

S5933 接口简介

目录:

一、	S5933 的结构简介	P2
二、	S5933 的主要引脚信号说明	P3 ~P7
三、	S5933 后端逻辑的电路设计	P8 ~P12
四、	S5933 的 PCI 配置	P13~P21
五、	PCI 总线操作寄存器组	P22~P27
六、	ADD-ON 总线操作寄存器组	P28~P33
七、	ADD-ON 总线的三种工作方式	P34~P43
八、	重要信号的时间参数	P44~P49
九、	FPGA 状态机设计举例	P50~P52

一、 S5933 的结构简介

S5933 是 AMCC 公司生产的一款 PCI 接口芯片，符合 PCI 规范 2.1 版本。S5933 提供三种物理接口：PCI 总线接口、外接总线接口 (Add-On interface) 和外接配置存储器 (非易失存储器 nvRAM) 接口 (见图 1)，其中，PCI 总线接口与 PCI 总线相连，配置存储器接口与非易失存储器连接 (在系统初始化时用来对 S5933 进行配置)，外接总线接口 (Add-On interface) 与用户设备的数据接口连接。本文主要介绍外接总线接口 (Add-On interface)。

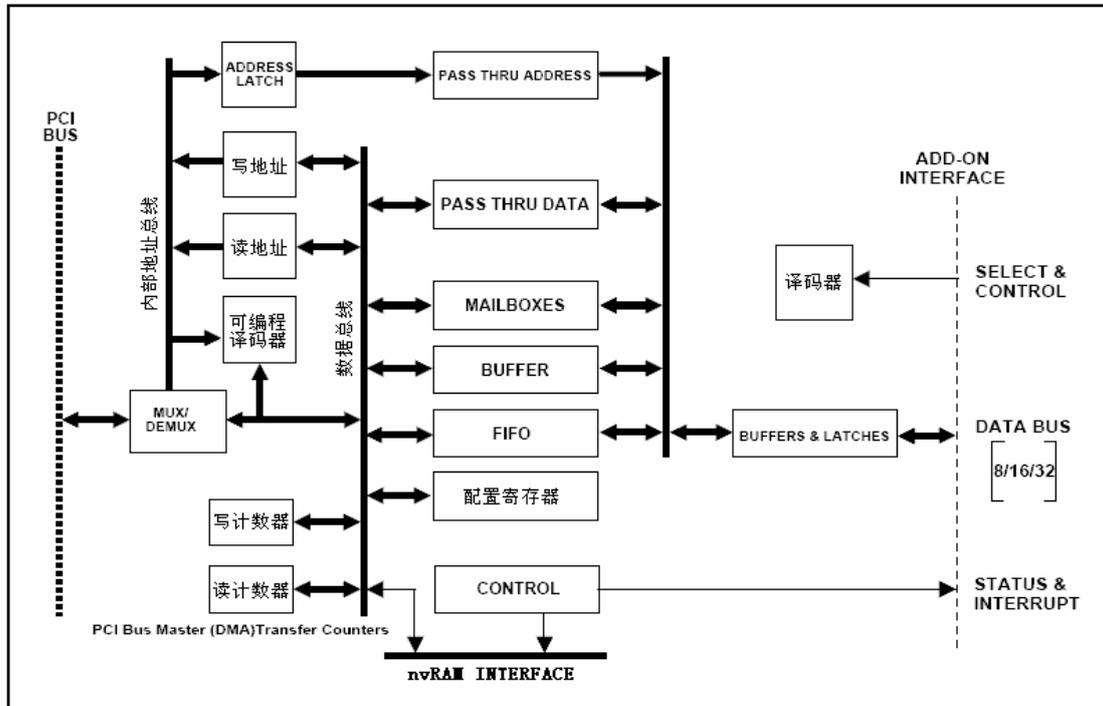


图1 S5933结构框图

S5933 内的寄存器分为两大组，PCI 配置寄存器组和操作寄存器组。对于 PCI 配置寄存器组，在上电时，可以由 S5933 从 nvRAM 的地址偏移 40h~7fh (byte) 载入，而在没有外接 nvRAM 或者外接 nvRAM 的地址 00h 中数据为 FFh 时，S5933 给配置寄存器写入默认值。操作寄存器组用于 S5933 同 ADD-ON 接口的操作配置和数据传送，如中断控制、状态查询、数据读写等。在操作寄存器中，从 PCI 总线访问的操作寄存器组，称为 PCI 操作寄存器组 (PCI BUS OPERATION REGISTERS)，地址为 PCI 操作寄存器基址 (BASE ADDRESS0) 加各寄存器偏移地址；而从 ADD-ON 接口访问的操作寄存器组，称为 ADD-ON 总线操作寄存器组 (ADD-ON BUS OPERATION REGISTERS)，访问的寄存器由 ADD-ON 接口上的信号线 ADDR[6:2] 选择。

二、 S5933 的主要引脚信号说明

S5933 的引脚信号如下图 2:

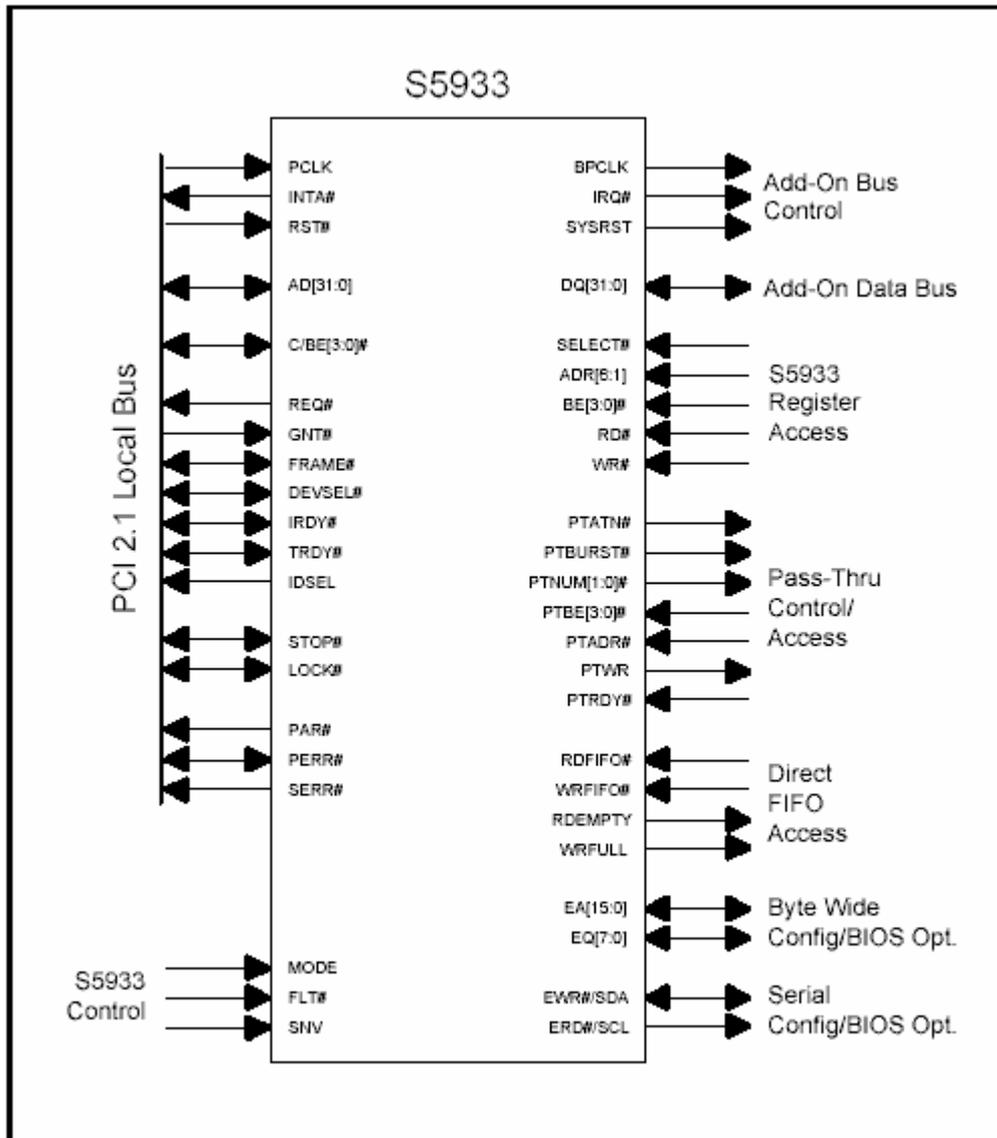


图2 S5933 Signal pins

上图 2 中左上部分信号 (PCI2.1 Local Bus), 用于同 5V PCI 插槽的连接, 直接连接到 PCI 插槽相应引脚即可。S5933 本身就是一个 PCI 协议译码器, 若要完全清楚 PCI 协议以及 PCI 总线上的信号, 请参考有关 PCI 协议及系统结构的资料。

1) PCI 总线接口信号 (PCI Local BUS)

信号	信号类型	说明
CLK	I	33Mhz PCI 时钟。PCI 总线上的所有信号以 CLK 的上升沿为参考。
RST#	I	复位信号。
AD[31:0]	I/O/S	PCI 地址/数据总线。
C/BE[3:0]#	I/O/S	PCI 总线命令和字节使能。在 PCI 总线操作的地址周期, 这些信号用来指定总线命令; 在数据周期, 用作字节使能信号。 C/BE[3:0]# Command(during address phase)

		0 0 0 0 中断确认 0 0 0 1 专用周期 0 0 1 0 IO 读 0 0 1 1 IO 写 0 1 0 0 保留 0 1 0 1 保留 0 1 1 0 存储器读 0 1 1 1 存储器写 1 0 0 0 保留 1 0 0 1 保留 1 0 1 0 配置读 1 0 1 1 配置写 1 1 0 0 存储器读——多倍 1 1 0 1 双地址周期 1 1 1 0 存储器读线 1 1 1 1 存储器写和无效
PAR	I/O/Z	偶校验。在地址段完成后一个时钟周期，或者在写交易的每个数据段中 IRDY#有效之后一个时钟周期，主设备驱动奇偶校验信号 PAR，在读交易的每个数据段中 TRDY#有效之后一个时钟周期，当前寻址目标也驱动奇偶校验信号 PAR。
FRAME#	I/O/Z	周期帧信号。由当前总线主设备驱动，指明交易的起始（当它第一次有效）和交易的持续时间（它有效的持续时间）。
IRDY#	I/O/Z	启动方准备好信号。由当前总线主设备驱动，在写交易中，有效的 IRDY#表明主设备正准备接收来自当前寻址目标的数据。
TRDY#	I/O/Z	目标准备好信号。由当前寻址目标驱动，当目标准备好完成当前数据段（数据传送）时，此信号有效。
STOP#	I/O/Z	目标有效 STOP#信号表示它希望主设备中止当前交易。
LOCK#	I	由当前总线主设备驱动，LOCK#有效时，表明主设备唯一占用目标设备。S5933 在作为目标设备时，可以被主设备锁定；S5933 在作为主设备时，不能锁定目标设备。
IDSEL	I	初始化设备选择信号。PCI 总线在访问一个设备的配置寄存器时作为芯片选择。
DEVSEL#	I/O/Z	设备选择信号。当一个目标设备译码出地址是选择自己时，有效此信号。
REQ#	0	总线请求。当 PCI 总线上的一个设备想成为总线主设备时，有效此信号。每一个 PCI 总线的设备有一个独立的 REQ#连接到 PCI 总线仲裁器。
GNT#	I	总线允许。由 PCI 总线仲裁器发给请求总线的设备。
PERR#	I/O/Z	奇偶校验错误。由接收数据方驱动。
SERR#	OC 输出	系统错误。任何设备都可驱动 SERR#以报告一个错误。
INTA#	OC 输出	中断 A。S5933 需要中断主机时，有效此信号。

2) 串行配置芯片 nvRAM 或 EEPROM 接口

如果仅需要配置 S5933，而不扩展 BIOS 时，建议给 S5933 外接串行 nvRAM 或 EEPROM 芯片，因为串行 nvRAM 或 EEPROM 芯片体积小、引脚少、价格便宜，而且此时并行接口可扩展其它用途。串行 nvRAM 或 EEPROM 芯片可以选用 AT24C01、AT24C02、AT24C04、AT24C08、AT24C16 等。

信号	信号类型	说明
SCL	I/O/Z	Serial Clock. 直接接到串行 nvRAM 或 EEPROM 的 SCL 引脚。
SDA	I/O/Z	Serial Data/Address. 直接接到 nvRAM 或 EEPROM 的 SDA 引脚。
SNV	I	高电平：外接串行 nvRAM 或 EEPROM。 低电平：外接并行 (byte-wide) EEPROM

3) 并行配置/BIOS 芯片 nvRAM 或 EEPROM 接口

信号	信号类型	说明																				
EA[15:0]	I/O/Z	扩展并行 nvRAM 或 EEPROM 的地址线引脚。从地址 0040h 到 007Fh 用于存放 S5933 的配置信息。扩展并行 nvRAM 或 EEPROM 的容量最小为 128bytes，最大为 64K bytes。当 S5933 接串行配置芯片时，EA[7:0]变为 PCI incoming mailbox 4, byte 3 (即 Add-On outgoing mailbox 4, byte 3) 数据输入线，EA8 成为数据触发线。如果 S5933 的中断控制设为写 PCI incoming mailbox 4, byte 3 产生 PCI 中断时，Add-On 后端逻辑可以用此方法给 PCI 总线产生中断。当 S5933 接串行配置芯片时，同时，EA15 成为 PCI to Add-On FIFO 满标志(FRF), EA14 成为 Add-On to PCI FIFO 空标志 (FWE)。 <table style="width: 100%; border: none;"> <tr> <td style="width: 50%;">Signal Pin</td> <td>Add-On outgoing Mailbox</td> </tr> <tr> <td>EA0/EMB0</td> <td>Mailbox 4, bit24</td> </tr> <tr> <td>EA1/EMB1</td> <td>Mailbox 4, bit25</td> </tr> <tr> <td>EA2/EMB2</td> <td>Mailbox 4, bit26</td> </tr> <tr> <td>EA3/EMB3</td> <td>Mailbox 4, bit27</td> </tr> <tr> <td>EA4/EMB4</td> <td>Mailbox 4, bit28</td> </tr> <tr> <td>EA5/EMB5</td> <td>Mailbox 4, bit29</td> </tr> <tr> <td>EA6/EMB6</td> <td>Mailbox 4, bit30</td> </tr> <tr> <td>EA7/EMB7</td> <td>Mailbox 4, bit31</td> </tr> <tr> <td>EA8/EMBCLK</td> <td>Mailbox 4, byte 3 load clock</td> </tr> </table>	Signal Pin	Add-On outgoing Mailbox	EA0/EMB0	Mailbox 4, bit24	EA1/EMB1	Mailbox 4, bit25	EA2/EMB2	Mailbox 4, bit26	EA3/EMB3	Mailbox 4, bit27	EA4/EMB4	Mailbox 4, bit28	EA5/EMB5	Mailbox 4, bit29	EA6/EMB6	Mailbox 4, bit30	EA7/EMB7	Mailbox 4, bit31	EA8/EMBCLK	Mailbox 4, byte 3 load clock
Signal Pin	Add-On outgoing Mailbox																					
EA0/EMB0	Mailbox 4, bit24																					
EA1/EMB1	Mailbox 4, bit25																					
EA2/EMB2	Mailbox 4, bit26																					
EA3/EMB3	Mailbox 4, bit27																					
EA4/EMB4	Mailbox 4, bit28																					
EA5/EMB5	Mailbox 4, bit29																					
EA6/EMB6	Mailbox 4, bit30																					
EA7/EMB7	Mailbox 4, bit31																					
EA8/EMBCLK	Mailbox 4, byte 3 load clock																					
ERD#	0	扩展并行 nvRAM 或 EEPROM 读信号。ERD#由低变高时，数据读入 S5933。此信号与串行配置 SCL 信号公用一个 S5933 IO 引脚。																				
EWR#	I/O/Z	扩展并行 nvRAM 或 EEPROM 写信号。EWR#由低变高时，数据由 S5933 写入并行 nvRAM 或 EEPROM。此信号与串行配置 SDA 信号公用一个 S5933 IO 引脚。																				
EQ[7:0]	I/O/Z	扩展并行 nvRAM 或 EEPROM 的数据总线。当 S5933 接串行配置芯片时，若 S5933 配置为 PCI 主设备，EQ4、EQ5、																				

		EQ6、EQ7 成为控制信号。
--	--	-----------------

4) ADD-ON 总线寄存器访问信号

信号	信号类型	说明																																						
DQ[31:0]	I/O/Z	ADD-ON 数据总线。																																						
ADR[6:2]	I	寄存器访问地址总线。 <div style="text-align: center;"> <table border="0" style="margin: auto;"> <tr> <td style="padding-right: 20px;">ADR[6:2]</td> <td>寄存器</td> </tr> <tr> <td>0 0 0 0 0</td> <td>ADD-ON Incoming Mailbox Reg. 1 (对应 PCI Outgoing Mailbox Reg. 1)</td> </tr> <tr> <td>0 0 0 0 1</td> <td>ADD-ON Incoming Mailbox Reg. 2 (对应 PCI Outgoing Mailbox Reg. 2)</td> </tr> <tr> <td>0 0 0 1 0</td> <td>ADD-ON Incoming Mailbox Reg. 3 (对应 PCI Outgoing Mailbox Reg. 3)</td> </tr> <tr> <td>0 0 0 1 1</td> <td>ADD-ON Incoming Mailbox Reg. 4 (对应 PCI Outgoing Mailbox Reg. 4)</td> </tr> <tr> <td>0 0 1 0 0</td> <td>ADD-ON Outgoing Mailbox Reg. 1 (对应 PCI Incoming Mailbox Reg. 1)</td> </tr> <tr> <td>0 0 1 0 1</td> <td>ADD-ON Outgoing Mailbox Reg. 2 (对应 PCI Incoming Mailbox Reg. 2)</td> </tr> <tr> <td>0 0 1 1 0</td> <td>ADD-ON Outgoing Mailbox Reg. 3 (对应 PCI Incoming Mailbox Reg. 3)</td> </tr> <tr> <td>0 0 1 1 1</td> <td>ADD-ON Outgoing Mailbox Reg. 4 (对应 PCI Incoming Mailbox Reg. 4)</td> </tr> <tr> <td>0 1 0 0 0</td> <td>ADD-ON FIFO 接口</td> </tr> <tr> <td>0 1 0 0 1</td> <td>PCI 总线主设备写地址</td> </tr> <tr> <td>0 1 0 1 0</td> <td>ADD-ON Pass-Thru 地址</td> </tr> <tr> <td>0 1 0 1 1</td> <td>ADD-ON Pass-Thru 数据</td> </tr> <tr> <td>0 1 1 0 0</td> <td>PCI 总线主设备读地址</td> </tr> <tr> <td>0 1 1 0 1</td> <td>ADD-ON Mailbox 空/满 状态</td> </tr> <tr> <td>0 1 1 1 0</td> <td>ADD-ON 中断控制寄存器</td> </tr> <tr> <td>0 1 1 1 1</td> <td>ADD-ON 控制/状态寄存器</td> </tr> <tr> <td>1 0 1 1 0</td> <td>PCI 总线主设备写计数器</td> </tr> <tr> <td>1 0 1 1 1</td> <td>PCI 总线主设备读计数器</td> </tr> </table></div>	ADR[6:2]	寄存器	0 0 0 0 0	ADD-ON Incoming Mailbox Reg. 1 (对应 PCI Outgoing Mailbox Reg. 1)	0 0 0 0 1	ADD-ON Incoming Mailbox Reg. 2 (对应 PCI Outgoing Mailbox Reg. 2)	0 0 0 1 0	ADD-ON Incoming Mailbox Reg. 3 (对应 PCI Outgoing Mailbox Reg. 3)	0 0 0 1 1	ADD-ON Incoming Mailbox Reg. 4 (对应 PCI Outgoing Mailbox Reg. 4)	0 0 1 0 0	ADD-ON Outgoing Mailbox Reg. 1 (对应 PCI Incoming Mailbox Reg. 1)	0 0 1 0 1	ADD-ON Outgoing Mailbox Reg. 2 (对应 PCI Incoming Mailbox Reg. 2)	0 0 1 1 0	ADD-ON Outgoing Mailbox Reg. 3 (对应 PCI Incoming Mailbox Reg. 3)	0 0 1 1 1	ADD-ON Outgoing Mailbox Reg. 4 (对应 PCI Incoming Mailbox Reg. 4)	0 1 0 0 0	ADD-ON FIFO 接口	0 1 0 0 1	PCI 总线主设备写地址	0 1 0 1 0	ADD-ON Pass-Thru 地址	0 1 0 1 1	ADD-ON Pass-Thru 数据	0 1 1 0 0	PCI 总线主设备读地址	0 1 1 0 1	ADD-ON Mailbox 空/满 状态	0 1 1 1 0	ADD-ON 中断控制寄存器	0 1 1 1 1	ADD-ON 控制/状态寄存器	1 0 1 1 0	PCI 总线主设备写计数器	1 0 1 1 1	PCI 总线主设备读计数器
ADR[6:2]	寄存器																																							
0 0 0 0 0	ADD-ON Incoming Mailbox Reg. 1 (对应 PCI Outgoing Mailbox Reg. 1)																																							
0 0 0 0 1	ADD-ON Incoming Mailbox Reg. 2 (对应 PCI Outgoing Mailbox Reg. 2)																																							
0 0 0 1 0	ADD-ON Incoming Mailbox Reg. 3 (对应 PCI Outgoing Mailbox Reg. 3)																																							
0 0 0 1 1	ADD-ON Incoming Mailbox Reg. 4 (对应 PCI Outgoing Mailbox Reg. 4)																																							
0 0 1 0 0	ADD-ON Outgoing Mailbox Reg. 1 (对应 PCI Incoming Mailbox Reg. 1)																																							
0 0 1 0 1	ADD-ON Outgoing Mailbox Reg. 2 (对应 PCI Incoming Mailbox Reg. 2)																																							
0 0 1 1 0	ADD-ON Outgoing Mailbox Reg. 3 (对应 PCI Incoming Mailbox Reg. 3)																																							
0 0 1 1 1	ADD-ON Outgoing Mailbox Reg. 4 (对应 PCI Incoming Mailbox Reg. 4)																																							
0 1 0 0 0	ADD-ON FIFO 接口																																							
0 1 0 0 1	PCI 总线主设备写地址																																							
0 1 0 1 0	ADD-ON Pass-Thru 地址																																							
0 1 0 1 1	ADD-ON Pass-Thru 数据																																							
0 1 1 0 0	PCI 总线主设备读地址																																							
0 1 1 0 1	ADD-ON Mailbox 空/满 状态																																							
0 1 1 1 0	ADD-ON 中断控制寄存器																																							
0 1 1 1 1	ADD-ON 控制/状态寄存器																																							
1 0 1 1 0	PCI 总线主设备写计数器																																							
1 0 1 1 1	PCI 总线主设备读计数器																																							
BE3#(ADR1)	I	Byte Enable 3(32-bit mode) or ADR1(16 bit mode)																																						
BE[2:0]#	I	Byte Enable 2 through 0.																																						
SELECT#	I	低电平片选 ADD-ON interface.																																						
WR#	I	Write strobe.																																						
RD#	I	Read strobe.																																						
MODE	I	数据总线 DQ 宽度选择, 32bits wide(MODE=低电平), 16bits wide(MODE=高电平)。																																						

5) 直接 FIFO 访问引脚 (DIRECT FIFO Access Pins)

DIRECT FIFO Access 信号使 S5933 内 FIFO 可以简单快速地得到访问, 使用 SELECT#、RD#、WR#、ADDR[6:2] (ADDR[6:2] = [01000b]时)同样可以访问到 S5933

内 FIFO, 但使用 DIRECT FIFO Access 信号访问 FIFO 更方便, 好处就是不需要 SELECT#, RD# 等信号的配合。

信号	信号类型	说明
WRFIFO#	I	Write FIFO。
RDFIFO#	I	Read FIFO。
WRFULL	0	Write FIFO Full。
RDEMPY	0	Read FIFO Empty。

6) 直通接口引脚 (Pass-Thru Interface Pins)

信号	信号类型	说明
PTATN#	0	Pass-Thru Attention。PTATN# 为低电平表示计算机访问 Pass-Thru Interface 已经被 S5933 成功译码, 而且, 接下来 Pass-Thru 数据寄存器必须被读或写。
PTBURST#	0	Pass-Thru Burst。
PTRDY#	I	Pass-Thru Ready。
PTNUM[1:0]	0	Pass-Thru Number。指示当前的 Pass-Thru 操作对应的 Base Address Register。如 PTNUM[1:0] = 00 代表 Base Address Register 1。Base Address Register 1~4 的具体值需要预先写到 S5933 的 EEPROM 配置芯片中。
PTBE[3:0]#	0	Pass-Thru Byte Enable。
PTADR#	I	Pass-Thru Address。在 PTATN# 有效时, ADD-ON 后端逻辑若置低 PTADR#, S5933 就会把相对于基地址 (Base Address Register) 的地址偏移量输出到数据总线 DQ 上。
PTWR	0	Pass-Thru Write。在 PTATN# 有效时, 若 PTWR 为高电平, 则表示计算机写 Pass-Thru 数据寄存器, 反之, 则表示计算机读 Pass-Thru 数据寄存器。

