210-960 MHz OOK/(G)FSK Transmitter SoC

CMT2187A
Sub-1G Transmitting Micro-controller
User Guide

1. This document describes the functions, operations, and usage of the CMT2187A. It is guidance for engineers who uses the CMT2187A.
2. This manual is limited by the length of the document, and the referred registers of the chip function modules are only listed. Please refer to the CMT2187A Register Detailed Manual for detailed register description. Users can understand the chip functions more efficiently by referring to this document.

MCU Feature

- CPU kernal
  - High performance single instruction period 1T-8051 kernal
  - Supports up to 26MHz (XOSC) or 24Mhz (HFOSC) operating frequencies with a maximum access efficiency of 20MIPS
  - Operating consumption
- Storage
  - 4-KB MTP program storage, support 10K erasing time
  - 512-Byte XRAM and 256-Byte IRAM
  - 512-Bit EEPROM，support 1,000,000 erasing time
- Power
  - Power on reset and low voltage detection
  - Embeded independent LDO provides power for CPU and digital citcuit
  - The embeded ultra-low power ULPLDO achieves the Retention function of CPU/RAM/SFR and some of the peripherals in STOP mode
- I/O
  - 11/9 multi-functional IO pins (SOP16 / SOP14)
  - Supports highly flexible peripheral function mapping
  - Support level change interrupt/wake up
- Clock source
  - Support up to 26Mhz XOSC (high speed frequency crystal oscillator)
  - Built-in high speed 24MHz HFOSC (±1% RC oscillator)
  - Built-in low power consumption 32kHz LFOSC (±1% RC oscillator)
- Debug onchip
  - CPU built-in 1-Wire debugger hardware circuit
  - Support Keil C51 for online program debugging
  - Supports 3 hardware breakpoints, single step debugging
- Peripheral
  - 1x UART
  - 1x SPI
  - 1x CDR（single wire RX input clock recovery）
  - 1x WDT（independent hardware）
  - 1x sleep timer（32KHz LFOSC）
  - 2x 16 bit simple timer
  - 2x 16 bit multi-Function timer（3 channel PWM/CCP）
  - 2x analog comparator
- Code Security
  - The burning serial ports and single-wire debugging interface have locking function

Sub-1G Transmitting Module Attributes

- Working frequency: 210 - 960MHz
- Modulated Mode: OOK / ASK
- Data rate: 0.5– 40kbps（OOK）
- Output power：+13dBm（Max.）
- Working current：24mA @+13dBm，433.92MHz CW
- Single-ended high efficiency Class E high frequency transmitting PA
- PA Ramping varies according to the data rate

Working Condition

- Temperature range is -40℃ - 85℃
- Working voltage range is 1.8V - 3.6V

Application

- Remote garage door control
- Remote door control system
- Consumer wireless remote control
- Smart home
- House security
- Source RFID tagging
- Wireless sensor network
- WM-Bus T1 mode

Package

- SOP14

SOP-14

8.65 x 6 x 1.75 mm

Introduction

The CMT2187A is a low-power SoC RF transmitter embedded with an enhanced 1T-8051 core，

1. which supports OOK/(G)FSK modulation wireless transmission function among 210 ~ 960 MHz。
2. The transmitter module not only provides high efficient single-ended PA, with adjustable output power from 0 to +13dBm, and only 24mA needed for +13dBm transmission;
3. But also provides 4-KB MTP program memory, 512-Byte XRAM, 256-Byte IRAM, and 512-bit EEPROM;
4. The built-in ultra-low power ULPLDO supports the chip to save CPU status, RAM data, and configuration register data in STOP mode
5. With 1-WIRE online simulation function, users can download the target debugging code directly to the on-chip MTP through the dedicated 1-WIRE debugger and Keil C51 software, which is very convenient.
6. It supports external 26MHz XO or built-in 24MHz HFOSC as the system main frequency and the built-in low power 32 kHz LFOSC can be used for low power timer wake-up;
7. It also supports single-wire input hardware clock recovery module, which is convinient for the kernel to collect the external data synchronously (such as RX receiving data).

Combined with CMOSTEK's NextGenRFTM series receivers, CMT2187A can be applied in a wide range of ultra-low power wireless network.

Product Information.

Part Number	Package	Dimension
CMT2187A-ESR14	SOP-14	8.65 mm x 6.00 mm x 1.75mm

Categories

1 System Architect

Embedded with a sub-1 GHz OOK / (G)FSK transmitter，CMT2187A is a high-performance 8051 wireless MCU. The user program is burned in the 4K Bytes MTP, which can be operated at clock frequency up to 26MHz. The chip integrates the below major modules:

- High-performance 8051 core with 1-Wire online debug circuit;
- Rich digital and analog peripheral resources.
- Sub-1G OOK / (G)FSK modulated transmitting module；

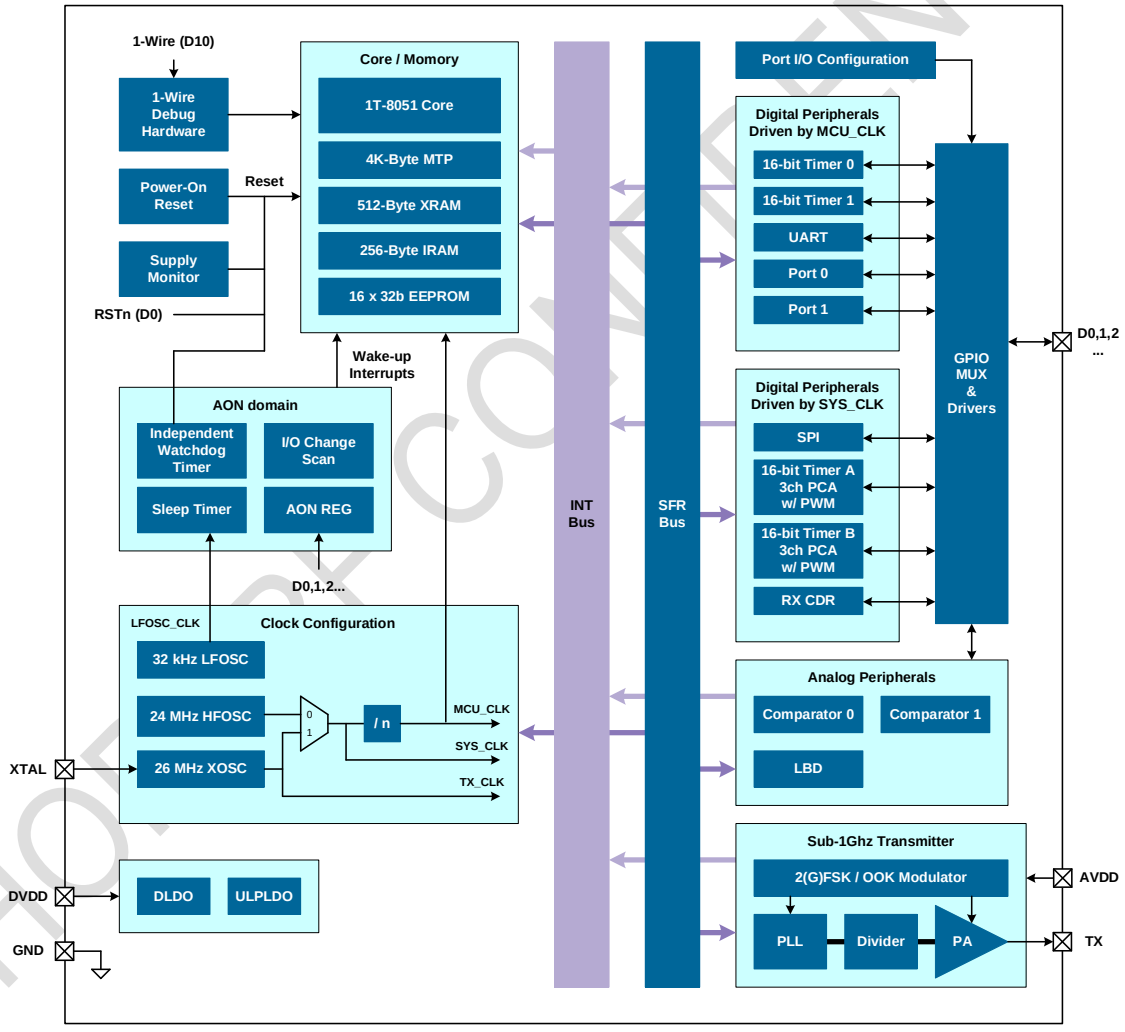

Chart 1- 1. System Block Diagram

2 System Operating Process and Working Mode

2.1 System Operating Process

The system operation process of CMT2187A is shown as follows:

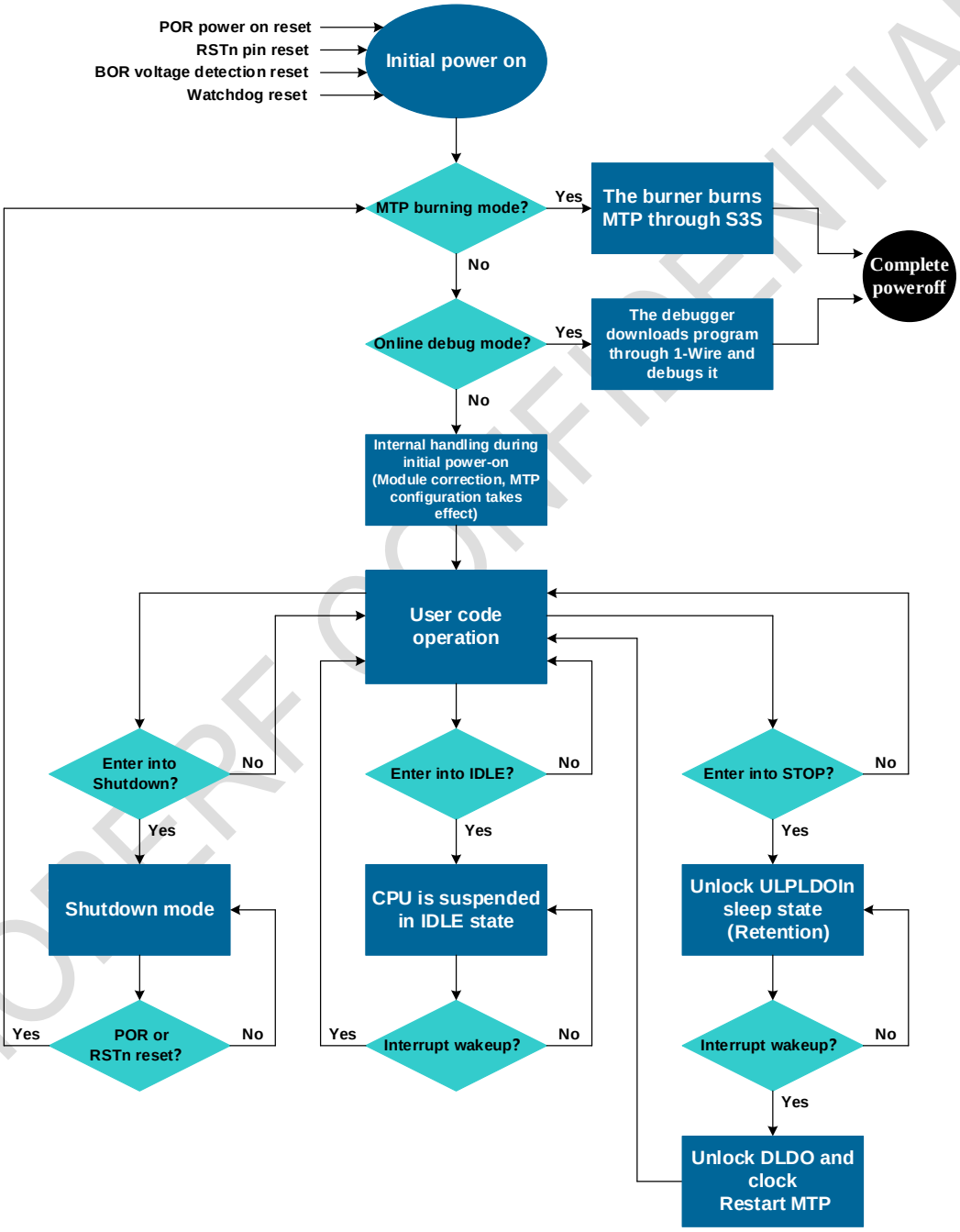

Chart 2- 1. System Operating Process Chart

As shown in the chart above, the initial chip power-on is triggered by enabling the DVDD pin and resetting the POR. When the RSTn pin and the BOR voltage detection are reset, as well as the watchdog reset takes effect, the chip will enter the power-on process, which is hereinafter referred to as "power-on". After power-on, a 6 ms time window will be opened, and if the burning command triggered by the S3S serial port is detected in the window, it will enter the burning mode, allowing the burner to burn the internal MTP; If a debug command triggered by the 1-Wire interface is detected in the window, it goes into debug mode, allowing users to debug user code through the Keil C51 software and debugger. After burning or debugging complete, the chip needs to be powered off before other operations to enabling again.

If the burn mode or debug mode is not triggered within 6 ms after power-on, the chip will continue the internal processing of the initial power-on, including power and clock correction, and configuration in the MTP Config area. The user code will then start to run from address 0x0000, during which the user can configure registers to put the chip into IDLE, or STOP mode. In SDN mode, it can only be wakeup by power-on reset or external pin reset. In IDLE mode, it can be wakeup by interrupts caused by I/O level changes or comparator output reversals. In STOP mode, user can wake up interrupts caused by I/O level change, sleep timer timeout, or comparator output, after which user can turn on the DLDO, clock, and MTP of the digital circuit power supply, so that the user code can make the chip to run under the pre-sleep state.

2.2 System Operating Mode

The chip has the following four working modes:

Table 2-1. Working Modes of CMT2187A

Working Mode	Description	Mode	Wakeup source
Normal	Normal Status	User program will automatically enter after burning and powered on	None
IDLE	● DLDO enabled ● System Clock（HFOS or XOSC）enabled ● CPU kernal suspended ● Peripherals work	Set the IDLE bit in the PCON register	I/O level change Comparator output
STOP (Retention)	● ULPLDO enabled ● System Clock（HFOS or XOSC）disabled ● CPU core，save all the storage, peripheral configurations: and state. ● LFOSC enabled, Always-On module and comparator work ● GPIO state remain unchanged	● Set the STOP bit in the PCON register ● Set the SLEEP bit in the AON_SFR_03 register	I/O level change Comparator output flipping Sleep timer timed out

Comparing from the power consumption in all four modes, Normal > IDLE > STOP. CMT2187A has two power

pins, the AVDD supply power to the internal RF circuit, and the DVDD supply power to the Always-On digital module and the analog module except RF. Most of the digital modules work under the built-in DLDO and can be switched to ULPLDO power supply at STOP to achieve low leakage Retention mode.

The Retention mode allows the chip to recover from its previous state immediately after a STOP wake-up and continue working without having to restart the program. In Retention mode, all RAM data is stored; MTP and EEPROM data can be stored without power.

Table 2-2. CMT2187A Stores Contents in STOP Mode

Storage	Saving data	Power Supply Mode
MTP	√	File Save
EEPROM	√	File Save
IRAM	√	ULPLDO
XRAM	√	ULPLDO

In Retention mode, both power-on reset (POR) and real-time voltage Monitor (Power Monitor) remain in working state. The following lists whether all functional modules save the SFR configuration and working status, whether they work, and their corresponding power supply mode.

Table 2-2. CMT2187A Stores Contents in STOP Mode

Number	Module name	Configuration saved	Save State	Work status	Power Supply Mode
1	Watch Dog Timer	√	√	√	DVDD
2	Sleep Timer	√	√	√	DVDD
3	Key Scan	√	√	√	DVDD
4	Comparator 0	√	√	√	DVDD
5	Comparator 1	√	√	√	DVDD
6	UID & CFG Register	√	√	×	DVDD
7	IO Configuration Status	√	√	×	DVDD
8	1T-8051 core	√	√	×	ULPLDO
9	Timer 0	√	√	×	ULPLDO
10	Timer 1	√	√	×	ULPLDO

Number	Module name	Configuration saved	Save State	Work status	Power Supply Mode
11	UART	√	√	×	ULPLDO
12	Port 0	√	√	×	ULPLDO
13	Port 1	√	√	×	ULPLDO
14	SPI	√	×	×	ULPLDO
15	Timer A	√	×	×	ULPLDO
16	Timer B	√	×	×	ULPLDO
17	CDR	√	×	×	ULPLDO
18	Sub-1G Transmitter	√	×	×	ULPLDO
19	LBD	√	×	×	Power off
20	1-Wire Debug	×	×	×	Power off

In the above table, numbered 1-7 modules exist in the Always-On (Always powered on) area, which is referred to as the AON area below. This area is directly powered by DVDD, and the leakage of the module is very small when it is not working. Among them, the watchdog, sleep timer, key scan, and two comparators can all enable or disable in STOP mode according to user configuration; However, IO configuration and status do not change in STOP mode in UID & CFG register.

CPU cores and peripherals of modules from 8 to 13 are driven by MCU_CLK. All configurations and current status of this part of circuit are saved in STOP mode with no other operation.

Modules of 14 - 18 are peripherals driven by SYS_CLK, of which all the circuit configurations are saved under STOP mode, while not saved the current working sate. User does not need to reconfigure the modules after wakeup, and will start working again, which is somehow like module reset automatically.

Modules 19 -20 are the power off modules under STOP mode, which will not save any content.

2.3 Protect Mechanism

To ensure the chip security after burning, a security mechanism is designed inside the chip. There is a READ_LOCK protection bit in the Config area of MTP. When this protection bit is burned, the user code and configuration of MTP cannot be read by S3S. If the user needs to unlock, the MTP can be re-burned through the burner, during which the original user code and configuration will be erased, including the READ_LOCK bit.

3 Debugging and Burning Interfaces

3.1 1-WIRE Online Debugging and Burning Interfaces

CMT2187A can be connected to the PC through the CMT2187A emulator of CMOSTEK for online debugging and MTP burning. The following shows figure of tool connection and interface connection between the debugger and CMT2187A. It should be noted that the 1-Wire debugging interface needs to occupy pin D10, and it is recommended that the user leave this pin empty during the debugging phase. MTP burning is implemented through the three-wire S3S interface.

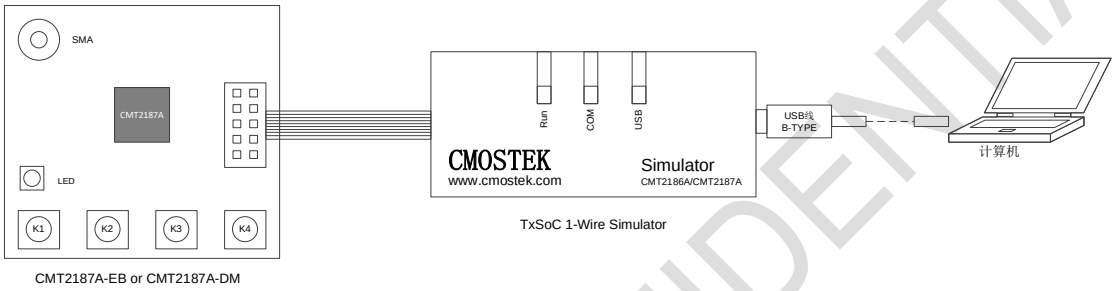

Chart 3-1. 1-Wire Tool Connecting Diagram

1-WIRE online debugging interface, general functions can be achieved in Keil C51 platform:

- Full speed operating, stop, single step execution, multi-step execution and other debugging modes;
- Supporting for software breakpoints (arbitrarily);
- Supports 3 hardware breakpoints
- Read and write R0~R7, part of the system status register, memory and other internal storage;
- *Reset key（RST symbol）disabled，it can only ahieved by using the exit:→ to reconnect.*

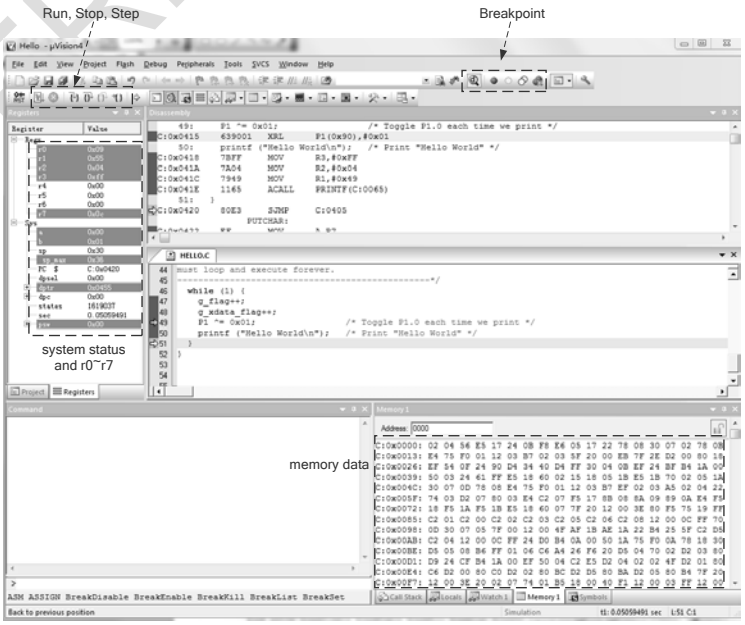

Chart 3-2. 1-Wire debugging screenshot on Keil C51

3.2 S3S Bus Burning Interface

The S3S bus is used for burning MTP, which is limited to burning and production tools, and is generally not open to users. If it was required to know the specific timing and communication protocol of S3S bus, please contact sales or agents of HOPERF.

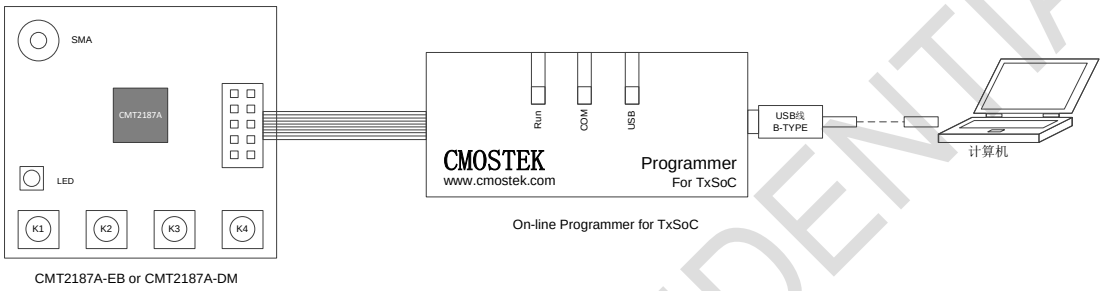

Chart 3-3. Online burning tool connection diagram

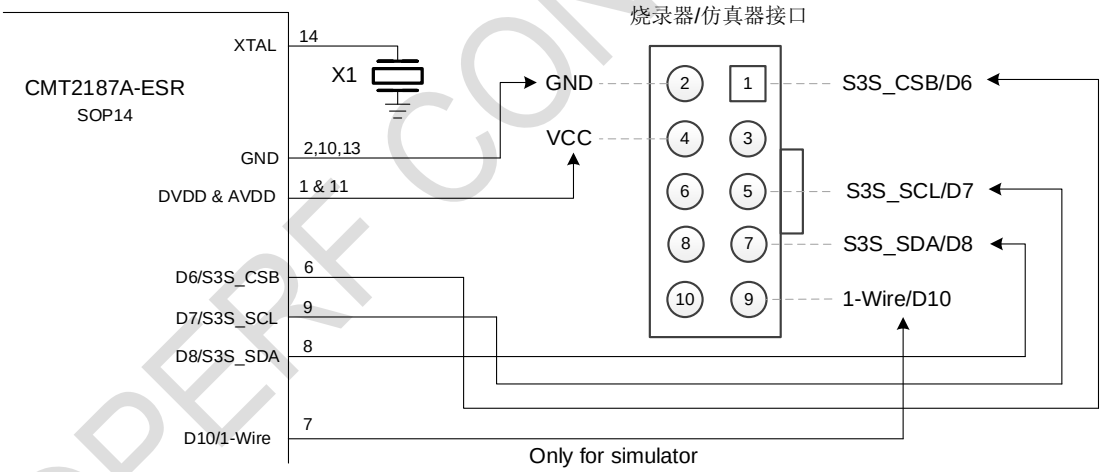

Chart 3-4. Burning/simulation Interface Connection Diagram

4 T8051XC3 Micro Controller

4.1 Processor Architecture

CMT2187A adopts T8051XC3 as the core controller of the system, including an enhanced 1T-8051 kernal, single period operating instruction, which is compatible to the MCS-51 command serial. The structure is shown as Chart 4-1.

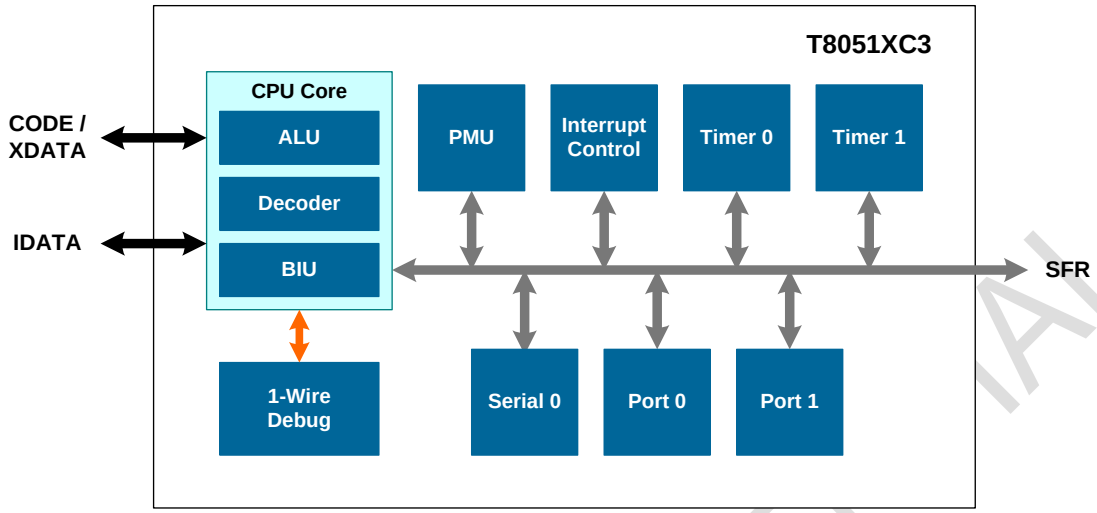

Chart 4- 1. T8051XC3 System Block Diagram

As shown in the figure, T8051XC3 includes the following parts:

- The CPU core is composed of BIU bus interface unit, Decoder instruction decoding unit, and ALU arithmetic logic unit
- The power management unit supports IDLE and STOP modes
- Interrupt control unit supports up to 8 external interrupts with level 2 interrupt priority control
- Two timers, Timer 0 and Timer 1
- One serial port, Serial Port 0, can implement UART mode
- An 11-bit parallel port, Port0 and Port1[1], is limited by the numbers of I/Os, and Port1 is only available in low 0-3 bits
- Single-wire (1-WIRE) online debugging module, support Keil C51 platform for software programming development and debugging

Note: [1] Port0 and Port1 come with the T8051XC3 kernel and are not directly equivalent to the GPIO of the chip. Comparing to the CPU kernal, the GPIO belongs to peripheral and Port0 and Port1 can be mapping to GPIOs.

T8051XC3 adopts an 8-bit SFR bus to connect the above mention peripherals. CMT2187A supports more peripherals, which are connected to the kernel via the SFR bus. In addition, the kernel uses a separate IDATA bus to connect to internal storage IRAM, and a shared CODE/XDATA bus to connect to MTP and XRAM respectively.

