深圳市晶之上电子科技有限公司
地址：深圳市龙华区观澜大布巷社区布新路149 电话：0755-83435615 传真：0755-88325137

承 认 书

APPROVAL SHEET

客户（CUSTOMER）：

品名（PART NAME）：

客户编号（CODE NO）：

型号（MODE NO）： SG8F7581

供应方签字

承认者	审 核	核 准

需求方签字

承认者	审 核	核 准

日期（DATE）： 2018 年 6 月 8 日

目录

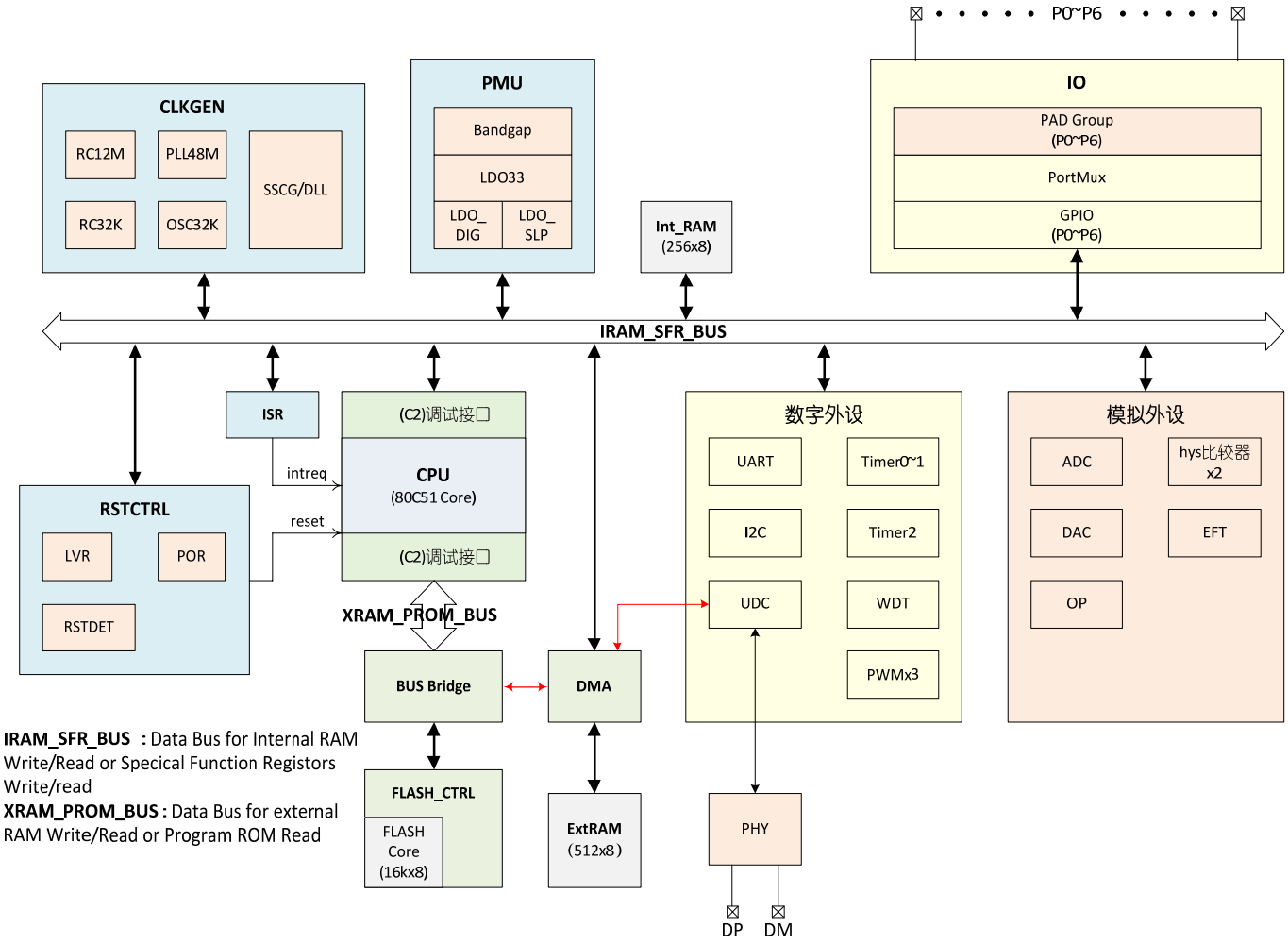
1 概述 2
1.1 主要特性 3
2 PAD MAP 及管脚说明 6
2.1 PAD MAP 6
2.2 管脚说明 7
2.3 LQFP48L 封装 Pin 图及尺寸 10
2.4 LQFP64L 封装 Pin 图及尺寸 11
3 功能描述 12
3.1 MCU 内核 12
3.2 时钟系统 26
3.3 系统模式 33
3.4 电源系统 35
3.5 中断系统 38
3.6 复位系统 46
3.7 通用 IO 控制器 49
3.8 计数器/定时器 62
3.9 PWM 74
3.10 USB 控制器 88
3.11 UART 控制器 98
3.12 I2C 控制器 107
3.13 数模转换器（DAC）/运算放大器（OPA）/电压比较器 117
3.14 ADC 转换器 127
3.15 端口复用 132
3.16 FLASH 控制器 134
3.17 DMA 控制器 141
4 修订记录 147

1 概述

SG8F7581 是一颗可用于键盘（机械背光键盘、无鬼键键盘等）以及游戏鼠标控制芯片、无线充电控制设备开发的低功耗、高性能的 8051 结构的微控制器。具有片内上电复位、LDO、时钟振荡器、看门狗、定时器、高精度 PWM、比较器和 ADC，另外还集成有 I2C 及 UART 标准通讯协议接口。芯片 FLASH 具有在系统重新编程能力，可用于非易失性数据存储，并允许 IAP(In Application Programming)。

片内的 2 线开发接口允许使用安装在最终应用系统上的产品 MCU 进行非侵入式(不占用片内资源)、全速、在线调试系统。调试器支持观察和修改存储器和寄存器，支持断点、单步、运行和停机命令。在使用片上调试系统时，所有的模拟和数字外设都可全功能运行。两个调试端口引脚可以与用户功能共享，使在系统调试功能不占用封装引脚。

SG8F7581 内部集成 8 位高性能 MCU、16K FLASH 程序存储器、256 字节内部 SRAM 以及 I/O 接口资源等。系统结构如下图



1.1 主要特性

- **基本特性**
  - 主频最高可达 48MHz 的 8 位高性能 MCU
  - 16KB FLASH 程序存储器
  - 支持在线调试、在线编程和 USB 端口烧录
  - 256Byte 内部通用 RAM 和 512Byte 扩展 XRAM
  - 支持 UART、I2C 接口
  - 支持 DMA 通讯
  - 内置高精度 PWM
  - 内部集成 2 个比较器
  - 内部集成 12 位高精度 ADC
- **FLASH 存储器**
  - 可以对特定区域实现 IAP
  - 容量 16KB
  - 每页 256Byte
  - 由 USB 端口可以对 FLASH 进行编程(需要编程器支持)
- **时钟系统**
  - 片内高速可编程 RC(12MHz)
  - 片内 PLL48M
  - 片内低速 RC(500KHz)
  - 支持系统时钟扩频技术（SSCG）
  - 片外 32KHz 晶体振荡器（可选）
  - FULL SPEED USB 应用下无需外部晶振，内置 USB 时钟恢复电路
- **电源系统**
  - 内置 5VTO3.3V LDO
  - 内置 5VTO1.8V LDO（数字系统供电）
  - 支持两种低功耗模式(IDLE/STOP)
- **GPIO**
  - 通用双向输入输出端口（电压 5V 或 3.3V 可选）

- 带有内置上拉电阻
- 可实现开漏输出
- 输出驱动能力可调（4mA/20mA 两档灌电流档位可选）

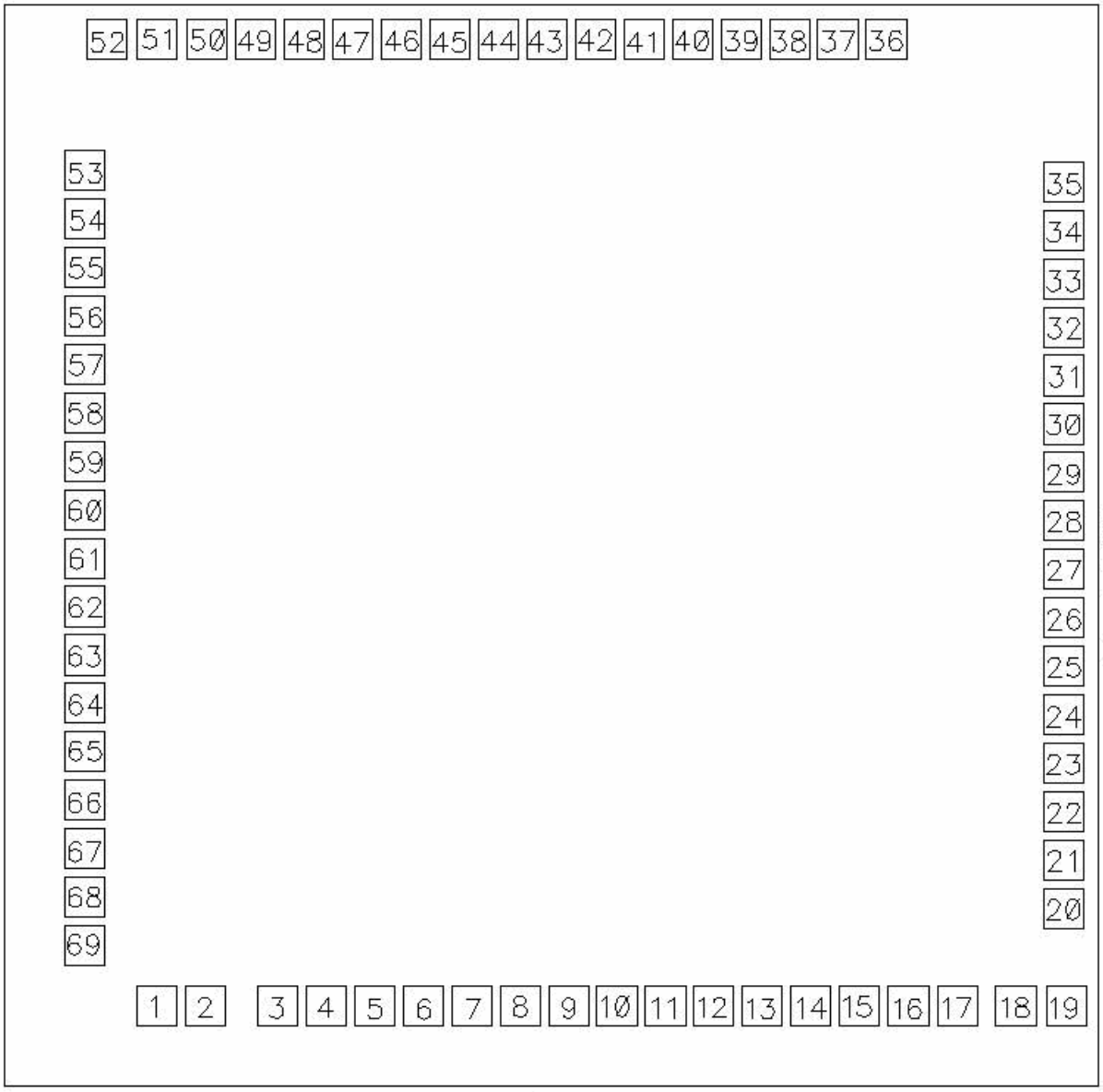
- **调试接口**
  - 全速、非侵入式的在系统调试接口(片内)
  - 单步调试
  - 可以读取或修改内部存储器及寄存器
- **USB 控制器**
  - 兼容 USB1.1 支持全速和低速两种模式
  - 支持 3 个端点
  - 支持 DMA 传输
  - 端点 0 数据深度 8byte
  - 端点 1 和端点 2 数据深度 64byte
- **UART 接口**
  - 可选择是否带奇偶校验位
  - 波特率可调
- **I2C 接口**
  - 可用作主机或从机
  - 支持 I2C STAND/FAST 模式
  - 从机模式兼容 8 位/10 位地址
- **计数器/定时器 0/1**
  - 兼容 MCS-51 的 Timer0/Timer1
  - 定时器时钟可选
  - 4 种工作模式
- 定时器 2
  - 定时器时钟可选
- **PWM0/1/2**
  - 互补输出以及死区时间控制
  - 高精度 PWM 调节功能
  - 系统时钟选择低频时，PWM 仍工作于高频
- **WDT**

- 可作为复位/唤醒源

- **ADC**
  - 12Bit 精度
  - 最大 250ksps
  - 参考电压四档可选，分别为 VDD5、VDD3O、外置或内置 2.0V
- **模拟比较器**
  - 片内集成 2 个电压比较器
  - 带迟滞控制
- **运算放大器**
  - 15 倍或 14 倍增益
  - 可配置正相放大模式（15 倍）或者反相放大模式（14 倍）
- **DAC**
  - 片内集成两个 6 位 DAC
  - 基准可选外置或者内置 1.21V

2 PAD MAP 及管脚说明

2.1 PAD MAP



PAD Diagram of SG8F7581
Substrate Size:X=2500um;Y=2400um
Substrate Connect GND

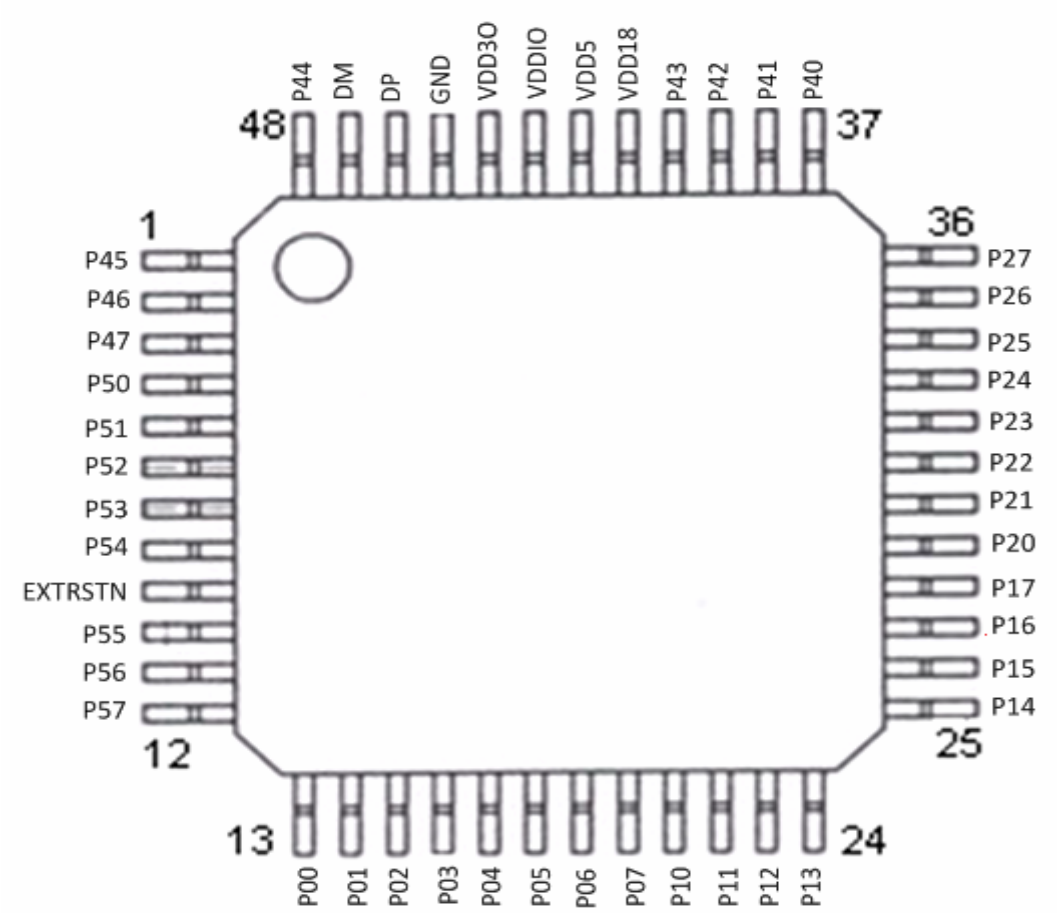
2.2 管脚说明

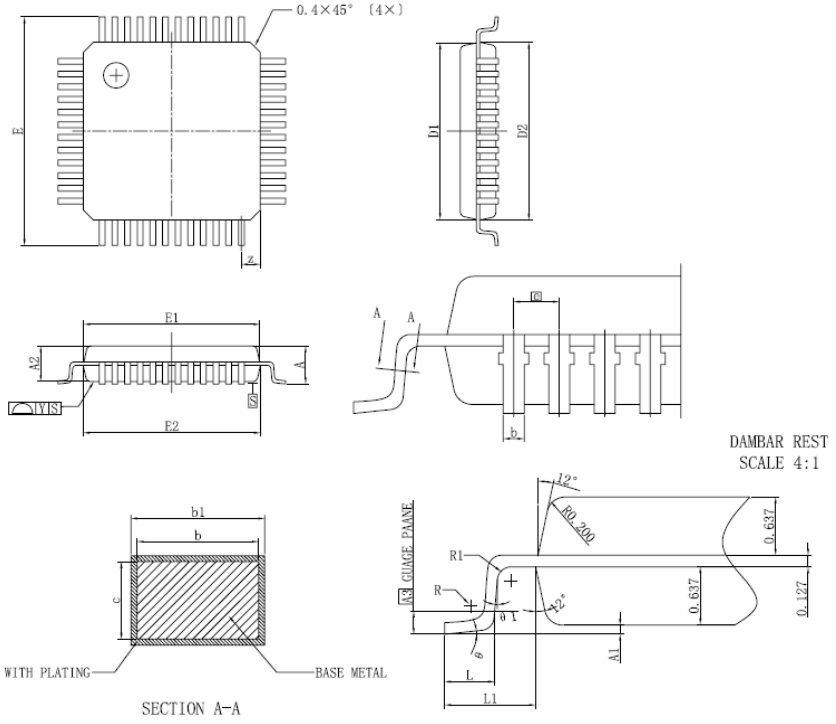
编号	管脚名	方向	说明
1	P65	IO	GPIO
2	P66	IO	GPIO/外接晶振 XC0
3	P67	IO	GPIO/外接晶振 XC1
4	FTEN	TEST	测试引脚
5	P00	IO	GPIO /Timer0 计数器时钟输入
6	P01	IO	GPIO /外部中断 INT0
7	P02	IO	GPIO
8	P03	IO	GPIO
9	P04	IO	GPIO
10	P05	IO	GPIO
11	P06	IO	GPIO/电压比较器 0 结果输出引脚
12	P07	IO	GPIO /电压比较器 1 结果输出引脚
13	P10	IO	GPIO /ADC 通道 0 模拟电压输入
14	P11	IO	GPIO /ADC 通道 1 模拟电压输入
15	P12	IO	GPIO /ADC 通道 2 模拟电压输入
16	GND	GND	系统地线
17	P13	IO	GPIO /ADC 通道 3 模拟电压输入
18	VDDIO	POWER	工作电压输入（根据应用与 VDD5 或 VDD3O 短接）
19	VPP	TEST	测试引脚
20	P14	IO	GPIO /ADC 通道 4 模拟电压输入
21	P15	IO	GPIO /ADC 通道 5 模拟电压输入
22	P16	IO	GPIO /ADC 通道 6 模拟电压输入
23	P17	IO	GPIO /ADC 通道 7 模拟电压输入
24	P20	IO	GPIO /ADC 通道 8 模拟电压输入
25	P21	IO	GPIO /ADC 通道 9 模拟电压输入
26	P22	IO	GPIO /ADC 通道 10 模拟电压输入
27	P23	IO	GPIO /ADC 通道 11 模拟电压输入

28	**P24**	**IO**	GPIO /ADC 通道 12 模拟电压输入
29	**P25**	**IO**	GPIO /ADC 通道 13 模拟电压输入
30	**P26**	**IO**	GPIO /ADC 通道 14 模拟电压输入 /电压比较器 0 测量电压输入
31	**P27**	**IO**	GPIO /ADC 通道 15 模拟电压输入 /电压比较器 0 外置基准电压输入
32	**P30**	**IO**	GPIO
33	**P31**	**IO**	GPIO
34	**P32**	**IO**	GPIO
35	**P33**	**IO**	GPIO
36	**P34**	**IO**	GPIO
37	**P35**	**IO**	GPIO
38	**P36**	**IO**	GPIO
39	**P37**	**IO**	GPIO
40	**P40**	**IO**	GPIO /DAC 基准电压输入
41	**P41**	**IO**	GPIO /电压比较器 1 测量电压输入/运算放大器电压输入
42	**P42**	**IO**	GPIO/Timer1 计数器时钟输入/电压比较器 1 基准电压输入
43	**P43**	**IO**	GPIO /外部中断 INT1
44	**VDD18**	**POWER**	1.8V LDO 电压输出
45	**VDD5**	**POWER**	供电电源电压输入
46	**VDDIO**	**POWER**	IO 端口工作电压（根据应用与 VDD5 或 VDD3O 短接）
47	**VDD3O**	**POWER**	3.3V LDO 电压输出
48	**AGND**	**GND**	模拟地线
49	**GND**	**GND**	地线
50	**DP**	**IO**	P70/USB DP/ PS2_CLK /C2 接口 1 时钟输入
51	**DM**	**IO**	P71/USB DM/PS2_DATA/C2 接口 1 数据输入输出
52	**P44**	**IO**	GPIO/I2C SCL
53	**P45**	**IO**	GPIO/I2C SDA
54	**P46**	**IO**	GPIO/UART TX 输出

55	**P47**	**IO**	GPIO/UART RX 输入
56	**P50**	**IO**	GPIO/PWM0H 输出
57	**P51**	**IO**	GPIO/PWM0L 输出
58	**P52**	**IO**	GPIO/PWM1H 输出
59	**P53**	**IO**	GPIO/PWM1L 输出
60	**P54**	**IO**	GPIO/C2 接口 0 数据输入输出引脚 2
61	**EXTRSTN**	**INPUT**	外部复位信号输入/C2 接口 0 时钟输入
62	**P55**	**IO**	GPIO/PWM2L 输出/C2 接口 0 数据输入输出引脚 1
63	**P56**	**IO**	GPIO/PWM2H 输出
64	**P57**	**IO**	GPIO
65	**P60**	**IO**	GPIO
66	**P61**	**IO**	GPIO
67	**P62**	**IO**	GPIO
68	**P63**	**IO**	GPIO
69	**P64**	**IO**	GPIO

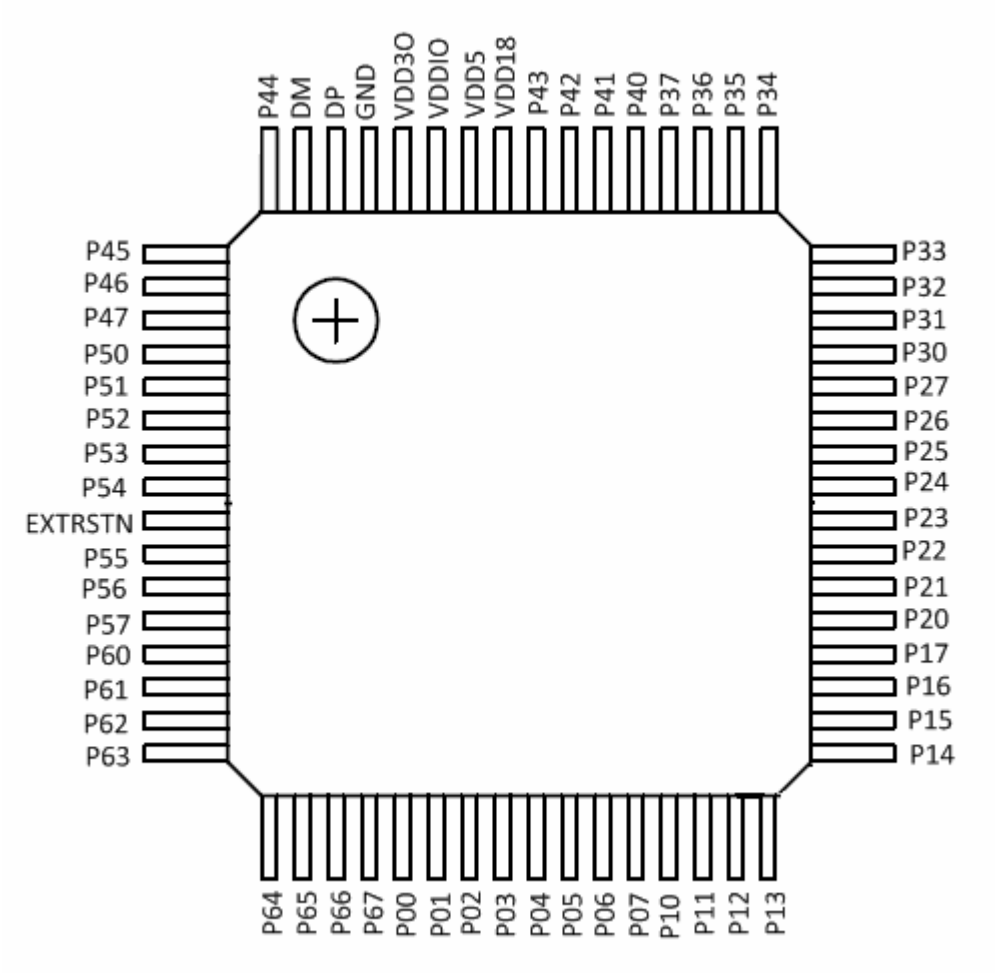
2.3 LQFP48L 封装 Pin 图及尺寸

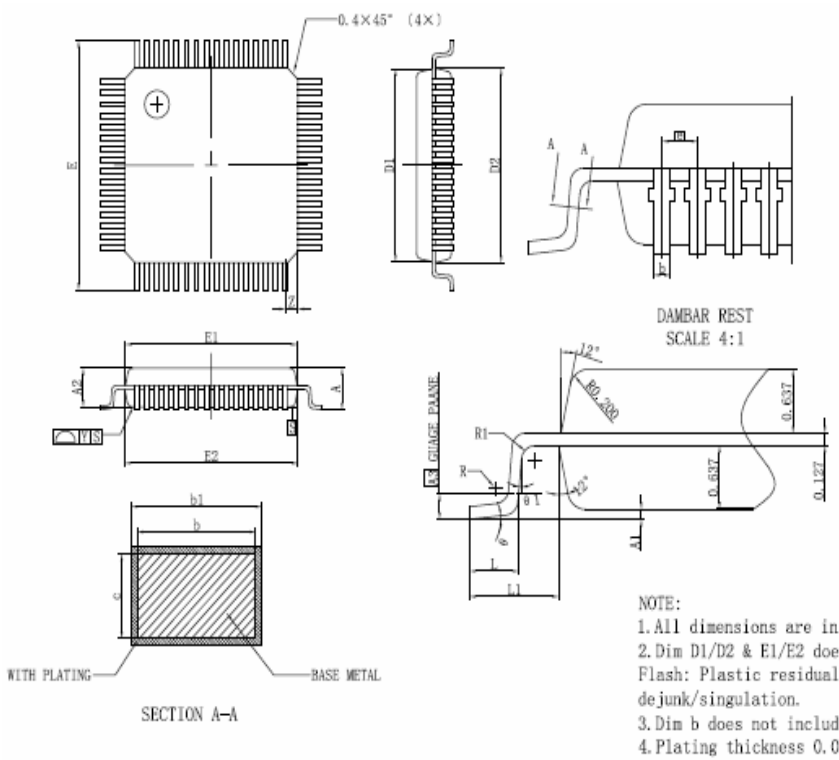




symbol	Min	Nom	Max
A	1.45	1.55	1.65
A1	0.01	---	0.21
A2	1.3	1.4	1.5
A3	---	0.254	
b	0.15	0.20	0.25
b1	0.16	0.22	0.28
c	---	0.127	---
D1	6.85	6.95	7.05
D2	6.9	7.00	7.10
E	8.8	9.00	9.20
E1	6.85	6.95	7.05
E2	6.9	7.00	7.10
e	---	0.5	---
L	0.43	---	0.71
L1	0.90	1.0	1.10
R	0.1	---	0.25
R1	0.1	---	---
θ	0	---	10°
θ1	0	---	---
y	---	---	0.1
Z	---	0.75	---

2.4 LQFP64L 封装 Pin 图及尺寸





symbol	Min	Nom	Max
A	1.45	1.55	1.65
A1	0.01	—	0.21
A2	1.3	1.4	1.5
[A3]	—	0.254	
b	0.13	0.18	0.23
b1	0.14	0.20	0.26
c	—	0.127	—
D1	6.85	6.95	7.05
D2	6.9	7.00	7.10
E	8.8	9.00	9.20
E1	6.85	6.95	7.05
E2	6.9	7.00	7.10
[e]	—	0.4	—
L	0.43	—	0.71
L1	0.90	1.0	1.10
R	0.1	—	0.25
R1	0.1	—	—
θ	0	—	10°
θ1	0	—	—
y	—	—	0.1
Z	—	0.5	—

NOTE:
1. All dimensions are in mm.
2. Dim D1/D2 & E1/E2 does not include plastic flash.
Flash: Plastic residual around body edge after dejunk/singulation.
3. Dim b does not include dambar protrusion/intrusion.
4. Plating thickness 0.005~0.015mm.

