

目录

第一部分 数字系统	1
第一章 数字集成电路测试系统硬件资源	2
1.1 数字测试系统结构.....	2
1.2 数字测试系统资源板配置.....	3
1.3 数字测试系统适配器.....	6
1.4 小结.....	8
第二章 数字集成电路测试系统软件	10
2.1 系统软件环境.....	10
2.2 系统软件版本.....	10
2.3 基于 C 语言测试程序开发.....	11
2.4 创建 (创建模式).....	15
2.5 设置 (设置模式).....	16
2.6 测试 (测试模式).....	18
2.7 环境版本.....	23
2.8 小结.....	27
第三章 数字集成电路测试指令	28
3.1 图形文件指令.....	28
3.2 指令用法.....	29
3.3 小结.....	31
3.4 编写图形范例 74LS00.....	32
3.5 程序函数目录.....	36
3.6 小结.....	51
第四章 被测器件数据手册	52
第五章 数字器件测试程序开发	53
5.1 [实例一] 54LS00.....	53
5.2 [实例二] 6264 存储器.....	60
5.3 小结.....	75

第二部分 模拟系统	76
第六章 模拟集成电路测试系统硬件资源	76
6.1 模拟测试系统结构	76
6.2 模拟测试系统资源板配置	77
6.3 小结:	80
6.4 模拟测试系统适配器	80
6.5 小结.....	85
第七章 模拟集成电路测试系统软件	86
第八章 模拟集成电路测试指令	87
8.1 模拟测试系统指令集	87
8.2 指令集中 41 条指令的详细用法.....	88
8.3 小结	99
第九章 被测器件数据手册	100
第十章 模拟器件测试程序开发	101
10.1 运放测试范例.....	101

ST3020 集成电路测试系统

第一部分 数字系统

第一章 数字集成电路测试系统硬件资源

1.1 数字测试系统结构

- Windows XP/VIN7/10 操作系统
- 最高测试速度达 20 MHz
- 最大管脚数为 128 PIN
- 测试程序开发语言采用 Visual C++ 6.0 / Visual Studio 2010/菜单编程
- 提供功能测试、交流参数测试和直流参数测试
- 图形发生器深度 1M x 4bit/Pin
- 算法图形发生器地址长度 24 位
- 定时精度 5ns
- 2 路数字器件电压源
- 1 路参数测量单元
- 连接探针台、机械手的接口

ATE 数字测试系统特点：测试系统资源丰富且全面，根据被测器件数据手册自行设计测试方案，理解测试系统硬件结构设计适配器，了解器件类别的测试方法，根据数据手册中的真值表时序用指令写图形向量，理解函数的使用规则编写测试程序。

测试结果=（测试设备使用说明+被测器件数据手册+器件测试方法（国标）+适配器（测试接口板）+图形文件+测试程序）× 调试

说明：为了方便初学者我们设备外接资源提供制作好的适配器供参考，数字器件通用的带拨码开关的测试接口板，省去了设计 PCB 与飞线环节。测存储器需自行设计适配器。

1.2 数字测试系统资源板配置

Solt1	Solt2	Solt3	Solt4-5	Solt6
SBB	PMU	DPS	CHB1-2	PGCB

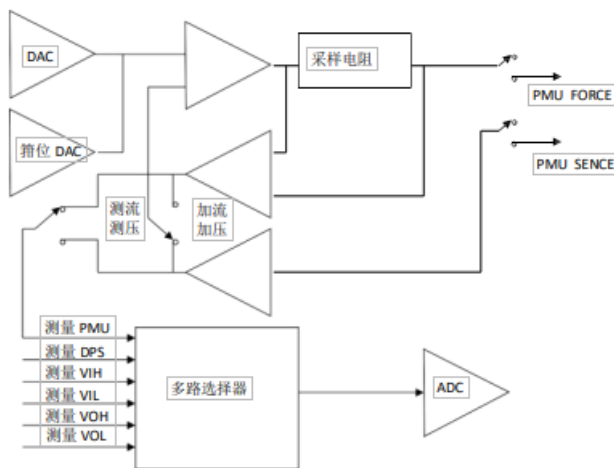
1.2.1 系统总线控制板 (SBB)

系统总线控制板提供 ST3020 测试系统与计算机接口卡、Handler/Prober、测试系统工作 状态指示灯的连接。计算机与测试仪通过网口连接到系统总线控制板，通过测试仪内部网口板转 68 路数据线与总线板链接通信。

1.2.2 参数测量单元/电压电流源 (PMU)

器件的输入与输出管脚的直流参数均用 PMU 测试。PMU 可以通过内部继电器连接到任意管脚，电压施加范围： $\pm 15V$

PMU 有以下两种工作方式：



(1) 加压测流 (FVMI) 方式

在 FVMI 方式中，驱动电压值通过数模转换器提供给输出驱动器；驱动电流由采样电阻采样，通过差分放大器转换成电压值，再由模数转换器读回电流值。箱位值可根据负载设值，箱位电路在这里起到限流保护作用，当负载电流超过箱位值时，PMU 输出变为恒流源，输出电流为箱位电流。测试系统根据箱位值自动选择测流量程。

(2) 加流测压 (FIMV) 方式

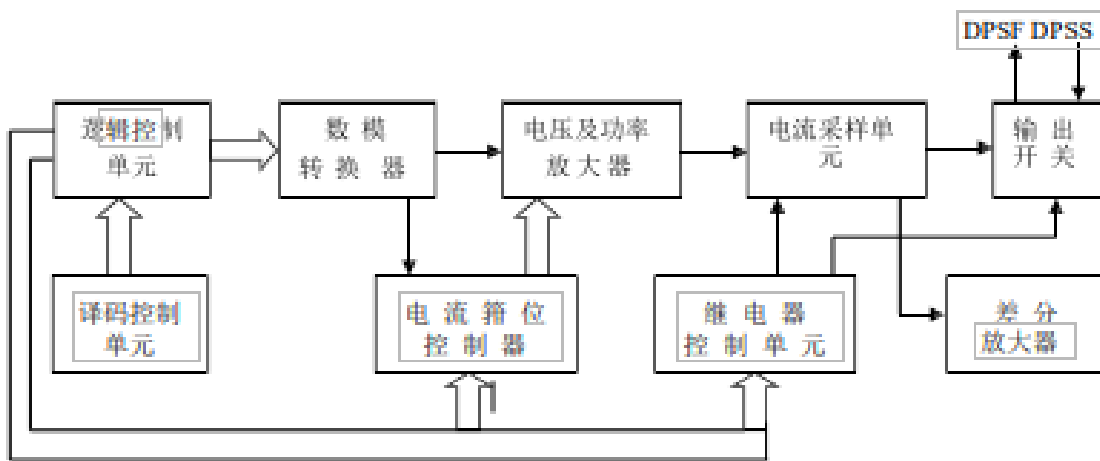
在 FIMV 方式中，驱动电流值通过数模转换器提供给输出驱动器；电压由模数转换器读回。箝位值可根据负载设值，箝位电路在这里起到限压保护作用，当负载电压超过箝位值时，PMU 输出变为恒压源，输出电压为箝位电压。测试系统根据箝位值自动选择测压量程。

1.2.3 器件电压源 (DPS) 主要功能：

(1) 被测器件的电源管脚需要连接到器件电压源 (DPS)，在测试过程中为被测器件提供工作电源，电压施加范围： $\pm 15V$ 。

(2) 测量器件电源的工作电流，测量电流范围： $\pm 250mA$ 。

器件电源板分别由 DPS1, DPS2 两路器件电源组成，它的主要组成部分：译码控制单元、逻辑控制单元、数模转换器、电压及功率放大器、电流箝位控制器、电流采样单元、差分放大器、继电器控制单元。



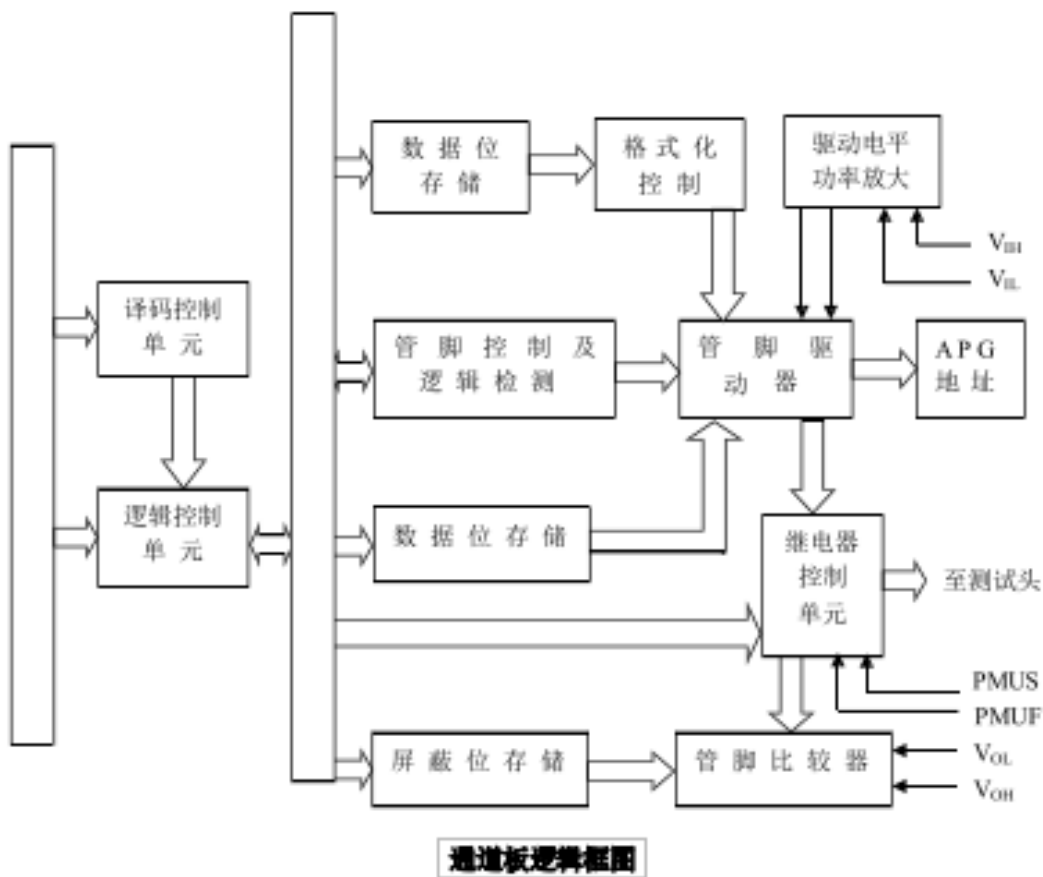
DPS 板逻辑框图

1.2.4 数字通道板 (CHB1-2) 主要功能：

(1) 在功能测试过程中，直接与被测器件的输入、输出管脚相连。

(2) 在功能测试过程中，向被测器件输入管脚按规定的电平、逻辑、格式和定时施加激励信号，同时检测输出管脚信号电平，并与预期的响应信号进行逻辑比较。

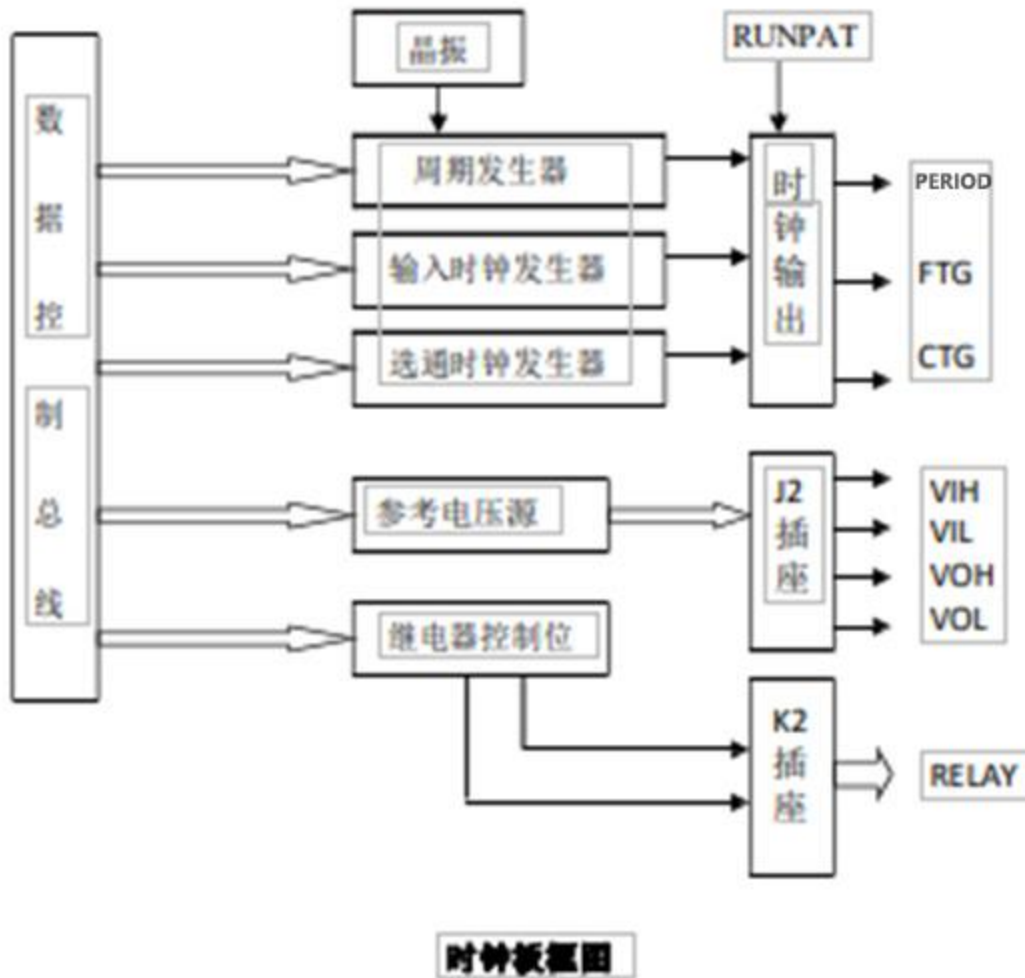
(3) 在直流测试过程中，完成参数测量单元 (PMU) 与被测器件的每一个管脚的切换。主要组成部分：译码控制、逻辑控制、数据位存储、驱动位存储、屏蔽位存储、格式化控制、管脚控制及检测单元、管脚驱动器、管脚比较器、继电器矩阵、驱动电平功率放大及自动算法产生器 (APG) 地址。



1.2.5 图形时钟板 (PGCB) 主要功能

图形部分由图形发生器 PATTERN 和算法图形发生器 APG 两部分组成。APG 主要用于算法图形的生成；例如存储器测试，测试图形可用 APG 算法生成，从而大大简化测试图形。

图形发生器的主要特性： 图形深度： 1M x4bit/Pin、 算法图形宽度： 24bit 。时钟部分包含三部分： 时钟发生器、 参考电压源、 继电器控制位。时钟发生器产生时钟周期、 前沿、 宽度、 选通等信号。



时钟周期信号定时范围： PERIOD 50ns~10ms FTG 前沿、宽度定时范围： 50ns~10ms
选通脉冲 CTG 定时范围： 50ns~10ms

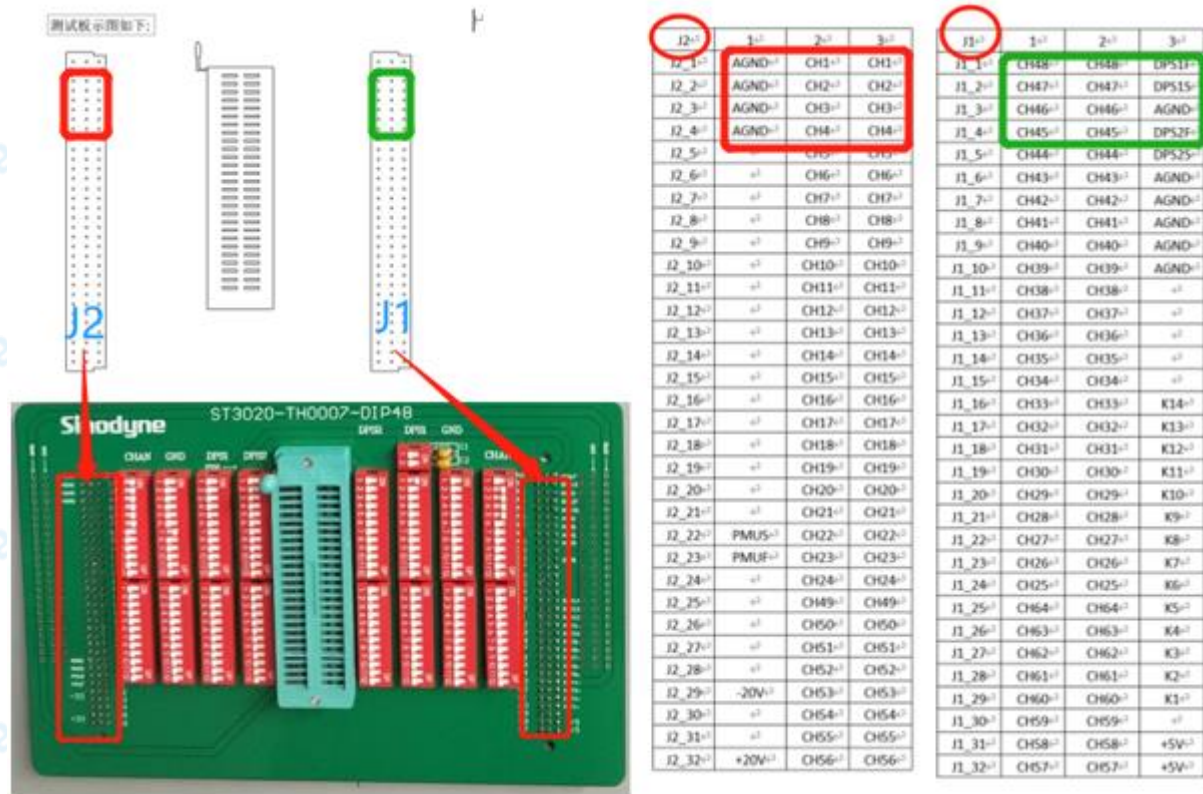
以上 1-5 类板卡属于设备内部功能结构。

小结：介绍测试系统结构，以上是测试设备机箱内部各板卡功能，理解测量原理，学习使用测量仪器仪表，更便于理解 ATE 测试系统的工作原理。

1.3 数字测试系统适配器

适配器是设备系统外接被测器件将各测试通道、DPS、AGND、控制继电器转接到适配器，通过适配器上的插座与待测器件连接。

1.3.1 适配器管脚定义如下：



测试板示意图如下：

J2	1 st	2 nd	3 rd
J2_1 st	AGND ⁻	CH1 ⁻	CH1 ⁻
J2_2 nd	AGND ⁻	CH2 ⁻	CH2 ⁻
J2_3 rd	AGND ⁻	CH3 ⁻	CH3 ⁻
J2_4 th	AGND ⁻	CH4 ⁻	CH4 ⁻
J2_5 th		CH5 ⁻	CH5 ⁻
J2_6 th		CH6 ⁻	CH6 ⁻
J2_7 th		CH7 ⁻	CH7 ⁻
J2_8 th		CH8 ⁻	CH8 ⁻
J2_9 th		CH9 ⁻	CH9 ⁻
J2_10 th		CH10 ⁻	CH10 ⁻
J2_11 th		CH11 ⁻	CH11 ⁻
J2_12 th		CH12 ⁻	CH12 ⁻
J2_13 th		CH13 ⁻	CH13 ⁻
J2_14 th		CH14 ⁻	CH14 ⁻
J2_15 th		CH15 ⁻	CH15 ⁻
J2_16 th		CH16 ⁻	CH16 ⁻
J2_17 th		CH17 ⁻	CH17 ⁻
J2_18 th		CH18 ⁻	CH18 ⁻
J2_19 th		CH19 ⁻	CH19 ⁻
J2_20 th		CH20 ⁻	CH20 ⁻
J2_21 th		CH21 ⁻	CH21 ⁻
J2_22 th	PMUF ⁻	CH22 ⁻	CH22 ⁻
J2_23 th	PMUF ⁻	CH23 ⁻	CH23 ⁻
J2_24 th		CH24 ⁻	CH24 ⁻
J2_25 th		CH49 ⁻	CH49 ⁻
J2_26 th		CH50 ⁻	CH50 ⁻
J2_27 th		CH51 ⁻	CH51 ⁻
J2_28 th		CH52 ⁻	CH52 ⁻
J2_29 th	-20V ⁻	CH53 ⁻	CH53 ⁻
J2_30 th		CH54 ⁻	CH54 ⁻
J2_31 th		CH55 ⁻	CH55 ⁻
J2_32 th	+20V ⁻	CH56 ⁻	CH56 ⁻

J1	1 st	2 nd	3 rd
J1_1 st	CH48 ⁻	CH48 ⁻	DPS1 ⁻
J1_2 nd	CH47 ⁻	CH47 ⁻	DPS1 ⁻
J1_3 rd	CH46 ⁻	CH46 ⁻	DPS1 ⁻
J1_4 th	CH45 ⁻	CH45 ⁻	DPS2 ⁻
J1_5 th	CH44 ⁻	CH44 ⁻	DPS2 ⁻
J1_6 th	CH43 ⁻	CH43 ⁻	AGND ⁻
J1_7 th	CH42 ⁻	CH42 ⁻	AGND ⁻
J1_8 th	CH41 ⁻	CH41 ⁻	AGND ⁻
J1_9 th	CH40 ⁻	CH40 ⁻	AGND ⁻
J1_10 th	CH39 ⁻	CH39 ⁻	AGND ⁻
J1_11 th	CH38 ⁻	CH38 ⁻	
J1_12 th	CH37 ⁻	CH37 ⁻	
J1_13 th	CH36 ⁻	CH36 ⁻	
J1_14 th	CH35 ⁻	CH35 ⁻	
J1_15 th	CH34 ⁻	CH34 ⁻	
J1_16 th	CH33 ⁻	CH33 ⁻	K14 ⁻
J1_17 th	CH32 ⁻	CH32 ⁻	K13 ⁻
J1_18 th	CH31 ⁻	CH31 ⁻	K12 ⁻
J1_19 th	CH30 ⁻	CH30 ⁻	K11 ⁻
J1_20 th	CH29 ⁻	CH29 ⁻	K10 ⁻
J1_21 th	CH28 ⁻	CH28 ⁻	K9 ⁻
J1_22 th	CH27 ⁻	CH27 ⁻	K8 ⁻
J1_23 th	CH26 ⁻	CH26 ⁻	K7 ⁻
J1_24 th	CH25 ⁻	CH25 ⁻	K6 ⁻
J1_25 th	CH24 ⁻	CH24 ⁻	K5 ⁻
J1_26 th	CH23 ⁻	CH23 ⁻	K4 ⁻
J1_27 th	CH22 ⁻	CH22 ⁻	K3 ⁻
J1_28 th	CH21 ⁻	CH21 ⁻	K2 ⁻
J1_29 th	CH20 ⁻	CH20 ⁻	K1 ⁻
J1_30 th	CH19 ⁻	CH19 ⁻	
J1_31 th	CH18 ⁻	CH18 ⁻	+5V ⁻
J1_32 th	CH17 ⁻	CH17 ⁻	+5V ⁻

注：PMUF：PMU 施加信号 PMUS：PMU 测量信号 K：表示继电器
+5：表示输出+5V 电源 ±20V：表示输出±20V 电源

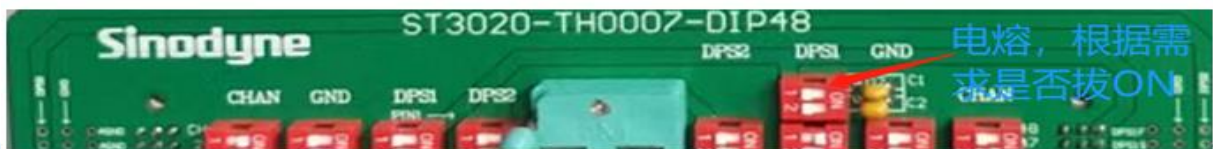
CHAN: GND: DPS1: DPS2: DPS2: DPS1: GND: CHAN:←

