



SUNWAY 申威

BMC 寄存器手册

2015 年 1 月

成都申威科技有限责任公司



免责声明

本文档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因文档使用不当造成的直接或间接损失，本公司不承担任何责任。

成都申威科技有限责任公司

Chengdu Sunway Technology Corporation Limited

地址：成都市华府大道四段电子科大科技园 D22 栋

Building D22, National University Science and technology park,

Section 4, Huafu Avenue, Chengdu

Mail: sales@swcpu.cn

Tel : 028-68769016

Fax: 028-68769019



阅读指南

《BMC 寄存器手册》主要描述了 BMC 卡内部地址空间分配原则、版本差异说明、内部寄存器详细介绍等内容。

文档修订

文档更新记录	文档名	BMC 寄存器手册
	版本号	V1.2
	创建人	研发部
	创建日期	2014-02-26

版本更新

版本号	更新内容	更新日期
V1.0	v1.0 罗列出了多个版本的差异，以及具体实现的模块中的寄存器	2014-02-26
V1.1	在 eeb2f 的版本中添加了 wdt 模块和在 rst_pwr 模块中加入了版本信息寄存器	2014-03-25
V1.2	重新规划 openrisc 的内部地址分配，统一添加看门狗模块和蜂鸣器模块	2014-09-10

技术支持

可通过邮箱或问题反馈网站向我司提交产品使用的问题，并获取技术支持。

售后服务邮箱：sales@swcpu.cn

问题反馈网址：<http://www.swcpu.cn/>

目 录

1	BMC 内部 open risc 地址空间分配.....	1
2	不同版本的差异.....	2
3	模块内部寄存器.....	3
3.1	spi 模块.....	3
3.2	xbus 模块.....	3
3.3	ipmi 模块.....	4
3.4	ps2 模块.....	8
3.5	iic 模块.....	8
3.6	gpio 模块.....	8
3.7	fan_ctrl 模块.....	9
3.8	电源管理 (pm) 和复位模块.....	10
3.9	维护 (cpu_mt) 模块.....	11
3.10	uart 模块.....	14
3.11	uart1 模块.....	14
3.12	wdt 模块.....	14

1 BMC 内部 openrisc 地址空间分配

BMC 内部包含 32 位的 openrisc 处理器，在具体实现时地址空间分配如下表：

模块名	模块描述	地址
spi	访问 spi flash 的接口	0000-0000~0FFF-FFFF
xbus	外部扩展接口	1000-0000~1FFF-FFFF
ipmi	智能平台管理接口	2000-0000~2FFF-FFFF
ps2	ps2 鼠标键盘接口	3000-0000~3FFF-FFFF
iic	i2c 接口	4000-0000~4FFF-FFFF
gpio	通用 io 口控制模块	5000-0000~5FFF-FFFF
fanctrl	风扇控制模块	6000-0000~6FFF-FFFF
pm	电源管理和复位模块	7000-0000~7FFF-FFFF
cpu_mt	CPU 维护模块	8000-0000~8FFF-FFFF
Uart0	串口模块 0	9000-0000~9FFF-FFFF
uart1	串口模块 1	A000-0000~AFFF-FFFF
wdt	看门狗模块	B000-0000~BFFF-FFFF
X	主板差异性的设计模块	C000-0000~CFFF-FFFF
Beep	蜂鸣器模块	D000-0000~DFFF-FFFF
保留		E000-0000~FFFF-FFFF

2 不同版本的差异

不同的版本因需求的不同,在具体实现的时候的也存在差异,集中体现在实现的功能模块上,详细见下表:

版本	例化的模块
MICROATX	Spi、 Xbus、 ps2、 Iic、 Gpio、 Fanctrl、 Pm、 Cpu_mt、 Uart0、 uart1

3 模块内部寄存器

3.1 spi 模块

Spi 模块在 BMC 内部的基址为 0x00000000，模块内部包括两个 32 位的寄存器，地址的读写在实现时使用不同的寄存器，说明如下：

【偏移 0x00e0_0000】寄存器读操作时每位定义

BIT	R/W	Name	Function
0	R	tx_wrreq	
7: 1	R	7'b0	保留
15: 8	R	rsp_ptr	
23: 16	R	cmd_ptr	
31: 24	R	tx_data[7:0]	

