

<<硬件/固件接口设计>>

图书基本信息

书名：<<硬件/固件接口设计>>

13位ISBN编号：9787302267010

10位ISBN编号：7302267014

出版时间：2011-11

出版时间：清华大学出版社

作者：斯特林汉姆

页数：274

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<硬件/固件接口设计>>

内容概要

无论您是一位硬件工程师还是固件工程师，斯特林汉姆编写的《硬件/固件接口设计——提高嵌入式系统开发效率的最佳实践》都将帮助您优化项目或产品中的硬件/固件接口，并理解、计划和消除因项目中的硬件和固件之间不具有最佳的兼容性而可能导致的问题。

《硬件/固件接口设计——提高嵌入式系统开发效率的最佳实践》给出的原则和最佳实践对于硬件工程师和固件工程师都是极具价值的资料。

其中的要点包括寄存器布局、中断、时序与性能、中止与错误等。

实际的案例研究将有助于巩固原则和最佳实践，并使您将精力集中于更清晰的设计、更快的进度，以及更好的实现。

<<硬件/固件接口设计>>

作者简介

作者：(美国)斯特林汉姆(Gary Stringham) 译者：张鼎 贺小川

<<硬件/固件接口设计>>

书籍目录

第1章 引言

1.1 硬件/固件接口

1.1.1 硬件、芯片及组件

1.1.2 固件和设备驱动程序

1.2 最佳实践

1.2.1 原则

1.2.2 原则和实践带来的好处

1.3 “一次成功”还意味着……

1.3.1 更易于编程

1.3.2 更易于调试

1.3.3 更易于变通地解决缺陷

1.4 目标读者

1.4.1 硬件工程师

1.4.2 固件工程师

1.4.3 本书在大学环境中的应用

1.5 项目的生命周期

1.6 案例研究

1.6.1 unity asic中的单色 视频组件

1.6.2 优秀示例的案例研究

1.7 小结

1.8 参考文献

第2章 原则

2.1 硬件/固件接口设计的七项 原则

2.1.1 协同设计

2.1.2 确定并遵循标准

2.1.3 均衡负载

2.1.4 设计要考虑兼容性

2.1.5 预测影响

2.1.6 设计要考虑意外

2.1.7 事先计划

2.2 小结

第3章 合作

3.1 第一步

3.1.1 角色

3.1.2 启动会议

3.2 正式合作

3.2.1 例会

3.2.2 初期的固件支持

3.2.3 联合开发技巧

3.2.4 后期的硬件支持

3.2.5 文档

3.3 非正式合作

3.3.1 正式的组织结构

3.3.2 硬件工程师的主动性

3.3.3 固件工程师的主动性

<<硬件/固件接口设计>>

3.3.4 合作解决问题

3.4 小结

3.5 参考文献

第4章 计划

4.1 行业标准

4.1.1 既有标准

4.1.2 实现标准

4.1.3 标准的扩展与创建

4.2 通用版本

4.3 兼容性

4.3.1 向后及向前兼容的范围

4.3.2 新旧版本的组合

4.4 缺陷

4.4.1 归档缺陷

4.4.2 修复缺陷

4.4.3 查找缺陷的测试计划

4.5 分析

4.5.1 共用引脚

4.5.2 缓冲区管理

4.5.3 硬件/固件交互

4.5.4 分析第三方的ip

4.6 事后检查

4.7 小结

第5章 文档

5.1 类型

5.1.1 文档的级别和类型

5.1.2 芯片级文档与组件级文档

5.1.3 支持与不支持文档

5.2 文档管理

5.2.1 文档标准

