Flash MCU with EEPROM

HT66F20-1/HT66F30-1
HT68F20-1/HT68F30-1

Revision: V1.40 Date: November 23, 2016

www.holtek.com

Table of Contents

Features

CPU Features

- Operating Voltage:
  - f_{SYS}=8MHz: 2.2V~5.5V
  - f_{SYS}=12MHz: 2.7V~5.5V
  - f_{SYS}=20MHz: 4.5V~5.5V
- Up to 0.2μs instruction cycle with 20MHz system clock at V_{DD}=5V
- Power down and wake-up functions to reduce power consumption
- Five oscillators:
  - External Crystal - HXT
  - External 32.768kHz Crystal - LXT
  - External RC - ERC
  - Internal RC - HIRC
  - Internal 32kHz RC - LIRC
- Multi-mode operation: NORMAL, SLOW, IDLE and SLEEP
- Fully integrated internal 4MHz, 8MHz and 12MHz oscillator requires no external components
- All instructions executed in one or two instruction cycles
- Table read instructions
- 63 powerful instructions
- 4-level subroutine nesting
- Bit manipulation instruction

Peripheral Features

- Flash Program Memory: 1K×16 ~ 2K×16
- Data Memory: 64×8 ~ 96×8
- True EEPROM Memory: 32×8 ~ 64×8
- Watchdog Timer function
- Up to 22 bidirectional I/O lines
- Software controlled 4-SCOM lines LCD driver with 1/2 bias
- Dual pin-shared external interrupts
- Multiple Timer Module for time measure, input capture, compare match output, PWM output or single pulse output function
- Serial Interfaces Module with Dual SPI and I^2C interfaces
- Dual Comparator functions
- Dual Time-Base functions for generation of fixed time interrupt signals
- 8-channel 12-bit resolution A/D converter – HT66F30-1/HT66F20-1
- Low voltage reset function
- Low voltage detect function
- Wide range of available package types
- Flash program memory can be re-programmed up to 100,000 times
- Flash program memory data retention > 10 years
- True EEPROM data memory can be re-programmed up to 1,000,000 times
- True EEPROM data memory data retention > 10 years

General Description

The HT66Fx0-1 and HT68Fx0-1 series are Flash Memory type with 8-bit high performance RISC architecture microcontrollers, designed for a wide range of applications. Offering users the convenience of Flash Memory multi-programming features, these devices also include a wide range of functions and features. Other memory includes an area of RAM Data Memory as well as an area of true EEPROM memory for storage of non-volatile data such as serial numbers, calibration data etc.

Analog features include a multi-channel 12-bit A/D converter and dual comparator functions. Multiple and extremely flexible Timer Modules provide timing, pulse generation and PWM generation functions. Communication with the outside world is catered for by including fully integrated SPI or I²C interface functions, two popular interfaces which provide designers with a means of easy communication with external peripheral hardware. Protective features such as an internal Watchdog Timer, Low Voltage Reset and Low Voltage Detector coupled with excellent noise immunity and ESD protection ensure that reliable operation is maintained in hostile electrical environments.

A full choice of HXT, LXT, ERC, HIRC and LIRC oscillator functions are provided including a fully integrated system oscillator which requires no external components for its 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.

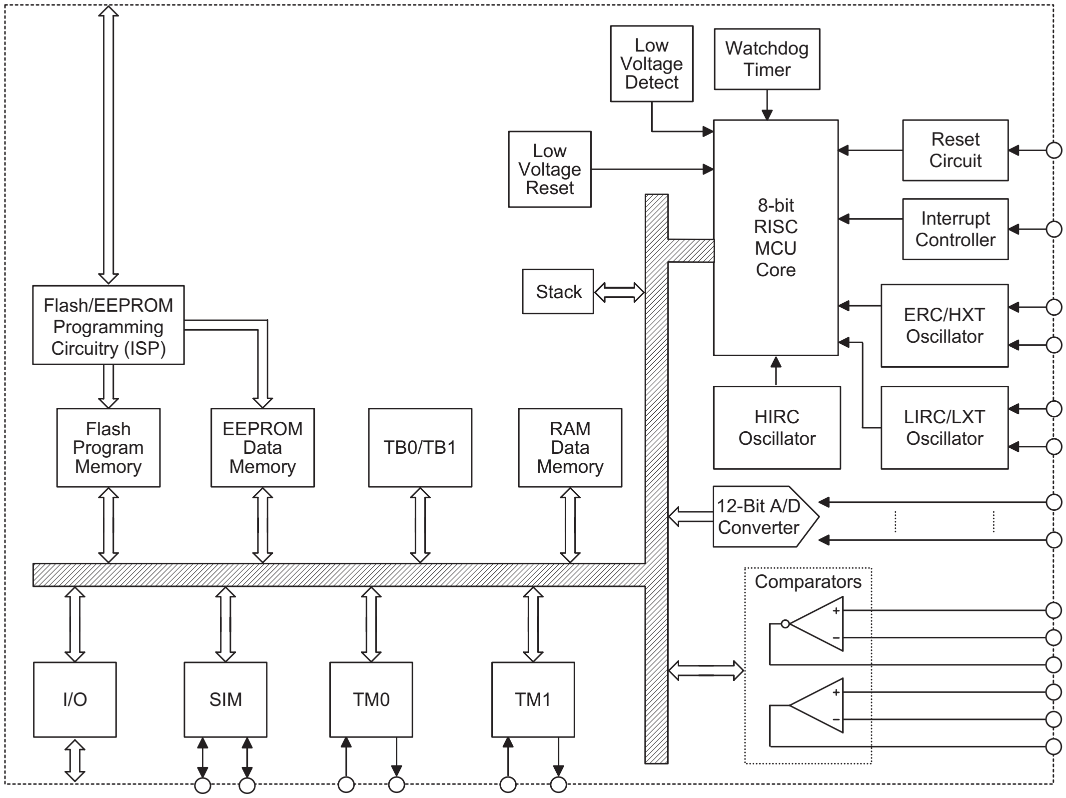
The inclusion of flexible I/O programming features, Time-Base functions along with many other features ensure that the devices will find excellent use in applications such as electronic metering, environmental monitoring, handheld instruments, household appliances, electronically controlled tools, motor driving in addition to many others.

Selection Table

Most features are common to all devices. The main features distinguishing them are Program Memory, Data Memory capacity, TM feature, A/D function, I/O count and package tyhpes. The following table summarises the main features of each device.

Part No.	Program Memory	Data Memory	Data EEPROM	I/O	Ext. Interrupt	A/D	Timer Module	SPI/ I²C	Time Base	Comp.	Stack	Package
HT68F20-1	1K×16	64×8	32×8	18	2	---	10-bit CTM×1 10-bit STM×1	√	2	2	4	16DIP/NSOP/SSOP 20DIP/SOP/SSOP
HT68F30-1	2K×16	96×8	64×8	22	2	---	10-bit CTM×1 10-bit ETM×1	√	2	2	4	16DIP/NSOP/SSOP 20DIP/SOP/SSOP 24SKDIP/SOP/ SSOP
HT66F20-1	1K×16	64×8	32×8	18	2	12-bitx8	10-bit CTM×1 10-bit STM×1	√	2	2	4	16DIP/NSOP/SSOP 20DIP/SOP/SSOP
HT66F30-1	2K×16	96×8	64×8	22	2	12-bitx8	10-bit CTM×1 10-bit ETM×1	√	2	2	4	16DIP/NSOP/SSOP 20DIP/SOP/SSOP 24SKDIP/SOP/ SSOP

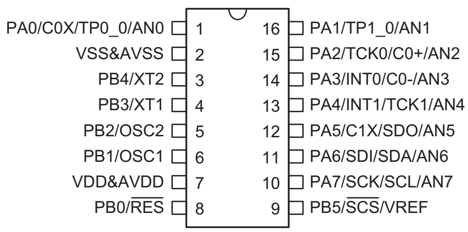
Block Diagram



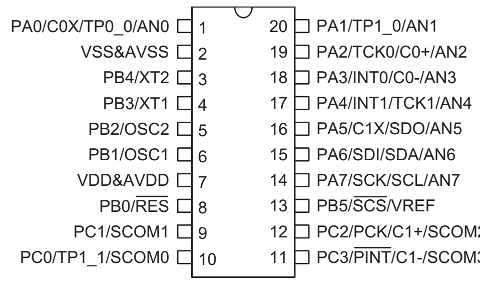
Note: Only the HT66F30-1 and HT66F20-1 devices have A/D function.

Pin Assignment

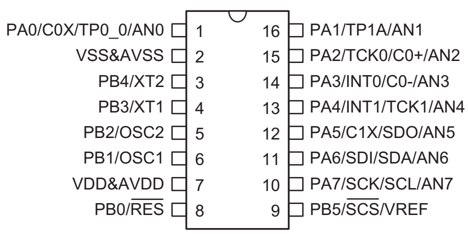
HT66F20-1 & HT66F30-1



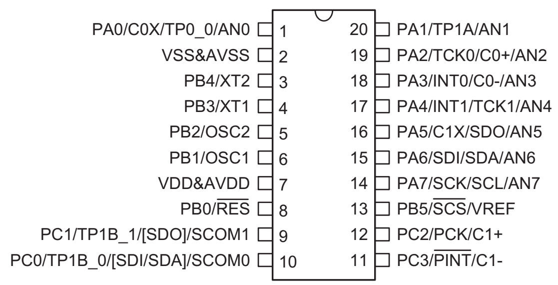
HT66F20-1
16 DIP-A/NSOP-A/SSOP-A



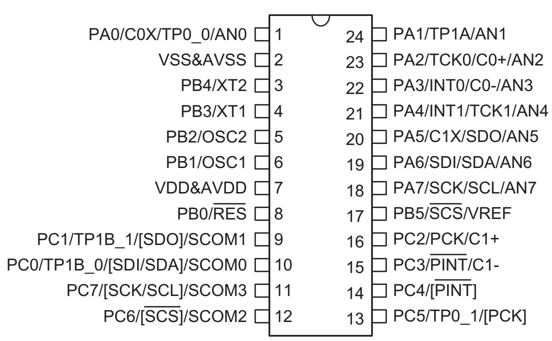
HT66F20-1
20 DIP-A/SOP-A/SSOP-A



HT66F30-1
16 DIP-A/NSOP-A/SSOP-A



HT66F30-1
20 DIP-A/SOP-A/SSOP-A

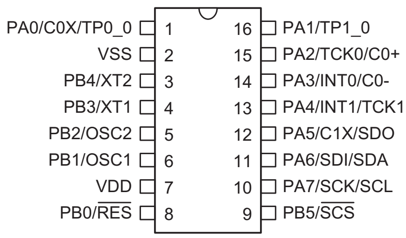


HT66F30-1
24 SKDIP-A/SOP-A/SSOP-A

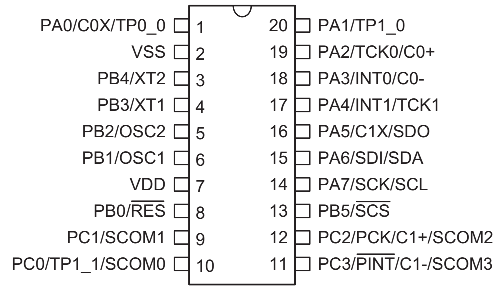
Note: 1. Bracketed pin names indicate non-default pinout remapping locations.

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.

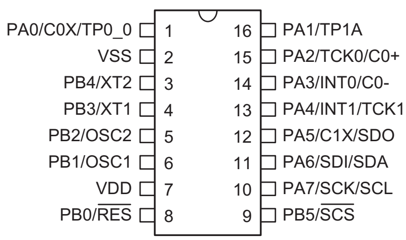
HT68F20-1 & HT68F30-1



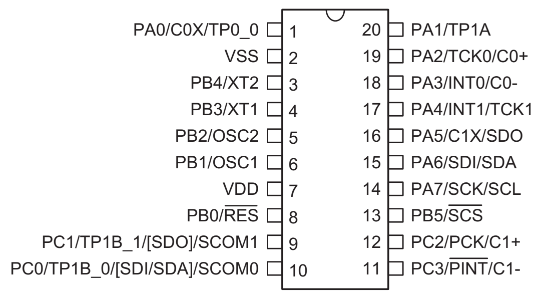
HT68F20-1
16 DIP-A/NSOP-A/SSOP-A



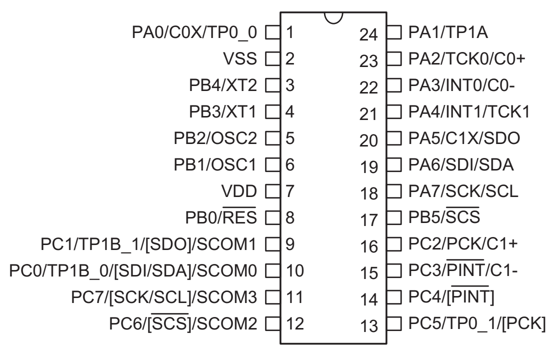
HT68F20-1
20 DIP-A/SOP-A/SSOP-A



HT68F30-1
16 DIP-A/NSOP-A/SSOP-A



HT68F30-1
20 DIP-A/SOP-A/SSOP-A



HT68F30-1
24 SKDIP-A/SOP-A/SSOP-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.

HT66F20-1

Pin Name	Function	OP	I/T	O/T	Pin-Shared Mapping
PA0~PA7	Port A	PAWU PAPU	ST	CMOS	—
PB0~PB5	Port B	PBPU	ST	CMOS	—
PC0~PC3	Port C	PCPU	ST	CMOS	—
AN0~AN7	ADC input	ACERL	AN	—	PA0~PA7
VREF	ADC reference input	ADCR1	AN	—	PB5
C0-, C1-	Comparator 0, 1 input	CP0C CP1C	AN	—	PA3, PC3
C0+, C1+	Comparator 0, 1 input		AN	—	PA2, PC2
C0X, C1X	Comparator 0, 1 output		—	CMOS	PA0, PA5
TCK0, TCK1	TM0, TM1 input	—	ST	—	PA2, PA4
TP0_0	TM0 I/O	TMPC0	ST	CMOS	PA0
TP1_0, TP1_1	TM1 I/O	TMPC0	ST	CMOS	PA1, PC0
INT0, INT1	Ext. Interrupt 0, 1	—	ST	—	PA3, PA4
$\overline{\text{PINT}}$	Peripheral Interrupt	—	ST	—	PC3
PCK	Peripheral Clock output	—	—	CMOS	PC2
SDI	SPI Data input	—	ST	—	PA6
SDO	SPI Data output	—	—	CMOS	PA5
$\overline{\text{SCS}}$	SPI Slave Select	—	ST	CMOS	PB5
SCK	SPI Serial Clock	—	ST	CMOS	PA7
SCL	I²C Clock	—	ST	NMOS	PA7
SDA	I²C Data	—	ST	NMOS	PA6
SCOM0~SCOM3	SCOM0~SCOM3	SCOMC	—	SCOM	PC0, PC1, PC2, PC3
OSC1	HXT/ERC pin	CO	HXT	—	PB1
OSC2	HXT pin	CO	—	HXT	PB2
XT1	LXT pin	CO	LXT	—	PB3
XT2	LXT pin	CO	—	LXT	PB4
$\overline{\text{RES}}$	Reset input	CO	ST	—	PB0
VDD	Power supply*	—	PWR	—	—
AVDD	ADC power supply*	—	PWR	—	—
VSS	Ground**	—	PWR	—	—
AVSS	ADC ground**	—	PWR	—	—

Note: I/T: Input type; O/T: Output type
OP: Optional by configuration option (CO) or register option
PWR: Power; CO: Configuration option; ST: Schmitt Trigger input
CMOS: CMOS output; NMOS: NMOS output
SCOM: Software controlled LCD COM; AN: Analog input pin
HXT: High frequency crystal oscillator LXT: Low frequency crystal oscillator
*: VDD is the device power supply while AVDD is the ADC power supply. The AVDD pin is bonded together internally with VDD.
**: VSS is the device ground pin while AVSS is the ADC ground pin. The AVSS pin is bonded together internally with VSS.

As the Pin Description Summary table applies to the package type with the most pins, not all of the above listed pins may be present on package types with smaller numbers of pins.

HT66F30-1

Pin Name	Function	OP	I/T	O/T	Pin-Shared Mapping
PA0~PA7	Port A	PAWU PAPU	ST	CMOS	—
PB0~PB5	Port B	PBPU	ST	CMOS	—
PC0~PC7	Port C	PCPU	ST	CMOS	—
AN0~AN7	ADC input	ACERL	AN	—	PA0~PA7
VREF	ADC reference input	ADCR1	AN	—	PB5
C0-, C1-	Comparator 0, 1 input	CP0C CP1C	AN	—	PA3, PC3
C0+, C1+	Comparator 0, 1 input		AN	—	PA2, PC2
C0X, C1X	Comparator 0, 1 output		—	CMOS	PA0, PA5
TCK0, TCK1	TM0, TM1 input	—	ST	—	PA2, PA4
TP0_0, TP0_1	TM0 I/O	TMPC0	ST	CMOS	PA0, PC5
TP1A	TM1 I/O	TMPC0	ST	CMOS	PA1
TP1B_0, TP1B_1	TM1 I/O	TMPC0	ST	CMOS	PC0, PC1
INT0, INT1	Ext. interrupt 0, 1	—	ST	—	PA3, PA4
$\overline{\text{PINT}}$	Peripheral interrupt	PRM0	ST	—	PC3 or PC4
PCK	Peripheral clock output	PRM0	—	CMOS	PC2 or PC5
SDI	SPI data input	PRM0	ST	—	PA6 or PC0
SDO	SPI data output	PRM0	—	CMOS	PA5 or PC1
$\overline{\text{SCS}}$	SPI slave select	PRM0	ST	CMOS	PB5 or PC6
SCK	SPI serial clock	PRM0	ST	CMOS	PA7 or PC7
SCL	I^2C clock	PRM0	ST	NMOS	PA7 or PC7
SDA	I^2C data	PRM0	ST	NMOS	PA6 or PC0
SCOM0~SCOM3	SCOM0~SCOM3	SCOMC	—	SCOM	PC0, PC1, PC6, PC7
OSC1	HXT/ERC pin	CO	HXT	—	PB1
OSC2	HXT pin	CO	—	HXT	PB2
XT1	LXT pin	CO	LXT	—	PB3
XT2	LXT pin	CO	—	LXT	PB4
$\overline{\text{RES}}$	Reset input	CO	ST	—	PB0
VDD	Power supply *	—	PWR	—	—
AVDD	ADC power supply *	—	PWR	—	—
VSS	Ground **	—	PWR	—	—
AVSS	ADC ground **	—	PWR	—	—

Note: I/T: Input type; O/T: Output type
OP: Optional by configuration option (CO) or register option
PWR: Power; CO: Configuration option; ST: Schmitt Trigger input
CMOS: CMOS output; NMOS: NMOS output
SCOM: Software controlled LCD COM; AN: Analog input pin
HXT: High frequency crystal oscillator; LXT: Low frequency crystal oscillator
*: VDD is the device power supply while AVDD is the ADC power supply. The AVDD pin is bonded together internally with VDD.
**: VSS is the device ground pin while AVSS is the ADC ground pin. The AVSS pin is bonded together internally with VSS.

As the Pin Description Summary table applies to the package type with the most pins, not all of the above listed pins may be present on package types with smaller numbers of pins.

HT68F20-1

Pin Name	Function	OP	I/T	O/T	Pin-Shared Mapping
PA0~PA7	Port A	PAWU PAPU	ST	CMOS	—
PB0~PB5	Port B	PBPU	ST	CMOS	—
PC0~PC3	Port C	PCPU	ST	CMOS	—
C0-, C1-	Comparator 0, 1 input	CP0C CP1C	AN	—	PA3, PC3
C0+, C1+	Comparator 0, 1 input		AN	—	PA2, PC2
C0X, C1X	Comparator 0, 1 output		—	CMOS	PA0, PA5
TCK0, TCK1	TM0, TM1 input	—	ST	—	PA2, PA4
TP0_0	TM0 I/O	TMPC0	ST	CMOS	PA0
TP1_0, TP1_1	TM1 I/O	TMPC0	ST	CMOS	PA1, PC0
INT0, INT1	Ext. Interrupt 0, 1	—	ST	—	PA3, PA4
$\overline{\text{PINT}}$	Peripheral Interrupt	—	ST	—	PC3
PCK	Peripheral Clock output	—	—	CMOS	PC2
SDI	SPI Data input	—	ST	—	PA6
SDO	SPI Data output	—	—	CMOS	PA5
$\overline{\text{SCS}}$	SPI Slave Select	—	ST	CMOS	PB5
SCK	SPI Serial Clock	—	ST	CMOS	PA7
SCL	I^{2}C Clock	—	ST	NMOS	PA7
SDA	I^{2}C Data	—	ST	NMOS	PA6
SCOM0~SCOM3	SCOM0~SCOM3	SCOMC	—	SCOM	PC0, PC1, PC2, PC3
OSC1	HXT/ERC pin	CO	HXT	—	PB1
OSC2	HXT pin	CO	—	HXT	PB2
XT1	LXT pin	CO	LXT	—	PB3
XT2	LXT pin	CO	—	LXT	PB4
$\overline{\text{RES}}$	Reset input	CO	ST	—	PB0
VDD	Power supply	—	PWR	—	—
VSS	Ground	—	PWR	—	—

Note: I/T: Input type
O/T: Output type
OP: Optional by configuration option (CO) or register option
PWR: Power
CO: Configuration option
ST: Schmitt Trigger input
CMOS: CMOS output
NMOS: NMOS output
SCOM: Software controlled LCD COM
AN: Analog input pin
HXT: High frequency crystal oscillator
LXT: Low frequency crystal oscillator

As the Pin Description Summary table applies to the package type with the most pins, not all of the above listed pins may be present on package types with smaller numbers of pins.

HT68F30-1

Pin Name	Function	OP	I/T	O/T	Pin-Shared Mapping
PA0~PA7	Port A	PAWU PAPU	ST	CMOS	—
PB0~PB5	Port B	PBPU	ST	CMOS	—
PC0~PC7	Port C	PCPU	ST	CMOS	—
C0-, C1-	Comparator 0, 1 input	CP0C CP1C	AN	—	PA3, PC3
C0+, C1+	Comparator 0, 1 input		AN	—	PA2, PC2
C0X, C1X	Comparator 0, 1 output		—	CMOS	PA0, PA5
TCK0, TCK1	TM0, TM1 input	—	ST	—	PA2, PA4
TP0_0, TP0_1	TM0 I/O	TMPC0	ST	CMOS	PA0, PC5
TP1A	TM1 I/O	TMPC0	ST	CMOS	PA1
TP1B_0, TP1B_1	TM1 I/O	TMPC0	ST	CMOS	PC0, PC1
INT0, INT1	Ext. interrupt 0, 1	—	ST	—	PA3, PA4
$\overline{\text{PINT}}$	Peripheral interrupt	PRM0	ST	—	PC3 or PC4
PCK	Peripheral clock output	PRM0	—	CMOS	PC2 or PC5
SDI	SPI data input	PRM0	ST	—	PA6 or PC0
SDO	SPI data output	PRM0	—	CMOS	PA5 or PC1
$\overline{\text{SCS}}$	SPI slave select	PRM0	ST	CMOS	PB5 or PC6
SCK	SPI serial clock	PRM0	ST	CMOS	PA7 or PC7
SCL	I^2C clock	PRM0	ST	NMOS	PA7 or PC7
SDA	I^2C data	PRM0	ST	NMOS	PA6 or PC0
SCOM0~SCOM3	SCOM0~SCOM3	SCOMC	—	SCOM	PC0, PC1, PC6, PC7
OSC1	HXT/ERC pin	CO	HXT	—	PB1
OSC2	HXT pin	CO	—	HXT	PB2
XT1	LXT pin	CO	LXT	—	PB3
XT2	LXT pin	CO	—	LXT	PB4
$\overline{\text{RES}}$	Reset input	CO	ST	—	PB0
VDD	Power supply	—	PWR	—	—
VSS	Ground	—	PWR	—	—

Note: I/T: Input type
O/T: Output type
OP: Optional by configuration option (CO) or register option
PWR: Power
CO: Configuration option
ST: Schmitt Trigger input
CMOS: CMOS output
NMOS: NMOS output
SCOM: Software controlled LCD COM
AN: Analog input pin
HXT: High frequency crystal oscillator
LXT: Low frequency crystal oscillator

As the Pin Description Summary table applies to the package type with the most pins, not all of the above listed pins may be present on package types with smaller numbers of pins.

Absolute Maximum Ratings

Supply Voltage .... V_{SS}−0.3V to V_{SS}+6.0V
Input Voltage .... V_{SS}−0.3V to V_{DD}+0.3V
Storage Temperature .... -50°C to 125°C
Operating Temperature .... -40°C to 85°C
I_{OH} Total .... -80mA
I_{OL} 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 these devices. Functional operation of these devices at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect devices reliability.

D.C. Characteristics

HT66F20-1/HT66F30-1

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage (HXT, ERC, HIRC)	—	f_{SYS}=8MHz	2.2	—	5.5	V
			f_{SYS}=12MHz	2.7	—	5.5	V
			f_{SYS}=20MHz	4.5	—	5.5	V
I_{DD1}	Operating Current, Normal Mode, f_{SYS}=f_H (HXT, ERC, HIRC)	3V	No load, f_{SYS}=f_H=4MHz, ADC off, WDT enable	—	0.7	1.1	mA
		5V		—	1.8	2.7	mA
		3V	No load, f_{SYS}=f_H=8MHz, ADC off, WDT enable	—	1.6	2.4	mA
		5V		—	3.3	5.0	mA
		3V	No load, f_{SYS}=f_H=12MHz, ADC off, WDT enable	—	2.2	3.3	mA
		5V		—	5.0	7.5	mA
I_{DD2}	Operating Current, Normal Mode, f_{SYS}=f_H (HXT)	5V	No load, f_{SYS}=f_H=20MHz, ADC off, WDT enable	—	6.0	9.0	mA
I_{DD3}	Operating Current, Slow Mode, f_{SYS}=f_L (LXT, LIRC)	3V	No load, f_{SYS}=f_L, ADC off, WDT enable	—	10	20	μA
		5V		—	30	50	μA
I_{IDLE0}	IDLE0 Mode Standby Current (LXT or LIRC on)	3V	No load, ADC off, WDT enable	—	1.5	3.0	μA
		5V		—	3.0	6.0	μA
I_{IDLE1}	IDLE1 Mode Standby Current (HXT, ERC, HIRC)	3V	No load, ADC off, WDT enable, f_{SYS}=12MHz on	—	0.55	0.83	mA
		5V		—	1.30	2.00	mA
I_{SLEEP0}	SLEEP0 Mode Standby Current (LXT and LIRC off)	3V	No load, ADC off, WDT disable	—	—	1	μA
		5V		—	—	2	μA
I_{SLEEP1}	SLEEP1 Mode Standby Current (LXT or LIRC on)	3V	No load, ADC off, WDT enable	—	1.5	3.0	μA
		5V		—	2.5	5.0	μA

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{IL1}	Input Low Voltage for I/O Ports or Input Pins except $\overline{RES}$ pin	—	—	0	—	$0.3V_{DD}$	V
V_{IH1}	Input High Voltage for I/O Ports or Input Pins except $\overline{RES}$ pin	—	—	$0.7V_{DD}$	—	V_{DD}	V
V_{IL2}	Input Low Voltage ($\overline{RES}$)	—	—	0	—	$0.4V_{DD}$	V
V_{IH2}	Input High Voltage ($\overline{RES}$)	—	—	$0.9V_{DD}$	—	V_{DD}	V
V_{LVR}	LVR Voltage Level	—	LVR Enable, 2.10V option	-5%	2.10	+5%	V
			LVR Enable, 2.55V option	-5%	2.55	+5%	V
			LVR Enable, 3.15V option	-5%	3.15	+5%	V
			LVR Enable, 4.20V option	-5%	4.20	+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}=4.4V	-5%	4.40	+5%	V
I_{LV}	Additional Power Consumption if LVR and LVD is Used	—	LVR enable, LVDEN=0	—	60	90	μA
			LVR disable, LVDEN=1	—	75	115	μA
			LVR enable, LVDEN=1	—	90	135	μA
V_{OL}	Output Low Voltage I/O Port	3V	I_{OL}=9mA	—	—	0.3	V
		5V	I_{OL}=20mA	—	—	0.5	V
V_{OH}	Output High Voltage I/O Port	3V	I_{OH}=-3.2mA	2.7	—	—	V
		5V	I_{OH}=-7.4mA	4.5	—	—	V
R_{PH}	Pull-high Resistance for I/O Ports	3V	—	20	60	100	kΩ
		5V		10	30	50	kΩ
I_{SCOM}	SCOM Operating Current	5V	SCOMC, ISEL[1:0]=00	17.5	25.0	32.5	μA
			SCOMC, ISEL[1:0]=01	35	50	65	μA
			SCOMC, ISEL[1:0]=10	70	100	130	μA
			SCOMC, ISEL[1:0]=11	140	200	260	μA
V_{SCOM}	V_{DD}/2 Voltage for LCD COM	5V	No load	0.475	0.500	0.525	V_{DD}
V_{125}	1.25V Reference with Buffer Voltage	—	—	-3%	1.25	+3%	V
I_{125}	Additional Power Consumption if 1.25V Reference with Buffer is used	—	—	—	200	300	μA

HT68F20-1/HT68F30-1

Ta=25˚C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage (HXT, ERC, HIRC)	—	f_{SYS}=8MHz	2.2	—	5.5	V
			f_{SYS}=12MHz	2.7	—	5.5	V
			f_{SYS}=20MHz	4.5	—	5.5	V
I_{DD1}	Operating Current, Normal Mode, f_{SYS}=f_H (HXT, ERC, HIRC)	3V	No load, f_{SYS}=f_H=4MHz, WDT enable	—	0.7	1.1	mA
		5V		—	1.8	2.7	mA
		3V	No load, f_{SYS}=f_H=8MHz, WDT enable	—	1.6	2.4	mA
		5V		—	3.3	5.0	mA
		3V	No load, f_{SYS}=f_H=12MHz, WDT enable	—	2.2	3.3	mA
		5V		—	5.0	7.5	mA
I_{DD2}	Operating Current, Normal Mode, f_{SYS}=f_H (HXT)	5V	No load, f_{SYS}=f_H=20MHz, WDT enable	—	6.0	9.0	mA
I_{DD3}	Operating Current, Slow Mode, f_{SYS}=f_L (LXT, LIRC)	3V	No load, f_{SYS}=f_L, WDT enable	—	10	20	μA
		5V		—	30	50	μA
I_{IDLE0}	IDLE0 Mode Standby Current (LXT or LIRC on)	3V	No load, WDT enable	—	1.5	3.0	mA
		5V		—	3.0	6.0	mA
I_{IDLE1}	IDLE1 Mode Standby Current (HXT, ERC, HIRC)	3V	No load, WDT enable, f_{SYS}=12MHz on	—	0.55	0.83	mA
		5V		—	1.30	2.00	mA
I_{SLEEP0}	SLEEP0 Mode Standby Current (LXT and LIRC off)	3V	No load, WDT disable	—	—	1	μA
		5V		—	—	2	μA
I_{SLEEP1}	SLEEP1 Mode Standby Current (LXT or LIRC on)	3V	No load, WDT enable	—	1.5	3.0	μA
		5V		—	2.5	5.0	μA
V_{IL1}	Input Low Voltage for I/O Ports or Input Pins except $\overline{RES}$ pin	—	—	0	—	$0.3V_{DD}$	V
V_{IH1}	Input High Voltage for I/O Ports or Input Pins except $\overline{RES}$ pin	—	—	$0.7V_{DD}$	—	V_{DD}	V
V_{IL2}	Input Low Voltage ($\overline{RES}$)	—	—	0	—	$0.4V_{DD}$	V
V_{IH2}	Input High Voltage ($\overline{RES}$)	—	—	$0.9V_{DD}$	—	V_{DD}	V
V_{LVR}	LVR Voltage Level	—	LVR Enable, 2.10V option	-5%	2.10	+5%	V
			LVR Enable, 2.55V option	-5%	2.55	+5%	V
			LVR Enable, 3.15V option	-5%	3.15	+5%	V
			LVR Enable, 4.20V option	-5%	4.20	+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}=4.4V	-5%	4.40	+5%	V

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{LV}	Additional Power Consumption if LVR and LVD is used	—	LVR enable, LVDEN=0	—	60	90	μA
			LVR disable, LVDEN=1	—	75	115	μA
			LVR enable, LVDEN=1	—	90	135	μA
V_{OL}	Output Low Voltage I/O Port	3V	I_{OL}=9mA	—	—	0.3	V
		5V	I_{OL}=20mA	—	—	0.5	V
V_{OH}	Output High Voltage I/O Port	3V	I_{OH}=-3.2mA	2.7	—	—	V
		5V	I_{OH}=-7.4mA	4.5	—	—	V
R_{PH}	Pull-high Resistance for I/O Ports	3V	—	20	60	100	kΩ
		5V		10	30	50	kΩ
I_{SCOM}	SCOM Operating Current	5V	SCOMC, ISEL[1:0]=00	17.5	25.0	32.5	μA
			SCOMC, ISEL[1:0]=01	35	50	65	μA
			SCOMC, ISEL[1:0]=10	70	100	130	μA
			SCOMC, ISEL[1:0]=11	140	200	260	μA
V_{SCOM}	V_{DD}/2 Voltage for LCD COM	5V	No load	0.475	0.500	0.525	V_{DD}

A.C. Characteristics

HT66F20-1/HT66F30-1

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{CPU}	Operating Clock	—	2.2V~5.5V	DC	—	8	MHz
			2.7V~5.5V	DC	—	12	MHz
			4.5V~5.5V	DC	—	20	MHz
f_{SYS}	System Clock (HXT)	—	2.2V~5.5V	0.4	—	8	MHz
			2.7V~5.5V	0.4	—	12	MHz
			4.5V~5.5V	0.4	—	20	MHz
f_{HIRC}	System Clock (HIRC)	3V/5V	Ta=25°C	-2%	4	+2%	MHz
		3V/5V	Ta=25°C	-2%	8	+2%	MHz
		5V	Ta=25°C	-2%	12	+2%	MHz
		3V/5V	Ta=0~70°C	-5%	4	+5%	MHz
		3V/5V	Ta=0~70°C	-4%	8	+4%	MHz
		5V	Ta=0~70°C	-5%	12	+3%	MHz
		2.2V~3.6V	Ta=0~70°C	-7%	4	+7%	MHz
		3.0V~5.5V	Ta=0~70°C	-5%	4	+9%	MHz
		2.2V~3.6V	Ta=0~70°C	-6%	8	+4%	MHz
		3.0V~5.5V	Ta=0~70°C	-4%	8	+9%	MHz
		3.0V~5.5V	Ta=0~70°C	-6%	12	+7%	MHz
		2.2V~3.6V	Ta=-40°C~85°C	-12%	4	+8%	MHz
		3.0V~5.5V	Ta=-40°C~85°C	-10%	4	+9%	MHz
		2.2V~3.6V	Ta=-40°C~85°C	-15%	8	+4%	MHz
		3.0V~5.5V	Ta=-40°C~85°C	-8%	8	+9%	MHz
		3.0V~5.5V	Ta=-40°C~85°C	-12%	12	+7%	MHz

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{ERC}	System Clock (ERC)	5V	Ta=25˚C, R=120kΩ*	-2%	8	+2%	MHz
		5V	Ta=0~70˚C, R=120kΩ*	-5%	8	+6%	MHz
		5V	Ta=-40˚C~85˚C, R=120kΩ*	-7%	8	+9%	MHz
		3.0V~5.5V	Ta=-40˚C~85˚C, R=120kΩ*	-9%	8	+10%	MHz
		2.2V~5.5V	Ta=-40˚C~85˚C, R=120kΩ*	-15%	8	+10%	MHz
f_{LXT}	System Clock (LXT)	—	—	—	32.768	—	kHz
f_{LIRC}	System Clock (LIRC)	5V	Ta=25˚C	-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_{RES}	External Reset Low Pulse Width	—	—	1	—	—	μs
t_{INT}	Interrupt Pulse Width	—	—	1	—	—	t_{SYS}
t_{LVR}	Low Voltage Width to Reset	—	—	120	240	480	μs
t_{LVD}	Low Voltage Width to Interrupt	—	—	20	45	90	μs
t_{LVDS}	LVDO stable time	—	—	15	—	—	μs
t_{BGS}	V_{BG} Turn on Stable Time	—	—	200	—	—	μs
t_{EERD}	EEPROM Read Time	—	—	—	45	90	μs
t_{EEWR}	EEPROM Write Time	—	—	—	2	4	ms
t_{SST}	System Start-up Timer Period (Wake-up from HALT)	—	f_{SYS}=HXT or LXT	—	1024	—	t_{SYS}
			f_{SYS}=ERC or HIRC	—	15~16	—	
			f_{SYS}=LIRC OSC	—	1~2	—	

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

HT68F20-1/HT68F30-1

Ta=25˚C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{CPU}	Operating Clock	—	2.2V~5.5V	DC	—	8	MHz
			2.7V~5.5V	DC	—	12	MHz
			4.5V~5.5V	DC	—	20	MHz
f_{SYS}	System Clock (HXT)	—	2.2V~5.5V	0.4	—	8	MHz
			2.7V~5.5V	0.4	—	12	MHz
			4.5V~5.5V	0.4	—	20	MHz
f_{HIRC}	System Clock (HIRC)	3V/5V	Ta=25˚C	-2%	4	+2%	MHz
		3V/5V	Ta=25˚C	-2%	8	+2%	MHz
		5V	Ta=25˚C	-2%	12	+2%	MHz
		3V/5V	Ta=0~70˚C	-5%	4	+5%	MHz
		3V/5V	Ta=0~70˚C	-4%	8	+4%	MHz
		5V	Ta=0~70˚C	-5%	12	+3%	MHz
		2.2V~3.6V	Ta=0~70˚C	-7%	4	+7%	MHz
		3.0V~5.5V	Ta=0~70˚C	-5%	4	+9%	MHz
		2.2V~3.6V	Ta=0~70˚C	-6%	8	+4%	MHz
		3.0V~5.5V	Ta=0~70˚C	-4%	8	+9%	MHz
		3.0V~5.5V	Ta=0~70˚C	-6%	12	+7%	MHz
		2.2V~3.6V	Ta=-40˚C~85˚C	-12%	4	+8%	MHz
		3.0V~5.5V	Ta=-40˚C~85˚C	-10%	4	+9%	MHz
		2.2V~3.6V	Ta=-40˚C~85˚C	-15%	8	+4%	MHz
		3.0V~5.5V	Ta=-40˚C~85˚C	-8%	8	+9%	MHz
		3.0V~5.5V	Ta=-40˚C~85˚C	-12%	12	+7%	MHz
f_{ERC}	System Clock (ERC)	5V	Ta=25˚C, R=120kΩ*	-2%	8	+2%	MHz
		5V	Ta=0~70˚C, R=120kΩ*	-5%	8	+6%	MHz
		5V	Ta=-40˚C~85˚C, R=120kΩ*	-7%	8	+9%	MHz
		3.0V~5.5V	Ta=-40˚C~85˚C, R=120kΩ*	-9%	8	+10%	MHz
		2.2V~5.5V	Ta=-40˚C~85˚C, R=120kΩ*	-15%	8	+10%	MHz
f_{LXT}	System Clock (LXT)	—	—	—	32.768	—	kHz
f_{LIRC}	System Clock (LIRC)	5V	Ta=25˚C	-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_{RES}	External Reset Low Pulse Width	—	—	1	—	—	μs
t_{INT}	Interrupt Pulse Width	—	—	1	—	—	t_{SYS}
t_{LVR}	Low Voltage Width to Reset	—	—	120	240	480	μs
t_{LVD}	Low Voltage Width to Interrupt	—	—	20	45	90	μs
t_{LVDS}	LVDO stable time	—	—	15	—	—	μs
t_{BGS}	VBG Turn on Stable Time	—	—	200	—	—	μs
t_{EERD}	EEPROM Read Time	—	—	—	45	90	μs
t_{EEWR}	EEPROM Write Time	—	—	—	2	4	ms
t_{SST}	System Start-up Timer Period (Wake-up from HALT)	—	f_{SYS}=HXT or LXT	—	1024	—	t_{SYS}
			f_{SYS}=ERC or HIRC	—	15~16	—	
			f_{SYS}=LIRC OSC	—	1~2	—	

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

2. * For f_{ERC}, as the resistor tolerance will influence the frequency a precision resistor is recommended.

3. To maintain the accuracy of the internal HIRC oscillator frequency, a 0.1μF decoupling capacitor should be connected between VDD and VSS and located as close to the device as possible.