4.2 Instruction sets:

The 8051 instruction set consists of 111 instructions, each consisting of 1,2 or 3 bytes. Instruction execution is calculated in a single clock cycle. See Appendix A for more information of all instructions and their execution cycles.

4.3 8051 Core Initial Register

The core 8051 initial associated register group is shown in the following table. For the specific content and meaning of each register, please refer to the CMT2187A Register Detailed Manual.

Table 4-1. Initial Registers of 8051 Core

Name	SFR Page	Address	Default Values	Funtion
P0	0	0x80	0x00	Port0 register, support bit access, corresponding to eight kernel ports of P0.0-p0.7
SP	0	0x81	0x00	Stack Pointer Register
DPL	0	0x82	0x00	Data pointer (DPTR) register, low 8 bits
DPH	0	0x83	0x00	Data pointer (DPTR) register, high 8 bits
PCON	0	0x87	0x00	Power Control Register
TCON	0	0x88	0x00	Timer0 and Timer1 Control Registers
TMOD	0	0x89	0x00	Timer0 and Timer1 Working Mode Registers
TL0	0	0x8A	0x00	Timer0 register low 8 bit
TL1	0	0x8B	0x00	Timer1 register low 8 bit
TH0	0	0x8C	0x00	Timer 0 register high 8 bit
TH1	0	0x8D	0x00	Timer1 register high 8 bit
P1	0	0x90	0x00	Port1 register, support bit access, corresponding to eight kernel ports from P1.0 - P0.7. Due to the I/O numbers, only P1.0 – P1.3 can be accessed.
SCON0	0	0x98	0x00	Serial port control register
SBUF0	0	0x99	0x00	Serial port data cache register
IEN0	0	0xA8	0x00	Interrupt enabling register 0
IPL0	0	0xB8	0x00	Interrupt priority register 0
PSW	0	0xD0	0x00	Program status/ marking register
ACC	0	0xE0	0x00	Accumulated Register
IEN1	0	0xE6	0x00	Interrupt enabling register 1
B	0	0xF0	0x00	B Register
IRCON1	0	0xF1	0x00	Peripheral interrupt request flag register
IPL1	0	0xF6	0x00	Interrupt priority register 1

5 Memory Structure

5.1 Introduction

The CMT216xA on-chip storage architecture is shown in Chart 5-1.

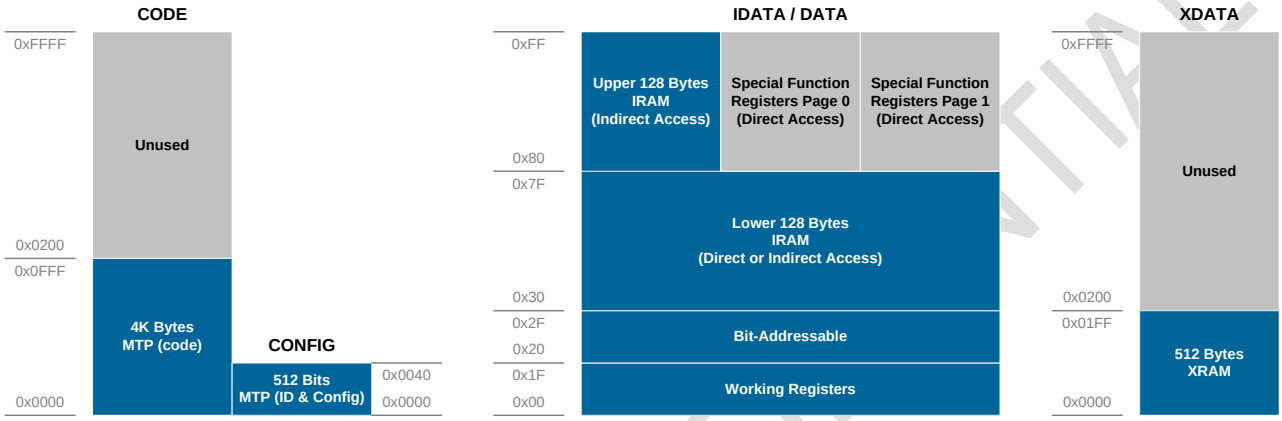

Chart 5-1. CMT2187A Storage and Logical Address

CMT2187A storage area consists of 3 spaces.

- **Program Code Space**

  The space where 8051 kernel code is stored and loaded to run, the carrier is 4K Bytes MTP that can be erased multiple times. MTP also supports configuration space of 512 bits for storing user IDs and some specific configurations of chip features. The contents of the code and configuration space are burned through the burner. The entire 4KB code space is available to the user, and the code starts executing at 0x0000.

- **Internal Data Space**

  The 8051 kernel contains 256 bytes of internal data space for fast access by the MCU. The internal DATA space can be divided into DATA, IDATA and SFR according to the access method, which is corresponding to the key words in the Keil C51 compiler, and the carrier is 256 Bytes of IRAM and SFR registers respectively. SFR is divided into two pages, which can be selected with the SFR_PAGE_SEL bit.

- **External Data Space**

  Storage of 8051 external data, XDATA is stored in 512 bytes of XRAM. Users can also store specific data that needs to be saved at power off in the 512-bit EEPROM, which can be accessed via SFR. In addition, there are some AON_REGS in the AON region, these registers are mainly used to configure

and control the modules and I/O of the AON region. Users can access them indirectly through SFR.

Table 5-1. Internal memory description;

Storage space	Storage	Logic address	Capabilities	Description
Program Code Space	MTP	0x0000 - 0x0FFF	4K bytes	User program run space, Keil C51 needs to use the keyword code to define variables.
Internal Data Space	IRAM	0x00 - 0x7F	128 bytes	IDATA low level bits, it can be both directly and indirectly accessed. In addition, a 16-byte addressable space is provided in the address range 0x20-0x2F. Keil C51 can be defined with the keyword data or idata, and variables accessed by bit can be defined by sbit.
		0x80 - 0xFF	128 bytes	IDATA high level bits supports indirect access. Keil C51 has to defined by keywords idata.
	SFR	0x80 - 0xFF	145 bytes	8051 can directly access the special feature register among the address range in internal RAM, including 2 pages of Page 0 and Page 1, which compare through SFR_PAGE_SEL bit of SFR.
External Data Space	XRAM	0x0000 - 0x01FF	512 bytes	Keil C51 needs to use the keyword code to define variables.
	EEPROM	0x00 - 0x1F	512 bits	16 Bit x 32 multi-programming memory. The kernel is accessed indirectly through SFR, or through the open source API program alignment to increase its usage.
	AON REG	0x00 - 0x1F	32 bytes	Registers located in the AON region are indirectly accessed by the kernel via SFR.

Notes:

[1]. After MTP is programmed, the burnt data (user program) will not be lost regardless of whether the system is powered or not, or which mode the system is operating in.

[2]. After the EEPROM is rewritten (requires the power supply being stable during the rewriting process), the rewritten data will not be lost regardless of whether the system is powered or not, or which mode the system is operating in.

[3] AON REG is in the regin of AON，content will not lost so long as the DVDD keeps powering.

[4] Content of IRAM，XRAM and some SFR can save under STOP mode.

5.2 Special Feature Registers（SFR）

8051 kernel can access SFR directly for it is the internal memory space. CMT2187A serial products are rich in features and related configured SFR, so we do access distribution by page, that is Page 0 and 1. Page 0 contains most of the peripheral configuration and control, and Page 1 contains registers of the EEPROM and the PA power configuration. Therefore, it is necessary to confirm that the Page point is correct when you

directly access the corresponding SFR, otherwise it is easy to cause configuration errors.

The SFR is powered by the ULPLDO in STOP mode, ensuring that most configurations of the peripherals can be saved in low leakage.

5.3 Always-on Domain Register (AON REG)

The always on (AON) domain system is powered directly by DVDD and it contains the watchdog, sleep timer, I/O change detection, and 32-byte register AON REG. Users can access AON REG indirectly through the AON_ADDR, AON_WDATA, and AON_RDATA registers in SFR. The objects controlled and configured by these registers include: the three peripherals in the AON domain above, the two analog comparators, and all I/Os. At the same time, there are 8 bytes in the AON REG, and when the chip is powered on for the first time, the system will automatically copy the 64-bit user ID in the MTP to the 8-byte register, which is convenient for users. Users are also free to use the 8-byte register for other purposes.

5.4 Memory Running Access Mode

The Memory Running Access Mode is shown in Table 5-2.

Table 5-2. Memory Running Access Mode

Memory Type	Access Method	Example
CODE	Constant definition in the program, using keyword "code"	uint8_t code array[3] = {0x12, 0x34, 0x56 };
XDATA	Variable definition in the program, using keyword “xdata”	uint8_t xdata tx_buf[64];
IDATA	Variable definition in the program, using keyword "idata"	uint8_t xdata tx_buf[3];
SFR	Direct Access Address[2]	IEN0= 0x00;
AON REG	Access through SFR[1]	None
EEPROM	Access via SFR, or API functions[2]	None
Notes: [1] The official will show how to access these registers. [2] The official will show how to access these registers and the API source code. The API can be used to increase the number of EEPROM erasing times.		

6 Reset Structure

The CMT2187A has four reset systems, including:

- **Power on reset（ POR）**

  The POR is enabled only once when the DVDD is powered on.

- **Voltage detection reset (BOR)**

  BOR is generated when abnormal fluctuations occur in DVDD to avoid chip confusion.

- **Pin Reset（ RSTn ）**

  The reset pin RSTn multiplexes D0 pin and it is enabled by default. Users can disable it after power-on if there is no need to use this function.

- **Watchdog Reset（WDT_RSTn）**

  Watchdog reset is a reset that prevents the program from running away or the system from crashing. When the user program is running normally, it is necessary to periodically "feed the dog" to avoid reset caused by the timeout of the watchdog timer.

These four resets all have the same effect, that is, after the reset is triggered, the chip will be re-powered for the first time.

7 Clock Structure

7.1 Clock Source

The CMT2187A has three master clock sources, namely the 26 MHz high speed crystal oscillator XOSC, the 24 MHz internal high speed RC oscillator HFOSC, and 32 kHz internal low speed RC oscillator LFOSC. A refined clock gating mechanism is embeded inside the chip so that users can save as much power as possible.

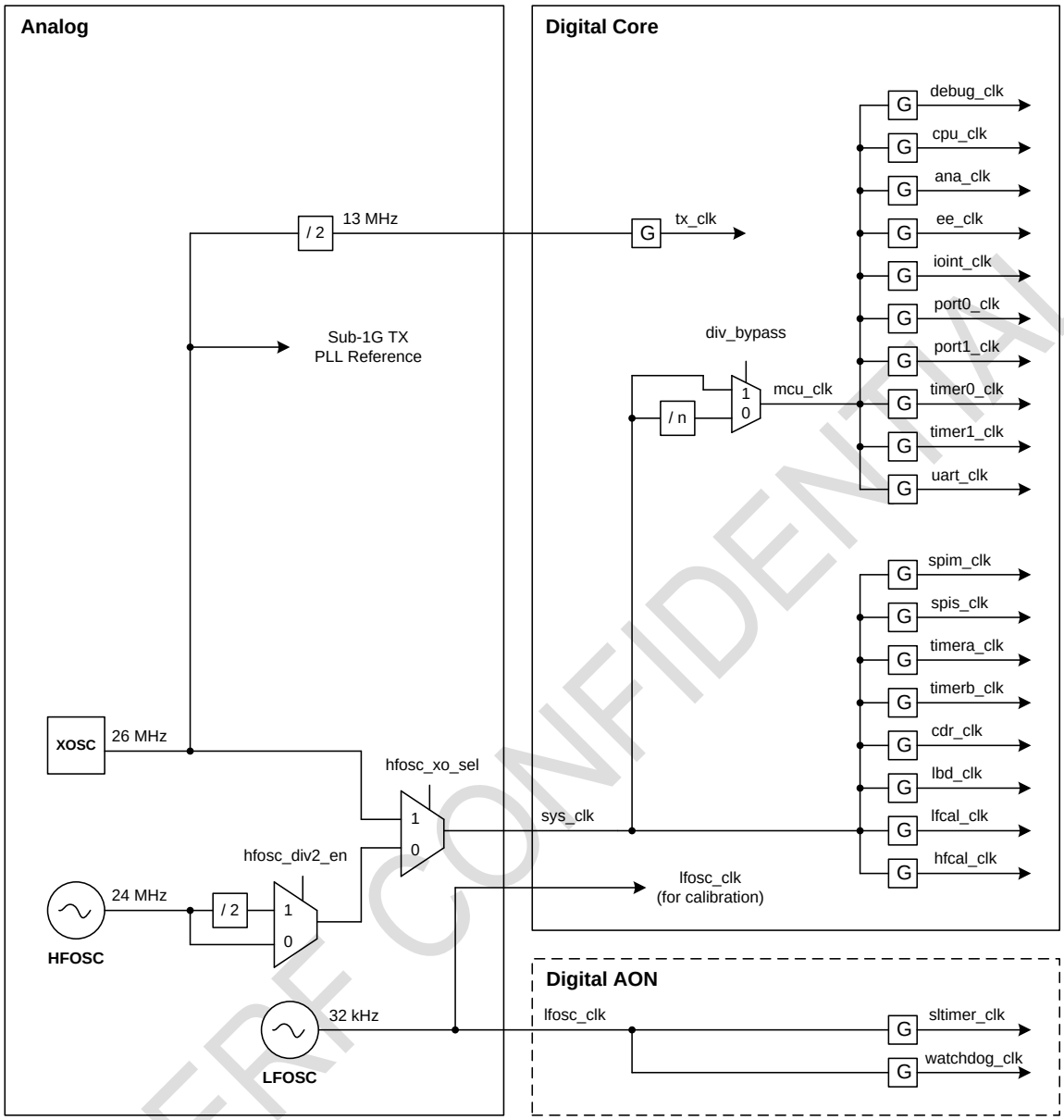

Chart 7-1. System Block Diagram

As shown in the figure above, the XOSC serves as the reference clock for the Sub-1G wireless transmitter PLL and is used to drive the digital transmission control and adjustment module after frequency division. The system master clock (SYS_CLK) is provided from HFOSC by default, and HFOSC can be calibrated to ±1% accuracy. If user wants to improve the accuracy of the master clock, the relevant configuration can be configured when burning the MTP, so that the chip automatically switches the master clock to XOSC after powering on, and the accuracy can be improved to the accuracy of the crystal oscillator itself, such as ±10ppm, while increasing some power consumption. LFOSC provides clocks for sleep timers and watch dog specifically that can achieve ±1% accuracy after calibration.

7.2 Clock Calibration

When the chip is shipped, HFOSC and LFOSC will be calibrated and the results will be burned to the MTP. The hardware correction module can also be invoked by manipulating the SFR register to correct the two clocks while using.

Calibration of HFOSC requires the use of XOSC as the reference clock. If there is no connection among the chip and external crystal oscillator while application (such as, non-wireles transmitting application), the XOSC cannot adapt normally, and HFOSC cannot be calibrated. Before correcting HFOSC, it is necessary to ensure that HFOSC serves as the clock source of SYS_CLK, set the HFOSC_DIV2_EN ratio in SFR to 1, and use the HFOSC clock as SYS_CLK after half frequency. This is to avoid excessive HFOSC frequency increase during the correction process, which will cause the system to malfunction.

Calibration of LFOSC requires SYS_CLK as the reference clock. If the user selects HFOSC as the clock source for SYS_CLK through MTP burning, it is recommended to correct HFOSC first and then LFOSC, for which the accuracy of HFOSC determines the accuracy of LFOSC.

The specific operation of correction can refer to the official open source routine code.

7.3 Clock Frequency Division

User can operate a frequency divider to divide SYS_CLK to generate MCU_CLK. The 8-bit frequency division coefficient of the divider can be configured from 1 to 255 except for 0. Therefore, the maximum operating frequency of MCU_CLK is 24 MHz (HFOSC) or 26 MHz (XOSC), and the minimum operating frequency is 94 kHz (HFOSC) or 102 kHz (XOSC).

SYS_CLK does not perform any frequency division during normal operation except for the situation described above at calibration (which will also affect MCU_CLK). Peripherals driven by SYS_CLK, the SPIM, SPIS, TIMERA, TIMERB, and CDR have their own operating frequency of SFR register configuration except for LBD, which uses a fixed clock frequency. Therefore, there is no need to operate frequency division of the driver clock.

LFOSC_CLK drives the sleep timer and the watchdog timer directly without any frequency division processing.

7.4 Clock Gate Control

In order to give fully play to the characteristics of low power consumption of the chip, the chip provides a unique clock gate for each module, which are not only control the clock module itself, but also control the clock of the

corresponding SFR register. Clock gating is turned on by default. It is recommended that the user turn off the gated clock of the module that does not need to work immediately at the beginning of the program after configuring all SFR, and turn on the clock only when the module needs to be configured, controlled, and used.

The following shows the corresponding control module for each clock gate, as well as the detailed SFR register:

Chart 7-1. Clock Gating of the Corresponding Modules and Registers

Clock Gate Control	Module	SFR Page	SFR Address	SFR Name
TX_CLK_EN	OOK / (G)FSK Transmitting controller and modulator	0	0xDD	ANA_CTL_0
		0	0xDE	ANA_CTL_1
		0	0xDF	ANA_CTL_2
		0	0xE1	ANA_CTL_3
		0	0xE8	PLLN
		0	0xE9	PLLK_H
		0	0xEA	PLLK_M
		0	0xEB	PLLK_L
		0	0xEC	TX_DR_0
		0	0xED	TX_DR_1
		0	0xEE	TX_DR_2
		0	0xEF	TX_SYM_BYTE
		0	0xF2	TX_SYM_CTL
		0	0xF3	TX_PKT_CTL
		0	0xF8	RAMP_STEP_H
		0	0xF9	RAMP_STEP_L
		0	0xFA	PA_IDAC_CODE
		0	0xFB	LBD_CTL_0[1]
		0	0xFC	LBD_CTL_1[1]
CPU_CLK_EN	CPU kernal	None	None	Switch automatically according to the working mode
DEBUG_CLK_EN	1-Wire debugger	None	None	Switch automatically based on whether to enter the debugging mode
EE_CLK_EN	EEPROM controller	1	0x2A	EE_CTL
		1	0x2B	EE_ADDR
		1	0x2C	EE_WDATA_H
		1	0x2D	EE_WDATA_L
		1	0x30	EE_RDATA_H
		1	0x31	EE_RDATA_L
		1	0x32	EE_STA

Clock Gate Control	Module	SFR Page	SFR Address	SFR Name
		1	0x33	EE_MANU
IOINT_CLK_EN	IO and Interrupt Controller	0	0x92	INTCTL_0
		0	0x93	INTCTL_1
		0	0x94	INTCTL_2
		0	0x95	INTCTL_3
		0	0xA1	GPIO_INA_SEL
		0	0xA2	GPIO_INB_SEL
		0	0xA3	GPIO_INC_SEL
		0	0xA4	GPIO_IND_SEL
		0	0xA5	GPIO_INE_SEL
		0	0xA6	GPIO_INF_SEL
		0	0xA9	GPIO_ING_SEL
		0	0xAA	GPIO_OUTA_SEL
		0	0xAB	GPIO_OUTB_SEL
		0	0xAC	GPIO_OUTC_SEL
		0	0xAD	GPIO_OUTD_SEL
		0	0xB0	GPIO_OUTE_SEL
		0	0xB1	GPIO_OUTF_SEL
PORT0_CLK_EN	Port 0	0	0x80	P0（8051 Initial Register）
PORT1_CLK_EN	Port 1	0	0x90	P1（8051 Initial Register）
TIMER0_CLK_EN	Timer 0	0	0x8A	TL0（8051 Initial Register）
		0	0x8C	TH0（8051 Initial Register）
TIMER1_CLK_EN	Timer 1	0	0x8B	TL1（8051 Initial Register）
		0	0x8D	TH1（8051 Initial Register）
UART_CLK_EN	UART 0	0	0x98	SCON0（8051 Initial Register）
		0	0x99	SBUF0（8051 Initial Register）
ANA_CLK_EN	Analog circuit controller	0	0xE2	ANA_CTL_4
		0	0xE3	ANA_CTL_5
		0	0xE4	ANA_CTL_6
		0	0xE5	ANA_CTL_7
		0	0xE7	ANA_CTL_8
SPIM_CLK_EN	SPI master machine	0	0x96	SPI_CTL_0
		0	0x97	SPI_CTL_1
SPIS_CLK_EN	SPI slave machine	0	0x96	SPI_CTL_0[2]
		0	0x97	SPI_CTL_1[2]
TIMERA_CLK_EN	Timer A	0	0xB7	TACLK_DIV_H
		0	0xB9	TACLK_DIV_L
		0	0xBA	TAC_H
		0	0xBB	TAC_L
		0	0xBC	TACCR0_H

Clock Gate Control	Module	SFR Page	SFR Address	SFR Name
		0	0xBD	TACCR0_L
		0	0xBE	TACCTL0_H
		0	0xBF	TACCTL0_L
		0	0xC0	TACCR1_H
		0	0xC1	TACCR1_L
		0	0xC2	TACCTL1_H
		0	0xC3	TACCTL1_L
		0	0xC4	TACCR2_H
		0	0xC5	TACCR2_L
		0	0xC6	TACCTL2_H
		0	0xC7	TACCTL2_L
		0	0xC8	TACNT_H
		0	0xC9	TACNT_L
TIMERB_CLK_EN	Timer B	0	0xCA	TBCLK_DIV_H
		0	0xCB	TBCLK_DIV_L
		0	0xCC	TBC_H
		0	0xCD	TBC_L
		0	0xCE	TBCCR0_H
		0	0xCF	TBCCR0_L
		0	0xD1	TBCCTL0_H
		0	0xD2	TBCCTL0_L
		0	0xD3	TBCCR1_H
		0	0xD4	TBCCR1_L
		0	0xD5	TBCCTL1_H
		0	0xD6	TBCCTL1_L
		0	0xD7	TBCCR2_H
		0	0xD8	TBCCR2_L
		0	0xD9	TBCCTL2_H
		0	0xDA	TBCCTL2_L
		0	0xDB	TBCNT_H
		0	0xDC	TBCNT_L
CDR_CLK_EN	Clock restorer	0	0x9D	CDR_DR_0
		0	0x9E	CDR_DR_1
		0	0x9F	CDR_DR_2
LBD_CLK_EN	Low voltage detector	None	None	Only control the LBD module[1]
LFOSC_CLK_EN	LFOSC calibration	None	None	Only control the LFOSC module[3]
HFOSC_CLK_EN	LFOSC calibration	None	None	Only control the HFOSC module[3]
SLTMR_CLK_EN	Sleep timer	None	None	Switches automatically through the AON_REG switch module[4]
WDG_CLK_EN	Watchdog Timer	None	None	Switches automatically through the

Clock Gate Control	Module	SFR Page	SFR Address	SFR Name
				AON_REG switch module[4]

Notes:

[1] The configuration of the LBD register affects part of the TX circuit function, so the gated clock is drivenby TX_CLK_EN, and the LBD_CLK_EN gated clock only drives the LBD module itself. It is necessary to detect the battery voltage before the transmission to compensate the transmission power in operation, so it is recommended that user turns on TX_CLK_EN and LBD_CLK_EN at the same time before transmission. If user needs to use the LBD module alone when not transmitting, enabling TX_CLK_EN to use the LBD_CTL_0 and LBD_CTL_1 registers, and opening LBD_CLK_EN to make the LBD module work.

[2] When SPIM_CLK_EN = 1 or SPIS_CLK_EN = 1, the clocks of both the SPI_CTL_0 and SPI_CTL_1 registers are turned on.

[3] When using the LFOSC or HFOSC correction module, it is control by the ANA_CTL_8 register, so ANA_CLK_EN is set to 1.

[4] All AON_REG does not require clock gating, and the clocks are only turned on when the CPU accesses them.

7.5 Related Register

Table 7-2. Related Register of System Clock

Name	SFR page	address	default values	Function
CLK_GATE_0	0	0x84	0x7F	Port0 register, support bit access, corresponding to eight kernel ports of P0.0-p0.7
CLK_GATE_1	0	0x85	0xFF	Stack Pointer Register
CLK_GATE_2	0	0x86	0x7F	Data pointer (DPTR) register, low 8 bits
MCU_CLK_DIV	0	0xFD	0x01	MCU_CLK frequency divider coefficient

8 Interrupts and Wakeup

8.1 Introduction

The interrupt control of CMT2187A has two main functions:

- The first: interrupt the current running process, and prioritize the interrupt service process;
- The second: wake up the system from low power mode;

The first function play the same role as the traditional microcontrollers, which is to response to interrupt handling during program operation, supported by all interrupt sources. The second function is to meet the needs of low-power applications, which is woke up through interruptions after the system enters all the low-power modes. In this case, only a limited number of interrupt sources can support the wakeup function. Here we call the interrupt source that can support wake up "wake up source" in order to better understand its working mechanism. Wake source and low power mode are related to each other, here are detail information of three low power modes of CMT2187A:

- **IDLE Mode**

  In IDLE mode, the 8051 kernel and the clock of the CPU_CLK will stop working, while the MCU_CLK will not stop. Thus, the initial and the system peripherals still work normally. This is the reason that the IDLE mode can be enabled by the interruption of these two peripherals.

- **STOP Mode**

  In STOP mode, the chip enters into sleep state and the MCU power supply switches from DLDO to ULPLDO to save the current working state with low power consumption. At this time, all clocks except LFOSC stop and can only rely on interrupts in the AON area to wake up, including I/O level changes, sleep timer timeout, and analog comparator output flip. If the user does not use sleep wakeup, the LFOSC and sleep timer can also not be turned on before entering STOP mode, which can further save power.

8.2 Wakeup Source

As mentioned in the previous section, only system peripherals in the normally open domain can support wakeup system in STOP mode, which we call the wakeup source. The wakeup source of CMT2187A mainly comes from the following three functional modules:

- **I/O Change Scan Module**

  The D0-D11 of CMT2187A can support this function. Users need to configure the GPIO that needs to detect wake-up before entering the STOP mode.

- **Sleep Timer Module**

  Low power sleep timer wakes up the STOP mode.

- **Analog Comparator Modular**

  The analog comparator is used to compare two input signals. When the comparison results change, it will trigger the interrupt wakeup system. User needs to configure the working mode of the comparator before entering the STOP mode.

8.3 Interrupt Source and Interrupt Control

The wakeup sources for CMT2187A have been described in the previous section. Since they support a pure wakeup system, it can also be understood as the wakeup source system. The CMT2187A interrupt source that will be introduced in this section is mainly associated with the 8051 operating, that is, the specific processing of the interrupt response based on the code operation.

The internal 8051 of CMT2187A supports 11 interrupt sources, namely:

- One Timer 0 interrupt；
- One Timer 1 interrupt；
- One Serial 0 (i.e. UART) interrupt;
- Eight external interrupt (hereinafter referred to as INT);

Each interrupt source can be independently enabled and a 2-level interrupt priority can be configured. Table 8-1 lists the interrupt vectors corresponding to the 11 interrupt sources and the corresponding interrupt source relationships.

Table 8-1. CMT2187A Interrupt Vector

Interrupt signal	Interrupt Vector	Interrupt source	Interrupt Request signal	Interrupt enable control	interrupt priority
0	0x0003	External interrupt 0	IE0	EX0	IPL0[0]
1	0x000B	Timer 0 interrupt	TF0	ET0	IPL0[1]
2	0x0013	External interrupt 1	IE1	EX1	IPL0[2]
3	0x001B	Timer 1 interrupt	TF1	ET1	IPL0[3]
4	0x0023	UART Interrupt	TI0/RI0	ES0	IPL0[4]
5	0x004B	External interrupt 2	IE2	EX2	IPL1[2]
6	0x0053	External interrupt 3	IE3	EX3	IPL1[3]
7	0x005B	External interrupt 4	IE4	EX4	IPL1[4]
8	0x0063	External interrupt 5	IE5	EX5	IPL1[5]
9	0x006B	External interrupt 6	IE6	EX6	IPL1[6]

10	0x0073	External interrupt 7	IE7	EX7	IPL1[7]
Notes: The T8051XC3 core controller has a minimum interrupt response time of 3 system clocks, which are provided by the internal 24MHz RC oscillator HFOSC or the external 26MHz crystal oscillator XOSC.					

