

产品名称	HR_C7000
文档描述	用户指南
密 级	内部公开

## HR\_C7000芯片用户指南

拟制 \_\_\_\_\_ 谢国军 \_\_\_\_\_ 日期 \_\_\_\_\_ 2016-9-13 \_\_\_\_\_  
审核人 \_\_\_\_\_  
批准 \_\_\_\_\_



浙江大华技术股份有限公司

## 格式编排规定：

(1) 编写文档时，请注意“（）”中的提示信息及其他斜体文字，“XXX”表示该部分可由作者自定义。

(2) 如果作者认为当前章节有扩充的必要，可以增加子章节，并自拟题目。

(3) 请务必遵守文档模版中的样式信息。

(4) 目录中必须包含：标题 1，标题 2，标题 3；标题 4 是否包含在目录中由作者决定。

(5) 请注意保持图形、表格风格一致（如：不同图形中模块方框的大小基本均匀，线条和图形的基本色为黑色，文字的基本字体字号一致）。

### 文档修改记录

修订人	完成日期	修订内容	版本号
谢国军	2016-9-13	初稿	V1.0
谢国军	2017-2-14	FullMask 版本修改，IO 封装描述	V2.0
谢国军	2017-2-22	外设接口描述修改，SPI、UART、GPIO 等	V2.1
谢国军	2017-3-15	FullMask 版本，pinmap 图修改	V2.2
谢国军	2017-4-17	FullMask 版本，Ram 空间变为 288K，对应地址空间修改	V2.3
谢国军	2017-6-16	pinmap 图更新最新	V2.4
谢国军	2017-6-27	Pinmap 管脚描述位图错误修正，USB 手册寄存器描述添加	V2.5
谢国军	2017-7-14	语音 216 比特信息获取对应地址空间描述，详见章节 10.4.2	V2.6
谢国军	2017-7-19	RF_RECV_DELAY 寄存器描述修改，新增 AF 模式下参数设置说明。	V2.7

# 目录

HR_C7000 芯片用户指南 .....	1
<b>1 前言 .....</b>	<b>31</b>
1.1 概述 .....	31
1.2 表格内容约定 .....	31
1.3 寄存器访问类型约定 .....	31
1.4 数值单位约定 .....	31
<b>2 产品概述 .....</b>	<b>32</b>
2.1 应用场景 .....	32
2.2 架构 .....	32
2.2.1 概述 .....	32
2.2.2 处理器内核 .....	33
2.2.3 DMR 标准 .....	33
2.2.4 模拟 FM .....	33
2.2.5 调制解调及信道编解码 .....	34
2.2.6 模拟 IP .....	34
2.2.7 声码器支持 .....	34
2.2.8 射频接口 .....	34
2.2.9 外围接口 .....	34
2.2.10 外部存储器接口 .....	34
2.3 启动模式 .....	34
<b>3 硬件特性 .....</b>	<b>35</b>
3.1 HR_C7000 封装 .....	35
3.2 HR_C7000 管脚分布 .....	37
3.3 HR_C7000 管脚排列表 .....	38
3.4 HR_C7000 管脚描述 .....	41
3.4.1 系统管脚 .....	41
3.4.2 NFC 管脚 .....	42
3.4.3 JTAG 管脚 .....	42
3.4.4 I2C 管脚 .....	42
3.4.5 UART 管脚 .....	43
3.4.6 PWM 管脚 .....	43
3.4.7 SPI 管脚 .....	43
3.4.8 SDIO 管脚 .....	44
3.4.9 I8080 (LCD) 管脚 .....	45
3.4.10 GPIO 管脚 .....	45
3.4.11 AK2401 管脚 .....	46
3.4.12 ADC 管脚 .....	47
3.4.13 DAC 管脚 .....	48
3.4.14 Codec 管脚 .....	48
3.4.15 电源和地管脚 .....	48

3.5	电性能参数 .....	50
3.5.1	电压参数.....	50
3.5.2	功耗参数.....	50
3.5.3	温度参数.....	50
3.5.4	IP 指标参数.....	51
4	系统.....	52
4.1	复位 .....	52
4.1.1	概述.....	52
4.1.2	功能描述.....	52
4.1.2.1	POR.....	53
4.1.2.2	软复位.....	54
4.1.3	寄存器概述.....	55
4.1.4	寄存器描述.....	55
4.1.4.1	SYS_SOFT_RSTN.....	55
4.2	时钟 .....	55
4.2.1	概述.....	55
4.2.2	功能描述.....	56
4.2.2.1	时钟接口.....	57
4.2.2.2	时钟频率.....	57
4.2.3	工作方式.....	57
4.2.3.1	初始化.....	57
4.2.3.2	重配置操作流程.....	57
4.2.3.3	PLL 配置 .....	59
4.2.4	寄存器概述.....	60
4.2.5	寄存器描述.....	60
4.2.5.1	CLK_MGR_REG0x04.....	60
4.2.5.2	CLK_MGR_REG0x08.....	61
4.2.5.3	CLK_MGR_REG0x0c.....	62
4.2.5.4	CLK_MGR_REG0x10.....	62
4.2.5.5	CLK_MGR_REG0x14.....	62
4.2.5.6	CLK_MGR_REG0x18.....	63
4.2.5.7	CLK_MGR_REG0x1c.....	63
4.2.5.8	CLK_MGR_REG0x20.....	63
4.2.5.9	CLK_MGR_REG0x24.....	63
4.2.5.10	CLK_MGR_REG0x28.....	64
4.2.5.11	CLK_MGR_REG0x2c.....	64
4.3	低功耗控制 .....	65
4.4	系统控制 .....	67
4.4.1	概述.....	67
4.4.2	特点.....	67
4.4.3	寄存器概述.....	67
4.4.4	寄存器描述.....	68
4.4.4.1	LCSFC_BAUDR .....	68
4.4.4.2	RTC_REQ_HOLD .....	68

4.4.4.3	QUAD_ENABLE .....	68
4.4.4.4	IO_DIPLEX0.....	68
4.4.4.5	IO_DIPLEX1.....	70
4.4.4.6	IO_DIPLEX2.....	72
4.4.4.7	IOMGR_REN_REG0.....	74
4.4.4.8	IOMGR_REN_REG1.....	76
4.4.4.9	IOMGR_REN_REG2.....	78
4.4.4.10	IOMGR_IE_IE0.....	81
4.4.4.11	IOMGR_IE_IE1 .....	83
4.4.4.12	IOMGR_IE_IE2.....	85
4.5	处理器子系统 .....	88
4.5.1	概述.....	88
4.5.2	特点.....	88
4.5.3	地址映射关系.....	89
4.5.4	启动.....	89
4.5.5	JTAG 调试.....	90
4.6	基带通信系统 .....	91
4.6.1	概述.....	91
4.6.2	特点.....	91
4.6.3	中断向量.....	92
4.6.4	寄存器概述.....	94
4.6.5	寄存器描述.....	95
4.6.5.1	DAC_CONTROL.....	95
4.6.5.2	ADC_CONTROL.....	95
4.6.5.3	AUDIO_CONTROL .....	96
4.6.5.4	AUDIO_BUFFER_CLR.....	97
4.6.5.5	LINEOUT_CTRL.....	97
4.6.5.6	CODEC_I2C_MUX.....	97
4.6.5.7	WORK_MODE .....	97
4.6.5.8	RF_MODE .....	98
4.6.5.9	SYS_INTERP_LIST .....	100
4.6.5.10	SYS_INTERP_MASK .....	100
4.6.5.11	SYS_INTERP_CLEAR .....	102
4.6.5.12	LAYER3_INTERP_LIST .....	102
4.6.5.13	LAYER3_INTERP_MASK .....	103
4.6.5.14	LAYER3_INTERP_CLEAR .....	104
4.6.5.15	INTERP_CLEAR.....	105
4.7	中断系统 .....	105
4.7.1	概述.....	105
4.7.2	功能描述.....	105
4.7.3	中断源分配.....	106
4.7.4	工作方式.....	107
4.7.4.1	初始化.....	107
4.7.4.2	中断状态查询.....	107

4.7.5	寄存器概述.....	107
4.7.6	寄存器描述.....	108
4.7.6.1	PIC_MODE.....	108
4.7.6.2	PIC_PO.....	108
4.7.6.3	PIC_MASK.....	108
4.7.6.4	PIC_VECTOR.....	109
4.7.6.5	PIC_COW1.....	109
4.7.6.6	PIC_PRIOR0-3.....	109
4.7.6.7	PIC_PRIOR4-7.....	110
4.7.6.8	PIC_PRIOR8-11.....	110
4.7.6.9	PIC_PRIOR12-15.....	110
4.7.6.10	PIC_PRIOR16-19.....	110
4.7.6.11	PIC_PRIOR20-23.....	110
4.7.6.12	PIC_PRIOR24-27.....	111
4.7.6.13	PIC_PRIOR28-31.....	111
4.7.6.14	PIC_COW2.....	111
4.7.6.15	PIC_SYNC.....	111
4.7.6.16	PIC_FFLAG_L.....	112
4.7.6.17	PIC_INT_RECORD_SEL.....	112
4.7.6.18	PIC_INT_ST.....	112
4.7.6.19	PIC_INT_ST_1.....	112
4.7.6.20	PIC_INT_ST_CNT.....	112
4.7.6.21	PIC_MODE_1.....	113
4.7.6.22	PIC_PO_1.....	113
4.7.6.23	PIC_MASK_1.....	113
4.7.6.24	PIC_PRIOR32-35.....	113
4.7.6.25	PIC_PRIOR36-39.....	114
4.7.6.26	PIC_PRIOR40-43.....	114
4.7.6.27	PIC_PRIOR44-47.....	114
4.7.6.28	PIC_PRIOR48-51.....	114
4.7.6.29	PIC_PRIOR52-55.....	114
4.7.6.30	PIC_PRIOR56-59.....	115
4.7.6.31	PIC_PRIOR60-63.....	115
4.7.6.32	PIC_FFLAG_H.....	115
4.8	定时器.....	115
4.8.1	概述.....	115
4.8.2	功能描述.....	116
4.8.3	工作方式.....	116
4.8.4	寄存器概述.....	116
4.8.5	寄存器描述.....	117
4.8.5.1	Timer1LoadCount.....	117
4.8.5.2	Timer1CurrentValue.....	117
4.8.5.3	Timer1ControlReg.....	117
4.8.5.4	Timer1EOI.....	117

4.8.5.5	Timer1IntStatust.....	118
4.8.5.6	Timer2LoadCount.....	118
4.8.5.7	Timer2CurrentValue .....	118
4.8.5.8	Timer2ControlReg .....	118
4.8.5.9	Timer2EOI .....	118
4.8.5.10	Timer2IntStatust.....	119
4.8.5.11	Timer3LoadCount.....	119
4.8.5.12	Timer3CurrentValue .....	119
4.8.5.13	Timer3ControlReg .....	119
4.8.5.14	Timer3EOI .....	119
4.8.5.15	Timer3IntStatust.....	119
4.8.5.16	Timer4LoadCount.....	120
4.8.5.17	Timer4CurrentValue .....	120
4.8.5.18	Timer4ControlReg .....	120
4.8.5.19	Timer4EOI .....	120
4.8.5.20	Timer4IntStatust.....	120
4.8.5.21	Timer5LoadCount.....	121
4.8.5.22	Timer5CurrentValue .....	121
4.8.5.23	Timer5ControlReg .....	121
4.8.5.24	Timer5EOI .....	121
4.8.5.25	Timer5IntStatust.....	121
4.8.5.26	Timer6LoadCount.....	121
4.8.5.27	Timer6CurrentValue .....	122
4.8.5.28	Timer6ControlReg .....	122
4.8.5.29	Timer6EOI .....	122
4.8.5.30	Timer6IntStatust.....	122
4.8.5.31	TimersIntStatus.....	122
4.8.5.32	TimersEOI.....	123
4.8.5.33	TimersRawIntStatus .....	123
4.8.5.34	TIMERS_COMP_VERSION .....	123
4.9	RTC.....	123
4.9.1	概述.....	123
4.9.2	功能描述.....	123
4.9.3	工作方式.....	123
4.9.3.1	RTC 初始化 .....	124
4.9.3.2	中断处理.....	124
4.9.4	寄存器概述.....	124
4.9.5	寄存器描述.....	125
4.9.5.1	RTC_VAL_S .....	125
4.9.5.2	RTC_VAL_M .....	125
4.9.5.3	RTC_VAL_H.....	125
4.9.5.4	RTC_VAL_D_L.....	125
4.9.5.5	RTC_VAL_D_H.....	126
4.9.5.6	RTC_MATCH_S.....	126

4.9.5.7	RTC_MATCH_M.....	126
4.9.5.8	RTC_MATCH_H.....	126
4.9.5.9	RTC_MATCH_D_L.....	126
4.9.5.10	RTC_MATCH_D_H.....	126
4.9.5.11	RTC_MATCH_VEN.....	127
4.9.5.12	RTC_MATCH_VSTAT.....	127
4.9.5.13	RTC_LOAD_S.....	127
4.9.5.14	RTC_LOAD_M.....	127
4.9.5.15	RTC_LOAD_H.....	127
4.9.5.16	RTC_LOAD_D_L.....	127
4.9.5.17	RTC_LOAD_D_H.....	128
4.9.5.18	RTC_CCR.....	128
4.9.5.19	RTC_LOAD_VSTAT.....	128
4.9.5.20	RTC_RSTAT.....	128
4.9.5.21	RTC_CLKDIV_NUMER_L.....	128
4.9.5.22	RTC_CLKDIV_NUMER_H.....	129
4.9.5.23	RTC_CLKDIV_NUMER_VLD.....	129
4.9.5.24	RTC_SOFT_RST_N.....	129
4.10	看门狗.....	129
4.10.1	概述.....	129
4.10.2	功能描述.....	129
4.10.3	工作方式.....	130
4.10.3.1	开启看门狗.....	130
4.10.3.2	关闭看门狗.....	130
4.10.3.3	正常工作.....	130
4.10.4	寄存器概述.....	131
4.10.5	寄存器描述.....	131
4.10.5.1	WDG_LOCK.....	131
4.10.5.2	WDG_OT_LOAD_H.....	131
4.10.5.3	WDG_OT_LOAD_L.....	131
4.10.5.4	WDG_RST_LOAD.....	132
4.10.5.5	WDG_EN.....	132
4.10.5.6	WDG_KNOCK.....	132
5	外围设备.....	132
5.1	I <sup>2</sup> C.....	132
5.1.1	概述.....	132
5.1.2	功能描述.....	132
5.1.3	时钟频率配置.....	133
5.1.4	工作方式.....	133
5.1.4.1	初始化.....	133
5.1.4.2	数据发送和接收.....	134
5.1.4.3	停用 I <sup>2</sup> C 控制器.....	134
5.1.5	寄存器概述.....	135
5.1.6	寄存器描述.....	136

5.1.6.1	IC_CON .....	136
5.1.6.2	IC_TAR .....	137
5.1.6.3	IC_DATA_CMD .....	137
5.1.6.4	IC_FS_SCL_HCNT .....	137
5.1.6.5	IC_FS_SCL_LCNT .....	137
5.1.6.6	IC_INTR_STAT .....	138
5.1.6.7	IC_INTR_MASK .....	138
5.1.6.8	IC_RAW_INTR_STAT .....	138
5.1.6.9	IC_RX_TL .....	139
5.1.6.10	IC_TX_TL .....	139
5.1.6.11	IC_CLR_INTR .....	139
5.1.6.12	IC_CLR_RX_UNDER .....	140
5.1.6.13	IC_CLR_RX_OVER .....	140
5.1.6.14	IC_CLR_TX_OVER .....	140
5.1.6.15	IC_CLR_RD_REQ .....	140
5.1.6.16	IC_CLR_TX_ABRT .....	140
5.1.6.17	IC_CLR_RX_DONE .....	141
5.1.6.18	IC_CLR_ACTIVITY .....	141
5.1.6.19	IC_CLR_STOP_DET .....	141
5.1.6.20	IC_CLR_START_DET .....	141
5.1.6.21	IC_ENABLE .....	141
5.1.6.22	IC_STATUS .....	142
5.1.6.23	IC_TXFLR .....	142
5.1.6.24	IC_RXFLR .....	142
5.1.6.25	IC_TX_ABRT_SOURCE .....	143
5.1.6.26	IC_ENABLE_STATUS .....	144
5.1.6.27	IC_START .....	144
5.2	SPI .....	144
5.2.1	概述 .....	144
5.2.2	功能描述 .....	144
5.2.2.1	功能特性 .....	144
5.2.2.2	典型应用 .....	145
5.2.2.3	时钟极性和相位可配置 .....	145
5.2.3	工作方式 .....	147
5.2.3.1	传输时序 .....	147
5.2.3.2	时钟与复位 .....	147
5.2.3.3	操作流程 .....	148
5.2.3.4	SPI 模块 4 种传输模式 .....	148
5.2.4	寄存器概述 .....	150
5.2.5	寄存器描述 .....	152
5.2.5.1	CTRLR0 .....	152
5.2.5.2	CTRLR1 (仅用于 master) .....	153
5.2.5.3	SSIENR .....	153
5.2.5.4	SER (仅用于 master) .....	153

5.2.5.5	BAUDR (仅用于 master)	153
5.2.5.6	TXFTLR	154
5.2.5.7	RXFTLR	154
5.2.5.8	TXFLR	154
5.2.5.9	RXFLR	154
5.2.5.10	SR	154
5.2.5.11	IMR	155
5.2.5.12	ISR	155
5.2.5.13	RISR	155
5.2.5.14	TXOICR	155
5.2.5.15	RXOICR	156
5.2.5.16	RXUICR	156
5.2.5.17	ICR	156
5.2.5.18	IDR	156
5.2.5.19	SSI_COMP_VERSION	156
5.2.5.20	DR	156
5.3	I80	157
5.3.1	概述	157
5.3.2	功能描述	157
5.3.3	工作方式	157
5.3.4	寄存器概述	157
5.3.5	寄存器描述	158
5.3.5.1	INDEX	158
5.3.5.2	DATA	158
5.3.5.3	WCFG	158
5.3.5.4	RCFG	158
5.3.5.5	SCFG	159
5.3.5.6	AC_MODE	159
5.4	UART	159
5.4.1	概述	159
5.4.2	功能描述	159
5.4.3	工作方式	160
5.4.4	寄存器概述	162
5.4.5	寄存器描述	163
5.4.5.1	RBR	163
5.4.5.2	THR	164
5.4.5.3	DLL	164
5.4.5.4	DLH	164
5.4.5.5	IER	164
5.4.5.6	IIR	165
5.4.5.7	FCR	166
5.4.5.8	LCR	166
5.4.5.9	MCR	167
5.4.5.10	LSR	167

5.4.5.11	USR	168
5.4.5.12	TFL	169
5.4.5.13	RFL	169
5.5	SDIO	169
5.5.1	概述	169
5.5.2	功能描述	169
5.5.3	工作方式	170
5.5.3.1	总体流程	170
5.5.3.2	电源管理	170
5.5.3.3	时钟控制	170
5.5.3.4	中断控制	171
5.5.3.5	状态检测	171
5.5.3.6	设备枚举	171
5.5.3.7	指令操作	173
5.5.3.8	数据传输	174
5.5.3.9	Auto-stop 操作	178
5.5.3.10	停止数据传输	178
5.5.3.11	擦除操作	178
5.5.4	寄存器概述	179
5.5.5	寄存器描述	180
5.5.5.1	CTRL(控制寄存器)	180
5.5.5.2	PWREN (上电使能寄存器)	181
5.5.5.3	CLKDIV (时钟分频寄存器)	181
5.5.5.4	CLKSRC(时钟源寄存器)	181
5.5.5.5	CLKENA(时钟使能寄存器)	182
5.5.5.6	TMOUT(Timeout 寄存器)	182
5.5.5.7	CTYPE(卡类型寄存器)	182
5.5.5.8	BLKSIZ(块大小寄存器)	182
5.5.5.9	BYTCNT (字节计数寄存器)	182
5.5.5.10	INTMASK (中断掩码寄存器)	183
5.5.5.11	CMDARG (command_argument 寄存器)	183
5.5.5.12	CMD (命令寄存器)	183
5.5.5.13	RESP0 (响应回复寄存器 0)	184
5.5.5.14	RESP1 (响应回复寄存器 1)	185
5.5.5.15	RESP2 (响应回复寄存器 2)	185
5.5.5.16	RESP3 (响应回复寄存器 3)	185
5.5.5.17	MINTSTS (屏蔽中断寄存器)	185
5.5.5.18	RINTSTS (原始中断状态寄存器)	186
5.5.5.19	STATUS (状态寄存器)	186
5.5.5.20	FIFOTH (FIFO 门限寄存器)	187
5.5.5.21	CDETECT (卡检测寄存器)	188
5.5.5.22	WRTPRT (写保护寄存器)	188
5.5.5.23	TCBCNT (CIU 模块传输的 byte 个数寄存器)	188
5.5.5.24	TBBCNT (Host 与 BIU_FIFO 交互数据 byte 个数寄存器)	188

5.5.5.25	DEBNCE (卡检测反跳寄存器 (单位为 host clock 数目))	188
5.5.5.26	USRID (用户 ID 寄存器)	189
5.5.5.27	VERID (软核版本号寄存器)	189
5.5.5.28	HCON (软核配置寄存器)	189
5.5.5.29	RST_n (卡复位寄存器)	189
5.5.5.30	CardThrCtl (读门限使能寄存器)	189
5.5.5.31	Back_end_power (Backend Power 寄存器)	190
5.5.5.32	ENABLE_SHIFT (相位调整控制寄存器)	190
5.5.5.33	DATA (FIFO 读写入口地址)	190
5.6	USB	190
5.6.1	概述	190
5.6.2	功能描述	190
5.6.3	工作方式	191
5.6.4	寄存器概述	193
5.6.5	寄存器描述	196
5.6.5.1	GAHBCFG	196
5.6.5.2	GUSBCFG	197
5.6.5.3	GRSTCTL	197
5.6.5.4	GINTSTS	198
5.6.5.5	GINTMSK	199
5.6.5.6	GRXSTSR/ GRXSTSP	200
5.6.5.7	GRXFSIZ	201
5.6.5.8	GNPTXFSIZ	201
5.6.5.9	GGPIO	201
5.6.5.10	GUID	202
5.6.5.11	GSYNPSID	202
5.6.5.12	GHWCFG1	203
5.6.5.13	GHWCFG2	203
5.6.5.14	GHWCFG3	203
5.6.5.15	GHWCFG4	204
5.6.5.16	GUSRSTS	204
5.6.5.17	DIEPTXF1/3	205
5.6.5.18	DCFG	205
5.6.5.19	DCTL	206
5.6.5.20	DSTS	207
5.6.5.21	DIEPMSK	207
5.6.5.22	DOEPMSK	208
5.6.5.23	DAINT	208
5.6.5.24	DAINTMSK	209
5.6.5.25	DTHRCTL	209
5.6.5.26	DIEPMPMSK	209
5.6.5.27	DIEPCTL0	210
5.6.5.28	DIEPINT0/1/3	211
5.6.5.29	DIEPTSIZ0	211

5.6.5.30	DIEPDMA0/1/3.....	212
5.6.5.31	DTXFSTS0/1/3.....	212
5.6.5.32	DOEPCTL0.....	212
5.6.5.33	DOEPINT0/2/4.....	213
5.6.5.34	DOEPTSIZE0.....	214
5.6.5.35	DOEPDMA0/2/4.....	214
5.6.5.36	DIEPCTL1/3.....	214
5.6.5.37	DIEPTSIZE1/3.....	216
5.6.5.38	DOEPCTL2/4.....	216
5.6.5.39	DOEPTSIZE2/4.....	217
5.6.5.40	PCGCCTL.....	218
5.7	GPIO.....	218
5.7.1	概述.....	218
5.7.2	功能描述.....	218
5.7.3	工作方式.....	218
5.7.4	寄存器概述.....	219
5.7.5	寄存器描述.....	220
5.7.5.1	SWPORTA_DR.....	220
5.7.5.2	SWPORTA_DDR.....	220
5.7.5.3	INTEN.....	221
5.7.5.4	INTMASK.....	221
5.7.5.5	INTTYPE_LEVEL.....	221
5.7.5.6	INT_POLARITY.....	221
5.7.5.7	INTSTATUS.....	221
5.7.5.8	RAW_INTSTATUS.....	221
5.7.5.9	DEBOUNCE.....	222
5.7.5.10	PORTA_EOI.....	222
5.7.5.11	EXT_PORTA.....	222
5.7.5.12	INTR_BOTH_EDGE.....	222
5.8	PWM.....	222
5.8.1	概述.....	222
5.8.2	功能描述.....	222
5.8.3	工作方式.....	223
5.8.3.1	Continuous 模式.....	223
5.8.3.2	One Shot 模式.....	224
5.8.4	寄存器概述.....	224
5.8.5	寄存器描述.....	225
5.8.5.1	PWM0_CFG.....	225
5.8.5.2	PWM0_RPT.....	225
5.8.5.3	PWM0_PER.....	226
5.8.5.4	PWM0_FP.....	226
5.8.5.5	PWM0_STATUS.....	226
5.8.5.6	PWM1_CFG.....	226
5.8.5.7	PWM1_RPT.....	227

5.8.5.8	PWM1_PER.....	227
5.8.5.9	PWM1_FP.....	227
5.8.5.10	PWM1_STATUS.....	227
5.8.5.11	PWM2_CFG.....	228
5.8.5.12	PWM2_RPT.....	228
5.8.5.13	PWM2_PER.....	229
5.8.5.14	PWM2_FP.....	229
5.8.5.15	PWM1_STATUS.....	229
5.8.5.16	PWM_INTR_STA.....	229
5.9	ADC 控制器.....	229
5.9.1	概述.....	229
5.9.2	功能描述.....	230
5.9.3	工作方式.....	230
5.9.3.1	单次扫描处理流程.....	230
5.9.3.2	单次扫描软件操作流程:.....	232
5.9.3.3	连续扫描处理流程.....	233
5.9.3.4	连续扫描软件操作流程.....	234
5.9.3.5	滤毛刺流程.....	235
5.9.3.6	低功耗流程.....	235
5.9.4	寄存器概述.....	235
5.9.5	寄存器描述.....	236
5.9.5.1	ADC_CTRL.....	236
5.9.5.2	ADC_INTR_DELTA.....	236
5.9.5.3	ADC_INTR.....	236
5.9.5.4	ADC_INTR_STA.....	237
5.9.5.5	ADC_SCAN_TIME.....	238
5.9.5.6	ADC_START.....	238
5.9.5.7	ADC_CTRL_STATE.....	238
5.9.5.8	ADC_INTR_THRESHOLD.....	239
5.9.5.9	ADC_CTRL_STOP.....	239
5.9.5.10	ADC_CH_VLD.....	240
5.9.5.11	ADC_PD_SEOC_TIME.....	240
5.9.5.12	ADC_P2S_EN.....	240
5.9.5.13	ADC_DATA_AB.....	240
5.9.5.14	ADC_DATA_CD.....	241
5.9.5.15	ADC_DATA_EF.....	241
5.9.5.16	ADC_DATA_GH.....	241
5.9.5.17	ADC_CTRL_VERID.....	241
5.9.5.18	ADC_DIR_DAT_OP.....	241
5.10	DAC 控制器.....	242
5.10.1	概述.....	242
5.10.2	功能描述.....	242
5.10.3	工作方式.....	242
5.10.4	寄存器概述.....	242

5.10.5 寄存器描述.....	242
5.10.5.1 DAC_PD_CTRL.....	242
5.10.5.2 DAC_PD_MODE_EN.....	242
5.10.5.3 DAC_DATA_A.....	243
5.10.5.4 DAC_DATA_B.....	243
5.10.5.5 DAC_DATA_C.....	243
5.10.5.6 DAC_CTRL_VERID.....	243
<b>6 CODEC 应用.....</b>	<b>244</b>
6.1 概述.....	244
6.2 功能描述.....	244
6.3 参考电路.....	245
6.3.1 MIC 输入.....	245
6.3.2 LINEOUT 输出.....	245
6.4 工作方式.....	246
6.4.1 工作模式描述.....	246
6.4.1.1 Power-off mode.....	247
6.4.1.2 Power-down mode.....	247
6.4.1.3 Playback（语音播放）模式.....	247
6.4.1.4 Sleep mode.....	247
6.4.1.5 Analog record mode.....	247
6.4.2 工作模式切换.....	247
6.4.2.1 Power-down 模式切换到语音播放模式.....	247
6.4.2.2 Power-down 模式切换到 sleep 模式.....	248
6.4.2.3 Sleep 模式到 CODEC ADC 的开启.....	248
6.4.2.4 Sleep 模式到音频播放模式.....	248
6.4.2.5 任意模式切换到 power-down 模式.....	249
6.4.3 Codec 工作流程.....	249
6.4.3.1 音频接口模式设置.....	249
6.4.3.2 音频发送（Analog record）.....	249
6.4.3.3 音频接收（Play back）.....	250
6.4.4 软静音功能（soft mute）.....	251
6.4.5 AGC.....	251
6.4.6 Soft clipping DRC.....	252
6.4.7 Wind Noise Filter.....	252
6.5 寄存器.....	253
6.5.1 寄存器概述.....	253
6.5.2 寄存器描述.....	253
6.5.2.1 SR1.....	253
6.5.2.2 SR2.....	254
6.5.2.3 MR.....	254
6.5.2.4 AICR_DAC.....	254
6.5.2.5 AICR_ADC.....	255
6.5.2.6 CR_MIC1.....	255
6.5.2.7 CR_MIC2.....	256

6.5.2.8	CR_DAC	256
6.5.2.9	CR_ADC	256
6.5.2.10	CR_MIX	256
6.5.2.11	DR_MIX	258
6.5.2.12	CR_VIC	259
6.5.2.13	CR_CK	259
6.5.2.14	FCR_DAC	259
6.5.2.15	SFCCR_DAC	259
6.5.2.16	SFFCR_DAC	259
6.5.2.17	FCR_ADC	260
6.5.2.18	CR_TIMER_h	260
6.5.2.19	CR_TIMER_l	260
6.5.2.20	ICR	260
6.5.2.21	IMR	261
6.5.2.22	IFR	261
6.5.2.23	IMR2	261
6.5.2.24	IFR2	261
6.5.2.25	GCR_DACL	262
6.5.2.26	GCR_DACR	262
6.5.2.27	GCR_MIC1	263
6.5.2.28	GCR_ADCL	263
6.5.2.29	GCR_ADCR	263
6.5.2.30	GCR_MIXDACL	263
6.5.2.31	GCR_MIXDACR	264
6.5.2.32	GCR_MIXADCL	264
6.5.2.33	GCR_MIXADCRCR	264
6.5.2.34	CR_DAC_AGC	265
6.5.2.35	DR_DAC_AGC	265
6.5.2.36	CR_ADC_AGC	266
6.5.2.37	DR_ADC_AGC	266
<b>7</b>	<b>音频通路</b>	<b>268</b>
7.1	DMR 模式	268
7.1.1	概述	268
7.1.2	工作方式	269
7.1.2.1	数字音频发送	269
7.1.2.2	数字音频接收	269
7.1.2.3	数字通话录音功能	270
7.1.2.3.1	数字发送录音	270
7.1.2.3.2	数字接收录音	270
7.2	FM 模式	270
7.2.1	概述	270
7.2.2	工作方式	270
7.2.2.1	模拟音频发送	270
7.2.2.1.1	音频源为 CODEC ADC	271

7.2.2.1.2	音频源为 TX RAM .....	271
7.2.2.2	模拟音频接收.....	271
7.2.2.3	模拟通话录音功能.....	271
7.2.2.3.1	模拟发送录音 .....	271
7.2.2.3.2	模拟接收录音 .....	271
7.2.2.4	模拟侧音播放.....	272
7.3	提示音通路 .....	272
7.3.1	概述.....	272
7.3.2	功能描述.....	272
7.3.2.1	基于查询方式的提示音写入 .....	272
7.3.2.2	基于中断方式的提示音写入 .....	272
<b>8</b>	<b>射频接口 .....</b>	<b>272</b>
8.1	DAC 发送.....	272
8.1.1	概述.....	272
8.1.2	功能描述.....	272
8.1.3	工作方式.....	273
8.1.3.1	两点调制.....	273
8.1.3.2	单点调制.....	274
8.2	ADC 接收.....	274
8.2.1	概述.....	274
8.2.2	功能描述.....	274
8.2.3	工作方式.....	275
8.2.3.1	AF 接收.....	275
8.2.3.2	中频接收.....	275
8.3	数字接收 .....	276
8.3.1	概述.....	276
8.3.2	功能描述.....	276
8.3.3	工作方式.....	276
<b>9</b>	<b>DMR 物理层应用.....</b>	<b>277</b>
9.1	概述 .....	277
9.2	寄存器 .....	278
9.2.1	寄存器概览.....	278
9.2.2	寄存器描述.....	279
9.2.2.1	WORK_MODE.....	279
9.2.2.2	RF_MODE.....	280
9.2.2.3	SIG_CENTER.....	281
9.2.2.4	TX_IF_FREQ .....	281
9.2.2.5	RF_CONTROL.....	282
9.2.2.6	RF_MOD_BIAS_CTRL .....	282
9.2.2.7	DEV_LIMITER .....	282
9.2.2.8	RSSI_BIAS_VALUE .....	282
9.2.2.9	THRESHOLD_VALUE .....	282
9.2.2.10	PHASE_OUT .....	283

9.2.2.11	SEND_DATA_SYNC_H	283
9.2.2.12	SEND_DATA_SYNC_L	283
9.2.2.13	SEND_RC_SYNC_H	284
9.2.2.14	SEND_RC_SYNC_L	284
9.2.2.15	RECV_MS_SYNC_H	284
9.2.2.16	RECV_MS_SYNC_L	284
9.2.2.17	RECV_BS_SYNC_H	284
9.2.2.18	RECV_BS_SYNC_L	285
9.2.2.19	RECV_TDMA1_SYNC_H	285
9.2.2.20	RECV_TDMA1_SYNC_L	285
9.2.2.21	RECV_TDMA2_SYNC_H	285
9.2.2.22	RECV_TDMA2_SYNC_L	286
9.2.2.23	RECV_RC_SYNC_H	286
9.2.2.24	RECV_RC_SYNC_L	286
9.2.2.25	RECV_INFO_COUNT	286
9.2.2.26	RECVFIR_COEF1	287
9.2.2.27	RECVFIR_COEF2	287
9.2.2.28	RECVFIR_COEF3	287
9.2.2.29	RECVFIR_COEF4	287
9.2.2.30	RECVFIR_COEF5	288
9.2.2.31	RECVFIR_COEF6	288
9.2.2.32	RECVFIR_COEF7	288
9.2.2.33	RECVFIR_COEF8	288
9.2.2.34	RECVFIR_COEF9	289
9.2.2.35	RECVFIR_COEF10	289
9.2.2.36	RECVFIR_COEF11	289
9.2.2.37	RECVFIR_COEF12	289
9.2.2.38	RECVFIR_COEF13	289
9.2.2.39	RECVFIR_COEF14	290
9.2.2.40	RECVFIR_COEF15	290
9.2.2.41	RECVFIR_COEF16	290
9.2.2.42	RECVFIR_COEF17	290
9.2.2.43	RX_IF_FREQ	291
9.2.2.44	RF_RECV_DELAY	291
9.2.2.45	MODEM_AK_CLK_CALB	291
9.2.2.46	AK2401_ADC_STATE	291
9.2.2.47	PHY_RECEIVE_STATE	292
9.2.2.48	LAYER2_CONTROL	292
9.2.2.49	LAYER2_TXRX_CTRL	292
9.3	中断向量	293
9.4	MEMORY 映射	293
9.4.1	物理层发送 Ram	294
9.4.2	物理层接收 Ram	294
9.5	物理层发送	294

9.5.1	概述.....	294
9.5.2	功能描述.....	294
9.5.3	工作方式.....	295
9.6	物理层接收.....	296
9.6.1	概述.....	296
9.6.2	功能描述.....	296
9.6.3	工作方式.....	296
<b>10</b>	<b>DMR 链路层应用.....</b>	<b>297</b>
10.1	概述.....	297
10.2	寄存器.....	297
10.2.1	寄存器概览.....	297
10.2.2	寄存器描述.....	298
10.2.2.1	WORK_MODE.....	298
10.2.2.2	PI_CRC_MASK.....	299
10.2.2.3	RS_H_MASK.....	299
10.2.2.4	RS_H_INIT.....	299
10.2.2.5	RS_T_MASK.....	300
10.2.2.6	RS_T_INIT.....	300
10.2.2.7	CSBK_CRC_MASK.....	300
10.2.2.8	MBCH_CRC_MASK.....	300
10.2.2.9	DATAH_CRC_MASK.....	300
10.2.2.10	CRC8_INIT.....	301
10.2.2.11	LOCAL_CC.....	301
10.2.2.12	VOICE_EMB_CTRL.....	301
10.2.2.13	SCRAMBLE_REG.....	301
10.2.2.14	RX_TYPE_INFO.....	302
10.2.2.15	RX_INFO.....	303
10.2.2.16	LAYER2_CONTROL.....	303
10.2.2.17	LAYER2_SLOTON.....	304
10.2.2.18	LAYER2_TXRX_CTRL.....	305
10.2.2.19	LAYER2_SLOT_CNT.....	306
10.2.2.20	LAYER2_MAX_CNT.....	306
10.2.2.21	LAYER2_SLOT_UPDATE.....	306
10.2.2.22	LAYER2_SEND_TYPE.....	307
10.2.2.23	LAYER2_STATUS.....	309
10.2.2.24	LAYER2_SLOT_PRE.....	309
10.2.2.25	LAYER2_BS_MODE.....	309
10.2.2.26	BS_SECOND_REG.....	309
10.2.2.27	BS_STATUS.....	310
10.2.2.28	BS_CACH.....	310
10.3	中断向量.....	310
10.4	MEMORY映射.....	312
10.4.1	链路层发送 Ram.....	312
10.4.2	链路层接收 Ram.....	313

10.5	链路层时隙管理 .....	314
10.5.1	概述.....	314
10.5.2	发送时隙.....	314
10.5.2.1	发送时隙主动建立模式.....	314
10.5.2.2	发送时隙接收建立模式.....	315
10.5.2.3	发送时隙临时建立模式.....	315
10.5.3	发送临时时隙.....	316
10.5.3.1	临时时隙主动建立模式.....	316
10.5.3.2	临时时隙接收建立模式.....	316
10.5.3.3	临时时隙发送建立模式.....	317
10.5.4	接收时隙.....	317
10.5.4.1	接收时隙主动建立模式.....	317
10.5.4.2	接收时隙被动建立模式.....	317
10.5.5	接收时隙同步策略.....	318
10.5.5.1	发送时隙和接收时隙同步.....	318
10.5.5.2	发送时隙和发送临时时隙同步.....	319
10.5.5.3	发送临时时隙和接收时隙同步.....	322
10.6	链路层发送 .....	324
10.6.1	概述.....	324
10.6.2	帧类型描述.....	324
10.6.2.1	语音帧.....	324
10.6.2.2	数据控制帧.....	325
10.6.2.3	RC 帧.....	327
10.6.3	组帧模式.....	327
10.6.4	帧类型定义.....	328
10.6.5	发送流程.....	329
10.7	链路层接收 .....	330
10.7.1	概述.....	330
10.7.2	功能描述.....	330
10.7.3	接收应用.....	330
<b>11</b>	<b>FM 应用 .....</b>	<b>331</b>
11.1	概述.....	331
11.2	寄存器.....	331
11.2.1	寄存器概览.....	331
11.2.2	寄存器描述.....	332
11.2.2.1	WORK_MODE.....	332
11.2.2.2	FM_BANDWIDTH.....	332
11.2.2.3	FM_DEV_COEF.....	333
11.2.2.4	FM_SQL.....	333
11.2.2.5	SUBVOICE_COEF.....	333
11.2.2.6	FM_MOD.....	333
11.2.2.7	CTCSS_ADDR .....	334
11.2.2.8	USER_DEFINE CTC.....	334
11.2.2.9	CDCSS_ADDR .....	335

11.2.2.10	SELCALL_TONE .....	335
11.2.2.11	TONE_SEND_CTRL .....	335
11.2.2.12	XTCSS_ADDR .....	336
11.2.2.13	FM_FUNCTION_COEF1 .....	336
11.2.2.14	FM_FUNCTION_COEF2 .....	336
11.2.2.15	FM_FUNCTION_COEF3 .....	336
11.2.2.16	FM_FUNCTION_COEF4 .....	336
11.2.2.17	FM_FUNCTION_COEF5 .....	337
11.2.2.18	FM_FUNCTION_COEF6 .....	337
11.2.2.19	FM_FUNCTION_COEF7 .....	337
11.2.2.20	FM_FUNCTION_COEF8 .....	337
11.2.2.21	FM_TONE1 .....	337
11.2.2.22	FM_TONE2 .....	338
11.2.2.23	FM_TONE3 .....	338
11.2.2.24	FM_TONE4 .....	338
11.2.2.25	FM_PTT .....	338
11.2.2.26	FM_DECODE_FLAG1 .....	338
11.2.2.27	FM_DECODE_FLAG2 .....	338
11.2.2.28	FM_ADDR_SW .....	339
11.3	中断向量 .....	339
11.4	MEMORY 映射 .....	340
11.4.1	FM 发送 Ram .....	340
11.4.2	FM 接收 Ram .....	340
11.5	SQL .....	340
11.6	CTCSS .....	340
11.6.1	概述 .....	340
11.6.2	功能描述 .....	340
11.6.2.1	CTCSS 发送 .....	341
11.6.2.2	CTCSS 接收 .....	342
11.6.3	工作方式 .....	344
11.6.3.1	CTCSS 发送 .....	344
11.6.3.2	CTCSS 接收 .....	345
11.7	CDCSS .....	345
11.7.1	概述 .....	345
11.7.2	功能描述 .....	345
11.7.3	工作方式 .....	345
11.7.3.1	CDCSS 发送 .....	345
11.7.3.2	CDCSS 接收 .....	346
11.8	DTMF .....	346
11.8.1	概述 .....	346
11.8.2	功能描述 .....	346
11.8.3	工作方式 .....	347
11.8.3.1	DTMF 发送 .....	347
11.8.3.2	DTMF 接收 .....	348

11.9	SELDCALL.....	349
11.9.1	概述.....	349
11.9.2	功能描述.....	349
11.9.3	工作方式.....	349
11.9.3.1	SELDCALL 发送.....	349
11.9.3.2	SELDCALL 接收.....	350
12	术语.....	350
13	缩略语.....	350
14	附录 A: 数字组呼应用.....	351
14.1	功能描述.....	351
14.2	组呼发送.....	352
14.3	组呼接收.....	353
15	附录 B: 数字短信应用.....	355
15.1	功能描述.....	355
15.2	短信业务时序.....	356
15.3	短信数据帧结构.....	357
15.3.1	数据帧头结构.....	357
15.3.2	数据帧结构.....	359
15.3.3	响应数据挂起时间.....	360
15.3.4	应答数据帧.....	361
15.4	IPv4 结构说明.....	362
15.4.1	IPv4 Packet Data.....	363
15.4.2	IPv4 Response Packet Data.....	364
15.5	非确认短信.....	365
15.5.1	非确认短信发送.....	365
15.5.2	非确认短信接收.....	366
15.6	确认短信.....	368
15.6.1	确认短信发送.....	368
15.6.2	确认短信接收.....	369
16	附录 C: 数字信令应用.....	370
16.1	功能描述.....	370
16.2	信令发送.....	370
16.3	信令接收.....	371
17	附录 D: 数字加密应用.....	372
17.1	概述.....	372
17.2	语音加密.....	373
17.2.1	静态基本加密.....	373
17.2.2	动态增强加密.....	373
17.2.2.1	加密参数定义.....	374
17.2.2.1.1	初始向量.....	374
17.2.2.1.2	KeyID 和 ALGOID.....	376

17.2.2.2	加密参数传递.....	377
17.2.2.2.1	初始密钥传递和加密过程.....	377
17.2.2.2.2	初始密钥接收和解密过程.....	378
17.2.2.3	加密控制帧.....	379
17.2.2.3.1	语音帧头.....	379
17.2.2.3.2	语音 PI 帧.....	380
17.2.2.3.3	语音 F 帧 EMB.....	380
17.3	数据加密.....	380
17.3.1	短信加密时序.....	380
17.3.2	短信加密参数传递.....	381
<b>18</b>	<b>附录 E: 数字 DCDM 应用.....</b>	<b>382</b>
18.1	概述.....	382
18.2	基本原理.....	382
18.2.1	状态机类型.....	382
18.2.1.1	领导者和时隙未知状态.....	382
18.2.1.2	领导者未知状态.....	382
18.2.1.3	领导者和时隙已知状态.....	383
18.2.1.4	领导者状态.....	383
18.2.2	数据帧定义.....	383
18.2.2.1	同步帧定义.....	383
18.2.2.2	色码定义.....	383
18.2.2.3	Channel Timing CSBK.....	383
18.2.2.3.1	CT_CSBK_Beacon.....	384
18.2.2.3.2	CT_CSBK_Prop.....	384
18.2.2.3.3	CT_CSBK_Term.....	384
18.2.2.3.4	CT_CSBK_Req.....	385
18.2.2.3.5	CT_CSBK_Resp.....	385
18.2.3	DCDM 状态流程.....	385
18.2.3.1	领导者和时隙未知状态.....	385
18.2.3.2	领导者和时隙未知状态流程.....	386
18.2.3.3	领导者未知状态流程.....	387
18.2.3.4	领导者和时隙已知状态流程.....	388
18.2.3.5	领导者状态流程.....	389
18.2.3.6	领导者 ID 冲突状态流程.....	390
18.2.3.7	CT_CSBK 信息评估状态流程.....	392
18.2.3.8	发送纠正 CT_CSBK 状态流程.....	392
18.2.3.9	接受领导者状态流程.....	393
18.2.3.10	指定新领导者状态流程.....	394
18.2.3.11	定时推送状态流程.....	395
18.3	应用实例.....	397
18.3.1	基本单元.....	397
18.3.1.1	时隙单元.....	397
18.3.1.2	领导者.....	397
18.3.1.3	世代.....	397

18.3.1.4	同步时期.....	399
18.3.1.5	标识符.....	399
18.3.1.6	信道定时操作码.....	400
18.3.1.7	新领导者标志.....	401
<b>18.3.2</b>	<b>CT_CSBK PDU.....</b>	<b>401</b>
18.3.2.1	CT_CSBK 结构.....	401
18.3.2.2	CT_CSBK 字段定义.....	401
<b>18.3.3</b>	<b>定时器定义.....</b>	<b>402</b>
18.3.3.1	CT_RHOT 说明.....	402
<b>18.3.4</b>	<b>最小化系统举例.....</b>	<b>405</b>
18.3.4.1	单机开机启动.....	405
18.3.4.2	开机领导者确认.....	406
18.3.4.3	信标消息发送.....	407
18.3.4.4	业务发送流程.....	408
18.3.4.5	Leader 脱机.....	409
<b>18.3.5</b>	<b>HR_C7000 应用.....</b>	<b>410</b>
18.3.5.1	同步字段应用.....	410
18.3.5.2	色码应用.....	411
18.3.5.3	时隙管理.....	411

## 图目录

图 1	HR_C7000 单芯片应用场景.....	32
图 2	芯片总体框图.....	33
图 3	HR_C7000 芯片封装顶视图.....	35
图 4	HR_C7000 芯片封装底视图.....	36
图 5	HR_C7000 芯片封装侧视图.....	36
图 6	HR_C7000 管脚分布图.....	38
图 7	复位整体框图.....	53
图 8	复位整体框图.....	53
图 9	POR 模块框图.....	54
图 10	POR 模块时序图.....	54
图 11	时钟分布框图.....	56
图 12	重配置流程图.....	59
图 13	PLL 电路示意图.....	60
图 14	HR_C7000 功能模块划分.....	66
图 15	HR_C7000 功能功耗控制分布.....	66
图 16	HR_C7000 低功耗工作流程.....	67
图 17	CPU 系统整体框图.....	88
图 18	系统启动流程图.....	90
图 19	CK803 JTAG 调试.....	91
图 20	中断控制器功能框图.....	105
图 21	看门狗关闭时序图.....	130

图 22 I <sup>2</sup> C 总线的数据传送过程 .....	133
图 23 主设备 SPI 的典型连接方式 .....	145
图 24 从设备 SPI 的典型连接方式 .....	145
图 25 SCPH=0 数据发送和采样时序 .....	146
图 26 SCPH=1 数据发送和采样时序 .....	147
图 27 SPI 数据串行时序 .....	147
图 28 SPI Transmit & Receive 模式 .....	149
图 29 SPI Transmit Only 模式 .....	149
图 30 SPI Receive Only 模式 .....	150
图 31 SPI EEPROM Read Transfer 模式 .....	150
图 32 8080 接口时序 .....	157
图 33 UART 帧格式 .....	159
图 34 中断方式数据发送流程图 .....	161
图 35 中断方式数据接收流程图 .....	162
图 36 SD 卡的枚举和初始化流程 .....	172
图 37 SD 存储器数据传输状态转移图 .....	175
图 38 SDIO 设备数据传输状态转移图 .....	175
图 39 CTRL OUT 传输时序 .....	191
图 40 CTRL IN 传输时序 .....	192
图 41 无数据 CTRL 传输时序 .....	192
图 42 单次 OUT 事务时序 .....	192
图 43 多次 OUT 事务时序 .....	193
图 44 单次 IN 事务时序 .....	193
图 45 多次 IN 事务时序 .....	193
图 46 PWM 输出时序图 .....	223
图 47 ONE_SHOT 普通模式示意图 .....	223
图 48 CONTINUOUS 模式示意图 .....	223
图 49 CONTINUOUS 模式软件操作示意图 .....	224
图 50 ADC_INTR_THRESHOLE 与 ADC_INTR_DELTA 关系示意图 .....	230
图 51 ADC_CTRL 单次扫描软件操作流程 .....	232
图 52 ADC_CTRL 连续扫描通道轮询示意图 .....	233
图 53 ADC_CTRL 连续扫描软件操作流程 .....	234
图 54 ADC_CTRL 低功耗模式时序简图 .....	235
图 55 Codec 模块功能框图 .....	244
图 56 Codec Mic 差分 and 单端原理图 .....	245
图 57 Codec Micbias 差分 and 单端原理图 .....	245
图 58 Codec PWM 输出原理图 .....	246
图 59 语音发送数据流转图 .....	249
图 60 语音接收数据流转图 .....	250
图 61 CODEC 自动增益控制 (AGC) 系统框图 .....	251
图 62 两点调制接口信号 .....	273
图 63 单点调制接口信号 .....	274
图 64 AF 接收参考接口电路 .....	275
图 65 中频接收参考接口电路 .....	276

图 66 HR_C7000 数字接口示意图.....	276
图 67 数字 SPI 接口时序图.....	277
图 68 DMR 数字基带物理层结构.....	277
图 69 DMR 标准发送帧结构.....	295
图 70 物理层解析数据顺序.....	296
图 71 物理层接收状态流程图.....	297
图 72 DMR 数字基带链路层结构.....	297
图 73 发送时隙主动建立时序图.....	315
图 74 发送时隙接收时隙建立时序图.....	315
图 75 发送时隙临时时隙建立时序图.....	316
图 76 接收时隙被动建立时序图.....	318
图 77 发送时隙同步到接收时隙时序图.....	319
图 78 接收时隙同步到发送时隙时序图.....	319
图 79 发送时隙同步到发送临时时隙时序图.....	322
图 80 接收时隙同步到发送临时时隙时序图.....	324
图 81 语音帧带同步帧结构.....	324
图 82 语音帧带 EMB 数据结构.....	324
图 83 数据帧、控制帧结构.....	325
图 84 控制信道 C_RC 帧结构(PDT).....	325
图 85 独立 RC 帧结构.....	327
图 86 时隙语音时序.....	328
图 87 连续语音时序.....	328
图 88 单时隙数据时序.....	328
图 89 双时隙数据时序.....	328
图 90 链路层发送时序图 1.....	329
图 91 链路层发送时序图 2.....	329
图 92 链路层接收状态流程图.....	330
图 93 CTCSS 发送状态流程图.....	344
图 94 DTMF 发送框图.....	347
图 95 DTMF 码元频率对应关系.....	347
图 96 DTMF 发送时序图.....	348
图 97 SELCALL 发送时序图.....	349
图 98 语音业务结构图.....	351
图 99 组呼发送业务时序图 1.....	352
图 100 组呼发送业务时序图 2.....	352
图 101 组呼接收业务时序图.....	354
图 102 组呼接收状态流程图.....	354
图 103 短信业务应用流程图.....	355
图 104 短信业务时序.....	356
图 105 非确认短信业务结构图.....	356
图 106 确认短信业务结构图.....	357
图 107 数据帧头 PDU 结构.....	357
图 108 非确认 Rate1/2 数据块结构图.....	359
图 109 数据帧尾 PDU 结构图.....	360

图 110 应答数据帧 PDU 结构图.....	361
图 111 反馈数据包体错误图样更新过程.....	362
图 112 IPv4 结构图.....	363
图 113 UDP 校验和计算各字段定义.....	363
图 114 非确认短信发送业务时序图.....	365
图 115 非确认短信接收业务时序图.....	366
图 116 短信接收流程图.....	367
图 117 确认短信发送业务时序图.....	368
图 118 确认短信接收业务时序图.....	369
图 119 CSBK 信令业务结构图.....	370
图 120 CSBK 信令业务发送时序图.....	370
图 121 CSBK 信令业务接收时序图.....	371
图 122 CSBK 信令业务接收流程图.....	372
图 123 静态加密过程示意图.....	373
图 124 动态加密过程示意图.....	373
图 125 PI 帧 PDU 结构图.....	374
图 126 初始向量生成流程.....	375
图 127 初始向量生成 EMB 传输方式.....	376
图 128 KeyID 和算法 ID 语音 F 帧传递方式.....	376
图 129 动态密钥传输更新示意图.....	377
图 130 动态密钥参数加密传递流程图.....	378
图 131 动态密钥参数解密传递流程图.....	379
图 132 非确认短信加密时序图.....	381
图 133 确认短信加密时序图.....	381
图 134 自定义数据帧头 PDU 结构图.....	381
图 135 信道定时 CSBK 终止.....	385
图 136 开机和信道改变状态流程图.....	386
图 137 领导者和时隙未知状态流程图.....	387
图 138 领导者未知状态流程图.....	388
图 139 领导者和时隙已知状态流程图.....	389
图 140 领导者状态流程图.....	390
图 141 领导者 ID 冲突状态流程图.....	391
图 142 CT_CSBK 信息评估状态流程图.....	392
图 143 发送纠正 CT_CSBK 状态流程图.....	393
图 144 接受领导者状态流程图.....	394
图 145 指定新领导者状态流程图.....	395
图 146 定时推送状态流程图.....	396
图 147 DCDM 模式下多个无线电设备覆盖图.....	398
图 148 信道定时扩散时序图.....	399
图 149 CT_CSBK 结构图.....	401
图 150 CT_RHOT 使用流程图.....	404
图 151 单机开机启动流程图.....	405
图 152 开机领导者确认流程图.....	407
图 153 信标消息发送流程图.....	408

图 154 业务发送流程图.....	409
图 155 Leader 脱机后流程图 .....	410
图 156 真双时隙入网状态管理流程图 .....	412

## 表目录

表 1 HR_C7000 芯片封装参数说明表.....	36
表 2 HR_C7000 管脚排列表 .....	38
表 3 I/O 类型说明 .....	41
表 4 上下拉约定.....	41
表 5 系统管脚.....	41
表 6 NFC 管脚.....	42
表 7 JTAG 管脚 .....	42
表 8 I2C 管脚.....	42
表 9 UART 管脚 .....	43
表 10 PWM 管脚.....	43
表 11 SPI 管脚 .....	43
表 12 SDIO 管脚 .....	44
表 13 I8080 管脚 .....	45
表 14 GPIO 管脚 .....	45
表 15 AK2401 管脚.....	46
表 16 ADC 管脚 .....	47
表 17 DAC 管脚 .....	48
表 18 Codec 管脚 .....	48
表 19 电源地管脚.....	48
表 20 电压参数.....	50
表 21 温度参数.....	50
表 22 IP 指标参数 .....	51
表 23 POR 设计参数表.....	54
表 24 时钟接口信号.....	57
表 25 时钟频率范围.....	57
表 26 地址映射关系.....	89
表 27 基带工作模式列表.....	91
表 28 基带通信系统中断列表.....	92
表 29 基带通信系统子中断列表.....	92
表 30 中断源列表.....	106
表 31 看门狗工作状态表.....	130
表 32 I <sup>2</sup> C 接口信号 .....	133
表 33 发送指令寄存器设置.....	173
表 34 SD 存储器读取操作指令设置.....	176
表 35 SD 存储器写入操作指令设置.....	177
表 36 Auto-Stop 支持的卡类型和传输模式 .....	178
表 37 图例符号定义.....	191

表 38 Codec Micbias 参数说明.....	245
表 39 Codec PWM 输出参数说明.....	246
表 40 Codec 工作模式定义.....	246
表 41 Codec 风噪滤波器参数.....	252
表 42 HR_C7000 基带发送控制寄存器配置.....	272
表 43 HR_C7000 基带接收控制寄存器配置.....	274
表 44 物理层中断类型列表.....	293
表 45 物理层系统中断子中断类型列表.....	293
表 46 物理层发送 Ram 分配.....	294
表 47 物理接收 Ram 分配.....	294
表 48 物理层发送配置表.....	295
表 49 物理层接收配置表.....	296
表 50 链路层中断类型列表.....	310
表 51 链路层系统中断子中断类型列表.....	311
表 52 链路层发送 Ram 分配.....	312
表 53 链路层接收 Ram 分配.....	313
表 54 组呼 72bit LC 信息表.....	324
表 55 链路层发送帧类型列表.....	328
表 56 FM 系统中断子中断类型列表.....	339
表 57 FM 发送 Ram 分配.....	340
表 58 FM 接收 Ram 分配.....	340
表 59 CTCSS 发送地址与标准亚音频频率的映射关系列表.....	341
表 60 CTCSS 接收地址与信号相关检测系数 coef 的映射关系列表.....	342
表 61 非确认数据帧头 PDU 定义.....	358
表 62 确认数据帧头 PDU 定义.....	358
表 63 数据帧体字节数表.....	359
表 64 1/2 速率非确认数据块(R_1_2_DATA) 的 PDU 内容.....	359
表 65 1/2 速率非确认最后数据块(R_1_2_LDATA) 的 PDU 内容.....	359
表 66 3/4 速率带确认数据块(R_3_4_DATA) 的 PDU 内容.....	360
表 67 3/4 速率带确认最后数据块(R_3_4_LDATA) 的 PDU 内容.....	360
表 68 数据帧尾 PDU 定义.....	361
表 69 应答数据帧头 PDU 定义.....	362
表 70 IPv4 Packet Data.....	363
表 71 IPv4 Response Packet Data.....	364
表 72 静态基本加密配置表.....	373
表 73 动态增强加密配置表.....	374
表 74 语音帧头定义.....	380
表 75 语音 PI 帧定义.....	380
表 76 Privacy Data Header.....	381
表 77 同步帧字段定义.....	383
表 78 世代 (Gen) 定义.....	397
表 79 同步时期 (SA) 定义.....	399
表 80 WATID 定义.....	399
表 81 Identifier (ID) 定义.....	400

表 82 DI 定义 .....	400
表 83 CTO 定义 .....	400
表 84 New Leader 定义 .....	401
表 85 CT_CSBK PDU 定义 .....	401
表 86 定时器常量定义 .....	402
表 87 CT_RHOT 值分布表 .....	402

## 寄存器表目录

寄存器表 1 复位寄存器概述（基地址：0x11000000） .....	55
寄存器表 2 时钟管理模块寄存器概述（基地址：0x11000000） .....	60
寄存器表 3 系统控制寄存器概述（基地址：0x11000000） .....	67
寄存器表 4 基带系统控制寄存器概述（基地址：0x11000000） .....	94
寄存器表 5 PIC 寄存器概述（基地址：0x1700_0000） .....	107
寄存器表 6 Timer 寄存器概述（基地址：0x1400_0000） .....	116
寄存器表 7 RTC 寄存器概述（基地址：0x1408_0000） .....	124
寄存器表 8 WDT 寄存器概述（基地址：0x1401_0000） .....	131
寄存器表 9 I2C0 寄存器概述（基地址为 0x14060000） .....	135
寄存器表 10 I2C1 寄存器概述（基地址为 0x14070000） .....	135
寄存器表 11 SPI0 寄存器概述（基地址：0x140A_0000） .....	150
寄存器表 12 SPI1 寄存器概述（基地址：0x140B_0000） .....	151
寄存器表 13 SPI2 寄存器概述（基地址：0x140E_0000） .....	152
寄存器表 14 I80 寄存器概述（基地址：0x1200_0000） .....	157
寄存器表 15 UART0 寄存器概述（基地址：0x1403_0000） .....	162
寄存器表 16 UART1 寄存器概述（基地址：0x1404_0000） .....	163
寄存器表 17 UART2 寄存器概述（基地址：0x1405_0000） .....	163
寄存器表 18 SDIO 寄存器概述（基地址：0x1500_0000） .....	179
寄存器表 19 SDIO 寄存器概述（基地址：0x1500_0000） .....	193
寄存器表 20 GPIOA 寄存器概述（基地址：0x1402_0000） .....	219
寄存器表 21 GPIOB 寄存器概述（基地址：0x1410_0000） .....	219
寄存器表 22 GPIOC 寄存器概述（基地址：0x1411_0000） .....	220
寄存器表 23 PWM 寄存器概述（基地址：0x140C_0000） .....	224
寄存器表 24 ADC 寄存器概述（基地址：0x140D_0000） .....	235
寄存器表 25 DAC 寄存器概述（基地址：0x140D_0000） .....	242
寄存器表 26 Codec 寄存器概述（基地址：0x1600_09C0） .....	253
寄存器表 27 物理层寄存器概述（基地址：0x1100_0000） .....	278
寄存器表 28 链路层寄存器概述（基地址：0x1100_0000） .....	297
寄存器表 29 FM 寄存器概述（基地址：0x1100_0000） .....	331

# 1 前言

## 1.1 概述

本文介绍 HR\_C7000 系列芯片的特性、逻辑结构，并详述各个功能块、工作方式、寄存器定义等，并描述了芯片的管脚定义和用途以及芯片的性能参数和封装尺寸。

## 1.2 表格内容约定

内容	说明
-	表示符号所在表格单元格无内容

## 1.3 寄存器访问类型约定

类型	说明
RO	只读，不可写
WO	只写
RW	可读可写
RC	读清零
WC	可读，写 1 清零，写 0 保持不变。

## 1.4 数值单位约定

类别	符号	对应数值
数据容量（如内存容量）	1K	1024
	1M	1,048,576
	1G	1,073,741,824
频率、数据速率等	1k	1,000
	1M	1,000,000
	1G	1,000,000,000

## 2 产品概述

HR\_C7000 芯片是一款应用于专业数字对讲机系统处理芯片，主要应用在手持数字对讲机终端，数字中转台，简化产品设计。内置高性能的数字基带调制解调，MCU 处理器，声码器和 AD/DA 接口以及语音 Codec 的模块，进一步提高单芯片的集成度。同时芯片集成 DMR 数字对讲的标准协议，以及在内置 MCU 中集成强大和丰富的特色功能和复杂协议补充，极大的降低了数字对讲系统的开发门槛。

### 2.1 应用场景

HR\_C7000 芯片是面向数字对讲机系统产品的一款 SOC 芯片。

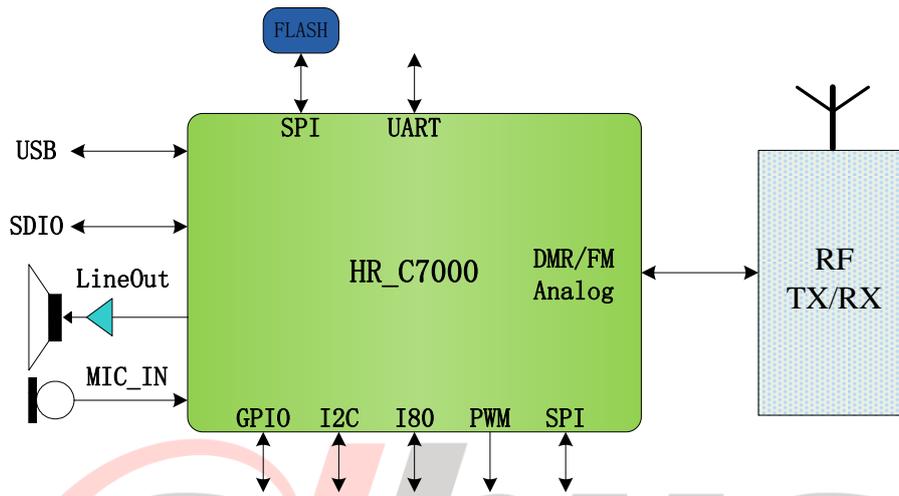


图 1 HR\_C7000 单芯片应用场景

## 2.2 架构

### 2.2.1 概述

HR\_C7000 是一颗数字对讲系统处理 SOC 芯片，内部除集成 CPU 外，还包括 SPI, UART, I2C, GPIO, PWM, ADC, DAC, SDIO, USB 等丰富的外设。集成高性能的 4FSK 调制解调器、信道编译码、协议处理器，同时还包括 RTC, POR, DAC, ADC 和 Codec 等模拟模块。

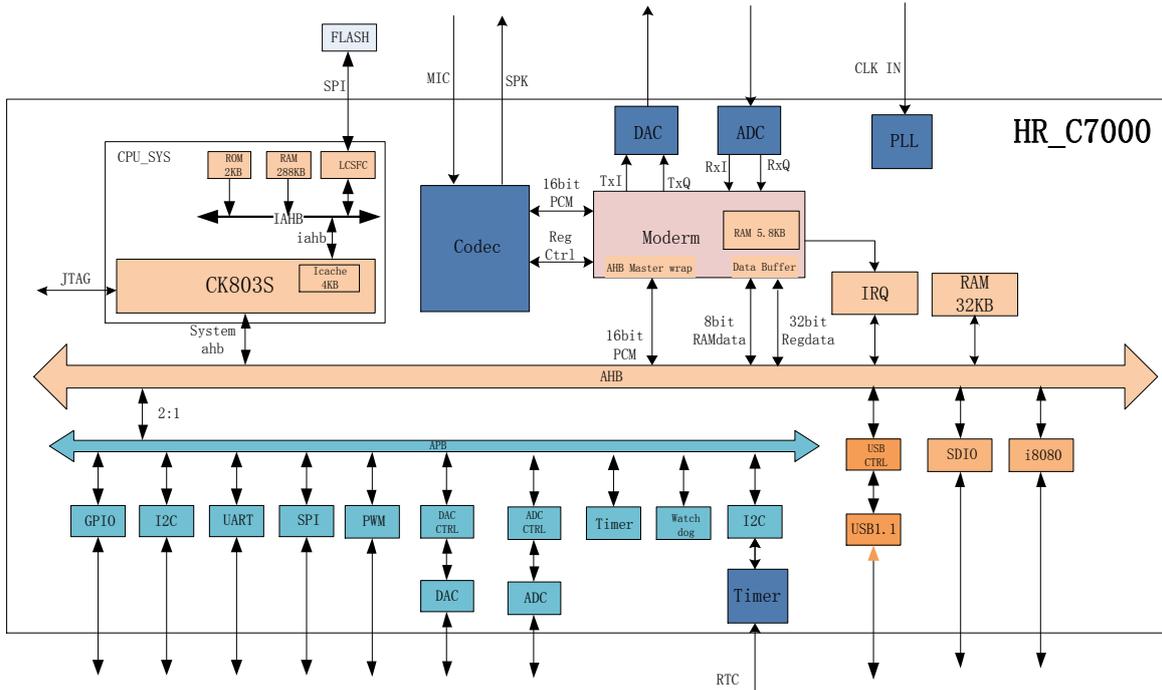


图 2 芯片总体框图

### 2.2.2 处理器内核

CPU 采用中天 32 位 CK803S 处理器。

### 2.2.3 DMR 标准

- 符合 ETSI TS102 361 (DMR) Tier I/II/III 标准的协议设计
- 支持物理层、数据链路层和呼叫控制层独立控制
- 支持真双时隙同步头检测
- 采用 TDMA 技术，支持全双工、半双工语音、数据通信及数话同传业务
- 支持 IP 数据业务
- 支持单频、双频中继
- 支持 4.8Kbps 和 9.6Kbps 数据传输
- 支持数字模拟智能检测
- 支持中继语音和数据功能
- 支持语音加密功能
- 支持真双时隙功能

### 2.2.4 模拟 FM

- 支持 12.5KHz/25KHz 信道通信
- 支持加重、去加重
- 支持 CDCSS/CTCSS 亚音处理
- 支持 2-tone/5-tone 处理
- 支持 DTMF 处理
- 支持模拟静噪功能

### 2.2.5 调制解调及信道编解码

- 高性能 4FSK 调制解调
- 集成协议规定的信道编解码器

### 2.2.6 模拟 IP

- 高性能的 10bit 位宽 ADC
- 高性能的 12bit 位宽 DAC
- LDO, 采用 3.3V 供电, 单一电源设计
- 高性能 PLL
- 高性能 Codec, 支持差分或单端 Mic 输入和 Line\_out 输出

### 2.2.7 声码器支持

- 支持 HR\_V3000(宏睿 AMBE+2)、SELP 声码器(清华)、AVDS 声码器(712)等 SPI 接口的声码器, 同时为数字录音、回放及提示音输入提供接口
- 支持数字语音加密

### 2.2.8 射频接口

- 发送射频接口采用单端输出, 模拟两点调制和数字 AK2401 接口
- 接收射频接口采用差分输入方式, 支持基带 IQ、中频和 AF, 支持 AK2401 数字接口
- 发送两路信号偏置、幅度大小可独立调节
- 支持用户配置 GPIO 控制射频通道

### 2.2.9 外围接口

- 支持 POR 和外部复位
- 集成 4 个 UART 接口, 其中 3 个可复用为 GPIO
- 集成 3 个 I2C 接口, 其中 2 个可复用为 GPIO, 1 个供 RTC 使用
- 集成 1 个 I8080 接口, 可复用为 GPIO
- 集成 4 个 SPI 接口, 支持主从模式; 其中 NFC SPI 固定用于 nor Flash 接口, 支持 QualSPI Flash, 另 3 个可复用为 GPIO
- 集成 3 个 PWM 接口, 可复用为 GPIO
- 集成 JTAG 接口, 可复用为 GPIO
- 集成 8 通道 ADC 接口, 10bit 位宽, 可复用为 GPIO
- 集成 3 通道 DAC 接口, 12bit 位宽, 可复用为 GPIO
- 集成 RTC, 支持 32.768K 晶体
- 集成 USB 接口, 支持 USB1.1 标准
- 集成 SDIO 接口, 支持单线和 4 线模式

### 2.2.10 外部存储器接口

- 支持 SPI Flash

## 2.3 启动模式

- 支持串口启动模式
- 支持 SPI Flash 启动

### 3 硬件特性

#### 3.1 HR\_C7000 封装

HR\_C7000 采用 STFBGA 封装，封装尺寸 11mm x 11mm x 1.27mm，管脚间距为 0.65mm，管脚总数为 220 个。

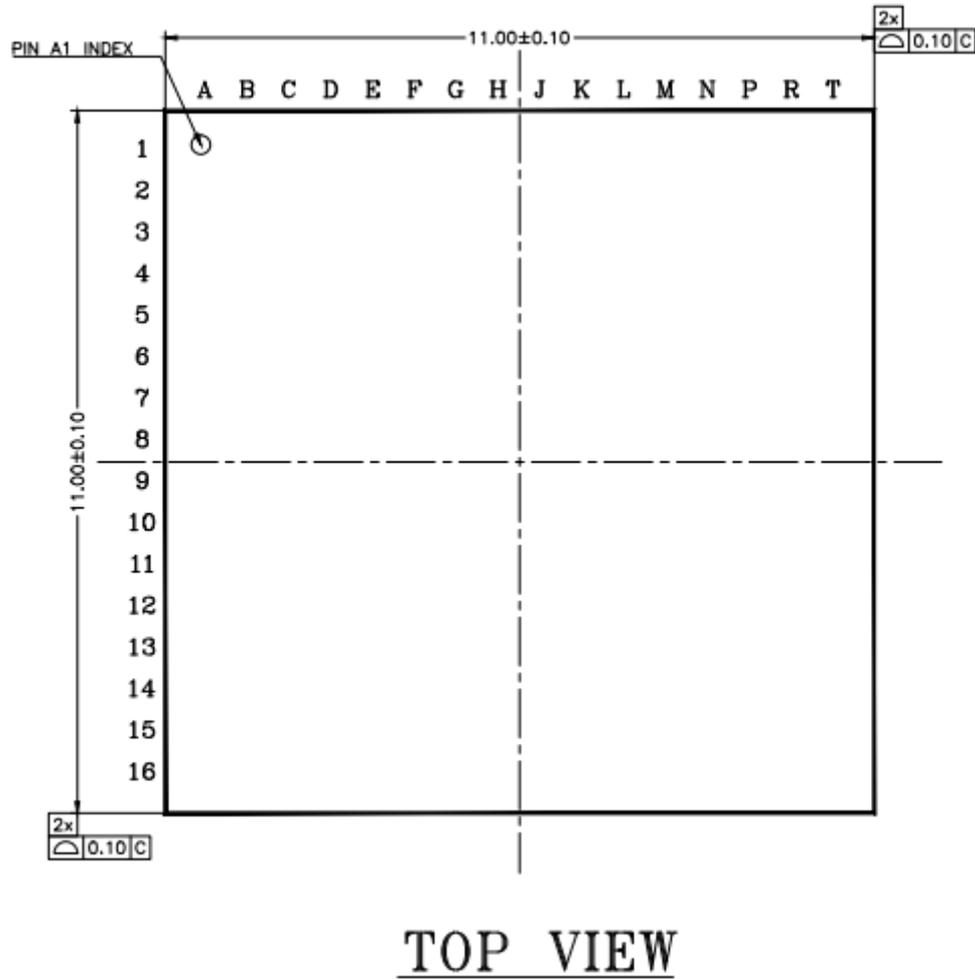


图 3 HR\_C7000 芯片封装顶视图

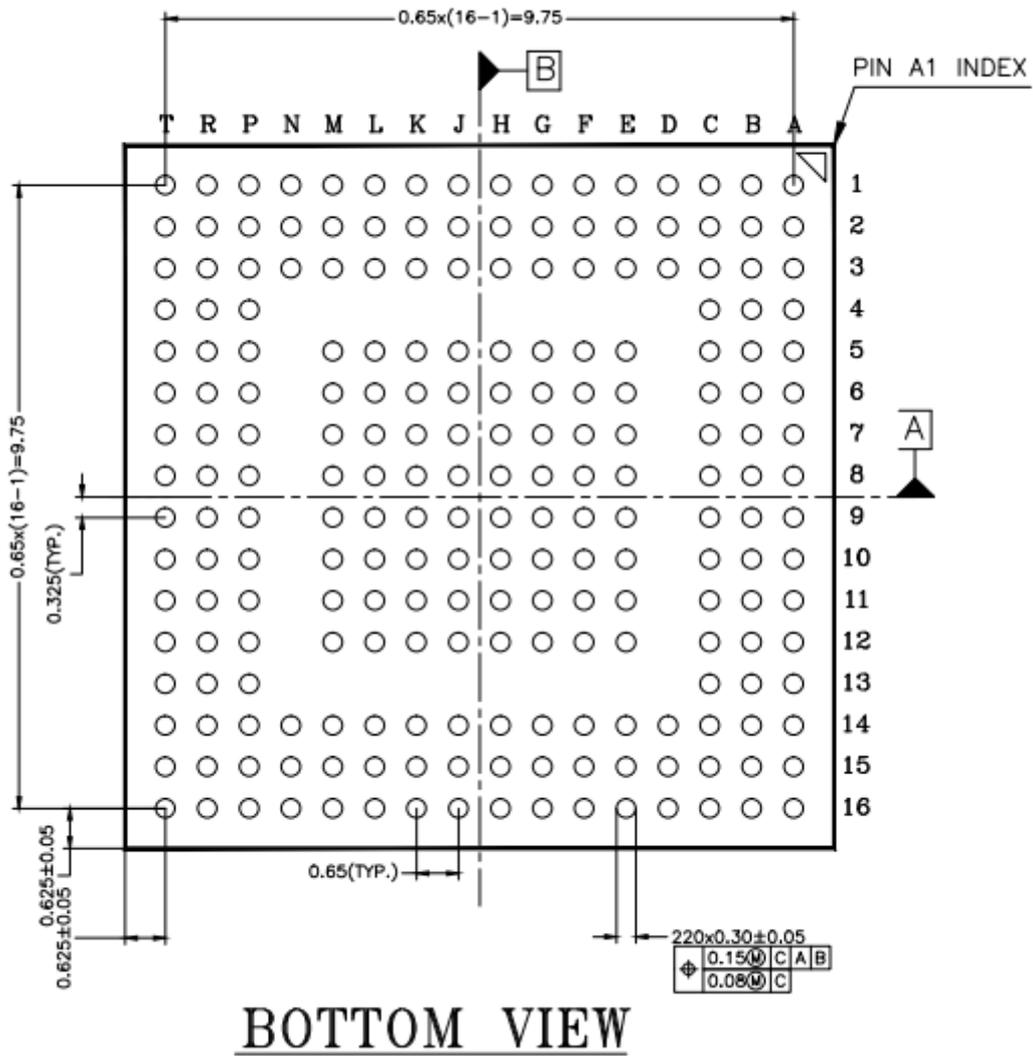


图 4 HR\_C7000 芯片封装底视图

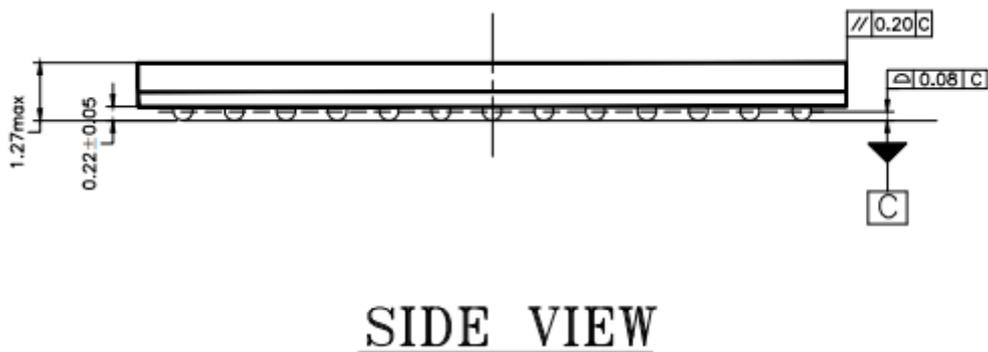


图 5 HR\_C7000 芯片封装侧视图

表 1 HR\_C7000 芯片封装参数说明表

Symbol	Dimension in mm		
	MIN	NOM	MAX

A			1.27
A1	0.16	0.21	0.26
A2	0.91	0.96	1.01
c	0.22	0.26	0.30
D	10.90	11.0	11.10
E	10.90	11.0	11.10
D1		10.10	
E1		10.10	
e		0.65	
b	0.25	0.30	0.35
aaa	0.15		
bbb	0.10		
ddd	0.08		
eee	0.15		
fff	0.08		
MD/ME	16/16		

### 3.2 HR\_C7000 管脚分布

HR\_C7000 共有 220 个管脚，管脚分布如下图所示：



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
A	DVSS	ADC1_IN	ADC3_IN	ADC6_IN	ADC7_IN	JTG_TDI	SPI0_MOSI	SPI0_CSN_0	I2C_SCL_1	UART1_TXD	PTB27	LCD_DB6	Codec_MiC_INP2	Codec_MiC_INP1	Codec_MiCBIAS	DVSS	A
B	ADC0_IN	ADC_VRE	ADC2_IN	ADC5_IN	JTG_TCK	JTG_TDO	SPI0_MISO	SPI0_CSN_1	I2C_SDA_1	PTB29	PTB26	LCD_DB7	Codec_MiC_INN2	Codec_MiC_INN1	Codec_VCAP	Codec_VREFP	B
C	DVSS	ADC_AVSS	ADC_AV1	ADC4_IN	JTG_TMS	JTG_RST_N	DVSS	SPI0_SCLK	UART1_RXD	PTB28	PTB25	PTB24	Codec_AVDD	Codec_LINOUT2_LP	Codec_LINOUT1_LP	DVDD25	C
D	RTC_CLK	RTC_CLK	LDO_RTC_AVDD33										Codec_AVSS33	DVSS	Codec_LINOUT_VSS		D
E	DVSS	DVSS	LDO_RTC_AVSS33	RTC_AVSS	DVSS	DVSS	LDO_DVDD1_2_1	LDO_AV1	LDO_AVSS	DVSS	DVDD33		DVSS	LCD_DB3	LCD_DB4		E
F	USB_DP	USB_DM	RTC_DVDD12		DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVDD33		LCD_DB3	LCD_DB2	LCD_DB1	F
G	DVSS	DVSS	USB_AVDD3_3		DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS		LCD_DB0	LCD_RD	LCD_WR	G
H	PR_G	PR_V	DVSS		DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	LDO_AVDD33_0		LCD_RS	LCD_CS	LCD_NRESET	H
J	DAC_VOUT	DAC_VOUT	DAC_AVDD33		DAC_VREF	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	LDO_AVSS	PWM_2	DVSS	UART2_TXD	J
K	DAC_VOUT	DAC_VOUT	DAC_AVSS33		DAC_VREF	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	LDO_DVDD1_2_0	UART2_RXD	PTB13	PTB14	K
L	DAC_VOUT	DVSS	ADC_AVDD33		DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS	DVSS		PTB15	PTB16	UART3_RXD	L
M	ADC_VIN	ADC_VIN	ADC_VREF		LDO_DVDD1_2_3	PLL_AVDD	PLL_AVSS	DVSS	DVSS	DVSS	DVDD33	DVDD33		UART3_TXD	PTB17	PWM_1	M
N	ADC_VIN	ADC_VIN	ADC_AVSS33											NFC_CSN	NFC_SCLK	NFC_MOSI	N
P	ADC_VIN	ADC_VIN	LDO_AVDD33_3_3	LDO_AVSS	PTB20	PTB23	PTB3	PTB6	I2C_SDA_0	UART0_TXD	SPI1_MISO	BS_INTE_R_IN	PTB8	PTB11	NFC_MISO	NFC_HOLD	P
R	ADC_AV1	POR_EN	NRESET	PTB19	PTB21	CLK_OUT	PTB2	PTB5	I2C_SCL_0	SPI1_CSN_0	SPI1_MOSI	BS_INTE_R_OUT	PTB7	PTB10	PTB12	NFC_WP	R
T	DVSS	OSC_CLK_IN	TEST_MODE	PTB18	PTB22	PTB0	PTB1	PTB4	UART0_RXD	SPI1_CSN_1	SPI1_SCLK	TIME_SLOT_R_IN	TIME_SLOT_T_IN	PTB9	PWM_0	DVSS	T
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

图 6 HR\_C7000 管脚分布图

### 3.3 HR\_C7000 管脚排列表

表 2 HR\_C7000 管脚排列表

位置	管脚名称	位置	管脚名称
A1	DVSS	J1	DAC_VOUT_A
A2	ADC1_IN	J2	DAC_VOUT_B
A3	ADC3_IN	J3	DAC_AVDD33
A4	ADC6_IN	J5	DAC_VREFP
A5	ADC7_IN	J6	DVSS
A6	JTG_TDI	J7	DVSS
A7	SPI0_MOSI	J8	DVSS
A8	SPI0_CSN_0	J9	DVSS
A9	I2C_SCL_1	J10	DVSS
A10	UART1_TXD	J11	DVSS

A11	PTB27	J12	LDO_AVSS
A12	LCD_DB6	J14	PWM_2
A13	Codec_MiC_INP2	J15	DVSS
A14	Codec_MiC_INP1	J16	UART2_TXD
A15	Codec_MICBIAS	K1	DAC_VOUT_MCUA
A16	DVSS	K2	DAC_VOUT_MCUB
B1	ADC0_IN	K3	DAC_AVSS33
B2	ADC_VREF_MCU	K5	DAC_VREFN
B3	ADC2_IN	K6	DVSS
B4	ADC5_IN	K7	DVSS
B5	JTG_TCK	K8	DVSS
B6	JTG_TDO	K9	DVSS
B7	SPI0_MISO	K10	DVSS
B8	SPI0_CSN_1	K11	DVSS
B9	I2C_SDA_1	K12	LDO_DVDD12_0
B10	PTB29	K14	UART2_RXD
B11	PTB26	K15	PTB13
B12	LCD_DB7	K16	PTB14
B13	Codec_MIC_INN2	L1	DAC_VOUT_MCUC
B14	Codec_MIC_INN1	L2	DVSS
B15	Codec_VCAP	L3	ADC_AVDD33
B16	Codec_VREFP	L5	DVSS
C1	DVSS	L6	DVSS
C2	ADC_AVSS33_MCU	L7	DVSS
C3	ADC_AVDD33_MCU	L8	DVSS
C4	ADC4_IN	L9	DVSS
C5	JTG_TMS	L10	DVSS
C6	JTG_RST_N	L11	DVSS
C7	DVSS	L12	DVSS
C8	SPI0_SCLK	L14	PTB15
C9	UART1_RXD	L15	PTB16
C10	PTB28	L16	UART3_RXD
C11	PTB25	M1	ADC_VINP_Q
C12	PTB24	M2	ADC_VINM_Q
C13	Codec_AVDD	M3	ADC_VREF
C14	Codec_LINOUT2_LP	M5	LDO_DVDD12_3
C15	Codec_LINOUT1_LP	M6	PLL_AVDD
C16	DVDD25	M7	PLL_AVSS
D1	RTC_CLK_IN	M8	DVSS
D2	RTC_CLK_OUT	M9	DVSS
D3	LDO_RTC_AVDD33	M10	DVSS
D14	Codec_AVSS33	M11	DVDD33
D15	DVSS	M12	DVDD33

D16	Codec_LINOUT_VSS	M14	UART3_TXD
E1	DVSS	M15	PTB17
E2	DVSS	M16	PWM_1
E3	LDO_RTC_AVSS33	N1	ADC_VINCM_Q
E5	RTC_AVSS	N2	ADC_VINP_I
E6	DVSS	N3	ADC_AVSS33
E7	DVSS	N14	NFC_CSN
E8	LDO_DVDD12_1	N15	NFC_SCLK
E9	LDO_AVDD33_1	N16	NFC_MOSI
E10	LDO_AVSS	P1	ADC_VINCM_I
E11	DVSS	P2	ADC_VINM_I
E12	DVDD33	P3	LDO_AVDD33_3
E14	DVSS	P4	LDO_AVSS
E15	LCD_DB5	P5	PTB20
E16	LCD_DB4	P6	PTB23
F1	USB_DP	P7	PTB3
F2	USB_DM	P8	PTB6
F3	RTC_DVDD12	P9	I2C_SDA_0
F5	DVSS	P10	UART0_TXD
F6	DVSS	P11	SPI1_MISO
F7	DVSS	P12	BS_INTER_IN
F8	DVSS	P13	PTB8
F9	DVSS	P14	PTB11
F10	DVSS	P15	NFC_MISO
F11	DVSS	P16	NFC_HOLD
F12	DVDD33	R1	ADC_AVDD33
F14	LCD_DB3	R2	POR_EN
F15	LCD_DB2	R3	NRESET
F16	LCD_DB1	R4	PTB19
G1	DVSS	R5	PTB21
G2	DVSS	R6	CLK_OUT
G3	USB_AVDD33	R7	PTB2
G5	DVSS	R8	PTB5
G6	DVSS	R9	I2C_SCL_0
G7	DVSS	R10	SPI1_CSN_0
G8	DVSS	R11	SPI1_MOSI
G9	DVSS	R12	BS_INTER_OUT
G10	DVSS	R13	PTB7
G11	DVSS	R14	PTB10
G12	DVSS	R15	PTB12
G14	LCD_DB0	R16	NFC_WP
G15	LCD_RD	T1	DVSS
G16	LCD_WR	T2	OSC_CLK_IN

H1	PR_G	T3	TEST_MODE
H2	PR_V	T4	PTB18
H3	DVSS	T5	PTB22
H5	DVSS	T6	PTB0
H6	DVSS	T7	PTB1
H7	DVSS	T8	PTB4
H8	DVSS	T9	UART0_RXD
H9	DVSS	T10	SPI1_CSN_1
H10	DVSS	T11	SPI1_SCLK
H11	DVSS	T12	TIME_SLOT_R_INTER
H12	LDO_AVDD33_0	T13	TIME_SLOT_T_INTER
H14	LCD_RS	T14	PTB9
H15	LCD_CS	T15	PWM_0
H16	LCD_NRESET	T16	DVSS

### 3.4 HR\_C7000 管脚描述

管脚 IO 类型说明如下表所示。

表 3 I/O 类型说明

I/O 类型	描述
I	输入信号
O	输出信号
I/O	双向输入/输出信号
P	电源
G	地

管脚 IO 上下拉说明如下表所示。

表 4 上下拉约定

上下拉	描述
U	上拉
D	下拉
-	无上下拉

IO 复用控制和 IO 上下拉控制寄存器参考章节 4.4 系统控制 IO 相关控制寄存器。

#### 3.4.1 系统管脚

表 5 系统管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下拉	电压 (V)	描述
T2	OSC_CLK_IN	I	-	-	3.3	时钟输入
R6	CLK_OUT	I/O	4	-	3.3	功能 0: GPIO PTC0 功能 1: PLL 时钟分频输出; 功能 2: RTC 测试时钟输出
R3	NRESET	I	-	U	3.3	外部复位输入信号
R2	POR_EN	I	-	U	3.3	POR 使能,

						1: 开启 POR 功能; 0: 关闭 POR 功能
T3	TEST_MODE	I	-	D	3.3	测试模式选择: 1: 进入 Test 模式; 0: 芯片正常工作模式
D1	RTC_CLK_IN	I	-	-	3.3	RTC 时钟输入
D2	RTC_CLK_OUT	O	-	-	3.3	RTC 时钟反馈输出
F1	USB_DP	I/O	-	-	3.3	USB 差分数据
F2	USB_DM	I/O	-	-	3.3	USB 差分数据

### 3.4.2 NFC 管脚

表 6 NFC 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下拉	电压 (V)	描述
N15	NFC_SCLK	O	16	U	3.3	NFC SPI 接口时钟, 最高支持 96MHz。
N14	NFC_CSN	O	8	U	3.3	片外 Nor Flash SPI 接口片选信号, 默认主模式
N16	NFC_MOSI	I/O	8	U	3.3	串行 SPI 的数据输出; QualSPI 的数据 I00
P15	NFC_MISO	I/O	8	U	3.3	串行 SPI 的数据输入; QualSPI 的数据 I01
R16	NFC_WP	I/O	8	U	3.3	串行 SPI 的 WP 控制信号; QualSPI 的数据 I02
P16	NFC_HOLD	I/O	8	U	3.3	串行 SPI 的 Hold 控制信号; QualSPI 的数据 I03

### 3.4.3 JTAG 管脚

表 7 JTAG 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下拉	电压 (V)	描述
C5	JTG_TMS	I/O	4	U	3.3	功能 0: JTAG 模式选择输入 功能 1: GPIO PTA0, 通用输入输出
B5	JTG_TCK	I/O	8	U	3.3	功能 0: JTAG 时钟输入 功能 1: GPIO PTA1, 通用输入输出
A6	JTG_TDI	I/O	4	D	3.3	功能 0: JTAG 数据输入 功能 1: GPIO PTA2, 通用输入输出
B6	JTG_TDO	I/O	4	D	3.3	功能 0: JTAG 数据输出 功能 1: GPIO PTA3, 通用输入输出
C6	JTG_RST_N	I/O	4	U	3.3	功能 0: JTAG 复位输入 功能 1: GPIO PTA4, 通用输入输出

### 3.4.4 I2C 管脚

表 8 I2C 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下拉	电压 (V)	描述
R9	I2C_SCL_0	I/O	4	U	3.3	功能 0: I2C_SCL_0, I2C0 时钟 功能 1: GPIO PTA5, 通用输入输出
P9	I2C_SDA_0	I/O	4	U	3.3	功能 0: I2C_SDA_0, I2C0 数据/地址 功能 1: GPIO PTA6, 通用输入输出
A9	I2C_SCL_1	I/O	4	U	3.3	功能 0: I2C_SCL_1, I2C1 时钟 功能 1: GPIO PTA7, 通用输入输出
B9	I2C_SDA_1	I/O	4	U	3.3	功能 0: I2C_SDA_1, I2C1 数据/地址 功能 1: GPIO PTA8, 通用输入输出

### 3.4.5 UART 管脚

表 9 UART 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下拉	电压 (V)	描述
T9	UART0_RXD	I	4	U	3.3	串口 0 数据输入, 可以用于调试和 boot 升级
P10	UART0_TXD	O	4	U	3.3	串口 0 数据输出, 可以用于调试和 boot 升级
C9	UART1_RXD	I/O	4	U	3.3	功能 0: UART1_RXD, 串口 1 数据输入 功能 1: GPIO PTA9, 通用输入输出
A10	UART1_TXD	I/O	4	U	3.3	功能 0: UART1_TXD, 串口 1 数据输出 功能 1: GPIO PTA10, 通用输入输出
K14	UART2_RXD	I/O	4	U	3.3	功能 0: UART1_RXD, 串口 2 数据输入 功能 1: GPIO PTA11, 通用输入输出
J16	UART2_TXD	I/O	4	U	3.3	功能 0: UART1_TXD, 串口 2 数据输出 功能 1: GPIO PTA12, 通用输入输出
L16	UART3_RXD	I/O	4	U	3.3	功能 0: UART1_RXD, 串口 3 数据输入 功能 1: GPIO PTA13, 通用输入输出 功能 2: 基带 RF 发送开始中断输出
M14	UART3_TXD	I/O	4	U	3.3	功能 0: UART1_TXD, 串口 3 数据输出 功能 1: GPIO PTA14, 通用输入输出 功能 2: 基带 RF 发送结束中断输出

### 3.4.6 PWM 管脚

表 10 PWM 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下拉	电压 (V)	描述
T15	PWM_0	I/O	4	U	3.3	功能 0: PWM_0, 脉宽调制 0 输出 功能 1: GPIO PTA15, 通用输入输出
M16	PWM_1	I/O	4	U	3.3	功能 0: PWM_1, 脉宽调制 1 输出 功能 1: GPIO PTA16, 通用输入输出
J14	PWM_2	I/O	4	D	3.3	功能 0: PWM_2, 脉宽调制 2 输出 功能 1: GPIO PTA17, 通用输入输出

### 3.4.7 SPI 管脚

表 11 SPI 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下拉	电压 (V)	描述
A8	SPI0_CSN_0	I/O	4	U	3.3	功能 0: SPI0 片选 0 输出, SPI0 主模式 功能 1: SPI0 片选输入, SPI0 从模式 功能 2: GPIO PTA18, 通用输入输出
B8	SPI0_CSN_1	I/O	4	U	3.3	功能 0: SPI0 片选 1 输出, SPI0 主模式 功能 1: GPIO PTA19, 通用输入输出
C8	SPI0_SCLK	I/O	8	D	3.3	功能 0: SPI0 时钟输出, SPI0 主模式 功能 1: SPI0 时钟输入, SPI0 从模式 功能 2: GPIO PTA20, 通用输入输出
A7	SPI0_MOSI	I/O	4	U	3.3	功能 0: SPI0 数据输出, SPI0 主模式 功能 1: SPI0 数据输入, SPI0 从模式 功能 2: GPIO PTA21, 通用输入输出
B7	SPI0_MISO	I/O	4	D	3.3	功能 0: SPI0 数据输入, SPI0 主模式 功能 1: SPI0 数据输出, SPI0 从模式 功能 2: GPIO PTA22, 通用输入输出
R10	SPI1_CSN_0	I/O	4	U	3.3	功能 0: SPI1 片选 0 输出, SPI1 主模式 功能 1: SPI1 片选输入, SPI1 从模式 功能 2: GPIO PTA23, 通用输入输出 功能 3: SDIO 数据, SDIO_DAT3

T10	SPI1_CSN_1	I/O	4	U	3.3	功能0: SPI1 片选1 输出, SPI1 主模式 功能1: GPIO PTA24, 通用输入输出 功能2: 基带系统中断输出
T11	SPI1_SCLK	I/O	8	D	3.3	功能0: SPI1 时钟输出, SPI1 主模式 功能1: SPI1 时钟输入, SPI1 从模式 功能2: GPIO PTA25, 通用输入输出 功能3: SDIO 时钟输出, SDIO_CLK
R11	SPI1_MOSI	I/O	4	U	3.3	功能0: SPI1 数据输出, SPI1 主模式 功能1: SPI1 数据输入, SPI1 从模式 功能2: GPIO PTA26, 通用输入输出 功能3: SDIO 命令, SDIO_CMD
P11	SPI1_MISO	I/O	4	U	3.3	功能0: SPI1 数据输入, SPI1 主模式 功能1: SPI1 数据输出, SPI1 从模式 功能2: GPIO PTA27, 通用输入输出 功能3: SDIO 数据, SDIO_DAT0
H15	LCD_CS	I/O	4	U	3.3	功能0: SPI2 片选输出, SPI2 主模式 功能1: SPI2 片选输入, SPI2 从模式 功能2: GPIO PTC3, 通用输入输出 功能3: I8080 CS , LCD_CS
H14	LCD_RS	I/O	8	U	3.3	功能0: SPI2 时钟输出, SPI2 主模式 功能1: SPI2 时钟输入, SPI2 从模式 功能2: GPIO PTC4, 通用输入输出 功能3: I8080 RS , LCD_RS
G14	LCD_DB0	I/O	4	D	3.3	功能0: SPI2 数据输入, SPI2 主模式 功能1: SPI2 数据输出, SPI2 从模式 功能2: GPIO PTC7, 通用输入输出 功能3: I8080 数据, LCD_DB0
F16	LCD_DB1	I/O	4	D	3.3	功能0: SPI2 数据输出, SPI2 主模式 功能1: SPI2 数据输入, SPI2 从模式 功能2: GPIO PTAC8, 通用输入输出 功能3: I8080 数据, LCD_DB1

### 3.4.8 SDIO 管脚

表 12 SDIO 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下拉	电压 (V)	描述
T11	SPI1_SCLK	I/O	8	D	3.3	功能0: SDIO 时钟输出, SDIO_CLK 功能1: SPI1 时钟输出, SPI1 主模式 功能2: SPI1 时钟输入, SPI1 从模式 功能3: GPIO PTA25, 通用输入输出
R11	SPI1_MOSI	I/O	4	U	3.3	功能0: SDIO 命令, SDIO_CMD 功能1: SPI1 数据输出, SPI1 主模式 功能2: SPI1 数据输入, SPI1 从模式 功能3: GPIO PTA26, 通用输入输出
P11	SPI1_MISO	I/O	4	U	3.3	功能0: SDIO 数据, SDIO_DAT0 功能1: SPI1 数据输入, SPI1 主模式 功能2: SPI1 数据输出, SPI1 从模式 功能3: GPIO PTA27, 通用输入输出
R12	BS_INTER_OUT	I/O	4	U	3.3	功能0: SDIO 数据, SDIO_DAT1 功能1: 基带 BS 模式中中断输出 功能2: GPIO PTA29 通用输入输出
P12	BS_INTER_IN	I/O	4	U	3.3	功能0: SDIO 数据, SDIO_DAT2 功能1: 基带 BS 模式中中断输入 功能2: GPIO PTA28, 通用输入输出
R10	SPI1_CSN_0	I/O	4	U	3.3	功能0: SDIO 数据, SDIO_DAT3 功能1: SPI1 片选0 输出, SPI1 主模式 功能2: SPI1 片选输入, SPI1 从模式 功能3: GPIO PTA23, 通用输入输出
T12	TIME_SLOT_R_IN TER	I/O	4	D	3.3	功能0: SDIO 卡检测, SDIO_CARD_DETECT 功能1: 基带 30ms 接收时隙中断输出 功能2: GPIO PTA30, 通用输入输出

T13	TIME_SLOT_T_IN TER	I/O	4	D	3.3	功能 0: SDIO 写保护, SDIO_CARD_CWPR 功能 1: 基带 30ms 发送时隙中断输出 功能 2: GPIO PTA31, 通用输入输出 功能 3: POR 复位输出
-----	-----------------------	-----	---	---	-----	--

### 3.4.9 I8080 (LCD) 管脚

表 13 I8080 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上 下 拉	电压 (V)	描述
H16	LCD_NRESET	I/O	4	U	3.3	GPIO PTC2, 通用输入输出
H15	LCD_CS	I/O	4	U	3.3	功能 0: I8080 CS, LCD_CS 功能 1: SPI2 片选输出, SPI2 主模式 功能 2: SPI2 片选输入, SPI2 从模式 功能 3: GPIO PTC3, 通用输入输出
H14	LCD_RS	I/O	8	U	3.3	功能 0: I8080 RS, LCD_RS 功能 1: SPI2 时钟输出, SPI2 主模式 功能 2: SPI2 时钟输入, SPI2 从模式 功能 3: GPIO PTC4, 通用输入输出
G16	LCD_WR	I/O	4	U	3.3	功能 0: I8080 WR, LCD_WR 功能 1: GPIO PTC5, 通用输入输出
G15	LCD_RD	I/O	4	U	3.3	功能 0: I8080 RD, LCD_RD 功能 1: GPIO PTC6, 通用输入输出
G14	LCD_DB0	I/O	4	D	3.3	功能 0: I8080 数据, LCD_DB0 功能 1: SPI2 数据输入, SPI2 主模式 功能 2: SPI2 数据输出, SPI2 从模式 功能 3: GPIO PTC7, 通用输入输出
F16	LCD_DB1	I/O	4	D	3.3	功能 0: I8080 数据, LCD_DB1 功能 1: SPI2 数据输出, SPI2 主模式 功能 2: SPI2 数据输入, SPI2 从模式 功能 3: GPIO PTAC8, 通用输入输出
F15	LCD_DB2	I/O	4	D	3.3	功能 0: I8080 数据, LCD_DB2 功能 1: GPIO PTAC9, 通用输入输出
F14	LCD_DB3	I/O	4	U	3.3	功能 0: I8080 数据, LCD_DB3 功能 1: GPIO PTAC10, 通用输入输出
E16	LCD_DB4	I/O	4	U	3.3	功能 0: I8080 数据, LCD_DB4 功能 1: GPIO PTAC11, 通用输入输出
E15	LCD_DB5	I/O	4	U	3.3	功能 0: I8080 数据, LCD_DB5 功能 1: GPIO PTAC12, 通用输入输出
A12	LCD_DB6	I/O	4	U	3.3	功能 0: I8080 数据, LCD_DB6 功能 1: GPIO PTAC13, 通用输入输出
B12	LCD_DB7	I/O	4	D	3.3	功能 0: I8080 数据, LCD_DB7 功能 1: GPIO PTAC14, 通用输入输出

### 3.4.10 GPIO 管脚

表 14 GPIO 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上 下 拉	电压 (V)	描述
T6	PTB0	I/O	4	U	3.3	GPIO PTB0, 通用输入输出
T7	PTB1	I/O	4	U	3.3	GPIO PTB1, 通用输入输出
R7	PTB2	I/O	4	U	3.3	功能 0: GPIO PTB2, 通用输入输出 功能 1: AK2401, AK_DAC_FS

P7	PTB3	I/O	4	U	3.3	功能 0: GPIO PTB3, 通用输入输出 功能 1: AK2401, AK_DAC_SCLK
T8	PTB4	I/O	4	D	3.3	功能 0: GPIO PTB4, 通用输入输出 功能 1: AK2401, AK_DAC_SDI
R8	PTB5	I/O	4	D	3.3	功能 0: GPIO PTB5, 通用输入输出 功能 1: AK2401, AK_ADC_FS
P8	PTB6	I/O	4	D	3.3	功能 0: GPIO PTB6, 通用输入输出 功能 1: AK2401, AK_ADC_SCLK
R13	PTB7	I/O	4	U	3.3	功能 0: GPIO PTB7, 通用输入输出 功能 1: AK2401, AK_ADC_SDO
P13	PTB8	I/O	4	U	3.3	GPIO PTB8, 通用输入输出
T14	PTB9	I/O	4	U	3.3	GPIO PTB9, 通用输入输出
R14	PTB10	I/O	4	U	3.3	GPIO PTB10, 通用输入输出
P14	PTB11	I/O	4	D	3.3	GPIO PTB11, 通用输入输出
R15	PTB12	I/O	4	D	3.3	GPIO PTB12, 通用输入输出
K15	PTB13	I/O	4	U	3.3	GPIO PTB13, 通用输入输出
K16	PTB14	I/O	4	U	3.3	GPIO PTB14, 通用输入输出
L14	PTB15	I/O	4	U	3.3	GPIO PTB15, 通用输入输出
L15	PTB16	I/O	4	U	3.3	GPIO PTB16, 通用输入输出
M15	PTB17	I/O	4	D	3.3	GPIO PTB17, 通用输入输出
T4	PTB18	I/O	4	D	3.3	GPIO PTB18, 通用输入输出
R4	PTB19	I/O	4	U	3.3	GPIO PTB19, 通用输入输出
P5	PTB20	I/O	4	U	3.3	GPIO PTB20, 通用输入输出
R5	PTB21	I/O	4	U	3.3	GPIO PTB21, 通用输入输出
T5	PTB22	I/O	4	U	3.3	GPIO PTB22, 通用输入输出
P6	PTB23	I/O	4	D	3.3	GPIO PTB23, 通用输入输出
C12	PTB24	I/O	4	D	3.3	GPIO PTB24, 通用输入输出
C11	PTB25	I/O	4	U	3.3	GPIO PTB25, 通用输入输出
B11	PTB26	I/O	4	U	3.3	GPIO PTB26, 通用输入输出
A11	PTB27	I/O	4	U	3.3	GPIO PTB27, 通用输入输出
C10	PTB28	I/O	4	U	3.3	GPIO PTB28, 通用输入输出
B10	PTB29	I/O	4	D	3.3	GPIO PTB29, 通用输入输出

### 3.4.11 AK2401 管脚

表 15 AK2401 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下 拉	电压 (V)	描述
R7	PTB2	I/O	4	U	3.3	功能 0: GPIO PTB2, 通用输入输出 功能 1: AK2401, AK_DAC_FS

P7	PTB3	I/O	4	U	3.3	功能 0: GPIO PTB3, 通用输入输出 功能 1: AK2401, AK_DAC_SCLK
T8	PTB4	I/O	4	D	3.3	功能 0: GPIO PTB4, 通用输入输出 功能 1: AK2401, AK_DAC_SDI
R8	PTB5	I/O	4	D	3.3	功能 0: GPIO PTB5, 通用输入输出 功能 1: AK2401, AK_ADC_FS
P8	PTB6	I/O	4	D	3.3	功能 0: GPIO PTB6, 通用输入输出 功能 1: AK2401, AK_ADC_SCLK
R13	PTB7	I/O	4	U	3.3	功能 0: GPIO PTB7, 通用输入输出 功能 1: AK2401, AK_ADC_SDO

### 3.4.12 ADC 管脚

表 16 ADC 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下拉	电压 (V)	描述
B1	ADC0_IN	I/O	4	D	3.3	功能 0: CPU ADC 通道 0 输入 功能 1: GPIO PTAC15, 通用输入输出
A2	ADC1_IN	I/O	4	D	3.3	功能 0: CPU ADC 通道 1 输入 功能 1: GPIO PTAC16, 通用输入输出
B3	ADC2_IN	I/O	4	D	3.3	功能 0: CPU ADC 通道 2 输入 功能 1: GPIO PTAC17, 通用输入输出
A3	ADC3_IN	I/O	4	D	3.3	功能 0: CPU ADC 通道 3 输入 功能 1: GPIO PTAC18, 通用输入输出
C4	ADC4_IN	I/O	4	D	3.3	功能 0: CPU ADC 通道 4 输入 功能 1: GPIO PTAC19, 通用输入输出
B4	ADC5_IN	I/O	4	D	3.3	功能 0: CPU ADC 通道 5 输入 功能 1: GPIO PTAC20, 通用输入输出
A4	ADC6_IN	I/O	4	D	3.3	功能 0: CPU ADC 通道 6 输入 功能 1: GPIO PTAC21, 通用输入输出
A5	ADC7_IN	I/O	4	D	3.3	功能 0: CPU ADC 通道 7 输入 功能 1: GPIO PTAC22, 通用输入输出
N2	ADC_VINP_I	I	-	-	3.3	基带 ADC I 路: 在差分模式下, I 路输入共模电压, 当 EXTCM=1 时候, 有外部提供; 当 EXTCM=0 时候, 该管脚 floating; 在单端模式下, 该管脚连接到模拟地; 在伪差分模式下, 该管脚与 VINM 同输入。
P1	ADC_VINM_I	I	-	-	3.3	基带 ADC I 路: 在差分输入模式下, 为 I 路输入的差分负端; 在伪差分模式下, 固定为 3.3V 电压输入。
P2	ADC_VINCM_I	I	-	-	3.3	基带 ADC I 路: 在差分输入模式下, 为 I 路输入的差分正端; 在为差分和单端模式下, 为信号输入端
M1	ADC_VINP_Q	I	-	-	3.3	基带 ADC Q 路: 在差分模式下, Q 路输入共模电压, 当 EXTCM=1 时候, 有外部提供; 当 EXTCM=0 时候, 该管脚 floating; 在单端模式下, 该管脚连接到模拟地; 在伪差分模式下, 该管脚与 VINM 同输入。
M2	ADC_VINM_Q	I	-	-	3.3	基带 ADC Q 路: 在差分输入模式下, 为 Q 路输入的差分负端; 在伪差分模式下, 固定为 3.3V 电压输入。
N1	ADC_VINCM_Q	I	-	-	3.3	基带 ADC Q 路: 在差分输入模式下, 为 Q 路输入的差分正端;

						在为差分 and 单端模式下，为信号输入端
--	--	--	--	--	--	-----------------------

### 3.4.13 DAC 管脚

表 17 DAC 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下拉	电压 (V)	描述
K1	DAC_VOUT_MCUA	I/O	4	U	3.3	功能 0: CPU DAC A 路输出 功能 1: GPIO PTAC23, 通用输入输出
K2	DAC_VOUT_MCUB	I/O	4	U	3.3	功能 0: CPU DAC B 路输出 功能 1: GPIO PTAC24, 通用输入输出
L1	DAC_VOUT_MCUC	I/O	4	U	3.3	功能 0: CPU DAC C 路输出 功能 1: GPIO PTAC25, 通用输入输出
J1	DAC_VOUT_A	0	-	-	3.3	基带 DAC I 路输出
J2	DAC_VOUT_B	0	-	-	3.3	基带 DAC Q 路输出

### 3.4.14 Codec 管脚

表 18 Codec 管脚

HR_C7000 PIN	管脚名称	类型	驱动 (mA)	上下拉	电压 (V)	描述
A14	Codec_MiC_INP1	I	-	-	-	MIC1 差分输入正端，或 Linein1 单端输入端
B14	Codec_MiC_INN1	I	-	-	-	MIC1 差分输入负端
A13	Codec_MiC_INP2	I	-	-	-	MIC2 差分输入正端，或 Linein2 单端输入端
B13	Codec_MiC_INN2	I	-	-	-	MIC2 差分输入负端
A15	Codec_MICBIAS	I	-	-	-	Codec Mic 偏置电压
C15	Codec_LINOUT1_LP	0	-	-	3.3	Codec LineOut1 输出, PWM 信号
C14	Codec_LINOUT2_LP	0	-	-	3.3	Codec LineOut2 输出, PWM 信号

### 3.4.15 电源和地管脚

表 19 电源地管脚

HR_C7000 PIN	管脚名称	类型	电压 (V)	描述
B2	ADC_VREF_MCU	P	3.3	MCU 外设接口 ADC 的参考电压
C3	ADC_AVDD33_MCU	P	3.3	MCU 外设接口 ADC 的供电电压
C2	ADC_AVSS33_MCU	G	-	MCU 外设接口 ADC 的模拟地
E3	LDO_RTC_AVSS33	G	-	RTC 模块的模拟地。
D3	LDO_RTC_AVDD33	P	3.3	RTC 模块的供电电压, RTC 需单独供电
E5	RTC_AVSS	G	-	RTC 模块的模拟地
F3	RTC_DVDD12	P	1.2	RTC 电路内置 LDO 的 core1.2 电压输出
G3	USB_AVDD33	P	3.3	USB 电源
H2	PR_V	G	-	保留

H1	PR_G	G	-	保留
K5	DAC_VREFN	G	-	基带 DAC 和 MCU DAC 共用的参考电压负端, 外部连接点模拟地
J5	DAC_VREFP	P	3.3	基带 DAC 和 MCU DAC 共用的参考电压正端, 外部连接点模拟 3.3V
J3	DAC_AVDD33	P	3.3	基带 DAC 和 MCU DAC 共用的模拟 3.3V 电源
K3	DAC_AVSS33	G	-	基带 DAC 和 MCU DAC 共用的模拟地
M3	ADC_VREF	P	-	基带 ADC 的模拟参考输出电压, 可以通过 bg_trim 和 buffer_trim 参考配置调制输出电压, 缺省值为 500mV。
N3	ADC_AVSS33	G	-	基带 ADC 的模拟地
R1	ADC_AVDD33	P	3.3	基带 ADC 的模拟 3.3V 供电
L3	ADC_AVDD33	P	3.3	基带 ADC 的模拟 3.3V 供电
M6	PLL_AVDD_1	P	3.3	PLL 模拟 3.3V 供电
M7	PLL_AVSS_0	G	-	PLL 模拟地
P3	LDO_AVDD33_3	P	3.3	内部 LDO 的供电电压, 典型值 3.3V。负责内部 core 电压转换
M5	LDO_DVDD12_3	P	1.2	LDO 输出 1.2 电压参考
P4	LDO_AVSS	G	-	LDO 模拟地
H12	LDO_AVDD33_0	P	3.3	内部 LDO 的供电电压, 典型值 3.3V。负责内部 core 电压转换
K12	LDO_DVDD12_0	P	1.2	LDO 输出 1.2 电压参考
J12	LDO_AVSS	G	-	LDO 模拟地
D16	Codec_LINOUT_VSS	G	-	LineOut1 和 LineOut2 两个 IO 地
C16	DVDD25	P	2.5	LineOut1 和 LineOut2 两个 IO 供电, 电压为 2.5V, 由 Codec_VREFP 提供
B16	Codec_VREFP	P	2.5	Codec 输出 2.5V 参考电压
C13	Codec_AVDD	P	3.3	Codec 的模拟电源 3.3V
D14	Codec_AVSS33	G	-	Codec 的模拟地
B15	Codec_VCAP	P	3.3	Codec 带隙电压
E9	LDO_AVDD33_1	P	3.3	内部 LDO 的供电电压, 典型值 3.3V。负责内部 core 电压转换
E8	LDO_DVDD12_1	P	1.2	LDO 输出 1.2 电压参考
E10	LDO_AVSS	G	-	LDO 模拟地
E12, F12, M11, M12	DVDD33	P	3.3	数字电源 3.3V
A1, A16, C1, C7, D15, E1, E2, E6, E7, E11, E14, F5, F6, F7, F8, F9, F10, F11, G1, G2, G5, G6, G7, G8, G9, G10, G11, G12, H3, H5, H6, H7, H8, H9, H10, H11, J6, J7, J8, J9, J10, J11, J15, K6, K7, K8, K9, K10, K11, L2, L3, L5, L6, L7, L8, L9, L10, L11, L12, M8, M9, M10, T1, T16	DVSS	G	-	数字地

### 3.5 电性能参数

#### 3.5.1 电压参数

表 20 电压参数

	名称	描述	MIN	TYP	MAX	单位
数字电压	DVDD33	数字电源供电	2.97	3.3	3.63	V
Codec 供电	DVDD25	Codec 的 LineOut 数字 PWM I/O 电源供电	2.35	2.5	2.65	V
	Codec_VREFP	Codec 参考电压输出	2.35	2.5	2.65	V
	Codec_MICBIAS	MICBIAS_V=0			2.08V	V
		MICBIAS_V=1			1.66	V
	Codec_AVDD	Codec 模拟电源	2.97	3.3	3.63	V
Codec_VCAP	Codec 带隙输出电压		2		V	
MCU 外设 ADC 电压	ADC_AVDD33_MCU	MCU ADC 供电	2.97	3.3	3.63	V
	ADC_VREF_MCU	MCU ADC 参考电压	0.5* AVDD33		0.9* AVDD33	V
RTC 供电	LDO_RTC_AVDD33	RTC 独立 LDO 供电	2.0	3.3	3.63	V
	RTC_DVDD12	RTC 独立 LDO core 供电	1.08	1.2	1.32	V
USB 供电	USB_AVDD33	USB 供电	2.8	3.3	3.63	V
DAC 供电	DAC_AVDD33	DAC 模拟供电	2.97	3.3	3.63	V
	DAC_VREFP	DAC 参考电压输入	2.97	3.3	3.63	V
PHY ADC 供电	ADC_AVDD33	基带 ADC 模拟供电	3.04	3.3	3.6	V
	ADC_VREF	基带 ADC 的参考电压输出	0.485	0.5	0.515	V
PLL 供电	PLL_AVDD_1	PLL 模拟供电	2.97	3.3	3.63	V
LDO 供电	LDO_AVDD33_1	LDO 1 模拟供电	2.97	3.3	3.63	V
	LDO_AVDD33_2	LDO 2 模拟供电	2.97	3.3	3.63	V
	LDO_AVDD33_3	LDO 3 模拟供电	2.97	3.3	3.63	V
	LDO_DVDD12_1	LDO 1 转换电压输出	1.08	1.2	1.32	V
	LDO_DVDD12_1	LDO 2 转换电压输出	1.08	1.2	1.32	V
	LDO_DVDD12_1	LDO 3 转换电压输出	1.08	1.2	1.32	V
参考地	DVSS	数字地		0		V
	Codec_LINOUT_VSS	Codec LineOut I/O 的模拟地		0		V
	Codec_AVSS33	Codec 模拟地		0		V
	ADC_AVSS33_MCU	MCU ADC 模拟地		0		V
	LDO_RTC_AVSS33	RTL 自带 LDO 模拟地		0		V
	RTC_AVSS	RTC 模拟地		0		V
	USB_AVSS33	USB 模拟地		0		V
	DAC_AVSS33	DAC 模拟地		0		V
	DAC_VREFN	DAC 负端参考		0		V
	ADC_AVSS33	ADC 模拟地		0		V
	PLL_AVSS_0	PLL 模拟地		0		V
LDO_AVSS	LDO 模拟地		0		V	
输入输出 电压门限	VIH	输入高电压	2.0		3.63	V
	VIL	输入低电压	-0.3		0.8	V
	VOL	输出低电压			0.4	V
	VOH	输出高电压	2.4			V

#### 3.5.2 功耗参数

#### 3.5.3 温度参数

表 21 温度参数

名称	描述	MIN	TYP	MAX	单位
Operating temp		-40	25	85	° C
Junction temp		-40	25	100	° C
Solder temp	器件可在 260C 过三次回流焊，并保持机械和电性能正常（高温区段时间约为 260° C 10s，）； 手工焊接，建议在高温条件时间尽量短，可参考一般返修条件 340° C 5~10s，		260		° C

## 3.5.4 IP 指标参数

表 22 IP 指标参数

分类	参数	描述	MIN	TYP	MAX	单位
DAC	Resistive load	Anolog output		1.5		kOhm
	Capacitve load	Anolog output			30	pF
	Voltage swing	Anolog output	VREFN		VREFP	V
	RES			12		bit
	DNL			±1		LSB
	INL			±1.5		LSB
	SNR					
	THD					
PHY ADC	VINDiff	差分输入电压 (VINP-VINM)			0.9	Vpp
	VINse	单端或伪差分输入电压, VINM=com mode of VINP			0.45	Vpp
	VINcm-dc	VINCM 共模电压 (DC 模式)	0.5		1.5	V
	VINcm-ac	VINCM 共模电压 (AC 模式)		0		V
	Vnol	AVDD Dyamic Noise			50	mVpp
	RES			10		bit
	Cin	VINP/VINM 输入电容		2		pF
	Cref	VREF 参考电压电容		100		nF
	DNL			±1		LSB
	INL			±2		LSB
	Gain Error	Internal reference, before BGTRIM correction	-4		+4	%
	Gain Error	Internal reference, after BGTRIM correction	-1		+1	%
	Gain Error	External reference	-1		+1	%
	OS	Input offset error	-5		+5	mV
	LAT	Analog input to digital output latency		12		cycles
	WT	Wake up time		120		cycles
	BTime	Buffer start-up time		15	20	uS
Total Stime	Complete ADC start up time		WT+ BTime		uS	
SNR			56		dB	
ENOB			9		bit	
MCU ADC	Input Range	8-channel single-ended input	0.01 VREF		0.99 VREF	V
	Input capacitance			1		pF
	RES				10	bit
	DNL		-0.2852		0.3599	LSB
	INL		-0.7344		0.9160	LSB
	Offset Error			0.5064		%FS
	Gain Error			-0.0432		%FS
	Data Latency			10		cycle
	SINAD	Fin = 1.03K		57.91		dB
	SFDR	Fin = 1.03K		63.81		dB
	THD +3HD	Fin = 1.03K		-64.64		dB
ENOB	Fin = 1.03K		9.33		bit	
USB	VOL	1.5K pull-up resistor on DP to USB_AVDD33			0.3	V
	VOH	15K pull-down reisor on DP/DM to AGND	2.8			V
	Vdiff	V(DP)-V(DM)	0.2			V
	Vcm	USB_AVDD33 > Vcm+0.8	0.8		2.5	V
	Vse-l	Single-ended recever low level input voltage			0.8	V
	Vse-h	Single-ended recever high level input voltage	2.0			V
	Fs	Sampling frequency	8	16	192	KHz
	Mclk			24		MHz

Codec	Duty cycle	Mclk	0.45	0.5	0.55	
	WT	From power-down to playback mode		181	493	ms
		From power-down to record mode		124	230	ms
	Micbias out current				4	mA
	Micbias out noise	A-weighted		20	40	uVrms
	Micbias capacitor	Decoupling capacitor Cmic	0.75	1	1.25	nF
	Input level	ADC full scale, Gain GID=0dB, GIM=20dB	0.189	0.212	0.239	Vpp
	SNR	ADC, A-weighted, 1KHz sine wave, full scale and gain 0dB	85	90		dB
	THD	ADC, 1KHz sine wave -1dB and gain 0dB		-80	-70	dB
	Dynamic range	ADC, A-weighted, 1KHz sine wave, full scale and gain 0dB	85	90		dB
	PSRR	ADC, 100mVpp 1KHz sinewave is applied to AVD, Gain GID=0dB, GIM=20dB		90		dB
	Input resisitance	ADC, GIM=0dB, differential config	132	160	200	KOhm
		ADC, GIM=20dB, differential	20	26	30	KOhm
		ADC, GIM=0dB, single-ended	92	115	138	KOhm
		ADC, GIM=20dB, single-ended	19	24	29	KOhm
	Input capacitance	ADC, Cbyline		1		uF
		ADC, includes 10uf for ESD, bongding and package pins cap			25	pF
	SNR	PWM out, A-weighted, 1KHz sine wave, full scale and gain 0dB	100	105		dB
	THD	PWM out, 1KHz sine wave -1dB and gain 0dB, Fs <=16KHz		-88	-80	dB
	THD+N	PWM out, 1KHz sine wave -1dB and gain 0dB, Fs <=16KHz		-87	-80	dB
	Dynamic Range	PWM out, 1KHz sine wave, full scale -60dB and volume=0dB	100	105		dB
	Gain range	PWM out, Godl 6-bit programmable range	-31		32	dB
	Gain step	PWM out,		1		dB
PWM frequency	MCLK =24MHz		750		kHz	
Modulation rate		25		75	%	
LDO	I max	Load current			100	mA
	Vtrip	Max ripple on Vout, Vin = 3.3V			100	mVpp
	Cout	External capacitor			4.7	uF