每横排只能一个拨 ON←

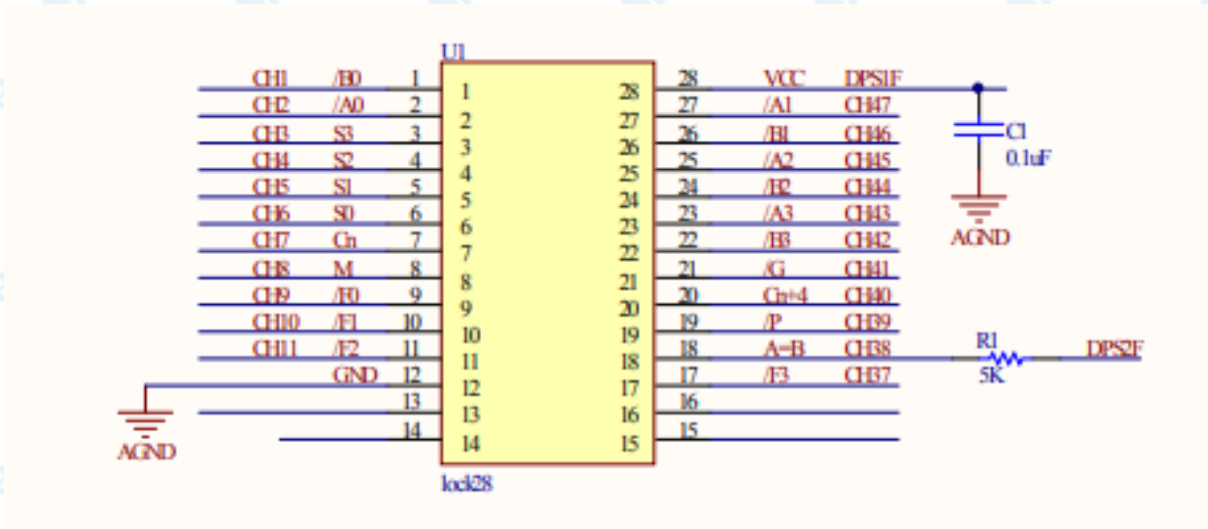
被测器件的电源管脚，须连接到 DPS1 或 DPS2，与后续的程序对应←

被测器件的地管脚对应 GND 拨码到 ON←

被测器件的其他输入或输出管脚须把对应的 CHAN 拨码到 ON←



说明：要理解这些结构才能设计出 PCB，从插座引过来的资源，红色的字是网络标号，相同的两个就代表他们是连接在一起的，如下图：



1.3.2 工作环境

- (1) 使用电压： 交流 220V±10%， 50Hz±2%， 功率： < 1.5KW
- (2) 操作温度： +23°C±3°C 相对湿度： < 70%
- (3) PC 配置： Intel Pentium 双核处理器
- (3) 2G 以上内存
- (4) 250G 以上硬盘空间 Windows XP/V7/V10 系统 Visual C++6.0 编程语言
- (5) 电网电压： AC220V±10%；
- (6) 电网频率： 50 Hz±5Hz
- (7) 功率： 500W -1500W (不同配置功率不同)
- (8) 湿度： 30%~60%

1.4 小结

被测器件与测试设备建立硬连接就是通过适配器，根据被测器件的封装工艺采购匹配的测试锁紧座，根据测试系统接口板原理图设计 PCB，最后将锁紧座焊接好，进行连接性测试。

图片中所提供的是带有拨码开关的适配器，是测试数字器件的通用适配器，器件管脚与设备通道的连接通过拨码来控制。自行设计时也可以采用飞线替代拨码的方式进行。

第二章 数字集成电路测试系统软件

2.1 系统软件环境

ST3020 集成电路测试系统软件是在 WindowsXP 环境下用 Visual C++ 6.0 作为系统开发工具开发工具，也可以在/win7/win10 环境下用 Visual Studio 2010 作为系统开发工具进行测试程序开发。

测试系统软件包括：测试处理、测试数据显示、数据统计、测试程序管理、测试程序框架自动生成、测试程序调试和设备自检，多种功能集成在一起。提供操作方便的用户界面，使用者通过菜单、工具条，快捷键等操作程序。

2.2 系统软件版本

Visual C++ 6.0 与 Visual Studio 2010 系统均支持两种版本：

2.2.1 C 语言编程开发测试程序

用 Visual C++ 6.0 或 Visual Studio 2010 作为开发工具。不需要对 C++ 的编写方法有很深的了解，系统自动生成测试程序框架，并提供多个专用的测试函数。测试程序开发人员只需编写测试过程，显示、统计等功能系统自动处理，不需要用户再次开发，提高测试程序的开发效率。本手册重点详细介绍用 C 语言开发测试程序系统软件操作规则。

采用以上版本，可以根据被测器件的特殊要求编写测试程序，灵活性高，程序测试速度快。Visual C++ 6.0 版本在 WindowsXP/win7/环境下运行，Visual Studio 2010 版本在 WindowsXP/win7/ win10 环境下运行。

2.2.2 菜单式开发测试程序

采用此版本，不必编写测试程序，只要将每项参数填写完整，即可进行测试，操作简单方便，容易掌握，适用于测试条件简单的器件。但测试速度比采用编程方式开发的测试程序慢。建议数模电零基础使用，方便简单支持自学。

2.2.3 在 WindowsXP/win7 环境下 Visual C++ 6.0 版本作为开发工具，C 语言版本的视频 https://www.bilibili.com/video/BV1FN41197NY/?spm_id_from=333.788.videocard.7

2.3 基于 C 语言测试程序开发

WindowsXP/win7 环境下 Visual C++ 6.0 作为开发测试程序的工具为例，测试程序以动态连接库的形式生成，由系统操作软件调用执行。在 WINDOWS 下运行 ST3020.exe 文件，开始执行系统软件。

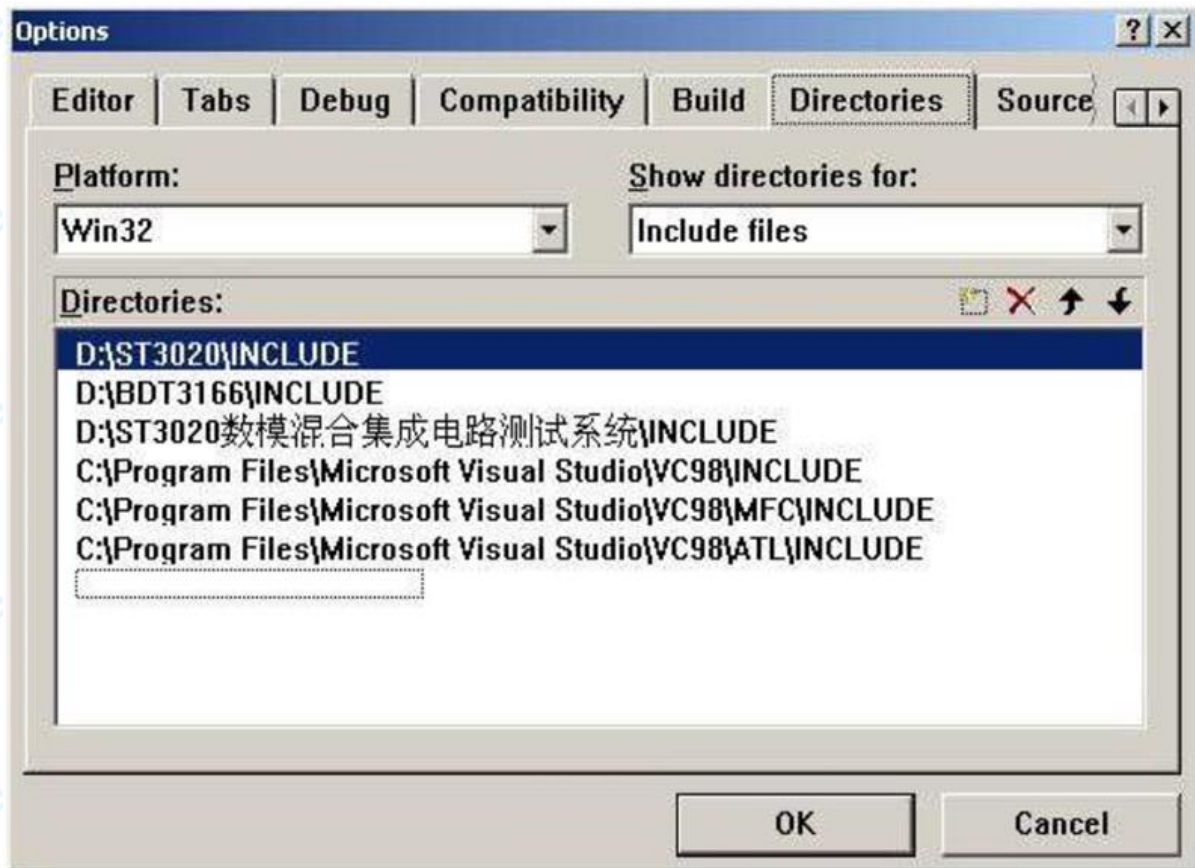
ST3020 系统软件有四种运行模式：创建、图形编辑、设置、测试。

2.3.1 设置 VC++环境

安装完成后，必须设置 Microsoft Visual C++6.0 的环境，具体步骤如下：

(1) 运行 VC++，进入到 Microsoft Visual C++6.0 的编辑环境。

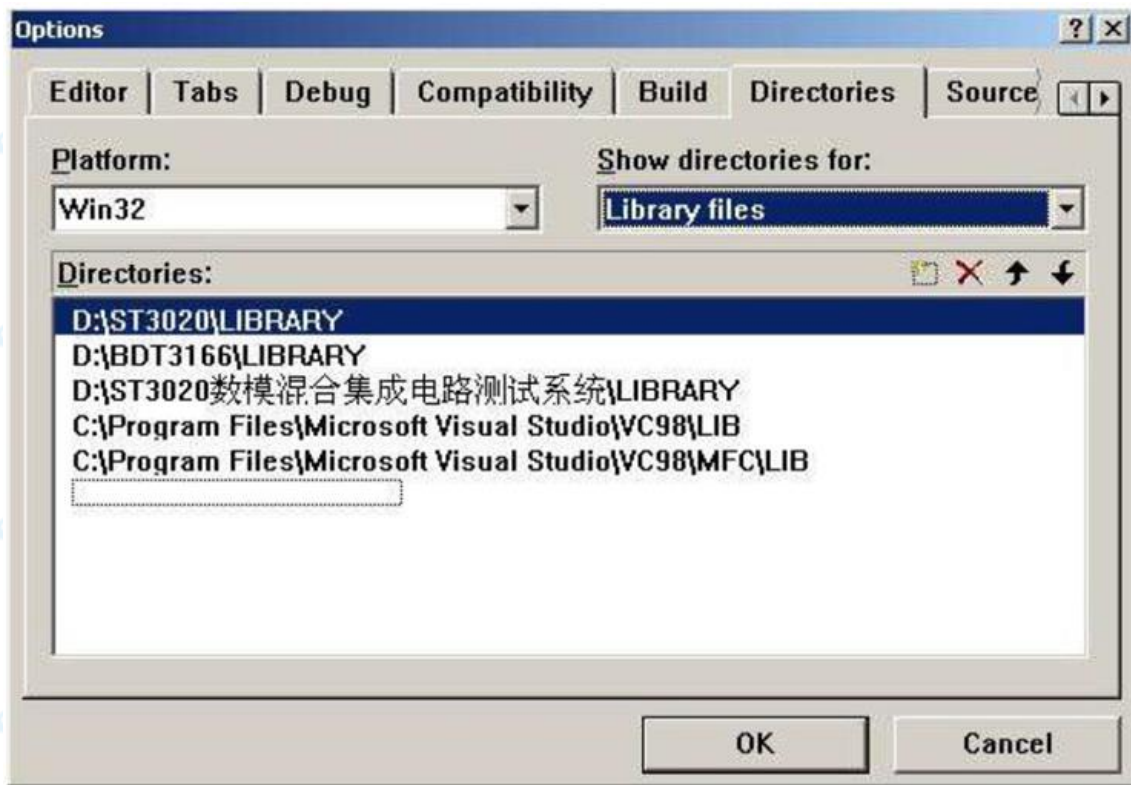
(2) 设置 include 路径，单击菜单项 Tools，单击 Options...，弹出 Option 窗口选择标签 Directories，对话框如下：



设置 include 路径

(3) 设置 Include files 的路径，在 Show directories for 列表中选择 Include files，在 Directories: 中添加新的目录，选择测试系统目录下的 INCLUDE 目录，例如上图中的 d:\ST3020\INCLUDE。

(4) 设置 Library files 的路径，在 Show directories for 列表中选择 Library files，在 Directories: 中添加新的目录，选择测试系统目录下的 LIBRARY 目录，例如下图中的 d:\ST3020\LIBRARY。



设置 Library files 路径

2.3.2 修改设置文件

安装完成测试软件后，要修改设置文件 setting.ini，保证设置文件的内容与安装的测试程序保持一致。设置文件内容如下：

[电阻值]

R1=500000

R2=50000

R3=5000

R4=500

R5=10

[校准文件路径]

校准文件路径=D:\ ST3020\Califile

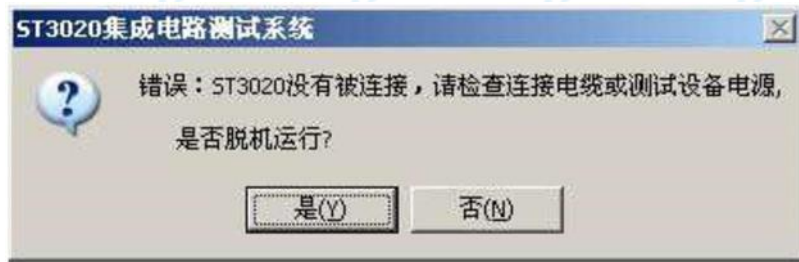
自检文件路径=D:\ ST3020\Califile

(1) 电阻值部分一般不改动，只有当数值与自检&校准板上电阻值不对应时才改动。

(2) 校准文件路径要与系统软件的安装路径对应，当系统软件安装的路径与设置的路径不对应时，修改设置文件。

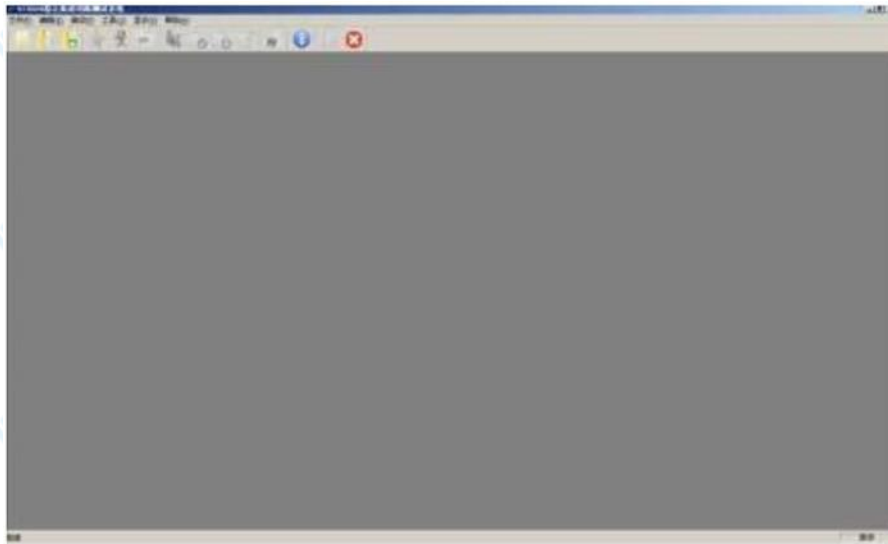
2.3.3 软件编辑方式及使用

在 WINDOWS 下运行 ST3020.exe 文件，开始执行系统软件，如系统未开机或电缆连线有问题，将出现以下窗口：



问题窗口示意图

系统正常连接后，将弹出用户名及密码窗口，填写用户名和密码正确后，系统进入如下窗口。如果没有设置密码，直接进入如下窗口：




系统主窗口

单击“X”按钮或“ESC”键退出。



此窗口显示各种模式进入按钮，各种模式说明如下。

2.4 创建（创建模式）

单击  按钮，创建新的测试程序库，单击后弹出测试程序选择窗口。

此窗口包括：**新建(N)**、**插入(I)**、**移出(R)**、“程序选择列表框”。

新建(N)：单击新建按钮，弹出对话框，填写文件名，建立新的测试程序动态库源文件。



新建窗口

插入(I)：向“程序选择列表框”中加入测试程序。单击按钮，显示文件对话框，选择测试程序，文件扩展名为 .dsw 的 C++工程文件。将此加入到列表框。

移出(R)：从“程序选择列表框”中移出测试程序选项，但不删除。



选择测试程序：在“程序选择列表框”中选中要用的测试程序，确定或双击，进入测试状态。

2.4.1 图形编译


图形编译与查错程序 ComplierDlg.exe 文件是一个单独的执行程序，包括编译图形文件，提示错误信息。执行 ComplierDlg.exe 文件，弹出如下对话框，具体功能如下：




编译图形对话框


在文件名编辑框中填写要编译的图形文件的完全路径，或单击 ，弹出文件对话框，查找扩展名为.mdc 的图形文件。单击  执行。如出错，弹出信息窗口，提示出错行及错误信息。

2.4.2 标记失效位

单击测试界面中 ，弹出文件对话框，选择与图形文件对应的文件，扩展名为 .mdc 文件。执行结果在扩展名为 .mdc 的图形源文件中，对应通道失效位标红，失效索引中，只有第一行失效标记有效。

在测试结果中“测功能”如显示 fall，查看具体数据得返回测试界面点  查看。

2.5 设置 (设置模式)

单击  按钮，弹出如下选择测试程序窗口：



选择测试程序窗口

在此模式下，可以进行测试程序库文件和图形文件的设置。

2.5.1 测试程序选择窗口

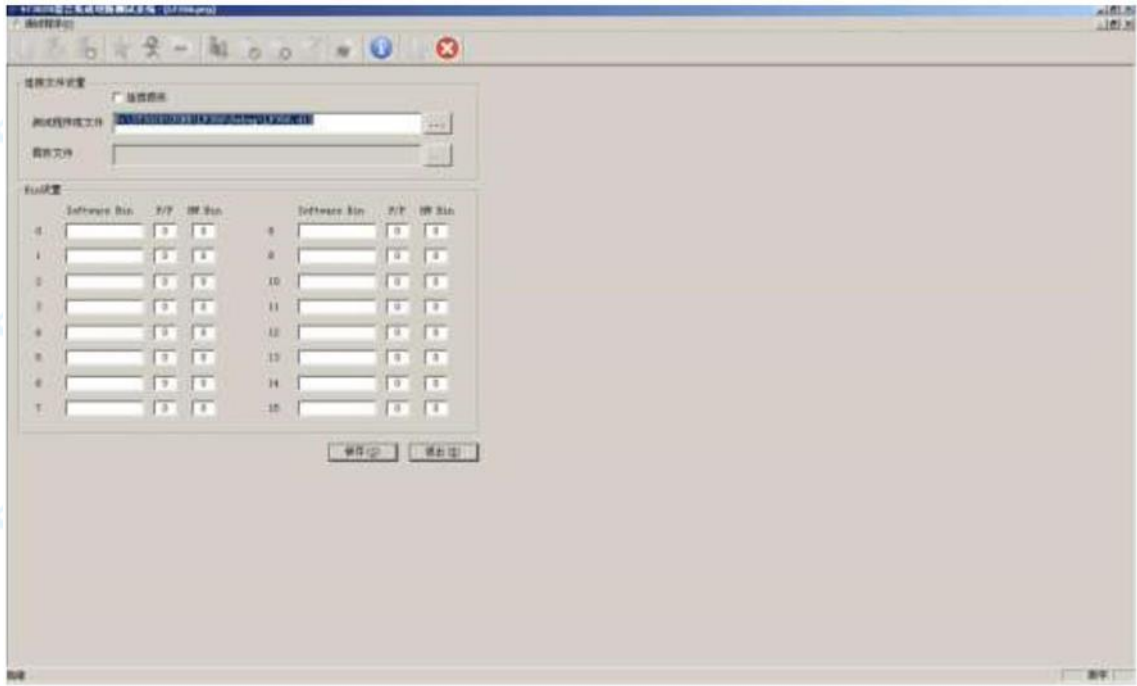
此窗口包括：**新建(N)**、**插入(I)**、**移出(R)**、“程序选择列表框”。

新建(N)：新建测试程序，弹出文件对话框，填写或选择文件名，建立新的测试文件，文件扩展名为 .prg 。进入文档窗口。

插入(I)：向“程序选择列表框”中加入测试程序。单击按钮，显示文件对话框，选择 测试程序，文件扩展名为 .prg 。进入文档窗口。

移出(R)：从“程序选择列表框”中移出测试程序选项，但不删除。选择测试程序：在“程序选择列表框”中选中要用的测试程序，确定或双击，进入设置状态。

2.5.2 主文档窗口



主文档窗口

测试程序库文件：用于连接的测试程序动态库，单击浏览按钮，打开文件对话框，选择库文件。

图形文件：与测试程序动态库对应的图形文件，扩展名为 .mdv，单击浏览，弹出文件对话框，选择图形文件。


Bin 设置：提示统计结果显示时 Bin 的设置。



退出：退出[F10]，自动提示保存文件。

以上工作，完成建立测试程序与系统的连接。Visual C++编辑环境，编写测试程序动态库。

2.6 测试（测试模式）

2.6.1 测试程序选择窗口

单击  按钮，显示测试程序选择窗口，如图 2-4，选择测试程序。此窗口包

括：、、“程序选择列表框”。

插入(I)

: 向“程序选择列表框”中加入测试程序。单击按钮，显示文件选择对话框，选择测试程序，文件扩展名为 .prg 。

插入(I)

: 从“程序选择列表框”中移出测试程序选项，但不删除。选择测试程序：在“程序选择列表框”中选中要用的测试程序，确定或双击，进入测试状态。



测试程序选择窗口

2.6.2 测试主窗口：

选择测试程序后，进入测试主窗口，如下图：



测试主窗口

测试窗口包括：菜单、工具条，用来操作测试过程。功能详细如下：



：新建批号。重新开始新的一批测试，但不退出测试程序时用。

菜单：工具 -> 新批号



：运行测试

菜单：测试 -> 运行测试； 快捷键：[F2]



：check 按钮，失效停止设置，选中为失效停止方式，取消为失效继续

菜单：工具 -> 设置-> 测试方式 -> 失效停止； 快捷键：[F4]



：check 按钮，显示方式设置，统计显示和详细测试显示，选中为统计方式

菜单：显示 -> 批次统计显示； 快捷键：[F5]



：Radio 按钮，选中，显示测试结果 Pass 的数据



：Radio 按钮，选中，显示测试结果 Fail 的数据



：Radio 按钮，选中，显示所有测试结果数据



：打印按钮，单击，打印当前测试的数据结果，如果在统计显示方式下，打印统计表；在测试数据显示方式下，打印全部测试结果



：图形向量失效显示按钮，进行功能项测试时，若失效，单击，可显示向量失效图。

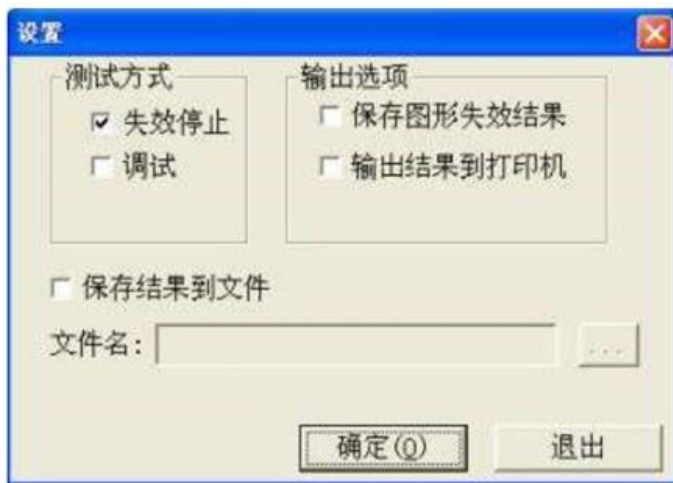


：退出； 快捷键 [F10]

2.6.3 设置测试选项


单击菜单：工具 -> 设置，弹出设置对话框。如下图窗口：

失效停止：设置失效返回，如选中，测试失效时停止测试，否则，失效后继续执行。



设置窗口

保存结果到文件：如保存测试结果数据，在测试前，选中此选项。

按  按钮，弹出保存文件对话框，键入文件名后，按保存按钮，测试结果保存到指定文件中。

输出结果到打印机：选中此选项，在测试过程中，每测试完一被测芯片，自动打印。

保存图形失效结果：选中此选项，当功能测试失效时，自动将测试结果保存到失效文件中，文件名与图形文件名相同，扩展名为 .log 。

2.6.4 其他菜单功能

单击菜单：工具 -> 序列号，显示序列号对话框，如下：在序列号编辑框中填写新的序列号。



序列号对话框

单击菜单：工具 -> 新批号，显示新建批号对话框（图 2-8），在批号编辑框中填写新的批号。如选择初始显示本窗口，每次进入测试主窗口时显示此对话框。



新建批号对话框

2.6.5 统计数据显示：

Bin_No	Bin_Name	Account	Percentage
[0]		0	0.00
[1]		0	0.00
[2]		0	0.00
[3]		0	0.00
[4]		0	0.00
[5]		0	0.00
[6]		0	0.00
[7]		0	0.00
[8]		0	0.00
[9]		0	0.00
[10]		0	0.00
[11]		0	0.00
[12]		0	0.00
[13]		0	0.00
[14]		0	0.00
[15]		0	0.00

统计数据显示窗口

统计显示方式下数据：显示内容为按分箱顺序显示分箱统计数。第一列显示分箱号，第二列显示在此分箱的测试数目，第三列显示分箱数占总数的百分比。

The screenshot shows the 'ST3020 集成电路测试系统 - [测试日志]' window. The status bar at the top indicates: 测试程序 : SN54LS00J.prg, 总数 : 1, 合格率 % : 100.00; 批次编号 : default, 通过总数 : 1; 操作人员 : , 失效总数 : 0. Below this is a table for 'serial: 1 pass Bin: 0'.