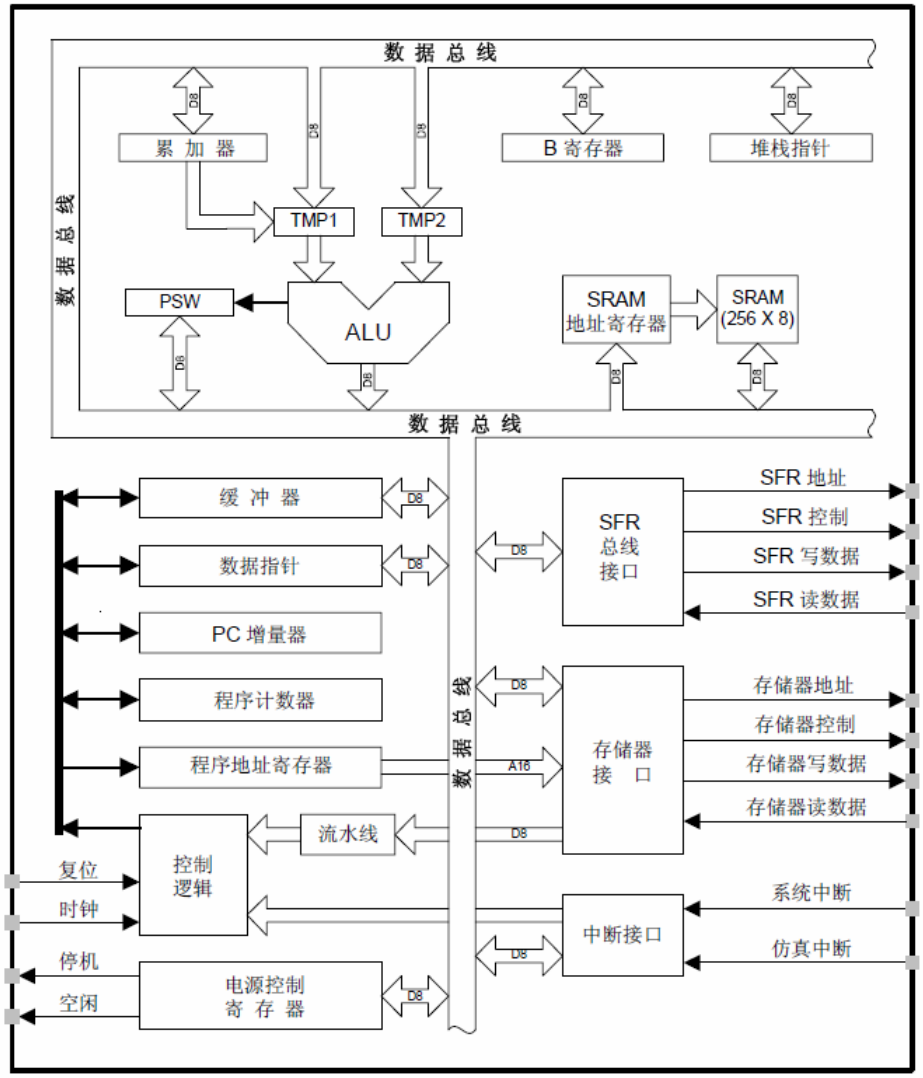
3 功能描述

3.1 MCU 内核

3.1.1 概述

SG8F7581 系统控制器内核采用 SG51 架构。SG51 与 MCS-51™ 指令集完全兼容，可以使用标准 803x/805x 的汇编器和编译器进行软件开发。

3.1.2 结构图



3.1.3 指令集

助记符	功能说明	字节数	指令时钟周期数
算术操作类指令			
ADD A,Rn	寄存器加到累加器	1	1
ADD A,direct	直接寻址字节加到累加器	2	2
ADD A,@Ri	间址RAM内容加到累加器	1	2
ADD A,#data	立即数加到累加器	2	2
ADDC A,Rn	寄存器加到累加器(带进位)	1	1
ADDC A,direct	直接寻址字节加到累加器(带进位)	2	2
ADDC A,@Ri	间址RAM加到累加器(带进位)	1	2
ADDC A,#data	立即数加到累加器(带进位)	2	2
SUBB A,Rn	累加器减去寄存器(带借位)	1	1
SUBB A,direct	累加器减去直接寻址字节(带借位)	2	2
SUBB A,@Ri	累加器减去间址RAM(带借位)	1	2
SUBB A,#data	累加器减去立即数(带借位)	2	2
INC A	累加器加1	1	1
INC Rn	寄存器加1	1	2
INC direct	直接寻址字节加1	2	3
INC @Ri	间址RAM加1	1	3
DEC A	累加器减1	1	1
DEC Rn	寄存器减1	1	2
DEC direct	直接寻址字节减1	2	3
DEC @Ri	间址RAM减1	1	3
INC DPTR	数据地址加1	1	1
MUL AB	累加器与寄存器B相乘	1	5
DIV AB	累加器除以寄存器B	1	5
DA A	累加器十进制调整	1	1
逻辑操作类指令			

ANL A,Rn	寄存器“与”到累加器	1	1
ANL A,direct	直接寻址字节“与”到累加器	2	2
ANL A,@Ri	间址RAM“与”到累加器	1	2
ANL A,#data	立即数“与”到累加器	2	2
ANL direct,A	累加器“与”到直接寻址字节	2	3
ANL direct,#data	立即数“与”到直接寻址字节	3	4
ORL A,Rn	寄存器“或”到累加器	1	1
ORL A,direct	直接寻址字节“或”到累加器	2	2
ORL A,@Ri	间址RAM“或”到累加器	1	2
ORL A,#data	立即数“或”到累加器	2	2
ORL direct,A	累加器“或”到直接寻址字节	2	3
ORL direct,#data	立即数“或”到直接寻址字节	3	4
XRL A,Rn	寄存器“异或”到累加器	1	1
XRL A,direct	直接寻址字节“异或”到累加器	2	2
XRL A,@Ri	间址RAM“异或”到累加器	1	2
XRL A,#data	立即数“异或”到累加器	2	2
XRL direct,A	累加器“异或”到直接寻址字节	2	3
XRL direct,#data	立即数“异或”到直接寻址字节	3	4
CLR A	累加器清零	1	1
CPL A	累加器求反	1	1
RL A	累加器循环左移	1	1
RLC A	带进位的累加器循环左移	1	1
RR A	累加器循环右移	1	1
RRC A	带进位的累加器循环右移	1	1
SWAP A	累加器内高低半字节交换	1	1
数据传输类指令			
MOV A,Rn	寄存器传送到累加器	1	1
MOV A,direct	直接寻址字节传送到累加器	2	2
MOV A,@Ri	间址RAM传送到累加器	1	2

MOV A,#data	立即数传送到累加器	2	2
MOV Rn,A	累加器传送到寄存器	1	2
MOV Rn,direct	直接寻址字节传送到寄存器	2	4
MOV Rn,#data	立即数传送到寄存器	2	2
MOV direct,A	累加器传送到直接寻址字节	2	3
MOV direct,Rn	寄存器传送到直接寻址字节	2	3
MOV direct1,direct2	直接寻址字节传送到直接寻址字节	3	4
MOV direct,@Ri	间址RAM传送到直接寻址字节	2	4
MOV direct,#data	立即数传送到直接寻址字节	3	3
MOV @Ri,A	累加器传送到间址RAM	1	3
MOV @Ri,direct	直接寻址字节传送到间址RAM	2	5
MOV @Ri,#data	立即数传送到间址RAM	2	3
MOV DPTR,#data16	16位常数装入DPTR	3	3
MOVC A,@A+DPTR	相对于DPTR的代码字节传送到累加器	1	3
MOVC A,@A+PC	相对于PC的代码字节传送到累加器	1	3
MOVX A,@Ri	外部RAM(8位地址)传送到累加器	1	3
MOVX A,@DPTR	外部RAM(16位地址)传送到累加器	1	3
MOVX @Ri,A	累加器传到外部RAM（8位地址）	1	4
MOVX @DPTR,A	累加器传到外部RAM（16位地址）	1	4
PUSH direct	直接寻址字节压入栈顶	2	4
POP direct	栈顶数据弹出到直接寻址字节	2	3
XCH A,Rn	寄存器和累加器交换	1	2
XCH A,direct	直接寻址字节与累加器交换	2	3
XCH A,@Ri	间址RAM与累加器交换	1	3
XCHD A,@Ri	间址RAM和累加器交换低半字节	1	3
位操作指令			
CLR C	清进位位	1	1
CLR bit	清直接寻址位	2	3
SETB C	进位位置1	1	1

SETB bit	直接寻址位置位	2	3
CPL C	进位位取反	1	1
CPL bit	直接寻址位取反	2	3
ANL C,bit	直接寻址位“与”到进位位	2	2
ANL C,/bit	直接寻址位的反码“与”到进位位	2	2
ORL C,bit	直接寻址位“或”到进位位	2	2
ORL C,/bit	直接寻址位的反码“或”到进位位	2	2
MOV C,bit	直接寻址位传送到进位位	2	2
MOV bit,C	进位位传送到直接寻址位	2	3
跳转类指令			
JC rel	若进位位为1则跳转	2	3
JNC rel	若进位位为零则跳转	2	3
JB bit,rel	若直接寻址位为1则跳转	3	4
JNB bit,rel	若直接寻址位为零则跳转	3	4
JBC bit,rel	若直接寻址位为1则跳转，并清除该位	3	4
ACALL addr11	绝对调用子程序	2	6
LCALL addr16	长调用子程序	3	6
RET	从子程序返回	1	4
RETI	从中断返回	1	4
AJMP addr11	绝对转移	2	3
LJMP addr16	长转移	3	4
SJMP rel	短转移（相对地址）	2	3
JMP @A+DPTR	相对DPTR的间接转移	1	2
JZ rel	累加器为0则转移	2	3
JNZ rel	累加器为非0则转移	2	3
CJNE A,direct,rel	比较直接寻址字节与累加器，不相等则转移	3	4
CJNE A,#data,rel	比较立即数与累加器，不相等则转移	3	4
CJNE Rn,#data,rel	比较立即数与寄存器，不相等则转移	3	4
CJNE @Ri,#data,rel	比较立即数与间接寻址RAM，不相等则转移	3	4

DJNZ Rn,rel	寄存器减1，不为零则转移	2	3
DJNZ direct,rel	直接寻址字节减1，不为零则转移	3	4
NOP	空操作	1	1

寄存器、操作数和寻址方式说明：

Rn – 当前选择的寄存器区的寄存器R0-R7。

@Ri – 通过寄存器R0-R1间接寻址的数据RAM地址。

rel – 相对于下一条指令第一个字节的8位有符号（2的补码）偏移量。SJMP和所有条件跳转指令使用。

direct – 8位内部数据存储器地址。可以是直接访问数据RAM地址（0x00-0x7F）或一个SFR地址（0x80-0xFF）。

#data – 8位立即数

#data16 – 16位立即数

bit – 数据RAM或SFR中的直接寻址位

addr11 – ACALL或AJMP使用的11位目的地址。目的地址必须与下一条指令第一个字节处于同一个2K字节的程序存储器页。

addr16 – LCALL或LJMP使用的16位目的地址。目的地址可以是8K程序存储器空间内的任何位置。

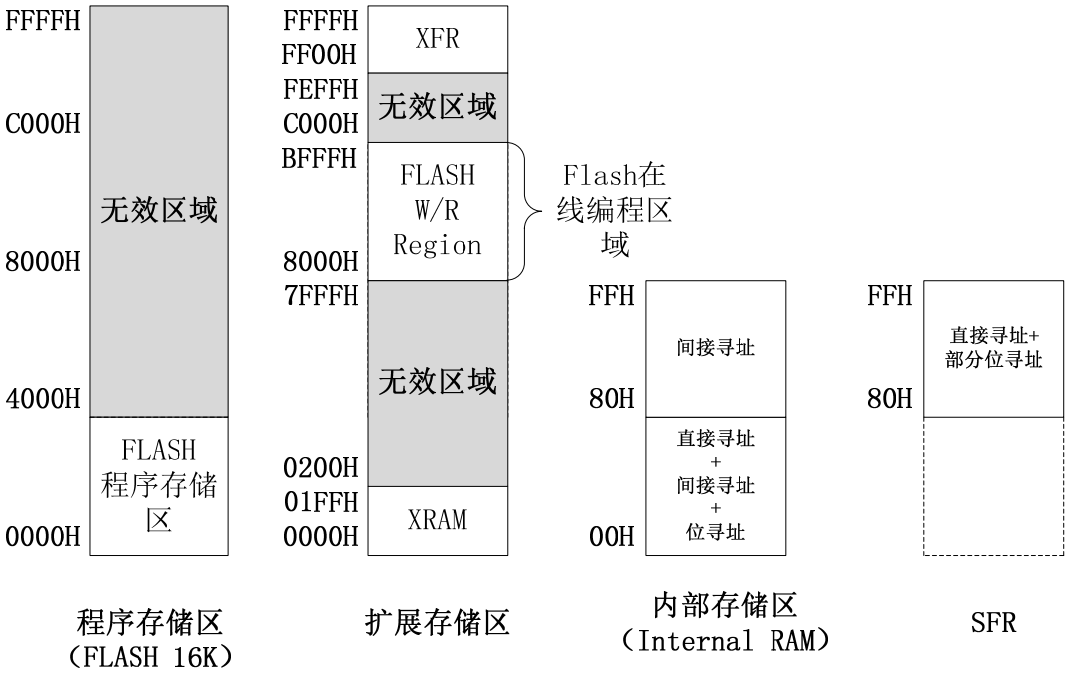
有一个未使用的操作码（0xA5），它执行与NOP指令相同的功能。

3.1.4 存储器

3.1.4.1概述

SG8F7581 芯片使用统一编址的程序和数据存储器，两者通过不同的指令进行访问。下图为存储器结构图。其中程序存储器由 16K FLASH 构成，数据存储器由 256Byte 内部数据存储器和 512Byte 扩展数据存储器组成。

3.1.4.2结构图



存储器结构图

3.1.4.3存储器说明

3.1.4.3.1 程序存储器（Program ROM）

SG51 支持 64KB 的程序存储空间。SG8F7581 内部集成 16KB 可在线编程 FLASH 作为程序存储器，寻址区 0x0000~0x3FFF。0x4000 地址以上区域保留。

程序存储器默认只读，但 SG8F7581 可通过设置程序存储器写允许位（SFR:PFCTL bit0），用 MOVX 指令对 FLASH 在线编程区写入来更新 FLASH 内部数据，从而提供更新程序代码以及将程序存储器用作非易失性存储机制的功能。详细见“ FLASH 控制器”。

3.1.4.3.2 内部数据存储器（Internal RAM）

内部数据存储器中地址位于 0x00~0x7F 的部分可以通过直接或间接寻址方式进行访问。地址为 0x80~0xFF 的部分由于地址和特殊功能寄存器(SFR)相同，所以只可以使用间接寻址的方式访问，而地址同样为 0x80~0xFF 的 SFR 可以通过直接或者位寻址（部分地址）方式进行访问。图 6-2-2 为内部存储器结构及其寻址方式

内部数据存储器中地址 0x00~0x1F 的部分为通用寄存器，共分为 4 个 bank，每个 bank 都有

8x8bit的寄存器，作为R0~R7是占用。软件可通过程序状态字PSW中的RS0(PSW.3)和RS1(PSW.4)位选择当前处于活动状态的bank。这允许在进入子程序或中断服务程序时进行快速现场切换。间接寻址方式使用R0和R1作为间接寄存器。

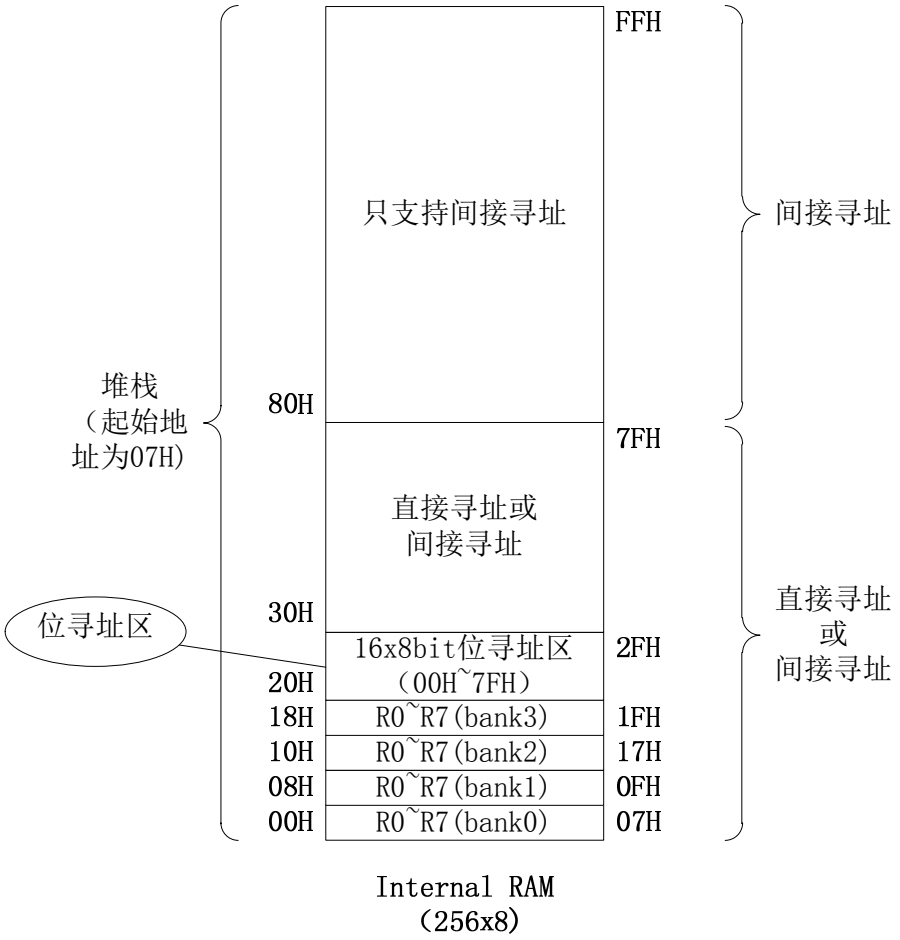


图 6-2-2 内部存储器结构及其寻址方式

3.1.4.3.2.1通用寄存器

内部数据存储器的低 32 字节，从地址 0x00 到 0x1F,可以作为 4 个 BANK 的通用寄存器访问。每个BANK内包含8个8bit寄存器，分别为R0~R7。同一时间内只能选择一个BANK区域作为R0~R7使用。PSW 中的 RS0(PSW.3)和 RS1(PSW.4)位用于选择当前的寄存器区(见 SFR 定义中关于 PSW 的说明)。这允许在进入子程序或中断服务程序时进行快速现场切换。另外间接寻址方式使用 R0 和 R1 作为间接寄存器。

3.1.4.3.2.2堆栈

程序的堆栈可以位于 256 字节数据存储器中的任何位置。堆栈区域用堆栈指针(SP,0x81)SFR 指令。SP 指向最后使用的位置。下一个压入堆栈的数据将被存放在 SP+1，然后 SP 加 1。复位后

堆栈指针被初始化为地址 0x07，因此第一个被压入堆栈的数据将被放在地址 0x08，这也是通用寄存器区 1 的第一个寄存器(R0)。如果使用不止一个寄存器区，SP 应被初始化为数据存储器中不用于数据存储的位置。堆栈深度最大可达到 256 字节。

3.1.4.3.2.3位寻址空间

除了直接访问按字节组织的数据存储器外，从 0x20 到 0x2F 的 16 个数据存储器单元还可以作为 128 个独立寻址位访问。每个位有一个位地址，从 0x00 到 0x7F。位于地址 0x20 的数据字节的为 0 具有位地址 0x00,位于 0x20 的数据字节的位 7 具有地址 0x07.位于 0x2F 的数据字节的位 7 具有位地址 0x7F。由所用指令的类型来区分是位寻址还是字节寻址。

另外，还有 128 个独立寻址位离散的分布在 SFR 寄存器中间（特点是 SFR 地址如果能被 8 整除，则支持位寻址，见下 SFR 描述），位编制地址从 0x80 到 0xFF。

3.1.4.3.3 扩展数据存储器（XRAM）

SG51 系列提供了 64KB 的扩展寻址区，可通过 MOVX 类指令间接寻址访问。

SG8F7581 片内集成了 512Byte 的扩展数据存储器（XRAM），位于扩展寻址区，寻址空间 0x0000~0x01FF。

3.1.4.3.4 特殊功能寄存器（SFR）

从 80H 到 0FFH 的直接寻址存储器空间为特殊功能寄存器(SFR)。SFR 提供对 SG8F7581 的资源和外设的控制及与这些资源和外设之间的数据交换。

任何时刻用直接寻址方式访问地址为 80H~0FFH 的存储器空间都将访问特殊功能寄存器(SFR)。地址以 0x0 或 0x8 结尾的 SFR(例如 P0、TCON、P1、SCON、IE 等)既可以按字节寻址也可以按位寻址，其它 SFR 只能按字节寻址。表 6-2-3 为特殊功能寄存器列表

表 6-2-3 特殊功能寄存器 SFR

Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111
F8	P7	DMACHEN	DMAINTF	DMAOF0	DMAOF1	DMAOF2	PFCTL	FLKEY
F0	B	OCPCN	CPCN	DAC0CN	DAC1CN	CP0MD	CP1MD	CPINT
E8	P6	OVPCN						RSTRSF
E0	ACC		PWM0DT	PWM1DT	PWM2DT	PWM0PRE	PWM1PRE	PWM2PRE
D8	P5	PWMHR	PWM0PL	PWM0PH	PWM1PL	PWM1PH	PWM2PL	PWM2PH

D0	PSW	PWMCN	PWM0DL	PWM0DH	PWM1DL	PWM1DH	PWM2DL	PWM2DH
C8	TMR2CN	TMR2L	TMR2H	TMR2RLL	TMR2RLH	TMR2RCK	USBINTEN	USBINTF
C0	P4	USBCON	EPT0CTL	EPT1CTL	EPT2CTL	USBADDR	USBSTA	CLKGATE
B8	IEN1	IEN2	IP0	IP1	CLKCN	CLKSEL	SSCGCN	WDTCTL
B0	P3	ADCCN	ADCANA	ADCCF	ADCMUX	ADCDATAH	ADCDATAL	WDTCON
A8	IEN0	I2CSTAT	I2CCON	I2CDATA	I2CSCLH	I2CSCLL	I2CADDR	-
A0	P2	PORTMUX	PORTIE	PORTIF	P4IE	P5IE	P4IF	P5IF
98	SCON0	SCON1	SBUF	SBRLL	SBRLH		P7DIR	PWRCON
90	P1	P0DIR	P1DIR	P2DIR	P3DIR	P4DIR	P5DIR	P6DIR
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	XADDRH
80	P0	SP	DPL	DPH	DPL1	DPH1	DPS	PCON
备注	(0)/(8)支持位寻址	(1) /(9)	(2) /(A)	(3) /(B)	(4) /C	(5) /(D)	(6) /(E)	(7) /(F)

3.1.4.3.5 扩展功能寄存器（XFR）

针对特殊功能寄存器 SFR 128 个地址空间相对不足，SG8F7581 设置了扩展功能寄存器 XFR，将部分外设的配置寄存器放在外部扩展地址存储区，XFR 寄存器只能通过 MOVX 类指令读写，寻址空间从 0xFF00~0xFFFF，总计 256 个地址区，内部寄存器地址分配如表 6-2-4

ADDRL [7:0]	**X000**	**X001**	**X010**	**X011**	**X100**	**X101**	**X110**	**X111**
0xF8 ~ 0x28	保留	保留	保留	保留	保留	保留	保留	保留
0x20	DMADPT0	DMADPT1	DMADPT2	DMABAH	DMABA0L	DMABA1L	DMABA2L	保留
0x18	P0INTWK	P1INTWK	P2INTWK	P3INTWK	P4INTWK	P5INTWK	P6INTWK	P7INTWK
0x10	PORTAN	P1AN	P2AN	保留	P4AN	保留	保留	保留
0x08	P0HDRV	P1HDRV	P2HDRV	P3HDRV	P4HDRV	P5HDRV	P6HDRV	P5PD
0x00	P0PU	P1PU	P2PU	P3PU	P4PU	P5PU	P6PU	P2PD

注：XFR 地址高字节 ADDRH 为 0xFF

3.1.5 内核相关寄存器

3.1.5.1地址映射

寄存器	地址	复位值	说明
ACC	0xE0	0x00	累加器
B	0xF0	0x00	B 寄存器
PSW	0xD0	0x00	程序状态字
SP	0x81	0x07	堆栈指针
DPH	0x83	0x00	数据指针 0 的高字节
DPL	0x82	0x00	数据指针 0 的低字节
DPH1	0x85	0x00	数据指针 1 的高字节
DPL1	0x84	0x00	数据指针 1 的高字节
DPS	0x86	0x00	数据指针选择寄存器
XADDRH	0x8F	0x00	扩展存储区寻址地址高字节

3.1.5.2寄存器描述

3.1.5.2.1 ACC：累加器

该寄存器作为算术操作用的累加器大部分指令使用累加器保存操作数，在指令码中简记为 A，另外，该寄存器在 SFR 中有固定的地址，可通过直接寻址的方式对其进行操作。

3.1.5.2.2 B：寄存器

寄存器 B 可用于乘除法运算

乘法指令 MUL AB ——ACC 与 B 相乘，乘积低 8 位存入 ACC，高 8 位存入 B；

触发指令 DIV AB——ACC 除以 B，商存入 ACC，余数存入 B；

3.1.5.2.3 PSW：程序状态字

位	寄存器名	读写	说明
7	**CY**	R/W	进位标志(Carry Flag) 1 = 表示在加法运算时产生进位，在减法运算时产生借位 0 = 表示无进位或借位发生
6	**AC**	R/W	辅助进位标志(Auxiliary Carry) 1 = 表示低半字节在加法运算里产生的进位，在减法运算时产生借位 0 =表示低半字节无进位或借位发生
5	**F0**	R/W	用户自定义
4-3	**RS.1~RS.0**	R/W	工作寄存器分组选择位(Register bank Select) 11 = bank3 地址 0x18~0x1F 10 = bank2 地址 0x10~0x17 01 = bank1 地址 0x08~0x0F 00 = bank0 地址 0x07~0x07 附：工作寄存器指的是 R0~R7
2	**OV**	R/W	有符号数（-128~127）运算溢出标志(Overflow flag) 1：运算结果溢出 0：运算结果未溢出
1	**F1**	R/W	用户自定义
0	**P**	R	ACC 的奇偶校验位(Parity Flag)

3.1.5.2.4 SP：堆栈指针

位	寄存器名	读写	说明
7-0	**SP**	R/W	堆栈指针用于记录栈顶位置，每次执行压栈操作(如 PUSH)时，指针自动加 1，执行出栈操作(如 POP)时，指针自动减 1； 复位后默认值为 0x07。

3.1.5.2.5 DPTR & DPTR1：数据指针

数据指针 DPTR 位宽为 16bit，用于扩展寻址区存储空间的访问。

SG51 核内有两个数据指针 DPTR/DPTR1：

- **DPTR：**

位	寄存器名	读写	说明
7~0	**DPH**	R/W	DPTR 高字节

位	寄存器名	读写	说明
7~0	**DPL**	R/W	DPTR 低字节

- **DPTR1:**

位	寄存器名	读写	说明
7~0	**DPH1**	R/W	DPTR1 高字节

位	寄存器名	读写	说明
7~0	**DPL1**	R/W	DPTR1 低字节

SG51 工作时，DPTR/DPTR1 同一时间只有一个数据指针处于活动状态（参与 DPTR 相关指令执行），软件可通过配置数据指针选择寄存器 DPSEL（SFR:DPS bit1）选择当前处于活动状态的数据指针。

3.1.5.2.6 DPS：数据指针选择

位	寄存器名	读写	说明
7-1	-		保留
0	DPSEL		数据指针选择寄存器 0：选择 DPTR 作为数据指针 1：选择 DPTR1 作为数据指针

3.1.5.2.7 XADDRH：XRAM 寻址地址高字节

位	寄存器名	读写	说明
7-0	**XADDRH**	R/W	用 MOVX A,@Ri 或 MOVX @Ri,A 指令读写 XRAM 时，由 Ri 间接寻址定位 XRAM 16bit 寻址地址低字节；而此时 XADDRH 用于定位 XRAM 16bit 寻址地址高字节；

3.2 时钟系统

3.2.1 概述

SG8F7581 内部集成 12MHz 的高频 RC 振荡器、500KHz 的低频 RC 振荡器以及 4 倍频 PLL(基于 12MHz 振荡器时钟工作，倍频输出频率 48MHz)。另外芯片可外接低频晶振作为芯片时钟源。

芯片内置 USB 时钟自动恢复电路，可根据 USB 输入信号自动校准时钟。

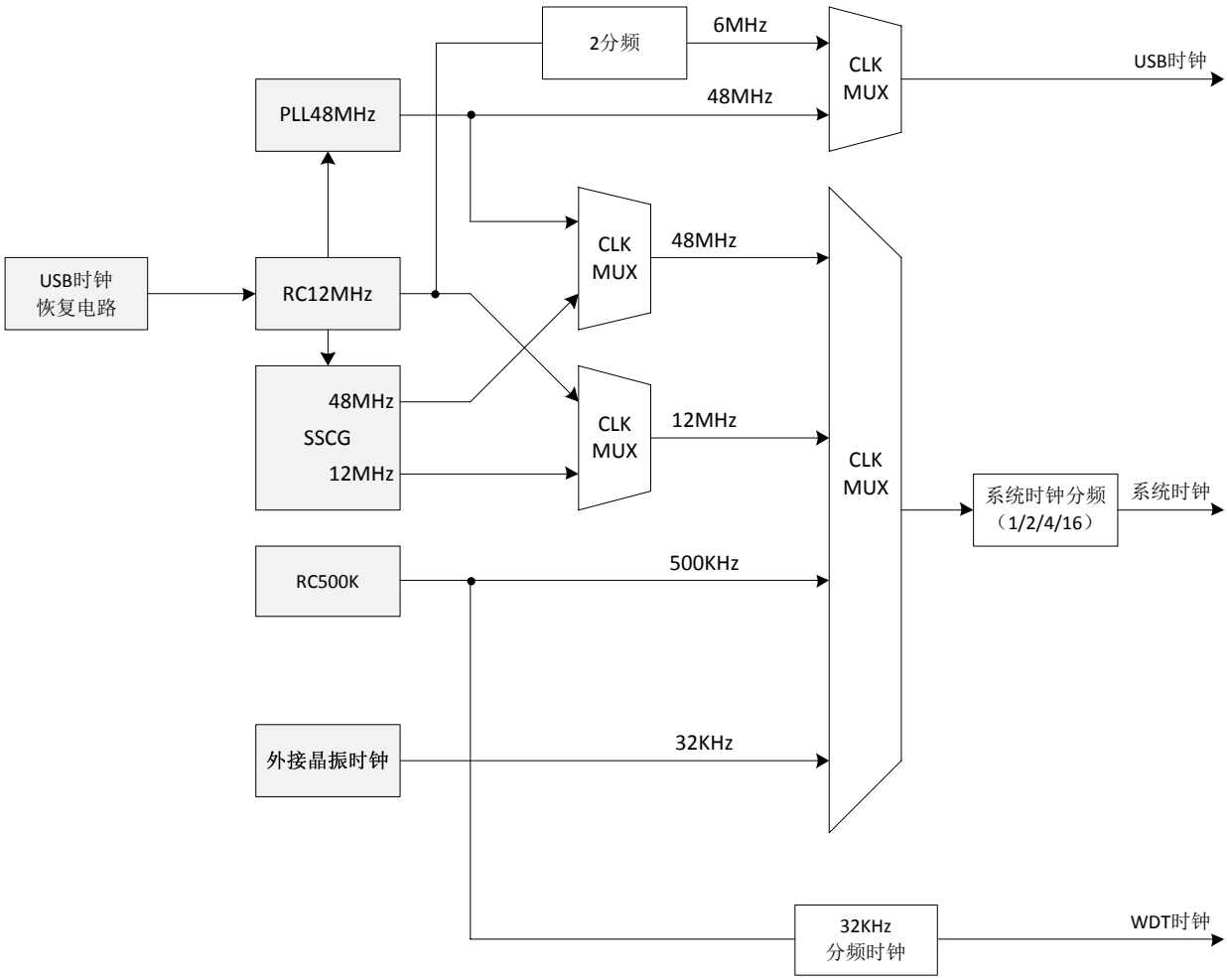
芯片为系统高频工作时钟（12MHz/48MHz）输出内置时钟扩频电路（SSCG）；同时芯片为所有外设（timer0/1/2，WDT，ADC，I2C，UART，GPIO 等）提供的系统时钟源独立可控，软件可选择性关断未使用外设的系统时钟源（注：关断外设系统时钟后，该外设相关寄存器都将处于不可读写状态），提高安规特性。

3.2.2 特征

- 高频 RC 振荡器(输出频率 12MHz)
- 4 倍时钟锁相环 PLL（输出频率 48MHz）
- 低频 RC 振荡器(输出频率 500KHz)
- 可外接低频 32KHz 晶体振荡器
- SSCG——系统时钟扩频输出（对高频时钟 12MHz 或 48MHz 倍频时钟进行扩频）
- 丰富的系统时钟源可选择
- 系统时钟可在程序运行期间自由选择切换
- USB 时钟自动校准电路
- 外设时钟门控功能

3.2.3 功能描述

3.2.3.1时钟系统结构图



3.2.3.2高频 RC 振荡器

SG8F7581 内置 12MHz 高频 RC 振荡器，用于给系统工作提供主时钟，并向 USB 控制器/ADC 等外设提供工作时钟；芯片出厂时会对高频 RC 进行出厂校准。

3.2.3.3锁相环（PLL）

SG8F7581 内置 4 倍时钟倍频电路，可将内部 12MHz RC 振荡器输出倍频至 48MHz。倍频时钟输出可选择作为系统时钟源，同时可作为全速 USB 的工作时钟。

3.2.3.4低频 RC 振荡器

芯片内置 500KHz 低频 RC 振荡器，低频时钟输出可作为系统时钟。另外该时钟经内部分频后输出 32KHz 时钟，可为 WDT 计时时钟或 PWM 计数器工作时钟。低频 RC 可由软件控制开关。

3.2.3.5晶体振荡器时钟输入

SG8F7581 提供 P66/P67 作为外接低频晶体振荡器脚位。当 OSCEN=1（CLKCN bit4）时，P66/P67 引脚自动关闭 GPIO 功能并复用为晶振时钟引脚。

注：外部晶振功能开启后，发生系统复位或系统进入 STOP 模式时，外部晶振不会强制关闭（上电复位或低压复位除外）。

3.2.3.6时钟扩频器（SSCG）

SG8F7581 内置时钟扩频电路，可对 12MHz 高频振荡器时钟进行扩频输出 48MHz/12MHz 两个扩频后时钟。时钟扩频器开启时，芯片将自动将高频时钟由 12MHz RC 振荡器和 48MHz PLL 切换至扩频器输出（USB 时钟除外）。

开启时钟扩频器可有效降低高频工作时芯片的 EMI 特性，但会少量增加芯片功耗。

3.2.3.7系统主时钟选择

系统时钟源选择——通过 CLKSEL 寄存器，软件可选择以下时钟作为系统时钟源：

- ✧ 12MHz 的高频振荡器输出时钟
- ✧ 48MHz PLL 输出时钟
- ✧ 低频 500KHz 振荡器输出时钟
- ✧ 外接低频晶振输出时钟。

系统时钟源切换——时钟切换时，目标系统时钟源如果处于关闭状态，将会被强制开启；被切换的系统时钟源如果被软件提前关闭且未被其它外设强制开启，则会自动关闭。

系统时钟分频——软件可选择对选定的系统时钟源进行 2/4/16 分频后作为系统时钟。

正常工作期间，系统时钟可在上述时钟源及系统时钟分频之间自由切换。

3.2.3.8 USB 时钟自动调节

SG8F7581 内置 USB 时钟恢复电路，由 USBTRIM（USBCON bit1）寄存器控制开启。该功能开启后，如果软件关闭高频 RC 振荡器的软件校准功能，该电路可根据 USB 主机输入波形自动调节 RC12M 振荡器频率，以保证 USB 通讯正常。

3.2.3.9 外设时钟门控功能

SG8F7581 对芯片内部主要外设（timer0/1/2、I2C、UART、UDC、PWM、GPIO 等）时钟配置了软件可控的门控单元（CLKGATE）。关闭这些门控单元时，外设相关的时钟网络将被强制关闭，用来减少外设时钟翻转造成的 EMI 问题；但门控时钟关闭时，这些外设将无法正常工作，且无法修改外设相关 SFR/XFR 寄存器的值（处于只读状态），因此要启用某外设时，必须先开启其对应的时钟门控单元。

3.2.4 相关寄存器

3.2.4.1地址映射

寄存器	地址	初始值	说明
CLKCN	0xBC	0xCC	时钟控制寄存器
CLKSEL	0xBD	0x8C	时钟选择寄存器
SSCGCN	0xBE	0x00	时钟扩频控制寄存器
CLKGATE	0xC8	0xFF	外设时钟门控寄存器