## 4 系统

### 4.1 复位

#### 4.1.1 概述

复位管理模块对整个芯片的复位、各功能模块的复位进行统一的管理，包括：

- 内部集成 POR 模块
- 上电复位的管理和控制
- 功能模块单独软复位控制
- 复位信号同步到各模块对应时钟域

#### 4.1.2 功能描述

HR\_C7000 芯片内置上电自动复位模块，在芯片管脚 POR\_EN 处于高电平有效的情况下，内部自动复用模块工作，芯片被全局复位，同时外部复位管脚 NRESET 也能起到复位作用。

当芯片管脚 POR\_EN 处于低电平的时候，内部自动复位电路不工作，芯片的全局复位由外部复位管脚 NRESET 完成。为保证外部复位有效，要求 NRESET 的低电平持续时间超过 400ns 以上。

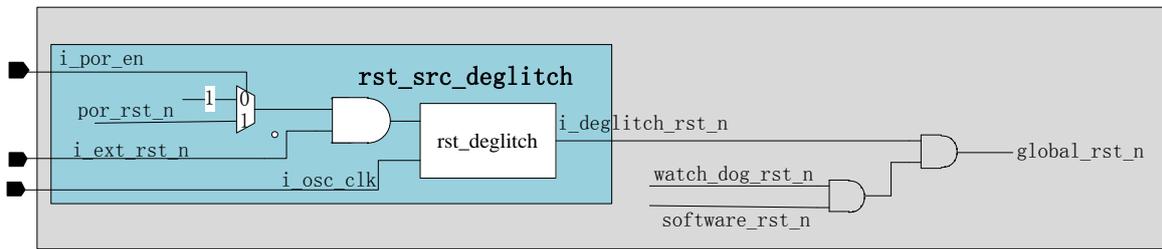


图 7 复位整体框图

芯片除了外部管脚或者内部 POR 模块进行的全局硬件复位之外，watch\_dog 异常也能触发系统的全局复位。另外，系统内部设置全部软复位寄存器，也能对系统各个模块（除了寄存器配置模块之外）进行主动复位。

内部模块功能除芯片整体复位控制外，还可以通过各自软复位进行复位，分为寄存器软复位和功能模块请求软复位。

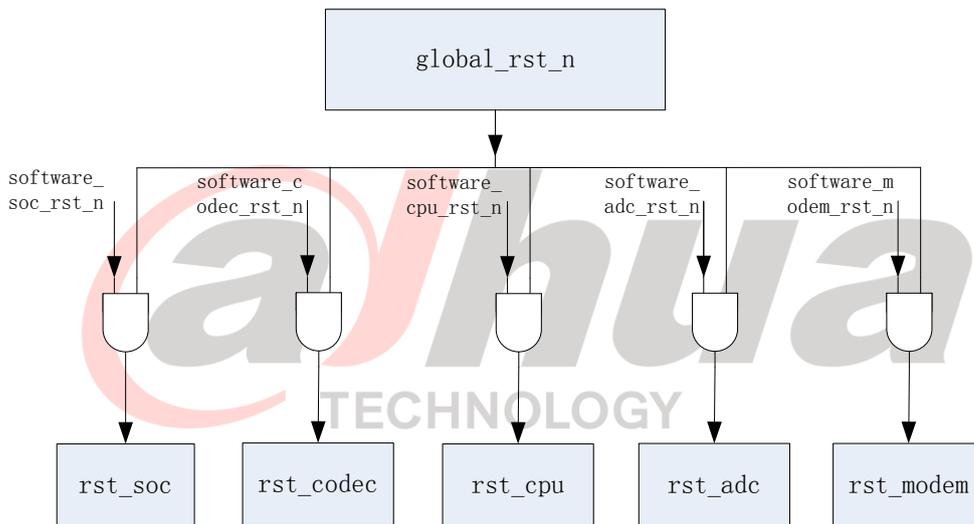


图 8 复位整体框图

#### 4.1.2.1 POR

POR (Power On Reset) 模块通过检测上电 VDD 电压来产生复位信号，当管脚 POR\_EN 信号为高时，内部 POR 有效，否则无效，通过外部复位管脚 RST\_N\_IN 实现芯片整体复位。具体模块框图和时序图如下图：

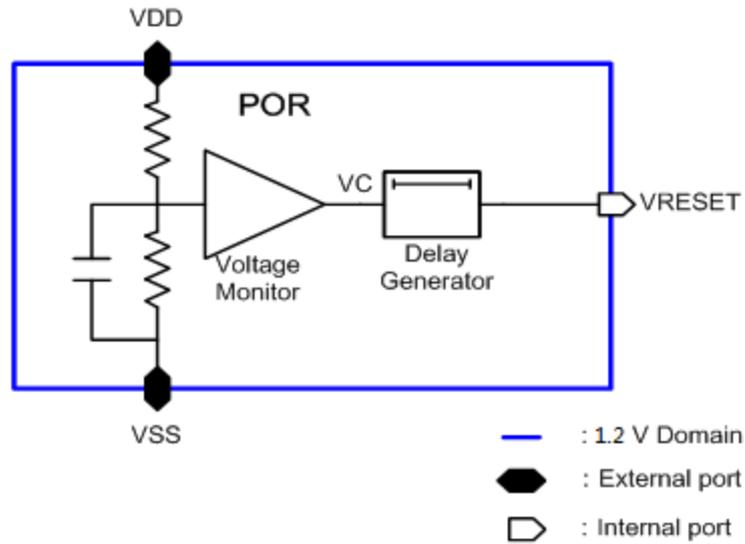


图 9 POR 模块框图

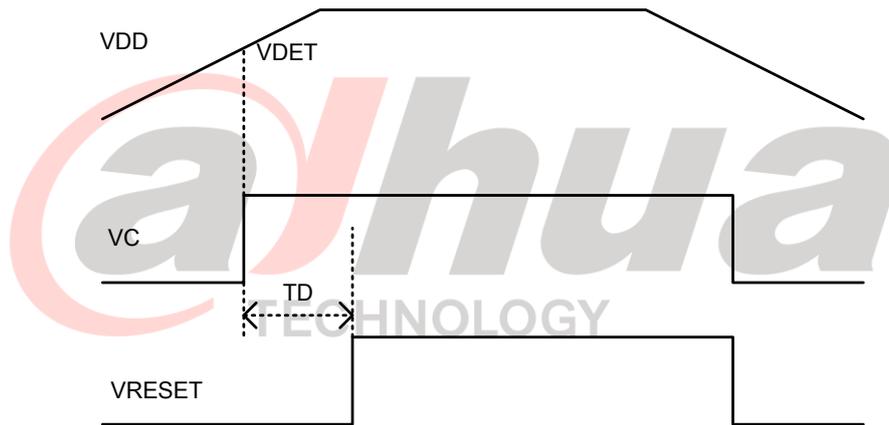


图 10 POR 模块时序图

表 23 POR 设计参数表

PARAMETER	SYMBOL	DESIGN RECOMMANDATONS			UNITS
		MIN	TYP	MAX	
Power Supply	VDD	1.08	1.2	1.32	V
Total Current	I <sub>CC</sub>	1.95	2.7	20.3	uA
Time delay	TD	35	50	80	mS
Detect Voltage	VDET	0.65	0.785	0.9	V

#### 4.1.2.2 软复位

芯片支持对各个模块的软复位，CPU 通过配置软复位寄存器，即可产生软复位信号。采用纯软件的方式 1，通过对软复位寄存器写 0 代表复位有效，自动释放软复位寄存器值为 1。

### 4.1.3 寄存器概述

寄存器表 1 复位寄存器概述（基地址：0x11000000）

偏移地址	名称	描述	页码
0x00	SYS_SOFT_RSTN	软复位寄存器控制	55

### 4.1.4 寄存器描述

#### 4.1.4.1 SYS\_SOFT\_RSTN

偏移地址：0x00

复位方式：H

Bits	Access	Name	Default	Description
[31:9]	-	reserved		
[8]	RW	cpu_soft_rstn	0x1	CPU 软复位控制，低有效
[7]	RW	sys_soft_rstn	0x1	系统全局软复位控制，低有效
[6]	RW	adc_ctrl_soft_rstn	0x1	内置CPU IP ADC软复位控制，低有效
[5]	RW	adc_soft_rstn	0x1	内置基带IP ADC软复位控制，低有效
[4]	RW	codec_soft_rstn	0x1	内置IP CODEC软复位控制，低有效
[3]	RW	audio_soft_rstn	0x1	音频接口模块软复位控制，低有效
[2]	RW	fm_soft_rstn	0x1	FM模块软复位控制，低有效
[1]	RW	phy_soft_rstn	0x1	物理层模块软复位控制，低有效
[0]	RW	protocol_soft_rstn	0x1	协议层模块软复位控制，低有效

## 4.2 时钟

### 4.2.1 概述

时钟管理模块对芯片时钟输入、时钟生成和控制进行统一的管理，包括：

- 时钟输入的管理和控制
- 时钟分频和控制
- 生成各模块的工作时钟

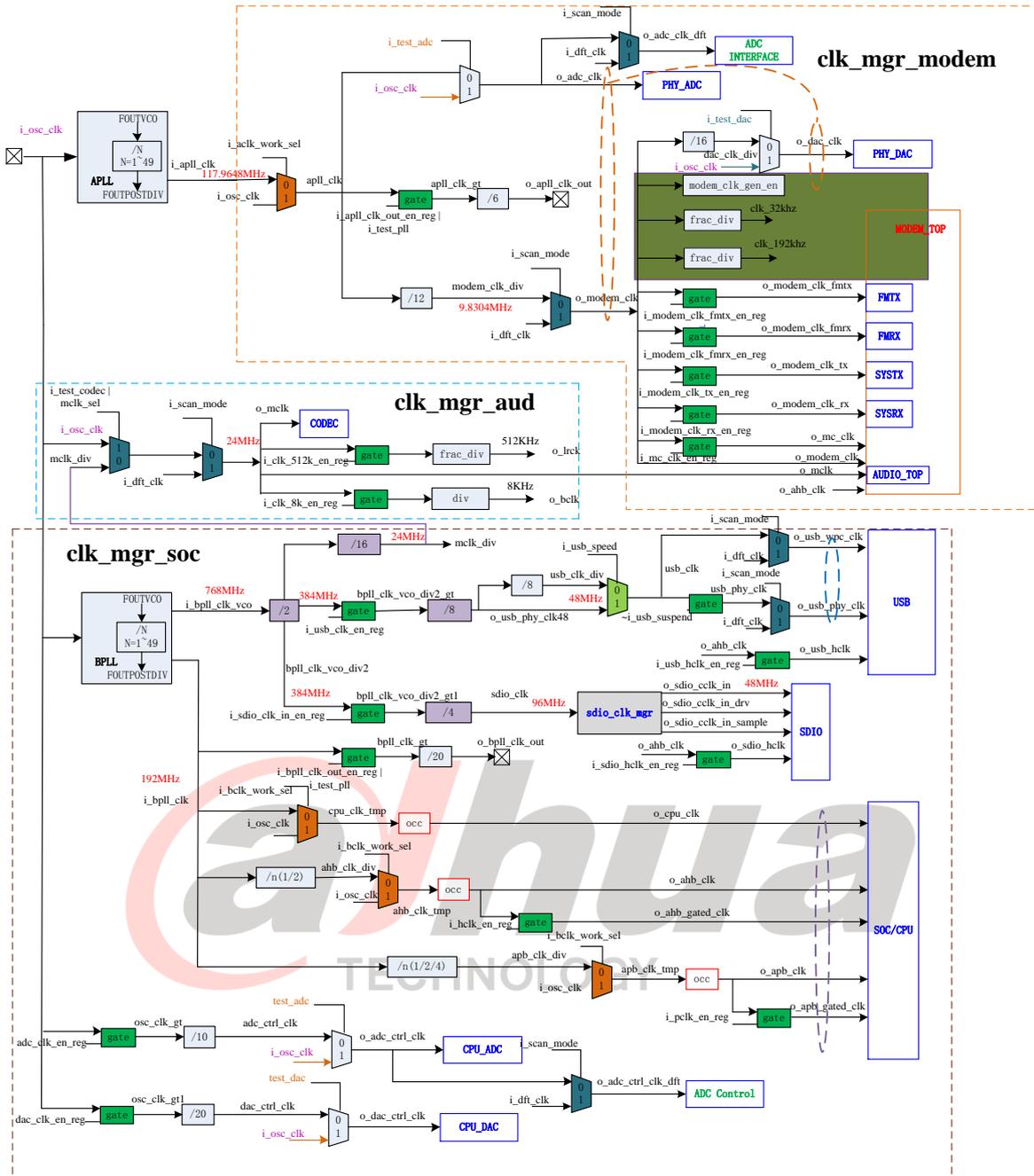


图 11 时钟分布框图

#### 4.2.2 功能描述

时钟单元有以下特点：

- 外置晶振提供 24MHz 时钟工作；
- 内部集成 2 个 PLL  
APLL 用于专门产生 Modem 的源时钟  
BPLL 用于产生系统时钟，如 CPU 时钟，AHB/APB 总线时钟
- 支持运行状态下对时钟的实时开关

#### 4.2.2.1 时钟接口

时钟接口信号如下表所示：

表 24 时钟接口信号

信号名	方向	描述
OSC_CLK_IN	I	外部晶振时钟输入，要求输入 Vpp 达 2.8V 以上
CLK_OUT	O	PLL 分频时钟输出，PLLA 和 PLLB 分频输出可选

#### 4.2.2.2 时钟频率

表 25 时钟频率范围

名称	描述	MIN	TYP	MAX	单位
OSC_CLK_IN	主时钟晶振	--	24	--	MHz
adc_clk	基带 ADC 工作时钟	--	117.9648	--	MHz
dac_clk	基带 DAC 工作时钟	0.0384	0.6144	1	MHz
mdem_clk	Modem 基带工作时钟	--	9.8304	--	MHz
mdem_clk_fm_tx	Modem 基带 FM 发送时钟	--	9.8304	--	MHz
mdem_clk_fm_rx	Modem 基带 FM 接收时钟	--	9.8304	--	MHz
mdem_clk_tx	Modem 基带发送时钟	--	9.8304	--	MHz
mdem_clk_rx	Modem 基带接收时钟	--	9.8304	--	MHz
mc_clk	Codec MC 接口时钟	--	9.8304	--	MHz
mclk	Codec 工作主时钟	--	24	--	MHz
bpll_clk_vco	BPLL VCO 输出时钟	--	--	768	MHz
usb_wpc_clk	usb 工作时钟	6	48	--	MHz
usb_phy_clk	usb 工作时钟	6	48	--	MHz
sdio_cclk_in	sdio 工作时钟	40	48	48	MHz
cpu_clk	cpu 内核工作时钟	24	120	192	MHz
ahb_clk	AHB 总线时钟	24	120	192	MHz
apb_clk	APB 总线时钟	24	60	96	MHz
adc_ctrl_clk	MCU 自带 ADC 工作时钟			2.2	MHz
dac_ctrl_clk	MCU 自带 DAC 工作时钟			1	MHz

#### 4.2.3 工作方式

##### 4.2.3.1 初始化

系统上电后，CPU 和总线均工作在 24M 晶振时钟。

##### 4.2.3.2 重配置操作流程

重配置流程是系统在默认时钟配置下正常启动后改变时钟配置的过程。

重配置的动作顺序是先准备需要配置的参数，将其写到相应寄存器，然后进行重配置使能，之后重配置动作就会自动完成。

如果希望系统工作于默认的 PLL 时钟配置下，只需要对 0x11000000 bit[1]、bit[0]写 1 即可，

各个带有门控的时钟默认是关闭的，如需打开门控，可在线实时配置。

在任何时候如果想改变 PLL 的配置，请先操作 PLL 的配置寄存器，之后进行对 0x11000000 bit[1]、bit[0]写 1 即可，重配置时，PLL 会暂时 POWER DOWN，CPU 和总线时钟会自动切换到晶振时钟下工作，在 PLL lock 后，将 CPU 和总线时钟配置寄存器 0x11000000 bit[3]、bit[2]配置为 0 切回到 PLL 时钟下工作。

重配置使用方法和重配置的流程参考下图，参考步骤如下：

上电时钟配置工作流程：

- 1、上电；
- 2、CK 从 bootrom 启动；
- 3、开始进行时钟配置；
- 4、时钟管理模块复位后的默认配置可产生主流工作场景下的工作频率，但各个时钟的门控是关闭的，需要 CPU 通过寄存器配置将其打开，开关时钟可在线实时进行
- 5、任何时候只有配置寄存器选择 PLL 输出时钟，工作时钟配置才会有效，系统起来后默认工作在 osc 时钟
- 6、若偏好默认配置，只需配置寄存器 CLK\_MGR\_REG0x04 的 bit0 和 bit1 为 1 表示开启 PLL 配置，如希望自定义配置，则需通过 CK 一一操作相应的 PLL 分频及相应工作时钟分频系数寄存器，再配置寄存器 CLK\_MGR\_REG0x04 的 bit0 和 bit1 为 1 表示开启 PLL 配置，之后 PLL 进入 POWER DOWN 状态(无论 PLL 当前是否为 POWER DOWN，该动作都会被执行)
- 7、芯片内部自动进行 PLL 配置过程，其中 PLL 执行 POWER DOWN 的时间约 2us；
- 8、通过读寄存器 CLK\_MGR\_REG0x04 的 Bit31，判断两个 PLL 是否都锁定标志，由于寄存器读写时间的需求，要求 PLL 锁定标志连续判断 10 次以上才可以认定 PLL 锁定，或者之间要求等待 100us 以后读取寄存器判断是否锁定；
- 9、PLL 锁定后，开始切换系统工作时钟，从晶振时钟切换到 PLL 输出的时钟，配置寄存器 CLK\_MGR\_REG0x04 的 Bit3 和 Bit2 为 0；
- 10、等待时钟准备 OK 标志，读取寄存器 CLK\_MGR\_REG0x04 的 Bit30 为高即可，由于寄存器读写时间的需求，要求 clk\_rdy 标志连续判断 10 次以上才可以认定；
- 11、时钟准备 OK 后，系统全局软复位操作；
- 12、若要重新配置时钟频率，重新执行 3—11。

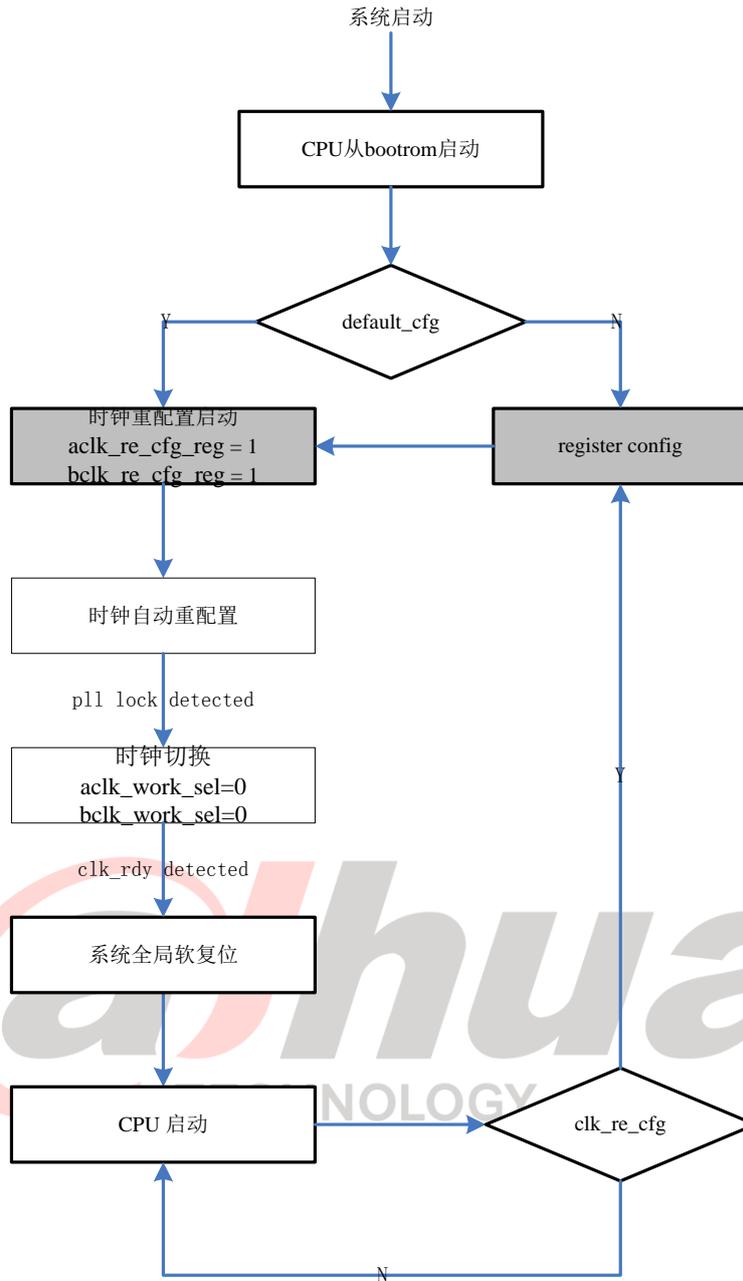


图 12 重配置流程图

灰色填充框为需要软件进行的操作。register\_config 表示 CPU 准备各个配置参数。

时钟管理模块是通过检验 clk\_cfg\_rdy 的上升沿来检测时钟重配置请求的。

时钟管理模块只有一个复位来源——上电复位，时钟重配置参数一旦经过写 CPU 写操作，便会暂存到时钟管理模块内部的暂存器中，不会因为 PLL 失锁引起的复位将其清掉。

#### 4.2.3.3 PLL 配置

芯片内部集成了 2 个 PLL，APLL 和 BPLL，每个 PLL 使用独立的配置寄存器，其中 APLL 工作在小数模式，BPLL 工作在整数模式。

PLL 频率计算方法：

- DSMPD=1，整数模式：

$$F_{OUTVCO} = F_{REF} / REFDIV * FB DIV$$

- $FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2$   
 ● DSMPD=0, 小数模式:  
 $FOUTVCO = FRED / REFDIV * (FBDIV + FRAC / (2^{24}))$   
 $FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2$

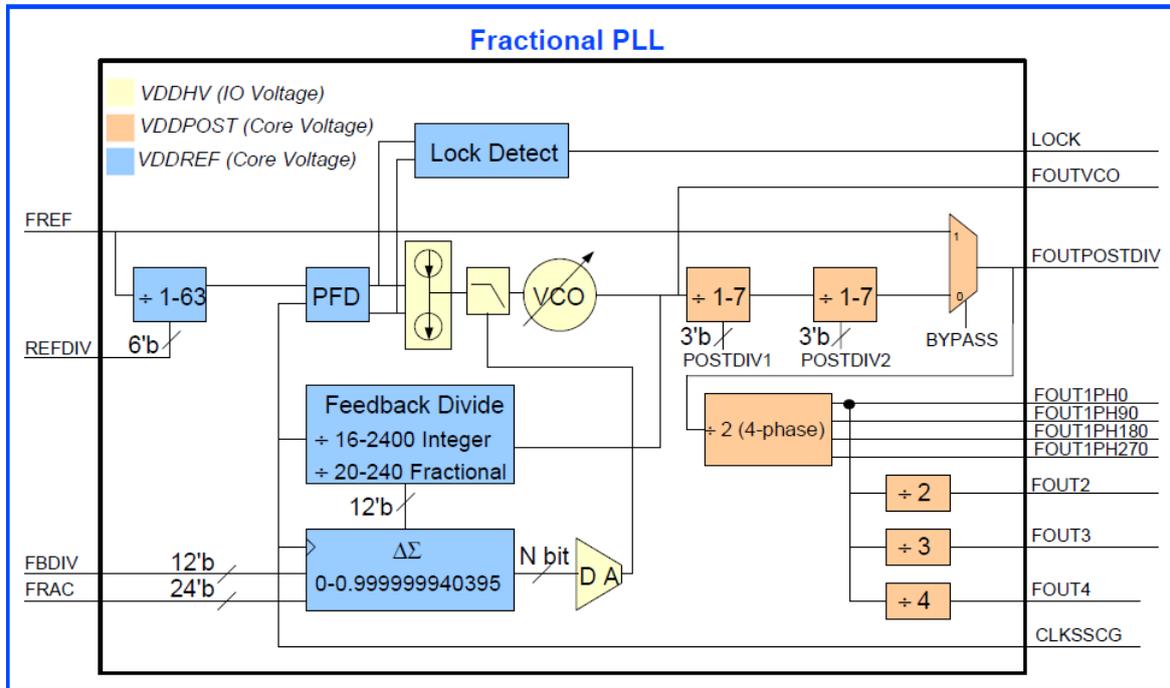


图 13 PLL 电路示意图

#### 4.2.4 寄存器概述

寄存器表 2 时钟管理模块寄存器概述 (基地址: 0x11000000)

偏移地址	名称	描述	页码
0x04	CLK_MGR_REG0x04	PLL重配置控制寄存器	
0x08	CLK_MGR_REG0x08	APLL分频配置寄存器	
0x0c	CLK_MGR_REG0x0c	APLL分频配置寄存器	
0x10	CLK_MGR_REG0x10	BPLL分频配置寄存器	
0x14	CLK_MGR_REG0x14	BPLL分频配置寄存器	
0x18	CLK_MGR_REG0x18	基带相关时钟分频系数寄存器	
0x1c	CLK_MGR_REG0x1c	FM 32K、192K分频系数寄存器	
0x20	CLK_MGR_REG0x20	Code I2S接口时钟分频系数寄存器	
0x24	CLK_MGR_REG0x24	总线相关时钟分频系数寄存器	
0x28	CLK_MGR_REG0x28	SDIO、USB时钟分频系数寄存器	
0x2c	CLK_MGR_REG0x2c	门控时钟控制开关寄存器	

#### 4.2.5 寄存器描述

##### 4.2.5.1 CLK\_MGR\_REG0x04

偏移地址: 0x04

复位方式: H

Bits	Access	Name	Default	Description
[31]	RO	pll_ld	0x0	总的PLL锁定标志(apll_ld&bpll_ld)
[30]	RO	clk_rdy	0x0	时钟切换准备好标志
[29:4]	-	reserved		
[3]	RW	bclk_work_sel	0x1	SOC系统时钟控制信号，复位后默认值为1。 0: SOC系统工作在PLL时钟； 1: SOC系统工作在晶振时钟。
[2]	RW	aclk_work_sel	0x1	基带系统时钟控制信号，复位后默认值为1。 0: 基带系统工作在PLL时钟； 1: 基带系统工作在晶振时钟。
[1]	RW	bclk_re_cfg_reg	0x0	BPLL时钟重配置信息准备好标志，由CPU发出，高有效
[0]	RW	aclk_re_cfg_reg	0x0	APLL时钟重配置信息准备好标志，由CPU发出，高有效

#### 4.2.5.2 CLK\_MGR\_REG0x08

偏移地址: 0x08

复位方式: H

Bits	Access	Name	Default	Description
[31]	RO	apll_ld	0x0	APLL 时钟输出锁定标志（系统起来后一定为1）
[30]	RO	apll_pd	0x0	APLL POWER DOWN标志： 1: power down; 0: 非power down。
[29]	RW	apll_dsmpd_reg	0x0	Power down Delta-Sigma Modulator: 0: DSM is active; 1: DSM is powered down。
[28]	RW	apll_dacpd_reg	0x0	Power down noise canceling DAC in FRAC mode: 0: DAC is active (default mode); 1: DAC is not active (test mode only)。
[27]	RW	apll_bypass_reg	0x0	APLL FOUTPOSTDIV旁路到FREF，高有效
[26]	RW	apll_foutvcopd_reg	0x1	APLL post vco输出关闭使能，高有效
[25]	RW	apll_foutpostdivpd_reg	0x0	APLL输出分频控制
[24]	RW	apll_fout4phasepd_reg	0x1	4phase 时钟产生器关闭使能(4phase输出也可以由apll_foutpostdivpd关闭)，高有效
[23:18]	RW	apll_refdiv_reg	0x1	APLL参考时钟分频系数(1~63)
[17:6]	RW	apll_fbdiv_reg	0x3a	APLL feedback div value(16~2400 in integer mode, 20~240 in fraction mode)

[5:3]	RW	apl_postdiv1_reg	0x6	APLL post div1(1~7)
[2:0]	RW	apl_postdiv2_reg	0x2	APLL post div2(1~7)

#### 4.2.5.3 CLK\_MGR\_REG0x0c

偏移地址: 0x0c

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	apl_frac	0xfb7e91	APLL小数部分分频值

#### 4.2.5.4 CLK\_MGR\_REG0x10

偏移地址: 0x10

复位方式: H

Bits	Access	Name	Default	Description
[31]	RO	bpll_ld	0x0	BPLL 时钟输出锁定标志 (系统起来后一定为1)
[30]	RO	bpll_pd	0x0	BPLL POWER DOWN标志: 1: power down; 0: 非power down。
[29]	RW	bpll_dsmpd_reg	0x1	Power down Delta-Sigma Modulator: 0: DSM is active; 1: DSM is powered down。
[28]	RW	bpll_dacpd_reg	0x1	Power down noise canceling DAC in FRAC mode: 0: DAC is active (default mode); 1: DAC is not active (test mode only)。
[27]	RW	bpll_bypass_reg	0x0	BPLL FOUTPOSTDIV旁路到FREF, 高有效
[26]	RW	bpll_foutvcopd_reg	0x0	BPLL post vco输出关闭使能, 高有效
[25]	RW	bpll_foutpostdivpd_reg	0x0	BPLL输出分频控制
[24]	RW	bpll_fout4phasepd_reg	0x1	4phase 时钟产生器关闭使能(4phase输出也可以由bpll_foutpostdivpd关闭), 高有效
[23:18]	RW	bpll_refdiv_reg	0x2	BPLL参考时钟分频系数(1~63)
[17:6]	RW	bpll_fbdiv_reg	0x28	BPLL feedback div value(16~2400 in integer mode, 20~240 in fraction mode)
[5:3]	RW	bpll_postdiv1_reg	0x2	BPLL post div1(1~7)
[2:0]	RW	bpll_postdiv2_reg	0x2	BPLL post div2(1~7)

#### 4.2.5.5 CLK\_MGR\_REG0x14

偏移地址: 0x14

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	bpll_frac	0x0	BPLL小数部分分频值

#### 4.2.5.6 CLK\_MGR\_REG0x18

偏移地址：0x18

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	RW	dac_clk_div_num	0x10	IP DAC工作时钟分频系数设置
[23:16]	RW	bclk_out_div_num	0xa	BPLL管脚输出时钟分频系数设置
[15:8]	RW	aclk_out_div_num	0xc	APLL管脚输出时钟分频系数设置
[7:0]	RW	modem_clk_div_num	0xc	基带工作系统时钟分频系数设置

#### 4.2.5.7 CLK\_MGR\_REG0x1c

偏移地址：0x1c

复位方式：H

Bits	Access	Name	Default	Description
[31:27]	RW	clk_32k_div_denom	0x5	32K小数分频系数——分母
[26:16]	RW	clk_32k_div_numer	0x600	32K小数分频系数——分子
[15:11]	RW	clk_192k_div_denom	0x5	192K小数分频系数——分母
[10:0]	RW	clk_192k_div_numer	0x100	192K小数分频系数——分子

#### 4.2.5.8 CLK\_MGR\_REG0x20

偏移地址：0x20

复位方式：H

Bits	Access	Name	Default	Description
[31]	RW	mclk_sel_reg	0x1	Codec MCLK 时钟选择： 0：PLL输出时钟分频所得； 1：晶振时钟。
[30:26]	RW	mclk_div_num	0x9	Codec MCLK所需24MHz时钟分频系数， mclk_div_num+1对应分频，时钟源为 BPLL的VCO输出时钟的2分频所得。
[25]	-	reserved		
[24:13]	RW	clk_8k_div_num	0xbb8	8K分频系数
[12:9]	RW	clk_512k_div_denom	0x8	512K小数分频系数——分母
[8:0]	RW	clk_512k_div_numer	0x177	512K小数分频系数——分子

#### 4.2.5.9 CLK\_MGR\_REG0x24

偏移地址：0x24

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	cpu_dac_clk_div_num	0x18	CPU DAC工作时钟分频系数设置,0表示不分频, dac_clk_div_num+1对应分频倍数
[15:12]	-	reserved		
[11:10]	RW	sdio_clk_dly_reg_drv	0x2	SDIO的cclk_in_drv时钟延时 延时 = n/4*cclk_in, 列如cclk_in=20ns, n=1时, 延时=5ns, 也即90°相移
[9:8]	RW	sdio_clk_dly_reg_sample	0x0	SDIO的cclk_in_sample时钟延时 延时 = n/4*cclk_in, 列如cclk_in=20ns, n=1时, 延时=5ns, 也即90°相移
[7:4]	RW	adc_clk_div_num	0xa	CPU ADC工作时钟分频系数设置,0表示不分频, adc_clk_div_num+1对应分频倍数
[3:1]	RW	apb_clk_div_num	0x1	AHB和APB时钟频率关系配置寄存器: 0, AHB:APB=1:1; 1, AHB:APB=1:2; 2, AHB:APB=1:4;
[0]	RW	cpu_clk_ratio_reg	0x0	AHB和CPU时钟频率关系配置寄存器: 0, AHB:CPU=1:1; 1, AHB:CPU=1:2;

#### 4.2.5.10 CLK\_MGR\_REG0x28

偏移地址: 0x28

复位方式: H

Bits	Access	Name	Default	Description
[31:21]	-	reserved		
[20:16]	RW	sdio_clk_div_num	0x2	SDIO时钟的2倍时钟(默认80MHz, 最高96MHz)分频系数, sdio_clk_div_num+1对应分频, 时钟源为BPLL的VCO输出时钟的2分频所得。
[15:5]	-	reserved		
[4:0]	RW	usb_clk_div_num	0x4	USB所需48MHz时钟分频系数, usb_clk_div_num+1对应分频, 时钟源为BPLL的VCO输出时钟的2分频所得。

#### 4.2.5.11 CLK\_MGR\_REG0x2c

偏移地址: 0x2c

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23]	RW	dac_clk_en	0x0	CPU DAC时钟门控使能, 高有效
[22]	RW	adc_clk_en	0x0	CPU低速ADC时钟门控使能, 高有效

[21]	RW	hclk_en	0x0	可控AHB总线上的外设的hclk时钟门控使能，高有效
[20]	RW	pclk_en	0x0	可控APB总线上的外设的pclk时钟门控使能，高有效
[19]	RW	sdio_hclk_en	0x0	sdio hclk输入时钟门控使能，高有效
[18]	RW	sdio_clk_en	0x0	sdio时钟门控使能，高有效
[17]	RW	usb_hclk_en	0x0	usb hclk输入时钟门控使能，高有效
[16]	RW	usb_clk_en	0x0	usb时钟门控使能，高有效
[15:11]	-	reserved		
[10]	RW	mc_clk_en	0x0	Codec MC接口配置时钟使能，高有效
[9]	RW	modem_clk_rx_en	0x0	基带接收时钟门控使能，高有效
[8]	RW	modem_clk_tx_en	0x0	基带发送时钟门控使能，高有效
[7]	RW	modem_clk_fmrx_en	0x0	FM接收时钟门控使能，高有效
[6]	RW	modem_clk_fmrx_en	0x0	FM发送时钟门控使能，高有效
[5]	-	reserved		
[4]	-	reserved		
[3]	RW	clk_512k_en_reg	0x0	512K时钟门控使能，高有效
[2]	RW	clk_8k_en_reg	0x0	8K时钟门控使能，高有效
[1]	RW	bclk_out_en_reg	0x0	BPLL管脚输出时钟门控使能，高有效
[0]	RW	aclk_out_en_reg	0x0	APLL管脚输出时钟门控使能，高有效

### 4.3 低功耗控制

HR\_C7000 芯片内部功耗分布主要在以下图所示的功能模块中。

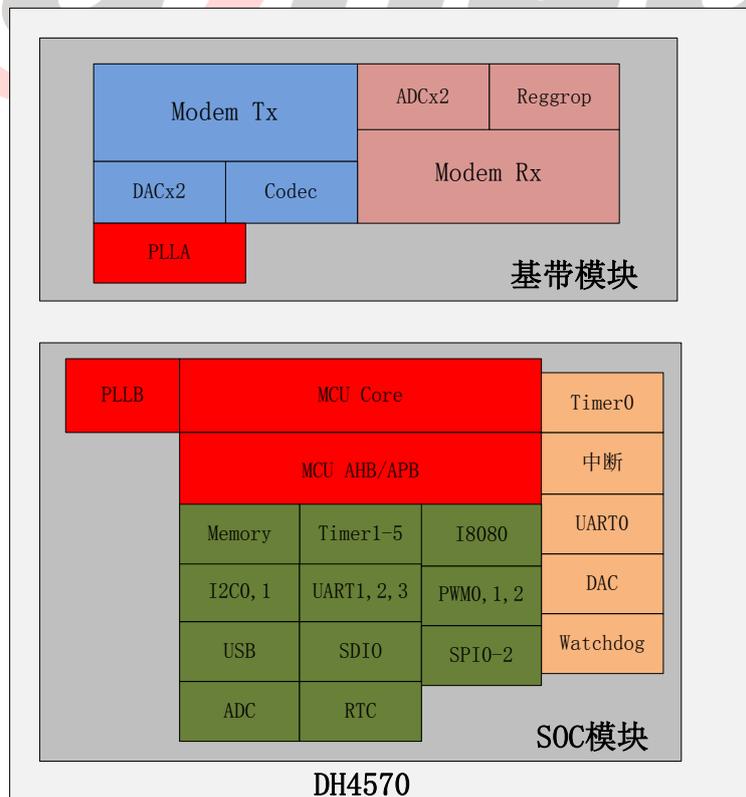


图 14 HR\_C7000 功能模块划分

HR\_C7000 主要功能模块分为两大部分：基带模块和 SOC 模块，其中基带模块包含有数字调制解调部分（Modem Tx 和 Modem Rx，以及相关的寄存器 Reggrop）；模拟部分包括两路 ADC 和两路 DAC，以及路音频 Codec 和一个 PLL 模块（PLLA）。

SOC 模块由 PLLB 模块产生时钟，主要功能块包括 MCU Core 单元,MCU 总线(AHB/APB) 以及总线的外设等部分。

以上这些模块功能在待机情况下，按照功能需要可以对部分模块进行时钟门控，按照上图中不同颜色区域模块，可以分别按照不同的控制方式进行处理，具体可以分成下图所示的 5 个功能类。

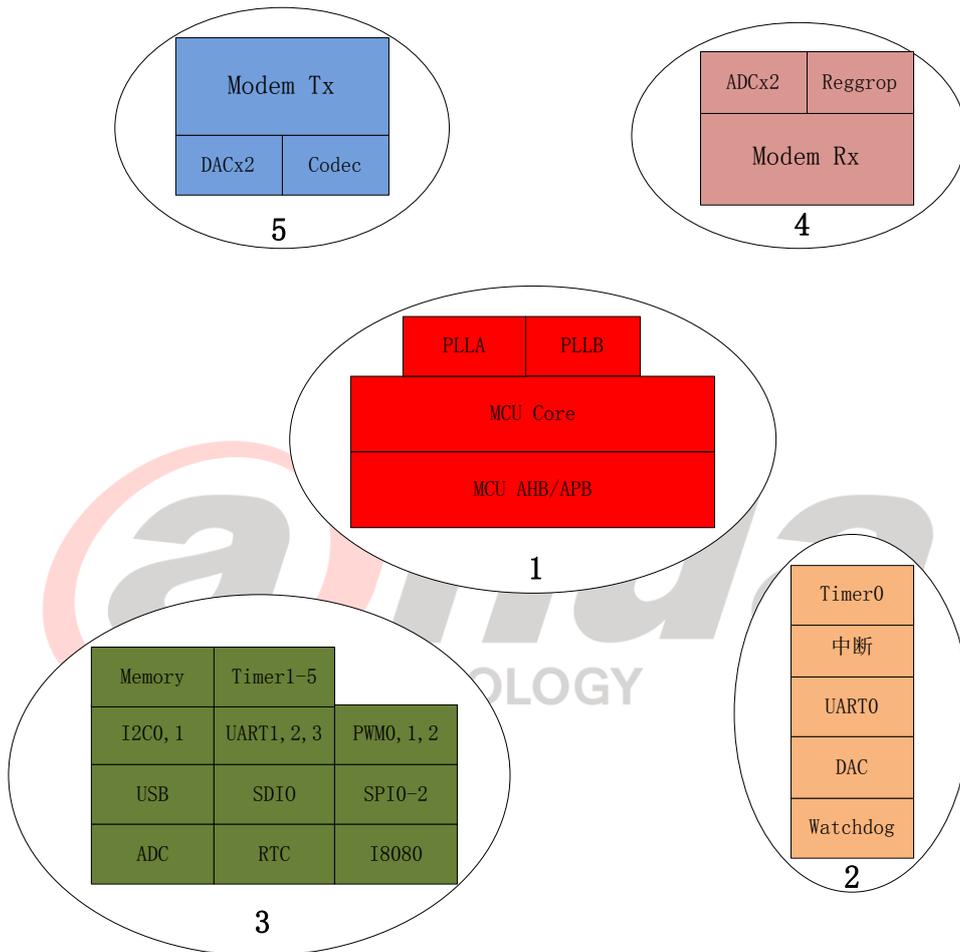


图 15 HR\_C7000 功能功耗控制分布

在待机模式下，2 和 4 包含的模块需要开启状态，保证正常的信号和中断的接入。1 包含的模块可以工作在低时钟频率（MCUCore 和 AHB 时钟和 APB 时钟均工作在晶振时钟）。其中在中频接收方案时候，4 包含的模块仅仅需要开启一个 ADC 工作，另外一个 ADC 可以关闭。

语音正常接收过程中，根据接收开始对应的提示中断，开关 1 包括的 Codec 的 DAC 的功能；接收完成后，重新关闭 Codec，回到待机状态。

语音发送过程中，根据发送开始对应的提示中断，开启基带的 DAC 和 Codec 的 ADC 功能；传统的单点调制方案仅仅需要开启一路 DAC。发送完成后，重新关闭 Codec 和 DAC；回到待机状态。

数据发送和接收过程中，根据发送和接收开始中断，开启基带的 DAC 功能和 Codec 的 DAC

功能完成基带收发和数据收发的提示音播放；收发完成后，关闭基带 DAC 和 Codec 的 DAC，重新返回待机状态。

具体流程如下图所示。

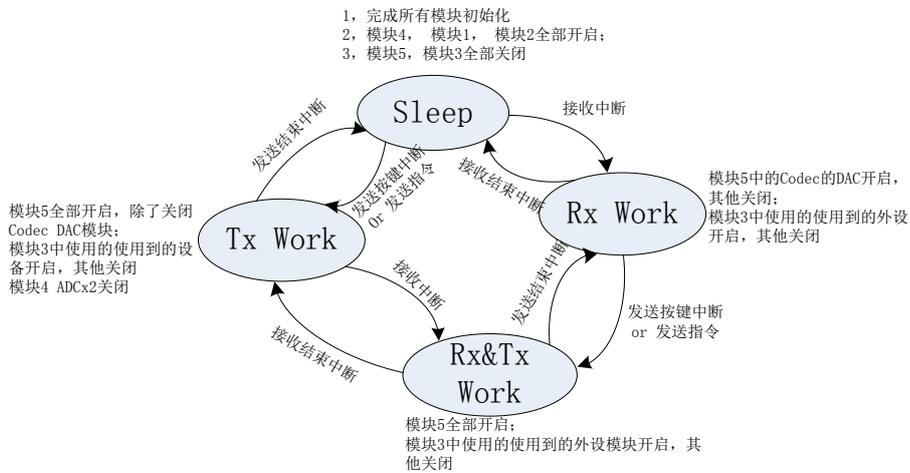


图 16 HR\_C7000 低功耗工作流程

## 4.4 系统控制

### 4.4.1 概述

系统控制主要包括 LCSFC SPI SCLK 时钟分频、LCSFC 四线控制、RTC 控制以及管脚复用控制。

### 4.4.2 特点

- LCSFC SPI SCLK 时钟分频设置
- LCSFC 工作配置
- RTC 输出钳位配置
- 管脚复用

### 4.4.3 寄存器概述

寄存器表 3 系统控制寄存器概述（基地址：0x11000000）

偏移地址	名称	描述	页码
0x30	LCSFC_BAUDR	LCSFC SPI SCLK 时钟分频	
0x58	RTC_REQ_HOLD	RTC 输出钳位保持寄存器控制	
0x5c	QUAD_ENABLE	LCSFC SPI 四线控制寄存器	
0x34	IO_DIPLEX0	IO 复用选择寄存器 0	
0x38	IO_DIPLEX1	IO 复用选择寄存器 1	
0x3c	IO_DIPLEX2	IO 复用选择寄存器 2	
0x40	IOMGR_REN_REG0	IO 上下拉使能寄存器 0	
0x44	IOMGR_REN_REG1	IO 上下拉使能寄存器 1	
0x48	IOMGR_REN_REG2	IO 上下拉使能寄存器 2	
0x4c	IOMGR_IE_IE0	IO 输入使能寄存器 0	
0x50	IOMGR_IE_IE1	IO 输入使能寄存器 1	

0x54	IOMGR_IE_IE2	IO 输入使能寄存器 2	
------	--------------	--------------	--

#### 4.4.4 寄存器描述

##### 4.4.4.1 LCSFC\_BAUDR

偏移地址：0x30

复位方式：H

Bits	Access	Name	Default	Description
[31:0]	RW	lcsfc_baudr	0x4	LCSFC SPI SCLK时钟分频设置，默认值为4

##### 4.4.4.2 RTC\_REQ\_HOLD

偏移地址：0x58

复位方式：H

Bits	Access	Name	Default	Description
[31:1]	-	reserved		
[0]	RW	reg_rtc_hold	0x1	RTC输出钳位保持寄存器控制

##### 4.4.4.3 QUAD\_ENABLE

偏移地址：0x5c

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	RW	sys_sample_delay	0x0	四线数据线输入采样延时设置，范围0~8
[23:1]	-	reserved		
[0]	RW	quad_enable	0x0	LCSFC SPI四线控制使能，高有效

##### 4.4.4.4 IO\_DIPLEX0

偏移地址：0x34

复位方式：H

Bits	Access	Name	Default	Description
[31]	RW	ptb7_sel	0x1	PTB7复用选择 1: GPIO1_PTB7 0: AK_ADC_SDO
[30:20]	-	reserved		
[19]	RW	pwm2_sel	0x1	PWM_2复用选择 1: GPIO_PTA17 0: PWM_2
[18]	RW	pwm1_sel	0x1	PWM_1复用选择 1: GPIO_PTA16 0: PWM_1
[17]	RW	pwm0_sel	0x1	PWM_0复用选择 1: GPIO_PTA15 0: PWM_0

[16:15]	RW	uart3_txd_sel	0x3	UART3_TXD复用选择 3: GPIO_PTA14 2: 保留 1: 基带rf_rx_inter 0: UART3_TXD
[14:13]	RW	uart3_rxd_sel	0x3	UART3_RXD复用选择 3: GPIO_PTA13 2: 保留 1: 基带rf_tx_inter 0: UART3_RXD
[12]	RW	uart2_txd_sel	0x1	UART2_TXD复用选择 1: GPIO_PTA12 0: UART2_TXD
[11]	RW	uart2_rxd_sel	0x1	UART2_RXD复用选择 1: GPIO_PTA11 0: UART2_RXD
[10]	RW	uart1_txd_sel	0x1	UART1_TXD复用选择 1: GPIO_PTA10 0: UART1_TXD
[9]	RW	uart1_rxd_sel	0x1	UART1_RXD复用选择 1: GPIO_PTA9 0: UART1_RXD
[8]	RW	i2c1_sda_sel	0x1	I2C_SDA_1 复用选择 1: GPIO_PTA8 0: I2C_SDA_1
[7]	RW	i2c1_scl_sel	0x1	I2C_SCL_1复用选择 1: GPIO_PTA7 0: I2C_SCL_1
[6]	RW	i2c0_sda_sel	0x1	I2C_SDA_0 复用选择 1: GPIO_PTA6 0: I2C_SDA_0
[5]	RW	i2c0_scl_sel	0x1	I2C_SCL_0复用选择 1: GPIO_PTA5 0: I2C_SCL_0
[4]	RW	jtg_rstn_sel	0x0	JTG_RST_N复用选择 1: GPIO_PTA4 0: JTG_RST_N
[3]	RW	jtg_tdo_sel	0x0	JTG_TDO复用选择 1: GPIO_PTA3 0: JTG_TDO
[2]	RW	jtg_tdi_sel	0x0	JTG_TDI复用选择 1: GPIO_PTA2 0: JTG_TDI
[1]	RW	jtg_tck_sel	0x0	JTG_TCK复用选择

				1: GPIO_PTA1 0: JTG_TCK
[0]	RW	jtg_tms_sel	0x0	JTG_TMS复用选择 1: GPIO_PTA0 0: JTG_TMS

#### 4.4.4.5 IO\_DIPLEX1

偏移地址: 0x38

复位方式: H

Bits	Access	Name	Default	Description
[31]	RW	ptb6_sel	0x1	PTB6复用选择 1: GPIO1_PTB6 0: AK_ADC_SCLK
[30]	RW	ptb5_sel	0x1	PTB4复用选择 1: GPIO1_PTB5 0: AK_ADC_FS
[29]	RW	ptb4_sel	0x1	PTB4复用选择 1: GPIO1_PTB4 0: AK_DAC_SDI
[28]	RW	ptb3_sel	0x1	PTB3复用选择 1: GPIO1_PTB3 0: AK_DAC_SCLK
[27]	RW	ptb2_sel	0x1	PTB2复用选择 1: GPIO1_PTB2 0: AK_DAC_FS
[26:25]	RW	time_slot_tx_sel	0x3	TIME_SLOT_T_INTER复用选择 3: GPIO_PTA31 2: SDIO sdio_card_write_prt 1: 保留 0: TIME_SLOT_T_INTER
[24:23]	RW	time_slot_rx_sel	0x3	TIME_SLOT_R_INTER复用选择 3: GPIO_PTA30 2: SDIO sdio_card_detect_n 1: 保留 0: TIME_SLOT_R_INTER
[22:21]	RW	bs_inter_out_sel	0x3	BS_INTER_OUT复用选择 3: GPIO_PTA29 2: SDIO sdio_cdata[1] 1: 保留 0: BS_INTER_OUT
[20:19]	RW	bs_inter_in_sel	0x3	BS_INTER_IN复用选择 3: GPIO_PTA28 2: SDIO sdio_cdata[2]

				1: POR_RST_N 0: BS_INTER_IN
[18:17]	RW	spi1_miso_sel	0x3	SPI1_MISO复用选择 3: GPIO_PTA27 2: SDIO sdio_cdata[0] 1: Slave SPI1_MISO 0: Master SPI1_MISO
[16:15]	RW	spi1_mosi_sel	0x3	SPI1_MOSI复用选择 3: GPIO_PTA26 2: SDIO sdio_ccmd 1: Slave SPI1_MOSI 0: Master SPI1_MOSI
[14:13]	RW	spi1_sclk_sel	0x3	SPI1_SCLK复用选择 3: GPIO_PTA25 2: SDIO sdio_cclk_out 1: Slave SPI1_SCLK 0: Master SPI1_SCLK
[12:11]	RW	spi1_csn1_sel	0x3	SPI1_CSN_1复用选择 3: GPIO_PTA24 2: 保留 1: 基带sys_inter 0: Master SPI1_CSN_1
[10:9]	RW	spi1_csn0_sel	0x3	SPI1_CSN_0复用选择 3: GPIO_PTA23 2: SDIO sdio_cdata[3] 1: Slave SPI1_CSN_0 0: Master SPI1_CSN_0
[8:7]	RW	spi0_miso_sel	0x3	SPI0_MISO复用选择 3: GPIO_PTA22 2: 保留 1: Slave SPI0_MISO 0: Master SPI0_MISO
[6:5]	RW	spi0_mosi_sel	0x3	SPI0_MOSI复用选择 3: GPIO_PTA21 2: 保留 1: Slave SPI0_MOSI 0: Master SPI0_MOSI
[4:3]	RW	spi0_sclk_sel	0x3	SPI0_SCLK复用选择 3: GPIO_PTA20 2: 保留 1: Slave SPI0_SCLK 0: Master SPI0_SCLK
[2]	RW	spi0_csn1_sel	0x1	SPI0_CSN_1复用选择 1: GPIO_PTA19

				0: SPI0_CSN_1
[1:0]	RW	spi0_csn0_sel	0x3	SPI0_CSN_0复用选择 3: GPIO_PTA18 2: 保留 1: Slave SPI0_CSN_0 0: Master SPI0_CSN_0

#### 4.4.4.6 IO\_DIPLEX2

偏移地址: 0x3c

复位方式: H

Bits	Access	Name	Default	Description
[31:30]	-	reserved		
[29]	RW	dac_vout_mcuc_sel	0x1	DAC_VOUT_MCUC复用选择 1: GPIO_PTC25 0: DAC_VOUT_MCUC
[28]	RW	dac_vout_mcub_sel	0x1	DAC_VOUT_MCUB复用选择 1: GPIO_PTC24 0: DAC_VOUT_MCUB
[27]	RW	dac_vout_mcu_a_sel	0x1	DAC_VOUT_MCUA复用选择 1: GPIO_PTC23 0: DAC_VOUT_MCUA
[26]	RW	adc7_in_sel	0x1	ADC7_IN复用选择 1: GPIO_PTC22 0: ADC7_IN
[25]	RW	adc6_in_sel	0x1	ADC6_IN复用选择 1: GPIO_PTC21 0: ADC6_IN
[24]	RW	adc5_in_sel	0x1	ADC5_IN复用选择 1: GPIO_PTC20 0: ADC5_IN
[23]	RW	adc4_in_sel	0x1	ADC4_IN复用选择 1: GPIO_PTC19 0: ADC4_IN
[22]	RW	adc3_in_sel	0x1	ADC3_IN复用选择 1: GPIO_PTC18 0: ADC3_IN
[21]	RW	adc2_in_sel	0x1	ADC2_IN复用选择 1: GPIO_PTC17 0: ADC2_IN
[20]	RW	adc1_in_sel	0x1	ADC1_IN复用选择 1: GPIO_PTC16 0: ADC1_IN
[19]	RW	adc0_in_sel	0x1	ADC0_IN复用选择

				1: GPIO_PTC15 0: ADC0_IN
[18]	RW	lcd_db7_sel	0x1	LCD_DB7复用选择 1: GPIO_PTC24 0: LCD_DB7
[17]	RW	lcd_db6_sel	0x1	LCD_DB6复用选择 1: GPIO_PTC13 0: LCD_DB6
[16]	RW	lcd_db5_sel	0x1	LCD_DB5复用选择 1: GPIO_PTC12 0: LCD_DB5
[15]	RW	lcd_db4_sel	0x1	LCD_DB4复用选择 1: GPIO_PTC11 0: LCD_DB4
[14]	RW	lcd_db3_sel	0x1	LCD_DB3复用选择 1: GPIO_PTC10 0: LCD_DB3
[13]	RW	lcd_db2_sel	0x1	LCD_DB2复用选择 1: GPIO_PTC9 0: LCD_DB2
[12:11]	RW	lcd_db1_sel	0x3	LCD_DB1复用选择 3: GPIO_PTC8 2: Slave SPI2_MOSI 1: Master SPI2_MOSI 0: LCD_DB1
[10:9]	RW	lcd_db0_sel	0x3	LCD_DB0复用选择 3: GPIO_PTC7 2: Slave SPI2_MISO 1: Master SPI2_MISO 0: LCD_DB0
[8]	RW	lcd_rd_sel	0x1	LCD_RD复用选择 1: GPIO_PTC6 0: LCD_RD
[7]	RW	lcd_wr_sel	0x1	LCD_WR复用选择 1: GPIO_PTC5 0: LCD_WR
[6:5]	RW	lcd_rs_sel	0x3	LCD_RS复用选择 3: GPIO_PTC4 2: Slave SPI2_SCLK 1: Master SPI2_SCLK 0: LCD_RS
[4:3]	RW	lcd_cs_sel	0x3	LCD_CS复用选择 3: GPIO_PTC3 2: Slave SPI2_CS

				1: Master SPI2_CS 0: LCD_CS
[2]	-	reserved		
[1:0]	RW	clk_out_sel	0x3	CLK_OUT复用选择 3: GPIO_PTC0 2: RTC_CLK 1: PLLB CLK_OUT 0: PLLA CLK_OUT

#### 4.4.4.7 IOMGR\_REN\_REG0

偏移地址: 0x40

复位方式: H

Bits	Access	Name	Default	Description
[31]	RW	time_slot_rx_ren	0x1	TIME_SLOT_R_INTER下拉使能 1: 使能关闭 0: 使能有效
[30]	RW	time_slot_tx_ren	0x1	TIME_SLOT_T_INTER下拉使能 1: 使能关闭 0: 使能有效
[29]	RW	bs_inter_out_ren	0x1	BS_INTER_OUT上拉使能 1: 使能关闭 0: 使能有效
[28]	RW	bs_inter_in_ren	0x1	BS_INTER_IN上拉使能 1: 使能关闭 0: 使能有效
[27]	RW	spi1_miso_ren	0x1	SPI1_MISO上拉使能 1: 使能关闭 0: 使能有效
[26]	RW	spi1_mosi_ren	0x1	SPI1_MOSI上拉使能 1: 使能关闭 0: 使能有效
[25]	RW	spi1_sclk_ren	0x1	SPI1_SCLK下拉使能 1: 使能关闭 0: 使能有效
[24]	RW	spi1_csn1_ren	0x1	SPI1_CSN_1上拉使能 1: 使能关闭 0: 使能有效
[23]	RW	spi1_csn0_ren	0x1	SPI1_CSN_0上拉使能 1: 使能关闭 0: 使能有效
[22]	RW	spi0_miso_ren	0x1	SPI0_MISO下拉使能 1: 使能关闭 0: 使能有效

[21]	RW	spi0_mosi_ren	0x1	SPI0_MOSI上拉使能 1: 使能关闭 0: 使能有效
[20]	RW	spi0_sclk_ren	0x1	SPI0_SCLK下拉使能 1: 使能关闭 0: 使能有效
[19]	RW	spi0_csn1_ren	0x1	SPI0_CSN_1上拉使能 1: 使能关闭 0: 使能有效
[18]	RW	spi0_csn0_ren	0x1	SPI0_CSN_0上拉使能 1: 使能关闭 0: 使能有效
[17]	RW	pwm2_ren	0x1	PWM_2下拉使能 1: 使能关闭 0: 使能有效
[16]	RW	pwm1_ren	0x1	PWM_1上拉使能 1: 使能关闭 0: 使能有效
[15]	RW	pwm0_ren	0x1	PWM_0上拉使能 1: 使能关闭 0: 使能有效
[14]	RW	uart3_txd_ren	0x1	UART3_TXD上拉使能 1: 使能关闭 0: 使能有效
[13]	RW	uart3_rxd_ren	0x1	UART3_RXD上拉使能 1: 使能关闭 0: 使能有效
[12]	RW	uart2_txd_ren	0x1	UART2_TXD上拉使能 1: 使能关闭 0: 使能有效
[11]	RW	uart2_rxd_ren	0x1	UART2_RXD上拉使能 1: 使能关闭 0: 使能有效
[10]	RW	uart1_txd_ren	0x1	UART1_TXD上拉使能 1: 使能关闭 0: 使能有效
[9]	RW	uart1_rxd_ren	0x1	UART1_RXD上拉使能 1: 使能关闭 0: 使能有效
[8]	RW	i2c1_sda_ren	0x1	I2C_SDA_1上拉使能 1: 使能关闭 0: 使能有效
[7]	RW	i2c1_scl_ren	0x1	I2C_SCL_1上拉使能

				1: 使能关闭 1: 使能有效
[6]	RW	i2c0_sda_ren	0x1	I2C_SDA_0上拉使能 1: 使能关闭 2: 使能有效
[5]	RW	i2c0_scl_ren	0x1	I2C_SCL_0上拉使能 1: 使能关闭 0: 使能有效
[4]	RW	jtg_rstn_ren	0x1	JTG_RST_N上拉使能 1: 使能关闭 0: 使能有效
[3]	RW	jtg_tdo_ren	0x1	JTG_TDO下拉使能 1: 使能关闭 0: 使能有效
[2]	RW	jtg_tdi_ren	0x1	JTG_TDI下拉使能 1: 使能关闭 0: 使能有效
[1]	RW	jtg_tck_ren	0x1	JTG_TCK上拉使能 1: 使能关闭 0: 使能有效
[0]	RW	jtg_tms_ren	0x1	JTG_TMS上拉使能 1: 使能关闭 0: 使能有效

#### 4.4.4.8 IOMGR\_REN\_REG1

偏移地址: 0x44

复位方式: H

Bits	Access	Name	Default	Description
[31]	RW	uart0_txd_ren	0x1	UART0_TXD上拉使能 1: 使能关闭 0: 使能有效
[30]	RW	uart0_rxd_ren	0x1	UART0_RXD上拉使能 1: 使能关闭 0: 使能有效
[29]	RW	nfc_miso_ren	0x1	NFC_MISO上拉使能 1: 使能关闭 0: 使能有效
[28]	RW	nfc_mosi_ren	0x1	NFC_MOSI上拉使能 1: 使能关闭 0: 使能有效
[27]	RW	nfc_sclk_ren	0x1	NFC_SCLK上拉使能 1: 使能关闭 0: 使能有效

[26]	RW	nfc_csn_ren	0x1	NFC_CSN上拉使能 1: 使能关闭 0: 使能有效
[25]	RW	nfc_wp_ren	0x1	NFC_WP上拉使能 1: 使能关闭 0: 使能有效
[24]	RW	nfc_hold_ren	0x1	NFC_HOLD上拉使能 1: 使能关闭 0: 使能有效
[23]	RW	dac_vout_mcuc_ren	0x1	DAC_VOUT_MCUC 上拉使能 1: 使能关闭 0: 使能有效
[22]	RW	dac_vout_mcub_ren	0x1	DAC_VOUT_MCUB 上拉使能 1: 使能关闭 0: 使能有效
[21]	RW	dac_vout_mcu_a_ren	0x1	DAC_VOUT_MCUA 上拉使能 1: 使能关闭 0: 使能有效
[20]	RW	adc7_in_ren	0x1	ADC7_IN 下拉使能 1: 使能关闭 0: 使能有效
[19]	RW	adc6_in_ren	0x1	ADC6_IN 下拉使能 1: 使能关闭 0: 使能有效
[18]	RW	adc5_in_ren	0x1	ADC5_IN下拉使能 1: 使能关闭 0: 使能有效
[17]	RW	adc4_in_ren	0x1	ADC4_IN下拉使能 1: 使能关闭 0: 使能有效
[16]	RW	adc3_in_ren	0x1	ADC3_IN下拉使能 1: 使能关闭 0: 使能有效
[15]	RW	adc2_in_ren	0x1	ADC2_IN下拉使能 1: 使能关闭 0: 使能有效
[14]	RW	adc1_in_ren	0x1	ADC1_IN下拉使能 1: 使能关闭 0: 使能有效
[13]	RW	adc0_in_ren	0x1	ADC0_IN下拉使能 1: 使能关闭 0: 使能有效
[12]	RW	lcd_db7_ren	0x1	LCD_DB7下拉使能

				1: 使能关闭 0: 使能有效
[11]	RW	lcd_db6_ren	0x1	LCD_DB6 上拉使能 1: 使能关闭 0: 使能有效
[10]	RW	lcd_db5_ren	0x1	LCD_DB5 上拉使能 1: 使能关闭 0: 使能有效
[9]	RW	lcd_db4_ren	0x1	LCD_DB4 上拉使能 1: 使能关闭 0: 使能有效
[8]	RW	lcd_db3_ren	0x1	LCD_DB3 上拉使能 1: 使能关闭 0: 使能有效
[7]	RW	lcd_db2_ren	0x1	LCD_DB2 下拉使能 1: 使能关闭 0: 使能有效
[6]	RW	lcd_db1_ren	0x1	LCD_DB1 下拉使能 1: 使能关闭 0: 使能有效
[5]	RW	lcd_db0_ren	0x1	LCD_DB0 下拉使能 1: 使能关闭 0: 使能有效
[4]	RW	lcd_rd_ren	0x1	LCD_RD上拉使能 1: 使能关闭 0: 使能有效
[3]	RW	lcd_wr_ren	0x1	LCD_WR 上拉使能 1: 使能关闭 0: 使能有效
[2]	RW	lcd_rs_ren	0x1	LCD_RS上拉使能 1: 使能关闭 0: 使能有效
[1]	RW	lcd_cs_ren	0x1	LCD_CS上拉使能 1: 使能关闭 0: 使能有效
[0]	RW	lcd_nreset_ren	0x1	LCD_NRESET上拉使能 1: 使能关闭 0: 使能有效

#### 4.4.4.9 IOMGR\_REN\_REG2

偏移地址: 0x48

复位方式: H

Bits	Access	Name	Default	Description
------	--------	------	---------	-------------

[31]	-	reserved		
[30]	RW	clk_out_ren	0x1	CLK_OUT 下拉使能 1: 使能关闭 0: 使能有效
[29]	RW	ptb29_ren	0x1	PTB29下拉使能 1: 使能关闭 0: 使能有效
[28]	RW	ptb28_ren	0x1	PTB28上拉使能 1: 使能关闭 0: 使能有效
[27]	RW	ptb27_ren	0x1	PTB27上拉使能 1: 使能关闭 0: 使能有效
[26]	RW	ptb26_ren	0x1	PTB26上拉使能 1: 使能关闭 0: 使能有效
[25]	RW	ptb25_ren	0x1	PTB25上拉使能 1: 使能关闭 0: 使能有效
[24]	RW	ptb24_ren	0x1	PTB24下拉使能 1: 使能关闭 0: 使能有效
[23]	RW	ptb23_ren	0x1	PTB23下拉使能 1: 使能关闭 0: 使能有效
[22]	RW	ptb22_ren	0x1	PTB22上拉使能 1: 使能关闭 0: 使能有效
[21]	RW	ptb21_ren	0x1	PTB21上拉使能 1: 使能关闭 0: 使能有效
[20]	RW	ptb20_ren	0x1	PTB20上拉使能 1: 使能关闭 0: 使能有效
[19]	RW	ptb19_ren	0x1	PTB19上拉使能 1: 使能关闭 0: 使能有效
[18]	RW	ptb18_ren	0x1	PTB18下拉使能 1: 使能关闭 0: 使能有效
[17]	RW	ptb17_ren	0x1	PTB17下拉使能 1: 使能关闭 0: 使能有效

[16]	RW	ptb16_ren	0x1	PTB16上拉使能 1: 使能关闭 0: 使能有效
[15]	RW	ptb15_ren	0x1	PTB15上拉使能 1: 使能关闭 0: 使能有效
[14]	RW	ptb14_ren	0x1	PTB14上拉使能 1: 使能关闭 0: 使能有效
[13]	RW	ptb13_ren	0x1	PTB13上拉使能 1: 使能关闭 0: 使能有效
[12]	RW	ptb12_ren	0x1	PTB12下拉使能 1: 使能关闭 0: 使能有效
[11]	RW	ptb11_ren	0x1	PTB11下拉使能 1: 使能关闭 0: 使能有效
[10]	RW	ptb10_ren	0x1	PTB10上拉使能 1: 使能关闭 0: 使能有效
[9]	RW	ptb9_ren	0x1	PTB9上拉使能 1: 使能关闭 0: 使能有效
[8]	RW	ptb8_ren	0x1	PTB8上拉使能 1: 使能关闭 0: 使能有效
[7]	RW	ptb7_ren	0x1	PTB7上拉使能 1: 使能关闭 0: 使能有效
[6]	RW	ptb6_ren	0x1	PTB6下拉使能 1: 使能关闭 0: 使能有效
[5]	RW	ptb5_ren	0x1	PTB5下拉使能 1: 使能关闭 0: 使能有效
[4]	RW	ptb4_ren	0x1	PTB4下拉使能 1: 使能关闭 0: 使能有效
[3]	RW	ptb3_ren	0x1	PTB3上拉使能 1: 使能关闭 0: 使能有效
[2]	RW	ptb2_ren	0x1	PTB2上拉使能

				1: 使能关闭 0: 使能有效
[1]	RW	ptb1_ren	0x1	PTB1上拉使能 1: 使能关闭 0: 使能有效
[0]	RW	ptb0_ren	0x1	PTB0上拉使能 1: 使能关闭 0: 使能有效

#### 4.4.4.10 IOMGR\_IE\_IE0

偏移地址: 0x4c

复位方式: H

Bits	Access	Name	Default	Description
[31]	RW	time_slot_rx_ie	0x1	TIME_SLOT_R_INTER输入使能 1: 使能有效 0: 使能关闭
[30]	RW	time_slot_tx_ie	0x1	TIME_SLOT_T_INTER输入使能 1: 使能有效 0: 使能关闭
[29]	RW	bs_inter_out_ie	0x1	BS_INTER_OUT输入使能 1: 使能有效 0: 使能关闭
[28]	RW	bs_inter_in_ie	0x1	BS_INTER_IN输入使能 1: 使能有效 0: 使能关闭
[27]	RW	spi1_miso_ie	0x1	SPI1_MISO输入使能 1: 使能有效 0: 使能关闭
[26]	RW	spi1_mosi_ie	0x1	SPI1_MOSI输入使能 1: 使能有效 0: 使能关闭
[25]	RW	spi1_sclk_ie	0x1	SPI1_SCLK输入使能 1: 使能有效 0: 使能关闭
[24]	RW	spi1_csn1_ie	0x1	SPI1_CSN_1输入使能 1: 使能有效 0: 使能关闭
[23]	RW	spi1_csn0_ie	0x1	SPI1_CSN_0输入使能 1: 使能有效 0: 使能关闭
[22]	RW	spi0_miso_ie	0x1	SPI0_MISO输入使能 1: 使能有效 0: 使能关闭

[21]	RW	spi0_mosi_ie	0x1	SPI0_MOSI输入使能 1: 使能有效 0: 使能关闭
[20]	RW	spi0_sclk_ie	0x1	SPI0_SCLK输入使能 1: 使能有效 0: 使能关闭
[19]	RW	spi0_csn1_ie	0x1	SPI0_CSN_1输入使能 1: 使能有效 0: 使能关闭
[18]	RW	spi0_csn0_ie	0x1	SPI0_CSN_0输入使能 1: 使能有效 0: 使能关闭
[17]	RW	pwm2_ie	0x1	PWM_2输入使能 1: 使能有效 0: 使能关闭
[16]	RW	pwm1_ie	0x1	PWM_1输入使能 1: 使能有效 0: 使能关闭
[15]	RW	pwm0_ie	0x1	PWM_0输入使能 1: 使能有效 0: 使能关闭
[14]	RW	uart3_txd_ie	0x1	UART3_TXD输入使能 1: 使能有效 0: 使能关闭
[13]	RW	uart3_rxd_ie	0x1	UART3_RXD输入使能 1: 使能有效 0: 使能关闭
[12]	RW	uart2_txd_ie	0x1	UART2_TXD输入使能 1: 使能有效 0: 使能关闭
[11]	RW	uart2_rxd_ie	0x1	UART2_RXD输入使能 1: 使能有效 0: 使能关闭
[10]	RW	uart1_txd_ie	0x1	UART1_TXD输入使能 1: 使能有效 0: 使能关闭
[9]	RW	uart1_rxd_ie	0x1	UART1_RXD输入使能 1: 使能有效 0: 使能关闭
[8]	RW	i2c1_sda_ie	0x1	I2C_SDA_1输入使能 1: 使能有效 0: 使能关闭
[7]	RW	i2c1_scl_ie	0x1	I2C_SCL_1输入使能

				1: 使能有效 0: 使能关闭
[6]	RW	i2c0_sda_ie	0x1	I2C_SDA_0输入使能 1: 使能有效 0: 使能关闭
[5]	RW	i2c0_scl_ie	0x1	I2C_SCL_0输入使能 1: 使能有效 0: 使能关闭
[4]	RW	jtg_rstn_ie	0x1	JTG_RST_N输入使能 1: 使能有效 0: 使能关闭
[3]	RW	jtg_tdo_ie	0x1	JTG_TDO输入使能 1: 使能有效 0: 使能关闭
[2]	RW	jtg_tdi_ie	0x1	JTG_TDI输入使能 1: 使能有效 0: 使能关闭
[1]	RW	jtg_tck_ie	0x1	JTG_TCK输入使能 1: 使能有效 0: 使能关闭
[0]	RW	jtg_tms_ie	0x1	JTG_TMS输入使能 1: 使能有效 0: 使能关闭

#### 4.4.4.11 IOMGR\_IE\_IE1

偏移地址: 0x50

复位方式: H

Bits	Access	Name	Default	Description
[31]	RW	uart0_txd_ie	0x1	UART0_TXD输入使能 1: 使能有效 0: 使能关闭
[30]	RW	uart0_rxd_ie	0x1	UART0_RXD输入使能 1: 使能有效 0: 使能关闭
[29]	RW	nfc_miso_ie	0x1	NFC_MISO输入使能 1: 使能有效 0: 使能关闭
[28]	RW	nfc_mosi_ie	0x1	NFC_MOSI输入使能 1: 使能有效 0: 使能关闭
[27]	RW	nfc_sclk_ie	0x1	NFC_SCLK输入使能 1: 使能有效 0: 使能关闭

[26]	RW	nfc_csn_ie	0x1	NFC_CSN输入使能 1: 使能有效 0: 使能关闭
[25]	RW	nfc_wp_ie	0x1	NFC_WP输入使能 1: 使能有效 0: 使能关闭
[24]	RW	nfc_hold_ie	0x1	NFC_HOLD输入使能 1: 使能有效 0: 使能关闭
[23]	RW	dac_vout_mcuc_ie	0x1	DAC_VOUT_MCUC输入使能 1: 使能有效 0: 使能关闭
[22]	RW	dac_vout_mcub_ie	0x1	DAC_VOUT_MCUB输入使能 1: 使能有效 0: 使能关闭
[21]	RW	dac_vout_mcu_a_ie	0x1	DAC_VOUT_MCUA输入使能 1: 使能有效 0: 使能关闭
[20]	RW	adc7_in_ie	0x1	ADC7_IN输入使能 1: 使能有效 0: 使能关闭
[19]	RW	adc6_in_ie	0x1	ADC6_IN输入使能 1: 使能有效 0: 使能关闭
[18]	RW	adc5_in_ie	0x1	ADC5_IN输入使能 1: 使能有效 0: 使能关闭
[17]	RW	adc4_in_ie	0x1	ADC4_IN输入使能 1: 使能有效 0: 使能关闭
[16]	RW	adc3_in_ie	0x1	ADC3_IN输入使能 1: 使能有效 0: 使能关闭
[15]	RW	adc2_in_ie	0x1	ADC2_IN输入使能 1: 使能有效 0: 使能关闭
[14]	RW	adc1_in_ie	0x1	ADC1_IN输入使能 1: 使能有效 0: 使能关闭
[13]	RW	adc0_in_ie	0x1	ADC0_IN输入使能 1: 使能有效 0: 使能关闭
[12]	RW	lcd_db7_ie	0x1	LCD_DB7输入使能

				1: 使能有效 0: 使能关闭
[11]	RW	lcd_db6_ie	0x1	LCD_DB6输入使能 1: 使能有效 0: 使能关闭
[10]	RW	lcd_db5_ie	0x1	LCD_DB5输入使能 1: 使能有效 0: 使能关闭
[9]	RW	lcd_db4_ie	0x1	LCD_DB4输入使能 1: 使能有效 0: 使能关闭
[8]	RW	lcd_db3_ie	0x1	LCD_DB3输入使能 1: 使能有效 0: 使能关闭
[7]	RW	lcd_db2_ie	0x1	LCD_DB2输入使能 1: 使能有效 0: 使能关闭
[6]	RW	lcd_db1_ie	0x1	LCD_DB1输入使能 1: 使能有效 0: 使能关闭
[5]	RW	lcd_db0_ie	0x1	LCD_DB0输入使能 1: 使能有效 0: 使能关闭
[4]	RW	lcd_rd_ie	0x1	LCD_RD输入使能 1: 使能有效 0: 使能关闭
[3]	RW	lcd_wr_ie	0x1	LCD_WR输入使能 1: 使能有效 0: 使能关闭
[2]	RW	lcd_rs_ie	0x1	LCD_RS输入使能 1: 使能有效 0: 使能关闭
[1]	RW	lcd_cs_ie	0x1	LCD_CS输入使能 1: 使能有效 0: 使能关闭
[0]	RW	lcd_nreset_ie	0x1	LCD_NRESET输入使能 1: 使能有效 0: 使能关闭

#### 4.4.4.12 IOMGR\_IE\_IE2

偏移地址: 0x54

复位方式: H

Bits	Access	Name	Default	Description
------	--------	------	---------	-------------

[31]	-	reserved		
[30]	RW	clk_out_ie	0x1	CLK_OUT输入使能 1: 使能有效 0: 使能关闭
[29]	RW	ptb29_ie	0x1	PTB29输入使能 1: 使能有效 0: 使能关闭
[28]	RW	ptb28_ie	0x1	PTB28输入使能 1: 使能有效 0: 使能关闭
[27]	RW	ptb27_ie	0x1	PTB27输入使能 1: 使能有效 0: 使能关闭
[26]	RW	ptb26_ie	0x1	PTB26输入使能 1: 使能有效 0: 使能关闭
[25]	RW	ptb25_ie	0x1	PTB25输入使能 1: 使能有效 0: 使能关闭
[24]	RW	ptb24_ie	0x1	PTB24输入使能 1: 使能有效 0: 使能关闭
[23]	RW	ptb23_ie	0x1	PTB23输入使能 1: 使能有效 0: 使能关闭
[22]	RW	ptb22_ie	0x1	PTB22输入使能 1: 使能有效 0: 使能关闭
[21]	RW	ptb21_ie	0x1	PTB21输入使能 1: 使能有效 0: 使能关闭
[20]	RW	ptb20_ie	0x1	PTB20输入使能 1: 使能有效 0: 使能关闭
[19]	RW	ptb19_ie	0x1	PTB19输入使能 1: 使能有效 0: 使能关闭
[18]	RW	ptb18_ie	0x1	PTB18输入使能 1: 使能有效 0: 使能关闭
[17]	RW	ptb17_ie	0x1	PTB17输入使能 1: 使能有效 0: 使能关闭

[16]	RW	ptb16_ie	0x1	PTB16输入使能 1: 使能有效 0: 使能关闭
[15]	RW	ptb15_ie	0x1	PTB15输入使能 1: 使能有效 0: 使能关闭
[14]	RW	ptb14_ie	0x1	PTB14输入使能 1: 使能有效 0: 使能关闭
[13]	RW	ptb13_ie	0x1	PTB13输入使能 1: 使能有效 0: 使能关闭
[12]	RW	ptb12_ie	0x1	PTB12输入使能 1: 使能有效 0: 使能关闭
[11]	RW	ptb11_ie	0x1	PTB11输入使能 1: 使能有效 0: 使能关闭
[10]	RW	ptb10_ie	0x1	PTB10输入使能 1: 使能有效 0: 使能关闭
[9]	RW	ptb9_ie	0x1	PTB9输入使能 1: 使能有效 0: 使能关闭
[8]	RW	ptb8_ie	0x1	PTB8输入使能 1: 使能有效 0: 使能关闭
[7]	RW	ptb7_ie	0x1	PTB7输入使能 1: 使能有效 0: 使能关闭
[6]	RW	ptb6_ie	0x1	PTB6输入使能 1: 使能有效 0: 使能关闭
[5]	RW	ptb5_ie	0x1	PTB5输入使能 1: 使能有效 0: 使能关闭
[4]	RW	ptb4_ie	0x1	PTB4输入使能 1: 使能有效 0: 使能关闭
[3]	RW	ptb3_ie	0x1	PTB3输入使能 1: 使能有效 0: 使能关闭
[2]	RW	ptb2_ie	0x1	PTB2输入使能

				1: 使能有效 0: 使能关闭
[1]	RW	ptb1_ie	0x1	PTB1输入使能 1: 使能有效 0: 使能关闭
[0]	RW	ptb0_ie	0x1	PTB0输入使能 1: 使能有效 0: 使能关闭

## 4.5 处理器子系统

### 4.5.1 概述

HR\_C7000 处理器采用中天 CK803S, 32 位地址与数据通路, AHB 接口, 工作频率最高为 192M。CK803S 是面向控制领域的 32 位高效能嵌入式 CPU 核, 具有低成本、低功耗、高代码密度的特点。CK803S 采用 16/32 位混合编码指令系统, 具有精简高效的 3 级流水线。同时系统集成片上 RAM 供 CPU 进行使用, SRAM 容量为 288KB。整体框架如下图:

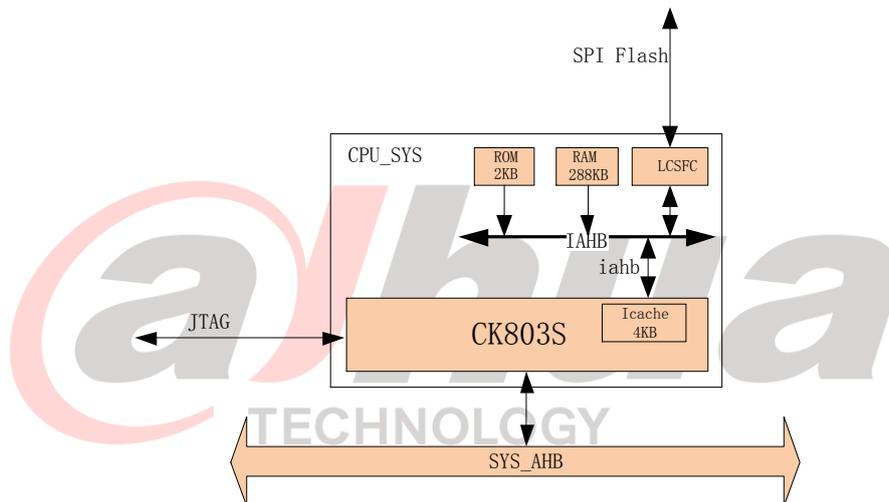


图 17 CPU 系统整体框图

### 4.5.2 特点

- 精简指令集处理器架构 (RISC)
- 32 位数据、16/32 位混合编码指令
- 16 个 32 位通用寄存器
- 集成 4KB 指令 Cache
- 3 级流水线
- 按序发射、按序执行、按序退休
- 可配置的多总线接口
- 支持多种处理器时钟和系统时钟比

### 4.5.3 地址映射关系

HR\_C7000 地址映射关系如下表:

表 26 地址映射关系

起始地址	结束地址	功能	大小	说明
0x0000_0000	0x0000_FFFF	BOOTROM 存储空间	2KB	
0x0001_0000	0x0005_7FFF	IRAM 存储空间	288KB	
0x0300_0000	0x03FF_FFFF	LCSFC 地址空间	16MB	Flash 运行空间
0x1100_0000	0x1100_FFFF	Modem	64KB	
0x1200_0000	0x1200_FFFF	i8080	64KB	
0x1300_0000	0x1303_FFFF	USB	256KB	
0x1400_0000	0x1400_FFFF	TIMER	64KB	
0x1401_0000	0x1401_FFFF	WDG	64KB	
0x1402_0000	0x1402_FFFF	GPIOA	64KB	
0x1403_0000	0x1403_FFFF	UART0	64KB	
0x1404_0000	0x1404_FFFF	UART1	64KB	
0x1405_0000	0x1405_FFFF	UART2	64KB	
0x1406_0000	0x1406_FFFF	I2C0	64KB	
0x1407_0000	0x1407_FFFF	I2C1	64KB	
0x1408_0000	0x1408_FFFF	I2C2	64KB	内部 RTC 专用
0x1409_0000	0x1409_FFFF	UART3	64KB	
0x140A_0000	0x140A_FFFF	SPI Master 0	64KB	
0x140B_0000	0x140B_FFFF	SPI Master 1	64KB	
0x140C_0000	0x140C_FFFF	PWM	64KB	
0x140D_0000	0x140D_FFFF	ADC	64KB	
0x140E_0000	0x140E_FFFF	SPI2	64KB	
0x140F_0000	0x140F_FFFF	DAC	64KB	
0x1410_0000	0x1410_FFFF	GPIOB	64KB	
0x1411_0000	0x1411_FFFF	GPIOC	64KB	
0x1412_0000	0x1412_FFFF	SPI Slave 0	64KB	
0x1413_0000	0x1413_FFFF	SPI Slave 1	64KB	
0x1414_0000	0x1414_FFFF	SPI Slave 2	64KB	
0x1415_0000	0x1415_FFFF	SPI Master 3	64KB	EFUSE 专用
0x1500_0000	0x1500_FFFF	SDIO	64KB	
0x1600_0000	0x1600_FFFF	Modem_Buffer	64KB	
0x1700_0000	0x1700_FFFF	PIC	64KB	
0x1800_0000	0x1800_7FFF	SRAM	32KB	SAHB 上的 SRAM

### 4.5.4 启动

正常工作模式下，CPU 支持两种系统启动方式：

- 串口调试模式：复位后，CPU 判断是否有程序从串口载入，若有则将程序载入内存，否则等待直到超时而进入 Flash 启动模式。

- Flash 启动：用户只需要将预定程序烧到 Flash 中，复位后且启动程序进入到 Flash 模式后，CPU 验证 Flash 程序的有效性。在发现 Flash 中的有效程序后，CPU 会通过 LCSFC 从 Flash 取指执行。

系统启动流程图如下图所示：

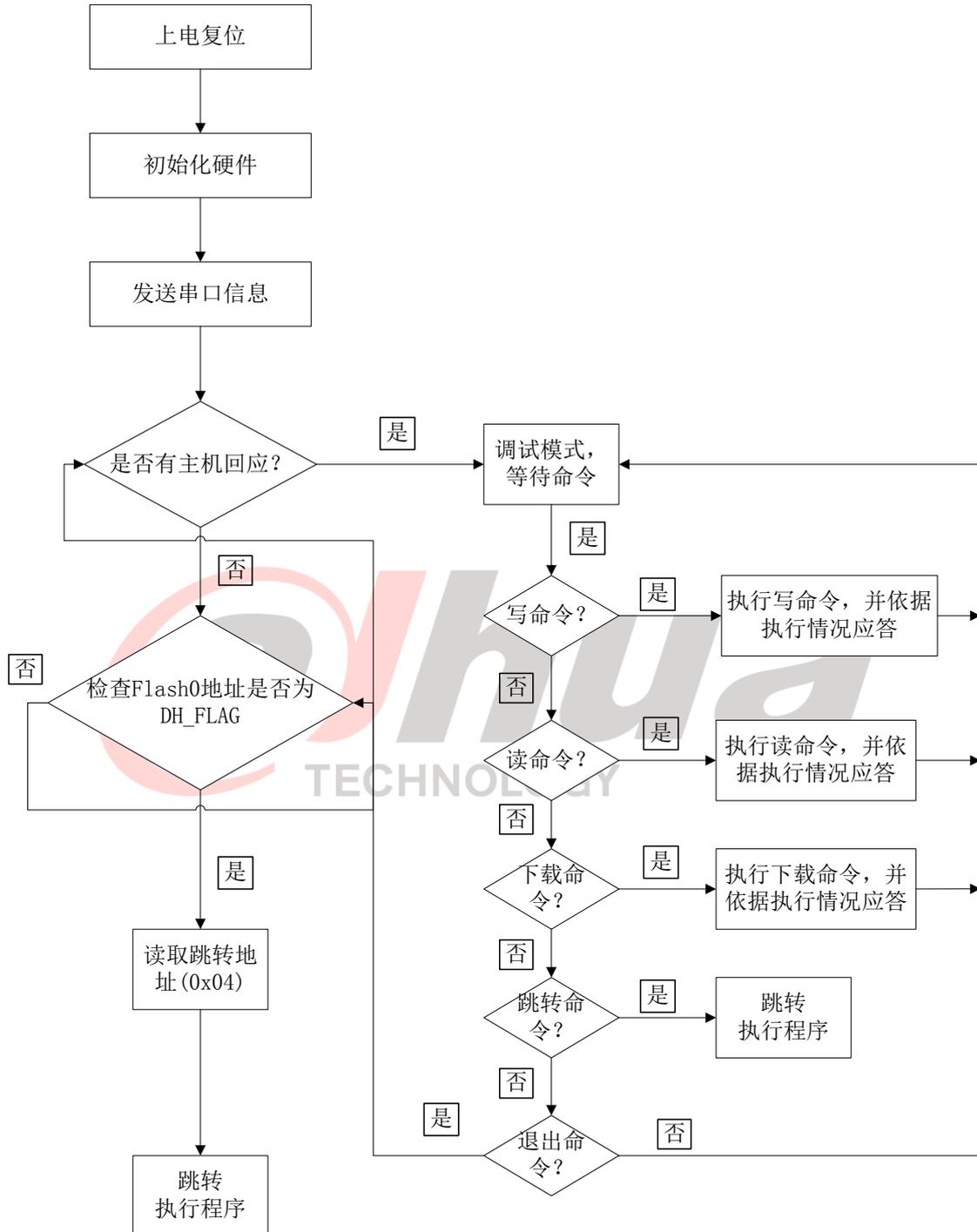


图 18 系统启动流程图

#### 4.5.5 JTAG 调试

CK803S CPU 支持 JTAG 调试。CK803S JTAG 调试接口的主要特性如下：

- 使用标准的 JTAG 协议进行调试
- 非侵入式获取 CPU 状态
- 支持软断点、8 个硬件中断
- 可以设置多个内存断点
- 检查和设置 CPU 寄存器的值
- 检查和改变内存的值
- 可进行指令单步执行或多步执行
- 快速下载程序
- 可在 CPU 复位之后或在普通用户模式下进入调试模式

CK803S 的调试工作是在调试软件、调试代理服务程序、调试器和调试接口的配合下完成的。调试接口在整个调试环境中的位置如下图所示。其中，调试软件和调试代理服务程序通过网络互联，调试代理服务程序和调试器通过 USB 连接，调试器与 CPU 的调试接口通过 JTAG 连接。

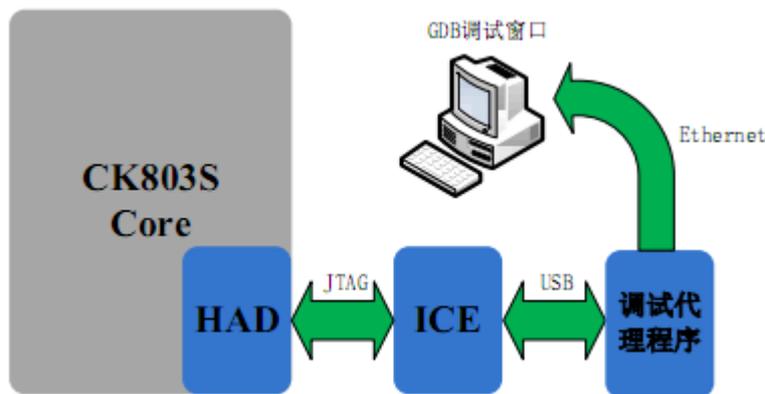


图 19 CK803 JTAG 调试

## 4.6 基带通信系统

### 4.6.1 概述

基带通信系统主要涉及基带工作模式、RF 模式以及模拟相关模块的设置。

### 4.6.2 特点

- DMR 物理层模式
- DMR 二层直通模式
- DMR 二层中继模式
- DMR 三层直通模式
- DMR 三层中继模式
- FM 模式
- DMR 二层 DCDM 模式
- DMR/FM 数模同检模式
- 支持多个 RF 接口模式
- 模拟模块 DAC、ADC、Code 控制设置

表 27 基带工作模式列表

序	工作模式	寄存器名称	寄存器配置值	描述
---	------	-------	--------	----

号				
1	DMR 物理层模式	WORK_MODE	0x22	时隙模式
			0x02	连续模式
2	DMR 二层直通模式	WORK_MODE	0x6a 或者 0x6b	
3	DMR 二层中继模式	WORK_MODE	0x6e	时隙 2
			0x6f	时隙 1
4	DMR 三层直通模式	WORK_MODE	0x73	
5	DMR 三层中继模式	WORK_MODE	0x76	时隙 2
			0x77	时隙 1
6	FM 模式	WORK_MODE	0x80	
7	DMR 二层 DCDM 模式	WORK_MODE	0x6a	TDMA2 时隙
			0x6b	TDMA1 时隙
8	DMR/FM 数模同检模式	WORK_MODE	0x16a 或 0x16b	二层模式数模同检

#### 4.6.3 中断向量

如下表定义了基带通信系统的中断向量列表。

表 28 基带通信系统中断列表

序号	名称	类型	描述
1	RF_TX_INTERP	电平触发, 高有效	射频切换中断, RF 发送开始中断, 通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
2	RF_RX_INTERP	电平触发, 高有效	射频切换中断, RF 接收开始中断, 通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
3	SYS_INTERP	电平触发, 高有效	基带系统中断, 有子中断见下表, 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除
4	TIME_SLOT_TX_INTERP	电平触发, 高有效	基带数字 30ms 发送时隙中断, 通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
5	TIME_SLOT_RX_INTERP	电平触发, 高有效	基带数字 30ms 接收时隙中断, 通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
6	PCM_WR_INTERP	电平触发, 高有效	Codec 音频数据流写入中断, 通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
7	PCM_RD_INTERP	电平触发, 高有效	Codec 音频数据流读取中断, 通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除

如下表定义了基带系统中断(SYS\_INTERP)的系统子中断向量列表。

表 29 基带通信系统子中断列表

序号	名称	类型	描述
1	DLL_RDY_INTERP_TX	电平触发, 高有效	二层发送处理中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽;

			通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
2	DLL_RDY_INTERP_RX	电平触发, 高有效	二层接收处理中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
3	DLL_LATELC_RX_INTERP	电平触发, 高有效	二层语音后接入中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEA 寄存器对应 bit 位写 1 清除。
4	DLL_FRAME_RX_INTERP	电平触发, 高有效	二层各类型帧信息接收中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEA 寄存器对应 bit 位写 1 清除。
5	DLL_SHORTLC_RX_INTERP	电平触发, 高有效	二层 shortlc 信息接收中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEA 寄存器对应 bit 位写 1 清除。
6	DLL_TX_SLOT_ABNORMAL	电平触发, 高有效	二层发送时隙异常中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEA 寄存器对应 bit 位写 1 清除。
7	DLL_RX_SLOT_ABNORMAL	电平触发, 高有效	二层接收时隙异常中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEA 寄存器对应 bit 位写 1 清除。
8	PHY_RECV_ABNORMAL	电平触发, 高有效	物理层接收时隙边界异常提示中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEA 寄存器对应 bit 位写 1 清除。
9	PHY_RDY_INTERP	电平触发, 高有效	物理层处理中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
10	PHY_RX_INTERP	电平触发, 高有效	物理层接收解帧中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit

			位屏蔽： 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
11	FM_TX_INTERP	电平触发，高有效	FM 发送数据处理中断： 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽： 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
12	FM_RX_INTERP	电平触发，高有效	FM 接收数据处理中断： 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽： 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
13	FM_SIG_RX_INTERP	电平触发，高有效	模拟功能接收检测中断： 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽： 通过 SYS_INTERP_CLEA 寄存器对应 bit 位写 1 清除。
14	RDY_NEXT_SHORT_LC	电平触发，高有效	中转台发送 CACH 中的 shortlc 信息准备中断 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽： 通过 SYS_INTERP_CLEA 寄存器对应 bit 位写 1 清除。

#### 4.6.4 寄存器概述

寄存器表 4 基带系统控制寄存器概述（基地址：0x11000000）

偏移地址	名称	描述	页码
0x70	DAC_CONTROL	基带 DAC 控制寄存器	
0x74	ADC_CONTROL	基带 ADC 控制寄存器	
0x80	AUDIO_CONTROL	基带音频通路控制寄存器	
0x84	AUDIO_BUFFER_CLR	基带音频通路缓存清除控制寄存器	
0x88	LINEOUT_CTRL	Codec Lineout 输出控制寄存器	
0x8c	CODEC_I2C_MUX	CodecI2C 控制寄存器	
0x100	WORK_MODE	基带工作模式设置	
0x104	RF_MODE	基带 RF 接口模式设置	
0x398	SYS_INTERP_LIST	系统中断向量列表	
0x39c	SYS_INTERP_MASK	系统中断向量掩码	
0x3a0	SYS_INTERP_CLEAR	系统中断向量清除	
0x3a4	LAYER3_INTERP_LIST	三层中断向量列表	
0x3a8	LAYER3_INTERP_MASK	三层中断向量掩码	

0x3ac	LAYER3_INTERP_CLEAR	三层中断向量清除	
0x3b0	INTERP_CLEAR	30ms 中断、rf 中断、pcm 中断清除	

#### 4.6.5 寄存器描述

##### 4.6.5.1 DAC\_CONTROL

偏移地址：0x70

复位方式：H

Bits	Access	Name	Default	Description
[31]	RW	dac_data_ctrl	0x1	基带调制输出的DAC数据方式： 0：补码方式；1：偏移码方式。
[30:6]	-	reserved		
[5]	RW	pwda	0x1	A路DAC电源控制， 0：power-on； 1：power-down
[4]	RW	pwdb	0x1	B路DAC电源控制， 0：power-on； 1：power-down
[3]	-	reserved		
[2]	RW	lpmodea	0x1	A路DAC低功耗模式， 0：关闭低功耗模式； 1：开启低功耗模式
[1]	RW	lpmodeb	0x1	B路DAC低功耗模式， 0：关闭低功耗模式； 1：开启低功耗模式
[0]	-	reserved		

##### 4.6.5.2 ADC\_CONTROL

偏移地址：0x74

复位方式：H

Bits	Access	Name	Default	Description
[31:30]	RO	adc_testout	0x0	测试输出状态
[29:28]	-	reserved		
[27:24]	RW	adc_testin	0x0	测试输入控制
[23:15]	-	reserved		
[14]	RW	adc_data_ctrl	0x1	ADC数据方式： 0：补码方式；1：偏移码方式。
[13:10]	RW	buffer_trim	0x0	
[9]	RW	intv_mode	0x0	
[8]	RW	enadc0	0x0	I路ADC功能使能。ADC I路输出数据在enadc0为高电平并持续唤醒时间之后有效（通常为120个工作时钟周期）。当enadc0设置为0时，ADC I路输出数据将

				被保持在最后一次状态值。
[7]	RW	enadc1	0x0	Q路ADC功能使能。ADC Q路输出数据在enadc1为高电平并持续唤醒时间之后有效（通常为120个工作时钟周期）。当enadc1设置为0时，ADC Q路输出数据将被保持在最后一次状态值。
[6]	RW	adc_enref	0x0	参考电平使能。0: off; 1: on
[5:2]	RW	adc_bgtrim	0x0	band-gap Trim配置值： 当使用内部参考时（enref=1），用于校准带隙电压与VREF之间的电平差值。 Bgtrim的默认值为‘0XXX’。通常8次校准就可以将band-gap与VREF之间的差异从+4%调整到+1%区间内（bgtrim值‘1000’到‘1111’）。
[1]	RW	adc_extcm	0x0	共模电压控制： 1: 共模电压由外部输入的VCM0/VCM1确定 0: 共模电压采用内部参考， (VINP0+VINM0)/2, (VINP1+VINM1)/2
[0]	RW	adc_semode	0x1	ADC输入方式可选： 0: VINP/VINM为差分输入； 1: VINP/VINM为单端输入，其中VINP连接信号输入源，VINM的连接方式分两类：AF模式下，VINM与VCM一并连接到固定电平（AF偏置）；中频模式下，VINM与VCM一并连接到GND。

#### 4.6.5.3 AUDIO\_CONTROL

偏移地址：0x80

复位方式：H

Bits	Access	Name	Default	Description
[31]	RO	ring_full_two	0x0	提示音PCM数据写入存储空满标志位判断，当该bit为1时候，表示fifo已经写满，停止向fifo中继续送数据，直到该bit为0后才能继续送数据。
[30:8]	-	reserved		
[7]	RW	ring_play_ctrl	0x0	选择提示音通路，高有效
[6]	RW	fm_play_ctrl	0x0	选择FM播放通路，高有效
[5]	RW	ahb_rd_voice_en	0x0	ahb读取音频使能开启，高有效；
[4]	RW	ahb_wr_voice_en	0x0	ahb写音频启动标志，高有效；
[3]	RW	fm_sidetone_en	0x0	模拟侧音播放使能，高有效；
[2]	RW	sel_i2s	0x0	I2S接口使能，基带测试模式下使用
[1]	RW	i2s_slave_mode	0x1	音频I2S接口主从模式设置，0主模式，1从模式，基带测试模式下使用

[0]	RW	tx_voice_source	0x0	语音帧发送（接收也需要配置）音源来源选择： 0：表示语音数据来自Codec或者声码器（应用用常规手持）； 1：表示语音数据来自mcu对应写入存储的ram空间（应用于二层BS发送模式）
-----	----	-----------------	-----	---

#### 4.6.5.4 AUDIO\_BUFFER\_CLR

偏移地址：0x84

复位方式：H

Bits	Access	Name	Default	Description
[30:2]	-	reserved		
[1]	WC	clr_rd_ram_cpu	0x0	清除声码器解码缓存buffer中数据的控制使能，该bit为1后清除该buffer中的值，然后系统自动回复该bit为0
[0]	WC	clr_wr_ram_cpu	0x0	清除声码器编码缓存buffer中数据的控制使能，该bit为1后清除该buffer中的值，然后系统自动回复该bit为0

#### 4.6.5.5 LINEOUT\_CTRL

偏移地址：0x88

复位方式：H

Bits	Access	Name	Default	Description
[31]	RO	standby_lo	0x1	Standby of the mercury DAC
[30:2]	-	reserved		
[1]	RW	line1out_en	0x0	Codec Line1out输出使能，高有效
[0]	RW	line2out_en	0x0	Codec Line2out输出使能，高有效

#### 4.6.5.6 CODEC\_I2C\_MUX

偏移地址：0x8c

复位方式：H

Bits	Access	Name	Default	Description
[31:8]	-	reserved		
[7]	RW	codec_si2c_reg	0x0	Codec IP寄存器配置接口选择： 0：MC接口；1：I2C接口。
[6:0]	RW	codec_i2c_address_reg	0x0	Codec IP I2C接口设备地址

#### 4.6.5.7 WORK\_MODE

偏移地址：0x100

复位方式：H

Bits	Access	Name	Default	Description
[31:9]	-	reserved		
[8]	RW	digital_analog_en	0x0	数字模拟同时检测开关，要求在默认配置数字模式下开启，在接收到数字信号或者模拟信号后，对应切换到数字模式或者模拟模式。 1：表示数模同时接收开启控制使能； 0：表示只开启数字或者模拟接收，根据Bit7设置。
[7]	RW	modulator_mode	0x0	工作模式设置： 0：表示DMR数字模式； 1：表示FM模拟模式。
[6]	RW	is_tier1_mode	0x1	0：表示TierI模式； 1：表示TierII模式。
[5]	RW	is_continuemode	0x1	0：表示Continue模式； 1：表示TimeSlot模式。
[4:3]	RW	layermode	0x1	层次模式设置： 0：表示物理层模式； 1：表示二层模式； 2：表示三层模式。
[2]	RW	is_repeater	0x0	0：表示非中继模式； 1：表示中继模式。
[1]	RW	is_aligned	0x1	0：表示偏移模式（非中继模式下的偏移表示单频模式）； 1：表示对齐模式。
[0]	RW	work_slot_sel	0x1	三层非中继模式必须设置成1； 0：表示时隙1； 1：表示时隙2。

#### 4.6.5.8 RF\_MODE

偏移地址：0x104

复位方式：H

Bits	Access	Name	Default	Description
[31]	RW	reserved	0x0	物理层接收测试模式下要求配置成1，正常模式设置为0
[30]	RW	reserved	0x0	
[29]	RW	reserved	0x0	物理层接收测试模式下要求配置成1，正常模式设置为0
[28:26]	-	reserved		
[25]	RW	phase_avg_en	0x0	AF模式均值计算开关，高有效
[24]	RW	af_mode	0x0	AF模式设置： 0：AF信号接收禁能；1：AF信号接收使能。

[23:16]	RW	af_recv_ctrl	0x0	AF输入信号的幅度增益控制。实际信号幅度=输入信号幅度*af_recv_ctrl/16
[15]	RW	rf_recviq_mode	0x0	PHY解调接收与实际RF接收的映射关系（以实际RF接收方案确认）： 0：表示不变； 1：表示相反。
[14:13]	RW	rf_recv_mode	0x0	RF接收模式： 0x0：表示中频接收模式； 0x1：表示中频IQ接收模式； 0x2：表示基带IQ接收模式。
[12]	RW	Adjustsyncmax	0x1	控制调整同步头检测门限使能： 0：表示固定同步检测门限设置值； 1：表示自动动态同步检测门限。 备注：LOCAL_CC寄存器的cc_opt设置为1时要求设置为0。
[11]	RW	use_dual_phy	0x0	分时隙开启两个接收机通道控制开关： 0：关闭双接收通道； 1：开启双接收通道。（应用于中继芯片接收模式、DCDM模式）
[10]	RW	two_point_ctr	0x0	两点调制测试使能： 0：表示关闭； 1：表示输出40Hz正弦波。
[9]	RW	up_ctr	0x0	时隙边界信号强度平滑上升和下降使能： 0：表示关闭； 1：表示开启。
[8]	RW	iq_just_debug	0x0	IQ两路平衡调试测试使能： 0：表示不变； 1：表示加偏置值。
[7]	RW	rf_transiq_mode	0x1	PHY调制发送与实际RF发送的映射关系（以实际RF发送方案确认）： 0：表示不变； 1：表示相反。
[6:5]	RW	rf_trans_mode	0x3	RF发送模式： 0x0：表示中频发送模式； 0x1：表示中频IQ发送模式； 0x2：表示基带IQ发送模式； 0x3：表示两点调制发送模式。
[4]	RW	ak2401_mode	0x0	AK2401模式设置： 0：AK2401模式关； 1：AK2401模式开。
[3:0]	RW	ak_adc_shift	0x0	AK2401模式下输入ADC信号移位控制寄存器

#### 4.6.5.9 SYS\_INTERP\_LIST

偏移地址：0x398

复位方式：H

Bits	Access	Name	Default	Description
[31:18]	-	reserved		
[17]	RO	rdy_next_short_lc	0x0	中转台发送CACH中的shortlc信息准备中断
[16]	RO	phy_recv_slot_abnormal	0x0	物理层模式下接收时隙边界异常提示中断
[15]	RO	fm_tx_interp	0x0	模拟发送数据中断
[14]	RO	fm_rx_interp	0x0	模拟接收数据中断
[13]	RO	fm_sig_rx_interp	0x0	模拟功能接收检测中断
[12]	RO	phy_rdy_interp	0x0	物理层模式下时隙提前中断
[11]	RO	phy_rx_interp	0x0	物理层数据接收中断，用以误码率测试
[10]	RO	ccl_rx_interp	0x0	三层接收状态中断
[9]	RO	ccl_tx_interp	0x0	三层发送状态中断
[8]	RO	gps_slot_abnormal	0x0	同播时隙边界异常中断
[7]	RO	gps_lost_interp	0x0	同播GPS脉冲丢失中断
[6]	RO	dll_rx_slot_abnormal	0x0	二层接收时隙异常中断
[5]	RO	dll_tx_slot_abnormal	0x0	二层发送时隙异常中断
[4]	RO	dll_shortlc_rx_interp	0x0	二层shortlc信息接收中断
[3]	RO	dll_frame_rx_interp	0x0	二层各类型帧信息接收中断
[2]	RO	dll_latelc_rx_interp	0x0	二层语音后接入中断
[1]	RO	dll_rdy_interp_rx	0x0	二层接收处理中断
[0]	RO	dll_rdy_interp_tx	0x0	二层发送处理中断

#### 4.6.5.10 SYS\_INTERP\_MASK

偏移地址：0x39c

复位方式：H

Bits	Access	Name	Default	Description
[31:18]	-	reserved		
[17]	RO	rdy_next_short_lc_mask	0x1	中转台发送CACH中的shortlc信息准备中断是否屏蔽： 0：屏蔽中断； 1：开启中断。
[16]	RW	phy_recv_slot_abnormal_mask	0x1	物理层模式下接收时隙边界异常提示中断： 0：屏蔽中断； 1：开启中断。
[15]	RW	fm_tx_interp_mask	0x1	模拟发送数据中断是否屏蔽：

				0: 屏蔽中断; 1: 开启中断。
[14]	RW	fm_rx_interp_mask	0x1	模拟接收数据中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[13]	RW	fm_sig_rx_interp_mask	0x1	模拟功能接收检测中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[12]	RW	phy_rdy_interp_mask	0x1	物理层模式下时隙提前中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[11]	RW	phy_rx_interp_mask	0x1	物理层数据接收中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[10]	RW	ccl_rx_interp_mask	0x1	三层接收状态中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[9]	RW	ccl_tx_interp_mask	0x1	三层发送状态中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[8]	RW	gps_slot_abnormal_mask	0x1	同播时隙边界异常中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[7]	RW	gps_lost_interp_mask	0x1	同播GPS脉冲丢失中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[6]	RW	dll_rx_slot_abnormal_mask	0x1	二层接收时隙异常中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[5]	RW	dll_tx_slot_abnormal_mask	0x1	二层发送时隙异常中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[4]	RW	dll_shortlc_rx_interp_mask	0x1	二层shortlc信息接收中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[3]	RW	dll_frame_rx_interp_mask	0x1	二层各类型帧信息接收中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[2]	RW	dll_latelc_rx_interp_mask	0x1	二层语音后接入中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[1]	RW	dll_rdy_interp_rx_mask	0x1	二层接收处理中断是否屏蔽: 0: 屏蔽中断;

				1: 开启中断。
[0]	RW	dll_rdy_interp_tx_mask	0x1	二层发送处理中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。

#### 4.6.5.11 SYS\_INTERP\_CLEAR

偏移地址: 0x3a0

复位方式: H

Bits	Access	Name	Default	Description
[31:17]	-	reserved		
[16:0]	RW	clear_sys_interp	0x0	写1自动清除系统中断向量列表

#### 4.6.5.12 LAYER3\_INTERP\_LIST

偏移地址: 0x3a4

复位方式: H

Bits	Access	Name	Default	Description
[31]	RO	ccl_latelc_rx_inter p	0x0	三层语音后接入中断
[30]	RO	ccl_frame_rx_inter p	0x0	三层各类型帧信息接收中断
[29]	RO	ccl_shortlc_rx_inter p	0x0	三层shortlc信息接收中断
[28]	RO	ccl_rx_voice_abnormal	0x0	三层语音接收异常中断
[27]	RO	ccl_rx_data_ok	0x0	三层短信接收CRC32校验OK中断
[26]	RO	ccl_rx_data_error	0x0	三层短信接收CRC32校验Error中断
[25]	RO	ccl_rx_data_abnormal	0x0	三层非确认短信接收异常中断
[24:15]	-	reserved		
[14]	RO	tx_complete_data_confirmed	0x0	三层短信业务发送完成最后一帧, 需确认等待反馈中断
[13]	RO	tx_complete_data	0x0	三层短信业务发送完成最后一帧, 无需确认等待反馈中断
[12]	RO	tx_data_finish	0x0	三层短信业务发送结束中断
[11]	RO	tx_data_allretry_start	0x0	三层短信全部重新发送开始中断
[10]	RO	tx_data_retry_start	0x0	三层短信部分重新发送开始中断
[9]	RO	tx_data_start	0x0	三层短信发送开始中断
[8]	RO	tx_v_embf_update	0x0	三层语音发送F帧的EMB区域更新中断
[7]	RO	tx_v_key_update	0x0	三层语音发送密钥更新中断
[6]	RO	tx_v_emb_update	0x0	三层语音发送EMB区域更新中断

[5]	RO	tx_voice_finish	0x0	三层语音发送结束中断
[4]	RO	tx_voice_start	0x0	三层语音发送开始中断
[3]	RO	tx_oacsu_req	0x0	三层OACSU请求呼叫中断
[2]	RO	tx_bs_overtime	0x0	三层BS激活超时中断
[1]	RO	tx_oacsu_overtime	0x0	三层OACSU呼叫请求超时中断
[0]	RO	tx_denied	0x0	三层发送请求拒绝中断

#### 4.6.5.13 LAYER3\_INTERP\_MASK

偏移地址：0x3a8

复位方式：H

Bits	Access	Name	Default	Description
[31]	RW	ccl_latelc_rx_interp_mask	0x1	三层语音后接入中断是否屏蔽： 0：屏蔽中断； 1：开启中断。
[30]	RW	ccl_frame_rx_interp_mask	0x1	三层各类型帧信息接收中断是否屏蔽： 0：屏蔽中断； 1：开启中断。
[29]	RW	ccl_shortlc_rx_interp_mask	0x1	三层shortlc信息接收中断是否屏蔽： 0：屏蔽中断； 1：开启中断。
[28]	RW	ccl_rx_voice_abnormal_mask	0x1	三层语音接收异常中断是否屏蔽： 0：屏蔽中断； 1：开启中断。
[27]	RW	ccl_rx_data_ok_mask	0x1	三层短信接收CRC32校验OK中断是否屏蔽： 0：屏蔽中断； 1：开启中断。
[26]	RW	ccl_rx_data_error_mask	0x1	三层短信接收CRC32校验Error中断是否屏蔽： 0：屏蔽中断； 1：开启中断。
[25]	RW	ccl_rx_data_abnormal_mask	0x1	三层非确认短信接收异常中断是否屏蔽： 0：屏蔽中断； 1：开启中断。
[24:15]	-	reserved		
[14]	RW	tx_complete_data_confirmed_mask	0x1	三层短信业务发送完成最后一帧，需确认等待反馈中断是否屏蔽： 0：屏蔽中断； 1：开启中断。
[13]	RW	tx_complete_data_mask	0x1	三层短信业务发送完成最后一帧，无需确认等待反馈中断是否屏蔽： 0：屏蔽中断； 1：开启中断。
[12]	RW	tx_data_finish_mask	0x1	三层短信业务发送结束中断是否屏蔽： 0：屏蔽中断；

				1: 开启中断。
[11]	RW	tx_data_allretry_start_mask	0x1	三层短信全部重新发送开始中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[10]	RW	tx_data_retry_start_mask	0x1	三层短信部分重新发送开始中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[9]	RW	tx_data_start_mask	0x1	三层短信发送开始中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[8]	RW	tx_v_embf_update_mask	0x1	三层语音发送F帧的EMB区域更新中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[7]	RW	tx_v_key_update_mask	0x1	三层语音发送密钥更新中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[6]	RW	tx_v_emb_update_mask	0x1	三层语音发送EMB区域更新中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[5]	RW	tx_voice_finish_mask	0x1	三层语音发送结束中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[4]	RW	tx_voice_start_mask	0x1	三层语音发送开始中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[3]	RW	tx_oacsu_req_mask	0x1	三层OACSU请求呼叫中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[2]	RW	tx_bs_overtime_mask	0x1	三层BS激活超时中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[1]	RW	tx_oacsu_overtime_mask	0x1	三层OACSU呼叫请求超时中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。
[0]	RW	tx_denied_mask	0x1	三层发送请求拒绝中断是否屏蔽: 0: 屏蔽中断; 1: 开启中断。

#### 4.6.5.14 LAYER3\_INTERP\_CLEAR

偏移地址: 0x3ac

复位方式: H

Bits	Access	Name	Default	Description
[31:0]	RW	clear_layer3_interp	0x0	写1自动清除三层中断向量列表

#### 4.6.5.15 INTERP\_CLEAR

偏移地址：0x3b0

复位方式：H

Bits	Access	Name	Default	Description
[31:6]	-	reserved		
[5]	WC	clear_pcm_rd_int	0x0	写1自动清除PCM数据读取中断
[4]	WC	clear_pcm_wr_int	0x0	写1自动清除PCM数据写入中断
[3]	WC	clear_rf_rx_inter	0x0	写1自动清除射频接收开始中断
[2]	WC	clear_rf_tx_inter	0x0	写1自动清除射频发送开始中断
[1]	WC	clear_time_rxslot_inte r	0x0	写1自动清除30ms接收时隙中断
[0]	WC	clear_time_txslot_inte r	0x0	写1自动清除30ms发送时隙中断

## 4.7 中断系统

### 4.7.1 概述

中断控制器是用来管理来自内部/外部的异常事件，中断CPU的正常执行程序，使CPU进入指定中断服务函数的模块。

### 4.7.2 功能描述

中断控制器具有以下特点：

- 支持64个中断源；
- 支持高电平、低电平触发，支持上升沿、下降沿触发；
- 支持中断源优先级可配；
- 支持中断源屏蔽；

中断控制器的功能框图如图所示：

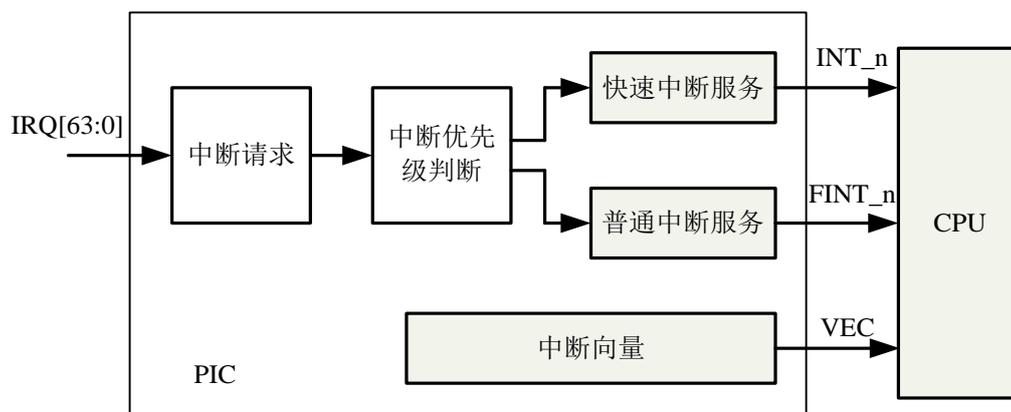


图 20 中断控制器功能框图

## 4.7.3 中断源分配

表 30 中断源列表

序号	中断源	默认中断向量	中断触发方式
0	保留	32	
1	TIMER_1	33	电平触发, 高有效
2	TIMER_2	34	电平触发, 高有效
3	TIMER_3	35	电平触发, 高有效
4	TIMER_4	36	电平触发, 高有效
5	TIMER_5	37	电平触发, 高有效
6	TIMER_6	38	电平触发, 高有效
7	I2C1	39	电平触发, 高有效
8	I2C0	40	电平触发, 高有效
9	I2C2	41	电平触发, 高有效
10	UART1	42	电平触发, 高有效
11	UART2	43	电平触发, 高有效
12	UART0	44	电平触发, 高有效
13	UART3	45	电平触发, 高有效
14	GPIOB	46	电平触发, 高有效
15	GPIOA	47	电平触发, 高有效
16	SPI0	48	电平触发, 高有效
17	GPIOC	49	电平触发, 高有效
18	PWM	50	电平触发, 高有效
19	RTC	51	电平触发, 高有效
20	SDIO	52	电平触发, 高有效
21	USB	53	电平触发, 高有效
22	SPI1	54	电平触发, 高有效
23	SPI2	55	电平触发, 高有效
24	LCSFC	56	电平触发, 高有效
25	DAC	57	电平触发, 高有效
26	ADC	58	电平触发, 高有效
27	PCM_rd_data_interp	59	电平触发, 高有效
28	PCM_wr_data_interp	60	电平触发, 高有效
29	System_inter	61	电平触发, 高有效
30	Time_slot_inter_tx	62	电平触发, 高有效
31	Time_slot_inter_rx	63	电平触发, 高有效
32	Rf_tx_inter	64	电平触发, 高有效
33	Rf_rx_inter	65	电平触发, 高有效
34	SlaveSPI0	66	电平触发, 高有效
35	SlaveSPI1	67	电平触发, 高有效
36	SlaveSPI2	68	电平触发, 高有效
37	Codec_interp	69	电平触发, 高有效

#### 4.7.4 工作方式

##### 4.7.4.1 初始化

中断控制器默认屏蔽所有中断请求，使用前需先配置中断控制器寄存器，步骤如下：

1. 配置中断触发方式：  
通过配置 PIC\_MODE 可设置中断源的触发方式；
2. 配置中断极性：  
通过配置 PIC\_PO 可设置中断源的触发极性；
3. 配置中断源优先级、快速中断等
  - 配置 PIC\_PRIOR0-3 ~ PIC\_PRIOR60-63，表示各个中断源优先级计数器的初始值。初始值小表示优先级高。如果不配置，则使用默认优先级；
  - 配置 PIC\_FFLAG\_L, PIC\_FFLAG\_H 可以将任意中断源配置成快速中断，默认全部是普通中断。C7000 仅支持普通中断，快速中断不支持使用。
4. 清中断状态（非必要）  
配置 PIC\_INT\_ST, PIC\_INT\_ST\_1 等于 0xffffffff，清楚中断记录寄存器 PIC\_INT\_ST, PIC\_INT\_ST\_1, PIC\_INT\_ST\_CNT。
5. 开启中断检测（重要）  
配置 PIC\_MASK, PIC\_MASK\_1 对应 bit 为 0，开启中断源中断请求检测功能；

##### 4.7.4.2 中断状态查询

通过查询 PIC\_INT\_ST，可获知哪些中断源发出过中断请求，每个中断源只能记录一次中断请求。对应位写1可清除状态。

#### 4.7.5 寄存器概述

寄存器表 5 PIC 寄存器概述（基地址：0x1700\_0000）

偏移地址	名称	描述	页码
0x00	PIC_MODE	低 32 位中断源触发方式选择	
0x04	PIC_PO	低 32 位中断源的触发极性选择	
0x08	PIC_MASK	低 32 位中断源是否屏蔽	
0x0c	PIC_VECTOR	中断向量基址	
0x10	PIC_COW1	中断结束	
0x14	PIC_PRIOR0-3	中断源 0-3 对应的计数器初始值	
0x18	PIC_PRIOR4-7	中断源 4-7 对应的计数器初始值	
0x1c	PIC_PRIOR8-11	中断源 8-11 对应的计数器初始值	
0x20	PIC_PRIOR12-15	中断源 12-15 对应的计数器初始值	
0x24	PIC_PRIOR16-19	中断源 16-19 对应的计数器初始值	
0x28	PIC_PRIOR20-23	中断源 20-23 对应的计数器初始值	
0x2c	PIC_PRIOR24-27	中断源 24-27 对应的计数器初始值	
0x30	PIC_PRIOR28-31	中断源 28-31 对应的计数器初始值	
0x34	PIC_COW2	控制寄存器	
0x38	PIC_SYNC	异步处理控制寄存器	
0x3c	PIC_FFLAG_L	低 32 位快速中断设置	