8.4 External Interrupt Mapping

There are 11 interrupt sources supported by the T8051XC3 mentioned above. Except for the three interrupt sources of Timer 0, Timer 1 and UART are non-selectable interrupt trigger sources, the other eight external interrupts can be flexibly selected as the interrupt sources. The CPU core is connected to the peripheral interrupt sources via the INT BUS (interrupt bus). There are 27 external interrupt sources, listed as follows:

- I/O input interrupt function, D0-D11, a total of 12;
- Timer A/B module, each group of timers has 4 interrupts, a total of 8;
- 1 FIFO blank signal interrupt triggering of Sub-1G transmitting module;
- 2 SPI module transmitting and receiving data interrupt triggering;
- 2 Comparator output interrupt;
- 1 CDR output interrupt;
- 1 sleep timer interrupt

External interrupts INT0 and INT1 are used to connect the three wake sources, INT2- INT7 are used to connect the interrupt sources of each peripheral and I/O, and the mappings are different. The following uses INT0, INT1 and INT2 as examples to draw the interrupt structure diagram.

As shown in the figure below, INT0 is connected to the comparator interrupt and IO input interrupt, and any comparator or I/O edge detection will trigger INT0 to be used as the wakeup source in STOP mode; INT1 is fixed to the sleep timer interrupt and is also used as a wakeup source in STOP mode. After the CPU is woken up, it enters into the interrupt response program. User can query the FLAG by SFR to determine the specific interrupt that wakes up the system and perform related processing. INT2 mainly maps to peripheral interrupts, and I/O interrupts only maps to D0-D3. Noted that in this case, changes in D0-D3 will affect INT0 and may map to INT2. Users should configure interrupt enabling and mapping appropriately, using INT0 when stopping and INT2 when the program is running.

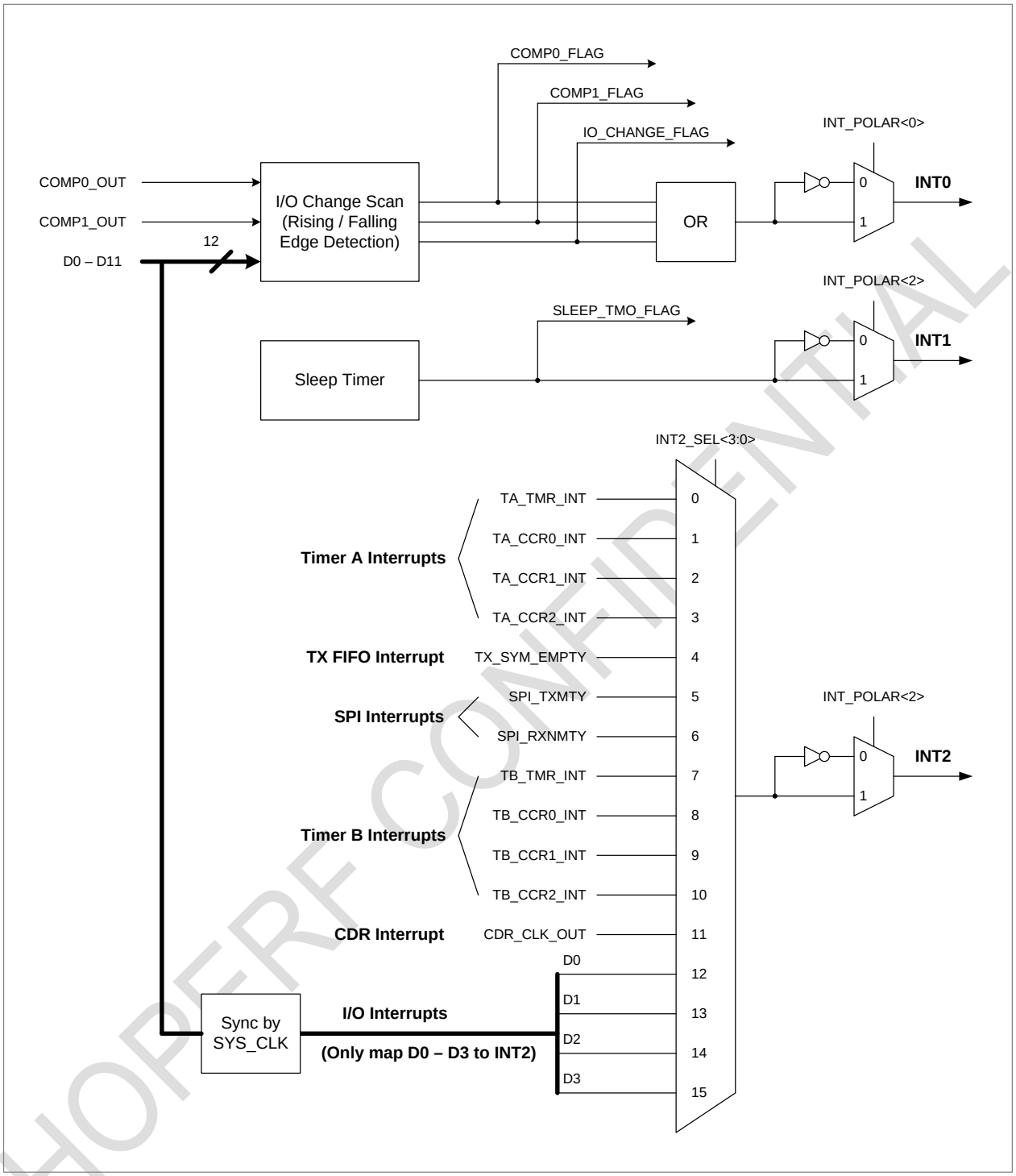

Chart 8-1. Peripheral INT0, INT1 and INT2 Mapping

The following is a detailed mapping between INT2-INT7 and each interrupt source:

Chart 8-2. Peripheral interrupt source and eight external interrupt mapping

Interrupt	Interrupt Selection	Interrupt source	Description	Interrupt	Interrupt Selection	Interrupt source	Description
INT2	INT2_SEL = 0000	TA_TMR_INT	Timer A Interrupt	INT3	INT3_SEL = 0000	TA_TMR_INT	Timer A Interrupt
	INT2_SEL = 0001	TA_CCR0_INT			INT3_SEL = 0001	TA_CCR0_INT	
	INT2_SEL = 0010	TA_CCR1_INT			INT3_SEL = 0010	TA_CCR1_INT	
	INT2_SEL = 0011	TA_CCR2_INT			INT3_SEL = 0011	TA_CCR2_INT	
	INT2_SEL = 0100	TX_SYM_EMPTY	Transmitter Interrupt		INT3_SEL = 0100	TX_SYM_EMPTY	Transmitter Interrupt
	INT2_SEL = 0101	SPI_TXMTY	SPI Interrupt		INT3_SEL = 0101	COMP0_OUT	Comparator interrupt
	INT2_SEL = 0110	SPI_RXNMTY			INT3_SEL = 0110	COMP1_OUT	
	INT2_SEL = 0111	TB_TMR_INT	Timer B Interrupt		INT3_SEL = 0111	TB_TMR_INT	Timer B Interrupt
	INT2_SEL = 1000	TB_CCR0_INT			INT3_SEL = 1000	TB_CCR0_INT	
	INT2_SEL = 1001	TB_CCR1_INT			INT3_SEL = 1001	TB_CCR1_INT	
	INT2_SEL = 1010	TB_CCR2_INT			INT3_SEL = 1010	TB_CCR2_INT	
	INT2_SEL = 1011	CDR_CLK_OUT	CDR Interrupt		INT3_SEL = 1011	CDR_CLK_OUT	CDR Interrupt
	INT2_SEL = 1100	D0	I/O input Interrupt		INT3_SEL = 1100	D4	I/O input Interrupt
	INT2_SEL = 1101	D1			INT3_SEL = 1101	D5	
	INT2_SEL = 1110	D2			INT3_SEL = 1110	D6	
	INT2_SEL = 1111	D3			INT3_SEL = 1111	D7	
INT4	INT4_SEL = 0000	TA_TMR_INT	Timer A Interrupt	INT5	INT5_SEL = 0000	TA_TMR_INT	Timer A Interrupt
	INT4_SEL = 0001	TA_CCR0_INT			INT5_SEL = 0001	TA_CCR0_INT	
	INT4_SEL = 0010	TA_CCR1_INT			INT5_SEL = 0010	TA_CCR1_INT	
	INT4_SEL = 0011	TA_CCR2_INT			INT5_SEL = 0011	TA_CCR2_INT	
	INT4_SEL = 0100	TX_SYM_EMPTY	Transmitter interrupt		INT5_SEL = 0100	TX_SYM_EMPTY	Transmitter interrupt
	INT4_SEL = 0101	SPI_TXMTY	SPI Interrupt		INT5_SEL = 0101	COMP0_OUT	Comparator interrupt
	INT4_SEL = 0110	SPI_RXNMTY			INT5_SEL = 0110	COMP1_OUT	
	INT4_SEL = 0111	TB_TMR_INT	Timer B Interrupt		INT5_SEL = 0111	TB_TMR_INT	Timer B Interrupt
	INT4_SEL = 1000	TB_CCR0_INT			INT5_SEL = 1000	TB_CCR0_INT	
	INT4_SEL = 1001	TB_CCR1_INT			INT5_SEL = 1001	TB_CCR1_INT	
	INT4_SEL = 1010	TB_CCR2_INT			INT5_SEL = 1010	TB_CCR2_INT	
	INT4_SEL = 1011	CDR_CLK_OUT	CDR Interrupt		INT5_SEL = 1011	CDR_CLK_OUT	CDR Interrupt
	INT4_SEL = 1100	D8	I/O input		INT5_SEL = 1100	D0	I/O input

Interrupt	Interrupt Selection	Interrupt source	Description	Interrupt	Interrupt Selection	Interrupt source	Description
	INT4_SEL = 1101	D9	Interrupt		INT5_SEL = 1101	D1	Interrupt
	INT4_SEL = 1110	D10			INT5_SEL = 1110	D2	
	INT4_SEL = 1111	D11			INT5_SEL = 1111	D3	
INT6	INT6_SEL = 0000	TA_TMR_INT	Timer A Interrupt	INT7	INT7_SEL= 0000	TA_TMR_INT	Timer A Interrupt
	INT6_SEL= 0001	TA_CCR0_INT			INT7_SEL = 0001	TA_CCR0_INT	
	INT6_SEL = 0010	TA_CCR1_INT			INT7_SEL = 0010	TA_CCR1_INT	
	INT6_SEL = 0011	TA_CCR2_INT			INT7_SEL = 0011	TA_CCR2_INT	
	INT6_SEL = 0100	TX_SYM_EMPTY	Transmitter interrupt		INT7_SEL = 0100	TX_SYM_EMPTY	Transmitter interrupt
	INT6_SEL = 0101	SPI_TXMTY	SPI Interrupt		INT7_SEL = 0101	COMP0_OUT	Comparator interrupt
	INT6_SEL = 0110	SPI_RXNMTY			INT7_SEL = 0110	COMP1_OUT	
	INT6_SEL = 0111	TB_TMR_INT	Timer B Interrupt		INT7_SEL = 0111	TB_TMR_INT	Timer B Interrupt
	INT6_SEL = 1000	TB_CCR0_INT			INT7_SEL = 1000	TB_CCR0_INT	
	INT6_SEL = 1001	TB_CCR1_INT			INT7_SEL = 1001	TB_CCR1_INT	
	INT6_SEL = 1010	TB_CCR2_INT			INT7_SEL = 1010	TB_CCR2_INT	
	INT6_SEL = 1011	CDR_CLK_OUT	CDR Interrupt		INT7_SEL = 1011	CDR_CLK_OUT	CDR Interrupt
	INT6_SEL = 1100	D4	I/O input Interrupt		INT7_SEL = 1100	D8	I/O input Interrupt
	INT6_SEL = 1101	D5			INT7_SEL = 1101	D9	
	INT6_SEL = 1110	D6			INT7_SEL = 1110	D10	
	INT6_SEL = 1111	D7			INT7_SEL = 1111	D11	

External interrupts INT0 and INT1 support both level and edge interrupts, while INT2-INT7 supports only edge interrupts. The polarity of the interrupt trigger is selected by the user by configuring the relevant SFR, the level trigger can be selected as a high or low level trigger, and the edge trigger can be selected as a rising or falling edge trigger.

8.5 Related Register

Table 8-3. System related register groups

Name	SFR page	address	default values	Function
TCON	0	0x88	0x00	Timer 1 Control Registers
IEN0	0	0xA8	0x00	Interrupt enabling register 0
IPL0	0	0xB8	0x00	Interrupt priority register 0

Name	SFR page	address	default values	Function
IEN1	0	0xE6	0x00	Interrupt enabling register 1
IRCON1	0	0xF1	0x00	Peripheral interrupt request flag register
IPL1	0	0xF6	0x00	Interrupt priority register 1
INTCTL_0	0	0x92	0x00	INT0 - INT7 Interrupt polarity selected register
INTCTL_1	0	0x93	0x00	INT2 and INT3 mapping configured registers
INTCTL_2	0	0x94	0x00	INT4 and INT5 mapping configured registers
INTCTL_3	0	0x95	0x00	INT6 and INT7 mapping configured registers

9 GPIO Module

9.1 Basic Function

The CMT2187A series chips support up to 6 GPIOs, which are GPIO0, GPIO1, GPIO6. GPIO7, GPIO8, GPIO10. And the operating modes are as followed.

Table 9-1. GPIO Operating Modes

Property 1	Property 2	Working Mode
Digital port[2]	Input mode[3]	Input mode only (floating input)
		Input mode with pull-up
		Input mode with pull-down
	Output mode	Open-drain output
		Push-pull output

Notes:

[1]. The analog input port serves as input of two analog comparators;

[2]. When it is used as a digital port, it is represented by D, such as D1, D2, etc., and the label corresponds to the serial number of GPIO.

[3]. Only when the GPIO operates in input mode, can it select to enable the IOC level detection function;

[4]. Support the selection of on-chip pull-up or pull-down with enabling control. The typical pull-up/ pull-down is 50 kΩ. Meanwhile, the pull-up also provides a very weak pull-up, typically a 500 kΩ pull-up resistor.

9.2 GPIO Structure Introduction

The functional block diagram of D0 to D11 is shown below:

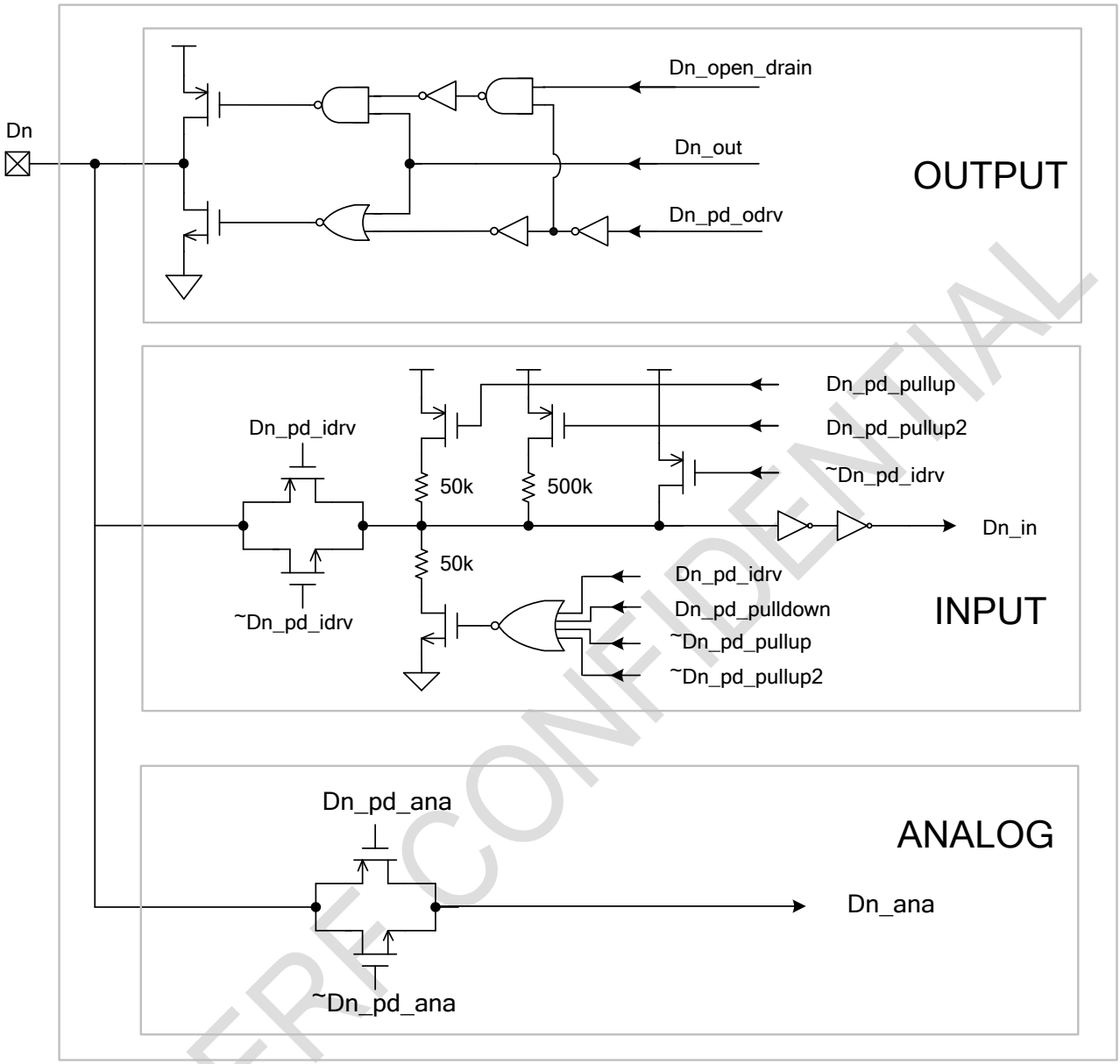

Chart 9-1. GPIO Functional Diagram

Table 9-2. Function Description of GPIOs

Port Name	Signal type	Description
Dn	General IO PAD	
Dn_open_drain	Configure register signal	The corresponding Dn open drain value is set in the 0x19 register AON_REG_19 and 0x1A register AON_REG_1A. 0: open_drain, 1: push_pull;
Dn_out	System internal control signal	When Dn is the digital output mode, Dn_out is the internal output signal.
Dn_pd_odrv	System internal	When Dn is used as the digital output mode, Dn_pd_odrv = 0,

Port Name	Signal type	Description
	control signal	otherwise Dn_pd_odrv = 1;
Dn_pd_idrv		When Dn is used as the digital input mode, Dn_pd_idrv = 0, otherwise Dn_pd_idrv = 1
	Configure register signal	When the Dn is used as a digital input mode, the corresponding registers are configured through the 0x15 registers AON_REG_15 and 0x16 registers AON_REG_16 to independently control whether the 50Kohm pull-up resistance of each Dn is enabled or not. Dn_pd_pullup， 0: Enable. 1 Disable
Dn_pd_pullup2	System internal control	When Dn is entered in digital input mode, whether all the digital input Dn of 500Kohm weak pullup resistance is enabled or not can be configured through the 0x10 register AON_REG_10[5] of pd_pullup_500K. 0: Enable，1: Disable ;
Dn_pd_pulldown	System internal control signal	When the Dn is entered in digital input mode, the corresponding registers are configured through the 0x17 registers AON_REG_17 and 0x18 registers AON_REG_18 to independently control whether the 50Kohm pull-down resistance of each Dn is enabled or not. Dn_pd_pulldown， 0: Enable，1: Disable; Note: When a pull-down resistor and a pull-up resistoris are enabled at the same time, the pull-up resistor has a higher priority.
Dn_pd_ana	System internal control signal	When Dn is in analog I/O mode, the internal analog signal line is connected to the system internal control signal via the 0x10 register AON_REG_10[4:0] Dn_ana.
Dn_ana	System internal control signal	Connect to the internal analog signal cable.

9.3 GPIO Digital Input

When GPIO is configured as a digital input:

- the output part is disabled.
- Whether the pull-up/pull-down resistance is enabled depends on the configuration related to IO pull-down in the AON register;
- The voltage on IO is sampled to the SFR registers of GPIO_IN_0 and GPIO_IN_1, which are readable by the software.

9.4 GPIO Digital Output

When GPIO is configured as output:

- the output channel is enabled.
- Open drain output mode：
  - If the output register is 0, the output NMOS is enabled.
  - If the output register is 1, the output NMOS and PMOS are disabled, and GPIO is in a high resistance state.
- Push-pull output mode:
  - If the output register is 0, the output NMOS is enabled and the output PMOS is disabled.
  - If the output register is 1, the output NMOS is disabled, and the output PMOS is enabled.
- In output mode, input mode is disabled and the input signal is pulled up internally, so GPIOn_in reads 1.

9.5 GPIO Analog Input and Output

When GPIO is configured in analog mode:

- The digital output function is disabled.
- The digital input mode is disabled and the input signal is forced to pull up internally, so GPIOn_in reads 1.

9.6 GPIO Digital Input Mapping

When the GPIO0-GPIO11 are in digital input mode, they are referred to as D0-D11, which can be used for I/O level flip detection to generate interrupts before the system clock SYS_CLK synchronization, and after synchronization they are used for the following three purposes:

- As an external input to the various peripherals
- As an input source for external interrupts INT2-INT7 (described in the External Interrupt Mapping section)
- As GPIO_IN_SFR<11:0>, user can read datas through the two SFR GPIO_IN_0<7:0> and GPIO_IN_1<7:0>

The following figure shows configuration t0_gpio_sel<3:0> as the external input of peripheral Timer 0

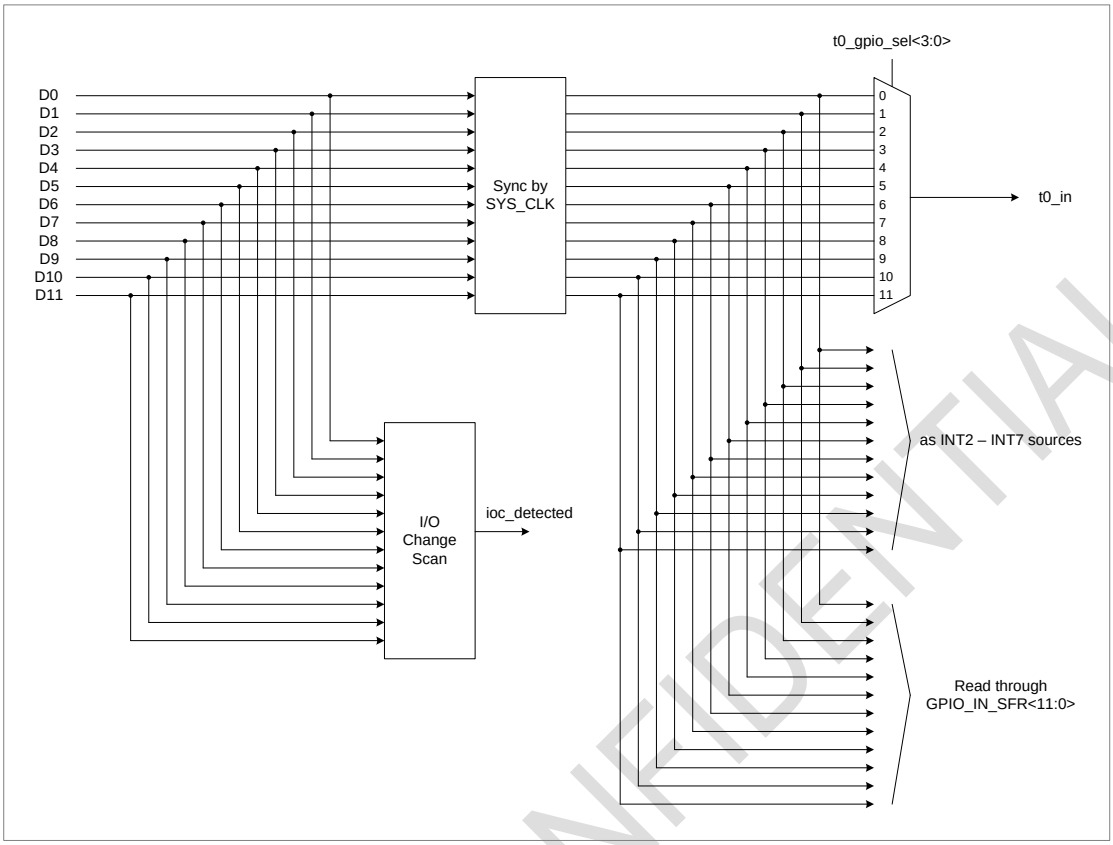

Chart 9-2. Functional Block Diagram of GPIO as Digital Inputs

By configuring the register t0_gpio_sel<3:0>, the user can select any digital input signal from D0-D11 that is synchronized by SYS_CLK to send to the external input t0_in of Timer 0. There are a total of 14 such as MUX, as shown in the following table:

Table 9-3. GPIO Input Mapping Register for each Peripheral

Peripheral Module	SFR	MUX Selection Signal	MUX Output	MUX Output Purpose
Timer 0	GPIO_INC_SEL	T0_GPIO_SEL<3:0>	t0_in	External signal input of the 8051 kernel initial peripheral Timer 0
		T0_INTN_GPIO_SEL<3:0>	t0_int0_n	Counting gate control input for the 8051 kernel initial peripheral Timer 0
Timer 1	GPIO_IND_SEL	T1_GPIO_SEL<3:0>	t1_in	External signal input of the 8051 kernel initial peripheral Timer 1
		T1_INTN_GPIO_SEL<3:0>	t1_int0_n	Counting gate input for the 8051 kernel initial peripheral Timer 1
Timer A	GPIO_INF_SEL	TA_CCI0_GPIO_SEL<3:0>	ta_cci0_in	One of the Timer A external capture source

Peripheral Module	SFR	MUX Selection Signal	MUX Output	MUX Output Purpose
		TA_CCI1_GPIO_SEL<3:0>	ta_cci1_in	One of the Timer A external capture source
Timer B	GPIO_ING_SEL	TB_CCI0_GPIO_SEL<3:0>	tb_cci0_in	One of the Timer B external capture source
		TB_CCI1_GPIO_SEL<3:0>	tb_cci1_in	One of the Timer B external capture source
SPI	GPIO_INA_SEL	NSS_IN_GPIO_SEL<3:0>	nss_in	Selected input of the SPI slave mode
		SCK_IN_GPIO_SEL<3:0>	sck_in	Clock input of the SPI slave mode
	GPIO_INB_SEL	MISO_IN_GPIO_SEL<3:0>	miso_in	Data input of the SPI slave mode
		MOSI_IN_GPIO_SEL<3:0>	mosi_int	Data output of the SPI master mode
UART	GPIO_INE_SEL	RXD0_GPIO_SEL<3:0>	rxd0_in	External input signal of the 8051 kernal initial peripheral UART
CDR		CDR_GPIO_SEL<3:0>	cdr_in	External signal input of CDR
Notes: [1] One GPIO input can be used as an external peripheral input, interrupt source, and map to the SFR at the same time, and user needs to avoid functional conflicts through proper configuration.				

