

ST3020 集成电路测试系统

使用手册

（高校专用）

北京信诺达泰思特科技股份有限公司

本手册适用机型包括**ST3020**，**ST3020-DS/AS**、**ST3020-D/64/128**、**ST3020-M/64**、**ST3020-M/32**。

目录

第一章	系统简介	1
1.	系统结构	1
1. 1	测试板安装表	2
1. 2	PMU/VIS (参数测量单元/电压电流源)	2
1. 3	DPS (器件电压源)	3
1. 4	数字通道板	4
1. 5	图形时钟板	6
1. 6	高压器件电压源(HDPS)	7
1. 7	模拟板 (Analog)	8
1. 8	系统总线控制板 (SBB)	10
1. 9	测试板	10
2.	系统环境	12
2. 1	配置表	13
2. 2	工作环境	13
3.	软件介绍	14
3. 1	简介	14
3. 2	软件安装	15
4.	软件编辑方式及使用	18
4. 1	创建 (创建模式)	19
4. 2	图形编译	19
4. 3	设置 (设置模式)	20
4. 4	测试 (测试模式)	23
第二章	指令与函数	28
1.	图形文件的生成、运行简介	28
2.	管脚定义	29
2. 1	管脚到通道定义	29
3.	图形文件编写步骤	30
4.	图形文件数据格式	31
5.	图形文件格式	32
5. 1	初始指定	32
5. 2	指令格式	32
5. 3	数据段标记	33
5. 4	图形结束	33
5. 5	注释	33
6.	图形文件指令	33
6. 1	指令集	33
6. 2	指令用法与范例	34
6. 3	图形文件指令解读	38
7.	测试程序编写函数	40
7. 1	函数目录	40
7. 2	函数用法与范例	41
第三章	器件手册信息提取	56
第四章	测试程序开发	57
1.	测试程序开发实例	57
第五章	接口板开发	69
第六章	安装调试	70
1.	硬件安装	70
2.	Handler 或 Prober 连接	70

3. 校准&自检适配板安装	71
4. 校准&自检程序启动	71
5. 校准程序使用说明	71
5. 1 校准项目	71
5. 2 校准操作步骤	72
5. 3 校准结果处理	73
6. 参数自检程序使用说明	74
6. 1 参数自检项目选择	74
6. 2 参数自检执行和结果处理	74
7. 功能自检程序使用说明	75
7. 1 功能自检模式选择	75
7. 2 功能自检执行	75
7. 3 注意事项	76
附录A: 校准&自检工具说明	77

第一章 系统简介

1. 系统结构

ST3020 集成电路测试系统可以快速、精确、全面地对各类（TTL、NMOS、CMOS）大、中、小规模数字集成电路以及运算放大器、比较器、模拟开关、音响电路、电话机电路、三端电源、A/D、D/A 等模拟、混合集成电路进行电参数测试。

适合集成电路生产厂家的生产测试（包括中测和成测）、设计部门进行设计验证、各类电子整机厂及质量监测部门进行集成电路进厂检验及质量/可靠性工程分析测试。

主要测试方式包括：合格/不合格方式（Pass/Fail）、数据记录方式（Data Log）。本系统具有专用外设接口，可与各类分选机、探针台等设备以完成 IC 的批量自动测试。

本系统体积小，稳定性高，操作方便，为目前国内市场上用于工业级生产与高校实验课堂的性能价格最高的 IC 测试系统。

注：硬件配置及实际功能以实际装机为准，本说明书仅对各项功能加以说明。

系统特征：➤ 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 或菜单编程
- 提供功能测试、交流参数测试和直流参数测试
- 可连接探针台、分选机等外接设备
- 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 路高压器件电压源
- 4 路偏置电源
- 1 路电压电流源/1 路参数测量单元
- 音频电压表/音频电压源

- 连接探针台、机械手的接口

ST3020 集成电路测试系统由 PMU/VIS 板、DPS 板、时钟图形板(PGCB)、通道板(CHB)、模拟功能板(AnalogB)、高压器件电压源(HDPS)、系统总线控制板(SBB)、计算机接口卡(PCI)等几部分组成,硬件结构的具体说明如下。

1. 1 测试板安装表

系统的测试板配置如下:系统内部提供 14 个插槽,每个插槽有特 定的使用功能,请正确安装。

表 1-1 测试板配置表

Solt1	Solt2	Solt3~10	Solt11	Solt12	Solt13	Solt14
HDPS	AnalogB	PMU	DPS	CHB1~8	PGCB	SBB

1. 2 PMU/VIS (参数测量单元/电压电流源)

系统提供一块参数测量单元(PMU/VIS),可以驱动、测量电压或电流,电压的驱动、测量范围为 $\pm 15\text{V}$,电流的驱动、测量范围为 $\pm 300\text{mA}$,PMU/VIS 板包括总线缓冲器,读写译码器,工作方式控制,继电器驱动电路,驱动电压电流值寄存器及数模转换器,箝位值寄存器及数模转换器,输出驱动器,反馈放大器,多路选择器,模数转换器等部分电路。

PMU/VIS 有以下两种工作方式:

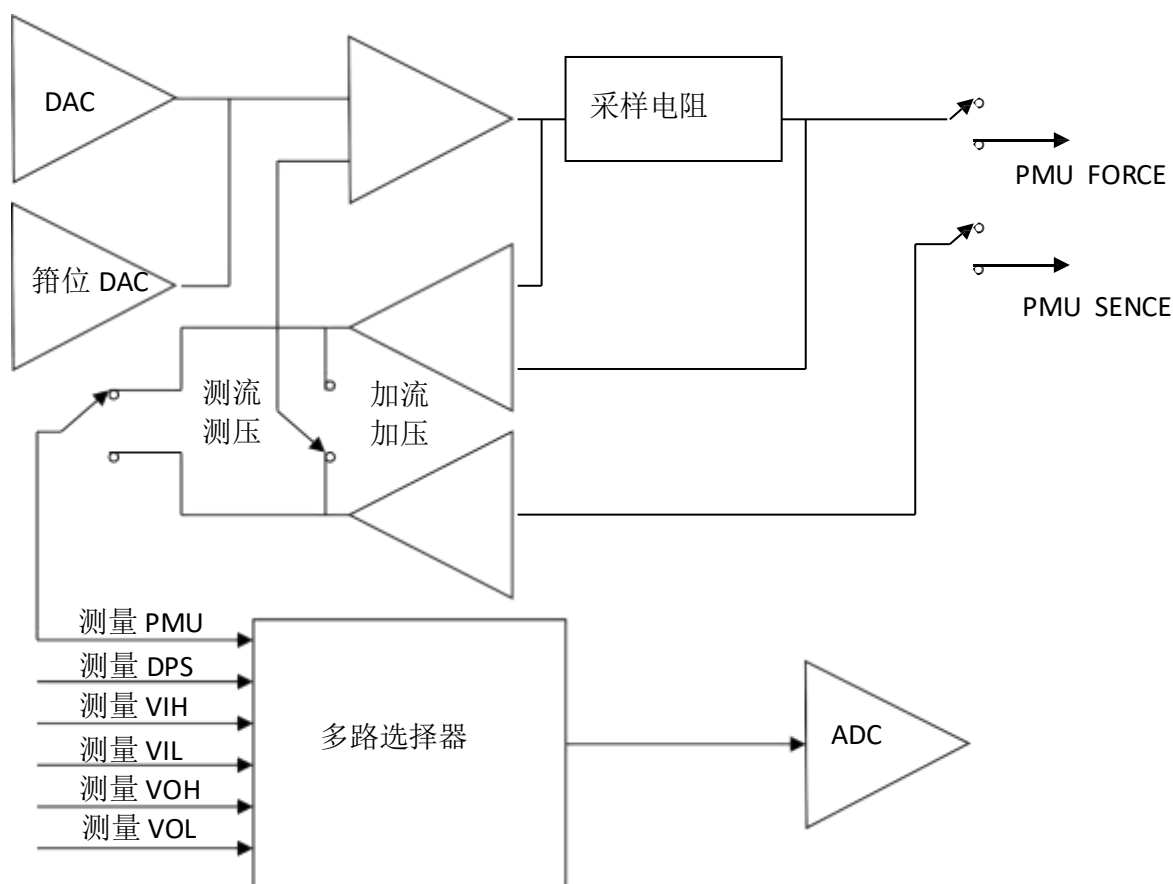
1. 加压测流(FVMI)方式

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值,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。



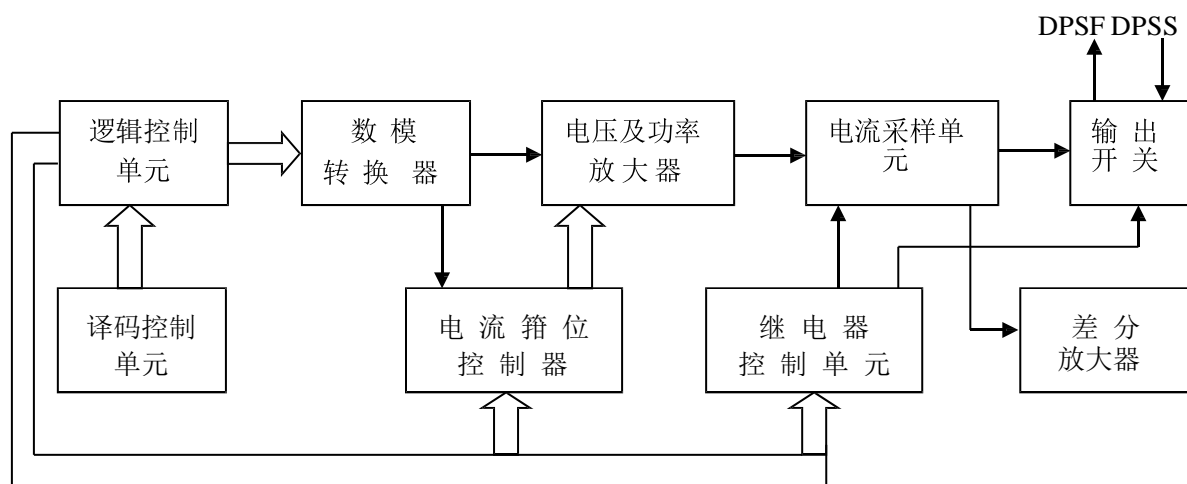
PMU 功能框图

1. 3 DPS（器件电压源）

主要功能：

- 1) 在测试过程中根据施加条件为被测器件提供器件电源，电压施加范围： $\pm 15\text{V}$ 。
- 2) 测量器件电源的工作电流，测量电流范围： $\pm 250\text{mA}$ 。

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、电流箱位控制器、电流采样单元、差分放大器、继电器控制单元。