0x40	PIC_RECORD_SEL	选择一个中断源记录中断次数	
0x44	PIC_INT_ST	0-31 中断请求记录	
0x48	PIC_INT_ST_1	32-63 中断请求记录	
0x4c	PIC_INT_CNT	中断源中断次数计数	
0x60	PIC_MODE_1	高 32 位中断源触发方式选择	
0x64	PIC_PO_1	高 32 位中断源的触发极性选择	
0x68	PIC_MASK_1	高 32 位中断源是否屏蔽	
0x6c	PIC_PRIOR32-35	中断源 32-35 对应的计数器初始值	
0x70	PIC_PRIOR36-39	中断源 36-39 对应的计数器初始值	
0x74	PIC_PRIOR40-43	中断源 40-43 对应的计数器初始值	
0x78	PIC_PRIOR44-47	中断源 44-47 对应的计数器初始值	
0x7c	PIC_PRIOR48-51	中断源 48-51 对应的计数器初始值	
0x80	PIC_PRIOR52-55	中断源 52-55 对应的计数器初始值	
0x84	PIC_PRIOR56-59	中断源 56-59 对应的计数器初始值	
0x88	PIC_PRIOR60-63	中断源 60-63 对应的计数器初始值	
0x8c	PIC_FFLAG_H	高 32 位快速中断设置	

#### 4.7.6 寄存器描述

##### 4.7.6.1 PIC\_MODE

偏移地址：0x00

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	R/W	edge_level	0xffffffff	每一位控制相应中断源的触发方式，IRQ0对应第0位，IRQ31对应第31位。 1: 边缘触发； 0: 电平触发。

##### 4.7.6.2 PIC\_PO

偏移地址：0x04

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	R/W	pol	0xffffffff	每一位控制相应中断源的触发极性，IRQ0对应第0位，IRQ31对应第31位。 1: 上升沿触发/高电平触发； 0: 下跳沿触发/低电平触发。

##### 4.7.6.3 PIC\_MASK

偏移地址：0x08

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	mask	0xffffffff	每一位控制相应中断源是否屏蔽，IRQ0对应第0位，

				IRQ31对应第31位。 1: 中断屏蔽; 0: 中断开放。
--	--	--	--	--------------------------------------

#### 4.7.6.4 PIC\_VECTOR

偏移地址: 0x0c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:7]	-	reserved	-	
[6:0]	RW	vector0	0x20	IRQ0对应的中断向量, 取值范围是32 ~ 112。 vector0是中断向量表的基址, 后面的IRQx 的VEC = x + vector0; 其中, x是中断源端口号。

#### 4.7.6.5 PIC\_COW1

偏移地址: 0x10

复位方式: H/S

Bits	Access	Name	Default	Description
[31:4]	-	reserved	-	
[3]	WO	feoi	0x0	当cpu处理完快速中断后, 将该位写1, 得到feoi后, 中断控制器会将FISR寄存器清零, 并且将FINT (低电平有效) 请求线拉高。 1 : CPU中断服务结束; 0 : CPU 还在处理中断服务。
[2]	WO	eoi	0x0	当cpu处理完中断后, 将该位写1, 得到eoi信号后, 中断控制器会将ISR寄存器清零, 并且将INT (低电平有效) 请求线拉高。 1 : CPU中断服务结束; 0 : CPU 还在处理中断服务。
[1:0]	-	reserved	-	

#### 4.7.6.6 PIC\_PRIOR0-3

偏移地址: 0x14

复位方式: H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq3	0x3	中断源3优先级计数器初始值, 值小优先级高
[23:16]	RW	prior_irq2	0x2	中断源2优先级计数器初始值, 值小优先级高
[15:8]	RW	prior_irq1	0x1	中断源1优先级计数器初始值, 值小优先级高
[7:0]	RW	prior_irq0	0x0	中断源0优先级计数器初始值, 值小优先级高

#### 4.7.6.7 PIC\_PRIOR4-7

偏移地址：0x18

复位方式：H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq7	0x7	中断源7优先级计数器初始值，值小优先级高
[23:16]	RW	prior_irq6	0x6	中断源6优先级计数器初始值，值小优先级高
[15:8]	RW	prior_irq5	0x5	中断源5优先级计数器初始值，值小优先级高
[7:0]	RW	prior_irq4	0x4	中断源4优先级计数器初始值，值小优先级高

#### 4.7.6.8 PIC\_PRIOR8-11

偏移地址：0x1C

复位方式：H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq11	0xb	中断源11优先级计数器初始值，值小优先级高
[23:16]	RW	prior_irq10	0xa	中断源10优先级计数器初始值，值小优先级高
[15:8]	RW	prior_irq9	0x9	中断源9优先级计数器初始值，值小优先级高
[7:0]	RW	prior_irq8	0x8	中断源8优先级计数器初始值，值小优先级高

#### 4.7.6.9 PIC\_PRIOR12-15

偏移地址：0x20

复位方式：H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq15	0xf	中断源15优先级计数器初始值，值小优先级高
[23:16]	RW	prior_irq14	0xe	中断源14优先级计数器初始值，值小优先级高
[15:8]	RW	prior_irq13	0xd	中断源13优先级计数器初始值，值小优先级高
[7:0]	RW	prior_irq12	0xc	中断源12优先级计数器初始值，值小优先级高

#### 4.7.6.10 PIC\_PRIOR16-19

偏移地址：0x24

复位方式：H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq19	0x13	中断源19优先级计数器初始值，值小优先级高
[23:16]	RW	prior_irq18	0x12	中断源18优先级计数器初始值，值小优先级高
[15:8]	RW	prior_irq17	0x11	中断源17优先级计数器初始值，值小优先级高
[7:0]	RW	prior_irq16	0x10	中断源16优先级计数器初始值，值小优先级高

#### 4.7.6.11 PIC\_PRIOR20-23

偏移地址：0x28

复位方式：H/S

Bits	Access	Name	Default	Description
------	--------	------	---------	-------------

[31:24]	RW	prior_irq23	0x17	中断源23优先级计数器初始值, 值小优先级高
[23:16]	RW	prior_irq22	0x16	中断源22优先级计数器初始值, 值小优先级高
[15:8]	RW	prior_irq21	0x15	中断源21优先级计数器初始值, 值小优先级高
[7:0]	RW	prior_irq20	0x14	中断源20优先级计数器初始值, 值小优先级高

#### 4.7.6.12 PIC\_PRIOR24-27

偏移地址: 0x2C

复位方式: H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq27	0x1b	中断源27优先级计数器初始值, 值小优先级高
[23:16]	RW	prior_irq26	0x1a	中断源26优先级计数器初始值, 值小优先级高
[15:8]	RW	prior_irq25	0x19	中断源25优先级计数器初始值, 值小优先级高
[7:0]	RW	prior_irq24	0x18	中断源24优先级计数器初始值, 值小优先级高

#### 4.7.6.13 PIC\_PRIOR28-31

偏移地址: 0x30

复位方式: H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq31	0x1f	中断源31优先级计数器初始值, 值小优先级高
[23:16]	RW	prior_irq30	0x1e	中断源30优先级计数器初始值, 值小优先级高
[15:8]	RW	prior_irq29	0x1d	中断源29优先级计数器初始值, 值小优先级高
[7:0]	RW	prior_irq28	0x1c	中断源28优先级计数器初始值, 值小优先级高

#### 4.7.6.14 PIC\_COW2

偏移地址: 0x34

复位方式: H/S

Bits	Access	Name	Default	Description
[31:2]	-	reserved	-	
[1]	RW	avec	0x1	1 : CPU 使用中断控制器送入的VEC; 0 : CPU 使用自动向量号0x 28, 或者 0x 2c。
[0]	RW	priority	0x0	1 : 采用用户自定义的中断源优先级设置, 即采用 prior_irq[x]的优先级设置, 初始值小表示优先级高。 0 : 采用默认的优先级设置 IRQ0 > IRQ1 > ..... > IRQ5 > IRQ6 > ..... > IRQ63。

#### 4.7.6.15 PIC\_SYNC

偏移地址: 0x38

复位方式: H/S

Bits	Access	Name	Default	Description
------	--------	------	---------	-------------

[31:8]	-	reserved	-	
[7:0]	RW	min_interval	0x0	异步处理时，第一个脉冲信号之后的电平持续时间，和两个时钟域的频率差有关。防止两个脉冲过近，导致两个脉冲异步处理后都丢失

#### 4.7.6.16 PIC\_FFLAG\_L

偏移地址：0x3c

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	fflag_l	0x0	1：表示快速中断，0：表示普通中断。快速中断的优先级大于所有普通中断，快速中断之间通过计数器值判定优先级，初始值小优先级高。

#### 4.7.6.17 PIC\_INT\_RECORD\_SEL

偏移地址：0x40

复位方式：H/S

Bits	Access	Name	Default	Description
[31:7]	-	reserved	-	
[6:0]	RW	irq_record_sel	0x0	选择一个中断源，记录中断次数（经过异步和mask处理后端中断请求）PIC_INT_ST_CNT。

#### 4.7.6.18 PIC\_INT\_ST

偏移地址：0x44

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RWC	int_state	0x0	记录低32个中断源的中断请求。对应位写1，则清除记录，同时清除PIC_INT_ST_CNT的值。

#### 4.7.6.19 PIC\_INT\_ST\_1

偏移地址：0x48

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RWC	int_state	0x0	记录高32个中断源的中断请求。对应位写1，则清除记录，同时清除PIC_INT_ST_CNT的值。

#### 4.7.6.20 PIC\_INT\_ST\_CNT

偏移地址：0x48

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RO	int_cnt	0x0	记录PIC_INT_RECORD_SEL选择的 中断源的中断请求次数。

#### 4.7.6.21 PIC\_MODE\_1

偏移地址：0x60

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	edge_level	0xffffffff	每一位控制相应中断源的触发方式，IRQ32对应第0位，IRQ63对应第31位。 1：边缘触发； 0：电平触发。

#### 4.7.6.22 PIC\_PO\_1

偏移地址：0x64

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	pol	0xffffffff	每一位控制相应中断源的触发极性，IRQ32对应第0位，IRQ63对应第31位。 1：上升沿触发 / 高电平触发； 0：下跳沿触发 / 低电平触发。

#### 4.7.6.23 PIC\_MASK\_1

偏移地址：0x68

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	mask	0xffffffff	每一位控制相应中断源是否屏蔽，IRQ32对应第0位，IRQ63对应第31位。 1：中断屏蔽； 0：中断开放。

#### 4.7.6.24 PIC\_PRIOR32-35

偏移地址：0x6c

复位方式：H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq35	0x27	中断源35优先级计数器初始值，值小优先级高
[23:16]	RW	prior_irq34	0x26	中断源34优先级计数器初始值，值小优先级高
[15:8]	RW	prior_irq33	0x25	中断源33优先级计数器初始值，值小优先级高
[7:0]	RW	prior_irq32	0x24	中断源32优先级计数器初始值，值小优先级高

#### 4.7.6.25 PIC\_PRIOR36-39

偏移地址: 0x70

复位方式: H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq39	0x27	中断源39优先级计数器初始值, 值小优先级高
[23:16]	RW	prior_irq38	0x26	中断源38优先级计数器初始值, 值小优先级高
[15:8]	RW	prior_irq37	0x25	中断源37优先级计数器初始值, 值小优先级高
[7:0]	RW	prior_irq36	0x24	中断源36优先级计数器初始值, 值小优先级高

#### 4.7.6.26 PIC\_PRIOR40-43

偏移地址: 0x74

复位方式: H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq43	0x2b	中断源43优先级计数器初始值, 值小优先级高
[23:16]	RW	prior_irq42	0x2a	中断源42优先级计数器初始值, 值小优先级高
[15:8]	RW	prior_irq41	0x29	中断源41优先级计数器初始值, 值小优先级高
[7:0]	RW	prior_irq40	0x28	中断源40优先级计数器初始值, 值小优先级高

#### 4.7.6.27 PIC\_PRIOR44-47

偏移地址: 0x78

复位方式: H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq47	0x2f	中断源47优先级计数器初始值, 值小优先级高
[23:16]	RW	prior_irq46	0x2e	中断源46优先级计数器初始值, 值小优先级高
[15:8]	RW	prior_irq45	0x2d	中断源45优先级计数器初始值, 值小优先级高
[7:0]	RW	prior_irq44	0x2c	中断源44优先级计数器初始值, 值小优先级高

#### 4.7.6.28 PIC\_PRIOR48-51

偏移地址: 0x7c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq51	0x33	中断源51优先级计数器初始值, 值小优先级高
[23:16]	RW	prior_irq50	0x32	中断源50优先级计数器初始值, 值小优先级高
[15:8]	RW	prior_irq49	0x31	中断源49优先级计数器初始值, 值小优先级高
[7:0]	RW	prior_irq48	0x30	中断源48优先级计数器初始值, 值小优先级高

#### 4.7.6.29 PIC\_PRIOR52-55

偏移地址: 0x80

复位方式：H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq55	0x37	中断源55优先级计数器初始值，值小优先级高
[23:16]	RW	prior_irq54	0x36	中断源54优先级计数器初始值，值小优先级高
[15:8]	RW	prior_irq53	0x35	中断源53优先级计数器初始值，值小优先级高
[7:0]	RW	prior_irq52	0x34	中断源52优先级计数器初始值，值小优先级高

#### 4.7.6.30 PIC\_PRIOR56-59

偏移地址：0x84

复位方式：H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq59	0x3b	中断源59优先级计数器初始值，值小优先级高
[23:16]	RW	prior_irq58	0x3a	中断源58优先级计数器初始值，值小优先级高
[15:8]	RW	prior_irq57	0x39	中断源57优先级计数器初始值，值小优先级高
[7:0]	RW	prior_irq56	0x38	中断源56优先级计数器初始值，值小优先级高

#### 4.7.6.31 PIC\_PRIOR60-63

偏移地址：0x88

复位方式：H/S

Bits	Access	Name	Default	Description
[31:24]	RW	prior_irq63	0x3f	中断源63优先级计数器初始值，值小优先级高
[23:16]	RW	prior_irq62	0x3e	中断源62优先级计数器初始值，值小优先级高
[15:8]	RW	prior_irq61	0x3d	中断源61优先级计数器初始值，值小优先级高
[7:0]	RW	prior_irq60	0x3c	中断源60优先级计数器初始值，值小优先级高

#### 4.7.6.32 PIC\_FFLAG\_H

偏移地址：0x8c

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	fflag_h	0x0	1：表示快速中断，0：表示普通中断。快速中断的优先级大于所有普通中断，快速中断之间通过计数器值判定优先级。

## 4.8 定时器

### 4.8.1 概述

HR\_C7000 包含 6 个独立的 32 位定时器（Timer1~6），实现定时、计数功能，可以供操作系统用作系统时钟，也可以供应用程序用作定时和计数。

#### 4.8.2 功能描述

定时器有以下特点：

- Timer1~6可单独使用，各自产生定时器中断；
- 支持自由运行和用户定义计数两种工作模式；
- 支持查询四个定时器的当前计数值。

#### 4.8.3 工作方式

定时器实现了 4 个相同但是相互独立的可编程定时器。定时器从一个固定的值向下计数，当计数值变为 0 时会产生中断。

每个定时器的初始值会从相应的加载计数寄存器（TimerNLoadCount）里加载。以下两种事件会触发从 TimerNLoadCount 寄存器加载初始值：

- 定时器在复位或被禁用后使能
- 定时器计数到 0

所有的状态寄存器和中断寄存器都可以在任何时刻访问。

#### 4.8.4 寄存器概述

寄存器表 6 Timer 寄存器概述（基地址：0x1400\_0000）

偏移地址	名称	描述	页码
0x00	Timer1LoadCount	定时器 1 加载寄存器	
0x04	Timer1CurrentValue	定时器 1 当前值寄存器	
0x08	Timer1ControlReg	定时器 1 控制寄存器	
0x0c	Timer1EOI	定时器 1 中断清除寄存器	
0x10	Timer1IntStatus	定时器 1 中断状态寄存器	
0x14	Timer2LoadCount	定时器 2 加载寄存器	
0x18	Timer2CurrentValue	定时器 2 当前值寄存器	
0x1c	Timer2ControlReg	定时器 2 控制寄存器	
0x20	Timer2EOI	定时器 2 中断清除寄存器	
0x24	Timer2IntStatus	定时器 2 中断状态寄存器	
0x28	Timer3LoadCount	定时器 3 加载寄存器	
0x2c	Timer3CurrentValue	定时器 3 当前值寄存器	
0x30	Timer3ControlReg	定时器 3 控制寄存器	
0x34	Timer3EOI	定时器 3 中断清除寄存器	
0x38	Timer3IntStatus	定时器 3 中断状态寄存器	
0x3c	Timer4LoadCount	定时器 4 加载寄存器	
0x40	Timer4CurrentValue	定时器 4 当前值寄存器	
0x44	Timer4ControlReg	定时器 4 控制寄存器	
0x48	Timer4EOI	定时器 4 中断清除寄存器	
0x4c	Timer4IntStatus	定时器 4 中断状态寄存器	
0x3c	Timer5LoadCount	定时器 5 加载寄存器	
0x40	Timer5CurrentValue	定时器 5 当前值寄存器	
0x44	Timer5ControlReg	定时器 5 控制寄存器	

0x48	Timer5EOI	定时器 5 中断清除寄存器	
0x4c	Timer5IntStatus	定时器 5 中断状态寄存器	
0x3c	Timer6LoadCount	定时器 6 加载寄存器	
0x40	Timer6CurrentValue	定时器 6 当前值寄存器	
0x44	Timer6ControlReg	定时器 6 控制寄存器	
0x48	Timer6EOI	定时器 6 中断清除寄存器	
0x4c	Timer6IntStatus	定时器 6 中断状态寄存器	
0xa0	TimersIntStatus	所有定时器的中断状态	
0xa4	TimersEOI	清除所有寄存器的中断	
0xa8	TimersRawIntStatus	所有定时器的未屏蔽的中断状态	
0xac	TIMERS_COMP_VERSION	定时器版本号	

## 4.8.5 寄存器描述

### 4.8.5.1 Timer1LoadCount

偏移地址: 0x0

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer1LoadCount	0x0	定时器1加载寄存器

### 4.8.5.2 Timer1CurrentValue

偏移地址: 0x4

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer1LoadValue	0x0	定时器1当前值寄存器

### 4.8.5.3 Timer1ControlReg

偏移地址: 0x8

复位方式: H/S

Bits	Access	Name	Default	Description
[31:3]	-	Reserved	-	保留
[2]	RW	TimerInterruptMask	0x0	定时器1中断屏蔽
[1]	RW	TimerMode	0x0	定时器1模式
[0]	RW	TimerEnable	0x0	定时器1使能

### 4.8.5.4 Timer1EOI

偏移地址: 0xc

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留

[0]	RC	EOI	0x0	读返回0，并清除定时器1的中断
-----	----	-----	-----	-----------------

#### 4.8.5.5 Timer1IntStatust

偏移地址：0x10

复位方式：H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留
[0]	RO	IntStatus	0x0	定时器1的中断状态

#### 4.8.5.6 Timer2LoadCount

偏移地址：0x14

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer2LoadCount	0x0	定时器2加载寄存器

#### 4.8.5.7 Timer2CurrentValue

偏移地址：0x18

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer2LoadValue	0x0	定时器2当前值寄存器

#### 4.8.5.8 Timer2ControlReg

偏移地址：0x1c

复位方式：H/S

Bits	Access	Name	Default	Description
[31:3]	-	Reserved	-	保留
[2]	RW	TimerInterruptMask	0x0	定时器2中断屏蔽
[1]	RW	TimerMode	0x0	定时器2模式
[0]	RW	TimerEnable	0x0	定时器2使能

#### 4.8.5.9 Timer2EOI

偏移地址：0x20

复位方式：H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留
[0]	RC	EOI	0x0	读返回0，并清除定时器1的中断

#### 4.8.5.10 Timer2IntStatust

偏移地址: 0x24

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留
[0]	RO	IntStatus	0x0	定时器2的中断状态

#### 4.8.5.11 Timer3LoadCount

偏移地址: 0x28

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer1LoadCount	0x0	定时器3加载寄存器

#### 4.8.5.12 Timer3CurrentValue

偏移地址: 0x2c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer1LoadValue	0x0	定时器3当前值寄存器

#### 4.8.5.13 Timer3ControlReg

偏移地址: 0x30

复位方式: H/S

Bits	Access	Name	Default	Description
[31:3]	-	Reserved	-	保留
[2]	RW	TimerInterruptMask	0x0	定时器3中断屏蔽
[1]	RW	TimerMode	0x0	定时器3模式
[0]	RW	TimerEnable	0x0	定时器3使能

#### 4.8.5.14 Timer3EOI

偏移地址: 0x34

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留
[0]	RC	EOI	0x0	读返回0, 并清除定时器3的中断

#### 4.8.5.15 Timer3IntStatust

偏移地址: 0x38

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留
[0]	RO	IntStatus	0x0	定时器3的中断状态

#### 4.8.5.16 Timer4LoadCount

偏移地址: 0x3c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer4LoadCount	0x0	定时器4加载寄存器

#### 4.8.5.17 Timer4CurrentValue

偏移地址: 0x40

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer4LoadValue	0x0	定时器4当前值寄存器

#### 4.8.5.18 Timer4ControlReg

偏移地址: 0x44

复位方式: H/S

Bits	Access	Name	Default	Description
[31:3]	-	Reserved	-	保留
[2]	RW	TimerInterruptMask	0x0	定时器4中断屏蔽
[1]	RW	TimerMode	0x0	定时器4模式
[0]	RW	TimerEnable	0x0	定时器4使能

#### 4.8.5.19 Timer4EOI

偏移地址: 0x48

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留
[0]	RC	EOI	0x0	读返回0, 并清除定时器4的中断

#### 4.8.5.20 Timer4IntStatust

偏移地址: 0x4c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留
[0]	RO	IntStatus	0x0	定时器4的中断状态

#### 4.8.5.21 Timer5LoadCount

偏移地址: 0x0

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer5LoadCount	0x0	定时器5加载寄存器

#### 4.8.5.22 Timer5CurrentValue

偏移地址: 0x4

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer5LoadValue	0x0	定时器5当前值寄存器

#### 4.8.5.23 Timer5ControlReg

偏移地址: 0x8

复位方式: H/S

Bits	Access	Name	Default	Description
[31:3]	-	Reserved	-	保留
[2]	RW	TimerInterruptMask	0x0	定时器5中断屏蔽
[1]	RW	TimerMode	0x0	定时器5模式
[0]	RW	TimerEnable	0x0	定时器5使能

#### 4.8.5.24 Timer5EOI

偏移地址: 0xc

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留
[0]	RC	EOI	0x0	读返回0, 并清除定时器5的中断

#### 4.8.5.25 Timer5IntStatust

偏移地址: 0x10

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留
[0]	RO	IntStatus	0x0	定时器5的中断状态

#### 4.8.5.26 Timer6LoadCount

偏移地址: 0x0

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer6LoadCount	0x0	定时器6加载寄存器

#### 4.8.5.27 Timer6CurrentValue

偏移地址: 0x4

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Timer6LoadValue	0x0	定时器6当前值寄存器

#### 4.8.5.28 Timer6ControlReg

偏移地址: 0x8

复位方式: H/S

Bits	Access	Name	Default	Description
[31:3]	-	Reserved	-	保留
[2]	RW	TimerInterruptMask	0x0	定时器6中断屏蔽
[1]	RW	TimerMode	0x0	定时器6模式
[0]	RW	TimerEnable	0x0	定时器6使能

#### 4.8.5.29 Timer6EOI

偏移地址: 0xc

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留
[0]	RC	EOI	0x0	读返回0, 并清除定时器6的中断

#### 4.8.5.30 Timer6IntStatust

偏移地址: 0x10

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	-	Reserved	-	保留
[0]	RO	IntStatus	0x0	定时器6的中断状态

#### 4.8.5.31 TimersIntStatus

偏移地址: 0xa0

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	TimersIntStatus	0x0	所有定时器的中断状态

#### 4.8.5.32 TimersEOI

偏移地址: 0xa4

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RC	TimersEOI	0x0	读返回0, 并清除所有定时器的中断

#### 4.8.5.33 TimersRawIntStatus

偏移地址: 0xa8

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	TimersRawIntStatus	0x0	所有定时器未屏蔽的中断状态

#### 4.8.5.34 TIMERS\_COMP\_VERSION

偏移地址: 0xac

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	TIMERS_COMP_VERSION	0x3230322a	模块版本

## 4.9 RTC

### 4.9.1 概述

HR\_C7000 包含一个实时时钟模块, 负责系统的计时, 可保存当前世界时间, 可以设置某个未来时间点产生中断信号。

### 4.9.2 功能描述

RTC 主要有以下特点:

- RTC 内部使用 6 位宽的秒分寄存器, 5 位宽的时寄存器, 16 位宽的日寄存器, 秒寄存器使用 1Hz 的时钟计数, 当计数到 60 分时寄存器加一, 秒寄存器自身清零。依次类推。其中天寄存器有 16 位, 可以计 65526 天, 相当于 179 年。
- 中断产生时间可配置, 并可以控制中断是否产生。
- RTC 初始时间可配置。
- 可以控制 RTC 开始或停止。
- 芯片掉电以后 RTC 依然工作, 使用外部纽扣电池为其供电。

### 4.9.3 工作方式

HR\_C7000 中 RTC 挂载于内部 I<sup>2</sup>C 上, 通过 I<sup>2</sup>C 总线进行操作。RTC 计时时钟为外部提供的 32.768KHz 晶体。

#### 4.9.3.1 RTC 初始化

RTC 在首次上电时，系统需要将 RTC 初始化。RTC 的初始化过程如下：

1. 拉低 rtc\_rst\_n，复位 RTC 的 I<sup>2</sup>C 接口模块。
2. 等待 60ns（3 个 apb 时钟周期）。
3. 配置 RTC\_SOFT\_RST\_N 寄存器为 0，执行软复位。（设备地址 7'b1110000）
4. 等待 25ms（计数器在 128Hz 的时钟下工作，异步复位的同步撤离需要 23.44ms）。
5. 配置 RTC\_SOFT\_RST\_N 寄存器为 1，撤销软复位。
6. 依次配置 RTC\_MATCH\_S、RTC\_MATCH\_M、RTC\_MATCH\_H、RTC\_MATCH\_D\_L、RTC\_MATCH\_D\_H，设置 RTC 中断产生的时间，配置 RTC\_MATCH\_VEN[0]为 1 更新中断产生时间（此位会自动清零），配置 RTC\_MATCH\_VSTAT[0]为 1 表示当前中断产生时间已经配置，如果需要重新配置中断产生时间时先将此位配置为 0，配置中断产生时间以后再将此位配置为 1。
7. 依次配置 RTC\_LOAD\_S、RTC\_LOAD\_M、RTC\_LOAD\_H、RTC\_LOAD\_D\_L、RTC\_LOAD\_D\_H，设置 RTC 的初始时间，配置 RTC\_CCR[2]为 1 更新当前的时间（此位会自动清零），配置 RTC\_LOAD\_VSTAT[0]为 1 表示初始时间已经设置，如果需要重新配置初始时间时先将此位配置为 0，配置初始时间以后再将此位配置为 1。
8. 配置 RTC\_CCR[0]为 1，使能中断。
9. 配置 RTC\_CCR[1]为 1，让计时器开始计时。
10. RTC 以 1HZ 的计数时钟频率，从 RTC\_LOAD\_S、RTC\_LOAD\_M、RTC\_LOAD\_H、RTC\_LOAD\_D\_L、RTC\_LOAD\_D\_H 中的值开始计数，当计数值到达 RTC\_MATCH\_S、RTC\_MATCH\_M、RTC\_MATCH\_H、RTC\_MATCH\_D\_L、RTC\_MATCH\_D\_H 中的值时，将根据 RTC\_CCR[0]的设置，决定是否产生中断。

#### 4.9.3.2 中断处理

系统收到 RTC 发出的中断以后，表示定时的时间到，用户可以执行相应的自定义操作，RTC 计数器任保持计数。RTC 中断的处理过程如下：

1. 读 RTC\_RSTAT，清除 RTC 的中断状态。
2. 如果还需要设置定时时间，先关闭中断使能，配置 RTC\_MATCH\_VSTAT[0]=0，向 RTC\_MATCH\_S、RTC\_MATCH\_M、RTC\_MATCH\_H、RTC\_MATCH\_D\_L、RTC\_MATCH\_D\_H 设置新的 RTC 比较值，配置 RTC\_MATCH\_VEN[0]=1，RTC\_MATCH\_VSTAT[0]=1 再开启中断使能。

#### 4.9.4 寄存器概述

寄存器表 7 RTC 寄存器概述（基地址：0x1408\_0000）

偏移地址	名称	描述	页码
0x00	RTC_VAL_S	RTC_VAL_S 的总体参数配置	
0x01	RTC_VAL_M	RTC_VAL_M 的总体参数配置	
0x02	RTC_VAL_H	RTC_VAL_H 的总体参数配置	
0x03	RTC_VAL_D_L	RTC_VAL_D_L 的总体参数配置	
0x04	RTC_VAL_D_H	RTC_VAL_D_H 的总体参数配置	
0x05	RTC_MATCH_S	RTC_MATCH_S 的总体参数配置	
0x06	RTC_MATCH_M	RTC_MATCH_M 的总体参数配置	
0x07	RTC_MATCH_H	RTC_MATCH_H 的总体参数配置	

0x08	RTC_MATCH_D_L	RTC_MATCH_D_L 的总体参数配置	
0x09	RTC_MATCH_D_H	RTC_MATCH_D_H 的总体参数配置	
0x0a	RTC_MATCH_VEN	RTC_MATCH_VEN 的总体参数配置	
0x0b	RTC_MATCH_VSTAT	RTC_MATCH_VSTAT 的总体参数配置	
0x0c	RTC_LOAD_S	RTC_LOAD_S 的总体参数配置	
0x0d	RTC_LOAD_M	RTC_LOAD_M 的总体参数配置	
0x0e	RTC_LOAD_H	RTC_LOAD_H 的总体参数配置	
0x0f	RTC_LOAD_D_L	RTC_LOAD_D_L 的总体参数配置	
0x10	RTC_LOAD_D_H	RTC_LOAD_D_H 的总体参数配置	
0x11	RTC_CCR	RTC_CCR 的总体参数配置	
0x12	RTC_LOAD_VSTAT	RTC_LOAD_VSTAT 的总体参数配置	
0x13	RTC_RSTAT	RTC_RSTAT 数配置	
0x14	RTC_CLKDIV_NUMER_L	RTC_CLKDIV_NUMER_L 数配置	
0x15	RTC_CLKDIV_NUMER_H	RTC_CLKDIV_NUMER_H 数配置	
0x16	RTC_CLKDIV_NUMER_VLD	RTC_CLKDIV_NUMER_VLD 数配置	
0x17	RTC_SOFT_RST_N	RTC_SOFT_RST_N 数配置	

#### 4.9.5 寄存器描述

##### 4.9.5.1 RTC\_VAL\_S

RTC\_VAL\_S 的总体参数配置

偏移地址: 0x0000

Bits	Access	Name	Default	Description
[7:6]	-	reserved	-	保留
[5:0]	RO	rtc_val_s	0x00	记录当前时间的秒位

##### 4.9.5.2 RTC\_VAL\_M

RTC\_VAL\_M 的总体参数配置

偏移地址: 0x0001

Bits	Access	Name	Default	Description
[7:6]	-	reserved	-	保留
[5:0]	RO	rtc_val_m	0x00	记录当前时间的分位

##### 4.9.5.3 RTC\_VAL\_H

RTC\_VAL\_H 的总体参数配置

偏移地址: 0x0002

Bits	Access	Name	Default	Description
[7:5]	-	reserved	-	保留
[4:0]	RO	rtc_val_h	0x00	记录当前时间的小时位

##### 4.9.5.4 RTC\_VAL\_D\_L

RTC\_VAL\_D\_L 的总体参数配置

偏移地址: 0x0003

Bits	Access	Name	Default	Description
[7:0]	RO	rtc_val_d_l	0x00	记录当前时间的天数的低八位

#### 4.9.5.5 RTC\_VAL\_D\_H

RTC\_VAL\_D\_H 的总体参数配置

偏移地址: 0x0004

Bits	Access	Name	Default	Description
[7:0]	RO	rtc_val_d_h	0x00	记录当前时间的天数的高八位

#### 4.9.5.6 RTC\_MATCH\_S

RTC\_MATCH\_S 的总体参数配置

偏移地址: 0x0005

Bits	Access	Name	Default	Description
[7:6]	-	reserved	-	保留
[5:0]	RW	rtc_match_s	0x00	与 count 值比较, 看是否产生中断

#### 4.9.5.7 RTC\_MATCH\_M

RTC\_MATCH\_M 的总体参数配置

偏移地址: 0x0006

Bits	Access	Name	Default	Description
[7:6]	-	reserved	-	保留
[5:0]	RW	rtc_match_m	0x00	与 count 值比较, 看是否产生中断

#### 4.9.5.8 RTC\_MATCH\_H

RTC\_MATCH\_H 的总体参数配置

偏移地址: 0x0007

Bits	Access	Name	Default	Description
[7:5]	-	reserved	-	保留
[4:0]	RW	rtc_match_h	0x00	与 count 值比较, 看是否产生中断

#### 4.9.5.9 RTC\_MATCH\_D\_L

RTC\_MATCH\_D\_L 的总体参数配置

偏移地址: 0x0008

Bits	Access	Name	Default	Description
[7:0]	RW	rtc_match_d_l	0x00	与 count 值比较, 看是否产生中断

#### 4.9.5.10 RTC\_MATCH\_D\_H

RTC\_MATCH\_D\_H 的总体参数配置

偏移地址: 0x0009

Bits	Access	Name	Default	Description
[7:0]	RW	rtc_match_d_h	0x00	与 count 值比较, 看是否产生中断

#### 4.9.5.11 RTC\_MATCH\_VEN

RTC\_MATCH\_VEN 的总体参数配置

偏移地址：0x000A

Bits	Access	Name	Default	Description
[7:1]	-	reserved	-	保留
[0]	RW	rtc_match_ven	0x00	匹配数据使能信号，如果为 1，则将用户设置的值写入到内部的匹配寄存器中，防止中断的误触发。

#### 4.9.5.12 RTC\_MATCH\_VSTAT

RTC\_MATCH\_VSTAT 的总体参数配置

偏移地址：0x000B

Bits	Access	Name	Default	Description
[7:1]	-	reserved	-	保留
[0]	RW	rtc_match_vstat	0x00	为防止在配置匹配寄存器时掉电引起数据的不确定性，用户在配置完 MATCH 相关寄存器后将其拉高，表示 MATCH 相关的寄存器已经配置完毕。下次需要配置 MATCH 相关寄存器前先将此位配置为 0，配置完后再将此位配置为 1。

#### 4.9.5.13 RTC\_LOAD\_S

RTC\_LOAD\_S 的总体参数配置

偏移地址：0x000C

Bits	Access	Name	Default	Description
[7:6]	-	reserved	-	保留
[5:0]	RW	rtc_load_s	0x00	记录预载入值的秒位

#### 4.9.5.14 RTC\_LOAD\_M

RTC\_LOAD\_M 的总体参数配置

偏移地址：0x000D

Bits	Access	Name	Default	Description
[7:6]	-	reserved	-	保留
[5:0]	RW	rtc_load_m	0x00	记录预载入值分位

#### 4.9.5.15 RTC\_LOAD\_H

RTC\_LOAD\_H 的总体参数配置

偏移地址：0x000E

Bits	Access	Name	Default	Description
[7:5]	-	reserved	-	保留
[4:0]	RW	rtc_load_h	0x00	记录预载入值的小时位

#### 4.9.5.16 RTC\_LOAD\_D\_L

RTC\_LOAD\_D\_L 的总体参数配置

偏移地址：0x000F

Bits	Access	Name	Default	Description
[7:0]	RW	rtc_load_d_l	0x00	记录预载入值的天数低八位

#### 4.9.5.17 RTC\_LOAD\_D\_H

RTC\_LOAD\_D\_H 的总体参数配置

偏移地址: 0x0010

Bits	Access	Name	Default	Description
[7:0]	RW	rtc_load_d_h	0x00	记录预载入值的天数高八位

#### 4.9.5.18 RTC\_CCR

RTC\_CCR 的总体参数配置

偏移地址: 0x0011

Bits	Access	Name	Default	Description
[7:3]	N/A	reserved	0x00	无效位, 默认为 0
[2]	RW	rtc_load	0x00	当该位为 1 时, count 从 load 寄存器载入预设值, 载入后该位会自动清零
[1]	RW	rtc_enable	0x00	让用户选择计数器是否计数。 0=计数器停止计数 1=计数器开始计数
[0]	RW	rtc_intr_en	0x00	让用户选择是否屏蔽中断。 0=中断不使能 1=中断使能

#### 4.9.5.19 RTC\_LOAD\_VSTAT

RTC\_LOAD\_VSTAT 的总体参数配置

偏移地址: 0x0012

Bits	Access	Name	Default	Description
[7:1]	-	reserved	-	保留
[0]	RW	rtc_load_vstat	0x00	为防止在预载寄存器时掉电引起数据的不确定性, 用户在配置完 LOAD 相关寄存器后将其拉高, 表示 LOAD 相关的寄存器已经配置完毕。下次需要配置 LOAD 相关寄存器前先将此位配置为 0, 配置完后再将此位配置为 1。

#### 4.9.5.20 RTC\_RSTAT

RTC\_RSTAT 数配置

偏移地址: 0x0013

Bits	Access	Name	Default	Description
[7:1]	-	reserved	-	保留
[0]	RC	rtc_rstat	0x00	0=中断没有产生 1=中断产生了 (忽略优先级) 该位可读清零

#### 4.9.5.21 RTC\_CLKDIV\_NUMER\_L

RTC\_CLKDIV\_NUMER\_L 数配置

偏移地址：0x0014

Bits	Access	Name	Default	Description
[7:0]	RW	rtc_clkdiv_numer_l	0x00	数字分频的低 8 位，总共为 16 位，可以设置 0~65535 之间的任意整数，设分频数为 t，那么 rtc 计时的周期为(晶振频率/t)hz, 如 rtc 如果使用 32.768k 的晶振，如果想要分得 1hz 的计时时钟，那么分频数为 32768。

#### 4.9.5.22 RTC\_CLKDIV\_NUMER\_H

RTC\_CLKDIV\_NUMER\_H 数配置

偏移地址：0x0015

Bits	Access	Name	Default	Description
[7:0]	RW	rtc_clkdiv_numer_h	0x00	数字分频的高 8 位，总共为 16 位，可以设置 0~65535 之间的任意整数，设分频数为 t，那么 rtc 计时的周期为(晶振频率/t)hz, 如 rtc 如果使用 32.768k 的晶振，如果想要分得 1hz 的计时时钟，那么分频数为 32768。

#### 4.9.5.23 RTC\_CLKDIV\_NUMER\_VLD

RTC\_CLKDIV\_NUMER\_VLD 数配置

偏移地址：0x0016

Bits	Access	Name	Default	Description
[7:1]	-	reserved	-	保留
[0]	RW	rtc_clkdiv_nummer_vld	0x00	分频系数有效信号，如果这一位为 0，使用默认 32768 的分频系数，如果为 1，使用用户设置的分频系数

#### 4.9.5.24 RTC\_SOFT\_RST\_N

RTC\_SOFT\_RST\_N 数配置

偏移地址：0x0017

Bits	Access	Name	Default	Description
[7:1]	-	reserved	-	保留
[0]	RW	rtc_soft_rst_n	0x01	0=软复位 1=软复位无效

## 4.10 看门狗

### 4.10.1 概述

看门狗用于在系统死机时对系统重新复位。

### 4.10.2 功能描述

看门狗主要有两个特点：

- 一旦有外部物理复位信号到来，看门狗输出低电平复位信号并持续 510 个系统时钟时间，延长物理复位时间，保证系统充分复位；
- 防止系统死机，软件需要在一定时间周期内发出敲门信号给看门狗，在预定时间内看门狗没有得到激励，则认为系统死机，将自动对系统进行复位，复位电平持续时间为 510

个系统时钟。

### 4.10.3 工作方式

#### 4.10.3.1 开启看门狗

系统启动后配置 `WD_CW.wde=1`，即可开启看门狗。

#### 4.10.3.2 关闭看门狗

关闭看门狗的操作略显繁琐，这是为了防止软件误操作或者电磁干扰等原因而意外关闭看门狗。关闭过程如下：

1. 在 `wde` 为高的情况下，将 `WD_CW.wdtoe` 置 1；
2. 将 `WD_CW.wde` 置 0，此时看门狗被关闭；
3. 将 `WD_CW.wdtoe` 置 0 为初始值。

关闭之后，看门狗的初始计数值，以及其他控制字均未被改变，可以在开启看门狗后继续使用，时序图如下：

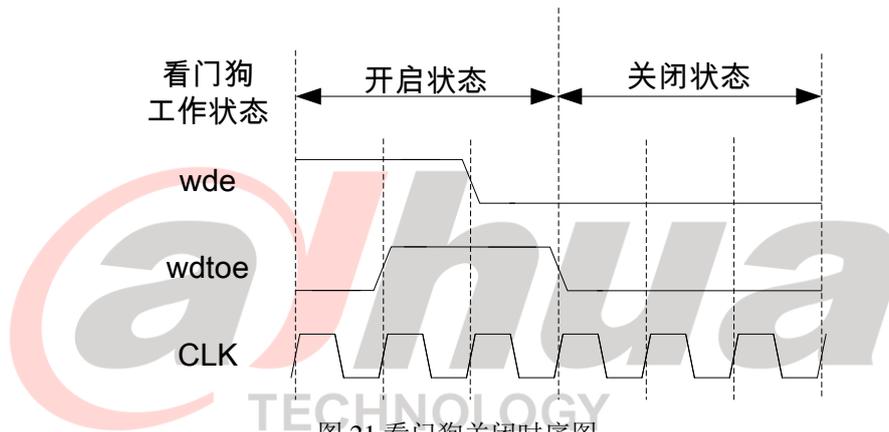


图 21 看门狗关闭时序图

表 31 看门狗工作状态表

wdtoe	wde	看门狗工作情况
0	0	不允许
0	1	开启
1	0	关闭（从11跳入才关闭，直接从10跳入不会关闭）
1	1	开启

#### 4.10.3.3 正常工作

正常工作模式下，需要系统周期性地写看门狗寄存器 `WD_CW.knocker`，如果超过指定周期没有写该寄存器，则看门狗认为系统死机，会重新复位系统。

1. 配置 `WD_TIMER`，设置看门狗周期初始值；
2. 配置 `WD_CW.wde=1` 开启看门狗；
3. 在内部计数器从初始值 `WD_TIMER` 计数到溢出值之前，配置一次 `WD_CW.knocker=1`，完

成敲门动作。敲门后，内部计数器重新从WD\_TIMER开始计数。

#### 4.10.4 寄存器概述

寄存器表 8 WDT 寄存器概述（基地址：0x1401\_0000）

偏移地址	名称	描述	页码
0x0	WDG_LOCK	看门狗锁定寄存器	
0x4	WD_OT_LOAD_H	看门狗超时计数器高位	
0x8	WD_OT_LOAD_L	看门狗超时计数器低位	
0xc	WDG_RST_LOAD	看门狗复位时间初始值	
0x10	WDG_EN	看门狗使能	
0x14	WDG_KNOCK	看门狗喂狗寄存器	

#### 4.10.5 寄存器描述

##### 4.10.5.1 WDG\_LOCK

偏移地址：0x0

复位方式：H

Bits	Access	Name	Default	Description
[31:0]	RW	WDG_LOCK	0x1	向该寄存器写入0x5ada_7200,可打开所有寄存器的写权限，向该寄存器写入其它值则关闭所有寄存器写权限。读该寄存器返回加锁状态： 0x0：允许写，未加锁 0x1：禁止写，已加锁

##### 4.10.5.2 WDG\_OT\_LOAD\_H

偏移地址：0x4

复位方式：H

Bits	Access	Name	Default	Description
[31:3]	RO	reserved	0x0	保留
[2:0]	RW	WDG_OT_LOAD_H	0x7	超时计数器高位初始值，高位计数器溢出时，表示超时，默认超时时间60s左右

##### 4.10.5.3 WDG\_OT\_LOAD\_L

偏移地址：0x8

复位方式：H

Bits	Access	Name	Default	Description
[31:0]	RO	WDG_OT_LOAD_L	0xc0000000	超时计数器低位初始值，高位计数器溢出时，表示超时，默认超时时间60s左右

#### 4.10.5.4 WDG\_RST\_LOAD

偏移地址: 0x c

复位方式: H

Bits	Access	Name	Default	Description
[31:25]	RO	reserved	0x0	保留
[24:0]	RW	WDG_RST_LO AD	0x03f80000	看门狗复位时间初始值, 计数到0x03ffffff后复位完成, 默认复位时间约20ms

#### 4.10.5.5 WDG\_EN

偏移地址: 0x10

复位方式: H

Bits	Access	Name	Default	Description
[31:1]	RO	reserved	0x0	保留
[0]	RW	WDG_EN	0x0	Watchdog使能信号 0: 看门狗关闭, 不会产生看门狗复位 1: 看门狗工作, 超时时间内没有喂狗, 会产生看门狗复位

#### 4.10.5.6 WDG\_KNOCK

偏移地址: 0x14

复位方式: H

Bits	Access	Name	Default	Description
[31:0]	RW	WDG_KNOCK	0x0	喂狗信号, 写0x55aadd22表示喂狗, 看门狗超时计数器重新开始计数。

## 5 外围设备

### 5.1 I<sup>2</sup>C

#### 5.1.1 概述

I<sup>2</sup>C (The Inter-Integrated Circuit) 接口有 2 个信号: SDA (串行数据或地址线) 和 SCL (串行时钟线)。HR\_C7000 I<sup>2</sup>C 模块作为 I<sup>2</sup>C 总线 Master, 可对 I<sup>2</sup>C 从设备进行通信, 遵循 I<sup>2</sup>C 总线协议 2.1 版本。

#### 5.1.2 功能描述

I<sup>2</sup>C 模块有以下特点:

- 支持标准的 100kbit/s 数据传输速率和快速模式 400kbit/s;
- 提供 TX FIFO 和 RX FIFO;
- 支持中断上报, 中断查询。

I<sup>2</sup>C 在传送数据 (Data Transfer) 时, SDA 上的数据信号必须在 SCL 处于高电平时保持稳定, 而只能在 SCL 处于低电平时变换相位。

START 传送的条件是: 在 SCL 处于高电平时, SDA 有一个 High-to-Low 的跳变。

STOP 传送的条件是: 在 SCL 处于高电平时, SDA 有一个 Low-to-High 的跳变。这个过程下图所示:

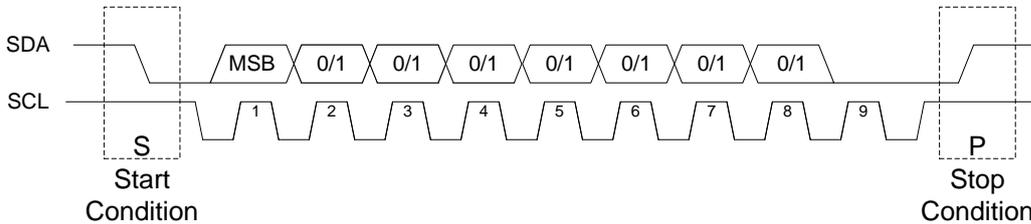


图 22 I<sup>2</sup>C 总线的数据传送过程

I<sup>2</sup>C 总线上的每个数据包是 8 位位宽, 外加一个 ACK 位, 所以一个完整的数据传送包括 8 个时钟周期。当 Receiver 接受完 8 位后, 将 SDA 置低来表示 ACK, 此时, Transmitter 需要释放 SDA 引线。如果 Slave Receiver 没有发送 ACK, 则表示 Receiver 没有收到数据, 那么 Master 可以发送 STOP 信号终止数据传送。如果 Master Receiver 没有发送 ACK, 则表示传送结束, 那么 Slave Transmitter 可以终止传送。

表 32 I<sup>2</sup>C 接口信号

信号名	方向	描述
I <sup>2</sup> C_SCL	B	I <sup>2</sup> C 串行时钟线
I <sup>2</sup> C_SDA	B	I <sup>2</sup> C 串行数据地址线

### 5.1.3 时钟频率配置

I<sup>2</sup>C 作为 master, 需要配置时钟频率。I<sup>2</sup>C 支持快速模式, 支持的最高数据速率为 400kb/s。根据公式 (1) 计算时钟频率。公式 (1) 中的 SCL\_PERIOD\_FS 表示数据速率。IC\_HCNT\_FS 表示快速模式下, 在 SCL 高电平有效期间, I<sup>2</sup>C 时钟周期计数; IC\_LCNT\_FS 表示快速模式下, 在 SCL 低电平有效期间, I<sup>2</sup>C 时钟周期计数。IC\_HCNT\_FS 是寄存器 IC\_FS\_SCL\_HCNT 的配置值加 8, IC\_LCNT\_FS 是寄存器 IC\_FS\_SCL\_LCNT 的配置值加 1。

$$\frac{SCL\_PERIOD\_FS}{IC\_HCNT\_FS + IC\_LCNT\_FS} = IC\_CLK\_PERIOD \quad \text{公式 (1)}$$

例如, 数据速率是 400kb/s 时, SCL\_PERIOD\_FS = 2.5us, I<sup>2</sup>C 工作时钟为 APB 时钟为 50M, 根据以上公式即可计算出 IC\_HCNT\_FS 和 IC\_LCNT\_FS, 进而寄存器 IC\_FS\_SCL\_HCNT 和 IC\_FS\_SCL\_LCNT 的配置值分别是 (IC\_HCNT\_FS - 8) 和 (IC\_LCNT\_FS - 1)。寄存器 IC\_FS\_SCL\_HCNT 最小值不小于 6, 寄存器 IC\_FS\_SCL\_LCNT 最小值不小于 8。

### 5.1.4 工作方式

#### 5.1.4.1 初始化

I<sup>2</sup>C 控制器初始化流程如下:

- 1) 配置寄存器 IC\_ENABLE, 将 I2C 关断;
- 2) 配置寄存器 IC\_INTR\_MASK, 屏蔽中断;

- 3) 配置寄存器 IC\_FS\_SCL\_HCNT 和 IC\_FS\_SCL\_LCNT，设置时钟频率；
- 4) 配置寄存器 IC\_CON；
- 5) 写从机地址寄存器 IC\_TAR；
- 6) 配置发送 FIFO 阈值寄存器 IC\_TX\_TL 和接收 FIFO 阈值寄存器 IC\_RX\_TL；
- 7) 配置寄存器 IC\_INTR\_MASK，打开中断配置；
- 8) 寄存器 IC\_ENABLE，使能 I2C。

#### 5.1.4.2 数据发送和接收

数据发送过程：

- 1、向寄存器 IC\_DATA\_CMD 写入数据，配置 IC\_START 寄存器，启动数据发送；
- 2、查询方式下，轮询寄存器 IC\_STATUS 监控发送 FIFO 的空满状态。中断方式下，当检测到发送 FIFO 阈值中断 TX\_EMPTY（发送 FIFO 数据量小于等于阈值时产生）时，写数据到发送 FIFO，同时查询寄存器 IC\_STATUS 判断 FIFO 空满，当 FIFO 满，则停止写数据；

注意：

- (1) 在发送数据完成之前，发送 FIFO 不能空，因为一旦发送 FIFO 为空，会触发 STOP 条件，导致数据发送结束。
- (2) 若需要关掉 I2C，需要保证 I2C 处于空闲状态，即查询并满足满足寄存器 IC\_STATUS bit0 为 0。

接收数据流程：

- 1、写寄存器 IC\_DATA\_CMD 的 bit[8]，配置 IC\_START 寄存器，发送读数据命令；
- 2、查询方式下，轮询寄存器 IC\_STATUS，当检测到接收 FIFO 不为空时，读寄存器 IC\_DATA\_CMD。中断方式下，检测到接收 FIFO 阈值中断 RX\_FULL（当接收 FIFO 数据大于设置阈值时触发）时，读接收寄存器 IC\_DATA\_CMD，同时查询寄存器 IC\_STATUS 将接收 FIFO 读空。

#### 5.1.4.3 停用 I<sup>2</sup>C 控制器

在 IC\_ENABLE 寄存器从 1 配为 0 之后，软件可以通过监测 IC\_ENABLE\_STATUS 寄存器来确定硬件是否完成关闭过程。

停用 I<sup>2</sup>C 控制器的过程为：

- 1) 定义一个时间间隔 (Ti2c\_poll) 等于 I<sup>2</sup>C 最高传输周期的 10 倍。例如如果最高传输速度是 400kbps，则 Ti2c\_poll 为 25us。
- 2) 定义最大超时参数，MAX\_T\_POLL\_COUNT，使任何大于此值的重复查询操作都会报错。
- 3) 禁止发起新的 I<sup>2</sup>C 传输，但是允许未完成的传输继续
- 4) 初始化 POLL\_COUNT 变量为 0。
- 5) 设置 IC\_ENABLE 寄存器为 0。
- 6) 读 IC\_ENABLE\_STATUE 寄存器，检查器 IC\_EN 位(第 0 位)。将 POLL\_COUNT 加一。如果 POLL\_COUNT 大于等于 MAX\_T\_POLL\_COUNT 则返回错误。
- 7) 如果 IC\_EN 位为 1，则等待 Ti2c\_poll 时间，并返回上一步。否则成功退出。

## 5.1.5 寄存器概述

寄存器表 9 I2C0 寄存器概述（基地址为 0x14060000）

偏移地址	名称	描述	页码
0x0	IC_CON	I2C 控制寄存器	
0x4	IC_TAR	I2C 目标地址寄存器	
0x10	IC_DATA_CMD	I2C 收发数据缓存和命令	
0x1c	IC_FS_SCL_HCNT	快速 SCL 高电平计数	
0x20	IC_FS_SCL_LCNT	快速 SCL 低电平计数	
0x2c	IC_INTR_STAT	中断状态	
0x30	IC_INTR_MASK	中断屏蔽	
0x34	IC_RAW_INTR_STAT	原始中断状态	
0x38	IC_RX_TL	接收 FIFO 阈值	
0x3c	IC_TX_TL	发送 FIFO 阈值	
0x40	IC_CLR_INTR	清除中断	
0x44	IC_CLR_RX_UNDER	清除中断 RX_UNDER	
0x48	IC_CLR_RX_OVER	清除中断 RX_OVER	
0x4c	IC_CLR_TX_OVER	清除中断 TX_OVER	
0x50	IC_CLR_RD_REQ	清除中断 RD_REQ	
0x54	IC_CLR_TX_ABRT	清除中断 TX_ABRT	
0x58	IC_CLR_RX_DONE	清除中断 RX_DONE	
0x5c	IC_CLR_ACTIVITY	清除中断 ACTIVITY	
0x60	IC_CLR_STOP_DET	清除中断 STOP_DET	
0x64	IC_CLR_START_DET	清除中断 START_DET	
0x6c	IC_ENABLE	I2C 使能寄存器	
0x70	IC_STATUS	I2C 状态寄存器	
0x74	IC_TXFLR	发送 FIFO 水位寄存器	
0x78	IC_RXFLR	接收 FIFO 水位寄存器	
0x80	IC_TX_ABRT_SOURCE	I2C 传输终止状态寄存器	
0x9c	IC_ENABLE_STATUS	I2C 使能状态寄存器	
0xa0	IC_START	控制数据开始发送	

寄存器表 10 I2C1 寄存器概述（基地址为 0x14070000）

偏移地址	名称	描述	页码
0x0	IC_CON	I2C 控制寄存器	
0x4	IC_TAR	I2C 目标地址寄存器	
0x10	IC_DATA_CMD	I2C 收发数据缓存和命令	
0x1c	IC_FS_SCL_HCNT	快速 SCL 高电平计数	
0x20	IC_FS_SCL_LCNT	快速 SCL 低电平计数	
0x2c	IC_INTR_STAT	中断状态	
0x30	IC_INTR_MASK	中断屏蔽	
0x34	IC_RAW_INTR_STAT	原始中断状态	
0x38	IC_RX_TL	接收 FIFO 阈值	
0x3c	IC_TX_TL	发送 FIFO 阈值	

0x40	IC_CLR_INTR	清除中断	
0x44	IC_CLR_RX_UNDER	清除中断 RX_UNDER	
0x48	IC_CLR_RX_OVER	清除中断 RX_OVER	
0x4c	IC_CLR_TX_OVER	清除中断 TX_OVER	
0x50	IC_CLR_RD_REQ	清除中断 RD_REQ	
0x54	IC_CLR_TX_ABRT	清除中断 TX_ABRT	
0x58	IC_CLR_RX_DONE	清除中断 RX_DONE	
0x5c	IC_CLR_ACTIVITY	清除中断 ACTIVITY	
0x60	IC_CLR_STOP_DET	清除中断 STOP_DET	
0x64	IC_CLR_START_DET	清除中断 START_DET	
0x6c	IC_ENABLE	I2C 使能寄存器	
0x70	IC_STATUS	I2C 状态寄存器	
0x74	IC_TXFLR	发送 FIFO 水位寄存器	
0x78	IC_RXFLR	接收 FIFO 水位寄存器	
0x80	IC_TX_ABRT_SOURCE	I2C 传输终止状态寄存器	
0x9c	IC_ENABLE_STATUS	I2C 使能状态寄存器	
0xa0	IC_START	控制数据开始发送	

## 5.1.6 寄存器描述

### 5.1.6.1 IC\_CON

偏移地址：0x00

复位方式：H/S

Bits	Access	Name	Default	Description
[15:7]	-	reserved	-	保留
[6]	RW	IC_SLAVE_DISA BLE	0x1	表示复位后，I2C是否作为slave。 0: salve使能 1: slave关断
[5]	RW	IC_RESTART_EN	0x1	当I2C作为master时，是否支持RESTART。 0: 关断 1: 使能
[4]	RW	IC_10BITADDR_ MASTER	0x1	作为master时，I2C传输的地址模式。 0: 7-bit地址模式 1: 10-bit地址模式
[3]	RW	IC_10BITADDR_ SLAVE	0x1	作为slave时，控制I2C响应的地址模式。 0: 7-bit地址模式 1: 10-bit地址模式
[2:1]	RW	SPEED	0x2	控制I2C的速度模式。 1: 标准模式（100 kbit/s） 2: 快速模式（400 kbit/s） 3: 高速模式（3.4 Mbit/s）（不支持）
[0]	RW	MASTER_MODE	0x1	I2C作为master是否使能。 0: master关断

				1: master使能
--	--	--	--	-------------

#### 5.1.6.2 IC\_TAR

偏移地址: 0x04

复位方式: H/S

Bits	Access	Name	Default	Description
[15:12]	-	reserved	-	保留
[11]	RW	SPECIAL	0x0	表示软件是否执行广播呼叫或起始字节命令。(不支持这两种功能)
[10]	RW	GC_OR_START	0x0	只有当bit11设为1时, 此位有效。表示软件执行的广播呼叫和起始字节命令的哪一种。 0: 广播呼叫 1: 起始字节
[9:0]	RW	IC_TAR	0x55	表示I2C作为master时, 对应的目标slave地址。

#### 5.1.6.3 IC\_DATA\_CMD

偏移地址: 0x10

复位方式: H/S

Bits	Access	Name	Default	Description
[15:9]	-	reserved	-	保留
[8]	WO	CMD	0x0	I2C作为master时, 读写控制位。 1: 读 0: 写
[7:0]	RW	DAT	0x0	发送到I2C总线或者接收到的I2C总线的的数据。

#### 5.1.6.4 IC\_FS\_SCL\_HCNT

偏移地址: 0x1C

复位方式: H/S

Bits	Access	Name	Default	Description
[15:0]	RW	IC_FS_SCL_HCNT	0x5a	设置在快速模式下, SCL时钟高电平有效时的ic_clk的周期个数

#### 5.1.6.5 IC\_FS\_SCL\_LCNT

偏移地址: 0x20

复位方式: H/S

Bits	Access	Name	Default	Description
[15:0]	RW	IC_FS_SCL_LCN	0xa0	设置在快速模式下, SCL时钟低电平有效时的ic_clk的周期个数

### 5.1.6.6 IC\_INTR\_STAT

偏移地址：0x2C

复位方式：H/S

Bits	Access	Name	Default	Description
[15:12]	-	reserved	-	保留
[11]	RO	R_GEN_CALL	0x0	详细解释参照IC_RAW_INTR_STAT寄存器
[10]	RO	R_START_DET	0x0	
[9]	RO	R_STOP_DET	0x0	
[8]	RO	R_ACTIVITY	0x0	
[7]	RO	R_RX_DONE	0x0	
[6]	RO	R_TX_ABRT	0x0	
[5]	RO	R_RD_REQ	0x0	
[4]	RO	R_TX_EMPTY	0x0	
[3]	RO	R_TX_OVER	0x0	
[2]	RO	R_RX_FULL	0x0	
[1]	RO	R_RX_OVER	0x0	
[0]	RO	R_RX_UNDER	0x0	

### 5.1.6.7 IC\_INTR\_MASK

偏移地址：0x30

复位方式：H/S

Bits	Access	Name	Default	Description
[15:12]	-	reserved	-	保留
[11]	RW	M_GEN_CALL	0x8ff	详细解释参照IC_RAW_INTR_STAT寄存器
[10]	RW	M_START_DET	0x8ff	
[9]	RW	M_STOP_DET	0x8ff	
[8]	RW	M_ACTIVITY	0x8ff	
[7]	RW	M_RX_DONE	0x8ff	
[6]	RW	M_TX_ABRT	0x8ff	
[5]	RW	M_RD_REQ	0x8ff	
[4]	RW	M_TX_EMPTY	0x8ff	
[3]	RW	M_TX_OVER	0x8ff	
[2]	RW	M_RX_FULL	0x8ff	
[1]	RW	M_RX_OVER	0x8ff	
[0]	RW	M_RX_UNDER	0x8ff	

### 5.1.6.8 IC\_RAW\_INTR\_STAT

偏移地址：0x34

复位方式：H/S

Bits	Access	Name	Default	Description
[15:12]	-	reserved	-	保留

[11]	RO	GEN_CALL	0x0	接收到广播呼叫地址时有效。（不支持）
[10]	RO	START_DET	0x0	master或者slave模式下，表示是否检测到START或RESTART
[9]	RO	STOP_DET	0x0	master或者slave模式下，表示是否检测到STOP
[8]	RO	ACTIVITY_NEG E	0x0	表示I2C由活跃状态变为空闲状态。（即I2C活跃状态的下降沿）
[7]	RO	RX_DONE	0x0	当I2C作为slave发送器时，如果master没有ACK已发送的字节，此位置为1，表示传输结束。
[6]	RO	TX_ABRT	0x0	I2C作为发送器，发送中止中断。产生原因查看IC_TX_ABRT_SOURCE寄存器
[5]	RO	RD_REQ	0x0	当I2C作为slave，其他master从I2C读数据时，会产生此中断。此中断有效时，I2C将SCL拉低并保持，直到cpu响应此中断写数据到 IC_DATA_CMD寄存器。
[4]	RO	TX_EMPTY	0x0	当发送FIFO数据低于等于 IC_TX_TL寄存器设的阈值时，会产生发送FIFO阈值中断。
[3]	RO	TX_OVER	0x0	当发送FIFO达到设定深度，且处理器继续写数据时，会产生发送溢出中断。
[2]	RO	RX_FULL	0x0	当接收FIFO数据大于等于IC_RX_TL寄存器设置的阈值时，会产生接收FIFO阈值中断。
[1]	RO	RX_OVER	0x0	当接收FIFO满，并继续接收数据时，产生溢出中断。
[0]	RO	RX_UNDER	0x0	当接收FIFO空了，处理器读数据时，产生下溢中断。

#### 5.1.6.9 IC\_RX\_TL

偏移地址：0x38

复位方式：H/S

Bits	Access	Name	Default	Description
[15:8]	-	reserved	-	保留
[7:0]	RW	TX_TL	0x6	设置接收FIFO的阈值，范围0-7。阈值设为7时，接收FIFO数据达到7，即会触发RX_FULL中断；阈值设为0-6时，大于此阈值才会触发RX_FULL中断。

#### 5.1.6.10 IC\_TX\_TL

偏移地址：0x3C

复位方式：H/S

Bits	Access	Name	Default	Description
[15:8]	-	reserved	-	保留
[7:0]	RW	TX_TL	0x6	设置发送FIFO的阈值，范围0-7，小于等于此阈值会触发TX_EMPTY中断。

#### 5.1.6.11 IC\_CLR\_INTR

偏移地址：0x40

复位方式: H/S

Bits	Access	Name	Default	Description
[15:1]	-	reserved	-	保留
[0]	RO	CLR_INTR	0x0	读此寄存器清除组合中断、所有单个中断和 IC_TX_ABRT_SOURCE 寄存器。只能清软件可清除中断。

#### 5.1.6.12 IC\_CLR\_RX\_UNDER

偏移地址: 0x44

复位方式: H/S

Bits	Access	Name	Default	Description
[15:1]	-	reserved	-	保留
[0]	RO	CLR_RX_UNDER	0x0	清除 RX_UNDER 中断。

#### 5.1.6.13 IC\_CLR\_RX\_OVER

偏移地址: 0x48

复位方式: H/S

Bits	Access	Name	Default	Description
[15:1]	-	reserved	-	保留
[0]	RO	CLR_RX_OVER	0x0	清除 RX_OVER 中断。

#### 5.1.6.14 IC\_CLR\_TX\_OVER

偏移地址: 0x4C

复位方式: H/S

Bits	Access	Name	Default	Description
[15:1]	-	reserved	-	保留
[0]	RO	CLR_TX_OVER	0x0	清除TX_OVER 中断。

#### 5.1.6.15 IC\_CLR\_RD\_REQ

偏移地址: 0x50

复位方式: H/S

Bits	Access	Name	Default	Description
[15:1]	-	reserved	-	保留
[0]	RO	CLR_RD_REQ	0x0	清除RD_REQ 中断。

#### 5.1.6.16 IC\_CLR\_TX\_ABRT

偏移地址: 0x54

复位方式: H/S

Bits	Access	Name	Default	Description
------	--------	------	---------	-------------

[15:1]	-	reserved	-	保留
[0]	RO	CLR_TX_ABRT	0x0	读寄存器可以清除TX_ABRT中断和IC_TX_ABRT_SOURCE寄存器。

#### 5.1.6.17 IC\_CLR\_RX\_DONE

偏移地址：0x58

复位方式：H/S

Bits	Access	Name	Default	Description
[15:1]	-	reserved	-	保留
[0]	RO	CLR_RX_DONE	0x0	清除RX_DONE 中断。

#### 5.1.6.18 IC\_CLR\_ACTIVITY

偏移地址：0x5c

复位方式：H/S

Bits	Access	Name	Default	Description
[15:1]	-	reserved	-	保留
[0]	RO	CLR_ACTIVITY	0x0	清除ACTIVITY 中断。

#### 5.1.6.19 IC\_CLR\_STOP\_DET

偏移地址：0x60

复位方式：H/S

Bits	Access	Name	Default	Description
[15:1]	-	reserved	-	保留
[0]	RO	CLR_STOP_DET	0x0	清除STOP_DET 中断。

#### 5.1.6.20 IC\_CLR\_START\_DET

偏移地址：0x64

复位方式：H/S

Bits	Access	Name	Default	Description
[15:1]	-	reserved	-	保留
[0]	RO	CLR_START_DE T	0x0	清除START_DET 中断。

#### 5.1.6.21 IC\_ENABLE

偏移地址：0x6c

复位方式：H/S

Bits	Access	Name	Default	Description
[15:1]	-	reserved	-	保留

[0]	RW	ENABLE	0x0	控制I2C使能。 0: I2C关断 1: I2c使能
-----	----	--------	-----	----------------------------------

#### 5.1.6.22 IC\_STATUS

偏移地址: 0x70

复位方式: H/S

Bits	Access	Name	Default	Description
[31:7]	-	reserved	-	保留
[6]	RO	SLV_ACTIVITY	0x0	slave状态机活跃状态。 0: slave状态机处于空闲态 1: slave状态机非空闲状态
[5]	RO	MST_ACTIVITY	0x0	master状态机活跃状态。 0: master状态机处于空闲态 1: master状态机非空闲状态
[4]	RO	RFF	0x0	接收FIFO彻底满。 0: 接收FIFO未滿 1: 接收FIFO滿
[3]	RO	RFNE	0x0	接收FIFO不空。 0: 接收FIFO空 1: 接收FIFO不空
[2]	RO	TFE	0x1	发送FIFO彻底空。 0: 发送FIFO未空 1: 发送FIFO空
[1]	RO	TFNFR	0x1	发送FIFO不空。 0: 发送FIFO滿 1: 发送FIFO未滿
[0]	RO	ACTIVITY	0x0	I2C处于活跃状态

#### 5.1.6.23 IC\_TXFLR

偏移地址: 0x74

复位方式: H/S

Bits	Access	Name	Default	Description
[31:4]	-	reserved	-	保留
[3:0]	RO	TXFLR	0x0	发送FIFO的有效数据个数

#### 5.1.6.24 IC\_RXFLR

偏移地址: 0x78

复位方式: H/S

Bits	Access	Name	Default	Description
[31:4]	-	reserved	-	保留

[3:0]	RO	RXFLR	0x0	接收FIFO的有效数据个数
-------	----	-------	-----	---------------

### 5.1.6.25 IC\_TX\_ABRT\_SOURCE

偏移地址：0x80

复位方式：H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved	-	保留
[15]	RO	ABRT_SLVRD_I NTX	0x0	当I2C作为slave-transmitter时，I2C请求发送数据，处理器向发送FIFO写读命令。
[14]	RO	ABRT_SLV_ARB LOST	0x0	作为Slave-Transmitter，在传输数据给master时，丢失仲裁。
[13]	RO	ABRT_SLVFLUS H_TXFIFO	0x0	作为Slave-Transmitter，I2C接收到读命令，发送FIFO中已经有一些数据，会发起 TX_ABRT中断释放原来的数据
[12]	RO	ARB_LOST	0x0	作为Master-Transmitter或者Slave-Transmitter时，失去仲裁
[11]	RO	ABRT_MASTER_ DIS	0x0	作为master时，在master模式关断时，访问master。
[10]	RO	ABRT_10B_RD_ NORSTR	0x0	作为Master-Receiver，RESTART功能关断，但在10-bit地址模式下发送读命令
[9]	RO	ABRT_SBYTE_N ORSTR	0x0	作为master，RESTART关断时，发送起始字节
[8]	RO	ABRT_HS_NORS TRT	0x0	作为master，RESTART功能关断，但是试图在高速模式下传输数据
[7]	RO	ABRT_SBYTE_A CKDET	0x0	作为master，I2C发送起始字节，且起始字节得到ack（正常情况，起始字节不需要ack）
[6]	RO	ABRT_HS_ACKD ET	0x0	作为master时，I2C处于高速模式时，高速主机码被ack。（正常情况下不需要ack）
[5]	RO	ABRT_GCALL_R EAD	0x0	作为master-tansmitter，I2C在读数据时发出了广播呼叫
[4]	RO	ABRT_GCALL_N OACK	0x0	作为master-tansmitter，发出的广播呼叫地址，没有slave响应。
[3]	RO	ABRT_TXDATA_ NOACK	0x0	作为master-tansmitter时，发出的地址得到ack，但接下来发出去的数据没有得slave的ack
[2]	RO	ABRT_10ADDR2 _NOACK	0x0	作为master，地址模式时10-bit模式，发出的第二个地址没有收到任何slave的回复。
[1]	RO	ABRT_10ADDR1 _NOACK	0x0	I2C作为master时，地址模式时10-bit模式，发出的第一个地址没有收到任何slave的回复。
[0]	RO	ABRT_7B_ADDR _NOACK	0x0	I2C作为master时，地址模式时7-bit模式，发出的地址没有收到任何slave的回复。

### 5.1.6.26 IC\_ENABLE\_STATUS

偏移地址：0x9c

复位方式：H/S

Bits	Access	Name	Default	Description
[31:3]	-	reserved	-	保留
[2]	RO	SLV_RX_DATA_LOST	0x0	只有当bit0是0时，此位有效，表示在slave-receive模式下，接收完 $\geq 1$ 个byte数据时，由于I2C使能被关断，放弃数据传输。当置为1时表示，传输已经进入数据阶段，且接收完至少1个byte数据，I2C使能被关断，不管是否收到ack，都放弃传输。置为0，表示I2C关断时，未到数据传输阶段。（不支持I2C作为slave）
[1]	RO	SLV_DISABLED_WHILE_BUSY	0x0	只有当bit0为0时，此位有效。表示，i2c作为slave时，由于I2C使能被关断导致数据传输终止。当I2C在Slave-Transmitter模式下正在接收地址字节或者Slave-Receiver模式下，I2C正在接收地址和数据字节时，cpu对IC_ENABLE寄存器写0，会导致此位置1。（不支持I2C作为slave）
[0]	RO	IC_EN	0x0	表示I2C是否使能。 0: I2C处于非活跃状态 1: I2C使能状态

### 5.1.6.27 IC\_START

偏移地址：0xa0

复位方式：H/S

Bits	Access	Name	Default	Description
[15:1]	-	reserved	-	保留
[0]	RW	START	0x0	控制I2C开始发送数据。此寄存器要在IC_ENABLE寄存器使能之后配。当fifo有数据后，此寄存器写1，数据才会开始发送。 0: 不开始 1: 开始

## 5.2 SPI

### 5.2.1 概述

SPI控制器实现数据的串并、并串转换，可以作为Master或Slave与外部设备进行同步串行通信。

### 5.2.2 功能描述

#### 5.2.2.1 功能特性

SPI 接口具有以下特点：

- 多个中断及中断屏蔽寄存器；
- 芯片有两组SPI接口，每个SPI接口支持最大2个slave；
- 支持中断源优先级可配；
- 时钟频率可编程；
- 与APB总线时钟同步；
- 数据发送/接收16bit位宽FIFO独立，FIFO深度128（spi\_slave FIFO深度是8）；
- 支持SPI全双工工作模式、支持数据帧传输极性及相位调节；
- 提供内部自回环模式测试；
- 支持4~16bit的软件可配置的SPI传输模式；
- 支持SCLK极性和相位可配置。

### 5.2.2.2 典型应用

SPI 模块作为主设备，连接两个 slave 设备的典型应用方式如下图所示：

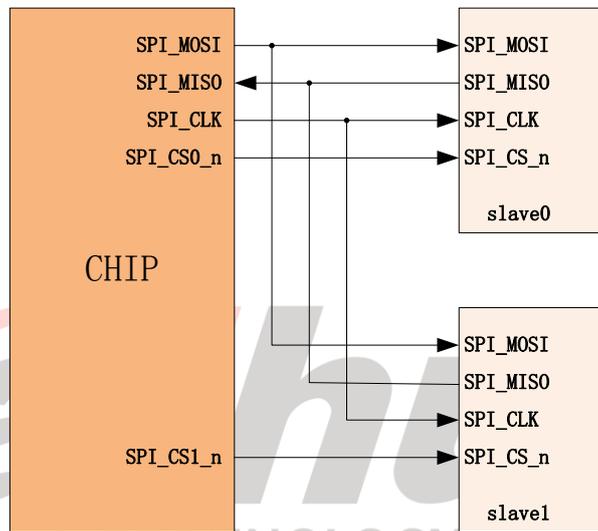


图 23 主设备 SPI 的典型连接方式

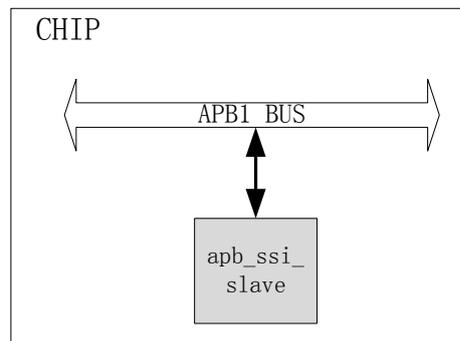


图 24 从设备 SPI 的典型连接方式

### 5.2.2.3 时钟极性和相位可配置

SPI 输出时钟 SCLK\_OUT 的极性和相位是可以通过寄存器配置（CTRL0 的 bit[7:6]），SCPH 表示相位，当 SCPH 配置为 0，表示在 SCLK\_OUT 的第一个时钟沿采样数据，为 1 表示在 SCLK\_OUT 的第二个时钟沿采样数据。SCPOL 表示 SCLK\_OUT 的极性，即 SCPOL 配置为 0，

表示 SCLK\_OUT 无效时为低电平, 为 1 表示 SCLK\_OUT 无效时为高电平。对于 SPI 对接的 SLAVE 设备来说, 同样适用。

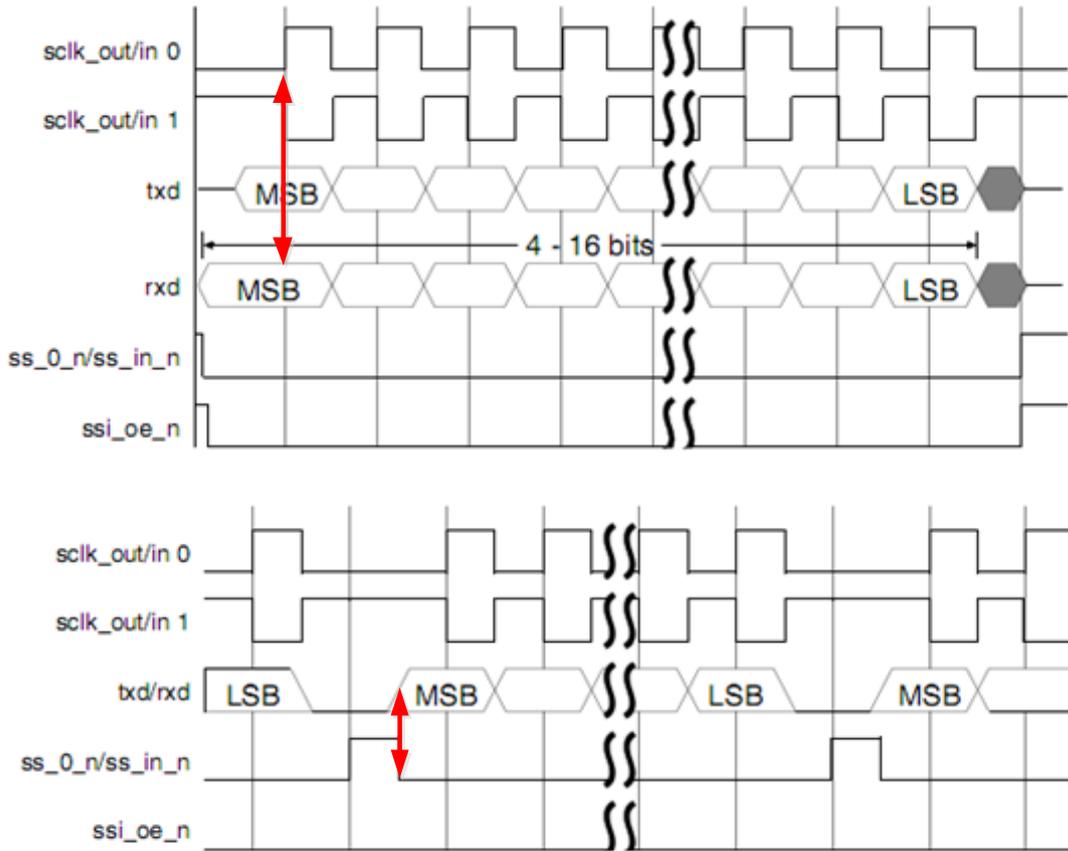
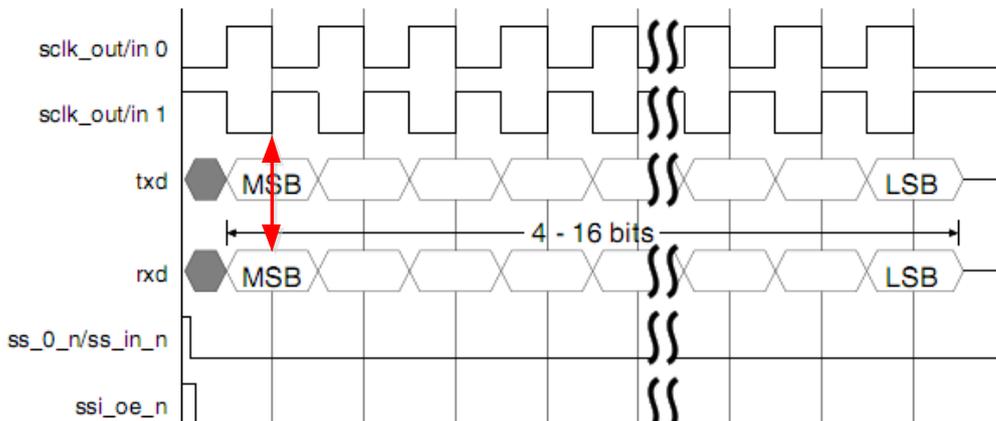


图 25 SCPH=0 数据发送和采样时序

上图中, SCPH=0, 数据采样是在 sclk\_out 的第一个边沿, 数据发送是在片选信号 ss\_0\_n 下降沿开始。



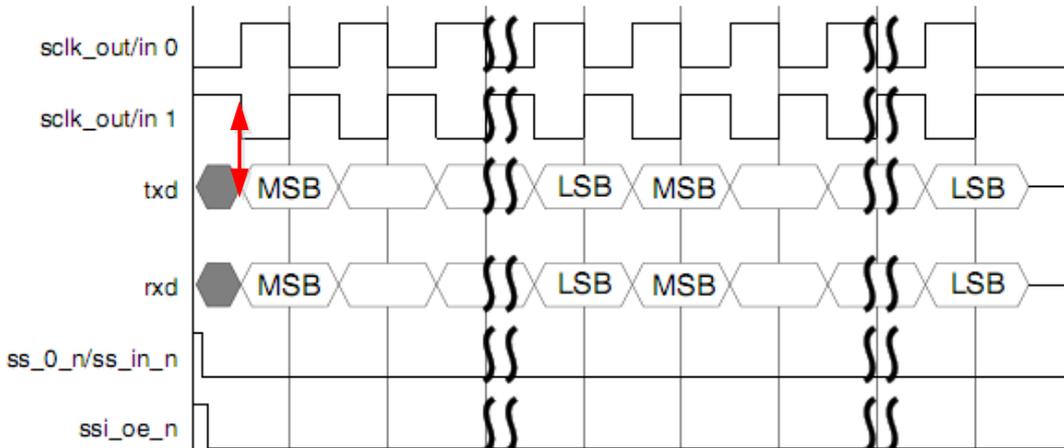


图 26 SCPH=1 数据发送和采样时序

上图中，SCPH=1，数据采样是在 sclk\_out 的第二个边沿，数据发送是在 sclk\_out 的第一个边沿。

### 5.2.3 工作方式

#### 5.2.3.1 传输时序

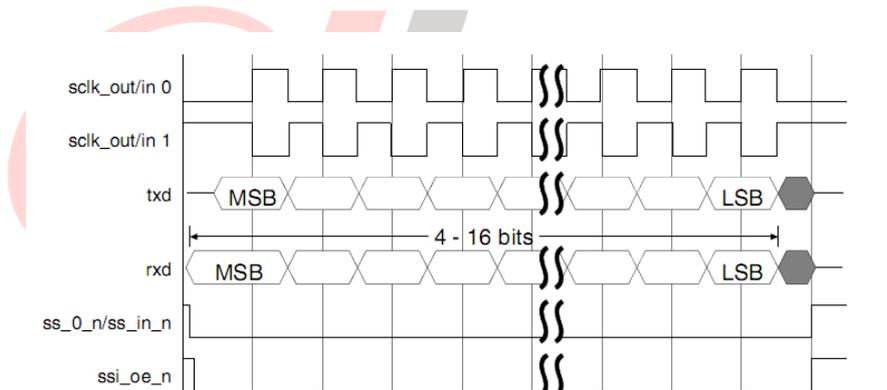


图 27 SPI 数据串行时序

串行数据在时钟下降沿生成，在时钟上升沿进行采样。

同一个命令执行完毕需要取消片选，下一条命名重新使能片选；同一条命令处理过程中取消片选会中断该命令。

为适应多种设备，该模块可设置数据串行传输顺序可以通过设置 FRF 寄存器实现：0，从 MSB 开始；1，从 LSB 开始。

#### 5.2.3.2 时钟与复位

输出 SPI 时钟 SPI\_CLK 时钟频率计算方式如下：

$$F_{\text{sclk\_out}} = F_{\text{pclk}} / \text{spi\_baudr}$$

$F_{\text{pclk}}$ ：SPI 的工作参考时钟，即 APB 总线时钟。

spi\_baudr 请查询寄存器 BAUDR。

### 5.2.3.3 操作流程

#### 初始化

初始化步骤如下：

- 1、设置 SPI\_SSIENR 为 0x0，禁用 SPI 模块；
- 2、写寄存器 SPI\_CTRLR0，配置帧格式及传输数据位宽等参数。需要接收数据时（Receive Only 和 EEPROM Read Transfer 模式），先跳至步骤 3 再跳至步骤 4，只发送数据不需要接收数据时，跳至步骤 4；
- 3、写寄存器 SPI\_CTRLR1，设置接收数据个数；
- 4、配置寄存器 SPI\_BAUDR，配置时钟分频因子；
- 5、中断方式下，设置寄存器 SPI\_IMR，使能相应的中断；查询方式下，应禁止使能相应中断；
- 6、中断方式下，设置寄存器 SPI\_TXFTLR 和 SPI\_RXFTLR。

#### 查询方式下的数据传输

具体步骤如下：

- 1、设置寄存器 SPI\_SSIENR 寄存器为 0x1，使能 SPI 模块；
- 2、设置寄存器 SPI\_SER 寄存器为 0x1，使能相应的 slave；
- 3、将需要发送的数据连续写到寄存器 SPI\_DR；
- 4、轮询寄存器 SPI\_SR，直到 bit0[busy]=0（表示不忙）、[TFE]=1（表示发送 FIFO 已空）、[RFNE]=1（表示接收 FIFO 非空），进入步骤 6；
- 5、读书数据，需保证读空接收 FIFO；
- 6、配置寄存器 SPI\_SSIENR 寄存器为 0x0，禁用 SPI；
- 7、配置寄存器 SPI\_SER 寄存器为 0x0，禁止与 slave 通信。

#### 中断方式下的数据传输

- 1、设置寄存器 SPI\_SSIENR 寄存器为 0x1，使能 SPI 模块；
- 2、设置寄存器 SPI\_SER 寄存器为 0x1，使能相应的 slave；
- 3、将需要发送的数据连续写到寄存器 SPI\_DR；
- 4、等待中断 RXFIS（接收 FIFO 数据超过 SPI\_RXFTLR 设置值），读出数据，循环直到读出所有数据；
- 5、配置寄存器 SPI\_SSIENR 寄存器为 0x0，禁用 SPI；
- 6、配置寄存器 SPI\_SER 寄存器为 0x0，禁止与 slave 通信。

### 5.2.3.4 SPI 模块 4 种传输模式

#### (1) Transmit & Receive 模式

Transmit & Receive 模式下，SPI 模块发送数据的同时接收数据，发送数据个数等于接收数据个数。

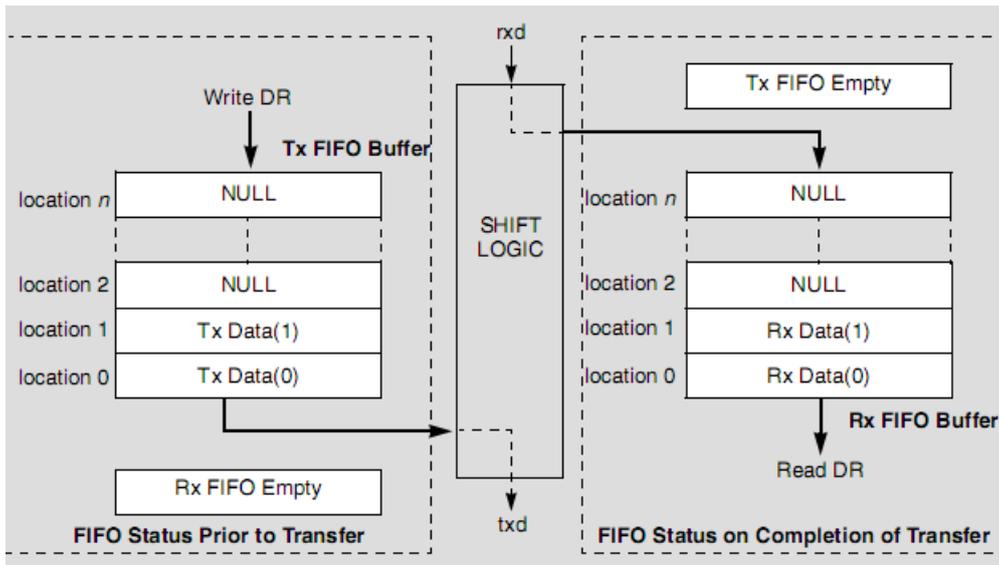


图 28 SPI Transmit & Receive 模式

(2) Transmit Only 模式

Transmit Only 模式下，SPI 模块只进行数据发送；对于接收的数据经过移位寄存器后不放入接收 FIFO 中。

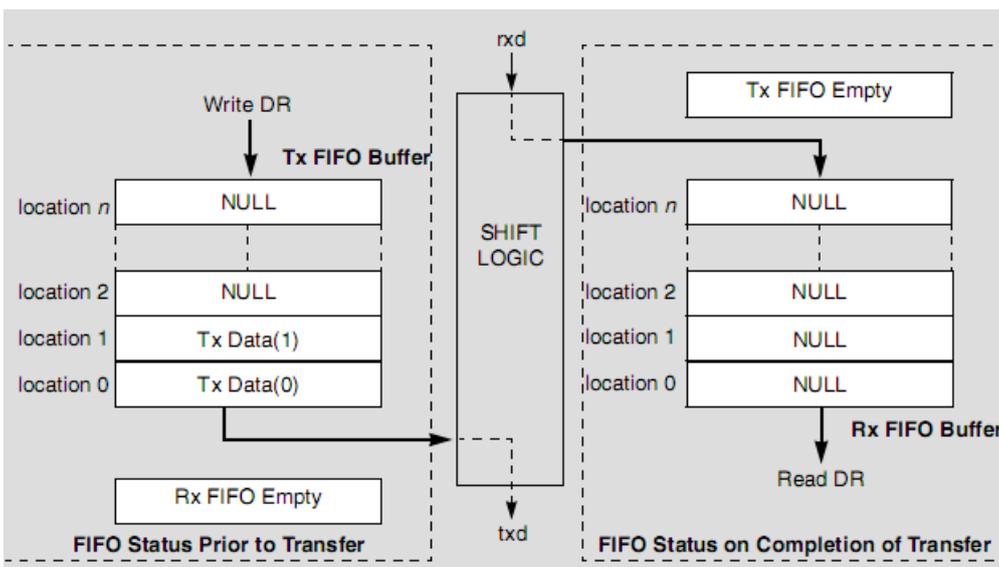


图 29 SPI Transmit Only 模式

(3) Receive Only 模式

Receive Only 模式下，SPI 模块只进行数据接收；但是为了启动数据接收，必须要在发送 FIFO 中随意写入一个数据，这个数据会被重复发送出去。

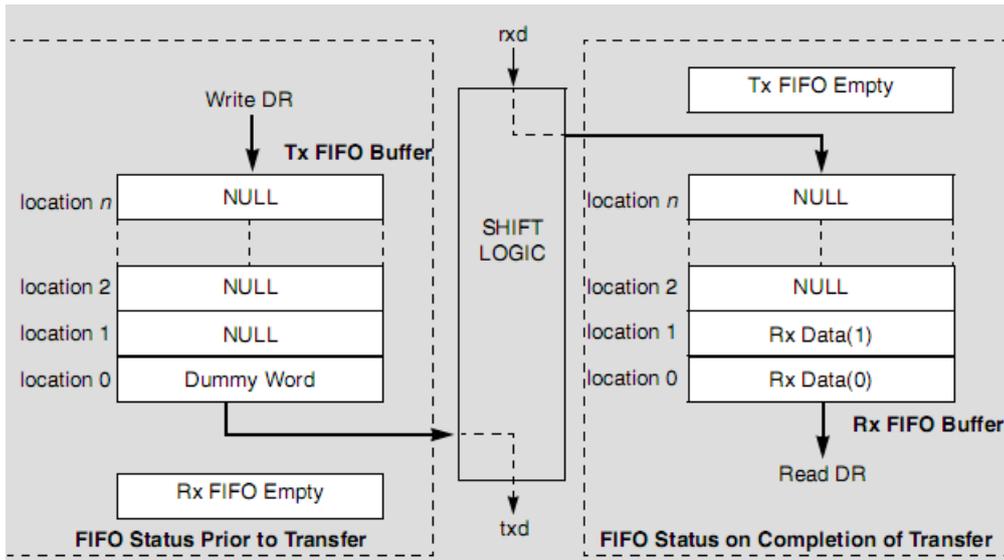


图 30 SPI Receive Only 模式

(4) EEPROM Read Transfer 模式（仅限于主设备 SPI）

EEPROM Read Transfer 模式下，SPI 模块先将发送 FIFO 中的数据发送出去后，再进行数据的接收；发送数据过程中不接收数据。注意必须要先将发送数据填入 FIFO 中后，在片选使能相应 slave 设备。

注意：EEPROM Read Transfer 模式下，命令长度至少为两拍，即发送命令至少需要写 FIFO 两次；若只有一次，则读回的数据的第一拍（frame 宽度为单位）不会写入到 FIFO 中。

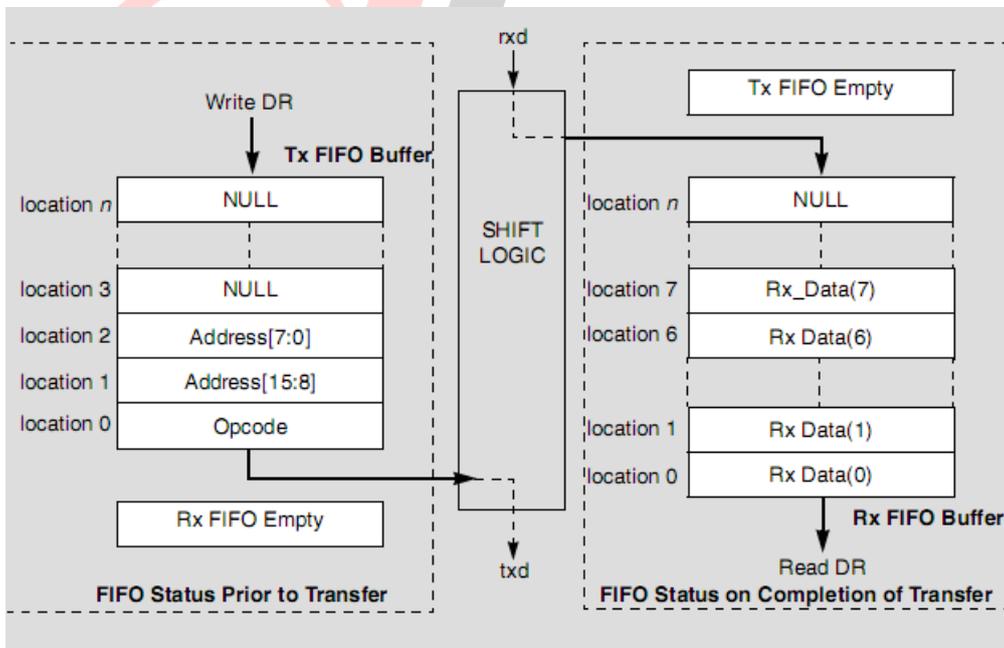


图 31 SPI EEPROM Read Transfer 模式

5.2.4 寄存器概述

寄存器表 11 SPI0 寄存器概述（基地址：0x140A\_0000）

偏移地址	名称	描述	页码
------	----	----	----

0x00	CTRLR0	传输方式及帧长度配置	
0x04	CTRLR1	接收数据帧个数	
0x08	SSIENR	SPI 模块使能	
0x10	SER	slave 片选使能	
0x14	BAUDR	时钟使能及分频选择	
0x18	TXFTLR	发送 FIFO 门限值	
0x1C	RXFTLR	接收 FIFO 门限值	
0x20	TXFLR	发送 FIFO 中数据个数	
0x24	RXFLR	接收 FIFO 中数据个数	
0x28	SR	SPI 模块状态	
0x2C	IMR	中断屏蔽寄存器	
0x30	ISR	被屏蔽的中断寄存器	
0x34	RISR	未被屏蔽的中断寄存器	
0x38	TXOICR	发送 FIFO overflow 中断清除	
0x3C	RXOICR	接收 FIFO overflow 中断清除	
0x40	RXUICR	接收 FIFO underflow 中断清除	
0x48	ICR	清除 ssi_txo_intr、ssi_rxu_intr、ssi_rxo_intr 中断	
0x58	IDR	模块 ID 寄存器	
0x5C	SSI_COMP_VERSION	版本号	
0x60	DR	数据发送/接收 FIFO 接口	

寄存器表 12 SPI1 寄存器概述（基地址：0x140B\_0000）

偏移地址	名称	描述	页码
0x00	CTRLR0	传输方式及帧长度配置	
0x04	CTRLR1	接收数据帧个数	
0x08	SSIENR	SPI 模块使能	
0x10	SER	slave 片选使能	
0x14	BAUDR	时钟使能及分频选择	
0x18	TXFTLR	发送 FIFO 门限值	
0x1C	RXFTLR	接收 FIFO 门限值	
0x20	TXFLR	发送 FIFO 中数据个数	
0x24	RXFLR	接收 FIFO 中数据个数	
0x28	SR	SPI 模块状态	
0x2C	IMR	中断屏蔽寄存器	
0x30	ISR	被屏蔽的中断寄存器	
0x34	RISR	未被屏蔽的中断寄存器	
0x38	TXOICR	发送 FIFO overflow 中断清除	
0x3C	RXOICR	接收 FIFO overflow 中断清除	
0x40	RXUICR	接收 FIFO underflow 中断清除	
0x48	ICR	清除 ssi_txo_intr、ssi_rxu_intr、ssi_rxo_intr 中断	
0x58	IDR	模块 ID 寄存器	
0x5C	SSI_COMP_VERSION	版本号	

0x60	DR	数据发送/接收 FIFO 接口	
------	----	-----------------	--

寄存器表 13 SPI2 寄存器概述（基地址：0x140E\_0000）

偏移地址	名称	描述	页码
0x00	CTRLR0	传输方式及帧长度配置	
0x04	CTRLR1	接收数据帧个数	
0x08	SSIENR	SPI 模块使能	
0x10	SER	slave 片选使能	
0x14	BAUDR	时钟使能及分频选择	
0x18	TXFTLR	发送 FIFO 门限值	
0x1C	RXFTLR	接收 FIFO 门限值	
0x20	TXFLR	发送 FIFO 中数据个数	
0x24	RXFLR	接收 FIFO 中数据个数	
0x28	SR	SPI 模块状态	
0x2C	IMR	中断屏蔽寄存器	
0x30	ISR	被屏蔽的中断寄存器	
0x34	RISR	未被屏蔽的中断寄存器	
0x38	TXOICR	发送 FIFO overflow 中断清除	
0x3C	RXOICR	接收 FIFO overflow 中断清除	
0x40	RXUICR	接收 FIFO underflow 中断清除	
0x48	ICR	清除 ssi_txo_intr、ssi_rxu_intr、ssi_rxo_intr 中断	
0x58	IDR	模块 ID 寄存器	
0x5C	SSI_COMP_VERSION	版本号	
0x60	DR	数据发送/接收 FIFO 接口	

## 5.2.5 寄存器描述

### 5.2.5.1 CTRLR0

偏移地址：0x00

复位方式：H/S

Bits	Access	Name	Default	Description
[31:12]	RO	reserved	0x0	预留
[11]	RW	SRL	0x0	自回环测试模式
[10]	RW	eENDING（作为 master）	0x0	大小端控制，1：大端；0：小端
[10]	RW	SLV_OE（作为 slave）	0x0	slave的txd是否使能。0：enable；1：disable
[9:8]	RW	TMOD	0x0	传输模式： 00-Transmit & Receive 01-Transmit Only 10-Receive Only 11-EEPROM Read（仅限于master SPI）

[7]	RW	SCPOL	0x0	时钟极性，表示时钟无效时的电平状态。0：SCLK无效时为低电平状态；1：CLK无效时为高电平状态。
[6]	RW	SCPH	0x0	时钟相位。0：表示在SCLK的第一个边沿采样数据；1：在SCLK的第二个边沿采样数据。
[5]	RW	M2L	0x0	选择发送顺序：0，从MSB开始；1，从LSB开始
[4:0]	RW	DFS	0x7	数据帧长度
[5: 4]	RW	FRF (slave)	0x0	选择串行协议。00：Motorola SPI（固定为SPI）
[3:0]	RW	DFS (slave)	0x7	数据帧长度

#### 5.2.5.2 CTRLR1 (仅用于master)

偏移地址：0x04

复位方式：H/S

Bits	Access	Name	Default	Description
[31:16]	RO	reserved	0x0	预留
[15:0]	RW	NDF	0x0	接收数据次数

#### 5.2.5.3 SSIENR

偏移地址：0x08

复位方式：H/S

Bits	Access	Name	Default	Description
[31:1]	RO	reserved	-	预留
[0]	RW	SSI_EN	0x0	spi模块使能，为1时禁止配置寄存器，为0时清空FIFO及状态寄存器

#### 5.2.5.4 SER (仅用于master)

偏移地址：0x10

复位方式：H/S

Bits	Access	Name	Default	Description
[31:2]	RO	reserved	0x0	预留
[1:0]	RW	SER	0x0	slave片选

#### 5.2.5.5 BAUDR (仅用于master)

偏移地址：0x14

复位方式：H/S

Bits	Access	Name	Default	Description
[31:16]	RO	reserved	-	预留
[15:0]	RW	SCKDV	0x0	输出时钟分频，只允许偶数分频，为0时禁用时钟

### 5.2.5.6 TXFTLR

偏移地址：0x18

复位方式：H/S

Bits	Access	Name	Default	Description
[31:8]	RO	reserved	-	预留
[7:0]	RW	TFT	0x0	发送数据FIFO门限值

### 5.2.5.7 RXFTLR

偏移地址：0x1c

复位方式：H/S

Bits	Access	Name	Default	Description
[31:8]	RO	reserved	0x0	预留
[7:0]	RW	RFT	0x0	接收数据FIFO门限值

### 5.2.5.8 TXFLR

偏移地址：0x20

复位方式：H/S

Bits	Access	Name	Default	Description
[31:8]	RO	reserved	0x0	预留
[7:0]	RO	TXTFL	0x0	发送数据FIFO中数据个数

### 5.2.5.9 RXFLR

偏移地址：0x24

复位方式：H/S

Bits	Access	Name	Default	Description
[31:8]	RO	reserved	0x0	预留
[7:0]	RO	RXTFL	0x0	接收数据FIFO中数据个数

### 5.2.5.10 SR

偏移地址：0x28

复位方式：H/S

Bits	Access	Name	Default	Description
[31:5]	RO	reserved	0x0	预留
[4]	RO	RFF	0x0	接收fifo满。0：不满；1：满
[3]	RO	RFNE	0x0	接收FIFO非空。0：空；1：不空
[2]	RO	TFE	0x1	发送FIFO空。0：不空；1：空
[1]	RO	TFNF	0x1	发送FIFO非满。0：满；1：不满
[0]	RO	BUSY	0x0	正在传输数据

### 5.2.5.11 IMR

偏移地址：0x2c

复位方式：H/S

Bits	Access	Name	Default	Description
[31:6]	RO	reserved	0x0	预留
[5]	RW	reserved	0x1	预留
[4]	RW	RXFIM	0x1	接收数据超过门限值中断屏蔽
[3]	RW	RXOIM	0x1	接收FIFO overflow中断屏蔽
[2]	RW	RXUIM	0x1	接收FIFO underflow中断屏蔽
[1]	RW	TXOIM	0x1	发送FIFO overflow中断屏蔽
[0]	RW	TXEIM	0x1	发送FIFO 空中断屏蔽

### 5.2.5.12 ISR

偏移地址：0x30

复位方式：H/S

Bits	Access	Name	Default	Description
[31:5]	RO	reserved	0x0	预留
[4]	RO	RXFIS	0x0	屏蔽的接收数据超过门限值中断
[3]	RO	RXOIS	0x0	屏蔽的接收FIFO overflow中断
[2]	RO	RXUIS	0x0	屏蔽的接收FIFO underflow中断
[1]	RO	TXOIS	0x0	屏蔽的发送FIFO overflow中断
[0]	RO	TXEIS	0x0	屏蔽的发送FIFO 空中断

### 5.2.5.13 RISR

偏移地址：0x34

复位方式：H/S

Bits	Access	Name	Default	Description
[31:5]	RO	reserved	0x0	预留
[4]	RO	RXFIR	0x0	未屏蔽的接收数据超过门限值中断
[3]	RO	RXOIR	0x0	未屏蔽的接收FIFO overflow中断
[2]	RO	RXUIR	0x0	未屏蔽的接收FIFO underflow中断
[1]	RO	TXOIR	0x0	未屏蔽的发送FIFO overflow中断
[0]	RO	TXEIR	0x0	未屏蔽的发送FIFO 空中断

### 5.2.5.14 TXOICR

偏移地址：0x38

复位方式：H/S

Bits	Access	Name	Default	Description
[31:1]	RO	reserved	0x0	预留
[0]	RO	TXOICR	0x0	清除发送FIFO overflow中断

### 5.2.5.15 RXOICR

偏移地址: 0x3c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	RO	reserved	0x0	预留
[0]	RO	RXOICR	0x0	清除接收FIFO overflow中断

### 5.2.5.16 RXUICR

偏移地址: 0x40

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	RO	reserved	0x0	预留
[0]	RO	RXUICR	0x0	清除接收FIFO underflow中断

### 5.2.5.17 ICR

偏移地址: 0x48

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	RO	reserved	0x0	预留
[0]	RO	ICR	0x0	清除发送FIFO overflow中断、接收FIFO overflow中断、接收FIFO underflow中断

### 5.2.5.18 IDR

偏移地址: 0x58

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	IDR	0x1122334 4	模块ID

### 5.2.5.19 SSI\_COMP\_VERSION

偏移地址: 0x5c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	SSI_COMP_VERSION	0x3331312 a	模块版本号

### 5.2.5.20 DR

偏移地址: 0x60

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	DR	0x0	FIFO读/写地址

### 5.3 I80

#### 5.3.1 概述

HR\_C7000 内置一 Intel I80 接口控制器，用于与 LCD 等器件接口。

#### 5.3.2 功能描述

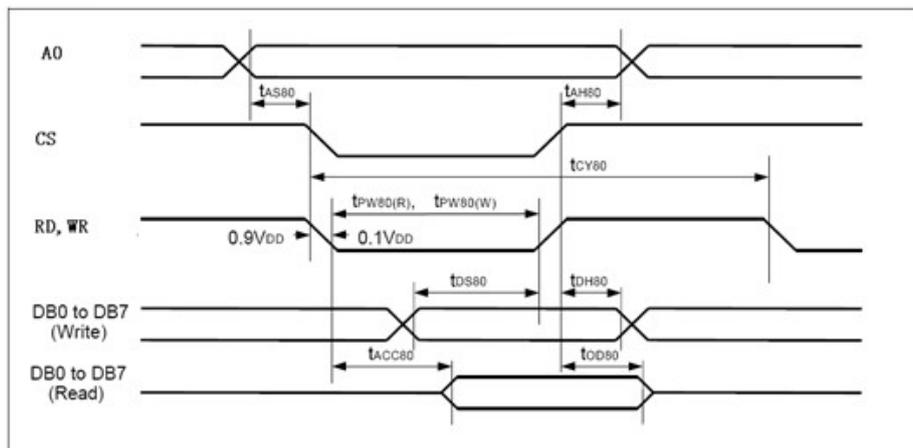
I80 控制器在芯片内连接到系统 AHB 总线，在芯片外连接到外部 I80 接口器件。

#### 5.3.3 工作方式

I80 控制器自动将对其寄存器的操作转换为符合 I80 总线要求的时序。I80 控制器内部有多种配置寄存器，可以实现 I80 各种时序参数和电平极性的灵活配置。

#### 8080并行总线接口

当PS为高电平、C86为低电平时，LCD模块的总线接口为8080并行总线接口，LCD的时序图如下：



当LCD模块处于8080总线模式时，原LCD模块上的EP引脚将作为8080总线的RD（读使能信号线）。

(VDD = 2.4 to 3.6V, Ta = -40 to +85°C)

Item	Signal	Symbol	Min.	Typ.	Max.	Unit	Remark
Address setup time	RS	tAS80	0	-	-	ns	
Address hold time	RS	tAH80	0	-	-	ns	
System cycle time	RS	tCY80	300	-	-	ns	
Pulse width (WRB)	RW_WRB	tPW80(W)	60	-	-	ns	
Pulse width (RDB)	E_RDB	tPW80(R)	60	-	-	ns	
Data setup time	DB7 to DB0	tDS80	40	-	-	ns	
Data hold time		tDH80	15	-	-	ns	
Read access time	DB0	tACC80	-	-	140	ns	CL = 100 pF
Output disable time		tOD80	10	-	100	ns	

图 32 8080 接口时序

#### 5.3.4 寄存器概述

寄存器表 14 I80 寄存器概述（基地址：0x1200\_0000）

偏移地址	名称	描述	页码
0x0	INDEX	寻址寄存器	

0x4	DATA	数据寄存器	
0x10	WCFG	写时序参数配置	
0x14	RCFG	读时序参数配置	
0x20	SCFG	配置寄存器	
0x2c	AC_MODE	总线模式寄存器	

### 5.3.5 寄存器描述

#### 5.3.5.1 INDEX

偏移地址: 0x00

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	INDEX0	0x0	端口Index寄存器

#### 5.3.5.2 DATA

偏移地址: 0x04

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	DATA0	0x0	端口数据寄存器

#### 5.3.5.3 WCFG

偏移地址: 0x10

复位方式: H/S

Bits	Access	Name	Default	Description
[31:24]	RO	reserved	0x0	保留
[23:16]	RW	TCNT_WHT0	0x1	写选通尾部无效周期
[15:8]	RW	TCNT_WL0	0x1	写选通有效周期
[7:0]	RW	TCNT_WHF0	0x1	写选通头部无效周期

#### 5.3.5.4 RCFG

偏移地址: 0x14

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	reserved	0x0	保留
[23:16]	RW	TCNT_RHT0	0x1	读选通尾部无效周期
[15:8]	RW	TCNT_RL0	0x1	读选通有效周期
[7:0]	RW	RCNT_RHF0	0x1	读选通头部无效周期

### 5.3.5.5 SCFG

偏移地址：0x20

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	P0_SCFG	0x0	端口配置寄存器

### 5.3.5.6 AC\_MODE

偏移地址：0x2c

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	BUSHS	0x0	总线模式控制寄存器

## 5.4 UART

### 5.4.1 概述

通用异步收发器UART（Universal Asynchronous Receiver Transmitter）是一个异步串行的通信接口，主要功能是将来自外围设备的数据进行串并转换之后传入内部总线，以及将数据进行并串转换之后输出到外部设备。UART 的主要功能是和外芯片的UART 进行对接，从而实现两芯片间的通信。HR\_C7000提供3个UART单元，默认RXD和TXD上下拉使能关闭，当不使用Uart功能时，建议软件配置打开上下拉使能寄存器，并在板级端口上将RXD固定为高电平或者低电平。

### 5.4.2 功能描述

UART 的一次帧传输主要包括起始信号、数据、校验位和结束信号，如功能描述所示。数据帧从某一UART 的TXD 端输出，从另一个UART 的RXD 端输入。

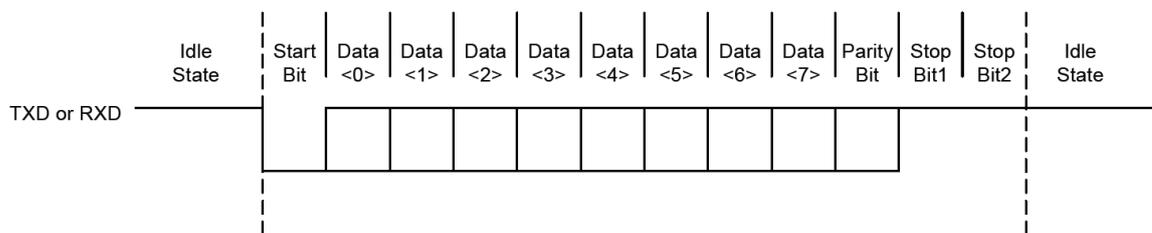


图 33 UART 帧格式

起始信号、数据、校验位和结束信号的含义如下：

- 起始信号（start bit）  
一个数据帧开始的标志，UART 协议规定TXD 信号出现一个低电平就表示一个数据帧的开始。在UART 不传输数据时，应该保持高电平。
- 数据信号（data bit）  
数据位宽可以根据不同的应用要求进行调整，可以配置成5bit/6bit/7bit/8bit 数据位宽。
- 校验位（parity bit）

校验位是1 比特纠错信号，UART 的校验位有奇校验、偶校验和固定校验位，同时支持校验位的使能和禁止。

- 结束信号（stop bit）

结束信号即数据帧的停止位，支持1 比特和2 比特停止位两种配置。数据帧的结束信号就是把TXD 拉成高电平。

### 5.4.3 工作方式

#### 波特率配置

波特率计算公式：当前波特率 = UART 参考时钟频率（APB 总线时钟）/（16 x 分频系数）。

分频系数通过寄存器 UART\_DLL 和 UART\_DLH 配置，UART\_DLL 配置低 8 位，UART\_DLH 配置高 8 位。

例如，配置波特率为 9600，工作时钟 pclk=25M，通过以上公式计算系数是十六进制 A2，即 UART\_DLL 配置为 0xa2，UART\_DLH 配置为 0x0。

#### 初始化：

- 1、配置中断使能寄存器 UART\_IER，屏蔽所有中断；
- 2、配置波特率除数锁存使能 UART\_LCR[7]、停止位个数 UART\_LCR[2]和数据位个数 UART\_LCR[1:0]；
- 3、写寄存器 UART\_DLL、UART\_DLH，配置波特率除数；
- 4、配置完波特率后，清除 UART\_LCR[7]，即写 UART\_LCR[7]=0；
- 5、配置寄存器 UART\_FCR。写 UART\_FCR[0]=1，使能 FIFO；写 UART\_FCR[5: 4]和 UART\_FCR[7: 6]分别配置发送 FIFO 和接收 FIFO 阈值。

#### 数据发送步骤：

- 1、将发送数据写入寄存器 UART\_THR，启动数据发送。
- 2、查询方式，轮询状态位 UART\_LSR[5]，查看 FIFO 是否满，决定是否继续向 FIFO 继续发送数据。中断方式，当接收到中断时，读中断 ID 寄存器 UART\_IIR，处理读到的中断（读到的中断即是最高优先级的），检测是否挂起发送 FIFO 阈值空中断，如果挂起，则继续往发送 FIFO 写数据，同时查询寄存器 UART\_LSR[5]，判断发送 FIFO 是否满，直到发送 FIFO 满，则停止继续写数据。  
中断方式数据发送步骤见如下流程图。

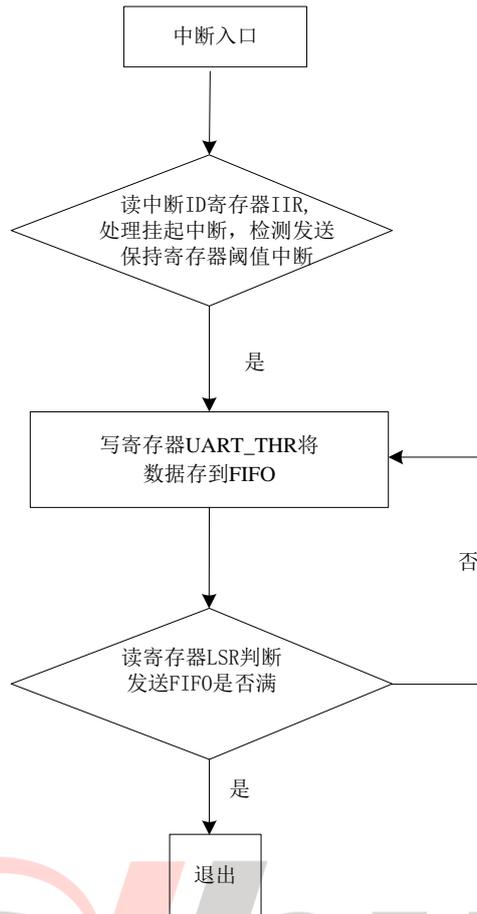


图 34 中断方式数据发送流程图

**数据接收步骤:**

查询方式，进行数据接收时，通过轮询 UART\_LSR[0]检测接收 FIFO 是否有数据准备好，有可用数据时，将数据读走。中断方式，当有中断挂起时，读寄存器 UART\_IIR，处理优先级最高的中断，检测是够有接收数据可用中断，若有，则读走接收 FIFO 数据，并读 UART\_LSR[0]，判断将 FIFO 读空。中断方式数据接收过程如下图所示。

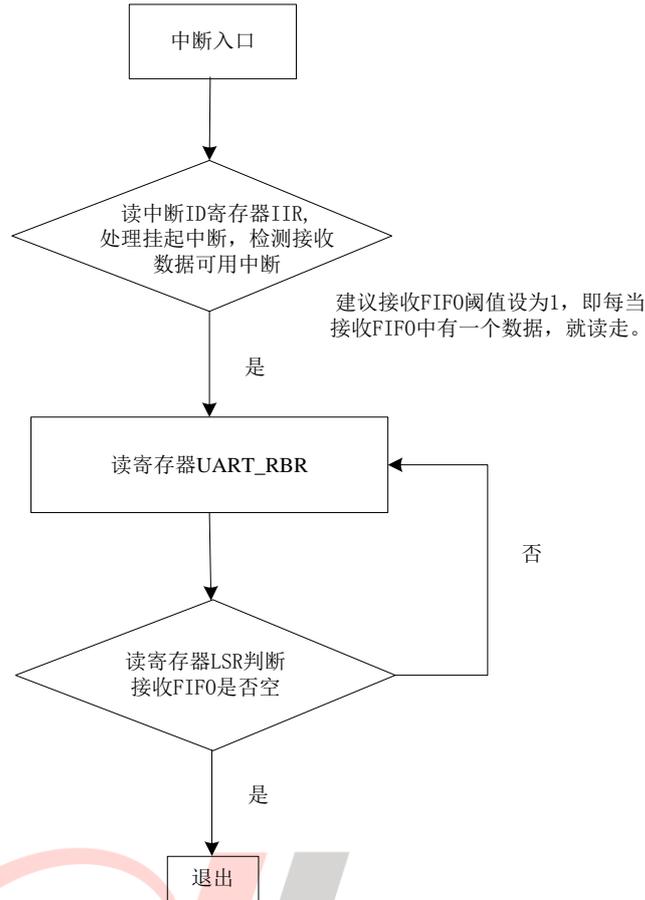


图 35 中断方式数据接收流程图

#### 5.4.4 寄存器概述

寄存器表 15 UART0 寄存器概述 (基地址: 0x1403\_0000)

偏移地址	名称	描述	页码
RBR	0x00	接收缓存寄存器	
THR	0x00	发送保持寄存器	
DLL	0x00	分频锁存低位	
DLH	0x04	分频锁存高位	
IER	0x04	中断使能寄存器	
IIR	0x08	中断标识寄存器	
FCR	0x08	FIFO 控制器寄存器	
LCR	0x0c	线控制寄存器	
MCR	0x10	Modem 控制寄存器	
LSR	0x14	线状态寄存器	
USR	0x7c	FIFO 状态标志	
TFL	0x80	发送 FIFO 中的数据数量	
RFL		接收 FIFO 中的数据数量。	

寄存器表 16 UART1 寄存器概述（基地址：0x1404\_0000）

偏移地址	名称	描述	页码
RBR	0x00	接收缓存寄存器	
THR	0x00	发送保持寄存器	
DLL	0x00	分频锁存低位	
DLH	0x04	分频锁存高位	
IER	0x04	中断使能寄存器	
IIR	0x08	中断标识寄存器	
FCR	0x08	FIFO 控制器寄存器	
LCR	0x0c	线控制寄存器	
MCR	0x10	Modem 控制寄存器	
LSR	0x14	线状态寄存器	
USR	0x7c	FIFO 状态标志	
TFL	0x80	发送 FIFO 中的数据数量	
RFL		接收 FIFO 中的数据数量。	

寄存器表 17 UART2 寄存器概述（基地址：0x1405\_0000）

偏移地址	名称	描述	页码
RBR	0x00	接收缓存寄存器	
THR	0x00	发送保持寄存器	
DLL	0x00	分频锁存低位	
DLH	0x04	分频锁存高位	
IER	0x04	中断使能寄存器	
IIR	0x08	中断标识寄存器	
FCR	0x08	FIFO 控制器寄存器	
LCR	0x0c	线控制寄存器	
MCR	0x10	Modem 控制寄存器	
LSR	0x14	线状态寄存器	
USR	0x7c	FIFO 状态标志	
TFL	0x80	发送 FIFO 中的数据数量	
RFL		接收 FIFO 中的数据数量。	

#### 5.4.5 寄存器描述

##### 5.4.5.1 RBR

偏移地址：0x00

复位方式：H

Bits	Access	Name	Default	Description
[31:8]	-	reserved	-	保留
[7:0]	RO	receive buffer	0x0	接收串行输入端口的数据

#### 5.4.5.2 THR

偏移地址: 0x00

复位方式: H

Bits	Access	Name	Default	Description
[31:8]	-	reserved	-	保留
[7:0]	WO	Transmit Holding Register	0x0	发送到串口的数据

#### 5.4.5.3 DLL

偏移地址: 0x00

复位方式: H

Bits	Access	Name	Default	Description
[31:8]	-	reserved	-	保留
[7:0]	RW	Divisor Latch (Low)	0x0	波特率除数低8位 (前提条件LCR[7]=1'b1)

#### 5.4.5.4 DLH

偏移地址: 0x04

复位方式: H

Bits	Access	Name	Default	Description
[31:8]	-	reserved	-	保留
[7:0]	RW	Divisor Latch (High)	0x0	波特率除数高8位 (前提条件LCR[7]=1'b1)

#### 5.4.5.5 IER

偏移地址: 0x04

复位方式: H

Bits	Access	Name	Default	Description
[31:8]	-	reserved	-	保留
[7]	RW	PTIME	0x0	用于enable/disableTHRE中断的产生。 0 = disabled 1 = enabled
[6:4]	-	Reserved		保留
[3]	RW	EDSSI	0x0	使能modem状态中断。 0 = disabled 1 = enabled
[2]	RW	ELSI	0x0	使能接收器线性状态中断 0 = disabled 1 = enabled

[1]	RW	ETBEI	0x0	使能发送保持寄存器空中断 0 = disabled 1 = enabled
[0]	RW	ERBFI	0x0	使能接收数据可用中断和字符超时中断（FIFO模式） 0 = disabled 1 = enabled

#### 5.4.5.6 IIR

偏移地址：0x08

复位方式：H

Bits	Access	Name	Default	Description
[31:8]	-	reserved	-	保留
[7:6]	RO	FIFOs Enabled	0x0	用于FIFO使能 00 = disabled 11 = enabled
[5:4]	-	reserved		保留
[3:0]	RO	Interrupt ID	0x01	中断ID。指示最高优先级待处理中断 0001 = no interrupt pending 0010 = THR empty 0100 = received data available 0110 = receiver line status 0111 = busy detect（当兼容16550标准时，此中断无） 1100 = character timeout

中断 ID				中断源和中断复位			
bit3	bit2	bit1	bit0	优先级	中断类型	中断源	中断复位
0	0	0	1	-	无	无	-
0	1	1	0	最高	receiver line status。	上溢/奇偶校验/帧错误（停止位不为1）。或间隔中断（rx 信号一直为0）。	读 LSR 寄存器。
0	1	0	0	第二	received data available。	接收 FIFO 数据大于等于阈值。	接收 FIFO 的数据小于门限值。
1	1	0	0	第二	character timeout。	接收 FIFO 有数据，且在设定时间内无数据进出接收 FIFO。	读寄存器 RBR。
0	0	1	0	第三	THR empty。	发送 FIFO 数据小于等于阈值。	读寄存器 IIR 或者发送 FIFO 数据大

