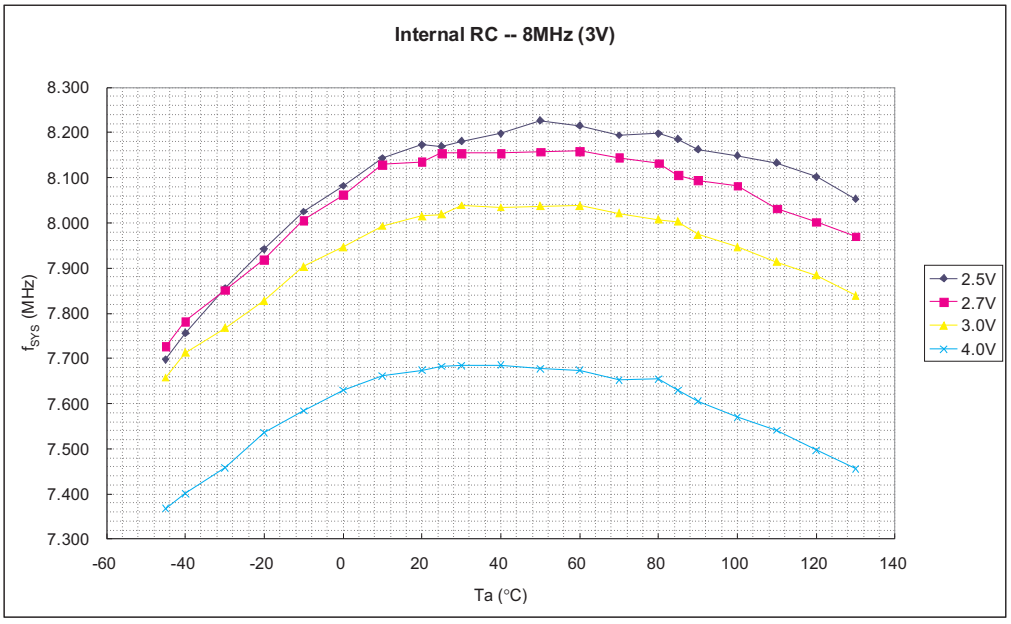
Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{POR}	VDD Start Voltage to Ensure Power-on Reset	—	—	—	—	100	mV
$R_{POR\ AC}$	VDD Raising Rate to Ensure Power-on Reset	—	—	0.035	—	—	V/ms
t_{POR}	Minimum Time for VDD Stays at V_{POR} to Ensure Power-on Reset	—	—	1	—	—	ms

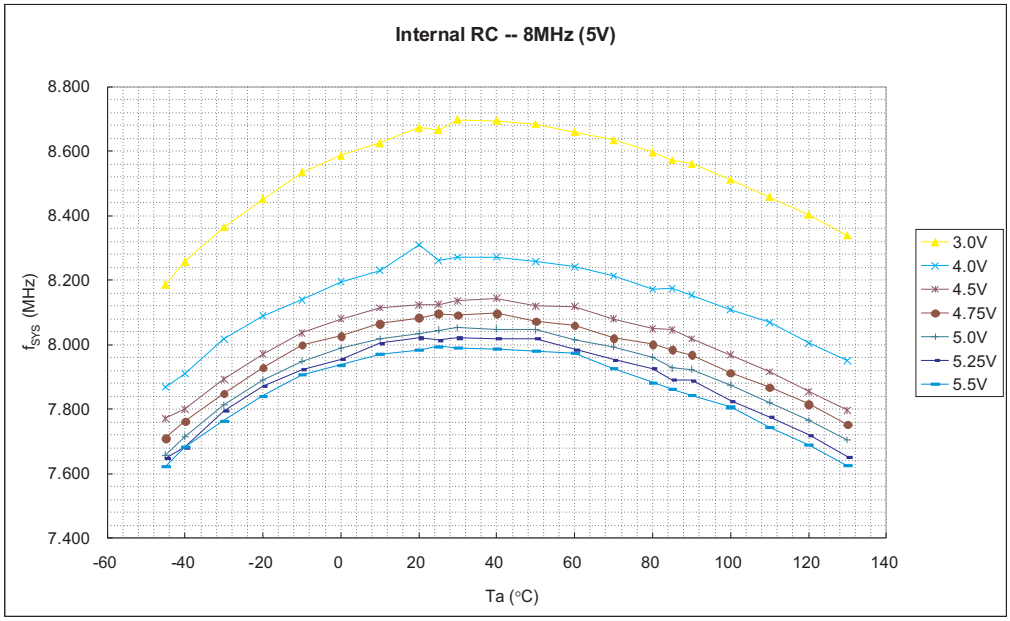


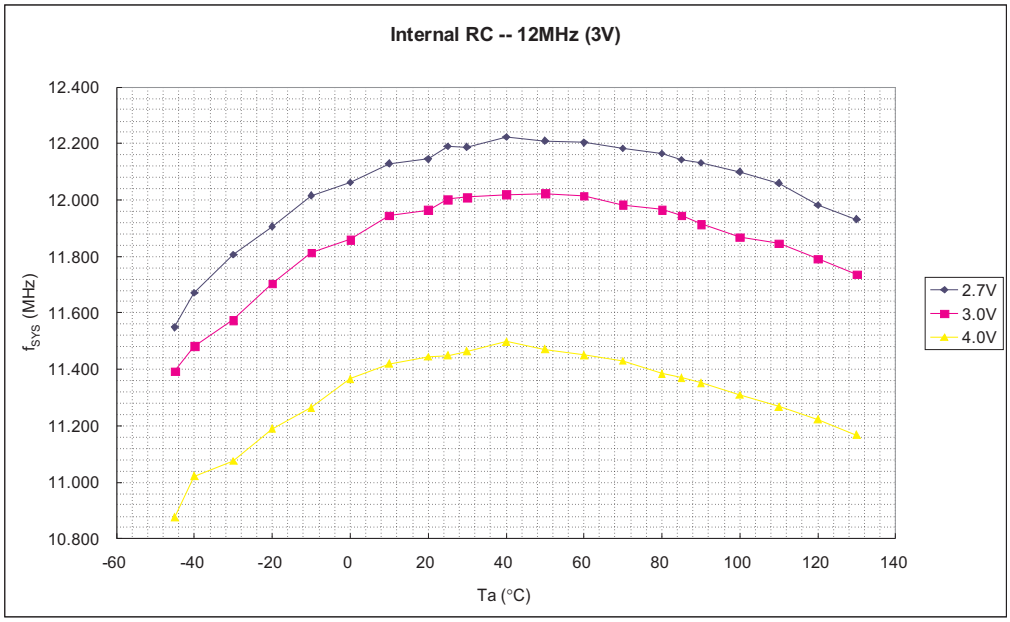
Oscillator Temperature/Frequency Characteristics

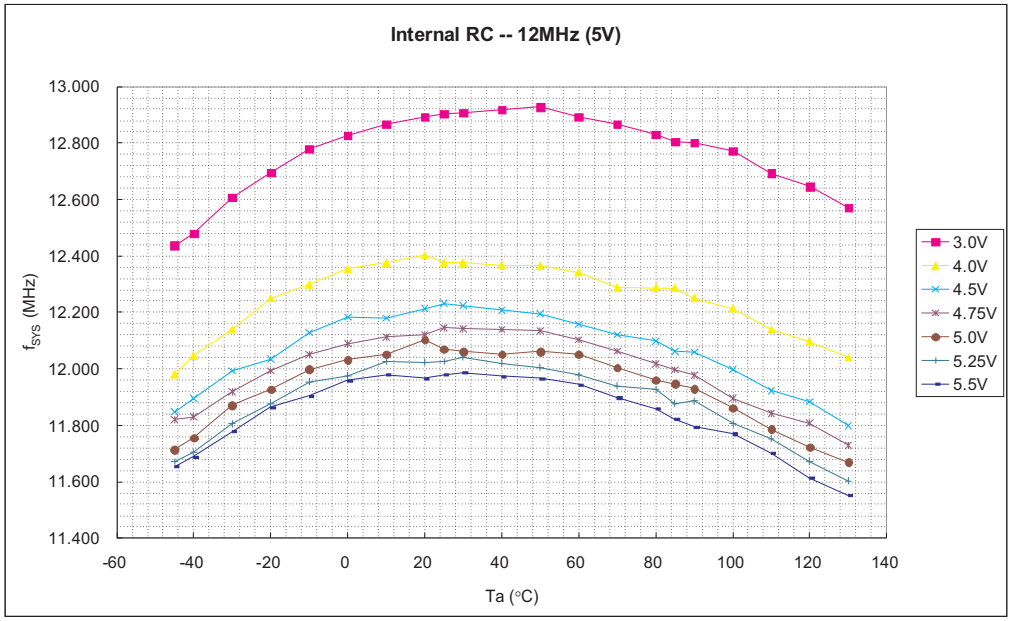
The following characteristic graphics depicts typical oscillator behavior. The data presented here is a statistical summary of data gathered on units from different lots over a period of time. This is for information only and the figures were not tested during manufacturing.

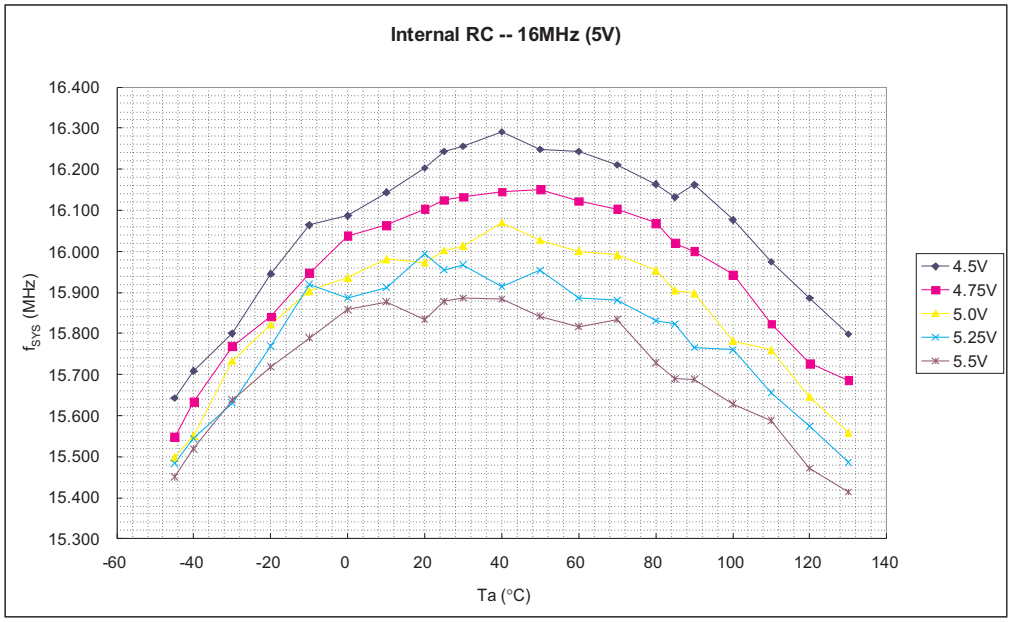
In some of the graphs, the data exceeding the specified operating range are shown for information purposes only. The device will operate properly only within the specified range.









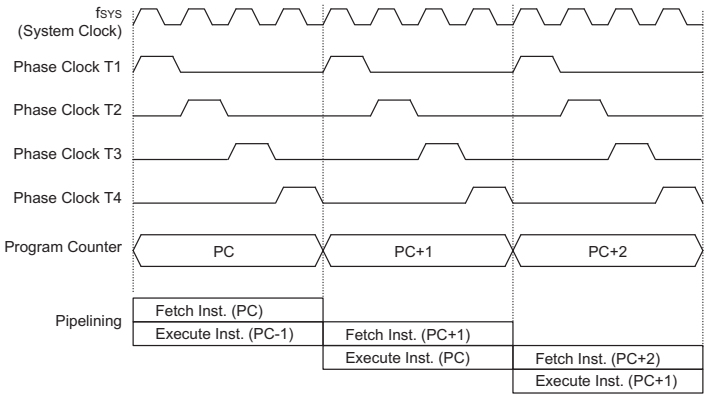


System Architecture

A key factor in the high-performance features of the Holtek range of microcontroller is attributed to their internal system architecture. The range of devices take advantage of the usual features found within RISC microcontroller providing increased speed of operation and enhanced performance. The pipelining scheme is implemented in such a way that instruction fetching and instruction execution are overlapped, hence instructions are effectively executed in one cycle, with the exception of branch or call instructions. An 8-bit wide ALU is used in practically all 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 control system with maximum reliability and flexibility. This makes the device suitable for low-cost, high-volume production for controller applications.

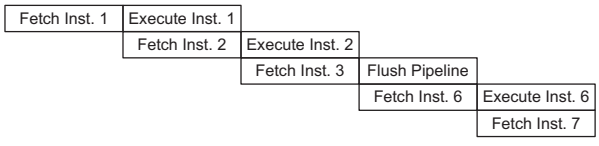
Clocking and Pipelining

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



System Clocking and Pipelining

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



Instruction Fetching

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.

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.

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.

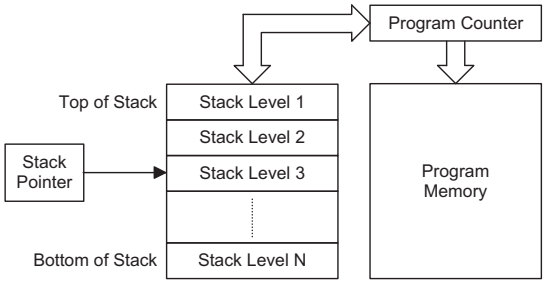
Stack

This is a special part of the memory which is used to save the contents of the Program Counter only. The stack has multiple levels depending upon the device and 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.

Device	Stack Levels
BS85B12-3	4
BS85C20-3	8



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

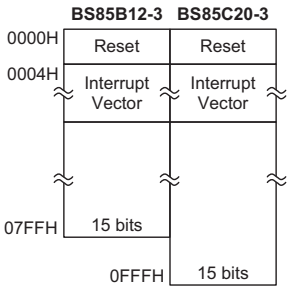
Flash Program Memory

The Program Memory is the location where the user code or program is stored. For this device series 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, these Flash devices 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×15 bits or 4K×15 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.

Device	Capacity
BS85B12-3	2Kx15
BS85C20-3	4Kx15



Flash 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 register, 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]" instructions, respectively. 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. Any unused bits in this transferred higher order byte will be read as "0".

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

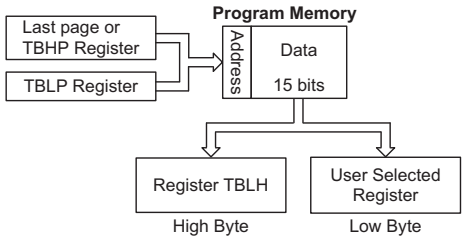


Table Program Example

The following example using the BS85B12-3 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 "700H" which refers to the start address of the last page within the 2K words Program Memory of the device. The table pointer is setup here to have an initial value of "06H". This will ensure that the first data read from the data table will be at the Program Memory address "706H" or 6 locations after the start of the last page. Note that the value for the table pointer is referenced to the first address of the present page if the "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.

```
Tempreg1  db ?      ; temporary register #1
tempreg2  db ?      ; temporary register #2
:
:
mov a,06h           ; initialise low table pointer - note that this address
mov tblp,a          ; is referenced
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 "706H" 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 "705H" transferred to tempreg2 and TBLH in
                    ; this example the data "1AH" is transferred to tempreg1 and data
                    ; "0FH" to register tempreg2
:
:
org 700h            ; sets initial address of program memory

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

In Circuit Programming

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.

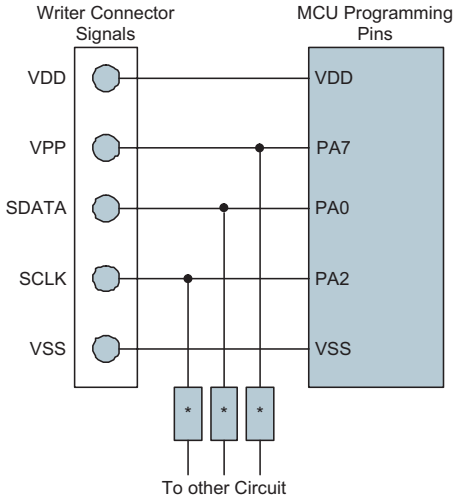
As an additional convenience, Holtek has provided a means of programming the microcontroller in-circuit using a 5-pin interface. This provides manufacturers with the possibility of manufacturing their circuit boards complete with a programmed or un-programmed microcontroller, and then programming or 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.

The Holtek Flash MCU to Writer Programming Pin correspondence table is as follows:

Holtek Writer Pin Name	Device Pin Name	Pin Description
SDATA	PA0	Serial Address and data -- read/write
SCLK	PA2	Address and data serial clock input
VPP	PA7	Reset input
VDD	VDD	Power Supply (5.0V)
VSS	VSS	Ground

The Program Memory and EEPROM data memory can both be programmed serially in-circuit using this 5-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 and one line for the reset. The technical details regarding the in-circuit programming of the devices are beyond the scope of this document and will be supplied in supplementary literature.

During the programming process the PA7 pin will be held low by the programmer disabling the normal operation of the microcontroller and taking control of the PA0 and PA2 I/O pins for data and clock programming purposes. The user must there take care 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.

RAM Data Memory

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

Structure

Divided into two sections, the first of these is an area of RAM, known as the Special Function Data Memory. Here are located registers which 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.

Device	Capacity	Bank 0	Bank 1	Bank 2
BS85B12-3	256×8	80H~FFH	80H~FFH	—
BS85C20-3	384×8	80H~FFH	80H~FFH	80H~FFH

General Purpose Data Memory

The second area of Data Memory is known as the General Purpose Data Memory, which is reserved for general purpose use. All locations within this area are read and write accessible under program control.

The overall Data Memory is subdivided into two or three banks. The Special Purpose Data Memory registers are accessible in all banks, with the exception of the EEC register at address 40H, which is only accessible in Bank 1. Switching between the different Data Memory banks is achieved by setting the Bank Pointer to the correct value. The start address of the Data Memory for all devices is the address 00H.

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 from Bank 0 while the IAR1 and MP1 register pair can access data from any bank. As the Indirect Addressing Registers are not physically implemented, reading the Indirect Addressing Registers indirectly will return a result of "00H" and writing to the registers indirectly will result in no operation.

Memory Pointers – MP0, MP1

Two Memory Pointers, known as MP0 and MP1 are provided. These Memory Pointers are physically implemented in the Data Memory and can be manipulated in the same way as normal registers providing a convenient way with which to 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 according to BP register. Direct Addressing can only be used with Bank 0, all other Banks must be addressed indirectly using MP1 and IAR1. Note that for this series of devices, the Memory Pointers, MP0 and MP1, are both 8-bit registers and used to access the Data Memory together with their corresponding indirect addressing registers IAR0 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 example shown above, no reference is made to specific RAM addresses.

BS85B12-3

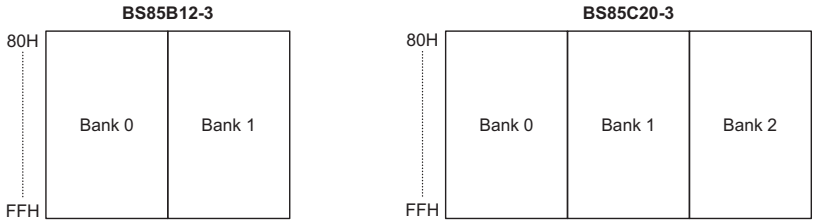
	Bank 0, Bank 1	Bank 0	Bank 1
00H	IAR0	Unused	EEC
01H	MP0	EEA	
02H	IAR1	EED	
03H	MP1	TMPC0	
04H	BP	Unused	
05H	ACC	PRM0	
06H	PCL	PRM1	
07H	TBLP	PRM2	
08H	TBLH	TM1C0	
09H	TBHP	TM1C1	
0AH	STATUS	TM1C2	
0BH	SMOD	TM1DL	
0CH	LVDC	TM1DH	
0DH	INTEG	TM1AL	
0EH	WDTC	TM1AH	
0FH	TBC	TM1BL	
10H	INTC0	TM1BH	
11H	INTC1	Unused	
12H	INTC2	Unused	
13H	Unused	Unused	
14H	MFI0	Unused	
15H	MFI1	Unused	
16H	MFI2	Unused	
17H	MF13	CTRL	
18H	PAWU	TKM016DH	
19H	PAPU	TKM016DL	
1AH	PA	Reserved	
1BH	PAC	Reserved	
1CH	PBPU	TKM0C0	
1DH	PB	TKM0C1	
1EH	PBC	TKM0C2	
1FH	PCPU	TKM0C3	
20H	PC	TKM116DH	
21H	PCC	TKM116DL	
22H	Unused	Reserved	
23H	Unused	Reserved	
24H	Unused	TKM1C0	
25H	Unused	TKM1C1	
26H	Unused	TKM1C2	
27H	Unused	TKM1C3	
28H	SLCDC0	TKM216DH	
29H	SLCDC1	TKM216DL	
2AH	SLCDC2	Reserved	
2BH	Unused	Reserved	
2CH	SLEDC0	TKM2C0	
2DH	SLEDC1	TKM2C1	
2EH	Unused	TKM2C2	
2FH	Unused	TKM2C3	
30H	Unused	Unused	
31H	Unused	Unused	
32H	Unused	Unused	
33H	Unused	Unused	
34H	Unused	Unused	
35H	I2CTOC	Unused	
36H	SIMC0	Unused	
37H	SIMC1	Unused	
38H	SIMD	Unused	
39H	SIMA/SIMC2	Unused	
3AH	TM0C0	Unused	
3BH	TM0C1	Unused	
3CH	TM0DL	Unused	
3DH	TM0DH	Unused	
3EH	TM0AL	Unused	
3FH	TM0AH	Unused	

BS85C20-3

	Bank 0~2	Bank 0, 2	Bank 1
00H	IAR0	Unused	EEC
01H	MP0	EEA	
02H	IAR1	EED	
03H	MP1	TMPC0	
04H	BP	TMPC1	
05H	ACC	PRM0	
06H	PCL	PRM1	
07H	TBLP	PRM2	
08H	TBLH	TM1C0	
09H	TBHP	TM1C1	
0AH	STATUS	TM1C2	
0BH	SMOD	TM1DL	
0CH	LVDC	TM1DH	
0DH	INTEG	TM1AL	
0EH	WDTC	TM1AH	
0FH	TBC	TM1BL	
10H	INTC0	TM1BH	
11H	INTC1	TM2C0	
12H	INTC2	TM2C1	
13H	INTC3	TM2DL	
14H	MFI0	TM2DH	
15H	MFI1	TM2AL	
16H	MFI2	TM2AH	
17H	MFI3	CTRL	
18H	PAWU	TKM016DH	
19H	PAPU	TKM016DL	
1AH	PA	Reserved	
1BH	PAC	Reserved	
1CH	PBPU	TKM0C0	
1DH	PB	TKM0C1	
1EH	PBC	TKM0C2	
1FH	PCPU	TKM0C3	
20H	PC	TKM116DH	
21H	PCC	TKM116DL	
22H	PDPU	Reserved	
23H	PD	Reserved	
24H	PDC	TKM1C0	
25H	PEPU	TKM1C1	
26H	PE	TKM1C2	
27H	PEC	TKM1C3	
28H	SLCDC0	TKM216DH	
29H	SLCDC1	TKM216DL	
2AH	SLCDC2	Reserved	
2BH	SLCDC3	Reserved	
2CH	SLEDC0	TKM2C0	
2DH	SLEDC1	TKM2C1	
2EH	SLEDC2	TKM2C2	
2FH	Unused	TKM2C3	
30H	MFI4	TKM316DH	
31H	MFI5	TKM316DL	
32H	Unused	Reserved	
33H	Unused	Reserved	
34H	Unused	TKM3C0	
35H	I2CTOC	TKM3C1	
36H	SIMC0	TKM3C2	
37H	SIMC1	TKM3C3	
38H	SIMD	TKM4DH	
39H	SIMA/SIMC2	TKM4DL	
3AH	TM0C0	Reserved	
3BH	TM0C1	Reserved	
3CH	TM0DL	TKM4C0	
3DH	TM0DH	TKM4C1	
3EH	TM0AL	TKM4C2	
3FH	TM0AH	TKM4C3	

Special Purpose Data Memory

Note: The "Reserved" bytes shown in the table must not be modified by the user.



General Purpose Data Memory

Bank Pointer – BP

For this series of devices, the Data Memory is divided into two or three banks. Selecting the required Data Memory area is achieved using the Bank Pointer. Bit 0 and 1 is used to select Data Memory Banks 0~2.

The Data Memory is initialised to Bank 0 after a reset, except for a WDT time-out reset in the Power Down Mode, in which case, the Data Memory bank remains unaffected. It should be noted that the Special Function Data Memory is not affected by the bank selection, which means that the Special Function Registers can be accessed from within any bank. Directly addressing the Data Memory will always result in Bank 0 being accessed irrespective of the value of the Bank Pointer. Accessing data from banks other than Bank 0 must be implemented using indirect addressing.

BP Register -- BS85B12-3

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

Bit 7~1 unimplemented, read as "0"

Bit 0 **DMBP0**: select data memory banks
0: bank 0
1: bank 1

BP Register -- BS85C20-3

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

Bit 7~2 unimplemented, read as "0"

Bit 1~0 **DMBP1, DMBP0**: select data memory banks
00: bank 0
01: bank 1
10: bank 2
11: undefined

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 pointer and indicates 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.

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
C is also affected by a rotate through carry instruction.

EEPROM Data Memory

The devices contain an area of internal EEPROM Data Memory. EEPROM, which stands for Electrically Erasable Programmable Read Only Memory, is by its nature a non-volatile form of re-programmable memory, with data retention even when its power supply is removed. By incorporating this kind of data memory, a whole new host of application possibilities are made available to the designer. The availability of EEPROM storage allows information such as product identification numbers, calibration values, specific user data, system setup data or other product information to be stored directly within the product microcontroller. The process of reading and writing data to the EEPROM memory has been reduced to a very trivial affair.

EEPROM Data Memory Structure

The EEPROM Data Memory capacity is 64×8 or 128×8 bits for this series of devices. Unlike the Program Memory and RAM Data Memory, the EEPROM Data Memory is not directly mapped into memory space and is therefore not directly addressable in the same way as the other types of memory. Read and Write operations to the EEPROM are carried out in single byte operations using an address and data register in Bank 0 and a single control register in Bank 1.

Device	Capacity	Address
BS85B12-3	64×8	00H ~ 3FH
BS85C20-3	128×8	00H ~ 7FH

EEPROM Registers

Three registers control the overall operation of the internal EEPROM Data Memory. These are the address register, EEA, the data register, EED and a single control register, EEC. As both the EEA and EED registers are located in Bank 0, they can be directly accessed in the same was as any other Special Function Register. The EEC register however, being located in Bank1, cannot be addressed directly and can only be read from or written to indirectly using the MP1 Memory Pointer and Indirect Addressing Register, IAR1. Because the EEC control register is located at address 40H in Bank 1, the MP1 Memory Pointer must first be set to the value 40H and the Bank Pointer register, BP, set to the value, 01H, before any operations on the EEC register are executed.

EEPROM Register List

- BS85B12-3

Name	Bit							
	7	6	5	4	3	2	1	0
EEA	—	—	D5	D4	D3	D2	D1	D0
EED	D7	D6	D5	D4	D3	D2	D1	D0
EEC	—	—	—	—	WREN	WR	RDEN	RD

- BS85C20-3

Name	Bit							
	7	6	5	4	3	2	1	0
EEA	—	D6	D5	D4	D3	D2	D1	D0
EED	D7	D6	D5	D4	D3	D2	D1	D0
EEC	—	—	—	—	WREN	WR	RDEN	RD

EEA Register

- BS85B12-3

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

"x" unknown

Bit 7, 6 unimplemented, read as "0"

Bit 5~0 Data EEPROM address
Data EEPROM address bit 5~bit 0

- BS85C20-3

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

"x" unknown

Bit 7 unimplemented, read as "0"

Bit 6~0 Data EEPROM address
Data EEPROM address bit 6~bit 0

EEC Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	WREN	WR	RDEN	RD
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 unimplemented, read as "0"

Bit 3 **WREN**: data EEPROM write enable
0: disable
1: enable
This is the Data EEPROM Write Enable Bit which must be set high before Data EEPROM write operations are carried out. Clearing this bit to zero will inhibit Data EEPROM write operations.

Bit 2 **WR**: EEPROM write control
0: Write cycle has finished
1: Activate a write cycle
This is the Data EEPROM Write Control Bit and when set high by the application program will activate a write cycle. This bit will be automatically reset to zero by the hardware after the write cycle has finished. Setting this bit high will have no effect if the WREN has not first been set high.

Bit 1 **RDEN**: Data EEPROM read enable
0: disable
1: enable
This is the Data EEPROM Read Enable Bit which must be set high before Data EEPROM read operations are carried out. Clearing this bit to zero will inhibit Data EEPROM read operations.

Bit 0 **RD**: EEPROM read control
0: read cycle has finished
1: activate a read cycle
This is the Data EEPROM Read Control Bit and when set high by the application program will activate a read cycle. This bit will be automatically reset to zero by the hardware after the read cycle has finished. Setting this bit high will have no effect if the RDEN has not first been set high.

Note: The WREN, WR, RDEN and RD can not be set to "1" at the same time in one instruction. The WR and RD can not be set to "1" at the same time.

EED 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	x	x	x	x	x	x	x	x

"x" unknown

Bit 7~0 Data EEPROM data
Data EEPROM data bit 7~bit 0

Reading Data from the EEPROM

To read data from the EEPROM, the read enable bit, RDEN, in the EEC register must first be set high to enable the read function. The EEPROM address of the data to be read must then be placed in the EEA register. If the RD bit in the EEC register is now set high, a read cycle will be initiated. Setting the RD bit high will not initiate a read operation if the RDEN bit has not been set. When the read cycle terminates, the RD bit will be automatically cleared to zero, after which the data can be read from the EED register. The data will remain in the EED register until another read or write operation is executed. The application program can poll the RD bit to determine when the data is valid for reading.

Writing Data to the EEPROM

The EEPROM address of the data to be written must first be placed in the EEA register and the data placed in the EED register. To write data to the EEPROM, the write enable bit, WREN, in the EEC register must first be set high to enable the write function. After this, the WR bit in the EEC register must be immediately set high to initiate a write cycle. These two instructions must be executed consecutively. The global interrupt bit EMI should also first be cleared before implementing any write operations, and then set again after the write cycle has started. Note that setting the WR bit high will not initiate a write cycle if the WREN bit has not been set. As the EEPROM write cycle is controlled using an internal timer whose operation is asynchronous to microcontroller system clock, a certain time will elapse before the data will have been written into the EEPROM. Detecting when the write cycle has finished can be implemented either by polling the WR bit in the EEC register or by using the EEPROM interrupt. When the write cycle terminates, the WR bit will be automatically cleared to zero by the microcontroller, informing the user that the data has been written to the EEPROM. The application program can therefore poll the WR bit to determine when the write cycle has ended.

Write Protection

Protection against inadvertent write operation is provided in several ways. After the device is powered-on the Write Enable bit in the control register will be cleared preventing any write operations. Also at power-on the Bank Pointer, BP, will be reset to zero, which means that Data Memory Bank 0 will be selected. As the EEPROM control register is located in Bank 1, this adds a further measure of protection against spurious write operations. During normal program operation, ensuring that the Write Enable bit in the control register is cleared will safeguard against incorrect write operations.

EEPROM Interrupt

The EEPROM write interrupt is generated when an EEPROM write cycle has ended. The EEPROM interrupt must first be enabled by setting the DEE bit in the relevant interrupt register. However as the EEPROM is contained within a Multi-function Interrupt, the associated multi-function interrupt enable bit must also be set. When an EEPROM write cycle ends, the DEF request flag and its associated multi-function interrupt request flag will both be set. If the global, EEPROM and Multi-function interrupts are enabled and the stack is not full, a jump to the associated Multi-function Interrupt vector will take place. When the interrupt is serviced only the Multi-function interrupt flag will be automatically reset, the EEPROM interrupt flag must be manually reset by the application program. More details can be obtained in the Interrupt section.

Programming Considerations

Care must be taken that data is not inadvertently written to the EEPROM. Protection can be enhanced by ensuring that the Write Enable bit is normally cleared to zero when not writing. Also the Bank Pointer could be normally cleared to zero as this would inhibit access to Bank 1 where the EEPROM control register exist. Although certainly not necessary, consideration might be given in the application program to the checking of the validity of new write data by a simple read back process.

When writing data the WR bit must be set high immediately after the WREN bit has been set high, to ensure the write cycle executes correctly. The global interrupt bit EMI should also be cleared before a write cycle is executed and then re-enabled after the write cycle starts.

Programming Examples

Reading Data from the EEPROM – Polling Method

```
MOV A, EEPROM_ADRES       ; user defined address
MOV EEA, A
MOV  A, 040H              ; setup memory pointer MP1
MOV MP1, A                ; MP1 points to EEC register
MOV A, 01H                ; setup Bank Pointer
MOV BP, A
SET IAR1.1                ; set RDEN bit, enable read operations
SET IAR1.0                ; start Read Cycle - set RD bit
BACK:
SZ  IAR1.0                ; check for read cycle end
JMP BACK
CLR IAR1                  ; disable EEPROM read/write
CLR BP
MOV A, EEDATA             ; move read data to register
MOV READ_DATA, A
```

Writing Data to the EEPROM – Polling Method