9.7 GPIO Digital Output Mapping

When GPIO0-GPIO11 is used as a digital output, the output signal source can be configured by SFR and selected from the following table:

Table 9-4. Signal source that selected from the GPIOn

Select Item	Function
gpio_out_sfr[n]	GPIO_OUT_0 and GPIO_OUT_1 registers
port0_out[n]	Port0[7:0] output
Port1_out[n]	Port1[3:0] output
ta_out0	Output of Timer A capture/comparison in module 0
ta_out1	Output of Timer A capture/comparison in module 1
ta_out2	Output of Timer A capture/comparison in module 2
tb_out0	Output of Timer B capture/comparison in module 0
tb_out1	Output of Timer B capture/comparison in module 1
tb_out2	Output of Timer B capture/comparison in module 2
sck_out	Clock output of SPI master mode
nss_out	Selected output of SPI master mode

Select Item	Function
mosi_out	Data output of SPI master mode
miso_out	Data input of SPI slave mode
csb_out	Access register chip selection output of CMT specific 4-wire in SPI master mode
fcsb_out	Access FIFO chip selection output of CMT specific 4-wire in SPI master mode
rxd0_out	UART output enabling signal
txd0_out	UART clock or data output
T0_ov	Signal output of timer0 Module overflow
t1_ov	Signal output of timer0 Module overflow

The mapping between GPIOn and each output signal source is shown in the following table:

Table 9-5. Mappings between GPIOn and Function Module Output

GPIOn	SFR	Selected signal and code value	Output signal source
GPIO0	GPIO_OUTA_SEL	GPIO0_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<0>
		GPIO0_OUT_SEL<3:0> = 4'd1	port0_out<0>
		GPIO0_OUT_SEL<3:0> = 4'd2	tb_ccr0_out
		GPIO0_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO0_OUT_SEL<3:0> = 4'd4	tb_ccr2_out
		GPIO0_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO0_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO0_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO0_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO0_OUT_SEL<3:0> = 4'd9	fcsb_out
		GPIO0_OUT_SEL<3:0> = 4'd10	txd0_out
		GPIO0_OUT_SEL<3:0> = 4'd11	ta_ccr0_out
		GPIO0_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO0_OUT_SEL<3:0> = 4'd13	ta_ccr2_out
		GPIO0_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO0_OUT_SEL<3:0> = 4'd15	t1_ov_out
GPIO1	GPIO_OUTA_SEL	GPIO1_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<1>
		GPIO1_OUT_SEL<3:0> = 4'd1	port0_out<1>
		GPIO1_OUT_SEL<3:0> = 4'd2	tb_ccr0_out
		GPIO1_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO1_OUT_SEL<3:0> = 4'd4	tb_ccr2_out
		GPIO1_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO1_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO1_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO1_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO1_OUT_SEL<3:0> = 4'd9	rxd0_out

GPIOn	SFR	Selected signal and code value	Output signal source
		GPIO1_OUT_SEL<3:0> = 4'd10	csb_out
		GPIO1_OUT_SEL<3:0> = 4'd11	ta_ccr0_out
		GPIO1_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO1_OUT_SEL<3:0> = 4'd13	ta_ccr2_out
		GPIO1_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO1_OUT_SEL<3:0> = 4'd15	t1_ov_out
GPIO2	GPIO_OUTB_SEL	GPIO2_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<2>
		GPIO2_OUT_SEL<3:0> = 4'd1	port0_out<2>
		GPIO2_OUT_SEL<3:0> = 4'd2	tb_ccr0_out
		GPIO2_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO2_OUT_SEL<3:0> = 4'd4	tb_ccr2_out
		GPIO2_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO2_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO2_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO2_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO2_OUT_SEL<3:0> = 4'd9	rxd0_out
		GPIO2_OUT_SEL<3:0> = 4'd10	txd0_out
		GPIO2_OUT_SEL<3:0> = 4'd11	ta_ccr0_out
		GPIO2_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO2_OUT_SEL<3:0> = 4'd13	ta_ccr2_out
		GPIO2_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO2_OUT_SEL<3:0> = 4'd15	t1_ov_out
GPIO3	GPIO_OUTB_SEL	GPIO3_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<3>
		GPIO3_OUT_SEL<3:0> = 4'd1	port0_out<3>
		GPIO3_OUT_SEL<3:0> = 4'd2	tb_ccr0_out
		GPIO3_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO3_OUT_SEL<3:0> = 4'd4	tb_ccr2_out
		GPIO3_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO3_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO3_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO3_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO3_OUT_SEL<3:0> = 4'd9	rxd0_out
		GPIO3_OUT_SEL<3:0> = 4'd10	txd0_out
		GPIO3_OUT_SEL<3:0> = 4'd11	ta_ccr0_out
		GPIO3_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO3_OUT_SEL<3:0> = 4'd13	ta_ccr2_out
		GPIO3_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO3_OUT_SEL<3:0> = 4'd15	t1_ov_out
GPIO4	GPIO_OUTB_SEL	GPIO4_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<4>
		GPIO4_OUT_SEL<3:0> = 4'd1	port0_out<4>
		GPIO4_OUT_SEL<3:0> = 4'd2	tb_ccr0_out

GPIOn	SFR	Selected signal and code value	Output signal source
		GPIO4_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO4_OUT_SEL<3:0> = 4'd4	tb_ccr2_out
		GPIO4_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO4_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO4_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO4_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO4_OUT_SEL<3:0> = 4'd9	fcsb_out
		GPIO4_OUT_SEL<3:0> = 4'd10	txd0_out
		GPIO4_OUT_SEL<3:0> = 4'd11	ta_ccr0_out
		GPIO4_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO4_OUT_SEL<3:0> = 4'd13	ta_ccr2_out
		GPIO4_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO4_OUT_SEL<3:0> = 4'd15	t1_ov_out
GPIO5	GPIO_OUTC_SEL	GPIO5_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<5>
		GPIO5_OUT_SEL<3:0> = 4'd1	port0_out<5>
		GPIO5_OUT_SEL<3:0> = 4'd2	tb_ccr0_out
		GPIO5_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO5_OUT_SEL<3:0> = 4'd4	tb_ccr2_out
		GPIO5_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO5_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO5_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO5_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO5_OUT_SEL<3:0> = 4'd9	fcsb_out
		GPIO5_OUT_SEL<3:0> = 4'd10	txd0_out
		GPIO5_OUT_SEL<3:0> = 4'd11	ta_ccr0_out
		GPIO5_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO5_OUT_SEL<3:0> = 4'd13	ta_ccr2_out
		GPIO5_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO5_OUT_SEL<3:0> = 4'd15	t1_ov_out
GPIO6	GPIO_OUTD_SEL	GPIO6_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<6>
		GPIO6_OUT_SEL<3:0> = 4'd1	port0_out<6>
		GPIO6_OUT_SEL<3:0> = 4'd2	tb_ccr0_out
		GPIO6_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO6_OUT_SEL<3:0> = 4'd4	tb_ccr2_out
		GPIO6_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO6_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO6_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO6_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO6_OUT_SEL<3:0> = 4'd9	rxd0_out
		GPIO6_OUT_SEL<3:0> = 4'd10	txd0_out
		GPIO6_OUT_SEL<3:0> = 4'd11	ta_ccr0_out

GPIOn	SFR	Selected signal and code value	Output signal source
		GPIO6_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO6_OUT_SEL<3:0> = 4'd13	ta_ccr2_out
		GPIO6_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO6_OUT_SEL<3:0> = 4'd15	t1_ov_out
GPIO7	GPIO_OUTD_SEL	GPIO7_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<7>
		GPIO7_OUT_SEL<3:0> = 4'd1	port0_out<7>
		GPIO7_OUT_SEL<3:0> = 4'd2	tb_ccr0_out
		GPIO7_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO7_OUT_SEL<3:0> = 4'd4	tb_ccr2_out
		GPIO7_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO7_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO7_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO7_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO7_OUT_SEL<3:0> = 4'd9	rxd0_out
		GPIO7_OUT_SEL<3:0> = 4'd10	txd0_out
		GPIO7_OUT_SEL<3:0> = 4'd11	ta_ccr0_out
		GPIO7_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO7_OUT_SEL<3:0> = 4'd13	ta_ccr2_out
		GPIO7_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO7_OUT_SEL<3:0> = 4'd15	t1_ov_out
GPIO8	GPIO_OUTE_SEL	GPIO8_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<8>
		GPIO8_OUT_SEL<3:0> = 4'd1	port1_out<0>
		GPIO8_OUT_SEL<3:0> = 4'd2	tb_ccr0_out
		GPIO8_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO8_OUT_SEL<3:0> = 4'd4	tb_ccr2_out
		GPIO8_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO8_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO8_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO8_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO8_OUT_SEL<3:0> = 4'd9	rxd0_out
		GPIO8_OUT_SEL<3:0> = 4'd10	txd0_out
		GPIO8_OUT_SEL<3:0> = 4'd11	ta_ccr0_out
		GPIO8_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO8_OUT_SEL<3:0> = 4'd13	ta_ccr2_out
		GPIO8_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO8_OUT_SEL<3:0> = 4'd15	t1_ov_out
GPIO9	GPIO_OUTE_SEL	GPIO9_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<9>
		GPIO9_OUT_SEL<3:0> = 4'd1	port1_out<1>
		GPIO9_OUT_SEL<3:0> = 4'd2	tb_ccr0_out
		GPIO9_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO9_OUT_SEL<3:0> = 4'd4	tb_ccr2_out

GPIOn	SFR	Selected signal and code value	Output signal source
		GPIO9_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO9_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO9_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO9_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO9_OUT_SEL<3:0> = 4'd9	rxd0_out
		GPIO9_OUT_SEL<3:0> = 4'd10	txd0_out
		GPIO9_OUT_SEL<3:0> = 4'd11	ta_ccr0_out
		GPIO9_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO9_OUT_SEL<3:0> = 4'd13	ta_ccr2_out
		GPIO9_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO9_OUT_SEL<3:0> = 4'd15	t1_ov_out
GPIO10	GPIO_OUTF_SEL	GPIO10_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<10>
		GPIO10_OUT_SEL<3:0> = 4'd1	port0_out<2>
		GPIO10_OUT_SEL<3:0> = 4'd2	tb_ccr0_out
		GPIO10_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO10_OUT_SEL<3:0> = 4'd4	tb_ccr2_out
		GPIO10_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO10_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO10_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO10_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO10_OUT_SEL<3:0> = 4'd9	rxd0_out
		GPIO10_OUT_SEL<3:0> = 4'd10	txd0_out
		GPIO10_OUT_SEL<3:0> = 4'd11	ta_ccr0_out
		GPIO10_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO10_OUT_SEL<3:0> = 4'd13	ta_ccr2_out
		GPIO10_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO10_OUT_SEL<3:0> = 4'd15	t1_ov_out
GPIO11	GPIO_OUTF_SEL	GPIO11_OUT_SEL<3:0> = 4'd0	gpio_out_sfr<11>
		GPIO11_OUT_SEL<3:0> = 4'd1	port0_out<3>
		GPIO11_OUT_SEL<3:0> = 4'd2	tb_ccr0_out
		GPIO11_OUT_SEL<3:0> = 4'd3	tb_cc1_out
		GPIO11_OUT_SEL<3:0> = 4'd4	tb_ccr2_out
		GPIO11_OUT_SEL<3:0> = 4'd5	nss_out
		GPIO11_OUT_SEL<3:0> = 4'd6	sck_out
		GPIO11_OUT_SEL<3:0> = 4'd7	miso_out
		GPIO11_OUT_SEL<3:0> = 4'd8	mosi_out
		GPIO11_OUT_SEL<3:0> = 4'd9	rxd0_out
		GPIO11_OUT_SEL<3:0> = 4'd10	txd0_out
		GPIO11_OUT_SEL<3:0> = 4'd11	ta_ccr0_out
		GPIO11_OUT_SEL<3:0> = 4'd12	ta_cc1_out
		GPIO11_OUT_SEL<3:0> = 4'd13	ta_ccr2_out

GPIOn	SFR	Selected signal and code value	Output signal source
		GPIO11_OUT_SEL<3:0> = 4'd14	t0_ov_out
		GPIO11_OUT_SEL<3:0> = 4'd15	t1_ov_out
Notes: [1]. The default mapping is controlled by the GPIO_OUT_SFR<11:0> register group and for the reason that the register group does not support the bit access mode, so the control output needs to follow the "read-change-write" mode.			

9.8 GPIO Level Flipping Detection

The GPIO level flip detection (I/O Change Scan) module located in the AON domain is used to detect the level flip of any I/O after the system enters the STOP mode, generating an interrupt source that wakes the system and sends it to the external interrupt INT0. When the system is running normally, users need to disable this module to avoid conflicts with other I/O functions.

To use this module, process of user program is shown as followed:

1. The module can only be opened when STOP, and the IO_EVENT_RST_N bit of WKINT_STA register in SFR PAGE0 is maintained as 0 when the program is running, that is, the whole module is in the reset state.
2. After selecting the GPIO to be detected, users can start configuring the GPIO and detection module via AON REG. Since the module is in reset mode, there will not have any misdetection during the configuration process.
3. The GPIO must be configured in digital input mode with digital output function disabled.
4. Configure the pull-up/pull-down resistance of the GPIO;
5. Configure the reversing polarity of the GPIO (rising or falling edge);
6. Enable the GPIO level flipping detection;
7. Set IO_EVENT_RST_N to 1 to release the detection module and configure INT0 through SFR;
8. The system enters the STOP mode.
9. When the GPIO detects a level flip and wakes up the system via INT0, it sets the IO_EVEN_RST_N bit to 0 after the relevant IO query and processing.

The common application scenario of this module is that GPIO connects external keys. Two connection methods are usually used. The following shows example of connecting keys with D0-D11:

- Independent key connection

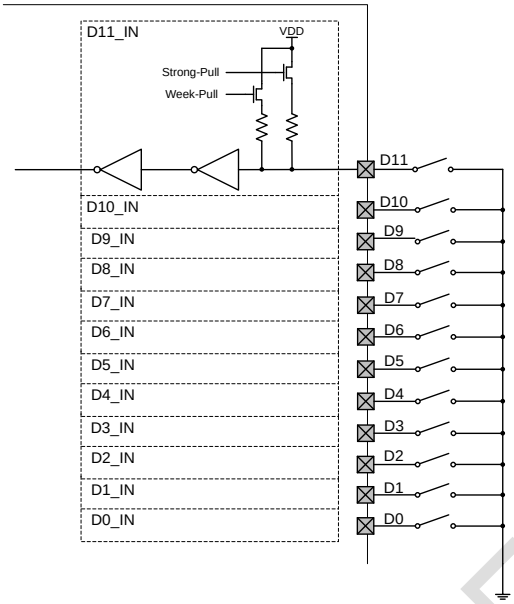

Chart 9-3. Independent Key Connection Diagram

D0-D11 in the figure below are all connected to the ground, so these ports can all enble the pull-up digital input port mode, and configure Dn_POLAR (n generally refers to any number from 0-11, each IO has a corresponding polarity selection bit) to 1, that is, the normal state is 1. When a key is pressed to port 0 and the falling edge is detected, the interrupt will be triggered.

- Matrix key connection

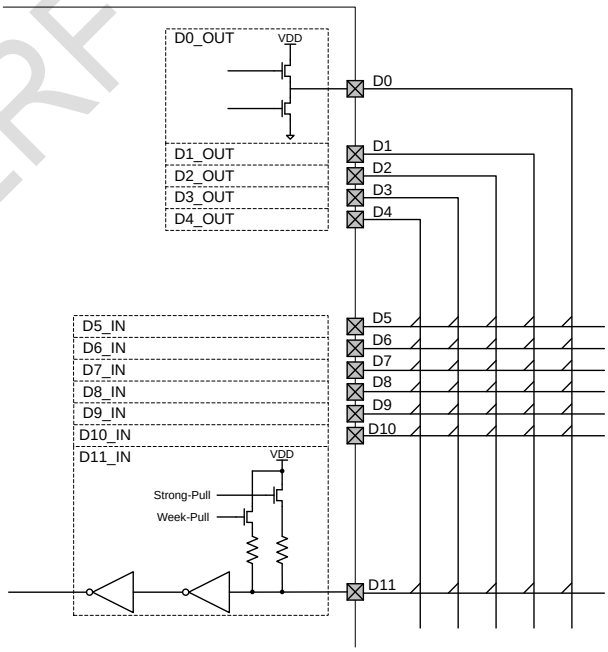

Chart 9-4. Matrix Key Connection Diagram

According to the above connection method, D0-D11 is a matrix keyboard consist of D5-D11, which is used as a set of input detection, and D0-D4, which is used as a set of output control. Before entering the STOP mode, user needs to configure D0-D4 as digital output mode, with output value of 0. The D5-D11 group of digital inputs can be configured to enable the detection function as independent keys. After entering STOP mode, any key on the Matrix keyboard will wake up the system via D5-D11. The software can normally scan the keyboard as general procedure to identify keys after waking up the system.

9.9 Related Register

Table 9-6. AON register groups of GPIO

Name	Address	Default Values	Funtion
AON_REG_10	0x10	0xE0	Enabling GPIO analog functions and configure the pull up and pull down resistance value
AON_REG_11	0x11	0x00	Digital output enabling bit of D0 - D7
AON_REG_12	0x12	0x00	Digital output of the enabling bit of D8 - D11
AON_REG_13	0x13	0x00	Digital input enabling bit of D0 - D7
AON_REG_14	0x14	0x00	Digital input enabling bit of D8 - D11
AON_REG_15	0x15	0xFF	Pull up resistance switch of D0 - D7
AON_REG_16	0x16	0x0F	Pull up resistance switch of D8 - D11
AON_REG_17	0x17	0xFF	Pull down resistance switch of D0 - D7
AON_REG_18	0x18	0x0F	Pull down resistance switch of D8 - D11
AON_REG_19	0x19	0xFF	Open drain switch of D0 - D7
AON_REG_1A	0x1A	0x0F	Open drain switch of D8 - D11
AON_REG_1B	0x1B	0x00	Level flip detection enable bit for D0-D7
AON_REG_1C	0x1C	0x00	Level flip detection enable bit for D8-D11
AON_REG_1D	0x1D	0x00	Level flip detection polarity selection for D0-D7
AON_REG_1E	0x1E	0x00	Level flip detection polarity selection for D8-D11

Table 9-7. SFR register groups of GPIO

Name	SFR page	address	default values	Functions
GPIO_INA_SEL	0	0xA1	0x00	GPIO Input function mapping
GPIO_INB_SEL	0	0xA2	0x00	GPIO Input function mapping
GPIO_INC_SEL	0	0xA3	0x00	GPIO Input function mapping
GPIO_IND_SEL	0	0xA4	0x00	GPIO Input function mapping
GPIO_INE_SEL	0	0xA5	0x00	GPIO Input function mapping
GPIO_INF_SEL	0	0xA6	0x00	GPIO Input function mapping

Name	SFR page	address	default values	Functions
GPIO_ING_SEL	0	0xA9	0x00	GPIO Input function mapping
GPIO_OUTA_SEL	0	0xAA	0x00	GPIO output function mapping
GPIO_OUTB_SEL	0	0xAB	0x00	GPIO output function mapping
GPIO_OUTC_SEL	0	0xAC	0x00	GPIO output function mapping
GPIO_OUTD_SEL	0	0xAD	0x00	GPIO output function mapping
GPIO_OUTE_SEL	0	0xB0	0x00	GPIO output function mapping
GPIO_OUTF_SEL	0	0xB1	0x00	GPIO output function mapping
GPIO_OUT_0	0	0xB3	0x00	GPIO output data via SFR configuration
GPIO_OUT_1	0	0xB4	0x00	GPIO output data via SFR configuration
GPIO_IN_0	0	0xB5	0x00	Readable GPIO input data from SFR
GPIO_IN_1	0	0xB6	0x00	Readable GPIO input data from SFR

10 Timer0 Module

10.1 Basic Function

Timer0 is a 16-Bit programmable timer/counter that can be configured with TMOD registers to select how it works, start or stop counting, and generate counting overflow interrupts. Timer0 supports 3 working modes and they are shown as Table 10-1.

Table 10-1. Working Modes in Timer0

TMOD. M01	TMOD.M00	Working Mode	Functions
0	0	Mode 0	8-Bit timing/counter with 5 bit prescale, i.e. 13-Bit timing/count mode
0	1	Mode 1	16-Bit timer/counter mode
1	0	Mode 2	8-Bit timing/counting mode with overloaded initial values
1	1	disable	

10.2 Timer0 Mode0

Block diagram of Timer0 mode 0 is shown below.

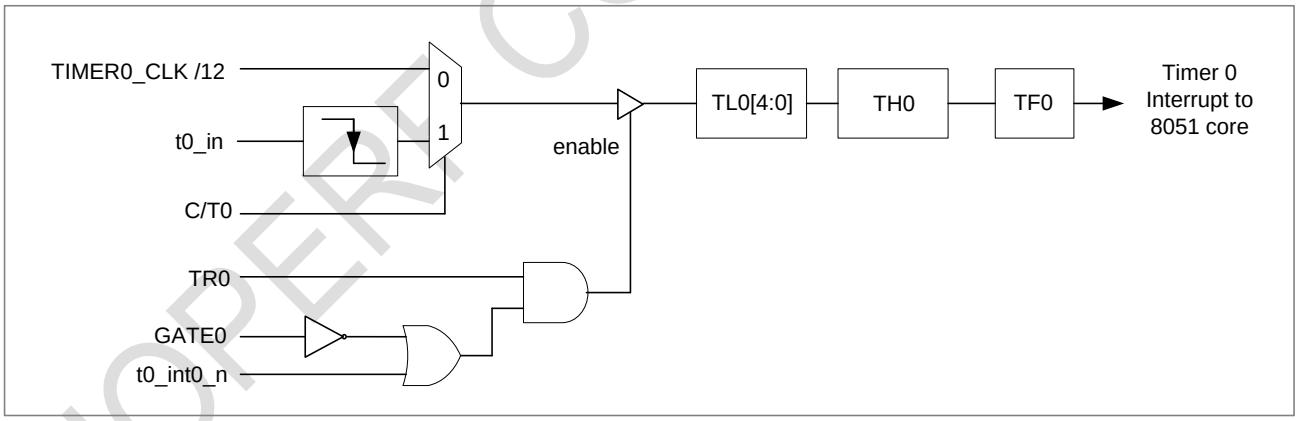

Chart 10-1. Block diagram of Timer 0 mode 0

When Timer0 works in Mode0, a 13-bit counter is combined from a 5-bit prescaler provided by TL0[4:0] and an 8-bit counter provided by TH0 :

- If TMOD.C/T0 is set to 0, timing mode is selected, and the timing clock source is the 12-prescale of FPCLK.
- If TMOD.C/T0 is set to 1, counting mode is selected, and the falling edge of the external input pin t0 is used as the counting signal.

- If TMOD.GATE0 is set to 1, the gate control trigger counting is selected. it requires high level of external interrupt t0_int0_n and Ton.tr0 to trigger Timer0 counting.
- If TMOD.GATE0 is set to 0, Timer0 counting is triggered as the TCON.TR0 is set to 1
- The total switch of Timer0 counting is TCON.TR0, the counting function is enabled when it is set 1 and disable when it is set 0.

10.3 Timer0 Mode1

Block diagram of Timer0 mode 1 is shown in the chart below.

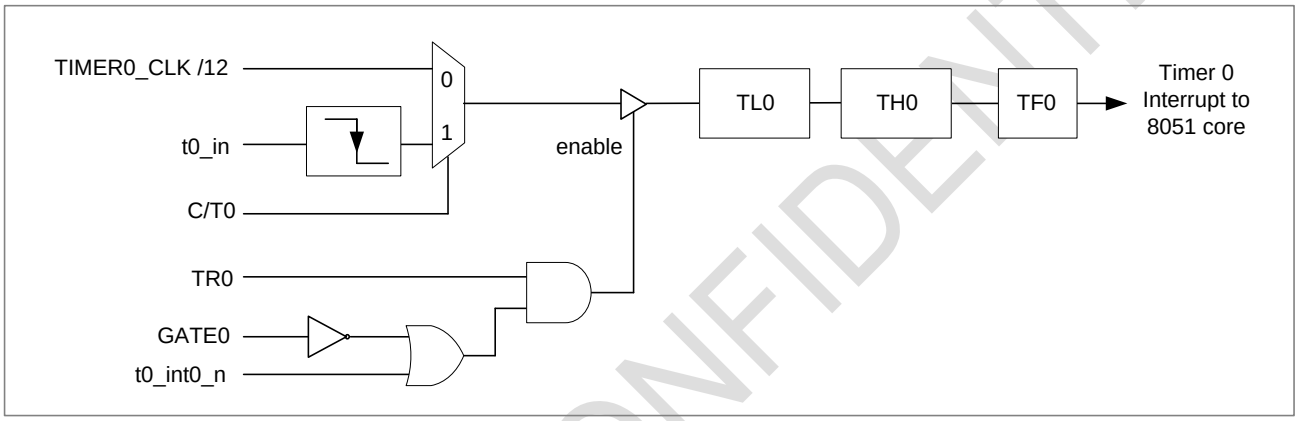

Chart 10-2. Block diagram of Timer 0 mode 1

When TIMER0 works in Mode0, it is a 16-bit counter, consisting of TL0 and TH0, the difference from Mode0 is only in the number of counter bits. Functions of other control bits are the same as Mode0.

10.4 Timer0 Mode2

Block diagram of Timer0 mode 2 is shown in the chart below.

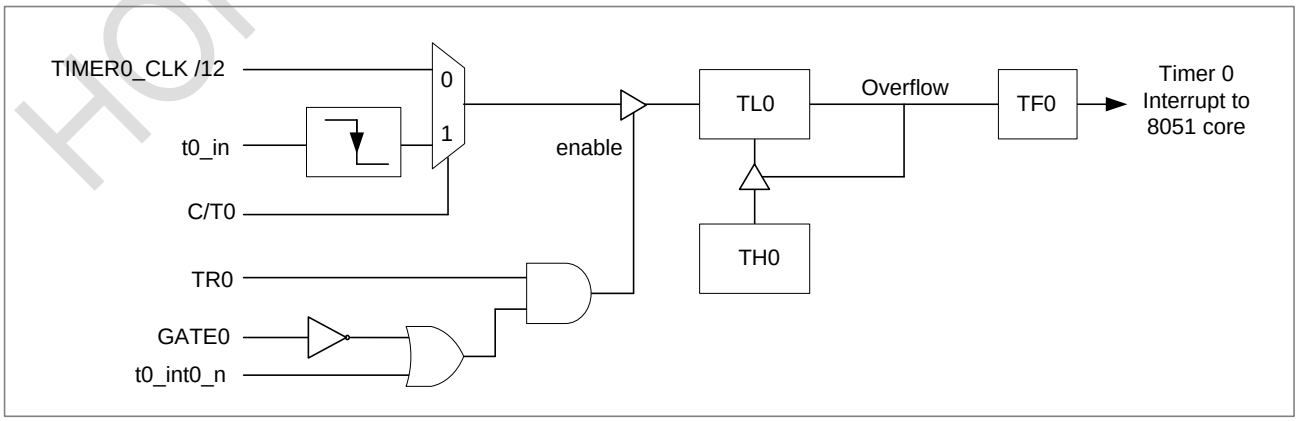

Chart 10-3. Block diagram of Timer 0 mode 2

When Timer0 works in Mode2, the counter is automatically overloads with initial value of 8 bits, and when the TL0 count overflows, it automatically loads the value saved by TH0 (the initial value), making TL0 re-count from the initial value. This is the main difference from Mode0/1, where the counter is cleared to zero after the count overflows. While other control bits function is the same as Moode0/1.

10.5 Related Register

Table 10-2. Register Group of Timer0 Module

Name	SFR page	address	default values	Functions
TCON	0	0x88	0x00	Timer0 and Timer1 Control Registers
TMOD	0	0x89	0x00	Timer0 and Timer1 Working Mode Registers
TL0	0	0x8A	0x00	Timer0 register low 8 bit
TH0	0	0x8C	0x00	Timer0 register high 8 bit

11 Timer1 Module

11.1 Basic Function

Timer1 is a 16-Bit programmable timer/counter that can be configured with TMOD registers to select how it works, start or stop counting, and generate count overflow interrupts. Timer 1 supports 3 kinds of working modes, which are shown in the following table.