							于阈值。
--	--	--	--	--	--	--	------

#### 5.4.5.7 FCR

偏移地址：0x08

复位方式：H

Bits	Access	Name	Default	Description
[31:8]	-	reserved	-	保留
[7:6]	WO	RCVR Trigger	0x0	设置产生接收数据可用中断的阈值。 当大于等于阈值时，产生中断，标志接收FIFO有数据可用。 00: 1; 01: 接收FIFO的1/4深度，（即设为4）； 10: 接收FIFO的1/2深度FIFO，（即设为8）； 11: 差2个字符达到接收FIFO的深度，（即设为14）
[5:4]	WO	TX Empty Trigger	0x0	设置产生TX空中断的阈值。 当小于等于此阈值时，会产生阈值中断，同时也会产生DMA请求。 00: FIFO 空； 01: FIFO中有2个字符； 10: FIFO的1/4深度，（即设为4）； 11: FIFO 的1/2深度，（即设为8）。
[3]	WO	DMA Mode	0x0	DMA模式选择。 0: 模式0； 1: 模式1。
[2]	WO	XMIT FIFO Reset	0x0	XMIT FIFO复位。可以复位发送FIFO的控制逻辑，将FIFO视为空
[1]	WO	RCVR FIFO Reset	0x0	RCVR FIFO复位。可以复位接收FIFO的控制逻辑，将FIFO视为空
[0]	WO	FIFO Enable	0x0	使能/关断发送FIFO和接收FIFO。

#### 5.4.5.8 LCR

偏移地址：0x0C

复位方式：H

Bits	Access	Name	Default	Description
[31:8]	-	reserved	-	保留
[7]	RW	DLAB	0x0	波特率除数锁存位。此比特位用于使能UART_DLL和UART_DLH寄存器配置波特率。初始化波特率后，此位必须清空
[6]	RW	Break	0x0	间隔产生。当置为1时，表示串口接口无串行数据传输
[5]	-	Stick Parit		保留，视为0
[4]	RW	EPS	0x0	偶校验选择。用于选择奇偶校验选择。 置1，偶校验；

				置0，奇校验。
[3]	RW	PEN	0x0	用于使能校验位。 0 = parity disabled 1 = parity enabled
[2]	RW	STOP	0x0	用于选择每个字符的stop位个数。 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) =0, 否则选择2 stop bit 无论选择几个stop位，接收器只检查第一个stop位。
[1:0]	RW	DLS	0x0	数据长度选择。用于选择每个字符的数据位的个数。 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits

#### 5.4.5.9 MCR

偏移地址：0x10

复位方式：H

Bits	Access	Name	Default	Description
[31:7]	-	reserved	-	保留
[6]	RW	SIRE	0x0	SIR模式使能
[5]	RW	AFCE	0x0	自动流控
[4]	RW	LoopBack	0x0	环路模式
[3]	RW	OUT2	0x0	OUT2
[2]	RW	OUT1	0x0	OUT1
[1]	RW	RTS	0x0	RTS
[0]	RW	DTR	0x0	DTR

#### 5.4.5.10 LSR

偏移地址：0x14

复位方式：H

Bits	Access	Name	Default	Description
[31:8]	-	reserved		保留
[7]	RO	RFE	0x0	接收FIFO错误位。当FIFO中至少有一种奇偶校验错误、帧错误、中断错误时置1。 0: 无错误; 1: 有错误
[6]	RO	TEMT	0x1	发送器空标志位。FIFO模式下，当发送移位寄存器和FIFO都空了时，会置位此比特位。

[5]	RO	THRE	0x1	发送保持寄存器空标志位。当UART_IER[7]=1时，表示TX FIFO满；当UART_IER[7]=0时，表示TX FIFO空；
[4]	RO	BI	0x0	间隔中断。用于指示串行输入数据中的中断序列。 UART模式下，当串行输入sin 保持为0的时间超过传送一个字符（start time+ data bits+ parity+ stop bits）的时间时，置位；
[3]	RO	FE	0x0	帧错误标志位。当接收器没有接收到有效的stop位时，会发生帧错误。
[2]	RO	PE	0x0	奇偶校验错误标志位。 0 = no parity error 1 = parity error
[1]	RO	OE	0x0	Overrun错误。IFO模式下，当FIFO满且接收了下一个数据时，会发生overrun错误，FIFO的数据被保留，移位寄存器的数据丢失。
[0]	RO	DR	0x0	数据准备位。表示，RBR或者接收FIFO中至少有一个字符。 0 = no data ready 1 = data ready

#### 5.4.5.11 USR

偏移地址：0x7c

复位方式：H/S

Bits	Access	Name	Default	Description
[31:5]	-	reserved	-	保留
4	R	RFF	0x0	接收 FIFO 满。用于表示接收 FIFO 完全满。 0 = Receive FIFO not full 1 = Receive FIFO Full
3	R	RFNE	0x0	接收 FIFO 不空。用于表示接收 FIFO 有一个或多个 entry。 0 = Receive FIFO is empty 1 = Receive FIFO is not empty
2	R	TFE	0x0	发送 FIFO 空。用于表示发送 FIFO 完全空。 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty
1	R	TFNF	0x0	发送 FIFO 不满。用于表示发送 FIFO 有一个或多个数据。 0 = Transmit FIFO is full 1 = Transmit FIFO is not full
0	R	BUSY	0x0	此位无效。

#### 5.4.5.12 TFL

偏移地址：0x80

复位方式：H/S

Bits	Access	Name	Default	Description
[31:FI FO_A DDR_ WIDT H+ 1]	-	reserved	-	保留
[FIFO_ ADDR _WID TH:0]	RO	Transmit FIFO Level	0x0	表示发送 FIFO 中的数据数量。

#### 5.4.5.13 RFL

偏移地址：0x84

复位方式：H/S

Bits	Access	Name	Default	Description
[31:FI FO_A DDR_ WIDT H+ 1]	-	reserved	-	保留
[FIFO_ ADDR _WID TH:0]	R	Receive FIFO Level	0x0	表示接收 FIFO 中的数据数量。

## 5.5 SDIO

### 5.5.1 概述

HR\_C7000 内置 SDIO 控制器，可以实现与 SD 卡、SDIO 设备等连接。

### 5.5.2 功能描述

SD/SDIO 控制器用于对 SD/SDIO 协议设备、MMC 协议设备和 CE-ATA 协议设备的读写等操作。控制器软核具有下列功能：

支持接口类型：

- SD 卡 (Secure Digital Memory)      **version 3.0** 协议
- SDIO (Secure Digital I/O)          **version 3.0** 协议
- CE-ATA(Consumer Electronics Advanced Transport Architecture)      version 1.1 协议
- Multimedia Cards(MMC version4.41, eMMC version4.5)

支持读写模式：

- 比特流读写 (SIDO 设备)
- 单块数据读写 (主要面向小容量 SD 卡)
- 连续多块数据读写 (主要面向大容量 SD 卡)

支持数据总线模式:

- 1bit 数据模式 (SD 模式、SDIO 模式、SPI 模式)
- 4bit 数据总线 (SD 模式、SDIO 模式)

其他特点:

- 控制器可支持 1 张 SD 卡或 SDIO 设备
- 支持指令完成信号中断, 可编程屏蔽
- cclk\_in 时钟标称值 50MHz, 可配分频得到 cclk\_out, 分频倍数为 1 或 2\*n (n=1~255)
- cclk\_in 时钟 1/4 相位精度可配得到采样时钟和驱动时钟
- 支持低功耗时钟门控

### 5.5.3 工作方式

#### 5.5.3.1 总体流程

SD/SDIO Controller 从上电到正常数据通信过程的基本流程如下:

- a) 打开外部设备的电源, 设置设备检测的时钟周期数;
- b) 设置时钟分频用于设备枚举 (100KHz~400KHz), 启动时钟;
- c) 使能全局中断, 屏蔽不需要的中断, 检测设备状态;
- d) 进行设备枚举, 完成设备完成初始化;
- e) 更换时钟频率 (标准模式最高 25MHz, 高速模式最高 50MHz),
- f) 设置与应用层相关的 FIFO 门限、超时门限、中断屏蔽等寄存器;
- g) 进入数据通信操作。

注: SD 卡上电完成后, 需要维持至少 74 个 cclk\_out 时钟后, 才允许进入枚举功能, 否则存储卡设备不能正常相应。cclk\_out 时钟周期远大于系统时钟, 故软件需留足够时间。

#### 5.5.3.2 电源管理

SD/SDIO Controller 电源管理功能:

- a) 通过寄存器配置 SD 设备的电源开关;  
 $PWREN[0] = 1'b1$ ; //使能#0 卡设备电源

#### 5.5.3.3 时钟控制

更换时钟频率操作需在指令及数据通信空闲状态进行, 更换时钟频率之前需先禁用时钟, 然后再使能新频率的时钟。

- a) 禁用时钟: 清零 clk\_enable 寄存器 (CLKENA@0x10); 然后设置 start\_cmd、update\_clock\_only、wait\_data\_complete (CMD@0x2c) 各位为 1, 发送时钟控制指令; 等待 start\_cmd 清零, 表示指令已被发送。  
 $CLKENA[15:0] = 16'h0$ ; //禁用所有存储卡的工作时钟  
 $CMD[31] = 1'b1, CMD[21] = 1'b1, CMD[13] = 1'b1$ ; //更新时钟设置参数
- b) 启用时钟: 设置时钟分频参数 CLKDIV 寄存器 (CLKDIV@0x08), 选定分频时钟源 (CLKSRC@0x0c), 并设置时钟使能 CLKENA 寄存器; 然后设置 start\_cmd、update\_clock\_only、wait\_data\_complete 各位为 1, 发送时钟控制指令; 等待 start\_cmd 清零, 表示指令已被发送。  
 $CLKDIV[7:0] = 8'h40$ ; //设置#0 分频器 128 分频

- $CLKSRC[1:0] = 2'h0;$  //选定#0 存储卡的 clk\_out 为#0 分频器时钟  
 $CLKENA[0] = 1'b0;$  //使能#0 存储卡 cclk 时钟  
 $CMD[31] = 1'b1, CMD[21] = 1'b1, CMD[13] = 1'b1;$  //更新时钟设置参数
- c) 设置时钟低功耗模式(可选): 通过设置  $CLKEN@0x10$  寄存器 bit[31:16]位, 使能使设备低功耗模式。在低功耗模式下, 卡设备处于 IDLE 状态超过 8 个时钟后, 控制器自动停止时钟输出, 以降低功耗。配置低功耗模式时, 需要使用 CMD 时钟更新指令(start\_cmd、update\_clock\_only) 才能有效。而且, 低功耗模式只限于 MMC/SD 卡使用; 对于 SDIO 设备, 时钟停止后将检测不到 SDIO 设备中断。
- $CLKEN[16] = 1'b1;$  //使能#0 卡设备时钟低功耗模式

注: 时钟控制指令只作用在 SD/SDIO Controller 内部, 不发送到外部设备, 因而不产生指令完成中断。

#### 5.5.3.4 中断控制

中断使能功能包括: 一个全局中断控制寄存器 int\_enable(CTRL@0x00, bit4), 控制全局中断使能, 只有当 int\_enable 有效时, 中断才会触发。

三组与中断相关的寄存器: INTMASK@0x24, MINTSTS@0x40 和 RINTSTS@0x044, 其中 INTMASK[15:0]位控制屏蔽 16 个特定功能的中断, INTMASK[31:16]位控制屏蔽对应 16 个 SDIO 中断, RINTSTS[31:0]显示中断触发状态 (与 INTMAST 状态无关), 而 MINTSTS[31:0]标示未被屏蔽的触发中断, 三者满足下述关系:

$$MINTSTS = MINTSTS \& RINTSTS;$$

INTMASK 寄存器清零时, 对应中断被屏蔽, 该中断触发后, 仅在 RINTSTS 寄存器中更新中断状态, 在 MINTSTS 无更新。响应 RINTSTS/MINSTS 中断后, 向该寄存器内写 1 清除中断状态。

#### 5.5.3.5 状态检测

STATUS@0x48 寄存器用于设备数据读写时 FIFO 状态、数据总线状态、发送状态等信息查询。控制器与设备通信前, 应先查询该寄存器, 确定控制器状态。

CMD@0x2c 寄存器用于主端发送指令控制、设备初始化、应答格式控制。

#### 5.5.3.6 设备枚举

设备进入数据传输过程之前, 必须先进行设备枚举过程, 进行设备识别及设备初始化操作。枚举过程时钟频率为 100~400KHz。

SD 设备枚举过程流程图如下, 具体参照 SD Specifications Part 1 Physical Layer Specification Version 3.00:

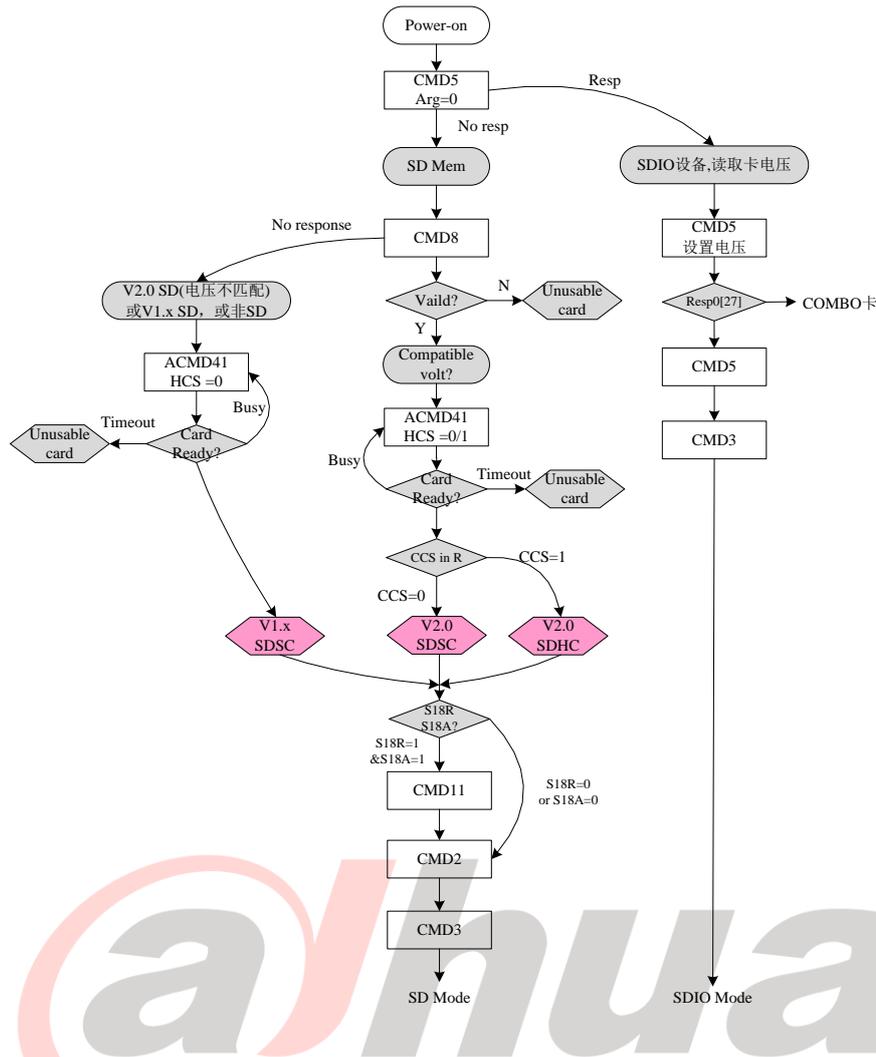


图 36 SD 卡的枚举和初始化流程

- a) 发送 CMD0 初始化卡设备。发送 CMD0 时需要设置 send\_initialization (CMD@0x2c, bit15) 值为 1, 控制器自动发送 80 个时钟的初始化序列;
- b) 识别卡类型。  
首先发送 CMD5, 若收到正确应答, 则为 SDIO 设备, 跳转至 d) 步骤进行 SDIO 设备枚举; 若应答超时或应答错误, 则发送 CMD8, 其 Argument 内容段如下:  
如果控制器收到 CMD8 的正确应答 R7, 则为是支持 SD2.0 以上协议的大容量设备; 接着发送 Argument 内容段如下所示的 ACMD41 指令 (ACMD41 需先发送 CMD55, 然后发送 CMD41, 设备自动将 CMD 识别为 ACMD41);  
如果控制器收到 ACMD41 的正确应答 R3, 则为 SD 存储卡; 否则为 MMC、CEATA 或无效卡 (项目中不对支持 MMC、CEATA 设备不做要求, 故不再累述这两张卡的枚举和读写操作); 检测应答 R3 中 busy bit 为状态, 若为零表示卡设备在初始化过程, 需要重复发送 ACMD41 指令, 等待应答中 busy bit 置位; 若应答超时, 则卡无效;  
如果控制器未收到 CMD8 的正确应答 R3, 则表示设备不是不支持 SD2.0 协议; 接着发送 CMD0 复位卡设备; 然后重新发送 Argument 如下的 ACMD41 指令;  
此时, 如控制器收 ACMD41 到应答 R3, 则卡为 SD 存储器, 否则为 MMC 或者 CEATA;  
注意: 必须先发送 CMD8 然后发送 ACMD41;
- c) 重复发送 ACMD41 直至检测到应答 R3 的 busy 位置位, 标志设备上电初始化完成;

d) 按照规定的指令顺序枚举卡设备。CMD2 在应答中获取卡设备的 CID 信息，CMD3 为卡设备提供新的偏移地址。若要发布新的偏移地址需重新发送 CMD3。

SD 卡：发送 CMD0，CMD8，ACMD41，CMD2，CMD3；

SDIO：发送 CMD5，如果功能数量正确则接着发送 CMD3；若是 memory present 置位，则说明是带有存储区的 COMBO 卡，之后按照 SD 存储卡完成枚举；

MMC：发送 CMD0，CMD1，CMD2，CMD3；

e) 识别 MMC/CE-ATA 卡

发送 CMD8 获取设备 EXT\_CSD 寄存器的 S\_CMD\_SET[4] 值为 1，则设备支持 ATA 模式。然后通过设置 EXT\_CSD 寄存器的 CMD\_SET[4] 激活 ATA 模式指令集，SD/SDIO Controller 通过 CMD6 更换指令模式；

如果设备应答不支持 ATA 模式：发送 CMD39 如果收到应答并在应答中包含 CE-ATA 标志，则是 CE-ATA v1.0 设备；否则是 MMC 卡。

### 5.5.3.7 指令操作

SD/SDIO Controller 指令包括时钟更新指令和 SD 协议标准指令。时钟更新指令不发送到外部指令总线上。指令总线上每次只能同时存在一条要求数据传输的指令。通过设置指令寄存器 (CMD@8'h2c) 发送指令。

某些特定指令在发送前，需先向 argument 寄存器 (CMDARG@0x28) 填充相应功能值。协议中规定没有 argument 的指令，相应 argument 内容建议清零。

当从卡设备接收到应答后，控制器自动将 Command done (RINTSTS@0x44) 置为 1，触发中断。应答错误可能触发 Response timeout error、Response CRC error 或 Response error 中断。短应答内容存放在 RESP0@0x30；长应答内容依次存放在 RESP0@0x30、RESP1@0x34、RESP2@0x38、RESP3@0x3c 四个寄存器中。

发送指令 (CMD@0x2c) 需设置的寄存器如下表所示：

表 33 发送指令寄存器设置

IP Main Features	Values	Comments
默认属性		
Bit[31], start_cmd	1	设置为 1 表示启动指令发送；当 CIU 取走指令后将 start_cmd 设置为 0；当 start_cmd 值为 1 时软件仍然发送指令时 Hardware Locked Write Error (RINTSTS@0x40) 会置为 1，触发错误中断。
Bit[29], use_hold_reg	0	CMD 和 DATA 不经过 HOLD Reg 发送
Bit[28], volt_switch	0	不进行电压切换
Bit[27:22]	0	与 SD 和 SDIO 无关
Bit[21], update_clock_registers_only	0/1	0: SD 指令，CIU 将指令发送到卡设备； 1: 时钟控制指令，CIU 不把指令发送到卡设备，只更新时钟相关寄存器。
Bit[20:16], card_number	x	卡设备偏移地址
Bit[15], send_initialization	0/1	0: 不初始化序列 1: 发送 80 个时钟的 '1'，初始化卡设备
Bit[14], stop_abort_cmd	0	0: 对当前指令无影响 1: 停止或放弃当前指令
Bit[12], send_auto_stop	0	0: 数据传输结束后，无自动停止指令





- 1) 单块或多块读数据
  - a) 向寄存器 CTRL@0x00 fifo\_reset 位写“1”，复位 FIFO 指针，查询等待该位自动清零。
  - b) 向寄存器 BYTCNT@0x20 写入待传输数据的字节数；
  - c) 向寄存器 BLKSIZ@0x1c 写入块的大小（SDHC 块大小固定为 512 字节）；
  - d) 向寄存器 CMDARG@0x28 写入数据的起始地址；
  - e) 设置寄存器 CMD。对于 SD/MMC 存储器，分别使用指令 CMD17/CMD18 进行单块/多块读取操作；对于 SDIO 设备，仅使用指令 CMD53 进行单块/多块读操作。一旦寄存器 CMD 被写入，SD/SDIO Controller 开始执行指令。当指令发送完成且接收到正确应答后，触发 Command Done 中断；
  - f) 检查寄存器 STATUS@0x48 fifo\_empty 状态位清“0”，开始从寄存器 FIFO@0x200 读取数据，以便 SD/SDIO Controller 接收后面的数据；同时检查数据错误中断，即寄存器 RINTSTS bit[7]和 RINTSTS bit[15]。此时，程序可以发送一个停止指令中止数据的传输。
  - g) 读取 FIFO 数据，直至 STATUS fifo\_empty 状态位置“1”。数据读取完成后，向 RINTSTS bit[3]写“1”，清除传输完成中断；
  - h) 若进行多块读取操作，且已将寄存器 CMD send\_auto\_stop 置“1”，SD/SDIO Controller 在数据读取完成后自动发送 CMD12 停止指令。必要时可以主动发送数据传输停止指令 CMD12 停止数据传输。

表 34 SD 存储器读取操作指令设置

IP Main Features	Values	Comments
默认属性		
Bit[31], start_cmd	1	设置为 1 表示启动指令发送；当 CIU 取走指令后将 start_cmd 设置为 0；当 start_cmd 值为 1 时软件仍然发送指令时 Hardware Locked Write Error (RINTSTS @0x40) 会置为 1，触发错误中断。
Bit[29], use_hold_reg	0/1	CMD 和 DATA 是否经过 HOLD Reg 发送
Bit[28], volt_switch	0	不进行电压切换
Bit[27:22]	0	与 SD 和 SDIO 无关
Bit[21], update_clock_registers_only	0	0: SD 指令，CIU 将指令发送到卡设备； 1: 时钟控制指令，CIU 不把指令发送到卡设备，只更新时钟相关寄存器。
Bit[20:16], card_number	x	卡设备偏移地址
Bit[15], send_initialization	0	0: 不初始化序列 1: 发送 80 个时钟的'1'，初始化卡设备
Bit[14], stop_abort_cmd	0	0: 对当前指令无影响 1: 停止或放弃当前指令
Bit[12], send_auto_stop	0/1	0: 数据传输结束后，无自动停止指令 1: 数据传输结束后，有自动停止指令
Bit[11], transfer_mode	0	0: 数据块传输；1: 数据流传输
Bit[10], read/write	0	0: 从卡设备读取；1: 向卡设备写入
Bit[9], data_expected	1	0: 非数据传输操作；1: 数据传输操作
Bit[7], response_length	0	0: 短应答；1: 长应答。
Bit[6], response_expect	1	0: 不需应答（如 CMD0/CMD4/CMD15 等） 1: 需要应答
Bit[5:0], cmd_index	x	指令编号

用户属性		
Bit[13], wait_prvdata_complete	1	等待数据传输操作完成才能发送下一条指令，一般设为 1（除非发送状态查询指令或传输中止指令）
Bit[8], check_response_crc	1	对应答 CRC 校验

## 2) 单块与多块写数据

- a) 向寄存器 CTRL fifo\_reset 写“1”，复位 FIFO 指针，查询等待直至该位自动清零；
- b) 向寄存器 BYTCNT 写入待传输数据的大小；
- c) 向寄存器 BLKSIZ 写入块的大小（SDHC 块大小固定为 512 字节）；
- d) 向寄存器 CMDARG 写入数据的起始地址；
- e) 读取 STATUS 寄存器信息，判断 FIFO 状态后，将数据写入 FIFO，即写寄存器 0x200，通常在开始时应先写满 FIFO；
- f) 设置寄存器 CMD。对于 SD/MMC 卡，分别使用指令 CMD24/CMD25 进行单块/多块写操作；对于 SDIO 卡，使用指令 CMD53 进行单块/多块写操作；
- g) 检查寄存器 STATUS@0x48 fifo\_empty 状态位置“1”，或 RINTSTS [TXDR]中断位置“1”，写 0x200 寄存器，向 FIFO 填充数据；同时应检查数据错误中断，即检查寄存器 RINTSTS [DCRC]位、RINTSTS [EBE]位、RINTSTS [HLE]位。如果有需要，程序可以发送一个停止指令以中止数据的传输。当寄存器 RINTSTS DTO 置“1”，数据传输结束，回写“1”，清除该中断；
- i) 若进行多块写入操作，且已将寄存器 CMD send\_auto\_stop 置“1”，控制器自动发送停止指令结束一次数据传输。必要时可以主动发送数据传输停止指令 CMD12 停止数据传输；
- j) 查询并等待寄存器 CMD [data\_busy]由“1”变为“0”。

表 35 SD 存储器写入操作指令设置

IP Main Features	Values	Comments
默认属性		
Bit[31], start_cmd	1	设置为 1 表示启动指令发送；当 CIU 取走指令后将 start_cmd 设置为 0；当 start_cmd 值为 1 时软件仍然发送指令时 Hardware Locked Write Error (RINTSTS @0x40) 会置为 1，触发错误中断。
Bit[29], use_hold_reg	0/1	CMD 和 DATA 是否经过 HOLD Reg 发送
Bit[28], volt_switch	0	不进行电压切换
Bit[27:22]	0	与 SD 和 SDIO 无关
Bit[21], update_clock_registers_only	0	0: SD 指令，CIU 将指令发送到卡设备； 1: 时钟控制指令，CIU 不把指令发送到卡设备，只更新时钟相关寄存器。
Bit[20:16], card_number	x	卡设备偏移地址
Bit[15], send_initialization	0	0: 不初始化序列 1: 发送 80 个时钟的'1'，初始化卡设备
Bit[14], stop_abort_cmd	0	0: 对当前指令无影响 1: 停止或放弃当前指令
Bit[12], send_auto_stop	0/1	0: 数据传输结束后，无自动停止指令 1: 数据传输结束后，有自动停止指令
Bit[11], transfer_mode	0	0: 数据块传输；1: 数据流传输

Bit[10], read/write	1	0: 从卡设备读取; 1: 向卡设备写入
Bit[9], data_expected	1	0: 不进行数据传输; 1: 数据传输指令
Bit[7], response_length	0	0: 短应答; 1: 长应答。
Bit[6], response_expect	1	0: CMD0/CMD4/CMD15 等指令不需要应答; 1: 需要应答。
Bit[5:0], cmd_index	x	指令编号
用户属性		
Bit[13], wait_prvdata_complete	1	等待数据传输操作完成才能发送下一条指令, 一般设为 1 (除非发送状态查询指令或传输中止指令)
Bit[8], check_response_crc	1	对应答 CRC 校验

### 5.5.3.9 Auto-stop 操作

在多块读写指令操作中, 需使用停止指令完成数据传输。停止指令可以软件通过主动发送 CMD12, 也可以在 CMD(0x2c)置位 Auto-stop 功能使 SD/SDIO Controller 自动发送 CMD12, 以便卡能返回相应的状态。CMD12 完成后会触发 RINTSTS(0x44) auto\_command\_done 中断, 用以判断该停止指令是否完成, 其应答内容保存在寄存器 RESP1(0x34)中。Auto-stop 功能的应用场合如下

表 36 Auto-Stop 支持的卡类型和传输模式

存储卡类型	传输模式	字节数(0x20)	可否使用 send_auto_stop	说明
MMC	Stream read/write	0	否	Open-end 流传输
MMC	Stream read/write	>0	是	字节传输后 Auto-stop
MMC	Single-block read/write	>0	否	字节数=0 非法
MMC	Multiple-block read/write	>0	是	传输完成后 Auto-stop
SDMEM	Single-block read/write	>0	否	字节数=0 非法
SDMEM	Multiple-block read/write	0	否	Open-end 多块操作
SDMEM	Multiple-block read/write	>0	是	传输完成后 Auto-stop
SDIO	Single-block read/write	>0	否	字节数=0 非法
SDIO	Multiple-block read/write	0, or >0	否	SDIO 不支持该功能

### 5.5.3.10 停止数据传输

- 1) 指令 CMD12 用于 MMC/SD 存储卡、SDIO 存储功能的数据传输停止。  
注意需要设置 CMD(0x2c) bit14 = 1'b1, bit13 = 1'b0
- 2) 指令 CMD52 设置 SDIO 卡的寄存器 CCCR 0x06 地址中 bit0-1(AS0-AS1)位, 用于 SDIO 的 I/O 功能的传输停止。  
注意需要设置 CMD(0x2c) bit14 = 1'b1, bit13 = 1'b0

### 5.5.3.11 擦除操作

- 1) 预擦除操作  
在多块写入操作前, 用 ACMD23 指令设置这些被写入块为可被预擦除模式时, 写入速度会快于未被设置相同操作。主端利用 ACMD23 告知存储卡写入/预擦除块数量。若主端在中间过程发起终止写入指令(CMD12), 余下的已经被设置预擦除模式的数据块的内容将处于

未知状态，内容可能已被擦除，也可能未被擦除。如果主端发送的数据块数多于 ACMD23 中给定的数据块数值，当接收到新数据块时，存储卡逐块擦除数据块。完成多块写操作后，这个数值(pre-erase-count)会复位成默认值 1。协议推荐在 CMD25 写入指令前先发送 ACMD23 指令，以提高存储卡写入速度。注意，ACMD23 必须在写入指令前一条发出，否则 pre-erase-count 会被其他指令清空。

## 2) 正常擦除操作

用 ERASE\_WR\_BLK\_START (CMD32) 指令和 ERASE\_WR\_BLK\_END(CMD33)指令要定义擦除数据块的起末地址，随之发送 ERASE(CM38)开始擦除操作。这三条指令必须顺序执行，否则存储卡应答 ERASE\_SEQ\_ERROR 错误，并复位整个序列。若其他指令插入 (SEND\_STATUS 指令除外)，存储卡置位 ERASE\_RESET，复位擦除指令队列，执行最后一条指令(插入的指令)。若擦除地址范围内含有写保护的数据段，必须跳过这些数据段，仅擦除未写保护的区域，并同时置位 WP\_ERASE\_SKIP。最小擦除范围由 CSD 寄存器中 Sector 段规定。

另外，擦除操作同样耗时较长，主端可以采用同写入操作类似的交织操作方式，以提高效率。SCR 寄存器中的 DATA\_STAT\_AFTER\_ERASE(bit 55)定义擦除后状态是 1 或是 0。

## 5.5.4 寄存器概述

寄存器表 18 SDIO 寄存器概述 (基地址: 0x1500\_0000)

偏移地址	名称	描述	页码
0x00	CTRL	控制寄存器	
0x04	PWREN	上电使能寄存器	
0x08	CLKDIV	时钟分频寄存器	
0x0c	CLKSRC	时钟源寄存器	
0x10	CLKENA	时钟使能寄存器	
0x14	TMOUT	超时寄存器	
0x18	CTYPE	卡类型寄存器	
0x1c	BLKSIZ	块大小寄存器	
0x20	BYTCNT	字节计数寄存器	
0x24	INTMASK	中断掩码寄存器	
0x28	CMDARG	指令内容段寄存器	
0x2c	CMD	命令寄存器	
0x30	RESP0	响应回复寄存器 0	
0x34	RESP1	响应回复寄存器 1	
0x38	RESP2	响应回复寄存器 2	
0x3c	RESP3	响应回复寄存器 3	
0x40	MINTSTS	中断掩码寄存器	
0x44	RINTSTS	中断状态寄存器	
0x48	STATUS	状态寄存器，主要用于检视控制器状态	
0x4c	FIFOTH	FIFO 门限寄存器	
0x50	CDETECT	卡检测寄存器	
0x54	WRTPRT	写保护寄存器	
0x58	GPIO	不使用	
0x5c	TCBCNT	CIU 模块传输的 byte 个数寄存器	

0x60	TBBCNT	Host 与 BIU_FIFO 交互数据的 byte 个数寄存器	
0x64	DEBNCE	卡检测反跳寄存器（单位为 host clock 数目）	
0x68	USRID	用户 ID 寄存器	
0x6C	VERID	软核版本 ID 寄存器	
0x70	HCON	硬件配置寄存器	
0x74	UHS_REG	不使用	
0x78	RST_n	硬复位寄存器	
0x80~0x98	DMA_CTRL	DMA 相关，不使用	
0x100	CardThrCtl	卡读门限使能寄存器	
0x104	Back_end_power	Back_end Power 寄存器	
0x108	UHS_REG_ECT	不使用	
0x10c	EMMC_DDR_REG	不使用	
0x110	ENABLE_SHIFT	转换相位控制寄存器	
>=0x200	DATA	FIFO 读写空间	

### 5.5.5 寄存器描述

#### 5.5.5.1 CTRL(控制寄存器)

偏移地址：0x00

复位方式：H/S

Bits	Access	Name	Default	Description
[31:26]	-	保留	0x0	-
[25]	-	use_internal_dmac	0x0	不使用
[24]	RW	enable_OD_pullup	0x0	芯片外部 open_drain 上拉：0-不使能，1-使能。
[23:20]	RW	card_voltage_b	0x0	卡 regulator_B 电压设置，输出给 card_volt_a。可选的，该 PORTS 口可作为输出口。
[19:16]	RW	card_voltage_b	0x0	卡 regulator_A 电压设置，输出给 card_volt_a。可选的，该 PORTS 口可作为输出口。
[15:12]	-	保留	0x0	-
[11]	RW	ceata_device_interru t_status	0x0	0- 关闭 CE_ATA 设备中的中断， 1- 使能 CE_ATA 设备中的中断。
[10]	RW	send_auto_stop_ccsd	0x0	在发送 CCSD 到 CE_ATA 设备后，发送内部产生的 STOP。必须同时设置 send_ccsd 与 send_auto_stop_ccsd
[9]	RW	send_ccsd	0x0	发送 command completion signal disable 到 CE_ATA 设备
[8]	RW	abort_read_data	0x0	0-不改变，1-复位数据状态机，在数据状态机返回到 IDLE 时该 BIT 位自动清除。
[7]	RW	send_irq_response	0x0	0-不改变，1-发送自动 IRQ 回复
[6]	RW	read_wait	0x0	0-清除读等待，1-读等待有效
[5]	RW	dma_enable	0x0	DMA 传输模式使能：值 1 使能，值 0 关闭使能
[4]	RW	int_enable	0x0	全局中断使能信号：值 1 使能，值 0 关闭使能

[3]	-	保留	0x0	-
[2]	RW	dma_reset	0x0	复位内部的 DMA 接口控制逻辑，在两个 AHB clocks 后自动清零。
[1]	RW	fifo_reset	0x0	复位 FIFO，在复位操作完成后自动清零。
[0]	RW	controller_reset	0x0	为 1 时复位控制器，在两个 AHB 和两个 cclk_in 时钟周期时该 bit 位自动清零。

注：H/S 代表硬件复位与软件复位

#### 5.5.5.2 PWREN (上电使能寄存器)

偏移地址：0x4

复位方式：H/S

Bits	Access	Name	Default	Description
[31:1]	-	保留	0x0	-
[0]	RW	power_enable	0x0	项目用采用一个控制器操作一张 SD 卡的配置，上电、时钟分频、时钟使能等均使用 1bit 控制位独立控制。下文仅说明单个控制器的配置，另一个控制器也采用相同操作，不再累述。 power_enable[0] = 1'b1，设备上电 power_enable[0] = 1'b0，设备掉电 其它位保留，清零

#### 5.5.5.3 CLKDIV (时钟分频寄存器)

偏移地址：0x8

复位方式：H/S

Bits	Access	Name	Default	Description
[31:8]	-	clk_divider1/2/3	0x0	不使用
[7:0]	RW	clk_divider0	0x0	#0 分频器(cclk_in 分频得到 cclk_out, 分频率=2 倍写入值) 其中 0 代表不分频，255 代表 510 倍分频，以此类推 项目中仅配置[7:0]，其它位[31:8]=24'h0

#### 5.5.5.4 CLKSRC(时钟源寄存器)

偏移地址：0xc

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RO	clk_source	0x0	选择每张 SD 储存器的驱动时钟，关系如下 00b: 选择 Clock divider_0, 01: 选择 Clock divider_1, 10b: 选择 Clock divider_2, 11: 选择 Clock divider_3, 项目每个控制器仅含有 clk_divder0，该段固定为 0

### 5.5.5.5 CLKENA(时钟使能寄存器)

偏移地址: 0x10

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	RW	cclk_low_power	0x0	16 张 SD 存储器或 1 张 MMC 存储器低功耗控制 0: 非低功耗模式; 1: 低功耗模式 项目只配置 bit[16], 其它位全部固定为 0
[15:0]	RW	cclk_enable	0x0	16 张 SD 存储器或 1 张 MMC 存储器时钟使能控制, 0: 时钟无效; 1: 时钟使能 项目只配置 bit[0], 其它位全部固定为 0

### 5.5.5.6 TMOUT(Timeout 寄存器)

偏移地址: 0x14

复位方式: H/S

Bits	Access	Name	Default	Description
[31:8]	RW	data_timeout	0xfffff	储存卡数据读取超时值设置, 储存卡时钟结束后计时, 以被选择卡的时钟 cclk_out 的周期值为单位计时
[7:0]	RW	response_timeout	0x40	回复超时设置, 以 cclk_out 的周期值为单位计时

### 5.5.5.7 CTYPE(卡类型寄存器)

偏移地址: 0x18

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	RW	card_width1	0x0	指示每个存储卡是否是 8-bit 类型 0: 非 8-bit 类型; 1: 8-bit 类型 项目无 8bit 应用, 该段全写 0
[15:0]	RW	card_width0	0x0	指示每个存储卡是是 1-bit 或 4-bit 类型 0: 1-bit 类型;1: 4-bit 类型 项目仅配置 bit[0], 其它位全写 0

### 5.5.5.8 BLKSIZ(块大小寄存器)

偏移地址: 0x1c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved	-	保留
[15:0]	RW	block_size	0x200	块大小 (单位: 字节)

### 5.5.5.9 BYTCNT (字节计数寄存器)

偏移地址: 0x20

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	byte_count	0x200	将要传输数据的字节数 对未知数据大小的传输, 须写入 0

#### 5.5.5.10 INTMASK (中断掩码寄存器)

偏移地址: 0x24

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	RW	sdio_int_mask	0x0	屏蔽 16 张存储卡的 SDIO 中断, 0: 屏蔽; 1: 使能 项目仅配置 bit[16], 其它位全写 0
[15:0]	RW	int_mask	0x0	屏蔽不希望的中断, 0: 屏蔽; 1: 使能 bit15: End-bit error(read)/Write no CRC(EBE) bit14: Auto Command Done(ACD) bit13: Start Bit Error(SBE)/Busy Complete Interrupt(BCI) bit12: Hardware Locked write Error(HLE) bit11: FIFO underrun/overrun Error(FRUN) bit10: Data starvation-by-host timeout(HTO)/Volt_switch_int bit9: Data Read Timeout(DRTO) bit8: Response Timeout(RTO) bit7: Data CRC Error(DCRC) bit6: Response CRC Error(RCRC) bit5: Receive FIFO Data Request(RXDR) bit4: Transmit FIFO Data Request(TXDR) bit3: Data Transfer Over(DTO) bit2: Command done(CD) bit1: Responses Error(RE) bit0: Card Detect(CD)

#### 5.5.5.11 CMDARG (command\_argument 寄存器)

偏移地址: 0x28

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	cmd_arg	0x0	传到卡的 32bit 的指令内容

#### 5.5.5.12 CMD (命令寄存器)

偏移地址: 0x2c

复位方式: H/S

Bits	Access	Name	Default	Description
[31]	RW	start_cmd	0x0	发送指令, CIU 接受完指令后, 该位自动清零

[30]	-	reserved	0x0	保留
[29]	RW	use_hold_reg	0x1	0: CMD 和 DATA 不通过 HOLD 寄存器发送到存储卡 1: CMD 和 DATA 通过 HOLD 寄存器发送到存储卡
[28]	RW	volt_switch	0x0	0: 无电压切换 1: 电压切换使能, 发 CMD11 时必须置位。
[27]	RW	boot_mode	0x0	Boot 模式选择: 0: 必须 boot 操作 1: 可选 boot 操作
[26]	RW	disable_boot	0x0	Boot 无效
[25]	RW	expect_boot_ack	0x0	(CIU) 希望获得 boot 的响应
[24]	RW	enable_boot	0x0	使能 boot
[23]	RW	ccs_expected	0x0	0: CE_ATA 设备内中断无效, 或命令不需要得到设备 CCS (命令完成信号)。 1: CE_ATA 设备内中断使能, RW_BLK 需要来自 CE_ATA 设备的 CCS (命令完成信号)。 项目无 CE-ATA 应用, 该位写 0
[22]	RW	read_ceata_device	0x0	0: 主端不向 CE_ATA 设备读取数据 1: 主端向 CE-ATA 设备读取数据 项目无 CE-ATA 应用, 该位写 0
[21]	RW	update_clock_registers_only	0x0	0: 普通指令序列 1: 不发送指令, 仅仅更新时钟寄存器
[20:16]	RW	card_number	0x0	当前有效的卡号
[15]	RW	send_initialization	0x0	0: 在发送指令前, 不发送初始化序列 1: 在发送指令前, 发送初始化序列 在上电后, 在发任何命令给卡之前必须进行 80 个 clocks 的卡初始化。
[14]	RW	stop_abort_cmd	0x0	0: 不停止或放弃当前进行中的数据传输 1: 停止或放弃当前进行中的数据传输
[13]	RW	wait_prvdata_complete	0x0	0: 无视上条数据传输是否完成, 发送新指令 1: 等待上条数据传输完成后, 发送新指令
[12]	RW	send_auto_stop	0x0	0: 数据传输结束后, 无 (自动) 停止指令发出 1: 数据传输结束后, 有 (自动) 停止指令发出
[11]	RW	transfer_mode	0x0	0: 数据块传输指令; 1: 数据流传输指令
[10]	RW	read/write	0x0	0: 读取; 1: 写入
[9]	RW	data_expected	0x0	0: 无数据读写; 1: 有数据读写
[8]	RW	check_response_crc	0x0	0: 不进行答复 CRC 检查; 1: 进行答复 CRC 检查
[7]	RW	response_length	0x0	0: 短答复; 1: 长答复
[6]	RW	response_expect	0x0	0: 无答复; 1: 有答复
[5:0]	RW	cmd_index	0x0	指令号

#### 5.5.5.13 RESP0 (响应回复寄存器 0)

偏移地址: 0x30

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	response 0	0x0	答复寄存器 0, 短应答或长应答 bit[31:0]

#### 5.5.5.14 RESP1 (响应回复寄存器 1)

偏移地址: 0x34

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	response 1	0x0	答复寄存器 1, 长应答 bit[63:32] auto_Stop 应答保存在该寄存器

#### 5.5.5.15 RESP2 (响应回复寄存器 2)

偏移地址: 0x38

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	response 2	0x0	答复寄存器 2, 长应答 bit[95:64]

#### 5.5.5.16 RESP3 (响应回复寄存器 3)

偏移地址: 0x3c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	response 3	0x0	答复寄存器 3, 长应答 bit[127:96]

#### 5.5.5.17 MINTSTS (屏蔽中断寄存器)

偏移地址: 0x40

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	RO	sdio_interrupt	0x0	未被屏蔽的存储卡 SDIO 中断状态, 每 bit 对应一张卡: bit[31]对应 card[15],bit[16]对应 card[0]。 0: 无 SDIO 中断从存储卡发出 1: 有 SDIO 中断从存储卡发出 项目仅使用 bit[16]
[15:0]	RO	int_status	0x0	中断状态信号: bit15: End-bit error(read)/Write no CRC(EBE) bit14: Auto Command Done(ACD) bit13: Start Bit Error(SBE)/Busy Complete Interrupt(BCI) bit12: Hardware Locked write Error(HLE) bit11: FIFO underrun/overrun Error(FRUN) bit10: Data starvation-by-host timeout(HTO)/Volt_switch_int bit9: Data Read Timeout(DRTO)

				bit8: Response Timeout(RTO) bit7: Data CRC Error(DCRC) bit6: Response CRC Error(RCRC) bit5: Receive FIFO Data Request(RXDR) bit4: Transmit FIFO Data Request(TXDR) bit3: Data Transfer Over(DTO) bit2: Command done(CD) bit1: Responses Error(RE) bit0: Card Detect(CD)
--	--	--	--	---

注: MISTATUS = RINTSTS and INTMASK

### 5.5.5.18 RINTSTS (原始中断状态寄存器)

偏移地址: 0x44

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	RO/W C	sdio_interrupt	0x0	16 张存储卡 SDIO 中断状态, 与掩码无关。每 bit 对应一张卡: bit[31]对应 card[15],bit[16]对应 card[0]。 0: 无 SDIO 中断从存储卡发出 1: 有 SDIO 中断从存储卡发出 项目仅使用 bit[16]
[15:0]	RO/R W	int_status	0x0	诸多中断使能信号, 与屏蔽无关, 对以下 bit 位写 1 可清零。 bit15: End-bit error(read)/Write no CRC(EBE) bit14: Auto Command Done(ACD) bit13: Start Bit Error(SBE)/Busy Complete Interrupt(BCI) bit12: Hardware Locked write Error(HLE) bit11: FIFO underrun/overrun Error(FRUN) bit10: Data starvation-by-host timeout(HTO)/Volt_switch_int bit9: Data Read Timeout(DRTO) bit8: Response Timeout(RTO) bit7: Data CRC Error(DCRC) bit6: Response CRC Error(RCRC) bit5: Receive FIFO Data Request(RXDR) bit4: Transmit FIFO Data Request(TXDR) bit3: Data Transfer Over(DTO) bit2: Command done(CD) bit1: Responses Error(RE) bit0: Card Detect(CD)

### 5.5.5.19 STATUS (状态寄存器)

偏移地址: 0x48

复位方式: H/S

Bits	Access	Name	Default	Description
[31]	RO	dma_req	0x0	DMA 请求信号状态位
[30]	RO	dma_ack	0x0	DMA 响应信号状态位
[29:17]	RO	fifo_count	0x0	FIFO 内写入数据的数量
[16:11]	RO	response_index	0x0	前条回复索引号
[10]	RO	data_state_mc_busy	0x1	数据发送或接受状态机忙
[9]	RO	data_busy	0x0	与 card_data[0]的值相反 0: 存储卡数据不忙; 1: 存储卡数据忙
[8]	RO	data_3_status	0x0	选定存储卡 card_datap[3]信号, 检测是否有卡。 0: 有存储卡; 1: 无存储卡
[7:4]	RO	command_fsm_states	0x0	指示命令状态机的当前状态: 0- IDLE 1- Send init sequence 2- Tx cmd start bit 3- Tx cmd bit 4- Tx cmd index + arg 5- Tx cmd crc7 6- Tx cmd end bit 7- Rx resp start bit 8- Rx resp IRQ response 9- Rx resp tx bit 10- Rx resp cmd idx 11- Rx resp data 12- Rx resp crc7 13- Rx resp end bit 14- Cmd path wait NCC 15- Wait, CMD_to_response turnaround
[3]	RO	fifo_full	0x0	FIFO 满标志
[2]	RO	fifo_empty	0x1	FIFO 空标志
[1]	RO	fifo_tx_watermark	0x1	FIFO 达到发送水线的值
[0]	RO	fifo_rx_watermark	0x0	FIFO 达到接收水线的值

#### 5.5.5.20 FIFOTH (FIFO 门限寄存器)

偏移地址: 0x4c

复位方式: H/S

Bits	Access	Name	Default	Description
[31]	-	reserved	0x0	保留
[30:28]	RW	DW_DMA_Mutiple_Transaction_Size	0x0	Burst 操作时的突发(块)大小, 须和 DW_DMA 控制器中 SRC/DEST_MSIZE 大小一致 000:1 transfer; 001:4; 010:8; 011:16; 100:32; 101:64; 110:128; 111:256 1 transfer 大小 = H_DATA_WIDTH
[26:16]	RW	RX_WMark	0x7f	接受门限设置, 当 FIFO 内数据量大于该门限时, 不再接收数据(项目中为 FIFO_DEPTH 为 128)
[15:12]	RW	reserved	0x0	保留
[11:0]	RW	TX_WMark	0x0	发送门限设置, 当 FIFO 内数据量小于该门限时, 不再发送数据

#### 5.5.5.21 CDETECT (卡检测寄存器)

偏移地址: 0x50

复位方式: H/S

Bits	Access	Name	Default	Description
[31:30]	RO	reserved	0x0	保留
[29:0]	RO	card_detect_n	card_detect_n_inputs	输入端口 card_detect_n 上的值 (每张卡 1bit), 0: 检测到卡; 1: 未检测到卡 项目仅使用 bit[0]

#### 5.5.5.22 WRTPRT (写保护寄存器)

偏移地址: 0x54

复位方式: H/S

Bits	Access	Name	Default	Description
[31:30]	RO	reserved	0x0	保留
[29:0]	RO	write_protect	card_write_prt_inputs	输入端口 card_write_prt 上的值 (每张卡 1bit), 0: 卡未写保护; 1: 卡写保护 项目仅使用 bit[0]

#### 5.5.5.23 TCBCNT (CIU 模块传输的 byte 个数寄存器)

偏移地址: 0x5c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	trans_card_byte_count	0x0	由模块 CIU 发送给 Card 的数据字节数量

#### 5.5.5.24 TBBCNT (Host 与 BIU\_FIFO 交互数据 byte 个数寄存器)

偏移地址: 0x60

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	trans_fifo_tbyte_count	0x0	Host 与 BIU-FIFO 之间传输数据的字节数量

#### 5.5.5.25 DEBNCE (卡检测反跳寄存器 (单位为 host clock 数目))

偏移地址: 0x64

复位方式: H/S

Bits	Access	Name	Default	Description
[31:24]	RW	reserved	0x0	保留
[23:0]	RW	debounce_count	0xfffff	反跳过滤逻辑的值, 已主端时钟周期为单位。典型值为

				5-25ms
--	--	--	--	--------

#### 5.5.5.26 USRID (用户 ID 寄存器)

偏移地址: 0x68

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	usrid	0x45700311	用户 ID 寄存器, 由用户设置。

#### 5.5.5.27 VERID (软核版本号寄存器)

偏移地址: 0x6c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	verid	0x5342250a	Synopsys 软核版本号寄存器

#### 5.5.5.28 HCON (软核配置寄存器)

偏移地址: 0x70

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	hcon	0x00c44cc1	用户生成 IP 前, 配置该寄存器, 实现与硬件配置关联的软件开发

#### 5.5.5.29 RST\_n (卡复位寄存器)

偏移地址: 0x78

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved	-	保留
[15:0]	RW	card_reset	0x1	MMC 卡复位控制(产生顶层端口上的 rst_n 信号) 0: 复位; 1: 激活状态 项目仅使用 bit[0]

#### 5.5.5.30 CardThrCtl (读门限使能寄存器)

偏移地址: 0x100

复位方式: H/S

Bits	Access	Name	Default	Description
[31:28]	-	reserved	0x0	保留
[N:16]	RW	CardRD Threshold	0x0	储存卡读取门限大小, N 由 FIFO 大小决定。

				项目中 FIFO_DEPTH 为 128, 故 N=27
[15:2]	-	reserved	0x0	保留
[1]	RW	BsyClrIntEn	0x0	使能忙清除中断 0: busy 清除中断无效; 1: busy 清除中断使能
[0]	RW	CardRdThrEn	0x0	储存卡读取门限使能 0: 读取门限无效; 1: 读取门限使能

#### 5.5.5.31 Back\_end\_power (Backend Power 寄存器)

偏移地址: 0x104

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved	0x0	保留
[15:0]	RW	back_end_power	0x0	back_end供电, 0: 停电; 1: 供电, 每张卡一个pin脚

#### 5.5.5.32 ENABLE\_SHIFT (相位调整控制寄存器)

偏移地址: 0x110

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Enable_shift	0x0	控制设计中默认的相移值, 每 2bit 控制 1 张存储卡 00: 默认相移; 01: 在下一个时钟上升沿使能相移 10: 在下一个时钟下降沿使能相移; 11: 保留

#### 5.5.5.33 DATA (FIFO 读写入口地址)

偏移地址:  $\geq 0x200$

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	FIFO	0x0	数据读写 FIFO 入口地址, 随意读写 FIFO, 可能导致逻辑紊乱

## 5.6 USB

### 5.6.1 概述

HR\_C7000 采用 Non-OTG Device 配置模式, 实现 USB1.1 Device 功能。

### 5.6.2 功能描述

Non-OTG Device 主要配置参数如下:

- 工作模式: Non-OTG Device
- 结构: 内部 DMA
- SG-DMA: 无
- HS PHY 接口: 无

- USB1.1 FS 收发器接口：专用 FS 接口
- 总 EP 数目：1 CTRL IN/OUT(EP0) + 2 IN(EP1/3) + 2 OUT(EP2/4)，共计 5 个 EP
- TxFIFO 模式：专用 TxFIFO
- 总 FIFO 深度：128(FIFO 位宽为 35bit)

### 5.6.3 工作方式

Dev 模式下，软核 MAC 接收 Host 发来的各种包，并检测包的完整性，具体流程如下：

- 当接收到 OUT 或 SETUP 信令包时，软核 MAC 等待并检测随之而来的数据包 PID 号，然后将数据写入 RxFIFO(可用)。数据包接收完成后，软核 MAC 检测数据包的完整性，并向 Host 做相应的应答。若软核在收到 OUT 信令但 RxFIFO 不可用时，MAC 应答 NAK；
- 当接收到 IN 信令包且 TxFIFO 可用时，MAC 自动从 TxFIFO 读取数据，并发送数据包，接着等待 Host 可能的应答。若收到 IN 信令但 TxFIFO 无数据时，MAC 应答 NAK。

软核收发数据包时，各种中断和配置有**严格**的前后时序关系，若配置内容或配置时间点不正确，则软核无法正常工作。本小节抽出通用的中断触发时序和配置时序，方便上层软件开发者查询，图例说明如下：

表 37 图例符号定义

编号	用例名称	描述
1		Host 包
2		Dev 包
3	黄色文字	触发 GINTSTS.RxFLVL 中断，有包写入 RxFIFO
4	蓝色文字	设置 IN/OUT 端口传输属性 若有数据发送，可向 TxFIFO 压入数据
5	黑色文字	传输完成中断，标示传输完成
6	时间先后关系	从左到右，从上到下

#### 1) 带有数据输出的 CTRL 传输

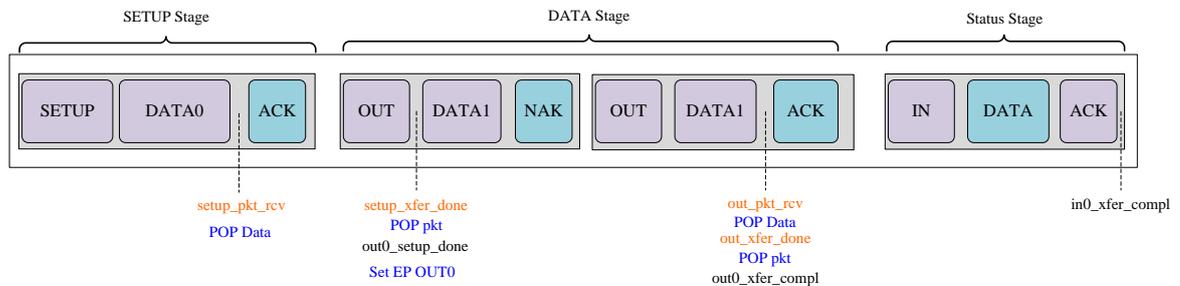


图 39 CTRL OUT 传输时序

2) 带有数据输入的 CTRL 传输时序

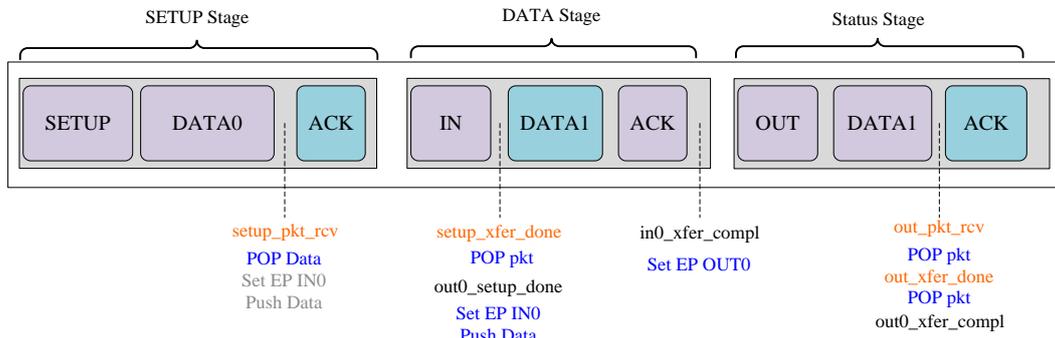


图 40 CTRL IN 传输时序

3) 无数据的 CTRL 传输时序

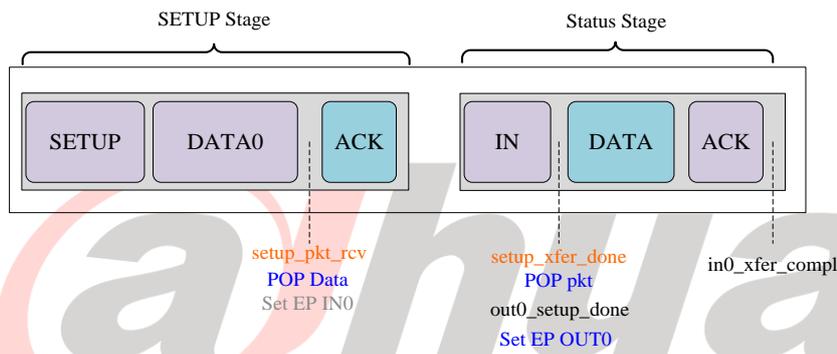


图 41 无数据 CTRL 传输时序

4) 单次 OUT 事务

在接收到 OUT 信令时，尚未正确设置 OUT 端口参数时的时序；若接收前已经配置好 OUT 端口参数，则 OUT 端口直接应答 ACK；

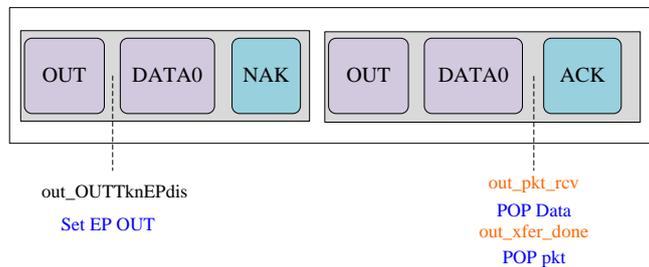


图 42 单次 OUT 事务时序

5) 多次 OUT 事务

设置 OUT 端口接收多个包后，应用软件在 RxFLvl 中断服务程序中从 RxFIFO 中不断读

取数据，当全部数据传输完成后，软核触发完成中断。若在此过程中，Host 未收到某个 ACK，Host 会重发该数据包，软核自动应答 ACK、且无 RxFLvl 中断触发，软件无需干预；

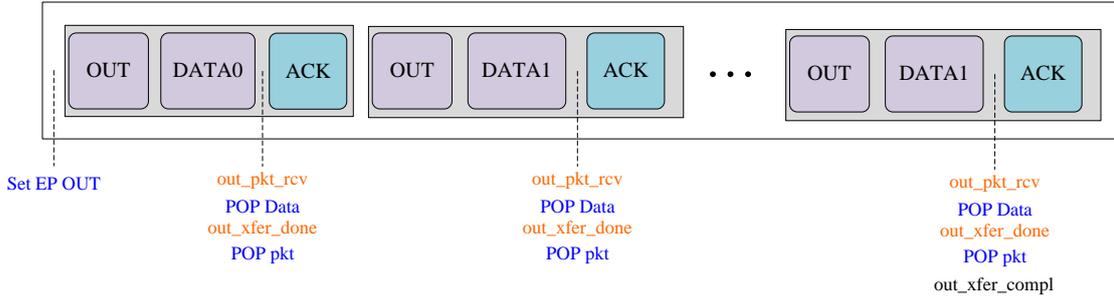


图 43 多次 OUT 事务时序

6) 单次 IN 事务时序

IN 端口在接收到 IN 信令时为 Tx FIFO 未就绪时的时序；若接收 IN 信令前，IN 端口设置及 Tx FIFO 均就绪时，则 IN 端口直接应答数据包；

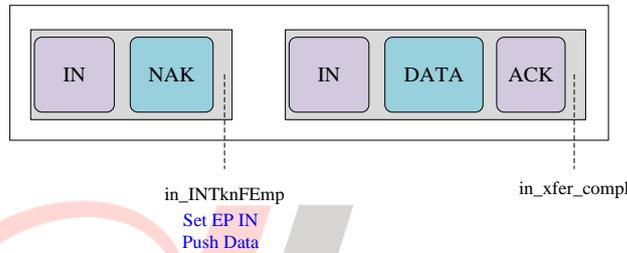


图 44 单次 IN 事务时序

7) 多次 IN 事务时序

设置 IN 端口连续发送多个包后，应用软件在 Tx FEmp 中断服务程序中向 Tx FIFO 不断压入数据，当全部数据传输完成后(也即均收到 Host 发送的 ACK)，软核触发完成中断。若在此过程中，Dev 未收到某个 ACK，软核自动重发该数据包，不需要软件干预；

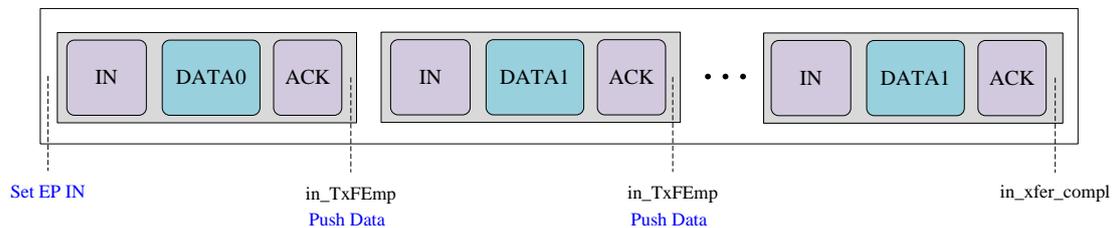


图 45 多次 IN 事务时序

5.6.4 寄存器概述

寄存器表 19 SDIO 寄存器概述（基地址：0x1500\_0000）

偏移地址	名称	描述	页码
0x008	GAHBCFG	AHB Configuration, 全局 AHB 配置寄存器	
0x00c	GUSBCFG	USB Configuration, 全局 USB 配置寄存器	
0x010	GRSTCTL	Reset, 全局复位控制寄存器	
0x014	GINTSTS	Interrupt, 全局中断状态寄存器	
0x018	GINTMSK	Interrupt Mask, 全局中断屏蔽控制寄存器	
0x01c	GRXSTSR	Receive Status Debug Read, 全局接收状态读取寄存器	
0x020	GRXSTSP	Status Read and Pop, 全局接收状态读出寄存器	
0x024	GRXFISZ	Receive FIFO Size, RxFIFO 分配控制寄存器	
0x028	GNPTXFISZ	Non-Periodic Transmit FIFO Size, Non-Periodic TxFIFO 分配控制寄存器	
0x038	GGPIO	General Purpose IO, 通用 IO 寄存器	
0x03c	GUID	User ID, 用户 ID 寄存器	
0x040	GSNPSID	Synopsys ID, 版本号只读寄存器(0x4f54_270a)	
0x044	GHWCFG1	User HW Config1, 软核配置状态寄存器 1	
0x048	GHWCFG2	User HW Config2, 软核配置状态寄存器 2	
0x04c	GHWCFG3	User HW Config3, 软核配置状态寄存器 3	
0x050	GHWCFG4	User HW Config4, 软核配置状态寄存器 4	
0x054	GUSRSTS	User States, 用户定义观察信号	
0x104	DIEPTXF1	IN Endpoint1 TxFIFO Depth, 端口 1 的 TxFIFO 深度配置寄存器	
0x108	DIEPTXF3	IN Endpoint3 TxFIFO Depth, 端口 3 的 TxFIFO 深度配置寄存器	
0x800	DCFG	Device Configuration, 设备总体配置寄存器	
0x804	DCTL	Device Control, 设备总体控制寄存器	
0x808	DSTS	Device Status, 设备总体状态寄存器	
0x810	DIEPMSK	Device IN Endpoint Common Interrupt Mask, 设备 IN 端口总体中断屏蔽	
0x814	DOEPMSK	Device OUT Endpoint Common Interrupt Mask, 设备 OUT 端口总体中断屏蔽寄存器	
0x818	DAINT	Device All Endpoints Interrupt, 设备全部端口中断状态寄存器	
0x81c	DAINTMSK	Device All Endpoints Interrupt Mask, 设备全部端口中断屏蔽控制寄存器	
0x830	DTHRCTL	Device Threshold Control, 设备门限控制寄存器	
0x834	DIEPEMPMSK	Device IN Endpoint FIFO Empty Interrupt Mask, 设备 IN 端口 FIFO 空中断屏蔽寄存器	
0x900	DIEPCTL0	Device Control IN Endpoint0 Control, 设备 IN EP0 控制寄存器	
0x908	DIEPINT0	Device IN Endpoint0 Interrupt, 设备 IN EP0 中断状态寄存器	
0x910	DIEPTSIZ0	Device IN Endpoint0 Transfer Size, 设备 IN EP0	

		传输大小设置寄存器	
0x914	DIEPDMA0	Device IN Endpoint0 DMA Addr, 设备 IN EP0 DMA 地址	
0x918	DTXFSTS0	Device IN Endpoint0 Transmit FIFO Status, 设备 IN EP0 TxFIFO 状态	
0xb00	DOEPCTRL0	Device Control OUT Endpoint0 Control, 设备 OUT EP0 控制寄存器	
0xb08	DOEPINT0	Device OUT Endpoint0 Interrupt, 设备 OUT EP0 中断状态寄存器	
0xb10	DOEPTSIZE0	Device OUT Endpoint0 Transfer Size, 设备 OUT EP0 传输大小设置寄存器	
0xb14	DOEPDMA0	Device OUT Endpoint0 DMA Addr, 设备 OUT EP0 DMA 地址	
0x920	DIEPCTL1	Device Control IN Endpoint1 Control, 设备 IN EP1 控制寄存器	
0x928	DIEPINT1	Device IN Endpoint1 Interrupt, 设备 IN EP1 中断状态寄存器	
0x930	DIEPTSIZE1	Device IN Endpoint1 Transfer Size, 设备 IN EP1 传输大小设置寄存器	
0x934	DIEPDMA1	Device IN Endpoint1 DMA Addr, 设备 IN EP1 DMA 地址	
0x938	DTXFSTS1	Device IN Endpoint1 Transmit FIFO Status, 设备 IN EP1 TxFIFO 状态	
0xb40	DOEPCTRL2	Device Control OUT Endpoint2 Control, 设备 OUT EP2 控制寄存器	
0xb48	DOEPINT2	Device OUT Endpoint2 Interrupt, 设备 OUT EP2 中断状态寄存器	
0xb50	DOEPTSIZE2	Device OUT Endpoint2 Transfer Size, 设备 OUT EP2 传输大小设置寄存器	
0xb54	DOEPDMA2	Device OUT Endpoint2 DMA Addr, 设备 OUT EP2 DMA 地址	
0x960	DIEPCTL3	Device Control IN Endpoint3 Control, 设备 IN EP3 控制寄存器	
0x968	DIEPINT3	Device IN Endpoint3 Interrupt, 设备 IN EP3 中断状态寄存器	
0x970	DIEPTSIZE3	Device IN Endpoint3 Transfer Size, 设备 IN EP3 传输大小设置寄存器	
0x974	DIEPDMA3	Device IN Endpoint3 DMA Addr, 设备 IN EP3 DMA 地址	
0x978	DTXFSTS3	Device IN Endpoint3 Transmit FIFO Status, 设备 IN EP3 TxFIFO 状态	
0xb80	DOEPCTRL4	Device Control OUT Endpoint4 Control, 设备 OUT EP4 控制寄存器	

0xb88	DOEPINT4	Device OUT Endpoint4 Interrupt, 设备 OUT EP4 中断状态寄存器	
0xb90	DOEPTSIZ4	Device OUT Endpoint4 Transfer Size, 设备 OUT EP4 传输大小设置寄存器	
0xb94	DOEPDMA4	Device OUT Endpoint4 DMA Addr, 设备 OUT EP4 DMA 地址	
0xe00	PCGCCTL	Power and Clock Gating Control, 电源和时钟门控制寄存器	
0x1000	DEVEP0FF	EP0 FIFO PUSH/POP 入口地址	
0x2000	DEVEP1FF	EP1 FIFO PUSH/POP 入口地址	
0x3000	DEVEP2FF	EP2 FIFO PUSH/POP 入口地址	
0x4000	DEVEP3FF	EP3 FIFO PUSH/POP 入口地址	
0x5000	DEVEP4FF	EP4 FIFO PUSH/POP 入口地址	

## 5.6.5 寄存器描述

### 5.6.5.1 GAHBCFG

偏移地址: 0x008

复位方式: H/S

Bits	Access	Name	Default	Description
[31:9]	-	reserved	0x0	保留
[8]	-	Non-usage	0x0	不使用
[7]	RW	NPTxFEmpLvl	0x0	Non-Periodic Tx FIFO Empty Level, 专用FIFO dev模式下 0: DIENPINTn.TxFEmp时, 标示IN EP Tx FIFO半满 1: DIENPINTn.TxFEmp时, 标示IN EP Tx FIFO全满
[6]	-	reserved	0x0	保留
[5]	RW	DMAEn	0x0	DMA Enable 0: Slave模式; 1: DMA模式
[4:1]	RW	HBstLen	0x0	Burst Length/Type, 配置DMA模式下dma_burst[3:0] 内部DMA配置时 4'h0: Single 4'h1: INCR 4'h3: INCR4 4'h5: INCR8 4'h7: INCR16 其他: 保留
[0]	RW	GlbIntrMsk	0x0	Global Interrupt Mask 0: 屏蔽所有中断 1: 非屏蔽所有中断

### 5.6.5.2 GUSBCFG

偏移地址：0x00c

复位方式：H/S

Bits	Access	Name	Default	Description
[31]	RW	corrtxpkt	0x0	Corrupt Tx Packet 仅用于调试模式，禁止置1
[30:29]	-	Non-usage	0x0	不使用
[28:26]	-	reserved	0x0	保留
[25:15]	RW	Non-usage	0x0	Non-usage
[14]	-	reserved	0x0	保留
[13:10]	RW	USBTrdTim	0x5	USB Turnaround Time 计算公式：(4*AHB_CLK + 1*PHY_CLK)/PHY_CLK
[9:7]	RW	Non-usage	0x0	Non-usage
[6]	RO	PHYSel	0x1	USB2.0 HSPHY or USB1.1 FS Serial Transceiver 1: USB1.1 FS Serial Transceiver 0: USB2.0高速PHY
[5]	RW	FSIntf	0x0	FS Serial Interace Select 0: 6-pin单向FS接口 1: 3-pin双向FS接口
[4:3]	RW	Non-usage	0x0	Non-usage
[2:0]	RW	TOutCal	0x0	HS/FS Timeout Calibration(USB定义FS模式timeout值为16-18个bit time，该段为兼容不同PHY延时而增加的延时) 计算公式：0.25bit time * 该值

### 5.6.5.3 GRSTCTL

偏移地址：0x010

复位方式：H/S

Bits	Access	Name	Default	Description
[31]	RO	AHBIdle	0x1	AHB Master Idle，标示AHB Master状态机处于idle态
[30]	RO	DMAReq	0x0	DMA Request Signal，标示DMA Req请求中，用于调试
[29:11]	-	reserved	0x0	保留
[10:6]	RW	FxFNum	0x0	TxFIFO Number 'h0: 复位host模式Non-periodic TxFIFO 复位dev模式Non-periodic TxFIFO(共享) 复位dev模式TxFIFO0(专用) 'h1: 复位host模式Periodic TxFIFO 复位dev模式Periodic TxFIFO1(共享) 复位dev模式TxFIFO1(专用) 'h2: 复位dev模式Periodic TxFIFO2(共享) 复位dev模式TxFIFO2(专用) ...

				'hf: 复位dev模式Periodic TxFIFO15(共享) 复位dev模式TxFIFO15(专用) 'h10: 复位全部TxFIFO
[5]	WS/SC	TxFFLSH	0x0	TxFIFO复位使能 置位前, 必须确认TxFIFO无读写操作, 而且应用必须在该位被自动清除后(8个phy_clk或hclk), 才能进行新的读写操作。
[4]	WS/SC	RxFFLSH	0x0	RxFIFO复位使能 置位前, 必须确认RxFIFO无读写操作, 而且应用必须在该位被自动清除后(8个phy_clk或hclk), 才能进行新的读写操作。
[3:2]	-	Non-usage	0x0	不使用
[1]	WS/SC	HSftRst	0x0	hclk soft reset, hclk时钟域软复位
[0]	WS/SC	CSftRst	0x0	hclk和phy_clk时钟域软复位

#### 5.6.5.4 GINTSTS

偏移地址: 0x014

复位方式: H/S

Bits	Access	Name	Default	Description
[31]	RO/WC	WkUpInt	0x0	Resume/Remote Wakeup Dectected中断 Dev模式: 监测到USB Resume时置位
[30:29]	-	Non-usage	0x5	不使用
[28]	RO/WC	ConIDStsChng	0x0	Connector ID Status Change中断
[27]	-	reserved	0x0	保留
[26:24]	-	Non-usage	0x4	不使用
[23]	-	reserved	0x0	保留
[22]	RO/WC	FetSusp	0x0	Data Fetch Suspended, 仅在DMA模式下有效
[21]	RO/WC	incomplSOOUT	0x0	Incomplete Isochronous OUT Transfer(dev-only) dev模式下, 至少有一个ISO OUT EP在当前帧内未完成传输, 和EOPF中断同时置位
[20]	RO/WC	incomplSOIN	0x0	Incomplete Isochronous IN Transfer(dev-only) dev模式下, 至少有一个ISO IN EP在当前帧内未完成传输, 和EOPF中断同时置位
[19]	RO	OEPInt	0x0	OUT Endpoints中断(dev-only) dev模式下, 若有OUT EP中断未处理时置位, 应用须读取DAINT和DOEPINTn判断处理具体中断事件
[18]	RO	IEPInt	0x0	IN Endpoints中断(dev-only) dev模式下, 若有IN EP中断未处理时置位, 应用须读取DAINT和DIEPINTn判断处理具体中断事件
[17]	RO/WC	EPMis	0x0	Endpoint Mismatch中断(dev-only) 只有在共享FIFO操作时有效。接收到IN token后, FIFO内的数据不匹配时, 触发
[16]	-	reserved	0x0	保留

[15]	RO/WC	EOPF	0x0	End of Periodic Frame中断(dev-only) 达到DCFG.PerFrInt设置的Periodic Frame Interval周期个数时触发
[14]	RO/WC	ISOOutDrip	0x0	ISO OUT Packet Dropped中断(dev-only) 当RxFIFO因空间不足导致软核不能接收ISO OUT包时触发
[13]	RO/WC	EnumDone	0x0	Enumeration Done中断(dev-only) 速度枚举结束后触发,应用必须读取DSTS获取枚举结果
[12]	RO/WC	USBRst	0x0	USB Reset中断(dev-only) 软核监测到USB reset信号后触发
[11]	RO/WC	USBSusp	0x0	USB Suspend中断(dev-only) 软核监测到suspend态后触发
[10]	RO/WC	ErlySusp	0x0	Early Suspend中断(dev-only) 软核监测到USB总线处于idle态时长超过3ms后触发
[9:8]	-	Non-usage	0x0	不使用
[7]	RO	GOUTNakEff	0x0	Global OUT NAK Effective中断(dev-only) DCTL.SGOUTNak置位时触发, DCTL.CGOUTNak置位时清除
[6]	RO	GINNakEff	0x0	Global IN Non-periodic NAK Effective中断(dev-only) DCTL.SGNPInNak置位时触发, DCTL.CGNPInNak置位时清除
[5]	RO	Non-usage	0x1	Non-Periodic TxFIFO Empty中断 仅在共享TxFIFO模式时有效
[4]	RO	RxFLvl	0x0	RxFIFO Non-Empty RxFIFO内至少有一个packet数据等待读取时触发
[3]	RO/WC	Sof	0x0	Start of Frame中断 dev模式下,软核接收到SOF时触发,应用随之读取DSTS获取帧号
[2]	RO	OTGInt	0x0	OTG中断 OTG协议事件发生时触发
[1]	RO/WC	ModeMis	0x0	Mode Mismatch中断 dev模式下的寄存器被用作host模式时触发,且AHB总线应答OKAY,但软核忽略该次寄存器访问操作
[0]	RO	CurMod	0x0	Current Mode of Operation 0: dev, 1: host

#### 5.6.5.5 GINTMSK

偏移地址: 0x018

复位方式: H/S

Bits	Access	Name	Default	Description
[31]	RW	WkUpIntMsk	0x0	Resume/Remote Wakeup Dectected中断屏蔽 0: 屏蔽, 1:使能 (下同)

[30:29]	RW	Non-usage	0x0	不使用
[28]	RW	ConIDStsChngMsk	0x0	Connector ID Status Change中断屏蔽
[27]	-	reserved	0x0	保留
[26:24]	RW	Non-usage	0x0	不使用
[23]	-	reserved	0x0	保留
[22]	RW	FetSuspMsk	0x0	Data Fetch Suspended中断屏蔽, 仅在DMA模式下有效
[21]	RW	incomplSOOUTMsk	0x0	Incomplete Isochronous OUT Transfer中断屏蔽 (dev-only)
[20]	RW	incompleSOINMsk	0x0	Incomplete Isochronous IN Transfer中断屏蔽(dev-only)
[19]	RW	OEPIntMsk	0x0	OUT Endpoints中断屏蔽(dev-only)
[18]	RW	IEPIntMsk	0x0	IN Endpoints中断屏蔽(dev-only)
[17]	RW	EPMisMsk	0x0	Endpoint Mismatch中断屏蔽(dev-only)
[16]	-	reserved	0x0	保留
[15]	RW	EOPFMsk	0x0	End of Periodic Frame中断屏蔽(dev-only)
[14]	RW	ISOOutDripMsk	0x0	ISO OUT Packet Dropped中断屏蔽(dev-only)
[13]	RW	EnumDoneMsk	0x0	Enumeration Done中断屏蔽(dev-only)
[12]	RW	USBRstMsk	0x0	USB Reset中断屏蔽(dev-only)
[11]	RW	USBSuspMsk	0x0	USB Suspend中断屏蔽(dev-only)
[10]	RW	ErlySuspMsk	0x0	Early Suspend中断屏蔽(dev-only)
[9:8]	RW	Non-usage	0x0	不使用
[7]	RW	GOUTNakEffMsk	0x0	Global OUT NAK Effective中断屏蔽(dev-only)
[6]	RW	GINNakEffMsk	0x0	Global IN Non-periodic NAK Effective中断屏蔽 (dev-only)
[5]	RW	NPTxFEmpMsk	0x0	Non-Periodic Tx FIFO Empty中断屏蔽
[4]	RW	RxFLvIMsk	0x0	Rx FIFO Non-Empty屏蔽
[3]	RW	SofMsk	0x0	Start of Frame中断屏蔽
[2]	RW	OTGIntMsk	0x0	OTG中断屏蔽
[1]	RW	ModeMisMsk	0x0	Mode Mismatch中断屏蔽
[0]	-	reserved	0x0	保留

#### 5.6.5.6 GRXSTSR/ GRXSTSP

偏移地址: 0x01c/0x020

复位方式: H/S

该寄存器实体存储在 Rx FIFO 中, 读取 GRXSTSR 和 GRXSTSP 寄存器时返回值完全一致, 区别在于读取 GRXSTSP 将顺便将该 32 位头部信息从 Rx FIFO 中 POP 出来, 而 GRXSTSR 仅仅是读取。

Bits	Access	Name	Default	Description
[31:25]	-	reserved	0x0	保留
[24:21]	RO	FrmNum	0x0	Frame Number, ISO OUT 帧号低四位
[20:17]	RO	PktSts	0x0	Packet Status, 标示接收包状态 h1: Global OUT NAK(触发中断)

				'h2: 接收到OUT包 'h3: OUT传输结束(触发中断) 'h4: SETUP传输结束(触发中断) 'h6: 接收到SETUP包 其他: 保留
[16:15]	RO	DPID	0x0	Data PID, 标示接收包状态 'h0: DATA0 'h1: DATA2 'h2: DATA1 'h3: MDATA
[14:4]	RO	BCnt	0x0	Packet Status, 标示接收包数据长度
[3:0]	RO	EPNum	0x0	Endpoint Number, 当前数据的端口号

注: GRXSTSP 仅用于 POP 包头信息, 似乎不可以 POP 包内数据信息, 数据信息需要 POP 相应 Rx FIFO 地址

### 5.6.5.7 GRXFSIZ

偏移地址: 0x024

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved	0x0	保留
[15:0]	RW	RxFDep	0x80	RxFIFO深度(宽度35位), min=16 该段根据SPRAM实际深度可调

### 5.6.5.8 GNPTXFSIZ

偏移地址: 0x028

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	RW	INEPTxF0Dep	0x80	IN EP Tx FIFO深度(宽度35位), min=16, 深度可调
[15:0]	RW	INEPTxF0StAddr	0x80	IN EP Tx FIFO起始地址, 可调

### 5.6.5.9 GGPIO

偏移地址: 0x038

复位方式: H/S

Bits	Access	Name	Default	Description
[31]	-	Reserved	0x0	保留
[30]	RW	PAD RcvEn	0x1	PAD差分接收使能 0: 无效; 1: 使能
[29:28]	RW	PADDPPullUp	0x0	PAD DP线DPPU控制 'h0/1: 软核dp_rpu1_en信号控制 'h2: 强制上拉使能 'h3: 强制上拉无效

[27:26]	RW	PADDPPullUpLO	0x0	PAD DP线DPPULO控制 'h0: 软核dp_rpu2_en信号控制 'h1: 软核dp_rpu1_en信号控制 'h2: 强制上拉使能 'h3: 强制上拉无效
[25:24]	RW	PADDMPullUp	0x0	PAD DM线DMPU控制 'h0/1: 软核dm_rpu_en信号控制 'h2: 强制上拉使能 'h3: 强制上拉无效
[23:22]	RW	PADDMPullUpLO	0x0	PAD DM线DMPULO控制 'h0/1: 软核dm_rpu_en信号控制 'h2: 强制上拉使能 'h3: 强制上拉无效
[21]	RW	PADPullDown	0x0	PAD下拉电阻控制 0: 下拉无效; 1: 下拉使能
[20]	-	Reserved	0x0	保留
[19]	RW	PADSpeed	0x1	PAD速率选择 0: LS模式; 1: FS模式
[18:17]	RW	PADSuspend	0x0	PAD挂起控制 'h0/1: 软核控制Suspend信号 'h2: 强制挂起使能 'h3: 强制挂起无效
[16]	RW	PADPowerDown	0x0	PAD掉线 0: 上电; 1: 掉电
[15:0]	-	gp_in	0x0	未使用

#### 5.6.5.10 GUID

偏移地址: 0x03c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	User ID	0x4570_0 322	用户自定义ID号

#### 5.6.5.11 GSYNPSID

偏移地址: 0x040

复位方式: N/A

Bits	Access	Name	Default	Description
[31:0]	RO	SynopsysID	0x4f54_270a	USB软核版本号

### 5.6.5.12 GHWCFG1

偏移地址：0x044

复位方式：

Bits	Access	Name	Default	Description
[31:10]	RO	Non-usage	0x0	EP方向，两比特表示EP方向 'h0: IN/OUT, 'h1: IN, 'h2: OUT, 'h3: 保留
[9:8]	RO	EP4Dir	0x2	EP4方向：OUT
[7:6]	RO	EP3Dir	0x1	EP3方向：IN
[5:4]	RO	EP2Dir	0x2	EP2方向：OUT
[3:2]	RO	EP1Dir	0x1	EP1方向：IN
[1:0]	RO	EP0Dir	0x0	EP0方向：IN/OUT

### 5.6.5.13 GHWCFG2

偏移地址：0x048

复位方式：

Bits	Access	Name	Default	Description
[31]	-	reserved	0x0	保留
[30:22]	RO	Non-usage	0x8a	不使用
[21:20]	-	reserved	0x0	保留
[19]	RO	DynFifoSizing	0x1	支持动态FIFO分配
[18:14]	RO	Non-usage	0x3	不使用
[13:10]	RO	NumDevEps	0x4	支持4个Dev端口(除EP0外)
[9:8]	RO	FSPhyType	0x1	FS PHY接口类型：专用FS接口
[7:6]	RO	HSPhyType	0x0	HS PHY接口类型：不支持HS PHY
[5]	RO	SingPnt	0x0	Point-to-Point(不使用)
[4:3]	RO	OtgArch	0x2	软核架构：内部DMA
[2:0]	RO	OtgMode	0x4	工作模式：Non-OTG Dev

### 5.6.5.14 GHWCFG3

偏移地址：0x04c

复位方式：

Bits	Access	Name	Default	Description
[31:16]	RO	DfifoDepth	0x7a	DFIFO有效深度：122
[15:12]	-	reserved	0x0	保留
[11]	RO	RstType	0x0	软核复位模式：异步复位
[10]	RO	OptFeature	0x1	移除软核可选功能：无可选功能
[9]	RO	VndctlSupt	0x0	是否支持Vendor控制接口：不支持
[8]	RO	I2CINtSel	0x0	是否支持I2C接口：不支持
[7]	RO	OtgEn	0x0	是否支持OTG：不支持

[6:4]	RO	PktSizeWidth	0x0	包计数器位宽：4位
[3:0]	RO	XferSizeWidth	0x0	传输字节计数器位宽：11位

#### 5.6.5.15 GHWCFG4

偏移地址：0x050

复位方式：

Bits	Access	Name	Default	Description
[31]	RO	SGDMA	0x0	是否支持SG-DMA动态配置：不支持
[30]	RO	SGDMAConfig	0x0	是否有SG-DMA配置：无
[29:26]	RO	INEPs	0x2	支持3个IN端口数目(包括EP0)
[25]	RO	DedFifoMode	0x1	专用TxFIFO：是
[24:20]	RO	Non-usage	0x3	不使用
[19:16]	RO	NumCtlEps	0x0	支持0个控制EP数目(除EP0外)
[15:14]	RO	PhyDataWidth	0x0	PHY接口位宽：8位
[13:6]	-	reserved	0x0	保留
[5]	RO	AhbFreq	0x1	AHB最小时钟是否大于60MHz：是
[4]	RO	EnablePwrOpt	0x0	是否支持功耗优化：不支持
[3:0]	RO	Non-usage	0x0	不使用

#### 5.6.5.16 GUSRSTS

偏移地址：0x054

复位方式：

Bits	Access	Name	Default	Description
[31:24]	RO	RxRevSpc	0x10	RxFIFO预留给CTRL传输的空间，双字单位
[23:16]	RO	RxAvailSpc	0x70	RxFIFO非CTRL传输可用空间，双字单位
[15]	-	Reserved	0x0	保留
[14]	RO	DFBusy	0x0	FIFO忙碌 1：忙碌，0：空闲
[13:9]	RO	RxPktNum	0x0	RxFIFO内包数目
[8:5]	RO	RxTknPID	0x0	接收包PID 4'h:1 OUT 4'h2: ACK 4'h3: DATA0 4'h6: SOF 4'h9: IN 4'ha: NAK 4'hb: DATA1 4'hc: PRE 4'hd: SETUP 4'he: STALL Ohters: Reserved

[4:0]	RO	TxRxFSMState	0x0	MAC层状态机当前状态 5'h0: IDLE 5'h1-5: 不使用 (Host模式) 5'h6: Receive Error 5'h7: Receive Data 5'h8: Send Handshake 5'h9: Send EOP 5'ha: Send Data 5'hb: Wait Handshake 5'hc: Check CRC16 5'hd: Wait Data 5'he: Test Mode 5'hf: Wait Bus Idle 5'h10: Wait Turnaround 5'h11: Data To Status Others: Reserved
-------	----	--------------	-----	--

### 5.6.5.17 DIEPTXF1/3

偏移地址: 0x104/0x108

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	RW	INEPnTxFDep	0x80	IN EP TxFIFO深度(有效位宽32), 可配置
[15:0]	RW	INEPnTxFStAddr	0x100	IN EP FIFO起始地址, DIEPTXF3: 0x180

注: 详细配置方法参加第节

### 5.6.5.18 DCFG

偏移地址: 0x800

复位方式: H/S

Bits	Access	Name	Default	Description
[31:26]	-	reserved	0x0	保留
[25:18]	-	Non-usage	0x8	不使用
[17:13]	-	reserved	0x0	保留
[12:11]	RW	PerFrInt	0x0	Periodic Frame Interval, 从SOF到软核置位End of Periodic Frame中断通的时间 'h0: 80% of the frame interval 'h1: 85% of the frame interval 'h2: 90% of the frame interval 'h3: 95% of the frame interval
[10:4]	RW	DevAddr	0x0	Dev地址, 应用软件必须在接收到SetAddr指令后设置该段
[3]	-	reserved	0x0	保留

[2]	RW	NZStsOUTHshk	0x0	长度非零的Status OUT握手方式 配置在控制传输中接收到长度非零的OUT事务时，软核在Status Stage的应答方式 1: 回应STALL包，并且将数据上报到应用 0: 将数据上报到应用，并根据DCTL.NAK/STALL回应握手包
[1:0]	RW	DevSpd	0x0	Dev速度 'h0: HS(USB2.0 PHY CLK=30/60MHz) 'h1: FS(USB2.0 PHY CLK=30/60MHz) 'h2: LS(USB1.1 PHY CLK=6MHz) 'h3: FS(USB1.1 PHY CLK=48MHz)

### 5.6.5.19 DCTL

偏移地址: 0x804

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved	0x0	保留
[15:13]	-	Non-usage	0x0	不使用
[12]	-	reserved	0x0	保留
[11]	RW	PWROnPrgDone	0x0	Power-On Programming Done 标示软核从power down模式唤醒后寄存器配置完成
[10]	WO	CGOUTNak	0x0	Clear Global OUT NAK 清除Global OUT NAK
[9]	WO	SGOUTNak	0x0	Set Global OUT NAK 置位后，软核中所有OUT端口应该NAK包使能该位前，务必清除GINTSTS.GOUTNakEff
[8]	WO	CGNPInNak	0x0	Clear Global Non-periodic IN NAK 清除Global Non-periodic IN NAK位
[7]	WO	SGNPInNak	0x0	Set Global Non-periodic IN NAK 置位后，软核中所有IN端口应该NAK包使能该位前，务必清除GINTSTS.GINNakEff
[6:4]	RW	TstCtl	0x0	测试控制 'h0: 测试模式无效 'h1: 测试J模式 'h2: 测试K模式 'h3: 测试SE0_NAK模式 'h4: 测试Packet模式 'h5: 测试Force_Enable模式 其他: 保留
[3]	RO	GOUTNakSts	0x0	Global OUT NAK状态 0: 根据FIFO状态和NAK/STAL设置已经发生握手包 1: 无数据写入RxFIOF。除SETUP事务外，向所有包

				应答NAK，丢弃所有ISO OUT。
[2]	RO	GNPINNakSts	0x0	Global Non-periodic IN NAK状态 0: 根据FIFO状态已经发生握手包 1: 所有non-periodic IN端口应答NAK
[1]	RW	SftDiscon	0x0	Soft Disconnect 置位后，软核处于非连接状态，不再有数据交互
[0]	RW	RmtWkUpSig	0x0	Remote Wakeup Signaling 置位后，软核发起远程信号唤醒USB Host，应用必须通过设置该位使得软核退出挂起状态。根据USB2.0规范，应用必须在设置该位后1-15ms清除。

### 5.6.5.20 DSTS

偏移地址：0x808

复位方式：H/S

Bits	Access	Name	Default	Description
[31:22]	-	reserved	0x0	保留
[21:8]	RO	SOFFrmNum	0x0	Frame Number of Received SOF，接收SOF帧号
[7:4]	-	reserved	0x0	保留
[3]	RO	ErrticErr	0x0	不确定错误 UTMI PHY状态错误上报该错误，同时软件进入挂起状态，且GINTSTS.ErlySusp置位。此时应用软件只能通过soft disconnect恢复
[2:1]	RO	EnumSpd	0x1	Enumerated Speed，标示软核上电枚举后的工作频率 'h0: HS(USB2.0 PHY CLK=30/60MHz) 'h1: FS(USB2.0 PHY CLK=30/60MHz) 'h2: LS(USB1.1 PHY CLK=6MHz) 'h3: FS(USB1.1 PHY CLK=48MHz)
[0]	RO	SuspSts	0x0	Suspend Status Dev模式下，软核监测到挂起条件后置位，软核随后进入挂起状态。下述任意一个条件下，软核退出挂起态： • phy_lin_state_i信号线上有电平变化； • DCTL.RmtWkUpSig置位

### 5.6.5.21 DIEPMSK

偏移地址：0x810

复位方式：H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved	0x0	保留
[9]	RW	BNAIntrMsk	0x0	Buffer Not Available中断屏蔽，0: 屏蔽, 1:使能(下同)
[8]	RW	TxfifoUndrnMsk	0x0	FIFO Underrun屏蔽

[7]	-	reserved	0x0	保留
[6]	RW	INEPNakEffMSK	0x0	IN Endpoint NAK Effective Mask
[5]	RW	INTknEPMisMsk	0x0	IN Token received with EP Mismatch Mask
[4]	RW	INTknTXFEEmpMsk	0x0	IN Token Received When TxFIFO Empty Mask
[3]	RW	TimeOUTMsk	0x0	Timeout Condition Mask
[2]	RW	AHBErrMsk	0x0	AHB Error Mask, DMA时有效
[1]	RW	EPDisbldMsk	0x0	Endpoint Disabled Interrupt Mask
[0]	RW	XferComplMsk	0x0	Transfer Completed Interrupt Mask

### 5.6.5.22 DOEPMASK

偏移地址: 0x814

复位方式: H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved	0x0	保留
[9]	RW	BNAInIntrMsk	0x0	BNA中断屏蔽, 0: 屏蔽, 1:使能(下同)
[8]	RW	OutPktErrMsk	0x0	OUT Packet Error Mask
[7]	-	reserved	0x0	保留
[6]	RW	Back2BackSETup	0x0	Back-to-Back Packets Received Mask
[5]	-	reserved	0x0	保留
[4]	RW	OUTTknEPdisMsk	0x0	OUT Token Received when Endpoint Disabled mask
[3]	RW	SetUPMsk	0x0	SETUP Phase Done Mask, 仅用于控制端口
[2]	RW	AHBErrMsk	0x0	AHB Error Mask
[1]	RW	EPDisbldMsk	0x0	Endpoint Disabled Interrupt Mask
[0]	RW	XferComplMsk	0x0	Transfer Completed Interrupt Mask

### 5.6.5.23 DAINTE

偏移地址: 0x818

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	RO	OutEPInt	0x0	每个位标示一个OUT端口 bit16: OUT EP0 bit31: OUT EP15
[15:0]	RO	InEpInt	0x0	每个位标示一个IN端口 bit0: IN EP0 bit15: IN EP15

#### 5.6.5.24 DAINMSK

偏移地址: 0x81c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	RO	OutEPIntMSK	0x0	每个位标示一个OUT端口, 0: 屏蔽, 1:使能 (下同) bit16: OUT EP0 bit31: OUT EP15
[15:0]	RO	InEpIntMSK	0x0	每个位标示一个IN端口 bit0: IN EP0 bit15: IN EP15

#### 5.6.5.25 DTHRCTL

偏移地址: 0x830

复位方式: H/S

Bits	Access	Name	Default	Description
[31:28]	-	reserved	0x0	保留
[27]	RW	ArbPrkEn	0x1	Arbiter Parking Enable 控制内部DMA暂停IN端口的仲裁
[26]	-	reserved	0x0	保留
[25:17]	RW	RxThrLen	0x8	Receive Threshold Length 接收门限, 最小值为8个DWord(32bit)
[16]	RW	RxThrEn	0x0	Receive Threshold Enable 接收门限使能
[15:11]	-	reserved	0x0	保留
[10:2]	RW	TxThrLen	0x8	TransmitThreshold Length 发送门限, 最小值为8个DWord(32bit)
[1]	RW	ISOThrEn	0x0	ISO IN Endpoints Threshold Enable ISO IN端口接收门限使能
[0]	RW	NonISOThrEn	0x0	Non-ISO IN Endpoints Threshold Enable 非ISO IN端口接收门限使能

#### 5.6.5.26 DIEPMPMSK

偏移地址: 0x834

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved	0x0	保留
[15:0]	RO	InEpTxfEmpMsk	0x0	每个位标示一个IN端口, 用于控制DIEPINn.TxfEmp 0: 屏蔽, 1:使能 bit0: IN EP0

				bit15: IN EP15
--	--	--	--	----------------

### 5.6.5.27 DIEPCTL0

偏移地址: 0x900

复位方式: H/S

Bits	Access	Name	Default	Description
[31]	WS/SC	EPEna	0x0	Endpoint使能 在非SG-DMA模式下, 置位时表示数据就绪 软核在置位下述任意中断前自动清除该位: endpoint disable transfer completed
[30]	WS/SC	EPDis	0x0	Endpoint disable, 只能在置位EPEna后使用 置位时强制终止数据传输(即便数据尚未结束), 软核在 置位Endpoint disable中断前自动清除该位
[29:28]	-	reserved	0x0	保留
[27]	WO	SNAK	0x0	Set NAK 置位后, 该控制端口应答NAK
[26]	WO	CNAK	0x0	Clear NAK, 清除SNAK位
[25:22]	RW	TxFNum	0x0	指定专用Tx FIFO号 表示分配给IN EP0的Tx FIFO编号, 必须为零
[21]	WS/SC	Stall	0x0	Stall握手 应用只能在接收到SETUP信令后, 才能设置该位。该 位优先级高于NAK、Global Non-Periodic IN NAK和 Global OUT NAK位
[20]	-	reserved	0x0	保留
[19:18]	RO	EPTType	0x0	端口类型, EP0只用于控制
[17]	RO	NAKSts	0x0	NAK状态 0: 软核根据FIFO状态发出non-NAK握手包 1: 软核发出NAK握手包 当置位后, 软核强制终止数据传输
[16]	-	reserved	0x0	保留
[15]	RO	USBActEP	0x1	USB Active端口, EP0总是处于激活状态
[14:11]	RW	NextEp	0x0	下个端口(仅用于共享FIFO配置, non-usage)
[10:2]	-	reserved	0x0	保留
[1:0]	RW	MPS	0x0	Maximum Packet Size h0: 64字节 h1: 32字节 h2: 16字节 h3: 8字节

### 5.6.5.28 DIEPINT0/1/3

偏移地址：0x908/0x928/0x968

复位方式：H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved	0x0	保留
[9]	RO/WC	Non-usage	0x0	不使用
[8]	RO/WC	TxfifoUndrn	0x0	Set DATA1 PID, IN 仅在专用TxFIFO且有门限的配置下有效
[7]	RO	TxFEmp	0x0	TxFIFO Empty 当TxFIFO半空或全空时，触发该中断。 GAHBCFG.NPTXFempLvl决定半空或全空中断。
[6]	RO/WC	INEPNakEff	0x0	IN Endpoint NAK Effective 软核监测到NAK为置位时触发该中断，置位 DIEPCTLn.CNAK时清除该中断
[5]	-	INTknEPMis	0x0	接收与EP不匹配的IN Token (non-periodic有效) 端口接收到IN Token时，non-periodic TxFIFO输出口的 数据却不是该EP时触发
[4]	RO/WC	INTknTXFemp	0x0	接收IN Token时TxFIFO空(non-periodic有效) 接收到IN指令时，对应EP的TxFIFO空
[3]	RO/WC	Timeout	0x0	Timeout Condition，适用条件如下： 专用TxFIFO配置：CTRL 软核监测到USB最后一个IN指令超时后触发
[2]	RO/WC	AHBErr	0x0	AHB Error 在内部DMA模式下，AHB读写出错时触发
[1]	RO/WC	EPDisbld	0x0	Endpoint Disable 端口被关闭时触发
[0]	RO/WC	XferCompl	0x0	传输完成 指定数量的传输已完成时触发

### 5.6.5.29 DIEPTSIZ0

偏移地址：0x910

复位方式：H/S

Bits	Access	Name	Default	Description
[31:21]	-	reserved	0x0	保留
[20:19]	RW	PktCnt	0x0	Packet Count EPO本次事务中USB包数目。软核向TxFIFO中发送完 成一个包后，该段自动减1
[18:7]	-	reserved	0x0	保留
[6:0]	RW	XferSize	0x0	Transfer Size EPO本次事务总字节长度，也即PktCnt包长度之和。软 核将发送数据压入TxFIFO后，该段值自动减1，当减

				至0时，触发传输完成中断。
--	--	--	--	---------------

### 5.6.5.30 DIEPDMA0/1/3

偏移地址：0x914/0x934/0x974

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	DMAAddr	0x0	EP0 IN DMA地址

### 5.6.5.31 DTXFSTS0/1/3

偏移地址：0x918/0x938/0x978

复位方式：H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved	0x0	保留
[15:0]	RO	InEpTxFSpcAvail	0x0	相应端口TxFIFO空间可用 'h0: EP0 TxFIFO满 'h1: 剩余1 word ... 'hn: 剩余n words( $0 \leq n \leq 32768$ ) 'h8000: 剩余32768 words 其他: 保留

### 5.6.5.32 DOEPCTL0

偏移地址：0xb00

复位方式：H/S

Bits	Access	Name	Default	Description
[31]	WS/SC	EPEna	0x0	Endpoint使能 在非SG-DMA模式下，置位时表示FIFO分配就绪，可以接收Host数据。 软核在置位下述任意中断前自动清除该位： SETUP Phase Done Transfer Completed Endpoint Disable(非法)
[30]	RO	EPDis	0x0	Endpoint disable, OUT EP0总是处于有效状态，不可无效
[29:28]	-	reserved	0x0	保留
[27]	WO	SNAK	0x0	Set NAK 置位后，该控制端口应答NAK

[26]	WO	CNAK	0x0	Clear NAK, 清除SNAK位
[25:22]	-	reserved	0x0	保留
[21]	WS/SC	Stall	0x0	Stall握手 应用只能在接收到SETUP信令后, 才能设置该位。该位优先级高于NAK、和Global OUT NAK位
[20]	RW	Snp	0x0	Snoop Mode 在Snoop模式下, 软核不检测OUT包是否正确, 而直接将其上报给应用
[19:18]	RO	EPTType	0x0	端口类型, EP0只用于控制
[17]	RO	NAKSts	0x0	NAK状态 0: 软核根据FIFO状态发出non-NAK握手包 1: 软核发出NAK握手包 当置位后, 软核强制终止数据传输
[16]	-	reserved	0x0	保留
[15]	RO	USBActEP	0x1	USB Active端口, EP0总是处于激活状态
[14:2]	-	reserved	0x0	保留
[1:0]	RO	MPS	0x0	Maximum Packet Size, 与DIEPCTL0.MPS一致 h0: 64字节 h1: 32字节 h2: 16字节 h3: 8字节

### 5.6.5.33 DOEPINT0/2/4

偏移地址: 0xb08/0xb48/0xb88

复位方式: H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved	0x0	保留
[9]	RO/WC	Non-usage	0x0	不使用
[8]	RO/WC	OutPktErr	0x0	OUT Packet Error 仅在专用TxFIFO且有门限的配置下有效
[7]	-	Non-usage	0x0	不使用
[6]	RO/WC	Back2BackSETup	0x0	Back2BackSETup(CTRL有效) 软核接收到三个以上连续SETUP包时触发
[5]	-	Non-usage	0x0	不使用
[4]	RO/WC	OUTTknEPdis	0x0	OUT Token Received when EP Disa(CTRL有效) 接收到OUT时, EP尚未使能
[3]	RO/WC	SetUp	0x0	SETUP Phase DONE(CTRL有效) 检测的SETUP Stage完成, 之后无连续的SETUP包时触发。该中断后, 应用可以开始解析SETUP包。
[2]	RO/WC	AHBErr	0x0	AHB Error 在内部DMA模式下, AHB读写出错时触发

[1]	RO/WC	EPDisbld	0x0	Endpoint Disable 端口被关闭时触发
[0]	RO/WC	XferCompl	0x0	传输完成 指定数量的传输已完成

#### 5.6.5.34 DOEPTSIZE

偏移地址：0xb10

复位方式：H/S

Bits	Access	Name	Default	Description
[31]	-	reserved	0x0	保留
[30:29]	RW	SUPCnt	0x0	SETUP Packet Count 端口可以接受的连续SETUP包数目，软核从Rx FIFO读出一个包后，该段自动减1 'h1: 1, 'h2: 2, 'h3: 3
[28:20]	-	reserved	0x0	保留
[19]	RW	PktCnt	0x0	Packet Count EP0本次事务中USB包数目。软核从Rx FIFO中读取完成一个包后，该段自动减为0
[18:7]	-	reserved	0x0	保留
[6:0]	RW	XferSize	0x0	Transfer Size EP0本次事务总字节长度。软核在接受一个字节并从Rx FIFO读出后，该段值自动减1，当减至0时，触发传输完成中断。

#### 5.6.5.35 DOEPDMA0/2/4

偏移地址：0xb14/0xb54/0xb94

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	DMAAddr	0x0	EP0 OUT DMA地址

#### 5.6.5.36 DIEPCTL1/3

偏移地址：0x920/0x960

复位方式：H/S

Bits	Access	Name	Default	Description
[31]	WS/SC	EPEna	0x0	Endpoint使能 置位表示发送数据就绪，且软核在置位下述任意中断前自动清除该位 Endpoint disable

				Transfer completed
[30]	WS/SC	EPDis	0x0	Endpoint Disable 置位时强制终止数据传输(即便数据尚未结束),且软核在置位Endpoint disable中断前自动清除该位。 该位只能在置位EPEna后使用
[29]	WO	SetD1PID	0x0	Set DATA1 PID 将端口Data PID值设置为DATA1
[28]	WO	SetD0PID	0x0	Set DATA0 PID 将端口Data PID值设置为DATA0
[27]	WO	SNAK	0x0	Set NAK 置位后, 该控制端口应答NAK
[26]	WO	CNAK	0x0	Clear NAK, IN/OUT 清除SNAK位
[25:22]	RW	TxFNum	0x0	指定专用Tx FIFO号 配置映射到EP1 IN和EP3 IN的Tx FIFO编号 DIEPCTL1.TxFNum = 0x1 DIEPCTL3.TxFNum = 0x2
[21]	RW	Stall	0x0	Stall握手(non-ISO下有效) 置位后, 端口只应答STALL。该位优先级高于NAK、Global Non-Periodic IN NAK和Global OUT NAK位, 应用清除该位。
[20]	-	Non-usage	0x0	不使用
[19:18]	RO	EPTYPE	0x0	端口传输类型 h0: Control(不支持) h1: ISO h2: Bulk h3: Intra
[17]	RO	NAKSts	0x0	NAK状态 0: 软核根据FIFO状态发出non-NAK握手包 1: 软核发出NAK握手包 当置位后, 数据传输如下: Non-ISO IN端口: 软核强制停止发送任何数据; ISO IN端口: 软核发送长度为零的数据包;
[16]	RO	DPID	0x0	Endpoint Data PID(Intra/Bulk有效) 端口即将接收/发送包的PID信息。在端口使能后, 应用必须为一个接收/发送包设置SetD0PID或SetD1PID信息。 0: DATA0, 1: DATA1
[15]	RW/SC	USBActEP	0x0	USB Active Endpoint 标示该端口在当前配置和接口描述符下是否处于激活状态。应用必须根据SetConfiguration和SetInterface指令, 确定是否配置该位。 软核在监测到USB reset后自动清除该位
[14:11]	-	Non-usage	0x0	不使用

[10:0]	RW	MPS	0x0	Maximum Packet Size 应该软件必须将最大包长度写置该段，以字节为单位
--------	----	-----	-----	--

### 5.6.5.37 DIEPTSIZ1/3

偏移地址：0x930/0x970

复位方式：H/S

Bits	Access	Name	Default	Description
[31]	-	reserved	0x0	保留
[30:29]	RW	MC	0x0	Multi Count 对于Periodic IN端口，该段指定每帧中ISO包数目。软核使用该值计算ISO数据PID。 'h1: 1, 'h2: 2, 'h3: 3 对于non-periodic IN端口(RO)，该段仅用于内部DMA模式。该段指定软核在切换到下一个EP前必须为当前EP准备的包数目。
[28:19]	RW	PktCnt	0x0	Packet Count 指定事务包数目，TxFIFO发出一个包，该段自动减
[18:0]	RW	XferSize	0x0	Transfer Size 端口本次事务总字节长度，每向TxFIFO写入一个字节，该段自动减1，减至零时触发中断

### 5.6.5.38 DOEPCTL2/4

偏移地址：0xb40/b80

复位方式：H/S

Bits	Access	Name	Default	Description
[31]	WS/SC	EPEna	0x0	Endpoint使能 置位时表示数据就绪，软核在置位下述任意中断前自动清除该位： Endpoint disable Transfer completed
[30]	WS/SC	EPDis	0x0	Endpoint disable 置位时强制终止数据传输(即便数据尚未结束)，软核在置位Endpoint Disable中断前自动清除该位。 只能在置位EPEna后使用
[29]	WO	SetD1PID	0x0	Set DATA1 PID(Intr/Bulk有效) 将端口Data PID值设置为DATA1
[28]	WO	SetD0PID	0x0	Set DATA0 PID(INT/Bulk有效) 将端口Data PID值设置为DATA0
[27]	WO	SNAK	0x0	Set NAK 置位后，该控制端口应答NAK

[26]	WO	CNAK	0x0	Clear NAK 清除SNAK位
[25:22]	-	Non-usage	0x0	不使用
[21]	RW	Stall	0x0	Stall握手(Non-ISO/CTRL有效) 置位后, 端口只应答STALL。该位优先级高于NAK、Global Non-Periodic IN NAK和Global OUT NAK位, 应用清除该位。
[20]	RW	SnP	0x0	Snoop Mode 在Snoop模式下, 软核不检测OUT包是否正确, 而直接将其上报给应用
[19:18]	RO	EPTYPE	0x0	端口传输类型 h0: Control h1: ISO h2: Bulk h3: Int
[17]	RO	NAKSts	0x0	NAK状态 0: 软核根据FIFO状态发出non-NAK握手包 1: 软核发出NAK握手包 当置位后, 数据传输如下: OUT端口: 软核强制停止接收任何数据;
[16]	RO	DPID	0x0	Endpoint Data PID(Intr/Bulk有效) 端口即将接收/发送包的PID信息。在端口使能后, 应用必须为一个接收/发送包设置SetD0PID或SetD1PID信息。 0: DATA0, 1: DATA1
[15]	RW/SC	USBActEP	0x0	USB Active Endpoint 标示该端口在当前配置和接口描述符下是否处于激活状态。应用必须根据SetConfiguration和SetInterface指令, 确定是否配置该位。 软核在监测到USB reset后自动清除该位
[14:11]	-	Non-usage	0x0	不使用
[10:0]	RW	MPS	0x0	Maximum Packet Size 应该软件必须将最大包长度写置该段, 以字节为单位

#### 5.6.5.39 DOEPTSIZ2/4

偏移地址: 0xb50/0xb90

复位方式: H/S

Bits	Access	Name	Default	Description
[31]	-	reserved	0x0	保留
[30:29]	RO	RxDPID	0x0	Received Data PID(ISO有效) 标示最后一个包的数据PID号 h0: DATA0

				'h1: DATA2 'h2: DATA1 'h3: MDATA
[28:19]	RW	PktCnt	0x0	Packet Count 指定事务包数目, RxFIFO接收一个包, 该段自动减1
[18:0]	RW	XferSize	0x0	Transfer Size 端口本次事务总字节长度, 每从RxFIFO读取一个字节, 该段自动减1, 减至零时触发中断

#### 5.6.5.40 PCGCCTL

偏移地址: 0xe00

复位方式: H/S

Bits	Access	Name	Default	Description
[31:4]	-	reserved	0x0	保留
[3]	RW	RstPdownModule	0x0	Reset Power-Down Modules, 仅在partial power-down模式下有效。应用在电源关闭后置位, 在电源打开且PHY时钟起来后清除该位。
[2]	-	Non-usage	0x0	不使用
[1]	RW	GateHclk	0x0	Gate Hclk 在USB挂起时, 应用可置位该位, 关闭hclk
[0]	RW	StopPclk	0x0	Stop Pclk 当Dev处于挂起或disconnect态时, 应用软件可置位该位关闭phy_clk

## 5.7 GPIO

### 5.7.1 概述

HR\_C7000 支持 3 组 GPIO (GPIOA~GPIOC), 每组最多可以有 32 个 GPIO。每个 GPIO 口可单独设为输入输出模式, 在输入模式支持中断输入。部分 GPIO 和功能管脚复用。

### 5.7.2 功能描述

GPIO 模块支持以下功能:

- 支持每个 GPIO 管脚可以设置为输入或输出, 可以作为外部中断源输入;
- 当设置为输出模式时, 每个 GPIO 管脚可以独立置位或清零;
- 在中断模式下, 支持上升沿, 下降沿, 双边沿, 高电平, 低电平这 5 种触发中断方式;
- 支持去抖功能, 去抖采样时间可配。

### 5.7.3 工作方式

GPIO 管脚在复位之后处于输入状态。

### 通用输入输出

每个管脚可以配置为输入或者输出，具体步骤：

- 1、 参照 IO 复用寄存器描述，配置管脚的相应位，使能需要使用的 GPIO 管脚功能；
- 2、 配置寄存器 GPIO\_SWPORTX(A-D)\_DDR，选择 GPIO 是作为输入还是输出；
- 3、 当配置为输入管脚时，读取寄存器 GPIO\_EXT\_PORTX(A-D)可查看输入信号值；当配置为输出管脚时，可通过写寄存器 GPIO\_SWPORTX(A-D)\_DR 控制 GPIO 管脚输出电平。

注意：当 GPIO 作为输出时，应不是能 GPIO 的中断功能，否则当输出信号符合中断条件时会产生中断。

### 中断操作

只有端口 A 具有中断功能，端口 B、C、D 均不具有中断功能，只可作为普通输入输出管脚。GPIO 中断可通过 GPIO\_INTTYPE\_LEVEL 和 GPIO\_INT\_POLARITY 等寄存器进行控制。当有多个中断同时发生时，将会统一汇集成一个中断上报。

配置过程如下：

- 1、 配置寄存器 GPIO\_INTTYPE\_LEVEL，选择边沿触发或者电平触发；
- 2、 配置寄存器 GPIO\_INT\_POLARITY，选择下降沿/上升沿触发和高电平/低电平触发；
- 3、 配置寄存器 GPIO\_PORTA\_EOI 为 0xff，清中断；
- 4、 配置寄存器 GPIO\_INTEN 使能中断。

注意：GPIO 管脚信号保持稳定，以免产生假中断。

边沿触发中断：软件必须清除该中断以使能下一次中断。

电平触发中断：外部中断源应该保持该电平直到处理器识别到该中断。

## 5.7.4 寄存器概述

寄存器表 20 GPIOA 寄存器概述（基地址：0x1402\_0000）

偏移地址	名称	描述	页码
0x00	SWPORTA_DR	GPIO 数据寄存器	
0x04	SWPORTA_DDR	GPIO 方向寄存器	
0x30	INTEN	中断使能寄存器	
0x34	INTMASK	中断屏蔽寄存器	
0x38	INTTYPE_LEVEL	中断类型器	
0x3c	INT_POLARITY	中断极性寄存器	
0x40	INTSTATUS	中断状态寄存器	
0x44	RAW_INTSTATUS	原始中断状态寄存器	
0x48	DEBOUNCE	去抖使能	
0x4c	PORTA_EOI	中断清除寄存器	
0x50	EXT_PORTA	端口状态	
0x78	GPIO_INTR_BOTH_EDGE	双边沿触发中断	

寄存器表 21 GPIOB 寄存器概述（基地址：0x1410\_0000）

偏移地址	名称	描述	页码
0x00	SWPORTA_DR	GPIO 数据寄存器	
0x04	SWPORTA_DDR	GPIO 方向寄存器	

0x30	INTEN	中断使能寄存器	
0x34	INTMASK	中断屏蔽寄存器	
0x38	INTTYPE_LEVEL	中断类型器	
0x3c	INT_POLARITY	中断极性寄存器	
0x40	INTSTATUS	中断状态寄存器	
0x44	RAW_INTSTATUS	原始中断状态寄存器	
0x48	DEBOUNCE	去抖使能	
0x4c	PORTA_EOI	中断清除寄存器	
0x50	EXT_PORTA	端口状态	
0x78	GPIO_INTR_BOTH_EDGE	双边沿触发中断	

寄存器表 22 GPIOC 寄存器概述（基地址：0x1411\_0000）

偏移地址	名称	描述	页码
0x00	SWPORTA_DR	GPIO 数据寄存器	
0x04	SWPORTA_DDR	GPIO 方向寄存器	
0x30	INTEN	中断使能寄存器	
0x34	INTMASK	中断屏蔽寄存器	
0x38	INTTYPE_LEVEL	中断类型器	
0x3c	INT_POLARITY	中断极性寄存器	
0x40	INTSTATUS	中断状态寄存器	
0x44	RAW_INTSTATUS	原始中断状态寄存器	
0x48	DEBOUNCE	去抖使能	
0x4c	PORTA_EOI	中断清除寄存器	
0x50	EXT_PORTA	端口状态	
0x78	GPIO_INTR_BOTH_EDGE	双边沿触发中断	

## 5.7.5 寄存器描述

### 5.7.5.1 SWPORTA\_DR

偏移地址：0x00

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	PortA Data Register	0x0	端口A输出数据

### 5.7.5.2 SWPORTA\_DDR

偏移地址：0x04

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	PortA Data Direction Register	0x0	控制端口A数据方向

### 5.7.5.3 INTEN

偏移地址: 0x30

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Interrupt enable	0x0	中断使能

### 5.7.5.4 INTMASK

偏移地址: 0x34

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Interrupt mask	0x0	控制是否屏蔽中断

### 5.7.5.5 INTTYPE\_LEVEL

偏移地址: 0x38

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Interrupt level	0x0	控制端口A的中断类型

### 5.7.5.6 INT\_POLARITY

偏移地址: 0x3c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Interrupt polarity	0x0	控制端口A的中断的极性

### 5.7.5.7 INTSTATUS

偏移地址: 0x40

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	Interrupt status	0x0	端口A中断状态

### 5.7.5.8 RAW\_INTSTATUS

偏移地址: 0x44

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	Raw interrupt status	0x0	端口A的屏蔽前的中断状态

### 5.7.5.9 DEBOUNCE

偏移地址: 0x48

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	Debounce	0x0	去抖使能寄存器

### 5.7.5.10 PORTA\_EOI

偏移地址: 0x4c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	PortA_EOL	0x0	PORTA中断清除寄存器。0: 不清除中断; 1: 清除中断

### 5.7.5.11 EXT\_PORTA

偏移地址: 0x50

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RO	External PortA	0x0	当端口A配置为输入, 读此寄存器读取输入数据

### 5.7.5.12 INTR\_BOTH\_EDGE

偏移地址: 0x78

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	Gpio_intr_both_edge	0x0	1: 双边沿触发中断 0: 非双边沿触发中断。 说明: 若要配置双边沿中断, 先配置gpio_inttype_level为边沿触发, 然后再配置0x78寄存器。当不使用双边沿中断功能时, 0x78寄存器写0。

## 5.8 PWM

### 5.8.1 概述

PWM 模块产生周期性的脉冲波形, 其中周期频率及占空比均可通过寄存器配置。HR\_C7000 提供 3 个独立的 PWM 模块。

### 5.8.2 功能描述

PWM 模块支持以下功能:

- 周期频率可设, 最大支持 54M;
- 占空比可设。

PWM 输出时序图如图所示, 其中 period 和高电平 phase 长度均可配置:



持续的时钟拍数,通过配置 PWM\_FP 来实现。一个周期持续的时钟拍数可以通过配置 PWM\_PER 来实现。

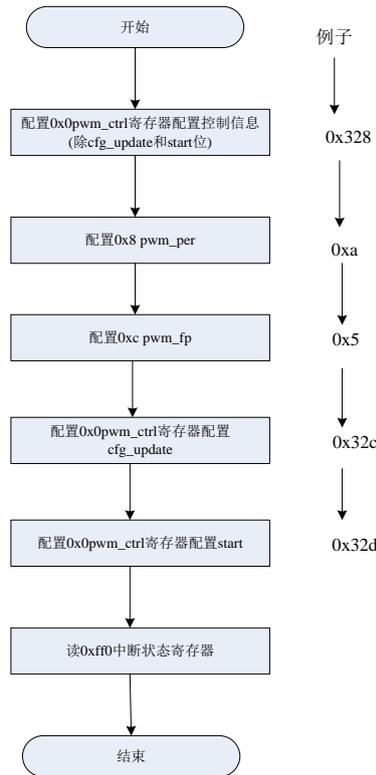


图 49 CONTINUOUS 模式软件操作示意图

### 5.8.3.2 One Shot 模式

配置 PWM\_CFG 的 bit[5:4]为 2'b01, 表示为 one shot 模式。然后配置一个周期第一个相位持续的时钟拍数,通过配置 PWM\_FP 来实现。一个周期持续的时钟拍数可以通过配置 PWM\_PER 来实现。重复的周期数可以通过配置 PWM\_RPT 来实现。

ONE\_SHOT 模式配置方式如下:

- 1、配置寄存器 PWM\_CTRL 的相关信息,除 bit0[start]和 bit2[cfg\_update]外;
- 2、配置寄存器 PWM\_RPT、PWM\_PER 和 PWM\_FP;
- 3、配置寄存器 PWM\_CTRL 的 bit2,写 1;
- 4、配置寄存器 PWM\_CTRL 的 bit0,写 1;
- 5、根据需求查看 PWM\_STATUS 和 PWM\_INTR\_STA 状态。

### 5.8.4 寄存器概述

寄存器表 23 PWM 寄存器概述 (基地址: 0x140C\_0000)

偏移地址	名称	描述	页码
0x0000	PWM0_CFG	PWM0 配置寄存器	
0x0004	PWM0_RPT	PWM0 ONE_SHOT 传输重复的周期数	
0x0008	PWM0_PER	PWM0 周期波形持续系统时钟的拍数	
0x000C	PWM0_FP	PWM0 一个周期第一个相位持续的拍数	