【偏移 0x00e0_0004】寄存器读操作时每位定义

BIT	R/W	Name	Function
7: 0	R	tx_data[7:0]	
15: 8	R	rsp_ptr	
23: 16	R	cmd_ptr	
24	R	tx_wrreq	
31: 25	R	7'b0	保留

【偏移 0x00e0_0000/4】寄存器写操作时每位定义

BIT	R/W	Name	Function
23: 0	--	--	
31: 24	W	wb_wrdat	

3.2 xbus 模块

Xbus 模块在 BMC 内部的基址为 0x10000000（32 所的基址是分别为 0xe0000000 和 0xf0000000），内部地址分配如下：

Xbus-SRAM	1000-0000~10FF-FFFF
Xbus-DM9000	1100-0000~11FF-FFFF
保留	1200-0000~13FF-FFFF

模块定义了一个配置寄存器，寄存器为 4 路独立的 cs_timing，说明如下：

【偏移 400_0000 cs_timing】寄存器位定义（复位值 0x00_10_01_11）

BIT	R/W	Name	Function
7: 0	R/W	cs0_cfg	配置 sram 的时序
15: 8	R/W	Cs1_cfg	配置 dm9000 的时序
23: 16	R/W	Cs2_cfg	保留
31: 24	R/W	Cs3_cfg	保留

cs_timing 定义如下:

BIT	R/W	Function
3: 0	R/W	保留
4	R/W	0: i/o 访问; 1: mem 访问
5	R/W	0: 16 位访问; 1: 8 位访问
7: 6	R/W	保留

3.3 ipmi 模块

智能平台管理接口 ipmi 是为 BMC 升级做的准备, 在 BMC 内部的基址为 0x20000000, 内部包括 bt (块传输, 256 字节) 和 kcs (字节传输) 两个模块, 目前此接口用于 BMC 和主 CPU 通讯。

1) BT 模块:

Table 11-1, BT Interface Registers

Offset	Read	Write
0	BT_CTRL - control register	
1	BMC2HOST buffer	HOST2BMC buffer
2	BT_INTMASK - interrupt mask register	

编址如下, 注此处的编址只是为模块内偏移地址, 基址为交叉开关路由使用, 统一分配:

寄存器名称	CPU 地址 (offset)	BMC 地址 (offset)	CPU 权 限	BMC 权 限	复位值
BT_CTRL_R	16'h380	8'hE4	R/W	R/W	8'h80
BMC2HOST_R	16'h384	8'hE5	RO	WO	8'h00
HOST2BMC_R	16'h384	8'hE5	WO	RO	8'h00
BT_INTMASK_R	16'h388	8'hE6	R/W	—	8'h00

B T_CTRL_R[7:0]在 BMC 内部对应的地址为 2100_0000+8'hE4(offset)

主 CPU 访问 B T_CTRL_R[7:0]的地址: 主 CPU 分配的 pci 的基址+16'h380(offset)

说明: BT 控制寄存器, 通过对本寄存器的设置来进行 BT 接口的控制。各位含义如下

BIT	R/W	Name	Function
0	R/W	CLR_WR_PTR	清除写指针, Host 对该寄存器写 1, 则将 HOST2BMC 的写指针移到缓冲起始地址。BMC 对该寄存器写 1, 则将 BMC2HOST 的写指针移到缓冲起始地址。该寄存器物理不存在, 读返回 0。
1	R/W	CLR_RD_PTR	清除读指针, Host 对该寄存器写 1, 则将 BMC2HOST 的读指针移到缓冲起始地址。BMC 对该寄存器写 1, 则将 HOST2BMC 的读指针移到缓冲起始地址。

			该寄存器物理不存在，读返回值 0。
2	R/W	H2B_ATN	HOST 填充完 HOST2BMC 缓冲后，通知 BMC 来取数据，用来作中断输出。
3	R/W	B2H_ATN	BMC 填充完 BMC2 HOST 缓冲后，通知 Host 来取数据，用来作中断输出。
4	R/W	EVT_ATN	当 BMC 有 IPMI 消息请求要发给 HOST 时，使能该位，通过 B2HI_EN 来决定是否生成中断给 HOST。
5	R/W	OEM0	可用来生成 HOST 给 BMC 的中断，暂时保留。
6	R/W	H_BUSY	“1”有效，有效时表示 HOST 正在处理数据，不能接收新的数据。
7	R/W	B_BUSY	复位为 1，当 BMC 完成初始化后清除该位。“1”有效，有效时表示 BMC 正在处理数据，不能接收新的数据。