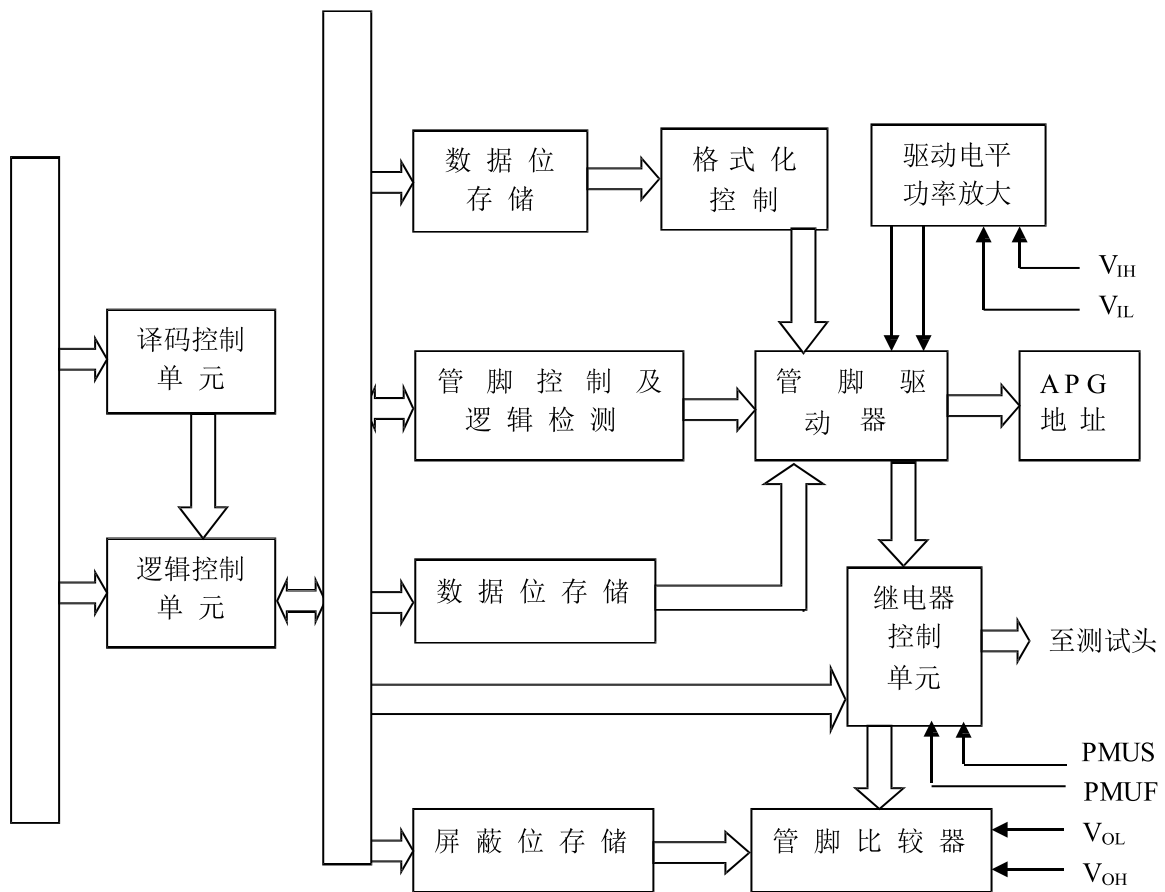
DPS 板逻辑框图

1. 4 数字通道板

主要功能：

- (1) 在功能测试过程中，直接与被测器件的 16 个输入、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，并与规定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。



通道板逻辑框图

ST3020 测试系统因为体积小最大可安装 8 块通道板，每块通道板有各自的地址码，通过拨码开关设置。在一台测试系统中，通道板地址码不能相同。用通道板上 3 位拨码开关（为 1、2、3）变化地址码，可选择 1 至 8 号通道。

1 2 3

on on on 选择 01 到 16 通道

off on on 选择 17 到 32 通道

on off on 选择 33 到 48 通道

off off on 选择 49 到 64 通道

on on off 选择 65 到 80 通道

off on off 选择 81 到 96 通道

on off off 选择 97 到 112 通道

off off off 选择 113 到 128 通道

每一块通道板提供 16 路测试通道，每路包括 1M Local Memory。在功能测试时，各通道独立施加输入电压、输出比较电平；共有四路：

输入高电平参考 V_{IH} 编程范围：0V~+15V

输入低电平参考 V_{IL} 编程范围：0V~+5V

输出高电平参考 V_{OH} 编程范围：0V~+15V

输出低电平参考 V_{OL} 编程范围：0V~+5V

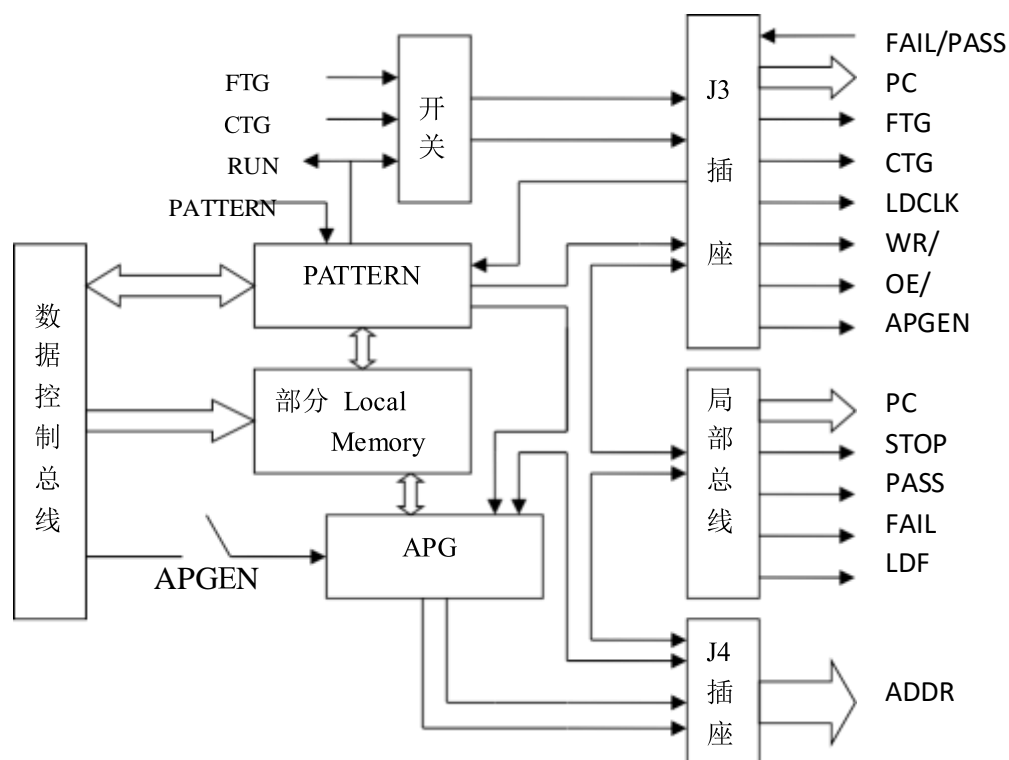
1.5 图形时钟板

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。PATTERN 包含：Local Memory 和 TPU（Test Processor Unit）两部分；其中 Local Memory 主要用于存储 Test Microinstruction，TPU 是实现 PATTERN 功能的执行部件；PATTERN 的功能主要用于控制 Local Memory 中图形执行的过程，同时将执行过程中的状态实时的反馈给主机。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



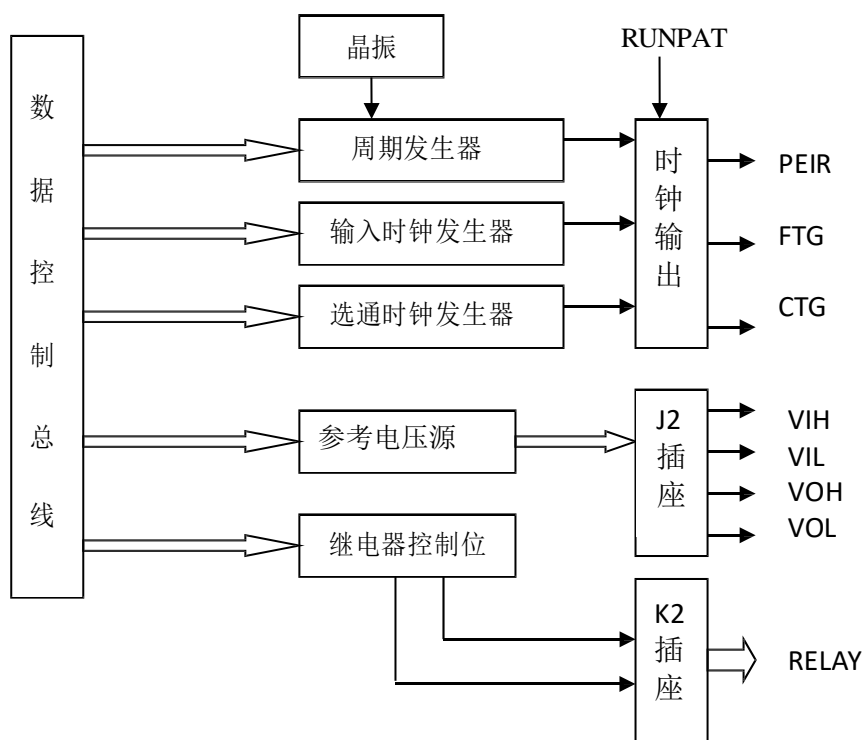
图形板框图

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：PERI → 50ns~10ms

FTG 前沿、宽度定时范围：50ns~10ms

选通脉冲 CTG 定时范围：50ns~10ms



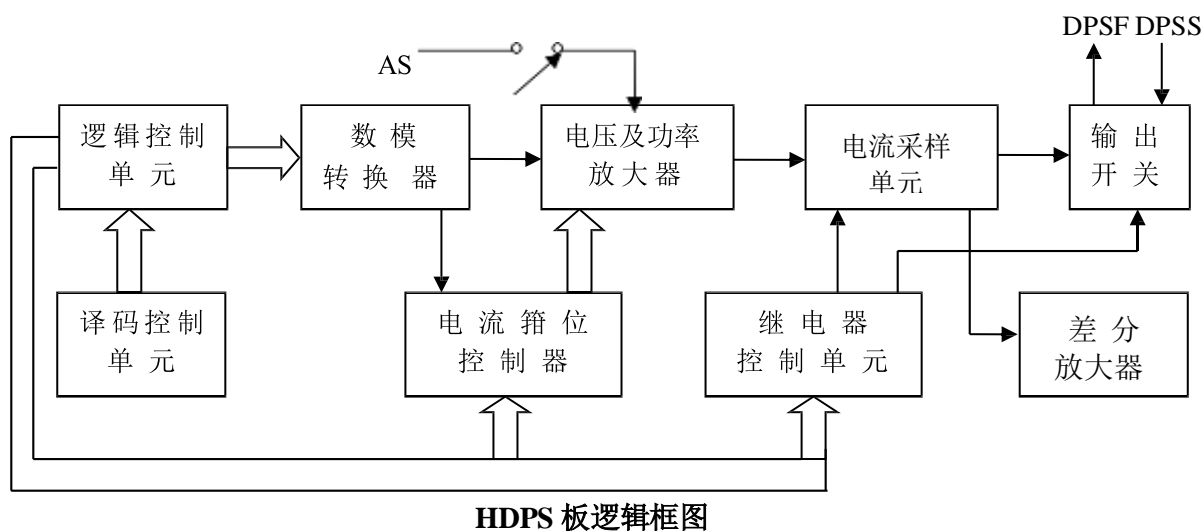
时钟板框图

1. 6 高压器件电压源(HDPS)

主要功能：

- 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，测量值最大为 5A。

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。



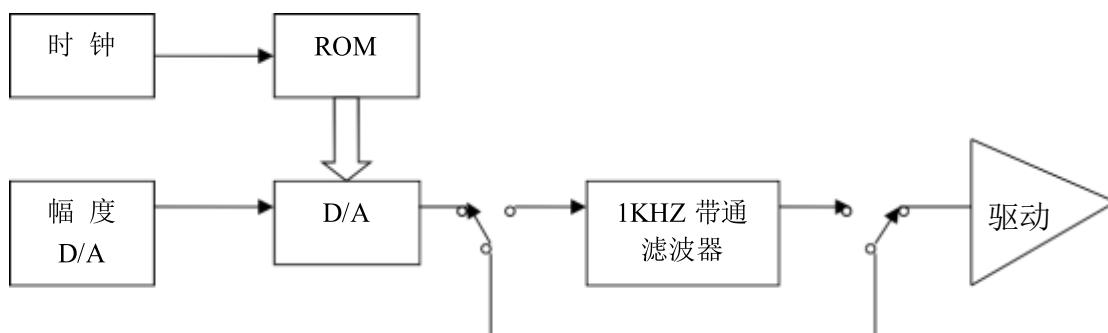
1.7 模拟板 (Analog)

