器件功能表数据库的面向对象程序设计技术

管建和 （中国地质大学（北京）信息工程学院 100083）

摘要：在Windows 95/98 系统环境下，用Visual C++开发语言实现测试仪软件系统，其中一个很重要的功能就是器件功能表数据库的建立，本文就这个问题讨论IC器件库面向对象的程序设计技术以及数据应用技术。这些技术具有实用性。

关键词：Windows　Visual C++　测试仪　功能表　面向对象

1　引言

通常，测试仪系统是由测试仪的硬件系统和软件系统组成。硬件系统主要由计算机和测试仪组成；软件系统是指测试仪软件系统。测试仪系统主要用于测试维修电路板上的IC芯片。测试仪软件系统的研制其中一个关键技术就是器件编程技术，通常器件编程主要有逻辑表达式法、功能表法、时序图法、专用测试语言法等几种方式。我们研制的测试仪系统选择所使用的编程语言为功能表方式。

测试仪系统其工作的基本原理是：根据IC器件的功能表（又称真值表）先向器件输入管脚输入表中输入端信息，然后从输出管脚读出信息，再与IC器件功能表输出端的逻辑信息进行比较，如相同则器件功能有效，否则器件为失效。测试仪系统工作原理示意图如图1。

IC器件的逻辑功能表是反映IC器件的逻辑功能信息，每一种器件都有自己的功能表（如图2），把各种IC器件的功能表信息存放在数据文件中就形成器件逻辑数据库了。

2　系统开发环境

硬件环境：CPU 586 硬盘2G RAM 64M 15" 显示器

软件环境：Windows 95/98及以后更新版本视窗操作系统；Visual C++ V4.0及其以后版本

3　器件逻辑功能表数据结构

通常器件的逻辑功能表，具有以下信息：

- 器件名称；
- 器件注解；
- 表头信息；
- 管脚信息；
- 输入端信息；
- 输出端信息；
- 器件特定管脚等其他信息；

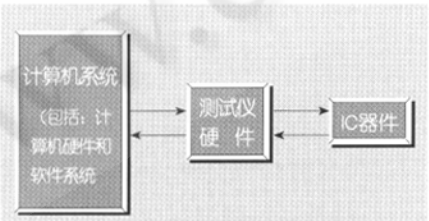


图 1

在Windows 95/98操作系统环境下，我们可以将上述测试器件功能表信息，用面向对象的程序设计语言Visual C++建立IC器件功能表类。定义形式如下：

```
class CChipObject: public Cobject
{
DECLARE_SERIAL(CChipObject)    //声明需要序列化
public:
CChipObject();// 构造函数
virtual void Serialize(CArchive& ar);  // 序列化方法实现
public: // IC器件信息描述
(1) 器件名称；
(2) 器件注解；
(3) 表头信息；
(4) 管脚信息；
(5) 输入端信息；
(6) 输出端信息；
(7) 器件特定管脚等其他信息；
};
```

上述IC器件类信息可以通过可视化界面

控制端				地址端												数据端							
/CE1	CE2	/WE	/OE	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0
20	26	27	22	2	23	21	24	25	3	4	5	6	7	8	9	10	19	18	17	16	15	13	12	11
L	H	L	H	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	H
L	H	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	H

图 2　6264 - 1 存储器功能表

图 3

动态建立。

4 器件逻辑功能表创建过程及算法

(1) 器件对象信息创建与存储：

① 首先在应用程序的文档类中建立列表类对象：

```
CObList m-ChipObjList;     //创建IC器件对象列表类
```

② 然后如增加器件对象则在IC器件编程应用视类中执行下列程序段来实现：

```
CTestAppDoc* pDoc=(CTestAppDoc*) GetDocument ();
ASSERT(pDoc->IsKindOf(RUNTIME-CLASS(CTestAppDoc)));
CChipObject *pChipObj=new CChipObject ();
pDoc->m-Position=pDoc->m-ChipObjList.AddTail(pChipObj);
```

其中：pDoc->m-position 是 POSITION 类型记忆器件在列表中的位置的指针；

(2) 通过人机交互在 IC 器件编程应用视类中建立器件信息，交互界面为(图 3)。

(3) 在文档类可以用下列函数中增加序列化语句进行数据存储：

```
void CTestAppDoc::Serialize(Carchive& ar)
{
m-ChipObjList. Serialize(ar): //增加的序列化语句
}
```

也可以用定制序列化存储技术实现数据存储和装载：

① 打开数据库装载

```
// 以 74 系列 IC 器件为例
void CTestAppDoc::Load74ChipLib (CString m-ChipLibFileName,)
{
m-ChipObjList.RemoveAll( );
//m-bLoadLib=TRUE;
CFile f;
if(f.Open(m-ChipLibFileName,CFile::modeRead)==FALSE)
return;
CArchive ar(&f,CArchive::load);
m-ChipObjList.Serialize(ar);

ar.Close();
f.Close();
}
```

