

规格说明书

SG8F6402

8-Bit Flash Microcontroller

Version 1.1

目录

1	系统概述.....	3
2	特性简介.....	3
3	管脚说明及 PAD MAP.....	5
3.1	管脚说明.....	5
3.2	PAD MAP.....	7
3.3	BONDING DIAGRAM.....	8
3.4	LQFP48L 封装 Pin 图.....	9
3.5	LQFP48L 封装尺寸图.....	10
4	结构图.....	11
4.1	SG51 微控制器内核.....	11
4.1.1	概述.....	11
4.1.2	特征.....	11
4.1.3	结构图.....	12
4.1.4	指令集.....	13
4.1.5	存储器组织.....	18
4.1.6	系统 SFR 寄存器映射.....	21
4.1.7	系统寄存器.....	22
4.2	系统模式.....	26
4.2.1	概述.....	26
4.2.2	特征.....	26
4.2.3	唤醒源.....	26
4.2.4	空闲模式 (IDLE).....	27
4.2.5	停机模式 (STOP).....	27
4.2.6	寄存器映射.....	27
4.2.7	寄存器说明.....	28
4.3	复位系统.....	29
4.3.1	概述.....	29
4.3.2	功能描述.....	29
4.3.3	寄存器映射.....	30
4.3.4	寄存器说明.....	30
4.4	中断系统.....	32
4.4.1	概述.....	32
4.4.2	特征.....	32
4.4.3	功能说明.....	32
4.4.4	SFR 寄存器映射.....	35
4.4.5	SFR 寄存器说明.....	35
4.5	时钟系统.....	39
4.5.1	概述.....	39
4.5.2	寄存器映射.....	39
4.5.3	寄存器说明.....	39

4.6	电源系统.....	41
4.6.1	概述.....	41
4.6.2	结构图.....	41
4.6.3	寄存器映射.....	41
4.6.4	寄存器说明.....	42
4.7	通用 IO 控制器.....	43
4.7.1	概述.....	43
4.7.2	特征.....	43
4.7.3	寄存器映射.....	44
4.7.4	寄存器说明.....	46
4.8	看门狗定时器.....	58
4.8.1	概述.....	58
4.8.2	特征.....	58
4.8.3	WDT 工作模式.....	58
4.8.4	超时周期.....	59
4.8.5	清零 WDT.....	59
4.8.6	非工作模式下 WDT 工作原理.....	59
4.8.7	寄存器映射.....	59
4.8.8	寄存器说明.....	59
4.9	定时器/计数器模块 Timer0~1.....	62
4.9.1	概述.....	62
4.9.1	寄存器映射.....	62
4.9.2	寄存器描述.....	62
4.10	USB 模块.....	68
4.10.1	概述.....	68
4.10.2	寄存器映射.....	68
4.10.3	控制寄存器说明.....	69
4.10.4	UDC 通信流程说明.....	76
4.10.5	UDC 通信举例.....	78
4.11	FLASH 控制器.....	79
4.11.1	程序存储器功能说明.....	79
4.11.2	FLASH 存储器编程.....	79
4.11.3	非易失性数据存储.....	82
4.11.4	安全选项.....	82
4.11.5	寄存器映射.....	84
4.11.6	寄存器说明.....	84
5	应用参考原理图.....	86
6	极限参数.....	86
7	电气参数.....	87
8	修订记录.....	90

1 系统概述

SG8F6402 是完全集成的混合信号片上系统型 MCU。片内集成了上电复位电路、LDO、看门狗、定时器、高精度 R/C 时钟、USB 控制器(low speed)、FLASH 程序/数据存储器等模块。其中 FLASH 程序/数据存储器可重复擦写，并允许 IAP(In Application Programming)。

片内的两线编程接口允许对安装在最终应用系统中产品的 MCU 进行非侵入式(不占用片内资源)、全速、在线系统调试及编程。支持调试程序时观察和修改存储器和寄存器、设置多个断点、单步/全速运行和停机等调试方法。在片上调试系统时，所有的模拟和数字外设都可全功能运行。

2 特性简介

● MCU

- 高速、流水结构的微控制器内核
- 指令代码完全兼容 MCS-8051
- 功耗控制支持空闲模式和停机模式
- 256Byte 内部 RAM 数据存储器
- 16K * 8Bit FLASH 程序存储器

● FLASH 存储器

- 可以对特定区域进行 IAP 操作
- 容量 16K * 8Bit
- 每页 256Byte
- 通过 USB 端口可以对 FLASH 进行编程(需要 IAP 支持)

● 时钟系统

- 片内高速可编程 R/C 时钟(12MHz)
- 片内低速 R/C 时钟(32KHz)

● 电源系统

- 工作电压 3.7V~5.5V 或者 3.3V (VDD 与 VDD3 短接)
- 内置 5V TO 3V LDO
- 内置 3V TO 1.8V LDO

● GPIO

- GPIO 带有内置上拉 (两档可调)

- 端口驱动电流(部分两档可调)
- LED 驱动功能
- **调试接口**
 - 全速、非侵入式的在系统调试接口(片内)
 - 支持 3 个断点
 - 单步调试
 - 可以读取或修改内部存储器及寄存器
- **USB**
 - 3 个端点
 - 端点 0 8bytes FIFO
 - 端点 1 48bytes FIFO
 - 端点 2 8bytes FIFO
- **Timer0/Timer1**
 - 兼容 MCS-51 的 Timer0/Timer1
 - 4 种工作模式
- **WDT**
 - 可从 IDLE/STOP 模式下唤醒 MCU
 - 可对 MCU 进行系统复位

3 管脚说明及 PAD MAP

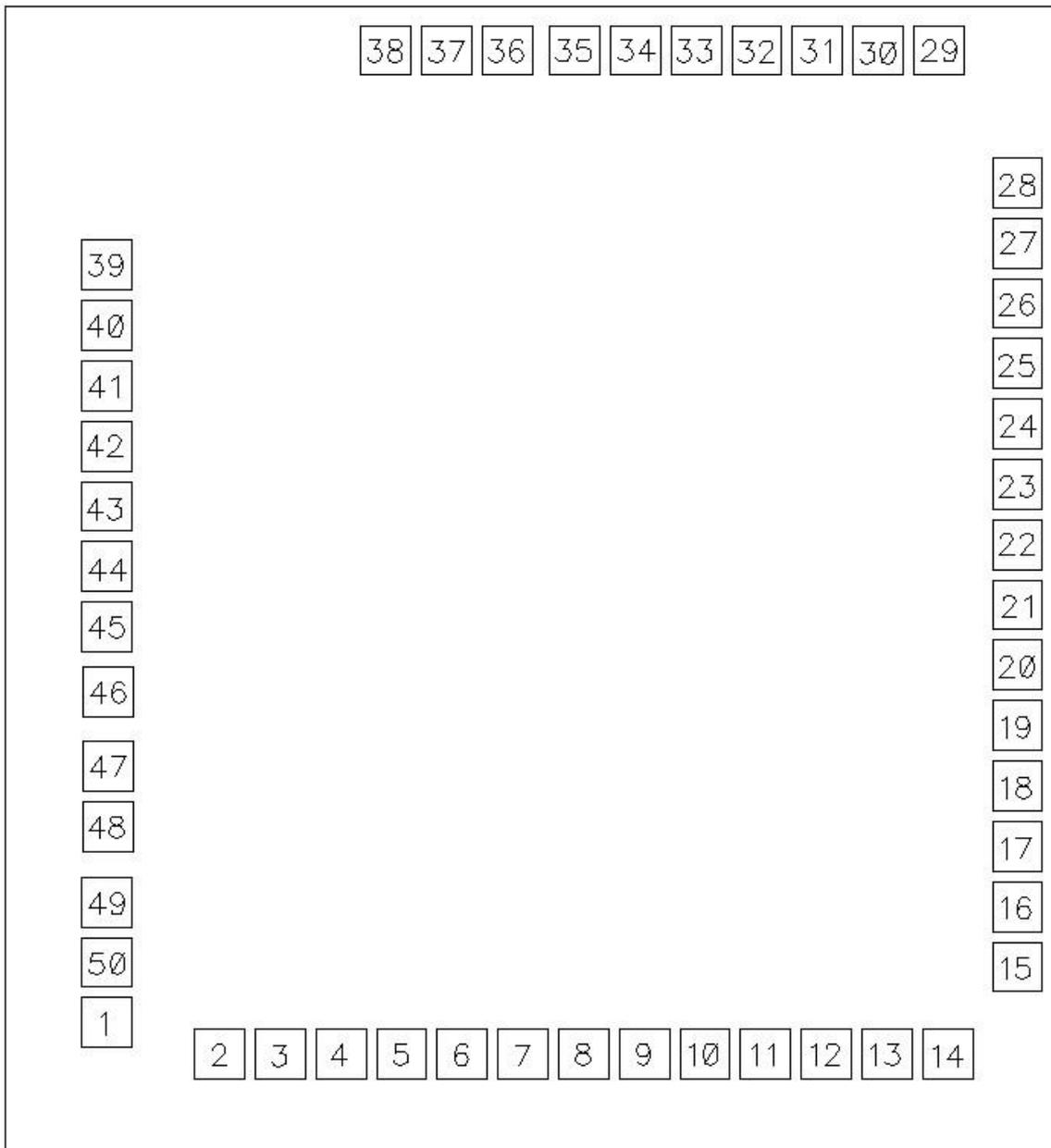
3.1 管脚说明

编号	管脚	方向	说明
1	P44	IN/OUT	GPIO
2	EXTRSTN	IN/OUT	外部复位信号输入脚(低电平有效)
		IN/OUT	C2 接口时钟输入
3	P45	IN/OUT	GPIO
			C2 接口数据输入/输出
4	P46	IN/OUT	GPIO;
5	P47	IN/OUT	GPIO;
6	P00	IN/OUT	GPIO;
7	P01	IN/OUT	GPIO;
8	P02	IN/OUT	GPIO;
9	P03	IN/OUT	GPIO;
10	P04	IN/OUT	GPIO;
11	P05	IN/OUT	GPIO;
12	P06	IN/OUT	GPIO;
13	P07	IN/OUT	GPIO;
14	FTEN	INPUT	测试模式控制引脚
15	P10	IN/OUT	GPIO;
16	P11	IN/OUT	GPIO;
17	P12	IN/OUT	GPIO;
18	P13	IN/OUT	GPIO;
19	P14	IN/OUT	GPIO;
20	P15	IN/OUT	GPIO;
21	GND	Power	系统地线
22	VDD3	Power	3.3V 电源输入

23	P16	IN/OUT	GPIO;
24	P17	IN/OUT	GPIO;
25	P20	IN/OUT	GPIO;
26	P21	IN/OUT	GPIO;
27	P22	IN/OUT	GPIO;
28	P23	IN/OUT	GPIO;
29	P24	IN/OUT	GPIO;
30	P25	IN/OUT	GPIO;
31	P26	IN/OUT	GPIO;
32	P27	IN/OUT	GPIO;
33	P30	IN/OUT	GPIO;
34	P31	IN/OUT	GPIO;
35	P32	IN/OUT	GPIO; 同时可作为 Timer0 外部时钟输入 T0
36	P33	IN/OUT	GPIO; 同时可作为 Timer1 外部时钟输入 T1
37	P34	IN/OUT	GPIO; 外部中断 INT1;
38	P35	IN/OUT	GPIO; 外部中断 INT0;
39	P36	IN/OUT	GPIO;
40	P37	IN/OUT	GPIO;
41	VDD18	Power	1.8V LDO 输出
42	VDD	Power	5V 电源输入
43	VDD3	Power	3.3V 电源输入
44	VDD3O	Power	3.3VLDO 输出
45	GND	Power	系统地线
46	VPP		Flash 测试脚
47	P40	INOUT	GPIO;
		INOUT	USB DP/PS2_CLK
48	P41	INOUT	GPIO;
		INOUT	USB DM/PS2_DATA