序号	名称	测量结果	P/F	单位	下限	上限
1	CON_1	-0.562	P	V	-1.900	-0.100
2	CON_2	-0.456	P	V	-1.900	-0.100
3	CON_3	-0.563	P	V	-1.900	-0.100
4	CON_4	-0.541	P	V	-1.900	-0.100
5	CON_5	-0.408	P	V	-1.900	-0.100
6	CON_6	-0.666	P	V	-1.900	-0.100
7	CON_42	-0.533	P	V	-1.900	-0.100
8	CON_43	-0.549	P	V	-1.900	-0.100
9	CON_44	-0.450	P	V	-1.900	-0.100
10	CON_45	-0.583	P	V	-1.900	-0.100
11	CON_46	-0.467	P	V	-1.900	-0.100
12	CON_47	-0.553	P	V	-1.900	-0.100
13	FUN		P			

测试结果显示图

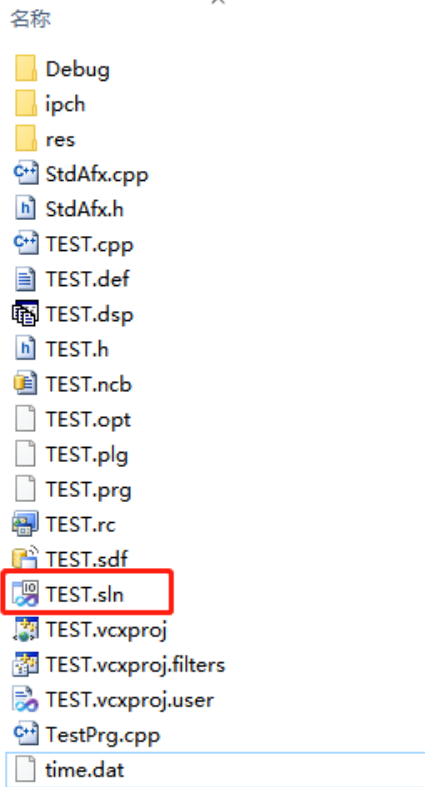
测试程序：当前测试程序名称。批次号：测试批号。总数：测试总数。合格总数：测试通过的总数。失效总数：测试失效总数。合格率：测试通过占测试总数的百分比。指示灯：表示测试状态，黄色表示正在测试，红色表示 FAIL，绿色表示 PASS。测试结果数据显示：详细显示每一项测试结果，黑色字表示 PASS，红色字表示 FAIL。

以上是基于 Visual C++ 6.0 版本的文字操作介绍。下面将 Visual Studio 2010 与 Visual C++ 6.0 不同点进行说明：

2.7 环境版本

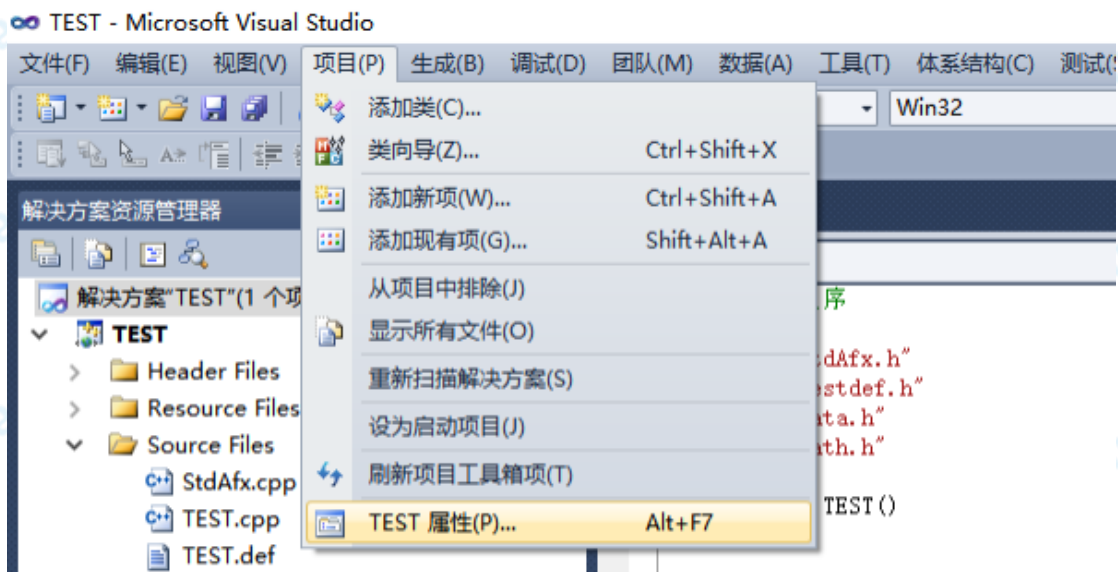
2.7.1 VS2010 的工程文件为后缀名为.sln 文件，如图一。

本地磁盘 (D:) > ST3020 > USER > TEST >

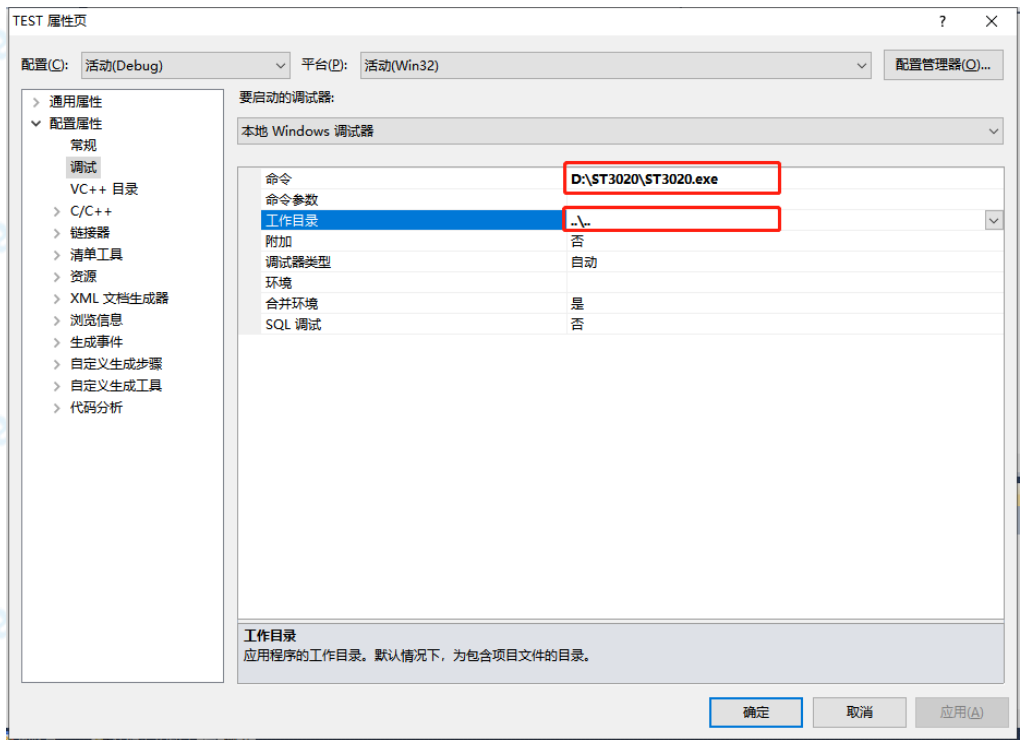


图一：工程文件

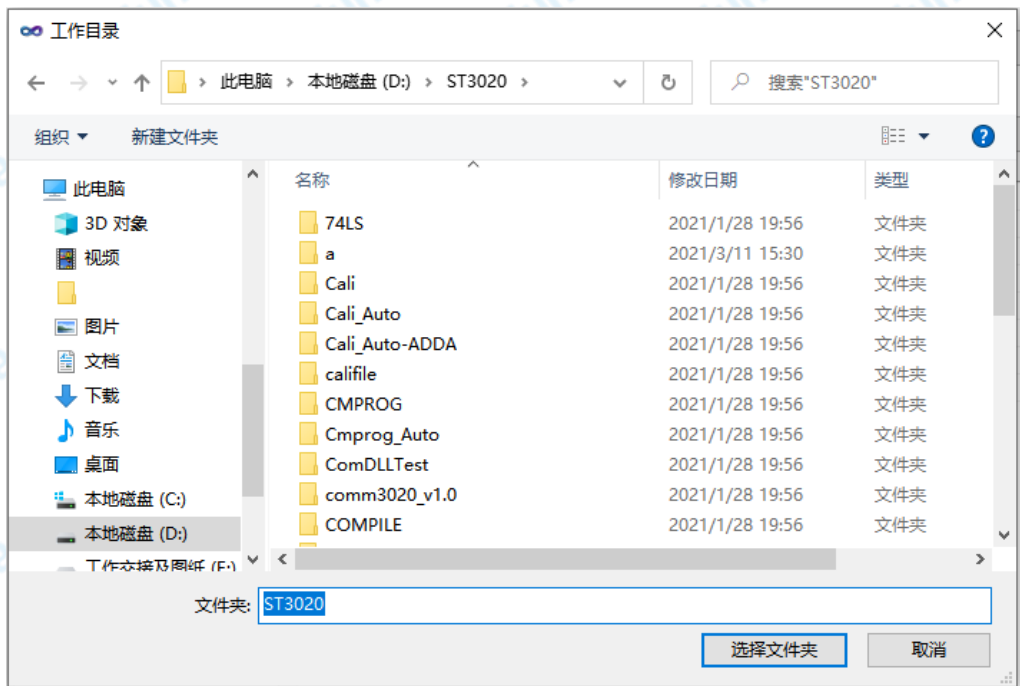
2.7.2 关于程序文件设置。打开“项目--XX 属性”，选中“配置属性--调试”，设置“命令”“工作属性”。“工作属性”的设置路径见图四。



图二

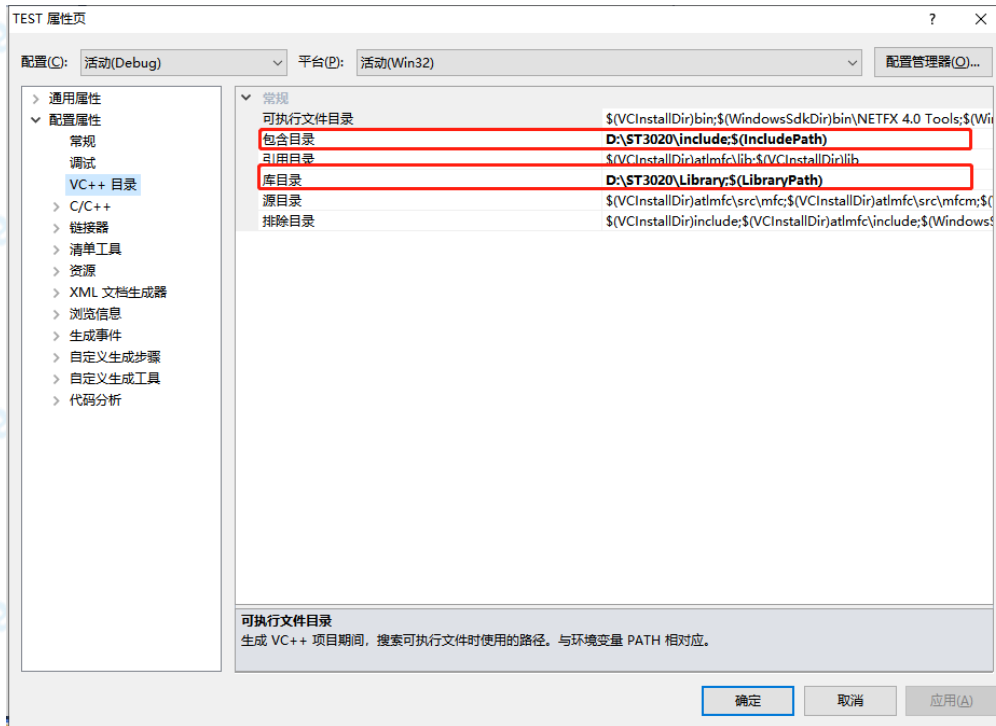


图三：属性界面



图四：工作目录路径设置

2.7.3 关于程序文件设置。打开“项目--XX 属性”，选中“配置属性--VC++目录”，设置“包含目录”“库目录”。见图五。



图五：VC++目录设置

2.7.4 程序的编译，如下图，点击生成选择“重新生成解决方案”。



图六：程序编译

2.8 小结

(1) 系统软件是测试的操作平台，菜单式与 C 语言都可以在 ST3020 系统上测试器件，数模电 C 语言零基础建议先从菜单式学习，C 语言编程灵活度强适用度高，更适合工作场景使用。

(2) isual C++ 6.0 版本与 Visual Studio 2010 因软件工具的区别，操作界面的细微差别，不影响测试程序开发过程及测试结果。

第三章 数字集成电路测试指令

测试程序包括两部分：

➤ 图形文件：用于描述测试图形及图形顺序流向控制。图形文件是给定格式的文本文件，先在任意文本编辑器环境下编辑图形文件的 ASCII 源文件，然后执行 Compliedlg.exe 程序编译源文件，转换成与测试系统硬件相适应的图形文件二进制目标文件。

➤ 测试代码：产生测试程序动态链接库文件。测试程序经过编译产生测试动态链接库文件，导入到测试软件中，点击测试按钮开始测试。

3.1 图形文件指令

序号	指令	解释
1	INC	顺次走一步，执行一次图形
2	RPT,n	重复送该图形 n+1 次，n 最大不能超过 4095
3	LDC,n	为 LOOP 循环指令定义循环次数，标号到 LOOP 指令间的图形循环 n+1 次，最大值不能超过 4095，最多可嵌套装三次 n 值
4	LOOP	如 LDC 装入的 n 值不等于 0：n 减 1，跳转到标号。如 LDC 装入的 n 值等于 0：顺次往下走，LOOP 指令支持 3 层 嵌套。

5	GONP,	从标号开始, 如失效: 跳转到标号, 如不失效, 到下一图形。
6	LDF	动态测量时的标志, 与 JMP 配合使用
7	JMP,	动态测量时, 从标号开始, 到 JMP 语句循环执行。
8	HALT	送该指令后停止, 图形发生器工作结束, 每个起始点 START_INDEX 对应 1 个 HALT
9	LDAR1, n	地址计数器 1 装载, 把”n” 载到内部地址计数器 1 中。
10	LDAR2, n	地址计数器 2 装载, 把”n” 载到内部地址计数器 2 中。
11	ADDR1	把地址计数器 1 的值输出到地管脚中
12	ADDR2	把地址计数器 2 的值输出到地管脚中
13	INCAR1	地址计数器 1 加 1
14	INCAR2	地址计数器 2 加 1
15	DECAR1	地址计数器 1 减 1
16	DECAR2	地址计数器 2 减 1
1-8 常规指令适应大部分器件, 9-16 指令只适应存储器测试。		
注意: 图形指令的第一列必须是空格 标号必须从第一列开始, 不能空格。		

3.2 指令用法

(1) INC, [功能]: 顺次走一步, 执行一次图形

(2) RPT, n, [功能]: 重复送该图形 n+1 次, n 最大不能超过 4095。

[范例]: RPT, 256 (0001 HLLL)

(3) LDC,n, [功能]: 为 LOOP 循环指令定义循环次数, 标号到 LOOP 指令间的图形循环 n+1 次, 最大值不能超过 4095, 最多可嵌套装三次 n 值

(4) LOOP, 标号, [功能]: 如 LDC 装入的 n 值不等于 0: n 减 1, 跳转到标号。如 LDC 装入的 n 值等于 0: 顺次往下走, LOOP 指令支持 3 层嵌套。

[范例]:

```
START_INDEX(0)
LDC, 4095      (0000 LLLL)
A INC         (0001 HLLL)
INC          (0001 HLLL)
LOOP, A      (0001 HLLL)
HALT        (0001 HLLL)
```

(5) GONP, 标号, [功能]: 从标号开始, 如失效: 跳转到标号, 如不失效, 到下一图形。一般 GONP 指令用于输出状态的匹配, 例如一个没有复位端的计数器上电后初始状态不确定时钟计数器输出

[范例]:

```
A INC      (0 XXXX)
GONP,A    (1 LLLL)
INC (0    LLLL)
INC (1    LLLH)
INC (0    LLLH)
INC (1    LLHL)
```

(6) LDF, [功能]: 动态测量时的标志, 与 JMP 配合使用

(7) JMP, 标号, [功能]: 动态测量时, 从标号开始, 到 JMP 语句循环执行。

[范例]:

```
START_INDEX(1)
LDF (0000 LLLL)
A INC (0001 HLLL)
INC (0001 HLLL)
JMP, A (0001 HLLL)
```

HALT (0001 HLLL)

(8) HALT, [功能]: 送该指令后停止图形发生器工作结束, 每个起始 START_INDEX 对应 1 个 HALT

[范例]: START_INDEX(1)

```
INC    (0001 HLLL)
HALT   (0001 HLLL)
```

(9) LDAR1, n, [功能]: 地址计数器 1 装载,把“n”装载到内部地址计数器 1 中。

(10) LDAR2, n, [功能]: 地址计数器 2 装载,把“n”装载到 内部地址计数器 2 中。

(11) ADDR1, [功能]: 把地址计数器 1 的值输出到地址管脚。

(12) ADDR2, [功能]: 把地址计数器 2 的值输出到地址管脚。

(13) INCAR1, [功能]: 地址计数器 1 加 1。

(14) INCAR2, [功能]: 地址计数器 2 加 1。

(15) DECAR1, [功能]: 地址计数器 1 减 1

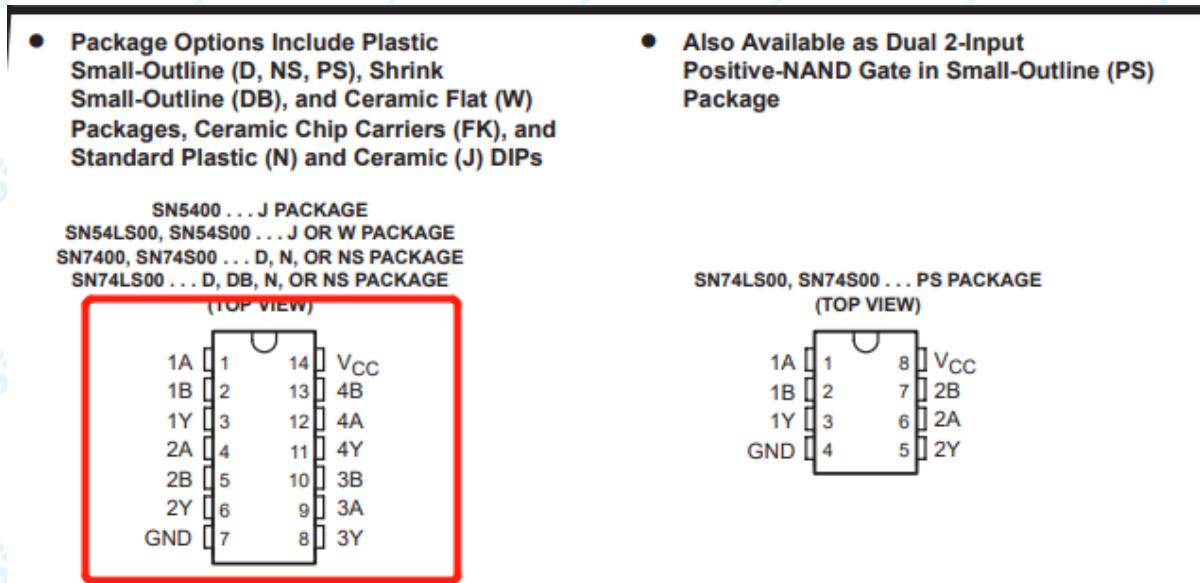
(16) DECAR2, [功能]: 地址计数器 2 减 1

3.3 小结

图形文件指令集是数字器件测试必须用到的, 图形的编写结合管脚定义、真值表、时序图等数据手册中的信息来设计图形文件。指令集中的各指令的解释与用法结合范例学习。器件测试先编写图形再编程序。

3.4 编写图形范例 74LS00

举例器件 74LS00 说明图形文件的编写



3.4.1 图形文件编写步骤:

- (1) 标准图形文件以.mdc 为扩展名, 文件的第一行为“MEM_SOURCE_15; ”。
- (2) 定义管脚是图形文件的一部分, 以“PINDEF”为开始标记, 定义管脚及管脚组定义管脚与通道对应关系 以“PIN_TO_CHANNEL”为开始标记, 定义管脚与通道的对应关系。
- (3) 编辑图形指令及数据 以“MAIN_F”为开始标记, 编辑测试图形的流程及图形
- (4) 结束标志 以“END.”为开始标记, 结束图形文件的编辑
- (5) 图形文件编译转换图形文件编辑完成后, 在 CompileDlg.exe 程序环境下, 填写或浏览 (查找) 要编辑的文件 -> 编译 -> 执行编译转换。转换成与源文件相同名称, 扩展名为 .mdv 的二进制文件

3.4.2 图形文件编示例

MEM_SOURCE_15; “第一行必照写”标识性“语句

PINDEF: ----- 管脚定义开始。

A1 =I,BIN,(1)

B1 =I,BIN,(2)

Y1 =O,BIN,(3)

A2 =I,BIN,(4)

B2 =I,BIN,(5)

Y2 =O,BIN,(6)

A3 =I,BIN,(43)

B3 =I,BIN,(44)

Y3 =O,BIN,(42)

A4 =I,BIN,(46)

B4 =I,BIN,(47)

Y4 =O,BIN,(45)

管脚与通道对应关系语句由两部分组成，第一部分是管脚号，第二部分是通道号，用“=”号分开，可以有两种格式，如范例为芯片引脚与通道连接对应关系。

通道连接 芯片引脚			芯片引脚 通道连接				
			U1				
CHB1	PIN1	1	1A	VCC	14	PIN14	DPS1
CHB2	PIN2	2	1B	4B	13	PIN13	CHB47
CHB3	PIN3	3	1Y	4A	12	PIN12	CHB46
CHB4	PIN4	4	2A	4Y	11	PIN11	CHB45
CHB5	PIN5	5	2B	3B	10	PIN10	CHB44
CHB6	PIN6	6	2Y	3A	9	PIN9	CHB43
AGND	PIN7	7	GND	3Y	8	PIN8	CHB42
			74LS00				

3.4.3 管脚释疑

<管脚名称> = <I|O|IO>,<BIN >,(通道号)

----- 用二进制方式编写图形的管脚定义

<管脚组名称> (数值..数值) = <I|O|IO>,<HEX >,(通道号)

----- 用十六进制方式编写图形的管脚定义

I: 表示为输入管脚

O: 表示为输出管脚

IO: 表示为输入/输出管脚

如启用菜单编程 74LS00，则管脚定义如下：

PIN_TO_CHANNEL

1..6=1..6

8..13=42..47

3 . 4 . 4 图形文件数据格式

BIN: 代表二进制方式， HEX: 代表十六进制方式。二进制方式：0 表示输入为低电平， 1 表示输入为高电平 ， L 表示输出为低电平， H 表示输出为高电平。十六进制方式：一位数据代表 4 个管脚， 如果前面有 T， 表示后面数据为输出。 X: 表示对应的通道不驱动， 不比较。

图形文件可以用二进制也可以用十六进制， 或两种在同一图形里出现均可， 在管脚定义时确定用哪种方式。

编写图形文件， 参考 datesheet 真值表： （用二进制举例）

MAIN_F-----“第一行必照写”标识性“语句

START_INDEX(0) { FUN }-----{ } 或 // 是注释， 自选

```

      {IIO IIO IIO IIO} ←
      {12 12 12 12} ←
      {AAA BBB CCC DDD} ←
INC   (00X 00X 00X 00X) ←
INC   (00H 00H 00H 00H) ←
INC   (01H 01H 01H 01H) ←
INC   (10H 10H 10H 10H) ←
INC   (11L 11L 11L 11L) ←
HALT  (11X 11X 11X 11X) ←
START_INDEX(1){ VOL IIL IOS} ←
INC   (11X 11X 11X 11X) ←
INC   (11X 11X 11X 11X) ←
INC   (11X 11X 11X 11X) ←
HALT  (11X 11X 11X 11X) ←
START_INDEX(2){ VOH IIH} ←
INC   (00X 00X 00X 00X) ←
INC   (00X 00X 00X 00X) ←
INC   (00X 00X 00X 00X) ←
HALT  (00X 00X 00X 00X) ←
END. ←
←
←
←
←