A/D Converter Characteristics

HT66F20-1/HT66F30-1

Ta=25˚C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
AV_{DD}	A/D Converter Operating Voltage	—	—	2.7	—	5.5	V
V_{ADI}	A/D Converter Input Voltage	—	—	0	—	V_{REF}	V
V_{REF}	A/D Converter Reference Voltage	—	—	2	—	AV_{DD}	V
DNL	Differential non-linearity	5V	t_{ADCK}=1.0μs	—	±1	+2	LSB
INL	Integral non-linearity	5V	t_{ADCK}=1.0μs	—	±2	+4	LSB
I_{ADC}	Additional Power Consumption if A/D Converter is used	3V	No load (t_{ADCK}=0.5μs)	—	0.90	1.35	mA
		5V	No load (t_{ADCK}=0.5μs)	—	1.20	1.80	mA
t_{ADCK}	A/D Converter Clock Period	—	—	0.5	—	10	μs
t_{ADC}	A/D Conversion Time (Include Sample and Hold Time)	—	12 bit A/D Converter	—	16	—	t_{ADCK}
t_{ADS}	A/D Converter Sampling Time	—	—	—	4	—	t_{ADCK}
t_{ON2ST}	A/D Converter On-to-Start Time	—	—	2	—	—	μs

Comparator Electrical Characteristics

Ta=25˚C

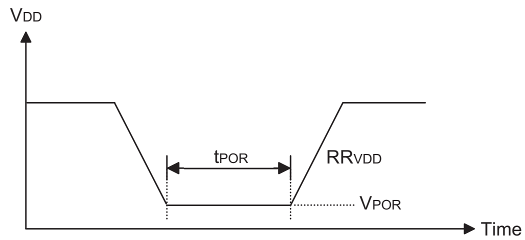
Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
V_{CMP}	Comparator operating voltage	—	—	2.2	—	5.5	V
I_{CMP}	Comparator operating current	3V	—	—	37	56	μA
		5V	—	—	130	200	μA
V_{CMPOS}	Comparator input offset voltage	—	—	-10	—	+10	mV
V_{HYS}	Hysteresis width	—		20	40	60	mV
V_{CM}	Comparator common mode voltage range	—	—	V_{SS}	—	V_{DD}-1.4V	V
A_{OL}	Comparator open loop gain	—	—	60	80	—	dB
t_{PD}	Comparator response time	3V	With 100mV overdrive[Note]	—	370	560	ns
		5V					

Note: Measured with comparator one input pin at V_{CM}=(V_{DD}-1.4)/2 while the other pin input transition from V_{SS} to (V_{CM}+100mV) or from V_{DD} to (V_{CM}-100mV).

Power on Reset Electrical Characteristics

Ta=25˚C

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



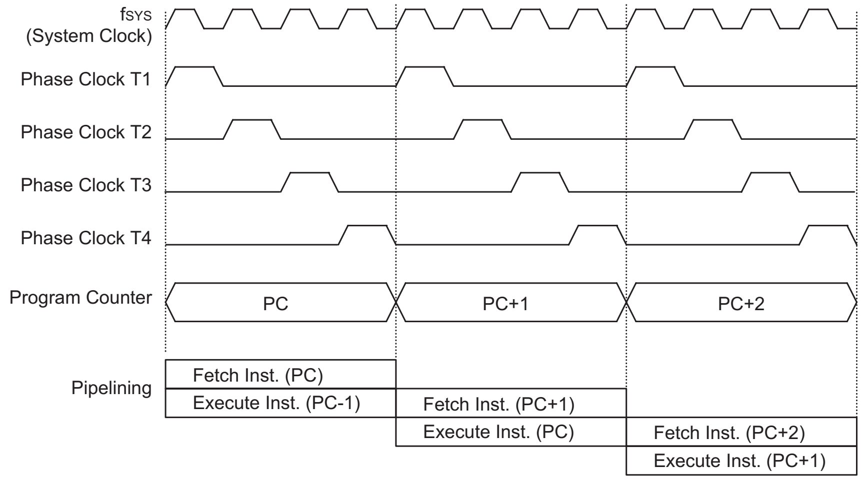
System Architecture

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

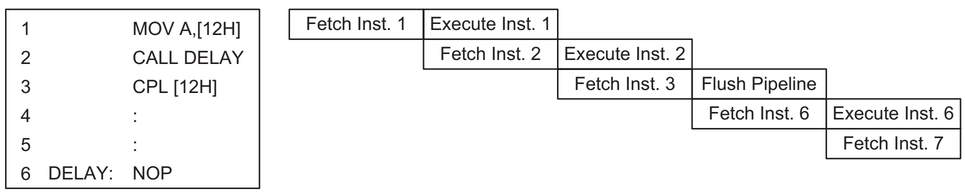
Clocking and Pipelining

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

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



System Clocking and Pipelining



Instruction Fetching

Program Counter

During program execution, the Program Counter is used to keep track of the address of the next instruction to be executed. It is automatically incremented by one each time an instruction is executed except for instructions, such as "JMP" or "CALL" that demand a jump to a non-consecutive Program Memory address. Only the lower 8 bits, known as the Program Counter Low Register, are directly addressable by the application program.

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

Device	Program Counter	
	Program Counter High Byte	PCL Register
HT66F20-1/HT68F20-1	PC9~PC8	PCL7~PCL0
HT66F30-1/HT68F30-1	PC10~PC8	PCL7~PCL0

Program Counter

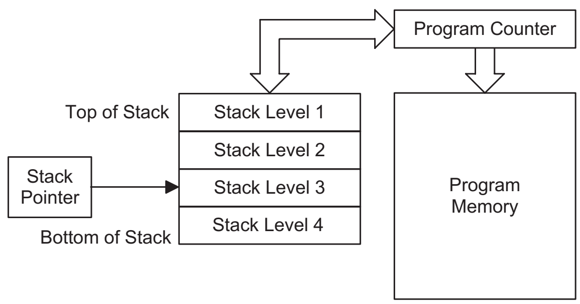
The lower byte of the Program Counter, known as the Program Counter Low register or PCL, is available for program control and is a readable and writeable register. By transferring data directly into this register, a short program jump can be executed directly; however, as only this low byte is available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

Stack

This is a special part of the memory which is used to save the contents of the Program Counter only. The stack has four levels 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.



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 these devices series the Program Memory are 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.

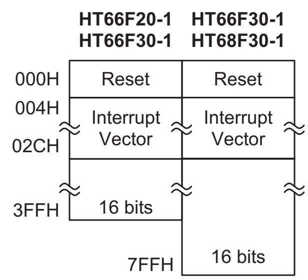
Structure

The Program Memory has a capacity of 1K×16 bits to 2K×16 bits. The Program Memory is addressed by the Program Counter and also contains data, table information and interrupt entries. Table data, which can be setup in any location within the Program Memory, is addressed by a separate table pointer register.

Device	Capacity
HT66F20-1 / HT68F20-1	1K×16
HT66F30-1 / HT68F30-1	2K×16

Special Vectors

Within the Program Memory, certain locations are reserved for the reset and interrupts. The location 000H is reserved for use by these devices reset for program initialisation. After a device reset is initiated, the program will jump to this location and begin execution.



Program Memory Structure

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.

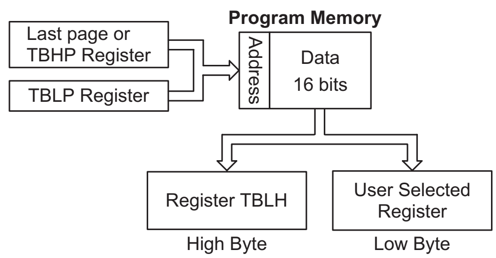


Table Program Example

The following example 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 Program Memory of the HT6xF30-1. 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.

Table Read Program Example

```
tempreg1 db    ?         ; temporary register #1
tempreg2 db    ?         ; temporary register #2
:
:
mov a,06h                ; initialise low table pointer - note that this address
                         ; is referenced
mov tblp, a              ; to the last page or present page
mov a, 07h               ; initialise high table pointer
mov tbhp, a
:
:
tabrdl 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
tabrdl 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

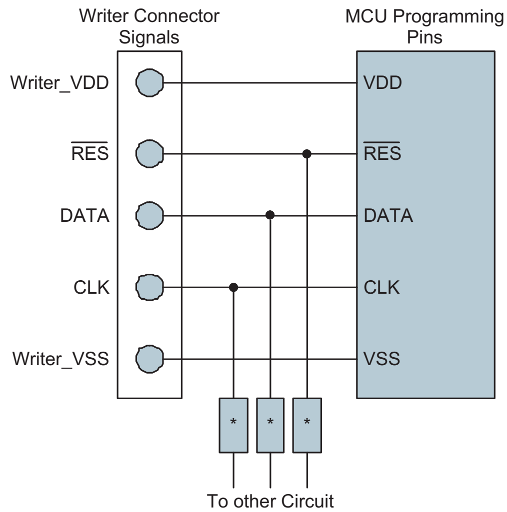
The provision of Flash type Program Memory provides the user with a means of convenient and easy upgrades and modifications to their programs on the same device.

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 sup plied with the latest program releases without removal and re-insertion of the device.

MCU Programming Pins	Function
PA0	Serial Data Input/Output
PA2	Serial Clock
$\overline{RES}$	Device Reset
VDD	Power Supply
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 ad ditional lines are required for the power supply and one line for the reset. The technical details regarding the incircuit programming of the device is beyond the scope of this document and will be supplied in supple mentary literature.

During the programming process the $\overline{RES}$ 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.

Programmer Pin	MCU Pins
$\overline{RES}$	PB0
DATA	PA0
CLK	PA2

Programmer and MCU Pins

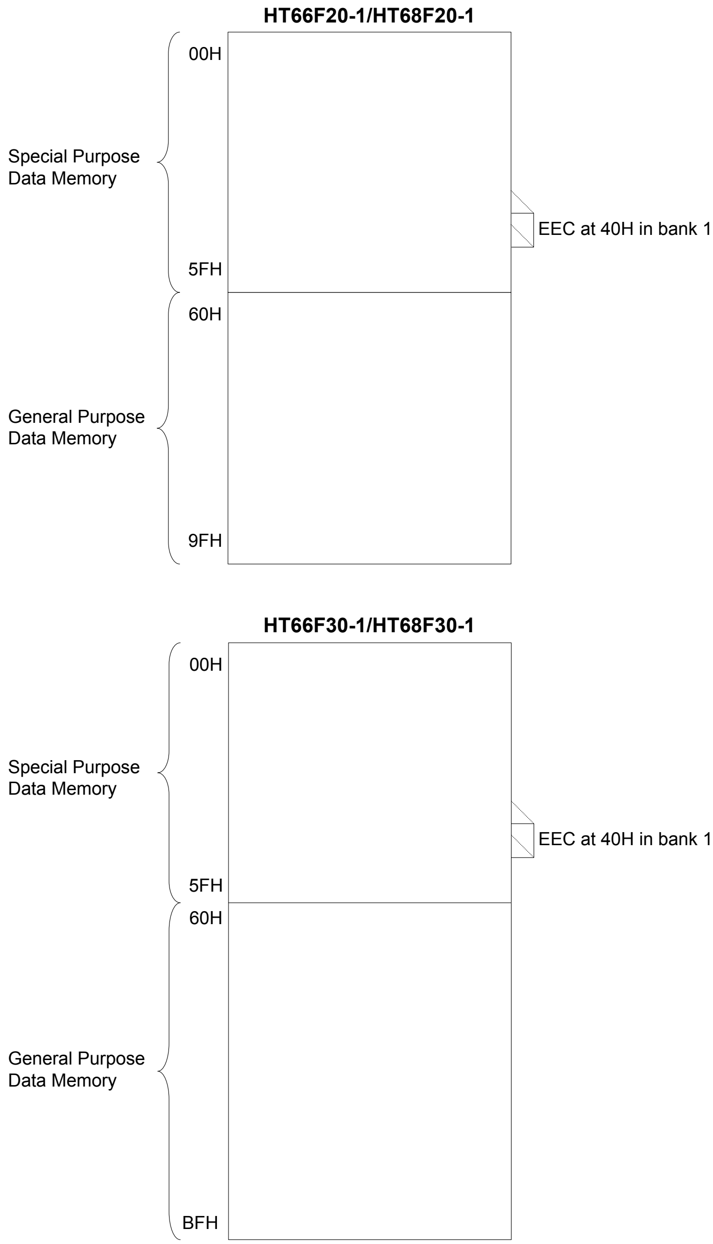
Data Memory

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.

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 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.



Data Memory Structure

Address	Bank 0, 1	Address	Bank 0	Bank 1
00H	IAR0	30H	Unused	
01H	MP0	31H	Unused	
02H	IAR1	32H	Unused	
03H	MP1	33H	Unused	
04H	BP	34H	CP0C	
05H	ACC	35H	CP1C	
06H	PCL	36H	SIMC0	
07H	TBLP	37H	SIMC1	
08H	TBLH	38H	SIMD	
09H	TBHP	39H	SIMA/SIMC2	
0AH	STATUS	3AH	TM0C0	
0BH	SMOD	3BH	TM0C1	
0CH	LVDC	3CH	TM0DL	
0DH	INTEG	3DH	TM0DH	
0EH	WDTC	3EH	TM0AL	
0FH	TBC	3FH	TM0AH	
10H	INTC0	40H	Unused	EEC
11H	INTC1	41H	EEA	
12H	INTC2	42H	EED	
13H	Unused	43H	TMPC0	
14H	MFI0	44H	Unused	
15H	MFI1	45H	Unused	
16H	MFI2	46H	Unused	
17H	Unused	47H	Unused	
18H	PAWU	48H	TM1C0	
19H	PAPU	49H	TM1C1	
1AH	PA	4AH	Unused	
1BH	PAC	4BH	TM1DL	
1CH	PBPU	4CH	TM1DH	
1DH	PB	4DH	TM1AL	
1EH	PBC	4EH	TM1AH	
1FH	PCPU	4FH	Unused	
20H	PC	50H	Unused	
21H	PCC	51H	Unused	
22H	Unused	52H	Unused	
23H	Unused	53H	Unused	
24H	Unused	54H	Unused	
25H	Unused	55H	Unused	
26H	Unused	56H	Unused	
27H	Unused	57H	Unused	
28H	Unused	58H	Unused	
29H	Unused	59H	Unused	
2AH	Unused	5AH	Unused	
2BH	Unused	5BH	Unused	
2CH	Unused	5CH	Unused	
2DH	Unused	5DH	Unused	
2EH	Unused	5EH	SCOMC	
2FH	Unused	5FH	Unused	

HT66F20-1 Special Purpose Data Memory

Address	Bank 0, 1	Address	Bank 0	Bank 1
00H	IAR0	30H	ADCR0	
01H	MP0	31H	ADCR1	
02H	IAR1	32H	ACERL	
03H	MP1	33H	Unused	
04H	BP	34H	CP0C	
05H	ACC	35H	CP1C	
06H	PCL	36H	SIMC0	
07H	TBLP	37H	SIMC1	
08H	TBLH	38H	SIMD	
09H	TBHP	39H	SIMA/SIMC2	
0AH	STATUS	3AH	TM0C0	
0BH	SMOD	3BH	TM0C1	
0CH	LVDC	3CH	TM0DL	
0DH	INTEG	3DH	TM0DH	
0EH	WDTC	3EH	TM0AL	
0FH	TBC	3FH	TM0AH	
10H	INTC0	40H	Unused	EEC
11H	INTC1	41H	EEA	
12H	INTC2	42H	EED	
13H	Unused	43H	TMPC0	
14H	MFI0	44H	Unused	
15H	MFI1	45H	PRM0	
16H	MFI2	46H	Unused	
17H	Unused	47H	Unused	
18H	PAWU	48H	TM1C0	
19H	PAPU	49H	TM1C1	
1AH	PA	4AH	TM1C2	
1BH	PAC	4BH	TM1DL	
1CH	PBPU	4CH	TM1DH	
1DH	PB	4DH	TM1AL	
1EH	PBC	4EH	TM1AH	
1FH	PCPU	4FH	TM1BL	
20H	PC	50H	TM1BH	
21H	PCC	51H	Unused	
22H	Unused	52H	Unused	
23H	Unused	53H	Unused	
24H	Unused	54H	Unused	
25H	Unused	55H	Unused	
26H	Unused	56H	Unused	
27H	Unused	57H	Unused	
28H	Unused	58H	Unused	
29H	Unused	59H	Unused	
2AH	Unused	5AH	Unused	
2BH	Unused	5BH	Unused	
2CH	Unused	5CH	Unused	
2DH	Unused	5DH	Unused	
2EH	ADRL	5EH	SCOMC	
2FH	ADRH	5FH	Unused	

HT66F30-1 Special Purpose Data Memory

Address	Bank 0, 1	Address	Bank 0	Bank 1
00H	IAR0	30H	Unused	
01H	MP0	31H	Unused	
02H	IAR1	32H	Unused	
03H	MP1	33H	Unused	
04H	BP	34H	CP0C	
05H	ACC	35H	CP1C	
06H	PCL	36H	SIMC0	
07H	TBLP	37H	SIMC1	
08H	TBLH	38H	SIMD	
09H	TBHP	39H	SIMA/SIMC2	
0AH	STATUS	3AH	TM0C0	
0BH	SMOD	3BH	TM0C1	
0CH	LVDC	3CH	TM0DL	
0DH	INTEG	3DH	TM0DH	
0EH	WDTC	3EH	TM0AL	
0FH	TBC	3FH	TM0AH	
10H	INTC0	40H	Unused	EEC
11H	INTC1	41H	EEA	
12H	INTC2	42H	EED	
13H	Unused	43H	TMPC0	
14H	MFI0	44H	Unused	
15H	MFI1	45H	Unused	
16H	MFI2	46H	Unused	
17H	Unused	47H	Unused	
18H	PAWU	48H	TM1C0	
19H	PAPU	49H	TM1C1	
1AH	PA	4AH	Unused	
1BH	PAC	4BH	TM1DL	
1CH	PBPU	4CH	TM1DH	
1DH	PB	4DH	TM1AL	
1EH	PBC	4EH	TM1AH	
1FH	PCPU	4FH	Unused	
20H	PC	50H	Unused	
21H	PCC	51H	Unused	
22H	Unused	52H	Unused	
23H	Unused	53H	Unused	
24H	Unused	54H	Unused	
25H	Unused	55H	Unused	
26H	Unused	56H	Unused	
27H	Unused	57H	Unused	
28H	Unused	58H	Unused	
29H	Unused	59H	Unused	
2AH	Unused	5AH	Unused	
2BH	Unused	5BH	Unused	
2CH	Unused	5CH	Unused	
2DH	Unused	5DH	Unused	
2EH	Unused	5EH	SCOMC	
2FH	Unused	5FH	Unused	

HT68F20-1 Special Purpose Data Memory

Address	Bank 0, 1	Address	Bank 0	Bank 1
00H	IAR0	30H	Unused	
01H	MP0	31H	Unused	
02H	IAR1	32H	Unused	
03H	MP1	33H	Unused	
04H	BP	34H	CP0C	
05H	ACC	35H	CP1C	
06H	PCL	36H	SIMC0	
07H	TBLP	37H	SIMC1	
08H	TBLH	38H	SIMD	
09H	TBHP	39H	SIMA/SIMC2	
0AH	STATUS	3AH	TM0C0	
0BH	SMOD	3BH	TM0C1	
0CH	LVDC	3CH	TM0DL	
0DH	INTEG	3DH	TM0DH	
0EH	WDTC	3EH	TM0AL	
0FH	TBC	3FH	TM0AH	
10H	INTC0	40H	Unused	EEC
11H	INTC1	41H	EEA	
12H	INTC2	42H	EED	
13H	Unused	43H	TMPC0	
14H	MFI0	44H	Unused	
15H	MFI1	45H	PRM0	
16H	MFI2	46H	Unused	
17H	Unused	47H	Unused	
18H	PAWU	48H	TM1C0	
19H	PAPU	49H	TM1C1	
1AH	PA	4AH	TM1C2	
1BH	PAC	4BH	TM1DL	
1CH	PBPU	4CH	TM1DH	
1DH	PB	4DH	TM1AL	
1EH	PBC	4EH	TM1AH	
1FH	PCPU	4FH	TM1BL	
20H	PC	50H	TM1BH	
21H	PCC	51H	Unused	
22H	Unused	52H	Unused	
23H	Unused	53H	Unused	
24H	Unused	54H	Unused	
25H	Unused	55H	Unused	
26H	Unused	56H	Unused	
27H	Unused	57H	Unused	
28H	Unused	58H	Unused	
29H	Unused	59H	Unused	
2AH	Unused	5AH	Unused	
2BH	Unused	5BH	Unused	
2CH	Unused	5CH	Unused	
2DH	Unused	5DH	Unused	
2EH	Unused	5EH	SCOMC	
2FH	Unused	5FH	Unused	

HT68F30-1 Special Purpose Data Memory

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.

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.

Bank Pointer – BP

The Data Memory is divided into two banks. Selecting the Data Memory area is achieved using the Bank Pointer. Bit 0 of the Bank Pointer is used to select Data Memory Banks 0 or 1.

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. 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.

As both the Program Memory and Data Memory share the same Bank Pointer Register, care must be taken during programming.

- **HT66F30-1/HT68F30-1**

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

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

All 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 up to 64×8 bits. 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
HT66F20-1/HT68F20-1	32×8	00H~1FH
HT66F30-1/HT68F30-1	64×8	00H~3FH

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

- **HT66F20-1/HT68F20-1**

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

- **HT66F30-1/HT68F30-1**

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

EEA Register

- **HT66F20-1/HT68F20-1**

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

"x" unknown

Bit 7~5 Unimplemented, read as "0"

Bit 4~0 **D4~D0**: Data EEPROM address
Data EEPROM address bit 4~bit 0

- **HT66F30-1/HT68F30-1**

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 **D5~D0**: Data EEPROM address
Data EEPROM address bit 5~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 Undefined, 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 **D7~D0**: Data EEPROM address
Data EEPROM address 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

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. The EEPROM address of the data to be written must then be placed in the EEA register and the data placed in the EED register. If the WR bit in the EEC register is now set high, an internal write cycle will then be initiated. 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 or read interrupt is generated when an EEPROM write or read 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 Consideration

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.

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, EED                     ; move read data to register
MOV  READ_DATA, A
```

