HOLTEK

Enhanced A/D Flash MCU with High Current LED Driver

HT66F24D/HT66F25D/HT66F26D

www.holtek.com

Note that the HT66F24D/HT66F26D devices, although mentioned in this datasheet, have already been phased out and are presently no longer available.

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
- Three Oscillator types
  - External high frequency Crystal – HXT
  - 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
- 61 powerful instructions
- 8-level subroutine nesting
- Bit manipulation instruction

Peripheral Features

- Program Memory: 2K×16 ~ 8K×16
- Data Memory: 96×8 ~ 384×8
- True EEPROM Memory: 64×8
- Watchdog Timer function
- Up to 26 bidirectional I/O lines
- Multiple pin-shared external interrupts
- Multiple Timer Module for time measure, input capture, compare match output, PWM output or single pulse output function
- Dual Time-Base functions for generation of fixed time interrupt signal
- Up to 8 channels 12-bit A/D converter
- Low voltage reset function
- Low voltage detect function
- LED Driver
- Wide range of available package types

General Description

The series of devices are Flash Memory A/D type 8-bit high performance RISC architecture microcontrollers. 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 feature includes a multi-channel 12-bit A/D converter. Multiple and extremely flexible Timer Modules provide timing, pulse generation and PWM generation functions. Protective features such as an internal Watchdog Timer, Low Voltage Reset and Low Voltage Detector coupled with excellent noise immunity and ESD protection ensure that reliable operation is maintained in hostile electrical environments.

A full choice of HXT, 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.
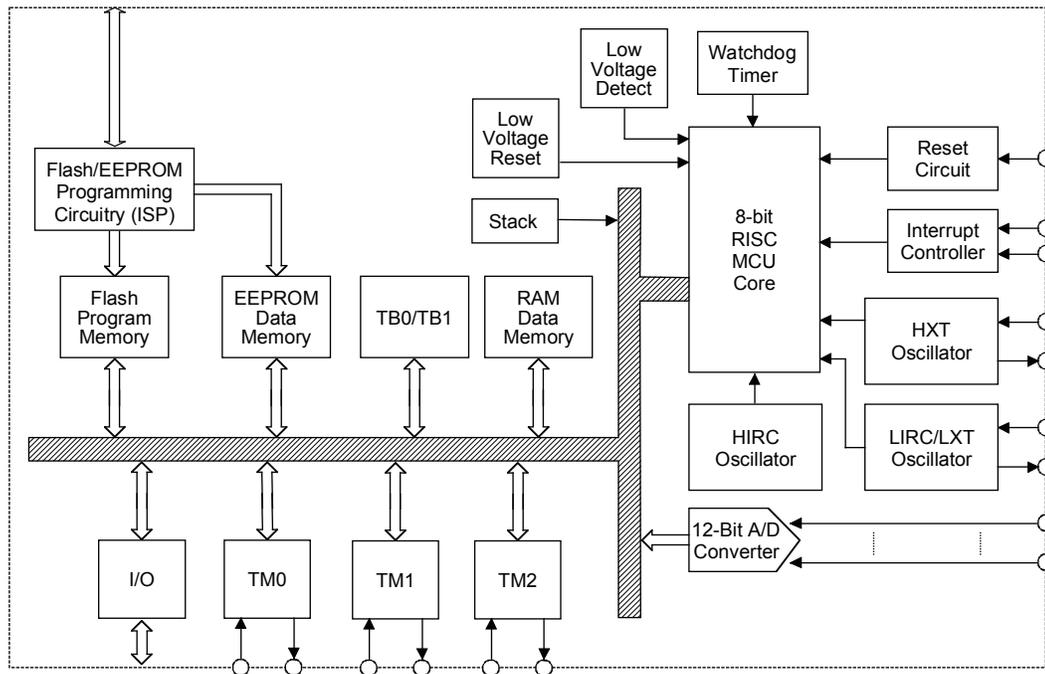
Selection Table

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

Part No.	V_{DD}	Program Memory	Data Memory	Data EEPROM	I/O	External Interrupt
HT66F24D	2.2V~5.5V	2K×16	96×8	64×8	22	2
HT66F25D	2.2V~5.5V	4K×16	192×8	64×8	26	2
HT66F26D	2.2V~5.5V	8k×16	384×8	64×8	26	2

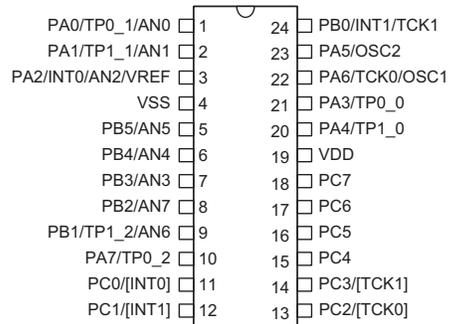| Part No.	A/D Converter	Timer Module	LED Driver	Stacks	Package
HT66F24D	12-bit×8	10-bit CTM×1 10-bit STM×1	8×6	8	16NSOP 20/24SOP
HT66F25D	12-bit×8	10-bit CTM×1 10-bit ETM×1	8×8	8	24/28SOP
HT66F26D	12-bit×8	10-bit CTM×1 10-bit STM×1 10-bit ETM×1	8×8	8	24/28SOP

Note: As devices exist in more than one package format, the table reflects the situation for the package with the most pins.
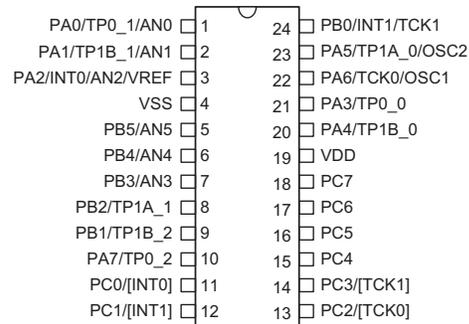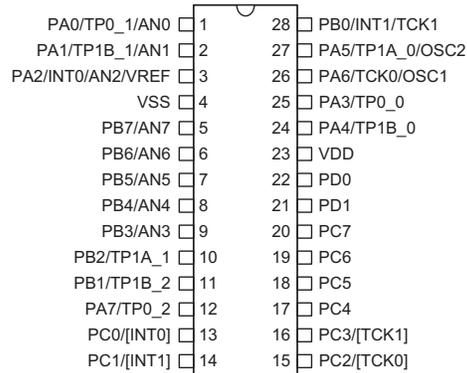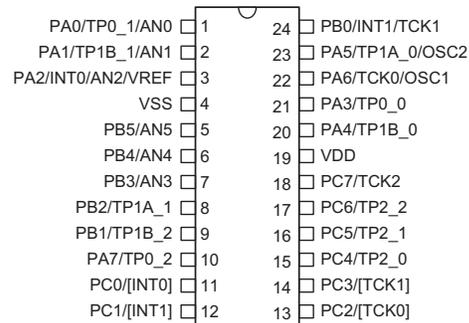
Block Diagram

Pin Assignment

HT66F24D
16 SOP-A
LED 3x3 (Source/Sink)

HT66F24D
20 SOP-A
LED 6x4 (Source/Sink)

PA0/TP0_1/AN0	1	24	PB0/INT1/TCK1
PA1/TP1_1/AN1	2	23	PA5/OSC2
PA2/INT0/AN2/VREF	3	22	PA6/TCK0/OSC1
VSS	4	21	PA3/TP0_0
PB5/AN5	5	20	PA4/TP1_0
PB4/AN4	6	19	VDD
PB3/AN3	7	18	PC7
PB2/AN7	8	17	PC6
PB1/TP1_2/AN6	9	16	PC5
PA7/TP0_2	10	15	PC4
PC0/[INT0]	11	14	PC3/[TCK1]
PC1/[INT1]	12	13	PC2/[TCK0]

HT66F24D
24 SOP-A
LED 8x6 (Source/Sink)

PA0/TP0_1/AN0	1	24	PB0/INT1/TCK1
PA1/TP1B_1/AN1	2	23	PA5/TP1A_0/OSC2
PA2/INT0/AN2/VREF	3	22	PA6/TCK0/OSC1
VSS	4	21	PA3/TP0_0
PB5/AN5	5	20	PA4/TP1B_0
PB4/AN4	6	19	VDD
PB3/AN3	7	18	PC7
PB2/TP1A_1	8	17	PC6
PB1/TP1B_2	9	16	PC5
PA7/TP0_2	10	15	PC4
PC0/[INT0]	11	14	PC3/[TCK1]
PC1/[INT1]	12	13	PC2/[TCK0]

HT66F25D
24 SOP-A
LED 8x6 (Source/Sink)

PA0/TP0_1/AN0	1	28	PB0/INT1/TCK1
PA1/TP1B_1/AN1	2	27	PA5/TP1A_0/OSC2
PA2/INT0/AN2/VREF	3	26	PA6/TCK0/OSC1
VSS	4	25	PA3/TP0_0
PB7/AN7	5	24	PA4/TP1B_0
PB6/AN6	6	23	VDD
PB5/AN5	7	22	PD0
PB4/AN4	8	21	PD1
PB3/AN3	9	20	PC7
PB2/TP1A_1	10	19	PC6
PB1/TP1B_2	11	18	PC5
PA7/TP0_2	12	17	PC4
PC0/[INT0]	13	16	PC3/[TCK1]
PC1/[INT1]	14	15	PC2/[TCK0]

HT66F25D
28 SOP-A
LED 8x8 (Source/Sink)

PA0/TP0_1/AN0	1	24	PB0/INT1/TCK1
PA1/TP1B_1/AN1	2	23	PA5/TP1A_0/OSC2
PA2/INT0/AN2/VREF	3	22	PA6/TCK0/OSC1
VSS	4	21	PA3/TP0_0
PB5/AN5	5	20	PA4/TP1B_0
PB4/AN4	6	19	VDD
PB3/AN3	7	18	PC7/TCK2
PB2/TP1A_1	8	17	PC6/TP2_2
PB1/TP1B_2	9	16	PC5/TP2_1
PA7/TP0_2	10	15	PC4/TP2_0
PC0/[INT0]	11	14	PC3/[TCK1]
PC1/[INT1]	12	13	PC2/[TCK0]

HT66F26D
24 SOP-A
LED 8x6 (Source/Sink)

PA0/TP0_1/AN0	1	28	PB0/INT1/TCK1
PA1/TP1B_1/AN1	2	27	PA5/TP1A_0/OSC2
PA2/INT0/AN2/VREF	3	26	PA6/TCK0/OSC1
VSS	4	25	PA3/TP0_0
PB7/AN7	5	24	PA4/TP1B_0
PB6/AN6	6	23	VDD
PB5/AN5	7	22	PD0
PB4/AN4	8	21	PD1
PB3/AN3	9	20	PC7/TCK2
PB2/TP1A_1	10	19	PC6/TP2_2
PB1/TP1B_2	11	18	PC5/TP2_1
PA7/TP0_2	12	17	PC4/TP2_0
PC0/[INT0]	13	16	PC3/[TCK1]
PC1/[INT1]	14	15	PC2/[TCK0]

HT66F26D
28 SOP-A
LED 8x8 (Source/Sink)

PA0/TP0_1/AN0/OCDSDA	1	28	PB0/INT1/TCK1/OCDSCK
PA1/TP1B_1/AN1	2	27	PA5/TP1A_0/OSC2
PA2/INT0/AN2/VREF	3	26	PA6/TCK0/OSC1
VSS	4	25	PA3/TP0_0
PB7/AN7	5	24	PA4/TP1B_0
PB6/AN6	6	23	VDD
PB5/AN5	7	22	PD0
PB4/AN4	8	21	PD1
PB3/AN3	9	20	PC7/TCK2
PB2/TP1A_1	10	19	PC6/TP2_2
PB1/TP1B_2	11	18	PC5/TP2_1
PA7/TP0_2	12	17	PC4/TP2_0
PC0/[INT0]	13	16	PC3/[TCK1]
PC1/[INT1]	14	15	PC2/[TCK0]

HT66V26D
28 SOP-A
LED 8x8 (Source/Sink)

Pin Descriptions

With the exception of the power pins, all pins on these devices can be referenced by their Port name, e.g. PA.0, PA.1 etc, which refer to the digital I/O function of the pins. However some of these Port pins are also shared with other function such as the Analog to Digital Converter, Timer Module pins etc. 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.

• **HT66F24D**

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	A/D Converter input	ACERL	AN	—	PA0~PA2,PB3~PB5, PB1~PB2
VREF	A/D Converter reference input	ADCR1	AN	—	PA2
TCK0	TM0 input	PRM0	ST	—	PA6, PC2
TCK1	TM1 input	PRM0	ST	—	PB0, PC3
TP0_0, TP0_1, TP0_2	TM0 I/O	TMPC0	ST	CMOS	PA3, PA0, PA7
TP1_0, TP1_1, TP1_2	TM1 I/O	TMPC0	ST	CMOS	PA4, PA1, PB1
INT0	External Interrupt 0	PRM0	ST	—	PA2, PC0
INT1	External Interrupt 1	PRM0	ST	—	PB0, PC1
OSC1	HXT pin	CO	HXT	—	PA6
OSC2	HXT pin	CO	—	HXT	PA5
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;   AN: Analog input pin;

HXT: High frequency crystal oscillator.

• **HT66F25D**

Pin Name	Function	OP	I/T	O/T	Pin-Shared Mapping
PA0~PA7	Port A	PAWU PAPU	ST	CMOS	—
PB0~PB7	Port B	PBPU	ST	CMOS	—
PC0~PC7	Port C	PCPU	ST	CMOS	—
PD0~PD1	Port D	PDPU	ST	CMOS	—
AN0~AN7	A/D Converter input	ACERL	AN	—	PA0~PA2, PB3~PB7
VREF	A/D Converter reference input	ADCR1	AN	—	PA2
TCK0	TM0 input	PRM0	ST	—	PA6, PC2
TCK1	TM1 input	PRM0	ST	—	PB0, PC3
TP0_0, TP0_1, TP0_2	TM0 I/O	TMPC0	ST	CMOS	PA3, PA0, PA7
TP1A_0, TP1A_1	TM1 I/O	TMPC0	ST	CMOS	PA5, PB2

Pin Name	Function	OP	I/T	O/T	Pin-Shared Mapping
TP1B_0,TP1B_1, TP1B_2	TM1 I/O	TMPC0	ST	CMOS	PA4, PA1, PB1
INT0	External Interrupt 0	PRM0	ST	—	PA2, PC0
INT1	External Interrupt 1	PRM0	ST	—	PB0, PC1
OSC1	HXT pin	CO	HXT	—	PA6
OSC2	HXT pin	CO	—	HXT	PA5
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

AN: Analog input pin

HXT: High frequency crystal oscillator

- **HT66F26D**

Pin Name	Function	OP	I/T	O/T	Pin-Shared Mapping
PA0~PA7	Port A	PAWU PAPU	ST	CMOS	—
PB0~PB7	Port B	PBPU	ST	CMOS	—
PC0~PC7	Port C	PCPU	ST	CMOS	—
PD0~PD1	Port D	PDPU	ST	CMOS	—
OCDSDA	OCDS Data/Address, for EV chip only.	—	ST	CMOS	PA0
OCDSCK	OCDS Clock pin, for EV chip only.	—	ST	—	PB0
AN0~AN7	A/D Converter input	ACERL	AN	—	PA0~PA2, PB3~PB7
VREF	A/D Converter reference input	ADCR1	AN	—	PA2
TCK0	TM0 input	PRM0	ST	—	PA6, PC2
TCK1	TM1 input	PRM0	ST	—	PB0, PC3
TCK2	TM2 input	—	ST		PC7
TP0_0, TP0_1, TP0_2	TM0 I/O	TMPC0	ST	CMOS	PA3, PA0, PA7
TP1A_0, TP1A_1	TM1 I/O	TMPC0	ST	CMOS	PA5, PB2
TP1B_0, TP1B_1, TP1B_2	TM1 I/O	TMPC0	ST	CMOS	PA4, PA1, PB1
TP2_0, TP2_1, TP2_2	TM2 I/O	TMPC0	ST	CMOS	PC4, PC5, PC6
INT0	External Interrupt 0	PRM0	ST	—	PA2, PC0
INT1	External Interrupt 1	PRM0	ST	—	PB0, PC1
OSC1	HXT pin	CO	HXT	—	PA6
OSC2	HXT pin	CO	—	HXT	PA5
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

AN: Analog input pin

HXT: High frequency crystal oscillator

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 ....-60°C to 150°C

Operating Temperature ....-40°C to 85°C

I_{OL} Total 100mA

I_{OH} Total ....-100mA

Total Power Dissipation 500mW

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

D.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage (HXT/HIRC OSC)	—	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, f_s=f_{SUB}=f_{LIRC} (HXT, HIRC)	3V	No load, f_H=4MHz, ADC off, WDT enable	—	0.7	1.1	mA
		5V		—	1.8	2.7	mA
		3V	No load, f_H=8MHz, ADC off, WDT enable	—	1.6	2.4	mA
		5V		—	3.3	5.0	mA
		3V	No load, 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, f_s=f_{SUB}=f_{LIRC} (HXT)	5V	No load, f_H=20MHz, ADC off, WDT enable	—	6.0	9.0	mA
I_{DD3}	Operating Current, Slow Mode, f_{SYS}=f_L (LIRC)	3V	No load, f_{SYS}=LIRC, ADC off, WDT enable	—	10	20	µA
		5V		—	20	40	µA
I_{IDLE1}	IDLE0 Mode Stanby Current (LIRC On)	3V	No load, ADC off, WDT enable	—	1.5	3.0	µA
		5V		—	3.0	6.0	µA
I_{IDLE2}	IDLE1 Mode Stanby Current (HXT, HIRC)	3V	No load, ADC off, WDT enable, f_{SYS} =12MHz on	—	0.55	0.83	mA
		5V		—	1.30	2.00	mA
I_{SLEEP1}	SLEEP0 Mode Stanby Current (LIRC Off)	3V	No load, ADC off, WDT disable	—	—	1.0	µA
		5V		—	—	2.0	µA
I_{SLEEP2}	SLEEP1 Mode Stanby Current (LIRC On)	3V	No load, ADC off, 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	—	—	0	—	0.3V_{DD}	V
V_{IH1}	Input High Voltage for I/O Ports or Input Pins	—	—	0.7V_{DD}	—	V_{DD}	V

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{OL1}	I/O Port Sink Current (PA0~PA6, PB0, D0~PD1)	3V	V_{OL}= 0.1V_{DD}	4	8	—	mA
		5V	V_{OL}= 0.1V_{DD}	10	20	—	mA
I_{OH1}	I/O Port, Source Current (PA0~PA6, PB0, D0~PD1)	3V	V_{OH}= 0.9V_{DD}	-2	-4	—	mA
		5V	V_{OH}= 0.9V_{DD}	-5	-10	—	mA
I_{OL2}	I/O Port Sink Current (PA7, PB1~PB7)	3V	V_{OL}= 0.1V_{DD}	16	32	—	mA
		5V	V_{OL}= 0.1V_{DD}	40	80	—	mA
I_{OH2}	I/O Port, Source Current (PA7, PB1~PB7)	3V	V_{OH}= 0.9V_{DD}	-2	-4	—	mA
		5V	V_{OH}= 0.9V_{DD}	-5	-10	—	mA
I_{OL3}	I/O Port Sink Current (PC0~PC7)	3V	V_{OL}= 0.1V_{DD}	4	8	—	mA
		5V	V_{OL}= 0.1V_{DD}	10	20	—	mA
I_{OH3}	I/O Port, Source Current (PC0~PC7)	3V	V_{OH}= 0.9V_{DD}	-4	-8	—	mA
		5V	V_{OH}= 0.9V_{DD}	-10	-20	—	mA
R_{PH}	Pull-high Resistance of I/O Ports	3V	—	20	60	100	kΩ
		5V	—	10	30	50	kΩ

A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
f_{CPU}	Operating Clock	2.2~5.5V	—	DC	—	8	MHz
		2.7~5.5V	—	DC	—	12	MHz
		4.5~5.5V	—	DC	—	20	MHz
f_{SYS}	System clock (HXT)	2.2~5.5V	—	0.4	—	8	MHz
		2.7~5.5V	—	0.4	—	12	MHz
		4.5~5.5V	—	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°C~70°C	-5%	4	+5%	MHz
		3V/5V	Ta=0°C~70°C	-4%	8	+4%	MHz
		5V	Ta=0°C~70°C	-5%	12	+3%	MHz
		2.2V~3.6V	Ta=0°C~70°C	-7%	4	+7%	MHz
		3.0V~5.5V	Ta=0°C~70°C	-5%	4	+9%	MHz
		2.2V~3.6V	Ta=0°C~70°C	-6%	8	+4%	MHz
		3.0V~5.5V	Ta=0°C~70°C	-4%	8	+9%	MHz
		3.0V~5.5V	Ta=0°C~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}	Condition				
f_{LIRC}	System Clock (LIRC)	5V	Ta = 25°C	-10%	32	+10%	kHz
		2.2V~5.5V	Ta = -40°C to 85°C	-30%	32	+60%	kHz
t_{TIMER}	TCKn Input Pin Pulse Width	—	—	0.3	—	—	µs
t_{INT}	Interrupt Pulse Width	—	—	10	—	—	µs
t_{SST}	System Start-up Timer Period (Wake-up from HALT, f_{SYS} off at HALT state)	—	f_{SYS} = HXT	128	—	—	t_{SYS}
		—	f_{SYS} = HIRC	16	—	—	t_{SYS}
		—	f_{SYS} = LIRC	2	—	—	t_{SYS}
	System Start-up Timer Period (Wake-up from HALT, f_{SYS} on at HALT state)	—	—	2	—	—	t_{SYS}
t_{RSTD}	System Reset Delay Time (Power On Reset)	—	—	25	50	100	ms
	System Reset Delay Time (Any Reset except Power On Reset)	—	—	8.3	16.7	33.3	ms
t_{EERD}	EEPROM Read Time	—	f_{SYS} = 20MHz	—	—	4	t_{SYS}
t_{EEWR}	EEPROM Write Time	—	—	1	2	4	ms

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

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

LVD and LVR Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{LVR1}	Low Voltage Reset Voltage	—	LVR Enable, V_{LVR} = 2.1V	-5%× Typ.	2.1	+5%× Typ.	V
V_{LVR2}		—	LVR Enable, V_{LVR} = 2.55V		2.55		V
V_{LVR3}		—	LVR Enable, V_{LVR} = 3.15V		3.15		V
V_{LVR4}		—	LVR Enable, V_{LVR} = 3.8V		3.8		V
V_{LVD1}	Low Voltage Detector Voltage	—	LVDEN = 1, V_{LVD} = 2.0V	-5%× Typ.	2.0	+5%× Typ.	V
V_{LVD2}		—	LVDEN = 1, V_{LVD} = 2.2V		2.2		V
V_{LVD3}		—	LVDEN = 1, V_{LVD} = 2.4V		2.4		V
V_{LVD4}		—	LVDEN = 1, V_{LVD} = 2.7V		2.7		V
V_{LVD5}		—	LVDEN = 1, V_{LVD} = 3.0V		3.0		V
V_{LVD6}		—	LVDEN = 1, V_{LVD} = 3.3V		3.3		V
V_{LVD7}		—	LVDEN = 1, V_{LVD} = 3.6V		3.6		V
V_{LVD8}		—	LVDEN = 1, V_{LVD} = 4.0V		4.0		V
I_{LVR}	Additional Power Consumption	3V	—	—	30	45	µA
		5V		—	60	90	µA

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{LVD}	Additional Power Consumption if LVD is used	3V	LVD disable → LVD enable (LVR enable)	—	30	45	µA
		5V		—	60	90	µA
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	—	For LVR enable, LVD off→on	15	—	—	µs
		—	For LVR disable, LVD off→on	15	—	—	µs
t_{SRESET}	Software Reset Width to Reset	—	—	45	90	120	µs

A/D Converter Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	A/D Converter Operating Voltage	—	—	2.2	—	5.5	V
V_{ADI}	A/D Converter Input Voltage	—	—	0	—	V_{REF}	V
V_{REF}	A/D Converter Reference Voltage	—	—	2	—	V_{DD}	V
V_{BG}	Reference Voltage with Buffer Voltage	—	—	-3%	1.25	+3%	V
DNL1	Differential Non-linearity	2.2V~2.7V	$V_{REF}=V_{DD}$, $t_{ADCK}=8µs$	—	±15	—	LSB
		2.7V~5.5V	$V_{REF}=V_{DD}$, $t_{ADCK}=0.5µs$	-3	—	+3	LSB
DNL2	Differential Non-linearity	2.7V	$V_{REF}=V_{DD}$, $t_{ADCK}=0.5µs$, Ta=-40°C~85°C	-4	—	+4	LSB
		3V					
		5V					
INL1	Integral Non-linearity	2.2V~2.7V	$V_{REF}=V_{DD}$, $t_{ADCK}=8µs$	—	±16	—	LSB
		2.7V~5.5V	$V_{REF}=V_{DD}$, $t_{ADCK}=0.5µs$	-4	—	+4	LSB
INL2	Integral Non-linearity	2.7V	$V_{REF}=V_{DD}$, $t_{ADCK}=0.5µs$, Ta=-40°C~85°C	-8	—	+8	LSB
		3V					
		5V					
I_{ADC}	Additional Power Consumption if A/D Converter is Used	3V	No load, $t_{ADCK}=0.5µs$	—	0.9	1.35	mA
		5V	No load, $t_{ADCK}=0.5µs$	—	1.2	1.8	mA
t_{ADCK}	A/D Converter Clock Period	2.2V~2.7V	—	8	—	10	µs
		2.7V~5.5V	—	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
t_{BGS}	V_{BG} Turn on Stable Time	—	—	200	—	—	µs

Power-on Reset Characteristics

Ta=25°C

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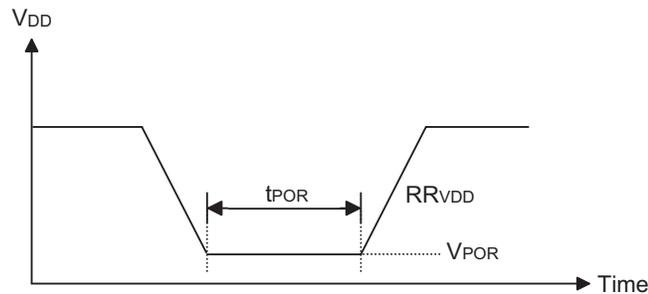
System Architecture

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 the device suitable for low-cost, high-volume production for controller applications.

Clocking and Pipelining

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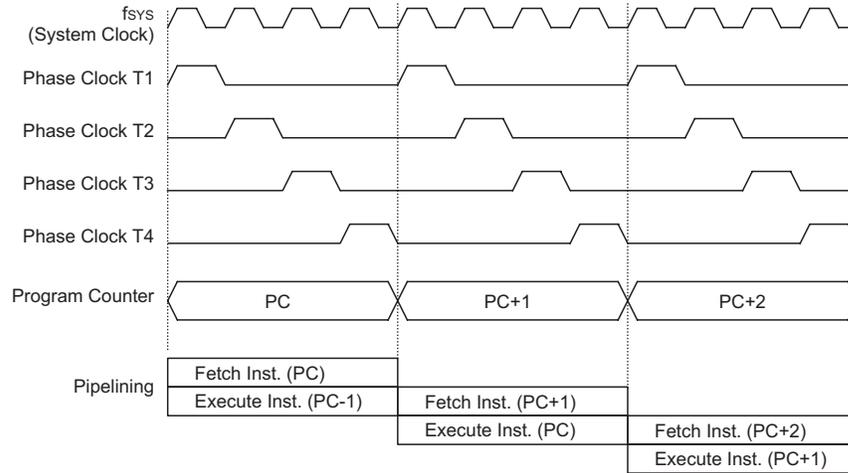
System Clocking and Pipelining
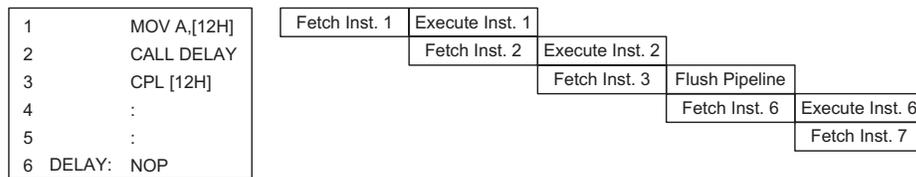
Instruction Fetching

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

Program Counter

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

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

Device	Program Counter	
	High Byte	**Low Byte (PCL Register)**
HT66F24D	PC10~PC8	PCL7~PCL0
HT66F25D	PC11~PC8	PCL7~PCL0
HT66F26D	PC12~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 multiple levels depending upon the device and is neither part of the data nor part of the program space, and is neither readable nor writeable. The activated level is indexed by the Stack Pointer, and is neither readable nor writeable. At a subroutine call or interrupt acknowledge signal, the contents of the Program Counter are pushed onto the stack. At the end of a subroutine or an interrupt routine, signaled by a return instruction, RET or RETI, the Program Counter is restored to its previous value from the stack. After a device reset, the Stack Pointer will point to the top of the stack.