49	P42	IN\OUT	GPIO
50	P43	IN\OUT	GPIO

3.2 PAD MAP

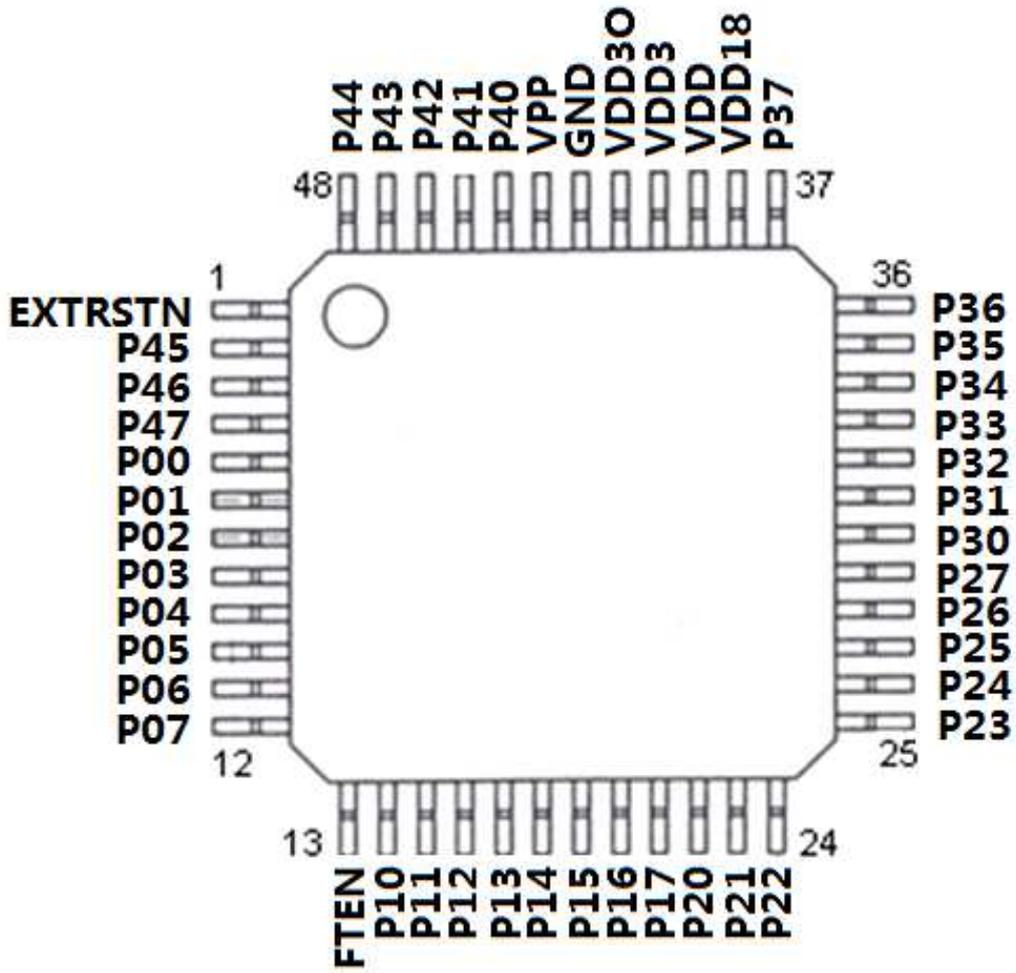


PAD Diagram of SG8F6402
Substrate Size:X=2000um;Y=2200um
Substrate Connect GND

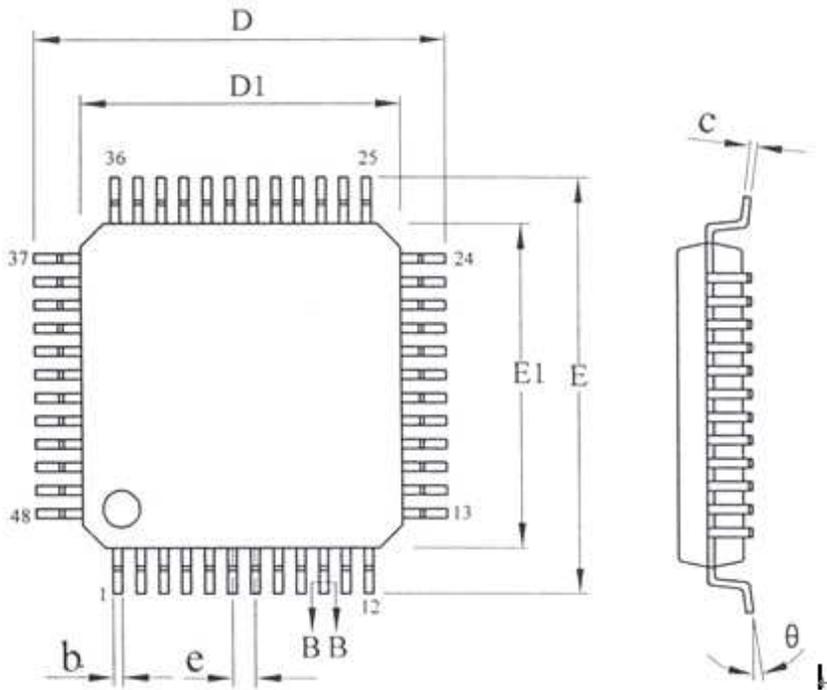
3.3 BONDING DIAGRAM

PIN NO	NAME	SG8F6402		PIN NO	NAME	SG8F6402	
		X	Y			X	Y
1	P44	142.40	186.00	26	P21	1418.60	1199.80
2	EXTRSTN	300.73	142.40	27	P22	1418.60	1284.80
3	P45	385.73	142.40	28	P23	1418.60	1369.80
4	P46	470.73	142.40	29	P24	1308.47	1556.60
5	P47	555.73	142.40	30	P25	1223.46	1556.60
6	P00	640.73	142.40	31	P26	1138.47	1556.60
7	P01	725.73	142.40	32	P27	1053.47	1556.60
8	P02	810.73	142.40	33	P30	968.47	1556.60
9	P03	895.73	142.40	34	P31	883.47	1556.60
10	P04	980.73	142.40	35	P32	798.47	1556.60
11	P05	1065.73	142.40	36	P33	703.47	1556.60
12	P06	1150.73	142.40	37	P34	618.47	1556.60
13	P07	1235.73	142.40	38	P35	533.46	1556.60
14	FTEN	1320.73	142.40	39	P36	142.40	1253.92
15	P10	1418.60	264.80	40	P37	142.40	1168.92
16	P11	1418.60	349.80	41	VDD18	142.40	1083.92
17	P12	1418.60	434.80	42	VDD5	142.40	998.60
18	P13	1418.60	519.80	43	VDD3	142.40	913.92
19	P14	1418.60	604.80	44	VDD30	142.40	829.24
20	P15	1418.60	689.80	45	GND	142.40	743.92
21	GND	1418.60	774.80	46	VPP	145.00	652.10
22	VDD3	1418.60	859.80	47	P40	145.01	547.96
23	P16	1418.60	944.80	48	P41	145.01	462.96
24	P17	1418.60	1029.80	49	P42	142.40	356.00
25	P20	1418.60	1114.80	50	P43	142.40	271.00

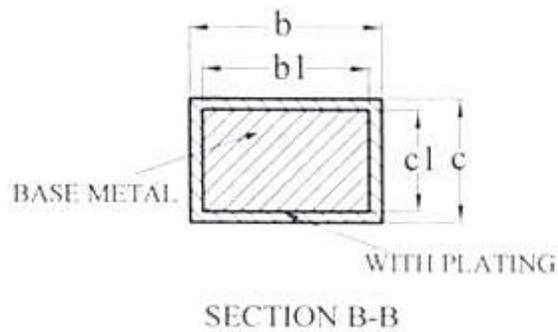
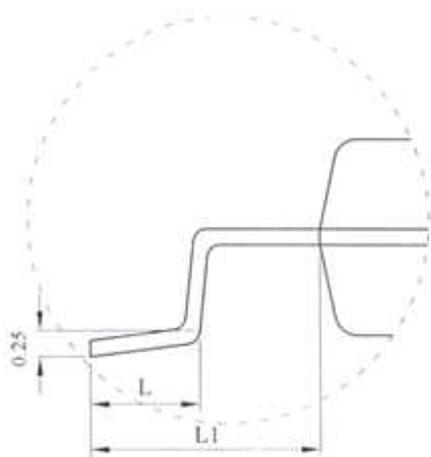
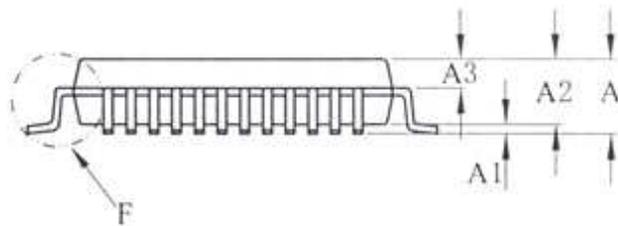
3.4 LQFP48L 封装 Pin 图



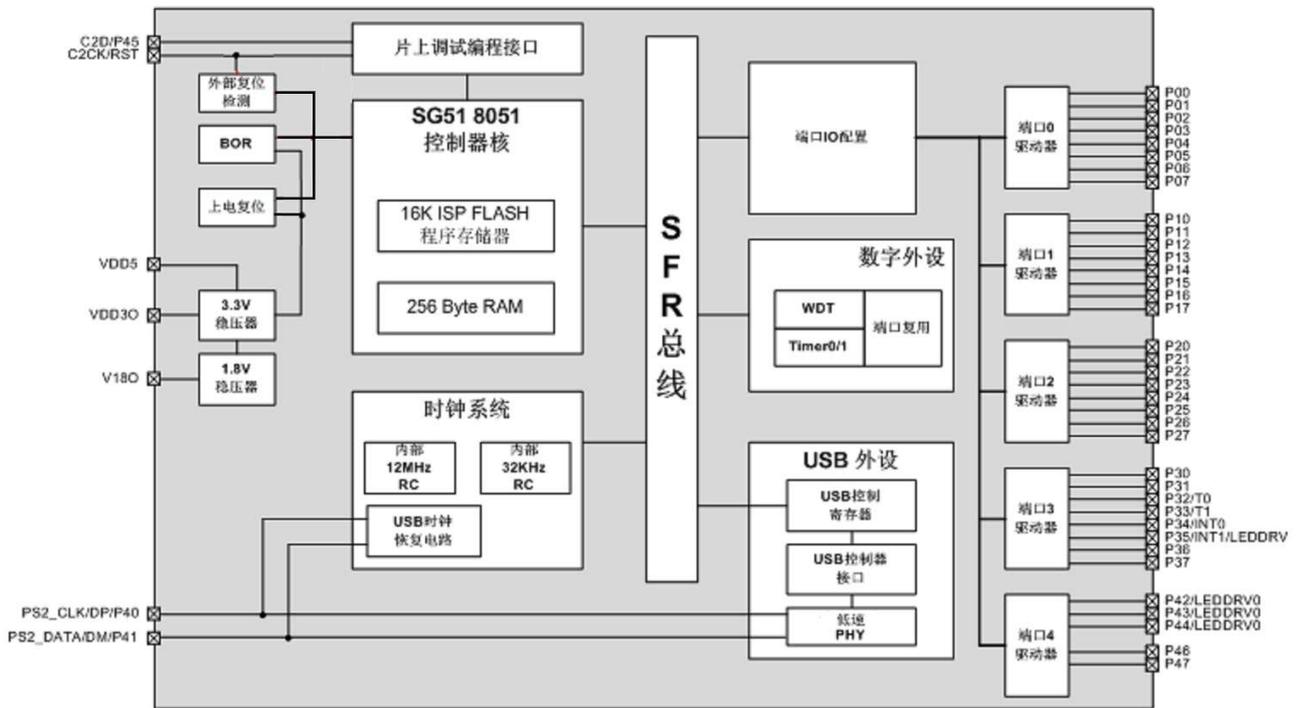
3.5 LQFP48L 封装尺寸图



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	0.15	0.25
A2	1.30	1.40	1.50
A3	0.54	0.64	0.74
b	0.19	—	0.27
b1	0.18	0.20	0.23
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.80	7.00	7.20
E	8.80	9.00	9.20
E1	6.80	7.00	7.20
e	0.50BSC		
L	0.35	0.50	0.65
L1	1.00BSC		
θ	0	—	8°



4 结构图



结构图

4.1 SG51 微控制器内核

4.1.1 概述

MCU 系统控制器的内核是 SG51 微控制器。SG51 与传统 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。