Writing Data from the EEPROM – Polling Method

```
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
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

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

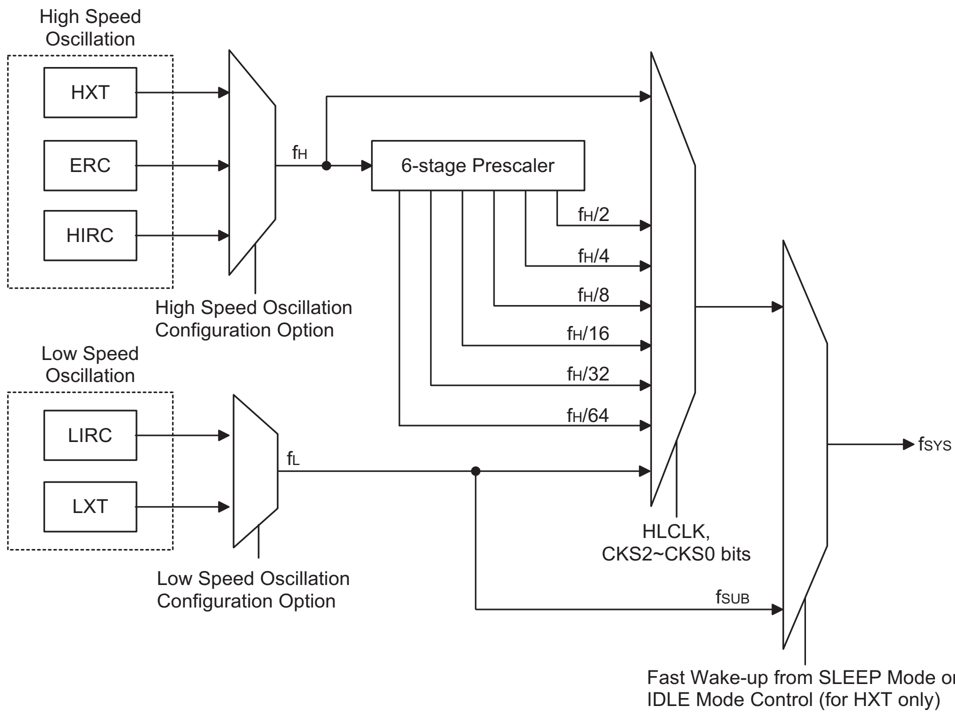
Type	Name	Freq.	Pins
External Crystal	HXT	400kHz~20MHz	OSC1/OSC2
External RC	ERC	8MHz	OSC1
Internal High Speed RC	HIRC	4, 8, 12MHz	—
External Low Speed Crystal	LXT	32.768kHz	XT1/XT2
Internal Low Speed RC	LIRC	32kHz	—

Oscillator Types

System Clock Configurations

There are five methods of generating the system clock, three high speed oscillators and two low speed oscillators. The high speed oscillators are the external crystal/ceramic oscillator, external RC network oscillator and the internal 4MHz, 8MHz or 12MHz RC oscillator. The two low speed oscillators are the internal 32kHz RC oscillator and the external 32.768kHz crystal oscillator. 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 and as the system clock can be dynamically selected.

The actual source clock used for each of the high speed and low speed oscillators is chosen via configuration options. The frequency of the slow speed or high speed system clock is also determined using the HLCLK bit and CKS2~CKS0 bits in the SMOD register. Note that two oscillator selections must be made namely one high speed and one low speed system oscillators. It is not possible to choose a no-oscillator selection for either the high or low speed oscillator.

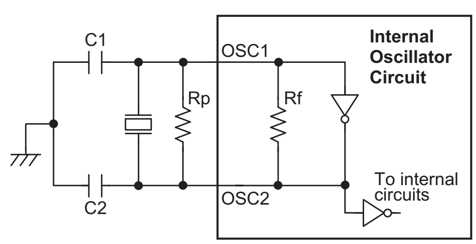


System Clock Configurations

External Crystal/Ceramic Oscillator – HXT

The External Crystal/Ceramic System Oscillator is one of the high frequency oscillator choices, which is selected via configuration option. For most crystal oscillator configurations, the simple connection of a crystal across OSC1 and OSC2 will create the necessary phase shift and feedback for oscillation, without requiring external capacitors. However, for some crystal types and frequencies, to ensure oscillation, it may be necessary to add two small value capacitors, C1 and C2. Using a ceramic resonator will usually require two small value capacitors, C1 and C2, to be connected as shown for oscillation to occur. The values of C1 and C2 should be selected in consultation with the crystal or resonator manufacturer's specification.

For oscillator stability and to minimise the effects of noise and crosstalk, it is important to ensure thatthe crystal and any associated resistors andcapacitors along with interconnectinglines are all located as close to the MCUas possible.



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

Crystal/Resonator Oscillator – HXT

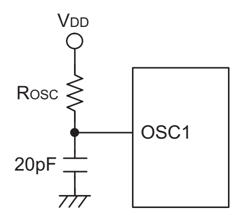
Crystal Oscillator C1 and C2 Values		
Crystal Frequency	C1	C2
12MHz	0pF	0pF
8MHz	0pF	0pF
4MHz	0pF	0pF
1MHz	100pF	100pF
Note: 1. C1 and C2 values are for guidance only.		

Crystal Recommended Capacitor Values

External RC Oscillator – ERC

Using the ERC oscillator only requires that a resistor, with a value between 56kΩ and 2.4MΩ, is connected between OSC1 and VDD, and a capacitor is connected between OSC1 and ground, providing a low cost oscillator configuration. It is only the external resistor that determines the oscillation frequency; the external capacitor has no influence over the frequency and is connected for stability purposes only. Device trimming during the manufacturing process and the inclusion of internal frequency compensation circuits are used to ensure that the influence of the power supply voltage, temperature and process variations on the oscillation frequency are minimised. As a resistance/frequency reference point, it can be noted that with an external 120kΩ resistor connected and with a 5V voltage power supply and temperature of 25˚C degrees, the oscillator will have a frequency of 8MHz within a tolerance of 2%. Here only the OSC1 pin is used, which is shared with I/O pin PB1, leaving pin PB2 free for use as a normal I/O pin.

For oscillator stability and to minimise the effects of noise and crosstalk, it is important to locate the capacitor and resistoras close to the MCU as possible.



External RC Oscillator — ERC

Internal RC Oscillator – HIRC

The internal RC oscillator is a fully integrated system oscillator requiring no external components. The internal RC oscillator has three fixed frequencies of either 4MHz, 8MHz or 12MHz. Device trimming during the manufacturing process and the inclusion of internal frequency compensation circuits are used to ensure that the influence of the power supply voltage, temperature and process variations on the oscillation frequency are minimised. As a result, at a power supply of either 3V or 5V and at a temperature of 25˚C degrees, the fixed oscillation frequency of 4MHz, 8MHz or 12MHz will have a tolerance within 2%. Note that if this internal system clock option is selected, as it requires no external pins for its operation, I/O pins PB1 and PB2 are free for use as normal I/O pins.

External 32.768kHz Crystal Oscillator – LXT

The External 32.768kHz Crystal System Oscillator is one of the low frequency oscillator choices, which is selected via configuration option. This clock source has a fixed frequency of 32.768kHz and requires a 32.768kHz crystal to be connected between pins XT1 and XT2. The external resistor and capacitor components connected to the 32.768kHz crystal are necessary to provide oscillation. For applications where precise frequencies are essential, these components may be required to provide frequency compensation due to different crystal manufacturing tolerances. During power-up there is a time delay associated with the LXT oscillator waiting for it to start-up.

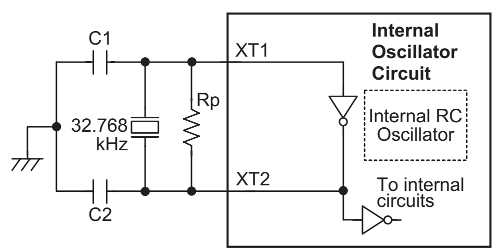
When the microcontroller enters the SLEEP or IDLE Mode, the system clock is switched off to stop microcontroller activity and to conserve power. However, in many microcontroller applications it may be necessary to keep the internal timers operational even when the microcontroller is in the SLEEP or IDLE Mode. To do this, another clock, independent of the system clock, must be provided.

However, for some crystals, to ensure oscillation and accurate frequency generation, it is necessary to add two small value external capacitors, C1 and C2. The exact values of C1 and C2 should be selected in consultation with the crystal or resonator manufacturer's specification. The external parallel feedback resistor, Rp, is required.

Some configuration options determine if the XT1/XT2 pins are used for the LXT oscillator or as I/O pins.

- If the LXT oscillator is not used for any clock source, the XT1/XT2 pins can be used as normal I/O pins.
- If the LXT oscillator is used for any clock source, the 32.768kHz crystal should be connected to the XT1/XT2 pins.

For oscillator stability and to minimise the effects of noise and crosstalk, it is important to ensure thatthe crystal and any associated resistors andcapacitors along with interconnectinglines are all located as close to the MCUas possible.



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

External LXT Oscillator

LXT Oscillator C1 and C2 Values		
Crystal Frequency	C1	C2
32.768kHz	10pF	10pF
Note: 1. C1 and C2 values are for guidance only. 2. R_P=5MΩ~10MΩ is recommended.		

32.768kHz Crystal Recommended Capacitor Values

LXT Oscillator Low Power Function

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

LXTLP Bit	LXT Mode
0	Quick Start
1	Low-power

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

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

Internal 32kHz Oscillator – LIRC

The Internal 32kHz System Oscillator is one of the low frequency oscillator choices, which is selected via configuration option. 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. As a result, at a power supply of 5V and at a temperature of 25˚C degrees, the fixed oscillation frequency of 32kHz will have a tolerance within 10%.

Supplementary Clocks

The low speed oscillators, in addition to providing a system clock source are also used to provide a clock source to two other devices functions. These are the Watchdog Timer and the Time Base Interrupts.

Operating Modes and System Clocks

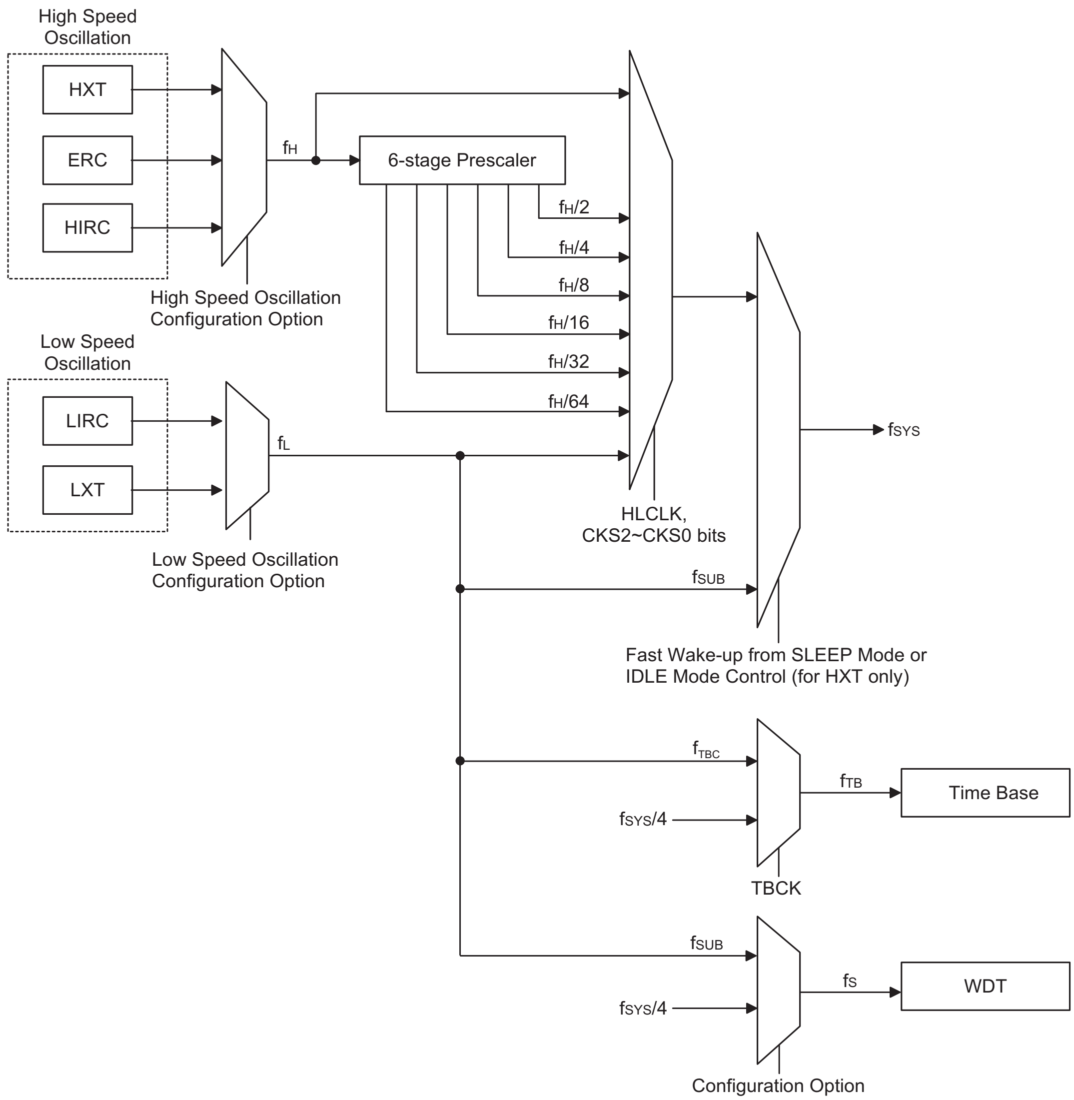
Present day applications require that their microcontrollers have high performance but often still demand that they consume as little power as possible, conflicting requirements that are especially true in battery powered portable applications. The fast clocks required for high performance will by their 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 devices have many different clock sources for both the CPU and peripheral function operation. By providing the user with a wide range of clock options using configuration options and register programming, a clock system can be configured to obtain maximum application performance.

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. The high speed system clock can be sourced from either a HXT, ERC or HIRC oscillator, selected via a configuration option. The low speed system clock source can be sourced from internal clock f_L. If f_L is selected then it can be sourced by either the LXT or LIRC oscillators, selected via a configuration option. The other choice, which is a divided version of the high speed system oscillator has a range of $f_H/2$~$f_H/64$.

There are two additional internal clocks for the peripheral circuits, the substitute clock, f_{SUB}, and the Time Base clock, f_{TBC}. Each of these internal clocks is sourced by either the LXT or LIRC oscillators, selected via configuration options. The f_{SUB} clock is used to provide a substitute clock for the microcontroller just after a wake-up has occurred to enable faster wake-up times.



System Clock Configurations

Note: When the system clock source f_{SYS} is switched to f_L from f_H, the high speed oscillation will stop to conserve the power. Thus there is no f_H~f_H/64 for peripheral circuit to use.

Together with $f_{SYS}/4$ it is also used as one of the clock sources for the Watchdog timer. The f_{TBC} clock is used as a source for the Time Base interrupt functions and for the TMs.

System Operation Modes

There are six 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 four modes, the SLEEP0, SLEEP1, IDLE0 and IDLE1 Mode are used when the microcontroller CPU is switched off to conserve power.

Operation Mode	Description				
	CPU	f_{SYS}	f_{SUB}	f_S	f_{TBC}
NORMAL Mode	On	f_H~f_H/64	On	On	On
SLOW Mode	On	f_L	On	On	On
IDLE0 Mode	Off	Off	On	On/Off	On
IDLE1 Mode	Off	On	On	On	On
SLEEP0 Mode	Off	Off	Off	Off	Off
SLEEP1 Mode	Off	Off	On	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 one of the high speed oscillators. This mode operates allowing the microcontroller to operate normally with a clock source will come from one of the high speed oscillators, either the HXT, ERC or HIRC oscillators. 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~CKS0 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 a slower speed clock source. The clock source used will be from one of the low speed oscillators, either the LXT or the LIRC. Running the microcontroller in this mode allows it to run with much lower operating currents. In the SLOW Mode, the f_H is off.

SLEEP0 Mode

The SLEEP0 Mode is entered when an HALT instruction is executed and when the IDLEN bit in the SMOD register is low. In the SLEEP0 mode the CPU will be stopped, and the f_{SUB} and f_S clocks will be stopped too, and the Watchdog Timer function is disabled. In this mode, the LVDEN is must set to "0". If the LVDEN is set to "1", it won't enter the SLEEP0 Mode.

SLEEP1 Mode

The SLEEP1 Mode is entered when an HALT instruction is executed and when the IDLEN bit in the SMOD register is low. In the SLEEP1 mode the CPU will be stopped. However, the f_{SUB} and f_S clocks will continue to operate if the LVDEN is "1" or the Watchdog Timer function is enabled and if its clock source is chosen via configuration option to come from the f_{SUB}.

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 will be inhibited from driving the CPU but some peripheral functions will remain operational such as the Watchdog Timer, TMs and SIM. In the IDLE0 Mode, the system oscillator will be stopped. In the IDLE0 Mode the Watchdog Timer clock, f_S, will either be on or off depending upon the f_S clock source. If the source is $f_{SYS}/4$ then the f_S clock will be off, and if the source comes from f_{SUB} then f_S will be on.

IDLE1 Mode

The IDLE1 Mode is entered when an 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 such as the Watchdog Timer, TMs and SIM. In the IDLE1 Mode, the system oscillator will continue to run, and this system oscillator may be high speed or low speed system oscillator. In the IDLE1 Mode the Watchdog Timer clock, f_S, will be on. If the source is $f_{SYS}/4$ then the f_S clock will be on, and if the source comes from f_{SUB} then f_S will be on.

Control Register

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

SMOD Register

Bit	7	6	5	4	3	2	1	0
Name	CKS2	CKS1	CKS0	FSTEN	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_{LXT} or f_{LIRC})
001: f_L (f_{LXT} or 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 can be either the LXT or LIRC, a divided version of the high speed system oscillator can also be chosen as the system clock source.

Bit 4 **FSTEN:** Fast Wake-up Control (only for HXT)
0: Disable
1: Enable

This is the Fast Wake-up Control bit which determines if the f_{SUB} clock source is initially used after these devices wake up. When the bit is high, the f_{SUB} clock source can be used as a temporary system clock to provide a faster wake up time as the f_{SUB} clock is available.

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 or a wake-up has occurred. The flag will be low when in the SLEEP0 Mode but after a wake-up has occurred, the flag will change to a high level after 1024 clock cycles if the LXT oscillator is used and 1~2 clock cycles if the LIRC oscillator is used.

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 these devices are 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 devices 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 1024 clock cycles if the HXT oscillator is used and after 15~16 clock cycles if the ERC or HIRC oscillator is used.

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 these devices 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 these devices will enter the SLEEP Mode when a HALT instruction is executed.

Bit 0 **HLCLK:** System clock selection
0: $f_H/2 \sim f_H/64$ or f_L
1: f_H
This bit is used to select if the f_H clock or the $f_H/2 \sim f_H/64$ 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 $f_H/2 \sim f_H/64$ 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.

Fast Wake-up

To minimise power consumption these devices can enter the SLEEP or IDLE0 Mode, where the system clock source to these devices will be stopped. However when these devices are woken up again, it can take a considerable time for the original system oscillator to restart, stabilise and allow normal operation to resume. To ensure the device is up and running as fast as possible a Fast Wake-up function is provided, which allows f_{SUB}, namely either the LXT or LIRC oscillator, to act as a temporary clock to first drive the system until the original system oscillator has stabilised. As the clock source for the Fast Wake-up function is f_{SUB}, the Fast Wake-up function is only available in the SLEEP1 and IDLE0 modes. When these devices are woken up from the SLEEP0 mode, the Fast Wake-up function has no effect because the f_{SUB} clock is stopped. The Fast Wake-up enable/disable function is controlled using the FSTEN bit in the SMOD register.

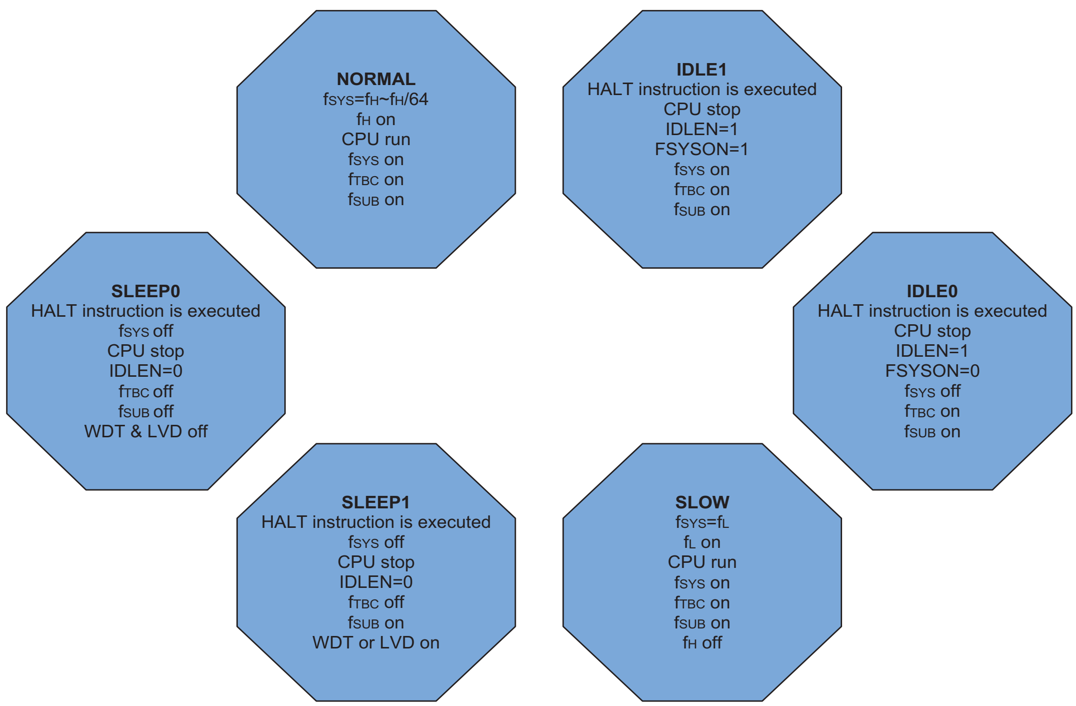
If the HXT oscillator is selected as the NORMAL Mode system clock, and if the Fast Wake-up function is enabled, then it will take one to two t_{SUB} clock cycles of the LIRC or LXT oscillator for the system to wake-up. The system will then initially run under the f_{SUB} clock source until 1024 HXT clock cycles have elapsed, at which point the HTO flag will switch high and the system will switch over to operating from the HXT oscillator.

If the ERC or HIRC oscillator or LIRC oscillator is used as the system oscillator then it will take 15~16 clock cycles of the ERC or HIRC or 1~2 cycles of the LIRC to wake up the system from the SLEEP or IDLE0 Mode. The Fast Wake-up bit, FSTEN will have no effect in these cases.

System Oscillator	FSTEN Bit	Wake-up Time (SLEEP0 Mode)	Wake-up Time (SLEEP1 Mode)	Wake-up Time (IDLE0 Mode)	Wake-up Time (IDLE1 Mode)
HXT	0	1024 HXT cycles	1024 HXT cycles		1~2 HXT cycles
	1	1024 HXT cycles	1~2 f_{SUB} cycles (System runs with f_{SUB} first for 1024 HXT cycles and then switches over to run with the HXT clock)		1~2 HXT cycles
ERC	x	15~16 ERC cycles	15~16 ERC cycles		1~2 ERC cycles
HIRC	x	15~16 HIRC cycles	15~16 HIRC cycles		1~2 HIRC cycles
LIRC	x	1~2 LIRC cycles	1~2 LIRC cycles		1~2 LIRC cycles
LXT	x	1024 LTX cycles	1024 LXT cycles		1~2 LXT cycles

Wake-Up Times

Note that if the Watchdog Timer is disabled, which means that the LXT and LIRC are all both off, then there will be no Fast Wake-up function available when these devices wake-up from the SLEEP0 Mode.



Operating Mode Switching

These devices 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.

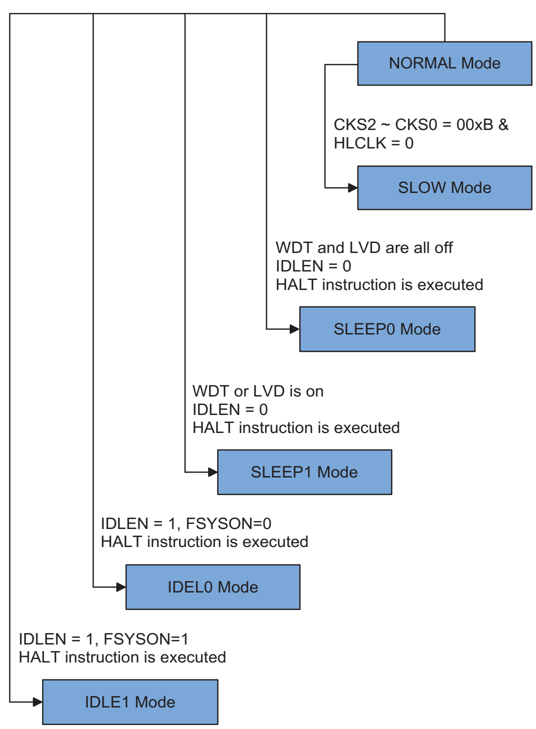
In simple terms, Mode Switching between the NORMAL Mode and SLOW Mode is executed using the HLCLK bit and CKS2~CKS0 bits in the SMOD register while Mode Switching from the NORMAL/SLOW Modes to the SLEEP/IDLE Modes is executed via the HALT instruction. When a HALT instruction is executed, whether these devices enter 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.

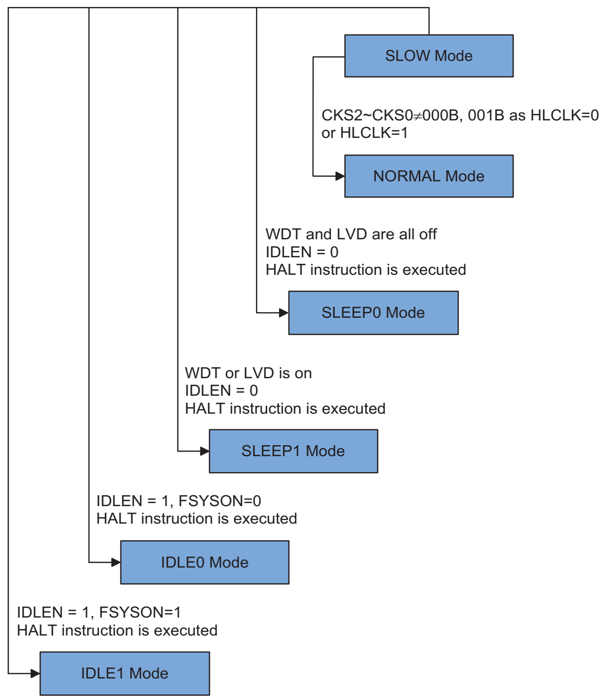
When the HLCLK bit switches to a low level, which implies that clock source is switched from the high speed clock source, f_H, to the clock source, $f_H/2$~$f_H/64$ or f_L. If the clock is from the f_L, the high speed clock source will stop running to conserve power. When this happens it must be noted that the $f_H/16$ and $f_H/64$ internal clock sources will also stop running, which may affect the operation of other internal functions such as the TMs and the SIM. The accompanying flowchart shows what happens when these devices move 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 is sourced from the LXT or the LIRC oscillators and therefore requires these oscillators to be stable before full mode switching occurs. This is monitored using the LTO bit in the SMOD register.





SLOW Mode to NORMAL Mode Switching

In SLOW Mode the system uses either the LXT or 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 SLEEP0 Mode

There is only one way for these devices to enter the SLEEP0 Mode and that is to execute the "HALT" instruction in the application program with the IDLEN bit in SMOD register equal to "0" and the WDT and LVD both off. When this instruction is executed under the conditions described above, the following will occur:

- The system clock, WDT clock and Time Base clock will be stopped and the application program will stop at the "HALT" instruction.
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and stopped no matter if the WDT clock source originates from the f_{SUB} clock or from the system clock.
- 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 SLEEP1 Mode

There is only one way for these devices to enter the SLEEP1 Mode and that is to execute the "HALT" instruction in the application program with the IDLEN bit in SMOD register equal to "0" and the WDT or LVD on. When this instruction is executed under the conditions described above, the following will occur:

- The system clock and Time Base clock will be stopped and the application program will stop at the "HALT" instruction, but the WDT or LVD will remain with the clock source coming from the f_{SUB} clock.
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and resume counting if the WDT clock source is selected to come from the f_{SUB} clock as the WDT is enabled.
- 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 these devices 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_{SUB} clock will be on.
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and resume counting if the WDT clock source is selected to come from the f_{SUB} clock and the WDT is enabled. The WDT will stop if its clock source originates from the system clock.
- 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 these devices 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 conditions described above, the following will occur:

- The system clock and Time Base clock and f_{SUB} 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 if the WDT is enabled regardless of the WDT clock source which originates from the f_{SUB} clock or from the system clock.
- 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 these devices 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 these devices. 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 unbonded 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. Also note that additional standby current will also be required if the configuration options have enabled the LXT or LIRC oscillator.

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, these devices will experience a full system reset, however, if these devices are 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 these devices 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.

Programming Considerations

The HXT and LXT oscillators both use the same SST counter. For example, if the system is woken up from the SLEEP0 Mode and both the HXT and LXT oscillators need to start-up from an off state. The LXT oscillator uses the SST counter after HXT oscillator has finished its SST period.

- If these devices are woken up from the SLEEP0 Mode to the NORMAL Mode, the high speed system oscillator needs an SST period. These devices will execute first instruction after HTO is "1". At this time, the LXT oscillator may not be stability if f_{SUB} is from LXT oscillator. The same situation occurs in the power-on state. The LXT oscillator is not ready yet when the first instruction is executed.
- If these devices are woken up from the SLEEP1 Mode to NORMAL Mode, and the system clock source is from HXT oscillator and FSTEN is "1", the system clock can be switched to the LXT or LIRC oscillator after wake up.
- There are peripheral functions, such as WDT, TMs and SIM, for which the f_{SYS} is used. If the system clock source is switched from f_H to f_L, the clock source to the peripheral functions mentioned above will change accordingly.
- The on/off condition of f_{SUB} and f_S depends upon whether the WDT is enabled or disabled as the WDT clock source is selected from f_{SUB}.

Watchdog Timer

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

Watchdog Timer Clock Source

The Watchdog Timer clock source is provided by the internal clock, f_S, which is in turn supplied by one of two sources selected by configuration option: f_{SUB} or $f_{SYS}/4$. The f_{SUB} clock can be sourced from either the LXT or LIRC oscillators, again chosen via a configuration option. 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 V_{DD}, temperature and process variations. The LXT oscillator is supplied by an external 32.768kHz crystal. The other Watchdog Timer clock source option is the $f_{SYS}/4$ clock. The Watchdog Timer clock source can originate from its own internal LIRC oscillator, the LXT oscillator or $f_{SYS}/4$. It is divided by a value of 2^8 to 2^{15}, using the WS2~WS0 bits in the WDTC register to obtain the required Watchdog Timer time-out period.

Watchdog Timer Control Register

A single register, WDTC, controls the required timeout period as well as the enable/disable operation. This register together with several configuration options control the overall operation of the Watchdog Timer.

WDTC Register

Bit	7	6	5	4	3	2	1	0
Name	FSYSON	WS2	WS1	WS0	WDTEN3	WDTEN2	WDTEN1	WDTEN0
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: $256/f_S$