5.2.2 何时编写文档

5.2.3 准确性

5.3 审阅

5.3.1 何时审阅

5.3.2 跟踪文档变化

5.3.3 固件工程师应承担的 审阅责任

5.4 内容

5.4.1 大体内容

5.4.2 模板文档示例

5.4.3 历史

5.4.4 功能与假设

5.4.5 参考和教程

5.4.6 术语表和勘误表

5.5 寄存器

5.5.1 寄存器文档

5.5.2 寄存器设计工具

5.5.3 寄存器表

<<硬件/固件接口设计>>

5.5.4 寄存器的细节和描述

5.6 数据位

5.6.1 寄存器映射格式

5.6.2 数据位的位置、类型及默认值

5.6.3 数据位的描述

5.6.4 中止的影响

5.6.5 测试及调试数据位

5.7 中断

5.7.1 边沿触发型中断和电平触发型中断

5.7.2 中断的使能与应答

5.7.3 中断尚未完全完成

5.7.4 无介入的重复中断

5.8 时间

5.8.1 时间范围

5.8.2 时间单位

5.9 错误

5.9.1 两种类型的错误

5.9.2 有关错误的丰富信息

5.9.3 发生错误后组件的状态

5.9.4 固件恢复的步骤

5.10 信息

5.10.1 非法配置

5.10.2 状态机

5.10.3 如何中止

5.11 小结

第6章 超级组件

6.1 超级组件的优点

6.1.1 组件的附属资源

6.1.2 包含未使用逻辑的理由

6.1.3 反对包含未使用逻辑的理由

6.2 联合

6.2.1 设计超级组件

6.2.2 制作超级模块

6.2.3 改进设计

6.2.4 添加未来的功能

6.2.5 超级组件的版本号

6.3 i/o信号

6.4 参数化

6.4.1 减少硅片空间

6.4.2 参数化风险的最小化

6.4.3 固件的参数化信息

6.4.4 可选的与固定的寄存器及数据位

6.5 小结

6.6 参考文献

第7章 设计

7.1 事件通知

<<硬件/固件接口设计>>

- 7.1.1 无指示
- 7.1.2 延时
- 7.1.3 状态位
- 7.1.4 中断
- 7.2 性能
 - 7.2.1 增大缓冲区
 - 7.2.2 提前工作
 - 7.2.3 调整
 - 7.2.4 裕度
- 7.3 上电
 - 7.3.1 上电时的交互
 - 7.3.2 i/o线路的上电状态
 - 7.3.3 组件级的电源控制
- 7.4 通信与控制
 - 7.4.1 错误信息
 - 7.4.2 dma功能
 - 7.4.3 i/o引脚共享
 - 7.4.4 隐藏实现细节
- 7.5 小结
- 第8章 寄存器
 - 8.1 寻址
 - 8.1.1 处理器访问
 - 8.1.2 芯片的基址
 - 8.1.3 组件的偏移量和基址
 - 8.1.4 寄存器的偏移量
 - 8.1.5 子组件
 - 8.1.6 迸发
 - 8.1.7 未使用的地址单元
 - 8.1.8 下一代芯片的变化
 - 8.2 位分配
 - 8.2.1 分配数据位的位置
 - 8.2.2 多位字段
 - 8.2.3 多寄存器字段
 - 8.2.4 未使用的数据位单元
 - 8.2.5 下一版本的变化
 - 8.2.6 数据位类型
 - 8.2.7 寄存器中的数据位类型
 - 8.2.8 根据操作模式编组
 - 8.2.9 组件的多次例化
 - 8.3 数据类型
 - 8.3.1 整数
 - 8.3.2 实数
 - 8.3.3 指针
 - 8.3.4 常数
 - 8.4 硬件标识
 - 8.4.1 芯片id和版本
 - 8.4.2 组件id和版本

