

4.2 逻辑结构设计

限于篇幅, 本文只介绍设备管理和预约实现中的核心表的逻辑结构。

设备管理模块中五个核心表的关系模式分别如下所示:

管理人员表 - admin (userID,userName,userPassword,pope-dom), 分别表示用户 ID, 用户姓名, 用户密码, 管理权限。

房间信息表 - roomInfo (roomID,roomSerial,deviceNum,userID), 分别表示房间 ID 号, 房间编号, 设备数量, 用户 ID。

设备信息表 - deviceInfo (deviceID,roomSerial,deskSerial,deviceSerial,deviceAssetSerial,deviceSortID,deviceType,currentState), 分别表示设备 ID 号, 房间编号, 座位编号, 设备序列号, 设备资产编号, 设备分类号, 设备型号, 当前状态。

设备问题历史记录表 - deviceHistoryP (deviceHistoryPID,deviceSerial,problemInfo,stateP,occurDate), 分别表示问题记录 ID 号, 设备序列号, 问题信息, 当前状态, 登记出问题的时间。

设备修复记录表 - deviceHistoryR (deviceHistoryRID,deviceSerial,repairInfo,stateR, repairDate), 分别表示修复记录 ID 号, 设备序列号, 修复信息, 当前状态, 修复日期。

在预约管理模块中, 预约功能是由三个表即课程表 - course, 预约表 - book, 审批表 - agree 共同关联实现, 且三个表中的字段相同, 情况比较特殊。

字段 xqID,roomID,KC12,KC34,KC56,KC78 分别表示星期几, 房间编号, 一二节大课, 三四节大课, 五六节大课, 七八节大课。依据高校课程特色, 假设编号为 305 的房间从周一至周五的课程安排可由如下二维表来表示, 当用于课程表 - course 的时候, 0 表示没有课, 1 表示有课; 当用于预约表 - book 的时候, 0 表示没有预约, 1 表示有预约; 当用于审批表 - agree 的时候, 0 表示没有同意预约, 1 表示同意预约。

305	星期一	星期二	星期三	星期四	星期五
KC12	0	1	1	0	0
KC34	1	0	0	1	1
KC56	1	1	1	0	1
KC78	0	0	1	0	1

依据此二维表, 可设计课程表 - course, 预约表 - book, 审批

表 - agree 的关系模型:

course/ agree/book (xqID,roomID,KC12,KC34,KC56,KC78)

现举具体例子来说明预约的实现, 假设查询 305 房间在星期二的第三四节课是否有预约成功, 具体实现条件为: 课程表 - course 中 roomID=305(房间 305),xqID=2(星期二) KC34=0(第三四节没有课); 预约表 - book 中 roomID=305(房间 305),xqID=2(星期二) KC34=1(预约第三四节课); 审批表 - agree 中 roomID=305(房间 305),xqID=2(星期二) KC34=1(同意预约第三四节课)则预约成功。对应的 C# 与 SQL 语句为:

```
//查找符合条件的行数
sqlCommand cmd = new sqlCommand("select count(*) from course
inner join book on course.roomID = book.roomID and book.KC34 = 1
inner join agree on book.roomID = agree.roomID and agree.KC34 = 1
where course.roomID = 305 and course.KC34 = 0",conn);
int i = conver.ToInt32(cmd.ExecuteScalar());
//找到有符合条件的行
if (i > 0)
{
return true;
}
//未找到符合条件的行
else
{
return false;
}
```

5. 结束语

本文介绍了基于 B/S 的实验室管理系统, 使用 C# 语言和 SQL Server2000 对其进行开发和实现。通过该系统对设备进行全程管理, 记录使用, 维修直至报废的整个动态过程, 并实现了网上实验预约功能。它不仅有效的降低了试验管理人员的工作难度及工作量, 并使得实验室的利用率得到大大提高, 深受广大师生的欢迎。

参考文献:

1. 卢银泉, 赖国明. 电脑知识与技术. 基于 C/S 的智能实验室管理系统的分析与设计. 2005.5
2. 卢凤珠, 吴胜达, 赵丽华. 宁波大学学报理工版 2004 年 12 月第 17 卷第 4 期. 基于 C/S 与 B/S 模式的实验室管理系统设计与实现
3. 李舒, 陈丽君. 辽宁大学学报自然科学版 2006 年第 33 卷第 3 期. 高校学生成绩管理系统的设计与实现

(上接第 97 页)

```
END LOOP;
FOR i IN 4 TO 6 LOOP
c(i) <= d2;
d2 <= d1;
d1 <= d0;
d0 <= '0';
END LOOP;
END IF;
END PROCESS;
END arc_cycle;
```

5. 采用 FPGA 实现循环码特性分析

(1)CRC(7, 4)循环码编码器仿真与实现, 采用 ModelSm 仿真器, 上述的循环码编码器仿真波形如图 2 所示。

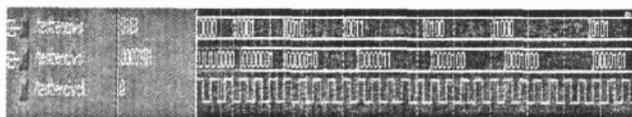


图 2 循环码编码器的仿真波形

(2)对设计进行综合, 其资源开销情况如下所示:

Target Part : ex64-s
Combinational Cells : 0 of 128 (0%)

Sequential Cells : 4 of 64 (6%)
Total Cells : 4 of 192 (3%)
Clock Buffers : 1
IO Cells : 12

6. 结语

从上述的数据可见, CRC 循环码的编码器不太复杂, 使用的资源非常少。因为循环码是在严密的代数学理论基础之上建立起来的, 检(纠)错的能力较强, 目前在理论上和实践上都有了较大的发展。采用 FPGA 现场实现循环码的编码器, 具有较好的应用价值。

参考文献:

1. 单亦先. 循环码的实现方法及其在前向纠错中的应用[J]. 石油大学学报(自然科学版), 2001,(05)
2. 朱宁, 葛元庆, 周润德, 羊性滋. 几种循环码识别电路的分析与设计[J]. 微电子学, 1997,(06)
3. EDA 技术在数字电路设计中的探讨[J]. 实验科学与技术, 2005,(01)
4. 李国华. 循环码在数字通信中的应用[J]. 信息技术, 2003,(07)
5. 周宦银, 朱玲赞. 循环码编码方法研究[J]. 现代电子技术, 2006,(19)