001: $512/f_S$
010: $1024/f_S$
011: $2048/f_S$
100: $4096/f_S$
101: $8192/f_S$
110: $16384/f_S$
111: $32768/f_S$
These three bits determine the division ratio of the Watchdog Timer source clock, which in turn determines the timeout period.

Bit 3~0 **WDTEN3, WDTEN2, WDTEN1, WDTEN0**: WDT Software Control
1010: Disable
Other: Enable

Watchdog Timer Operation

The Watchdog Timer operates by providing a device reset when its timer overflows. This means that in the application program and during normal operation the user has to strategically clear the Watchdog Timer before it overflows to prevent the Watchdog Timer from executing a reset. This is done using the clear watchdog 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. Some of the Watchdog Timer options, such as enable/disable, clock source selection and clear instruction type are selected using configuration options. In addition to a configuration option to enable/disable the Watchdog Timer, there are also four bits, WDTEN3~WDTEN0, in the WDTC register to offer an additional enable/disable control of the Watchdog Timer. To disable the Watchdog Timer, as well as the configuration option being set to disable, the WDTEN3~WDTEN0 bits must also be set to a specific value of "1010". Any other values for these bits will keep the Watchdog Timer enabled, irrespective of the configuration enable/disable setting. After power on these bits will have the value of 1010. If the Watchdog Timer is used it is recommended that they are set to a value of 0101 for maximum noise immunity. Note that if the Watchdog Timer has been disabled, then any instruction relating to its operation will result in no operation.

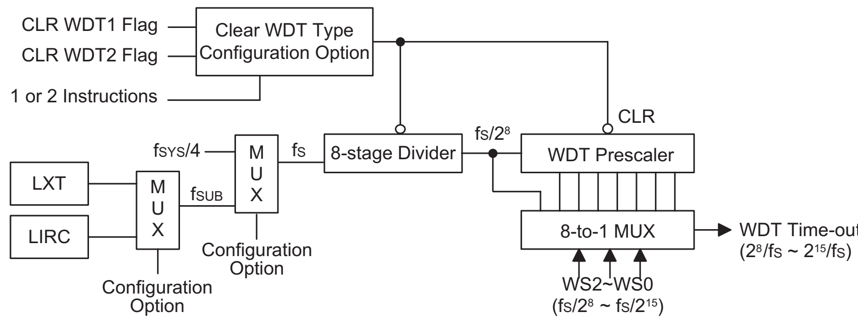
WDT Configuration Option	WDTEN3~WDTEN0 Bits	WDT
WDT Enable	xxxx	Enable
WDT Disable	Except 1010	Enable
WDT Disable	1010	Disable

Watchdog Timer Enable/Disable Control

Under normal program operation, a Watchdog Timer time-out will initialise a device reset and set the status bit TO. However, if the system is in the SLEEP or IDLE Mode, when a Watchdog Timer time-out occurs, the TO bit in the status register will be set and only the Program Counter and Stack Pointer will be reset. Three methods can be adopted to clear the contents of the Watchdog Timer. The first is an external hardware reset, which means a low level on the $\overline{RES}$ pin, the second is using the Watchdog Timer software clear instructions and the third is via a HALT instruction.

There are two methods of using software instructions to clear the Watchdog Timer, one of which must be chosen by configuration option. The first option is to use the single "CLR WDT" instruction while the second is to use the two commands "CLR WDT1" and "CLR WDT2". For the first option, a simple execution of "CLR WDT" will clear the WDT while for the second option, both "CLR WDT1" and "CLR WDT2" must both be executed alternately to successfully clear the Watchdog Timer. Note that for this second option, if "CLR WDT1" is used to clear the Watchdog Timer, successive executions of this instruction will have no effect, only the execution of a "CLR WDT2" instruction will clear the Watchdog Timer. Similarly after the "CLR WDT2" instruction has been executed, only a successive "CLR WDT1" instruction can clear the Watchdog Timer.

The maximum time out period is when the 2^{15} division ratio is selected. As an example, with a 32.768kHz LXT 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. If the $f_{SYS}/4$ clock is used as the Watchdog Timer clock source, it should be noted that when the system enters the SLEEP or IDLE0 Mode, then the instruction clock is stopped and the Watchdog Timer may lose its protecting purposes. For systems that operate in noisy environments, using the f_{SUB} clock source is strongly recommended.



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.

In addition to the power-on reset, situations may arise where it is necessary to forcefully apply a reset condition when the is running. One example of this is where after power has been applied and the is already running, the $\overline{RES}$ line is forcefully pulled low. In such a case, known as a normal operation reset, some of the registers remain unchanged allowing the to proceed with normal operation after the reset line is allowed to return high.

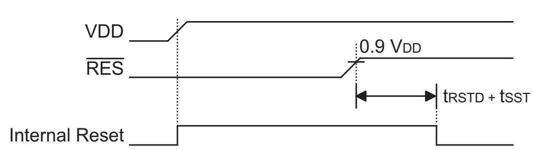
Another type of reset is when the Watchdog Timer overflows and resets the microcontroller. All types of reset operations result in different register conditions being setup. Another reset exists in the form of a Low Voltage Reset, LVR, where a full reset, similar to the $\overline{RES}$ reset is implemented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

Reset Functions

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

Power-on Reset

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



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

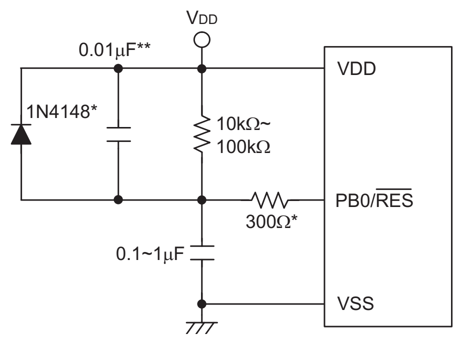
Power-on Reset Timing Chart

$\overline{\text{RES}}$ Pin

As the reset pin is shared with PB.0, the reset function must be selected using a configuration option. Although the microcontroller has an internal RC reset function, if the V_{DD} power supply rise time is not fast enough or does not stabilise quickly at power-on, the internal reset function may be incapable of providing proper reset operation. For this reason it is recommended that an external RC network is connected to the $\overline{\text{RES}}$ pin, whose additional time delay will ensure that the $\overline{\text{RES}}$ pin remains low for an extended period to allow the power supply to stabilise. During this time delay, normal operation of the microcontroller will be inhibited. After the $\overline{\text{RES}}$ line reaches a certain voltage value, the reset delay time t_{RSTD} is invoked to provide an extra delay time after which the microcontroller will begin normal operation. The abbreviation SST in the figures stands for System Start-up Timer.

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

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



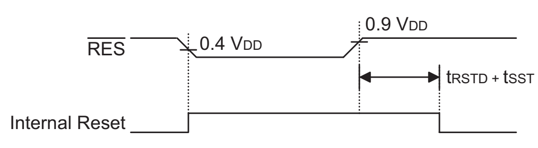
Note: * It is recommended that this component is added for added ESD protection.

** It is recommended that this component is added in environments where power line noise is significant.

Extern $\overline{\text{RES}}$ Circuit

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

Pulling the $\overline{\text{RES}}$ Pin low using external hardware will also execute a device reset. In this case, as in the case of other resets, the Program Counter will reset to zero and program execution initiated from this point.

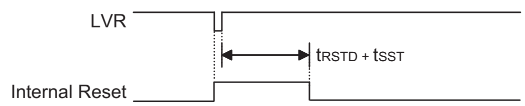


Note: t_{RSTD} is power-on delay, typical time=100ms

$\overline{\text{RES}}$ Reset Timing Chart

Low Voltage Reset – LVR

These microcontrollers contain a low voltage reset circuit in order to monitor the supply voltage of these devices, which are selected via a configuration option. If the supply voltage of the device drops to within a range of 0.9V~V_{LVR} such as might occur when changing the battery, the LVR will automatically reset the device internally. 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.

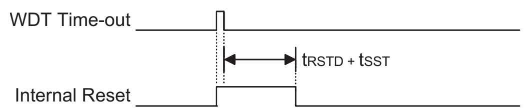


Note: t_{RSTD} is power-on delay, typical time=100ms

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 $\overline{RES}$ pin reset except that the Watchdog time-out flag TO will be set to "1".

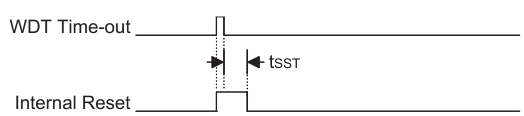


Note: t_{RSTD} is power-on delay, typical time=100ms

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 ERC or HIRC.
The t_{SST} is 1024 clock for HXT or LXT. 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	$\overline{\text{RES}}$ or LVR reset during NORMAL or SLOW Mode operation
1	u	WDT time-out reset during NORMAL or SLOW Mode operation
1	1	WDT time-out reset during IDLE or SLEEP Mode operation

"u" stands for unchanged

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

Item	Condition After RESET
Program Counter	Reset to zero
Interrupts	All interrupts will be disabled
WDT	Clear after reset, WDT begins counting
Timer/Event Counter	Timer Counter will be turned off
Input/Output Ports	I/O ports will be setup as inputs, and AN0~AN7 is as A/D input pin.
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.

HT66F20-1

Register	Reset (Power-on)	$\overline{\text{RES}}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
MP0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
BP	---- ---0	---- ---0	---- ---0	---- ---u
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	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	---- --xx	---- --uu	---- --uu	---- --uu
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
MFI0	--00 --00	--00 --00	--00 --00	--uu --uu

Register	Reset (Power-on)	RES or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
MFI1	--00 --00	--00 --00	--00 --00	--uu --uu
MFI2	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
PB	--11 1111	--11 1111	--11 1111	--uu uuuu
PBC	--11 1111	--11 1111	--11 1111	--uu uuuu
PCPU	---- 0000	---- 0000	---- 0000	---- uuuu
PC	---- 1111	---- 1111	---- 1111	---- uuuu
PCC	---- 1111	---- 1111	---- 1111	---- uuuu
ADRL(ADREF=0)	xxxx ----	xxxx ----	xxxx ----	uuuu ----
ADRL(ADREF=1)	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
ADRH(ADREF=0)	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
ADRH(ADREF=1)	---- xxxx	---- xxxx	---- xxxx	---- uuuu
ADCR0	0110 -000	0110 -000	0110 -000	uuuu -uuu
ADCR1	00-0 -000	00-0 -000	00-0 -000	uu-u -uuu
ACERL	1111 1111	1111 1111	1111 1111	uuuu uuuu
CP0C	1000 0--1	1000 0--1	1000 0--1	uuuu u--u
CP1C	1000 0--1	1000 0--1	1000 0--1	uuuu u--u
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
TM0AH	---- --00	---- --00	---- --00	---- --uu
EEA	- - - x xxxx	- - - x xxxx	- - - x xxxx	- - - 0 0000
EED	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
EEC	---- 0000	---- 0000	---- 0000	---- uuuu
TMPC0	--01 ---1	--01 ---1	--01 ---1	--uu ---u
TM1C0	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1C1	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
SCOMC	0000 0000	0000 0000	0000 0000	uuuu uuuu

Note: "u" stands for unchanged
"x" stands for unknown
"-" stands for unimplemented

HT66F30-1 Register

Register	Reset (Power-on)	$\overline{RES}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
MP0	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
MP1	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
BP	---- ---0	---- ---0	---- ---0	---- ---u
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	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	---- -xxx	---- -uuu	---- -uuu	---- -uuu
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
MFI0	--00 --00	--00 --00	--00 --00	--uu --uu
MFI1	-000 -000	-000 -000	-000 -000	-uuu -uuu
MFI2	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
PB	--11 1111	--11 1111	--11 1111	--uu uuuu
PBC	--11 1111	--11 1111	--11 1111	--uu 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
ADRL(ADREF=0)	xxxx ----	xxxx ----	xxxx ----	uuuu ----
ADRL(ADREF=1)	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
ADRH(ADREF=0)	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
ADRH(ADREF=1)	---- xxxx	---- xxxx	---- xxxx	---- uuuu
ADCR0	0110 -000	0110 -000	0110 -000	uuuu -uuu
ADCR1	00-0 -000	00-0 -000	00-0 -000	uu-u -uuu
ACERL	1111 1111	1111 1111	1111 1111	uuuu uuuu
CP0C	1000 0--1	1000 0--1	1000 0--1	uuuu u--u
CP1C	1000 0--1	1000 0--1	1000 0--1	uuuu u--u
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

Register	Reset (Power-on)	$\overline{RES}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
TM0DH	---- --00	---- --00	---- --00	---- --uu
TM0AL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0AH	---- --00	---- --00	---- --00	---- --uu
EEA	--xx xxxx	--xx xxxx	--xx xxxx	--uu uuuu
EED	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
EEC	---- 0000	---- 0000	---- 0000	---- uuuu
TMPC0	1-01 --01	1-01 --01	1-01 --01	u-uu --uu
PRM0	---- -000	---- -000	---- -000	---- -uuu
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
SCOMC	0000 0000	0000 0000	0000 0000	uuuu uuuu

Note: " - " stands for not implement
" u " stands for unchanged
" x " stands for unknown

HT68F20-1

Register	Reset (Power-on)	$\overline{RES}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
MP0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
BP	---- ---0	---- ---0	---- ---0	---- ---u
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	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	---- --xx	---- --uu	---- --uu	---- --uu
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
MFI0	--00 --00	--00 --00	--00 --00	--uu --uu
MFI1	--00 --00	--00 --00	--00 --00	--uu --uu
MFI2	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

Register	Reset (Power-on)	$\overline{RES}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
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
PB	--11 1111	--11 1111	--11 1111	--uu uuuu
PBC	--11 1111	--11 1111	--11 1111	--uu uuuu
PCPU	---- 0000	---- 0000	---- 0000	---- uuuu
PC	---- 1111	---- 1111	---- 1111	---- uuuu
PCC	---- 1111	---- 1111	---- 1111	---- uuuu
CP0C	1000 0--1	1000 0--1	1000 0--1	uuuu u--u
CP1C	1000 0--1	1000 0--1	1000 0--1	uuuu u--u
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
TM0AH	---- --00	---- --00	---- --00	---- --uu
EEA	---x xxxx	---x xxxx	---x xxxx	- - - 0 0000
EED	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
EEC	---- 0000	---- 0000	---- 0000	---- uuuu
TMPC0	--01 ---1	--01 ---1	--01 ---1	--uu ---u
TM1C0	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1C1	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
SCOMC	0000 0000	0000 0000	0000 0000	uuuu uuuu

Note: "u" stands for unchanged
"x" stands for unknown
"–" stands for unimplemented

HT68F30-1 Register

Register	Reset (Power-on)	$\overline{RES}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
MP0	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
MP1	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
BP	---- ---0	---- ---0	---- ---0	---- ---u
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	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	---- -xxx	---- -uuu	---- -uuu	---- -uuu
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
MFI0	--00 --00	--00 --00	--00 --00	--uu --uu
MFI1	-000 -000	-000 -000	-000 -000	-uuu -uuu
MFI2	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
PB	--11 1111	--11 1111	--11 1111	--uu uuuu
PBC	--11 1111	--11 1111	--11 1111	--uu 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
CP0C	1000 0--1	1000 0--1	1000 0--1	uuuu u--u
CP1C	1000 0--1	1000 0--1	1000 0--1	uuuu u--u
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
TM0AH	---- --00	---- --00	---- --00	---- --uu
EEA	--xx xxxx	--xx xxxx	--xx xxxx	--uu uuuu
EED	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
EEC	---- 0000	---- 0000	---- 0000	---- uuuu
TMPC0	1-01 --01	1-01 --01	1-01 --01	u-uu --uu

Register	Reset (Power-on)	$\overline{RES}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
PRM0	- - - - - 0 0 0	- - - - - 0 0 0	- - - - - 0 0 0	- - - - - u u u
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	- - - - - - 0 0	- - - - - - 0 0	- - - - - - 0 0	- - - - - - u u
TM1AL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1AH	- - - - - - 0 0	- - - - - - 0 0	- - - - - - 0 0	- - - - - - u u
TM1BL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1BH	- - - - - - 0 0	- - - - - - 0 0	- - - - - - 0 0	- - - - - - u u
SCOMC	0000 0000	0000 0000	0000 0000	uuuu uuuu

Note: “ - ” stands for not implement
“ u ” stands for unchanged
“ x ” stands for unknown

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.

These devices provide bidirectional input/output lines labeled with port names PA~PC 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 Port Register List

HT66F20-1/HT68F20-1 Register

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAWU	PAWU7	PAWU 6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PBPU	—	—	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PB	—	—	PB5	PB4	PB3	PB2	PB1	PB0
PBC	—	—	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PCPU	—	—	—	—	PCPU3	PCPU2	PCPU1	PCPU0
PC	—	—	—	—	PC3	PC2	PC1	PC0
PCC	—	—	—	—	PCC3	PCC2	PCC1	PCC0

HT66F30-1/HT68F30-1 Register

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAWU	PAWU7	PAWU 6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PBPU	—	—	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PB	—	—	PB5	PB4	PB3	PB2	PB1	PB0
PBC	—	—	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PCPU	PCPU7	PCPU6	PCPU5	PCPU4	PCPU3	PCPU2	PCPU1	PCPU0
PC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
PCC	PCC7	PCC6	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0

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 registers, namely PAPU~PCPU, and are implemented using weak PMOS transistors.

PAPU Register

Bit	7	6	5	4	3	2	1	0
Name	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
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

PBPU Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

PCPU Register

- HT66F20-1/HT68F20-1

Bit	7	6	5	4	3	2	1	0
Name	PCPU7	PCPU6	PCPU5	PCPU4	PCPU3	PCPU2	PCPU1	PCPU0
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

- HT66F30-1/HT68F30-1

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PCPU3	PCPU2	PCPU1	PCPU0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~6 "—" Unimplemented, read as "0"

PAPUn/PBPUn/PCPUn: Pull-high function 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	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **PAWU7~PAWU0:** Port A wake-up control
0: Disable
1: Enable

I/O Port Control Registers

Each I/O port has its own control register known as PAC~PCC, to control the input/output configuration. With this control register, each CMOS output or input can be reconfigured dynamically under software control. Each pin of the I/O ports is directly mapped to a bit in its associated port control register. For the I/O pin to function as an input, the corresponding bit of the control register must be written as a "1". This will then allow the logic state of the input pin to be directly read by instructions. When the corresponding bit of the control register is written as a "0", the I/O pin will be setup as a CMOS output. If the pin is currently setup as an output, instructions can still be used to read the output register. However, it should be noted that the program will in fact only read the status of the output data latch and not the actual logic status of the output pin.

PAC Register

Bit	7	6	5	4	3	2	1	0
Name	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
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

PBC Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	1	1	1	1	1	1

PCC Register

• HT66F20-1/HT68F20-1

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	PCC3	PCC2	PCC1	PCC0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	1	1	1	1

• HT66F30-1/HT68F30-1

Bit	7	6	5	4	3	2	1	0
Name	PCC7	PCC6	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0
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 7~6 "—": Unimplemented, read as "0"

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

Pin-remapping 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 is a PRM0 register to establish certain pin functions. This pin-remapping function is only available for the HT66F30-1 and HT68F30-1 devices.

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 devices include a PRM0 register which can select the functions of certain pins. The HT66F30-1 and HT68F30-1 devices include a PRM0 register which can select the functions of certain pins.

PRM0 Register – HT66F30-1/HT68F30-1

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	PCPRM	SIMPS0	PCKPS
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	0	0	0

Bit 7~3 Unimplemented, read as "0"

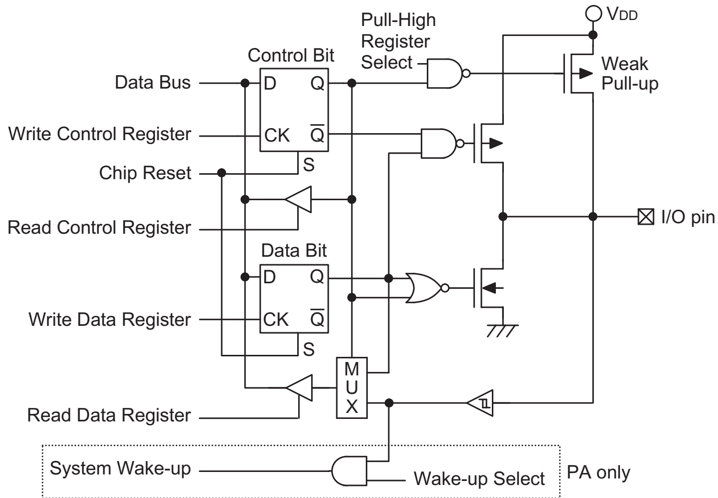
Bit 2 **PCPRM:** PC1~PC0 pin-shared function Pin Remapping Control
0: No change
1: TP1B_0 on PC0 change to PA6, TP1B_1 on PC1 change to PA7 if SIMPS0=1

Bit 1 **SIMPS0:** SIM Pin Remapping Control
0: SDO on PA5; SDI/SDA on PA6; SCK/SCL on PA7; $\overline{\text{SCS}}$ on PB5
1: SDO on PC1; SDI/SDA on PC0; SCK/SCL on PC7; $\overline{\text{SCS}}$ on PC6

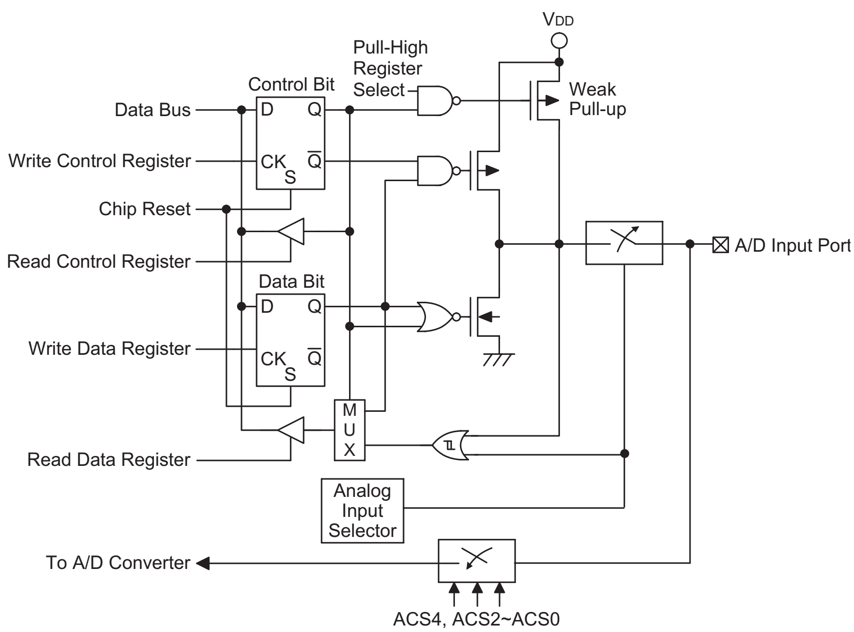
Bit 0 **PCKPS:** PCK and $\overline{\text{PINT}}$ Pin Remapping Control
0: PCK on PC2; $\overline{\text{PINT}}$ on PC3
1: PCK on PC5; $\overline{\text{PINT}}$ on PC4

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



A/D Input/Output Structure

Programming Considerations

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

Port A has the additional capability of providing wake-up 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 devices 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 multiple interrupts. The addition of input and output pins for each TM ensures that users are provided with timing units with a wide and flexible range of features.

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

Introduction

The devices contain two TMs having a reference name of TM0 and TM1. 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.

TM 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 Compact Type, Standard Type and Enhanced Type TM which are shown in the table together with their individual reference name, TM0, TM1.

Device	TM0	TM1
HT66F20-1/HT68F20-1	10-bit CTM	10-bit STM
HT66F30-1/HT68F30-1	10-bit CTM	10-bit ETM

TM Name/Type Reference

TM Operation

The different types of TM offer a diverse range of functions, from simple timing operations to PWM signal generation. The key to understanding how the TM operates is to see it in terms of a free running 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 TMn 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_L clock source or the external TCKn pin. Note that setting these bits to the value 101 will select an undefined 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 has 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 devices are different, the details are provided in the accompanying table.

All TM output pin names have a "_n" suffix. Pin names that include a "_0" or "_1" 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
HT66F20-1 HT68F20-1	TCK0 TP0_0	TCK1 TP1_0, TP1_1	—	TMPC0
HT66F30-1 HT68F30-1	TCK0 TP0_0, TP0_1	—	TCK1 TP1A, TP1B_0, TP1B_1	TMPC0

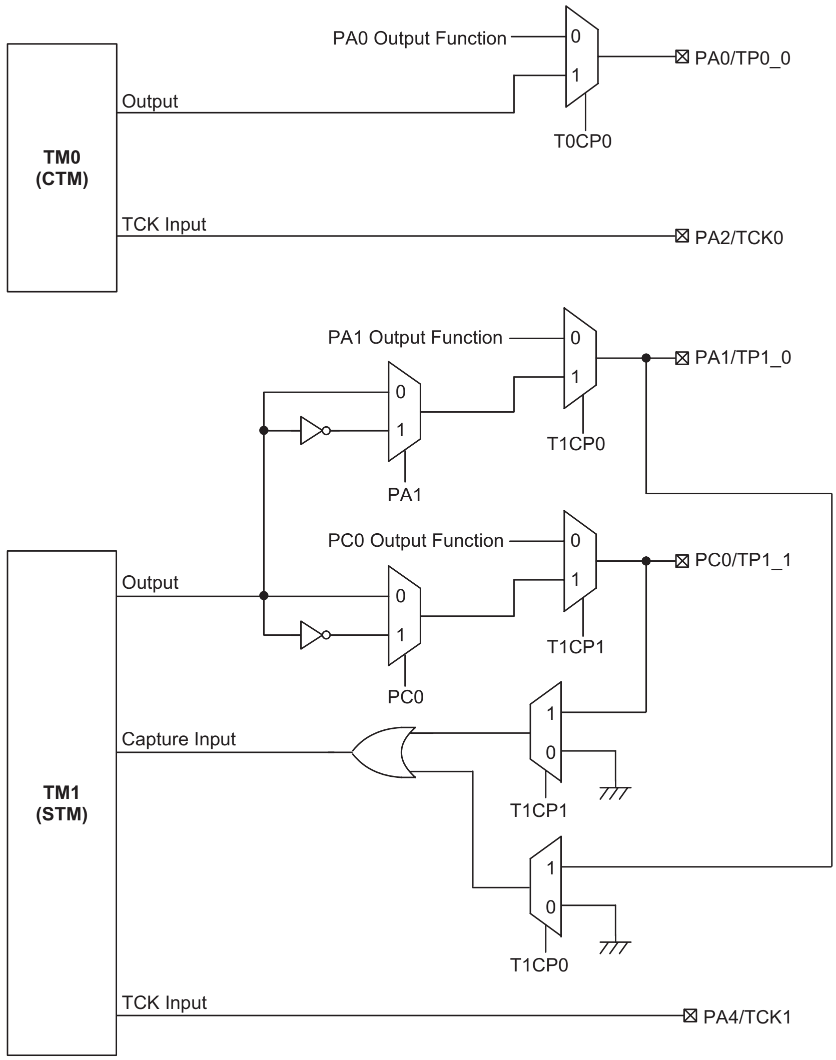
TM Input/Output Pins

TM Input/Output Pin Control

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.

Device	Bit							
	7	6	5	4	3	2	1	0
HT66F20-1 HT68F20-1	—	—	T1CP1	T1CP0	—	—	—	T0CP0
HT66F30-1 HT68F30-1	T1ACP0	—	T1BCP1	T1BCP1	—	—	T0CP1	T0CP0

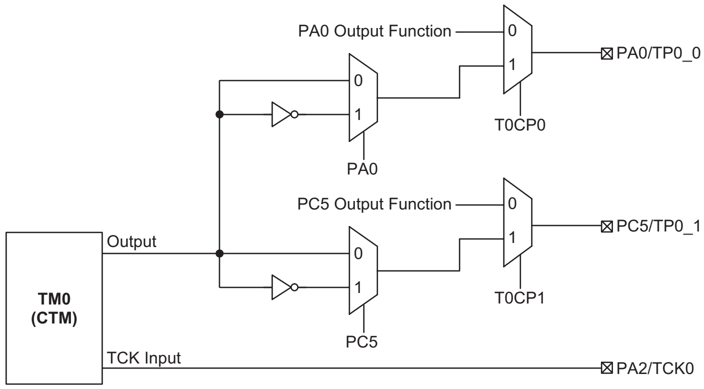
TM Input/Output Pin Control Registers List



HT66F20-1/HT68F20-1 TM Function Pin Control Block Diagram

Note: 1. The I/O register data bits shown are used for TM output inversion control.

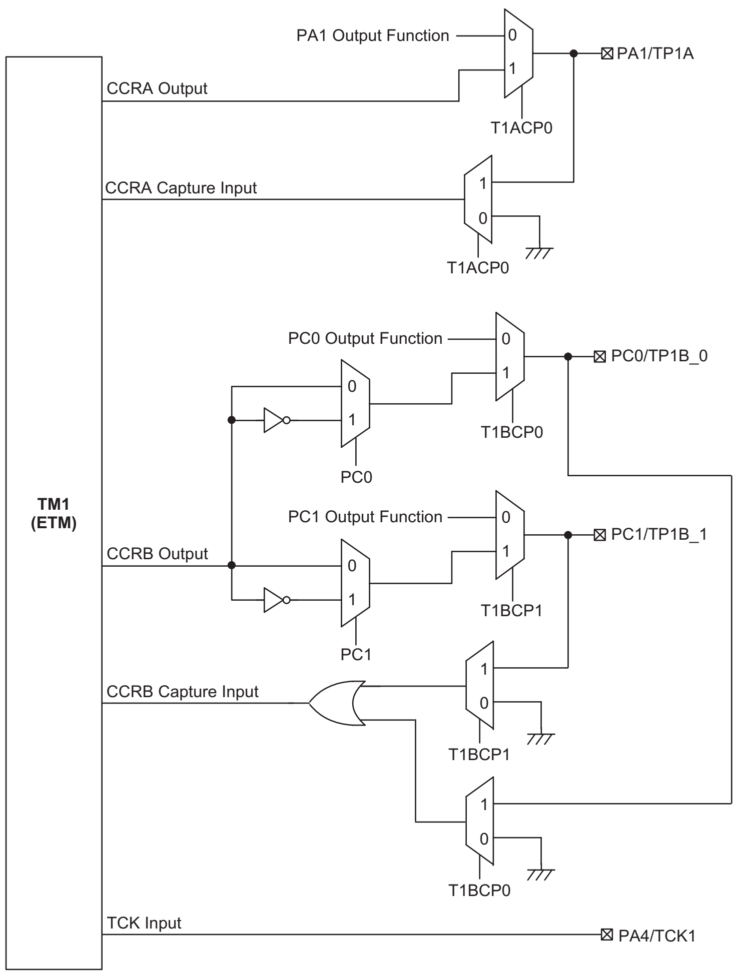
2. In the Capture Input Mode, the TM pin control register must never enable more than one TM input.



HT66F30-1/HT68F30-1 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.



HT66F30-1/HT68F30-1 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.

TMPC0 Register

- **HT66F20-1/HT68F20-1**

Bit	7	6	5	4	3	2	1	0
Name	—	—	T1CP1	T1CP1	—	—	—	T0CP0
R/W	—	—	R/W	R/W	—	—	—	R/W
POR	—	—	0	1	—	—	—	1

Bit 7~6 "—" Unimplemented, read as "0"

Bit 5 **T1CP1:** TP1_1 Pin Control
0: Disable
1: Enable

Bit 4 **T1CP0:** TP1_0 Pin Control
0: Disable
1: Enable

Bit 3~1 "—" Unimplemented, read as "0"

Bit 0 **T0CP0:** TP0_0 Pin Control
0: Disable
1: Enable

- **HT66F30-1/HT68F30-1**

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

Bit 7 **T1ACP0:** TP1A pin Control
0: Disable
1: Enable

Bit 6 Unimplemented, read as "0"

Bit 5 **T1BCP1:** TP1B_1 pin Control
0: Disable
1: Enable

Bit 4 **T1BCP0:** TP1B_0 pin Control
0: Disable
1: Enable

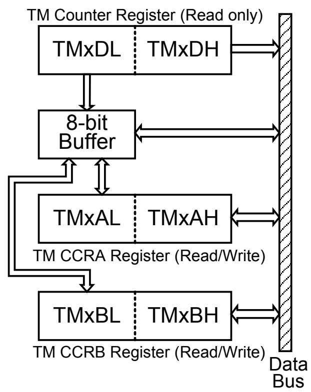
Bit 3~2 Unimplemented, read as "0"

Bit 1 **T0CP1:** TP0_1 pin Control
0: Disable
1: Enable

Bit 0 **T0CP0:** TP0_0 pin Control
0: Disable
1: Enable

Programming Considerations

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



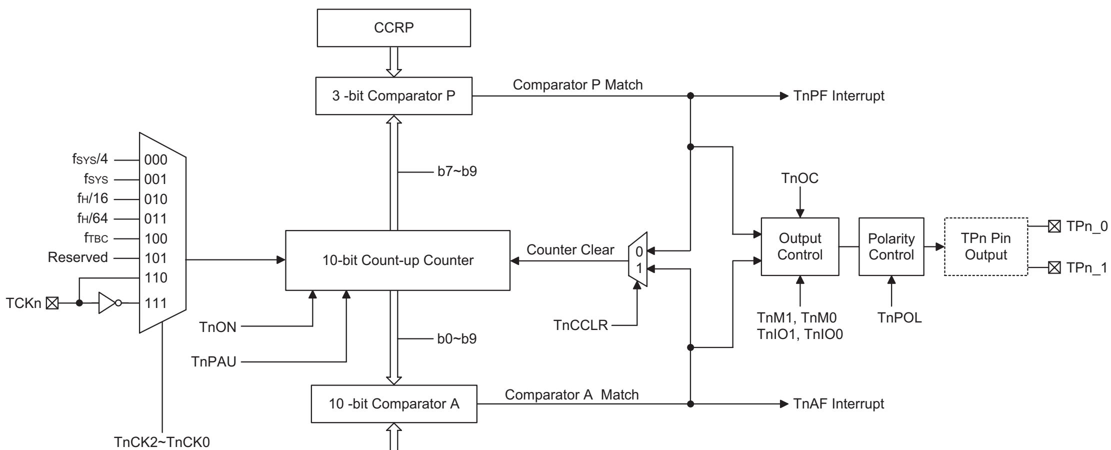
The following steps show the read and write procedures:

- 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 two 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. The two external output pins can be the same signal or the inverse signal.

Device	TM Type	TM Name	TM Input Pin	TM Output Pin
HT66F20-1/HT68F20-1	10-bit CTM	TM0	TCK0	TP0_0
HT66F30-1/HT68F30-1	10-bit CTM	TM0	TCK0	TP0_0, TP0_1



Compact Type TM Block Diagram (n=0)

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 T0ON 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 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
TM0C0	T0PAU	T0CK2	T0CK1	T0CK0	T0ON	T0RP2	T0RP1	T0RP0
TM0C1	T0M1	T0M0	T0IO1	T0IO0	T0OC	T0POL	T0DPX	T0CCLR
TM0DL	D7	D6	D5	D4	D3	D2	D1	D0
TM0DH	—	—	—	—	—	—	D9	D8
TM0AL	D7	D6	D5	D4	D3	D2	D1	D0
TM0AH	—	—	—	—	—	—	D9	D8

Compact TM Register List

TM0DL 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 **TM0DL:** TM0 Counter Low Byte Register bit 7~bit 0
TM0 10-bit Counter bit 7~bit 0

TM0DH 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 **TM0DH:** TM0 Counter High Byte Register bit 1~bit 0
TM0 10-bit Counter bit 9~bit 8

TM0AL 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 **TM0AL:** TM0 CCRA Low Byte Register bit 7~bit 0
TM0 10-bit CCRA bit 7~bit 0

TM0AH 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 **TM0AH:** TM0 CCRA High Byte Register bit 1~bit 0
TM0 10-bit CCRA bit 9~bit 8

TM0C0 Register

Bit	7	6	5	4	3	2	1	0
Name	T0PAU	T0CK2	T0CK1	T0CK0	T0ON	T0RP2	T0RP1	T0RP0
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 **T0PAU:** TM0 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 **T0CK2~T0CK0:** Select TM0 Counter clock
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{TBC}
101: Undefined
110: TCK0 rising edge clock
111: TCK0 falling edge clock

These three bits are used to select the clock source for the TM0. 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 **T0ON:** TM0 Counter On/Off Control
0: Off
1: On

This bit controls the overall on/off function of the TM0. Setting the bit high enables the counter to run, clearing the bit disables the TM0. Clearing this bit to zero will stop the counter from counting and turn off the TM0 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 TM0 is in the Compare Match Output Mode then the TM0 output pin will be reset to its initial condition, as specified by the T0OC bit, when the T0ON bit changes from low to high.

Bit 2~0 **T0RP2~T0RP0:** TM0 CCRP 3-bit register, compared with the TM0 Counter bit 9~bit 7 Comparator P Match Period
000: 1024 TM0 clocks
001: 128 TM0 clocks
010: 256 TM0 clocks
011: 384 TM0 clocks
100: 512 TM0 clocks
101: 640 TM0 clocks
110: 768 TM0 clocks
111: 896 TM0 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 T0CCLR bit is set to zero. Setting the T0CCLR 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.

TM0C1 Register

Bit	7	6	5	4	3	2	1	0
Name	T0M1	T0M0	T0IO1	T0IO0	T0OC	T0POL	T0DPX	T0CCLR
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 **T0M1~T0M0:** Select TM0 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 T0M1 and T0M0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4 **T0IO1~T0IO0:** Select TP0_0, TP0_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 TM0 output pin changes state when a certain condition is reached. The function that these bits select depends upon in which mode the TM0 is running.

In the Compare Match Output Mode, the T0IO1 and T0IO0 bits determine how the TM0 output pin changes state when a compare match occurs from the Comparator A. The TM0 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 TM0 output pin should be setup using the T0OC bit in the TM0C1 register. Note that the output level requested by the T0IO1 and T0IO0 bits must be different from the initial value setup using the T0OC bit otherwise no change will occur on the TM0 output pin when a compare match occurs. After the TM0 output pin changes state it can be reset to its initial level by changing the level of the T0ON bit from low to high.

In the PWM Mode, the T0IO1 and T0IO0 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 only change the values of the T0IO1 and T0IO0 bits only after the TM0 has been switched off. Unpredictable PWM outputs will occur if the T0IO1 and T0IO0 bits are changed when the TM is running.

Bit 3 **T0OC:** TP0_0, TP0_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 TM0 output pin. Its operation depends upon whether TM0 is being used in the Compare Match Output Mode or in the PWM Mode. It has no effect if the TM0 is in the Timer/Counter Mode. In the Compare Match Output Mode it determines the logic level of he TM0 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 **T0POL:** TP0_0, TP0_1 Output polarity Control
0: Non-invert
1: Invert
This bit controls the polarity of the TP0_0 or TP0_1 output pin. When the bit is set high the TM0 output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM0 is in the Timer/Counter Mode.

Bit 1 **T0DPX:** TM0 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 **T0CCLR:** Select TM0 Counter clear condition
0: TM0 Comparator P match
1: TM0 Comparator A match
This bit is used to select the method which clears the counter. Remember that the Compact TM0 contains two comparators, Comparator A and Comparator P, either of which can be selected to clear the internal counter. With the T0CCLR 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 T0CCLR bit is not used in the PWM Mode.

Compact Type TM Operating Modes

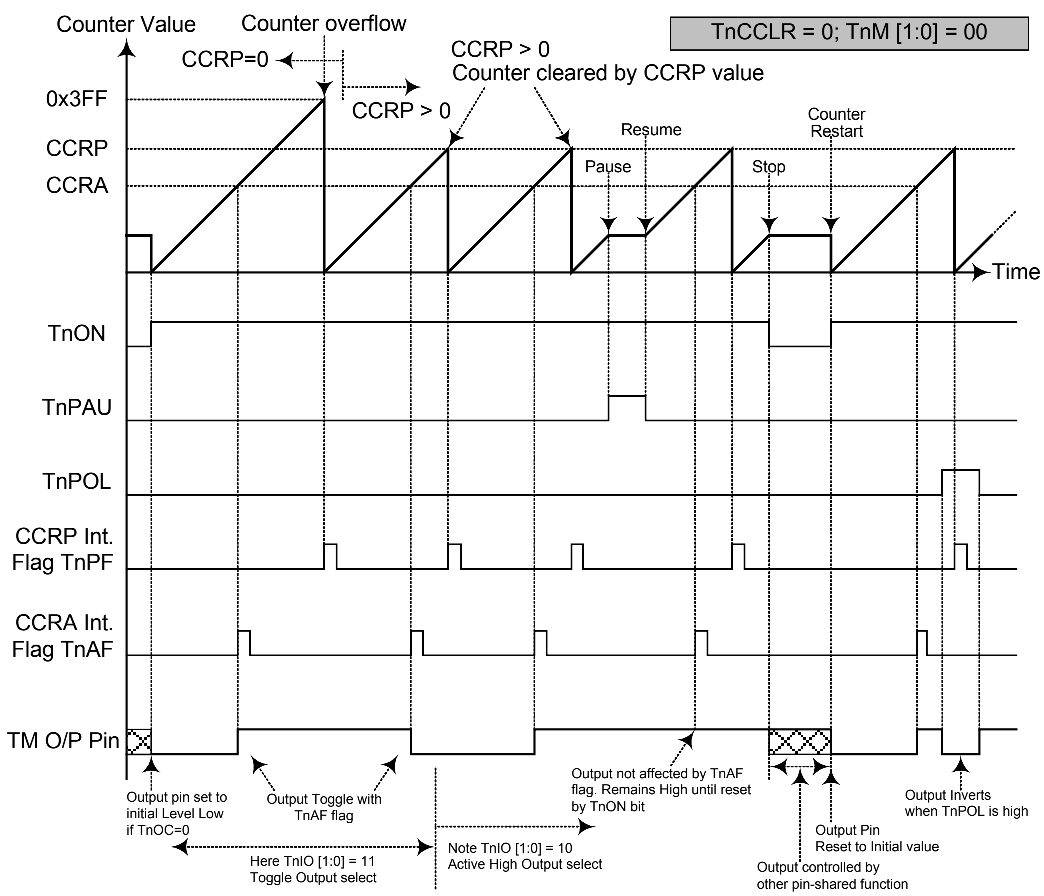
The Compact Type TM can operate in one of three operating modes, Compare Match Output Mode, PWM Mode or Timer/Counter Mode. The operating mode is selected using the T0M1 and T0M0 bits in the TM0C1 register.

Compare Match Output Mode

To select this mode, bits T0M1 and T0M0 in the TM0C1 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 T0CCLR 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 T0AF and T0PF interrupt request flags for the Comparator A and Comparator P respectively, will both be generated.

If the T0CCLR bit in the TM0C1 register is high then the counter will be cleared when a compare match occurs from Comparator A. However, here only the T0AF interrupt request flag will be generated even if the value of the CCRP bits is less than that of the CCRA registers. Therefore when T0CCLR is high no T0PF 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 T0AF 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 a T0AF interrupt request flag is generated after a compare match occurs from Comparator A. The T0PF 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 T0IO1 and T0IO0 bits in the TM0C1 register. The TM output pin can be selected using the T0IO1 and T0IO0 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 T0ON bit changes from low to high, is setup using the T0OC bit. Note that if the T0IO1 and T0IO0 bits are zero then no pin change will take place.



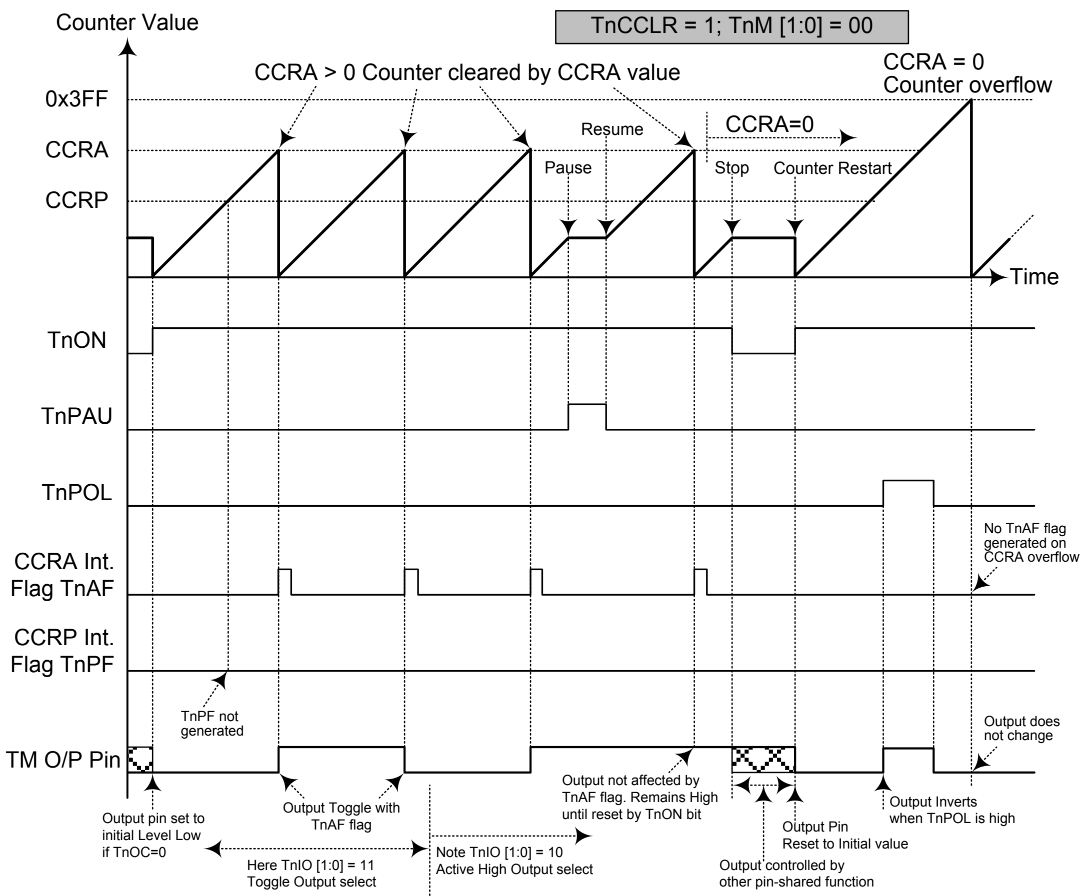
Compare Match Output Mode – TnCCLR=0

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

4. n=0



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
5. n=0

Timer/Counter Mode

To select this mode, bits T0M1 and T0M0 in the TM0C1 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 T0M1 and T0M0 in the TM0C1 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 T0CCLR 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 T0DPX bit in the TM0C1 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 T0OC bit in the TM0C1 register is used to select the required polarity of the PWM waveform while the two T0IO1 and T0IO0 bits are used to enable the PWM output or to force the TM output pin to a fixed high or low level. The T0POL bit is used to reverse the polarity of the PWM output waveform.

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,

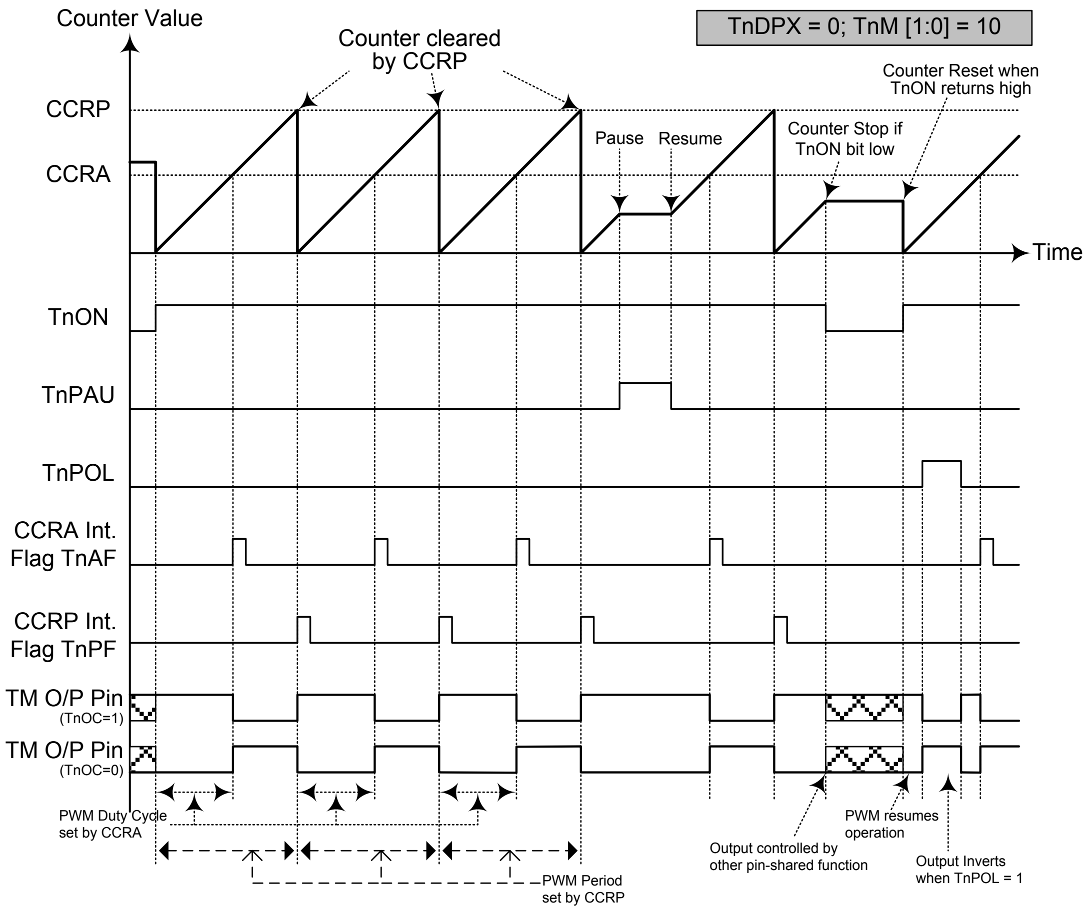
The CTM PWM output frequency=$(f_{SYS}/4)/512=f_{SYS}/2048$=7.8125 kHz, duty=128/512=25%.

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 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

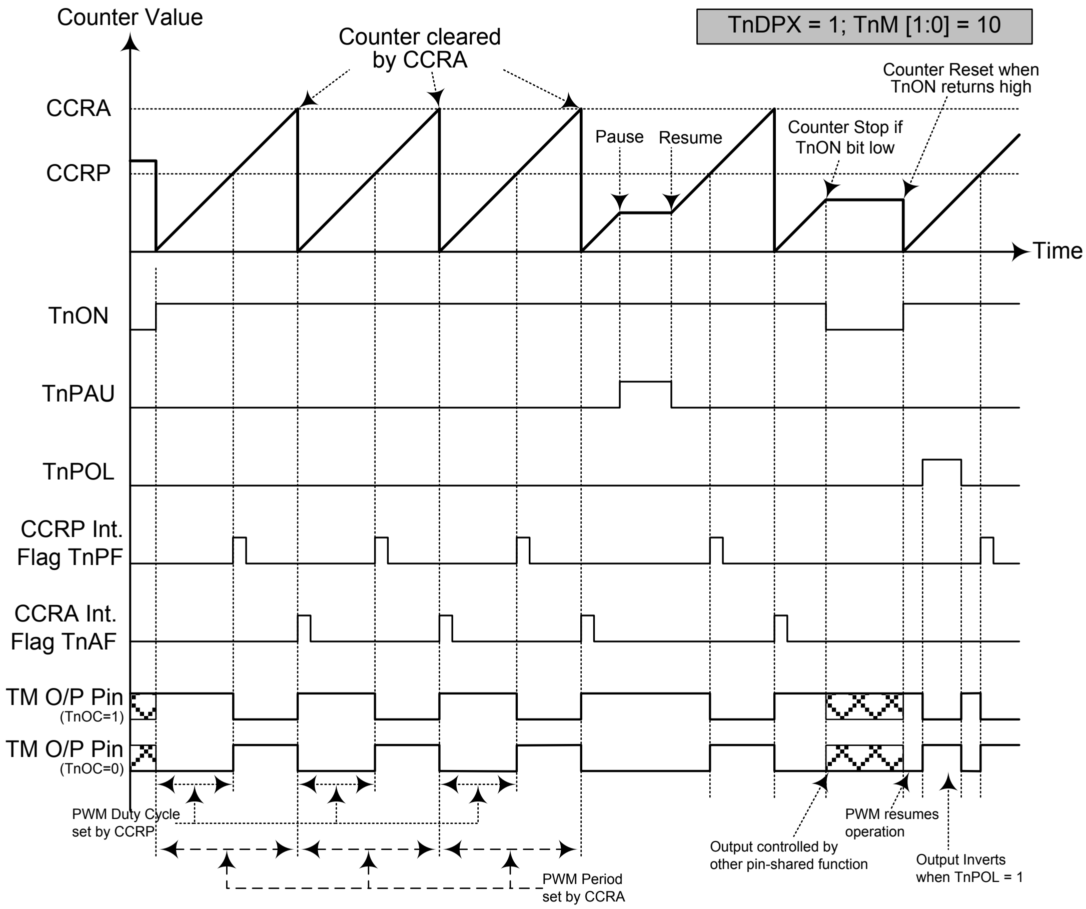
Note: 1. Here TnDPX=0 – Counter cleared by CCRP

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

5. n=0



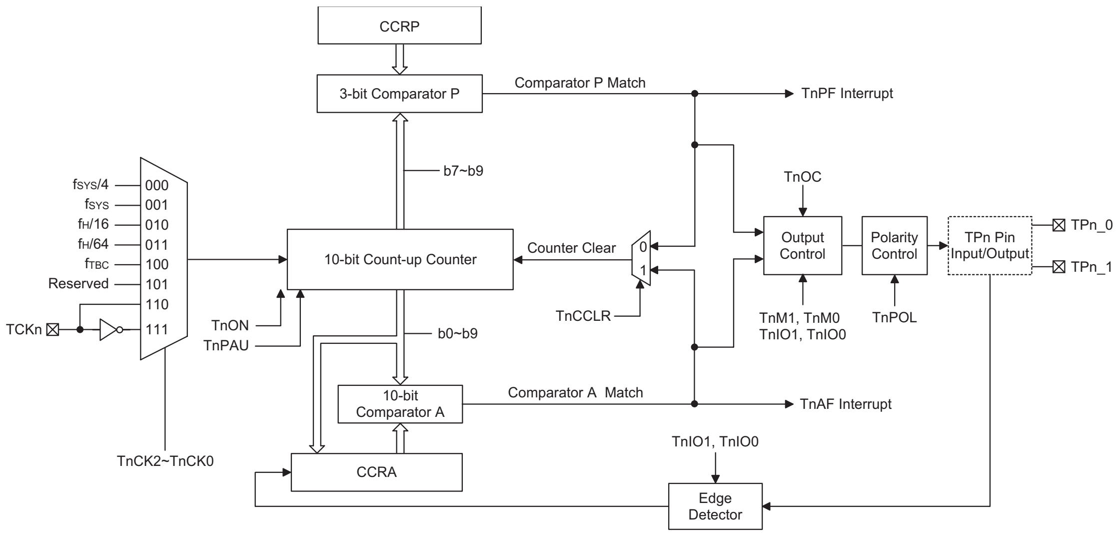
PWM Mode – TnDPX=1

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
5. n=0

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 two external output pins. The Standard Type TM is only contained in the HT66F20-1 and HT68F20-1 devices.

Device	TM Type	TM Name	TM Input Pin	TM Output Pin
HT66F20-1/HT68F20-1	10-bit STM	TM1	TCK1	TP1_0, TP1_1
HT66F30-1/HT68F30-1	—	—	—	—



HT66F20-1/HT68F20-1 Standard Type TM Block Diagram (n=1)

Standard TM Operation

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

The only way of changing the value of the 10-bit counter using the application program, is to clear the counter by changing the 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 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.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1C0	T1PAU	T1CK2	T1CK1	T1CK0	T1ON	T1RP2	T1RP1	T1RP0
TM1C1	T1M1	T1M0	T1IO1	T1IO0	T1OC	T1POL	T1DPX	T1CCLR
TM1DL	D7	D6	D5	D4	D3	D2	D1	D0
TM1DH	D15	D14	D13	D12	D11	D10	D9	D8
TM1AL	D7	D6	D5	D4	D3	D2	D1	D0
TM1AH	D15	D14	D13	D12	D11	D10	D9	D8

10-bit Standard TM Register List – HT66F20-1/HT68F20-1

TM1C0 Register

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, T1CK1, 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: 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 **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 CCRP 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

Bit	7	6	5	4	3	2	1	0
Name	T1M1	T1M0	T1IO1	T1IO0	T1OC	T1POL	T1DPX	T1CCLR
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 **T1M1~T1M0:** Select TM1 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 T1M1 and T1M0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4 **T1IO1~T1IO0:** Select TP1_0, TP1_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 TP1_0, TP1_1
01: Input capture at falling edge of TP1_0, TP1_1
10: Input capture at falling/rising edge of TP1_0, TP1_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 T1IO1 and T1IO0 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 T1OC bit in the TM1C1 register. Note that the output level requested by the T1IO1 and T1IO0 bits must be different from the initial value setup using the T1OC 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 T1IO1 and T1IO0 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 T1IO1 and T1IO0 bits only after the TM has been switched off. Unpredictable PWM outputs will occur if the T1IO1 and T1IO0 bits are changed when the TM is running.

Bit 3 **T1OC:** TP1_0, TP1_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 **TnPOL:** TP1_0, TP1_1 Output polarity Control
0: Non-invert
1: Invert

This bit controls the polarity of the TP1_0 or TP1_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 **T1DPX:** 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 **T1CCLR:** Select TM1 Counter clear condition
0: TM1 Comparatror P match
1: TM1 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 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.

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 **TM1DL:** TM1 Counter Low Byte Register bit 7~bit 0
TM1 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 **TM1DH:** TM1 Counter High Byte Register bit 1~bit 0
TM1 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 **TM1AL:** TM1 CCRA Low Byte Register bit 7~bit 0
TM1 10-bit CCRA bit 7~bit 0

TMnAH Register

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

Bit 7~2 Unimplemented, read as "0"

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

Standard Type TM Operating Modes

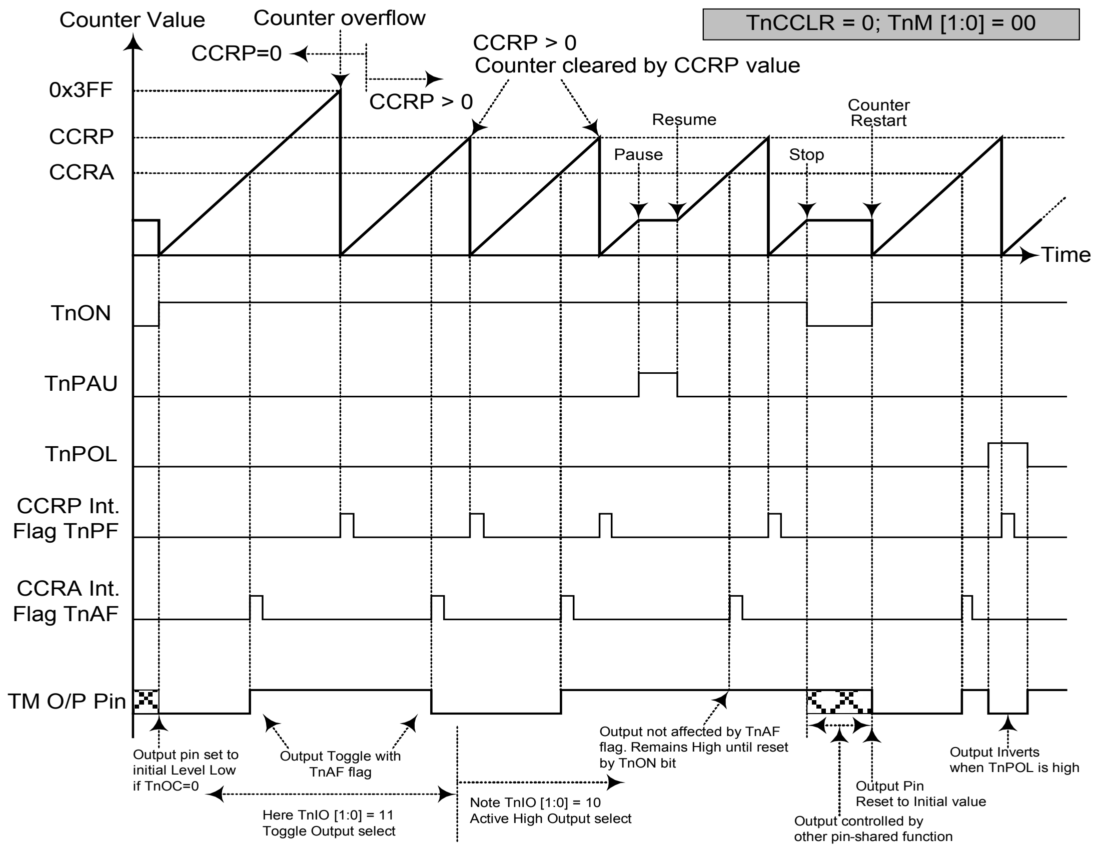
The Standard Type TM can operate in one of five operating modes, Compare Match Output Mode, PWM Output Mode, Single Pulse Output Mode, Capture Input Mode or Timer/Counter Mode. The operating mode is selected using the T1M1 and T1M0 bits in the TM1C1 register.

Compare Match Output Mode

To select this mode, bits T1M1 and T1M0 in the TM1C1 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 T1CCLR 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 T1AF and T1PF interrupt request flags for Comparator A and Comparator P respectively, will both be generated.

If the T1CCLR bit in the TM1C1 register is high then the counter will be cleared when a compare match occurs from Comparator A. However, here only the T1AF interrupt request flag will be generated even if the value of the CCRP bits is less than that of the CCRA registers. Therefore when T1CCLR is high no T1PF interrupt request flag will be generated. In the Compare Match Output Mode, the CCRA can not be set to "0".

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 T1AF interrupt request flag is generated after a compare match occurs from Comparator A. The T1PF 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 T1IO1 and T1IO0 bits in the TM1C1 register. The TM output pin can be selected using the T1IO1 and T1IO0 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 T1ON bit changes from low to high, is setup using the T1OC bit. Note that if the T1IO1 and T1IO0 bits are zero then no pin change will take place.



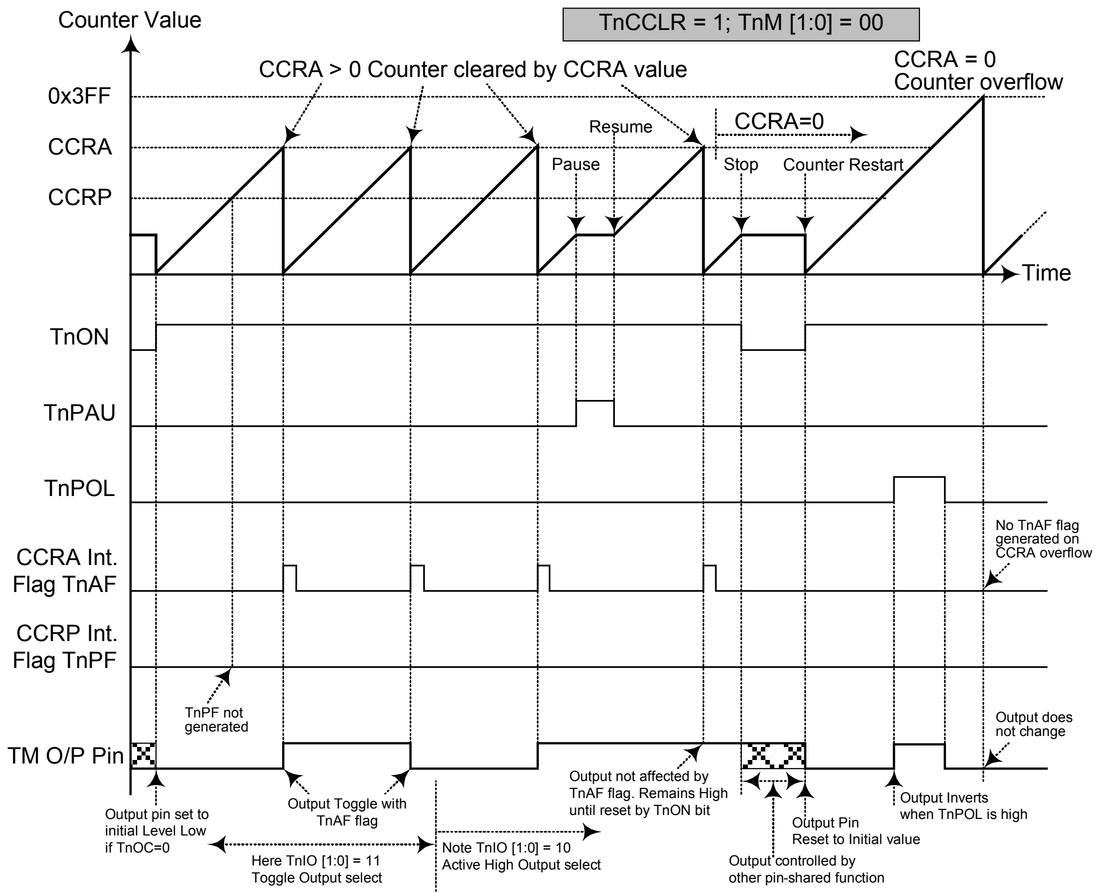
Compare Match Output Mode – TnCCLR=0

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

4. n=1



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
5. n=1

Timer/Counter Mode

To select this mode, bits T1M1 and T1M0 in the TM1C1 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 T1M1 and T1M0 in the TM1C1 register should be set to 10 respectively and also the T1IO1 and T1IO0 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 T1CCLR bit has no effect as the PWM period. Both of the CCRAand 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 T1DPX bit in the TM1C1 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 T1OC bit in the TM1C1 register is used to select the required polarity of the PWM waveform while the two T1IO1 and T1IO0 bits are used to enable the PWM output or to force the TM output pin to a fixed high or low level. The T1POL bit is used to reverse the polarity of the PWM output waveform.

10-bit STM, PWM Mode, Edge-aligned Mode, T1DPX=0

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

If f_{SYS}=12MHz, TM clock source select $f_{SYS}/4$, CCRP = 100b, CCRA = 128

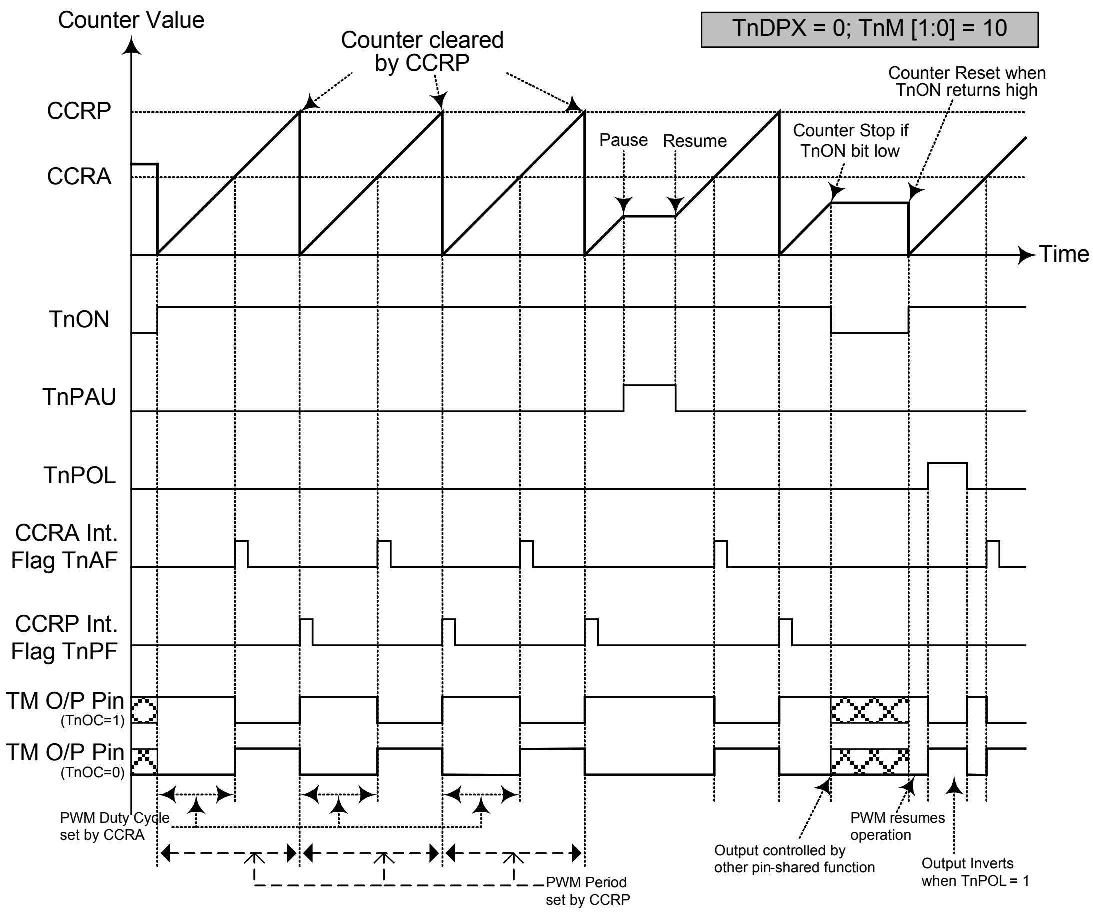
The STM PWM output frequency = $(f_{SYS}/4) / 512 = f_{SYS}/2048$ = 5.8594kHz, duty = 128/512 = 25%.

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%.

10-bit STM, PWM Mode, Edge-aligned Mode, T1DPX=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 CCRAregister value together with the TM clock while the PWM duty cycle is defined by the CCRP register value.



PWM Mode – TnDPX=0

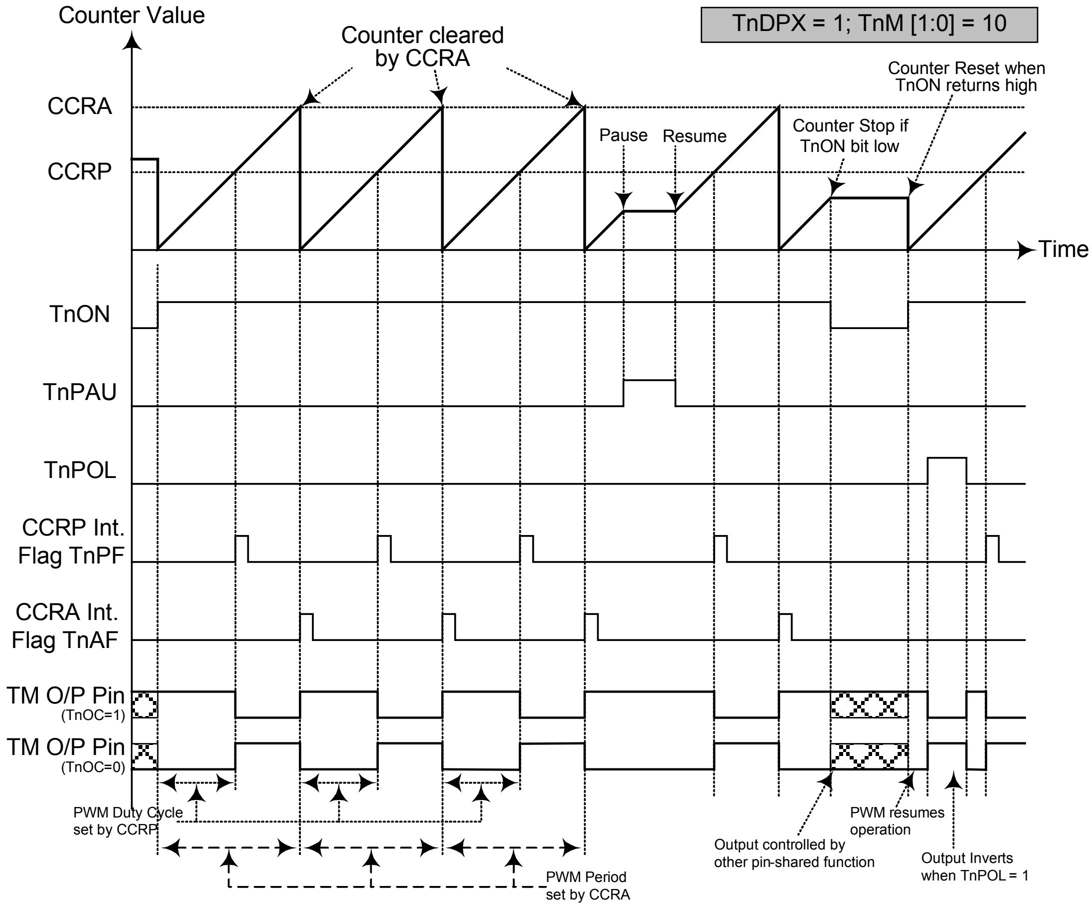
Note: 1. Here TnDPX=0, Counter cleared by CCRP

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

5. n=1



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 running even when TnIO [1:0]=00 or 01

4. The TnCCLR bit has no influence on PWM operation

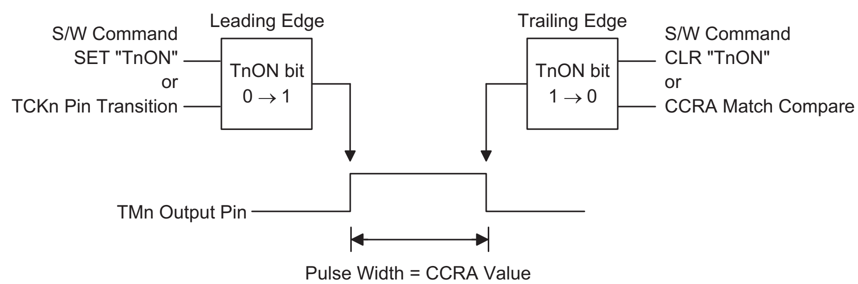
5. n=1

Single Pulse Mode

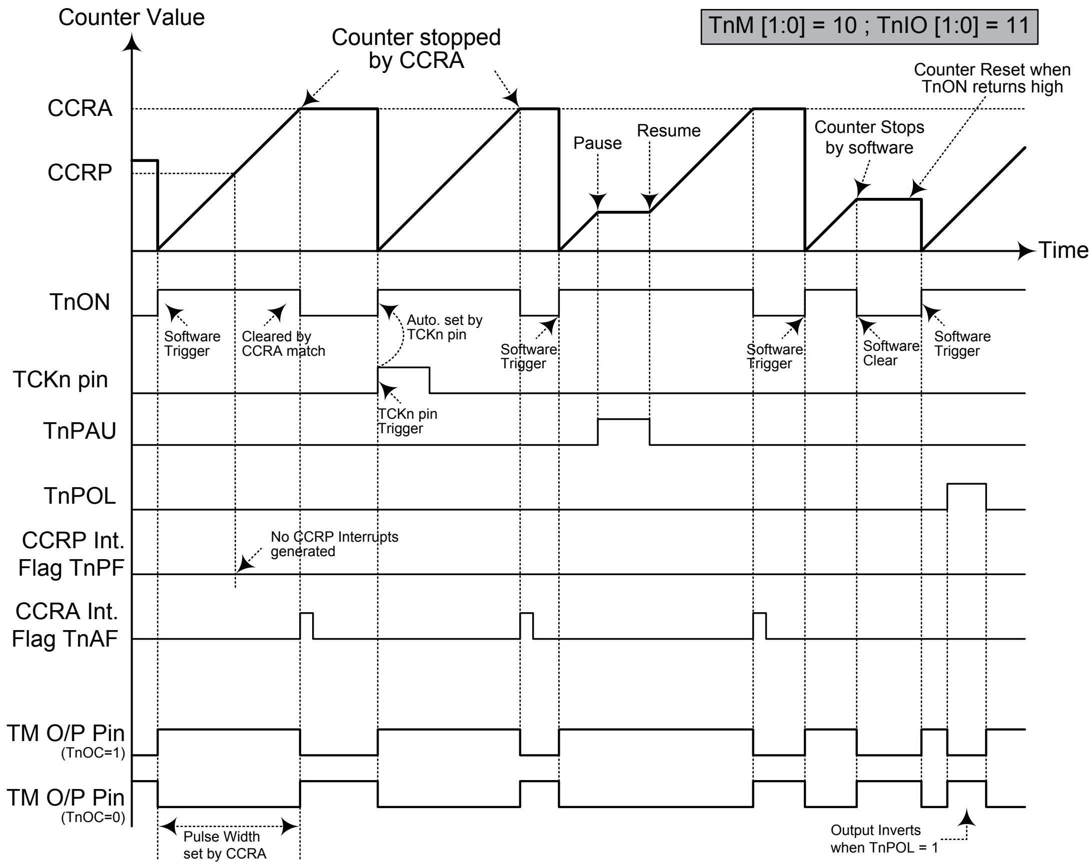
To select this mode, bits T1M1 and T1M0 in the TM1C1 register should be set to 10 respectively and also the T1IO1 and T1IO0 bits should be set to 11 respectively. The Single Pulse Output Mode, as the name suggests, will generate a single shot pulse on the TM output pin.

The trigger for the pulse output leading edge is a low to high transition of the T1ON bit, which can be implemented using the application program. However in the Single Pulse Mode, the T1ON bit can also be made to automatically change from low to high using the external TCK1 pin, which will in turn initiate the Single Pulse output. When the T1ON bit transitions to a high level, the counter will start running and the pulse leading edge will be generated. The T1ON bit should remain high when the pulse is in its active state. The generated pulse trailing edge will be generated when the T1ON 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 T1ON 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 T1ON bit changes from low to high when the counter restarts. In the Single Pulse Mode CCRP is not used. The T1CCLR and T1nDPX bits are not used in this Mode.



Single Pulse Generation



Single Pulse Mode

Note: 1. Counter stopped by CCRA

2. CCRP is not used

3. The pulse 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

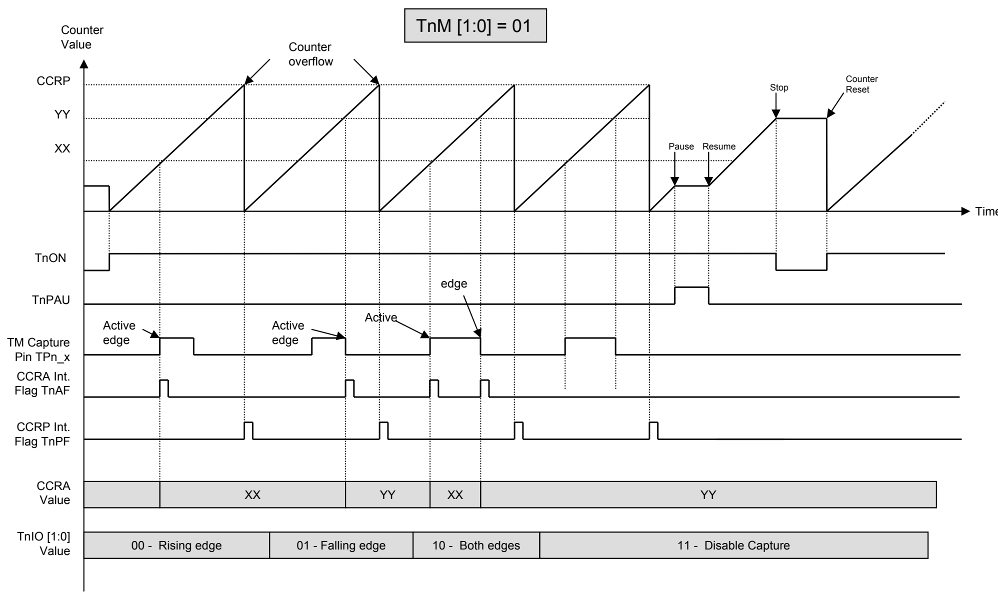
6. n=1

Capture Input Mode

To select this mode bits T1M1 and T1M0 in the TM1C1 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 TP1_0 or TP1_1 pin, whose active edge can be a rising edge, a falling edge or both rising and falling edges; the active edge transition type is selected using the T1IO1 and T1IO0 bits in the TM1C1 register. The counter is started when the T1ON bit changes from low to high which is initiated using the application program.

When the required edge transition appears on the TP1_0 or TP1_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 TP1_0 or TP1_1 pin the counter will continue to free run until the T1ON 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 T1IO1 and T1IO0 bits can select the active trigger edge on the TP1_0 or TP1_1 pin to be a rising edge, falling edge or both edge types. If the T1IO1 and T1IO0 bits are both set high, then no capture operation will take place irrespective of what happens on the TP1_0 or TP1_1 pin, however it must be noted that the counter will continue to run.

As the TP1_0 or TP1_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 T1CCLR and T1DPX bits are not used in this Mode.



Capture Input Mode

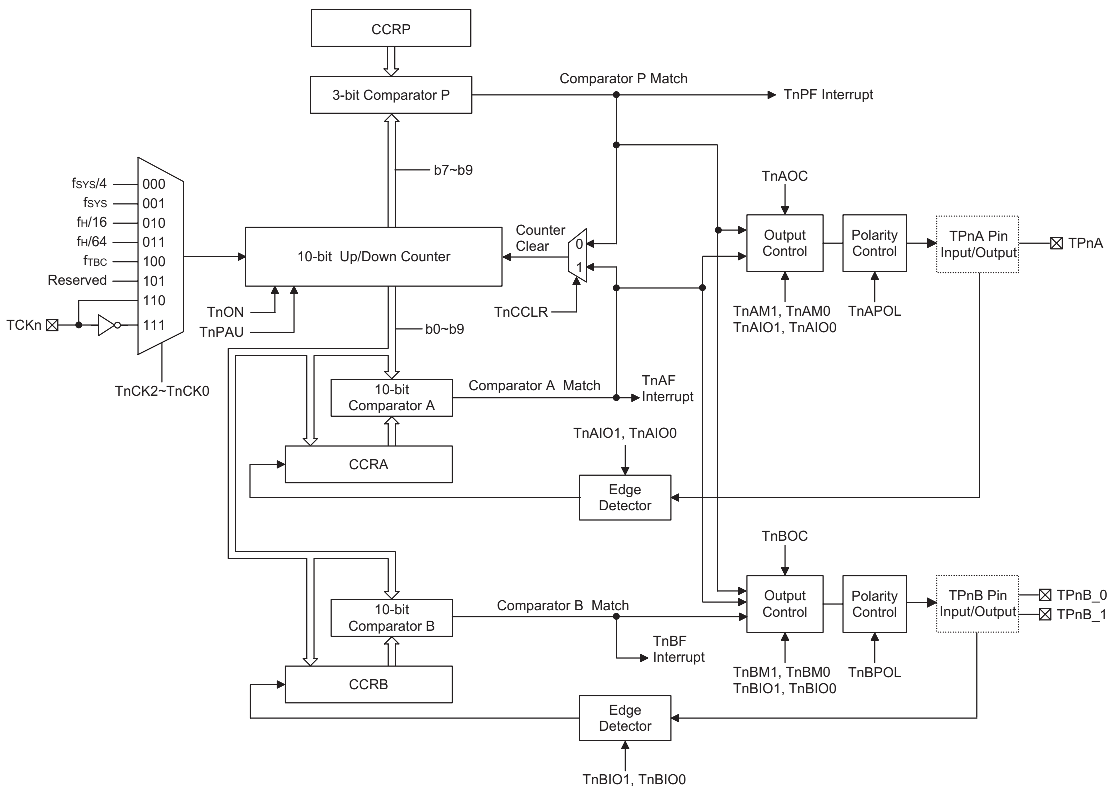
Note: 1. TnM [1:0]=01 and active edge set by the TnIO [1:0] bits

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.
6. n=1; x=0 or 1.

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 external output pins.

CTM	Name	TM No.	TM Input Pin	TM Output Pin
HT66F20-1/HT68F20-1	—	—	—	—
HT66F30-1/HT68F30-1	10-bit ETM	TM1	TCK1	TP1A; TP1B_0, TP1B_1



Enhanced Type TM Block Diagram (n=1)

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 T1ON 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 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	T1PAOL	T1CDN	T1CCLR
TM1C2	T1BM1	T1BM0	T1BIO1	T1BIO0	T1BOC	T1PBOL	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 – HT66F30-1/HT68F30-1

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: Reserved
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 and 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 CCRP 3-bit register, compared with the TM1 Counter bit 9~bit 7 Comparator P Match Period
000: 1024 TM1clocks
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
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 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
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 **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 Count up or down flag
0: Count up
1: Count down

Bit 0 **T1CCLR:** Select TM1 Counter clear condition
0: TM1 Comparatror P match
1: TM1 Comparatror 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 P can 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/W	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 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 TP1B_0, TP1B_1
01: Input capture at falling edge of TP1B_0, TP1B_1
10: Input capture at falling/rising edge of TP1B_0, TP1B_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 T1BIO1 and T1BIO0 bits determine how the TM output pin changes state when a compare match occurs from the Comparator B. 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 B. 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 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 **T1BPOL:** TP1B_0, TP1B_1 Output polarity Control
0: Non-invert
1: Invert
This bit controls the polarity of the TP1B_0, TP1B_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 1~0 **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 CCRA 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 CCRA 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 CCRB 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 CCRB 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 T1AM1 and T1AM0 bits in the TM1C1, and the T1BM1 and T1BM0 bits in the TM1C2 register.

ETM Operation 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	—	—	—	—	√

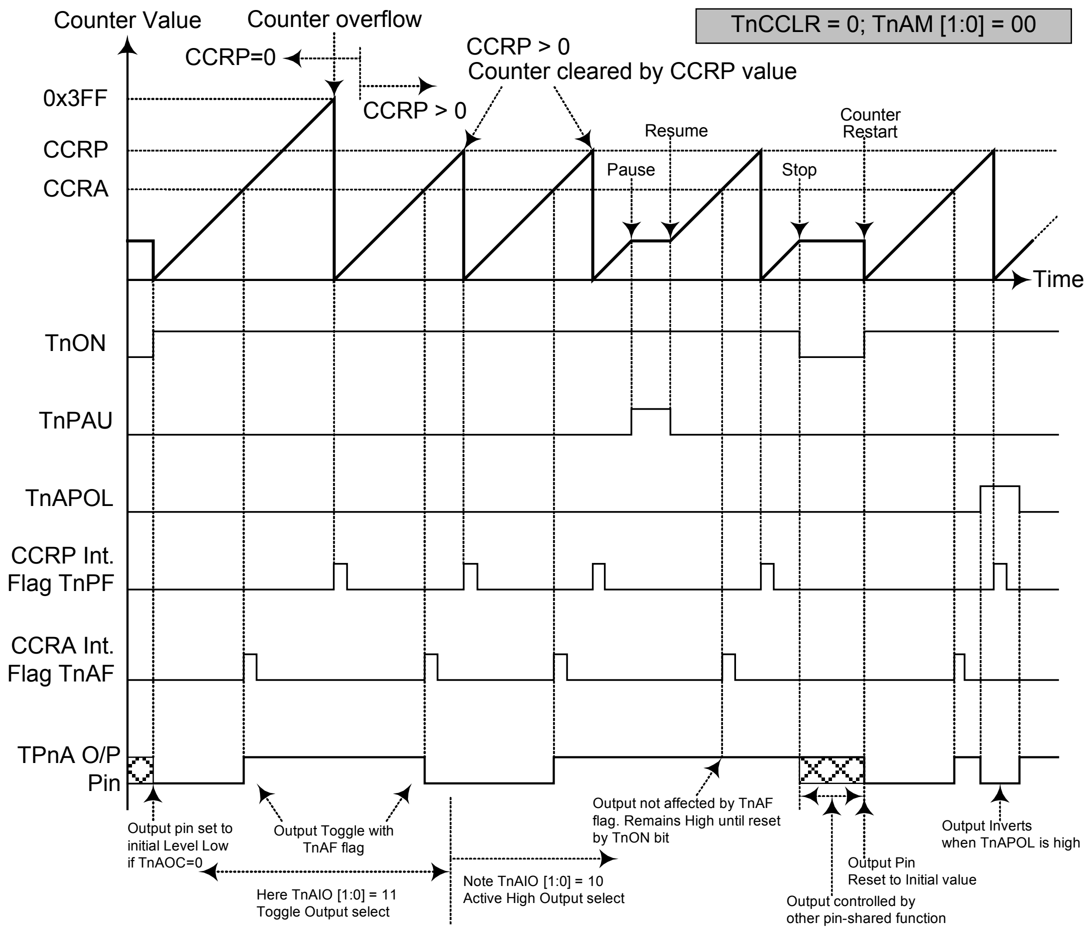
"√": permitted; "—": not permitted

Compare Match Output Mode

To select this mode, bits T1AM1, T1AM0 and T1BM1, T1BM0 in the TM1C1/TM1C2 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 T1CCLR 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 T1AF and T1PF interrupt request flags for Comparator A and Comparator P respectively, will both be generated.

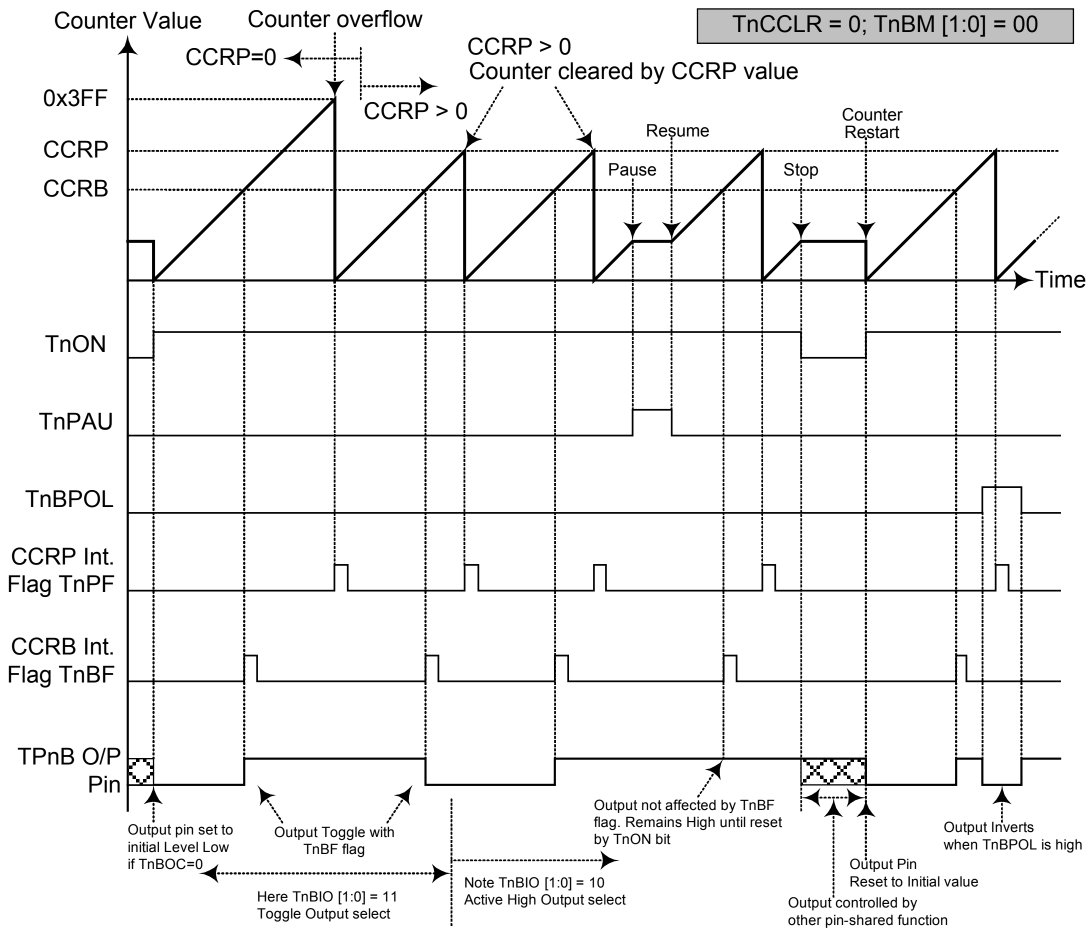
If the T1CCLR bit in the TM1C1 register is high then the counter will be cleared when a compare match occurs from Comparator A. However, here only the T1AF interrupt request flag will be generated even if the value of the CCRP bits is less than that of the CCRA registers. Therefore when T1CCLR is high no T1PF interrupt request flag will 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 a T1AF or T1BF interrupt request flag is generated after a compare match occurs from Comparator A or Comparator B. The T1PF 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 T1AIO1 and T1AIO0 bits in the TM1C1 register for ETM CCRA, and the T1BIO1 and T1BIO0 bits in the TM1C2 register for ETM CCRB. The TM output pin can be selected using the T1AIO1, T1AIO0 bits (for the TP1A pin) and T1BIO1, T1BIO0 bits (for the TP1B_0, TP1B_1 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, is setup after the T1AOC or T1BOC bit for TP1A or TP1B_0, TP1B_1 output pins. Note that if the T1AIO1, T1AIO0 and T1BIO1, T1BIO0 bits are zero then no pin change will take place.



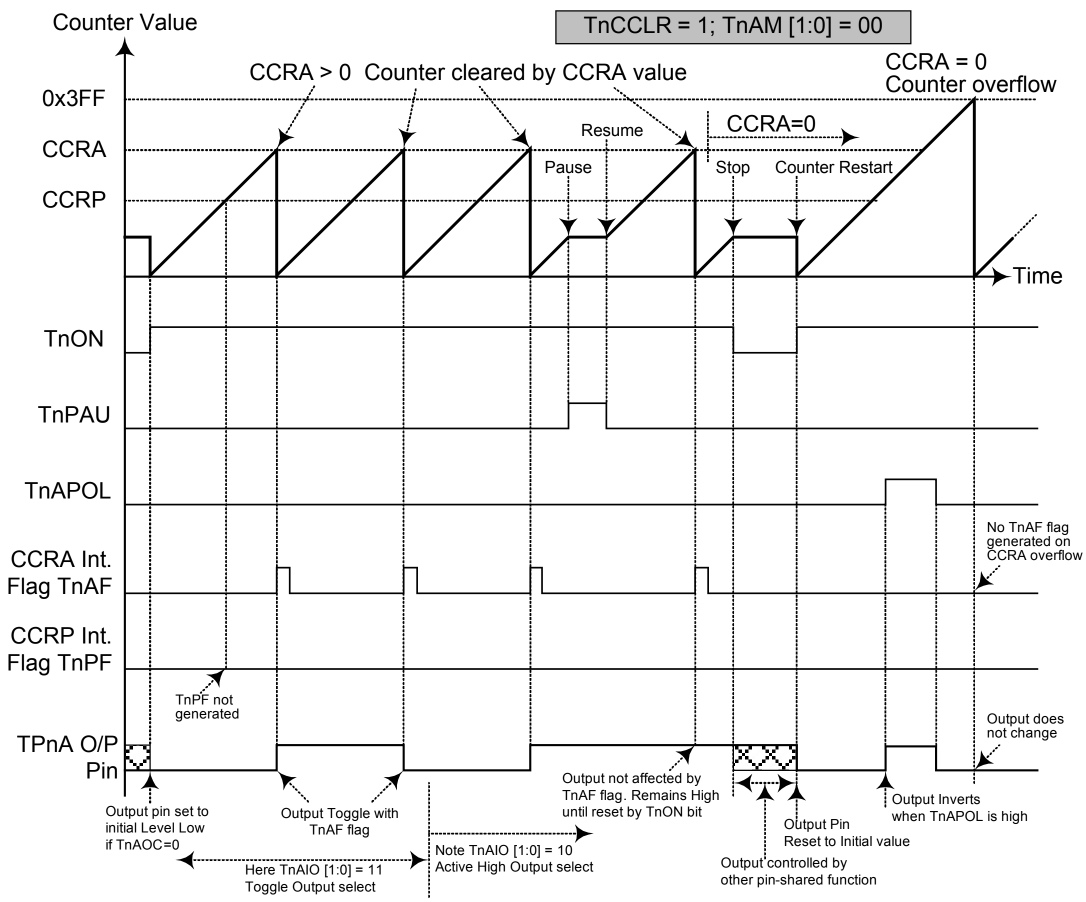
ETM CCRA Compare Match Output Mode – TnCCLR=0

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



ETM CCRB Compare Match Output Mode – TnCCLR=0

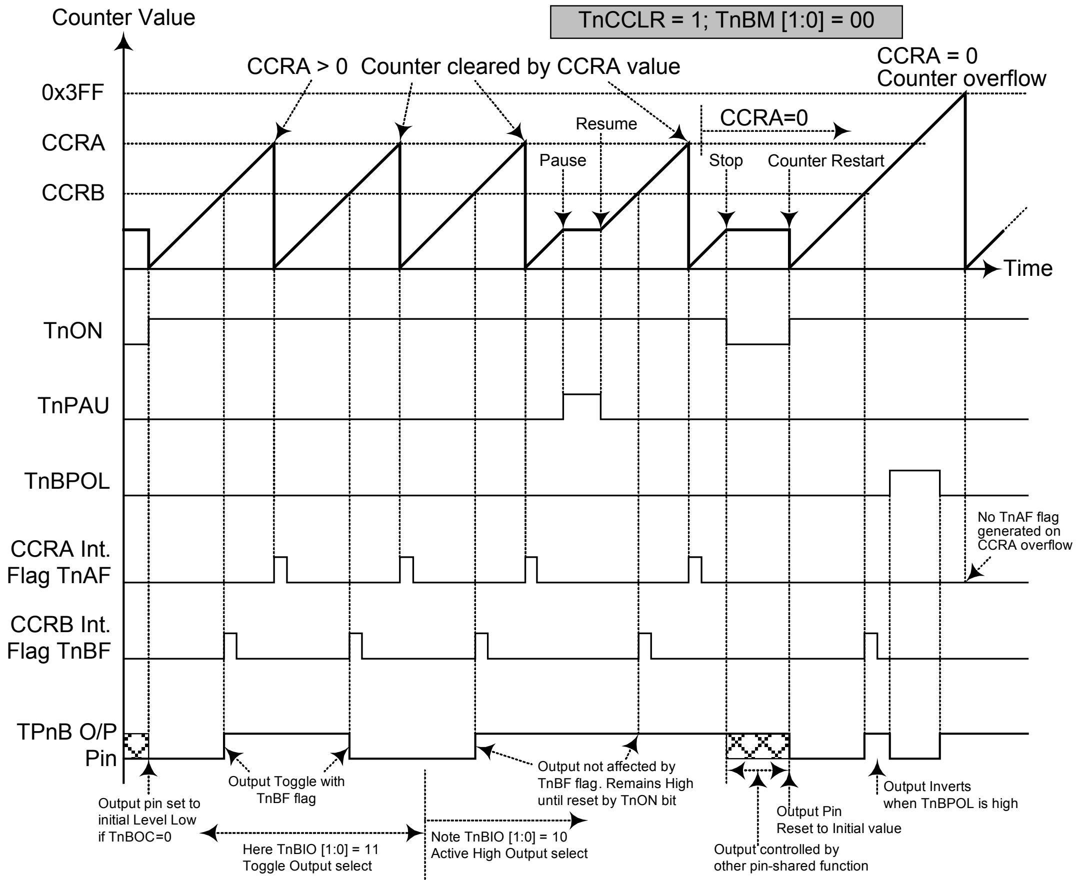
Note: 1. With TnCCLR=0 the Comparator P match will clear the counter
2. TPnB output pin controlled only by TnBF flag
3. Output pin reset to initial state by TnON bit rising edge
4. n=1



ETM CCRA Compare Match Output Mode – TnCCLR=1

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

2. TPnA output pin controlled only by TnAF flag
3. TPnA output pin reset to initial state by TnON rising edge
4. TnPF flags not generated when TnCCLR=1
5. n=1



ETM CCRB Compare Match Output Mode – TnCCLR=1

Note: 1. With TnCCLR=1 the Comparator A match will clear the counter
2. TPnB output pin controlled only by TnBF flag
3. TPnB output pin reset to initial state by TnON rising edge
4. TnPF flags not generated when TnCCLR=1
5. n=1

Timer/Counter Mode

To select this mode, bits T1AM1, T1AM0 and T1BM1, T1BM0 in the TM1C1 and TM1C2 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, T1AM1, T1AM0 and T1BM1, T1BM0 should be set to 10 respectively and also the T1AIO1, T1AIO0 and T1BIO1, T1BIO0 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 T1CCLR bit is used to determine in which way the PWM period is controlled. With the T1CCLR 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 TP1B_0 and TP1B_1 output pins). The CCRP bits are not used and TP1A output pin is not used. The PWM output can only be generated on the TP1B_0 and TP1B_1 output pins. With the T1CCLR 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 TP1A and TP1B_0/TP1B_1 pins.

The T1PWM1 and T1PWM0 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 T1AOC and T1BOC bits in the TM1C1 and TM1C2 register are used to select the required polarity of the PWM waveform while the two T1AIO1, T1AIO0 and T1BIO1, T1BIO0 bits pairs are used to enable the PWM output or to force the TM output pin to a fixed high or low level. The T1APOL and T1BPOL bit are used to reverse the polarity of the PWM output waveform.