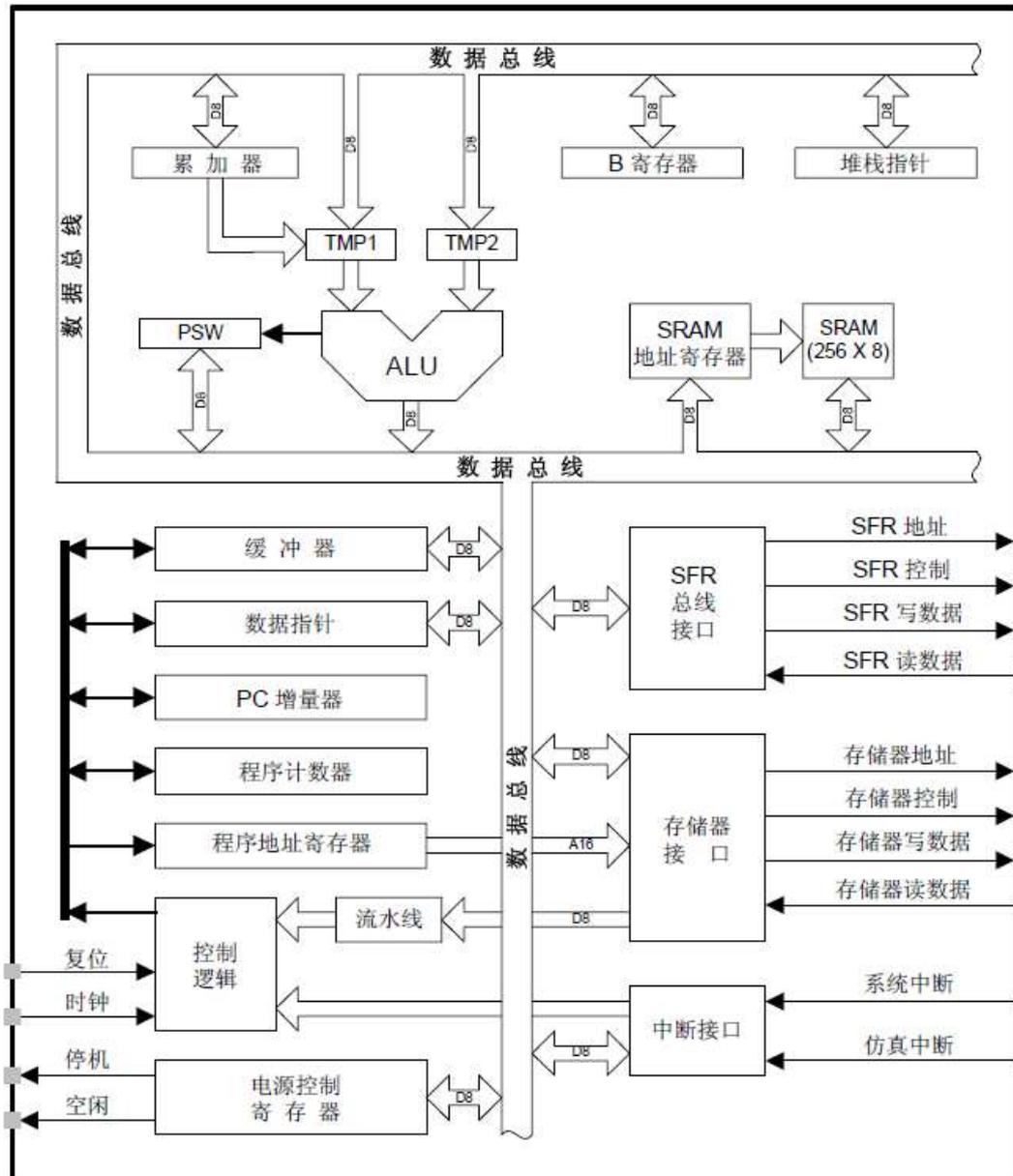
SG51 的采用流水线结构与标准的 8051 结构相比指令执行速度有很大提高。

4.1.2 特征

- 与 MCS-51 指令集完全兼容
- 0~12MHz 的时钟频率
- 256 字节内部 RAM
- 电源管理方式

- 片内调试逻辑

4.1.3 结构图



4.1.4 指令集

助记符	功能说明	字节数	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

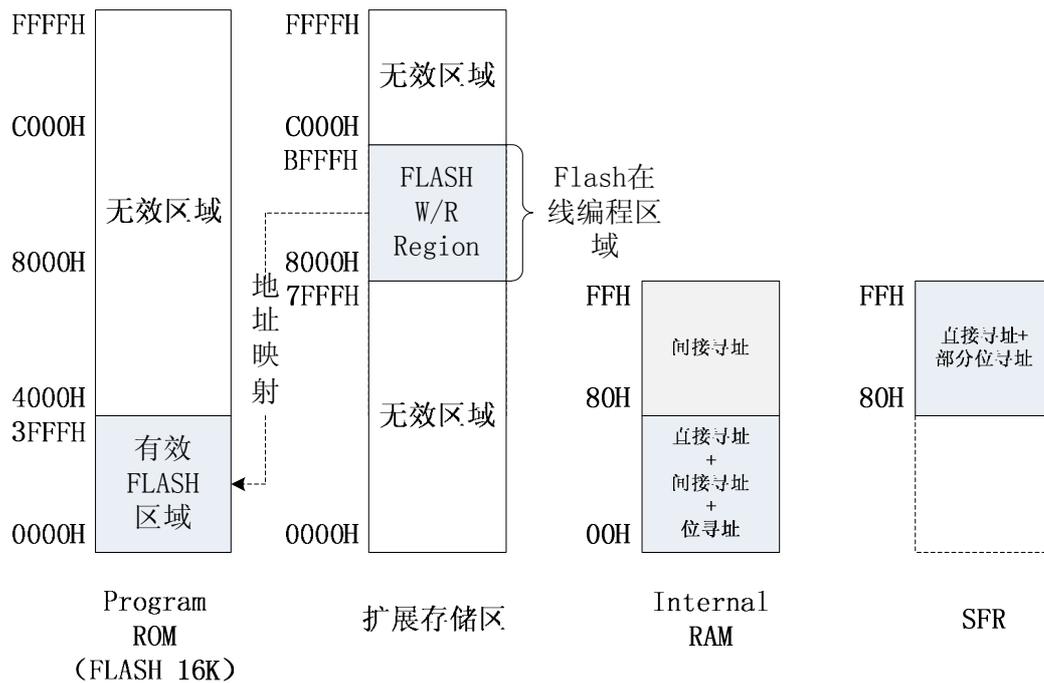
addr11 – ACALL或AJMP使用的11位目的地址。目的地址必须与下一条指令第一个字节处于同一个2K字节的程序存储器页。

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。

4.1.5 存储器组织

4.1.5.1 概述

SG51 系统控制器的存储器组织与标准 8051 的存储器组织类似。有两个独立的存储器空间：程序存储器和数据存储器。程序和数据存储器共享同一个地址空间，但用不同的指令类型访问。



4.1.5.2 程序存储器 (Program ROM)

SG51 支持 64KB 的程序存储空间。SG8F6402 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。

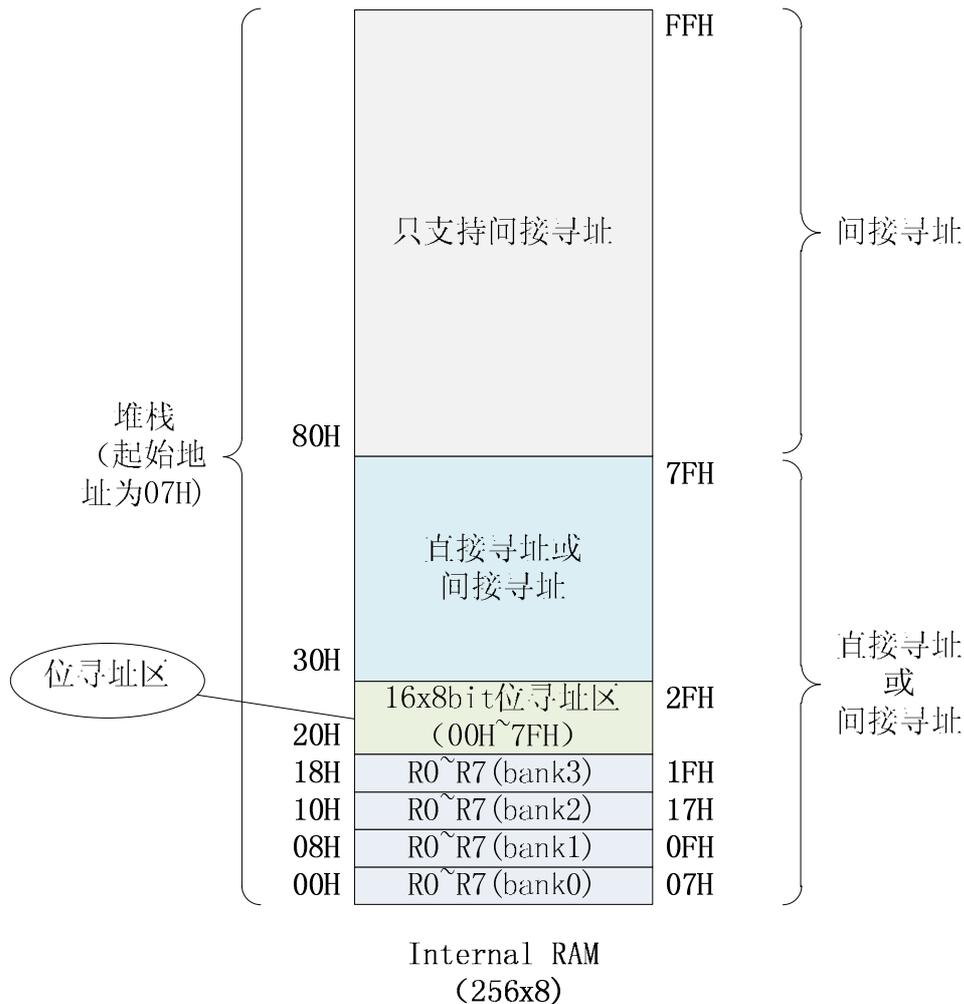
程序存储器默认只读，但 SG8F6402 可通过设置程序存储器写允许位 (SFR:PFCTL bit0)，用 MOVX 指令对 FLASH 在线编程区写入来更新 FLASH 内部数据，从而提供更新程序代码以及将程序存储器用作非易失性存储机制的功能。详细见“FLASH 控制器”。

4.1.5.3 数据存储器 (Internal RAM)

数据存储器空间中有 256 字节的内部 RAM,位于地址 0x00 到 0xFF 的地址空间。数据存储器中的低 128 字节用于通用寄存器和临时存储器。可以用直接或间接寻址方式访问。

数据存储器中的高 128 字节只能用间接寻址访问。该存储区与特殊功能寄存器(SFR)占据相同的地址空间，但物理上与 SFR 空间是分开的。当寻址高于 0x7F 的地址时，指令所用的寻址方式决定了 CPU 访问数据存储器的高 128 字节还是访问 SFR。使用直接寻址方式的指令将访问 SFR 空间，间接寻址高于 0x7F 地址的指令将访问数据存储器的高 128 字节。

该存储区结构如下图：



4.1.5.3.1 通用寄存器

数据存储器的低 32 字节，从地址 0x00 到 0x1F, 可以作为 4 个通用寄存器访问。每个区有 8 个 8 位寄存器，称为 R0~R7。在某一时刻只能选择一个寄存器区。程序状态字中的 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。间接寻址方式使用 R0 和 R1 作为间接寄存器。

4.1.5.3.2 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。

MCS-51 汇编语言允许用 XX.B 的形式代替位地址，XX 为字节地址，B 为寻址位在字节中的位置。

例如，指令：

```
MOV C, 22h.3
```

将字节地址 0x22 中的位 3（位地址为 0x13）传送到进位标志。

4.1.5.3.3 堆栈(Stack)

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。

4.1.5.4 扩展存储区（FLASH 编程区）

SG51 提供了 64KB 的外部扩展数据存储空间，该区域可用 MOVX 指令进行读写操作；SG8F6402 将该区域部分地址(0x8000~0xBFFF)分配为 FLASH 在线编程区域，可通过设置程序存储器写允许位（SFR:PFCTL bit0），用 MOVX 指令对该区域写值来更新 FLASH 内部数据。

4.1.5.5 特殊功能寄存器 SFR(Special Function Register)

从 0x80 到 0xFF 的直接寻址存储器空间为特殊功能寄存器(SFR).SFR 提供对 SG8F6402 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 0x80~0xFF 的存储器空间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,所有其它 SFR 只能按字节寻址。SFR 空间中未使用的地址保留为将来使用,访问这些地址会产生不确定的结果,应予以避免。

Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111
F8	USBCON	EPT0D	EPT1D	EPT2D	UFCLR	PFCTL	FLKEY	
F0	B							
E8								RSTRSF
E0	ACC	P0EF	P1EF	P2EF	P3EF	P4EF		
D8								
D0	PSW	WDTCN	WDTCTL					
C8								
C0	P4							
B8	IEN1	IEN2	P3INTF	P3INTE	PORTAN	P3AN	P4AN	
B0	P3	OSCHCN		OSCLCN			IP0	IP1
A8	IEN0	PORTUP	P3UPL	P3UPH	P4UP	PORTDRV	PORTWK	
A0	P2	EPT0CTL	EPT1CTL	EPT2CTL	USBADDR	USBSTA	USBINTEN	USBINTF
98							PS2UP	PWRCON
90	P1	P0DIR	P1DIR	P2DIR	P3DIR	P4DIR		PORTIF
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	XADDRH
80	P0	SP	DPL	DPH	DPL1	DPH1	DPS	PCON
备注	支持位寻址	-	-	-	-	-	-	-

4.1.6 系统 SFR 寄存器映射

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 的低字节
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

4.1.7 系统寄存器

4.1.7.1 程序计数器 PC(Program Counter)

PC 用于发出指令地址，指向程序存储器要读取的指令。指令按照顺序执行时，PC 具有自增功能，能够始终指向下一条要被取出的指令存储地址；而遇到跳转类指令时，PC 则会发出指令跳转到目标地址，存储器在目标地址读取指令继续执行。

4.1.7.2 累加器 ACC(Acumulator register)

该寄存器作为算术操作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。

4.1.7.3 B 寄存器(B regiseter)

B 寄存器 8bit 宽度，专门用于支持乘除法运算。

相关指令：

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；

除法指令 DIV AB ——ACC 除以 B，商存入 ACC，余数存入 B；

在乘除法运算之外的操作中，B 寄存器可作为一个暂存器使用。

4.1.7.4 程序状态字 PSW(Program Status Word register)

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		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 0x00~0x07 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		用户自定义
0	P	R	ACC 的奇偶校验位(Parity Flag)

4.1.7.5 堆栈指针 SP(Stack Pointer)

位	寄存器名	读写	说明
7~0	SP.7~SP.0	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。

4.1.7.6 数据指针 DPTR & DPTR1

数据指针 DPTR 位宽为 16bit，用于访问间接寻址存储器 XRAM。

SG51 核内有两个数据指针 DPTR/DPTR1，都存在于 SFR 存储区：

● **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，两个数据指针 DPTR/DPTR1 同一时间内只有一个处于活动状态（参与 DPTR 相关指令执行），软件可通过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。

4.1.7.7 数据指针选择 DPS（Data Pointer Select）

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

4.1.7.8 XRAM 寻址地址高字节 XADDRH

位	寄存器名	读写	说明
7-0	XADDRH	R/W	用 MOVX A,@Ri 或 MOVX @Ri,A 指令间接寻址读写 XRAM 时, 由 Ri(R0/R1)寄存器定位 XRAM 16bit 寻址地址低字节; 而 XADDRH 用于定位 XRAM 16bit 寻址地址高字节;