```

```

A1 =I.BIN,(1) ←
B1 =I.BIN,(2) ←
Y1 =C.BIN,(3) ←
A2 =I.BIN,(4) ←
B2 =I.BIN,(5) ←
Y2 =C.BIN,(6) ←
A3 =I.BIN,(43) ←
B3 =I.BIN,(44) ←
Y3 =C.BIN,(42) ←
A4 =I.BIN,(46) ←
B4 =I.BIN,(47) ←
Y4 =C.BIN,(45) ←

```

FUNCTION TABLE
(each gate)

INPUTS		OUTPUT
A	B	Y
H	H	L
L	X	H
X	L	H

3.4.5 图形文件格式

源文件：ASCII 文件 .mdc 文件。图形指令由多个指令段组成，每段由 START_INDEX () 开始，HALT (图形) 结束。图形指令有四种格式。（例举）

指令 (图形)----- INC (0001 HLLL)

标号 指令 (图形) ----- A INC (0001 HLLL)

指令, 参数 (图形) ----- LDC, 4095 (0000 LLLL)

标号 指令, 参数 (图形) A LDC, 4095 (0000 LLLL)

注：无标号时第 1 列必须为空格。标号必须从第 1 列开始。指令与图形之间、参数与图形之间必须空格

3.4.6 数据段标记

START_INDEX (起始点标号) ----- 必须从第一列开始，起始点标号可为 0，至 47 间的一个数字。起始点标号用于在执行图形时，标明执行哪一段图形。在 VisualC++中

编写测试程序时，在调用函数 RUN_PATTERN()时,第 1 个参数就是起始点标号,与此定义相同，相互对应使用。

3.4.7 图形结束

END. ----- 必须从第 1 列开始，标志图形文件结束。

3.4.8 注释

{ } ----- 注释一行

// ----- 注释一行或在一条指令的结尾注释

3.5 程序函数目录

序号	函数	解释
1	PMU_CONDITIONS()	设置 PMU 测量条件
2	PMU_MEASURE()	PMU 测量
3	MAT_PMU_MEASURE	连续测量
4	SET_DPS()	设置 DPS 条件
5	DPS_MEASURE()	DPS 测量
6	DPS_OFF()	关闭 DPS
7	FORMAT()	格式化波形
8	SET_INPUT_LEVEL()	设置输入参考电平
9	SET_OUTPUT_LEVEL()	设置输出参考电平
10	RUN_PATTERN()	执行一段图形
11	SET_PERIOD()	设置时钟周期
12	SET_TIMING()	设置时钟
13	SET_RELAY()	闭合指定继电器，其他继电器打开

14	CLOSE_RELAY()	闭合指定继电器，其他保持原状态
15	CLEAR_RELAY()	清继电器
16	BIN ()	设置分箱号
17	SHOW_RESULT()	显示测试结果到显示设备
18	SHOW_RESULT()	显示测试结果到显示设备
19	SHOW_RESULT()	显示测试结果到显示设备
20	CLEAR_ALL()	总清
21	SET_PMU()	设置 PMU
22	CLEAR_PMU()	清 PMU
23	Delay ()	延时 (单位: ms)

3.5.1 函数用法

(1) PMU_CONDITIONS(),

[函数原形]: void PMU_CONDITIONS (unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit);

[功能]: 设置 PMU 的测试条件，在使用精密测量单元 PMU 测量前先设置好，选用不同的模式，决定 PMU 测量方法。箱位值作为限流或限压的保护值，并且根据箱位值自动选择测量量程。

[参数说明]

Mode : 测量模式，有两种选择：

FVMI ----- 加压测流

FIMV ----- 加流测压

注意：默认电流流入被测器件的方向为正，流出为负。

Value: 施加值, 范围: 电压: $\pm 15V$, 电流: $\pm 300mA$,

Value_Unit: 施加单位, FVMI 方式, 单位可以选 V 或 MV,

FIMV 方式, 单位可以选单位可以选 MA 或 UA。

Clamp_Value: 箝位值范围: 电压: $0 \sim 15V$, 电流: $0 \sim 300mA$

Clamp_Unit: 箝位单位, FVMI 方式,

电流箝位, 单位可以选 MA 或 UA,

FIMV 方式, 电压箝位, 单位可以选 V 或 MV

设置 PMU 测量模式为加压测流方式, 施加电压 $5V$, 箝位电流 $10mA$ 。

[范例]: 74LS00 连接性测试指令用 PMU 的应用:

```
//CON-TEST
```

```
SET_DPS(1,0,V,20,MA);
```

```
PMU_CONDITIONS(FIMV,-0.1,MA,2,V);
```

```
if(!PMU_MEASURE("1-6,42-47",20,"CON_",V,-0.1,-1.9))
```

```
BIN(1);
```

注: 括号里的参数、上下限其目的是检测器件内部二级管是否导通。

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

连接性测试就是为了检测管脚与测试道通的连接

PARAMETER	TEST CONDITIONS‡	SN5400			SN7400			UNIT
		MIN	TYP§	MAX	MIN	TYP§	MAX	
V _{IK}	V _{CC} = MIN, I _I = -12 mA			-1.5			-1.5	V
V _{OH}	V _{CC} = MIN, V _{IL} = 0.8 V, I _{OH} = -0.4 mA	2.4	3.4		2.4	3.4		V
V _{OL}	V _{CC} = MIN, V _{IH} = 2 V, I _{OL} = 16 mA		0.2	0.4		0.2	0.4	V
I _I	V _{CC} = MAX, V _I = 5.5 V			1			1	mA
I _{IH}	V _{CC} = MAX, V _I = 2.4 V			40			40	µA
I _{IL}	V _{CC} = MAX, V _I = 0.4 V			-1.6			-1.6	mA
I _{OS} ¶	V _{CC} = MAX	-20		-55	-18		-55	mA
I _{CCH}	V _{CC} = MAX, V _I = 0 V		4	8		4	8	mA
I _{CCL}	V _{CC} = MAX, V _I = 4.5 V		12	22		12	22	mA

‡ For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

§ All typical values are at V_{CC} = 5 V, T_A = 25°C.

¶ Not more than one output should be shorted at a time.

输入输出管脚必须用PMU测

(2) PMU_MEASURE (),

[函数原形]: `BOOL PMU_MEASURE (CString csPin, unsigned int tDelay, CString csItem, CString csUnit, double fUpLimit, double fDnLimit);`
`double PMU_MEASURE (unsigned int pin, unsigned int tDelay);`

[功能]: 使用 PMU 测量直流参数。第一种类型的函数测量直流参数，比较上下限，返回 PASS, FAIL 值。第二种函数类型直接测量单一通道，返回测量值

[参数说明]: csPin: 通道号, 范围: 1 ~ 128, tDelay: 测量延迟时间, 单位毫秒

csItem: 测量项目名称, csUnit: 测量单位

fUpLimit : 测量结果上限值, 如无上限值, 填写 No_UpLimit。

fDnLimit : 测量结果下限值, 如无下限值, 填写 No_LoLimit。

pin : 通道号, 范围: 1 ~128

[返回值]: 返回失效值 1: Pass 0: Fail 实际测量值

[范例]: 74LS00 连接性测试指令用 PMU_MEASURE ()的应用:

```
//CON-TEST
SET_DPS(1,0,V,20,MA);
PMU_CONDITIONS(FIMV,-0.1,MA,2,V);
if(!PMU_MEASURE("1-6,42-47",20,"CON_",V,-0.1,-1.9))
    BIN(1);
```

注意: 默认电流流入被测器件的方向为正, 流出为负。

PARAMETER	TEST CONDITION [†]	SN54LS00			SN74LS00			UNIT
		MIN	TYP [‡]	MAX	MIN	TYP [‡]	MAX	
V _{IK}	V _{CC} = MIN, I _I = -18 mA			-1.5			-1.5	V

```
//VIK-TEST
SET_DPS(1,4.5,V,50,MA);
PMU_CONDITIONS(FIMV,-18,MA,2,V);
if(!PMU_MEASURE("1,2,4,5,43,44,46,47",5,"VIK_",V,No_UpLimit,-1.5))
    BIN(3);
```

其他器件如需要测量后有返回值, 则用: `double Result` 指令

PMU_CONDITIONS (FIMV, 5, V, 10, mA);

Result = PMU_MEASURE (3,10);

注：在使用此函数前，必须先设 PMU_CONDITIONS 函数

(3) MAT_PMU_MEASURE (),

[函数原形]: void MAT_PMU_MEASURE(int pin,unsigned int tDelay, int ipNumber,
int fFreq,double *fValue)

[参数说明]:

pin: 通道号

tDelay: 测量延迟时间, 单位: 毫秒 ms

ipNumber: 采样点数 (输入范围: 1~2048)

fFreq: 采样频率 (实际采样频率: 50MHz/fFreq)

*fValue: 测量返回值。指针型。程序中需定义数组, 数组个数应大于采样点数。

(4) SET_DPS (),

[函数原形]: void SET_DPS (BYTE No, double Vdd, unsigned int Vdd_Unit,double
Iclamp,unsigned int Iclamp_Unit)

[功能]: 设置 DPS 的测量条件。

[参数说明]: No: 选择 DPS 的通路, DPS1-----第一路 DPS

DPS2-----第二路 DPS, Vdd: 设置 DPS 的电压值, 范围: ±15V

Vdd_Unit : 单位可为 V 或 MV, Iclamp : 电流箝位值, 范围: 250mA

Iclamp_Unit : 电流箝位单位, 可为 MA 或 UA

[范例]: 74LS00 参数 VIK 测试, 指令用 SET_DPS 的应用

```
//VIK-TEST
```

```
SET_DPS(1,4.75,V,50,MA);
```

```
PMU_CONDITIONS(FIMV,-18,MA,2,V);
```

```
if(!PMU_MEASURE("1,2,4,5,43,44,46,47",5,"VIK_",V,No_UpLimit,-1.5))
```

BIN(3);

		SN54LS00			SN74LS00			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V _{CC}	Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V _{IH}	High-level input voltage	2			2			V
V _{IL}	Low-level input voltage				0.7			V
I _{OH}	High-level output current				-0.4			mA
I _{OL}	Low-level output current				4			mA
T _A	Operating free-air temperature	-55		125	0		70	°C

NOTE 4: All unused inputs of the device must be held at V_{CC} or GND to ensure proper device operation. Refer to the TI application report, *Implications of Slow or Floating CMOS Inputs*, literature number SCBA004.

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITION†	SN54LS00			SN74LS00			UNIT
		MIN	TYP‡	MAX	MIN	TYP‡	MAX	
V _{IK}	V _{CC} = MIN, I _I = -18 mA				-1.5			V

(5) DPS_MEASURE ()

[函数原形] `BOOL DPS_MEASURE (BYTE No, BYTE IRange, unsigned int Delays, CString csItem, CString csUnit, double fIccMax, Double fIccMin);`
`Double DPS_MEASURE (BYTE No, BYTE IRange, unsigned int Delays);`

[功能]: 使用 DPS 直接测量电流值, 比较上、下限是否失效, 显示测试结果到显示设备。第二种形式用于直接返回被测电流值

[参数说明]: No : 测量通道选择。DPS1-----第一路, DPS2-----第二路

IRange : 测流量程, R2UA----- 2uA 档, R20UA ----- 20uA 档

R200UA----- 200uA 档, R2MA ----- 2mA 档

R20MA ----- 20mA 档, R200MA ----- 250mA 档

Delays : 测量延迟时间, 单位毫秒, CsItem : 测试项目名称

CsUnit : 测量值单位,

fIccMax : 测量值上限, 如无上限, 填写 No_UpLimit。

fIccMin : 测量值下限, 如无下限, 填写 No_LoLimit。

[返回值]。失效返回: 1: Pass, 0: Fail, 测量电流值

[范例]: 74LS00 参数 ICC 测试, 指令用 DPS_MEASURE ()

```
//ICCH-TEST
```

```
SET_DPS(1,5.25,V,10,MA);
```

```
SET_INPUT_LEVEL(4.5,0);
```

```
RUN_PATTERN(2,1,0,0);
```

```
if(!DPS_MEASURE(1,R20MA,5,"ICCH",MA,1.6,No_LoLimit))
```

```
    BIN(8);
```

```
//ICCL-TEST
```

```
SET_DPS(1,5.25,V,10,MA);
```

```
SET_INPUT_LEVEL(4.5,0);
```

```
SET_OUTPUT_LEVEL(2.5,2.5);
```

```
RUN_PATTERN(1,1,0,0);
```

```
if(!DPS_MEASURE(1,R20MA,5,"ICCL",MA,4.4,No_LoLimit))
```

```
    BIN(8);
```

PARAMETER	TEST CONDITIONST	SN54LS00		SN74LS00		UNIT		
		MIN	TYP‡	MAX	MIN		TYP‡	MAX
V _{IK}	V _{CC} = MIN, I _I = -18 mA			-1.5		-1.5	V	
V _{OH}	V _{CC} = MIN, V _{IL} = MAX, I _{OH} = -0.4 mA	2.5	3.4		2.7	3.4	V	
V _{OL}	V _{CC} = MIN, V _{IH} = 2 V	I _{OL} = 4 mA		0.25	0.4	0.25	0.4	
		I _{OL} = 8mA				0.35	0.5	
I _I	V _{CC} = MAX, V _I = 7 V			0.1		0.1	mA	
I _{IH}	V _{CC} = MAX, V _I = 2.7V			20		20	µA	
I _{IL}	V _{CC} = MAX, V _I = 0.4 V			-0.4		-0.4	mA	
I _{OS} §	V _{CC} = MAX	-20		-100	-20		-100	mA
ICCH	V _{CC} = MAX, V _I = 0 V		0.8	1.6		0.8	1.6	mA
ICCL	V _{CC} = MAX, V _I = 4.5 V		2.4	4.4		2.4	4.4	mA

其他器件如需要测量后有返回值, 则用: double Result 指令

```
Result = DPS_MEASURE (DPS1, R20MA, 5);
```

(6) DPS_OFF ()

[函数原形]: void DPS_OFF (BYTE No)

[功能]: 关闭 DPS, DPS 单元与测试板断开。

[参数说明]: No : DPS 选择

DPS1 ----- 第一路, DPS2 ----- 第二路

[范例] DPS 有关闭功能, 根据需要选用就可将第二路 DPS 关闭

DPS_OFF (DPS2)

(7) FORMAT ()

格式化通道, 在驱动图形方式时, 选用此格式驱动通道。当前设置只影响设置的通道, 不改变其他通道的格式化方式, 可选择四种格式, 具体波形请参照图形说明。

[函数原形]: Void FORMAT (BYTE fmt, Cstring csChannel)

[参数说明]

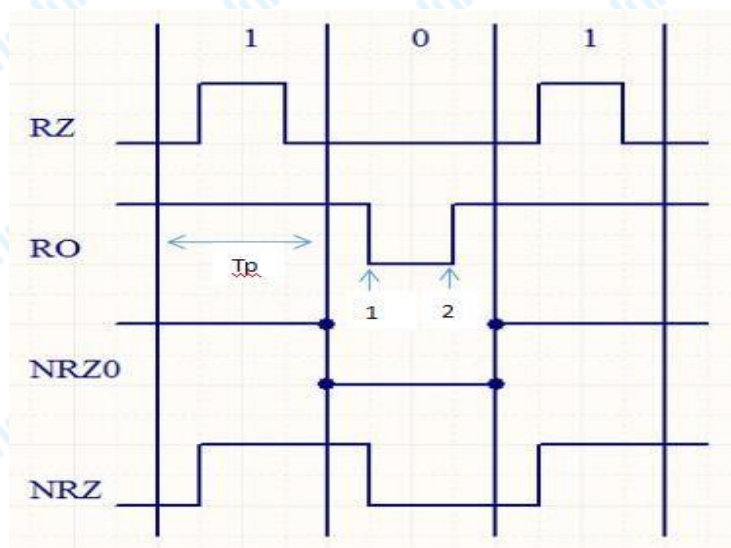
fmt : 格式化模式

NRZ ----- 非归零

RO----- 归一

RZ----- 归零

NRZ0 -- 非归零



CsChannel: 通道号数据, 是一串包含整数、", -"分隔符的字符串, ST3020 最多 128 个通路, 通道编号 1-128, 1 表前沿 2 表后沿 3 是选通点 Tp 表周期

[范例] 74LS00 功能测试, 输入管脚波形格式用 NRZ0

//Function

SET_DPS(1,4.75,V,50,MA);

SET_INPUT_LEVEL(2,0.8);

```

SET_OUTPUT_LEVEL(2.7,0.4);
SET_PERIOD(2000);           {周期, 根据需要自行设置值}
SET_TIMING(200,1400,1700); {前沿-后沿-选通}
FORMAT(NRZ0,"1,2,4,5,44,43,47,46"); {输入管脚用 NRZ0}
if(!RUN_PATTERN("FUN_MIN",0,1,0,0))
    BIN(2);

```

根据器件输入管脚的需要, 可选择适合的波形格式。

(8) SET_INPUT_LEVEL(): 设置全部管脚输入驱动电平值。

[函数原形]: void SET_INPUT_LEVEL (double Vih, double Vil)

[参数说明]: Vih: 输入高电平, 单位 V, 范围: 0V ~ +15V

Vil: 输入低电平, 单位 V, 范围: 0V ~ +5V

[范例] 74LS00 功能测试, 输入高低电平设置, 用指令 SET_INPUT_LEVEL();

	SN54LS00			SN74LS00			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
VCC Supply voltage	4.5	5	5.5	4.75	5	5.25	V
V _{IH} High-level input voltage	2			2			V
V _{IL} Low-level input voltage	0.7			0.8			V
I _{OH} High-level output current	-0.4			-0.4			mA
I _{OL} Low-level output current	4			8			mA
T _A Operating free-air temperature	-55		125	0		70	°C

//Function

```

SET_DPS(1,4.75,V,50,MA);
SET_INPUT_LEVEL(2,0.8);
SET_OUTPUT_LEVEL(2.7,0.4);
SET_PERIOD(2000);
SET_TIMING(200,1400,1700);
FORMAT(NRZ0,"1,2,4,5,44,43,47,46");
if(!RUN_PATTERN("FUN_MIN",0,1,0,0))
    BIN(2);

```

(9) SET_OUTPUT_LEVEL(): 设置所有管脚输出比较电平。

[函数原形]: void SET_OUTPUT_LEVEL (double Voh, double Vol)

[参数说明]: Voh: 输出高电平, 单位 V, 范围: 0V ~ +15V

Vol: 输出低电平, 单位 V, 范围: 0V ~ +5V

[范例] 74LS00 功能测试, 输出比较电平设置, 用指令 SET_OUTPUT_LEVEL()

PARAMETER	TEST CONDITIONST	SN54LS00			SN74LS00			UNIT		
		MIN	TYP‡	MAX	MIN	TYP‡	MAX			
V _{IK}	V _{CC} = MIN, I _I = -18 mA			-1.5			-1.5	V		
V _{OH}	V _{CC} = MIN, V _{IL} = MAX, I _{OH} = -0.4 mA	2.5	3.4		2.7	3.4		V		
V _{OL}	V _{CC} = MIN, V _{IH} = 2 V	I _{OL} = 4 mA			0.25	0.4		0.25	0.4	V
		I _{OL} = 8 mA						0.35	0.5	

//Function

```
SET_DPS(1,4.75,V,50,MA);
```

```
SET_INPUT_LEVEL(2,0.8);
```

```
SET_OUTPUT_LEVEL(2.7,0.4);
```

```
SET_PERIOD(2000);
```

```
SET_TIMING(200,1400,1700);
```

```
FORMAT(NRZ0,"1,2,4,5,44,43,47,46");
```

```
if(!RUN_PATTERN("FUN_MIN",0,1,0,0))
```

```
    BIN(2);
```

(10) RUN_PATTERN(): 运行图形, 返回 PASS 或 FAIL。

[函数原形]: BOOL RUN_PATTERN (int start_idx, int get_fail, int apgen, int time_range)BOOL RUN_PATTERN (CString csItem ,int start_idx,int get_fail,int apgen, int time_range)

第一种格式: 只运行图形, 不在显示设备上显示

第二种格式: 用于测试功能, 测试完成后, 显示测试结果

[参数说明]: csItem : 测试项目名称, 用于功能测量时使用

start_idx: 运行图形的索引号, 指定执行哪一段图形

get_fail: 测试模式选择

0 或 GO----- 失效后继续执

1 或 NOGO ----- 失效返回

apgen : 设置 apgen 测量方式用于测试存储器时将通道板转换为地址板使用。

0 或 NOAPG ----- 不设置 apgen 方式

1 或 APGEN1----- 设置 1-8 通道板为 apgen 方式

2 或 APGEN2----- 设置 1-16 通道板为 apgen 方式

3 或 APGEN3 ----- 设置 1-24 通道板为 apgen 方式

time_range : 设置图形运行的时间长度, 单位 ms , 超时自动退出, 返回测试失效。写 0 表示默认时间为 30 秒。

[返回值]: 返回失效值 0: 对应通道 FAIL, 1: 对应通道 PASS

[范例] 74LS00 功能测试, 运行图形, 用指令 RUN_PATTERN()

//Function

```
SET_DPS(1,4.75,V,50,MA);
SET_INPUT_LEVEL(2,0.8);
SET_OUTPUT_LEVEL(2.7,0.4);
SET_PERIOD(2000);
SET_TIMING(200,1400,1700);
FORMAT(NRZ0,"1,2,4,5,44,43,47,46");
if(!RUN_PATTERN("FUN_MIN",0,1,0,0))
    BIN(2);
```

RUN_PATTERN (0, 1, 0, 0): "0"是执行图形索引号为 0, "1"指失效返回, "0"不
"不使用 APGEN, 最后一个"0"是指不设运行时间范围, 默认时间为 30 秒。执行功能测
量, 测试项名称"Function", 索引号为 0, 失效返回, 不使用

RUN_PATTERN ("Function" 0, 1, 0, 0), 在运行图形前, 必须先设时钟周期, 前
沿, 后沿, 选通, 否则不能运行。AGPEN 方式, 不设运行时间范围

(11) SET_PERIOD (): 设置一个测试周期时间长度。

[函数原形]: void SET_PERIOD (unsigned int period)

[参数说明]: period : 时钟周期, 单位 ns, 范围: 大于 50ns ,小于 10ms

[范例] 74LS00 功能测试, 设置周期, 用指令 SET_PERIOD()

```
//Function
```

```
SET_DPS(1,4.75,V,50,MA);
```

```
SET_INPUT_LEVEL(2,0.8);
```

```
SET_OUTPUT_LEVEL(2.7,0.4);
```

```
SET_PERIOD(2000);           {周期, 根据需要自行设置值}
```

```
SET_TIMING(200,1400,1700); {前沿-后沿-选通}
```

```
FORMAT(NRZ0,"1,2,4,5,44,43,47,46"); {输入管脚用 NRZ0}
```

```
if(!RUN_PATTERN("FUN_MIN",0,1,0,0))
```

```
    BIN(2);
```

(12) SET_TIMING (): 设置波形格式时间。

[函数原形]: void SET_TIMING (double LeadEdge,double EndEdge,double Ctg)

[参数说明]:

LeadEdge: 前沿, 设置波形格式开始时间, 单位 ns, 范围: 大于 10ns

EndEgde: 后沿, 设置波形的后沿, 单位 ns, 范围: 大于 20 ns

Ctg: 选通, 设置每个周期的 (选通) 时间点对器件的输出状态进行比较, 与图形的预期状态相同为合格, 不同则为失效。单位 ns, 范围: 大于 10ns

[范例] 74LS00 功能测试, 设置波形格式、输出比较时间, 用指令 SET_TIMING

//Function

SET_DPS(1,4.75,V,50,MA);

SET_INPUT_LEVEL(2,0.8);

SET_OUTPUT_LEVEL(2.7,0.4);

SET_PERIOD(2000); {周期, 根据需要自行设置值}

SET_TIMING(200,1400,1700); **{前沿-后沿-选通}**

FORMAT(NRZ0,"1,2,4,5,44,43,47,46"); {输入管脚用 NRZ0}

if(!RUN_PATTERN("FUN_MIN",0,1,0,0))

 BIN(2);

(13) SET_RELAY(): 闭合指定继电器, 其他继电器打开

[函数原形]: void SET_RELAY (CString sRelay)

[参数说明]: sRelay: 继电器号, 为一串整数字符或一串由“, -”连接的整数范围, 中间不能有空格。

[范例]根据器件需要选择是否用这条指令

设置继电器 1, 3, 5, 6, 7, 8 合上, 其他继电器打开

SET_RELAY (“1, 3, 5, 6, 7, 8”); 或 SET_RELAY (“1, 3, 5-8”);

(14) CLOSE_RELAY(): 闭合指定继电器, 其他继电器保持原状态

[函数原形]: Void CLOSE_RELAY (CString sRelay)

[参数说明]: sRelay: 继电器号, 为一串整数字符或一串由“, -”连接的整数范围, 中间不能有空格。

[范例]: 根据器件需要选择是否用这条指令

闭合继电器 1, 3, 4, 5, 8, 9, 10, 再闭合 12

```
SET_RELAY ( "1, 3-5, 8-9");
```

```
CLOSE_RELAY("12");
```

(15) CLEAR_RELAY(): 打开指定继电器

[函数原形]: Void CLEAR_RELAY(CStringsRelay)

[参数说明]: sRelay: 继电器号, 为一串整数字符或一串由“, -”连接的整数范围, 中间不能有空格。

[范例]: 根据器件需要选择是否用这条指令

设置 2, 3, 7, 8, 9 继电器后, 打开 3 继电器

```
SET_RELAY ( "2, 3, 7-9");
```

```
CLEAR_RELAY ("3");
```

(16) BIN () : 用于测试完成后的分箱处理。

[函数原形]: Void BIN (BYTE bin)

[参数说明] : bin : 分箱号

[范例]: 如果 “con” 测量失效, 设置分箱号为 1。判断器件是否合格

(17) SHOW_RESULT(): 显示测试项目及测试结果

[函数原形]: void SHOW_RESULT (CString csItem, double measure_value, int csUnit, double fUpLimit, double fDnLimit)

显示测试结果。

(18) SHOW_RESULT()

[函数原形]: void SHOW_RESULT (CString csItem, double measure_value, CString csUnit, double fUpLimit, Double fDnLimit)

单位为标准单位 A, MA, V, MV, 可定义非标准单位。

(19) SHOW_RESULT(CString csItem BOOL bTest)

显示功能测量结果,

[参数说明]: csItem: 测试项名称, measure_value: 测量值

csUnit: 单位: 使用函数 16、17, 测量结果自动转换单位显示; 使用函数 18, 直接显示测量值和单位, 单位可为任意字符串, 如“Ω”

fUpLimit: 上限, fDnLimit: 下限, bTest: FAIL 或 0 显示测试项失效, PASS 或 1 显示测试项合格

(20) CLEAR_ALL()总清系统.

