

HT8550

用户手册

炬泉光电科技（上海）股份有限公司

目录

1	芯片概况	9
1.1	简介	9
1.2	基本特性	9
1.3	应用范围	10
1.4	系统框图	10
1.5	缩略语	11
2	引脚说明	12
2.1	引脚框图	12
2.2	引脚功能	12
3	电气规格	14
3.1	极限参数	14
3.2	电气特性	14
4	电源与时钟	15
4.1	电源系统	15
4.2	低功耗	15
4.2.1	门控时钟	15
4.2.2	时钟调整	15
4.2.3	MCU IDLE模式	15
4.2.4	MCU STOP模式	16
4.2.5	特殊功能寄存器	16
4.3	时钟系统	17
4.3.1	PLL时钟	17
4.3.2	时钟源选择和时钟分频	17
4.3.3	特殊功能寄存器	18
4.4	复位系统	21
4.4.1	上电/掉电复位	21
4.4.2	外部引脚复位	22
4.4.3	看门狗复位	22
4.4.4	软件复位	22
4.4.5	特殊功能寄存器	22
5	控制单元(MCU)	23
5.1	MCU架构	23
5.2	存储单元	23
5.2.1	概述	23
5.2.2	片内资源访问	24
5.2.3	片外资源访问	26
5.3	DMA	26
5.3.1	概况	26
5.3.2	DMA 操作流程	26
5.3.3	特殊功能寄存器	26
5.4	中断系统	32
5.4.1	概述	32
5.4.2	基于GPIO的中断	33
5.4.3	DMA中断	33
5.4.4	RTC中断	33
5.4.5	QMD中断	33
5.4.6	CC中断	33
5.4.7	掉电检测中断	33
5.4.8	中断列表	33
5.4.9	中断优先级	34
5.4.10	中断处理	35

5.4.11	特殊功能寄存器	35
5.5	定时器	40
5.5.1	概述	40
5.5.2	定时器 0 详细说明	40
5.5.3	定时器 1 详细说明	41
5.5.4	定时器 2 详细说明	42
5.5.5	特殊功能寄存器	42
5.6	GPIO	46
5.6.1	特殊功能寄存器	48
6	其他外设	49
6.1	FLASH控制器	50
6.1.1	Flash 控制器简介	50
6.1.2	FLASH作为程序扩展	50
6.1.3	特殊功能寄存器	50
6.2	SPI接口	53
6.2.1	概述	53
6.2.2	主机模式	54
6.2.3	从机模式	54
6.2.4	特殊功能寄存器	54
6.3	I2C接口	56
6.3.1	概述	56
6.3.2	特殊功能寄存器	57
6.4	UART接口	68
6.4.1	概述	68
6.4.2	红外调制	68
6.4.3	UART0 接口	70
6.4.4	UART1 接口	72
6.4.5	特殊功能寄存器	73
6.5	RTC单元	76
6.5.1	概况	76
6.5.2	RTC开启和停止	77
6.5.3	RTC寄存器读写	77
6.5.4	RTC报警	77
6.5.5	特殊功能寄存器	78
6.6	看门狗定时器	80
6.6.1	概述	80
6.6.2	看门狗使能	80
6.6.3	看门狗计数器清零	81
6.6.4	特殊功能寄存器	81
6.7	JTAG调试	81
7	AFE相关寄存器配置	81
7.1	寄存器描述	81
7.1.1	ANALOG_CFG1	82
7.1.2	ANALOG_CFG2	82
7.1.3	VCC_LOSS	82
7.1.4	AGC_LOOP_CFG1	82
7.1.5	AGC_LOOP_CFG2	83
7.1.6	AGC_LOOP_CFG3	83
7.1.7	AGC_LOOP_STATUS1	83
7.1.8	AGC_LOOP_STATUS2	83
7.1.9	ALC_CONTROL	84
7.2	ALC功能描述	84
7.3	掉电检测	84
7.4	AGC的配置	85

8	MCU与PHY交互	85
8.1	概述	85
8.2	PHY功能简介	85
8.3	参数设置	86
9	芯片封装尺寸	86
10	附录	86
10.1	寄存器列表	86

List of Tables

表 1 缩略语	11
表 2 引脚概述	14
表 3 极限参数	14
表 4 电气特性	15
表 5 电源控制寄存器列表	16
表 6 电源控制寄存器(PCON 0x87H)	16
表 7 时钟模块表格	18
表 8 时钟相关寄存器列表	19
表 9 ADC和QMD时钟分频控制寄存器(ADC_QMD_CLK_CTRL 0xACh)	19
表 10 外部Flash时钟分频控制寄存器(FLASH_CLK_DIV 0xD6H)	19
表 11 时钟开启控制寄存器(CLK_ON 0xE5H)	20
表 12 时钟开启控制寄存器(CLK_OFF 0xE6H)	20
表 13 MAC时钟分频控制寄存器(MAC_CLK_DIV 0xF1H)	21
表 14 时钟控制寄存器(CLK_CTRL 0xF2H)	21
表 15 复位寄存器列表	22
表 16 复位设置寄存器(RST_SET 0xE7H)	23
表 17 复位撤销寄存器(RST_CLR 0xE8H)	23
表 18 内部数据存储通用工作寄存器区	25
表 19 DMA寄存器列表	27
表 20 DMASEL寄存器描述(DMASEL 0xA1H)	27
表 21 DMAS0 寄存器描述(DMAS0 0xA2H)	27
表 22 DMAS1 寄存器描述(DMAS1 0xA3H)	28
表 23 DMAS2 寄存器描述(DMAS2 0xA4H)	28
表 24 DMAS3 寄存器描述(DMAS3 0xA5H)	28
表 25 DMAT0 寄存器描述(DMAT0 0xA6H)	28
表 26 DMAT1 寄存器描述(DMAT1 0xA7H)	28
表 27 DMAT2 寄存器描述(DMAT2 0xB1H)	28
表 28 DMAT3 寄存器描述(DMAT3 0xB2H)	28
表 29 DMAC0 寄存器描述(DMAC0 0xB3H)	29
表 30 DMAC1 寄存器描述(DMAC1 0xB4H)	29
表 31 DMACSR寄存器描述(DMACSR 0xB5H)	30
表 32 DMATC寄存器描述(DMATC 0xB6H)	31
表 33 DMASC寄存器描述(DMASC 0xB7H)	31
表 34 MISC_PARAM寄存器描述(MISC_PARAM 0xA1H)	31
表 35 MISC_ADDR寄存器描述(MISC_ADDR 0xA2H)	32
表 36 CRC和扰码的表格	32
表 37 HT8550 中断系统	34
表 38 中断优先级	34
表 39 中断优先级组成员	35
表 40 中断寄存器列表	35
表 41 IEN2 寄存器描述(IEN2 0x9AH)	35
表 42 IEN0 寄存器描述(IEN0 0xA8H)	36
表 43 IEN1 寄存器描述(IEN1 0xB8H)	37
表 44 IRCON寄存器描述(IRCON 0xC0H)	38
表 45 MAC_INTR_MASK寄存器描述(MAC_INTR_MASK 0xFAH)	39
表 46 GPIO_INTR_FLAGS寄存器描述(GPIO_INTR_FLAGS 0xFBH)	39
表 47 PHY_INTR_FLAGS寄存器描述(PHY_INTR_FLAGS 0xFCH)	39
表 48 定时器 0 和定时器 1,定时器 2 寄存器列表	42
表 49 定时器 0、定时器 1 控制寄存器 (TCON 0x88H)	43
表 50 定时器 0、定时器 1 模式寄存器(TMOD 0x89H)	44
表 51 定时器 1 状态寄存器(TH1 0x8DH)	44

表 52	定时器 1 状态寄存器(TL1 0x8BH)	44
表 53	定时器 0 状态寄存器(TH0 0x8CH)	44
表 54	定时器 0 状态寄存器(TL0 0x8AH)	45
表 55	定时器 2 控制寄存器 (T2CON 0xC8H)	45
表 56	定时器 2 状态寄存器(TH2 0xCDH)	45
表 57	定时器 2 状态寄存器(TL2 0xCCH)	46
表 58	GPIO定义	48
表 59	GPIO寄存器列表	48
表 60	GPIO中断极性选择寄存器(GPIO_INTR_POLARITY 0xF9H)	49
表 61	GPIO port选择寄存器(GPIO_MODE_SEL 0xFEH)	49
表 62	GPIO 模式选择寄存器(GPIO_CTRL 0xFFH)	49
表 63	FLASH控制器寄存器列表	51
表 64	Flash控制寄存器(FLASH_CTRL 0x8FH)	51
表 65	Flash数据寄存器(FLASH_DATA 0x91H)	51
表 66	Flash状态寄存器(FLASH_STATUS 0x96H)	52
表 67	Flash复制寄存器(FLASH_COPY 0xABH)	52
表 68	Flash页复制寄存器(FLASH_COPY_PAGES 0xBEH)	52
表 69	Flash异或寄存器(Flash Memory XOR 0xD5H)	53
表 70	Flash存储编程锁定寄存器(Program Memory Lock 0xD7H)	53
表 71	SPI相关寄存器列表	54
表 72	SPI控制寄存器(SPSTA 0xE1H)	55
表 73	SPI状态寄存器(SPCON 0xE2H)	56
表 74	SPI数据寄存器(SPDAT 0xE3H)	56
表 75	SPI从机选择寄存器(SPSSN 0xE4H)	56
表 76	I2C 时序特征	57
表 77	I2C寄存器列表	58
表 78	I2C数据寄存器(I2CDAT 0xDAH)	58
表 79	I2C地址寄存器(I2CADR 0xDBH)	58
表 80	I2C控制寄存器(I2CCON 0xDCH)	59
表 81	I2C状态寄存器(I2CSTA 0xDDH)	59
表 82	I2C Status in Master Transmitter Mode	61
表 83	I2C Status in Master Receiver Mode	62
表 84	I2C Status in slave Receiver Mode	65
表 85	I2C Status in slave Transmitter Mode	67
表 86	I2C Status - miscellaneous states	68
表 87	红外相关寄存器	69
表 88	串口红外控制寄存器(UART_IR_CTRL 0xBCH)	69
表 89	串口红外频分寄存器(UART_IR_DIV 0xBDH)	70
表 90	UART0 常用波特率	70
表 91	UART1 常用波特率	72
表 92	UART寄存器列表	74
表 93	串口 0 控制寄存器(S0CON 0x98H)	74
表 94	串口 0 缓冲寄存器(S0BUF 0x99H)	75
表 95	串口 1 控制寄存器(S1CON 0x9BH)	75
表 96	串口 1 缓冲寄存器(S1BUF 0x9CH)	75
表 97	串口 1 波特率发生器的重载值低位(S1RELL 0x9DH)	76
表 98	串口 0 波特率发生器的重载值低位(S0RELL 0xAAH)	76
表 99	串口 0 波特率发生器的重载值高位(S0RELH 0xBAH)	76
表 100	串口 1 波特率发生器的重载值高位(S1RELH 0xBBH)	76
表 101	AD 控制寄存器(ADCON 0xD8H)	76
表 102	RTC寄存器列表	78
表 103	RTC选择寄存器(RTCSEL 0xCEH)	78

表 104	RTC数据寄存器(RTCDAT 0xCFH)	79
表 105	RTC中断使能寄存器(IEN4 0xD1H).....	79
表 106	RTCC RTC时钟控制寄存器(RTCSEL=0x4)	80
表 107	看门狗寄存器列表	81
表 108	看门狗加载寄存器(WDTREL 0x86H).....	81
表 109	模拟前端寄存器列表	82
表 110	ANALOG_CFG1 寄存器(ANALOG_CFG1 0xF3H).....	82
表 111	ANALOG_CFG2 寄存器(ANALOG_CFG2 0xC9H).....	82
表 112	VCC_LOSS寄存器(VCC_LOSS 0xBFH)	82
表 113	AGC_LOOP_CFG1 寄存器(AGC_LOOP_CFG1 0xF4H)	83
表 114	AGC_LOOP_CFG2 寄存器(AGC_LOOP_CFG2 0xF7H)	83
表 115	AGC_LOOP_CFG3 寄存器(AGC_LOOP_CFG3 0xF8H)	83
表 116	AGC_LOOP_STATUS1 寄存器(AGC_LOOP_STATUS1 0xFDH)	83
表 117	AGC_LOOP_STATUS2 寄存器(AGC_LOOP_STATUS2 0xD4H)	84
表 118	ALC_CONTROL寄存器(ALC_CONTROL 0xAFH).....	84

List of Figures

图 1 HT8550 系统框图.....	10
图 2 引脚分布图.....	12
图 3 存储器空间（片内MCU访问）.....	24
图 4 内部数据存储器功能结构.....	25
图 5 SPI数据输出帧格式.....	54
图 6 I2C 输出时序.....	57
图 7 I2C 输入时序.....	57
图 8 I2C 协议.....	57
图 9 串口红外调制.....	69
图 10 UART0 模式 0 的帧格式.....	71
图 11 UART0 模式 1 发送帧格式.....	71
图 12 UART0 模式 2 帧格式.....	71
图 13 RTC框图.....	77
图 14 ALC框图.....	84
图 15 掉电检测功能示意图.....	85
图 16 交互帧结构图.....	85
图 17 LQFP48 package information(单位 mm).....	86

1 芯片概况

1.1 简介

HT8550是一款高性能电力线载波通信芯片，采用先进的数模混合设计技术与工艺，提供低功耗、高灵敏度、高抗干扰能力的电力线数据通信，可实现各种类型的数据传输及远程抄表应用。HT8550是高集成度SOC芯片，它将模拟前端、数字调制解调、基带数据处理和8051 MCU以及FLASH存储器等模块完全在单芯片上实现。

HT8550支持多种模式及双载波发送与接收，包括通信的载波频率、双载波间隔，传输速率以及调制方式均可灵活配置。HT8550内部集成12bit高精度ADC、45dB动态范围信号放大器、自动增益控制环路，使模拟信号接收处理得到进一步优化；内置10比特DAC及低通滤波器，使得发送时产生的带外干扰得以抑制，HT8550支持时域分集接收技术可有效提高对抗电力线脉冲干扰能力；支持双载波频域分集模式可极大增强对抗频域窄带干扰性能。

HT8550采用DPSK调制方式、高效可靠的前向纠错技术、可灵活配置的传输模式，使其可以在各种不同噪声的电力线环境下实现自适应可靠通信。HT8550片内集成8051内核，支持多种外围接口，并能够完成MAC层及以上协议层所需各种功能及应用。

1.2 基本特性

- 工作电压：5V
- 采用0.162um CMOS工艺
- 片内集成8051内核，兼容8051指令集和总线结构
- 片内集成128KB程序存储内存以及8KB数据存储内存
- 调制方式：DBPSK，DQPSK，D8PSK
- 片内集成12位ADC和高动态范围自动增益控制电路
- 片内集成45dB可变增益放大器
- 片内集成10位DAC
- 可编程载波：3~500kHz
- 接收灵敏度优于0dBuV
- 符号传输速率可调，最高可达41.67Kbaud，净数据传输率最高可达55Kbps
- 推荐符号速率15.625Kbaud，净数据传输率2.524Kbps
- 支持长包传输：255字节
- 支持时域重叠交织及分集接收技术支持频域分集接收技术：双载波模式
- 可编程卷积编码效率：1/2，3/4
- 支持芯片工作状态监控：接收信号强度指示，接收信噪比
- 可动态配置传输模式
- 片内集成硬件看门狗电路

- 内置低功耗RTC
- 支持UART、SPI、I²C 等接口
- 支持多种DMA方式
- 支持ALC功能

1.3 应用范围

- 远程自动抄表
- 智能家居
- 路灯控制
- 工控自动化
- 智能楼宇控制
- 远程监控

1.4 系统框图

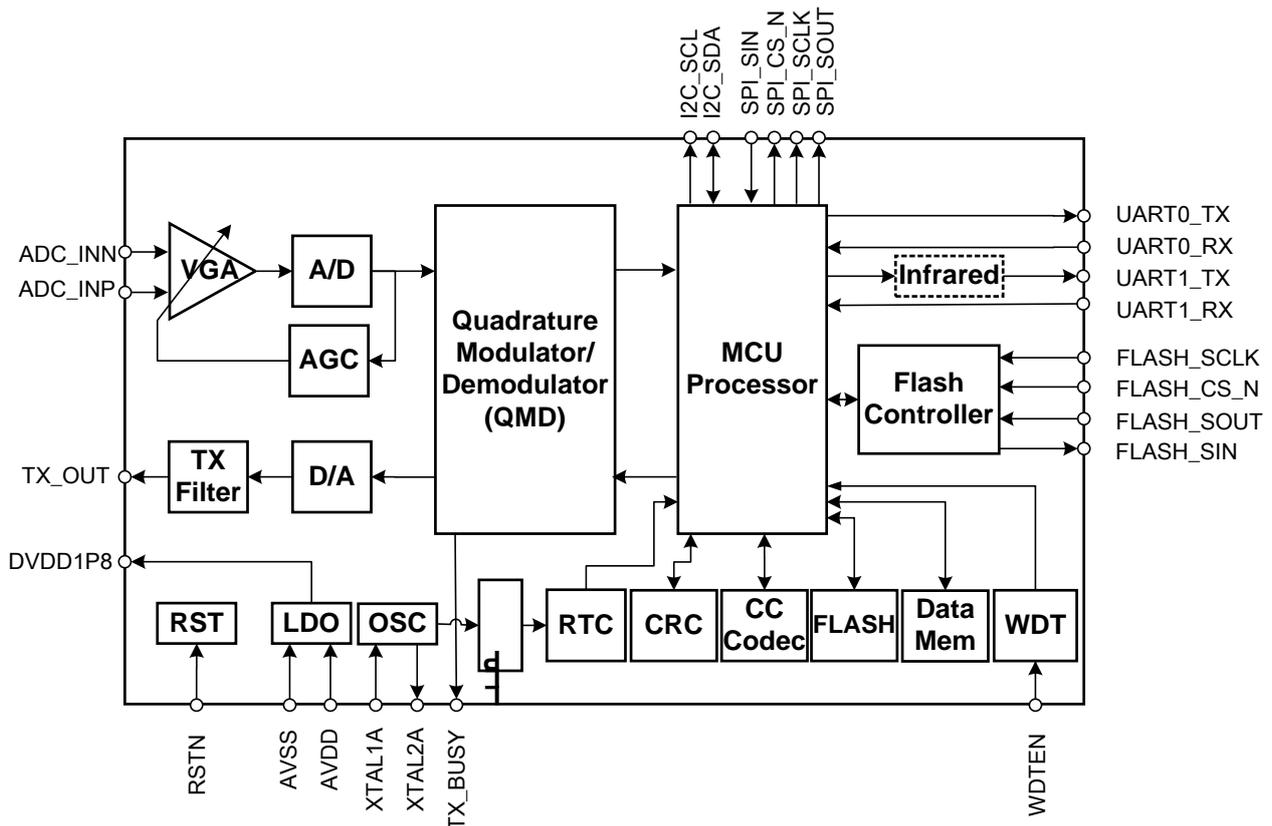


图 1 HT8550 系统框图

1.5 缩略语

缩略语	英文原文
ADC	Analog to Digital Converter
AGC	Auto Gain Control
AMP	Amplifier
AMR	Automatic Meter Reading
ARQ	Automatic Repeat Request
BOR	Brown Out Reset
BPF	Band Pass Filter
CRC	Cyclic redundancy Check
CSMA	Carrier Sense Multiple Access
DC	Dual Carrier
DDS	Directly Digital Synthesizer
DM	Data memory
DNL	Differential Non-linearity
DPSK	Differential Phase-shift keying
FCH	Frame Control Header
FEC	Forward Error Correction
GPIO	General Purpose IO
INL	Integral Non-linearity
IRQ	Interrupt Request
LDO	Low Drop -Out Regulator
LSB	Least Significant Bit
MSB	Most Significant Bit
OSC	Oscillator
OSR	Over Sample Rate
PLC	Power line communication
PM	Program memory
POR	Power On Reset
RC	Repetition Code
RSSI	Receiver Signal Strength Indicator
RTC	Real Time Clock
SC	Single Carrier
SCM	System Clock Managment
SFR	Special Function Register
SPI	Serial Peripheral Interface
UAM	User Application Mode
WDT	Watch Dog Timer

表 1 缩略语

2 引脚说明

2.1 引脚框图

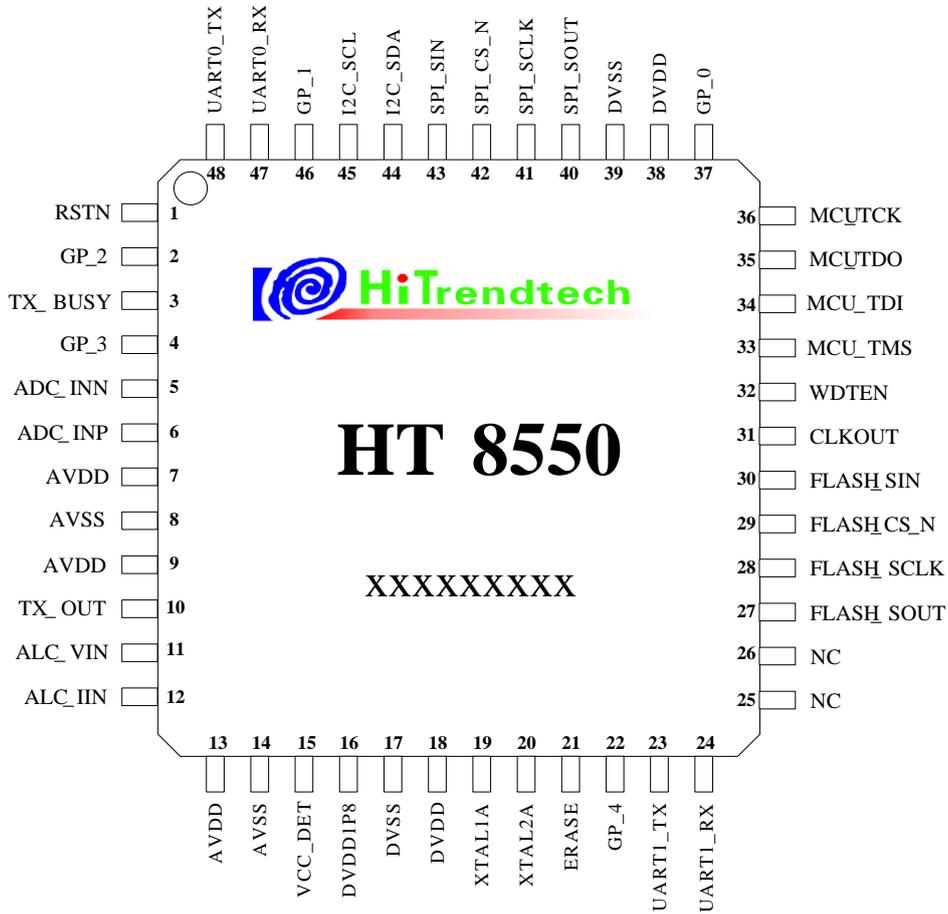


图 2 引脚分布图

2.2 引脚功能

Pin No.	Pin Name	Default Function
1	RSTN	External Reset signal, low active
2	GP_2	General Purpose Input/Output
3	TX_BUSY	TX busy indicator
4	GP_3	General Purpose Input/Output
5	ADC_INN	RX negative signal input
6	ADC_INP	RX positive signal input
7	AVDD	5V Analog power supply
8	AVSS	Analog power ground
9	AVDD	5V Analog power supply
10	TX_OUT	TX signal output
11	ALC_VIN	Sampling voltage signal as input for ALC circuit
12	ALC_IIN	Sampling current signal as input for ALC circuit
13	AVDD	5V Analog power supply
14	AVSS	Analog power ground
15	VCC_DET	VCC power detect
16	DVDD1P8	1.8V digital core power supply, add a 10uF external decoupling capacitor
17	DVSS	Digital ground
18	DVDD	5V Digital IO power supply
19	XTAL1A	24M Crystal Oscillator Input
20	XTAL2A	24M Crystal Oscillator output
21	ERASE	Erase flash program,negative pulse enable
22	GP_4	General Purpose Input/Output
23	UART1_TX	MCU UART1 data transmitting
24	UART1_RX	MCU UART1 data receiving
25	NC	No connection
26	NC	No connection
27	FLASH_SOUT	FLASH download data out
28	FLASH_SCLK	FLASH download clk
29	FLASH_CS_N	FLASH download control port
30	FLASH_SIN	FLASH download data in
31	CLKOUT	MAC clock output
32	WDTEN	On chip watch_dog enable,high active
33	MCU_TMS	JTAG Mode Select Input
34	MCU_TDI	JTAG Data Input
35	MCU_TDO	JTAG Data Ouput
36	MCU_TCK	JTAG Clock
37	GP_0	General Purpose Input/Output
38	DVDD	5V Digital IO power supply
39	DVSS	Digital ground
40	SPI_SOUT	SPI serial data output
41	SPI_SCLK	SPI serial clock
42	SPI_CS_N	SPI chip select
43	SPI_SIN	SPI serial data input
44	I2C_SDA	I2C interface serial data
45	I2C_SCL	I2C interface serial clock
46	GP_1	General Purpose Input/Output
47	UART0_RX	MCU UART0 data transmitting
48	UART0_TX	MCU UART0 data receiving

