# 机载嵌入式 PCI 总线硬件设计和软件测试

摘要：介绍机载嵌入式 PCI 总线硬件设计、软件测试方法。在分布式系统结构设计具有典型性和实用性。内容充实，信息量大，工程实践性强。

关键词：PCI 总线 地址映射 主设备

PCI 总线起源于微型计算机，已经成为微型计算机事实上的总线标准。因其众多的功能、强大的兼容性而独领风骚。为 PCI 局部总线设计的器件是针对 PCI 而 不是针对特定的 CPU 处理器，独立于处理器的升级。其目标是实现电流尽可能小的系统，功耗低。软件透明，在和 PCI 设备之间通信时，软件驱动之间使用相同 的命令集和状态定义。随着嵌入式计算机的发展，PCI 总线也越来越多地被引入到嵌入式系统中。本文介绍在"十五"预研项目中实现嵌入式 PCI 总线的一些经 验体会，与大家切磋。

## 1 PCI 总线概述

PCI（Peripheral Component Interconnect）总线，即外设部件互联总线。在 PCI 应用系统中，如果某设备取得了总线控制权，就称其为"主设备"（master），而被主设备选中以进行通信的设备称为"从设备"（slave）。

系统信号包括复位信号 RST 和时钟信号 CLK。仲裁信号有总线申请 REQ 和总线授权信号 GNT。接口控制信号包括主设备启动 PCI 交易的 FRAME 信号、主设备的交易数据有效信号 IRDY 和目标完成本次数据交易的信号 TRDY 等。PCI 没有一般数据周期的读写信号，而是采用命令编码形式定义本次 PCI 周期 的读写属性。每个 PCI 周期由主设备启动，在第一个时钟周期，AD[31..0]信号承载地址信息，C/BE[3..0]的组合代表命令，定义 PCI 周 期。第二个时钟手，AD[31..0]由提供数据一方驱动，C/BE[3..0]的组合代表有效的字节。详细情况可参阅参考文献，亦即 PCI 规范。

## 2 PCI 总线设计

目前有众多的能支持 PCI 总线的厂家和芯片。其中以美国 PLX 公司的 PCI9056 功能最完备，使用简易。下面的设计以 PCI9056 为主要模型。

### 2. 1 PCI 系统时钟

PIC 总线的信号驱动采用反射波方 式而不是传统的入射波。这样，对各设备采样时刻的偏差要求很高，亦即时钟的偏斜（skew）应尽可能小，最大的时钟偏斜≤2ns。最好整个 PCI 系统各设 备采用同一时刻；但同一时钟的驱动能

力有限，不可能同步驱动所有 PCI 设备。IDT74CT3807 时钟驱动器可以解决这个问题。它将一个时钟源泉转换为 10 个等同的时钟，各时钟之间的偏斜≤50ps。图 2 是 PCI 时钟解决方案。

显然，除去中央资源，这个 PCI 系统最多可以带 9 个设备，对于嵌入式系统来说已经足够了。在 PCI 底板上，为保证时钟的偏斜率，各 PCI 时钟必须走等长线。

2．2 中央资源和 PCI 适配器

在 PCI 系统总线中，必须存在一个 PCI 主桥（Host）管理整个总线。主桥提供系统信号和进行 PCI 部迟疑不决仲裁。PCI9056 可以工作在 Host 方式下，也可以作为普通 PCI 适配器。图 3 是两种工作模式原理。

当 HOSTEN 引脚连接地时， PCI9056 工作在 Host 模式，亦即成为系统的中央资源，在 Host 模式时，PCI9056 的局部复位 LRESET 是输入，接收来自局部 CPU 等的复 位，然后在 RST 产生 PCI 总线复位，去复位整个 PCI 总线上的其它设备。而工作在适配器模式下的 PCI9056 刚好相反，RST 接收来自 PCI 总线的复 位，然后通过 LRESET 去复位 PCI 设备内的其它器件。一般地，把主桥的 PCI8056 设置为总线仲裁器。注意二者仲裁信号 REQ 和 GNT 互换连接。