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

If the stack is overflow, the first Program Counter save in the stack will be lost.
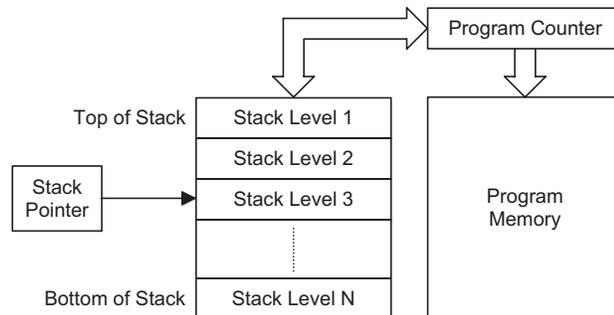
Device	Stack Levels
HT66F24D	8
HT66F25D	8
HT66F26D	8

Arithmetic and Logic Unit – ALU

The arithmetic-logic unit or ALU is a critical area of the that carries out arithmetic and logic operations of the instruction set. Connected to the main 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, , ORM, XORM, CPL, CPLA
- Rotation RRA, RR, RRCA, RRC, RLA, RL, RLCA, RLC
- Increment and Decrement INCA, INC, DECA, DEC
- Branch decision, JMP, SZ, SZA, SNZ, SIZ, SDZ, SIZA, SDZA, CALL, RET, RETI

Flash Program Memory

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

Structure

The Program Memory has a capacity of 2Kx16 bits to 8Kx16 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
HT66F24D	2K × 16
HT66F25D	4K × 16
HT66F26D	8K × 16
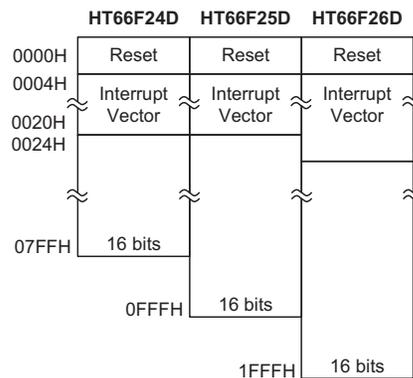
Program Memory Structure

Special Vectors

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

Look-up Table

Any location within the Program Memory can be defined as a look-up table where programmers can store fixed data. To use the look-up table, the table pointer must first be setup by placing the address of the look up data to be retrieved in the table pointer register, TBLP and TBHP. These registers define the total address of the look-up table.

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

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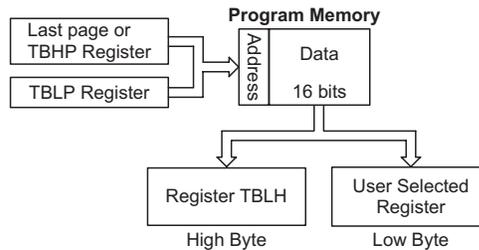
Table Program Example

The following example 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 HT66F24D device. The table pointer is setup here to have an initial value of "06H". This will ensure that the first data read from the data table will be at the Program Memory address "706H" or 6 locations after the start of the last page. Note that the value for the table pointer is referenced to the first address of the present page if the "TABRD [m]" instruction is being used. The high byte of the table data which in this case is equal to zero will be transferred to the TBLH register automatically when the "TABRD [m]" instruction is executed.

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

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
mov  tblp,a      ; is referenced
mov  a,07h       ; initialise high table pointer
mov  tbhp,a
       :
       :
tabrd tempreg1 ; transfers value in table referenced by table pointer data at
               ; program memory address 706H transferred to tempreg1 and TBLH

dec  tblp        ; reduce value of table pointer by one
tabrd tempreg2 ; transfers value in table referenced by table pointer data at
               ; program memory address 705H transferred to tempreg2 and TBLH
               ; in this example the data 1AH is transferred to tempreg1 and
               ; data 0FH to register tempreg2
:
:
org 700h         ; sets initial address of program memory

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

In Circuit Programming

The provision of Flash type Program Memory provides the user with a means of convenient and easy upgrades and modifications to their programs on the same device. As an additional convenience, Holtek has provided a means of programming the microcontroller in-circuit using a 5-pin interface. This provides manufacturers with the possibility of manufacturing their circuit boards complete with a programmed or un-programmed microcontroller, and then programming or upgrading the program at a later stage. This enables product manufacturers to easily keep their manufactured products supplied with the latest program releases without removal and re-insertion of the device.

Holtek Writer Pins	MCU Programming Pins	Pin Description
ICPDA	PA0	Programming Serial Data
ICPCK	PA2	Programming Clock
ICPMS	PB0	Programming Mode Select
VDD	VDD	Power Supply
VSS	VSS	Ground

The Program Memory can be programmed serially in-circuit using this 5-wire interface. Data is downloaded and uploaded serially on a single pin with an additional line for the clock. Two additional lines are required for the power supply and one line for the reset. The technical details regarding the in-circuit programming of the devices are beyond the scope of this document and will be supplied in supplementary literature.

During the programming process the ICPMS 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.

Writer Connector Signals

MCU Programming Pins

VDD

ICPMS

ICPDA

ICPCK

VSS

VDD

PB0

PA0

PA2

VSS

* * *

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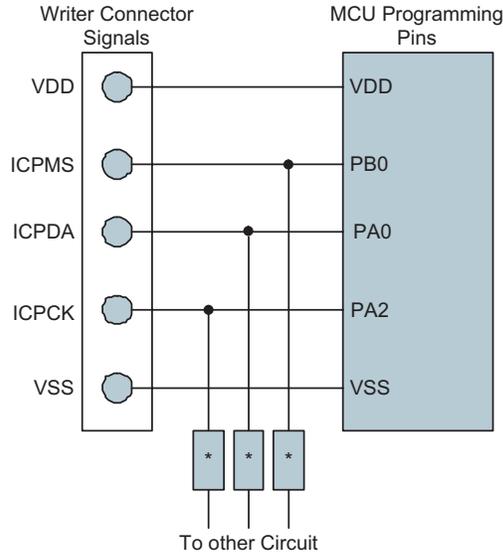
Note: * may be resistor or capacitor. The resistance of * must be greater than 1kΩ or the capacitance of * must be less than 1nF.

On-Chip Debug Support – OCDS

There is an EV chip named HT66V26D which is used to emulate the HT66F2xD series of devices. The HT66V26D device also provides the "On-Chip Debug" function to debug the HT66F2xD series of devices during development process. The devices, HT66F2xD and HT66V26D, are almost functional compatible except the "On-Chip Debug" function and package types. Users can use the HT66V26D device to emulate the HT66F2xD series of devices behaviors by connecting the OCDSDA and OCDSCK pins to the Holtek HT-IDE development tools. The OCDSDA pin is the OCDS Data/Address input/output pin while the OCDSCK pin is the OCDS clock input pin. When users use the HT66V26D EV chip for debugging, the corresponding pin functions shared with the OCDSDA and OCDSCK pins in the HT66F2xD series of devices will have no effect in the HT66V26D EV chip. However, the two OCDS pins which are pin-shared with the ICP programming pins are still used as the Flash Memory programming pins for ICP. For more detailed OCDS information, refer to the corresponding document named "Holtek e-Link for 8-bit MCU OCDS User's Guide".

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

RAM Data Memory

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

Structure

Divided into two sections, the first of these is an area of RAM, known as the Special Function Data Memory. Here are located registers which are necessary for correct operation of the device. Many of these registers can be read from and written to directly under program control, however, some remain protected from user manipulation.

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

The overall Data Memory is subdivided into up to three banks, the structure of which depends upon the device chosen. The Special Purpose Data Memory registers are accessible in all banks. Switching between the different Data Memory banks is achieved by setting the Bank Pointer to the correct value. The start address of the Special Purpose Data Memory for all devices is the address 00H while the start address of the General Purpose Data Memory is the address 40H or 80H for different chosen device.
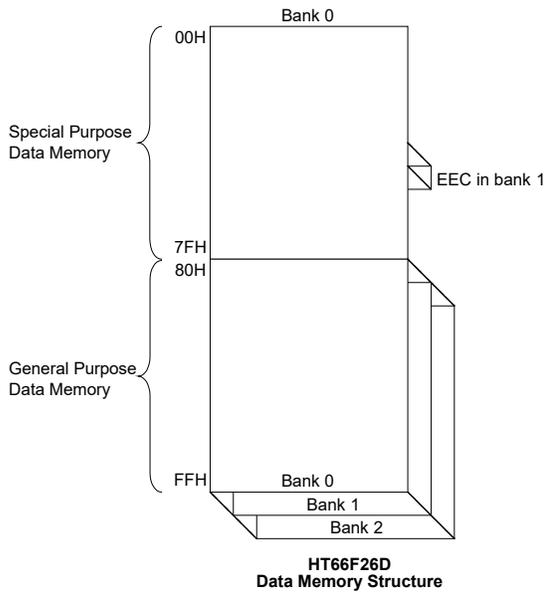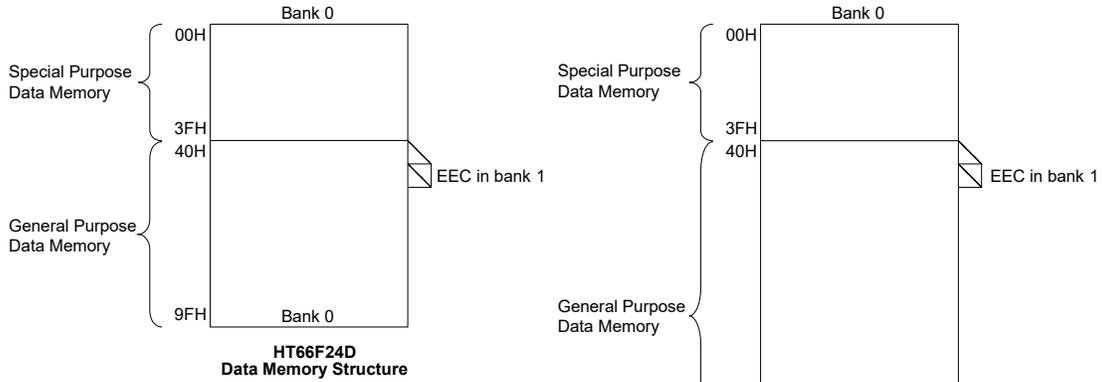
RAM Data Memory Structure

Device	Capacity	Bank : Address
HT66F24D	96 × 8	Bank 0 : 40H ~ 9FH
HT66F25D	192 × 8	Bank 0 : 40H ~ FFH
HT66F26D	384 × 8	Bank 0 : 80H ~ FFH Bank 1 : 80H ~ FFH Bank 2 : 80H ~ FFH

General Purpose Data Memory Structure

	Bank 0	Bank 1		Bank 0	Bank 1
00H	IAR0		22H	Unused	
01H	MP0		24H		
02H	IAR1		25H	PRM0	
03H	MP1		26H	TMPC0	
04H	BP		27H	Unused	
05H	ACC		28H	ADRL	
06H	PCL		29H	ADRH	
07H	TBLP		2AH	ADCR0	
08H	TBLH		2BH	ADCR1	
09H	TBHP		2CH	ACERL	
0AH	STATUS		2DH	Unused	
0BH	SMOD		2EH	TM0C0	
0CH	LVDC		2FH	TM0C1	
0DH	INTEG		30H	TM0DL	
0EH	WDTC		31H	TM0DH	
0FH	TBC		32H	TM0AL	
10H	INTC0		33H	TM0AH	
11H	INTC1		34H	TM1C0	
12H	INTC2		35H	TM1C1	
13H	MFI0		36H	Unused	
14H	MFI1		37H	TM1DL	
15H	MFI2		38H	TM1DH	
16H	CTRL		39H	TM1AL	
17H	LVRC		3AH	TM1AH	
18H	PAWU		3BH	Unused	
19H	PAPU		3DH		
1AH	PA		3EH	EEA	
1BH	PAC		3FH	EED	
1CH	PBPU		40H	Data RAM	EEC
1DH	PB				
1EH	PBC		9FH		
1FH	PCPU				
20H	PC				
21H	PCC				

▓ : Unused, read as 00H

HT66F24D Special Purpose Data Memory Structure

	Bank 0	Bank 1
00H	IAR0	
01H	MP0	
02H	IAR1	
03H	MP1	
04H	BP	
05H	ACC	
06H	PCL	
07H	TBLP	
08H	TBLH	
09H	TBHP	
0AH	STATUS	
0BH	SMOD	
0CH	LVDC	
0DH	INTEG	
0EH	WDTC	
0FH	TBC	
10H	INTC0	
11H	INTC1	
12H	INTC2	
13H	MFI0	
14H	MFI1	
15H	MFI2	
16H	CTRL	
17H	LVRC	
18H	PAWU	
19H	PAPU	
1AH	PA	
1BH	PAC	
1CH	PBPU	
1DH	PB	
1EH	PBC	
1FH	PCPU	
20H	PC	
21H	PCC	

	Bank 0	Bank 1
22H	PDPU	
23H	PD	
24H	PDC	
25H	PRM0	
26H	TMPC0	
27H	Unused	
28H	ADRL	
29H	ADRH	
2AH	ADCR0	
2BH	ADCR1	
2CH	ACERL	
2DH	Unused	
2EH	TM0C0	
2FH	TM0C1	
30H	TM0DL	
31H	TM0DH	
32H	TM0AL	
33H	TM0AH	
34H	TM1C0	
35H	TM1C1	
36H	TM1C2	
37H	TM1DL	
38H	TM1DH	
39H	TM1AL	
3AH	TM1AH	
3BH	TM1BL	
3CH	TM1BH	
3DH	Unused	
3EH	EEA	
3FH	EED	
40H		EEC
⋮	Data RAM	
FFH		

▨ : Unused, read as 00H

HT66F25D Special Purpose Data Memory Structure

	Bank 0~2		Bank 0~2		Bank 0, 2	Bank 1
00H	IAR0	1CH	PBPU	34H	TM1C0	
01H	MP0	1DH	PB	35H	TM1C1	
02H	IAR1	1EH	PBC	36H	TM1C2	
03H	MP1	1FH	PCPU	37H	TM1DL	
04H	BP	20H	PC	38H	TM1DH	
05H	ACC	21H	PCC	39H	TM1AL	
06H	PCL	22H	PDPU	3AH	TM1AH	
07H	TBLP	23H	PD	3BH	TM1BL	
08H	TBLH	24H	PDC	3CH	TM1BH	
09H	TBHP	25H	PRM0	3DH	Unused	
0AH	STATUS	26H	TMPC0	3EH	EEA	
0BH	SMOD	27H	TMPC1	3FH	EED	
0CH	LVDC	28H	ADRL	40H		EEC
0DH	INTEG	29H	ADRH	41H	TM2C0	
0EH	WDTC	2AH	ADCR0	42H	TM2C1	
0FH	TBC	2BH	ADCR1	43H	Unused	
10H	INTC0	2CH	ACERL	44H	TM2DL	
11H	INTC1	2DH	MFI3	45H	TM2DH	
12H	INTC2	2EH	TM0C0	46H	TM2AL	
13H	MFI0	2FH	TM0C1	47H	TM2AH	
14H	MFI1	30H	TM0DL	48H		
15H	MFI2	31H	TM0DH			
16H	CTRL	32H	TM0AL		Unused	
17H	LVDC	33H	TM0AH			
18H	PAWU			FFH		
19H	PAPU					
1AH	PA					
1BH	PAC					

☐ : Unused, read as 00H

HT66F26D Special Purpose Data Memory Structure

Special Function Register Description

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

Indirect Addressing Registers – IAR0, IAR1

The Indirect Addressing Registers, IAR0 and IAR1, although having their locations in normal RAM register space, do not actually physically exist as normal registers. The method of indirect addressing for RAM data manipulation uses these Indirect Addressing Registers and Memory Pointers, in contrast to direct memory addressing, where the actual memory address is specified. Actions on the IAR0 and IAR1 registers will result in no actual read or write operation to these registers but rather to the memory location specified by their corresponding Memory Pointers, MP0 or MP1. 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 while 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
begin:
mov  a,04h              ; setup size of block
mov  block, a
mov  a, offset adres1   ; Accumulator loaded with first RAM address
mov  mp0, a             ; setup memory pointer with first RAM address
loop:
clr  IAR0              ; clear the data at address defined by MP0
inc  mp0              ; increment memory pointer
sdz  block            ; check if last memory location has been cleared
jmp  loop
continue:
```

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

Bank Pointer – BP

Selecting the required Data Memory area is achieved using the Bank Pointer. The Bank Pointer BP bits are used to select Data Memory Banks 0~2. The Data Memory is initialised to Bank 0 after a reset, except for a WDT time-out reset in the Power down Mode, in which case, the Data Memory bank remains unaffected. It should be noted that the Special Function Data Memory is not affected by the bank selection, which means that the Special Function Registers can be accessed from within any bank. Directly addressing the Data Memory will always result in Bank 0 being accessed irrespective of the value of the Bank Pointer. Accessing data from banks other than Bank 0 must be implemented using indirect addressing.

BP Register – HT66F24D/HT66F25D

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

Bit 7~1   unimplemented, read as "0"

Bit 0   **DMBP0**: Data Memory Bank Pointer
   0: Bank 0
   1: Bank 1

BP Register – HT66F26D

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	DMBP1	DMBP0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~1   unimplemented, read as "0"

Bit 0   **DMBP1~DMBP0**: Data Memory Bank Pointer
   00: Bank 0
   01: Bank 1
   10: Bank 2
   11: Reserved

Accumulator – ACC

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

Program Counter Low Register – PCL

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

Look-up Table Registers – TBLP, TBHP, TBLH

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

Status Register – STATUS

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

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

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

- **C** is set if an operation results in a carry during an addition operation or if a borrow does not take place during a subtraction operation; otherwise C is cleared. C is also affected by a rotate through carry instruction.

- **AC** is set if an operation results in a carry out of the low nibbles in addition, or no borrow from the high nibble into the low nibble in subtraction; otherwise AC is cleared.

- **Z** is set if the result of an arithmetic or logical operation is zero; otherwise Z is cleared.

- **OV** is set if an operation results in a carry into the highest-order bit but not a carry out of the highest-order bit, or vice versa; otherwise OV is cleared.

- **PDF** is cleared by a system power-up or executing the "CLR WDT" instruction. PDF is set by executing the "HALT" instruction.

- **TO** is cleared by a system power-up or executing the "CLR WDT" or "HALT" instruction. TO is set by a WDT time-out.

In addition, on entering an interrupt sequence or executing a subroutine call, the status register will not be pushed onto the stack automatically. If the contents of the status registers are important and if the subroutine can corrupt the status register, precautions must be taken to correctly save it.

STATUS Register

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

"X" unknown

Bit 7, 6   Unimplemented, read as "0"

Bit 5   **TO**: Watchdog Time-Out flag

     0: After power up or executing the "CLR WDT" or "HALT" instruction

     1: A watchdog time-out occurred.

Bit 4   **PDF**: Power down flag

     0: After power up or executing the "CLR WDT" instruction

     1: By executing the "HALT" instruction

Bit 3   **OV**: Overflow flag

     0: no overflow

     1: an operation results in a carry into the highest-order bit but not a carry out of the highest-order bit or vice versa.

Bit 2   **Z**: Zero flag

     0: The result of an arithmetic or logical operation is not zero

     1: The result of an arithmetic or logical operation is zero

Bit 1   **AC**: Auxiliary flag

     0: no auxiliary carry

     1: an operation results in a carry out of the low nibbles in addition, or no borrow from the high nibble into the low nibble in subtraction

Bit 0   **C**: Carry flag

     0: no carry-out

     1: an operation results in a carry during an addition operation or if a borrow does not take place during a subtraction operation

     C is also affected by a rotate through carry instruction.

EEPROM Data Memory

The device contains 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 64x8 bits for this series of devices. Unlike the Program Memory and RAM Data Memory, the EEPROM Data Memory is not directly mapped into memory space and is therefore not directly addressable in the same way as the other types of memory. Read and Write operations to the EEPROM are carried out in single byte operations using an address and data register in Bank 0 and a single control register in Bank 1.

Device	Capacity	Address
HT66F24D	64 × 8	00H ~ 3FH
HT66F25D	64 × 8	00H ~ 3FH
HT66F26D	64 × 8	00H ~ 3FH

EEPROM Data Memory Structure

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

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

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

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

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

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	0	0	0	0	0	0	0	0

Bit 7~0   Data EEPROM data bit 7 ~ 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 7~0   "—" Unimplemented, read as 0

Bit 3   **WREN**: Data EEPROM Write Enable

   0: Disable

   1: Enable

This is the Data EEPROM Write Enable Bit which must be set high before Data EEPROM write operation are carried out. Clearing this bit to zero will inhibit Data EEPROM write operations.

Bit 2   **WR**: Data 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 operation are carried out. Clearing this bit to zero will inhibit Data EEPROM read operations.

Bit 0   **RD**: Data 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.

Reading Data from the EEPROM

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

Writing Data to the EEPROM

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

Write Protection

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

EEPROM Interrupt

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

Programming Considerations

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

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

Programming Examples

Reading data from the EEPROM - polling method

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

Writing Data to 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
CLR  EMI
SET  IAR1.3           ; set WREN bit, enable write operations
SET  IAR1.2           ; Start Write Cycle - set WR bit - executed immediately
                      ; after set WREN bit
SET  EMI
BACK:
SZ   IAR1.2           ; check for write cycle end
JMP  BACK
CLR  IAR1             ; disable EEPROM read/write
CLR  BP
```

Oscillator

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

Oscillator Overview

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, the device has the flexibility to optimize the performance/power ratio, a feature especially important in power sensitive portable applications.

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

Oscillator Types

System Clock Configurations

There are three system oscillators, two high speed and one low speed oscillators. The high speed oscillators are the external crystal/ceramic oscillator – HXT and the internal RC oscillator - HIRC. The low speed oscillator is the internal 32 kHz oscillator – LIRC. 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.
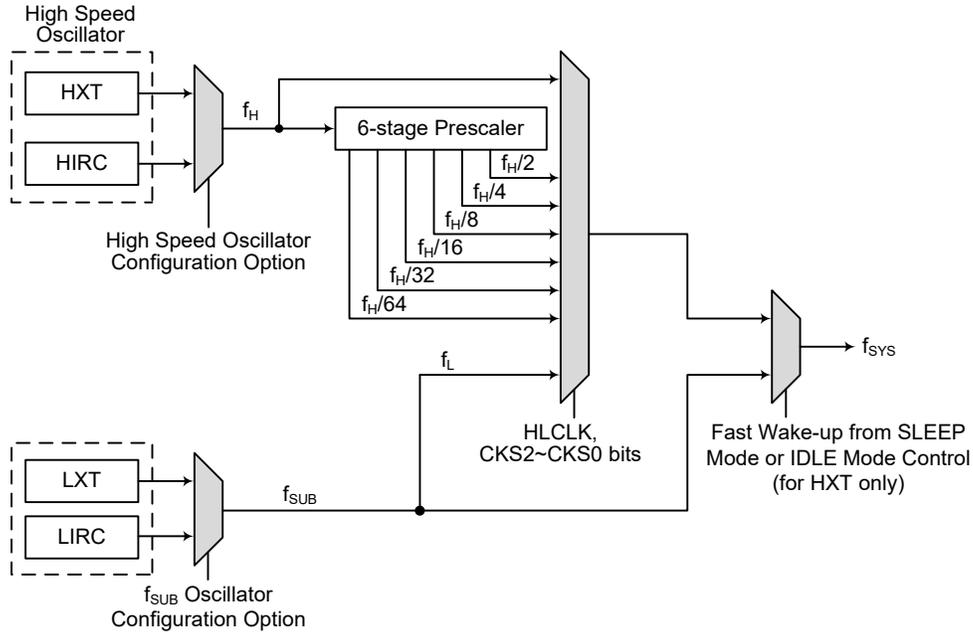
System Clock Configurations

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 that the crystal and any associated resistors and capacitors along with inter connecting lines are all located as close to the MCU as possible.
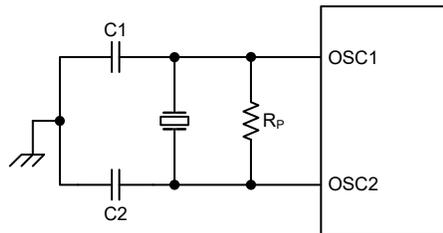
Note: 1. R_P is normally not required. C1 and C2 are required.
2. Although not shown OSC1/OSC2 pins have a parasitic capacitance of around 7pF.

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: C1 and C2 values are for guidance only.		

Crystal Recommended Capacitor Values

Internal High Speed 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 PA6 and PA5 are free for use as normal I/O pins.

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 32 kHz 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 32 kHz will have a tolerance within 10%.

Supplementary Oscillator

The low speed oscillator, in addition to providing a system clock source, is also used to provide a clock source to the Time Base Interrupts function.

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

The main system clock can come from a high frequency f_H or low frequency f_{SUB} 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 or HIRC oscillator, selected via a configuration option. The low speed system clock source can be sourced from internal clock f_{SUB}. If f_{SUB} is selected, then it can be sourced by the LIRC oscillator. The other choice, which is a divided version of the high speed system oscillator has a range of $f_H/2 \sim f_H/64$. Note that when the system clock source f_{SYS} is switched to f_{SUB} from f_H, the high speed oscillation will stop to conserve the power. Thus there is no $f_H \sim f_H/64$ for peripheral circuit to use.

There is an additional internal clock for the peripheral circuits, the substitute clock, f_{SUB}. The internal clock is sourced by the LIRC oscillator. 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. Together with $f_{SYS}/4$ it is also used as one of the clock sources for the Time Base interrupt function. The f_{SUB} clock is also used as a clock source for the TMs.
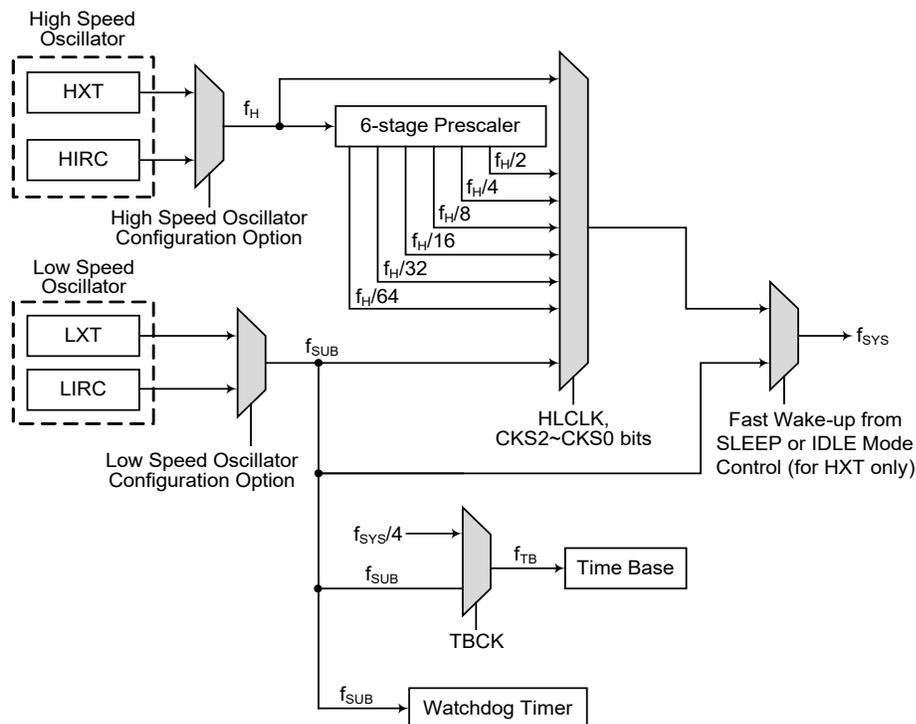
System Operation Modes

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.

Operating Mode	Description		
	CPU	f_{SYS}	f_{SUB}
NORMAL Mode	On	$f_H \sim f_H/64$	On
SLOW Mode	On	f_L	On
IDLE0 Mode	Off	Off	On
IDLE1 Mode	Off	On	On
SLEEP0 Mode	Off	Off	Off
SLEEP1 Mode	Off	Off	On

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 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 the low speed oscillator 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 SLEEP Mode is entered when a HALT instruction is executed and the IDLEN bit in the SMOD register is low. In the SLEEP0 mode the CPU will be stopped, and the f_{SUB} clock 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 SLEEP Mode is entered when a HALT instruction is executed and the IDLEN bit in the SMOD register is low. In the SLEEP1 mode the CPU will be stopped. However, the f_S clock will continue to operate if the LVDEN is set to 1 or the Watchdog Timer function is enabled since its clock source comes from the LIRC oscillator.

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 CTRL 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 and TMs. In the IDLE0 Mode, the system oscillator will be stopped while the f_{SUB} clock will be on.

IDLE1 Mode

The IDLE1 Mode is entered when a HALT instruction is executed and when the IDLEN bit in the SMOD register is high and the FSYSON bit in the CTRL 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 and TMs. In the IDLE1 Mode, the system oscillator will continue to run, and this system oscillator may be high speed or low speed system oscillator while the f_{SUB} clock will be on.

Control Register

The corresponding system control registers, SMOD and CTRL, are used for overall control of the internal clocks within the device.

SMOD Register

Bit	7	6	5	4	3	2	1	0
Name	CKS2	CKS1	CKS0	FSTEN	LTO	HTO	IDLEN	HLCLK
R/W	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_{SUB} (f_{LIRC})
    001: f_{SUB} (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 the 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 the device wakes 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 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 the device is powered on and then changes to a high level after the high speed system oscillator is stable. Therefore this flag will always be read as 1 by the application program after device power-on. The flag will be low when in the SLEEP or IDLE0 Mode but after a wake-up has occurred, the flag will change to a high level after 128 clock cycles if the HXT oscillator is used and after 15~16 clock cycles if the 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 the device will enter the IDLE Mode. In the IDLE1 Mode the CPU will stop running but the system clock will continue to keep the peripheral functions operational as the FSYSON bit is high. If FSYSON bit is low, the CPU and the system clock will all stop in IDLE0 mode. If the bit is low the device will enter the SLEEP Mode when a HALT instruction is executed.

Bit 0   **HLCLK**: System clock selection

    0: $f_H/2 \sim f_H/64$ or f_{SUB}

    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_{SUB} 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_{SUB} clock will be selected. When system clock switches from the f_H clock to the f_{SUB} clock, the f_H clock will be automatically switched off to conserve power.