BT_INTMASK_R 在 BMC 内部对应的地址为 2100_0000+8'hE6(offset)

主 CPU 访问 BT_INTMASK_R 的地址：主 CPU 分配的 pci 的 bar0 基址+16'h388(offset)

说明：此寄存器用来有 Host 控制，BMC 可以生成哪些中断给 Host。格式如下：

BIT	R/W	Name	Function
0	R/W	B2H_IRQ_EN	BMC 向 HOST 的中断使能信号，“1”有效
1	R/W	B2H_IRQ	BMC 向 HOST 的中断有效信号，可作为中断输出信号，“1”有效。边沿置 1，写 1 清。
4: 2	R/W	OEM1 OEM2 OEM3	Reserved for definition by platform manufacturer for BIOS/SMI Handler use. Generic IPMI software must write this bit as 0, and ignore the value on read.
6: 5	R/W	rsvd	保留
7	R/W	BMC_HWRST	(可选) Host 对该地址写，会触发 BMC 硬复位，读返回 0。

Host 到 bmc 的流程：

- BMC 在初始化的最后清 B_BUSY，BMC 通过写入 210000e4 地址 0x80 实现；

- CPU 等待 B_BUSY=0 且 H2B_ATN=0, 此时 BMC 处于等待 CPU 给其中断状态;
- CPU 清写指针 CLR_WR_PTR, CPU 通过写入 BT_CTRL_R 地址 0x01 实现, BMC 延续上一状态;
- CPU 往 HOST2BMC_BUFFER 写 1 到 n 个数据, BMC 延续上一状态;
- CPU 置 H2B_ATN, CPU 通过写入 BT_CTRL_R 地址 0x04 实现, BMC 延续上一状态;
- BMC 检测到 H2B_ATN=1 后置 B_BUSY, BMC 通过写入 210000e4 地址 0x80 实现;
- BMC 清 H2B_ATN, BMC 通过写入 210000e4 地址 0x04 实现;
- BMC 清读指针, BMC 写入 210000e4 地址 0x02;
- BMC 读 HOST2BMC_BUFFER, BMC 通过 256 次 210000e5 地址实现;
- BMC 清 B_BUSY, BMC 通过写入 210000e4 地址 0x80 实现;
- BMC 处理命令。

bmc 到 Host 的流程:

- CPU 等待 B2H_ATN=1, BMC 等待 H_BUSY=0;
- CPU 打开中断使能, CPU 写入 BT_INTMASK_R 地址 0x01;
- BMC 清写指针, BMC 写入 210000e4 地址 0x01;
- BMC 往 BMC2HOST_BUFFER 写 1 到 n 个数据, BMC 通过写 1 到 n 次数据到 210000e5 实现;
- BMC 置 B2H_ATN, BMC 通过写入 210000e4 地址 0x08 实现;
- CPU 置 H_BUSY, CPU 通过写入 BT_CTRL_R 地址 0x40;
- CPU 判断是 B2H_ATN 还是 EVT_ATN, CPU 通过读 BT_CTRL_R 地址实现, 如果是 B2H_ATN 进入步骤 h, 如果是 EVT_ATN 暂时不知道处理流程。
- CPU 清 B2H_ATN, CPU 通过写入 BT_CTRL_R 地址 0x08 实现;
- CPU 清读指针, CPU 写 BT_CTRL_R 地址 0x02 实现;
- CPU 读 n 次 BMC2HOST_BUFFER;
- CPU 清 H_BUSY, CPU 通过写入 BT_CTRL_R 地址 0x40;

2) kcs 接口

Figure 9-5, KCS Interface Registers

	7	6	5	4	3	2	1	0	I/O address
Status (ro)	S1	S0	OEM 2	OEM 1	C/D#	SMS_ATN	IBF	OBF	base+1
Command (wo)									base+1
Data_Out (ro)									base+0
Data_In (wo)									base+0

Reserved bits must be written as '0' and ignored during reads. Software should not assume that reserved bits return a constant value.