```
CLR EMI
MOV A, EEPROM_ADRES       ; user defined address
MOV EEA, A
MOV A, EEPROM_DATA        ; user defined data
MOV EED, A
MOV A, 040H               ; setup memory pointer MP1
MOV MP1, A                ; MP1 points to EEC register
MOV A, 01H                ; setup Bank Pointer
MOV BP, A
SET IAR1.3                ; set WREN bit, enable write operations
SET IAR1.2                ; Start Write Cycle - set WR bit - executed immediately
                          ; after set WREN bit
SET  EMI
BACK:
SZ  IAR1.2                ; check for write cycle end
JMP BACK
CLR IAR1                  ; disable EEPROM read/write
CLR BP
```

Oscillator

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

Oscillator Overview

The devices include two internal oscillators, a low speed oscillator and high speed oscillator. Both can be chosen as the clock source for the main system clock however the slow speed oscillator is also used as a clock source for other functions such as the Watchdog Timer, Time Base and Timer Modules. Both oscillators require no external components for their implementation. All oscillator options are selected using registers. The high speed oscillator provides higher performance but carries with it the disadvantage of higher power requirements, while the opposite is of course true for the low speed oscillator. With the capability of dynamically switching between fast and slow system clock, the device has the flexibility to optimise the performance/power ratio, a feature especially important in power sensitive portable applications.

Type	Name	Freq.
Internal High Speed	HIRC	8, 12 or 16MHz
Internal Low Speed	LIRC	32kHz

Oscillator Types

System Clock Configurations

There are two methods of generating the system clock, a high speed internal clock source and low speed internal clock source. The high speed oscillator is an internal 8MHz, 12MHz or 16MHz RC oscillator while the low speed oscillator is an internal 32kHz RC oscillator. Both oscillators are fully integrated and do not require external components. Selecting whether the low or high speed oscillator is used as the system oscillator is implemented using the HLCLK bit and CKS2 ~ CKS0 bits in the SMOD register allowing the system clock to be dynamically selected.

Internal High Speed RC Oscillator – HIRC

The internal High Speed RC oscillator is a fully integrated system oscillator requiring no external components. The internal RC oscillator has a power on default frequency of 8 MHz but can be selected to be either 8MHz, 12MHz or 16MHz using the HIRCS1 and HIRCS0 bits in the CTRL register. 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.

CTRL Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	HIRCS1	HIRCS0	—	—	D1	D0
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

″x″ unknown

Bit 7, 6 unimplemented, read as ″0″

Bit 5, 4 **HIRCS1**, **HIRCS0**: High frequency clock select
00: 8MHz
01: 16MHz
10: 12MHz
11: 8MHz

Bit 3, 2 unimplemented, read as ″0″

Bit 1, 0 **D1**, **D0**: These bits must be set to the binary value ″00″

Internal Low Speed RC Oscillator – LIRC

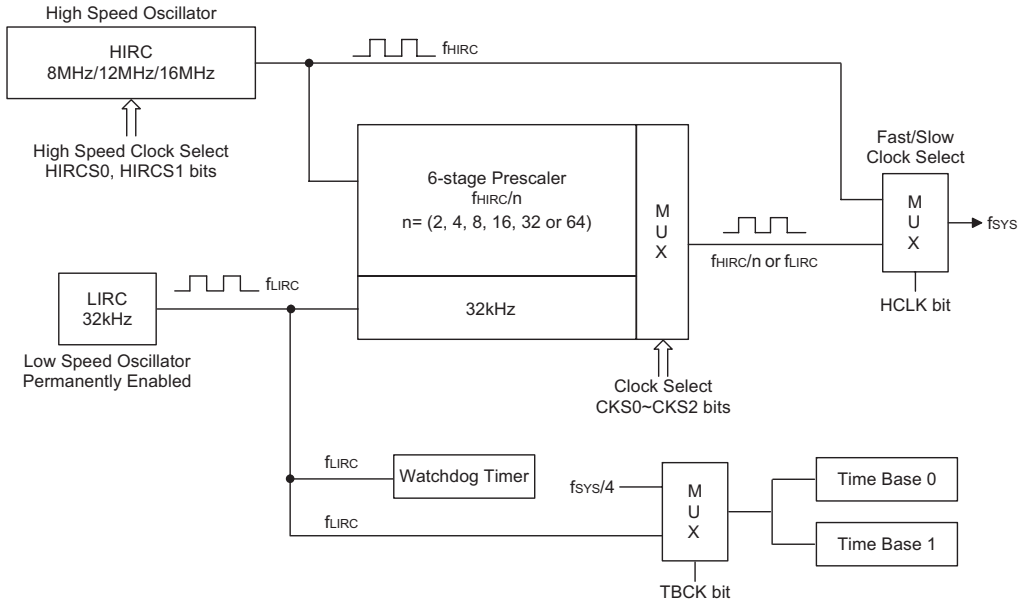
The Internal 32kHz System Oscillator is the low frequency oscillator. It is a fully integrated RC oscillator with a typical frequency of 32kHz at 5V, requiring no external components for its implementation. Device trimming during the manufacturing process and the inclusion of internal frequency compensation circuits are used to ensure that the influence of the power supply voltage, temperature and process variations on the oscillation frequency are minimised. After power on this LIRC oscillator will be permanently enabled; there is no provision to disable the oscillator using register bits.

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 these devices 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 main system clock, can come from either a high frequency, f_H, or low frequency, f_L, source, and is selected using the HLCLK bit and CKS2~CKS0 bits in the SMOD register. Both the high and low speed system clocks are sourced from internal RC oscillators.



System Clock Configurations

Control Register

A single register, SMOD, is used for overall control of the internal clocks within the device.

SMOD Register

Bit	7	6	5	4	3	2	1	0
Name	CKS2	CKS1	CKS0	D4	LTO	HTO	IDLEN	HLCLK
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
POR	0	0	0	0	0	0	1	1

Bit 7~5 **CKS2~CKS0**: The system clock selection when HLCLK is "0"
000: f_L (f_{LIRC})
001: f_L (f_{LIRC})
010: f_H/64
011: f_H/32
100: f_H/16
101: f_H/8
110: f_H/4
111: f_H/2
These three bits are used to select which clock is used as the system clock source. In addition to the system clock source, which is LIRC, a divided version of the high speed system oscillator can also be chosen as the system clock source.

Bit 4 Undefined bit
These bits can be read or written by user software program.

Bit 3 **LTO**: Low speed system oscillator ready flag
0: not ready
1: ready
This is the low speed system oscillator ready flag which indicates when the low speed system oscillator is stable after power on reset.

Bit 2 **HTO**: High speed system oscillator ready flag
0: not ready
1: ready
This is the high speed system oscillator ready flag which indicates when the high speed system oscillator is stable. This flag is cleared to "0" by hardware when the device is powered on and then changes to a high level after the high speed system oscillator is stable. Therefore this flag will always be read as "1" by the application program after device power-on. The flag will be low when in the SLEEP or IDLE0 Mode but after a wake-up has occurred, the flag will change to a high level after 15~16 clock cycles.

Bit 1 **IDLEN**: IDLE Mode control
0: disable
1: enable
This is the IDLE Mode Control bit and determines what happens when the HALT instruction is executed. If this bit is high, when a HALT instruction is executed the device will enter the IDLE Mode. In the IDLE1 Mode the CPU will stop running but the system clock will continue to keep the peripheral functions operational, if FSYSON bit is high. If FSYSON bit is low, the CPU and the system clock will all stop in IDLE0 mode. If the bit is low the device will enter the SLEEP Mode when a HALT instruction is executed.

Bit 0 **HLCLK**: system clock selection
0: f_H/2 ~ f_H/64 or f_L
1: f_H
This bit is used to select if the f_H clock or the f_H/2 ~ f_H/64 or f_L clock is used as the system clock. When the bit is high the f_H clock will be selected and if low the f_H/2 ~ f_H/64 or f_L clock will be selected. When system clock switches from the f_H clock to the f_L clock and the f_H clock will be automatically switched off to conserve power.

System Operation Modes

There are five different modes of operation for the microcontroller, 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 NORMAL Mode and SLOW Mode. The remaining three modes, the SLEEP, IDLE0 and IDLE1 Mode are used when the microcontroller CPU is switched off to conserve power.

Operation Mode	Description			
	CPU	f_{SYS}	f_{LIRC}	f_{TBC}
NORMAL Mode	On	f_H~ f_H/64	On	On
SLOW Mode	On	f_L	On	On
IDLE0 Mode	Off	Off	On	On
IDLE1 Mode	Off	On	On	On
SLEEP Mode	Off	Off	On	Off

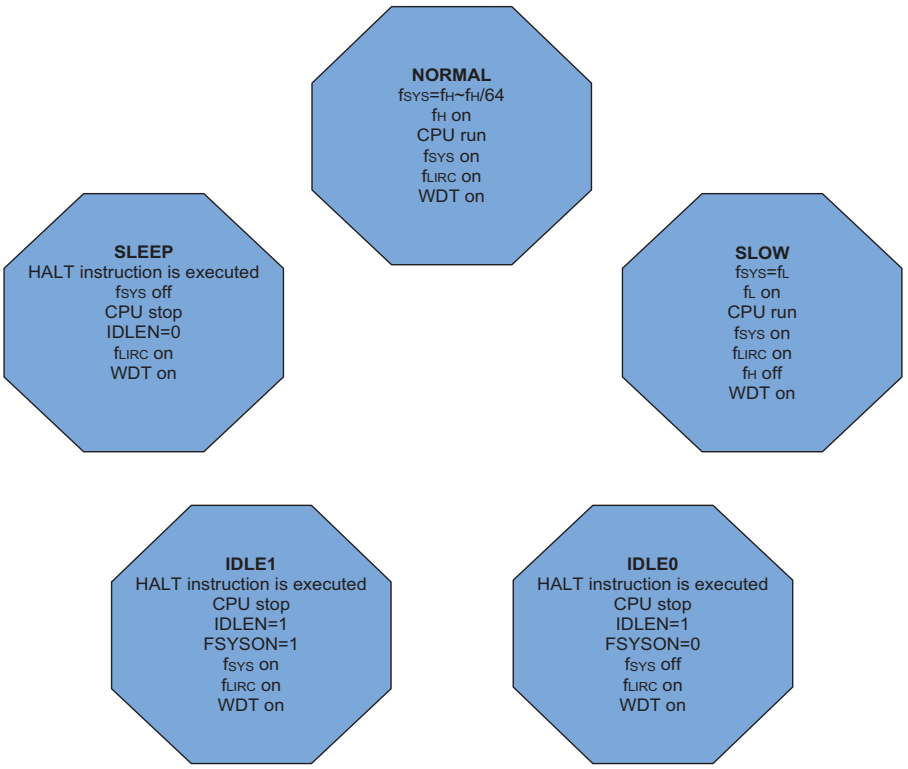
- NORMAL Mode
  As the name suggests this is one of the main operating modes where the microcontroller has all of its functions operational and where the system clock is provided by the high speed oscillator. The high speed oscillator will however first be divided by a ratio ranging from 1 to 64, the actual ratio being selected by the CKS2~LCKS0 and HLCLK bits in the SMOD register. Although a high speed oscillator is used, running the microcontroller at a divided clock ratio reduces the operating current.

- SLOW Mode
  This is also a mode where the microcontroller operates normally although now with the slow speed clock source. Running the microcontroller in this mode allows it to run with much lower operating currents. In the SLOW Mode, the high speed clock is off.

- SLEEP Mode
  The SLEEP Mode is entered when a HALT instruction is executed and when the IDLEN bit in the SMOD register is low. In the SLEEP mode the CPU will be stopped however as the f_{LIRC} oscillator continues to run the Watchdog Timer will continue to operate.

- IDLE0 Mode
  The IDLE0 Mode is entered when a HALT instruction is executed and when the IDLEN bit in the SMOD register is high and the FSYSON bit in the WDTC register is low. In the IDLE0 Mode the system oscillator the system oscillator will be stopped and will therefore be inhibited from driving the CPU.

- IDLE1 Mode
  The IDLE1 Mode is entered when a HALT instruction is executed and when the IDLEN bit in the SMOD register is high and the FSYSON bit in the WDTC register is high. In the IDLE1 Mode the system oscillator will be inhibited from driving the CPU but may continue to provide a clock source to keep some peripheral functions operational. In the IDLE1 Mode, the system oscillator will continue to run, and this system oscillator may be the high speed or low speed system oscillator.

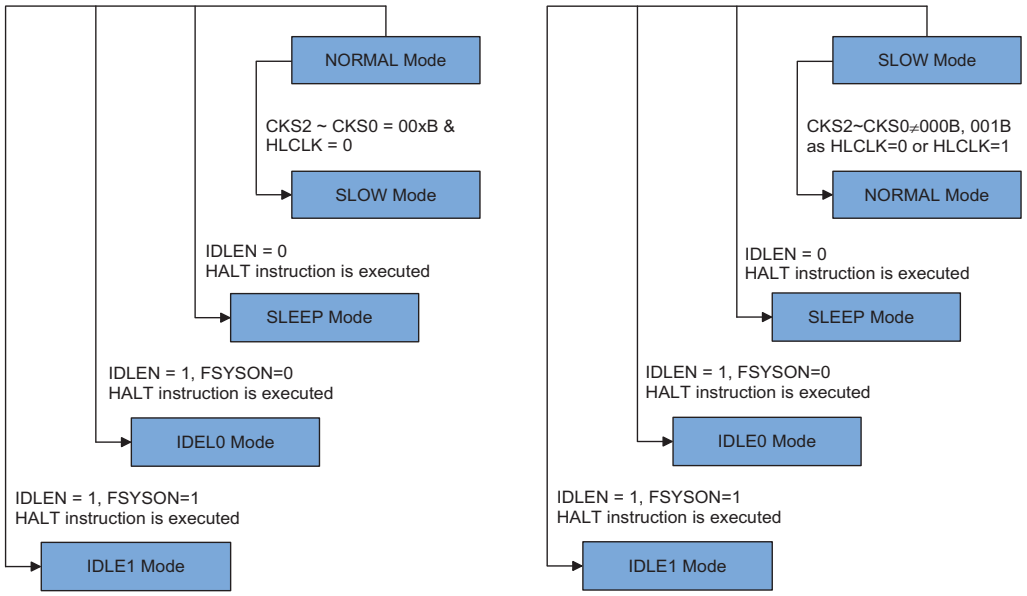


Operating Mode Switching

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 NORMAL Mode and SLOW Mode is executed using the HLCLK bit and CKS2~CKS0 bits in the SMOD register while Mode Switching from the NORMAL/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 IDLEN bit in the SMOD register and FSYSON in the WDTC register.

When the HLCLK bit switches to a low level, which implies that clock source is switched from the high speed clock source, f_{HIRC}, to the clock source, $f_{HIRC}/2$~$f_{HIRC}/64$ or f_{LIRC}. If the clock is from f_{HIRC}, the high speed clock source will stop running to conserve power. When this happens it must be noted that the $f_{HIRC}/16$ and $f_{HIRC}/64$ internal clock sources will also stop running. The accompanying flowchart shows what happens when the device moves between the various operating modes.



NORMAL Mode to SLOW Mode Switching

When running in the NORMAL 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 set the HLCLK bit to ″0″ and set the CKS2~CKS0 bits to ″000″ or ″001″ in the SMOD 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 clock is sourced from the LIRC oscillator.

SLOW Mode to NORMAL Mode Switching

In SLOW Mode the system uses the LIRC low speed system oscillator. To switch back to the NORMAL Mode, where the high speed system oscillator is used, the HLCLK bit should be set to ″1″ or HLCLK bit is ″0″, but CKS2~CKS0 is set to ″010″, ″011″, ″100″, ″101″, ″110″ or ″111″. As a certain amount of time will be required for the high frequency clock to stabilise, the status of the HTO bit is checked. The amount of time required for high speed system oscillator stabilization depends upon which high speed system oscillator type is used.

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 the IDLEN bit in SMOD register equal to ″0″. 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, but the f_{LIRC} clock will be on.
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and resume counting.
- 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.

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 IDLEN bit in SMOD register equal to "1" and the FSYSON bit in WDTC register equal to "0". 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, but the Time Base clock and f_{LIRC} clock will be on.
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and resume counting.
- 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.

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 the IDLEN bit in SMOD register equal to "1" and the FSYSON bit in WDTC register equal to "1". When this instruction is executed under the with conditions described above, the following will occur:

- The system clock and f_{LIRC} clock will be on and the application program will stop at the "HALT" instruction.
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and resume counting.
- 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.

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 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 devices which have different package types, as there may be unbonbed pins. These must either be setup as outputs or if setup 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.

In the IDLE1 Mode the system oscillator is on, if the system oscillator is from the high speed system oscillator, the additional standby current will also be perhaps in the order of several hundred micro-amps.

Wake-up

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

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

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

Each pin on Port A can be setup using the PAWU register to permit a negative transition on the pin to wake-up the system. When a Port 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.

System Oscillator	Wake-up Time (SLEEP Mode)	Wake-up Time (IDLE0 Mode)	Wake-up Time (IDLE1 Mode)
HIRC	15~16 HIRC cycles		1~2 HIRC cycles
LIRC	1~2 LIRC cycles		1~2 LIRC cycles

Wake-Up Times

Programming Considerations

The high speed and low speed oscillators both use the same SST counter. For example, if the system is woken up from the SLEEP Mode the HIRC oscillator needs to start-up from an off state.

- If the device is woken up from the SLEEP Mode to the NORMAL Mode, the high speed system oscillator needs an SST period. The device will execute the first instruction after HTO is high.

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 low speed oscillator, f_{LIRC}. The Watchdog Timer source clock is then subdivided by a ratio of 2^8 to 2^{15} to give longer timeouts, the actual value being chosen using the WS2~WS0 bits in the WDTC register. The LIRC internal oscillator has an approximate period of 32kHz at a supply voltage of 5V.

However, it should be noted that this specified internal clock period can vary with VDD, temperature and process variations.

Watchdog Timer Control Register

A single register, WDTC, controls the required timeout period.

WDTC Register

Bit	7	6	5	4	3	2	1	0
Name	FSYSON	WS2	WS1	WS0	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	1	1	1	0	1	0

Bit 7 **FSYSON**: f_{SYS} control in IDLE Mode
0: disable
1: enable

Bit 6~4 **WS2, WS1, WS0** : WDT time-out period selection
000: 256/f_{LIRC}
001: 512/f_{LIRC}
010: 1024/f_{LIRC}
011: 2048/f_{LIRC}
100: 4096/f_{LIRC}
101: 8192/f_{LIRC}
110: 16384/f_{LIRC}
111: 32768/f_{LIRC}
These three bits determine the division ratio of the Watchdog Timer source clock, which in turn determines the timeout period.

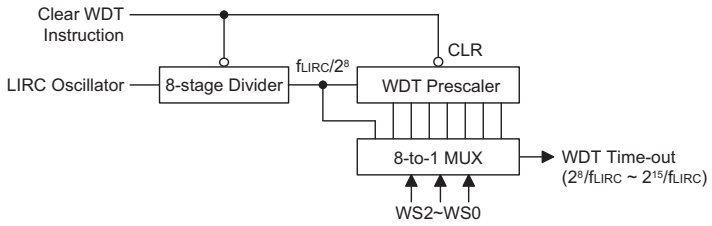
Bit 3~0 Undefined bit
These bits can be read or written by user software program.

Watchdog Timer Operation

In these devices the Watchdog Timer supplied by the f_{LIRC} oscillator and is therefore always on. 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 instructions. If the program malfunctions for whatever reason, jumps to an unkown location, or enters an endless loop, these clear instructions will not be executed in the correct manner, in which case the Watchdog Timer will overflow and reset the device.

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 an external hardware reset, the second is using the Watchdog Timer software clear instructions and the third is via a HALT instruction. The Watchdog Timer is cleared using a single CLR WDT instruction.

The maximum time out period is when the 2^{15} division ratio is selected. As an example, with the LIRC oscillator as its source clock, this will give a maximum watchdog period of around 1 second for the 2^{15} division ratio, and a minimum timeout of 7.8ms 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 microcontroller. In this case, internal circuitry will ensure that the microcontroller, after a short delay, will be in a well defined state and ready to execute the first program instruction. After this power-on reset, certain important internal registers will be set to defined states before the program commences. One of these registers is the Program Counter, which will be reset to zero forcing the microcontroller to begin program execution from the lowest Program Memory address.

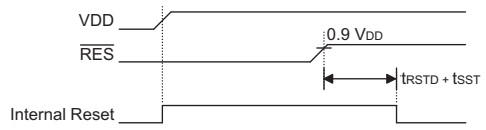
Another type of reset is when the Watchdog Timer overflows and resets the microcontroller. All types of reset operations result in different register conditions being setup. Another reset exists in the form of a Low Voltage Reset, LVR, where a full reset, is implemented in situations where the power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

Reset Functions

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

Power-on Reset

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

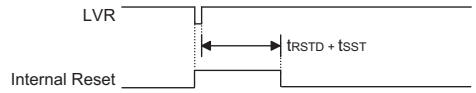


Note: t_{RSTD} is power-on delay, typical time=100ms

Power-On Reset Timing Chart

Low Voltage Reset – 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, the LVR will automatically reset the device internally. The LVR includes the following specifications: For a valid LVR signal, a low voltage, i.e., a voltage in the range between 0.9V~V_{LVR} must exist for greater than the value t_{LVR} specified in the A.C. characteristics. If the low voltage state does not exceed t_{LVR}, the LVR will ignore it and will not perform a reset function. One of a range of specified voltage values for V_{LVR} can be selected using configuration options. The LVR function is permanently on in these devices.

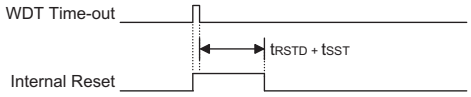


Note: t_{RSTD} is power-on delay, typical time=100ms

Low Voltage Reset Timing Chart

Watchdog Time-out Reset during Normal Operation

The Watchdog time-out Reset during normal operation is the same as a hardware power-on reset except that the Watchdog time-out flag TO will be set to "1".



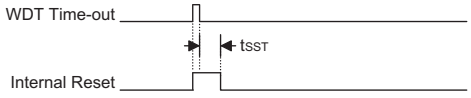
Note: t_{RSTD} is power-on delay, typical time=100ms

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 flag will be set to "1". Refer to the A.C. Characteristics for t_{SST} details.

Note: The t_{SST} is 15~16 clock cycles if the system clock source is provided by HIRC. The t_{SST} is 1~2 clock for LIRC.



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 NORMAL or SLOW Mode operation
1	u	WDT time-out reset during NORMAL or SLOW Mode operation
1	1	WDT time-out reset during IDLE or SLEEP Mode operation

Note: "u" stands for unchanged

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

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

The different kinds of resets all affect the internal registers of the microcontroller in different ways. To ensure reliable continuation of normal program 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. Note that where more than one package type exists the table will reflect the situation for the larger package type.

Register	BS85B12-3	BS85C20-3	Power-on Reset	LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE or SLEEP)
MP0	•	•	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1	•	•	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
BP	•		----- ---0	----- ---0	----- ---0	----- ---u
BP		•	----- --00	----- --00	----- --00	----- --uu
ACC	•	•	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	•	•	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	•	•	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	•	•	-xxx xxxx	-uuu uuuu	-uuu uuuu	-uuu uuuu
TBHP	•		----- -xxx	----- -uuu	----- -uuu	----- -uuu
TBHP		•	----- xxxx	----- uuuu	----- uuuu	----- uuuu
STATUS	•	•	--00 xxxx	--uu uuuu	--1u uuuu	--11 uuuu
SMOD	•	•	0000 0011	0000 0011	0000 0011	uuuu uuuu
LVDC	•	•	--00 -000	--00 -000	--00 -000	--uu -uuu
INTEG	•	•	---- 0000	---- 0000	---- 0000	---- uuuu
WDTC	•	•	0111 1010	0111 1010	0111 1010	uuuu uuuu
TBC	•	•	0011 0111	0011 0111	0011 0111	uuuu uuuu
INTC0	•	•	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
INTC2	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
INTC3		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI0	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI1	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI2	•	•	-000 -000	-000 -000	-000 -000	-uuu -uuu
MFI3	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PAWU	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PAPU	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PA	•	•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	•	•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBPU	•		--00 0000	--00 0000	--00 0000	--uu uuuu
PBPU		•	0000 0000	0000 0000	0000 0000	uuuu uuuu

Register	BS85B12-3	BS85C20-3	Power-on Reset	LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE or SLEEP)
PB	•		--11 1111	--11 1111	--11 1111	--uu uuuu
PB		•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBC	•		--11 1111	--11 1111	--11 1111	--uu uuuu
PBC		•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PCPU	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PC	•	•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PCC	•	•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PDPU		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PD		•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PDC		•	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEPU		•	--00 0000	--00 0000	--00 0000	--uu uuuu
PE		•	--11 1111	--11 1111	--11 1111	--uu uuuu
PEC		•	--11 1111	--11 1111	--11 1111	--uu uuuu
SLCDC0	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
SLCDC1	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
SLCDC2	•		--00 0000	--00 0000	--00 0000	--uu uuuu
SLCDC2		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
SLCDC3		•	0-00 0000	0-00 0000	0-00 0000	u-uu uuuu
SLEDC0	•		--00 0000	--00 0000	--00 0000	--uu uuuu
SLEDC0		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
SLEDC1	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
SLEDC2		•	--00 0000	--00 0000	--00 0000	--uu uuuu
MFI4		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI5		•	--00 --00	--00 --00	--00 --00	--uu --uu
I2CTOC	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
SIMC0	•	•	1110 000-	1110 000-	1110 000-	uuuu uuu-
SIMC1	•	•	1000 0001	1000 0001	1000 0001	uuuu uuuu
SIMD	•	•	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
SIMA/SIMC2	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0C0	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0C1	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0DL	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0DH	•	•	---- --00	---- --00	---- --00	---- --uu
TM0AL	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu

Register	BS85B12-3	BS85C20-3	Power-on Reset	LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE or SLEEP)
TM0AH	•	•	---- --00	---- --00	---- --00	---- --uu
EEA	•		--00 0000	--00 0000	--00 0000	--uu uuuu
EEA		•	-000 0000	-000 0000	-000 0000	-uuu uuuu
EED	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TMPC0	•	•	1001 --01	1001 --01	1001 --01	uuuu --uu
TMPC1		•	---- --01	---- --01	---- --01	---- --uu
PRM0	•		-0-0 -0-0	-0-0 -0-0	-0-0 -0-0	-u-u -u-u
PRM0		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PRM1	•		0000 -0-0	0000 -0-0	0000 -0-0	uuuu -u-u
PRM1		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
PRM2	•		0000 --00	0000 --00	0000 --00	uuuu --uu
PRM2		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1C0	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1C1	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1C2	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1DL	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1DH	•	•	---- --00	---- --00	---- --00	---- --uu
TM1AL	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1AH	•	•	---- --00	---- --00	---- --00	---- --uu
TM1BL	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1BH	•	•	---- --00	---- --00	---- --00	---- --uu
TM2C0		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2C1		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2DL		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2DH		•	---- --00	---- --00	---- --00	---- --uu
TM2AL		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2AH		•	---- --00	---- --00	---- --00	---- --uu
CTRL	•	•	--00 --00	--00 --00	--00 --00	--uu --uu
TKM016DH	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM016DL	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM0C0	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM0C1	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM0C2	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM0C3	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu

Register	BS85B12-3	BS85C20-3	Power-on Reset	LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE or SLEEP)
TKM116DH	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM116DL	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM1C0	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM1C1	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM1C2	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM1C3	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM216DH	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM216DL	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM2C0		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM2C1	•	•	0000 ----	0000 ----	0000 ----	uuuu ----