CTRL Register

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

Bit 7   **FSYSON**: f_{SYS} Control in IDLE Mode

    0: Disable

    1: Enable

Bit 6~3   "—": Unimplemented, read as 0

Bit 2   **LVRF**: LVR function reset flag

    0: Not occur

    1: Occurred

This bit is set to 1 when a specific Low Voltage Reset situation condition occurs. This bit can only be cleared to 0 by the application program.

Bit 1   **LRF**: LVR Control register software reset flag

   0: Not occur

   1: Occurred

This bit is set to 1 if the LVRC register contains any non defined LVR voltage register values. This in effect acts like a software reset function. This bit can only be cleared to 0 by the application program.

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

   0: Not occur

   1: Occurred

This bit is set to 1 by the WDT Control register software reset and cleared by the application program. Note that this bit can only be cleared to 0 by the application program.

Fast Wake-up

To minimise power consumption the device can enter the SLEEP or IDLE0 Mode, where the system clock source to the device will be stopped. However when the device is 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 the 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 the device is 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 the Fast Wake-up function is enabled, then it will take one to two t_{SUB} clock cycles of the LIRC oscillator for the system to wake-up. The system will then initially run under the f_{SUB} clock source until 128 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 HIRC oscillators or LIRC oscillator is used as the system oscillator then it will take 15~16 clock cycles of the HIRC or 1~2 cycles of the LIRC oscillator 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	128 HXT cycles	128 HXT cycles		1~2 HXT cycles
	1	128 HXT cycles	1~2 f_{SUB} cycles (System runs first with f_{SUB} for 128 HXT cycles and then switches over to run with the HXT clock)		1~2 HXT 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

Wake-up Times   "X": don't care

Note that if the Watchdog Timer is disabled, which means that the LIRC oscillator is off, then there will be no Fast Wake-up function available when the device wakes-up from the SLEEP0 Mode.

NORMAL
$f_{SYS}=f_H\sim f_H/64$
f_H on
CPU run
f_{SYS} on
f_{SUB} on

IDLE1
HALT instruction is executed
CPU stop
IDLEN=1
FSYSON=1
f_{SYS} on
f_{SUB} on

SLEEP0
HALT instruction is executed
f_{SYS} off
CPU stop
IDLEN=0
f_{SUB} off
WDT & LVD off

IDLE0
HALT instruction is executed
CPU stop
IDLEN=1
FSYSON=0
f_{SYS} off
f_{SUB} on

SLEEP1
HALT instruction is executed
f_{SYS} off
CPU stop
IDLEN=0
f_{SUB} on
WDT or LVD on

SLOW
$f_{SYS}=f_{SUB}$
CPU run
f_{SYS} on
f_{SUB} on
f_H off

Operating Mode Switching

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

In simple terms, Mode Switching between the NORMAL Mode and SLOW Mode is executed using the HLCLK bit and CKS2~CKS0 bits in the SMOD register while Mode Switching from the NORMAL/SLOW Modes to the SLEEP/IDLE Modes is executed via the HALT instruction. When a HALT instruction is executed, whether the device enters the IDLE Mode or the SLEEP Mode is determined by the condition of the IDLEN bit in the SMOD register and FSYSON in the CTRL 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, $f_H/2\sim f_H/64$ or f_{SUB}. If the clock is from the f_{SUB}, 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. The accompanying flowchart shows what happens when the device moves between the various operating modes.

NORMAL Mode to SLOW Mode Switching

When running in the NORMAL Mode, which uses the high speed system oscillator, and therefore consumes more power, the system clock can switch to run in the SLOW Mode by set the HLCLK bit to 0 and set the CKS2~CKS0 bits to 000B or 001B 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 LIRC oscillator and therefore requires these oscillators to be stable before full mode switching occurs. This is monitored using the LTO bit in the SMOD register.
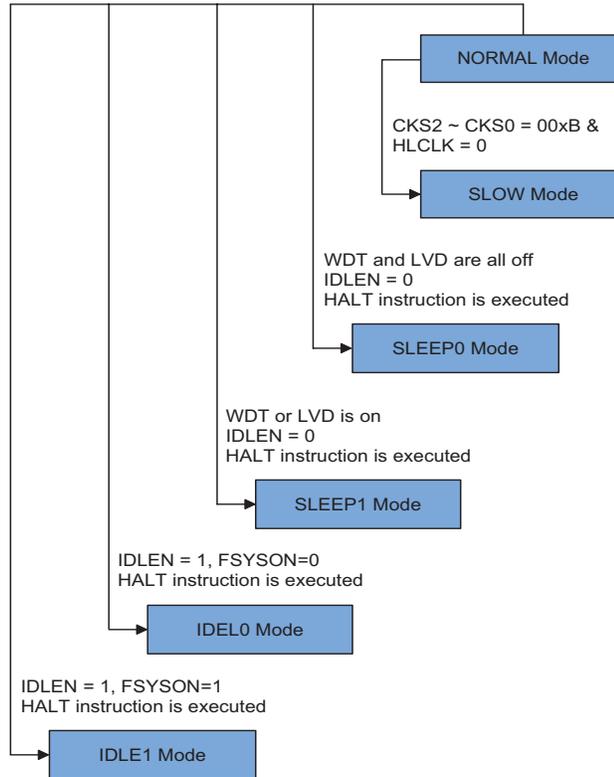
SLOW Mode to NORMAL Mode Switching

In SLOW Mode the system uses the LIRC low speed system oscillator. To switch back to the NORMAL Mode, where the high speed system oscillator is used, the HLCLK bit should be set to 1 or HLCLK bit is 0 but CKS2~CKS0 is set to 010B, 011B, 100B, 101B, 110B or 111B. 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.
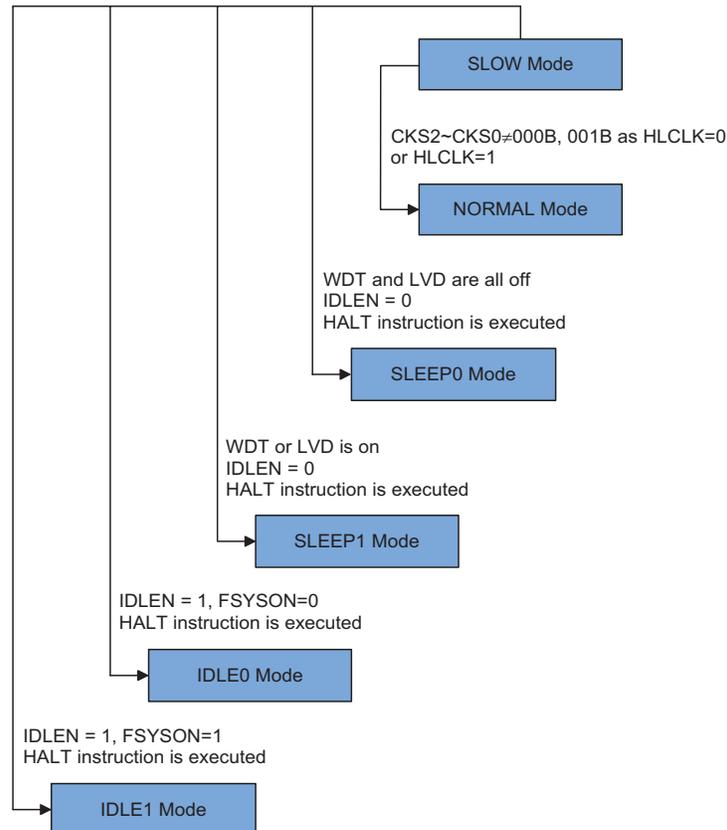
Entering the SLEEP0 Mode

There is only one way for the device 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.

- 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 the device 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
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and resume counting as the WDT function 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.

In SLOW Mode the system uses the LIRC low speed system oscillator. To switch back to the NORMAL Mode, where the high speed system oscillator is used, the HLCLK bit should be set to 1 or HLCLK bit is 0 but CKS2~CKS0 is set to 010B, 011B, 100B, 101B, 110B or 111B. 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.

- In the status register, the Power Down flag, PDF, will be set and the Watchdog time-out flag, TO, will be cleared.

Entering the IDLE0 Mode

There is only one way for the device to enter the IDLE0 Mode and that is to execute the HALT instruction in the application program with the IDLEN bit in SMOD register equal to 1 and the FSYSON bit in the CTRL 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 while the f_{SUB} clock and time base clock which is derived from the 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 function 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 IDLE1 Mode

There is only one way for the device to enter the IDLE1 Mode and that is to execute the HALT instruction in the application program with the IDLEN bit in SMOD register equal to 1 and the FSYSON bit in the CTRL register equal to 1. When this instruction is executed under the conditions described above, the following will occur:

- The system clock, 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 function 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.

Standby Current Considerations

As the main reason for entering the SLEEP or IDLE Mode is to keep the current consumption of the device to as low a value as possible, perhaps only in the order of several micro-amps except in the IDLE1 Mode, there are other considerations which must also be taken into account by the circuit designer if the power consumption is to be minimised. Special attention must be made to the I/O pins on the device. All high-impedance input pins must be connected to either a fixed high or low level as any floating input pins could create internal oscillations and result in increased current consumption. This also applies to devices which have different package types, as there may be 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 option has enabled the 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 falling edge on Port A
- A system interrupt
- A WDT overflow

If the device is woken up by a WDT overflow, a Watchdog Timer reset will be initiated. If a WDT time-out occurs, it will set TO flag and cause a wake-up that only resets the Program Counter and Stack Pointer, the other flags remain in their original status.

Each pin on Port A can be setup using the PAWU register to permit a negative transition on the pin to wake-up the system. When a Port A pin wake-up occurs, the program will resume execution at the instruction following the HALT instruction. If the system is woken up by an interrupt, then two possible situations may occur. The first is where the related interrupt is disabled or the interrupt is enabled but the stack is full, in which case the program will resume execution at the instruction following the HALT instruction. In this situation, the interrupt which woke-up the device will not be immediately serviced, but will rather be serviced later when the related interrupt is finally enabled or when a stack level becomes free. The other situation is where the related interrupt is enabled and the stack is not full, in which case the regular interrupt response takes place. If an interrupt request flag is set high before entering the SLEEP or IDLE Mode, the wake-up function of the related interrupt will be disabled.

Programming Considerations

The HXT oscillator use the same SST counter. For example, if the system is woken up from the SLEEP0 Mode and the HXT oscillator need to start-up from an off state.

If the device is woken up from the SLEEP0 Mode to the NORMAL Mode, the high speed system oscillator needs an SST period. The device will execute first instruction after the HTO bit is set to 1.

If the device is woken up from the SLEEP1 Mode to NORMAL Mode, and the system clock source is from the HXT oscillator and the FSTEN bit is set to 1, the system clock can first be switched to the LIRC oscillator after wake up.

There are peripheral functions, such as TMs, for which the f_{SYS} is used. If the system clock source is switched from the high speed clock f_H to low speed clock f_{SUB}, the clock source to the peripheral functions mentioned above will change accordingly.

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 source from the LIRC oscillator. The LIRC internal oscillator has an approximate frequency 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 Watchdog Timer source clock is then subdivided by a ratio of 2^8 to 2^{18} to give longer timeouts, the actual value being chosen using the WS2~WS0 bits in the WDTC register.

Watchdog Timer Control Register

A single register, WDTC, controls the required 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	WE4	WE3	WE2	WE1	WE0	WS2	WS1	WS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	1	0	0	1	1

Bit 7~3 **WE4~WE0**: WDT function software control

If the WDT configuration option is "always enable":
  10101 or 01010: Enabled
  Other: Reset MCU

If the WDT configuration option is "controlled by the WDT control register":
  10101: Disabled
  01010: Enabled
  Other: Reset MCU

When these bits are changed by the environmental noise to reset the microcontroller, the reset operation will be activated after 2~3 LIRC clock cycles and the WRF bit in the CTRL register will be set to 1.

Bit 2~0   **WS2~WS0**: WDT Time-out period selection

000: $2^8/f_S$
001: $2^{10}/f_S$
010: $2^{12}/f_S$
011: $2^{14}/f_S$
100: $2^{15}/f_S$
101: $2^{16}/f_S$
110: $2^{17}/f_S$
111: $2^{18}/f_S$

CTRL Register

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

"x": unknown

Bit 7   **FSYSON**: f_{SYS} Control in IDLE Mode

Describe elsewhere.

Bit 6~3   "—": Unimplemented, read as 0

Bit 2   **LVRF**: LVR function reset flag

Describe elsewhere.

Bit 1   **LRF**: LVR Control register software reset flag

Describe elsewhere.

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

 0: Not occur

 1: Occurred

This bit is set to 1 by the WDT Control register software reset and cleared by the application program. Note that this bit can only be cleared to 0 by the application program.

Watchdog Timer Operation

The Watchdog Timer operates by providing a device reset when its timer overflows. This means that in the application program and during normal operation the user has to strategically clear the Watchdog Timer before it overflows to prevent the Watchdog Timer from executing a reset. This is done using the clear watchdog instructions. If the program malfunctions for whatever reason, jumps to an unknown 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 always on select and clear instruction type are selected using configuration options. With regard to the Watchdog Timer enable/disable function, there are also five bits, WE4~WE0, in the WDTC register to offer additional enable/disable and reset control of the Watchdog Timer. If the WDT configuration option is determined that the WDT function is always enabled, the WE4~WE0 bits still have effects on the WDT function. When the WE4~WE0 bits value is equal to 01010B or 10101B, the WDT function is enabled. However, if the WE4~WE0 bits are changed to any other values except 01010B and 10101B, which is caused by the environmental noise, it will reset the microcontroller after 2~3 LIRC clock cycles. If the WDT configuration option is determined that the WDT function is controlled by the WDT control register, the WE4~WE0 values can determine which mode the WDT operates in. The WDT function will be disabled when the WE4~WE0 bits are set to a value of 10101B. The WDT function will be enabled if the WE4~WE0 bits value is equal to 01010B. If the WE4~WE0 bits are set to any other values by the environmental noise, except 01010B and 10101B, it will reset the device after 2~3 LIRC clock cycles. After power on these bits will have the value of 01010B.

Watchdog Timer Enable/Disable Control

WDT Configuration Option	WE4 ~ WE0 Bits	WDT Function
Always Enable	01010B or 10101B	Enable
	Any other value	Reset MCU
Controlled by WDT Control Register	10101B	Disable
	01010B	Enable
	Any other value	Reset MCU

"x": don't care.

Under normal program operation, a Watchdog Timer time-out will initialise a device reset and set the status bit TO. However, if the system is in the SLEEP or IDLE Mode, when a Watchdog Timer time-out occurs, the TO bit in the status register will be set and only the Program Counter and Stack Pointer will be reset. Three methods can be adopted to clear the contents of the Watchdog Timer. The first is a WDT reset, which means a certain value except 01010B and 10101B written into the WE4~WE0 bit filed, the second is using the Watchdog Timer software clear instructions and the third is via a HALT instruction.

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

The maximum time out period is when the 2^{18} division ratio is selected. As an example, with a 32 kHz LIRC oscillator as its source clock, this will give a maximum watchdog period of around 8 second for the 2^{18} division ratio, and a minimum timeout of 7.8ms for the 2^8 division ration.

Watchdog timer
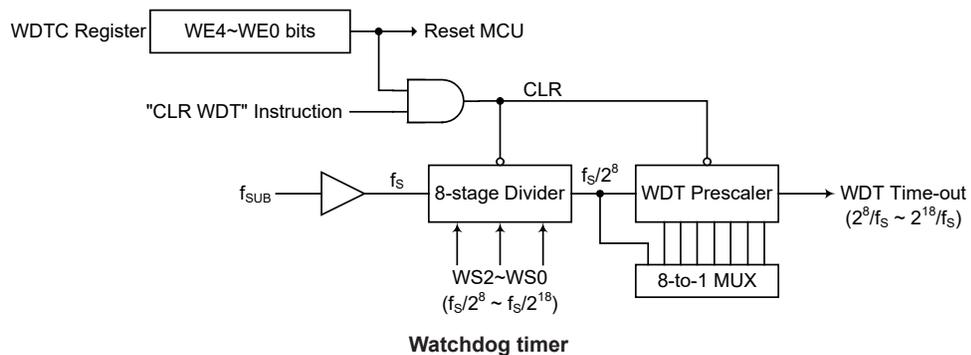
Reset and Initialisation

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

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

Reset Functions

There are 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.
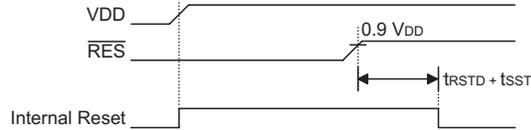
Power-On Reset Timing Chart

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

Low Voltage Reset – LVR

The microcontroller contains a low voltage reset circuit in order to monitor the supply voltage of the device. The LVR function is always enabled with a specific LVR voltage V_{LVR}. If the supply voltage of the device drops to within a range of $0.9V\sim V_{LVR}$ such as might occur when changing the battery, the LVR will automatically reset the device internally and the LVRF bit in the CTRL register will also be set to 1. For a valid LVR signal, a low supply voltage, i.e., a voltage in the range between $0.9V\sim V_{LVR}$ must exist for a time greater than that specified by t_{LVR} in the A.C. characteristics. If the low supply voltage state does not exceed this value, the LVR will ignore the low supply voltage and will not perform a reset function. The actual V_{LVR} value can be selected by the LVS bits in the LVRC register. If the LVS7~LVS0 bits are changed to some certain values by the environmental noise, the LVR will reset the device after 2~3 LIRC clock cycles. When this happens, the LRF bit in the CTRL register will be set to 1. After power on the register will have the value of 01010101B. Note that the LVR function will be automatically disabled when the device enters the power down mode.
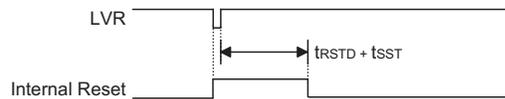
Low Voltage Reset Timing Chart

Note: t_{RSTD} is power-on delay, typical time=16.7ms

• LVRC Register

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

Bit 7~0   **LVS7~LVS0**: LVR Voltage Select control
  01010101: 2.1V
  00110011: 2.55V
  10011001: 3.15V
  10101010: 3.8V

Any other value: Generates MCU reset − register is reset to POR value

When an actual low voltage condition occurs, as specified by one of the four defined LVR voltage values above, an MCU reset will be generated. The reset operation will be activated after 2~3 LIRC clock cycles. In this situation the register contents will remain the same after such a reset occurs.

Any register value, other than the four defined LVR values above, will also result in the generation of an MCU reset. The reset operation will be activated after 2~3 LIRC clock cycles. However in this situation the register contents will be reset to the POR value.

• CTRL Register

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

Bit 7   **FSYSON**: f_{SYS} Control in IDLE Mode
  Describe elsewhere.

Bit 6~3   "**—**": Unimplemented, read as 0

Bit 2   **LVRF**: LVR function reset flag
  0: Not occur
  1: Occurred
  This bit is set to 1 when a specific Low Voltage Reset situation condition occurs. This bit can only be cleared to 0 by the application program.

Bit 1   **LRF**: LVR Control register software reset flag
 0: Not occur
  1: Occurred
  This bit is set to 1 if the LVRC register contains any non defined LVR voltage register values. This in effect acts like a software reset function. This bit can only be cleared to 0 by the application program.

Bit 0   **WRF**: WDT Control register software reset flag
  Describe elsewhere.

Watchdog Time-out Reset during Normal Operation

The Watchdog time-out Reset during normal operation will perform a full reset and also the Watchdog time-out flag TO will be set to "1".
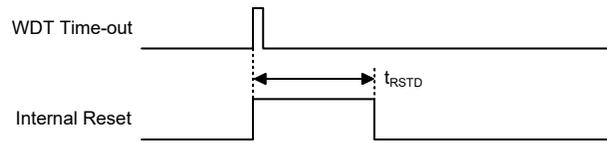

WDT Time-out Reset during Normal Operation Timing Chart

Note: t_{RSTD} is power-on delay, typical time=16.7ms

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.
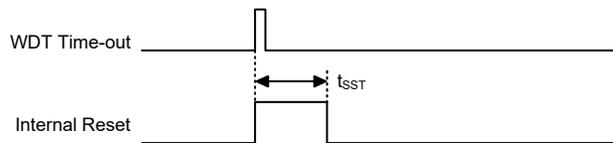
WDT Time-out Reset during SLEEP or IDLE Timing Chart

Note: The t_{SST} is 15~16 clock cycles if the system clock source is provided by HIRC.
The SST is 128 clock cycles for HXT. The SST is 1~2 clock cycles for LIRC.

Reset Initial Conditions

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

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

Note: "u" stands for unchanged.

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

Item	Condition After Power-on Reset
Program Counter	Reset to zero
Interrupts	All interrupt will be disabled
WDT, Time Base	Clear after reset, WDT begins counting
Timer Modules	Timer Counter will be turned off
Input/Output Ports	I/O ports will be setup as inputs and AN0~AN7 as A/D input pins
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.

HT66F24D

Register	Power On Reset	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
LVRC	0101 0101	uuuu uuuu	0101 0101	uuuu uuuu
CTRL	0--- -x00	0--- -100	0--- -000	u--- -uuu
INTEG	---- 0000	---- 0000	---- 0000	---- uuuu
WDTC	0101 0011	0101 0011	0101 0011	uuuu uuuu
TBC	0011 -111	0011 -111	0011 -111	uuuu -uuu
INTC0	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	0000 0000	0000 0000	0000 0000	uuuu uuuu
INTC2	---0 ---0	---0 ---0	---0 ---0	---u ---u
MFI0	--00 --00	--00 --00	--00 --00	--uu --uu
MFI1	--00 --00	--00 --00	--00 --00	--uu --uu
MFI2	--00 --00	--00 --00	--00 --00	--uu --uu
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
PRM0	0000 ----	0000 ----	0000 ----	uuuu ----
TMPC0	--00 0000	--00 0000	--00 0000	--uu uuuu
ADRL (ADRFS=0)	xxxx ----	xxxx ----	xxxx ----	uuuu ----
ADRL (ADRFS=1)	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
ADRH (ADRFS=0)	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
ADRH (ADRFS=1)	---- xxxx	---- xxxx	---- xxxx	---- uuuu
ADCR0	0110 -000	0110 -000	0110 -000	uuuu -uuu

Register	Power On Reset	LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
ADCR1	0 0 - 0 - 0 0 0	0 0 - 0 - 0 0 0	0 0 - 0 - 0 0 0	u u - u - u u u
ACERL	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
TM0C0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
TM0C1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
TM0DL	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
TM0DH	- - - - - - 0 0	- - - - - - 0 0	- - - - - - 0 0	- - - - - - u u
TM0AL	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
TM0AH	- - - - - - 0 0	- - - - - - 0 0	- - - - - - 0 0	- - - - - - u u
TM1C0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
TM1C1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
TM1DL	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
TM1DH	- - - - - - 0 0	- - - - - - 0 0	- - - - - - 0 0	- - - - - - u u
TM1AL	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
TM1AH	- - - - - - 0 0	- - - - - - 0 0	- - - - - - 0 0	- - - - - - u u

Note: "u" stands for unchanged

"x" stands for "unknown"

"-" stands for unimplemented

HT66F25D

Register	Power On Reset	LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
MP0	x x x x x x x x	x x x x x x x x	x x x x x x x x	u u u u u u u u
MP1	x x x x x x x x	x x x x x x x x	x x x x x x x x	u u u u u u u u
BP	- - - - - - - 0	- - - - - - - 0	- - - - - - - 0	- - - - - - - u
ACC	x x x x x x x x	u u u u u u u u	u u u u u u u u	u u u u u u u u
PCL	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
TBLP	x x x x x x x x	u u u u u u u u	u u u u u u u u	u u u u u u u u
TBLH	x x x x x x x x	u u u u u u u u	u u u u u u u u	u u u u u u u u
TBHP	- - - - x x x x	- - - - u u u u	- - - - u u u u	- - - - u u u u
STATUS	- - 0 0 x x x x	- - u u u u u u	- - 1 u u u u u	- - 1 1 u u u u
SMOD	0 0 0 0 0 0 1 1	0 0 0 0 0 0 1 1	0 0 0 0 0 0 1 1	u u u u u u u u
LVDC	- - 0 0 - 0 0 0	- - 0 0 - 0 0 0	- - 0 0 - 0 0 0	- - u u - u u u
LVRC	0 1 0 1 0 1 0 1	u u u u u u u u	0 1 0 1 0 1 0 1	u u u u u u u u
CTRL	0 - - - - x 0 0	0 - - - - 1 0 0	0 - - - - 0 0 0	u - - - - u u u
INTEG	- - - - 0 0 0 0	- - - - 0 0 0 0	- - - - 0 0 0 0	- - - - u u u u
WDTC	0 1 0 1 0 0 1 1	0 1 0 1 0 0 1 1	0 1 0 1 0 0 1 1	u u u u u u u u
TBC	0 0 1 1 - 1 1 1	0 0 1 1 - 1 1 1	0 0 1 1 - 1 1 1	u u u u - u u u
INTC0	- 0 0 0 0 0 0 0	- 0 0 0 0 0 0 0	- 0 0 0 0 0 0 0	- u u u u u u u
INTC1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
INTC2	- - - 0 - - - 0	- - - 0 - - - 0	- - - 0 - - - 0	- - - u - - - u
MFI0	- - 0 0 - - 0 0	- - 0 0 - - 0 0	- - 0 0 - - 0 0	- - u u - - u u
MFI1	- 0 0 0 - 0 0 0	- 0 0 0 - 0 0 0	- 0 0 0 - 0 0 0	- u u u - u u u
MFI2	- - 0 0 - - 0 0	- - 0 0 - - 0 0	- - 0 0 - - 0 0	- - u u - - u u
PAWU	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
PAPU	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
PA	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PAC	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PBPU	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
PB	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PBC	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PCPU	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
PC	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PCC	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PDPU	- - - - - - 0 0	- - - - - - 0 0	- - - - - - 0 0	- - - - - - u u
PD	- - - - - - 1 1	- - - - - - 1 1	- - - - - - 1 1	- - - - - - u u
PDC	- - - - - - 1 1	- - - - - - 1 1	- - - - - - 1 1	- - - - - - u u
PRM0	0 0 0 0 - - - -	0 0 0 0 - - - -	0 0 0 0 - - - -	u u u u - - - -
TMPC0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
ADRL (ADRFS=0)	x x x x - - - -	x x x x - - - -	x x x x - - - -	u u u u - - - -
ADRL (ADRFS=1)	x x x x x x x x	x x x x x x x x	x x x x x x x x	u u u u u u u u
ADRH (ADRFS=0)	x x x x x x x x	x x x x x x x x	x x x x x x x x	u u u u u u u u
ADRH (ADRFS=1)	- - - - x x x x	- - - - x x x x	- - - - x x x x	- - - - u u u u
ADCR0	0 1 1 0 - 0 0 0	0 1 1 0 - 0 0 0	0 1 1 0 - 0 0 0	u u u u - u u u
ADCR1	0 0 - 0 - 0 0 0	0 0 - 0 - 0 0 0	0 0 - 0 - 0 0 0	u u - u - u u u

Register	Power On Reset	LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
ACERL	1111 1111	1111 1111	1111 1111	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
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

Note: "u" stands for unchanged

"x" stands for "unknown"

"-" stands for unimplemented

HT66F26D

Register	Power On Reset	LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
MP0	x x x x x x x x	x x x x x x x x	x x x x x x x x	u u u u u u u u
MP1	x x x x x x x x	x x x x x x x x	x x x x x x x x	u u u u u u u u
BP	- - - - - - 0 0	- - - - - - 0 0	- - - - - - 0 0	- - - - - - - u
ACC	x x x x x x x x	u u u u u u u u	u u u u u u u u	u u u u u u u u
PCL	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
TBLP	x x x x x x x x	u u u u u u u u	u u u u u u u u	u u u u u u u u
TBLH	x x x x x x x x	u u u u u u u u	u u u u u u u u	u u u u u u u u
TBHP	- - - x x x x x	- - - u u u u u	- - - u u u u u	- - - u u u u u
STATUS	- - 0 0 x x x x	- - u u u u u u	- - 1 u u u u u	- - 1 1 u u u u
SMOD	0 0 0 0 0 0 1 1	0 0 0 0 0 0 1 1	0 0 0 0 0 0 1 1	u u u u u u u u
LVDC	- - 0 0 - 0 0 0	- - 0 0 - 0 0 0	- - 0 0 - 0 0 0	- - u u - u u u
LVRC	0 1 0 1 0 1 0 1	u u u u u u u u	0 1 0 1 0 1 0 1	u u u u u u u u
CTRL	0 - - - - x 0 0	0 - - - - 1 0 0	0 - - - - 0 0 0	u - - - - u u u
INTEG	- - - - 0 0 0 0	- - - - 0 0 0 0	- - - - 0 0 0 0	- - - - u u u u
WDTC	0 1 0 1 0 0 1 1	0 1 0 1 0 0 1 1	0 1 0 1 0 0 1 1	u u u u u u u u
TBC	0 0 1 1 - 1 1 1	0 0 1 1 - 1 1 1	0 0 1 1 - 1 1 1	u u u u - u u u
INTC0	- 0 0 0 0 0 0 0	- 0 0 0 0 0 0 0	- 0 0 0 0 0 0 0	- u u u u u u u
INTC1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
INTC2	- - - 0 - - - 0	- - - 0 - - - 0	- - - 0 - - - 0	- - - u - - - u
MFI0	- - 0 0 - - 0 0	- - 0 0 - - 0 0	- - 0 0 - - 0 0	- - u u - - u u
MFI1	- 0 0 0 - 0 0 0	- 0 0 0 - 0 0 0	- 0 0 0 - 0 0 0	- u u u - u u u
MFI2	- - 0 0 - - 0 0	- - 0 0 - - 0 0	- - 0 0 - - 0 0	- - u u - - u u
MFI3	- - 0 0 - - 0 0	- - 0 0 - - 0 0	- - 0 0 - - 0 0	- - u u - - u u
PAWU	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
PAPU	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
PA	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PAC	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PBPU	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
PB	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PBC	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PCPU	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
PC	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PCC	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	u u u u u u u u
PDPU	- - - - - - 0 0	- - - - - - 0 0	- - - - - - 0 0	- - - - - - u u
PD	- - - - - - 1 1	- - - - - - 1 1	- - - - - - 1 1	- - - - - - u u
PDC	- - - - - - 1 1	- - - - - - 1 1	- - - - - - 1 1	- - - - - - u u
PRM0	0 0 0 0 - - - -	0 0 0 0 - - - -	0 0 0 0 - - - -	u u u u - - - -
TMPC0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	u u u u u u u u
TMPC1	- - - - - 0 0 0	- - - - - 0 0 0	- - - - - 0 0 0	- - - - - u u u
ADRL (ADRFS=0)	x x x x - - - -	x x x x - - - -	x x x x - - - -	u u u u - - - -
ADRL (ADRFS=1)	x x x x - - - -	x x x x - - - -	x x x x - - - -	u u u u u u u u
ADRH (ADRFS=0)	x x x x x x x x	x x x x x x x x	x x x x x x x x	u u u u u u u u
ADRH (ADRFS=1)	x x x x x x x x	x x x x x x x x	x x x x x x x x	- - - - u u u u

Register	Power On Reset	LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (IDLE)
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
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
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
TM1BL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1BH	---- --00	---- --00	---- --00	---- --uu
TM2C0	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2C1	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2DL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2DH	---- --00	---- --00	---- --00	---- --uu
TM2AL	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2AH	---- --00	---- --00	---- --00	---- --uu

Note: "u" stands for unchanged

"x" stands for "unknown"

"-" stands for unimplemented

Input/Output Ports

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

The device provides bidirectional input/output lines labeled with port names PA~PD. 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 Resistor Lists

HT66F24D

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PBPU	—	—	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PB	—	—	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PBC	—	—	PB5	PB4	PB3	PB2	PB1	PB0
PCPU	PCPU7	PCPU6	PCPU5	PCPU4	PCPU3	PCPU2	PCPU1	PCPU0
PC	PCC7	PCC6	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0
PCC	PC7	PC4	PC5	PC4	PC3	PC2	PC1	PC0

HT66F25D/HT66F26D

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PBPU	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PB	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PBC	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
PCPU	PCPU7	PCPU6	PCPU5	PCPU4	PCPU3	PCPU2	PCPU1	PCPU0
PC	PCC7	PCC6	PCC5	PCC4	PCC3	PCC2	PCC1	PCC0
PCC	PC7	PC4	PC5	PC4	PC3	PC2	PC1	PC0
PDPU	—	—	—	—	—	—	PDPU1	PDPU0
PD	—	—	—	—	—	—	PDC1	PDC0
PDC	—	—	—	—	—	—	PD1	PD0

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 PAPU~PDPU and are implemented using weak PMOS transistors.