编址如下, 注此处的编址只是为模块内偏移地址, 基址为交叉开关路由使用, 统一分配:

寄存器名称	主 CPU 地址 (offset)	BMC 地址 (offset)	CPU 权 限	BMC 权 限	复位 值
-------	----------------------	--------------------	------------	------------	---------

Data_In_R	12'h350	12'hCA2	WO	RO	8'h00
Data_Out_R	12'h350	12'hCA2	RO	WO	8'h00
Status_R	12'h354	12'hCA0	RO	W/R	8'h00
Command_R	12'h354	12'hCA3	WO	RO	8'h00

注：BMC 内部对应的基址为 2000_0000，主 CPU 访问的基址为 pci 的 bar0 基址。

Status_R 的命令寄存器说明：

BIT	R/W	Name	Function
0	R/W	OBF	输出数据有效标志
1	R/W	IBF	输入数据有效标志
2	R/W	SMS_ATN	表明 BMC 有事务需要系统软件处理。
3	R/W	C_D_n	表明最后写入的是 Command_R 还是 Data_In_R，“1”表示 Command_R。
5: 4	R/W	OEM	保留
7: 6	R/W	S1 : S0	KCS 读写状态机状态

S1: S0 状态定义如下表：

S1 (bit 7)	S0 (bit 6)	Definition
0	0	IDLE_STATE. Interface is idle. System software should not be expecting nor sending any data.
0	1	READ_STATE. BMC is transferring a packet to system software. System software should be in the “Read Message” state.
1	0	WRITE_STATE. BMC is receiving a packet from system software. System software should be writing a command to the BMC.
1	1	ERROR_STATE. BMC has detected a protocol violation at the interface level, or the transfer has been aborted. System software can either use the “Get_Status” control code to request the nature of the error, or it can just retry the command.

Command_R 的命令寄存器说明：

KCS Interface Control Codes				
Code	Name	Description	Target register	Output Data Register
60h	GET_STATUS / ABORT	Request Interface Status / Abort Current operation	Command	Status Code
61h	WRITE_START	Write the First byte of an Write Transfer	Command	N/A.
62h	WRITE_END	Write the Last byte of an Write Transfer	Command	N/A
68h	READ	Request the next data byte (该值不写入 command, 而写入 Data_In 寄存器)	Data_In	Next byte

Data_In_R:

说明：数据输入寄存器

功能：写入后，IBF 有效。读出后，IBF 清 0。不支持读写同时发生（协议保证不会出现此现象）。如果对该地址写，而 IBF=1，则报错。如果对该地址读，而 IBF=0，则报错。

Data_Out_R:

说明：数据输出寄存器

读出条件：Data_Out_CPU_REN &&OBF。

功能：写入后，OBF 有效。读出后，OBF 清 0。不支持读写同时发生（协议保证不会出现此现象）。如果对该地址写，而 OBF=1，则报错。如果对该地址读，而 OBF=0，则报错。

3.4 ps2 模块

ps2 在 BMC 内部的基址为 0x30000000 (32 所的基址是 0x10000000)，包括 60h 和 64h 两个接口，说明如下表：

主 CPU 地址	BMC 内部地址	位宽	属性	功能说明
Pci 的基址+60H	0x3000_0060	8	RW	数据缓冲寄存器
Pci 的基址+64H	0x3000_0064	8	RW	状态和控制寄存器

3.5 iic 模块

iic 在 BMC 内部的基址为 0x40000000，模块内寄存器编址如下表：

寄存器名	BMC 内部地址	位宽	属性	功能说明
PRERlo	0x4000_0000	8	RW	时钟 Prescale 低 8 位
PRERhi	0x4000_0004	8	RW	时钟 Prescale 高 8 位
CTR	0x4000_0008	8	RW	控制寄存器
TXR	0x4000_000C	8	W	传输寄存器
RXR	0x4000_000C	8	R	接收寄存器
CR	0x4000_0010	8	W	命令寄存器
SR	0x4000_0010	8	R	状态寄存器
SLADR	0x4000_001C	8	RW	做从设备时的地址

3.6 gpio 模块

gpio 模块在 BMC 内部的基址为 0x50000000，模块内部包括 4 个 32 位的寄存器，寄存器设置如下：

寄存器偏移 (offset)	BMC Read	BMC Write
0x0	GPIO_O	
0x4	GPIO_OE_N	
0x8	GPIO_I	清 GPIO_O
0xC		置 GPIO_O