TKM2C2	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM2C3	•	•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM316DH		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM316DL		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM3C0		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM3C1		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM3C2		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM3C3		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM416DH		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM416DL		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM4C0		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM4C1		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM4C2		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
TKM4C3		•	0000 0000	0000 0000	0000 0000	uuuu uuuu
EEC	•	•	---- 0000	---- 0000	---- 0000	---- uuuu

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~PE. 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.

I/O Register List

BS85B12-3

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAWU	D7	D6	D5	D4	D3	D2	D1	D0
PAPU	D7	D6	D5	D4	D3	D2	D1	D0
PA	D7	D6	D5	D4	D3	D2	D1	D0
PAC	D7	D6	D5	D4	D3	D2	D1	D0
PBPU	—	—	D5	D4	D3	D2	D1	D0
PB	—	—	D5	D4	D3	D2	D1	D0
PBC	—	—	D5	D4	D3	D2	D1	D0
PCPU	D7	D6	D5	D4	D3	D2	D1	D0
PC	D7	D6	D5	D4	D3	D2	D1	D0
PCC	D7	D6	D5	D4	D3	D2	D1	D0

BS85C20-3

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAWU	D7	D6	D5	D4	D3	D2	D1	D0
PAPU	D7	D6	D5	D4	D3	D2	D1	D0
PA	D7	D6	D5	D4	D3	D2	D1	D0
PAC	D7	D6	D5	D4	D3	D2	D1	D0
PBPU	D7	D6	D5	D4	D3	D2	D1	D0
PB	D7	D6	D5	D4	D3	D2	D1	D0
PBC	D7	D6	D5	D4	D3	D2	D1	D0
PCPU	D7	D6	D5	D4	D3	D2	D1	D0
PC	D7	D6	D5	D4	D3	D2	D1	D0
PCC	D7	D6	D5	D4	D3	D2	D1	D0
PDPU	D7	D6	D5	D4	D3	D2	D1	D0
PD	D7	D6	D5	D4	D3	D2	D1	D0
PDC	D7	D6	D5	D4	D3	D2	D1	D0
PEPU	—	—	D5	D4	D3	D2	D1	D0
PE	—	—	D5	D4	D3	D2	D1	D0
PEC	—	—	D5	D4	D3	D2	D1	D0

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 an input have the capability of being connected to an internal pull-high resistor. These pull-high resistors are selected using the register PAPU~PEPU, and are implemented using weak PMOS transistors.

BS85B12-3: PAPU, PCPU Registers
BS85C20-3: PAPU, PBPU, PCPU, PDPU Registers

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 PxPU Port bit 7~bit 0 Pull-High control
0: disable
1: enable

BS85B12-3: PBPU Registers
BS85C20-3: PEPU Registers

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

Bit 7~4 unimplemented, read as "0"
Bit 3~0 **PxPU**: Port bit 5~bit 0 Pull-High control
0: disable
1: enable

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.

PAWU 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 7~5 unimplemented, read as "0"

Bit 4~0 PAWU: Port A bit 7~bit 0 wake-up control
0: disable
1: enable

I/O Port Control Register

The I/O port has its own control register known as PAC~PEC, 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 port 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.

BS85B12-3: PAC, PCC Registers
BS85C20-3: PAC, PBC, PCC, PDC Registers

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	1	1	1	1	1	1	1	1

Bit 7~0 I/O Port bit 7 ~ bit 0 input/output control
0: output
1: input

BS85B12-3: PBC Registers
BS85C20-3: PEC Registers

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

Bit 7~4 unimplemented, read as "0"

Bit 3~0 **PxC**: Port bit 5~bit 0 input/output control
0: output
1: input

Pin Re-mapping 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. The way in which the pin function of each pin is selected is different for each function and a priority order is established where more than one pin function is selected simultaneously. Additionally there are a series of PRM0, PRM1 and PRM2 registers to establish certain pin functions.

Pin-remapping 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 PRM0, PRM1, PRM2 registers which can select the functions of certain pins.

Pin-remapping Register List

- BS85B12-3

Register Name	Bit							
	7	6	5	4	3	2	1	0
PRM0	—	SCSPS0	—	SDIPS0	—	SCKPS0	—	SDOPS0
PRM1	INT1PS	INT0PS	TCK1PS	TCK0PS	—	PINTS0	—	PCKPS0
PRM2	TP1B2PS	TP1B1PS	TP1B0PS	TP1APS	—	—	TP01PS	TP00PS

- BS85C20-3

Register Name	Bit							
	7	6	5	4	3	2	1	0
PRM0	SCSPS1	SCSPS0	SDIPS1	SDIPS0	SCKPS1	SCKPS0	SDOPS1	SDOPS0
PRM1	INT1PS	INT0PS	TCK1PS	TCK0PS	PINTS1	PINTS0	PCKPS1	PCKPS0
PRM2	TP1B2PS	TP1B1PS	TP1B0PS	TP1APS	TP21PS	TP20PS	TP01PS	TP00PS
SLCDC3	TCK2PS	—	SEG21EN	SEG20EN	SEG19EN	SEG18EN	SEG17EN	SEG16EN

PRM0 Register -- BS85B12-3

Bit	7	6	5	4	3	2	1	0
Name	—	SCSPS0	—	SDIPS0	—	SCKPS0	—	SDOPS0
R/W	—	R/W	—	R/W	—	R/W	—	R/W
POR	—	0	—	0	—	0	—	0

Bit 7 unimplemented, read as "0"

Bit 6 **SCSPS0**: $\overline{SCS}$ pin remapping control
0: $\overline{SCS}$ on PA3
1: $\overline{SCS}$ on PC3

Bit 5 unimplemented, read as "0"

Bit 4 **SDIPS0**: SDI/SDA pin remapping control
0: SDI/SDA on PA0
1: SDI/SDA on PC2

Bit 3 unimplemented, read as "0"

Bit 2 **SCKPS0**: SCK/SCL pin remapping control
0: SCK/SCL on PA2
1: SCK/SCL on PC1

Bit 1 unimplemented, read as "0"

Bit 0 **SDOPS0**: SDO pin remapping control
0: SDO on PA7
1: SDO on PC0

PRM0 Register -- BS85C20-3

Bit	7	6	5	4	3	2	1	0
Name	SCSPS1	SCSPS0	SDIPS1	SDIPS0	SCKPS1	SCKPS0	SDOPS1	SDOPS0
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 **SCSPS1~SCSPS0**: $\overline{SCS}$ pin remapping control
00: $\overline{SCS}$ on PA3
01: $\overline{SCS}$ on PC3
10: $\overline{SCS}$ on PE2
11: undefined

Bit 5~4 **SDIPS1~SDIPS0**: SDI/SDA pin remapping control
00: SDI/SDA on PA0
01: SDI/SDA on PC2
10: SDI/SDA on PE4
11: undefined

Bit 3~2 **SCKPS1~SCKPS0**: SCK/SCL pin remapping control
00: SCK/SCL on PA2
01: SCK/SCL on PC1
10: SCK/SCL on PE3
11: undefined

Bit 1~0 **SDOPS1~SDOPS0**: SDO pin remapping control
00: SDO on PA7
01: SDO on PC0
10: SDO on PE5
11: undefined

PRM1 Register -- BS85B12-3

Bit	7	6	5	4	3	2	1	0
Name	INT1PS	INT0PS	TCK1PS	TCK0PS	—	PINTS0	—	PCKPS0
R/W	R/W	R/W	R/W	R/W	—	R/W	—	R/W
POR	0	0	0	0	—	0	—	0

Bit 7 **INT1PS**: INT1 pin remapping control
0: INT1 on PA1
1: INT1 on PC5

Bit 6 **INT0PS**: INT0 pin remapping control
0: INT0 on PA4
1: INT0 on PC4

Bit 5 **TCK1PS**: TCK1 pin remapping control
0: TCK1 on PA1
1: TCK1 on PC5

Bit 4 **TCK0PS**: TCK0 pin remapping control
0: TCK0 on PA4
1: TCK0 on PC4

Bit 3 unimplemented, read as "0"

Bit 2 **PINTS0**: $\overline{\text{PINT}}$ pin remapping control
0: PINT on PB5
1: PINT on PC7

Bit 1 unimplemented, read as "0"

Bit 0 **PCKPS0**: PCK pin remapping control
0: PCK on PB4
1: PCK on PC6

PRM1 Register -- BS85C20-3

Bit	7	6	5	4	3	2	1	0
Name	INT1PS	INT0PS	TCK1PS	TCK0PS	PINTS1	PINTS0	PCKPS1	PCKPS0
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 **INT1PS**: INT1 pin remapping control
0: INT1 on PA1
1: INT1 on PC5

Bit 6 **INT0PS**: INT0 pin remapping control
0: INT0 on PA4
1: INT0 on PC4

Bit 5 **TCK1PS**: TCK1 pin remapping control
0: TCK1 on PA1
1: TCK1 on PC5

Bit 4 **TCK0PS**: TCK0 pin remapping control
0: TCK0 on PA4
1: TCK0 on PC4

Bit 3~2 **PINTS1~PINTS0**: $\overline{\text{PINT}}$ pin remapping control
00: $\overline{\text{PINT}}$ on PB5
01: $\overline{\text{PINT}}$ on PC7
10: $\overline{\text{PINT}}$ on PE0
11: undefined

Bit 1~0 **PCKPS1~PCKPS0**: PCK pin remapping control
00: PCK on PB4
01: PCK on PC6
10: PCK on PE1
11: undefined

PRM2 Register -- BS85B12-3

Bit	7	6	5	4	3	2	1	0
Name	TP1B2PS	TP1B1PS	TP1B0PS	TP1APS	—	—	TP01PS	TP00PS
R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W
POR	0	0	0	0	—	—	0	0

Bit 7 **TP1B2PS**: TP1B_2 pin remapping cControl
0: TP1B_2 on PB4
1: TP1B_2 on PC4

Bit 6 **TP1B1PS**: TP1B_1 pin remapping control
0: TP1B_1 on PB3
1: TP1B_1 on PC3

Bit 5 **TP1B0PS**: TP1B_0 pin remapping control
0: TP1B_0 on PB2
1: TP1B_0 on PC2

Bit 4 **TP1APS**: TP1A pin remapping control
0: TP1A on PB5
1: TP1A on PC5

Bit 3~2 unimplemented, read as ″0″

Bit 1 TP01PS: TP0_1 Pin Remapping Control
0: TP0_1 on PB1
1: TP0_1 on PC1

Bit 0 **TP00PS**: TP0_0 Pin Remapping Control
0: TP0_0 on PB0
1: TP0_0 on PC0

PRM2 Register -- BS85C20-3

Bit	7	6	5	4	3	2	1	0
Name	TP1B2PS	TP1B1PS	TP1B0PS	TP1APS	TP21PS	TP20PS	TP01PS	TP00PS
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 **TP1B2PS**: TP1B_2 pin remapping control
0: TP1B_2 on PB4
1: TP1B_2 on PC4

Bit 6 **TP1B1PS**: TP1B_1 pin remapping control
0: TP1B_1 on PB3
1: TP1B_1 on PC3

Bit 5 **TP1B0PS**: TP1B_0 pin remapping control
0: TP1B_0 on PB2
1: TP1B_0 on PC2

Bit 4 **TP1APS**: TP1A pin remapping coontrol
0: TP1A on PB5
1: TP1A on PC5

Bit 3 **TP21PS**: TP2_1 pin remapping coontrol
0: TP2_1 on PB2
1: TP2_1 on PE2

Bit 2 **TP20PS**: TP2_0 pin remapping coontrol
0: TP2_0 on PB1
1: TP2_0 on PE1

Bit 1 **TP01PS**: TP0_1 pin remapping coontrol
0: TP0_1 on PB1
1: TP0_1 on PC1

Bit 0 **TP00PS**: TP0_0 pin remapping coontrol
0: TP0_0 on PB0
1: TP0_0 on PC0

SLCDC3 Register -- BS85C20-3

Bit	7	6	5	4	3	2	1	0
Name	TCK2PS	—	SEG21EN	SEG20EN	SEG19EN	SEG18EN	SEG17EN	SEG16EN
R/W	R/W	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	—	0	0	0	0	0	0

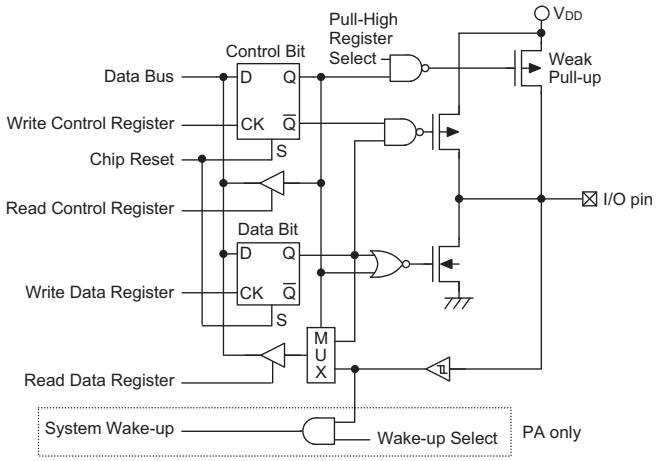
Bit 7 TCK2PS: TCK2 pin remapping control
0: TCK2 on PC6
1: TCK2 on PD0

Bit 6 unimplemented, read as ″0″

Bit 5~0 described elsewhere

I/O Pin Structures

The accompanying diagrams illustrate the internal structures of some generic I/O pin types. As the exact logical construction of the I/O pin will differ from these drawings, they are supplied as a guide only to assist with the functional understanding of the I/O pins. The wide range of pin-shared structures does not permit all types to be shown.



Generic 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 register will be set high. This means that all I/O pins will default to an input state, the level of which depends on the other connected circuitry and whether pull-high selections have been chosen. If the port control register, PAC~PEC, is then programmed to setup some pins as outputs, these output pins will have an initial high output value unless the associated port data register, PA~PE, is 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 functions. 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 setup to have this function.

Timer Modules – TM

One of the most fundamental functions in any microcontroller device is the ability to control and measure time. To implement time related functions each device includes several Timer Modules, abbreviated to the name TM. The TMs are multi-purpose timing units and serve to provide operations such as Timer/Counter, Input Capture, 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 either two or three individual 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, Standard and Enhanced TM sections.

Introduction

The devices contain from two to three TMs depending upon which device is selected with each TM having a reference name of TM0, TM1 and TM2. Each individual TM can be categorised as a certain type, namely Compact Type TM, Standard Type TM or Enhanced Type TM. Although similar in nature, the different TM types vary in their feature complexity. The common features to all of the Compact, Standard and Enhanced TMs will be described in this section, the detailed operation regarding each of the TM types will be described in separate sections. The main features and differences between the three types of TMs are summarised in the accompanying table.

Function	CTM	STM	ETM
Timer/Counter	√	√	√
I/P Capture	—	√	√
Compare Match Output	√	√	√
PWM Channels	1	1	2
Single Pulse Output	—	1	2
PWM Alignment	Edge	Edge	Edge & Centre
PWM Adjustment Period & Duty	Duty or Period	Duty or Period	Duty or Period

TM Function Summary

Each device in the series contains a specific number of either Compact Type, Standard Type and Enhanced Type TM units which are shown in the table together with their individual reference name, TM0~TM2.

Device	TM0	TM1	TM2
BS85B12-3	10-bit CTM	10-bit ETM	—
BS85C20-3	10-bit CTM	10-bit ETM	10-bit STM

TM Name/Type Reference

TM Operation

The three 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 counter whose value is then compared with the value of pre-programmed internal comparators. When the free running 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 TnCK2~TnCK0 bits in the TM control registers. The clock source can be a ratio of either the system clock f_{SYS} or the internal high clock f_H, the f_{TBC} clock source or the external TCKn pin. Note that setting these bits to the value 101 will select a reserved clock input, in effect disconnecting the TM clock source. The TCKn pin clock source is used to allow an external signal to drive the TM as an external clock source or for event counting.

TM Interrupts

The Compact and Standard type TMs each have two internal interrupts, one for each of the internal comparator A or comparator P, which generate a TM interrupt when a compare match condition occurs. As the Enhanced type TM has three internal comparators and comparator A or comparator B or comparator P compare match functions, it consequently has three internal interrupts. 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

Each of the TMs, irrespective of what type, has one TM input pin, with the label TCKn. The TM input pin, is essentially a clock source for the TM and is selected using the TnCK2~TnCK0 bits in the TMnC0 register. This external TM input pin allows an external clock source to drive the internal TM. This external TM input pin is shared with other functions but will be connected to the internal TM if selected using the TnCK2~TnCK0 bits. The TM input pin can be chosen to have either a rising or falling active edge.

The TMs each have one or more output pins with the label TPn. 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 TPn output pin is also the pin where the TM generates the PWM output waveform. As the TM output pins are pin-shared with other function, the TM output function must first be setup using registers. A single bit in one of the registers determines if its associated pin is to be used as an external TM output pin or if it is to have another function. The number of output pins for each TM type and device is different, the details are provided in the accompanying table.

All TM output pin names have an ″_n″ suffix. Pin names that include a ″_1″ or ″_2″ suffix indicate that they are from a TM with multiple output pins. This allows the TM to generate a complimentary output pair, selected using the I/O register data bits.

Device	CTM	STM	ETM	Registers
BS85B12-3	TP0_0, TP0_1	—	TP1A, TP1B_0, TP1B_1, TP1B_2	TMPC0
BS85C20-3	TP0_0, TP0_1	TP2_0, TP2_1	TP1A, TP1B_0, TP1B_1, TP1B_2	TMPC0, TMPC1

TM Output Pins

TM Input/Output Pin Control Registers

Selecting to have a TM input/output or whether to retain its other shared function, is implemented using one or two registers, with a single bit in each register corresponding to a TM input/output pin. Setting the bit high will setup the corresponding pin as a TM input/output, if reset to zero the pin will retain its original other function.

Registers	Device	Bit							
		7	6	5	4	3	2	1	0
TMPC0	All	T1ACP0	T1BCP2	T1BCP1	T1BCP0	—	—	T0CP1	T0CP0
TMPC0	BS85C20-3	—	—	—	—	—	—	T2CP1	T2CP0

TM Input/Output Pin Control Registers List

TMPC0 Register -- All devices

Bit	7	6	5	4	3	2	1	0
Name	T1ACP0	T1BCP2	T1BCP1	T1BCP0	—	—	T0CP1	T0CP0
R/W	R/W	R/W	R/W	R/W	—	—	R/W	R/W
POR	1	0	0	1	—	—	0	1

Bit 7 **T1ACP0**: TP1A pin Control
0: disable
1: enable

Bit 6 **T1BCP2**: TP1B_2 pin Control
0: disable
1: enable

Bit 5 **T1BCP1**: TP1B_1 pin Control
0: disable
1: enable

Bit 4 **T1BCP0**: TP1B_0 pin Control
0: disable
1: enable

Bit 3~2 Unimplemented, read as "0"

Bit 1 **T0CP1**: TP0_1 pin Control
0: disable
1: enable

Bit 0 **T0CP0**: TP0_0 pin Control
0: disable
1: enable

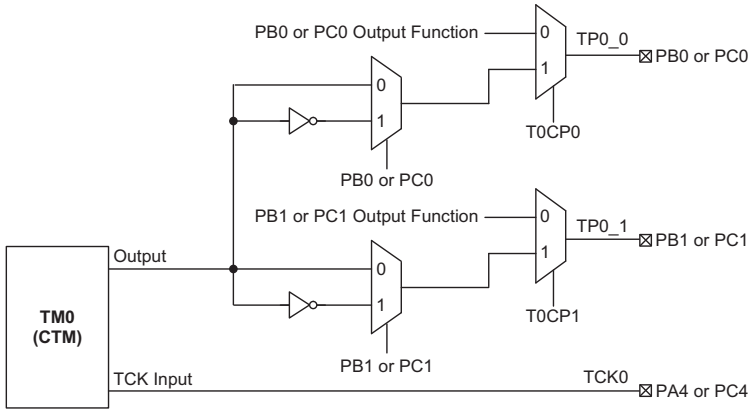
TMPC1 Register -- BS85C20-3

Bit	7	6	5	4	3	2	1	0
Name	—	—			—	—	T2CP1	T2CP0
R/W	—	—			—	—	R/W	R/W
POR	—	—			—	—	0	1

Bit 7~2 Unimplemented, read as "0"

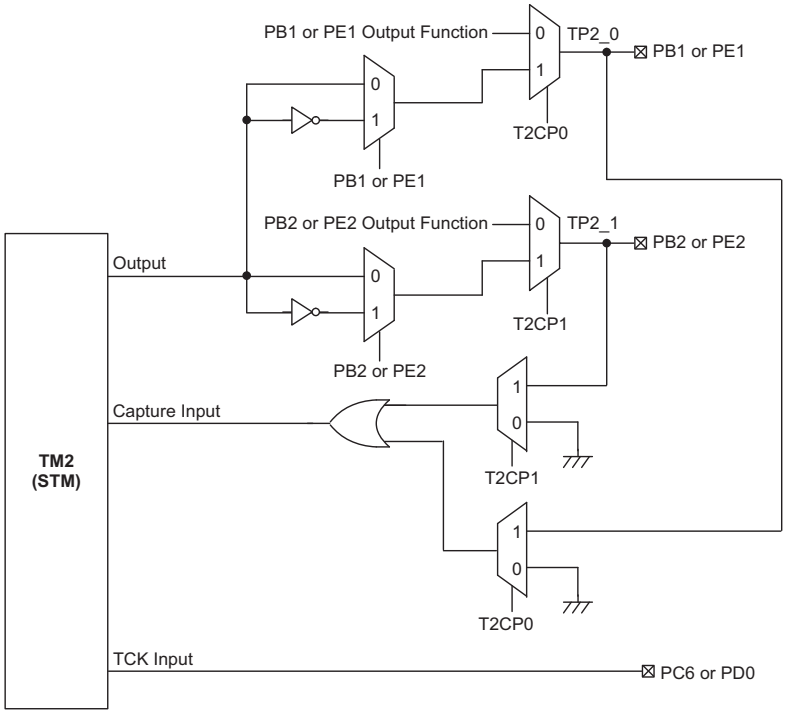
Bit 1 **T2CP1**: TP2_1 pin control
0: disable
1: enable

Bit 0 **T2CP0**: TP2_0 pin control
0: disable
1: enable



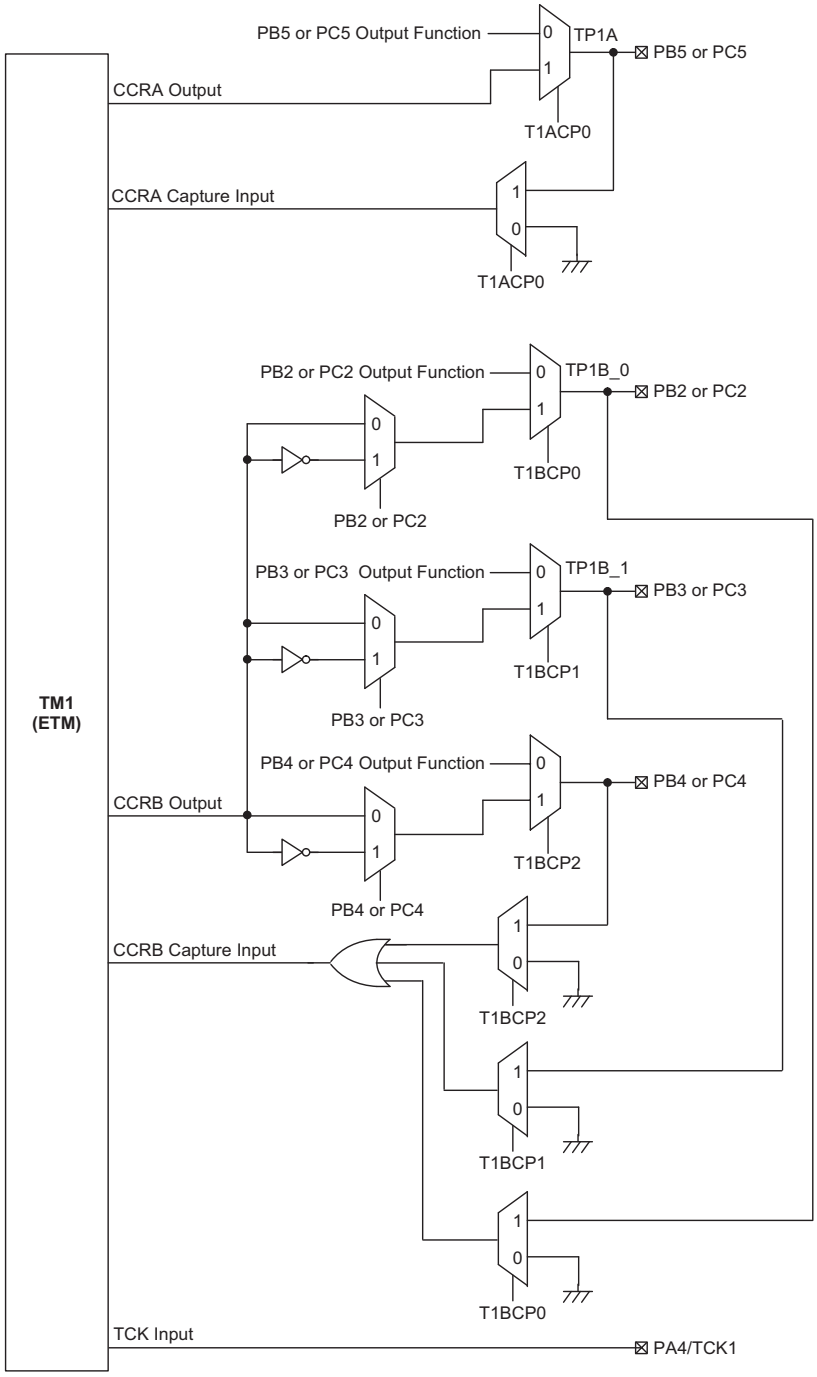
TM0 Function Pin Control Block Diagram

Note: 1. The I/O register data bits shown are used for TM output inversion control.
2. In the Capture Input Mode, the TM pin control register must never enable more than one TM input.



TM2 Function Pin Control Block Diagram -- BS85C20-3 only

Note: 1. The I/O register data bits shown are used for TM output inversion control.
2. In the Capture Input Mode, the TM pin control register must never enable more than one TM input.

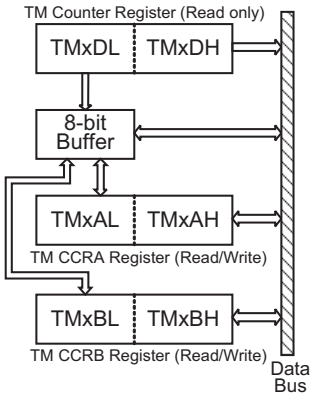


TM1 Function Pin Control Block Diagram

Note: 1. The I/O register data bits shown are used for TM output inversion control.
2. In the Capture Input Mode, the TM pin control register must never enable more than one TM input.

Programming Considerations

The TM Counter Registers and the Capture/Compare CCRA and CCRB registers, being either 10-bit or 16-bit, 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 CCRB 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 CCRB low byte registers, named TMxAL and TMxBL, using the following access procedures. Accessing the CCRA or CCRB 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 CCRB or CCRA
  - Step 1. Write data to Low Byte TMxAL or TMxBL
    - note that here data is only written to the 8-bit buffer.
  - Step 2. Write data to High Byte TMxAH or TMxBH
    - 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 and CCRB or CCRA
  - Step 1. Read data from the High Byte TMxDH, TMxAH or TMxBH
    - 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 TMxDL, TMxAL or TMxBL
    - this step reads data from the 8-bit buffer.

Compact Type TM – CTM

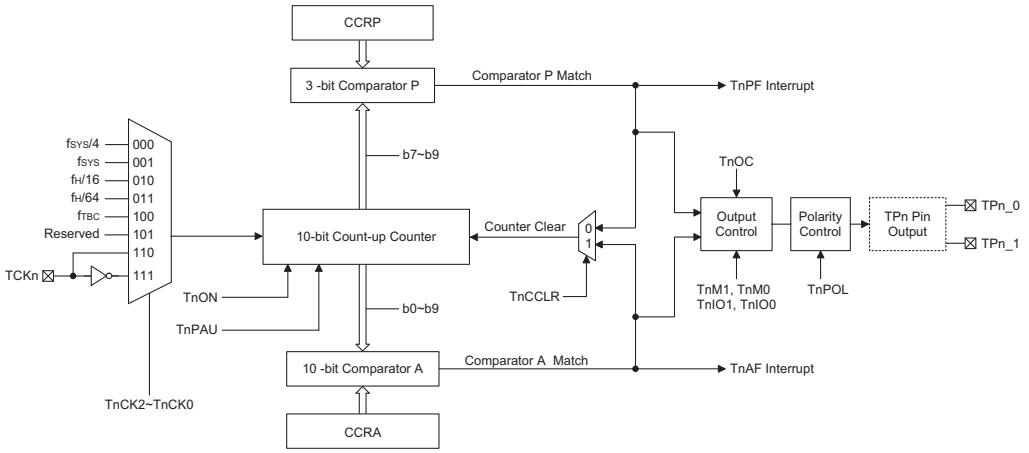
Although the simplest form of the three TM types, the Compact TM type still contains three operating modes, which are Compare Match Output, Timer/Event Counter and PWM Output modes. The Compact TM can also be controlled with an external input pin and can drive one or two external output pins. These two external output pins can be the same signal or the inverse signal.

CTM	Name	TM No.	TM Input Pin	TM Output Pin
BS85B12-3	10-bit CTM	0	TCK0	TP0_0, TP0_1
BS85C20-3	10-bit CTM	0	TCK0	TP0_0, TP0_1

Compact TM Operation

At 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 is three bits wide whose value is compared with the highest three bits in the counter while the CCRA is the ten bits and therefore compares 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 TnON 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 TM 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 an output pin. All operating setup conditions are selected using relevant internal registers.



Compact Type TM Block Diagram

Compact Type TM Register Description

Overall operation of the Compact TM is controlled using six 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 the three CCRP bits.

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMnC0	TnPAU	TnCK2	TnCK1	TnCK0	TnON	TnRP2	TnRP1	TnRP0
TMnC1	TnM1	TnM0	TnIO1	TnIO0	TnOC	TnPOL	TnDPX	TnCCLR
TMnDL	D7	D6	D5	D4	D3	D2	D1	D0
TMnDH	—	—	—	—	—	—	D9	D8
TMnAL	D7	D6	D5	D4	D3	D2	D1	D0
TMnAH	—	—	—	—	—	—	D9	D8

Compact TM Register List (n=0)

TMnDL 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 **TMnDL**: TMn Counter Low Byte Register bit 7 ~ bit 0
TMn 10-bit Counter bit 7 ~ bit 0

TMnDH 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 **TMnDH**: TMn Counter High Byte Register bit 1 ~ bit 0
TMn 10-bit Counter bit 9 ~ bit 8

TMnAL 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 **TMnAL**: TMn CCRA Low Byte Register bit 7 ~ bit 0
TMn 10-bit CCRA bit 7 ~ bit 0

TMnAH 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 TMn 10-bit CCRA bit 9 ~ bit 8

TMnC0 Register

Bit	7	6	5	4	3	2	1	0
Name	TnPAU	TnCK2	TnCK1	TnCK0	TnON	TnRP2	TnRP1	TnRP0
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 **TnPAU**: TMn 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 TM 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 **TnCK2~TnCK0**: Select TMn Counter clock
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{TBC}
101: undefined
110: TCKn rising edge clock
111: TCKn falling edge clock
These three bits are used to select the clock source for the TM. Selecting the Reserved clock input will effectively disable the internal counter. 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_{TBC} are other internal clocks, the details of which can be found in the oscillator section.

Bit 3 **TnON**: TMn Counter On/Off Control
0: Off
1: On
This bit controls the overall on/off function of the TM. Setting the bit high enables the counter to run, clearing the bit disables the TM. Clearing this bit to zero will stop the counter from counting and turn off the TM 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.
If the TM is in the Compare Match Output Mode then the TM output pin will be reset to its initial condition, as specified by the TnOC bit, when the TnON bit changes from low to high.

Bit 2~0 **TnRP2~TnRP0**: TMn CCRP 3-bit register, compared with the TMn Counter bit 9~bit 7 Comparator P Match Period
000: 1024 TMn clocks
001: 128 TMn clocks
010: 256 TMn clocks
011: 384 TMn clocks
100: 512 TMn clocks
101: 640 TMn clocks
110: 768 TMn clocks
111: 896 TMn 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 TnCCLR bit is set to zero. Setting the TnCCLR 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.

TMnC1 Register

Bit	7	6	5	4	3	2	1	0
Name	TnM1	TnM0	TnIO1	TnIO0	TnOC	TnPOL	TnDPX	TnCCLR
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 **TnM1~TnM0**: Select TMn Operating Mode
00: Compare Match Output Mode
01: Undefined
10: PWM Mode
11: Timer/Counter Mode
These bits setup the required operating mode for the TM. To ensure reliable operation the TM should be switched off before any changes are made to the TnM1 and TnM0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4 **TnIO1~TnIO0**: Select TPn_0, TPn_1 output function
Compare Match Output Mode
00: No change
01: Output low
10: Output high
11: Toggle output
PWM 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 TM output pin changes state when a certain condition is reached. The function that these bits select depends upon in which mode the TM is running.
In the Compare Match Output Mode, the TnIO1 and TnIO0 bits determine how the TM output pin changes state when a compare match occurs from the Comparator A. The TM 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 TM output pin should be setup using the TnOC bit in the TMnC1 register. Note that the output level requested by the TnIO1 and TnIO0 bits must be different from the initial value setup using the TnOC bit otherwise no change will occur on the TM output pin when a compare match occurs. After the TM output pin changes state it can be reset to its initial level by changing the level of the TnON bit from low to high.

In the PWM Mode, the TnIO1 and TnIO0 bits determine how the TM 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 change the values of the TnIO1 and TnIO0 bits only after the TMn has been switched off. Unpredictable PWM outputs will occur if the TnIO1 and TnIO0 bits are changed when the TM is running

Bit 3 **TnOC**: TPn_0, TPn_1 Output control bit
Compare Match Output Mode
0: Initial low
1: Initial high
PWM Mode
0: Active low
1: Active high
This is the output control bit for the TM output pin. Its operation depends upon whether TM is being used in the Compare Match Output Mode or in the PWM Mode. It has no effect if the TM is in the Timer/Counter Mode. In the Compare Match Output Mode it determines the logic level of the TM output pin before a compare match occurs. In the PWM Mode it determines if the PWM signal is active high or active low.

Bit 2 **TnPOL**: TPn_0, TPn_1 Output polarity Control
0: Non-invert
1: Invert
This bit controls the polarity of the TPn_0 or TPn_1 output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1 **TnDPX**: TMn PWM period/duty 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 **TnCCLR**: Select TMn Counter clear condition
0: TMn Comparatror P match
1: TMn Comparatror 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 TnCCLR 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 TnCCLR bit is not used in the PWM Mode.

Compact Type TM Operating Modes

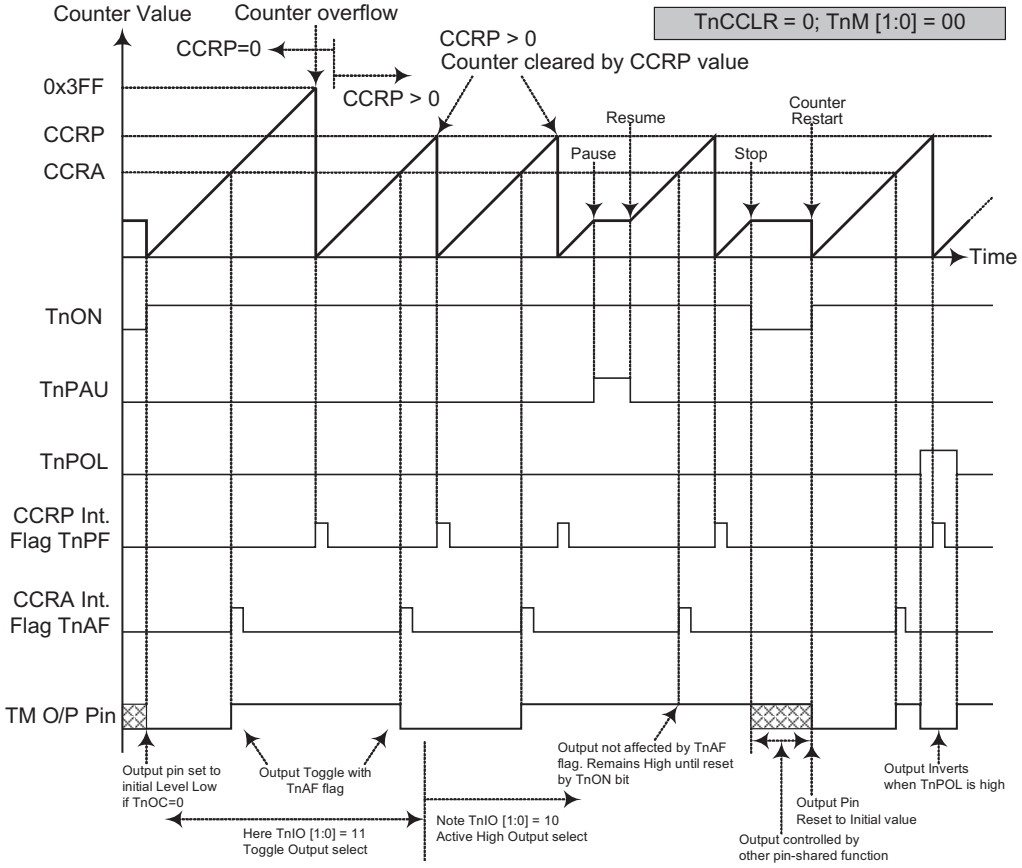
The Compact Type TM can operate in one of three operating modes, Compare Match Output Mode, PWM Mode or Timer/Counter Mode. The operating mode is selected using the TnM1 and TnM0 bits in the TMnC1 register.

Compare Match Output Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 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 TnCCLR bit is low, there are two ways in which the counter can be cleared. One is when a compare match occurs from Comparator P, the other is when the CCRP bits are all zero which allows the counter to overflow. Here both TnAF and TnPF interrupt request flags for the Comparator A and Comparator P respectively, will both be generated.

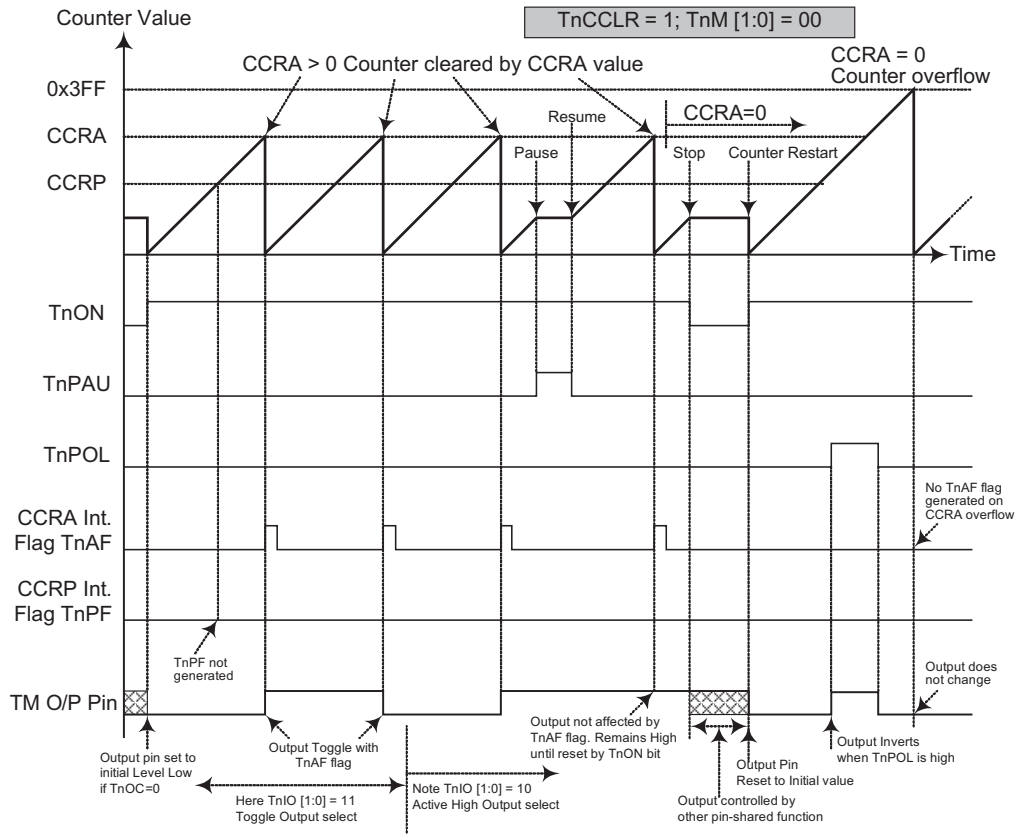