PAPU 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: I/O Port A bit 7 ~ bit 0 Pull-high Control
　　　　　　　0: Disable
　　　　　　　1: Enable

PBPU Register

- **HT66F24D**

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

- **HT66F25D/HT66F26D**

Bit	7	6	5	4	3	2	1	0
Name	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
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

　　　　　"—": Unimplemented, read as 0
PBPUn: I/O Port B bit n Pull-high Control
　　　　　　0: Disable
　　　　　　1: Enable

PCPU Register

- **HT66F24D/HT66F25D/HT66F26D**

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

　　　　　"—": Unimplemented, read as 0
PCPUn: I/O Port C bit n Pull-high Control
　　　　　　0: Disable
　　　　　　1: Enable

PDPU Register

- **HT66F25D/HT66F26D**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	PDPU1	PDPU0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

"—": Unimplemented, read as 0

PDPUn: I/O Port D bit n Pull-high Control
　　0: Disable
　　1: Enable

Port A Wake-up

The HALT instruction forces the microcontroller into the SLEEP or IDLE Mode which preserves power, a feature that is important for battery and other low-power applications. Various methods exist to wake-up the microcontroller, one of which is to change the logic condition on one of the Port A pins from high to low. This function is especially suitable for applications that can be woken up via external switches. Each pin on Port A can be selected individually to have this wake-up feature using the PAWU register.

PAWU Register

Bit	7	6	5	4	3	2	1	0
Name	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　　**PAWU**: Port A bit 7 ~ bit 0 Wake-up Control
　　　　0: Disable
　　　　1: Enable

I/O Port Control Registers

Each I/O port has its own control register known as PAC~PDC, 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 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 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

Bit 7~0　　**PAC7~PAC0**: I/O Port A bit 7 ~ bit 0 Input/Output Control
　　　　0: Output
　　　　1: Input

PBC Register

- **HT66F24D**

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

- **HT66F25D/HT66F26D**

Bit	7	6	5	4	3	2	1	0
Name	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
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

"—": Unimplemented, read as 0

PBCn: I/O Port B bit n Input/Output Control
             0: Output
             1: Input

PCC Register

- **HT66F24D/HT66F25D/HT66F26D**

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

"—": Unimplemented, read as 0

PCCn: I/O Port C bit n Input/Output Control
             0: Output
             1: Input

PDC Register

- **HT66F25D/HT66F26D**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	PDC1	PDC0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	1	1

Bit 7~2   "—" Unimplemented, read as 0

Bit 1~0   **PDC1~PDC0**: I/O Port D bit 1 ~ bit 0 Input/Output Control
             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.

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.

PRM0 Register

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

Bit 7   **INT1PS**: INT1 Pin Remapping Control
                0: INT1 on PB0
                1: INT1 on PC1

Bit 6   **INT0PS**: INT0 Pin Remapping Control
                0: INT1 on PA2
                1: INT1 on PC0

Bit 5   **TCK1PS**: TCK1 Pin Remapping Control
                0: TCK1 on PB0
                1: TCK1 on PC3

Bit 5   "—" Unimplemented, read as 0

Bit 4   **TCK0PS**: TCK0 Pin Remapping Control
                0: TCK0 on PA6
                1: TCK0 on PC2

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

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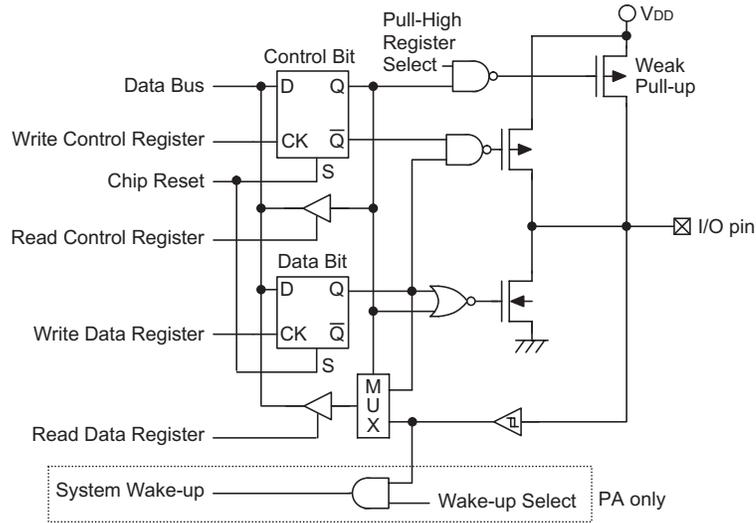
Generic Input/Output Structure
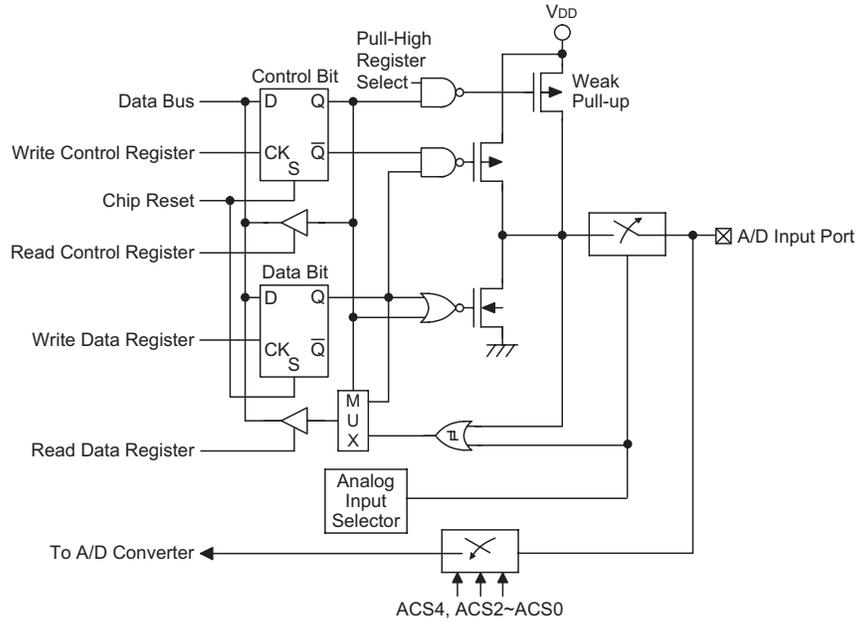
A/D Input/Output Structure

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

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

Introduction

The devices contain up to three TMs with each TM having a reference name of TM0, TM1 and TM2. Each individual TM can be categorised as a certain type, namely Compact Type TM (CTM), Standard Type TM (STM) or Enhanced Type TM (ETM). 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 and the detailed operation regarding each of the TM types will be described in separate sections. The main features and differences between the three types of TMs are summarised in the accompanying table.

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

TM Function Summary

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

Device	TM0	TM1	TM2
HT66F24D	10-bit CTM	10-bit STM	—
HT66F25D	10-bit CTM	10-bit ETM	—
HT66F26D	10-bit CTM	10-bit ETM	10-bit STM

TM Name/Type Reference

TM Operation

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

TM Clock Source

The clock source which drives the main counter in each TM can originate from various sources. The selection of the required clock source is implemented using the TnCK2~TnCK0 bits in the TM control registers. The clock source can be a ratio of either the system clock f_{SYS} or the high speed clock f_H, the f_{SUB} clock source or the external TCKn pin. Note that setting these bits to the value 101 will select a reserved clock input, in effect disconnecting the TM clock source. The TCKn pin clock source is used to allow an external signal to drive the TM as an external clock source or for event counting.

TM Interrupts

The Compact and Standard type TMs each have two internal interrupts, one for each of the internal comparator A or comparator P, which generate a TM interrupt when a compare match condition occurs. As the Enhanced type TM has three internal comparators and comparator A or comparator B or comparator P compare match functions, it consequently has three internal interrupts. When a TM interrupt is generated it can be used to clear the counter and also to change the state of the TM output pin.

TM External Pins

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

The TMs each have one or more output pins with the label TPn. When the TM is in the Compare Match Output Mode, these pins can be controlled by the TM to switch to a high or low level or to toggle when a compare match situation occurs. The external TPn output pin is also the pin where the TM generates the PWM output waveform. As the TM output pins are pin-shared with other function, the TM output function must first be setup using registers. A single bit in one of the registers determines if its associated pin is to be used as an external TM output pin or if it is to have another function. The number of output pins for each TM type and device is different, the details are provided in the accompanying table. All TM output pin names have a "_n" suffix. Pin names that include a "_1" or "_2" suffix indicate that they are from a TM with multiple output pins. This allows the TM to generate a complimentary output pair, selected using the I/O register data bits.

Device	CTM	STM	ETM	Register
HT66F24D	TP0_0, TP0_1, TP0_2	TP1_0, TP1_1, TP1_2	—	TMPC0
HT66F25D	TP0_0, TP0_1, TP0_2	—	TP1A_0, TP1A_1 TP1B_0, TP1B_1, TP1B_2	TMPC0
HT66F26D	TP0_0, TP0_1, TP0_2	TP2_0, TP2_1, TP2_2	TP1A_0, TP1A_1 TP1B_0, TP1B_1, TP1B_2	TMPC0 TMPC1

TM Output Pins

TM Input/Output Pin Control Registers

Selecting to have a TM input/output or whether to retain its other shared function is implemented using one register, 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	Register	Bit							
		7	6	5	4	3	2	1	0
HT66F24D	TMPC0	—	—	T1CP2	T1CP1	T1CP0	T0CP2	T0CP1	T0CP0
HT66F25D	TMPC0	T1ACP1	T1ACP0	T1BCP2	T1BCP1	T1BCP0	T0CP2	T0CP1	T0CP0
HT66F26D	TMPC0	T1ACP1	T1ACP0	T1BCP2	T1BCP1	T1BCP0	T0CP2	T0CP1	T0CP0
	TMPC1	—	—	—	—	—	T2CP2	T2CP1	T2CP0

TM Input/Output Pin Control Register List
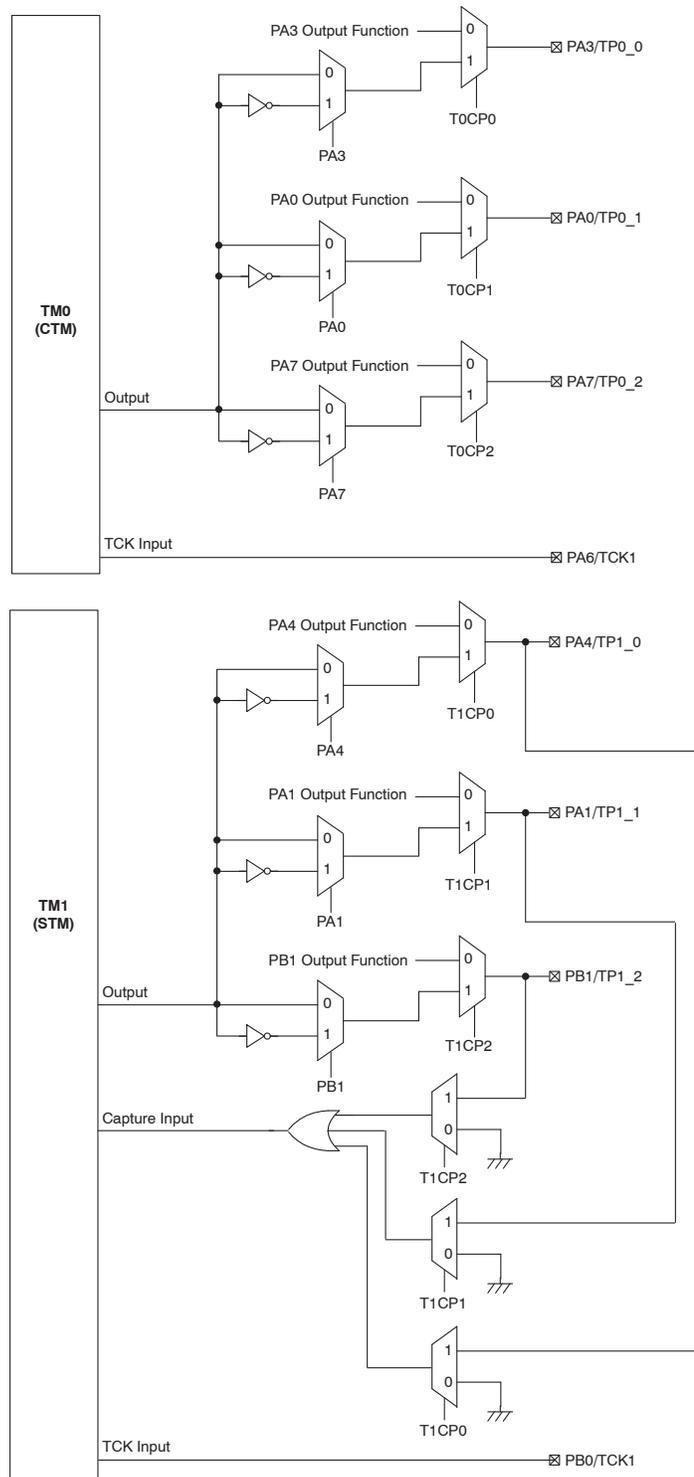
Note: (1) The I/O register data bits shown are used for TM output inversion control.

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

HT66F24D TM0 & TM1 Function Pin Control Block Diagram
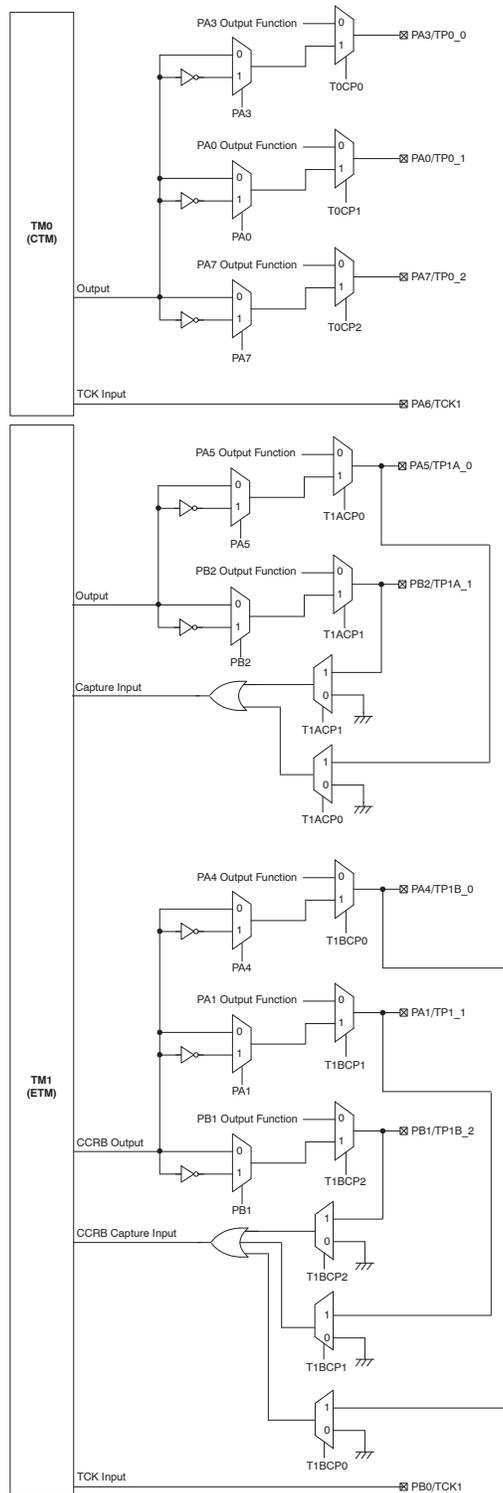
Note: (1) The I/O register data bits shown are used for TM output inversion control.

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

HT66F25D TM0 & TM1 Function Pin Control Block Diagram
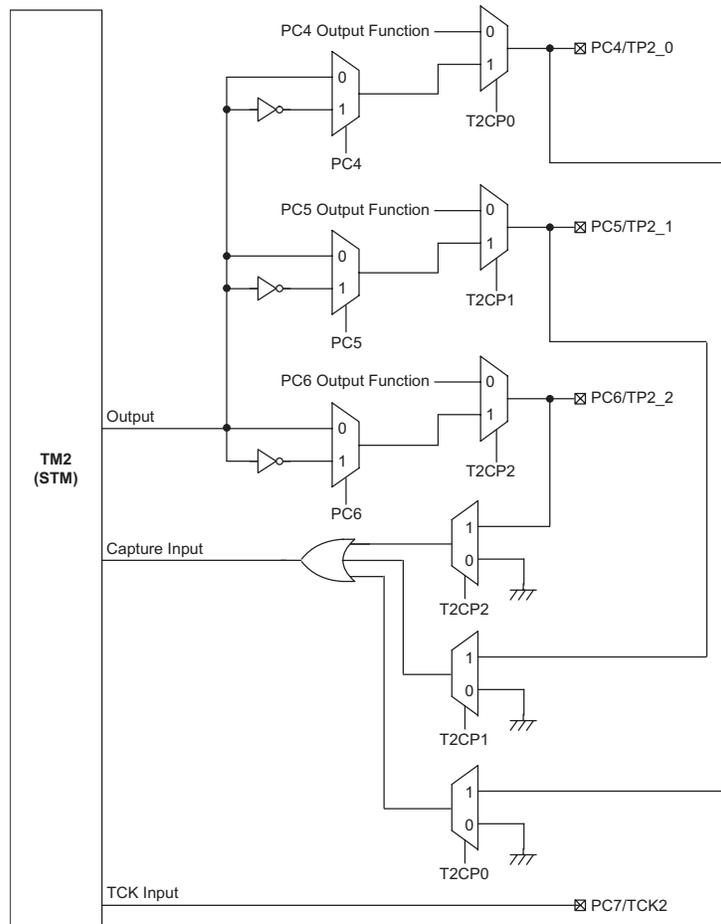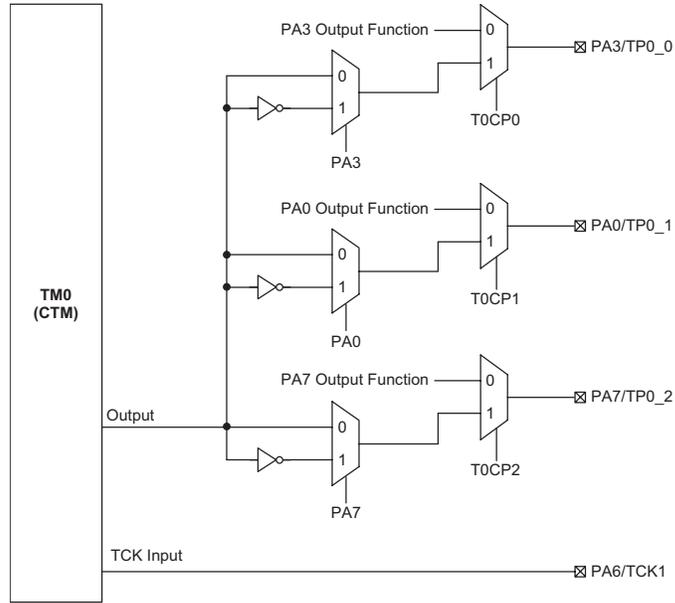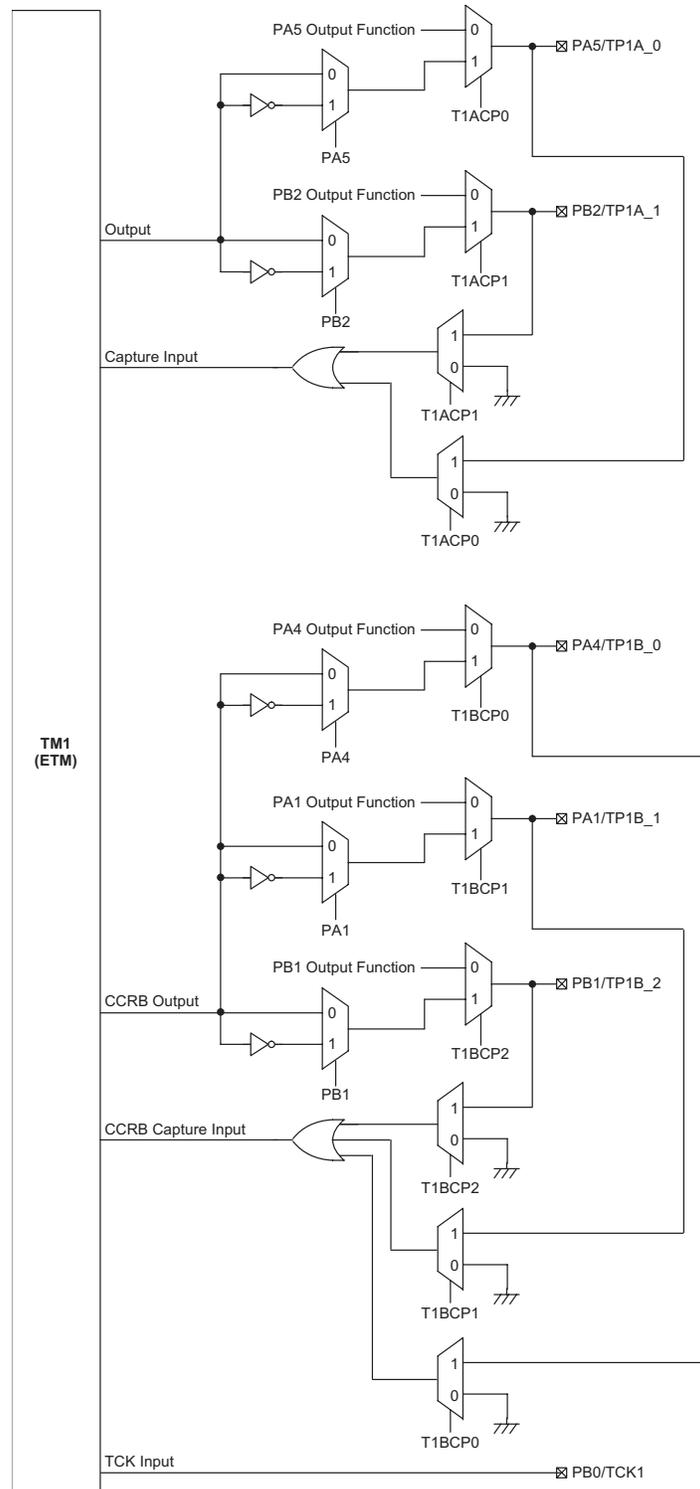
Note: (1) The I/O register data bits shown are used for TM output inversion control.

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

HT66F26D TM Function Pin Control Block Diagram

TMPC0 Register

- **HT66F24D**

Register Name	Bit							
	7	6	5	4	3	2	1	0
Name	—	—	T1CP2	T1CP1	T1CP0	T0CP2	T0CP1	T0CP0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	0	0	0	0	0	0

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

Bit 5   **T1CP2**: TP1_2 pin enable Control
    0: Disable
    1: Enable

Bit 4   **T1CP1**: TP1_1 pin enable Control
    0: Disable
    1: Enable

Bit 3   **T1CP0**: TP1_0 pin enable Control
    0: Disable
    1: Enable

Bit 2   **T0CP2**: TP0_2 pin enable Control
    0: Disable
    1: Enable

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

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

• **HT66F25D/HT66F26D**

Register Name	Bit							
	7	6	5	4	3	2	1	0
Name	T1ACP1	T1ACP0	T1BCP2	T1BCP1	T1BCP0	T0CP2	T0CP1	T0CP0
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 **T1ACP1**: TP1A_1 pin enable Control
    0: Disable
    1: Enable

Bit 6 **T1ACP0**: TP1A_0 pin enable Control
    0: Disable
    1: Enable

Bit 5 **T1BCP2**: TP1B_2 pin enable Control
    0: Disable
    1: Enable

Bit 4 **T1BCP1**: TP1B_1 pin enable Control
    0: Disable
    1: Enable

Bit 3 **T1BCP0**: TP1B_0 pin enable Control
    0: Disable
    1: Enable

Bit 2 **T0CP2**: TP0_2 pin enable Control
    0: Disable
    1: Enable

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

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

TMPC1 Register

• **HT66F26D**

Register Name	Bit							
	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	T2CP2	T2CP1	T2CP0
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	0	0	0

Bit 7~3 "—" Unimplemented, read as 0

Bit 2 **T2CP2**: TP2_2 pin enable Control
    0: Disable
    1: Enable

Bit 1 **T2CP1**: TP2_1 pin enable Control
    0: Disable
    1: Enable

Bit 0 **T2CP0**: TP2_0 pin enable Control
    0: Disable
    1: Enable

Programming Considerations

The TM Counter Registers and the Capture/Compare CCRA and CCRB registers, being 10-bit, all have a low and high byte structure. The high bytes can be directly accessed, but as the low bytes can only be accessed via an internal 8-bit buffer, reading or writing to these register pairs must be carried out in a specific way. The important point to note is that data transfer to and from the 8-bit buffer and its related low byte only takes place when a write or read operation to its corresponding high byte is executed. As the CCRA and CCRB registers are implemented in the way shown in the following diagram and accessing these register pairs is carried out in a specific way described above, it is recommended to use the "MOV" instruction to access the CCRA and CCRB low byte registers, named TMxAL and TMxBL, using the following access procedures. Accessing the CCRA or CCRB low byte registers without following these access procedures will result in unpredictable values.
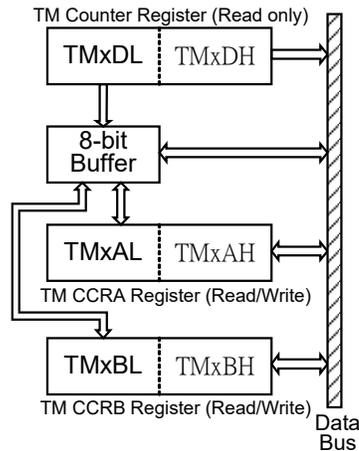
The following steps show the read and write procedures:

Writing Data to CCRB or CCRA

- Step 1. Write data to Low Byte TMxAL or TMxBL - note that here data is only written to the 8-bit buffer.

- Step 2. Write data to High Byte TMxAH or TMxBH - here data is written directly to the high byte registers and simultaneously data is latched from the 8-bit buffer to the Low Byte registers.

Reading Data from the Counter Registers and CCRB or CCRA

- Step 1. Read data from the High Byte TMxDH, TMxAH or TMxBH - here data is read directly from the High Byte registers and simultaneously data is latched from the Low Byte register into the 8-bit buffer.

- Step 2. Read data from the Low Byte TMxDL, TMxAL or TMxBL - this step reads data from the 8-bit buffer.