4.2 系统模式

4.2.1 概述

SG8F6402 有两种可软件编程的电源管理模式：空闲模式和停止模式。在空闲模式下，CPU 停止运行，而外设和时钟处于活动状态；在停止模式下，CPU 停止运行，所有的中断都处于非活动状态，内部振荡器停止工作。由于在空闲模式下时钟仍然运行，所以 MCU 功耗与进入空闲模式之前的系统时钟频率和处于活动状态的外设数目有关；停止模式的功耗为最小。

4.2.2 特征

- 空闲模式下可被任意中断唤醒（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）
- STOP 模式下可被端口唤醒（唤醒后系统进入中断处理子程序）、WDT 唤醒（不会执行系统复位）；此外，可被外部复位或低压复位唤醒（执行系统复位），但不可以被中断源唤醒。

4.2.3 唤醒源

- 端口唤醒：
P0/P1/P2/P3/P4 端口状态变化唤醒，如果有端口中断且中断允许，将进入相应中断处理程序，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 模式下只有不受系统时钟控制的中断源（如端口中断/WDT 中断）才能唤醒；
- 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；

4.2.4 空闲模式 (IDLE)

将空闲模式选择位(PCON.0)置 1 导致 SG8F6402 停止 CPU 运行并进入空闲模式，在执行完对该位置 1 的指令后 MCU 立即进入空闲模式。所有内部寄存器和存储器都保持原来的数据不变。所有的模拟和数字外设空闲模式期间都可以保持活动状态。

有被允许的中断发生或复位有效将结束空闲模式。当有一个被允许的中断发生时，空闲模式选择位(PCON.0)被清 0，CPU 将继续工作。该中断将得到服务程序响应，中断返回(RETI)后将开始执行设置空闲模式选择位的那条指令的下一条指令。如果空闲模式因下一个内部或外部复位而结束，则 SG8F6402 进行正常的复位过程并从地址 0x0000 开始执行程序。

系统在进入空闲模式前，如果 WDT 被允许，WDT 将可以产生一个内部看门狗复位，从而结束空闲方式。这一个功能可以防止系统不会因为对 PCON 寄存器的意外写入而导致永久性停机。如果不需要这种功能，可以在进入空闲模式之前禁止 WDT，这将进一步节省功耗，允许系统一直保持在空闲状态，等待一个外部激励唤醒系统。

4.2.5 停机模式 (STOP)

将停机模式选择位置 1 导致芯片进入停机模式，在执行完对该位置 1 的指令后 MCU 立即进入停机模式。在停机模式，内部振荡器、CPU 和所有的数字外设都停止工作。在进入停机模式之前，每个模拟外设都可以被单独关断。

进入停机模式后，由系统复位/端口唤醒/WDT 溢出/与系统时钟无关的中断使得系统退出停机模式。(如果唤醒源相关的中断使能，系统唤醒并进入工作模式后会先进入相应的中断处理程序，中断返回(RETI)后将开始执行设置停止模式选择位的那条指令的下一条指令)。

4.2.6 寄存器映射

	地址	读写	初始值	描述
PCON	0x87	WriteOnly	0x00	系统模式控制寄存器

4.2.7 寄存器说明

4.2.7.1 系统模式控制寄存器(PCON)

位	名称	读写	复位值	描述
7~2	-			保留
1	STOP	R/W	0	<p>停机模式使能</p> <p>将该位置 1，MCU 进入停机模式。</p> <p>写操作</p> <p>1 = MCU 进入停机方式(内部振荡器停止运行)</p> <p>0 = 没有影响</p> <p>读操作：</p> <p>读取时该位总为 0</p>
0	IDLE	R/W	0	<p>空闲模式使能</p> <p>将该位置 1，MCU 进入空闲模式。</p> <p>写操作：</p> <p>1 = MCU 进入空闲方式。(关闭供给 MCU 内核的时钟信号，但定时器、中断、USB 和模拟外设等仍保持活动状态。)</p> <p>0 = 没有影响</p> <p>读操作：</p> <p>读取时该位总为 0</p>

4.3 复位系统

4.3.1 概述

- 多个复位源：
 - ◇ 上电复位
 - ◇ 掉电复位
 - ◇ FLASH 错误复位
 - ◇ 软件复位
 - ◇ USB 复位

4.3.2 功能描述

4.3.2.1 上电复位

在上电期间，芯片保持在复位状态，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。

4.3.2.2 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)，此时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；

4.3.2.3 外部复位

SG8F6402 内置一个外部复位检测模块，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。

4.3.3 寄存器映射

寄存器	地址	上电复位值	说明
RSTRSF	0xEF	00uu_uu11	复位状态及控制寄存器

注：u 位 undefine，表示初值/复位值不确定。

4.3.4 寄存器说明

4.3.4.1 复位状态及控制寄存器 RSTRSF

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（默认） 0：关闭 USB 作为系统复位源的功能
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 读写错误 0：最后一次系统复位源不是来自 FLASH 读写错误
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	读：软件强制复位标志

			<p>1: 最后一次系统复位源来自软件复位</p> <p>0: 最后一次系统复位源不是来自软件复位</p> <p>写 1: 执行一次软件强制复位。写 0 无任何意义</p>
2	WDTRSF	R/W	<p>读: 看门狗定时器复位标志</p> <p>1: 最后一次系统复位源来自 WDT 溢出复位</p> <p>0: 最后一次系统复位源不是来自 WDT 溢出复位</p> <p>写: 看门狗复位使能控制</p> <p>1: 开启 WDT 作为系统复位源的功能 (默认)</p> <p>0: 关闭 WDT 作为系统复位源的功能</p>
1	LVRSF	R/W	<p>读: 低压复位标志</p> <p>1: 最后一次系统复位源来自低压复位或上电复位</p> <p>0: 最后一次系统复位源不是来自低压复位和上电复位</p> <p>写:</p> <p>1: 低压检测复位功能开启</p> <p>0: 低压检测复位功能关闭</p>
0	PORSF	R	<p>读: 低压复位标志</p> <p>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</p> <p>0: 最后一次系统复位源不是来自上电复位</p>

注: 对于既作为复位源使能 (写) 又作为复位标志 (读) 的位, 读-修改-写指令只能读和修改复位源使能位; 适用于 USBRSF 位。

4.4 中断系统

4.4.1 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 强制执行一个 LCALL 指令跳转到预定地址，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。

4.4.2 特征

- 2 级中断优先级
- 相同优先级中断下的轮询队列
- 外部中断源电平检测，沿检测可选

4.4.3 功能说明

4.4.3.1 中断源

SG8F6402 支持 9 个中断源。如果相应中断开启，且中断标志被允许，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。

4.4.3.2 外设中断源控制

SG8F6402 配置了 IEN0~IEN2 三组寄存器用于不同外设送出中断请求控制。

其中 EA 寄存器(IEN0 bit7)作为全局中断控制位。EA 为 0 时，CPU 不会响应任何中断源发出的中断请求。

某些外设中含有多个中断源，这些中断源会由各自独立的中断标志和中断控制位（在各自外设模块中设置）；而 IEN0~IEN2 寄存器中也会有一个针对该外设的总的中断控制开关（也可以理解为中断向量入口开关，控制该中断向量入口对应的所有中断），比如，USBIE(IE1 bit6)为 0 时，CPU 不会响应 USB 模块送出的所有中断请求。

4.4.3.3 端口中断

SG8F6402 一共有 40 个 IO 引脚，分为 P0~P4 共 5 组，其中 P30~P37/P40/P41 端口拥有端口中断功能，其端口状态检测模块可发出中断唤醒信号，使 MCU 进入对应的中断处理程序。

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。

P3 端口中的每个端口（P30~P37）分配一个中断标志位(SFR:P3INTF P30IF~P37IF)，每个中断标志寄存器都有一个对应的中断使能控制寄存器(SFR:P3INTE P30IE~P37IE)，用于控制各个端口中断的产生。另外对于 P3 端口中断，由 P3IE(SFR: IEN2 bit4)控制其中断源开启/关闭，如果 P3IE 为 0，MCU 将不会响应来自 P3 任一端口的中断。

P40/P41 端口共用一个中断标志位（SFR:PORTIF P4IF），由 P4IE（SFR: IEN2 bit3）控制中断入口开关。

4.4.3.4 外部中断 INT0/INT1

SG8F6402 设置引脚 P34、P35 作为外部中断 INT1/INT0 输入脚。注意外部中断（INT0/INT1）与 P34/P35 的端口中断（P34IF/P35IF）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 8051 通用的外部中断输入，中断标志位——IE0/IE1，中断使能位——EINT0/EINT1。

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（见“端口复用”部分说明）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。

IT0	IN0PL	INT0 中断
1	0	低电平有效，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	1	高电平有效，电平触发

IT1	IN1PL	INT1 中断
1	0	低电平有效，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	1	高电平有效，电平触发

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。

4.4.3.5 中断优先级

- 共设有 2 级中断优先级
- 9 个中断源在 IP 寄存器中都有单独的优先级控制位。
- 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低优先级的中断中断。
- 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，而且它们的优先级相同，则有固定的轮询队列决定哪个一个中断先得到服务。

4.4.3.6 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/擦除等操作执行期间，CPU 有可能暂停执行指令。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。

4.4.3.7 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	优先级控制
外部中断 0	0x0003	轮 询 队 列	IE0(TCON.1)	Y	eint0(IE0.6)	ip0.7
timer0 溢出	0x000B		TF0(TCON.5)	Y	t0ie(IE0.5)	ip0.6
外部中断 1	0x0013		IE1(TCON.3)	Y	eint1(IE0.4)	ip0.5

timer1 溢出	0x001B	列 由 上 到 下	TF1(TCON.6)	Y	t1ie(IE0.3)	ip0.4
usb 中断	0x0023		略	N	usbie(IE1.6)	ip0.3
WDT 中断	0x002B		WDTIF(WDTCF.2)	N	wdtie(IE1.0)	ip0.2
EFT 中断	0x0033		EFTIF(PWRCON.0)	N	eftie(IE2.7)	ip0.1
P3 端口中 断	0x003B		P30IF~P37IF(P3IF)	N	p3ie(IE2.4)	ip0.0
P4 端口中 断	0x0043		PORTIF	N	p4ie(IE2.3)	ip1.7

4.4.4 SFR 寄存器映射

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	0xB6	0x00	中断优先级配置寄存器 0
IP1	0xB7	0x00	中断优先级配置寄存器 1

4.4.5 SFR 寄存器说明

4.4.5.1 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 控制位： 0: 禁止外部中断 0

			1: 允许 int0 的中断请求
5	T0IE	R/W	timer0 中断控制位: 0: 禁止 timer0 中断 1: 允许 timer0 中断请求
4	EINT1	R/W	外部中断 1 控制位: 0: 禁止外部中断 1 1: 允许 int1 的中断请求
3	T1IE	R/W	timer1 中断控制位: 0: 禁止 timer1 中断 1: 允许 timer1 中断请求
2~0	保留		

4.4.5.2 IEN1——中断允许寄存器 1

位	寄存器名	读写	说明
7	保留		
6	USBIE	R/W	USB 中断控制位: 0: 禁止 USB 中断 1: 允许 USB 中断请求
5~1	保留		
0	WDTIE	R/W	WDT 中断控制位: 0: 禁止 WDT 中断 1: 允许 WDT 中断

4.4.5.3 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
6~5	保留		
4	P3IE	R/W	端口 3 中断控制位： 0：禁止端口 3 中断 1：允许端口 3 中断请求
3	P4IE	R/W	端口 4 中断控制位： 0：禁止端口 4 中断 1：允许端口 4 中断请求
2~0	保留	-	保留

4.4.5.4 中断优先级控制寄存器 IP0~IP1

● IP0

位	寄存器名	读写	说明
7	IP_IE0	R/W	外部中断 0 优先级配置位 1：配置为高优先级 0：配置为低优先级
6	IP_TF0	R/W	timer0 溢出中断优先级配置位 1：配置为高优先级 0：配置为低优先级
5	IP_IE1	R/W	外部中断 1 优先级配置位 1：配置为高优先级 0：配置为低优先级

4	IP_TF1	R/W	timer1 溢出中断优先级配置位 1: 配置为高优先级 0: 配置为低优先级
3	IP_USBIF	R/W	usb 中断优先级配置位 1: 配置为高优先级 0: 配置为低优先级
2	IP_WDTIF	R/W	WDT 中断优先级配置位 1: 配置为高优先级 0: 配置为低优先级
1	IP_EFTIF	R/W	EFT 中断优先级配置位 1: 配置为高优先级 0: 配置为低优先级
0	IP_P3IF	R/W	P3 端口中断优先级配置位 1: 配置为高优先级 0: 配置为低优先级

● IP1

位	寄存器名	读写	说明
7	IP_P4IF	R/W	P4 端口中断优先级配置位 1: 配置为高优先级 0: 配置为低优先级
6~0	保留		

4.5 时钟系统

4.5.1 概述

- 高频 RC 振荡器(12MHz)
- 低频 RC 振荡器(32KHz)
- RC 系统时钟可分频
- 系统时钟可在运行过程中切换
- USB 时钟自动校准

4.5.2 寄存器映射

寄存器	地址	复位值	说明
OSCHCN	0xB1	0000_0011	内部高频振荡器控制寄存器
OSCLCN	0xB3	1p00_0000	内部低频振荡器控制寄存器