(21) SET_PMU()

[函数原形]: void SET_PMU(unsigned int Mode, double Value, unsigned int Value_Unit, doubleClamp_Value, unsigned int Clamp_Unit)

[参数说明]: Mode: 选择模式, 有两种选择:

FVMI ----- 加压测流, FIMV ----- 加流测压

Value: 施加值, 范围: 电压: ±15V,

电流: ±250mA, Value_Unit: 施加值单位

FVMI 方式: 单位可以选 V 或 MV

FIMV 方式: 单位可以选 UA 或 MA

Clamp_Value: 箝位值, 范围: 测压范围: 0~15V

测流范围: 0~250mA, Clamp_Unit: 箝位单位:

FVMI 方式: 单位可以选 UA 或 MA

FIMV 方式: 单位可以选 V 或 MV

(22) CLEAR_PMU(): 清 PMU

(23) Delay(): 设置延迟时间 (单位: ms)

[函数原形]: void Delay(double fMs)

[参数说明]: fMs: 延迟时间, 单位: ms, double 型

[范例]: Delay(0.1); 延迟 100us。

3.6 小结

22 条函数通过理解函数原型领会测量原理, 结合系统硬件资源与环境理解运行, 通过事例应用, 例如如何调用图形生成测试向量等方式进行测试。

第四章 被测器件数据手册

略


不同器件不同方法取不同信息。要读通器件功能，工作方式，性能特点，写图形先定义管脚、看真值表、时序图。根据器件类别与特点，结合系统手册的指令写出不遗漏全覆盖的图形文件。

写程序时则细看数据手册中的“最大”“高小”“典型值”的指标，函数的应用根据指标参数定义调用图形，写出完整的程序文件。

具体每个器件不同，数据手册“datasheet”中所列出信息不同，对英文阅读能力有一定要求。但专业的关键词汇也基本固定，所以多读细读手册，是了解测试信息的重要来源。

第五章 数字器件测试程序开发

5.1 [实例一] 54LS00

(1) 下载  SN54LS00数据手册.pdf，通读并理解器件性能与工作**原理**，根据 ST3020 测试系统资源写测试方案，包括但不限于 IC 测试原理，ST3020 系统测试方法，解读数据手册参数指标、图形文件、测试程序等内容。

(2) 根据 ST3020 测试系统接口资源与数据手册中的管脚电路设计 PCB 做接口板 (适配器)，根据封装工艺采购芯片卡座并完成焊接。

(3) 图形文件编写：

由器件逻辑功能编写图形文件，此器件是根据数据手册中的真值表编写，如：

MEM_SOURCE_15;

PINDEF

A1 =I,BIN,(1)

B1 =I,BIN,(2)

Y1 =O,BIN,(3)

A2 =I,BIN,(4)

B2 =I,BIN,(5)

Y2 =O,BIN,(6)

A3 =I,BIN,(43)

B3 =I,BIN,(44)

Y3 =O,BIN,(42)

A4 =I,BIN,(46)

B4 =I,BIN,(47)

Y4 =O,BIN,(45)

MAIN_F

START_INDEX(0) { FUN } (这段图形测功能，不遗漏全覆盖)

{ IIO IIO IIO IIO }

{ 12 12 12 12 }

{ AAA BBB CCC DDD }

INC (00X 00X 00X 00X)

INC (00H 00H 00H 00H)

```

INC      (01H 01H 01H 01H)
INC      (10H 10H 10H 10H)
INC      (00H 00H 00H 01H)
INC      (00H 00H 00H 10H)
INC      (00H 00H 00H 11L)
INC      (00H 00H 01H 00H)
INC      (00H 00H 10H 00H)
INC      (00H 00H 11L 00H)
INC      (00H 01H 00H 00H)
INC      (00H 10H 00H 00H)
INC      (00H 11L 00H 00H)
INC      (01H 00H 00H 00H)
INC      (11L 00H 00H 00H)
INC      (11L 00H 00H 00H)
INC      (11L 11L 11L 11L)
HALT     (11X 11X 11X 11X)

START_INDEX(1){ VOL  IIL  IOS}  这段图形测“VOL  IIL  IOS“这三个参数

INC      (11X 11X 11X 11X)
INC      (11X 11X 11X 11X)
INC      (11X 11X 11X 11X)
HALT     (11X 11X 11X 11X)

START_INDEX(2) { VOH  IIH }  这段图形测“VOH  IIH “这三个参数

INC      (00X 00X 00X 00X)
INC      (00X 00X 00X 00X)
INC      (00X 00X 00X 00X)
HALT     (00X 00X 00X 00X)

END.

```

说明：图形文件编写根据对与非门理解及测试时效，不遗漏全覆盖就可。

此代码仅供参考非标答。

(4) 测试程序

```

#include "StdAfx.h"
#include "testdef.h"
#include "data.h"
#include "math.h"

void PASCAL SN54LS00J()  新建文件时系统自动产生的头文件

```

接下来进行连接性测试（开短路测试）

```

{
//CON-TEST
SET_DPS(1,0,V,20,MA);
PMU_CONDITIONS(FIMV,-0.1,MA,2,V);
if(!PMU_MEASURE("1-6,42-47",20,"CON_",V,-0.1,-1.9))
BIN(1);

//Function          这段功能测试（调用图形文件）
SET_DPS(1,4.5,V,50,MA);
SET_INPUT_LEVEL(2,0.7);
SET_OUTPUT_LEVEL(2.5,0.4);
SET_PERIOD(2000);
SET_TIMING(200,1400,1700);
FORMAT(NRZ0,"1,2,4,5,44,43,47,46");
if(!RUN_PATTERN("FUN_MIN",0,1,0,0))
BIN(2);

```

```

SET_DPS(1,5,V,50,MA);
SET_INPUT_LEVEL(2,0.7);
SET_OUTPUT_LEVEL(2.5,0.4);
SET_PERIOD(2000);
SET_TIMING(200,1400,1700);
FORMAT(NRZ0,"1,2,4,5,44,43,47,46");
if(!RUN_PATTERN("FUN",0,1,0,0))
BIN(2);

```

```

SET_DPS(1,5.5,V,50,MA);
SET_INPUT_LEVEL(2,0.7);
SET_OUTPUT_LEVEL(2.5,0.4);
SET_PERIOD(2000);
SET_TIMING(200,1400,1700);
FORMAT(NRZ0,"1,2,4,5,44,43,47,46");
if(!RUN_PATTERN("FUN_MAX",0,1,0,0))
BIN(2);

```

说明：代码编写根据对与非门的理解及测试时效，不遗漏全覆盖就可。

此代码仅供参考非标答。

```
//VIK-TEST          此段开始测直流参数
```

```
SET_DPS(1,4.5,V,50,MA);
```

```

PMU_CONDITIONS(FIMV,-18,MA,2,V);    此处设置条件“加流测压方式”
if(!PMU_MEASURE("1,2,4,5,43,44,46,47",5,"VIK_",V,No_UpLimit,-1.5)) 进行测量
    BIN(3);

//VOH-TEST
SET_DPS(1,4.5,V,50,MA);
SET_INPUT_LEVEL(4.5,0);
SET_OUTPUT_LEVEL(2.5,2.5);

RUN_PATTERN(2,1,0,0);    使输出为高电平

PMU_CONDITIONS(FIMV,-0.4,MA,5,V);
if(!PMU_MEASURE("3,6,42,45",5,"VOH_",V,No_UpLimit,2.5))
    BIN(4);

//VOL-TEST
SET_DPS(1,4.5,V,50,MA);

RUN_PATTERN(1,1,0,0);    使输出为低电平

PMU_CONDITIONS(FIMV,4,MA,0.6,V);
if(!PMU_MEASURE("3,6,42,45",5,"VOL_",V,0.4,No_LoLimit))
    BIN(5);

//II-TEST
SET_DPS(1,5.5,V,50,MA);
RUN_PATTERN(2,1,0,0);
PMU_CONDITIONS(FVMI,7,V,0.19,MA);
if(!PMU_MEASURE("1,2,4,5,43,44,46,47",5,"II_",MA,0.1,No_LoLimit))
    BIN(6);

//IIH-TEST
RUN_PATTERN(2,1,0,0);
PMU_CONDITIONS(FVMI,2.7,V,50,UA);
if(!PMU_MEASURE("1,2,4,5,43,44,46,47",5,"IIH_",UA,20,No_LoLimit))
    BIN(6);

//IIL-TEST
RUN_PATTERN(1,1,0,0);
PMU_CONDITIONS(FVMI,0.4,V,1,MA);
if(!PMU_MEASURE("1,2,4,5,43,44,46,47",5,"IIL_",MA,No_UpLimit,-0.4))
    BIN(6);

//IOS-TEST

```

```

SET_DPS(1,5.5,V,150,MA);
RUN_PATTERN(2,1,0,0);
    PMU_CONDITIONS(FVMI,0,V,130,MA);
if(!PMU_MEASURE("3,6,42,45",8,"IOS_",MA,-20,-100))
    BIN(7);

//ICCH-TEST
SET_DPS(1,5.5,V,10,MA);
SET_INPUT_LEVEL(4.5,0);
RUN_PATTERN(2,1,0,0);
if(!DPS_MEASURE(1,R20MA,5,"ICCH",MA,1.6,No_LoLimit))
    BIN(8);

//ICCL-TEST
SET_DPS(1,5.5,V,19,MA);
SET_INPUT_LEVEL(4.5,0);
    SET_OUTPUT_LEVEL(2.5,2.5);
RUN_PATTERN(1,1,0,0);
if(!DPS_MEASURE(1,R20MA,5,"ICCL",MA,4.4,No_LoLimit))
    BIN(8);

```

7. 测试结果

器件名称: SN54LS00J

操作员 :

批号 Lot: default

2022/10/11 16:17

器件号: 1 合格

序号	项目名称	测量值	P/F	单位	下限	上限
1	CON_1	-0.561164		V	-1.900	-0.100
2	CON_2	-0.447325		V	-1.900	-0.100
3	CON_3	-0.642739		V	-1.900	-0.100
4	CON_4	-0.458891		V	-1.900	-0.100
5	CON_5	-0.500287		V	-1.900	-0.100
6	CON_6	-0.644565		V	-1.900	-0.100
7	CON_42	-0.529508		V	-1.900	-0.100
8	CON_43	-0.539857		V	-1.900	-0.100
9	CON_44	-0.414451		V	-1.900	-0.100
10	CON_45	-0.656132		V	-1.900	-0.100
11	CON_46	-0.440628		V	-1.900	-0.100
12	CON_47	-0.542292		V	-1.900	-0.100

	FUN	PASS			
13	FUN				
14	VOH_3	2.958175	V	2.500	
15	VOH_6	3.046408	V	2.500	
16	VOH_42	3.102390	V	2.500	
17	VOH_45	3.027545	V	2.500	
18	VOL_3	0.274404	V		0.400
19	VOL_6	0.193121	V		0.400
20	VOL_42	0.188105	V		0.400
21	VOL_45	0.309092	V		0.400
22	VIK_1	-0.904509	V	-1.500	
23	VIK_2	-0.942862	V	-1.500	
24	VIK_4	-0.923381	V	-1.500	
25	VIK_5	-1.011652	V	-1.500	
26	VIK_43	-1.179672	V	-1.500	
27	VIK_44	-0.998260	V	-1.500	
28	VIK_46	-0.912423	V	-1.500	
29	VIK_47	-1.025045	V	-1.500	
30	II_1	0.001821	mA		0.100
31	II_2	0.001841	mA		0.100
32	II_4	0.001787	mA		0.100
33	II_5	0.001767	mA		0.100
34	II_43	0.001773	mA		0.100
35	II_44	0.001800	mA		0.100
36	II_46	0.001787	mA		0.100
37	II_47	0.001807	mA		0.100
38	IIH_1	1.665142	uA		20.000
39	IIH_2	1.692174	uA		20.000
40	IIH_4	1.678658	uA		20.000
41	IIH_5	1.665142	uA		20.000
42	IIH_43	1.665142	uA		20.000
43	IIH_44	1.651626	uA		20.000
44	IIH_46	1.712447	uA		20.000
45	IIH_47	1.705690	uA		20.000
46	III_1	-0.215457	mA	-0.400	
47	III_2	-0.195050	mA	-0.400	
48	III_4	-0.218430	mA	-0.400	
49	III_5	-0.200321	mA	-0.400	
50	III_43	-0.218159	mA	-0.400	
51	III_44	-0.201469	mA	-0.400	
52	III_46	-0.218024	mA	-0.400	
53	III_47	-0.199848	mA	-0.400	
54	IOS_3	-34.348547	mA	-100.000	-20.000
55	IOS_6	-40.268177	mA	-100.000	-20.000
56	IOS_42	-43.100472	mA	-100.000	-20.000


57	IOS_45	-35.543212	mA	-100.000	-20.000
58	ICCH	0.959948	mA		1.600
59	ICCL	3.272217	mA		4.400

器件号: 2 合格

序号	项目名称	测量值	P/F	单位	下限	上限
1	CON_1	-0.458282		V	-1.900	-0.100
2	CON_2	-0.535596		V	-1.900	-0.100
3	CON_3	-0.557512		V	-1.900	-0.100
4	CON_4	-0.466805		V	-1.900	-0.100
5	CON_5	-0.538640		V	-1.900	-0.100
6	CON_6	-0.533770		V	-1.900	-0.100
7	CON_42	-0.681091		V	-1.900	-0.100
8	CON_43	-0.418713		V	-1.900	-0.100
9	CON_44	-0.505158		V	-1.900	-0.100
10	CON_45	-0.541075		V	-1.900	-0.100
11	CON_46	-0.533770		V	-1.900	-0.100
12	CON_47	-0.464979		V	-1.900	-0.100
13	FUN	PASS				
14	VOH_3	3.098131		V	2.500	
15	VOH_6	3.003813		V	2.500	
16	VOH_42	2.959392		V	2.500	
17	VOH_45	2.989817		V	2.500	
18	VOL_3	0.191549		V		0.400
19	VOL_6	0.294649		V		0.400
20	VOL_42	0.338886		V		0.400
21	VOL_45	0.282925		V		0.400
22	VIK_1	-0.995216		V	-1.500	
23	VIK_2	-1.092619		V	-1.500	
24	VIK_4	-0.907553		V	-1.500	
25	VIK_5	-0.884420		V	-1.500	
26	VIK_43	-1.090183		V	-1.500	
27	VIK_44	-1.110882		V	-1.500	
28	VIK_46	-1.062180		V	-1.500	
29	VIK_47	-1.020175		V	-1.500	
30	II_1	0.001821		mA		0.100
31	II_2	0.001834		mA		0.100
32	II_4	0.001780		mA		0.100
33	II_5	0.001861		mA		0.100
34	II_43	0.001760		mA		0.100
35	II_44	0.001760		mA		0.100
36	II_46	0.001780		mA		0.100

37	II_47	0.001760	mA		0.100
38	IIH_1	1.698932	uA		20.000
39	IIH_2	1.725963	uA		20.000
40	IIH_4	1.678658	uA		20.000
41	IIH_5	1.658384	uA		20.000
42	IIH_43	1.698932	uA		20.000
43	IIH_44	1.658384	uA		20.000
44	IIH_46	1.685416	uA		20.000
45	IIH_47	1.692174	uA		20.000
46	IIIL_1	-0.203159	mA	-0.400	
47	IIIL_2	-0.207956	mA	-0.400	
48	IIIL_4	-0.205388	mA	-0.400	
49	IIIL_5	-0.212821	mA	-0.400	
50	IIIL_43	-0.206267	mA	-0.400	
51	IIIL_44	-0.213700	mA	-0.400	
52	IIIL_46	-0.205186	mA	-0.400	
53	IIIL_47	-0.212619	mA	-0.400	
54	IOS_3	-35.221055	mA	-100.000	-20.000
55	IOS_6	-39.878905	mA	-100.000	-20.000
56	IOS_42	-42.201118	mA	-100.000	-20.000
57	IOS_45	-37.664077	mA	-100.000	-20.000
58	ICCH	0.958591	mA		1.600
59	ICCL	3.274931	mA		4.400

5.2 [实例二] 6264 存储器

(1) 下载  UT6264B_datasheet.pdf

通读并理解器件性能与工作原理。

根据 ST3020 测试系统资源写测试方

案，如右图：

注意：datasheet 中所有参数、指

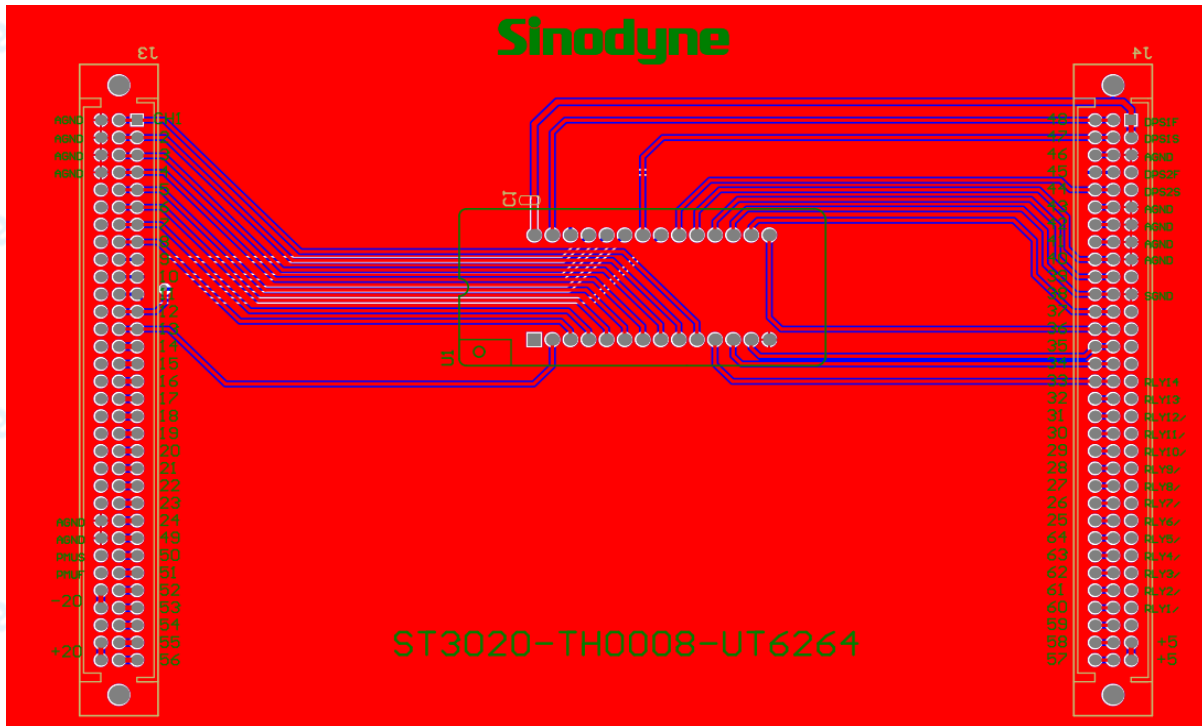
标，不遗漏全覆盖。

(2) 根据测试方案设计接口板

(适配器) 画 PCB 制板、采购卡座

进行焊接，如下图：

1.2 项目设计情况
▲ 1.3 IC测试及设备测试原理
1.3.1 IC测试原理
1.3.2 ST3020系统测试原理
▲ 1.4 设计难点
1.4.1 “走步”算法的实现
1.4.2 自制适配器
▲ 2. 芯片概述
2.1 芯片介绍
2.2 引脚描述
▲ 3. 功能测试
3.1 功能测试算法概述
3.2 功能测试方法：
3.3 管脚定义及其通道的对应
▲ 3.4 功能测试



注意：管脚定义与测试程序中的代码对应

(3) 图形文件

定义管脚：

MEM_SOURCE_15;

PINDEF

A0	= I,	BIN, (1)
A1	= I,	BIN, (2)
A2	= I,	BIN, (3)
A3	= I,	BIN, (4)
A4	= I,	BIN, (5)
A5	= I,	BIN, (6)
A6	= I,	BIN, (7)
A7	= I,	BIN, (8)
A8	= I,	BIN, (9)
A9	= I,	BIN, (10)
A10	= I,	BIN, (11)
A11	= I,	BIN, (12)
A12	= I,	BIN, (13)

I/O0 = IO, BIN, (33)

I/O1 = IO, BIN, (34)

I/O2 = IO, BIN, (35)

I/O3 = IO, BIN, (36)
I/O4 = IO, BIN, (37)
I/O5 = IO, BIN, (38)
I/O6 = IO, BIN, (39)
I/O7 = IO, BIN, (40)

CE1 = I, BIN, (44)
CE2 = I, BIN, (45)
OE = I, BIN, (47)
WE = I, BIN, (48)

(4)用走步法写图形

MAIN_F

START_INDEX(1) VOH-TEST

```
{AAAAAAAAAAAAA IIIIII CCOW}  
{0123456789111 01234567 EEEE}  
{ 012 12 }  
INC (000000000000 XXXXXXXX 1111)  
INC (000000000000 XXXXXXXX 1111)  
INC (000000000000 11111111 1111)  
INC (000000000000 11111111 0110)//write  
INC (000000000000 11111111 0110)  
INC (000000000000 XXXXXXXX 1111)  
INC (000000000000 HHHHHHHH 0101)//read  
INC (000000000000 HHHHHHHH 0101)  
INC (000000000000 HHHHHHHH 0101)  
HALT (000000000000 HHHHHHHH 0101)
```

START_INDEX(2) VOL-TEST

```
INC (000000000000 XXXXXXXX 1111)  
INC (000000000000 XXXXXXXX 1111)  
INC (000000000000 00000000 1111)  
INC (000000000000 00000000 0110)  
INC (000000000000 00000000 0110)  
INC (000000000000 XXXXXXXX 1111)  
INC (000000000000 LLLLLLLL 0101)  
INC (000000000000 LLLLLLLL 0101)  
INC (000000000000 LLLLLLLL 0101)  
HALT (000000000000 LLLLLLLL 0101)
```