Compact Type TM – CTM

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

CTM	Name	TM No.	TM Input Pin	TM Output Pin
HT66F24D HT66F25D HT66F26D	10-bit CTM	0	TCK0	TP0_0, TP0_1, TP0_2

Compact TM Operation

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

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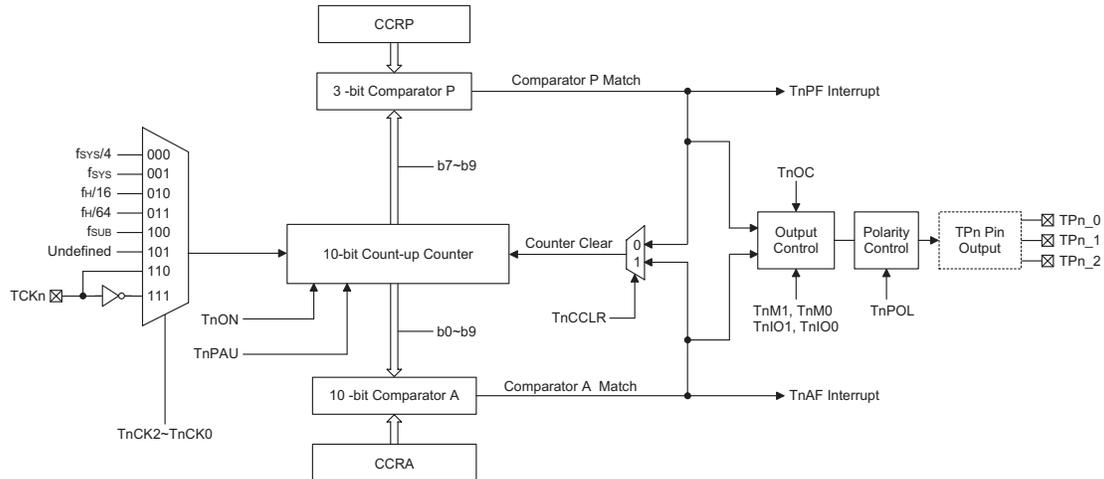
Compact Type TM Block Diagram

Compact Type TM Register Description

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

CTM Register List

Register Name	Bit							
	7	6	5	4	3	2	1	0
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

10-bit Compact TM Register List

TM0DL Register

Register Name	Bit							
	7	6	5	4	3	2	1	0
TM0C0	D7	D6	D5	D4	D3	D2	D1	D0
TM0C1	R	R	R	R	R	R	R	R
TM0DL	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

Register Name	Bit							
	7	6	5	4	3	2	1	0
TM0C0	—	—	—	—	—	—	D9	D8
TM0C1	—	—	—	—	—	—	R	R
TM0DL	—	—	—	—	—	—	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

Register Name	Bit							
	7	6	5	4	3	2	1	0
TM0C0	D7	D6	D5	D4	D3	D2	D1	D0
TM0C1	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TM0DL	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

Register Name	Bit							
	7	6	5	4	3	2	1	0
TM0C0	—	—	—	—	—	—	D9	D8
TM0C1	—	—	—	—	—	—	R/W	R/W
TM0DL	—	—	—	—	—	—	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

Register Name	Bit							
	7	6	5	4	3	2	1	0
TM0C0	T0PAU	T0CK2	T0CK1	T0CK0	T0ON	T0RP2	T0RP1	T0RP0
TM0C1	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TM0DL	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_{SUB}

  101: undefined, can not be selected.

  110: TCK0 rising edge clock

  111: TCK0 falling edge clock

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

Bit 3   **T0ON**: TM0 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.

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 T0OC bit, when the T0ON bit changes from low to high.

Bit 2~0   **T0RP2~T0RP**0: TM0 CCRP 3-bit register, compare 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

Register Name	Bit							
	7	6	5	4	3	2	1	0
TM0C0	T0M1	T0M0	T0IO1	T0IO0	T0OC	T0POL	T0DPX	T0CCLR
TM0C1	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TM0DL	0	0	0	0	0	0	0	0

Bit 7~6   **T0M1~T0M0**: Select TM0 Operation 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, TP0_2 output function

Compare Match Output Mode

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

PWM Mode

   00: PWM output inactive state
   01: PWM output active state
   10: PWM output
   11: Undefined

Timer/Counter Mode

   Unused

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

In the Compare Match Output Mode, the T0IO1 and T0IO0 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 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 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 T0ON bit from low to high.

In the PWM Mode, the TnIO1 and TnIO0 bits determine how the TM output pin changes state when a certain compare match condition occurs. The PWM output function is modified by changing these two bits. It is necessary to change the values of the TnIO1 and TnIO0 bits only after the TMn has been switched off. Unpredictable PWM outputs will occur if the TnIO1 and TnIO0 bits are changed when the TM is running

Bit 3   **T0OC**: TP0_0, TP0_1, TP0_2 output control bit

Compare Match Output Mode

  0: Initial low

  1: Initial high

PWM Mode

  0: Active low

  1: Active high

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

Bit 2   **T0POL**: TP0_0, TP0_1, TP0_2 output Polarity control

  0: Non-invert

  1: Invert

This bit controls the polarity of the TP0_0, TP0_1 or TP0_2 output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1   **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 TM 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 Operation Modes

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

Compare Match Output Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register, should be set to 00B respectively. In this mode once the counter is enabled and running it can be cleared by three methods. These are a counter overflow, a compare match from Comparator A and a compare match from Comparator P. When the TnCCLR bit is low, there are two ways in which the counter can be cleared. One is when a compare match occurs from Comparator P, the other is when the CCRP bits are all zero which allows the counter to overflow. Here both TnAF and TnPF interrupt request flags for the Comparator A and Comparator P respectively, will both be generated.

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

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

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

Counter Value

TnCCLR = 1; TnM [1:0] = 00

CCRA > 0 Counter cleared by CCRA value

CCRA = 0
Counter overflow

0x3FF

CCRA

Resume

CCRA=0

CCRP

Pause

Stop

Counter Restart

Time

TnON

TnPAU

TnPOL

No TnAF flag generated on CCRA overflow

CCRA Int. Flag TnAF

CCRP Int. Flag TnPF

TnPF not generated

Output does not change

TM O/P Pin

Output not affected by TnAF flag. Remains High until reset by TnON bit

Output pin set to initial Level Low if TnOC=0

Output Toggle with TnAF flag

Output Pin Reset to Initial value

Output Inverts when TnPOL is high

Here TnIO [1:0] = 11 Toggle Output select

Note TnIO [1:0] = 10 Active High Output select

Output controlled by other pin-shared function

Compare Match Output Mode – TnCCLR=1

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

2. The TM output pin is controlled only by the TnAF flag

3. The output pin is reset to its initial state by a TnON bit rising edge

4. The TnPF flag is not generated when TnCCLR=1

Timer/Counter Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 11 respectively. The Timer/Counter Mode operates in an identical way to the Compare Match Output Mode generating the same interrupt flags. The exception is that in the Timer/Counter Mode the TM output pin is not used. Therefore the above description and Timing Diagrams for the Compare Match Output Mode can be used to understand its function. As the TM output pin is not used in this mode, the pin can be used as a normal I/O pin or other pin-shared function.

PWM Output Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 10 respectively. The PWM function within the TM is useful for applications which require functions such as motor control, heating control, illumination control etc. By providing a signal of fixed frequency but of varying duty cycle on the TM output pin, a square wave AC waveform can be generated with varying equivalent DC RMS values.

As both the period and duty cycle of the PWM waveform can be controlled, the choice of generated waveform is extremely flexible. In the PWM mode, the TnCCLR bit has no effect on the PWM operation. Both of the CCRA and CCRP registers are used to generate the PWM waveform, one register is used to clear the internal counter and thus control the PWM waveform frequency, while the other one is used to control the duty cycle. Which register is used to control either frequency or duty cycle is determined using the TnDPX bit in the TMnC1 register. The PWM waveform frequency and duty cycle can therefore be controlled by the values in the CCRA and CCRP registers.

An interrupt flag, one for each of the CCRA and CCRP, will be generated when a compare match occurs from either Comparator A or Comparator P. The TnOC bit in the TMnC1 register is used to select the required polarity of the PWM waveform while the two TnIO1 and TnIO0 bits are used to enable the PWM output or to force the TM output pin to a fixed high or low level. The TnPOL bit is used to reverse the polarity of the PWM output waveform.

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

4. The TnCCLR bit has no influence on PWM operation

PWM Mode – TnDPX=1

Note: 1. Here TnDPX = 1 – Counter cleared by CCRA

2. A counter clear sets the PWM Period

3. The internal PWM function continues running even when TnIO [1:0] = 00 or 01

4. The TnCCLR bit has no influence on PWM operation

Standard Type TM – STM

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

STM	Name	TM No.	TM Input Pin	TM Output Pin
HT66F24D	10-bit STM	1	TCK1	TP1_0, TP1_1, TP1_2
HT66F25D	—	—	—	—
HT66F26D	10-bit STM	2	TCK2	TP2_0, TP2_1, TP2_2

Standard TM Operation

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

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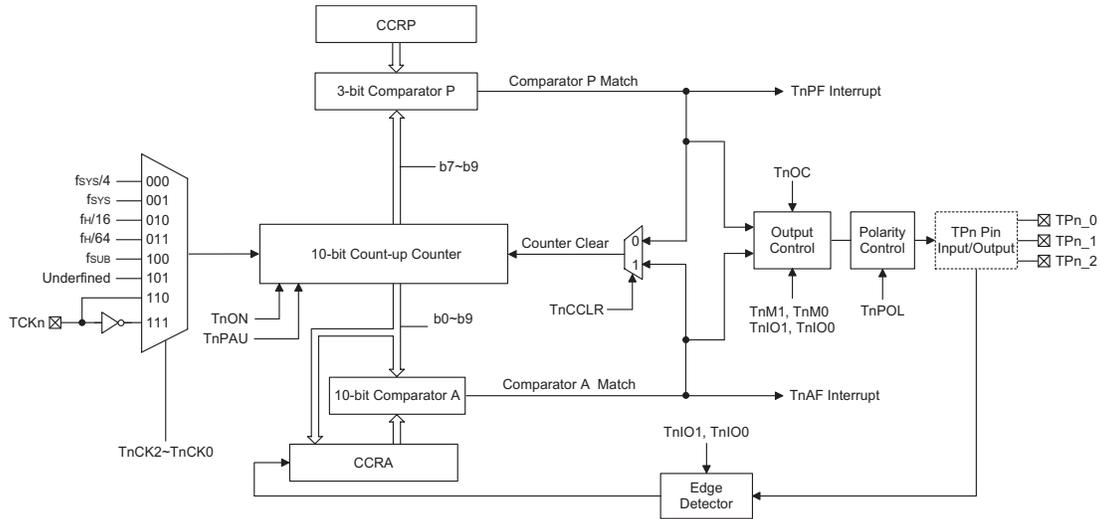
Standard Type TM Block Diagram

Standard Type TM Register Description

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

STM Register List – HT66F24D/HT66F26D

Register Name	Bit							
	7	6	5	4	3	2	1	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	—	—	—	—	—	—	D9	D8
TM1AL	D7	D6	D5	D4	D3	D2	D1	D0
TM1AH	—	—	—	—	—	—	D9	D8

10-bit Standard TM Register List (n=1 for HT66F24D; n=2 for HT66F26D)

TMnDL Register

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

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

TMn 10-bit Counter bit 7 ~ bit 0

TMnDH Register

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

Bit 7~2 "—" Unimplemented, read as 0

Bit 1~0 **TMnDH**: TMn Counter High Byte Register bit 1 ~ bit 0

TMn 10-bit Counter bit 9 ~ bit 8

TMnAL Register

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

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

TMn 10-bit CCRA bit 7 ~ bit 0

TMnAH Register

Register Name	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   **TMnAH**: TMn CCRA High Byte Register bit 1 ~ bit 0

TMn 10-bit CCRA bit 9 ~ bit 8

TMnC0 Register

Register Name	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   **TnPAU**: TMn Counter Pause Control

 0: Run

 1: Pause

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

Bit 6~4   **TnCK2~TnCK0**: Select TMn Counter clock

 000: $f_{SYS}/4$

 001: f_{SYS}

 010: $f_H/16$

 011: $f_H/64$

 100: f_{SUB}

 101: Undefined, can not be selected.

 110: TCKn rising edge clock

 111: TCKn falling edge clock

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

Bit 3   **TnON**: TMn Counter On/Off Control

 0: Off

 1: On

This bit controls the overall on/off function of the TM. Setting the bit high enables the counter to run 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.

If the TM is in the Compare Match Output Mode then the TM output pin will be reset to its initial condition, as specified by the TnOC bit, when the TnON bit changes from low to high.

Bit 2~0   **TnRP2~TnRP0**: TMn CCRP 3-bit register, compare with the TMn counter bit 9~bit 7

Comparator P Match Period

 000: 1024 TMn clocks

 001: 128 TMn clocks

 010: 256 TMn clocks

 011: 384 TMn clocks

 100: 512 TMn clocks

 101: 640 TMn clocks

 110: 768 TMn clocks

 111: 896 TMn clocks

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

TMnC1 Register

Register Name	Bit							
	7	6	5	4	3	2	1	0
Name	TnM1	TnM0	TnIO1	TnIO0	TnOC	TnPOL	TnDPX	TnCCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6   **TnM1~TnM0**: Select TMn Operation 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 TnM1 and TnM0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4   **TnIO1~TnIO0**: Select TPn_0, TPn_1, TPn_2 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 TPn_0, TPn_1, TPn_2

 01: Input capture at falling edge of TPn_0, TPn_1, TPn_2

 01: Input capture at falling/rising edge of TPn_0, TPn_1, TPn_2

 11: Input capture disabled

Timer/Counter Mode

 Unused.

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

In the Compare Match Output Mode, the TnIO1 and TnIO0 bits determine how the TM output pin changes state when a compare match occurs from the Comparator A. The TM output pin can be setup to switch high, switch low or to toggle its present state when a compare match occurs from the Comparator A. When the bits are both zero, then no change will take place on the output. The initial value of the TM output pin should be setup using the TnOC bit in the TMnC1 register. Note that the output level requested by the TnIO1 and TnIO0 bits must be different from the initial value setup using the TnOC bit otherwise no change will occur on the TM output pin when a compare match occurs. After the TM output pin changes state, it can be reset to its initial level by changing the level of the T1ON bit from low to high.

In the PWM Mode, the TnIO1 and TnIO0 bits determine how the TM output pin changes state when a certain compare match condition occurs. The PWM output function is modified by changing these two bits. It is necessary to change the values of the TnIO1 and TnIO0 bits only after the TM has been switched off. Unpredictable PWM outputs will occur if the TnIO1 and TnIO0 bits are changed when the TM is running

Bit 3   **TnOC**: TPn_0, TPn_1, TPn_2 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**: TPn_0, TPn_1, TPn_2 output Polarity control

  0: Non-invert

  1: Invert

This bit controls the polarity of the TPn_0, TPn_1 or TPn_2 output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1   **TnDPX**: TMn PWM period/duty Control

  0: CCRP – period; CCRA – duty

  1: CCRP – duty; CCRA – period

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

Bit 0   **TnCCLR**: Select TMn Counter clear condition

  0: TMn Comparator P match

  1: TMn Comparator 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 TnCCLR bit set high, the counter will be cleared when a compare match occurs from the Comparator A. When the bit is low, the counter will be cleared when a compare match occurs from the Comparator P or with a counter overflow. A counter overflow clearing method can only be implemented if the CCRP bits are all cleared to zero. The TnCCLR bit is not used in the PWM Mode, Single Pulse or Input Capture Mode.

Standard Type TM Operation Modes

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

Compare Match Output Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register, should be set to 00 respectively. In this mode once the counter is enabled and running it can be cleared by three methods. These are a counter overflow, a compare match from Comparator A and a compare match from Comparator P. When the TnCCLR bit is low, there are two ways in which the counter can be cleared. One is when a compare match from Comparator P, the other is when the CCRP bits are all zero which allows the counter to overflow. Here both TnAF and TnPF interrupt request flags for Comparator A and Comparator P respectively, will both be generated.

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

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

Compare Match Output Mode – TnCCLR=0

Note: 1. With TnCCLR=0, a Comparator P match will clear the counter

2. The TM output pin is controlled only by the TnAF flag

3. The output pin is reset to its initial state by a TnON bit rising edge

Compare Match Output Mode – TnCCLR=1

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

2. The TM output pin is controlled only by the TnAF flag

3. The output pin is reset to its initial state by a TnON bit rising edge

4. A TnPF flag is not generated when TnCCLR=1

Timer/Counter Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 11 respectively. The Timer/Counter Mode operates in an identical way to the Compare Match Output Mode generating the same interrupt flags. The exception is that in the Timer/Counter Mode the TM output pin is not used. Therefore the above description and Timing Diagrams for the Compare Match Output Mode can be used to understand its function. As the TM output pin is not used in this mode, the pin can be used as a normal I/O pin or other pin-shared function.

PWM Output Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 10 respectively. The PWM function within the TM is useful for applications which require functions such as motor control, heating control, illumination control etc. By providing a signal of fixed frequency but of varying duty cycle on the TM output pin, a square wave AC waveform can be generated with varying equivalent DC RMS values.

As both the period and duty cycle of the PWM waveform can be controlled, the choice of generated waveform is extremely flexible. In the PWM mode, the TnCCLR bit has no effect on the PWM operation. Both of the CCRA and CCRP registers are used to generate the PWM waveform, one register is used to clear the internal counter and thus control the PWM waveform frequency, while the other one is used to control the duty cycle. Which register is used to control either frequency or duty cycle is determined using the TnDPX bit in the TMnC1 register. The PWM waveform frequency and duty cycle can therefore be controlled by the values in the CCRA and CCRP registers.

An interrupt flag, one for each of the CCRA and CCRP, will be generated when a compare match occurs from either Comparator A or Comparator P. The TnOC bit in the TMnC1 register is used to select the required polarity of the PWM waveform while the two TnIO1 and TnIO0 bits are used to enable the PWM output or to force the TM output pin to a fixed high or low level. The TnPOL bit is used to reverse the polarity of the PWM output waveform.

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

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

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

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

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

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

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

The PWM output period cycle 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 running even when TnIO [1:0] = 00 or 01

4. The TnCCLR bit has no influence on PWM operation

PWM Mode – TnDPX=1

Note: 1. Here TnDPX=1 – Counter cleared by CCRA

2. A counter clear sets the PWM Period

3. The internal PWM function continues running even when TnIO [1:0] = 00 or 01

4. The TnCCLR bit has no influence on PWM operation

Single Pulse Mode

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

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

Single Pulse Generation
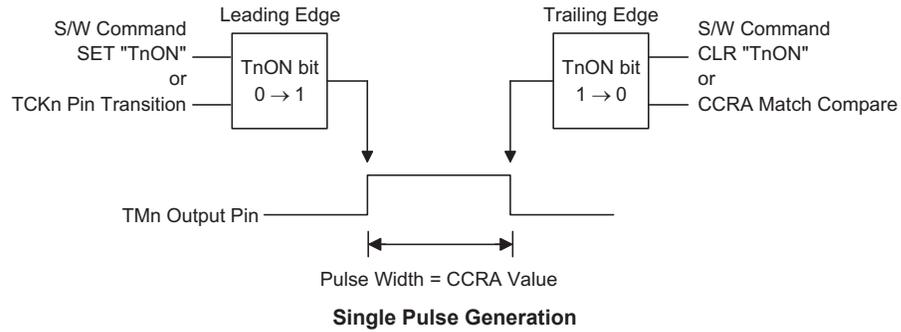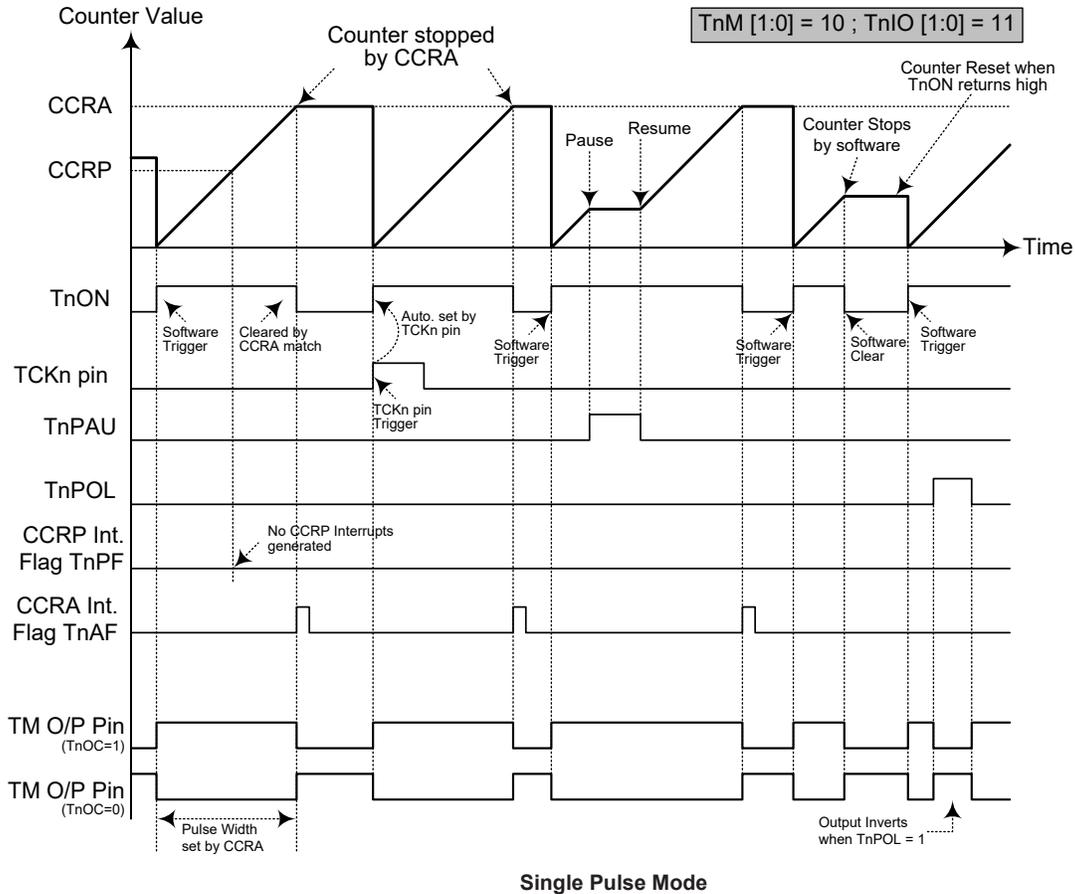
Single Pulse Mode

Note: 1. Counter stopped by CCRA

2. CCRP is not used

3. The pulse is triggered by the TCKn pin or by setting the TnON bit high

4. A TCKn pin active edge will automatically set the TnON bit high.

5. In the Single Pulse Mode, TnIO [1:0] must be set to "11" and can not be changed.

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

Capture Input Mode

To select this mode bits TnM1 and TnM0 in the TMnC1 register should be set to 01 respectively. This mode enables external signals to capture and store the present value of the internal counter and can therefore be used for applications such as pulse width measurements. The external signal is supplied on the TPn_0, TPn_1 or TPn_2 pin, whose active edge can be either a rising edge, a falling edge or both rising and falling edges; the active edge transition type is selected using the TnIO1 and TnIO0 bits in the TMnC1 register. The counter is started when the TnON bit changes from low to high which is initiated using the application program.

When the required edge transition appears on the TPn_0, TPn_1 or TPn_2 pin, the present value in the counter will be latched into the CCRA registers and a TM interrupt generated. Irrespective of what events occur on the TPn_0, TPn_1 or TPn_2 pin the counter will continue to free run until the TnON bit changes from high to low. When a CCRP compare match occurs, the counter will reset back to zero; in this way the CCRP value can be used to control the maximum counter value. When a CCRP compare match occurs from Comparator P, a TM interrupt will also be generated. Counting the number of overflow interrupt signals from the CCRP can be a useful method in measuring long pulse widths. The TnIO1 and TnIO0 bits can select the active trigger edge on the TPn_0, TPn_1 or TPn_2 pin to be a rising edge, falling edge or both edge types. If the TnIO1 and TnIO0 bits are both set high, then no capture operation will take place irrespective of what happens on the TPn_0, TPn_1 or TPn_2 pin, however it must be noted that the counter will continue to run.

As the TPn_0, TPn_1 or TPn_2 pin is pin shared with other functions, care must be taken if the TM is in the Input Capture Mode. This is because if the pin is setup as an output, then any transitions on this pin may cause an input capture operation to be executed. The TnCCLR and TnDPX bits are not used in this Mode.
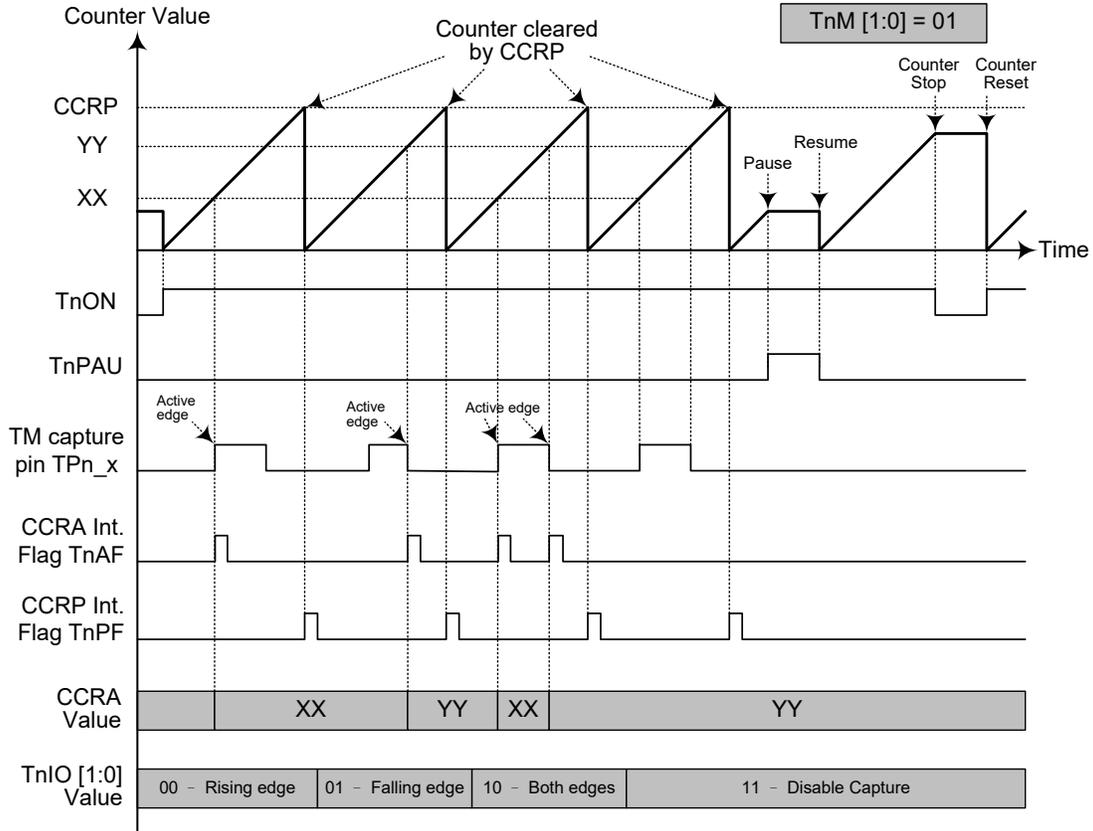
Capture Input Mode

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

2. A TM Capture input pin active edge transfers the counter value to CCRA

3. TnCCLR bit not used

4. No output function – TnOC and TnPOL bits are not used

5. CCRP determines the counter value and the counter has a maximum
count value when CCRP is equal to zero.
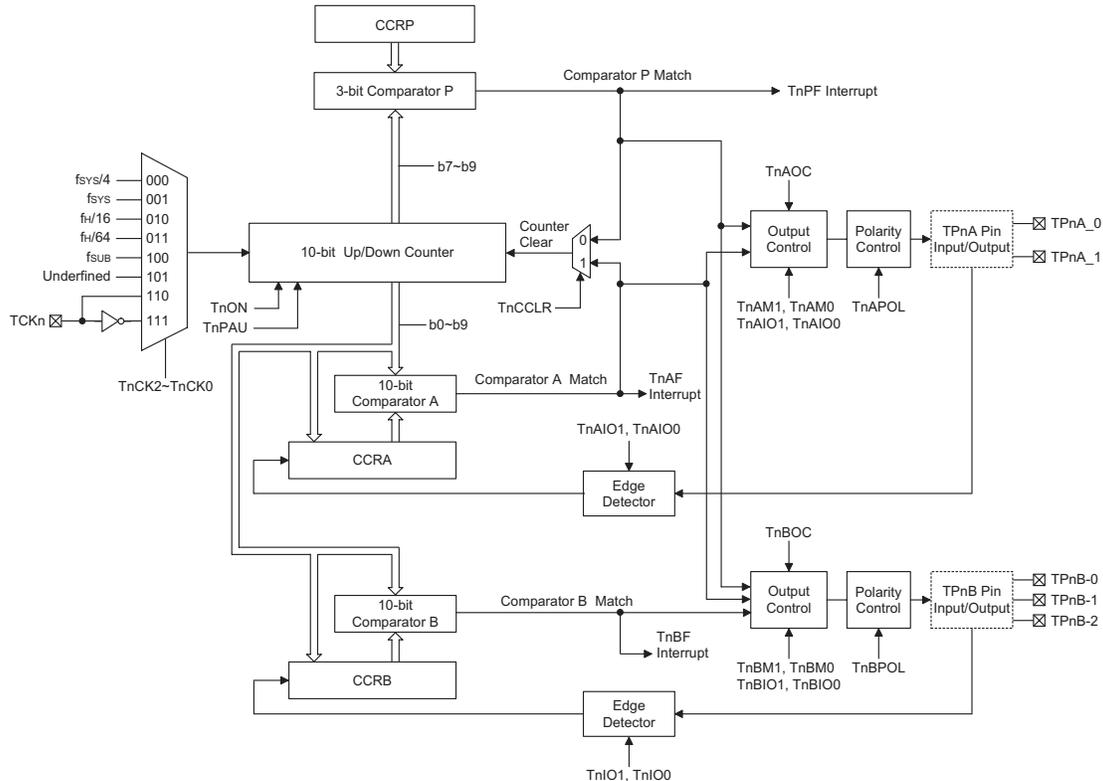
Enhanced Type TM – ETM

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

ETM	Name	TM No.	TM Input Pin	TM Output Pin
HT66F24D	—	—	—	—
HT66F25D/HT66F26D	10-bit ETM	1	TCK1	TP1A_0, TP1A_1 TP1B_0, TP1B_1, TP1B_2

Enhanced TM Operation

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

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

Enhanced Type TM Block Diagram

Enhanced Type TM Register Description

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

ETM Register List – HT66F25D/HT66F26D

Register	Bit							
Name	7	6	5	4	3	2	1	0
TM1C0	T1PAU	T1CK2	T1CK1	T1CK0	T1ON	T1RP2	T1RP1	T1RP0
TM1C1	T1AM1	T1AM0	T1AIO1	T1AIO0	T1AOC	T1APOL	T1CDN	T1CCLR
TM1C2	T1BM1	T1BM0	T1BIO1	T1BIO0	T1BOC	T1BPOL	T1PWM1	T1PWM0
TM1DL	D7	D6	D5	D4	D3	D2	D1	D0
TM1DH	—	—	—	—	—	—	D9	D8
TM1AL	D7	D6	D5	D4	D3	D2	D1	D0
TM1AH	—	—	—	—	—	—	D9	D8
TM1BL	D7	D6	D5	D4	D3	D2	D1	D0
TM1BH	—	—	—	—	—	—	D9	D8

10-bit Enhanced TM Register List

TM1DL Register

Register	Bit							
Name	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

Register	Bit							
Name	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

Register	Bit							
Name	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

Register Name	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

Register Name	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

Register Name	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

TM1C0 Register

Register Name	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_{SUB}

      101: Undefined, can not be selected.