② 实现数据存储

```
void CTestAppDoc:: Save74ChipLib ( CString m-ChipLibFileName)
{
CFile f;
if(f.Open(m-ChipLibFileName,CFile::modeCreate|CFile::modeWrite)==FALSE)
return;
CArchive ar(&f,CArchive::store);
m-ChipObjList.Serialize(ar);
ar.Close();
f.Close();
}
```

(4) 如器件建立完毕，即可在应用框架中用<存储>菜单项或按钮保存器件信息；

(通常，<存储>菜单项或按钮功能由系统缺省建立，也可自建)

5 器件测试方式实现技术

器件测试有以下几种方式：

- 单步测试
- 诊断测试
- 循环测试
- 断点测试
- 快速测试
- 失效测试

除了上述测试之外，还有其他几种辅助性测试，这些测试均要用IC器件的测试信息，获取某 IC 测试器件信息可以用下述方法实现

(使用 pChipObj 指向器件对象信息)：

(1) 首先，在一个文档一视应用框架中，要建立取器件信息控件对象如组合框，列表框等；

并在控件对象所在窗口中要初始化器件对象名称信息。取名 m-ChooseChip，

(2) 然后，建立选择器件消息印射函数，其形式如下：

```
void CTestFormView::OnSelchange-Choosechipcombi();
```

函数体编程主要程序段：

```
ASSERT (pDoc->IsKindOf (RUNTIME-CLASS(CTestAppDoc)));
SetDlgItemText (IDC-STARTTEST, "启动测试");
Int nIndex=m-ChooseChip.->GetCurSel ();
// 其中: m-ChooseChip为取器件信息的控件对象
if (nIndex==CB-ERR)
return;
POSITION m-pos = pDoc->m-ChipObjList->FindIndex( nIndex);
If (m-pos)
{
CChipObject *pChipObj=(CchipObject*) pDoc->m-ChipObjList->GetAt (m-pos);
// 通过 pChipObj 器件对象指针取用器件信息
pChipObj->.....
}
```