If the TnCCLR bit in the TMnC1 register is high then the counter will be cleared when a compare match occurs from Comparator A. However, here only the TnAF interrupt request flag will be generated even if the value of the CCRP bits is less than that of the CCRA registers. Therefore when TnCCLR is high no TnPF interrupt request flag will be generated. If the CCRA bits are all zero, the counter will overflow when its reaches its maximum 10-bit, 3FF Hex, value, however here the TnAF interrupt request flag will not be generated.

As the name of the mode suggests, after a comparison is made, the TM output pin will change state. The TM output pin condition however only changes state when an TnAF interrupt request flag is generated after a compare match occurs from Comparator A. The TnPF interrupt request flag, generated from a compare match occurs from Comparator P, will have no effect on the TM output pin. The way in which the TM output pin changes state are determined by the condition of the TnIO1 and TnIO0 bits in the TMnC1 register. The TM output pin can be selected using the TnIO1 and TnIO0 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 TM output pin, which is setup after the TnON bit changes from low to high, is setup using the TnOC bit. Note that if the TnIO1 and TnIO0 bits are zero then no pin change will take place.



Compare Match Output Mode -- TnCCLR = 0

Note: 1. With TnCCLR=0, a Comparator P match will clear the counter
2. The TM output pin is controlled only by the TnAF flag
3. The output pin is reset to its initial state by a TnON bit rising edge



Compare Match Output Mode -- TnCCLR = 1

Note: 1. With TnCCLR=1, a Comparator A match will clear the counter
2. The TM output pin is controlled only by the TnAF flag
3. The output pin is reset to its initial state by a TnON bit rising edge
4. The TnPF flag is not generated when TnCCLR=1

Timer/Counter Mode

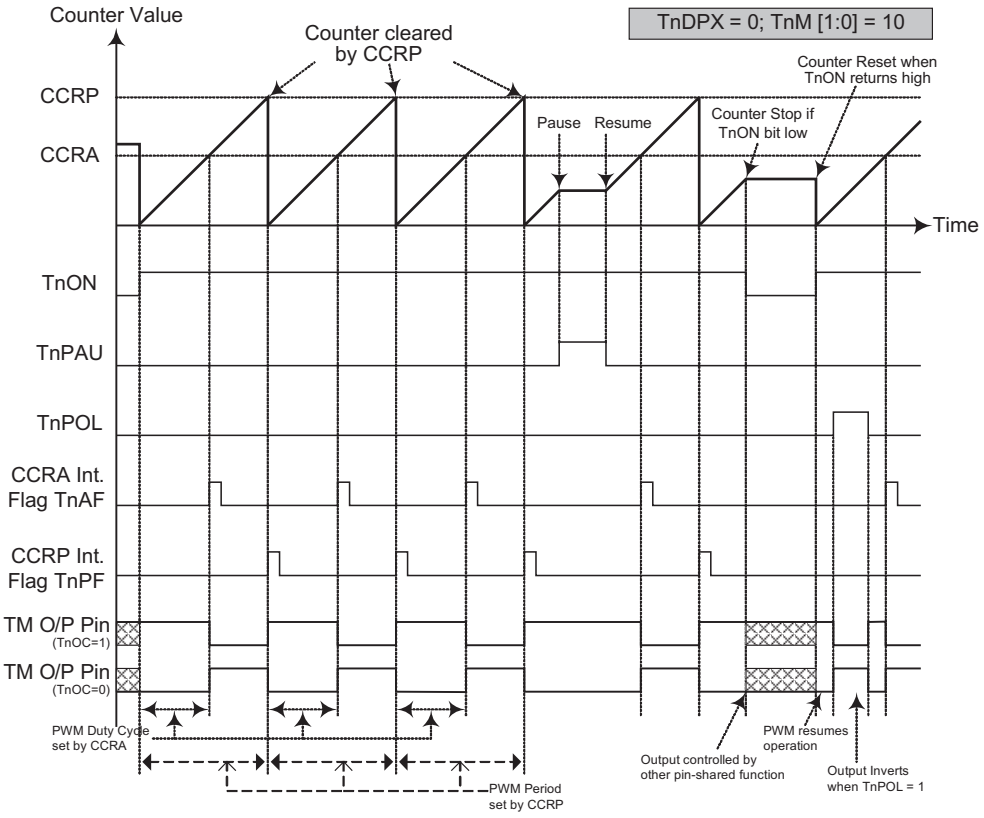
To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 11 respectively. The Timer/Counter Mode operates in an identical way to the Compare Match Output Mode generating the same interrupt flags. The exception is that in the Timer/Counter Mode the TM output pin is not used. Therefore the above description and Timing Diagrams for the Compare Match Output Mode can be used to understand its function. As the TM 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 TnM1 and TnM0 in the TMnC1 register should be set to 10 respectively. The PWM function within the TM 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 TM 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 mode, the TnCCLR bit has no effect on the PWM operation. Both of the CCRA and CCRP registers are used to generate the PWM waveform, one register is used to clear the internal counter and thus control the PWM waveform frequency, while the other one is used to control the duty cycle. Which register is used to control either frequency or duty cycle is determined using the TnDPX bit in the TMnC1 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 TnOC bit in the TMnC1 register is used to select the required polarity of the PWM waveform while the two TnIO1 and TnIO0 bits are used to enable the PWM output or to force the TM output pin to a fixed high or low level. The TnPOL bit is used to reverse the polarity of the PWM output waveform.



PWM Mode -- TnDPX = 0

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

CTM, PWM Mode, Edge-aligned Mode, T0DPX=0

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

If f_{SYS} = 16MHz, TM clock source is $f_{SYS}/4$, CCRP = 100b and CCRA =128,

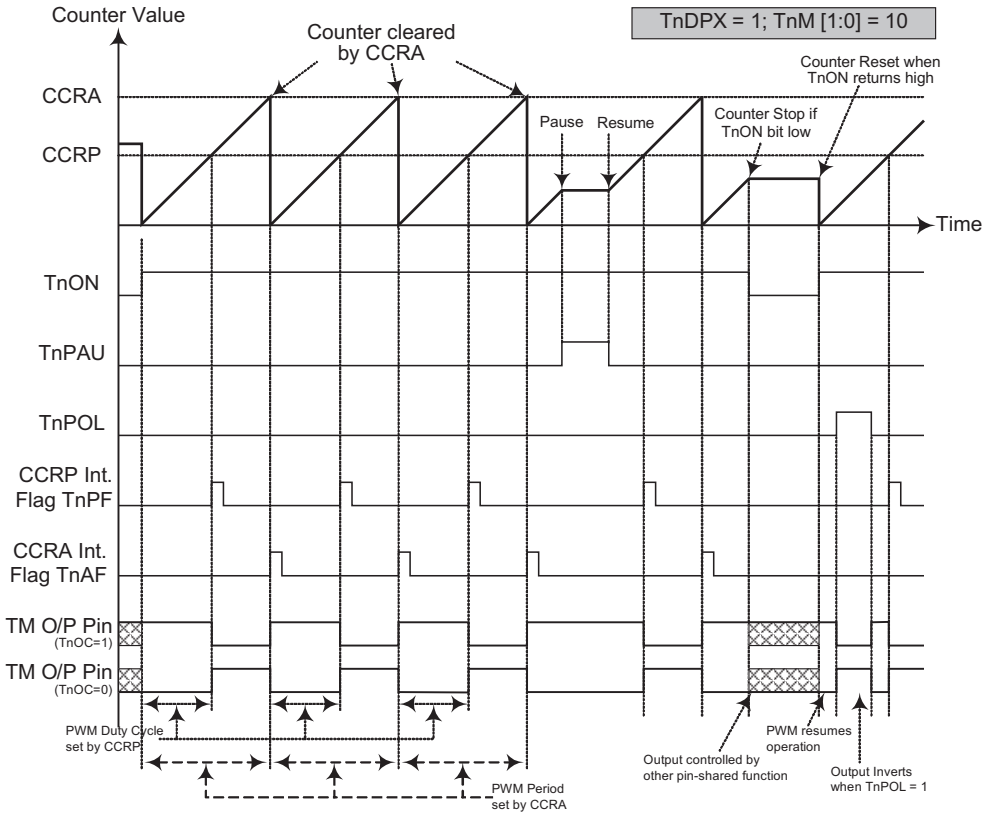
The CTM PWM output frequency = $(f_{SYS}/4) / 512 = f_{SYS}/2048$ = 7.8125 kHz, duty = 128/512 = 25%.

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%.

CTM, PWM Mode, Edge-aligned Mode, T0DPX=1

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

The output period is determined by the CCRA register value together with the TM clock while the PWM duty cycle is defined by the CCRP register value.



PWM Mode -- TnDPX = 1

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

Standard Type TM – STM

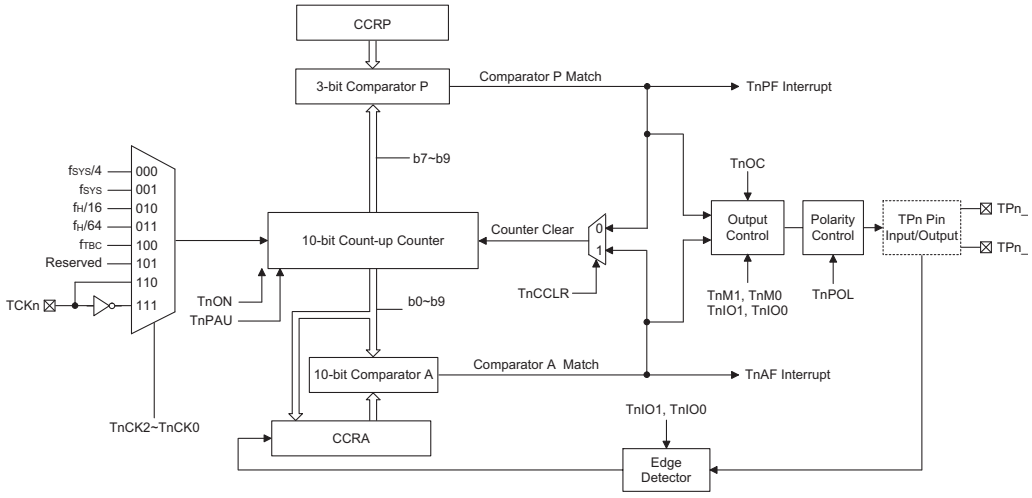
The Standard Type TM contains five operating modes, which are Compare Match Output, Timer/Event Counter, Capture Input, Single Pulse Output and PWM Output modes. The Standard TM can also be controlled with an external input pin and can drive one or two external output pins.

CTM	Name	TM No.	TM Input Pin	TM Output Pin
BS85B12-3	—	—	—	—
BS85C20-3	10-bit STM	2	TCK2	TP2_0, TP2_1

Standard TM Operation

At 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 is three bits wide whose value is compared with the highest three bits in the counter while the CCRA is the ten bits and therefore compares 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 TnON 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 TM interrupt signal will also usually be generated. The Standard 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 an output pin. All operating setup conditions are selected using relevant internal registers.



Standard Type TM Block Diagram

Standard Type TM Register Description

Overall operation of the Standard 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 the three CCRP bits.

STM Register List

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM2C0	T2PAU	T2CK2	T2CK1	T2CK0	T2ON	T2RP2	T2RP1	T2RP0
TM2C1	T2M1	T2M0	T2IO1	T2IO0	T2OC	T2POL	T2DPX	T2CCLR
TM2DL	D7	D6	D5	D4	D3	D2	D1	D0
TM2DH	—	—	—	—	—	—	D9	D8
TM2AL	D7	D6	D5	D4	D3	D2	D1	D0
TM2AH	—	—	—	—	—	—	D9	D8

10-bit Standard TM Register List

TM2C0 Register

Bit	7	6	5	4	3	2	1	0
Name	T2PAU	T2CK2	T2CK1	T2CK0	T2ON	T2RP2	T2RP1	T2RP0
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 **T2PAU**: TM2 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 TM 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 **T2CK2~T2CK0**: Select TM2 Counter clock
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{TBC}
101: undefined
110: TCK2 rising edge clock
111: TCK2 falling edge clock
These three bits are used to select the clock source for the TM. Selecting the Reserved clock input will effectively disable the internal counter. 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_{TBC} are other internal clocks, the details of which can be found in the oscillator section.

Bit 3 **T2ON**: TM2 Counter On/Off Control
0: Off
1: On
This bit controls the overall on/off function of the TM. Setting the bit high enables the counter to run, clearing the bit disables the TM. Clearing this bit to zero will stop the counter from counting and turn off the TM 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 TM is in the Compare Match Output Mode then the TM output pin will be reset to its initial condition, as specified by the T2OC bit, when the T2ON bit changes from low to high.

Bit 2~0 **T2RP2~T2RP0**: TM2 CCRP 3-bit register, compared with the TM2 Counter bit 9~bit 7 Comparator P Match Period
000: 1024 TM2 clocks
001: 128 TM2 clocks
010: 256 TM2 clocks
011: 384 TM2 clocks
100: 512 TM2 clocks
101: 640 TM2 clocks
110: 768 TM2 clocks
111: 896 TM2 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 T2CCLR bit is set to zero. Setting the T2CCLR 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.

TM2C1 Register

Bit	7	6	5	4	3	2	1	0
Name	T2M1	T2M0	T2IO1	T2IO0	T2OC	T2POL	T2DPX	T2CCLR
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 **T2M1~T2M0**: Select TM2 Operating Mode
00: Compare Match Output Mode
01: Capture Input Mode
10: PWM Mode or Single Pulse Output Mode
11: Timer/Counter Mode
These bits setup the required operating mode for the TM. To ensure reliable operation the TM should be switched off before any changes are made to the T2M1 and T2M0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4 **T2IO1~T2IO0**: Select TP2_0, TP2_1 output function
Compare Match Output Mode
00: No change
01: Output low
10: Output high
11: Toggle output
PWM Mode/Single Pulse Output Mode
00: PWM output inactive state
01: PWM output active state
10: PWM output
11: Single pulse output
Capture Input Mode
00: Input capture at rising edge of TP2_0, TP2_1
01: Input capture at falling edge of TP2_0, TP2_1
10: Input capture at falling/rising edge of TP2_0, TP2_1
11: Input capture disabled
Timer/counter Mode:
Unused

These two bits are used to determine how the TM output pin changes state when a certain condition is reached. The function that these bits select depends upon in which mode the TM is running.
In the Compare Match Output Mode, the T2IO1 and T2IO0 bits determine how the TM output pin changes state when a compare match occurs from the Comparator A. The TM 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 TM output pin should be setup using the T2OC bit in the TM2C1 register. Note that the output level requested by the T2IO1 and T2IO0 bits must be different from the initial value setup using the T2OC bit otherwise no change will occur on the TM output pin when a compare match occurs. After the TM output pin changes state it can be reset to its initial level by changing the level of the T2ON bit from low to high.
In the PWM Mode, the T2IO1 and T2IO0 bits determine how the TM 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 change the values of the T2IO1 and T2IO0 bits only after the TM has been switched off. Unpredictable PWM outputs will occur if the T2IO1 and T2IO0 bits are changed when the TM is running

Bit 3 **T2OC**: TP2_0, TP2_1 Output control bit

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

Bit 2 **T2POL**: TP2_0, TP2_1 Output polarity Control
0: non-invert
1: invert
This bit controls the polarity of the TP2_0 or TP2_1 output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1 **T2DPX**: TM1 PWM period/duty 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 **T2CCLR**: Select TM1 Counter clear condition
0: TM2 Comparatror P match
1: TM2 Comparatror A match
This bit is used to select the method which clears the counter. Remember that the Standard TM contains two comparators, Comparator A and Comparator P, either of which can be selected to clear the internal counter. With the T2CCLR 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 T2CCLR bit is not used in the PWM, Single Pulse or Input Capture Mode.

TM2DL 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 **TM2DL**: TM2 Counter Low Byte Register bit 7~bit 0
TM2 10-bit Counter bit 7~bit 0

TM2DH 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 **TM2DH**: TM2 Counter High Byte Register bit 1~bit 0
TM2 10-bit Counter bit 9~bit 8

TM2AL 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 **TM2AL**: TM2 CCRA Low Byte Register bit 7~bit 0
TM2 10-bit CCRA bit 7~bit 0

TM2AH Registe

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 **TM2AH**: TM2 CCRA High Byte Register bit 1~bit 0
TM2 10-bit CCRA bit 9~bit 8

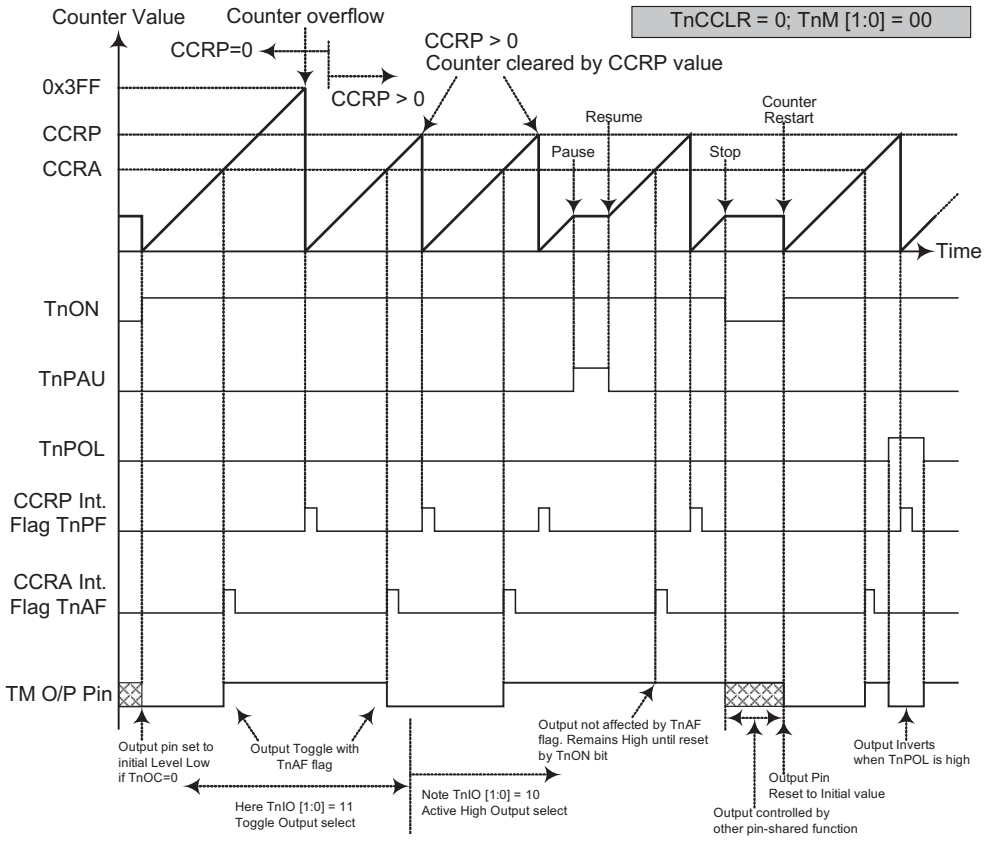
Standard Type TM Operating Modes

The Standard Type TM can operate in one of five operating modes, Compare Match Output Mode, PWM Mode, Single Pulse Output Mode, Capture Input Mode or Timer/Counter Mode. The operating mode is selected using the TnM1 and TnM0 bits in the TMnC1 register.

Compare Match Output Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 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 TnCCLR 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 TnAF and TnPF interrupt request flags for Comparator A and Comparator P respectively, will both be generated.

If the TnCCLR bit in the TMnC1 register is high then the counter will be cleared when a compare match occurs from Comparator A. However, here only the TnAF interrupt request flag will be generated even if the value of the CCRP bits is less than that of the CCRA registers. Therefore when TnCCLR is high no TnPF interrupt request flag will be generated. In the Compare Match Output Mode, the CCRA can not be set to 0.



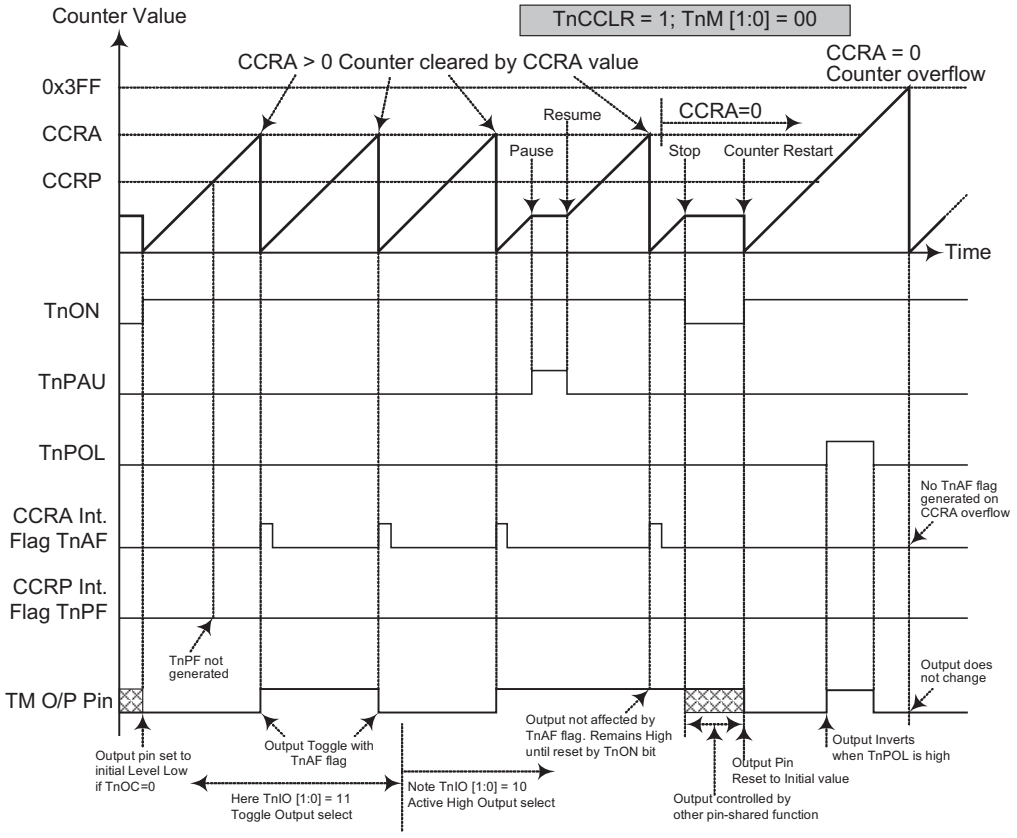
Compare Match Output Mode -- TnCCLR = 0

Note: 1. With TnCCLR=0 a Comparator P match will clear the counter
2. The TM output pin is controlled only by the TnAF flag
3. The output pin is reset to itsinitial state by a TnON bit rising edge

As the name of the mode suggests, after a comparison is made, the TM output pin, will change state. The TM output pin condition however only changes state when a TnAF interrupt request flag is generated after a compare match occurs from Comparator A. The TnPF interrupt request flag, generated from a compare match occurs from Comparator P, will have no effect on the TM output pin. The way in which the TM output pin changes state are determined by the condition of the TnIO1 and TnIO0 bits in the TMnC1 register. The TM output pin can be selected using the TnIO1 and TnIO0 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 TM output pin, which is setup after the TnON bit changes from low to high, is setup using the TnOC bit. Note that if the TnIO1 and TnIO0 bits are zero then no pin change will take place.

Timer/Counter Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 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 TM 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 TM output pin is not used in this mode, the pin can be used as a normal I/O pin or other pin-shared function.



Compare Match Output Mode -- TnCCLR = 1

Note: 1. With TnCCLR=1 a Comparator A match will clear the counter
2. The TM output pin is controlled only by the TnAF flag
3. The output pin is reset to its initial state by a TnON bit rising edge
4. A TnPF flag is not generated when TnCCLR=1

PWM Output Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 10 respectively. The PWM function within the TM 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 TM 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 mode, the TnCCLR bit has no effect on the PWM operation. Both of the CCRA and CCRP registers are used to generate the PWM waveform, one register is used to clear the internal counter and thus control the PWM waveform frequency, while the other one is used to control the duty cycle. Which register is used to control either frequency or duty cycle is determined using the TnDPX bit in the TMnC1 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 TnOC bit in the TMnC1 register is used to select the required polarity of the PWM waveform while the two TnIO1 and TnIO0 bits are used to enable the PWM output or to force the TM output pin to a fixed high or low level. The TnPOL bit is used to reverse the polarity of the PWM output waveform.

- STM, PWM Mode, Edge-aligned Mode, T0DPX=0

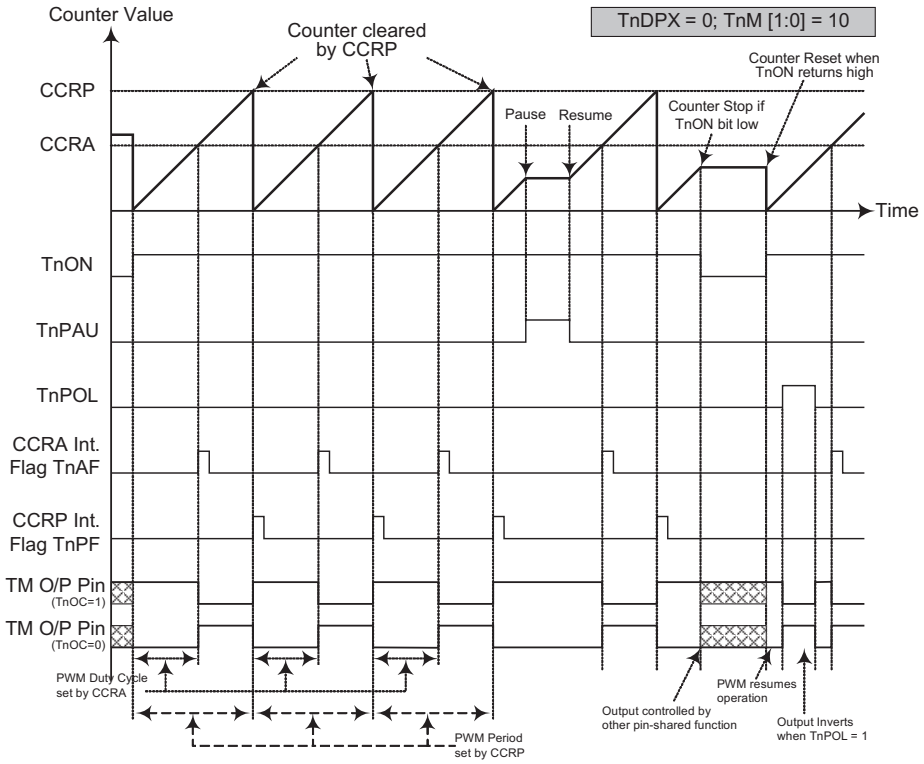
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	128	256	384	512	640	768	896	1024
Duty	CCRA							

If f_{SYS} = 16MHz, TM clock source is $f_{SYS}/4$, CCRP = 100b and CCRA =128,
The STM PWM output frequency = $(f_{SYS}/4) / 512 = f_{SYS}/2048$ = 7.8125 kHz, 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%.

- STM, PWM Mode, Edge-aligned Mode, T0DPX=1

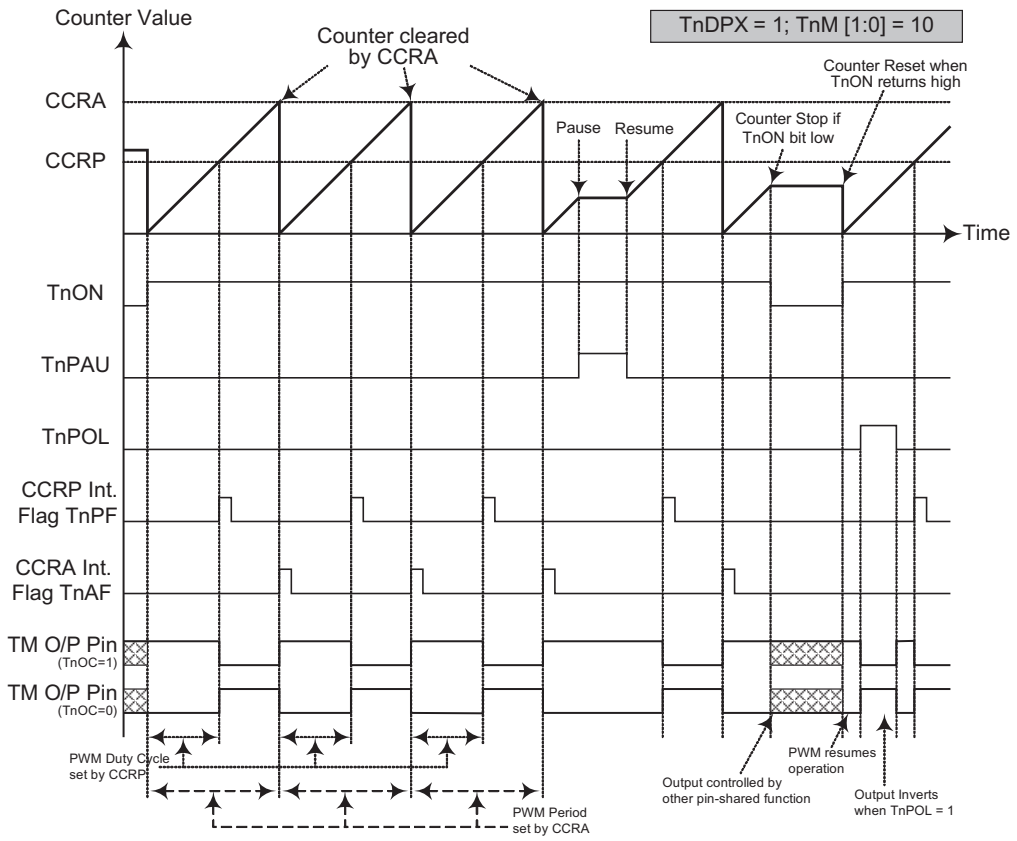
CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	CCRA							
Duty	128	256	384	512	640	768	896	1024

The PWM output period is determined by the CCRA register value together with the TM clock while the PWM duty cycle is defined by the CCRP register value.



PWM Mode -- TnDPX = 0

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



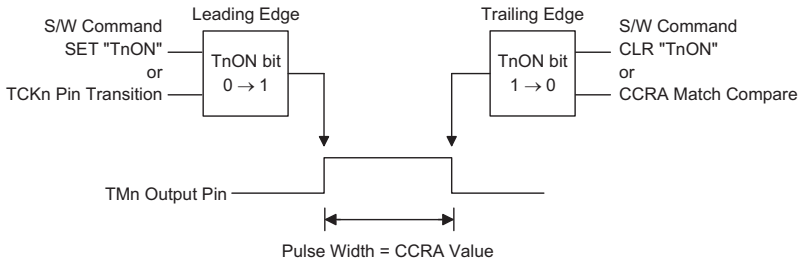
PWM Mode -- TnDPX = 1

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

Single Pulse Mode

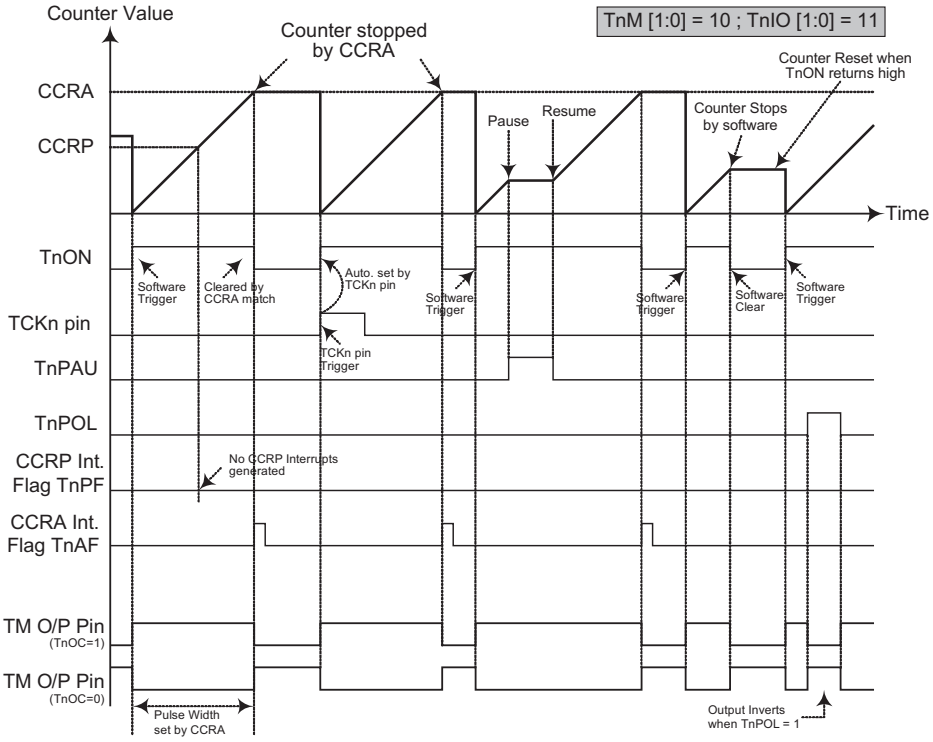
To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 10 respectively and also the TnIO1 and TnIO0 bits should be set to 11 respectively. The Single Pulse Output Mode, as the name suggests, will generate a single shot pulse on the TM output pin.

The trigger for the pulse output leading edge is a low to high transition of the TnON bit, which can be implemented using the application program. However in the Single Pulse Mode, the TnON bit can also be made to automatically change from low to high using the external TCKn pin, which will in turn initiate the Single Pulse output. When the TnON bit transitions to a high level, the counter will start running and the pulse leading edge will be generated. The TnON bit should remain high when the pulse is in its active state. The generated pulse trailing edge will be generated when the TnON bit is cleared to zero, which can be implemented using the application program or when a compare match occurs from Comparator A.



Single Pulse Generation

However a compare match from Comparator A will also automatically clear the TnON 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 TM interrupt. The counter can only be reset back to zero when the TnON bit changes from low to high when the counter restarts. In the Single Pulse Mode CCRP is not used. The TnCCLR and TnDPX bits are not used in this Mode.



Single Pulse Mode

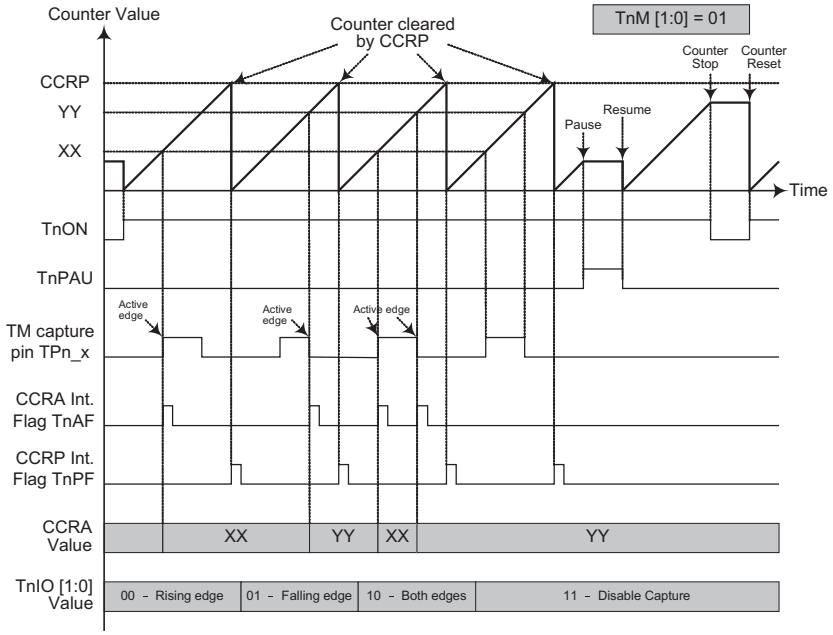
Note: 1. Counter stopped by CCRA
2. CCRP is not used
3. The pulse is triggered by the TCKn pin or by setting the TnON bit high
4. A TCKn pin active edge will automatically set the TnON bit high
5. In the Single Pulse Mode, TnIO [1:0] must be set to "11" and can not be changed.

Capture Input Mode

To select this mode bits TnM1 and TnM0 in the TMnC1 register should be set to 01 respectively. This mode enables external signals to capture and store the present value of the internal counter and can therefore be used for applications such as pulse width measurements. The external signal is supplied on the TPn_0 or TPn_1 pin, whose active edge can be either a rising edge, a falling edge or both rising and falling edges; the active edge transition type is selected using the TnIO1 and TnIO0 bits in the TMnC1 register. The counter is started when the TnON bit changes from low to high which is initiated using the application program.

When the required edge transition appears on the TPn_0 or TPn_1 pin, the present value in the counter will be latched into the CCRA registers and a TM interrupt generated. Irrespective of what events occur on the TPn_0 or TPn_1 pin the counter will continue to free run until the TnON bit changes from high to low. When a CCRP compare match occurs, the counter will reset back to zero; in this way the CCRP value can be used to control the maximum counter value. When a CCRP compare match occurs from Comparator P, a TM interrupt will also be generated. Counting the number of overflow interrupt signals from the CCRP can be a useful method in measuring long pulse widths. The TnIO1 and TnIO0 bits can select the active trigger edge on the TPn_0 or TPn_1 pin to be a rising edge, falling edge or both edge types. If the TnIO1 and TnIO0 bits are both set high, then no capture operation will take place irrespective of what happens on the TPn_0 or TPn_1 pin, however it must be noted that the counter will continue to run.

As the TPn_0 or TPn_1 pin is pin shared with other functions, care must be taken if the TM is in the Input Capture Mode. This is because if the pin is setup as an output, then any transitions on this pin may cause an input capture operation to be executed. The TnCCLR and TnDPX bits are not used in this Mode.



Capture Input Mode

Note: 1.. TnM [1:0] = 01 and active edge set by the TnIO [1:0] bits
2. A TM Capture input pin active edge transfers the counter value to CCRA
3. TnCCLR bit not used
4. No output function -- TnOC and TnPOL bits are not used
5. CCRP determines the counter value and the counter has a maximum count value when CCRP is equal to zero.

Enhanced Type TM – ETM

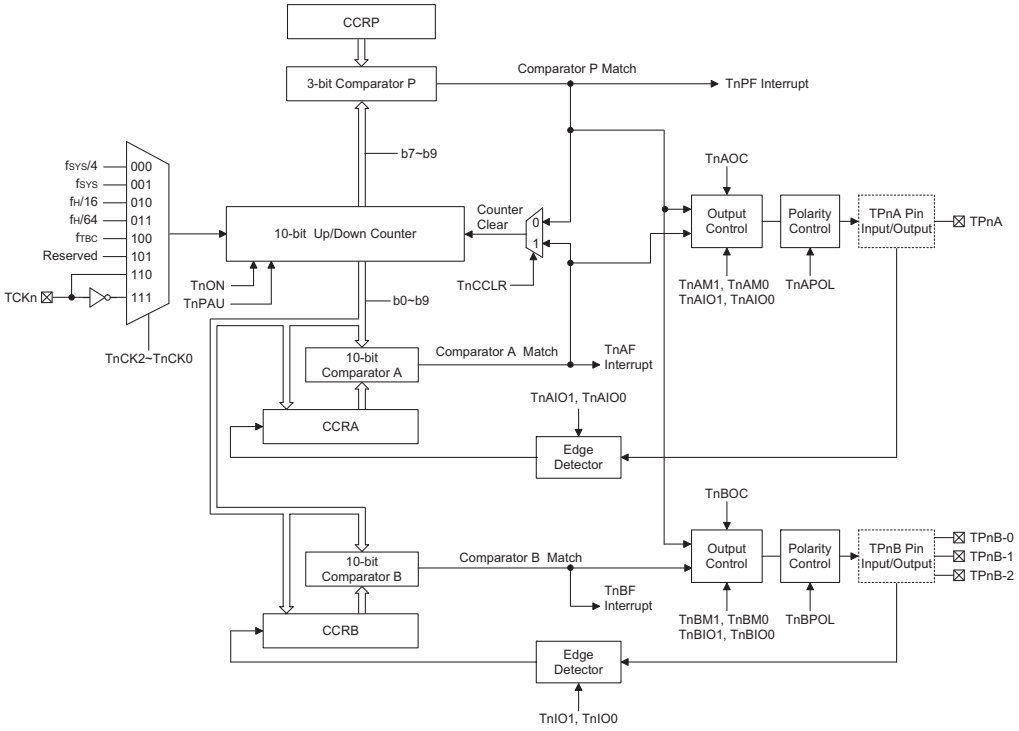
The Enhanced Type TM contains five operating modes, which are Compare Match Output, Timer/Event Counter, Capture Input, Single Pulse Output and PWM Output modes. The Enhanced TM can also be controlled with an external input pin and can drive three or four external output pins.

CTM	Name	TM No.	TM Input Pin	TM Output Pin
BS85B12-3	10-bit ETM	1	TCK1	TP1A, TP1B_0, TP1B_1, TP1B_2
BS85C20-3	10-bit ETM	1	TCK1	TP1A, TP1B_0, TP1B_1, TP1B_2

Enhanced TM Operation

At its core is a 10-bit count-up/count-down counter which is driven by a user selectable internal or external clock source. There are three internal comparators with the names, Comparator A, Comparator B and Comparator P. These comparators will compare the value in the counter with the CCRA, CCRB and CCRP registers. The CCRP comparator is 3-bits wide whose value is compared with the highest 3-bits in the counter while CCRA and CCRB are 10-bits wide and therefore 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 TnON 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 TM interrupt signal will also usually be generated. The Enhanced 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 output pins. All operating setup conditions are selected using relevant internal registers.



Enhanced Type TM Block Diagram

Enhanced Type TM Register Description

Overall operation of the Enhanced 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 CCRB value. The remaining three registers are control registers which setup the different operating and control modes as well as the three CCRP bits.

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM1C0	T1PAU	T1CK2	T1CK1	T1CK0	T1ON	T1RP2	T1RP1	T1RP0
TM1C1	T1AM1	T1AM0	T1AIO1	T1AIO0	T1AOC	T1APOL	T1CDN	T1CCLR
TM1C2	T1BM1	T1BM0	T1BIO1	T1BIO0	T1BOC	T1BPOL	T1PWM1	T1PWM0
TM1DL	D7	D6	D5	D4	D3	D2	D1	D0
TM1DH	—	—	—	—	—	—	D9	D8
TM1AL	D7	D6	D5	D4	D3	D2	D1	D0
TM1AH	—	—	—	—	—	—	D9	D8
TM1BL	D7	D6	D5	D4	D3	D2	D1	D0
TM1BH	—	—	—	—	—	—	D9	D8

10-bit Enhanced TM Register List (if ETM is TM1)

10-bit Enhanced TM Register List

- TM1C0 Register -- 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	T1PAU	T1CK2	T1CK1	T1CK0	T1ON	T1RP2	T1RP1	T1RP0
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 **T1PAU**: TM1 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 TM 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 **T1CK2~T1CK0**: Select TM1 Counter clock
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{TBC}
101: Undefined
110: TCK1 rising edge clock
111: TCK1 falling edge clock

These three bits are used to select the clock source for the TM. Selecting the Reserved clock input will effectively disable the internal counter. 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_{TBC} are other internal clocks, the details of which can be found in the oscillator section.

Bit 3 **T1ON**: TM1 Counter On/Off Control
0: Off
1: On
This bit controls the overall on/off function of the TM. Setting the bit high enables the counter to run, clearing the bit disables the TM. Clearing this bit to zero will stop the counter from counting and turn off the TM 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 TM is in the Compare Match Output Mode then the TM output pin will be reset to its initial condition, as specified by the T1OC bit, when the T1ON bit changes from low to high.

Bit 2~0 **T1RP2~T1RP0**: TM1 3-bit register, compared with the TM1 Counter bit 9~bit 7
Comparator P Match Period
000: 1024 TM1 clocks
001: 128 TM1 clocks
010: 256 TM1 clocks
011: 384 TM1 clocks
100: 512 TM1 clocks
101: 640 TM1 clocks
110: 768 TM1 clocks
111: 896 TM1 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 T1CCLR bit is set to zero. Setting the T1CCLR 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.

- TM1C1 Register -- 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	T1AM1	T1AM0	T1AIO1	T1AIO0	T1AOC	T1APOL	T1CDN	T1CCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **T1AM1~T1AM0**: Select TM1 CCRA Operating Mode
00: Compare Match Output Mode
01: Capture Input Mode
10: PWM Mode or Single Pulse Output Mode
11: Timer/Counter Mode
These bits setup the required operating mode for the TM. To ensure reliable operation the TM should be switched off before any changes are made to the T1AM1 and T1AM0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4 **T1AIO1~T1AIO0**: Select TP1A output function
Compare Match Output Mode
00: No change
01: Output low
10: Output high
11: Toggle output
Mode/ Single Pulse Output Mode
00: PWM output inactive state
01: PWM output active state
10: PWM output
11: Single pulse output
Capture Input Mode
00: Input capture at rising edge of TP1A
01: Input capture at falling edge of TP1A
10: Input capture at falling/rising edge of TP1A
11: Input capture disabled

Timer/counter Mode
Unused

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

In the Compare Match Output Mode, the T1AIO1 and T1AIO0 bits determine how the TM output pin changes state when a compare match occurs from the Comparator A. The TM 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 TM output pin should be setup using the T1AOC bit in the TM1C1 register. Note that the output level requested by the T1AIO1 and T1AIO0 bits must be different from the initial value setup using the T1AOC bit otherwise no change will occur on the TM output pin when a compare match occurs. After the TM output pin changes state it can be reset to its initial level by changing the level of the T1ON bit from low to high.

In the PWM Mode, the T1AIO1 and T1AIO0 bits determine how the TM 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 change the values of the T1AIO1 and T1AIO0 bits only after the TM has been switched off. Unpredictable PWM outputs will occur if the T1AIO1 and T1AIO0 bits are changed when the TM is running