      110: TCK1 rising edge clock

      111: TCK1 falling edge clock

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

Register Name	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 Operation 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_0, TP1A_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 TP1A_0, TP1A_1

    01: Input capture at falling edge of TP1A_0, TP1A_1

    01: Input capture at falling/rising edge of TP1A_0, TP1A_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 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_0, TP1A_1output 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_0, TP1A_1output Polarity control

  0: Non-invert

  1: Invert

This bit controls the polarity of the TP1A_0 or TP1A_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   **T1CDN**: TM1 Counter count up or down flag

  0: Count Up

  1: Count Down

Bit 0   **T1CCLR**: Select TM1 Counter clear condition

  0: TM1 Comparator P match

  1: TM1 Comparator A match

This bit is used to select the method which clears the counter. Remember that the Enhanced TM contains three comparators, Comparator A, Comparator B and Comparator P, but only Comparator A or Comparator 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 Single Pulse or Input Capture Mode.

TM1C2 Register

Register Name	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 Operation Mode

00: Compare Match Output Mode

01: Capture Input Mode

10: PWM Mode or Single Pulse Output Mode

11: Timer/Counter Mode

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

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

Compare Match Output Mode

00: No change

01: Output low

10: Output high

11: Toggle output

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, TP1B_2

01: Input capture at falling edge of TP1B_0, TP1B_1, TP1B_2

01: Input capture at falling/rising edge of TP1B_0, TP1B_1, TP1B_2

11: Input capture disabled

Timer/Counter Mode

Unused.

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

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

In the PWM Mode, the T1BIO1 and T1BIO0 bits determine how the TM output pin changes state when a certain compare match condition occurs. The PWM output function is modified by changing these two bits. It is necessary to change the values of the T1BIO1 and T1BIO0 bits only after the TM has been switched off. Unpredictable PWM outputs will occur if the T1BIO1 and T1BIO0 bits are changed when the TM is running

Bit 3   **T1BOC**: TP1B_0, TP1B_1, TP1B_2 output control bit

Compare Match Output Mode

  0: Initial low

  1: Initial high

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, TP1B_2 output Polarity control

  0: Non-invert

  1: Invert

This bit controls the polarity of the TP1B_0, TP1B_1, TP1B_2 output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1~0   **T1PWM1**~T1PWM0: Select PWM Mode

  00: Edge aligned

  01: Centre aligned, compare match on count up

  10: Centre aligned, compare match on count down

  11: Centre aligned, compare match on count up or down

Enhanced Type TM Operation Modes

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

ETM Operating Mode	CCRA Compare Match Output Mode	CCRA Timer/ Counter Mode	CCRB PWM Output Mode	CCRB Single Pulse Output Mode	CCRB 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 TnAM1, TnAM0 and TnBM1, TnBM0 in the TMnC1/TMnC2 registers should be all cleared to zero. In this mode once the counter is enabled and running it can be cleared by three methods. These are a counter overflow, a compare match from Comparator A and a compare match from Comparator P. When the TnCCLR bit is low, there are two ways in which the counter can be cleared. One is when a compare match occurs from Comparator P, the other is when the CCRP bits are all zero which allows the counter to overflow. Here both the TnAF and TnPF interrupt request flags for Comparator A and Comparator P respectively, will both be generated.

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

As the name of the mode suggests, after a comparison is made, the TM output pin, will change state. The TM output pin condition however only changes state when a TnAF or TnBF interrupt request flag is generated after a compare match occurs from Comparator A or Comparator B. The TnPF interrupt request flag, generated from a compare match from Comparator P, will have no effect on the TM output pin. The way in which the TM output pin changes state is determined by the condition of the TnAIO1 and TnAIO0 bits in the TMnC1 register for ETM CCRA, and the TnBIO1 and TnBIO0 bits in the TMnC2 register for ETM CCRB. The TM output pin can be selected using the TnAIO1, TnAIO0 bits (for the TPnA_0 or TPnA_1 pin) and TnBIO1, TnBIO0 bits (for the TPnB_0, TPnB_1 or TPnB_2 pins) to go high, to go low or to toggle from its present condition when a compare match occurs from Comparator A or a compare match occurs from Comparator B. The initial condition of the TM output pin, which is setup after the TnON bit changes from low to high, is setup using the TnAOC or TnBOC bit for TPnA_0, TPnA_1 or TPnB_0, TPnB_1, TPnB_2 output pins. Note that if the TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits are zero then no pin change will take place.
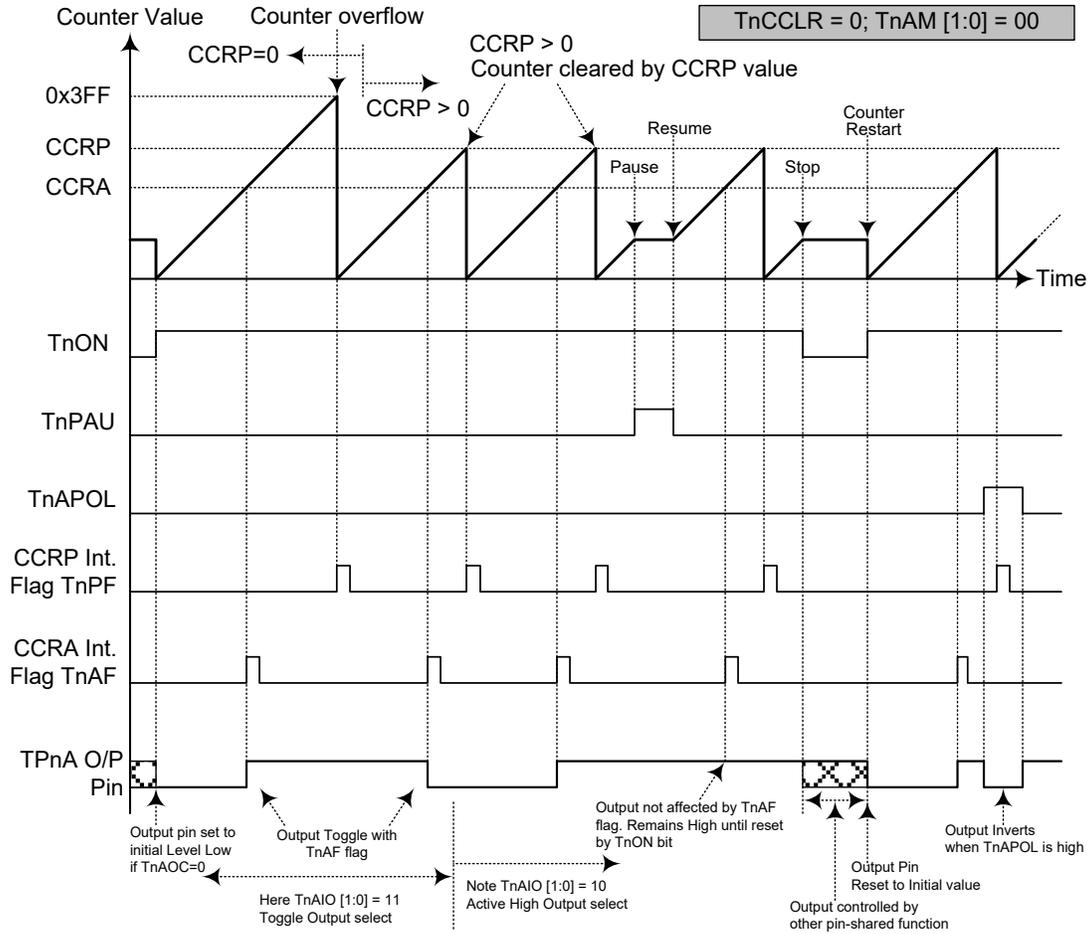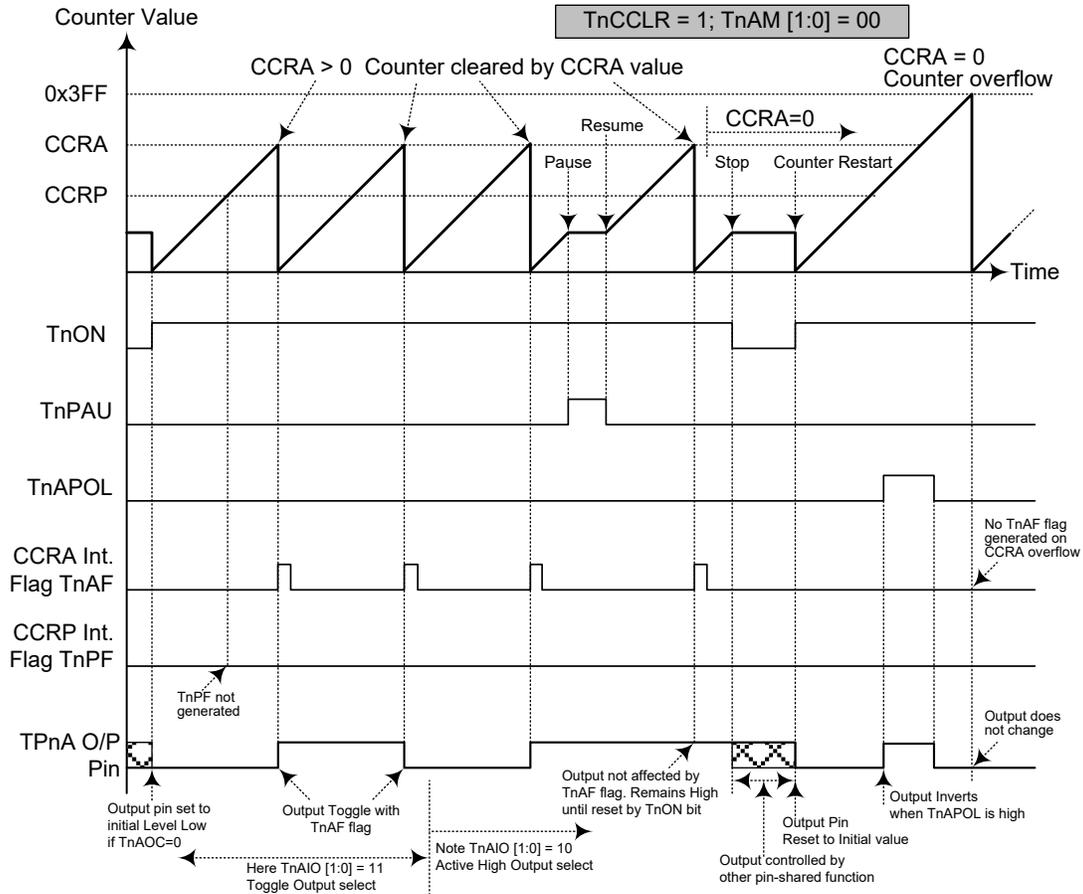
ETM CCRA Compare Match Output Mode – TnCCLR=0

Note: 1. With TnCCLR=0, a Comparator P match will clear the counter

2. The TPnA output pin is controlled only by the TnAF flag

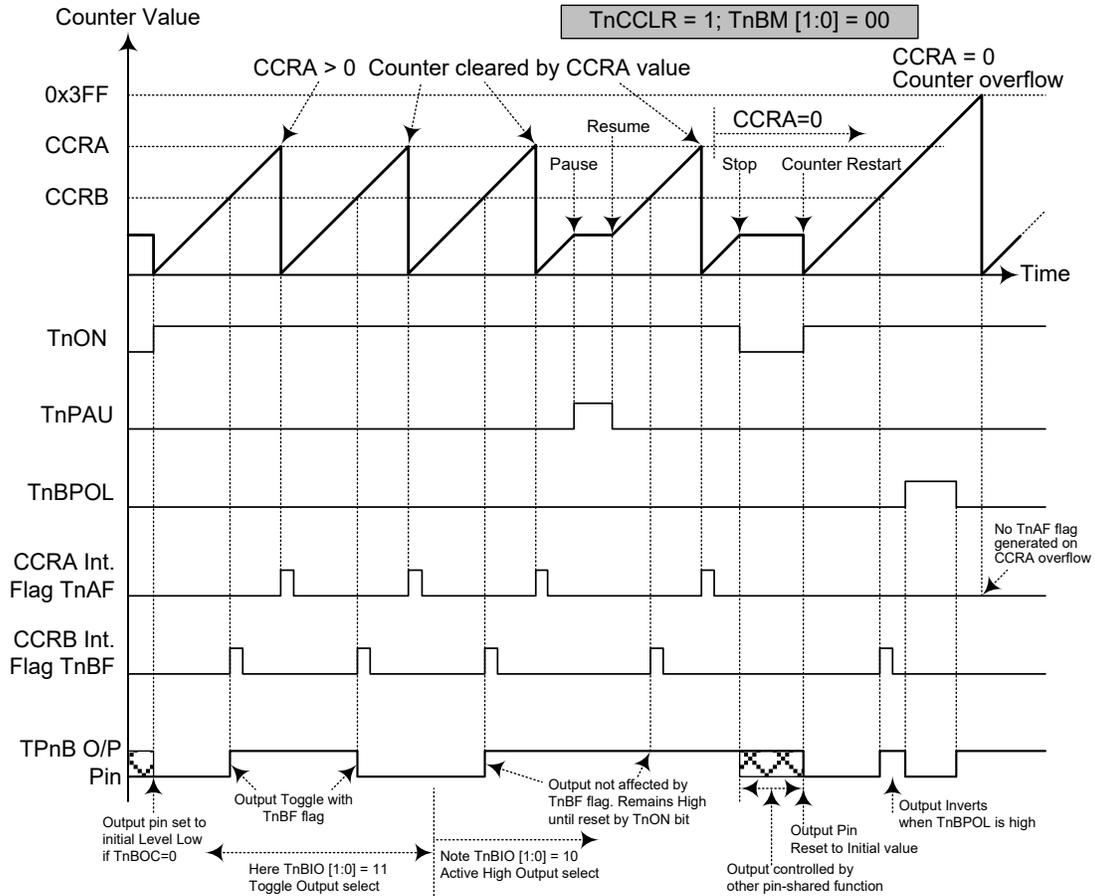3. The output pin is reset to its initial state by a TnON bit rising edge
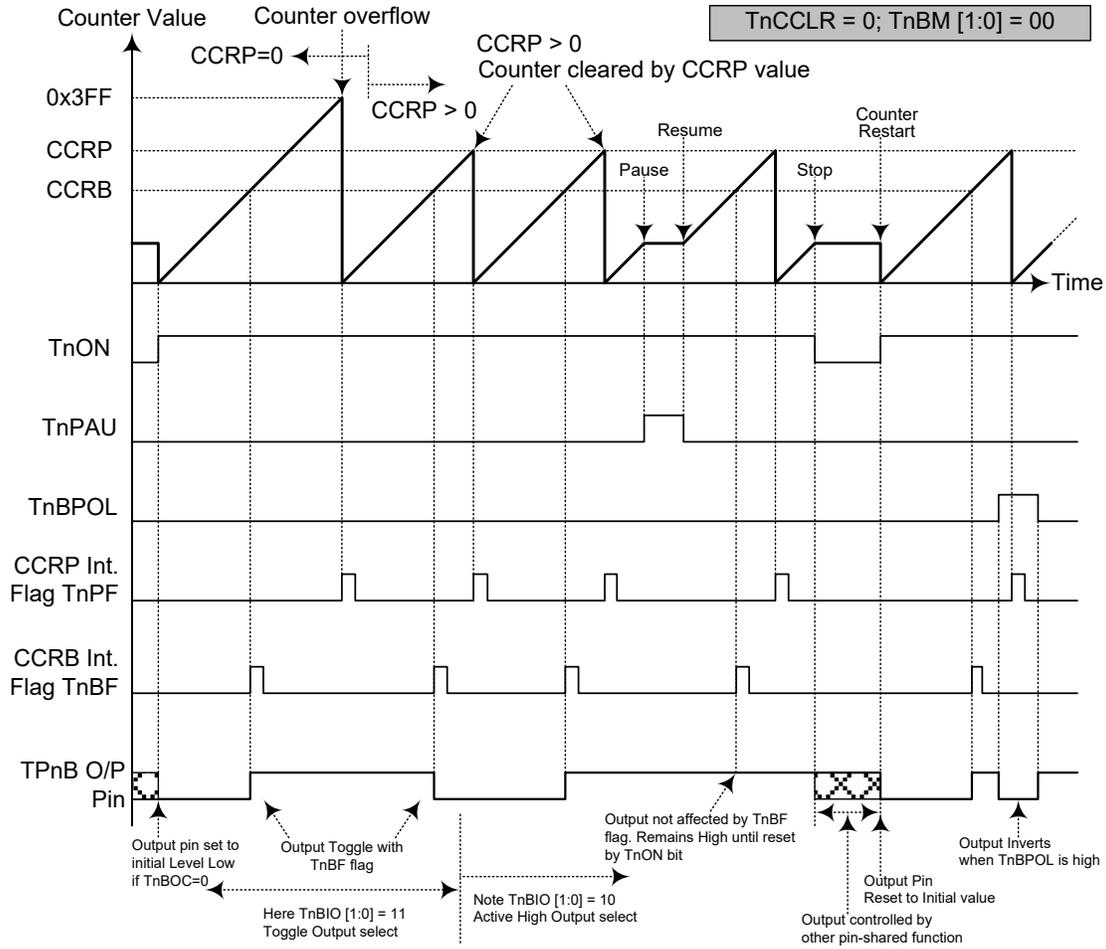
ETM CCRB Compare Match Output Mode – TnCCLR=0

Note: 1. With TnCCLR=0, a Comparator P match will clear the counter

2. The TPnB output pin is controlled only by the TnBF flag

3. The output pin is reset to its initial state by a TnON bit rising edge

ETM CCRA Compare Match Output Mode – TnCCLR=1

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

2. The TPnA output pin is controlled only by the TnAF flag

3. The TPnA output pin is reset to its initial state by a TnON bit rising edge

4. The TnPF flag is not generated when TnCCLR=1

ETM CCRB Compare Match Output Mode – TnCCLR=1

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

2. The TPnB output pin is controlled only by the TnBF flag

3. The TPnB output pin is reset to its initial state by a TnON bit rising edge

4. The TnPF flag is not generated when TnCCLR=1

Timer/Counter Mode

To select this mode, bits TnAM1, TnAM0 and TnBM1, TnBM0 in the TMnC1 and TMnC2 register should all be set high. The Timer/Counter Mode operates in an identical way to the Compare Match Output Mode generating the same interrupt flags. The exception is that in the Timer/Counter Mode the TM output pin is not used. Therefore the above description and Timing Diagrams for the Compare Match Output Mode can be used to understand its function. As the TM output pin is not used in this mode, the pin can be used as a normal I/O pin or other pin-shared function.

PWM Output Mode

To select this mode, the required bit pairs, TnAM1, TnAM0 and TnBM1, TnBM0 should be set to 10 respectively and also the TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits should be set to 10 respectively. The PWM function within the TM is useful for applications which require functions such as motor control, heating control, illumination control etc. By providing a signal of fixed frequency but of varying duty cycle on the TM output pin, a square wave AC waveform can be generated with varying equivalent DC RMS values.

As both the period and duty cycle of the PWM waveform can be controlled, the choice of generated waveform is extremely flexible. In the PWM mode, the TnCCLR bit is used to determine in which way the PWM period is controlled. With the TnCCLR bit set high, the PWM period can be finely controlled using the CCRA registers. In this case the CCRB registers are used to set the PWM duty value (for TPnB output pins). The CCRP bits are not used and TPnA output pins are not used. The PWM output can only be generated on the TPnB output pins. With the TnCCLR bit cleared to zero, the PWM period is set using one of the eight values of the three CCRP bits, in multiples of 128. Now both CCRA and CCRB registers can be used to setup different duty cycle values to provide dual PWM outputs on their relative TPnA and TPnB pins.

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

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

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

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

If f_{SYS} = 16MHz, TM clock source select $f_{SYS}/4$, CCRP = 100b, CCRA = 128 and CCRB = 256,

The TP1A PWM output frequency = ($f_{SYS}/4$) / 512 = $f_{SYS}/2048$ = 7.8125kHz, duty = 128/512 = 25%.

The TP1B_n PWM output frequency = ($f_{SYS}/4$) / 512 = $f_{SYS}/2048$ = 7.8125kHz, duty = 256/512 = 50%.

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

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

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

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

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

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

CCRA	1	2	3	511	512	1021	1022	1023
Period	2	4	6	1022	1024	2042	2044	2046
B Duty	(CCRB × 2) - 1							
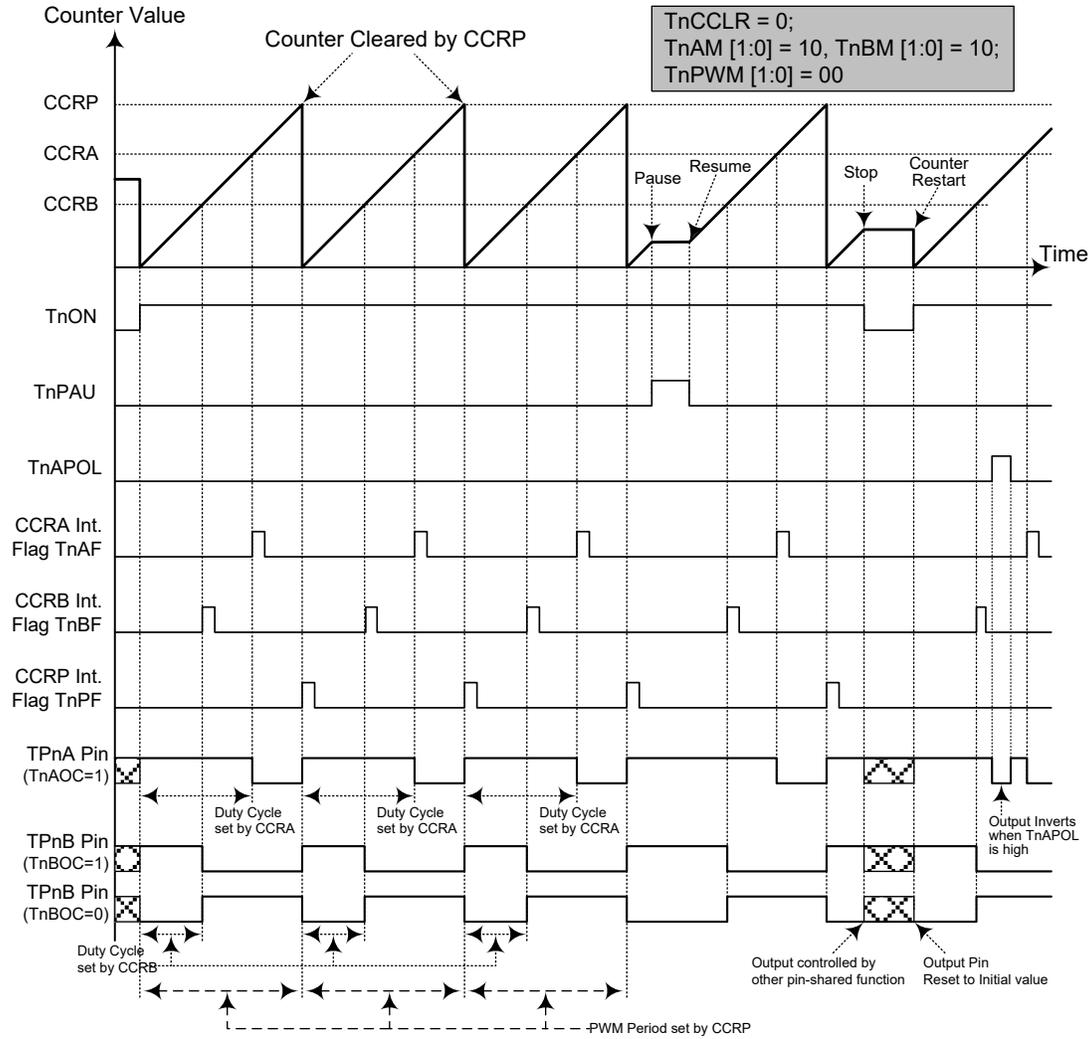
ETM PWM Mode – Edge Aligned

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

2. The internal PWM function continues running even when TnAIO [1:0] (or TnBIO [1:0]) = 00 or 01

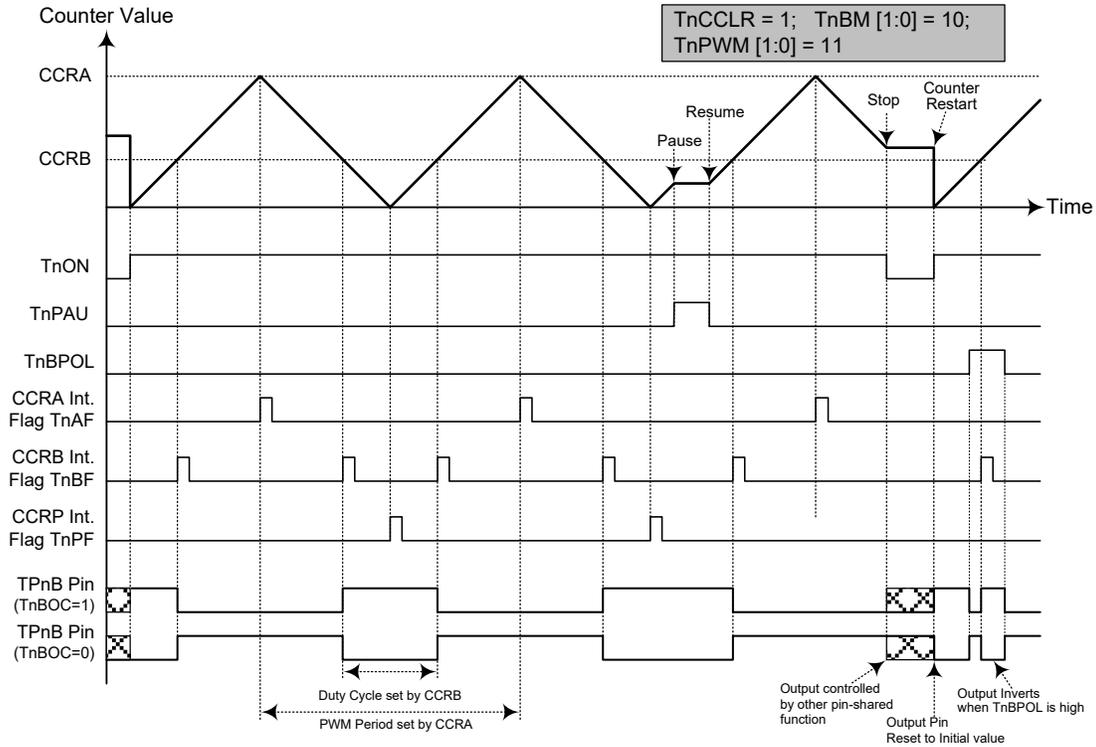3. CCRA controls the TPnA PWM duty and CCRB controls the TPnB PWM duty
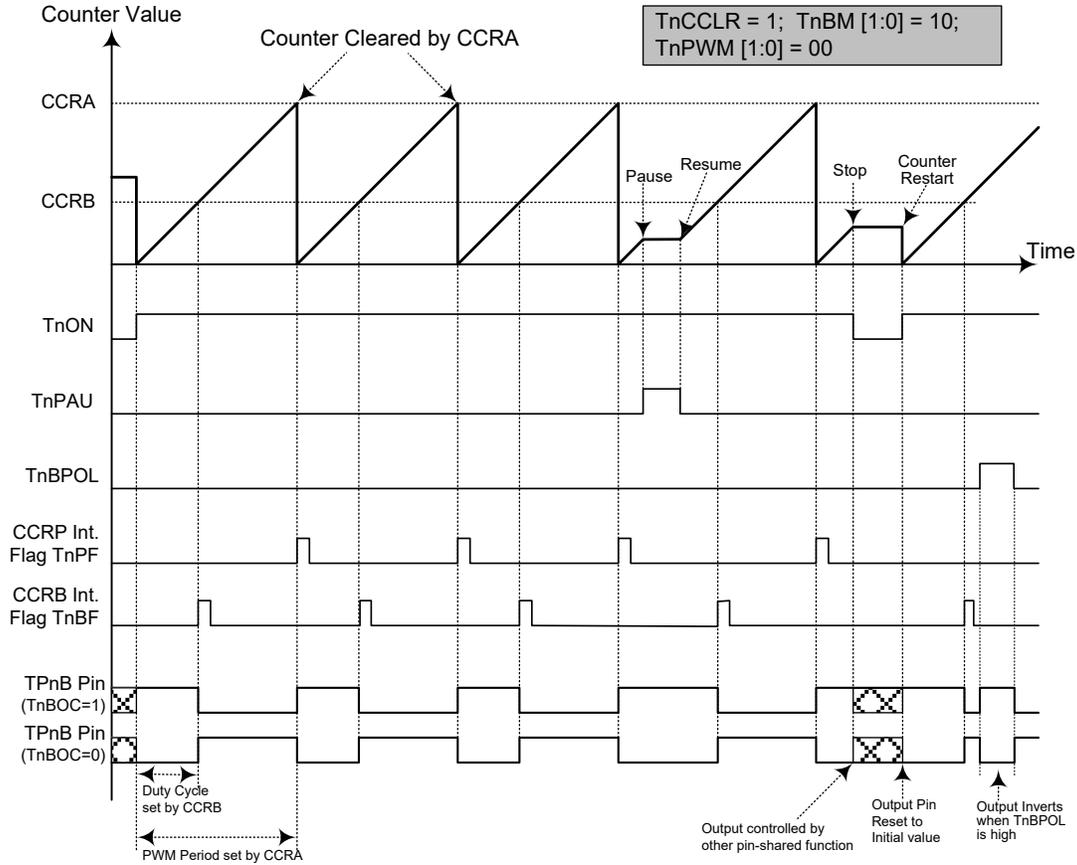
ETM PWM Mode – Edge Aligned