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。



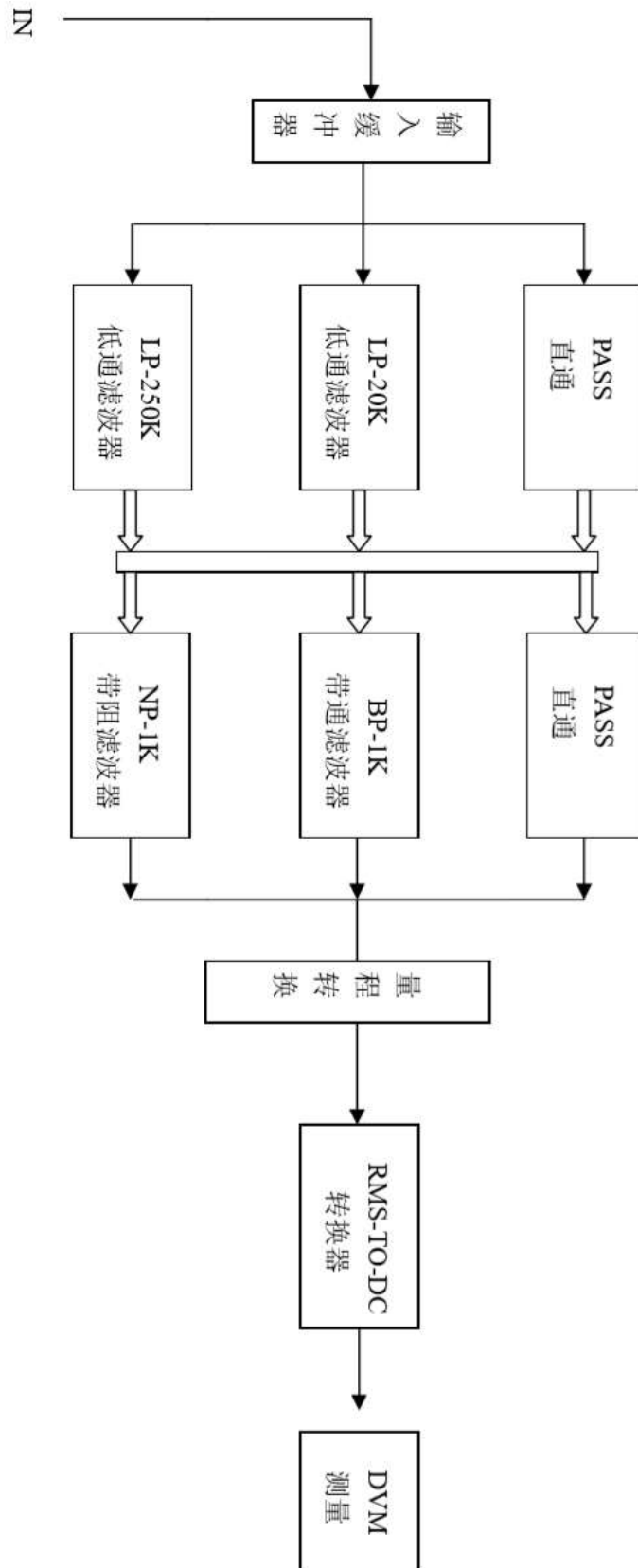
AS 逻辑框图

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~±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：

- (1) 由 T1 端输入，选择测量信号周期、脉冲宽度或斜率。
- (2) 由 T2 端输入，选择测量信号周期、脉冲宽度或斜率。
- (3) 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~17S

门限电压： 两路

电压范围： -10V~+10V

5)、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 $I_K \leq 50\text{mA}$ 。

1.8 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。计算机接口卡在测试系统内部通过 50 芯电缆，连接到系统总线控制板。

1.9 测试板

将各测试通道、DPS、AGND、控制继电器转接到测试板，再通过测试板上的插座与待测器件连接。

测试板各管脚定义如下：

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

J1↵	1↵	2↵	3↵
J1_1↵	CH48↵	CH48↵	DPS1F↵
J1_2↵	CH47↵	CH47↵	DPS1S↵
J1_3↵	CH46↵	CH46↵	AGND↵
J1_4↵	CH45↵	CH45↵	DPS2F↵
J1_5↵	CH44↵	CH44↵	DPS2S↵
J1_6↵	CH43↵	CH43↵	AGND↵
J1_7↵	CH42↵	CH42↵	AGND↵
J1_8↵	CH41↵	CH41↵	AGND↵
J1_9↵	CH40↵	CH40↵	AGND↵
J1_10↵	CH39↵	CH39↵	AGND↵
J1_11↵	CH38↵	CH38↵	↵
J1_12↵	CH37↵	CH37↵	↵
J1_13↵	CH36↵	CH36↵	↵
J1_14↵	CH35↵	CH35↵	↵
J1_15↵	CH34↵	CH34↵	↵
J1_16↵	CH33↵	CH33↵	K14↵
J1_17↵	CH32↵	CH32↵	K13↵
J1_18↵	CH31↵	CH31↵	K12↵
J1_19↵	CH30↵	CH30↵	K11↵
J1_20↵	CH29↵	CH29↵	K10↵
J1_21↵	CH28↵	CH28↵	K9↵
J1_22↵	CH27↵	CH27↵	K8↵
J1_23↵	CH26↵	CH26↵	K7↵
J1_24↵	CH25↵	CH25↵	K6↵
J1_25↵	CH64↵	CH64↵	K5↵
J1_26↵	CH63↵	CH63↵	K4↵
J1_27↵	CH62↵	CH62↵	K3↵
J1_28↵	CH61↵	CH61↵	K2↵
J1_29↵	CH60↵	CH60↵	K1↵
J1_30↵	CH59↵	CH59↵	↵
J1_31↵	CH58↵	CH58↵	+5V↵
J1_32↵	CH57↵	CH57↵	+5V↵

注： PMUF： PMU 施加信号

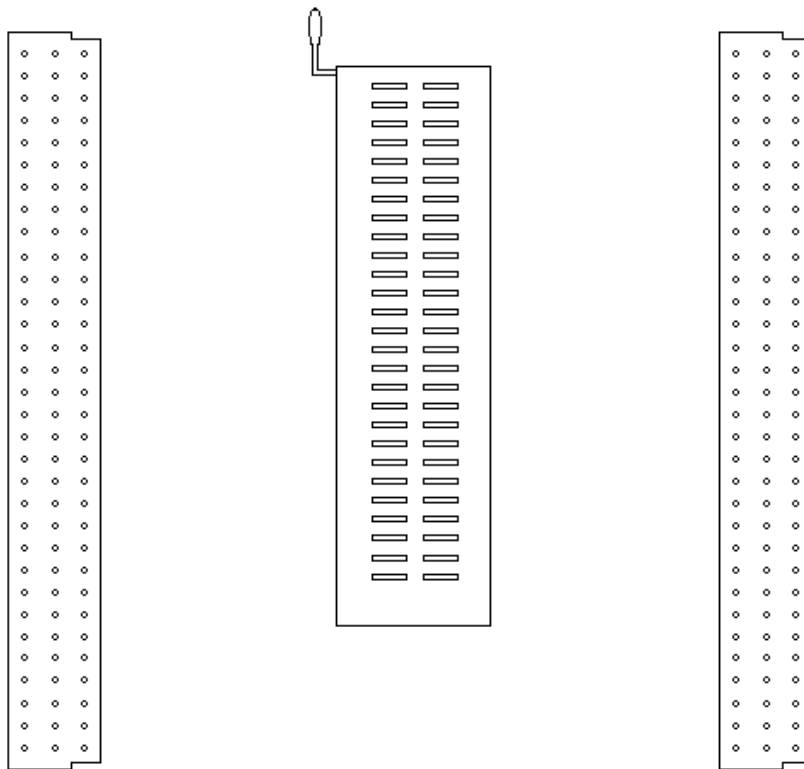
PMUS:： PMU 测量信号

K ： 表示继电器

+5 : 表示输出+5V 电源

±20V : 表示输出±20V 电源

测试板示图如下:



2. 系统环境

2.1 配置表

ST3020 集成电路测试系统包含以下配置及选件：

ST3020 集成电路测试系统配置表				
	名称	型号	数量	说明
必 选 配 置	系统基础外设	SYS	1	包含系统主机柜、插箱、系统底板、系统电源、主控计算机、显示器、系统软件等
	PCIE 接口板/网口	PCIE	1	装于计算机内，与系统主机通讯
	系统接口板	SBB	1	系统接口控制器
	参数测量单元	PMUB	1	提供一路精密直流测量单元(PMU)及一路电压电流源(VIS)功能
	测试台母板	DIB	1	系统输出信号分配板
数 字 配 件	器件电压源	DPSB	≤1	提供两路器件电压源
	系统时钟图形板	PGCB	≤1	提供系统时钟及 Pattern 地址
	数字通道板	CHB20-16	≤8	20MHz 数字通道板，每板 16 个数字通道
模 拟 配 件	高压器件电压源	HDPSB	≤1	提供两路高压器件电压源
	模拟功能板	AnalogB	≤1	提供一路音频信号源、一路音频电压表、两路时间测量单元、四路偏置电压源及 40 个继电器控制位
选 配 件	校准适配卡	CALCard	≤1	用于测试系统的自检、校准
	标准数字测试适配卡	DTHCard	≤1	通用 DIP40 测试适配卡
	标准模拟测试适配器	OPTAdapter	≤1	包含运算放大器、三端稳压器测试功能
	标准模拟测试适配卡	ATHCard	≤1	包括单运放、双运放、78 系列、79 系列测试适配卡
	ADDA 测试适配器	ADTAdapter	≤1	

2.2 工作环境

工作环境如下：

使用电压： 交流 220V±10%，50Hz±2%，功率： <1.5KW

操作温度： +23℃±3℃ 相对湿度： <70%

PC 配置： Intel Pentium 双核处理器

2G 以上内存

250G 以上硬盘空间 Windows XP/V7/V10 系统 Visual C++6.0 编程语言

电网电压： AC220V±10%；

电网频率： 50 Hz±5Hz

功率： 500W -1500W（不同配置功率不同）

湿度：30%~60%

3. 软件介绍

3.1 简介

ST3020 集成电路测试系统软件是在 WindowsXP /win7/win10 环境下，利用 Visual C++ 6.0 作为 系统开发工具开发的集成测试管理系统。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。

本系统有两种版本：

一、 用于支持编程方式开发测试程序的系统软件

ST3020 系统的测试程序编写也利用 Visual C++ 6.0 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。

采用此版本，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。

二、 用于支持菜单方式开发测试程序的系统软件

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。

3. 2 软件安装

3. 2. 1 文件构成

系统软件包括执行文件和动态连接库。

ST3020.exe	—————	系统执行主文件
ST3020Menu.exe	—————	菜单执行主文件
ToolForST3020.exe	—————	工具调用程序
user	—————	存放用户测试程序的默认目录
Califile	—————	存放校准数据，自检结果目录
*.Tst	—————	菜单编写的测试程序
*.mdc	—————	图形文件
*.mdv	—————	编译后的图形文件
*.log	—————	失效保存文件

3. 2. 2 安装测试系统软件

ST3020 测试系统软件被制作成安装程序包，自动进行安装。

3. 2. 3 安装 VC++

ST3020 集成电路测试系统是在 Microsoft Visual C++6.0 环境下开发，被测器件的测试程序库也是在 Microsoft Visual C++6.0 环境下开发。如果要开发或修改测试程序，必须安装 Microsoft Visual C++6.0。