Bit 3 **T1AOC**: TP1A Output control bit

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

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

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

Bit 2 **T1APOL**: TP1A Output polarity Control
0: Non-invert
1: Invert

This bit controls the polarity of the TP1A output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1 **T1CDN**: TM1 Counter count up or down flag
0: Count up
1: Count down

Bit 0 **T1CCLR**: Select TM1 Counter clear condition
0: TM1 Comparator P match
1: TM1 Comparator A match

This bit is used to select the method which clears the counter. Remember that the Enhanced TM contains three comparators, Comparator A, Comparator B and Comparator P, but only Comparator A or Comparator Pan be selected to clear the internal counter. With the T1CCLR 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 T1CCLR bit is not used in the PWM, Single Pulse or Input Capture Mode.

- TM1C2 Register -- 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	T1BM1	T1BM0	T1BIO1	T1BIO0	T1BOC	T1BPOL	T1PWM1	T1PWM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **T1BM1~T1BM0**: Select TM1 CCRB Operating Mode
00: Compare Match Output Mode
01: Capture Input Mode
10: PWM Mode or Single Pulse Output Mode
11: Timer/Counter mode

These bits setup the required operating mode for the TM. To ensure reliable operation the TM should be switched off before any changes are made to the T1BM1 and T1BM0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4 **T1BIO1~T1BIO0**: Select TP1B_0, TP1B_1, TP1B_2 output function

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

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

Capture Input Mode
00: Input capture at rising edge of TP1B_0, TP1B_1, TP1B_2
01: Input capture at falling edge of TP1B_0, TP1B_1, TP1B_2
10: Input capture at falling/rising edge of TP1B_0, TP1B_1, TP1B_2
11: Input capture disabled

Timer/counter Mode
Unused

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

In the Compare Match Output Mode, the T1BIO1 and T1BIO0 bits determine how the TM output pin changes state when a compare match occurs from the Comparator A. The TM 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 TM output pin should be setup using the T1BOC bit in the TM1C2 register. Note that the output level requested by the T1BIO1 and T1BIO0 bits must be different from the initial value setup using the T1BOC bit otherwise no change will occur on the TM output pin when a compare match occurs. After the TM output pin changes state it can be reset to its initial level by changing the level of the T1ON bit from low to high.

In the PWM Mode, the T1BIO1 and T1BIO0 bits determine how the TM 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 change the values of the T1BIO1 and T1BIO0 bits only after the TM has been switched off. Unpredictable PWM outputs will occur if the T1BIO1 and T1BIO0 bits are changed when the TM is running

Bit 3 **T1BOC**: TP1B_0, TP1B_1, TP1B_2 Output control bit
Compare Match Output Mode
0: Initial low
1: Initial high
Mode/ Single Pulse Output Mode
0: Active low
1: Active high
This is the output control bit for the TM output pin. Its operation depends upon whether TM is being used in the Compare Match Output Mode or in the PWM Mode/ Single Pulse Output Mode. It has no effect if the TM is in the Timer/Counter Mode. In the Compare Match Output Mode it determines the logic level of the TM output pin before a compare match occurs. In the PWM Mode it determines if the PWM signal is active high or active low.

Bit 2 **T1BPOL**: TP1B_0, TP1B_1, TB1B_2 Output polarity Control
0: Non-invert
1: Invert
This bit controls the polarity of the TP1B_0, TP1B_1, TP1B_2 output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1~0 **T1PWM1~T1PWM0**: Select PWM Mode
00: Edge aligned
01: Centre aligned, compare match on count up
10: Centre aligned, compare match on count down
11: Centre aligned, compare match on count up or down

- TM1DL Register -- 10-bit ETM

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 **TM1DL**: TM1 Counter Low Byte Register bit 7~bit 0
TM1 10-bit Counter bit 7~bit 0

- TM1DH Register -- 10-bit ETM

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 **TM1DH**: TM1 Counter High Byte Register bit 1~bit 0
TM1 10-bit Counter bit 9~bit 8

- TM1AL Register -- 10-bit ETM

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 **TM1AL**: TM1 Low Byte Register bit 7~bit 0
TM1 10-bit CCRA bit 7~bit 0

• TM1AH Register -- 10-bit ETM

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 **TM1AH**: TM1 High Byte Register bit 1~bit 0
TM1 10-bit CCRA bit 9~bit 8

• TM1BL Register -- 10-bit ETM

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 **TM1BL**: TM1 Low Byte Register bit 7~bit 0
TM1 10-bit CCRB bit 7~bit 0

• TM1BH Register -- 10-bit ETM

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 **TM1BH**: TM1 High Byte Register bit 1~bit 0
TM1 10-bit CCRB bit 9 ~ bit 8

Enhanced Type TM Operating Modes

The Enhanced Type TM can operate in one of five operating modes, Compare Match Output Mode, PWM Output Mode, Single Pulse Output Mode, Capture Input Mode or Timer/Counter Mode. The operating mode is selected using the TnAM1 and TnAM0 bits in the TMnC1, and the TnBM1 and TnBM0 bits in the TMnC2 register.

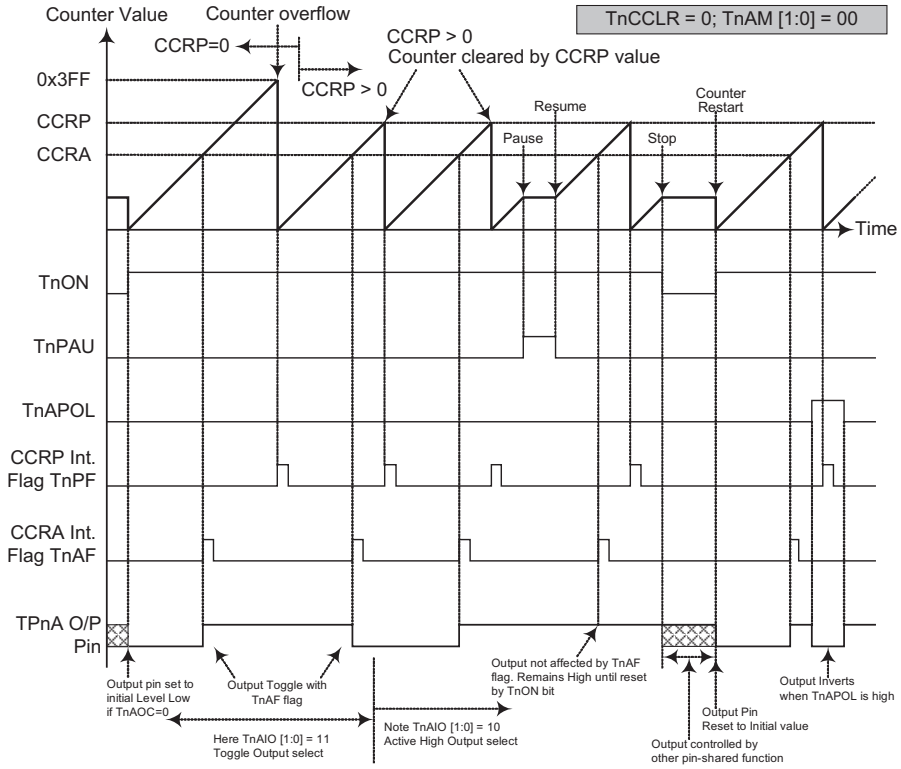
ETM Operating Mode	CCRA Compare Match Output Mode	CCRA Timer/Counter Mode	CCRA PWM Output Mode	CCRA Single Pulse Output Mode	CCRA Input Capture Mode
CCRB Compare Match Output Mode	√	—	—	—	—
CCRB Timer/Counter Mode	—	√	—	—	—
CCRB PWM Output Mode	—	—	√	—	—
CCRB Single Pulse Output Mode	—	—	—	√	—
CCRB Input Capture Mode	—	—	—	—	√

"√": permitted; "—" : not permitted

Compare Output Mode

To select this mode, bits TnAM1, TnAM0 and TnBM1, TnBM0 in the TMnC1/TMnC2 registers should be all cleared to zero. 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 TnCCLR bit is low, there are two ways in which the counter can be cleared. One is when a compare match occurs from Comparator P, the other is when the CCRP bits are all zero which allows the counter to overflow. Here both the TnAF and TnPF interrupt request flags for Comparator A and Comparator P respectively, will both be generated.

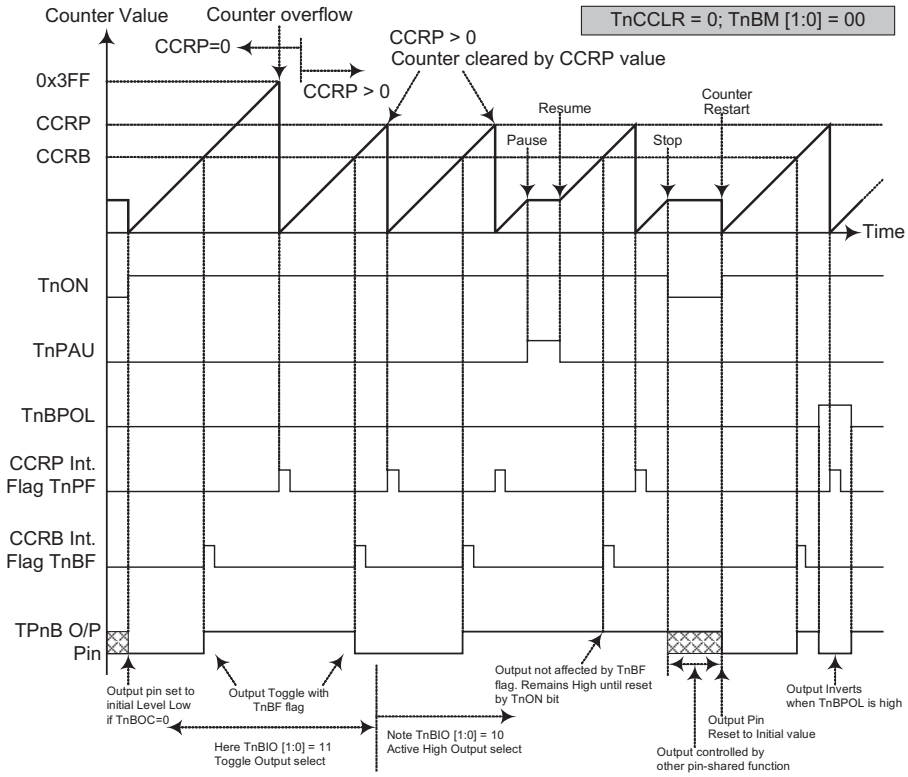
If the TnCCLR bit in the TMnC1 register is high then the counter will be cleared when a compare match occurs from Comparator A. However, here only the TnAF interrupt request flag will be generated even if the value of the CCRP bits is less than that of the CCRA registers. Therefore when TnCCLR is high no TnPF interrupt request flag will be generated.



ETM CCRA Compare Match Output Mode -- TnCCLR = 0

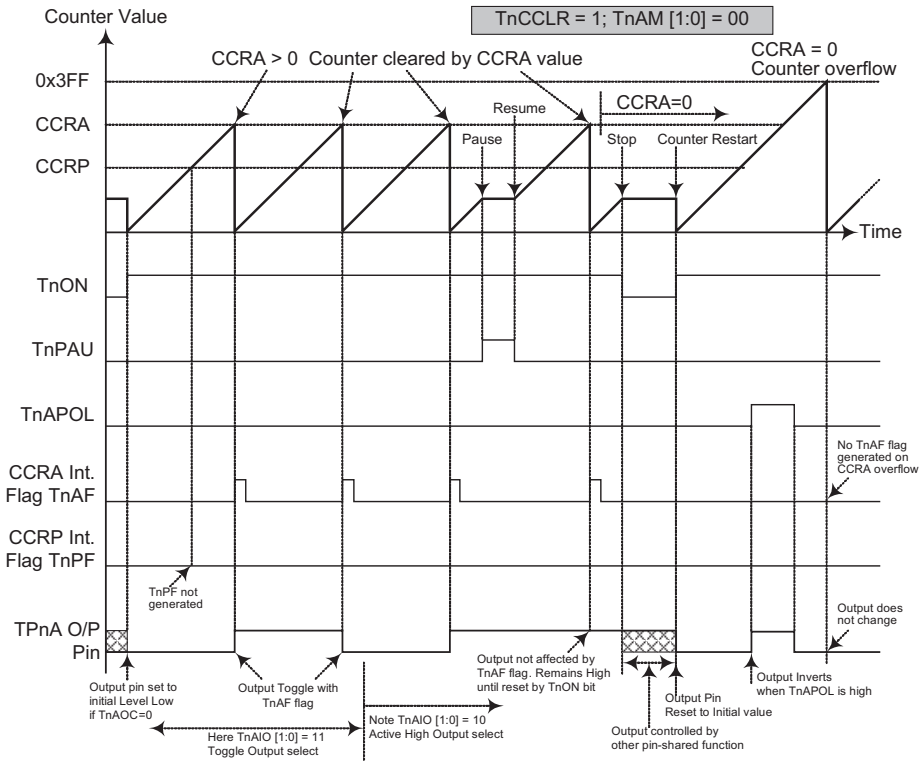
Note: 1. With TnCCLR=0 a Comparator P match will clear the counter
2. The TPnA output pin is controlled only by the TnAF flag
3. The output pin is reset to its initial state by a TnON bit rising edge

As the name of the mode suggests, after a comparison is made, the TM output pin, will change state. The TM output pin condition however only changes state when an TnAF or TnBF interrupt request flag is generated after a compare match occurs from Comparator A or Comparator B. The TnPF interrupt request flag, generated from a compare match from Comparator P, will have no effect on the TM output pin. The way in which the TM output pin changes state is determined by the condition of the TnAIO1 and TnAIO0 bits in the TMnC1 register for ETM CCRA, and the TnBIO1 and TnBIO0 bits in the TMnC2 register for ETM CCRB. The TM output pin can be selected using the TnAIO1, TnAIO0 bits (for the TPnA pin) and TnBIO1, TnBIO0 bits (for the TPnB_0, TPnB_1 or TPnB_2 pins) to go high, to go low or to toggle from its present condition when a compare match occurs from Comparator A or a compare match occurs from Comparator B. The initial condition of the TM output pin, which is setup after the TnON bit changes from low to high, is setup using the TnAOC or TnBOC bit for TPnA or TPnB_0, TPnB_1, TPnB_2 output pins. Note that if the TnAIO1,TnAIO0 and TnBIO1, TnBIO0 bits are zero then no pin change will take place.



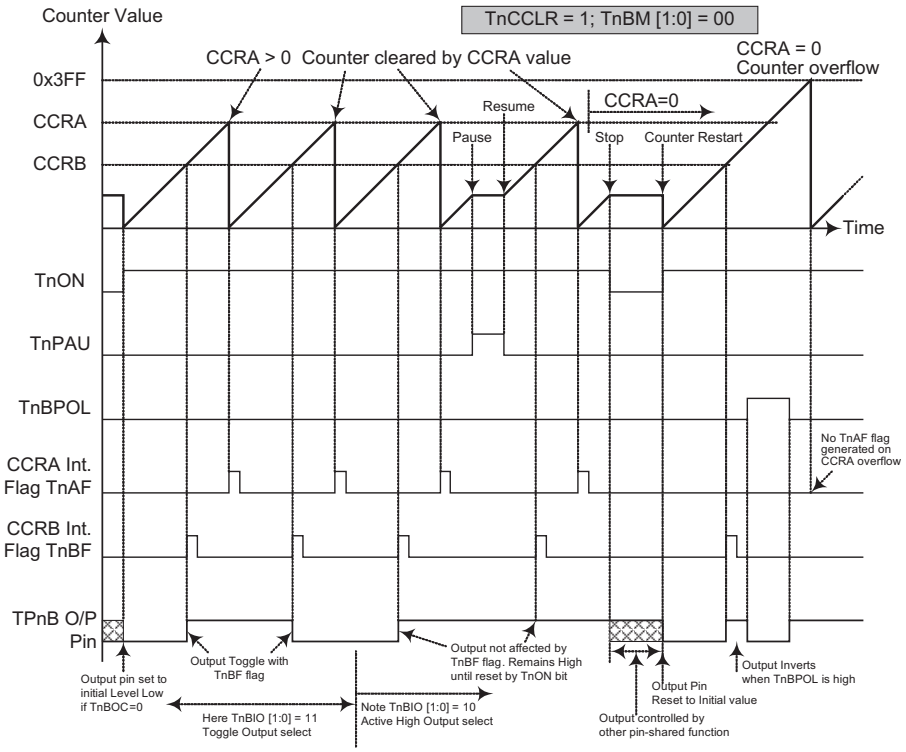
ETM CCRB Compare Match Output Mode -- TnCCLR = 0

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



ETM CCRA Compare Match Output Mode -- TnCCLR = 1

Note: 1. With TnCCLR=1 a Comparator A match will clear the counter
2. The TPnA output pin is controlled only by the TnAF flag
3. The TPnA output pin is reset to its initial state by a TnON bit rising edge
4. The TnPF flag is not generated when TnCCLR=1



ETM CCRB Compare Match Output Mode -- TnCCLR = 1

Note: 1. With TnCCLR=1 a Comparator A match will clear the counter

2. The TPnB output pin is controlled only by the TnBF flag
3. The TPnB output pin is reset to its initial state by a TnON bit rising edge
4. The TnPF flag is not generated when TnCCLR=1

Timer/Counter Mode

To select this mode, bits TnAM1, TnAM0 and TnBM1, TnBM0 in the TMnC1 and TMnC2 register should all be set high. 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 TM 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 TM 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, the required bit pairs, TnAM1, TnAM0 and TnBM1, TnBM0 should be set to 10 respectively and also the TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits should be set to 10 respectively. The PWM function within the TM 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 TM 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 mode, the TnCCLR bit is used to determine in which way the PWM period is controlled. With the TnCCLR bit set high, the PWM period can be finely controlled using the CCRA registers. In this case the CCRB registers are used to set the PWM duty value (for TPnB output pins). The CCRP bits are not used and TPnA output pin is not used. The PWM output can only be generated on the TPnB output pins. With the TnCCLR bit cleared to zero, the PWM period is set using one of the eight values of the three CCRP bits, in multiples of 128. Now both CCRA and CCRB registers can be used to setup different duty cycle values to provide dual PWM outputs on their relative TPnA and TPnB pins.

The TnPWM1 and TnPWM0 bits determine the PWM alignment type, which can be either edge or centre type. In edge alignment, the leading edge of the PWM signals will all be generated concurrently when the counter is reset to zero. With all power currents switching on at the same time, this may give rise to problems in higher power applications. In centre alignment the centre of the PWM active signals will occur sequentially, thus reducing the level of simultaneous power switching currents.

Interrupt flags, one for each of the CCRA, CCRB and CCRP, will be generated when a compare match occurs from either the Comparator A, Comparator B or Comparator P. The TnAOC and TnBOC bits in the TMnC1 and TMnC2 register are used to select the required polarity of the PWM waveform while the two TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits pairs are used to enable the PWM output or to force the TM output pin to a fixed high or low level. The TnAPOL and TnBPOL bit are used to reverse the polarity of the PWM output waveform.

ETM, PWM Mode, Edge-aligned Mode, TnCCLR=0

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

If f_{SYS} = 16MHz, TM clock source select $f_{SYS}/4$, CCRP = 100b, CCRA = 128 and CCRB = 256,
The TP1A PWM output frequency = $(f_{SYS}/4) / 512 = f_{SYS}/2048$ = 7.8125kHz, duty = 128/512 = 25%.
The TP1B_n PWM output frequency = $(f_{SYS}/4) / 512 = f_{SYS}/2048$ = 7.8125kHz, duty = 256/512 = 50%.
If the Duty value defined by CCRA or CCRB register is equal to or greater than the Period value, then the PWM output duty is 100%.

ETM, PWM Mode, Edge-aligned Mode, TnCCLR=1

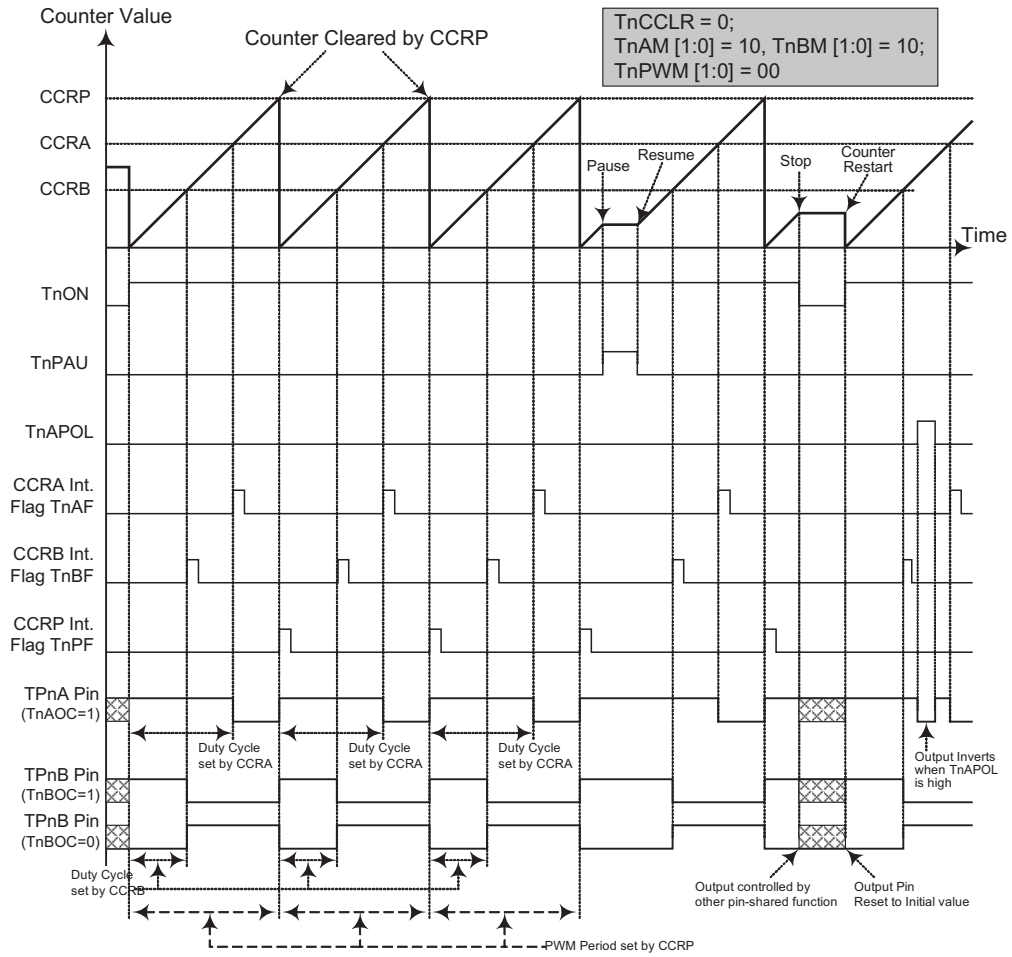
CCRA	1	2	3	511	512	1021	1022	1023
Period	1	2	3	511	512	1021	1022	1023
B Duty	CCRB							

ETM, PWM Mode, Center-aligned Mode, TnCCLR=0

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	256	512	768	1024	1280	1536	1792	2046
A Duty	(CCRA×2)–1							
B Duty	(CCRB×2)–1							

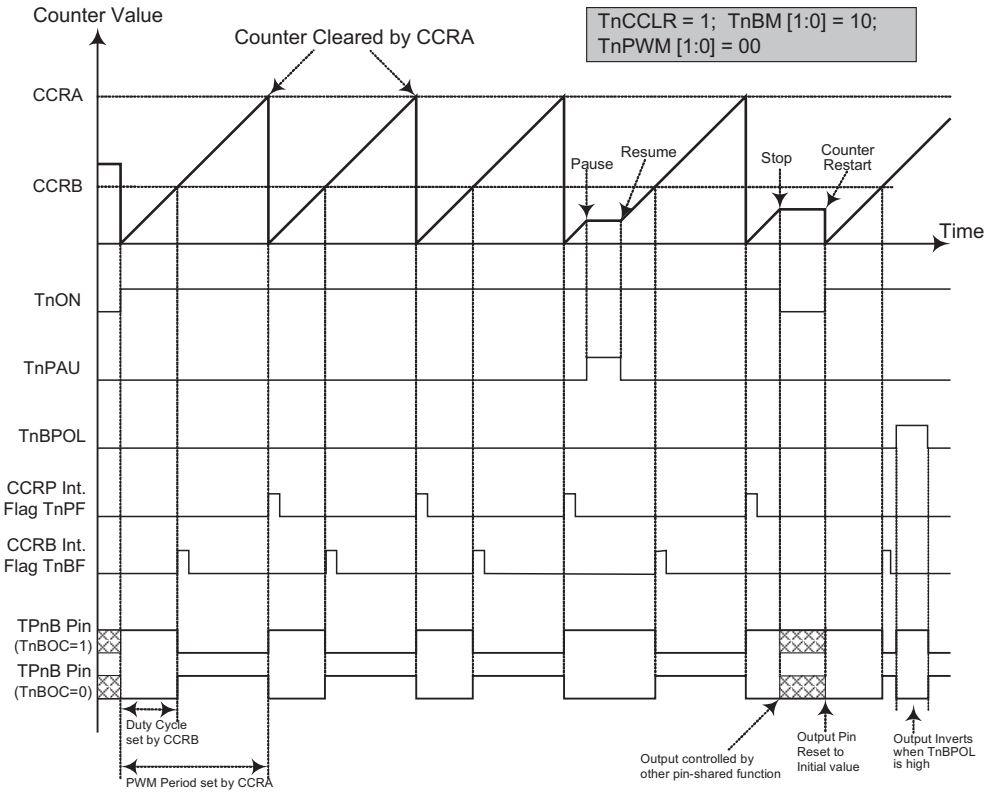
ETM, PWM Mode, Center-aligned Mode, TnCCLR=1

CCRA	1	2	3	511	512	1021	1022	1023
Period	2	4	6	1022	1024	2042	2044	2046
B Duty	(CCRB×2)–1							



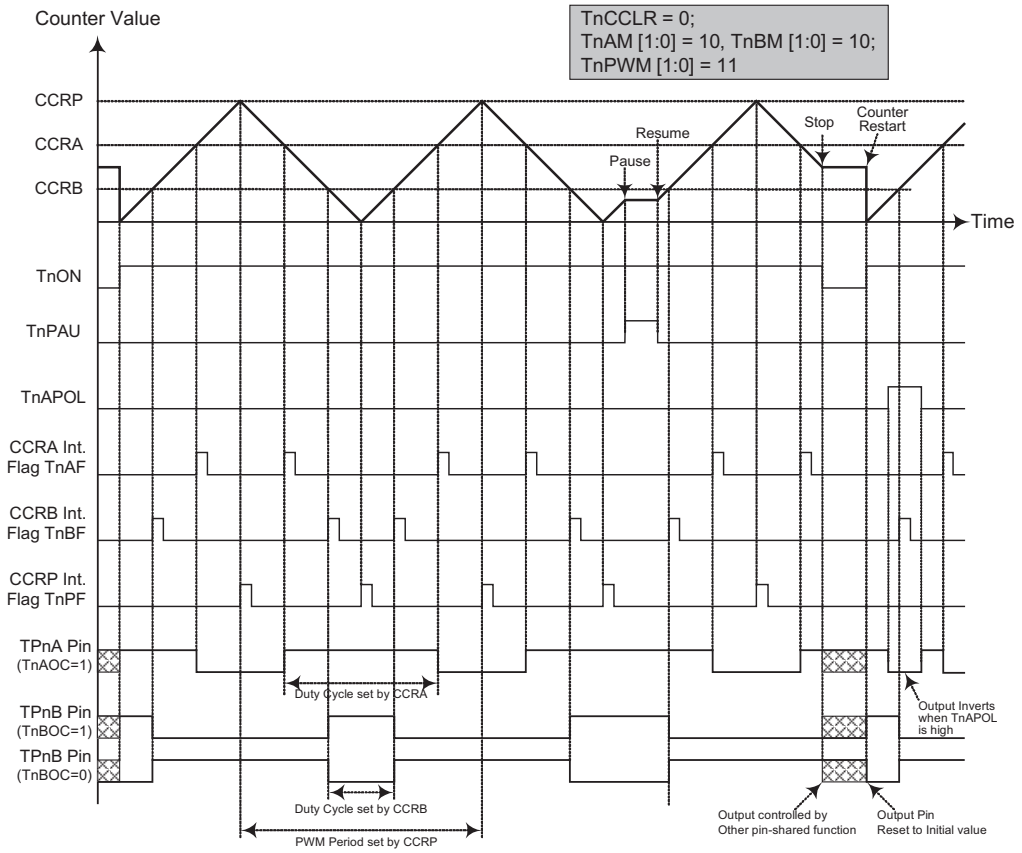
ETM PWM Mode -- Edge Aligned

Note: 1. Here TnCCLR=0 therefore CCRP clears counter and determines the PWM period
2. The internal PWM function continues running even when TnAIO [1:0] (or TnBIO [1:0]) = 00 or 01
3. CCRA controls the TPnA PWM duty and CCRB controls the TPnB PWM duty



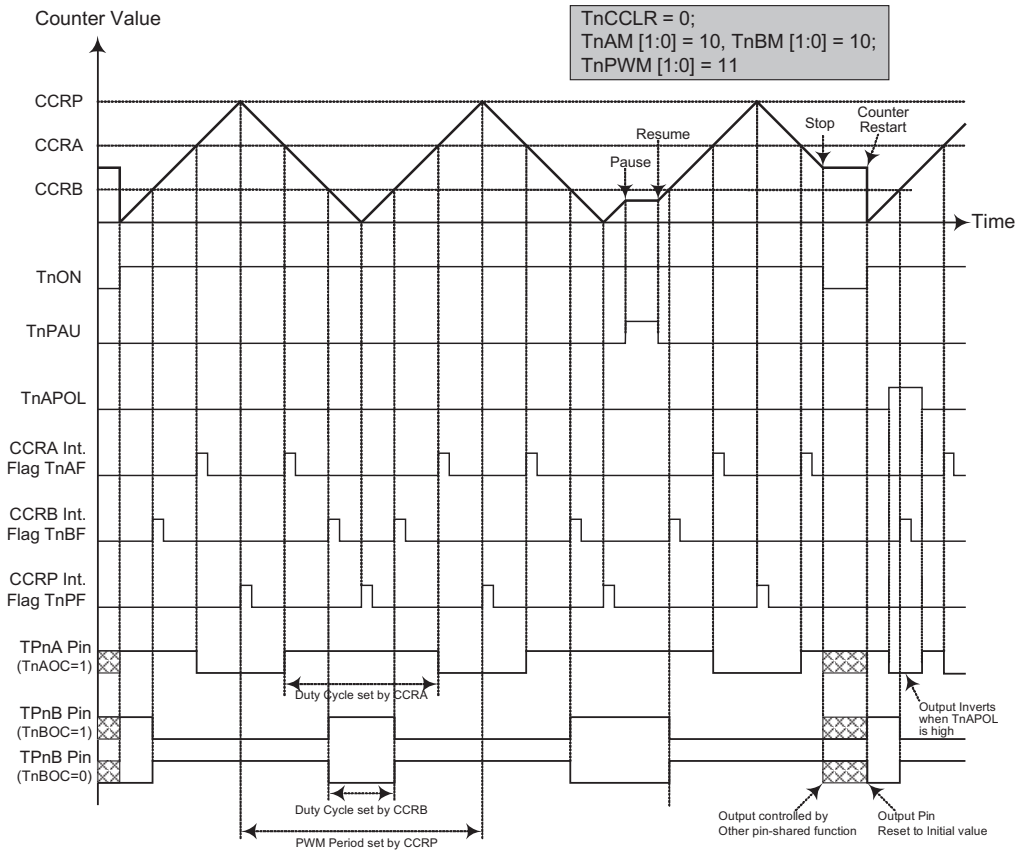
ETM PWM Mode -- Edge Aligned

Note: 1. Here TnCCLR=1 therefore CCRA clears the counter and determines the PWM period
2. The internal PWM function continues running even when TnBIO [1:0] = 00 or 01
3. The CCRA controls the TPnB PWM period and CCRB controls the TPnB PWM duty
4. Here the TM pin control register should not enable the TPnA pin as a TM output pin.



ETM PWM Mode -- Centre Aligned

Note: 1. Here TnCCLR=0 therefore CCRP clears the counter and determines the PWM period
2. TnPWM [1:0] =11 therefore the PWM is centre aligned
3. The internal PWM function continues running even when TnAIO [1:0] (or TnBIO [1:0]) = 00 or 01
4. CCRA controls the TPnA PWM duty and CCRB controls the TPnB PWM duty
5. CCRP will generate an interrupt request when the counter decrements to its zero value



ETM PWM Mode -- Centre Aligned

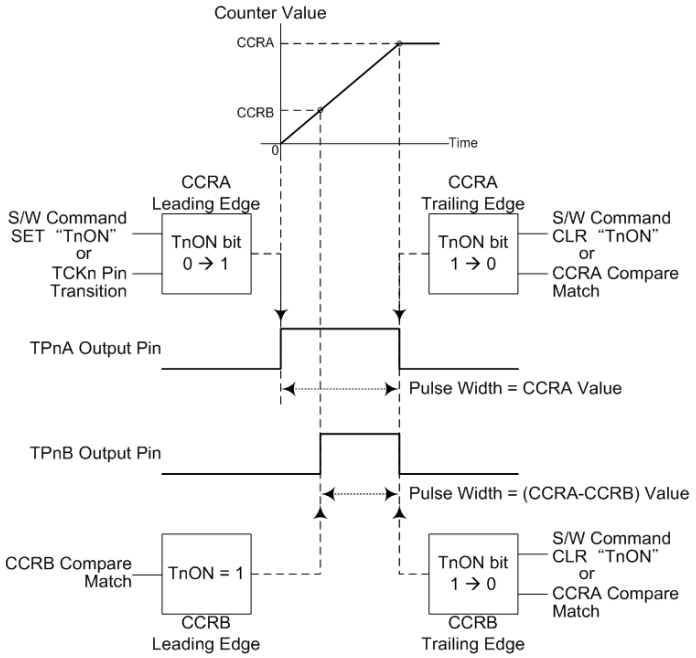
Note: 1. Here TnCCLR=1 therefore CCRA clears the counter and determines the PWM period
2. TnPWM [1:0] =11 therefore the PWM is centre aligned
3. The internal PWM function continues running even when TnBIO [1:0] = 00 or 01
4. CCRA controls the TPnB PWM period and CCRB controls the TPnB PWM duty
5. CCRP will generate an interrupt request when the counter decrements to its zero value

Single Pulse Output Mode

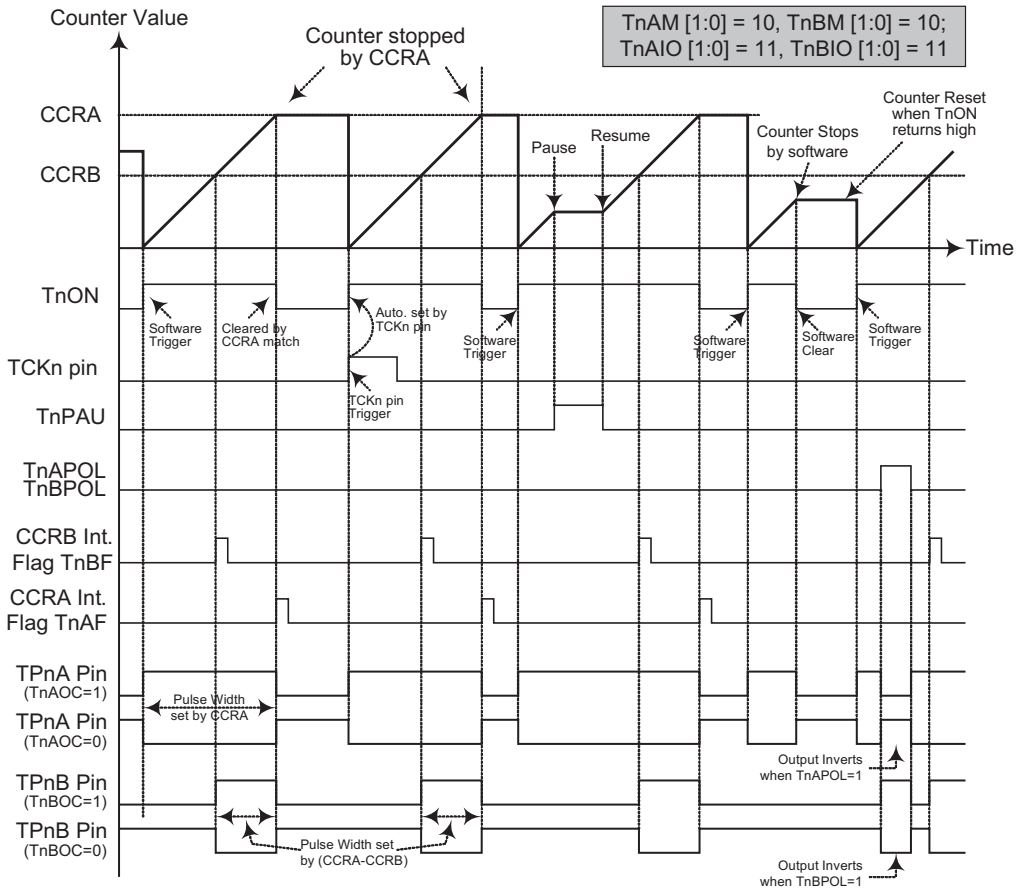
To select this mode, the required bit pairs, TnAM1, TnAM0 and TnBM1, TnBM0 should be set to 10 respectively and also the corresponding TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits should be set to 11 respectively. The Single Pulse Output Mode, as the name suggests, will generate a single shot pulse on the TM output pin.

The trigger for the pulse TPnA output leading edge is a low to high transition of the TnON bit, which can be implemented using the application program. The trigger for the pulse TPnB output leading edge is a compare match from Comparator B, which can be implemented using the application program. However in the Single Pulse Mode, the TnON bit can also be made to automatically change from low to high using the external TCKn pin, which will in turn initiate the Single Pulse output of TPnA. When the TnON bit transitions to a high level, the counter will start running and the pulse leading edge of TPnA will be generated. The TnON bit should remain high when the pulse is in its active state. The generated pulse trailing edge of TPnA and TPnB will be generated when the TnON 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 TnON bit and thus generate the Single Pulse output trailing edge of TPnA and TPnB. In this way the CCRA value can be used to control the pulse width of TPnA. The CCRA-CCRB value can be used to control the pulse width of TPnB. A compare match from Comparator A and Comparator B will also generate TM interrupts. The counter can only be reset back to zero when the TnON bit changes from low to high when the counter restarts. In the Single Pulse Mode CCRP is not used. The TnCCLR bit is also not used.



Single Pulse Generation



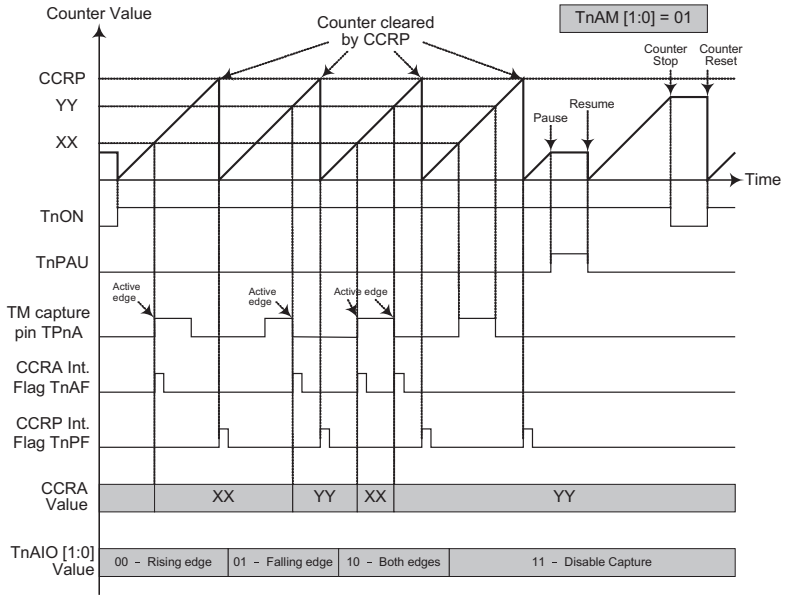
ETM -- Single Pulse Mode

Note: 1. Counter stopped by CCRA
2. CCRP is not used
3. The pulse is triggered by the TCKn pin or by setting the TnON bit high
4. A TCKn pin active edge will automatically set the TnON bit high
5. In the Single Pulse Mode, TnAIO [1:0] and TnBIO [1:0] must be set to ″11″ and can not be changed.

Capture Input Mode

To select this mode bits TnAM1, TnAM0 and TnBM1, TnBM0 in the TMnC1 and TMnC2 registers should be set to 01 respectively. This mode enables external signals to capture and store the present value of the internal counter and can therefore be used for applications such as pulse width measurements. The external signal is supplied on the TPnA and TPnB_0, TPnB_1, TPnB_2 pins, whose active edge can be either a rising edge, a falling edge or both rising and falling edges; the active edge transition type is selected using the TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits in the TMnC1 and TMnC2 registers. The counter is started when the TnON bit changes from low to high which is initiated using the application program.

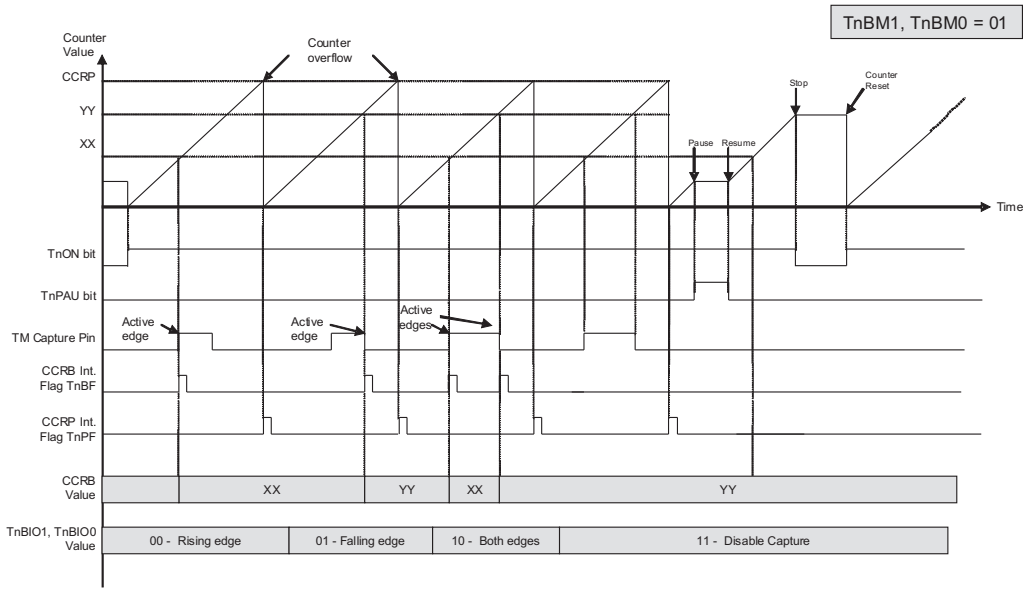
When the required edge transition appears on the TPnA and TPnB_0, TPnB_1, TPnB_2 pins the present value in the counter will be latched into the CCRA and CCRB registers and a TM interrupt generated. Irrespective of what events occur on the TPnA and TPnB_0, TPnB_1, TPnB_2 pins the counter will continue to free run until the TnON bit changes from high to low. When a CCRP compare match occurs the counter will reset back to zero; in this way the CCRP value can be used to control the maximum counter value. When a CCRP compare match occurs from Comparator P, a TM interrupt will also be generated. Counting the number of overflow interrupt signals from the CCRP can be a useful method in measuring long pulse widths. The TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits can select the active trigger edge on the TPnA and TPnB_0, TPnB_1, TPnB_2 pins to be a rising edge, falling edge or both edge types. If the TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits are both set high, then no capture operation will take place irrespective of what happens on the TPnA and TPnB_0, TPnB_1, TPnB_2 pins, however it must be noted that the counter will continue to run.



ETM CCRA Capture Input Mode

Note: 1. TnAM [1:0] = 01 and active edge set by the TnAIO [1:0] bits
2. The TM Capture input pin active edge transfers he counter value to CCRA
3. TnCCLR bit not used
4. No output function -- TnAOC and TnAPOL bits not used
5. CCRP determines the counter value and the counter has a maximum count value when CCRP is equal to zero.

As the TPnA and TPnB_0, TPnB_1, TPnB_2 pins are pin shared with other functions, care must be taken if the TM is in the Capture Input Mode. This is because if the pin is setup as an output, then any transitions on this pin may cause an input capture operation to be executed. The TnCCLR, TnAOC, TnBOC, TnAPOL and TnBPOL bits are not used in this mode.



ETM CCRB Capture Input Mode

Note: 1. TnBM [1:0] = 01 and active edge set by the TnBIO [1:0] bits
2. The TM Capture input pin active edge transfers the counter value to CCRB
3. TnCCLR bit not used
4. No output function -- TnBOC and TnBPOL bits not used
5. CCRP determines the counter value and the counter has a maximum count value when CCRP is equal to zero.

Touch Key Function

Each device provides multiple touch key functions. The touch key function is fully integrated and requires no external components, allowing touch key functions to be implemented by the simple manipulation of internal registers.

Touch Key Structure

The touch keys are pin shared with the PC and PD logic I/O pins, as well as having dedicated pins. For the pin shared touch keys, the touch key function is chosen using register bits. Keys are organised into groups of four, with each group known as a module and having a module number, M0 to M4. Each module contains its own control logic circuits and register set. Examination of the register names will reveal the module number it is referring to.

Device	Keys - n	Touch Key Module	Touch Key	Shared I/O Pin
BS85B12-3	12	M0	K1~K4	PC0~PC3
		M1	K5~K8	PC4~PC7
		M2	K9~K12	Dedicated Pins
BS85C20-3	20	M0	K1~K4	PC0~PC3
		M1	K5~K8	PC4~PC7
		M2	K9~K12	Dedicated Pins
		M3	K13~K16	PD0~PD3
		M4	K17~K20	PD4~PD7

Touch Key Module/Pin Reference Table

Touch Key Register Definition

Each touch key module, which contains four touch key functions, has its own suite of registers. The following table shows the register set for each touch key module. The Mn within the register name refers to the Touch Key module number and has a range of M0 to M4.

Name	Usage
TKMn16DH	16-bit C/F counter high byte
TKMn16DL	16-bit C/F counter low byte
TKMnC0	Control Register 0 Key Select
TKMnC1	Control Register 1 Internal reference. Touch pad reference.
TKMnC2	Control Register 2 Counter on-off and clear control/reference clock control/TKST start bit
TKMnC3	Control Register 3 Counter overflow bits

Register Listing

Register Name	Bit							
	7	6	5	4	3	2	1	0
TKMn16DH	D7	D6	D5	D4	D3	D2	D1	D0
TKMn16DL	D7	D6	D5	D4	D3	D2	D1	D0
TKMnC0	MnMXS1	MnMXS0	D5	D4	D3	D2	D1	D0
TKMnC1	MnK4OEN	MnK3OEN	MnK2OEN	MnK1OEN	MnK4IO	MnK3IO	MnK2IO	MnK1IO
TKMnC2	Mn16CTON	D6	MnST	MnROEN	MnRCCLR	Mn16CTCLR	D1	MnROS
TKMnC3	D9	D8	MnRCOV	Mn16CTOV	D3	MnROVS2	MnROVS1	MnROVS0

Register Content Summary

TKMn16DH 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 Module n 16-bit counter high byte contents

TKMn16DL 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 Module n 16-bit counter low byte contents

TKMnC0 Register

Bit	7	6	5	4	3	2	1	0
Name	MnMXS1	MnMXS0	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

Bits 7~6 **MnMXS1**, **MnMXS0**: Multiplexer Key Select

Bit		Module Number				
MnMXS1	MnMXS0	M0	M1	M2	M3	M4
0	0	Key 1	Key 5	Key 9	Key 13	Key 17
0	1	Key 2	Key 6	Key 10	Key 14	Key 18
1	0	Key 3	Key 7	Key 11	Key 15	Key 19
1	1	Key 4	Key 8	Key 12	Key 16	Key 20

Bit 5~0 **D5~D0**: These bits must be set to the binary value ″011000″

TKMnC1 Register

Bit	7	6	5	4	3	2	1	0
Name	MnK4OEN	MnK3OEN	MnK2OEN	MnK1OEN	MnK4IO	MnK3IO	MnK2IO	MnK1IO
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

For the BS85B12-3 n=0~2 while for the BS85C20-3 n=0~4.

Bits 7~4 MnK4OEN~ MnK1OEN: key selector control

MnK4OEN	M0	M1	M2	M3	M4
	Key 4	Key 8	Key 12	Key 16	Key 20
0	Disable				
1	Enable				

MnK3OEN	M0	M1	M2	M3	M4
	Key 3	Key 7	Key 11	Key 15	Key 19
0	Disable				
1	Enable				

MnK2OEN	M0	M1	M2	M3	M4
	Key 2	Key 6	Key 10	Key 14	Key 18
0	Disable				
1	Enable				

MnK1OEN	M0	M1	M2	M3	M4
	Key 1	Key 5	Key 9	Key 13	Key 17
0	Disable				
1	Enable				

Bits 3~0 I/O Pin or Touch Key Function Select

MnK4IO	M0	M1	M3	M4
	PC3/Key 4	PC7/Key 8	PD3/Key 16	PD7/Key 20
0	I/O pin			
1	Touch Key			

MnK3IO	M0	M1	M3	M4
	PC2/Key 3	PC6/Key 7	PD2/Key 15	PD6/Key 19
0	I/O pin			
1	Touch Key			

MnK2IO	M0	M1	M3	M4
	PC1/Key 2	PC5/Key 6	PD1/Key 14	PD5/Key 18
0	I/O pin			
1	Touch Key			

MnK1IO	M0	M1	M3	M4
	PC0/Key 1	PC4/Key 5	PD0/Key 13	PD4/Key 17
0	I/O pin			
1	Touch Key			

TKMnC2 Register

Bit	7	6	5	4	3	2	1	0
Name	Mn16CTON	D6	MnST	MnROEN	MnRCCLR	Mn16CTCLR	D1	MnROS
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 **Mn16CTON**: 16-bit C/F counter control
0: disable
1: enable

Bit 6 **D6**: This bit must be cleared to zero.

Bit 5 **MnST**: Time slot counter start control
0: time slot counter stopped
0 → 1: enable time slot counter.
When this bit changes from low to high the time slot counter will be enabled and the touch sense procedure started. When the time slot counter has completed its counting an interrupt will be generated.

Bit 4 **MnROEN**: Reference clock control
0: disable
1: enable

Bit 3 **MnRCCLR**: Time slot counter clear control
0: no change
1: clear counter
This bit must be first set to 1 and then to 0.

Bit 2 **Mn16CTCLR**: 16-bit C/F counter clear control
0: no change
1: clear counter
This bit must be first set to 1 and then to 0.

Bit 1 **D1**: This bit must be cleared to zero.

Bit 0 **MnROS**: Time slot counter clock source
0: reference clock
1: sense key oscillator
M0:K4, M1:K8, M2:K12, M3:K16, M4:K20

TKMnC3 Register

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	MnRCOV	Mn16CTOV	D3	MnROVS2	MnROVS1	MnROVS0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **D7, D6**: Read only bits -- unknown values

Bit 5 **MnRCOV**: Time slot counter overflow flag
0: no overflow
1: overflow

Bit 4 **Mn16CTOV**: 16-bit C/F counter overflow flag
0: no overflow
1: overflow

Bit 3 **D3**: This bit must be cleared to zero.

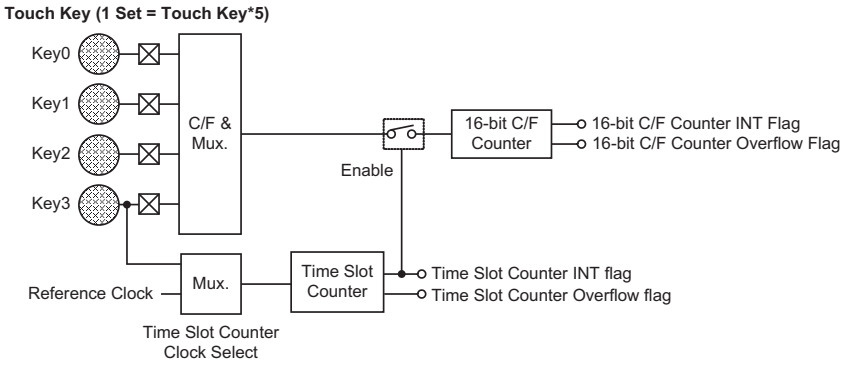
Bit 2~0 **MnROVS2~MnROVS0**: Time slot counter overflow time setup
000: 64 count
001: 128 count
010: 256 count
011: 512 count
100: 1024 count
101: 2048 count
110: 4096 count
111: 8192 count

Touch Key Operation

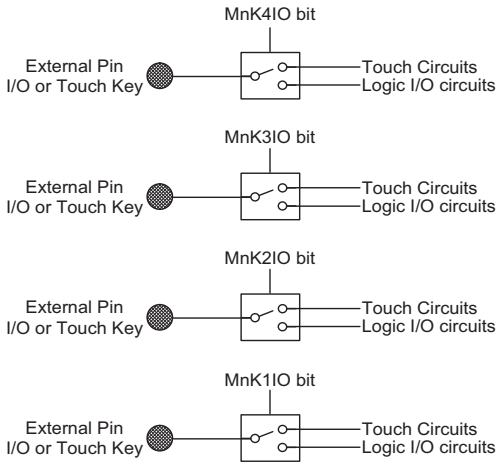
When a finger touches or is in proximity to a touch pad, the capacitance of the pad will increase. By using this capacitance variation to change slightly the frequency of the internal sense oscillator, touch actions can be sensed by measuring these frequency changes. Using an internal programmable divider the reference clock is used to generated a fixed time period. By counting the number of generated clock cycles from the sense oscillator during this fixed time period touch key actions can be determined.