PCI9056 作适配器时，仲裁信号使用 REQ 和 GNT。当它作为总线仲裁器时，还有其它的请求应答信号对可以使用。图 3 中没有表示出来。

2．3 加电初始化

众所周知，PCI 总线的地址是可以根据系统动态调整的。各个模块在 PCI 空间所占的地址和长度取决于内部配置寄存器。系统程序一般在加电时就检测整个系统所需的空间，分配各 PCI 设备的基址和所需存储器空间、I/O 空间。这个初始化过程可以用三个办法完成。

如果 PCI9056 设计为系统的主 桥（北桥），亦即中央资源（Host），则在它的局部总线一侧都存在 CPU。寄存器的初始化可以由 CPU 进行，也可以由存入 EEPROM 的内部参数自动装 入。如果 PCI9056 作为一般 PCI 设备的适配器，一般没有 CPU，可以由初始化过的主桥通过 PCI 配置周期来设置内部寄存器，也可以由 EEPROM 在 加电时自动载入。图 4 中的 EEPROM 采用仙童公司的 FM93C56 或 FM93CS56 皆可。由局部 CPU 可设置内部所有寄存器，控制逻辑应产生 CCS 片 选。若不采用 EEPROM 加电自动载入初始化参数时，应该在 DI/DO 引脚下拉 1kΩ的电阻。除由局部 CPU 初始化和 EEPROM 加电自动初始化外，更多 的 PCI 设备由主桥通过 PCI 总线来动态初始化。系统软件要保证各个设备的 PCI 空间不重叠。

3 软件设计

PCI 总线不易调试，不但在于硬件设计复杂，还在于驱动软件有相当的难度。但只要把几个基本概念和功能巧妙地体现在软件中，整个驱动软件就很清晰了。以下代码是在 TMS320C6701 环境下的一些成熟的驱动函数。

3．1 配置主桥作为 PCI 总线的主设备

在初始化 PCI 总线其它设备时，中央资源作为 PCI 总线的主设备。此时由局部 CPU 设置所有寄存器，并不产生任务 PCI 周期，亦即局部逻辑必须译码产生 CCS 信号。

```
//功能：配置中央资源的 PCI9056 作为主设备时的参数

//入口参数

//Range：映像范围长度

//PCIBAddr： PCI 基址

//LocBAM:局部存储器基址

//LocBAMI： 局部 I/O 配置基址

void ConfigHostMaster(UINT Range,UINT PCIBAddr,UINT
LocBAM,UINT LocBAMI)

{UINT Aword;

//PCI 命令码寄存器 CNTRL

//D3..0=PCI Read Command Code for DMA

// 缺省 1110b:Memory Line Read,存储器行读

//D7..4=PCI Write Command Code for DMA

// 缺省 0111b:MemoryWrite, 存储器写

//D11..8=PCI MemoryRead Command Code for Direct Master,

// 缺省 0110b:MemoryRead 存储器读

//D15..12=PCI MemoryWrite Command Code for Direct Master.

// 缺省 1110b:MemoryWrite， 存储器写
```

```
//D30=1:复位 PCI 方
//就用这个缺省值，即 000F.767EH
*（int *）LOC_CNTRL=*(int *)LOC_CNTRL|0x400000000;
Aword=0x000FFFFF;
While(Aword--); //复位持续
*（int *）LOC_CNTRL=*(int*)LOC_CNTRL & 0xBFFFFFFF;
//清掉软件复位
//PCI 仲裁控制器 PCIARB
*（int *）LOC_PCIARB=0x00000001;//中央资源要当 PCI 总线仲裁器
//---Direct Master-to-PCI 地址映射--
//局部基址+Range-->PCI 地址映射.--
//局部基址+Range-->PCI 基址+Range
//1.范围寄存器 DMRR
//长度范围值 DMRR
//长度范围值应该是 64KB 的倍数，亦即 D15..0=0000H
//而填入值应该是长度值的被码，即变反+1，如
//64KB=0001.0000H-->FFFE.FFFFH+1=FFFF.0000H
//1MB=0010.0000H-->FFEF.FFFFH+1=FFF0.0000H
*(int *)LOC_DMRR=(～Range)+1;//映像范围
//2.局部存储器基址 DMLBAM（P11-29）
//D15..0:Reserved.
//D31..16:基址高 16 位，必须是范围值的倍数
*（int *）LOC_DMLBAM=LocBAM;
```

```
//3.局部 I/O Configuration 基址 DMLBAI

//当配置访问使能时，对这个寄存器所指的基址进行访问，将在产生 PCI 配置
周期

*（int *）LOC_DMLBAI=LocBAMI;

//4.PCI 基址寄存器 DMPBAM（P11-30）*(int
*)LOC_DMPBAM=PCIBAddr|0xE3;

*(int *)LOC_DMCFGA=0x00000000;//暂时不产生配置周期

*（int *）LOC_DMDAC=0x0; //高 32 位地址始终为 0，需要双地址

//5.命令寄存器 PCICR（P11-8）

//D0=IO Space=1:要响应 I/O 周期.

//D2=Master Enable=1:允许做 Master.

*(int *)PCI9056_PCICR|=0x00000007;

return;

}
```