表 2 引脚概述

Note: 表2中引脚功能描述只限于默认属性，多数数字电路引脚都是复用的，GPIO、中断等均可扩展，具体应用请参照相关章节。

3 电气规格

3.1 极限参数

ITEM	RANGE	UNITS
VDD to GND	-0.5~+6.0	V
IO to GND	-0.5~+6.0	V
Continuous Power Dissipation	1500 (derate 25mW/°C, above 70°C)	mW
Operating Temperature Range	-40~+85	°C
Storage Temperature Range	-50~+150	°C
Lead Temperature(soldering,10S)	300	°C

表 3 极限参数

注意：超过极限参数使用芯片，将可能导致芯片永久性失效。

3.2 电气特性

(电源电压：4.5V~5.5V，工作温度范围：-40°C~85°C，典型工作温度25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN TYP MAX	UNIT S
DC CHARACTERISTICS				
Supply Voltage	V_{DD}		4.5 5.0 5.5	V
Supply Current	I_{DD}	RX mode @12MHz	10.5 13.0 17.8	mA
		TX mode @12MHz	12	
		Listen Mode@1.2MHz	7.2 9.6 14.4	
High-level Input Voltage	V_{IH}		4.0	V
Low-level Input Voltage	V_{IL}		1.0	V
High-level Output Voltage	V_{OH}	Source 5mA	4.2	V
Low-level Output Voltage	V_{OL}	Sink 5mA	0.8	V
AC CHARACTERISTICS				
RECEIVER				
Sensitivity		5.0V, 25°C	1	μV (rms)
VGA Dynamic Range			-3~42	dB

ADC	Full Scale	V _{pp}	2	V
	Sample Frequency	F _s	12/24	MHz
	Resolution	OSR=32	12	Bits
		OSR=16	10	
	INL		±0.5	LSB
	DNL		±0.5	LSB
	SFDR		70	dB
TRANSMITTER				
Output Range	V _{pp}		1.0	V
Pre-Drvier Gain Range			-0.5 ~ 10.5	dB
DAC	Resolution		10	Bits
	INL		±0.5	LSB
	DNL		±0.5	LSB
	SFDR		70	dB

表 4 电气特性

4 电源与时钟

4.1 电源系统

HT8550 具有多组供电电源，其中DVDD和DVSS为数字电路提供电源，AVDD和AVSS为模拟电路提供电源。

HT8550 片内集成了1.8V LDO，因此它只需采用5V外部单电源供电。

4.2 低功耗

HT8550采用了一系列低功耗设计，如门控时钟、时钟调整、MCU IDLE模式、MCU STOP模式等低功耗设置。根据需要灵活地配置系统工作时钟，休眠或开启相关功能模块，可有效降低芯片功耗。

4.2.1 门控时钟

HT8550定义了寄存器CLK_ON和CLK_OFF（详见4.3 时钟系统），用于开启和关闭一些功能模块的时钟，以降低功耗。例如，在接收（RX）阶段，可以关闭发送（TX）时钟（其他情形详见《应用手册》）。

4.2.2 时钟调整

HT8550中的时钟处理模块中，有多组时钟分频器（详见4.3 时钟系统），可以用来调整各个功能模块的工作时钟，使其在较低的工作频率下，获得较低的功耗。

4.2.3 MCU IDLE 模式

配置寄存器PCON[0] = 1，可控制MCU进入IDLE模式。当MCU进入IDLE模式后，8051

外设时钟仍然正常运行，但是MCU核的时钟将被关闭，从而降低功耗。当MCU收到任何中断或者复位后，将自动从IDLE模式切换到正常工作模式。

4.2.4 MCU STOP 模式

配置寄存器PCON[1] = 1，可控制MCU进入STOP模式。当MCU进入STOP模式后，8051外设时钟和MCU核的时钟将被关闭，从而降低功耗。当MCU收到PHY中断，DMA中断，RTC中断，GPIO中断或掉电中断或复位后，将自动从STOP模式切换到正常工作模式。STOP模式比IDLE模式节省更多的功耗，但是需要外部中断将MCU唤醒。

4.2.5 特殊功能寄存器

电源寄存器列表如下：

地址	名称	复位值	功能简介
0x87	PCON	0x08	电源控制寄存器

表 5 电源控制寄存器列表

4.2.5.1 PCON

位	位名称	复位值	描述	属性
7	SMOD	0	串口 0 波特率选择位 设置为 1 时，波特率将翻倍。	R/W
6	WDT_TM	0	看门狗内部测试标志位 设置为 1 时，看门狗输入时钟=系统时钟/12。 CPU 正常模式下，必须设置为 0。	R/W
5	ISR_TM	0	中断服务函数内部测试标志位 设置为 1 时，内部中断源只能通过 GPIO 口控制。 CPU 正常模式下，必须设置为 0。	R/W
4	PMW	0	程序空间写使能位 设置为 1 时，MCU 可以写程序空间。CPU 正常工作模式下，必须设置为 0。	R/W
3	P2SEL	1	地址空间扩展选择位 当 CPU 执行 MOVX 指令是，选择 16 位地址扩展方式。CPU 正常工作模式下，必须设置为 1。	R/W
2	GF0	0	通用标志位 可以作为 bit 变量使用。	R/W
1	STOP	0	STOP 模式控制位 设置为 1 时，CPU 将进入 STOP 模式。读返回值总是 0。	R/W
0	IDLE	0	IDLE 模式控制位 设置为 1 时，CPU 将进入 IDLE 模式。读返回值总是 0。	R/W

表 6 电源控制寄存器(PCON 0x87H)

4.3 时钟系统

HT8550采用了24MHz晶体振荡器，片内PLL将其倍频至48MHz。HT8550的PHY子系统和MAC子系统都配有时钟处理模块，将输入时钟进行选频和分频后，提供给PHY和MCU作为工作时钟。

HT8550内部产生32.768KHz时钟，作为RTC时钟和各功能模块工作时钟源。

4.3.1 PLL 时钟

PLL的时钟源为外部晶体振荡器（OSC）提供的24MHz时钟，PLL固定将输入时钟倍频2，然后输出48MHz时钟给MCU和PHY系统。HT8550内部有时钟锁定检测模块。时钟锁定标志将反映在寄存器CLK_CTRL[6]。MCU通过查验该寄存器，可以判断PLL是否进入锁定状态。如果为1，说明PLL已经锁定，从而可以将时钟源从OSC切换到PLL。

4.3.2 时钟源选择和时钟分频

时钟源选择模块有3路时钟源，24MHz OSC时钟、48MHz PLL时钟和32.768KHz RTC时钟。时钟源选择模块负责选取其中一路时钟，作为工作时钟源。时钟分频模块中有多组时钟分频器，将工作时钟源分频后，提供给MAC子系统和PHY子系统的各个功能模块。具体如下表：

功能模块	输入时钟频率范围	说明
时钟源选择	32kHz, 24MHz, 48MHz	时钟源选择模块负责从 24MHz OSC 时钟、48MHz PLL 时钟和 32.768KHz RTC 时钟 3 路时钟中，选取其中一路时钟，作为工作时钟源。工作时钟源为各功能模块时钟分频器的源头。
MAC 子系统	Min:32kHz Max:16MHz	MAC 时钟必须一直处于开启状态。可以通过配置寄存器 MAC_CLK_DIV 调整 MAC 时钟的频率。 注：MAC 时钟不得高于 16MHz。
QMD 模块	6MHz, 12MHz	系统复位后，QMD 时钟默认为关闭状态。在进行 QMD 寄存器读写前，必须配置寄存器 CLK_ON 开启 QMD 时钟。 QMD 默认工作时钟为 12MHz。配置寄存器 ADC_QMD_CLK_CTRL 可调整其值。 注： 1. 在发送状态下，QMD 不支持 6MHz 工作模式。 2. 在接收状态下，当 QMD 处于 6MHz 工作模式时，要求 ADC 时钟为 12MHz，此时，符号速率小于 12Ksps；当 QMD 处于 12MHz 模式时，ADC 时钟可以为 12MHz/24MHz。 3. 只有当 QMD 模块处于 IDLE 模式下，才可以将 QMD 时钟在 12MHz/6MHz 之间切换。
ADC 模块	12MHz 24MHz	配置寄存器 CLK_ON 可以控制 ADC 时钟的开关，以降低功耗。 注：

		<ol style="list-style-type: none"> 当 QMD 处于 RX 模式时，ADC 时钟将被强制打开。为了降低功耗，推荐将 CLK_ON 寄存器中 ADC 时钟开关的 bit 置 0。 当载波频率低于 187.5kHz 时，推荐将 ADC 时钟配置为 12MHz；高于 187.5kHz 时，将 ADC 时钟配置为 24MHz。通过配置寄存器 ADC_QMD_CLK_CTRL 可以改变 ADC 的时钟。
DAC 模块	6MHz	配置寄存器 CLK_ON 可以控制 DAC 时钟的开关，以降低功耗。 注：当 QMD 处于 TX 模式时，DAC 时钟将被强制打开。为了降低功耗，推荐将 CLK_ON 寄存器中 DAC 时钟开关的 bit 置 0。
Flash 控制器	Min:32kHz Max:48MHz	访问片外 Flash 的串行时钟，默认为关闭。 当需要访问片外扩展时，可配置寄存器 CLK_ON 开启该时钟。 也可通过 FLASH_CLK_DIV 寄存器调整该时钟频率。

表 7 时钟模块表格

4.3.3 特殊功能寄存器

时钟相关寄存器列表如下：

地址	名称	复位值	功能简介
0xAC	ADC_QMD_CLK_CTRL	0x0	ADC 和 QMD 时钟控制 注： <ol style="list-style-type: none"> 在对 CLK_CTRL 寄存器进行写操作后，等待 10 个时钟周期后，方可对本寄存器进行写操作。 只在 QMD 处于 IDLE 模式时才可修改本寄存器值。
0xD6	FLASH_CLK_DIV	0x0	外部 Flash 时钟分频控制
0xE5	CLK_ON	0x09	时钟开启控制寄存器 注： <ol style="list-style-type: none"> 对本寄存器的相应位写入 1，可开启相应时钟；写入 0，无影响。 当且仅当相应位被写入 1 时，相应位会被置 1，其他条件不会令其置 1（例外：CLK_ON[0]始终为 1）。 读取该寄存器，可判断哪些功能模块的时钟已开启。
0xE6	CLK_OFF	0x29	时钟关闭控制寄存器 注： <ol style="list-style-type: none"> 对本寄存器的相应位写入 1，可关闭相应时钟；写入 0，无影响。 读取该寄存器，将得到 CLK_ON

地址	名称	复位值	功能简介
			的值。 3. 读取该寄存器，可判断哪些功能模块的时钟已开启。
0xF1	MAC_CLK_DIV	0x01	MAC 时钟分频控制
0xF2	CLK_CTRL	0x10	时钟控制寄存器

表 8 时钟相关寄存器列表

4.3.3.1 ADC_QMD_CLK_CTRL

位	位名称	复位值	描述	属性
7:2	-	0x00	保留位。	-
1	-	0	QMD 时钟控制 0: QMD 时钟为 12MHz。 1: QMD 时钟为 6MHz（只有在接收模式下，当 ADC 时钟为 12MHz，且符号率（symbol rate）低于 12K 时，才可使用该配置。在发送模式下不可使用该配置；如果该位已被置 1，在进入发送模式前，须手动配置为 0）。	R/W
0	-	0	ADC 时钟控制 0: ADC 采样时钟为 24MHz，采样率为 1.5Msps。 1: ADC 采样时钟为 12MHz，采样率为 750Ksps。	R/W

表 9 ADC 和 QMD 时钟分频控制寄存器(ADC_QMD_CLK_CTRL 0xACh)

4.3.3.2 FLASH_CLK_DIV

位	位名称	复位值	描述	属性
7:2	-	0x00	保留位。	-
1:0	-	00	Flash 时钟分频系数（工作时钟源/flash 时钟） 00: 不分频； 01: 2 分频； 10: 3 分频； 11: 4 分频； 例如，该寄存器值为 0x1，系统时钟源为 48MHz PLL 时钟，则提供给 Flash 的时钟为 24MHz。	R/W

表 10 外部 Flash 时钟分频控制寄存器(FLASH_CLK_DIV 0xD6H)

4.3.3.3 CLK_ON

位	位名称	复位值	描述	属性
7:6	-	00	保留位。	-

5	-	0	开启 Flash 时钟 对该位写入 1 时，可开启 flash 时钟。 当 CLK_OFF[5] 被写入 1 时，CLK_ON[5] 将被清零。	R/W
4	-	0	开启 QMD 时钟 该位写入 1 时，可开启 QMD 时钟。 当 CLK_OFF[4] 被写入 1 时，CLK_ON[4] 将被清零。	R/W
3	-	1	保留位。	-
2	-	0	开启 ADC 时钟 对该位写入 1 时，将开启 ADC 时钟。 如果该位为 0，当 QMD 进入接收模式时，ADC 时钟也可以自动被开启。此时，该位仍保持 0。 当 CLK_OFF[2] 被写入 1 时，CLK_ON[2] 将被清零。	R/W
1	-	0	开启 DAC 时钟 对该位写入 1 时，将开启 DAC 时钟。 如果该位为 0，当 QMD 进入发送模式时，DAC 时钟也可以自动被开启。此时，该位仍保持 0。 当 CLK_OFF[1] 被写入 1 时，CLK_ON[1] 将被清零。	R/W
0	-	1	开启 MAC 时钟 注：始终开启，不会关闭。	R

表 11 时钟开启控制寄存器(CLK_ON 0xE5H)

4.3.3.4 CLK_OFF

位	位名称	复位值	描述	属性
7:6	-	00	保留位。	-
5	-	1	关闭 Flash 时钟 写 1 关闭时钟，写 0 无效。	R/W
4	-	0	关闭 QMD 时钟 写 1 关闭时钟，写 0 无效。	R/W
3	-	1	保留位。	-
2	-	0	关闭 ADC 时钟 当 QMD 进入接收模式时，即使对该位写入 1，也不会关闭 ADC 时钟，但 CLK_ON[2] 将会被清零。	R/W
1	-	0	开启 DAC 时钟 当 QMD 进入发送模式时，即使对该位写入 1，也不会关闭 DAC 时钟，但 CLK_ON[1] 将会被清零。	R/W
0	-	1	保留位。	-

表 12 时钟开启控制寄存器(CLK_OFF 0xE6H)

4.3.3.5 MAC_CLK_DIV

位	位名称	复位值	描述	属性
7:0	-	0x01	MAC 时钟分频系数(工作时钟源/MAC 时钟) n: MAC 时钟为系统时钟的 n+1 分频。 例如, 该寄存器值为 0x3, 系统时钟为 24MHz OSCar 时钟, 则 MAC 时钟为 6MHz。 注: MAC 时钟最高频率为 16MHz。	R/W

表 13 MAC 时钟分频控制寄存器(MAC_CLK_DIV 0xF1H)

4.3.3.6 CLK_CTRL

位	位名称	复位值	描述	属性
7	-	0	保留位。	-
6	-	0	PLL 锁定标志位 0: PLL 未锁定 1: PLL 已锁定	R
5:4	-	01	PLL 锁定检测精度设置 00: bypass, PLL 锁定标志始终为 1; 01: 低精度; 10: 中精度; 11: 高精度;	R/W
3	-	0	bit[3],bit[0]结合, 控制时钟源选择 00: 24MHz OSC 时钟作为工作时钟源; 01: 48MHz PLL 时钟作为工作时钟源; 1X: 32.787KHz 时钟作为系统时钟源, 系统进入低功耗模式。	R/W
2		0	关闭 PLL 控制位 0: 开启 PLL; 1: 关闭 PLL;	R/W
1		0	保留位。	-
0		0	bit[3],bit[0]结合, 控制时钟源选择	R/W

表 14 时钟控制寄存器(CLK_CTRL 0xF2H)

4.4 复位系统

HT8550支持5种复位方式:

- 上电复位POR(Power-On Reset)
- 掉电复位BOR(Brown Out Reset)
- 外部引脚/RST复位
- 看门狗复位
- 软件复位

4.4.1 上电/掉电复位

当电源加到芯片上时, 将会发生上电复位, 复位电平3.5V。片上也集成了BOR功能, 当

电源电压跌落至3.3V或以下时，将会发生掉电复位。

4.4.2 外部引脚复位

外部复位引脚RSTN出现低电平时，将引起系统复位（参考复位电压为0.7V，即低于0.7V时芯片复位）。该引脚应连接100kΩ上拉电阻与0.1uF电容构成的上电检测电路。

4.4.3 看门狗复位

看门狗是一个特殊的定时器，依据系统时钟计时，计满预定时间则发出溢出脉冲，产生看门狗复位。如果在溢出脉冲发生前将看门狗计数器清零，则不会产生看门狗复位信号。看门狗复位只是将MCU核复位，其他寄存器将保留上一次的值。

4.4.4 软件复位

通过设置寄存器RST_SET[0] = 1，可产生reboot复位，系统将重新启动。将MCU核和外设将初始化，但是时钟分频相关的寄存器为上一次的值。

4.4.5 特殊功能寄存器

复位相关寄存器列表如下：

地址	名称	复位值	功能简介
0xE7	RST_SET	0x0	复位设置 注： 1. 对本寄存器的相应位写入 1，可产生相应复位信号；写入 0，无影响。 2. 位[4:2],[0]被写入 1 后，将产生相应复位信号。产生的复位信号将在复位动作完成后，自动撤销。复位信号撤消后，读取该寄存器，相应位为先前写入的值 1。
0xE8	RST_CLR	0x0	复位撤销 注： 1. 读取本寄存器，将得到 RST_SET 的值。

表 15 复位寄存器列表

4.4.5.1 RST_SET

位	位名称	复位值	描述	属性
7	-	0	boot 完成状态标志	R
6	-	0	boot 锁定状态标志 (Flash 验证失败标志)	R
5	-	0	保留位。	R
4	-	0	设置外部 flash 复位	R/W
3	-	0	设置卷积编码器复位	R/W
2	-	0	设置 QMD 复位	R/W
1	-	0	设置 8051 核复位 对该位写入 1，该位将被置 0，并使 8051 核处于复位状态。 对 RST_CLR[1]写入 1，RST_SET[1]将被置 1，同时，8051 核进入工作状态。	R/W
0	-	0	设置 reboot 复位	R/W

表 16 复位设置寄存器(RST_SET 0xE7H)

4.4.5.2 RST_CLR

位	位名称	复位值	描述	属性
7:2	-	0x0	保留位。	R
1	-	0	撤销 8051 核复位 对该位写入 1，可撤销 8051 复位信号，RST_SET[1]变为 1，8051 核进入工作状态；写入 0，无影响。	R/W
0	-	0	保留位。	R

表 17 复位撤销寄存器(RST_CLR 0xE8H)

5 控制单元(MCU)

5.1 MCU 架构

HT8550采用R8051XC内核，具有和8051兼容的体系架构。

- 支持多种工作时钟，最高可运行在20MHz
- 8KB数据存储空间
- 128KB FLASH程序存储空间
- Real Time Clock (RTC)
- DMA
- 片外FLASH扩展
- 2个UART接口，包含红外调制与解调功能
- SPI主/从接口
- I2C 主/从接口
- 3 个定时器
- 硬件看门狗
- 8个外部中断
- 22 个GPIO

5.2 存储单元

5.2.1 概述

对HT8550片内MCU来说，存储器结构与标准8051相似，可以利用片内存储资源，也可以用片外存储资源。

片内存储资源：

- 内部RAM
- 外部数据存储器
- 程序存储器

- 特殊功能寄存器 (SFR)

片外存储资源:

- 可接外部flash

片外可扩展Flash为程序存储器, 最高可达512KB

5.2.2 片内资源访问

对HT8550片内MCU来说, 存储器结构与标准8051相似, 有3个存储空间:

- 128KB程序存储器 (Program Memory)。
- 256B内部数据存储器 (Internal Data Memory)。
- 8KB扩展数据存储器 (External Data Memory)。

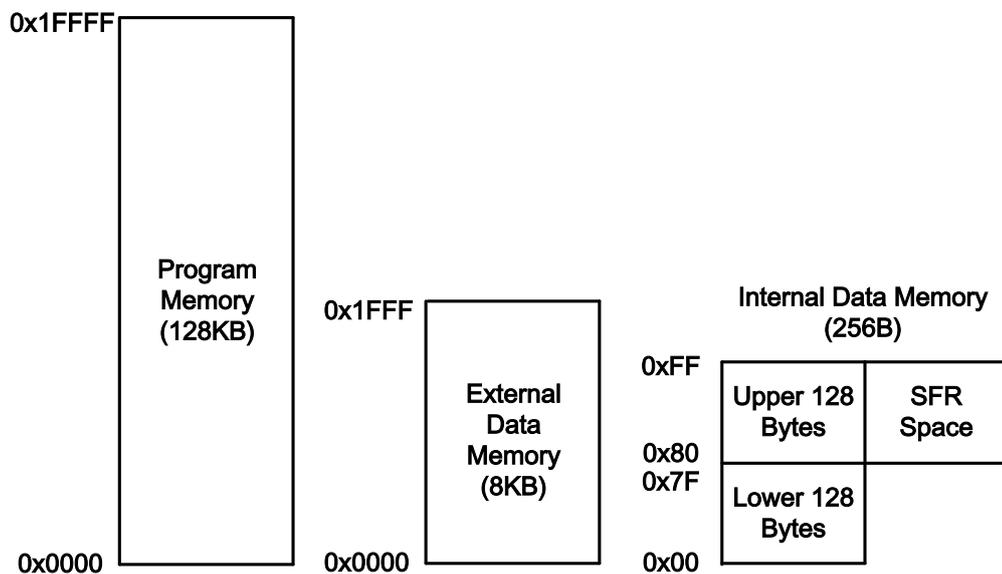


图 3 存储器空间 (片内 MCU 访问)

程序存储器、内部数据存储器 and 扩展数据存储器是分开的, 分别有自己的寻址系统、控制信号和功能。程序存储器用来存放程序和一些常量, 用MOVC指令访问; 数据存储器用来存放程序运行时需要的变量, 内部数据存储器用MOV指令访问, 扩展数据存储器用MOVX指令访问。在扩展存储模式下, 通过将FLASH映射到PM, 可实现在FLASH上直接运行程序。

内部数据存储器是最灵活的地址空间。它分为物理上独立且性质不同的三个区: 0x00~0x7F单元组成的128字节地址空间的RAM区; 0x80~0xFF空间组成的高128字节的RAM区; 128字节地址空间的特殊功能寄存器区。