3.2.4.2寄存器描述

3.2.4.2.1 CLKCN：时钟控制寄存器

位	名称	读写	说明
7	HFEN	R/W	12MHz 高频 RC 使能位： 1 = 使能 12MHz RC 振荡器。

			0 = 禁止 12MHz RC 振荡器。
6	LFEN	R/W	低频振荡器使能位 1 = 低振荡器开启 0 = 低频振荡器关闭
5	PLLEN	R/W	PLL 使能控制位 1 = PLL 开启 0 = PLL 关闭 注：PLL 开启时，将强制开启 12MHz RC 振荡器。
4	OSCEN	R/W	外接晶振使能控制位 1 = 开启外接晶振时钟； 0 = 关闭外接晶振时钟；
3	HFRDY	R	12MHz 高频 RC 稳定标志位： 1 = 高频 RC 振荡器输出稳定。 0 = 高频 RC 振荡器无输出或输出未稳定。
2	LFRDY	R	低频时钟输出稳定标志位 1 = 低频振荡器输出稳定 0 = 低频振荡器无输出或输出未稳定
1	PLLRDY	R	PLL 时钟输出稳定标志位 1 = PLL 48MHz 时钟输出已稳定 0 = PLL48MHz 时钟无输出或输出未稳定。
0	OSCRDY	R	OSC 时钟输出稳定标志位 1 = OSC 时钟输出已稳定 0 = OSC 时钟无输出或输出未稳定。

3.2.4.2.2 CLKSEL：时钟选择寄存器

位	名称	读写	说明
7	HFSF	R	高频时钟源选择标志位： 1 = 当前系统时钟源是 12MHz 时钟； 0 = 当前系统时钟源不是 12MHz 时钟；

6	LFSF	R	1 = 当前系统时钟源是 500KHz 低频时钟； 0 = 当前系统时钟源不是 500KHz 低频时钟；
5	PLLSF	R	1 = 当前系统时钟源为 48MHz 时钟； 0 = 当前系统时钟源不是 48MHz 时钟；
4	OSCSF	R	1 = 当前系统时钟源为外部晶振时钟； 0 = 当前系统时钟源不是外部晶振时钟；
3~2	SCDIV	R/W	系统时钟分频选择位： 00 = SCSEL 选择的时钟源 16 分频后作为系统时钟； 01 = SCSEL 选择的时钟源 4 分频后作为系统时钟； 10 = SCSEL 选择的时钟源 2 分频后作为系统时钟； 11= SCSEL 选择的时钟源不分频直接作为系统时钟；(默认)
1~0	SCSEL	R/W	系统时钟源选择位： 00 = 选择 12MHz 时钟作为系统时钟源；(默认) 01 = 选择 48MHz 时钟作为系统时钟源； 10 = 选择 500KHz 低频时钟作为系统时钟源； 11 = 选择外部晶振时钟作为系统时钟源（需要提前开启外部晶振使能）；

3.2.4.2.3 SSCGCN: 时钟扩频控制寄存器

位	名称	读写	说明
7	SSEN	R/W	时钟扩频控制位： 1 = 时钟扩频功能开启，由时钟扩频输出提供 12MHz/48MHz 时钟源。 0 = 时钟扩频功能关闭，由高频 RC 和 PLL 提供 12MHz/48MHz 时钟源。
6	SSRDY	R	时钟扩频输出稳定标志位： 1 = 扩频时钟（12MHz/48MHz）输出已稳定； 0 = 扩频时钟无输出或输出未稳定
5~0	保留	R	-

3.2.4.2.4 CLKGATE：外设工作时钟控制寄存器

注：外设工作时钟关闭状态下，芯片内部该外设相关的时钟网络被关闭，可有效减小芯片 EMI；但该状态下外设将无法工作，且所有该外设相关的 sfr/xfr 处于只读状态。

位	名称	读写	说明
7	T01GATE	R/W	timer0/timer1 工作时钟使能位： 1 = 开启 timer0/timer1 工作时钟； 0 = 关闭 timer0/timer1 工作时钟。
6	T2GATE	R/W	timer2 工作时钟使能位： 1 = 开启 timer2 工作时钟； 0 = 关闭 timer2 工作时钟。
5	I2CGATE	R/W	I2C 控制器工作时钟使能位： 1 = 开启 I2C 控制器工作时钟； 0 = 关闭 I2C 控制器工作时钟。
4	UARTGATE	R/W	UART 控制器工作时钟使能位： 1 = 开启 UART 控制器工作时钟； 0 = 关闭 UART 控制器工作时钟。
3	UDCGATE	R/W	UDC 工作时钟使能位： 1 = 开启 UDC 控制器工作时钟； 0 = 关闭 UDC 控制器工作时钟。
2	PWMGATE	R/W	PWM0/1/2 工作时钟使能位： 1 = 开启 PWM0/1/2 工作时钟； 0 = 关闭 PWM0/1/2 工作时钟。
1	ADCGATE	R/W	ADC 工作时钟使能位： 1 = 开启 ADC 工作时钟； 0 = 关闭 ADC 工作时钟。
0	GPIOGATE	R/W	GPIO 控制器工作时钟使能位： 1 = 开启 GPIO 控制器工作时钟； 0 = 关闭 GPIO 控制器工作时钟。

3.3 系统模式

3.3.1 概述

SG8F7581 共有 2 种低功耗模式：空闲模式(IDLE)和停机模式(STOP)。

3.3.2 特征

- **空闲模式**

  空闲模式下，MCU 内核停止工作，外设和时钟系统继续处于活动状态。

  MCU 进入空闲模式后，可被任意中断唤醒（唤醒后系统进入中断处理子程序），也可被任何复位唤醒（会执行系统复位）

- **停机模式**

  停机模式下，内部振荡器、MCU 内核和外设模块都进入停止状态；因此停机模式下芯片功耗最低。另外，进入停机模式前，软件可控制关闭大 LDO(见电源系统说明)，进一步减小芯片功耗。

  MCU 进入停机模式后，可被端口唤醒（端口唤醒时，如果端口中断开启，MCU 进入相应端口中断的中断处理子程序）、WDT 溢出等唤醒源唤醒；可被端口复位或低压复位唤醒（执行系统复位），但不可被基于系统时钟工作的中断源（如 Timer0/1/2 中断）唤醒。MCU 退出停机模式时，如果唤醒源相关的中断也处于使能状态，MCU 会立刻跳入相应中断处理程序处理该中断。

3.3.3 唤醒源

- 端口唤醒：

  P0/P1/P2/P3/P4 端口状态变化唤醒，IDLE/STOP 模式下均可唤醒；

- WDT 唤醒：

  WDT 计数溢出唤醒，IDLE/STOP 模式下均可唤醒；

- 中断唤醒：

  各个中断源产生中断唤醒，IDLE 模式下可唤醒，STOP 模式下只有非基于系统时钟工作的中断源（如端口中断/WDT 中断）才能唤醒，而基于系统时钟而产生的中断（如 timer0/1/2 溢出中断）无法将 MCU 从 STOP 模式唤醒；

- 复位唤醒：

外部复位/低压复位/WDT 溢出复位唤醒，唤醒时执行系统复位，IDLE/STOP 模式下均可唤醒；

3.3.4 相关寄存器

3.3.4.1地址映射

名称	地址	读写	初始值	描述
PCON	0x87	R/W	0x00	系统模式控制寄存器

3.3.4.2寄存器描述

3.3.4.2.1 PCON：系统模式控制寄存器

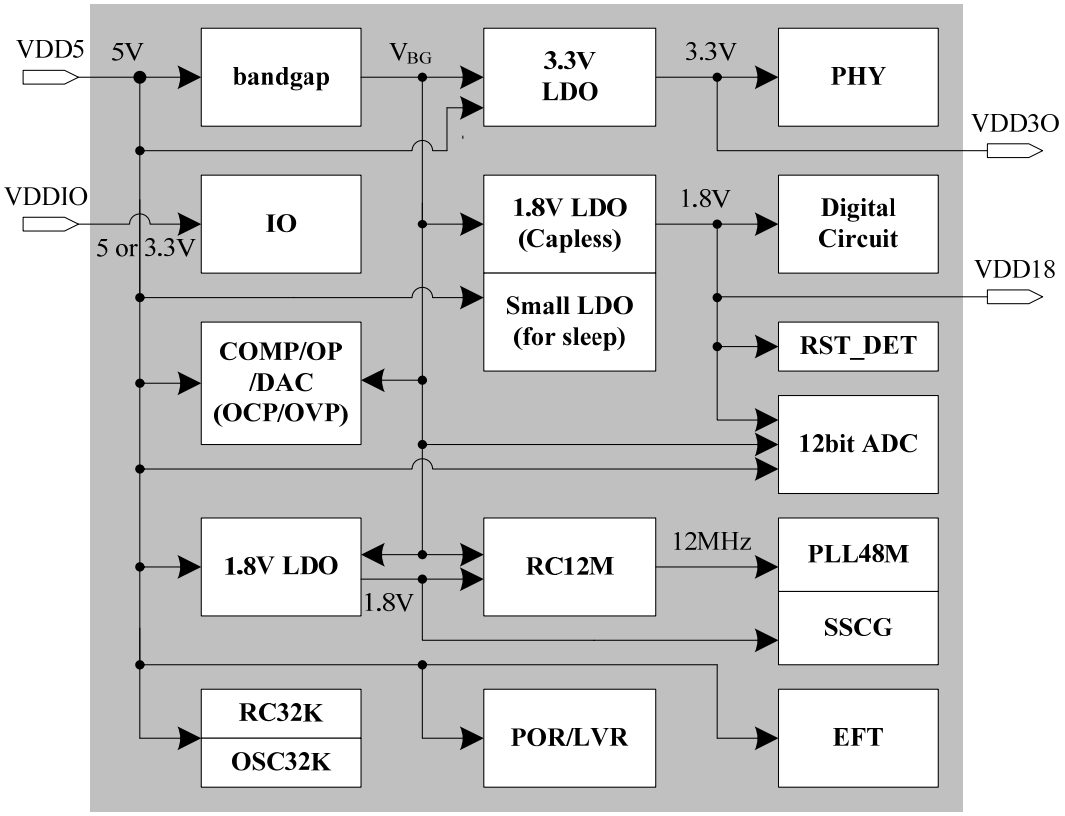
位	名称	读写	描述
7	保留		
1	STOP	R/W	1 = MCU 进入停机方式 0 = 没有影响
0	IDLE	R/W	1 = MCU 进入空闲方式 0 = 没有影响

3.4 电源系统

3.4.1 概述

- 5V 电源输入，电压工作范围 2.6V~5.5V
- 内置电压基准
- USB 物理接口（PHY）采用 3.3V LDO 供电，3.3V LDO 可给片外其它设备供电（负载电流小于 50mA）
- 数字内核采用 1.8V LDO 供电
- 高频时钟产生电路采用 1.8V 供电
- 低频时钟产生电路采用 5V 供电
- IO 供电为 5V 或 3.3V

3.4.2 结构图



注：

①当电源电压为 5V（偏差范围为 4.5V~5.5V），IO 对外接口为 5V 时，VDDIO 和 VDD5 连接在一起；对外接口为 3.3V 时，VDDIO 和 VDD3O 连接在一起。

②当电源电压为 3.3V（偏差范围为 3V~3.6V），VDDIO、VDD5、VDD3O 三者需要连接在一起，内部 3.3V LDO 关闭，此时 IO 以及 PHY 对外接口电压为 3.3V。

③当电源电压变化范围为 2.6V~5.5V 时，VDDIO 需要和 VDD5 短接在一起,内部 3.3V LDO 关闭，此时 USB 相关功能不能使用。

3.4.3 相关寄存器

3.4.3.1地址映射

寄存器	地址	初始值	说明
PWRCON	0x9F	0xE0	电源管理寄存器

3.4.3.2寄存器描述

3.4.3.2.1 PWRCON：电源管理寄存器

位	名称	读写	描述
7	LDO33EN	R/W	3.3V LDO 控制位 1 = 3.3V LDO 工作模式由 LDO33MD 控制 0 = 3.3V LDO 强制关闭
6	LDO33MD	R/W	3.3V LDO 模式控制位 1 = 3.3V LDO 一直开启 0 = 3.3V LDO 工作/IDLE 模式下开启；STOP 模式下自动关闭
5	LDO18MD	R/W	1.8V LDO 模式控制位 1 = 1.8V 大 LDO 一直开启 0 = 1.8V 大 LDO 工作/IDLE 模式下开启；STOP 模式下自动关闭 注：1.8V 小 LDO 处于常开状态，不能被关闭。
4	保留	R	-
3	保留	R/W	-

2:1	EFT_CTRL	R/W	EFT 控制位 00 = 关闭 EFT 检测模块 01 = 在检测到 EFT 干扰后产生 EFT 中断 10 = 在检测到 EFT 干扰时提前停止 MCU 内核 11 = 在检测到 EFT 干扰后自动停止 MCU 内核
0	EFTIF	R/W	EFT 中断标志寄存器 1 = 发生了 EFT 中断 0 = 未发生 EFT 中断 硬件无法自动清零，软件写‘0’清零，写 1 无效。

3.5 中断系统

3.5.1 概述

- 当一个片内外设或外部源满足有效地中断条件时，相应的中断标志被置为逻辑'1'。中断标志置'1'与否不受中断允许/禁止状态的影响。
- 如果一个中断源被允许，则在中断标志被置位时将产生一个中断。
- 一旦当前指令执行完，CPU 产生一个 LCALL 到预定地址，开始执行中断服务程序(ISR)
- 每个 ISR 必须以 RETI 指令结束,使程序回到中断前执行的那条指令的下一条指令.

3.5.2 特征

- 4 级中断优先级
- 相同优先级下的中断轮询队列
- 外部中断源电平检测，沿检测可选

3.5.3 功能说明

3.5.3.1MCU 中断源和中断向量

MCU 支持 20 个中断源。如果中断标志被允许，系统将产生一个中断请求，CPU 将转向与该中断标志对应的 ISR 地址。

3.5.3.2中断屏蔽

SG8F7581 设有一个全局中断控制位 EA(IEN0 bit7)作为全局中断开关。

另外，中断系统对于 17 个中断源设有各自独立的控制位（IEN0 bit6~bit0，IEN1，IEN2）；用于控制各个中断源开启/屏蔽。

中断源被屏蔽后，MCU 仍可通过相应中断标志位识别被屏蔽的中断。

SG8F7581 某些中断源中含有多个中断标志位（如 USB 中断），则每个中断标志位设有独立的

使能开关，具体设置参考各模块相关寄存器描述。

3.5.3.3端口中断

SG8F7581 一共有 58 个 IO 引脚，分为 P0~P7 共 5 组，其中每个引脚都有独立的端口状态检测模块，用于送出端口中断唤醒信号。端口中断的产生不受系统时钟影响，在 STOP 模式下系统时钟源关闭时，端口变化仍能够产生中断唤醒信号，来唤醒 MCU 并产生对应的中断标志。

P0/P1/P2/P3/P6/P7 共用一个中断入口，各组端口都只有一个中断标志位（SFR:PORTIF 寄存器，P0IF/P1IF/P2IF/P3IF/P6IF/P7IF 位），由 PORTIE 相应位控制对各组端口中断分别做屏蔽；另外可通过 PxIE(IEN0 bit0)对 PORTIF 整组中断源作中断入口屏蔽。

P4/P5 两组端口中每个端口都拥有独立的中断标志寄存器（SFR:P4IF P40IF~P47IF，SFR:P5IF P50IF~P57IF）和中断使能控制寄存器（SFR:P4IE P40IE~P47IE，SFR:P5IE P50IE~P57IE），分别控制 P4/P5 各端口的中断使能及标志。另外对于 P4/P5 端口中断，由 P4IE/P5IE(SFR：IEN2 bit6/bit5)控制其中断源入口开启/屏蔽，如果 P4IE=0，MCU 将不会响应来自 P4 任一端口的中断，P5IE 同样的道理。

3.5.3.4外部中断 INT0/INT1

SG8F7581 设置引脚 P01、P43 分别作为外部中断 INT1/INT0 输入脚。注意外部中断（INT0/INT1）与 P01/P43 的端口中断是两种不同的中断（由不同的中断检测模块进行中断检测），前者作为 SG51 通用的外部中断输入，中断标志位——IE0/IE1，中断使能位——EINT0/EINT1，与后者有不同的中断入口地址。

不同于端口中断，外部中断可在 IDLE 模式下唤醒 MCU，但无法在 STOP 模式下唤醒 MCU 或产生中断（STOP 模式下，端口中断检测模块仍可工作，但外部中断检测模块停止工作）。

两个外部中断源 INT0 和 INT1 可被配置为低电平有效或高电平有效，边沿触发或电平触发。CKCON 寄存器中的 IN0PL（INT0 极性）和 IN1PL（INT1 极性）位用于选择高电平有效还是低电平有效；TCON 中的 IT0 和 IT1 用于选择电平或边沿触发。下面的表列出了可能的配置组合。

芯片检测到对应的外部中断引脚状态变化时，会置起相应的中断标志位 IE0/IE1(TCON 寄存器 bit1/bit3)；如果此时外部中断使能开启（IEN0 bit6/bit4），MCU 会响应该中断，转入中断处理程序，并自动清零对应中断标志位 IE0/IE1。

IT0	IN0PL	INT0 中断

IT1	IN1PL	INT1 中断

1	0	低电平有效，边沿触发	1	0	低电平有效，边沿触发
1	1	高电平有效，边沿触发	1	1	高电平有效，边沿触发
0	0	低电平有效，电平触发	0	0	低电平有效，电平触发
0	1	高电平有效，电平触发	0	1	高电平有效，电平触发

3.5.3.5中断标志清除

SG8F7581 中，外部中断 0/外部中断 1/TIMER0 中断/TIMER1 中断这四种中断源的中断标志在 CPU 响应中断后会自动清除，因此软件在各自的中断处理子程序中读取到的对应中段标志位一直为 0，无需软件清 0。另外需要注意如果软件向这四种中断标志位内写‘1’则会置起对应中断标志位，此时如果相应的中断请求被允许，CPU 将会立刻响应该中断。

其它中断源的中断标志无法硬件自动清除，需要软件写‘0’清除；而软件写‘1’无任何效果。

3.5.3.6中断优先级

- 共设有 4 级中断优先级
- 17 个中断源被分为 6 组，每组都可以被编程为 4 级优先级中的任意一级。中断源分组如下：

group	中断源			
0	外部中断 0	timer0 溢出	外部中断 1	-
1	timer1 溢出	Uart 中断	timer2 中断	-
2	PORT 端口中断	I2C 中断	USB 中断	-
3	ADC 中断	DMA 中断		-
4	比较器 0	比较器 1	WDT 中断	EFT 中断
5	P4 端口中断	P5 端口中断		

- 一个低优先级的中断服务程序可以被高优先级的中断所中断，但高优先级的中断不能被中断。
- 每个中断在 SFR(IP0,IP1)中都有一对配置其优先级的中断优先级设置位，缺省值为最低优先级，优先级配置如下：

IP1.n	IP0.n	Group n 组中中断源优先级配置
0	0	level0(最低优先级，默认值)
0	1	level1

1	0	level2
1	1	level3(最高优先级)

- 如果两个中断同时发生，具有高优先级的中断先得到服务。
- 如果两个中断同时发生，它们的优先级相同，则有固定的轮询队列决定哪个一个中断先得到服务。

3.5.3.7中断响应时间

中断响应时间取决于中断发生时 CPU 的状态。中断系统在每个系统时钟周期对中断标志采样并对优先级译码。最快的响应时间为 6 个系统时钟周期：一个周期用于检测中断，5 个周期完成对 ISR 的长调用（LCALL）。如果中断标志有效时 CPU 正在执行 RETI 指令，则需要再执行一条指令才能进入中断服务程序。如果 CPU 正在执行一个具有相同或更高优先级的中断的 ISR，则新中断要等到当前 ISR 执行完（包括 RETI 和下一条指令）才能得到服务。

注意：在 FLASH 写/擦除操作期间以及外设通过 DMA 访问 XRAM 时 CPU 执行 MOVX 指令期间，CPU 暂停执行指令。对于在 CPU 暂停执行指令期间发生的中断，中断服务响应时间将延长。这些情况下的中断延迟时间由标准中断服务响应过程（如前所述）和 CPU 暂停执行指令的时间决定。

3.5.3.8中断参照表

中断源	中断向量	轮询队列	中断标志	硬件清除	中断允许	Group	优先级控制
外部中断 0	0x0003	轮询队列由上到下	IE0(TCON.1)	Y	EINT0(IE0.6)	0	ip1.0 ip0.0
timer0 溢出	0x000B		TF0(TCON.5)	Y	T0IE(IE0.5)		
外部中断 1	0x0013		IE1(TCON.3)	Y	EINT1(IE0.4)		
timer1 溢出	0x001B		TF1(TCON.6)	Y	T1IE(IE0.3)	1	ip1.1 ip0.1
UART 中断	0x0023		TI(SCON.1) RI(SCON.0)	N	UARTIE(IE0.2)		
timer2 溢出	0x002B		TF2H(TMR2CN.7) TF2L(TMR2CN.6)	N	T2IE(IE0.1)		
PORT 端口	0x0033		P7IF(PORTIF.7) P6IF(PORTIF.6)	N	PXIE(IE0.0)	2	ip1.2 ip0.2

			P3IF(PORTIF.3) P2IF(PORTIF.2) P1IF(PORTIF.1) P0IF(PORTIF.0)				
I2C 中断	0x003B		特殊	N	I2CIE(IE1.7)		
USB 中断	0x0043		特殊	N	USBIE(IE1.6)		
ADC	0x004B		ADCIF	N	ADCIE(IE1.5)		
DMA 中断	0x0053		DCH1IF (DMAIF.0) DCH2IF(DMAIF.1) DCH3IF(DMAIF.2) DCH4IF(DMAIF.3) DCH5IF(DMAIF.4)	N	DMAIE(IE1.4)	3	ip1.3 ip0.3
比较器 0	0x005B		CP0RIF(CPINT.3) CP0FIF(CPINT.2)	N	CP0IE(IE1.2)		
比较器 1	0x0063		CP1RIF(CPINT.7) CP1FIF(CPINT.6)	N	CP1IE(IE1.1)	4	ip1.4 ip0.4
WDT 中断	0x006B		WDTIF(WDTCF.2)	N	WDTIE(IE1.0)		
EFT 中断	0x0073		EFTIF(PWRCON.0)	N	EFTIE(IE2.7)		
P4 端口	0x007B		P4IF bit7~bit0	N	PORT4IE(IE2.6)	5	ip1.5 ip0.5
P5 端口	0x0083		P5IF bit7~bit0	N	PORT5IE(IE2.5)		

3.5.4 相关寄存器

3.5.4.1地址映射

寄存器	地址	初始值	描述
IEN0	0xA8	0x00	中断允许寄存器 0
IEN1	0xB8	0x00	中断允许寄存器 1
IEN2	0xB9	0x00	中断允许寄存器 2
IP0	0xBA	0x00	中断优先级配置寄存器 0

IP1	0xBB	0x00	中断优先级配置寄存器 1

3.5.4.2寄存器描述

3.5.4.2.1 IEN0：中断允许寄存器 0

位	寄存器名	读写	说明
7	EA	R/W	全局中断控制位： 0：全局中断使能关闭（仍可读取对应的中断标志位） 1：全局中断使能开启，每个中断由相应的中断使能位控制。
6	EINT0	R/W	外部中断 0 控制： 0：禁止外部中断 0 1：允许 INT0 的中断请求
5	T0IE	R/W	timer0 中断控制 0：禁止 timer0 中断 1：允许 timer0 中断请求
4	EINT1	R/W	外部中断 1 控制： 0：禁止外部中断 1 1：允许 INT1 的中断请求
3	T1IE	R/W	timer1 中断控制 0：禁止 timer1 中断 1：允许 timer1 中断请求
2	UARTIE	R/W	UART 中断控制 0：禁止 UART 中断 1：允许 UART 中断请求
1	T2IE	R/W	timer2 中断控制 0：禁止 timer2 中断 1：允许 timer2 中断请求
0	PxIE	R/W	PORTIF 中断控制： 0：禁止 SFR: PORTIF 中所列的端口中断 1：允许 SFR: PORTIF 中所列的端口中断请求

3.5.4.2.2 IEN1：中断允许寄存器 1

位	寄存器名	读写	说明
7	I2CIE	R/W	I2C 中断控制 0：禁止 I2C 中断 1：允许 I2C 中断请求
6	USBIE	R/W	USB 中断控制 0：禁止 USB 中断 1：允许 USB 中断请求
5	ADCIE	R/W	ADC 中断控制： 0：禁止 ADC 中断 1：允许 ADC 中断请求
4	DMAIE	R/W	DMA 中断控制 0：禁止 DMA 中断 1：允许 DMA 中断请求
3	保留	R	-
2	CP0IE	R/W	比较器 0 中断控制： 0：禁止比较器 0 中断 1：允许比较器 0 中断请求
1	CP1IE	R/W	比较器 1 中断控制： 0：禁止比较器 1 中断 1：允许比较器 1 中断请求
0	WDTIE	R/W	WDT 中断控制 0：禁止 WDT 中断 1：允许 WDT 中断

3.5.4.2.3 IEN2：中断允许寄存器 2

位	寄存器名	读写	说明
7	EFTIF	R/W	EFT 中断控制位： 0：禁止 EFT 中断 1：允许 EFT 中断
6	PORT4IE	R/W	P4 端口中断控制： 0：禁止 P4 端口中断 1：允许 P4 端口中断请求
5	PORT5IE	R/W	P5 端口中断控制： 0：禁止 P5 端口中断 1：允许 P5 端口中断请求
4~0	保留	-	保留

3.5.4.2.4 IP0~IP1：中断优先级控制寄存器

✧ IP0

位	寄存器名	读写	说明
7~6	保留	-	保留
5~0	IP0.5~ IP0.0	-	优先级配置位 配置方式见下下表

✧ IP1

位	寄存器名	读写	说明
7~6	保留	-	保留
5~0	IP1.5 ~ IP1.0	-	优先级配置位 配置方式见下表

✧ IP1,IP0 优先级配置方式

IP1.x	IP0.x	Group x 中断源优先级

0	0	level0(最低优先级，默认值)
0	1	level1
1	0	level2
1	1	level3(最高优先级)

3.6 复位系统

3.6.1 概述

- 多个复位源：
  - ✧ 上电复位
  - ✧ 低压复位
  - ✧ FLASH 读取错误复位
  - ✧ 软件复位
  - ✧ USB 复位
  - ✧ 外部复位

3.6.2 功能描述

3.6.2.1上电复位

在上电期间，MCU 保持在复位状态，直到 VDD 上升到超过 V_{RST} 电平。从复位开始到退出复位状态要经过一个上电复位延时（$T_{PORDelay}$）。

上电复位结束时，PORSF（RSTRSFbit0）和 LVRSF（RSTRSF bit1）被硬件置‘1’，此时 RSTSRC 寄存器中的所有其它复位标志都是不确定的。PORSF 可被除了上电复位之外的任何复位源清 0（包括低压复位）。由于所有的复位都导致程序从同一个地址（0x0000）开始执行，软件可以通过读 PORSF 标志来确定是否为上电产生的复位。

3.6.2.2低压复位

芯片工作期间，如果电源电压低于低压复位阈值电压（1.5V 左右），如果芯片的低压复位功能开启，芯片将强行进入系统复位状态，直至电源电压重新高于该阈值电压。

上电复位结束后，芯片低压复位功能处于开启状态。可通过更改 LVRSF 寄存器（RSTRSF bit1）写入值来开启或关闭低压复位功能。

低压复位结束时，硬件将 LVRSF 标志位自动置“1”，并将其它状态标志清零；

3.6.2.3外部复位

SG8F7581 内置一个外部复位检测模块(RSTDet)，用于检测外部复位端口 RST 的输入信号。

当外部复位端口 RST 保持低电平时间大于 50us 时，外部复位检测模块将产生复位信号对全芯片进行系统复位，同时，PINRSF 标志位（RSTRSF bit4）将被硬件置 1，其余复位标志位自动清零。

3.6.3 相关寄存器

3.6.3.1地址映射

寄存器	地址	复位值	说明
RSTRSF	0xEF	0uuu_uuuu	复位控制/状态寄存器

3.6.3.2寄存器描述

3.6.3.2.1 RSTRSF：复位控制/状态寄存器

位	名称	读写	说明
7	保留		保留
6	USBRSF	R/W	读：USB 复位标志信号 1：最后一次系统复位源来自 USB 0：最后一次系统复位源不是来自 USB 写：USB 复位使能控制 1：开启 USB 作为系统复位源的功能 0：关闭 USB 作为系统复位源的功能（默认）
5	FLASHRSF	R	读：FLASH 读写错误复位标志

			1：最后一次系统复位源来自 FLASH 读取（读取地址溢出）错误 0：最后一次系统复位源不是来自 FLASH 读取（读取地址溢出）错误
4	PINRSF	R	读：芯片外部管脚复位标志 1：最后一次系统复位源来自外部管脚复位 0：最后一次系统复位源不是来自外部管脚复位
3	SWRSF	R/W	读：软件强制复位标志 1：最后一次系统复位源来自软件复位 0：最后一次系统复位源不是来自软件复位 写 1：执行一次软件强制复位。写 0 无任何意义
2	WDTRSF	R/W	读：看门狗定时器复位标志 1：最后一次系统复位源来自 WDT 溢出复位 0：最后一次系统复位源不是来自 WDT 溢出复位 写：看门狗复位使能控制 1：开启 WDT 作为系统复位源的功能（默认） 0：关闭 WDT 作为系统复位源的功能
1	LVRSF	R/W	读：低压复位标志 1：最后一次系统复位源来自低压复位或上电复位 0：最后一次系统复位源不是来自低压复位和上电复位 写： 1：低压检测复位功能开启 0：低压检测复位功能关闭
0	PORSF	R/W	读：低压复位标志 1：最后一次系统复位源来自上电复位，这时候除 LVRSF 外的其它的复位标志的值不确定 0：最后一次系统复位源不是来自上电复位

3.7 通用 IO 控制器

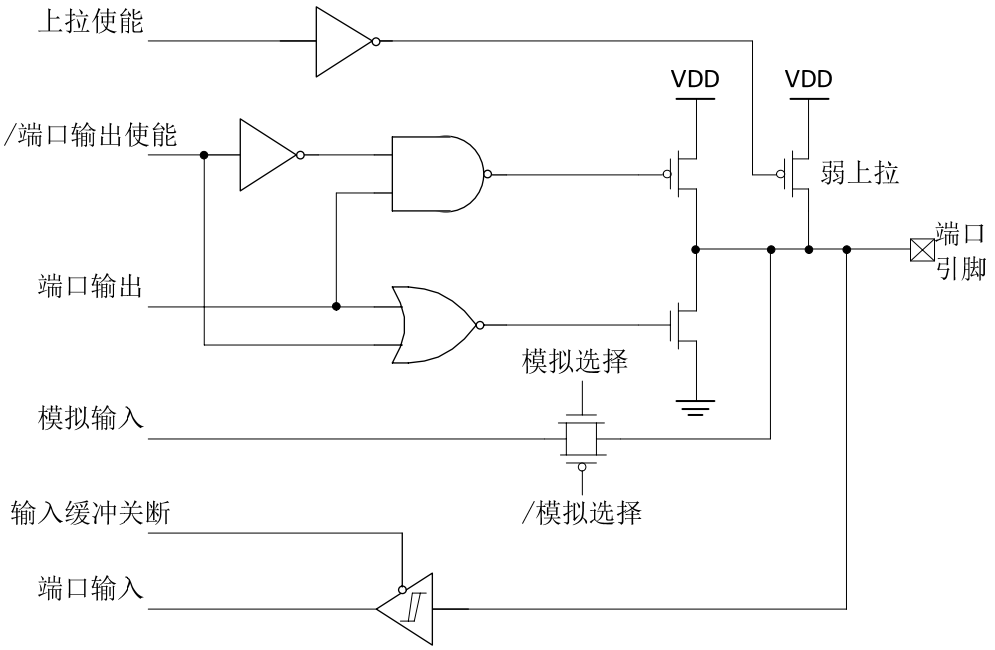
3.7.1 概述

数字和模拟资源可以通过 58 个 I/O 管脚使用，每个端口都可以被定义为通用的 I/O(GPIO)或模拟端口（由端口复用模块控制是否作为模拟端口）。

3.7.2 特征

- 可配置为模拟或数字端口（由端口复用模块控制）
- 可配置为双向端口
- 开漏输出功能
- IO 输出 2 档驱动能力可选 (4mA/20mA)
- 内置可控上拉
- 部分 IO 可控下拉
- 可作为中断唤醒源
- P0/P1/P2/P3/P6/P7 每组端口共用一个中断标志位（PORTIF 相应位）
- P4/P5 端口中， 每个端口都有独立的中断标志位及中断使能控制
- P70/P71 端口可配置作为 USB DP/DM 端口、或 PS2_CLK/PS2_DATA 端口

3.7.3 端口单元框图



3.7.4 开漏功能

SG8F7581 并未配置专门的开漏输出控制寄存器。如果需要实现开漏功能，可将上图中“端口输出”配置为‘0’，之后通过控制切换“端口输出使能”来实现引脚对外输出‘0’或‘高阻’状态。

3.7.5 驱动能力选择

SG8F7581 为 P0~P6 引脚配置了驱动能力控制寄存器 XFR:P0HDRV~P6HDRV，用于控制端口驱动能力。通过 P0HDRV~P6HDRV 控制，P0~P6 所有引脚都有 4mA、20mA 两档输出驱动电流能力可供选择。