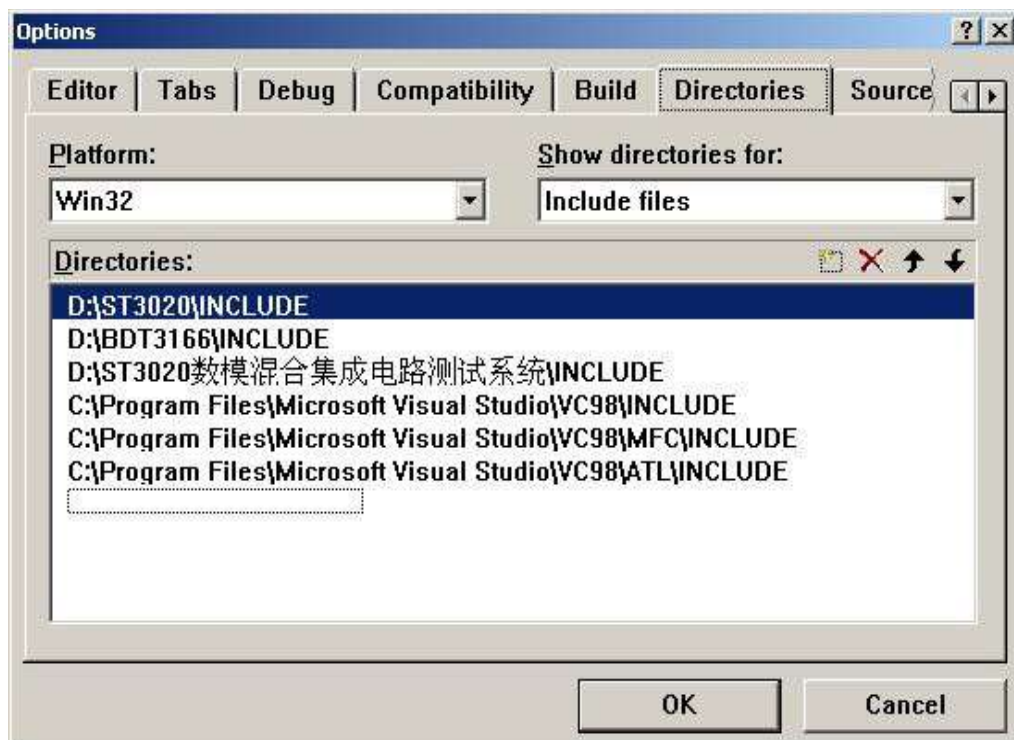
安装方法，按照 Microsoft Visual C++6.0 的安装向导自动进行安装。

3. 2. 4 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。

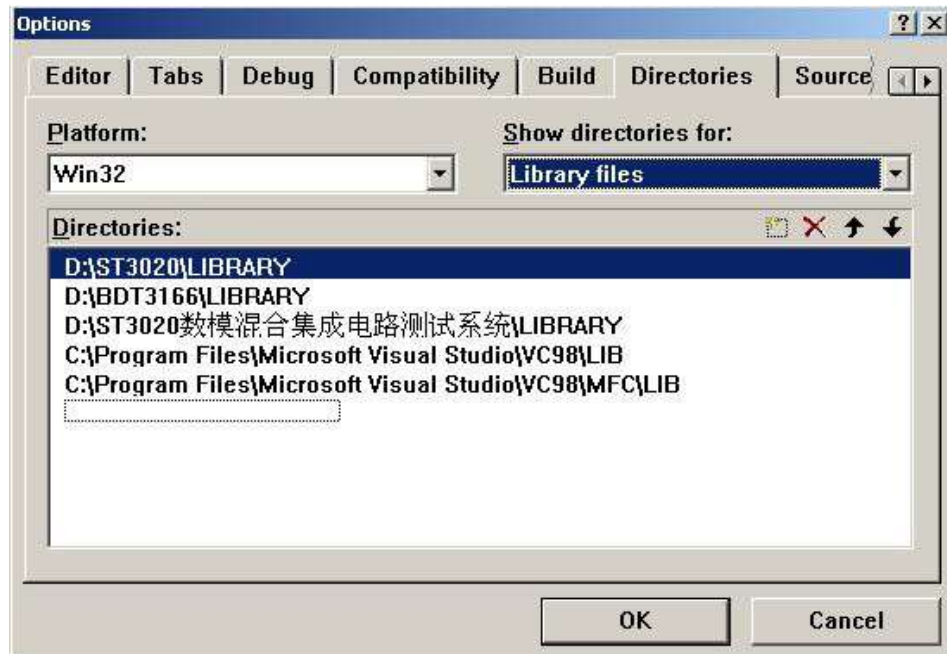
2. 设置 include 路径，单击菜单项 Tools，单击 Options...，弹出 Option 窗口，选择标签 Directories，对话框如下：



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

3. 2. 5 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=12.21

[校准文件路径]

校准文件路径=D:\ST3020\Califile

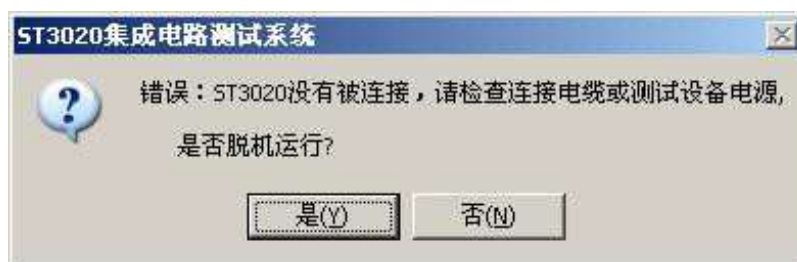
自检文件路径=D:\ST3020\Califile

1. 电阻值部分一般不改动，只有当数值与自检&校准板上的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。

4. 软件编辑方式及使用

ST3020 系统软件有四种运行模式：创建、图形编辑、设置、测试。

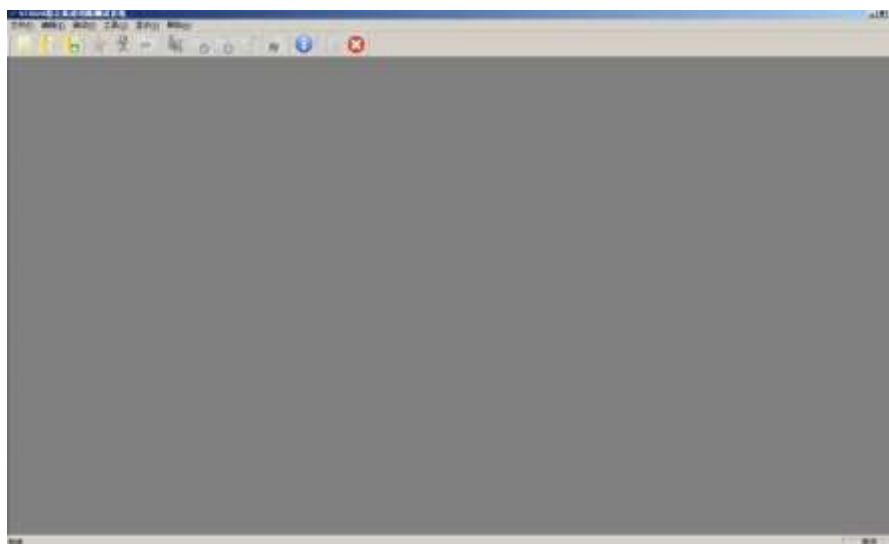
在 WINDOWS 下运行 ST3020.exe 文件，开始执行系统软件，如系统未开机或电缆连线有问题，将出现以下窗口：



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。



系统菜单栏

此窗口显示各种模式进入按钮，各种模式说明如下。

4.1 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。

4.2 图形编译

图形编译与查错程序 `ComplieDlg.exe` 文件是一个单独的执行程序，包括两个功能：



- a) 编译图形文件，提示错误信息；
- b) 标记图形运行的失效行及失效通道。

执行 CompieDlg.exe 文件，弹出如下对话框，具体功能如下：




编译图形对话框


4. 2. 1 编译图形

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。

4. 2. 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。

4. 3 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。

4. 3. 1 测试程序选择窗口

此窗口包括：**新建(N)**、**插入(I)**、**移出(E)**、“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。


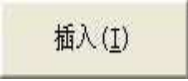

插入(I)：向“程序选择列表框”中加入测试程序。单击按钮，显示文件对话框，选择测试程序，文件扩展名为 .prg。进入文档窗口。

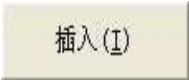
移出(E)：从“程序选择列表框”中移出测试程序选项，但不删除。


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。

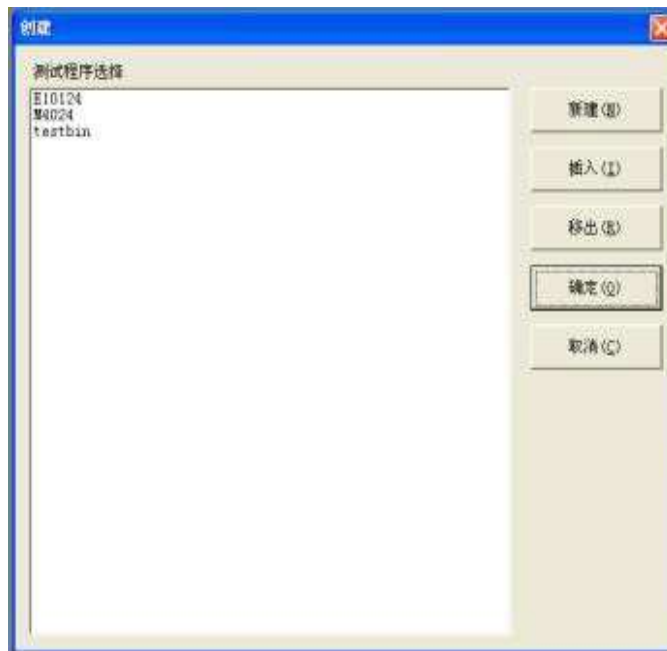
4. 4 测试（测试模式）

4. 4. 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”。

：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。

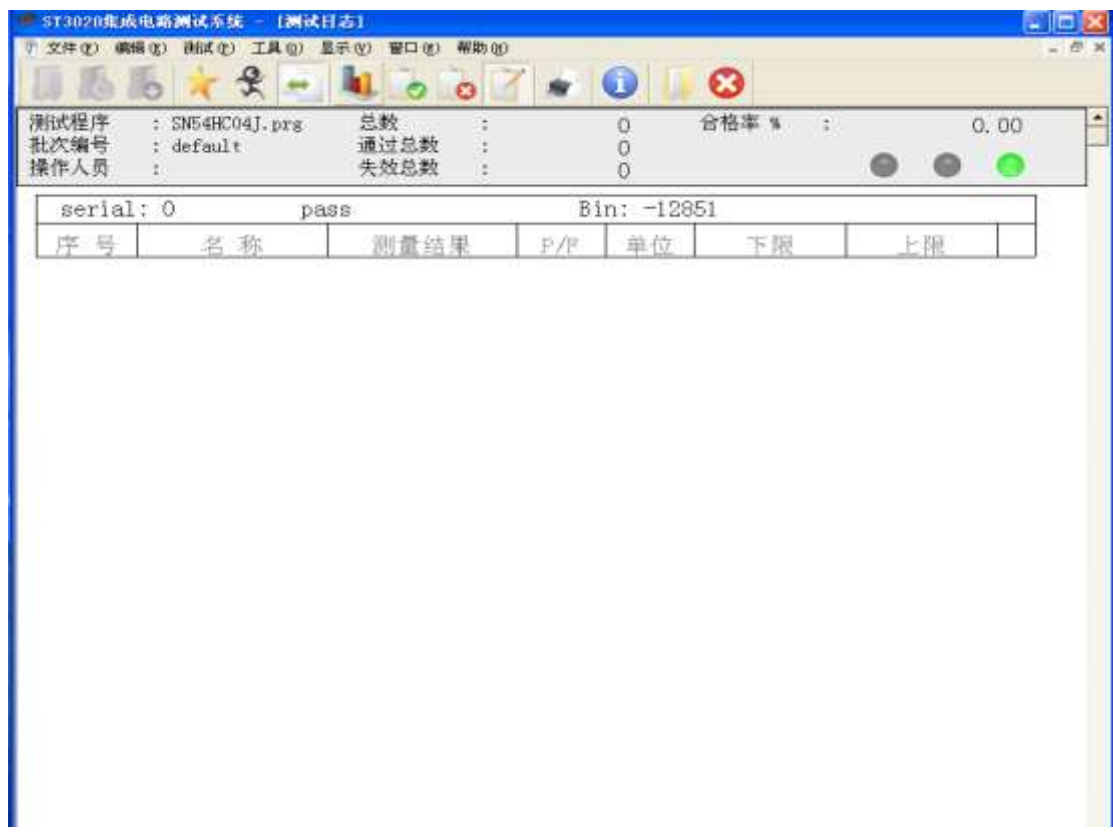
：从“程序选择列表框”中移出测试程序选项，但不删除。选择测试程序：在“程序选择列表框”中选中要用的测试程序，确定或双击，进入测试状态。