START_INDEX(3) IIL-TEST

```
{AAAAAAAAAAAAA IIIIII CCOW}  
{0123456789111 01234567 EEEE}  
{ 012 12 }  
INC (000000000000 XXXXXXXX 1111)
```

```

INC      (111111111111 11111111 1111)
INC      (111111111111 11111111 1111)
INC      (111111111111 11111111 1111)
HALT     (111111111111 11111111 1111)
START_INDEX(4) IIL-TEST
INC      (000000000000 00000000 0000)
INC      (000000000000 00000000 0000)
INC      (000000000000 00000000 0000)
INC      (000000000000 00000000 0000)
HALT     (000000000000 00000000 0000)
START_INDEX(5) IOZ-TEST
INC      (000000000000 XXXXXXXX 1010)
INC      (000000000000 XXXXXXXX 1010)
INC      (000000000000 XXXXXXXX 1010)
INC      (000000000000 XXXXXXXX 1010)
HALT     (000000000000 XXXXXXXX 1010)
START_INDEX(6) ICC
          {AAAAAAAAAAAAA IIIIII CCOW}
          {0123456789111 01234567 EEEE}
          {          012          12  }

INC      (000000000000 XXXXXXXX 0111)
INC      (000000000000 XXXXXXXX 0111)
LDAR1,0 (000000000000 XXXXXXXX 0111)
ADDR1   (000000000000 XXXXXXXX 0111)
LDF     (000000000000 XXXXXXXX 0111)
C1  INC (000000000000 10111111 0110)//write
      JMP,C1 (000000000000 11111111 0110)//read
      INC (000000000000 XXXXXXXX 0111)
      HALT (000000000000 XXXXXXXX 0111)

START_INDEX(19) FUN-TEST
          {AAAAAAAAAAAAA IIIIII CCOW}
          {0123456789111 01234567 EEEE}
          {          012          12  }

INC      (000000000000 XXXXXXXX 1111)
INC      (000000000000 XXXXXXXX 1111)

INC      (000000000000 XXXXXXXX 1111)
LDAR1,0 (000000000000 XXXXXXXX 1111)
ADDR1   (000000000000 XXXXXXXX 1111)
LDC,1   (000000000000 XXXXXXXX 1111)
E2  INC (000000000000 XXXXXXXX 1111)

```

```

LDC,4095 (000000000000 11111111 1111) // WRITE-FF
E1 INC (000000000000 11111111 0110)
  INCAR1 (000000000000 11111111 1111)
  INC (000000000000 11111111 0110)
  LOOP, E1 (000000000000 11111111 1111)
  INC (000000000000 XXXXXXXX 1111)
  LOOP, E2 (000000000000 XXXXXXXX 1111)

  LDAR1,0 (000000000000 XXXXXXXX 1111)
  LDC,1 (000000000000 XXXXXXXX 1111)
E4 INC (000000000000 XXXXXXXX 1111)
  LDC,4095 (000000000000 XXXXXXXX 0111) // READ-FF
E3 INC (000000000000 HHHHHHHH 0101)
  INCAR1 (000000000000 XXXXXXXX 0111)
  INC (000000000000 HHHHHHHH 0101)
  LOOP, E3 (000000000000 XXXXXXXX 0111)
  INC (000000000000 XXXXXXXX 1111)
  LOOP, E4 (000000000000 XXXXXXXX 1111)

  LDAR1,0 (000000000000 XXXXXXXX 1111)
  ADDR1 (000000000000 XXXXXXXX 1111)
  LDC,1 (000000000000 XXXXXXXX 1111)
E10 INC (000000000000 XXXXXXXX 1111)
  LDC,4095 (000000000000 XXXXXXXX 1111)
E9 INC (000000000000 XXXXXXXX 0111)
  INC (000000000000 00000000 0110)//write 0 addr 00
  INC (000000000000 LLLLLLLL 0101)//read L
  LDC,4094 (000000000000 XXXXXXXX 0111)
E7 INC (000000000000 XXXXXXXX 0111)
  INCAR1 (000000000000 XXXXXXXX 0111)
  INC (000000000000 HHHHHHHH 0101) // READ H
  LOOP, E7 (000000000000 XXXXXXXX 0111)
  LDC,4094 (000000000000 XXXXXXXX 0111)
E8 INC (000000000000 XXXXXXXX 0111)
  INCAR1 (000000000000 XXXXXXXX 0111)
  INC (000000000000 HHHHHHHH 0101) // READ H
  LOOP, E8 (000000000000 XXXXXXXX 0111)
  INC (000000000000 XXXXXXXX 1111)//8190+1 8191
  INCAR1 (000000000000 XXXXXXXX 0111)//8191+1 8192
  INC (000000000000 HHHHHHHH 0101) // READ H
  INCAR1 (000000000000 XXXXXXXX 0111)//8193
  INC (000000000000 11111111 0110)//write 1 addr0
  INCAR1 (000000000000 XXXXXXXX 0111) //next

```

```

INC      (000000000000 XXXXXXXX 0111)
LOOP, E9 (000000000000 XXXXXXXX 1111)
INC      (000000000000 XXXXXXXX 0111)
LOOP, E10 (000000000000 XXXXXXXX 0111)
HALT     (000000000000 XXXXXXXX 1111)

```

START_INDEX(20) FUN-TEST

```

{AAAAAAAAAAAAA IIIIII CCOW}
{0123456789111 01234567 EEEE}
{          012          12  }

INC      (000000000000 XXXXXXXX 1111)
INC      (000000000000 XXXXXXXX 1111)

INC      (000000000000 XXXXXXXX 1111)
LDAR1,0 (000000000000 XXXXXXXX 1111)
ADDR1    (000000000000 XXXXXXXX 1111)
LDC,1    (000000000000 XXXXXXXX 1111)
F2 INC    (000000000000 XXXXXXXX 1111)
LDC,4095 (000000000000 00000000 1111) // WRITE-00
F1 INC    (000000000000 00000000 0110)
INCAR1   (000000000000 00000000 1111)
INC      (000000000000 00000000 0110)
LOOP, F1 (000000000000 00000000 1111)
INC      (000000000000 XXXXXXXX 1111)
LOOP, F2 (000000000000 XXXXXXXX 1111)

LDAR1,0 (000000000000 XXXXXXXX 1111)
ADDR1    (000000000000 XXXXXXXX 1111)
LDC,1    (000000000000 XXXXXXXX 1111)
F4 INC    (000000000000 XXXXXXXX 1111)
LDC,4095 (000000000000 XXXXXXXX 0111) // READ-00
F3 INC    (000000000000 LLLLLLLL 0101)
INCAR1   (000000000000 XXXXXXXX 0111)
INC      (000000000000 LLLLLLLL 0101)
LOOP, F3 (000000000000 XXXXXXXX 0111)
INC      (000000000000 XXXXXXXX 1111)
LOOP, F4 (000000000000 XXXXXXXX 1111)

LDAR1,0 (000000000000 XXXXXXXX 1111)
ADDR1    (000000000000 XXXXXXXX 1111)
LDC,1    (000000000000 XXXXXXXX 1111)
F10 INC   (000000000000 XXXXXXXX 1111)
LDC,4095 (000000000000 XXXXXXXX 1111)

```

```

F9  INC      (000000000000 XXXXXXXXX 0111)
    INC      (000000000000 11111111 0110)//write 1   addr  00
    INC      (000000000000 11111111 0111)
    INC      (000000000000 HHHHHHHH 0101)//read H
    LDC,4094 (000000000000 XXXXXXXXX 0111)
F7  INC      (000000000000 XXXXXXXXX 0111)
    INCAR1   (000000000000 XXXXXXXXX 0111)
    INC      (000000000000 LLLLLLLL 0101) // READ L
    LOOP, F7 (000000000000 XXXXXXXXX 0111)
    LDC,4094 (000000000000 XXXXXXXXX 0111)
F8  INC      (000000000000 XXXXXXXXX 0111)
    INCAR1   (000000000000 XXXXXXXXX 0111)
    INC      (000000000000 LLLLLLLL 0101) // READ L
    LOOP, F8 (000000000000 XXXXXXXXX 0111)
    INC      (000000000000 XXXXXXXXX 1111)//8190+1 8191
    INCAR1   (000000000000 XXXXXXXXX 0111)//8191+1 8192
    INC      (000000000000 LLLLLLLL 0101) // READ L
    INCAR1   (000000000000 XXXXXXXXX 0111)//8193
    INC      (000000000000 00000000 0110)//write 0   addr0
    INCAR1   (000000000000 XXXXXXXXX 0111) //next
    INC      (000000000000 XXXXXXXXX 0111)
    LOOP, F9 (000000000000 XXXXXXXXX 1111)
    INC      (000000000000 XXXXXXXXX 0111)
    LOOP, F10 (000000000000 XXXXXXXXX 0111)
    HALT     (000000000000 XXXXXXXXX 1111)

```

START_INDEX(10) IOZ-TEST

```

    INC      (000000000000 XXXXXXXXX 1111)
    INC      (000000000000 XXXXXXXXX 1111)
    INC      (000000000000 XXXXXXXXX 1111)
    INC      (000000000000 XXXXXXXXX 1111)
    HALT     (000000000000 XXXXXXXXX 1111)

```

END.

(1) 测试程序的编写

```
//UT6264_LL 测试程序
```

```
#include "StdAfx.h"
```

```
#include "testdef.h"
```

```
#include "data.h"
```

```
#include "math.h"
```

```
void PASCAL UT6264_LL()
```

```
{
```

```
/******
```

```
    本测试程序，样片 UT6264BPC-70LL
```

```
    分箱说明:
```

```
    连接测试 (CON--TEST) -----BIN (1)
```

```
    功能测试 (FUN--TEST) -----BIN (2)
```

```
    输出电压 (VOH--TEST) -----BIN (3)
```

```
    输出电压 (VOL--TEST) -----BIN (4)
```

```
    输入电流 (ILI--TEST) -----BIN (5)
```

```
    输出电流 (IOZ--TEST) -----BIN (6)
```

```
    电源电流 (ICC--TEST) -----BIN (7)
```

```
    静态功耗 (ISB--TEST) -----BIN (8)
```

```
    静态功耗 (ISB2-TEST) -----BIN (9)
```

```
*****/
```

```
    long iTimeStart,iTimeEnd;
```

```
    iTimeStart = GetTickCount();
```

```
    //连接性测试
```

```
        SET_DPS(1,0,V,30,MA);
```

```
PMU_CONDITIONS(FIMV,-0.1,MA,2,V);  
if(!PMU_MEASURE("1-13,33-40,44,45,47,48",5,"CON",V,-0.1,-1))  
    BIN(1);
```

```
//功能测试
```

```
CLEAR_ALL();  
SET_DPS(1,4.5,V,100,MA);  
Delay(10);  
SET_INPUT_LEVEL(2.2,0.8);  
SET_OUTPUT_LEVEL(2.4,0.4);  
SET_PERIOD(500);  
SET_TIMING(100,350,450);
```

```
FORMAT(NRZ0,"1-13,33-40,44,45");//A0-A12,I/O1-I/O8,CE
```

```
FORMAT(RO,"48");//WE
```

```
FORMAT(NRZ0,"47");//OE
```

```
if(!RUN_PATTERN("FUN_FF",19,1,2,200000))
```

```
    BIN(2);
```

```
if(!RUN_PATTERN("FUN_00",20,1,2,200000))
```

```
    BIN(2);
```

```
SET_DPS(1,5.5,V,100,MA);
```

```
Delay(10);
```

```
SET_INPUT_LEVEL(2.2,0.8);
```

```
SET_OUTPUT_LEVEL(2.4,0.4);
```

```
SET_PERIOD(500);
```

```
SET_TIMING(100,350,450);
```

```
FORMAT(NRZ0,"1-13,33-40,44,45");//A0-A12,I/O1-I/O8,CE
```

```
FORMAT(RO,"48");//WE
```

```
FORMAT(NRZ0,"47");//OE
```

```
if(!RUN_PATTERN("FUN_FF_MAX",19,1,2,200000))
```

```
    BIN(2);
```

```
if(!RUN_PATTERN("FUN_00_MAX",20,1,2,200000))
```

```
    BIN(2);
```

```
// VOH 测试
```

```
/*
```

```
测试条件: IOH=-1mA
```

```
*/
```

```
CLEAR_ALL();
```

```
SET_DPS(1,4.5,V,100,MA);
```

```
SET_INPUT_LEVEL(4,0);
```

```
SET_OUTPUT_LEVEL(2.4,0.4);
```

```
    SET_PERIOD(800);
```

```
    SET_TIMING(250,480,600);
```

```
FORMAT(NRZ0,"1-13,33-40,44,45");
```

```
FORMAT(RO,"48");
```

```
FORMAT(NRZ0,"47");
```

```
RUN_PATTERN(1,1,0,0);
```

```
PMU_CONDITIONS(FIMV,-1,MA,5,V);
```

```
if(!PMU_MEASURE("33-40",5,"VOH_",V,No_UpLimit,2.4))
```

```
    BIN(3);
```

```
// VOL 测试
```

```
/*
```

测试条件: IOH=4mA

*/

SET_DPS(1,4.5,V,100,MA);

SET_INPUT_LEVEL(4,0);

SET_OUTPUT_LEVEL(2.4,0.4);

SET_PERIOD(800);

SET_TIMING(250,480,600);

FORMAT(NRZ0,"1-13,33-40,44,45");

FORMAT(RO,"48");

FORMAT(NRZ0,"47");

RUN_PATTERN(2,1,0,0);

PMU_CONDITIONS(FIMV,4,MA,1,V);

if(!PMU_MEASURE("33-40",5,"VOL_",V,0.4,No_LoLimit))

 BIN(4);

// ILI 测试

/*

测试条件: $VSS \leq VIN \leq VCC$

*/

SET_DPS(1,5.5,V,100,MA);

SET_INPUT_LEVEL(5.5,0);

SET_OUTPUT_LEVEL(2.4,0.4);

SET_PERIOD(800);

SET_TIMING(250,480,600);

FORMAT(NRZ0,"1-13,33-40,44,45");

FORMAT(RO,"48");

FORMAT(NRZ0,"47");

RUN_PATTERN(3,1,0,0);

 PMU_CONDITIONS(FVMI,0,V,0.02,MA);

```
if(!PMU_MEASURE("1-13,44,45,47,48",5,"ILI_1_",UA,1,-1))
    BIN(5);
```

```
RUN_PATTERN(4,1,0,0);
```

```
    PMU_CONDITIONS(FVMI,5.5,V,0.02,MA);
```

```
if(!PMU_MEASURE("1-13,44,45,47,48",5,"ILI_1_",UA,1,-1))
    BIN(5);
```

```
//ILO
```

```
/*
```

```
测试条件：  $VSS \leq VI/O \leq VCC$ ; /CE1=VIH;
```

```
or CE2 = VIL;or /OE=VIH;or /WE = VIL
```

```
*/
```

```
SET_DPS(1,5.5,V,100,MA);
```

```
SET_INPUT_LEVEL(2.2,0.8);
```

```
SET_OUTPUT_LEVEL(2.4,0.4);
```

```
    SET_PERIOD(800);
```

```
    SET_TIMING(250,480,600);
```

```
FORMAT(NRZ0,"1-13,33-40,44,45");
```

```
FORMAT(RO,"48");
```

```
FORMAT(NRZ0,"47");
```

```
RUN_PATTERN(5,1,0,0);
```

```
    PMU_CONDITIONS(FVMI,0,V,0.02,MA);
```

```
if(!PMU_MEASURE("33-40",5,"ILO_1_",UA,1,-1))
```

```
    BIN(6);
```

```

RUN_PATTERN(5,1,0,0);
    PMU_CONDITIONS(FVMI,5.5,V,0.02,MA);
    if(!PMU_MEASURE("33-40",5,"ILO_2_",UA,1,-1))
        BIN(6);

// ICC 测试

/*
测试条件: cycle time =min;Ii/o =0mA;/CE1=VIL,CE2=VIH
*/

SET_DPS(1,5.5,V,100,MA);
SET_INPUT_LEVEL(2.2,0.8);
SET_OUTPUT_LEVEL(2.4,0.4);
    SET_PERIOD(70);
    SET_TIMING(10,45,55);
    FORMAT(NRZ0,"1-13,33-40,44,45");
    FORMAT(RO,"48");
    FORMAT(RO,"47");
    RUN_PATTERN(6,1,0,0);
    if (!DPS_MEASURE(1,R200MA,5,"ICC",MA,45,No_LoLimit))
        BIN(7);
    SET_MASKJMP();

/*
测试条件: cycle time =1us;Ii/o =0mA;
/CE1=0.2V,CE2=VCC-0.2V;
other pins at0.2V or VCC-0.2V
*/

SET_DPS(1,5.5,V,60,MA);

```

```
SET_INPUT_LEVEL(5.3,0.2);
SET_PERIOD(1000);
SET_TIMING(250,750,800);
RUN_PATTERN(6,1,0,0);
Delay(20);
if (!DPS_MEASURE(1,R200MA,5,"ICC1_1us",MA,30,No_LoLimit))
    BIN(7);
SET_MASKJMP();

/*
测试条件: cycle time =500ns;Ii/o =0mA;
/CE1=0.2V,CE2=VCC-0.2V;
other pins at0.2V or VCC-0.2V
*/
SET_DPS(1,5.5,V,60,MA);
SET_INPUT_LEVEL(5.3,0.2);
SET_PERIOD(500);
SET_TIMING(50,300,450);
RUN_PATTERN(6,1,0,0);
if (!DPS_MEASURE(1,R200MA,5,"ICC2_500ns",MA,15,No_LoLimit))
    BIN(7);
SET_MASKJMP();
```

```
//ISB 测试
```

```
/*
测试条件:
/CE1=VIH or CE2 = VIL
```

```

*/
SET_DPS(1,5.5,V,20,MA);
SET_INPUT_LEVEL(2.2,0.8);
SET_OUTPUT_LEVEL(2.4,0.4);
    SET_PERIOD(800);
    SET_TIMING(250,480,600);
FORMAT(NRZ0,"1-13,33-40,44,45");
FORMAT(RO,"48");
FORMAT(NRZ0,"47");
RUN_PATTERN(5,1,0,0);
if (!DPS_MEASURE(1,R20MA,5,"ISB",MA,3,No_LoLimit))
    BIN(7);

/*
ISB1 测试条件:

/CE1 ≥ VCC-0.2V;   or CE2 ≤ 0.2V;
other pins at 0.2V or Vcc-0.2V

*/
SET_DPS(1,5.5,V,0.2,MA);
SET_INPUT_LEVEL(5.3,0.2);
RUN_PATTERN(5,1,0,0);
if (!DPS_MEASURE(1,R200UA,5,"ISB1",UA,50,No_LoLimit))
    BIN(7);

iTimeEnd = GetTickCount();
SHOW_RESULT("测试时间",iTimeEnd-
iTimeStart,"ms",No_UpLimit,No_LoLimit);
}

```

5.3 小结

数字器件 54LS00 手册中有几个相似的不同型号的器件可作为练习，静态存储器

6264 注意地址管脚的定义与《国标》测试方法的使用。

测试范例中的程序（代码）是基于每个测试工程师对被测器件（datasheet）的理解自编测试方案而编写的，答案不唯一，仅供参考。

第二部分 模拟系统

第六章 模拟集成电路测试系统硬件资源

6.1 模拟测试系统结构

- Windows XP/WIN7/10 操作系统
- 测试程序开发语言采用 Visual C++ 6.0 / Visual Studio 2010
- 提供功能测试、交流参数测试和直流参数测试
- 1 路高压器件电压源
- 1 路参数测量单元
- 1 路电压电流源
- 音频电压表/音频电压源
- 4 路偏置电源
- 2 路时间测量单元
- 连接探针台、机械手的接口

ATE 模拟测试系统特点：测试系统资源丰富且全面，根据被测器件数据手册自行设计测试方案，理解测试系统硬件结构设计适配器，了解器件类别的测试方法，根据数据手册中电路图理解指令的使用规则编写测试程序。

测试结果=（测试设备使用说明+被测器件数据手册+器件测试方法（国标）+适配器

（测试接口板）+测试程序）× 调试

说明：为了方便初学者我们设备外接资源提供制作好的适配器供参考，模拟系统设计了可测试单、双、四运放的测试接口板，省去了自行设计电路环节。测运放之外的其他模拟器件需自行设计适配器。

6.2 模拟测试系统资源板配置

Solt11	Solt2	Solt3	Solt4
SBB	PMU/VIS	HDPS	AnalogB

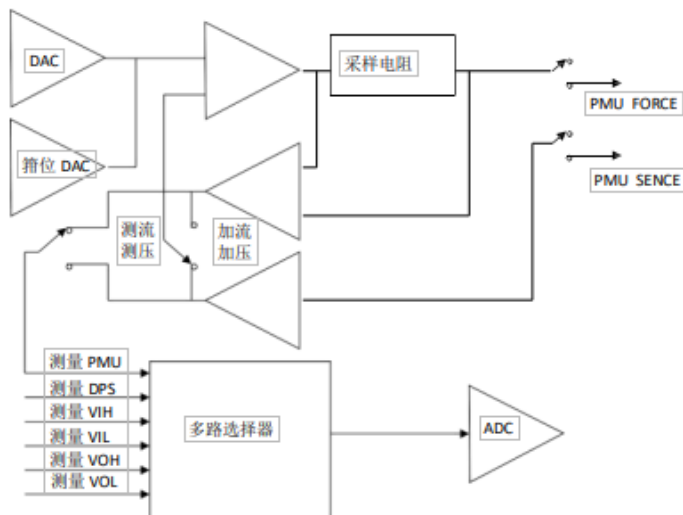
6.2.1 系统总线控制板 (SBB)

系统总线控制板提供 ST3020 测试系统与计算机接口卡、Handler/Prober、测试系统工作状态指示灯的连接。计算机与测试仪通过网口连接到系统总线控制板，通过测试仪内部网口板转 68 路数据线与总线板链接通信。

6.2.2 参数测量单元/电压电流源 (PMU/VIS)

PMU/VIS，可以驱动、测量电压或电流，电压的驱动、测量范围为 $\pm 15V$ ，电流的驱动、测量范围为 $\pm 300mA$ ，

PMU/VIS 有以下两种工作方式：



(1) 加压测流 (FVMI)

方式

在 FVMI 方式中，驱动电压值通过数模转换器提供给输出驱动器；驱动电流由采样电阻采样，通过差分放大器转换成电压值，再由模数转换器读回电流值。箝位值可根据

负载设定，箝位电路在这里起到限流保护作用，当负载电流超过箝位值时，PMU 输出变为恒流源，输出电流为箝位电流。测试系统根据箝位值自动选择测流量程。

(2) 加流测压 (FIMV) 方式

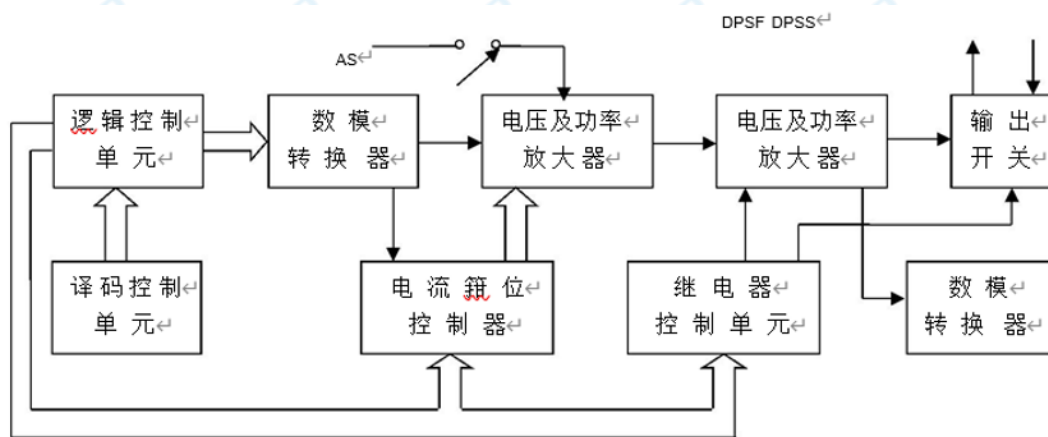
在 FIMV 方式中，驱动电流值通过数模转换器提供给输出驱动器；电压由模数转换器读回。箝位值可根据负载设值，箝位电路在这里起到限压保护作用，当负载电压超过箝位值时，PMU 输出变为恒压源，输出电压为箝位电压。测试系统根据箝位值自动选择测压量程。

6.2.3 高压器件电压源 (HDPS)

(1) 在测试过程中根据施加条件为被测器件提供两路器件电源，其中一路电压施加范围：0V~50V，另一路电压施加范围：-50V~0V。

(2) 测量器件电源的工作电流，测量值最大 200mA。

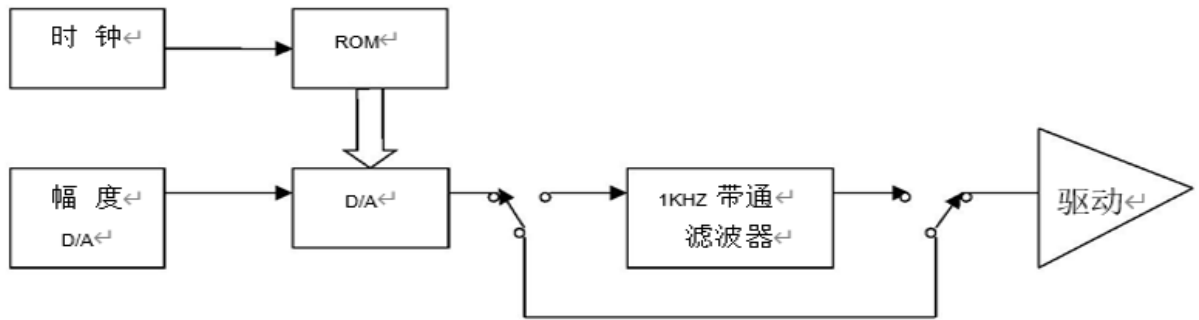
高压器件电源板分别由 DPS1 和 DPS2 两路器件电源组成，它的主要组成部分：译码控制单元、逻辑控制单元、数模转换器、电压及功率放大器、电流箝位控制器、电流采样单元、差分放大器、继电器控制单元。