内部数据存储器中不同的地址区域功能结构如图5所示。

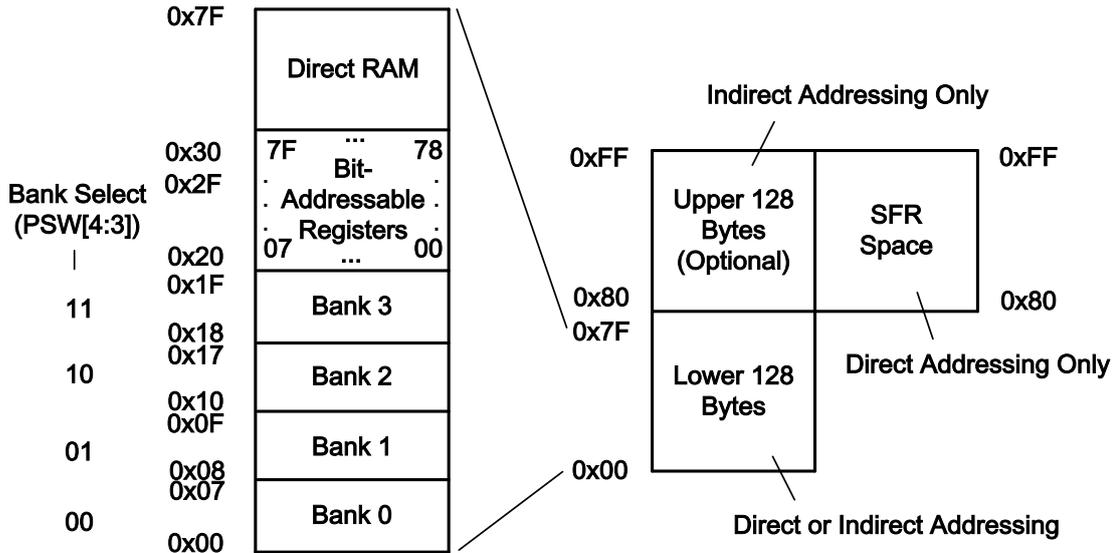


图 4 内部数据存储器功能结构

其中0x00~0x1F共32个字节是四个通用工作寄存器区，每一个区有八个工作寄存器R0~R7。每个区中R0~R7地址见表10。当前程序使用的工作寄存器区是由状态字PSW（SFR 0xD0）中的RS1和RS0来指示的。MCU通过对PSW中RS1和RS0位内容的修改，就能任选一个工作寄存器区。这个特点使HT8550具有快速现场保护的功能。如果用户程序不需要四个工作寄存器区，则不用的工作寄存器区单元可以作一般的RAM使用。Keilc的C编译环境中经常使用某一组R0-R7传递参数，因此需要使用宏命令#pragma NOAREGS防止寄存器组切换，避免出现参数传递错误。

内部数据存储器的0x20~0x2F为位寻址区。位寻址区的每一位都可以视作软件触发器，由程序直接进行位处理。通常把各种程序状态标志、位控制变量设在位寻址区。同样，位寻址单元也可以作为一般的数据缓冲器使用。

0 区		1 区		2 区		3 区	
地址	寄存器	地址	寄存器	地址	寄存器	地址	寄存器
00H	R0	08H	R0	10H	R0	18H	R0
01H	R1	09H	R1	11H	R1	19H	R1
02H	R2	0AH	R2	12H	R2	1AH	R2
03H	R3	0BH	R3	13H	R3	1BH	R3
04H	R4	0CH	R4	14H	R4	1CH	R4
05H	R5	0DH	R5	15H	R5	1DH	R5
06H	R6	0EH	R6	16H	R6	1EH	R6
07H	R7	0FH	R7	17H	R7	1FH	R7

表 18 内部数据存储器通用工作寄存器区

在实际的程序中需要堆栈以保存MCU的现场，堆栈原则上可以设在内部数据存储器的任意区域内，但一般设在30H~FFH的范围内。栈顶的位置由栈指针SP指出。

5.2.3 片外资源访问

片外可以扩展一个512KB的flash存储器，该flash可作为外部程序存储器，地址从0x80000开始，可通过PAGESEL特殊寄存器进行地址选择跳到外部flash。与片外flash进行连接的是SPI协议模块，所以片外flash必须是带spi接口的。详细配置参考《flash控制器》。

5.3 DMA

5.3.1 概况

HT8550包含两条独立的DMA线程，能够实现DM和DM之间的数据DMA操作；此外DMA模块还可以实现以下功能：

- 利用DMA读取或写入MCU的SPI接口
- 利用DMA读取或写入MCU的UART

DMA模块还具有CRC校验功能和扰码生成功能，其中线程0可以生成CRC校验结果，线程1可以生成扰码结果。每个DMA线程都有各自的特征多项式，并支持8、16、32位CRC校验。DMA传输的数据长度没有包括校验位和扰码。如果DMA源地址和目的地址相同，则不会向目标地址写数据，但是会产生CRC结果或者扰码结果。当选择CRC校验时，计算出来的校验位会与数据后的校验位进行对比，如果不一致，将会产生错误标志位。

每个DMA线程都能在DMA结束的时候对8051请求中断。可以通过读取DMACSR.5判断中断源。

5.3.2 DMA 操作流程

- 1) 通过配置寄存器DMASEL选择线程；
- 2) 如果线程选择5，则可以配置CRC多项式或者扰码多项式以及校验的初始值；当DMA传输完成后，硬件按照配置的公式计算CRC或者扰码的结果。
- 3) 完成第2步后，需要设置线程为0或者1，通过配置寄存器DMAS0-4和DMAT0-4设置DMA传输的源地址和目标地址；设置的地址为32位的全局地址；
- 4) 通过配置寄存器DMAC0-1，选择DMA传输的长度；单位为字节或者比特；
- 5) 通过配置寄存器DMATC和寄存器DMASC选择DMA传输的目标地址和源地址存储类型，以及DMA传输时，地址自动增减或不变；
- 6) 通过配置寄存器DMACSR，可以选择是否使能DMA传输完成中断；
- 7) 最后，通过配置寄存器DMACSR启动DMA传输。

5.3.3 特殊功能寄存器

DMA 寄存器列表如下：

地址	名称	复位值	功能简介
0xA1	DMASEL	0x0	DMA 线程选择寄存器。
0xA2	DMAS0	0x0	DMA 配置和状态寄存器，根据 DMASEL 寄存器的配置，可以控制 3 个线程。
0xA3	DMAS1	0x0	
0xA4	DMAS2	0x0	
0xA5	DMAS3	0x0	

地址	名称	复位值	功能简介
0xA6	DMAT0	0x0	
0xA7	DMAT1	0x0	
0xB1	DMAT2	0x0	
0xB2	DMAT3	0x0	
0xB3	DMAC0	0x0	
0xB4	DMAC1	0x0	
0xB5	DMACSR	0x0	
0xB6	DMATC	0x0	
0xB7	DMASC	0x0	

表 19 DMA 寄存器列表

5.3.3.1 DMASEL

位	位名称	复位值	描述	属性
7:5	-	0	保留位。读取该 3 位，得到 0x0。	R
4:3	-	0	DMA 错误标志位 Bit4: 线程 1 的 DMA 错误标志位; Bit3: 线程 0 的 DMA 错误标志位; 当 DMASEL[4:3]置位时,代表 FIFO 有溢出或者下溢,并且会产生中断,无论 DMA 中断是否使能。可以通过写 1,清除相应的错误标志位,并且将 DMA 复位。可以写两次 1 到相应的标志位确保线程被完全复位。	R/W
2:0	-	0	线程选择 000 = Thread0; 001 = Thread1; 010 = Reserved; 011 = Reserved; 100 = Reserved; 101 = MISC Parameters Register Select;	R/W

表 20 DMASEL 寄存器描述(DMASEL 0xA1H)

5.3.3.2 DMAS0

当 DMASEL[2:0] = 000b 或者 001b 时,以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:0	-	0	DMA 源地址 0 DMAS0[7:0]=Source Address[7:0]。	R/W

表 21 DMAS0 寄存器描述(DMAS0 0xA2H)

5.3.3.3 DMAS1

当 DMASEL[2:0] = 000b 或者 001b 时,以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:0	-	0	DMA 源地址 1 DMAS0[7:0]=Source Address[15:8]。	R/W

表 22 DMAS1 寄存器描述(DMAS1 0xA3H)

5.3.3.4 DMAS2

当 DMASEL[2:0] = 000b 或者 001b 时, 以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:0	-	0	DMA 源地址 2 DMAS0[7:0]=Source Address[23:16]。	R/W

表 23 DMAS2 寄存器描述(DMAS2 0xA4H)

5.3.3.5 DMAS3

当 DMASEL[2:0] = 000b 或者 001b 时, 以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:0	-	0	DMA 源地址 3 DMAS0[7:0]=Source Address[31:24]。	R/W

表 24 DMAS3 寄存器描述(DMAS3 0xA5H)

5.3.3.6 DMAT0

当 DMASEL[2:0] = 000b 或者 001b 时, 以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:0	-	0	DMA 目的地址 0 DMAS0[7:0]= Target Address [7:0]。	R/W

表 25 DMAT0 寄存器描述(DMAT0 0xA6H)

5.3.3.7 DMAT1

当 DMASEL[2:0] = 000b 或者 001b 时, 以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:0	-	0	DMA 目的地址 1 DMAS0[7:0]= Target Address [15:8]。	R/W

表 26 DMAT1 寄存器描述(DMAT1 0xA7H)

5.3.3.8 DMAT2

当 DMASEL[2:0] = 000b 或者 001b 时, 以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:0	-	0	DMA 目的地址 2 DMAS0[7:0]= Target Address [23:16]。	R/W

表 27 DMAT2 寄存器描述(DMAT2 0xB1H)

5.3.3.9 DMAT3

当 DMASEL[2:0] = 000b 或者 001b 时, 以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:0	-	0	DMA 目的地址 3 DMAS0[7:0]= Target Address [31:24]。	R/W

表 28 DMAT3 寄存器描述(DMAT3 0xB2H)

5.3.3.10 DMAC0

当 DMASEL[2:0] = 000b 或者 001b 时，以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:0	-	0	DMA 传输长度 0 当 DMASEL[2:0] = 000b 时， 单位：字节。传输的长度不包含 CRC，当选择 CRC 编码模式时，硬件计算的 CRC 结果会自动保存到目标地址的结尾；当选择 CRC 译码模式时，在源地址的后面必须增加 CRC 的值，硬件在数据传输完成后，会读取该值与计算的值进行比较。 当 DMASEL[2:0] = 001b 时， 单位：比特。当扰码器使能时，扰码器输出的数据是以字节为单位，如果输入的比特不是 8 的倍数，扰码器将在后面补 0。当扰码器关闭时，扰码器输出的结果只是将比特转化为字节。 DMAC0 [7:0]=Transfer Count[7:0] 0 代表不传送，1 代表传送 1 字节。	R/W

表 29 DMAC0 寄存器描述(DMAC0 0xB3H)

5.3.3.11 DMAC1

当 DMASEL[2:0] = 000b 或者 001b 时，以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:0	-	0	DMA 传输长度 1 DMAC1 [7:0]=Transfer Count[15:8]。	R/W

表 30 DMAC1 寄存器描述(DMAC1 0xB4H)

5.3.3.12 DMACSR

当 DMASEL[2:0] = 000b 时，以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7	-	0	启动 DMA。在 DMA 启动后，该位自动清为 0，此时读取得到 0。	R/W
6	-	0	DMA 忙状态，只读。	R/W
5	-	0	DMA 传送完成状态，写 0 清除。	R/W
4	-	0	DMA 中断使能。写 1 使能。	R/W
3	-	0	CRC 校验结果错误标志位，如果校验不正确，此位为 1。	R/W

2	-	0	CRC 模式 0 = 解码。要传输的数据和 CRC 多项式将会经过 CRC 单元，但是传输的数据长度不包括校验码。最终的校验码会和 0 比较。只有指定长度的数据会被写到目的地址。 1 = 编码。 指定长度的数据会经过 CRC 单元，最终的校验码会放在目的地址末尾。要传输的数据量不需要包括 CRC 校验码。 要传输的数据和 CRC 多项式将会经过 CRC 单元，但是传输的数据长度不包括校验码。最终的校验码会和 0 比较。只有指定长度的数据会被写到目的地址。	R/W
1:0	-	0	[1:0] CRC 多项式类型: 00=Disable; 01=1 Byte; 10=2 bytes; 11=4 bytes;	R/W

当 DMASEL[2:0] = 001b 时，以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7	-	0	开始 DMA，硬件清除。	R/W
6	-	0	DMA 忙状态，只读。	R/W
5	-	0	DMA 传送完成状态，写 0 清除。	R/W
4	-	0	DMA 中断使能。	R/W
3:2	-	0	保留。	R
1	-	0	初始化扰码状态机。写 1 复位，硬件自动清零。	R/W
0	-	0	保留。	R

表 31 DMACSR 寄存器描述(DMACSR 0xB5H)

5.3.3.13 DMATC

当 DMASEL[2:0] = 000b 或者 001b 时，以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:5	-	0	保留。	R
4:2	-	0	目标类型 000: Data Memory; 001: 保留; 010: 保留; 011: 保留; 100: SPI; 101: SERIAL0; 110: 保留; 111: 保留;	R/W

1:0	-	0	地址更新控制 00: 固定地址; 01: 自动增加; 10: 自动减少;	R/W
-----	---	---	---	-----

表 32 DMATC 寄存器描述(DMATC 0xB6H)

5.3.3.14 DMASC

当 DMASEL[2:0] = 000b 或者 001b 时, 以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:5	-	0	保留。	R
4:2	-	0	目标类型 000: Data Memory; 001: 保留; 010: 保留; 011: 保留; 100: SPI; 101: SERIAL0; 110: 保留; 111: 保留;	R/W
1:0	-	0	地址更新控制 00: 固定地址; 01: 自动增加; 10: 自动减少;	R/W

表 33 DMASC 寄存器描述(DMASC 0xB7H)

5.3.3.15 MISC_PARAM

当 DMASEL[2:0] = 101b 时, 以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7	-	0	置 1, 数据 bit 反向, 仅对 CRC 有效。	R/W
6	-	0	置 1, 先处理字节中的低位	R/W
5	-	0	置 1, CRC 结果取反。	R/W
4	-	0	为 1 时, CRC 初始值为全 1; 为 0 时, 初值为 0。	R/W
3:0	-	0	保留。	R

表 34 MISC_PARAM 寄存器描述(MISC_PARAM 0xA1H)

5.3.3.16 MISC_ADDR

当 DMASEL[2:0] = 101b 时, 以下寄存器对 MCU 可见

位	位名称	复位值	描述	属性
7:5	-	0	表格选择 000=CRC 多项式; 001=CRC 校验码/ 初始值;	R/W

4:0	-	0	表格的偏移地址，详细参考下表。	R/W
-----	---	---	-----------------	-----

表 35 MISC_ADDR 寄存器描述(MISC_ADDR 0xA2H)

CRC Polynomial & CRC Table			
名称	偏移量	复位值	描述
CRC Poly0	Offset 0	0x00	假设 $X^n = 1$ 。如果 CRC-8 X^8+X^2+X+1 ，则 Polynomial = 0x7 Polynomial[7:0]
CRC Poly1	Offset 1	0x00	Polynomial[15:8]
CRC Poly2	Offset 2	0x00	Polynomial[23:16]
CRC Poly3	Offset 3	0x00	Polynomial[31:24]
CRC Checksum0 Thread 0	Offset 0	0x00	Checksum[7:0]
CRC Checksum1 Thread 0	Offset 1	0x00	Checksum[15:8]
CRC Checksum2 Thread 0	Offset 2	0x00	Checksum[23:16]
CRC Checksum3 Thread 0	Offset 3	0x00	Checksum[31:24]
Scrambler Table			
SCR_Poly0	Offset 0	0x00	Scrambler Polynomial[7:0]
SCR_Poly1	Offset 1	0x00	Scrambler Polynomial[15:8]
SCR_State0	Offset 2	0x00	Scrambler State[7:0]
SCR_State1	Offset 3	0x00	Scrambler State[15:8]

表 36 CRC 和扰码的表格

5.4 中断系统

5.4.1 概述

HT8550中断系统支持8个外部中断和外设中断。

其中8个外部中断产生的条件如下：

- 将GPIO设置为外部中断模式，可以触发相应的外部中断。
- 2个DMA中断。
- RTC中断
- PHY中断(QMD中断(PHY_INTR[0])，卷积编码中断(PHY_INTR[2]))
- 掉电检测中断

外设中断包括：定时器T0、T1、T2中断、串口0，1中断，SPI中断，I2C中断。

5.4.2 基于 GPIO 的中断

设置寄存器GPIO_CTRL，将相应GPIO设置为外部中断（External Interrupt）功能，则相应GPIO可作为外部中断源（详见5.6 GPIO）。寄存器PHY_INTR_FLAGS反映了处于外部中断功能的GPIO的状态，以便于MCU确定中断源。PHY_INTR_FLAGS的值直接来自GPIO引脚，不受GPIO_INTR_POLARITY的影响。例如，GPIO port 0, 7, 9, 19被设置为外部中断功能，MCU收到外部中断后，读取寄存器PHY_INTR_FLAGS值为0x08，则可断定中断源为GPIO port 19。

外部中断与GPIO的对应关系：

Ext.Intr[0] = GPIO port 0 or GPIO port 8 or GPIO port 16

Ext.Intr[1] = GPIO port 1 or GPIO port 9 or GPIO port 17

Ext.Intr[2] = GPIO port 2 or GPIO port 10 or GPIO port 18

Ext.Intr[3] = GPIO port 3 or GPIO port 11 or GPIO port 19

Ext.Intr[4] = GPIO port 4 or GPIO port 12 or GPIO port 20

Ext.Intr[5] = GPIO port 5 or GPIO port 13 or GPIO port 21

Ext.Intr[6] = GPIO port 6 or GPIO port 14

Ext.Intr[7] = GPIO port 7 or GPIO port 15

5.4.3 DMA 中断

通过设置DMA中断使能寄存器(DMACSR[4]，详见 5.3 DMA)使得 DMA线程在完成DMA传输后，发出中断信号。其中，DMA线程0产生的中断对应Ext.Intr[6]，DMA线程1产生的中断对应Ext.Intr[5]。

5.4.4 RTC 中断

设置RTC中断使能寄存器（IEN4[4]，详见《中断使能寄存器》）后，RTC报警中断发生时，将在Rxt.Intr[0]出现。此时，MCU需进一步检查寄存器RTCC（详见6.3 RTC单元），以确定中断源。

5.4.5 QMD 中断

通过设置QMD_INTR_EN，可开启QMD中断。QMD中断发生后，将在Ext.Intr[1]上发生。详细信息请参考《应用手册》。

5.4.6 CC 中断

当卷积编码或解码结束后，都可以在Ext.Intr[3]上产生中断。MCU可查验寄存器PHY_INTR_FLAGS和CC_CONFIG以确定中断源。

5.4.7 掉电检测中断

掉电检测功能用于判断片外高压电源的跌落状态。外部电压通过引脚VCC_DET输入，与内部1.2V参考电压相比较。当外部电压低于1.2V时，在Ext.Intr[4]上产生中断信号。MCU可以查验寄存器VCC_LOSS以确定中断源。

5.4.8 中断列表

中断列表如下：

HT8550 中断	自然 优先级	中断 向量	中断向 量号	中断使能 (EA=1)	中断标志

External Interrupt 0/ RTC	0	0003H	0	IEN0.0 /IEN4.5	GPIF0/ RTCC.1
Serial 1 Interrupt	1	0083H	16	IEN2.0	S1CON.0 S1CON.1
External Interrupt 7 I2C_interrupt	2	0043H	8	IEN1.0	GPIF.7
Timer 0 Interrupt	3	000BH	1	IEN0.1	TCON.5
External Interrupt 2 SPI_interrupt	4	004BH	9	IEN1.1	GPIF.2
PHY_INTR[0]/ External Interrupt 1	5	0013H	2	PHIF.0 IEN0.2	PHIF.3/GPIF.1
PHY_INTR[2]/ External Interrupt 3	6	0053H	10	PHIF.2 IEN1.2	PHIF.5/GPIF.3
Timer 1 Interrupt	7	001BH	3	IEN0.3	TCON.7
External Interrupt 4 VCC_LOSS	8	005BH	11	IEN1.3	GPIF.4
Serial 0 Interrupt	9	0023H	4	IEN0.4	S0CON.0 S0CON.1
DMA_TH1_INTR/ External Interrupt 5	10	0063H	12	DMACSR.4 IEN1.4	DMACSR.5 /GPIF.5
Timer 2 Interrupt	11	002BH	5	IEN0.5 IEN1.7	IRCON.6 IRCON.7
DMA_TH0_INTR/ External Interrupt 6	12	006BH	13	DMACSR.4 IEN1.5	DMACSR.5/ GPIF.6

表 37 HT8550 中断系统

5.4.9 中断优先级

HT8550可设定 4 个中断优先级，只能根据固定的中断向量组进行向量组的优先级调整。优先级的调整通过寄存器IP0，IP1来设置。

4 个中断优先级如下 表所示：

IP1. X	IP0. X	优先级
0	0	Level 0(lowest)
0	1	Level 1
1	0	Level 2
1	1	Level 3(highest)

表 38 中断优先级

中断向量组与相应的中断优先级如下表所示：

中断向量组	组优先级控制位	高优先级	中优先级	低优先级
0	IP1.0 IP0.0	int_vect_03	int_vect_83	int_vect_43
1	IP1.1 IP0.1	int_vect_0B	int_vect_8B	int_vect_4B
2	IP1.2 IP0.2	int_vect_13	int_vect_93	int_vect_53
3	IP1.3 IP0.3	int_vect_1B	int_vect_9B	int_vect_5B
4	IP1.4 IP0.4	int_vect_23	int_vect_A3	int_vect_63
5	IP1.5 IP0.5	int_vect_2B	int_vect_AB	int_vect_6B

表 39 中断优先级组成员

5.4.10 中断处理

中断系统遵循下列两条基本规则：

1. 低优先级中断源可被高优先级中断源所中断，而高优先级中断源不能被同级或低优先级的中断源所中断；
2. 一种中断源不管是高优先级或低优先级，一旦得到响应，与它同级的中断源不能再中断它。

当同时收到几个同一优先级中断时，响应哪一个中断源取决于内部查询顺序。其优先级排列见上表中同级中断优先级列。

值得指出的是，EX0中断、EX6中断、EX5中断等都包含了若干个中断源。以EX0中断为例，当EX0中断发生时，需要查询相应的标志位GPIF0，RTCC.1来确定中断源。

5.4.11 特殊功能寄存器

中断寄存器列表如下：

地址	名称	复位值	功能简介
0x9A	IEN2	0x0	IEN2 中断使能
0xA8	IEN0	0x0	IEN0 中断使能
0xB8	IEN1	0x0	IEN1 中断使能
0xC0	IRCON	0x0	中断请求控制寄存器
0xFA	MAC_INTR_MASK	0x0	MAC 中断屏蔽
0xFB	GPIO_INTR_FLAGS	0x0	GPIO 外部中断标志
0xFC	PHY_INTR_FLAGS	0x0	PHY 中断标志

表 40 中断寄存器列表

5.4.11.1 IEN2

IEN2寄存器为8051内核寄存器。

位	位名称	复位值	描述	属性
7:1	-	0x0	保留位	R
0	ES1	0	串口 1 中断使能 当 ES1=0，串口 1 中断关闭； 当 ES1=1，并且 EAL=1，串口 1 中断使能。	R/W

表 41 IEN2 寄存器描述(IEN2 0x9AH)