7) 系统引脚 (System Pins)

信号	信号类型	说明
SYSRST#	0	System Reset。计算机主机 reset 的缓冲输出, 此信号与 PCI 总线时钟不同步。
BPCLK	0	Buffered PCI Clock。PCI 总线时钟的缓冲输出。
IRQ#	0	中断信号。S5933 中断 Add-On 接口上的逻辑。
FLT#	I	Float。此信号置低, S5933 的所有输出将浮空。

三、 S5933 后端逻辑的电路设计

1. 原理图设计的例子

情况 1: S5933 配置 PCI 总线从设备, 并且 S5933 只接一个 FPGA, 使用通信方式 FIFO、Mailbox 和 Pass-Thru。如果 FPGA 的 IO 脚不够用, 可以把 Pass-Thru 去掉。S5933 与 FPGA 的接口如下图 3。

图 3 中 MODE 信号接低电平表示 S5933 Add-On 接口数据宽度为 32bits, SNV 接高电平表示 S5933 外接串行配置芯片, SELECT# 信号接低电平表示 S5933 永远被 FPGA 选中, PTBE[3:0]# 没有连接到 FPGA 表示默认 Pass-Thru 数据宽度 32bits。

由于 S5933 只有一条数据总线 DQ[31:0] 被 FIFO、Mailbox 和 Pass-Thru 共用, FPGA 必须保证某一时刻只有一个通信方式处于激活状态, 对于 FIFO、Mailbox 接口, FPGA 是主控方, S5933 是从方, 而 Pass-Thru 接口 S5933 是主控方, FPGA 为从方。对于 FIFO、Mailbox 接口, FPGA 很容易使它们不冲突。在 FPGA 访问 FIFO 期间, 若 S5933 通过 Pass-Thru 访问 FPGA 时, FPGA 应暂停 FIFO 的访问, 高阻数据总线 DQ[31:0], 响应 Pass-Thru, 等 Pass-Thru 通信完成以后, 再接着访问 FIFO 接口。在 FPGA 访问 Mailbox 期间, 若 S5933 通过 Pass-Thru 访问 FPGA 时, 由于 Mailbox 通信周期比较短(约 2 个 PCI 时钟周期), FPGA 可以暂不响应 Pass-Thru, 等 Mailbox 通信完成以后, FPGA 再响应 Pass-Thru。

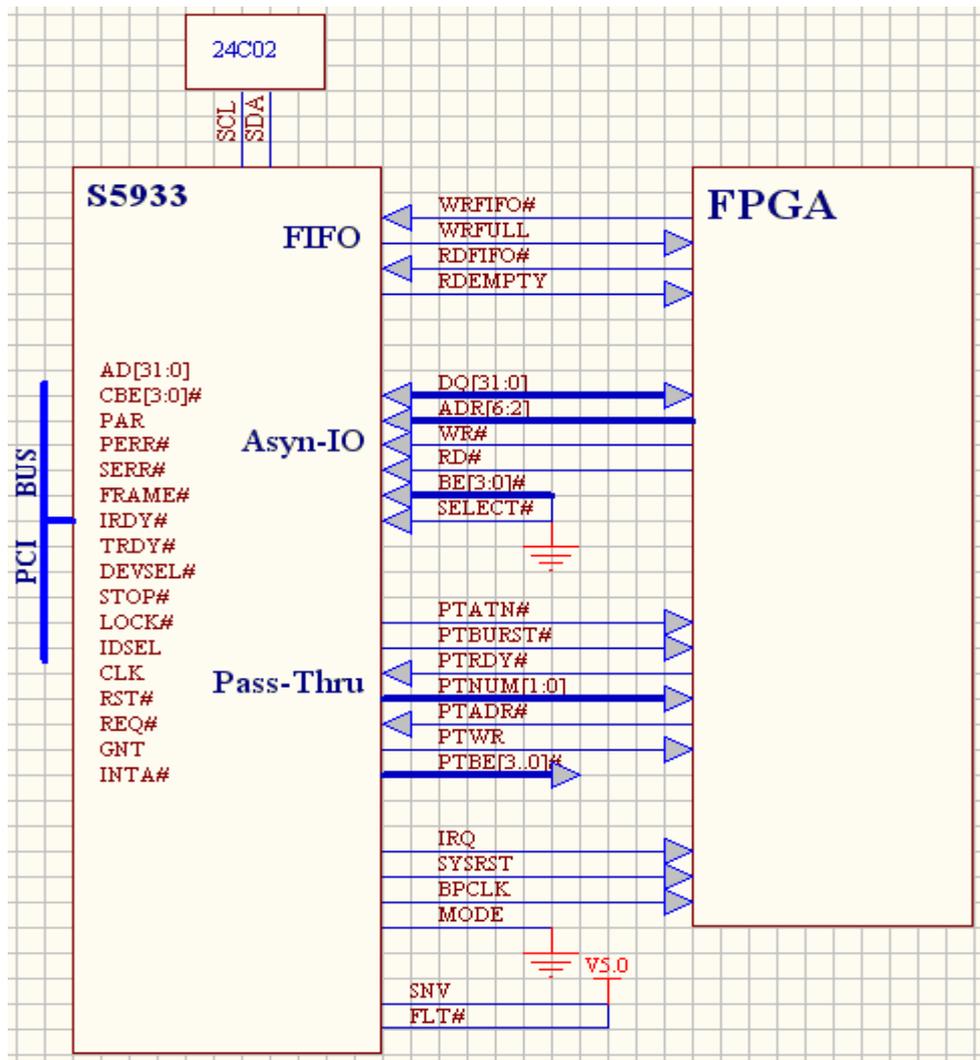


图 3 S5933 例图 1

情况 2: S5933 配置 PCI 总线从设备, 并且 S5933 接两片 FPGA, 一片 FPGA 使用通信方式 FIFO、Mailbox 和 Pass-Thru, 另一片 FPGA 使用通信方式 Pass-Thru。S5933 与 FPGA 的接口如下图 4。

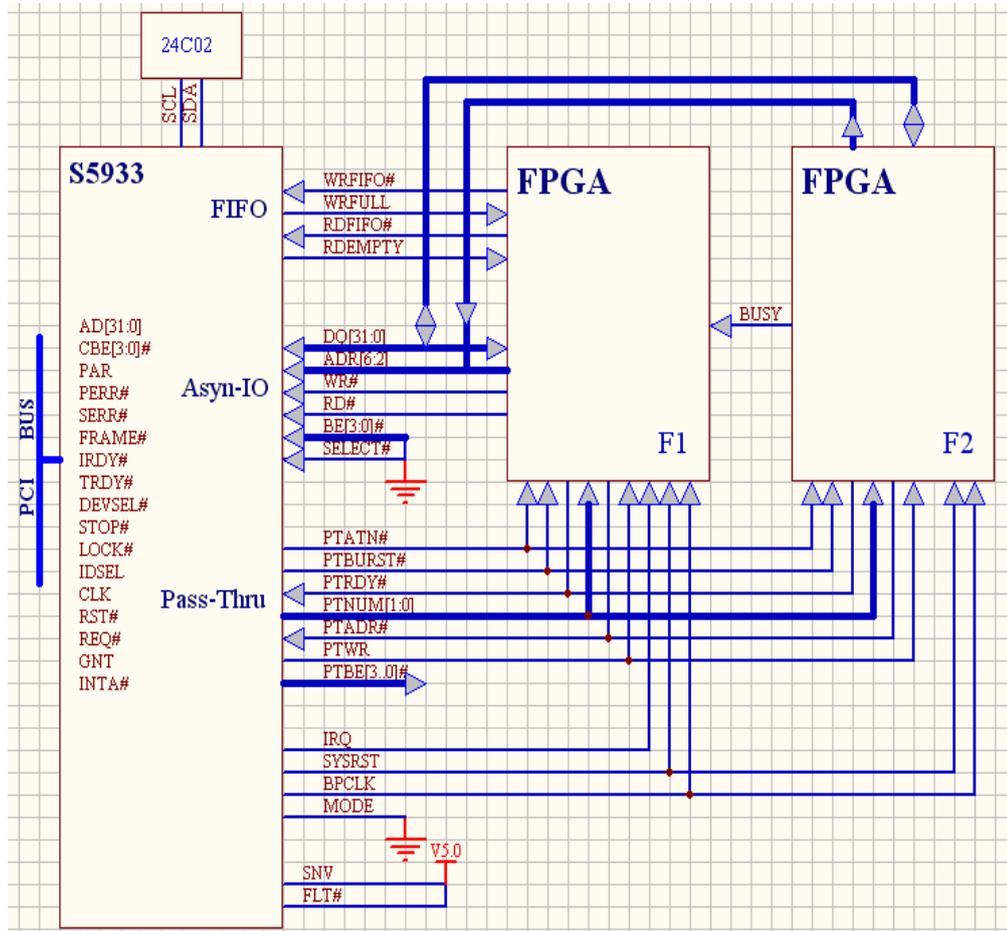


图 4 S5933 例图 2

图 4 中 MODE 信号接低电平表示 S5933 Add-On 接口数据宽度为 32bits, SNV 接高电平表示 S5933 外接串行配置芯片, SELECT# 信号接低电平表示 S5933 永远被 FPGA 选中, PTBE[3:0]# 没有连接到 FPGA 表示默认 Pass-Thru 数据宽度 32bits。

假设 F1 的 Pass-Thru 映射在 Base Address 1 上, F2 的 Pass-Thru 映射在 Base Address 2 上, 也就是说当 S5933 输出 PTATN# 为低电平, 并且 PTNUM[1:0] 为 “00” 时, S5933 要通过 Pass-Thru 与 F1 通信; 当 S5933 输出 PTATN# 为低电平, 并且 PTNUM[1:0] 为 “01” 时, S5933 要通过 Pass-Thru 与 F2 通信。在 F1 与 F2 之间增加一个信号线 BUSY, 由 F2 输出到 F1, 当 S5933 通过 Pass-Thru 与 F2 通信时, F2 输出 BUSY 为有效高电平到 F1, 否则输出低电平。其实也可以不要 BUSY 信号, 在 F1 内做逻辑检测 Pass-Thru 是否与 F2 通信也可达到同样目的。

从图 4 中可以看出, F1 和 F2 共同驱动的输出有 DQ[31:0]、ADR[6:2]、PTRDY#、PTADR#。给 PTRDY#、PTADR# 接上拉电阻, 以防在控制信号上出现不确定状态。F1 的逻辑和情况 1 差别不大, F2 只有 Pass-Thru, 注意用 PTNUM[1:0] 选择 F1 或者 F2 的 Pass-Thru 接口。

对 F1 来说, 这几个信号的驱动逻辑如下:

```
DQ[31:0] = write FIFO data      WHEN( WRFIFO# == true ) ELSE
```

```

write Mailbox data  WHEN( WR#    == true ) ELSE
write Pass-Thru data WHEN( PTATN# == true AND PTNUM[1:0] ==
                        "00" AND PTADR# == false AND PTWR
                        == '0' ) ELSE
X" ZZZZ" ;
ADR[6:2] = Mailbox address WHEN( WR# == true OR RD# == true ) ELSE
X "2C"      WHEN(PTATN# == true AND PTNUM[1:0] == "00" )ELSE
"Z" ;
PTRDY# = Pass-Thru 状态机产生的值 WHEN(PTATN# == true AND PTNUM[1:0] ==
                        "00" )ELSE
"Z" ;
PTADR# = Pass-Thru 状态机产生的值 WHEN(PTATN# == true AND PTNUM[1:0] ==
                        "00" )ELSE
"Z" ;

```

对 F2 来说，这几个信号的驱动逻辑如下：

```

DQ[31:0] = write Pass-Thru data WHEN( PTATN# == true AND PTNUM[1:0] ==
                        "01" AND PTADR# == false AND PTWR
                        == '0' ) ELSE
X" ZZZZ" ;
ADR[6:2] = X "2C"      WHEN(PTATN# == true AND PTNUM[1:0] == "01" )ELSE
"Z" ;
PTRDY# = Pass-Thru 状态机产生的值 WHEN(PTATN# == true AND PTNUM[1:0] ==
                        "01" )ELSE
"Z" ;
PTADR# = Pass-Thru 状态机产生的值 WHEN(PTATN# == true AND PTNUM[1:0] ==
                        "01" )ELSE
"Z" ;

```

情况 3: S5933 配置 PCI 总线主设备，并且 S5933 接一片 FPGA，FPGA 使用通信方式 FIFO、Mailbox 和 Pass-Thru，如果 FPGA 的 IO 脚不够用，可以把 Pass-Thru 去掉。S5933 与 FPGA 的接口如下图 5。

图 5 中 MODE 信号接低电平表示 S5933 Add-On 接口数据宽度为 32bits，SNV 接高电平表示 S5933 外接串行配置芯片，SELECT# 信号接低电平表示 S5933 永远被 FPGA 选中。

在 S5933 配置为 PCI 主设备时，读/写地址和读/写计数寄存器只能在 Add-On 端访问，所以这些寄存器的值只能通过 Mailbox 或 Pass-Thru 传给 FPGA，然后再通过写 S5933 寄存器的方法从 Add-On 接口写入 S5933。

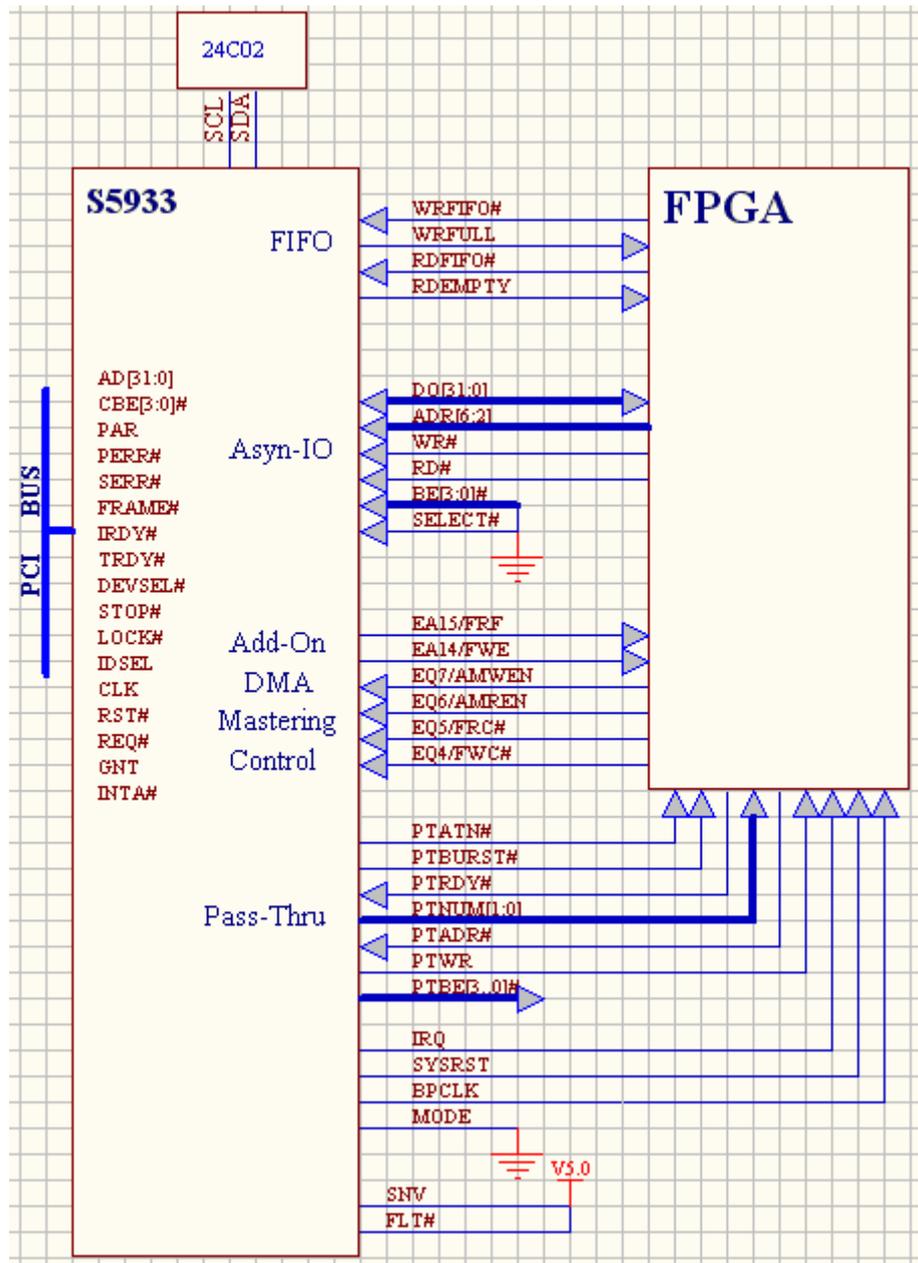


图 5 S5933 例图 3

2. PCB 布线的建议

在 PCB 布线时，要注意在 PCI 规范中，除了 PCI CLK 信号以外的其它信号用信号的反射波驱动，所以驱动的信号只用了要求电压的一半，另一半靠反射来提升，所以对信号线的长度有要求：对 S5933 到 PCI 总线的每个 32 位接口数据信号的最大连线长度是 1500mil，64 位扩展信号的附加信号的连线长度最大为 2000mil，PCI CLK 的长度为 2500mil \pm 100mil，如果不够长度可以多绕几圈。以上关于布线长度的要求我是在别人的文章中看到的，我没有在正规的资料或协议中看到有关布线长度的严格要求。从我的经验来看，PCI 接口的布线要求不高，把 PCI CLK 布得稍微长一点，一般情况下不会有问題。下图 6 为 PCI 布线的一个例子。

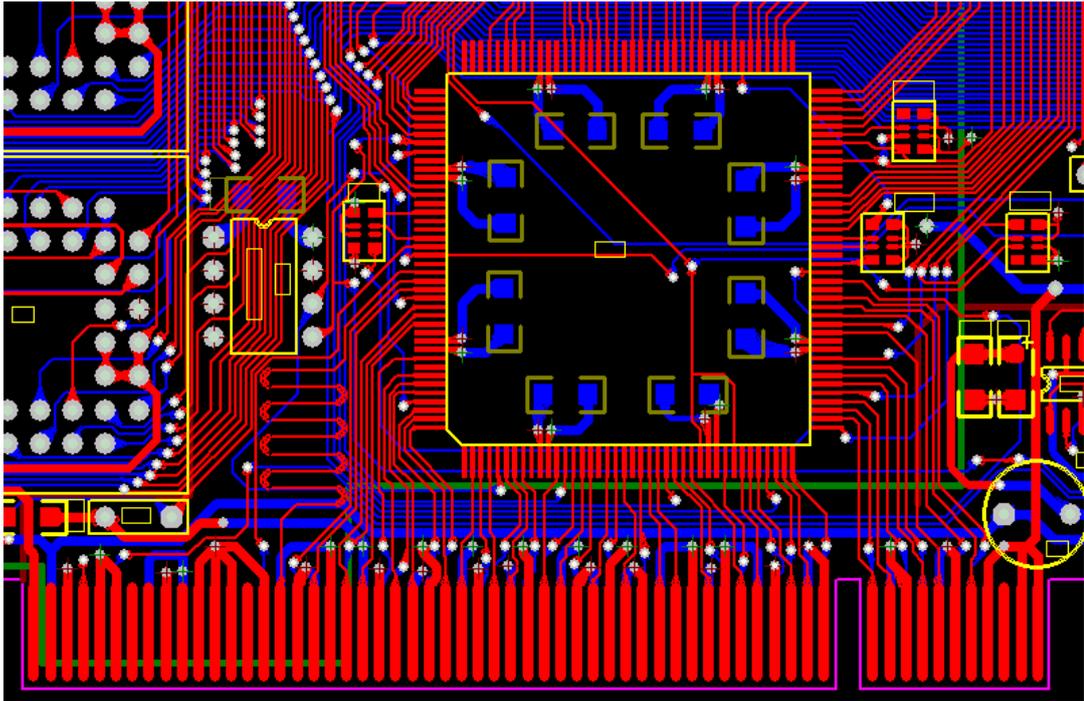


图6 S5933 的 PCI 布线例子

在 PCI 插槽的信号中，有两个信号 PRSNT1 和 PRSNT2 要留意一下，计算机主板就是靠这两个信号判断这个插槽上是否有卡、以及卡的功耗。具体含义如下：

0 表示浮空，1 表示接地		
PRSNT1	PRSNT2	
0	0	没有卡
0	1	15W
1	0	25W
1	1	7.5W

四、 S5933 的 PCI 配置

PCI 配置信息必须事先写到外接配置芯片 nvRAM 或 EEPROM 的 40h~7fh (byte), 以便在上电时, 由 S5933 加载到 PCI 配置寄存器, 如果不外接配置芯片或者外接配置芯片地址 00h 中数据为 FFh, S5933 加载默认信息到 PCI 配置寄存器, 将无法实现 FIFO 和 Pass-Thru 工作方式。PCI 配置寄存器在计算机枚举 PCI 总线时由计算机访问, PCI 协议中有专用的命令访问 PCI 配置寄存器。PCI 配置寄存器的分布如下图 7。

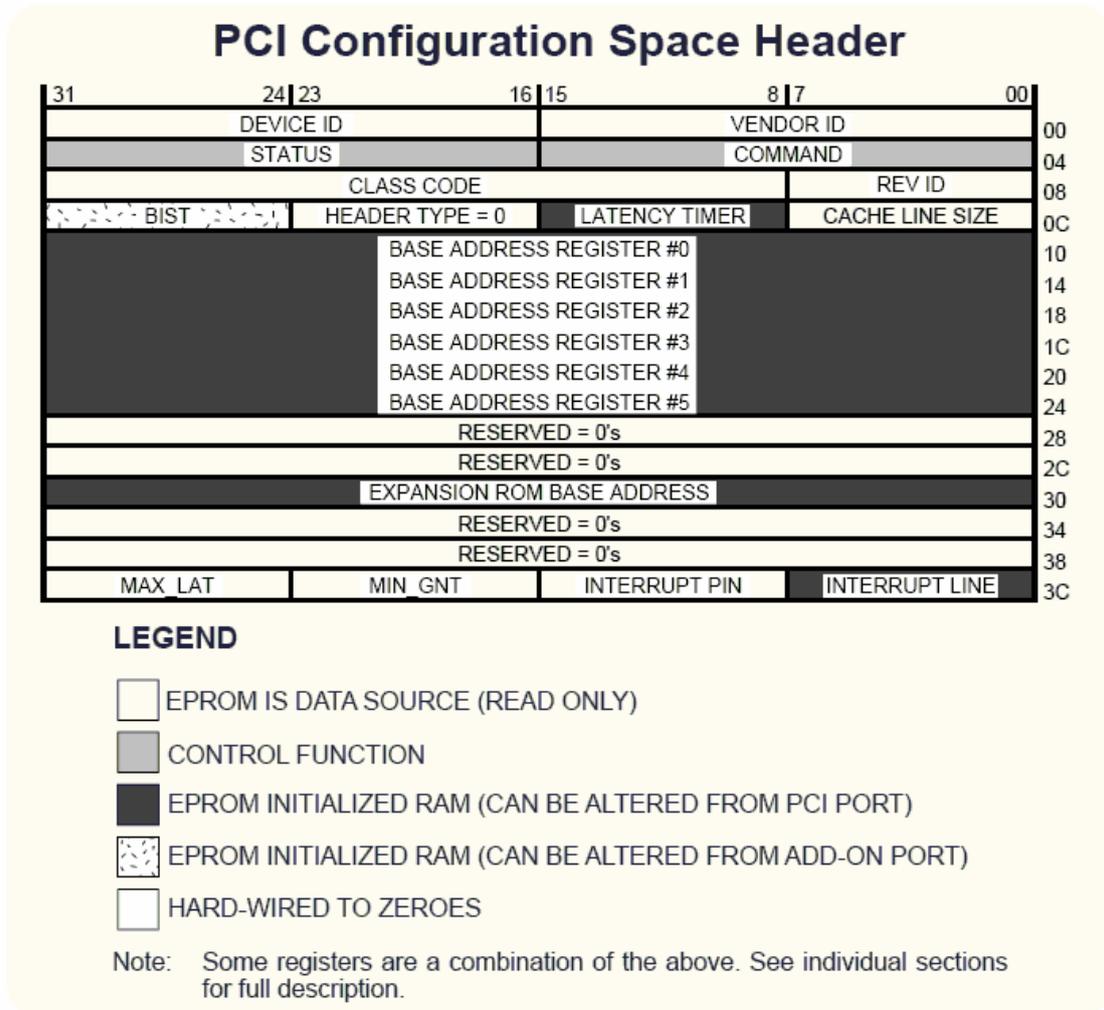


图 7 PCI 配置寄存器的分布

VENDOR ID (供应商 ID) 上电时被 S5933 设为 10E8h, 此值被扩展 nvRAM 偏移 040h~041h 的值覆盖。供应商 ID 原则上要到 PCI 协议的管理组织联系, 每年需付一定的费用。如果卡的销量不大, 可以自己随便定一个值, 最好不要与市场上的知名品牌的值一样。