0x0010	PWM0_STATUS	PWM0 状态寄存器	
0x0020	PWM1_CFG	PWM1 配置寄存器	
0x0024	PWM1_RPT	PWM1 ONE_SHOT 传输重复的周期数	
0x0028	PWM1_PER	PWM1 周期波形持续系统时钟的拍数	
0x002C	PWM1_FP	PWM1 一个周期第一个相位持续的拍数	
0x0030	PWM1_STATUS	PWM1 状态寄存器	
0x0040	PWM2_CFG	PWM2 配置寄存器	
0x0044	PWM2_RPT	PWM2 ONE_SHOT 传输重复的周期数	
0x0048	PWM2_PER	PWM2 周期波形持续系统时钟的拍数	
0x004C	PWM2_FP	PWM2 一个周期第一个相位持续的拍数	
0x0050	PWM2_STATUS	PWM2 状态寄存器	
0xff0	PWM_INTR_STA	所有 PWM 中断状态	

### 5.8.5 寄存器描述

#### 5.8.5.1 PWM0\_CFG

偏移地址: 0x0

复位方式: H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved	-	保留
[9]	RW	inactout	0x0	pwm 不被使能时输出的电平。
[8]	RW	fpout	0x0	pwm 一个周期内第一个相位的输出电平
[7:6]	RW	evtrig	0x0	在 ONE_SHOT 模式下的事件触发模式 1:上升沿触发 2:下降沿触发 0,3:普通模式,即不根据外部信号进行 pwm 信号的输出
[5:4]	RW	mode	0x0	PWM 的模式, 1:ONE_SHOT 模式 2:CONTINUOUS 模式 0,3:保留, 不进行 PWM 波形输出
[3]	RW	inten	0x0	中断使能
[2]	RW	cfg_update	0x0	在 CONTINUOUS 模式时, 会在中间时刻更新 PWM_PER 和 PWM_FP 两个寄存器。当更改寄存器值后对 cfg_update 进行从 0 到 1 的配置(表示两个寄存器值进行同时更新)
[1]	RW	soft_rst	0x0	软件复位使能, 用于在任意时刻关闭 pwm 的波形输出。
[0]	RW	start	0x0	pwm 开始编码使能, 上升沿有效(即从 0 变到 1)。

#### 5.8.5.2 PWM0\_RPT

偏移地址: 0x4

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved	-	保留
[15:0]	RW	inactout	0x0	一次 ONE_SHOT 传输重复的周期数。

### 5.8.5.3 PWM0\_PER

偏移地址: 0x8

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	pwm0_rpt	0x0	一个周期波形持续系统时钟的拍数。

### 5.8.5.4 PWM0\_FP

偏移地址: 0xC

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	pwm0_fp	0x0	一个周期波形第一个相位持续的拍数。如果 pwm_fp 大于 pwm_per, 则所有有效信号都为 fpout, 如果 pwm_fp 为 0, 则所有有效信号都为!fpout

### 5.8.5.5 PWM0\_STATUS

偏移地址: 0x10

复位方式: H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved	-	保留
[9]	RW	pwm_out	0x0	pwm 波形的当前值
[8]	RW	pwm_busy	0x0	pwm状态机的状态 0: 空闲状态 1: 工作状态
[7:0]	RW	pwm_cycle	0x0	在CONTINUOUS模式下, 同一组 pwm_per&pwm_fp参数已经输出的周期个数

### 5.8.5.6 PWM1\_CFG

偏移地址: 0x20

复位方式: H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved	-	保留
[9]	RW	inactout	0x0	pwm 不被使能时输出的电平。
[8]	RW	fpout	0x0	pwm 一个周期内第一个相位的输出电平
[7:6]	RW	evtrig	0x0	在 ONE_SHOT 模式下的事件触发模式 1:上升沿触发 2:下降沿触发

				0,3:普通模式,即不根据外部信号进行 pwm 信号的输出
[5:4]	RW	mode	0x0	PWM 的模式, 1:ONE_SHOT 模式 2:CONTINUOUS 模式 0,3:保留, 不进行 PWM 波形输出
[3]	RW	inten	0x0	中断使能
[2]	RW	cfg_update	0x0	在 CONTINUOUS 模式时, 会在中间时刻更新 PWM_PER 和 PWM_FP 两个寄存器。当更改寄存器值后对 cfg_update 进行从 0 到 1 的配置(表示两个寄存器值进行同时更新)
[1]	RW	soft_rst	0x0	软件复位使能, 用于在任意时刻关闭 pwm 的波形输出。
[0]	RW	start	0x0	pwm 开始编码使能, 上升沿有效(即从 0 变到 1)。

#### 5.8.5.7 PWM1\_RPT

偏移地址: 0x24

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved	-	保留
[15:0]	RW	inactout	0x0	一次 ONE_SHOT 传输重复的周期数。

#### 5.8.5.8 PWM1\_PER

偏移地址: 0x28

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	pwm0_rpt	0x0	一个周期波形持续系统时钟的拍数。

#### 5.8.5.9 PWM1\_FP

偏移地址: 0x2C

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	pwm0_fp	0x0	一个周期波形第一个相位持续的拍数。如果 pwm_fp 大于 pwm_per, 则所有有效信号都为 fpout, 如果 pwm_fp 为 0, 则所有有效信号都为!fpout

#### 5.8.5.10 PWM1\_STATUS

偏移地址: 0x30

复位方式: H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved	-	保留
[9]	RW	pwm_out	0x0	pwm 波形的当前值
[8]	RW	pwm_busy	0x0	pwm状态机的状态 0: 空闲状态 1: 工作状态
[7:0]	RW	pwm_cycle	0x0	在CONTINUOUS模式下, 同一组 pwm_per&pwm_fp参数已经输出的周期个数

#### 5.8.5.11 PWM2\_CFG

偏移地址: 0x40

复位方式: H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved	-	保留
[9]	RW	inactout	0x0	pwm 不被使能时输出的电平。
[8]	RW	fpout	0x0	pwm 一个周期内第一个相位的输出电平
[7:6]	RW	evtrig	0x0	在 ONE_SHOT 模式下的事件触发模式 1:上升沿触发 2:下降沿触发 0,3:普通模式,即不根据外部信号进行 pwm 信号的输出
[5:4]	RW	mode	0x0	PWM 的模式, 1:ONE_SHOT 模式 2:CONTINUOUS 模式 0,3:保留, 不进行 PWM 波形输出
[3]	RW	inten	0x0	中断使能
[2]	RW	cfg_update	0x0	在 CONTINUOUS 模式时, 会在中间时刻更新 PWM_PER 和 PWM_FP 两个寄存器。当更改寄存器 值后对 cfg_update 进行从 0 到 1 的配置(表示两个寄 存器值进行同时更新)
[1]	RW	soft_rst	0x0	软件复位使能, 用于在任意时刻关闭 pwm 的波形输 出。
[0]	RW	start	0x0	pwm 开始编码使能, 上升沿有效(即从 0 变到 1)。

#### 5.8.5.12 PWM2\_RPT

偏移地址: 0x44

复位方式: H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved	-	保留
[15:0]	RW	inactout	0x0	一次 ONE_SHOT 传输重复的周期数。

### 5.8.5.13 PWM2\_PER

偏移地址：0x48

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	pwm2_rpt	0x0	一个周期波形持续系统时钟的拍数。

### 5.8.5.14 PWM2\_FP

偏移地址：0x4C

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	pwm2_fp	0x0	一个周期波形第一个相位持续的拍数。如果 pwm_fp 大于 pwm_per，则所有有效信号都为 fpout，如果 pwm_fp 为 0，则所有有效信号都为!fpout

### 5.8.5.15 PWM1\_STATUS

偏移地址：0x30

复位方式：H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved	-	保留
[9]	RW	pwm_out	0x0	pwm 波形的当前值
[8]	RW	pwm_busy	0x0	pwm状态机的状态 0: 空闲状态 1: 工作状态
[7:0]	RW	pwm_cycle	0x0	在CONTINUOUS模式下，同一组 pwm_per&pwm_fp参数已经输出的周期个数

### 5.8.5.16 PWM\_INTR\_STA

偏移地址：0xff0

复位方式：H/S

Bits	Access	Name	Default	Description
[31:3]	-	reserved	-	保留
[2]	RO	pwm2_intr	0x0	pwm2输出的中断状态
[1]	RO	pwm1_intr	0x0	pwm1输出的中断状态
[0]	RO	pwm0_intr	0x0	pwm0输出的中断状态

## 5.9 ADC 控制器

### 5.9.1 概述

HR\_C7000 内置 ADC。ADC 控制器用于根据上层软件配置，控制 ADC 正常采样，并将采样数据上传。

### 5.9.2 功能描述

- ADC 控制器具有如下功能：
- 数据转换速率 200KSPS
- 采样精度 10 bit
- 8 个 ADC 通道
- 支持单通道扫描功能
- 支持连续扫描功能
- 支持连续扫描时低功耗模式，ADC 自动开启或退出低功耗状态
- 采用可屏蔽中断方式上报 ADC 采样数据
- 支持采样初始值及变化阈值配置，自动过滤采样信号都抖动
- 支持 ADC 异常状态中断，并设计 ADC 强制软复位功能
- 支持可调 ADC 上电/掉电保护时间间隔
- 支持将 ADC 采样数据直接串行输出到片外

### 5.9.3 工作方式

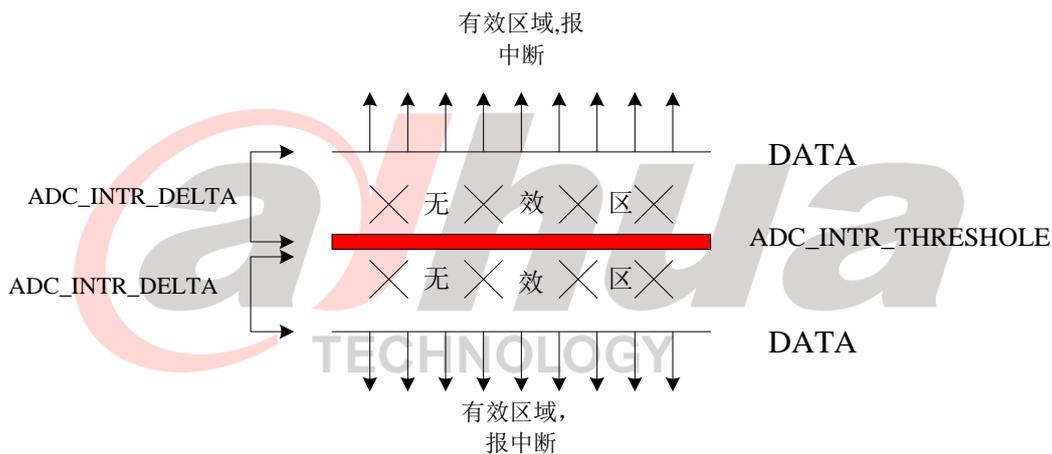


图 50 ADC\_INTR\_THRESHOLE 与 ADC\_INTR\_DELTA 关系示意图

上图中，从 ADC 采样到的数据 DATA，与 ADC\_INTR\_THRESHOLE 做差值，差值的绝对值大于 ADC\_INTR\_DELTA 的数据，上报中断。

#### 5.9.3.1 单次扫描处理流程

在单次扫描模式下，配置扫描模式(ADC\_CTRL[2]=1'b0)，是否支持低功耗(ADC\_CTRL[0])、有效通道号 ADC\_CH\_VLD 等相关控制信息，ADC\_CTRL 启动(ADC\_START)，启动一次扫描，等待 ADC 转换结果，ADC\_CTRL 获取 ADC 转换值，上报中断。单次扫描不滤毛刺。

具体操作流程如下：

- 1) 配置控制器 ADC\_CTRL(0x0) = 0x8，软复位 ADC 模块和控制器；
- 2) 配置控制器 ADC\_CTRL\_STOP (0x20) 寄存器的 bit1，软复位控制器。
- 3) 读取 ADC\_CTRL\_STATE(0x18)和 ADC\_INTR\_STA(0x0c)，确认控制器当前状态，若

ADC\_CTRL\_STATE 寄存器 ADC\_CTRL\_BUSY 位和 SAMP\_FSM\_STATE 位不为零，则需要重复步骤 3)，继续复位控制器直至控制器进入 IDLE 态；

- 4) 配置 ADC\_CTRL(0x0) = 0x0，使能单次扫描模式；
- 5) 配置 ADC\_INTR\_THRESHOLD(0x1c)，设置采样数据触发中断水平；
- 6) 配置 ADC\_DATA\_DELTA(0x4)，设置采样信号与触发水平值误差范围；
- 7) 配置 ADC\_PD\_SEOC\_TIME(0x28)，设置 PD 上电和掉电保护时间段；
- 8) 配置 ADC\_P2S\_EN(0x2c)，开启或关闭 ADC 数据串行输出；
- 9) 配置 ADC\_INTR(0x08)[16] = 1'b1，及 ADC\_INTR[8:0]使能相应中断；
- 10) 配置 ADC\_CH\_VLD(0x24)，使能采样通道。若此时配置多通道有效，控制器仅输出最低通道号数据，例如配置 ADC\_CH\_VLD=0x6，则仅输出 B 通道数据；
- 11) 设置 ADC\_START(0x14)=0x1，启动 ADC 单次单通道采样；
- 12) 等待 ADC\_INTR\_STA(0x0c)相应通道号中断置位后，从 0x30 或 0x34 或 0x38 寄存器中读取相应采样数据，并清除中断寄存器(0x0c)。

注意：

- a) 步骤 1)-2)和 5)-9)仅需要在第一次操作时配置，之后操作无需再重复；每次启动新一次扫描，其他步骤都需要重复操作。
- b) 务必在开启采样前读清除 ADC\_INTR\_STA(0x0c)以及配置 ADC\_STOP=0x2 复位控制器，
- c) 不得在 ADC\_START 后至 ADC\_CTRL\_BUSY 清零前，写入软复位(也即 ADC\_CTRL(0x00) bit4 和 ADC\_CTRL\_STOP(0x20)) 之外的任何寄存器。



5.9.3.2 单次扫描软件操作流程:

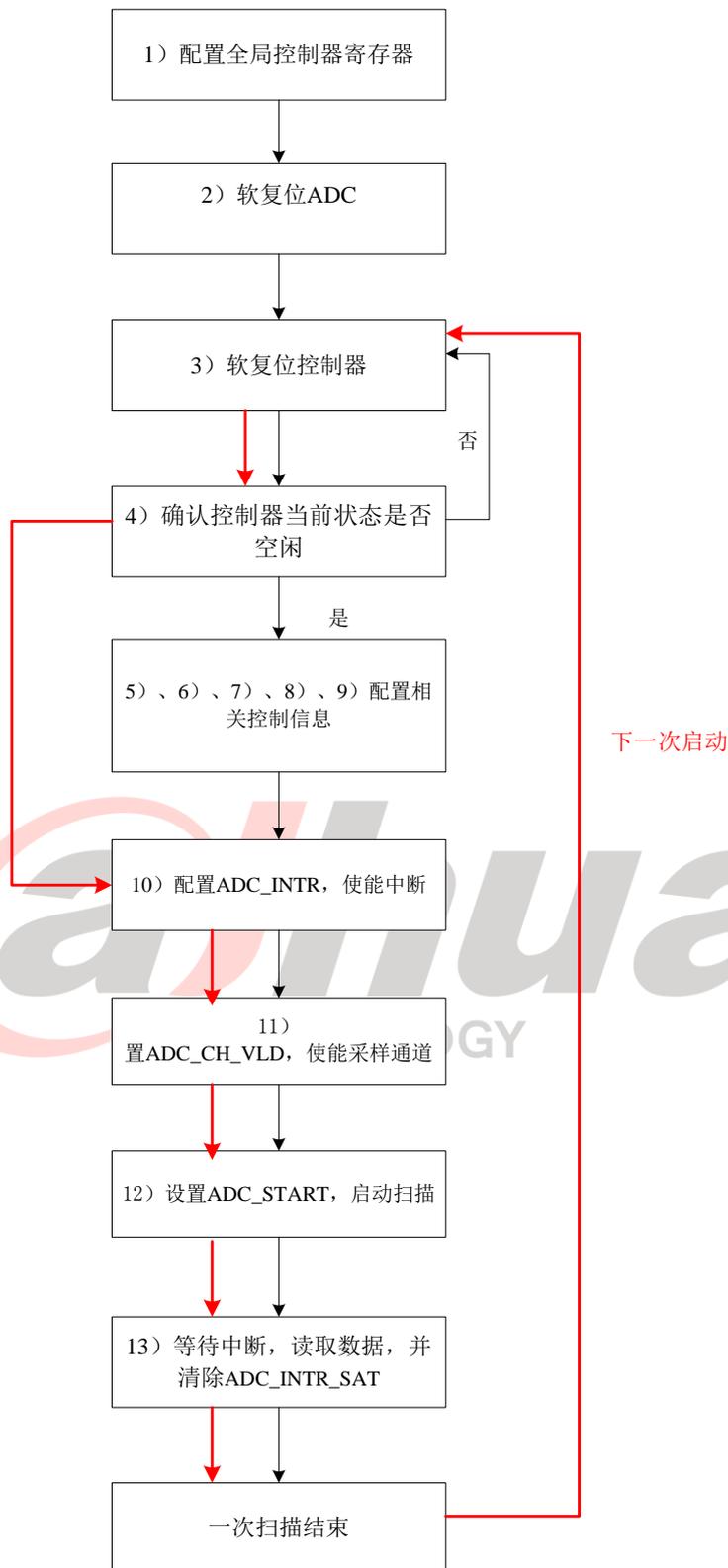


图 51 ADC\_CTRL 单次扫描软件操作流程

上图中，黑色箭头流程为复位以后，第一次扫描软件操作流程。红色箭头部分为再次启动软件操作流程。

### 5.9.3.3 连续扫描处理流程

在连续扫描模式下，配置扫描模式（ADC\_CTRL[2]=1'b1），是否滤毛刺（ADC\_CTRL[1]）、是否支持低功耗模式（ADC\_CTRL[0]）连续扫描时间间隔 ADC\_SCAN\_TIME、有效通道号 ADC\_CH\_VLD，ADC\_CTRL 启动（ADC\_START），启动扫描，在一个时间间隔 Tscan 内完成一个有效通道的扫描，在下一个扫描时刻到来时，启动对下一个有效通道的扫描。

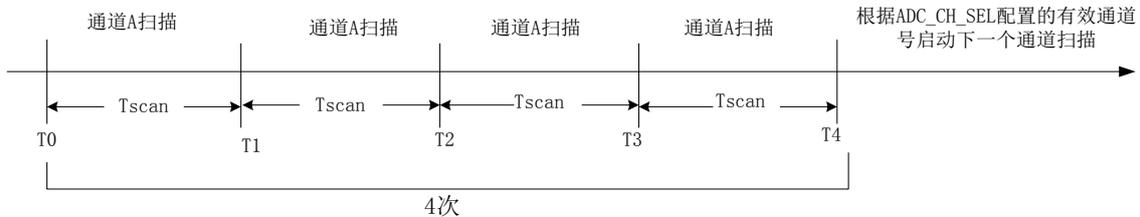


图 52 ADC\_CTRL 连续扫描通道轮询示意图

具体操作流程如下：

- 1) 配置控制器 ADC\_CTRL(0x0) = 0x8，复位 ADC 模块和控制器；
- 2) 配置控制器 ADC\_CTRL\_STOP (0x20) 寄存器的 bit1，软复位控制器
- 3) 读取 ADC\_CTRL\_STATE(0x18) 和 ADC\_INTR\_STA(0x0c)，确认控制器当前状态，若 ADC\_CTRL\_STATE 寄存器 ADC\_CTRL\_BUSY 位和 SAMP\_FSM\_STATE 位不为零，则需要重复步骤 3)，继续复位控制器直至控制器进入 IDLE 态；
- 4) 配置 ADC\_CTRL(0x0)，使能连续扫描模式，其中 ADC\_CTRL= 0x4 表示使能连续扫描模式；ADC\_CTRL= 0x5 表示使能低功耗连续扫描模式；ADC\_CTRL= 0x6 表示使能滤毛刺连续扫描模式；ADC\_CTRL= 0x7 表示使能低功耗滤毛刺连续扫描模式；
- 5) 配置 ADC\_INTR\_THRESHOLD(0x1c)，设置采样数据触发中断水平；
- 6) 配置 ADC\_DATA\_DELTA(0x4)，设置采样信号与触发水平值误差范围；
- 7) 配置 ADC\_PD\_SEOC\_TIME(0x28)，设置 PD 上电和掉电保护时间段；（步骤 8）必须先于步骤 9）配置）
- 8) 配置 ADC\_SCAN\_TIME(0x18)，设置扫描时间，若扫描时间值小于 ADC\_PD\_SEOC\_TIME (0x28) EOC\_PD\_TIME+12，则实际写入值为 EOC\_PD\_TIME+12，且 ADC\_CTRL\_STATE (0x18) 寄存器 SCAN\_TIME\_ERR 置位；
- 9) 配置 ADC\_P2S\_EN(0x2c)，开启或关闭 ADC 数据串行输出；
- 10) 配置 ADC\_INTR(0x08)[16] = 1'b1，及 ADC\_INTR[8:0]使能相应中断；
- 11) 配置 ADC\_CH\_VLD(0x24)，使能采样通道。控制器按照从低到高的顺序依次采样各通道，例如配置 ADC\_CH\_VLD=0x6，控制器依次输出 B、C 通道数据；
- 12) 设置 ADC\_START(0x14)=0x1，启动 ADC 连续采样；
- 13) 等待 ADC\_INTR\_STA(0x0c) 相应通道号中断置位后，从 0x30 或 0x34 或 0x38 寄存器中读取相应采样数据，并清除中断寄存器(0x0c)。

注意：

- a) 若相关配置参数不变，步骤 1)-2) 和 5)-10) 仅需要在第一次操作时配置，之后操作无需再重复；每次启动新一次扫描，其他步骤都需要重复操作
- b) 务必在开启采样前读清除 ADC\_INTR\_STA(0x0c) 以及配置 ADC\_STOP=0x3 复位控制器；
- c) 连续采样结束时务必配置 ADC\_CTRL\_STOP=0x1，终止连续采样模式；

- d) 不得在 ADC\_START 后至 ADC\_CTRL\_BUSY 清零前, 写入软复位(也即 ADC\_CTRL(0x00) bit4 和 ADC\_CTRL\_STOP(0x20)) 之外的任何寄存器。

#### 5.9.3.4 连续扫描软件操作流程

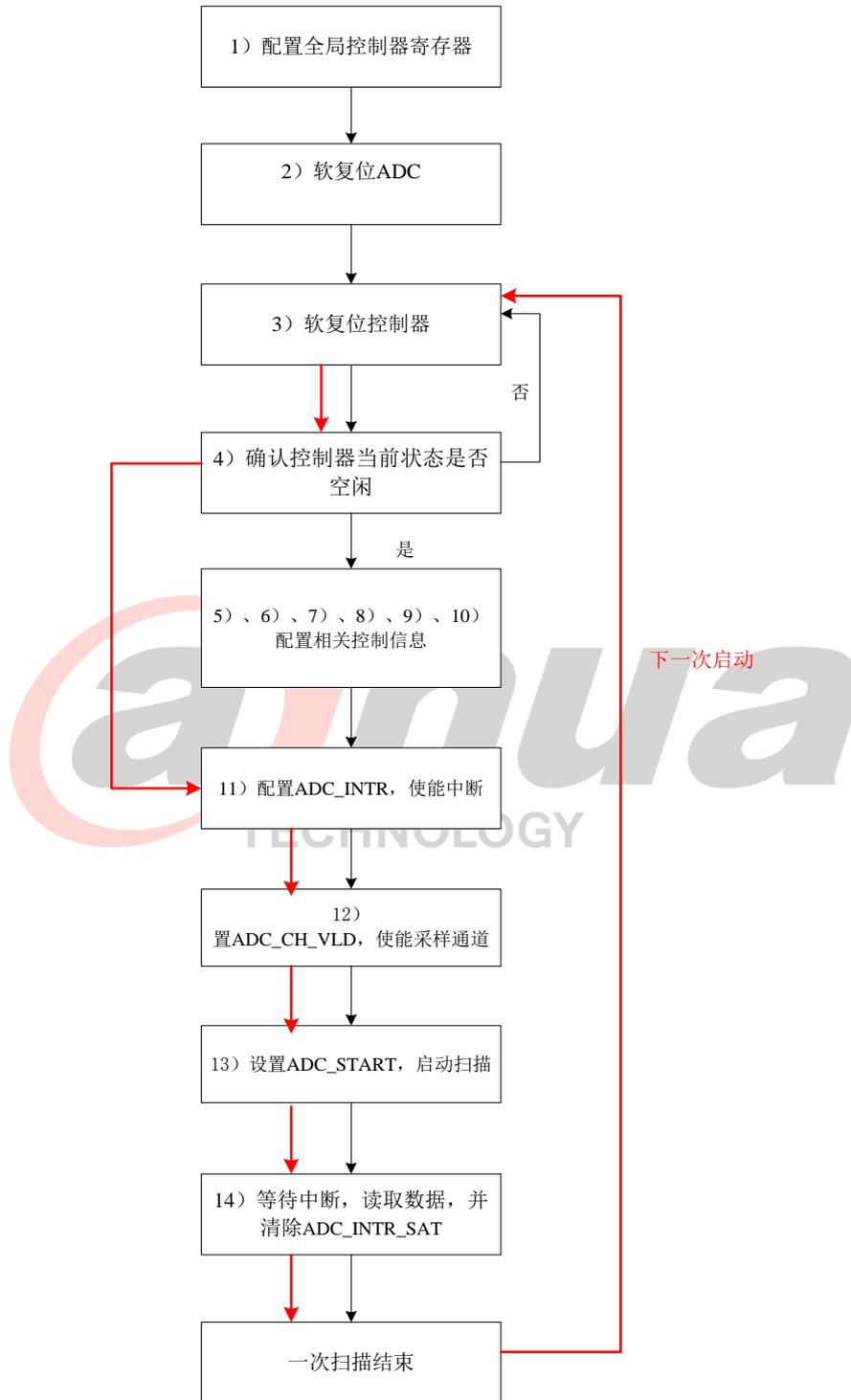


图 53 ADC\_CTRL 连续扫描软件操作流程

上图中, ADC\_CTRL 连续扫描软件操作流程图中, 下一次启动时, 配置相关控制步骤, 若

是否滤毛刺、是否支持低功耗等控制信息不变，则只需要重新配置 ADC\_CH\_VLD 和 ADC\_START 寄存器即可。如果其他配置需要改变，也可重新配置其他寄存器。

黑色箭头流程为复位以后，第一次扫描软件操作流程。红色箭头部分为再次启动软件操作流程。

### 5.9.3.5 滤毛刺流程

连续扫描模式下，滤毛刺过程为，每个通道都以 4 次数据为一个单位，每采样 4 次数据取平均值输出。对滤毛刺后得到的有效扫描值，上报中断并置扫描完成标志位。置位 ADC\_CTRL(0x1) GLITCH 位使能该功能。

### 5.9.3.6 低功耗流程

连续扫描模式下，当使能低功耗模式时，每次 ADC 转换完成后进入低功耗模式，下次扫描时候到来时，现将 ADC 从低功耗模式唤醒，再次启动 ADC 转换。置位 ADC\_CTRL(0x1) POWRR\_DOWN 位使能该功能。

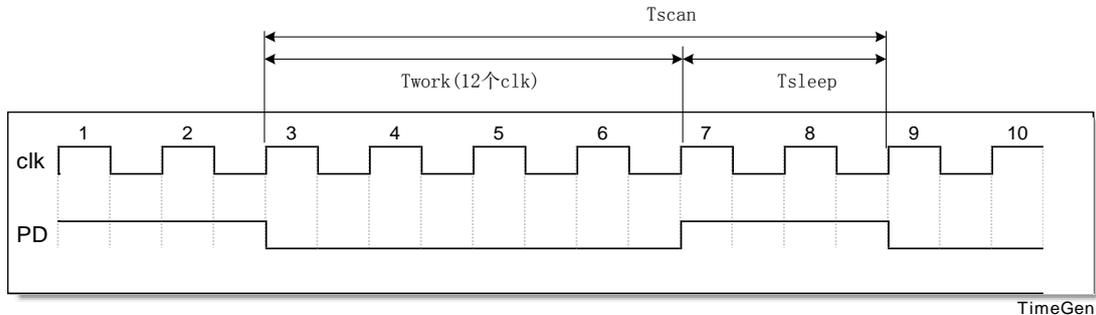


图 54 ADC\_CTRL 低功耗模式时序简图

ADC\_CTRL 的扫描时间间隔是  $T_{scan}$ ，ADC 的转换时间为  $T_{work}$ ， $T_{work}$  以后，ADC 进入低功耗模式，低功耗的时间为  $T_{sleep}$ ，等下次  $T_{scan}$  时间到，再次进入  $T_{work}$ 。

## 5.9.4 寄存器概述

寄存器表 24 ADC 寄存器概述（基地址：0x140D\_0000）

偏移地址	名称	描述	页码
0x0	ADC_CTRL	ADC 控制寄存器	
0x4	ADC_INTR_DELTA	ADC 中断误差范围	
0x8	ADC_INTR	ADC 中断控制寄存器	
0xc	ADC_INTR_STA	ADC 中断状态寄存器	
0x10	ADC_SCAN_TIME	ADC 扫描时间间隔设置	
0x14	ADC_START	ADC 启动	
0x18	ADC_CTRL_STATE	ADC 控制和状态寄存器	
0x1c	ADC_INTR_THRESHOLD	ADC 中断初始值	
0x20	ADC_CTRL_STOP	ADC 配置寄存器	
0x24	ADC_CH_VLD	ADC 通道配置寄存器	
0x28	ADC_PD_SEOC_TIME	ADC 掉电时间寄存器	

0x2c	ADC_P2S_EN	ADC 串行数据输出寄存器	
0x30	ADC_DATA_AB	ADC 通道 AB 扫描值	
0x34	ADC_DATA_CD	ADC 通道 CD 扫描值	
0x38	ADC_DATA_EF	ADC 通道 EF 扫描值	
0x3c	ADC_DATA_GH	ADC 通道 GH 扫描值	
0x40	ADC_CTRL_VERID	ADC 版本寄存器	
0x44	ADC_DIR_DAT_OP	ADC EOC 后直接输出 ADC 数据	

## 5.9.5 寄存器描述

### 5.9.5.1 ADC\_CTRL

偏移地址：0x00

复位方式：H/S

Bits	Access	Name	Default	Description
[31:4]	-	reserved		保留
[3]	RW	PD_FORCE_GIGH	0x0	写1软复位ADC和控制器，正常工作时必须写0清0
[2]	RW	MODE_SEL	0x0	扫描模式选择。0：单次扫描；1：连续扫描
[1]	RW	GLITCH	0x0	是否滤毛刺。0：不滤毛刺；1：滤毛刺。（单次扫描不需要滤毛刺）
[0]	RW	POWER_DOWN	0x0	是否支持低功耗模式。0：不支持；1：支持

### 5.9.5.2 ADC\_INTR\_DELTA

偏移地址：0x04

复位方式：H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved		保留
[9:0]	RW	ADC_DATA_DELT A	0x0	设置中断误差范围，采样数据与 ADC_INTR_THRESHOLE(0x24)寄存器配置值差值大于(不含等于)时，控制器才能触发数据中断、输出数据。 详细关系请见图50。

### 5.9.5.3 ADC\_INTR

偏移地址：0x08

复位方式：H/S

Bits	Access	Name	Default	Description
[31:17]	-	reserved	-	保留
[16]	RW	INTR_EN	0x0	总中断使能。1：enable；0：disable
[15:9]	-	reserved	-	保留

[8]	RW	INTR_ADC_ERR	0x0	ADC异常中断使能
[7]	RW	INTR_EN_H	0x0	CHH扫描有效值中断使能。1: enable; 0: disable
[6]	RW	INTR_EN_G	0x0	CHG扫描有效值中断使能。1: enable; 0: disable
[5]	RW	INTR_EN_F	0x0	CHF扫描有效值中断使能。1: enable; 0: disable
[4]	RW	INTR_EN_E	0x0	CHE扫描有效值中断使能。1: enable; 0: disable
[3]	RW	INTR_EN_D	0x0	CHD扫描有效值中断使能。1: enable; 0: disable
[2]	RW	INTR_EN_C	0x0	CHC扫描有效值中断使能。1: enable; 0: disable
[1]	RW	INTR_EN_B	0x0	CHB扫描有效值中断使能。1: enable; 0: disable
[0]	RW	INTR_EN_A	0x0	CHA扫描有效值中断使能。1: enable; 0: disable

#### 5.9.5.4 ADC\_INTR\_STA

偏移地址: 0x0c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:25]	-	reserved		保留
[24]	RO	ADC_ERR_RAW	0x0	未屏蔽的ADC异常中断, 软件读清0 EOC信号未在SOC信号后第11cycles出现时, 触发该中断
[23]	RO	CHANNEL_H_INTR_RAW	0x0	未屏蔽的通道H中断标志, 软件读清0. 0: 无中断, 1: 有中断
[22]	RO	CHANNEL_G_INTR_RAW	0x0	未屏蔽的通道G中断标志, 软件读清0. 0: 无中断, 1: 有中断
[21]	RO	CHANNEL_F_INTR_RAW	0x0	未屏蔽的通道F中断标志, 软件读清0. 0: 无中断, 1: 有中断
[20]	RO	CHANNEL_E_INTR_RAW	0x0	未屏蔽的通道E中断标志, 软件读清0. 0: 无中断, 1: 有中断
[19]	RO	CHANNEL_D_INTR_RAW	0x0	未屏蔽的通道D中断标志, 软件读清0. 0: 无中断, 1: 有中断
[18]	RO	CHANNEL_C_INTR_RAW	0x0	未屏蔽的通道C中断标志, 软件读清0. 0: 无中断, 1: 有中断
[17]	RO	CHANNEL_B_INTR_RAW	0x0	未屏蔽的通道B中断标志, 软件读清0. 0: 无中断, 1: 有中断
[16]	RO	CHANNEL_A_INTR_RAW	0x0	未屏蔽的通道A中断标志, 软件读清0. 0: 无中断, 1: 有中断
[15:9]	-	reserved		保留
[8]	RO	ADC_ERR_INTR	0x0	ADC异常中断, 软件读清0 EOC信号未在SOC信号后第11cycle出现时, 触发该中断
[7]	RO	CHANNEL_H_INTR	0x0	通道H中断标志, 软件读清0。 0: 无中断, 1: 有中断
[6]	RO	CHANNEL_G_INTR	0x0	通道G中断标志, 软件读清0。 0: 无中断, 1: 有中断
[5]	RO	CHANNEL_F_INTR	0x0	通道F中断标志, 软件读清0。

				0: 无中断, 1: 有中断
[4]	RO	CHANNEL_E_INTR	0x0	通道E中断标志, 软件读清0。 0: 无中断, 1: 有中断
[3]	RO	CHANNEL_D_INTR	0x0	通道D中断标志, 软件读清0。 0: 无中断, 1: 有中断
[2]	RO	CHANNEL_C_INTRN	0x0	通道C中断标志, 软件读清0。 0: 无中断, 1: 有中断
[1]	RO	CHANNEL_B_INTR	0x0	通道B中断标志, 软件读清0。 0: 无中断, 1: 有中断
[0]	RO	CHANNEL_A_INTR	0x0	通道A中断标志, 软件读清0。 0: 无中断, 1: 有中断

#### 5.9.5.5 ADC\_SCAN\_TIME

偏移地址: 0x10

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	RW	SCAN_TIME	0x2c	扫描时间间隔设置, 以clk个数为单位。配置值最小值为 (adc_eoc_pd_time+12, 十进制数)。如果小于最小值, 则按最小值算, 同时ADC_CTRL_STATE[3]置位, 同时0x20寄存器SCAN_TIME_ERR置位。注意: 配此寄存器之前, 必须先配0x34寄存器。

#### 5.9.5.6 ADC\_START

偏移地址: 0x14

复位方式: H/S

Bits	Access	Name	Default	Description
[31:1]	-	reserved		保留
[0]	WO	ADC_START	0x0	写1, 启动adc_ctrl模块。单次扫描模式时, EOC信号的下一拍会将此位清0; 连续扫描时, adc_ctrl_stop寄存器的写使能信号会将此位清0。

#### 5.9.5.7 ADC\_CTRL\_STATE

偏移地址: 0x18

复位方式: H/S

Bits	Access	Name	Default	Description
[31:9]	-	reserved		保留
[8]	R/W	EOC_WIDTH_ERR	0x0	EOC信号宽度大于一个clk时置位。任意值清0。
[7]	R/W	SOC_EOC_TIME_OUT	0x0	SOC与EOC间隔不等于11个clk时置位。写任意值清0。

[6]	R/W	SCAN_TIME_ERROR	0x0	ADC_SCAN_TIME寄存器配置值小于最小值。写任意值清0。
[5:1]	RO	SAMP_FSM_STATE	0x0	ADC_CTRL采样数据状态机状态 9: 过渡状态, G_SAMPLE_S 8: 采样通道H数据, G_SAMPLE_H。 7: 采样通道G数据, G_SAMPLE_G。 6: 采样通道F数据, G_SAMPLE_F。 5: 采样通道E数据, G_SAMPLE_E。 4: 采样通道D数据, G_SAMPLE_D。 3: 采样通道C状态, G_SAMPLE_C。 2: 采样通道B状态, G_SAMPLE_B。 1: 采样通道A状态, G_SAMPLE_A。 0: 空闲状态, G_IDLE
[0]	RO	ADC_CTRL_BUSY	0x0	ADC_CTRL模块状态指示。 0: idle 1: busy

#### 5.9.5.8 ADC\_INTR\_THRESHOLE

偏移地址: 0x1c

复位方式: H/S

Bits	Access	Name	Default	Description
[31:0]	-	reserved		保留
[9: 0]	RW	ADC_INTR_THRES HOLD	0x0	ADC 中断初始值, 当转换结果与此初始值相减的差值结果大于 adc_intr_delta 的误差阈值时, 会触发上报中断。 详细关系见图50

#### 5.9.5.9 ADC\_CTRL\_STOP

偏移地址: 0x20

复位方式: H/S

Bits	Access	Name	Default	Description
[31:2]	-	reserved		保留
[1]	WO	ADC_CTRL_RESET	0x0	写1, 软复位控制器, 自清零
[0]	WO	ADC_CTRL_STOP	0x0	只在连续扫描模式下生效, 连续扫描模式下, 扫描完最后一个有效通道后停止自动扫描, 自清0。(连续扫描模式下, 向此寄存器写1可以停止adc_ctrl的自动扫描功能, 需要重新写ADC_START和ADC_CH_VLD寄存器才能重新启动自动扫描)

### 5.9.5.10 ADC\_CH\_VLD

偏移地址：0x24

复位方式：H/S

Bits	Access	Name	Default	Description
[31:8]	-	reserved		保留
[7]	RW	ADC_CHH_VLD	0x0	连续扫描模式下，通道H是否有效。0：无效；1：有效。
[6]	RW	ADC_CHG_VLD	0x0	连续扫描模式下，通道G是否有效。0：无效；1：有效。
[5]	RW	ADC_CHF_VLD	0x0	连续扫描模式下，通道F是否有效。0：无效；1：有效。
[4]	RW	ADC_CHE_VLD	0x0	连续扫描模式下，通道E是否有效。0：无效；1：有效。
[3]	RW	ADC_CHD_VLD	0x0	连续扫描模式下，通道D是否有效。0：无效；1：有效。
[2]	RW	ADC_CHC_VLD	0x0	连续扫描模式下，通道C是否有效。0：无效；1：有效。
[1]	RW	ADC_CHB_VLD	0x0	连续扫描模式下，通道B是否有效。0：无效；1：有效。
[0]	RW	ADC_CHA_VLD	0x0	连续扫描模式下，通道A是否有效。0：无效；1：有效。

备注：单次扫描模式，ADC\_CH\_VLD 仅最低位通道有效；多次扫描时，从低位通道到高位通道依次扫描。

### 5.9.5.11 ADC\_PD\_SEOC\_TIME

偏移地址：0x28

复位方式：H/S

Bits	Access	Name	Default	Description
[31:16]	-	reserved		保留
[15:8]	RW	EOC_PD_TIME	0xa	EOC下降沿与PD上升沿的时间间隔。以ADC_clk为单位。配置值小于default值时，取default值。
[7:0]	RW	PD_SOC_TIME	0x20	SOC上升沿与PD下降沿的时间间隔。以ADC_clk为单位。配置值小于default值时，取default值。

### 5.9.5.12 ADC\_P2S\_EN

偏移地址：0x2c

复位方式：H/S

Bits	Access	Name	Default	Description
[31:1]	-	reserved		保留
[0]	RW	ADC_P2S_EN	0x0	使能采样数据串行输出，仅用于调试。 1: enable; 0: disable

### 5.9.5.13 ADC\_DATA\_AB

偏移地址：0x30

复位方式：H/S

Bits	Access	Name	Default	Description
[31: 26]	-	reserved		保留
[25:16]	RO	CHANNEL_B_DATA	0x0	通道B扫描值
[15:10]	-	reserved		保留

[9: 0]	RO	CHANNEL_A_DATA	0x0	通道A扫描值
--------	----	----------------	-----	--------

#### 5.9.5.14 ADC\_DATA\_CD

偏移地址：0x34

复位方式：H/S

Bits	Access	Name	Default	Description
[31:26]	-	reserved		保留
[25:16]	RO	CHANNEL_D_DATA	0x0	通道D扫描值
[15:10]	-	reserved		保留
[9:0]	RO	CHANNEL_C_DATA	0x0	通道C扫描值

#### 5.9.5.15 ADC\_DATA\_EF

偏移地址：0x38

复位方式：H/S

Bits	Access	Name	Default	Description
[31: 26]	-	reserved		保留
[25:16]	RO	CHANNEL_F_DATA	0x0	通道F扫描值
[15:10]	-	reserved		保留
[9: 0]	RO	CHANNEL_E_DATA	0x0	通道E扫描值

#### 5.9.5.16 ADC\_DATA\_GH

偏移地址：0x3c

复位方式：H/S

Bits	Access	Name	Default	Description
[31: 26]	-	reserved		保留
[25:16]	RO	CHANNEL_H_DATA	0x0	通道H扫描值
[15:10]	-	reserved		保留
[9: 0]	RO	CHANNEL_G_DATA	0x0	通道G扫描值

#### 5.9.5.17 ADC\_CTRL\_VERID

偏移地址：0x40

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RO	ADC_CTRL_VERID	0x20151216	控制器版本号（仅用于debug）

#### 5.9.5.18 ADC\_DIR\_DAT\_OP

偏移地址：0x44

复位方式：H/S

Bits	Access	Name	Default	Description
[31:10]	-	reserved		保留

[9:0]	RO	ADC_DIR_DAT_OP	0x0	EOC后直接输出ADC数据（仅用于debug）
-------	----	----------------	-----	-------------------------

## 5.10 DAC 控制器

### 5.10.1 概述

HR\_C7000 内置一个 3 通道 DAC，用于输出低速模拟控制信号。

### 5.10.2 功能描述

DAC 控制器用于与内部 DAC 接口，实现 CPU 对 DAC 的控制。

### 5.10.3 工作方式

HR\_C7000 的 DAC 控制器通过 CPU 配置寄存器，直接与内部 DAC 交互，向 DAC 发送数据、使能及低功耗控制信号。

### 5.10.4 寄存器概述

寄存器表 25 DAC 寄存器概述（基地址：0x140D\_0000）

偏移地址	名称	描述	页码
0x0	DAC_PD_CTRL	DAC 使能	
0x4	DAC_PD_MODE_EN	DAC 低功耗使能	
0x8	DAC_DATA_A	DAC 通道 A 数据	
0xc	DAC_DATA_B	DAC 通道 B 数据	
0x10	DAC_DATA_C	DAC 通道 C 数据	
0x14	DAC_CTRL_VERID	DAC 控制器版本	

### 5.10.5 寄存器描述

#### 5.10.5.1 DAC\_PD\_CTRL

偏移地址：0x00

复位方式：H/S

Bits	Access	Name	Default	Description
[31:3]	-	reserved		保留
[2]	RW	DAC_PD_C	0x1	DAC通道C上下电控制。1：掉电；0：上电
[1]	RW	DAC_PD_B	0x1	DAC通道B上下电控制。1：掉电；0：上电
[0]	RW	DAC_PD_A	0x1	DAC通道A上下电控制。1：掉电；0：上电

#### 5.10.5.2 DAC\_PD\_MODE\_EN

偏移地址：0x04

复位方式：H/S

Bits	Access	Name	Default	Description
[31:3]	-	reserved		保留
[2]	RW	DAC_PD_MODE_C EN	0x1	DAC通道C低功耗模式使能。1：使能；0：关断

[1]	RW	DAC_PD_MODE_B EN	0x1	DAC通道B低功耗模式使能。1：使能；0：关断
[0]	RW	DAC_PD_MODE_A EN	0x1	DAC通道A低功耗模式使能。1：使能；0：关断

### 5.10.5.3 DAC\_DATA\_A

偏移地址：0x08

复位方式：H/S

Bits	Access	Name	Default	Description
[31:12 ]	-	reserved		保留
[11:0]	RW	DAC_DATA_A	0x0	DAC通道A的数据

### 5.10.5.4 DAC\_DATA\_B

偏移地址：0x0c

复位方式：H/S

Bits	Access	Name	Default	Description
[31:12 ]	-	reserved		保留
[11:0]	RW	DAC_DATA_B	0x0	DAC通道B的数据

### 5.10.5.5 DAC\_DATA\_C

偏移地址：0x10

复位方式：H/S

Bits	Access	Name	Default	Description
[31:12 ]	-	reserved		保留
[11:0]	RW	DAC_DATA_C	0x0	DAC通道C的数据

### 5.10.5.6 DAC\_CTRL\_VERID

偏移地址：0x14

复位方式：H/S

Bits	Access	Name	Default	Description
[31:0]	RW	DAC_CTRL_VERID	0x20160 229	控制器版本号（仅用于debug）

## 6 Codec 应用

### 6.1 概述

HR\_C7000 芯片内置高性能音频 Codec IP，支持差分 and 单端 Mic 输入，ADC 端支持 AGC 自动增益控制。输出支持两路 lineout 单端输出，PWM 输出模式。

### 6.2 功能描述

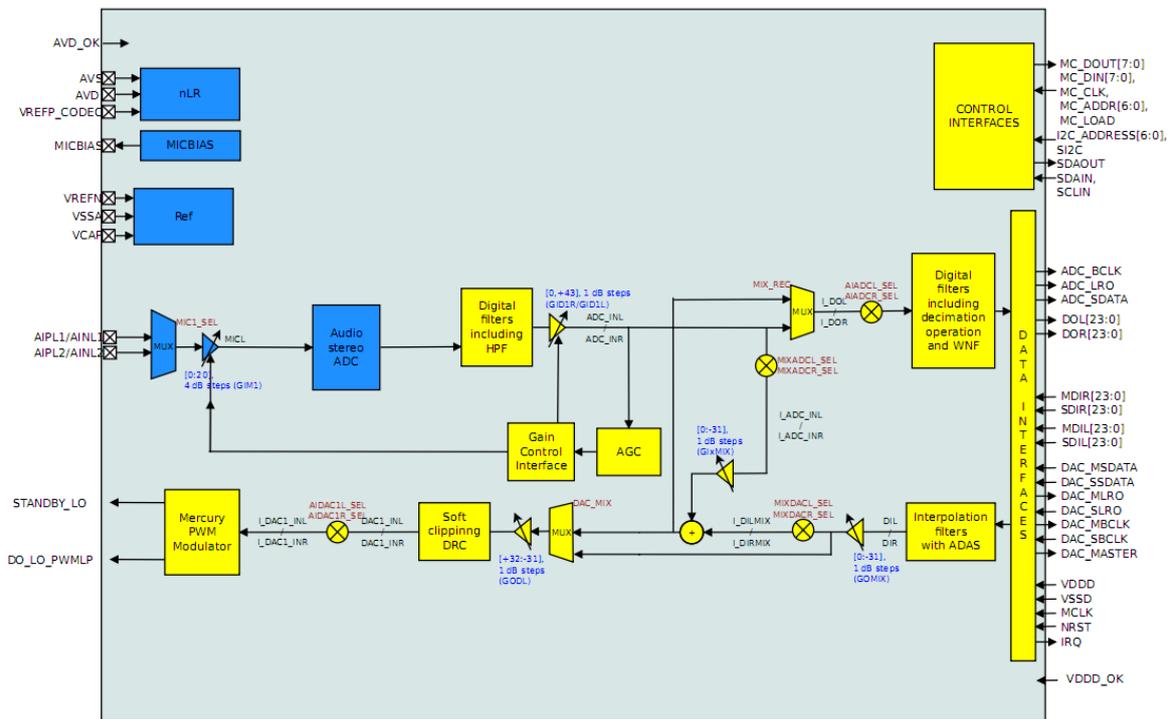


图 55 Codec 模块功能框图

Codec 具有以下特点：

- 主时钟频率24MHz；
- 支持Pop噪声消除；
- 24bit-16bit 线性 PCM 模式输出；
- 支持8K, 16K, 24K, 48K采用频率；
- Fast ADC 唤醒时间典型值100ms；
- 支持自动增益控制（AGC）；
- 支持可编程风噪滤波（WNF）；
- 两路差分/单端Mic输入支持；
- 两路单端Lineout输出支持，Lineout为PWM调制输出。
- ADC工作主模式
- DAC可以选择工作在主模式和从模式

### 6.3 参考电路

#### 6.3.1 MIC 输入

Codec 的 ADC 输入支持差分 Mic 的输入方式和单端 Mic 的输入方式。下图为两种不同的连接方式的原理图示意图。

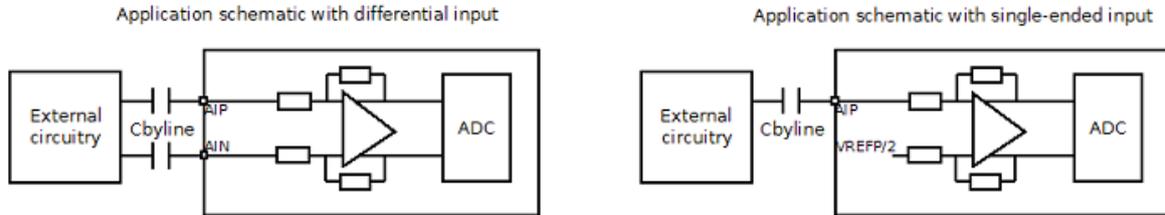


图 56 Codec Mic 差分 and 单端原理图

外部的拾音电路需要通过一组滤波电容 Cbyline 与 Codec 内部隔离，电容大小推荐 1uF。对上述差分或者单端的 Microphone 应该，为了获得更好的性能，减少高频部分干扰，需要有效利用 Micbias 应用电路。

Micbias 应用电路如下图，

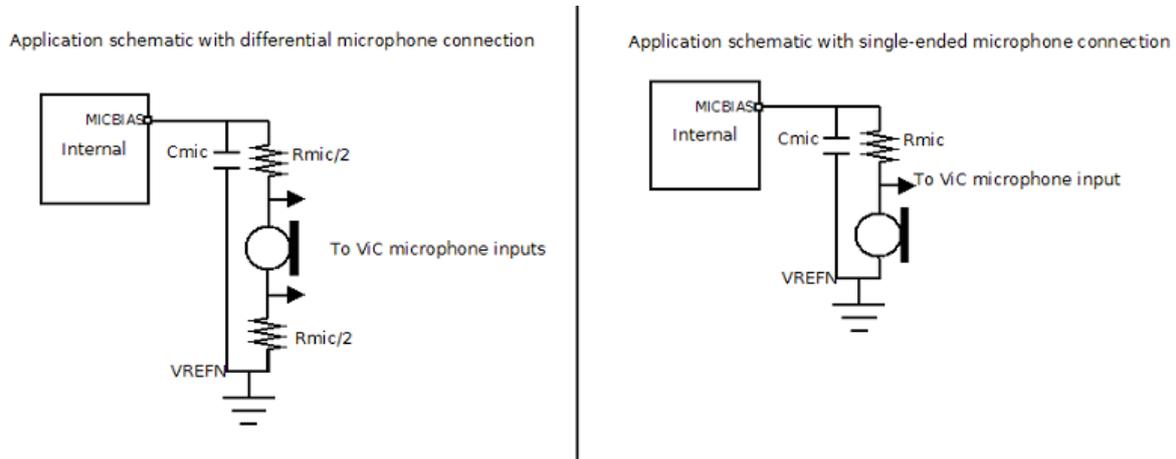


图 57 Codec Micbias 差分 and 单端原理图

Micbias 输出电压依据配置参数的不同，可以配置不同的输出典型值。滤波电路的 Cmic 和 Rmic。

表 38 Codec Micbias 参数说明

Parameter	Test conditions	Min.	Typ	Max.	Unit
Micbias output level	MICBIAS_V = 0		2.08		V
	MICBIAS_V = 1		1.66		
Micbias output current				4	mA
Micbias output noise	A-Weighted		20	40	uVrms
Micbias decoupling capacitor	Cmic	0.75	1	1.25	nF
VCAP output voltage			2		V

#### 6.3.2 LINEOUT 输出

HR\_C7000 内置的 Codec 模块的 DAC 输出是通过 PWM 数字输出的信号调制后实现 DAC 的功能。

芯片的两路 Lineout 输出（Lineout1 和 Lineout2 为同一个 PWM 调制输出相同波形，通过各自的使能控制，可以使得两路 Lineout 同时输出，也可以控制其中任意一路 Lineout 单端输出。

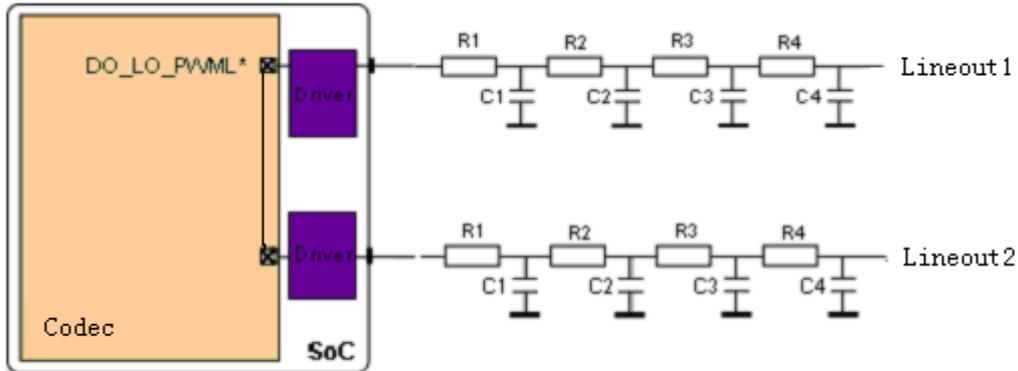


图 58 Codec PWM 输出原理图

上述 PWM 输出的外部整流电路的参考器件推荐如下。

表 39 Codec PWM 输出参数说明

R1	PWM outputs external resistors	2.7	±1%	kOhm
R2		2.7		
R3		2.7		
R4		2.7		
C1	PWM outputs external capacitors	220	±10%	pF
C2		220		
C3		220		
C4		220		

## 6.4 工作方式

### 6.4.1 工作模式描述

表 40 Codec 工作模式定义

Mode	SB	SB_SLEEP	SB_DAC	SB_DAC2	SB_AICR_ADC	SB_AICR_DAC	SB_MIC1	SB_MICBIAS1	SB_ADC	ADC_LEFT_ONLY
Register signal values after a reset mode	1	1	1	1	1	1	1	1	1	1
Power-down mode	1	X	1	1	1	1	X	X	1	X
Playback mode with analog	0	1	0	1	1	0	X	X	1	X

power-down, DAC to BO* ouputs										
Sleep mode	0	1	1	1	1	1	X	X	1	X
Analog Record mode with mono inputs, MIC1 input	0	0	1	1	0	1	0	0	0	1

#### 6.4.1.1 Power-off mode

关闭 CODEC 工作电压。

#### 6.4.1.2 Power-down mode

芯片上电后，对 CODEC 进行复位，此时进入 power-down 模式。在此模式下，CODEC 相关的模拟、数字电源均处于开启状态，但全局 standby (SB) 信号有效，且 codec 的功耗处于最低状态。

#### 6.4.1.3 Playback (语音播放) 模式

此模式下语音播放通路上的数字滤波以及 PWM 调制器处于开启状态。而模拟部分的电源可以通过“SB=1”来关闭。

在当前模式下，禁止进行 codec 复位操作。

#### 6.4.1.4 Sleep mode

此模式下语音播放通路上的数字滤波以及 PWM 调制器处于开启状态。而模拟部分的电源可以通过“SB=1”来关闭。

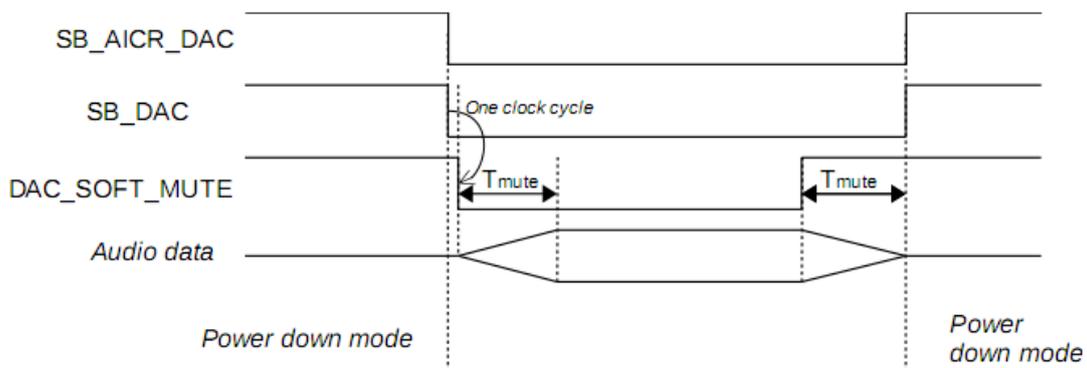
在当前模式下，禁止进行 codec 复位操作。

#### 6.4.1.5 Analog record mode

CODEC 的 ADC 处于工作模式。

### 6.4.2 工作模式切换

#### 6.4.2.1 Power-down 模式切换到语音播放模式



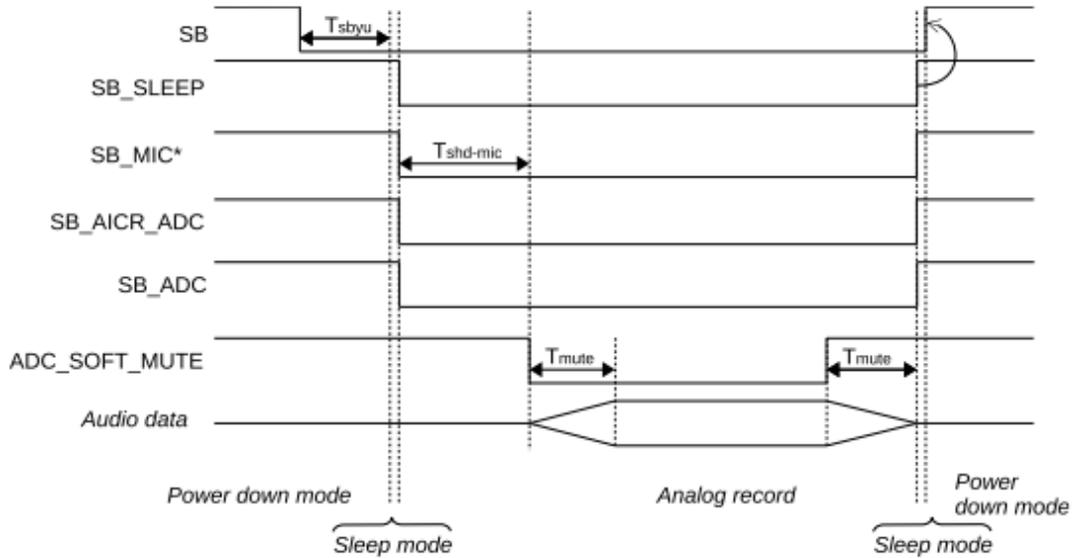
- 配置寄存器 SB\_AICR\_DAC (0x1600\_09C8[4]=0)，开启 DAC 音频接口使能；
- 配置寄存器 SB\_DAC (0x1600\_09CD[4]=0)，开启 DAC
- 配置寄存器 DAC\_SOFT\_MUTE (0x1600\_09CD[7]=0)，关闭 DAC 软静音

其中  $T_{mute}$  最大时延不超过 24ms。

#### 6.4.2.2 Power-down 模式切换到 sleep 模式

设置  $SB=0$ ，CODEC 将从 Power-down 状态切换到休眠模式。

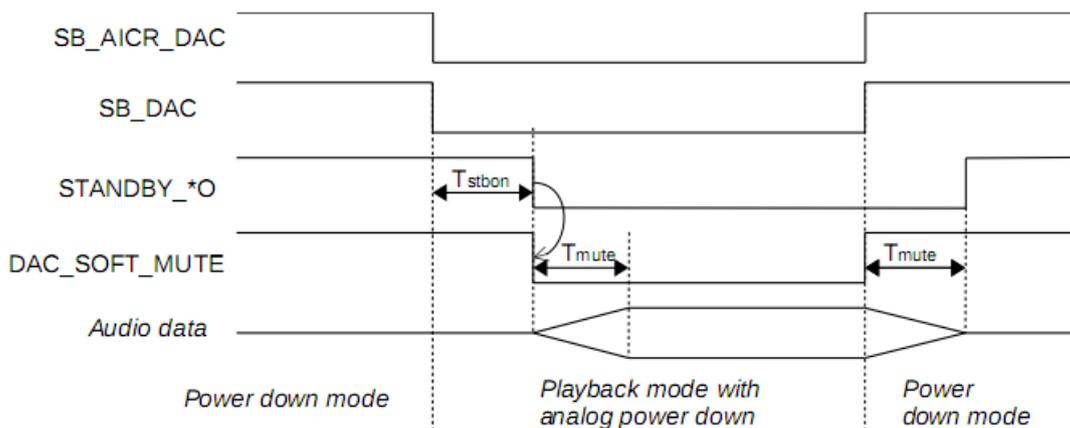
#### 6.4.2.3 Sleep 模式到 CODEC ADC 的开启



- 配置寄存器  $SB$  ( $0x1600\_09D2[0]=0$ )，CODEC 由 power-down 模式切换到休眠模式；
- 配置寄存器  $SB\_SLEEP$  ( $0x1600\_09D2[1]=0$ )，由休眠模式切换到工作模式；
- 配置寄存器  $SB\_MIC1$  ( $0x1600\_09CB[4]=0$ )，开启 MIC1 端口
- 配置寄存器  $SB\_AICR\_ADC$  ( $0x1600\_09C9[4]=0$ )，开启 ADC 音频接口
- 配置寄存器  $SB\_ADC$  ( $0x1600\_09CF[4]=0$ )，开启 ADC
- 配置寄存器  $ADC\_SOFT\_MUTE$  ( $0x1600\_09CF[7]=0$ )，关闭 ADC 软静音功能。

其中  $T_{sbyu}$  与  $T_{shd-mic}$  均为为 50~100ms

#### 6.4.2.4 Sleep 模式到音频播放模式



- 配置寄存器 SB (0x1600\_09D2[0]=0), CODEC 由 power-down 模式切换到休眠模式;
- 配置寄存器 SB\_AICR\_DAC (0x1600\_09C8[4]=0), 开启 DAC 音频接口使能;
- 配置寄存器 SB\_DAC (0x1600\_09CD[4]=0), 开启 DAC
- 配置寄存器 DAC\_SOFT\_MUTE (0x1600\_09CD[7]=0), 关闭 DAC 软静音
- 查询寄存器 0x1100\_0088[31], CPU 获取 STANDBY\_LO 状态, 以开启/关闭扬声器。

其中 Tstbon 在 12ms 以内。

#### 6.4.2.5 任意模式切换到 power-down 模式

- 配置寄存器 SB (0x1600\_09D2[0]=1), CODEC 进入 power-down 模式。
- 配置寄存器 SB\_AICR\_DAC (0x1600\_09C8[4]=1), 关闭 DAC 音频接口使能。
- 配置寄存器 SB\_DAC (0x1600\_09CD[4]=1), 关闭 DAC。
- 配置寄存器 SB\_AICR\_ADC (0x1600\_09C9[4]=1), 关闭 ADC 音频接口。
- 配置寄存器 SB\_ADC (0x1600\_09CF[4]=1), 关闭 ADC。

### 6.4.3 Codec 工作流程

#### 6.4.3.1 音频接口模式设置

HR\_C7000 集成的 CODEC 音频接口为特定配置, 而 CODEC 在复位后, 其内部寄存器的默认值与应用的期望值存在差异, 因此需要重新配置。包括 ADC/DAC 的主、从模式, 信号位宽等。

- 配置寄存器 0x1600\_09C8[7:6] = '11', 选择 DAC 输入信号位宽为 24bit;
- 配置寄存器 0x1600\_09C8[5] = '0', 选择 DAC 工作于主模式;
- 配置寄存器 0x1600\_09C8[1:0] = '00', 选择 DAC 音频接口为并行模式;
- 配置寄存器 0x1600\_09C9[7:6] = '11', 选择 ADC 输出信号位宽为 24bit;
- 配置寄存器 0x1600\_09C9[1:0] = '00', 选择 ADC 音频接口为并行模式;

#### 6.4.3.2 音频发送 (Analog record)

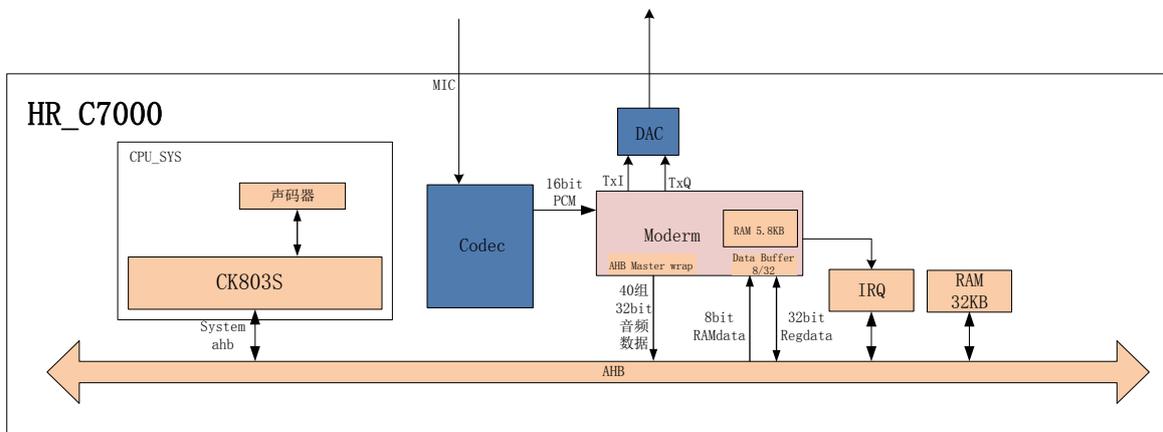


图 59 语音发送数据流转图

如上图所示, 为 MIC 端语音信号到发送端的数据流转图。首先, MIC 端的音频通过 CODEC ADC 转化为 16bit 并行数据以及相应的 8KHz 使能信号。然后, 对于数字/模拟又有所区别: 数

字模式下，音频数据流通过 AHB 输出到声码器，完成音频编码后再返回到 MODEM 的数据缓存中进行 DMR 编码并发送。而模拟模式下，codec ADC 输出的音频数据则是直接在 MODEM 内部完成相应的语音音频调制，并最终给到发送端。具体过程可以参考数据接口手册中的详细说明。

综上所述，无论是数字还是模拟模式，对于 MIC 端的音频发送过程，对 CODEC 的配置是相同的，开启 codec ADC。

- 配置寄存器 0x1100\_0080[0] = 0 ----- 选择音频源为 CODEC ADC;
- 配置寄存器 SB\_SLEEP (0x1600\_09D2[1]=0)，由休眠模式切换到工作模式;
- 配置寄存器 SB\_MIC1 (0x1600\_09CB[4]=0)，开启 MIC1 端口;
- 配置寄存器 SB\_AICR\_ADC (0x1600\_09C9[4]=0)，开启 ADC 音频接口;
- 配置寄存器 SB\_ADC (0x1600\_09CF[4]=0)，开启 ADC;
- 配置寄存器 ADC\_SOFT\_MUTE (0x1600\_09CF[7]=0)，关闭 ADC 软静音功能。

#### 6.4.3.3 音频接收 (Play back)

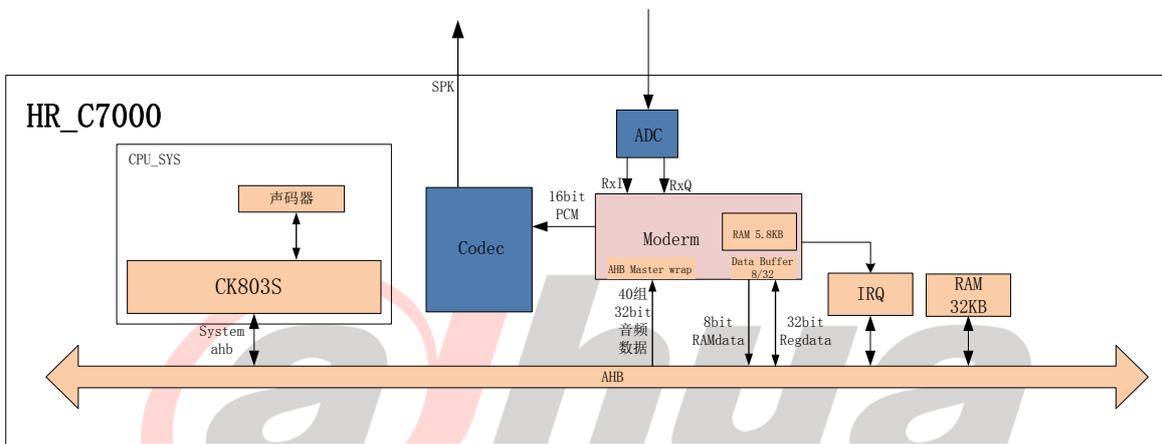


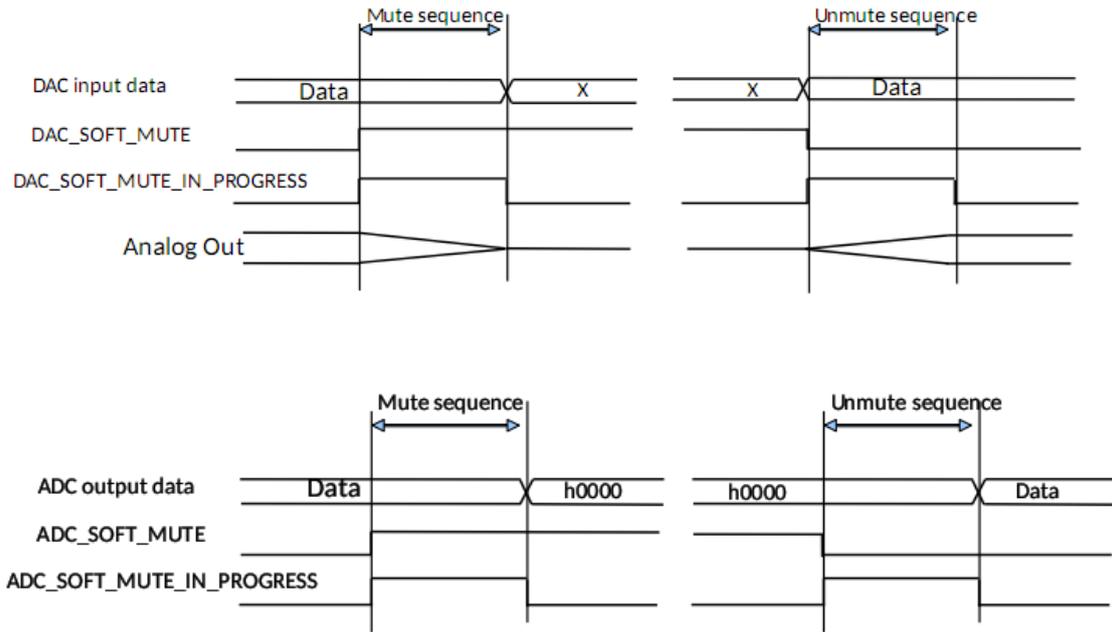
图 60 语音接收数据流转图

如上图所示，为接收解调后的音频信号到 CODEC Line\_out/HPout 的数据流转图。对于模拟模式，射频接收信号在 MODEM 内部完成 FM 解调，输出的音频数据直接给到 CODEC DAC 端。而对于数字模式，DMR 解调后的数据给到声码器解码，再将解码后的音频数据返回到 MODEM 端，并最终给到 CODEC DAC 端。

综上所述，无论是数字还是模拟模式的音频播放过程，对 CODEC 的配置是相同的。开启 codec DAC。

- 配置寄存器 SB\_SLEEP (0x1600\_09D2[1]=0)，由休眠模式切换到工作模式;
- 配置寄存器 SB\_AICR\_DAC (0x1600\_09C8[4]=0)，开启 DAC 音频接口使能;
- 配置寄存器 SB\_DAC (0x1600\_09CD[4]=0)，开启 DAC;
- 配置寄存器 DAC\_SOFT\_MUTE (0x1600\_09CD[7]=0)，关闭 DAC 软静音。

#### 6.4.4 软静音功能 (soft mute)



CODEC 内置 ADC/DAC 软静音功能,通过分别通过寄存器 0x1600\_09CF[7](adc\_soft\_mute)、0x1600\_09CD[7] (dac\_soft\_mute) 置 ‘1’ 有效。

CODEC 的输出信号将在 Mute sequence (unmute sequence) 时间段内逐步增加 (减小) 信号幅值。通常情况下, 该时间持续在 24ms 以内。

#### 6.4.5 AGC

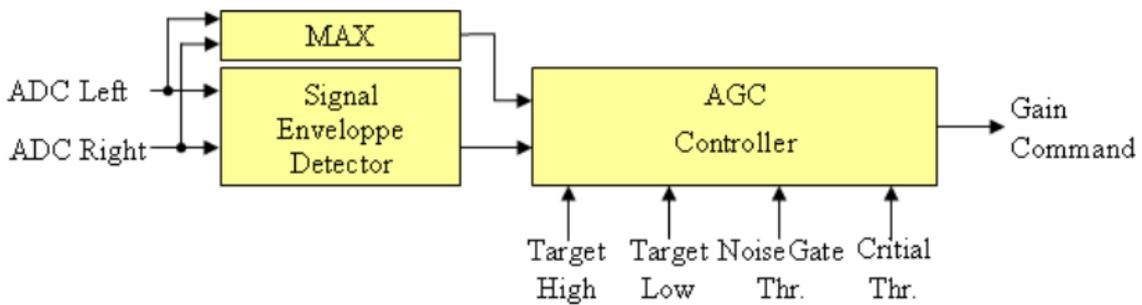


图 61 CODEC 自动增益控制 (AGC) 系统框图

上图为 CODEC ADC 的自动增益控制 (AGC) 系统框图。ADC 产生的信号首先通过信号包络检测模块 (signal envelope detector), 计算信号均值, 然后再 AGC 控制模块中完成 4 种门限的判决, 并产生控制指令返回到 ADC 单元。

四组门限值的定义如下:

边界门限 (Critical threshold): 判断 ADC 当前的输出值是否饱和, 当 ADC 输出值大于该门限时, AGC 将迅速的减小 ADC 输出增益。CRIT\_THR = FS - 3dB, 即边界门限由 CODEC ADC 的满幅值 (Full Scale) 决定, 用户无需配置。

高目标值 (Target High)、低目标值 (Target Low): 这两组门限值用来定义信号幅度的期望

区间，分别为  $Target\_high = (Target + 2dB)$ ;  $Target\_low = (Target - 4dB)$ 。当信号处于该区间内，AGC 不做任何调整。AGC 完成一次信号幅度增加或减小所持续的时间，分别由寄存器 DCY 与 ATK 进行设置。

噪声门限 (Noise Gate): 该门限用于判决当前 ADC 的输出为噪声或者是有效音频信号。当 ADC 输出值低于 NG\_THR 时，被判定为噪声信号，并且此时 AGC 不做任何调整。NG\_THR 由寄存器配置。其定义为  $NG\_THR = MIN\_NG\_THR + a$  (in dB)。以 1KHz 单音信号为例，MIN\_NG\_THR 代表-68dB。

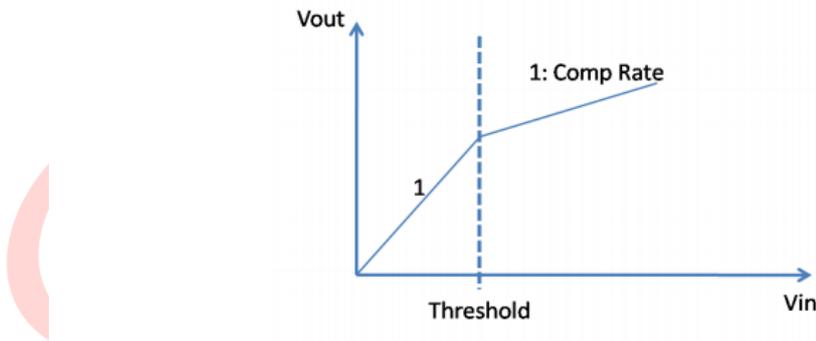
应用实例 (TARGET=满幅-6dB):

- 配置寄存器(0x1600\_09F0 =0x18);
- 配置寄存器(0x1600\_09EF =0x40); 将缓存于 reg\_0x1600\_09F0 中的数据，锁存到间接地址为 00(ADC\_AGC\_0)的寄存器中;
- 配置寄存器(0x1600\_09EF =0x80)，开启 ADC 的 AGC 功能;

注意: AGC 的使能一定要等参数配置完成以后，单独开启!

#### 6.4.6 Soft clipping DRC

当 DAC 输入数据高于设定的门限值时，对超出的部分进行音频压缩，以降低音频幅值。



signal above threshold at the output of the soft clipping DRC

$$SCDRC_{ioutput} = \frac{(DAC_{input} \times GOMIX * \times GOD*) - Threshold}{CompRate} + Threshold$$

#### 6.4.7 Wind Noise Filter

CODEC 内置有风噪滤波器，由寄存器 FCR\_ADC (0x1600\_0017) 选择开启或关闭，并支持 3 组降噪模式设置。

表 41 Codec 风噪滤波器参数

Filter characteristics	24MHz		
Wind Noise Filter corner frequency	Mode 1	-3dB	59 Hz
	Mode 2	-3dB	117 Hz
	Mode 3	-3dB	235 Hz

## 6.5 寄存器

### 6.5.1 寄存器概述

寄存器表 26 Codec 寄存器概述（基地址：0x1600\_09C0）

偏移地址	名称	描述	页码
0x0	SR1	状态寄存器	
0x1	SR2	状态寄存器	
0x7	MR	ADC 静音控制	
0x8	AICR_DAC	DAC 音频接口控制	
0x9	AICR_ADC	ADC 音频接口控制	
0xB	CR_MIC1	MIC1 控制寄存器	
0xC	CR_MIC2	MIC2 控制寄存器	
0xD	CR_DAC	DAC 控制寄存器	
0xF	CR_ADC	ADC 控制寄存器	
0x10	CR_MIX	数字混叠控制寄存器	
0x11	DR_MIX	数字混叠控制间接寄存器	
0x12	CR_VIC	Codec 的 sleep 模式控制寄存器	
0x13	CR_CK	MCLK 控制寄存器	
0x14	FCR_DAC	DAC 采样时钟配置寄存器	
0x17	FCR_ADC	ADC 采样时钟配置寄存器	
0x18	CR_TIMER_h	计数器高 8bit	
0x19	CR_TIMER_l	计数器低 8bit	
0x1A	ICR	中断控制寄存器	
0x1B	IMR	中断 MASK 控制寄存器	
0x1C	IFR	中断标志定义	
0x1D	IMR2	定时器标志 MASK 控制寄存器	
0x1E	IFR2	定时器标志定义	
0x1F	GCR_DACL	DAC 增益控制	
0x23	GCR_MIC1	Mic1 增益控制	
0x25	GCR_ADCL	ADC 增益控制	
0x27	GCR_MIXDACL	混叠 DAC 增益控制	
0x29	GCR_MIXADCL	混叠 ADC 增益控制	
0x2B	CR_DAC_AGC	DAC 增益控制寄存器	
0x2C	DR_DAC_AGC	DAC AGC 间接地址的读写数据	
0x2F	CR_ADC_AGC	ADC 增益控制寄存器	
0x30	DR_ADC_AGC	ADC AGC 间接地址的读写数据	

### 6.5.2 寄存器描述

#### 6.5.2.1 SR1

偏移地址：0x00

复位方式：H

Bits	Access	Name	Default	Description
------	--------	------	---------	-------------

[7]	RO	pon_ack	0x0	codec power on状态响应 0: codec未进入工作模式; 1: codec进入工作模式
[6]	RO	irq_ack	0x0	中断挂起状态寄存器 0: 未包含待处理的中断; 1: 当前存在未响应的中断
[5]	RO	reserved	0x0	
[4]	RO	dac_locked	0x0	DAC ADAS锁定状态, 1有效(该功能在当前应用中不涉及)
[3:0]	RO	reserved	0x0	

### 6.5.2.2 SR2

偏移地址: 0x01

复位方式: H

Bits	Access	Name	Default	Description
[7:5]	RO	reserved	0x0	
[4]	RO	dac_unknown_fs	0x0	DAC_ADAS锁定后的应答标志: 0: 输入信号的FS处于codec支持的频率范围内; 1: 输入信号的FS超出codec支持的范围(此时的性能指标无法得到保证), 该功能在当前应用中不涉及。
[3:0]	RO	reserved	0x0	

### 6.5.2.3 MR

偏移地址: 0x07

复位方式: H

Bits	Access	Name	Default	Description
[7:5]	RO	reserved	0x0	
[4:3]	RO	adc_mute	0x0	00: ADC非静音; 01: ADC正在被静音; 10: ADC即将结束静音状态; 11: ADC处于静音状态
[2]	RO	reserved	0x0	
[1:0]	RO	dac_mute	0x0	00: DAC非静音; 01: DAC正在被静音; 10: DAC即将结束静音状态; 11: DAC处于静音状态

### 6.5.2.4 AICR\_DAC

偏移地址: 0x08

复位方式: H

Bits	Access	Name	Default	Description
[7:6]	RW	dac_adwl	0x3	串行传输方式下，DAC音频数据位宽设置： 00: 16bit; 01: 18bit; 10: 20bit; 11: 24bit。 复位后初始设置为24bit，modem内部16bit信号已做扩位处理。
[5]	RW	dac_slave	0x0	DAC音频接口主从模式配置，0: Master; 1: Slave
[4]	RW	sb_aicr_dac	0x1	DAC音频接口使能，0: active; 1: power-down
[3:2]	RW	reserved	0x0	
[1:0]	RW	dac_audioif	0x3	DAC音频接口模式选择， 00: Parallel; 01: Left-justified; 10: DSP; 11: I2S 复位后初始配置为I2S模式，需要固定配置为Parallel模式

#### 6.5.2.5 AICR\_ADC

偏移地址：0x09

复位方式：H

Bits	Access	Name	Default	Description
[7:6]	RW	adc_adwl	0x3	串行传输方式下，ADC音频数据位宽设置， 00: 16bit; 01: 18bit; 10: 20bit; 11: 24bit 复位后初始设置为24bit，modem内部16bit信号已做扩位处理。
[5]	RW	adc_slave	0x0	ADC音频接口主从模式配置，0: Master; 1: Slave
[4]	RW	sb_aicr_adc	0x1	ADC音频接口使能，0: active; 1: power-down
[3:2]	RW	reserved	0x0	
[1:0]	RW	adc_audioif	0x3	ADC音频接口模式选择， 00: Parallel; 01: Left-justified; 10: DSP; 11: I2S 复位后初始配置为I2S模式，需要固定配置为Parallel模式

#### 6.5.2.6 CR\_MIC1

偏移地址：0x0B

复位方式：H

Bits	Access	Name	Default	Description

[7]	RW	reserved	0x0	
[6]	RW	micdiff1	0x0	MIC1输入模式选择, 0: 单端输入; 1: 差分输入
[5]	RW	sb_micbias1	0x1	MIC1 BIAS使能, 0: active; 1: power-down
[4]	RW	sb_mic1	0x1	MIC1 使能, 0: active; 1: power-down
[3]	RW	micbias1_v	0x0	MICbias 电压选择, 0: 2.08V; 1: 1.66V
[2:0]	RW	reserved	0x0	

### 6.5.2.7 CR\_MIC2

偏移地址: 0x0C

复位方式: H

Bits	Access	Name	Default	Description
[7]	RW	reserved	0x0	
[6]	RW	micdiff2	0x0	MIC1输入模式选择, 0: AIPL1/AINL1; 1: AIPL2/AINL2
[5:0]	RW	reserved	0x30	

### 6.5.2.8 CR\_DAC

偏移地址: 0x0D

复位方式: H

Bits	Access	Name	Default	Description
[7]	RW	dac_soft_mute	0x1	DAC软静音模式, 0: 软静音关闭; 1: 软静音开启
[6:5]	RW	reserved	0x1	
[4]	RW	sb_dac	0x1	Mercury DAC工作使能, 0: active; 1: power-down
[3:1]	RW	reserved	0x0	
[0]	RW	dac_zero_n	0x0	DO_LO_PWM*N管脚输出禁止, 0: active; 1: 钳制在VSSD

### 6.5.2.9 CR\_ADC

偏移地址: 0x0F

复位方式: H

Bits	Access	Name	Default	Description
[7]	RW	adc_soft_mute	0x1	ADC软静音模式, 0: 软静音关闭; 1: 软静音开启
[6]	RW	adc_dmic_sel	0x0	数字滤波器输入选择, 0: 模拟mic; 1: 数字mic(无)
[5]	RW	adc_left_only	0x1	ADC右声道屏蔽, 0: 启用右声道; 1: 禁止右声道
[4]	RW	sb_adc	0x1	ADC工作使能, 0: 开启; 1: 关闭
[3:0]	RW	reserved	0x0	

### 6.5.2.10 CR\_MIX

偏移地址: 0x10

复位方式: H

Bits	Access	Name	Default	Description
[7]	RW	mix_en	0x0	数字混叠使能, 0: 数字混叠关闭; 1: 数字混叠开启
[6]	RW	mix_load	0x0	数字混叠间接地址空间读写控制 0: 读; 1: 写
[5:0]	RW	mix_add	0x0	数字混叠间接地址

MIX_0 (0x00)	Digital mixer control register 0	[7:6]	RW	aidacl_sel	0x0	DAC1 modulator 输入音频左声道音源选择, 00: I_DAC1_INL = DAC1_INL; 01: I_DAC1_INL = DAC1_INR; 10: I_DAC1_INL = (DAC1_INL+DAC1_INR)/2; 11: I_DAC1_INL = 0
		[5:4]	RW	aidacr_sel	0x0	DAC1 modulator 输入音频右声道音源选择, 00: I_DAC1_INR = DAC1_INR; 01: I_DAC1_INR = DAC1_INL; 10: I_DAC1_INR = (DAC1_INL+DAC1_INR)/2; 11: I_DAC1_INR = 0
		[3:1]	RW	reserved	0x0	
		[0]	RW	dac_mix	0x0	DAC1 输入音频混叠使能, 0: 仅 DAC; 1: DAC+ADC
MIX_1 (0x01)	Digital mixer control register 1	[7:6]	RW	mixdacl_sel	0x0	DAC 给到混叠器的输入音频左声道音源选择, 00: I_DILMIX = DIL; 01: I_DILMIX = DIR; 10: I_DILMIX = (DIL+DIR)/2; 11: I_DILMIX = 0
		[5:4]	RW	mixdacr_sel	0x0	DAC 给到混叠器的输入音频右声道音源选择, 00: I_DIRMIX = DIR; 01: I_DIRMIX = DIL; 10: I_DIRMIX = (DIL+DIR)/2; 11: I_DIRMIX = 0

		[3:0]	RW	reserved	0x0	
MIX_2 (0x02)	Digital mixer control register 2	[7:6]	RW	aiadcl_sel	0x0	ADC 给到端口的左声道音源选择, 00: DOL = I_DOL; 01: DOL = I_DOR; 10: DOL = (I_DOL+I_DOR)/2; 11: DOL = 0
		[5:4]	RW	aiadcr_sel	0x0	ADC 给到端口的右声道音源选择, 00: DOR = I_DOR; 01: DOR = I_DOL; 10: DOR = (I_DOL+I_DOR)/2; 11: DOR = 0
		[3:1]	RW	reserved	0x0	
		[0]	RW	mix_rec	0x0	ADC 输出数据的混叠使能, 0: 仅 ADC; 1: ADC+DAC
MIX_3 (0x03)	Digital mixer control register 3	[7:6]	RW	mixadcl_sel	0x0	ADC 给到混叠器的输入音频左声道音源选择, 00: I_ADC_INL = ADC_INL; 01: I_ADC_INL = ADC_INR; 10: I_ADC_INL = (ADC_INL+ADC_INR)/2; 11: I_ADC_INL = 0
		[5:4]	RW	mixadcr_sel	0x0	ADC 给到混叠器的输入音频右声道音源选择, 00: I_ADC_INR = ADC_INR; 01: I_ADC_INR = ADC_INL; 10: I_ADC_INR = (ADC_INL+ADC_INR)/2; 11: I_ADC_INR = 0
		[3:0]	RW	reserved	0x0	

#### 6.5.2.11 DR\_MIX

偏移地址: 0x11

复位方式: H

Bits	Access	Name	Default	Description
[7:0]	RW	mix_data	0x0	数字混叠间接地址的读写数据

### 6.5.2.12 CR\_VIC

偏移地址：0x12

复位方式：H

Bits	Access	Name	Default	Description
[7:2]	RW	reserved	0x03	
[1]	RW	sb_sleep	0x1	sleep 模式， 0: active; 1: sleep
[0]	RW	sb	0x1	所有的 codec 模拟部分关闭模式， 0: active; 1: analog power-down

### 6.5.2.13 CR\_CK

偏移地址：0x13

复位方式：H

Bits	Access	Name	Default	Description
[7]	RW	reserved	0x0	
[6]	RW	mclk_div	0x1	0: MCLK/1 (保留为 Dolphin 应用); 1: MCLK/2
[5]	RW	reserved	0x0	
[4]	RW	shutdown_clock	0x0	关闭 MCLK 时钟, 0: active; 1: MCLK 关闭 (恒为 0), power-down 模式下可以关闭
[3:0]	RW	crystal	0x0	指明 MCLK 频率 (当 MCLK_DIV=1), 0000: MCLK=24MHz; 0010: MCLK=26MHz

### 6.5.2.14 FCR\_DAC

偏移地址：0x14

复位方式：H

Bits	Access	Name	Default	Description
[7:4]	RW	reserved	0x0	
[3:0]	RW	dac_freq	0x0	DAC 采样时钟选择, 0000:8KHz; 其它略

### 6.5.2.15 SFCCR\_DAC

偏移地址：0x15

复位方式：H

Bits	Access	Name	Default	Description
[7]	RW	dac_freq_valid	0x0	0: Inactive; 1: master 模式下 FS 校准功能开启 (本应用中为固定配置, 无需更改)
[6:0]	RW	dac_freq_adj_h	0x0	

### 6.5.2.16 SFFCR\_DAC

偏移地址：0x16

复位方式：H

Bits	Access	Name	Default	Description
[7:0]	RW	dac_freq_adj_l	0x0	FS 与 FS0 之间差异的协调配置字 $dac\_freq\_adj=128 * (MCLK/2) * (1/FS-1/FS0)$ (该功能在当前应用中不涉及)

#### 6.5.2.17 FCR\_ADC

偏移地址: 0x17

复位方式: H

Bits	Access	Name	Default	Description
[7]	RW	reserved	0x0	
[6]	RW	adc_hpf	0x0	ADC 高通滤波器使能, 0: 关闭; 1: 开启
[5:4]	RW	adc_wnf	0x0	ADC 风噪滤波器使能, 00: 关闭; 01: ADC 风噪滤波器 Mode1; 10: ADC 风噪滤波器 Mode2; 11: ADC 风噪滤波器 Mode3;
[3:0]	RW	adc_freq	0x0	ADC 采样时钟选择, 0000:8KHz; 其它略

#### 6.5.2.18 CR\_TIMER\_h

偏移地址: 0x18

复位方式: H

Bits	Access	Name	Default	Description
[7:0]	RW	count_h	0x0	计时器, 以慢时钟递减(慢时钟通常为 512MCLK), 当 count 递减到 0 时, 产生中断。 计时周期最小为 $(3+count) * slow\_clock$ ; 计时周期最大为 $(4+count) * slow\_clock + 10 * MCLK$

#### 6.5.2.19 CR\_TIMER\_l

偏移地址: 0x19

复位方式: H

Bits	Access	Name	Default	Description
[7:0]	RW	count_l	0x0	与 count_h 配合使用

#### 6.5.2.20 ICR

偏移地址: 0x1a

复位方式: H

Bits	Access	Name	Default	Description
[7:6]	RW	int_form	0x0	中断信号输出方式, 00:IRQ 为持续高电平; 01:IRQ 为持续低电平; 10:IRQ 为正向脉冲, 脉冲宽度为 8 个 MC_CLK (Parallel) 或者 8 个 MCLK (I2C);

				11: IRQ 为负向脉冲, 脉冲宽度为 8 个 MC_CLK (Parallel) 或者 8 个 MCLK (I2C);
[5:0]	RW	reserved	0x0	

#### 6.5.2.21 IMR

偏移地址: 0x1b

复位方式: H

Bits	Access	Name	Default	Description
[7]	RW	adas_lock_mask	0x1	屏蔽 adas 锁定标志
[6:3]	RW	reserved	0xf	
[2]	RW	adc_mute_mask	0x1	屏蔽 ADC 静音标志
[1]	RW	reserved	0x1	
[0]	RW	dac_mute_mask	0x1	屏蔽 DAC 静音标志

#### 6.5.2.22 IFR

偏移地址: 0x1c

复位方式: H

Bits	Access	Name	Default	Description
[7]	RW	adas_lock_event	0x0	ADAS 锁定事件标志, 当 DAC ADAS 在锁定与失锁之间切换时, 产生该标志 (该功能在当前应用中不涉及, DAC 恒工作于 Master 模式)
[6:3]	RW	reserved	0x0	
[2]	RW	adc_mute_event	0x0	ADC 软静音模式事件标志, 当 ADC 软静音模式开启或关闭时, 产生该标志 (该功能在当前应用中不涉及)
[1]	RW	reserved	0x0	
[0]	RW	dac_mute_event	0x0	DAC 软静音模式事件标志, 当 DAC 软静音模式开启或关闭时, 产生该标志 (该功能在当前应用中不涉及)

#### 6.5.2.23 IMR2

偏移地址: 0x1d

复位方式: H

Bits	Access	Name	Default	Description
[7:5]	RW	reserved	0x7	
[4]	RW	TIMER_END_MASK	0x1	定时器中断屏蔽
[3:0]	RW	reserved	0xf	

#### 6.5.2.24 IFR2

偏移地址: 0x1e

复位方式: H

Bits	Access	Name	Default	Description
[7:5]	RW	reserved	0x0	
[4]	RW	TIMER_END	0x0	定时器计数结束中断
[3:0]	RW	reserved	0x0	

#### 6.5.2.25 GCR\_DACL

偏移地址: 0x1f

复位方式: H

Bits	Access	Name	Default	Description
[7]	RW	lrgod	0x0	DAC1 左右声道共用左声道的配置使能, 0: 相互独立配置; 1: 共用左声道配置
[6]	RW	reserved	0x0	
[5:0]	RW	godl	0x0	DAC1 左声道增益配置, 011111: -31dB; 011110: -30dB; 累加 1dB 000001: -1dB 000000: 0dB; 111111: 1dB; 111110: 2dB; 累加 1dB 100001: 31dB; 100000: 32dB;

#### 6.5.2.26 GCR\_DACR

偏移地址: 0x20

复位方式: H

Bits	Access	Name	Default	Description
[7:6]	RW	reserved	0x0	
[5:0]	RW	godr	0x0	DAC1 右声道增益配置, 011111: -31dB; 011110: -30dB; 累加 1dB 000001: -1dB 000000: 0dB; 111111: 1dB; 111110: 2dB; 累加 1dB 100001: 31dB; 100000: 32dB;

### 6.5.2.27 GCR\_MIC1

偏移地址：0x23

复位方式：H

Bits	Access	Name	Default	Description
[7:3]	RW	reserved	0x0	
[2:0]	RW	gim1	0x0	MIC1 增益步进幅度 (boost stage) 000: 0dB; 001: 4dB; 010: 8dB; 011: 12dB; 100: 16dB; 其它配置均为 20dB;

### 6.5.2.28 GCR\_ADCL

偏移地址：0x25

复位方式：H

Bits	Access	Name	Default	Description
[7]	RW	lrgid	0x0	ADC 左右声道共用左声道的配置使能, 0: 相互独立配置; 1: 共用左声道配置
[6]	RW	reserved	0x0	
[5:0]	RW	gidl	0x0	ADC 左声道增益配置, 000000: 0dB; 000001: 1dB; 累加 1dB 101010: 42dB; 其它: 43dB; 当 gidl 值发生改变时, ADC 数据输出在 64 个采样 时钟后生效。此外高通滤波器的开启, 也会引入通 路上的延迟。

### 6.5.2.29 GCR\_ADCR

偏移地址：0x26

复位方式：H

Bits	Access	Name	Default	Description
[7:6]	RW	reserved	0x0	
[5:0]	RW	gidr	0x0	ADC 右声道增益配置, 000000: 0dB; 000001: 1dB; 累加 1dB 101010: 42dB; 其它: 43dB;

### 6.5.2.30 GCR\_MIXDACL

偏移地址：0x27

复位方式：H

Bits	Access	Name	Default	Description
[7]	RW	lrgomix	0x0	混叠器DAC左右声道增益共用左声道的配置使能， 0：相互独立配置；1：共用左声道配置
[6]	RW	reserved	0x0	
[5:0]	RW	gomixl	0x0	混叠器 DAC 左声道增益配置， 000000： 0dB； 000001： -1dB； 递减 1dB 011111： -31dB；

#### 6.5.2.31 GCR\_MIXDACR

偏移地址：0x28

复位方式：H

Bits	Access	Name	Default	Description
[7:6]	RW	reserved	0x0	
[5:0]	RW	gomixr	0x0	混叠器 DAC 右声道增益配置， 000000： 0dB； 000001： -1dB； 递减 1dB 011111： -31dB；

#### 6.5.2.32 GCR\_MIXADCL

偏移地址：0x29

复位方式：H

Bits	Access	Name	Default	Description
[7]	RW	lrgimix	0x0	混叠器ADC左右声道增益共用左声道的配置使能， 0：相互独立配置；1：共用左声道配置
[6]	RW	reserved	0x0	
[5:0]	RW	gimixl	0x0	混叠器 ADC 左声道增益配置， 000000： 0dB； 000001： -1dB； 递减 1dB 011111： -31dB；

#### 6.5.2.33 GCR\_MIXADCR

偏移地址：0x2a

复位方式：H

Bits	Access	Name	Default	Description
------	--------	------	---------	-------------

[7:6]	RW	reserved	0x0	
[5:0]	RW	gimixr	0x0	混叠器 ADC 右声道增益配置, 000000: 0dB; 000001: -1dB; 递减 1dB 011111: -31dB;

#### 6.5.2.34 CR\_DAC\_AGC

偏移地址: 0x2b

复位方式: H

Bits	Access	Name	Default	Description
[7]	RW	dac_agc_en	0x0	DAC AGC 使能, 0: 自动增益关闭; 1: 自动增益开启
[6]	RW	dac_agc_load	0x0	DAC AGC 间接地址空间读写控制 0: 读; 1: 写
[5:0]	RW	dac_agc_add	0x0	DAC AGC 间接地址

#### 6.5.2.35 DR\_DAC\_AGC

偏移地址: 0x2c

复位方式: H

Bits	Access	Name	Default	Description
[7:0]	RW	dac_agc_data	0x0	DAC AGC 间接地址的读写数据

DAC_AGC_0(0x00)	DAC1 DRC Control register 0	[7]	RW	lr_drc	0x0	DAC1 DRC 左右声道共用左声道的配置使能, 0: 相互独立配置; 1: 共用左声道配置
		[6:5]	RW	reserved	0x0	
		[4:0]	RW	lthres	0x0	DAC1 左声道 DRC 压缩门限 (compression threshold), 00000: 0dB; 00001: -1dB; 递减 1dB 11111: -31dB;
DAC_AGC_1(0x01)	DAC1	[7:3]	RW	reserved	0x0	

	DRC Control register 1		[2:0]	RW	lcomp	0x0	DAC1 左声道 DRC 压缩率 (compression rate), 000: 1; 001: 2; step X 2 101: 32;
DAC_AGC_2(0x02)	DAC1 DRC Control register 2		[7:5]	RW	reserved	0x0	
			[4:0]	RW	rthres	0x0	DAC1 右声道 DRC 压缩门限 (compression threshold), 00000: 0dB; 00001: -1dB; 递减 1dB 11111: -31dB;
DAC_AGC_3(0x03)	DAC1 DRC Control register 3		[7:3]	RW	reserved	0x0	
			[2:0]	RW	rcomp	0x0	DAC1 右声道 DRC 压缩率 (compression rate), 000: 1; 001: 2; step X 2 101: 32;

### 6.5.2.36 CR\_ADC\_AGC

偏移地址: 0x2f

复位方式: H

Bits	Access	Name	Default	Description
[7]	RW	adc_agc_en	0x0	ADC AGC 使能, 0: 自动增益关闭; 1: 自动增益开启
[6]	RW	adc_agc_load	0x0	ADC AGC 间接地址空间读写控制 0: 读; 1: 写
[5:0]	RW	adc_agc_add	0x0	ADC AGC 间接地址

### 6.5.2.37 DR\_ADC\_AGC

偏移地址: 0x30

复位方式: H

Bits	Access	Name	Default	Description
[7:0]	RW	adc_agc_data	0x0	ADC AGC 间接地址的读写数据

ADC_AGC_0(0x00)	ADC AGC Control Register 0	[7]	RW	reserved	0x0	
		[6]	RW	agc_stereo	0x0	ADC AGC 配置, 左右声道共用左声道的配置使能, 0: 相互独立配置; 1: 共用左声道配置
		[5:2]	RW	target	0xd	ADC 输出幅度目标区间 0000: Max_target; 0001: Max_target-1.5dB; 递减 1.5dB 1111: Max_target-22.5dB
		[1:0]	RW	reserved	0x0	
ADC_AGC_1(0x01)	ADC AGC Control Register 1	[7]	RW	ng_en	0x0	noise gate 使能, 1 有效
		[6:4]	RW	ng_thr	0x0	noise gate 门限值 000: MIN_NG_THR; 001: MIN_NG_THR+6dB; 递增 6dB 111: MIN_NG_THR+42dB
		[3:0]	RW	hold	0x7	AGC 功能启动之前的信号保持时间 0000: 0ms 0001: 2ms step * 2 1000: 256ms ..... 1111: 32.768s

ADC_AGC_2(0x02)	ADC AGC Control Register 2	[7:4]	RW	atk	0x4	增益 ramp 下降时间 0000 to 0001: 保留 0010: 96ms 递增 32ms 1111: 512ms
		[3:0]	RW	dcy	0x4	增益 ramp 上升时间 0000 to 0001: 保留 0010: 96ms 递增 32ms 1111: 512ms
ADC_AGC_3(0x03)	ADC AGC Control Register 3	[7:5]	RW	reserved	0x0	
		[4:0]	RW	agc_max	0x1f	ADC 通路最大增益值 00000: 0 递增 1.5dB 01111: 22.5 10000: 23 10001: 23 10010: 23 10011: 24.5 递增 1.5 11111: 42.5
ADC_AGC_4(0x04)	ADC AGC Control Register 4	[7:5]	RW	reserved	0x0	
		[4:0]	RW	agc_min	0x00	ADC 通路最小增益值 00000: 0 递增 1.5dB 01111: 22.5 10000: 23 10001: 23 10010: 23 10011: 24.5 递增 1.5 11111: 42.5

## 7 音频通路

### 7.1 DMR 模式

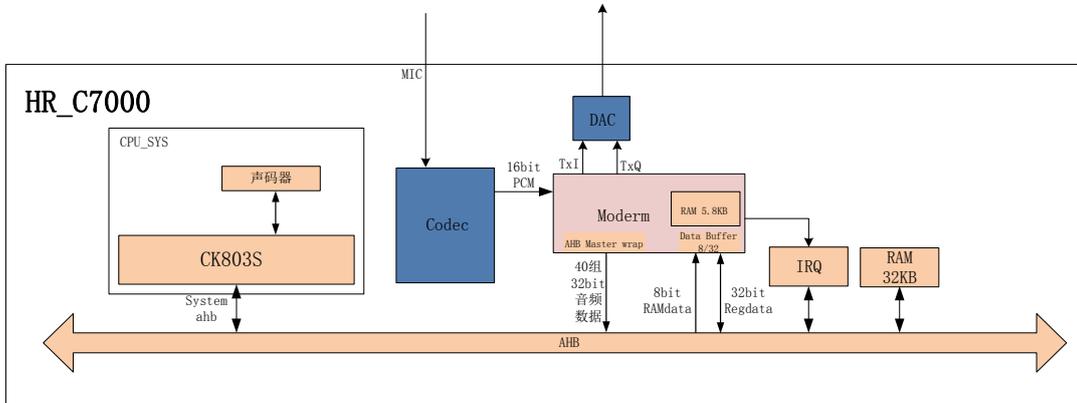
#### 7.1.1 概述

HR\_C7000 内置声码器模块，支持对音频数据的编解码功能。数字模式下，实现对 codec 音频数据的编码发送，以及解码播放功能。同时，支持数字通话录音功能。

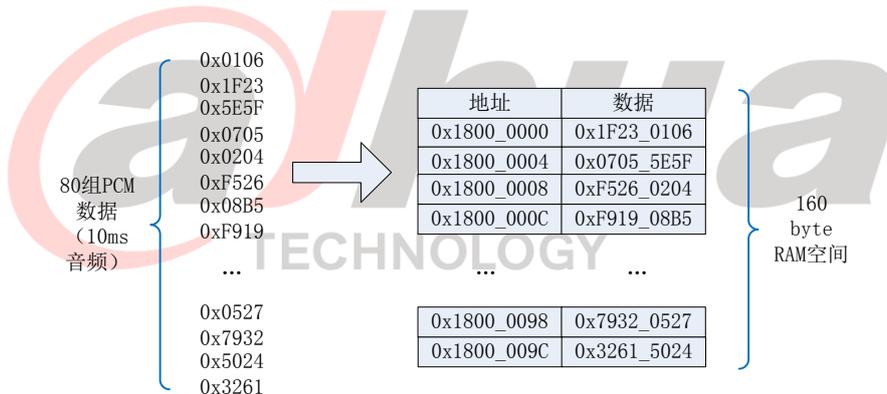
## 7.1.2 工作方式

### 7.1.2.1 数字音频发送

如下图所示，为数字模式下的发送音频数据链路图，



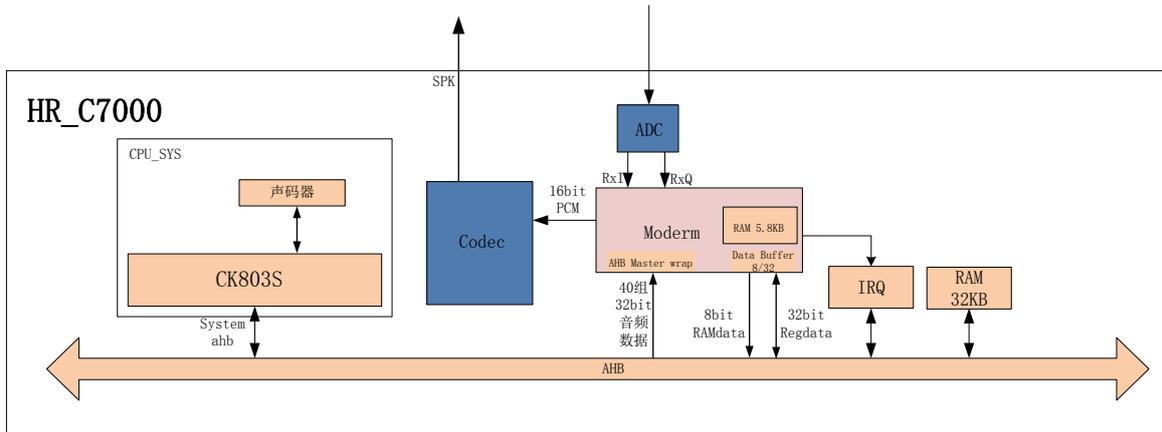
音频 MIC/LINE IN 信号经过内置 codec ADC 实现模数转换，输出 16bit 位宽的 PCM 音频码流，该数据流通过相邻两组 16bit 数据两两组合，构成 32bit 数据（先产生的 16bit 数据存放在 32bit 中低位，后产生的存放于高位）。每完成 40 组 32bit 数据的组合，则缓存至内置 RAM 0x1800\_0000~1800\_009F 区域，并输出中断 PCM\_wr\_data\_interp（irq28，详见 3.6.3 中断向量描述）。



CPU 检测到第 28 号内部中断源时，将上述 RAM 中的数据输出到声码功能区进行编码，并清除该中断（向 Modem 寄存器 0x1100\_03B0 的 bit4 写“1”），编码结果输出到 TX RAM 的 0x1600\_0960~1600\_097A 区间。

### 7.1.2.2 数字音频接收

如下图所示，为数字模式下的接收音频播放过程，



数字接收获取到的 27byte 数据通过 AHB 读取(图中 8bit RAMdata 通道, 地址 **0x1600\_0960~1600\_097A**, 该过程可参考 AHB 读取 RX RAM 的过程)并输出到 CPU 内置声码功能区。

经过解码后, 得到长度为 10ms 的音频数据 (160byte), 该数据暂存于地址区间为 **0x1800\_00A0~1800\_0140** 的字段。内置 CODEC 通过 AHB 总线从该区域获取音频数据, 并以 10ms 为间隔, 周期性的产生数据读取完成标志的中断 (**irq27**)。CPU 检测到第 27 号内部中断源时, 再次将新的内置声码器解码后的数据写向 **0x1800\_00A0~1800\_0140**, 并清除该中断 (向 Modem 寄存器 **0x1100\_03B0** 的 bit5 写 “1”)。

### 7.1.2.3 数字通话录音功能

#### 7.1.2.3.1 数字发送录音

数字发送过程中, CPU 可以选择保存编码前的音频数据 (**0x1800\_0000~1800\_009F**), 也可以保存编码后的音频数据, 即 CPU 写向 **0x1600\_0960~1600\_097A** 的数据。

#### 7.1.2.3.2 数字接收录音

数字接收过程中, CPU 可以选择保存解码前的音频数据, 即 CPU 从 **0x1600\_0960~1600\_097A** 读取获得的数据, 也可以保存解码后的音频数据 (**0x1800\_00A0~1800\_0140**)。

## 7.2 FM 模式

### 7.2.1 概述

- 支持模拟音频数据的发送与接收;
- 支持中继模式下, 音频数据的中转发送;
- 支持模拟通话录音保存

### 7.2.2 工作方式

#### 7.2.2.1 模拟音频发送

模拟发送音频有两种数据源: CODEC ADC 输出的 PCM 码流以及保存于 TX RAM 中的模拟发送音频数据。

### 7.2.2.1.1 音频源为 CODEC ADC

音频 PCM 码流到 FM 的发送通路，其数据流无需 CPU 搬移，只需要配置 HR\_C7000 工作于 FM 模式，在 PTT 发送有效的时间内，同时保证 CODEC ADC 使能已经开启即可。

Modem 寄存器: 0x1100\_0080[0] = 0 ----- 选择音频源为 CODEC ADC  
 0x1100\_0100[7] = 1 ----- 选择 DH4570 工作于 FM 模式  
 0x1100\_0560[0] ----- PTT 发送使能，高电平有效  
 CODEC 寄存器: 0x1600\_09C9[5] = 0 ----- CODEC ADC 工作于 MASTER 模式  
 0x1600\_09C9[4] = 0 ----- CODEC ADC 功能开启  
 0x1600\_09C9[1:0] = 00 ----- CODEC ADC 输出并行 PCM 码流

### 7.2.2.1.2 音频源为 TX RAM

Modem 寄存器: 0x1100\_0080[0] = 1 ----- 选择音频源为 TX RAM 中的数据。

这一方式通常被应用于中继模式下，基站转发音频数据源为 TX RAM 的 0x1600\_0030~0x1600\_042F 区间内，等分为 2 片数据空间（0x1600\_0030~0x1600\_022F 和 0x1600\_0230~0x1600\_042F）。以 AHB 或者 SPI 的方式进行乒乓访问，每片区域 512 byte，可保存 256 个 16bit 音频数据（32ms 的音频播放长度）。

### 7.2.2.2 模拟音频接收

FM 接收解调输出的音频数据到内置 CODEC DAC 端口，其数据流无需 CPU 搬移，只需要配置 DH4570 工作于 FM 模式，同时保证 CODEC DAC 使能以及开启即可。

Modem 寄存器: 0x1100\_0080[0] = 0 ----- 选择将音频通过 CODEC DAC 进行播放  
 0x1100\_0100[7] = 1 ----- 选择 DH4570 工作于模拟模式  
 CODEC 寄存器: 0x1600\_09C8[5] = 0 ----- CODEC DAC 工作于 MASTER 模式  
 0x1600\_09C8[4] = 0 ----- CODEC DAC 功能开启  
 0x1600\_09C8[1:0] = 00 ----- CODEC DAC 输入并行 PCM 码流

### 7.2.2.3 模拟通话录音功能

#### 7.2.2.3.1 模拟发送录音

配置 Modem 寄存器: 0x1100\_0080[0] = 0 ----- 选择音频源为 CODEC ADC  
 0x1100\_0100[7] = 1 ----- 选择 DH4570 工作于 FM 模式

此时模拟发送录音功能默认就是开启的。CODEC ADC 输出的音频数据给到 FM 发送模块的同时，也被保存到 TX RAM 中的 0x1600\_0030~0x1600\_042F 区间。

CPU 访问该区域并对数据进行搬运保存。

#### 7.2.2.3.2 模拟接收录音

配置 Modem 寄存器: 0x1100\_0100[7] = 1 ----- 选择 DH4570 工作于 FM 模式

此时模拟接收录音功能默认开启。FM 接收解调产生的音频信号在输出到 CODEC DAC 的同时，也被保存到 RX RAM 中的 0x1600\_04E0~0x1600\_08DF 区间。

CPU 访问该区域并对数据进行搬运保存。

#### 7.2.2.4 模拟侧音播放

配置 Modem 寄存器：0x1100\_0080[3] = 1 ----- 开启侧音播放功能。

主要应用于模拟模式下的 DTMF、SelCall 等场景下，将发送的信令信号在本地进行语音播放（此时需要将相应的 CODEC DAC 使能开启）。

### 7.3 提示音通路

#### 7.3.1 概述

AHB 总线支持提示音数据的写入，有两种方式可供选择：基于查询方式的写入以及基于中断方式的写入。

#### 7.3.2 功能描述

##### 7.3.2.1 基于查询方式的提示音写入

AHB 写向 0x1600\_0980~1600\_09BF 区间，地址无间隔循环累加。

通过查询基带寄存器 0x1100\_0080 的 bit31，判断 RAM 区间空满状态（“1”为满，“0”为空）。这种方式的缺点是需要 CPU 不断查询。

##### 7.3.2.2 基于中断方式的提示音写入

将提示音数据写向 0x1800\_00A0~1800\_0140，共计 80 个 16bit 音频信号(10ms)。CODEC DAC 将会以 8KHz 速率自动取走这些数据，并产生内部中断。当 CPU 检测到第 27 号内部中断源 **irq27**，则再次向 0x1800\_00A0~1800\_0140 写入新的数据。

这种方式的较查询方式，在使用上更为便利，且 CPU 无需守候。

## 8 射频接口

### 8.1 DAC 发送

#### 8.1.1 概述

HR\_C7000 内置两个高性能 DAC，采用单端输出，支持两点调制和单点调制射频接口，两路信号幅度、偏置分别可调。

用户可以通过配置寄存器选择相应的发送接口，两路输出信号偏置和两路输出信号的幅度。

此外，为了控制芯片的功耗，用户可以通过设置寄存器在 DAC 不工作时将其关闭。

#### 8.1.2 功能描述

表 42 HR\_C7000 基带发送控制寄存器配置

偏移地址	名称	描述	其他
0x70	DAC_CONTROL	DAC 控制寄存器	
0x104	RF_MODE	发送 RF 模式设置	
0x108	SIG_CENTER	发送 IQ 两路信号偏置设置或者两点调制信号偏置设置	
0x10c	TX_IF_FREQ	发送中频频率字配置	
0x110	RF_CONTROL	RF 发送提前量、发送中断提前量设置	
0x114	RF_MOD_BIAS_CTRL	[7:0]: Mod1 发送信号幅度大小配置	

		[15:8]: Mod2 发送信号幅度大小配置; 两点调制偏置	
		[25:16]: Mod1 控制电平大小配置, 用于 外部 1846s 芯片晶振稳定性校正	

### 8.1.3 工作方式

#### 8.1.3.1 两点调制

通过设置寄存器 RF\_MODE[6:5]为 0x3, 使 HR\_C7000 工作于两点调制模式, 可以通过配置 SIG\_CENTER 寄存器分别调节两点调制信号的偏置值。如下图所示, 两点调制信号经过两个运放调节其信号偏置及信号幅度得到 MOD1、MOD2 两路信号分别控制晶振及 VCO, 实现两点调制, 其中两路 APC 由 HR\_C7000 自带的 DAC\_MCU 输出模拟信号, 可用于调节电压幅度。

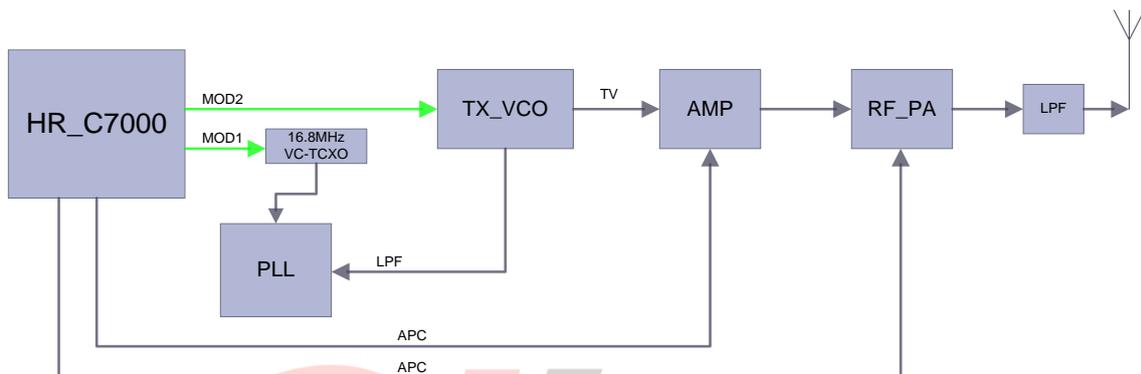


图 62 两点调制接口信号

两点调制模式下, 发送随机信号时差分输出幅度最大约为 2240mV。

通过配置寄存器 RF\_MODE[7]可以调整调制频偏映射关系, 配置 RF\_MODE[7]=0, 对应符号与调制频偏的关系为:

符号	调制频偏
+3	1944 Hz
+1	648 Hz
-1	-648 Hz
-3	-1944 Hz

配置 RF\_MODE[7]=1, 对应符号与调制频偏的关系为:

符号	调制频偏
+3	-1944 Hz
+1	-648 Hz
-1	648 Hz
-3	1944 Hz

通过设置寄存器 SIG\_CENTER[31:24]可调整输出 MOD1 路偏置, 调整范围约为  $\pm 422\text{mV}$ , 最小调整步径为 3.3mV。

通过设置寄存器 SIG\_CENTER[23:16]可调整输出 MOD2 路偏置, 调整范围约为  $\pm 422\text{mV}$ , 最小调整步径为 3.3mV。

通过设置寄存器 RF\_MODE[9], 可选择发送平缓上升起点, 配置 RF\_MODE[9]=1'b0, 则平缓上升起点为 1.65V, 若配置 RF\_MODE[9]=1'b1, 则平缓上升起点为 0V。

通过设置寄存器 RF\_MODE[10]=1'b1, 可发送 40Hz 方波, 用于两点调制调试。

通过设置寄存器 RF\_CONTROL [13:8], 可以配置射频控制中断 RF\_TX\_INTER 相对 30ms 时隙边界提前量, 可调范围为 0 $\mu$ s~6300 $\mu$ s, 最小调整步径为 100 $\mu$ s。

通过设置寄存器 RF\_MOD\_BIAS\_CTR[7:0]可调整输出 MOD1 路幅度, 可调范围约为 8.75mV~2240mV, 最小调整步径为 8.75mV。

通过设置寄存器 RF\_MOD\_BIAS\_CTR[15:8]可调整输出 MOD2 路幅度, 可调范围约为 8.75mV~2240mV, 最小调整步径为 8.75mV。

当发送采用两点调试模式, 如果接收射频通道需要 HR\_C7000 输出直流电压用于控制晶振的电压, 则需要配置 DAC\_CONTROL 寄存器使该路 DAC 处于常开状态。此时, 可以通过配置寄存器 RF\_MOD\_BIAS\_CTR[25:16], 设置 MOD1 在接收状态下的输出电压值, 调节范围为 0~3.3V。

### 8.1.3.2 单点调制

通过设置寄存器 RF\_MODE[6:5]为 0x3, 使 HR\_C7000 工作于两点调制模式, 在两点模式下如果仅仅使用 HR\_C7000 基带 DAC 输出的 1 路 Mod1 信号进行发送信号调制, 则为单点调制方式。可以通过配置 SIG\_CENTER[31:24]寄存器调节单点调制信号的偏置值。如下图所示, 单点调制信号经过 RF 专用芯片 RDA1846S 后, 在经过 PA 放大后低通发射。APC 由 HR\_C7000 自带的 DAC\_MCU 输出模拟信号, 可用于调节电压幅度。

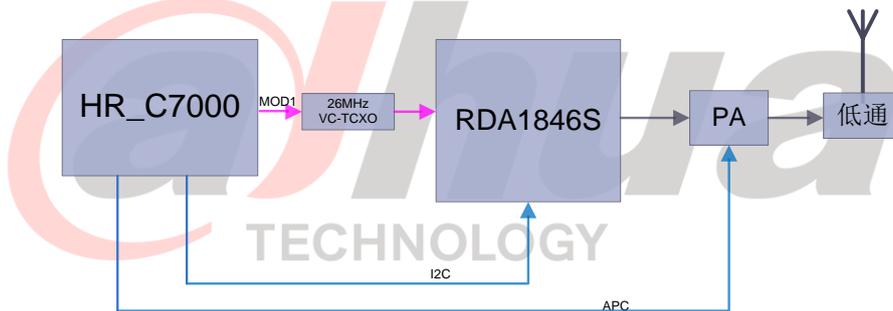


图 63 单点调制接口信号

## 8.2 ADC 接收

### 8.2.1 概述

HR\_C7000 内置两个高性能 ADC, 支持 AF 和中频等射频接口, 并支持两路幅度和偏置分别可调。

通过寄存器配置控制两路 ADC 信号满量程输入时的电压。

此外, 为了控制芯片的功耗, 可以通过配置选择是由 MCU 自动根据接收时隙将相应的 ADC 在发送时隙置为休眠模式, 或者由 MCU 控制相应 ADC 的工作状态。