ETM, PWM Mode, Edge – aligned Mode, T1CCLR=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 is $f_{SYS}/4$, CCRP=100b and 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 the 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, T1CCLR=1

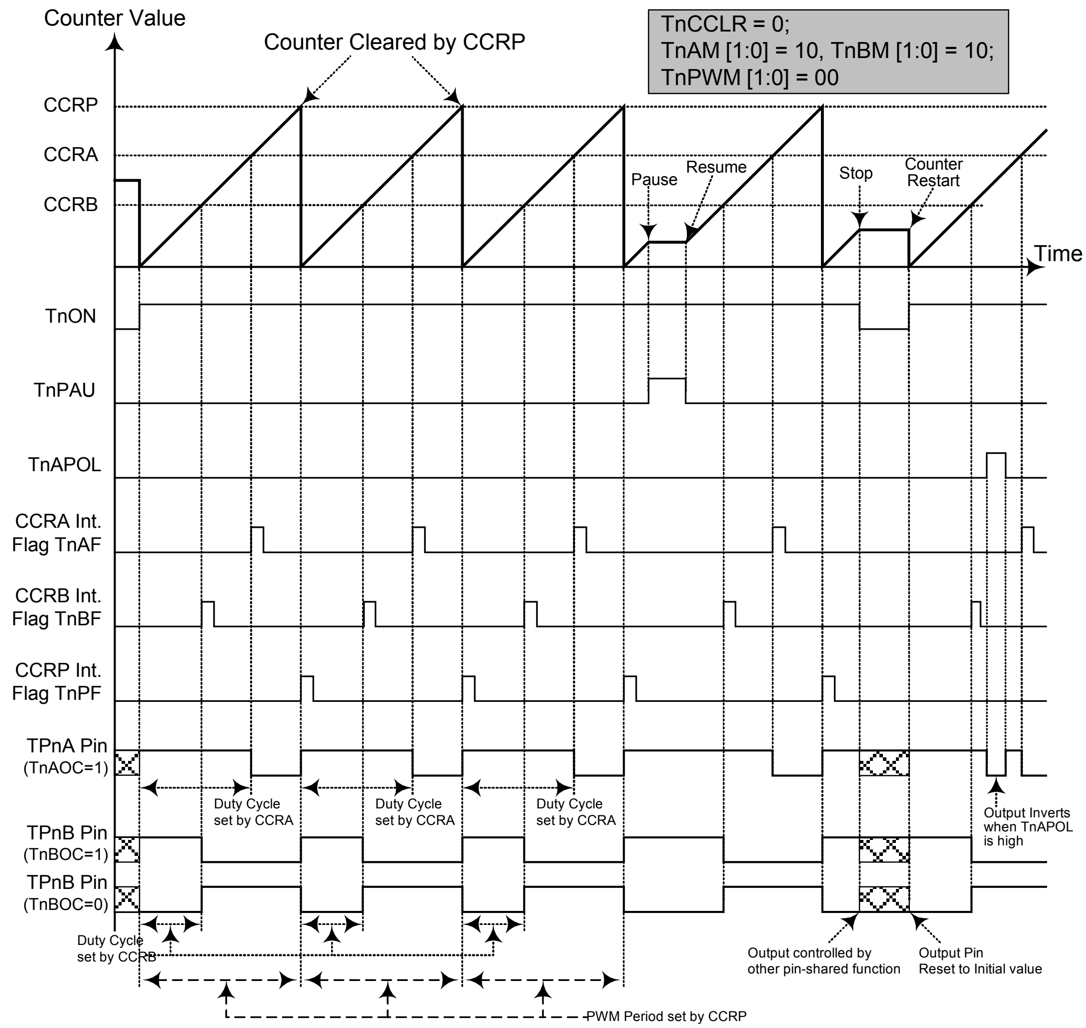
CCRA	1	2	3	511	512	1021	1022	1023
Period	1	2	3	511	512	1021	1022	1023
B Duty	CCRB							

ETM, PWM Mode, Center – aligned Mode, T1CCLR=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							

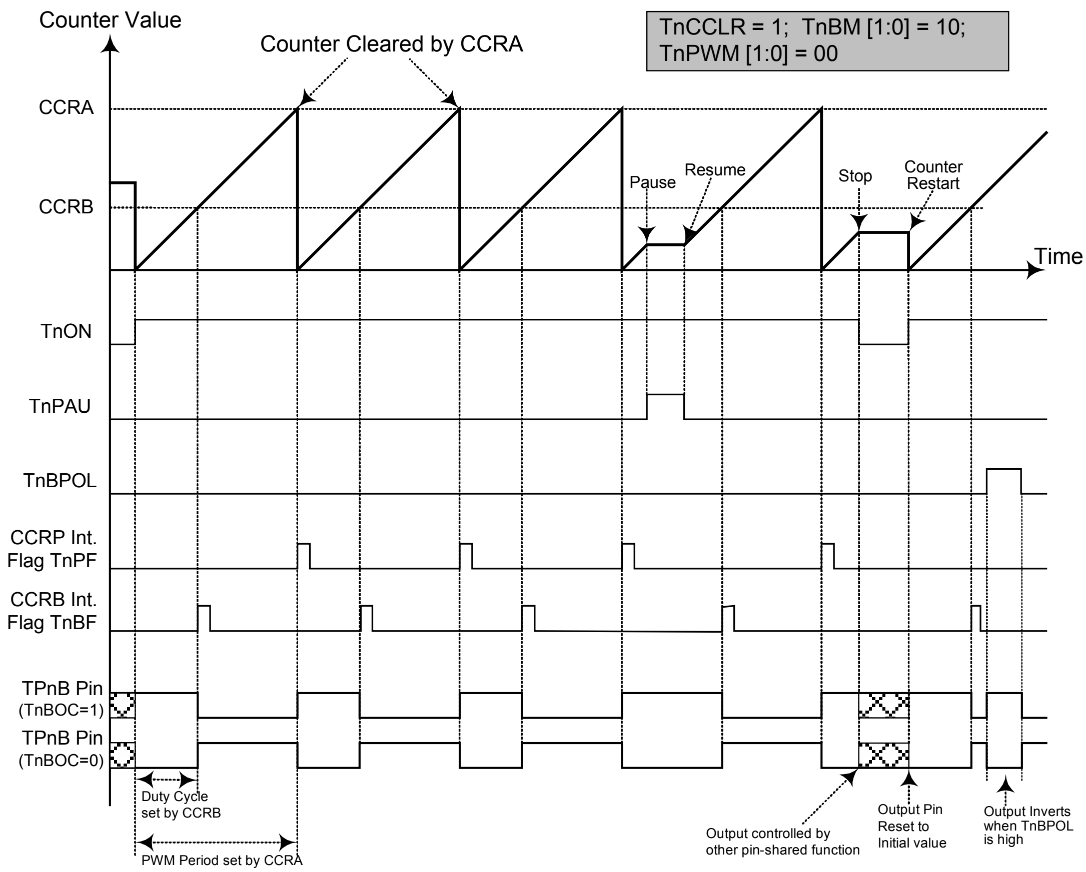
ETM, PWM Mode, Center – aligned Mode, T1CCLR=1

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

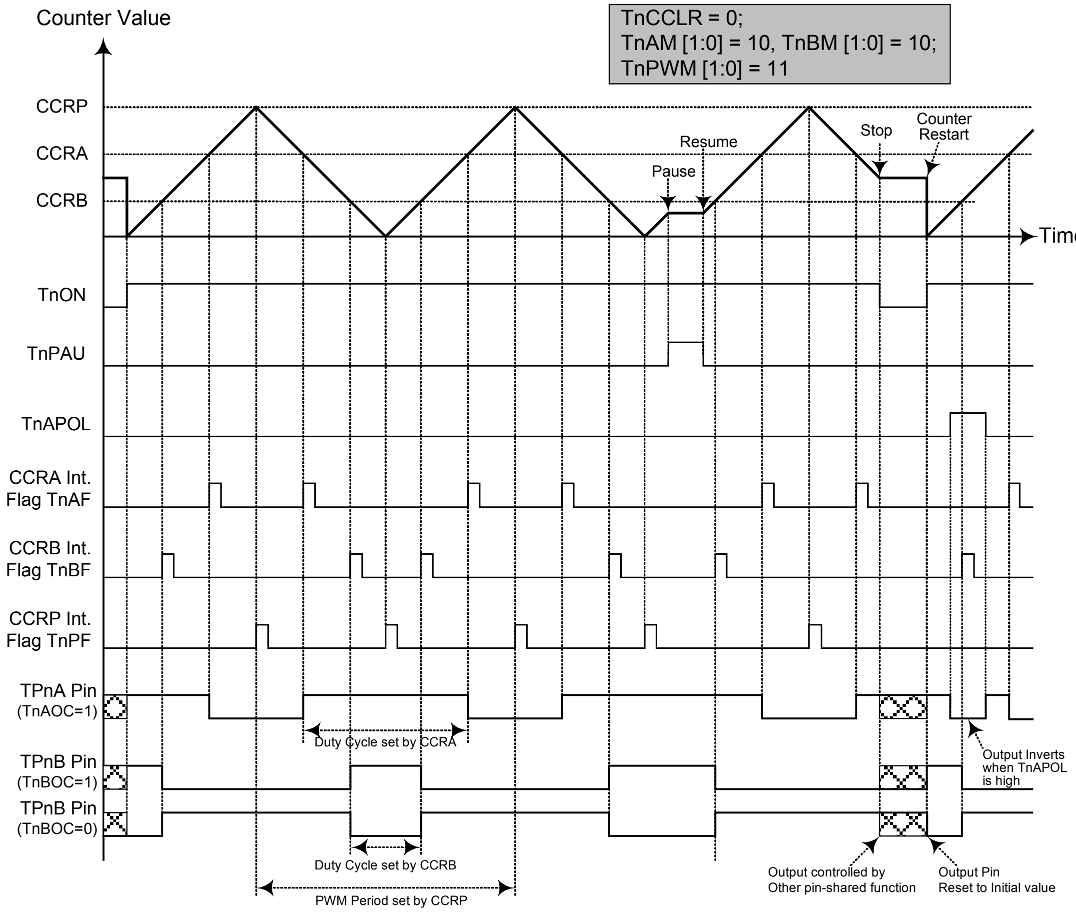
Note: 1. Here TnCCLR=0 therefore CCRP clears counter and determines PWM period
2. Internal PWM function continues even when TnAIO [1:0] (or TnBIO [1:0])=00 or 01
3. CCRA controls TPnA PWM duty and CCRB controls TPnB PWM duty
4. n=1



ETM PWM Mode – Edge Aligned

Note: 1. Here TnCCLR=1 therefore CCRA clears counter and determines PWM period

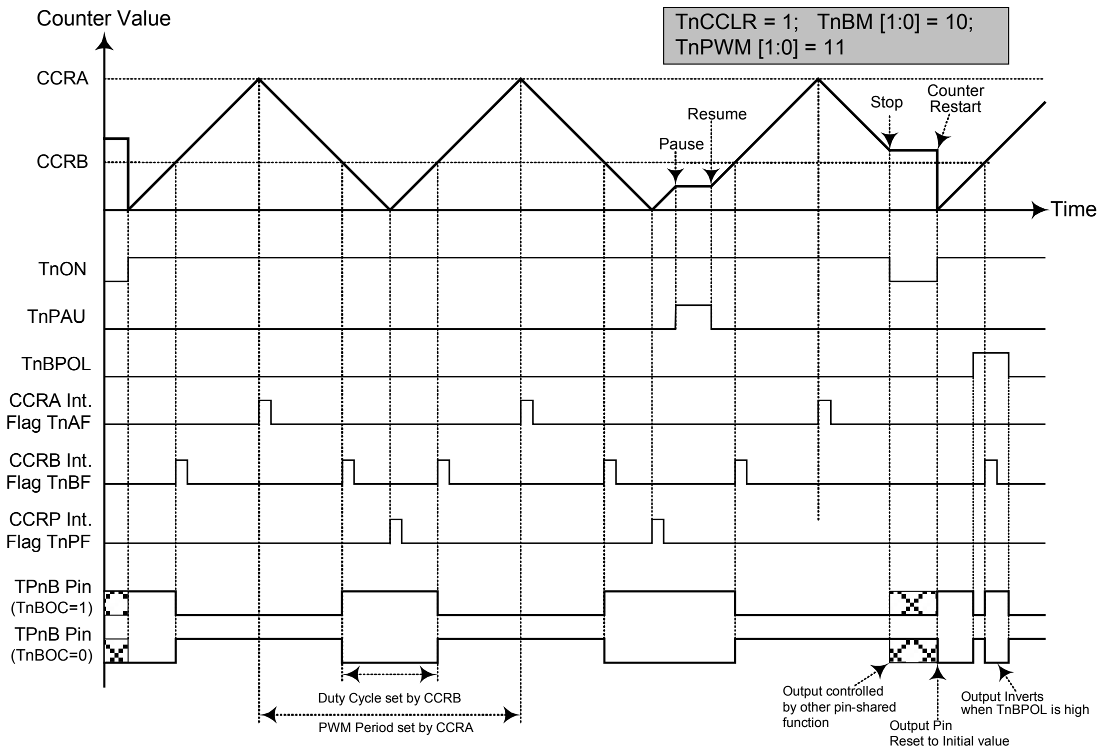
2. Internal PWM function continues even when TnBIO [1:0]=00 or 01
3. CCRA controls TPnB PWM period and CCRB controls TPnB PWM duty
4. Here the TM pin control register should not enable the TPnA pin as a TM output pin
5. n=1



ETM PWM Mode – Centre Aligned

Note: 1. Here TnCCLR=0 therefore CCRP clears counter and determines PWM period

2. TnPWM1/TnPWM0=11 therefore PWM is centre aligned
3. Internal PWM function continues even when TnAIO [1:0] (or TnBIO [1:0])=00 or 01
4. CCRA controls TPnA PWM duty and CCRB controls TPnB PWM duty
5. CCRP will generate an interrupt request when the counter decrements to its zero value
6. n=1



ETM PWM Mode – Centre Aligned

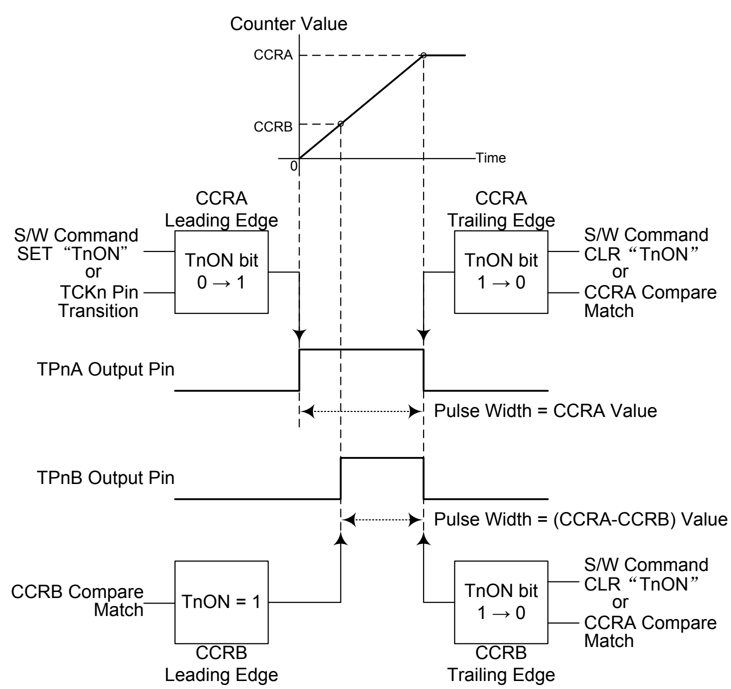
Note: 1. Here TnCCLR=1 therefore CCRA clears counter and determines PWM period
2. TnPWM1/TnPWM0=11 therefore PWM is centre aligned
3. Internal PWM function continues even when TnBIO [1:0]=00 or 01
4. CCRA controls the TPnB PWM period and CCRB controls the TPnB PWM duty
5. CCRP will generate an interrupt request when the counter decrements to its zero value
6. n=1

Single Pulse Output Mode

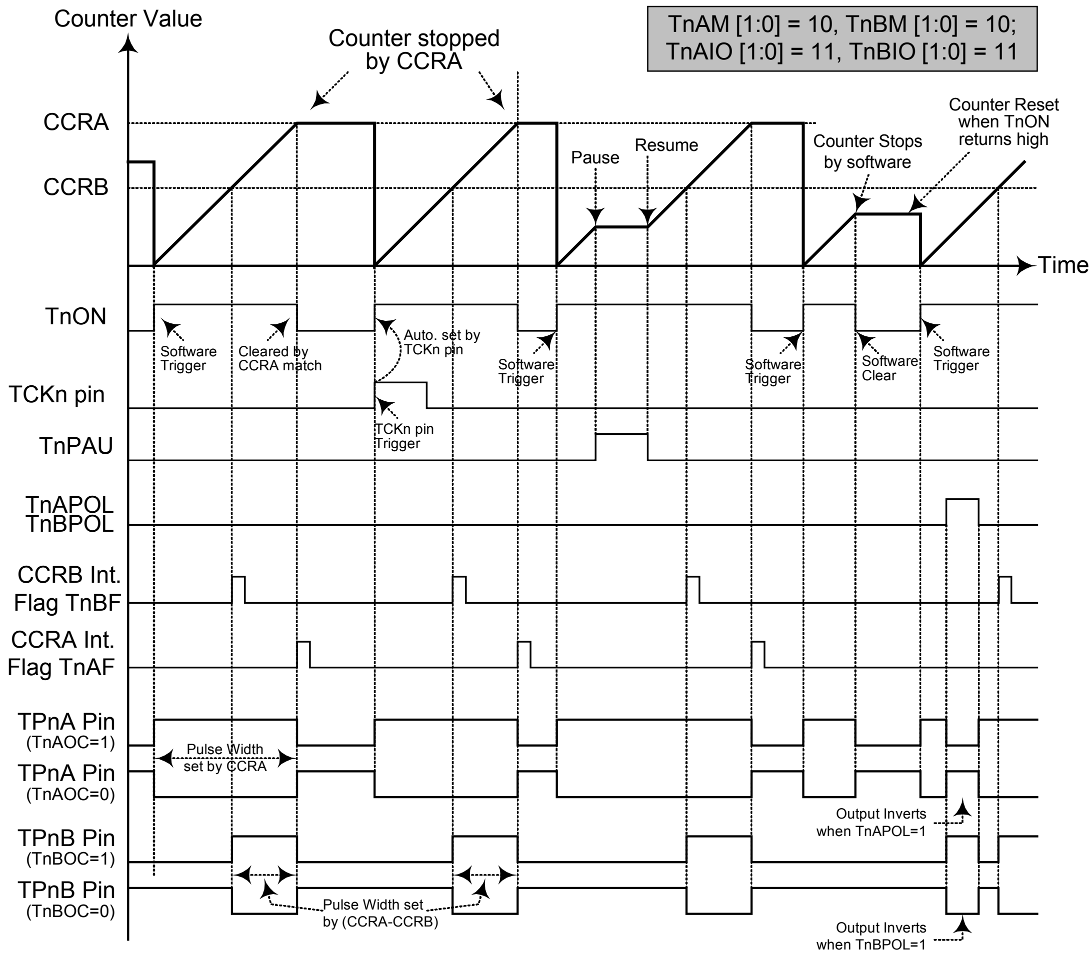
To select this mode, the required bit pairs, T1AM1, T1AM0 and T1BM1, T1BM0 should be set to 10 respectively and also the corresponding T1AIO1, T1AIO0 and T1BIO1, T1BIO0 bits should be set to 11 respectively. The Single Pulse Output Mode, as the name suggests, will generate a single shot pulse on the TM output pin.

The trigger for the pulse TP1A output leading edge is a low to high transition of the T1ON bit, which can be implemented using the application program. The trigger for the pulse TP1B 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 T1ON bit can also be made to automatically change from low to high using the external TCK1 pin, which will in turn initiate the Single Pulse output of TP1A. When the T1ON bit transitions to a high level, the counter will start running and the pulse leading edge of TP1A will be generated. The T1ON bit should remain high when the pulse is in its active state. The generated pulse trailing edge of TP1A and TP1B will be generated when the T1ON 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 T1ON bit and thus generate the Single Pulse output trailing edge of TP1A and TP1B. In this way the CCRA value can be used to control the pulse width of TP1A. The CCRA-CCRB value can be used to control the pulse width of TP1B. 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 T1ON bit changes from low to high when the counter restarts. In the Single Pulse Mode CCRP is not used. The T1CCLR bit is also not used.



Single Pulse Generation (n=1)



ETM – Single Pulse Mode

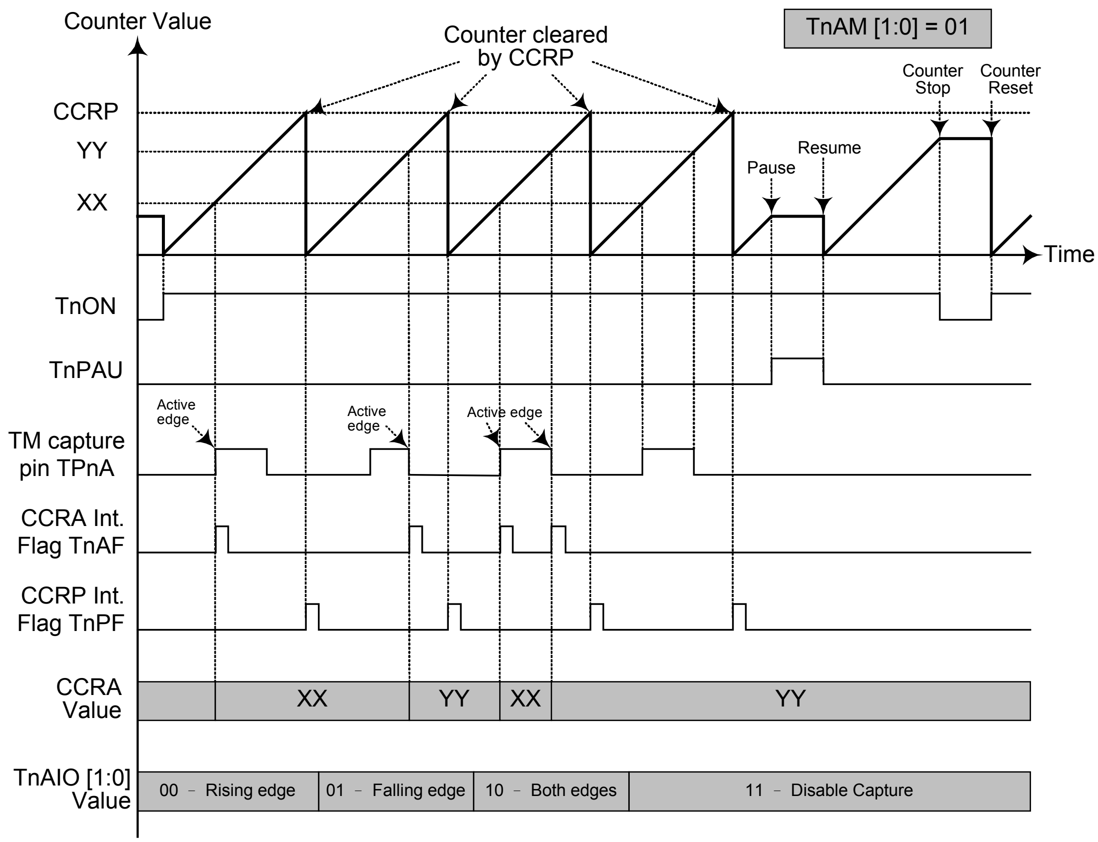
Note: 1. Counter stopped by CCRA
2. CCRP is not used
3. The pulse triggered by the TCKn pin or by setting the TnON bit high
4. A TCKn pin active edge will automatically set the TnON bit high.
5. In the Single Pulse Mode, TnAIO [1:0] and TnBIO [1:0] must be set to “11” and can not be changed.
6. n=1

Capture Input Mode

To select this mode bits T1AM1, T1AM0 and T1BM1, T1BM0 in the TM1C1 and TM1C2 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 TP1A and TP1B_0, TP1B_1 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 T1AIO1, T1AIO0 and T1BIO1, T1BIO0 bits in the TM1C1 and TM1C2 registers. The counter is started when the T1ON bit changes from low to high which is initiated using the application program.

When the required edge transition appears on the TP1A and TP1B_0, TP1B_1 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 TP1A and TP1B_0, TP1B_1 pins the counter will continue to free run until the T1ON 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 T1AIO1, T1AIO0 and T1BIO1, T1BIO0 bits can select the active trigger edge on the TP1A and TP1B_0, TP1B_1 pins to be a rising edge, falling edge or both edge types. If the T1AIO1, T1AIO0 and T1BIO1, T1BIO0 bits are both set high, then no capture operation will take place irrespective of what happens on the TP1A and TP1B_0, TP1B_1 pins, however it must be noted that the counter will continue to run.

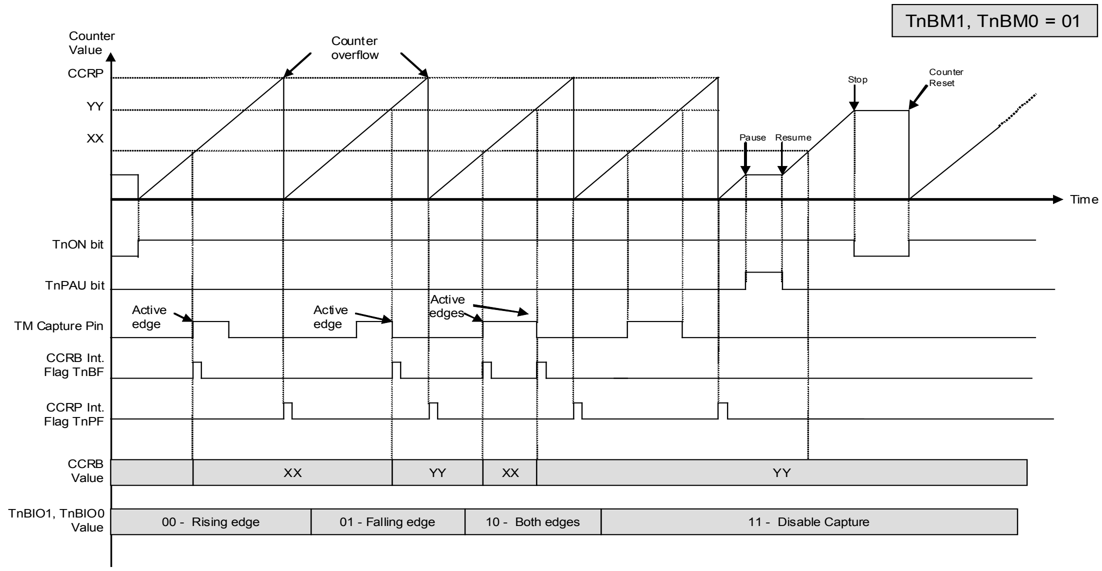
As the TP1A and TP1B_0, TP1B_1 pins are pin shared with other functions, care must be taken if the TM is in the Capture Input Mode. This is because if the pin is setup as an output, then any transitions on this pin may cause an input capture operation to be executed. The T1CCLR, T1AOC, T1BOC, T1APOL and T1BPOL bits are not used in this mode.



ETM CCRA Capture Input Mode

Note: 1. T1AM [1:0] = 01 and active edge set by the T1AIO [1:0] bits

2. TM Capture input pin active edge transfers 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.
6. n=1



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. The TnCCLR bit is not used
4. No output function – TnBOC and TnBPOL bits are not used
5. CCRP determines the counter value and the counter has a maximum count value when CCRP is equal to zero.
6. n=1

Analog to Digital Converter

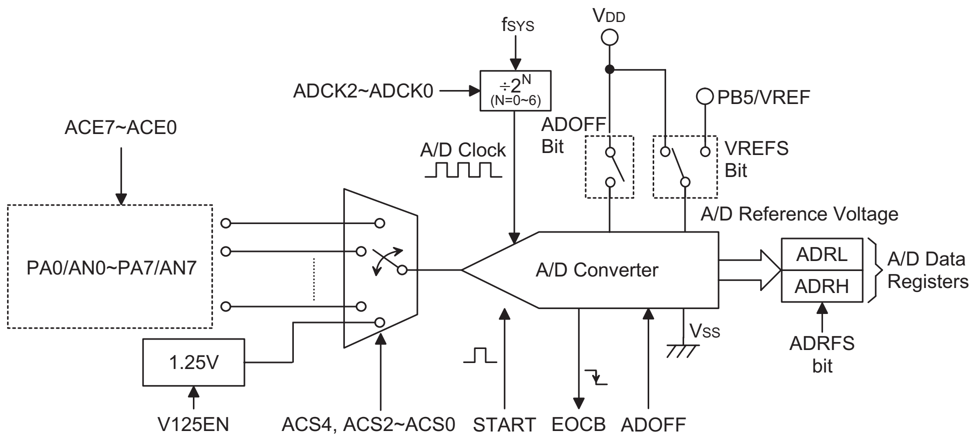
The need to interface to real world analog signals is a common requirement for many electronic systems. However, to properly process these signals by a microcontroller, they must first be converted into digital signals by A/D converters. By integrating the A/D conversion electronic circuitry into the microcontroller, the need for external components is reduced significantly with the corresponding follow-on benefits of lower costs and reduced component space requirements. The A/D Converter is only contained in the HT66F30-1 and HT66F20-1 devices.

A/D Overview

The HT66F30-1 contains a multi-channel analog to digital converter which can directly interface to external analog signals, such as that from sensors or other control signals and convert these signals directly into either a 12-bit digital value.

Part No.	Input Channels	A/D Channel Select Bits	Input Pins
HT66F30-1/HT66F20-1	8	ACS4, ACS2~ACS0	AN0~AN7

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



A/D Converter Structure

A/D Converter Register Description

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

Register Name	Bit							
	7	6	5	4	3	2	1	0
ADRL(ADRFS=0)	D3	D2	D1	D0	—	—	—	—
ADRL(ADRFS=1)	D7	D6	D5	D4	D3	D2	D1	D0
ADRH(ADRFS=0)	D11	D10	D9	D8	D7	D6	D5	D4
ADRH(ADRFS=1)	—	—	—	—	D11	D10	D9	D8
ADCR0	START	EOCB	ADOFF	ADRFS	—	ACS2	ACS1	ACS0
ADCR1	ACS4	V125EN	—	VREFS	—	ADCK2	ADCK1	ADCK0
ACERL	ACE7	ACE6	ACE5	ACE4	ACE3	ACE2	ACE1	ACE0

A/D Converter Register List – HT66F30-1/HT66F20-1

A/D Converter Data Registers – ADRL, ADRH

As the HT66F30-1 or HT66F20-1 device contains an internal 12-bit A/D converter, it requires two data registers to store the converted value. These are a high byte register, known as ADRH, and a low byte register, known as ADRL. After the conversion process takes place, these registers can be directly read by the microcontroller to obtain the digitised conversion value. As only 12 bits of the 16-bit register space is utilised, the format in which the data is stored is controlled by the ADRFS bit in the ADCR0 register as shown in the accompanying table. D0~D11 are the A/D conversion result data bits. Any unused bits will be read as zero.

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

A/D Data Registers

A/D Converter Control Registers – ADCR0, ADCR1, ACERL

To control the function and operation of the A/D converter, three control registers known as ADCR0, ADCR1 and ACERL are provided. These 8-bit registers define functions such as the selection of which analog channel is connected to the internal A/D converter, the digitised data format, the A/D clock source as well as controlling the start function and monitoring the A/D converter end of conversion status. The ACS2~ACS0 bits in the ADCR0 register and ACS4 bit is the ADCR1 register define the ADC input channel number. As the devices contain only one actual analog to digital converter hardware circuit, each of the individual 8 analog inputs must be routed to the converter. It is the function of the ACS4 and ACS2~ACS0 bits to determine which analog channel input pins or internal 1.25V is actually connected to the internal A/D converter.

The ACERL control register contains the ACE7~ACE0 bits which determine which pins on Port A is used as analog inputs for the A/D converter input and which pins are not to be used as the A/D converter input. Setting the corresponding bit high will select the A/D input function, clearing the bit to zero will select either the I/O or other pin-shared function. When the pin is selected to be an A/D input, its original function whether it is an I/O or other pin-shared function will be removed. In addition, any internal pull-high resistors connected to these pins will be automatically removed if the pin is selected to be an A/D input.

ADCR0 Register

Bit	7	6	5	4	3	2	1	0
Name	START	EOCB	ADOFF	ADRFS	—	ACS2	ACS1	ACS0
R/W	R/W	R	R/W	R/W	—	R/W	R/W	R/W
POR	0	1	1	0	—	0	0	0

Bit 7 **START:** Start the A/D conversion
0→1→0: Start
0→1: Reset the A/D converter and set EOCB to "1"
This bit is used to initiate an A/D conversion process. The bit is normally low but if set high and then cleared low again, the A/D converter will initiate a conversion process. When the bit is set high the A/D converter will be reset.

Bit 6 **EOCB:** End of A/D conversion flag
0: A/D conversion ended
1: A/D conversion in progress
This read only flag is used to indicate when an A/D conversion process has completed. When the conversion process is running, the bit will be high.

Bit 5 **ADOFF** : ADC module power on/off control bit
0: ADC module power on
1: ADC module power off
This bit controls the power to the A/D internal function. This bit should be cleared to zero to enable the A/D converter. If the bit is set high then the A/D converter will be switched off reducing the device power consumption. As the A/D converter will consume a limited amount of power, even when not executing a conversion, this may be an important consideration in power sensitive battery powered applications.
Note: 1. it is recommended to set ADOFF=1 before entering IDLE/SLEEP Mode for saving power.
2. ADOFF=1 will power down the ADC module.

Bit 4 **ADRFS:** ADC Data Format Control
0: ADC Data MSB is ADRH bit 7, LSB is ADRL bit 4
1: ADC Data MSB is ADRH bit 3, LSB is ADRL bit 0
This bit controls the format of the 12-bit converted A/D value in the two A/D data registers. Details are provided in the A/D data register section.

Bit 3 Unimplemented, read as "0"

Bit 2~0 **ACS2, ACS1, ACS0:** Select A/D channel (when ACS4 is "0")
000: AN0
001: AN1
010: AN2
011: AN3
100: AN4
101: AN5
110: AN6
111: AN7
These are the A/D channel select control bits. As there is only one internal hardware A/D converter each of the eight A/D inputs must be routed to the internal converter using these bits. If bit ACS4 in the ADCR1 register is set high then the internal 1.25V will be routed to the A/D Converter.

ADCR1 Register

Bit	7	6	5	4	3	2	1	0
Name	ACS4	V125EN	—	VREFS	—	ADCK2	ADCK1	ADCK0
R/W	R/W	R/W	—	R/W	—	R/W	R/W	R/W
POR	0	0	—	0	—	0	0	0

Bit 7 **ACS4:** Select Internal 1.25V bandgap voltage as ADC input
0: Disable
1: Enable

This bit enables the1.25V bandgap voltage to be connected to the A/D converter. The V125EN bit must first have been set to enable the bandgap circuit 1.25V voltage to be used by the A/D converter. When the ACS4 bit is set high, the bandgap 1.25V voltage will be routed to the A/D converter and the other A/D input channels disconnected.

Bit 6 **V125EN:** Internal 1.25V Control
0: Disable
1: Enable

This bit controls the internal Bandgap circuit on/off function to the A/D converter. When the bit is set high the bandgap voltage 1.25V can be used as an A/D converter input. If the bandgap voltage 1.25V is not used by the A/D converter and the LVR/LVD function is disabled then the bandgap reference circuit will be automatically switched off to conserve power. When 1.25V is switched on for use by the A/D converter, a time t_{BG} should be allowed for the bandgap circuit to stabilise before implementing an A/D conversion.

Bit 5 Unimplemented, read as "0"

Bit 4 **VREFS:** Select ADC reference voltage
0: Internal ADC power
1: VREF pin

This bit is used to select the reference voltage for the A/D converter. If the bit is high, then the A/D converter reference voltage is supplied on the external VREF pin. If the pin is low, then the internal reference is used which is taken from the power supply pin VDD.

Bit 3 Unimplemented, read as "0"

Bit 2~0 **ADCK2, ADCK1, ADCK0:** Select ADC clock source
000: f_{SYS}
001: $f_{SYS}/2$
010: $f_{SYS}/4$
011: $f_{SYS}/8$
100: $f_{SYS}/16$
101: $f_{SYS}/32$
110: $f_{SYS}/64$
111: Undefined

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

ACERL Register

Bit	7	6	5	4	3	2	1	0
Name	ACE7	ACE6	ACE5	ACE4	ACE3	ACE2	ACE1	ACE0
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 **ACE7:** Define PA7 is A/D input or not
0: Not A/D input
1: A/D input, AN7

Bit 6 **ACE6:** Define PA6 is A/D input or not
0: Not A/D input
1: A/D input, AN6

Bit 5 **ACE5:** Define PA5 is A/D input or not
0: Not A/D input
1: A/D input, AN5

Bit 4 **ACE4:** Define PA4 is A/D input or not
0: Not A/D input
1: A/D input, AN4

Bit 3 **ACE3:** Define PA3 is A/D input or not
0: Not A/D input
1: A/D input, AN3

Bit 2 **ACE2:** Define PA2 is A/D input or not
0: Not A/D input
1: A/D input, AN2

Bit 1 **ACE1:** Define PA1 is A/D input or not
0: Not A/D input
1: A/D input, AN1

Bit 0 **ACE0:** Define PA0 is A/D input or not
0: Not A/D input
1: A/D input, AN0

A/D Operation

The START bit in the ADCR0 register is used to start and reset the A/D converter. When the microcontroller sets this bit from low to high and then low again, an analog to digital conversion cycle will be initiated. When the START bit is brought from low to high but not low again, the EOCB bit in the ADCR0 register will be set high and the analog to digital converter will be reset. It is the START bit that is used to control the overall start operation of the internal analog to digital converter.

The EOCB bit in the ADCR0 register is used to indicate when the analog to digital conversion process is complete. This bit will be automatically set to “0” by the microcontroller after a conversion cycle has ended. In addition, the corresponding A/D interrupt request flag will be set in the interrupt control register, and if the interrupts are enabled, an appropriate internal interrupt signal will be generated. This A/D internal interrupt signal will direct the program flow to the associated A/D internal interrupt address for processing. If the A/D internal interrupt is disabled, the microcontroller can be used to poll the EOCB bit in the ADCR0 register to check whether it has been cleared as an alternative method of detecting the end of an A/D conversion cycle.

The clock source for the A/D converter, which originates from the system clock f_{SYS}, can be chosen to be either f_{SYS} or a subdivided version of f_{SYS}. The division ratio value is determined by the ADCK2~ADCK0 bits in the ADCR1 register.

Although the A/D clock source is determined by the system clocky, f_{SYS}, and by bits ADCK2~ADCK0, there are some limitations on the maximum A/D clock source speed that can be selected. As the minimum value of permissible A/D clock period, t_{ADCK}, is 0.5μs, care must be taken for system clock frequencies equal to or greater than 4MHz. For example, if the system clock operates at a frequency of 4MHz, the ADCK2~ADCK0 bits should not be set to "000". Doing so will give A/D clock periods that are less than the minimum A/D clock period which may result in inaccurate A/D conversion values. Refer to the following table for examples, where values marked with an asterisk * show where, depending upon the device, special care must be taken, as the values may be less than the specified minimum A/D Clock Period.

f_{SYS}	A/D Clock Period (t_{ADCK})							
	ADCK2, ADCK1, ADCK0 =000 (f_{SYS})	ADCK2, ADCK1, ADCK0 =001 (f_{SYS}/2)	ADCK2, ADCK1, ADCK0 =010 (f_{SYS}/4)	ADCK2, ADCK1, ADCK0 =011 (f_{SYS}/8)	ADCK2, ADCK1, ADCK0 =100 (f_{SYS}/16)	ADCK2, ADCK1, ADCK0 =101 (f_{SYS}/32)	ADCK2, ADCK1, ADCK0 =110 (f_{SYS}/64)	ADCK2, ADCK1, ADCK0 =111
1MHz	1μs	2μs	4μs	8μs	16μs	32μs	64μs	Undefined
2MHz	500ns	1μs	2μs	4μs	8μs	16μs	32μs	Undefined
4MHz	250ns*	500ns	1μs	2μs	4μs	8μs	16μs	Undefined
8MHz	125ns*	250ns*	500ns	1μs	2μs	4μs	8μs	Undefined
12MHz	83ns*	167ns*	333ns*	667ns	1.33μs	2.67μs	5.33μs	Undefined

A/D Clock Period Examples

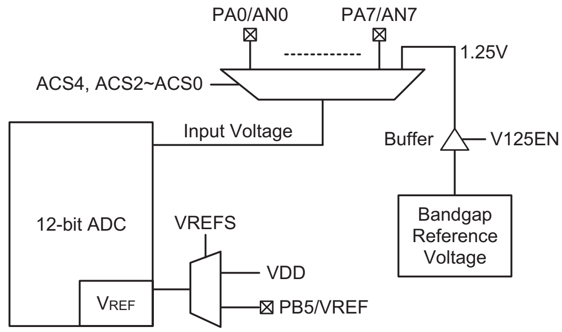
Controlling the power on/off function of the A/D converter circuitry is implemented using the ADOFF bit in the ADCR0 register. This bit must be zero to power on the A/D converter. Even if no pins are selected for use as A/D inputs by clearing the ACE7~ACE0 bits in the ACERL registers, if the ADOFF bit is zero then some power will still be consumed. In power conscious applications it is therefore recommended that the ADOFF is set high to reduce power consumption when the A/D converter function is not being used.

The reference voltage supply to the A/D Converter can be supplied from either the positive power supply pin, VDD, or from an external reference sources supplied on pin VREF. The desired selection is made using the VREFS bit. As the VREF pin is pin-shared with other functions, when the VREFS bit is set high, the VREF pin function will be selected and the other pin functions will be disabled automatically.

A/D Input Pins

All of the A/D analog input pins are pin-shared with the I/O pins on Port A as well as other functions. The ACE7~ACE0 bits in the ACERL register, determine whether the input pins are setup as A/D converter analog inputs or whether they have other functions. If the ACE7~ACE0 bits for its corresponding pin is set high then the pin will be setup to be an A/D converter input and the original pin functions disabled. In this way, pins can be changed under program control to change their function between A/D inputs and other functions. All pull-high resistors, which are setup through register programming, will be automatically disconnected if the pins are setup as A/D inputs. Note that it is not necessary to first setup the A/D pin as an input in the PAC port control registers to enable the A/D input as when the ACE7~ACE0 bits enable an A/D input, the status of the port control register will be overridden.

The A/D converter has its own reference voltage pin, VREF, however the reference voltage can also be supplied from the power supply pin, a choice which is made through the VREFS bit in the ADCR1 register. The analog input values must not be allowed to exceed the value of V_{REF}.



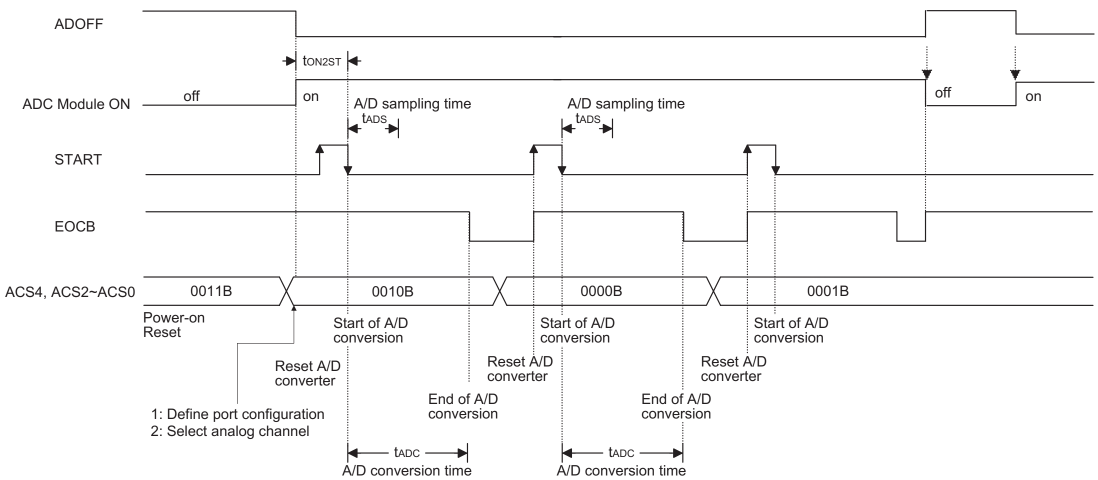
A/D Input Structure

Summary of A/D Conversion Steps

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

- Step 1
  Select the required A/D conversion clock by correctly programming bits ADCK2~ADCK0 in the ADCR1 register.
- Step 2
  Enable the A/D by clearing the ADOFF bit in the ADCR0 register to zero.
- Step 3
  Select which channel is to be connected to the internal A/D converter by correctly programming the ACS4, ACS2~ACS0 bits which are also contained in the ADCR1 and ADCR0 register.
- Step 4
  Select which pins are to be used as A/D inputs and configure them by correctly programming the ACE7~ACE0 bits in the ACERL register.
- Step 5
  If the interrupts are to be used, the interrupt control registers must be correctly configured to ensure the A/D converter interrupt function is active. The master interrupt control bit, EMI, and the A/D converter interrupt bit, ADE, must both be set high to do this.
- Step 6
  The analog to digital conversion process can now be initialised by setting the START bit in the ADCR0 register from low to high and then low again. Note that this bit should have been originally cleared to zero.
- Step 7
  To check when the analog to digital conversion process is complete, the EOCB bit in the ADCR0 register can be polled. The conversion process is complete when this bit goes low. When this occurs the A/D data register ADRL and ADRH can be read to obtain the conversion value. As an alternative method, if the interrupts are enabled and the stack is not full, the program can wait for an A/D interrupt to occur.
  Note: When checking for the end of the conversion process, if the method of polling the EOCB bit in the ADCR0 register is used, the interrupt enable step above can be omitted.

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



A/D Conversion Timing

Programming Considerations

During microcontroller operations where the A/D converter is not being used, the A/D internal circuitry can be switched off to reduce power consumption, by setting bit ADOFF high in the ADCR0 register. When this happens, the internal A/D converter circuits will not consume power irrespective of what analog voltage is applied to their input lines. If the A/D converter input lines are used as normal I/Os, then care must be taken as if the input voltage is not at a valid logic level, then this may lead to some increase in power consumption.

A/D Transfer Function

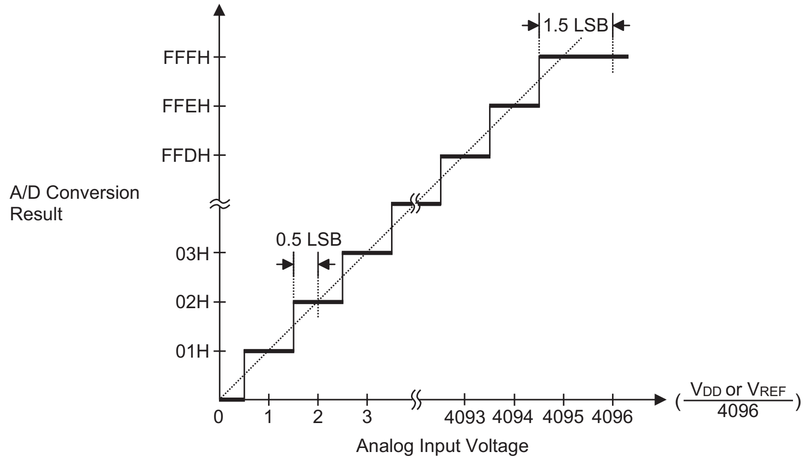
As the HT66F30-1 contains a 12-bit A/D converter, its full-scale converted digitised value is equal to FFFH. Since the full-scale analog input value is equal to the V_{DD} or V_{REF} voltage, this gives a single bit analog input value of V_{DD} or V_{REF} divided by 4096.

$$1\ \text{LSB}=(V_{DD}\ \text{or}\ V_{REF})\div 4096$$

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

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

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



Ideal A/D Transfer Function

A/D Programming Example

The following two programming examples illustrate how to setup and implement an A/D conversion. In the first example, the method of polling the EOCB bit in the ADCR0 register is used to detect when the conversion cycle is complete, whereas in the second example, the A/D interrupt is used to determine when the conversion is complete.

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

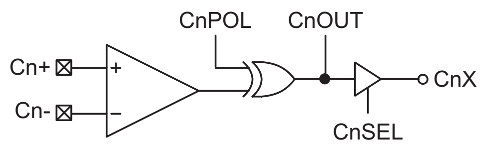
```
clr  ADE                     ; disable ADC interrupt
mov  a,03H
mov  ADCR1,a                 ; select fSYS/8 as A/D clock and switch off 1.25V
clr  ADOFF
mov  a,0Fh                   ; setup ACERL to configure pins AN0~AN3
mov  ACERL,a
mov  a,00h
mov  ADCR0,a                 ; enable and connect AN0 channel to A/D converter
:
start_conversion:
     clr START               ; high pulse on start bit to initiate conversion
     set START               ; reset A/D
     clr START               ; start A/D
polling_EOC:
     sz  EOCB                ; poll the ADCR0 register EOCB bit to detect end
                             ; of A/D conversion
     jmp polling_EOC         ; continue polling
     mov a,ADRL              ; read low byte conversion result value
     mov ADRL_buffer,a       ; save result to user defined register
     mov a,ADRH              ; read high byte conversion result value
     mov ADRH_buffer,a       ; save result to user defined register
:
:
jmp  start_conversion        ; start next a/d conversion
```

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