3.7.6 可控上拉

SG8F7581 为 P0~P6 引脚配置有可控上拉功能，由 XFR 寄存器 XFR:P0PU~P6PU 控制。芯片复位时上拉功能默认开启（P55 引脚除外，该引脚上电/复位时默认开启端口下拉）。

另外 P0~P6 引脚上拉电阻分 10KΩ 和 20KΩ 两档，可由 codeoption：SEL_RES 位选择统一配置（需要烧写 FLASH）。

注：P70~P71 配置有 USB 上拉（SFR:USBCON bit6）和 PS2 上拉(SFR:USBCON bit0)，分

别用于 P70/P71 的 USB/PS2 功能，但这两个端口用作在 GPIO 端口时无可控上拉（内置 2MΩ 下拉电阻）

3.7.7 可控下拉

SG8F7581 为部分引脚配置有可控下拉功能(P20~P27，P55 引脚)，由 XFR 寄存器 P2PD/P5PD 控制；下拉电阻阻值约 10KΩ。芯片上电/复位时 P2 下拉功能默认关闭，需要软件开启；而 P55 引脚下拉功能默认处于开启状态。

3.7.8 相关寄存器

3.7.8.1地址映射

寄存器	地址	复位值	说明
P0	0x80	0x00	P0 端口数据寄存器
P1	0x90	0x00	P1 端口数据寄存器
P2	0xA0	0x00	P2 端口数据寄存器
P3	0xB0	0x00	P3 端口数据寄存器
P4	0xC0	0x00	P4 端口数据寄存器
P5	0xD8	0x00	P5 端口数据寄存器
P6	0xE8	0x00	P6 端口数据寄存器
P7	0xF8	0x00	P7 端口数据寄存器
P0DIR	0x91	0xFF	P0 端口方向控制寄存器
P1DIR	0x92	0xFF	P1 端口方向控制寄存器
P2DIR	0x93	0xFF	P2 端口方向控制寄存器
P3DIR	0x94	0xFF	P3 端口方向控制寄存器
P4DIR	0x95	0xFF	P4 端口方向控制寄存器
P5DIR	0x96	0xFF	P5 端口方向控制寄存器
P6DIR	0x97	0xFF	P6 端口方向控制寄存器
P7DIR	0x9E	0x03	P7 端口方向控制寄存器
PORTIE	0xA2	0x00	P0/P1/P2/P3/P6/P7 端口中断屏蔽寄存器
PORTIF	0xA3	0x00	P0/P1/P2/P3/P6/P7 端口中断标志寄存器
P4IE	0xA4	0x00	P4 端口中断屏蔽寄存器
P5IE	0xA5	0x00	P5 端口中断屏蔽寄存器
P4IF	0xA6	0x00	P4 端口中断标志寄存器
P5IF	0xA7	0x00	P5 端口中断标志寄存器
P0PU	0xFF00	0xFF	**XFR**:P0 端口上拉使能控制寄存器
P1PU	0xFF01	0xFF	**XFR** :P1 端口上拉使能控制寄存器
P2PU	0xFF02	0xFF	**XFR** :P2 端口上拉使能控制寄存器

P3PU	0xFF03	0xFF	**XFR** :P3 端口上拉使能控制寄存器
P4PU	0xFF04	0xFF	**XFR** :P4 端口上拉使能控制寄存器
P5PU	0xFF05	0xDF	**XFR** :P5 端口上拉使能控制寄存器
P6PU	0xFF06	0xFF	**XFR** :P6 端口上拉使能控制寄存器
P2PD	0xFF07	0x00	**XFR** :P2 端口下拉控制寄存器
P0HDRV	0xFF08	0x00	**XFR** :P0 端口驱动电流控制寄存器
P1HDRV	0xFF09	0x00	**XFR** :P1 端口驱动电流控制寄存器
P2HDRV	0xFF0A	0x00	**XFR** :P2 端口驱动电流控制寄存器
P3HDRV	0xFF0B	0x00	**XFR** :P3 端口驱动电流控制寄存器
P4HDRV	0xFF0C	0x00	**XFR** :P4 端口驱动电流控制寄存器
P5HDRV	0xFF0D	0x00	**XFR** :P5 端口驱动电流控制寄存器
P6HDRV	0xFF0E	0x00	**XFR** :P6 端口驱动电流控制寄存器
P5PD	0XFF0F	0x20	**XFR** :P5 端口下拉控制寄存器
PORTAN	0xFF10	0x00	**XFR :**P0/P3/P5/P6/P7 数字输入缓冲控制寄存器
P1AN	0xFF11	0x00	**XFR :**P1 数字输入缓冲控制寄存器
P2AN	0xFF12	0x00	**XFR :**P2 数字输入缓冲控制寄存器
P4AN	0xFF14	0x00	**XFR :**P4 数字输入缓冲控制寄存器
P0INTWK	0xFF18	0x00	**XFR** :P0 端口中断和唤醒检测控制寄存器
P1INTWK	0xFF19	0x00	**XFR** :P1 端口中断和唤醒检测控制寄存器
P2INTWK	0xFF1A	0x00	**XFR** :P2 端口中断和唤醒检测控制寄存器
P3INTWK	0xFF1B	0x00	**XFR** :P3 端口中断和唤醒检测控制寄存器
P4INTWK	0xFF1C	0x00	**XFR** :P4 端口中断和唤醒检测控制寄存器
P5INTWK	0xFF1D	0x00	**XFR** :P5 端口中断和唤醒检测控制寄存器
P6INTWK	0xFF1E	0x00	**XFR** :P6 端口中断和唤醒检测控制寄存器
P7INTWK	0xFF1F	0x00	**XFR** :P7 端口中断和唤醒检测控制寄存器

3.7.8.2寄存器描述

3.7.8.2.1 端口数据寄存器

3.7.8.2.1.1 P0~P6 端口数据寄存器

位	名称	读写	说明
7~0	Pmn (m=6~0, n=7~0)	R/W	端口数据寄存器 写操作——写入端口寄存器供 I/O 引脚输出的数据 1：输出逻辑高电平 0：输出逻辑低电平 注 1：如果 PxDIR 将端口设为输入端口，或端口复用至其他功能(如 I2C/UART 复用等)时，Pmn 寄存器的值将不再影响端口状态。 读操作——读取 I/O 引脚当前的逻辑电平 1：相应引脚为逻辑高电平 0：相应引脚为逻辑低电平 注 2：先读后写类指令（ANL、ORL、XRL、JBC、CPL、INC、DEC、DJNZ）以及位操作指令（MOV，CLR，SETB）操作该寄存器时，先读出端口寄存器（非引脚逻辑电平）的值，修改后再写回 SFR。

3.7.8.2.1.2 P7 端口数据寄存器

位	名称	读写	说明
7~2	保留	R	-
1	P71	R/W	P71 端口数据寄存器 写操作——写入端口寄存器供 I/O 引脚输出的数据 1：输出逻辑高电平 0：输出逻辑低电平 注 1：如果 P71DIR 关闭端口输出使能，或 USBUP=1（USBCON bit6）时，寄存器的值将不再影响端口状态。 读操作——读取 I/O 引脚当前的逻辑电平

			1：相应引脚为逻辑高电平 0：相应引脚为逻辑低电平 注 2：先读后写类指令（ANL、ORL、XRL、JBC、CPL、INC、DEC、DJNZ）以及位操作指令（MOV，CLR，SETB）操作该寄存器时，先读出端口寄存器（非引脚逻辑电平）的值，修改后再写回 SFR。
0	P70	R/W	P70 端口数据寄存器 写操作——写入端口寄存器供 I/O 引脚输出的数据 1：输出逻辑高电平 0：输出逻辑低电平 注 1：如果 P70DIR 关闭端口输出使能，或 USBUP=1（USBCON bit6）时 P70 寄存器的值将不再影响端口状态。 读操作——读取 I/O 引脚当前的逻辑电平 1：相应引脚为逻辑高电平 0：相应引脚为逻辑低电平 注 2：先读后写类指令（ANL、ORL、XRL、JBC、CPL、INC、DEC、DJNZ）以及位操作指令（MOV，CLR，SETB）操作该寄存器时，先读出端口寄存器（非引脚逻辑电平）的值，修改后再写回 SFR。

3.7.8.2.2 端口方向控制寄存器

3.7.8.2.2.1 P0DIR~P6DIR (P0~P6)端口方向控制寄存器

位	名称	读写	说明
7~0	PmnDIR (m=6~0, n=7~0)	R/W	Pmn(m=6~0,n=7~0)端口方向控制位 1 = 相应引脚为数字输入引脚，数字输出功能关闭 0 = 相应引脚数字输入输出功能开启（数字输出功能开启，且 MCU 仍可读取当前端口逻辑电平）

3.7.8.2.2.2 P7DIR(P7)端口方向控制寄存器

位	名称	读写	说明

7~2	保留	R	-
1	P71DIR	R/W	P71 端口方向控制位 1 = P71 引脚为数字输入引脚，数字输出功能关闭 0 = P71 引脚数字输入输出功能开启（数字输出功能开启，且 MCU 仍可读取当前端口逻辑电平）
0	P70DIR	R/W	P70 端口方向控制位 1 = P71 引脚为数字输入引脚，数字输出功能关闭 0 = P71 引脚数字输入输出功能开启（数字输出功能开启，且 MCU 仍可读取当前端口逻辑电平）

3.7.8.2.3 数字输入缓冲控制寄存器

3.7.8.2.3.1 PORTAN(P0/P3/P5/P6)数字输入缓冲控制寄存器

位	名称	读写	说明
7	保留	R	-
6	P6AN	R/W	P60~P67 数字输入缓冲控制位 0= 开启 P60~P67 数字输入功能 1 = 关闭P60~P67数字输入缓冲功能(上电默认,引脚读出值恒为‘1’)
5	P5AN	R/W	P50~P57 数字输入缓冲控制位 0 = 开启 P50~P57 数字输入功能 1 = 关闭P50~P57数字输入缓冲功能(上电默认,引脚读出值恒为‘1’)
4	保留	R	-
3	P3AN	R/W	P30~P37 数字输入缓冲控制位 0 = 开启 P30~P37 数字输入功能 1 = 关闭P30~P37数字输入缓冲功能(上电默认,引脚读出值恒为‘1’)
2	保留	R	-
1	保留	R	-
0	P0AN	R/W	P00~P07 数字输入缓冲控制位 0 = 开启 P00~P07 数字输入功能 1 = 关闭P00~P07数字输入缓冲功能(上电默认,引脚读出值恒为‘1’)

3.7.8.2.3.2 P1AN(P10~P17)数字输入缓冲控制寄存器

位	名称	读写	说明
7~0	P1nAN (n=7~0)	R/W	P1n（n=7~0）数字输入缓冲控制位 0 = 开启 P1n 数字输入功能 1 = 关闭 P1n 数字输入缓冲功能（上电默认,引脚读出值恒为‘1’）

3.7.8.2.3.3 P2AN(P20~P27)数字输入缓冲控制寄存器

位	名称	读写	说明
7~0	P2nAN (n=7~0)	R/W	P2n（n=7~0）数字输入缓冲控制位 0 = 开启 P2n 数字输入功能 1 = 关闭 P2n 数字输入缓冲功能（上电默认,引脚读出值恒为‘1’）

3.7.8.2.3.4 P4AN(P40~P47)数字输入缓冲控制寄存器

位	名称	读写	说明
7~0	P4nAN (n=7~0)	R/W	P4n（n=7~0）数字输入缓冲控制 0 = 开启 P4n 数字输入功能 1 = 关闭 P4n 数字输入缓冲功能（上电默认,引脚读出值恒为‘1’）

3.7.8.2.4 P0PU~P6PU(P0~P6)端口上拉控制寄存器

位	名称	读写	说明
7~0	PmnPU (m=6~0, n=7~0)	R/W	Pmn(m=6~0,n=7~0)端口上拉控制寄存器 1 = 相应引脚内部上拉开启 0 = 相应引脚内部上拉关闭

注：P55PU 复位默认值为‘0’，其余为‘1’

3.7.8.2.5 端口下拉控制寄存器

3.7.8.2.5.1 P2PD(P20~P27)下拉控制寄存器

位	名称	读写	说明

7~0	P2nPD (n=7~0)	R/W	P2n(n=7~0)端口下拉控制寄存器 1 = 相应引脚内部下拉开启 0 = 相应引脚内部下拉关闭（默认）

3.7.8.2.5.2 P5PD(P55)下拉控制寄存器

位	名称	读写	说明
7~6	保留	R	-
5	P55PD	R/W	P55 端口下拉控制寄存器 1 = P55 引脚内部下拉开启(默认) 0 = P55 引脚内部下拉关闭
4~0	保留	R	-

3.7.8.2.6 P0HDRV~P6HDRV 端口驱动电流控制寄存器

位	名称	读写	说明
7~0	PmnHDRV (m=6~0， n=7~0)	R/W	Pmn(m=6~0,n=7~0)端口输出驱动电流控制寄存器 1 = 相应引脚驱动电流 20mA 0 = 相应引脚驱动电流 4mA（默认）

3.7.8.2.7 端口中断唤醒检测控制寄存器

***注：**该功能开启，检测相应的端口状态变化时，

如果芯片处于工作状态，则会产生相应中断（对应中断控制位 PxxIE 使能时）；

如果芯片处于 IDLE 或 STOP 模式，则会将芯片唤醒并将对应的中断标志位置 1，然后芯片进入工作模式并按中断优先级处理该中断（对应中断控制位 PxxIE 使能时）；

3.7.8.2.7.1 P0INTWK~P6INTWK (P0~P6)端口状态检测控制寄存器

位	名称	读写	说明
7~0	PmnINTWK (m=6~0, n=7~0)	R/W	Pmn(m=7~0,n=7~0)端口状态检测功能控制位 端口检测结果用作唤醒源（STOP 或 IDLE 模式下）和中断源 1 = 开启端口状态变化检测功能；

			0 = 关闭端口状态变化检测功能；（默认） *注：P0~P6 如果输出使能，将自动关闭端口状态变化检测功能。

3.7.8.2.7.2 P7INTWK (P7)端口状态检测控制寄存器

位	名称	读写	说明
7~2	保留	R	-
1	P71INTWK	R/W	P71 端口状态检测功能控制位 1 = 开启端口状态变化检测功能； 0 = 关闭端口状态变化检测功能；（默认）
0	P70INTWK	R/W	P70 端口状态检测功能控制位 1 = 开启端口状态变化检测功能； 0 = 关闭端口状态变化检测功能；（默认）

3.7.8.2.8 端口中断控制/标志寄存器

3.7.8.2.8.1 PORTIF(P0/P1/P2/P3/P6/P7)端口中断标志寄存器

位	名称	读写	说明
7	P7IF	R/W	P7 端口中断标志位： 1：P7 端口组中有端口产生中断； 0：P7 端口组没有有端口产生中断。 硬件无法清除，软件写“0”清 0。
6	P6IF	R/W	P6 端口中断标志位： 1：P6 端口组中有端口产生中断； 0：P6 端口组没有有端口产生中断。 硬件无法清除，软件写“0”清 0。
5~4	保留	R	-
3	P3IF	R/W	P3 端口中断标志位： 1：P3 端口组中有端口产生中断； 0：P3 端口组没有有端口产生中断。 硬件无法清除，软件写“0”清 0。

2	P2IF	R/W	P2 端口中断标志位： 1：P2 端口组中有端口产生中断； 0：P2 端口组没有有端口产生中断。 硬件无法清除，软件写“0”清 0。
1	P1IF	R/W	P1 端口中断标志位： 1：P1 端口组中有端口产生中断； 0：P1 端口组没有有端口产生中断。 硬件无法清除，软件写“0”清 0。
0	P0IF	R/W	P0 端口中断标志位： 1：P0 端口组中有端口产生中断； 0：P0 端口组没有有端口产生中断。 硬件无法清除，软件写“0”清 0。

3.7.8.2.8.2 P4IF(P4)端口中断标志寄存器

位	名称	读写	说明
7~0	P4nIF (n=7~0)	R/W	P4n(n=7~0)端口中断标志位 1：P4n 端口产生生端口中断； 0：P4n 端口未产生中断。 硬件无法清除，软件写“0”清 0。

3.7.8.2.8.3 P5IF(P5)端口中断标志寄存器

位	名称	读写	说明
7~0	P5nIF (n=7~0)	R/W	P5n(n=7~0)端口中断标志位 1：P5n 端口产生生端口中断； 0：P5n 端口未产生中断。 硬件无法清除，软件写“0”清 0。

3.7.8.2.8.4 PORTIE (P0/P1/P2/P3/P6/P7)端口中断控制寄存器

位	名称	读写	说明
7	P7IE	R/W	P7 端口中断使能位：

			1：允许 P7 端口中断； 0：屏蔽 P7 端口中断。
6	P6IE	R/W	P6 端口中断使能位： 1：允许 P6 端口中断； 0：屏蔽 P6 端口中断。
5~4	保留	R	-
3	P3IE	R/W	P3 端口中断使能位： 1：允许 P3 端口中断； 0：屏蔽 P3 端口中断。
2	P2IE	R/W	P2 端口中断使能位： 1：允许 P2 端口中断； 0：屏蔽 P2 端口中断。
1	P1IE	R/W	P1 端口中断使能位： 1：允许 P1 端口中断； 0：屏蔽 P1 端口中断。
0	P0IE	R/W	P0 端口中断使能位： 1：允许 P0 端口中断； 0：屏蔽 P0 端口中断。

3.7.8.2.8.5 P4IE(P4)端口中断控制寄存器

位	名称	读写	说明
7~0	P4nIE (n=7~0)	R/W	P4n(n=7~0)端口中断控制位： 1：允许 P4n 端口中断； 0：屏蔽 P4n 端口中断。

3.7.8.2.8.6 P5IE(P5)端口中断控制寄存器

位	名称	读写	说明
7~0	P5nIE (n=7~0)	R/W	P5n（n=7~0）端口中断控制位： 1：允许 P5n 端口中断； 0：屏蔽 P5n 端口中断。

3.8 计数器/定时器

3.8.1 Timer 1/0

3.8.1.1概述

SG8F7581 包含两个 16 位的通用计数器/定时器：Timer0、 Timer1，都可配置为定时器或计数器操作。

Timer 0 和 Timer 1 有四种工作模式可选择——由两个 SFR 寄存器（TMOD 和 TCON）来选择相应的模式配置。

在定时器模式下，定时器 0 或定时器 1 寄存器每个系统时钟周期计数值加 1。

在计数器模式下，当在相应的输入引脚 T0（P00）或 T1(P42)上观察到下降沿时计数器值加 1。识别一个从“1”到“0”的跳变需要 2 个机器周期，所以，最大的输入计数速率为系统时钟频率的 1/2。虽然在占空比上没有限制，但是要确保正确的识别 0 或 1 状态，输入需要在至少 1 个系统时钟周期时间保持稳定。

3.8.1.2相关寄存器

3.8.1.2.1 地址映射

寄存器	地址	初始值	说明
TCON	0x88	0x00	Timer0/Timer1 控制寄存器
TMOD	0x89	0x11	Timer0/Timer1 模式寄存器
TL0	0x8A	0x00	Timer0 计数器低字节
TL1	0x8B	0x00	Timer1 计数器低字节
TH0	0x8C	0x00	Timer0 计数器高字节
TH1	0x8D	0x00	Timer1 计数器低字节
CKCON	0x8E	0x00	Timer0/1 时钟控制寄存器

3.8.1.2.2 寄存器描述

3.8.1.2.2.1 TCON(Timer0/Timer1)控制寄存器

位	名称	读/写	描述
7	**TF1**	R/W	**Timer1 的溢出标志位** 1= Timer1 计数溢出； 0= Timer1 未发生计数溢出； 注：硬件置 1，当 CPU 响应中断时自动清零； 另外，该位允许软件写 1，置起中断标志位。
6	**TR1**	R/W	**Timer1 使能控制位** 1= Timer1 使能； 0= Timer1 停止工作；
5	**TF0**	R/W	**Timer0 的溢出标志位** 1= Timer0 计数溢出； 0= Timer0 未发生计数溢出； 注：硬件置 1，当 CPU 响应中断时自动清零； 另外，该位允许软件写 1，置起中断标志位。
4	**TR0**	R/W	**Timer0 使能控制位** 1= Timer0 使能； 0= Timer0 停止工作；
3	**IE1**	R/W	**INT1 检测状态标志** 1= INT1 引脚检测到相应状态（相应状态由 IT1/IN1PL 定义）； 0= INT1 引脚未检测到相应状态; 注：硬件置 1，CPU 响应中断时自动清零； 另外该位可以硬件置'1'，直接产生中断标志（建议在沿检测状态下进行该动作
2	**IT1**	R/W	**INT1 检测方式选择位** 1= 上升沿/下降沿检测(由 CKCON bit7—IN1PL 决定检测沿，IN1PL=1 时为上升沿检测，IN1PL=0 时为下降沿检测)； 0= 高电平/低电平检测(由 IN1PL 决定检测电平，IN1PL=1 时为

			高电平检测，IN1PL=0 时为低电平检测)；
1	**IE0**	R/W	**INT0 检测状态标志** 1= INT0 引脚检测到相应状态（相应状态由 IT0/IN0PL 定义）； 0= INT0 引脚未检测到相应状态 注：硬件置 1，CPU 响应中断时自动清零；也可软件清零。 另外该位可以硬件置'1'，直接产生中断标志（建议在沿检测状态下进行该动作。
0	**IT0**	R/W	**INT0 检测方式选择位** 1= 上升沿/下降沿检测(由 CKCON bit6—IN0PL 决定检测沿，IN0PL=1 时为上升沿检测，IN0PL=0 时为下降沿检测)； 0= 高电平/低电平检测(由 IN0PL 决定检测电平，IN0PL=1 时为高电平检测，IN0PL=0 时为低电平检测)；

3.8.1.2.2.2 TMOD(Timer0/Timer1)模式寄存器

位	名称	读/写	描述
7	T1GATE	R/W	**Timer1 外部控制开关（INT1）使能位** 1= 开启外部控制开关（P43 用作 INT1 输入脚）。 TR1（TCON bit6）置 1 时，如果 P43 引脚为高电平，Timer1 就会在 T1(P42 端口)的每个下降沿加 1。 0= 禁用外部控制开关。Timer1 计数不受 INT0 控制。 注：T1GATE=1 时，由 P4INTWK 控制的 P43 端口沿检测功能强制关闭，P43 作为 INT1 的电平/边沿检测功能开启（见 TCON 寄存器）；P43 作为 GPIO 功能仍可正常使用。
6	T1_T_C	R/W	**Timer1 计数器/定时器模式选择位** 1= timer1 用作计数器； 0= timer1 用作定时器。
5~4	T1_MOD	R/W	**Timer1 工作方式控制位** 具体工作方式见下表： T1_MOD | 功能说明 00 | 13 位计数器/定时器。由 TL1 寄存器的低 5 位 和

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td>TH1 寄存器的高 8 位组成。 TL1 的高 3 位无效（可设置为零）。</td></tr>
<tr><td>01</td><td>16 位计数器/定时器。(默认)</td></tr>
<tr><td>10</td><td>自动 reload 的 8 位计数器/定时器。Reload 值保存在 TH1。每一个机器周期 TL1 加 1。当 TL1 溢出，TH1 的值就被复制到 TL1。</td></tr>
<tr><td>11</td><td>定时器 1 不工作。</td></tr>
</table>
</td></tr>
<tr><td>3</td><td>T0GATE</td><td>R/W</td><td>Timer0 外部控制开关（INT0）使能位
1= 开启外部控制开关（P01 用作 INT0 输入脚）。
TR0（TCON bit4）置 1 时，如果 P01 引脚为高电平，Timer0 就会在 T0(P00 端口)的每个下降沿加 1。
0= 禁用外部控制开关。Timer0 计数不受 INT0 控制。
注：T0GATE=1 时，由 P0INTWK 控制的 P01 端口沿检测功能强制关闭，P01 作为 INT0 的电平/边沿检测功能开启（见 TCON 寄存器）；P01 作为 GPIO 功能仍可正常使用。</td></tr>
<tr><td>2</td><td>T0_T_C</td><td>R/W</td><td>Timer0 计数器/定时器模式选择位
1= timer0 用作计数器；
0= timer0 用作定时器。</td></tr>
<tr><td>1~0</td><td>T0_MOD</td><td>R/W</td><td>Timer0 工作方式控制位
具体工作方式见下表：
<table>
<tr><td>T0_MOD</td><td>功能说明</td></tr>
<tr><td>00</td><td>13 位计数器/定时器。由 TL0 寄存器的低 5 位 和 TH0 寄存器的高 8 位组成。 TL0 的高 3 位无效（写 TL0 时高 3 位自动清 0）。</td></tr>
<tr><td>01</td><td>16 位计数器/定时器。（默认）</td></tr>
<tr><td>10</td><td>自动 reload 的 8 位计数器/定时器。Reload 值保存在 TH0。每一个机器周期 TL0 加 1。当 TL0 溢出，TH0 的值就被复制到 TL0。</td></tr>
<tr><td>11</td><td>Timer0 的 TH0 和 TL0 作为两个独立的 8 位定时器/计数器工作。此时：</td></tr>
</table>
</td></tr>
</table>

				计数器 TL0 受 TR0 和 T0GATE 控制，计数溢出时 TF0 置 1。 计数器 TH0 受 TR1 控制，计数溢出时 TF1 置 1（TIMER1 不再工作）。

3.8.1.2.2.3 CKCON(Timer0/1/2)时钟控制寄存器

位	名称	读/写	描述
7	IN1PL	R/W	INT1 极性控制位 1= INT1 检测方式为高电平或上升沿 0= INT1 检测方式为低电平或下降沿
6	IN0PL	R/W	INT0 极性控制位 1= INT0 检测方式为高电平或上升沿 0= INT0 检测方式为低电平或下降沿
5~4	保留	R	-
3	T1CKCON	R/W	定时器 1 时钟源选择位： 0：Timer1 定时器模式下使用系统时钟（默认） 1：Timer1 定时器模式下使用分频位 SCA[1:0]定义的时钟
2	T0CKCON	R/W	定时器 0 时钟源选择为： 0：Timer0 定时器模式使用系统时钟（默认） 1：Timer0 定时器模式使用分频位 SCA[1:0]定义的时钟
1-0	SCA[1:0]	R/W	定时器 0/1 分频时钟选择： 如果定时器 0/1 被配置为使用分频时钟，则这些位控制时钟分频数。 SCA[1] SCA[0] 分频时钟 0 0 系统时钟/4 0 1 系统时钟/12 1 0 系统时钟/ 8 1 1 外部晶振时钟 注：外部晶振时钟输入将在与系统时钟同步后作为定时器的计数时钟。

3.8.1.2.2.4 TL0(Timer0)低字节寄存器

位	名称	读/写	描述
7~0	**TH0**	R/W	Timer0 高字节计数器

3.8.1.2.2.5 TH0(Timer0)高字节寄存器

位	名称	读/写	描述
7~0	**TH0**	R/W	Timer0 高字节计数器

3.8.1.2.2.6 TL1(Timer1)低字节寄存器

位	名称	读/写	描述
7~0	**TL1**	R/W	Timer1 低字节计数器

3.8.1.2.2.7 TH1(Timer1)高字节寄存器

位	名称	读/写	描述
7~0	**TH1**	R/W	Timer1 高字节计数器

3.8.2 Timer 2

3.8.2.1概述

Timer2 是一个 16 位的计数器/定时器，支持多种工作模式，有多个时钟源可作为其计数时钟。

3.8.2.2工作模式描述

3.8.2.2.1 16 位自动重装模式

当 Timer2 工作在 16 位自动重装模式时，如果 Timer2 溢出，Timer2 会根据重载寄存器（TMR2RLH 和 TMR2RLL）中设置的计数初值重新计数。

3.8.2.2.2 双 8 位自动重装模式

当 Timer2 工作在双 8 位自动重装模式时，两个 8 位定时器 TMR2H 和 TMR2L 独立工作。TMR2H 和 TMR2L 在各自溢出时，分别重载 TMR2RLH 和 TMR2RLL 中设置的计数初值重新开始计数。TMR2CN 中的 TR2 是 TMR2H 的运行控制位，而 TMR2L 在该模式下总是处于运行模式。两个定时

器的时钟源可以通过定时器时钟控制寄存器 TMR2RCK 中 CKCON_T2H 和 CKCON_T2L 位分别控制。在中断使能的情况下，TMR2H 或 TMR2L 任意一个溢出都将产生定时器 2 中断。

3.8.2.2.3 USB 帧起始捕捉模式

当 T2CE=1，T2CSS=0 时，定时器工作在 USB 起始帧捕捉模式，该模式可以通过捕捉 USB 主机发送的 SOF 帧来校准内部高频振荡器。使能 Timer2 计数后，每次收到 SOF 信号，Timer2 计数器(TMR2H:TMR2L)中的计数值中的被分别锁存到 Timer2 重载寄存器(TMR2RLH:TMR2RLL)中，并将 TF2H 中断标志置'1'，产生 Timer2 中断（如果中断使能）。该模式用于通过计算两次 SOF 之间的计数差值来校准内部高频振荡器。

3.8.2.2.4 低频振荡器下降沿捕捉模式

当 T2CE=1，T2CSS=1 时，定时器工作在低频振荡器下降沿帧捕捉模式，该模式可以用来校准内部低频振荡器。计数方式（以及中断产生方式）同 USB 起始帧捕捉模式。

3.8.2.3相关寄存器

3.8.2.3.1 地址映射

寄存器	地址	初始值	说明
TMR2CN	0xC8	0x00	Timer 2 控制寄存器
TMR2L	0xC9	0x 00	Timer2 低字节寄存器
TMR2H	0xCA	0x 00	Timer2 高字节寄存器
TMR2RLL	0xCB	0x 00	Timer2 重载寄存器低字节
TMR2RLH	0xCC	0x 00	Timer2 重载寄存器高字节
TMR2RCK	0xCD	0x 00	Timer2 时钟设置寄存器

3.8.2.3.2 寄存器描述

3.8.2.3.2.1 TMR2CN(Timer2)控制寄存器

位	名称	读写	描述
7	**TF2H**	R/W	非捕捉模式下（T2CE=0）为高字节溢出中断标志位： 1 =定时器 2 高字节发生溢出； 0 = 未检测到定时器 2 高字节溢出；

			捕捉模式下（T2CE=1）为捕捉标志位： 1 =捕捉到捕捉源信号（捕捉源 SOF/LFO 由 T2CSS 决定）； 0 =未捕捉到捕捉源信号。 注：该标志位无法硬件自动清 0，必须软件清 0； 如果清零 TR2 位，该位将自动清零
6	**TF2L**	R/W	定时器 2 低字节溢出中断标志位 1 = 定时器 2 低字节发生溢出 0 = 未检测到定时器 2 低字节溢出（不能硬件自动清 0，必须软件清 0） 注：如果清零 TR2 位，该位将自动清零
5	**TF2LEN**	R/W	定时器 2 低字节中断允许位 1 = 允许定时器 2 低字节中断 0 = 禁止定时器 2 低字节中断
4	**T2CE**	R/W	定时器 2 捕捉允许位 1 = 允许捕捉 0 = 禁止捕捉
3	**T2SPLIT**	R/W	定时器 2 双 8 位方式允许位 1 = 定时器 2 工作在双 8 位自动重装载定时器方式 0 = 定时器 2 工作在 16 位自动重装载定时器方式
2	**TR2**	R/W	定时器 2 运行控制 1 = 允许定时器 2 0 = 禁止定时器 2
1	**T2CSS**	R/W	定时器 2 捕捉源选择 1 = 捕捉源为低频振荡器下降沿 LFO 0 = 捕捉源为 UDB SOF 事件 注：T2CE=1 时该位有效
0	**CAPF**	R/W	定时器 2 捕捉标志位 1 = 捕捉到外部事件 0 = 未捕捉到外部事件（不能硬件自动清 0，必须软件清 0） 注：如果清零 TR2 位，该位将自动清零

			注：T2CE=1 时该位有效

3.8.2.3.2.2 TMR2RLL(Timer2)重载寄存器低字节

位	名称	读写	描述
7-0	**TMR2RLL**	R/W	TMR2RLH 是 16 位定时器 2 重载寄存器的低字节

3.8.2.3.2.3 TMR2RLH(Timer2)重载寄存器高字节

位	名称	读写	描述
7-0	**TMR2RLH**	R/W	TMR2RLH 是 16 位定时器 2 重载寄存器的高字节

3.8.2.3.2.4 TMR2L(Timer2)低字节寄存器

位	名称	读写	描述
7-0	**TMR2L**	R/W	TMR2L 是 16 位定时器 2 的低字节

3.8.2.3.2.5 TMR2H(Timer2)高字节寄存器

位	名称	读写	描述
7-0	**TMR2H**	R/W	TMR2H 是 16 位定时器 2 的高字节

3.8.2.3.2.6 TMR2RCK(Timer2)时钟设置寄存器

位	名称	读写	描述
7-6	保留	R	保留
5	**CKCON_T2H**	R/W	定时器 2 高八位定时器时钟分频控制： 1：选择原始时钟； 0：选择分频时钟。
4	**CKCON_T2L**	R/W	定时器 2 低八位定时器时钟分频控制： 1：选择原始时钟； 0：选择分频时钟。
3-2	**T2CKSEL**	R/W	定时器 2 时钟选择位： 00：选择 12MHz 时钟； 01：选择 48MHz 时钟； 10：选择 500KHz 时钟； 11：选择外部晶振时钟。