DEVICE ID (设备 ID) 上电时被 S5933 设为 4750h, 此值被扩展 nvRAM 偏移 042h~043h 的值覆盖。设备 ID 原则上要到 AMCC 公司去申请。操作系统根据供应商 ID 和设备 ID 判断一个驱动程序是否属于一个设备。

COMMAD (命令寄存器) 上电值 0000h, 控制设备响应 PCI 访问的能力。它控制设备是否响应 I/O 空间或存储器空间的访问。计算机可通过 PCI 总线读写此寄存器。

STATUS (状态寄存器) 上电值 0080h, 包含了 PCI 总线相关事件信息。计算机可通过 PCI 总线读写此寄存器。

REV ID (版本 ID) 上电时被 S5933 设为 0000h, 此值被扩展 nvRAM 偏移 048h 的值覆盖。

CLASS CODE (分类代码) 上电时被 S5933 设为 FF0000h, 此值被扩展 nvRAM 偏移 049h~4Bh

的值覆盖。

CACHE LINE SIZE (Cache 行容量寄存器), 在 S5933 中硬件赋值 00h, 不能被改写。

LATENCY TIMER (延迟定时器) 上电时被 S5933 设为 00h, 此值被扩展 nvRAM 偏移 04Dh 的值覆盖。延迟定时器定义了 PCI 交易周期的最小量。

HEADER TYPE (首部类型寄存器) 上电时被 S5933 设为 00h, 此值被扩展 nvRAM 偏移 04Eh 的值覆盖。首部类型寄存器的 bit7=0 表示此设备是单功能设备, 否则为多功能设备。

BIST (内置自检寄存器) 上电时被 S5933 设为 00h, 此值被扩展 nvRAM 偏移 04Fh 的值覆盖。

EXPANSION ROM BASE ADDRESS (扩展 BIOS 基地址寄存器) 上电时被 S5933 设为 00000000h, 此值被扩展 nvRAM 偏移 70h~73h 的值覆盖。

INTERRUPT LINE (中断线寄存器) 上电时被 S5933 设为 FFh, 此值被扩展 nvRAM 偏移 7Ch 的值覆盖。中断线寄存器用于识别功能的 PCI 中断请求引脚路由到中断控制器的那个输入。例如, 在 PC 环境中, 这个寄存器值 00h 至 0Fh 对应中断控制器的 IRQ0 到 IRQ15, 值 FFh 表示“无连接”, 值 10h 到 FEh 保留。

INTERRUPT PIN (中断引脚寄存器) 上电时被 S5933 设为 01h, 此值被扩展 nvRAM 偏移 7Dh 的值覆盖。值 01h 到 04h 对应于 PCI 插槽中断请求引脚 INTA#至 INTD#。

Min_Gnt (时间片请求寄存器) 上电时被 S5933 设为 00h, 此值被扩展 nvRAM 偏移 7Eh 的值覆盖。这个寄存器只由总线主设备实现, 寄存器值表示只要主设备启动一个交易, 它就将 PCI 总线所有权保持多久(以达到好的性能)。这个值表示设备需要多久的突发段(以 250ns 递增), 00h 值表示设备对此没有严格要求。

Max_Lat (优先级请求) 上电时被 S5933 设为 00h, 此值被扩展 nvRAM 偏移 7Fh 的值覆盖。此寄存器值表示设备需要“多快”访问 PCI 总线(以 250ns 递增), 00h 值表示设备对此没有严格要求。

BASE ADDRESS REGISTER #0 对应 PCI 操作寄存器的基地址, BASE ADDRESS REGISTER #1~5 分别对应 4 个 Pass-Thru 通道基地址。基地址寄存器在计算机初始化时, 以查询方式确定映射方式(I/O 或内存映射)、每个基地址上要实现的存储器大小、IO 的地址范围和 IO 空间的大小, 然后由计算机写入物理地址, 具体过程参考 PCI 协议。

另外, nvRAM 或 EEPROM 的偏移 45h 在 S5993 上有特殊定义, 否则 FIFO 不能使用, 45h 的定义如下。

Bit 7 Bus Master Register Access

0 读/写地址和读/写计数寄存器只能在 ADD-ON 端访问, 即 S5933 是主设备。

1 读/写地址和读/写计数寄存器只能在 PCI 端访问, 即 S5933 是从设备。(default)

Bit 6 RDFIFO#, RD# Operation

0 同步模式 - RDFIFO#和 RD#用作使能信号, S5933 用 PCI 时钟上升沿触发数据。

1 异步模式 - S5933 用 RDFIFO#和 RD#上升沿触发数据。(default)

Bit 5 WRFIFO#, WR# Operation

0 同步模式 - WDFIFO#和 WD#用作使能信号, S5933 用 PCI 时钟上升沿触发数据。

1 异步模式 - S5933 用 WDFIFO#和 WD#上升沿触发数据。(default)

Bit 0 Target Latency Timer Enable

0 Disable PCI Latency Timer Time Out - Will not disconnect with retry if cannot issue TRDY in specified time

1 Enable PCI Latency Timer Time Out - Will be PCI 2.1 compliant

AMCC 提供一个小程序 NVBUILD.EXE 可以方便地产生配置 S5933 的数据, 在纯 DOS 下, 甚至可以用此程序把数据直接写入串行 nvRAM 或 EEPROM(当然开机前必须把 PCI 卡插入 PCI

插槽)。以下简单介绍如何使用此程序。

- 1) 在 DOS 提示符下输入 NVBUILD.EXE 回车，或在 WINDOWS 下双击 NVBUILD.EXE。出现下图 8 画面，用光标键选择菜单，回车进入选择菜单的子菜单。
“Load Memory Image” 主要用于从文件或卡上串行 nvRAM 或 EEPROM 中上载数据，可识别 .bin 或 .hex 文件。
“Save Memory Image” 用于把当前数据存入文件，可存为 .bin 或 .hex 文件。
“Edit Memory Image” 用于修改编辑配置数据。
“Write to Device” 把当前配置数据写入 S5933 外接的串行 nvRAM 或 EEPROM。
“Display Memory Image” 用 16 进制显示当前配置数据。
“Exit” 推出程序。

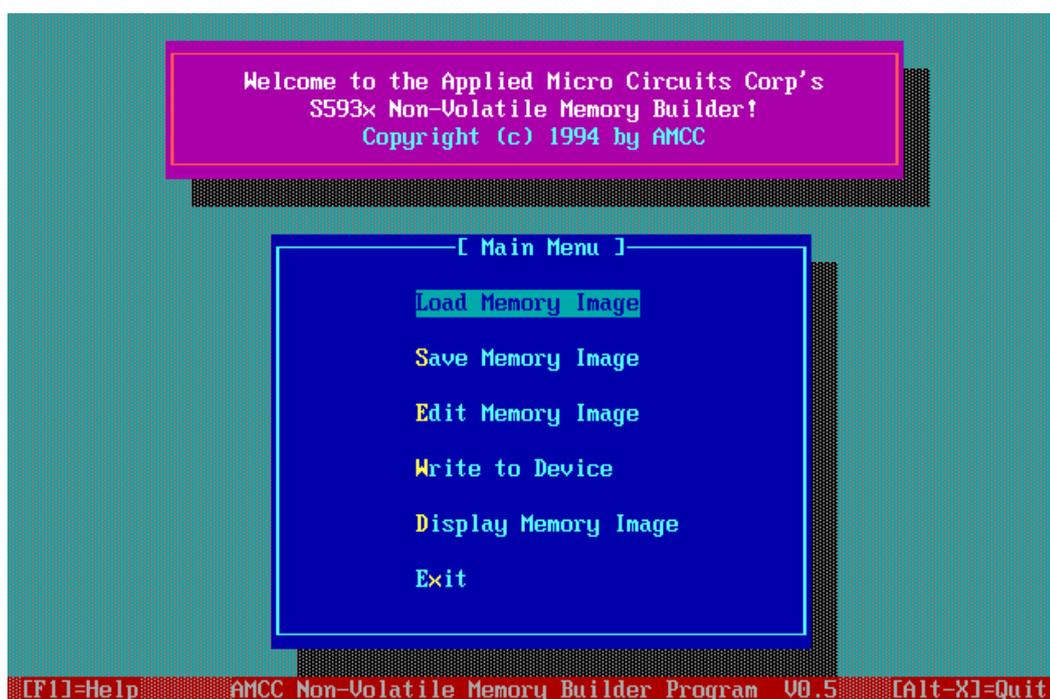


图 8 NVBUILD.EXE 主菜单

- 2) 选 “Edit Memory Image” 菜单回车，出现图 9。

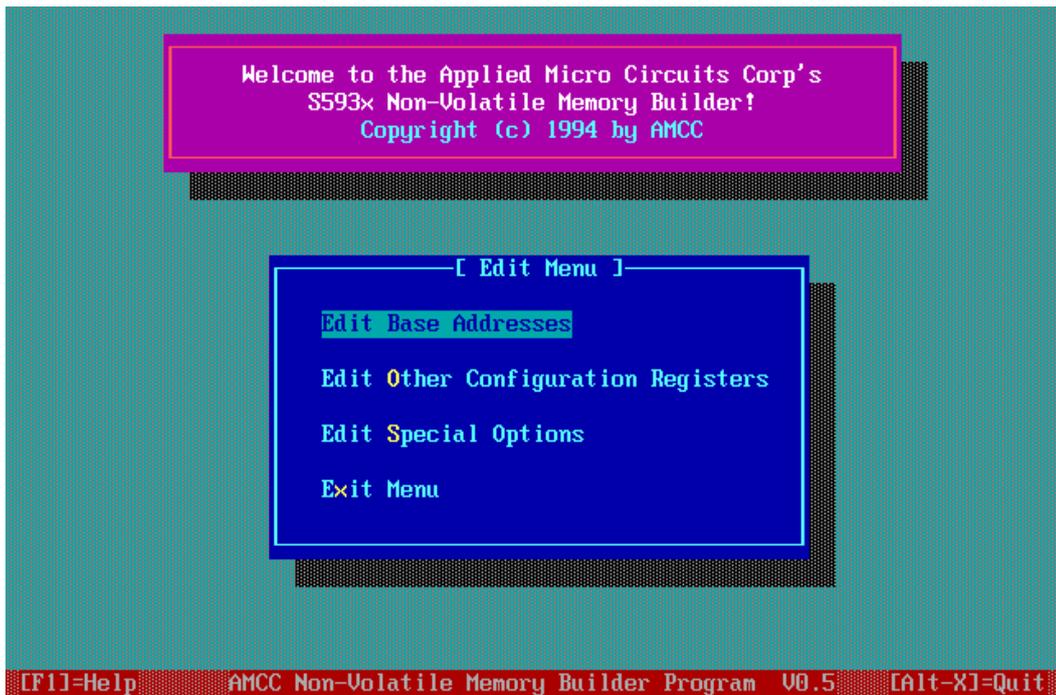


图9 编辑子菜单

“Edit Base Addresses” 菜单用于编辑 BASE ADDRESS REGISTER #1~5 的属性，这四个基地址分别对应 4 个 Pass-Thru 通道基地址。

“Edit Other Configuration Registers” 菜单用于编辑 ‘DEVICE ID’、‘VENDOR ID’ 等寄存器的内容。

“Edit Special Options” 菜单用于编辑 S5933 是主/从设备、FIFO 操作是否用同步模式，即串行 EEPROM 中 45h 的内容。

3) 选 “Edit Base Addresses” 菜单回车，出现图 10。

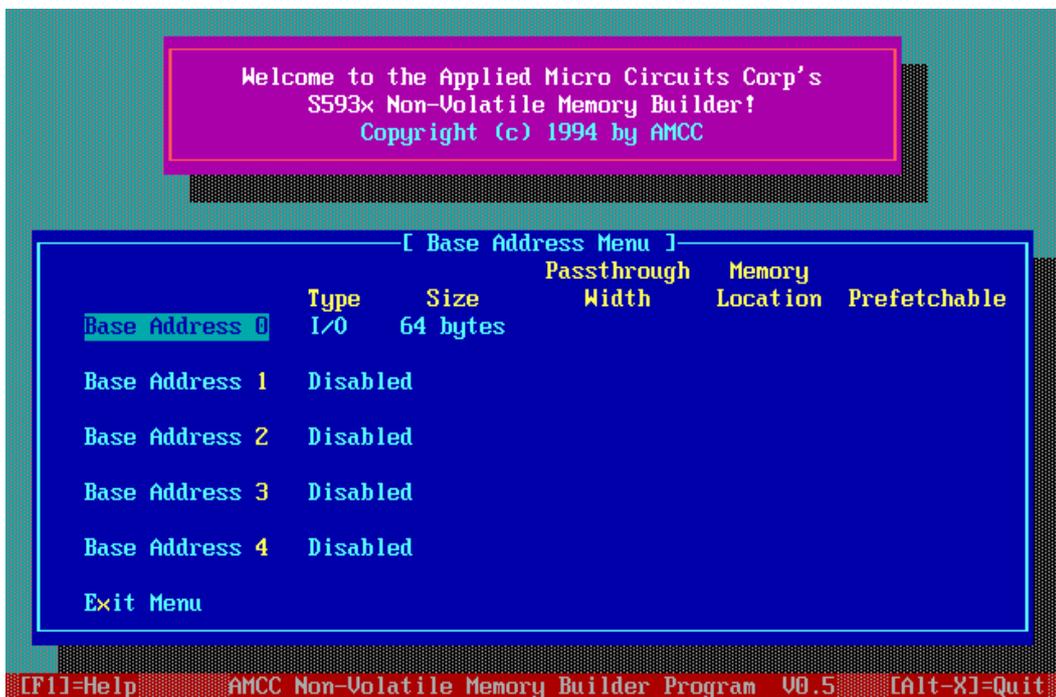


图 10

在图 10 下，可选一个基地址进行配置，本例配置 “Base Address 1”，把光标移动

到“Base Address 1”回车，出现下图 11。

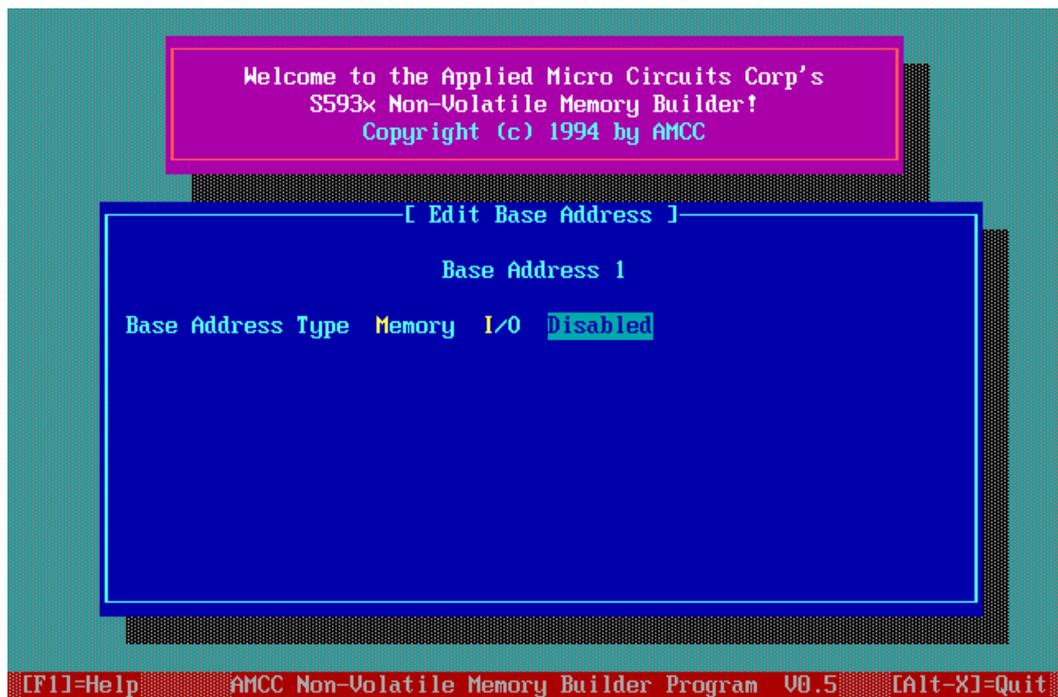


图 11 “Base Address 1”子菜单

可选“Memory”把“Base Address 1”配置到内存空间，选“I/O”把“Base Address 1”配置到 I/O 空间。若选“I/O”回车，出现图 12。