(3) 初始化测试信息，形成测试用数据；

(4) 根据测试方式及测试仪测试原理向测试仪发送接受测试数据，给出测试结果信息或图示结果。

6 器件库维护

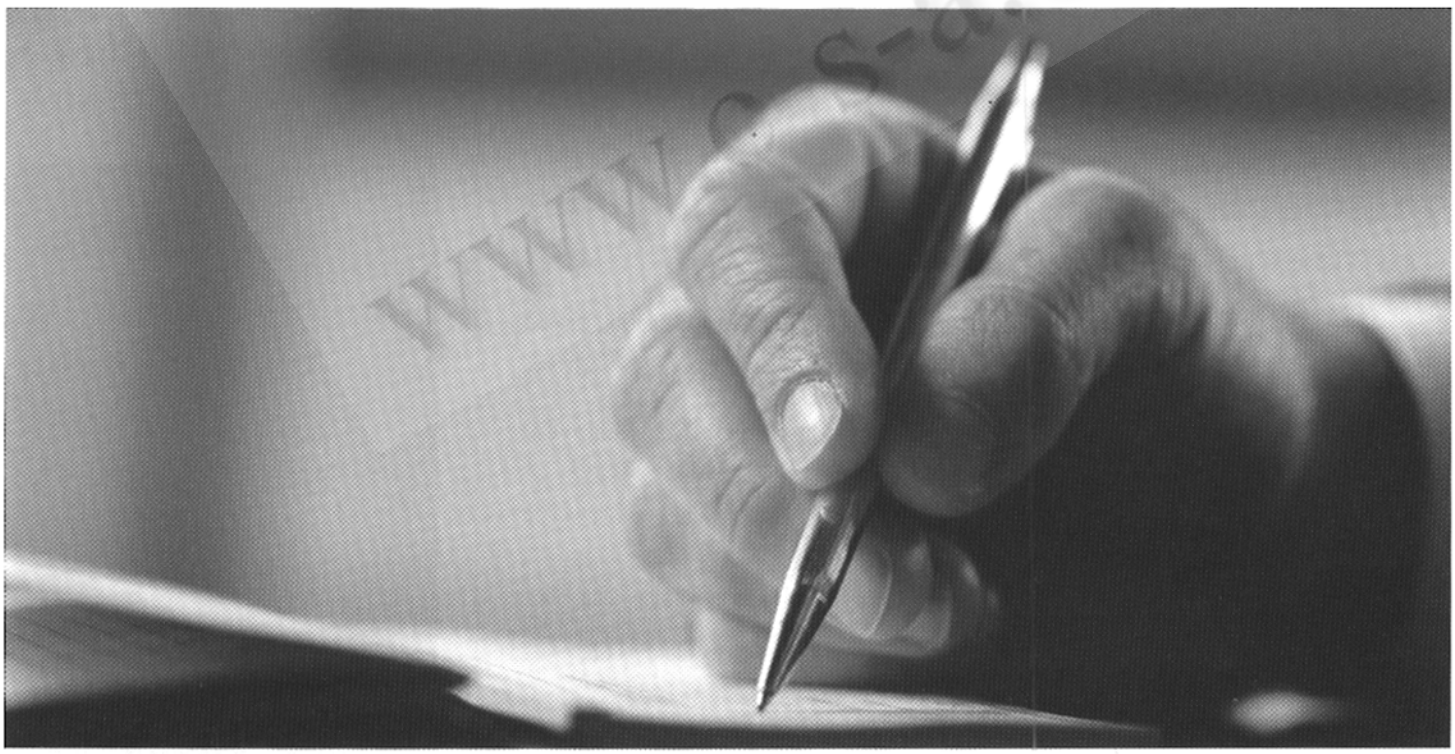
器件库维护主要实现以下功能：

- 增加器件功能表
- 删除器件功能表
- 复制器件功能表
- 修改器件功能表

限于篇幅，下面针对增加器件功能表、删除器件功能表给出实现技术：

增加器件功能表

```
CTl4040Doc* pDoc=(CTl4040Doc*) GetDocument();
ASSERT(pDoc->IsKindOf(RUNTIME-CLASS(CTl4040Doc))); // 获取文档类对象
if(!pDoc->m-position)// 是否指向有效 IC 器件对象
return;
CChipObj *pChipObj=new CChipObj();
m-position=pDoc->m-position=pDoc->m-ChipObjList.AddTail(pChipObj);
// 在表的尾部添加 IC 功能表（真值表）
pDoc->m-NewChip=FALSE;
//IC 功能表（真值表）缺省式样属性初始化
pChipObj=(CChipObj*)m-pList->GetAt (m-position);
```




```
    m-Inputx=pChipObj->m-Inputx;
    m-Inputy=pChipObj->m-Inputy;
    m-Outputy=pChipObj->m-Outputy;
    UpdateData(FALSE); // 更新控件数值
    pDoc->UpdateAllViews(NULL); // 视图中绘出 IC 功能表（真值表）
    删除器件功能表
    CDeleteConfirm dlg;
    if(dlg.DoModal()!=IDOK) return;  //删除确认对话框
    CTl4040Doc*pDoc=(CTl4040Doc*)GetDocument(); // 获取文档类对象
    ASSERT(pDoc->IsKindOf(RUNTIME-CLASS(CTl4040Doc)));
    if(!pDoc->m-position) // 是否指向有效 IC 器件对象
    return;
    POSITION temp-pos; // 选择 IC 器件功能表位置
    temp-pos=m-position;
    pDoc->m-ChipObjList.GetNext(temp-pos); // 将下一个 IC 器件对象置为当前对象
    if(temp-pos==NULL)  //判断是否为有效 IC 器件对象
    {
    temp-pos=m-position;
    pDoc->m-ChipObjList.GetPrev(temp-pos);
    // 无效，将前一个 IC 器件对象置为当前对象
    if(temp-pos==NULL)
    {
    CChipObj* pChipObj=new CChipObj();
    m-pList->AddHead(pChipObj);
    m-position=pDoc->m-position=m-pList->GetHeadPosition();
    }
    }
    CChipObj* pChipObj=(CChipObj*)pDoc->m-ChipObjList.GetAt( m-position);
    pDoc->m-ChipObjList.RemoveAt(m-position);
    // 删除器件对象
    pDoc->m-position=m-position=temp-pos;
        pDoc->m-NewChip=FALSE;
        // 更新视类 - 画出当前功能表
        pDoc->UpdateAllViews(NULL);
```

7 结论

实现测试仪软件系统一个很重要的技术就是器件编程。如何将器件信息组织好并具有很好的人机交互界面是测试仪行业实现器件编程一个很重要的技术领域。本文根据多年来开发体会写出的这篇论文为其他开发类似产品具有较好地启发作用。全文主要介绍了面向对象可视化建立器件编程数据库的编程技术。

参考文献

1 Visual C++ 技术内幕，王国印译，清华大学出版社。

2 Microsoft 基本类库参考手册，张军等译，清华大学出版社。

3 Secrets of the Visual C++ masters, Namir Clement Shammas, Prentice-Hall International Inc。

4 MFC 开发 Windows 95/NT4 应用程序，孙风英 等译，清华大学出版社。

5 TTL 集成电路设计和应用手册，北京市半导体器件二厂。