注：u(undefined)表示初值/复位值不确定；p 表示保持复位前状态。

4.5.3 寄存器说明

4.5.3.1 内部高频振荡器分频控制寄存器 OSCHCN

位	名称	读写	说明
7~2	保留		
1~0	IFCN	R/W	RC12M 分频比控制位： （用于提供系统时钟，仅在 CLKSEL bit0:PLLSEL=0 时有效） 11 = 不分频； 10 = 2 分频； 01 = 4 分频； 00 = 8 分频；

4.5.3.2 内部低频振荡器控制寄存器 OSCLCN

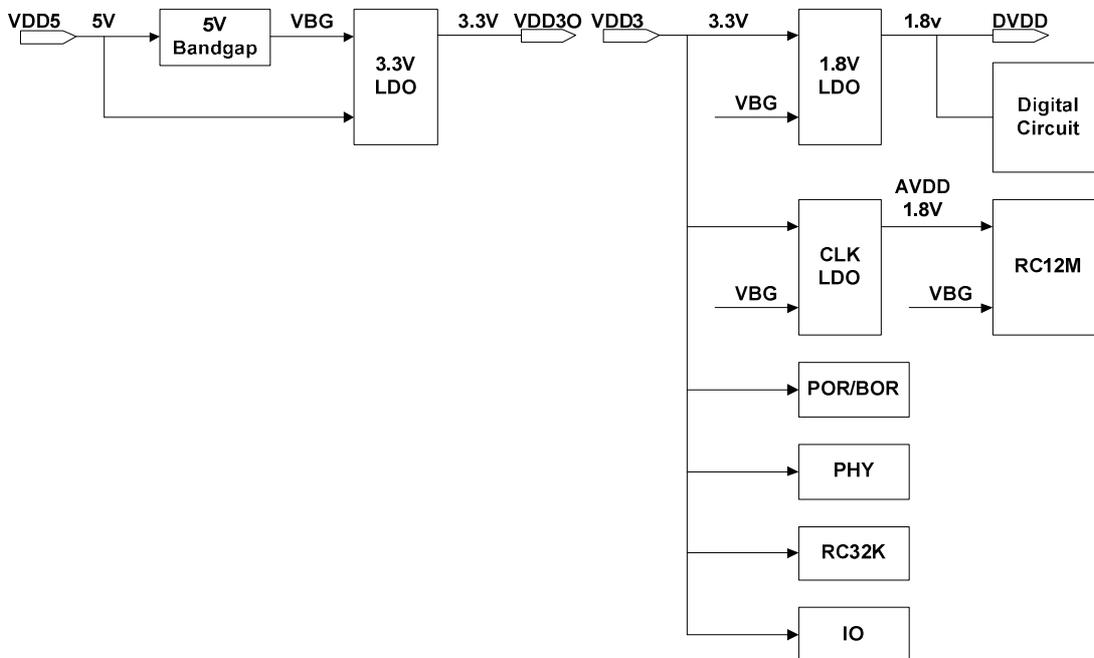
位	名称	读写	说明
7	LFEN	R/W	RC32K 使能控制寄存器 1 = RC32K 振荡器开启 0 = RC32K 振荡器关闭 注：EFT 功能开启时会强制开启 RC32K 时钟；
6	LFREADY	R	RC32K 稳定标志位 1 = RC32K 时钟输出稳定 0 = RC32K 时钟输出未稳定
5~0	保留	-	-

4.6 电源系统

4.6.1 概述

- 电压基准
- USB 物理接口供电 LDO3.3V
- 数字内核供电 LDO1.8V
- 模拟电路供电

4.6.2 结构图



- ①5V 应用时，VDD5 输入 5V，3.3V 输出 VDD30 和 3.3V 输入 VDD3 绑定在一起；
- ②数字电路 LDO 上电后一直工作。
- ③RC12M 用单独的一个 1.8V LDO 供电。

4.6.3 寄存器映射

寄存器	地址	上电复位值	说明
PWRCON	0x9F	1100_0000	电源管理寄存器

4.6.4 寄存器说明

4.6.4.1 电源管理寄存器(PWRCON)

位	名称	读写	描述
7:4	保留		
3	*EFT_MX		EFT 端口复用使能控制位： 1= EFT 端口复用开启； P06 复用作 EFT 模拟模块输出 <code>eft_ana</code> P07 复用作 EFT 数字模块输出 <code>eft_flag</code> 0= EFT 端口复用关闭；
2:1	EFT_CTRL		EFT 控制位 00 = 关闭 EFT 检测模块 01 = 在检测到 EFT 干扰后只产生 EFT 中断 10 = 检测到 EFT 干扰后提前停止 MCU 内核工作 11 = 检测到 EFT 干扰则自动停止 MCU 内核工作
0	EFTIF		EFT 中断标志寄存器 1 = 发生了 EFT 中断 0 = 未发生 EFT 中断

注：PWRCON 寄存器标注“*”寄存器位仅用于测试

4.7 通用 IO 控制器

4.7.1 概述

数字和模拟资源可以通过 40 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）。

4.7.2 特征

- 部分 I/O 口可配置为模拟或数字端口
- 可配置 I/O 口方向
- 可实现开漏模式（由 Px 数据寄存器和 PxDIR 方向寄存器配合实现）
- 部分 IO 口具有两档驱动能力（2.5mA/20mA 两档可选）
- 部分 IO 口具有定制 LED 驱动功能（2mA/4mA/8mA 三档可选）
- 二个并联的上拉电阻，分别可控
- 可作为唤醒源
- P3 端口中，每个端口都有独立的中断标志位及中断使能控制

4.7.2.1 开漏输出

SG8F6402 并未配置专门的开漏输出控制寄存器。如果需要实现开漏输出功能，可将 P0~P4 端口数据寄存器相应位配置为“0”，之后通过切换该端口的方向控制寄存器，来实现输出‘0’或输入高阻；

4.7.2.2 输出驱动

SG8F6402 为 P33~P35、P42~44 两组端口分别配置了输出驱动控制寄存器（PORTDRV），用于控制端口驱动能力。P33~P35、P42~44 两组端口有两种驱动方式可选：由 P3LEDDRV 和 P4LEDDRV 控制的 LED 驱动功能，由 P3HDRV 和 P4HDRV 控制的 GPIO 驱动功能。

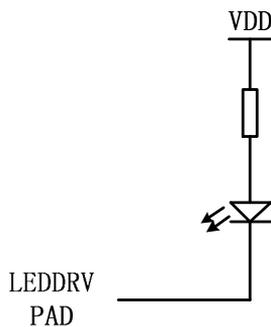
- **GPIO 驱动:**

这两组端口的 LED 驱动功能关闭时（PxLEDDRV=00），GPIO 驱动能力由 PxHDRV 控制。此时，有 2.5mA、20mA 两档驱动电流能力可选；使用 20mA 档时，注意 IO 外接负载驱动电流不能过大，以防止

LDO 电流输出能力不足导致电压不稳。

● **LED 驱动:**

要开启 P33~P35、P42~P44 两组端口的 LED 驱动功能，需要将 PxHDRV 设置为 20mA 驱动档，并开启 LED 驱动功能（由 PxLEDDRIV 寄存器选择任意档的驱动电流）。同时，将端口方向控制寄存器设置为输出，之后由端口数据寄存器（P3,P4）控制是否开启对外驱动电流输出。端口驱动 LED 方式如下图：



LED 驱动电流 2mA、4mA、8mA 有 3 档可选。

需要注意的是 P33~P35、P42~P44 两组端口同一时间只能开启一组端口的 LED 驱动功能（内置 LDO 驱动能力有限，对外驱动电流过大可能导致 LDO 输出电压不稳）。

另外，当芯片进入 STOP 模式时，LED 对外驱动将自动关闭。

4.7.2.3 上拉控制

SG8F6402 为指定 IO 配置了两档上拉电阻（10KΩ、20KΩ），由不同寄存器控制两档上拉的开启或关闭。同时，两档上拉可同时开启（上拉电阻并联）以提供阻值更小的上拉能力。

4.7.3 寄存器映射

寄存器	地址	复位值	说明
P0	0x80	0000_0000	P0 端口数据寄存器
P1	0x90	0000_0000	P1 端口数据寄存器
P2	0xA0	0000_0000	P2 端口数据寄存器
P3	0xB0	0000_0000	P0 端口数据寄存器
P4	0xC0	0000_0000	P4 端口数据寄存器

P0DIR	0x91	1111_1111	P0 端口方向控制寄存器
P1DIR	0x92	1111_1111	P1 端口方向控制寄存器
P2DIR	0x93	1111_1111	P2 端口方向控制寄存器
P3DIR	0x94	1111_1111	P3 端口方向控制寄存器
P4DIR	0x95	1111_1111	P4 端口方向控制寄存器
PORTUP	0xA9	0111_0000	P0~P2 端口上拉控制寄存器
P3UPL	0xAA	1111_0000	P30~P33 端口上拉控制寄存器
P3UPH	0xAB	1111_0000	P34~P37 端口上拉控制寄存器
P4UP	0xAC	1111_0000	P4 端口扩展上拉控制寄存器
PORTDRV	0xAD	0000_0000	端口驱动控制寄存器
PORTWK	0xAE	0000_0000	端口唤醒控制寄存器
PORTIF	0x97	0000_0000	端口中断寄存器
P3INTF	0xBA	0000_0000	P3 端口中断标志寄存器
P3INTE	0xBB	0000_0000	P3 端口中断屏蔽寄存器
PORTAN	0xBC	0000_0000	P0/P1/P2 数字输入功能控制寄存器
P3AN	0xBD	0000_0000	P3 数字输入功能控制寄存器
P4AN	0xBE	0000_0000	P4 数字输入功能控制寄存器
P0EF	0xE1	0000_0000	P0 端口沿检测状态标志寄存器
P1EF	0xE2	0000_0000	P1 端口沿检测状态标志寄存器
P2EF	0xE3	0000_0000	P2 端口沿检测状态标志寄存器
P3EF	0xE4	0000_0000	P3 端口沿检测状态标志寄存器
P4EF	0xE5	0000_0000	P4 端口沿检测状态标志寄存器
PS2UP	0x9E	0000_0000	PS2 上拉控制寄存器

4.7.4 寄存器说明

4.7.4.1 端口数据寄存器 (P0~P4)

位	名称	读写	说明
7~0	Px.n	R/W	<p>端口数据寄存器，bit7~bit0 分别对应该组的 7 个端口</p> <p>写操作——写入端口寄存器供 I/O 引脚输出的数据</p> <p>1: 输出逻辑高电平</p> <p>0: 输出逻辑低电平</p> <p>注 1: 如果 PxDIR 将端口设为输入端口，或端口复用至其他功能时，Px.n 寄存器的值将不再影响端口状态。</p> <p>读操作——读取 I/O 引脚当前的逻辑电平</p> <p>1: 相应引脚为逻辑高电平</p> <p>0: 相应引脚为逻辑低电平</p> <p>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。</p>

4.7.4.2 端口方向控制寄存器 (P0DIR~P4DIR)

位	名称	读写	说明
7~0	PxxDIR.n	R/W	<p>端口方向寄存器，bit7~bit0 分别对应该组的 7 个端口</p> <p>1: 相应引脚为数字输入引脚，数字输出功能关闭</p> <p>0: 相应引脚数字输出功能开启</p>

4.7.4.3 P0~P2 端口上拉控制寄存器 (PORTUP)

位	名称	读写	说明
7	保留		
6	P2UP2	R/W	P2 端口 20K 上拉控制寄存器位 1: P2 端口内部 20K 上拉开启 0: P2 端口内部 20K 上拉关闭
5	P1UP2	R/W	P1 端口 20K 上拉控制寄存器位 1: P1 端口内部 20K 上拉开启 0: P1 端口内部 20K 上拉关闭
4	P0UP2	R/W	P0 端口 20K 上拉控制寄存器位 1: P0 端口内部 20K 上拉开启 0: P0 端口内部 20K 上拉关闭
3	-	-	保留
2	P2UP1	R/W	P2 端口 10K 上拉控制寄存器位 1: P2 端口内部 10K 上拉开启 0: P2 端口内部 10K 上拉关闭
1	P1UP1	R/W	P1 端口 10K 上拉控制寄存器位 1: P1 端口内部 10K 上拉开启 0: P1 端口内部 10K 上拉关闭
0	P0UP1	R/W	P0 端口 10K 上拉控制寄存器位 1: P0 端口内部 10K 上拉开启 0: P0 端口内部 10K 上拉关闭