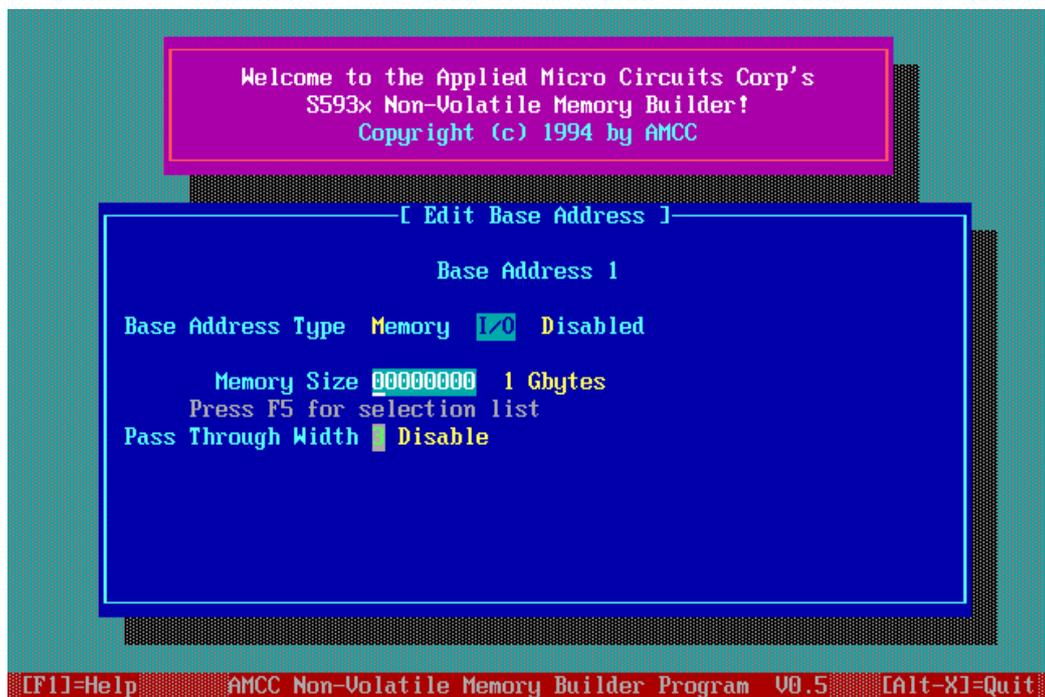


图 12

按功能键“F5”出现图 13。

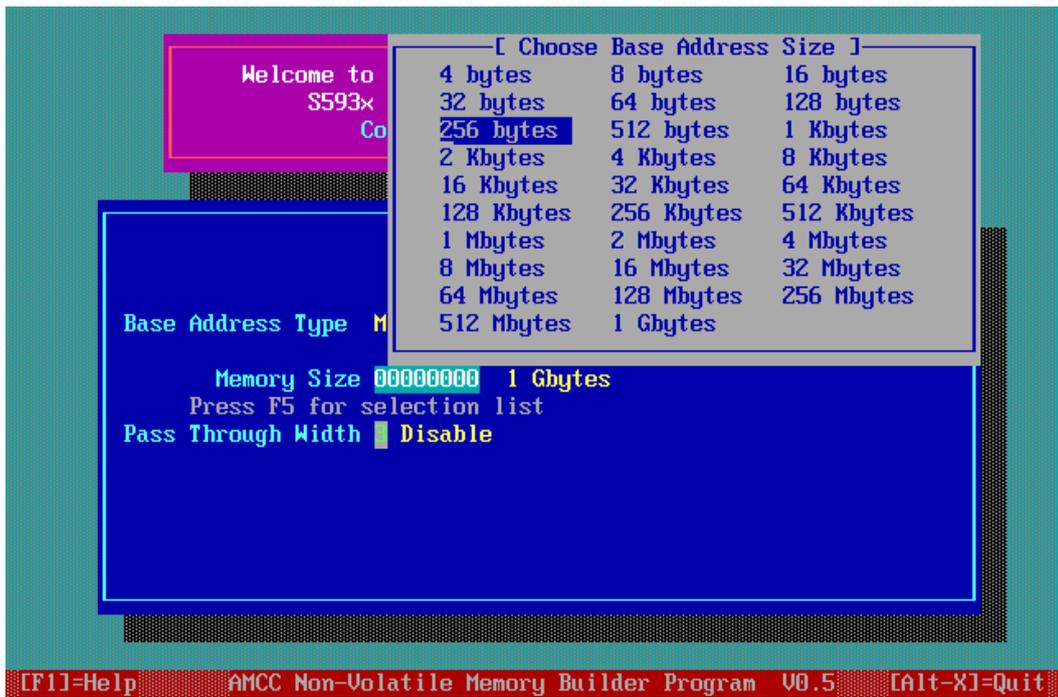


图 13

用光标键选 I/O 空间大小，本例选“256bytes”回车，出现下图 14。

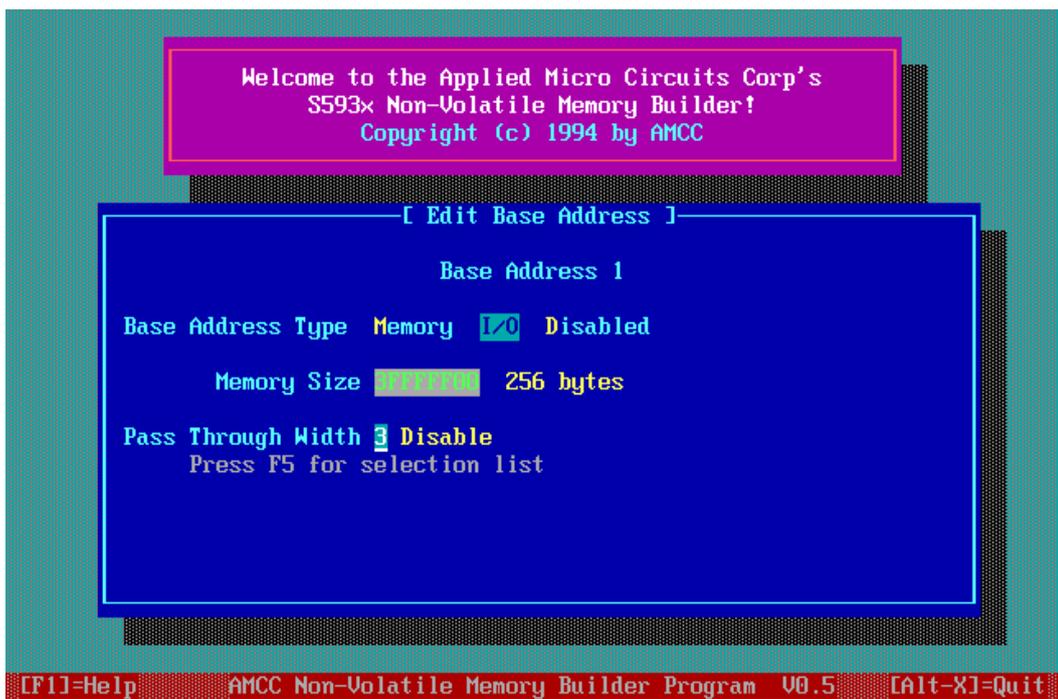


图 14

按功能键“F5”出现图 15，选数据总线宽度。

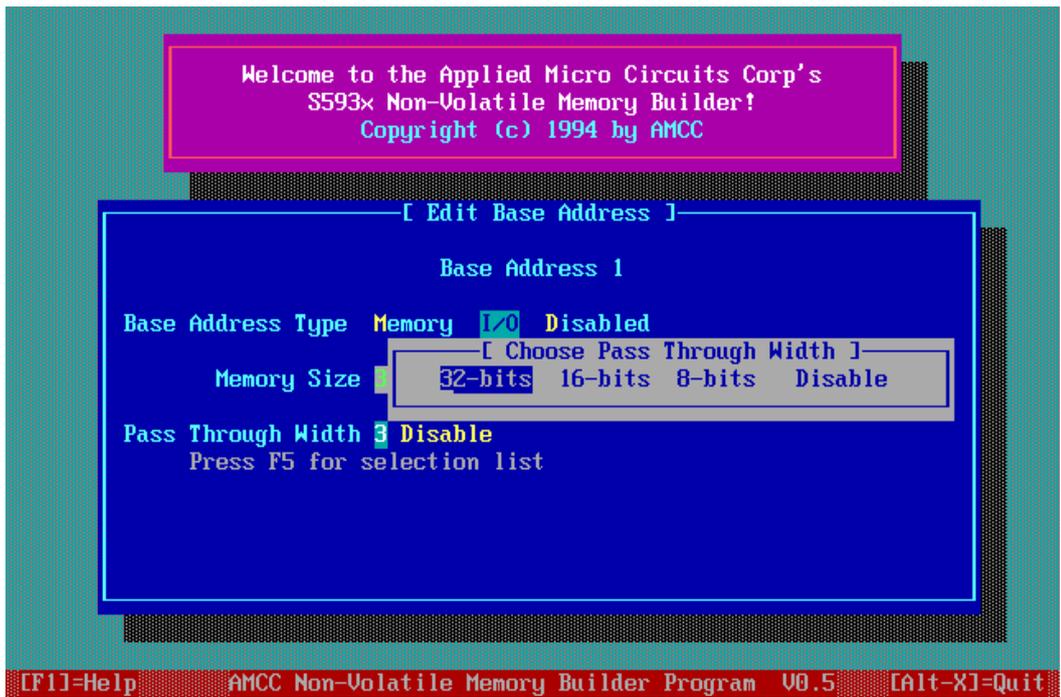


图 15

用光标键数据总线宽度，本例选“32-bites”回车后，再按一次回车出现下图 16。

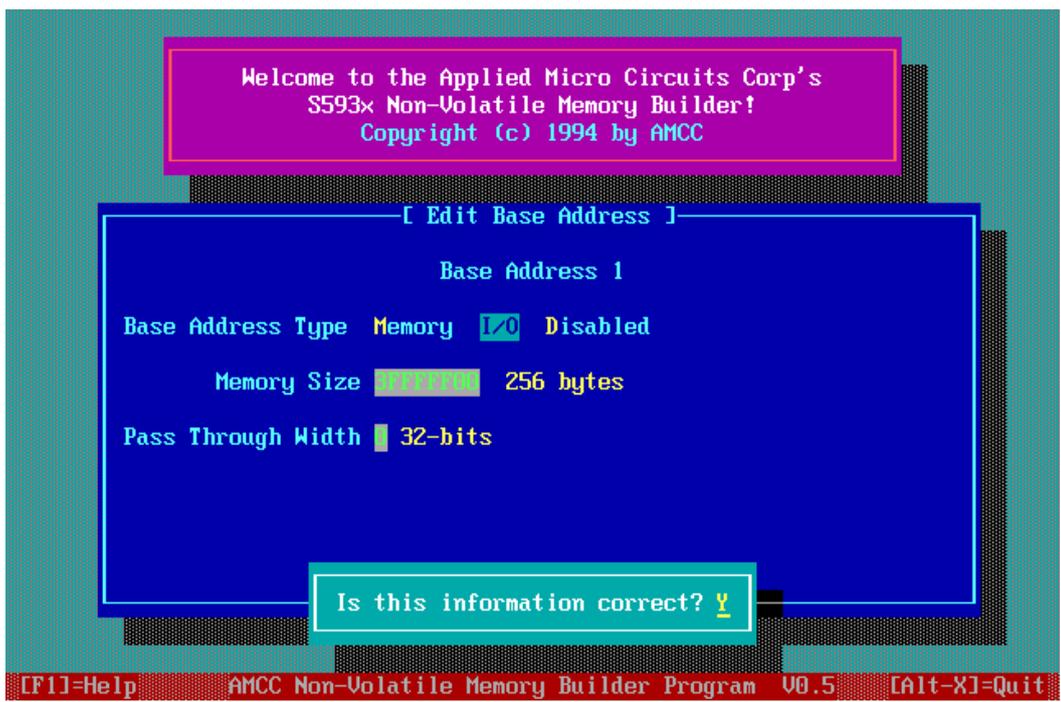


图 16

在图 16 的情况下，再按一次回车出现下图 17。

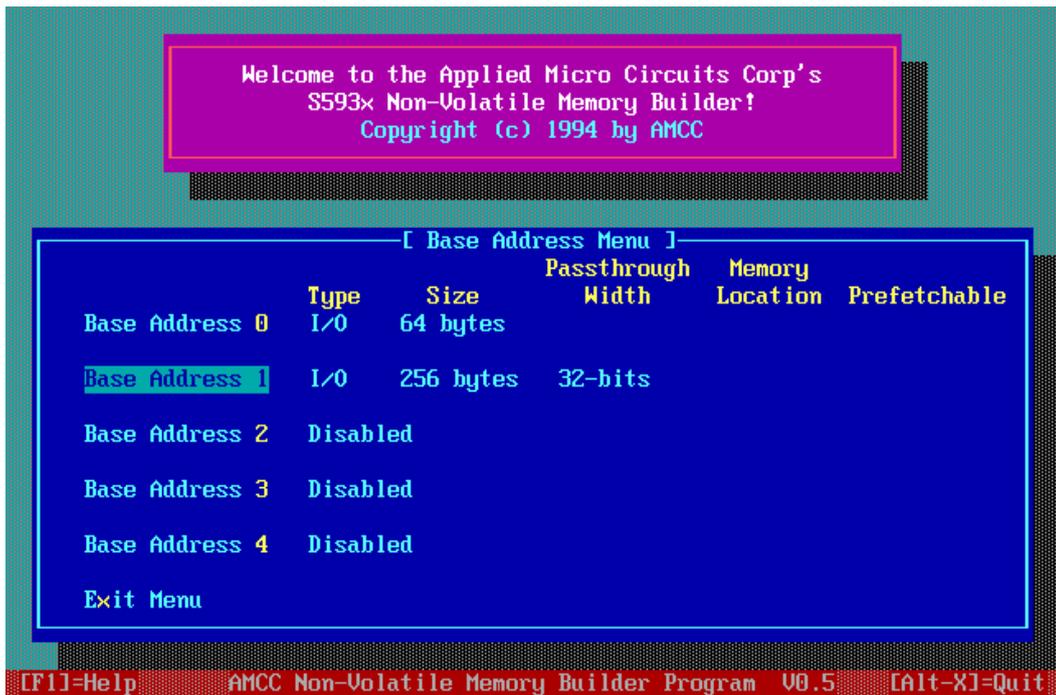


图 17

在图 17 时，说明“Base Address 1”已经配置到 I/O 空间，占用 256bytes 的 I/O 空间，并且数据总线宽度为 32bits。

重复以上步骤可配置其它基地址。

4) 在图 9 时，选“Edit Other Configuration Registers”菜单回车，出现图 18。

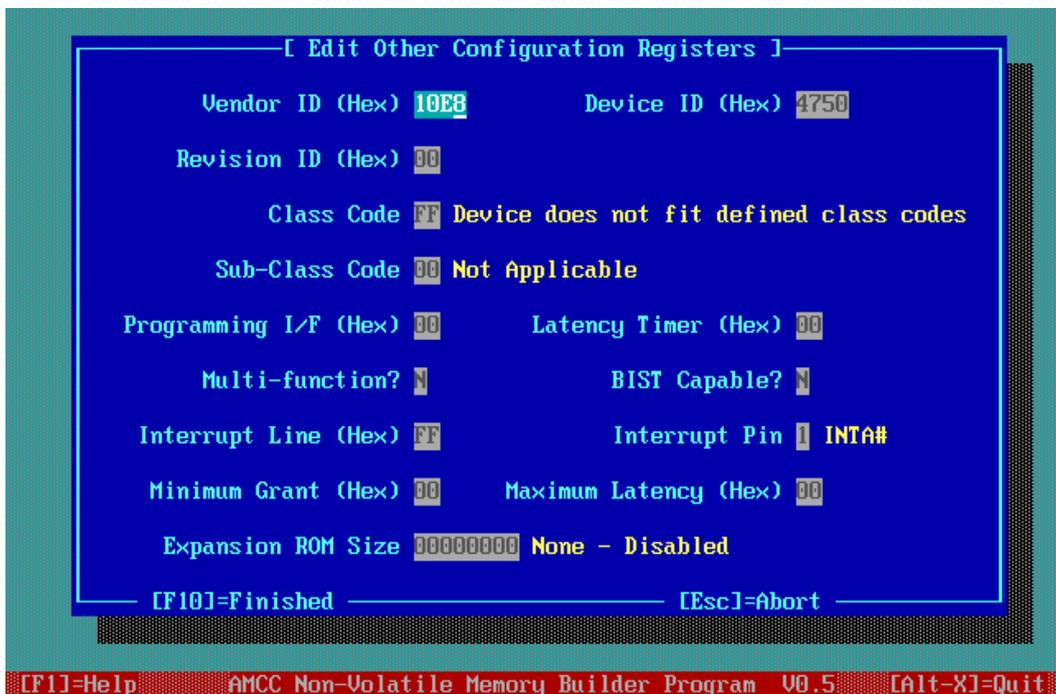


图 18

在显示图 18 菜单时，可以编辑“Vender ID”等值，所有寄存器值设置完以后，按功能键“F10”确认，菜单显示回到图 9。

5) 在图 9 时，选“Edit Special Options”菜单回车，出现图 19。

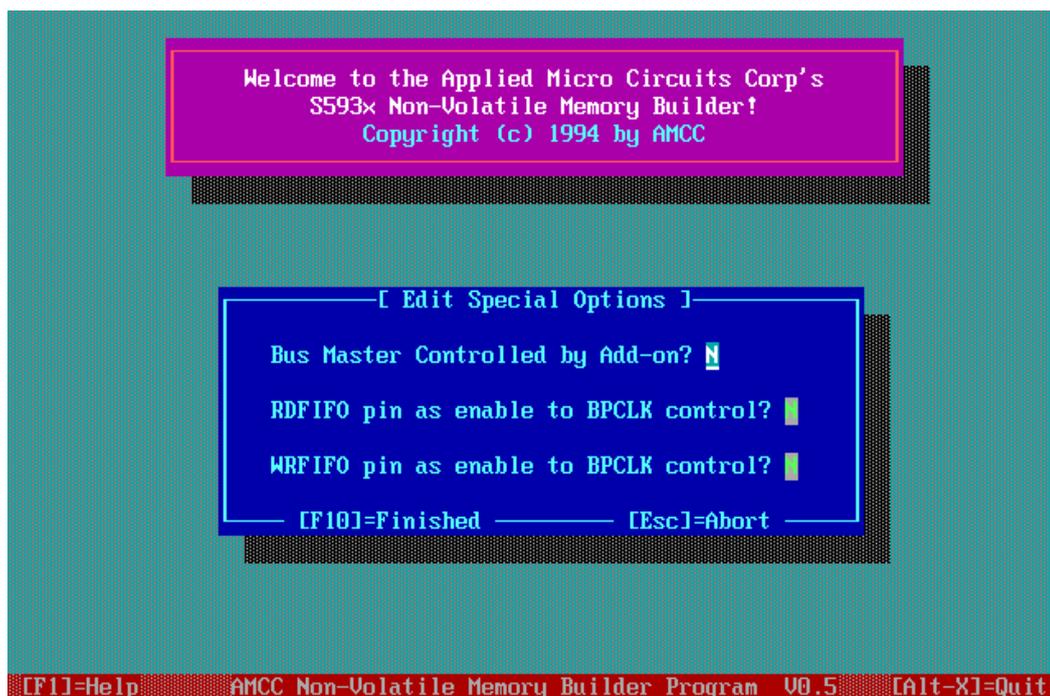


图 19

在显示图 19 菜单时，可以编辑编辑 S5933 是主/从设备、FIFO 操作是否用同步模式，所有值设置完以后，按功能键“F10”确认，菜单显示回到图 9。

6) 至此所有设置完成，在图 9 下选“Exit Menu”回车后，回到菜单 8。在菜单 8 下保存数据以备以后使用，也可直接把数据写入串行配置芯片。

五、 PCI 总线操作寄存器组 (PCI BUS OPERATION REGISTERS)

PCI 总线操作寄存器共有 16 个 DWORD 寄存器，地址空间 (IO or memory) 分配在 BASE ADDRESS REGISTER #0 上，如下图 20 所示。这些寄存器只能由主机通过 PCI 总线访问。

Address Offset	Abbreviation	Register Name
00h	OMB1	Outgoing Mailbox Register 1
04h	OMB2	Outgoing Mailbox Register 2
08h	OMB3	Outgoing Mailbox Register 3
0Ch	OMB4	Outgoing Mailbox Register 4
10h	IMB1	Incoming Mailbox Register 1
14h	IMB2	Incoming Mailbox Register 2
18h	IMB3	Incoming Mailbox Register 3
1Ch	IMB4	Incoming Mailbox Register 4
20h	FIFO	FIFO Register port (bidirectional)
24h	MWAR	Master Write Address Register
28h	MWTC	Master Write Transfer Count Register
2Ch	MRAR	Master Read Address Register
30h	MRTC	Master Read Transfer Count Register
34h	MBEF	Mailbox Empty/Full Status
38h	INTCSR	Interrupt Control/Status Register
3Ch	MCSR	Bus Master Control/Status Register

图 20 PCI 总线操作寄存器

- 1) Outgoing Mailbox 寄存器用于把 PCI 总线上的少量数据传 (如命令、参数) 送到 Add-On 端, Incoming Mailbox 用于把 Add-On 端的少量数据传送到 PCI 总线上。FIFO 寄存器是双向的, 用于 PCI 总线与 Add-On 端的大批量数据交换。
- 2) 写地址寄存器 MWAR, 当 S5933 为 PCI 总线从设备时, PCI 总线端访问此寄存器才有意义。用 FIFO 把数据从 Add-On 端传送到 PCI 总线时, 此寄存器的内容指向 PC 主机内存的地址, 每传送一个 DWORD 数据此寄存器的内容自动加 4, 以指向下一个待写内存的地址。此寄存器的低两 bits 总保持 0, 所以传送的数据必须以 DWORD 为边界。
- 3) 写计数器寄存器 MWTC, 当 S5933 为 PCI 总线从设备时, PCI 总线端访问此寄存器才有意义。此寄存器与写地址寄存器 MWAR 配合一起使用, 传送开始前给此寄存器写入需要传送的数据总量 (bytes), 以后每传送一个 DWORD 数据此寄存器的内容自动减 4, 当此寄存器的 bit3~bit25 为 0 时表示本次传送完成, S5933 给 PCI 总线或 Add-On 端产生一个中断。当此寄存器的 bit0~bit25 可以读写, bit26~bit31 被 S5933 硬件保持为 0。
- 4) 读地址寄存器 MRAR, 当 S5933 为 PCI 总线从设备时, PCI 总线端访问此寄存器才有意义。用 FIFO 把数据从 PCI 总线传送到 Add-On 端时, 此寄存器的内容指向 PC 主机内存的地址, 每传送一个 DWORD 数据此寄存器的内容自动加 4, 以指向下一个待读内存的地址。此寄存器的低两 bits 总保持 0, 所以传送的数据必须以 DWORD 为边界。
- 5) 读计数器寄存器 MRTC, 当 S5933 为 PCI 总线从设备时, PCI 总线端访问此寄存器才有意义。此寄存器与读地址寄存器 MRAR 配合一起使用, 传送开始前给此寄存器写入需要传送的数据总量 (bytes), 以后每传送一个 DWORD 数据此寄存器的内容自动减 4, 当此寄存器的 bit3~bit25 为 0 时表示本次传送完成, S5933 给 PCI 总线或

Add-On 端产生一个中断。当此寄存器的 bit0~bit25 可以读写, bit26~bit31 被 S5933 硬件保持为 0。

6) Mailbox 空满状态寄存器 MBEF, 此寄存器只读, 每一 bit 的意义如下:

Bit	描述
31: 16	<p>Incoming Mailbox Status, 指示某个 Incoming Mailbox 已经被 Add-On 端写入, 但没有被 PCI 总线端取走。一个 bit 代表 Mailbox 的一个 byte, 若某一个 bit 为 1 表示某个 byte 为满状态。</p> <p>Bit31 = Incoming Mailbox 4 byte 3 Bit30 = Incoming Mailbox 4 byte 2 Bit29 = Incoming Mailbox 4 byte 1 Bit28 = Incoming Mailbox 4 byte 0 Bit27 = Incoming Mailbox 3 byte 3 Bit26 = Incoming Mailbox 3 byte 2 Bit25 = Incoming Mailbox 3 byte 1 Bit24 = Incoming Mailbox 3 byte 0 Bit23 = Incoming Mailbox 2 byte 3 Bit22 = Incoming Mailbox 2 byte 2 Bit21 = Incoming Mailbox 2 byte 1 Bit20 = Incoming Mailbox 2 byte 0 Bit19 = Incoming Mailbox 1 byte 3 Bit18 = Incoming Mailbox 1 byte 2 Bit17 = Incoming Mailbox 1 byte 1 Bit16 = Incoming Mailbox 1 byte 0</p>
15: 0	<p>Outgoing Mailbox Status, 指示某个 Outgoing Mailbox 已经被 PCI 总线端写入, 但没有被 Add-On 端取走。一个 bit 代表 Mailbox 的一个 byte, 若某一个 bit 为 1 表示某个 byte 为满状态。</p> <p>Bit15 = Outgoing Mailbox 4 byte 3 Bit14 = Outgoing Mailbox 4 byte 2 Bit13 = Outgoing Mailbox 4 byte 1 Bit12 = Outgoing Mailbox 4 byte 0 Bit11 = Outgoing Mailbox 3 byte 3 Bit10 = Outgoing Mailbox 3 byte 2 Bit09 = Outgoing Mailbox 3 byte 1 Bit08 = Outgoing Mailbox 3 byte 0 Bit07 = Outgoing Mailbox 2 byte 3 Bit06 = Outgoing Mailbox 2 byte 2 Bit05 = Outgoing Mailbox 2 byte 1 Bit04 = Outgoing Mailbox 2 byte 0 Bit03 = Outgoing Mailbox 1 byte 3 Bit02 = Outgoing Mailbox 1 byte 2 Bit01 = Outgoing Mailbox 1 byte 1 Bit00 = Outgoing Mailbox 1 byte 0</p>

7) 中断配置/状态寄存器 INTCSR, 此寄存器的上电值为 00000000h。用此寄存器设置产生 PCI 总线中断的条件, 以及读此寄存器的有关 bit 判断当前中断产生的源, 和

写此寄存器的有关 bit 来清除当前中断。中断源有 6 个，分别为：写计数器到达 0、读计数器到达 0、某个 Outgoing Mailbox 变空、某个 Incoming Mailbox 变满、目标设备异常、主设备异常。此寄存器的结构如下图 21。

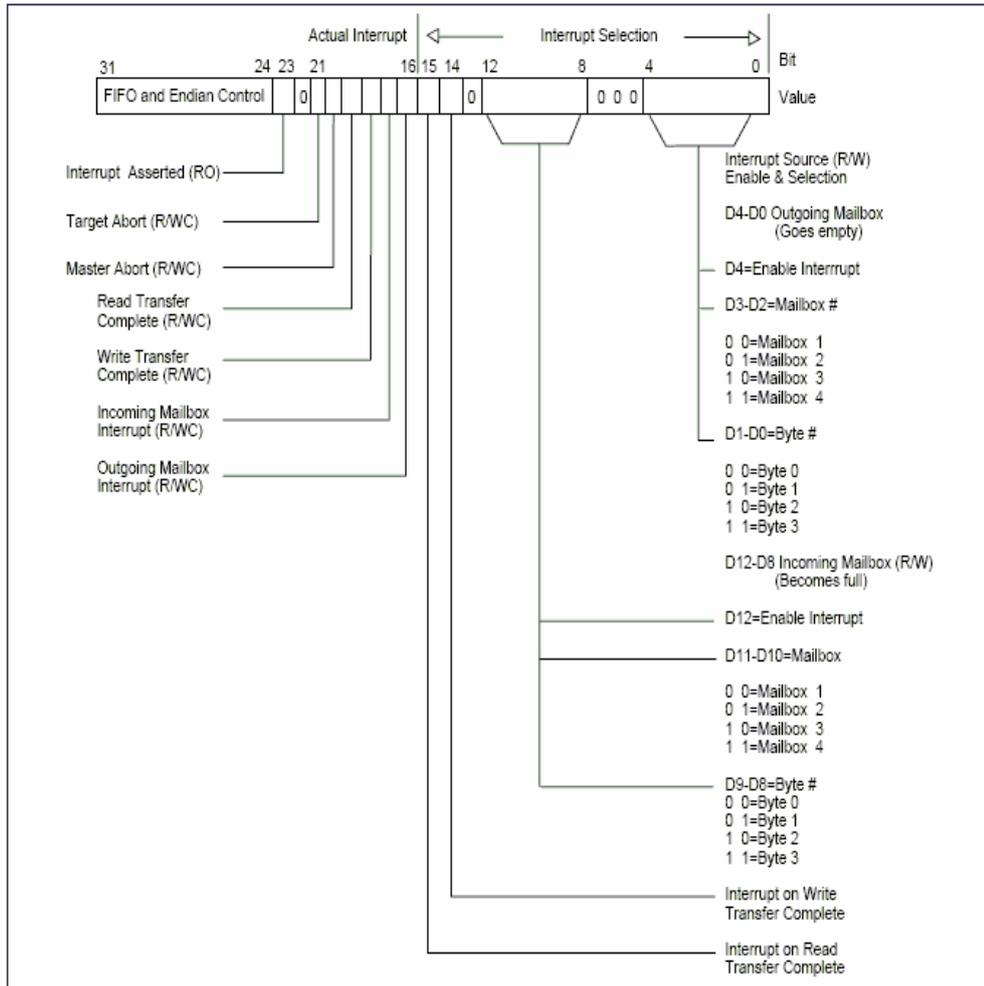


图 21 INTERRUPT CONTROL/STATUS REGISTER (INTCSR)

INTCSR 每个 bit 的具体意义如下：

Bit	描述
31: 24	FIFO and Endian 控制
23	有中断产生，这是一个只读状态位。此 bit 为 1 表示有 1 个或多个中断源发生中断，它是此寄存器 bit19 至 bit16 的逻辑或。
22	保留。
21	目标设备异常中断。在一个 PCI 总线周期中，当 S5933 为当前总线主设备时，若 S5933 检测到目标设备异常，S5933 就置 1 此 bit 给 PCI 总线产生中断。写 1 到此 bit 将复位此 bit 为 0，写 0 到此 bit 不影响此 bit 的状态。
20	主设备异常中断。在一个 PCI 总线周期中，当 S5933 为当前总线主设备时，若没有目标设备响应，S5933 就置 1 此 bit 给 PCI 总线产生中断。写 1 到此 bit 将复位此 bit 为 0，写 0 到此 bit 不影响此 bit 的状态。
19	读传送完成中断。在读传送进行中，若读计数器寄存器 MRTC 到达 0，S5933 就置 1 此 bit 给 PCI 总线产生中断。写 1 到此 bit 将复位此 bit 为 0，写 0 到此 bit 不影响此 bit 的状态。

18	写传送完成中断。在写传送进行中,若写计数器寄存器 MWTC 到达 0, S5933 就置 1 此 bit 给 PCI 总线产生中断。写 1 到此 bit 将复位此 bit 为 0, 写 0 到此 bit 不影响此 bit 的状态。
17	Incoming Mailbox 变满中断。当本寄存器的 bit12 到 bit8 所选的 Mailbox 被 Add-On 端写时, S5933 就置 1 此 bit 给 PCI 总线产生中断。写 1 到此 bit 将复位此 bit 为 0, 写 0 到此 bit 不影响此 bit 的状态。
16	Outgoing Mailbox 变空中断。当本寄存器的 bit4 到 bit0 所选的 Mailbox 被 Add-On 端读时, S5933 就置 1 此 bit 给 PCI 总线产生中断。写 1 到此 bit 将复位此 bit 为 0, 写 0 到此 bit 不影响此 bit 的状态。
15	给此 bit 写 1 使能读传送完成时产生中断。
14	给此 bit 写 1 使能写传送完成时产生中断。
13	保留。
12	给此 bit 写 1 使能 Incoming Mailbox 变满时产生中断。
11: 10	Incoming Mailbox 中断选择, 此 2bits 的值选择那个 Incoming Mailbox 是中断源。[00]b 选择 Incoming Mailbox 1, [01]b 选择 Incoming Mailbox 2, [10]b 选择 Incoming Mailbox 3, [11]b 选择 Incoming Mailbox 4。
9: 8	Incoming Mailbox byte 中断选择, 此 2bits 的值选择由 bit11 到 10 指定的 Incoming Mailbox 中那个 byte 为中断源。[00]b 选择 byte 0, [01]b 选择 byte 1, [10]b 选择 byte 2, [11]b 选择 byte 3。
7: 5	保留。
4	给此 bit 写 1 使能 Outgoing Mailbox 变空时产生中断。
3: 2	Outgoing Mailbox 中断选择, 此 2bits 的值选择那个 Outgoing Mailbox 是中断源。[00]b 选择 Outgoing Mailbox 1, [01]b 选择 Outgoing Mailbox 2, [10]b 选择 Outgoing Mailbox 3, [11]b 选择 Outgoing Mailbox 4。
1: 0	Outgoing Mailbox byte 中断选择, 此 2bits 的值选择由 bit3 到 2 指定的 Outgoing Mailbox 中那个 byte 为中断源。[00]b 选择 byte 0, [01]b 选择 byte 1, [10]b 选择 byte 2, [11]b 选择 byte 3。