测试程序选择窗口


4. 4. 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。

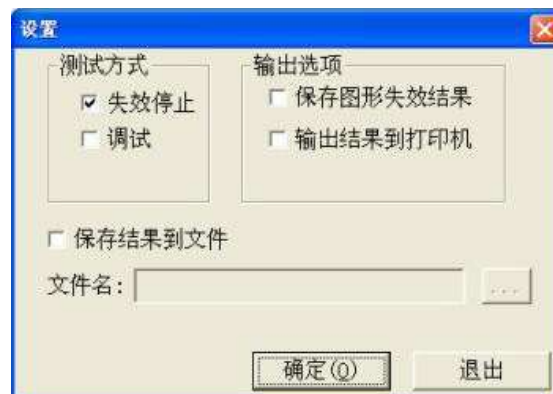


： 退出；快捷键 [F10]

4. 4. 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 。

4. 4. 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

4. 4. 5 数据显示说明：

测试程序	: LS7400.prg	总数	:	0	通过率 %	:	0.00
批次编号	:	通过总数	:	0			
操作人员	:	失效总数	:	0			
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			

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。

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

第二章 指令与函数

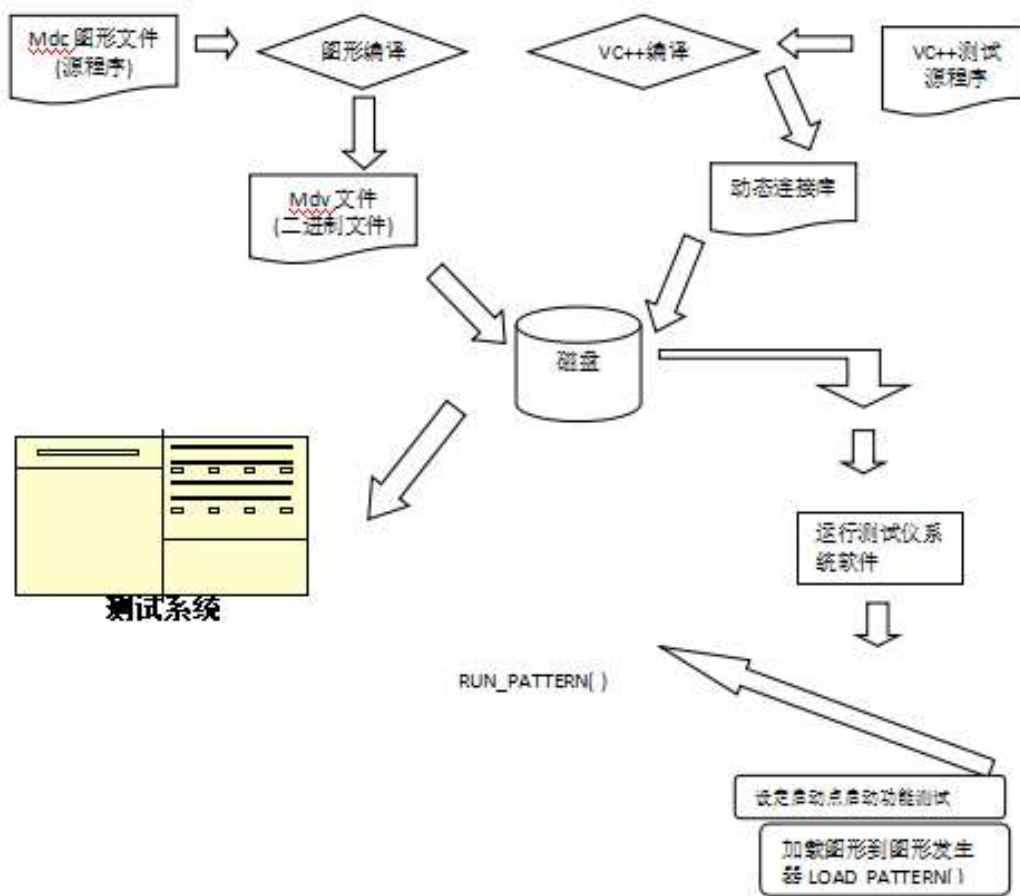
本系统使用 Visual C++ 6.0 作为开发测试程序的工具，测试程序以动态连接库的形式生成，由系统操作软件调用执行。

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。

➤ 测试程序：产生测试程序动态连接库文件。测试程序执行函数 `LOAD_PATTERN()` 函数，把目标图形文件装载到测试系统的存储器，设定好启动点，执行 `RUN_PATTERN()` 函数，启动测试。

1. 图形文件的生成、运行简介



图形文件生成、运行示意图

2. 管脚定义

PINDEF: ----- 管脚定义开始。

<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

[范例]:

OUTP = O , BIN , (9)

IN1= I , BIN , (10)

DATA(0..7) = IO, HEX , (8,7,6,5,4,3,2,1)

ADDR(0..5) = I , HEX , (12,13,14,18,19,20)

2. 1 管脚到通道定义

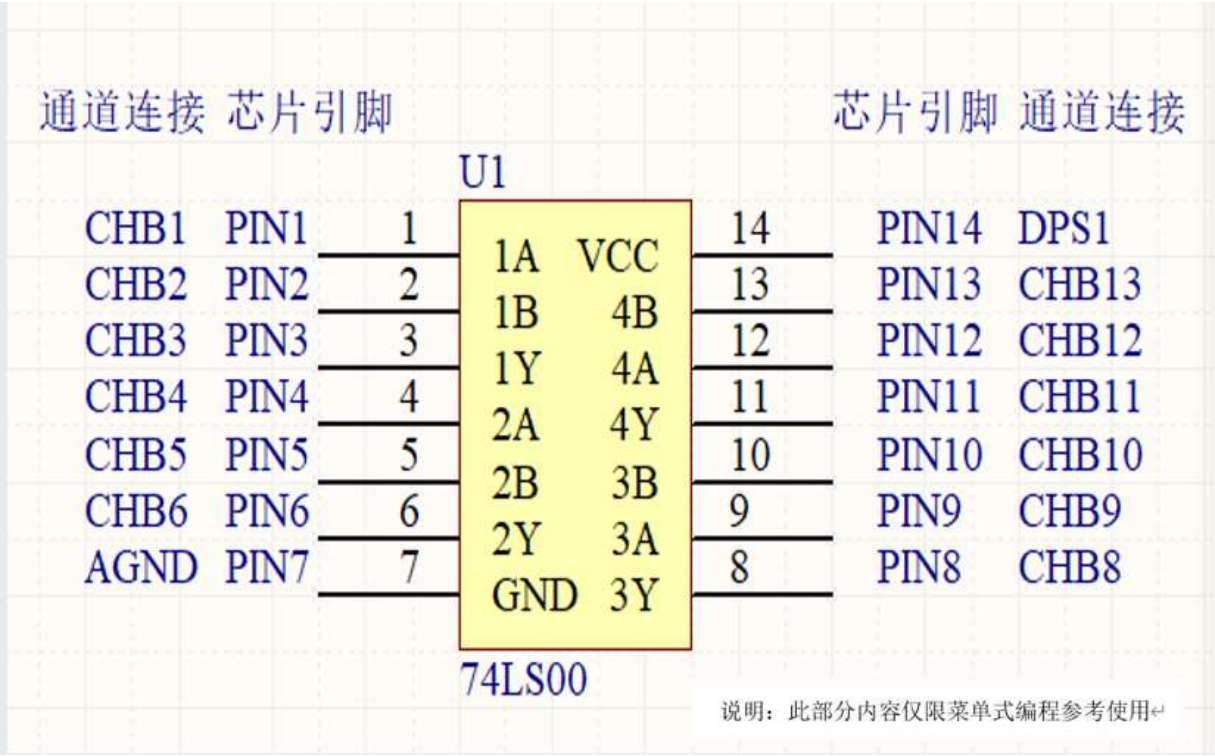
PIN_TO_CHANNEL:-----从第一列开始，在下一行开始编辑管脚与通道对应关系

[范例]:

1..6 = 1..6

8..13 = 42..47

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。



注意：必须一一对应，管脚数与通道数要相等。

3. 图形文件编写步骤

标准图形文件以.mdc 为扩展名，文件的第一行为“MEM_SOURCE_15; ”。

- 定义管脚 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

- 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

- 结束标志 以“END.”为开始标记，结束图形文件的编辑

- 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

4. 图形文件数据格式

0、1: 二进制方式，0 为输入低，1 为输入高

L、H: 二进制方式，L 为输出低，H 为输出高

T: 十六进制方式，表示后面数据为输出

X: 表示对应的通道不驱动，不测试

[范例] m74ls32.mdc 图形文件

MEM_SOURCE_15;

PINDEF

INPUT(1..8) = I, HEX, (1,2,4,5,43,44,46,47)

OUT(1..4) = O,HEX, (3,6,42,45)

PIN_TO_CHANNEL

1..6 = 1..6

8..13 = 42..47

MAIN_F

START_INDEX(0)

INC (00 T0)

INC (55 TF)

INC (AA TF)

INC (FF TF)

HALT (FF TF)

START_INDEX(1)

INC (00 TX)

HALT (00 TX)

START_INDEX(2)

INC (FF TX)

HALT (FF

TX)

END.

5. 图形文件格式

源文件：ASCII 文件 .mdc 文件

5.1 初始指定

MEM_SOURCE_15; ----- 第一行必须以“MEM_SOURCE_15;”开始，作为图形文件的标志

5.2 指令格式

MAIN_F: ----- 必须从第一列开始，图形指令及数据段开始标志，表示下面的语句是指令和数据的。

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。

指令 (图形)

标号 指令 (图形)

指令，参数 (图形)

标号 指令，参数 (图形)

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

5. 3 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 Visual C++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。

5. 4 图形结束

END. ----- 必须从第 1 列开始，标志图形文件结束。

5. 5 注释

{ } ----- 注释一行
// ----- 注释一行或在一条指令的结尾注释

6. 图形文件指令

6. 1 指令集

序号	指令	注释
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, 标号	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, 标号	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，标号	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, 把当前图形中地址管脚对应的 F 数据装载到 内部地址计数器 1 中。
10	LDAR2, n	地址计数器 2 装载, 把当前图形中地址管脚对应的 F 数据装载到 内部地址计数器 2 中。
11	ADDR1	当前图形送出后把地址输出取地址计数器 1 的值
12	ADDR2	当前图形送出后把地址输出取地址计数器 2 的值
13	INCAR1	地址计数器 1 加 1, 当前图形送出后把内部地址计数器 1 加 1
14	INCAR2	地址计数器 2 加 1, 当前图形送出后把内部地址计数器 2 加 1
15	DECAR1	地址计数器 1 减 1, 当前图形送出后把内部地址计数器 1 减 1
16	DECAR2	地址计数器 2 减 1, 当前图形送出后把内部地址计数器 2 减 1
<p>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。</p> <p>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。</p>		

6. 2 指令用法与范例

6.2.1 INC

[功能]

顺次走一步，执行一次图形

6.2.2 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)

6.2.3 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

6.2.4 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)
```

6.2.5 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。

[范例]

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
A INC	(0	XXXX)
GONP, A	(1	LLLL)

INC	(0	LLLL)
INC	(1	LLLH)
INC	(0	LLLH)
INC	(1	LLHL)

6.2.6 LDF

[功能]

动态测量时的标志，与 **JMP** 配合使用

6.2.7 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)

```

6.2.8 HALT

[功能]

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**

[范例]

```

START_INDEX(1)
INC          (0001 HLLL)
HALT         (0001 HLLL)

```


6.2.9 LDAR1, n

[功能]

地址计数器 1 装载，把当前图形中地址管脚对应的 F 数据装载到 内部地址计数器 1 中。

6.2.10 LDAR2, n

[功能]

地址计数器 2 装载,把当前图形中地址管脚对应的 F 数据装载到 内部地址计数器 2 中。

6.2.11 ADDR1

[功能]

当前图形送出后把地址输出取地址计数器 1 的值

6.2.12 ADDR2

[功能]

当前图形送出后把地址输出取地址计数器 2 的值

范例参考第五章示例。

6.2.13 INCAR1

[功能]