Each touch key module contains four touch key inputs which are either dedicated touch key pins or are shared logical I/O pins. If shared, the desired function is selected using register bits. Each touch key has its own independent sense oscillator. There are therefore four sense oscillators within each touch key module. Each Touch Key module also has its own interrupt vector and set of interrupts flags.

During this reference clock fixed interval, the number of clock cycles generated by the sense oscillator is measured, and it is this value that is used to determine if a touch action has been made or not. At the end of the fixed reference clock time interval, a Touch Key interrupt signal will be generated.



Touch Switch Module Block Diagram



Touch Key or I/O Function Select

Touch Key Interrupt

Each touch key module, which consists of four touch keys, has two independent interrupts, one for each of the, 16-bit C/F counter and time slot counter.

The time slot counter interrupt has its own interrupt vector while the 16-bit C/F counter interrupts are contained within the Multi-function interrupts and therefore do not have their own vector. Care must be taken during programming as the 16-bit C/F counter interrupt flags contained within the Multi-function interrupts will not be automatically reset upon entry into the interrupt service routine but rather must be reset manually by the application program. More details regarding the touch key interrupts are located in the interrupt section of the datasheet.

Programming Considerations

After the relevant registers are setup, the touch key detection process is initiated the changing the MnST bit from low to high. This will enable and synchronise all relevant oscillators. The MnRCOV flag, which is the time slot counter flag will go high and remain high until the counter overflows. When this happens an interrupt signal will be generated.

When the external touch key size and layout are defined, their related capacitances will then determine the sensor oscillator frequency.

Serial Interface Module – SIM

These devices contain a Serial Interface Module, which includes both the four line SPI interface or the two line I^2C interface types, to allow an easy method of communication with external peripheral hardware. Having relatively simple communication protocols, these serial interface types allow the microcontroller to interface to external SPI or I^2C based hardware such as sensors, Flash or EEPROM memory, etc. The SIM pins are pin shared with other I/O pins and must be selected using the SIMEN bit in the SIMC0 register. As both interface types share the same pins and registers, the choice of whether the SPI or I^2C type is used is made using the SIM operating mode control bits, named SIM2~SIM0, in the SIMC0 register.

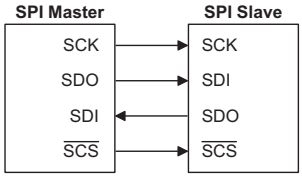
SPI Interface

The SPI interface is often used to communicate with external peripheral devices such as sensors, Flash or EEPROM memory devices etc. Originally developed by Motorola, the four line SPI interface is a synchronous serial data interface that has a relatively simple communication protocol simplifying the programming requirements when communicating with external hardware devices.

The communication is full duplex and operates as a slave/master type, where the device can be either master or slave. Although the SPI interface specification can control multiple slave devices from a single master, but this device provided only one $\overline{SCS}$ pin. If the master needs to control multiple slave devices from a single master, the master can use I/O pin to select the slave devices.

SPI Interface Operation

The SPI interface is a full duplex synchronous serial data link. It is a four line interface with pin names SDI, SDO, SCK and $\overline{SCS}$. Pins SDI and SDO are the Serial Data Input and Serial Data Output lines, SCK is the Serial Clock line and $\overline{SCS}$ is the Slave Select line. As the SPI interface pins are pin-shared with normal I/O pins and with the I^2C function pins, the SPI interface must first be enabled by setting the correct bits in the SIMC0 and SIMC2 registers. Communication between devices connected to the SPI interface is carried out in a slave/master mode with all data transfer initiations being implemented by the master. The Master also controls the clock signal. As the device only contains a single $\overline{SCS}$ pin only one slave device can be utilized. The $\overline{SCS}$ pin is controlled by software, set CSEN bit to "1" to enable $\overline{SCS}$ pin function, set CSEN bit to "0" the $\overline{SCS}$ pin will be as I/O function.

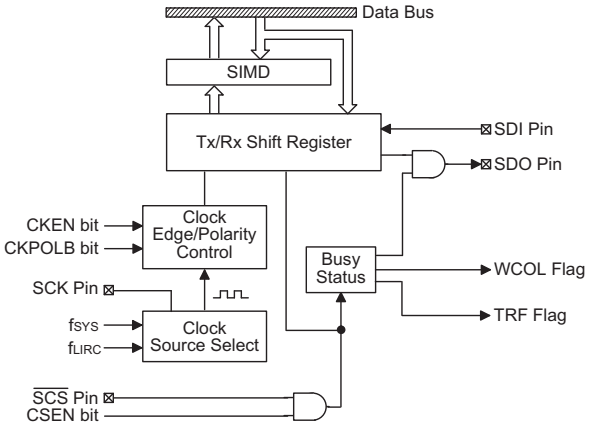


SPI Master/Slave Connection

The SPI function in this device offers the following features:

- Full duplex synchronous data transfer
- Both Master and Slave modes
- LSB first or MSB first data transmission modes
- Transmission complete flag
- Rising or falling active clock edge

The status of the SPI interface pins is determined by a number of factors such as whether the device is in the master or slave mode and upon the condition of certain control bits such as CSEN and SIMEN.



SPI Block Diagram

SPI Registers

There are three internal registers which control the overall operation of the SPI interface. These are the SIMD data register and two registers SIMC0 and SIMC2. Note that the SIMC1 register is only used by the I^2C interface.

Register Name	Bit							
	7	6	5	4	3	2	1	0
SIMC0	SIM2	SIM1	SIM0	PCKEN	PCKP1	PCKP0	SIMEN	—
SIMD	D7	D6	D5	D4	D3	D2	D1	D0
SIMC2	D7	D6	CKPOLB	CKEG	MLS	CSEN	WCOL	TRF

SPI Register List

The SIMD register is used to store the data being transmitted and received. The same register is used by both the SPI and I^2C functions. Before the device writes data to the SPI bus, the actual data to be transmitted must be placed in the SIMD register. After the data is received from the SPI bus, the device can read it from the SIMD register. Any transmission or reception of data from the SPI bus must be made via the SIMD register.

SIMD 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	x	x	x	x	x	x	x	x

"x" unknown

There are also two control registers for the SPI interface, SIMC0 and SIMC2. Note that the SIMC2 register also has the name SIMA which is used by the I²C function. The SIMC1 register is not used by the SPI function, only by the I²C function. Register SIMC0 is used to control the enable/disable function and to set the data transmission clock frequency. Although not connected with the SPI function, the SIMC0 register is also used to control the Peripheral Clock Prescaler. Register SIMC2 is used for other control functions such as LSB/MSB selection, write collision flag etc.

SIMC0 Register

Bit	7	6	5	4	3	2	1	0
Name	SIM2	SIM1	SIM0	PCKEN	PCKP1	PCKP0	SIMEN	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—
POR	1	1	1	0	0	0	0	—

Bit 7~5 **SIM2, SIM1, SIM0**: SIM Operating Mode Control
000: SPI master mode; SPI clock is $f_{SYS}/4$
001: SPI master mode; SPI clock is $f_{SYS}/16$
010: SPI master mode; SPI clock is $f_{SYS}/64$
011: SPI master mode; SPI clock is f_{LIRC}
100: Unused
101: SPI slave mode
110: I²C slave mode
111: Unused
These bits setup the overall operating mode of the SIM function. As well as selecting if the I²C or SPI function, they are used to control the SPI Master/Slave selection and the SPI Master clock frequency. The SPI clock is a function of the system clock but can also be chosen to be sourced from the TM0. If the SPI Slave Mode is selected then the clock will be supplied by an external Master device.

Bit 4 **PCKEN**: PCK Output Pin Control
0: Disable
1: Enable

Bit 3~2 **PCKP1, PCKP0**: Select PCK output pin frequency
00: f_{SYS}
01: $f_{SYS}/4$
10: $f_{SYS}/8$
11: TM0 CCRP match frequency/2

Bit 1 **SIMEN**: SIM Control
0: disable
1: enable
The bit is the overall on/off control for the SIM interface. When the SIMEN bit is cleared, the SDI, SDO, SCK and $\overline{SCS}$, or SDA and SCL lines will be as I/O function and the SIM operating current will be reduced to a minimum value. When the bit is high the SIM interface is enabled. If the SIM is configured to operate as an SPI interface via the SIM2~SIM0 bits, the contents of the SPI control registers will remain at the previous settings when the SIMEN bit changes from low to high and should therefore be first initialised by the application program. If the SIM is configured to operate as an I²C interface via the SIM2~SIM0 bits and the SIMEN bit changes from low to high, the contents of the I²C control bits such as HTX and TXAK will remain at the previous settings and should therefore be first initialised by the application program while the relevant I²C flags such as HCF, HAAS, HBB, SRW and RXAK will be set to their default states.

Bit 0 unimplemented, read as "0"

SIMC2 Register

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	CKPOLB	CKEG	MLS	CSEN	WCOL	TRF
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 Undefined bit
This bit can be read or written by user software program.

Bit 5 **CKPOLB**: Determines the base condition of the clock line
0: the SCK line will be high when the clock is inactive
1: the SCK line will be low when the clock is inactive
The CKPOLB bit determines the base condition of the clock line, if the bit is high, then the SCK line will be low when the clock is inactive. When the CKPOLB bit is low, then the SCK line will be high when the clock is inactive.

Bit 4 **CKEG**: Determines SPI SCK active clock edge type
CKPOLB=0
0: SCK is high base level and data capture at SCK rising edge
1: SCK is high base level and data capture at SCK falling edge
CKPOLB=1
0: SCK is low base level and data capture at SCK falling edge
1: SCK is low base level and data capture at SCK rising edge
The CKEG and CKPOLB bits are used to setup the way that the clock signal outputs and inputs data on the SPI bus. These two bits must be configured before data transfer is executed otherwise an erroneous clock edge may be generated. The CKPOLB bit determines the base condition of the clock line, if the bit is high, then the SCK line will be low when the clock is inactive. When the CKPOLB bit is low, then the SCK line will be high when the clock is inactive. The CKEG bit determines active clock edge type which depends upon the condition of CKPOLB bit.

Bit 3 **MLS**: SPI Data shift order
0: LSB
1: MSB
This is the data shift select bit and is used to select how the data is transferred, either MSB or LSB first. Setting the bit high will select MSB first and low for LSB first.

Bit 2 **CSEN**: SPI $\overline{\text{SCS}}$ pin Control
0: Disable
1: Enable
The CSEN bit is used as an enable/disable for the $\overline{\text{SCS}}$ pin. If this bit is low, then the $\overline{\text{SCS}}$ pin will be disabled and as I/O function. If the bit is high the $\overline{\text{SCS}}$ pin will be enabled and used as a select pin.

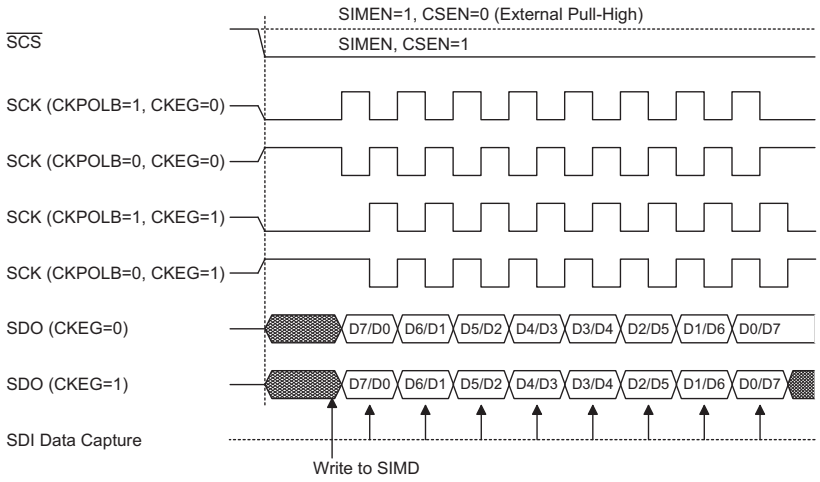
Bit 1 **WCOL**: SPI Write Collision flag
0: No collision
1: Collision
The WCOL flag is used to detect if a data collision has occurred. If this bit is high it means that data has been attempted to be written to the SIMD register during a data transfer operation. This writing operation will be ignored if data is being transferred. The bit can be cleared by the application program.

Bit 0 **TRF**: SPI Transmit/Receive Complete flag
0: Data is being transferred
1: SPI data transmission is completed
The TRF bit is the Transmit/Receive Complete flag and is set ″1″ automatically when an SPI data transmission is completed, but must set to ″0″ by the application program. It can be used to generate an interrupt.

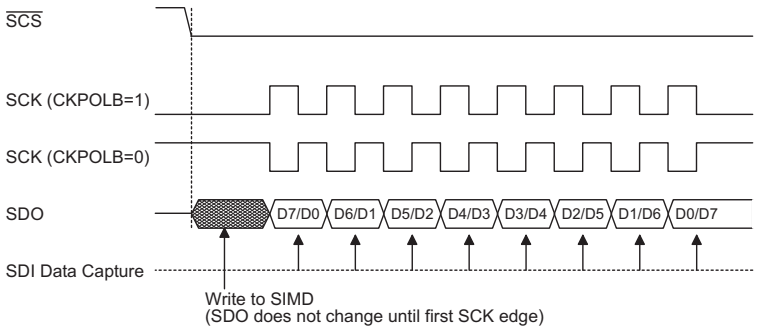
SPI Communication

After the SPI interface is enabled by setting the SIMEN bit high, then in the Master Mode, when data is written to the SIMD register, transmission/reception will begin simultaneously. When the data transfer is complete, the TRF flag will be set automatically, but must be cleared using the application program. In the Slave Mode, when the clock signal from the master has been received, any data in the SIMD register will be transmitted and any data on the SDI pin will be shifted into the SIMD register. The master should output an $\overline{SCS}$ signal to enable the slave device before a clock signal is provided. The slave data to be transferred should be well prepared at the appropriate moment relative to the $\overline{SCS}$ signal depending upon the configurations of the CKPOLB bit and CKEG bit. The accompanying timing diagram shows the relationship between the slave data and $\overline{SCS}$ signal for various configurations of the CKPOLB and CKEG bits.

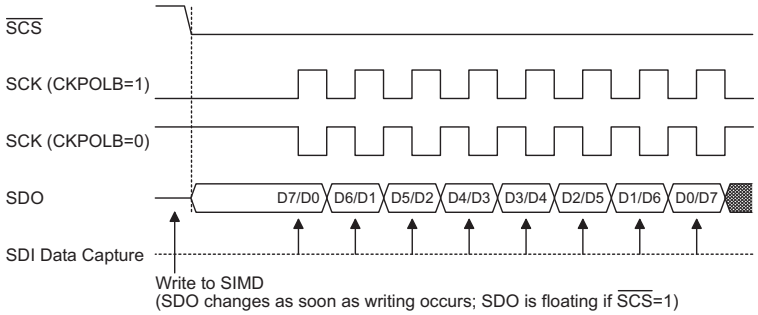
The SPI will continue to function even in the IDLE Mode.



SPI Master Mode Timing

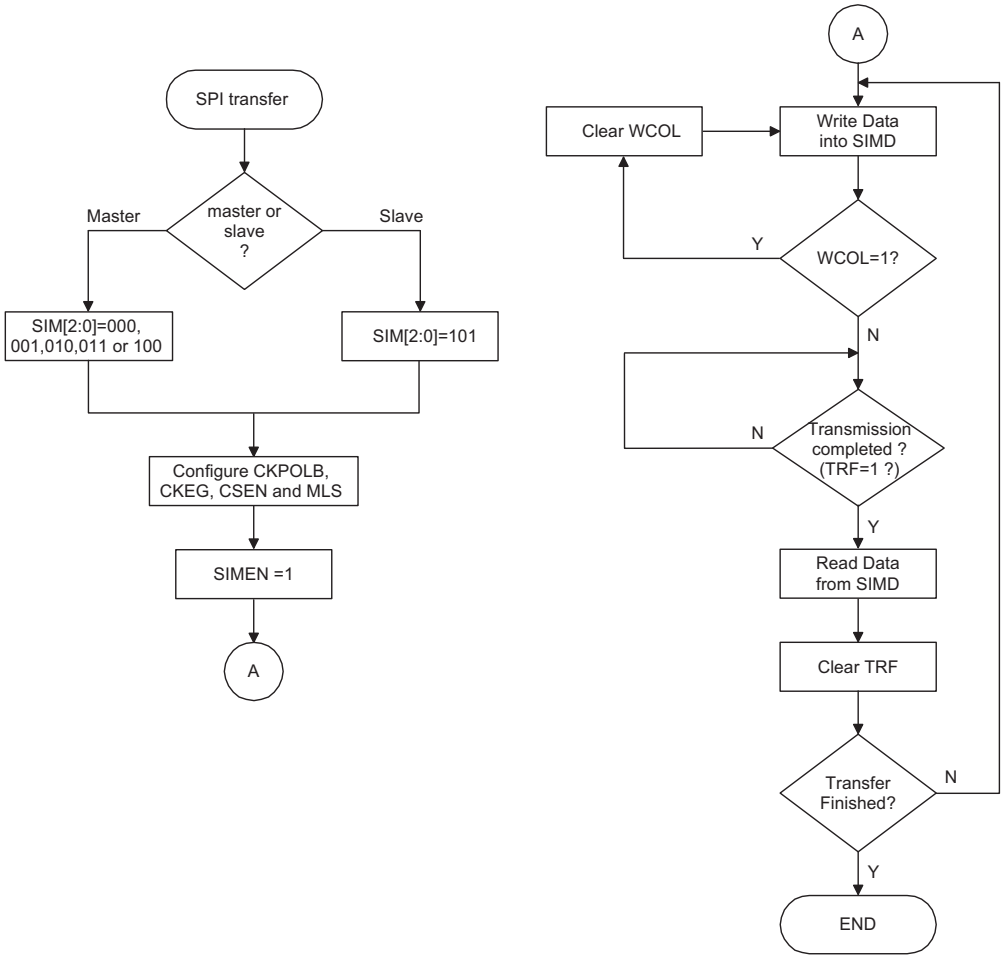


SPI Slave Mode Timing – CKEG=0



Note: For SPI slave mode, if SIMEN=1 and CSEN=0, SPI is always enabled and ignores the SCS level.

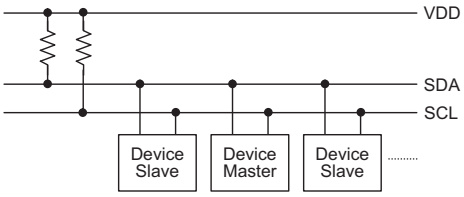
SPI Slave Mode Timing – CKEG=1



SPI Transfer Control Flowchart

I²C Interface

The I²C interface is used to communicate with external peripheral devices such as sensors, EEPROM memory etc. Originally developed by Philips, it is a two line low speed serial interface for synchronous serial data transfer. The advantage of only two lines for communication, relatively simple communication protocol and the ability to accommodate multiple devices on the same bus has made it an extremely popular interface type for many applications.

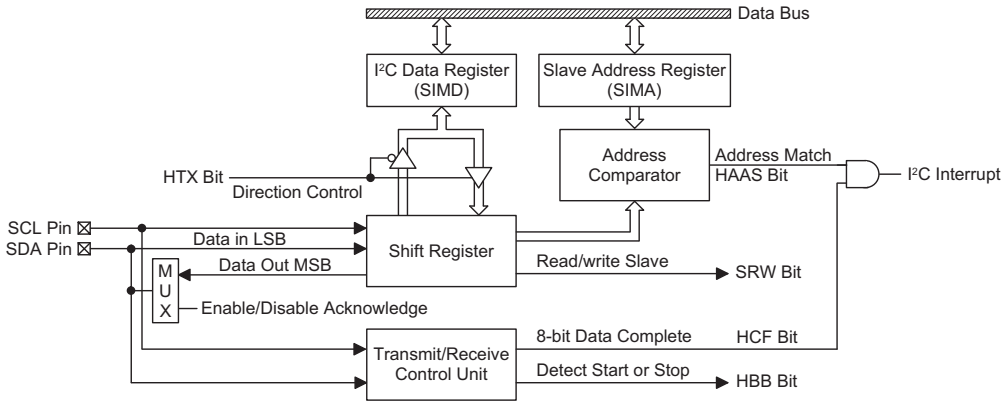


I²C Master Slave Bus Connection

I²C Interface Operation

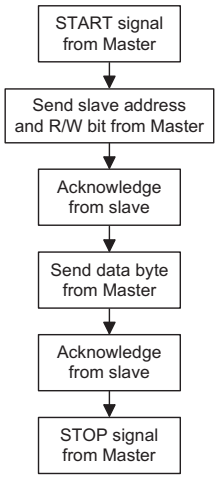
The I²C serial interface is a two line interface, a serial data line, SDA, and serial clock line, SCL. As many devices may be connected together on the same bus, their outputs are both open drain types. For this reason it is necessary that external pull-high resistors are connected to these outputs. Note that no chip select line exists, as each device on the I²C bus is identified by a unique address which will be transmitted and received on the I²C bus.

When two devices communicate with each other on the bidirectional I²C bus, one is known as the master device and one as the slave device. Both master and slave can transmit and receive data, however, it is the master device that has overall control of the bus. For these devices, which only operates in slave mode, there are two methods of transferring data on the I²C bus, the slave transmit mode and the slave receive mode.



I²C Block Diagram

The debounce time, if selected, can be chosen to be either 1 or 2 system clocks.



I²C Registers

There are four control registers associated with the I²C bus, SIMC0, SIMC1, SIMA and I2CTOC and one data register, SIMD. The SIMD register, which is shown in the above SPI section, is used to store the data being transmitted and received on the I²C bus. Before the microcontroller writes data to the I²C bus, the actual data to be transmitted must be placed in the SIMD register. After the data is received from the I²C bus, the microcontroller can read it from the SIMD register. Any transmission or reception of data from the I²C bus must be made via the SIMD register. The SIM pins are pin shared with other I/O pins and must be selected using the SIMEN bit in the SIMC0 register.

Note that the SIMA register also has the name SIMC2 which is used by the SPI function. Bit SIMEN and bits SIM2~SIM0 in register SIMC0 are used by the I²C interface.

Register Name	Bit							
	7	6	5	4	3	2	1	0
SIMC0	SIM2	SIM1	SIM0	PCKEN	PCKP1	PCKP0	SIMEN	—
SIMC1	HCF	HAAS	HBB	HTX	TXAK	SRW	IAMWU	RXAK
SIMD	D7	D6	D5	D4	D3	D2	D1	D0
SIMA	IICA6	IICA5	IICA4	IICA3	IICA2	IICA1	IICA0	D0
I2CTOC	I2CTOEN	I2CTOF	I2CTOS5	I2CTOS4	I2CTOS3	I2CTOS2	I2CTOS1	I2CTOS0

I²C Register List

SIMC0 Register

Bit	7	6	5	4	3	2	1	0
Name	SIM2	SIM1	SIM0	PCKEN	PCKP1	PCKP0	SIMEN	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—
POR	1	1	1	0	0	0	0	—

Bit 7~5 **SIM2, SIM1, SIM0**: SIM Operating Mode Control
000: SPI master mode; SPI clock is $f_{SYS}/4$
001: SPI master mode; SPI clock is $f_{SYS}/16$
010: SPI master mode; SPI clock is $f_{SYS}/64$
011: SPI master mode; SPI clock is f_{LIRC}
100: SPI master mode; SPI clock is TM0 CCRP match frequency/2
101: SPI slave mode
110: I^2C slave mode
111: Unused mode
These bits setup the overall operating mode of the SIM function. As well as selecting if the I^2C or SPI function, they are used to control the SPI Master/Slave selection and the SPI Master clock frequency. The SPI clock is a function of the system clock but can also be chosen to be sourced from the TM0. If the SPI Slave Mode is selected then the clock will be supplied by an external Master device.

Bit 4 **PCKEN**: PCK Output Pin Control
Described elsewhere

Bit 3~2 **PCKP1, PCKP0**: Select PCK output pin frequency
Described elsewhere

Bit 1 **SIMEN**: SIM Control
0: disable
1: enable
The bit is the overall on/off control for the SIM interface. When the SIMEN bit is cleared, the SDI, SDO, SCK and $\overline{SCS}$, or SDA and SCL lines will be as I/O function and the SIM operating current will be reduced to a minimum value. When the bit is high the SIM interface is enabled. If the SIM is configured to operate as an SPI interface via the SIM2~SIM0 bits, the contents of the SPI control registers will remain at the previous settings when the SIMEN bit changes from low to high and should therefore be first initialised by the application program. If the SIM is configured to operate as an I^2C interface via the SIM2~SIM0 bits and the SIMEN bit changes from low to high, the contents of the I^2C control bits such as HTX and TXAK will remain at the previous settings and should therefore be first initialised by the application program while the relevant I^2C flags such as HCF, HAAS, HBB, SRW and RXAK will be set to their default states.

Bit 0 unimplemented, read as ″0″

SIMC1 Register

Bit	7	6	5	4	3	2	1	0
Name	HCF	HAAS	HBB	HTX	TXAK	SRW	IAMWU	RXAK
R/W	R	R	R	R/W	R/W	R	R/W	R
POR	1	0	0	0	0	0	0	1

Bit 7 **HCF**: I^2C Bus data transfer completion flag
0: Data is being transferred
1: Completion of an 8-bit data transfer
The HCF flag is the data transfer flag. This flag will be zero when data is being transferred. Upon completion of an 8-bit data transfer the flag will go high and an interrupt will be generated.

Bit 6 **HAAS**: I^2C Bus address match flag
0: Not address match
1: Address match
The HASS flag is the address match flag. This flag is used to determine if the slave device address is the same as the master transmit address. If the addresses match then this bit will be high, if there is no match then the flag will be low.

Bit 5 **HBB**: I^2C Bus busy flag
0: I^2C Bus is not busy
1: I^2C Bus is busy
The HBB flag is the I^2C busy flag. This flag will be ″1″ when the I^2C bus is busy which will occur when a START signal is detected. The flag will be set to ″0″ when the bus is free which will occur when a STOP signal is detected.

Bit 4 **HTX**: Select I^2C slave device is transmitter or receiver
0: Slave device is the receiver
1: Slave device is the transmitter

Bit 3 **TXAK**: I^2C Bus transmit acknowledge flag
0: Slave send acknowledge flag
1: Slave do not send acknowledge flag
The TXAK bit is the transmit acknowledge flag. After the slave device receipt of 8-bits of data, this bit will be transmitted to the bus on the 9th clock from the slave device. The slave device must always set TXAK bit to ″0″ before further data is received.

Bit 2 **SRW**: I^2C Slave Read/Write flag
0: Slave device should be in receive mode
1: Slave device should be in transmit mode
The SRW flag is the I^2C Slave Read/Write flag. This flag determines whether the master device wishes to transmit or receive data from the I^2C bus. When the transmitted address and slave address is match, that is when the HAAS flag is set high, the slave device will check the SRW flag to determine whether it should be in transmit mode or receive mode. If the SRW flag is high, the master is requesting to read data from the bus, so the slave device should be in transmit mode. When the SRW flag is zero, the master will write data to the bus, therefore the slave device should be in receive mode to read this data.

Bit 1 **IAMWU**: I^2C address match wake-up control
0: disable
1: enable
This bit should be set to ″1″ to enable I^2C address match wake-up from SLEEP or IDLE Mode.

Bit 0 **RXAK**: I^2C Bus Receive acknowledge flag
0: Slave receive acknowledge flag
1: Slave do not receive acknowledge flag
The RXAK flag is the receiver acknowledge flag. When the RXAK flag is ″0″, it means that a acknowledge signal has been received at the 9th clock, after 8 bits of data have been transmitted. When the slave device in the transmit mode, the slave device checks the RXAK flag to determine if the master receiver wishes to receive the next byte. The slave transmitter will therefore continue sending out data until the RXAK flag is ″1″. When this occurs, the slave transmitter will release the SDA line to allow the master to send a STOP signal to release the I^2C Bus.

I2CTOC Register

Bit	7	6	5	4	3	2	1	0
Name	I2CTOEN	I2CTOF	I2CTOS5	I2CTOS4	I2CTOS3	I2CTOS2	I2CTOS1	I2CTOS0
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 **I2CTOEN**: I^2C Time-out Control
0: disable
1: enable

Bit 6 **I2CTOF**: Time-out flag
0: no time-out
1: time-out occurred

Bit 5~0 **I2CTOS5~I2CTOS0**: Time-Out Time Definition
I^2C time-out clock source is $f_{LIRC}/32$.
I^2C Time-Out time is given by: [I2CTOS5 : I2CTOS0]+1) x $(32/f_{LIRC})$

The SIMD register is used to store the data being transmitted and received. The same register is used by both the SPI and I^2C functions. Before the device writes data to the I^2C bus, the actual data to be transmitted must be placed in the SIMD register. After the data is received from the I^2C bus, the device can read it from the SIMD register. Any transmission or reception of data from the I^2C bus must be made via the SIMD register.

SIMD 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	x	x	x	x	x	x	x	x

"x" unknown

SIMA Register

Bit	7	6	5	4	3	2	1	0
Name	IICA6	IICA5	IICA4	IICA3	IICA2	IICA1	IICA0	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	x	x	x	x	x

"x" unknown

Bit 7~1 **IICA6~ IICA0**: I^2C slave address
IICA6~ IICA0 is the I^2C slave address bit 6~bit 0.
The SIMA register is also used by the SPI interface but has the name SIMC2. The SIMA register is the location where the 7-bit slave address of the slave device is stored. Bits 7~1 of the SIMA register define the device slave address. Bit 0 is not defined.
When a master device, which is connected to the I^2C bus, sends out an address, which matches the slave address in the SIMA register, the slave device will be selected. Note that the SIMA register is the same register address as SIMC2 which is used by the SPI interface.

Bit 0 Undefined bit
This bit can be read or written by user software program.

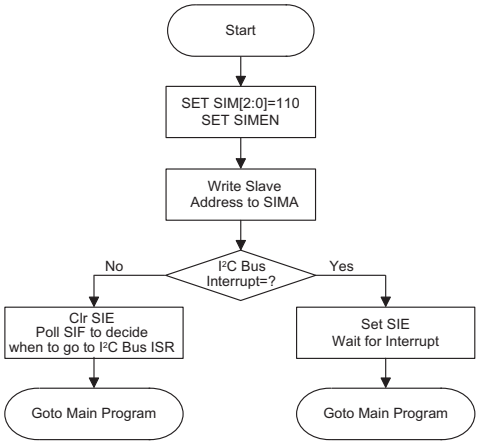
I²C Bus Communication

Communication on the I²C bus requires four separate steps, a START signal, a slave device address transmission, a data transmission and finally a STOP signal. When a START signal is placed on the I²C bus, all devices on the bus will receive this signal and be notified of the imminent arrival of data on the bus. The first seven bits of the data will be the slave address with the first bit being the MSB. If the address of the slave device matches that of the transmitted address, the HAAS bit in the SIMC1 register will be set and an I²C interrupt will be generated. After entering the interrupt service routine, the slave device must first check the condition of the HAAS bit to determine whether the interrupt source originates from an address match or from the completion of an 8-bit data transfer. During a data transfer, note that after the 7-bit slave address has been transmitted, the following bit, which is the 8th bit, is the read/write bit whose value will be placed in the SRW bit. This bit will be checked by the slave device to determine whether to go into transmit or receive mode. Before any transfer of data to or from the I²C bus, the microcontroller must initialise the bus, the following are steps to achieve this:

Step 1
Set the SIM2~SIM0 and SIMEN bits in the SIMC0 register to "1" to enable the I²C bus.

Step 2
Write the slave address of the device to the I²C bus address register SIMA.

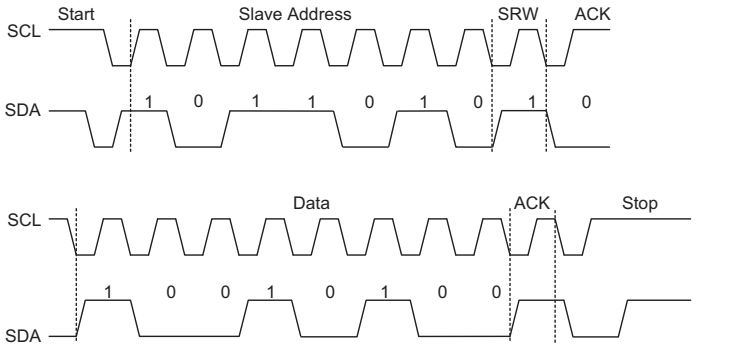
Step 3
Set the SIME and SIM Muti-Function interrupt enable bit of the interrupt control register to enable the SIM interrupt and Multi-function interrupt.



I²C Bus Initialisation Flow Chart

I²C Bus Start Signal

The START signal can only be generated by the master device connected to the I²C bus and not by the slave device. This START signal will be detected by all devices connected to the I²C bus. When detected, this indicates that the I²C bus is busy and therefore the HBB bit will be set. A START condition occurs when a high to low transition on the SDA line takes place when the SCL line remains high.