【偏移 0x00 GPIO_O】寄存器位定义（复位值 32'b0_10_10010_111_1000）

BIT	R/W	Name	Function
3: 0	R/W	TEST_OUT[3: 0]_H	
8: 4	R/W	CFG_CORE[4:0]_H	根据申威 1610 处理器硬件接口手册中表 4-2 设置核心工作频率，一般应设置为非全“0”值（全“0”值为 PLL 旁路方式，用于申威 1610 处理器的测试）。
10:9	R/W	INIT_MODE[1:0]_H	应设置值为“11”，其它设置值用于申威 1610 处理器的测试。
11	R/W	保留	[4.7K 电阻接地]
15:12	R/W	CFG_MM[3:0]_H	根据申威 1610 处理器硬件接口手册中表 4-3 设置存储控

			制器工作频率，一般应设置为非全“0”值（全“0”值为 PLL 旁路方式，用于申威 1610 处理器的测试）
23:16	R/W	保留	保留

【偏移 0x04 GPIO_OE_N】寄存器位定义（复位值 32'hfff_C000）

BIT	R/W	Name	Function
7: 0	R/W	GPIO_OE_N[7:0]	输入输出使能，0 表示输出，1 表示输入
15: 8	R/W	GPIO_OE_N[15:8]	
23: 16	R/W	GPIO_OE_N[23:16]	
31: 24	R/W	GPIO_OE_N[31:24]	

对于偏移 0x08、0x0C，模块内部是这样实现的：对应同样的地址空间，读和写操作对应不同的两个寄存器。写操作时 GPIO_O 的 32 位都是写 1 清和写 1 置。

3.7 fan_ctrl 模块

Fan_ctrl 模块在 BMC 内部的基址为 0x60000000，模块内部包括 5 个 32 位的寄存器寄存器，说明如下：

FAN	BMC 内部地址	位宽	属性	功能说明
FAN0_DutyCycle/RPM	0x6000-0000	32	RW,0x7F	风扇 0 的运行周期任务占空比
FAN1_DutyCycle/RPM	0x6000-0004	32	RW,0x7F	风扇 1 的运行周期任务占空比
FAN2_DutyCycle/RPM	0x6000-0008	32	RW,0x7F	风扇 2 的运行周期任务占空比
FAN3_DutyCycle/RPM	0x6000-000C	32	RW,0x7F	风扇 3 的运行周期任务占空比
pre_div	0x6000-0010	32	RW,0x03	预扫描频率分频器系数

FANX_DutyCycle 寄存器域描述如下：

名称	范围	类型	描述
—	[31:8]	—	保留。
DutyCycle	[7:0]	RW	脉冲有效比例

描述：修改 pwm 波的占空比，00 时占空比最小，对应风扇的转速最慢，ff 时占空比最大，对应风扇的转速最快。

pre_div 寄存器域描述如下：

名称	范围	类型	描述
—	[31:8]	—	保留。

pre_div	[7:0]	RW	预扫描频率分频器系数
---------	-------	----	------------

EEB 升级版中在 pci 桥的通路上做了对 fan_ctrl 模块的地址译码，可作为主 cpu 访问的地址，具体实现的地址为 pci 的 bar1 的 io 基址+0x340（基址 0000_0340），所以 FANx_DutyCycle/RPM 的地址为 0000_0340+x 乘 4 得到，pre_div 的地址为 0000_0240。

3.8 电源管理（pm）和复位模块

电源管理和复位模块在 BMC 内部的基址为 0x70000000，模块内部包括 32 位的寄存器 GPIO、80H 和 FPGA_VERSION，寄存器说明如下：