1-0	**T2DIV**	R/W	定时器 2 时钟分频选择位： 00：不分频； 01：2 分频； 10：4 分频； 11：8 分频。

3.8.3 看门狗定时器（WDT）

3.8.3.1概述

看门狗定时器是系统定时器，重要用于将系统从意外事件(程序意外跑飞等情况)中恢复。WDT 计数期间，如果软件未在超时周期内清除 WDT 计数器，WDT 将发生溢出。此时：

如果 WDT 系统复位功能使能，将直接产生系统复位；

如果 WDT 系统复位功能关闭，WDT 中断使能，则将产生 WDT 中断；

WDT 还可用于将系统从 IDLE 或 STOP 模式下唤醒（WDT 复位开启则进行复位唤醒，WDT 复位关闭则直接唤醒）。

3.8.3.2特征

- 32k 时钟源
- 多种工作模式
- 可配置的典型超时周期为 4ms 到 128s
- IDLE 或 STOP 模式下仍可工作

3.8.3.3功能描述

3.8.3.3.1 WDT 工作模式

✓ WDT 始终使能

WDTMODE 为 11 时，WDT 始终使能

✓ WDT 空闲睡眠和挂起时禁止

WDTMODE 为 10 时，WDT 在工作模式下（STOP=0 & IDLE=0）使能，其余模式下无效。

✓ WDT 由软件控制

WDTMODE 为 01 时，WDT 使能由寄存器位 SWDTEN 控制，是否使能由软件决定。

✓ WDT 禁止工作

WDTMODE 为 00 时，WDT 禁止工作

3.8.3.3.2 WDT 清零

发生以下任一情况是时，WDT 清零：

- 复位；
- WDTCLR (WDTCTL:bit1)写 1；
- 系统进行模式切换（正常工作模式与 IDLE 或 STOP 模式之间切换）
- WDT 进行模式配置：WDT 模式、预分频、使能控制（也就是对 WDTCON 寄存器进行写操作时）

3.8.3.3.3 系统模式切换时 WDT 状态说明

芯片进入 IDLE 或 STOP 模式时，WDT 清零后重新计数（WDT 使能开启情况下）。

芯片从 IDLE 模式返回工作模式时，WDT 再次清零后重新计数（WDT 使能开启情况下）。

唤醒源将芯片从 STOP 模式唤醒时，WDT 将在唤醒源发出唤醒信号之后一直保持零状态（时钟起振过程中），直至系统进入工作模式才重新开始计数（WDT 使能开启情况下）。

3.8.3.4相关寄存器

3.8.3.4.1 地址映射

寄存器	地址	初始值	描述
WDTCON	0xB7	0x49	WDT 的控制寄存器
WDTCTL	0xBF	0x00	WDT 的 clr 指定寄存器

3.8.3.4.2 寄存器描述

3.8.3.4.2.1WDTCON：WDT 控制寄存器

位	名字	读写	描述
7	保留		
6:5	WDTMODE	RW	WDT 模式控制位 11 = WDT 始终使能 10 = WDT 工作模式下使能，IDLE 或 STOP 模式下禁止 01 = WDT 使能由寄存器位 SWDTEN 控制 00 = WDT 禁止工作
4	SWDTEN	RW	WDT 软件使能控制位 WDTMODE 为 01 1 = 使能 WDT 0 = 禁止 WDT
3:0	WDTPS	RW	WDT 周期选择控制位 0000 = 1:32 (典型间隔 4ms) 0001 = 1:64 (典型间隔 8ms) 0010 = 1:128 (典型间隔 16ms) 0011 = 1:256 (典型间隔 32ms) 0100 = 1:512 (典型间隔 64ms) 0101 = 1:1024 (典型间隔 128ms) 0110 = 1:2048 (典型间隔 256ms) 0111 = 1:4096 (典型间隔 512ms) 1000 = 1:8192 (典型间隔 1s) 1001 = 1:16384 (典型间隔 2s) 1010 = 1:32768 (典型间隔 4s) 1011 = 1:65536 (典型间隔 8s) 1100 = 1:131072 (典型间隔 16s) 1101 = 1:262144 (典型间隔 32s) 1110 = 1:524288 (典型间隔 64s) 1111 = 1:1048576 (典型间隔 128s)

3.8.3.4.2.2WDTCTL：WDT 状态及清除寄存器

位	名字	读写	描述
7	WDTIF	RW	WDT 溢出中断标志 写： 1 = 已发生 WDT 计数溢出 0 = 未发生 WDT 计数溢出 清零：软件向该寄存器位写 0 来清除溢出标志，硬件不会自动清零
6:1	保留		
0	WDTCLR	RW	WDT 清零指令控制位 1 = 清零 wdt 0 = 无操作

3.9 PWM

3.9.1 概述

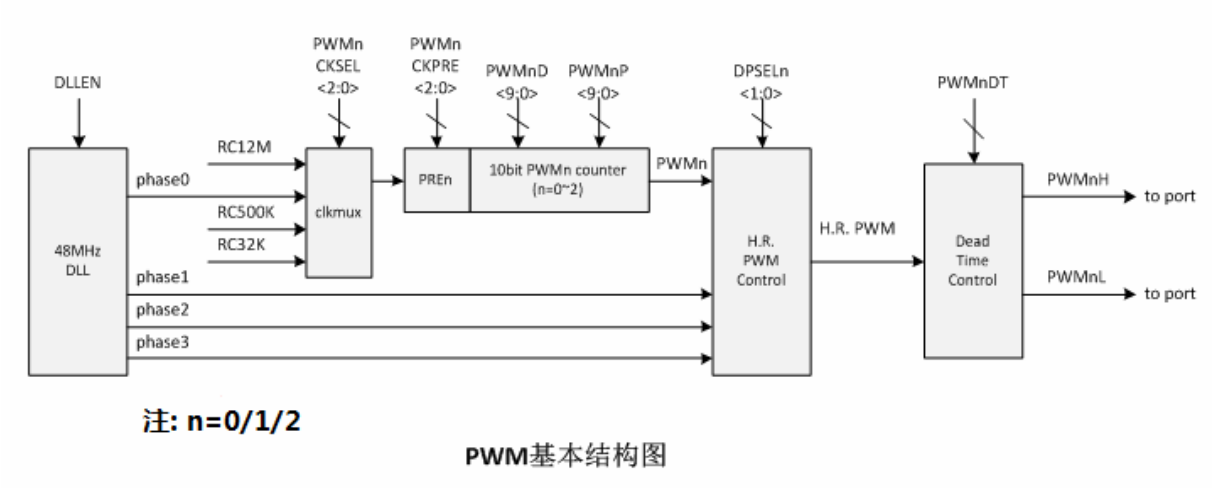
SG8F7581 内置 3 个 10 位 PWM 脉宽调制器——PWM0/PWM1/PWM2。其中每个 PWM 有 4 路计数时钟源可选。各 PWM 占空比/周期在工作期间独立可调；各 PWM 模块均含两路输出引脚，输出极性分别可调，可实现互补输出功能。

高精度调节功能——SG8F7581 为 PWM 的 48MHz 时钟源提供有 DLL 模块。PWM 工作在 48MHz 时钟源时，软件可通过配置 DLL 控制器，实现对 PWM 更高精度的占空比调节能力。

PWM 互补输出/死区时间——SG8F7581 各 PWM 都可通过两个引脚输出，输出极性可调，可实现同向或互补式输出；另外互补输出信号之间可插入死区时间。

四线 PWM 模式——SG8F7581 为 PWM0 提供四线输出模式，此模式输出可用于全桥开关控制。

3.9.2 功能描述



PWM基本结构图

3.9.2.1 PWM 时钟源

SG8F7581 为 PWM0/1/2 提供了 4 路可选时钟源：

- 12MHz 高频 RC
- 48MHz DLL 时钟（4phase4 路时钟）
- 500K 低频 RC
- 32K 低频 RC

软件可通过配置 SFR:PWM0PRE/PWM1PRE/PWM2PRE 来选择各 PWM 的工作时钟。其中，选择 48MHz DLL 时钟需要提前开启 DLLEN(PWMHR bit7)并等待 DLL 输出稳定标志（PWMHR bit6）。

注：如果 DLLEN 开启且 PWM 选择 48MHz DLL 时钟源，可以使用 PWM 的高精度调节/死区时间功能。在其它时钟源下 PWM 高精度调节、死区产生电路将无法正常工作(两路同向/互补输出功能仍有效)。

另外，PWM0/1/2 分别有前置预分频器，为 PWM 计数器提供 2~16 分频计数时钟。

3.9.2.2 PWM 周期、占空比调节

PWM0/1/2 输出信号的周期、占空比由各自对应的周期、占空比控制寄存器控制。

PWM 输出信号周期，占空比计算公式如下：

PWMn(n=0~2)周期计算公式：

Tperiod= (<PWMnPH:PWMnPL> + 1) * (PWMnCKPRE+1) * Tpwmclk

(其中 Tpwmclk 为当前 PWM 计数时钟源周期，由 SFR:PWMnPRE 寄存器中 PWMnCKSEL 选择)。

PWMn(n=0~2)占空比计算公式（不含高精度调节位）：

Duty = ((<PWMnDH:PWMnDL> + 1) / (<PWMnPH:PWMnPL> + 1)) *100%

PWM 使能开启后，PWM 将按照使能开启前软件写入的占空比/周期值（默认为全 0）对外输出 PWM 信号。

PWM 工作期间，软件可以随时调整占空比/周期——先向相应寄存器写入新的占空比/周期值，后将 SFR:PWMCN 中 PWMnRL(n=0~2)位置 1；硬件会在当前 PWM 周期结束后，重新载入新写入的周期/占空比值，同时 PWMnRL 位自动清零。如果未将 PWMnRL(n=0~2)位置 1，PWM 将仍旧按照前一次配置的周期/占空比运行。

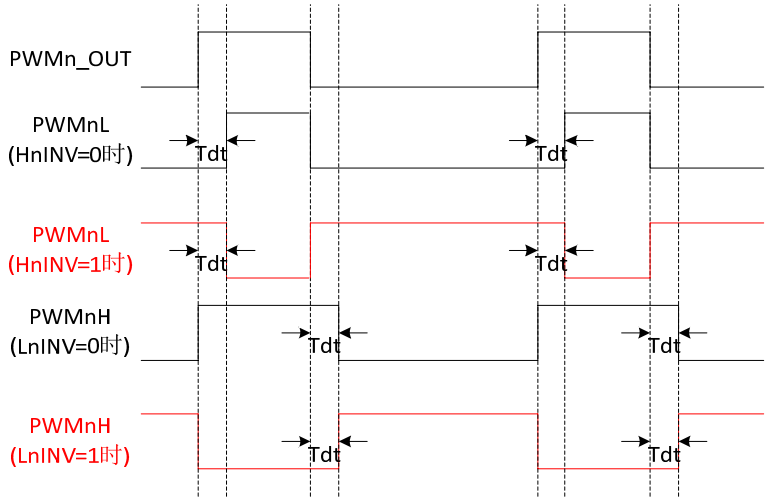
3.9.2.3 PWM 输出极性/死区时间控制

PWM0~PWM2 每个 PWM 分配有 2 路输出引脚，可通过 SFR:PORTMUX 开启相应引脚的端口复用功能。

注意开启 SFR:PORTMUX 中 PWM 输出引脚复用后，如果相应的 PWM 使能位（PWMCN bit5~bit3）关闭，该引脚方向/输出数据/上下拉使能等仍受 GPIO 相关控制位控制；而相应的 PWM 使能位（PWMCN bit5~bit3）开启后，该引脚将强制输出 PWM 信号。

软件通过 SFR:PWMnDT(n=0~2)，可更改 PORTMUX 输出极性，使两路输出同向或反向互补。

PWM 的两路 PWMnH/PWMnL 输出可用于驱动外部 MOS 管 P 管和 N 管，之间经过死区时间产生电路可插入死区时间，死区时间长度 Tdt 由 PWMnDT(n=0~2)控制。



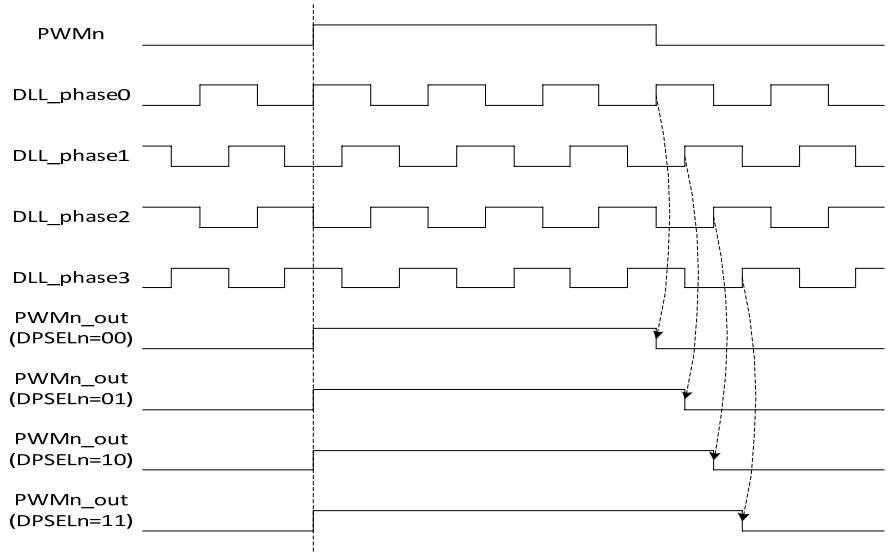
注：图中n=0~2；
Tdt表示死区时间；

PWM互补输出+死区时间示意图

3.9.2.4 高精度（H.R.）PWM

SG8F7581 内置一个 48MHz DLL，以 12MHz RC 时钟为输入时钟源，进行 x4 时钟倍频并输出 4 路不同 phase 的 48MHz 时钟。

当 PWM 选用 DLL 48MHz 时钟作为时钟源时，高精度调节电路（H.R）根据 DPSELn(PWMHR bit5~0,n=0~2)，选择不同的 phase 信号对原 PWM 输出进行更高精度的占空比调整。如下图所示：

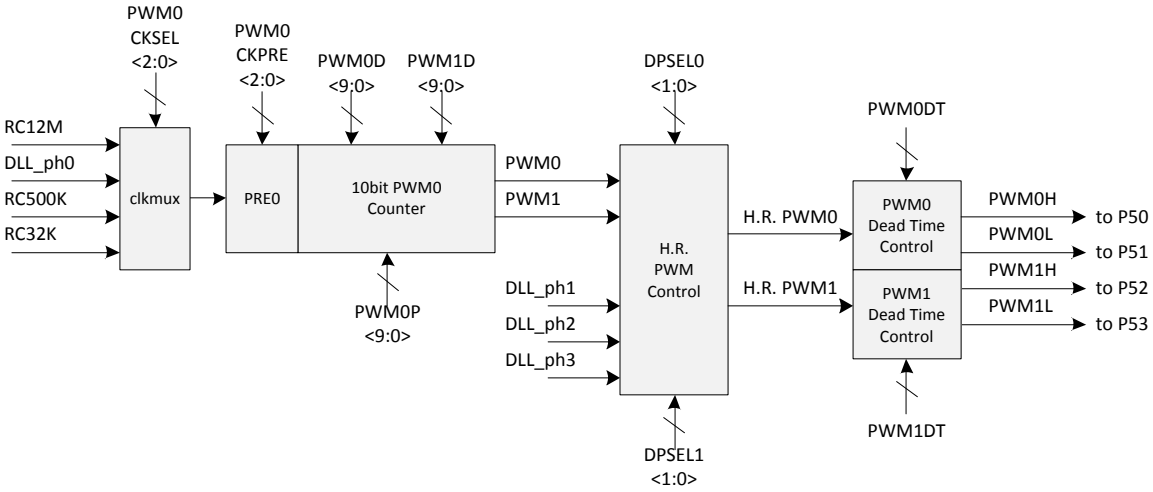


H.R PWM输出时序图

经 H.R.电路调整后，最终 PWM 输出信号占空比计算公式如下：

DUTYn = ((<PWMnDH:PWMnDL>+(DPSELn/4) + 1) / (<PWMnPH:PWMnPL> + 1)) *100%
(n=0~2)

3.9.2.5 PWM0 四线模式



PWM4线模式结构图

SG8F7581 为 PWM0 提供四线输出模式，用于全桥开关等的 4 线控制方案。

- 软件通过 L4MODE（SFR:PWMCN bit7）控制位开启 4 线输出模式。

原 PWM0/PWM1 的 P50~P53 四个输出引脚分别用作 4 线模式下 4 路 PWM 信号输出。

- **使能控制：**

四线模式下，PWM0EN 同时控制 PWM 四线输出信号。当 PWM0EN=1 时，P50~P53 同时对外输出 PWM 信号。（需要提前开启各引脚端口复用）

- **周期控制：**

四线模式下，PWM 四线输出周期相同，受 SFR:PWM0P(PWM0PH:PWM0PL)控制

- **占空比控制：**

PWM 四线输出中，

PWM0H/PWM0L 输出信号占空比受 SFR:PWM0D(PWM0DH:PWM0DL)控制；

PWM1H/PWM1L 输出信号占空比受 SFR:PWM1D(PWM1DH:PWM1DL)控制；

注：DLLEN 开启后，PWM0H/PWM0L 占空比高精度调节受 DPSEL0（PWMHR bit1~0）控制；PWM1H/PWM1L 占空比高精度调节受 DPSEL1（PWMHR bit3~2）控制；

- **死区时间/正反向输出：**

四线模式下，P50~P53 四个引脚的死区时间控制和正/反向输出功能与原 PWMn 控制方式相同，PWM0H/PWM0L 由 PWM0DT 控制死区时间和输出极性；PWM1H/PWM1L 由 PWM1DT 控制死区时间和输出极性；

3.9.3 相关寄存器

3.9.3.1地址映射

寄存器	地址	初始值	说明
PWMCN	0xD1	0x00	PWM0/1/2 控制寄存器
PWM0DL	0xD2	0x00	PWM0 占空比控制低字节
PWM0DH	0xD3	0x00	PWM0 占空比控制高字节
PWM1DL	0xD4	0x00	PWM1 占空比控制低字节
PWM1DH	0xD5	0x00	PWM1 占空比控制高字节
PWM2DL	0xD6	0x00	PWM2 占空比控制低字节
PWM2DH	0xD7	0x00	PWM2 占空比控制高字节
PWMHR	0xD9	0x00	PWM0/1/2 高精度调节控制寄存器
PWM0PL	0xDA	0x00	PWM0 周期控制低字节
PWM0PH	0xDB	0x00	PWM0 周期控制高字节
PWM1PL	0xDC	0x00	PWM1 周期控制低字节
PWM1PH	0xDD	0x00	PWM1 周期控制高字节
PWM2PL	0xDE	0x00	PWM2 周期控制低字节
PWM2PH	0xDF	0x00	PWM2 周期控制高字节
PWM0DT	0xE2	0x00	PWM0 死区控制寄存器
PWM1DT	0xE3	0x00	PWM1 死区控制寄存器
PWM2DT	0xE4	0x00	PWM2 死区控制寄存器
PWM0PRE	0xE5	0x00	PWM0 时钟控制寄存器
PWM1PRE	0xE6	0x00	PWM1 时钟控制寄存器
PWM2PRE	0xE7	0x00	PWM2 时钟控制寄存器

3.9.3.2寄存器描述

3.9.3.2.1 PWMCN(PWM)控制寄存器

位	名称	读写	描述
7	L4MODE	R/W	4 线模式控制位： 1 = 开启 4 线模式 0 = 关闭 4 线模式
6	保留	R	-
5	PWM2EN	R/W	PWM2 使能位 1=使能 PWM 0=禁止 PWM
4	PWM1EN	R/W	PWM1 使能位 1=使能 PWM 0=禁止 PWM
3	PWM0EN	R/W	PWM0 使能位 1=使能 PWM 0=禁止 PWM
2	PWM2RL	R/W	PWM2 占空比/周期更新控制位： 1=当前 PWM2 计数周期结束后，更新其占空比/周期设定至软件最后一次写入值。 0=PWM2 占空比/周期仍按照上一次更新时软件设定值工作。 注：软件写'1'，更新成功后自动清零。
1	PWM1RL	R/W	PWM1 占空比/周期更新控制位： 1=当前 PWM1 计数周期结束后，更新其占空比/周期设定至软件最后一次写入值。 0=PWM1 占空比/周期仍按照上一次更新时软件设定值工作 注：软件写'1'，更新成功后自动清零。
0	PWM0RL	R/W	PWM0 占空比/周期更新控制位：

			1=当前 PWM0 计数周期结束后，更新其占空比/周期设定至软件最后一次写入值。 0=PWM0 占空比/周期仍按照上一次更新时软件设定值工作 注：软件写'1'，更新成功后自动清零。

*注：即更改 PWM 的周期与占空比后需向 PWMCN[2:0]写 1,PWM 将在下一周期输出新的占空比与周期信号，然后 PWMCN[2:0]自动清零。若写入 0，则 PWM 仍按照原周期与占空比输出。

3.9.3.2.2 PWM0PL(PWM0)周期控制低字节

位	名称	读写	描述
7:0	PWM0PL	R/W	PWM0 周期控制低 8 位

3.9.3.2.3 PWM0PH(PWM0)周期控制高字节

位	名称	读写	描述
7:2	保留	R	-
1:0	PWM0PH	R/W	PWM0 周期控制高 2 位

3.9.3.2.4 PWM1PL(PWM1)周期控制低字节

位	名称	读写	描述
7:0	PWM1PL	R/W	PWM1 周期控制低 8 位

3.9.3.2.5 PWM1PH(PWM1)周期控制高字节

位	名称	读写	描述
7:2	保留	R	-
1:0	PWM1PH	R/W	PWM1 周期控制高 2 位

3.9.3.2.6 PWM2PL(PWM2)周期控制低字节

位	名称	读写	描述
7:0	PWM2PL	R/W	PWM2 周期控制低 8 位

3.9.3.2.7 PWM2PH(PWM2)周期控制高字节

位	名称	读写	描述
7:2	保留	R	-
1:0	PWM2PH	R/W	PWM2 周期控制高 2 位

3.9.3.2.8 PWM0DL(PWM0)占空比低字节

位	名称	读写	描述
7:0	PWM0DL	R/W	PWM0 占空比控制低 8 位

3.9.3.2.9 PWM0DH(PWM0)占空比高字节

位	名称	读写	描述
7:2	保留	R/W	-
1:0	PWM0DH	R/W	PWM0 占空比控制高 2 位

3.9.3.2.10 PWM1DL(PWM1)占空比低字节

位	名称	读写	描述
7:0	PWM1DL	R/W	PWM1 占空比控制低 8 位

3.9.3.2.11 PWM1DH(PWM1)占空比高字节

位	名称	读写	描述
7:2	保留	R/W	-
1:0	PWM1DH	R/W	PWM1 占空比控制高 2 位

3.9.3.2.12 PWM2DL(PWM2)占空比低字节

位	名称	读写	描述
7:0	PWM2DL	R/W	PWM2 占空比控制低 8 位

3.9.3.2.13 PWM2DH(PWM2)占空比高字节

位	名称	读写	描述
7:2	保留	R/W	-
1:0	PWM2DH	R/W	PWM2 占空比控制高 2 位

3.9.3.2.14 PWM0PRE(PWM0)时钟控制寄存器

位	名称	读写	描述
7:6	保留	R	-
5:4	PWM0CKSEL	R/W	PWM0 计数时钟选择位： 00:12MHz 时钟 01:48MHz 时钟 10:500KHz 时钟 11:32KHz 时钟 注：选择 48MHz 时钟时，需要先使能 DLLEN——PWMHR bit7，并等待 DLL 工作稳定。
3-0	PWM0CKPRE	R/W	PWM0 计数时钟预分频选择位： 0000：不分频 0001：2 分频 …… 1111: 16 分频

3.9.3.2.15 PWM1PRE(PWM1)时钟控制寄存器

位	名称	读写	描述
7:6	保留	R	-
5:4	PWM1CKSEL	R/W	PWM1 计数时钟选择位： 00:12MHz 时钟 01:48MHz 时钟 10:500KHz 时钟 11:32KHz 时钟 注：选择 48MHz 时钟时，需要先使能 DLLEN——PWMHR bit7，并等待 DLL 工作稳定。
3:0	PWM1CKPRE	R/W	PWM1 计数时钟预分频选择位： 0000：不分频 0001：2 分频 ……. 1111: 16 分频

3.9.3.2.16 PWM2PRE(PWM2)时钟控制寄存器

位	名称	读写	描述
7:6	保留	R	-
5:4	PWM2CKSEL	R/W	PWM2 计数时钟选择位： 00:12MHz 时钟 01:48MHz 时钟 10:500KHz 时钟 11:32KHz 时钟 注：选择 48MHz 时钟时，需要先使能 DLLEN——PWMHR bit7，并等待 DLL 工作稳定。
3-0	PWM2CKPRE	R/W	PWM2 计数时钟预分频选择位： 0000：不分频 0001：2 分频

			…… 1111: 16 分频

3.9.3.2.17 PWMHR(PWM0/1/2)高精度调节控制寄存器

位	名称	读写	描述
7	DLLEN	R/W	DLL 使能控制/状态位 写操作： 1 = DLL 开启 0 = DLL 关闭 读操作： 1 = DLL 开启后工作稳定； 0 = DLL 未开启或开启后工作未稳定。
6	保留	R	-
5:4	DPSEL2	R/W	PWM2 输出的 DLL Phase 选择位： 00 = 选择 phase0 01 = 选择 phase1 10 = 选择 phase2 11 = 选择 phase3
3:2	DPSEL1	R/W	PWM1 输出的 DLL Phase 选择位： 00 = 选择 phase0 01 = 选择 phase1 10 = 选择 phase2 11 = 选择 phase3
1:0	DPSEL0	R/W	PWM0 输出的 DLL Phase 选择位： 00 = 选择 phase0 01 = 选择 phase1 10 = 选择 phase2 11 = 选择 phase3

3.9.3.2.18 PWM0DT(PWM0)输出控制寄存器

位	名称	读写	描述
7	H0INV	R/W	PWM0H(P50)输出极性控制位 1 = PWM0H 输出反向； 0 = PWM0H 输出正向；
6	L0INV	R/W	PWM0L(P51)输出极性控制位 1 = PWM0L 输出反向； 0 = PWM0L 输出正向；
5	保留		
4	DT0EN	R/W	PWM0H/PWM0L 输出死区控制位 1 = 开启死区功能 0 = 关闭死区功能
3:0	DT0SEL	R/W	PWM0H/PWM0L 输出死区时间选择位 死区时间计算公式：T_{DT} = (DT0SEL<3:0>) / 48MHz;

3.9.3.2.19 PWM1DT(PWM1)输出控制寄存器

位	名称	读写	描述
7	H1INV	R/W	PWM1H(P52)输出极性控制位 1 = PWM1H 输出反向； 0 = PWM1H 输出正向；
6	L1INV	R/W	PWM1L(P53)输出极性控制位 1 = PWM1L 输出反向； 0 = PWM1L 输出正向；
5	保留		
4	DT1EN	R/W	PWM1H/PWM1L 输出死区控制位 1 = 开启死区功能 0 = 关闭死区功能
3:0	DT1SEL	R/W	PWM1H/PWM1L 输出死区时间选择位 死区时间计算公式：T_{DT} = (DT0SEL<3:0>) / 48MHz;

3.9.3.2.20 PWM2DT(PWM2)输出控制寄存器

位	名称	读写	描述
7	H2INV	R/W	PWM2H(P56)输出极性控制位 1 = PWM2H 输出反向； 0 = PWM2H 输出正向；
6	L2INV	R/W	PWM2L(P55)输出极性控制位 1 = PWM2L 输出反向； 0 = PWM2L 输出正向；
5	保留	R	-
4	DT2EN	R/W	PWM2H/PWM2L 输出死区控制位 1 = 开启死区功能 0 = 关闭死区功能
3:0	DT2SEL	R/W	PWM2H/PWM2L 输出死区时间选择位 死区时间计算公式：T_{DT} = (DT0SEL<3:0>) / 48MHz;

3.10 USB 控制器

3.10.1 概述

SG8F7581 兼容全速和低速两种 USB 通信协议，具有 3 个可双向传输的端点，分别为端点 0，端点 1 和端点 2；其中端点 0 数据深度 8byte，端点 1 和 2 数据深度为 64byte。

3.10.2 功能描述

3.10.2.1 UDC-DMA 方式说明

SG8F7581 UDC 通信相关的数据都存储于扩展数据存储器（XRAM）中，UDC 需要通过 DMA 方式与 XRAM 进行数据传输。

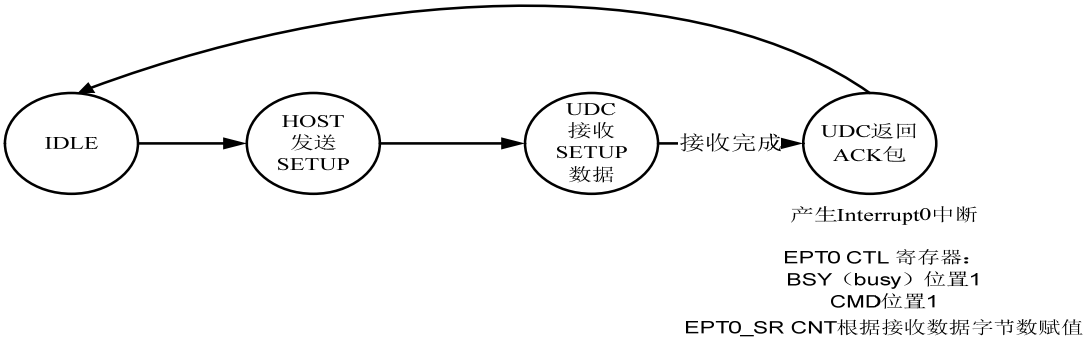
DMA 控制器为 UDC 设置了 3 个专用数据通道，分别用于 UDC EPT0~EPT2 三个端点的数据传输。UDC 开启之前，软件需要通过配置 DMA 相关寄存器，使能并设定 3 个数据通道的配置参数（如数据深度 DMASZ，数据存储基地址 DMABAL/DMABAH 等），DMA 详细配置方式请参见“DMA 控制器”。

USB 与主机通信期间，软件需通过 MOVX 指令读写 XRAM 相应存储区（DMA 控制器设定的 EPT0~EPT2 三个端点数据存储区）数据，和 UDC 控制器进行数据交互，数据交互方式参见“UDC 通信举例”。

3.10.2.2 UDC 通信流程说明

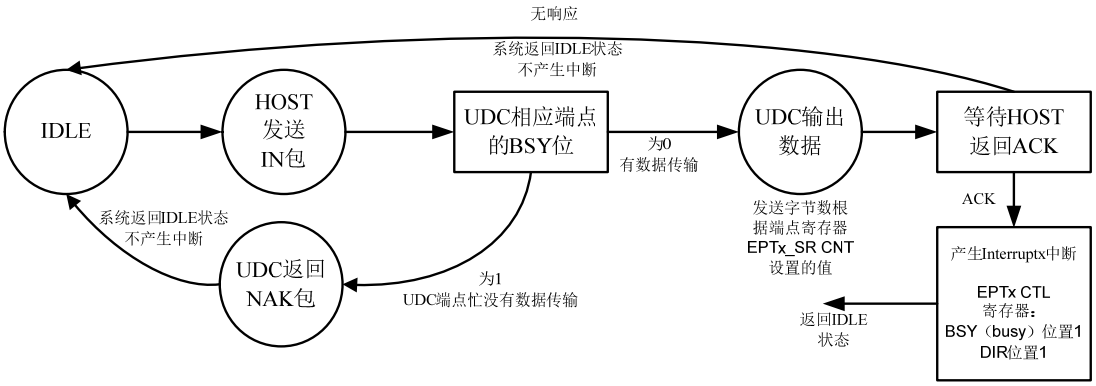
UDC 与 HOST 通讯只通过 3 种通讯包，分别为：SETUP 包，中断 IN 包和中断 OUT 包。下面对各包的 UDC 的响应及相应操作作详细说明。

3.10.2.2.1 SETUP 包处理说明



端点 0 接收和处理 SETUP 包，当 HOST 发送 SETUP 包时，端点 0 状态寄存器 BSY（busy）位无论为何值，端点 0 都会正常接收 SETUP 包并产生中断。