5.4.11.2 IEN0

IEN0寄存器为8051内核寄存器。

位	位名称	复位值	描述	属性
7	EAL	0	全局中断使能 当 EAL=0, 关闭所有中断; 当 EAL=1, 全体中断使能, 若要打开某一个中断, 还需要打开它对应的中断使能位。	R/W
6	WDT	0	看门狗更新标志 看门狗更新寄存器, 需要连续先设置 SWDT(IEN1.6)为 1, 然后设置 WDT 为 1, 可以清零看门狗计数器。当设置该位为 1 后, 看门狗计数器在下一条指令后将被清零。	R/W
5	ET2	0	定时器 2 中断使能 当 ET2=0, 定时器 2 中断关闭; 当 ET2=1 并且 EAL=1, 定时器 2 中断使能。	R/W
4	ES0	0	串口 0 中断使能 当 ES0=0, 串口 0 中断关闭; 当 ES0=1 并且 EAL=1, 串口 0 中断使能。	R/W
3	ET1	0	定时器 1 溢出中断使能 当 ET1=0, 定时器 1 溢出中断关闭; 当 ET1=1 并且 EAL=1, 定时器 1 溢出中断使能。	R/W
2	EX1	0	外部中断 1 使能 当 EX1=0, 外部中断 1 关闭; 当 EX1=1 并且 EAL=1, 外部中断 1 使能, 由于 PHY[0]中断与 EX1 复用, 当需要使用 PHY[0]中断时, 需要将 EX1 置 1。	R/W
1	ET0	0	定时器 0 溢出中断使能 当 ET0=0, 定时器 0 溢出中断关闭; ET0=1 并且 EAL=1, 定时器 0 溢出中断使能。	R/W
0	EX0	0	外部中断 0 使能 当 EX0=0, 外部中断 0 关闭; 当 EX0=1 并且 EAL=1, 外部中断 0 使能, 由于 RTC 中断与 EX0 复用, 当使用 RTC 中断时, 需要将 EX0 置 1。	R/W

表 42 IEN0 寄存器描述(IEN0 0xA8H)

5.4.11.3 IEN1

IEN0寄存器为8051内核寄存器。

位	位名称	复位值	描述	属性
---	-----	-----	----	----

7	EXEN2	0	定时器 2 外部重载中断使能 当 EXEN2=0, 定时器 2 外部重载中断关闭; 当 EXEN2=1 并且 EAL=1, 定时器 2 外部重载中断使能。	R/W
6	SWDT	0	看门狗开始/刷新标志 看门狗软件使能/更新寄存器, 置 1, 则使能看门狗。如果连续设置 SWDT 为 1, 然后设置 WDT(IEN0.6)为 1, 可以清零看门狗计数器, 当看门狗计数器清零后, 该位自动被清零。	R/W
5	EX6	0	外部中断 6 使能 当 EX6 =0, 外部中断 6 和 DMA 线程 0 中断关闭 当 EX6 =1 并且 EAL=1, 外部中断 6 和 DMA 线程 0 中断使能, 但是 DMA 线程 0 使能, 还需要打开它对应的中断使能位。	R/W
4	EX5	0	外部中断 5 和 DMA 线程 1 中断使能 当 EX5 =0, 外部中断 5 和 DMA 线程 1 中断关闭; 当 EX5 =1 并且 EAL=1, 外部中断 5 和 DMA 线程 1 中断使能, 但是 DMA 线程 1 使能, 还需要打开它对应的中断使能位。	R/W
3	EX4	0	外部中断 4 使能 当 EX4 =0, 外部中断 4 关闭; 当 EX4 =1 并且 EAL=1, 外部中断 4 使能。	R/W
2	EX3	0	外部中断 3 和 PHY 中断 2 使能 当 EX3 =0, 外部中断 3 和 PHY 中断 2 关闭; 当 EX3 =1 并且 EAL=1, 外部中断 3 和 PHY 中断 2 使能, 但是 PHY 中断 2 使能, 还需要打开它对应的中断使能位。	R/W
1	EX2	0	外部中断 2 使能 当 EX2 =0, 外部中断 2 关闭; 当 EX2 =1 并且 EAL=1, 外部中断 2 使能。	R/W
0	EX7	0	外部中断 7 使能 当 EI2C =0, 外部中断 7 关闭; 当 EI2C =1 并且 EAL=1, 外部中断 7 使能。	R/W

表 43 IEN1 寄存器描述(IEN1 0xB8H)

5.4.11.4 IRCON

IRCON为8051内核寄存器。

位	位名称	复位值	描述	属性
7	EXF2	0	定时器 2 重载中断标志 当 EXF2=0, 表示定时器 2 重载中断未发生; 当 EXF2=1, 表示定时器 2 重载中断已发生。	R/W

6	TF2	0	定时器 2 溢出中断标志 当 TF2=0, 表示定时器 2 溢出中断未发生; 当 TF2=1, 表示定时器 2 溢出中断已发生。	R/W
5	IEX6	0	外部中断 6 和 DMA 线程 0 中断标志 当 IEX6=0, 表示外部中断 6 和 DMA 线程 0 中断未发生; 当 IEX6=1, 表示外部中断 6 或 DMA 线程 0 中断已发生。	R/W
4	IEX5	0	外部中断 5 和 DMA 线程 1 中断标志 当 IEX5=0, 表示外部中断 5 和 DMA 线程 1 中断未发生; 当 IEX5=1, 表示外部中断 5 或 DMA 线程 1 中断已发生。	R/W
3	IEX4	0	外部中断 4 中断标志 当 IEX4=0, 表示外部中断 4 中断未发生; 当 IEX4=1, 表示外部中断 4 中断已发生。	R/W
2	IEX3	0	外部中断 3 标志 当 IEX3=0, 表示外部中断 3 未发生; 当 IEX3=1, 表示外部中断 3 已发生。	R/W
1	IEX2	0	外部中断 2 和 PHY 中断 1 中断标志 当 IEX2=0, 表示外部中断 2 和 PHY 中断 1 中断未发生; 当 IEX2=1, 表示外部中断 2 或 PHY 中断 1 中断已发生。	R/W
0	IEX7	0	外部中断 7 中断标志 当 IEX7=0, 表示外部中断 7 中断未发生; 当 IEX7=1, 表示外部中断 7 中断已发生。	R/W

表 44 IRCON 寄存器描述(IRCON 0xC0H)

5.4.11.5 MAC_INTR_MASK

中断屏蔽寄存器MAC_INTR_MASK非8051内核寄存器。

位	位名称	复位值	描述	属性
---	-----	-----	----	----

7:0	-	0x0	<p>[7: 0]=0, 屏蔽外部中断</p> <p>EXT_INT0 = !(MASK[0] & (GPIO_INTR[0] RTC_INTR)) => R8051 ISR Vector 0x03</p> <p>EXT_INT1 = !(MASK[1] & (GPIO_INTR[1] QMD_INTR)) => R8051 ISR Vector 0x13</p> <p>EXT_INT2 = !(MASK[2] & GPIO_INTR[2]) => R8051 ISR Vector 0x4B</p> <p>EXT_INT3 = !(MASK[3] & (GPIO_INTR[3] CC_INTR)) => R8051 ISR Vector 0x53</p> <p>EXT_INT4 = (MASK[4] & (GPIO_INTR[4] VCC_LOSS)) => R8051 ISR Vector 0x5B</p> <p>EXT_INT5 = (MASK[5] & (GPIO_INTR[5] DMA_TH1_INTR)) => R8051 ISR Vector 0x63</p> <p>EXT_INT6 = (MASK[6] & (GPIO_INTR[6] DMA_TH0_INTR)) => R8051 ISR Vector 0x6B</p> <p>EXT_INT7 = (MASK[7] & GPIO_INTR[7]) => R8051 ISR Vector 0x43</p>	R/W
-----	---	-----	---	-----

表 45 MAC_INTR_MASK 寄存器描述(MAC_INTR_MASK 0xFAH)

5.4.11.6 GPIO_INTR_FLAGS

位	位名称	复位值	描述	属性
7:0	-	0x0	<p>GPIO 外部中断标志</p> <p>该寄存器直接反映了设置为外部中断功能的 GPIO 的状态。MAC 可依据该寄存器判断中断源。该寄存器的值直接来自 GPIO 引脚，不受 GPIO_INTR_POLARITY 的影响。</p>	R

表 46 GPIO_INTR_FLAGS 寄存器描述(GPIO_INTR_FLAGS 0xFBH)

5.4.11.7 PHY_INTR_FLAGS

位	位名称	复位值	描述	属性
7:6	-	00	保留位	-
5:3	-	000	<p>PHY 中断标志位</p> <p>相应位为高时，代表中断发生。MCU 写 1，可清除相应标志位。</p>	R/W
2:0	-	000	<p>PHY 屏蔽位</p> <p>写 0 时屏蔽中断</p>	R/W

表 47 PHY_INTR_FLAGS 寄存器描述(PHY_INTR_FLAGS 0xFCH)

5.5 定时器

5.5.1 概述

HT8550内部有三个16位可编程的定时器，定时器T0、定时器T1和定时器T2。

每一个定时器都由两个独立的8位寄存器组成：

Timer0: TL0和TH0

Timer1: TL1和TH1

Timer2: TL2和TH2

定时器0和定时器1都有4种工作模式，由TMOD和TCON控制。定时器的时钟为系统时钟。这四种工作模式是：

- 模式0: 13位定时器
- 模式1: 16位定时器
- 模式2: 自动装入时间常数的8位定时器
- 模式3: 两个8bit定时器（只对Timer0有效）

5.5.2 定时器 0 详细说明

在定时器模式中，定时器0每12个时钟周期加1，在每12个周期的时钟信号之后，它的计数会加1。

5.5.2.1 模式 0

设置寄存器“tmod”的标志位tmod[1:0]=00可进入该模式。

在这种模式中，要使定时器工作，需要清除标志位tmod.2，寄存器中记录的是定时的输入时钟数。

定时器作为13位的寄存器（取“tl0”的低5位“th0”所有位）。“tl0”的高3位无效。这就使定时器成为一个13位的counter，每12个时钟周期加1。当定时器溢出时，“tcon.5”置位，同时产生定时器溢出中断。程序进入中断后，该位被自动清零。定时器可被软件/硬件控制。标志位“tcon.4”可以控制定时器的停止（清0）和运行（置1）。

5.5.2.2 模式 1

设置寄存器“tmod”的标志位tmod[1:0]=01可进入该模式。

模式1与模式0的唯一区别就是低字节寄存器不再被分为低5位和高3位两部分，整个低字节都用于计数器。在模式1中，定时器是一个16位的计数器。

5.5.2.3 模式 2

设置寄存器“tmod”的标志位tmod[1:0]=10可进入该模式。

在这种模式中，要使定时器工作，需要清除标志位tmod.2，寄存器中记录的是定时的输入时钟数。

在这种模式下，只有低字节（“tl0”）会每12个时钟周期加1。

在这种模式下，定时器是一个8位的重载定时器。当定时器0溢出时，“tcon.5”置位，同时产生定时器溢出中断。程序进入中断后，该位被自动清零。

当溢出发生时，新的值将会从高字节（“th0”）载入到低字节（“tl0”）中。

定时器可被软件或者硬件控制。标志位“tcon.4”可以控制定时器的停止（清0）和运行（置1）。

5.5.2.4 模式3

设置寄存器“tmod”的标志位tmod[1:0]=11可进入该模式。

在这种模式中，要使定时器工作，需要清除标志位tmod.2，寄存器中记录的是定时的输入时钟数。

在这种模式下，低字节（“tl0”）会每12个时钟周期加1。高字节（“th0”）会每12个时钟周期加1。

当定时器低字节0溢出时，“tcon.5”置位，同时产生定时器溢出中断。程序进入中断后，该位被自动清零。当定时器的高字节溢出时，“tcon[7]”置位，同时产生定时器溢出中断。程序进入中断后，该位被自动清零。

在这种模式中，定时器0的低字节受“tcon.4”控制。高字节受“tcon.6”控制，当“tcon[6]”置位时，高字节开始计数，当“tcon[4]”置位时，低字节开始计数。

5.5.3 定时器1 详细说明

在定时器模式中，定时器1每12个时钟周期加1，就是说，在每12个周期的时钟信号之后，它的计数会加1。

5.5.3.1 模式0

设置寄存器“tmod”的标志位tmod[5:4]=00可进入该模式。

在这种模式中，要使定时器工作，需要清除标志位tmod.6，寄存器中记录的是定时的输入时钟数。

定时器1被分为两个8位寄存器，低字节和高字节；低字节又被分为两部分：低5位和高3位。这就使定时器1成为一个13位的counter，每12个时钟周期加1。当定时器1溢出时，“tcon.7”置位，同时产生定时器溢出中断。程序进入中断后，该位被自动清零。定时器可被软件或者硬件控制。标志位“tcon.6”可以控制定时器的停止（清0）和运行（置1）。

5.5.3.2 模式1

设置寄存器“tmod”的标志位tmod[5:4]=01可进入该模式。

模式1与模式0的唯一区别就是低字节寄存器不再被分为低5位和高3位两部分，在模式1中，定时器1是一个16位的计数器。

5.5.3.3 模式2

设置寄存器“tmod”的标志位tmod[5:4]=10可进入该模式。

在这种模式中，要使定时器工作，需要清除标志位tmod.6，寄存器中记录的是定时的输入时钟数。

在这种模式下，只有低字节 (“t1”) 会每12个时钟周期加1

在这种模式下，定时器1是一个8位的重载定时器。当低字节溢出时，“tcon[7]”置位，同时产生定时器溢出中断。程序进入中断后，该位被自动清零。

当溢出发生时，新的值将会从高字节 (“th1”) 载入到低字节 (“t1”) 中。

定时器1可被软件控制。标志位 “tcon.6” 可以控制定时器的停止 (清0) 和运行 (置1)。定时器可被软件或者硬件控制。标志位 “tcon.6” 可以控制定时器的停止 (清0) 和运行 (置1)。

5.5.3.4 模式3

设置寄存器 “tmod” 的标志位 tmod[5: 4]=11可进入该模式。

在这种模式下，定时器1被关闭 (只有定时器0可在模式3下操作)。

5.5.4 定时器 2 详细说明

Timer2 是一个 16bit 的定时器/计数器。

5.5.4.1 定时器功能

a) 定时器模式

设置寄存器 T2CON 的标志位 t2i0=1 和 t2i1=0 进入该模式。定时器 2 有两个频率的时钟可以选择，当标志位 t2ps=0 时，每 12 个时钟周期 count 加 1；当标志位 t2ps=1 时，每 24 个时钟周期 count 加 1。

b) 门控定时器模式

设置寄存器 T2CON 的标志位 t2i0=1 和 t2i1=1 进入该模式。与模式 a)相同，该模式中，有两个频率的时钟可以选择，每 12 或者 24 个时钟周期 count 加 1；同时受外部信号 t2 的控制。当 t2=0 时，Timer2 停止。

c) 重新载入

当 Timer2 溢出时，TH2/TL2 自动从 CRC 寄存器中载入 16 位数据。

5.5.5 特殊功能寄存器

由于外部计数管脚拉低并没有引出来，所以不存在计数器功能，但保留了相应寄存器。

地址	名称	复位值	功能简介
0x88	TCON	0x00	定时器控制寄存器
0x89	TMOD	0x00	定时器方式寄存器
0x8A	TL0	0x00	定时器 0 低 8 位
0x8B	TL1	0x00	定时器 1 低 8 位
0x8C	TH0	0x00	定时器 0 高 8 位
0x8D	TH1	0x00	定时器 1 高 8 位
0xC8	T2CON	0x40	定时器 2 控制寄存器
0xCC	TL2	0x00	定时器 2 低 8 位
0xCD	TH2	0x00	定时器 2 高 8 位

表 48 定时器 0 和定时器 1,定时器 2 寄存器列表

5.5.5.1 TCON

位	位名称	复位值	描述	属性
---	-----	-----	----	----

7	TF1	0	定时器 1 溢出中断标志 当定时器 1 溢出时，由硬件置位。 当进入中断处理程序后，由硬件自动清除；也可以由软件清除。	R/W
6	TR1	0	定时器 1 运行控制位 清除该位，定时器 1 将停止运行。	R/W
5	TF0	0	定时器 0 溢出中断标志 当定时器 0 溢出时，由硬件置位。 当进入中断处理程序后，由硬件自动清除；也可以由软件清除。	R/W
4	TR0	0	定时器 0 运行控制位 清除该位，定时器 0 将停止运行。	R/W
3	IE1	0	外部中断 1 标志 当外部中断 EX1 产生时（电平触发或沿触发），由硬件置位。 当进入中断处理程序后，由硬件清除。	R/W
2	IT1	0	外部中断 1 类型控制位 TCON.2=1：外部中断 1 由输入引脚的下降沿触发； TCON.2=0：外部中断 1 由输入引脚的低电平触发；	R/W
1	IE0	0	外部中断 0 标志 当外部中断 EX0 产生时（电平触发或沿触发），由硬件置位。 当进入中断处理程序后，由硬件清除。	R/W
0	IT0	0	外部中断 0 类型控制位 TCON.0=1：外部中断 0 由输入引脚的下降沿触发； TCON.0=0：外部中断 0 由输入引脚的低电平触发；	R/W

表 49 定时器 0、定时器 1 控制寄存器 (TCON 0x88H)

5.5.5.2 TMOD

位	位名称	复位值	功能描述	属性
7	GATE	0	未使用	R/W
6	C/\bar{T}	0	定时器 1 方式选择位 $C/\bar{T}=0$ ：定时器方式。	R/W
5	M1	0	定时器 1 模式控制位 如下表所示。	R/W
4	M0	0		R/W
3	GATE	0	未使用	R/W
2	C/\bar{T}	0	定时器 0 方式选择位 $C/\bar{T}=0$ ：定时器方式。	R/W
1	M1	0	定时器 0 模式控制位	R/W

0	M0	0	如下表所示。	R/W
---	----	---	--------	-----

M1	M0	方式	说明
0	0	0	13 位定时器, TLO (TL1) 中的低 5 位和 TH0(TH1)中的 8 位。
0	1	1	16 位定时器。
1	0	2	8 位定时器, 具有自动再装入功能, 装入在 TLO(TL1)溢出时发生, 装入的值由 TH0(TH1)提供。
1	1	3	定时器 1 停止操作, 定时器 0 分成两个 8 位的独立计数器: TLO 使用控制位 tr0, 溢出时将 tf0 置位; TH0 使用控制位 tr1, 溢出时将 tf1 置位。

表 50 定时器 0、定时器 1 模式寄存器(TMOD 0x89H)

5.5.5.3 TH1

位	位名称	复位值	功能描述	属性
7:0	-	0	该寄存器反映了 Timer 1 的状态, 作为定时器 1 状态寄存器的高字节。该地址被分配在 SFR 存储空间。	R/W

表 51 定时器 1 状态寄存器(TH1 0x8DH)

5.5.5.4 TL1

位	位名称	复位值	功能描述	属性
7:0	-	0	该寄存器反映了 Timer 1 的状态, 作为定时器 1 状态寄存器的低字节。该地址被分配在 SFR 存储空间。	R/W

表 52 定时器 1 状态寄存器(TL1 0x8BH)

这两个寄存器存储了定时器1的状态值, TH1存储高字节, TL1存储低字节。

5.5.5.5 TH0

位	位名称	复位值	功能描述	属性
7:0	-	0	该寄存器反映了 Timer 0 的状态, 作为定时器 0 状态寄存器的高字节。该地址被分配在 SFR 存储空间。	R/W

表 53 定时器 0 状态寄存器(TH0 0x8CH)

5.5.5.6 TL0

位	位名称	复位值	功能描述	属性
---	-----	-----	------	----

7:0	-	0	该寄存器反映了 Timer 0 的状态，作为定时器 0 状态寄存器的低字节。该地址被分配在 SFR 存储空间。	R/W
-----	---	---	---	-----

表 54 定时器 0 状态寄存器(TL0 0x8AH)

这两个寄存器存储了定时器0的状态值，TH0存储高字节，TL0存储低字节。

5.5.5.7 T2CON

位	位名称	复位值	功能描述	属性
7	T2PS	0	时钟选择位 t2ps = 0 – 定时器 2 的时钟选择为系统时钟的 1/12; t2ps = 1 – 定时器 2 的时钟选择为系统时钟的 1/24;	R/W
6	I3FR	1	外部中断 “int3” 触发方式选择 0 – 下降沿; 1 – 上升沿;	R/W
5	I2FR	0	外部中断 “int2” 触发方式选择 0 – 下降沿; 1 – 上升沿;	R/W
4:3	-	0	保留。	R
2	T2CM	0	保留。	R/W
1	T2I1	0	定时器 2 输入选择(t2i1, t2i0) 00—定时器 2 中止工作; 01—输入时钟为 fsys/12 或者 fsys/24; 10—定时器 2 的计数由引脚 “t2” 的下降沿驱动; 11—输入时钟由引脚 “t2” 确定为 fsys/12 或者 fsys/24;	R/W
0	T2I0	0		R/W

表 55 定时器 2 控制寄存器 (T2CON 0xC8H)

5.5.5.8 TH2

位	位名称	复位值	功能描述	属性
7:0	-	0	该寄存器反映了 Timer 2 的状态，作为定时器 2 状态寄存器的高字节。该地址被分配在 SFR 存储空间。	R/W

表 56 定时器 2 状态寄存器(TH2 0xCDH)

5.5.5.9 TL2

位	位名称	复位值	功能描述	属性
---	-----	-----	------	----

7:0	-	0	该寄存器反映了 Timer 2 的状态，作为定时器 2 状态寄存器的低字节。该地址被分配在 SFR 存储空间。	R/W
-----	---	---	---	-----

表 57 定时器 2 状态寄存器(TL2 0xCCH)

这两个寄存器存储了定时器 2 的状态值，TH2 存储高字节，TL2 存储低字节。

5.6 GPIO

HT8550包含22个双向GPIO，并可以将GPIO作为外部中断输入。其中部分GPIO可以被复用为I²C、SPI接口或其他测试接口。

Pin No.	Pin Name	Function	GPIO	Linked to MCU I/O port	External Interrupt
37	GP_0	General Purpose Input (default) General Purpose Output External Interrupt	GPIO port 0	Port0.0	Ext.Intr[0]
46	GP_1	General Purpose Input (default) General Purpose Output External Interrupt	GPIO port 1	Port0.1	Ext.Intr[1]
2	GP_2	Func.0: Reserved Func.1: PDM output General Purpose Input (default) General Purpose Output External Interrupt	GPIO port 2	Port0.2	Ext.Intr[2]
4	GP_3	General Purpose Input (default) General Purpose Output External Interrupt	GPIO port 3	Port0.3	Ext.Intr[3]
22	GP_4	General Purpose Input (default) General Purpose Output External Interrupt	GPIO port 4	Port0.4	Ext.Intr[4]
31	CLKOUT	Func.0: MAC clock detect port (default) Func.1: Reserved General Purpose Input General Purpose Output External Interrupt	GPIO port 5	Port0.5	Ext.Intr[5]
32	WDTEN	Func.0: Watch dog enable (default) Func.1: Reserved General Purpose Input General Purpose Output External Interrupt	GPIO port 6	Port0.6	Ext.Intr[6]
3	TX_BUSY	Func.0: TX busy flag (default) Func.1: Reserved General Purpose Input General Purpose Output External Interrupt	GPIO port 7	Port0.7	Ext.Intr[7]

47	UART0_RX	Func.0: reserved (default) Func.1: R8051 UART0 RX General Purpose Input General Purpose Output External Interrupt	GPIO port 8	Port1.0	Ext.Intr[0]
48	UART0_TX	Func.0: reserved (default) Func.1: R8051 UART0 TX General Purpose Input General Purpose Output External Interrupt	GPIO port 9	Port1.1	Ext.Intr[1]
27	FLSH_SO	Func.0: Flash serial data output (default) Func.1: Reserved General Purpose Input General Purpose Output External Interrupt	GPIO port 10	Port1.2	Ext.Intr[2]
28	FLSH_SCLK	Func.0: Flash clock (default) Func.1: Reserved General Purpose Input General Purpose Output External Interrupt	GPIO port 11	Port1.3	Ext.Intr[3]
29	FLSH_CS_N	Func.0: Flash CS_N (default) Func.1: Reserved General Purpose Input General Purpose Output External Interrupt	GPIO port 12	Port1.4	Ext.Intr[4]
30	FLSH_SI	Func.0: Flash serial data input (default) Func.1: Reserved General Purpose Input General Purpose Output External Interrupt	GPIO port 13	Port1.5	Ext.Intr[5]
24	UART1_RX	Func.0: R8051 UART1 RX (default) Func.1: reserved General Purpose Input General Purpose Output External Interrupt	GPIO port 14	Port1.6	Ext.Intr[6]
23	UART1_TX	Func.0: R8051 UART1 TX (default) Func.1: reserved General Purpose Input General Purpose Output External Interrupt	GPIO port 15	Port1.7	Ext.Intr[7]
44	I2C_SDA	Func.0: R8051 I2C SDA (default) Func.1: Reserved General Purpose Input General Purpose Output External Interrupt	GPIO port 16	Port2.0	Ext.Intr[0]
45	I2C_SCL	Func.0: R8051 I2C SCL (default) Func.1: Reserved	GPIO port 17	Port2.1	Ext.Intr[1]