- 8) 主设备控制/状态寄存器 MCSR, 此寄存器的上电值为 000000E6h。此寄存器的结构如下图 22。

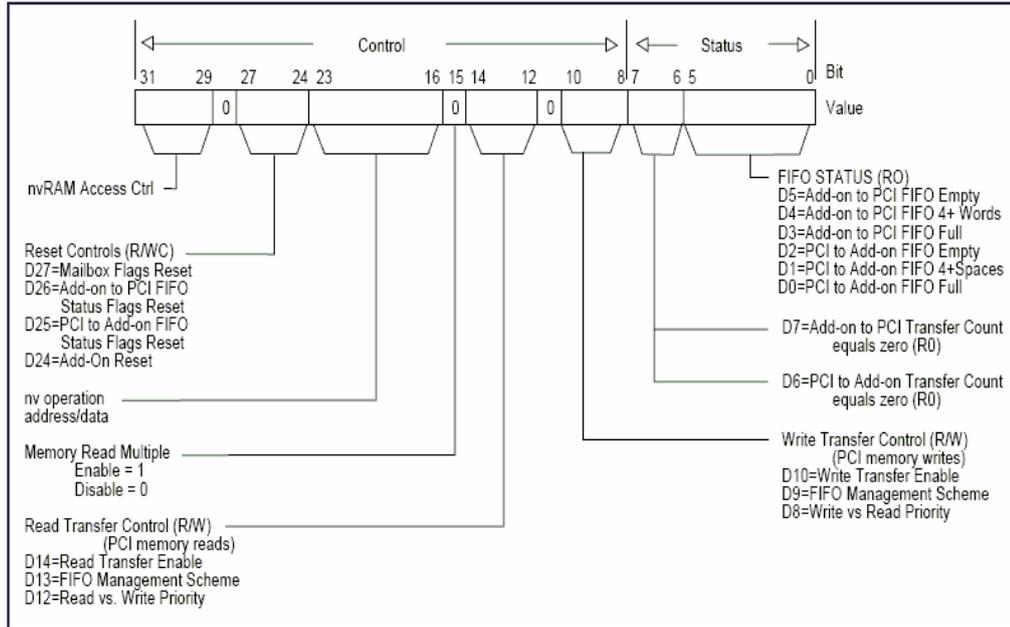


图 22 MASTER CONTROL/STATUS REGISTER (MCSR)

MCSR 每个 bit 的具体意义如下:

Bit	描述																																								
31: 29	nvRAM 访问控制信号。用这几 bits 和 bit23 至 bit16 可完成扩展 nvRAM 的访问。访问时, 先写 low-order 地址, 再写 high-order 地址, 最后访问数据 data 字节。Bit31 即用作使能, 又用作准备好信号, 若要启动一个访问, 必须给 bit31 写入 1, 而下一个访问必须等到 bit31 变为 0 时才可继续。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>D31</th> <th>D30</th> <th>D29</th> <th>W/R</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>X</td> <td>W</td> <td>Inactive</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>W</td> <td>Load low address byte</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>W</td> <td>Load high address byte</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>W</td> <td>Begin write</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>W</td> <td>Begin read</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>R</td> <td>Ready</td> </tr> <tr> <td>1</td> <td>X</td> <td>X</td> <td>R</td> <td>Busy</td> </tr> </tbody> </table>	D31	D30	D29	W/R		0	X	X	W	Inactive	1	0	0	W	Load low address byte	1	0	1	W	Load high address byte	1	1	0	W	Begin write	1	1	1	W	Begin read	0	X	X	R	Ready	1	X	X	R	Busy
D31	D30	D29	W/R																																						
0	X	X	W	Inactive																																					
1	0	0	W	Load low address byte																																					
1	0	1	W	Load high address byte																																					
1	1	0	W	Begin write																																					
1	1	1	W	Begin read																																					
0	X	X	R	Ready																																					
1	X	X	R	Busy																																					
28	此 bit 为 1 表示把 FIFO 接成回环模式。																																								
27	Mailbox flag Reset。给此 bit 写入 1 复位所有 Mailbox 的状态标志, 即所有 Mailbox 的状态将变为空 (EMPTY)。给此 bit 写入 1 后没有必要再写入 0 复位此 bit, 读此 bit 永远返回 0。																																								
26	Add-On to PCI FIFO Status Reset。给此 bit 写入 1 把 Add-On to PCI FIFO 的状态复位到空 (EMPTY)。给此 bit 写入 1 后没有必要再写入 0 复位此 bit, 读此 bit 永远返回 0。																																								
25	PCI to Add-On FIFO Status Reset。给此 bit 写入 1 把 PCI to Add-On FIFO 0 的状态复位到空 (EMPTY)。给此 bit 写入 1 后没有必要再写入 0 复位此 bit, 读此 bit 永远返回 0。																																								
24	Add-On pin reset。给此 bit 写入 1 将使 Add-On 接口的复位引脚 SYSRST#变为有效, 必须写入 0 才能使 SYSRST#变为无效。																																								

32: 16	扩展 nvRAM 的 address/data 接口。
15	在 S5933 为总线主设备时，使能多重内存读。
14	读传送使能。给此 bit 写入 1，若此时读计数器寄存器非 0，S5933 就启动读传送。给此 bit 写入 0 将挂起当前传送。
13	读 FIFO 方案。若此 bit 置 1，只有在空 FIFO 的数量等于或大于 4 时，S5933 才会请求 PCI 总线控制权。若此 bit 置 0，只有在空 FIFO 的数量等于或小于 1 时，S5933 才会请求 PCI 总线控制权。如果 S5933 已经控制 PCI 总线，此 bit 的值无意义。
12	读与写优先级。若此 bit 为 1，而 bit8 为 0，读传送优先级高于写传送。若此 bit 为 1，而 bit8 为 1，优先级在读传送与写传送之间交替。
11	保留。
10	写传送使能。给此 bit 写入 1，若此时写计数器寄存器非 0，S5933 就启动写传送。给此 bit 写入 0 将挂起当前传送。
9	写 FIFO 方案。若此 bit 置 1，只有在填写 FIFO 的数量等于或大于 4 时，S5933 才会请求 PCI 总线控制权。若此 bit 置 0，在填写 FIFO 的数量等于 1 时，S5933 就请求 PCI 总线控制权。如果 S5933 已经控制 PCI 总线，此 bit 的值无意义。
8	写与读优先级。若此 bit 为 1，而 bit12 为 0，写传送优先级高于读传送。若此 bit 为 1，而 bit12 为 1，优先级在写传送与读传送之间交替。
7	读此 bit 为 1 时表示写计数器寄存器为 0 值。
6	读此 bit 为 1 时表示读计数器寄存器为 0 值。
5	读此 bit 为 1 时表示 Add-On to PCI FIFO 空 (EMPTY)。
4	读此 bit 为 1 时表示 Add-On to PCI FIFO 至少有 4 个有效。
3	读此 bit 为 1 时表示 Add-On to PCI FIFO 满 (FULL)。
2	读此 bit 为 1 时表示 PCI to Add-On FIFO 空 (EMPTY)。
1	读此 bit 为 1 时表示 PCI to Add-On FIFO 至少有 4 个有效。
0	读此 bit 为 1 时表示 PCI to Add-On FIFO 满 (FULL)。

六、 ADD-ON 总线操作寄存器组 (ADD-ON BUS OPERATION REGISTERS)

Add-On 总线操作寄存器共有 18 个 DWORD 寄存器，这些寄存器只能通过 Add-On 接口访问。这些寄存器与 PCI 总线操作寄存器非常相似，最重要的区别是在 Add-On 端没有 MASTER CONTROL/STATUS REGISTER (MCSR)，而增加了两个 Pass-Thru 寄存器。如下图 23 所示。

Address	Abbreviation	Register Name
00h	AIMB1	Add-On Incoming Mailbox Register #1
04h	AIMB2	Add-On Incoming Mailbox Register #2
08h	AIMB3	Add-On Incoming Mailbox Register #3
0Ch	AIMB4	Add-On Incoming Mailbox Register #4
10h	AOMB1	Add-On Outgoing Mailbox Register #1
14h	AOMB2	Add-On Outgoing Mailbox Register #2
18h	AOMB3	Add-On Outgoing Mailbox Register #3
1Ch	AOMB4	Add-On Outgoing Mailbox Register #4
20h	AFIFO	Add-On FIFO port
24h	MWAR ¹	Bus Master Write Address Register
28h	APTA	Add-On Pass-Through Address
2Ch	APTD	Add-On Pass-Through Data
30h	MRAR ¹	Bus Master Read Address Register
34h	AMBEF	Add-On Mailbox Empty/Full Status
38h	AINT	Add-On Interrupt control
3Ch	AGCSTS	Add-On General Control and Status Register
58h	MWTC ¹	Bus Master Write Transfer Count
5Ch	MRTC ¹	Bus Master Read Transfer Count

图 23 Add-On 总线操作寄存器

- 1) Add-On Incoming Mailbox 寄存器指向 PCI 操作寄存器中的 Outgoing Mailbox，这些寄存器只读。Add-On Outgoing Mailbox 指向 PCI 操作寄存器中的 Incoming Mailbox，这些寄存器只写。AFIFO 寄存器是双向的，与 PCI 操作寄存器中的 FIFO 对应。
- 2) 写地址寄存器 MWAR，当 Add-On 为总线主设备时 (S5933 配置为 PCI 总线主设备)，Add-On 端访问此寄存器才有意义。用 FIFO 把数据从 Add-On 端传送到 PCI 总线时，此寄存器的内容指向 PC 主机内存的地址，每传送一个 DWORD 数据此寄存器的内容自动加 4，以指向下一个待写内存的地址。此寄存器的低两 bits 总保持 0，所以传送的数据必须以 DWORD 为边界。
- 3) Pass-Thru 地址 (偏移) 寄存器 APTA，当 PC 主机访问基地址 1~4 中的某个地址范围时，此寄存器保存 PCI 交易地址周期中的地址数据。
- 4) Pass-Thru 数据寄存器 APTD，当 PC 主机写基地址 1~4 中的某个地址范围时，此寄存器保存 PCI 交易数据周期中的数据。当 PC 主机读基地址 1~4 中的某个地址范围时，在 PCI 交易数据周期期间，S5933 把此寄存器的内容输出到 PCI 总线。
- 5) 读地址寄存器 MRAR，当 Add-On 为总线主设备时 (S5933 配置为 PCI 总线主设备)，Add-On 端访问此寄存器才有意义。用 FIFO 把数据从 PCI 总线传送到 Add-On 端时，此寄存器的内容指向 PC 主机内存的地址，每传送一个 DWORD 数据此寄存器的内容自动加 4，以指向下一个待读内存的地址。此寄存器的低两 bits 总保持 0，所以传送的数据必须以 DWORD 为边界。
- 6) Add-On 端 Mailbox 状态寄存器 AMBEF，此寄存器只读，每一 bit 的意义如下：

Bit	描述
31: 16	<p>Outgoing Mailbox Status, 指示某个 Outgoing Mailbox 已经被 Add-On 总线端写入, 但没有被 PCI 端取走。一个 bit 代表 Mailbox 的一个 byte, 若某一个 bit 为 1 表示某个 byte 为满状态。</p> <p>Bit31 = Outgoing Mailbox 4 byte 3 Bit30 = Outgoing Mailbox 4 byte 2 Bit29 = Outgoing Mailbox 4 byte 1 Bit28 = Outgoing Mailbox 4 byte 0 Bit27 = Outgoing Mailbox 3 byte 3 Bit26 = Outgoing Mailbox 3 byte 2 Bit25 = Outgoing Mailbox 3 byte 1 Bit24 = Outgoing Mailbox 3 byte 0 Bit23 = Outgoing Mailbox 2 byte 3 Bit22 = Outgoing Mailbox 2 byte 2 Bit21 = Outgoing Mailbox 2 byte 1 Bit20 = Outgoing Mailbox 2 byte 0 Bit19 = Outgoing Mailbox 1 byte 3 Bit18 = Outgoing Mailbox 1 byte 2 Bit17 = Outgoing Mailbox 1 byte 1 Bit16 = Outgoing Mailbox 1 byte 0</p>
15: 0	<p>Incoming Mailbox Status, 指示某个 Incoming Mailbox 已经被 PCI 端写入, 但没有被 Add-On 端取走。一个 bit 代表 Mailbox 的一个 byte, 若某一个 bit 为 1 表示某个 byte 为满状态。</p> <p>Bit15 = Incoming Mailbox 4 byte 3 Bit14 = Incoming Mailbox 4 byte 2 Bit13 = Incoming Mailbox 4 byte 1 Bit12 = Incoming Mailbox 4 byte 0 Bit11 = Incoming Mailbox 3 byte 3 Bit10 = Incoming Mailbox 3 byte 2 Bit09 = Incoming Mailbox 3 byte 1 Bit08 = Incoming Mailbox 3 byte 0 Bit07 = Incoming Mailbox 2 byte 3 Bit06 = Incoming Mailbox 2 byte 2 Bit05 = Incoming Mailbox 2 byte 1 Bit04 = Incoming Mailbox 2 byte 0 Bit03 = Incoming Mailbox 1 byte 3 Bit02 = Incoming Mailbox 1 byte 2 Bit01 = Incoming Mailbox 1 byte 1 Bit0 = Incoming Mailbox 1 byte 0</p>

- 7) Add-On 中断控制/状态寄存器 AINT, 此寄存器的上电值为 00000000h。用此寄存器设置产生 Add-On 接口中断的条件, 以及读此寄存器的有关 bit 判断当前中断产生的源, 和写此寄存器的有关 bit 来清除当前中断。中断源有 6 个, 分别为: 某个 Outgoing Mailbox 变空、某个 Incoming Mailbox 变满、PC 主机发布 Built-in self test、写计数器到达 0、读计数器到达 0、目标/主设备设备异常。此寄存器的结构

如下图 24。

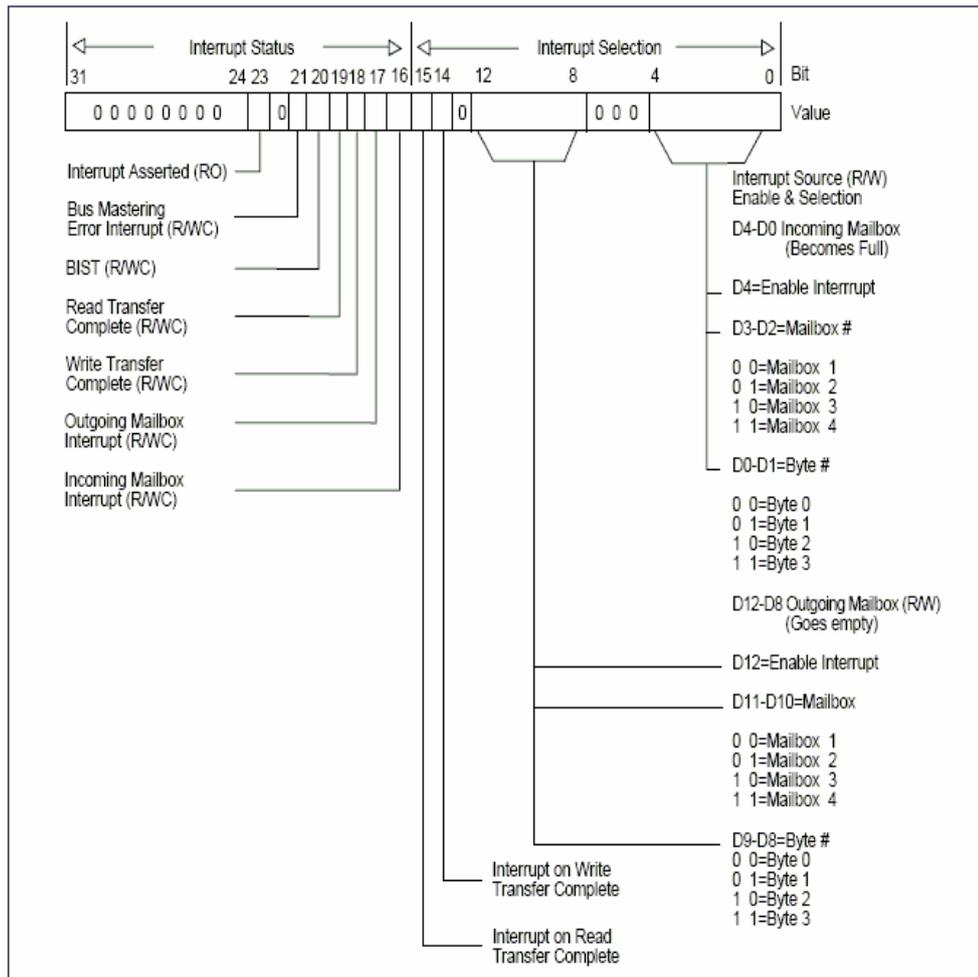


图 24 Add-0n INTERRUPT CONTROL/STATUS REGISTER (AINT)

AINT 每个 bit 的具体意义如下：

Bit	描述
31: 24	保留。
23	有中断产生，这是一个只读状态位。此 bit 为 1 表示有 1 个或多个中断源发生中断，它是此寄存器 bit20, bit17 至 bit16 的逻辑或。
22	保留。
21	目标或主设备异常中断。
20	PC 主机发布 Built-in self test 中断。
19	读传送完成中断。在读传送进行中，若读计数器寄存器 MRTC 到达 0, S5933 就置 1 此 bit 给 PCI 总线产生中断。写 1 到此 bit 将复位此 bit 为 0, 写 0 到此 bit 不影响此 bit 的状态。
18	写传送完成中断。在写传送进行中，若写计数器寄存器 MWTC 到达 0, S5933 就置 1 此 bit 给 PCI 总线产生中断。写 1 到此 bit 将复位此 bit 为 0, 写 0 到此 bit 不影响此 bit 的状态。
17	Outgoing Mailbox 变空中断。当本寄存器的 bit12 到 bit8 所选的 Mailbox 被 PCI 总线读时, S5933 就置 1 此 bit 给 Add-0n 端产生中断。写 1 到此 bit 将复位此 bit 为 0, 写 0 到此 bit 不影响此 bit 的状态。
16	Incoming Mailbox 变满中断。当本寄存器的 bit4 到 bit0 所选的

	Mailbox 被 PCI 总线写时, S5933 就置 1 此 bit 给 Add-On 端产生中断。写 1 到此 bit 将复位此 bit 为 0, 写 0 到此 bit 不影响此 bit 的状态。
15	给此 bit 写 1 使能读传送完成时 (读传送计数器到达 0) 产生中断。
14	给此 bit 写 1 使能写传送完成时 (写传送计数器到达 0) 产生中断。
13	保留。
12	给此 bit 写 1 使能 Outgoing Mailbox 变空时产生中断。
11: 10	Outgoing Mailbox 中断选择, 此 2bits 的值选择那个 Outgoing Mailbox 是中断源。[00]b 选择 Outgoing Mailbox 1, [01]b 选择 Outgoing Mailbox 2, [10]b 选择 Outgoing Mailbox 3, [11]b 选择 Outgoing Mailbox 4。
9: 8	Outgoing Mailbox byte 中断选择, 此 2bits 的值选择由 bit11 到 10 指定的 Outgoing Mailbox 中那个 byte 为中断源。[00]b 选择 byte 0, [01]b 选择 byte 1, [10]b 选择 byte 2, [11]b 选择 byte 3。
7: 5	保留。
4	给此 bit 写 1 使能 Incoming Mailbox 变满时产生中断。
3: 2	Incoming Mailbox 中断选择, 此 2bits 的值选择那个 Incoming Mailbox 是中断源。[00]b 选择 Incoming Mailbox 1, [01]b 选择 Incoming Mailbox 2, [10]b 选择 Incoming Mailbox 3, [11]b 选择 Incoming Mailbox 4。
1: 0	Incoming Mailbox byte 中断选择, 此 2bits 的值选择由 bit3 到 2 指定的 Incoming Mailbox 中那个 byte 为中断源。[00]b 选择 byte 0, [01]b 选择 byte 1, [10]b 选择 byte 2, [11]b 选择 byte 3。

8) Add-On 一般控制/状态寄存器 AGCSTS, 此寄存器的上电值为 000000F4h。此寄存器的结构如下图 25。

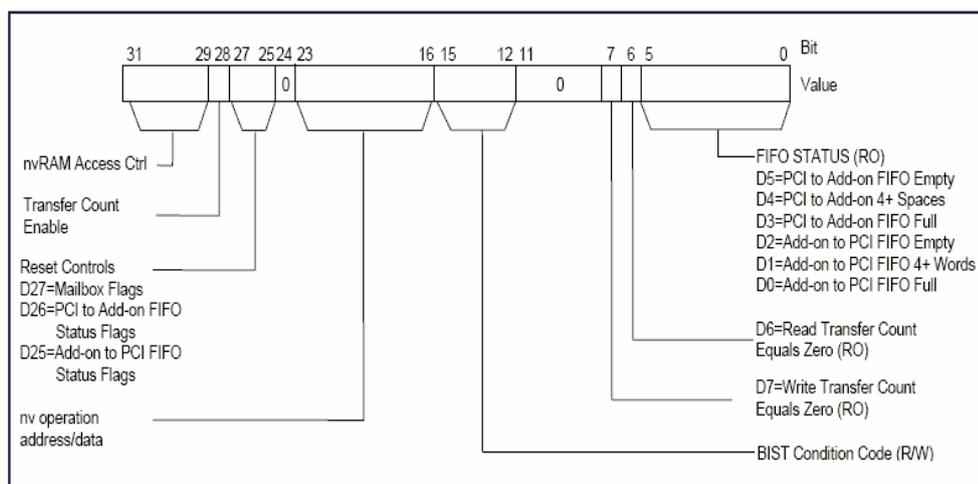


图 25 Add-On general CONTROL/STATUS REGISTER (AGCSTS)

AGCSTS 每个 bit 的具体意义如下:

Bit	描述
31: 29	nvRAM 访问控制信号。用这几 bits 和 bit23 至 bit16 可完成扩展 nvRAM 的访问。访问时, 先写 low-order 地址, 再写 high-order 地址, 最后访问数据 data 字节。Bit31 即用作使能, 又用作准备好信号, 若要启动一个访问, 必须给 bit31 写入 1, 而下一个访

	问必须等到 bit31 变为 0 时才可继续。																																								
	<table border="1"> <thead> <tr> <th>D31</th> <th>D30</th> <th>D29</th> <th>W/R</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>X</td> <td>X</td> <td>W</td> <td>Inactive</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>W</td> <td>Load low address byte</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>W</td> <td>Load high address byte</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>W</td> <td>Begin write</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>W</td> <td>Begin read</td> </tr> <tr> <td>0</td> <td>X</td> <td>X</td> <td>R</td> <td>Ready</td> </tr> <tr> <td>1</td> <td>X</td> <td>X</td> <td>R</td> <td>Busy</td> </tr> </tbody> </table> <p>注意：虽然 PCI 总线端和 Add-On 端都可访问 nvRAM，但必须保证不同时访问。</p>	D31	D30	D29	W/R		0	X	X	W	Inactive	1	0	0	W	Load low address byte	1	0	1	W	Load high address byte	1	1	0	W	Begin write	1	1	1	W	Begin read	0	X	X	R	Ready	1	X	X	R	Busy
D31	D30	D29	W/R																																						
0	X	X	W	Inactive																																					
1	0	0	W	Load low address byte																																					
1	0	1	W	Load high address byte																																					
1	1	0	W	Begin write																																					
1	1	1	W	Begin read																																					
0	X	X	R	Ready																																					
1	X	X	R	Busy																																					
28	传送计数器使能。若此 bit 置 1，传送计数器将用于 Add-On 端发起的总线主传送，否则忽略传送计数器。																																								
27	Mailbox flag Reset。给此 bit 写入 1 复位所有 Mailbox 的状态标志，即所有 Mailbox 的状态将变为空 (EMPTY)。给此 bit 写入 1 后没有必要再写入 0 复位此 bit，读此 bit 永远返回 0。																																								
26	PCI to Add-On FIFO Status Reset。给此 bit 写入 1 把 PCI to Add-On FIFO 0 的状态复位到空 (EMPTY)。给此 bit 写入 1 后没有必要再写入 0 复位此 bit，读此 bit 永远返回 0。																																								
25	Add-On to PCI FIFO Status Reset。给此 bit 写入 1 把 Add-On to PCI FIFO 的状态复位到空 (EMPTY)。给此 bit 写入 1 后没有必要再写入 0 复位此 bit，读此 bit 永远返回 0。																																								
24	保留。																																								
32: 16	扩展 nvRAM 的 address/data 接口。																																								
15: 12	BIST condition code。Bit15 到 bit12 分别代表 BIST 寄存器的 bit3 到 bit0。																																								
11: 8	保留。																																								
7	读此 bit 为 1 时表示读计数器寄存器为 0 值。																																								
6	读此 bit 为 1 时表示写计数器寄存器为 0 值。																																								
5	读此 bit 为 1 时表示 PCI to Add-On FIFO 空 (EMPTY)。																																								
4	读此 bit 为 1 时表示 PCI to Add-On FIFO 至少有 4 个有效。																																								
3	读此 bit 为 1 时表示 PCI to Add-On FIFO 满 (FULL)。																																								
2	读此 bit 为 1 时表示 Add-On to PCI FIFO 空 (EMPTY)。																																								
1	读此 bit 为 1 时表示 Add-On to PCI FIFO 至少有 4 个有效。																																								
0	读此 bit 为 1 时表示 Add-On to PCI FIFO 满 (FULL)。																																								