地址计数器 1 加 1，当前图形送出后把内部地址计数器 1 加 1

6.2.14 INCAR2

[功能]

地址计数器 2 加 1，当前图形送出后把内部地址计数器 2 加 1

6.2.15 DECAR1

[功能]

地址计数器 1 减 1，当前图形送出后把内部地址计数器 1 减 1

6.2.16 DECAR2

[功能]

地址计数器 2 减 1，当前图形送出后把内部地址计数器 2 减 1

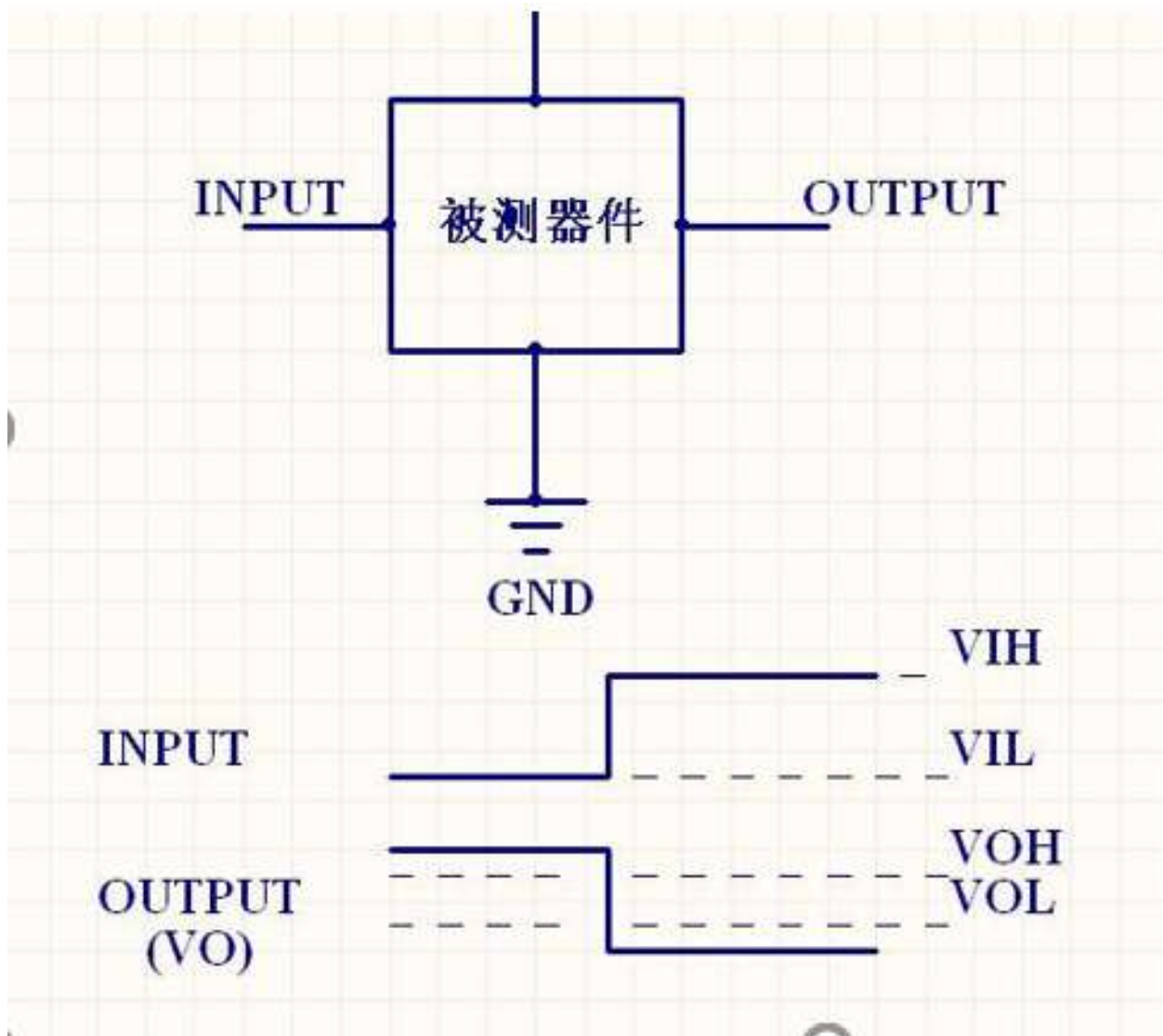
6. 3 图形文件指令解读

6.3.1 指令说明

SET_INPUT_LEVEL(double Vih ,double Vil); //设置全部管脚输入驱动电平值。

SET_OUTPUT_LEVEL(double Voh, double Vol); //设置所有管脚输出比较电平。

如下图所示，设置被测器件输入高电平 **V_{IH}**（“1”），输入低电平 **V_{IL}**（“0”）。被测器件的输出为 **V_O**，如果 **V_O>V_{OH}**,则测试系统判定其为高电平；如果 **V_O<V_{OL}**,则测试系统判定其为低电平。

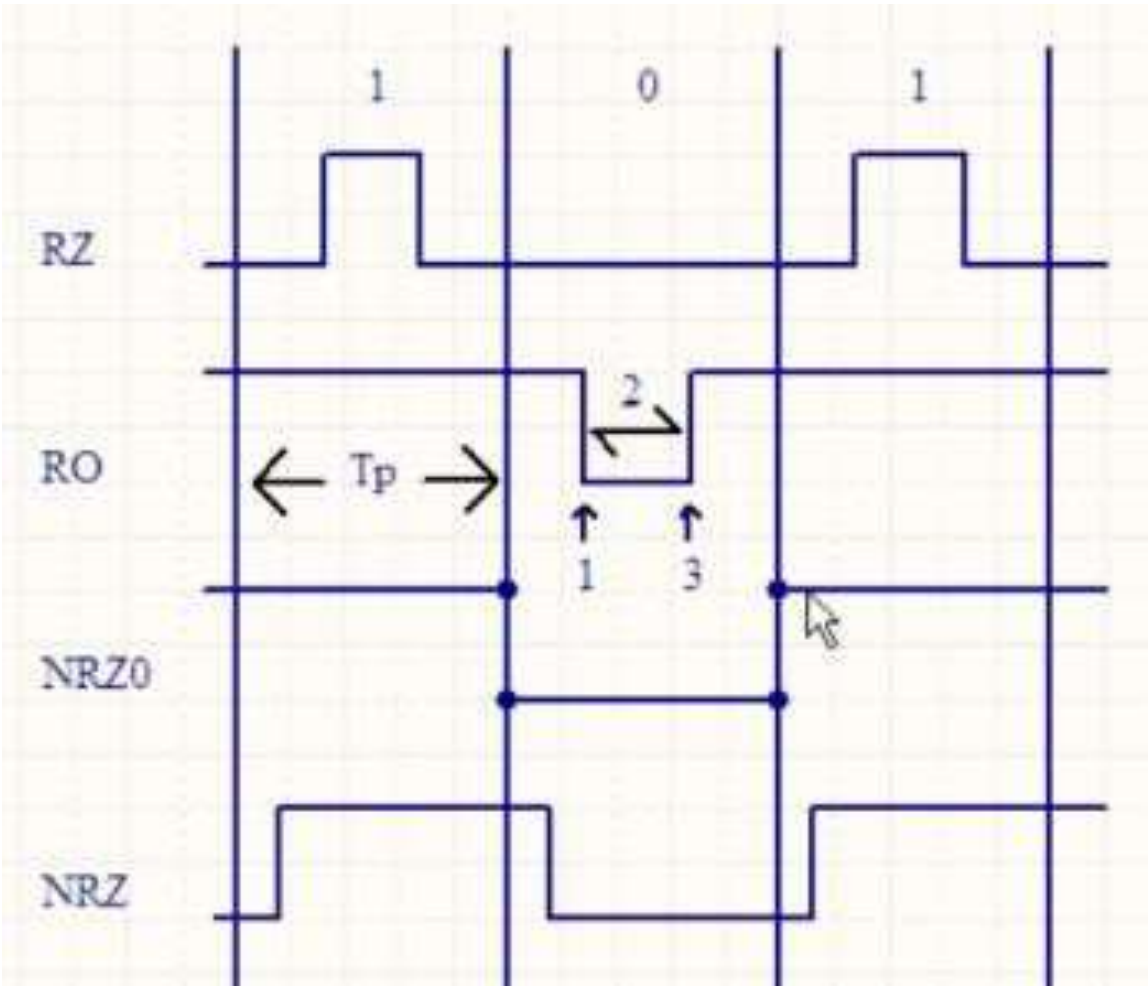


6.3.2 波形驱动格式说明

ST3020 集成电路测试系统可以对每一个管脚设置驱动波形，当设置某一个管脚时，不改变其他管脚的格式。ST3020 测试系统支持四种波形驱动格式：

NRZ	不归零方式
NRZ0	满周期
RO	归一格式
RZ	归零格式

详细波形格式如下图：



```
SET_TIMING(1,2,3); //1: 前沿      2.脉宽  3.选通点
SET_PERIOD(Tp); //Tp: 周期
```

7. 测试程序编写函数

ST-3020 测试系统提供的用户指令共 22 条，用户可根据顺序查询详细说明。

ST-3020 数字集成电路测试系统通常包含以下板卡：PMUB、DPS（器件电压源）、CHB（数字通道板）、PGCB（图形时钟板）、SBB（系统总线控制板）。

7. 1 函数目录

序号	函数	注释
1	PMU_CONDITIONS()	设置 PMU 测量条件
2	PMU_MEASURE()	PMU 测量

3	SET_DPS()	设置 DPS 条件
4	DPS_MEASURE()	DPS 测量
5	DPS_OFF()	关闭 DPS
6	FORMAT()	格式化波形
7	SET_INPUT_LEVEL()	设置输入参考电平
8	SET_OUTPUT_LEVEL()	设置输出参考电平
9	RUN_PATTERN()	执行一段图形
10	SET_PERIOD()	设置时钟周期
11	SET_TIMING()	设置时钟
12	SET_RELAY()	闭合指定继电器，其他继电器打开
13	CLOSE_RELAY()	闭合指定继电器，其他保持原状态
14	CLEAR_RELAY()	清继电器
15	BIN ()	设置分箱号
16	SHOW_RESULT()	显示测试结果到显示设备
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	CLEAR_ALL()	总清
20	SET_PMU()	设置 PMU
21	CLEAR_PMU()	清 PMU
22	Delay ()	延时（单位：ms）

7. 2 函数用法与范例

7.2.1 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。箝位值作为限流或限压的保护值，并且根据箝位值自动选择测量量程。

[参数]

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：±15V

电流：±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 ~15V

电流：0 ~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

[范例]

设置 PMU 测量模式为加压测流方式，施加电压 5V，箝位电流 10mA。

PMU_CONDITIONS (FVMI, 5, V, 10, mA)

7.2.2 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]

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]

用 PMU_CONDITIONS 函数设置 PMU 测量条件后，测试名为“con”，用 PMU 测量

1, 3, 7, 8, 9 通道上器件管脚的电流，上限 -0.1mA，下限-1.9mA，延时 10ms.