		General Purpose Input General Purpose Output External Interrupt			
41	SPI_SCLK	Func.0: Reserved (default) Func.1: R8051 SCLK General Purpose Input General Purpose Output External Interrupt	GPIO port 18	Port2.2	Ext.Intr[2]
40	SPI_SOUT	Func.0: Reserved (default) Func.1: R8051 SOUT General Purpose Input General Purpose Output External Interrupt	GPIO port 19	Port2.3	Ext.Intr[3]
43	SPI_SIN	Func.0: Reserved (default) Func.1: R8051 SIN General Purpose Input General Purpose Output External Interrupt	GPIO port 20	Port2.4	Ext.Intr[4]
42	SPI_CS_N	Func.0: Reserved (default) Func.1: R8051 CS_N General Purpose Input General Purpose Output External Interrupt	GPIO port 21	Port2.5	Ext.Intr[5]

表 58 GPIO 定义

为配置22个GPIO的功能，HT8550内定义了2个SFR：GPIO_MODE_SEL和GPIO_CTRL。HT8550的22个GPIO每2个一组，被分为11组，各组被编址为0x0~0xA。分组及编址情况由寄存器GPIO_MODE_SEL定义。需要配置某个GPIO时，先通过配置GPIO_MODE_SEL选中该GPIO所在的组，再通过GPIO_CTRL配置该GPIO的功能。实际上，对应每一组GPIO，GPIO_CTRL都有一个备份，即对应GPIO_MODE_SEL所标示的11个地址，GPIO_CTRL 共有11个备份。当某一个GPIO分组地址被设置后，对GPIO_CTRL的修改，将改变该组所对应的GPIO的功能。GPIO_CTRL[7:4]用于控制已选中的组中编号为奇数的GPIO，GPIO_CTRL[3:0]用于控制已选中的组中编号为偶数的GPIO。

例如，设置GPIO_MODE_SEL = 0xA，将选中GPIO port 21(pin42)和GPIO port 20(pin43)。然后，设置GPIO_CTRL = 0x52，则pin43被设置为general purpose output pin，pin42被设置为R8051 SPI CS_N。

5.6.1 特殊功能寄存器

GPIO寄存器列表

地址	名称	复位值	功能简介
0xF9	GPIO_INTR_POLARITY	0x0	GPIO 中断极性选择
0xFE	GPIO_MODE_SEL	0x0	GPIO port 选择
0xFF	GPIO_CTRL	0x0	GPIO 模式选择

表 59 GPIO 寄存器列表

5.6.1.1 GPIO_INTR_POLARITY

位	位名称	复位值	描述	属性
7:0	-	0x00	GPIO 中断极性选择 Bit[n]=1: 外部中断 n 低有效; Bit[n]=0: 外部中断 n 高有效;	R/W

表 60 GPIO 中断极性选择寄存器(GPIO_INTR_POLARITY 0xF9H)

5.6.1.2 GPIO_MODE_SEL

位	位名称	复位值	描述	属性
7:0	-	00	GPIO port 选择 在进行 GPIO 功能配置前, 须先选中期望配置的 GPIO, 使其进入功能可配置的状态。 0x0: GPIO port 1 和 GPIO port 0 0x1: GPIO port 3 和 GPIO port 2 0x2: GPIO port 5 和 GPIO port 4 0x3: GPIO port 7 和 GPIO port 6 0x4: GPIO port 9 和 GPIO port 8 0x5: GPIO port 11 和 GPIO port 10 0x6: GPIO port 13 和 GPIO port 12 0x7: GPIO port 15 和 GPIO port 14 0x8: GPIO port 17 和 GPIO port 16 0x9: GPIO port 19 和 GPIO port 18 0xA: GPIO port 21 和 GPIO port 20 Others: 无效, 不选中任何 GPIO	R/W

表 61 GPIO port 选择寄存器(GPIO_MODE_SEL 0xFEH)

5.6.1.3 GPIO_CTRL

位	位名称	复位值	描述	属性
7:0	-	[0x0]: 0x00* [0x1]: 0x00 [0x2]: 0x40 [0x3]: 0x44 [0x4]: 0x44 [0x5]: 0x44 [0x6]: 0x04 [0x7]: 0x00 [0x8]: 0x00 [0x9]: 0x44 [0xA]: 0x44	GPIO 模式选择 [7:4] 控制已选中组中编号为奇数的 GPIO [3:0] 控制已选中组中编号为偶数的 GPIO Control Decode: 0000 : General purpose Input 0010 : General purpose Output 0100 : Function 0(Func.0) 0101 : Function 1(Func.1) 1000 : External Interrupt	R/W

表 62 GPIO 模式选择寄存器(GPIO_CTRL 0xFFH)

6 其他外设

HT8550的其他功能有:

- FLASH控制器
- SPI接口
- UART接口
- I2C接口
- JTAG接口
- RTC
- 看门狗

6.1 FLASH 控制器

6.1.1 Flash 控制器简介

HT8550包含一个专用SPI接口，用来与片外SPI FLASH相连，串行传输时钟最高为48MHz。

如果接口速率超过24MHz，需要设置FLASH_CTRL寄存器的SLOW MISO和FAST READ COMMAND两个功能位。

如需使用外部Flash，需要通过配置寄存器GPIO_CTRL实现。配置过程如下：

- (1) 设置GPIO_MODE_SEL = 0x5
- (2) 设置GPIO_CTRL[7:0] = 0x44
- (3) 设置GPIO_MODE_SEL = 0x6
- (4) 设置GPIO_CTRL[7:0] = 0x44

6.1.2 FLASH 作为程序扩展

通过程序跳转到外部Flash映射的地址，HT8550可以从外部Flash执行。

从Flash中读取数据会失速，所以不需要将HT8550的频率运行的比外部Flash时钟慢。但对节能模式来说，这是推荐的。

接口是串行的，最好的方式是每八个Flash时钟读一个byte。如果Flash的时钟快过HT8550的八倍，HT8550在线性代码执行的时候性能不会降低。由于代码分支将会冲掉预缓存，因此需要一个新的读命令发给Flash。这个过程HT8550自己会处理，但将影响HT8550的性能。

由于HT8550从外部Flash中读取内容时会将数据与寄存器FLASH_MEMORY_XOR的值异或，因此写入Flash之前需要先将程序内容与FLASH_MEMORY_XOR的值异或处理。

6.1.3 特殊功能寄存器

Flash 控制相关寄存器如下：

地址	名称	复位值	功能简介
0x8F	FLASH_CTRL	0x00	Flash 控制寄存器
0x91	FLASH_DATA	0x00	Flash 数据寄存器
0x96	FLASH_STATUS	0x00	Flash 状态寄存器
0xAB	FLASH_COPY	0x00	Flash 复制寄存器
0xBE	FLASH_COPY_PAGES	0x80	Flash 页复制寄存器
0xD5	Flash Memory XOR	0x7B	Flash 异或寄存器

地址	名称	复位值	功能简介
0xD7	Program Memory Lock	0x00	存储编程锁定寄存器

表 63 FLASH 控制器寄存器列表

6.1.3.1 FLASH_CTRL

位	位名称	复位值	描述	属性
7	-	0	存取模式: 0 = 只读程序存储器模式: 外部 Flash 作为只读程序存储器 这是启动后的默认值, HT8550 在这个模式下可以从外部 Flash 执行程序或者使用 DMA 从 Flash 移动数据。该模式下不支持写操作。 1 = SPI 主模式. Flash 作为标准的 SPI 从设备, 使用 FLASH_DATA 寄存器给 Flash 发送或者接收数据。 这个模式用来设置, 删除或者写 Flash。 在这个模式下, DMA 和目标读取无效。	R/W
6	-	0	片选: 仅用在 SPI 主模式。用户在使能之前必须先检查主模式标志位 0 = 不选(CS 输出高); 1 = 选中(CS 输出低);	R/W
5	-	0	快速读取模式 在 DMA 或目标模式中, 这条指令接口发送 FAST_READ 给外部 FLASH。当串行时钟运行在高频率时, 这个是必需设置。	R/W
4	-	0	时钟极性 (CPOL)。	R/W
3	-	0	时钟相位 (CPHA)。	R/W
2	-	0	快读设置使能 (Slow MISO): 当使能时, 串行输入数据在发射沿采样一个完整周期。当 Flash 时钟运行在高频率时, 该位需要设置。	R/W
1	-	0	保留;	R
0	-	0	内部 Flash 写使能: 为了阻止内部 Flash 被意外写, 如果需要写或删除内部 Flash 内容, 需要将该位置 1。	R/W

表 64 Flash 控制寄存器(FLASH_CTRL 0x8FH)

6.1.3.2 FLASH_DATA

位	位名称	复位值	描述	属性
7:0	-	0	需要传输的数据写入这个寄存器, 并且写入的数据被复制到移位寄存器中, 同时输入的数据也从这里获得。	R/W

表 65 Flash 数据寄存器(FLASH_DATA 0x91H)

6.1.3.3 FLASH_STATUS

位	位名称	复位值	描述	属性
7:2	-	0	保留	R
1	-	0	主模式标志 在设置存取模式后被硬件置位。在片选和写数据到 FLASH_DATA 寄存器之前，该位必须检测到为“1”。同时，这个寄存器在接收到第一个非主模式请求之前不会被清除。	R
0	-	0	主模式传输完成标志 在 SPI 主模式下字节传输完成后由硬件置位。 通过读 FLASH DATA 寄存器清除该位。 注意：为了清除内部数据 FIFO，每次传输后必需读取 FLASH DATA 寄存器。	R

表 66 Flash 状态寄存器(FLASH_STATUS 0x96H)

6.1.3.4 FLASH_COPY

位	位名称	复位值	描述	属性
7:0	-	0x0	Flash 复制 . 该寄存器写"x86" 将会复制整个外部 Flash 内容到内部 Flash, 这可以用来升级当前内部 Flash 的代码。 复制完成后，芯片重启。 为了起作用，在写该寄存器之前，内部 Flash 写使能必需打开。 当程序存储锁定时，只有芯片本身可以写这个寄存器。 该寄存器读的值一直是 0。	R/W

表 67 Flash 复制寄存器(FLASH_COPY 0xABH)

6.1.3.5 FLASH_COPY_PAGES

位	位名称	复位值	描述	属性
7:0	-	0x80	从外部 Flash 复制大小为 1KB 的页内容到内部 Flash。这个寄存器可以设置成小于 x80 以加速复制速度。不管该寄存器值是多少，整个内部 flash 都会被删除。	R/W

表 68 Flash 页复制寄存器(FLASH_COPY_PAGES 0xBEH)

6.1.3.6 Flash Memory XOR

位	位名称	复位值	描述	属性
---	-----	-----	----	----

7:0	-	0x7B	所有从外部 Flash 传入的数据都会跟这个寄存器的值异或。这些包括启动时数据传输，取指令，DMA 传输，以及主模式下任何的读取。这个寄存器的值在启动以后能够被改变，但内容不能被读取。	W
-----	---	------	--	---

表 69 Flash 异或寄存器(Flash Memory XOR 0xD5H)

6.1.3.7 Program Memory Lock

位	位名称	复位值	描述	属性
7:0	-	0x0	1=锁定；0 = 不锁定；	R/W

表 70 Flash 存储编程锁定寄存器(Program Memory Lock 0xD7H)

6.2 SPI 接口

6.2.1 概述

SPI接口可以实现在MCU和外围设备（包含MCU）之间的全双工同步串行通讯。这里提到的MCU或者外围设备必须包含SPI模块。

SPI模块可以被编程实现以主模式或从模式工作。包含下列特征：

- 全双工模式
- 三线同步传输
- 主机和从机模式
- 7种主机波特率
- 从机时钟最高至fs/4
- 极性和相位可编程的串行时钟
- 写冲突处理机制
- 8位数据传输，高字节在前，低字节在后
- 8位从机选择接口，控制外部从机
- 与主机MCU的专用功能寄存器接口
- 无二义端口，标准的SPI

GPIO配置为SPI功能：

- (1) 设置GPIO_MODE_SEL = 0x9
- (2) 设置GPIO_CTRL = 0x55
- (3) 设置GPIO_MODE_SEL = 0xA
- (4) 设置GPIO_CTRL = 0x55

在8051-SPI模式下，片内MCU可以通过配置SFR值来改变8051-SPI接口的主/从模式，以及时钟极性与相位。

下图显示了数据传输的帧格式。根据SPI的设置，数据的每一位在主时钟（“scko”）的上升沿（“cpol”=’0’）或下降沿（“cpol”=’1’）被发送。数据在主时钟（“scko”）的下降沿（“cpol”

=’0’) 或上升沿 (“cpol” =’1’) 被接收。这适用于主模式或从模式的发送器/接收器，前提是 “scko” 是传输过程中的主时钟。如果 “cpha” 被置位，第一位 (MSB) 将在 “scko” 的第一个动态沿时通过 “mosio” / “misoo” 被发送。如果 “cpha” 被清零，第一位 (MSB) 将在 “scko” 的第一个动态沿之前半个周期被发送。

除此之外，输入数据 (主模式是 “misoi”，从模式是 “mosii”) 在每一位传输一半时被采样，在这个时钟周期的相反的电平上，数据被移位到输出信号 “mosio” 上。

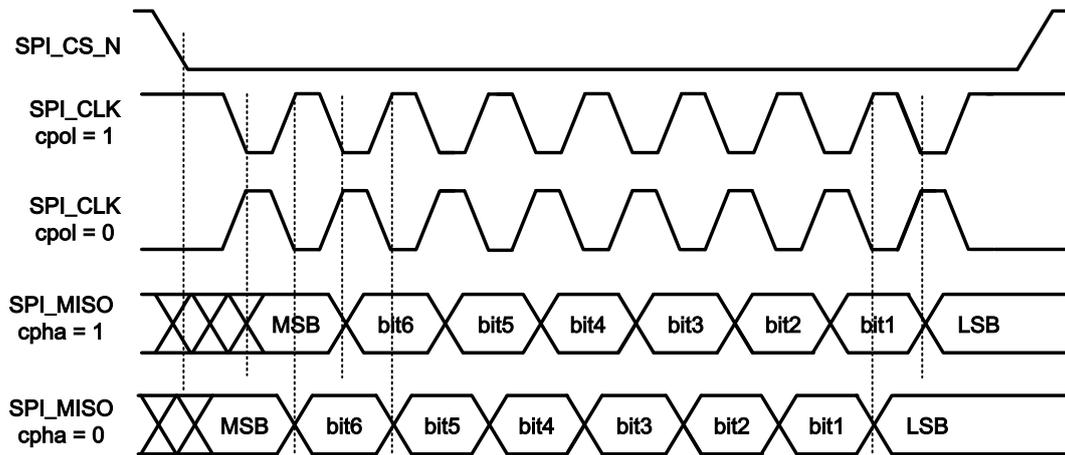


图 5 SPI 数据输出帧格式

6.2.2 主机模式

SPI默认为主机模式。

在主机模式中，SPI等待程序向寄存器SPDAT中写入数据。如果向SPDAT的写入动作完成，传输就开始。在时钟SPI_CLK的发送沿，数据被移位到输出引脚SPI_MOSI上。

6.2.3 从机模式

首先，需要写寄存器SPCON中的mstr=0，以配置8051-SPI进入从机模式。另外配置spen=1以打开8051-SPI模块使能。

在从机模式中，SPI等待输入信号SPI_CS_N的低电平。当捕捉到SPI_CS_N的下降沿，传输开始，直到传输完成，SPI_CS_N都需要保持低电平状态。寄存器SPCON中cpha的状态决定传输的开始位置：当cpha被清零，从机必须在 SPI_CLK信号的第一个下降沿之前开始传输；当cpha被置位，从机会把SPI_CLK信号的第一个下降沿做为传输的开始标志。

6.2.4 特殊功能寄存器

SPI控制相关寄存器如下：

地址	名称	复位值	功能简介
0xE1	SPSTA	00H	SPI 状态寄存器
0xE2	SPCON	14H	SPI 控制寄存器
0xE3	SPDAT	00H	SPI 数据寄存器
0xE4	SPSSN	FFH	SPI 从机选择寄存器

表 71 SPI 相关寄存器列表

6.2.4.1 SPSTA

位	位名称	复位值	功能描述	属性
7	spif	0	数据传输标志 当传输完成时由硬件置位；传输过程中由硬件复位，也可通过读寄存器“spsta”“spdat”来复位；	R/W
6	wcol	0	写冲突标志 当写“spdat”冲突时由硬件置位；当传输完成无冲突发生时由硬件复位，也可通过访问寄存器“spsta”“spdat”复位；	R/W
5	sserr	0	同步从机错误标志 在接收完成前当“ssn”输入有效时，被硬件置位；关闭 SPI 模块可清除该位（设置spen=0）；	R/W
4	modf	0	模式故障标志 当“ssn”引脚状态与设置的模式有冲突时，硬件自动置位；当“ssn”引脚恢复合适的电平状态时，硬件自动复位；也可以由软件读“spsta”寄存器来复位；	R/W
3:0	-	0	保留。	R

表 72 SPI 控制寄存器(SPSTA 0xE1H)

6.2.4.2 SPCON

位	位名称	复位值	功能描述	属性
7	spr2	0	SPI 时钟速率控制位 主模式时，SPI 时钟速率由“spr2”“spr1”“spr0”控制。	R/W
6	spen	0	SPI 使能位 spen=0，关闭 spi 模块；spen=1，打开 spi 模块；	R/W
5	ssdis	0	SS 控制位 ssdis=0：在主/从模式中打开“ssn”输入； ssdis=1：在主/从模式中关闭“ssn”输入，该情况下不会产生“modf”中断请求；在从机模式中，若“cpha”=0，则该位无效；	R/W
4	mstr	0	SPI 模式选择位 mstr=0：从机模式 mstr=1：主机模式	R/W
3	cpol	0	时钟极性 cpol=0：“sck”在空闲状态时被设置为低电平； cpol=1：“sck”在空闲状态时被设置为高电平；	R/W
2	cpha	0	时钟相位	R/W
1	spr1	0	SPI 时钟速率控制位	R/W

0	spr0	0	主模式时，SPI 时钟速率由“spr2”“spr1”“spr0”控制			
			spr2	spr1	spr0	SPI 时钟速率
			0	0	0	Fsys/2
			0	0	1	Fsys/4
			0	1	0	Fsys/8
			0	1	1	Fsys/16
			1	0	0	Fsys/32
			1	0	1	Fsys/64
			1	1	0	Fsys/128
1	1	1	不产生主时钟			

表 73 SPI 状态寄存器(SPCON 0xE2H)

6.2.4.3 SPDAT

位	位名称	复位值	功能描述	属性
7:0	-	0	寄存器 SPDAT 是“接收数据”寄存器的一个读/写缓冲。当向 SPDAT 中写入数据，是直接写入移位寄存器中（没有传输缓冲）；从 SPDAT 中读数据，返回的是接收缓冲中的数据，而非移位寄存器。	R/W

表 74 SPI 数据寄存器(SPDAT 0xE3H)

6.2.4.4 SPSSN

位	位名称	复位值	功能描述	属性
7:0	-	0xFF	SPSSN 是一个可读/写寄存器，它的每一位都可用来选择独立的外部 SPI 从机设备。	R/W

表 75 SPI 从机选择寄存器(SPSSN 0xE4H)

6.3 I2C 接口

6.3.1 概述

I2C 模块提供一个符合标准 I2C 总线规范的串行接口，可用于 HT8550 片内 MCU 与片外器件通信。该 I2C 接口可配置为主或从模式。I2C 数据以 8-bit 进行双向传输，标准模式下可达 100kbps 的传输速率，快速模式可达 400kbps 的速率。I2C 引脚 I2C_SCL (pin1) 和 I2C_SDA (pin3) 默认功能为 GPIO，如需使用 I2C 总线，需要通过配置寄存器 GPIO_CTRL 实现。配置过程如下：

- (1) 设置 GPIO_MODE_SEL = 0x8
- (2) 设置 GPIO_CTRL[7:0] = 0x44

PARAMETER	SYMBOL	CONDITIO NS	MIN MAX	TYP	UNIT S
Clock High to Input Transition	T _{CHDX}		4.7		μs
Clock Pulse Width High	T _{CHCL}		4		μs
Input Low to Clock Low (START)	T _{DLCL}		4		μs

Clock Low to Input Transition	T_{CLDX}		0	μs
Clock Pulse Width Low	T_{CLCH}		4.7	μs
Input Transition to Clock Transition	T_{DXCX}		250	ns
Clock High to Input High(STOP)	T_{CHDH}		4.7	μs
Input High to Input Low(Bus Free)	T_{DHDL}		4.7	μs
Clock Low to Next Data Out Valid	T_{CLQV}		0.3	3.5 μs
Data Out Hold Time	T_{CLQX}		300	ns
Clock Frequency	F_C		400	kHz

表 76 I2C 时序特征

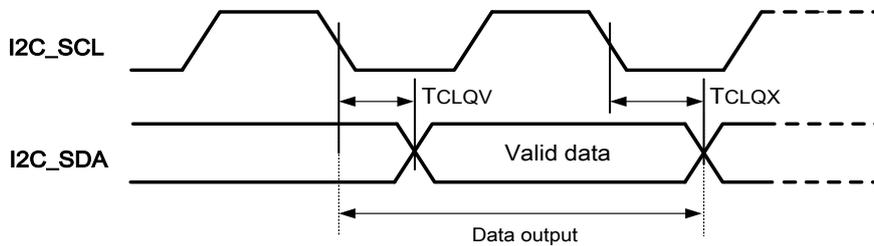


图 6 I2C 输出时序

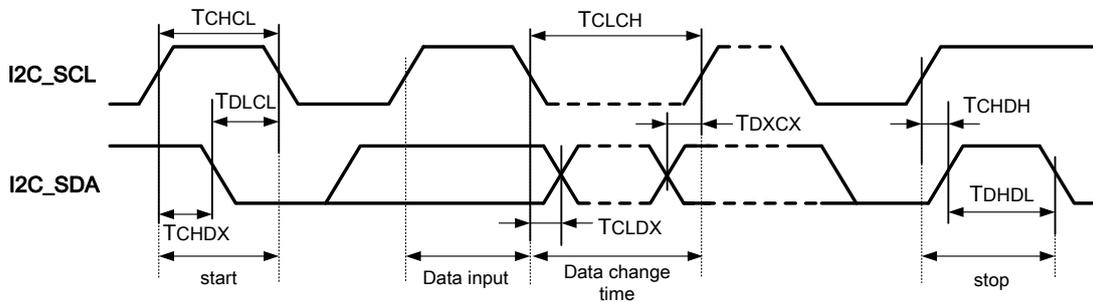


图 7 I2C 输入时序

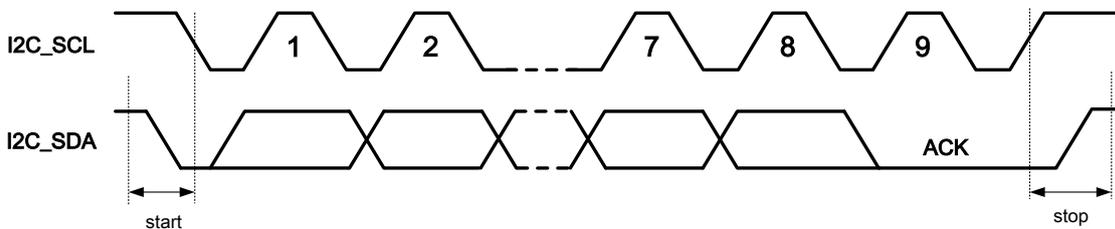


图 8 I2C 协议

6.3.2 特殊功能寄存器

片内MCU可以通过以下4个SFR: I2CCON、I2CSTA、I2CDAT、I2CADR对I2C接口进行

配置。

地址	名称	复位值	功能简介
0xDA	I2CDAT	0x00	数据寄存器
0xDB	I2CADR	0x00	自身从机地址寄存器
0xDC	I2CCON	0x00	控制寄存器
0xDD	I2CSTA	0xF8	状态寄存器