【偏移 0x00 GPIO】寄存器位定义（复位值 0x00_00_3f_06）

BIT	R/W	Name	Function
3: 0	R/W	rst_type	定义复位类型
6: 4	RO	reserved	复位值为 000
7	RO	id_wdts	看门狗复位，高有效（此位目前只针对 eeb2f，其他版本仍为保留值 0）
8	RO	id_sftrst	PCIE 软复位，软件对 80 端口写 0xC3 该位置 1 为有效位
9	RO	id_wkrst	面板复位控制位，按下面板复位按钮置 1 为有效位
10	RO	id_brdrst	BMC 常备电复位控制位，高有效
11	RO	CPU_PG	CPU 电源好的标志位，高有效
12	RO	CPU_RST_N	软件复位 pcie 桥标志，0：正在复位，1：结束复位
31:13	RO	保留	保留

rst_type 定义如下：

复位类型	复位期间值	复位结束后值
按钮复位（板载/面板）	1001	0001
软复位（南桥/80 端口）	1010	0010
上电复位	1011	0011
板载按钮待机复位	1100	0100
看门狗复位	1101	0101

【偏移 0x04 PORT_80H】寄存器位定义（复位值 0x55_aa_2f_01）

BIT	R/W	Name	Function
7: 0	R/W	reserved	reserved
15: 8	R/W	reserved	reserved
23: 16	R/(BYTE WRITE)	OR1200 复位控制寄存器	80 端口，BMC 复位控制寄存器，只支持单字节写，地址偏移为 0x05。
31: 24	R/(BYTE WRITE)	80h port byte	OS 访问 80 端口对应的 BMC 内部地址。

os_80h_port[23:16]寄存器值的说明：

寄存器值	含义
0xf0	BMC 关机
0x0f	BMC 开机
0xff	ATX_PSON 无效
0xc3	Pcie 软复位
0xaa	上电复位初值，没有特殊功能

OS 访问的 I/O 空间地址 0x308，OpenRISC 看到的地址 0x70000004，对应 os_80h_port[31:24]。

【偏移 0x10 FPGA_VERSION】寄存器位定义（eeb2f 的复位值 0x0010_0103）

BIT	R/W	Name	Function
7: 0	RO	board_type	主板类型即架构，目前有 4 种架构
15: 8	RO	msg_of_board	主板子型号，具体参见主板子型号表格
23: 16	RO	version	高 4 位表示版本号，低 4 位表示子版本号；例子：0x10 表示 v1.0
31: 24	RO	reserved	保留

主板子型号表格：

主版架构（主型号）	子型号	
1: SW-ICH1	1: SW-ICH1 验证板 2: 套片公板 3: 瘦客户机 4: 万方多单元服务器 5: 套片 Micro ATX 6: EEB_Security 51 所高等级密码安全平台 7: BWRouter 比威 8: 万方多单元_ICH	
2: CS5536	1: ATX 2: Micro ATX 3: EEB1.0 4: 浪潮加固笔记本 5: 浪潮桌面 6: 浪潮笔记本 7: 709 主板 8: 华中数控 9: 千兆防火墙 10: 同芯恒通主板 11: 中机 12: 浪潮多单元服务器 13: 浪潮 2U 服务器 14: 万方加固本 15: 万方台式机	
3: 64 位 IO 主板	1: EEB2.0 2: 浪潮多单元服务器 3: 万方大数据机 4: 万方多单元服务器 5: 万方 2U 服务器	
4: 32 位 IO 主板(非 CS5536)	1: 迈普路由器	
其他		

例子：浪潮加固笔记本最初版本的 **FPGA_VERSION: 0X0010_0402**

3.9 维护 (cpu_mt) 模块

BMC 与 CPU 的维护接口，采内部的地址空间范围是 8000-0000~8000-000F。寄存器设置如下：

寄存器偏移 (offset)	BMC Read	BMC Write
0x0	BT_CTRL	
0x4	Host2BMC buffer	BMC2Host buffer
0x8	GPIO_OE	
0xC	GPIO_Q	GPIO_D

【偏移 0x00 BT_CTRL】寄存器位定义

BIT	R/W	Name	Function
0	R	B2H_FIFO empty	命令 FIFO 空
1	R	H2B_FIFO empty	响应 FIFO 空
2	R	H2B_ATN	响应包收齐, BMC 取完后自动清
3	R/S	B2H_ATN	BMC 启动向 CPU 发请求, 发完后自动清
7:4	R	保留	保留
8	R/W	SW_DCOK	CPU 电源好
9	R/W	SW_RESET_N	CPU 复位信号, 低有效
14:10	R	保留	保留
15	R/W	MT_RESET_N	维护接口逻辑复位, 低有效
31:16	R	保留	保留