注：同一端口 10K/20K 上拉同时开启时，二者处于并联状态。

4.7.4.4 P37~P34 端口上拉控制寄存器(P3UPH)

位	名称	读写	说明
7	P37UP2	R/W	P37 端口 20K 上拉控制寄存器位 1: P37 端口内部 20K 上拉开启 0: P37 端口内部 20K 上拉关闭
6	P36UP2	R/W	P36 端口 20K 上拉控制寄存器位 1: P36 端口内部 20K 上拉开启 0: P36 端口内部 20K 上拉关闭
5	P35UP2	R/W	P35 端口 20K 上拉控制寄存器位 1: P35 端口内部 20K 上拉开启 0: P35 端口内部 20K 上拉关闭
4	P34UP2	R/W	P34 端口 20K 上拉控制寄存器位 1: P34 端口内部 20K 上拉开启 0: P34 端口内部 20K 上拉关闭
3	P37UP1	R/W	P37 端口 10K 上拉控制寄存器位 1: P37 端口内部 10K 上拉开启 0: P37 端口内部 10K 上拉关闭
2	P36UP1	R/W	P36 端口 10K 上拉控制寄存器位 1: P36 端口内部 10K 上拉开启 0: P36 端口内部 10K 上拉关闭
1	P35UP1	R/W	P35 端口 10K 上拉控制寄存器位 1: P35 端口内部 10K 上拉开启 0: P35 端口内部 10K 上拉关闭
0	P34UP1	R/W	P34 端口 10K 上拉控制寄存器位 1: P34 端口内部 10K 上拉开启 0: P34 端口内部 10K 上拉关闭

4.7.4.5 P33~P30 端口上拉控制寄存器(P3UPL)

位	名称	读写	说明
7	P33UP2	R/W	P33 端口 20K 上拉控制寄存器位 1: P33 端口内部 20K 上拉开启 0: P33 端口内部 20K 上拉关闭
6	P32UP2	R/W	P32 端口 20K 上拉控制寄存器位 1: P32 端口内部 20K 上拉开启 0: P32 端口内部 20K 上拉关闭
5	P31UP2	R/W	P31 端口 20K 上拉控制寄存器位 1: P31 端口内部 20K 上拉开启 0: P31 端口内部 20K 上拉关闭
4	P30UP2	R/W	P30 端口 20K 上拉控制寄存器位 1: P30 端口内部 20K 上拉开启 0: P30 端口内部 20K 上拉关闭
3	P33UP1	R/W	P33 端口 10K 上拉控制寄存器位 1: P33 端口内部 10K 上拉开启 0: P33 端口内部 10K 上拉关闭
2	P32UP1	R/W	P32 端口 10K 上拉控制寄存器位 1: P32 端口内部 10K 上拉开启 0: P32 端口内部 10K 上拉关闭
1	P31UP1	R/W	P31 端口 10K 上拉控制寄存器位 1: P31 端口内部 10K 上拉开启 0: P31 端口内部 10K 上拉关闭
0	P30UP1	R/W	P30 端口 10K 上拉控制寄存器位 1: P30 端口内部 10K 上拉开启 0: P30 端口内部 10K 上拉关闭

4.7.4.6 P4 端口上拉控制寄存器(P4UP)

位	名称	读写	说明
7	P4up2	R/W	P47~P45 端口 20K 上拉控制位 1: P47~P45 端口内部 20K 上拉开启 0: P47~P45 端口内部 20K 上拉关闭
6	P44UP2	R/W	P44 端口 20K 上拉控制寄存器位 1: P44 端口内部 20K 上拉开启 0: P44 端口内部 20K 上拉关闭
5	P43UP2	R/W	P43 端口 20K 上拉控制寄存器位 1: P43 端口内部 20K 上拉开启 0: P43 端口内部 20K 上拉关闭
4	P42UP2	R/W	P42 端口 20K 上拉控制寄存器位 1: P42 端口内部 20K 上拉开启 0: P42 端口内部 20K 上拉关闭
3	P4UP1	R/W	P47~P45 端口 10K 上拉控制寄存器位 1: P47~P45 端口内部 10K 上拉开启 0: P47~P45 端口内部 10K 上拉关闭
2	P44UP1	R/W	P44 端口 10K 上拉控制寄存器位 1: P44 端口内部 10K 上拉开启 0: P44 端口内部 10K 上拉关闭
1	P43UP1	R/W	P43 端口 10K 上拉控制寄存器位 1: P43 端口内部 10K 上拉开启 0: P43 端口内部 10K 上拉关闭
0	P42UP1	R/W	P42 端口 10K 上拉控制寄存器位 1: P42 端口内部 10K 上拉开启 0: P42 端口内部 10K 上拉关闭

4.7.4.7 端口驱动控制寄存器 (PORTDRV)

位	名称	读写	说明
7~6	P4LEDDR	R/W	P42~P44 端口 LED 驱动功能控制位 00: 关闭 P42~P44 端口 LED 驱动功能 (默认) 01: P42~P44 端口 LED 驱动电流 2mA 10: P42~P44 端口 LED 驱动电流 5mA 11: P42~P44 端口 LED 驱动电流 8mA
5~4	P3LEDDR	R/W	P35~P37 端口 LED 驱动功能控制位 00: 关闭 P33~P35 端口 LED 驱动功能 (默认) 01: P33~P35 端口 LED 驱动电流 2mA 10: P33~P35 端口 LED 驱动电流 5mA 11: P33~P35 端口 LED 驱动电流 8mA
3~2	-	-	保留
1	P4HDRV	R/W	P42~P44 端口输出高驱动能力控制位 0: P42~P44 端口输出低驱动 (默认) 1: P42~P44 端口输出高驱动
0	P3HDRV	R/W	P33~P35 端口输出高驱动能力控制位 0: P33~P35 端口输出低驱动 (默认) 1: P33~P35 端口输出高驱动

注: P3、P4 引脚的 LEDDRV 功能同一时间只能开启一组, 否则可能影响芯片正常工作; STOP 模式下, LEDDRV 功能自动关闭, 唤醒后又自动开启。

4.7.4.8 端口数字输入功能控制寄存器

4.7.4.8.1 P0/P1/P2 数字输入功能控制寄存器 (PORTAN)

位	名称	读写	说明
7~3	-	-	保留
2	P2AN	R/W	P2 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
1	P1 AN	R/W	P1 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
0	P0 AN	R/W	P0 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启

4.7.4.8.2 P3 数字输入功能控制寄存器 (P3AN)

位	名称	读写	说明
7	P37AN	R/W	P37 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
6	P36AN	R/W	P36 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
5	P35AN	R/W	P35 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启

4	P34AN	R/W	P34 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
3	P33AN	R/W	P33 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
2	P32AN	R/W	P32 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
1	P31 AN	R/W	P31 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
0	P30 AN	R/W	P30 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启

4.7.4.8.3 P4 数字输入功能控制寄存器 (P4AN)

位	名称	读写	说明
7	P47AN	R/W	P47 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
6	P46AN	R/W	P46 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
5	P45AN	R/W	P45 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启

4	P44AN	R/W	P44 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
3	P43AN	R/W	P43 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
2	P42AN	R/W	P42 端口数字输入功能控制位 1: 数字输入缓冲关断 0: 数字输入缓冲开启
1~0	-	-	保留

4.7.4.9 端口唤醒控制寄存器 (PORTWK)

位	名称	读写	说明
7~6	-	-	保留
5	usbwk	R/W	P40/P41 端口唤醒使能控制位 1: 开启端口唤醒功能 0: 关闭端口唤醒功能
4	P4WK	R/W	P4 端口唤醒使能控制位 (P40/P41 除外) 1: 开启端口唤醒功能 0: 关闭端口唤醒功能
3	P3WK	R/W	P3 端口唤醒使能控制位 1: 开启端口唤醒功能 0: 关闭端口唤醒功能
2	P2WK	R/W	P2 端口唤醒使能控制位 1: 开启端口唤醒功能 0: 关闭端口唤醒功能
1	P1WK	R/W	P1 端口唤醒使能控制位

			1: 开启端口唤醒功能 0: 关闭端口唤醒功能
0	P0WK	R/W	P0 端口唤醒使能控制位 1: 开启端口唤醒功能 0: 关闭端口唤醒功能

4.7.4.10 端口检测状态标志寄存器 P0EF~P4EF

4.7.4.10.1 Px 端口检测状态寄存器 PxEF

位	名称	读写	说明
7~0	PxEF.n	R	端口检测结果标志寄存器，bit7~bit0 分别对应该组的 7 个端口； 读取该寄存器的值可用于查询相应端口是否发生状态变化。 读： 1: 检测到了相应端口的状态变化（根据该端口上拉电阻的开关设定分为上升沿检测/状态变化检测） 0: 未检测到相应端口的状态变化 注 1: 端口的上拉使能开启时，该端口检测方式为下降沿检测；上拉使能关闭时，端口检测方式为状态变化检测。 注 2: 以下操作会对各状态位进行清零操作： <ul style="list-style-type: none"> ➢ 系统复位（任何复位源产生的系统复位，清除所有状态位）； ➢ 芯片进入 IDLE 或 STOP 模式； ➢ 端口对应的中断标志位清零； ➢ 端口中断使能关闭，且端口唤醒功能关闭。 ➢ 端口复用（端口复用作其他功能时，清除对应端口的状态位）。

4.7.4.11 端口中断寄存器

4.7.4.11.1 端口中断控制寄存器 PORTIF

位	名称	读写	说明
7~5	-	-	保留
4	P4IF	R/W	P40/P41 端口中断标志位 1: P40/P41 端口发生状态变化中断; 0: P40/P41 端口未发生状态变化中断;
3~0	-	-	保留

4.7.4.11.2 P3 端口中断 P3INTF

位	名称	读写	说明
7~0	P3xIF	R/W	P3x 端口中断标志位: 1: P3x 端口产生中断; 0: P3x 端口未产生中断。 硬件无法清除, 软件写“1”无效, 写“0”清 0。

4.7.4.11.3 P3 端口中断 P3INTE

位	名称	读写	说明
7~0	P3xIE	R/W	P3x 端口中断控制位: 1: 允许 P3x 端口产生中断; 0: 禁止 P3x 端口中断。

4.7.4.1 PS2 上拉控制寄存器 PS2UP

位	名称	读写	说明
7~2	-	-	保留
1	PS2UP_PHY	R/W	P40/P41 端口 PS2 上拉控制位 1: 开启 PS2 上拉 0: 关闭 PS2 上拉
0	-	-	保留

4.8 看门狗定时器

4.8.1 概述

看门狗定时器是系统定时器，主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）。

4.8.2 特征

- 32KHZ 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

4.8.3 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

4.8.4 超时周期

超时周期由 **WDTPS** 控制

4.8.5 清零 WDT

发生以下任何条件时，WDT 清零：

- 复位；
- WDTCLR 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）

4.8.6 非工作模式下 WDT 工作原理

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

系统从 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）。

系统从 STOP 模式唤醒，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）。

4.8.7 寄存器映射

寄存器	地址	初始值	描述
WDTCON	0xD1	0x49	WDT 的控制寄存器
WDTCTL	0xD2	0x00	WDT 的 clr 指定寄存器

4.8.8 寄存器说明

4.8.8.1 WDT 控制寄存器 WDTCON

位	名字	读写	描述
7	-	-	保留

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

4.8.8.2 WDT 状态及清除寄存器 WDTCTL

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

4.9 定时器/计数器模块 Timer0~1

4.9.1 概述

SG8F6402 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。

定时器 0 和定时器 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 12 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（P32）或 T1(P33)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。

4.9.1 寄存器映射

寄存器	地址	初始值	说明
TCON	0x88	0x00	Timer0/Timer1 控制寄存器
TMOD	0x89	0x00	Timer0/Timer1 模式控制寄存器
TL0	0x8A	0x00	Timer0 计数器低字节
TH0	0x8B	0x00	Timer0 计数器高字节
TL1	0x8C	0x00	Timer1 计数器低字节
TH1	0x8D	0x00	Timer1 计数器高字节
CKCON	0x8E	0x00	Timer0/1/2 时钟源、控制寄存器

4.9.2 寄存器描述

4.9.2.1 TMOD

位	名称	读/写	描述
7	T1GATE	R/W	Timer1 外部控制开关（INT1）使能位

			<p>1= 开启外部控制开关（P34 用作 INT1 输入脚）。</p> <p>Timer1 外部控制开关开启后，当 TR1（TCON bit6）置 1 时，如果 INT1(P34 引脚)为高电平，Timer1 就会在 T1(P33 端口)的每个下降沿加 1。</p> <p>0= 禁用外部控制开关。Timer1 计数不受 INTO 控制。</p> <p>注：T1GATE=1 时，由 P3INTWK 控制的 P34 端口沿检测功能强制关闭，P34 作为 INT1 的电平/边沿检测功能开启（见 TCON 寄存器）；</p> <p>P34 作为 GPIO 功能仍可正常使用。</p>										
6	T1_T_C	R/W	<p>Timer1 计数器/定时器模式选择位</p> <p>1= timer1 用作计数器；</p> <p>0= timer1 用作定时器。</p>										
5~4	T1_MOD	R/W	<p>Timer1 工作方式控制位</p> <p>具体工作方式见下表：</p> <table border="1" data-bbox="614 1120 1380 1691"> <thead> <tr> <th>T1_MOD</th> <th>功能说明</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>13 位计数器/定时器。由 TL1 寄存器的低 5 位和 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。</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> </tbody> </table>	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 TH1 寄存器的高 8 位组成。TL1 的高 3 位无效（可设置为零）。	01	16 位计数器/定时器。	10	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。	11	定时器 1 不工作。
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 TH1 寄存器的高 8 位组成。TL1 的高 3 位无效（可设置为零）。												
01	16 位计数器/定时器。												
10	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。												
11	定时器 1 不工作。												
3	T0GATE	R/W	<p>Timer0 外部控制开关（INT0）使能位</p> <p>1= 开启外部控制开关（P35 用作 INT0 输入脚）。</p> <p>Timer0 外部控制开关开启后，当 TR0（TCON bit4）置 1 时，如果 INT0(P35 引脚)为高电平，Timer0 就会在 T0(P32 引脚)的每个下降沿加 1。</p>										