```
clr  ADE                 ; disable ADC interrupt
mov  a,03H
mov  ADCR1,a                ; select fSYS/8 as A/D clock and switch off 1.25V
Clr  ADOFF
mov  a,0Fh                  ; setup ACERL to configure pins AN0~AN3
mov  ACERL,a
mov  a,00h
mov  ADCR0,a                ; enable and connect AN0 channel to A/D converter
Start_conversion:
     clr START              ; high pulse on START bit to initiate conversion
     set START              ; reset A/D
     clr START              ; start A/D
     clr ADF                ; clear ADC interrupt request flag
     set ADE                ; enable ADC interrupt
     set EMI                ; enable global interrupt
:
:
                            ; ADC interrupt service routine
ADC_ISR:
     mov acc_stack,a        ; save ACC to user defined memory
     mov a,STATUS
     mov status_stack,a     ; save STATUS to user defined memory
:
:
     mov a,ADRL             ; read low byte conversion result value
     mov adrl_buffer,a      ; save result to user defined register
     mov a,ADRH             ; read high byte conversion result value
     mov adrh_buffer,a      ; save result to user defined register
:
:
EXIT_INT_ISR:
     mov a,status_stack
     mov STATUS,a           ; restore STATUS from user defined memory
     mov a,acc_stack        ; restore ACC from user defined memory
     reti
```

Comparators

Two independent analog comparators are contained within these devices. These functions offer flexibility via their register controlled features such as power-down, polarity select, hysteresis etc. In sharing their pins with normal I/O pins the comparators do not waste precious I/O pins if there functions are otherwise unused.



Comparator

Comparator Operation

The devices contain two comparator functions which are used to compare two analog voltages and provide an output based on their difference. Full control over the two internal comparators is provided via two control registers, CP0C and CP1C, one assigned to each comparator. The comparator output is recorded via a bit in their respective control register, but can also be transferred out onto a shared I/O pin. Additional comparator functions include, output polarity, hysteresis functions and power down control.

Any pull-high resistors connected to the shared comparator input pins will be automatically disconnected when the comparator is enabled. As the comparator inputs approach their switching level, some spurious output signals may be generated on the comparator output due to the slow rising or falling nature of the input signals. This can be minimised by selecting the hysteresis function will apply a small amount of positive feedback to the comparator. Ideally the comparator should switch at the point where the positive and negative inputs signals are at the same voltage level, however, unavoidable input offsets introduce some uncertainties here. The hysteresis function, if enabled, also increases the switching offset value.

Comparator Registers

There are two registers for overall comparator operation, one for each comparator. As corresponding bits in the two registers have identical functions, they following register table applies to both registers.

Register Name	Bit							
	7	6	5	4	3	2	1	0
CP0C	C0SEL	C0EN	C0POL	C0OUT	C0OS	—	—	C0HYEN
CP1C	C1SEL	C1EN	C1POL	C1OUT	C1OS	—	—	C1HYEN

Comparator Registers List

CP0C Register

Bit	7	6	5	4	3	2	1	0
Name	C0SEL	C0EN	C0POL	C0OUT	C0OS	—	—	C0HYEN
R/W	R/W	R/W	R/W	R	R/W	—	—	R/W
POR	1	0	0	0	0	—	—	1

Bit 7 **C0SEL:** Select Comparator pins or I/O pins
0: I/O pin select
1: Comparator pin select
This is the Comparator pin or I/O pin select bit. If the bit is high the comparator will be selected and the two comparator input pins will be enabled. As a result, these two pins will lose their I/O pin functions. Any pull-high configuration options associated with the comparator shared pins will also be automatically disconnected.

Bit 6 **C0EN:** Comparator On/Off control
0: Off
1: On
This is the Comparator on/off control bit. If the bit is zero the comparator will be switched off and no power consumed even if analog voltages are applied to its inputs. For power sensitive applications this bit should be cleared to zero if the comparator is not used or before the devices enter the SLEEP or IDLE mode.

Bit 5 **C0POL:** Comparator output polarity
0: output not inverted
1: output inverted
This is the comparator polarity bit. If the bit is zero then the C0OUT bit will reflect the non-inverted output condition of the comparator. If the bit is high the comparator C0OUT bit will be inverted.

Bit 4 **C0OUT:** Comparator output bit
C0POL=0
0: C0+ < C0-
1: C0+ > C0-
C0POL=1
0: C0+ > C0-
1: C0+ < C0-
This bit stores the comparator output bit. The polarity of the bit is determined by the voltages on the comparator inputs and by the condition of the C0POL bit.

Bit 3 **C0OS:** Output path select
0: C0X pin
1: Internal use
This is the comparator output path select control bit. If the bit is set to "0" and the C0SEL bit is "1" the comparator output is connected to an external C0X pin. If the bit is set to "1" or the C0SEL bit is "0" the comparator output signal is only used internally by the devices allowing the shared comparator output pin to retain its normal I/O operation.

Bit 2~1 Unimplemented, read as "0"

Bit 0 **C0HYEN:** Hysteresis Control
0: Off
1: On
This is the hysteresis control bit and if set high will apply a limited amount of hysteresis to the comparator, as specified in the Comparator Electrical Characteristics table. The positive feedback induced by hysteresis reduces the effect of spurious switching near the comparator threshold.

CP1C Register

Bit	7	6	5	4	3	2	1	0
Name	C1SEL	C1EN	C1POL	C1OUT	C1OS	—	—	C1HYEN
R/W	R/W	R/W	R/W	R	R/W	—	—	R/W
POR	1	0	0	0	0	—	—	1

Bit 7 **C1SEL:** Select Comparator pins or I/O pins
0: I/O pin select
1: Comparator pin select
This is the Comparator pin or I/O pin select bit. If the bit is high the comparator will be selected and the two comparator input pins will be enabled. As a result, these two pins will lose their I/O pin functions. Any pull-high configuration options associated with the comparator shared pins will also be automatically disconnected.

Bit 6 **C1EN:** Comparator On/Off control
0: Off
1: On
This is the Comparator on/off control bit. If the bit is zero the comparator will be switched off and no power consumed even if analog voltages are applied to its inputs. For power sensitive applications this bit should be cleared to zero if the comparator is not used or before the devices enter the SLEEP or IDLE mode.

Bit 5 **C1POL:** Comparator output polarity
0: output not inverted
1: output inverted
This is the comparator polarity bit. If the bit is zero then the C1OUT bit will reflect the non-inverted output condition of the comparator. If the bit is high the comparator C1OUT bit will be inverted.

Bit 4 **C1OUT:** Comparator output bit
C1POL=0
0: C1+ < C1-
1: C1+ > C1-
C1POL=1
0: C1+ > C1-
1: C1+ < C1-
This bit stores the comparator output bit. The polarity of the bit is determined by the voltages on the comparator inputs and by the condition of the C1POL bit.

Bit 3 **C1OS:** Output path select
0: C1X pin
1: Internal use
This is the comparator output path select control bit. If the bit is set to "0" and the C1SEL bit is "1" the comparator output is connected to an external C1X pin. If the bit is set to "1" or the C1SEL bit is "0" the comparator output signal is only used internally by the devices allowing the shared comparator output pin to retain its normal I/O operation.

Bit 2~1 Unimplemented, read as "0"

Bit 0 **C1HYEN:** Hysteresis Control
0: Off
1: On
This is the hysteresis control bit and if set high will apply a limited amount of hysteresis to the comparator, as specified in the Comparator Electrical Characteristics table. The positive feedback induced by hysteresis reduces the effect of spurious switching near the comparator threshold.

Comparator Interrupt

Each also possesses its own interrupt function. When any one of the changes state, its relevant interrupt flag will be set, and if the corresponding interrupt enable bit is set, then a jump to its relevant interrupt vector will be executed. Note that it is the changing state of the C0OUT or C1OUT bit and not the output pin which generates an interrupt. If the microcontroller is in the SLEEP or IDLE Mode and the Comparator is enabled, then if the external input lines cause the Comparator output to change state, the resulting generated interrupt flag will also generate a wake-up. If it is required to disable a wake-up from occurring, then the interrupt flag should be first set high before entering the SLEEP or IDLE Mode.

Programming Considerations

If the comparator is enabled, it will remain active when the microcontroller enters the SLEEP or IDLE Mode, however as it will consume a certain amount of power, the user may wish to consider disabling it before the SLEEP or IDLE Mode is entered.

As comparator pins are shared with normal I/O pins the I/O registers for these pins will be read as zero (port control register is "1") or read as port data register value (port control register is "0") if the comparator function is enabled.

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 interface pins are pin-shared with other I/O pins therefore the SIM interface function must first be selected using a configuration option. 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. These pull-high resistors of the SIM pin-shared I/O are selected using pull-high control registers, and also if the SIM function is enabled.

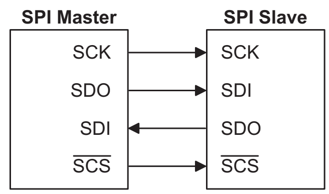
SPI Interface

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.

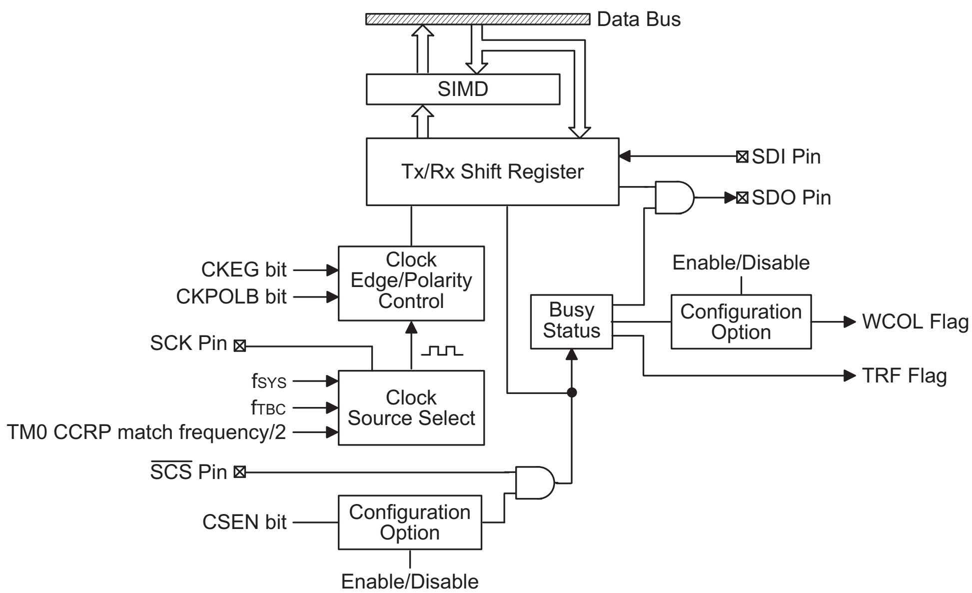
The communication is full duplex and operates as a slave/master type, where the devices can be either master or slave. Although the SPI interface specification can control multiple slave devices from a single master, but these devices provided only one $\overline{SCS}$ pin. If the master needs to control 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 other functions and with the I²C function pins, the SPI interface must first be selected by the correct bits in the SIMC0 and SIMC2 registers. After the SPI option has been selected, it can also be additionally disabled or enabled using the SIMEN bit in the SIMC0 register. 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 floating state.



SPI Master/Slave Connection



SPI Bolck Diagram

The SPI function in these devices 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
- WCOL bit enabled or disable select

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

There are several configuration options associated with the SPI interface. One of these is to enable the SIM function which selects the SIM pins rather than normal I/O pins. Note that if the configuration option does not select the SIM function then the SIMEN bit in the SIMC0 register will have no effect. Another two SPI configuration options determine if the CSEN and WCOL bits are to be used.

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^{2}C 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

SIM Registers 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^{2}C functions. Before the devices write 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 devices 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 I^{2}C function. The SIMC1 register is not used by the SPI function, only by the I^{2}C 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_{TBC}
100: SPI master mode; SPI clock is TM0 CCRP match frequency/2
101: SPI slave mode
110: I^2C slave mode
111: Non SIM function

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 TM0. If the SPI Slave Mode is selected then the clock will be supplied by an external Master devices.

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 to zero to disable the SIM interface, the SDI, SDO, SCK and $\overline{SCS}$, or SDA and SCL lines will lose their SPI or I^2C function and the SIM operating current will be reduced to a minimum value. When the bit is high the SIM interface is enabled. The SIM configuration option must have first enabled the SIM interface for this bit to be effective.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"

• 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 the application 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 $\overline{SCS}$ pin Control
0: Disable
1: Enable
The CSEN bit is used as an enable/disable for the $\overline{SCS}$ pin. If this bit is low, then the $\overline{SCS}$ pin will be disabled and placed into I/O pin or the other functions. If the bit is high the $\overline{SCS}$ pin will be enabled and used as a select pin.
Note that using the CSEN bit can be disabled or enabled via configuration option.

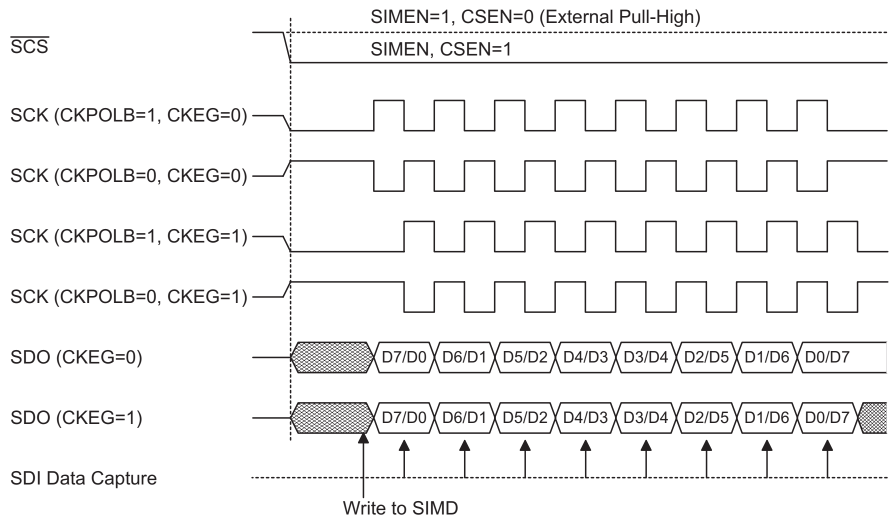
Bit 1 **WCOL:** SPI Write Collision flag
0: No collision
1: Collision
The WCOL flag is used to detect if a data collision has occurred. If this bit is high it means that data has been attempted to be written to the SIMD register during a data transfer operation. This writing operation will be ignored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

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.

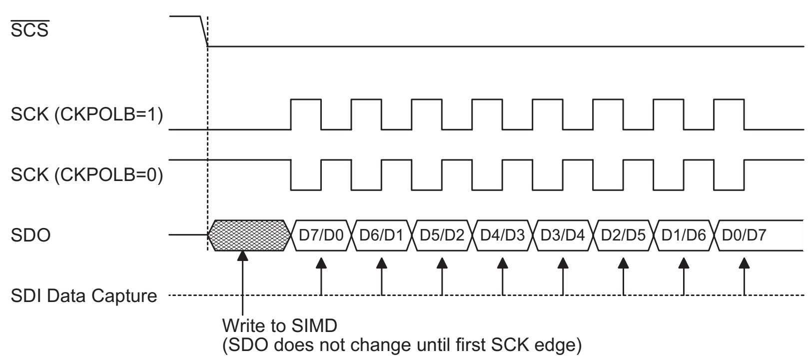
SPI Communication

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 devices 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.

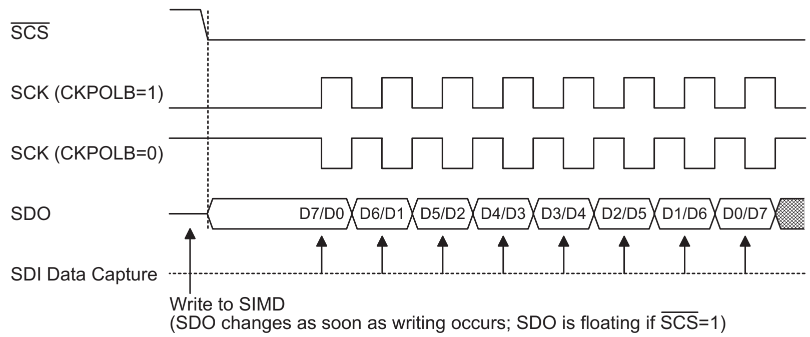
The SPI will continue to function even in the IDLE Mode.



SPI Master Mode Timing

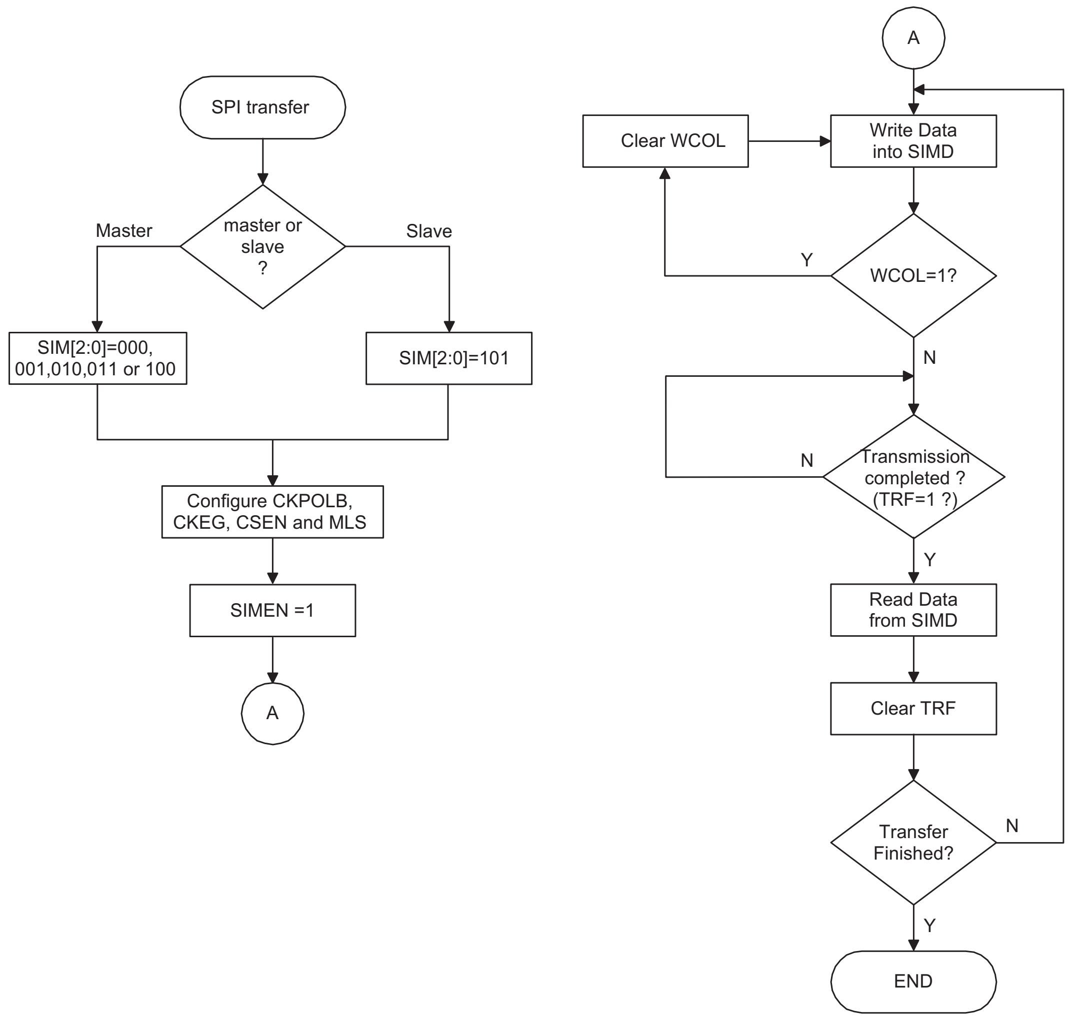


SPI Slave Mode Timing – CKEG=0



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

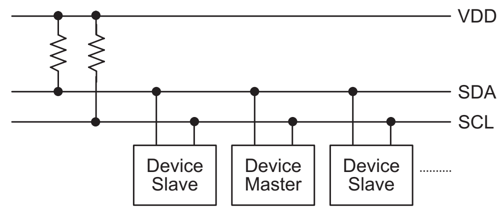
SPI Slave Mode Timing – CKEG=1



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.

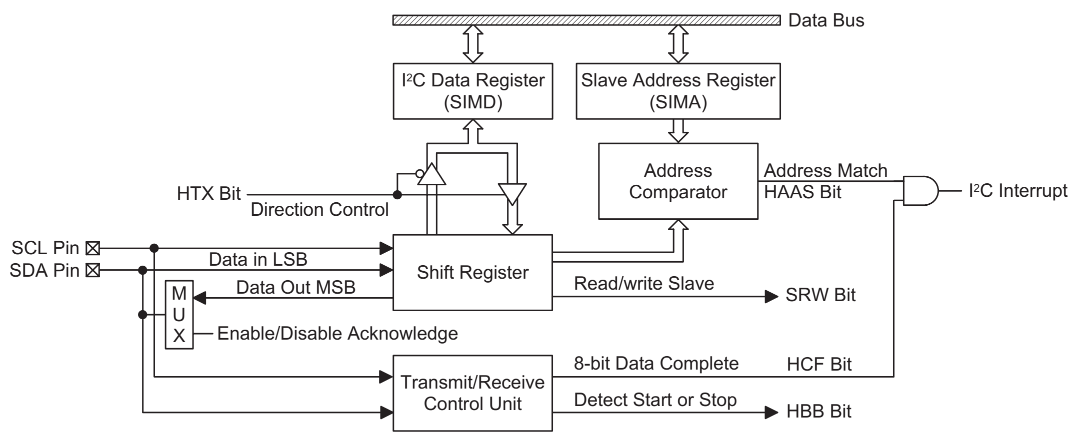


I²C Master Slave Bus Connection

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 operate in slave mode, there are two methods of transferring data on the I²C bus, the slave transmit mode and the slave receive mode.

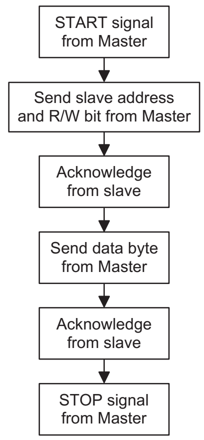


I²C Block Diagram

There are several configuration options associated with the I²C interface. One of these is to enable the function which selects the SIM pins rather than normal I/O pins. Note that if the configuration option does not select the SIM function then the SIMEN bit in the SIMC0 register will have no effect. A configuration option exists to allow a clock other than the system clock to drive the I²C interface. Another configuration option determines the debounce time of the I²C interface. This uses the internal clock to in effect add a debounce time to the external clock to reduce the possibility of glitches on the clock line causing erroneous operation. The debounce time, if selected, can be chosen to be either 2 or 4 system clocks. To achieve the required I²C data transfer speed, there exists a relationship between the system clock, f_{SYS}, and the I²C debounce time. For either the I²C Standard or Fast mode operation, users must take care of the selected system clock frequency and the configured debounce time to match the criterion shown in the following table.

I²C Debounce Time Selection	I²C Standard Mode (100kHz)	I²C Fast Mode (400kHz)
No debounce	f_{SYS} > 2MHz	f_{SYS} > 5MHz
2 system clock debounce	f_{SYS} > 4MHz	f_{SYS} > 10MHz
4 system clock debounce	f_{SYS} > 8MHz	f_{SYS} > 20MHz

I²C Minimum f_{SYS} Frequency



I²C Registers

There are three control registers associated with the I²C bus, SIMC0, SIMC1 and SIMA, 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.

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	HANS	HBB	HTX	TXAK	SRW	IAMWU	RXAK
SIMD	D7	D6	D5	D4	D3	D2	D1	D0
SIMA	IICA6	IICA5	IICA4	IICA3	IICA2	IICA1	IICA0	D0

I²C Registers 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_{TBC}
100: SPI master mode; SPI clock is TM0 CCRP match frequency/2
101: SPI slave mode
110: I^2C slave mode
111: Non SIM function

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
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 to zero to disable the SIM interface, the SDI, SDO, SCK and $\overline{SCS}$, or SDA and SCL lines will be in a floating condition and the SIM operating current will be reduced to a minimum value. When the bit is high the SIM interface is enabled. The SIM configuration option must have first enabled the SIM interface for this bit to be effective. If the SIM is configured to operate as an SPI interface via 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:** I²C 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:** I²C Bus address match flag
0: Not address match
1: Address match
The HAAS 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:** I²C Bus busy flag
0: I²C Bus is not busy
1: I²C Bus is busy
The HBB flag is the I²C busy flag. This flag will be "1" when the I²C 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 I²C slave device is transmitter or receiver
0: Slave device is the receiver
1: Slave device is the transmitter

Bit 3 **TXAK:** I²C 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:** I²C 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 I²C Slave Read/Write flag. This flag determines whether the master device wishes to transmit or receive data from the I²C 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:** I²C Address Match Wake-up Control
0: Disable
1: Enable - must be cleared by the application program after wake-up
This bit should be set to 1 to enable the I²C address match wake up from the SLEEP or IDLE Mode. If the IAMWU bit has been set before entering either the SLEEP or IDLE mode to enable the I²C address match wake up, then this bit must be cleared by the application program after wake-up to ensure correction device operation.

Bit 0 **RXAK:** I²C Bus Receive acknowledge flag
0: Slave receive acknowledge flag
1: Slave does 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 I²C Bus.

The SIMD register is used to store the data being transmitted and received. The same register is used by both the SPI and I²C functions. Before the devices write 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 devices 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

• SIMA Register

Bit	7	6	5	4	3	2	1	0
Name	IICA6	IICA5	IICA4	IICA3	IICA2	IICA1	IICA0	—
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 7~1 **IICA6~IICA0:** I²C slave address
IICA6~IICA0 is the I²C 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.

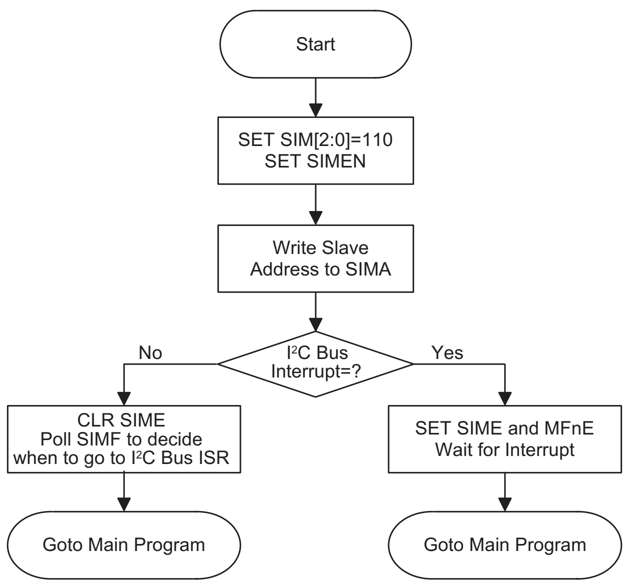
When a master device, which is connected to the I²C bus, sends out an address, which matches the slave address in the SIMA register, the slave device will be selected. Note that the SIMA register is the same register address as SIMC2 which is used by the SPI interface.

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.

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 devices 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.

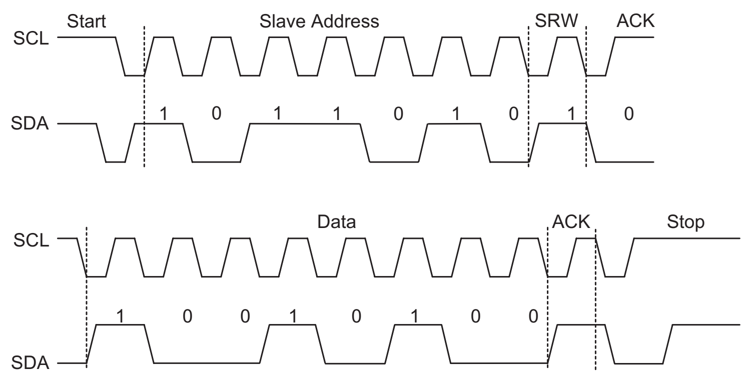
I²C Bus Slave Address Acknowledge Signal

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.

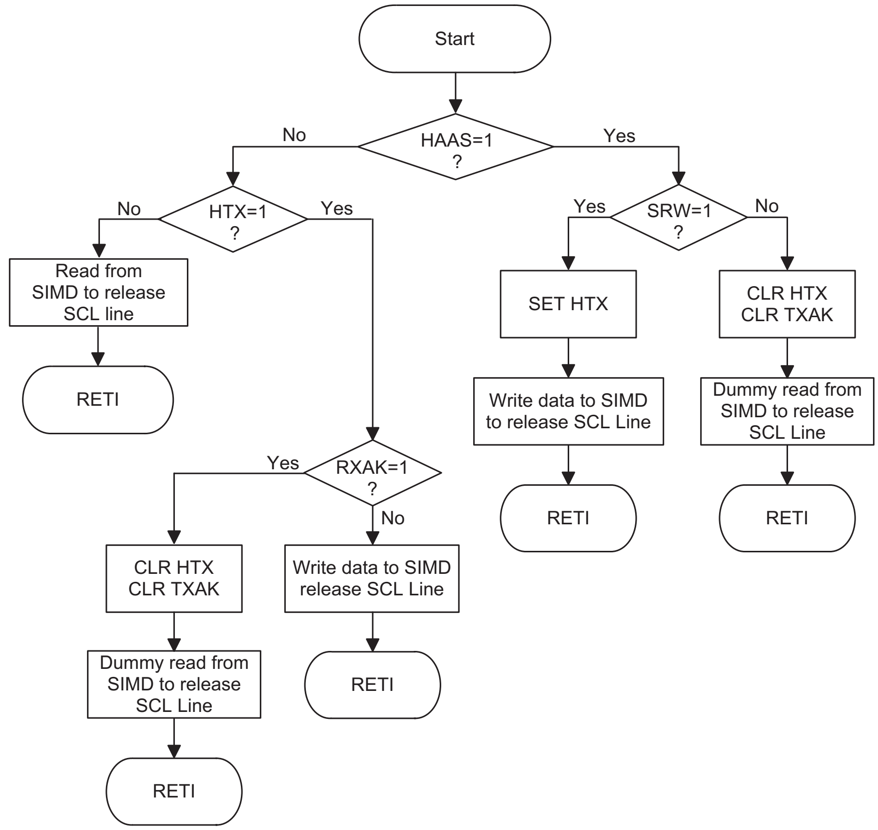


S=Start (1 bit)
SA=Slave Address (7 bits)
SR=SRW bit (1 bit)
M=Slave device send acknowledge bit (1 bit)
D=Data (8 bits)
A=ACK (RXAK bit for transmitter, TXAK bit for receiver 1 bit)
P=Stop (1 bit)

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 devices 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



I²C Bus ISR flow Chart

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	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_{TBC}
100: SPI master mode; SPI clock is TM0 CCRP match frequency/2
101: SPI slave mode
110: I^{2}C slave mode
111: Unused mode

These bits setup the overall operating mode of the SIM function. As well as selecting if the I^{2}C 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 to zero to disable the SIM interface, the SDI, SDO, SCK and $\overline{SCS}$, or SDA and SCL lines will be in a floating condition and the SIM operating current will be reduced to a minimum value. When the bit is high the SIM interface is enabled. The SIM configuration option must have first enabled the SIM interface for this bit to be effective. Note that when the SIMEN bit changes from low to high the contents of the SPI control registers will be in an unknown condition and should therefore be first initialised by the application program.

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 Timer Module or an A/D converter 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. These devices contain several external interrupt and internal interrupts functions. The external interrupts are generated by the action of the external INT0~INT1 and $\overline{\text{PINT}}$ pins, while the internal interrupts are generated by various internal functions such as the TMs, Comparators,Time Base, LVD, EEPROM, SIM and the A/D converter.

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~INTC2 registers which setup the primary interrupts, the second is the MFI0~MFI2 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	—	—
Comparator	CPnE	CPnF	n=0 or 1
INTn Pin	INTnE	INTnF	n=0~1
A/D Converter	ADE	ADF	HT66F20-1/HT66F30-1 only
Multi-function	MFnE	MFnF	n=0~3
Time Base	TBnE	TBnF	n=0 or 1
SIM	SIME	SIMF	—
LVD	LVE	LVF	—
EEPROM	DEE	DEF	—
$\overline{\text{PINT}}$ Pin	XPE	XPF	—
TM	TnPE	TnPF	n=0~1
	TnAE	TnAF	
	TnBE	TnBF	n=1

Interrupt Register Bit Naming Conventions

Interrupt Register Contents

- **HT66F20-1**

Name	Bit							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	CP0F	INT1F	INT0F	CP0E	INT1E	INT0E	EMI
INTC1	ADF	MF1F	MF0F	CP1F	ADE	MF1E	MF0E	CP1E
INTC2	MF3F	TB1F	TB0F	MF2F	MF3E	TB1E	TB0E	MF2E
MFI0	—	—	T0AF	T0PF	—	—	T0AE	T0PE
MFI1	—	—	T1AF	T1PF	—	—	T1AE	T1PE
MFI2	DEF	LVF	XPF	SIMF	DEE	LVE	XPE	SIME

- **HT66F30-1**

Name	Bit							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	CP0F	INT1F	INT0F	CP0E	INT1E	INT0E	EMI
INTC1	ADF	MF1F	MF0F	CP1F	ADE	MF1E	MF0E	CP1E
INTC2	MF3F	TB1F	TB0F	MF2F	MF3E	TB1E	TB0E	MF2E
MFI0	—	—	T0AF	T0PF	—	—	T0AE	T0PE
MFI1	—	T1BF	T1AF	T1PF	—	T1BE	T1AE	T1PE
MFI2	DEF	LVF	XPF	SIMF	DEE	LVE	XPE	SIME

- **HT68F20-1**

Name	Bit							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	CP0F	INT1F	INT0F	CP0E	INT1E	INT0E	EMI
INTC1	—	MF1F	MF0F	CP1F	—	MF1E	MF0E	CP1E
INTC2	MF3F	TB1F	TB0F	MF2F	MF3E	TB1E	TB0E	MF2E
MFI0	—	—	T0AF	T0PF	—	—	T0AE	T0PE
MFI1	—	—	T1AF	T1PF	—	—	T1AE	T1PE
MFI2	DEF	LVF	XPF	SIMF	DEE	LVE	XPE	SIME

- **HT68F30-1**

Name	Bit							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	CP0F	INT1F	INT0F	CP0E	INT1E	INT0E	EMI
INTC1	—	MF1F	MF0F	CP1F	—	MF1E	MF0E	CP1E
INTC2	MF3F	TB1F	TB0F	MF2F	MF3E	TB1E	TB0E	MF2E
MFI0	—	—	T0AF	T0PF	—	—	T0AE	T0PE
MFI1	—	T1BF	T1AF	T1PF	—	T1BE	T1AE	T1PE
MFI2	DEF	LVF	XPF	SIMF	DEE	LVE	XPE	SIME

INTEG Register

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

Bit 7~4 Unimplemented, read as "0”

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

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

INTC0 Register

Bit	7	6	5	4	3	2	1	0
Name	—	CP0F	INT1F	INT0F	CP0E	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 **CP0F:** Comparator 0 Interrupt Request Flag
0: No request
1: Interrupt request

Bit 5 **INT1F:** INT1 interrupt request flag
0: No request
1: Interrupt request

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

Bit 3 **CP0E:** Comparator 0 Interrupt Control
0: Disable
1: Enable

Bit 2 **INT1E:** INT1 interrupt control
0: Disable
1: Enable

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

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

INTC1 Register

- **HT66F20-1/HT66F30-1**

Bit	7	6	5	4	3	2	1	0
Name	ADF	MF1F	MF0F	CP1F	ADE	MF1E	MF0E	CP1E
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 **ADF:** A/D Converter Interrupt Request Flag
0: No request
1: Interrupt request

Bit 6 **MF1F:** Multi-function Interrupt 1 Request Flag
0: No request
1: Interrupt request

Bit 5 **MF0F:** Multi-function Interrupt 0 Request Flag
0: No request
1: Interrupt request

Bit 4 **CP1F:** Comparator 1 Interrupt Request Flag
0: No request
1: Interrupt request

Bit 3 **ADE:** A/D Converter Interrupt Interrupt Control
0: Disable
1: Enable

Bit 2 **MF1E:** Multi-function Interrupt 1 Control
0: Disable
1: Enable

Bit 1 **MF0E:** Multi-function Interrupt 0 Control
0: Disable
1: Enable

Bit 0 **CP1E:** Comparator 1 Interrupt Control
0: Disable
1: Enable

• **HT68F20-1/HT68F30-1**

Bit	7	6	5	4	3	2	1	0
Name	—	MF1F	MF0F	CP1F	—	MF1E	MF0E	CP1E
R/W	—	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	—	0	0	0	—	0	0	0

Bit 7 Unimplenented, read as "0"

Bit 6 **MF1F:** Multi-function Interrupt 1 Request Flag
0: No request
1: Interrupt request

Bit 5 **MF0F:** Multi-function Interrupt 0 Request Flag
0: No request
1: Interrupt request

Bit 4 **CP1F:** Comparator 1 Interrupt Request Flag
0: No request
1: Interrupt request

Bit 3 Unimplenented, read as "0"

Bit 2 **MF1E:** Multi-function Interrupt 1 Control
0: Disable
1: Enable

Bit 1 **MF0E:** Multi-function Interrupt 0 Control
0: Disable
1: Enable

Bit 0 **CP1E:** Comparator 1 Interrupt Control
0: Disable
1: Enable

INTC2 Register

Bit	7	6	5	4	3	2	1	0
Name	MF3F	TB1F	TB0F	MF2F	MF3E	TB1E	TB0E	MF2E
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 **TB1F:** Time Base 1 Interrupt Request Flag
0: No request
1: Interrupt request

Bit 5 **TB0F:** Time Base 0 Interrupt Request Flag
0: No request
1: Interrupt request

Bit 4 **MF2F:** Multi-function Interrupt 2 Request Flag
0: No request
1: Interrupt request

Bit 3 **MF3E:** Multi-function Interrupt 3 Control
0: Disable
1: Enable

Bit 2 **TB1E:** Time Base 1 Interrupt Control
0: Disable
1: Enable

Bit 1 **TB0E:** Time Base 0 Interrupt Control
0: Disable
1: Enable

Bit 0 **MF2E:** Multi-function Interrupt 2 Control
0: Disable
1: Enable

MFI0 Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	T0AF	T0PF	—	—	T0AE	T0PE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

Bit 7~6 Unimplemented, read as "0"

Bit 5 **T0AF:** TM0 Comparator A match interrupt request flag
0: no request
1: interrupt request

Bit 4 **T0PF:** TM0 Comparator P match interrupt request flag
0: no request
1: interrupt request

Bit 3~2 Unimplemented, read as "0"

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

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

MFI1 Register

- **HT66F20-1/HT68F20-1**

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

Bit 7~6 Unimplemented, read as "0"

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 3~2 Unimplemented, read as "0"

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

- **HT66F30-1/HT68F30-1**

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

MFI2 Register

Bit	7	6	5	4	3	2	1	0
Name	DEF	LVF	XPF	SIMF	DEE	LVE	XPE	SIME
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 **SIMF:** SIM 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 **SIME:** SIM Interrupt Control
0: Disable
1: Enable

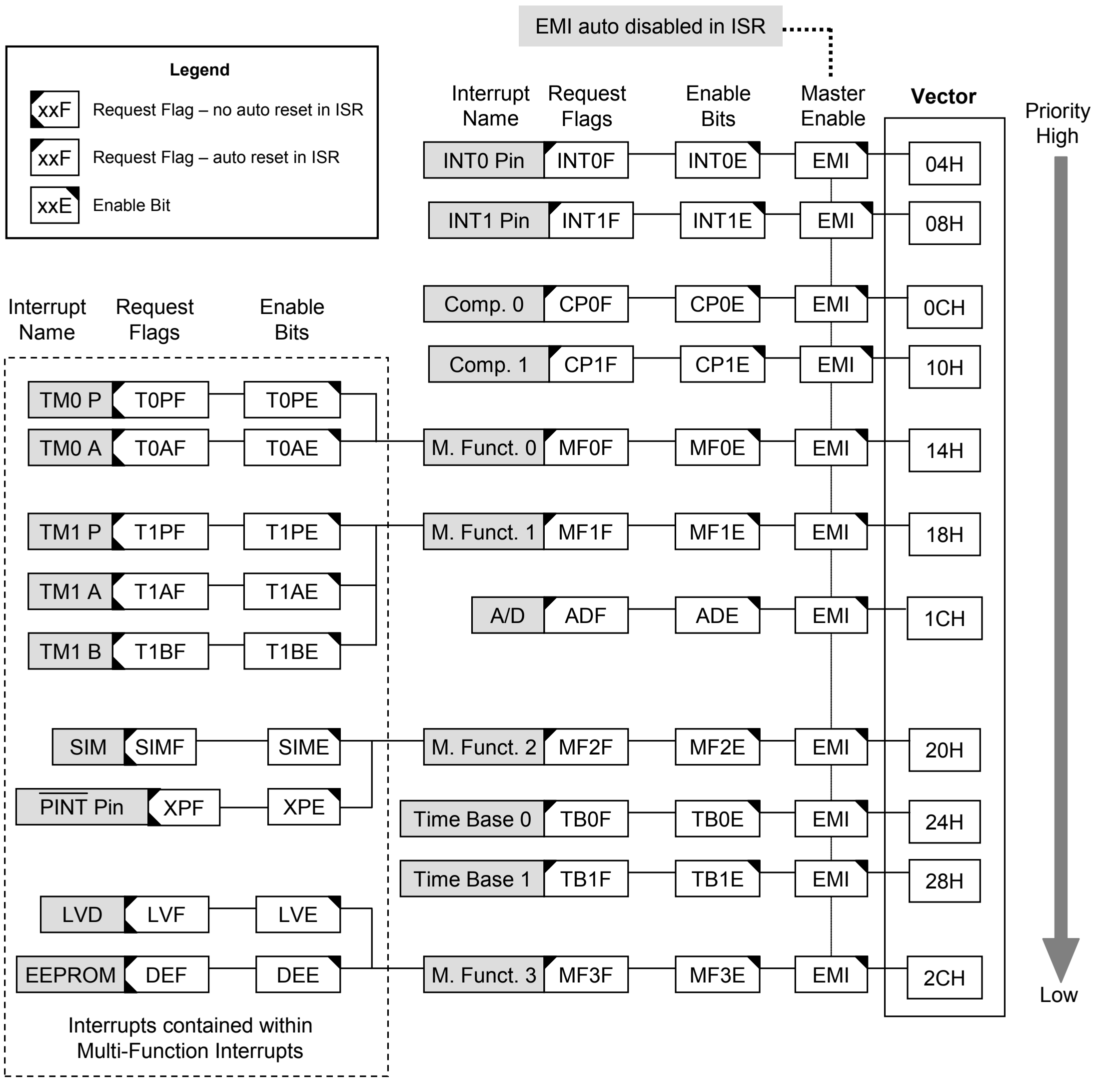
Interrupt Operation

When the conditions for an interrupt event occur, such as a TM Compare P, Compare A or Compare B match or A/D conversion completion 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.

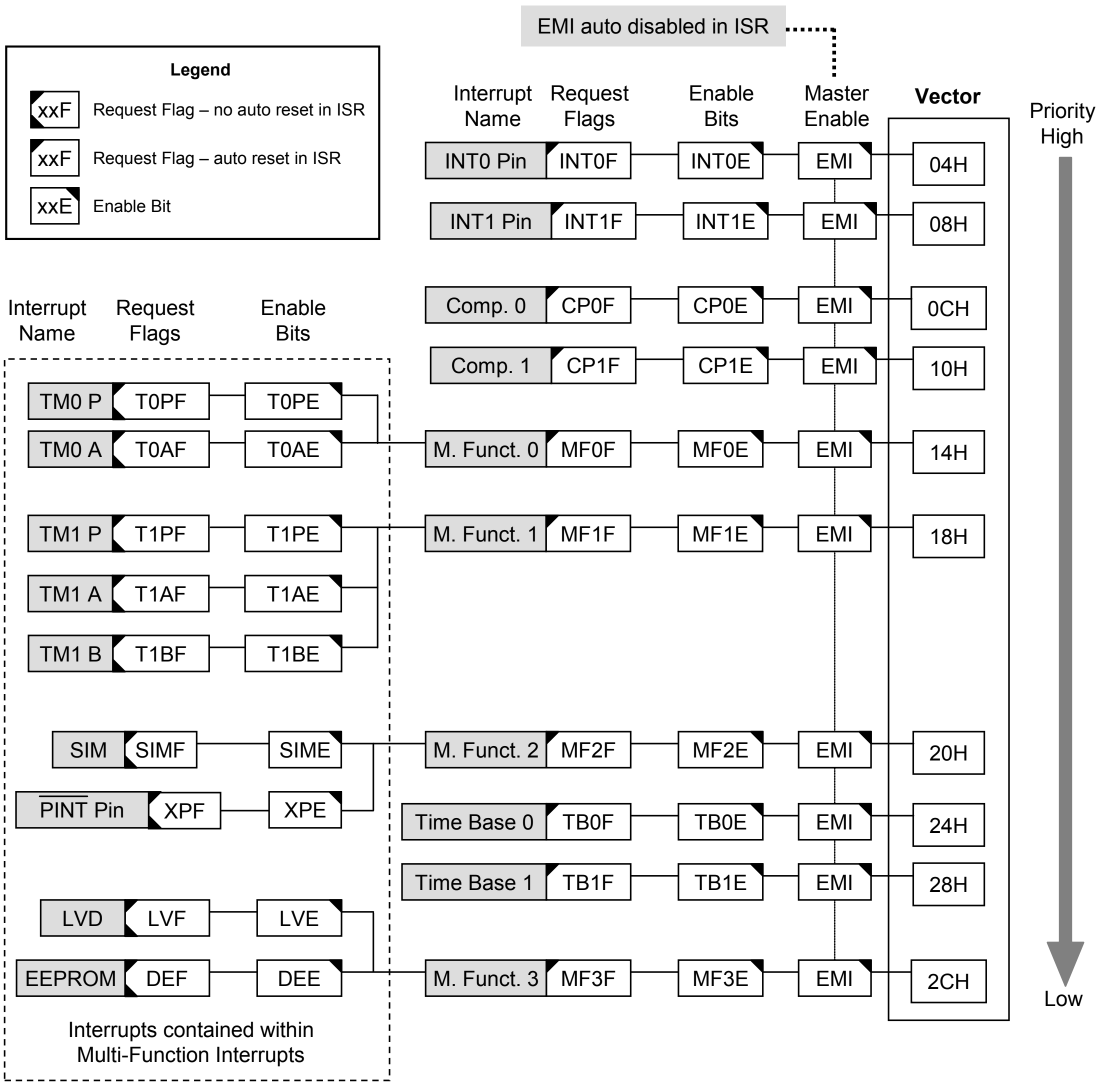
When an interrupt is generated, the Program Counter, which stores the address of the next instruction to be executed, will be transferred onto the stack. The Program Counter will then be loaded with a new address which will be the value of the corresponding interrupt vector. The microcontroller will then fetch its next instruction from this interrupt vector. The instruction at this vector will usually be a “JMP” 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”, 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 is in SLEEP or IDLE Mode.



Interrupt Structure – HT66F20-1/HT66F30-1



Interrupt Structure – HT68F20-1/HT68F30-1

External Interrupt

The external interrupts are controlled by signal transitions on the pins INT0~INT1. An external interrupt request will take place when the external interrupt request flags, INT0F~INT1F are set, which will occur when a transition, whose type is chosen by the edge select bits, appears on the external interrupt pins. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and respective external interrupt enable bit, INT0E~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 pins are pin-shared with I/O pins, they can only be configured as external interrupt pins if their 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 flags, INT0F~INT1F, 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 pins 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.