PMU_CONDITIONS (FVMI, 5, V, 10, mA)

PMU_MEASURE (“1,3,7-9”,10,”con”,mA,-0.1,-1.9)

加压测流，测量通道 3 上器件管脚的电流，PMU_CONDITIONS 设置 PMU 测量条件，PMU_MEASURE 函数测量通道 3 上器件管脚的电流，延时 10ms，函数返回值为测量值。

注意：默认电流流入被测器件的方向为正，流出为负。

```
double    Result;
```

```
PMU_CONDITIONS (FVMI, 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

7.2.3 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]

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: $\pm 15V$

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]

设置第一路 DPS, 施加电压 5.5V, 限流 20uA

SET_DPS (DPS1, 5.5, V, 20, UA) 或

SET_DPS (1, 5.5, V, 20, UA)

7.2.4 DPS_MEASURE ()

[函数原形]

BOOL DPS_MEASURE (BYTE No, BYTE IRange,
unsigned int Delayms, CString csItem,
CString csUnit, double fIccMax,
Double fIccMin);

Double DPS_MEASURE (BYTE No, BYTE IRange, unsigned int Delayms);

[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]

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]

使用第一路 DPS 测量电流，选用 20mA 档测量，延时 10ms，测量电流值上限最大值 10mA，下限最大值 2mA，测量值单位 mA，测量项目名称“ICC”，返回值为 PASS 或 FAIL。

DPS_MEASURE (DPS1, R20MA, 10, “ICC”, mA, 10, 2)

直接测量电流值，使用 DPS1 测量电流，使用 20mA 档测量，延时 5ms，测量结果返回电流值

double Result;

Result = DPS_MEASURE (DPS1, R20MA, 5);

7.2.5 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]

No : DPS 选择

DPS1 ----- 第一路

DPS2 ----- 第二路

[范例]

将第二路 DPS 关闭

DPS_OFF (DPS2)

7.2.6 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]

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，最多 128 个通路, 通道编号 1-128

[范例]

格式化通道 1, 2, 3, 4, 7 为 NRZ0 方式，再格式化通道 1, 3, 6 为 RZ 方式

FORMAT (NRZ0, “1-4,7”)

FORMAT (RZ, “1,3,6”)

7.2.7 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]

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]

设置全部管脚输入电平, 低电平 0.8 V, 高电平 3V

```
SET_INPUT_LEVEL (3, 0.8)
```

7.2.8 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]

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]

设置输出比较电平, 高电平 5V, 低电平 3V

```
SET_OUTPUT_LEVEL (5, 3)
```

7.2.9 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]

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 ----- 失效后继续执行

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。用于 GONP 执行匹配时使用。0 表示时间范围不限制。

[返回值]

返回失效值 0: 对应通道 FAIL

1: 对应通道 PASS

[范例]

执行图形，索引号为 0，失效返回，不使用 APGEN，不设运行时间范围

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

执行功能测量，测试项名称“Function”，索引号为 0，失效返回，不使用 APGEN 方式，不设运行时间范围

`RUN_PATTERN (“Function” 0, 1, 0, 0)`

注：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

7.2.10 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]

设置时钟周期为 500ns

`SET_PERIOD (500)`

7.2.11 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，设定波形格式开始时间，单位 ns，范围：大于 10ns

EndEgde: 后沿，设定波形的宽度，单位 ns，范围：大于 20 ns

Ctg: 选通，设定测量取样时间，单位 ns，范围：大于 10ns

[范例]

设置时间前沿为 30ns，后沿 100ns，选通为 120ns

```
SET_TIMING (30,100,120 )
```

7.2.12 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”);
```


7.2.13 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”);

7.2.14 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”);

7.2.15 BIN ()

用于测试完成后的分箱处理。

[函数原形]

```
Void BIN (BYTE bin)
```

[参数]

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) ;
```

7.2.16 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)
```

7.2.17 SHOW_RESULT()

```
void SHOW_RESULT (CString csItem, double measure_value,  
                  CString csUnit,  
                  double fUpLimit,
```

Double fDnLimit)

显示测试结果，7.2.16 单位为标准单位 A，MA，V，MV，7.2.17 可定义非标准单位。

7.2.18 SHOW_RESULT(CString csItem BOOL bTest)

显示功能测量结果，

[参数]

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

7.2.19 CLEAR_ALL()

总清系统。

7.2.20 SET_PMU()

[函数原形]

```
void SET_PMU(unsigned int Mode, double Value, unsigned int Value_Unit, double
Clamp_Value, unsigned int Clamp_Unit)
```

[参数]

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: $\pm 15\text{V}$,

电流: $\pm 250\text{mA}$

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: $0 \sim 15\text{V}$

测流范围: $0 \sim 250\text{mA}$

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

7.2.21 CLEAR_PMU()

清 PMU

7.2.22 Delay()

设置延迟时间 (单位: ms)

[函数原形]

void Delay(double fMs)

[参数]

fMs: 延迟时间, 单位: ms, double 型

[范例]

Delay(0.1);; 延迟 100us。

第三章 器件手册信息提取

器件手册↵	1.对应图形文件模块提取手册中的信息：功能描述（文字、真值表、时序图等）↵ 2.对应测试程序模块提取手册中的信息：测试参数表（项目名称、测试条件、限值标准等）↵
任务↵	编写测试方案↵ 器件说明，可测指标、测试方法、图形文件编译、程序文件编写、测试结果、报告撰写↵

不同器件不同方法取不同信息

第四章 测试程序开发

1. 测试程序开发实例

1. 1[实例一] 62256 存储器

62256.mdc 图形文件部分源程序:

MEM_SOURCE_15;

PINDEF

ADDR(1..15) = I, HEX, (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)

DATA(1..8) = IO, HEX, (17,18,19,35,36,37,38,39)

CS = I, BIN, (40)

OE = I, BIN, (42)

WE = I, BIN, (47)

PIN_TO_CHANNEL

10..3 = 1..8

25 = 9

24 = 10

21 = 11

23 = 12

2= 13

26 = 14

1= 15

20 = 40

22 = 42

27 = 47

11..13 = 17..19

15..19 = 35..39

MAIN_F

START_INDEX(0)

INC	(0000 00 0 1 1)	
LDAR1,0	(0000 00 0 1 1)	
ADDR1	(0000 00 0 1 1)	
LDC,7	(0000 00 0 1 1)	//循环块 D 中的内容执行 8 次, 2 的 3 次方
D INC	(0000 00 0 1 1)	
LDC,4095	(0000 xx 0 1 1)	//循环块中的内容执行 4096 次 2 的 12 次方,
E INCAR1	(0000 FF 0 1 0)	由于 LOOP 循环 嵌套, E 中内容共循环次数
INC	(0000 FF 0 1 1)	2 的 15 次方
INC	(0000 TFF 0 0 1)	//T: 十六进制方式, 表示后面的数据为输出
LOOP,E	(0000 xx 0 1 1)	
INC	(0000 00 0 1 1)	
LOOP,D	(0000 00 0 1 1)	
INC	(0000 00 0 1 1)	
LDAR1,0	(0000 00 0 1 1)	
ADDR1	(0000 00 0 1 1)	
LDC,7	(0000 00 0 1 1)	
F INC	(0000 00 0 1 1)	
LDC,4095	(0000 xx 0 1 1)	
G INCAR1	(0000 00 0 1 0)	
INC	(0000 00 0 1 1)	
INC	(0000 T00 0 0 1)	
LOOP,G	(0000 xx 0 1 1)	
INC	(0000 00 0 1 1)	
LOOP,F	(0000 00 0 1 1)	

INC	(0000 00 0 1 1)
HALT	(0000 00 1 1 1)

S

START_INDEX(1)

INC	(0000 00 0 1 1)
INC	(0000 00 0 1 0)
INC	(0000 00 0 1 1)
INC	(0000 Txx 0 0 1)
HALT	(0000 Txx 0 0 1)

START_INDEX(2)

INC	(0000 00 0 1 1)
INC	(0000 FF 0 1 1)
INC	(0000 FF 0 1 0)
INC	(0000 FF 0 1 1)
INC	(0000 Txx 0 0 1)
HALT	(0000 Txx 0 0 1)
END.	

1. 2 [实例二] 74HC14

MEM_SOURCE_15;

PINDEF

I1 =I,BIN,(1)

O1=O,BIN,(2)

I2 =I,BIN,(3)

O2=O,BIN,(4)

I3 =I,BIN,(5)

O3=O,BIN,(6)

O4=O,BIN,(42)

I4 =I,BIN,(43)

O5=O,BIN,(44)

I5 =I,BIN,(45)

O6=O,BIN,(46)

I6 =I,BIN,(47)

PIN_TO_CHANNEL

1..6 = 1..6

8..13 = 42..47

MAIN_F

START_INDEX(0) // FUNCTION TEST

{IOIOIOOIOIOI}

{112233445566}

INC (XXXXXXXXXXXXXX)

INC (0H0H0HH0H0H0)

INC (1L1L1LL1L1L1)

INC (0H0H0HH0H0H0)

INC (1L0H0HH0H0H0)

INC (0H1L0HH0H0H0)

INC (0H0H1LH0H0H0)

INC (0H0H0HL1H0H0)

INC (0H0H0HH0L1H0)

INC (0H0H0HH0H0L1)

INC (1L1L1LL1L1L1)

INC (0H1L1LL1L1L1)

INC (1L0H1LL1L1L1)

INC (1L1L0HL1L1L1)

INC (1L1L1LH0L1L1)

INC (1L1L1LL1H0L1)

INC (1L1L1LL1L1H0)

INC (XXXXXXXXXXXXXX)

HALT (XXXXXXXXXXXXXX)

START_INDEX(1)// POSITIVE/NAGATIVE-GOING VOLTAGETEST

INC (XXXXXXXXXXXXX)

INC (0X0X0XX0X0X0)

INC (0X0X0XX0X0X0)

INC (0X0X0XX0X0X0)

INC (1X1X1XX1X1X1)

INC (1X1X1XX1X1X1)

HALT (1X1X1XX1X1X1)

START_INDEX(2)//OUTPUT_H INPUT_L

INC (0X0X0XX0X0X0)

INC (0X0X0XX0X0X0)

INC (0X0X0XX0X0X0)

HALT (0X0X0XX0X0X0)

START_INDEX(3)//OUTPUT_L INPUT_H

INC (1X1X1XX1X1X1)

INC (1X1X1XX1X1X1)

INC (1X1X1XX1X1X1)

HALT (1X1X1XX1X1X1)

START_INDEX(4)// POSITIVE/NAGATIVE-GOING VOLTAGETEST

INC (1L1L1LL1L1L1)

INC (1L1L1LL1L1L1)

INC (1L1L1LL1L1L1)

INC (1L1L1LL1L1L1)

HALT (1L1L1LL1L1L1)

END.

测试程序：

//M74HC14 测试程序

#include "StdAfx.h"

#include "testdef.h"

#include "data.h"

void PASCAL M74HC14()

{

int i;

double VT, V_2, V_4, V_6, VH, V_OUT;

CString str;

int re;

// 连接性测试

SET_DPS(1,0,V,20,MA);//已知 VCC 连接 DPS，设置 VCC 为 0V

PMU_CONDITIONS(FIMV,-0.1,MA,2,V);//设置 PMU，施加-0.1M 电流，箝位 2V

if(!PMU_MEASURE("1-6,42-47",5,"CON",V,-0.1,-1.9))//PMU 测试及显示

BIN(1);

// 功能测试

SET_DPS(1,2,V,150,MA);

SET_INPUT_LEVEL(1.5,0.3);//设置输入驱动电平

SET_OUTPUT_LEVEL(1.9,0.1);//设置输出比较电平

SET_PERIOD(3000);//设置周期

SET_TIMING(100,1000,2000);//前沿，脉宽，选通点

FORMAT(NRZ0,"1,3,5,9,11,13");//设置指定通道波形格式为 NRZ0

if(!RUN_PATTERN("FUN_2",0,1,0,0))//运行图形文件（0），功能测试

BIN(2);

```

        // ICC 测试
SET_DPS(1,6,V,150,MA);
SET_INPUT_LEVEL(6,0);
SET_OUTPUT_LEVEL(5,1);
RUN_PATTERN(2,1,0,0);
if(!DPS_MEASURE(1,R20UA,5,"ICC",UA,2,No_LoLimit))
    BIN(3);
SET_INPUT_LEVEL(0,0);
// VOH 测试
// VOH_2V 测试
SET_DPS(1,2,V,150,MA);
SET_INPUT_LEVEL(2,0);
RUN_PATTERN(2,1,0,0);
PMU_CONDITIONS(FIMV,-0.02,MA,3,V);
if(!PMU_MEASURE("2,4,6,42,44,46",5,"VOH2_",V,No_UpLimit,1.9))
    BIN(4);
// VOL 测试
// VOL_2V 测试 SET_DPS(1,2,V,150,MA);
SET_INPUT_LEVEL(2,0);
RUN_PATTERN(3,1,0,0);
PMU_CONDITIONS(FIMV,0.02,MA,0.3,V);
if(!PMU_MEASURE("2,4,6,42,44,46",5,"VOL2_",V,0.1,No_LoLimit))
    BIN(5);