### 8.2.2 功能描述

表 43 HR\_C7000 基带接收控制寄存器配置

偏移地址	名称	描述	其他
0x74	ADC_CONTROL	ADC 控制寄存器	

0x104	RF_MODE	接收 RF 模式设置	
0x108	SIG_CENTER	接收 IQ 两路偏置设置	
0x1b0	RX_IF_FREQ	接收中频频率字配置	
0x120	THRESHOLD_VALUE	配置到达检测模块检测门限、定时同步模块检测门限	
0x128	PHASE_OUT	AF 接收模式下的相位幅度（含增益）的区间。分别由最大值、最小值表示。	

### 8.2.3 工作方式

#### 8.2.3.1 AF 接收

通过设置 RF\_MODE[24]=1'b1，可使 HR\_C7000 工作于 AF 模式。

使用 AF 模式的接收框图如下所示。

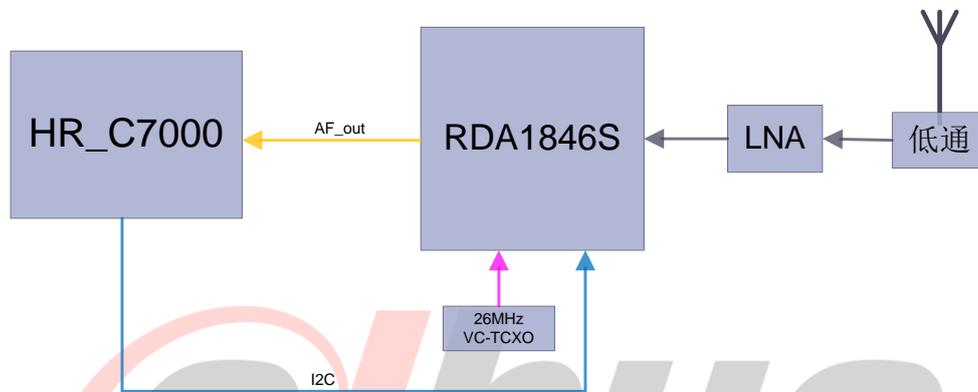


图 64 AF 接收参考接口电路

AF 接收模式，进入 HR\_C7000 信号为调频后的信号，需要在芯片内部调制 AF 信号的增益和偏置。

其中 AF 信号增益由寄存器 RF\_MODE[23:16]进行调节。0~255 分别表示 0~15.9375 倍幅度增益，即单位步进为 1/16 倍。通过实时读取寄存器 PHASE\_OUT 获取当前增益配置下的 AF 相位幅度区间（对于标准的信号强度为-47dBm，调制频偏 2.749KHz 的正负 3 信号，AF 相位幅度区间在-1300~1300 附近），从而实现 AF 幅度粗调。为进一步提升数字接收性能，AF 的增益微调，可以借助于接收误码率测试同步进行，以此获取 RF\_MODE[23:16]与当前外围硬件设计相匹配的经验配置值。

AF 信号偏置则 CPU 总线上的 DAC 提供一个偏置一点，配置 AF 接收差分 ADC 的 N 端进行调节，AF 信号端连接到 ADC 的 P 端。

除上述幅度及偏置的调节外，建议开启寄存器 AF\_MODE[25]，“1”有效，可以有效避免射频强信号下，高频信号的包络对 AF 相位信号的干扰。

#### 8.2.3.2 中频接收

通过设置 RF\_MODE[14:13]=2'b00，可使 HR\_C7000 工作于中频模式，通过配置 SIG\_CENTER 寄存器可对接收中频信号的偏置进行调节，同时可通过配置 RX\_IF\_FREQ 对 HR\_C7000 接收的中频频率进行设置。设置附表寄存器 THRESHOLD\_VALUE，可设定定时同步模块检测门限、到达检测模块检测门限。

使用中频模式的接收框图如下所示。

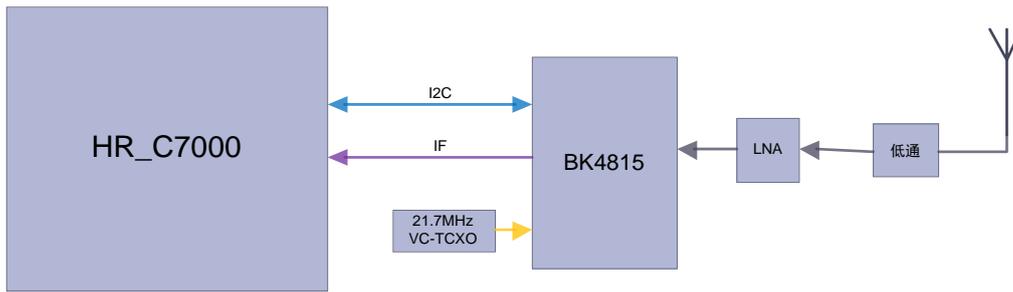


图 65 中频接收参考接口电路

接收信号经滤波、放大后送入 BK4815 专用芯片解调得到的中频 IF 信号直接送入 HR\_C7000 的 AD 管脚 ADC\_IVINP，而 ADC 的负端 ADC\_IVINN 与 VCM 一并连接到接收中频信号的偏置电压。需要注意的是，低中频滤波器主要用于滤除邻道干扰信号，当信道间隔为 25KHz 时，滤波器带宽可选  $\pm 7.5\text{KHz}$ ，当信道间隔为 12.5KHz 时，滤波器带宽可选  $\pm 3.75\text{KHz}$  或  $\pm 4.5\text{KHz}$ 。

## 8.3 数字接收

### 8.3.1 概述

HR\_C7000 支持数字接口接收模式。

### 8.3.2 功能描述

数字接口采用 SPI 单向接收接口，接口示意图如下图所示。

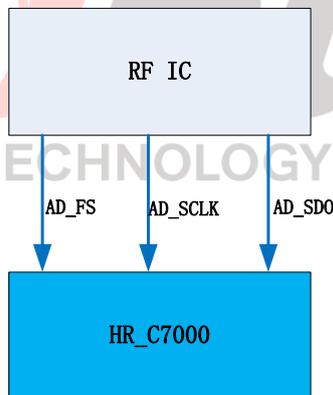


图 66 HR\_C7000 数字接口示意图

### 8.3.3 工作方式

接收如上图，其中 AD\_FS 为 ADC 采样时钟，频率为 38.4KHz；AD\_SCLK 为 ADC 串行数据时钟，频率为 2.4576MHz；AD\_SDO 为 ADC 串行输出数据。

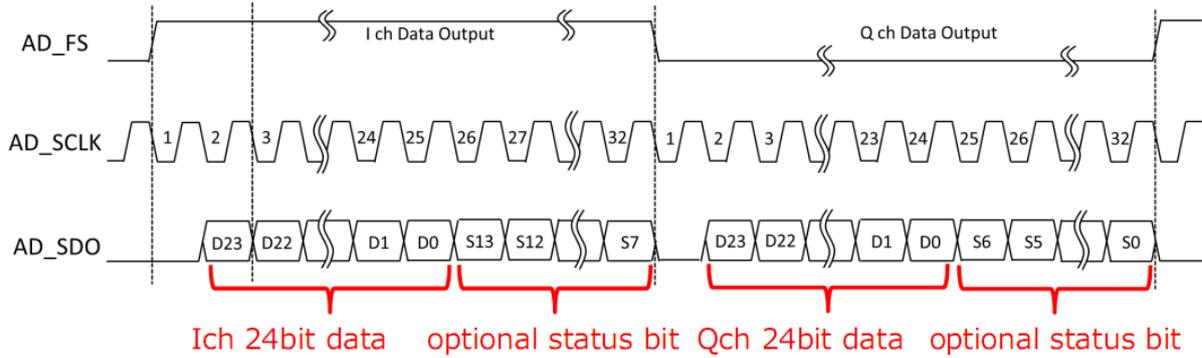


图 67 数字 SPI 接口时序图

详细接口时序如上图。

AD\_FS 用于控制 ADC 的 IQ 两路通道输入，AD\_SDO 最高支持 24bit 精度输入，但是由于 HR\_C7000 接收通路支持最高 16bit 接收精度，可以支持选择 AD\_SDO 串行数据的 24bit 中的任意 16bit 数据作为 ADC 采样数据输入。

AD\_SDO 中的可选状态数据 S13-S0 包括 RSSI 值，AGC 状态值和其他信息，通过 SPI 输入到 HR\_C7000，可以通过内部相关寄存器获取到这些信息。

## 9 DMR 物理层应用

### 9.1 概述

HR\_C7000 芯片基带 DMR 数字部分采用灵活的分层设计模型，针对不同的用户需求，灵活开放不同的层次供用户使用。

分层设计采用三层构架，如图所示。

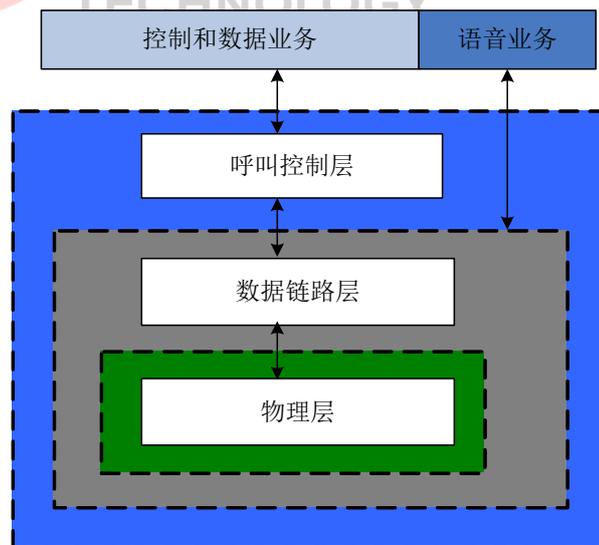


图 68 DMR 数字基带物理层结构

物理层（一层）模式主要解决基带或低中频信号的通道滤波和信号的调制解调过程。用户使

用物理层模式需要自行解决信道编解码以及所有通信协议栈的处理，具有最大的开发灵活度和开发工作量，物理层模式实际主要应用于收发性能指标测试。此外，物理层的一些基本参数将同样应用于其它工作模式。

## 9.2 寄存器

### 9.2.1 寄存器概览

寄存器表 27 物理层寄存器概述（基地址：0x1100\_0000）

偏移地址	名称	描述	页码
0x100	WORK_MODE	基带工作模式设置，本文设置为数字一层模式	
0x104	RF_MODE	收发 RF 模式设置	
0x108	SIG_CENTER	发送、接收 IQ 两路偏置设置	
0x10c	TX_IF_FREQ	发送中频频率字配置	
0x110	RF_CONTROL	RF 收发中断提前量设置	
0x114	RF_MOD_BIAS_CTRL	发送信号幅度大小调节	
0x118	DEV_LIMITER	发送频偏门限值	
0x11c	RSSI_BIAS_VALUE	信号能量检测计算值和 ADC 直流偏置检测	
0x120	THRESHOLD_VALUE	信号同步检测门限设置	
0x128	PHASE_OUT	AF 接收模式下的相位输入幅度检测	
0x12c	SEND_DATA_SYNC_H	发送数据同步帧字段	
0x130	SEND_DATA_SYNC_L	发送数据同步帧字段	
0x134	SEND_RC_SYNC_H	发送 RC 同步帧字段	
0x138	SEND_RC_SYNC_L	发送 RC 同步帧字段	
0x13c	RECV_MS_SYNC_H	接收 MS 数据同步帧字段	
0x140	RECV_MS_SYNC_L	接收 MS 数据同步帧字段	
0x144	RECV_BS_SYNC_H	接收 BS 数据同步帧字段	
0x148	RECV_BS_SYNC_L	接收 BS 数据同步帧字段	
0x14c	RECV_TDMA1_SYNC_H	接收 TDMA1 数据同步帧字段	
0x150	RECV_TDMA1_SYNC_L	接收 TDMA1 数据同步帧字段	
0x154	RECV_TDMA2_SYNC_H	接收 TDMA2 数据同步帧字段	
0x158	RECV_TDMA2_SYNC_L	接收 TDMA2 数据同步帧字段	
0x15c	RECV_RC_SYNC_H	接收 RC 数据同步帧字段	
0x160	RECV_RC_SYNC_L	接收 RC 数据同步帧字段	
0x164	RECV_INFO_COUNT	接收计算 FSKErrror	
0x16c	RECVFIR_COEF1	接收 FIR 系数	
0x170	RECVFIR_COEF2	接收 FIR 系数	
0x174	RECVFIR_COEF3	接收 FIR 系数	
0x178	RECVFIR_COEF4	接收 FIR 系数	
0x17c	RECVFIR_COEF5	接收 FIR 系数	
0x180	RECVFIR_COEF6	接收 FIR 系数	
0x184	RECVFIR_COEF7	接收 FIR 系数	
0x188	RECVFIR_COEF8	接收 FIR 系数	
0x18c	RECVFIR_COEF9	接收 FIR 系数	
0x190	RECVFIR_COEF10	接收 FIR 系数	

0x194	RECVFIR_COEF11	接收 FIR 系数	
0x198	RECVFIR_COEF12	接收 FIR 系数	
0x19c	RECVFIR_COEF13	接收 FIR 系数	
0x1a0	RECVFIR_COEF14	接收 FIR 系数	
0x1a4	RECVFIR_COEF15	接收 FIR 系数	
0x1a8	RECVFIR_COEF16	接收 FIR 系数	
0x1ac	RECVFIR_COEF17	接收 FIR 系数	
0x1b0	RX_IF_FREQ	接收中频频率字配置	
0x1b4	RF_RECV_DELAY	接收通道延时量抵消设置	
0x1b8	MODEM_AK_CLK_CALB	AK2401 时钟调节控制寄存器	
0x1bc	AK2401_ADC_STATE	AK2401ADC 接收状态寄存器	
0x1c0	PHY_RECEIVE_STATE	物理层接收状态信息	
0x400	LAYER2_CONTROL	物理层收发模式控制，与二层复用	
0x408	LAYER2_TXRX_CTRL	物理层收发控制，与二层复用	

## 9.2.2 寄存器描述

### 9.2.2.1 WORK\_MODE

偏移地址：0x100

复位方式：H

Bits	Access	Name	Default	Description
[31:9]	-	reserved		
[8]	RW	digital_analog_en	0x0	数字模拟同时检测开关，要求在默认配置数字模式下开启，在接收到数字信号或者模拟信号后，对应切换到数字模式或模拟模式。 1：表示数模同时接收开启控制使能； 0：表示只开启数字或者模拟接收，根据Bit7设置。
[7]	RW	modulator_mode	0x0	工作模式设置： 0：表示DMR数字模式； 1：表示FM模拟模式。
[6]	RW	is_tier1_mode	0x1	0：表示TierI模式； 1：表示TierII模式。
[5]	RW	is_continuemode	0x1	0：表示Continue模式； 1：表示TimeSlot模式。
[4:3]	RW	layermode	0x1	层次模式设置： 0：表示物理层模式； 1：表示二层模式； 2：表示三层模式。
[2]	RW	is_repeater	0x0	0：表示非中继模式；

				1: 表示中继模式。
[1]	RW	is_aligned	0x1	0: 表示偏移模式（非中继模式下的偏移表示单频模式）； 1: 表示对齐模式。
[0]	RW	work_slot_sel	0x1	三层非中继模式必须设置成1； 0: 表示时隙1； 1: 表示时隙2。

### 9.2.2.2 RF\_MODE

偏移地址：0x104

复位方式：H

Bits	Access	Name	Default	Description
[31]	RW	reserved	0x0	物理层接收测试模式下要求配置成1，正常模式设置为0
[30]	RW	reserved	0x0	
[29]	RW	reserved	0x0	物理层接收测试模式下要求配置成1，正常模式设置为0
[28:26]	-	reserved		
[25]	RW	phase_avg_en	0x0	AF模式均值计算开关，高有效
[24]	RW	af_mode	0x0	AF模式设置： 0: AF信号接收禁能；1: AF信号接收使能。
[23:16]	RW	af_recv_ctrl	0x0	AF接收模式ADC输入信号直接做phase信号的幅度控制
[15]	RW	rf_recviq_mode	0x0	PHY解调接收与实际RF接收的映射关系（以实际RF接收方案确认）： 0: 表示不变； 1: 表示相反。
[14:13]	RW	rf_recv_mode	0x0	RF接收模式： 0x0: 表示中频接收模式； 0x1: 表示中频IQ接收模式； 0x2: 表示基带IQ接收模式。
[12]	RW	adjustsyncmax	0x1	控制调整同步头检测门限使能： 0: 表示固定同步检测门限设置值； 1: 表示自动动态同步检测门限。 备注：LOCAL_CC寄存器的cc_opt设置为1时要求设置为0。
[11]	RW	use_dual_phy	0x0	分时隙开启两个接收机通道控制开关： 0: 关闭双接收通道； 1: 开启双接收通道。（应用于中继芯片接收模式、DCDM模式）
[10]	RW	two_point_ctr	0x0	两点调制测试使能：

				0: 表示关闭; 1: 表示输出40Hz正弦波。
[9]	RW	up_ctr	0x0	时隙边界信号强度平滑上升和下降使能: 0: 表示关闭; 1: 表示开启。
[8]	RW	iq_just_debug	0x0	IQ两路平衡调试测试使能: 0: 表示不变; 1: 表示加偏置值。
[7]	RW	rf_transiq_mode	0x1	PHY调制发送与实际RF发送的映射关系(以实际RF发送方案确认): 0: 表示不变; 1: 表示相反。
[6:5]	RW	rf_trans_mode	0x3	RF发送模式: 0x0: 表示中频发送模式; 0x1: 表示中频IQ发送模式; 0x2: 表示基带IQ发送模式; 0x3: 表示两点调制发送模式。
[4]	RW	ak2401_mode	0x0	AK2401模式设置: 0: AK2401模式关; 1: AK2401模式开。
[3:0]	RW	ak_adc_shift	0x0	AK2401模式下输入ADC信号移位控制寄存器

### 9.2.2.3 SIG\_CENTER

偏移地址: 0x108

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	RW	trans_i_sig_center	0x0	发送I路偏置值设置0~255
[23:16]	RW	trans_q_sig_center	0x0	发送Q路偏置值设置0~255
[15:8]	RW	recv_i_sig_center	0x0	接收I路偏置值设置0~255
[7:0]	RW	recv_q_sig_center	0x0	接收Q路偏置值设置0~255

### 9.2.2.4 TX\_IF\_FREQ

偏移地址: 0x10c

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	if_freq_tx	0x0bb800	发送中频频率字配置

### 9.2.2.5 RF\_CONTROL

偏移地址：0x110

复位方式：H

Bits	Access	Name	Default	Description
[31:21]	-	reserved		
[20:16]	RW	tx_pre_on	0x4	发送提前量配置值，步进100us
[15:14]	-	reserved		
[13:8]	RW	rf_pre_on_tx	0xa	RF发送中断提前量设置，步进100us
[7:6]	-	reserved		
[5:0]	RW	rf_pre_on_rx	0xa	RF接收中断提前量设置，步进100us

### 9.2.2.6 RF\_MOD\_BIAS\_CTRL

偏移地址：0x114

复位方式：H

Bits	Access	Name	Default	Description
[31:26]	-	reserved		
[25:16]	RW	two_point_bias	0x0	
[15:8]	RW	sig_reduce	0x0	IQ模式下调整IQ路幅度大小。 两点模式下调整Mod2幅度大小。 0保持不变，1~255从小到大变化
[7:0]	RW	phase_reduce	0x0	两点模式下调整Mod1幅度大小。 0保持不变，1~255从小到大变化

### 9.2.2.7 DEV\_LIMITER

偏移地址：0x118

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	-	reserved		
[15:0]	RW	dev_limiter	0xff00	限制发送频偏门限值

### 9.2.2.8 RSSI\_BIAS\_VALUE

偏移地址：0x11c

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	RO	rss_i_value	0x0	信号能量检测计算值
[15:8]	RO	adc_bias_i	0x0	ADC的I路输入直流偏置检测
[7:0]	RO	adc_bias_q	0x0	ADC的Q路输入直流偏置检测

### 9.2.2.9 THRESHOLD\_VALUE

偏移地址：0x120

复位方式: H

Bits	Access	Name	Default	Description
[31:29]	-	reserved		
[28:24]	RW	sync_real_limit	0x9	信号同步检测比值门限, 当同步检测值满足同步头检测低门限, 但不满足同步头检测高门限时, 需要对该比值进行判决
[23:16]	RW	maxvalue_low_limit	0x64	信号同步头检测低门限
[15:8]	RW	maxvalue	0x78	信号同步头检测高门限
[7:0]	RW	sig_cor_value	0x67	信号到达检测阈值

#### 9.2.2.10 PHASE\_OUT

偏移地址: 0x128

复位方式: H

Bits	Access	Name	Default	Description
[31:30]	-	reserved		
[29:16]	RO	phase_max_out	0x0	AF接收模式下的相位输入幅度检测时, 14bits相位信号最大值
[15:14]	-	reserved		
[13:0]	RO	phase_min_out	0x0	AF接收模式下的相位输入幅度检测时, 14bits相位信号最小值

#### 9.2.2.11 SEND\_DATA\_SYNC\_H

偏移地址: 0x12c

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	send_data_sync1	0xd5	发送数据同步帧字段[47:40]
[15:8]	RW	send_data_sync2	0xd7	发送数据同步帧字段[39:32]
[7:0]	RW	send_data_sync3	0xf7	发送数据同步帧字段[31:24]

#### 9.2.2.12 SEND\_DATA\_SYNC\_L

偏移地址: 0x130

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	send_data_sync4	0x7f	发送数据同步帧字段[23:16]
[15:8]	RW	send_data_sync5	0xd7	发送数据同步帧字段[15:8]
[7:0]	RW	send_data_sync6	0x57	发送数据同步帧字段[7:0]

### 9.2.2.13 SEND\_RC\_SYNC\_H

偏移地址：0x134

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	send_rc_sync1	0x77	发送RC同步帧字段[47:40]
[15:8]	RW	send_rc_sync2	0xd5	发送RC同步帧字段[39:32]
[7:0]	RW	send_rc_sync3	0x5f	发送RC同步帧字段[31:24]

### 9.2.2.14 SEND\_RC\_SYNC\_L

偏移地址：0x138

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	send_data_sync4	0x7f	发送数据同步帧字段[23:16]
[15:8]	RW	send_data_sync5	0xd7	发送数据同步帧字段[15:8]
[7:0]	RW	send_data_sync6	0x57	发送数据同步帧字段[7:0]

### 9.2.2.15 RECV\_MS\_SYNC\_H

偏移地址：0x13c

复位方式：H

Bits	Access	Name	Default	Description
[31]	RW	recv_ms_sync_en	0x1	接收MS数据同步帧使能开关，高有效（除三层模式以外）
[30:24]	-	reserved		
[23:16]	RW	recv_ms_sync1	0xd5	接收MS数据同步帧字段[47:40]
[15:8]	RW	recv_ms_sync2	0xd7	接收MS数据同步帧字段[39:32]
[7:0]	RW	recv_ms_sync3	0xf7	接收MS数据同步帧字段[31:24]

### 9.2.2.16 RECV\_MS\_SYNC\_L

偏移地址：0x140

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	recv_ms_sync4	0x7f	接收MS数据同步帧字段[23:16]
[15:8]	RW	recv_ms_sync5	0xd7	接收MS数据同步帧字段[15:8]
[7:0]	RW	recv_ms_sync6	0x57	接收MS数据同步帧字段[7:0]

### 9.2.2.17 RECV\_BS\_SYNC\_H

偏移地址：0x144

复位方式: H

Bits	Access	Name	Default	Description
[31]	RW	recv_bs_sync_en	0x1	接收BS数据同步帧使能开关, 高有效(除三层模式以外)
[30:24]	-	reserved		
[23:16]	RW	recv_bs_sync1	0xdf	接收BS数据同步帧字段[47:40]
[15:8]	RW	recv_bs_sync2	0xf5	接收BS数据同步帧字段[39:32]
[7:0]	RW	recv_bs_sync3	0x7d	接收BS数据同步帧字段[31:24]

#### 9.2.2.18 RECV\_BS\_SYNC\_L

偏移地址: 0x148

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	recv_bs_sync4	0x75	接收BS数据同步帧字段[23:16]
[15:8]	RW	recv_bs_sync5	0xdf	接收BS数据同步帧字段[15:8]
[7:0]	RW	recv_bs_sync6	0x5d	接收BS数据同步帧字段[7:0]

#### 9.2.2.19 RECV\_TDMA1\_SYNC\_H

偏移地址: 0x14c

复位方式: H

Bits	Access	Name	Default	Description
[31]	RW	recv_tdma1_sync_en	0x1	接收TDMA1数据同步帧使能开关, 高有效(除三层模式以外)
[30:24]	-	reserved		
[23:16]	RW	recv_tdma1_sync1	0xf7	接收TDMA1数据同步帧字段[47:40]
[15:8]	RW	recv_tdma1_sync2	0xfd	接收TDMA1数据同步帧字段[39:32]
[7:0]	RW	recv_tdma1_sync3	0xd5	接收TDMA1数据同步帧字段[31:24]

#### 9.2.2.20 RECV\_TDMA1\_SYNC\_L

偏移地址: 0x150

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	recv_tdma1_sync4	0xdd	接收TDMA1数据同步帧字段[23:16]
[15:8]	RW	recv_tdma1_sync5	0xfd	接收TDMA1数据同步帧字段[15:8]
[7:0]	RW	recv_tdma1_sync6	0x55	接收TDMA1数据同步帧字段[7:0]

#### 9.2.2.21 RECV\_TDMA2\_SYNC\_H

偏移地址: 0x154

复位方式: H

Bits	Access	Name	Default	Description
[31]	RW	recv_tdma2_sync_en	0x1	接收TDMA2数据同步帧使能开关, 高有效 (除三层模式以外)
[30:24]	-	reserved		
[23:16]	RW	recv_tdma2_sync1	0xd7	接收TDMA2数据同步帧字段[47:40]
[15:8]	RW	recv_tdma2_sync2	0x55	接收TDMA2数据同步帧字段[39:32]
[7:0]	RW	recv_tdma2_sync3	0x7f	接收TDMA2数据同步帧字段[31:24]

#### 9.2.2.22 RECV\_TDMA2\_SYNC\_L

偏移地址: 0x158

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	recv_tdma2_sync4	0x5f	接收TDMA2数据同步帧字段[23:16]
[15:8]	RW	recv_tdma2_sync5	0xf7	接收TDMA2数据同步帧字段[15:8]
[7:0]	RW	recv_tdma2_sync6	0xf5	接收TDMA2数据同步帧字段[7:0]

#### 9.2.2.23 RECV\_RC\_SYNC\_H

偏移地址: 0x15c

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	recv_rc_sync1	0x77	接收RC同步帧字段[47:40]
[15:8]	RW	recv_rc_sync2	0xd5	接收RC同步帧字段[39:32]
[7:0]	RW	recv_rc_sync3	0x5f	接收RC同步帧字段[31:24]

#### 9.2.2.24 RECV\_RC\_SYNC\_L

偏移地址: 0x160

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	recv_rc_sync4	0x7d	接收RC同步帧字段[23:16]
[15:8]	RW	recv_rc_sync5	0xfd	接收RC同步帧字段[15:8]
[7:0]	RW	recv_rc_sync6	0x77	接收RC同步帧字段[7:0]

#### 9.2.2.25 RECV\_INFO\_COUNT

偏移地址: 0x164

复位方式: H

Bits	Access	Name	Default	Description
------	--------	------	---------	-------------

[31:24]	RO	sync_error_num	0x0	每一帧的同步帧类型帧的同步字段误码统计
[23:9]	-	reserved		
[8]	RW	fsk_error_en	0x0	FSKError输出使能开关： 1：表示输出FSKError统计值；0：表示输出EVM统计值。
[7:0]	RO	fsk_error	0x0	统计FSKError值或者EVM统计值

#### 9.2.2.26 RECVFIR\_COEF1

偏移地址：0x16c

复位方式：H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef0	0x003	接收FIR系数1
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef1	0x003	接收FIR系数2

#### 9.2.2.27 RECVFIR\_COEF2

偏移地址：0x170

复位方式：H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef2	0x001	接收FIR系数3
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef3	0xffe	接收FIR系数4

#### 9.2.2.28 RECVFIR\_COEF3

偏移地址：0x174

复位方式：H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef4	0xffb	接收FIR系数5
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef5	0xffb	接收FIR系数6

#### 9.2.2.29 RECVFIR\_COEF4

偏移地址：0x178

复位方式：H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		

[27:16]	RW	recvfir_coef6	0xfff	接收FIR系数7
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef7	0x006	接收FIR系数8

#### 9.2.2.30 RECVFIR\_COEF5

偏移地址：0x17c

复位方式：H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef8	0x00c	接收FIR系数9
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef9	0x009	接收FIR系数10

#### 9.2.2.31 RECVFIR\_COEF6

偏移地址：0x180

复位方式：H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef10	0xffe	接收FIR系数11
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef11	0xfef	接收FIR系数12

#### 9.2.2.32 RECVFIR\_COEF7

偏移地址：0x184

复位方式：H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef12	0xfe9	接收FIR系数13
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef13	0xff4	接收FIR系数14

#### 9.2.2.33 RECVFIR\_COEF8

偏移地址：0x188

复位方式：H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef14	0x00c	接收FIR系数15
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef15	0x024	接收FIR系数16

**9.2.2.34 RECVFIR\_COEF9**

偏移地址: 0x18c

复位方式: H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef16	0x026	接收FIR系数17
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef17	0x00a	接收FIR系数18

**9.2.2.35 RECVFIR\_COEF10**

偏移地址: 0x190

复位方式: H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef18	0xfdc	接收FIR系数19
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef19	0xfbd	接收FIR系数20

**9.2.2.36 RECVFIR\_COEF11**

偏移地址: 0x194

复位方式: H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef20	0xfc9	接收FIR系数21
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef21	0x006	接收FIR系数22

**9.2.2.37 RECVFIR\_COEF12**

偏移地址: 0x198

复位方式: H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef22	0x052	接收FIR系数23
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef23	0x076	接收FIR系数24

**9.2.2.38 RECVFIR\_COEF13**

偏移地址: 0x19c

复位方式: H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		

[31:28]	-	reserved		
[27:16]	RW	recvfir_coef24	0x047	接收FIR系数25
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef25	0xfca	接收FIR系数26

#### 9.2.2.39 RECVFIR\_COEF14

偏移地址: 0x1a0

复位方式: H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef26	0xf47	接收FIR系数27
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef27	0xf23	接收FIR系数28

#### 9.2.2.40 RECVFIR\_COEF15

偏移地址: 0x1a4

复位方式: H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef28	0xfaf	接收FIR系数29
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef29	0x0ed	接收FIR系数30

#### 9.2.2.41 RECVFIR\_COEF16

偏移地址: 0x1a8

复位方式: H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef30	0x281	接收FIR系数31
[15:12]	-	reserved		
[11:0]	RW	recvfir_coef31	0x3d2	接收FIR系数32

#### 9.2.2.42 RECVFIR\_COEF17

偏移地址: 0x1ac

复位方式: H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	recvfir_coef32	0x455	接收FIR系数33
[15:0]	-	reserved		

**9.2.2.43 RX\_IF\_FREQ**

偏移地址: 0x1b0

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	if_freq_rx	0x0bb800	接收中频频率字配置

**9.2.2.44 RF\_RECV\_DELAY**

偏移地址: 0x1b4

复位方式: H

Bits	Access	Name	Default	Description
[31:19]	-	reserved		
[18:0]	RW	rf_recv_delay	0x0	接收通道延时量设置, 步进100ns AF接收模式下, 要求设置2000; 其它方案默认值0。

**9.2.2.45 MODEM\_AK\_CLK\_CALB**

偏移地址: 0x1b4

复位方式: H

Bits	Access	Name	Default	Description
[31:16]	-	reserved		
[15]	RW	clk_inner_calb_en	0x0	AK2401接收模式(异步晶振方式), 基带时钟自适应调整功能开关 0: 关闭, 1: 开启, 默认为关闭
[14]	RW	clk_ext_calb_en	0x0	AK2401接收模式(异步晶振方式), 基带时钟固定配置调整功能开关 0: 关闭, 1: 开启, 默认为关闭
[13:9]	-	reserved		
[8:0]	RW	clk_ext_calb_coef	0x0	基带时钟固定配置调整系数。每间隔clk_ext_calb_coef个2048基带时钟计数周期调整一次en_sample的位置。 满足等式: $(1 \pm (1/2048 * \text{clk\_ext\_calb\_coef})) * \text{en\_sample} = \text{ak\_ad\_fs}$ ; 其中en_sample为基带的38.4KHz采样率, ak_ad_fs为AK外挂的晶振时钟情况下的ADC输出使能

**9.2.2.46 AK2401\_ADC\_STATE**

偏移地址: 0x1bc

复位方式: H

Bits	Access	Name	Default	Description
[31:14]	-	reserved		
[13:8]	RO	ak_rssi	0x0	AK ADC RSSI值
[7:0]	RO	ak_status	0x0	AK ADC接收状态值

#### 9.2.2.47 PHY\_RECEIVE\_STATE

偏移地址：0x1c0

复位方式：H

Bits	Access	Name	Default	Description
[31:0]	RO	phy_receive_state	0x0	物理层接收状态信息

#### 9.2.2.48 LAYER2\_CONTROL

偏移地址：0x400

复位方式：H

Bits	Access	Name	Default	Description
[31:8]	-	reserved		
[7]	RW	txen	0x0	物理层发送同步使能开关，0为关闭，1为开启
[6]	RW	rxen	0x0	物理层接收同步使能开关，0为关闭，1为开启
[5:2]	-	reserved		
[1]	RW	tx_master_mode	0x0	物理层发送时隙建立模式，0被动模式，1主动模式
[0]	RW	rx_master_mode	0x0	物理层接收时隙建立模式，0被动模式，1主动模式，默认要求设置成0。

#### 9.2.2.49 LAYER2\_TXRX\_CTRL

偏移地址：0x408

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23]	RW	syncfail_layer2_rx	0x0	物理层失去接收时隙同步控制开关： 0：维持不变； 1：释放同步时隙，重新建立。
[22:8]	-	reserved		
[7]	RW	txnextsloten	0x0	下一时隙发送使能： 0：即将到来的时隙中断开始不发送； 1：即将到来的时隙中断开始发送。
[6]	RW	rxnextsloten	0x0	下一时隙接收使能： 0：即将到来的时隙中断开始不接收； 1：即将到来的时隙中断开始接收。

[5:1]	-	reserved		
[0]	RW	autotest	0x0	物理层测试使能，高有效

### 9.3 中断向量

DMR 数字物理层模式下，定义中断类型如下表所示。

表 44 物理层中断类型列表

序号	名称	类型	描述
1	RF_TX_INTERP	电平触发，高有效	射频切换中断，RF 发送开始中断，通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
2	RF_RX_INTERP	电平触发，高有效	射频切换中断，RF 接收开始中断，通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
3	SYS_INTERP	电平触发，高有效	基带系统中断，有子中断见下表，通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除
4	TIME_SLOT_TX_INTERP	电平触发，高有效	基带数字 30ms 发送时隙中断，通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
5	TIME_SLOT_RX_INTERP	电平触发，高有效	基带数字 30ms 接收时隙中断，通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除

表 45 物理层系统中断子中断类型列表

序号	名称	类型	描述
1	PHY_RDY_INTERP	电平触发，高有效	物理层处理中断： 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽； 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
2	PHY_RX_INTERP	电平触发，高有效	物理层接收解帧中断： 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽； 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
3	PHY_RECV_ABNORMAL	电平触发，高有效	物理层接收时隙边界异常提示中断： 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽； 通过 SYS_INTERP_CLEA

### 9.4 Memory 映射

基带 Ram 模块是 HR\_C7000 芯片基带数字功能主要的用户数据交互存储空间，根据不同的功能设计主要分为发送和接收 2 个 Ram，数据位宽为 8bit，深度为 1200。

### 9.4.1 物理层发送 Ram

物理层发送 Ram 的基地址为 0x16000000，深度为 1200，下表定义了各个地址段内容分配。

表 46 物理层发送 Ram 分配

偏址	名称	描述
0x00~0x23	SEND_DATA0	物理层发送 1 帧数据，与 SEND_DATA1 交替
0x24~0x47	SEND_DATA1	物理层发送 1 帧数据，与 SEND_DATA0 交替

### 9.4.2 物理层接收 Ram

物理层接收 Ram 的基地址为 0x160004b0，深度为 1200，下表定义了各个地址段内容分配。

表 47 物理接收 Ram 分配

偏址	名称	描述
0x30~0x53	RECV_INFO	时隙模式下，0x30~0x50 为 264bits 解帧信息； 连续模式下，0x30~0x32 为 CACH 24bits 信息，0x33~0x53 为 264bits 解帧信息

## 9.5 物理层发送

### 9.5.1 概述

物理层发送主要应用于发送性能指标测试和配合接收性能指标测试。物理层的发送测试操作相对简单，只需要配置寄存器启动发送以及配置发送的数据内容，发送数据内容是直接发送不经过任何的编码过程。

### 9.5.2 功能描述

首先简单介绍一下发送数据内容，根据 DMR 标准对应 2 比特数据为一个符号，其符号映射关系如下表所示。

Bit	映射的符号
00	+1
01	+3
10	-1
11	-3

因此物理层发送支持+1, -1, +3, -3, +1-1, +3-3 六种最基础的信号测试，例如需要发送+3-3 则配置数据值为 0x5f，每一帧数据 36bytes，分别写入发送数据存储的两块地址。

当然物理层发送必须要支持标准的 DMR 结构帧，如下图所示。

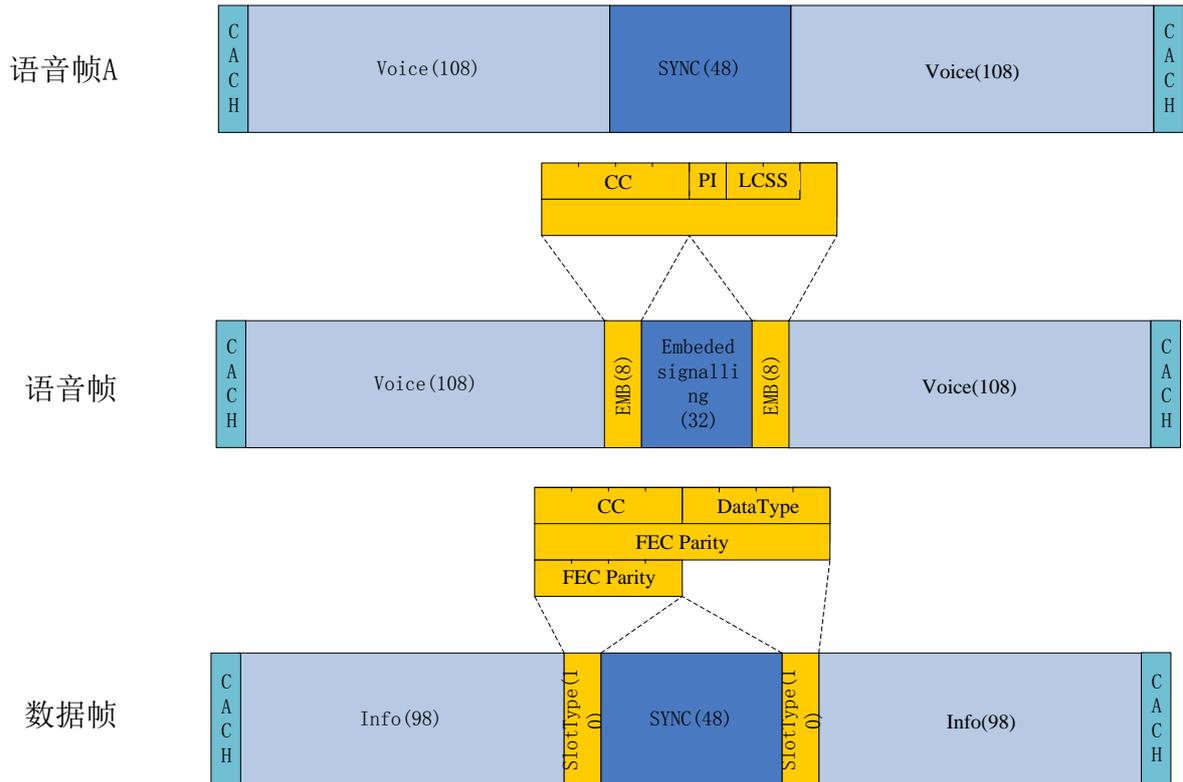


图 69 DMR 标准发送帧结构

### 9.5.3 工作方式

表 48 物理层发送配置表

序号	寄存器/Ram 地址	配置值	描述
1	WORK_MODE	0x22 或 0x02	0x22 表示物理层时隙模式；0x02 表示物理层连续模式
2	RF_MODE	0x60	两点发送中频接收模式
3	LAYER2_CONTROL	0x82	物理层主动发送模式
4	LAYER2_TXRX_CTRL	0x81	物理层主动发送使能
5	0x16000000~0x16000023		发送数据存储地址 1
6	0x16000024~0x16000047		发送数据存储地址 2

通常我们将 SYNC 字段区域填充标准的同步字，将数据部分填写随机数，产生 36bytes 的一帧数据，分别写入发送数据存储的两块地址。

参考误码测试随机数如下，

{0x80,0x0c,0x4F,0xC3,0x26,0xFE,0xBB,0xC1,0xD6,0x26,0x23,0x14,0x6F,0xFD,0x9E,0xD5,0xD7,0xF7,0x7F,0xD7,0x57,0x61,0xE8,0x5C,0x92,0x83,0x94,0xEA,0xFB,0x6B,0x3E,0x59,0xC4,0x0D,0x60,0x01}。

## 9.6 物理层接收

### 9.6.1 概述

物理层接收主要应用于接收的误码率测试。

### 9.6.2 功能描述

- 支持数据控制帧接收
- 支持语音帧接收
- 支持时隙接收和连续接收两种方式

如下图定义了物理层接收解调输出的信息内容，对应从接收 Ram 中可以获取。

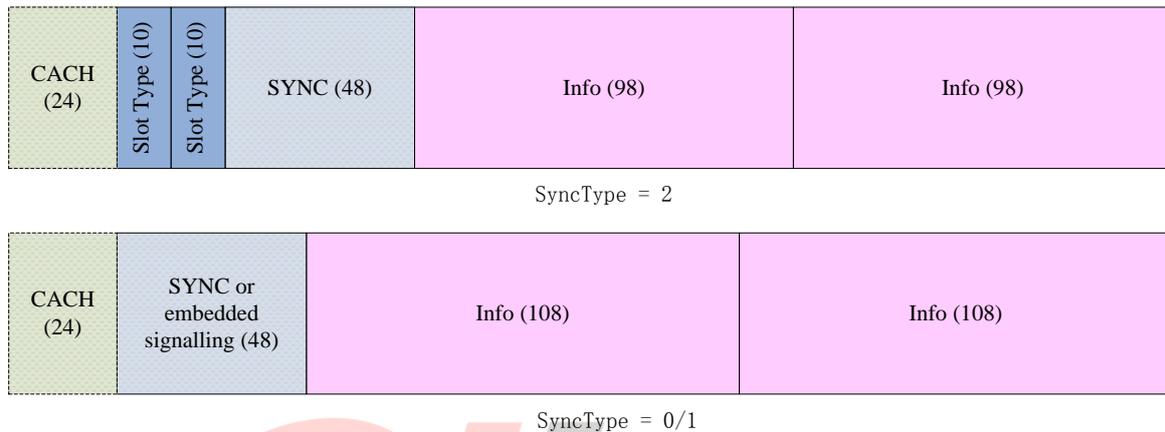


图 70 物理层解析数据顺序

### 9.6.3 工作方式

表 49 物理层接收配置表

序号	寄存器/Ram 地址	配置值	描述
1	WORK_MODE	0x22 或 0x02	0x22 表示物理层时隙模式；0x02 表示物理层连续模式
2	RF_MODE	0xa0000060	两点发送中频接收模式
3	LAYER2_CONTROL	0x40	物理层被动接收模式
4	LAYER2_TXRX_CTRL	0x41	物理层被动接收使能

物理层接收状态流程如下图所示。

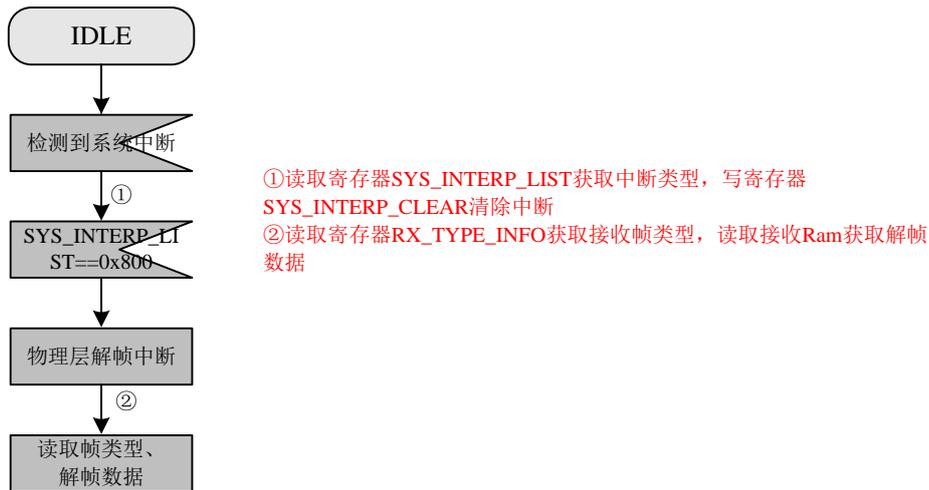


图 71 物理层接收状态流程图

## 10 DMR 链路层应用

### 10.1 概述

HR\_C7000 芯片基带 DMR 数字部分采用灵活的分层设计模型，针对不同的用户需求，灵活开放不同的层次供用户使用。

链路层（二层）模式主要是开放物理层所有内容的基础上，完成信道的编解码以及交织解交织和校验等部分的工作，如下图中灰色虚线框中所定义的功能。用户仅需要解决通信协议栈的处理流程，具有较大的开发灵活度和适度的开发工作量。

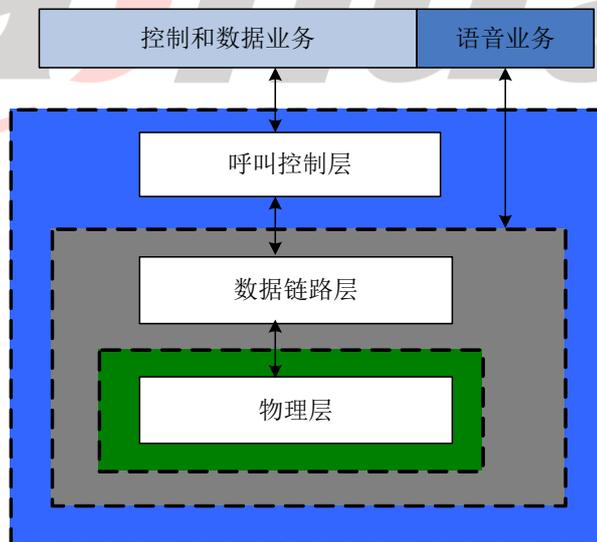


图 72 DMR 数字基带链路层结构

### 10.2 寄存器

#### 10.2.1 寄存器概览

寄存器表 28 链路层寄存器概述（基地址：0x1100\_0000）

偏移地址	名称	描述	页码
0x100	WORK_MODE	基带工作模式设置, 本文设置为数字二层模式	
0x350	PI_CRC_MASK	PI 帧 CRC16 的掩码和初值设置	
0x354	RS_H_MASK	语音帧头的 RS24 码的掩码设置	
0x358	RS_H_INIT	语音帧头的 RS24 码的初值设置	
0x35c	RS_T_MASK	帧尾的 RS24 码的掩码设置	
0x360	RS_T_INIT	帧尾的 RS24 码的初值设置	
0x364	CSBK_CRC_MASK	CSBK 帧的 CRC16 的掩码和初值设置	
0x368	MBCH_CRC_MASK	MBC 帧头的 CRC16 的掩码和初值设置	
0x36c	DATAH_CRC_MASK	数据帧头的 CRC16 的掩码和初值设置	
0x380	CRC8_INIT	ShortLC 帧的 CRC8 的初值设置	
0x384	LOCAL_CC	本机收发 CC 设置	
0x388	VOICE_EMB_CTRL	语音帧 EMB 区域 PI 设置和语音帧 F 的 EMB 填充设置	
0x38c	SCRAMBLE_REG	语音简单加密模式密钥设置	
0x390	RX_TYPE_INFO	接收帧信息描述	
0x39c	RX_INFO	接收 RC 帧信息	
0x400	LAYER2_CONTROL	二层模式控制	
0x404	LAYER2_SLOTON	二层时隙开关控制	
0x408	LAYER2_TXRX_CTRL	二层收发使能控制	
0x40c	LAYER2_SLOT_CNT	二层时隙计数初值设置	
0x410	LAYER2_MAX_CNT	二层时隙最大计数修正值设置	
0x414	LAYER2_SLOT_UPDATE	二层时隙更新设置	
0x418	LAYER2_SEND_TYPE	二层发送帧类型测试	
0x41c	LAYER2_STATUS	二层时隙状态指示	
0x424	LAYER2_SLOT_PRE	二层处理中断位置提前配置	
0x428	LAYER2_BS_MODE	中转台模式设置	
0x430	BS_SECOND_REG	中转台同播模式 GPS 秒时间设置寄存器	
0x434	BS_STATUS	BS 中转台同播模式下状态寄存器	
0x438	BS_CACH	中转台发送 CACH 设置	

## 10.2.2 寄存器描述

### 10.2.2.1 WORK\_MODE

偏移地址: 0x100

复位方式: H

Bits	Access	Name	Default	Description
[31:9]	-	reserved		
[8]	RW	digital_analog_en	0x0	数字模拟同时检测开关, 要求在默认配置数字模式下开启, 在接收到数字信号或者模拟信号后, 对应切换到数

				字模式或模拟模式。 1: 表示数模同时接收开启控制使能; 0: 表示只开启数字或者模拟接收, 根据Bit7设置。
[7]	RW	modulator_mode	0x0	工作模式设置: 0: 表示DMR数字模式; 1: 表示FM模拟模式。
[6]	RW	is_tier1_mode	0x1	0: 表示TierI模式; 1: 表示TierII模式。
[5]	RW	is_continuemode	0x1	0: 表示Continue模式; 1: 表示TimeSlot模式。
[4:3]	RW	layermode	0x1	层次模式设置: 0: 表示物理层模式; 1: 表示二层模式; 2: 表示三层模式。
[2]	RW	is_repeater	0x0	0: 表示非中继模式; 1: 表示中继模式。
[1]	RW	is_aligned	0x1	0: 表示偏移模式(非中继模式下的偏移表示单频模式); 1: 表示对齐模式。
[0]	RW	work_slot_sel	0x1	三层非中继模式必须设置成1; 0: 表示时隙1; 1: 表示时隙2。

#### 10.2.2.2 PI\_CRC\_MASK

偏移地址: 0x350

复位方式: H

Bits	Access	Name	Default	Description
[31:16]	RW	pi_crc16_init_reg	0x0	PI帧CRC16的初始值
[15:0]	RW	pi_crc16_mask	0x6969	PI帧CRC16的掩码

#### 10.2.2.3 RS\_H\_MASK

偏移地址: 0x354

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	rs24_head_mask	0x969696	语音帧头RS24的掩码

#### 10.2.2.4 RS\_H\_INIT

偏移地址: 0x358

复位方式: H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	rs24_head_init_reg	0x0	语音帧头RS24的初值

#### 10.2.2.5 RS\_T\_MASK

偏移地址：0x35c

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	rs24_term_mask	0x969696	语音或者数据帧尾RS24的掩码

#### 10.2.2.6 RS\_T\_INIT

偏移地址：0x360

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	rs24_term_init_reg	0x0	语音或者数据帧尾RS24的初值

#### 10.2.2.7 CSBK\_CRC\_MASK

偏移地址：0x364

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	RW	csbk_crc16_init_reg	0x0	CSBK帧CRC16的初始值
[15:0]	RW	csbk_crc16_mask	0xa5a5	CSBK帧CRC16的掩码

#### 10.2.2.8 MBCH\_CRC\_MASK

偏移地址：0x368

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	RW	mbch_crc16_init_reg	0x0	MBC帧头CRC16的初始值
[15:0]	RW	mbch_crc16_mask	0xa5a5	MBC帧头CRC16的掩码

#### 10.2.2.9 DATAH\_CRC\_MASK

偏移地址：0x36c

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	RW	dath_crc16_init_reg	0x0	数据帧头CRC16的初始值
[15:0]	RW	dath_crc16_mask	0xa5a5	数据帧头CRC16的掩码

### 10.2.2.10 CRC8\_INIT

偏移地址：0x380

复位方式：H

Bits	Access	Name	Default	Description
[31:8]	-	reserved		
[7:0]	RW	crc8_init_reg	0x0	shortlc中CRC8的初值

### 10.2.2.11 LOCAL\_CC

偏移地址：0x384

复位方式：H

Bits	Access	Name	Default	Description
[31:17]	-	reserved		
[16]	RW	encryption_type	0x0	语音加密类型设置： 0：表示普通加密； 1：表示增强加密。
[15:10]	-	reserved		
[9:8]	RW	cc_opt	0x0	接收CC匹配方式： 0：接收CC正确且与本地相等时才进行解帧处理； 1：对任意接收正确的CC都匹配，进行解帧处理； 2、3：对所有接收帧不进行解帧处理。
[7:4]	-	reserved		
[3:0]	RW	cc	0x1	本地CC设置

### 10.2.2.12 VOICE\_EMB\_CTRL

偏移地址：0x388

复位方式：H

Bits	Access	Name	Default	Description
[31:2]	-	reserved		
[1]	RW	burstf_emb_ctrl	0x0	语音F帧嵌入式信息选择： 写0表示直接从CPU 获取； 写1表示内部进行RC编码。
[0]	RW	pi_bit	0x0	语音帧BCDE EMB区域PI信息配置

### 10.2.2.13 SCRAMBLE\_REG

偏移地址：0x38c

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	RW	dscramble_reg	0x0	语音简单加密方式，接收端解密寄存器配置

[15:0]	RW	scramble_reg	0x0	语音简单加密方式，发送端加密寄存器配置
--------	----	--------------	-----	---------------------

#### 10.2.2.14 RX\_TYPE\_INFO

偏移地址：0x390

复位方式：H

Bits	Access	Name	Default	Description
[31]	-	reserved		
[30]	RO	crc8_error_flag	0x0	shortlc信息的CRC8校验结果： 0表示正确； 1表示错误。
[29]	RO	golay_check_flag	0x0	数据帧SLOT_TYPE区域Golay译码结果： 0表示正确； 1表示错误。
[28]	RO	qr_check_flag	0x0	语音帧EMB区域QR译码结果： 0表示正确； 1表示错误。
[27]	RO	cs_check_flag	0x0	EMB后接入信息的（Checksum）奇偶校验结果： 0表示正确； 1表示错误。
[26]	RO	crc16_error_flag	0x0	CRC16校验方式的数据帧校验结果： 0表示正确； 1表示错误。
[25]	RO	rs24_error_flag	0x0	RS24校验方式的数据帧校验结果： 0表示正确； 1表示错误。
[24]	RO	data_block_error	0x0	CRC9的数据帧和CRC32的反馈数据包体校验结果： 0表示正确； 1表示错误。
[23]	RO	cach_at	0x0	接收获取到的AT
[22]	RO	cach_tc	0x0	接收获取到的TC
[21:20]	RO	cach_lcsc	0x0	接收获取到的CACH的LCSS[1:0]
[19]	RO	current_rx_slot	0x0	当前接收时隙标号
[18]	RO	save_soft_finish	0x0	软信息存储完成标志
[17]	RO	pi	0x0	接收到的PI
[16:15]	RO	lcsc	0x0	接收到的语音LCSS
[14:12]	RO	current_voice_burst	0x0	当前接收的语音帧标号1~6，分别代表语音帧ABCDEF
[11:8]	RO	data_type	0x0	接收到数据帧类型

[7:4]	RO	cc	0x0	接收到的色码
[3:2]	RO	recv_sync_type	0x0	接收到的同步帧序列类型 00: MS 01: BS 10: TDMA1 11: TDMA2
[1:0]	RO	sync_type	0x0	00表示无同步头, 01表示语音, 10表示数据, 11表示RC

#### 10.2.2.15 RX\_INFO

偏移地址: 0x394

复位方式: H

Bits	Access	Name	Default	Description
[31:25]	-	reserved		
[24:16]	RO	tdma_slot_num	0x0	集群模式下从shortlc信息中获取的复帧号
[15:11]	-	reserved		
[10:0]	RO	rc_info	0x0	接收到的RC 11比特信息

#### 10.2.2.16 LAYER2\_CONTROL

偏移地址: 0x400

复位方式: H

Bits	Access	Name	Default	Description
[31:8]	-	reserved		
[7]	RW	txen	0x0	二层发送同步使能开关, 0为关闭, 1为开启
[6]	RW	rxen	0x0	二层接收同步使能开关, 0为关闭, 1为开启
[5]	RW	layer2slot	0x0	配置当前二层使用的时隙状态, 应用于二层模式平滑切换到三层时隙: 0: 当前时隙为非工作时隙; 1: 当前时隙为工作时隙。
[4]	RW	txtmp_master_mode	0x0	常规二层发送临时保留时隙建立模式, 0被动模式, 1主动模式
[3]	RW	rx_master_auto_update	0x0	二层接收主动模式下, 接收时隙边界更新标志, 0不更新, 1自动随接收同步更新。
[2]	RW	tx_slave_auto_mode	0x0	常规二层发送, 被动模式下时隙建立方式: 1, 自动建立收发同步; 0, 手动建立接收启动发送时隙。
[1]	RW	tx_master_mode	0x0	常规二层发送时隙建立模式, 0被动模式, 1主动模式
[0]	RW	rx_master_mode	0x0	二层接收时隙建立模式, 0被动模式, 1主

				动模式，默认要求设置成0。
--	--	--	--	---------------

### 10.2.2.17 LAYER2\_SLOTON

偏移地址：0x404

复位方式：H

Bits	Access	Name	Default	Description
[31]	RO	o_dll_tx_slot_on	0x0	常规二层模式发送时隙开启标志，高有效
[30]	RO	o_dll_tx_slot_on_tmp	0x0	常规二层模式临时保留时隙开启标志，高有效
[29]	RO	o_dll_rx_slot_on	0x0	常规二层模式接收时隙开启标志，高有效
[28]	RO	o_bs_tx_slot_on	0x0	中继芯片发送模式时隙开启标志，高有效
[27]	RO	o_bs_rx_slot_on	0x0	中继芯片接收模式时隙开启标志，高有效
[26:14]	-	reserved		
[13]	WC	dll_rx_close_txtmp	0x0	常规二层模式下临时保留时隙结束开关，写1有效，芯片自动清除（同时要求在Bit1 dll_tx_slot_on_tmp有效）。
[12]	WC	dll_rx_open_txtmp	0x0	常规二层模式下接胡搜时隙建立临时保留时隙启动开关，写1有效，芯片建立时隙后自动清除（同时要求在Bit1 dll_tx_slot_on_tmp有效以及 txtmp_master_mode为被动模式）。
[11]	RW	dll_tx_slot_frm_tmp	0x0	常规二层模式下建立临时保留时隙启动开关，由临时时隙启动控制
[10]	WC	dll_tx_close_txtmp	0x0	常规二层模式下临时保留时隙结束开关，写1有效，芯片自动清除（同时要求在Bit1 dll_tx_slot_on_tmp有效）。
[9]	WC	dll_txtmp_close_tx	0x0	常规二层模式下临时保留发送时隙建立发送时隙结束开关，写1有效，芯片自动清除。（Bit0 dll_tx_slot_on有效以及 txtmp_master_mode为被动模式）
[8]	WC	dll_rx_close_tx	0x0	常规二层模式下被动模式建立发送时隙结束开关，写1有效，芯片自动清除。（同时要求tx_slave_auto_mode设为0以及在Bit0 dll_tx_slot_on有效）
[7]	WC	dll_tx_open_txtmp	0x0	常规二层模式下发送时隙建立临时保留时隙启动开关，写1有效，芯片建立时隙后自动清除（同时要求在Bit1 dll_tx_slot_on_tmp有效以及 txtmp_master_mode为被动模式）。
[6]	WC	dll_txtmp_open_tx	0x0	常规二层主动模式下临时保留发送时隙建立发送时隙启动开关，写1有效，芯片自动清除。（Bit0 dll_tx_slot_on有效 同时Bit11 dll_tx_slot_frm_tmp有效）

[5]	WC	dll_rx_open_tx	0x0	常规二层模式下被动模式建立发送时隙启动开关，写1有效，芯片建立时隙后自动清除。（同时要求tx_slave_auto_mode设为0以及在Bit0 dll_tx_slot_on有效）
[4]	RW	bs_rx_slot_on	0x0	中继芯片主动接收模式建立接收时隙使能，高电平有效
[3]	RW	bs_tx_slot_on	0x0	中继芯片发送模式建立发送时隙使能，高电平有效
[2]	RW	dll_rx_slot_on	0x0	常规二层主动接收模式下建立接收时隙使能，高电平有效
[1]	RW	dll_tx_slot_on_tmp	0x0	常规二层模式下建立发送临时保留时隙使能，高电平有效
[0]	RW	dll_tx_slot_on	0x0	常规二层模式下建立发送时隙使能，高电平有效

#### 10.2.2.18 LAYER2\_TXRX\_CTRL

偏移地址：0x408

复位方式：H

Bits	Access	Name	Default	Description
[31]	WC	tx_complete_cpu	0x0	二层连续发送模式下发送结束配置用以切换RF中断控制，写1有效
[30:24]	-	reserved		
[23]	RW	syncfail_layer2_rx	0x0	二层失去接收时隙同步控制开关： 0：维持不变； 1：释放同步时隙，重新建立。
[22]	RW	syncfail_layer2_tx	0x0	二层失去发送同步控制开关： 0：维持不变； 1：释放同步时隙，重新建立。
[21:16]	-	reserved		
[15]	RW	begin_v_layer2	0x0	二层模式下，语音接入成功标志，软件写入1后，语音开始正常接入语音，但在不接收语音的那个时隙需要将bit置零。
[14:8]	-	reserved		
[7]	RW	txnextsloten	0x0	下一时隙发送使能： 0：即将到来的时隙中断开始不发送； 1：即将到来的时隙中断开始发送。
[6]	RW	rxnextsloten	0x0	下一时隙接收使能： 0：即将到来的时隙中断开始不接收； 1：即将到来的时隙中断开始接收。
[5:1]	-	reserved		
[0]	RW	autotest	0x0	物理层测试使能，高有效

### 10.2.2.19 LAYER2\_SLOT\_CNT

偏移地址：0x40c

复位方式：H

Bits	Access	Name	Default	Description
[31]	RW	rx_slot_flag_init	0x0	常规二层或者中继接收时隙标号初值（主动模式下）
[30]	RW	tx_slot_flag_init	0x0	常规二层或者中继发送时隙标号初值（主动模式下）
[29]	WC	txtmpslot_cnt_set_en	0x0	常规二层或者中继发送时隙计数初值配置使能（主动模式下）
[28:20]	RW	txtmpslot_cnt_set	0x0	常规二层或者中继发送时隙计数初值配置（主动模式下）
[19]	WC	rxslot_cnt_set_en	0x0	常规二层或者中继接收时隙计数初值配置使能（主动模式下）
[18:10]	RW	rxslot_cnt_set	0x0	常规二层或者中继接收时隙计数初值配置（主动模式下）
[9]	WC	txslot_cnt_set_en	0x0	常规二层或者中继发送时隙计数初值配置使能（主动模式下）
[8:0]	RW	txslot_cnt_set	0x0	常规二层或者中继发送时隙计数初值配置（主动模式下）

### 10.2.2.20 LAYER2\_MAX\_CNT

偏移地址：0x410

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	-	reserved		
[15]	RW	fine_tune_sign_rx	0x0	常规二层或者中继接收时隙修正符号 1: 缩短时隙 0: 延长时隙
[14:12]	-	reserved		
[11:8]	RW	fine_tune_cnt_rx	0x0	常规二层或者中继接收时隙修正值，每增减1表示调整100us
[7]	RW	fine_tune_sign_tx	0x0	常规二层或者中继发送时隙修正符号 1: 缩短时隙 0: 延长时隙
[6:4]	-	reserved		
[3:0]	RW	fine_tune_cnt_tx	0x0	常规二层或者中继发送时隙修正值，每增减1表示调整100us

### 10.2.2.21 LAYER2\_SLOT\_UPDATE

偏移地址：0x414

复位方式: H

Bits	Access	Name	Default	Description
[31:14]	-	reserved		
[13]	WC	rx_sync_txtmp_flag_en	0x0	常规二层接收时隙同步到发送临时保留时隙时, 时隙标号更新标志, 高有效
[12]	-	reserved		
[11]	WC	txtmp_sync_tx_flag_en	0x0	常规二层发送临时保留时隙同步到发送时隙时, 时隙标号更新标志, 高有效
[10]	WC	tx_sync_txtmp_flag_en	0x0	常规二层发送时隙同步到发送临时保留时隙时, 时隙标号更新标志, 高有效
[9:6]	-	reserved		
[5]	WC	rx_sync_txtmp_en	0x0	常规二层接收时隙同步到发送临时保留时隙控制
[4]	WC	bs_tx_sync_rx_en	0x0	中继发送时隙同步到中继接收时隙控制
[3]	WC	txtmp_sync_tx_en	0x0	常规二层发送临时保留时隙同步到发送时隙控制
[2]	WC	tx_sync_txtmp_en	0x0	常规二层发送时隙同步到发送临时保留时隙控制
[1]	WC	rx_sync_tx_en	0x0	常规二层接收时隙同步到发送时隙控制
[0]	WC	tx_sync_rx_en	0x0	常规二层发送时隙同步到接收时隙控制

#### 10.2.2.22 LAYER2\_SEND\_TYPE

偏移地址: 0x418

复位方式: H

Bits	Access	Name	Default	Description
[31:30]	RW	encodetype1011	0x0	自定义数据类型帧 (DataType为1011) 的编解码类型选择: 00 表示 BPTC96 编解码; 01 表示卷积 3/4 编解码; 10 表示无编解码; 11 表示 BPTC72 编解码;
[29:28]	RW	encodetype1100	0x0	自定义数据类型帧 (DataType为1100) 的编解码类型选择: 00 表示 BPTC96 编解码; 01 表示卷积 3/4 编解码; 10 表示无编解码; 11 表示 BPTC72 编解码;
[27:26]	RW	encodetype1101	0x0	自定义数据类型帧 (DataType为1101) 的编解码类型选择: 00 表示 BPTC96 编解码; 01 表示卷积 3/4 编解码; 10 表示无编解码; 11 表示 BPTC72 编解码;

[25:24]	RW	encodetype1110	0x0	自定义数据类型帧 (DataType为1110) 的编解码类型选择: 00 表示 BPTC96 编解码; 01 表示卷积 3/4 编解码; 10 表示无编解码; 11 表示 BPTC72 编解码;
[23:22]	RW	encodetype1111	0x0	自定义数据类型帧 (DataType为1111) 的编解码类型选择: 00 表示 BPTC96 编解码; 01 表示卷积 3/4 编解码; 10 表示无编解码; 11 表示 BPTC72 编解码;
[21:13]	-	reserved		
[12]	RW	data_embrc_en	0x0	数据帧SYNC区域嵌入RC信息控制使能, 高有效
[11]	RW	data_embrc_ctrl	0x0	数据控制帧EMB区域插入RC信息选择控制 0: 表示选择直接从CPU 获取; 1: 表示内部进行RC编码处理
[10]	WC	bs_tx_data_rdy	0x0	BS下行转发帧类型数据准备完成控制: 0: 未完成转发数据, 芯片自动组帧发送IDLE帧; 1: 下行转发数据准备完成, 在时隙边界芯片自动清除标志。
[9]	WC	bs_tx_data_source	0x0	BS下行转发的数据帧数据选择: 0: 默认常规芯片完成编码组帧; 1: 直接发送无需进行编码的196比特信息, 在时隙边界芯片自动清除标志。
[8]	RW	crc_calc_sel	0x0	CRC校验方式选择控制: 0: 芯片自动完成校验计算; 1: CRC校验计算由CPU完成。
[7:4]	RW	localdatatype	0x0	每一个发送时隙的数据帧data_type值、语音A~F帧以及RC帧类型值
[3]	RW	localvod	0x0	发送帧类型选择: 0: 数据帧; 1: 语音或者RC帧。
[2]	RW	seconddataheader	0x0	data_type为数据帧头时对应数据二头控制选择: 0: 常规数据帧头; 1: 第二数据帧头。
[1:0]	RW	locallcss	0x0	每一个发送时隙的 LCSS

### 10.2.2.23 LAYER2\_STATUS

偏移地址：0x41c

复位方式：H

Bits	Access	Name	Default	Description
[31]	RO	tx_slot_choose	0x0	二层发送时隙标号
[30]	RO	rx_slot_choose	0x0	二层接收时隙标号
[29]	RO	tx_slot_choose_tmp	0x0	二层临时发送时隙标号
[28:27]	-	reserved		
[26:18]	RO	tx_bit_cnt_tmp	0x0	二层临时发送时隙计数值读取
[17:9]	RO	rx_bit_cnt	0x0	二层接收时隙计数值读取
[8:0]	RO	tx_bit_cnt	0x0	二层发送时隙计数值读取

### 10.2.2.24 LAYER2\_SLOT\_PRE

偏移地址：0x424

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	rdy_slot_pre	0x55	默认值(不可随意修改)
[15:8]	RW	rdy_interp_pre_rx	0x5	接收二层处理中断位置提前配置
[7:0]	RW	rdy_interp_pre_tx	0x5	发送二层处理中断位置提前配置

### 10.2.2.25 LAYER2\_BS\_MODE

偏移地址：0x428

复位方式：H

Bits	Access	Name	Default	Description
[31:12]	-	reserved		
[11:8]	RW	bs_idle_type	0x9	双频中继BS下行发送数据类型配置
[7:6]	-	reserved		
[5:4]	RW	gps_update_rate	0x0	同频同播模式下时隙更新调整周期设置： 00：表示每隔6s； 01：表示每隔3s； 10：表示每隔1s。
[3]	RW	gps_pps_en	0x0	同频同播模式
[2]	RW	bs_tx_mode	0x0	1表示芯片工作于BS发送模式
[1]	RW	bs_rx_mode	0x0	1表示芯片工作于BS接收模式
[0]	RW	bs_single_freq	0x0	BS模式选择： 0：双频中继模式；1：单频中继模式。

### 10.2.2.26 BS\_SECOND\_REG

偏移地址：0x430

复位方式：H

Bits	Access	Name	Default	Description
[31:6]	-	reserved		
[5:0]	RW	bs_second_reg	0x0	秒时间值，范围0~59

#### 10.2.2.27 BS\_STATUS

偏移地址：0x434

复位方式：H

Bits	Access	Name	Default	Description
[31:30]	RO	gps_lcss_cnt	0x0	GPS创建的LCSS循环计数器
[29:28]	RO	lcss_counter	0x0	CACH的LCSS循环计数器
[27:19]	RO	gps_bit_cnt	0x0	同频同播模式下GPS创建的本地30ms时隙计数器
[18:8]	RO	bs_slot_num	0x0	同频同播模式下60ms计数的帧号，60s为周期循环。
[7:6]	-	reserved		
[5:0]	RO	bs_second_lock	0x0	时隙启动使用的秒时间锁定值

#### 10.2.2.28 BS\_CACH

偏移地址：0x438

复位方式：H

Bits	Access	Name	Default	Description
[31:2]	-	reserved		
[1]	RW	bs_cach_at	0x0	BS发送模式配置AT信息
[0]	RW	bs_cach_tc	0x0	BS发送模式配置TC信息

### 10.3 中断向量

DMR 数字链路层模式下，定义中断类型如下表所示。

表 50 链路层中断类型列表

序号	名称	类型	描述
1	RF_TX_INTERP	电平触发，高有效	射频切换中断，RF 发送开始中断，通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
2	RF_RX_INTERP	电平触发，高有效	射频切换中断，RF 接收开始中断，通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
3	SYS_INTERP	电平触发，高有效	基带系统中断，有子中断见下表，通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除
4	TIME_SLOT_TX_INTERP	电平触发，高有效	基带数字 30ms 发送时隙中断，通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
5	TIME_SLOT_RX_INTERP	电平触发，高有效	基带数字 30ms 接收时隙中断，通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除

6	PCM_WR_INTERP	电平触发, 高有效	Codec 音频数据流写入中断, 通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除
7	PCM_RD_INTERP	电平触发, 高有效	Codec 音频数据流读取中断, 通过 INTERP_CLEAR 寄存器对应 bit 位写 1 清除

表 51 链路层系统中断子中断类型列表

序号	名称	类型	描述
1	DLL_RDY_INTERP_TX	电平触发, 高有效	二层发送处理中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
2	DLL_RDY_INTERP_RX	电平触发, 高有效	二层接收处理中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
3	DLL_LATELC_RX_INTERP	电平触发, 高有效	二层语音后接入中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEA
4	DLL_FRAME_RX_INTERP	电平触发, 高有效	二层各类型帧信息接收中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEA
5	DLL_SHORTLC_RX_INTERP	电平触发, 高有效	二层 shortlc 信息接收中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEA
6	DLL_TX_SLOT_ABNORMAL	电平触发, 高有效	二层发送时隙异常中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEA
7	DLL_RX_SLOT_ABNORMAL	电平触发, 高有效	二层接收时隙异常中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过SYS_INTERP_CLEA
8	PHY_RECV_ABNORMAL	电平触发, 高有效	物理层接收时隙边界异常提示中断: 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽; 通过 SYS_INTERP_CLEA

## 10.4 Memory 映射

基带 Ram 模块是 HR\_C7000 芯片基带数字功能主要的用户数据交互存储空间，根据不同的功能设计主要分为发送和接收 2 个 Ram，数据长度为 8，深度为 1200。

### 10.4.1 链路层发送 Ram

链路层发送 Ram 的基地址为 0x16000000，深度为 1200，下表定义了各个地址段内容分配。

表 52 链路层发送 Ram 分配

偏址	名称	描述
0x00~0x0b	PI Header	0x00~0x09: 共计 80bit 为控制信息; 0x0a~0x0b: 共计 16bit 为校验信息, MCU 可选。
0x00~0x0b	LC Header	0x00~0x08: 共计 72bit 为控制信; 0x09~0x0b: 共计 24bit 为校验信息, MCU 可选。
0x00~0x0b	Terminator	0x00~0x08: 共计 72bit 为控制信; 0x09~0x0b: 共计 24bit 为校验信息, MCU 可选。
0x00~0x0b	CSBK	0x00~0x09: 共计 80bit 为控制信息; 0x0a~0x0b: 共计 16bit 为校验信息, MCU 可选。
0x00~0x0b	MBC Header	0x00~0x09: 共计 80bit 为控制信息; 0x0a~0x0b: 共计 16bit 为校验信息, MCU 可选。
0x00~0x0b	MBC Block	0x00~0x0b: 共计 96bit 为控制信息;
0x00~0x0b	DataHeader	0x00~0x09: 共计 80bit 为控制信息; 0x0a~0x0b: 共计 16bit 为校验信息, MCU 可选。
0x00~0x0b	DataRate1_2	0x00~0x0b: 共计 96bit 为数据信息;
0x00~0x11	DataRate3_4	0x00~0x0b: 共计 144bit 为数据信息;
0x00~0x17	DataRate1	0x00~0x0b: 共计 192bit 为数据信息;
0x00~0x0a	C_RC (PDT)	0x00、0x01 的 bit7-bit5: 共计 11bit RC 信息; 0x02~0x08: 共计 56 bit 控制信息; 0x09~0x0a: 共计 16 bit 校验信息;
0x00~0x09	EMB LC	0x00~0x08: 共计 72bit 为控制信; 0x09: bit7-bit3 共计 5bit 为校验信息, MCU 可选。 此处信息在要准备发送语音帧 A 时同时准备。
0x29~0x2C	EMBF	语音 F 帧填充信息或者 0x29 存放超帧序号 (KeyID), 0x2A 高 3bit 存放加密序号 (ALOG ID)
0x30~0x4a	VoiceBurstA	共计 216bit 为语音帧信息, 只有在语音发送数据选择内部 Ram 模式下有效
0x30~0x4a	VoiceBurstB	共计 216bit 为语音帧信息, 只有在语音发送数据选择内部 Ram 模式下有效
0x30~0x4a	VoiceBurstC	共计 216bit 为语音帧信息, 只有在语音发送数据选择内部 Ram 模式下有效
0x30~0x4a	VoiceBurstD	共计 216bit 为语音帧信息, 只有在语音发送数据选择内部 Ram 模式下有效

0x30~0x4a	VoiceBurstE	共计 216bit 为语音帧信息，只有在语音发送数据选择内部 Ram 模式下有效
0x30~0x4a	VoiceBurstF	共计 216bit 为语音帧信息，只有在语音发送数据选择内部 Ram 模式下有效
0x4b~0x50	EMB RC	数据控制帧嵌入的 48bit 的 RC 信息或者 0x4b、0x4c 的高 11bit 的 RC 编码器输入
0x30~0x48	196bits info	196bit 控制信息
0x495~0x4af	SendVoiceKey	共计 216bit, 27 bytes

#### 10.4.2 链路层接收 Ram

链路层接收 Ram 的基地址为 0x160004b0，深度为 1200，下表定义了各个地址段内容分配。

表 53 链路层接收 Ram 分配

偏址	名称	描述
0x00~0x0b	PI Header	0x00~0x09: 共计 80bit 为控制信息; 0x0a~0x0b: 共计 16bit 为校验信息。
0x00~0x0b	LC Header	0x00~0x08: 共计 72bit 为控制信; 0x09~0x0b: 共计 24bit 为校验信息。
0x00~0x0b	Terminator	0x00~0x08: 共计 72bit 为控制信; 0x09~0x0b: 共计 24bit 为校验信息。
0x00~0x0b	CSBK	0x00~0x09: 共计 80bit 为控制信息; 0x0a~0x0b: 共计 16bit 为校验信息。
0x00~0x0b	MBC Header	0x00~0x09: 共计 80bit 为控制信息; 0x0a~0x0b: 共计 16bit 为校验信息。
0x00~0x0b	MBC Block	0x00~0x0b: 共计 96bit 为控制信息;
0x00~0x0b	DataHeader	0x00~0x09: 共计 80bit 为控制信息; 0x0a~0x0b: 共计 16bit 为校验信息。
0x00~0x0b	DataRate1_2	0x00~0x0b: 共计 96bit 为数据信息;
0x00~0x11	DataRate3_4	0x00~0x11: 共计 144bit 为数据信息;
0x00~0x17	DataRate1	0x00~0x17: 共计 192bit 为数据信息;
0x00~0x0b	Idle	0x00~0x0b: 共计 96bit 为控制信息;
0x00~0x08	C_RC (PDT)	0x00~0x06: 共计 56 bit 控制信息; 0x07~0x08: 共计 16 bit 校验信息; RC 译码结果 11bit 信息保存在寄存器中。
0x1f~0x24	EMB_48_INFO_0	时隙 0 接收的 48bit EMB 区域信息
0x25~0x2a	EMB_48_INFO_1	时隙 1 接收的 48bit EMB 区域信息
0x2b~0x2f	ShortLC	接收端 CACH 字段 36bit ShortLC 数据, 其中 0x2f 的低 4bit 为无用数据

0x36~0x50 或 0x39~0x53	216bit_voice_info _0	当前时隙 0 解析结果，语音帧的 216bit 信息 时隙接收模式时，0x36~0x50； 连续接收模式时，0x39~0x53；
0x66~0x80 或 0x69~0x83	216bit_voice_info _1	当前时隙 1 解析结果，语音帧的 216bit 信息 时隙接收模式时，0x66~0x80； 连续接收模式时，0x69~0x83；
0x495~0x4af	RecvVoiceKey	共计 216bit，27 bytes

## 10.5 链路层时隙管理

### 10.5.1 概述

链路层模式下时隙管理主要涉及到时隙的建立、同步更新的过程，按照功能区分主要分发送时隙和接收时隙，同时发送时隙又包括一套发送临时保留时隙的机制。

### 10.5.2 发送时隙

发送时隙的建立方式包括主动模式和被动模式，其中被动模式又包括接收自动建立发送时隙、接收手动建立发送时隙以及发送临时时隙建立发送时隙三种方式。

发送时隙的同步过程，包括发送同步到其它时隙和发送时隙被同步两个过程。

#### 10.5.2.1 发送时隙主动建立模式

发送时隙主动建立模式实现步骤如下：

- 1、配置发送的工作使能 txen、发送时隙建立模式，配置寄存器 LAYER2\_CONTROL，参考值 0xd2；（默认开启 rxen 以及临时时隙建立主动模式）
- 2、配置配置时隙计数初值和初始标号，配置寄存器 LAYER2\_SLOT\_CNT；
- 3、配置发送时隙启动使能开关，配置寄存器 LAYER2\_SLOTON 的 Bit0 为 1；
- 4、时隙使能启动后，时隙计数器自动以 0~287 循环计数，计满 287 后产生 30ms 发送时隙中断，时隙计数值可以通过寄存器 LAYER2\_STATUS 读取；
- 5、产生独立的发送时隙中断，通过寄存器 INTERP\_CLEAR 清除 30ms 发送时隙中断。

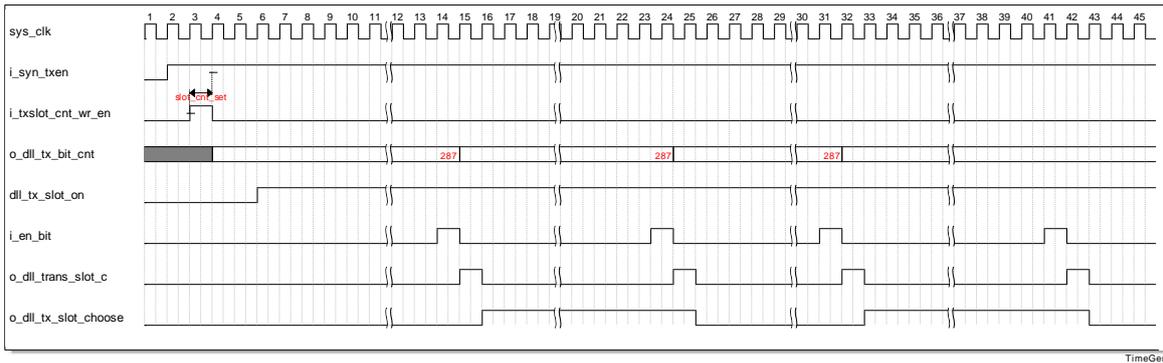


图 73 发送时隙主动建立时序图

### 10.5.2.2 发送时隙接收建立模式

发送时隙由接收时隙建立模式实现步骤：

- 1、 配置发送的工作使能 txen、发送时隙建立模式，配置寄存器 LAYER2\_CONTROL，参考值 0xd0 表示接收时隙产生后需要手动建立，0xd4 表示接收时隙产生后自动建立；（默认开启 rxen 以及临时时隙建立主动模式）
- 2、 配置发送时隙启动使能开关，配置寄存器 LAYER2\_SLOTON 的 Bit0 为 1；
- 3、 若设置成自动建立模式，则接收时隙中断产生时将自动建立发送的时隙；
- 4、 若设置成手动建立模式，则需要先检测到接收时隙 30ms 中断后，配置寄存器 LAYER2\_SLOTON 为 0x21 后建立发送 30ms 时隙；
- 5、 时隙使能启动后，时隙计数器自动以 0~287 循环计数，计满 287 后产生 30ms 发送时隙中断，时隙计数值可以通过寄存器 LAYER2\_STATUS 读取；
- 6、 产生独立的发送时隙中断，通过寄存器 INTERP\_CLEAR 清除 30ms 发送时隙中断。

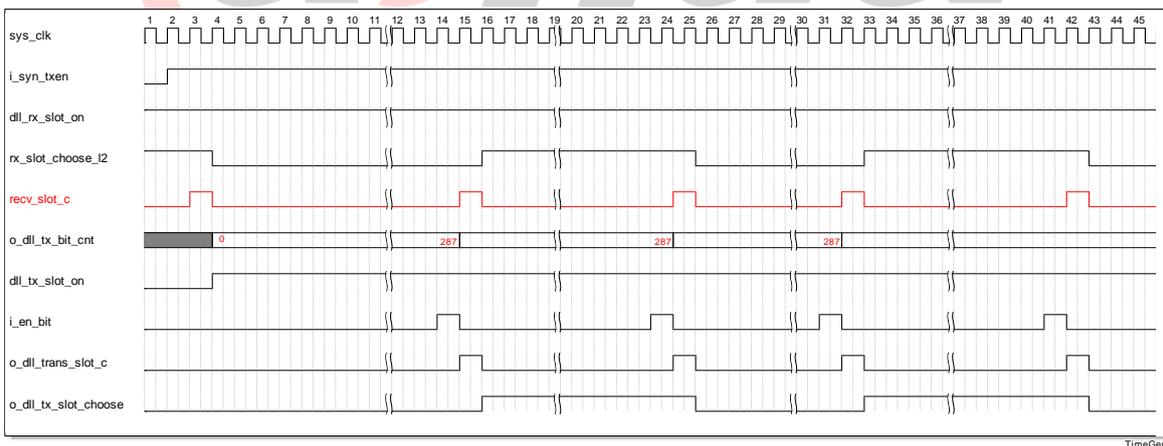


图 74 发送时隙接收时隙建立时序图

### 10.5.2.3 发送时隙临时建立模式

发送时隙由临时时隙建立模式实现步骤：

- 1、配置发送的工作使能 txen、发送时隙建立模式，配置寄存器 LAYER2\_CONTROL，参考值 0xd0；（默认开启 rxen 以及临时时隙建立主动模式）
- 2、配置发送时隙启动使能开关，配置寄存器 LAYER2\_SLOTON 的 Bit0 为 1；
- 3、需要先检测到临时时隙 30ms 开启后，配置寄存器 LAYER2\_SLOTON 为 0x843 后建立发送 30ms 时隙（前提临时时隙已经建立）；
- 4、时隙使能启动后，时隙计数器自动以 0~287 循环计数，计满 287 后产生 30ms 发送时隙中断，时隙计数值可以通过寄存器 LAYER2\_STATUS 读取；
- 5、产生独立的发送时隙中断，通过寄存器 INTERP\_CLEAR 清除 30ms 发送时隙中断。

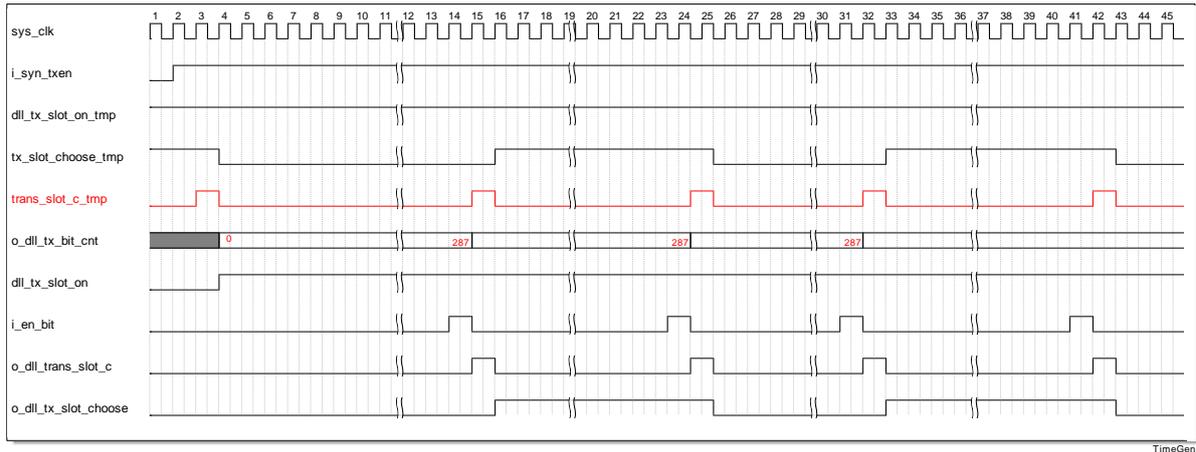


图 75 发送时隙临时时隙建立时序图

### 10.5.3 发送临时时隙

发送临时时隙的建立方式包括主动模式和被动模式，其中被动模式又包括接收时隙建立发送临时时隙、发送时隙建立发送临时时隙两种方式。

发送临时时隙的同步过程，包括发送临时同步到其它时隙和发送临时时隙被同步两个过程。

#### 10.5.3.1 临时时隙主动建立模式

临时时隙主动建立模式实现步骤如下：

- 1、配置临时时隙建立模式，配置寄存器 LAYER2\_CONTROL，参考值 0xd2；
- 2、配置配置时隙计数初值和初始标号，配置寄存器 LAYER2\_SLOT\_CNT；
- 3、配置临时时隙启动使能开关，配置寄存器 LAYER2\_SLOTON 的 Bit0 为 2；
- 4、临时时隙使能启动后，临时时隙计数器自动以 0~287 循环计数，临时时隙计数值可以通过寄存器 LAYER2\_STATUS 读取；

#### 10.5.3.2 临时时隙接收建立模式

临时时隙接收建立模式实现步骤如下：

- 1、配置临时时隙建立模式，配置寄存器 LAYER2\_CONTROL，参考值 0xc2；
- 2、配置临时时隙启动使能开关，配置寄存器 LAYER2\_SLOTON 的 Bit1 为 2；
- 3、配置接收建立临时时隙启动开关，配置寄存器 LAYER2\_SLOTON 的 Bit12 为 1；
- 4、临时时隙启动后，临时时隙计数器自动以 0~287 循环计数，临时时隙计数值可以通过寄存器 LAYER2\_STATUS 读取；

### 10.5.3.3 临时时隙发送建立模式

临时时隙发送建立模式实现步骤如下：

- 1、 配置临时时隙建立模式，配置寄存器 LAYER2\_CONTROL，参考值 0xc2；
- 2、 配置临时时隙启动使能开关，配置寄存器 LAYER2\_SLOTON 的 Bit1 为 2；
- 3、 配置发送建立临时时隙启动开关，配置寄存器 LAYER2\_SLOTON 的 Bit7 为 1；
- 4、 临时时隙启动后，临时时隙计数器自动以 0~287 循环计数，临时时隙计数值可以通过寄存器 LAYER2\_STATUS 读取；

### 10.5.4 接收时隙

接收时隙的建立方式包括主动模式和被动模式，其中被动模式通过物理层接收到同步时隙后自动建立的过程。

#### 10.5.4.1 接收时隙主动建立模式

接收时隙主动建立模式实现步骤如下：

- 1、 配置接收的工作使能 rxen、接收时隙建立模式，配置寄存器 LAYER2\_CONTROL，参考值 0xd3；（默认开启 txen 以及临时时隙建立主动模式）
- 2、 配置配置时隙计数初值和初始标号，配置寄存器 LAYER2\_SLOT\_CNT；
- 3、 配置接收时隙启动使能开关，配置寄存器 LAYER2\_SLOTON 的 Bit2 为 1；
- 4、 时隙使能启动后，时隙计数器自动以 0~287 循环计数，计满 287 后产生 30ms 接收时隙中断，时隙计数值可以通过寄存器 LAYER2\_STATUS 读取；
- 5、 产生独立的接收时隙中断，通过寄存器 INTERP\_CLEAR 清除 30ms 接收时隙中断。

#### 10.5.4.2 接收时隙被动建立模式

接收时隙被动建立模式实现步骤如下：

- 1、 配置接收的工作使能 rxen、接收时隙建立模式，配置寄存器 LAYER2\_CONTROL，参考值 0xd2；（默认开启 txen 以及临时时隙建立主动模式）
- 2、 由物理层接收到时隙边界后自动启动接收时隙计数器；
- 3、 时隙启动后，时隙计数器自动以 0~287 循环计数，计满 287 后产生 30ms 接收时隙中断，时隙计数值可以通过寄存器 LAYER2\_STATUS 读取；
- 4、 产生独立的接收时隙中断，通过寄存器 INTERP\_CLEAR 清除 30ms 接收时隙中断；
- 5、 时隙边界自动跟随物理层最新检测到的时隙边界。

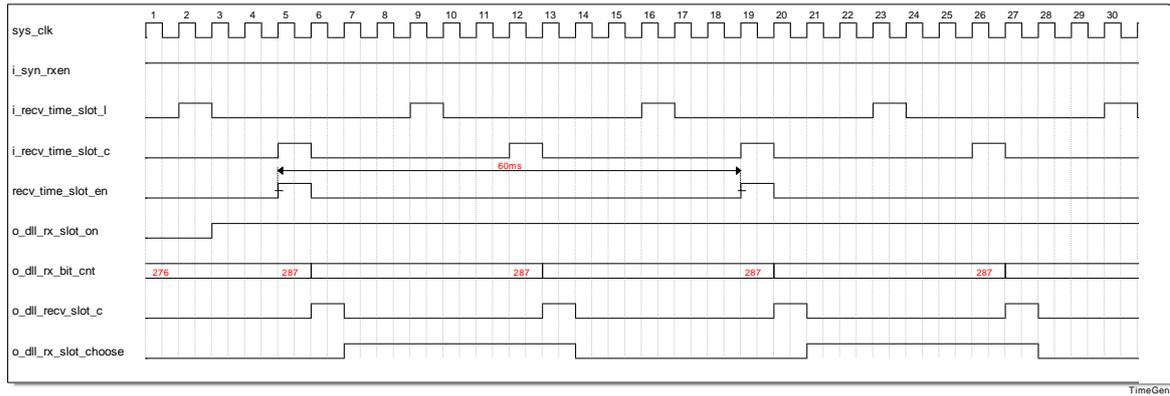


图 76 接收时隙被动建立时序图

### 10.5.5 接收时隙同步策略

时隙同步策略，主要是针对发送时隙、临时时隙和接收时隙三者彼此之间的同步方式。主要包括如下几种策略：

- 1、 发送时隙同步更新到接收时隙；
- 2、 接收时隙同步更新到发送时隙；
- 3、 发送时隙同步更新到发送临时时隙；
- 4、 发送临时时隙同步更新到发送时隙；
- 5、 接收时隙同步更新到发送临时时隙；

#### 10.5.5.1 发送时隙和接收时隙同步

发送时隙和接收时隙之间的同步策略主要有发送时隙同步到接收时隙和接收时隙同步到发送时隙。

发送时隙同步到接收时隙通过配置寄存器 `LAYER2_SLOT_UPDATE` 为 `0x1` 实现，其对应的参考时序图如下图所示。具体实现是通过发送 30ms 时隙边界产生同步脉冲到接收时隙，接收时隙根据同步需求以及接收的时隙计数值确认当前是否同步更新至发送的时隙边界，计数值范围在 `[276,287]`或`[0,11]`，则接收时隙自动更新至发送的时隙计数值 0，否则接收时隙给出时隙边界异常提示中断。

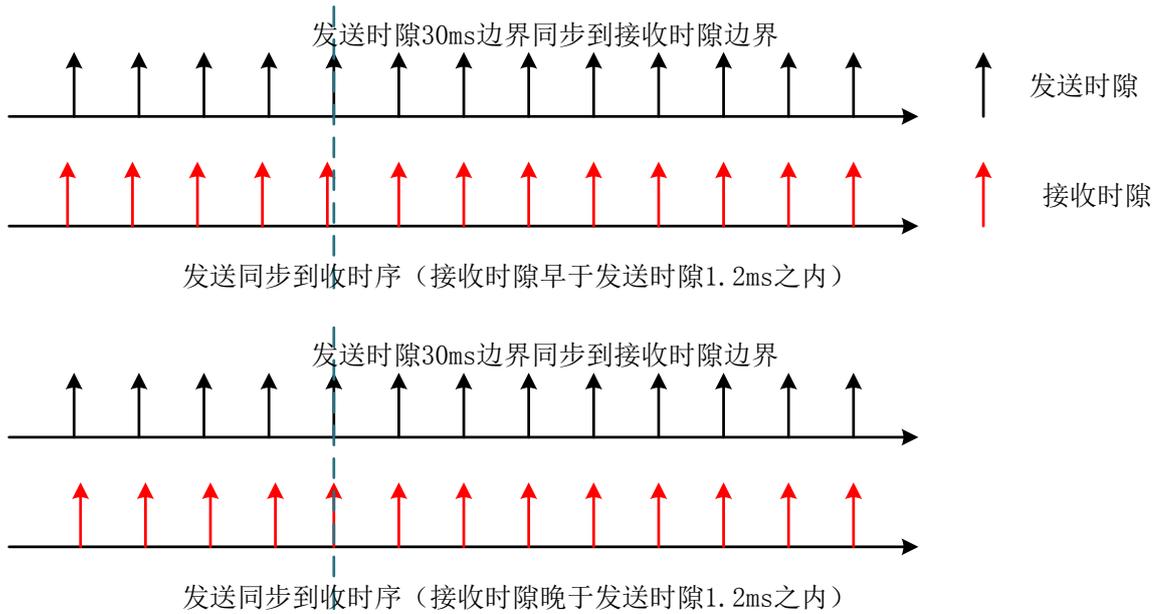


图 77 发送时隙同步到接收时隙时序图

接收时隙同步到发送时隙通过配置寄存器 LAYER2\_SLOT\_UPDATE 为 0x2 实现，其对应的参考时序图如下图所示。具体实现是通过接收 30ms 时隙边界产生同步脉冲到发送时隙，发送时隙根据同步需求以及发送的时隙计数值确认当前是否同步更新至接收的时隙边界，计数值范围在 [276,287]或[0,11]，则发送时隙自动更新至接收的时隙计数值 0，否则发送时隙给出时隙边界异常提示中断。

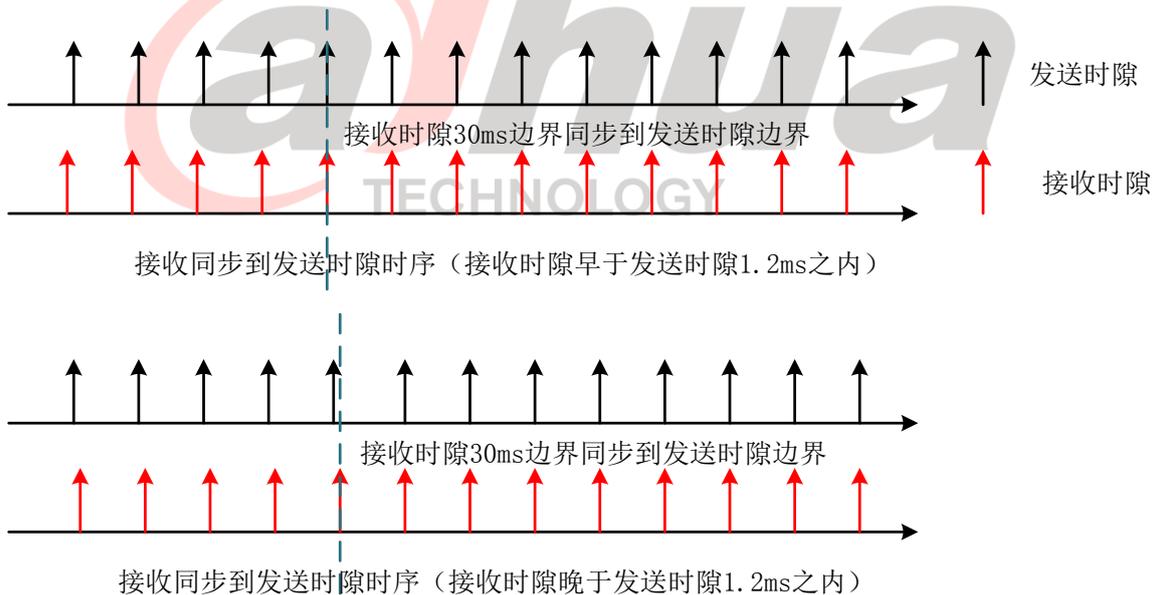


图 78 接收时隙同步到发送时隙时序图

### 10.5.5.2 发送时隙和发送临时时隙同步

发送时隙和接收时隙之间的同步策略主要有发送时隙同步到发送临时时隙和发送临时时隙同步到发送时隙。

发送时隙同步更新到发送临时时隙，自动更新时隙标号：

通过配置寄存器 LAYER2\_SLOT\_UPDATE 为 0x404 实现,其对应的参考时序图如下图所示。具体实现是通过发送 30ms 时隙边界产生同步脉冲到临时时隙,临时时隙计数值直接更新至发送时隙计数值 0,临时时隙的 30ms 边界更新脉冲条件是发送更新脉冲来时其计数值大于 257,临时时隙标号根据设置自动更新时隙标号。

发送时隙同步更新到发送临时时隙,不更新时隙标号;

通过配置寄存器 LAYER2\_SLOT\_UPDATE 为 0x4 实现,其对应的参考时序图如下图所示。具体实现是通过发送 30ms 时隙边界产生同步脉冲到临时时隙,临时时隙计数值直接更新至发送时隙计数值 0,临时时隙的 30ms 边界更新脉冲条件是发送更新脉冲来时其计数值大于 257,临时时隙标号根据设置不更新时隙标号。



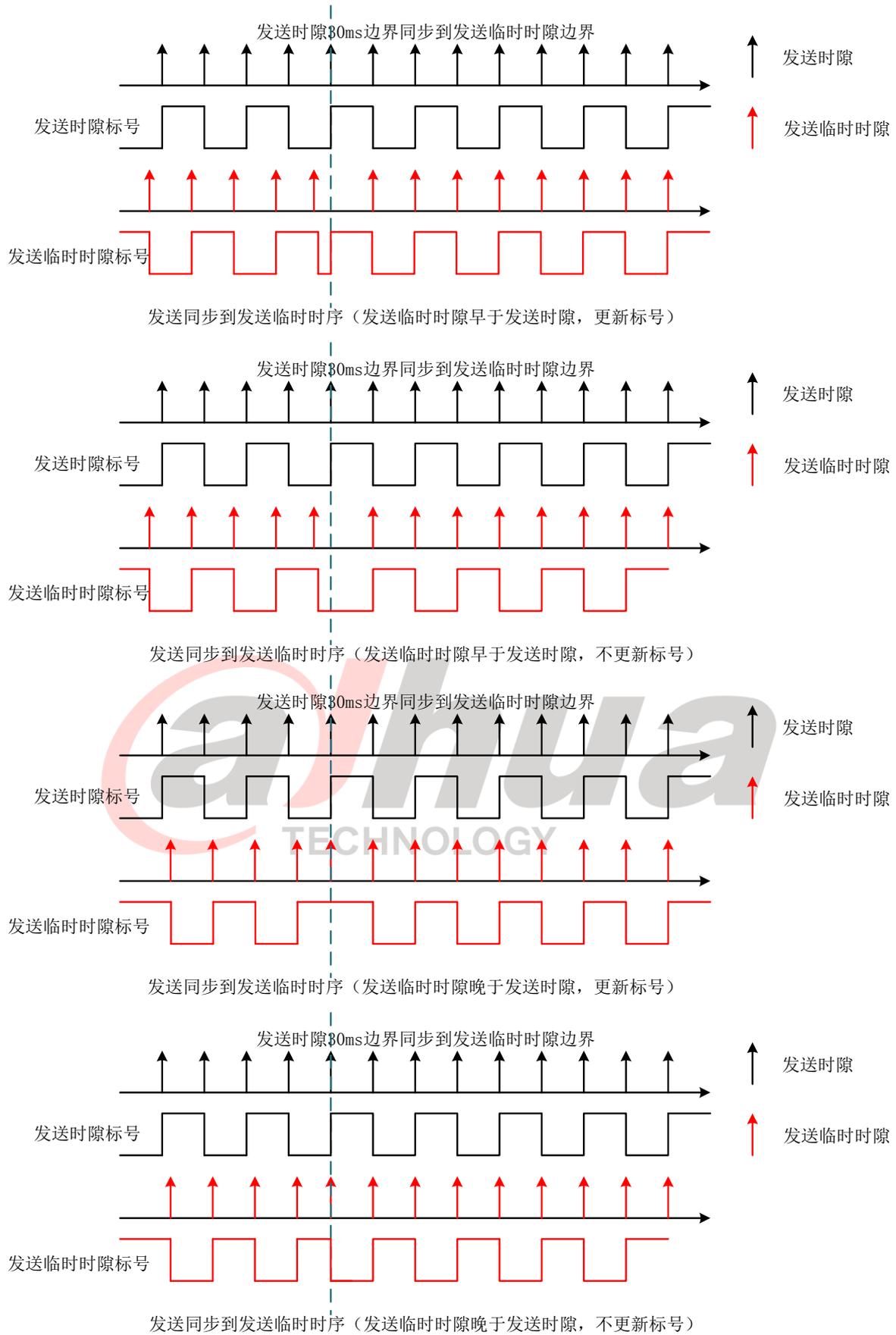


图 79 发送时隙同步到发送临时时隙时序图

### 10.5.5.3 发送临时时隙和接收时隙同步

发送临时时隙同步更新到发送时隙，自动更新时隙标号：通过配置寄存 LAYER2\_SLOT\_UPDATE 为 0x808 实现，其对应的参考时序图如下图所示。具体实现是通过临时时隙 30ms 时隙边界产生同步脉冲到发送时隙，发送时隙计数值直接更新至临时时隙计数值 0，发送时隙的 30ms 边界更新脉冲条件是临时更新脉冲来时其计数值大于 257，发送时隙标号根据设置自动更新时隙标号。

发送临时时隙同步更新到发送时隙，不更新时隙标号：通过配置寄存器 LAYER2\_SLOT\_UPDATE 为 0x8 实现，其对应的参考时序图如下图所示。具体实现是通过临时时隙 30ms 时隙边界产生同步脉冲到发送时隙，发送时隙计数值直接更新至临时时隙计数值 0，发送时隙的 30ms 边界更新脉冲条件是临时更新脉冲来时其计数值大于 257，发送时隙标号不更新时隙标号。



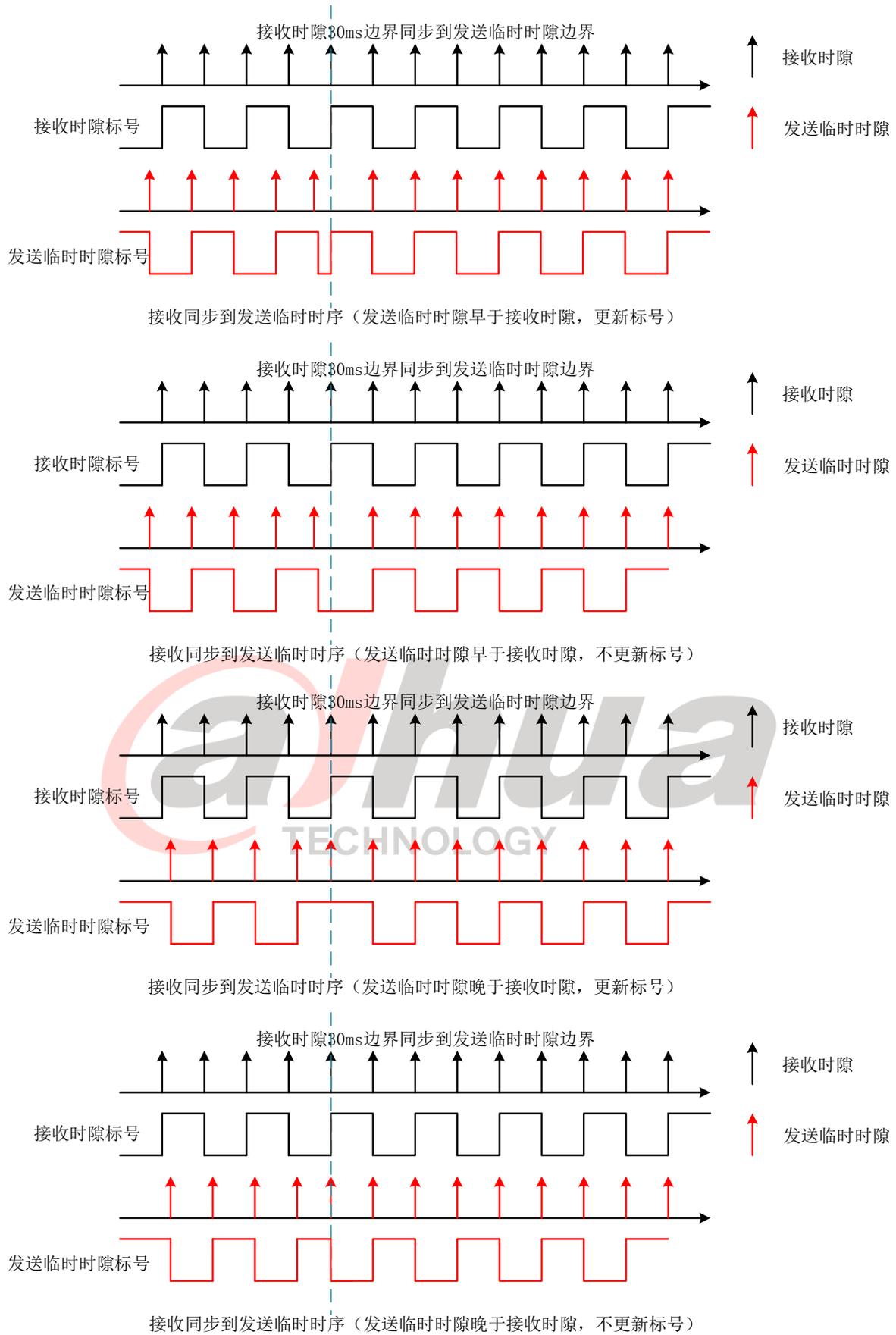


图 80 接收时隙同步到发送临时时隙时序图

## 10.6 链路层发送

### 10.6.1 概述

链路层发送支持 DMR 标准的各种类型帧的发送，主要包括语音帧、数据帧和 RC 帧三类。

### 10.6.2 帧类型描述

#### 10.6.2.1 语音帧

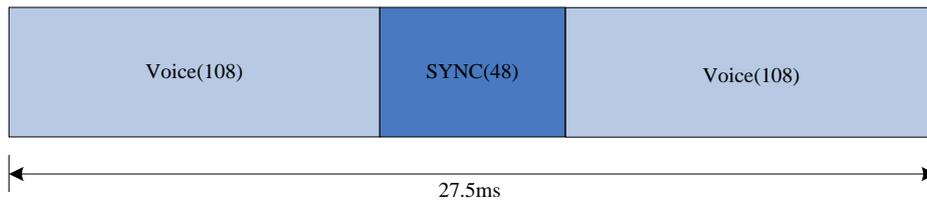


图 81 语音帧带同步帧结构

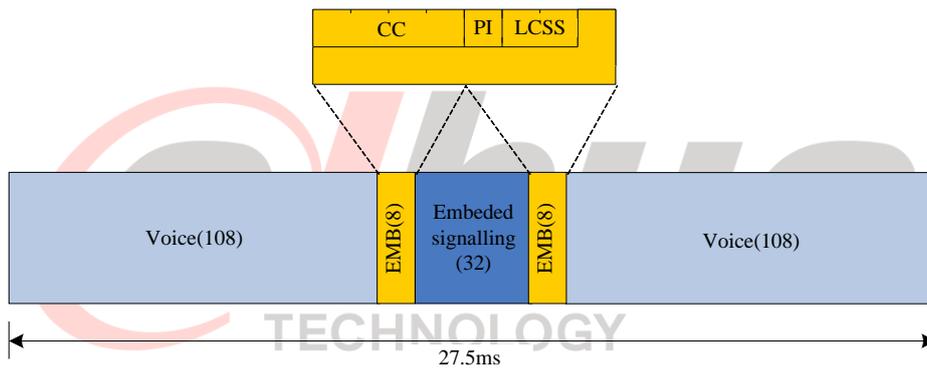


图 82 语音帧带 EMB 数据结构

支持六种语音帧，并且按超帧规则，确定语音帧中采用同步帧或者 LC 或者 Null，按照标准组建 A(SYNC)、B(LC)、C(LC)、D(LC)、E(LC)、F(Null)超帧，详细包括：

- a) 支持同步帧选择加入；
- b) 支持 EMB 7bit 加入，对 EMB 进行 QR(16,7,6)编码；
- c) 支持 LC 72bit 加入，加入 5bit CS 码，进行可变长度的 BPTC 编码、交织，并加入到 4 个时隙中(128bit)；

表 54 组呼 72bit LC 信息表

Information element	Length	Remark
Protect Flag (PF)	1	
Reserved	1	0
Full Link Control Opcode (FLCO)	6	000000
Feature set ID (FID)	8	00000000
Service Option	8	

Group address	24	
Source address	24	

d) 支持 Null 时隙选择加入；

### 10.6.2.2 数据控制帧

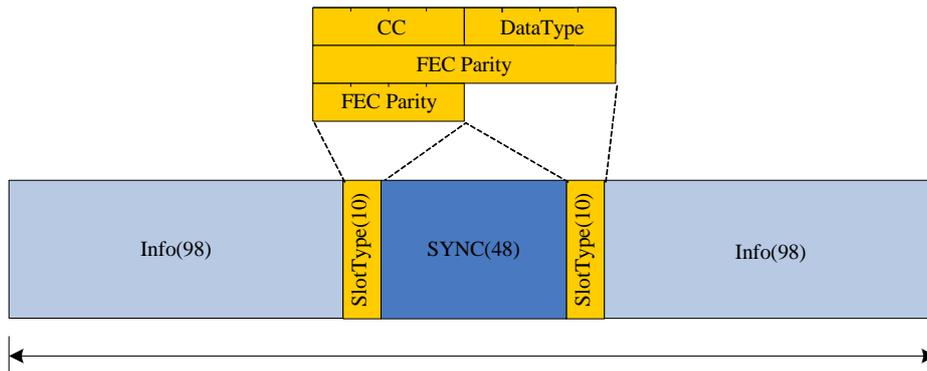


图 83 数据帧、控制帧结构

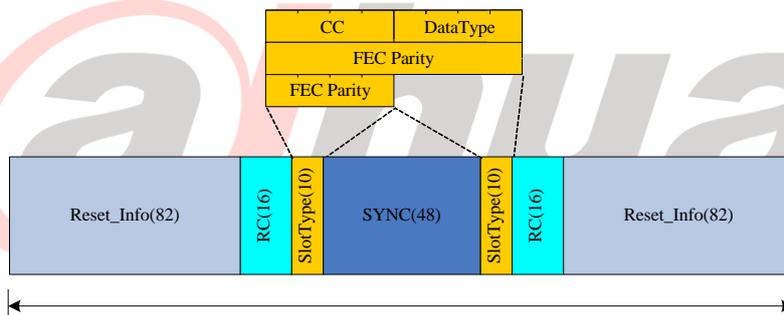


图 84 控制信道 C\_RC 帧结构(PDT)

数据帧结构主要由三部分构成：控制或者数据信息、同步帧和 Slot Type 信息，其中控制或者数据信息部分支持四种编码方式（Rate1 code、BPTC (196,96)、Rate3\_4 Trellis code、BPTC (164,72)），Slot Type 信息由 CC、Data Type 进行 Golay (20,8) 编码得到。

#### 1、PI Frame

- 支持语音 PI Header，支持加入 80bits PI，CRC16bits 校验并加入 CRC 掩码，进行 BPTC(196, 96)编码、交织；
- 支持 SYNC 加入；
- 支持加入 Slot Type(20bits)，包括 CC、DataType，进行 Golay(20, 8)编码；

#### 2、LC Frame

- 支持加入 72bit LC，CRC24bit 校验，并加入 CRC 掩码(Header 和 Terminator 区别)，进行 BPTC(196, 96)编码，组建语音 Head 和 Terminator 帧；
- 支持 SYNC 加入；

- c) 支持加入 Slot Type(20bit), 包括 CC、DataType, 进行 Golay(20, 8)编码;
  - d) 支持用 72bitLC 信息组建 128bitsEMB 嵌入信息, 用于嵌入到语音的 EMB 区域;
  - e) 支持 LC Frame 信息动态更新嵌入到 EMB 区域;
- 3、CSBK Frame
- a) 支持加入 80bits CSBK, CRC16bits 校验并加入 CRC 掩码, 进行 BPTC(196, 96)编码、交织;
  - b) 支持 SYNC 加入;
  - c) 支持加入 Slot Type(20bit), 包括 CC、DataType, 进行 Golay(20, 8)编码;
- 4、MBC Header Frame
- a) 支持加入 80bits MBC Header, CRC16bits 校验并加入 CRC 掩码, 进行 BPTC(196, 96)编码、交织;
  - b) 支持 SYNC 加入;
  - c) 支持加入 Slot Type(20bit), 包括 CC、DataType, 进行 Golay(20, 8)编码;
- 5、MBC Block Frame
- a) 支持加入 96bits MBC Block, 进行 BPTC(196, 96)编码、交织;
  - b) 支持加入 80bitMBC lastBlock, 进行 CRC16bit 校验, 进行 BPTC(196, 96)编码、交织;
  - c) 支持 SYNC 加入;
  - d) 支持加入 Slot Type(20bit), 包括 CC、DataType, 进行 Golay(20, 8)编码;
- 6、Data Header Frame
- a) 支持加入 80bits Data Header, CRC16bits 校验并加入 CRC 掩码, 进行 BPTC(196, 96)编码、交织;
  - b) 支持 SYNC 加入;
  - c) 支持加入 Slot Type(20bit), 包括 CC、DataType, 进行 Golay(20, 8)编码;
- 7、Data Rate 1/2 Frame
- a) 支持加入 96bits Data Rate 1/2, 进行 BPTC(196, 96)编码、交织;
  - b) 支持 SYNC 加入;
  - c) 支持加入 Slot Type(20bit), 包括 CC、DataType, 进行 Golay(20, 8)编码;
- 8、Data Rate 3/4 Frame
- a) 支持加入 144bits Data Rate 1/2, 进行 Trellis(196, 144)编码、交织;
  - b) 支持 SYNC 加入;
  - c) 支持加入 Slot Type(20bit), 包括 CC、DataType, 进行 Golay(20, 8)编码;
- 9、Idle Frame
- a) 支持加入 96bits Idle Frame, 进行 BPTC(196, 96)编码、交织;
  - b) 支持 SYNC 加入;
  - c) 支持加入 Slot Type(20bit), 包括 CC、DataType, 进行 Golay(20, 8)编码;
- 10、Data Rate 1 Frame
- a) 支持加入 144bits Data Rate 1, 进行 Trellis(196, 192)编码、交织;
  - b) 支持 SYNC 加入;
  - c) 支持加入 Slot Type(20bit), 包括 CC、DataType, 进行 Golay(20, 8)编码;
- 11、Reserved Frame
- a) 支持 Reserved Frame 四种编码方式可选;
  - b) 支持 SYNC 加入;
  - c) 支持加入 Slot Type(20bit), 包括 CC、DataType, 进行 Golay(20, 8)编码;

### 10.6.2.3 RC 帧

RC 帧包括两种，独立 RC 控制帧和嵌入式 RC 帧。

- a) 支持加入 7bit EMB，进行 QR(16,7,6) 编码；
- b) 支持加入 11bit RC 信号，进行可变长度 BPTC，对 32bit 进行交织，加入到 RC 单元；

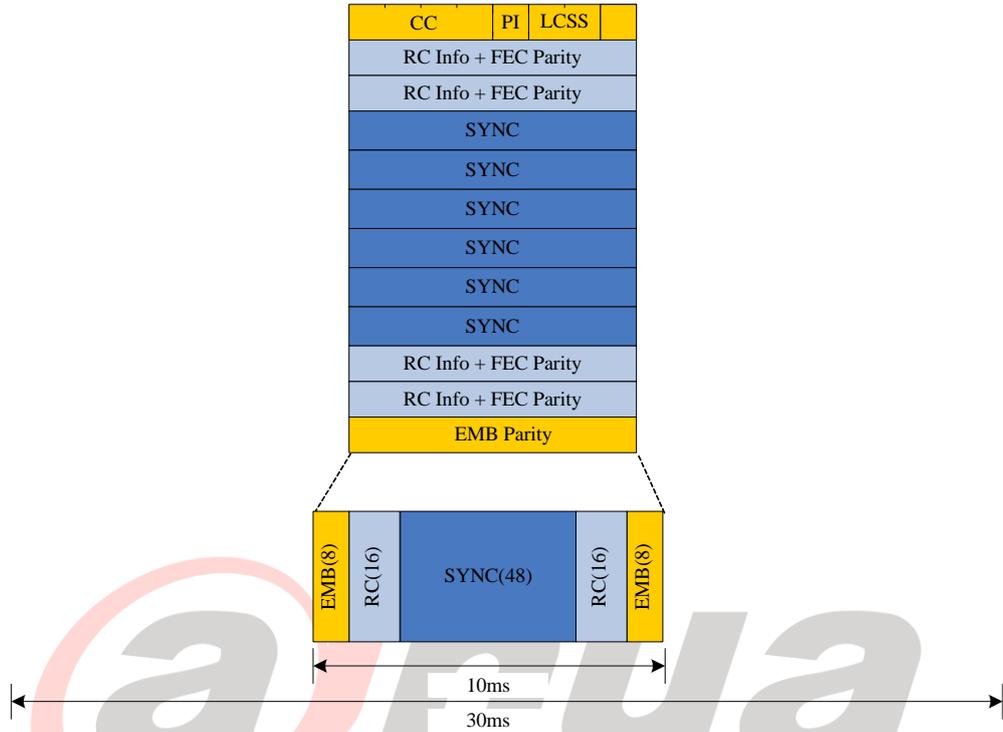


图 85 独立 RC 帧结构

### 10.6.3 组帧模式

连续模式：

- a) 支持语音超帧组帧，可设 LC Header、PI Header 或仅 PI Header 模式，结束自动构成超帧并加入 LC Terminator；
- b) 支持数据 4.8kbps 组帧，加入数据 Data Header、数据和 Data Terminator；
- c) 支持数据 9.6kbps 组帧，加入数据 Data Header、数据和 Data Terminator；

时隙模式：

- a) 支持语音超帧组帧，可设 LC Header、PI Header 或仅 PI Header 模式，结束自动构成超帧并加入 LC Terminator；
- b) 支持各种数据类型的 4.8kbps 组帧，加入数据 Data Header、数据和 LC Terminator；
- c)