Table 11-1. Work Modes in Timer 1

TMOD. M11	TMOD.M10	Work Mode	Functions
0	0	Mode 0	8-Bit timing/counter with 5-bit prescale, i.e. 13-Bit timing/count mode
0	1	Mode 1	16-Bit timing/counting mode
1	0	Mode 2	8-Bit timing/counting mode with overloaded initial values
1	1	Disable	

11.2 Timer1 Mode0

Block diagram of Timer1 mode 0 is shown below.

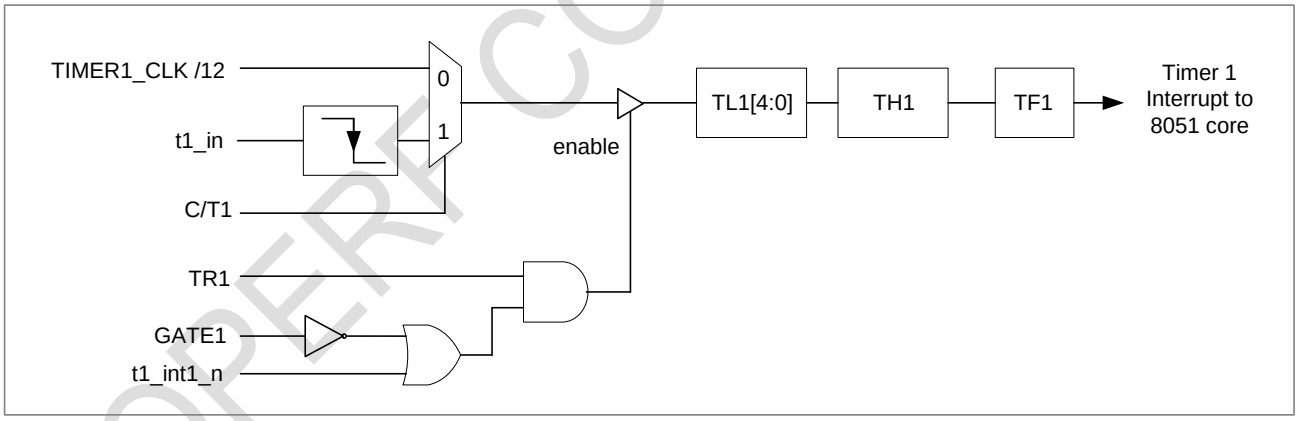

Chart 11-1. Block Diagram of Timer1 Mode0

When Timer1 works in Mode0, a 13-bit counter is combined from a 5-bit prescaler provided by TL1[4:0] and an 8-bit counter provided by TH1:

- If TMOD.C/T1 is set to 0, timing mode is selected, and the timing clock source is the 12-prescale of FPCLK.
- If TMOD.C/T1 is set to 1, counting mode is selected, and the falling edge of the external input pin t1 is used as the counting signal.

- If TMOD.GATE1 is set to 1, the gate control trigger counting is selected. it requires high level of external interrupt t1_int1_n and TCON.TR1 to trigger Timer1 counting.
- If TMOD.GATE1 is set to 0,Timer1 counting is triggered as the TCON.TR1 is set to 1
- The total switch of Timer1 counting is TCON.TR1, the counting function is enabled when it is set 1 and disable when it is set 0.

11.3 Timer1 Mode1

Block diagram of Timer1 mode 1 is shown below.

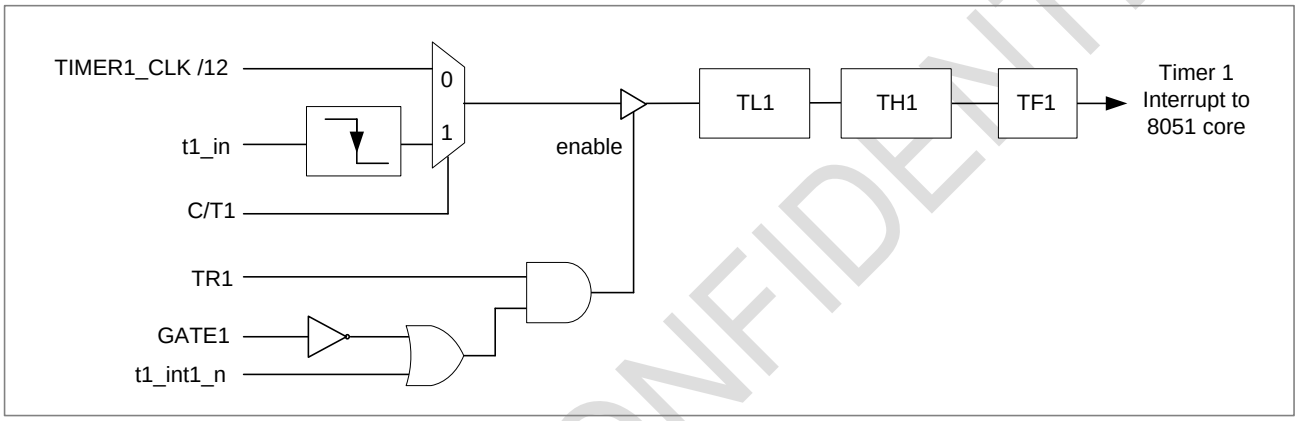

Chart 11-2. Block Diagram in Timer1 Mode1

When TIMER1 works in Mode1, it is a 16-bit counter, consisting of TL1 and TH1, and differs from Mode0 only in the number of counter bits. While other control bit function is the same as Moode0.

11.4 Timer1 Mode2

Block diagram of Timer1 mode 2 is shown below.

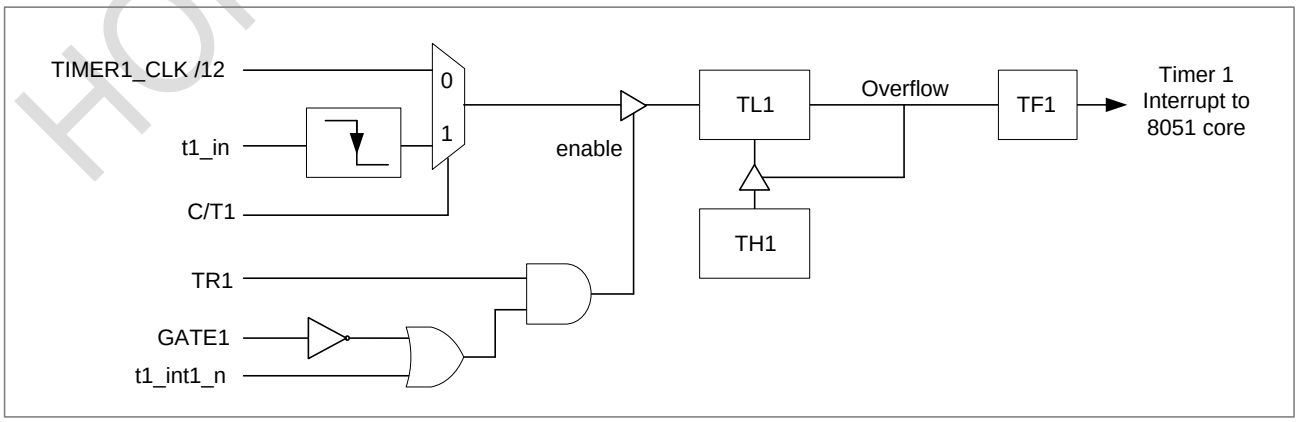

Chart 11-3. Block Diagram of Timer1 Mode2

When Timer1 works in Mode2, the counter is automatically overloads with initial value of 8 bits, and when the TL1 count overflows, it automatically loads the value saved by TH1 (the initial value), making TL1 re-count from the initial value.

This is the main difference from Mode0/1, where the counter is cleared to zero after the count overflows.

While other control bit function is the same as Moode0/1.

11.5 Related Register

Table 11-2. Register Group of Timer1 Module

Name	SFR page	address	default values	Functions
TCON	0	0x88	0x00	Timer0 and Timer1 Control Registers
TMOD	0	0x89	0x00	Timer0 and Timer1 Working Mode Registers
TL1	0	0x8B	0x00	Timer1 register low 8 bit
TH1	0	0x8D	0x00	Timer1 register high 8 bit
USART_SEL	0	0x9F	0x01	Prescaler options of Timer1 clock source

12 SPI Module

12.1 Basic Function

The serial peripheral interface (SPI) allows the chip and peripherals to communicate in a half/full duplex, synchronous and serial manner. It supports master mode and slave mode from the operating manner view. The master provides communication clock to the slave. The interface also supports multi-master configuration or 1-wire bi-directional simplex synchronous transmission (3-wire mode).

Usually, 4 pins need to be connected between the master and slave SPI devices.

- MISO: master-in-slave-out pin. The pins are received from master device and transmitted form slave device. The pin can send data from the slave device to the master device.
- MOSI; master-out-slave-in pin. The pins received from slave device and transmitted form master device. The pin can send data from the master device to the slave device.
- SCK: serial data synchronous clock, which is output from the master device to the slave device.
- NSS：slave device selection enabling, an optional pin used for the master device to select the target slave device. It allows the master device to communicate with a specific slave device individually, avoiding conflicts on the data lines. The NSS pin of the slave device can be driven by the master device as a standard IO. Once enabled (SSOE bit), the NSS pin can also be used as an output pin and can be pulled low when the SPI is set to master mode; At this time, all SPI devices with their NSS pins connecting to the master NSS pin will get low levels and will enter into slave mode automatically if they are set as NSS hardware mode.

The SPI interface timing of master and slave mode are shown in Chart 12-1 and Chart 12-2 respectively. Among them, the sampling clock phase CPHA of SCK is configured by register SPI_CTL_1.SPI_EDGE_SEL. If it is 1, SPI will send data on the first edge of the clock and sample data on the second edge; if it is set 0, SPI will sample data on the first edge of the clock and send data on the second edge. CPOL determines the status of the SCK when it is idle. When CPOL is 1, the SCK line will remain high during idle. When CPOL is 0, the SCK line will remain low during idle.

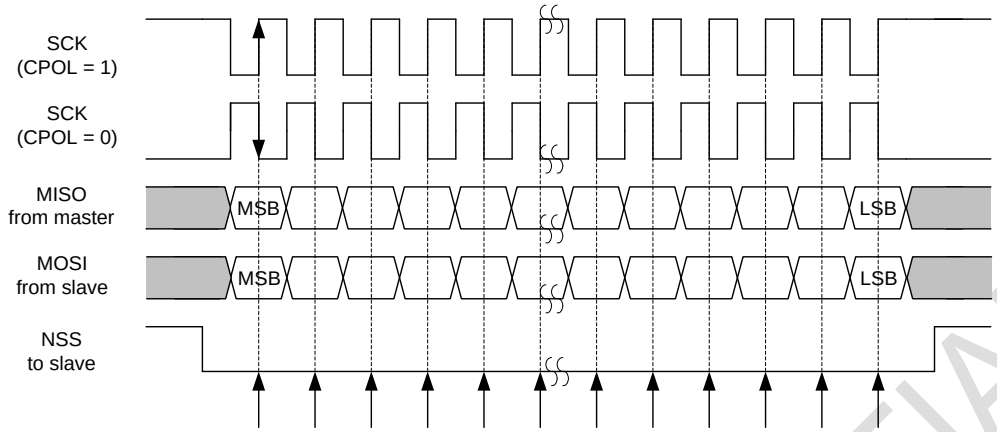

Chart 12-1. SPI Interface Timing（CPHA = 1）

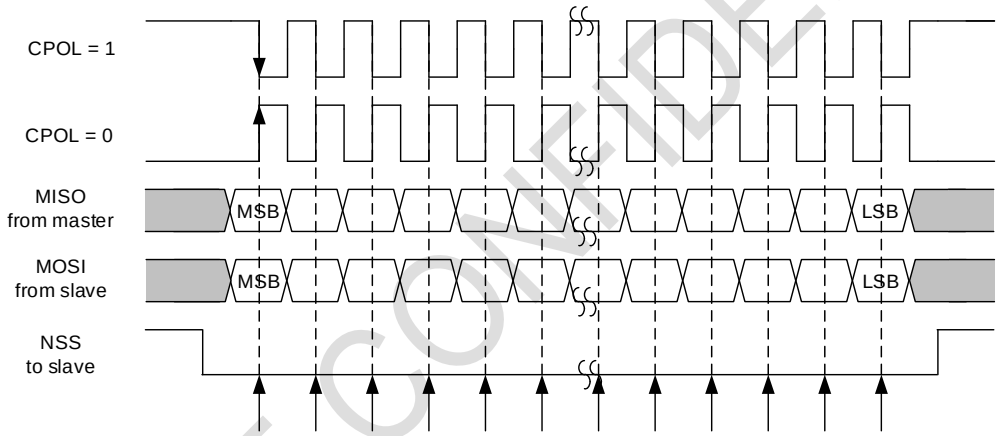

Chart 12-2. SPI Interface Timing（CPHA = 0）

12.2 Configuration Option

- **Master Mode and Slave Mode Setting**

The SPI module contains the complete program of master mode and slave mode. User can set the currently controlled SPI as a master or slave device by configuring register SPI_CTL_1.SPI_MS_SEL, so as to communicate with the other slave or master devices. If SPI_MS_SEL is set to 1'b1, the current SPI module will be set to master mode; if SPI_MS_SELl is set to 1'b0, the current SPI module will be set to slave mode.

- **Configuration of Clock Phase and Clock Polarity**

To set the clock phase, the variable SPI_CTL_1.SPI_EDGE_SEL needs to be configured. If this variable is set to 1, SPI will sample the second edge of the clock, and if 0 is set, SPI will sample the first edge of the clock.

Clock polarity can be set by configuring register SPI_CTL_1.SPI_CKPOL_SEL. If SPI_CKPOL_SEL is set 1,

SCK will be high in IDLE state; if SPI_CKPOL_SEL is set 0, SCK will be low in IDLE state. The following figure shows the edge of data sampling and transmitting when configured with different clock polarities and phases.

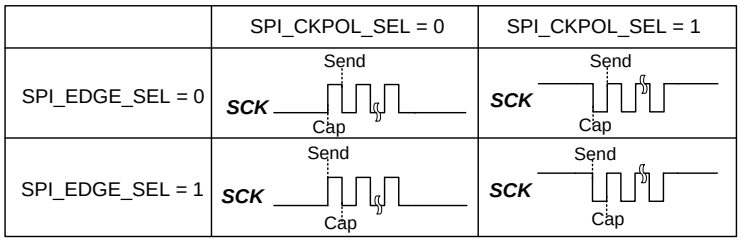

Chart 12-3. Diagram of SPI Interface Timing （CPHA = 1）

- **Setting of the transmitting data bit width**

When register SPI_CTL_0.SPI_8B16B_SEL is set to 1, SPI will select to transmit 16bit data, master device will transmit 16 cycles of clocks to the slave device, and the slave will also collect 16 bits of data in turn along the corresponding clock. When SPI_8B16B_SEL is set to 0, only 8 bits of data are transmitted, which are the lower 8 bits of the 16 bits in SPI_DATA.

- **Baud Rate Setting**

Table 12-1. SPI Module Rate

SPI_CTL_0.SPI_MBR Setting Value	Transfer rate:（sys_clk is the system clock, default by 24Mhz）
3'b000	sys_clk/4
3'b001	sys_clk/8
3'b010	sys_clk/16
3'b011	sys_clk/24
3'b100	sys_clk/32
3'b101	sys_clk/64
3'b110	sys_clk/128
3'b111	sys_clk/256

- **The setting of high-low bit priority transmission**

If the input signal SPI_CTL_1.SPI_LSBF is set to 1, the low-level data will be transmitted first; if it is set to 0, the high-level data will be transmitted first.

12.3 Working Mode

Both master and slave device have four working modes and they can be separated into 2 types, which

includes full duplex and half-duplex. Three variables of SPI_CTL_0.SPI_BIDI_MODE ; SPI_CTL_0.SPI_BIDI_OEN；SPI_CTL_0.SPI_RX_ONLY can be configured as the following list and acheive 4 working modes.

Table 12-2. SPI Module Operating Modes

SPI_BIDI_MODE	SPI_RX_ONLY	SPI_BIDI_OEN	Mode Selection Priority	Working Mode
1'b0	1'b0	Random value	First Level	Full duplex normal mode
1'b0	1'b1	Random value	Second level	Full duplex read only mode
1'b1	Random value	1'b1	Third level	Half duplex write only mode
1'b1	Random value	1'b0	Fourth level	Half duplex **read only mode**

1. Full-duplex normal mode: When a master or slave device is configured in this mode, data is sent and received synchronously. Taking the master device as an example, the data transmitting port MOSI sends data out, while the MISO port also receives data from the slave device.

2. Full duplex read-only mode: The receiving end of the master device and slave device in this mode works normally, while the transmitting end always sends 0.

3. Half-duplex write only mode: In this mode, a host or slave device has only three external interfaces. For example, on a master device, SCK sends out clocks, NSS transmission is enabled, and the last I/O port is allocated to MOSI to transmitting data.

3. Half-duplex read only mode: In this mode, a host or slave device has only three external interfaces. For example, on a master device, SCK sends out clocks, NSS transmission is enabled, and the last I/O port is allocated to MOSI to receive data.

Application:
The SPI can communicate with the S3S interface by switching back and forth between read-only and write only in half duplex. When the SPI wants to send an address to the S3S interface, it can be set to the half-duplex write only mode, and map the MOSI to the PAD to connect with the SDA of the S3S to realize the address writing. When the SPI wants to read out the data transmit by S3S, it can set the SPI to half-duplex read-only mode and map the MISO to the PAD to connect to the SDA of S3S in order to read out the data transmitted by S3S. In this case, data communication between SPI and S3S can be realized when only three physical wires are connected.

12.4 Status Flag

The SPI module provides the bus status through 3 status flags to serve software monitoring.

- Busy flag（SPI_CTL2.SPI_BUSY flag bit），when the SPI is in the process of transmission, the flag bit wil be pulled high.

- Send buffer free flag (SPI_CTL2.SPI_TXMTY flag bit); When the SPI is configured with new transmiting data, the flag bit will be pull low. When the new transmitting data is configured and transmited successfully, the flag bit will be pull high.

- Receive buffer is not empty (SPI_CTL2.SPI_RXNMTY flag bit); When the SPI completes a cycle of data transmission, it pulls up the SPI_RXNMTY flag bit to indicate that the data has been successfully sent and received. When the user reads the data received in SPI_DATA, the flag bit will be automatically pull low.

12.5 Related Register

Table 12-3. Register Group of SPI Module

Name	SFR page	address	default values	Functions
SPI_CTL_0	Page0	0x96	0x00	SPI module control register 0
SPI_CTL_1	Page0	0x97	0x00	SPI module control register 1
SPI_CTL_2	Page0	0x9A	0x00	SPI module control register 2
SPI_DATA_H	Page0	0x9B	0x00	SPI module data high bytes
SPI_DATA_L	Page0	0x9C	0x00	SPI module data low bytes

13 UART Module

13.1 Basic Function

The on-chip UART of CMT2187A is a flexible full-duplex asynchronous transceiver fully compatible with the 8051 architecture. The baud rate is configured by the software and supports the following 4 operating modes.

- Mode0: synchronous shift mode, with baud rate fixed to UART_CLK / 12
- Mode1：8-Bit UART mode，with configurable baud rate generated by the internal Timer 1；
- Mode2：9-Bit UART mode，with baud rate of UART_CLK / 64 or UART_CLK / 32；
- Mode3：9-Bit UART mode，with configurable baud rate generated by the internal Timer 1；

Users can select the interrupts of UART module's operating mode, communication enabling, and data transmitting/receiving by configuring SCON0.

13.2 Synchronous Shift Mode (Mode0)

The synchronous shift mode is a synchronous operating mode of the UART module, similar to the SPI slave mode, to have serial communication with other 8051s. It adopts half-duplex communication with the serial line RxD being used as bidirectional input & output data interface, and the TxD generating the fixed serial baud rate clock with a clock frequency of UART_CLK / 12.. When SCON0.SM00 and SCON0.SM10 are configured as 00, the UART module selects to operate in Mode 0, and its transmission architecture is shown in the below figure.

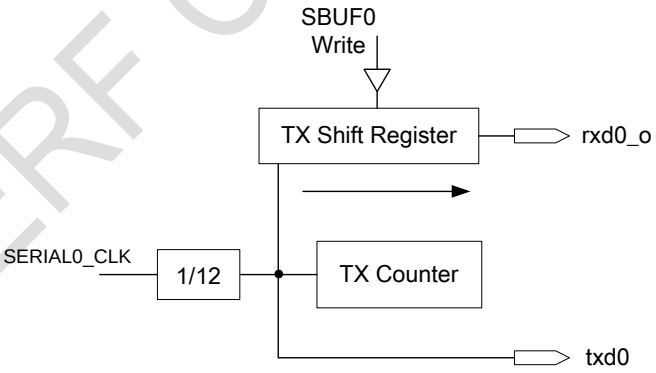

Chart 13-1. Transmission Block Diagram for UART in Mode 0

In the process of transmission, when data is written to the transmitting buffer SBUF, a positive pulse is generated, and the serial port starts to send data. At this time, SCON0.REN0 needs to be set as 0. After transmission complete, the sending interrupt flag bit SCON0.TI0 is set to 1. The sequence diagram is as follows.

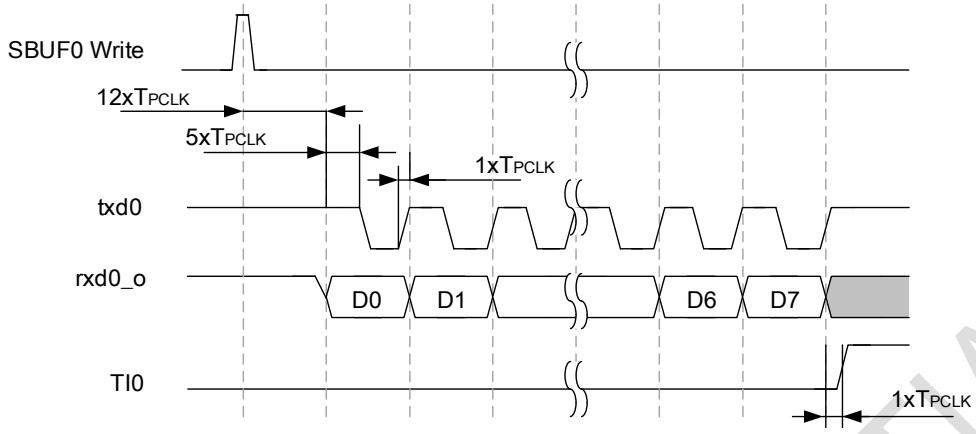

Chart 13-2. Transmission Timing Diagram for UART in Mode 0

Mode0 receiving structure diagram is shown in the below figure.

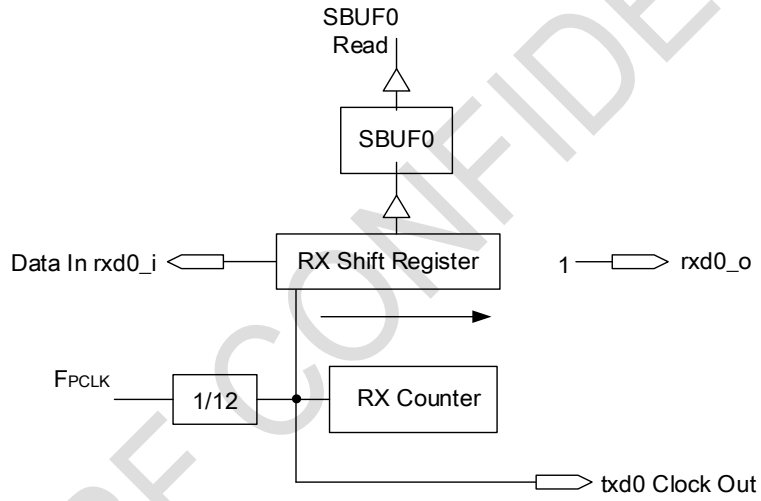

Chart 13-3. Receiving Diagram for UART in Mode 0

When it receives in mode0, the receiving enabling bit SCON0.REN 0 is set 1, and when the receiving interrupt flag bit SCON0.RI0 is cleared, a positive pulse is generated and the serial port starts receiving data. The TxD output shift clock samples the data on the RxD line. When 8-bit data is received, the receiving interrupt flag bit SCON0.RI0 is set 1 and the data is buffered in SBUF0. The timing diagram is shown in the below figure.

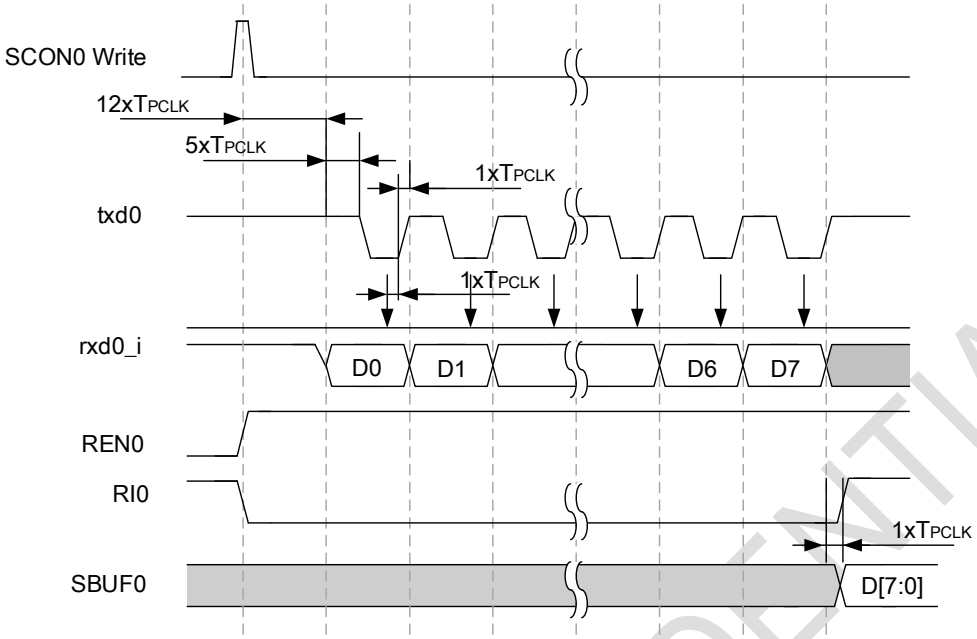

Chart 13-4. Receiving Timing Diagram of UART in Mode 0

13.3 Asynchronous Full-duplex Mode with Configurable Baud Rate (Mode 1 and Mode 3)

Both mode 1 and mode 3 of the UART module are asynchronous full-duplex transceiver mode with variable baud rates. The only difference between them is that mode 1 is an 8-bit data transmitting and receiving mode, while mode 3 is a 9-bit data transmitting and receiving mode.

Settings Serial Port 0 as Mode1:

- Configure SCON0.SM00 and SCON0.SM10 as 01.
- Generate baud rate through Timer1 and PCON.SMOD1 bits.
- When receiving, set the receiving enabling bit SCON0.REN0 to 1.
- When sending, write data to the transmission buffer SBUF.

Settings Serial Port 0 as Mode3:

- Configure SCON0.SM00 and SCON0.SM10 as 11.
- Generate baud rate through Timer1 and PCON.SMOD1 bits.
- When receiving, set the receiving enabling bit SCON0.REN0 to 1.
- When transmitting, write data to the transmission buffer SBUF and SCON0.TB8.

The frame formats of mode 1 and mode 3 are shown in the below figures.