```

// IIN 测试

RUN_PATTERN(2,1,0,0);

PMU_CONDITIONS(FVMI,0,V,0.002,MA);

if(!PMU_MEASURE("1,3,5,43,45,47",5,"IIL_",MA,No_UpLimit,-0.0001))

 BIN(6);

RUN_PATTERN(3,1,0,0);

PMU_CONDITIONS(FVMI,6,V,0.002,MA);

if(!PMU_MEASURE("1,3,5,43,45,47",5,"IIH_",MA,0.0001,No_LoLimit))

 BIN(6);

SET_INPUT_LEVEL(0,0);

//THRESHLD VOLTAGE POSITIVE-GOING TEST (+2V)

SET_DPS(1,2,V,150,MA);

VT = 1;

SET_OUTPUT_LEVEL(1.9,0.1);

for(i=1;i<10;i++)

{

 SET_INPUT_LEVEL(VT,0);

 re = RUN_PATTERN(4,1,0,0);

 if(re==1)

 {

 VT=VT + 0.1; SET_INPUT_LEVEL(VT,0);

 re = RUN_PATTERN(4,1,0,0);

 if(re==1)

 {

 break;

 }

```

    }

    VT=VT + 0.1;

}

V_2 = VT - 0.15;

str.Format("VT+(2V)");

if((V_2 > 1.5)||(V_2 < 1.0))
    BIN(7);

SHOW_RESULT(str,V_2,V,1.5,1.0);

//THRESHLD VOLTAGE NEGATIVE-GOING TEST (+2V)

for(i=1;i<15;i++)

{

    SET_INPUT_LEVEL(VT,0);

    re = RUN_PATTERN(4,1,0,0);
    if(re==0)

    {

        VT = VT - 0.1;

        SET_INPUT_LEVEL(VT,0); re =
        RUN_PATTERN(4,1,0,0);

        if(re==0)

        {

            break;

        }

    }

    VT=VT - 0.1;

}

VT = VT + 0.15;

```

```

str.Format("VT-(2V)");

if((VT < 0.3 )||(VT > 1.0))

BIN(8);

SHOW_RESULT(str,VT,V,0.9,0.3);


//HYSTERESIS VOLTAGE TEST (+2V)
VH = V_2 - VT;
str.Format("VH(2V)");
if((VH < 0.2)|| (VH > 1.2))
BIN(9);
SHOW_RESULT(str,VH,V,1.2,0.2);
SET_INPUT_LEVEL(0,0);

```


第五章 接口板开发

socket-adapter	<p>设计:</p> <ol style="list-style-type: none"> 1、PCB 软件熟练使用 (protel、Altium®Designer 等) 2、器件管脚数对应测试系统通道数, 可用拨码开关做通用板卡。(限数字器件) 3、模拟器件接口板比较复杂, 不能做通用板卡。 <p>焊接:</p> <ol style="list-style-type: none"> 1.→找对应的元器件、接插件, 根据不同封装工艺选择卡座等 2.→忌短路、虚焊等, 检查焊接符合工艺要求
任务	<p>实操+校正</p> <p>通过连接性测试</p>

第六章 安装调试

1. 硬件安装

ST3020 集成电路测试系统硬件安装步骤如下：

- a) 将 IC 测试主机安放于固定位置
- b) 将电源线插入 主机后面的电源插座上
- c) 打开计算机机箱，将接口卡插入 PCI 插槽（或者看着网线连接网口）
- d) 将 50 芯电缆线一端与接口卡的插座连接，一端与 主机后面的 50PIN 插座连接
- e) 如安装 Prober 或 Handler，将 36 芯电缆线一端与 主机后面的 36PIN 插座连接，另一端与 Prober 或 Handler 连接。（详细看 9.4 Handler 或 Prober 连接）

2. Handler 或 Prober 连接

ST3020 集成电路测试系统 Handler 或 Prober 连接的控制信号详细说明如下：

- 1、在测试设备内，由 25 芯电缆连接线连接到系统总线控制板的 J4 Connection port。
- 2、连接信号定义如下：

36 芯电缆	名称	36 芯电缆	名称
插头序号		插头序号	
1	VCC	19	VCC
2	START	20	START/
3	GND	21	EOW
4	EOW/	22	GND
5	BIN1	23	BIN2
6	BIN3	24	BIN4
7	GND	25	EOTL/

8	EOT	26	GND
9	CAT0/	27	GND
10	CAT1/	28	GND
11	CAT2/	29	GND
12	CAT3	30	GND
13	CAT4	31	GND
14	CAT5	32	GND
15	CAT6	33	GND
16	CAT7	34	GND
17	CAT8	35	GND
18	CAT9	36	GND

3. 校准&自检适配板安装

校准&自检适配板，用于设备的校准和参数自检。在运行校准自检程序之前，将其正确安装在系统测试台。

ST3020 共包含两种校准自检板，分别为 ST3000-080011-CALB、ST3020-090010-CALB，

使用中根据软件提示更换不同校准板。在运行功能自检前，将通用测试板安装在系统测试台上。

4. 校准&自检程序启动

校准&自检程序集成在测试运行环境中，可由 ToolForST3020.exe 调用，启动自检程序；也可在工作目录下直接启动自检程序，校准程序的文件名称为：CALI.EXE，参数自检的文件名称为：SELFTEST.EXE，功能自检的文件名称为：SELFMemory.EXE。

5. 校准程序使用说明

5.1 校准项目

校准项目包括 PMU、VIS、VS、HDPS、AS、AVM、DMV、DPS、Vih/Vil 和 Voh/Vol。

DPS、Vih/Vil 和 Voh/Vol 部分使用标号为 ST3020-090010-CALB 的校准板，利用 5

个精密电阻对测试机进行校准，在校准板上分别标识为 R1、R2、R3、R4 和 R5。PMU、VIS、VS、HDPS、AS、AVM、DMV 部分使用标号为 ST3000-080011-CALB

的校准板，利用 6 个精密电阻对测试机进行校准，在校准板上分别标识为 R1、R2、R3、R4、R5 和 R6。

各项校准操作步骤见 2.2。

5. 2 校准操作步骤

1) 安装校准板

2) 校准前准备 在运行校准程序前，在系统要求的运行环境下（环境温度：22°C~26°C，相对湿度：

30%~60%），首先将测试系统预热半小时以上，将正表笔连接到校准板的 TP1，负表笔连接到校准板的模拟地 AGND 点，然后对校准板上的运放进行调零处理。该运放在校准板下端中部，调零时要用短路子将运放输入接地。调零结束后恢复原位。



图B-1 校准板运放短路子

3 校准

运行校准程序 CALIB.EXE 后，弹出 ST-3020 测试系统校准程序（图 B-2）对话框。

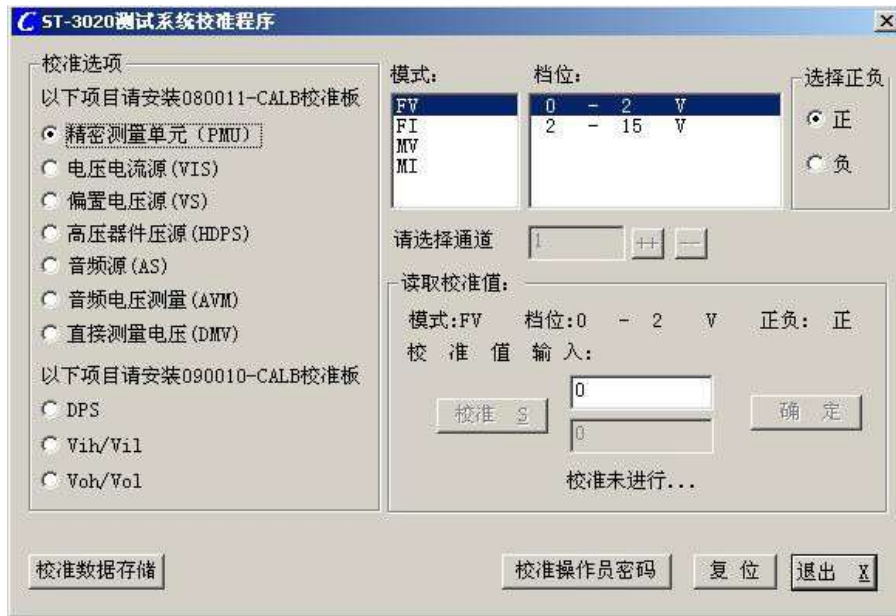


图 B-2 校准程序界面

- a. 根据需要选择校准选项;
- b. 选择“模式”框内需要校准的内容;
- c. 在“档位”栏选择需要校准的档位;
- d. 在“选择正负”栏选择需要校准的项目;
- e. 按“校准”按钮开始校准,然后将数字电压表的读值输入到“校准”按钮右侧上方的编辑框中。
- f. 按“确定”按钮后,再将下一个数字电压表的读值输入到“校准”按钮右侧下方的另一编辑框。
- g. 再按“确定”按钮,即完成一个项目的校准。用户可根据实际需要选择需要校准的项目。

5.3 校准结果处理

全部校准完成以后,单击 **校准数据存储** 按钮,可将最新校准数据保存到 CaliFile 目录下的各校准文件中;

如果在校准过程当中停止校准,应单击 **复位** 按钮,将系统复位。所有操作完成后,单击 **退出** 按钮,退出校准程序。

6. 参数自检程序使用说明

运行参数自检程序，弹出 ST-3020 测试系统自检程序对话框（图B-3）。在此对话框中，进行如下操作



图 B-3 自检程序界面

6. 1 参数自检项目选择

参数自检项目包括 PMU、VIS、VS、HDPS、Analog、DPS、Vih/Vil、Voh/Vol。用户可以分别对各个项目进行自检，也可以选择全部，对所有项目进行自检。

6. 2 参数自检执行和结果处理

选择要进行的自检项目后，按 **执行自检** 按钮，开始执行自检，自检合格或失

失败结果直接显示在屏幕上，同时结果存储在 SelfTest.TXT 文件中。

PMU	参数测量单元加压、加流、测流、测压自检
VIS	电压电流源加压、加流、测流、测压自检
VS	偏置源加压自检
HDPS	高压器件电压源加压、测流自检
Analog	模拟功能板参数自检
DPS	器件电压源加压、测流自检
Vih/Vil	管脚驱动电压自检
Voh/Vol	管脚比较电压自检

7. 功能自检程序使用说明

运行功能自检程序，弹出 ST-3020 测试系统功能自检程序对话框。



图 B-4 功能自检程序界面

7. 1 功能自检模式选择


功能自检是对设备中存储器及运行图形情况的自检。

7. 2 功能自检执行

执行自检前，先设置要检测的通道板速率及板号。如设置错误，会出现检测失效或

不能进行完全检测。

各选项设置完成后，选择自检项目，选择全部通道板，将依次检测所有通道板，也

可以根据需要选择单一通道板或多块通道板。最后单击  执行。执行结束后，在被测通道板后标记“PASS”或“FAIL”。

7.3 注意事项

1. 在测试器件前，安装测试板，必须在关机的状态下把测试板安装到测试头上。
2. 检查测试板的拨码开关是否正确，如不正确，重新设置，设置拨码开关顺序为，先断开通道，再连接电源或接地。
3. 在需要外加负载时，必须在断电情况下操作。
4. 先打开计算机，然后接通测试系统电源开关，待机器稳定后，开始测试。
5. 启动系统文件 ST3020Menu.exe(菜单方式)或 ST3020.exe(编程方式)，调入测试程序测试。如提示“测试系统没有准备好”，请检查是否接通测试系统，检查计算机与测试仪之间的连接电缆是否正确。
6. 如在运行程序过程中出现死机状态，请关闭测试系统，键入 Ctrl+Alt+Del，关闭执行程序。然后重新接通测试系统电源，启动测试程序。
7. 为保证系统安全，在安装接口卡时，必须关闭计算机，断开计算机的 220V 电源线。在连接电缆前，必须关闭计算机和测试系统，不能在开机状态下连接电缆。在安装测试板和自检板之前，必须关闭测试系统电源。

附录 A：校准&自检工具说明