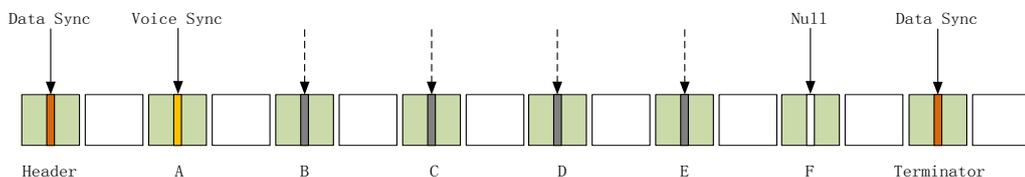


图 86 时隙语音时序

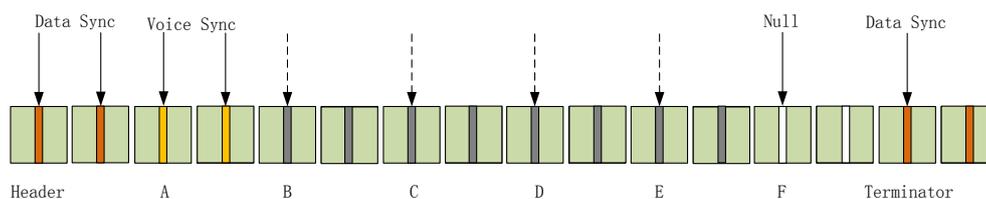


图 87 连续语音时序

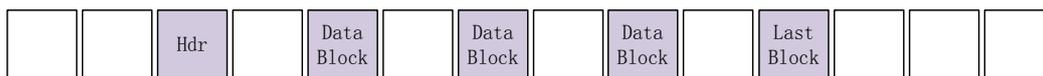


图 88 单时隙数据时序

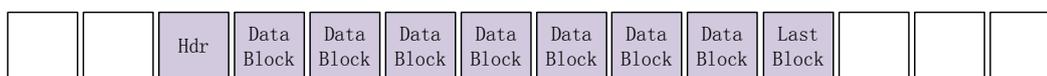


图 89 双时隙数据时序

#### 10.6.4 帧类型定义

如下表定义了链路层模式下支持发送的帧列表，包括语音超帧和各个数据类型定义帧，帧类型发送由寄存器 LAYER2\_SEND\_TYPE 配置。

表 55 链路层发送帧类型列表

时隙帧类型	LocalDataType	语音	备注
语音 LC Header	0001	0	
语音 PI Header	0000	0	
语音 A	0000	1	
语音 B	0001	1	
语音 C	0010	1	
语音 D	0011	1	
语音 E	0100	1	
语音 F	0101	1	
RC 帧	0110	1	
语音 Terminator	0010	0	
CSBK	0011	0	
MBC Header	0100	0	
MBC Intermedia	0101	0	
MBC Last	0101	0	
DataHeader	0110	0	
DataRate1_2	0111	0	
DataRate3_4	1000	0	
DataRate1	1010	0	

Idle	1001	0	BS 发送的 IDLE 帧类型的数据类型可以配置
Reserved	1011	0	
Reserved	1100	0	
Reserved	1011	0	
Reserved	1110	0	
Reserved	1111	0	

### 10.6.5 发送流程

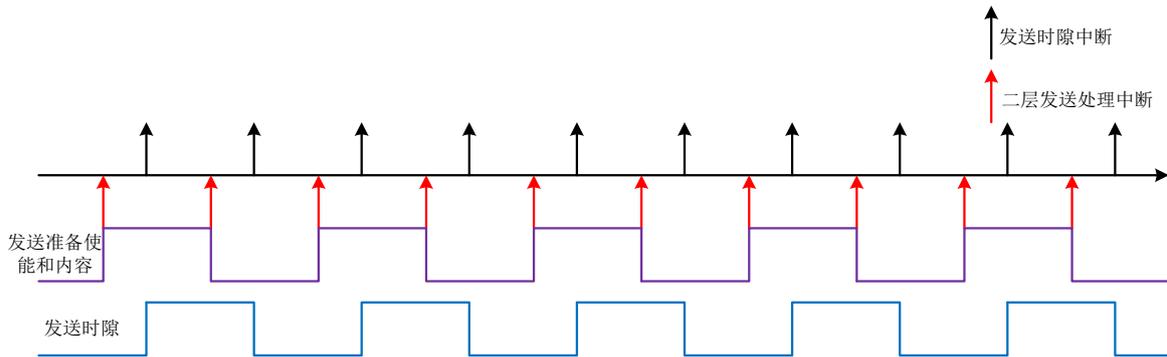


图 90 链路层发送时序图 1

链路层发送步骤如下：

- 1、 建立发送时隙，产生发送时隙中断和二层发送处理中断，参考章节 10.5；
- 2、 在发送二层处理中断时隙开关，下一个时隙准备发送使能，通过寄存器 LAYER2\_TXRX\_CTRL 控制实现；
- 3、 在开启下一个时隙发送使能的同时，配置发送帧类型和发送的数据内容，帧类型通过寄存器 LAYER2\_SEND\_TYPE 设置，发送内容通过二层发送 Ram 对应帧类型地址填写(详见章节 10.4.1)。

以上步骤是参考上图二层发送时序图，采用在二层发送处理中断准备发送内容。

此外发送内容准备也可以在时隙中断处理如下图所示。

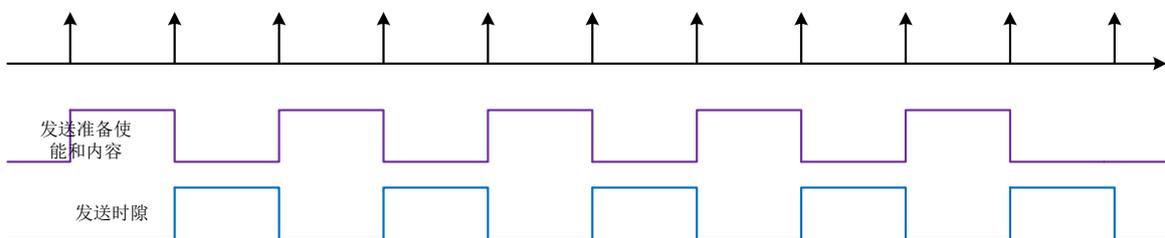


图 91 链路层发送时序图 2

## 10.7 链路层接收

### 10.7.1 概述

链路层支持所有类型的帧发送和接收。

### 10.7.2 功能描述

接收系统中断主要由解帧中断和后接入中断两类构成。其中解帧中断应用于各种数据控制帧、语音帧的接收处理，后接入中断应用于语音延迟接入或者语音加密密钥信息的传递。接收中断的状态处理流程如下图所示。

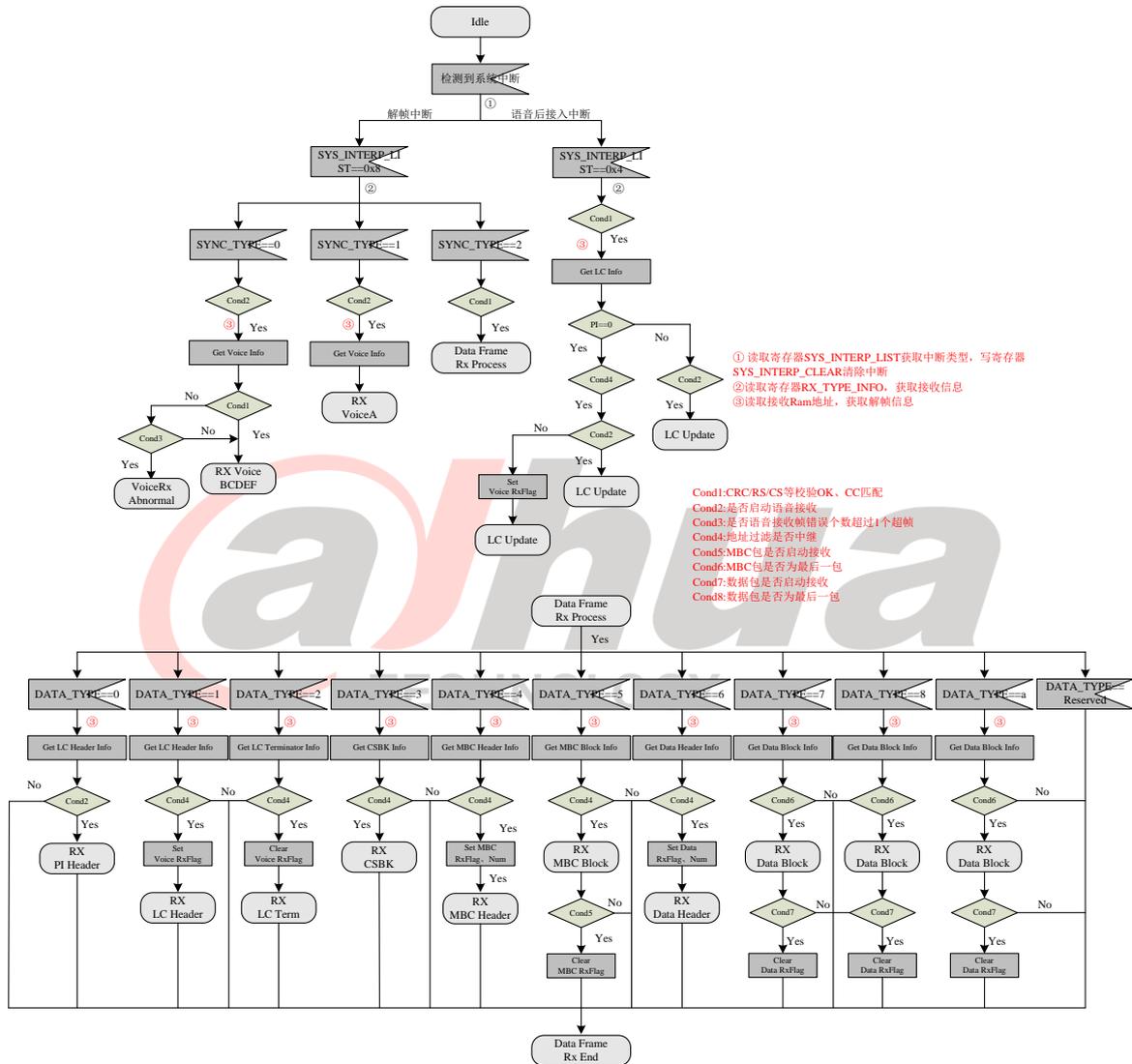


图 92 链路层接收状态流程图

### 10.7.3 接收应用

详细参考附录应用功能实例。

## 11 FM 应用

### 11.1 概述

HR\_C7000 的 FM 功能包含了模拟亚音 CTCSS、数字亚音 CDCSS、双音多频 DTMF 以及信令 SELCALL、SQL 等模式。各模式下，默认支持标准模式，保留有用户自定义可配置方式，以满足差异化的应用需求。

### 11.2 寄存器

#### 11.2.1 寄存器概览

寄存器表 29 FM 寄存器概述（基地址：0x1100\_0000）

偏移地址	名称	描述	页码
0x100	WORK_MODE	基带工作模式设置，设置为 FM 模式	
0x500	FM_BANDWIDTH	FM 宽、窄带设置	
0x504	FM_DEV_COEF	FM 音频发送限幅，以及音频调制频偏系数	
0x508	FM_SQL	只读，SQL 模式下的静噪输出值	
0x50C	SUBVOICE_COEF	亚音调制频偏，接收采用深度以及接收门限	
0x510	FM_MOD	FM 工作模式设置	
0x518	CTCSS_ADDR	CTCSS 亚音发送、接收地址	
0x51C	USER_DEFINE_CTC	用户自定义 CTCSS 发送频率	
0x520	CDCSS_ADDR	CDCSS 发送、接收地址	
0x524	SELCALL_TONE	信令发送频率	
0x528	TONE_SEND_CTRL	信令/DTMF 发送控制	
0x530	FM_FUNCTION_COEF1	接收解调系数 1	
0x534	FM_FUNCTION_COEF2	接收解调系数 2	
0x538	FM_FUNCTION_COEF3	接收解调系数 3	
0x53C	FM_FUNCTION_COEF4	接收解调系数 4	
0x540	FM_FUNCTION_COEF5	接收解调系数 5	
0x544	FM_FUNCTION_COEF6	接收解调系数 6	
0x548	FM_FUNCTION_COEF7	接收解调系数 7	
0x54C	FM_FUNCTION_COEF8	接收解调系数 8	
0x550	FM_TONE1	DTMF 发送频率 tone_1 及 tone_2	
0x554	FM_TONE2	DTMF 发送频率 tone_3 及 tone_4	
0x558	FM_TONE3	DTMF 发送频率 tone_5 及 tone_6	
0x55C	FM_TONE4	DTMF 发送频率 tone_7 及 tone_8	
0x560	FM_PTT	FM 发送 PTT	
0x564	FM_DECODE_FLAG1	信令解码输出码元	
0x568	FM_DECODE_FLAG2	DTMF 解码输出码元、CDCSS、CTCSS 检测到标志	
0x56C	FM_ADDR_SW	FM 收发音频数据地址空间选择控制	

## 11.2.2 寄存器描述

### 11.2.2.1 WORK\_MODE

偏移地址：0x100

复位方式：H

Bits	Access	Name	Default	Description
[31:9]	-	reserved		
[8]	RW	digital_analog_en	0x0	数字模拟同时检测开关，要求在默认配置数字模式下开启，在接收到数字信号或者模拟信号后，对应切换到数字模式或模拟模式。 1：表示数模同时接收开启控制使能； 0：表示只开启数字或者模拟接收，根据Bit7设置。
[7]	RW	modulator_mode	0x0	工作模式设置： 0：表示DMR数字模式； 1：表示FM模拟模式。
[6]	RW	is_tier1_mode	0x1	0：表示TierI模式； 1：表示TierII模式。
[5]	RW	is_continuemode	0x1	0：表示Continue模式； 1：表示TimeSlot模式。
[4:3]	RW	layermode	0x1	层次模式设置： 0：表示物理层模式； 1：表示二层模式； 2：表示三层模式。
[2]	RW	is_repeater	0x0	0：表示非中继模式； 1：表示中继模式。
[1]	RW	is_aligned	0x1	0：表示偏移模式（非中继模式下的偏移表示单频模式）； 1：表示对齐模式。
[0]	RW	work_slot_sel	0x1	三层非中继模式必须设置成1； 0：表示时隙1； 1：表示时隙2。

### 11.2.2.2 FM\_BANDWIDTH

偏移地址：0x500

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	-	reserved		
[15:12]	-	reserved		
[11:8]	-	reserved		
[7]	-	reserved		
[6]	RW	fmbpfon	0x1	FM带通滤波器开关：0表示关闭，1表示开启。

[5]	-	reserved		
[4]	RW	fmpreemphasis	0x1	FM预加重开关：0表示关闭，1表示开启。
[3]	RW	fmbandwidth	0x1	FM发送带宽选择：0表示12.5K，1表示25K。
[2]	RW	fmbandwidth_adj	0x1	FM 模式下，采用基带IQ及中频、中频IQ发送时，发送邻道抑制滤波器的带宽选择。0表示12.5KHz带宽，1表示25KHz带宽。
[1]	RW	fmbandwidth_r	0x1	FM接收带宽选择：0表示12.5K，1表示25K。
[0]	RW	fmbandwidth_ctc	0x1	亚音CTCSS接收带宽选择：0表示12.5K，1表示25K。

### 11.2.2.3 FM\_DEV\_COEF

偏移地址：0x504

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	tr_sig_lim	0x0f	FM发端限幅调制系数
[15:8]	RW	fm_dev_coef_r	0x10	FM收端调制频偏系数
[7:0]	RW	fm_dev_coef_t	0x10	FM发端调制频偏系数

### 11.2.2.4 FM\_SQL

偏移地址：0x508

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	-	reserved		
[15:0]	RO	fm_sql	0x0	FM 模式下，静噪信号计算值

### 11.2.2.5 SUBVOICE\_COEF

偏移地址：0x50c

复位方式：H

Bits	Access	Name	Default	Description
[31:28]	-	reserved		
[27:16]	RW	sample_size	0x7d0	FM各功能模块接收采样深度
[15:8]	RW	subvoice_dev_coef	0x28	亚音频调制频偏
[7:0]	RW	freq_amp_limit	0x10	FM各功能模块接收频率能量解调门限

### 11.2.2.6 FM\_MOD

偏移地址：0x510

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:16]	RW	inter_mask_limit	0xa	重复码元判决门限。

				1. 当前码元与上一次检测码元相同，并且更新的时间间隔大于该门限时，表示两者为独立存在的码元信息，允许系统中断输出； 2. 反之，如果时间间隔小于该门限，两者将被视为同一码元，系统中断被屏蔽。 3. 如果用户希望不区分是否为持续码元，凡是有解码就输出的话，可以将门限设置为0 4. 门限步进长度与DTMF解码时间保持一致（与采样深度sample_size相关），例如当门限设置为2时，相同码元更新时间间隔小于等于2倍的单位检测时间，被视为同一码元。
[15:14]	RW	xtcss_length	0x0	XTCSS 发送信令长度： 00:40ms; 01:60ms; 10:80ms; 11:100ms;
[13:12]	RW	ctcss_phase_revers e	0x1	CTCSS尾音翻转模式： 00: 尾音静噪功能关闭； 01: 尾音相位翻转240° ； 10: 尾音相位翻转180° ； 11: reserved
[11]	RW	phase_f_en_t	0x0	CDCSS发送相位正反，0:正； 1: 反
[10]	RW	phase_f_en_r	0x0	CDCSS接收相位正反，0:正； 1: 反
[9]	RW	mude_code_en_in	0x1	CDCSS 发送尾音关闭码使能，0关闭，1开启。
[8]	RW	eom_en	0x0	发送 XTCSS 结束信息开关，0关闭，1开启。
[7:5]	-	reserved		
[4]	RW	selcall_tone_mode	0x0	Selcall模式开关配置，高有效
[3]	RW	ctcss_mode	0x0	CTCSS模式开关配置，高有效
[2]	RW	cdcoss_mode	0x0	CDCSS模式开关配置，高有效
[1]	RW	dtmf_mode	0x0	DTMF模式开关配置，高有效
[0]	RW	xtcss_mode	0x0	XTCSS模式开关配置，高有效

### 11.2.2.7 CTCSS\_ADDR

偏移地址：0x518

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	-	reserved		
[15:8]	RW	ctcss_recv_addr	0x0	CTCSS亚音接收地址
[7:0]	RW	ctcss_send_addr	0x0	CTCSS亚音发送地址

### 11.2.2.8 USER\_DEFINE\_CTC

偏移地址：0x51c

复位方式：H

Bits	Access	Name	Default	Description
[31]	RW	user_define_en	0x0	用户自定义功能开启标志，“1”：开启，“0”：关闭
[30:21]	-	reserved		

[20:0]	RW	user_define_ctc_st ep	0xe14	CTCSS用户自定义频率设置，由公式： $(\text{待发送频率} / 32000\text{Hz采样时钟})$ 经过 $2^{21}$ 量化后得到。默认为55Hz
--------	----	--------------------------	-------	---

### 11.2.2.9 CDCSS\_ADDR

偏移地址：0x520

复位方式：H

Bits	Access	Name	Default	Description
[31:25]	-	reserved		
[24:16]	RW	cdcscs_recv_addr	0x0	CDCSS亚音接收地址
[15:9]	-	reserved		
[8:0]	RW	cdcscs_send_addr	0x0	CDCSS亚音发送地址

### 11.2.2.10 SELCALL\_TONE

偏移地址：0x524

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	-	reserved		
[15:0]	RW	selcall_tone	0x0fd9	2-Tone/5-Tone待发送的频率

### 11.2.2.11 TONE\_SEND\_CTRL

偏移地址：0x528

复位方式：H

Bits	Access	Name	Default	Description
[31:10]	-	reserved		
[9]	RW	selcall_send	0x0	信令发送使能，“1”：模块发送信令，“0”：信令发送间隔或停止发送 备注：该使能可以用作信令发送期间的侧音播放控制，“1”表示播放侧音；“0”表示关闭侧音。与侧音功能开关sidevoice_en“与逻辑”配合使用。
[8]	RW	selcall_finish	0x0	信令发送结束标志，由软件给出信令发送结束标志：“1”发送完毕，“0”表示未完成。 备注：“0”状态时，用于信令发送过程中，屏蔽语音信号。
[7]	RW	dtmf_send	0x0	码元发送使能，“1”：模块发送码元，“0”：码元间隔或停止发送； 备注：该使能可以用作DTMF序列发送期间的侧音播放控制使能，“1”表示播放侧音；“0”表示关闭侧音。与侧音功能开关sidevoice_en“与逻辑”配合使用。
[6]	RW	dtmf_finish	0x0	码元发送结束标志，对于“自动发码”方式，在码元发送结束后，由软件给出码元发送结束标志：“1”发送完毕，“0”表示未完成。

				备注：“0”状态时，用于DTMF码元发送过程中，屏蔽语音信号。
[5:4]	-	reserved		
[3:0]	RW	dtmf_send_tone	0x0	待发送的码元，由软件在dtmf_send有效之前配置

#### 11.2.2.12 XTCSS\_ADDR

偏移地址：0x52c

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:8]	RW	xtcss_tone	0x0fd9	XTCSS发送频率配置高8bit，(待发送频率/32000Hz采样时钟)经过 $2^{16}$ 量化后得到
[7:0]	RW	xtcss_recv_addr	0x0	XTCSS接收地址

#### 11.2.2.13 FM\_FUNCTION\_COEF1

偏移地址：0x530

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	coef	0x7ecd9d	FM亚音相关各功能接收解调系数1（697Hz）

#### 11.2.2.14 FM\_FUNCTION\_COEF2

偏移地址：0x534

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	coef_2nd	0x7e8a34	FM亚音相关各功能接收解调系数2（770Hz）

#### 11.2.2.15 FM\_FUNCTION\_COEF3

偏移地址：0x538

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	coef_3rd	0x7e368c	FM亚音相关各功能接收解调系数3（852Hz）

#### 11.2.2.16 FM\_FUNCTION\_COEF4

偏移地址：0x53c

复位方式：H

Bits	Access	Name	Default	Description

[31:24]	-	reserved		
[23:0]	RW	coef_4th	0x7dd245	FM亚音相关各功能接收解调系数4（941Hz）

#### 11.2.2.17 FM\_FUNCTION\_COEF5

偏移地址：0x540

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	coef_5th	0x7c690d	FM亚音相关各功能接收解调系数5（1209Hz）

#### 11.2.2.18 FM\_FUNCTION\_COEF6

偏移地址：0x544

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	coef_6th	0x7b9f03	FM亚音相关各功能接收解调系数6（1336Hz）

#### 11.2.2.19 FM\_FUNCTION\_COEF7

偏移地址：0x548

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	coef_7th	0x7aa7a8	FM亚音相关各功能接收解调系数7（1477Hz）

#### 11.2.2.20 FM\_FUNCTION\_COEF8

偏移地址：0x54c

复位方式：H

Bits	Access	Name	Default	Description
[31:24]	-	reserved		
[23:0]	RW	coef_8th	0x7979f5	FM亚音相关各功能接收解调系数8（1633Hz）

#### 11.2.2.21 FM\_TONE1

偏移地址：0x550

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	RW	tone_1	0x0593	DTMF模式发送频率配置697Hz;
[15:0]	RW	tone_2	0x0629	DTMF模式发送频率配置770Hz;

### 11.2.2.22 FM\_TONE2

偏移地址：0x554

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	RW	tone_3	0x06d1	DTMF模式发送频率配置852Hz;
[15:0]	RW	tone_4	0x0787	DTMF模式发送频率配置941Hz;

### 11.2.2.23 FM\_TONE3

偏移地址：0x558

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	RW	tone_5	0x09ac	DTMF模式发送频率配置1209Hz;
[15:0]	RW	tone_6	0x0ab0	DTMF模式发送频率配置1336Hz;

### 11.2.2.24 FM\_TONE4

偏移地址：0x55c

复位方式：H

Bits	Access	Name	Default	Description
[31:16]	RW	tone_7	0x0bd1	DTMF模式发送频率配置1477Hz;
[15:0]	RW	tone_8	0x0d10	DTMF模式发送频率配置1633Hz;

### 11.2.2.25 FM\_PTT

偏移地址：0x560

复位方式：H

Bits	Access	Name	Default	Description
[31:1]	-	reserved		
[0]	RW	fm_ptt	0x0	配置FM发送PTT指令，高有效

### 11.2.2.26 FM\_DECODE\_FLAG1

偏移地址：0x564

复位方式：H

Bits	Access	Name	Default	Description
[31:4]	RO		0x0	
[3:0]	RO	selcall_tone_recv	0x0	Selcall检测码元输出

### 11.2.2.27 FM\_DECODE\_FLAG2

偏移地址：0x568

复位方式：H

Bits	Access	Name	Default	Description
------	--------	------	---------	-------------

[31:26]	-	reserved		
[25]	RO	xtc_private_call_mode	0x0	XTCSS个呼标志寄存器
[24]	RO	xtc_all_call_mode	0x0	XTCSS全呼标志寄存器
[23:16]	RO	xtcss_type	0x0	XTCSS解调类型标志寄存器
[15:12]	-	reserved		
[11:8]	RO	dtmf_recv_tone	0x0	DTMF解码输出
[7:2]	-	reserved		
[1]	RO	cdcoss_voice_en	0x0	CDCSS检测到标志，开启扬声器
[0]	RO	ctcss_voice_en	0x0	CTCSS检测到标志，开启扬声器

### 11.2.2.28 FM\_ADDR\_SW

偏移地址：0x56c

复位方式：H

Bits	Access	Name	Default	Description
[31:2]	-	reserved		
[1:0]	RO	fm_addr_sw	0x0	FM收发音频数据地址空间选择控制： 1：表示CPU读或者写低地址数据； 2：表示CPU读或者写高地址数据。

## 11.3 中断向量

FM 模式下，定义 SYS\_INTERP 中断类型如下表所示。

表 56 FM 系统中断子中断类型列表

序号	名称	类型	描述
1	FM_TX_INTERP	电平触发，高有效	FM 发送数据处理中断： 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽； 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
2	FM_RX_INTERP	电平触发，高有效	FM 接收数据处理中断： 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽； 通过 SYS_INTERP_CLEAR 寄存器对应 bit 位写 1 清除。
3	FM_SIG_RX_INTERP	电平触发，高有效	模拟功能接收检测中断： 通过 SYS_INTERP_MASK 寄存器的对应 bit 位屏蔽； 通过 SYS_INTERP_CLEA

## 11.4 Memory 映射

基带 Ram 模块是 HR\_C7000 芯片基带数字功能主要的用户数据交互存储空间，根据不同的功能设计主要分为发送和接收 2 个 Ram，数据长度为 8，深度为 1200。

### 11.4.1 FM 发送 Ram

FM 发送 Ram 的基地址为 0x16000000，深度为 1200，下表定义了各个地址段内容分配。

表 57 FM 发送 Ram 分配

偏址	名称	描述
0x030~0x22f	FM_VOICE_1	共计 512bytes 数据，即 256 个 PCM 音频数据。FM 发送的语音数据存放地址 1，与地址 2 乒乓使用，低位地址存放高 8bitsPCM 音频数据
0x230~0x42f	FM_VOICE_2	共计 512bytes 数据，即 256 个 PCM 音频数据。FM 发送的语音数据存放地址 2，与地址 1 乒乓使用，低位地址存放高 8bitsPCM 音频数据

### 11.4.2 FM 接收 Ram

FM 接收 Ram 的基地址为 0x160004b0，深度为 1200，下表定义了各个地址段内容分配。

表 58 FM 接收 Ram 分配

偏址	名称	描述
0x030~0x22f	FM_VOICE_1	共计 512bytes 数据，即 256 个 PCM 音频数据。FM 接收的语音数据存放地址 1，与地址 2 乒乓使用，低位地址存放高 8bitsPCM 音频数据
0x230~0x42f	FM_VOICE_2	共计 512bytes 数据，即 256 个 PCM 音频数据。FM 接收的语音数据存放地址 2，与地址 1 乒乓使用，低位地址存放高 8bitsPCM 音频数据

## 11.5 SQL

HR\_C7000 FM 模式支持 SQL 静噪模式，FM 模式下系统默认支持 SQL 功能开启，CPU 通过读取 0x508[15:0]获得 SQL 输出值，该值越高表明当前的接收进程中，未包含有音频带内有效信号（或者即使存在，但是空中信号弱）。此时用户可以通过设置门限值，以抉择是否关闭音频播放通路（高于门限值即关闭）。

通过实际调试，可以设定多组门限参数，门限的高低亦即 SQL 静噪等级。

## 11.6 CTCSS

### 11.6.1 概述

HR\_C7000 FM 模式支持 CTCSS 发送和接收，支持标准频率和自定义频率的发送和接收功能。

### 11.6.2 功能描述

CTCSS 功能特点：

- 支持标准频率发送，共计 51 组；
- 支持自定义频率发送；
- 支持标准频率接收，共计 51 组；

- 支持自定义频率接收;
- 支持 CTCSS 尾音发送功能;

### 11.6.2.1 CTCSS 发送

标准模式下, HR\_C7000 支持 51 组 CTCSS 模拟亚音发送频率, 范围 62.5~254.1Hz, 发送地址与标准亚音频率的映射关系如下表所示。

表 59 CTCSS 发送地址与标准亚音频率的映射关系列表

CTCSS 发送地址	亚音频率 (Hz)	正弦信号相位 Step (十进制)
0x01	67	4391
0x02	71.9	4712
0x03	74.4	4876
0x04	77	5046
0x05	79.7	5223
0x06	82.5	5407
0x07	85.4	5597
0x08	88.5	5800
0x09	91.5	5997
0x0a	94.8	6213
0x0b	97.4	6383
0x0c	100	6554
0x0d	103.5	6783
0x0e	107.2	7025
0x0f	110.9	7268
0x10	114.8	7524
0x11	118.8	7786
0x12	123	8061
0x13	127.3	8343
0x14	131.8	8638
0x15	136.5	8946
0x16	141.3	9260
0x17	146.2	9581
0x18	151.4	9922
0x19	156.7	10269
0x1a	162.2	10630
0x1b	167.9	11003
0x1c	173.8	11390
0x1d	179.9	11790
0x1e	186.2	12203
0x1f	192.8	12635
0x20	203.5	13337
0x21	210.7	13808
0x22	218.1	14293
0x23	225.7	14791

0x24	233.6	15309
0x25	241.8	15847
0x26	250.3	16404
0x27	69.3	4542
0x28	62.5	4096
0x29	159.8	10473
0x2a	165.5	10846
0x2b	171.3	11226
0x2c	177.3	11620
0x2d	183.5	12026
0x2e	189.9	12445
0x2f	196.6	12884
0x30	199.5	13074
0x31	206.5	13533
0x32	229.1	15014
0x33	254.1	16653

对于标准模式，用户只需要关心期望发送的亚音的频率所对应的 CTCSS 发送地址，通过寄存器 CTCSS\_ADDR 配置实现。

对于非标准应用情景，用户需要自定义发送频率时，需要软件计算对应频率的正弦信号相位 Step（如上表第 3 列）配置到寄存器 USER\_DEFINE\_CTC，其计算公式如下：

$$\text{user\_define\_ctc\_step} = \text{round}(\text{发送频率} * 65.536)$$

需要注意的是，计算结果为十进制数，需要转换为二进制数配置到寄存器 USER\_DEFINE\_CTC 的低 21 位，若不足 21bit，则高位补零。

### 11.6.2.2 CTCSS 接收

标准模式下，HR\_C7000 支持 51 组 CTCSS 模拟亚音接收频率，范围 62.5~254.1Hz，接收地址与信号相关检测系数 coef 的映射关系如下表所示：

表 60 CTCSS 接收地址与信号相关检测系数 coef 的映射关系列表

CTCSS 接收地址	亚音频率 (Hz)	检测系数 coef(十进制)
0x01	67	8376996
0x02	71.9	8375236
0x03	74.4	8374291
0x04	77	8373273
0x05	79.7	8372179
0x06	82.5	8371005
0x07	85.4	8369746
0x08	88.5	8368352
0x09	91.5	8366956
0x0a	94.8	8365367
0x0b	97.4	8364075
0x0c	100	8362749

0x0d	103.5	8360908
0x0e	107.2	8358893
0x0f	110.9	8356808
0x10	114.8	8354534
0x11	118.8	8352119
0x12	123	8349496
0x13	127.3	8346716
0x14	131.8	8343704
0x15	136.5	8340448
0x16	141.3	8337005
0x17	146.2	8333368
0x18	151.4	8329373
0x19	156.7	8325158
0x1a	162.2	8320632
0x1b	167.9	8315778
0x1c	173.8	8310577
0x1d	179.9	8305013
0x1e	186.2	8299066
0x1f	192.8	8292619
0x20	203.5	8281692
0x21	210.7	8274010
0x22	218.1	8265839
0x23	225.7	8257157
0x24	233.6	8247820
0x25	241.8	8237792
0x26	250.3	8227037
0x27	69.3	8376186
0x28	62.5	8378504
0x29	159.8	8322626
0x2a	165.5	8317842
0x2b	171.3	8312803
0x2c	177.3	8307408
0x2d	183.5	8301640
0x2e	189.9	8295479
0x2f	196.6	8288805
0x30	199.5	8285845
0x31	206.5	8278523
0x32	229.1	8253177
0x33	254.1	8222111

对于标准模式，用户只需要关心期望检测频率所对应的 CTCSS 接收地址，即通过配置寄存器 CTCSS\_ADDR 实现。

对于非标准应用情景，用户需要自定义接收频率时，需要软件计算对应频率的检测系数 Coef（如上图第 3 列）配置到寄存器 FM\_FUNCTION\_COEF1，其计算公式如下：

$$\text{Coef} = 2 * \text{COS}(2 * \pi * \text{待检测频率} / 8000) * 2^{22}$$

需要注意的是，在使用科学计算器时，余弦 cos 求解的对象为弧度单位，而非角度单位。计算结果为十进制数，需要转换为二进制数配置到寄存器 FM\_FUNCTION\_COEF1 的低 24 位，若不足 24bit，则高位补零。

### 11.6.3 工作方式

#### 11.6.3.1 CTCSS 发送

如下图描述了 CTCSS 发送的状态流程图。

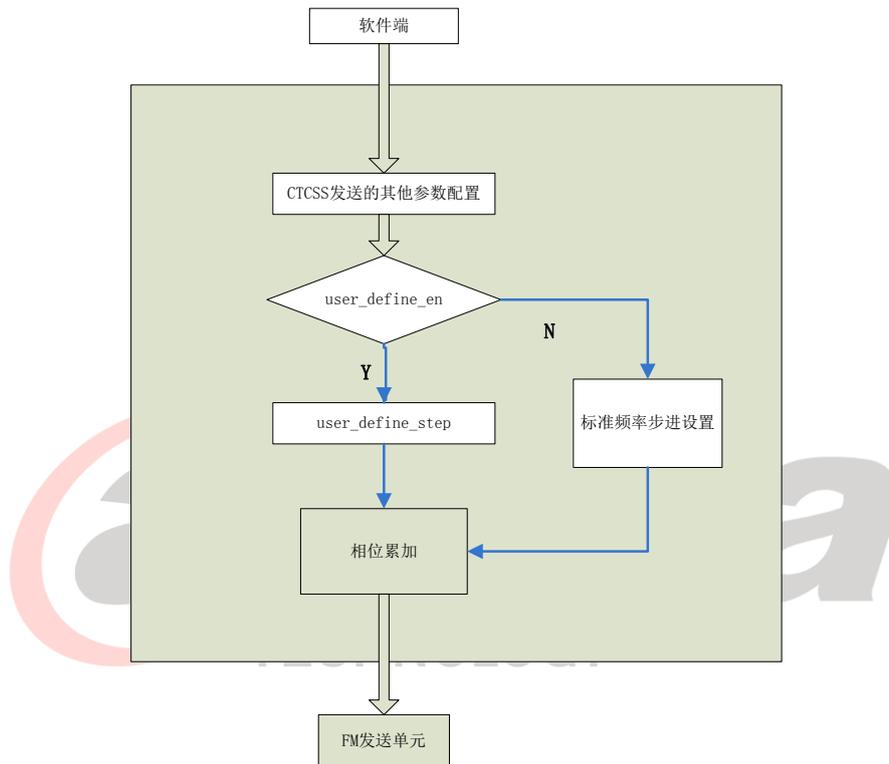


图 93 CTCSS 发送状态流程图

CTCSS 发送配置步骤如下，标准频率和自定义频率二选一：

- 配置寄存器 FM\_MOD[3]=1 ----- 选择 CTCSS 模式；
- 配置寄存器 FM\_MOD[13:11]，设置 CTCSS 尾音翻转模式；
- 若发送标准频率，配置寄存器 CTCSS\_ADDR[7:0]，对应地址参考 CTCSS 发送地址与标准亚音频率的映射关系列表；
- 若发送自定义频率，配置寄存器 USER\_DEFINE\_CTC[31]为“1”开启 CTCSS 自定义配置使能；
- 若发送自定义频率，计算正弦信号相位 Step 值配置到寄存器 USER\_DEFINE\_CTC [20:0]；
- 配置寄存器 SUBVOICE\_COEF[15:8] ----- 调整模拟亚音发送调制频偏（通过综合测试仪等手段观察。通常窄带发送为 350Hz，宽带为 600Hz 左右），调节精度在 10Hz 左右；
- 
- 配置寄存器 FM\_PTT [0] = 1，PTT 有效，开始发送；

- 配置寄存器 FM\_PTT [0] = 0, PTT 失效, 结束发送, 发送将在 CTCSS 尾音发送完成后结束。

### 11.6.3.2 CTCSS 接收

CTCSS 接收配置步骤如下, 标准频率和自定义频率二选一:

- 配置寄存器 FM\_MOD[3]=1 ----- 选择 CTCSS 模式;
- 若接收标准频率, 配置寄存器 CTCSS\_ADDR[15:8], 对应地址参考 CTCSS 接收地址与信号相关检测系数 coef 的映射关系列表;
- 若接收自定义频率, 配置寄存器 USER\_DEFINE\_CTC[31]为“1”开启 CTCSS 自定义配置使能;
- 若接收自定义频率, 计算检测系数 Coef 值配置到寄存器 FM\_FUNCTION\_COEF1[23:0];
- 配置寄存器 SUBVOICE\_COEF[7:0] ----- 调整 CTCSS 信号检测门限;
- 如果接收到的亚音频率与接收设置的频率匹配后, 软件将检测到 FM 亚音检测到的系统中断, 读取寄存器 FM\_DECODE\_FLAG2[0], 通常 1 表示 CTCSS 亚音接收开始标志, 0 表示 CTCSS 亚音接收结束标志。

## 11.7 CDCSS

### 11.7.1 概述

HR\_C7000 FM 模式支持 CDCSS 发送和接收过程。

### 11.7.2 功能描述

CDCSS 功能特点:

- 支持标准码和非标准码发送;
- 支持标准码和非标准码接收;
- 支持尾音发送。

### 11.7.3 工作方式

#### 11.7.3.1 CDCSS 发送

- 配置寄存器 FM\_MOD [2]=1 ----- 选择 CDCSS 模式;
- 默认支持标准 CDCSS 亚音频配置, 支持正向、反向设置, 标准发送 DCS 码如下表所示

OFF	053N	131N	205 N	261 N	343 N	432 N	516 N	645 N
017N	054N	132N	212 N	263 N	346 N	445 N	523 N	654 N
023N	065N	134 N	223 N	265 N	351 N	446 N	526 N	662 N
025N	071N	143 N	225 N	266 N	356 N	452 N	532 N	664 N
026N	072N	145 N	226 N	271 N	364 N	454 N	546 N	703 N
031N	073N	152 N	243 N	274 N	365 N	455 N	565 N	712 N
032N	074N	155 N	244 N	306 N	371 N	462 N	606 N	723 N
036N	114N	156 N	245 N	311 N	411 N	464 N	612 N	731 N
043N	115N	162 N	246 N	315 N	412 N	465 N	624 N	732 N
047N	116N	165 N	251 N	325 N	413 N	466 N	627 N	734 N
050N	122N	172 N	252 N	331 N	423 N	503 N	631 N	743 N
051N	125N	174 N	255 N	332 N	431 N	506 N	632 N	754 N

- 配置寄存器 CDCSS\_ADDR[8:0] ----- 设置 CDCSS 发送码元，例如 CDCSS\_ADDR [8:0]=“001001100”对应上表中的 114；
- 支持非标准模式 CDCSS 码元发送，例如，上表中没有 173 这一选项，但是依然可以通过配置 CDCSS\_ADDR[8:0]=“001111011”实现发送。
- 配置寄存器 SUBVOICE\_COEF[15:8] ----- 调整数字亚音发送调制频偏（通过综合测试仪等手段观察。通常窄带发送为 350Hz，宽带为 600Hz 左右）。
- 支持发送 134.4Hz 尾音关闭单音。

### 11.7.3.2 CDCSS 接收

- 配置寄存器 FM\_MOD [2]=1 ----- 选择 CDCSS 模式；
- 默认支持标准 CDCSS 亚音频配置，支持正向、反向设置。
- 配置 MODEM 寄存器 CDCSS\_ADDR[24:16] ----- 设置 CDCSS 接收码元，例如 CDCSS\_ADDR[24:16]=“001001100”对应上表中的 114；
- 支持非标准模式 CDCSS 码元接收，例如，上表中没有 173 这一选项，但是依然可以通过配置 CDCSS\_ADDR[24:16]=“001111011”实现检测。
- 默认开启 134.4Hz 尾音静噪功能，通过配置 FM\_FUNCTION\_COEF1[23:0] = 0x7f499d(设置静噪检测频率为 134.4Hz)，配置寄存器 SUBVOICE\_COEF[7:0] ----- 调整 CDCSS 尾音检测门限。

## 11.8 DTMF

### 11.8.1 概述

HR\_C7000 FM 模式支持 DTMF 信令的发送和接收过程。

### 11.8.2 功能描述

DTMF 功能特点：

- 支持标准和非标准 DTMF 信令发送；
- 支持标准和非标准 DTMF 信令接收；

- 支持 DTMF 发送侧音播放。

DTMF 信号由 4 组高频信号和 4 组低频信号，一共有 16 种方式组合产生。其中低频信号比高频信号的幅值低 2.5dB。高频信号与低频信号叠加后经调制频偏系数加权，由 FM 调制后输出。

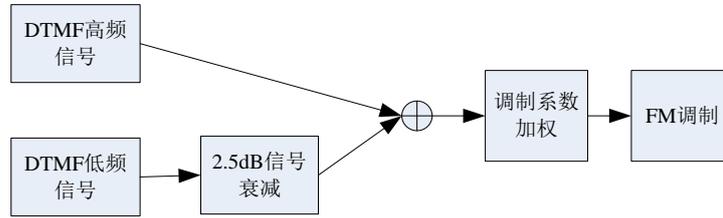


图 94 DTMF 发送框图

常用标准 DTMF 码元频率如下图所示：

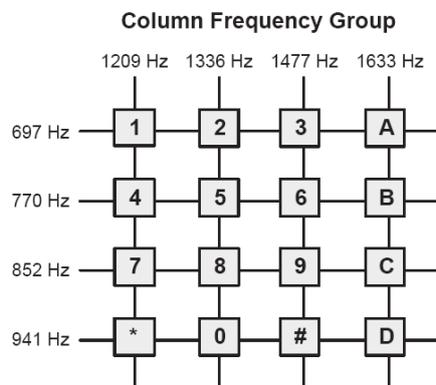


图 95 DTMF 码元频率对应关系

通常 DTMF 信号发送于音频信号的开始之前。

DTMF 的解调过程即分析空中信号频率分布，根据 DTMF 组合方式逆向解码。计算空中信号在 8 组频率下了频响幅值，分别选取高频部分中的最大幅值和低频部分中的最大幅值。这两者的组合即可以确定 DTMF 码。

### 11.8.3 工作方式

#### 11.8.3.1 DTMF 发送

- 配置寄存器 FM\_MOD[1]=1 ----- 选择 DTMF 模式；
- 配置寄存器 SUBVOICE\_COEF[15:8] ----- 调整 DTMF 发送调制频偏（通过综合测试仪等手段观察。通常在 1.8KHz 左右，选择连续无间断的 DTMF 发送码测试）。
- 配置 DTMF 码元所对应的频率
- 配置 FM\_TONE1[31:16]=16'h0593 （频率 697Hz 对应相位值）；
- 配置 FM\_TONE1[15:0]=16'h0629 （频率 770Hz 对应相位值）；
- 配置 FM\_TONE2[31:16]=16'h06D1 （频率 852Hz 对应相位值）；
- 配置 FM\_TONE2[15:0]=16'h0787 （频率 941Hz 对应相位值）；
- 配置 FM\_TONE3[31:16]=16'h09AC （频率 1209Hz 对应相位值）；
- 配置 FM\_TONE3[15:0]=16'h0AB0 （频率 1336Hz 对应相位值）；
- 配置 FM\_TONE4[31:16]=16'h0BD1 （频率 1477Hz 对应相位值）；
- 配置 FM\_TONE4[15:0]=16'h0D10 （频率 1633Hz 对应相位值）。

其计算公式为： $step = round(发送频率 * 2.048)$

DTMF 发送时序简图如下：

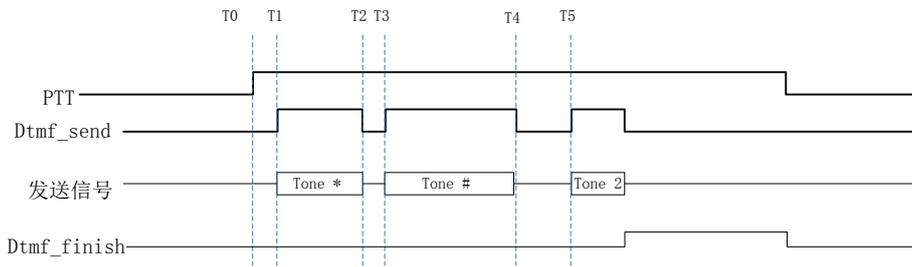


图 96 DTMF 发送时序图

- 配置 FM\_PTT[0] = 1，PTT 有效，建立载波（图中 T0 时刻）。
  - 配置 TONE\_SEND\_CTRL[6] = 0，确保在发送 DTMF 信令的过程中，屏蔽语音信号。
  - 配置 AUDIO\_CONTROL[3]，选择开启或关闭侧音播放功能（“1”为开启）。
  - 配置 TONE\_SEND\_CTRL[3:0]，设置 DTMF 待发送码元。
  - 配置 TONE\_SEND\_CTRL[7]=1，发送当前码元信号（图中 T1 时刻）。
  - 配置 TONE\_SEND\_CTRL[7]=0，结束当前码元发送（图中 T2 时刻）。
  - 配置 TONE\_SEND\_CTRL[3:0]，设置下一组 DTMF 码元。
  - 配置 TONE\_SEND\_CTRL[7]=1，发送第二组码元信号（图中 T3 时刻）。
  - 配置 TONE\_SEND\_CTRL[7]=0，结束第二组码元发送（图中 T4 时刻）。
- 依此类推。
- 所以码元发送完毕后，配置 TONE\_SEND\_CTRL[6]=1，开启语音信号（图中 Dtmf\_finish 上升沿时刻）。

### 11.8.3.2 DTMF 接收

- 配置寄存器 FM\_MOD[1]=1 ----- 选择 DTMF 模式；
  - 配置 SUBVOICE\_COEF[27:16]，设置 DTMF 解调时间，标准模式为 12'd1056(32KHz, 33ms,  $round(32000 * 0.033)$ )，支持 30ms 非标准模式（建议固定配置，解调时间设置在 33ms 附近，对于检测信号远大于 30ms 的情况，DTMF 检测模块会以连续的等间隔中断输出检测码元，或者在设置了中断间隔屏蔽寄存器的情况下，仅输出单次码元）。
  - 配置寄存器 SUBVOICE\_COEF[7:0] ----- 调整 DTMF 信号检测门限；
  - 配置 FM\_FUNCTION\_COEF1[23:0] = 24'h7ecd9d （频率 697Hz 对应解调值）；
  - 配置 FM\_FUNCTION\_COEF2[23:0] = 24'h7e8a34 （频率 770Hz 对应解调值）；
  - 配置 FM\_FUNCTION\_COEF3[23:0] = 24'h7e368c （频率 852Hz 对应解调值）；
  - 配置 FM\_FUNCTION\_COEF4[23:0] = 24'h7dd245 （频率 941Hz 对应解调值）；
  - 配置 FM\_FUNCTION\_COEF5[23:0] = 24'h7c690d （频率 1209Hz 对应解调值）；
  - 配置 FM\_FUNCTION\_COEF6[23:0] = 24'h7b9f03 （频率 1336Hz 对应解调值）；
  - 配置 FM\_FUNCTION\_COEF7[23:0] = 24'h7aa7a8 （频率 1477Hz 对应解调值）；
  - 配置 FM\_FUNCTION\_COEF8[23:0] = 24'h7979f5 （频率 1633Hz 对应解调值）；
- 计算公式： $2 * \cos(2 * \pi * 待检测频率 / 32000) * 2^{\wedge} 22$ ；
- 解码结果中，如果相邻的检测结果为同一码元，通过两次检测的时间间隔判断其为同一码元或者为相互独立的码元。配置 FM\_MOD[23:16]，设置最小间隔门限。例如 FM\_MOD[23:16]=5，则表示如果同一码元检测结果，如果出现的时间在 165ms（33ms\*5）以内，

则被认为是上一组码元的延续，芯片不输出中断标志。反之相邻间隔大于该值，则视其为两次独立的解码。

- 每接收到芯片输出的中断信息，读取 FM\_DECODE\_FLAG2[11:8]，获取码元信息。

## 11.9 SELCALL

### 11.9.1 概述

HR\_C7000 FM 模式支持 2-tone 和 5-tone 信令的发送和接收过程。

### 11.9.2 功能描述

SELCALL 功能特点：

- 支持 2-tone 信令发送；
- 支持 5-tone 信令发送；
- 支持 2-tone 信令接收；
- 支持 5-tone 信令接收；
- 支持 2/5-tone 发送侧音播放。

### 11.9.3 工作方式

#### 11.9.3.1 SELCALL 发送

- 配置寄存器 FM\_MOD[4]=1 ----- 选择 SELCALL 信令模式（2/5 音）；
- 配置寄存器 SUBVOICE\_COEF[15:8] ----- 调整信令发送调制频偏（通过综合测试仪等手段观察。通常在 1.8KHz 左右，选择连续无间断的单音发送码测试）。

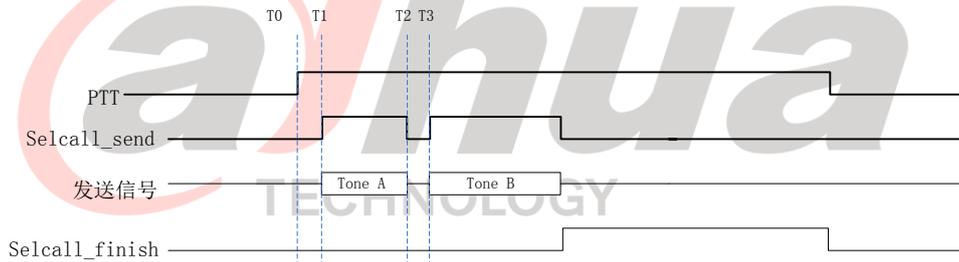


图 97 SELCALL 发送时序图

- 配置 FM\_PTT[0] = 1, PTT 有效，建立载波（图中 T0 时刻）。
  - 配置 TONE\_SEND\_CTRL[8] = 0, 确保在发送 SELCALL 信令的过程中，屏蔽语音信号。
  - 配置 AUDIO\_CONTROL[3], 选择开启或关闭侧音播放功能（“1”为开启）。
  - 配置 SELCALL\_TONE [15:0], 设置信令发送频率；计算公式  $step = round(发送频率 * 2.048)$  例如 16'd4057 对应的频率为 1981Hz 单音。
  - 配置 TONE\_SEND\_CTRL[9]=1, 发送当前 SELCALL 信令（图中 T1 时刻）。
  - 配置 TONE\_SEND\_CTRL[9]=0, 结束当前信令发送（图中 T2 时刻）。
  - 配置 SELCALL\_TONE [15:0], 设置下一组信令发送频率
  - 配置 TONE\_SEND\_CTRL[9]=1, 发送第二组信令（图中 T3 时刻）。
  - 配置 TONE\_SEND\_CTRL[9]=0, 结束第二组信令发送。
- 依此类推。
- 所以码元发送完毕后，配置 TONE\_SEND\_CTRL[8]=1, 开启语音信号（图中 Selcall\_finish 上升沿时刻）。

### 11.9.3.2 SELCALL 接收

- 配置寄存器 FM\_MOD[4]=1 ----- 选择 SELCALL 信令模式 (2/5 音);
- 配置 0x50C[27:16], 设置 SELCALL 解调时间, 标准模式为 12'd1056(32KHz, 33ms, round(32000\*0.033)), 支持压缩为 20ms (建议固定配置, 解调时间设置在 33ms 附近, 对于检测信号远大于 20ms 的情况, SELCALL 检测模块会以连续的等间隔中断输出检测码元)。
- 配置寄存器 SUBVOICE\_COEF[7:0] ----- 调整 SELCALL 信号检测门限。
- 配置 FM\_FUNCTION\_COEF1~8----- 最高支持 8 组频率对比检测。
- 每接收到芯片输出的中断信息, 读取 FM\_DECODE\_FLAG1[3:0], 获取 SELCALL 信令码元信息。

## 12 术语

AHB	由英国 ARM 公司开发、开放的高性能、用于连接高速外设的总线架构
APB	由英国 ARM 公司开发、开放的高性能、用于连接低速外设的总线架构
DMR	一种数字对讲标准协议
FM	一种模拟对讲模式
CK803S	杭州中天处理器

## 13 缩略语

### B

BLC	Black Level Correction	黑电平校正
-----	------------------------	-------

### D

DAC	Digital to Analog Converter	数字模拟转换器
DGC	Digital Gain Control	数字增益控制

### G

GPIO	General Purpose Input Output	通用输入输出
------	------------------------------	--------

### I

I2C	The Inter-Integrated Circuit	一种串行总线协议标准
IRQ	Interrupt Request	中断相应

### J

JTAG                      Joint Test Action Group                      联合测试组

## P

PLL                      Phase-Loced Loop                      锁相环  
 POR                      Power On Reset                      上电自复位  
 PWM                      Pulse Width Modulation                      脉宽调制

## R

RISC                      Reduced Instruction Set Computer                      精简指令集计算机  
 UART                      Universal Asynchronous Receiver Transmitter                      通用异步收发器

## S

SPI                      Serial Peripheral Interface                      串行外围设备接口

## 14 附录 A：数字组呼应用

### 14.1 功能描述

如下图定义了常规系统下语音业务时序图，包括语音帧头、语音的超帧、语音帧尾。在常规系统下，定义的语音帧头 72 比特 LC 信息(参考 9.6.2.1)、语音 EMB 区域嵌入在 BCDE 四组的嵌入式 EMBLC 以及语音帧尾定义的 72 比特 LC 信息是完全相同的定义。

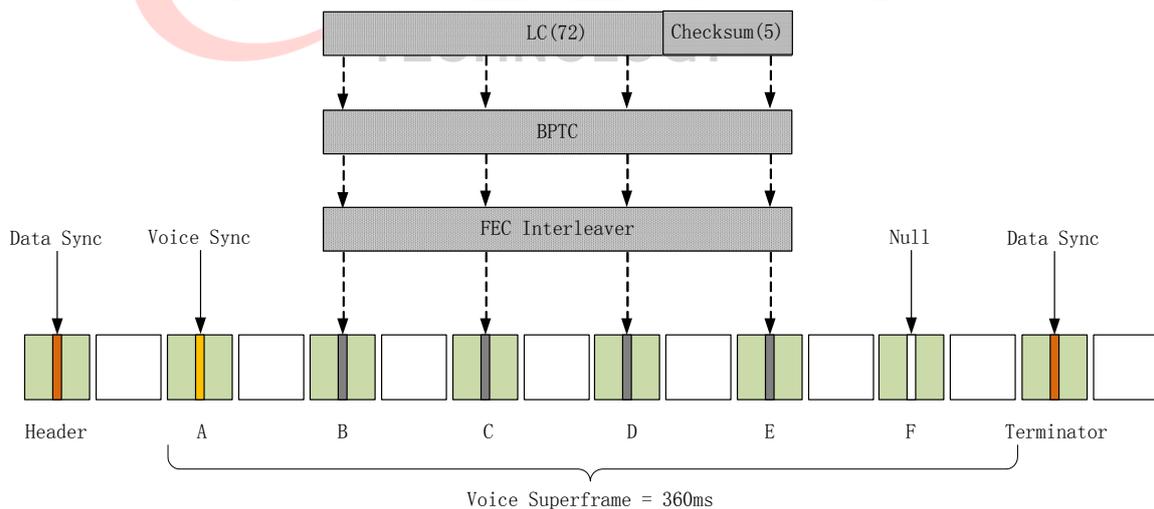


图 98 语音业务结构图

## 14.2 组呼发送

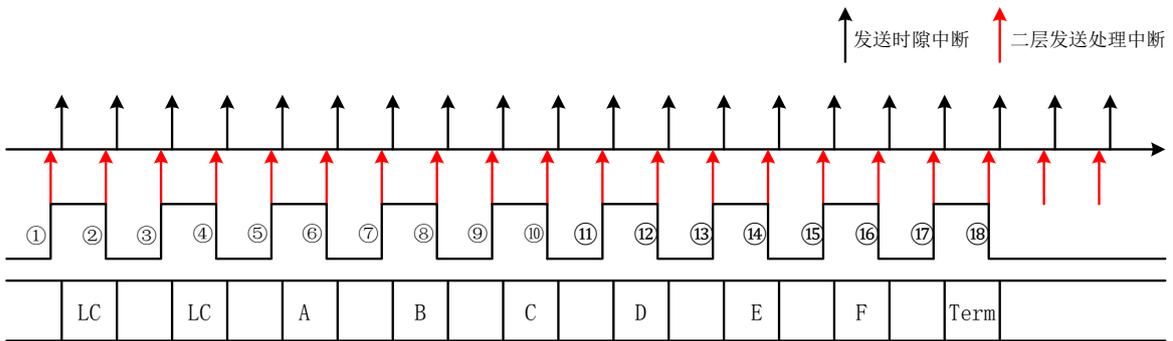


图 99 组呼发送业务时序图 1

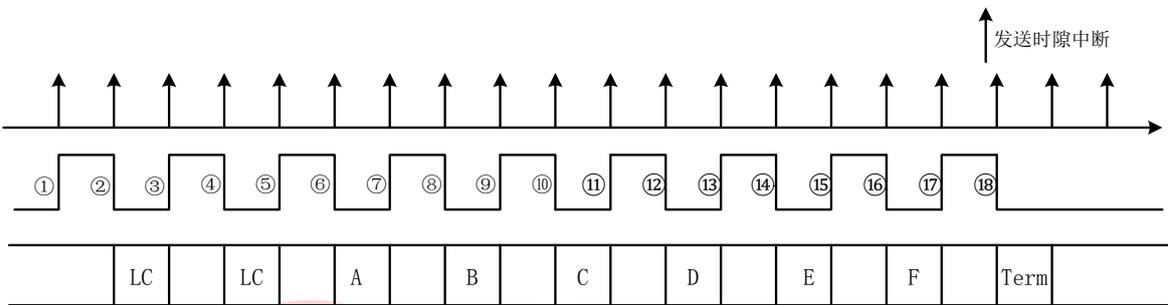


图 100 组呼发送业务时序图 2

参考组呼发送时序图 1、2 两种方式，组呼发送步骤如下（已知发送时隙建立）：

步骤①：

- a) 准备语音 LC72bits 的组呼业务信息，将 72Bits 信息写入发送 Ram 地址 0x16000000~0x16000008;
- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x10 表示发送语音帧头 LC;
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;

步骤③：

重复步骤①，表示发送第二个语音帧头，实际若要需求可以再发送一个语音帧头。

步骤⑤：

- a) 准备语音 LC72bits 的组呼业务信息，将 72Bits 信息写入发送 Ram 地址 0x16000000~0x16000008，用于语音超帧的 EMB 区域编码填充;
- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x08 表示发送语音帧 A;
- c) 如果语音数据选择从发送 Ram 获取，则需要向发送 Ram 地址 0x16000030~0x1600004a 写入 27bytes 的语音数据流。
- d) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;

步骤⑦：

- a) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x19 表示发送语音帧 B;
- b) 如果语音数据选择从发送 Ram 获取，则需要向发送 Ram 地址 0x16000030~0x1600004a 写入 27bytes 的语音数据流。
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;

步骤⑨:

- a) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x2B 表示发送语音帧 C;
- b) 如果语音数据选择从发送 Ram 获取, 则需要向发送 Ram 地址 0x16000030~0x1600004a 写入 27bytes 的语音数据流。
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;

步骤⑪:

- a) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x3B 表示发送语音帧 D;
- b) 如果语音数据选择从发送 Ram 获取, 则需要向发送 Ram 地址 0x16000030~0x1600004a 写入 27bytes 的语音数据流。
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;

步骤⑬:

- a) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x4A 表示发送语音帧 E;
- b) 如果语音数据选择从发送 Ram 获取, 则需要向发送 Ram 地址 0x16000030~0x1600004a 写入 27bytes 的语音数据流。
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;

步骤⑮:

- a) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x50 表示发送语音帧 F;
- b) 语音 F 帧 EMB 区域的填充可以通过寄存器 VOICE\_EMB\_CTRL 控制, 对应内容存放在发送 Ram 地址区间 0x16000029~0x1600002C;
- c) 如果语音数据选择从发送 Ram 获取, 则需要向发送 Ram 地址 0x16000030~0x1600004a 写入 27bytes 的语音数据流;
- d) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;

步骤⑰:

- a) 准备语音 LC72bits 的组呼业务信息, 将 72Bits 信息写入发送 Ram 地址 0x16000000~0x16000008;
- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x20 表示发送语音帧尾 LC;
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;

其余步骤②④⑥⑧⑩⑫⑭⑯⑱, 均为配置寄存器 LAYER2\_TXRX\_CTRL 为 0x0 或者 0x40 表示下一个时隙发送关闭, 接收开启或者关闭。

### 14.3 组呼接收

组呼接收业务的启动, 通过检测到语音帧头的系统中断或者语音语音后接入的系统中断, 且都满足组呼地址匹配的情况下开启时隙接收语音业务, 对应的设计时序如下图所示。

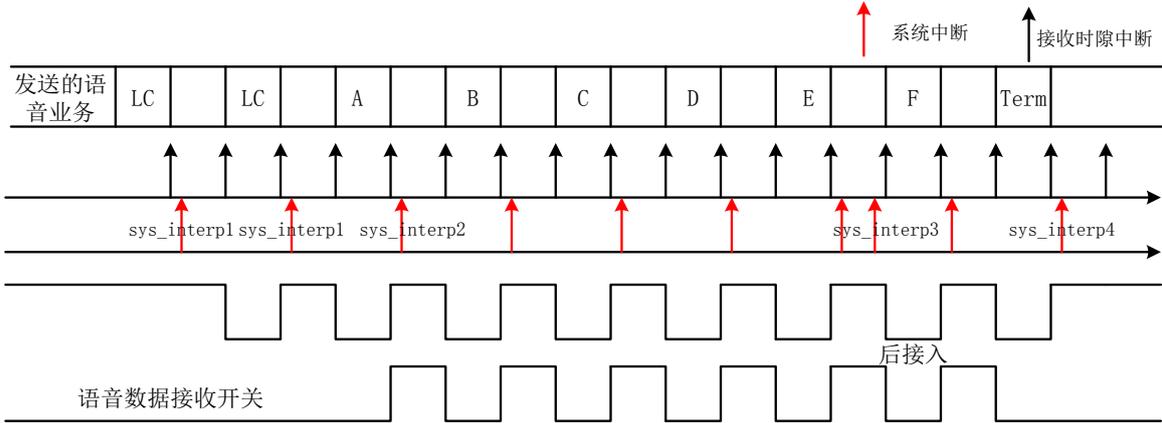


图 101 组呼接收业务时序图

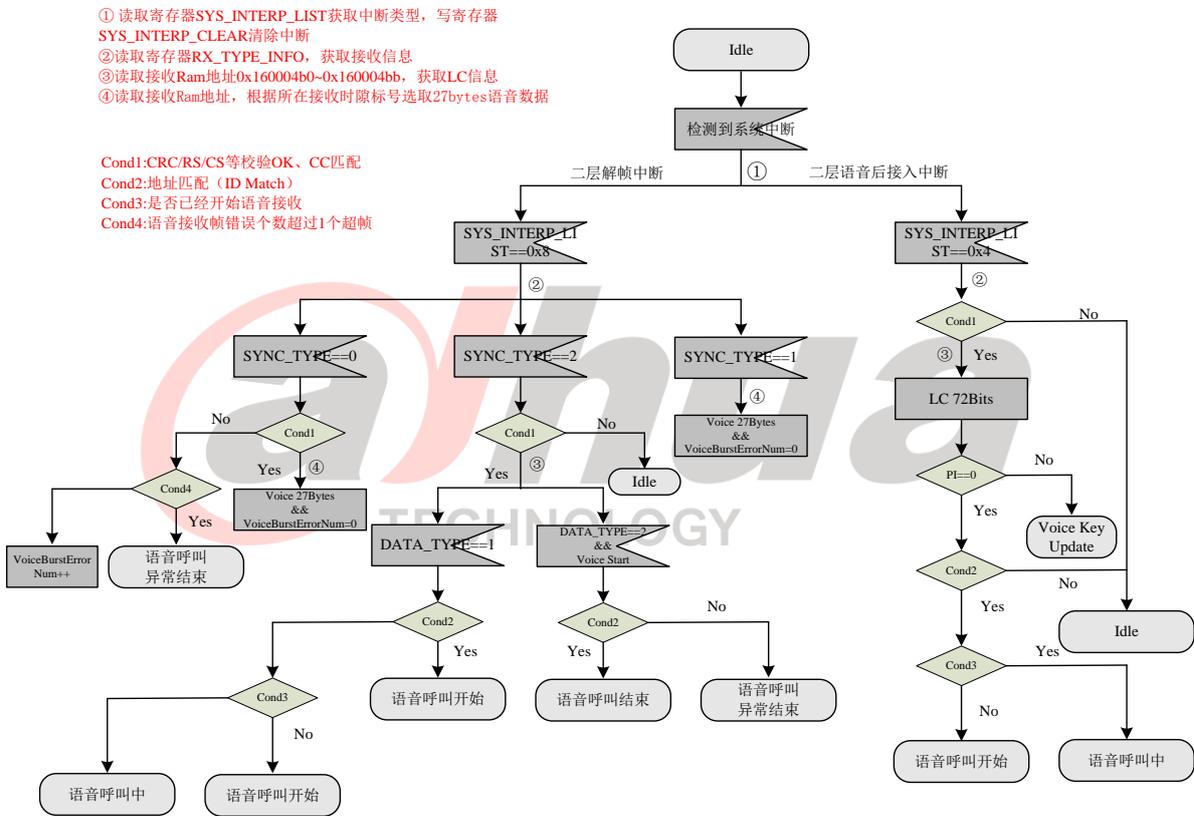


图 102 组呼接收状态流程图

组呼业务接收步骤如下：

- 1、接收机盲收检测到解帧系统中断，读取寄存器RX\_TYPE\_INFO，判断是否接收正确的语音帧头LC信息，然后读取接收Ram地址0x160004b0~0x160004bb，判断接收到的组地址是否匹配；
- 2、若地址匹配，则开始启动时隙接收语音业务，开始时隙开关寄存器LAYER2\_TXRX\_CTRL寄存器；
- 3、当时隙中断处检测接收到语音帧A信息，则开始启动时隙接收语音数据内容，通过寄

- 寄存器LAYER2\_TXRX\_CTRL控制；
- 4、当检测到语音后接入系统中断时，读取接收Ram地址0x160004b0~0x160004bb，判断接收到的组地址是否匹配：若当前无语音业务，后接入信息正确且地址匹配则可以启动时隙接收语音业务（注意：语音数据的接收需要等到下一超帧的帧A启动）；若接收错误在已有语音业务情况下可以关闭语音接收提示语音异常结束；
  - 5、当检测到语音帧尾系统中断，读取寄存器RX\_TYPE\_INFO，判断是否接收正确的语音帧头LC信息，然后读取接收Ram地址0x160004b0~0x160004bb，判断接收到的组地址是否匹配；

## 15 附录 B：数字短信应用

### 15.1 功能描述

常规短信业务功能中，主要存在两种短信息的功能类型，一种是非确认短信，另一种是确认短信。其中定义组地址的短信默认为非确认短信数据帧格式，定义个地址的短信可以由CPS设置确认还是非确认短信，下文主要介绍常规模式下两种短信业务应用。如下图定义了常规数字对讲系统IP类型短信业务框图。

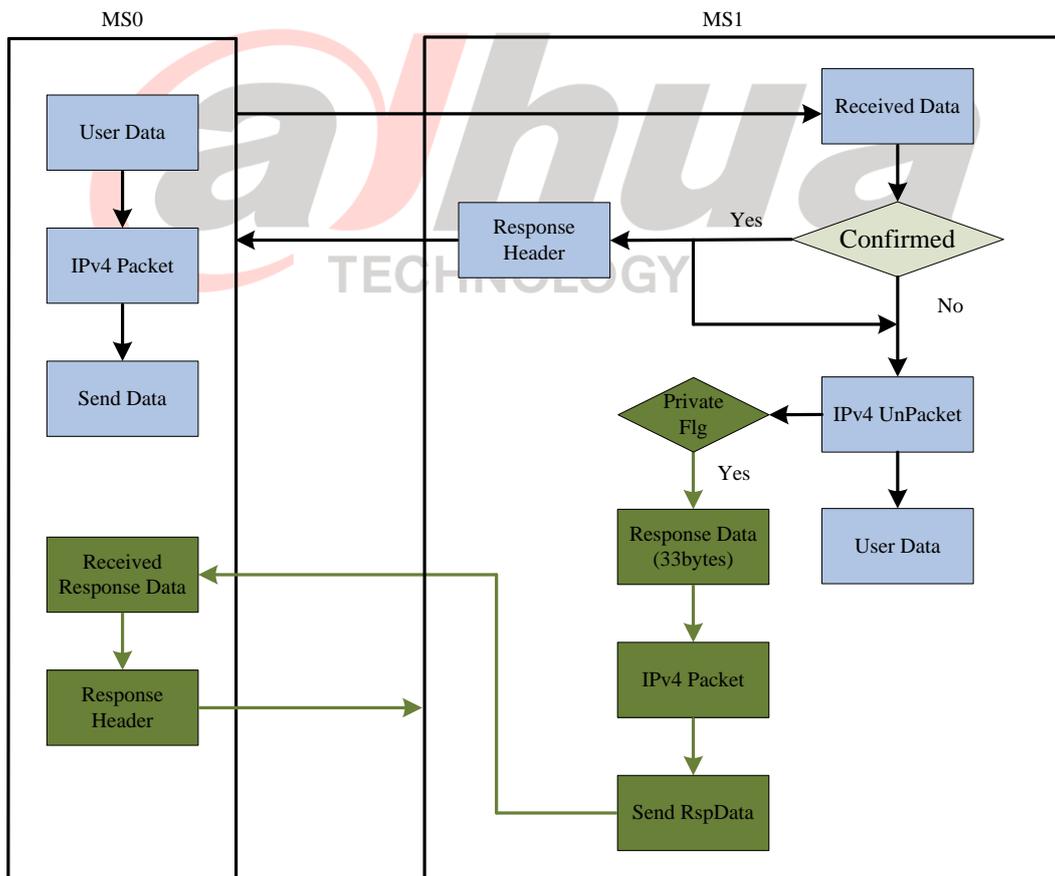


图 103 短信业务应用流程图

常规系统的短信业务使用IP数据业务的格式，对应的使用协议定义的IP非确认帧头和IP确认帧头格式，实际发送的短信内容需要进行IPV4的封装后再发送，接收端在显示短信时需要先解封IPV4的封装数据后将实际接收到的短信内容显示保存。

如上图定义了IP短信业务的工作流程。其大致过程描述如下：

- 1、 用户编辑完要发送的短信内容，软件开始将用户数据进行IPV4格式封装准备发送的内容，对应采用不同的值定义IP非确认和确认封装数据；接收MS在完全正确接收短信数据的前提下，若当前接收短信类型为确认短信则需要先发送一个Response Header至发送端表示接收ACK；
- 2、 随即无论是IP确认还是IP非确认，接收MS开始解析IPv4结构数据内容，在满足IPv4帧格式相关校验结果后保存实际有效接收的短信息内容；
- 3、 若当前为个地址短信接收，则需要以个地址短信的方式发送一个33bytes的Response Data，其也满足IPv4结构；
- 4、 对于主动发送IP个地址短信内容的MS来说，其完全结束本次短信需要依赖接收到来自接收端MS的33Bytes的Response Data。

## 15.2 短信业务时序

短信常规业务下按是否使用数据二头区分可以分为单帧头和双帧头数据时序，对应如下图所示，通过第二个数据帧头携带自定义的控制信息实际如加密应用。

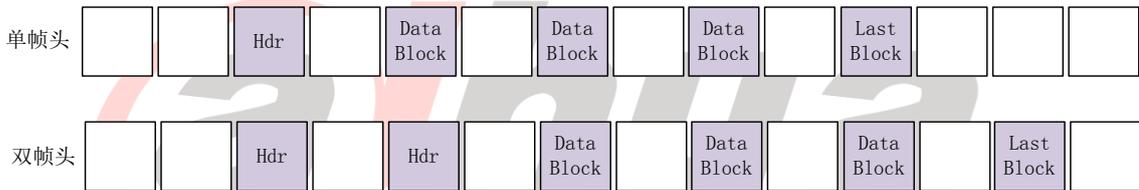


图 104 短信业务时序

默认定义组地址发送短信类型为非确认的短信格式，采用数据包体的帧格式为 Rate1/2 速率数据，无需确认应答。

如下图定义了发送一条空白短信，非确认短信发送实际所需要的发送数据包数。

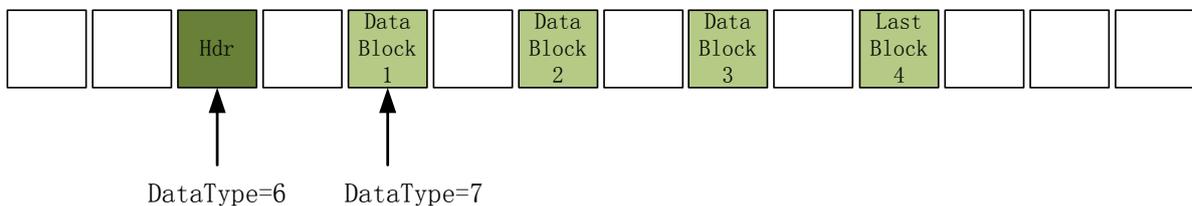


图 105 非确认短信业务结构图

定义了个地址短信类型有非确认的短信格式和确认的短信格式两种模式，其中确认短信采用数据包体的帧格式为 Rate3/4 速率，其中关于非确认短信的时序定义与组地址的过程是一致的，但是在整个短信收发处理的流程过程中，个地址与组地址模式的差异在于，对于个地址发送的短信，无论是确认模式还是非确认模式，在正确处理完当前发送的短信内容后，接收方在收到短信后都将发送一条反馈的短信至发送方，该短信的发送模式由 CPS 预先设定的是确认还是非确认

决定。

如下图定义了确认短信发送实现的时序关系图，图中接收方需要对当前的接收正确与否做出反馈的数据帧头，其中灰色部分的数据包体只有当前接收存在错误帧的模式下启动，用来表示对应的错误帧序号。

在正确完成短信的接收后，接收方将开始发起一条定义为反馈信息的短信。

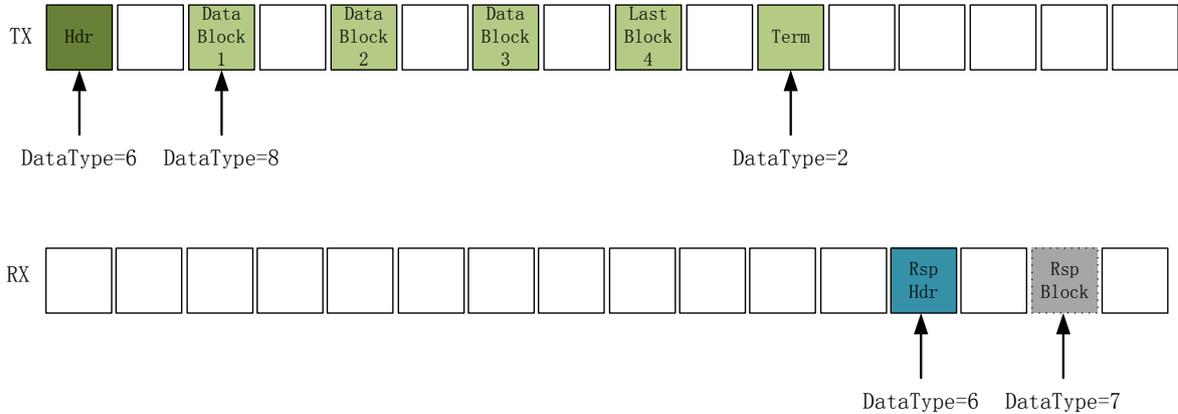


图 106 确认短信业务结构图

### 15.3 短信数据帧结构

短信业务由数据帧头、数据帧体、确认数据帧尾、响应数据帧头以及响应数据反馈包体几种类型组成。

#### 15.3.1 数据帧头结构

如下图定义了数据帧头 PDU 结构图。

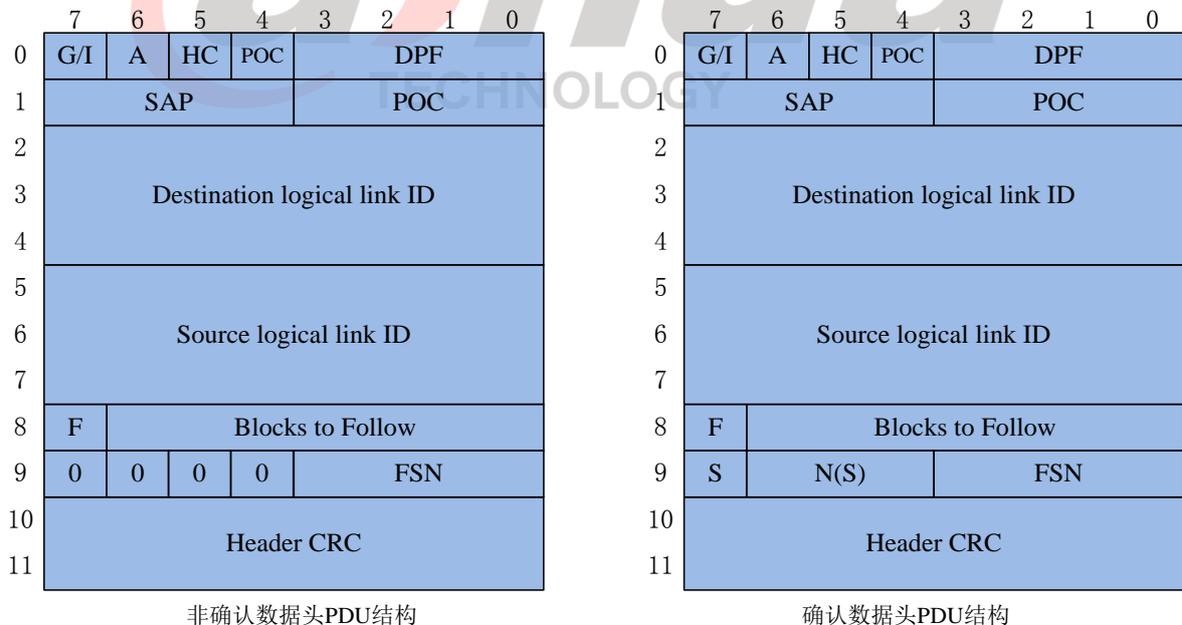


图 107 数据帧头 PDU 结构

表 61 非确认数据帧头 PDU 定义

Information element	Length	Value	Remark
G/I	1	1	目的地址是个地址还是组地址
A	1	0	
HC	1	0	完整数据头使用
POC	1		填充字节个数的最高位（以实际计算所得）
DPF	4	0010	IP UnConfirmed Data
SAP	4	0100	SAP 指向数据帧头后的数据类型，默认 0100，若存在数据二头则此处为 1001
POC	4		填充字节个数的低 4 位（以实际计算所得）
Destination Address	24		目的地址
Source Address	24		源地址
F	1		1 表示首次发送，0 表示重发
Blocks to Follow	7		发送数据包总数
Reserved	4	0000	
FSN	4	0000	
Header CRC	16		帧头 CRC16 的校验结果

表 62 确认数据帧头 PDU 定义

Information element	Length	Value	Remark
G/I	1	0	目的地址是个地址还是组地址
A	1	1	需要应答确认，1 表示需要，0 表示不需要
HC	1	0	完整数据头使用
POC	1		填充字节个数的最高位（以实际计算所得）
DPF	4	0011	IP Confirmed Data
SAP	4	0100	SAP 指向数据帧头后的数据类型，默认 0100，若存在数据二头则此处为 1001
POC	4		填充字节个数的低 4 位（以实际计算所得）
Destination Address	24		目的地址
Source Address	24		源地址
F	1		1 表示首次发送，0 表示重发
Blocks to Follow	7		发送数据包总数（以实际计算所得）
S	1	1	Default
N(S)	3		发送序列 0~7
FSN	4	1000	单帧确认帧序列值 1000
Header CRC	16		帧头 CRC16 的校验结果

### 15.3.2 数据帧结构

如下表定义了各种速率类型所占用的数据帧体携带数据内容字节数，实际在 IP 格式短信应用中使用了 Unconfirmed Rate 1/2 和 Confirmed Rate 3/4 两种类型。

表 63 数据帧体字节数表

确认/非确认类型	编码速率	每包字节数
Confirmed	Rate 1/2	10
Confirmed	Rate 3/4	16
Confirmed	Rate 1	22
Unconfirmed	Rate 1/2	12
Unconfirmed	Rate 3/4	18
Unconfirmed	Rate 1	24

速率1/2数据块(R\_1\_2\_DATA)PDU长度是96比特，用于非确认业务时包含96比特用户数据，其内容如下表所示。

表 64 1/2 速率非确认数据块(R\_1\_2\_DATA) 的 PDU 内容

信息	长度(Bit)	注解
用户数据	96	用户数据域

表 65 1/2 速率非确认最后数据块(R\_1\_2\_LDATA) 的 PDU 内容

信息	长度(Bit)	注解
用户数据	64	用户数据域
整个数据校验	32	CRC32校验值

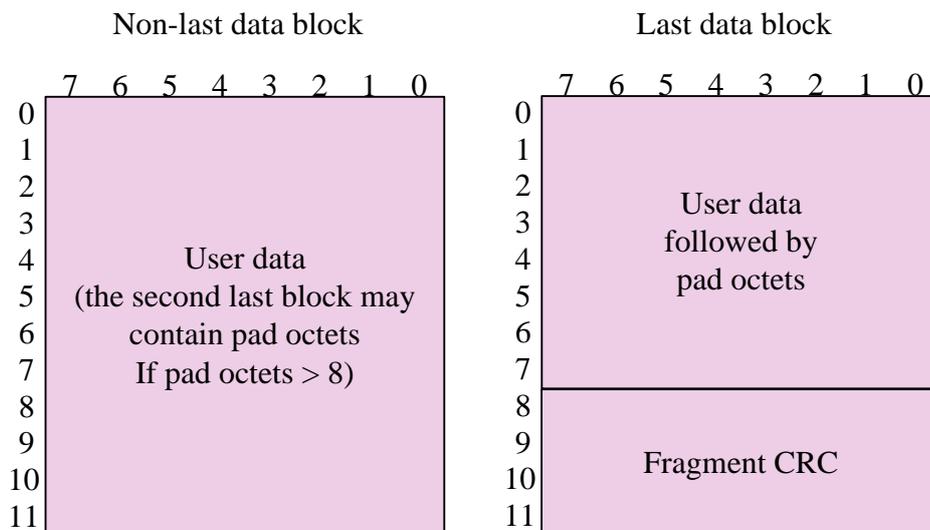


图 108 非确认 Rate1/2 数据块结构图

速率 3/4 数据块(R\_3\_4\_DATA)PDU 长度是 144 比特，用于带确认业务时包含 128 比特用户

数据和 7 比特数据块序列号，还有 9 比特的 CRC 校验数据，其内容如下表所示。

表 66 3/4 速率带确认数据块(R\_3\_4\_DATA) 的 PDU 内容

信息	长度(Bit)	注解
数据块的序列号 (DBSN)	7	
数据块校验	9	CRC9用于数据块序列号以及用户数据
用户数据	128	用户数据域

表 67 3/4 速率带确认最后数据块(R\_3\_4\_LDData) 的 PDU 内容

信息	长度(Bit)	注解
数据块的序列号 (DBSN)	7	
数据块校验	9	CRC9用于数据块序列号以及用户数据
用户数据	96	用户数据域
整个数据校验	32	CRC32校验值

### 15.3.3 响应数据挂起时间

接收端需发送一个响应以确认接收到数据，要保证响应的快速传输，系统需要为响应预留信道。此步骤被称为“数据响应挂起时间”。

在直通模式下，发送的 MS 需要通过发射“数据结束 LC 信令”来表示数据响应挂起时间的开始，接收方应按礼貌方式发送响应。要区别数据结束 LC 信令与其它突发，则需要设置 SLOT Type 中的数据类型的信息为“LC 结束标志”。

在中转模式下，基站需要通过发送“数据结束 LC 信令”以表示数据响应挂起时间开始。为避免冲突，中转台应在数据响应挂起期间设置 CACH 的 AT 位为繁忙。移动台应在其“数据响应挂起时间”内以非礼貌方式回一个响应。

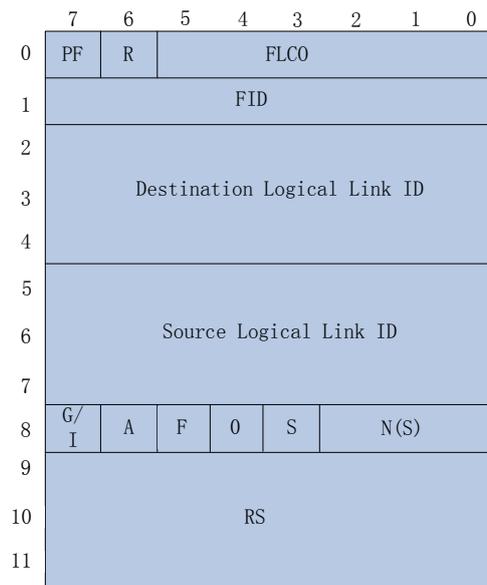


图 109 数据帧尾 PDU 结构图

表 68 数据帧尾 PDU 定义

Information element	Length	Value	Remark
PF	1	0	保留
R	1	0	保留
FLCO	6	110000	数据帧尾
FID	8	00000000	
Destination Address	24		目的地址
Source Address	24		源地址
G/I	1		目的地址是个地址
A	1	0	
F	1	0	1 表示首次发送, 0 表示重发
0	1		
S	1	1	
N(S)			发送序列 0~7

### 15.3.4 应答数据帧

确认短信过程需要发送应答反馈帧, 如下图描述的是确认短信应答反馈帧信息, 其主要由反馈数据帧头以及其后紧跟数据帧两种组成, 其中反馈帧头结构信息与数据帧头结构信息一致, 只在控制字段上存在不同的定义值, 如下表描述的是反馈数据帧头的控制信息。

反馈数据包体只存在于实际接收过程中出现错误包的信息时, 其使用 Rate1/2 速率数据帧结构, 其最大可以携带 64 个 blocks 的错误包序列号信息, 对应位置值为 0 表示该序列号的数据包接收错误。由于确认短信数据帧的包序列号由 7 个 bits 构成, 因而实际应用当中可以在反馈数据帧头后最多发 2 个数据包体。

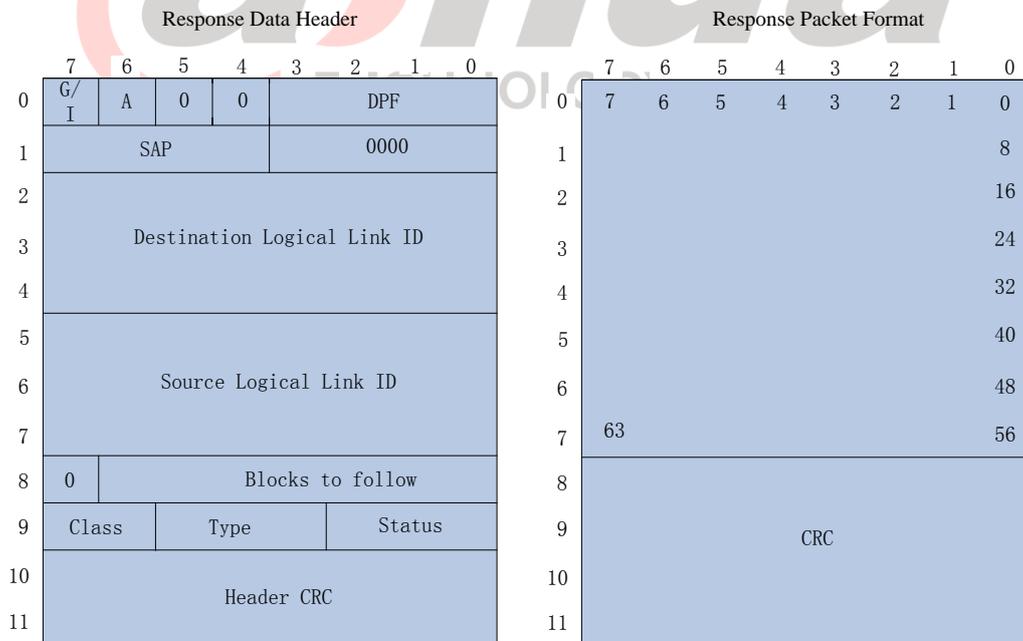


图 110 应答数据帧 PDU 结构图

表 69 应答数据帧头 PDU 定义

Information element	Length	Value	Remark
G/I	1	0	目的地址是个地址
A	1	0	
Reserved	1	0	保留
Reserved	1	0	保留
DPF	4	0001	Response Packet
SAP	4	0100	
Reserved	4	0000	保留
Destination Address	24		目的地址
Source Address	24		源地址
Reserved	1	0	保留
Blocks to Follow	7		发送反馈数据包体总数
Class	2	0	00: ACK; 01: NACK; 10: SACK
Type	3		范围: 000~110, 常用 000 为 SACK, 001 为 ACK
Status	3		发送序列 0~7

反馈数据包体主要应用确认短信的重传机制，只需要重传对应的错误数据包即可完成重传。如下图举例了反馈数据包体反映的错误数据包重收后对应的错误图样更新过程。

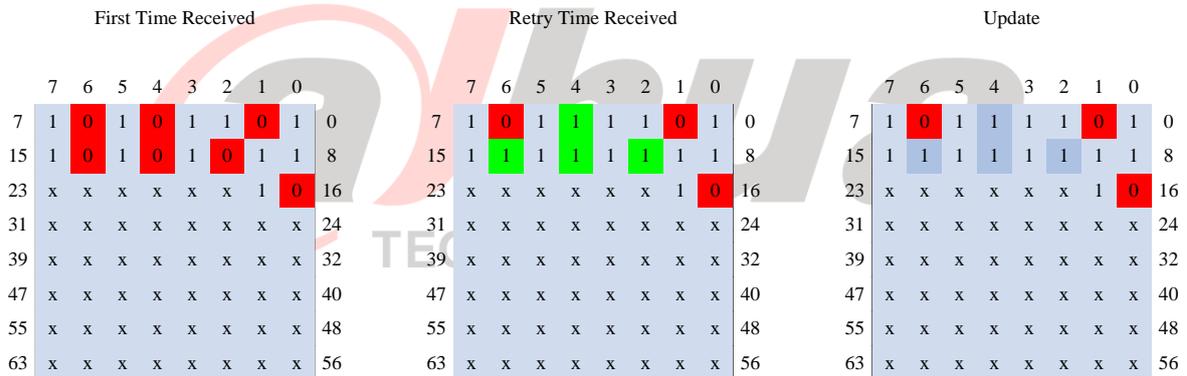


图 111 反馈数据包体错误图样更新过程

## 15.4 IPv4 结构说明

如下图定义 IPv4 的结构图。

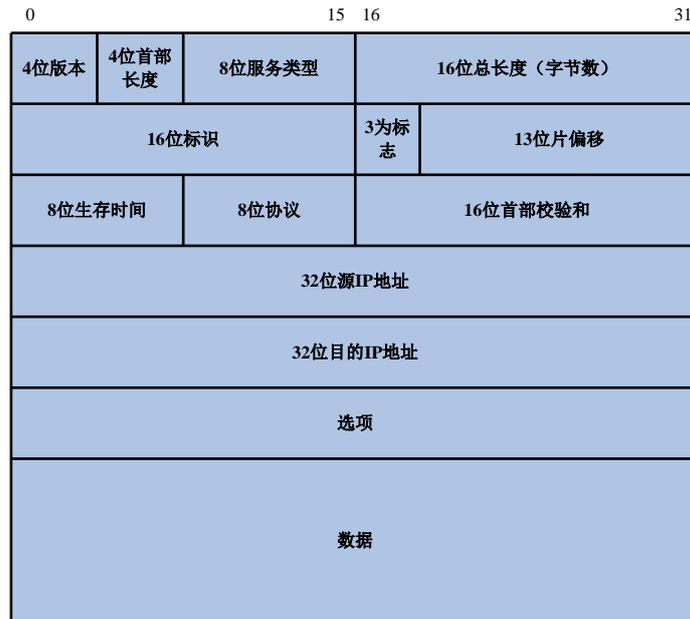


图 112 IPv4 结构图

如下图定义了UDP校验和计算各字段定义的格式。



图 113 UDP 校验和计算各字段定义

### 15.4.1 IPv4 Packet Data

参考 IPv4 的结构图,关于 IP 数据封装过程中,在原有的发送数据内容前加入 34bytes 的 IPv4 的帧结构内容,其详细信息如下表说明。

表 70 IPv4 Packet Data

字段	长度	值	描述
版本	4	0100	
首部长度	4	0101	
服务类型	8	00000000	
总长度	16		其值为 34+实际数据内容长度。
标识	16		对讲机开机后每发送一次短信对应

			的 Index+1。
标志	3	000	
偏移	13	0000000000000	
生存时间	8	01000000	
协议	8	00010001	
首部校验和	16		IPv4Header 前 20bytes 经 CheckSum 计算得到。
源地址	32		源地址由 12+对讲机本机个地址。
目的地址	32		若目标地址为组地址则由 225+目标组地址构成；若目标地址为个地址则由 12+目标个地址构成。
源端口号	16	0x0FA7	16 进制
目的端口号	16	0x0FA7	16 进制
UDP 长度	16		其值为 14+实际数据内容长度。
UDP 校验和	16		
选项+数据长度	16		其值为 4+实际数据内容长度。
选项 1	8		0xe0 代表确认；0xa0 代表非确认。
选项 2	16		0x0081~0x009f 循环
选项 3	8	0x04	
。 。 。			实际数据内容

#### 15.4.2 IPv4 Response Packet Data

参考IPv4的结构图，在关于IP Response数据封装过程中，其为固定的33bytes的IPv4的帧结构内容，其详细信息如下表说明。

表 71 IPv4 Response Packet Data

字段	长度	值	描述
版本	4	0100	
首部长度	4	0101	
服务类型	8	00000000	
总长度	16	33	其值为 33。
标识	16		其值由接收到的确认短信中上表的（选项 2 - 0x0081）。
标志	3	000	
偏移	13	0000000000000	
生存时间	8	01000000	
协议	8	00010001	
首部校验和	16		IPv4Header 前 20bytes 经 CheckSum 计算得到。
源地址	32		源地址由 12+对讲机本机个地址。
目的地址	32		目的地址由 12+目标个地址构成。
源端口号	16	0x0FA7	16 进制
目的端口号	16	0x0FA7	16 进制

UDP 长度	16	14	
UDP 校验和	16		
选项+数据长度	16	3	
选项 1	8	0xBF	
选项 2	16		其值由接收到的确认短信中上表的 (选项 2 - 0x0080)。

## 15.5 非确认短信

### 15.5.1 非确认短信发送

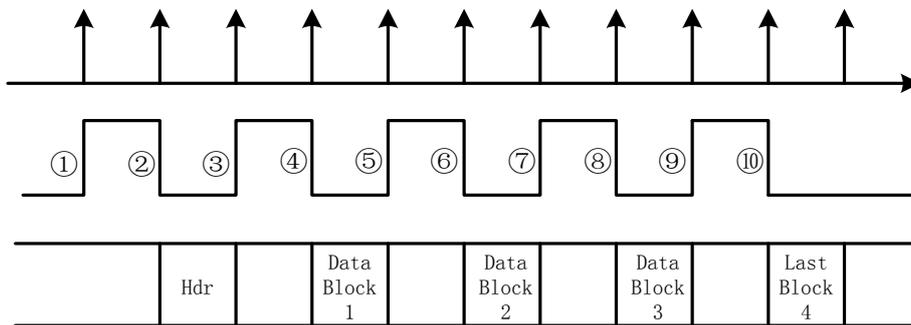


图 114 非确认短信发送业务时序图

参考非确认短信发送时序图，其发送步骤如下（已知发送时隙建立）：

步骤①：

- a) 根据发送数据速率（参考各速率非确认数据包体结构图）、数据长度，计算所需要发送的数据包体个数，填充字节数（填 0），计算数据内容的 CRC32，划分每一个数据包体需要携带的数据内容；
- b) 根据数据计算所得的数据包个数、填充字节数以及发送的短信方式，组帧 80bits 的数据帧头信息，将 80Bits 信息写入发送 Ram 地址 0x16000000~0x16000009；
- c) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x60 表示发送数据帧头；
- d) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送；

步骤③：

- a) 将第 1 包 96/144/192Bits 三种速率（选择其一）信息写入发送 Ram 地址 0x16000000~0x16000017；
- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x70/0x80/0xa0 表示发送数据包体；
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送；

步骤⑤：

- a) 将第 2 包 96/144/192Bits 三种速率（选择其一）信息写入发送 Ram 地址 0x16000000~0x16000017；
- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x70/0x80/0xa0 表示发送数据包体；
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送；

步骤⑦：

- a) 将第 3 包 96/144/192Bits 三种速率（选择其一）信息写入发送 Ram 地址 0x16000000~0x16000017；
- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x70/0x80/0xa0 表示发送数据包体；

- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送；
- 步骤⑨：
- a) 将第 4 包 96/144/192Bits 三种速率（选择其一）信息含 CRC32 的校验结果写入发送 Ram 地址 0x16000000~0x16000017；
  - b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x70/0x80/0xa0 表示发送数据包体；
  - c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送；

其余步骤②④⑥⑧⑩，均为配置寄存器LAYER2\_TXRX\_CTRL为0x0或者0x40表示下一个时隙发送关闭，接收开启或者关闭。

### 15.5.2 非确认短信接收

非确认短信接收业务的启动，通过检测到数据帧头的系统中断，且都满足地址匹配的情况下开启时隙接收短信业务，对应的设计时序如下图所示。

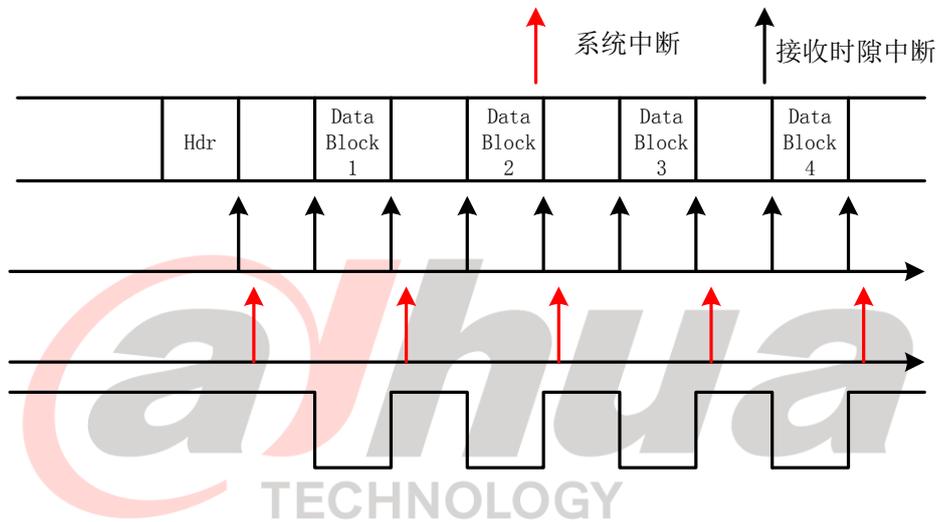


图 115 非确认短信接收业务时序图

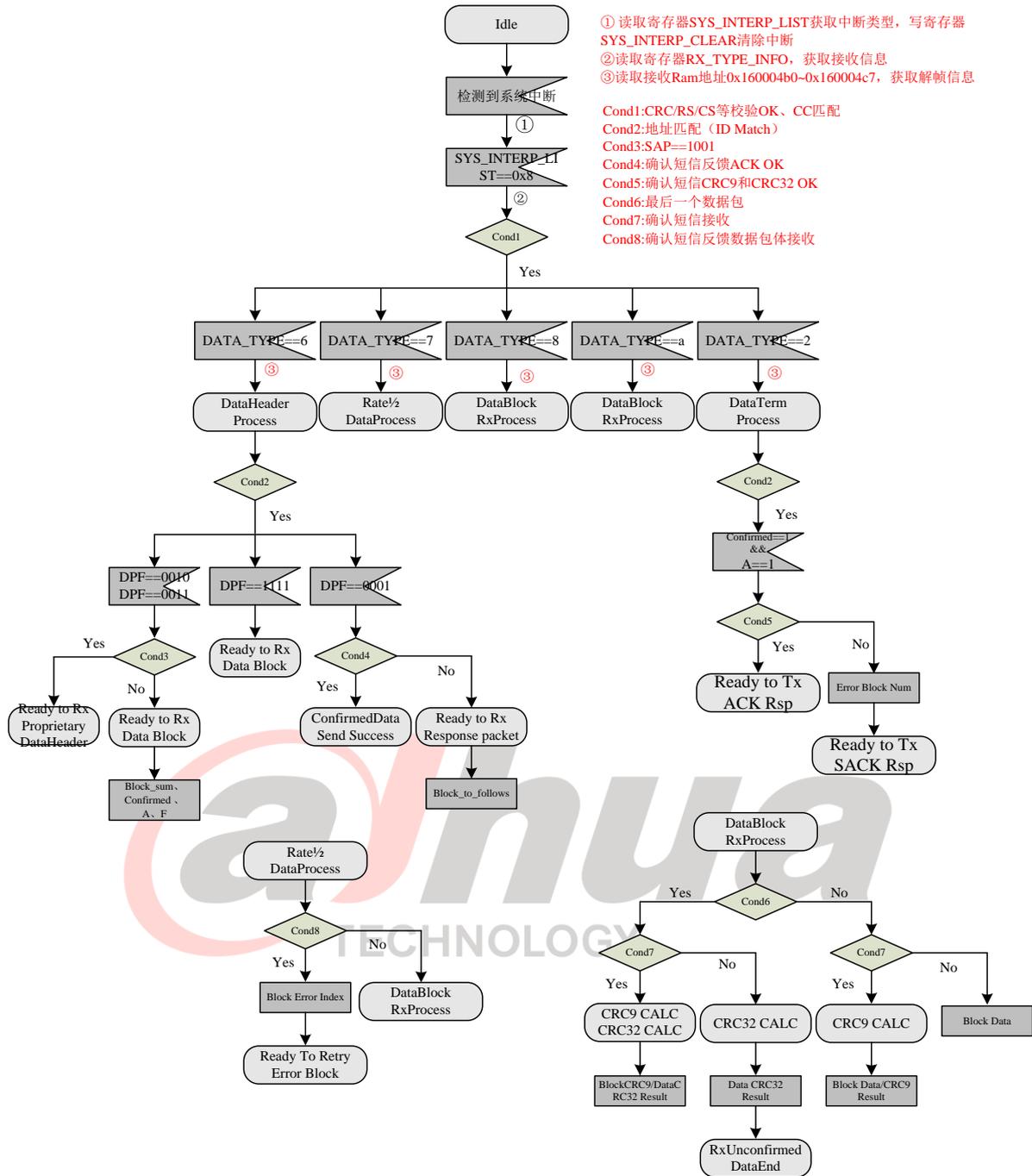


图 116 短信接收流程图

非确认短信业务接收步骤如下：

- 1、接收机盲收检测到解帧系统中断，读取寄存器RX\_TYPE\_INFO，判断是否接收正确的数据帧头信息，然后读取接收Ram地址0x160004b0~0x160004bc，判断接收到的地址是否匹配；
- 2、若地址匹配，则开始启动时隙接收非确认短信业务，开始时隙开关寄存器LAYER2\_TXRX\_CTRL寄存器；

- 3、根据解帧系统中断，读取寄存器RX\_TYPE\_INFO，依次对应从接收Ram读取相应的数据包体信息，直至最后一包接收完成；
- 4、将所有接收到的数据包体依次进行CRC32校验计算，判断接收信息是否完全正确。

## 15.6 确认短信

### 15.6.1 确认短信发送

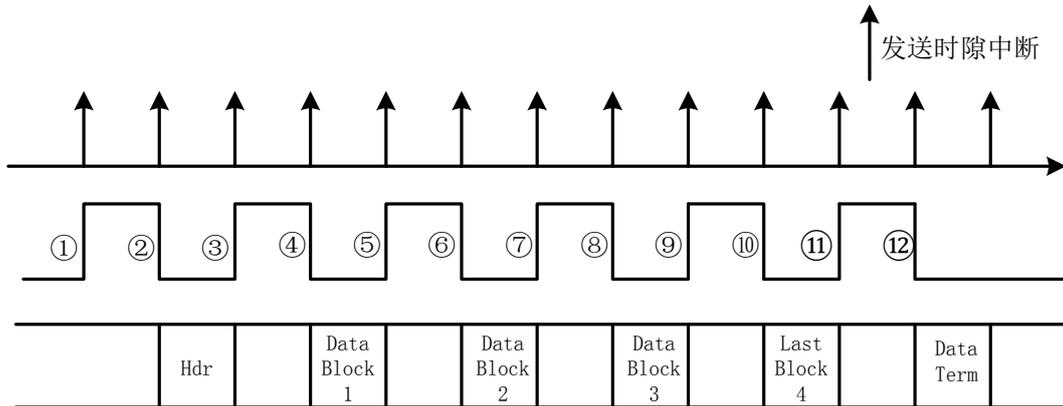


图 117 确认短信发送业务时序图

参考确认短信发送时序图，其发送步骤如下（已知发送时隙建立）：

步骤①：

- a) 根据发送数据速率（参考各速率确认数据包体结构图）、数据长度，计算所需要发送的数据包体个数，填充字节数（填 0），计算数据内容的 CRC32，划分每一个数据包体需要携带的数据内容；
- b) 根据数据计算所得的数据包个数、填充字节数以及发送的短信方式，组帧 80bits 的数据帧头信息（参考章节 6.1），将 80Bits 信息写入发送 Ram 地址 0x16000000~0x16000009；
- c) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x60 表示发送数据帧头；
- d) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送；

步骤③：

- a) 将第 1 包 96/144/192Bits 三种速率（选择其一）信息包含本包数据内容的 CRC9 校验结果写入发送 Ram 地址 0x16000000~0x16000017；
- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x70/0x80/0xa0 表示发送数据包体；
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送；

步骤⑤：

- a) 将第 2 包 96/144/192Bits 三种速率（选择其一）信息包含本包数据内容的 CRC9 校验结果写入发送 Ram 地址 0x16000000~0x16000017；
- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x70/0x80/0xa0 表示发送数据包体；
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送；

步骤⑦：

- a) 将第 3 包 96/144/192Bits 三种速率（选择其一）信息包含本包数据内容的 CRC9 校验结果写入发送 Ram 地址 0x16000000~0x16000017；

- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x70/0x80/0xa0 表示发送数据包体;
  - c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;
- 步骤⑨:
- a) 将第 4 包 96/144/192Bits 三种速率 (选择其一) 信息包含本包数据内容的 CRC9 校验结果和所有数据内容的 CRC32 校验结果写入发送 Ram 地址 0x16000000~0x16000017;
  - b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x70/0x80/0xa0 表示发送数据包体;
  - c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;
- 步骤⑩:
- a) 准备语音 LC72bits 的确认短信帧尾信息, 将 72Bits 信息写入发送 Ram 地址 0x16000000~0x16000008;
  - b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x20 表示发送帧尾 LC;
  - c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;
- 其余步骤②④⑥⑧⑩⑫, 均为配置寄存器 LAYER2\_TXRX\_CTRL 为 0x0 或者 0x40 表示下一个时隙发送关闭, 接收开启或者关闭, 发送完成帧尾后要求开启接收用于接收确认短信反馈。

### 15.6.2 确认短信接收

确认短信接收业务的启动, 通过检测到数据帧头的系统中断, 且都满足地址匹配的情况下开启时隙接收短信业务, 在接收到数据结束帧尾后需要向对方反馈 ACK, 下图反映的正确接收到短信的时序图。

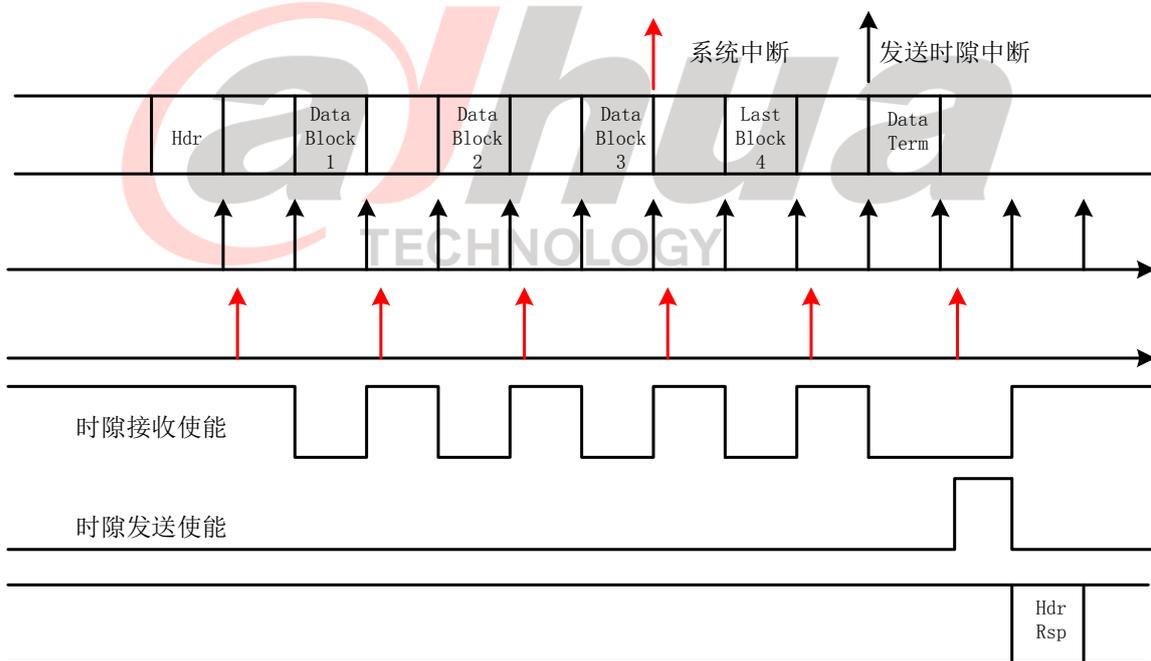


图 118 确认短信接收业务时序图

确认短信业务接收步骤如下 (参考短信接收流程图):

- 1、接收机盲收检测到解帧系统中断, 读取寄存器 RX\_TYPE\_INFO, 判断是否接收正确的数据帧头信息, 然后读取接收 Ram 地址 0x160004b0~0x160004bb, 判断接收到的地址是

- 否匹配；
- 2、若地址匹配，则开始启动时隙接收确认短信业务，开始时隙开关寄存器 LAYER2\_TXRX\_CTRL 寄存器；
  - 3、根据解帧系统中断，读取寄存器 RX\_TYPE\_INFO，依次对应从接收 Ram 读取相应的数据包体信息，并相应计算每一包数据的 CRC9 校验计算，直至最后一包接收完成；
  - 4、将所有接收到的数据包体依次进行 CRC32 校验计算，判断接收信息是否完全正确。
  - 5、所有数据包的 CRC9 正确且数据内容的 CRC32 正确，则接收端需要反馈 ACK 信息至发送端。

## 16 附录 C：数字信令应用

### 16.1 功能描述

如下图定义了常规系统下 CSBK 信令业务时序图，包括前导 CSBK 控制包、功能 CSBK 控制包和反馈 CSBK 控制包。

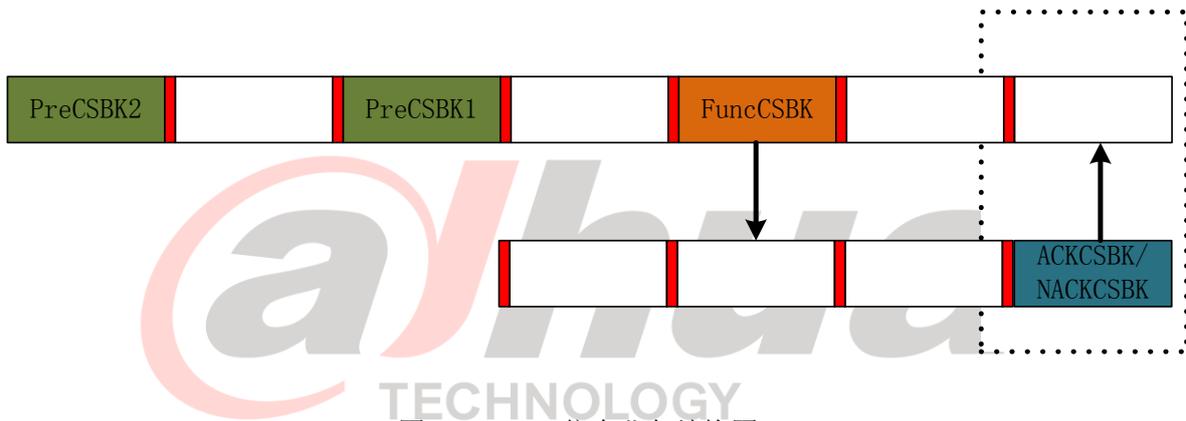


图 119 CSBK 信令业务结构图

### 16.2 信令发送

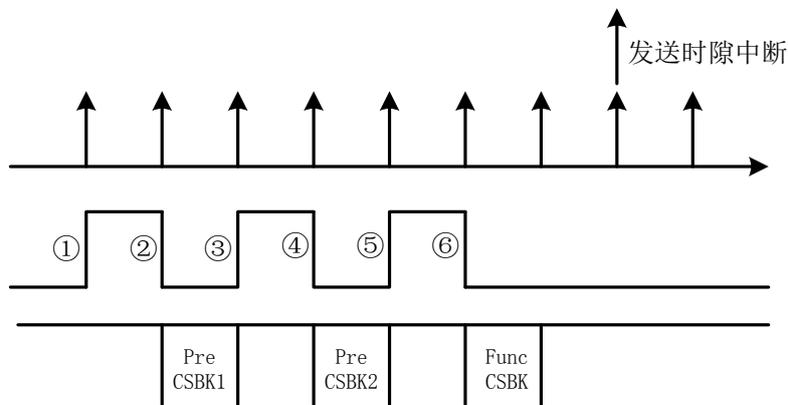


图 120 CSBK 信令业务发送时序图

参考 CSBK 信令业务发送时序图，其发送步骤如下（已知发送时隙建立）：

步骤①:

- a) 准备第一个前导 CSBK80bits 的控制信息写入发送 Ram 地址 0x16000000~0x16000009;
- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x30 表示发送数据帧头;
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;

步骤③:

- a) 准备第二个前导 CSBK80bits 的控制信息写入发送 Ram 地址 0x16000000~0x16000009;
- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x30 表示发送数据帧头;
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;

步骤⑤:

- a) 准备功能 CSBK80bits 的控制信息写入发送 Ram 地址 0x16000000~0x16000009;
- b) 配置寄存器 LAYER2\_SEND\_TYPE 为 0x30 表示发送数据帧头;
- c) 配置寄存器 LAYER2\_TXRX\_CTRL 为 0x80 表示下一个时隙开启发送;

其余步骤②④⑥, 均为配置寄存器 LAYER2\_TXRX\_CTRL 为 0x0 或者 0x40 表示下一个时隙发送关闭, 接收开启或者关闭, 发送完成功能 CSBK 包后要求开启接收用于接收 CSBK 应答反馈。

### 16.3 信令接收

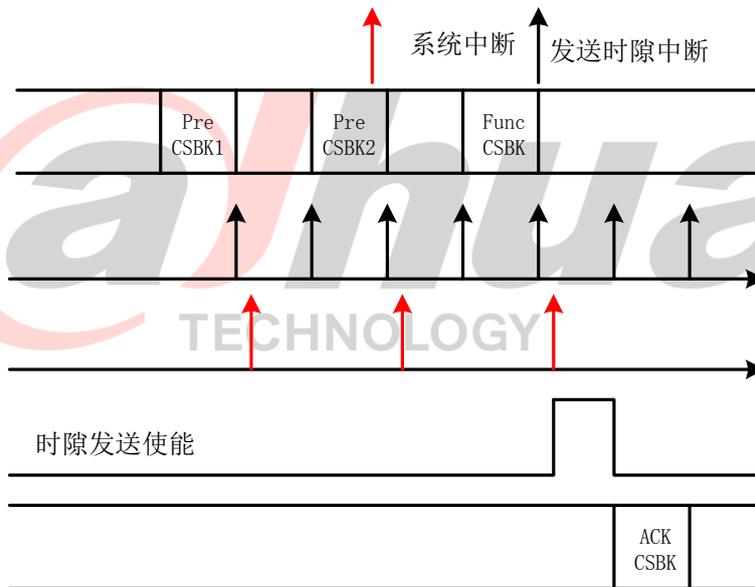


图 121 CSBK 信令业务接收时序图

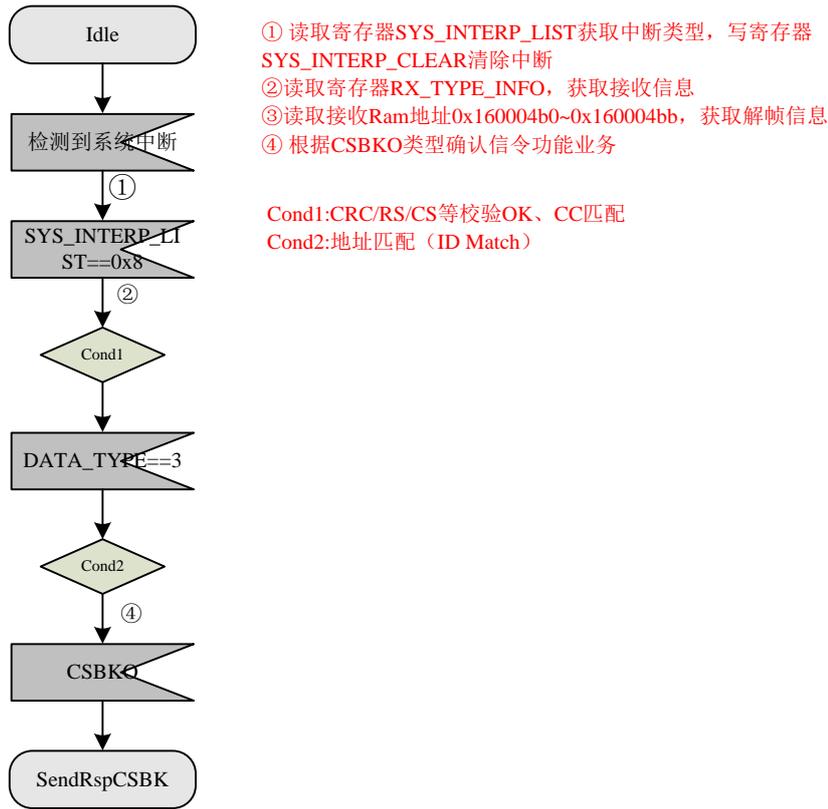


图 122 CSBK 信令业务接收流程图

CSBK信令业务接收步骤如下：

- 1、接收机盲收检测到解帧系统中断，读取寄存器RX\_TYPE\_INFO，判断是否接收正确的CSBK控制信息，然后读取接收Ram地址0x160004b0~0x160004bb，判断接收到的地址是否匹配；
- 2、若地址匹配，且满足功能定义的CSBK信令功能，则接收端需要反馈CSBK信令至发送端。

## 17 附录 D：数字加密应用

### 17.1 概述

HR\_C7000 数字对讲系统加密功能，主要是提供一种方式保持通信业务是秘密的，通信业务包括语音和数据，其中 HR\_C7000 语音业务支持静态基本加密和动态增强加密两种方式。使用加密功能的数字信道发送的语音和数据只有启用同一设置功能的对讲机用户可以接收短信和接听语音业务，其它任何密钥不匹配的对讲机用户都将无法接入通信业务。

数字加密功能只针对语音和数据信息加密保护。二层的语音和数据帧头、数据反馈包、链接控制数据都是不被加密保护的。这就意味着源和目的的地址、组地址都是不被加密保护的。控制信令例如遥毙、远程监听、对讲机检测、呼叫提示和嵌入式 RC 信令也都是不被加密保护的。

数字加密功能应用于所有的数字工作模式并且贯穿于发送对讲机与目的对讲机间的通信。

数字加密功能应用于保护个呼语音呼叫、组呼语音呼叫、全呼呼叫、紧急呼叫和所有的分组数据业务包括个、组、非确认、确认。

## 17.2 语音加密

HR\_C7000 支持语音加密功能。根据不同的加密需求，用户可以选择配置成的静态基本加密方式和动态增强加密方式。

### 17.2.1 静态基本加密

静态基本加密方式采用配置加密寄存器值，HR\_C7000 根据寄存器设置值产生相应的加解密密钥用以发送和接收过程，其过程示意图参考下图，其配置要求参考下表。

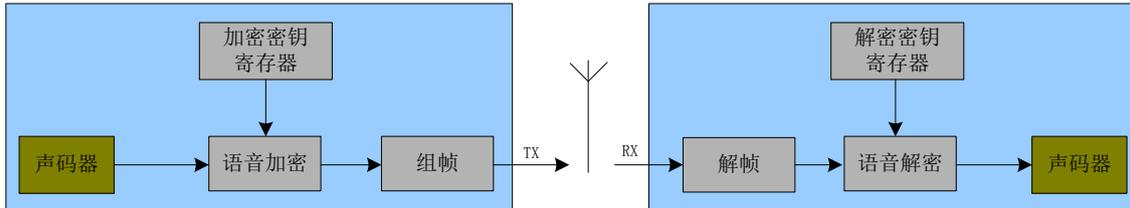


图 123 静态加密过程示意图

表 72 静态基本加密配置表

序号	寄存器/Ram 地址	配置值	描述
1	LOCAL_CC	0x1	加密模式选择
2	SCRAMBLE_REG		配置发送和接收密钥寄存器值

### 17.2.2 动态增强加密

如下图定义了语音加密发送，语音解密接收过程示意图。密钥的组成包括本地储存的密钥值（由 KeyID 获取）和初始向量值，对应由算法 ID 产生相应的密钥。加密参数只传递算法序号、使用本地密钥的 ID 值、初始向量，本地的密钥值不进行空口传递。