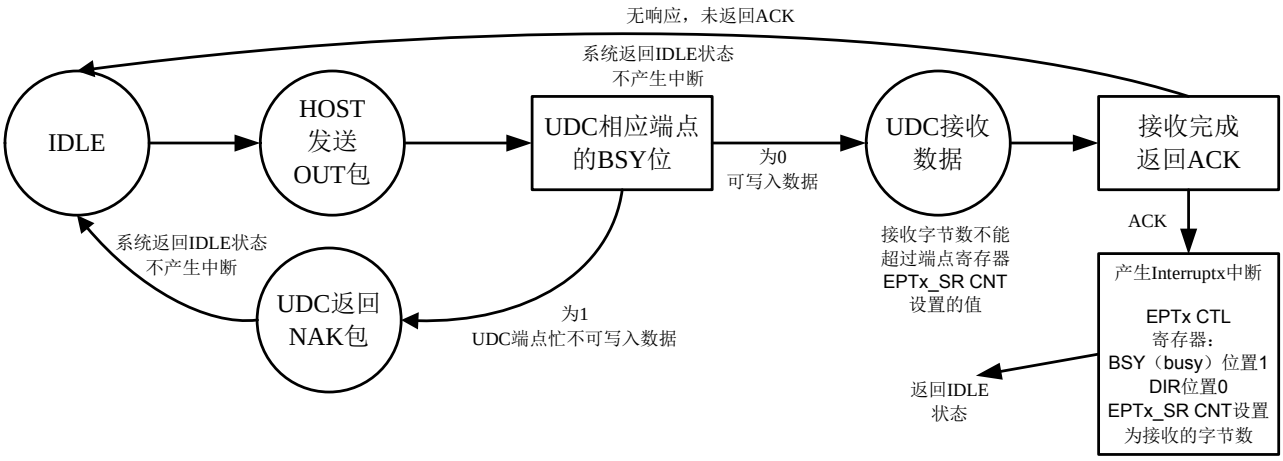
3.10.2.2.2 中断传输 IN 包处理说明



一旦相应端点的 BSY 位置为 0，UDC 在 IN 包到来时，立即通过 DMA 读取并发送 XRAM 中的数据，不论通过 DMA 控制器取出的数据是否有效。

所以 MCU 响应 IN 包时，应该先将要发送的数据通过 DMA 写入 XRAM，然后再将相应的 BSY 位置 0。

3.10.2.2.3 中断传输 OUT 包处理说明



UDC 接收完成数据后，如果没有返回 ACK，表示数据接收中产生异常，例如：CRC 校验错误，DATA 的 PID 同步错误，或者接收数据超出寄存器 EPTx_SR CNT 设置的值。此时系统将直接返回 IDLE，不会产生 Interrupx 中断,会产生 Interrup error 中断。

3.10.2.3 UDC 通信举例

以上电后枚举传输的第一个数据包为例说明如何响应和操作 UDC。

通过查看协议分析仪上的数据，看到主机在 RESET 后，会向设备下发 GET_DESCRIPTOR 的标准设备请求。

Transfer	L	Control	ADDR	ENDP	bRequest	wValue	wIndex	Descriptors	Time
18	S	GET	2	0	GET_DESCRIPTOR	DEVICE type	0x0000	DEVICE descriptor	0 ns

Transaction	L	SETUP	ADDR	ENDP	D	T	R	bRequest	wValue	wIndex	wLength	ACK
519	S	0xB4	2	0	D->H	S	D	GET_DESCRIPTOR	DEVICE type	0x0000	18	0x4B

Packet #	L	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle
5170	S	00000001	0xB4	2	0	0x15	3.00	4

Packet #	L	Sync	DATA0	DATA	CRC16	EOP	Idle
5171	S	00000001	0xC3	80 06 00 01 00 00 12 00	0x072F	3.00	5

Packet #	L	Sync	ACK	EOP	Idle
5172	S	00000001	0x4B	3.00	1332

Transaction	L	IN	ADDR	ENDP	T	DATA	ACK
520	S	0x96	2	0	1	12 01 10 01 00 00 00 08	0x4B

Transaction	L	IN	ADDR	ENDP	T	DATA	ACK
521	S	0x96	2	0	0	4F 1C 34 00 10 01 01 02	0x4B

Transaction	L	IN	ADDR	ENDP	T	DATA	ACK
522	S	0x96	2	0	1	00 01	0x4B

Transaction	L	OUT	ADDR	ENDP	T	DATA	ACK
523	S	0x87	2	0	1		0x4B

图 3-2-1 SETUP 包 GET_DESCRIPTOR 请求

设备硬件接收到 SETUP 包后，将数据包中的 8btye 数据通过 DMA 装入 XRAM 指定存储区，完成判断方向等操作后返回 ACK 并产生 EPT0 中断，等待主机确认并发出 IN 包。软件在硬件返回

ACK 产生中断后，进入中断服务程序，处理数据包，并通过 MOVX 指令将需要发送的数据写入 XRAM 相关存储区。设备在接收到主机的 IN 包后，通过 DMA 读取并发送 XRAM 之前写入的数据。

注释：1）描述符的第一个 byte 中包含设备将要发送数据的长度

2）主机不接收超过 SETUP 长度要求的数据，接收要求长度的数据后，会下发 OUT 包

3）软件中用于循环计数的变量值等于描述表中描述的将要发送的字节长度

3.10.3 相关寄存器

3.10.3.1 地址映射

寄存器	地址	初始值	说明
USBCON	0xC1	0x00	USB 控制寄存器
EPT0CTL	0xC2	0x88	端点 0 控制寄存器
EPT1CTL	0xC3	0x80	端点 1 控制寄存器
EPT2CTL	0xC4	0x80	端点 2 控制寄存器
USBADDR	0xC5	0x00	USB 地址寄存器
USBSTA	0xC6	0x00	USB 数据同步控制寄存器
USBINTE	0xCE	0x00	USB 中断允许寄存器
USBINTF	0xCF	0x00	USB 中断状态寄存器

3.10.3.2 功能描述

3.10.3.2.1 USBCON(USB)控制寄存器

位	名称	读写	说明
7	PHYEN	R/W	USB PHY 挂起控制 1 = USB PHY 使能(USB 模式) 0 = USB PHY 禁止（GPIO 或 PS2 模式）
6	USB_UP	R/W	USB 上拉控制

			1 = 使能 USB 上拉电阻 0 = 禁止 USB 上拉电阻
5	TX_EN	R/W	USB 输出使能 1：使能 PHY 输出 0：禁止 PHY 输出
4	TX_DATA	R/W	USB 输出数据 1：不唤醒主机 0：唤醒主机
3	HSPEED	R/W	USB 全速使能 1：使能 USB 工作在全速模式下 0：使能 USB 工作在低速模式下
2	PS2OV	R/W	PS2 自动判断标志位 1：P70/P71 外接 HOST PS2 接口 0：P70/P71 没有外接 HOST PS2 接口
1	USBTRIM	R/W	UDC 自动调节功能控制位 1：开启 UDC 时钟自动调节功能(调节 RC12M 输出频率) 0：关闭 UDC 时钟自动调节功能
0	PS2UP	R/W	PS2 上拉控制位 1：开启 P70/P71 PS2 上拉功能 0：关闭 P70/P71 PS2 上拉功能

3.10.3.2.2 EPT0CTL 端点 0 控制寄存器

位	名称	读取	说明	缺省值
7	0BSY	R/W	端点 0 忙 busy, 当 0BSY = 1 时表示忙，端点 0 对各 PID 包响应： SETUP (h) => ACK (d) OUT (h) => NACK (d) IN (h) => NACK (d)； 当 0BSY = 0 时，端点 0 对各 PID 包响应： SETUP (h) => ACK (d)	1’b1

			OUT (h) = > DATA (h) = > ACK (d) IN (h) => DATA (d) => ACK (h)。 （注：d = device，h = host）	
6	0DIR	R/W	总线方向标志位 0：Next package Host => Device,； 1：Next package Device => Host	1’b0
5	CMD	R/W	Setup 标志位； 1：current package 是 Setup 包	1’b0
4	保留	R	—	1’b0
3:0	SRCNT	R/W	端点 0 接收与发送计数器 SR CNT 字节数 0000: 8 字节 0001: 1 字节 0010: 2 字节 …….. 0111: 7 字节 1000: 8 字节 注：SRCNT 设置为 0 后，端点 0 接收和发送数据为 8 字节。	4’h8

3.10.3.2.3 EPT1CTL 端点 1 控制寄存器

位	名称	读取	说明	缺省值
7	1BSY	R/W	端点 1 忙 1BSY = 1 表示忙，端点 1 对各 PID 包响应： IN (h) => NACK (d) OUT (h) => NACK (d) 1BSY = 0 时，端点 1 对各 PID 包响应： IN (h) => DATA (d) => ACK (h) OUT (h) = > DATA (h) = > ACK (d)	1’b1

			（注：d = device，h = host）	
6	1DIR	R/W	总线方向标志位 0： Next package Host => Device,； 1： Next package Device => Host	1’b0
5:0	SRCNT	R/W	端点 1 接收与发送计数器 SEND CNT 字节数 000000: 64 字节 000001: 1 字节 000010: 2 字节 …….. 011111: 31 字节 100000: 32 字节 …….. 111111： 63 字节 注：SRCNT 设置为 0 后，端点 1 接受和发送数据数为 64 字节。	6’h0

3.10.3.2.4 EPT2CTL 端点 2 控制寄存器

位	名称	读取	功说明	缺省值
7	2BSY	R/W	端点 2 忙 2BSY = 1 表示忙，端点 2 对各个 PID 包响应： IN (h) => NACK (d) OUT (h) => NACK (d) 2BSY = 0 时，端点 2 对各个 PID 包响应： IN (h) => DATA (d) => ACK (h) OUT (h) = > DATA (h) = > ACK (d) （注：d = device，h = host）	1’b1
6	2DIR	R/W	总线方向标志位	1’b0

			0： Next package Host => Device,； 1： Next package Device => Host	
5:0	SR CNT	R/W	端点 2 接收与发送计数器 SEND CNT 字节数 000000: 64 字节 000001: 1 字节 000010: 2 字节 …….. 011111: 31 字节 100000: 32 字节 …….. 111111： 63 字节 注：SRCNT 设置为 0 后，端点 2 接受和发送数据 64 字节	6’h0

3.10.3.2.5 USBADDR（USB）地址寄存器

位	名称	读取	功能	缺省值
7	标记	R/W	端点 0 响应主机最后一个 IN 包后置 1； 将等待主机返回 OUT 的 0 包，该位自动清零	1’b0
6:0	设备地址寄存器	R/W	设备地址，当 SetAddress 命令完成时，设定此寄存器。	7’h00

3.10.3.2.6USBSTA USB 数据同步控制寄存器

位	名称	读取	功能	缺省值
7	EPT1 TCLEAR	R/W	端点 1 清除数据同步，下一个数据包 PID 设置为 DATA0	1’b0
6	EPT2	R/W	端点 2 清除数据同步，下一个数据包 PID 设置为 DATA0	1’b0

	TCLEAR			
5	2STL	R/W	设置为 1 时 Endpoint2 进入 Stall 状态； 设置为 0 时 Endpoint2 为正常状态。	1’b0
4	1STL	R/W	设置为 1 时 Endpoint1 进入 Stall 状态； 设置为 0 时 Endpoint1 为正常状态。	1’h0
3	0STL	R/W	设置为 1 时 Endpoint0 进入 Stall 状态； 设置为 0 时 Endpoint0 为正常状态。	1’h0
2	ZPACK2	R/W	为 1 时，UDC 发送一个为 0 的数据包	1’b0
1	ZPACK1	R/W	为 1 时，UDC 发送一个为 0 的数据包	1’b0
0	ZPACK0	R/W	为 1 时，UDC 发送一个为 0 的数据包	1’b0

3.10.3.2.7 USBINTE USB 中断控制寄存器

位	名称	读取	功能	缺省值
7	INT_URST_EN	R/W	USB 复位中断使能位	1’b0
6~5	保留	R	-	2’b0
4	INT_ERR_EN	R/W	USB 传输异常中断使能位 控制 DATA PID Err、DATA Packet Err、 HSHK_ACK_Err 中断	1’b0
3	INT_3MS_EN	R/W	3MS 中断使能位	1’b0
2	INT2_EN	R/W	端点 2 中断使能位	1’b0
1	INT1_EN	R/W	端点 1 中断使能位	1’b0
0	INT0_EN	R/W	端点 0 中断使能位	1’b0

3.10.3.2.8 USBINTF USB 中断状态寄存器

位	名称	读取	功能	缺省值
7	INT_URST_FLAG	R/W	1-- USB 总线产生复位信号 写 1 清零	1’b0
6	DATA PID Err	R/W	1-- 表示数据 PID 同步错误状态中断标志位; 写 1 清零;（属于传输异常中断）	1’b0

5	DATA Packet Err	R/W	1-- 表示数据包接收 bit strip 错误或 CRC16 检测错误中断标志位; 写 1 清零;（属于传输异常中断）	1'b0
4	HSHK_ACK_Err	R/W	1-- 表示接收主机 ACK 包超时或者错误中断标志位，读后需要清除为 0 写 1 清零;（属于传输异常中断）	1'b0
3	INT_3MS_FLAG	R/W	1-- 3 毫秒中断，表示 USB 总线上超过 3MS 没有数据变化。 写 1 清零	1'b0
2	INT2_FLAG	R/W	1-- 端点 2 中断; 写 1 清零	1'b0
1	INT1_FLAG	R/W	1-- 端点 1 中断; 写 1 清零	1'b0
0	INT0_FLAG	R/W	1-- 端点 0 中断; 写 1 清零	1'b0

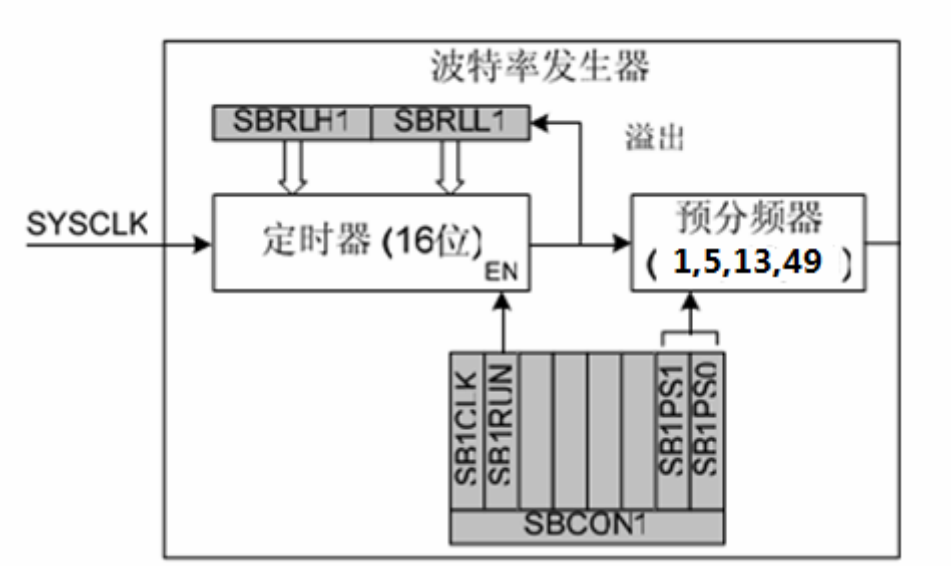
3.11 UART 控制器

3.11.1 概述

UART 是一个异步、全双工串口，它提供多种数据格式选择。UART 包含一个由 16 位定时器和可编程预分频器构成的专用波特率发生器，能产生很宽范围的波特率。有多个时钟源可用于产生标准波特率。数据传输支持 8 位数据长度和 9 位数据长度，第 9 位可作为奇偶校验位或数据位。

3.11.2 功能描述

3.11.2.1 波特率发生器



波特率发生器框图（上图中预分频器连接在定时器前面，先分频系统时钟）

UART 波特率是由一个专用的 16 位定时器产生的。该定时器使用控制器内核时钟（SYSCLK）工作，并有一个预分频器，可选择 1、5、13 或 49 分频。定时器和预分频器选项的组合允许在多种不同的 SYSCLK 频率下都可以有很宽的波特率选择范围。

用三个寄存器（SBCON1、SBRLH1 和 SBRLL1）来配置波特率发生器。UART 波特率发生器控制寄存器使能或禁止波特率发生器，并为定时器选择预分频值。使用 UART 时，波特率发生器必须被使能。寄存器 SBRLH1 和 SBRLL1 保持该专用定时器的 16 位重载值。定时器从重载值开始向上计数，每个时钟加 1。定时器发生溢出（从 0xFFFF 到 0x0000）时立即被重新装载。对于可靠的 UART 操作，建议不要将 UART 波特率配置为大于 SYSCLK/16。UART 的波特率由下列方程决定：

$$波特率 = \frac{SYSCLK}{65536 - (SBRLH:SBRLL)} \times \frac{1}{预分频值}$$

对应标准波特率的定时器设置（使用内部振荡器）

系统时钟	目标波特率（bps）	目标波特率(bps)	波特率误差	振荡器分频系数	SBPS[1:0](预分频位)	SBRLH1:SBRLL1重载值
SYSCLK 12MHz	230400	230769	0.16%	52	11	0xFFCC
	115200	115384	0.16%	104	11	0xFF98
	57600	57692	0.16%	208	11	0xFF30
	28800	28776	0.08%	417	11	0xFE5F
	19200	19200	0.0%	625	11	0xFD8F
	9600	9600	0.0%	1250	11	0xFB1E
	4800	4800	0.0%	2500	11	0xF63C
	2400	2400	0.0%	5000	11	0xEC78
	1200	1200	0.0%	10000	11	0xD8F0
SYSCLK 48MHz	230400	230769	0.16%	208	11	0xFF30
	115200	115107	0.08%	417	11	0xFE5F
	57600	57623	0.04%	833	11	0xFCBF
	28800	28794	0.02%	1667	11	0xF97D
	19200	19200	0.0%	2500	11	0xF63C
	9600	9600	0.0%	5000	11	0xEC78
	4800	4800	0.0%	10000	11	0xD8F0
	2400	2400	0.0%	20000	11	0xB1E0
	1200	1200	0.0%	40000	11	0x63C0

3.11.2.2 8 位 UART

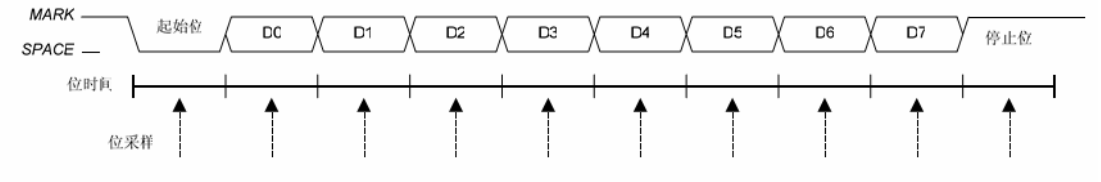
在 8 位 UART 方式，每个数据字节共使用 10 位：一个起始位、8 个数据位（LSB 在先）和一个停止位。数据从 TX 引脚发送，在 RX 引脚接收。在接收时，8 个数据位存入 SBUF。

当软件向 SBUF 寄存器写入一个字节时开始数据发送。在发送结束时（停止位开始）发送中断标志 TI（SCON1.1）被置 1。在接收允许 REN（SCON0.4）被置 1 后，接收可以在任何时刻开始。收到停止位后，如果满足下述条件则数据字节将被装入到接收寄存器 SBUF：RI 必须为逻辑 0；如果 MCE=1，则停止位必须为 1；在发生接收数据溢出的情况下，先接收到的 8 位数据被锁存到 SBUF，而后面的溢出数据被丢弃。

如果这些条件满足，则 8 位数据被存入 SBUF，RI 标志被置位。

如果这些条件不满足，则不装入 SBUF 和 RBX，RI 标志也不会被置 1。如果中断被允许在 TI

或 RI 置位时将产生一个中断。

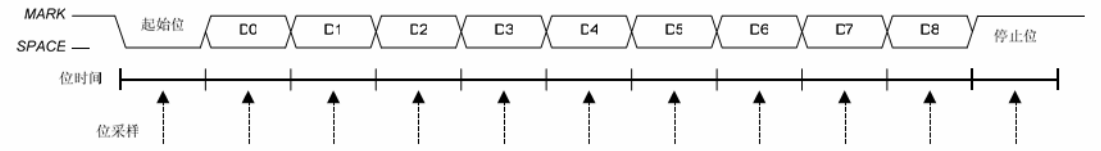


8 位 UART 时序图

3.11.2.3 9 位 UART（第九位为奇偶校验位）

在 9 位 UART 方式，每个数据字节共使用 11 位：一个起始位、8 个数据位（LSB 在先）、一个奇偶校验位和一个停止位。硬件可以自动产生和检测奇偶位（偶、奇校验可选）。发送时停止位为 2 个波特周期，接收时停止位为 1 个波特周期。

如果 MCE 为逻辑 1，则第九位必须为逻辑 1（当 MCE 为逻辑 0 时，第九位数据的状态并不重要）。如果这些条件满足，8 位数据被存入 SBUF，第九位被存入 RBX，RI 标志被置位。如果这些条件不满足，则不装入 SBUF 和 RBX，RI 标志也不会置 1。如果中断被允许，TI 或 RI 置位时将产生一个中断。



9 位 UART 时序图

3.11.2.4 奇偶校验说明

UART 可使用 PSEL 位选择奇校验或者偶校验。

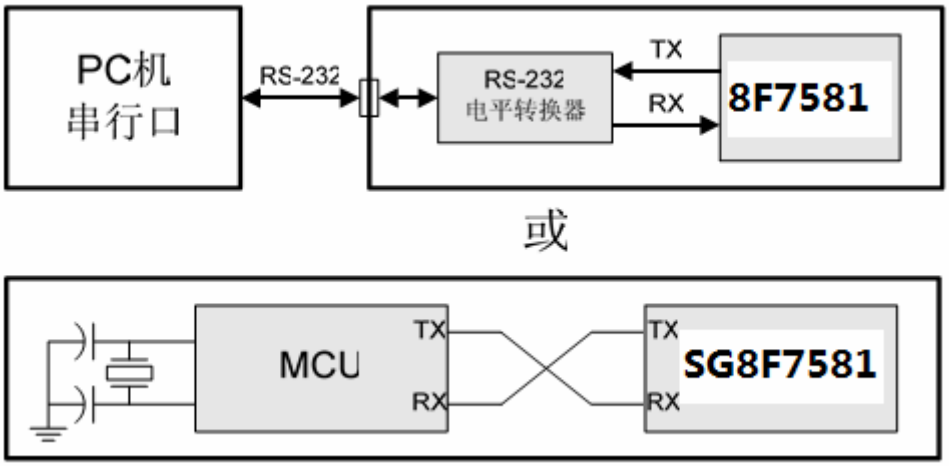
如果为奇校验，在发送过程中，发送的数据被 UART 检验，如果为奇数，则补充 0 使发送的整个 9 位数据为奇数；如果为偶数，则补充 1 使发送的整个 9 位数据为奇数。在接收过程中，接收到的 9 位数据将被检验，结果为奇数则校验正确，结果为偶数则校验错误，PERR 位会被置 1。

如果为偶校验，在发送过程中，发送的数据被 UART 检验，如果为奇数，则补充 1 使发送的整个 9 位数据为偶数；如果为偶数，则补充 0 使发送的整个 9 位数据为偶数。在接收过程中，接收到的 9 位数据将被检验，结果为偶数则校验正确，结果为奇数则校验错误，PERR 位会被置 1。

3.11.2.5 配置和操作

在典型的 UART 通信中，一个器件的发送（TX）输出被连接到其它器件的接收（RX）输入，

可以直接连接，也可以通过总线收发器。



典型 UART 连接图

3.11.2.6 数据发送

数据发送是双缓冲的，当软件向 SBUF1 寄存器写入一个字节时开始数据发送。写 SBUF 时将数据保存在发送保持寄存器，同时发送保持寄存器空标志（THRE）被清 0。如果 UART 移位寄存器为空（即没有数据在发送），则数据将被置入移位寄存器，且 THRE 被置 1。如果数据发送正在进行，则数据将保存在发送保持寄存器中，直到当前的发送过程结束。在发送结束时（停止位开始）发送中断标志 TI（SCON1.1）被置 1。如果中断被使能，则在 TI 置位时会产生中断。

硬件会根据所选择的奇偶位类型用 PSEL 产生奇偶位，并将其加到数据域之后。

如果额外位功能被使能（XBE = 1），且奇偶位功能被禁止（PARITYEN = 0），则 TBX 位将被发送（在额外位的位置）。当奇偶位功能被使能（PARITYEN = 1）时，硬件会根据所选择的奇偶位类型（用 PSEL 选择）产生奇偶位，并将其加到数据域之后。注意：当奇偶位被使能时，额外位功能不可用。

3.11.2.7 数据接收

在接收允许位 REN（SCON1.4）被置 1 后，数据接收可以在任何时刻开始。收到停止位后，如果满足下述条件则数据字节将被装入到接收缓冲寄存器：接收缓存寄存器必须未满；停止位必须为 1。在接收缓冲寄存器已满的情况下，接收的字节被丢弃，并会产生接收缓冲寄存器溢出错误（寄存器 SCON0 中的 OVR 被置 1）。如果停止位为逻辑 0，则接收数据不会被保存到接收缓冲寄存器中。如果接收条件满足，则数据被保存到接收缓冲寄存器中，且 RI 标志被置 1。如果中断被使能，

则在 RI 置位时会产生中断。

硬件会在接收数据时根据所选择的奇偶位类型（用 PSEL 选择）检查接收到的停止位。如果接收到的字节检查奇偶错误，则 PERR 标志被置 1。该标志必须用软件清 0。

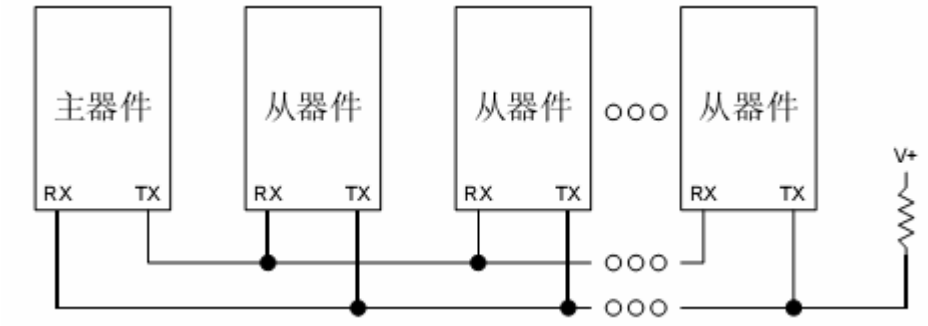
如果额外位功能被使能（XBE = 1），且奇偶位功能被禁止（PARITYEN = 0），则缓存中最早字节的额外位可以从 RBX 位读出。如果额外位功能未被使能，奇偶功能被使能，则 RBX 代表缓存中最早字节的奇偶校验位。如果奇偶位功能被使能（PARITYEN = 1），硬件会在接收数据时根据所选择的奇偶位类型（用 PSEL 选择）检查接收到的停止位。如果接收到的字节具有奇偶错误，则 PERR 标志被置 1。该标志必须用软件清 0。注意：当奇偶位被使能时，额外位功能不可用。

3.11.2.8 多机通信

9 位 UART 方式通过使用第 9 数据位可以支持一个主处理器与一个或多个从处理器之间的多机通信。当主机要发送数据给一个或多个从机时，它先发送一个用于选择目标的地址字节。地址字节与数据字节的区别是：地址字节的第 9 位为逻辑 1；数据字节的 9 位总是设置为逻辑 0。

如果从机的 MCE 位被置 1 ，则只有当 UART 接收到的第九位为逻辑 1（RBX = 1）并收到有效的停止位后 UART 才会产生中断。在 UART 的中断处理程序中，软件将接收到的地址与从机自身的 8 位地址进行比较。如果地址匹配，从机将清除它的 MCE 位以允许后面接收数据字节时产生中断。未被寻址的从机仍保持其 MCE 位为 1，在收到后续的数据字节时不产生中断，从而忽略收到的数据。一旦接收完整个消息，被寻址的从机将它的 MCE 位重新置 1 以忽略所有的数据传输，直到它收到下一个地址字节。

可以将多个地址分配给一个从机，或将一个地址分配给多个从机，从而允许同时向多个从机“广播”发送。主机可以被配置为接收所有数据传输，或通过实现某种协议使主/从角色能临时变换以允许原来的主机和从机之间进行半双工通信。



UART 多机方式连接图

3.11.2.9 方式配置表

PERITYEN	XBE	
0	0	8 位模式
0	1	9 位额外位
1	忽略	9 位奇偶位

另外，当配置 MCE 为 1 时，即多机模式时，PERITYEN 位必须设置为 0，XBE 位必须设置为 1，即使用 9 位额外位的配置方式

3.11.3 相关寄存器

3.11.3.1 地址映射

寄存器	地址	初始值	说明
SCON0	0x98	0x00	UART 控制寄存器 0
SCON1	0x99	0x04	UART 控制寄存器 1
SBUF	0x9A	0x00	UART 串行数据缓冲寄存器
SBRLL	0x9B	0x00	UART 波特率控制寄存器低字节
SBRLH	0x9C	0x00	UART 波特率控制寄存器高字节

3.11.3.2 寄存器描述

3.11.3.2.1 SCON0 UART 控制寄存器 0

位	名字	读写	描述
7	保留		
6	PSEL	R/W	PSEL：奇偶校验选择位 0：奇校验。 1：偶校验。
5	SBRUN	R/W	SBRUN：波特率发生器使能。 0：波特率发生器禁止。 1：波特率发生器使能。 该位被置 1 后，UART 被使能
4	REN	R/W	REN：接收允许

			0：UART 接收禁止。 1：UART 接收允许。
3	PARITYEN	R/W	PARITYEN：奇偶功能使能 0：禁止 UART 奇偶校验功能。 1：使能 UART 奇偶校验功能。
2	MCE	R/W	MCE：多处理器通信使能 0：RI 在停止位为 1 时置位。 1：RI 在停止位和额外位均为 1 置位（必须用 XBE1 使能额外位）。 注意：当硬件奇偶位功能被使能时，该位不可用。
1~0	SBPS	R/W	SBPS：波特率预分频选择 00：预分频 = 13。 01：预分频 = 5。 10：预分频 = 49。 11：预分频 = 1。

3.11.3.2.2 SCON1 UART 控制寄存器 1

位	名字	读写	描述
7	OVR	R/W	OVR：接收缓冲寄存器溢出标志 0：未发生接收缓冲寄存器溢出。 1：发生了接收缓冲寄存器溢出 缓冲寄存器已满，新接收的字符会被丢弃 该位必须用软件清 0
6	TBX	R/W	TBX：额外发送位 当 XBE=1 时，TBX 的值作为额外数据被发送 当 PARITYEN=1 时，该位数据无效
5	PERR	R/W	PERR：校验位 0：校验正确。 1：校验错误。 该位必须用软件清 0

4	RBX	R/W	RBX1：额外接收位 当 XBE= 1 时，RBX 缓存额外位数据 当 PARITYEN=1 时，该位数据无效
3	XBE	R/W	XBE：额外位使能 0：额外位禁止。 1：额外位使能。
2	THRE	R	THRE：发送数据寄存器空标志 0：发送数据寄存器不空，不能写 SBUF。 1：发送数据寄存器为空，可以写 SBUF。
1	TI	R/W	TI：发送中断标志 0：未发送 1：发送完成 该位必须用软件清 0
0	RI	R/W	RI：接收中断标志 0：未接收 1：接收完成 该位必须用软件清 0

3.11.3.2.3 SBUF UART 串行数据缓冲寄存器

位	名字	读写	描述
7~0	SBUF	R/W	该 SFR 用于从 UART 发送数据和从 UART 接收缓冲寄存器接收数据。 **写：**向 SBUF 写入一个字节即启动发送过程。当数据被写入 SBUF 时，它首先进入发送保持寄存器等待串行发送。当发送移位寄存器可用时，数据被传送到移位寄存器，此时可再次向 SBUF 写数据（需要满足 THRE 不为 0 即发送保持寄存器为空的状态才可写 SBUF）。 **读：**读 SBUF 时从接收缓冲寄存器中提取数据。读 SBUF 时，返回接收缓冲寄存器中的字节，该字节被从接收缓冲寄存器中清除。如果接收缓冲寄存器中还有数据字节可读，则 RI 位将保持在逻辑

			1 状态，不能被软件清除。

3.11.3.2.4 SBRLL/SBRLH UART 波特率控制寄存器

✧ 波特率控制寄存器高字节 **SBRLH**

位	名字	读写	描述
7~0	SBRLH	R/W	波特率控制寄存器高字节

✧ 波特率控制寄存器低字节 **SBRLL**

位	名字	读写	描述
7~0	SBRLL	R/W	波特率控制寄存器高字节

3.12 I2C 控制器

3.12.1 概述

I2C-BUS 控制器，支持 I2C 总线标准协议与功能。

3.12.2 特征

- 兼容标准 I2C 总线接口，可分配为主机/从机。
- 主机模式下支持多主机仲裁
- I2C 传输速率可调（最高支持 1M bit/s 传输速率(FastMode Plus)）
- 串行时钟同步允许具有不同位速率的设备通过一条串行总线进行通信
- 串行时钟同步用作握手机制以挂起及恢复串行传输；

3.12.3 功能描述

3.12.3.1 数据传输模式

典型的 I2C 总线配置如下图所示。根据方向位的状态（R/W），I2C 总线上可能存在以下两种类型的数据传输方式：

- 由主发送器向从接收器传输数据——主机发送的第一个字节是从机地址。接下来是数据字节数。从机每接收一个字节后返回一个应答位；
- 由从发送器向主接收器传输数据——由主机发送第一个字节（从机地址）。然后从机返回一个应答位。接下来是由从机发送数据字节到主机。主机接收到每个字节（最后一个字节除外）后返回一个应答位。接收到最后一个字节后，主机返回“非应答”位。主机设备产生所有的串行时钟脉冲及起始和停止条件，以停止或重复起始条件结束本次传输。由于重复起始条件也是下一次串行传输的开始，因此不释放 I2C 总线。