表 77 I2C 寄存器列表

6.3.2.1 I2CDAT

位	位名称	复位值	描述	属性
7:0	-	0	寄存器 I2CDAT 包含一个字节的数 据，是将要被传送到总线上的数 据，或者是刚从总线上接收到的数 据。寄存器 I2CDAT 没有设置影子寄 存器，也没有双缓存，所以当 I2C 中断发生时，MCU 需要及时从它 读取数据，以免数据丢失。	R/W

表 78 I2C 数据寄存器(I2CDAT 0xDAH)

6.3.2.2 I2CADR

位	位名称	复位值	描述	属性
7:1	-	0	I2C 从机地址 (7 位)	R/W
0	-	0	广播地址 如果该位置 1，则从设备响应广播 地址，否则，不响应广播地址。	R/W

表 79 I2C 地址寄存器(I2CADR 0xDBH)

6.3.2.3 I2CCON

位	位名称	复位值	描述	属性
7	CR2	0	Clock rate bit 2	R/W
6	ENS1	0	I2C 使能位 ens1=0: 关闭 IIC 模块; ens1=1: 打 开 IIC 模块;	R/W
5	STA	0	开始标志 sta=1: 检查 IIC 总线的状态, 如果 空闲则生成开始信号;	R/W
4	STO	0	停止标志 sto=1: 当处于主机模式, 则向总 线传输停止信号	R/W
3	SI	0	中断标志 当进入 25 种 IIC 状态之一时, “si” 由硬件置位, 唯一不置位的状态是 “F8h”; 该位必须被软件写“1”复 位。	R/W

2	AA	0	生成应答标志 aa=1 : 应答在以下情况下被返回: 接收到自身作为从机的地址; gc 被置位的情况下接收到地址呼叫; 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成; aa=0 : 非应答在以下情况下被返回: 主机接收模式下一个字节接收完成; 从机接收模式下一个字节接收完成;	R/W
1	CR1	0	Clock rate bit 1	R/W
0	CR0	0	Clock rate bit 0	R/W

CR2	CR1	CR0	Bit frequency				Clock divided by
			6MHz	12MHz	16MHz	24MHz	
0	0	0	23	47	63	92	256
0	0	1	27	54	71	108	224
0	1	0	31	63	83	124	192
0	1	1	37	75	100	148	160
1	0	0	6.25	12.5	17	25	960
1	0	1	50	100	133	200	120
1	1	0	100	200	266	400	60
1	1	1	bclk input divided by 8				

表 80 I2C 控制寄存器(I2CCON 0xDCH)

寄存器I2CCON包含I2C模块的使能位“ens1”，时钟频率控制位(“cr0”，“cr1”，“cr2”)，“开始”和“停止”信号的发送位，控制ACK信号的标志位“aa”，和中断标志位“si”。

6.3.2.4 I2CSTA

位	位名称	复位值	描述	属性
7	-	1	I2C 状态代号	R/W
6	-	1		R/W
5	-	1		R/W
4	-	1		R/W
3	-	1		R/W
2	-	0	保留，读的值为 0	R
1	-	0		R
0	-	0		R

表 81 I2C 状态寄存器(I2CSTA 0xDDH)

寄存器I2CSTA反映I2C模块的实时状态。这个寄存器的低三位始终为0。总共有26种可能的状态。当进入25种状态的其中一种时，都会产生中断；唯一一种不产生中断的情况是状态F8h。

在下表中，“SLA”指从机地址，“R”指与从机地址一起传送的读/写位是读，“W”指与从机地址一起传送的读/写位是写。

Status	Status of	Application	Software Response to	Next Action taken
--------	-----------	-------------	----------------------	-------------------

Code	I2C	to/from I2CDAT	I2CCON				by I2C Hardware
			sta	sto	si	aa	
08H	已发送 START 信号	Load SLA+W	X	0	0	X	发送 SLA+W 信号 接收 ACK 信号
10H	已发送 Repeated START 信号	Load SLA+W	X	0	0	X	同上
		Or Load SLA+R	X	0	0	X	发送 SLA+R 信号 I2C 将被切换到“master receiver”模式
18H	已发送 SLA+W 信号; 已接收 ACK 信号	Load data byte	0	0	0	X	发送 Data byte; 接收 ACK 信号
		or no action	1	0	0	X	发送 Repeated START 信号;
		or no action	0	1	0	X	发送 STOP 信号; 复位“sto”标志
		or no action	1	1	0	X	发送 STOP 信号后 接一个 START 信号; 复位“sto”标志
20H	已发送 SLA+W 信号; 已接收“not ACK”信号	Load data byte	0	0	0	X	发送 Data byte; 接收 ACK 信号
		or no action	1	0	0	X	发送 Repeated START 信号
		or no action	0	1	0	X	发送 STOP 信号; 复位“sto”标志
		or no action	1	1	0	X	发送 STOP 信号后 接一个 START 信号; 复位“sto”标志
28H	I2CDAT 中的 Data byte 已发送; 已接收 ACK 信号	Load data byte	0	0	0	X	发送 Data byte; 接收 ACK 信号
		or no action	1	0	0	X	发送 Repeated START 信号
		or no action	0	1	0	X	发送 STOP 信号;

		or no action	1	1	0	X	复位“sto”标志 发送 STOP 信号后接一个 START 信号; 复位“sto”标志
30H	I2CDAT 中的 Data byte 已发送;	Load data byte	0	0	0	X	发送 Data byte; 接收 ACK 信号
		or no action	1	0	0	X	发送 Repeated START 信号;
		or no action	0	1	0	X	发送 STOP 信号; 复位“sto”标志
		or no action	1	1	0	X	发送 STOP 信号后接一个 START 信号; 复位“sto”标志
38H	SLA+R/W 信号或 data bytes 仲裁失败	Load data byte	0	0	0	X	释放 I2C 总线; 进入“not addressed slave”状态
		or no action	1	0	0	X	当总线变为空闲 (free) 状态时, 发送一个 START 信号

表 82 I2C Status in Master Transmitter Mode

Status Code	Status of I2C	Application to/from I2CDAT	Software Response to I2CCON				Next Action taken by I2C Hardware
			sta	sto	si	aa	
08H	已发送 START 信号	Load SLA+R	X	0	0	X	发送 SLA+R 信号 发送 ACK 信号
10H	已发送 Repeated START 信号	Load SLA+R	X	0	0	X	同上
		Or Load SLA+W	X	0	0	X	发送 SLA+W 信号 I2C 将被切换至“master transmitter”模式
38H	“not ACK”位仲裁失败	No action or	0	0	0	X	释放 I2C 总线; I2C 将进入“slave”模式

		no action	1	0	0	X	当总线变为空闲 (free) 状态时, 发送一个 START 信号
40H	已发送 SLA+R 信号; 已接收 ACK 信号	No action	0	0	0	0	接收 Data byte; 返回 “not ACK” 信号
		no action	0	0	0	1	接收 Data byte; 返回 “ACK” 信号
48H	已发送 SLA+R 信号; 已接收 “not ACK” 信号	No action	1	0	0	X	发送 Repeated START 信号
		or no action	0	1	0	X	发送 STOP 信号; the 复位“sto”标志
		or no action	1	1	0	X	发送 STOP 信号后 接一个 START 信号; 复位“sto”标志
50H	已接收 Data byte; 已返回 ACK 信号	Read data byte	0	0	0	0	接收 Data byte; 返回 “not ACK” 信号
		or read data byte	0	0	0	1	接收 Data byte; 返回 ACK 信号
58H	已接收 Data byte; 已返回 “not ACK” 信号	Read data byte	1	0	0	X	发送 Repeated START 信号
		or read data byte	0	1	0	X	发送 STOP 信号; 复位“sto”标志
		or read data byte	1	1	0	X	发送 STOP 信号后 接一个 START 信号; 复位“sto”标志

表 83 I2C Status in Master Receiver Mode

Status Code	Status of I2C	Application to/from I2CDAT	Software Response to I2CCON				Next Action taken by I2C Hardware
			sta	sto	si	aa	
60H	已接收自身 SLA+W	No action	X	0	0	0	接收 Data byte, 返回 “not ACK” 信号

	信号; 已返回 ACK 信号	or no action	X	0	0	1	接收 Data byte 返回 ACK 信号
68H	在 SLA+R/W 中输掉仲裁; 已接收自身 SLA+W; 已返回 ACK 信号	No action	X	0	0	0	接收 Data byte, 返回“not ACK”信号
		Or no action	X	0	0	1	接收 Data byte, 返回 ACK 信号
70H	已发送 General call address (00H); 已返回 ACK 信号	No action Or	X	0	0	0	接收 Data byte, 返回“not ACK”信号
		no action	X	0	0	1	接收 Data byte, 返回 ACK 信号
78H	在 SLA+R/W 中输掉仲裁; 已接收 general call address, 已返回 ACK 信号	No action	X	0	0	0	接收 Data byte, 返回“not ACK”信号
		Or no action	X	0	0	1	接收 Data byte, 返回 ACK 信号
80H	先前已寻 址到自身 SLA 地址; 已接收 DATA; 已返回 ACK 信号	No action	X	0	0	0	接收 Data byte, 返回“not ACK”信号
		Or no action	X	0	0	1	接收 Data byte, 返回 ACK 信号
88H	先前已寻 址到自身 SLA 地址; 已接收 DATA; 已返回“not ACK”信号	Read data byte	0	0	0	0	切换至“not addressed slave”模式;
		Or					不识别自身从机地址 (SLA) 和 general call address
		read data byte or	0 0	0 0	0 0	1	切换至“not addressed slave”模式;

		read data byte	1	0	0	0	可识别自身 SLA 或 general call address
		or					切换至 “not addressed slave” 模式;
		read data byte	1	0	0	1	不识别自身 SLA 和 general call address; 当总线恢复空闲状态时, 发送 START 信号
							切换至 “not addressed slave” 模式;
							可识别自身 SLA 或 general call address; 当总线恢复空闲状态时, 发送 START 信号
90H	先前已寻址到 eneral call address; 已接收 DATA; 已返回 ACK 信号	Read data byte Or read data byte	X X	0 0	0 0	0 1	接收 Data byte, 返回 “not ACK” 信号 接收 Data byte; 返回 ACK 信号
98H	先前已寻址到 eneral call address; 已接收 DATA; 已返回 ACK 信号	Read data byte Or read data byte or read data byte or read data byte	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	切换至 “not addressed slave” 模式; 不识别自身 SLA 和 general call address; 切换至 “not addressed slave” 模式; 可识别自身 SLA 或 general call address; 切换至 “not addressed slave” 模式; 不识别自身 SLA 和 general call address;

							当总线恢复空闲状态时，发送 START 信号 切换至 “not addressed slave” 模式； 可识别自身 SLA 或 general call address； 当总线恢复空闲状态时，发送 START 信号
A0H	当仍旧寻址 SLV/REC 或 SLV/TRX 时，接收到 STOP 信号或 repeated START 信号	No action Or no action or no action or no action	0 0 1 1	0 0 0 0	0 0 0 0	0 1 0 1	切换至 “not addressed slave” 模式； 不识别自身 SLA 和 general call address； 切换至 “not addressed slave” 模式； 可识别自身 SLA 或 general call address； 切换至 “not addressed slave” 模式； 不识别自身 SLA 和 general call address； 当总线恢复空闲状态时，发送 START 信号 切换至 “not addressed slave” 模式； 可识别自身 SLA 或 general call address； 当总线恢复空闲状态时，发送 START 信号

表 84 I2C Status in slave Receiver Mode

Status	Status of	Application	Software Response to	Next Action taken
--------	-----------	-------------	----------------------	-------------------

Code	I2C	to/from I2CDAT	I2CCON				by I2C Hardware
			sta	sto	si	aa	
A8H	已接收 SLA+R 信号；已返回 ACK 信号	Load data byte	X	0	0	0	发送 last data byte； 接收 ACK 信号
		Or load data byte	X	0	0	1	发送 data byte； 接收 ACK 信号
B0H	SLA+R/W 作为主仲裁失败；已经 SLA+R 信号；已返回 ACK 信号	Load data byte	X	0	0	0	发送 last data byte； 接收 ACK 信号
		Or load data byte	X	0	0	1	发送 data byte； 接收 ACK 信号
B8H	已发送 data byte；已接收 ACK 信号	Load data byte	X	0	0	0	发送 last data byte； 接收 ACK 信号
		Or load data byte	X	0	0	1	发送 data byte； 接收 ACK 信号
C0H	已发送 data byte； 已接收 no ACK 信号	no action	0	0	0	0	切换到 “not addressed slave” 模式；不识别自身的 slave address 或者 general call address；
		or					
		no action	0	0	0	1	切换到 “not addressed slave” 模式；
		or					
		no action	1	0	0	0	识别自身的 slave address 或者 general call address；
		or					
		no action	1	0	0	1	切换到 “not addressed slave” 模式； 不识别自身的 slave address 或者 general call address； 当总线空闲的时候，发送 START 信号
							切换到 “not addressed slave” 模

							式; 识别自身的 slave address 或者 general call address; 当总线空闲的时候, 发送 START 信号
C8H	已发送 last data byte; 已接收 ACK 信号	no action	0	0	0	0	切换到 “not addressed slave” 模式; 不识别自身的 slave address 或者 general call address;
		or					
		no action	0	0	0	1	切换到 “not addressed slave” 模式;
		or					
		no action	1	0	0	0	识别自身的 slave address 或者 general call address;
		or					
		no action	1	0	0	1	切换到 “not addressed slave” 模式; 不识别自身的 slave address 或者 general call address; 当总线空闲的时候, 发送 START 信号
		or					
		no action	1	0	0	1	切换到 “not addressed slave” 模式; 识别自身的 slave address 或者 general call address; 当总线空闲的时候, 发送 START 信号

表 85 I2C Status in slave Transmitter Mode

Status Code	Status of I2C	Application to/from I2CDAT	Software Response to I2CCON				Next Action taken by I2C Hardware
			sta	sto	si	aa	
F8H	没有相关的有效状	No action	No action				等待或者进行当前的数据传输

	态 信 息 ; si=0						
00H	MST 或者 slave 模式下的总线错误	No action	0	1	0	X	只有内部硬件是有效的在“master” or “addressed slave”模式下. 在所有的情况下, 释放总线, 并且 I2C 切换到 “not addressed slave” 模式. 复位 “sto”标志位.

表 86 I2C Status - miscellaneous states

6.4 UART 接口

6.4.1 概述

HT8550支持2个串口：串口0和串口1。串口0对应的管脚名为UART0_TX和UART0_RX；串口1对应的管脚名为UART1_TX和UART1_RX。

串口0提供一个灵活的全双工同步/异步通信的接收器/发送器，支持四种工作模式，包括一种同步工作方式和三种异步工作方式：

- 模式0：串口0作为同步接收器/发送器，TX管脚产生移位时钟，RX管脚为输入或者输出的数据，串行数据为8位数据，波特率固定为 $F_{sys}/12$ 。
- 模式1：串口0作为异步接收器/发送器，串行数据共10位，即1位起始位，8位数据位，1位停止位，波特率可调。
- 模式2：串口0作为异步接收器/发送器，串行数据共11位，即1位起始位，9位数据位，1位停止位，波特率定为 $F_{sys}/32$ 或者定为 $F_{sys}/64$ 。
- 模式3：模式3与模式2的功能一样，但是模式3的波特率可调。

串口1提供一个灵活的全双工异步通信的接收器/发送器，有两种工作模式：

- 模式A：串口1作为异步接收器/发送器，串行数据共11位，即1位起始位，9位数据位，1位停止位，波特率可调。
- 模式B：串口1作为异步接收器/发送器，串行数据共10位，即1位起始位，8位数据位，1位停止位，波特率可调。
- HT8550 的2个串口与红外调制功能共用同一个逻辑,默认调制频率为38KHz。

6.4.2 红外调制

6.4.2.1 功能描述

MCU的两个串行接口共享逻辑都可以将TX信号调制成红外功能。共享逻辑由当TX为低时转换信号的计数器组成。这个计数器频率可以通过UART_IR_DIV寄存器来编程，默认值为大约38KHz。

可以通过UART_IR_CTRL寄存器来控制组合UART使能红外调制。

TX信号输出可以翻转，用来控制停止状态为低或者高。

6.4.2.2 时序图

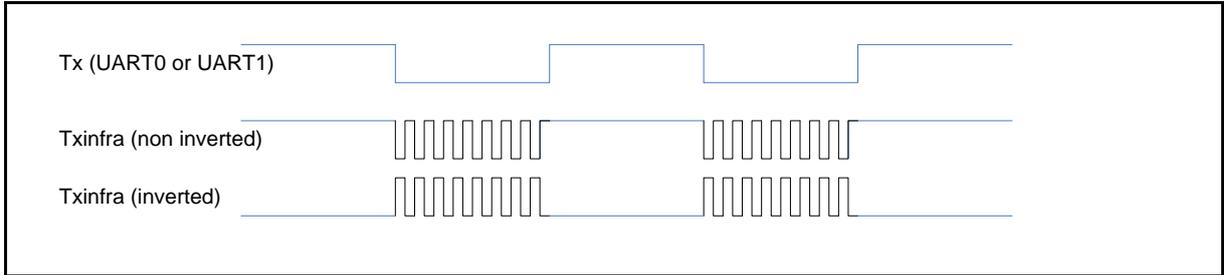


图 9 串口红外调制

6.4.2.3 特殊功能寄存器

地址	名称	复位值	功能描述
0xBC	UART_IR_CTRL	0x00	串口红外控制寄存器
0xBD	UART_IR_DIV	0x3C	串口红外频分寄存器

表 87 红外相关寄存器

6.4.2.3.1 UART_IR_CTRL

位	位名称	复位值	描述	属性
7:4	-	0	保留	R
3	-	0	串口 1 反转输出: 0: 当串口 TX 为高时(空闲), 输出高, 当 TX 为低时输出红外载波 1: 当串口 TX 为高时(空闲), 输出低, 当 TX 为低时输出红外载波.	R/W
2	-	0	串口 1 红外使能: 0: 串口 TX 1: 当 TX 为低时, 输出红外载波	R/W
1	-	0	串口 0 反转输出: 0: 当串口 TX 为高时(空闲), 输出高, 当 TX 为低时输出红外载波 1: 当串口 TX 为高时(空闲), 输出低, 当 TX 为低时输出红外载波	R/W
0	-	0	串口 0 红外使能: 0: 串口 TX 1: 当 TX 为低时, 输出红外载波	R/W

表 88 串口红外控制寄存器(UART_IR_CTRL 0xBCH)

*为了使用红外模块，必须通过 GPIO CTRL 寄存器将 UART 管脚配置成 UART 功能

6.4.2.3.2 UART_IR_DIV

位	位名称	复位值	描述	属性
---	-----	-----	----	----

7:0	-	0x3C	分频偏移量 N。红外频率是从 24MHz 振荡器分下来的，偏移量 N 加上 256，以形成 1/2 的除数。 IR = 24M / (2*(N+256)). 默认值是 IR = 24M / 2(60 + 256) = 37,974 Hz *如果红外功能已经使能，不要再改变频分值。	R/W
-----	---	------	---	-----

表 89 串口红外频分寄存器(UART_IR_DIV 0xBDH)

6.4.3 UART0 接口

6.4.3.1 串口 0 配置

串口0与GPIO复用，当作为串口0使用时，需要设置GPIO的功能如下：

- (1) 设置GPIO_MODE_SEL = 0x4;
- (2) 设置GPIO_CTRL[7:0] = 0x55;

6.4.3.2 串口 0 波特率

当串口0选择模式2时，波特率有两种选择：当PCON.7(即SMOD位)=1时，波特率为系统时钟的1/32；当PCON.7(即SMOD位)=0时，波特率为系统时钟的1/64。

在串口0的模式1、模式3时，波特率是可变的。

当ADCON.7=1时，波特率由波特率生成器和SMOD位的值确定：

$$\text{波特率} = \frac{2^{\text{SMOD}} \times f_{\text{cpu}}}{64 \times (2^{10} - \text{S0REL})}$$

其中SMOD是PCON.7的值；S0REL是10位无符号数，寄存器S0RELH和S0RELL的值；fMCU是系统时钟。

下表列举了常用波特率对应的S0REL值（ADCON.7=1,PCON.7=1,fcpu=24MHz）：

常用波特率	S0RELH	S0RELL
1200	0x01	0x87
2400	0x02	0xC8
4800	0x03	0x64
9600	0x03	0xB2
14400	0x03	0xCC
19200	0x03	0xD9
38400	0x03	0xEC

表 90 UART0 常用波特率

当ADCON.7=0时，波特率由定时器1的溢出速率和SMOD位的值确定：

$$\text{波特率} = \frac{2^{\text{SMOD}}}{32} \times \text{Timer1 溢出速率}$$

6.4.3.3 模式 0

模式0是一种标准的同步通信方式，TX管脚产生移位时钟，RX管脚为输入或者输出的数据。当MCU向S0BUF寄存器写入数据后，立刻启动发送，8位的数据将以Fsys/12的固定波特率从RX管脚输出，低位在前。当S0CON寄存器中的REN0被置1后，RX管脚的数据将存储在

S0BUF寄存器中。每帧包含8位数据信息，其波形如下图：

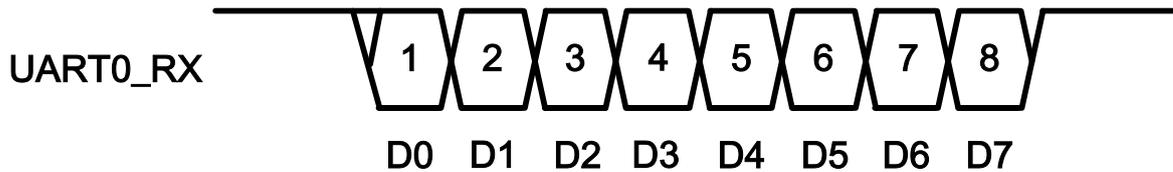


图 10 UART0 模式 0 的帧格式

6.4.3.4 模式 1

模式1是一种标准的异步通信方式，TX管脚产生移位时钟，RX管脚为输入或者输出的数据。每帧包含10位数据信息：1位起始位（0），8位数据位（低位在前），1位停止位（1），其波形如下图所示：

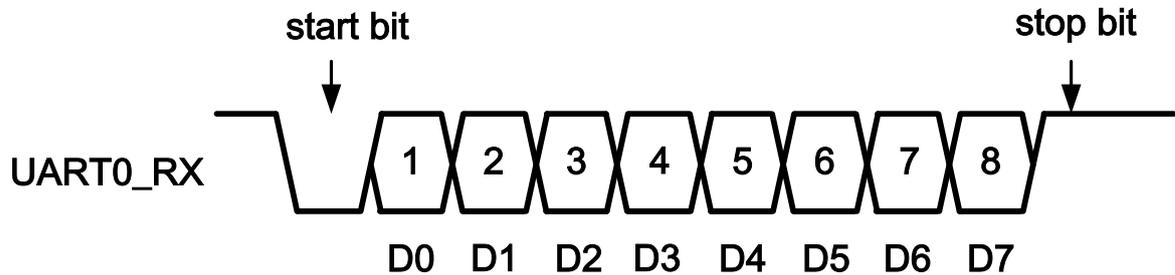


图 11 UART0 模式 1 发送帧格式

在模式1中 当MCU向S0BUF寄存器写入数据后，立刻启动发送，10位的数据将以 $F_{sys}/12$ 的固定波特率从RX管脚输出。当S0CON寄存器中的REN0被置1后，RX管脚的数据将存储在S0BUF寄存器中。

模式1数据传输的波特率可调，详细修改方参考《串口0波特率》一章。

6.4.3.5 模式 2

模式2也是一种标准的异步通信方式。使用9位数据位通信，TX管脚产生移位时钟，RX管脚为输入或者输出的数据。每帧包含11位数据信息：1位起始位（0），8位数据位（低位在前），1位可编程的第9位数据位，1位停止位（1），其波形如下图所示：