S=Start (1 bit)
SA=Slave Address (7 bits)
SR=SRW bit (1 bit)
M=Slave device send acknowledge bit (1 bit)
D=Data (8 bits)
A=ACK (RXAK bit for transmitter, TXAK bit for receiver 1 bit)
P=Stop (1 bit)

S	SA	SR	M	D	A	D	A	--------	S	SA	SR	M	D	A	D	A	--------	P

Note: * When a slave address is matched, the device must be placed in either the transmit mode and then write data to the SIMD register, or in the receive mode where it must implement a dummy read from the SIMD register to release the SCL line.

I²C Communication Timing Diagram

Slave Address

The transmission of a START signal by the master will be detected by all devices on the I²C bus. To determine which slave device the master wishes to communicate with, the address of the slave device will be sent out immediately following the START signal. All slave devices, after receiving this 7-bit address data, will compare it with their own 7-bit slave address. If the address sent out by the master matches the internal address of the microcontroller slave device, then an internal I²C bus interrupt signal will be generated. The next bit following the address, which is the 8th bit, defines the read/write status and will be saved to the SRW bit of the SIMC1 register. The slave device will then transmit an acknowledge bit, which is a low level, as the 9th bit. The slave device will also set the status flag HAAS when the addresses match.

As an I²C bus interrupt can come from two sources, when the program enters the interrupt subroutine, the HAAS bit should be examined to see whether the interrupt source has come from a matching slave address or from the completion of a data byte transfer. When a slave address is matched, the device must be placed in either the transmit mode and then write data to the SIMD register, or in the receive mode where it must implement a dummy read from the SIMD register to release the SCL line.

I²C Bus Read/Write Signal

The SRW bit in the SIMC1 register defines whether the slave device wishes to read data from the I²C bus or write data to the I²C bus. The slave device should examine this bit to determine if it is to be a transmitter or a receiver. If the SRW flag is "1" then this indicates that the master device wishes to read data from the I²C bus, therefore the slave device must be setup to send data to the I²C bus as a transmitter. If the SRW flag is "0" then this indicates that the master wishes to send data to the I²C bus, therefore the slave device must be setup to read data from the I²C bus as a receiver.

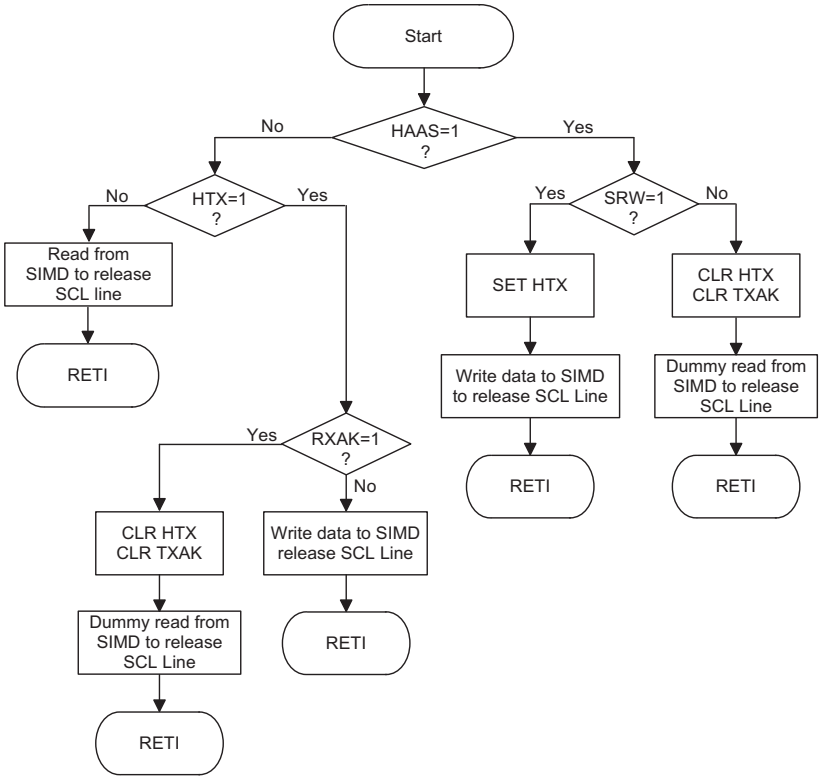
I²C Bus Slave Address Acknowledge Signal

After the master has transmitted a calling address, any slave device on the I²C bus, whose own internal address matches the calling address, must generate an acknowledge signal. The acknowledge signal will inform the master that a slave device has accepted its calling address. If no acknowledge signal is received by the master then a STOP signal must be transmitted by the master to end the communication. When the HAAS flag is high, the addresses have matched and the slave device must check the SRW flag to determine if it is to be a transmitter or a receiver. If the SRW flag is high, the slave device should be setup to be a transmitter so the HTX bit in the SIMC1 register should be set to "1". If the SRW flag is low, then the microcontroller slave device should be setup as a receiver and the HTX bit in the SIMC1 register should be set to "0".

I²C Bus Data and Acknowledge Signal

The transmitted data is 8-bits wide and is transmitted after the slave device has acknowledged receipt of its slave address. The order of serial bit transmission is the MSB first and the LSB last. After receipt of 8-bits of data, the receiver must transmit an acknowledge signal, level "0", before it can receive the next data byte. If the slave transmitter does not receive an acknowledge bit signal from the master receiver, then the slave transmitter will release the SDA line to allow the master to send a STOP signal to release the I²C Bus. The corresponding data will be stored in the SIMD register. If setup as a transmitter, the slave device must first write the data to be transmitted into the SIMD register. If setup as a receiver, the slave device must read the transmitted data from the SIMD register.

When the slave receiver receives the data byte, it must generate an acknowledge bit, known as TXAK, on the 9th clock. The slave device, which is setup as a transmitter will check the RXAK bit in the SIMC1 register to determine if it is to send another data byte, if not then it will release the SDA line and await the receipt of a STOP signal from the master.

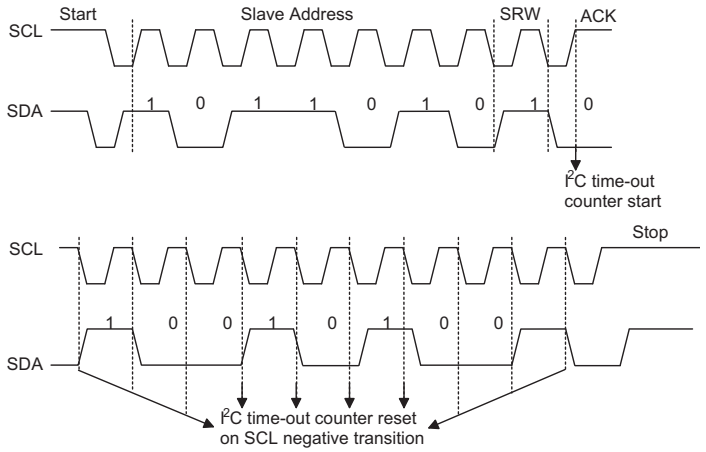


I²C Bus ISR Flow Chart

I²C Time-out Control

In order to reduce the problem of I²C lockup due to reception of erroneous clock sources, clock, a time-out function is provided. If the clock source to the I²C is not received then after a fixed time period, the I²C circuitry and registers will be reset.

The time-out counter starts counting on an I²C bus "START" & "address match" condition, and is cleared by an SCL falling edge. Before the next SCL falling edge arrives, if the time elapsed is greater than the time-out setup by the I2CTOC register, then a time-out condition will occur. The time-out function will stop when an I²C "STOP" condition occurs.



I²C Time-out

When an I²C time-out counter overflow occurs, the counter will stop and the I2CTOEN bit will be cleared to zero and the I2CTF bit will be set high to indicate that a time-out condition as occurred. The time-out condition will also generate an interrupt which uses the I²C interrrupt vector. When an I²C time-out occurs the I²C internal circuitry will be reset and the registers will be reset into the following condition:

Register	After I²C Time-out
SIMDR, SIMAR, SIMC0	No change
SIMC1	Reset to POR condition

I²C Registers After Time-out

The I2CTOF flag can be cleared by the application program. There are 64 time-out periods which can be selected using bits in the I2CTOC register. The time-out time is given by the formula:

((1~64) × 32) / f_{LIRC}. This gives a range of about 1ms to 64ms. Note also that the LIRC oscillator is continuously enabled.

Peripheral Clock Output

The Peripheral Clock Output allows the device to supply external hardware with a clock signal synchronised to the microcontroller clock.

Peripheral Clock Operation

As the peripheral clock output pin, PCK, is shared with I/O line, the required pin function is chosen via PCKEN in the SIMC0 register. The Peripheral Clock function is controlled using the SIMC0 register. The clock source for the Peripheral Clock Output can originate from either the TM0 CCRP match frequency/2 or a divided ratio of the internal f_{SYS} clock. The PCKEN bit in the SIMC0 register is the overall on/off control, setting PCKEN bit to ″1″ enables the Peripheral Clock, setting PCKEN bit to ″0″ disables it. The required division ratio of the system clock is selected using the PCKP1 and PCKP0 bits in the same register. If the device enters the SLEEP Mode this will disable the Peripheral Clock output.

SIMC0 Register

Bit	7	6	5	4	3	2	1	0
Name	SIM2	SIM1	SIM0	PCKEN	PCKPSC1	PCKPSC0	SIMEN	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—
POR	1	1	1	0	0	0	0	—

Bit 7~5 **SIM2~SIM0**: SIM Operating Mode Control
Described elsewhere

Bit 4 **PCKEN**: PCK Output Pin Control
0: disable
1: enable

Bit 3~2 **PCKPSC1**, **PCKPSC0**: Select PCK output pin frequency
0: f_{SYS}
1: $f_{SYS}/4$
2: $f_{SYS}/8$
3: TM0 CCRP match frequency/2

Bit 1 **SIMEN**: SIM Control
Described elsewhere

Bit 0 Unimplemented, read as ″0″

Interrupts

Interrupts are an important part of any microcontroller system. When an external event or an internal function such as a Touch Action or Timer Module requires microcontroller attention, their corresponding interrupt will enforce a temporary suspension of the main program allowing the microcontroller to direct attention to their respective needs. The devices contains several external interrupt and internal interrupts functions. The external interrupt is generated by the action of the external INT pin, while the internal interrupts are generated by various internal functions such as the Touch Keys, Timer Module, Time Base, SIM 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 depends upon the device chosen but fall into three categories. The first is the INTC0~INTC3 registers which setup the primary interrupts, the second is the MFI0~MFI5 registers which setup the Multi-function interrupts. Finally there is an INTEG register to setup the external interrupt 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	—	—
INT Pin	INTnE	INTnF	N=0 or 1
Touch Key Module	TKMnE	TKMnF	n=0~4
SIM	SIM	SIF	—
EEPROM	DEE	DEF	—
Multi-function	MFnE	MFnF	n=0~5
Time Base	TBnE	TBnF	N=0 or 1
LVD	LVF	LVE	—
External Peripheral	XPE	XPF	—
TM	TnPE	TnPF	n=0~2
	TnAE	TnAF	n=0~2
	TnBE	TnBF	n=0~2

Interrupt Register Bit Naming Conventions

Interrupt Register Contents

BS85B12-3

Name	Bit							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	SIMF	INT1F	INT0F	SIME	INT1E	INT0E	EMI
INTC1	TB0F	TKM2F	TKM1F	TKM0F	TB0E	TKM2E	TKM1E	TKM0E
INTC2	MF3F	MF2F	MF1F	MF0F	MF3E	MF2E	MF1E	MF0E
MFI0	M116CTF	D6	M016CTF	D4	M116CTE	D2	M016CTE	D0
MFI1	T0AF	T0PF	M216CTF	D4	T0AE	T0PE	M216CTE	D0
MFI2	—	T1BF	T1AF	T1PF	—	T1BE	T1AE	T1PE
MFI3	DEF	LVF	XPF	TB1F	DEE	LVE	XPE	TB1E

BS85C20-3

Name	Bit							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	SIMF	INT1F	INT0F	SIME	INT1E	INT0E	EMI
INTC1	TB0F	TKM2F	TKM1F	TKM0F	TB0E	TKM2E	TKM1E	TKM0E
INTC2	MF3F	MF2F	MF1F	MF0F	MF3E	MF2E	MF1E	MF0E
INTC3	MF5F	MF4F	TKM4F	TKM3F	MF5E	MF4E	TKM4E	TKM3E
MFI0	M116CTF	D6	M016CTF	D4	M116CTE	D2	M016CTE	D0
MFI1	T0AF	T0PF	M216CTF	D4	T0AE	T0PE	M216CTE	D0
MFI2	—	T1BF	T1AF	T1PF	—	T1BE	T1AE	T1PE
MFI3	DEF	LVF	XPF	TB1F	DEE	LVE	XPE	TB1E
MFI4	M416CTF	D6	M316CTF	D4	M416CTE	D2	M316CTE	D0
MFI5	—	—	T2AF	T2PF	—	—	T2AE	T2PE

INTEG Register – All devices

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 7~2 unimplemented, read as "0"

Bit 1~0 **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 – All devices

Bit	7	6	5	4	3	2	1	0
Name	—	SIMF	INT1F	INT0F	SIME	INT1E	INT0E	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 **SIMF**: SIM interrupt request flag
0: no request
1: interrupt request

Bit 5 **INT1F**: INT1 pin interrupt request flag
0: no request
1: interrupt request

Bit 4 **INT0F**: INT0 pin interrupt request flag
0: No request
1: Interrupt request

Bit 3 **SIME**: SIM interrupt control
0: disable
1: enable

Bit 2 **INT1E**: INT1 pin interrupt control
0: disable
1: enable

Bit 1 **INT0E**: INT0 pin interrupt control
0: disable
1: enable

Bit 0 **EMI**: Global interrupt control
0: disable
1: enable

INTC1 Register – All devices

Bit	7	6	5	4	3	2	1	0
Name	TB0F	TKM2F	TKM1F	TKM0F	TB0E	TKM2E	TKM1E	TKM0E
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 **TB0F**: Time Base 0 interrupt request flag
0: no request
1: interrupt request

Bit 6 **TKM2F**: Touch Key Module 2 interrupt request flag
0: no request
1: interrupt request

Bit 5 **TKM1F**: Touch Key Module 1 interrupt request flag
0: no request
1: interrupt request

Bit 4 **TKM0F**: Touch Key Module 0 interrupt request flag
0: no request
1: interrupt request

Bit 3 **TB0E**: Time Base 0 interrupt control
0: disable
1: enable

Bit 2 **TKM2E**: Touch Key Module 2 interrupt control
0: disable
1: enable

Bit 1 **TKM1E**: Touch Key Module 1 interrupt control
0: disable
1: enable

Bit 0 **TKM0E**: Touch Key Module 0 interrupt control
0: disable
1: enable

INTC2 Register – All devices

Bit	7	6	5	4	3	2	1	0
Name	MF3F	MF2F	MF1F	MF0F	MF3E	MF2E	MF1E	MF0E
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 **MF3F**: Multi-function interrupt 3 request flag
0: no request
1: interrupt request

Bit 6 **MF2F**: Multi-function interrupt 2 request flag
0: no request
1: interrupt request

Bit 5 **MF1F**: Multi-function interrupt 1 request flag
0: no request
1: interrupt request

Bit 4 **MF0F**: Multi-function interrupt 0 request flag
0: no request
1: interrupt request

Bit 3 **MF3E**: Multi-function interrupt 3 control
0: disable
1: enable

Bit 2 **MF2E**: Multi-function interrupt 2 control
0: disable
1: enable

Bit 1 **MF1E**: Multi-function interrupt 1 control
0: disable
1: enable

Bit 0 **MF0E**: Multi-function interrupt 0 control
0: disable
1: enable

INTC3 Register – BS85C20 only

Bit	7	6	5	4	3	2	1	0
Name	MF5F	MF4F	TKM4F	TKM3F	MF5E	MF4E	TKM4E	TKM3E
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 **MF5F**: Multi-function interrupt 5 request flag
0: no request
1: interrupt request

Bit 6 **MF4F**: Multi-function interrupt 4 request flag
0: no request
1: interrupt request

Bit 5 **TKM4F**: Touch Key Module 4 interrupt request flag
0: no request
1: interrupt request

Bit 4 **TKM3F**: Touch Key Module 3 interrupt request flag
0: no request
1: interrupt request

Bit 3 **MF5E**: Multi-function interrupt 5 control
0: disable
1: enable

Bit 2 **MF4E**: Multi-function interrupt 4 control
0: disable
1: enable

Bit 1 **TKM4E**: Touch Key Module 4 interrupt control
0: disable
1: enable

bit 0 **TKM3E**: Touch Key Module 3 interrupt control
0: disable
1: enable

MFI0 Register – All devices

Bit	7	6	5	4	3	2	1	0
Name	M116CTF	D6	M016CTF	D4	M116CTE	D2	M016CTE	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 **M116CTF**: Touch Key Module 1 16-bit counter interrupt request flag
0: no request
1: interrupt request

Bit 6 **D6**: This bit must be cleared to zero

Bit 5 **M016CTF**: Touch Key Module 0 16-bit counter interrupt request flag
0: no request
1: interrupt request

Bit 4 **D4**: This bit must be cleared to zero

Bit 3 **M116CTE**: Touch Key Module 1 16-bit timer interrupt control
0: disable
1: enable

Bit 2 **D2**: This bit must be cleared to zero

Bit 1 **M016CTE**: Touch Key Module 0 16-bit timer interrupt control
0: disable
1: enable

Bit 0 **D0**: This bit must be cleared to zero

MFI1 Register – All devices

Bit	7	6	5	4	3	2	1	0
Name	T0AF	T0PF	M216CTF	D4	T0AE	T0PE	M216CTE	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 **T0AF**: TM0 Comparator A match interrupt request flag
0: no request
1: interrupt request

Bit 6 **T0PF**: TM0 Comparator P match interrupt request flag
0: no request
1: interrupt request

Bit 5 **M216CTF**: Touch Key Module 2 16-bit counter interrupt request flag
0: no request
1: interrupt request

Bit 4 **D4**: This bit must be cleared to zero

Bit 3 **T0AE**: TM0 Comparator A match interrupt control
0: disable
1: enable

Bit 2 **T0PE**: TM0 Comparator P match interrupt control
0: disable
1: enable

Bit 1 **M216CTE**: Touch Key Module 2 16-bit counter interrupt control
0: disable
1: enable

Bit 0 **D0**: This bit must be cleared to zero

MFI2 Register – All devices

Bit	7	6	5	4	3	2	1	0
Name	—	T1BF	T1AF	T1PF	—	T1BE	T1AE	T1PE
R/W	—	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	—	0	0	0	—	0	0	0

Bit 7 unimplemented, read as "0"

Bit 6 **T1BF**: TM1 Comparator B match interrupt request flag
0: no request
1: interrupt request

Bit 5 **T1AF**: TM1 Comparator A match interrupt request flag
0: no request
1: interrupt request

Bit 4 **T1PF**: TM1 Comparator P match interrupt request flag
0: no request
1: interrupt request

Bit 3 unimplemented, read as "0"

Bit 2 **T1BE**: TM1 Comparator B match interrupt control
0: disable
1: enable

Bit 1 **T1AE**: TM1 Comparator A match interrupt control
0: disable
1: enable

Bit 0 **T1PE**: TM1 Comparator P match interrupt control
0: disable
1: enable

MFI3 Register – All devices

Bit	7	6	5	4	3	2	1	0
Name	DEF	LVF	XPF	TB1F	DEE	LVE	XPE	TB1E
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 **DEF**: Data EEPROM interrupt request flag
0: no request
1: interrupt request

Bit 6 **LVF**: LVD interrupt request flag
0: no request
1: interrupt request

Bit 5 **XPF**: External peripheral interrupt request flag
0: no request
1: interrupt request

Bit 4 **TB1F**: Time Base 1 interrupt request flag
0: no request
1: interrupt request

Bit 3 **DEE**: Data EEPROM interrupt control
0: disable
1: enable

Bit 2 **LVE**: LVD interrupt control
0: disable
1: enable

Bit 1 **XPE**: External Peripheral interrupt control
0: disable
1: enable

Bit 0 **TB1E**: Time Base 1 interrupt control
0: disable
1: enable

MFI4 Register – BS85C20 only

Bit	7	6	5	4	3	2	1	0
Name	M416CTF	D6	M316CTF	D4	M416CTE	D2	M316CTE	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 **M416CTF**: Touch Key Module 4 16-bit counter interrupt request flag
0: no request
1: interrupt request

Bit 6 **D6**: This bit must be cleared to zero

Bit 5 **M316CTF**: Touch Key Module 3 16-bit counter interrupt request flag
0: no request
1: interrupt request

Bit 4 **D4**: This bit must be cleared to zero

Bit 3 **M416CTE**: Touch Key Module 4 16-bit counter interrupt control
0: disable
1: enable

Bit 2 **D2**: This bit must be cleared to zero

Bit 1 **M316CTE**: Touch Key Module 3 16-bit counter interrupt control
0: disable
1: enable

Bit 0 **D0**: This bit must be cleared to zero

MFI5 Register – BS85C20 only

Bit	7	6	5	4	3	2	1	0
Name	—	—	T2AF	T2PF	—	—	T2AE	T2PE
R/W	—	—	R/W	R/W	—	—	R/W	R/W

POR	—	—	0	0	—	—	0	0

Bit 7~6 unimplemented, read as "0"

Bit 5 **T2AF**: TM2 Comparator A match interrupt request flag
0: no request
1: interrupt request

Bit 4 **T2PF**: TM2 Comparator P match interrupt request flag
0: no request
1: interrupt request

Bit 3~2 unimplemented, read as "0"

Bit 1 **T2AE**: TM2 Comparator A match interrupt control
0: disable
1: enable

Bit 0 **T2PE**: TM2 Comparator P match interrupt control
0: disable
1: enable

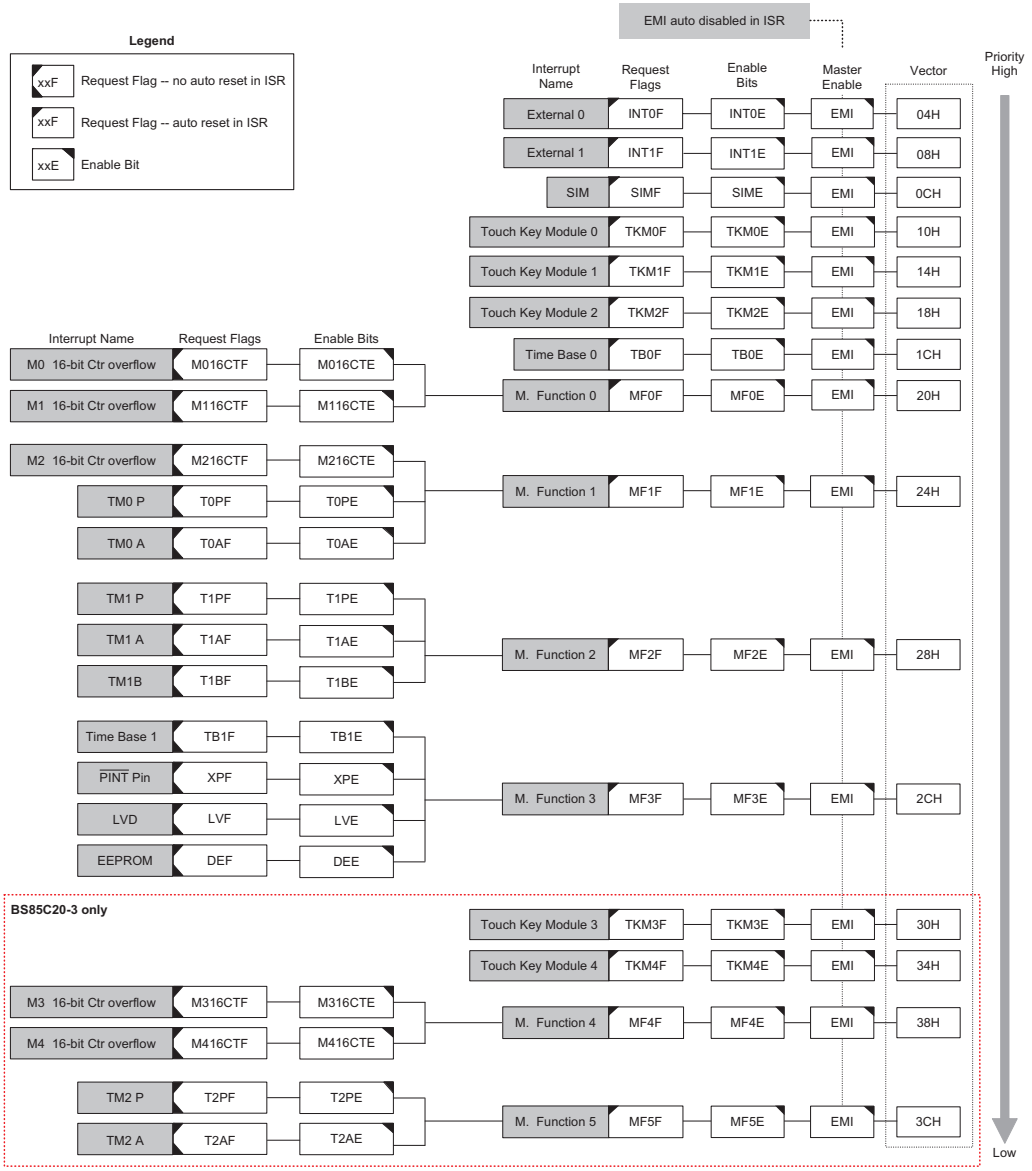
Interrupt Operation

When the conditions for an interrupt event occur, such as a Touch Key Counter overflow, Timer Module overflow, etc. 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 instruction which will jump to another section of program which is known as the interrupt service routine. Here is located the code to control the appropriate interrupt. The interrupt service routine must be terminated with a RETI instruction, which retrieves the original Program Counter address from the stack and allows the microcontroller to continue with normal execution at the point where the interrupt occurred.

The various interrupt enable bits, together with their associated request flags, are shown in the accompanying diagrams with their order of priority. Some interrupt sources have their own individual vector while others share the same multi-function interrupt 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 enters the SLEEP or IDLE Mode.



Interrupt Structure

External Interrupt

The external interrupt is controlled by signal transitions on the INT0 and INT1 pins. An external interrupt request will take place when the external interrupt request flag, INT0F or INT1F, is set, which will occur when a transition, whose type is chosen by the edge select bits, appears on the external interrupt pin. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and respective external interrupt enable bit, INT0E or INT1E, 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 pin is pin-shared with I/O pin, it can only be configured as external interrupt pin if the external interrupt enable bit in the corresponding interrupt register has been set. The pin must also be setup 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 pin, a subroutine call to the external interrupt vector, will take place. When the interrupt is serviced, the external interrupt request flag, INTF, will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts. Note that any pull-high resistor selections 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.

Multi-function Interrupt

Within these devices there are four or six Multi-function interrupts. Unlike the other independent interrupts, these interrupts have no independent source, but rather are formed from the Touch Key Module, Timer Module, Low Voltage Detector, EEPROM, External Peripheral and Time Base interrupt sources.

A Multi-function interrupt request will take place when any of the Multi-function interrupt request flags, MFnF are set. The Multi-function interrupt flags will be set when any of their included functions generate an interrupt request flag. To allow the program to branch to its respective interrupt vector address, when the Multi-function interrupt is enabled and the stack is not full, and either one of the interrupts contained within each of Multi-function interrupt occurs, a subroutine call to one of the Multi-function interrupt vectors will take place. When the interrupt is serviced, the related

Multi-Function request flag, will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts.

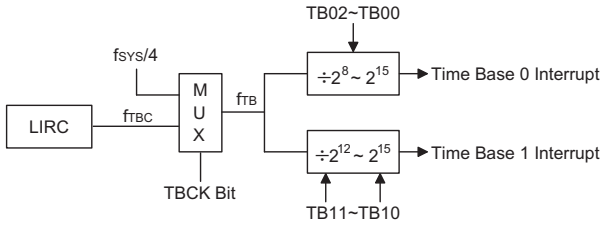
However, it must be noted that, although the Multi-function Interrupt flags will be automatically reset when the interrupt is serviced, the request flags from the original source of the Multi-function interrupts, namely the Touch Key module timer interrupts, will not be automatically reset and must be manually reset by the application program.

Time Base Interrupts

The function of the Time Base Interrupts is to provide regular time signal in the form of an internal interrupt. They are controlled by the overflow signals from their respective timer functions. When these happens their respective interrupt request flags, TB0F or TB1F will be set. To allow the program to branch to their respective interrupt vector addresses, the global interrupt enable bit, EMI and Time Base enable bits, TB0E or TB1E, must first be set.

When the interrupt is enabled, the stack is not full and the Time Base overflows, a subroutine call to their respective vector locations will take place. When the interrupt is serviced, the respective interrupt request flag, TB0F or TB1F, will be automatically reset and the EMI bit will be cleared to disable other interrupts.

The purpose of the Time Base Interrupt is to provide an interrupt signal at fixed time periods. Their clock sources originate from the internal clock source f_{TB}. This f_{TB} input clock passes through a divider, the division ratio of which is selected by programming the appropriate bits in the TBC register to obtain longer interrupt periods whose value ranges. The clock source that generates f_{TB}, which in turn controls the Time Base interrupt period, can originate from several different sources, as shown in the System Operating Mode section.



Time Base Structure

TBC Register

Bit	7	6	5	4	3	2	1	0
Name	TBON	TBCK	TB11	TB10	D3	TB02	TB01	TB00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	1	1	0	1	1	1

Bit 7 **TBON**: TB0 and TB1 Control
0: disable
1: enable

Bit 6 **TBCK**: Select f_{TB} Clock
0: f_{TBC}
1: $f_{SYS}/4$

Bit 5~4 **TB11~TB10**: Select Time Base 1 Time-out Period
0: $4096/f_{TB}$
1: $8192/f_{TB}$
2: $16384/f_{TB}$
3: $32768/f_{TB}$

Bit 3 Undefined bit
This bit can be read or written by user software program.

Bit 2~0 **TB02~TB00**: Select Time Base 0 Time-out Period
0: $256/f_{TB}$
1: $512/f_{TB}$
2: $1024/f_{TB}$
3: $2048/f_{TB}$
4: $4096/f_{TB}$
5: $8192/f_{TB}$
6: $16384/f_{TB}$
7: $32768/f_{TB}$

External Peripheral Interrupt

The External Peripheral Interrupt operates in a similar way to the external interrupt and is contained within the Multi-function Interrupt. A Peripheral Interrupt request will take place when the External Peripheral Interrupt request flag, XPF, is set, which occurs when a negative edge transition appears on the $\overline{\text{PINT}}$ pin. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, external peripheral interrupt enable bit, XPE, and associated Multi-function interrupt enable bit, must first be set. When the interrupt is enabled, the stack is not full and a negative transition appears on the External Peripheral Interrupt pin, a subroutine call to the respective Multi-function Interrupt, will take place. When the External Peripheral Interrupt is serviced, the EMI bit will be automatically cleared to disable other interrupts, however only the Multi-function interrupt request flag will be also automatically cleared.

As the XPF flag will not be automatically cleared, it has to be cleared by the application program. The external peripheral interrupt pin is pin-shared with several other pins with different functions. It must therefore be properly configured to enable it to operate as an External Peripheral Interrupt pin.

LVD Interrupt

The Low Voltage Detector Interrupt is contained within the Multi-function 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, Low Voltage Interrupt enable bit, LVE, and associated Multi-function interrupt enable bit, 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 Multi-function Interrupt vector, will take place. When the Low Voltage Interrupt is serviced, the EMI bit will be automatically cleared to disable other interrupts, however only the Multi-function interrupt request flag will be also automatically cleared. As the LVF flag will not be automatically cleared, it has to be cleared by the application program.

TM Interrupts