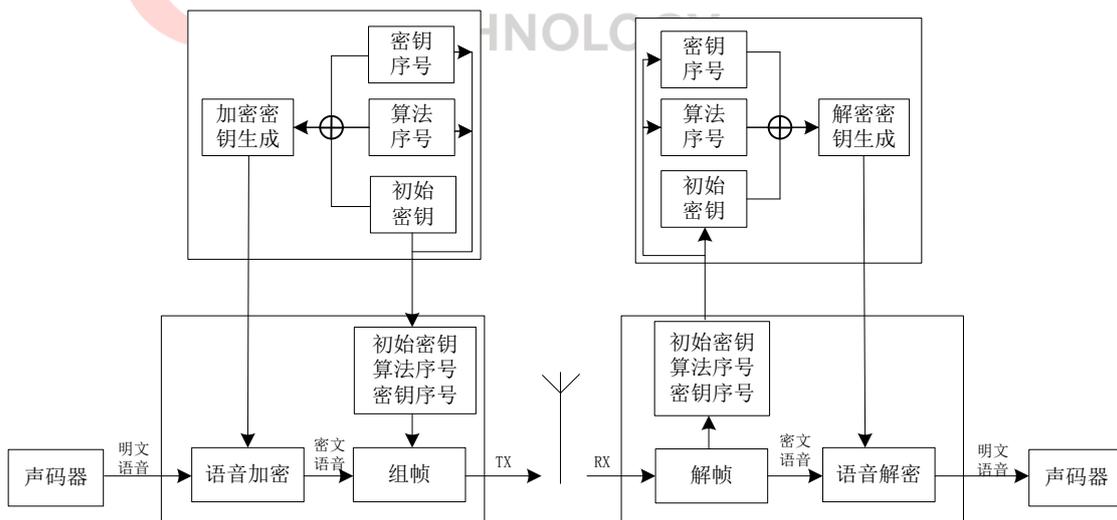


图 124 动态加密过程示意图

表 73 动态增强加密配置表

序号	寄存器/Ram 地址	配置值	描述
1	LOCAL_CC	0x10001	加密模式选择
2	VOICE_EMB_CTRL		PI 标志位控制
3	0x16000029		发送传递 KeyID 值, F 帧 EMB 区域传递使用
4	0x1600002a		高 3 比特位代表发送传递 ALOG ID 值, F 帧 EMB 区域传递使用
5	0x16000000~0x16000009		PI 帧传递加密参数
6	0x16000000~0x16000008		EMBLC 传递加密参数
7	0x16000495~0x160004af		发送 216bits 密钥, 供发送语音加密使用
8	0x16000945~0x1600095f		发送 216bits 密钥, 供接收语音解密使用

### 17.2.2.1 加密参数定义

加密参数主要有初始向量、加密标识符 (KeyID) 和加密算法三部分组成, 其中 PI 帧内容将包含上述三部分, 语音超帧后接入 EMB 区域传递初始向量, 语音帧 F 的 EMB 区域将传递 KeyID 和加密算法两部分信息。

如下图定义了 PI 帧的结构内容。



图 125 PI 帧 PDU 结构图

#### 17.2.2.1.1 初始向量

初始向量定义 48bits 的信息, 其可以通过 PI 帧传递或者语音超帧传递。

PI 帧传递方式见上图 PI 帧 PDU 结构图。

如下图定义了语音超帧传递初始向量 IV 的一种方式 (用户可以自定义), 主要是通过 CRC8、CRC16 生成 72 比特的改变的初始向量值。

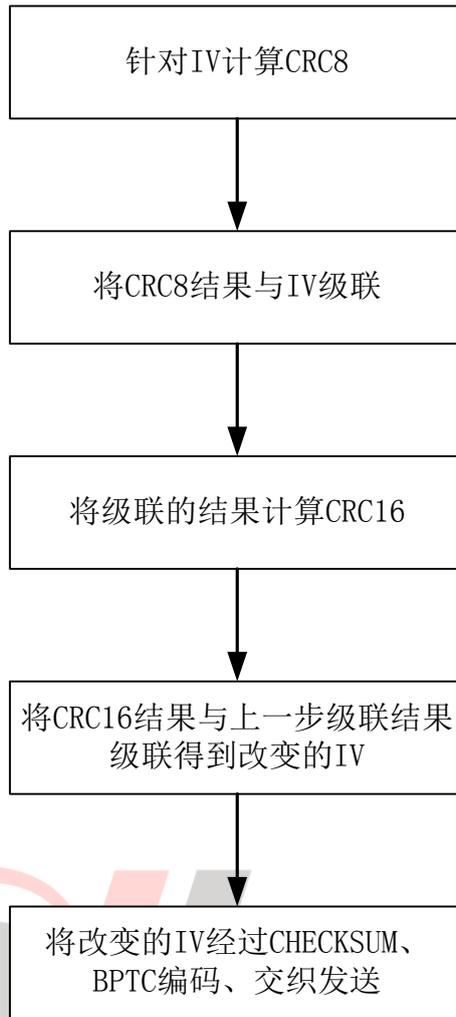


图 126 初始向量生成流程

如下图定义了初始向量实际在超帧内传输的结构。

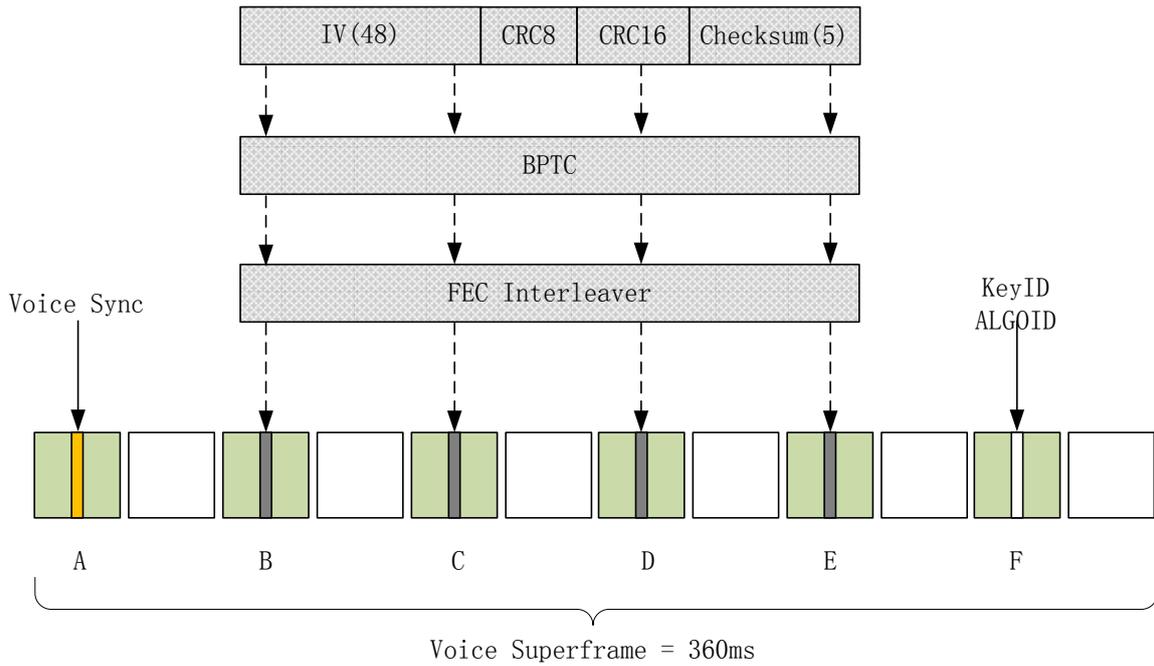


图 127 初始向量生成 EMB 传输方式

#### 17.2.2.1.2 KeyID 和 ALGOID

KeyID 是对应本地保存的密钥 ID 值索引，用以查找本地保存的密钥值。

算法 ID 是密钥计算的方式选择，通常可以有 ARC4、AES、DES 等加密算法实现方式。

上述两个加密参数可以有 PI 帧直接传递，或者是以语音 F 帧传递，语音 F 帧传递实现方式如下图所示。

已知 KeyID 为 8 比特，算法 ID 为 3 比特，11 比特的信息经过 FEC 产生 32 比特的嵌入式信息。该嵌入式信息在语音 F 帧的 EMB 区域传递。FEC 使用协议中规定的 BPTC (32,11) 的编码方式。

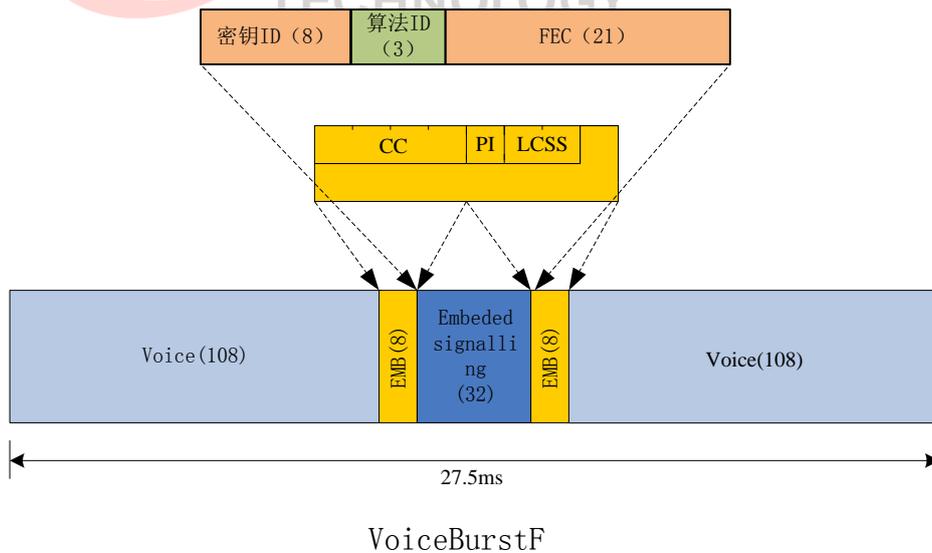


图 128 KeyID 和算法 ID 语音 F 帧传递方式

### 17.2.2.2 加密参数传递

如下图定义描述了加密参数传递更新的过程示意图，选择以密钥和语音后接入信息 1:1 的方式传递。

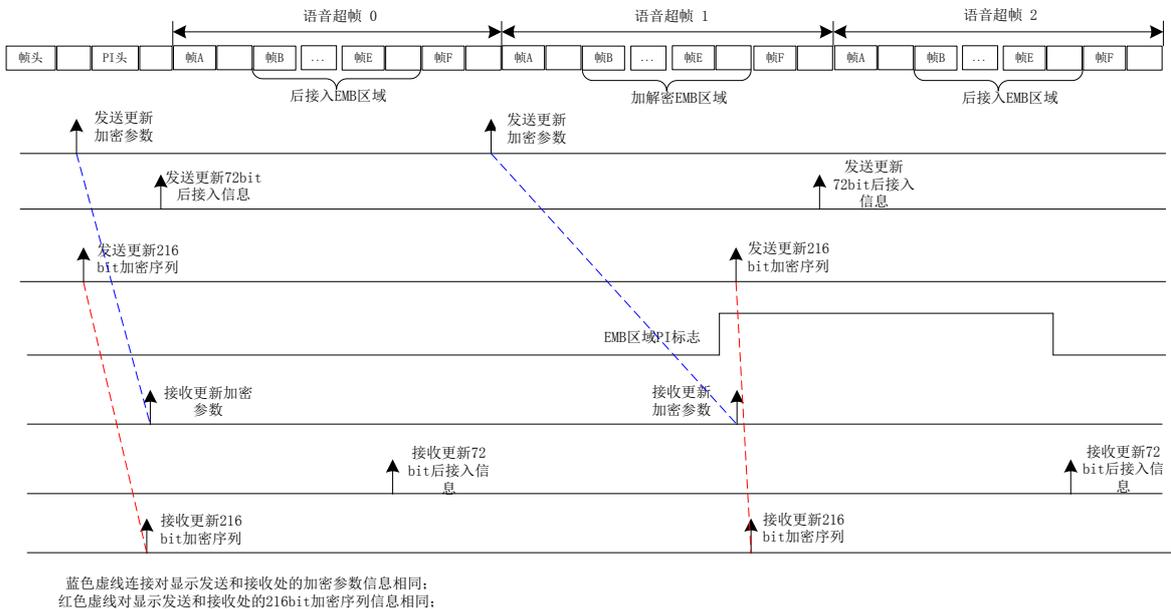


图 129 动态密钥传输更新示意图

#### 17.2.2.2.1 初始密钥传递和加密过程

更新密钥的过程中，定义第一个语音超帧号为 0，依次累加。

超帧号标记为 0, 2, 4, 6, …，这些帧的 EMB 区域需要加载的信息为语音后接入信息，该信息一共 72bit，要求在帧 A 语音数据发送开始后将这一信息写入发送端 Ram 0x16000000~0x16000008 地址空间。同时设置寄存器 VOICE\_EMB\_CTRL 的 PI 标志位 0。

超帧需要标记为 1, 3, 5, 7, …，这些帧的 EMB 区域需要加载的信息为初始密钥信息，该信息一共 72bit，要求在帧 A 语音数据发送开始后将这一信息写入发送端 Ram 0x16000000~0x16000008 地址空间。同时设置寄存器 VOICE\_EMB\_CTRL 的 PI 标志位 1。

加密语音通信中，在语音帧头之后，语音帧 A 发送之前，需要发送 1 个或多个语音 PI 帧头，该帧头中包含了初始加密密钥、加密算法序号和密钥 ID。完成 PI 帧发送后，需要根据 PI 帧发送的信息生成 216bit 的语音加密序列，并将生成的 216bit 密钥信息存入发送端的 RAM 地址 0x16000495~0x160004af 共 27byte 空间。

完成首次 216bit 的语音加密序列生成后，下一次新的加密序列生成在超帧序号为 1 的超帧帧 F 发送之前的时隙完成，这样依次在 3, 5, 7, …，语音超帧的帧 F 之前进行 216bit 加密序列更新。

在准备发送 F 帧之前的空闲时隙，将当前超帧序号信息写入发送端 RAM 的 0x16000029 地址，算法 ID 写入 0x1600002A 的高 3bit。

如下图定义了动态密钥参数的传递状态流程图。

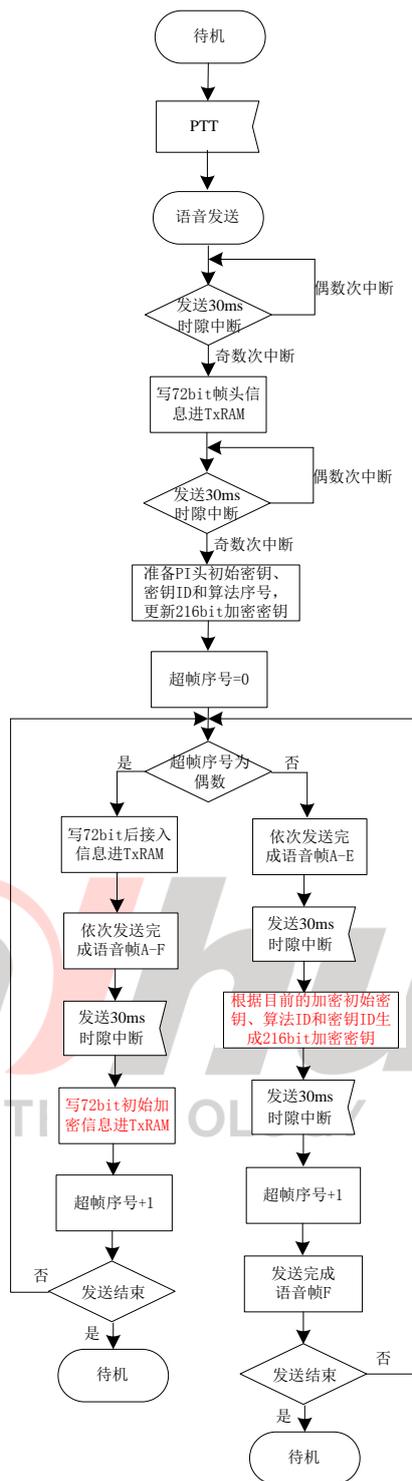


图 130 动态密钥参数加密传递流程图

#### 17.2.2.2.2 初始密钥接收和解密过程

语音解密流程步骤如下：

- 1、接收端接收到语音帧头，同时满足语音帧头控制字段中加密比特位为 1，则开启语音加密模式接收；
- 2、接收到 PI 帧信息，通过接收 Ram 获取 PI 帧的控制信息，得到密钥 ID、算法 ID 和初始密钥向量，计算语音密钥 216bits，并将其写入接收 Ram 的解密密钥存储地址；

- 3、步骤 1、2 描述的是第一次正常通过 PI 帧获取语音密钥的方法，如果 PI 帧丢失情况下，则将通过后接入中断，判断 PI 标志位为 1 的情况下获取加密初始向量参数，通过 F 帧 EMB 区域获取密钥 ID 和算法 ID 参数，计算语音密钥 216bits，并将其写入接收 Ram 的解密密钥存储地址；
- 4、在每一次后接入中断，根据 PI 标志位判断是语音后接入信息还是加密参数信息，在语音 F 帧的 EMB 区域获取密钥 ID 和算法 ID；
- 5、每一次超帧 PI 标志位为 1 时，在完成语音 F 帧接收中断后更新接收端的语音密钥。

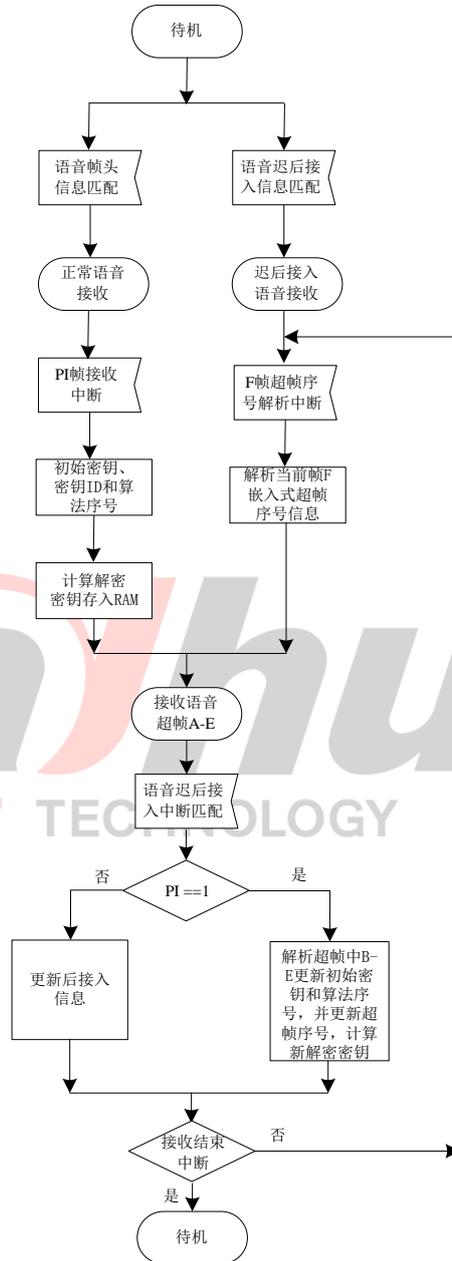


图 131 动态密钥参数解密传递流程图

### 17.2.2.3 加密控制帧

#### 17.2.2.3.1 语音帧头

如下表定义了语音帧头控制字段，与常规语音帧头差异在于 FID 以及对应 serviceoption 选项

中开启了 Privacy 字段为 1。

表 74 语音帧头定义

信息单元	长度	描述
PF	1	0
Reserved	1	0
FLCO	6	组呼 000000, 个呼 000011
<b>FID</b>	<b>8</b>	<b>00010000</b>
Service Options	Emergency	1 为紧急呼叫
	Privacy	1 为加密模式
	Reserved	2
	Broadcast	1 为全呼
	OVCN	1 0
	Priority level	2 00
Target addr	24	
Source addr	24	
RS24	24	

#### 17.2.2.3.2 语音 PI 帧

表 75 语音 PI 帧定义

信息单元	长度	描述
<b>FID</b>	<b>8</b>	<b>00010000</b>
初始向量 IV	48	
KeyID	8	密钥 ID 值
<b>Voice Key</b>	<b>5</b>	<b>00100, 语音加密模式标志</b>
<b>ALGOID</b>	<b>3</b>	<b>加密算法</b>
<b>Reserved</b>	<b>8</b>	
CRC16	16	

#### 17.2.2.3.3 语音 F 帧 EMB

语音 F 帧 EMB 区域填充的是 32 比特的信息。该部分内容由 KeyID 8bits 和 ALGOID 3bits 经过 BPTC (32,11) 编码, 再进行交织后填充到 F 帧的 EMB 区域。在选定信道的语音通信过程中, 一旦 CPS 设置确认后 F 帧 EMB 区域信息保持不变。

### 17.3 数据加密

数据加密采用增强型加密方式, 其加密参数定义方式与语音加密保持一致, 加密参数传递方式与语音加密有差异。

#### 17.3.1 短信加密时序

如下图定义了常规系统下短信业务的时序图, 相比于常规短信多了一个数据二头传递加密参数, 包括初始向量、KeyID、ALGOID。

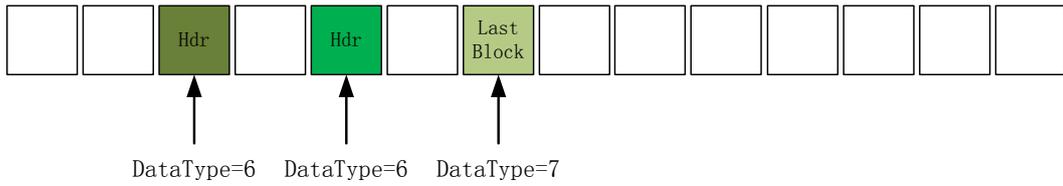


图 132 非确认短信加密时序图

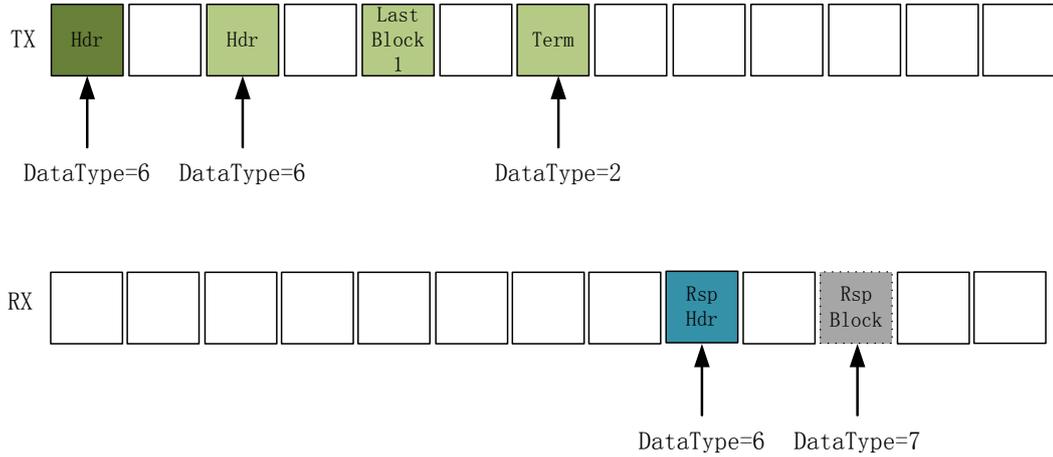


图 133 确认短信加密时序图

17.3.2 短信加密参数传递

短信加密参数传递使用了自定义短信帧头的发送，其结构如下图所示。

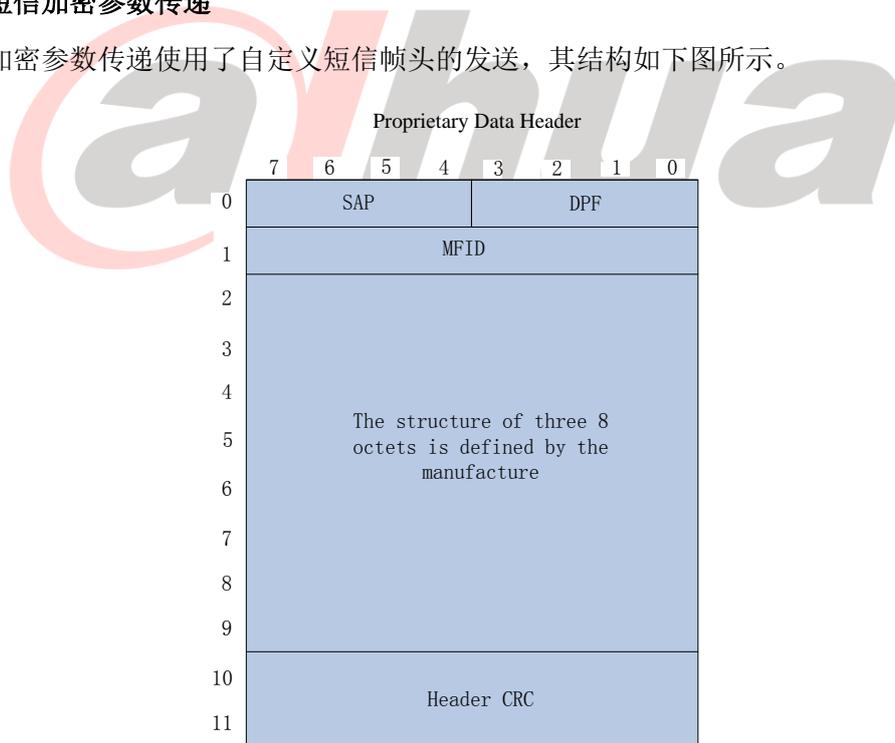


图 134 自定义数据帧头 PDU 结构图

表 76 Privacy Data Header

Information element	Length	Value	Remark
SAP	4	0100	常规短信为 0100

DPF	4	1111	
MFID	8	00010000	
初始向量 IV	48		
KeyID	8		
Data Privacy	5	00010	
ALGOID	3	001	
Reserved	16	0000000000000000	
Header CRC	16		

## 18 附录 E：数字 DCDM 应用

### 18.1 概述

DCDM 模式应用于对讲机工作在直通模式下实现双时隙通话的功能。通常情况下，为了两个时隙同时并独立地传输，一般需要中继台来提供时间基准，而 DCDM 模式下对讲机可以自动的同步和协作，不再需要中继台提供时间基准。因此现在可以同时使用两个时隙，容量倍增并提高频率效能而不必购买中继台和相关辅助设备。

### 18.2 基本原理

DCDM 模式使用独立的 TDMA 方式传输分享一个信道，有一点必须要考虑即保持两个 MS 传输过程中使用对齐的时隙边界以避免时隙间的相互干扰。为了降低时隙间的干扰几率，所有在广域系统中的 MS 单元使用相同的信道时隙定义是很重要的。一台在广域系统中的 MS 被指定为广域时隙领导，并且设置广域系统信道的时隙。其它的 MS 单元相互分享 TDMA 时隙信息。这样就扩展了时隙信息从领导 MS 到系统中的所有 MS 单元。

#### 18.2.1 状态机类型

DCDM 模式下使用了 Leader MS 的定义，具体在实现过程中就涉及到 4 个高级状态，分别是领导者和时隙未知状态、领导者未知状态、领导者和时隙已知状态以及领导者状态。下文将一一进行描述说明。

##### 18.2.1.1 领导者和时隙未知状态

在领导者和时隙未知状态下，MS 不知道当前信道的时隙状态。该状态通常发生在 MS 开机启动或者 MS 改变不同的工作频率。一台 MS 在同一工作频率下改变信道时隙从时隙 1 到时隙 2 的情况下它将保持当前的时隙状态。此外，MS 也不知道当前信道时隙的领导者。在该状态下 MS 将有如下三种的处理方式：获取领导者和信道时隙、指定一个领导者以及被指定为领导者。

##### 18.2.1.2 领导者未知状态

在领导者未知状态下，MS 通过此前获取得到时隙信息。但是 MS 不知道当前信道时隙的领导者，并且也不能够再假定先前的时隙领导者接收当前的时隙领导者。在该状态下 MS 具有如 3.1.1 状态同样的处理方式。当 MS 需要传输时，MS 将以上一次知道的信道时隙发送一最低程度降低时隙间的干扰。

### 18.2.1.3 领导者和时隙已知状态

在领导者和时隙已知状态下，MS 知道当前的信道时隙和信道时隙领导者。在该状态下 MS 有如下的五种策略方式：指定一个领导者、被指定为领导者、接受一个新的领导者、发送领导者时隙推送、发送领导者时隙纠正。

### 18.2.1.4 领导者状态

在领导者状态下，MS 是时隙领导者并且设置信道时隙。在该状态下 MS 有如下五中策略方式：指定一个新的领导者、接受一个新的领导者、解决领导者 ID 冲突、发送信标信号、发送领导者时隙纠正。

## 18.2.2 数据帧定义

### 18.2.2.1 同步帧定义

当传输广域的定时信息时，MS 将使用规定时隙的同步字样与其它 MS 单元联合在一起。如下表定义了 TDMA Direct Mode 模式下的同步字段定义。

表 77 同步帧字段定义

TDMA direct mode time slot 1													
Voice	Hex	5	D	5	7	7	F	7	7	5	7	F	F
	Binary	0101	1101	0101	0111	0111	1111	0111	0111	0101	0111	1111	1111
Data	Hex	F	7	F	D	D	5	D	D	F	D	5	5
	Binary	1111	0111	1111	1101	1101	0101	1101	1101	1111	1101	0101	0101
TDMA direct mode time slot 2													
Voice	Hex	7	D	F	F	D	5	F	5	5	D	5	F
	Binary	0111	1101	1111	1111	1101	0101	1111	0101	0101	1101	0101	1111
Data	Hex	D	7	5	5	7	F	5	F	F	7	F	5
	Binary	1101	0111	0101	0101	0111	1111	0101	1111	1111	0111	1111	0101

### 18.2.2.2 色码定义

与 DMR 其它工作模式一样，DCDM 模式下也定义使用了色码。两个时隙的色码设置可以是相同的也可以是不同的。此外，一个“*All Site*”色码 (F) 被定义使用在 DCDM 系统中。在 DCDM 系统中接收到一个色码值为 F 被认为是合格的色码。这个“*All Site*”色码只被应用于传输信道定时 CSBK 控制包时，这个包可能是在时隙 1 或者时隙 2。MS 可以接收全域的色码在规定的时隙和非规定的时隙上。这样处理减少了为支持广域时隙传输所需要的 CSBK 包的数量。

### 18.2.2.3 Channel Timing CSBK

信道定时信息是通过一个 Channel Timing CSBK (CT\_CSBK) 包传输的。当接收到一个 CT\_CSBK 的时候 MS 将评估 CT\_CSBK 信息元素包括领导者 ID、领导者动态标识符、世代、同步时期、信道定时操作码、新领导者标志、源动态标识符、源 ID。然而，由于所有的发送可能不在时隙边界开始的，这样就要求 MS 解码非对齐时隙方式传输的帧。

如果 MS 获取到接收到的信道定时信息指示比已知的定时信息要好，则 MS 将使用新接收到的 CT\_CSBK 信息去设置信道定时信息。如果 MS 获取到的几首信道定时信息不比已知的定时信

息要好，则 MS 将继续使用当前信道时隙定时。CT\_CSBK 可能会被接收在 MS 单元规定的时隙上也有可能非规定的时隙上。因此 MS 将接收任意时隙的 CT\_CSBK 包。这样就可以减少需要传输的 CT\_CSBK 包的数量。下文将列举 5 种不同类型的 CT\_CSBK 包。

#### 18.2.2.3.1 CT\_CSBK\_Beacon

领导者 MS 周期性地传输的信道定时 CT\_CSBK 叫信标 CT\_CSBK (CT\_CSBK\_Beacon)。CT\_CSBK\_Beacon 在领导者指定的时隙上使用合适的时隙同步字样和全域色码，以 12.5KHz 的直通模式下的礼貌性地尝试策略发送。如果当前信道在任意时隙上存在 RF 活动，则 CT\_CSBK\_Beacon 将不会被立即发送。这样能够确保 CT\_CSBK\_Beacon 将携带正确的信道时隙定时被发送。如信道存在 RF 活动，MS 将排队等待在一段延时后重新尝试发送。一台 MS 接收到 CT\_CSBK\_Beacon，如果接收到是广域定时领导者则其将以此设置其信道时隙。两次 CT\_CSBK\_Beacon 发送之间的间隔叫做信标间隔。

#### 18.2.2.3.2 CT\_CSBK\_Prop

CT\_CSBK\_Prop 可以被认为是强制传播消息。CT\_CSBK\_Prop 包主要应用于非领导者 MS 在接收到 CT\_CSBK\_Beacon 或者 CT\_CSBK\_Prop 后，通过 CT\_CSBK\_Prop 包扩散传输正确的广域定时信息到所有的在地理范围以内的 MS 单元。CT\_CSBK\_Prop 在指定的时隙上使用合适的时隙同步字样和全域色码，以 12.5KHz 的直通模式下的礼貌性地尝试策略发送。如果当前信道在任意时隙上存在 RF 活动，则 CT\_CSBK\_Prop 将不会被立即发送。这样能够确保 CT\_CSBK\_Prop 将携带正确的信道时隙定时被发送。如信道存在 RF 活动，MS 将排队等待在一段延时后重新尝试发送。一台 MS 接收到 CT\_CSBK\_Prop，如果接收到是广域定时领导者信息则其将以此设置其信道时隙。

为了减少信道上传输的 CT\_CSBK 包数量，CT\_CSBK\_Prop 在一些特定的情况下将会被取消发送。一种滑动窗的 CT\_RHOT 策略是有必要被用于扩散不同 MS 单元的传输 CT\_CSBK\_Prop 消息，这样信道时隙定时信息将会更大可能地到达广域范围。一个 MS 发送完 CT\_CSBK\_Prop 后，下一次发送 CT\_CSBK\_Prop 包的时间使用制定的 CT\_RHOT 值分布在下限 2.16 秒到上线 3.24 秒使用 60ms 增量。如果 MS 某次取消了 CT\_CSBK\_Prop 的传输，在这种情况下 MS 将减少 CT\_RHOT 表的上限和下限各 120ms。这样可以提高传输 CT\_CSBK\_Prop 的几率比先前的时间。这个 120ms 的减量使用直到预定的 CT\_CSBK\_Prop 包发送或者下限范围值到 0 时，CT\_RHOT 范围将被设置回 2.16s 到 3.24 秒。

#### 18.2.2.3.3 CT\_CSBK\_Term

CT\_CSBK\_Term 可以被认为是终止消息符。它将作为帧尾传输在特定的传输业务之后。一台 MS 将立即紧跟一个语音传输、数据传输或 CSBK 传输之后，除了一下两种情况以外：CT\_CSBK\_Term 将不会被发送在一个确认数据的应答数据帧头或者确认 CSBK 包之后如 UU\_Ans\_Rsp 用于 OACSU 个呼。

如下图定义了基本业务中关于信道定时终止帧的使用方式，其中 CT\_CSBK\_Term 将替代原有的确认短信的帧尾。此外应答的 CSBK、确认应答都不需要添加 CT\_CSBK\_Term。

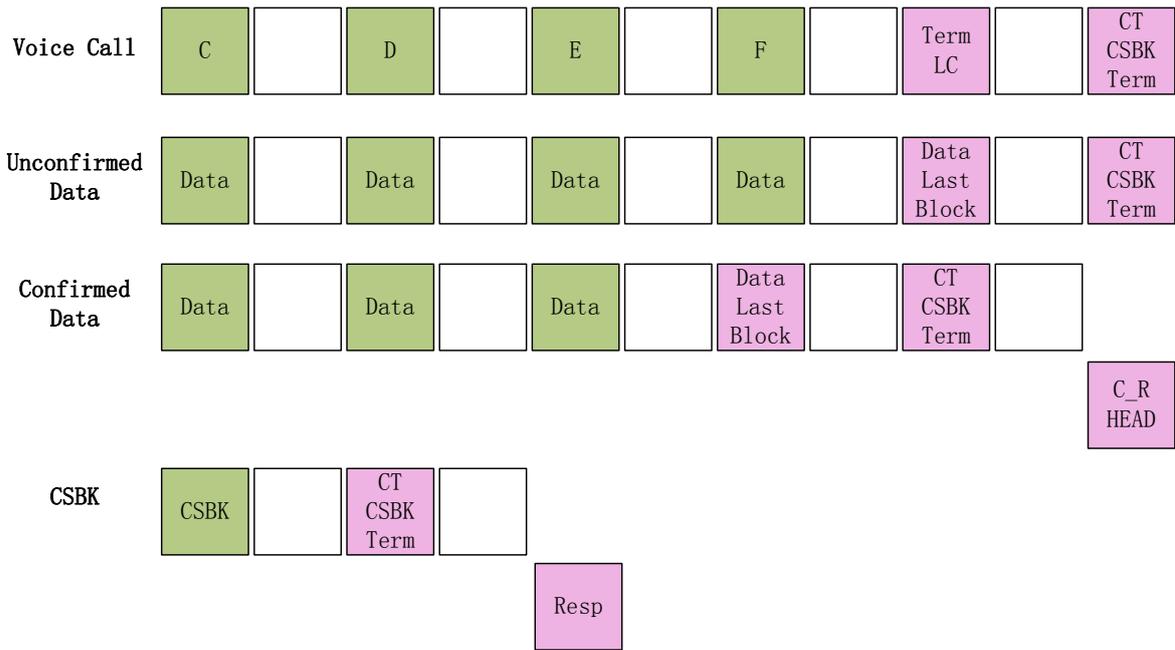


图 135 信道定时 CSBK 终止

#### 18.2.2.3.4 CT\_CSBK\_Req

CT\_CSBK\_Req 请求包用于发送请求获知当前领导者和信道定时信息。在发送之前，MS 采用随机等待时间 CT\_RHOT。CT\_CSBK\_Req 的发送只在信道空闲时候发送。

#### 18.2.2.3.5 CT\_CSBK\_Resp

CT\_CSBK\_Resp 用于被传输作为当前领导者和信道定时信息请求的应答，同时也被用于作为一个纠错信息帧当 MS 观察到不正确的信道定时信息时。发送采用的处理机制也是随机等待计时器 CT\_RHOT。

### 18.2.3 DCDM 状态流程

本章节将对 DCDM 模式下的状态流程加以描述，主要涉及到 4 个主状态关于领导者和定时信息确认的过程。

#### 18.2.3.1 领导者和时隙未知状态

如下图描述了对讲机开机启动或者是从其它信道切换到 DCDM 模式信道下，对讲机需要处理的流程。MS 需要初始化它自己的广域定时 ID (MS\_WATID)，它是由 MS\_DI 和 MS\_ID 两部分组成的。同时还需要设置世代 (MS\_Gen)、同步时期 (MS\_SA) 以及它的领导者 WATID，这三个变量初值均设置为 0。初值设置完成后，MS 需要启动领导者搜索定时器，进入领导者和定时未知的状态。

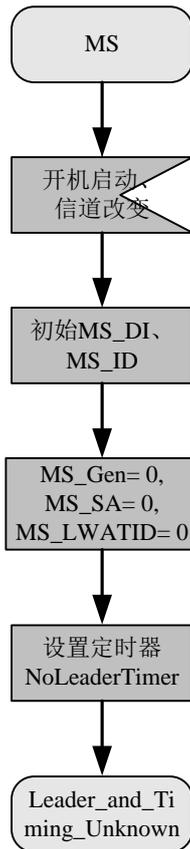


图 136 开机和信道改变状态流程图

### 18.2.3.2 领导者和时隙未知状态流程

如下图描述的是领导者和时隙未知下的状态流程图。主要涉及到三个状态分支流程。

- 1、当 MS 在该状态下提出发送请求，MS 将建立发送流程，但是由于 MS 是在不知领导者前提下发送的，则该业务发送的结尾 CT\_CSBK\_Term 终止帧时将使用 SDI 为 0 的值。
- 2、MS 接收到 CT\_CSBK 帧，则将进入 CT\_CSBK 帧评估分析状态；
- 3、MS 在搜索 Leader 超时后，MS 将主动发送 CT\_CSBK\_Req 以获取所需信息，MS 将首先启动发送随机挂起等待计时器 CT\_RHOT，该状态下又存在两天分支。第一，在等待过程中若接收到 CT\_CSBK 包，判断 CT\_CSBK 包中的信息，如果接收到的 LWATID 等于 0 或者 LWATID 不为 0 且 LDI 小于 MS 的 MS\_DI 则 MS 只获取到了接收时隙信息将继续在发送 CT\_CSBK\_req 等待状态，如果接收到的 LWATID 不为 0 且 LDI 大于等于 MS 的 MS\_DI 则 MS 将释放 CT\_RHOT 定时器进入 CCE 状态。第二，当 CT\_RHOT 计时满后，如果当前信道空闲则 MS 将发送 CT\_CSBK 然后再回到领导者和时隙未知状态等待，此处 CT\_CSBK 包设置 Gen、SA、LWATID、NL、CTO 均为 0，SDI 设置为 MS\_DI，SID 设置为 MS\_ID。如果当前信道是忙的则发送拒绝 MS 重新设置 CT\_RHOT 尝试发送，尝试发送的时间上限为 2 分钟后，2 分钟后将取消发送返回最初状态。

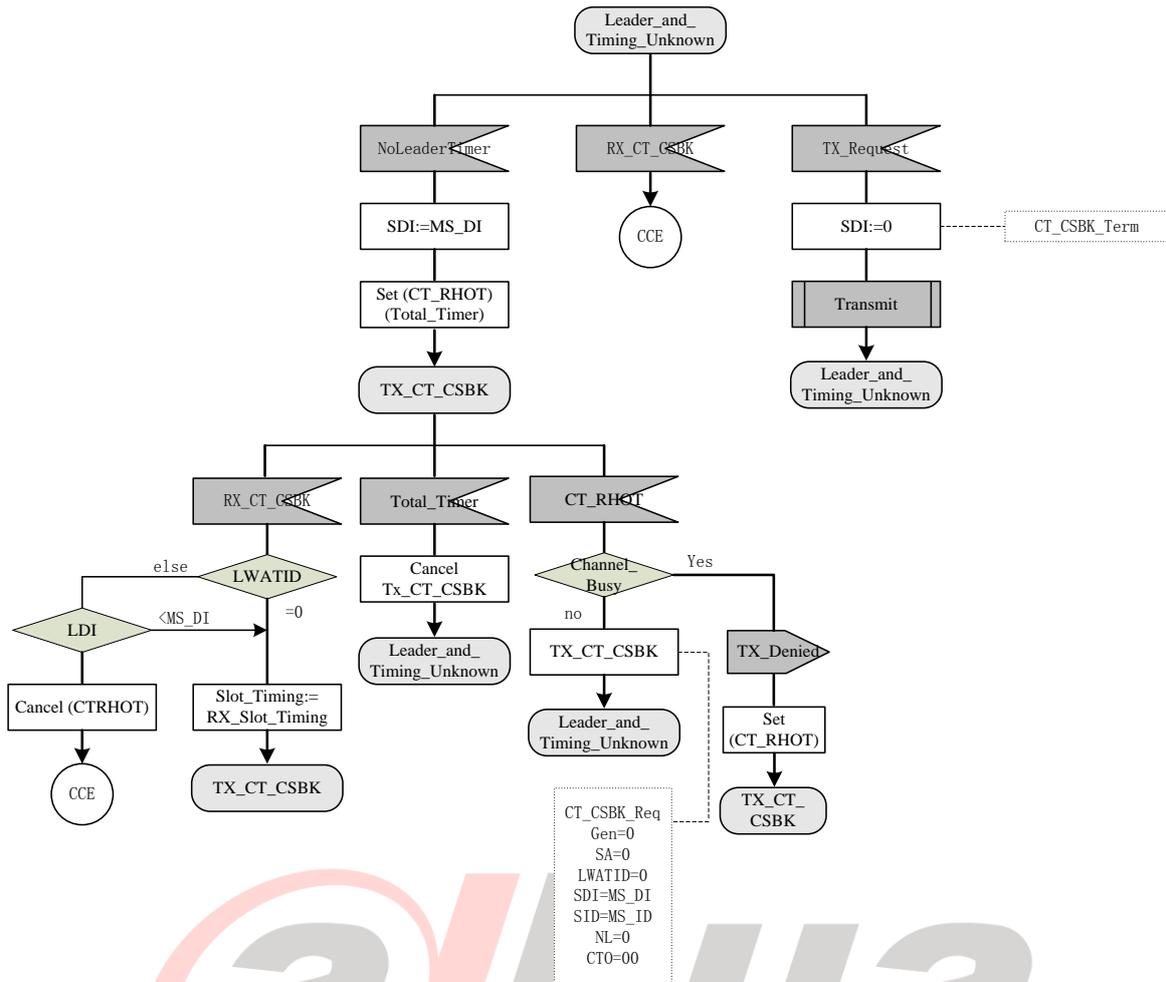


图 137 领导者和时隙未知状态流程图

### 18.2.3.3 领导者未知状态流程

如下图定义了时隙已知领导者未知下的状态流程图。其处理的流程与领导者和时隙均未知的情况处理类似。主要涉及到三个状态分支流程。

- 1、当 MS 在该状态下提出发送请求，MS 将建立发送流程。
- 2、MS 接收到 CT\_CSBK 帧，则将进入 CT\_CSBK 帧评估分析状态；
- 3、MS 在搜索 Leader 超时后，MS 将主动发送 CT\_CSBK\_Req 以获取所需信息，MS 将首先启动发送随机挂起等待定时器 CT\_RHOT，该状态下又存在两个分支。第一，在等待过程中若接收到 CT\_CSBK 包，判断 CT\_CSBK 包中的信息，如果接收到的 LWATID 等于 0 或者 LWATID 不为 0 且 LDI 小于 MS 的 MS\_DI 则 MS 只获取到了接收时隙信息将继续在发送 CT\_CSBK\_req 等待状态，如果接收到的 LWATID 不为 0 且 LDI 大于等于 MS 的 MS\_DI 则 MS 将释放 CT\_RHOT 定时器进入 CCE 状态。第二，当 CT\_RHOT 计时满后，如果当前信道空闲则 MS 将发送 CT\_CSBK 然后再回到领导者和时隙未知状态等待，此处 CT\_CSBK 包设置 Gen、SA、LWATID、NL、CTO 均为 0，SDI 设置为 MS\_DI，SID 设置为 MS\_ID。如果当前信道是忙的则发送拒绝 MS 重新设置 CT\_RHOT 尝试发送，尝试发送的时间上限为 2 分钟后，2 分钟后将取消发送返回最初状态。

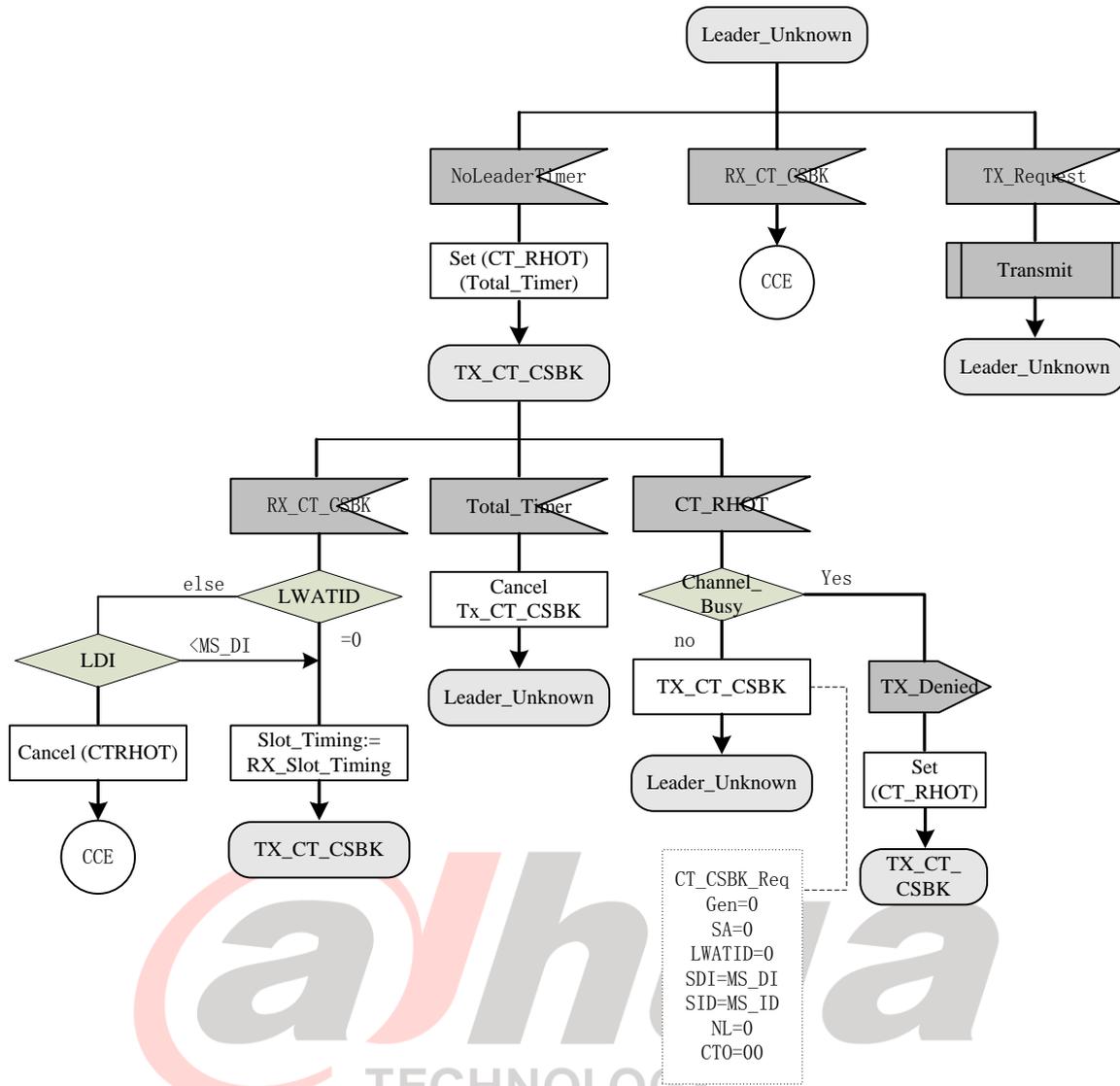


图 138 领导者未知状态流程图

#### 18.2.3.4 领导者和时隙已知状态流程

如下图描述了 MS 在领导者和时隙已知情况下状态流程图，主要有四条分支描述说明。

- 1、当 MS 在该状态下提出发送请求，MS 将建立发送流程；
- 2、当 MS 接收到 CT\_CSBK 帧，则将进入 CT\_CSBK 帧评估分析状态；
- 3、当 SyncAgeWarning（失去同步信息之前值为 9 分钟）定时器计满后，MS 将主动发送 CT\_CSBK 请求以获取更新同步信息，MS 将启动发送等待计时器 CT\_RHOT，在发送等待过程中若 MS 接收到 CT\_CSBK 包，判断 CT\_CSBK 包的信息，一种情况 CTO 为 10 或者 11 并且 LWATID 大于 MS\_LWATID，另一种情况 CTO 为 10 或者 11 且 LWATID 等于 MS\_LWATID 并且 SA 小于 MS\_SA，上述两种情况下 MS 将释放 CT\_RHOT 进入 CCE 状态，其余接收到的 CT\_CSBK 情况下 MS 将保持当前 CT\_CSBK 的发送状态不变。在 CT\_RHOT 满足之后，MS 将根据当前的信道空闲状态选择是否发送，MS 尝试发送的最大时间为 2 分钟。
- 4、当 SyncAge 满足是，MS 将释放已知状态信息，重新设置 NoLeaderTimer 进入领导者未知状态。

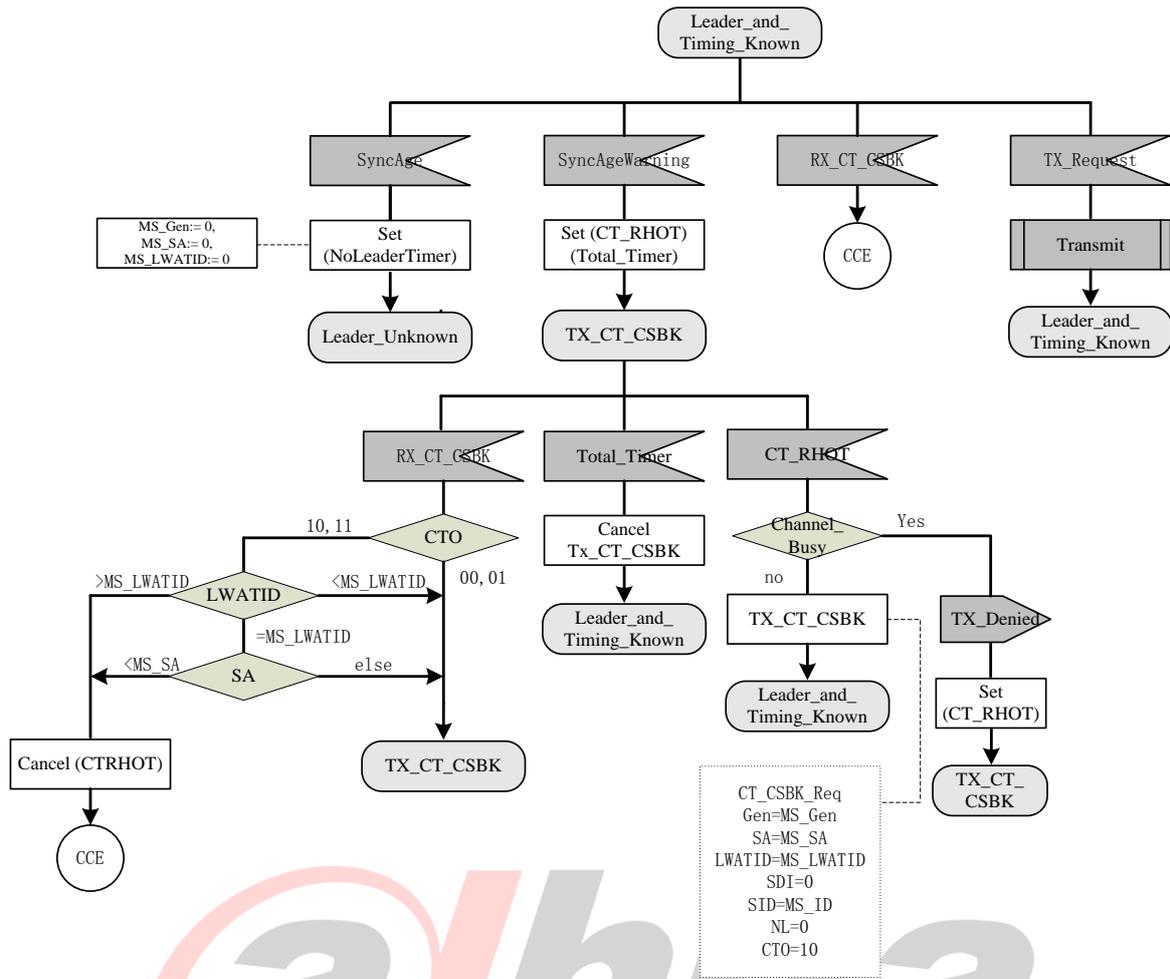


图 139 领导者和时隙已知状态流程图

18.2.3.5 领导者状态流程

如下图描述了 MS 处于领导者状态下的工作处理流程,主要涉及到三个分支,接收 CT\_CSBK 处理流程、发送请求以及发送信标间隔 CT\_CSBK。主要关于接收到的 CT\_CSBK 处理过程描述如下: 首先判断接收到的 LWATID 与 MS 的 MS\_LWATID 比较, 存在三个分支:

- 1、当接收到的 LWATID 大于 MS\_LWATID 时, 则 MS 接收接入接收领导者处理流程;
- 2、当接收到的 LWATID 等于 MS\_LWATID 时, 则 MS 进入领导者 ID 冲突处理流程;
- 3、当接收到的 LWATID 小于 MS\_LWATID 时, 则需要比较接收到的 SDI 与 MS\_SDI, 如果 SDI 大于 MS\_SDI, 则 MS 进入接受新的领导者处理流程, 反之则进入发送 CT\_CSBK 纠错信息状态处理流程。

此过程也同样应用于 LDR 状态的判断处理过程, 即领导者 CT\_CSBK 信息评估分析。

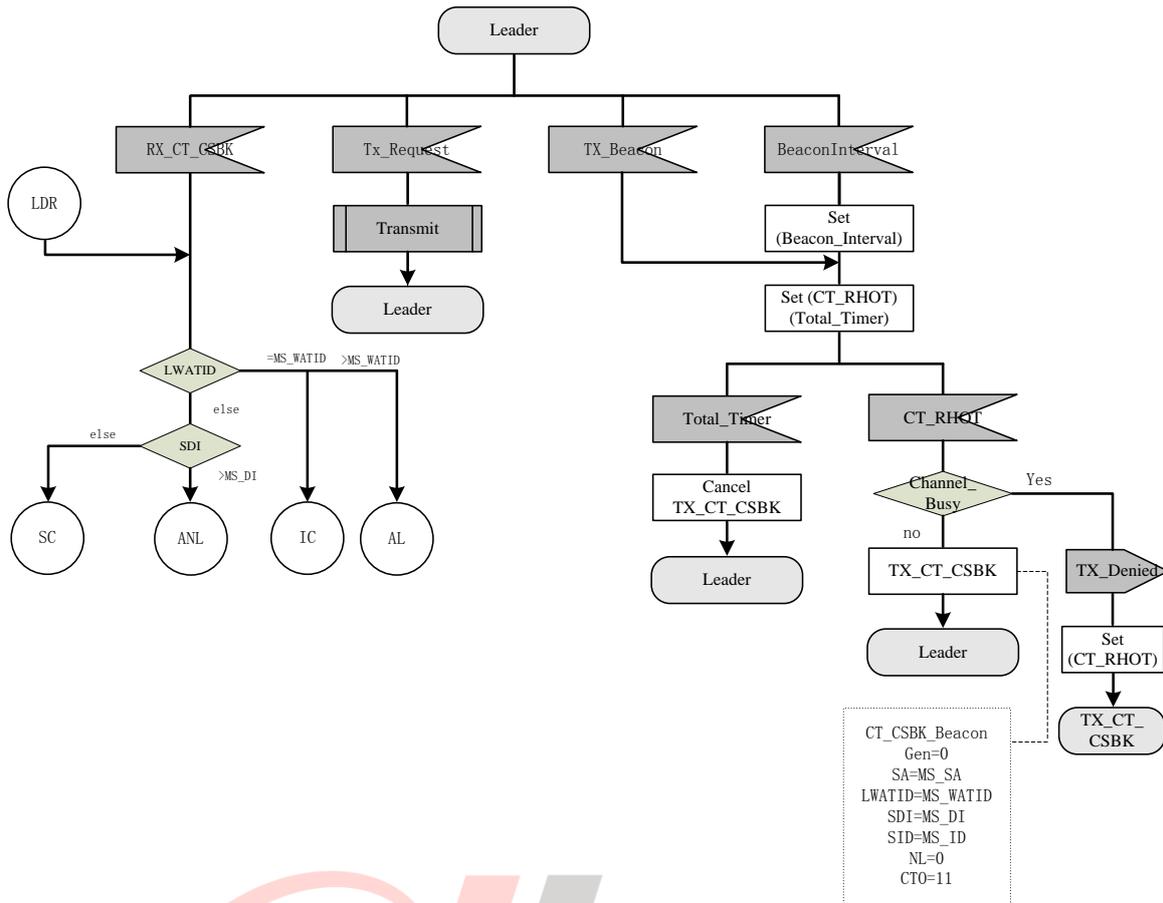


图 140 领导者状态流程图

### 18.2.3.6 领导者 ID 冲突状态流程

如下图描述的领导者 ID 冲突情况领导者 MS 的处理流程图。领导者 ID 冲突出现在领导者接收到的 CT\_CSBK 的 DI、ID 信息与 MS 相同情况下，其处理的主要判断过程如下：

- 1、如果接收到的 SA 与 MS\_SA 相差为+-5 秒范围之内，则 MS 将仍然认为其为领导者；
- 2、如果接收到的 SA 与 MS\_SA 相差为 5 秒的倍数，则 MS 将仍然认为其为领导者并进入发送纠错定时 CSBK 状态；
- 3、除了上述两种情况以外，MS 将产生一个新的 ID 与当前 MS\_ID 不同的前提下，比较接收到的 LID 与 NewMSID，如果 NewMSID 大于 LID 则 MS 仍然为领导者，否则只能接收领导者。

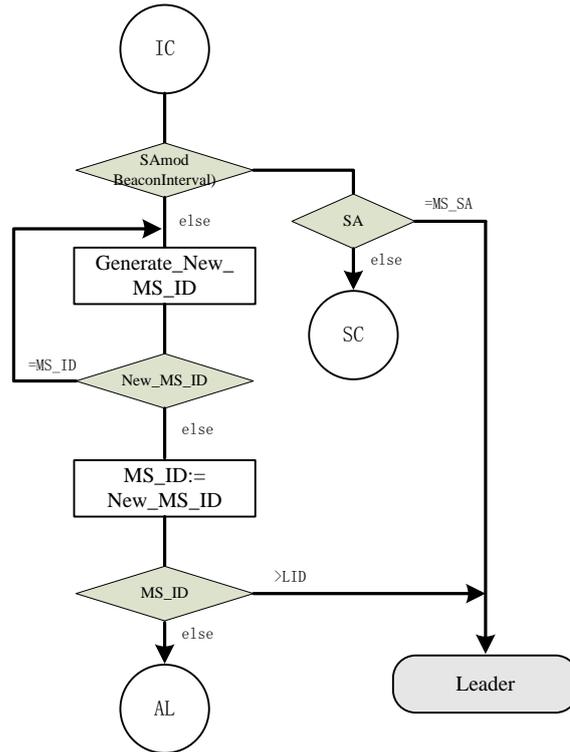


图 141 领导者 ID 冲突状态流程图



18.2.3.7 CT\_CSBK 信息评估状态流程

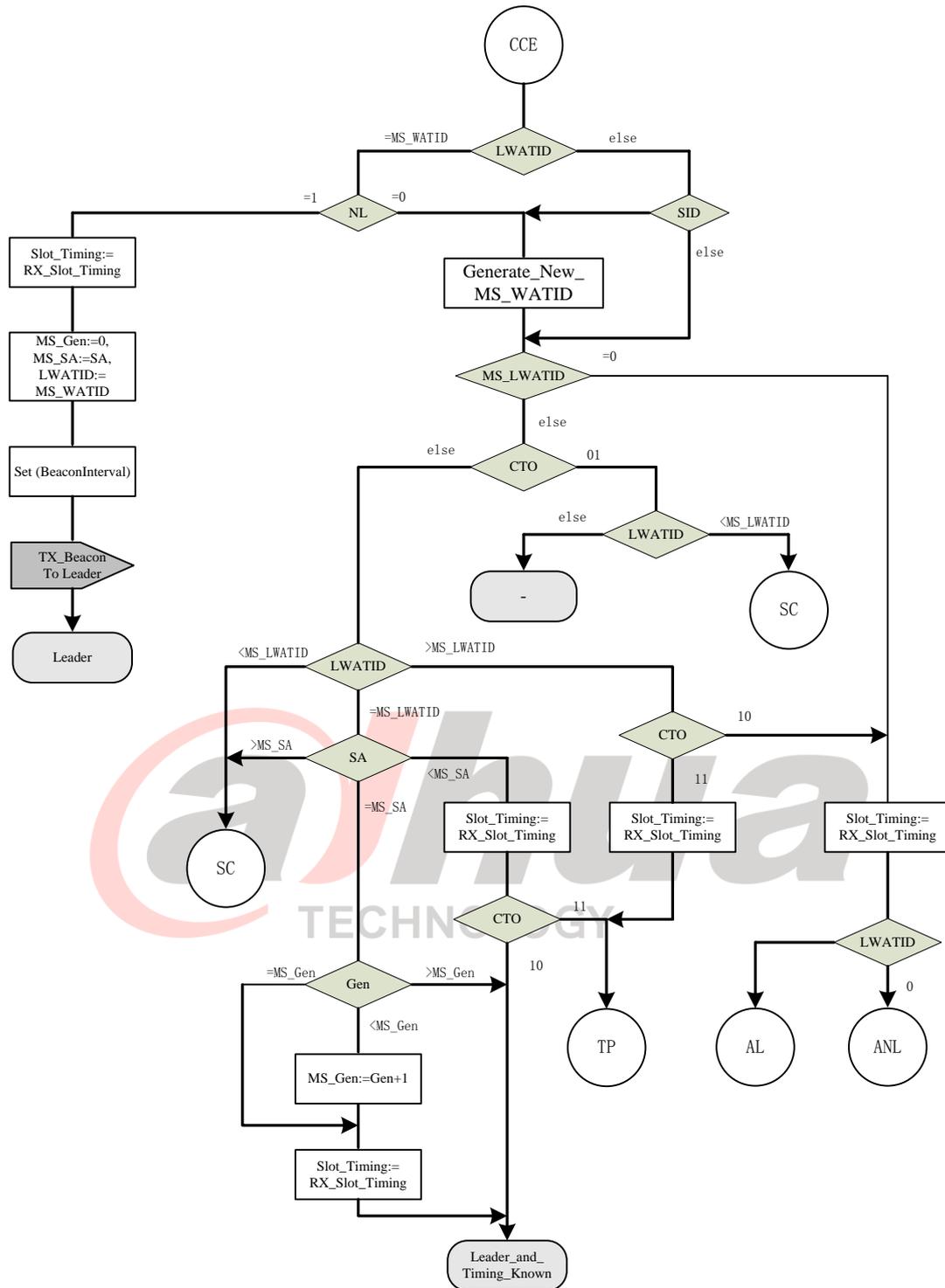


图 142 CT\_CSBK 信息评估状态流程图

18.2.3.8 发送纠正 CT\_CSBK 状态流程

如下图描述的是发送纠正 CT\_CSBK 的状态流程图。

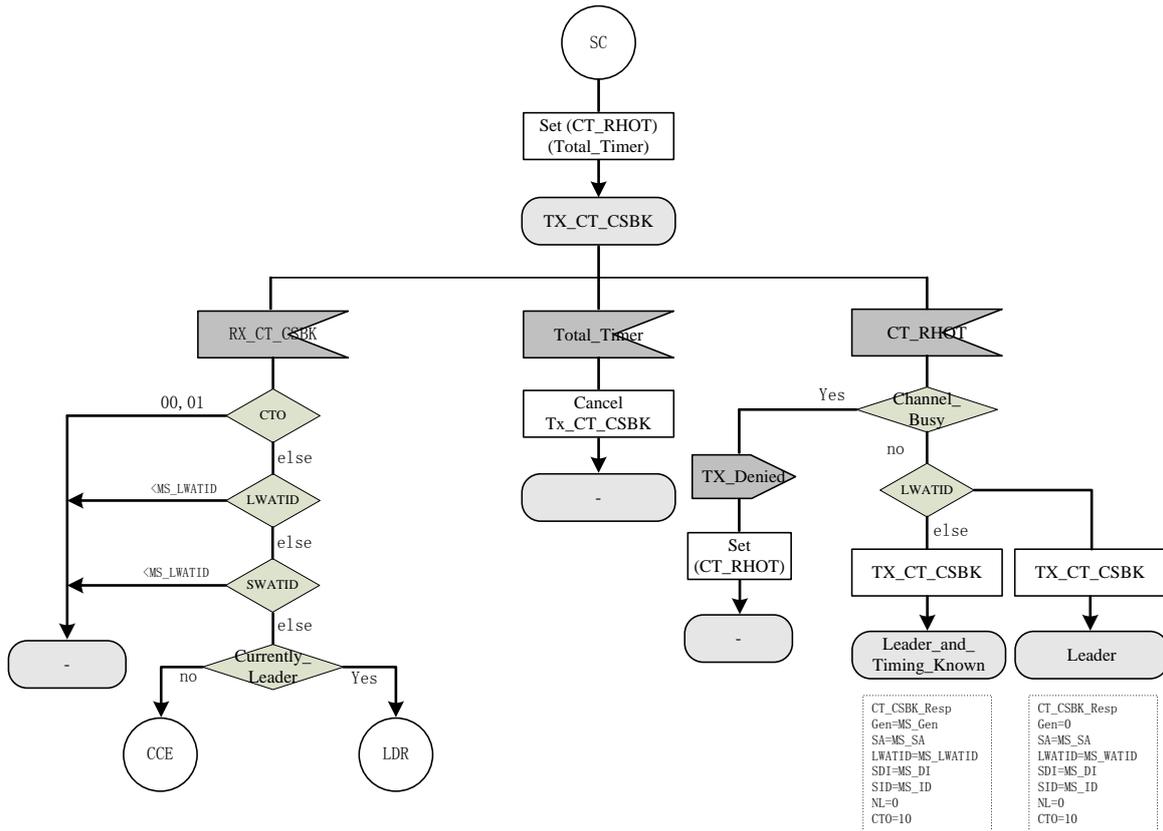


图 143 发送纠正 CT\_CSBK 状态流程图

### 18.2.3.9 接受领导者状态流程

如下图描述的是接受领导者的状态流程图。MS 通过接收到的 CT\_CSBK 中的 CTO 字段判断是否接受领导者 MS。

如果接收到的 CTO 为 00 或者 01，则 MS 将不接受领导者；

如果接收到的 CTO 为 10 或者 11，MS 将接受信道定时信息和信道定时参数，MS 将设置 SyncAge 和 SyncAgeWarning 定时器初值为接收到的 SA 值。

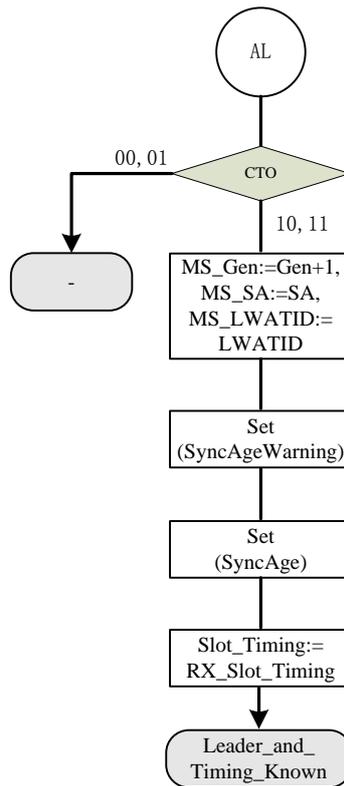


图 144 接受领导者状态流程图

### 18.2.3.10 指定新领导者状态流程

如下图描述的是指定新领导者的状态流程。

- 1、 首先设定 Gen 为 1，设定 MS\_LWATID 为接收到的 SWATID，MS 启动 CT\_RHOT 进入发送 CT\_CSBK 等待状态；
- 2、 MS 在随机等待状态过程中接收到 CT\_CSBK，将对定时信息进行判断处理如下：如果接收到 LWATID 大于 MS\_LWATID 或者接收到的 SWATID 大于 MS\_LWATID，则 MS 将释放 CT\_RHOT，取消发送；其余情况下 MS 将维持原有的状态；
- 3、 当 CT\_RHOT 满足时，MS 判断信道空闲状态发送，如果信道忙 MS 将重新设置 CT\_RHOT，启动重新尝试发送，最大尝试时间为 2 分钟。

备注：这里在指派 Leader 之前添加了一级关于 SDI 的优先级判断，在协议标准资料中未涉及到这一点，但是实际在测试过程中发现存在该功能的限制。

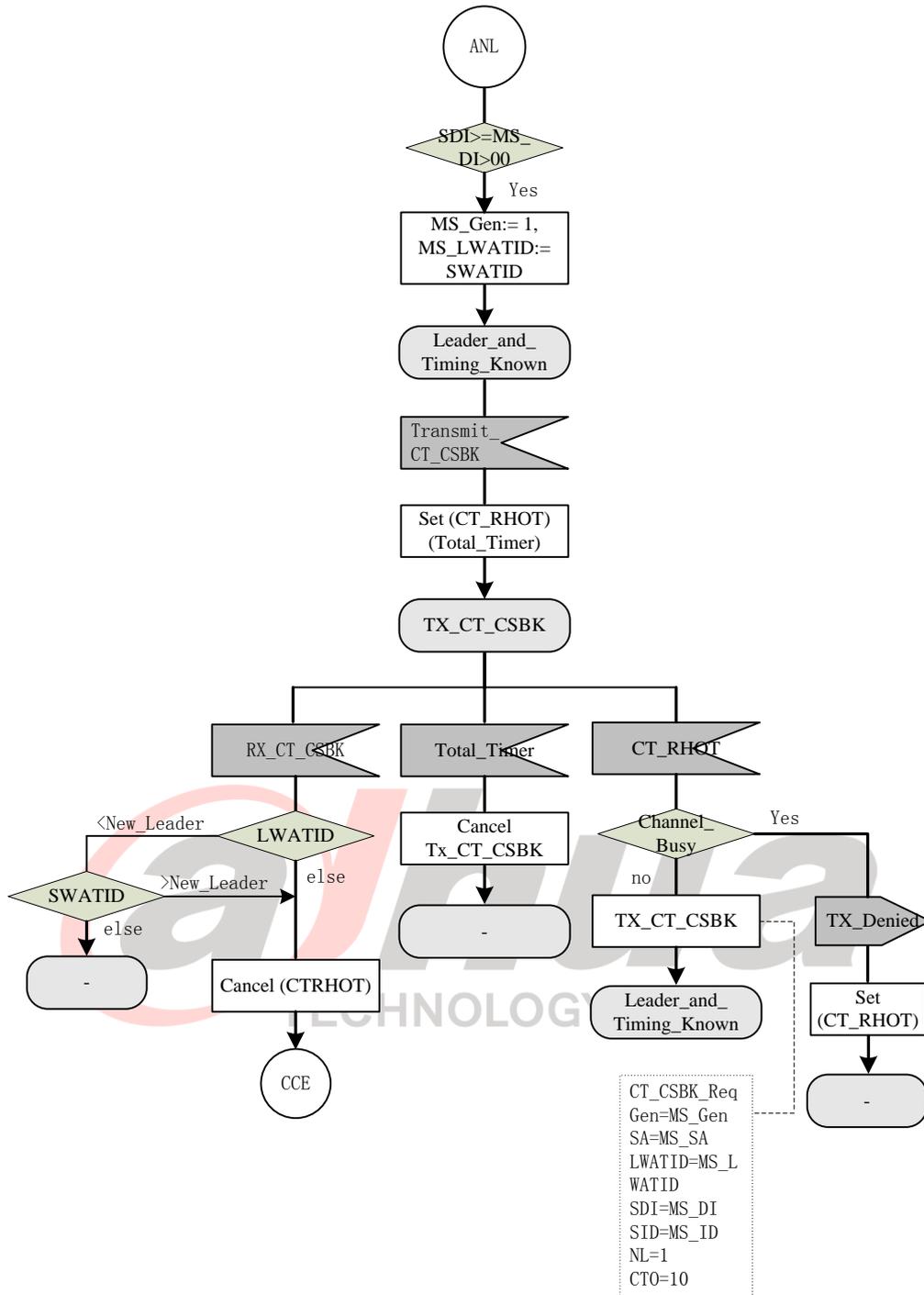


图 145 指定新领导者状态流程图

### 18.2.3.11 定时推送状态流程

如下图描述了信道定时推送发送的状态流程图。

- 1、首先 MS 从接收到的 CT\_CSBK 中获取定时参数，Gen、SA 等，其中 Gen+1；
- 2、MS 将启动 CT\_RHOT 定时器启动尝试发送 CT\_CSBK；
- 3、MS 进入 CT\_CSBK 发送状态后，在 CT\_RHOT 未满足之前若接收到 CT\_CSBK 包，则 MS 将分析 CT\_CSBK 包的信息做如下判断处理：如果接收到的 LWATID 大于

MS\_LWATID 或者接收到的 SWATID 大于 MS\_WATID, 则 MS 将取消发送推送 CT\_CSBK, 进入 CCE 状态; 如果接收到的 LWATID 等于 MS\_LWATID, 并且 CTO 等于 11, SA 等于 MS\_SA 则 MS 将释放 CT\_RHOT, 取消推送发送回到领导和时隙已知状态; 其余情况下 MS 将维持原有状态。

- 4、当 CT\_RHOT 满足后, MS 将判断时隙空闲状况, 若信道空闲则 MS 将发送定时推送 CT\_CSBK, 若信道忙 MS 将发送拒绝重新启动发送等待, 总的最大尝试时间为 2 分钟。

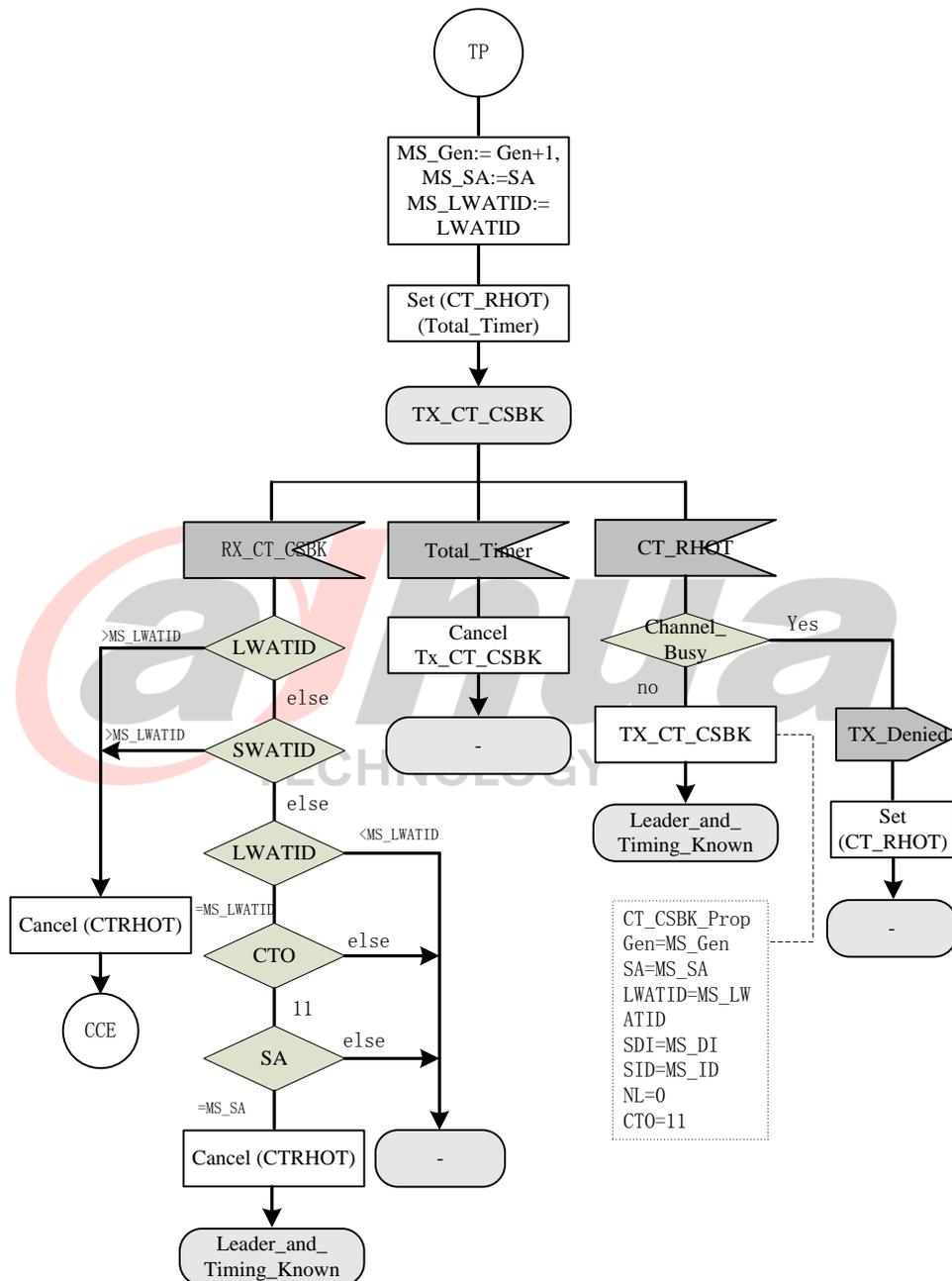


图 146 定时推送状态流程图

## 18.3 应用实例

DCDM 模式实现其关键点在于信道定时信息的设置和获取，即 Channel Timing CSBK 的收发流程处理。对应的发送类型如上一章节描述，主要存在五种类型的 CT\_CSBK 传输，其中 CT\_CSBK\_Beacon 用于 Leader MS 周期性间隔发送，CT\_CSBK\_Prop 用于非领导者 MS 信道定时推送，CT\_CSBK\_Term 用于常规业务的结束终止符，CT\_CSBK\_Req 用于获取定时信息发送请求，CT\_CSBK\_Resp 用于发送纠正信道定时信息。

DCDM 模式下的 MS 主要会经历上文描述的四个工作状态，本章节不做详细地描述。

### 18.3.1 基本单元

在 DCDM 功能模式下，主要会涉及到几个重要的基本单元，该信息来源于 CT\_CSBK 信令中，本节将对一些最基本的单元概念加以解释说明。

#### 18.3.1.1 时隙单元

DCDM 模式是以 TDMA 的方式工作，分别定义了 TDMA Slot1 和 TDMA Slot2，通常由写频软件设置时一个信道只能设定一个工作时隙，对应的工作时隙设定后，对讲机将以对应的时隙同步帧发送和对应的时隙同步帧业务接收。

关于时隙定义及相关说明如下：

- 1、 DCDM 模式下，时隙设定后，对应的发送同步帧类型以时隙定义区分，接收可以分别接收两个时隙的同步帧类型，但是对于语音、数据等功能业务的接收只能处理设定的工作时隙，对于时隙的获取还是 CT\_CSBK 信息的获取可以在另外一个时隙接收判断；
- 2、 对于同频率下两个时隙间信道切换过程，DCDM 模式下其对应的信道定时相关信息不改变；
- 3、 对于同频率下两个时隙信道切换时，当切换到另一个时隙时，对讲机将在另一个时隙上发送 TP（信道定时推送信息）；
- 4、 对于不同频率下两个信道切换或开机时，对讲机将从最初始的状态获取信道相关信息；
- 5、 同频率下两个时隙信道的色码设置可以是不同的，但是对于 CT\_CSBK 帧必须使用色码为 F。

#### 18.3.1.2 领导者

DCDM 模式系统下，Leader MS 是整个系统运行的关键，无论是刚开机的还是工作中的对讲机，其工作过程中的第一步是通过各种方式获取信道定时信息，包括用于建立时隙的边界和 Leader MS 的标识信息。每一个系统中的对讲机根据其对应的优先顺序和系统工作状态，可以指定和接受 Leader 的信息。整个系统在工作过程中可能会因为系统中的 Leader 丢失而进行重新的指定定义。

作为 DCDM 系统中的 Leader，其主要需要间隔性发送信标消息用以维持系统的信道定时信息（详细可以参考领导者状态流程）。

#### 18.3.1.3 世代

如下表为 CT\_CSBK 中的世代 (Generation) 的定义，世代为 0 是作为 DCDM 系统的领导者，是 DCDM 系统工作时唯一的一代。

表 78 世代 (Gen) 定义

信息单元	长度	值	描述
------	----	---	----

Generation	5	0	MS 为领导者或者领导者未知时设置
		其它值	

如下图简单举例说明了世代所对应的范围覆盖图，其中领导者的 0 世代为唯一的。

通常情况的流程如下：

- 1、第 0 代无线电设备发送信标消息；
- 2、第 1 代无线电设备接收到信标消息后，发送强制传播消息扩散信道定时信息；
- 3、第 2 代无线电设备接收到第 1 代强制传播消息后，发送强制传播消息；
- 4、以此类推扩散覆盖范围。

需要指出的是第  $n+1$  代无线电设备只能从第  $n$  代无线电设备接收信道定时信息，一旦出现跨代的接收则认为信息是不正确的。

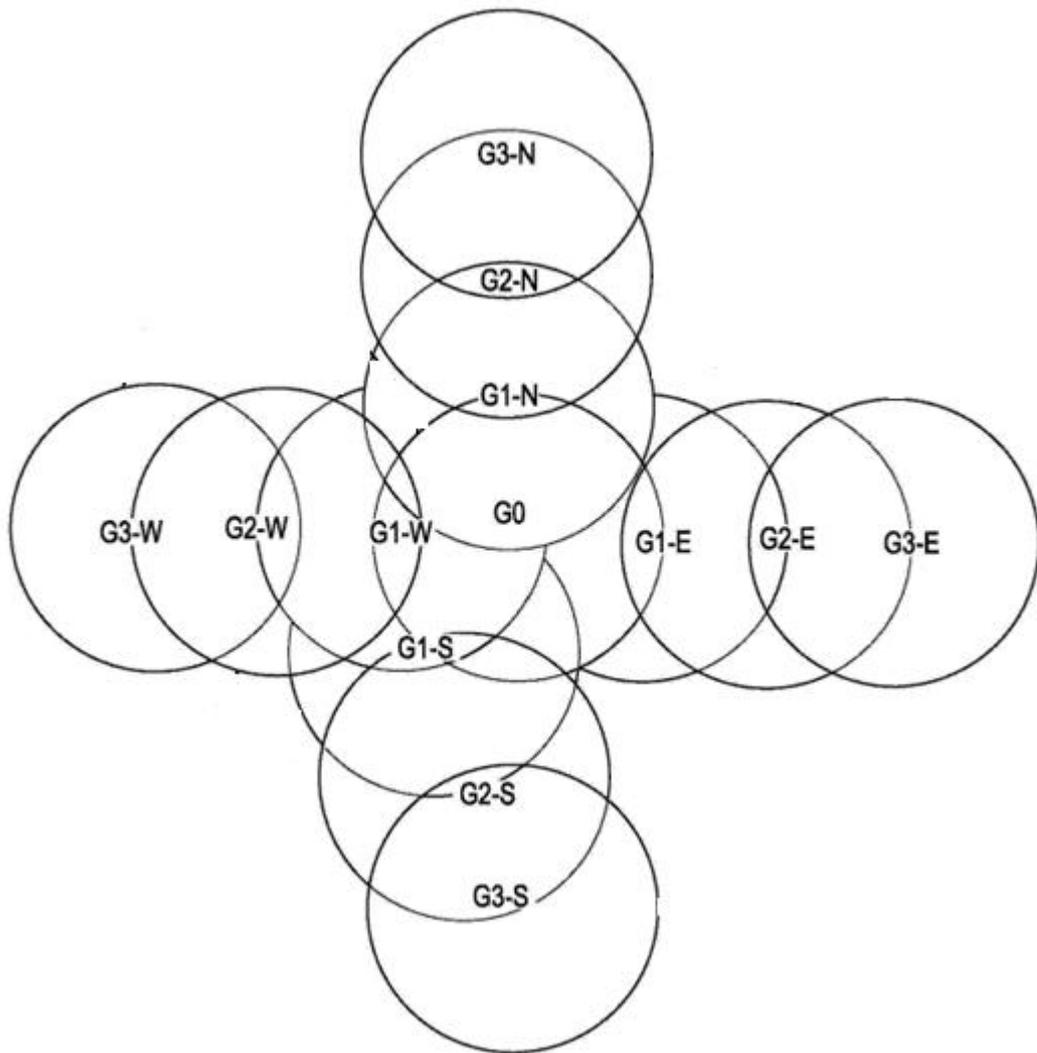


图 147 DCDM 模式下多个无线电设备覆盖图

此外，需要解释说明两点：

- 1、如上图所示，当第 0 代发送信标消息后，按照正常机制，对应的第 1 代设备 G1-N、G1-S、G1-W、G1-E 都将被要求发送强制传播消息，如何解决 4 个第 1 代设备的发送冲突问题，这里提出了一个叫随机延时挂起的概念，系统中假设 G1-N 发送了 CT\_CSBK 信息，则对应的其余几个设备在接收到该信息后将自动取消发送，在定时器定义章节将做详细说明。

- 2、第二点是关于第 0、1、2、3...代发送的时间，其对应的关系是依次发送的，如下图所示，可以把信标间隔划分为若干个世代的发送时间。

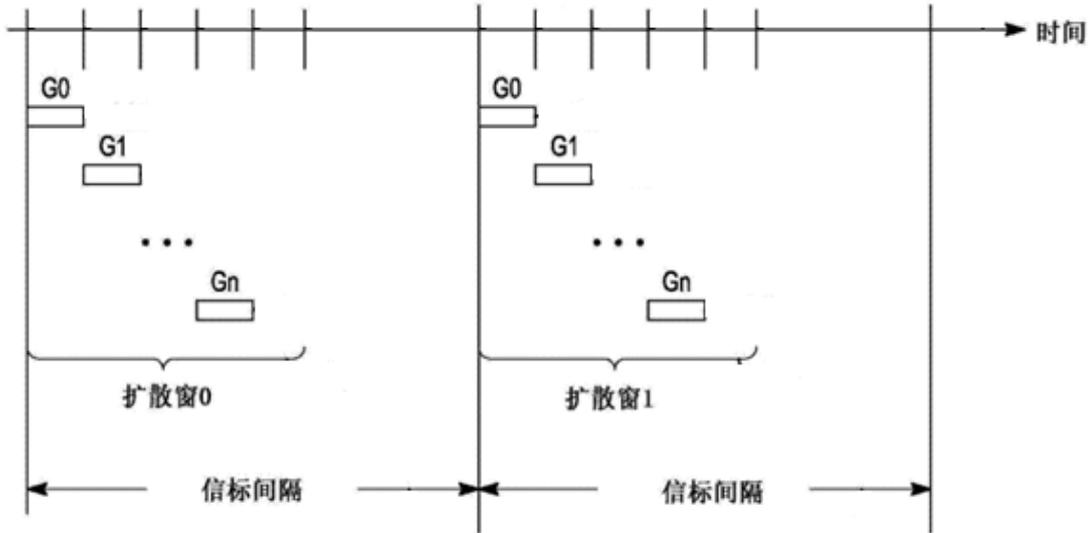


图 148 信道定时扩散时序图

#### 18.3.1.4 同步时期

如下表为 CT\_CSBK 中的同步时期 (Sync Age) 的定义。

表 79 同步时期 (SA) 定义

信息单元	长度	值	描述
Sync Age	11	0	MS 为领导者或者领导者未知时设置
		其它值	自从上次信标中的 Sync Age 增量, 增量为 500ms

表格中 0 值的描述是按照协议标准中指出的, 但是通常情况下, 领导者在发送信标消息时对应的 SA 值并不一定为 0 值, 其原因是由发送 CT\_RHOT 计时器的存在。

SA 的使用需要配合 SyncAge 和 SyncAgeWarning 两个定时器。

#### 18.3.1.5 标识符

DCDM 系统中的标识符主要有 ID 标识符、动态 (DI) 标识符以及由两者组成的 WATID, 如下文所述。

同时为了区分定义领导者和其它设备, DCDM 系统中定义了领导者标识符和源标识符。在上一章节 2.3 中定义的变量说明如下:

- LDI: 领导者动态标识符;
- SDI: MS 源动态标识符;
- MS\_DI: MS 的初始动态标识符;
- LWATID: 接收的领导者 WATID
- MS\_WATID: MS 源 WATID
- MS\_LWATID: MS 的领导者 WATID

WATID 是由 DI 和 ID 两部分组成的, 定义为广域的定时 ID, 通常对于 MS 来说存在自己所在系统的 MS\_LWATID 和 MS\_SWATID, 系统中用于评估信道定时的重要依据为接收的 LWATID 和 SWATID。

表 80 WATID 定义

信息单元	长度	值	描述
Wide Area Timing Identifier	2		DI 值
	20		ID 值

如下表定义了 DCDM 模式下 CT\_CSBK 的 ID(Identifier)字段的定义说明。ID 区分为 Leader ID 和 Source ID, 对于 Leader MS 来说其 LID 和 SID 是相同的, 对于非领导者 MS 其 LID 为系统中的 Leader ID, 对于领导者未知情况下均设置为 0 值。

表 81 Identifier (ID) 定义

信息单元	长度	值	描述
Identifier	20	0	开机或者信道改变时源 ID 设置为 0。 领导者未知情况下领导者 ID 值设置为 0。
		其它值	MS 的 ID 指示

补充说明一点设备的 ID 值会在 DCDM 动态系统中存在改变, 当存在 ID 值的冲突情况下, 详细可以参考 2.3 章节中的状态流程图。

如下表定义了 Dynamic Identity 的列表值说明。在实际系统 CT\_CSBK 接收处理过程中, DI 是重要的定时信息判断依据。

表 82 DI 定义

信息单元	长度	值	描述
Dynamic Identifier	2	00	开机或者信道改变时 SDI 值 领导者未知情况下 LDI 值
		01	领导者优先级低
		10	领导者优先级中
		11	领导者优先级高

关于 DI 的定义补充说明一点, 每一个对讲机设备的初始 DI 值可能不一样, 但是有一点必须确定的是 DI 值为 00 的无法确认为系统的初始领导者。当系统中领导者确认情况下, 非领导者 MS 发送 CT\_CSBK 信息时其对应的 SDI 值为 00, 当系统中发生领导者丢失情况下, 其将重新回到开机的初始值。同样处于对电池的保护, MS 可能会改变其 MS\_DI 值为 00 以放弃其领导的角色。实际测试发现在长时间的测试过程中, 当 Leader 的 MS 电池处于低电量时系统中的领导者会发送改变。

#### 18.3.1.6 信道定时操作码

信道定时操作码 (Channel Timing Opcode), 通常用于指示 CT\_CSBK 的类型, 区别不同状态下的 CT\_CSBK 类型。

表 83 CTO 定义

信息单元	长度	值	描述
CT Opcode	2	00	未对准请求
		01	未对准终止符
		10	信道定时状态
		11	信道定时推送

需要指出的是, 在 CT\_CSBK\_Term 发送时, CTO 值的设置需要依据如下原则:

- 1、如果 MS 所在系统中无领导者信息时, 其值将设置为 00;
- 2、如果 MS 所在系统中有领导者信息并且当前所在信道的另一个时隙无 DMR 的业务信号时,

其值将设置为 10;

- 如果 MS 所在系统中有领导者信息并且当前所在信道的另一个时隙存在 DMR 的业务信号时，对于领导者 MS 来说设置为 10，但是对于非领导者 MS 来说需要设置为 01。

### 18.3.1.7 新领导者标志

表 84 New Leader 定义

信息单元	长度	值	描述
NL	1	0	MS 接收当前领导者
		1	当发送时，MS 指定新领导者； 当接收时，目标 MS 被指定为新的领导者

## 18.3.2 CT\_CSBK PDU

### 18.3.2.1 CT\_CSBK 结构

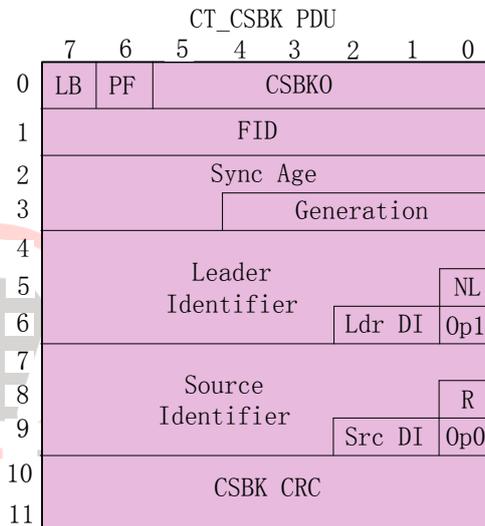


图 149 CT\_CSBK 结构图

### 18.3.2.2 CT\_CSBK 字段定义

如下表定义了 CT\_CSBK PDU 的定义列表。下文将对几个关键字段的定义加以说明。

表 85 CT\_CSBK PDU 定义

信息单元	长度	描述
LB	1	设置为 1
PF	1	设置为 0
CSBKO	6	设置为 000111
FID	8	设置为 00000000
SyncAge	11	
Generation	5	
Leader Identifier	20	
New Leader	1	

Leader Dynamic Identifier	2	
Channel Timing Op1	1	
Source Identifier	20	
Reserved	1	
Source Dynamic Identifier	20	
Channel Timing Op0	1	
CRC	16	CRC16 的校验结果信息

### 18.3.3 定时器定义

如下表定义了 DCDM 系统中使用的相关定时器。其中关于 CT\_RHOT 的定义相对来说较为特殊，下文将详细描述说明。

表 86 定时器常量定义

名称	值	描述
CT_RHOT	0~3.24 秒	CT_CSBK 传输过程随机挂起等待定时器，主要用于降低 CT_CSBK 传输的冲突。 在领导者和时隙未知情况下定义了范围 0~3.24 秒之间，增量为 60ms。 在领导者和时隙已知情况下，范围为 2.16 秒到 3.24 秒之间，增量为 60ms。 中间出现状态 CT_CSBK 传输取消，减少增量范围 120ms
NoLeader	4.5 分钟	开机或信道改变时，MS 在发送 CT_CSBK 请求之前等待搜索领导者的时间
SyncAge	10 分钟	MS 需要更新同步信息定时，增量为 500ms
SyncAgeWarning	9 分钟	MS 需要更新同步信息定时之前提醒
BeaconDuration	最小 600ms	CT_CSBK_Beacon 和 CT_CSBK_Prop 传输持续时间
BeaconInterval	4.5 分钟	信标间隔传输时间
CTDuration	最小 180ms	CT_CSBK 请求和 CT_CSBK 响应传输时间
TxTimer	2 分钟	发送 CT_CSBK 尝试的最大时间

#### 18.3.3.1 CT\_RHOT 说明

CT\_RHOT，即 CT\_CSBK transmission Random HoldOff Timer，可以理解为 CT\_CSBK 传输随机挂起等待定时器。它主要用来降低系统中 CT\_CSBK 传输的冲突概率。

根据协议标准中提到的 CT\_RHOT 的定时器范围为 0~3.24 秒，增量为 60ms，本文举例采用滑动窗均匀分布的方式，具有 1.08 秒的窗口尺寸的跨过 3.24 秒的滑动窗均匀分布，选择用于 CT\_RHOT 的超时值。具体的分布表如下表所示，对应每行定义均匀分布函数 U（下限，上限）。

表 87 CT\_RHOT 值分布表

		均匀滑动窗（单位：秒）	
N	M	下限	上限
0	18	2.16	3.24
1	17	2.04	3.12

2	16	1.92	3.00
3	15	1.80	2.88
4	14	1.68	2.76
5	13	1.56	2.64
6	12	1.44	2.52
7	11	1.32	2.40
8	10	1.20	2.28
9	9	1.08	2.16
10	8	0.96	2.04
11	7	0.84	1.92
12	6	0.72	1.80
13	5	0.60	1.68
14	4	0.48	1.56
15	3	0.36	1.44
16	2	0.24	1.32
17	1	0.12	1.20
18	0	0.00	1.08



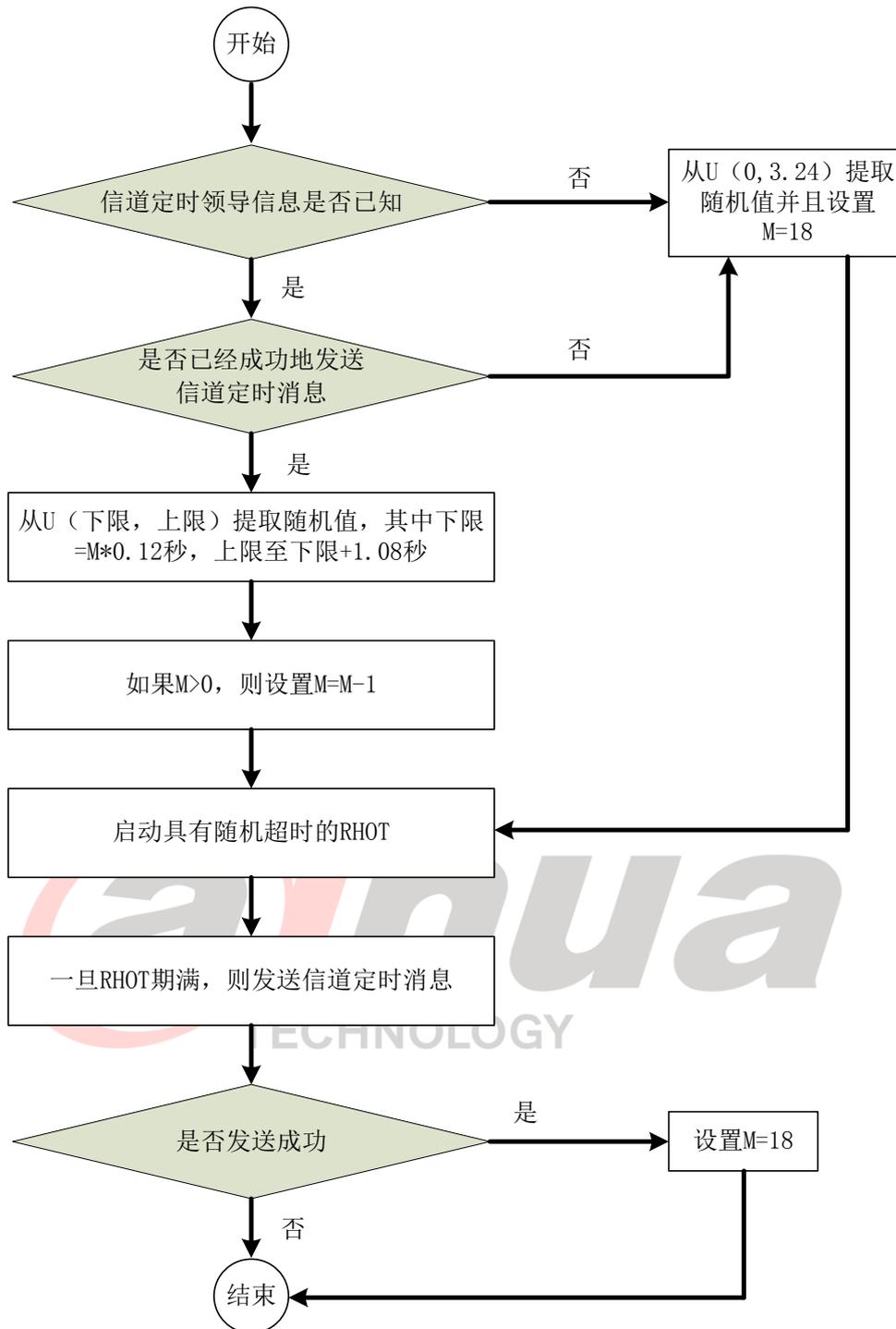


图 150 CT\_RHOT 使用流程图

如上图定义描述了 CT\_RHOT 使用的具体流程，其具体的流程如下：

- 1、 第一步判断当前是否已经知晓信道定时领导信息，如果处于未知状态，则从  $U(0, 3.24)$  中随机取值设置 CT\_RHOT，同时设置 M 对应为初始 18；
- 2、 第二步判断是否已经发送过信道定时消息，即各种 CT\_CSBK 类型帧，若无则采用与第一步未知状态下的方式；
- 3、 由已知的 M 值，在对应的  $U(\text{下限}, \text{上限})$  均匀分布函数中取随机值，然后将已知的 M 值递减；

- 4、 设置完 CT\_RHOT 值后开始启动定时器；
- 5、 CT\_RHOT 计满后开始启动发送信道定时消息；根据信息发送是否成功判断，若成功则设置 M 值为 18，若不成功则结束保留当前的 M 值。

### 18.3.4 最小化系统举例

本章节简单介绍一个最小化DCDM系统的工作流程,以DCDM系统从无到有过程简单介绍。

#### 18.3.4.1 单机开机启动

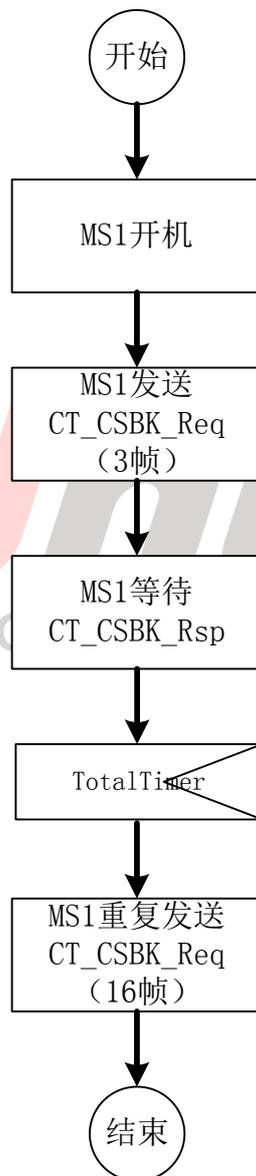


图 151 单机开机启动流程图

如上图描述了正常情况单个 MS 开机后的处理流程,即如何去获取有效的 DCDM 系统的定时信息过程,其主要的状态流程可以参考 17.2.3.1 的过程,这里做简要的概况主要分为以下几

步:

- 1、 启动 NoleaderTimer, 超时后开启发送 CT\_CSBK\_Req 请求包, 请求包以 3 帧 CT\_CSBK 为主, 对应的帧信息设置参考 17.2.3.1;
- 2、 发送完成后等待系统中的响应, 整个发送的超时时间设置为 2 分钟后;
- 3、 2 分钟后仍然未收到任何反馈信号, 则 MS 将重新发送请求信息, 实测该数据帧数量为 16 帧, 然后结束发送请求操作。  
备注, 整个请求发过程只会执行一次, 后续系统即处于空闲接收状态, 若要发送也是以 CTO 为 00 的操作码方式发送。

#### 18.3.4.2 开机领导者确认

如下图定义描述了正常情况下 DCDM 系统开机或者不同频率信道切换后领导确立过程。其对应的主要步骤如下描述:

- 1、 MS1 先开机率先满足 NoleaderTimer, MS1 发送 CT\_CSBK\_Req;
- 2、 MS2 接收到来自 MS1 的请求包, 判断当前的 CT\_CSBK 帧信息, 主要在 LWATID 为 0 零时判断其 SDI 值来确立是否具有 Leader 资格, 有一点必须说明的对于 SDI 为 00 的是无法被指定为 Leader 的;
- 3、 若 SDI 值非零, 则 MS2 认为 MS1 可以作为系统领导者, MS2 将发送 CT\_CSBK\_Rsp 用以指派 MS1 为 DCDM 系统的领导者;
- 4、 MS1 在准确接收到被指定为 Leader 的信息后, 主要以 NL=1 为标志, 确立自己为系统的领导者。
- 5、 补充说明, 在最小化系统的两台机子中, 如果率先开机的 MS1 不具备领导者的资格, 则 MS2 将随后启动与 MS1 一样的请求发送过程, 如果 MS2 也不具备的话, 则系统中将继续保持无领导者的状态, 直到有新的机子进入系统。



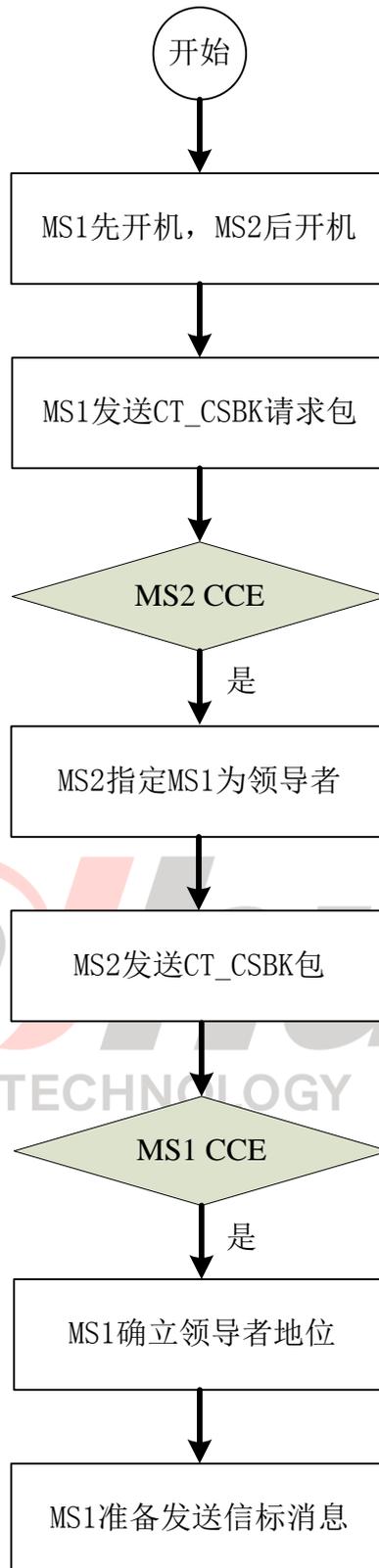


图 152 开机领导者确认流程图

#### 18.3.4.3 信标消息发送

以确立 MS1 为 Leader 前提下举例说明关于信标消息的说明。

首先需要说明的是信标消息是由 Leader 发送的，对应发送间隔为每 4.5 分钟，对应发送的时间长度为 600ms（10 帧）。

当 MS1 处于同频率下的时隙切换时，对应的 MS1 切换到另一工作时隙后将在另一时隙立刻发送信标消息。

系统中 MS2 接收到信标消息后，经过 CT\_CSBK CCE 后发送强制传播消息，也可以认为是定时推送消息，对应的参考 17.2.3.11。

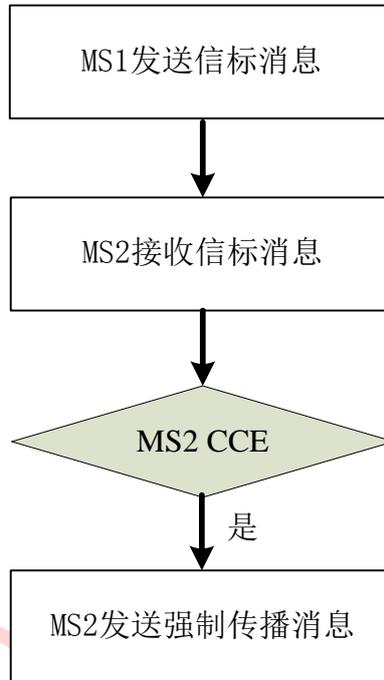


图 153 信标消息发送流程图

#### 18.3.4.4 业务发送流程

如下图描述了 DCDM 系统下业务发送的流程，主要涉及到 CT\_CSBK\_Term 的定义发送，对于 CTO 的定义。

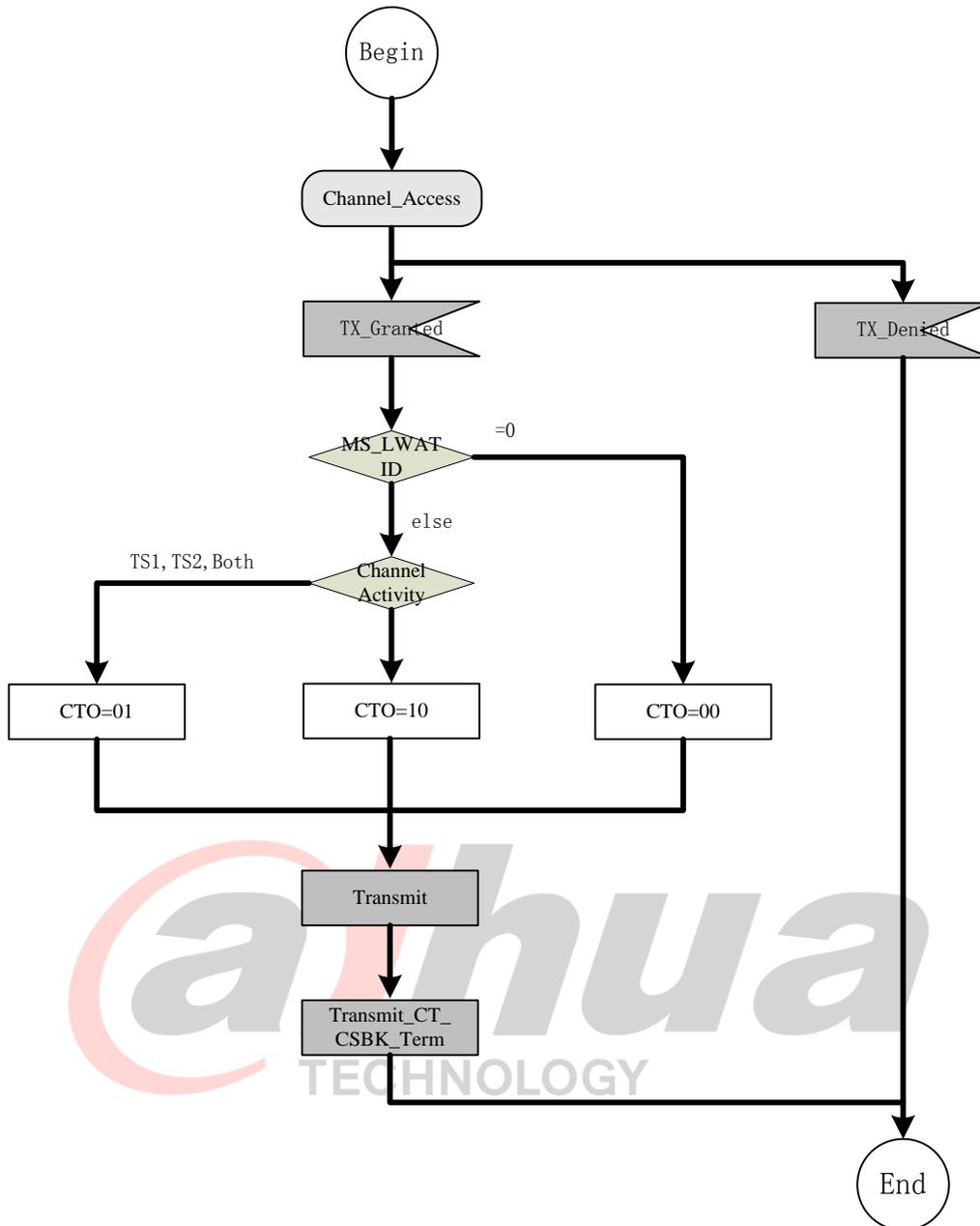


图 154 业务发送流程图

#### 18.3.4.5 Leader 脱机

以上述系统为例，MS1 关机后，系统将处于 Leader 脱机状态，MS 在上次接收到 Leader 的信标消息后启动 SyncAgeWarning 和 SyncAge 两个定时器，两者对应的时间分别为 9 分钟和 10 分钟。

定时器 9 分钟满足时对应启动发送类似收到信标消息后需要发送的强制传播消息，该帧信息中仍然包含以 MS1 为 Leader 的定时信息，对应可以参考 2.3.4 的处理流程。

定时器 10 分钟满足时 MS2 进入 LeaderUnknown 的状态，启动新的 Leader 尝试获取过程，对应可以参考 17.2.3.3。

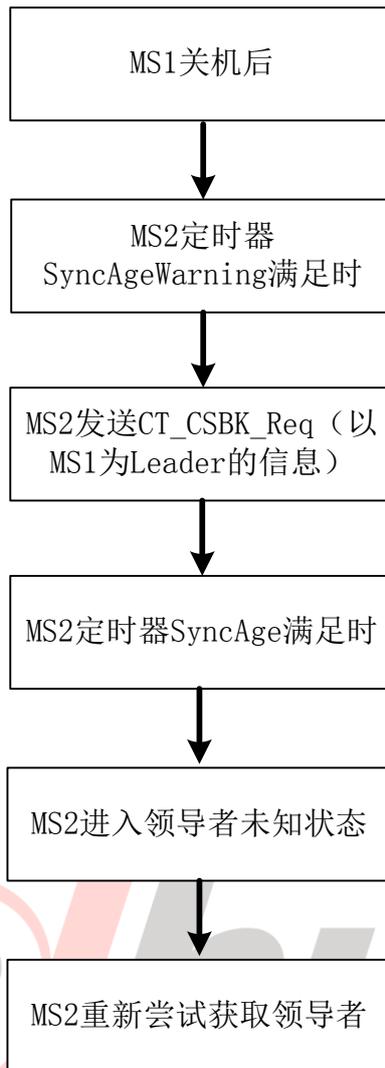


图 155 Leader 脱机后流程图

### 18.3.5 HR\_C7000 应用

HR\_C7000 支持 DCDM 功能的开发，且 HR\_C7000 工作在数字 DMR 的二层模式下实现 DCDM 相关功能的业务实现，其常规的业务帧的收发机制参考二层应用手册，本文不再详细描述。本章节主要描述 HR\_C7000 在时隙 DCDM 功能所需要的特殊应用场景，包括时隙管理、色码、同步字段控制、双接收通道设置等。

#### 18.3.5.1 同步字段应用

DCDM 模式下 TDMA1 SYNC 对应 TDMA1 SLOT，TDMA2 SYNC 对应 TDMA2 SLOT，DCDM 模式下时隙 Slot 定义需要依赖同步帧字段加以划分。

同步字段的应用主要包括发送和接收，其中对于发送来说需要分时隙交替的发送 TDMA1 和 TDMA2 SYNC，发送同步字段信息通过配置寄存器 SEND\_DATA\_SYNC\_H 和 SEND\_DATA\_SYNC\_L，配置值详见表 1。

DCDM 应用与常规手持或者中转业务彼此间不支持互通功能，因此对于 DCDM 接收机来说只支持 TDMA1 和 TDMA2 SYNC 的分时隙接收检测，不支持 MS、BS SYNC 字段接收检测。同时需要注意 TDMA1 和 TDMA2 SYNC 是独立的时隙接收同步，因此需要两套不同时隙同步接收

机制来保证 TDMA1 和 TDMA2 业务的正常接收。

DCDM 同步接收需要 HR\_C7000 进行如下设置：

- 1、 关闭 MS、BS SYNC 接收，通过配置寄存器 RECV\_MS\_SYNC\_H[31]=0，RECV\_BS\_SYNC\_H[31]=0；
- 2、 开启 TDMA1、TDMA2 SYNC 检测，配置寄存器 RECV\_TDMA1\_SYNC\_H、RECV\_TDMA1\_SYNC\_L、RECV\_TDMA2\_SYNC\_H、RECV\_TDMA2\_SYNC\_L；
- 3、 开启双通道接收机，配置寄存器 RF\_MODE[11]=1；
- 4、 接收使能需要常开。

#### 18.3.5.2 色码应用

由于 CT\_CSBK 包使用了 ALL Site 的色码 (0xF)，其不同于其它语音或者数据业务的色码，这就要求接收机要支持不同的色码接收，因此需要对于 HR\_C7000 设置成色码无需匹配即解帧处理，色码匹配的过程需要软件处理，对应通过设置寄存器 LOCAL\_CC[9:8]=1。CT\_CSBK 包发送时设置寄存器 LOCAL\_CC[3:0]=1111。

#### 18.3.5.3 时隙管理

DCDM 模式与其他模式的区别在于手台之间的时隙管理，达到互相不冲突的目的。业务流程基本其他模式一致，除了在业务中添加消息终止符。在一定区域内，时隙是有领导者所建立（主动建立），并且一致保持，除非转换为非领导者。领导者会定期发送信标，告之非领导者的时隙定时。非领导者接收到领导者的时隙信息后（被动建立），时隙不关闭，时隙会根据接收到信标进行同步调整。在时隙同步更新超时之后，还未接收到有效的时隙信标帧，就退回到领导者未知状态，但是建立的广域时隙还是不关闭，在下次发送的时候，利用最近同步的时隙发送，可以尽可能的减少时隙冲突。除非关机或切换频道，才退出真双时隙状态并关闭时隙。

在真双时隙模式下，相关定时器的时间是默认固定的，参考标准协议规定。但是，领导者信息和时隙同步的最大超时时间为 10 分钟，所以，需要保证在 10 分钟之内没有接收到时隙信标帧或时隙推送帧的情况下，依然可以保证两个时隙的信号不交叠冲突，所以，需要保证不同对讲机之间的时间精度一致性。

真双时隙模式下，手台之间通过广域标识符来区别不同的手台，以及通过广域标识符之间的关系来建立领导者，其中领导者的 DI 值一定大于等于时隙被领导者的 DI，领导者的广域标识符一定大于非领导者的广域标识符。

以下是真双时隙模式下，关于入网状态的流程控制处理。

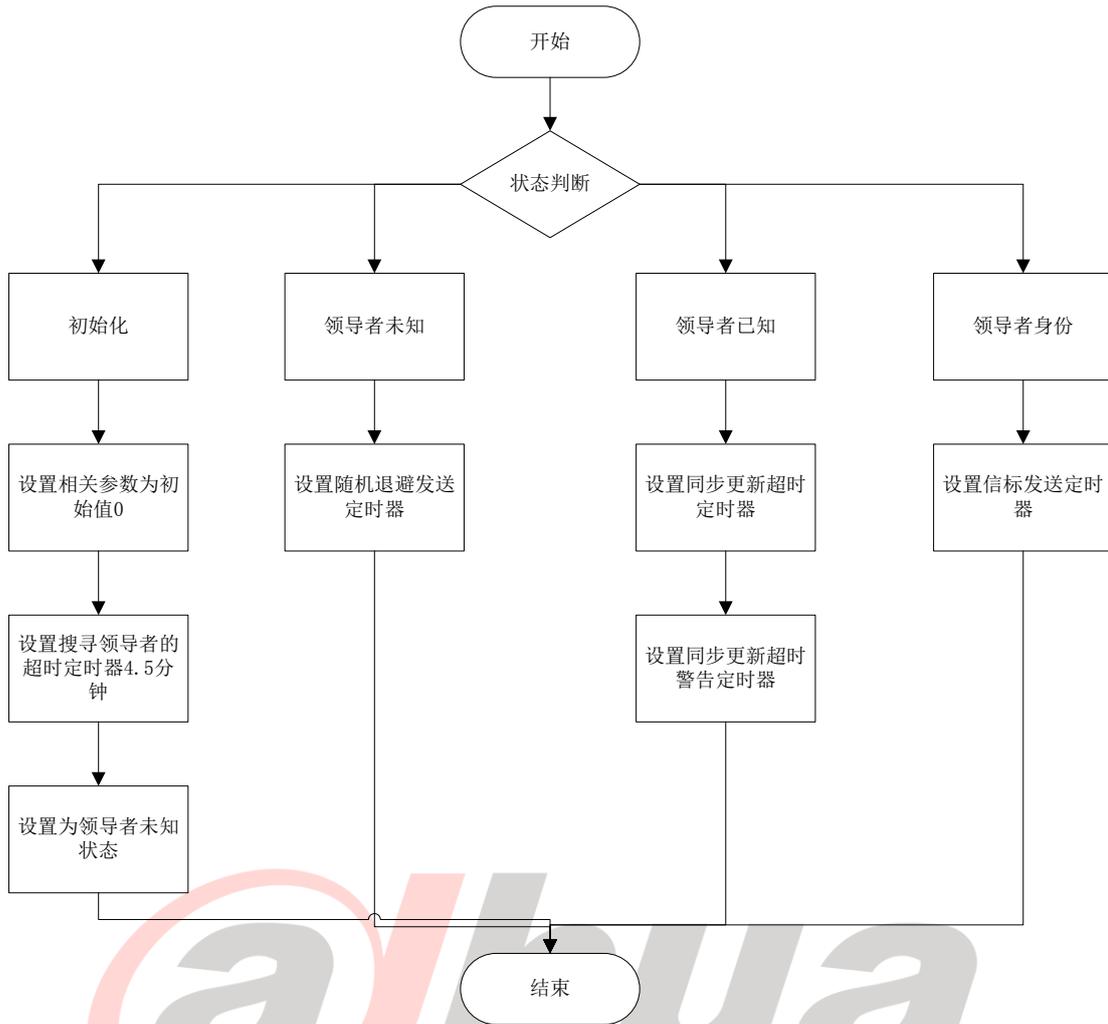


图 156 真双时隙入网状态管理流程图

DCDM 模式下，时隙管理有如下几种应用场景：

- 1、在未获取广域时隙之前接收其它未获取广域时隙整机发送的信息；
- 2、在未获取广域时隙之前接收已知领导者同步时隙整机发送的信息，此时整机可以获取真双时隙的广域时隙；
- 3、在获取真双时隙的广域时隙后接收未知时隙整机发送的信息；
- 4、在获取真双时隙的广域时隙后接收已知时隙整机发送的信息，涉及到时隙更新或者纠正；
- 5、未知时隙整机需要发送信息，直接主动建立发送；
- 6、已知时隙整机需要发送信息，就需要用获取的广域时隙轴建立时隙并发送信息；

基于上述时隙应用场景，HR\_C7000 时隙管理机制既要满足收发时隙的独立处理，又要满足发送时隙和广域建立时隙独立，以满足在空闲时刻可以接收任意时隙边界发送的信号，以实现时隙管理模式。

HR\_C7000 二层时隙管理详细参考 HR\_C7000 应用手册（DMR 数字二层应用）。

我们可以定义 DCDM 广域时隙使用 HR\_C7000 定义的发送临时时隙，DCDM 发送和接收时隙使用 HR\_C7000 各自定义的收发时隙。如此即可以满足 DCDM 时隙管理的需求。