- 9) Add-On 端写计数器寄存器 MWTC，当 Add-On 为总线主设备时 (S5933 配置为 PCI 总线主设备)，Add-On 端访问此寄存器才有意义。每传送一个 DWORD 数据此寄存器的内容自动减 4，当此寄存器的 bit3~bit25 到达 0 时，S5933 可能给 PCI 总线或 Add-On 端产生一个中断。当此寄存器的 bit0~bit25 可以读写，bit26~bit31 被 S5933 硬件保持为 0。若 Add-On 一般控制/状态寄存器 AGCSTS 的 bit28 被写入 0，此寄存器被忽略。
- 10) Add-On 端读计数器寄存器 MRTC，当 Add-On 为总线主设备时 (S5933 配置为 PCI 总线主设备)，Add-On 端访问此寄存器才有意义。每传送一个 DWORD 数据此寄存器的内容自动减 4，当此寄存器的 bit3~bit25 到达 0 时，S5933 可能给 PCI 总线

或 Add-On 端产生一个中断。当此寄存器的 bit0~bit25 可以读写, bit26~bit31 被 S5933 硬件保持为 0。若 Add-On 一般控制/状态寄存器 AGCSTS 的 bit28 被写入 0, 此寄存器被忽略。

七、 ADD-ON 总线的三种工作方式

S5933 具有 Pass-Thru、Mailbox、FIFO 三种工作方式。一般情况下,Pass-Thru 和 Mailbox 用于计算机和 S5933 的后端逻辑 (如 FPGA) 之间的参数或命令传送, FIFO 用于高速批量数据传送。

(一) Pass-Thru

只有 S5933 为 PCI 从设备 (PCI target) 时, Pass-Thru 才能使用, 也就是说, PC 主机永远是 Pass-Thru 通信的发起方。要使用 Pass-Thru 通信, 必须给 S5933 外接 nvRAM 配置芯片, 以便给 Pass-Thru 在 PC 主机上分配空间, S5933 配置空间中 BASE ADDRESS REGISTER #1~5 分别对应 4 个 Pass-Thru 通道基地址。S5933 可以实现四条 Pass-Thru 通道, 可以实现单周期和突发 (burst) 数据传送。下面就 Pass-Thru 的工作时序做一些介绍。

单周期 Pass-Thru 写时序

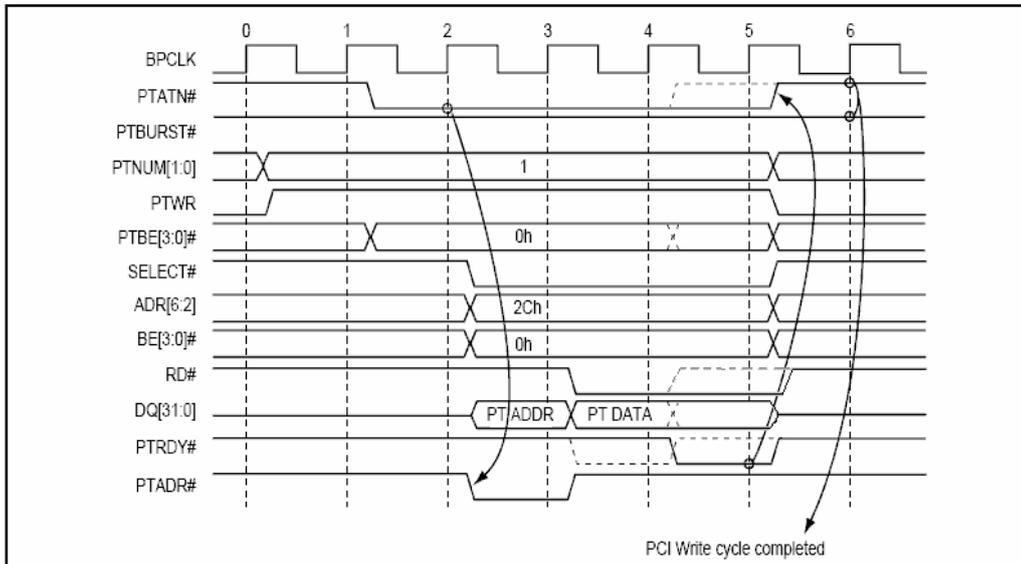


图 26 Pass-Thru 单周期写时序

- Clock0 : PCI 总线写 Pass-Thru 通道 1, 地址偏移写入 Pass-Thru 地址寄存器。
- Clock1 : PCI 总线数据写入 Pass-Thru 数据寄存器, S5933 把 PTATN#置低电平, 一个 Pass-Thru 写过程开始。
- Clock2 : 在 Clock2 上升沿, S5933 的后端逻辑 (FPGA) 检测到 PTATN#为低时, 立即输出以下信号到 S5933 予以响应, 置低 SELECT#, PTBE[3:0]#= 0000b (32bits 数据宽度), 置低 PTADR# (以便读出地址偏移), ADR[6:2]=2Ch(Pass-Thru 数据寄存器的地址)。同时, 对 S5933 输出的其它信号进行译码, 在本例中, PTNUM[1:0]=01b 指示 Pass-Thru 通道 1, PTWR=1 指示 Pass-Thru 写, PTBURST#=1 指示非突发传输(是单周期传输)。
- Clock3 : S5933 的后端逻辑 (FPGA) 采样数据总线 DQ[31:0], 即为地址偏移。同时, 置高 PTADR# (读出地址偏移结束), 置低 RD# (读出数据), 若后端逻辑 (FPGA) 的地址译码速度足够快, 置低 PTRDY#, 本次 Pass-Thru 访问将会在 Clock4 结束。
- Clock4 : S5933 的后端逻辑 (FPGA) 若在 Clock3 时置低 PTRDY#, Clock4 的上升沿就应该采样数据总线 DQ[31:0], 即为 Pass-Thru 本次写入数据, 同时, 后端逻辑 (FPGA) 无效相关信号, 本次 Pass-Thru 的访问结束。若后端逻辑 (FPGA) 的地址译码速度比较慢, 可以在此时置低 PTRDY#, 本次

Pass-Thru 访问将会在 Clock5 结束。

Clock5 : PTATN#和 PTBURST#在 Clock5 的上升沿由 S5933 置高电平, 本次 Pass-Thru 的访问结束, 后端逻辑 (FPGA) 无效相关信号。

单周期 Pass-Thru 读时序

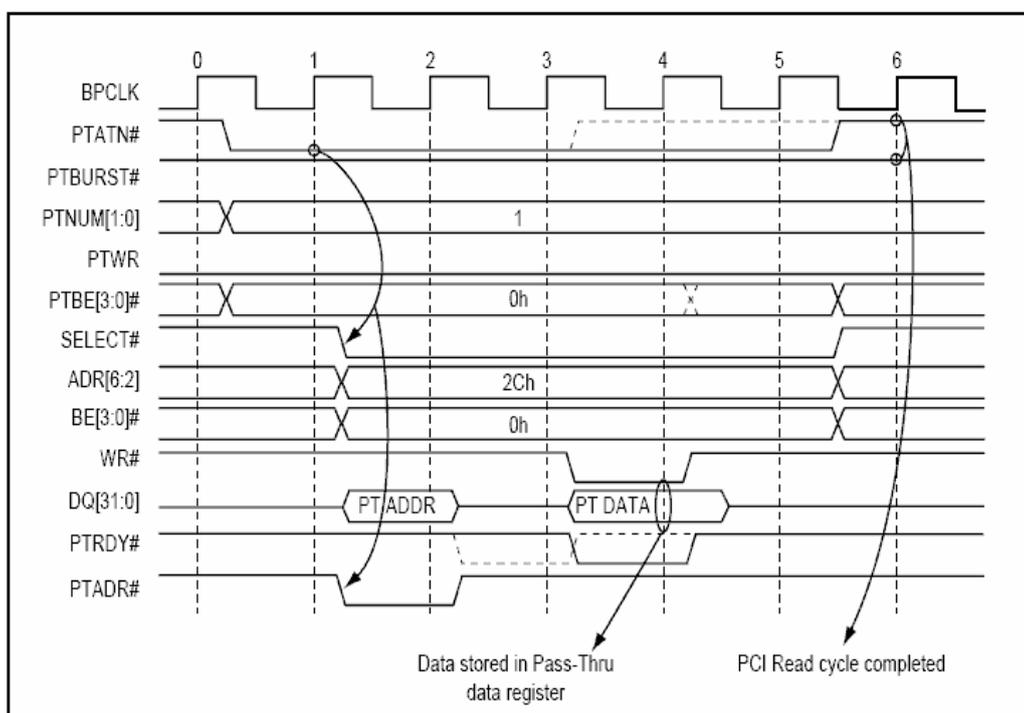


图 27 单周期 Pass-Thru 读时序

Clock0 : PCI 总线把地址偏移写入 Pass-Thru 地址寄存器, S5933 把 PTATN#置低电平, PTWR 置低, 一个 Pass-Thru 读过程开始。

Clock1 : 在 Clock1 上升沿, S5933 的后端逻辑 (FPGA) 检测到 PTATN#为低时, 立即输出以下信号到 S5933 予以响应, 置低 SELECT#, PTBE[3:0]#= 0000b (32bits 数据宽度), 置低 PTADR# (以便读出地址偏移), ADR[6:2]=2Ch(Pass-Thru 数据寄存器的地址)。同时, 对 S5933 输出的其它信号进行译码, 在本例中, PTNUM[1:0]=01b 指示 Pass-Thru 通道 1, PTWR=0 指示 Pass-Thru 读, PTBURST#=1 指示非突发传输 (是单周期传输)。

Clock2 : S5933 的后端逻辑 (FPGA) 采样数据总线 DQ[31:0], 即为地址偏移。同时, 置高 PTADR# (读出地址偏移结束)。后端逻辑 (FPGA) 可以在 Clock2 内对地址进行译码。数据总线 DQ[31:0]方向: S5933 to 后端逻辑。

Clock3 : S5933 的后端逻辑 (FPGA) 在 Clock3 的上升沿置低 PTRDY#和 WR#, 同时, 输出数据到数据总线 DQ[31:0]。数据总线 DQ[31:0]方向: 后端逻辑 to S5933。

Clock4 : 后端逻辑 (FPGA) 无效相关信号。若后端逻辑 (FPGA) 输出信号 WR#或数据 DQ[31:0]等速度比较慢, 可以不在 Clock3 置低 PTRDY#, 而在 Clock4 置低 PTRDY#, 以插入一个等待周期。

Clock5 : PTATN#和 PTBURST#在 Clock5 的上升沿由 S5933 置高电平, 本次 Pass-Thru 的访问结束。

Pass-Thru Burst 写时序

S5933 在置低 PTATN#的同时置低 PTBURST#, 表明突发传送。PTBURST#在还剩最后

一个数据时，由 S5933 置高变为无效，即 PTATN#为低电平，而 PTBURST#为高电平时，表明将要传输最后一个数据。注意在下图 6 中，在 Clock8 的上升沿，S5933 把 PTATN# 输出为无效电平，这是由 PCI 总线插入的一个等待。

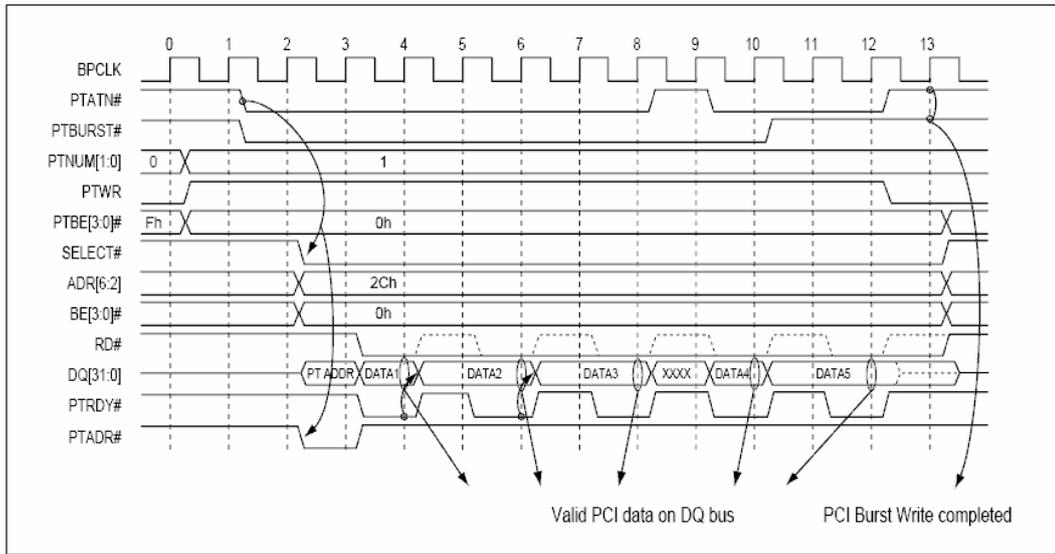


图 28 Pass-Thru burst write 时序

Pass-Thru Burst 读时序

S5933 在置低 PTATN#的同时置低 PTBURST#，表明突发传送。PTBURST#在还剩最后一个数据时，由 S5933 置高变为无效，即 PTATN#为低电平，而 PTBURST#为高电平时，表明将要传输最后一个数据。注意在下图 7 中，对 S5933 的后端逻辑 (FPGA)，最好不要在 Clock2 的上升沿，就有效 WR#、PTRDY#，输出数据到数据总线 DQ[31:0]上，应该等待一个时钟周期，在 Clock3 的上升沿再输出这些信号，以便给 S5933 让出数据总线 DQ[31:0]留一定时间，否则数据总线 DQ[31:0]上可能产生冲突。

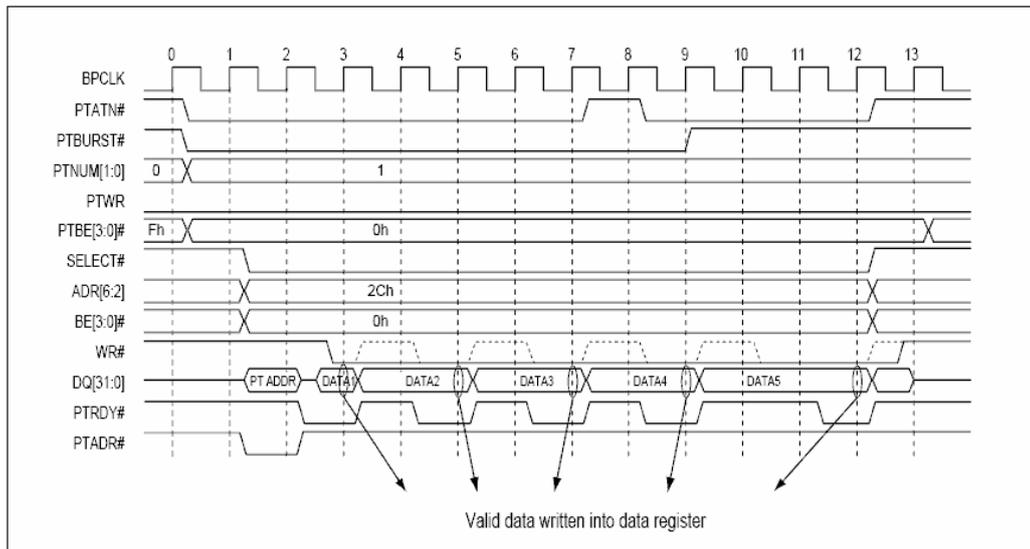


图 29 Pass-Thru burst read 时序

(二) FIFO 工作方式

S5933 中有两个独立的 FIFO (8×32bits)，PCI to Add-On FIFO 和 Add-On to PCI FIFO，S5933 根据读写方向自动选择使用那个 FIFO。

● FIFO ENDIAN

为了适应各种系统的要求，数据在通过 FIFO 时，S5933 可以改变数据的排列顺序，中断配置/状态寄存器(INTCSR)的 bit31 到 bit24 用于完成这些设置。INTCSR 的 bit31 到 bit24 结构如下图 30。

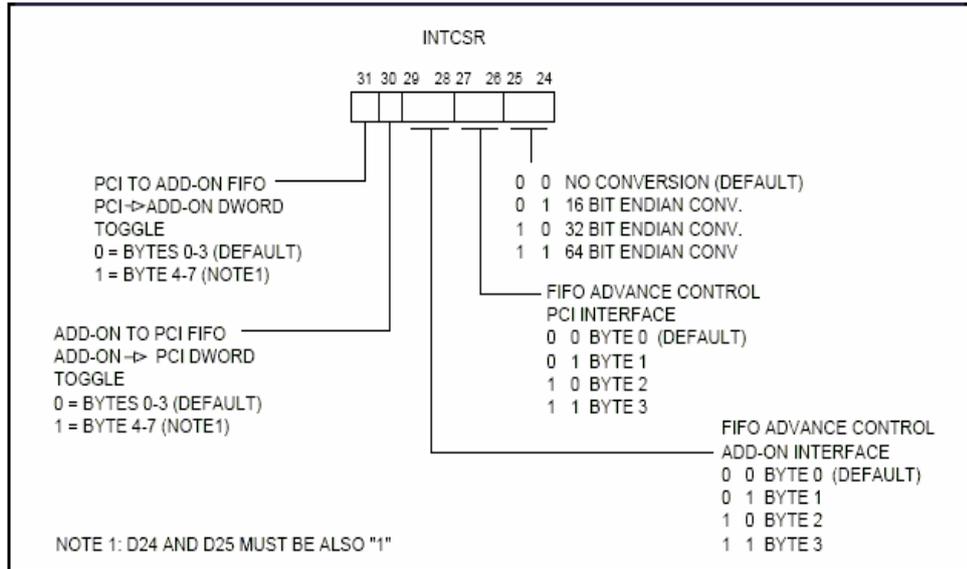


图 30 INTCSR FIFO Advance and Endian Control Bits

INTCSR 的 bit29 到 bit26 设置 FIFO 的状态改变由 FIFO 的那个 byte 决定。例如把 bit29 到 bit28 设置为 10，表示只有 Add-On 端改写 FIFO 的 byte2 时，Add-On to PCI FIFO 的状态才改变，这种设置在 Add-On 端数据宽度为 16bits 非常有用，Add-On 端可以先写 FIFO 的低 16bits (BE[3:0]# = 1100)，再写 FIFO 的高 16bits (BE[3:0]# = 0011)，就可把 Add-On 端 16bits 数据转换成 PCI 总线端的 32bits 数据。

INTCSR 的 bit31 到 bit30 和 bit25 到 bit24 设置 FIFO 的 ENDIAN 方式，ENDIAN 方式总共有 16BIT、32BIT 和 64BIT 三种。其中 64BIT ENDIAN 用于把 64bits 数据转换为 32bits 数据，以便在 32bits 的 PCI 总线上传输。只有在使用 64BIT ENDIAN 时，INTCSR 的 bit31 到 bit30 才有意义，bit31 和 bit30 的初始状态决定第一次传输的数据是低 32bits 还是高 32bits。三种 FIFO ENDIAN 方式分别如下图 31、32、33 所示。

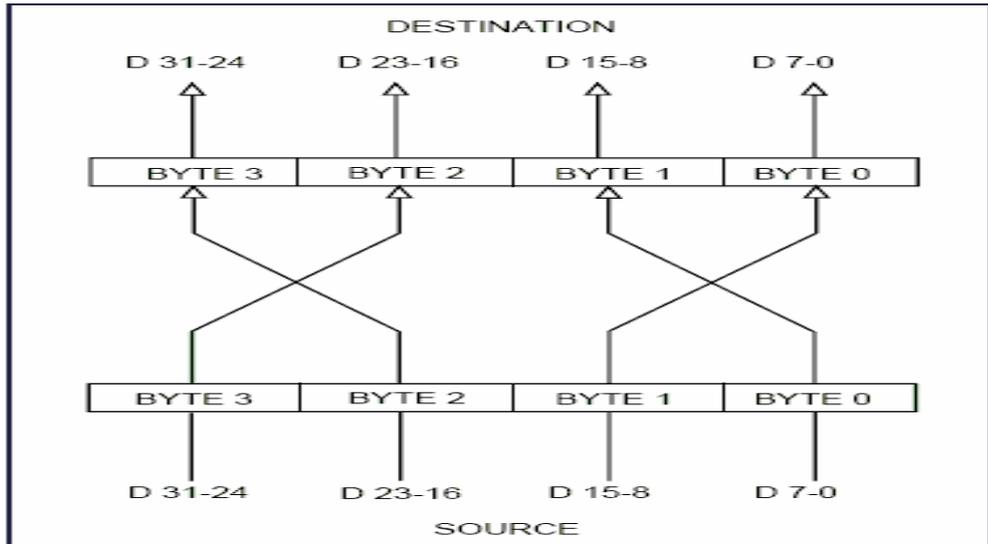


图 31 16 BIT ENDIAN

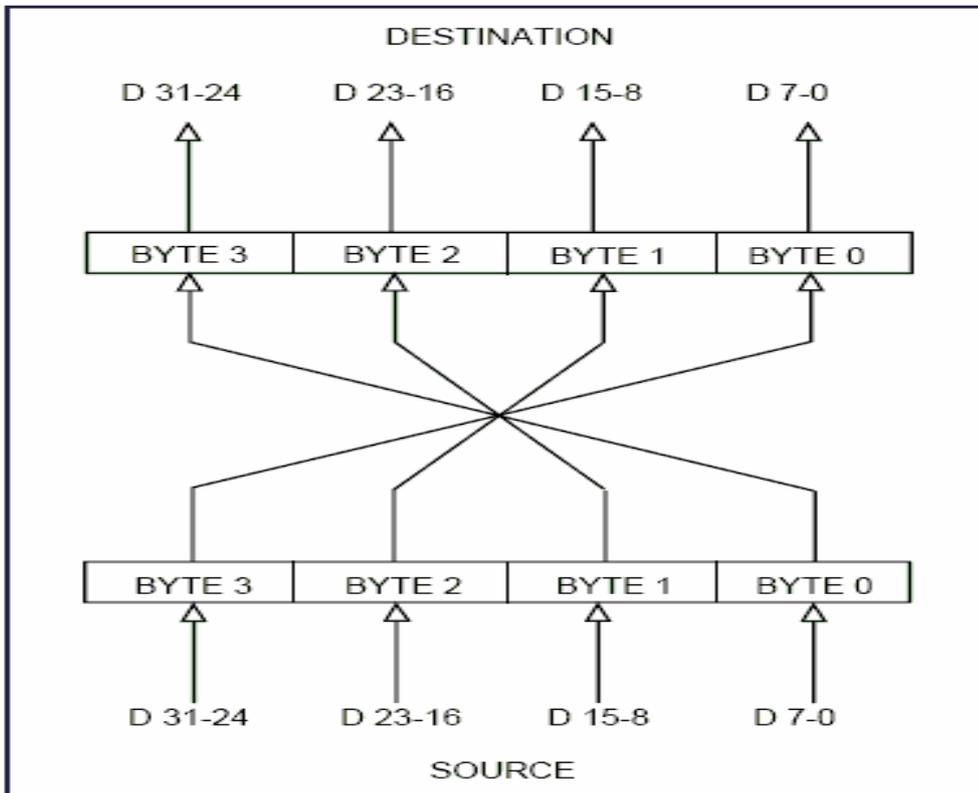


图 32 31 BIT ENDIAN

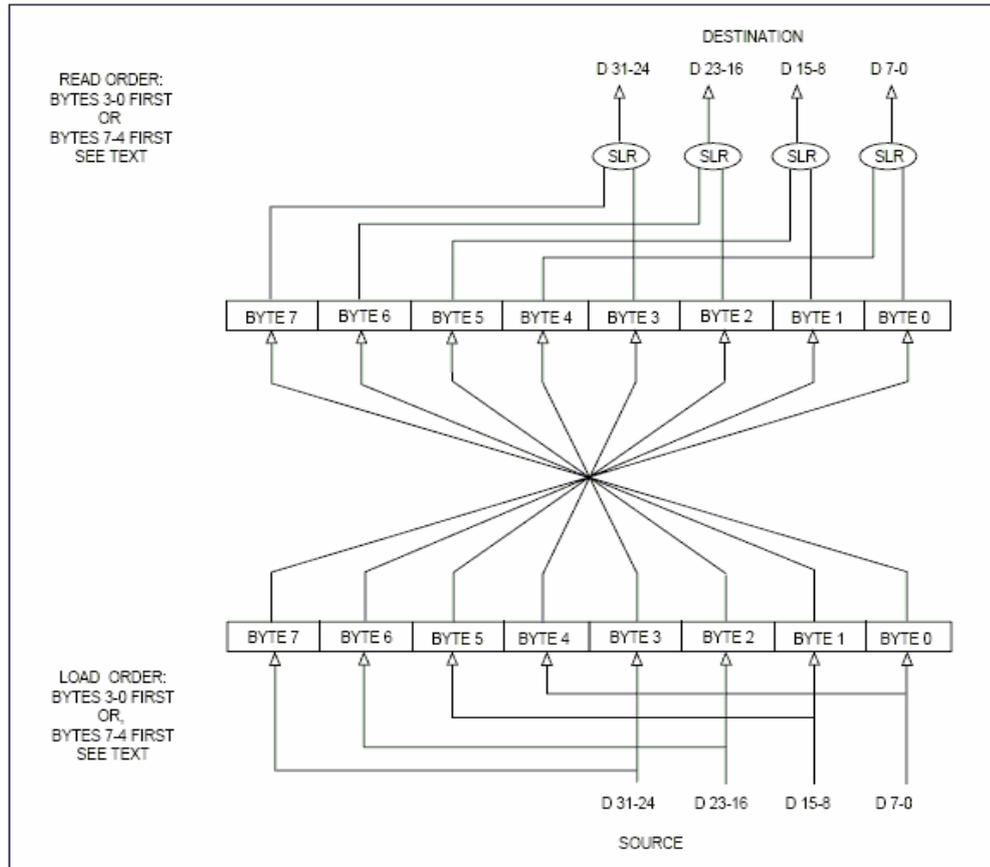


图 33 64 BIT ENDIAN

● S5933 配置为从设备时的 FIFO 数据传输

当把 S5933 配置为从设备时 (nvRAM 45h 的 bit7=1)，读地址 (MRAR)、写地址 (MWAR)、读计数 (MRTC)、写计数器 (MWTC) 寄存器只能在 PCI 总线端访问。在 PC 主机作为主控 FIFO (DMA) 传送时，PC 主机需要完成相应的设置方可启动 FIFO (DMA) 传送。在 PC 主机启动 FIFO (DMA) 传送后，当计数器寄存器 (MRTC 或 MWTC) 的值到达 0 时，S5933 可以给 PC 主机产生一个中断，表明此次 FIFO (DMA) 传送结束。PC 主机设置 S5933 的步骤描述如下。

- 1) 确定 FIFO 的 ENDIAN 方式。确定 INTCSR 寄存器的 bit31 到 bit24。
- 2) 使能中断。如果想在数据传输完成时 (计数器寄存器 (MRTC 或 MWTC) 的值到达 0)，让 S5933 给 PC 主机产生中断就置 1 INTCSR 寄存器的 bit15 或 bit14。

INTCSR bit15 使能读传输完成中断

INTCSR bit14 使能写传输完成中断

- 3) 复位 FIFO 状态。

MCSR bit26 Reset Add-On to PCI FIFO flags.