复位值: 0x0000_0103。

MT_RESET_N 既复位 FPGA 自身的 CPU 维护接口逻辑, 也复位 CPU 的维护接口逻辑。

B2H_ATN: BMC 将维护命令内容全部填入 BMC2Host 缓冲后, 通知接口控制逻辑将缓冲中的内容发给 CPU, 写 1 置。

H2B_ATN: CPU 对维护命令做出响应, 响应内容全部填入 Host2BMC 缓冲后, 通知 BMC 来取维护的响应。

从 BMC 角度看的通常处理流程:

- 1)、在**命令和响应缓冲都空**的情况下, BMC 往 BMC2Host 缓冲按字节顺序填命令包, 发送缓冲变为“非空”;
- 2)、BMC 将所有维护命令字节都写入 BMC2Host 缓冲后, **BMC 置 B2H_ATN**;
- 3)、维护接口逻辑开始将维护命令发送给 CPU, 命令发送完后, B2H_ATN 自动清零;
- 4)、CPU 处理命令返回响应, 之后返回响应包, 填入 Host2BMC 缓冲, 响应缓冲变为“非空”;
- 5)、维护接口逻辑收齐全部响应包信息后, 升起 H2B_ATN;
- 6)、BMC 看到 **H2B_ATN** 后, 取出响应, 响应取完后, H2B_ATN 自动清除;
- 7)、BMC 可以重复 1) ~ 6) 处理新的维护命令。

【偏移 0x04 FIFO】寄存器位定义

建议用 32 位访问, 每次读请求, 从 buffer 中取一个字节放在低位 Q[8:0], 其中 Q[8]和 Q[31]位为校验报错位, 每次写请求往 buffer 中填入一个字节, 数据来自低位 D[7:0]。