Comparator Interrupt

The comparator interrupts are controlled by the two internal comparators. A comparator interrupt request will take place when the comparator interrupt request flags, CP0F or CP1F, are set, a situation that will occur when the comparator output changes state. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and comparator interrupt enable bits, CP0E and CP1E, must first be set. When the interrupt is enabled, the stack is not full and the comparator inputs generate a comparator output transition, a subroutine call to the comparator interrupt vector, will take place. When the interrupt is serviced, the comparator interrupt request flags, CP0F and CP1F, will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts.

Multi-function Interrupt

Within these devices are four Multi-function interrupts. Unlike the other independent interrupts, these interrupts have no independent source, but rather are formed from other existing interrupt sources, namely the TM Interrupts, SIM Interrupt, External Peripheral Interrupt, LVD interrupt and EEPROM Interrupt.

A Multi-function interrupt request will take place when any of the Multi-function interrupt request flags, MF0F~MF3F 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 TM Interrupts, SIM Interrupt, External Peripheral Interrupt, LVD interrupt and EEPROM Interrupt will not be automatically reset and must be manually reset by the application program.

A/D Converter Interrupt

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

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.

TBC Register

Bit	7	6	5	4	3	2	1	0
Name	TBON	TBCK	TB11	TB10	LXTLP	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

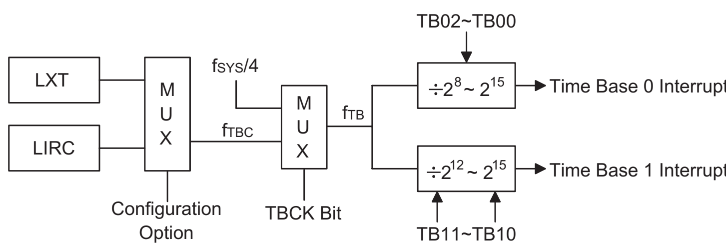
Bit 7 **TBON:** TB0 and TB1 Control
0: Disable
1: Enable

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
00: $4096/f_{TB}$
01: $8192/f_{TB}$
10: $16384/f_{TB}$
11: $32768/f_{TB}$

Bit 3 **LXTLP:** LXT Low Power Control
0: Disable
1: Enable

Bit 2~0 **TB02~TB00:** Select Time Base 0 Time-out Period
000: $256/f_{TB}$
001: $512/f_{TB}$
010: $1024/f_{TB}$
011: $2048/f_{TB}$
100: $4096/f_{TB}$
101: $8192/f_{TB}$
110: $16384/f_{TB}$
111: $32768/f_{TB}$



Time Base Interrupts

Serial Interface Module Interrupts

The Serial Interface Module Interrupt, also known as the SIM interrupt, is contained within the Multi-function 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, and Muti-function interrupt enable bits, 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 Multi-function Interrupt vector, will take place. When the Serial Interface 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 SIMF flag will not be automatically cleared, it has to be cleared by the application program.

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.

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 or read 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 or read 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.

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 TM each has two interrupts, while the Enhanced Type TM has three interrupts. All of the TM interrupts are contained within the Multi-function Interrupts. For the Compact and Standard Type TM 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.

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 these devices are 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, MF0F~MF3F, 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 the SLEEP or IDLE Mode, the wake up being generated when the interrupt request flag changes from low to high. If it is required to prevent a certain interrupt from waking up the microcontroller then its respective request flag should be first set high before enter SLEEP or IDLE Mode.

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

To return from an interrupt subroutine, either 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 determined. A low voltage condition is indicated when the LVDO bit is set. If the LVDO bit is low, this indicates that the V_{DD} voltage is above the preset low voltage value. The LVDEN bit is used to control the overall on/off function of the low voltage detector. Setting the bit high will enable the low voltage detector. Clearing the bit to zero will switch off the internal low voltage detector circuits. As the low voltage detector will consume a certain amount of power, it may be desirable to switch off the circuit when not in use, an important consideration in power sensitive battery powered applications.

LVDC Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	LVDO	LVDEN	—	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

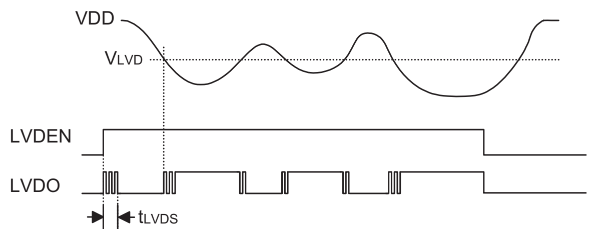
Bit 4 **LVDEN:** Low Voltage Detector Control
0: Disable
1: Enable

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: 4.4V

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 4.4V. 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.

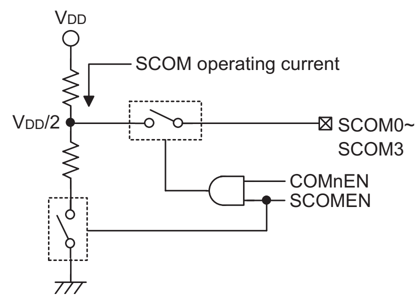
SCOM Function for LCD

The devices have the capability of driving external LCD panels. The common pins for LCD driving, SCOM0~SCOM3, are pin shared with certain pin on the PC0~PC1, PC6~PC7 port. The LCD signals (COM and SEG) are generated using the application program.

LCD Operation

An external LCD panel can be driven using this device by configuring the PC0~PC1, PC6~PC7 pins as common pins and using other output ports lines as segment pins. The LCD driver function is controlled using the SCOMC register which in addition to controlling the overall on/off function also controls the bias voltage setup function. This enables the LCD COM driver to generate the necessary $V_{DD}/2$ voltage levels for LCD 1/2 bias operation.

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



LCD COM Bias

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

Output Control

LCD Bias Control

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

SCOMC Register

- **HT66F30-1/HT68F30-1**

Bit	7	6	5	4	3	2	1	0
Name	D7	ISEL1	ISEL0	SCOMEN	COM3EN	COM2EN	COM1EN	COM0EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

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

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

Bit 4 **SCOMEN:** SCOM module Control
0: Disable
1: Enable

Bit 3 **COM3EN:** PC7 or SCOM3 selection
0: GPIO
1: SCOM3

Bit 2 **COM2EN:** PC6 or SCOM2 selection
0: GPIO
1: SCOM2

Bit 1 **COM1EN:** PC1 or SCOM1 selection
0: GPIO
1: SCOM1

Bit 0 **COM0EN:** PC0 or SCOM0 selection
0: GPIO
1: SCOM0

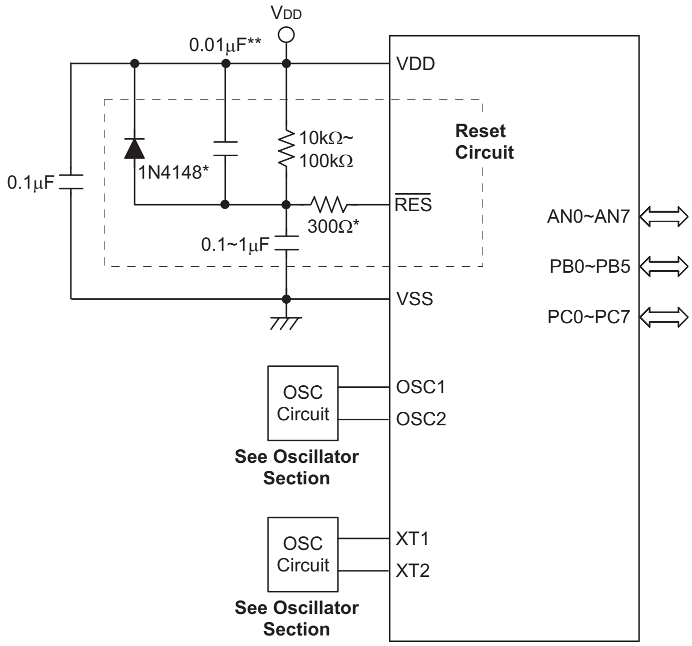
Configuration Options

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

No.	Options
Oscillator Options	
1	High Speed System Oscillator Selection - f_H: 1. HXT 2. ERC 3. HIRC
2	Low Speed System Oscillator Selection - f_{SUB}: 1. LXT 2. LIRC
3	WDT Clock Selection - f_S: 1. f_{SUB} 2. $f_{SYS}/4$
4	HIRC Frequency Selection: 1. 4MHz 2. 8MHz 3. 12MHz
Note: The f_{SUB} and the f_{TBC} clock source are LXT or LIRC selection by the f_L configuration option.	
Reset Pin Options	
5	PB0/$\overline{RES}$ Pin Options: 1. $\overline{RES}$ pin 2. I/O pin
Watchdog Options	
6	Watchdog Timer Function: 1. Enable 2. Disable
7	CLR WDT Instructions Selection: 1. 1 instructions 2. 2 instructions
LVR Options	
8	LVR Function: 1. Enable 2. Disable
9	LVR Voltage Selection: 1. 2.10V 2. 2.55V 3. 3.15V 4. 4.20V
SIM Options	
10	SIM Function: 1. Enable 2. Disable
11	SPI - WCOL bit: 1. Enable 2. Disable
12	SPI - CSEN bit: 1. Enable 2. Disable
13	I^2C Debounce Time Selection: 1. No debounce 2. 2 system clock debounce 3. 4 system clock debounce

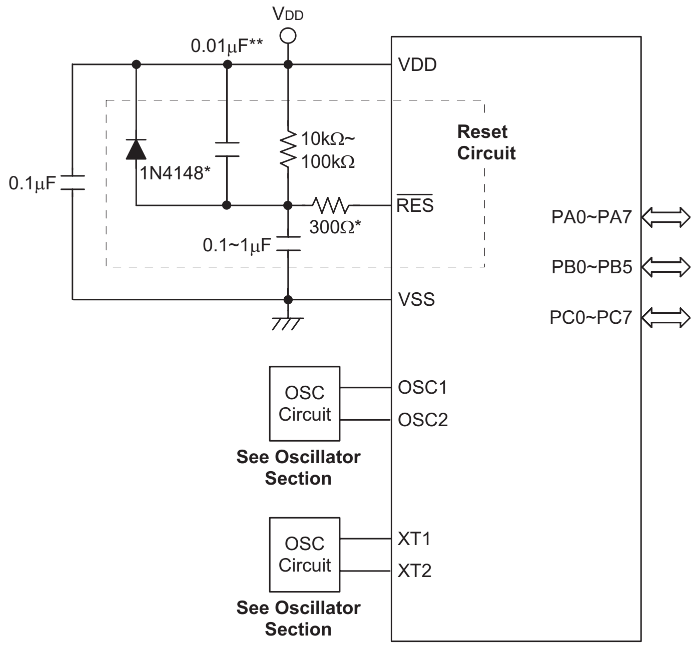
Application Circuits

HT66F20-1/HT66F30-1



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

HT68F20-1/HT68F30-1



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

Instruction Set

Introduction

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

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

Instruction Timing

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

Moving and Transferring Data

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

Arithmetic Operations

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

Logical and Rotate Operation

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

Branches and Control Transfer

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

Bit Operations

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

Table Read Operations

Data storage is normally implemented by using registers. However, when working with large amounts of fixed data, the volume involved often makes it inconvenient to store the fixed data in the Data Memory. To overcome this problem, Holtek microcontrollers allow an area of Program Memory to be 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

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

Note: 1. For skip instructions, if the result of the comparison involves a skip then two cycles are required, if no skip takes place only one cycle is required.

2. Any instruction which changes the contents of the PCL will also require 2 cycles for execution.

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 bit wise logical AND operation. The result is stored in the Accumulator.
Operation	ACC ← ACC ″AND″ x
Affected flag(s)	Z

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

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

CALL addr	Subroutine call
Description	Unconditionally calls a subroutine at the specified address. The Program Counter then increments by 1 to obtain the address of the next instruction which is then pushed onto the stack. The specified address is then loaded and the program continues execution from this new address. As this instruction requires an additional operation, it is a two cycle instruction.
Operation	Stack ← Program Counter + 1 Program Counter ← addr
Affected flag(s)	None

CLR [m]	Clear Data Memory
Description	Each bit of the specified Data Memory is cleared to 0.
Operation	[m] ← 00H
Affected flag(s)	None

CLR [m].i	Clear bit of Data Memory
Description	Bit i of the specified Data Memory is cleared to 0.
Operation	[m].i ← 0
Affected flag(s)	None

CLR WDT	Clear Watchdog Timer
Description	The TO, PDF flags and the WDT are all cleared.
Operation	WDT cleared TO ← 0 PDF ← 0
Affected flag(s)	TO, PDF

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 ← $\overline{[m]}$
Affected flag(s)	Z

DAA [m]	Decimal-Adjust ACC for addition with result in Data Memory
Description	Convert the contents of the Accumulator value to a BCD (Binary Coded Decimal) value resulting from the previous addition of two BCD variables. If the low nibble is greater than 9 or if AC flag is set, then a value of 6 will be added to the low nibble. Otherwise the low nibble remains unchanged. If the high nibble is greater than 9 or if the C flag is set, then a value of 6 will be added to the high nibble. Essentially, the decimal conversion is performed by adding 00H, 06H, 60H or 66H depending on the Accumulator and flag conditions. Only the C flag may be affected by this instruction which indicates that if the original BCD sum is greater than 100, it allows multiple precision decimal addition.
Operation	[m] ← ACC + 00H or [m] ← ACC + 06H or [m] ← ACC + 60H or [m] ← ACC + 66H
Affected flag(s)	C

DEC [m]	Decrement Data Memory
Description	Data in the specified Data Memory is decremented by 1.
Operation	[m] ← [m] − 1
Affected flag(s)	Z

DECA [m]	Decrement Data Memory with result in ACC
Description	Data in the specified Data Memory is decremented by 1. The result is stored in the Accumulator. The contents of the Data Memory remain unchanged.
Operation	ACC ← [m] − 1
Affected flag(s)	Z

HALT	Enter power down mode
Description	This instruction stops the program execution and turns off the system clock. The contents of the Data Memory and registers are retained. The WDT and prescaler are cleared. The power down flag PDF is set and the WDT time-out flag TO is cleared.
Operation	TO ← 0 PDF ← 1
Affected flag(s)	TO, PDF

INC [m]	Increment Data Memory
Description	Data in the specified Data Memory is incremented by 1.
Operation	[m] ← [m] + 1
Affected flag(s)	Z

INCA [m]	Increment Data Memory with result in ACC
Description	Data in the specified Data Memory is incremented by 1. The result is stored in the Accumulator. The contents of the Data Memory remain unchanged.
Operation	ACC ← [m] + 1
Affected flag(s)	Z

JMP addr	Jump unconditionally
Description	The contents of the Program Counter are replaced with the specified address. Program execution then continues from this new address. As this requires the insertion of a dummy instruction while the new address is loaded, it is a two cycle instruction.
Operation	Program Counter ← addr
Affected flag(s)	None

MOV A,[m]	Move Data Memory to ACC
Description	The contents of the specified Data Memory are copied to the Accumulator.
Operation	ACC ← [m]
Affected flag(s)	None

MOV A,x	Move immediate data to ACC
Description	The immediate data specified is loaded into the Accumulator.
Operation	ACC ← x
Affected flag(s)	None

MOV [m],A	Move ACC to Data Memory
Description	The contents of the Accumulator are copied to the specified Data Memory.
Operation	[m] ← ACC
Affected flag(s)	None

NOP	No operation
Description	No operation is performed. Execution continues with the next instruction.
Operation	No operation
Affected flag(s)	None

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

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

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

RET	Return from subroutine
Description	The Program Counter is restored from the stack. Program execution continues at the restored address.
Operation	Program Counter ← Stack
Affected flag(s)	None

RET A,x	Return from subroutine and load immediate data to ACC
Description	The Program Counter is restored from the stack and the Accumulator loaded with the specified immediate data. Program execution continues at the restored address.
Operation	Program Counter ← Stack ACC ← x
Affected flag(s)	None

RETI	Return from interrupt
Description	The Program Counter is restored from the stack and the interrupts are re-enabled by setting the EMI bit. EMI is the master interrupt global enable bit. If an interrupt was pending when the RETI instruction is executed, the pending Interrupt routine will be processed before returning to the main program.
Operation	Program Counter ← Stack EMI ← 1
Affected flag(s)	None

RL [m]	Rotate Data Memory left
Description	The contents of the specified Data Memory are rotated left by 1 bit with bit 7 rotated into bit 0.
Operation	[m].(i+1) ← [m].i; (i=0~6) [m].0 ← [m].7
Affected flag(s)	None

RLA [m]	Rotate Data Memory left with result in ACC
Description	The contents of the specified Data Memory are rotated left by 1 bit with bit 7 rotated into bit 0. The rotated result is stored in the Accumulator and the contents of the Data Memory remain unchanged.
Operation	ACC.(i+1) ← [m].i; (i=0~6) ACC.0 ← [m].7
Affected flag(s)	None

RLC [m]	Rotate Data Memory left through Carry
Description	The contents of the specified Data Memory and the carry flag are rotated left by 1 bit. Bit 7 replaces the Carry bit and the original carry flag is rotated into bit 0.
Operation	[m].(i+1) ← [m].i; (i=0~6) [m].0 ← C C ← [m].7
Affected flag(s)	C

RLCA [m]	Rotate Data Memory left through Carry with result in ACC
Description	Data in the specified Data Memory and the carry flag are rotated left by 1 bit. Bit 7 replaces the Carry bit and the original carry flag is rotated into the bit 0. The rotated result is stored in the Accumulator and the contents of the Data Memory remain unchanged.
Operation	ACC.(i+1) ← [m].i; (i=0~6) ACC.0 ← C C ← [m].7
Affected flag(s)	C

RR [m]	Rotate Data Memory right
Description	The contents of the specified Data Memory are rotated right by 1 bit with bit 0 rotated into bit 7.
Operation	[m].i ← [m].(i+1); (i=0~6) [m].7 ← [m].0
Affected flag(s)	None

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

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

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

Affected flag(s): None

RRC [m] Rotate Data Memory right through Carry

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

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

Affected flag(s): C

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

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

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

Affected flag(s): C

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

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

Operation: ACC ← ACC − [m] − $\overline{C}$

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

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

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

Operation: [m] ← ACC − [m] − $\overline{C}$

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

SDZ [m] Skip if decrement Data Memory is 0

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

Operation: [m] ← [m] − 1
Skip if [m]=0

Affected flag(s): None

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

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

Operation: ACC ← [m] − 1
Skip if ACC=0

Affected flag(s): None

SET [m] Set Data Memory

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

Operation: [m] ← FFH

Affected flag(s): None

SET [m].i Set bit of Data Memory

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

Operation: [m].i ← 1

Affected flag(s): None

SIZ [m] Skip if increment Data Memory is 0

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

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

Affected flag(s): None

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

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

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

Affected flag(s): None

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

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

Operation: Skip if [m].i ≠ 0

Affected flag(s): None

SUB A,[m] Subtract Data Memory from ACC

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

Operation: ACC ← ACC − [m]

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

SUBM A,[m]	Subtract Data Memory from ACC with result in Data Memory
Description	The specified Data Memory is subtracted from the contents of the Accumulator. The result is stored in the Data Memory. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.
Operation	[m] ← ACC − [m]
Affected flag(s)	OV, Z, AC, C

SUB A,x	Subtract immediate data from ACC
Description	The immediate data specified by the code is subtracted from the contents of the Accumulator. The result is stored in the Accumulator. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.
Operation	ACC ← ACC − x
Affected flag(s)	OV, Z, AC, C

SWAP [m]	Swap nibbles of Data Memory
Description	The low-order and high-order nibbles of the specified Data Memory are interchanged.
Operation	[m].3~[m].0 ↔ [m].7~[m].4
Affected flag(s)	None

SWAPA [m]	Swap nibbles of Data Memory with result in ACC
Description	The low-order and high-order nibbles of the specified Data Memory are interchanged. The result is stored in the Accumulator. The contents of the Data Memory remain unchanged.
Operation	ACC.3~ACC.0 ← [m].7~[m].4 ACC.7~ACC.4 ← [m].3~[m].0
Affected flag(s)	None

SZ [m]	Skip if Data Memory is 0
Description	If the contents of the specified Data Memory is 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.
Operation	Skip if [m]=0
Affected flag(s)	None

SZA [m]	Skip if Data Memory is 0 with data movement to ACC
Description	The contents of the specified Data Memory are copied to the Accumulator. If the value is zero, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.
Operation	ACC ← [m] Skip if [m]=0
Affected flag(s)	None

SZ [m].i	Skip if bit i of Data Memory is 0
Description	If bit i of the specified Data Memory is 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0, the program proceeds with the following instruction.
Operation	Skip if [m].i=0
Affected flag(s)	None

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

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

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

Affected flag(s): None

TABRDL [m] Read table (last page) to TBLH and Data Memory

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

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

Affected flag(s): None

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

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

Operation: ACC ← ACC ″XOR″ [m]

Affected flag(s): Z

XORM A,[m] Logical XOR ACC to Data Memory

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

Operation: [m] ← ACC ″XOR″ [m]

Affected flag(s): Z

XOR A,x Logical XOR immediate data to ACC

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

Operation: ACC ← ACC ″XOR″ x

Affected flag(s): Z

Package Information

Note that the package information provided here is for consultation purposes only. As this information may be updated at regular intervals users are reminded to consult the Holtek website for the latest version of the Package/Carton Information.

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

- Further Package Information (include Outline Dimensions, Product Tape and Reel Specifications)
- Packing Meterials Information
- Carton information

16-pin DIP (300mil) Outline Dimensions

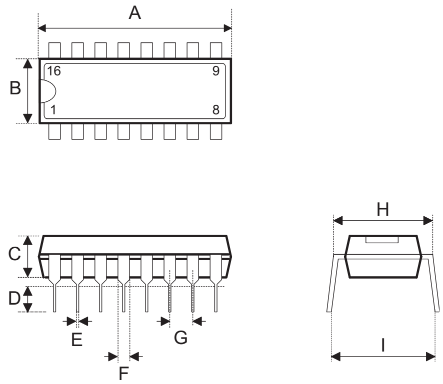


Fig1. Full Lead Packages

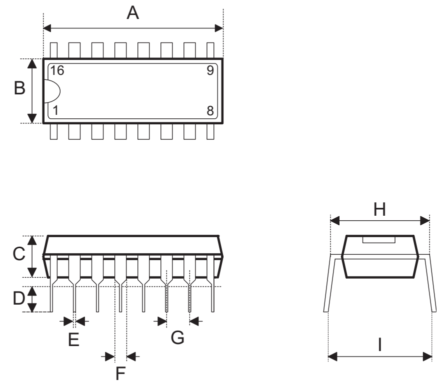


Fig1. 1/2 Lead Packages

Fig 1

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.780	0.790	0.800
B	0.240	0.250	0.280
C	0.115	0.130	0.195
D	0.115	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.060	0.070
G	—	0.100 BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	19.81	20.07	20.32
B	6.10	6.35	7.11
C	2.92	3.30	4.95
D	2.92	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.52	1.78
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

Fig 2

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.745	0.765	0.785
B	0.275	0.285	0.295
C	0.120	0.135	0.150
D	0.110	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.050	0.060
G	—	0.100 BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

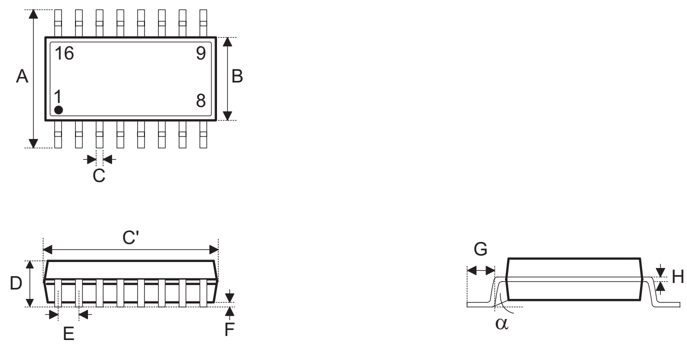
Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	18.92	19.43	19.94
B	6.99	7.24	7.49
C	3.05	3.43	3.81
D	2.79	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.27	1.52
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

Fig 2

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.735	0.755	0.775
B	0.240	0.250	0.280
C	0.115	0.130	0.195
D	0.115	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.060	0.070
G	—	0.100 BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	18.67	19.18	19.69
B	6.10	6.35	7.11
C	2.92	3.30	4.95
D	2.92	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.52	1.78
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

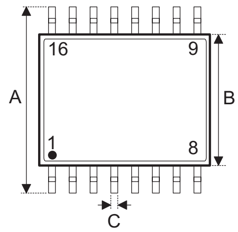
16-pin NSOP (150mil) Outline Dimensions

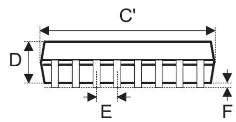


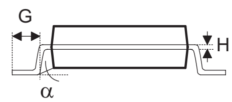
Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.236 BSC	—
B	—	0.154 BSC	—
C	0.012	—	0.020
C'	—	0.390 BSC	—
D	—	—	0.069
E	—	0.050 BSC	—
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	6.000 BSC	—
B	—	3.900 BSC	—
C	0.31	—	0.51
C'	—	9.900 BSC	—
D	—	—	1.75
E	—	1.270 BSC	—
F	0.10	—	0.25
G	0.40	—	1.27
H	0.10	—	0.25
α	0°	—	8°

16-pin SSOP (150mil) Outline Dimensions







Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.236 BSC	—
B	—	0.154 BSC	—
C	0.008	—	0.012
C’	—	0.193 BSC	—
D	—	—	0.069
E	—	0.025 BSC	—
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	6.000 BSC	—
B	—	3.900 BSC	—
C	0.20	—	0.30
C’	—	4.900 BSC	—
D	—	—	1.75
E	—	0.635 BSC	—
F	0.10	—	0.25
G	0.41	—	1.27
H	0.10	—	0.25
α	0°	—	8°

20-pin DIP (300mil) Outline Dimensions

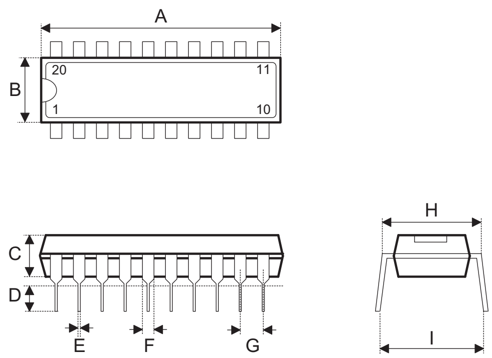


Fig1. Full Lead Packages

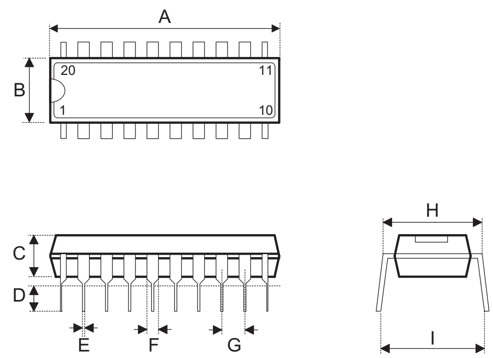


Fig2. 1/2 Lead Packages

See Fig1

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.980	1.030	1.060
B	0.240	0.250	0.280
C	0.115	0.130	0.195
D	0.115	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.060	0.070
G	—	0.100 BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

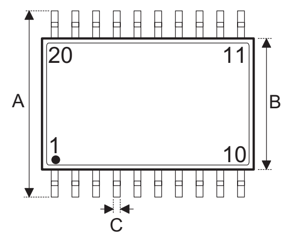
Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	24.89	26.16	26.92
B	6.10	6.35	7.11
C	2.92	3.30	4.95
D	2.92	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.52	1.78
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

See Fig 2

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.945	0.965	0.985
B	0.275	0.285	0.295
C	0.120	0.135	0.150
D	0.110	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.050	0.060
G	—	0.100 BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	24.00	24.51	25.02
B	6.99	7.24	7.49
C	3.05	3.43	3.81
D	2.79	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.27	1.52
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

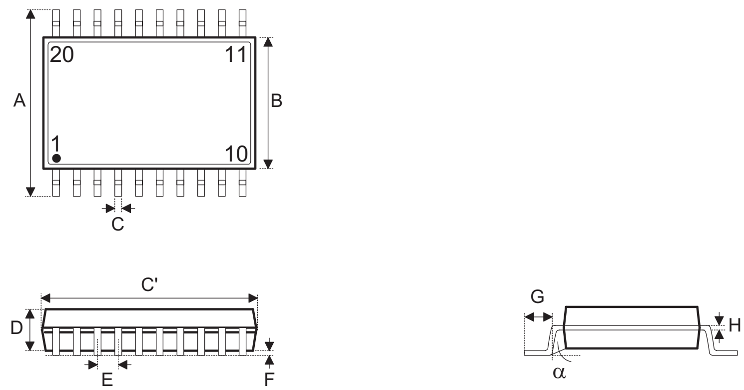
20-pin SOP (300mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.406 BSC	—
B	—	0.295 BSC	—
C	0.012	—	0.020
C’	—	0.504 BSC	—
D	—	—	0.104
E	—	0.050 BSC	—
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	—	10.30 BSC	—
B	—	7.50 BSC	—
C	0.31	—	0.51
C’	—	12.80 BSC	—
D	—	—	2.65
E	—	1.27 BSC	—
F	0.10	—	0.30
G	0.40	—	1.27
H	0.20	—	0.33
α	0°	—	8°

20-pin SSOP (150mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.236 BSC	—
B	—	0.155 BSC	—
C	0.008	—	0.012
C’	—	0.341 BSC	—
D	—	—	0.069
E	—	0.025 BSC	—
F	0.004	—	0.0098
G	0.016	—	0.05
H	0.004	—	0.01
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	6.000 BSC	—
B	—	3.900 BSC	—
C	0.20	—	0.30
C’	—	8.660 BSC	—
D	—	—	1.75
E	—	0.635 BSC	—
F	0.10	—	0.25
G	0.41	—	1.27
H	0.10	—	0.25
α	0°	—	8°

24-pin SKDIP (300mil) Outline Dimensions

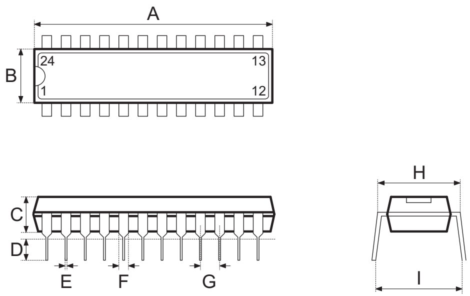


Fig1. Full Lead Packages

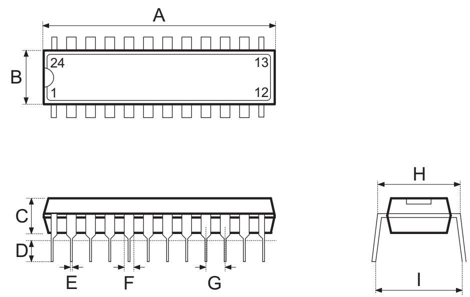


Fig2. 1/2 Lead Packages

See Fig1

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	1.230	1.250	1.280
B	0.240	0.250	0.280
C	0.115	0.130	0.195
D	0.115	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.060	0.070
G	—	0.100 BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	31.24	31.75	32.51
B	6.10	6.35	7.11
C	2.92	3.30	4.95
D	2.92	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.52	1.78
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

See Fig2

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	1.160	1.185	1.195
B	0.240	0.250	0.280
C	0.115	0.130	0.195
D	0.115	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.060	0.070
G	—	0.100 BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

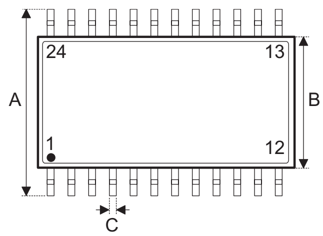
Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	29.46	30.10	30.35
B	6.10	6.35	7.11
C	2.92	3.30	4.95
D	2.92	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.52	1.78
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

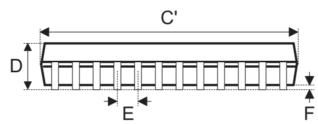
See fig20

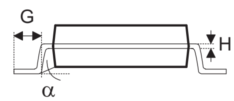
Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	1.145	1.165	1.185
B	0.275	0.285	0.295
C	0.120	0.135	0.150
D	0.110	0.130	0.150
E	0.014	0.018	0.022
F	0.045	0.050	0.060
G	—	0.100 BSC	—
H	0.300	0.310	0.325
I	—	—	0.430

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	29.08	29.59	30.10
B	6.99	7.24	7.49
C	3.05	3.43	3.81
D	2.79	3.30	3.81
E	0.36	0.46	0.56
F	1.14	1.27	1.52
G	—	2.54 BSC	—
H	7.62	7.87	8.26
I	—	—	10.92

24-pin SOP (300mil) Outline Dimensions



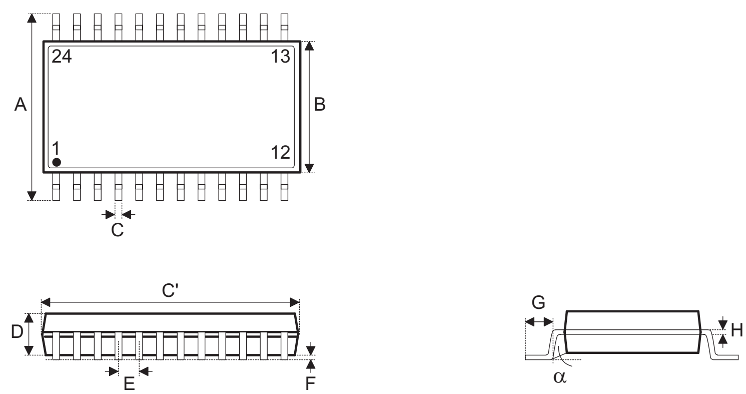




Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.406 BSC	—
B	—	0.295 BSC	—
C	0.012	—	0.020
C’	—	0.606 BSC	—
D	—	—	0.104
E	—	0.050 BSC	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	10.30 BSC	—
B	—	7.50 BSC	—
C	0.31	—	0.51
C’	—	15.40 BSC	—
D	—	—	2.65
E	—	1.27 BSC	—
F	0.10	—	0.30
G	0.40	—	1.27
H	0.20	—	0.33
α	0°	—	8°

24-pin SSOP(150mil) Outline Dimensions



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

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	6.000 BSC	—
B	—	3.900 BSC	—
C	0.20	—	0.30
C’	—	8.660 BSC	—
D	—	—	1.75
E	—	0.635 BSC	—
F	0.10	—	0.25
G	0.41	—	1.27
H	0.10	—	0.25
α	0°	—	8°