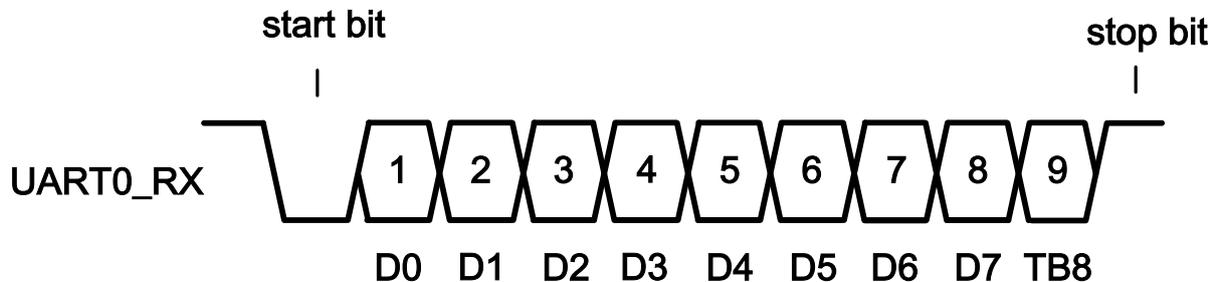


图 12 UART0 模式 2 帧格式

在数据发送时，串行口控制寄存器中的TB8作为第9位数据同时发送，发送完毕TB8被清除。当接收时，只有接收到第9位为1时才将串行口的中断标志RI置“1”。按照模式2进行数据传送时不允许进行奇/偶校验，第9位数据可作为数据/地址标志位。

模式2数据传输的波特率固定为Fsys/32或者定为Fsys/64。

6.4.3.6 模式3

模式3的数据传送方式和模式2相同，只是模式3的串口波特率可调，详细修改方参考《串口0波特率》一章。

6.4.3.7 多机通信

模式2和模式3中数据总共有9位，其中第9位可以用来作为多机通信的功能。

当寄存器S0CON中多机通信使能位置1后，MCU只有收到第9位数据位等于1时，才产生接收中断，否则，不产生接收中断。

例如，当主机和多个从设备通过串口进行通信时，从设备将多机通信使能位置1，主设备使用模式2或模式3发送1帧数据，数据中包含8位的地址和第9位为1，所有从设备将收到这帧数据，当从设备收到的地址与从设备在网络中的地址一致时，从设备将关闭多机通信使能位。此时，主设备将发送剩下的帧信息，每个数据的第9位置0，这样其他从设备将忽略这些数据，而地址匹配的从设备会接收这些数据，当从设备和主设备通信完成后，从设置就可以再次使能多机通信。

6.4.4 UART1 接口

6.4.4.1 串口1配置

串口1与GPIO复用，当作为串口1使用时，需要设置GPIO的功能如下：

- (1) 设置GPIO_MODE_SEL = 0x7;
- (2) 设置GPIO_CTRL[7:0] = 0x44;

6.4.4.2 串口1波特率

当串口1使用模式A或模式B时，波特率是可变的，波特率计算公式为：

$$\text{波特率} = \frac{f_{\text{sys}}}{32 \times (2^{10} - S1REL)}$$

其中，Fsys为系统时钟。

下表列举了常用波特率对应的S1REL值（Fsys = 24MHz）：

常用波特率	S1RELH	S1RELL
1200	0x01	0x87
2400	0x02	0xC8
4800	0x03	0x64
9600	0x03	0xB2
14400	0x03	0xCC
19200	0x03	0xD9
38400	0x03	0xEC

表 91 UART1 常用波特率

6.4.4.3 模式 A

模式A是一种异步通信方式，每帧包含9位数据信息，波特率可变。波特率生成器“S1RELH”“S1RELL”用于同步输入和输出传输。串口1的波特率不受寄存器“pcon”的“smod”位作用。TXD1引脚为数据输出端，当向寄存器“S1BUF”中写入数据后，传输开始。输出数据每帧包含11位数据信息：1位起始位（0），8位数据位（低位在前），1位可编程的第9位数据位（来自寄存器“S1CON”的“tb81”位），1位停止位（1）。RXD1引脚为数据输入端，当RXD1引脚的下降沿出现，开始接收。寄存器“S1BUF”完成接收后，输入数据才可继续，第9位数据存放在寄存器“S1CON”的“rb81”位。在接收期间，寄存器“S1BUF”“rb81”保持不变，直到接收完成。数据收发时，第9位数据可作为奇/偶校验或数据/地址标志位使用。

6.4.4.4 模式 B

模式B是一种异步通信方式，每帧包含8位数据信息，波特率可变。波特率生成器“S1RELH”“S1RELL”用于同步输入和输出传输。波特率不受寄存器“PCON”的“smod”位作用。TXD1引脚为数据输出端，当向寄存器“S1BUF”中写入数据后，传输开始。输出数据每帧包含10位数据信息：1位起始位（0），8位数据位（低位在前），1位停止位（1）。RXD1引脚为数据输入端，当RXD1引脚的下降沿出现，开始接收。寄存器“S1BUF”完成接收后，输入数据才可继续。在接收期间，寄存器“S1BUF”“rb81”保持不变，直到接收完成。

6.4.4.5 多机通信

模式A和模式B中数据总共有9位，其中第9位可以用来作为多机通信的功能。

当寄存器S1CON中多机通信使能位置1后，MCU只有收到第9位数据位等于1时，才产生接收中断，否则，不产生接收中断。

例如，当主机和多个从设备通过串口进行通信时，从设备将多机通信使能位置1，主设备使用模式2或模式3发送1帧数据，数据中包含8位的地址和第9位为1，所有从设备将收到这帧数据，当从设备收到的地址与从设备在网络中的地址一致时，从设备将关闭多机通信使能位。此时，主设备将发送剩下的帧信息，每个数据的第9位置0，这样其他从设备将忽略这些数据，而地址匹配的从设备会接收这些数据，当从设备和主设备通信完成后，从设备就可以再次使能多机通信。

6.4.5 特殊功能寄存器

串口相关寄存器如下：

地址	名称	复位值	功能描述
0x87	PCON	0x00	电源控制寄存器
0x98	S0CON	0x00	串口 0 控制寄存器
0x99	S0BUF	0x00	串口 0 数据缓冲寄存器
0x9B	S1CON	0x00	串口 1 控制寄存器
0x9C	S1BUF	0x00	串口 1 数据缓冲寄存器
0x9D	S1RELL	0x03	串口 1 波特率发生器的重载值低位
0xAA	S0RELL	0xD9	串口 0 波特率发生器的重载值低位
0xBA	S0RELH	0x03	串口 0 波特率发生器的重载值高位

0xBB	S1RELH	0x00	串口 1 波特率发生器的重载值高位
0xD8	ADCON	0x00	串口 0 波特率选择寄存器

表 92 UART 寄存器列表

6.4.5.1 PCON

寄存器描述参考《电源与时钟》章节。

6.4.5.2 S0CON

位	位名称	复位值	描述	属性
7	SM0	0	串行工作方式选择位: 00: 模式 0; 01: 模式 1; 02: 模式 2; 03: 模式 3;	R/W
6	SM1	0		R/W
5	SM20	0	多机交互使能位 置 1, 则使能多机通信功能。	R/W
4	REN0	0	允许接收位 置 1 时, 允许串口 0 接收; 置 0 时, 禁止串口 0 接收。	R/W
3	TB80	0	发送数据位的第 9 位 模式 2、3 中是被发出去的第 9 位数据位。可以用来做奇偶校验或多机通信, 由软件控制。	R/W
2	RB80	0	接收数据位的第 9 位 当串口 0 配置为模式 2、3 时, 代表收到的第 9 数据位; 当串口 0 配置为模式 1 时, 并且 SM20=1, 代表停止位; 当串口 0 配置为模式 0 时, 该位无效。	R/W
1	TIO	0	发送中断标志 当串口 0 发送完成时, 由硬件置位; 必须由软件清零。 如果串口 0 配置为模式 0, 则发送完 D7 后, 标志位被置 1; 其他模式, 则发送停止位前, 标志位置 1。	R/W
0	RI0	0	接收中断标志 当串口 0 接收完成时, 由硬件置位; 必须由软件清零。 如果串口 0 配置为模式 0, 则接收完 D7 后, 标志位被置 1; 其他模式, 则接收停止位前, 标志位置 1。	R/W

表 93 串口 0 控制寄存器(S0CON 0x98H)

6.4.5.3 S0BUF

位	位名称	复位值	描述	属性
---	-----	-----	----	----

7:0	-	0x00	对寄存器 S0BUF 写操作，则串口 0 自动启动串口发送；对寄存器 S0BUF 读操作，则从串行接收缓存中读取串口 0 收到的数据。	R/W
-----	---	------	---	-----

表 94 串口 0 缓冲寄存器(S0BUF 0x99H)

6.4.5.4 S1CON

位	位名称	复位值	描述	属性
7	SM	0	串行工作方式选择位： 0：模式 A；串口 1 作为 9 位的串口使用。 1：模式 B，串口 1 作为 8 位的串口使用。	R/W
6	-	0	保留，读返回总是 0。	R
5	SM21	0	多机交互使能位 置 1，则使能多机通信功能。	R/W
4	REN1	0	允许接收位 置 1 时，允许串口 1 接收；置 0 时，禁止串口 1 接收。	R/W
3	TB81	0	发送数据位的第 9 位 模式 A 中是被发出去的第 9 位数据位。可以用来做奇偶校验或多机通信，由软件控制。	R/W
2	RB81	0	接收数据位的第 9 位 当串口 1 配置为模式 A 时，代表收到的第 9 数据位；当串口 1 配置为模式 B 时，并且 SM20=1，代表停止位；	R/W
1	TI1	0	发送中断标志 当串口 1 发送完成时，由硬件置位；必须由软件清零。 如果串口 1 配置为模式 A 或 B，则发送停止位前，标志位置 1。	R/W
0	RI1	0	接收中断标志 当串口 0 接收完成时，由硬件置位；必须由软件清零。 如果串口 1 配置为模式 A 或 B，则接收停止位前，标志位置 1。	R/W

表 95 串口 1 控制寄存器(S1CON 0x9BH)

6.4.5.5 S1BUF

位	位名称	复位值	描述	属性
7:0	-	0x00	对寄存器 S1BUF 写操作，则串口 1 自动启动串口发送；对寄存器 S1BUF 读操作，则从串行接收缓存中读取串口 1 收到的数据。	R/W

表 96 串口 1 缓冲寄存器(S1BUF 0x9CH)

6.4.5.6 S1RELL

位	位名称	复位值	描述	属性
7:0	-	0x00	可以配置串口 1 波特率。为加载寄存器的 BIT[7:0]。	R/W

表 97 串口 1 波特率发生器的重载值低位(S1RELL 0x9DH)

6.4.5.7 S0RELL

位	位名称	复位值	描述	属性
7:0	-	0x00	可以配置串口 0 波特率。为加载寄存器的 BIT[7:0]。	R/W

表 98 串口 0 波特率发生器的重载值低位(S0RELL 0xAAH)

6.4.5.8 S0RELH

位	位名称	复位值	描述	属性
7:2	-	0	保留，读返回 0	R
1:0	-	0	可以配置串口 0 波特率。为加载寄存器的 BIT[9:8]。	R/W

表 99 串口 0 波特率发生器的重载值高位(S0RELH 0xBAH)

6.4.5.9 S1RELH

位	位名称	复位值	描述	属性
7:2	-	0	保留，读返回 0	R
1:0	-	0	可以配置串口 1 波特率。为加载寄存器的 BIT[9:8]。	R/W

表 100 串口 1 波特率发生器的重载值高位(S1RELH 0xBBH)

6.4.5.10 ADCON

位	位名称	复位值	描述	属性
7	BD	0	串口 0 波特率选择寄存器。当设置为 1 时，串口 0 波特率由内部时钟控制；当设置为 0 时，串口 0 波特率由定时器 1 的溢出时间控制。详细波特率设置参考《UART 接口》一章。	R/W
6:0	-	0	保留，读返回 0	R

表 101 AD 控制寄存器(ADCON 0xD8H)

6.5 RTC 单元

6.5.1 概况

RTC单元精度为1/256秒。允许用户软件读取秒，分钟，小时，星期和日期。日期由一个16位的寄存器表示，由用户软件自定义日期的格式。RTC总时间达179年之久，支持闹钟功能和周期性中断功能。能够实现秒、分、小时周期中断。通过中断可以将MCU从IDLE/STOP状态中恢复。RTC框图如下：

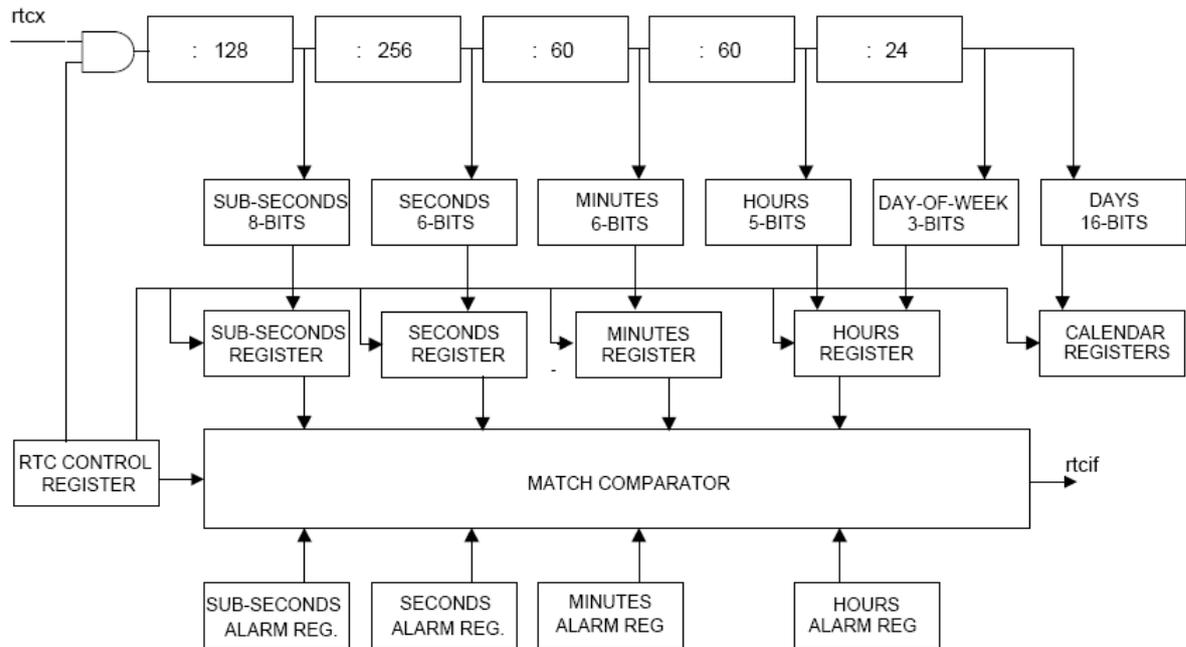


图 13 RTC 框图

RTC的工作时钟源由系统内部分配得到32.768KHz的时钟，供RTC电路工作。RTC中断与外部中断0复用，中断向量号为0。

6.5.2 RTC 开启和停止

RTC的开启和关闭由寄存器RTCC.0控制，当软件设置该位为1时，RTC时钟(rtcx)将驱动所有RTC寄存器，并且RTC内部计数器开始计数；当软件设置该位为0时，所有RTC寄存器的值将保持不变。

6.5.3 RTC 寄存器读写

读写RTC寄存器时，需要将RTCWE位（写使能）或RTCRES（读使能）置1。当RTCWE或RTCRES置1时，当前时间保持器将停止更新直到该位被清零。推荐软件中读写操作必须在1ms内完成，如果用户软件在读写RTC寄存器的过程中，亚秒计数事件产生，则RTCWE（RTCRES）清零后，硬件会在最快时间内处理该事件。

RTCWE和RTCRES不能同时被置1，如果用户软件这样操作，则硬件不响应。如果用户软件将该位置1后，1.95ms内没有将该位清零，则硬件自动将其清零。

向RTC寄存器中写入错误的数据会导致RTC给出错误时间。例如写入的值超出了小时，分钟的范围。

6.5.4 RTC 报警

报警功能：当RTC计数器的值与报警寄存器的值匹配后就会产生中断，并将中断标志位置1(rtcif)，如果RTC中断被使能，则MCU可以响应中断，并且将从IDLE/STOP模式中唤醒；如果中断没有被使能，MCU也能从IDLE/STOP模式中唤醒，但是不产生中断。

当用户软件设置RTC报警寄存器时，并不需要将RTCWE(RTCRES)置1。RTC报警器中没有日期报警的设置，因此由用户自定义报警日期。当用户软件写入不合法的值得到报警寄存器

中(例如向RTAH中写入61), 会导致RTC永远不能匹配, 从而不能产生中断。

RTC报警寄存器由RTASS, RTAS, RTAM, RTAH组成, 当用户软件中只使能其中1个报警寄存器作为比较时, 则RTC在一个周期内产生1次中断; 当用户软件中使能了多个报警寄存器作为比较时, 则RTC只有在每一个时间都匹配的情况下, 才产生中断。

使用RTC报警功能, 产生中断的流程:

- 1)清除ERTC位(IEN4.5寄存器), 关闭RTC中断; 清除所有比较使能寄存器(RTCC.7-4);
- 2)写报警寄存器(RTASS, RTAS, RTAM, RTAH), 设置报警时间点;
- 3)使能报警位和使能RTC中断位。

6.5.5 特殊功能寄存器

RTC寄存器列表如下:

地址	名称	复位值	功能描述
0xCE	RTCSEL	0x00	RTC 选择寄存器
0xCF	RTCDATA	0x00	RTC 数据寄存器
RTCSEL=0x4	RTCCC	0x00	RTC 控制寄存器
RTCSEL=0x6	RTCSS	0x00	RTC 亚秒寄存器[0: 255]. (只读)
RTCSEL=0x7	RTCS	0x00	RTC 秒寄存器[0: 59]
RTCSEL=0x8	RTCM	0x00	RTC 分寄存器[0: 59]
RTCSEL=0x9	RTCH	0x00	RTC 小时寄存器[0: 23]
RTCSEL=0xA	RTCD0	0x00	RTC 日寄存器 (低位)
RTCSEL=0xB	RTCD1	0x00	RTC 日寄存器 (高位)
RTCSEL=0x0	RTASS	0x00	闹钟亚秒比较寄存器[0: 255].
RTCSEL=0x1	RTAS	0x00	闹钟秒比较寄存器[0: 59].
RTCSEL=0x2	RTAM	0x00	闹钟分比较寄存器[0: 59].
RTCSEL=0x3	RTAH	0x00	闹钟小时比较寄存器[0: 23].
0xD1	IEN4	0x00	RTC 中断使能寄存器

表 102 RTC 寄存器列表

6.5.5.1 RTCSEL

位	位名称	复位值	描述	属性
7:4	-	0	保留。	R
3:0	-	0	A 4-bit pointer to internal registers of the RTC. RTCSEL 寄存器实现 RTC 内部特殊寄存器的选择。当要读写 RTC 内部寄存器, 可以先配置 RTCSEL 为相应的值, 然后再读写 RTCDAT 寄存器。	R/W

表 103 RTC 选择寄存器(RTCSEL 0xCEH)

注意: 当使用间接地址读写 RTCDATA 时, 如果需要连续写 RTCDATA, 则必须保证 2 次写间隔时间大于 30.5ms。

6.5.5.2 RTCDATA

位	位名称	复位值	描述	属性
7:0	-	0	RTC 数据寄存器,用于 RTC 内部寄存器数据读写。当 RTCSEL 寄存器选择不同的值后, RTCDATA 的定义也随之变化。详细描述参考下表。	R/W

表 104 RTC 数据寄存器(RTCDAT 0xCFH)

6.5.5.3 IEN4

位	位名称	复位值	描述	属性
7:6	-	0	保留。	R
5	ERTC	0	当 ERTC 置 1 时,使能 RTC 中断;否则,禁止 RTC 中断。	R/W
4:0	-	0	保留。	

表 105 RTC 中断使能寄存器(IEN4 0xD1H)

6.5.5.4 间接访问寄存器

当 RTCSEL = 4 时, RTCDATA 定义如下:

位	位名称	复位值	描述	属性
7	SSCE	0	亚秒比较使能: 置 1, 使能 RTCSS 与 RTASS 寄存器的比较; 置 0, 忽略 RTCSS 与 RTASS 寄存器的比较。	R/W
6	SCE	0	秒比较使能 置 1, 使能 RTCS 与 RTAS 寄存器的比较; 置 0, 忽略 RTCS 与 RTAS 寄存器的比较。	R
5	MCE	0	分比较使能 置 1, 使能 RTCM 与 RTAM 寄存器的比较; 置 0, 忽略 RTCM 与 RTAM 寄存器的比较。	R/W
4	HCE	0	小时比较使能 置 1, 使能 RTCH 与 RTAH 寄存器的比较; 置 0, 忽略 RTCH 与 RTAH 寄存器的比较。	R/W
3	RTCRC	0	读使能 当读 RTCSS, RTCS, RTCM, RTCH, RTCD0, RTCD1 寄存器时, 需要将该位置 1。 注意该位不能和 RTCC.2 同时置 1。	R/W
2	RTCWE	0	写使能 当写 RTCSS, RTCS, RTCM, RTCH, RTCD0, RTCD1 寄存器时, 需要将该位置 1。 注意该位不能和 RTCC.3 同时置 1。	R/W
1	RTCIF	0	中断状态标志位 当 RTC 产生中断后, 该位被置 1。需要软件将其清零。软件写 1, 不能产生中断。 当 RTCC.7-4 全部清零时, 该位自动清零。	R/W

0	RTCE	0	RTC 使能 置 1, RTC 运行; 置 0, RTC。	R/W
---	------	---	---	-----

表 106 RTCC RTC 时钟控制寄存器(RTCSEL=0x4)

当 RTCSEL = 6 时, RTCDATA 为只读寄存器, 定义为 RTCSS, 表示 1/256 秒。该寄存器从 0 到 255 变化, 不能写。当 RTCWE 位清零后, RTCSS 自动清零。

当 RTCSEL = 7 时, RTCDATA 为可读写寄存器, 定义为 RTCS, 表示秒。该寄存器从 0 到 59 变化。

当 RTCSEL = 8 时, RTCDATA 为可读写寄存器, 定义为 RTCM, 表示分钟。该寄存器从 0 到 59 变化。

当 RTCSEL = 9 时, RTCDATA 为可读写寄存器, 定义为 RTCH, 其中 BIT[4:0]表示小时, BIT[7:5]表示星期。BIT[4:0]从 0 到 23 变化, BIT[7:5]从 1 到 7 变化。

当 RTCSEL = 10 时, RTCDATA 为可读写寄存器, 定义为 RTCD0。

当 RTCSEL = 11 时, RTCDATA 为可读写寄存器, 定义为 RTCD1。RTCD0, RTCD1 寄存器表示日期。由用户软件决定日期的格式, RTCD1 存储高字节, RTCD0 存储低字节。

当 RTCSEL = 0 时, RTCDATA 为可读写寄存器, 定义为 RTASS, 亚秒报警寄存器。该值由用户软件设置。

当 RTCSEL = 1 时, RTCDATA 为可读写寄存器, 定义为 RTAS, 秒报警寄存器。该值由用户软件设置。

当 RTCSEL = 2 时, RTCDATA 为可读写寄存器, 定义为 RTAM, 分钟报警寄存器。该值由用户软件设置。

当 RTCSEL = 3 时, RTCDATA 为可读写寄存器, 定义为 RTAH, 小时报警寄存器。该值由用户软件设置。

6.6 看门狗定时器

6.6.1 概述

HT8550可通过看门狗定时器 (Watchdog Timer) 监控软、硬件, 以免跑飞。HT8550的看门狗是一个15位的计数器, 每24或384个时钟周期加1。如果软件在786336个时钟周期 (24MHz 时钟下约32.5ms) 或1258136个时钟周期 (24MHz时钟下约0.5s) 内没有清零看门狗计数器, 看门狗计数器将溢出, 产生内部复位信号对系统进行复位。