I2C 接口有 4 个操作模式：主机发送模式、主机接收模式、从机发送模式、从机接收模式。

3.12.3.1.1 主机发送模式

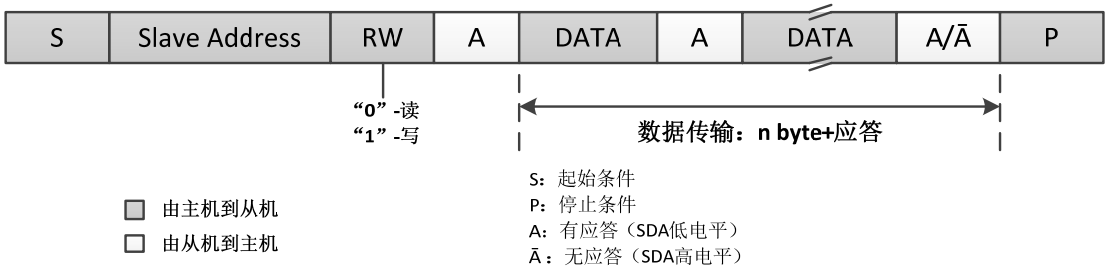
主机发送模式下，主机向从机发送数据。进入主机发送模式之前需要对 I2CCON 寄存器进行初始值化配置如下表：

Bit	7	6	5	4	3	2	1	0
I2CCON	-	I2CEN	STA	STO	SI	AA	-	-
Value	-	1	1	0	0	0	-	-

I2C 传输一次发送 8 个数据位，主机模式下发送的第一个字节包含接收器件的 7bit 从机地址+数据（读写）方向位。在主机发送模式下，需要将数据方向位写为 0，表示执行 I2C 写操作。

软件置位 STA 时，I2C 控制器将自动进入主机发送模式。一旦总线空闲，控制器就会向总线发送 I2C 起始信号，并将 SI 位置 1，I2CSTAT 寄存器状态代码自动更新为 0x08。软件依据 I2CSTAT 状态值将从机地址和数据方向位（SLA+W）写入 I2CDATA 寄存器，并清零 SI 中断标志位。

当控制器将从机地址+R/W 位发送完毕后，自动将 SI 位置 1；此时 I2CSTA 根据总线状态可能变化为 0x18、0x20、0x38，而如果从模式时，I2CSTA 也可能为 0x68、0x78、0xB0。各状态码含义请参考下 I2CSTAT 状态码。

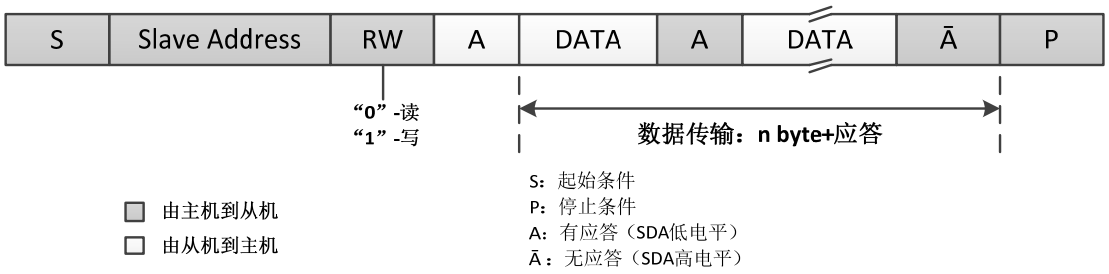


图：主机发送模式下数据传输格式

3.12.3.1.2 主机接收模式

在主机接收模式下，主机将接受从机发送的数据。该模式传输发起方式与主机发送模式相同。发送完起始条件后，中断服务程序必须将从地址(SLA)和数据方向位（R/W）装入 I2C 数据寄存器（I2CDATA），然后清零 SI 位；其中数据方向位（R/W）应为 1 ，以指示读操作。

发送完从地址和数据方向位并接收到应答位后，SI 位置位，状态寄存器将显示状态代码。对于主模式，状态代码可能为 0x40，0x48 或 0x38。对于从模式，状态代码可能为 0x68，0x78 或 0xB0。各状态码含义请参考下 I2CSTAT 状态码。



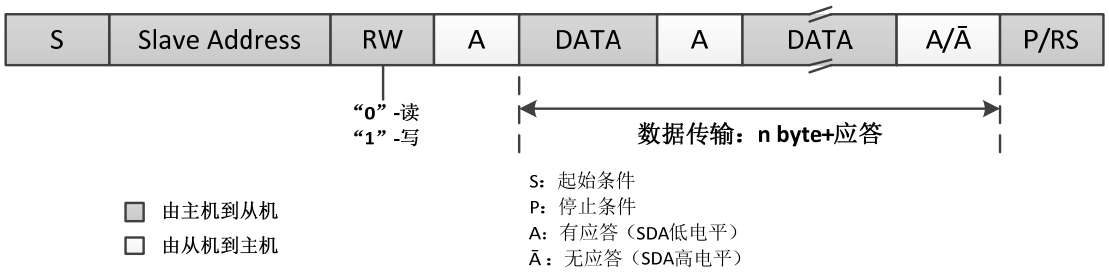
图：主机接收模式下数据传输格式

3.12.3.1.3从机接收模式

在从机接收模式下，从机接受主机发送来的数据。软件需要初始化从机地址寄存器 I2CADDR，并按下表对 I2CCON 初始化配置：

Bit	7	6	5	4	3	2	1	0
I2CCON	-	I2CEN	STA	STO	SI	AA	-	-
Value	-	1	0	0	0	1	-	-

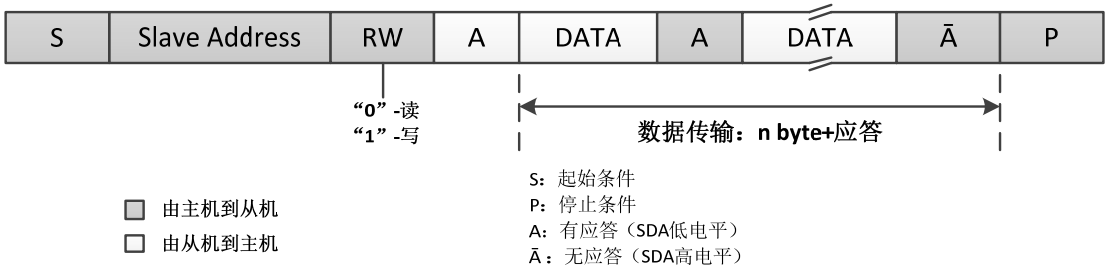
初始化配置完成后，控制器等待被 I2CADDR 定义的器件地址或通用地址匹配寻址。寻址匹配后，如果数据方向位为 0（W），则进入从机接收模式；如果数据方向位为 1（R），则进入从机发送模式。接收到寻址地址及方向位后，SI 自动置 1，并且软件可从 I2CSTAT 读取到一个有效的状态码。状态码含义请参考下 I2CSTAT 状态码。



图：从机接收模式下数据传输格式

3.12.3.1.4从机发送模式

在从机发送模式下，接收和处理第一个字节的方式与从机接收模式相同。但该模式下，接收到的数据方向位为 0，指示主机读操作（从机发送）。之后根据 SCL 输入的串行时钟，控制器将软件写入 I2CDATA 的数据由 SDA 线串行发送给通信主机。



图：从机发送模式下数据传输格式

3.12.3.2 数据速率/占空比控制（主机）

软件必须设定寄存器 I2CSCLH 和 I2CSCLL 的值来选择适当的数据速率和占空比。I2CSCLH 定义了 SCL 高电平期间的系统时钟周期数，I2SCLL 定义了 SCL 低电平期间的系统时钟周期数。I2C 最终 SCL 输出频率由下面公式得(Fsysclk 为系统时钟频率)：

$$I2C_{bitfreq} = F_{sysclk}/(I2CSCLH+I2CSCLL)$$

I2SCLL 和 I2SCLH 的值必须确保数据速率在适当的 I2C 数据速率范围内。各寄存器的值必须大于或等于 4。下表表给出了根据系统时钟频率和 I2SCLL 及 I2SCLH 值计算出来的 I2C 总线速率：

I2C 模式	I2C 位频率（bit/s）	SYSCLK 频率（Hz）					
		48M	24M	12M	6M	3M	500K
标准模式	100K	480	240	120	60	30	10
快速模式	400K	120	60	30	15	-	-
快速模式 Plus	1M	96	48	24	12	-	-

表：用于选择 I2C 时钟值的 I2CSCLH+I2CSCLL

注意 I2CSCLL 和 I2CSCLH 的值不一定要相同。软件可通过设定这两个寄存器可以得到 SCL 的不同占空比。

3.12.3.3 I2CSTAT 状态码

3.12.3.3.1主机发送模式

状态码（I2CSTAT）	I2C 总线和硬件的状态	应用软件执的响应					I2C 硬件执行的下一个操作
		I2CDATA 读写操作	I2CCON 写入操作				
			STA	ST	SI	A	

				O		A	
08H	START 发送完成	写 SLA+W	x	0	0	x	发送 SLA+W，并准备接收 ACK
10H	RSTART 发送完成	SLA+W/R	x	0	0	x	发送 SLA+W，并准备接收 ACK
18H	主机发送 SLA+W 并接收到 ACK	写发送数据	0	0	0	x	发送数据，并准备接收 ACK
		x	1	0	0	x	发送 RSTART
		x	0	1	0	x	发送 STOP
		x	1	1	0	x	发送 START 后发送 STOP
20H	主机发送 SLA+W 并接收到 NAK	写发送数据	0	0	0	x	发送数据，并准备接收 ACK
		x	1	0	0	x	发送 RSTART
		x	0	1	0	x	发送 STOP
		x	1	1	0	x	发送 START 后发送 STOP
28H	主机发送数据并接收到 ACK	写发送数据	0	0	0	x	发送数据，并准备接收 ACK
		x	1	0	0	x	发送 RSTART
		x	0	1	0	x	发送 STOP
		x	1	1	0	x	发送 START 后发送 STOP
30H	主机发送数据并接收到 NAK	写发送数据	0	0	0	x	发送数据，并准备接收 ACK
		x	1	0	0	x	发送 RSTART
		x	0	1	0	x	发送 STOP
		x	1	1	0	x	发送 START 后发送 STOP
38H	主机失去总线仲裁	X	0	0	0	x	I2C 进入从机模式
		X	1	0	0	x	I2C 进入主机模式，并发送 START

3.12.3.3.2主机接收模式

状态码（I2CSTAT）	I2C 总线和硬件的状态	软件执行操作					I2C 硬件执行的下一个操作
		I2CDATA 读写操作	I2CCON 写入操作				
			STA	STO	SI	AA	
08H	START 发送完成	SLA+W	x	0	0	x	发送 SLA+W，并准备接收 ACK

10H	RSTART 发送完成	SLA+W/R	x	0	0	x	发送 SLA+W，并准备接收 ACK
38H	主机失去总线仲裁	x	0	0	0	x	I2C 进入从机模式
		x	1	0	0	x	I2C 进入主机模式，并发送 START
40H	主机发送 SLA+R 并接收到 ACK	x	0	0	0	0	接收数据，并发送 NAK
		x	0	0	0	1	接收数据，并发送 ACK
48H	主机发送 SLA+R 并接收到 NAK	x	1	0	0	x	发送 RSTART
		x	0	1	0	x	发送 STOP
		x	1	1	0	x	发送 START 后发送 STOP
50H	主机接收数据并发送 ACK	读接收数据	0	0	0	0	接收数据，并发送 NAK
		读接收数据	0	0	0	1	接收数据，并发送 ACK
58H	主机接收数据并发送 NAK	读接收数据	1	0	0	x	发送 RSTART
		读接收数据	0	1	0	x	发送 STOP
		读接收数据	1	1	0	x	发送 START 后发送 STOP

3.12.3.3.3从机接收模式

状态码（I2CSTAT）	**I2C 总线和硬件的状态**	**软件执行操作**					**I2C 硬件执行的下一个操作**
		I2CDATA 读写操作	**I2CCON 写入操作**				
			STA	**STO**	**SI**	**AA**	
60H	从机接收 SLA+W 并发送 ACK	x	x	0	0	0	接收数据，发送 NAK
		x	x	0	0	1	接收数据，发送 ACK
68H	主机模式下失去仲裁，从机模式下接收 SLA+W，并发送 ACK	x	x	0	0	0	接收数据，发送 NAK
		x	x	0	0	1	接收数据，发送 ACK
70H	从机接收广播地址，并发送 ACK	x	x	0	0	0	接收数据，发送 NAK
		x	x	0	0	1	接收数据，发送 ACK

78H	主机模式下失去仲裁，从机模式下接收广播地址，并发送 ACK	x	0	0	0	0	接收数据，并发送 NAK
		x	0	0	0	1	接收数据，并发送 ACK
80H	接收到匹配地址，并发送 ACK	读接收数据	x	0	0	0	接收数据，并发送 NAK
		读接收数据	x	0	0	1	接收数据，并发送 ACK
88H	接收到匹配地址，并发送 NAK	读接收数据	0	0	0	0	进入不可被识别的从接模式
		读接收数据	0	0	0	1	进入从接模式
		读接收数据	1	0	0	0	发送 START，不可以被识别为从机
		读接收数据	1	0	0	1	发送 START，可以被识别为从机
90H	接收到广播地址，并发送 ACK	读接收数据	x	0	0	0	接收数据，并发送 NAK
		读接收数据	x	0	0	1	接收数据，并发送 ACK
98H	接收到广播地址，并发送 NAK	读接收数据	0	0	0	0	进入不可被识别的从接模式
		读接收数据	0	0	0	1	进入从接模式
		读接收数据	1	0	0	0	发送 START，不可以被识别为从机
		读接收数据	1	0	0	1	发送 START，可以被识别为从机
A0H	接收到 RSTART 或者 STOP	x	0	0	0	0	进入不可被识别的从接模式
		x	0	0	0	1	进入从接模式
		x	1	0	0	0	发送 START，不可以被识别为从机
		x	1	0	0	1	发送 START，可以被识别为从机

3.12.3.3.4从机发送模式

状态码（I2CSTAT）	**I2C 总线和硬件的状态**	**软件执行操作**					**I2C 硬件执行的下一个操作**
		I2CDATA 读写操作	**I2CCON 写入操作**				
			STA	**STO**	**SI**	**AA**	
A8H	从机接收 SLA+R 并发送 ACK	写发送数据	x	0	0	0	发送最后 1byte 数据，并接收 ACK
		写发送数据	x	0	0	1	发送数据，并接收 ACK

B0H	发送 SLA+W/R 时失去仲裁，接收到 SLA+R，并发送 ACK	写发送数据	x	0	0	0	发送最后 1byte 数据，并接收 ACK
		写发送数据	x	0	0	1	发送数据，并接收 ACK
B8H	从机发送数据并接收到 ACK	写发送数据	x	0	0	0	发送最后 1byte 数据，并接收 ACK
		写发送数据	x	0	0	1	发送数据，并接收 ACK
C0H	从机发送数据并接收到 NAK	x	0	0	0	0	进入不可被识别的从接模式
		x	0	0	0	1	进入从接模式
		x	1	0	0	0	发送 START，不可以被识别为从机
		x	1	0	0	1	发送 START，可以被识别为从机
C0H	从机发送最后 1byte 数据，并接收到 ACK	x	0	0	0	0	进入不可被识别的从接模式
		x	0	0	0	1	进入从接模式
		x	1	0	0	0	发送 START，不可以被识别为从机
		x	1	0	0	1	发送 START，可以被识别为从机

3.12.4 相关寄存器

3.12.4.1 地址映射

寄存器	地址	初值	说明
I2CSTAT	0xA9	0x00	I2C 状态寄存器
I2CCON	0xAA	0x00	I2C 控制寄存器
I2CDATA	0xAB	0x00	I2C 数据寄存器
I2CSCLH	0xAC	0x00	I2C 时钟速率控制寄存器高 8 位
I2CSCLL	0xAD	0x00	I2C 时钟速率控制寄存器低 8 位
I2CADDR	0xAE	0x00	I2C 从机地址寄存器

3.12.4.2 寄存器描述

3.12.4.2.1 I2CSTAT I2C 传输状态寄存器

位	名称	读写	说明
7~3	I2CSTAT	R	I2C 状态码（详见下描述）
2~0	保留	-	-

3.12.4.2.2 I2CCON I2C 控制寄存器

位	名称	读写	说明
7	保留	-	-
6	I2CEN	R/W	I2C 使能控制位 1：使能 I2C 模块 0：禁止 I2C 模块
5	STA	R/W	启动传输控制位 写： 1：I2C 进入主机模式并发送起始信号 0：I2C 进入从机模式 读： 1：I2C 工作在主机模式 0：I2C 工作在从机模式
4	STO	R/W	结束传输控制位 写： 1：I2C 发送传输停止信号 0：无效
3	SI	R/W	I2C 中断标志位 1：I2C 发生中断 0：I2C 未发生中断 注：当I2CSTAT改变时SI自动置位,该位写1无效，写0自动清零。
2	AA	R/W	ACK 控制位

			1：传输结束发送 ACK 0：传输结束发送 NAK
1	保留	-	-
0	GCA	R/W	广播传输使能位

3.12.4.2.3 I2CADDR：I2C 地址寄存器

位	名称	读写	说明
7	保留		
6~0	I2CADDR	R/W	I2C从机地址寄存器： 包含7 位从地址，用于从模式下的I2C 接口操作，主机模式下无效。最低位决定从机是否对通用调用地址作出响应

3.12.4.2.4 I2CDATA：I2C 数据寄存器

位	名称	读写	说明
7~0	I2CDATA	R/W	该寄存器保存已接收或将要发送的数据值： 写操作：写入将要发送的数据 读操作：读取已接收到的数据

注：SI 位(I2CCON bit3)置位时，即只有在该寄存器没有进行字节移位时，CPU 才可以 I2CDATA 进行读/写操作。只要 SI 位置位，I2CDATA 中的数据就保持不变。I2DATA 中的数据总是从右向左移位：要发送的第一位是 MSB（位 7），接收到一个字节后，接收到数据的第一位放在 I2DATA 的 MSB 位。

3.12.4.2.5 I2CSCLH：I2C 时钟速率控制高字节

位	名称	读写	说明
7~0	I2CSCLH	R/W	I2C 时钟速率控制高字节

3.12.4.2.6 I2CSCLL：I2C 时钟速率控制低字节

位	名称	读写	说明
7~0	I2CSCLL	R/W	I2C 时钟速率控制低字节

3.13 数模转换器（DAC）/运算放大器（OPA）/电压比较器

SG8F7581 内部集成了运算放大器(OPA)、数模转换器（DAC）以及两个电压比较器（CPT0/CPT1）。

3.13.1 功能描述

3.13.1.1 数模转换器（DAC0/DAC1）

芯片内置 DAC，两路模拟电压输出 DAC0/DAC1，分别用于为两个电压比较器（CPT0/CPT1）提供基准电压；两路 DAC 输出电压可分别由软件控制调节。另外，DAC 的基准电压可选择采用内置基准或外部引脚（由 DACREF——DAC0CN bit7 控制）；

3.13.1.2 运算放大器(OPA)

OPA 将引脚输入电压进行 15 倍（或 14 倍）放大；放大后的电压可送至 CPT1 作为比较电压输入（软件选择 OPA 输出作为比较电压输入）；也可作为 ADC 模拟输入电压。

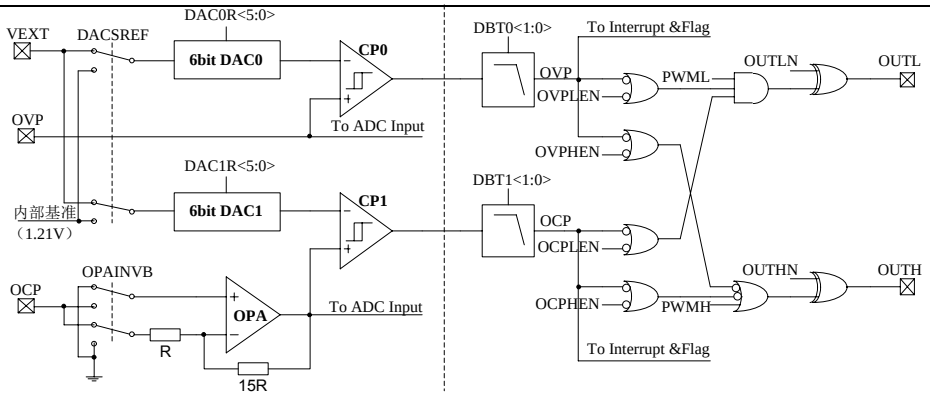
3.13.1.3 电压比较器（CP0/CP1）

芯片内置 2 个带迟滞控制的电压比较器，软件可设置比较器是否有迟滞；可选择采用 DAC 输出电压基准或外置电压基准，还可选择开启数字去抖电路，用于滤除电压不稳造成的比较结果输出抖动。

另外通过端口复用可将去抖电路处理后的比较器比较结果输出至端口，也可由软件直接从 CPCN 寄存器直接读取比较结果。同时比较器控制器设计了相关中断，当比较器输出结果变化时，会产生相应中断，并置起中断标志。另外比较器输出结果还可以作为 MCU 唤醒源。

3.13.1.4 过压/过流保护功能

SG8F7581 内置过压/过流控制功能，作为电源充放电电路的过压/过流保护。软件通过开启 OVPEN/OCPEN 控制位，可将上述 DAC/OPA/CP0/CP1 组合为过压/过流测电路，如下图所示。



该电路监测 OVP/OCP 端口电压，当出现过压/过流时，硬件可自动切断用于充放电的 PWM 引脚输出，从而使充放电电压/电流维持在可控水平。

3.13.2 相关寄存器

3.13.2.1 地址映射

寄存器	地址	初始值	说明
OVPCN	0xE9	0x00	过压控制寄存器
OCPCN	0xF1	0x00	过流控制寄存器
CPCN	0xF2	0x00	比较器和运算放大器控制/状态寄存器
DAC0CN	0xF3	0x80	DAC0 控制寄存器
DAC1CN	0xF4	0x00	DAC1 控制寄存器
CP0MD	0xF5	0x00	比较器 0 模式控制寄存器
CP1MD	0xF6	0x00	比较器 1 模式控制寄存器
CPINT	0xF7	0x00	比较器中断控制/状态寄存器

3.13.2.2 寄存器描述

3.13.2.2.1 OCPCN 过流保护电路控制寄存器

位	名称	读写	说明
7	OCPEN	R/W	过流保护电路使能位： 1 = 开启过流保护电路 0 = 关闭过流保护电路

			注：若 OCPEN 位为 0，则过流保护功能关闭，CP1、OP、DAC1 均可以配置成使能或关闭；若 OCPEN 为 1，过流保护功能使能，OP、DAC1 和 CP1 都被使能
6	保留	R	-
5	OCP2HEN	R/W	PWM2H（P56）过流保护功能控制位： 1 = 使能 0 = 关闭 注：该位用来控制发生过流时是否将 PWM2H 强制无效。若 OCPEN 和 OCP2HEN 都为 1，则发生过流时 PWM2H 将被强制无效；若 OCP2HEN 为 0，则发生过流时 PWM2H 引脚不受影响
4	OCP2LEN	R/W	PWM2L（P55）过流保护功能控制位： 1 = 使能 0 = 关闭 注：该位用来控制发生过流时是否将 PWM2L 强制无效。若 OCPEN 和 OCP2LEN 都为 1，则发生过流时 PWM2L 将被强制无效；若 OCP2LEN 为 0，则发生过流时 PWM2L 引脚不受影响
3	OCP1HEN	R/W	PWM1H（P52）过流保护功能控制位： 1 = 使能 0 = 关闭 注：该位用来控制发生过流时是否将 PWM1H 强制无效。若 OCPEN 和 OCP1HEN 都为 1，则发生过流时 PWM1H 将被强制无效；若 OCP1HEN 为 0，则发生过流时 PWM1H 引脚不受影响
2	OCP1LEN	R/W	PWM1L（P53）过流保护功能控制位： 1 = 使能 0 = 关闭 注：该位用来控制发生过流时是否将 PWM1L 强制

			无效。若 OCPEN 和 OCP1LEN 都为 1，则发生过流时 PWM1L 将被强制无效；若 OCP1LEN 为 0，则发生过流时 PWM1L 引脚不受影响
1	OCP0HEN	R/W	PWM0H(P50)过流保护功能控制位： 1 = 使能 0 = 关闭 注：该位用来控制发生过流时是否将 PWM0H 强制无效。若 OCPEN 和 OCP0HEN 都为 1，则发生过流时 PWM0H 将被强制无效；若 OCP0HEN 为 0，则发生过流时 PWM0H 引脚不受影响
0	OCP0LEN	R/W	PWM0L(P51)过流保护功能控制位： 1 = 使能 0 = 关闭 注：该位用来控制发生过流时是否将 PWM0L 强制无效。若 OCPEN 和 OCP0LEN 都为 1，则发生过流时 PWM0L 将被强制无效；若 OCP0LEN 为 0，则发生过流时 PWM0L 引脚不受影响

3.13.2.2.2 OVPCN 过压保护电路控制寄存器

位	名称	读写	说明
7	OVPEN	R/W	过压保护电路使能位: 1 = 开启过压保护电路 0 = 关闭过压保护电路 注：若 OVPEN 位为 0，则过压保护功能关闭，CP0、DAC0 均可以配置成使能或关闭；若 OVPEN 位为 1，过流保护功能使能，CP1、DAC0 都被使能
6	保留	R	-
5	OVPH2EN	R/W	PWM2H(P56)过压保护功能控制位： 1 = 使能

			0 = 关闭 注：该位用来控制发生过压时是否将 PWM2H 强制无效。若 OVPEN 和 OVP2HEN 都为 1，则发生过过时 PWM2H 将被强制无效；若 OVP2HEN 为 0，则发生过过时 PWM2H 引脚不受影响
4	OVPL2EN	R/W	PWM2L(P55)过压保护功能控制位： 1 = 使能 0 = 关闭 注：该位用来控制发生过压时是否将 OUTL 强制无效。若 OVPEN 和 OVP2LEN 都为 1，则发生过过时 PWM2L 将被强制无效；若 OVP2LEN 为 0，则发生过过时 PWM2L 引脚不受影响
3~0	保留	R	-

3.13.2.2.3 CPCN 比较器和运算放大器控制/状态寄存器

位	名称	读写	说明
7	CP1EN	R/W	比较器 1 使能位 1 = 比较器 1 使能 0 = 比较器 1 禁止 注：ENOCP=1 时比较器 1 将自动开启。
6	CP1HYST	R/W	电压迟滞使能： 1：比较器 1 迟滞功能使能 0：比较器 1 迟滞功能关闭
5	CP1OUT	R	比较器 1 输出状态标志 1 = 电压值 CP1+>CP1- 0 = 电压值 CP1+<CP1-
4	CP0EN	R/W	比较器 0 使能位 1 = 比较器 0 使能 0 = 比较器 0 禁止

			注：ENOVP=1 时比较器 0 将强制开启。
3	CP0HYST	R/W	电压迟滞使能： 1：比较器 0 迟滞功能使能 0：比较器 0 迟滞功能关闭
2	CP0OUT	R	比较器 0 输出状态标志 1 = 电压值 CP0+>CP0- 0 = 电压值 CP0+<CP0-
1	OPAEN	R/W	运算放大器功能控制 1 = 开启运算放大器，对 P41 输入模拟电压进行 15 倍增益放大。 0 = 关闭运算放大器，P41 输入模拟电压不进行增益放大。 注：ENOCP=1 时，运算放大器强制开启，对 P41 模拟输入电平进行 15 倍增益放大
0	OPAINVB	R/W	运算放大器模式控制位 1 = 同相 0 = 反相（默认）

3.13.2.2.4 DAC0CN 模数转换器 0 控制寄存器

位	名称	读写	说明
7	DACSREF	R/W	DAC 基准电压选择位： 0 = 选择内置 1.21V 基准电压 1 = 选择外部基准电压输入（P40 引脚输入，开启该功能时需要提前将 P40 置为输入端口并开启 AN 功能）
6	保留	R	-
5~0	DAC0REF	R/W	DAC0 数字量输入寄存器： DAC0 转换公式 Vdac0 = Vref/64*（DAC0REF+1）； 其中 Vref 电压值由 DACREF(CPTCN bit0)选择

3.13.2.2.5 DAC1CN 模数转换器 1 控制寄存器

位	名称	读写	说明
7~6	保留	R	-
5~0	DAC1REF	R/W	DAC1 数字量输入寄存器： DAC1 转换电压公式 Vdac1 = Vref/64*（DAC1REF+1）； 其中 Vref 电压值由 DACREF(CPTCN bit0)选择

3.13.2.2.6 CP0MD 比较器 0 模式控制寄存器

位	名称	读写	说明
7	CP0MX	R/W	模拟比较器 0 端口复用控制位： 1：模拟比较器 0 端口复用功能开启： P26 作为比较器 0 的 CP0+ 模拟输入 0：关闭比较器 0 的 CP0+ 模拟输入通道 P26； 注 1：当 OVPEN =1 时，将强制开启 CP0 端口复用功能； 注 2：开启端口复用前，需要提前将相应引脚置为输出功能关闭并开启 AN 功能
6	CP0OE	R/W	比较器 0 结果输出使能控制位： 1：比较器 0 比较结果从 P06 端口输出； 0：比较器 0 比较结果不从端口输出。
5	CP0PL	R/W	比较器 0 输出相位控制(CP0OE=1 时有效) 1：电压值 CP0+>CP0-时，端口输出低电平，反之输出高电平 0：电压值 CP0+>CP0-时，端口输出高电平，反之输出低电平
4~3	保留	R	-
2	CP0SREF	R/W	比较器 0 参考电压内置/外置选择寄存器： 1：比较器 0 选择 P27 端口输入电压作为参考电压 0：比较器 0 选择 DAC0 输出作为参考电压

			注：OVPEN=1 时，强制选择 DAC0 输出作为 CP0 基准电压
1~0	DBT0	R/W	比较器 CP0 去抖时间选择位： 00：无去抖 01：去抖时间为 8*1/fH 10：去抖时间为 16*1/fH 11：去抖时间为 32*1/fH 注: fH 为系统时钟频率

3.13.2.2.7 CP1MD 比较器 1 模式控制寄存器

位	名称	读写	说明
7	CP1MX	R/W	模拟比较器 1 端口复用控制位： 1：模拟比较器 1 端口复用功能开启： P41 作为比较器 1 的 CP1+ 模拟输入； 0：关闭 P41 到 CP1+的模拟输入通道； 注 1：当 ENOCP=1 时，将强制 P41 作为比较器 1 的 CP1+ 模拟输入； 注 2：开启端口复用前，需要提前将相应引脚置为输出功能关闭并开启 AN 功能 注 3：端口复用开启后，如果 OPAEN=0，P41 模拟输入电压直接作为 CP1+输入；如果 OPAEN=1，P41 模拟输入电压经由 OPA 放大 15 倍增益后作为 CP1+输入
6	CP1OE	R/W	比较器 1 结果输出使能控制位： 1：当 CP1MX=1 时，比较器 1 比较结果从 P07 端口输出； 0：比较器 1 比较结果不从端口输出。
5	CP1PL	R/W	比较器 1 输出极性控制位(CP1OE=1 有效) 1：电压值 CP1+>CP1-时，端口输出低电平，反之

			输出高电平 0：电压值 CP1+>CP1-时，端口输出高电平，反之输出低电平
4~3	保留	R	-
2	CP1SREF	R/W	比较器 1 参考电压内置/外置选择寄存器： 1：比较器 1 选择 P42 模拟输入电压作为参考电压 0：比较器 1 选择 DAC1 输出作为参考电压 注：OCPEN=1 时，强制选择 DAC1 输出作为 CP1 基准电压
1~0	DBT1	R/W	比较器 CP1 去抖时间选择位： 00：无去抖 01：去抖时间为 8*1/fH 10：去抖时间为 16*1/fH 11：去抖时间为 32*1/fH 注: FH 为系统时钟频率

3.13.2.2.8 CPINT 比较器中断控制/状态寄存器

位	名称	读写	说明
7	CP1RIF	R/W	比较器 1 上升沿中断标志，必须用软件清 0 1 = 自该标志位被清除后，发生了比较器 1 上升沿中断 0 = 自该标志位被清除后，没有发生比较器 1 上升沿中断
6	CP1FIF	R/W	比较器 1 下降沿中断标志，必须用软件清 0 1 = 自该标志位被清除后，发生了比较器 1 下降沿中断 0 = 自该标志位被清除后，没有发生比较器 1 下降沿中断
5	CP1RIE	R/W	比较器 1 上升沿中断使能

			1 = 使能比较器 1 上升沿中断 0 = 禁止比较器 1 上升沿中断
4	CP1FIE	R/W	比较器 1 下降沿中断使能 1 = 使能比较器 1 下降沿中断 0 = 禁止比较器 1 下降沿中断
3	CP0RIF	R/W	比较器 0 上升沿中断标志，必须用软件清 0 1 = 自该标志位被清除后，发生了比较器 0 上升沿中断 0 = 自该标志位被清除后，没有发生比较器 0 上升沿中断
2	CP0FIF	R/W	比较器 0 下降沿中断标志，必须用软件清 0 1 = 自该标志位被清除后，发生了比较器 0 下降沿中断 0 = 自该标志位被清除后，没有发生比较器 0 下降沿中断
1	CP0RIE	R/W	比较器 0 上升沿中断使能 1 = 使能比较器 0 上升沿中断 0 = 禁止比较器 0 上升沿中断
0	CP0FIE	R/W	比较器 0 下降沿中断使能 1 = 使能比较器 0 下降沿中断 0 = 禁止比较器 0 下降沿中断