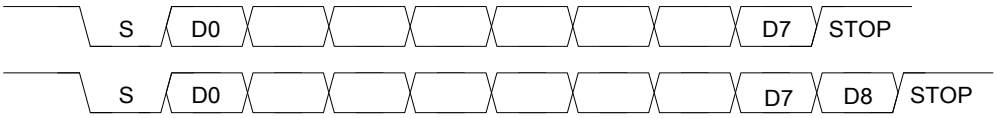

Chart 13-5. Frame Format of Mode 1 and Mode 3 of UART

The baud rates in mode 1 and mode 3 depend on overflow rate of Timer1. The transmission and receiving block diagrams are shown in the below figures.

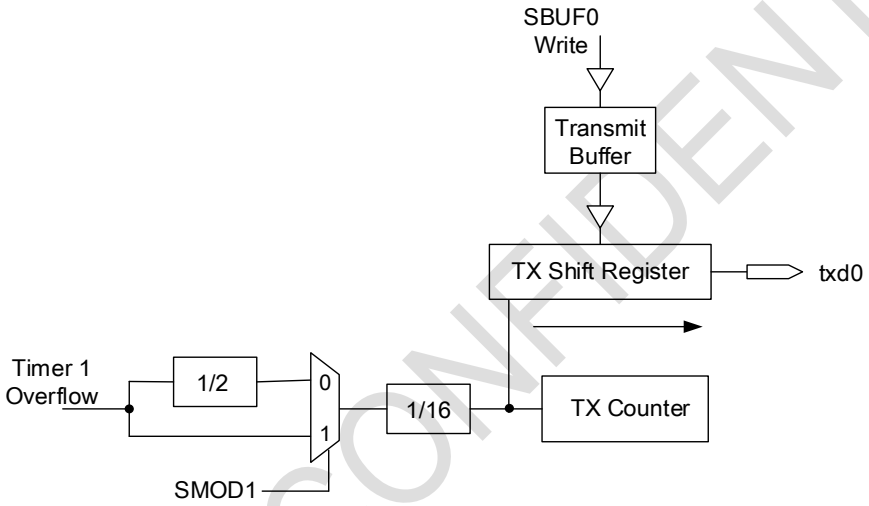

Chart 13-6. Transmission Block Diagram of UART in Mode 1 and Mode 3

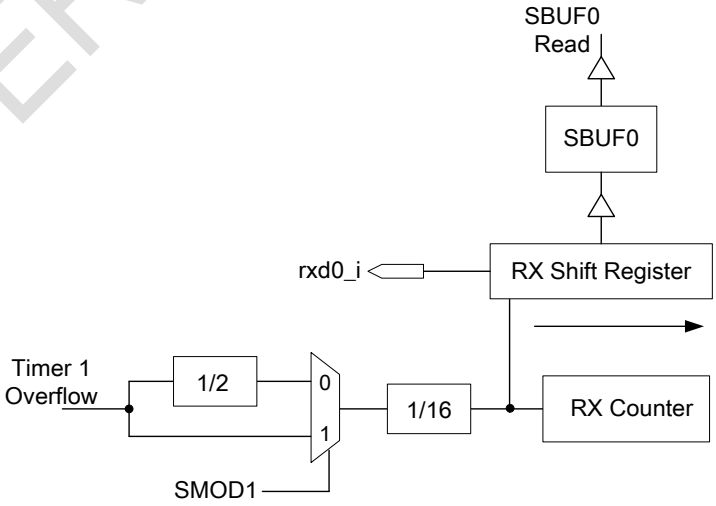

Chart 13-7. Receiving Block Diagram of UART in Mode 1 and Mode 3

Configure Timer1 to generate the baud rate:

- TMOD.M1[1:0] is configured as 10 (Timer1 in mode 2), TMOD.GATE1 is configured as 0, and TMOD.C/T1 is configured as 0.
- Write the 8-bit initial count value to the TH1.
- TCON.TR1 is configured as 1 and starts timing.

The baud rate is calculated according to the following formula:

$$\text{Baud\_Rate} = 2^{SMOD1} \times \frac{F_{PCLK}}{[32 \times 12 \times (256 - TH1)]}$$

It is equivalent to:

$$\text{Baud}_{\text{Rate}} = 2^{SMOD1} \times \frac{Timer1_{Overflow\_Rate}}{32}$$

During transmission, when 8-bit data is written to the transmission buffer SBUF0 (in mode3, write the data to SCON0.TB8 first, then write the 8-bit data to SBUF0), a positive pulse is generated, and the serial port starts transmitting data. First send the 1-bit as start bit 0, then send the LSB bit of SBUF, send the 8-bit data in turn, then send SCON0.TB8 (mode 3), and finally send the 1-bit stop bit 1. When the data transmission completes, set the transmission interrupt flag bit SCON0.TI0 to 1. The timing diagram is shown in the figure below.

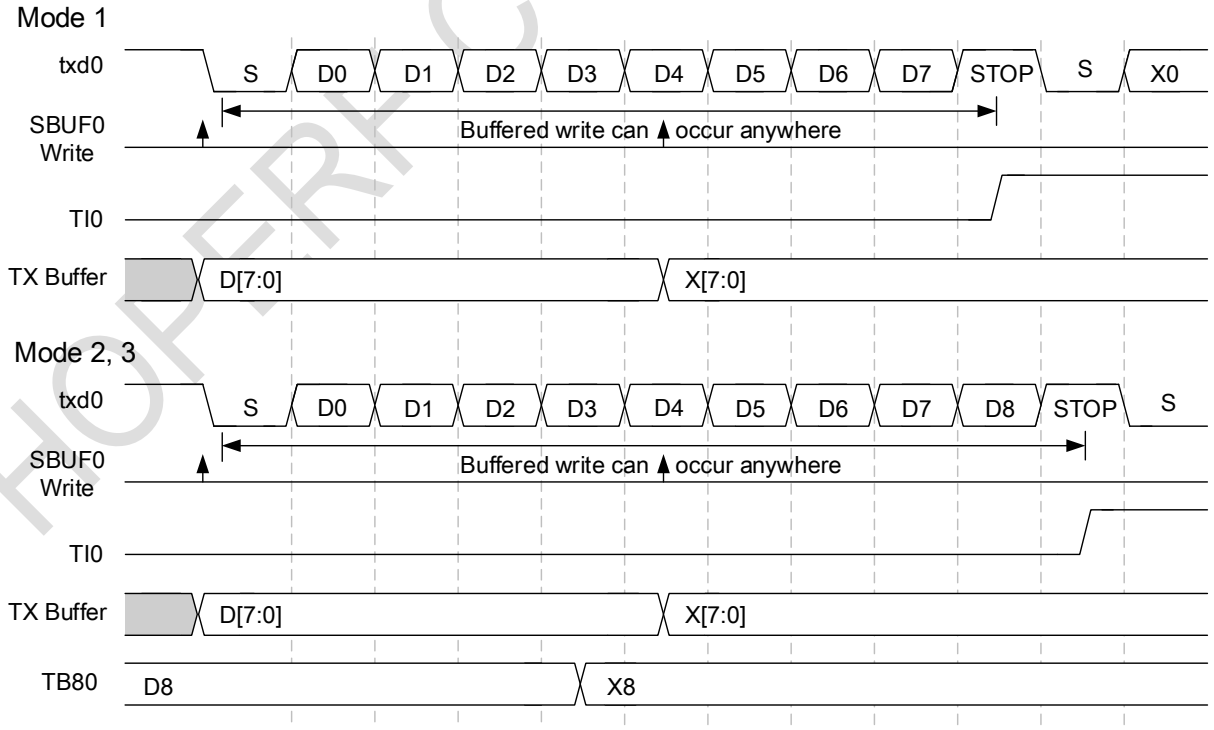

Chart 13-8. Asynchronous Transmission Timing Diagram for UART in Mode 1 and Mode 3

When the mode 1 or mode 3 receiving mode of UART is enabled, the receiving enable bit SCON0.REN0 must be set 1 first, then the RxD pin state is sampled at 16 times of the baud rate, then wait for the falling edge of the start bit. While the falling edge is sampled, it is checked again whether RxD is 0 during start bit sampling, to confirm whether it is a valid start bit. If it is not 0, continue to sample the start bit. When it is confirmed as a valid start bit, it starts to receive 8-bit / 9-bit data, and each bit of data is sampled in the middle position. In the 8-bit receiving state, checking that whether the sampling stop bit is valid and copy it to SCON0.RB80, and set the receiving interrupt flag bit SCON0.RI0 to 1. If the stop bit is wrong, the FE0 (framing error) flag is set to 1. In the 9-bit receiving, after sampling the 9th bit, the interrupt bit flag SCON0.RI0 is set to 1 in the position of the stop bit, and the 9th bit is written to SCON0.RB80. The stop bit error is only used to generate the FE0 flag. When SCON0.RI0 is set to 1, SBUF loads the received 8-bit data. The timing diagram is shown in the figure below.

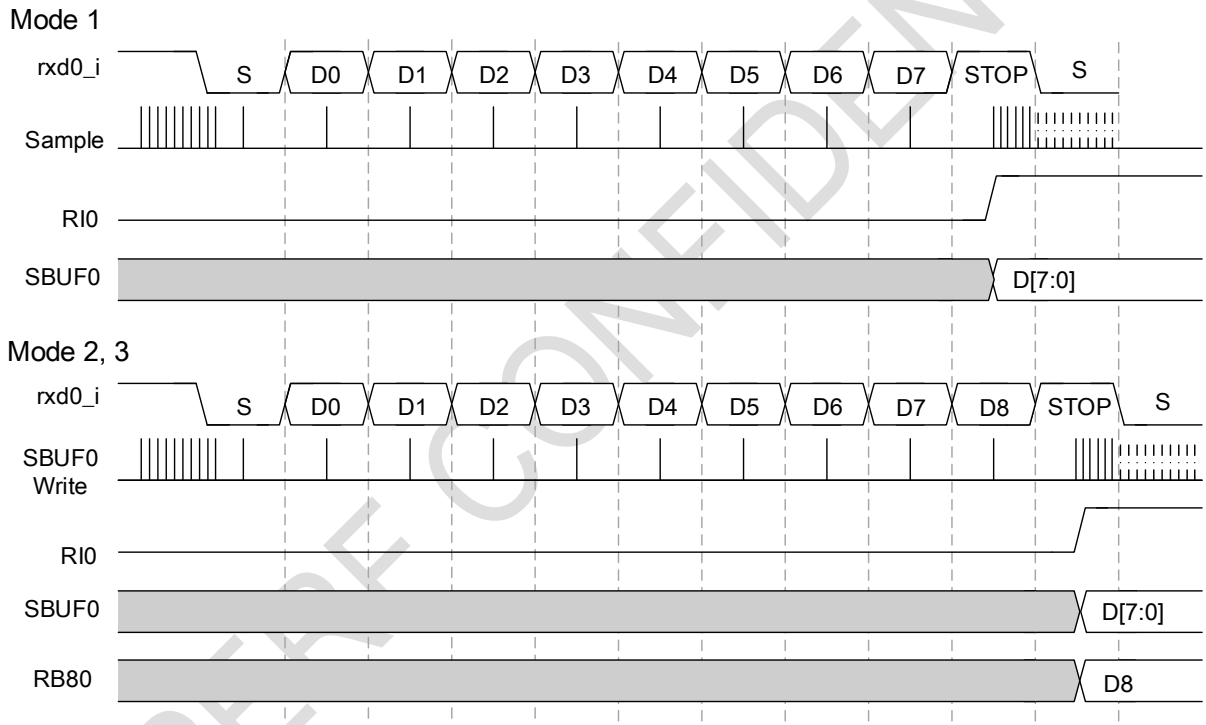

Chart 13-9. Asynchronous Receiving Timing Diagram of UART (Mode 1 and Mode 3)

13.4 Asynchronous Full-duplex Mode with Fixed Baud Rate (Mode 2)

Mode 2 is a 9-bit asynchronous full-duplex transceiver operating mode with fixed baud rate. The transmitting and receiving block diagrams are shown in the below figure.

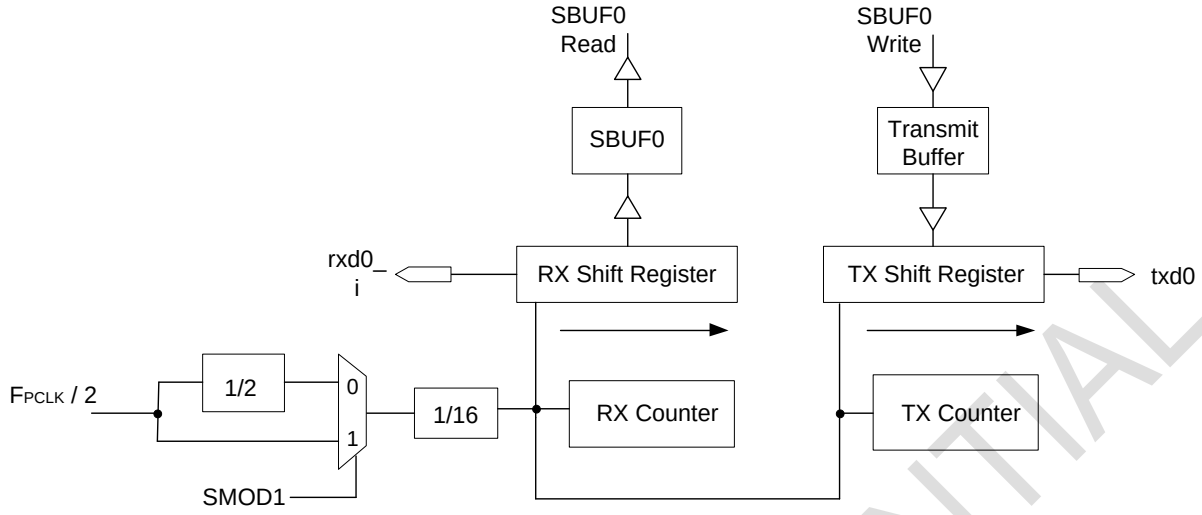

Chart 13-10. Transmitting and Receiving Block Diagram of UART Mode 2

Steps to setup UART to mode 2 are as follows.

- Configure SCON0.SM00 and SCON0.SM10 as 10.
- When receiving, set the receiving enabling bit SCON0.REN0 to 1.
- When transmitting, write data to the transmission buffer SBUF and SCON0.TB8.

The only difference between mode 2 and mode 3 is the baud rate difference. The baud rate of mode 2 is calculated as follows.

$$\text{Baud\_Rate} = 2^{\text{SMOD1}} \times \frac{F_{\text{PCLK}}}{64}$$

Both mode 2 and mode 3 of UART support multi-machine communication, which just needs to set SCON0.SM20 to 1. In this mode, a master can send data to multiple slaves through the serial line. A salve can be recognized by the master only when it receives the 9th bit of RB8 data as 1. The remaining 8-bit data is used to transmit the slave address. It can receive full data only if the address is matched. The further data stream is only sent to the identified slave devices. The 9th bit of the data stream needs to be set to 0 thus other slaves cannot recognize the data.

13.5 Enhanced Mode of USART

Among the various operating modes of UART, only in mode 1 and mode 3 can achieve different baud rates according to the Timer1 overflow period. It's the most simple but practical when Timer 1 operating in mode 2. However, the CMT2187A HFOSC clock source supports relatively limited frequencies, namely 24 MHz and external 13 MHz (generated by 26 MHz/2). Therefore, the corresponding baud rate options are also limited. Detail informations are listed in the below table.

Table 13-1. UART Baud Rate Error in Standard Mode (Timer1 in Mode2)

Target baud rate	SMOD=1			SMOD=0		
	Setting Value.of TH1	Actual Baud Rate:	Error Value	Setting Value.of TH1	Actual Baud Rate:	Error Value
300	--	--	--	48	300.48	0.16%
600	48	601	0.17%	152	601	0.17%
1200	152	1202	0.17%	204	1202	0.17%
2400	204	2404	0.17%	230	2404	0.17%
4800	230	4808	0.17%	243	4808	0.17%
9600	243	9615	0.16%	249	8929	-6.99%
14400	247	13889	-3.55%	252	15625	8.51%
19200	249	17857	-6.99%	253	20833	8.51%

Note: Select the HFOSC clock with frequency of 24MHz (i.e. F_{PCLK} = 24MHz).

Table 13-1 shows that only the six common low rate of 300, 600, 1200, 2400, 4800, 9600 are less errors and meet the requirement of using. The higher baud rate is basically unable to be used. In order to enable CMT2187A to support more baud rate options, it is embedded with enhanced mode on chip. CMT2187A can set USART_SEL (located in SFR register USART_CTL) to 1, otherwise canceling the 12 frequency divider in front of the Timer1 clock source and directly provide Timer1 from FPCLK as the clock source, as shown in the following figure.

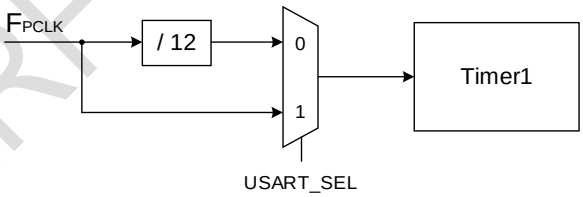

Chart 13-11. Schematic diagram of the Timer1 clock source in enhanced mode

- When USART_SEL=0, the Timer1 clock source is F_{PCLK}/12 according to the standard 51 architecture.
- When USART_SEL=1, the Timer1 clock source is in enhanced mode and is directly provided by F_{PCLK}.

The baud rate errors calculated in enhanced mode are shown below.

Table 13-2. UART baud rate errors in enhanced mode (Timer1 Mode 2)

Target Baud rate	SMOD=1			SMOD=0		
	TH1 Setting Value	Actual Baud rate	Error rate	TH1 Setting Value	Actual Baud rate:	Error rate
4800	--	--	--	100	4808	0.16%
9600	100	9615	0.16%	178	9615	0.16%
14400	152	14423	0.16%	204	14423	0.16%
19200	178	19230	0.16%	217	19231	0.16%
38400	217	38462	0.16%	236	37500	-2.34%
56000	229	55556	-0.79%	243	57692	3.02%
57600	230	57692	0.16%	243	57692	0.16%
115200	243	115385	0.16%	249	107143	-6.99%

13.6 Related Register

Name	SFR page	address	default values	Functions
PCON	0	0x87	0x00	Power Control Register
SCON0	0	0x98	0x00	Serial port control register
SBUF0	0	0x99	0x00	Serial port 0 data cache register
USART_SEL	0	0x9F	0x01	Timer1 clock source prescaler selection

14 Timer A/Timer B Module

Both Timer A and Timer B consist of a 16-bit timer/counter and three capture/comparators, enabling multiple capture/comparators, PWM outputs, and time intervals for counting trigger conditions. Timer A has multiple interrupt modes that trigger overflows and capture/comparators from the timer/counter. Characteristics of Timer A / Timer B includes：

- 16-bit timing/counter supporting 4 operating modes;
- Support for configuring the the clock source as the system clock 1 ~ 65535 frequency division .
- 2 or 3 configurable capture/comparators;
- Configurable PWM output;
- Asynchronous input sampling;
- Quickly capturing the interrupt source

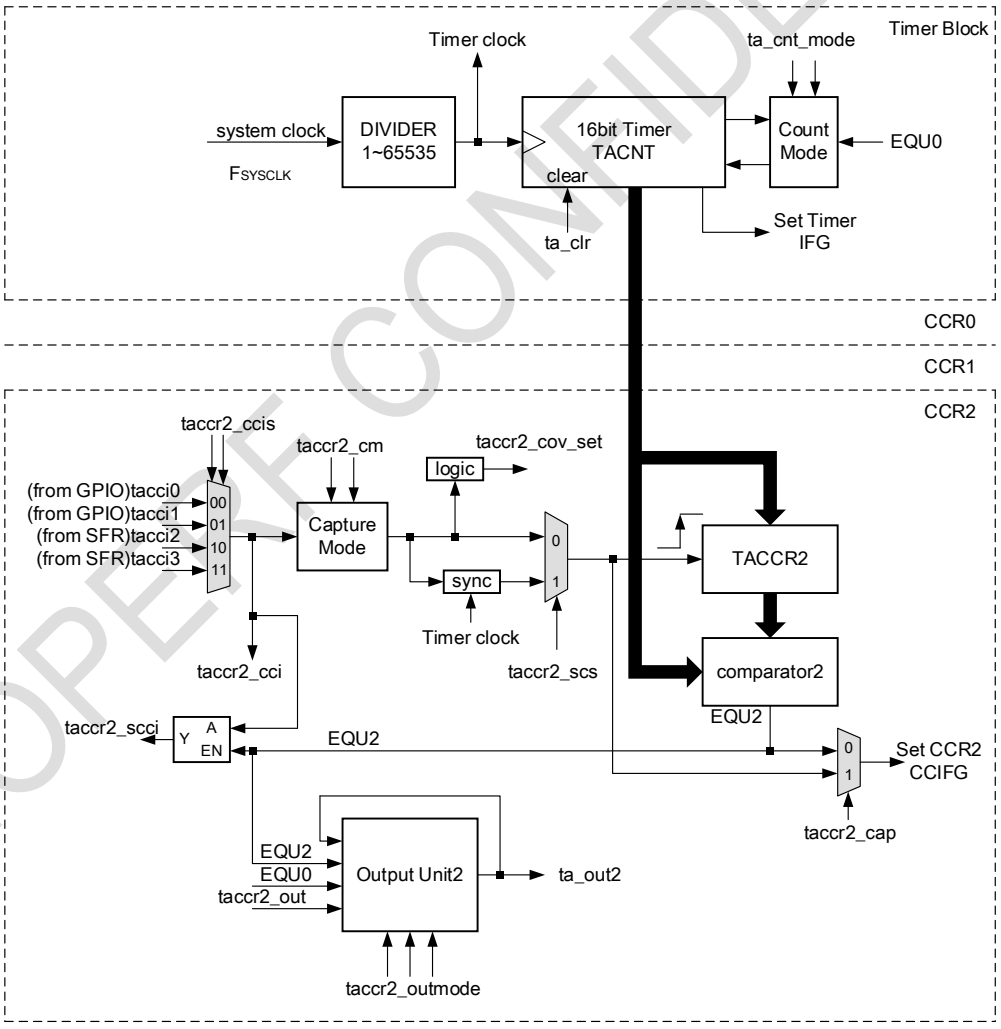

Chart 14-1. Diagram of Timer A Structure

Note: The structure of Timer A and Timer B are identical. The figure above shows the structure of Timer A.

14.1 Operation Method

The operation of Timer A/Timer B module is controlled by software. TACCR0TH/TBCCR0TH mentioned in this section is a configurable 16-bit count threshold. It is consists of register TACCR0TH_H/TBCCR0TH_H and TACCR0TH_L /TBCCR0TH_L.
Table 14-1 lists the counting modes of the 16-bit timing/counter (TACNT and TBCNT) according to different working modes.

Table 14-1. Work Modes of Timer

TA/TB_CNT_MODE	Working Mode	Description
00	Stop	TACNT/TBCNT does not work and stops counting , namely count stop mode
01	Up	TACNT/TBCNT increments repeatedly from 0 to the TACCR0TH/TBCCR0TH value, namely count up mode
10	Continuous	TACNT/TBCNT repeatedly increments from 0 to 0xFFFF, namely continuous counting mode
11	Up/ Down	TACNT/TBCNT repeatedly increments from 0 to TACCR0TH/TBCCR0TH,then decreasing to 0, in cycle, namely continuous counting mode.

As can be seen from the above table, Timer A or Timer B includes four working modes: Stop, Up, Continuous, Up/Down, and the working mode can be determined by configuring TA_CNT_MODE or TB_CNT_MODE.
When it is necessary to temporarily modify the operating mode of Timer A or Timer B (except the modification of interrupt enabling and interrupt flag), suggest stopping the TACNT or TBCNT counting first to avoid unpredictable misoperation.

In count up mode, TACNT or TBCNT generates an interrupt once it reaches the setting threshold TACCR0TH/TBCCR0TH.
In continuous count mode, TACNT or TBCNT will generate an interrupt once it reaches 0xFFFF.
In Up/Down mode, TACNT or TBCNT will generate interrupt once decrement to 0x0001.

Before starting Timer A or Timer B, configure TAC_L.TA_CNT_MODE or TBC_L.TB_CNT_MODE (non-stop mode), count threshold TACCR0TH or TBCCR0TH value (≠0), capture/compare parameters (see section 4.14.5). Then set TAC_H.TA_START or TBC_H.TB_START to zero first and then to 1 to trigger effectively.

During counting, user can set TACL.TA_CLR or TBCL.TB_CLR to 1 to clear most of the counter configuration. Take Timer A as an example, values that TA_CLR can clear include: Count clock source frequency division value TACLK_DIV, count value TACNT, counter working mode TA_CNT_MODE and count threshold TACCR0TH.

14.2 Up Mode

In Up mode, user can configure the count threshold TACCR0TH or TBCCR0TH to any value, and TACNT (or TBCNT) will increase from 0 to the threshold TACCR0TH (or TBCCR0TH) repeatedly, with the count period being TACCR0TH (or TBCCR0TH) +1. When the TACNT (or TBCNT) count reaches the threshold, it immediately returns to 0 to re-count.

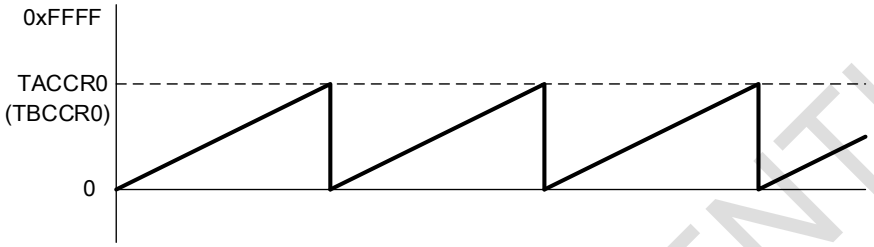

Chart 14-2. Schematic for Timer A/ Timer B Works in Up Mode

When TACNT (or TBCNT) counts to TACCR0TH or TBCCR0TH and overflows back to 0, the interrupt flag TA_CCR0_INT or TB_CCR0_INT is set, The interrupt flag TA_TMR_INT (or TB_TMR_INT) of Timer A (or Timer B) is set one beat later than TA_CCR0_INT. The following diagram shows the generation diagram of two different interrupts after the pre-split value TA_CLK_DIV is set to 3:

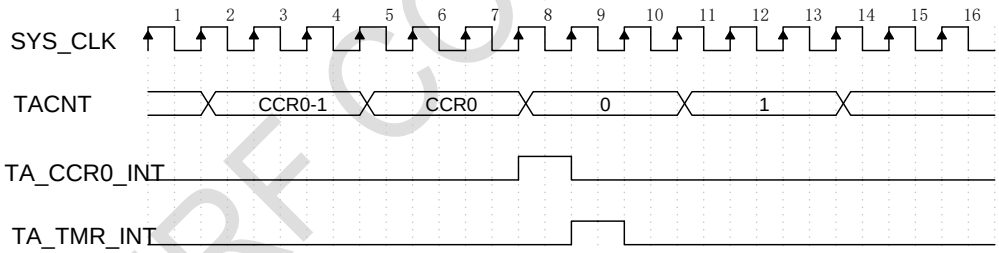

Chart 14-3. Schematic for Interrupt of Timer A Working in Up Mode

It is not recommended to modify the value of TACCR0TH (or TBCCR0TH) during the counting process of TACNT (or TBCNT). In case users have modifications forcedly, it will have the below 2 effect depending on different conditions.