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

2. The internal PWM function continues running even when TnBIO [1:0] = 00 or 01

3. CCRA controls the TPnB PWM period and CCRB controls the TPnB PWM duty

4. Here the TM pin control register should not enable the TPnA pin as a TM output pin.
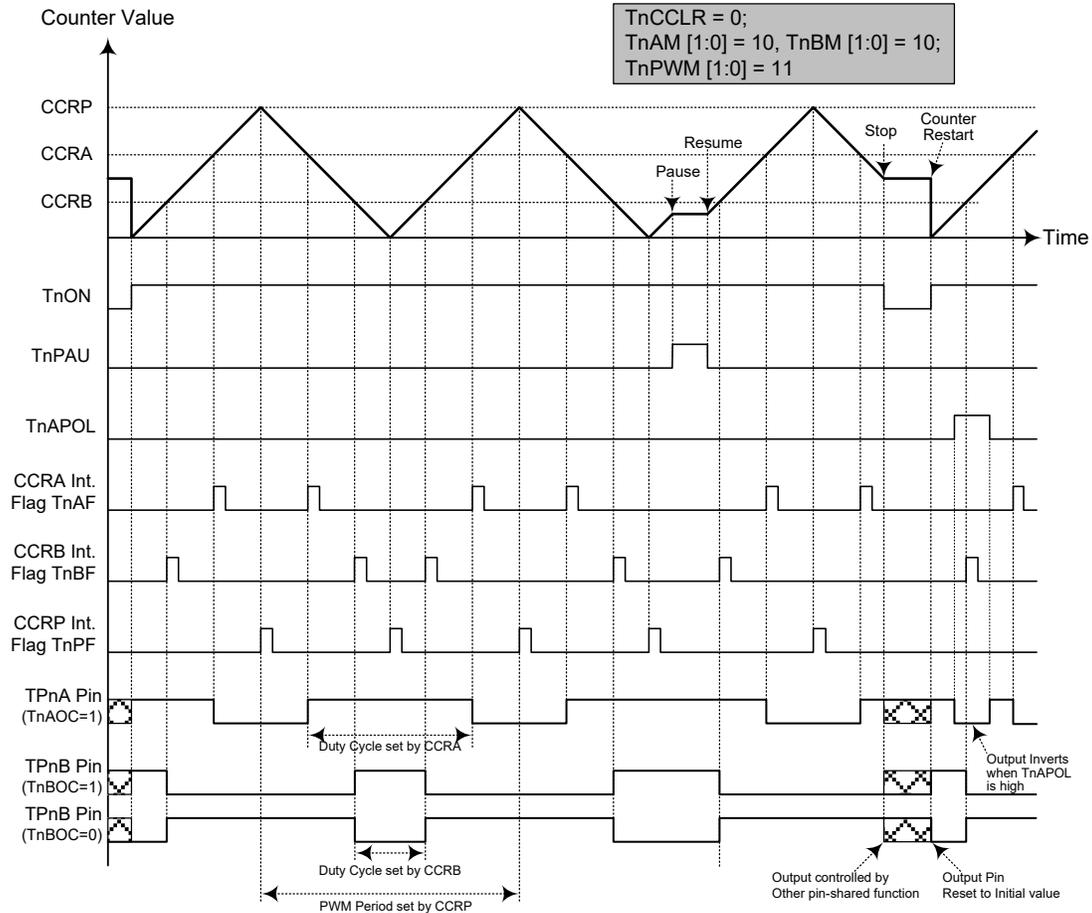
ETM PWM Mode – Centre Aligned

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

2. TnPWM [1:0] =11 therefore the PWM is centre aligned

3. The internal PWM function continues running even when TnAIO [1:0] (or TnBIO [1:0]) = 00 or 01

4. CCRA controls the TPnA PWM duty and CCRB controls the TPnB PWM duty

5. CCRP will generate an interrupt request when the counter decrements to its zero value

ETM PWM Mode – Centre Aligned

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

2. TnPWM [1:0] =11 therefore the PWM is centre aligned

3. The internal PWM function continues running even when TnBIO [1:0] = 00 or 01

4. CCRA controls the TPnB PWM period and CCRB controls the TPnB PWM duty

5. CCRP will generate an interrupt request when the counter decrements to its zero value

Single Pulse Output Mode

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

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

However a compare match from Comparator A will also automatically clear the TnON bit and thus generate the Single Pulse output trailing edge of TPnA and TPnB. In this way the CCRA value can be used to control the pulse width of TPnA. The CCRA-CCRB value can be used to control the pulse width of TPnB. A compare match from Comparator A and Comparator B will also generate TM interrupts. The counter can only be reset back to zero when the TnON bit changes from low to high when the counter restarts. In the Single Pulse Mode CCRP is not used. The TnCCLR bit is also not used.
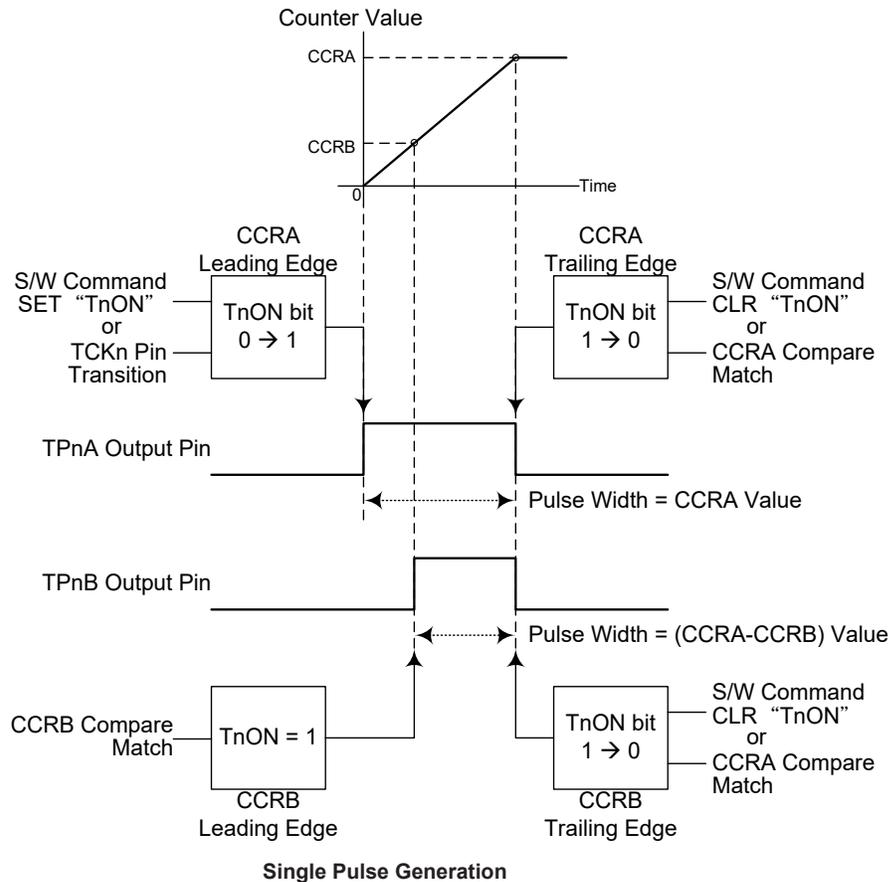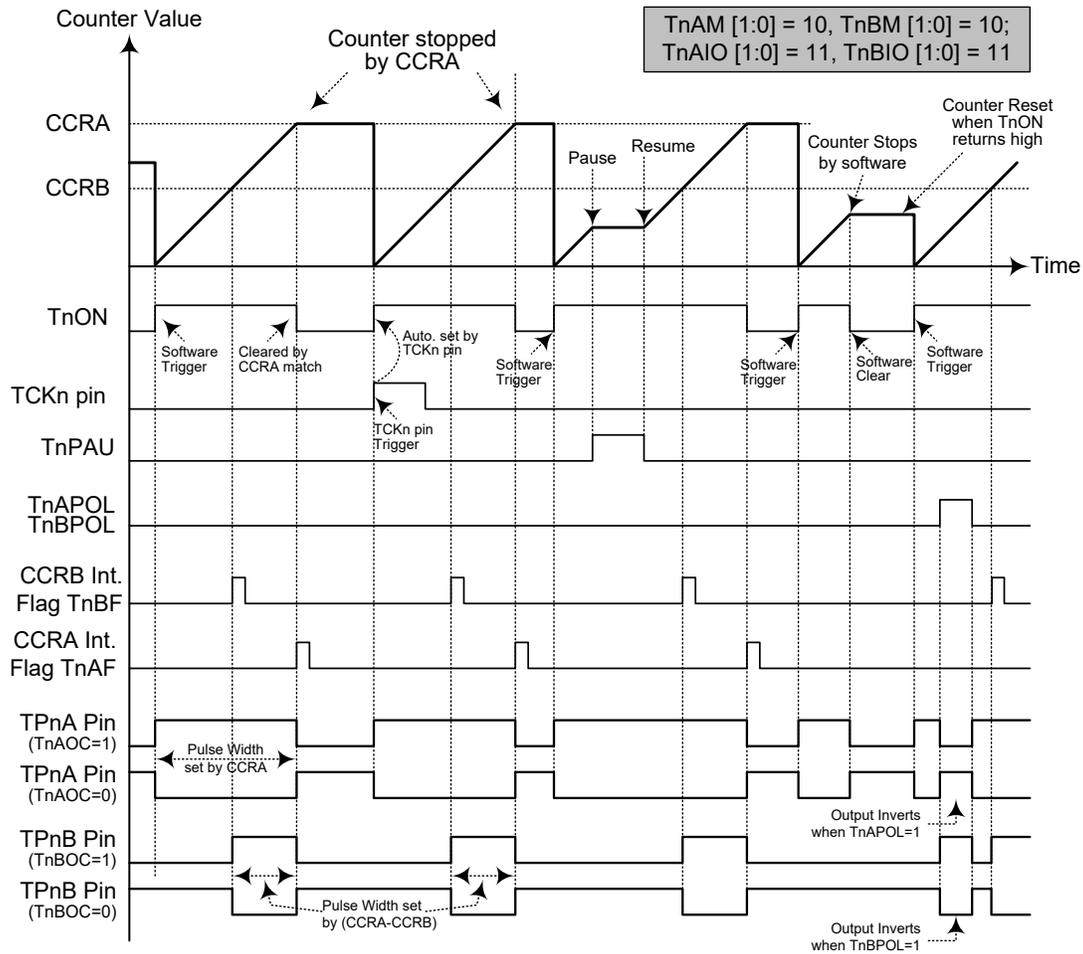

Single Pulse Generation

Single Pulse Mode

Note: 1. Counter stopped by CCRA

2. CCRP is not used

3. The pulse is triggered by the TCKn pin or by setting the TnON bit high

4. A TCKn pin active edge will automatically set by the TnON bit high.

5. In the Single Pulse Mode, TnAIO [1:0] and TnBIO [1:0] must be set to "11" and can not be changed.

Capture Input Mode

To select this mode bits TnAM1, TnAM0 and TnBM1, TnBM0 in the TMnC1 and TMnC2 registers should be set to 01 respectively. This mode enables external signals to capture and store the present value of the internal counter and can therefore be used for applications such as pulse width measurements. The external signal is supplied on the TPnA_0, TPnA_1 and TPnB_0, TPnB_1, TPnB_2 pins, whose active edge can be either a rising edge, a falling edge or both rising and falling edges; the active edge transition type is selected using the TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits in the TMnC1 and TMnC2 registers. The counter is started when the TnON bit changes from low to high which is initiated using the application program.

When the required edge transition appears on the TPnA_0, TPnA_1 and TPnB_0, TPnB_1, TPnB_2 pins the present value in the counter will be latched into the CCRA and CCRB registers and a TM interrupt generated. Irrespective of what events occur on the TPnA_0, TPnA_1 and TPnB_0, TPnB_1, TPnB_2 pins the counter will continue to free run until the TnON bit changes from high to low. When a CCRP compare match occurs the counter will reset back to zero; in this way the CCRP value can be used to control the maximum counter value. When a CCRP compare match occurs from Comparator P, a TM interrupt will also be generated. Counting the number of overflow interrupt signals from the CCRP can be a useful method in measuring long pulse widths. The TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits can select the active trigger edge on the TPnA_0, TPnA_1 and TPnB_0, TPnB_1, TPnB_2 pins to be a rising edge, falling edge or both edge types. If the TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits are both set high, then no capture operation will take place irrespective of what happens on the TPnA_0, TPnA_1 and TPnB_0, TPnB_1, TPnB_2 pins, however it must be noted that the counter will continue to run.

As the TPnA_0, TPnA_1 and TPnB_0, TPnB_1, TPnB_2 pins are pin shared with other functions, care must be taken if the TM is in the Capture Input Mode. This is because if the pin is setup as an output, then any transitions on this pin may cause an input capture operation to be executed. The TnCCLR, TnAOC, TnBOC, TnAPOL and TnBPOL bits are not used in this mode.
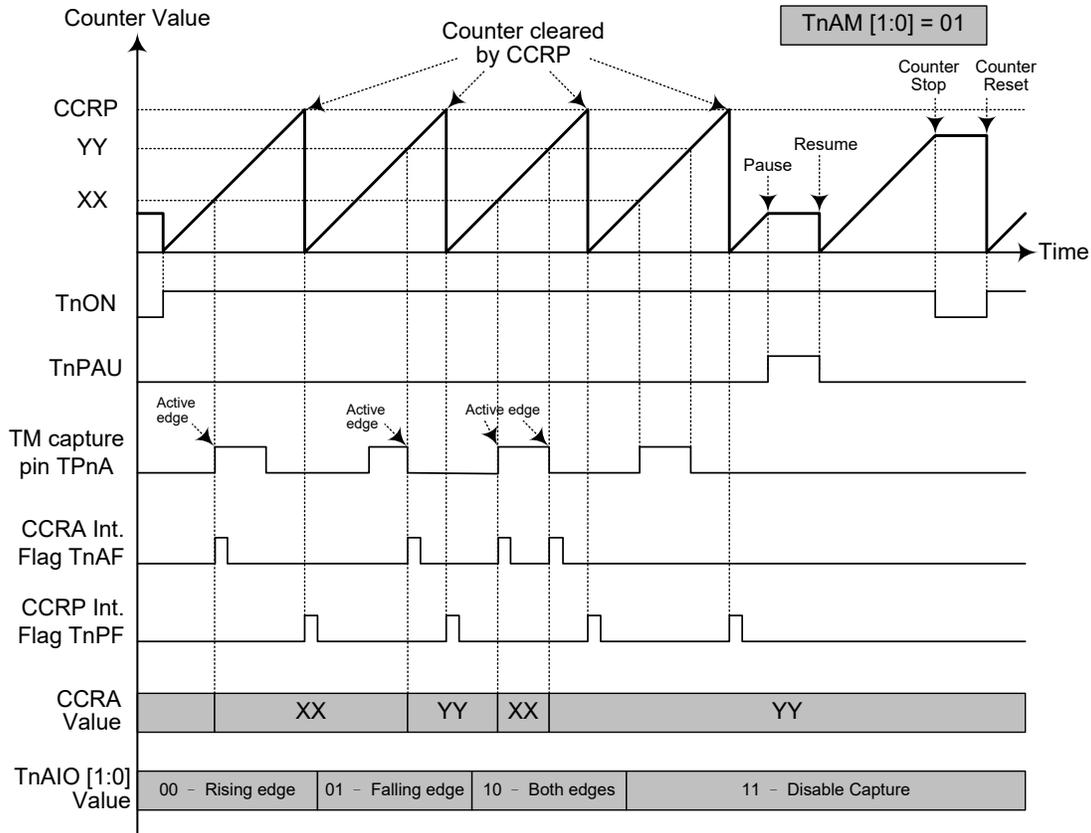
ETM CCRA Capture Input Mode

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

2. The TM Capture input pin active edge transfers the 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.
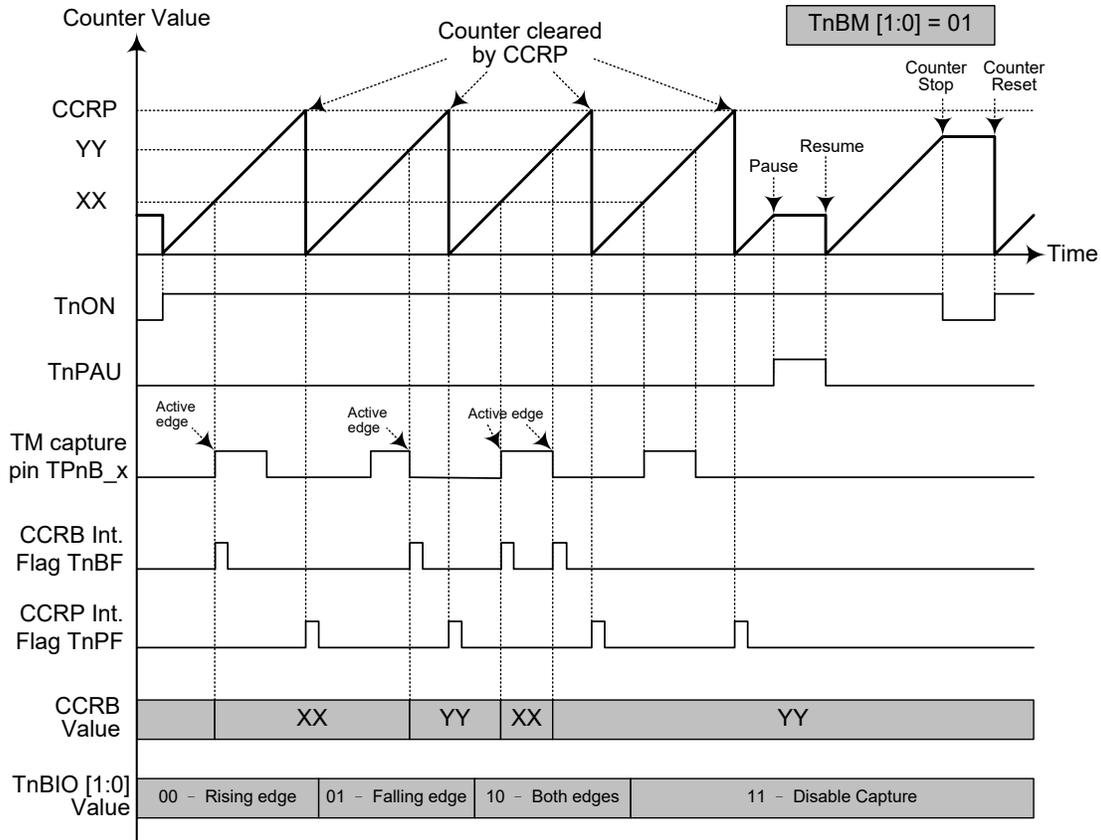
ETM CCRB Capture Input Mode

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

2. The TM Capture input pin active edge transfers the counter value to CCRB

3. TnCCLR bit not used

4. No output function – TnBOC and TnBPOL bits not used

5. CCRP determines the counter value and the counter has a maximum count value when CCRP is equal to zero.

Analog to Digital Converter

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

A/D Overview

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

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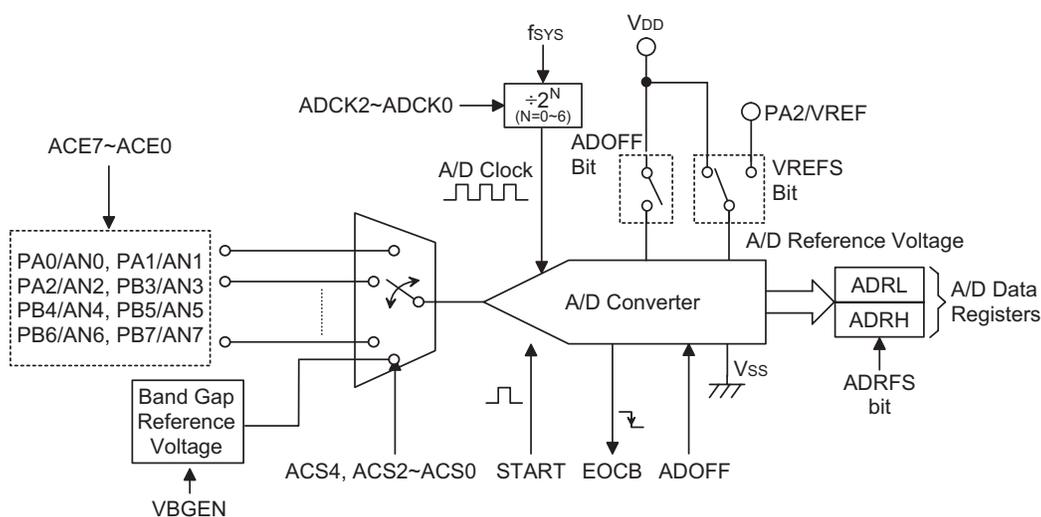
A/D Converter Structure

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	VBGEN	—	VREFS	—	ADCK2	ADCK1	ADCK0
ACERL	ACE7	ACE6	ACE5	ACE4	ACE3	ACE2	ACE1	ACE0

A/D Converter Register List

A/D Converter Data Registers – ADRL, ADRH

As the devices contain an internal 12-bit A/D converter, they require 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 device contains only one actual analog to digital converter hardware circuit, each of the individual 4 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 reference voltage, V_{BG}, is actually connected to the internal A/D converter.

The ACERL control register contains the ACER7~ACER0 bits which determine which pins on I/O ports are used as analog inputs for the A/D converter input and which pins are not to be used as the A/D converter input. 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

$0 \rightarrow 1 \rightarrow 0$: start

$0 \rightarrow 1$: 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**: A/D module on/off control bit

0: A/D module power on

1: A/D 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**: A/D Data Format control bit

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~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

ADCR1 Register

Bit	7	6	5	4	3	2	1	0
Name	ACS4	VBGEN	—	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 band gap reference voltage as ADC input Control

0: Disable

1: Enable

This bit enables the internal band gap reference voltage to be connected to the A/D converter. The VBGEN bit must first have been set to enable the band gap circuit reference voltage to be used by the A/D converter. When the ACS4 bit is set high, the band gap reference voltage will be routed to the A/D converter and the other A/D input channels disconnected.

Bit 6   **VBGEN**: Internal Band Gap reference voltage 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 band gap voltage can be used by the A/D converter. If the band gap reference voltage is not used by the A/D converter and the LVR/LVD function is disabled then the band gap reference circuit will be automatically switched off to conserve power. When the band gap reference voltage is switched on for use by the A/D converter, a time t_{BG} should be allowed for the band gap 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~ADCK0**: Select ADC converter 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, can not be used.

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 I/O pin is A/D input or not
    0: Not A/D input
    1: A/D input, AN7

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

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

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

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

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

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

Bit 0   **ACE0**: Define I/O pin 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 clock f_{SYS}, and by bits ADCK2~ADCK0, there are some limitations on the A/D clock source speed range that can be selected. As the recommended range of permissible A/D clock period, t_{ADCK}, is from 0.5μs to 10μs, care must be taken for selected system clock frequencies. For example, if the system clock operates at a frequency of 4MHz, the ADCK2~ADCK0 bits should not be set to 000B or 110B. Doing so will give A/D clock periods that are less than the minimum A/D clock period or greater than the maximum 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~ ADCK0 = 000 (f_{SYS})	ADCK2~ ADCK0 = 001(f_{SYS}/2)	ADCK2~ ADCK0 = 010(f_{SYS}/4)	ADCK2~ ADCK0 = 011(f_{SYS}/8)	ADCK2~ ADCK0 = 100(f_{SYS}/16)	ADCK2~ ADCK0 = 101(f_{SYS}/32)	ADCK2~ ADCK0 = 110(f_{SYS}/64)	ADCK2~ 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. When the ADOFF bit is cleared to zero to power on the A/D converter internal circuitry a certain delay, as indicated in the timing diagram, must be allowed before an A/D conversion is initiated. Even if no pins are selected for use as A/D inputs by clearing the ACE7~ACE0 bits in the ACERL register, 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 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 as well as other functions. The ACE7~ACE0 bits in the ACERL register determines 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 register 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 VREF.


A/D Input Structure
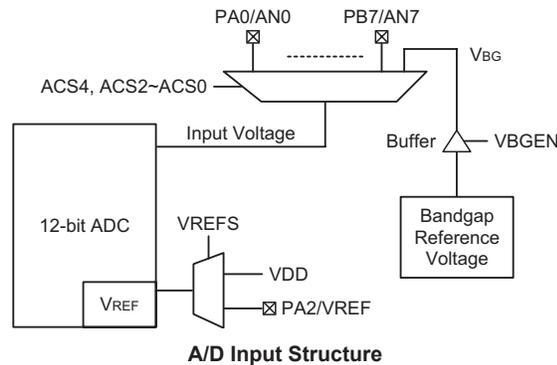
Summary of A/D Conversion Steps

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 and ACS2~ACS0 bits which are also contained in the ADCR1 and ADCR0 registers.

- 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 to 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 to low again. Note that this bit should have been originally cleared to 0.

- 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 registers 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 ADCR 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 16 t_{ADCK} where t_{ADCK} is equal to the A/D clock period.