3.14 ADC 转换器

SG8F7581 集成一个多通道 250ksps 的 12 位逐次逼近型 ADC，内置参考电压发生器，有多个触发源。

3.14.1 特征

- 12 位精度 SAR ADC
- 250ksps
- 内置参考电压发生器
- 多种 ADC 启动方式

3.14.2 相关寄存器

3.14.2.1 地址映射

寄存器	地址	初值	说明
ADCCN	0xB1	0x00	ADC 控制寄存器
ADCANA	0xB2	0x03	ADC 模拟控制寄存器
ADCCF	0xB3	0x00	ADC 配置寄存器
ADCMUX	0xB4	0x00	ADC 通道选择寄存器
ADCDATAH	0xB5	0x00	ADC 数据高字节寄存器
ADCDATAL	0xB6	0x00	ADC 数据低字节寄存器

3.14.2.2 寄存器描述

3.14.2.2.1 ADCCN ADC 控制寄存器

位	名称	读写	复位值	说明
7	ADC_EN	R/W	0	ADC 使能位 1 ： ADC 使能。 0 ： ADC 禁止。

6	ADC_TM	R/W	0	ADC 工作模式控制位 1 ：ADC 工作在单次转换模式下，ADC 启动转换后只完成一次转换 0 ：ADC 工作在连续转换模式下，ADC 启动转换后进行连续转换 ADC 启动转换的方式由 ADC_CM[2:0]控制
5	ADCINT	R/W	0	ADC 转换结束中断标志 1 ：ADC 完成了一次数据转换 0 ：从最后一次将该位清'0'后，ADC 还没有完成一次数据转换 该位需要用软件清 0
4	ADC_BUSY	R/W	0	ADC 忙标志 读： 1 ：ADC 正在进行转换 0 ：ADC 转换结束或 ADC 未启动转换 写： 1 ：若 ADC_CM[2:0]=000，则启动转换 0 ：无作用
3	保留	R/W	0	-
2-0	ADC_CM	R/W	000	ADC 转换启动方式选择 000：向 ADCBUSY 置 1 时启动 ADC 转换 001：定时器 0 溢出启动 ADC 转换 010：定时器 1 溢出启动 ADC 转换 011：定时器 2 溢出启动 ADC 转换 100：P57 端口上升沿启动 ADC 转换

3.14.2.2.2 ADCCF ADC 配置寄存器

位	名称	读写	复位值	说明
7	ADCLJST	R/W	0	ADC 左对齐选择位

				1 ： ADCDATAH:ADCDATAL 中的数据左对齐 0 ： ADCDATAH :ADCDATAL 中的数据右对齐
6-3	保留			
2-0	ADCSC	R/W	0	ADC 转换时钟控制位 000 ： ADC 工作时钟为 12MHz 时钟 2 分频 001 ： ADC 工作时钟为 12MHz 时钟 4 分频 010 ： ADC 工作时钟为 12MHz 时钟 6 分频 011 ： ADC 工作时钟为 12MHz 时钟 8 分频 100 ： ADC 工作时钟为 12MHz 时钟 12 分频 101 ： ADC 工作时钟为 12MHz 时钟 16 分频 110 ： ADC 工作时钟为 12MHz 时钟 24 分频 111 ： ADC 工作时钟为 12MHz 时钟 48 分频

3.14.2.2.3 ADCMUX ADC 通道选择寄存器

位	名称	读写	说明
7~5	保留	R	-
4-0	ADC_MUX	R/W	ADC 通道选择位 0x00 ： ADC 通道关闭 0x01 ： P10 作为 ADC 的输入通道 0x02 ： P11 作为 ADC 的输入通道 0x03 ： P12 作为 ADC 的输入通道 0x04 ： P13 作为 ADC 的输入通道 0x05 ： P14 作为 ADC 的输入通道 0x06 ： P15 作为 ADC 的输入通道 0x07 ： P16 作为 ADC 的输入通道 0x08 ： P17 作为 ADC 的输入通道 0x09 ： P20 作为 ADC 的输入通道 0x0A ： P21 作为 ADC 的输入通道 0x0B ： P22 作为 ADC 的输入通道

			0x0C ： P23 作为 ADC 的输入通道 0x0D ： P24 作为 ADC 的输入通道 0x0E ： P25 作为 ADC 的输入通道 0x0F ： P26 作为 ADC 的输入通道 0x10 ： P27 作为 ADC 的输入通道 0x11~0x1D: 保留（ADC 通道关闭） 0x1E ： OPA 输出作为 ADC 的输入通道 0x1F ： V_{vdd5}/3 电压作为 ADC 的输入通道

注：ADC_MUX=0x1F 用于检测电源电压，开启该通道 ADC 检测功能时需要开启 PGA_EN (ADCANA bit7)并将 ADC 放大器增益倍数设为 1 倍（PGA_GAIN=00）

3.14.2.2.4 ADCANA ADC 模拟控制寄存器

位	名称	读写	说明
7	PGA_EN	R/W	ADC 放大器使能位 1 ： 使能 ADC 放大器 0 ： 禁止 ADC 放大器
6-5	PGA_GAIN	R/W	ADC 放大器增益倍数 00 ： 增益放大倍数为 1 01 ： 增益放大倍数为 2 10 ： 增益放大倍数为 3 11 ： 增益放大倍数为 5 转换结果为输入信号根据放大倍数放大后的电压值
4	保留	R/W	-
3-2	ADC_SUBI	R/W	ADC 减电流控制位 00 ： 电流不减小 01 ： 电流减小 25% 10 ： 电流减小 50% 11 ： 电流减小 75%
1-0	ADC_SVREF	R/W	ADC 参考电压控制位

			00 ： 参考电压 2.0V 01 ： 参考电压为 P40 引脚输入电压 10 ： 参考电压为 VDD3O 电压 11 ： 参考电压为 VDD5 电压

注：当 ADC_SUBI 选择 01/10/11 时，或者 ADC_SVREF 选择 00/01 时，建议把 ADCCSC 的值设置为 001 及以上。

3.14.2.2.5 ADCDATAH ADC 数据寄存器高字节

位	名称	读写	说明
7-0	ADCDATAH	R/W	ADC 数据高字节 ADCLJST = 0 时 位 7~4 读出值为 0，位 3-0 为 12 位 ADC 数据的高 4 位 ADCLJST = 1 时 位 7-0 是 12 位 ADC 数据的高 8 位

3.14.2.2.6 ADCDATAL ADC 数据寄存器低字节

位	名称	读写	说明
7-0	ADCDATAL	R/W	ADC 数据低字节 ADCLJST =0 时 位 7-0 是 12 位 ADC 数据的低 8 位 ADCLJST = 1 时 位 7-4 是 12 位 ADC 数据的低 4 位，位 3-0 的为 0

3.15 端口复用

3.15.1 概述

SG8F7581 配置了一组端口复用控制寄存器，用于满足 I2C/UART/PCA 等外围设备的连接。当端口复用被开启时，仍可读取端口的数据。

3.15.2 相关寄存器

3.15.2.1 地址映射

寄存器	地址	初值	说明
PORTMUX	0xA1	0x00	端口复用控制寄存器

3.15.2.2 功能描述

3.15.2.2.1 PORTMUX 端口复用控制寄存器

位	名称	读写	说明
7	UARTMX	R/W	UART 端口复用控制位： 1：UART 端口复用开启： P46 复用为 UART 发送端 TX； P47 复用为 UART 接收端 RX； 0：关闭复用功能； 注：端口复用开启后，P46/P47 端口方向仍受 P4DIR 控制：需要将 P4DIR[6]配置为输出，UART TX 数据才能输出至端口。
6	I2CMX	R/W	I2C 端口复用功能控制位 1 = I2C 端口复用开启： P44：复用作 I2C SCL（端口开漏）； P45：复用作 I2C SDA（端口开漏）； 0 = 关闭复用功能；

5	PWM2HMX	R/W	PWM2H 输出复用： 1 = PWM2H 复用开启，P56 复用作 PWM2H 输出 0 = PWM2H 复用关闭
4	PWM2LMX	R/W	PWM2L 输出复用： 1 = PWM2L 复用开启，P55 复用作 PWM2L 输出 0 = PWM2L 复用关闭
3	PWM1HMX	R/W	PWM1H 输出复用： 1 = PWM1H 复用开启，P52 复用作 PWM1H 输出 0 = PWM1H 复用关闭
2	PWM1LMX	R/W	PWM1L 输出复用： 1 = PWM1L 复用开启，P53 复用作 PWM1L 输出 0 = PWM1L 复用关闭
1	PWM0HMX	R/W	PWM0H 输出复用： 1 = PWM0H 复用开启，P50 复用作 PWM0H 输出 0 = PWM0H 复用关闭
0	PWM0LMX	R/W	PWM0L 输出复用： 1 = PWM0L 复用开启，P51 复用作 PWM0L 输出 0 = PWM0L 复用关闭

3.16 FLASH 控制器

SG8F7581 集成 16KB FLASH 程序存储器，通过编程器可以进行烧录和调试，也可以通过软件代码进行在线编程

3.16.1 功能描述

3.16.1.1 程序存储器功能说明

8051 的单指令宽度是 8 位的，而 Flash 每个地址的数据宽度为 16 位，所以 Flash 每次会预取出两个指令，再根据当前指令地址输出相应指令数据。由于 Flash 每个地址数据包括两个单字节指令，所以 Flash 的地址使用的是 ADDRBUS 地址的高 15 位，最低位作为判断高低字节数据的输出，当前地址的最低位为 0，则输出低字节数据，为 1 则输出高字节数据。

当程序产生跳转指令时，Flash 控制器会自动判断地址总线是否发生了跳转，根据地址是否递增判断，并且会立即发出指令输出无效信号（INS_EN 为低），使 MCU 停止执行，3 个周期后，跳转的指令从 Flash 中读出并放到数据总线上，并将指令有效信号恢复（INS_EN 为高）。

当发生对 Flash 写操作时，Flash 控制器会将指令有效信号 INS_EN 置 0，暂停 MCU 指令执行；写操作结束后，指令有效信号 INS_EN 重新置 1，MCU 继续执行指令。

3.16.1.2 FLASH 存储器编程

对 FLASH 存储器编程的最简单方法是使用编程工具，通过 C2 接口编程，这是对未被初始化过的器件的唯一编程方法。

为了保证 FLASH 内容的正确性，强烈建议在用软件对 FLASH 存储器进行写或擦除操作的系统中使能片内 VDD 监视器。如果在 VDD 监视器未被使能的情况下进行读或写操作，将会产生 FLASH 错误复位。

3.16.1.3 FLASH 锁定和关键码功能

从用户软件写和擦除 FLASH 受 FLASH 锁定和关键码功能的保护。在进行 FLASH 操作之前，必须按顺序向 FLASH 锁定和关键码寄存器（FLKEY）写入正确的关键码。关键码为：0xA5，0xF1。写关键码的时序并不重要，但必须按顺序写。如果写关键码的顺序不对或写入

了错误的关键码，FLASH 写和擦除操作将被禁止，直到下一次系统复位。如果在正确写入关键码之前进行了 FLASH 写或擦除操作，FLASH 写和擦除也将被禁止。每次 FLASH 写和擦除操作之后，FLASH 锁定功能复位；在进行下一次 FLASH 写或擦除操作之前，必须重新写关键码。SFR 定义 1.3.2 给出了 FLKEY 寄存器的详细说明。

3.16.1.4 FLASH 擦除

可以用软件使用 MOVX 指令对 FLASH 存储器编程，像一般的操作数一样为 MOVX 指令提供待编程的地址和数据字节。在使用 MOVX 指令对 FLASH 存储器写入之前，必须先允许 FLASH 写操作。允许 FLASH 写操作的过程是：1）按顺序向 FLASH 锁定寄存器（FLKEY）写入 FLASH 关键码；2)将程序存储写允许位 PSWE（PFCTL.0）设置为逻辑‘1’(这将使 MOVX 操作指向目标 FLASH 存储器)。PSWE 位将保持置位状态，直到被软件清除。

写 FLASH 存储器可以清除数据位，但不能使数据位置‘1’，只有擦除操作能将 FLASH 中的数据位置‘1’。所以在写入新值之前，必须先擦除待编程的地址。FLASH 存储器是以 256 字节的扇区为单位组织的，一次擦除操作将擦除整个扇区（将扇区内的所有字节置为 0xFF)。擦除一个扇区（页）的步骤如下：

1. 禁止中断（建议这样做)。
2. 向 FLKEY 写第一个关键码：0xA5。
3. 向 FLKEY 写第二个关键码：0xF1。
4. 置‘1’PSEE 位（寄存器 PFCTL)，以允许 FLASH 扇区擦除。
5. 置‘1’PSWE 位（寄存器 PFCTL)，以允许 FLASH 写入。
6. 用 MOVX 指令向待擦除页内的任何一个地址写入一个数据字节。
7. 清除 PSWE 位（寄存器 PFCTL)。
8. 清除 PSEE 位（寄存器 PFCTL)。

3.16.1.5 FLASH 写操作

FLASH 存储器可以一次写一个字节，也可以一次写两个字节（一组)。寄存器 PFE0CN（SFR 定义 10.1)中的 FLBWE 位控制在一次 FLASH 写操作写入一个或两个字节。当 FLBWE 被清‘0’时，每次 FLASH 写操作写入一个字节；当 FLBWE 被置‘1’时，每次 FLASH 写操作写入两个字节（块写)。块写时间与单字节写的时间相同，在向 FLASH 存储器写入大量数据时

可以节省时间。

在单字节写 FLASH 期间，字节数据是分别写入的，每个 MOVX 写指令执行一次 FLASH 写操作。单字节写 FLASH 的建议步骤如下：

用软件对 FLASH 字节编程的步骤如下：

1. 禁止中断（建议这样做）。
2. 清除 FLBWE 位（寄存器 PFE0CN），以选择单字节写方式。
3. 置‘1’PSWE 位（寄存器 PFCTL）。
4. 清除 PSEE 位（寄存器 PFCTL）。
5. 向 FLKEY 写第一个关键码：0xA5。
6. 向 FLKEY 写第二个关键码：0xF1。
7. 用 MOVX 指令向扇区内的目标地址写入一个数据字节。
8. 清除 PSWE 位。
9. 重新使能中断。

重复步骤 5-7，直到写完每个字节。

对于 FLASH 块写，只在每个块的最后一个字节被写入（用 MOVX 写指令）后才执行 FLASH 写过程。一个 FLASH 写入块为两字节，从偶地址到奇地址。写操作必须按顺序进行（即先写以 0b 结尾的地址，后写以 1b 结尾的地址）。FLASH 写过程发生在对以 1b 结尾的地址进行的 MOVX 写操作之后。如果块中的某个字节不需要被更新，则应向该字节写 0xFF。FLASH 块写的建议步骤如下：

1. 禁止中断（建议这样做）。
2. 置‘1’FLBWE 位（寄存器 PFE0CN），以选择块写方式。
3. 置‘1’PFCTL 中的 PSWE 位。
4. 清除 PFCTL 中的 PSEE 位。
5. 向 FLKEY 写第一个关键码：0xA5。
6. 向 FLKEY 写第二个关键码：0xF1。
7. 用 MOVX 指令向块中的偶地址（以 0b 结尾）写入第一个数据字节。
8. 向 FLKEY 写第一个关键码：0xA5。
9. 向 FLKEY 写第二个关键码：0xF1。
10. 用 MOVX 指令向块中的奇地址（以 1b 结尾）写入第二个数据字节。
11. 清除 PSWE 位。
12. 重新允许中断。

重复步骤 5-10，直到写完每个块。

3.16.1.6 非易失性数据存储

FLASH 存储器除了用于存储程序代码之外还可以用于非易失性数据存储。这就允许在程序运行时计算和存储类似标定系数这样的数据。数据写入时用 MOVX 指令，读出时用 MOVC 指令。注意：MOVX 读指令总是指向 XRAM。

尽管 FLASH 存储器可以每次写一个字节，但必须首先擦除整个扇区。为了修改一个多字节数据集中的某一个字节，整个数据集必须被保存到一个临时存储区。接下来将扇区擦除，更新数据集，最后将数据集写回到原扇区。

3.16.1.7 安全选项

SG51 提供了安全选项以保护 FLASH 存储器不会被软件意外修改，以及防止产权程序代码和常数被读取。程序存储器写允许（PFCTL 寄存器中的 PSWE）和程序存储器擦除允许（PFCTL 寄存器中的 PSEE）位保护 FLASH 存储器不会被软件意外修改。在用软件修改 FLASH 存储器的内容之前，PSWE 必须被置为逻辑‘1’；在用软件擦除 FLASH 存储器之前，PSWE 位和 PSEE 位都必须被置为逻辑‘1’。此外，SG51 还提供了可以防止通过 C2 接口读取产权程序代码和常数这一安全功能。

保存在 FLASH 信息空间（INFO 区）的第 4 个字节中的安全锁定字节保护 FLASH 存储器，使其不能被非保护代码或通过 C2 接口读、写或擦除。FLASH 安全机制允许用户从 0 页（地址 0x0000 ~ 0x00FF）开始锁定 n 个 256 字节的 FLASH 页，其中 n 是安全锁定字节的反码。注意：当任何一个其他 FLASH 页被锁定时，包含 FLASH 安全锁定字节的页也被锁定。见下面的例子。

安全锁定字节： 11111101 b
反码： 00000010 b
被锁定的FLASH页： 3（前两个FLASH页 + 锁定字节页）
被锁定的地址： 前两个FLASH页： 0x0000 ~ 0x01FF
FLASH 锁定安全区字节页： INFO 区的第一页 0x0400 ~ 0x04FF

1. 除包含锁定字节的页之外，任何未被锁定的页均可被读、写或擦除。
2. 被锁定的页不能被读、写或擦除。

3. 包含锁定字节的页不能被擦除，在未被锁定时可以被读或写。

4. 读锁定字节的内容总是被允许。

5. 总是允许追加锁定页（将锁定字节中的‘1’改写为‘0’）；

6. 不能对 FLASH 页解除锁定（将锁定字节中的‘0’改写为‘1’）。

7. 保留区不能被读、写或擦除。访问保留区或任何被锁定页的操作将导致 FLASH 错误型系统复位。

FLASH 安全级别取决于对 FLASH 访问的方式。有 3 种可被限制的访问方式：经 C2 调试接口的读、写和擦除，在非锁定页执行的用户固件，在锁定页执行的用户固件。

经 C2 调试接口访问 FLASH：

1. 任何未锁定的页均可被读、写或擦除。

2. 被锁定的页不能被读、写或擦除。

3. 包含锁定字节的页在未被锁定时可以被读、写或擦除。

4. 读锁定字节的内容总是被允许。

5. 不允许追加锁定页（将锁定字节中的‘1’改写为‘0’）。

6. 对 FLASH 页解除锁定（将锁定字节中的‘0’改写为‘1’）需要使用 C2 器件擦除命令，这将擦除所有页，包括含有锁定字节的页和锁定字节本身。

7. 保留区不能被读、写或擦除。

在未锁定页执行的用户固件访问 FLASH：

在被锁定页执行的用户固件访问 FLASH：

1. 除包含锁定字节的页之外，任何未被锁定的页均可被读、写或擦除。

2. 除包含锁定字节的页之外，任何被锁定的页都可以被读、写或擦除。

3. 包含锁定字节的页不能被擦除，只能被读或写。

5. 总是允许追加锁定页（将锁定字节中的‘1’改写为‘0’）；

6. 不能对 FLASH 页解除锁定（将锁定字节中的‘0’改写为‘1’）。

7. 保留区不能被读、写或擦除。访问保留区或任何被锁定页的操作将导致 FLASH 错误型系统复位。

8. 读锁定字节的内容总是被允许。

3.16.2 相关寄存器

3.16.2.1 地址映射

寄存器	地址偏移	初始值	说明
PFCTL	0xFE	40h	FLASH 读写控制寄存器
FLKEY	0XFF	00h	FLASH 锁定和关键码寄存器

3.16.2.2 功能描述

3.16.2.2.1PFCTL FLASH 读写控制寄存器

位	名字	读写	初始值	描述
7	FOSE	R/W	0	0：禁止 FLASH 单稳态定时器。 1：使能 FLASH 单稳态定时器。 建议使能
6~4	FREQ	R/W	011	FLASH 工作频率定位寄存器 100：48MHz 011：12MHz 010：6MHz 001：3MHz 000：1.5MHz 需要配置为和当前系统时钟频率一致
3	FLBWE	R/W	0	FLBWE：FLASH 块写使能位。 该位控制软件对 FLASH 存储器的块写操作。 0：高字节和低字节分别写入 1：按 2 字节为一组写入
2	-	R	0	保留
1	PSEE	R/W	0	PSEE：程序存储擦除允许 将该位置‘1’后允许擦除 FLASH 存储器中的一个页（前提是 PSWE 位也被置‘1’）。在将该位置‘1’后，用 MOVX 指令进行一次写操作将擦除包含 MOVX 指令寻址地址

				的那个 FLASH 页。用于写操作的数据可以是任意值。 0：禁止擦除 FLASH 存储器。 1：允许擦除 FLASH 存储器。
0	PSWE	R/W	0	PSWE：程序存储写允许 将该位置‘1’后允许用 MOVX 指令向 FLASH 存储器写一个字节。在写数据之前必须先进行擦除。 0：禁止写 FLASH 存储器。 1：允许写 FLASH 存储器；MOVX 写指令寻址 FLASH 存储器。

3.16.2.2.2FLKEY FLASH 锁定和关键码寄存器

位	名字	读写	初始值	描述
7~0	FLKEY	R/W	8‘b0	FLKEY：FLASH 锁定和关键码寄存器 写：在进行 FLASH 擦除和写操作之前必须写该寄存器。在该寄存器被写入关键码 0xA5 和 0xF1 之前，FLASH 保持锁定状态。写操作的时间并不重要，但必须按顺序写。如果写 FLKEY 操作不正确或在正确写入关键码之前进行了 FLASH 操作，则 FLASH 将被锁定，直到下一次系统复位。 读：位 7-2 输出为 0； 位 1-0 指示当前的 FLASH 锁定状态 00：FLASH 写/擦除被锁定。 01：第一个关键码已被写入（0xA5）。 10：FLASH 处于解锁状态（允许写/擦除） 11：FLASH 写/擦除操作被禁止，直到下一次复位。

3.17 DMA 控制器

3.17.1 概述

DMA 系统允许在没有 CPU 参与的情况下，在外部存储器 XRAM 和外设 SFR 寄存器之间直接进行可变长度（数据长度由 DMASZ 寄存器设定）的数据传输。

DMA 进行 XRAM 读写期间，CPU 可以进行 XRAM 读写之外的其它任务。另外，在 DMA 工作期间，CPU 可以进入 IDLE 模式，同时 FLASH 会进入 STANBY 状态，以节省功耗。

DMA 通过高的外设数据吞吐量提高芯片运行效率。

SG8F7581 DMA 用于支持 USB 外设的数据传输（USB 的 3 个端点分别占用 3 个数据通道）；外设的 DMA 功能需要配置相应寄存器来工作。DMA 的 3 个通道分别支持外设的不同传输功能。每个通道在数据传输完毕（接受或发送数据长度达到 DMASZ 设定值）后，DMA 控制器会产生相应通道的中断请求，并置起对应的中断标志位。

3.17.2 功能描述

3.17.2.1 DMA 通道

DMA 包含了 3 个独立的通道:

- 通道 0——USB EPT0(端点 0)数据传输通道
- 通道 1——USB EPT1(端点 1)数据传输通道
- 通道 2——USB EPT2(端点 2)数据传输通道

3.17.2.2 DMA 传输配置

DMA 共有 3 个通道，各通道基地址(DMABAH、DMABA0L~DMABA2L)、数据深度寄存器（DMADPT0~DMADPT2）以及偏移地址寄存器（DMAOF0~DMAOF2）分别可配。

- **通道基地址寄存器**(DMABAH/DMABA0L~DMABA2L)——用于配置各通道传输（发送或接收）的起始地址；
- **通道偏移地址寄存器**（DMAOF0~DMAOF2）——用于指示传输过程中 XRAM 实际存取地址相对于通道基地址的偏移量（最大为 255）。偏移地址作为有效传输数据字节计数器使用，通道关闭时，偏移地址自动归零；通道开启后，外设通过 DMA 方式从通道设定的

基地址开始传输数据，每个字节的数据传输完毕后，对应通道的 DMAOFn(n=0~2)自动加 1。

注意当通道偏移地址增加至通道设定的深度时（DMADPTn-1），如果继续数据传输，该通道偏移地址将会清 0，并重新加计数。

另外，如果软件写通道的中断标志寄存器，也会自动清零相应的通道偏移地址。

- **DMA 读写 XRAM 当前目标地址——**由每个通道对应的基地址寄存器和偏移地址寄存器按下公式计算获得。

XRAM 目标传输地址 = 当前通道基地址 + 当前通道偏移地址；

- **数据传输深度寄存器**(DMADPT0~DMADPT2)——规定对应 DMA 通道的最大传输字节数。每个通道单次可进行最大数据长度为 256 字节的数据传输（DMADPTn==0 时为默认最大可设定深度）。
- **通道使能控制寄存器**（DMACHEN）用于允许配置完毕的通道开启 DMA 数据传输。

通道开启期间（DMACHEN 对应位置'1'），建议软件不要对相应的 DMABAH、DMABALn、DMADPTn、DMAOFn 寄存器进行写操作，避免造成传输数据错乱。

3.17.2.3 DMA 中断

软件配置 DMAINTEN 寄存器来开启对应通道的中断；目标通道中断开启后，如果数据传输（发送或接受）达到该通道 DMADPT0~DMADPT2 设定的数据深度，DMA 控制器产生相应的通道中断并置 1 中断标志位。如果屏蔽中断，不会产生中断，但会置 1 中断标志位。

3.17.2.4 DMA 仲裁

如果 DMA 访问 XRAM 与 MCU 执行 MOVX 类指令读写 XRAM 产生冲突，可由 DMA 仲裁器仲裁 XRAM 访问优先级。同一时间只有一个通道可以访问 XRAM；

优先级排序：

1) 通道 0，1，2（给 USB 使用）具有最高的优先级（USB3 个端点不会同时传输数据）。
2) MCU 内核访问 XRAM 具有次级优先级（发生 XRAM 读写冲突时 MCU 将会暂停指令执行，至所有 XRAM 读写完成后才继续执行指令）。

DMA 通道基于优先级工作，高优先级的通道首先被服务。当高优先级的通道被占用用于数据传输时，低优先级通道的请求信号产生时，便产生仲裁。低优先级的请求会保持到高优先级的通道传输完毕，然后继续低优先级的数据传输。

3.17.3 相关寄存器

3.17.3.1 地址映射

寄存器	地址	初始值	说明
DMACHEN	0xF1	0x00	DMA 通道使能寄存器
DMAINTF	0xF2	0x00	DMA 中断标志寄存器
DMAOF0	0xF3	0x00	DMA 通道 0 偏移地址寄存器
DMAOF1	0xF4	0x00	DMA 通道 1 偏移地址寄存器
DMAOF2	0xF5	0x00	DMA 通道 2 偏移地址寄存器
DMADPT0	0xFF20	0x00	**XFR:** DMA 通道 0 存储区深度控制寄存器
DMADPT1	0xFF21	0x00	**XFR:** DMA 通道 1 存储区深度控制寄存器
DMADPT2	0xFF22	0x00	**XFR:** DMA 通道 2 存储区深度控制寄存器
DMABAH	0xFF23	0x00	**XFR:** DMA 通道 0/1/2 存储区基地址高字节寄存器
DMABA0L	0xFF24	0x00	**XFR:** DMA 通道 0 存储区基地址低字节寄存器
DMABA1L	0xFF25	0x00	**XFR:** DMA 通道 1 存储区基地址低字节寄存器
DMABA2L	0xFF26	0x00	**XFR:** DMA 通道 2 存储区基地址低字节寄存器

3.17.3.2 寄存器描述

3.17.3.2.1DMACHEN：DMA 通道控制寄存器

位	名字	读写	描述
7	DCH2IE	R/W	通道 2 中断使能开关（USB2） 0 = 通道 2 中断屏蔽。 1 = 通道 2 中断使能。
6	DCH1IE	R/W	通道 1 中断使能开关（USB1） 0 = 通道 1 中断屏蔽。 1 = 通道 1 中断使能。
5	DCH0IE	R/W	通道 0 中断使能开关（USB0）

			0 = 通道 0 中断屏蔽。 1 = 通道 0 中断使能。
4~3	保留	R	-
2	DCH2EN	R/W	DMA 通道 2 使能开关（USB ept2 通道） 0 = 通道禁止。 1 = 通道使能。
1	DCH1EN	R/W	DMA 通道 1 使能开关（USB ept1 通道） 0 = 通道禁止。 1 = 通道使能。
0	DCH0EN	R/W	DMA 通道 0 使能开关（USB ept0 通道） 0 = 通道禁止。 1 = 通道使能。

注 1：DMACHEN 用于开启目标通道传输使能。通道禁止时硬件自动清零地址偏移寄存器 DMAOF。

3.17.3.2.2DMAINTF：DMA 中断标志寄存器

位	名字	读写	描述
7~3	保留	R	-
2	CH2IF	R/W	通道 2 中断标志位（USB ept 2 通道） 0 = 通道 2 未产生中断。 1 = 通道 2 产生通道地址溢出中断。 注：硬件置‘1’，软件清‘0’
1	CH1IF	R/W	通道 1 中断标志位（USB ept 1 通道） 0 = 通道 1 未产生中断。 1 = 通道 1 产生通道地址溢出中断。 注：硬件置‘1’，软件清‘0’
0	CH0IF	R/W	通道 0 中断标志位（USB ept 0 通道） 0 = 通道 0 未产生中断。 1 = 通道 0 产生通道地址溢出中断。 注：硬件置‘1’，软件清‘0’

注：地址溢出标志产生—当传输过程中 DMAOFn==DMADPTn-1 时，如果继续下一个数据传输，则

认为传输达到最大设定深度，且地址溢出，此时硬件自动将该通道的中断标志置‘1’。

3.17.3.2.3DMA 基地址寄存器

注：DMABAH 中各通道基地址高位和对应通道基地址低字节（DMABAnL）拼接后构成完整的 DMA 通道基地址（9bit 基地址）。

3.17.3.2.3.1 DMABAH DMA 通道 0~2 基地址高位

位	名字	读写	描述
7~2	保留	R	-。
2	CHBA2H	R/W	DMA 通道 2 基地址高位
1	CHBA1H	R/W	DMA 通道 1 基地址高位
0	CHBA0H	R/W	DMA 通道 0 基地址高位

3.17.3.2.3.2 DMABA0L DMA 通道 0 基地址低字节

位	名字	读写	描述
7~0	CH0BAL	R/W	DMA 通道 0 基地址低字节

3.17.3.2.3.3 DMABA1L DMA 通道 1 基地址低字节

位	名字	读写	描述
7~0	CH1BAL	R/W	DMA 通道 1 基地址低字节

3.17.3.2.3.4 DMABA2L DMA 通道 2 基地址低字节

位	名字	读写	描述
7~0	CH2BAL	R/W	DMA 通道 2 基地址低字节

3.17.3.2.4DMA 偏移地址

3.17.3.2.4.1 DMAOF0 DMA 通道 0 偏移地址

位	名字	读写	描述
7~0	CH0OF	R/W	DMA 通道 0 偏移地址

3.17.3.2.4.2 DMAOF1 DMA 通道 1 偏移地址

位	名字	读写	描述
7~0	CH1OF	R/W	DMA 通道 1 偏移地址

3.17.3.2.4.3 DMAOF2 DMA 通道 2 偏移地址

位	名字	读写	描述
7~0	CH2OF	R/W	DMA 通道 2 偏移地址

注：各通道偏移地址会在以下情况下自动清零：

- 通道使能关闭
- 软件清零通道中断标志位
- 通道传输至 DMADPTn 定义的最大深度(DMAOFn==DMADPTn-1)后，继续传输下一个数据。

3.17.3.2.5DMA 数据深度控制寄存器

3.17.3.2.6DMADPT0 DMA 通道 0 数据深度控制寄存器

位	名字	读写	描述
7~0	CH0DPT	R/W	通道 0 传输数据最大深度控制位： 用于配置通道 0 传输数据的最大深度。 注：数据深度值配置为 0x00（默认）时表示通道数据深度为 256。

3.17.3.2.7DMADPT1 DMA 通道 1 数据深度控制寄存器

位	名字	读写	描述
7~0	CH1DPT	R/W	通道 1 传输数据最大深度控制位： 用于配置通道 1 传输数据的最大深度。 注：数据深度值配置为 0x00（默认）时表示通道数据深度为 256。

3.17.3.2.8DMADPT2 DMA 通道 2 数据深度控制寄存器

位	名字	读写	描述
7~0	CH2DPT	R/W	通道 2 传输数据最大深度控制位： 用于配置通道 2 传输数据的最大深度。 注：数据深度值配置为 0x00（默认）时表示通道数据深度为 256。

4 修订记录

版本	日期	修订内容	修订者
V1.0	2016-04-07	初始版本	刘星
V1.1	2016-6-7	更新 PADMAP 和一些寄存器地址	刘星
V1.2	2016-7-1	修改错误描述	刘星
V1.3	2016-7-12	更新 UART 模块的参数	刘星