1. *If the new TACCR0TH (or TBCCR0TH) modification value is greater than the previous one, or greater than the current TACNT (or TBCNT) count value, TACNT (or TBCNT) will continue counting to the new TACCR0TH (or TBCCR0TH) value and then return to 0 to recount.*
2. *If the new TACCR0TH (or TBCCR0TH) modification value is less than the previous TACNT (or TBCNT) count value, the TACNT（or TBCNT）will be immediately set 0, and it will continue counting to new TACCR0TH (or TBCCR0TH).*

14.3 Continuous Mode

In continuous mode, TACNT (or TBCNT) is increased from 0 to 0xFFFF repeatedly, reset and then counts from 0. In this mode, the 3 sets of capturer/comparator CCR0 ~ CCR2 have the same function and operate independently, which is different with the up mode. In Up mode, TACCR0TH (or TBCCR0TH) is the period value of the TACNT (or TBCNT) count.

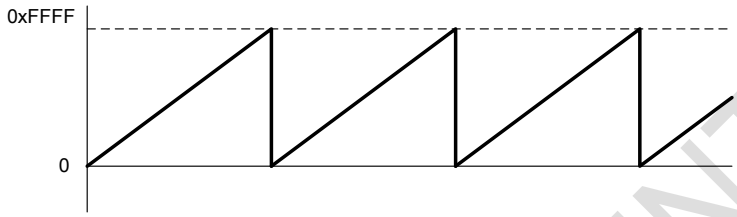

Chart 14-4. Schematic for Timer A Works in Continuous Mode

In the continuous mode, the capture/compare module can generate interrupts separately, as shown in the below figure.

- When TACNT (TBCNT) counts to the TACCR0TH+1 (TBCCR0TH+1) value, the interrupt flag TA_CCR0_INT (TB_CCR0_INT) of the comparator ta_ccr0 or tb_ccr0 is set.
- When TACNT (TBCNT) counts to 0xFFFF and then returns to 0 to re-count, the Timer interrupt flag TA_TMR_INT (TB_TMR_INT) is set.

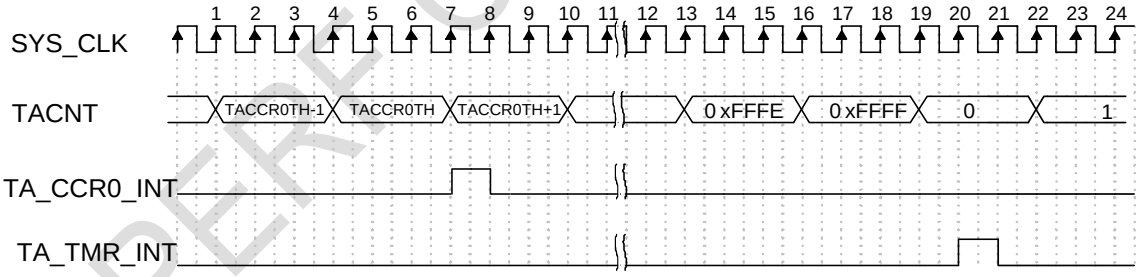

Chart 14-5. Schematic for Interrupt of Timer A Working in Continuous Mode

The Continuous mode can be used to generate independent time intervals and output frequencies by the above ways. In the case of comparator ccr0, Timer A will generate an interrupt when counting to TACCR0TH. After the software detects this interrupt in the register, it can configure the comparison threshold TACCR0TH to TACCR0TH + n, where n is the set period value and n<0XFFFF. Keep updating the value of TACCR0TH in cycles and interrupts of n period is generated.

Therefore, three sets of capture/comparators ta_ccr0~ ta_ccr2 (tb_ccr0~ tb_ccr2) are used to produce three sets of independent time interval and frequency outputs, as shown in the figure below, where TACCR0THa, TACCR0THb, TACCR0THc, and TACCR0THd are the values

calculated via the equation of TACCR0TH =TACCR0TH+ n, the same goes for ta_ccr1.

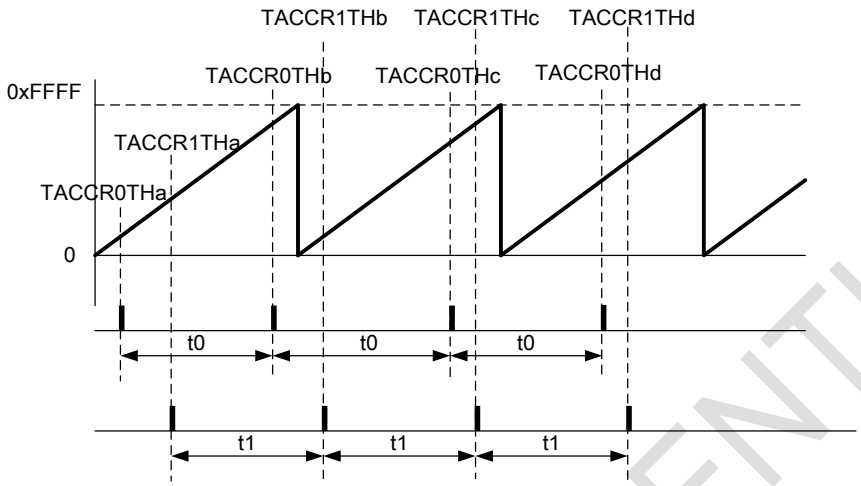

Chart 14-6. Schematic for Independent Operation of Each Capture/Compare of Timer A (Same for Timer B)

14.4 Up / Down Mode

In Up/Down mode, TACNT (TBCNT) increments repeatedly from 0 to TACCR0TH (TBCCR0TH) and then decrements to 0. One period is twice the TACCR0TH (TBCCR0TH) value.

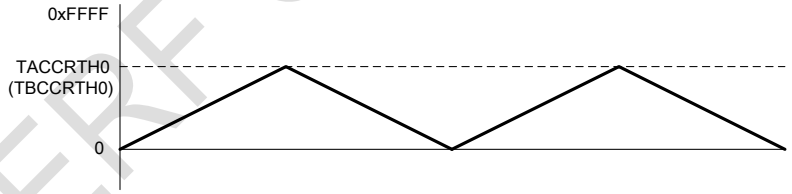

Chart 14-7. Schematic for Timer A / B Works in Up / Down Mode

TA_CCR0_INT (TB_CCR0_INT) interrupt of ta_ccr0 and tb_ccr0 and TA_TMR_INT (TB_TMR_INT) interrupt flags of TACNT (TBCNT) are distributed in the first and second half in one cycle. Similarly to Up mode, when TACNT (TBCNT) counts to the TACCR0TH (TBCCR0TH) value, interrupt flag TA_CCR0_INT (TB_CCR0_INT) of ta_ccr0 is set. When TACNT (TBCNT) counts to the threshold of ta_ccr0 and returns to 0 to count again, the Timer A (Timer B) interrupt flag TA_TMR_INT (TB_TMR_INT) is set.

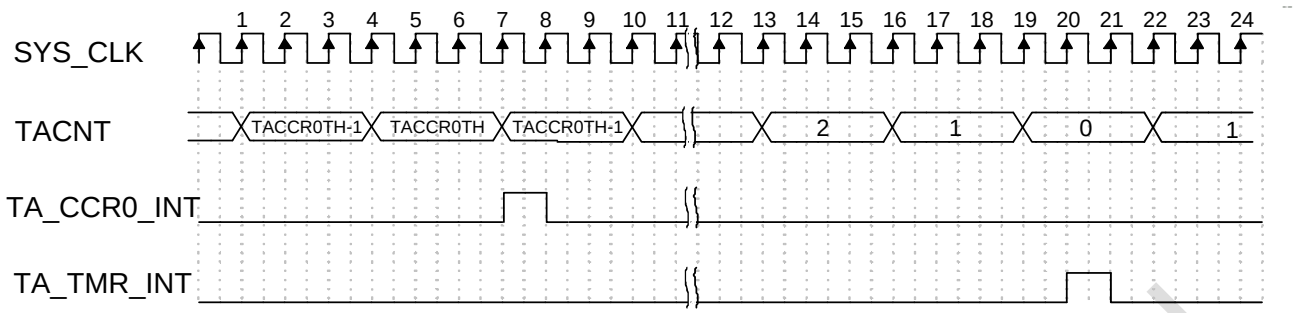

Chart 14-8. Schematic for Interrupt of Timer A Working in Up/ Down Mode

The up / down mode can support the application requiring Dead Time between two output signals. For example, 2 outputs that drive an H-bridge synchronously while cannot output a high level at the same time to avoid overload.

$$T_{Dead}=T_{Timer}x（TACCR1TH–TACCR2TH）$$

In above,

1. T_{Dead} refers to the duration of the dead time
2. T_{Timer} refers to the clock period of TACNT or TBCNT
3. TACCR1TH and TACCR2TH refer to the configuration values of the 2 sets of capturer/comparators.

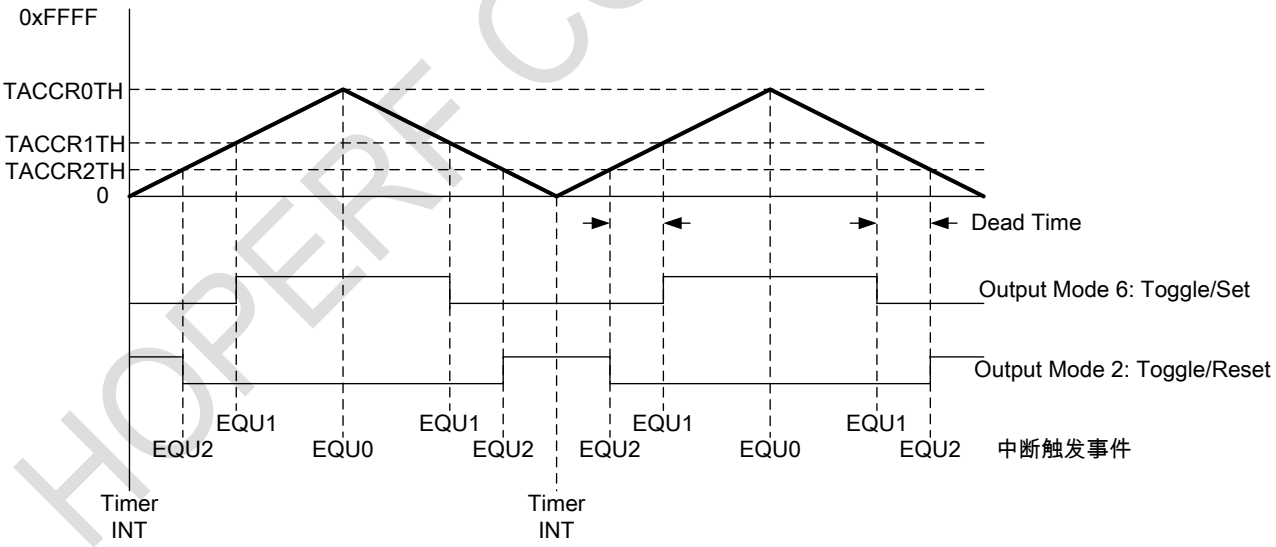

Chart 14-9. Schematic for Up / Down Mode with Dead Time Control

14.5 Capture/compare Module

Timer A / B contains 2 to 3 independent capture/compare modules for capturing TACNT (or TBCNT) data or generating time intervals. Note that, in Up and Up / Down modes, TACCR0 (TBCCR0) is used as a period register and cannot store captured values.

In Continuous mode，both ta_ccr0~ ta_ccr2（tb_ccr0~ tb_ccr2）can store captured values.

- **Capture Mode**

If TACCTL0_H.TA_CCR0_FUNC_MODE to TA_CCR2_FUNC_MODE is set to 1, the corresponding capture/comparison module enters the capture mode. Capture mode is used to record time related events, such as speed estimates or time measurements. There are 4 capture sources, of which TACCI0 and TACCI1 are from GPIOn (configurable selection, see Section 4.9 GPIO Module for details) and TAC_H.TA_CCI2_IN_SFR and TAC_H.TA_CCI3_IN_SFR are from internal SFR registers, accessible to software. By configuring TA_CCR0_SRC_SEL - TA_CCR1_SRC_SEL can select different capture sources (CCI0 - CCI3) for three capture channels. Timer B is exactly the same,the same hereinafter.
Configure TACCR0_CM ~ TACCR2_CM to select the capture mode of the corresponding capture/compare module as rising edge, falling edge or double edge trigger. After successful capture, the TACNT value will be stored in the TACCRn register of the corresponding capture/comparison module, and their interrupt flag TA_CCR0_INT~TA_CCR2_INT will be set at the same time.

Configure TACCR0_SCS ~ TACCR2_SCS to select whether to perform system clock synchronization on the capture source.

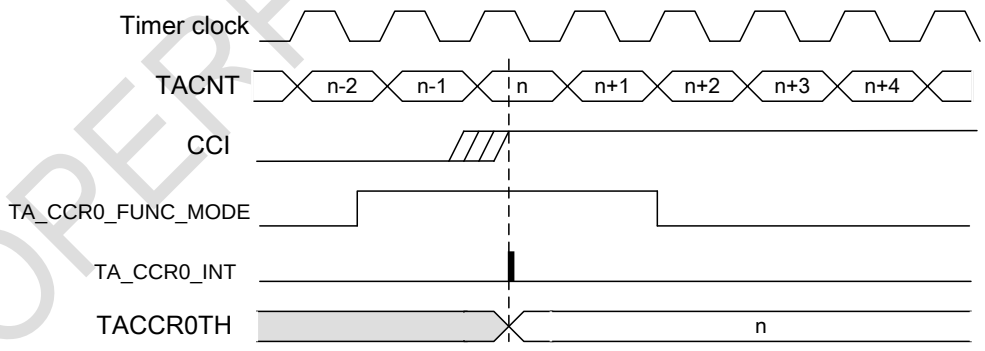

Chart 14-10. Schematic for Timer A Capture Mode

If the previous capture result has not been read, and the capture source is triggered again and generate capture overflow, the COV flag of the corresponding capture/comparison module will be set to 1, which will be re-captured after being cleared by the software.

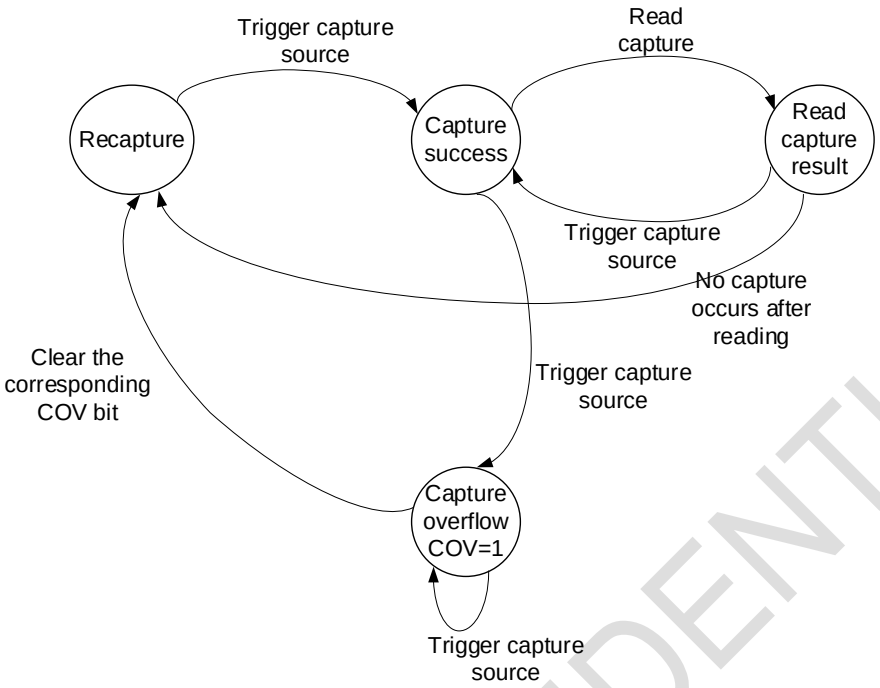

Chart 14-11. Schematic for Capture Mode State and Interrupt of Timer A

- **Compare mode**

When TA_CCR0_FUNC_MODE ~ TA_CCR2_FUNC_MODE is set to 0, the corresponding capture/compare module enters the compare mode. The compare mode is used to generate a PWM output signal or generate an interrupt at a specific time interval. When TACNT counts to TACCR0TH ~ TACCR2TH：

1) The corresponding interrupt flag TACCR0 /1/2 is set to 1.
2) The corresponding count is equal means that the signal of EQU0 ~ EQU2 is set to 1;
3) EQU0~EQU2 affects the output signal according to different output modes;
4) The capture source selected by each comparator is stored in the corresponding TA_CCR0_SRC ~ TA_CCR2_SRC register;

- **Output Units**

Each capture/comparison module contains an output unit to generate output signal such as PWM signal. Each output unit is based on EQU0 and EQU1/EQU2 signals and can be combined into eight output modes.

TA_CCR0_OUT_MODE ~ TA_CCR2_OUT_MODE is the output configuration register for the corresponding capture/compare module, where output modes of 2, 3, 6, and 7 do not apply to output the 0 unit because EQUx = EQU0. (EQUx means EQU1 and EQU2)

Table 14-2. Various Modes of Output Unit

OUTMODE	Mode	Description	REMARKS
000	OUTPUT	Direct mode, the output TA_OUTx is configured by register CCRx_OUT.	Applicable to the 3 capture/compare modules.
001	bits	When TACNT counts to TACCRxTH, the output TA_OUTx is set and the state retains until Timer A is reset or the output mode changes and affects the output..	Applicable to the 3 capture/compare modules.
010		When TACNT counts to TACCRxTH, the output TA_OUTx flips. When TAR counts to TACCR0TH, output TA_OUTx resets.	Only applicable to capture 1 and 2
011	Set bits / reset	When TACNT counts to TACCRxTH, the output TA_OUTx is set. When TAR counts to TACCR0TH, output TA_OUTx resets.	Only applicable to capture 1 and 2
100	Flip	When TACNT counts to TACCRxTH, the output TA_OUTx flip	Applicable to the 3 capture/compare modules.
101	Reset	When TACNT counts to TACCRxTH, the output TA_OUTx is reset and the state retains until the output mode changes and affects the output..	Applicable to the 3 capture/compare modules.
110	Flip / Reset	When TACNT counts to TACCRxTH, the output TA_OUTx flips. When TAR counts to TACCR0TH, output TA_OUTx is set..	Only applicable to capture 1 and 2
111	Resets/ bits	When TACNT counts to TACCRxTH, the output TA_OUTx reset. When TAR counts to TACCR0TH, output TA_OUTx is set..	Only applicable to capture 1 and 2

Notice:

1. *TACCRxTH refers to TACCR1TH or TACCR2TH；When it is in Timer B，it refers to TBCCR1TH or TBCCR2TH；*
2. *TA_OUTx refers to TA_OUT1 or TA_OUT2；When it is in Timer B，it refers to TB_OUT1 or TB_OUT2；*

14.6 Examples for Various Modes

- **Example for output in up mode**

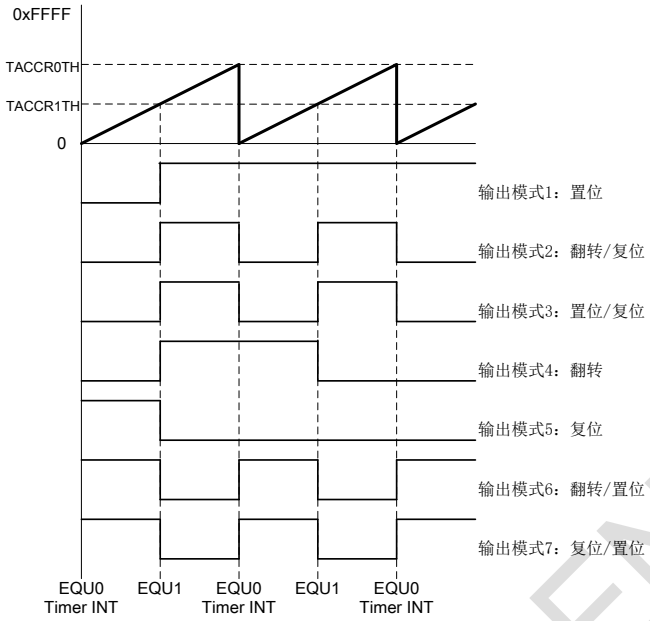

Chart 14-12. Schematic for Timer A Output in Up Mode

- **Example for output in continuous mode**

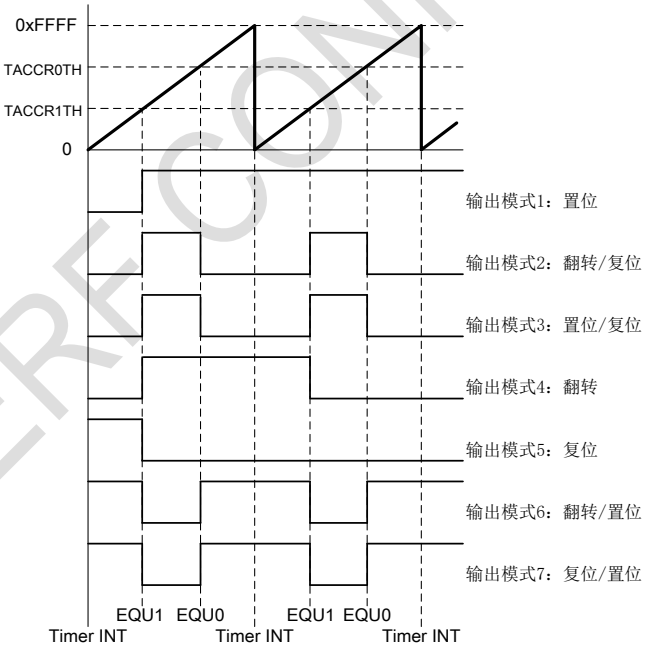

Chart 14-13. Schematic for Timer A Output in Continuous Mode

- **Example for output in up / down mode**

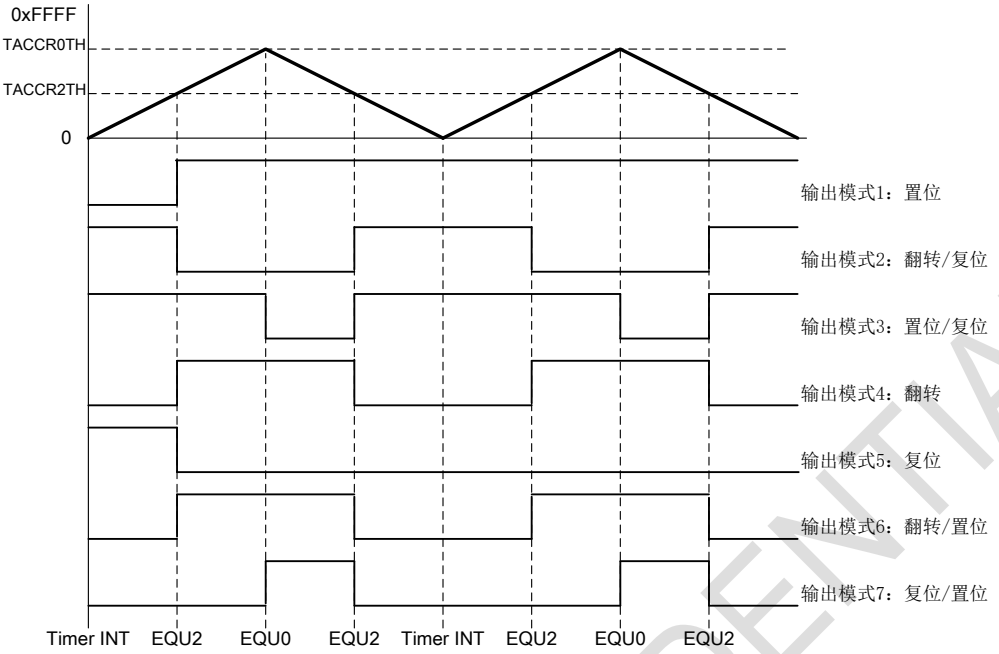

Chart 14-14. Schematic for Timer A Output Up / Down Mode

14.7 Related Register

Table 14-3. 16-Bit Timer A and Timer B Register Set List

Name	SFR page	address	default values	Functions
TACLK_DIV_H	0	0xB7	0x00	TimerA clock source frequency division register
TACLK_DIV_L	0	0xB9	0x00	
TAC_H	0	0xBA	0x00	Timer A Control Registers
TAC_L	0	0xBB	0x00	
TACCR0TH_H	0	0xBC	0x00	TimerA capture/compare module 0 register
TACCR0TH_L	0	0xBD	0x00	
TACCTL0_H	0	0xBE	0x00	TimerA capture/compare module 0 control register
TACCTL0_L	0	0xBF	0x00	
TACCR1TH_H	0	0xC0	0x00	TimerA capture/compare module 1 register
TACCR1TH_L	0	0xC1	0x00	
TACCTL1_H	0	0xC2	0x00	TimerA capture/compare module 1 control register
TACCTL1_L	0	0xC3	0x00	
TACCR2TH_H	0	0xC4	0x00	TimerA capture/compare module 2 register
TACCR2TH_L	0	0xC5	0x00	
TACCTL2_H	0	0xC6	0x00	TimerA capture/compare module 2 control register
TACCTL2_L	0	0xC7	0x00	
TACNT_H	0	0xC8	0x00	TimerA 16-bit real-time count register,

Name	SFR page	address	default values	Functions
TACNT_L	0	0xC9	0x00	
TBCLK_DIV_H	0	0xCA	0x00	TimerB clock source frequency division register
TBCLK_DIV_L	0	0xCB	0x00	
TBC_H	0	0xCC	0x00	TimerB control register
TBC_L	0	0xCD	0x00	
TBCCR0TH_H	0	0xCE	0x00	TimerB capture/compare module 0 register
TBCCR0TH_L	0	0xCF	0x00	
TBCCTL0_H	0	0xD1	0x00	TimerB capture/compare module 0 control register
TBCCTL0_L	0	0xD2	0x00	
TBCCR1TH_H	0	0xD3	0x00	TimerB capture/compare module 1 register
TBCCR1TH_L	0	0xD4	0x00	
TBCCTL1_H	0	0xD5	0x00	TimerB capture/compare module 1 control register
TBCCTL1_L	0	0xD6	0x00	
TBCCR2TH_H	0	0xD7	0x00	TimerB capture/compare module 2 register
TBCCR2TH_L	0	0xD8	0x00	
TBCCTL2_H	0	0xD9	0x00	TimerB capture/compare module 2 control register
TBCCTL2_L	0	0xDA	0x00	
TBCNT_H	0	0xDB	0x00	TimerB 16-bit real-time count register,
	0	0xDC	0x00	

15 Watchdog (WDT) Module

15.1 Basic Function

A hardware watchdog timer (WDT) module is integrated in the system to monitor the system running status. After the system starts running, users can start the watchdog timer to count. If the watchdog timer is not cleared within the configured time duration, the system will generate a whole reset. This reset is equivalent to restarting the system.

15.2 Related Register

Table 15-1. Watchdog Related Register List

Name	address	default values	Functions
AON_REG_03	0x03	0x00	Watchdog Control Register

16 Sleep TimerModule

16.1 Basic Function

CMT2187A provides a built-in Sleep Timer to satisfy the low-power sleep and periodically wake-up applications. Clock source is provided by the system auxiliary clock (LFOSC).

The Sleep Timer starts counting only in STOP mode and the system is woke up when the count overflows. After the system is woke up and in running state, the Sleep Timer stops timing until the system enters STOP mode.

16.2 LPOSC Calibration

The 32kHz LPOSC frequency will drift with the change of process, temperature and voltage, and the chip will perform an LPOSC calibration at factory and store the configured value inside the chip. During the working process, user can call the calibration module timely to calibrate, the specific operation can refer to the routine.