当 PCI 总线上的其它设备需要访问中央资源时，主桥就成为从设备，所以应该配置其 PCI 空间到局部空间映射的参数。这个过程是一个逆变换，代码在此省略。

3. 2 配置 PCI 总线从设备

当中央资源作为主设备访问其它 PCI 设备时，必须对从设备进行适当的初始化。一般选用 PCI 配置周期设置 PCI 桥的关键配置寄存器，然后用普通 PCI 存储器周期设置其它寄存器。

```
Void ConfigPCISlave(UINT And,UINT Range,UINT PCIBAddr,UINT
LRegPBA,UINT LocBAddr)

//功能:通过主桥产生 PCI 配置周期，去配置 PCI 总线上作为从方的 PCI9056

//入口参数:

//ADn:设备号，亦即和被配置设备的 IDSEL#相连的 PCI 地址
```

```
//Range:范围长度
//PCIBAddr:PCI 基址
//LRegPBA:内部寄存器的 PCI 存储器基址（范围 512B），
//亦即 PCIBAR0 值
//LocBAddr:局部基址
{UINT Abit32W,LROffset;
//------用配置周期-----------------------
//设置命令寄存器 PCICR
//D0=IO Space=1:要响应 I/O 周期
//D1=Memory Space=1:要响应 Memory 周期
//D2=Master Enable=1:允许做 Master
ABit32W=GetConfigReg(ADn,PCI9056_PCICR)|0x07;
//读原值并置位
SetConfigReg(and,PCI9056_PCICR,ABit32W);
//PCIBAR0:其它寄存器的 PCI 基址.
SetConfigReg(ADn,PCI9056_PCIBAR0,LRegPBA);
//设置 PCI 基址，以便能访问其它寄存器
SetConfigReg(ADn,PCI9056_PCIBAR2,PCIBAddr);
//-----以下用 PCI 存储器周期-----------
//1.Space0 的局部地址范围 LAS0RR
LROffset=LRegPBA&0x001FFFFF;//取局部寄存器 PCI 基址的位移
ABit32W=～Range+1; //计算范围值的补码
SetPCIReg(LROffset,PCI_LAS0RR,Abit32W);
```

```
//2.Space0 的局部基址 LAS0BA

SetPCIReg(LROffset,PCI_LAS0BA,LocBAddr|1);

SetPCIReg(LROffset,PCI_EROMBA,0x38);

//4.Space 0/ROM 的局部总线描述符 LBRD0（P11-27）

    //D1D0=11:Space 0-32Bit 数据宽度.(复位缺省)

//D5..2=内部等待状态计数器.

//D6=1:需要 READY#信号.(复位缺省)

//D7=1:允许连续 Burst

//D7=0:Burst-4 Mode (复位缺省)

//D8=0: Space 0 允许预取

//D9=0: 扩展 ROM 允许预取

//用缺省值 40430043H

SetPCIReg(LROffset,PCI_LBRD0,0x40430043);

ABit32W=GetPCIReg(LROffset,PCI_LBRD0);

if(ABit32W!=0x40430043)

printf("局部总线描述符 LBRD0 缺省值出错=%8x",ABit32W);

return;

}
```

PCI 总线上的其它设备身份是动态变化的,所以对有能力做 PCI 主设备的,应该配置其局部空间到 PCI 空间映射的参数。

结语

PCI 局部总线规范也在更新，向更快更强更省电的方向迈进。时钟速率由最初的 33MHz 提高到 66MHz，数据宽度也由 32 位扩展到可支持 64 位，工作电压从 5V 转变为 3.3V，使功耗更小。可以预计，嵌入式 PCI 总线将极大提高机载嵌入式计算机系统的总体性能。