<<硬件/固件接口设计>>

8.5 通信与控制

8.5.1 必要的信息

8.5.2 组件中的排队任务

8.5.3 一致的寄存器内容

8.5.4 访问原子寄存器

8.6 小结

第9章 中断

9.1 设计

9.1.1 中断超级模块

9.1.2 分级的中断结构

9.1.3 中断共享

9.1.4 中断源信号的完整性

9.1.5 中断触发的类型

9.2 待决寄存器

9.2.1 应答中断

9.2.2 中断位置的次序

9.3 使能寄存器

9.3.1 使能中断

9.3.2 使能控制中断

9.3.3 使能的默认设置

9.4 可选的寄存器

9.4.1 中断源状态寄存器

9.4.2 抢占寄存器

9.4.3 原子使能寄存器/原子关闭 寄存器

9.4.4 屏蔽寄存器

9.4.5 例化寄存器

9.4.6 可选寄存器的地址

9.5 中断模块回顾

9.5.1 中断通道

9.5.2 中断模块

9.5.3 外部连接

9.6 双边沿触发

9.6.1 利用两个中断通道

9.6.2 头边沿和尾边沿中断的 通道位置

9.7 使用中断模块

9.7.1 何时分配中断通道

9.7.2 重复中断

9.7.3 地址映射

9.8 小结

第10章 中止等

10.1 定义

10.2 停止

10.3 复位

10.4 中止

10.4.1 中止的必要性

10.4.2 固件与中止的交互

10.4.3 中止的行为

<<硬件/固件接口设计>>

10.4.4 中止组件间的交互

10.5 小结

第11章 钩子

11.1 针对钩子的设计

11.1.1 增加哪些钩子

11.1.2 增加寄存器

11.1.3 查找潜在问题区域

11.1.4 删除变通措施

11.2 查看

11.2.1 内部寄存器

11.2.2 信号

11.2.3 存储器

11.2.4 状态机

11.3 打探

11.3.1 破坏性的读写

11.3.2 输入和输出信号

11.3.3 重写寄存器

11.4 监视

11.4.1 事件跟踪

11.4.2 定时器

11.4.3 数据观察

11.5 其他钩子

11.5.1 旁路

11.5.2 测试和调试所需的附加资源

11.5.3 专用处理器

11.6 小结

第12章 结束语

12.1 要点

12.2 受益

12.3 硬件/固件接口设计的七项原则

12.4 产品终于可以运转了！

开始发货吧！

附录a 最佳实践

附录b 电动车控制器的规范

附录c 将本书作为大学教材

附录d 术语表

<<硬件/固件接口设计>>

章节摘录

版权页：插图：设想一下驾驶员（是设备驱动程序还是汽车驾驶员？）要打开车盖才能开车上路，还要在车盖底下用一根长杆子才能控制油门。由于刹车无法工作，因此汽车侧面还装有控制杆用来压下刹车制动器。安全气囊已经预先充气，因为传感器无法工作。这些都不算什么，因为他的头还得探出窗外，因为前面的车盖打开了。气囊已经充气，这使得方向盘上的喇叭不稳定。因此只好用胶带缠在车侧面的浮子罐喇叭代替。因为主油箱漏油，所以车顶还加装了一个油箱，并将油管通入发动机。加装的油管比设计的更粗，因为发动机需要的汽油比预想的多。车头还安装了尺寸更大的散热器和风扇，因为发动机运转起来比预想的温度更高。变通使用的刹车制动器使车门无法打开，因此车门上装有梯子，以便驾驶员进出汽车。车里还有一位惊慌的乘客，他害怕和驾驶员坐在这样的车里。驾驶员正将头探出窗外看看该如何行驶。他看起来很欢欣、很有成就感。他说，“终于可以运转了！开始发货！

！

”图12-2展示了这辆车。

这辆车之所以还能出发，是因为它的硬件和固件中包含足够多的钩子以便使其运转起来。尽管这辆汽车的质量不太好，但是它还能运转，能将乘客送到目的地，还能卖给客户。这其中的很多问题将在下一代汽车中修改。在分析充满漏洞的视频处理器组件时，我们修复了固件并且顺利发货。然而，客户永远不知道必须要这么做。在客户看来，这就是一台正确工作的打印机。并且我们在未增加重新流片的花费和时延的情况下成功研发出产品。

<<硬件/固件接口设计>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>