A/D Conversion Timing
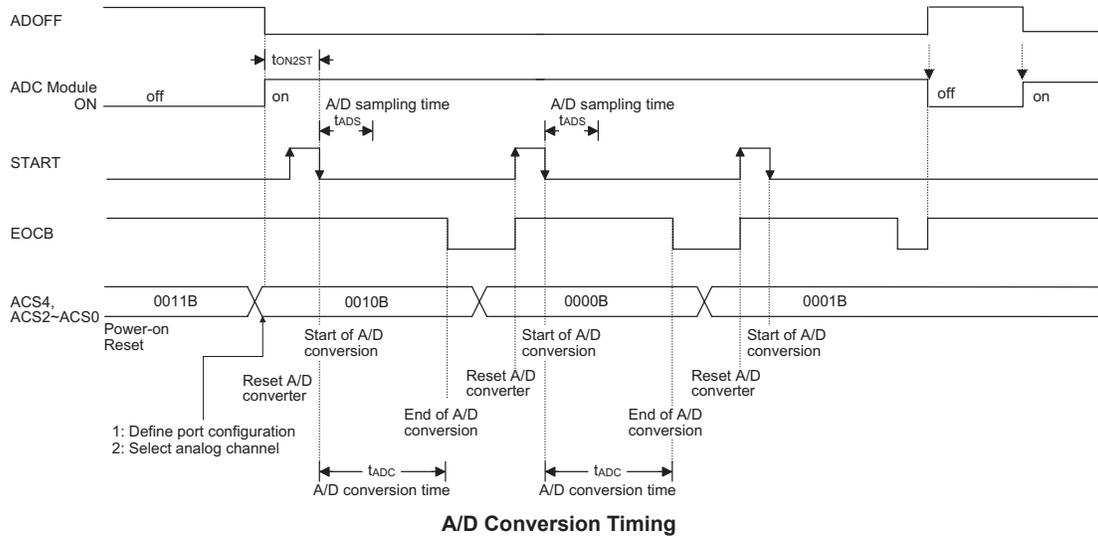
Programming Considerations

During microcontroller operates 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.
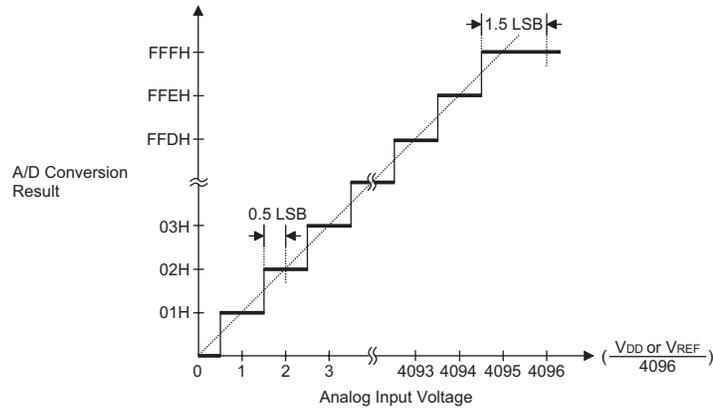
A/D Transfer Function

As the devices contain 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 VDD or V_{REF} voltage, this gives a single bit analog input value of V_{DD} or V_{REF} divided by 4096.

```
1 LSB = (V_DD or V_REF) ÷ 4096
```

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

```
A/D input voltage = A/D output digital value × (V_DD or V_REF) ÷ 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} level.

Ideal A/D Transfer Function

A/D Programming Examples

The following two programming examples illustrate how to setup and implement an A/D conversion. In the first example, the method of polling the 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 1: 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 the band gap
                        ; reference voltage
clr  ADOFF
mov  a,0Fh              ; setup ACERL register 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 2: using the interrupt method to detect the end of conversion

```
    clr  ADE                ; disable ADC interrupt
    mov  a,03H
    mov  ADCR1,a            ; select f_SYS/8 as A/D clock and switch off the band gap reference voltage
    clr  ADOFF
    mov  a,0Fh              ; setup ACERL register 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
```

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. The device contains several external interrupt and internal interrupts functions. The external interrupts are generated by the action of the external INT0 and INT1 pins while the internal interrupts are generated by various internal functions such as the TMs, Time Base, LVD 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~MFI3 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	—	—
INTn Pins	INTnE	INTnF	n = 0 or 1
Multi-function	MFnE	MFnF	n = 0 ~3
A/D converter	ADE	ADF	—
Time Base	TBnE	TBnF	n = 0 or 1
LVD	LVE	LVF	—
EEPROM	DEE	DEF	—
TM	TnPE	TnPF	n = 0 or 1 or 2
	TnAE	TnAF	
	TnBE	TnBF	n = 1

Interrupt Register Bit Naming Conventions

HT66F24D Interrupt Register List

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

HT66F25D Interrupt Register List

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

HT66F26D Interrupt Register List

Register Name	Bit							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	MF0F	INT1F	INT0F	MF0E	INT1E	INT0E	EMI
INTC1	TB1F	TB0F	ADF	MF1F	TB1E	TB0E	ADE	MF1E
INTC2	—	—	MF3F	MF2F	—	—	MF3E	MF2E
MFI0	—	—	T0AF	T0PF	—	—	T0AE	T0PE
MFI1	—	T1BF	T1AF	T1PF	—	T1BE	T1AE	T1PE
MFI2	—	—	DEF	LVF	—	—	DEE	LVE
MFI3	—	—	T2AF	T2PF	—	—	T2AE	T2PE

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
     01: both rising and falling edges

Bit 1~0   **INT0S1~INT0S0**: interrupt edge control for INT0 pin
     00: disable
     01: rising edge
     10: falling edge
     01: both rising and falling edges

INTC0 Register

Bit	7	6	5	4	3	2	1	0
Name	—	MF0F	INT1F	INT0F	MF0E	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   **MF0F**: Multi-function 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   **MF0E**: Multi-function 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

Bit	7	6	5	4	3	2	1	0
Name	TB1F	TB0F	ADF	MF1F	TB1E	TB0E	ADE	MF1E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7   **TB1F**: Time Base 1 Interrupt request flag
       0: no request
       1: interrupt request

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

Bit 5   **ADF**: A/D Converter interrupt request flag
       0: no request
       1: interrupt request

Bit 4   **MF1F**: Multi-function 1 Interrupt request flag
       0: no request
       1: interrupt request

Bit 3   **TB1E**: Time Base 1 Interrupt control
       0: disable
       1: enable

Bit 2   **TB0E**: Time Base 0 Interrupt control
       0: disable
       1: enable

Bit 1   **ADE**: A/D Converter interrupt control
       0: disable
       1: enable

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

INTC2 Register - HT66F24D/HT66F25D

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	MF2F	—	—	—	MF2E
R/W	—	—	—	R/W	—	—	—	R/W
POR	—	—	—	0	—	—	—	0

Bit 7~5   unimplemented, read as "0"

Bit 4   **MF2F**: Multi-function 2 Interrupt request flag
       0: no request
       1: interrupt request

Bit 3~1   unimplemented, read as "0"

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

INTC2 Register - HT66F26D

Bit	7	6	5	4	3	2	1	0
Name	—	—	MF3F	MF2F	—	—	MF3E	MF2E
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

Bit 7~6   unimplemented, read as "0"

Bit 5   **MF3F**: Multi-function 3 Interrupt request flag
                 0: no request
                 1: interrupt request

Bit 4   **MF2F**: Multi-function 2 Interrupt request flag
                 0: no request
                 1: interrupt request

Bit 3~2   unimplemented, read as "0"

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

Bit 0   **MF2E**: Multi-function 2 Interrupt 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 - HT66F24D

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

MFI1 Register - HT66F25D/HT66F26D

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
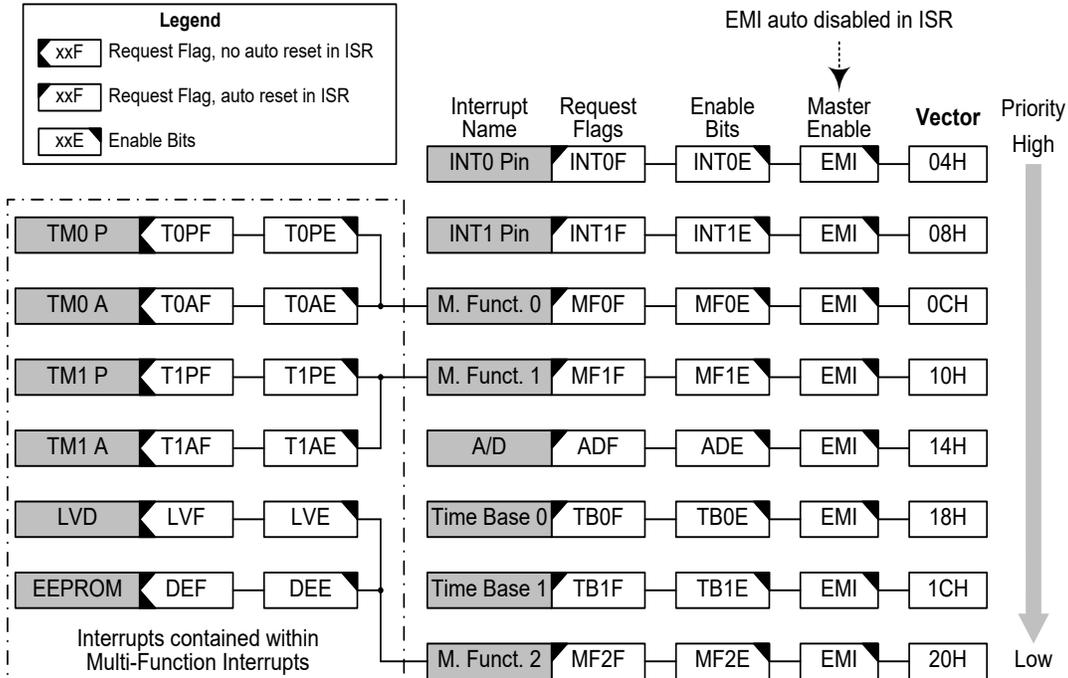   0: disable
       1: enable

MFI2 Register

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

Bit 7~6   unimplemented, read as "0"

Bit 5   **DEF**: Data EEPROM Interrupt request flag
  0: no request
  1: interrupt request

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

Bit 3~2   unimplemented, read as "0"

Bit 1   **DEE**: Data EEPROM Interrupt control
  0: disable
  1: enable

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

MFI3 Register - HT66F26D

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

Bit 7~6   unimplemented, read as "0"

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

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

Bit 3~2   unimplemented, read as "0"

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

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

Interrupt Operation

When the conditions for an interrupt event occur, such as a TM Comparator P, Comparator A or Comparator 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.
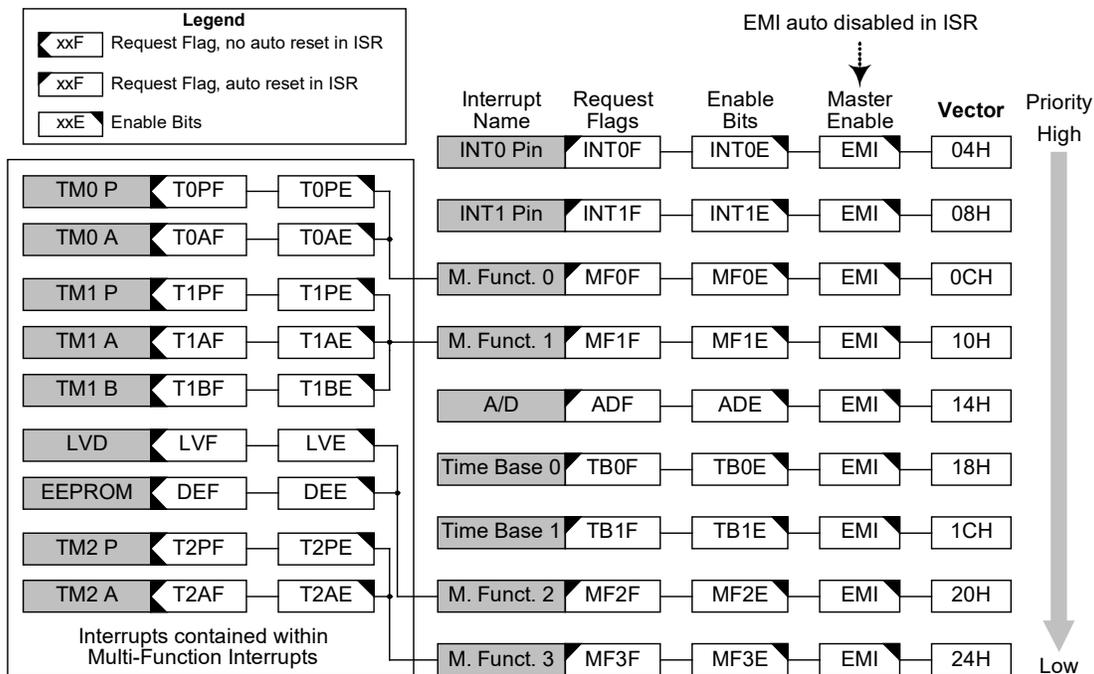
If an interrupt requires immediate servicing while the program is already in another interrupt service routine, the EMI bit should be set after entering the routine, to allow interrupt nesting. If the stack is full, the interrupt request will not be acknowledged, even if the related interrupt is enabled, 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 Scheme – HT66F24D

Interrupt Scheme – HT66F25D

EMI auto disabled in ISR

Interrupt Name	Request Flags	Enable Bits	Master Enable	Vector	Priority
					High
INT0 Pin	INT0F	INT0E	EMI	04H	
INT1 Pin	INT1F	INT1E	EMI	08H	
M. Funct. 0	MF0F	MF0E	EMI	0CH	
M. Funct. 1	MF1F	MF1E	EMI	10H	
A/D	ADF	ADE	EMI	14H	
Time Base 0	TB0F	TB0E	EMI	18H	
Time Base 1	TB1F	TB1E	EMI	1CH	
M. Funct. 2	MF2F	MF2E	EMI	20H	
M. Funct. 3	MF3F	MF3E	EMI	24H	Low

Legend

xxF	Request Flag, no auto reset in ISR
xxF	Request Flag, auto reset in ISR
xxE	Enable Bits

TM0 P	T0PF	T0PE
TM0 A	T0AF	T0AE
TM1 P	T1PF	T1PE
TM1 A	T1AF	T1AE
TM1 B	T1BF	T1BE
LVD	LVF	LVE
EEPROM	DEF	DEE
TM2 P	T2PF	T2PE
TM2 A	T2AF	T2AE

Interrupts contained within Multi-Function Interrupts

Interrupt Scheme – HT66F26D

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.

Multi-function Interrupt

Within these devices there are up to nine 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, LVD or Data EEPROM Interrupts.

A Multi-function interrupt request will take place when any of the Multi-function interrupt request flags, MF0F~MF2F 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, LVD or Data EEPROM Interrupts, 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 Interrupt

The function of the Time Base Interrupt is to provide regular time signal in the form of an internal interrupt. It is controlled by the overflow signal from the respective timer function. When this happens, the 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 the corresponding Time Base enable bit, 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	—	TB02	TB01	TB00
R/W	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	0	0	1	1	—	1	1	1

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

Bit 6   **TBCK**: Time Base clock f_{TB} selection
         0: f_{SUB}
         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   unimplemented, read as "0"

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: $32768/f_{TB}$
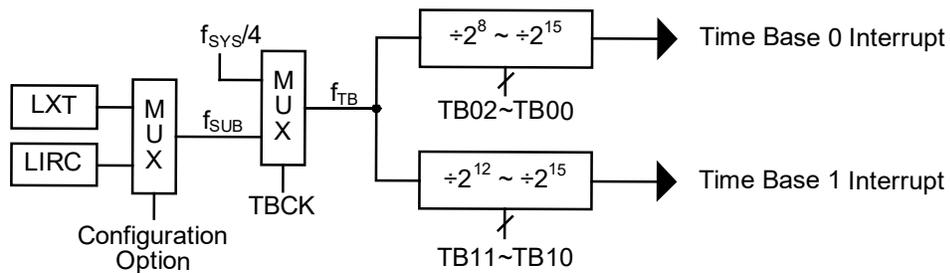
LVD Interrupt

The Low Voltage Detector Interrupt is contained within the Multi-function Interrupt. A 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 Interrupt

The Compact and Standard Type TMs have two interrupts each, while the Enhanced Type TM has three interrupts. All of the TM interrupts are contained within the Multi-function Interrupts in these devices. For each of the Compact and Standard Type TMs there are two interrupt request flags TnPF and TnAF and two enable bits TnPE and TnAE. For the Enhanced Type TM there are three interrupt request flags TnPF, TnAF and TnBF and three enable bits TnPE, TnAE and TnBE. A TM interrupt request will take place when any of the TM request flags are set, a situation which occurs when a TM comparator P, A or B match situation happens.

To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, respective TM Interrupt enable bit, and relevant Multi-function Interrupt enable bit, MFnE, must first be set. When the interrupt is enabled, the stack is not full and a TM comparator match situation occurs, a subroutine call to the relevant Multi-function Interrupt vector locations, will take place. When the TM interrupt is serviced, the EMI bit will be automatically cleared to disable other interrupts, however only the related MFnF flag will be automatically cleared. As the TM interrupt request flags will not be automatically cleared, they have to be cleared by the application program.

Interrupt Wake-up Function

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

Programming Considerations

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

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

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

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

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

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

Low Voltage Detector – LVD

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

LVD Register

The Low Voltage Detector function is controlled using a single register with the name LVDC. Three bits in this register, VLVD2~VLVD0, are used to select one of eight fixed voltages below which a low voltage condition will be 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　　　**LVDON**: 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: 4.0V

LVD Operation

The Low Voltage Detector function operates by comparing the power supply voltage, VDD, with a pre-specified voltage level stored in the LVDC register. This has a range of between 2.0V and 4.0V. When the power supply voltage, V_{DD}, falls below this pre-determined value, the LVDO bit will be set high indicating a low power supply voltage condition. 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
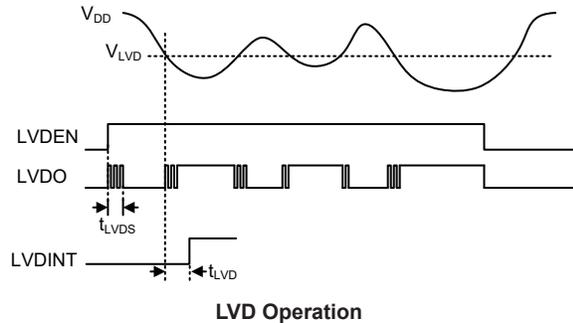
The Low Voltage Detector also has its own interrupt which is contained within one of the Multi-function interrupts, providing an alternative means of low voltage detection, in addition to polling the LVDO bit. The interrupt will only be generated after a delay of t_{LVD} after the LVDO bit has been set 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.

Configuration Options

Configuration options refer to certain options within the MCU that are programmed into the device during the programming process. During the development process, these options are selected using the HT-IDE software development tools. As these options are programmed into the device using the hardware programming tools, once they are selected they cannot be changed later 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: HXT, HIRC
2	High Speed Internal RC Frequency Selection - f_{HIRC}: 4MHz, 8MHz or 12MHz
3	f_{SUB} Oscillator Selection – f_{SUB}: LIRC
Watchdog Options	
4	Watchdog Timer: always enabled or controlled by software register

Application Circuit
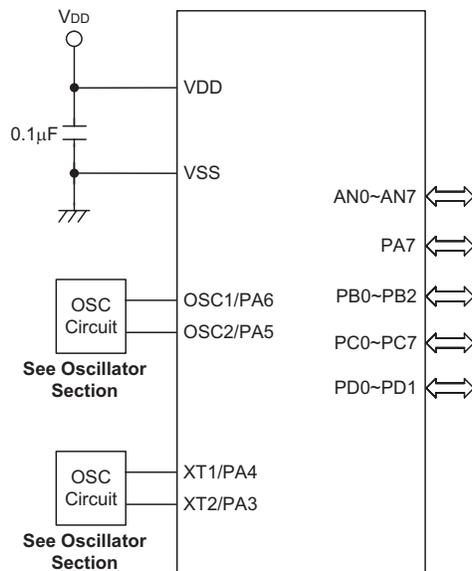
Instruction Set

Introduction

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

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

Instruction Timing

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

Moving and Transferring Data

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

Arithmetic Operations

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

Logical and Rotate Operation

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

Branches and Control Transfer

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

Bit Operations

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

Table Read Operations

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

Other Operations

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

Instruction Set Summary

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

Table Conventions

x: Bits immediate data

m: Data Memory address

A: Accumulator

i: 0~7 number of bits

addr: Program memory address

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

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

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

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

Instruction Definition

ADC A,[m]	Add Data Memory to ACC with Carry
Description	The contents of the specified Data Memory, Accumulator and the carry flag are added. The result is stored in the Accumulator.
Operation	ACC ← ACC + [m] + C
Affected flag(s)	OV, Z, AC, C

ADCM A,[m]	Add ACC to Data Memory with Carry
Description	The contents of the specified Data Memory, Accumulator and the carry flag are added. The result is stored in the specified Data Memory.
Operation	[m] ← ACC + [m] + C
Affected flag(s)	OV, Z, AC, C

ADD A,[m]	Add Data Memory to ACC
Description	The contents of the specified Data Memory and the Accumulator are added. The result is stored in the Accumulator.
Operation	ACC ← ACC + [m]
Affected flag(s)	OV, Z, AC, C

ADD A,x	Add immediate data to ACC
Description	The contents of the Accumulator and the specified immediate data are added. The result is stored in the Accumulator.
Operation	ACC ← ACC + x
Affected flag(s)	OV, Z, AC, C

ADDM A,[m]	Add ACC to Data Memory
Description	The contents of the specified Data Memory and the Accumulator are added. The result is stored in the specified Data Memory.
Operation	[m] ← ACC + [m]
Affected flag(s)	OV, Z, AC, C

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

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

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

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

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

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

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

CPL [m]	Complement Data Memory
Description	Each bit of the specified Data Memory is logically complemented (1′s complement). Bits which previously contained a 1 are changed to 0 and vice versa.
Operation	$[m] \leftarrow \overline{[m]}$
Affected flag(s)	Z

CPLA [m]	Complement Data Memory with result in ACC
Description	Each bit of the specified Data Memory is logically complemented (1′s complement). Bits which previously contained a 1 are changed to 0 and vice versa. The complemented result is stored in the Accumulator and the contents of the Data Memory remain unchanged.
Operation	$ACC \leftarrow \overline{[m]}$
Affected flag(s)	Z

DAA [m]	Decimal-Adjust ACC for addition with result in Data Memory
Description	Convert the contents of the Accumulator value to a BCD (Binary Coded Decimal) value resulting from the previous addition of two BCD variables. If the low nibble is greater than 9 or if AC flag is set, then a value of 6 will be added to the low nibble. Otherwise the low nibble remains unchanged. If the high nibble is greater than 9 or if the C flag is set, then a value of 6 will be added to the high nibble. Essentially, the decimal conversion is performed by adding 00H, 06H, 60H or 66H depending on the Accumulator and flag conditions. Only the C flag may be affected by this instruction which indicates that if the original BCD sum is greater than 100, it allows multiple precision decimal addition.
Operation	[m] ← 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 $[m] \leftarrow [m] - 1$

Affected flag(s) Z

DECA [m] Decrement Data Memory with result in ACC

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

Operation $ACC \leftarrow [m] - 1$

Affected flag(s) Z

HALT Enter power down mode

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

Operation $TO \leftarrow 0$
$PDF \leftarrow 1$

Affected flag(s) TO, PDF

INC [m] Increment Data Memory

Description Data in the specified Data Memory is incremented by 1.

Operation $[m] \leftarrow [m] + 1$

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 $ACC \leftarrow [m] + 1$

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 $\leftarrow$ 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 $ACC \leftarrow [m]$

Affected flag(s) None

MOV A,x Move immediate data to ACC

Description The immediate data specified is loaded into the Accumulator.

Operation $ACC \leftarrow x$

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 $[m] \leftarrow ACC$

Affected flag(s) None

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

SBC A,[m]	Subtract Data Memory from ACC with Carry
Description	The contents of the specified Data Memory and the complement of the carry flag are subtracted from the Accumulator. The result is stored in the Accumulator. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.
Operation	$ACC \leftarrow ACC - [m] - \overline{C}$
Affected flag(s)	OV, Z, AC, C

SBCM A,[m]	Subtract Data Memory from ACC with Carry and result in Data Memory
Description	The contents of the specified Data Memory and the complement of the carry flag are subtracted from the Accumulator. The result is stored in the Data Memory. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.
Operation	$[m] \leftarrow ACC - [m] - \overline{C}$
Affected flag(s)	OV, Z, AC, C

SDZ [m]	Skip if decrement Data Memory is 0
Description	The contents of the specified Data Memory are first decremented by 1. If the result is 0 the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.
Operation	$[m] \leftarrow [m] - 1$ Skip if [m]=0
Affected flag(s)	None

SDZA [m]	Skip if decrement Data Memory is zero with result in ACC
Description	The contents of the specified Data Memory are first decremented by 1. If the result is 0, the following instruction is skipped. The result is stored in the Accumulator but the specified Data Memory contents remain unchanged. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0, the program proceeds with the following instruction.
Operation	$ACC \leftarrow [m] - 1$ Skip if ACC=0
Affected flag(s)	None

SET [m]	Set Data Memory
Description	Each bit of the specified Data Memory is set to 1.
Operation	$[m] \leftarrow FFH$
Affected flag(s)	None

SET [m].i	Set bit of Data Memory
Description	Bit i of the specified Data Memory is set to 1.
Operation	$[m].i \leftarrow 1$
Affected flag(s)	None

SIZ [m]   Skip if increment Data Memory is 0

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

Operation $[m] \leftarrow [m] + 1$
                             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 $ACC \leftarrow [m] + 1$
                             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 $[m].i \neq 0$

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 $ACC \leftarrow ACC - [m]$

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 $[m] \leftarrow ACC - [m]$

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 $ACC \leftarrow ACC - x$

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 $[m].3\sim[m].0 \leftrightarrow [m].7\sim[m].4$

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   The contents of the specified Data Memory are read out and then written to the specified Data
                  Memory again. If the contents of the specified Data Memory is 0, the following instruction is
                  skipped. As this requires the insertion of a dummy instruction while the next instruction is
                  fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the
                  following instruction.

Operation   Skip if [m]=0

Affected flag(s) None

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

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

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

Affected flag(s) None

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

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

Operation   Skip if [m].i=0

Affected flag(s) None

TABRD [m]   Read table (specific page or current page) to TBLH and Data Memory

Description   The low byte of the program code addressed by the table pointer (TBHP and TBLP or only
                  TBLP if no TBHP) is moved to the specified Data Memory and the high byte moved to
                  TBLH.

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

Affected flag(s) None

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

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

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

Affected flag(s) None

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

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

Operation   ACC ← ACC ″XOR″ [m]

Affected flag(s) Z

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

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

Package Information

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

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

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

- Packing Meterials Information

- Carton information

16-pin NSOP (150mil) Outline Dimensions
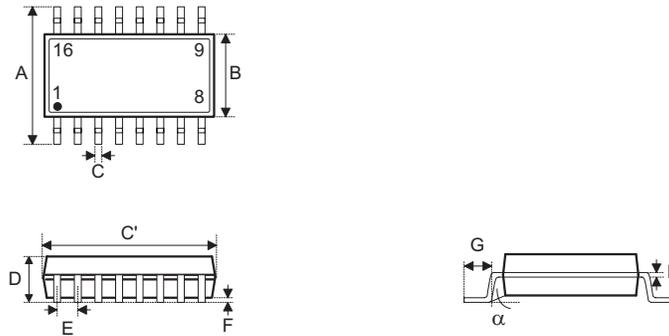
Symbol	Dimensions in inch		
	Min.	**Nom.**	**Max.**
A	—	0.236 BSC	—
B	—	0.154 BSC	—
C	0.012	—	0.020
C'	—	0.390 BSC	—
D	—	—	0.069
E	—	0.050 BSC	—
F	0.004	—	0.010
G	0.016	—	0.050
H	0.004	—	0.010
α	0°	—	8°

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°

20-pin SOP (300mil) Outline Dimensions
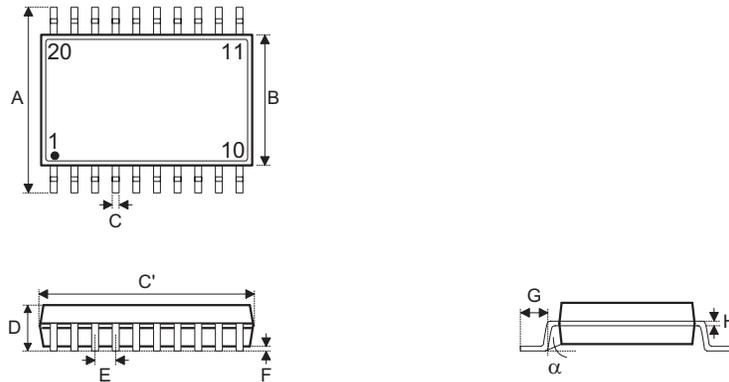
Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.406 BSC	—
B	—	0.295 BSC	—
C	0.012	—	0.020
C'	—	0.504 BSC	—
D	—	—	0.104
E	—	0.050 BSC	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

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°

24-pin SOP (300mil) Outline Dimensions
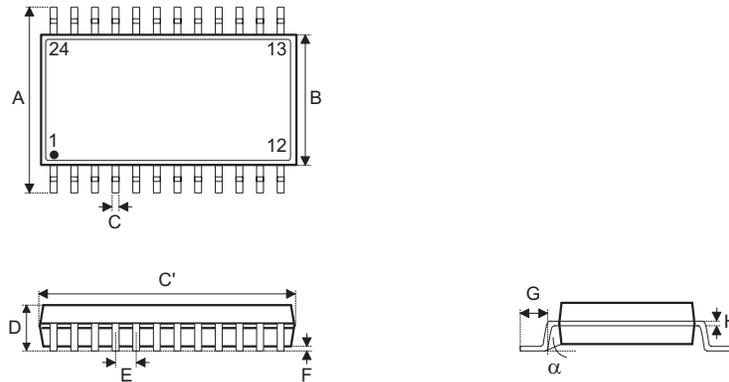
Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.406 BSC	—
B	—	0.295 BSC	—
C	0.012	—	0.020
C'	—	0.606 BSC	—
D	—	—	0.104
E	—	0.050 BSC	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

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°

28-pin SOP (300mil) Outline Dimensions
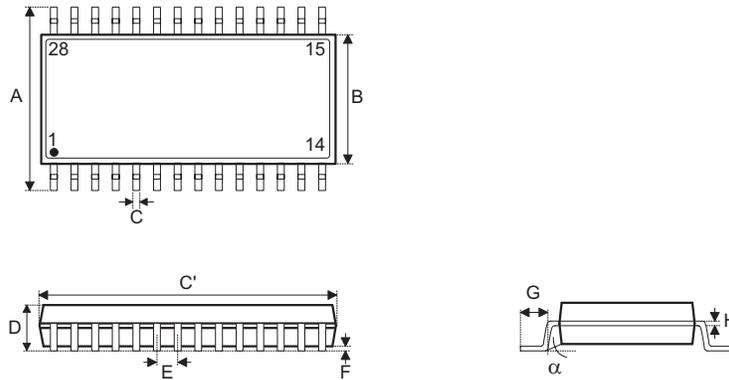
Symbol	Dimensions in inch		
	Min.	**Nom.**	**Max.**
A	—	0.406 BSC	—
B	—	0.295 BSC	—
C	0.012	—	0.020
C'	—	0.705 BSC	—
D	—	—	0.104
E	—	0.050 BSC	—
F	0.004	—	0.012
G	0.016	—	0.050
H	0.008	—	0.013
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	**Nom.**	**Max.**
A	—	10.30 BSC	—
B	—	7.5 BSC	—
C	0.31	—	0.51
C'	—	17.9 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°