			<p>0= 禁用外部控制开关。Timer0 计数不受 INT0 控制。</p> <p>注：T0GATE=1 时，由 P3INTWK 控制的 P35 端口沿检测功能强制关闭，P35 作为 INT0 的电平/边沿检测功能开启（见 TCON 寄存器）；</p> <p>P35 作为 GPIO 功能仍可正常使用。</p>										
2	T0_T_C	R/W	<p>Timer0 计数器/定时器模式选择位</p> <p>1= timer0 用作计数器；</p> <p>0= timer0 用作定时器。</p>										
1~0	T0_MOD	R/W	<p>Timer0 工作方式控制位</p> <p>具体工作方式见下表：</p> <table border="1" data-bbox="614 873 1380 1680"> <thead> <tr> <th>T0_MOD</th> <th>功能说明</th> </tr> </thead> <tbody> <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（可设置为零）。</td> </tr> <tr> <td>01</td> <td>16 位计数器/定时器。</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： 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。</td> </tr> </tbody> </table>	T0_MOD	功能说明	00	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（可设置为零）。	01	16 位计数器/定时器。	10	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。	11	Timer0 的 TH0 和 TL0 作为两个独立的 8 位定时器/计数器工作。此时： 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。
T0_MOD	功能说明												
00	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（可设置为零）。												
01	16 位计数器/定时器。												
10	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。												
11	Timer0 的 TH0 和 TL0 作为两个独立的 8 位定时器/计数器工作。此时： 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。												

4.9.2.2 TCON

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; 也可软件清零。
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; 也可软件清零。
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; 也可软件清零。
2	IT1	R/W	INT1 检测方式选择位 1= 上升沿/下降沿检测(由 CKCON bit3—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。
0	IT0	R/W	INT0 检测方式选择位 1= 上升沿/下降沿检测(由 CKCON bit2—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);
--	--	--	--

4.9.2.3 TH1

位	名称	读/写	描述
7~0	TH1	R/W	Timer1 高字节计数器

4.9.2.4 TL1

位	名称	读/写	描述
7~0	TL1	R/W	Timer1 低字节计数器

4.9.2.5 TH0

位	名称	读/写	描述
7~0	TH0	R/W	Timer0 高字节计数器

4.9.2.6 TL0

位	名称	读/写	描述
7~0	TL0	R/W	Timer0 低字节计数器

4.9.2.7 CKCON

位	名称	读/写	描述
7~4	保留	-	-
3	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
2	INOPL	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
1	保留	-	-
0	保留	-	-

4.10 USB 模块

4.10.1 概述

SG8F6402 的 UDC 模块设计为兼容 Low Speed 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 和端点 2 FIFO 大小为 8byte，端点 1 FIFO 大小为 48byte。

与以往 UDC 不同之处在于，UDC 与内核 SG51 的接口为同步设计，代替了以往的异步锁存器设计，所以 UDC 输入的系统时钟频率应设置为不分频或 2 分频（见分频控制寄存器 OSCHCN）。

4.10.2 寄存器映射

寄存器	地址	初始值	说明
USBCON	0xF8	0x00	USB 控制寄存器
EPT0CTL	0xA1	0x00	端点 0 控制寄存器
EPT1CTL	0xA2	0x00	端点 1 控制寄存器
EPT2CTL	0xA3	0x00	端点 2 控制寄存器
USBADDR	0xA4	0x00	USB 地址寄存器
USBSTA	0xA5	0x00	USB 数据同步控制寄存器
USBINTE	0xA6	0x00	USB 中断允许寄存器
USBINTF	0xA7	0x00	USB 中断状态寄存器
EPT0D	0xF9	0x00	端点 0fifo 数据
EPT1D	0xFA	0x00	端点 1fifo 数据
EPT2D	0xFB	0x00	端点 2fifo 数据
UFCLR	0xFC	0x00	UDC FIFO 控制寄存器

4.10.3 控制寄存器说明

4.10.3.1 USBCON 寄存器

位	名称	读写	说明
7	PHYEN	RW	USB PHY 挂起控制 1 = USB PHY 使能 0 = USB PHY 禁止
6	USB_UP	RW	USB 上拉控制 1 = 使能 USB 上拉电阻 0 = 禁止 USB 上拉电阻
5	TX_EN	RW	USB 输出使能 1: 使能 PHY 输出 0: 禁止 PHY 输出
4	TX_DATA	RW	USB 输出数据 1: DP=1 DM=0 0: DP=0 DM=1
3:2	保留		
1	USBTRIM	RW	UDC 自动调节功能控制位 1: 开启 UDC 时钟自动调节功能 0: 关闭 UDC 时钟自动调节功能
0	UDCCLKEN	RW	UDC 时钟源控制位 1: 开启 UDC 时钟 0: 关闭 UDC 时钟

4.10.3.2 EPT0CTL 寄存器

位	名称	读取	功能	缺省值
7	OBSY	R/W	端点 0 忙 busy,	1'b1

			<p>当 OBSY = 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);</p> <p>当 OBSY = 0 时，端点 0 对各 PID 包响应： SETUP (h) => ACK (d) OUT (h) => DATA (h) => ACK (d) IN (h) => DATA (d) => ACK (h)。</p> <p>(注： d = device, h = host)</p>	
6	ODIR	R/W	<p>总线方向标志位</p> <p>0: Next package Host => Device,;</p> <p>1: Next package Device => Host</p>	1'b0
5	CMD	R/W	<p>Setup 标志位；</p> <p>1: current package 是 Setup 包</p>	1'b0
4	保留	R	—	1'b0
3:0	SRCNT	R/W	<p>端点 0 接收与发送计数器</p> <p>SR CNT 字节数</p> <p>0000: 0 字节</p> <p>0001: 1 字节</p> <p>0010: 2 字节</p> <p>.....</p> <p>0111: 7 字节</p> <p>1000 8 字节</p>	4'h8

注：SR CNT 设置为 0 后，端点 0 将无法接受数据，发送出的数据包为 0 包。

4.10.3.3 EPT1CTL 寄存器

位	名称	读取	功能	缺省值
---	----	----	----	-----

7	1BSY	R/W	<p>端点 1 忙</p> <p>1BSY = 1 表示忙，端点 1 对各 PID 包响应： IN (h) => NACK (d) OUT (h) => NACK (d)</p> <p>1BSY = 0 时，端点 1 对各 PID 包响应： IN (h) => DATA (d) => ACK (h) OUT (h) => DATA (h) => ACK (d)</p> <p>(注：d = device, h = host)</p>	1'b1
6	1DIR	R/W	<p>总线方向标志位</p> <p>0: Next package Host => Device,;</p> <p>1: Next package Device => Host</p>	1'b0
5:0	SRCNT	R/W	<p>端点 1 接收与发送计数器</p> <p>SEND CNT 字节数</p> <p>00000: 0 字节</p> <p>00001: 1 字节</p> <p>00010: 2 字节</p> <p>00011: 3 字节</p> <p>00100: 4 字节</p> <p>00101: 5 字节</p> <p>00110: 6 字节</p> <p>00111: 7 字节</p> <p>01000: 8 字节</p> <p>01001: 9 字节</p> <p>01010: 10 字节</p> <p>01011: 11 字节</p>	6'h0

			01100:	12 字节	
			01101:	13 字节	
			01110:	14 字节	
			.		
			.		
			.		
			101111:	47 字节	
			11000:	48 字节	

注：SR CNT 设置为 0 后，端点 1 将无法接受数据，发送出的数据包为 0 包。

4.10.3.4 EPT2CTL 寄存器

位	名称	读取	功能	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 0: Next package Host => Device,; 1: Next package Device => Host	1'b0
5:4	——	——	——	2'b0
3:0	SR CNT	R/W	端点 2 接收与发送计数器 SEND CNT 字节数	4'h0

			0000:	0 字节	
			0001:	1 字节	
			0010:	2 字节	
				
			0111:	7 字节	
			1000:	8 字节	

注：SR CNT 设置为 0 后，端点 2 将无法接受数据，发送出的数据包为 0 包。

4.10.3.5 USBADDR——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

4.10.3.6 USBSTA——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 清楚数据同步，下一个数据包 PID 设置为 DATA0	1'b0
6	EPT2 TCLEAR	R/W	端点 2 清楚数据同步，下一个数据包 PID 设置为 DATA0	1'b0
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；	1'h0

			设置为 0 时 Endpoint0 为正常状态。 (注: 端点 0 硬件会自动清此设置位)	
2	ZPACK2	R	为 1 时, UDC 发送一个为 0 的数据包	1'b1
1	ZPACK1	R	为 1 时, UDC 发送一个为 0 的数据包	1'b1
0	ZPACK0	R	为 1 时, UDC 发送一个为 0 的数据包	1'b0

注: ZPACK*为只读, 当相应 SRCNT 寄存器设置为 0 时, 该位为 1。

4.10.3.7 USBINTE——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	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

4.10.3.8 USBINTF——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;	1'b0

			写 1 清零; (属于传输异常中断)	
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

4.10.3.9 EPT0D——端点 0 FIFO 数据

位	名称	读取	功能	缺省值
7:0	EPT0D	R/W	端点 0 FIFO 数据, 深度 8byte	8'bx

4.10.3.10 EPT1D——端点 1 FIFO 数据

位	名称	读取	功能	缺省值
7:0	EPT1D	R/W	端点 1 FIFO 数据, 深度 48byte	8'bx

4.10.3.11 EPT2D——端点 2 FIFO 数据

位	名称	读取	功能	缺省值
7:0	EPT2D	R/W	端点 2 FIFO 数据, 深度 8byte	8'bx

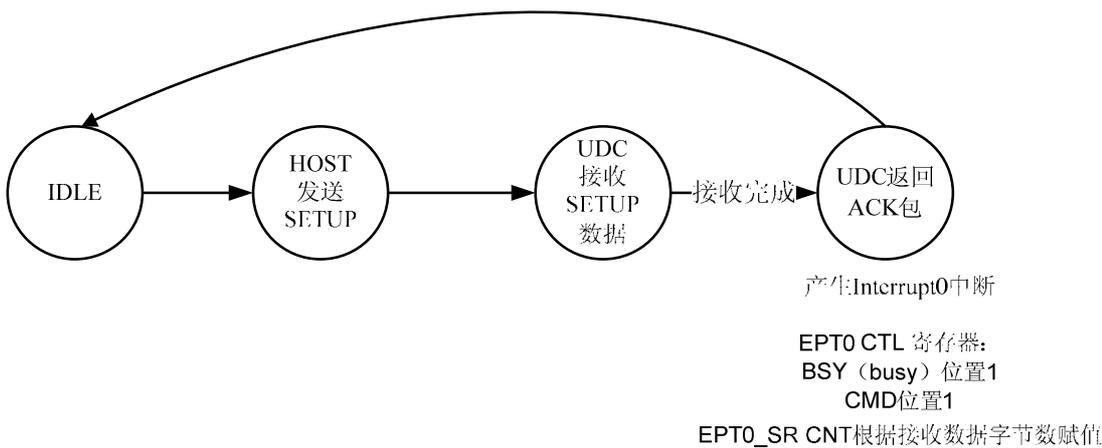
4.10.3.12 UFCLR——UDC FIFO 控制寄存器

位	名称	读取	功能	缺省值
7:3	——	R	——	5'b0
2	EPT2_FIFO_CLR	R/W	写入 1 则端点 2 的 FIFO 读写指针归 0, 该位自动清 0	1'b0
1	EPT1_FIFO_CLR	R/W	写入 1 则端点 1 的 FIFO 读写指针归 0, 该位自动清 0	1'b0
0	EPT0_FIFO_CLR	R/W	写入 1 则端点 0 的 FIFO 读写指针归 0, 该位自动清 0	1'b0

4.10.4 UDC 通信流程说明

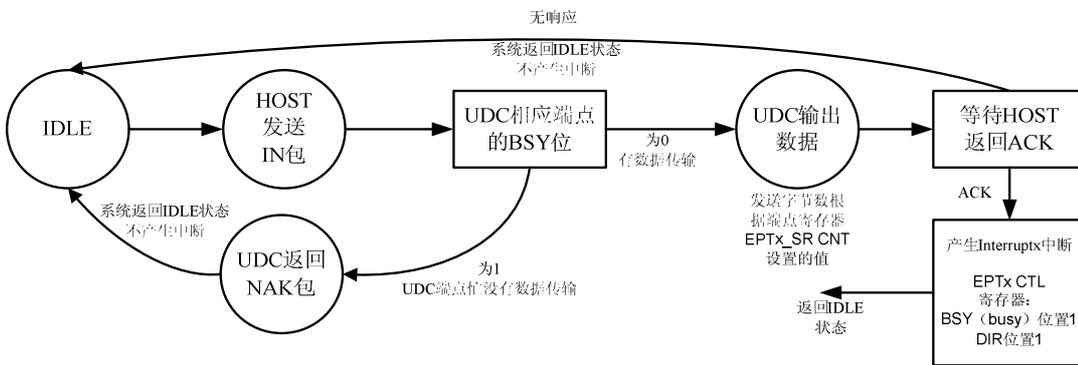
UDC 与 host 之间通过 3 种通讯包通讯，分别为：SETUP 包，中断 IN 包和中断 OUT 包。下面对各包的 UDC 的响应及相应操作作详细说明。