如果用字节访问, 需要使用【偏移 0x07】。

【偏移 0x08 MTGPIO_OE】寄存器位定义

BIT	R/W	Name	Function
[23:0]	R/W	MTGPIO_OE	控制 GPIO 管脚输出
[31:24]		保留	保留

复位后的值为“001F,FFF0”, 通常情况下不需要修改此寄存器的值。

【偏移 0x0C MTGPIO_DAT】寄存器位定义

BIT	R/W	Name	Function
[24:0]	R/W	MTGPIO_DAT	对管脚输出数据寄存器写, 或读取管脚值。

[31:25]	RO	Ver	设计版本, 制度
---------	----	-----	----------

复位后的值为“0000,0000”。MTGPIO_DAT 的具体含义, 见管脚定义, SW-2F 和 SW-3 的含义各不相同。

CPU 接口: 考虑到适应 SW-2F 和 SW-3 三款 CPU, 管脚定义进行调整。

管脚名	说明	SW-2F 用法	SW-3 用法
SWMT_CLK	给 CPU 的维护时钟	同左√	同左√
SWMT_CLK#	给 CPU 的维护时钟, 相位取反	同左√	同左√
SWMT_TX	CPU 的串口发送	同左√	同左√
SWMT_RX	CPU 的串口接收	同左√	同左√
SWMT_RESE T#	CPU 的维护接口复位	同左√	同左√
SW_RESET#	CPU 的热复位	同左√	同左√
SW_DCOK#	CPU 的上电复位	同左√	同左√
SW_GPIO[0]	CPU 用 GPIO, 输入	TEST_OUT[0]_H	MT_FSM[0]
SW_GPIO[1]	CPU 用 GPIO, 输入	TEST_OUT[1]_H	MT_FSM[1]
SW_GPIO[2]	CPU 用 GPIO, 输入	TEST_OUT[2]_H	MT_FSM[2]
SW_GPIO[3]	CPU 用 GPIO, 输入	TEST_OUT[3]_H	MT_FSM[3]
SW_GPIO[4]	CPU 用 GPIO, 输出	同左√	CFG_GCLKPLL[0]
SW_GPIO[5]	CPU 用 GPIO, 输出	同左√	CFG_GCLKPLL[1]
SW_GPIO[6]	CPU 用 GPIO, 输出	同左√	CFG_GCLKPLL[2]
SW_GPIO[7]	CPU 用 GPIO, 输出	同左√	CFG_GCLKPLL[3]
SW_GPIO[8]	CPU 用 GPIO, 输出	同左√	CFG_GCLKPLL[4]
SW_GPIO[9]	CPU 用 GPIO, 输出	INIT_MODE_H[0]	INIT_MODE[0]
SW_GPIO[10]	CPU 用 GPIO, 输出	INIT_MODE_H[1]	INIT_MODE[1]
SW_GPIO[11]	CPU 用 GPIO, 输出	[4.7K 电阻接地]	GCLKPLL_BYPASS
SW_GPIO[12]	CPU 用 GPIO, 输出	同左√	CFG_MCCLKPLL[0]
SW_GPIO[13]	CPU 用 GPIO, 输出	同左√	CFG_MCCLKPLL[1]
SW_GPIO[14]	CPU 用 GPIO, 输出	同左√	CFG_MCCLKPLL[2]
SW_GPIO[15]	CPU 用 GPIO, 输出	同左√	CFG_MCCLKPLL[3]
SW_GPIO[16]	CPU 用 GPIO, 输出	CFG_XBX_H[0]	CFG_SPECLKPLL[0]
SW_GPIO[17]	CPU 用 GPIO, 输出	CFG_XBX_H[1]	CFG_SPECLKPLL[1]
SW_GPIO[18]	CPU 用 GPIO, 输出	CFG_XBX_H[2]	CFG_SPECLKPLL[2]
SW_GPIO[19]	CPU 用 GPIO, 输出	CFG_XBX_H[3]	CFG_SPECLKPLL[3]
SW_GPIO[20]	CPU 用 GPIO, 输出	/	CFG_SPECLKPLL[4]
SW_GPIO[21]	CPU 用 GPIO, 输入	[4.7K 电阻接地]	CLOBS
SW_GPIO[22]	CPU 专用 GPIO, 输入	同左√	同左√
SW_GPIO[23]	CPU 专用 GPIO, 输入	TEST_OUT[4]_H	MT_INITEND

SW_PRSENT#	CPU 的在位标识, BMC 输入	由主板设计定	由主板设计定
------------	-------------------	--------	--------

3.10 uart 模块

内部寄存器编址:

LegacyIO 地址	BMC 内部地址	位宽	属性	功能说明
0x3F8	0x9000-0000	8	R	接收缓存
0x3F8	0x9000-0000	8	W	发送缓存
0x3F9	0x9000-0001	8	RW	中断使能
0x3FA	0x9000-0002	8	R	中断识别
0x3FA	0x9000-0002	8	W	FIFO 控制
0x3FB	0x9000-0003	8	RW	线控制
0x3FC	0x9000-0004	8	W	模式控制
0x3FD	0x9000-0005	8	R	线状态
0x3FE	0x9000-0006	8	R	模式状态

3.11 uart1 模块

内部寄存器编址:

LegacyIO 地址	BMC 内部地址	位宽	属性	功能说明
0x3F8	0xa000-0000	8	R	接收缓存
0x3F8	0xa000-0000	8	W	发送缓存
0x3F9	0xa000-0001	8	RW	中断使能
0x3FA	0xa000-0002	8	R	中断识别
0x3FA	0xa000-0002	8	W	FIFO 控制
0x3FB	0xa000-0003	8	RW	线控制
0x3FC	0xa000-0004	8	W	模式控制
0x3FD	0xa000-0005	8	R	线状态
0x3FE	0xa000-0006	8	R	模式状态

3.12 wdt 模块

模块在 BMC 内部的基址为 0xb0000000, 模块内部包括 2 个 32 位的寄存器, 寄存器设置如下:

寄存器	BMC 内部地址	位宽	属性	功能说明
TC	0xb000-0000	32	R	计时器
TC_CTRL	0xb000-0004	32	W	计时器控制寄存器

【偏移 0x00 TC】寄存器位定义

名称	范围	类型	描述	功能说明
----	----	----	----	------

—	[31:24]	—	保留。	计时器
TC	[23:0]	RW,0x7ffff	计时器的计数 值。	计时器控制寄存器

【偏移 0x04 TC_CTRL】 寄存器位定义

名称	范围	类型	描述
—	[31:2]	—	保留。
TC_Overflow	[1]	RW,0	计数器发生溢出，硬件置， 软件清。
TC_ENA	[0]	RW,0	计时器计数使能，为“1”时 计时器开始计数。