MCSR bit25 Reset PCI to Add-On FIFO flags.

- 4) 确定 FIFO 管理方案。以便决定 S5933 在什么条件下才去申请 PCI 总线控制权。

MCSR bit13 PCI to Add-On FIFO management scheme

MCSR bit09 Add-On to PCI FIFO management scheme

- 5) 确定读、写 FIFO 优先级。

- MCSR bit12 Read vs. write priority
MCSR bit08 Write vs. read priority
- 6) 确定传输源/目标地址。这些地址是 PC 主机 DMA 时, 申请到的主机内存的首地址, S5933 要求这些地址以 DWORD 为边界。
MWAR all 写传输时主机内存的首地址
MRAR all 读传输时主机内存的首地址
- 7) 确定传输数据的 byte 总量。虽然 S5933 不要求给 MWTC 或 MRTC 写入的数据被 4bytes 整除, 但不被 4bytes 整除所剩余的几个数据不能被 S5933 传输, 所以传输的数据要想办法以 DWORD 为边界。
MWTC all 写传输时数据的 byte 数量
MRTC all 读传输时数据的 byte 数量
- 8) 启动 FIFO (DMA) 传输。
MCSR bit14 启动读传输 (PCI to Add-On FIFO)
MCSR bit10 启动写传输 (Add-On to PCI FIFO)

如果在步骤 2) 使能了中断或者使能了 Mailbox 中断, PC 主机驱动程序必须提供中断响应程序。当有中断发生时, 中断响应程序首先读 INTCSR 寄存器, 以确定中断源 (Mailbox 中断、目标设备异常中断、主设备异常中断、读传输完成中断或写传输完成中断), 然后响应相应的中断, 最后给 INTCSR 寄存器的相应 bit 写入 1 清除相应的中断。

● S5933 配置为主设备时的 FIFO 数据传输

当把 S5933 配置为主设备时 (nvRAM 45h 的 bit7=0), S5933 扩展 6 个 IO 信号, 以方便 FIFO 传送。这几个信号是:

信号	复用信号	信号类型	说明
FRF	E_ADDR(15)	0	PCI to Add-On FIFO is full.
FWE	E_ADDR(14)	0	Add-On to PCI FIFO is empty.
AMWEN	EQ(7)	I	置高, Add-On 主控 FIFO 写使能, S5933 开始申请 PCI 总线的控制权, 若得到控制权就进行数据传送, 直到 AMWEN 为低才放弃 PCI 总线的控制权。
AMREN	EQ(6)	I	置高, Add-On 主控 FIFO 读使能, S5933 开始申请 PCI 总线的控制权, 若得到控制权就进行数据传送, 直到 AMREN 为低才放弃 PCI 总线的控制权。
FRC#	EQ(5)	I	置低, 复位 PCI to Add-On FIFO 指针和状态。
FWC#	EQ(4)	I	置低, 复位 Add-On to PCI FIFO 指针和状态。

在 S5933 作为主设备时, 读地址 (MRAR)、写地址 (MWAR)、读计数 (MRTC)、写计数器 (MWTC) 寄存器只能在 Add-On 端访问, 这些寄存器的值可以通过 Mailbox 或 Pass-Thru 传给 S5933 的后端逻辑 (如 FPGA), 然后再由 Add-On 端写入相应寄存器。AMWEN 和 AMREN 的优先级最高, 高于控制/状态寄存器 (MCSR) 的设置以及读/写计数寄存器的状态, 即不管 MCSR 如何设置, 也不管读/写计数寄存器是否为

0, 都可以通过 AMWEN 和 AMREN 启停 FIFO 传送。

在 Add-0n 端逻辑启动 FIFO 传输开始前, 必须进行必要的寄存器设置。Add-0n 端发起 FIFO (DMA) 传输时, 设置 S5933 的步骤描述如下。

- 1) 确定 FIFO 的 ENDIAN 方式。确定 INTCSR 寄存器的 bit31 到 bit24。(注: INTCSR 由 PC 主机通过 PCI 总线设置)
- 2) 确定使用或不使用传输计数器寄存器 (MWTC 和 MRTC)。
AGCSTS bit28 Enable/disable 读写传输计数器寄存器
- 3) 确定使用或不使用中断。若在步骤 1) 使能传输计数器寄存器, 当传输计数器寄存器 (MWTC 或 MRTC) 计数到达 0 时, S5933 可以给 Add-0n 端产生一个中断。
AINT bit15 Enable/disable MRTC 计数到达 0 时中断
AINT bit14 Enable/disable MWTC 计数到达 0 时中断
- 4) 复位 FIFO 状态。
AGCSTS bit25 Reset Add-0n to PCI FIFO flags。
AGCSTS bit26 Reset PCI to Add-0n FIFO flags。
- 5) 确定 FIFO 管理方案。以便决定 S5933 在什么条件下才去申请 PCI 总线控制权。(注: MCSR 由 PC 主机通过 PCI 总线设置)
MCSR bit13 PCI to Add-0n FIFO management scheme
MCSR bit09 Add-0n to PCI FIFO management scheme
- 6) 确定读、写 FIFO 优先级。
MCSR bit12 Read vs. write priority
MCSR bit08 Write vs. read priority
- 7) 确定传输源/目标地址。这些地址是 PC 主机申请到的主机内存的首地址, S5933 要求这些地址以 DWORD 为边界。
MWAR all 写传输时主机内存的首地址
MRAR all 读传输时主机内存的首地址
- 8) 确定传输数据的 byte 总量。虽然 S5933 不要求给 MWTC 或 MRTC 写入的数据被 4 (bytes) 整除, 但不被 4 (bytes) 整除所剩余的几个数据不能被 S5933 传输, 所以传输的数据要想办法以 DWORD 为边界。
MWTC all 写传输时数据的 byte 数量
MRTC all 读传输时数据的 byte 数量
- 9) 启动 FIFO (DMA) 传输。
用 Add-0n 端信号 AMREN 或 AMWEN 控制 FIFO (DMA) 传输启停。

如果在步骤 3) 使能了中断或者使能了 Mailbox 中断, Add-0n 端逻辑必须提供中断响应逻辑。当有中断发生时, 中断响应逻辑首先读 AINT 寄存器, 以确定中断源 (Mailbox 中断、目标/主设备设备异常中断、读传输完成中断或写传输完成中断), 然后处理相应的中断, 最后给 AINT 寄存器的相应 bit 写入 1 清除相应的中断。

[注: 把 S5933 配置为主设备的情况, 我没有真正实践过, 所以写的不一定正确, 仅供参考。](#)

● FIFO 操作的时序

FIFO 操作在 Add-0n 端的时序不是很复杂, 简单介绍如下:

异步 FIFO 访问 (读, 用 RDFIFO#)

Asynchronous FIFO Register RDFIFO# Access Example

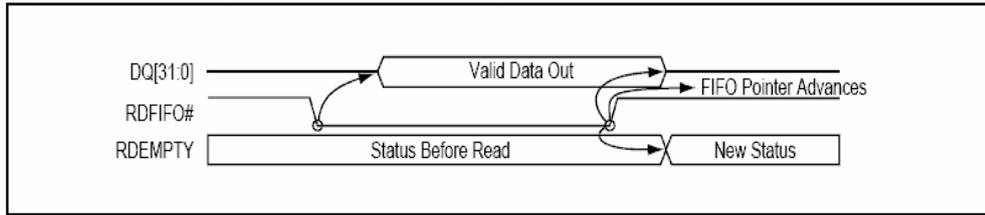


图 34 异步 FIFO 读

要让 FIFO 访问工作在异步方式，需要把 nvRAM 45h 的 bit6、bit5 置 1。从图 8 可以看出，异步 FIFO 访问时，把 RDFIFO#（或 WRFIFO#）置低电平选中 FIFO，在 RDFIFO#（或 WRFIFO#）的上升沿 FIFO 指针加 1，并且锁存数据、改变 FIFO 空满状态。异步 FIFO 访问对后端逻辑速度要求低，缺点是不能突发传输，传输速度低。

同步 FIFO 访问（读，用 RDFIFO#）

Synchronous FIFO Register Burst RDFIFO# Access Example

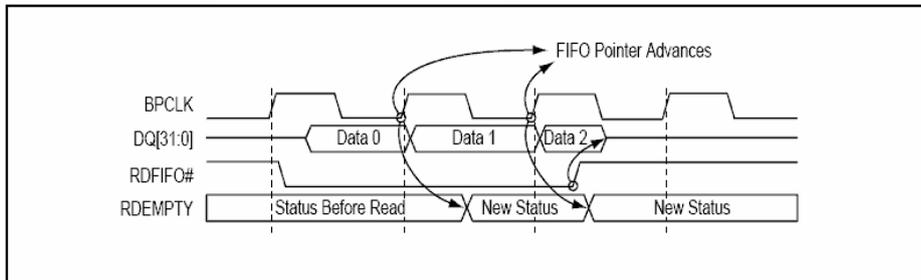


图 35 同步 FIFO 读（用 RDFIFO#信号）

要让 FIFO 访问工作在同步方式，需要把 nvRAM 45h 的 bit6、bit5 置 0。从图 35 可以看出，RDFIFO#(WRFIFO#)在 FIFO 访问期间一直保持有效低电平，在 PCI 时钟 BPClk 的上升沿，FIFO 指针加 1、锁存数据、改变 FIFO 空满状态。

由于 FIFO 对外其实就是一个寄存器，也可以用访问寄存器的方法访问 FIFO，如下图 36 所示。

Synchronous FIFO Register Burst Read Access Example

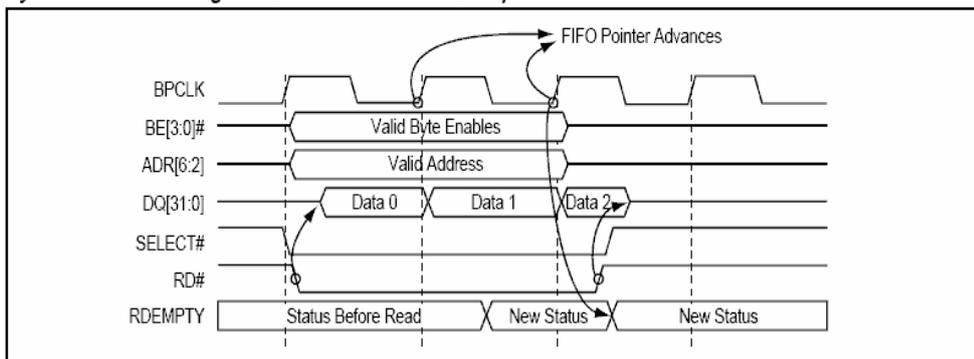


图 36 同步 FIFO 读（用 RD#信号）

(三) Mailbox 方式

S5933 有输入、输出 Mailbox 寄存器各四个，可以采用中断或查询方式访问。中断/状态寄存器（INTCSR）用以实现中断的使能配置、查询和清除，Mailbox 状态寄存器（MBEF）用于查询 Mailbox 状态，通过控制/状态寄存器（MCSR）可以复位 Mailbox 的状态。

因为对某个确定的 Mailbox 寄存器就是一个普通的寄存器，所以在访问时序上不存在同步访问的问题，用异步访问时序即可，访问时序同其它寄存器一样，Add-On 端访问寄存器的时序如下图 37、38。

Asynchronous Add-On Operation Register Read

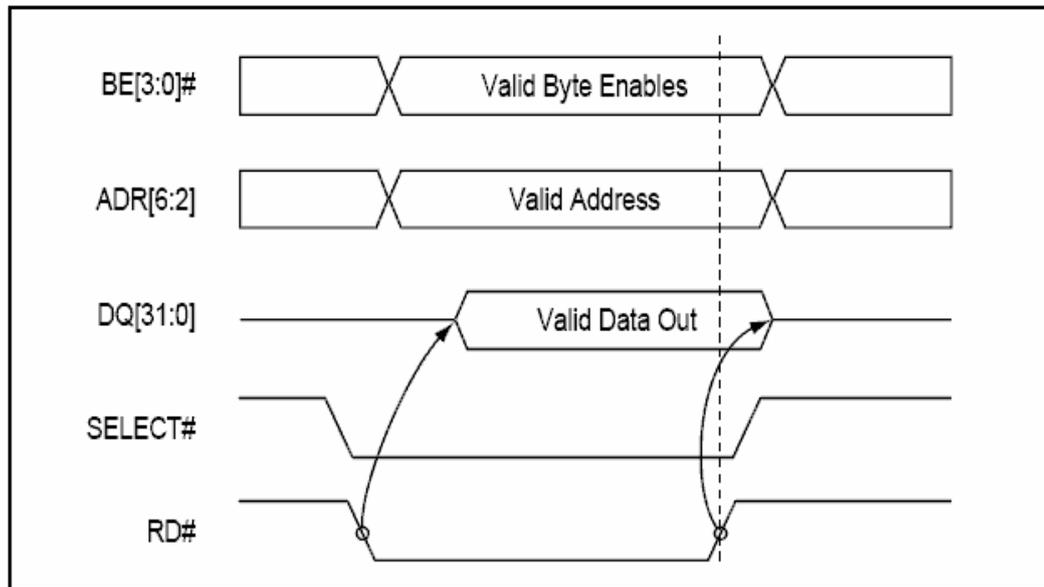


图 37 Add-On 端寄存器（包括 Mailbox）读时序

Asynchronous Add-On Operation Register Write

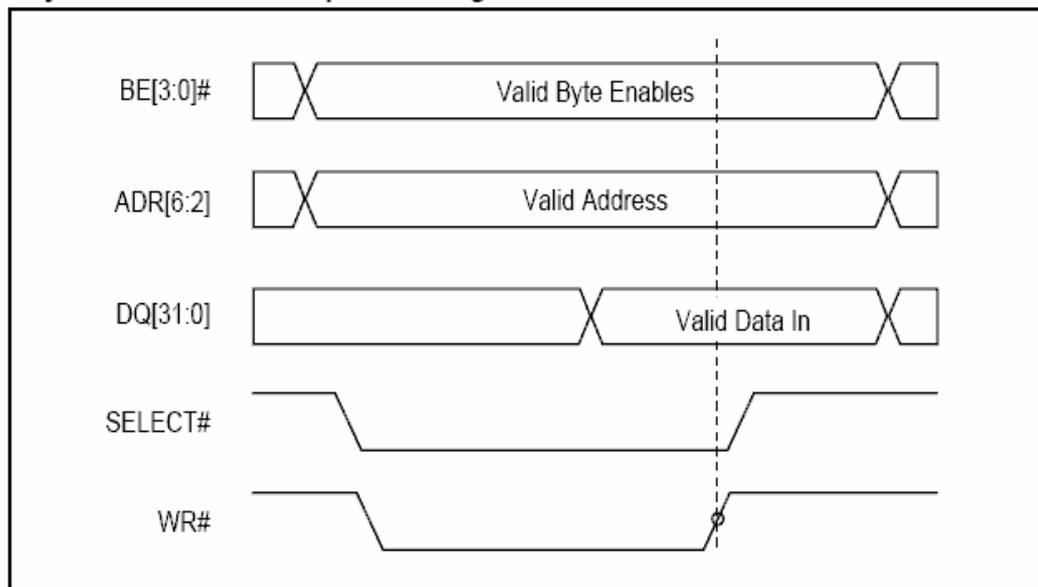


图 38 Add-On 端寄存器（包括 Mailbox）写时序

八、重要信号的时间参数

➤ 异步 RDFIFO#信号时序

Functional Operation Range ($V_{CC}=5.0V$ 5%, $0^{\circ}C$ to $70^{\circ}C$ T_a , 50 pf load on outputs).

Symbol	Parameter	Min	Max	Units	Notes
t_{135}	RDFIFO# High Time	17		ns	
t_{136}	RDFIFO# Low Time	17		ns	
t_{137}	RDFIFO# Low to DQ[31:0] Driven		21	ns	
t_{139}	RDFIFO# High to DQ[31:0] Float		20	ns	
t_{140}	DQ[31:0] Hold from RDFIFO# Rising Edge	5		ns	
t_{181}	PCI to ADD-ON FIFO RDEEMPTY Valid from RDFIFO# Rising Edge		15	ns	
t_{182}	PCI to ADD-ON FIFO FRF Valid from RDFIFO# Rising Edge		85	ns	

Asynchronous RDFIFO# Timing

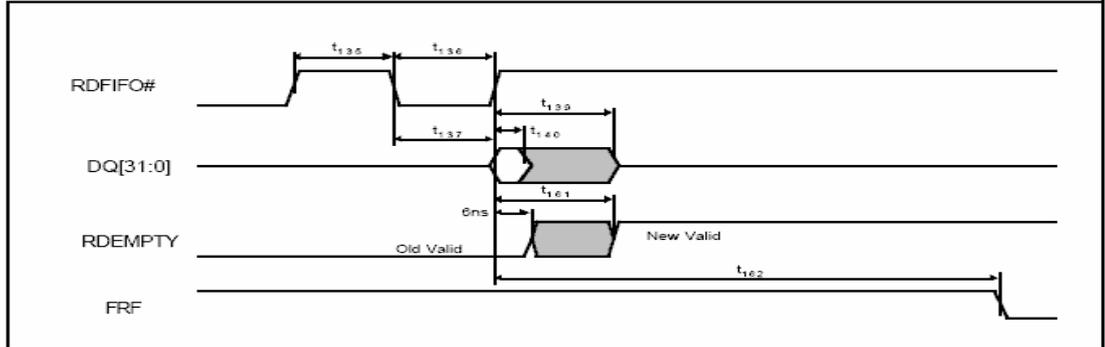


图 39 Asynchronous RDFIFO# Timing

从图 39 看出，RDFIFO# High Time 最小为 17ns，即两次读（RDFIFO#）之间的间隔必须大于 17ns。RDFIFO# Low to DQ[31:0] Driven 最大为 21ns，若给后续电路的建立时间（Setup Time）取 10ns，则 RDFIFO# 为 Low 的时间应不小于 21ns+10ns=31ns。

➤ 异步 WRFIFO#信号时序

Functional Operation Range ($V_{CC}=5.0V$ 5%, $0^{\circ}C$ to $70^{\circ}C$ T_a , 50 pf load on outputs).

Symbol	Parameter	Min	Max	Units	Notes
t_{141}	WRFIFO# High Time	2		ns	
t_{142}	WRFIFO# Low Time	17		ns	
t_{143}	DQ[31:0] Setup to WRFIFO# Rising Edge	4		ns	
t_{143a}	DQ[31:0] Hold from WRFIFO# Rising Edge	2		ns	
t_{183}	ADD-ON to PCI FIFO WRFULL Valid from WRFIFO# Rising Edge		16	ns	
t_{184}	ADD-ON to PCI FIFO FWE Valid from WRFIFO# Rising Edge		28	ns	

Asynchronous WRFIFO# Timing

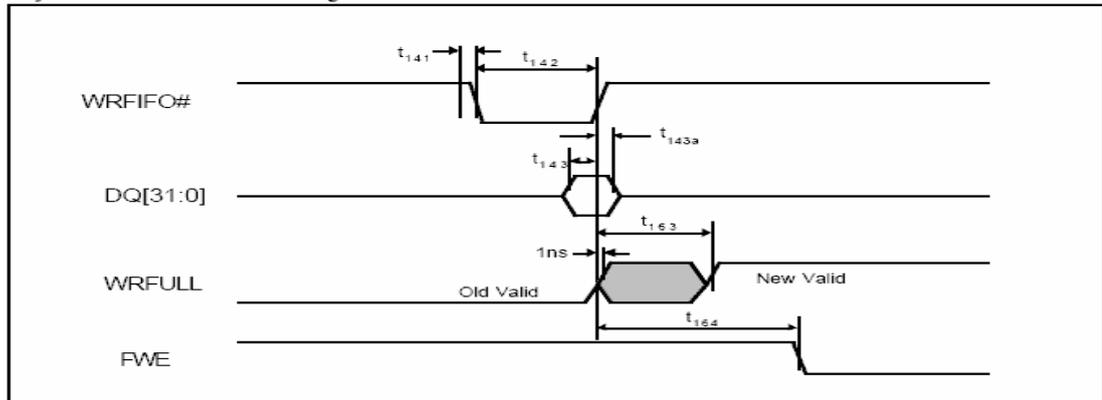


图 40 Asynchronous WRFIFO# Timing

从图 40 看出，WRFIFO# High Time 最小为 2ns，即两次写（WRFIFO#）之间的间隔必须大于 2ns。WRFIFO# Low Time 最小为 17ns，如果 S5933 的后续电路 IO 输出速度达不到 17ns-4ns=13ns，那么 WRFIFO# 的低电平时间就应该延长。

➤ 同步 RDFIFO#信号时序

Functional Operation Range ($V_{CC}=5.0V$ 5%, $0^{\circ}C$ to $70^{\circ}C$ T_a , 50 pf load on outputs).

Symbol	Parameter	Min	Max	Units	Notes
t_{144}	RDFIFO# Setup to BPCLK Rising Edge	8	26	ns	1
t_{145}	RDFIFO# Low Time	8		ns	
t_{146}	RDFIFO# Low to DQ[31:0] Driven		12	ns	
t_{148}	RDFIFO# High to DQ[31:0] Float		3	ns	
t_{149}	DQ[31:0] Valid from BPCLK Rising Edge		16	ns	3
t_{155}	PCI to ADD-ON FIFO RDEEMPTY Valid from BPCLK Rising Edge		12	ns	2
t_{156}	PCI to ADD-ON FIFO FRF Valid from BPCLK Rising Edge		80	ns	

Notes:

1. Min and Max times are indicated to allow increased valid data time as shown by dashed lines.
2. State change of RDEEMPTY shown below is reference only. Actual state change would indicate no Data 3 available.
3. Valid applies after first access. First access is async with following as sync accesses.

Synchronous RDFIFO# Timing

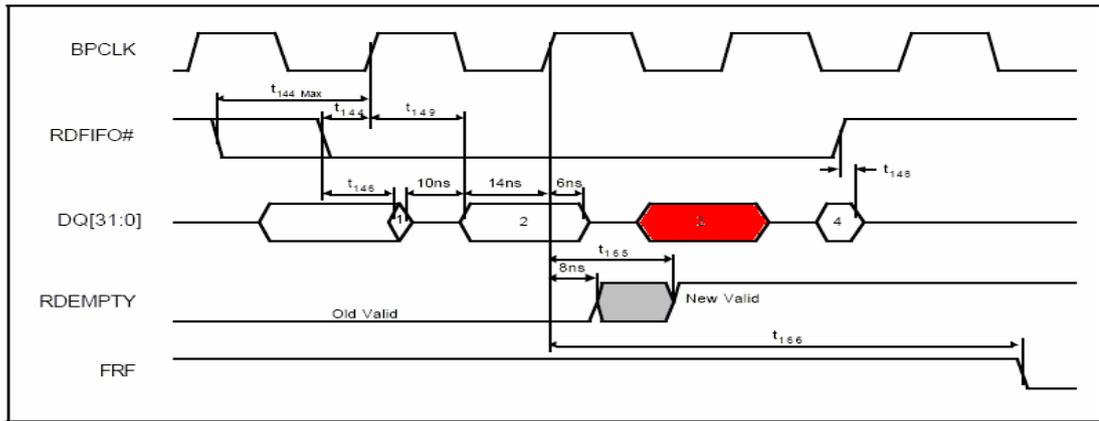


图 41 Synchronous RDFIFO# Timing

依图41的情况，数据3（红色）应该是一个无效数据，因为RDEEMPTY在数据2后变为了有效，指示数据2以后数据无效。

➤ 同步 WRFIFO#信号时序

Functional Operation Range ($V_{CC}=5.0V$ 5%, $0^{\circ}C$ to $70^{\circ}C$ T_a , 50 pf load on outputs).