16.3 Related Register

Table 16-1. Sleep Timer Module Related Register Set List

Name	address	default values	Functions
AON_REG_01	0x01	0x00	Sleep timer control and configuration
AON_REG_02	0x01	0x00	Sleep timer control and configuration

Table 16-2. LFOSC Calibration and Output Related Register List

Name	SFR page	address	default values	Funtions
ANA_CTL_4	0	0xE2	0x04	LFOSC Calibration Control
ANA_CTL_8	0	0xE7	0x80	LFOSC Calibration Control

17 Low Voltage Reset (LVR)

Low voltage reset refers to the reset signal of the chip when the power supply voltage falls below the VLVR threshold voltage. The default reset release voltage of the CMT2187A is 1.8V when it is powered on for the first time. The subsequent reset release voltage and hysteresis detection function are enabled.

Table 18-1. Low Voltage Reset Configuration

Name	Address	Default Value	Function
AON_REG_06	0x06	0x00	LVR Configured register

18 Low Voltage Detection Module

18.1 Basic Function

CMT2187A is embedded with a low voltage monitoring module, which is used for detecting the power supply voltage before transmission. If the voltage is found to be insufficient, the transmission power will be compensated. The transmission power compensation is calculated automatically by the on-chip transmitter according to the current voltage detection value without user setting. User only needs to trigger the LBD detection once before transmitting.

Users can also use the LBD module to meet application needs at any time. User can first set the value of LBD_TH<3:0>, and then set LBD_START to 1 to trigger the LBD module to work. After the LBD module completes the work, it will automatically set LBD_DONE to 1, and obtain the value of LBD_CODE<3:0> that represents the current DVDD voltage. If LBD_CODE<3:0>≥LBD_TH<3:0>, then LBD_STATUS is equal to 1, otherwise it is equal to 0. When the user queries LBD_DONE bit 1, it can read LBD_STATUS and sdLBD_CODE<3:0>, and obtain the corresponding voltage value according to the following table.

Table 19-1. Voltage Thresholds of LBD Modules

LBD_CODE<3:0>	Voltage	LBD_CODE<3:0>	Voltage
0	1.45 V	8	3.05 V
1	1.65 V	9	3.25 V
2	1.85 V	10	3.45 V
3	2.05 V	11	3.65 V
4	2.25 V	12	3.85 V
5	2.45 V	13	4.05 V
6	2.65 V	14	4.25 V
7	2.85 V	15	4.45 V

It should be noted that this LBD module does not work after the system enters STOP mode, and is

unrelated to the real-time Supply Monitor function of the system. The former is used to detect a specific voltage value, and the latter is used to detect a voltage mutation to produce system reset.

18.2 Related Register

Table 19-2. Related Register of LBD Module

Name	SFR page	Address	Default Value	Function
LBD_CTL_0	0	0xFB	0x00	LBD Configure and control register
LBD_CTL_1	0	0xFC	0x00	LBD Configure and control register

19 Sub-1G Transmitting Module

19.1 Introduction

Sub-1G transmitter module is embedded in CMT2187A, adopting the phase locked loop (PLL) of fractional frequency division technology and using 26MHz XOSC, which realizes different transmission frequency points through different frequency division ratio set by software. The module supports OOK/(G)FSK modulation, which is based on the direct synthesis of RF frequency. The carrier frequency is generated by a low-noise fractional frequency synthesizer, and the modulation data is transmitted by an efficient power amplifier (PA). Its general block diagram is shown below.

The following is the schematic of the transmitter structure. The baseband signal is generated by the user program and processed by Gaussian transform, frequency control, AFC/DSM control, RAMP control and other modules successively. According to the configuration related to transmission, the frequency control value is transmitted to the PLL module, the power control value and the current control value are transmitted to the PA module, and complete PA correction process synchronously.

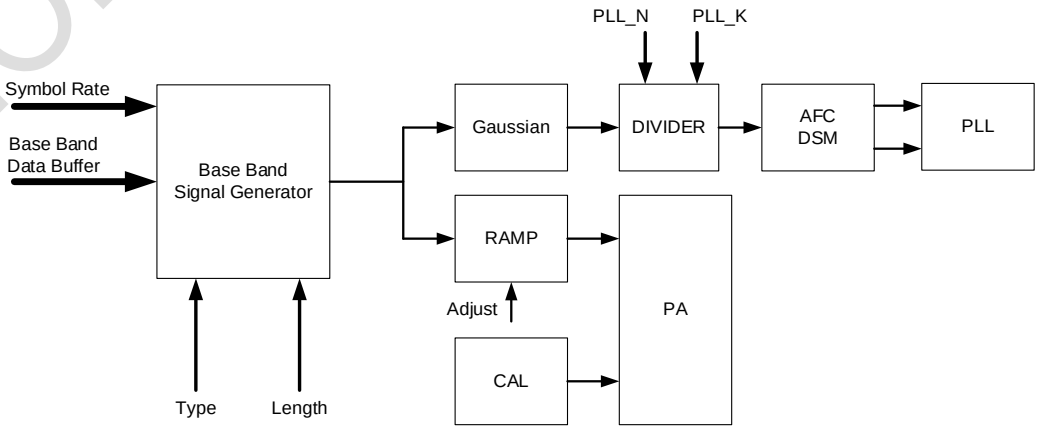

Chart 20-1. Block Diagram of CMT2187A UHF Transmitting

The locking time of PLL is about 150uS, and OOK is achieves by directly adjusting the output amplitude of PA. The CMT2187A can be configured in the frequency range of 210MHz-960mhz, while not includes frequency range from 480MHz-630MHz.

19.2 PA Output Method

CMT2187A adopts an efficient single-ended Class E PA structure, with transmitting power up to +13dBm PA has the feature of adjustable RAMP slope, as shown in the figure below. By using the TxSoC RFPDK tool software, after setting the target working parameters, the corresponding PA Ramping parameter will be automatically generated. The use of PA Ramping can effectively suppress the noise caused at transmitting process.

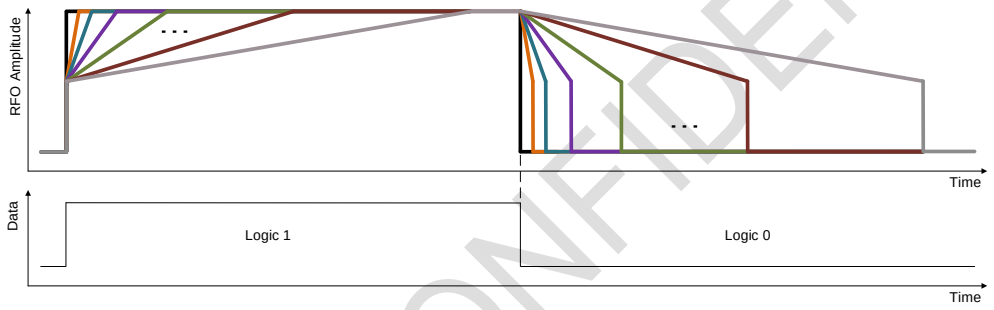

Chart 20-2. PA Ramping Sequence

19.3 Transmitting Process of Buffer Mode

The transmitting process is a process in which the transmitting module controls and simulates PLL and PA according to baseband signals. According to the baseband data source, it can be divided into:

- **Buffer Transmitting Mode**
- **Direct Transmitting Mode**

This section describes the transmitting process of Buffer mode. In Buffer mode, user needs to configure (prepare) the content, type, length and transmission data rate of the baseband signal before transmission (that is, the format and content of the packet to be sent), and then fill the transmission mode to the baseband cache. See the figure below for details.

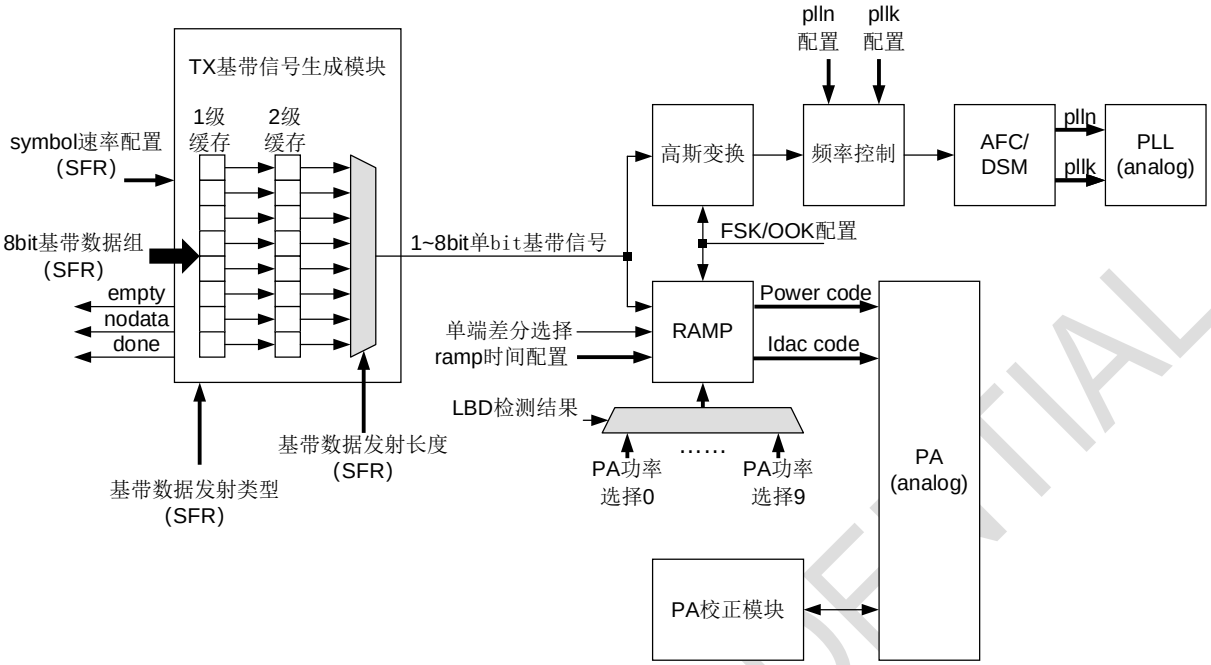

Chart 20-3. Structure Diagram for Transmitting Module

There are several important signals and registers associated with user adoption:

- The Symbol rate configuration (TX_DR register set), i.e., the relationship between parameter of each symbol transmiting time (TSYM_TIME) and the transmitting rate (BR) as well as crystal frequency (FXTAL, i.e. 26MHz) are as follows:

$$T_{SYM\_TIME} = \frac{BR \times 2^{23}}{F_{XTAL}}$$

- Baseband data serial register (TX_SYM_BYTE register) : that is, the cache register for transmitting data, which is the data entry of the transmission module, supporting 1 to 8 bits of the transmitting data to be filled, and the specific effective transmission length, is determined by "baseband data transmission length"; However, after completing the register data transmission, the empty flag is generated, and the software continues to fill the next set of baseband data to be transmitted until all the data complete transmitting.
- Transmission length of baseband data (TX_SYM_CTL register): it can configure and determine the effective transmission length of TX_SYM_BYTE with range from 1 - 8 bits. For example, if the baseband data transmission length is set to 4bits effective, only the lowest 4bits of TX_SYM_BYTE data are transmitted each time.

- Baseband data transmitting type: The working mode after transmitting the last bit of data (no further filling), which can be configured to: stop transmitting, cycle transmitting the last filling data group, transmitting normal 0 or transmitting normal 1.
- empty flag: TX baseband signal generation module includes 2 level cache, when the data of level 1 cache is stored to level 2, the empty flag takes effect, you can continue to fill the next group of data to be transmitted in level 1 cache;
- nodata flag: When the last bit of data is transmitted, the subsequent data group is no longer filled, and the nodata flag takes effect.
- done flag: Only the first baseband data transmission type is valid. When the last bit of data is transmitted and no more data is filled, the done flag takes effect.

Notes:

1. *As described above, the Buffer mode seems to involve many parameters, and user can refer to routines such as TX_PACKET_SAMPLE.*
2. *Most of the configuration parameters can be generated by TxSoC RFPDK tool software without manual calculation.*

19.4 Transmitting Process of Direct Mode

The direct mode is much simpler than the buffer mode and is enabled by configuring the TX_DIRECT_EN (located in the TX_SYM_CTL register). In this mode, the TX baseband signal generation module shown in Figure 20-3 does not work. Data "1" and "0" are transmitted directly by controlling the TX_SYM_BYTE register Bit0 At the same time, because the "TX baseband signal generation module" is disabled, the length of each symbol (data) transmission time (that is, the transmission rate) is completely controlled by the software of 8051.

Although the direct transmission mode is simple compared to the buffer transmission mode, there are advantages and disadvantages between the two, and the specific comparison is shown in the following table.

Table 20-1. Comparison of the CMT2187A Direct mode and Buffer mode

Contrasting	Buffer Mode	Direct Mode	Description
Accuracy of TX Rate	High	Low	Buffer mode is generated by TX baseband signal generation module rate clock, this clock source is from 26MHz crystal, so the rate accuracy is ppm level; The Direct mode is software-controlled and the clock source is from the internal 24MHz RC oscillator.

19.5 Related Register

Table 20-2. Register Group of Sub-1G Transmitting Module

Name	SFR page	Address	Default Value	Function
PLLN	0	0xE8	0x42	PLL N value configuration register
PLLK_H	0	0xE9	0x00	PLL K configuration register, K value refers to medium 8 bit
PLLK_M	0	0xEA	0xC5	
PLLK_L	0	0xEB	0xC1	PLL K configuration register, K value refers to low 8 bit
TX_DR_0	0	0xEC	0x00	Transmitting data rate configured register in high 8 bit
TX_DR_1	0	0xED	0x00	Transmitting data rate configured register in medium 8 bit
TX_DR_2	0	0xEE	0x01	Transmitting data rate configured register in low 8 bit
TX_SYM_BYTE	0	0xEF	0x00	Transmitting code register
TX_SYM_CTL	0	0xF2	0x00	Transmitting code control register
TX_PKT_CTL	0	0xF3	0x00	Transmitting code configured register
FREQ_DEV_H	0	0xF4	0x00	Frequency offset configured register in high 8 bit
FREQ_DEV_M	0	0xF5	0x00	Frequency offset configured register in medium 8 bit
FREQ_DEV_L	0	0xF7	0x00	Frequency offset configured register in low 8 bit
RAMP_STEP_H	0	0xF8	0x00	RAMP Configured register in high 8 bit
RAMP_STEP_L	0	0xF9	0x00	RAMP Configured register in low 8 bit
PA_IDAC_CODE	0	0xFA	0x00	Transmitting data rate configured register
PA_POWER_TH_9	1	0xC0	0x00	Transmitting data rate compensating register 9, 3.85V
PA_POWER_TH_8	1	0xC1	0x00	Transmitting data rate compensating register 8, 3.65V
PA_POWER_TH_7	1	0xC2	0x00	Transmitting data rate compensating register 7, 3.45V
PA_POWER_TH_6	1	0xC3	0x00	Transmitting data rate compensating register 6, 3.25V
PA_POWER_TH_5	1	0xC4	0x00	Transmitting data rate compensating register 5, 3.05V
PA_POWER_TH_4	1	0xC5	0x00	Transmitting data rate compensating register 4, 2.85V
PA_POWER_TH_3	1	0xC6	0x00	Transmitting data rate compensating register 3, 2.65V
PA_POWER_TH_2	1	0xC7	0x00	Transmitting data rate compensating register 2, 2.45V
PA_POWER_TH_1	1	0xC8	0x00	Transmitting data rate compensating register 1, 2.25V

Name	SFR page	Address	Default Value	Function
PA_POWER_TH_0	1	0xC9	0x00	Transmitting data rate compensating register 0, 2.05V

20 Package Outline

The CMT2187A package information is shown in the following:

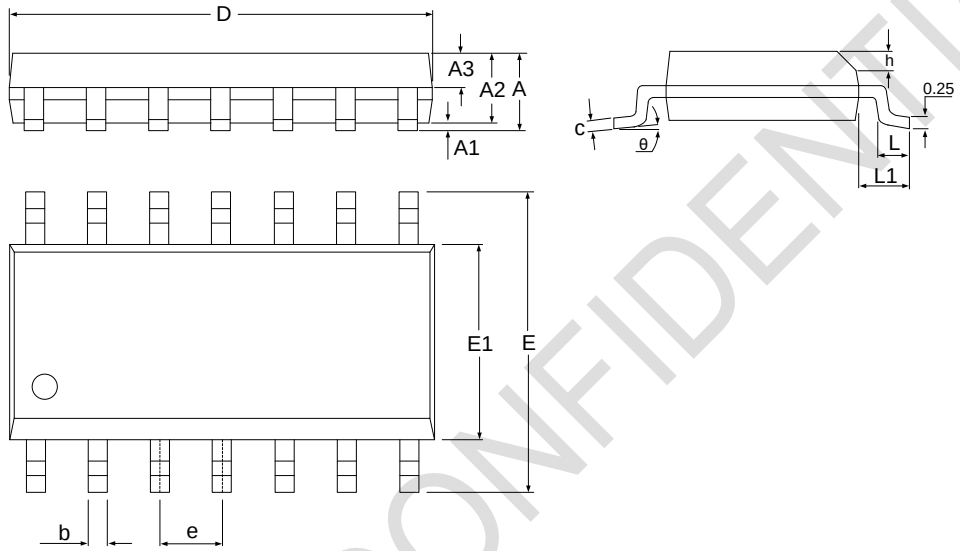

Chart 21-1. SOP14 Package

Table 21-1. SOP14 Package Scale

Symbol	Dimension (mm)		
	Minimum	Typical	Maximum
A	-	-	1.75
A1	0.05	-	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	-	0.48
c	0.21	-	0.26
D	8.45	8.65	8.85
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27 BSC		
L	0.25	-	0.50
L1	1.05 BSC		
θ	0	-	8°

21 Top Silk Printing

Chart 22-1. CMT2187A Top Silk Printing

Table 22-1. CMT2187A Top Silking Description

Silking method	Laser
Pin 1 Label	Circle Diameter = 1 millimeter
Font	0.6 mm, right alignment:
Font size	0.4 mm
First Line Printing	CMT2187A，represents for CMT2187A；
Second Line Printing	YYWW is the date number set by the packaging factory. YY represents the last 2 digits of the year and WW represents the work week ①②③④⑤⑥ is the internal tracking number

22 Other Related Document

Table 23-1. Other Related Document of CMT2187A

Document	Document Name	Description

Number		
	CMT2187A Develop environment construction	Construction of develop environments, quick user guide
	CMT2187A Register Description Manual	Detail information of the registers

23 Revise History

Table 24-1. CMT2187A Revise History

Version	Update chapter	Update records	Released Date
0.1	All	Initial version	2024/09/27
0.2	All	Review and delete the SDR related description	2025/3/31

24 Contacts

Shenzhen Hope Microelectronics Co., Ltd.

30th floor of 8th Building, C Zone, Vanke Cloud City, Xili Sub-district, Nanshan, Shenzhen, GD, P.R. China

Zip code: 518052

Phone: +86 - 755 - 82973805

Sales: sales@hoperf.com

Website: www.hoperf.com

Appendix A

Table 25-1. T8051XC3 Instruction Set

No.	Mnemonic	Description	Opcode	Bytes	Cycles
Arithmetic Operations					
1	ADD A, Rn	Add Register to Accumulator	0x28.. ..0x2F	1	3
2	ADD A, direct	Add direct data to Accumulator	0x25	2	3
3	ADD A, @Ri	Add indirect data to Accumulator	0x26 0x27	1	4
4	ADD A, #data	Add immediate data to Accumulator	0x24	2	2
5	ADDC A, Rn	Add Register to Accumulator with Carry	0x38.. ..0x3F	1	3
6	ADDC A, direct	Add direct data to Accumulator with Carry	0x35	2	3
7	ADDC A, @Ri	Add indirect data to Accumulator with Carry	0x36 0x37	1	4
8	ADDC A, #data	Add immediate data to Accumulator with Carry	0x34	2	2
9	SUBB A, Rn	Subtract Register from Accumulator with Borrow	0x98.. ..0x9F	1	3
10	SUBB A, direct	Subtract direct data from Accumulator with Borrow	0x95	2	3
11			0x96 0x97	1	4
12	SUBB A, #data	Subtract immediate data from Accumulator with Borrow	0x94	2	2
13	INC A	Increment Accumulator	0x04	1	1
14	INC Rn	Increment Register	0x08.. ..0x0F	1	3
15	INC direct	Increment direct data	0x05	2	3
16	INC @Ri	Increment indirect data	0x06 0x07	1	4
17	INC DPTR	Increment Data Pointer register	0xA3	1	2
18	DEC A	Decrement Accumulator	0x14	1	1
19	DEC Rn	Decrement Register	0x18.. ..0x1F	1	3
20	DEC direct	Decrement direct data	0x15	2	3
21	DEC @Ri	Decrement indirect data	0x16 0x17	1	4
22	MUL AB	Multiply A by B	0xA4	1	8
23	DIV AB	Divide A by B	0x84	1	8

No.	Mnemonic	Description	Opcode	Bytes	Cycles
24	DA A	Decimal Adjust	0xD4	1	1
Logical Operations					
25	ANL A, Rn	AND Accumulator with Register	0x58.. ..0x5F	1	3
26	ANL A, direct	AND Accumulator with direct data	0x55	2	3
27	ANL A, @Ri	AND Accumulator with indirect data	0x56 0x57	1	4
28	ANL A, #data	AND Accumulator with immediate data	0x54	2	2
29	ANL direct, A	AND direct data with Accumulator	0x52	2	3
30			0x53	3	3
31	CLR A	Clear Accumulator	0xE4	1	1
32	CPL A	Couple Accumulator	0xF4	1	1
33	ORL A, Rn	OR Accumulator with Register	0x48.. ..0x4F	1	3
34	ORL A, direct	OR Accumulator with direct data	0x45	2	3
35	ORL A, @Ri	OR Accumulator with indirect data	0x46 0x47	1	4
36	ORL A, #data	OR Accumulator with immediate data	0x44	2	2
37	ORL direct, A	OR direct data with Accumulator	0x42	2	3
38	ORL direct, #data	OR direct data with immediate data	0x43	3	3
39			0x23	1	1
40	RLC A	Rotate Accumulator left through Carry	0x33	1	1
41	RR A	Rotate Accumulator right	0x03	1	1
42	RRC A	Rotate Accumulator right through Carry	0x13	1	1
43	SWAP A	Swap Accumulator	0xC4	1	1
44	XRL A, Rn	Exclusive-OR Accumulator with Register	0x68.. ..0x6F	1	3
45	XRL A, direct	Exclusive-OR Accumulator with direct data	0x65	2	3
46	XRL A, @Ri	Exclusive-OR Accumulator with indirect data	0x66.. ..0x67	1	4
47	XRL A, #data	Exclusive-OR Accumulator with immediate data	0x64	2	3
48	XRL direct, A	Exclusive-OR direct data with Accumulator	0x62	2	3
49	XRL direct, #data	Exclusive-OR direct data with immediate data	0x63	3	3
Data Transfer Operations					
50	MOV A, Rn r	Move Register to Accumulator	0xE8.. ..0xEF	1	2
51	MOV A, direct	Move direct data to Accumulator	0xE5	2	2

No.	Mnemonic	Description	Opcode	Bytes	Cycles
52	MOV A, @Ri	Move indirect data to Accumulator	0xE6 0xE7	1	3
53	MOV A, #data	Move immediate data to Accumulator	0x74	2	2
54	MOV Rn, A	Move Accumulator to Register	0xF8.. ..0xFF	1	1
55	MOV Rn, direct	Move direct data to Register	0xA8.. ..0xAF	2	2
56	MOV Rn, #data	Move immediate data to Register	0x78.. ..0x7F	2	2
57	MOV direct, A	Move Accumulator to direct	0xF5	2	2
58	MOV direct, Rn	Move Register to direct	0x88.. ..0x8F	2	3
59	MOV direct, direct	Move direct data to direct	0x85	3	3
60	MOV direct, @Ri	Move indirect data to direct	0x86 0x87	2	4
61	MOV direct, #data	Move immediate data to direct	0x75	3	3
62	MOV @Ri, A	Move Accumulator to indirect	0xF6 0xF7	1	2
63	MOV @Ri, direct	Move direct data to indirect	0xA6 0xA7	2	3
64	MOV @Ri, #data	Move immediate data to indirect	0x76 0x77	2	3
65	MOV DPTR, #data16	Move 16-bit immediate data to Data Pointer	0x90	3	3
66	MOVC A, @A+DPTR	Move Code relative to DPTR to Accumulator	0x93	1	4
67	MOVC A, @A+PC	Move Code relative to PC to Accumulator	0x83	1	4
68	MOVX A, @Ri	Move XDATA (by Register) to Accumulator	0xE2 0xE3	1	4
69	MOVX A, @DPTR	Move XDATA (by DPTR) to Accumulator	0xE0	1	3
70	MOVX @Ri, A	Move Accumulator to XDATA (by Register)	0xF2 0xF3	1	4
71			0xF0	1	3
72	PUSH direct	Push direct data onto stack	0xC0	2	2
73	POP direct	Pop direct from stack	0xD0	2	3
74	XCH A, Rn	Exchange Accumulator with Register	0xC8.. ..0xCF	1	2
75	XCH A, direct	Exchange Accumulator with direct	0xC5	2	2
76	XCH A, @Ri	Exchange Accumulator with indirect	0xC6 0xC7	1	3

No.	Mnemonic	Description	Opcode	Bytes	Cycles
77	XCHD A, @Ri	Exchange Accumulator nibble with indirect	0xD6 0xD7	1	3
Boolean (Bit-Wise) Operations					
78	ANL C, bit	AND Carry with direct bit	0x82	2	2
79	ANL C, /bit	AND Carry with direct bit inverted	0xB0	2	2
80	CLR C	Clear Carry	0xC3	1	1
81	CLR bit	Clear direct bit	0xC2	2	3
82	CPL C	Couple Carry	0xB3	1	1
83	CPL bit	Couple direct bit	0xB2	2	3
84	MOV C, bit	Move direct bit to Carry	0xA2	2	2
85	MOV bit, C	Move Carry to direct bit	0x92	2	3
86	ORL C, bit	OR Carry with direct bit	0x72	2	2
87	ORL C, /bit	OR Carry with direct bit inverted	0xA0	2	2
88	SETB C	Set Carry	0xD3	1	1
89	SETB bit	Set direct bit	0xD2	2	3
Program Branch Operation					
90	ACALL addr11	Absolute call	0baaa10001	2	3
91	AJMP addr11	Absolute jump	0baaa00001	2	3
92	CJNE A, direct, rel	Compare Accumulator to direct data and jump if not equal	0xB5	3	4
93	CJNE A, #data, rel	Compare Accumulator to immediate data and jump if not equal	0xB4	3	3
94	CJNE Rn, #data, rel	Compare Register to immediate data and jump if not equal	0xB8.. ..0xBF	3	4
95	CJNE @Ri, #data, rel	Compare indirect to immediate data and jump if not equal	0xB6 0xB7	3	5
96	DJNZ Rn, rel	Decrement Register and jump if not zero	0xD8.. ..0xDF	2	3
97	DJNZ, direct, rel	Decrement direct and jump if not zero	0xD5	3	4
98	JB bit, rel	Jump if direct bit is set	0x20	3	3
99	JBC bit, rel	Jump if direct bit is set and clear it	0x10	3	3
100	JC rel	Jump if Carry is set	0x40	2	3
101	JZ rel	Jump if Accumulator is zero	0x60	2	3
102	JMP @A+DPTR	Jump relatively to DPTR	0x73	1	2
103	JNC rel	Jump if Carry is cleared	0x50	2	3
104	JNB bit, rel	Jump if direct bit is cleared	0x30	3	3
105	JNZ rel	Jump if Accumulator is not zero	0x70	2	3
106	LCALL addr16	Long call	0x12	3	4
107	LJMP addr16	Long jump	0x02	3	3

No.	Mnemonic	Description	Opcode	Bytes	Cycles
108	RET	Return from subroutine	0x22	1	3
109	RETI	Return from interrupt	0x32	1	3
110	SJMP rel	Short jump	0x80	2	3
111	NOP	No operation	0x00	1	1