6.6.2 看门狗使能

6.6.2.1 硬使能

HT8550有专门的看门狗使能引脚WDTEN。如果在系统复位的过程中 (RSTN接GND) WDTEN接VCC, 并保持到系统复位完成 (RSTN拉高), 则看门狗被开启; 否则, 如果在系统复位过程中和复位完成后, WDTEN时钟接GND, 则看门狗不工作。

注意: 系统工作中, 如果没有复位产生, 则改变WDTEN电平, 将无影响。

6.6.2.2 软使能

如果WDTEN接GND, 系统上电后, 看门狗不工作。如果要开启看门狗, 可通过配置相关寄存器swdt (IEN1.6), 开启看门狗。

注意: 看门狗一旦开启, 将无法关闭。除非将系统复位, 并将WDTEN接GND, 并保持到系统复位完成, 才可关闭看门狗。

6.6.3 看门狗计数器清零

当连续将wdt (IEN0.6) 和swdt (IEN1.6) 写1后, 看门狗计数器将自动清零, 中间不可以再插入其他指令。看门狗计数器的高7位可以通过设置wdtrel寄存器的BIT[6:0]重新载入。当wdtrel的值增加, 看门狗溢出的时间将变短。

6.6.4 特殊功能寄存器

看门狗寄存器如下:

地址	名称	复位值	功能描述
0x86	WDTREL	0x00	看门狗加载寄存器

表 107 看门狗寄存器列表

6.6.4.1 WDTREL

位	位名称	复位值	描述	属性
7	PS	0	BIT7=0 时, 看门狗计数器每 24 个机器周期, 自动增 1; 当 BIT7=1 时, 看门狗计数器每 384 个机器周期, 自动增 1。	R/W
6:0	-	0	BIT[6:0]的值将自动加载到内部看门狗计数器的高 7 位, 当软件将看门狗计数器清零时。	R/W

表 108 看门狗加载寄存器(WDTREL 0x86H)

6.7 JTAG 调试

支持标准的JTAG接口, 用户可以通过引脚TCK、TDI、TDO、TMS实现在线编程和在线调试。

7 AFE 相关寄存器配置

7.1 寄存器描述

HT8550与模拟前端电路相关的寄存器如下表, 另外集成高压侧供电电源掉电检测功能, 以便于用户软件判断交流断电时保存必要的数。据。

其中模拟电路的power、AGC、TX_VGA、ALC等功能可根据应用需求进行灵活配置, 具体定义如下:

地址	名称	复位值	功能描述
0xF3	ANALOG_CFG1	0x08	模拟前端电路控制寄存器
0xC9	ANALOG_CFG2	0x00	
0xBF	VCC_LOSS	0x00	
0xF4	AGC_LOOP_CFG1	0x00	
0xF7	AGC_LOOP_CFG2	0x00	
0xF8	AGC_LOOP_CFG3	0x00	
0xFD	AGC_LOOP_STATUS1	0x00	
0xD4	AGC_LOOP_STATUS2	0x00	
0xAF	ALC_CONTROL	0x00	

表 109 模拟前端寄存器列表

7.1.1 ANALOG_CFG1

位	位名称	复位值	描述	属性
7:5	-	0	芯片测试用。Default=0。	R/W
4	DAC_EN	0	[4:0]均为 1 有效。单独 RX 工作时，可设置 [4:0]=01111。单独 TX 工作时，可设置 [4:0]=11000。	R/W
3	REF_EN	1		R/W
2	VGA_EN	0		R/W
1	BUF_EN	0		R/W
0	ADC_EN	0		R/W

表 110 ANALOG_CFG1 寄存器(ANALOG_CFG1 0xF3H)

7.1.2 ANALOG_CFG2

位	位名称	复位值	描述	属性
7:6	-	0	RX POWER 控制，决定 receiver 功耗。00 为 6.5mA，01 和 10 为 9mA，11 为 12.5mA。功耗和性能有折中关系，推荐配置 10。	R/W
5:4	-	0	TX ALC gain 控制，配置 11、10/01、00 分别对应 PA 发送功率为 1W、0.5W、0.25W（外部电压、电流采样电路请见 APP note）。	R/W
3	-	0	片上 TX ALC 使能，1 有效。	R/W
2:0	-	0	片上 TX VGA 增益设置，如设置此值，应使 bit3=0。有八种增益设置，对应数值如下： 000: -0.5dB 001: 1dB 010: 2.5dB 011: 4dB 100: 5.5dB 101: 7dB 110: 8.5dB 111: 10dB	R/W

表 111 ANALOG_CFG2 寄存器(ANALOG_CFG2 0xC9H)

7.1.3 VCC_LOSS

位	位名称	复位值	描述	属性
7:3	-	0	ALC 单元 U 通道的实时信号幅度。	R
2	-	0	VCC_loss 的实时状态。	R
1	-	0	VCC_loss 中断标志位。当 BIT[0]=1 时，并且产生 VCC_loss 信号，则该标志位被置 1。用户软件写 1，清零。	R/W
0	-	0	VCC_loss 中断使能，写 1 使能中断。	R/W

表 112 VCC_LOSS 寄存器(VCC_LOSS 0xBFH)

7.1.4 AGC_LOOP_CFG1

位	位名称	复位值	描述	属性
---	-----	-----	----	----

7:0	-	0	模拟电路配置寄存器。用户软件不需要配置。	R/W
-----	---	---	----------------------	-----

表 113 AGC_LOOP_CFG1 寄存器(AGC_LOOP_CFG1 0xF4H)

7.1.5 AGC_LOOP_CFG2

位	位名称	复位值	描述	属性
7:4	-	0	RX通道选择关闭AGC时 手动设置VGA的增益, 0000~1111 对应 -3~42dB, 精度为3dB。	R/W
3:0	-	0	RX_VGA 最大增益限制。	R/W

表 114 AGC_LOOP_CFG2 寄存器(AGC_LOOP_CFG2 0xF7H)

7.1.6 AGC_LOOP_CFG3

位	位名称	复位值	描述	属性
7	-	0	1:由MCU对tx gain进行控制; 0: ALC 功能开启。	R/W
6	-	0	保留位, default 0。	R
5	-	0	接收信号载频选择, 当载频小于375kHz时, 设置为0, 大于375kHz时设置为1。	R/W
4	-	0	AGC使能, 1使能AGC自动调节。	R/W
3:1	-	0	AGC单步时间设定, 选项如下: 3'b000 refresh every 2.4ms 3'b001 refresh every 0.3ms 3'b010 refresh every 0.7ms 3'b011 refresh every 1.0ms 3'b100 refresh every 1.4ms 3'b101 refresh every 1.7ms 3'b110 refresh every 2.1ms 3'b111 refresh every 0.17ms	R/W
0	-	0	AGC目标终值设定, 0表示1V (Vpp); 1表示0.5V (Vpp)。	R/W

表 115 AGC_LOOP_CFG3 寄存器(AGC_LOOP_CFG3 0xF8H)

7.1.7 AGC_LOOP_STATUS1

位	位名称	复位值	描述	属性
7:6	-	0	保留。	R
5	-	0	输入信号强度指示。	R
4	-	0	RX_AGC 锁定指示。	R
3	-	0	RX_AGC 增益值。	R
2:0	-	0	保留。	R

表 116 AGC_LOOP_STATUS1 寄存器(AGC_LOOP_STATUS1 0xFDH)

7.1.8 AGC_LOOP_STATUS2

位	位名称	复位值	描述	属性
7:0	-	0	输入信号的均值	R

表 117 AGC_LOOP_STATUS2 寄存器(AGC_LOOP_STATUS2 0xD4H)

7.1.9 ALC_CONTROL

位	位名称	复位值	描述	属性
7:3	-	0	ALC 单元 I 通道实时信号幅度。	R
2:0	-	0	TX VGA 增益配置，当 AGC_LOOP_CFG3[7]=1 时，TX gain 不再由片上 ALC 单元控制，此时控制权限交由 MCU 执行。	R/W

表 118 ALC_CONTROL 寄存器(ALC_CONTROL 0xAFH)

7.2 ALC 功能描述

ALC 功能用于对发送信号的幅度进行控制，当开启 ALC 功能后，由片上设定输出目标幅度（功率），输出幅度（功率）将不受负载阻抗的影响而保持定值，详细参考寄存器 ANALOG_CFG2[5:4]。

ALC 电路功能框图

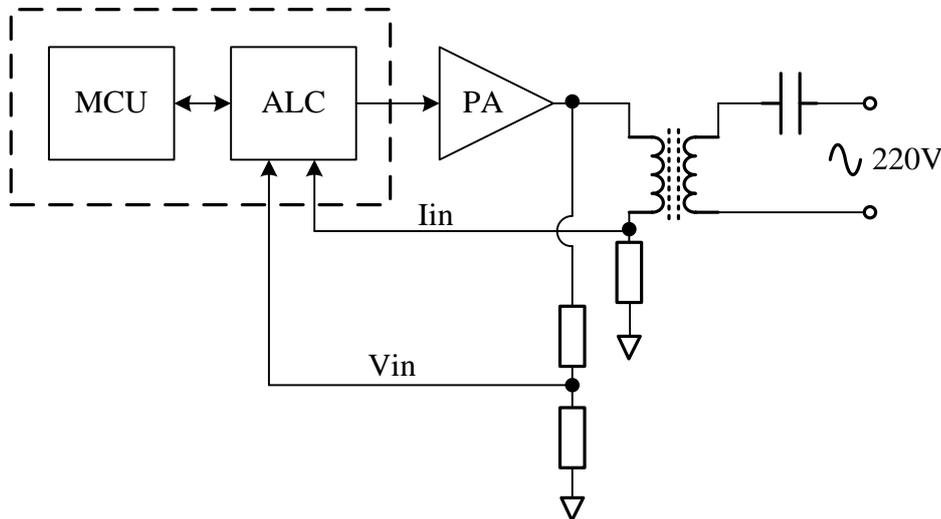


图 14 ALC 框图

图中方框所示为芯片内部电路。

ALC 电路也可与 MCU 之间交互数据，当配置 AGC_LOOP_CFG3 bit[7]=1 时，MCU 读取 Vin 和 lin 的实时信号，并由 MCU 直接控制 tx_gain（ANALOG_CFG2[2:0]）的数值，从而达到控制输出信号电平的目的。

7.3 掉电检测

掉电检测功能用于判断片外高压电源的跌落状态，外部电压通过引脚 VCC_DET 输入，与内部 1.2V 参考电压相比较，并将结果送入 MCU。使用者可灵活改变片外分压电阻的比例得到需要的电压跌落判断阈值。

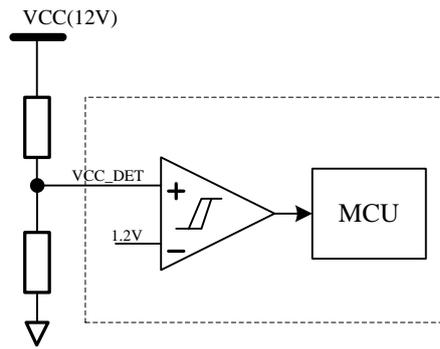


图 15 掉电检测功能示意图

7.4 AGC 的配置

为了处理接收通道大范围的信号变动，HT8550 具有完善的 AGC 电路，AGC 的终值和步长都是可以设定的。根据调制信号峰均比的大小，AGC 终值可以设置 1V 或者 0.5V。AGC 的单步时间可在 0.17mS-2.4mS 间调整，使用者可根据信道上具体的噪声干扰和信号时变状况灵活设置。详细使用方法请参考《HT8550 应用手册》。

8 MCU 与 PHY 交互

8.1 概述

MCU与PHY之间交互通过MCU的数据存储空间和PHY中断完成。其中PHY拥有独立的DMA功能，能访问MCU内部8K的数据空间，PHY相关配置的寄存器映射在MCU数据空间0xF000-0xFFFF，MCU通过访问数据存储空间来配置这些寄存器。PHY作为MCU的协处理器，为半双工模式，有IDLE，TX，RX三种工作状态。

MCU与PHY之间的通信协议可通过软件灵活的配置。默认的通信帧格式定义如下：



图 16 交互帧结构图

其中preamble用于物理层的帧同步；FCH中包含了payload的长度，调制方式，卷积编码模式，分集模式，和CRC5校验位；payload最大支持255字节。

8.2 PHY 功能简介

PHY 拥有独立的寄存器，能够灵活的配置 PHY 层各模块的属性，并且拥有自动关闭和开启时钟的功能，从而使 PHY 消耗的电流最小。PHY 主要支持以下功能：

- 支持 DBPSK, DQPSK, D8PSK 模式，符号速率可调，最高可达 41.67kbaud；
- 支持双载波发送和接收；

- 支持交织及鲁棒模式发送;
- 支持载波频率范围为 3KHz-500KHz, 任意可调;
- 支持信号强度值估计和信号 SNR 值估计, 信号 RSSI 值估计;
- 支持 PDM 调制发送;
- 支持停止发送功能;

详细的配置过程参考《应用手册》。

8.3 参数设置

PHY参数通过MAC层原语向PHY传递参数, PHY层将给MAC响应, 并在下一次发送时, 使用最新的配置, 详细的配置过程和原语定义, 参考《应用手册》。

9 芯片封装尺寸

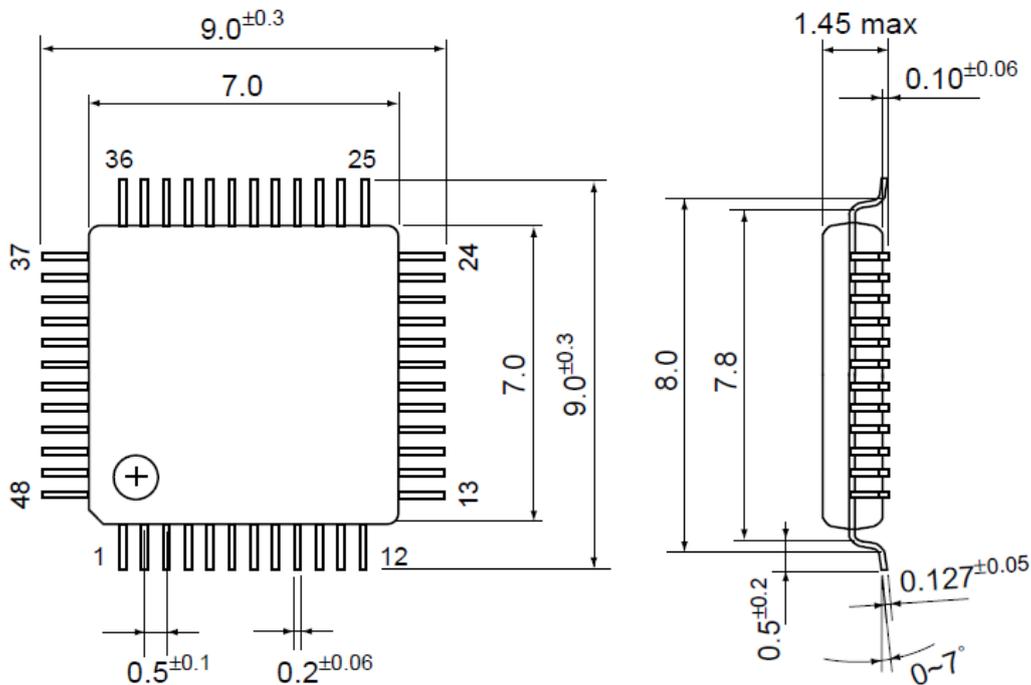


图17 LQFP48 package information(单位 mm)

10 附录

10.1 寄存器列表

地址	名称	复位值	功能简介	索引
0x87	PCON	0x08	电源控制寄存器	4.2.5
0xAC	ADC_QMD_CLK_CTRL	0x10	ADC 和 QMD 时钟控制	4.3.3
0xD6	FLASH_CLK_DIV	0x09	外部 Flash 时钟分频控制	4.3.3
0xE5	CLK_ON	0x29	时钟开启控制寄存器	4.3.3

地址	名称	复位值	功能简介	索引	
0xE6	CLK_OFF	0x0	时钟关闭控制寄存器	4.3.3	
0xF1	MAC_CLK_DIV	0x01	MAC 时钟分频控制	4.3.3	
0xF2	CLK_CTRL	0x0	时钟控制寄存器	4.3.3	
0xE7	RST_SET	0x0	复位设置	4.4.5	
0xE8	RST_CLR	0x0	复位撤销	4.4.5	
0xA1	DMASEL	0x0	DMA 线程选择寄存器	5.3.3	
0xA2	DMAS0	0x0	DMA 配置和状态寄存器, 根据 DMASEL 寄存器的配置, 可以控制 3 个线程	5.3.3	
0xA3	DMAS1	0x0		5.3.3	
0xA4	DMAS2	0x0		5.3.3	
0xA5	DMAS3	0x0		5.3.3	
0xA6	DMAT0	0x0		5.3.3	
0xA7	DMAT1	0x0		5.3.3	
0xB1	DMAT2	0x0		5.3.3	
0xB2	DMAT3	0x0		5.3.3	
0xB3	DMAC0	0x0		5.3.3	
0xB4	DMAC1	0x0		5.3.3	
0xB5	DMACSR	0x0		5.3.3	
0xB6	DMATC	0x0		5.3.3	
0xB7	DMASC	0x0		5.3.3	
0x9A	IEN2	0x0		IEN2 中断使能	5.4.11
0xA8	IEN0	0x0		IEN0 中断使能	5.4.11
0xB8	IEN1	0x0	IEN1 中断使能	5.4.11	
0xC0	IRCON	0x0	中断请求控制寄存器	5.4.11	
0xFA	MAC_INTR_MASK	0x0	MAC 中断屏蔽		
0xFB	GPIO_INTR_FLAGS	0x0	GPIO 外部中断标志	5.4.11	
0xFC	PHY_INTR_FLAGS	0x0	PHY 中断标志	5.4.11	
0x88	TCON	0x00	定时器控制寄存器	5.5.5	
0x89	TMOD	0x00	定时器方式寄存器	5.5.5	
0x8A	TL0	0x00	定时器 0 低 8 位	5.5.5	
0x8B	TL1	0x00	定时器 1 低 8 位	5.5.5	
0x8C	TH0	0x00	定时器 0 高 8 位	5.5.5	
0x8D	TH1	0x00	定时器 1 高 8 位	5.5.5	
0xC8	T2CON	0x40	定时器 2 控制寄存器	5.5.5	
0xCC	TL2	0x00	定时器 2 低 8 位	5.5.5	
0xCD	TH2	0x00	定时器 2 高 8 位	5.5.5	
0xF9	GPIO_INTR_POLARITY	0x0	GPIO 中断极性选择	5.6.1	
0xFE	GPIO_MODE_SEL	0x0	GPIO port 选择	5.6.1	
0xFF	GPIO_CTRL	0x0	GPIO 模式选择	5.6.1	
0x8F	FLASH_CTRL	0x00	Flash 控制寄存器	6.1.3	
0x91	FLASH_DATA	0x00	Flash 数据寄存器	6.1.3	
0x96	FLASH_STATUS	0x00	Flash 状态寄存器	6.1.3	
0xAB	FLASH_COPY	0x00	Flash 复制寄存器	6.1.3	
0xBE	FLASH_COPY_PAGES	0x80	Flash 页复制寄存器	6.1.3	

地址	名称	复位值	功能简介	索引
0xD5	Flash Memory XOR	0x7B	Flash 异或寄存器	6.1.3
0xD7	Program Memory Lock	0x00	存储编程锁定寄存器	6.1.3
0xE1	SPSTA	00H	SPI 状态寄存器	6.2.4
0xE2	SPCON	14H	SPI 控制寄存器	6.2.4
0xE3	SPDAT	00H	SPI 数据寄存器	6.2.4
0xE4	SPSSN	FFH	SPI 从机选择寄存器	6.2.4
0xDA	I2CDAT	0x00	数据寄存器	6.3.2
0xDB	I2CADR	0x00	自身从机地址寄存器	6.3.2
0xDC	I2CCON	0x00	控制寄存器	6.3.2
0xDD	I2CSTA	0xF8	状态寄存器	6.3.2
0xBC	UART_IR_CTRL	0x00	串口红外控制寄存器	6.4.2.3
0xBD	UART_IR_DIV	0x3C	串口红外频分寄存器	6.4.2.3
0x87	PCON	0x00	电源控制寄存器	6.4.5
0x98	S0CON	0x00	串口 0 控制寄存器	6.4.5
0x99	S0BUF	0x00	串口 0 数据缓冲寄存器	6.4.5
0x9B	S1CON	0x00	串口 1 控制寄存器	6.4.5
0x9C	S1BUF	0x00	串口 1 数据缓冲寄存器	6.4.5
0x9D	S1RELL	0x03	串口 1 波特率发生器的重载值低位	6.4.5
0xAA	S0RELL	0xD9	串口 0 波特率发生器的重载值低位	6.4.5
0xBA	S0RELH	0x03	串口 0 波特率发生器的重载值高位	6.4.5
0xBB	S1RELH	0x00	串口 1 波特率发生器的重载值高位	6.4.5
0xD8	ADCON	0x00	串口 0 波特率选择寄存器	6.4.5
0xCE	RTCSEL	0x00	RTC 选择寄存器	6.5.5
0xCF	RTC DATA	0x00	RTC 数据寄存器	6.5.5
0xD1	IEN4	0x00	RTC 中断使能寄存器	6.5.5
0x86	WDTREL	0x00	看门狗加载寄存器	6.6.4
0xF3	ANALOG_CFG1	0x08	模拟前端电路控制寄存器	7.1
0xC9	ANALOG_CFG2	0x00		
0xBF	VCC_LOSS	0x00		
0xF4	AGC_LOOP_CFG1	0x00		
0xF7	AGC_LOOP_CFG2	0x00		
0xF8	AGC_LOOP_CFG3	0x00		
0xFD	AGC_LOOP_STATUS1	0x00		
0xD4	AGC_LOOP_STATUS2	0x00		
0xAF	ALC_CONTROL	0x00		

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [Microprocessors - MPU category](#):

Click to view products by [HiTrend Tech manufacturer](#):

Other Similar products are found below :

[MC7457RX1267LC](#) [MCIMX6D5EYM12AD](#) [A2C00010998 A](#) [ALXD800EEXJCVD C3](#) [LS1020ASE7KQB](#) [LS1020ASN7KQB](#)
[LS1020AXE7KQB](#) [A2C00010729 A](#) [T1022NSE7MQB](#) [MPC8313EVRADDC](#) [BOXSTCK1A8LFCL](#) [LS1021ASE7KQB](#) [LS1021ASN7KQB](#)
[T1024NXN7MQA](#) [T1040NSN7PQB](#) [MCIMX6L3EVN10AB](#) [T1024NXE7MQA](#) [T1023NSE7PQA](#) [CM8063501521600S R19L](#) [NS7520B-1-](#)
[C55](#) [LS1043AXE7MQB](#) [T1024NXN7PQA](#) [LS1012AXE7HKA](#) [T4240NSN7PQB](#) [MVF30NN152CKU26](#) [FH8067303534005S R3ZM](#)
[R9A07G044L24GBG#AC0](#) [SVF311R3K2CKU2](#) [HW8076502640002S R38F](#) [R7S721030VLFP#AA0](#) [MCF5208CVM166](#)
[MCIMX6L2EVN10AB](#) [MCIMX6U7CVM08AC](#) [TEN54LSDV23GME](#) [MC68302AG33C](#) [MC68302EH16C](#) [MCF5233CVM150](#)
[MCIMX6G1CVM05AB](#) [MPC8314ECVRAGDA](#) [MPC8315VRAGDA](#) [MPC8541VTAPF](#) [NS9750B-0-C125](#) [PIC16F1828-I/SS](#) [W78C032C-](#)
[40PL](#) [PIC16F690T-I/SS](#) [PIC18F13K50-I/SS](#) [PIC16F877-20/PQ](#) [PIC16F727-I/PT](#) [PIC16F1847-I/SS](#) [PIC16F1823-I/SL](#)