Symbol	Parameter	Min	Max	Units	Notes
t_{150}	WRFIFO# Setup to BPCLK Rising Edge	12		ns	
t_{150a}	WRFIFO# Hold Time to BPCLK Rising Edge		0	ns	
t_{151}	DQ[31:0] Setup to BPCLK Rising Edge	7			
t_{151a}	DQ[31:0] Hold from BPCLK Rising Edge		0		
t_{157}	ADD-ON to PCI WRFULL Valid from BPCLK Rising Edge		11	ns	1
t_{158}	ADD-ON to PCI FIFO FWE Valid from BPCLK Rising Edge		26	ns	

Notes:

1. State change of WRFULL shown below is reference only. Actual change would indicate no Data 3 written.

Synchronous WRFIFO# Timing

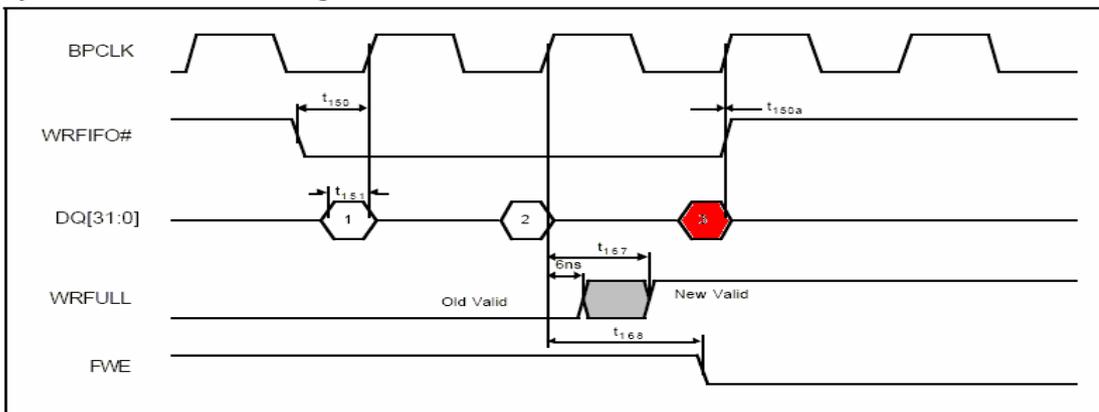


图 42 Synchronous WRFIFO# Timing

依图 42 的情况，数据 3（红色）应该是一个无效数据，因为 WRFULL 在数据 2 后变为了有效，指示数据 2 以后数据无效。

➤ 异步 RD# FIFO 和寄存器 时序

Functional Operation Range ($V_{CC}=5.0V$ 5%, $0^{\circ}C$ to $70^{\circ}C$ T_a , 50 pf load on outputs).

Symbol	Parameter	Min	Max	Units	Notes
t_{110}	SELECT# Setup to RD# Rising Edge	10		ns	
t_{114a}	SELECT# Hold from RD# Rising Edge	-1		ns	
t_{114}	ADR[6:2] Setup to RD# Rising Edge	18		ns	
t_{114a}	ADR[6:2] Hold from RD# Rising Edge	0			
t_{118}	BE[3:0]# Setup to RD# Rising Edge	12		ns	
t_{118a}	BE[3:0]# Hold from RD# Rising Edge	0		ns	
t_{129}	RD# High Time	16		ns	
t_{130}	RD# Low Time	15		ns	
t_{133}	DQ[31:0] Valid from RD# Falling Edge	15		ns	
t_{133a}	DQ[31:0] Hold from RD# Rising Edge	3		ns	
t_{152}	RDEEMPTY Status Valid from RD# Rising Edge		10	ns	
t_{153}	FRF Status Valid from RD# Rising Edge		75	ns	

Asynchronous RD# FIFO and Registers Timing

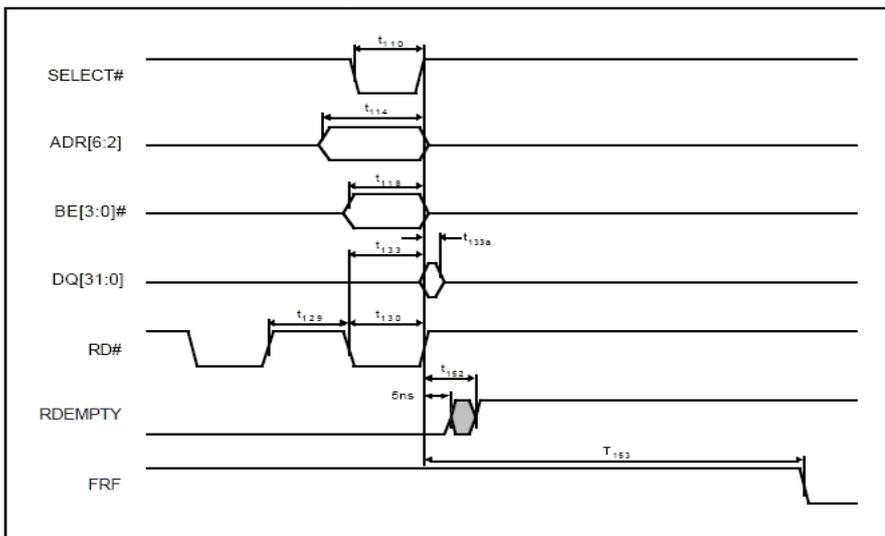


图 43 Asynchronous RD# FIFO and Registers Timing

用 SELECT#、ADR[6:2] 等信号同样可以访问 S5933 的 FIFO，但用 RDFIFO#、WRFIFO# 使用信号少，使访问 FIFO 更简便。

➤ 异步 WR# FIFO 和寄存器 时序

Functional Operation Range ($V_{CC}=5.0V$ 5%, $0^{\circ}C$ to $70^{\circ}C$ T 50 pf load on outputs).

Symbol	Parameter	Min	Max	Units	Notes
t_{111}	SELECT# Setup to WR# Rising Edge	7		ns	
t_{111a}	SELECT# Hold from WR# Rising Edge	0		ns	
t_{115}	ADR[6:2] Setup to WR# Rising Edge	8		ns	
t_{115a}	ADR[6:2] Hold from WR# Rising Edge	0		ns	
t_{119}	BE[3:0]# Setup to WR# Rising Edge	5		ns	
t_{119a}	BE[3:0]# Hold from WR# Rising Edge	0		ns	
t_{131}	WR# High Time	TBD		ns	
t_{132}	WR# Low Time	4		ns	
t_{134}	DQ[31:0] Setup to WR# Rising Edge	2		ns	
t_{134a}	DQ[31:0] Hold from WR# Rising Edge	3		ns	
t_{154}	WRFULL Status Valid from WR# Rising Edge		27	ns	
t_{155}	FWE Status Valid from WR# Rising Edge		40	ns	

Asynchronous WR# FIFO and Registers Timing

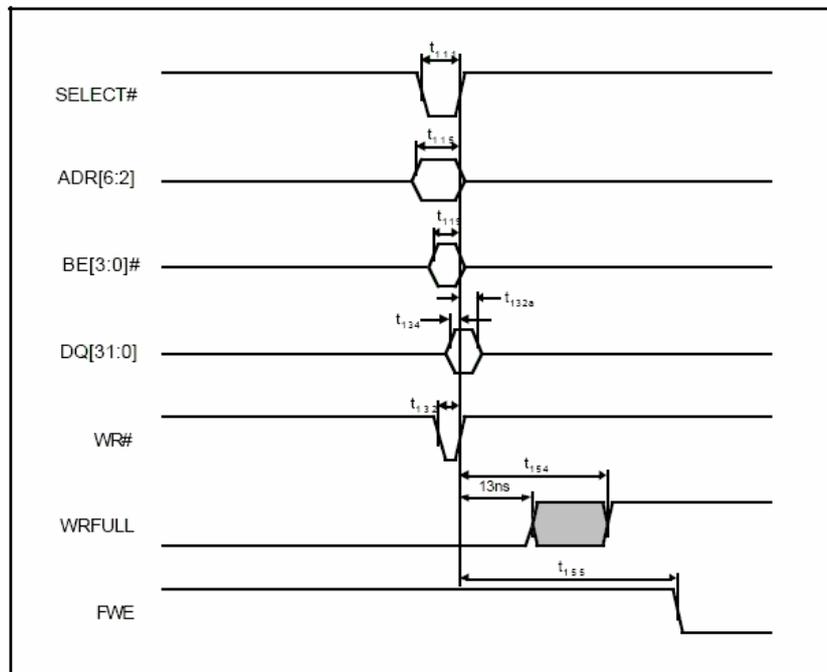


图 44 Asynchronous WR# FIFO and Registers Timing

➤ Pass-Thru 接口时序

Functional Operation Range ($V_{CC}=5.0V \pm 5\%$, $0^{\circ}C$ to $70^{\circ}C$, 50 pF load on outputs)

Symbol	Parameter	Min	Max	Units	Notes
t_{10a}	SELECT# Setup to BPCLK Rising Edge	3		ns	
t_{11a}	SELECT# Hold from BPCLK Rising Edge	2		ns	
t_{13}	ADR[6:2], BE[3:0]# Setup to BPCLK Rising Edge	5		ns	
t_{14}	ADR[6:2], BE[3:0]# Hold from BPCLK Rising Edge	2		ns	
t_{17}	RD# Low to DQ[31:0] Driven		13	ns	1
t_{24}	Pass-Thru Status Valid from BPCLK Rising Edge		5	ns	
t_{25}	Pass-Thru Status Hold from BPCLK Rising Edge	0		ns	
t_{26}	PTRDY# Setup to BPCLK Rising Edge	5		ns	
t_{27}	PTRDY# Hold from BPCLK Rising Edge	3		ns	
t_{28}	PCICKL to BPCLK delay	2	6.5	ns	
t_{29}	RD#, WR# Setup to BPCLK Rising Edge	5		ns	
t_{30}	RD#, WR# Hold from BPCLK Rising Edge	2		ns	
t_{31}	DQ[31:0] Setup to BPCLK Rising Edge	5		ns	
t_{32}	DQ[31:0] Hold from BPCLK Rising Edge	2		ns	
t_{33}	DQ[31:0] Valid from BPCLK Rising Edge		15	ns	
t_{34}	DQ[31:0] Float from RD# Rising Edge		12	ns	

Notes:

1. This timing also applies to the use of BE[3:0]# to control DQ[31:0] drive.

图 45 Pass-Thru 接口时序时间参数表

Pass-Thru Data Register Read Timing

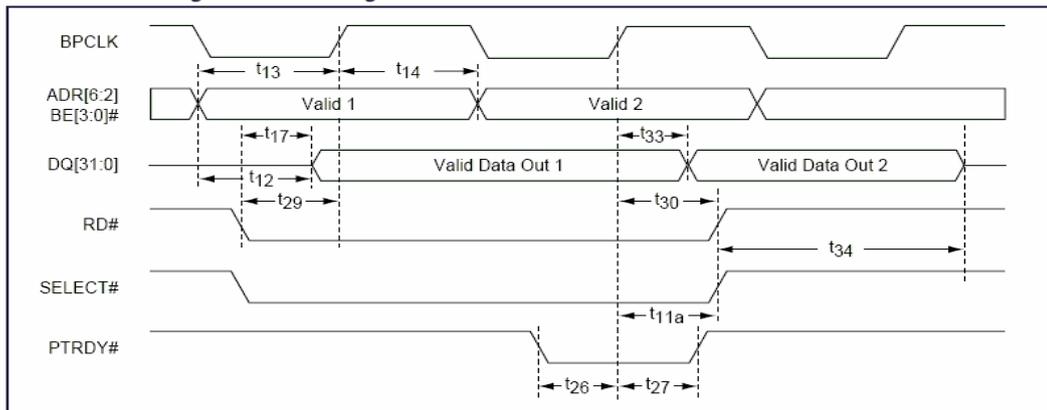


图 46 Pass-Thru 接口读时序

Pass-Thru Data Register Write Timing

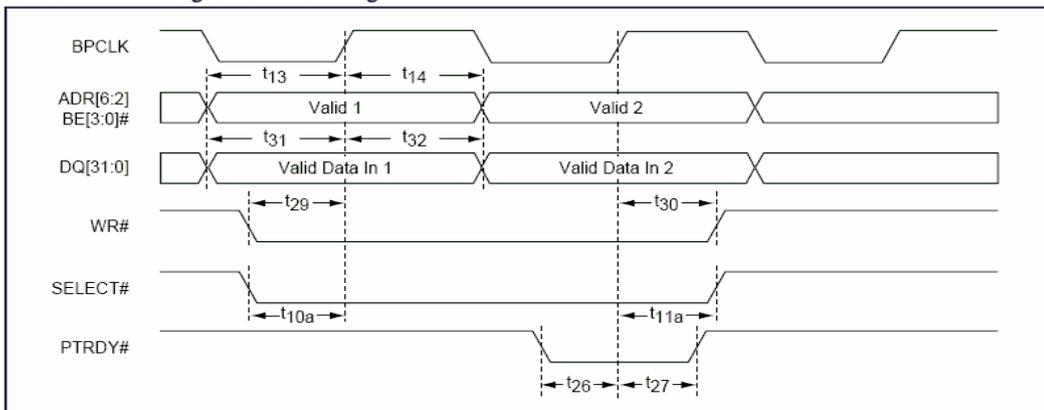


图 47 Pass-Thru 接口写时序

Pass-Thru Status Indicator Timing

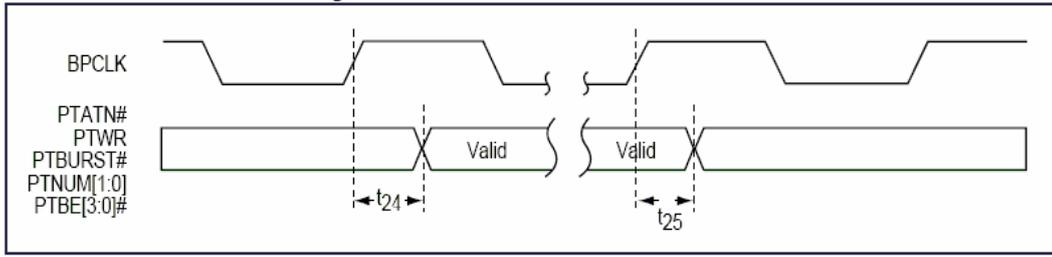


图 48 Pass-Thru 接口状态指示信号时序

九、 FPGA 状态机设计举例

◇ 单周期 Pass-Thru 读、写状态机设计

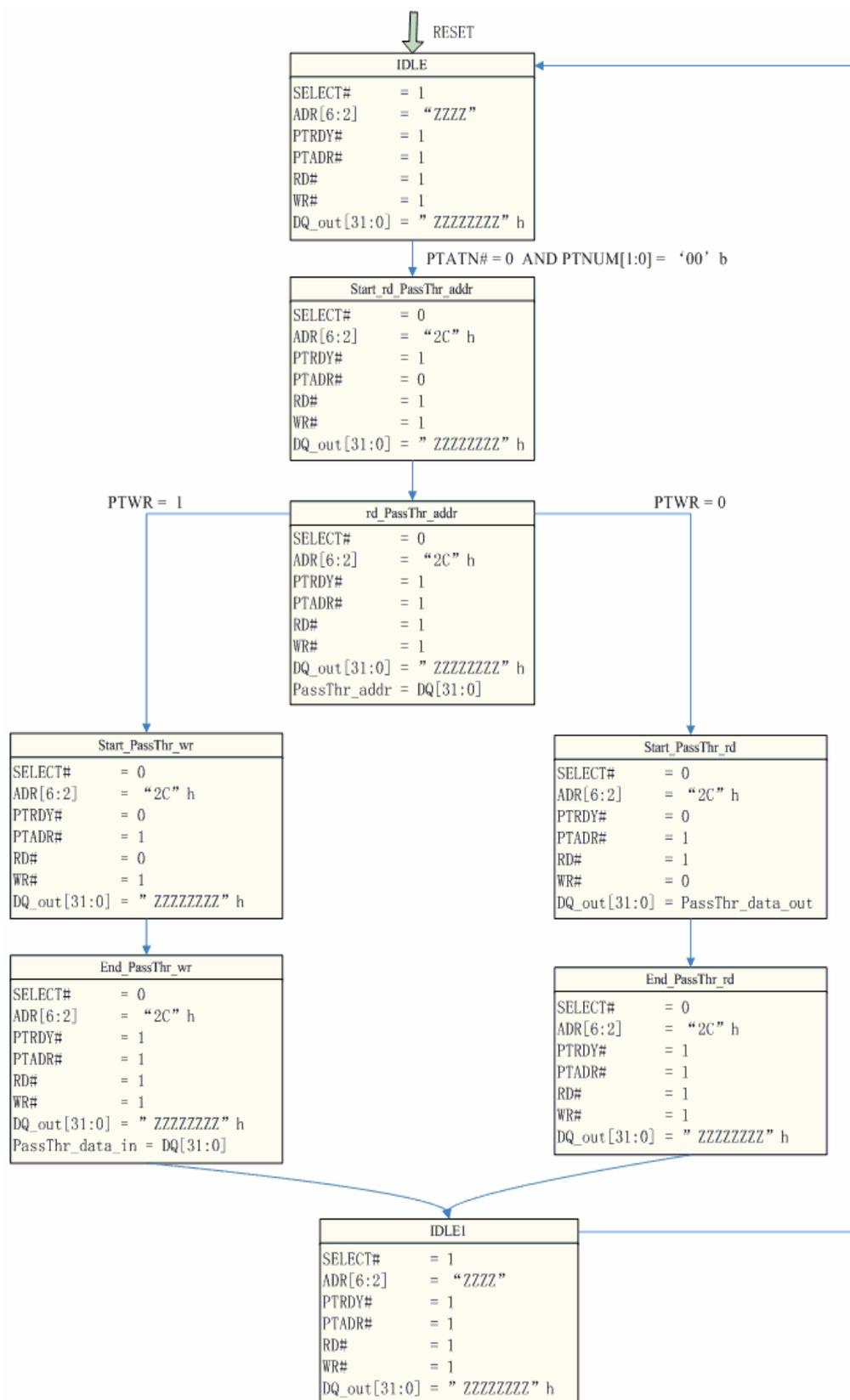


图 49 单周期 Pass-Thru 读、写状态机

假设此 FPGA 的 Pass-Thru 空间配置到基地址 1，所以 S5933 输出 PATAN# 为 Low，

并且 PTNUM[1:0]='00' b 时, 选择此 FPGA 的 Pass-Thru 接口。在 FPGA 内实现三个寄存器 PassThr_addr、PassThr_data_out、PassThr_data_in, PassThr_addr 用于存放从 S5933 读到的 Pass-Thru 的地址偏移, PassThr_data_in 用于存放从 S5933 读到的 Pass-Thru 的数据, PassThr_data_out 用于存放输出给 S5933 的 Pass-Thru 的数据。DQ_out[31:0] 是 FPGA 输出到 S5933 数据总线 DQ[31:0] 上的数据, 当 DQ_out[31:0]="ZZZZZZZ" h 时, 说明 FPGA 不驱动 S5933 的数据总线 DQ[31:0], 即数据方向为从 S5933 到 FPGA; 当 DQ_out[31:0]=PassThr_data_out 时, 说明 FPGA 驱动 S5933 的数据总线 DQ[31:0], 即数据方向为从 FPGA 到 S5933。

◇ S5933 的 FIFO 写访问框图

假设 S5933 配置为从设备方式, S5933 通过 Pass-Thru 给 FPGA 发送命令, 并通过 FIFO 接口读取 FPGA 外接 SDRAM 上的数据。若 PC 要读 FPGA 上 SDRAM 内的数据, 首先从操作系统申请一块内存, 要足够本次 DMA 传输所需求, 再设置 S5933 上的有关寄存器 (见 P39~P40), 然后通过 S5933 的 Pass-Thru 给 FPGA 发送命令启动 DMA 传输。对 PC 主机软件来说, 并不定必须了解所读写数据是从 S5933 上的寄存器, 还是从 Pass-Thru 接口获取, 它只是在访问系统的 IO 或内存空间而已。Add_On_busy 为有效时 (高有效), 表示 S5933 的 Add-On 接口正被 Pass-Thru 占用, 逻辑为 Add_On_busy = !PTATN#, WRFULL 是 S5933 输出的 FIFO 满信号。传送数据的流程如下图 50。

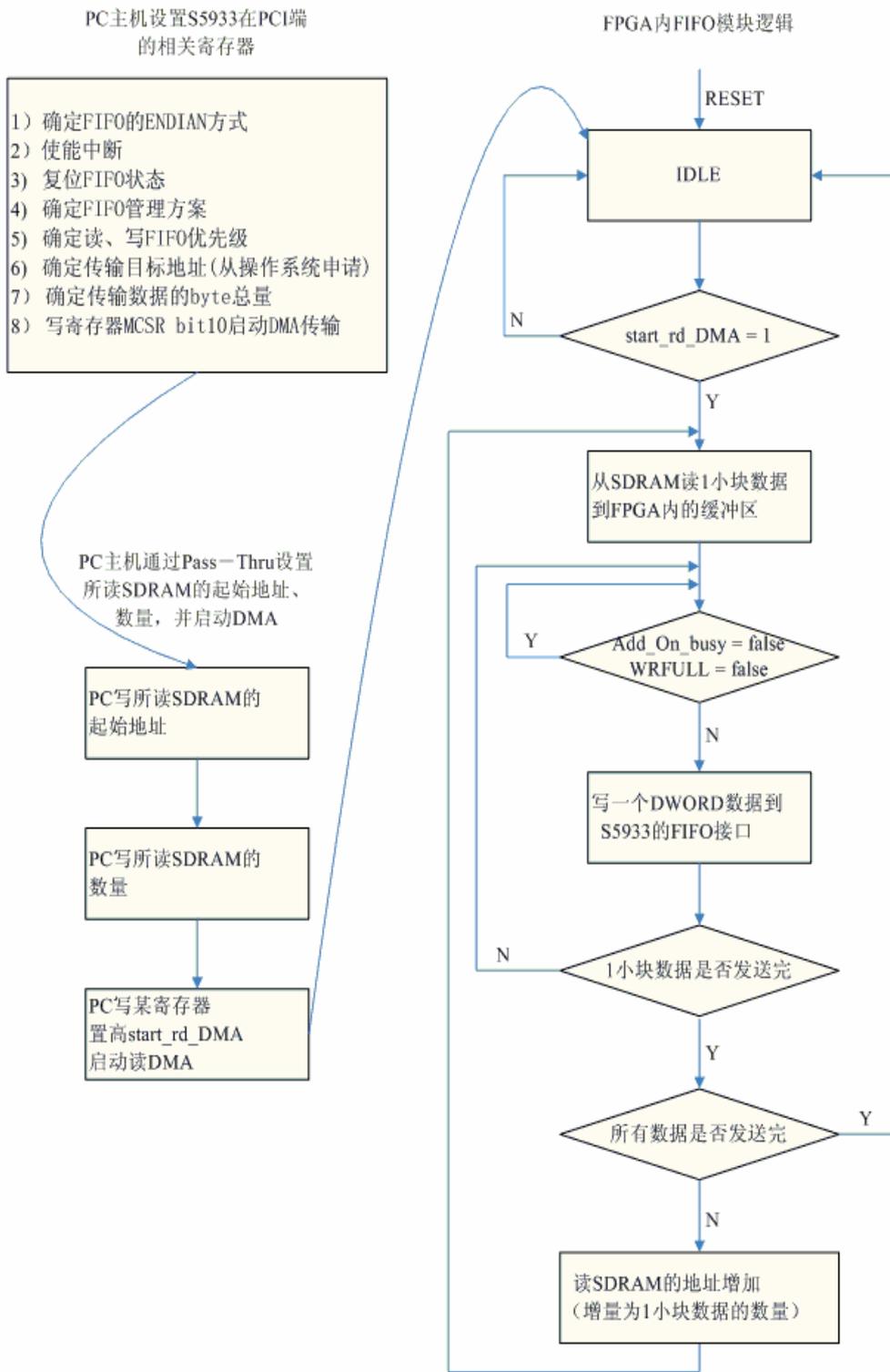


图 50 S5933 的 FIFO 写访问框图

BobWang 2006 03 08
 wdl67@sina.com