The Compact and Standard Type TMs have two interrupts each, while the Enhanced Type TM has three interrupts. All of the TM interrupts are contained within the Multi-function Interrupts. For each of the Compact and Standard Type TMs there are two interrupt request flags TnPF and TnAF and two enable bits TnPE and TnAE. For the Enhanced Type TM there are three interrupt request flags TnPF, TnAF and TnBF and three enable bits TnPE, TnAE and TnBE. 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, A or B match situation happens.

To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, respective TM Interrupt enable bit, and relevant Multi-function Interrupt enable bit, MFnE, 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 Multi-function Interrupt vector locations, will take place. When the TM interrupt is serviced, the EMI bit will be automatically cleared to disable other interrupts, however only the related MFnF flag will be automatically cleared. As the TM interrupt request flags will not be automatically cleared, they have to be cleared by the application program.

EEPROM Interrupt

The EEPROM Interrupt, is contained within the Multi-function Interrupt. An EEPROM Interrupt request will take place when the EEPROM Interrupt request flag, DEF, is set, which occurs when an EEPROM Write cycle ends. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, EEPROM Interrupt enable bit, DEE, and associated Multi-function interrupt enable bit, must first be set. When the interrupt is enabled, the stack is not full and an EEPROM Write cycle ends, a subroutine call to the respective Multi-function Interrupt vector, will take place. When the EEPROM Interrupt is serviced, the EMI bit will be automatically cleared to disable other interrupts, however only the Multi-function interrupt request flag will be also automatically cleared. As the DEF flag will not be automatically cleared, it has to be cleared by the application program.

Touch Key Interrupts

For a Touch Key interrupt to occur, the global interrupt enable bit, EMI, and the corresponding Touch Key interrupt enable TKMnE must be first set. An actual Touch Key interrupt will take place when the Touch Key request flag. TKMnF, is set, a situation that will occur when the 13-bit time slot counter in the relevant Touch Key module overflows. When the interrupt is enabled, the stack is not full and the Touch Key time slot counter overflow occurs, a subroutine call to the relevant Touch Key interrupt vector, will take place. When the interrupt is serviced, the Touch Key interrupt request flag, TKMnF, will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts.

SIM Interrupt

A SIM Interrupt request will take place when the SIM Interrupt request flag, SIMF, is set, which occurs when a byte of data has been received or transmitted by the SIM interface. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and the Serial Interface Interrupt enable bit, SIME, must first be set. When the interrupt is enabled, the stack is not full and a byte of data has been transmitted or received by the SIM interface, a subroutine call to the respective interrupt vector, will take place. When the Serial Interface Interrupt is serviced, the SIM interrupt request flag, SIF, will be automatically cleared and the EMI bit will 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 pins, a low power supply voltage or comparator input change 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.

Where a certain interrupt is contained within a Multi-function interrupt, then when the interrupt service routine is executed, as only the Multi-function interrupt request flags, MFnF, will be automatically cleared, the individual request flag for the function needs to be 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 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 a 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.

Low Voltage Detector – LVD

Each device has a Low Voltage Detector function, also known as LVD. This enabled the device to monitor the power supply voltage, V_{DD}, and provide 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 detemined. 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	—	VLVD2	VLVD1	VLVD0
R/W	—	—	R	R/W	—	R/W	R/W	R/W
POR	—	—	0	0	—	0	0	0

Bit 7~6 unimplemented, read as "0"

Bit 5 **LVDO**: LVD Output Flag
0: No Low Voltage Detect
1: Low Voltage Detect

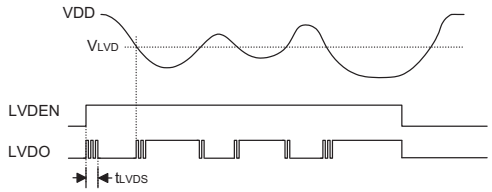
Bit **LVDEN**: Low Voltage Detector Control
0: Disable
1: Enable

Bit 3 unimplemented, read as "0"

Bit 2~0 **VLVD2 ~ VLVD0**: Select LVD Voltage
000: 2.0V
001: 2.2V
010: 2.4V
011: 2.7V
100: 3.0V
101: 3.3V
110: 3.6V
111: 4.2V

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 4.2V. 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. The Low Voltage Detector function is supplied by a reference voltage which will be automatically enabled. When the device is powered down the low voltage detector will remain active if the LVDEN bit is 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 which is contained within one of the Multi-function interrupts, 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. When the device is powered down the Low Voltage Detector will remain active if the LVDEN bit is high. 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 SLEEP or 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 SLEEP or IDLE Mode.

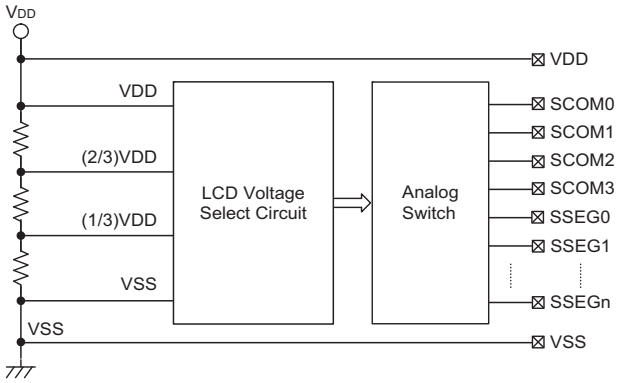
LCD Driver – SCOM and SSEG Function

The devices can drive LCD panels by simulating LCD signals on their I/O pins using the application program. Both Command and Segment signals can be emulated in this way.

LCD Operation

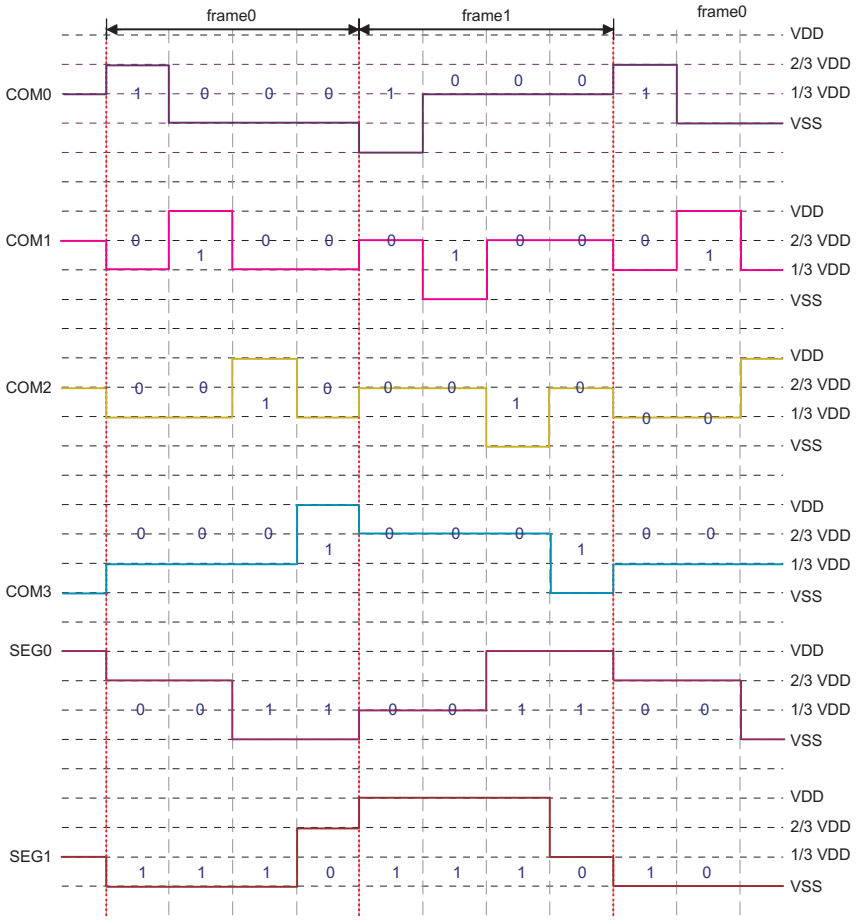
The LCD driving Common pins, SCOM0~SCOM3, and Segment pins, SSEG0~SSEGn, are pin shared with other I/O pins. These LCD driving pins are configured using a series of LCD control registers which in addition to controlling the overall on/off function also controls the bias voltage setup function. This enables the LCD COM and SEG driver to generate the necessary VSS, (1/3)VDD, (2/3)VDD voltage and VDD levels for full LCD 1/3 bias operation.

The SLCDEN bit in the LCD control register is the overall master control for the LCD driver, and this bit is used in conjunction with the COMnEN and SEGnEN bits to select which I/O Port pins are used for LCD driving. Note that the Port Control register does not need to first setup the pins as outputs to enable the LCD driver operation.



LCD Driver Structure

The accompanying waveform diagram shows a typical 1/3 Bias LCD waveform generated using the application program. Note that the depiction of a "1" in the diagram illustrates an illuminated LCD pixel. The COM signal polarity generated on pins SCOM0~SCOM3, whether 0 or 1, are generated using the corresponding I/O data registers, which are bits PB0~PB3 in the PB register.



Note: The logical values shown in the diagram are the PB I/O register values, PB0~PB3.

1/3 Bias LCD Waveform

A cyclic LCD waveform includes two frames, known as Frame 0 and Frame 1 for which the following offers a functional explanation.

In Frame 0

To select Frame 0 clear the FRAME bit to 0.

In frame 0, the COM signal output can have a value of VDD, or have a Vbias value of 1/3 VDD. The SEG signal can have a value of VSS, or have a Vbias value of 2/3 VDD.

In Frame 1

In frame 1, the COM signal output can have a value of VSS, have a Vbias value of 2/3 VDD. The SEG signal can have a value of VDD have a Vbias value of 1/3 VDD.

The COM0~COMn waveform is controlled by the application program using the FRAME bit, and the corresponding I/O data register for the respective COM pin to determine whether the COM0~COMn output has a value of either VDD, VSS or Vbias. The SEG0~SEGm waveform is controlled in a similar way using the FRAME bit and the corresponding I/O data register for the respective SEG pin to determine whether the SEG0~SEGn output has a value of either VDD, VSS or Vbias.

LCD Bias Control

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

LCD Driver Registers

SLCDC0 Register

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

Bit 7 **FRAME**: Present Frame output select -- Frame 0 or Frame 1
0: Frame 0
1: Frame 1

Bit 6~5 **ISEL1**, **ISEL0**: SCOM and SSEG operating current selection -- V_{DD}=5V
0: 25μA
1: 50μA
2: 100μA
3: 200μA

Bit 4 **SLCDEN**: SCOM and SSEG module on/off control
0: disable
1: enable
The SCOMn and SSEGm lines can be enabled using COMnEN and SEGmEN if SLCDEN=1. When SLCDEN=0, then the SCOMn and SSEGm outputs will be fixed at a VDD level.

Bit 3 **COM3EN**: SCOM3 or other function selection
0: Other function
1: SCOM3

Bit 2 **COM2EN**: SCOM2 or other function selection
0: Other function
1: SCOM2

Bit 1 **COM1EN**: SCOM1 or other function selection
0: Other function
1: SCOM1

Bit 0 **COM0EN**: SCOM0 or other function selection
0: Other function
1: SCOM0

SLCDC1 Register

Bit	7	6	5	4	3	2	1	0
Name	SEG7EN	SEG6EN	SEG5EN	SEG4EN	SEG3EN	SEG2EN	SEG1EN	SEG0EN
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 **SEG7EN~SEG0EN**: SSEG7~SSEG0 or other function selection
0: Other function
1: SSEG7~SSEG0

SLCDC2 Register – BS85B12-3

Bit	7	6	5	4	3	2	1	0
Name	—	—	SEG13EN	SEG12EN	SEG11EN	SEG10EN	SEG9EN	SEG8EN
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 5~0 **SEG13EN~SEG8EN**: SSEG13~SSEG8 or other function selection
0: Other function
1: SSEG13~SSEG8

SLCDC2 Register – BS85C20-3

Bit	7	6	5	4	3	2	1	0
Name	SEG15EN	SEG14EN	SEG13EN	SEG12EN	SEG11EN	SEG10EN	SEG9EN	SEG8EN
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 unimplemented, read as "0"
Bit 5~0 **SEG15EN~SEG8EN**: SSEG15~SSEG8 or other function selection
0: Other function
1: SSEG15~SSEG8

SLCDC3 Register – BS85C20-3

Bit	7	6	5	4	3	2	1	0
Name	TCK2PS	—	SEG21EN	SEG20EN	SEG19EN	SEG18EN	SEG17EN	SEG16EN
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 **TCK2PS**: TCK2 Pin Remapping Control
Described elsewhere
Bit 6 unimplemented, read as "0"
Bit 5~0 **SEG21EN~SEG16EN**: SSEG21~SSEG16 or other function selection
0: Other function
1: SSEG21~SSEG16

LED Driver

The devices contain an LED driver function offering high current output drive capability which can be used to drive external LEDs.

LED Driver Operation

Depending upon which device is chosen various I/O pins have a capability of providing LED high current drive outputs.

Device	LED Drive Pins
BS85B12-3	PA0~PA7 (high source current) PB0~PB5 (high sink current)
BS85C20-3	PA0~PA7 (high source current) PB0~PB7 (high sink current) PE0~PE5 (high source current)

Whether a normal current sink capability or high current sink capability is used, the selection is made using the SLEDCn registers.

LED Driver Registers

SLEDC0 Register – BS85B12-3

Bit	7	6	5	4	3	2	1	0
Name	—	—	D5	D4	D3	D2	D1	D0
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 5~0 **D5~D0**: PB5~PB0 I/O output sink current select
0: Normal output sink current
1: ×2 output sink current

SLEDC0 Register – BS85C20-3

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**: PB7~PB0 I/O output sink current select
0: Normal output sink current
1: ×2 output sink current

SLEDC1 Register – All Devices

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**: PA7~PA0 I/O output source current select
0: Normal output source current
1: ×2 output source current

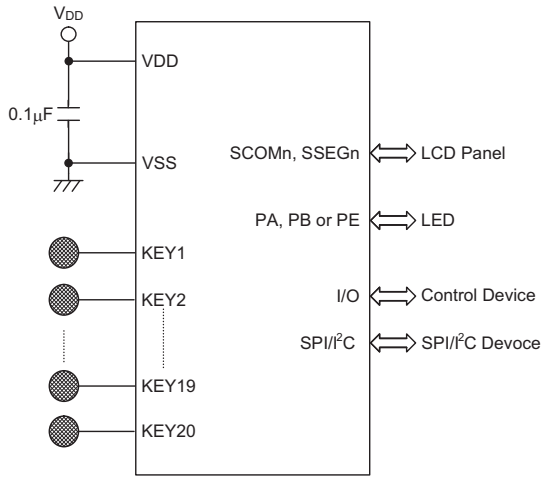
SLEDC2 Register – BS85C20-3

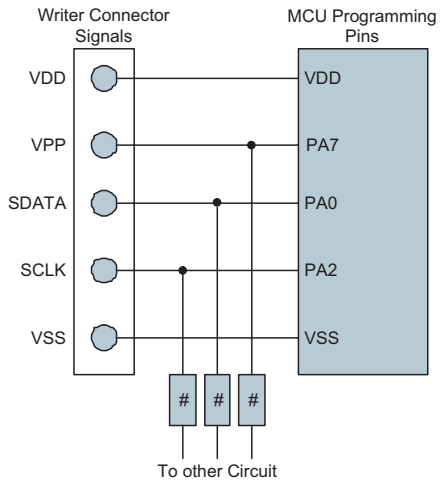
Bit	7	6	5	4	3	2	1	0
Name	—	—	D5	D4	D3	D2	D1	D0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

Bit 7~6 unimplemented, read as "0"

Bit 5~0 **D5~D0**: PE5~PE0 I/O output source current select
0: Normal output source current
1: ×2 output source current

Application Circuits





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

Instruction Set

Introduction

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

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

Instruction Timing

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

Moving and Transferring Data

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

Arithmetic Operations

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

Logical and Rotate Operations

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

Branches and Control Transfer

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

Bit Operations

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

Table Read Operations

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

Other Operations

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

Instruction Set Summary

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

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

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

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

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

Instruction Definition

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

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

Operation: ACC ← ACC + [m] + C

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

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

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

Operation: [m] ← ACC + [m] + C

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

ADD A,[m] Add Data Memory to ACC

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

Operation: ACC ← ACC + [m]

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

ADD A,x Add immediate data to ACC

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

Operation: ACC ← ACC + x

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

ADDM A,[m] Add ACC to Data Memory

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

Operation: [m] ← ACC + [m]

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

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

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

Operation: ACC ← ACC "AND" [m]

Affected flag(s): Z

AND A,x Logical AND immediate data to ACC

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

Operation: ACC ← ACC "AND" x

Affected flag(s): Z

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

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

Operation: [m] ← ACC "AND" [m]

Affected flag(s): Z

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

CPL [m] Complement Data Memory

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

Operation: $[m] \leftarrow \overline{[m]}$

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: $ACC \leftarrow \overline{[m]}$

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:
$[m] \leftarrow ACC + 00H$ or
$[m] \leftarrow ACC + 06H$ or
$[m] \leftarrow ACC + 60H$ or
$[m] \leftarrow ACC + 66H$

Affected flag(s): C

DEC [m] Decrement Data Memory

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

Operation: $[m] \leftarrow [m] - 1$

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: $ACC \leftarrow [m] - 1$

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:
$TO \leftarrow 0$
$PDF \leftarrow 1$

Affected flag(s): TO, PDF

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

OR A,x	Logical OR immediate data to ACC
Description	Data in the Accumulator and the specified immediate data perform a bitwise logical OR operation. The result is stored in the Accumulator.
Operation	ACC ← ACC ″OR″ x
Affected flag(s)	Z
ORM A,[m]	Logical OR ACC to Data Memory
Description	Data in the specified Data Memory and the Accumulator perform a bitwise logical OR operation. The result is stored in the Data Memory.
Operation	[m] ← ACC ″OR″ [m]
Affected flag(s)	Z
RET	Return from subroutine
Description	The Program Counter is restored from the stack. Program execution continues at the restored address.
Operation	Program Counter ← Stack
Affected flag(s)	None
RET A,x	Return from subroutine and load immediate data to ACC
Description	The Program Counter is restored from the stack and the Accumulator loaded with the specified immediate data. Program execution continues at the restored address.
Operation	Program Counter ← Stack ACC ← x
Affected flag(s)	None
RETI	Return from interrupt
Description	The Program Counter is restored from the stack and the interrupts are re-enabled by setting the EMI bit. EMI is the master interrupt global enable bit. If an interrupt was pending when the RETI instruction is executed, the pending Interrupt routine will be processed before returning to the main program.
Operation	Program Counter ← Stack EMI ← 1
Affected flag(s)	None
RL [m]	Rotate Data Memory left
Description	The contents of the specified Data Memory are rotated left by 1 bit with bit 7 rotated into bit 0.
Operation	[m].(i+1) ← [m].i; (i = 0~6) [m].0 ← [m].7
Affected flag(s)	None
RLA [m]	Rotate Data Memory left with result in ACC
Description	The contents of the specified Data Memory are rotated left by 1 bit with bit 7 rotated into bit 0. The rotated result is stored in the Accumulator and the contents of the Data Memory remain unchanged.
Operation	ACC.(i+1) ← [m].i; (i = 0~6) ACC.0 ← [m].7
Affected flag(s)	None

RLC [m] Rotate Data Memory left through Carry

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

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

Affected flag(s): C

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

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

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

Affected flag(s): C

RR [m] Rotate Data Memory right

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

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

Affected flag(s): None

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

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

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

Affected flag(s): None

RRC [m] Rotate Data Memory right through Carry

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

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

Affected flag(s): C

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

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

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

Affected flag(s): C

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

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

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

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

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

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

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

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

SDZ [m] Skip if decrement Data Memory is 0

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

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

Affected flag(s) None

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

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

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

Affected flag(s) None

SET [m] Set Data Memory

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

Operation $[m] \leftarrow FFH$

Affected flag(s) None

SET [m].i Set bit of Data Memory

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

Operation $[m].i \leftarrow 1$

Affected flag(s) None

SIZ [m] Skip if increment Data Memory is 0

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

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

Affected flag(s) None

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

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

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

Affected flag(s) None

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

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

Operation Skip if [m].i ≠ 0

Affected flag(s) None

SUB A,[m] Subtract Data Memory from ACC

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

Operation ACC ← ACC – [m]

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

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

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

Operation [m] ← ACC – [m]

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

SUB A,x Subtract immediate data from ACC

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

Operation ACC ← ACC – x

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

SWAP [m]	Swap nibbles of Data Memory
Description	The low-order and high-order nibbles of the specified Data Memory are interchanged.
Operation	[m].3~[m].0 ↔ [m].7 ~ [m].4
Affected flag(s)	None
SWAPA [m]	Swap nibbles of Data Memory with result in ACC
Description	The low-order and high-order nibbles of the specified Data Memory are interchanged. The result is stored in the Accumulator. The contents of the Data Memory remain unchanged.
Operation	ACC.3 ~ ACC.0 ← [m].7 ~ [m].4 ACC.7 ~ ACC.4 ← [m].3 ~ [m].0
Affected flag(s)	None
SZ [m]	Skip if Data Memory is 0
Description	If the contents of the specified Data Memory is 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.
Operation	Skip if [m] = 0
Affected flag(s)	None
SZA [m]	Skip if Data Memory is 0 with data movement to ACC
Description	The contents of the specified Data Memory are copied to the Accumulator. If the value is zero, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.
Operation	ACC ← [m] Skip if [m] = 0
Affected flag(s)	None
SZ [m].i	Skip if bit i of Data Memory is 0
Description	If bit i of the specified Data Memory is 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0, the program proceeds with the following instruction.
Operation	Skip if [m].i = 0
Affected flag(s)	None
TABRDC [m]	Read table (current page) to TBLH and Data Memory
Description	The low byte of the program code (current page) addressed by the table pointer (TBLP) is moved to the specified Data Memory and the high byte moved to TBLH.
Operation	[m] ← program code (low byte) TBLH ← program code (high byte)
Affected flag(s)	None

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

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

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

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

Package Information

24-pin SKDIP (300mil) Outline Dimensions

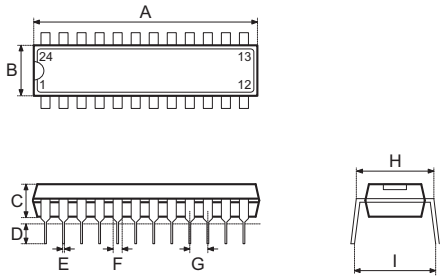


Fig1. Full Lead Packages

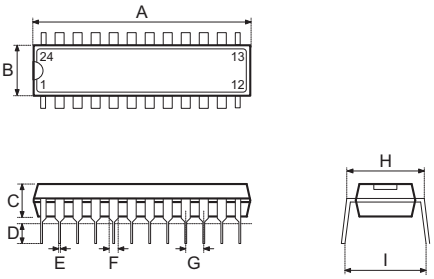


Fig2. 1/2 Lead Packages

MS-001d (see fig1)

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

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

MS-001d (see fig2)

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

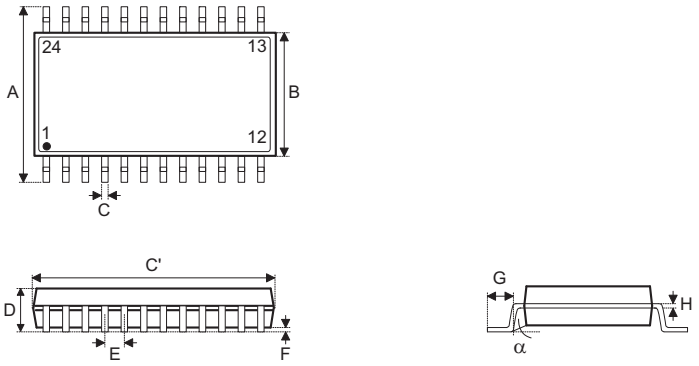
Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	29.46	—	30.35
B	6.10	—	7.11
C	2.92	—	4.95
D	2.92	—	3.81
E	0.36	—	0.56
F	1.14	—	1.78
G	—	2.54	—
H	7.62	—	8.26
I	—	10.92	—

MO-095a (see fig2)

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

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

24-pin SOP (300mil) Outline Dimensions

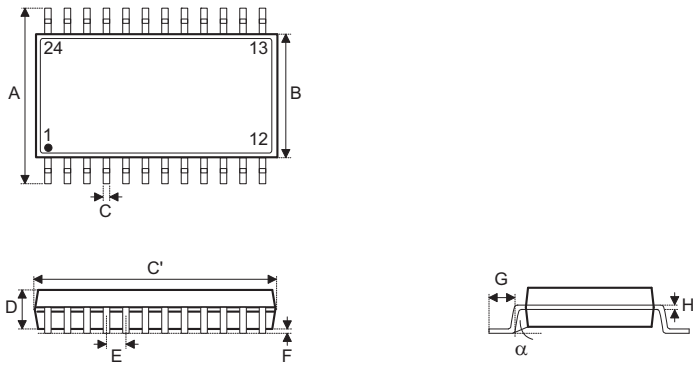


MS-013

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

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

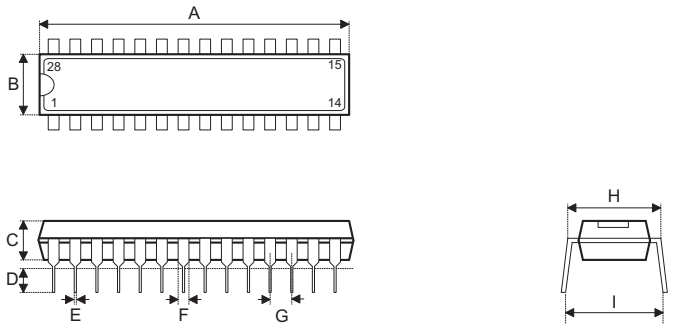
24-pin SSOP (150mil) Outline Dimensions



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

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

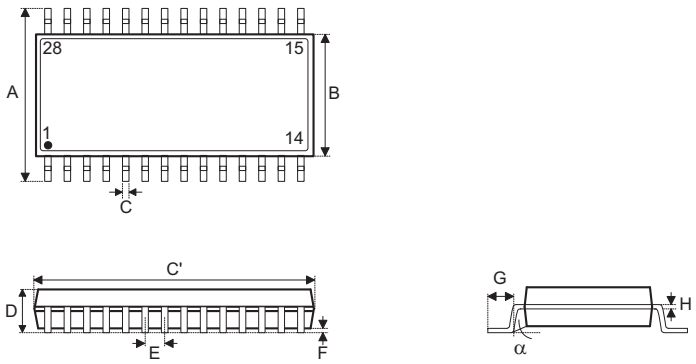
28-pin SKDIP (300mil) Outline Dimensions



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

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

28-pin SOP (300mil) Outline Dimensions

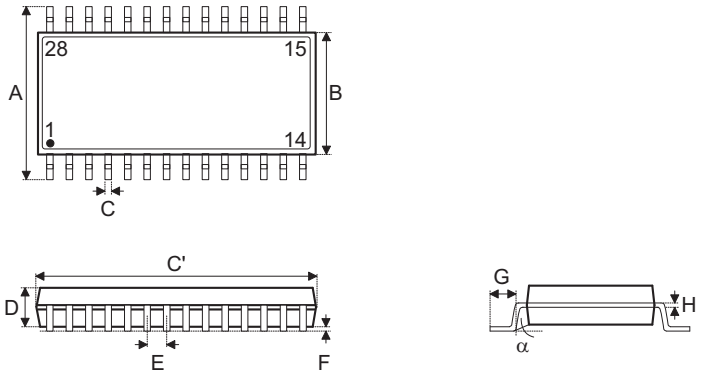


MS-013

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

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

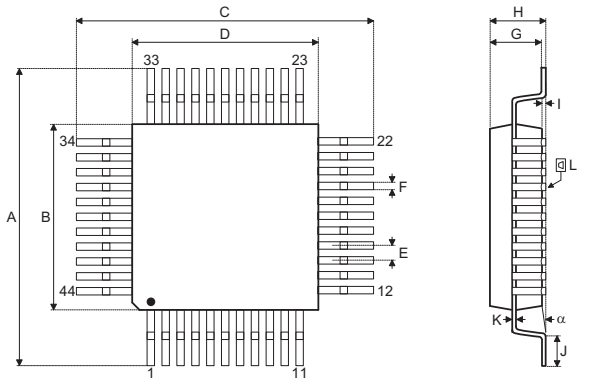
28-pin SSOP (150mil) Outline Dimensions



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

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

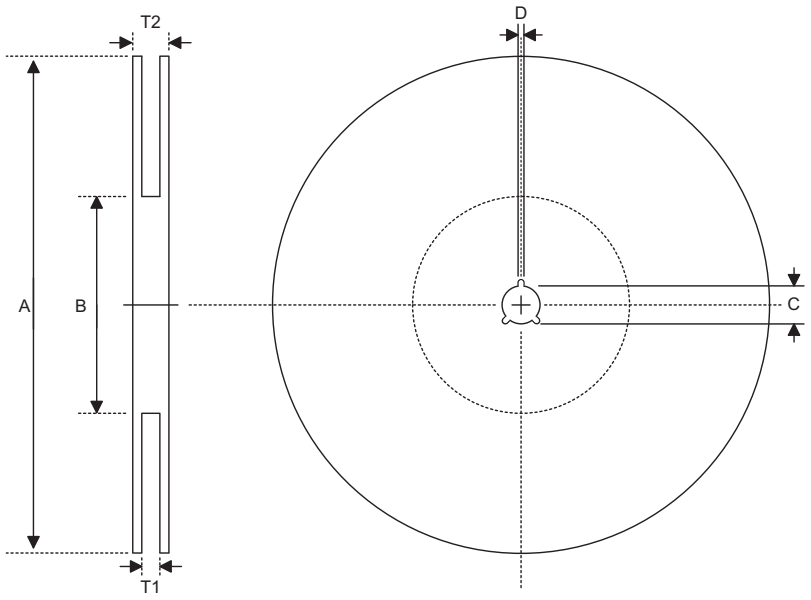
44-pin QFP (10mm×10mm) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.512	—	0.528
B	0.390	—	0.398
C	0.512	—	0.528
D	0.390	—	0.398
E	—	0.031	—
F	—	0.012	—
G	0.075	—	0.087
H	—	—	0.106
I	0.010	—	0.020
J	0.029	—	0.037
K	0.004	—	0.008
L	—	0.004	—
α	0°	—	7°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	13.00	—	13.40
B	9.90	—	10.10
C	13.00	—	13.40
D	9.90	—	10.10
E	—	0.80	—
F	—	0.30	—
G	1.90	—	2.20
H	—	—	2.70
I	0.25	—	0.50
J	0.73	—	0.93
K	0.10	—	0.20
L	—	0.10	—
α	0°	—	7°

Reel Dimensions



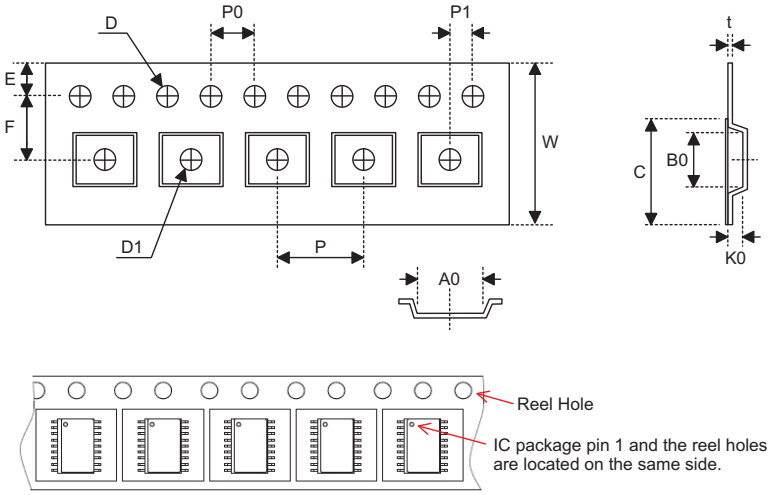
SOP 24W (300mil), SOP 28W (300mil)

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

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

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

Carrier Tape Dimensions



SOP 24W

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

SSOP 24S (150mil)

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

SOP 28W (300mil)

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

SSOP 28S (150mil)

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

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

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

Holtek Semiconductor Inc. (Shenzhen Sales Office)
5F, Unit A, Productivity Building, No.5 Gaoxin M 2nd Road, Nanshan District, Shenzhen, China 518057
Tel: 86-755-8616-9908, 86-755-8616-9308
Fax: 86-755-8616-9722

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