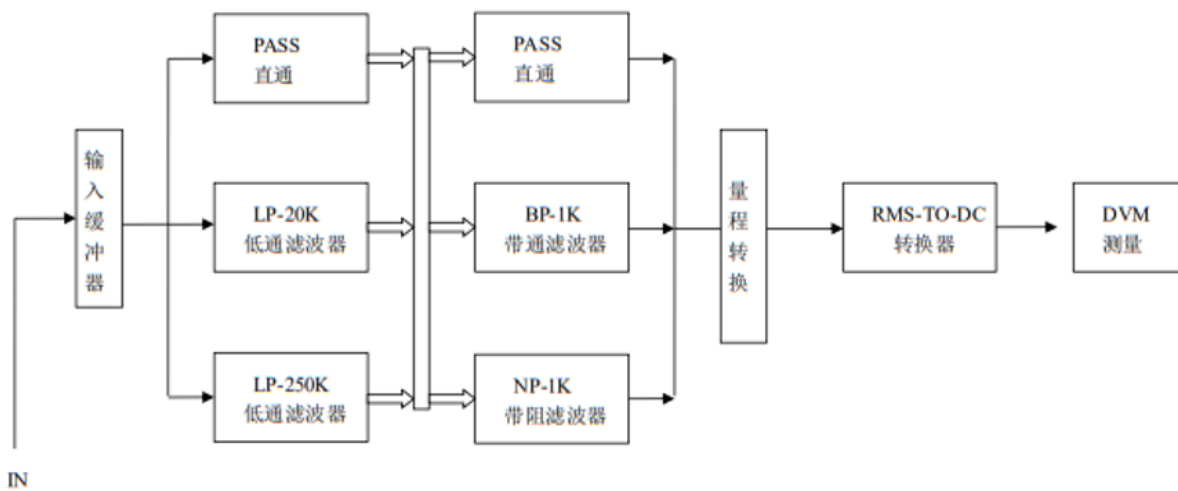
6.2.4 模拟板 (Analog)

模拟功能板是集音频信号源、音频电压表、时间测量单元、偏置电压源、继电器控制位为一体的多功能电路板，具有施加交流信号、测量信号幅度、信号时间、控制继电器的功能。多功能的集成保证了板卡的可靠性。多种的用户指令可灵活使用。

(1) 音频电压源(AS)。音频电压源可以产生频率为 1HZ~200KHZ，幅度为 0~4Vrms 的正弦波。音频电压源主要包括幅度控制电路、图形存储电路、时钟控制电路、波形合成电路及输出驱动电路。



音频波形存储在图形存储器中，通过时钟控制电路产生存储器的地址，依次将波形。数据逐点输出到 D/A，产生正弦波信号。输出幅度由一个 12 位 D/A 控制，如果



经过 1KHz。带通滤波器可产生 1KHz 低失真度正弦信号，音频源输出阻抗为 50Ω，输出电流为 20mA。

(2) 音频电压表(AVM) 主要功能：测量音频范围电压信号的幅度。AVM 由直通、20K/250K 低通、1K 带通、1K 带阻滤波器和 RMS-TO-DC 转换器等部分组成；各个通路的设置以及量程的使用均可由编程控制；测量信号范围 0 ~ 4Vrms。AVM 的量程选择分别为：4V、400MV、40MV、4MV。

(3) 偏置电压源(VS) 偏置电压源共有四路: VS1、VS2、VS3、VS4。每路提供电压范围 $0 \sim \pm 15V$ 。

(4) 时间测量单元(TMU)。用于测量信号的间隔时间, 其包含 T1 和 T2 两个输入信号端。可以用于:

- A. 由 T1 端输入, 选择测量信号周期、脉冲宽度或斜率。
- B. 由 T2 端输入, 选择测量信号周期、脉冲宽度或斜率。
- C. 由 T1 和 T2 端同时输入时, 除了可以完成 (1) 和 (2) 的功能外还可以选择

T1 和 T2 的前后沿, 测量边沿之间的时间间隔。

时间测量范围: $200nS \sim 17S$; 门限电压: 两路; 电压范围: $-10V \sim +10V$

(2) 继电器控制位(RELAY) 。继电器控制位共 40 位, K1 ~ K40; 使用固定线圈电压—5V。每个继电器控制位可提供驱动电流 $I_K \leq 50mA$ 。

以上 1-4 类板卡属于设备内部功能结构。

6.3 小结:

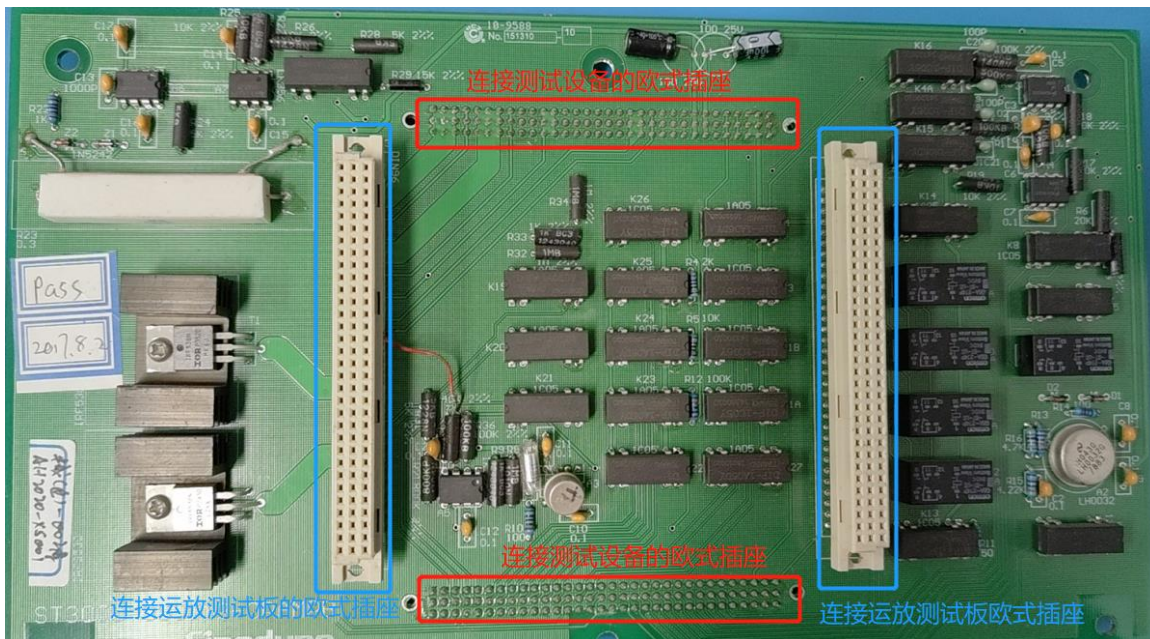
介绍模拟测试系统结构, 以上是测试设备机箱内部各板卡功能, 理解测量原理, 学习使用测量仪器仪表, 更便于理解 ATE 测试系统的工作原理。

6.4 模拟测试系统适配器

适配器是设备系统外接被测器件的测试板, 如: 运放包资源板专用来测试运放。也可自己根据需要设计 PCB, 搭建适合器件测试的电路环境。

6.4.1 单运放测试

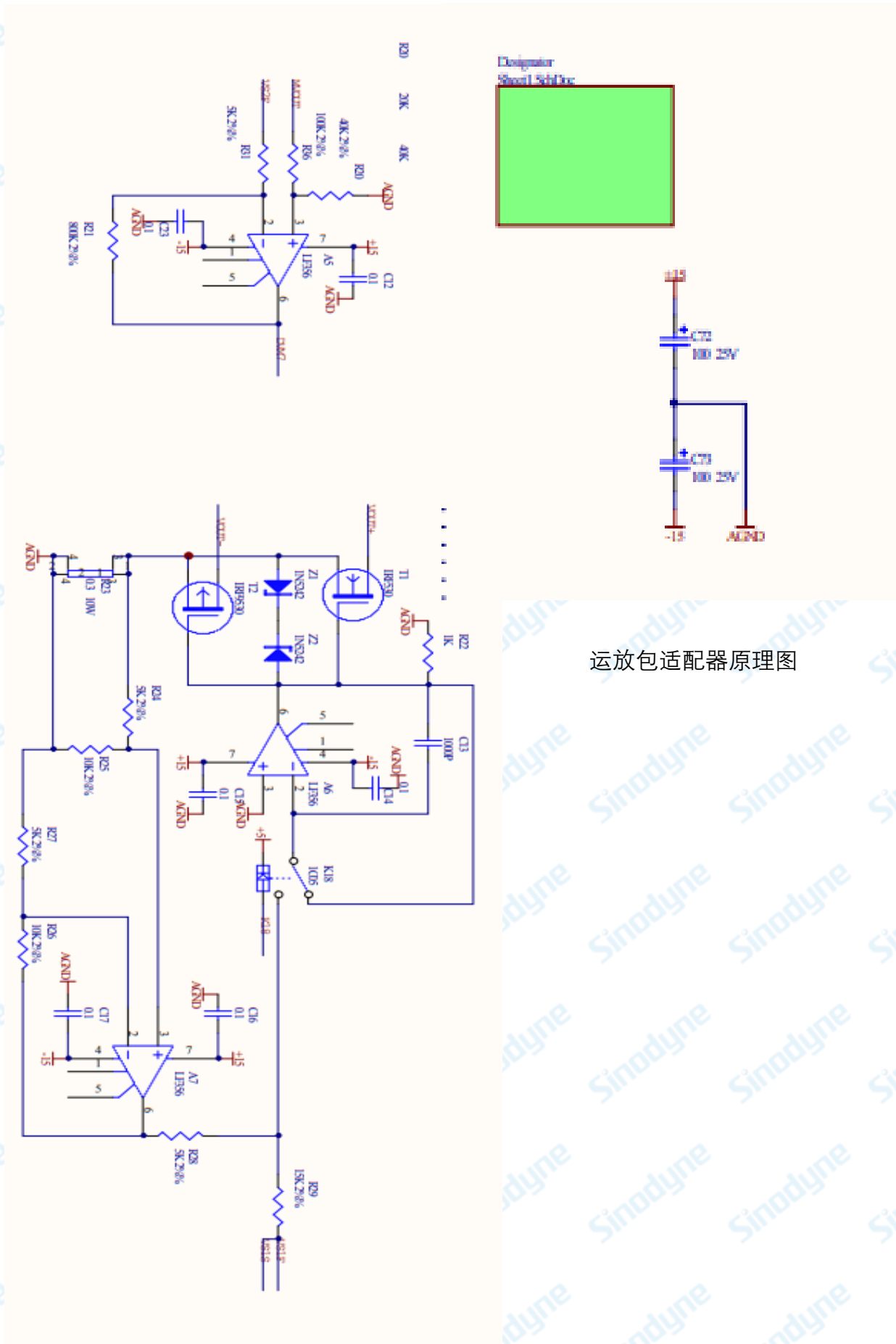
此适配器可测单运放、双运放、四运放。

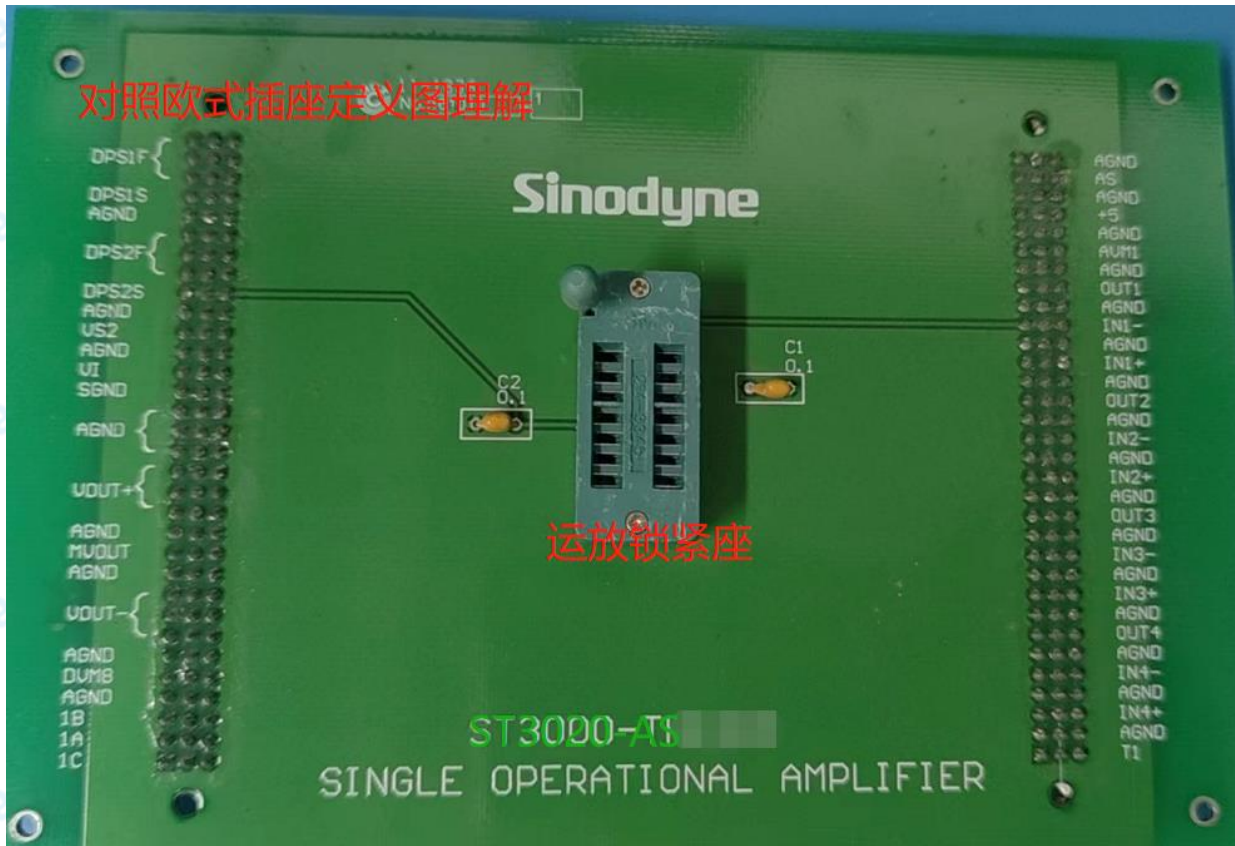


链接测试设备运放包（正面）

96CON	J1A	J1B	J1C	PA	DB	J1C
1	1 AS	33 AS	65 AS	1 T1	33 N29	65 N24
2	2 +5	34 +5	66 +5	2 T1	34 N28	66 N23
3	3 +5	35 +5	67 +5	3 T2	35 N27	67 N22
4	4 +12V	36 +12V	68 +12V	4 T2	36 N26	68 N21
5	5 AVMC	37 AVMC	69 AVMC	5 GND	37 N25	69 N20
6	6 AVMI	38 AVMI	70 AVMI	6 GND	38 K16	70 N19
7	7 -15	39 -15	71 -15	7 GND	39 K15	71 N18
8	8 +15	40 +15	72 +15	8 GND	40 K14	72 N17
9	9 AGND	41 AGND	73 AGND	9 DVM1	41 K13	73 N16
10	10 AGND	42 AGND	74 AGND	10 DVM2	42 K12	74 N15
11	11 AGND	43 AGND	75 AGND	11 DVM3	43 K11	75 N14
12	12 SGND	44 SGND	76 SGND	12 DVM4	44 K10	76 N13
13	13 VIS	44 VIS	76 VIS	13 DVM5	44 K9	76 N12
14	14 VIE	46 VIE VIS	78 VIE	14 DVM6	45 K8	77 N11
15	15 PML5	47 PML5	79 PML5	15 DVM7	46 K7	78 N10
16	16 PML6	48 PML6	80 PML6	16 DVM8	47 K6	79 N9
17	17 VSIS	49 VSIS	81 VSIS	17 DDD16	48 K5	80 N8
18	18 VSIF	50 VSIF	82 VSIF	18 DDD15	49 K4	81 N7
19	19 VSIS	51 VSIS	83 VSIS	19 DDD14	50 K3	82 N6
20	20 VSIF	52 VSIF	84 VSIF	20 DDD13	51 K2	83 N5
21	21 VSIS	53 VSIS	85 VSIS	21 DDD12	52 K1	84 N4
22	22 VSIF	54 VSIS VSIF	86 VSIF	22 DDD11	53 K77	85 N3
23	23 VSIS	55 VSIS	87 VSIS	23 DDD10	54 K26	86 N2
24	24 VSIF	56 VSIF	88 VSIF	24 DDD9	55 K25	87 N1
25	25 DPSS	57 DPSS	89 DPSS	25 DDD8	56 K24	88 N0
26	26 DPSE	58 DPSE	90 DPSE	26 DDD7	57 K23	89 DDD23
27	27 DPSE	59 DPSE	91 DPSE	27 DDD6	58 K22	90 DDD22
28	28 DPSE	60 DPSE	92 DPSE	28 DDD5	59 K21	91 DDD21
29	29 DPSE	61 DPSE	93 DPSE	29 DDD4	60 K20	92 DDD20
30	30 DPSE	62 DPSE	94 DPSE	30 DDD3	61 K19	93 DDD19
31	31 DPSE	63 DPSE	95 DPSE	31 DDD2	62 K18	94 DDD18
32	32 DPSE	64 DPSE	96 DPSE	32 DDD1	63 K17	95 DDD17
					64	96 DDD16

连接测试设备欧式插座定义图





连接适配器的运放接口板

BA			BB			BC		
1	1	DPS1F	33	33	DPS1F	65	65	DPS1F
2	2	DPS1F	34	34	DPS1F	66	66	DPS1F
3	3	DPS1F	35	35	DPS1F	67	67	DPS1F
4	4	DPS1S	36	36	DPS1S	68	68	DPS1S
5	5	AGND	37	37	AGND	69	69	AGND
6	6	DPS2F	38	38	DPS2F	70	70	DPS2F
7	7	DPS2F	39	39	DPS2F	71	71	DPS2F
8	8	DPS2F	40	40	DPS2F	72	72	DPS2F
9	9	DPS2S	41	41	DPS2S	73	73	DPS2S
10	10	AGND	42	42	AGND	74	74	AGND
11	11	VS2F	43	43	VS2F	75	75	VS2F
12	12	AGND	44	44	AGND	76	76	AGND
13	13	VTF	45	45	VTF	77	77	VTF
14	14	SGND	46	46	SGND	78	78	SGND
15	15	AGND	47	47	AGND	79	79	AGND
16	16	AGND	48	48	AGND	80	80	AGND
17	17	AGND	49	49	AGND	81	81	AGND
18	18	VOUT+	50	50	VOUT+	82	82	VOUT+
19	19	VOUT+	51	51	VOUT+	83	83	VOUT+
20	20	VOUT+	52	52	VOUT+	84	84	VOUT+
21	21	AGND	53	53	AGND	85	85	AGND
22	22	MUCUT	54	54	MUCUT	86	86	MUCUT
23	23	AGND	55	55	AGND	87	87	AGND
24	24	VOUT-	56	56	VOUT-	88	88	VOUT-
25	25	VOUT-	57	57	VOUT-	89	89	VOUT-
26	26	VOUT-	58	58	VOUT-	90	90	VOUT-
27	27	AGND	59	59	AGND	91	91	AGND
28	28	DUM8	60	60	DUM8	92	92	DUM8
29	29	AGND	61	61	AGND	93	93	AGND
30	30	K5	62	62	K5	94	94	K5
31	31	K2	63	63	K2	95	95	K2
32	32	K17	64	64	K17	96	96	K17

运放测试接口板欧式插座定义图 (一)

HA			HB			HC		
1	1	AGND	33	33	AGND	65	65	AGND
2	2	AS	34	34	AS	66	66	AS
3	3	AGND	35	35	AGND	67	67	AGND
4	4	+5	36	36	+5	68	68	+5
5	5	AGND	37	37	AGND	69	69	AGND
6	6	AVM2	38	38	AVM2	70	70	AVM2
7	7	AGND	39	39	AGND	71	71	AGND
8	8	OUT1	40	40	OUT1	72	72	OUT1
9	9	AGND	41	41	AGND	73	73	AGND
10	10	TN1-	42	42	TN1-	74	74	TN1-
11	11	AGND	43	43	AGND	75	75	AGND
12	12	TN1+	44	44	TN1+	76	76	TN1+
13	13	AGND	45	45	AGND	77	77	AGND
14	14	OUT2	46	46	OUT2	78	78	OUT2
15	15	AGND	47	47	AGND	79	79	AGND
16	16	TN2-	48	48	TN2-	80	80	TN2-
17	17	AGND	49	49	AGND	81	81	AGND
18	18	TN2+	50	50	TN2+	82	82	TN2+
19	19	AGND	51	51	AGND	83	83	AGND
20	20	OUT3	52	52	OUT3	84	84	OUT3
21	21	AGND	53	53	AGND	85	85	AGND
22	22	TN3-	54	54	TN3-	86	86	TN3-
23	23	AGND	55	55	AGND	87	87	AGND
24	24	TN3+	56	56	TN3+	88	88	TN3+
25	25	AGND	57	57	AGND	89	89	AGND
26	26	OUT4	58	58	OUT4	90	90	OUT4
27	27	AGND	59	59	AGND	91	91	AGND
28	28	TN4-	60	60	TN4-	92	92	TN4-
29	29	AGND	61	61	AGND	93	93	AGND
30	30	TN4+	62	62	TN4+	94	94	TN4+
31	31	AGND	63	63	AGND	95	95	AGND
32	32	TI	64	64	TI	96	96	TI

运放测试接口板欧式插座定义图（二）

6.5 小结

被测器件与测试设备建立硬连接就是通过适配器，根据被测器件的封装工艺采购匹配的测试锁紧座，根据测试系统接口板原理图设计 PCB，最后将锁紧座焊接好，进行连接性测试。

如测运放，则可直接用随设备配套的运放适配器，也可自行设计适配器；如测其他模拟器件，可自行搭建整套电路环境设计 PCB 插在测试设备上使用。

第七章 模拟集成电路测试系统软件

此章节参考数字集成电路测试系统软件操作流程。

第八章 模拟集成电路测试指令

8.1 模拟测试系统指令集

序号	指令	解释
1	SET_VIS ()	设置电压电流源
2	VI_MEASURE ()	测量电压电流
3	CLEAR_VIS ()	清电压电流源
4	SET_DPS_POS ()	设置 HDPS 正电源电压
5	DPS_MEASURE_POS ()	测量 HDPS 正电源电流
6	DPS_OFF_POS ()	关闭 HDPS 正电源
7	SET_DPS_NEG ()	设置 HDPS 负电源电压
8	DPS_MEASURE_NEG ()	测量 HDPS 负电源电流
9	DPS_OFF_NEG ()	关闭 HDPS 负电源
10	SET_VS1 ()	设置第一路偏置电压源电压
11	SET_VS2 ()	设置第二路偏置电压源电压
12	SET_VS3 ()	设置第三路偏置电压源电压
13	SET_VS4 ()	设置第四路偏置电压源电压
14	CLEAR_VS ()	清偏置电压源
15	SET_AS ()	设置音频电压源输出交流
16	SET_AS_DC ()	设置音频电压源输出直流
17	SET_AS_SQUARE ()	设置音频电压源输出方波
18	RUN_AS_PATTERN ()	设置音频源输出波形
19	LOAD_AS_PATTERN ()	加载音频源波形
20	SET_POS_RIPPLE ()	设置负电源纹波

21	CLEAR_POS_RIPPLE ()	清正电源纹波
22	SET_NEG_RIPPLE ()	设置负电源纹波
23	CLEAR_NEG_RIPPLE ()	清负电源纹波
24	SET_AVM_PATH ()	设置音频电压通路
25	AVM_MEASURE ()	测量音频电压
26	TIME_MEASURE_CONDITIONS ()	设置时间测量条件
27	TIME_MEASURE_EXECUTE ()	时间测量 (与上条指令连用)
28	MEASURE_PERIOD ()	测量周期
29	DVM_MEASURE ()	直接电压测量
30	MAT_DVM_MEASURE ()	连续电压测量
31	DVM_DIFF_MEASURE ()	间隔时间直接电压测量
32	MEASURE_POS_REGVO ()	测量电压调整器/三端电源的输出电压(正电压)
33	MEASURE_NEG_REGVO ()	测量电压调整器/三端电源的输出电压(负电压)
34	SET_RELAY ()	闭合指定继电器, 其他继电器打开
35	CLOSE_RELAY ()	闭合指定继电器, 其他保持原状态
36	CLEAR_RELAY ()	清继电器
37	SHOW_RESULT ()	显示测试结果 (标准单位/非标准单位/测试功能)
38	Delay ()	延时 (单位: ms)
39	BIN ()	设置分箱号
40	CLEAR_ALL ()	清系统
41	DFT ()	傅里叶变换