4.10.4.1 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。

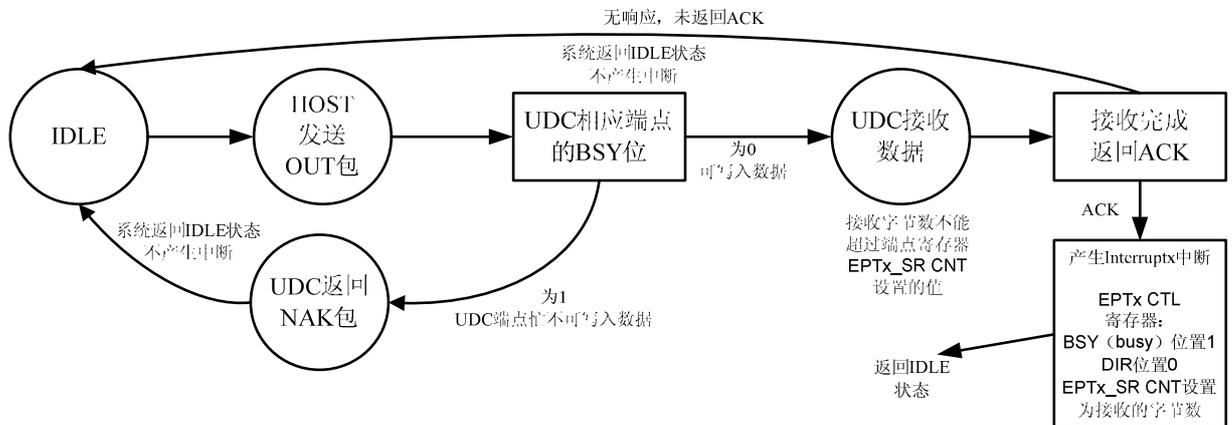
4.10.4.2 中断传输 IN 包处理说明



一旦相应端点的 BSY 位置为 0，UDC 在 IN 包到来时，立即将 FIFO 中的数据发出，不会去判断 FIFO 是否为空。

所以 MCU 响应 IN 包时，应该先将要发送的数据写入相应的端点 FIFO，然后再将相应的 BSY 位置 0。

4.10.4.3 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；则不会产生端点 Interruptx 中断，但会产生 Interrupt Error 中断。

4.10.5 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	CRC8	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 8byte 数据装入 FIFO，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，并将需要发送的数据写入 FIFO。设备在接收到主机的 IN 包后，发送 FIFO 中数据。

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，接收到正确的数据长度的后，会发送 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

4.11 FLASH 控制器

4.11.1 程序存储器功能说明

SG51 的单指令宽度是 8 位的，而 Flash 每个地址的数据宽度为 16 位，所以每次操作 Flash 时会预取出两个指令，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 SG51 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，从而暂停 SG51 的执行指令；写操作结束后，指令有效信号 INS_EN 置 1，此时 SG51 可以继续执行指令。

4.11.2 FLASH 存储器编程

对 FLASH 存储器编程的最简单的方法是使用供应商提供的编程工具，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应用系统需要使能片内 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。

4.11.2.1 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。

4.11.2.2 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 密钥码；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 写第一个密钥码：0xA5。
3. 向 FLKEY 写第二个密钥码：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)。

4.11.2.3 FLASH 写：

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。

4.11.3 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。

4.11.4 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，CIP-51 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 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。

4.11.5 寄存器映射

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

4.11.6 寄存器说明

4.11.6.1 PFCTL: FLASH 读写控制寄存器

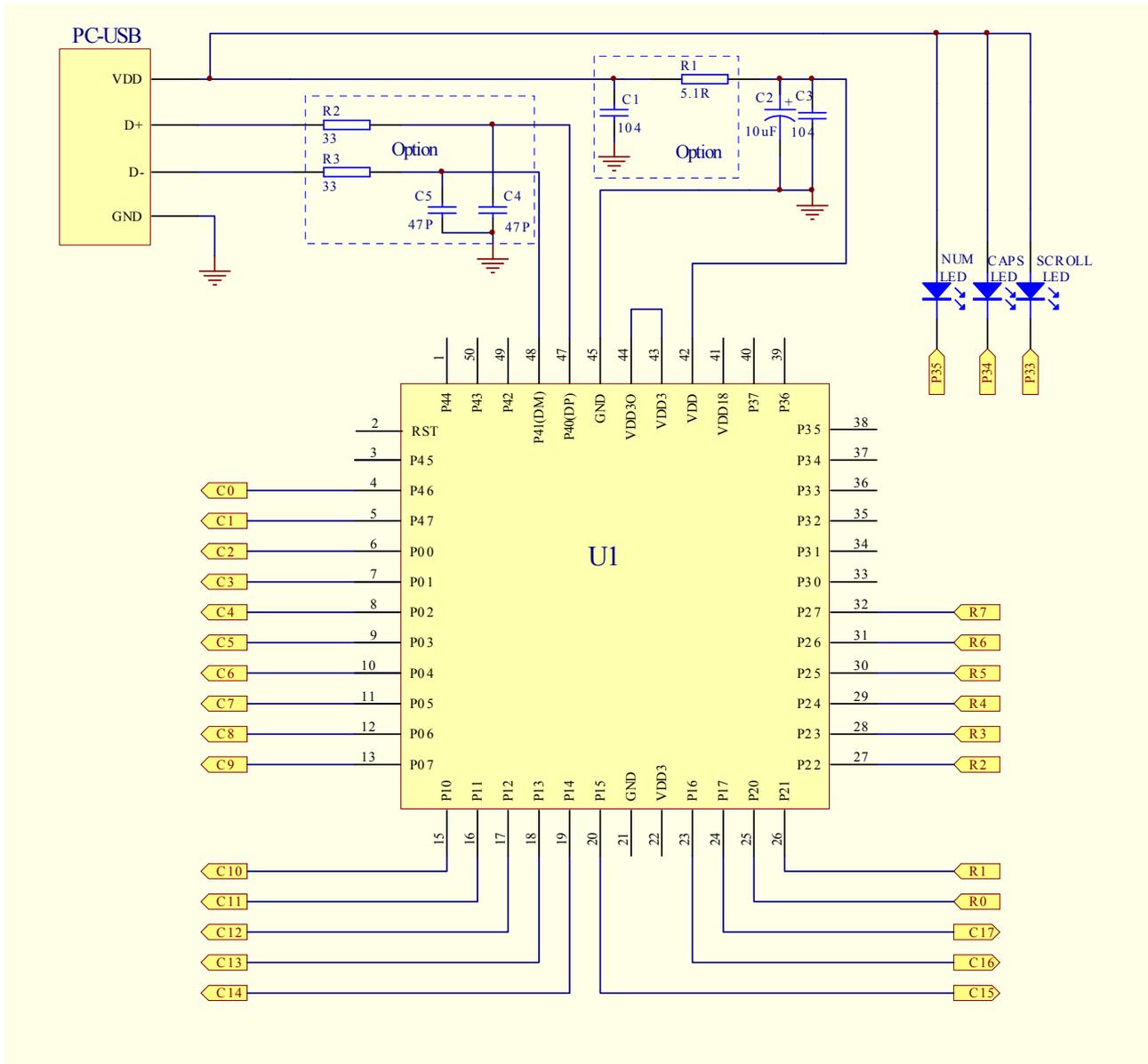
位	名字	读写	初始值	描述								
7	FOSE	R/W	0	<p>FOSE: FLASH 单稳态定时器使能</p> <p>该位用于使能 FLASH 读操作单稳态定时器。当 FLASH 单稳态定时器被禁止时, FLASH 读出放大器在整个 FLASH 读周期内被使能。当系统时钟频率低于 12MHz 时, 禁止 FLASH 单稳态定时器会增加系统功耗。</p> <p>0: 禁止 FLASH 单稳态定时器。</p> <p>1: 使能 FLASH 单稳态定时器。</p>								
6~4	FREQ	R/W	3'b011	<p>FREQ:</p> <table style="margin-left: 20px;"> <tr><td>011</td><td>12Mhz</td></tr> <tr><td>010</td><td>6Mhz</td></tr> <tr><td>001</td><td>3MHz</td></tr> <tr><td>000</td><td>1.5MHz</td></tr> </table> <p>该寄存器仅仅指明当前系统时钟, 不能设置 Flash 控制器的时钟, Flash 控制器时钟采用系统时钟。Flash 控制器使用该寄存器来调整内部计数器延时周期数。</p>	011	12Mhz	010	6Mhz	001	3MHz	000	1.5MHz
011	12Mhz											
010	6Mhz											
001	3MHz											
000	1.5MHz											
3	FLBWE	R/W	1'b0	<p>FLBWE: FLASH 块写使能位。</p> <p>该位控制软件对 FLASH 存储器的块写操作。</p> <p>0: 软件 FLASH 写操作的每个字节都被单独写入。</p> <p>1: FLASH 字节按 2 字节为一组写入。</p>								
2	-	R	1'b0	保留								
1	PSEE	R/W	1'b0	<p>PSEE: 程序存储擦除允许</p> <p>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。</p> <p>0: 禁止擦除 FLASH 存储器。</p> <p>1: 允许擦除 FLASH 存储器。</p>								
0	PSWE	R/W	1'b0	<p>PSWE: 程序存储写允许</p> <p>将该位置'1'后允许用 MOVX 指令向 FLASH 存储器写</p>								

				<p>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。</p>
--	--	--	--	--

4.11.6.2 FLKEY: 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	<p>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。</p>

5 应用参考原理图



6 极限参数

标识符	描述	最小	最大	单位
T _{OPR}	工作温度	0	70	°C
T _{STG}	储存温度	-50	85	°C
V _I	输入电平范围	-0.5	6.0	V
V _O	输出电压	-0.5	6.0	V
V _{CC}	电源电压	-0.5	5.5	V

7 电气参数

(T=25°C, Vdd=5V, Vss=0V)

符号	参数	测试环境	最小	典型	最大	单位
3.3V Regulator						
V_{reg}	Output voltage of 3.3v Regulator	Vdd=4.5V~5.5V	3.0	3.3	3.6	V
V_{reset}	Low Power Reset detecting the Voltage		1.0	1.6	2.4	V
MCU operation						
I_{il}	Input Leakage Current for input pins	VIN=VDD3, VSS	-	-	±1	uA
I_{cc}	VDD operating supply current Normal frequency operation mode	Clock type Freq = 12MHz Output pins floating	-	3.3	5	mA
I_{sb1}	Operating supply current 1-idle mode	All input and I/O pins at VDD3 Output pins floating WDT enabled	-	1	-	mA
I_{sb2}	Operating supply current 2-Sleep mode	All input and I/O pins at VDD3 Output pins floating WDT enable	-	200	300	uA
GPIO Pins						
V_{ih}	Input High Voltage	Port0&Port1&Port2&Port3	-	3.3	3.6	V
V_{il}	Input Low Voltage	Port0&Port2&Port3	-	-	0.7	V
		Port1	-	-	0.7	V

	Output High Voltage (P00~P07, P10~P17, P20~P27, P30~P32, P36~P37, P45~P47)	Isink = 2mA VDD3= 3.3V	2.7	-	3.6	V
V _{oh1}	Output High Voltage-Low Drive (P33~P35, P42~P44)	Isink = 2mA VDD3= 3.3V	2.7	-	3.6	V
V _{ol1}	Output Low Voltage (P00~P07,P10~P17, P20~P27, P30~P32, P36~P37, P45~P47)	Isink = 2mA VDD3= 3.3V	0	-	0.4	V
V _{ol2}	Output Low Voltage-Low Drive (P33~P35, P42~P44)	Isink = 2.5mA VDD3= 3.3V	0	-	0.4	V
V _{ol3}	Output Low Voltage-High Drive (P33~P35, P42~P44)	Isink = 20mA Vdd = 5V	0	-	0.4	V
I _{LED1}	Constant LED current output (P33~P35, P42~P44)	PXLEDDRV=01	-	2	-	mA
I _{LED2}	Constant LED current output (P33~P35, P42~P44)	PXLEDDRV=10	-	5	-	mA
I _{LED3}	Constant LED current output (P33~P35, P42~P44)	PXLEDDRV=11	-	8	-	mA
V _{LED}	LED Voltage Drop		1.5	-	3.6	V

R_{PH1}	Pull-high resistor (P00~P07, P10~P17, P20~P27, P30~P32, P36~P37, P45 ~P47)	PXUP1=1 PXUP2=0 Input pin with pull-high resistor	7	10	14	K Ω
R_{PH2}	Pull-high resistor (P00~P07, P10~P17,P20~P27, P30~P32,P36~P37, P45 ~P47)	PXUP1=1 PXUP2=0 Input pin with pull-high resistor	14	20	29	K Ω
USB Interface						
V_{oh}	Static Output High	USB operation Mode	2.8	-	3.6	V
V_{ol}	Static Output Low		-	-	0.3	V
V_{di}	Differential Input Sensitivity		0.2	-	-	V
V_{cm}	Differential Input Command Mode Range		0.8	-	2.5	V
V_{se}	Single Ended Receiver Threshold		0.8	-	2.0	V
C_{in}	Transceiver Capacitance		-	-	20	PF
V_{rg}	Output Voltage of internal Regulator		3.0	-	3.6	V

8 修订记录

版本	日期	修订内容	修订者
V1.0	2015-08-26	初始版本	刘星
V1.1	2016-11-24	更新 IC 工作电压范围	刘星