8.2 指令集中 41 条指令的详细用法

(1) SET_VIS ()电压电流源

[函数原形]: void SET_VIS(unsigned int Mode, double Value, unsigned int Value_Unit, double Clamp_Value, unsigned int Clamp_Unit)

[参数说明]: 参照“SET_PMU”

(2) VI_MEASURE()测量电压电流

[函数原形]: double VI_MEASURE(unsigned int tDelay)

[参数说明]: tDelay: 测量延迟时间, 单位: MS; [返回值] 实际测量值

(3) CLEAR_VIS()清电压电流源

(4) SET_DPS_POS()设置 HDPS 正电源电压

[函数原形]: void SET_DPS_POS(double Vdd, unsigned int Vdd_Unit, double Iclamp, unsigned int Iclamp_Unit)

[参数说明]: Vdd: 设置 HDPS 正电源电压值, 范围: 0 ~ +50V

Vdd_Unit: 电压单位: 选择 MV 或 V

Iclamp: 电流箝位值, 范围: 0~5A

clamp_Unit: 箝位单位: 选择 UA、mA、A

注意: 如果被测器件电流大于 200mA, 则

SET_DPS_POS()DPS_MEASURE_POS() DPS_OFF_POS()三条指令应连用, 执行的总时间不能超过 **50ms**;

(5) DPS_MEASURE_POS()测量 HDPS 正电源电流

[函数原形]: double DPS_MEASURE_POS(unsigned int uiDelaysms)

[参数说明]: uiDelaysms: 测量延迟时间, 单位: ms

(6) DPS_OFF_POS()关闭 HDPS 正电源

(7) SET_DPS_NEG()设置 HDPS 负电源电压

参照 SET_DPS_POS()

(8) DPS_MEASURE_NEG()测量 HDPS 负电源电流

参照 DPS_MEASURE_POS()

(9) DPS_OFF_NEG()关闭 HDPS 负电源

(10) SET_VS1()设置第一路偏置电压源电压

[函数原形]: void SET_VS1(double fVolt,int iVolUnit)

[参数说明]: fVolt: 设置电压值, 范围: -15V~+15V

iVolUnit: 电压单位: 选择 MV 或 V

(11) SET_VS2()参照 SET_VS1 ()

(12) SET_VS3()参照 SET_VS1 ()

(13) SET_VS4()参照 SET_VS1 ()

(14) CLEAR_VS()清电压源

(15) SET_AS ()设置音频电压源输出交流

[函数原形]: void SET_AS(double fVol,int iVolUnit,double freq, int iUnit)

[参数说明]: fVol:: 音频源电压值, 范围: 0~4Vrms

iVolUnit: 电压单位: 选择 MV 或 V

freq: 频率或周期值, 范围: 1Hz~200KHz

iUnit: 频率或周期的单位: 选择 HZ、KHZ、MS、US、NS

(16) SET_AS_DC()设置音频电压源输出直流

[函数原形]: void SET_AS_DC(double fVol, int iVolUnit)

[参数说明]: fVol:: 音频源电压值, 范围: -5.0V~+5.0V

iVolUnit: 电压单位: 选择 MV 或 V

(17) SET_AS_SQUARE()设置音频电压源输出方波

[函数原形]: void SET_AS_SQUARE(double fVol,int iVolUnit,double freq, int iUnit)

参照 SET_AS ()

(18) RUN_AS_PATTERN () 设置音频源输出波形。

与 LOAD_AS_PATTERN 配合使用。

[函数原形]: void RUN_AS_PATTERN(WORD mode,WORD bank,WORD
freq,double fVol)

[参数说明]:

mode: 方式。

0: 131072 点/段; 共 1 段 ;

1: 65535 点/段; 共 2 段;

2: 32768 点/段; 共 4 段;

3: 16384 点/段; 共 8 段

4: 8192 点/段; 共 16 段

5: 4096 点/段; 共 32 段

6: 2048 点/段; 共 64 段

7: 1024 点/段; 共 128 段

8: 512 点/段; 共 256 段

bank: 段号

freq: 采样频率: 100M/freq(freq 输入范围:10~65535)

fVol:音频源电压值。输入范围: 10V。

(19) LOAD_AS_PATTERN () 加载音频源波形。

[函数原形]: bool LOAD_AS_PATTERN(WORD mode,WORD bank,WORD* data)

[参数说明]:

mode: 方式。

0: 131072 点/段; 共 1 段 ;

1: 65535 点/段; 共 2 段;

2: 32768 点/段; 共 4 段;

3: 16384 点/段; 共 8 段

4: 8192 点/段; 共 16 段

5: 4096 点/段; 共 32 段

6: 2048 点/段; 共 64 段

7: 1024 点/段; 共 128 段

8: 512 点/段; 共 256 段

bank: 段号

* data:数据

(20) SET_POS_RIPPLE()设置正电源纹波

[函数原形]: void SET_POS_RIPPLE(double fVol, int iVolUnit, double freq)

[参数说明]: fVol: 电压值, 范围: 0~1V, iVolUnit: 电压单位: 选择 MV 或 V

freq: 频率值, 单位: Hz, 范围: 1Hz~3KHz

(21) CLEAR_POS_RIPPLE()清正电源纹波

(22) SET_NEG_RIPPLE()设置负电源纹波

[函数原形]: void SET_NEG_RIPPLE(double fVol, int iVolUnit, double freq)

[参数说明]: fVol: 电压值, 范围: 0~1V

iVolUnit: 电压单位: 选择 MV 或 V

freq: 频率值, 单位: Hz, 范围: 1Hz~3KHz

(23) CLEAR_NEG_RIPPLE()清负电源纹波

(24) SET_AVM_PATH()设置音频电压测量通路

[函数原形]: void SET_AVM_PATH(int iLpSel,int iBpSel)

[参数说明]: iLpSel: 通路选择: LPPASS 或 0 ----- PASS

LP20 或 1 ----- 20KHz 低通

LP250 或 2 ----- 250KHz 低通

iBpSel: 带通选择: BPPASS 或 0 ----- PASS

BP1K 或 1 ----- 1KHz 带通

BPNOT 或 2 ----- 1KHz 带阻

音频电压通路, 选择上面两个参数的组合, 共九种。

(25) AVM_MEASURE()测量音频电压

[函数原形]: double AVM_MEASURE(int iMeasNo, double fVoltage, int iVolUnit,

unsigned int tDelay)

[参数说明]: iMeasNo: AVM 通道编号, 1-2

fVoltage: 测量量程

iVolUnit: 电压单位: 选择 MV V

tDelay: 测量延迟时间, 单位 MS

[返回值] 实际测量值

(26) TIME_MEASURE_CONDITIONS()设置时间测量条件

[函数原形]: void TIME_MEASURE_CONDITIONS(int iStartT, int iEdgeStart,double
fVol1, intiUnit1,int iStopT,int iEdgeStop,double fVol2, int iUnit2);

[参数说明]: iStartT : 定时起始端, 选择 T1、T2

iEdgeStart: 定时起始沿, 选择 POS、NEG

fVol1 : 定时起始端电压值, 范围: -10V~10V

iUnit1 : 定时起始端电压单位, 选择 MV 或 V

iStopT : 定时结束端, 选择 T1、T2

iEdgeStop: 定时结束沿, 选择 POS、NEG

fVol2: 定时结束端电压值, 范围: -10V~10V

iUnit2: 定时结束端电压单位, V 、 MV

(27) TIME_MEASURE_EXECUTE()时间测量 (与 TIME_MEASURE_CONDITIONS

指令连用)

[函数原形]: double TIME_MEASURE_EXECUTE(unsigned int iDelay);

[参数说明]: iDelay: 测量时间, 单位: ms; [返回值] 实际测量值

(28) MEASURE_PERIOD()测量周期

[函数原形]: double MEASURE_PERIOD(int iStartT, double fVoltage, int

iVolUnit, intiNum, int iDelay)

[参数说明]: iStartT: 定时起始端子, 选择 T1、T2

fVoltage: 门槛电压值, 范围: -10V~+10V

iVolUnit: 电压单位: 选择 MV 或 V

iNum: 周期数

iDelay: 测量时间, 单位: ms

[返回值] 实际测量值

(29) DVM_MEASURE()直接电压测量

[函数原形]: double DVM_MEASURE(int iMeasNo, double fVoltage, int iVolUnit, unsigned int tDelay)

[参数说明]: iMeasNo: DVM 通道编号, 1-8

fVoltage: 测量量程, 输入范围: -15V~+15V,

iVolUnit: 电压单位: 选择 MV 或 V

tDelay: 测量延迟时间, 单位: ms

[返回值] 实际测量值

[范例]: DVM 通道选择 1, 电压预期测量值为 0.05V, 测量延迟时间 1ms.

```
double fmeaVol = DVM_MEASURE(1, 0.06,V,10);
```

(30) MAT_DVM_MEASURE () 连续电压测量

[函数原形]: void MAT_DVM_MEASURE(int iMeasNo,double fVoltage, int iVolUnit,unsigned int tDelay, int ipNumber,in fFreq,double *fValue)

[参数说明]:

iMeasNo: 通道号

fVoltage: 量程电压

tDelay: 测量延迟时间, 单位: 毫秒 ms

ipNumber: 采样点数 (输入范围: 1~2048)

fFreq: 采样频率 (实际采样频率: 50MHz/fFreq)

*fValue: 测量返回值。指针型。程序中需定义数组, 数组个数应大于采样点数。

(31) DVM_DIFF_MEASURE() 间隔时间直接电压测量,

[函数原形]: double DVM_DIFF_MEASURE(int iMeasNo, double fVoltage, int iVolUnit, unsigned int tDelay)

[参数说明]: iMeasNo: DVM 通道编号, 1-8

fVoltage: 测量量程, 输入范围: -15V~+15V, 输入原则同

DVM_MEASURE()

iVolUnit: 电压单位: 选择 MV 或 V

tDelay : 测量间隔时间, 单位: ms

[返回值] 两次测量的电压差

(32) MEASURE_POS_REGVO () 测量电压调整器/三端电源的输出电压(正电压)

[函数原形]: double MEASURE_POS_REGVO(double fIload,int iUnit,double fVout,int

iVunit, int iRtimes)

[参数说明]: fIload: 负载电流, 范围: 0~5A

iUnit: 负载电流, 单位: 选择 UA、MA、A

fVout: 测量电压的预期值。

iVunit: 电压单位: 选择 MV 或 V

iRtimes: 测试次数, 最大为 6

(33) MEASURE_NEG_REGVO () 测量电压调整器/三端电源的输出电压(负电压)

参照 MEASURE_POS_REGVO ()

(34) SET_RELAY() 闭合指定继电器, 其他继电器打开

[函数原形]: void SET_RELAY (CString sRelay)

[参数说明]: sRelay : 继电器号, 为一串整数字符或一串由“, -”连接的整数范围, 中间不能有空格。

[范例]: 设置继电器 1, 3, 5, 6, 7, 8 合上, 其他继电器打开

```
SET_RELAY ("1, 3, 5, 6, 7, 8"); 或
```

```
SET_RELAY ("1, 3, 5-8");
```

(35) CLOSE_RELAY() 闭合指定继电器, 其他继电器保持原状态

[函数原形]: void CLOSE_RELAY (CString sRelay)

[参数说明]: sRelay : 继电器号, 为一串整数字符或一串由“, -”连接的整数范围, 中间不能有空格。

[范例]: 闭合继电器 1, 3, 4, 5, 8, 9, 10, 再闭合 12

```
SET_RELAY ( "1, 3-5, 8-9");
```

```
CLOSE_RELAY ("12");
```

(36) CLEAR_RELAY()打开指定继电器

[函数原形]: void CLEAR_RELAY(CString sRelay)

[参数说明]: sRelay : 继电器号, 为一串整数字符或一串由“, -”连接的整数范围, 中间不能有空格。

[范例]: 设置 2, 3, 7, 8, 9 继电器后, 打开 3 继电器

```
SET_RELAY ( "2, 3, 7-9");
```

```
CLEAR_RELAY ("3");
```

(37) SHOW_RESULT()显示标准单位 A, MA, V, MV 测试结果

[函数原形]: void SHOW_RESULT (CString csItem, double measure_value, int csUnit, double fUpLimit, double fDnLimit)void

SHOW_RESULT () 显示非标准单位测试结果

(CString csItem, double measure_value, CString csUnit, double fUpLimit, double fDnLimit)

SHOW_RESULT(CString csItem BOOL bTest)显示功能测量结果

[参数说明]: csItem: 测试项名称

measure_value: 测量值

csUnit: 单位: 使用, 测量结果自动转换单位显示; 使用函数 55, 接显示测量值和单位, 单位可为任意字符串, 如“Ω”

fUpLimit: 上限

DnLimit: 下限

bTest: FAIL 或 0 显示测试项失效

PASS 或 1 显示测试项合格

(38) Delay()设置延迟时间 (单位: ms)

[函数原形]: void Delay(double fMs)

[参数说明]: fMs: 延迟时间, 单位: ms, double 型

[范例]: Delay(0.1); 延迟 100us。

(39) BIN () 用于测试完成后的分箱处理。

[函数原形]: void BIN (BYTE bin)

[参数说明]: bin : 分箱号

[范例]: 如果 “con” 测量失效, 设置分箱号为 1。

```
if (! PMU_MEASURE (“1,3,7-9”,10,”con”,mA,-0.1,-1.9 ))  
    BIN (1) ;
```

(40) CLEAR_ALL()清系统

(41) DFT () 傅里叶变换

[函数原形]: BOOL DFT(double* dataRe, int numData,double* fftRe)

[功能] 傅里叶变换

[参数说明]

dataRe: 数据输入

numData: 采样数 (必须是 2 的整次幂)

fftRe: 数据输出

$$Ff = Fs/numData$$

Ff: 频域分辨率

Fs: 采样频率

numData: 采样点数

经过傅里叶变换计算出频域各个频点的幅值，保存在数组 fftRe 中

数组下标 (n) *Ff 等于这个频率点的频率。

8.3 小结

37 条指令通过理解函数原型，对应系统的测试资源进行调用，了解其测试方法及外围电路的搭建的环境进行测试程序的编写与验证。

第九章 被测器件数据手册

略

不同器件不同方法取不同信息。要读通器件功能，工作方式，性能特点，根据器件类别结合测试方法搭建外围电路进行测试程序的编写。

写程序时则细看数据手册中的“最大”“高小”“典型值”的指标，函数的应用根据指标参数定义调用图形，写出完整的程序文件。

具体每个器件不同，数据手册“datasheet”中所列出信息不同，对英文阅读能力有一定要求。但专业的关键词汇也基本固定，所以多读细读手册，是了解测试信息的重要来源。

第十章 模拟器件测试程序开发

10.1 运放测试范例

[实例一] LF356

//LF356 测试程序

```
#include "StdAfx.h"
```

```
#include "testdef.h"
```

```
#include "data.h"
```

```
#include "math.h"
```

```
bool flag=true;
```

```
double caldata;
```

```
void PASCAL LF356()
```

```
{
```

```
double M_V,V1,V2,I1,I2,IB1,IB2,Vos,M_V1;
```

```
// 校准(测量-1nA 电流所对应的积分电压)
```

```
if(flag)
```

```
{
```

```
    SET_VS3(1,V);
```

```
    SET_RELAY("19");
```

```
Delay(20);  
CLOSE_RELAY("21");  
  
Delay(20);  
caldata= DVM_DIFF_MEASURE(6,15,V,150);  
CLEAR_VS();  
flag = false;  
}
```

```
//电源电流
```

```
SET_DPS_NEG(-15,V,20,MA);  
  
SET_DPS_POS(15,V,20,MA);  
  
SET_VS4(0,V); SET_RELAY("4,9");  
  
I1 = DPS_MEASURE_POS(10)*1000;  
I2 = DPS_MEASURE_NEG(10)*1000;  
if(!SHOW_RESULT("IDD",I1,"mA",10,No_LoLimit))
```

```
    BIN(1);
```

```
if(!SHOW_RESULT("ISS",I2,"mA",No_UpLimit,-10))
```

```
    BIN(1);
```

```
//Input Bias Current (IB)
```

```
SET_RELAY("4,7,9,13");
```

```
Delay(20);
```

```
CLOSE_RELAY("21");
```

```
Delay(20);
```

```
V2 = DVM_DIFF_MEASURE(6,5,V,300)/2;
IB2 = -V2/caldata*1000;

if(!SHOW_RESULT("IB(-)",IB2,"pA",200,-200))
    BIN(2);
SET_RELAY("4,9,13");
Delay(20);
CLOSE_RELAY("21");
Delay(20);
V1 = DVM_DIFF_MEASURE(6,5,V,300)/2;
IB1 = -V1/caldata*1000;
if(!SHOW_RESULT("IB(+)",IB1,"pA",200,-200))
    BIN(2);
```

```
//Input Offset Current(Ios)
I1 = IB1-IB2;
if(!SHOW_RESULT("Ios",I1,"pA",50,-50))
    BIN(3);
```

```
//Input Offset VOLTage(Vos)
SET_RELAY("4,9");
M_V=DVM_MEASURE(4,5,V,10)/401;
```

```
Vos = M_V*1000;
if(!SHOW_RESULT("Vos",Vos,"mV",10,-10))
    BIN(4);
// Supply Voltage Rejection Ratio(PSRR)
SET_DPS_NEG(-5,V,20,MA);
SET_DPS_POS(5,V,20,MA);
V1 = DVM_MEASURE(4,5,V,10)/401;
V2 = fabs(M_V-V1);
M_V = 20*log10(20/V2);
if(!SHOW_RESULT("PSRR",M_V,"dB",No_UpLimit,80))
    BIN(5);
// Output Voltage Swing (VO)
SET_DPS_NEG(-15,V,20,MA);
SET_DPS_POS(15,V,20,MA);
SET_PMU(FVMI,0,V,50,MA);
SET_VS4(-15,V);
CLOSE_RELAY("24");
V1 = DVM_MEASURE(5,15,V,10);
SET_VS4(15,V);
V2 = DVM_MEASURE(5,15,V,10);
if(!SHOW_RESULT("+VO(RL=10K)",V1,"V",No_UpLimit,12))
```

```
    BIN(6);
if(!SHOW_RESULT("-VO(RL=10K)",V2,"V",-12,No_LoLimit))

    BIN(6);
CLEAR_RELAY("24");
CLOSE_RELAY("25");

V2 = DVM_MEASURE(5,15,V,10);
SET_VS4(-15,V);

V1 = DVM_MEASURE(5,15,V,10);
if(!SHOW_RESULT("+VO(RL=2K)",V1,"V",No_UpLimit,10))

    BIN(6);
if(!SHOW_RESULT("-VO(RL=2K)",V2,"V",-10,No_LoLimit))

    BIN(6);

// Avo

SET_VS4(-10,V);
V1 = DVM_MEASURE(4,5,V,10)/401;

Delay(10);
SET_VS4(10,V);
V2 = DVM_MEASURE(4,5,V,10)/401;

M_V1 = fabs(V1-V2);
if(M_V1==0.0)

M_V1 = 0.000004;
```

```
M_V = 20/(M_V1*1000);
if(!SHOW_RESULT("Avo",M_V,"V/mV",No_UpLimit,25))

    BIN(7);
CLEAR_RELAY("25");
CLEAR_VS();

// Common_Mode Rejection Ratio (CMRR)
SET_DPS_POS(5,V,20,MA);

SET_DPS_NEG(-25,V,20,MA);
SET_VS4(10,V);

V1 = DVM_MEASURE(4,5,V,10)/401;
SET_VS4(0,V);
SET_DPS_NEG(-5,V,20,MA);
SET_DPS_POS(25,V,20,MA);
SET_VS4(-10,V);

V2 = DVM_MEASURE(4,5,V,10)/401;
CLEAR_VS();

M_V = fabs(V2-V1);
if(M_V<0.000001)
M_V = 0.000001;
M_V = 20/M_V;
M_V = 20*log10(M_V);

if(!SHOW_RESULT("CMRR",M_V,"dB",No_UpLimit,80))
```

```
    BIN(8);  
    // Input Common_Mode Voltage Range (Vcm)  
  
    M_V = fabs(V1-Vos/1000);  
    if(M_V<0.000001)  
        M_V = 0.000001;  
  
    M_V = 10/M_V;  
    M_V = 20*log10(M_V);  
  
    if(!SHOW_RESULT("Vcm(+)",M_V,"dB",No_UpLimit,74))  
        BIN(9);
```

```
    M_V = fabs(V2-Vos/1000);  
    if(M_V<0.000001)  
        M_V = 0.000001;  
  
    M_V = 10/M_V;  
    M_V = 20*log10(M_V);  
  
    if(!SHOW_RESULT("Vcm(-)",M_V,"dB",No_UpLimit,74))  
        BIN(9);
```

```
    // Gain-Bandwidth Product
```

```
    SET_DPS_POS(15,V,20,MA);
```

```
SET_DPS_NEG(-15,V,20,MA);
SET_VS4(0,V);
SET_RELAY("4,9,14");
SET_AS(1,V,100000);
SET_AVM_PATH(LPPASS,BPPASS);
V1 = AVM_MEASURE(1,4,V,100);
V2 = AVM_MEASURE(2,2,V,100);
M_V = 40.1*V2/V1;
if(!SHOW_RESULT("GBW",M_V,"MHZ",No_UpLimit,No_LoLimit))
    BIN(10);

// SLEW RATE
SET_RELAY("6,27");

TIME_MEASURE_CONDITIONS(T2,POS,-4,V,T2,POS,4,V);
M_V = TIME_MEASURE_EXECUTE(100)*1000000;
M_V = 8/M_V;
if(!SHOW_RESULT("SR",M_V,"V/uS",No_UpLimit,No_LoLimit))
    BIN(11);
}
```

第 6 版系统手册补充说明

手册 P91-P92 页 (18), (19) 两条指令补充说明如下。

(18) RUN_AS_PATTERN () 设置音频源输出波形。与 LOAD_AS_PATTERN 配合使用。

[函数原形]: void RUN_AS_PATTERN(WORD mode,WORD bank,WORD freq,double fVol)

[参数说明]:

mode: 方式。

0: 131072 点/段; 共 1 段 ;

1: 65535 点/段; 共 2 段;

2: 32768 点/段; 共 4 段;

3: 16384 点/段; 共 8 段

4: 8192 点/段; 共 16 段

5: 4096 点/段; 共 32 段

6: 2048 点/段; 共 64 段

7: 1024 点/段; 共 128 段

8: 512 点/段; 共 256 段

bank: 段号

freq: 是对 100MHz 的系统时钟进行分频, 输入范围:10~65535, 实际采样频率=100MHz/freq

fVol:音频源电压值。输入范围: 10V。

(19) LOAD_AS_PATTERN () 加载音频源波形。

[函数原形]: bool LOAD_AS_PATTERN(WORD mode,WORD bank,WORD* data)

[参数说明]:

mode: 方式。

0: 131072 点/段; 共 1 段 ;

1: 65535 点/段; 共 2 段;

2: 32768 点/段; 共 4 段;

3: 16384 点/段; 共 8 段

4: 8192 点/段; 共 16 段

5: 4096 点/段; 共 32 段

6: 2048 点/段; 共 64 段

7: 1024 点/段; 共 128 段

8: 512 点/段; 共 256 段

bank: 段号

* data:数组 (波形数据 data 作为 D/A 转换器的输入, 需要转换成十六进制数据, 0XFFFF 对应负满度电压, 0X0000 对应正满度电压)。

例 1：输出一个频率 1KHZ，振幅 1V 的正弦波。假设我们采用 512 个点描这个正弦波，可以采用方式 8，段号 1。

频率计算：采样频率 $F=1\text{KHZ}\times 512=512\text{KHz}$ ， $\text{freq}=100\text{MHz}/F=100\text{MHz}/512\text{KHz}=195$ （取整），
实际产生的频率= $100\text{MHz}/\text{freq}/512=100\text{MHz}/195/512=1001.602\text{Hz}$

```
WORD data[512];
for (int i = 0;i<512;i++)
{
    data[i] = (sin(2*3.1415926*i/512)+1)/2*65535;
}

LOAD_AS_PATTERN(8,1,data);
RUN_AS_PATTERN(8,1,195,1);//100M/x    100M/195/512 = 1001.602
```

例 2：输出一个频率 1KHZ，幅值 2V 的方波。假设我们采用 512 个点描这个方波，同样采用方式 8，段号 2。

```
WORD data1[512];
for (int i = 0;i<512;i++)
{
    if(i<256)
    {
        data1[i]=0x0000;
    }
    else
    {
        data1[i]=0xffff;
    }
}

LOAD_AS_PATTERN(8,2,data1);
RUN_AS_PATTERN(8,2,195,2);//100M/x    100M/195/512
```