

## 目录

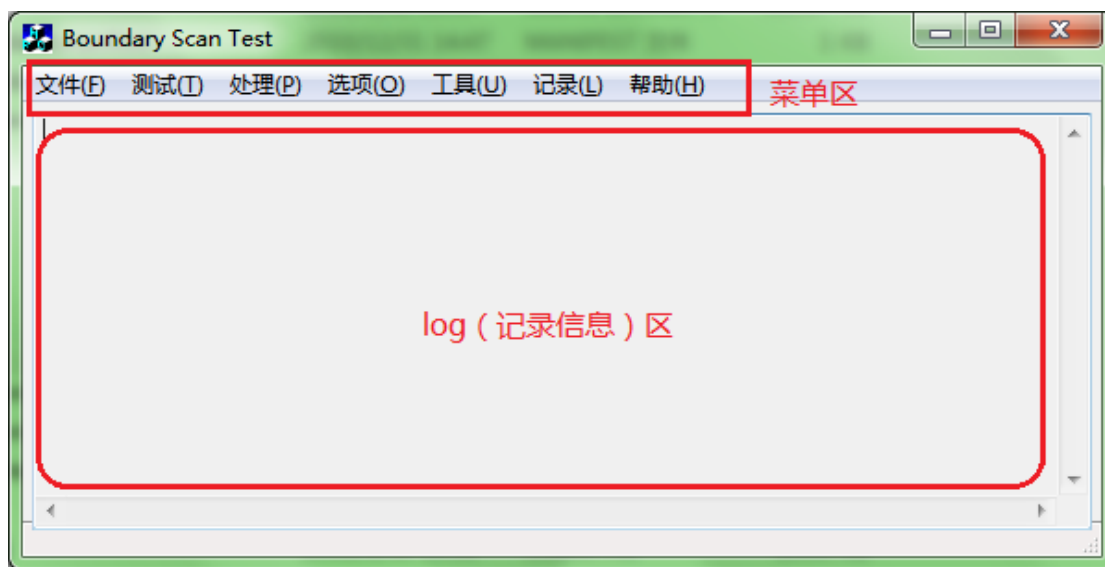
主界面说明.....	3
操作准备.....	3
手工测试.....	5
准备.....	5
如果有 JTAG 链上的所有器件的 BSDL 文件 .....	6
如果没有 .....	8
前置器件指令长度(Prefix Instruction Length).....	8
后置器件指令长度(Postfix Instruction Length) .....	8
设置.....	8
操作(Operation) .....	9
设置输入输出(I/O).....	9
快捷设置.....	9
设置工具(Utilities) .....	11
根据管脚位置(Pin Location)查找 BSC 号(BSC No.) .....	11
根据端口名称(Port Name)查找 BSC 号(BSC No.) .....	12
输出设置.....	12
输入设置.....	13
更多器件及完成.....	13
测试.....	13
器件互联测试.....	14
启动测试.....	14
测试设置.....	15
参数释义.....	15
一个实例.....	16
技巧.....	16
运行测试.....	17
不同方式启动测试时界面区别 .....	19
单个器件测试.....	20
测试准备.....	20
输出设置(Output Setting)说明 .....	21
期望输入(Expected Input)说明 .....	22
保存.....	22
工具.....	22
别名.....	22
自动设置.....	23
通过正规表达式批量设置.....	23
修改已经保存的设置.....	25
测试步骤.....	25
举例 1.....	26
正常情形.....	27
管脚之间短路.....	27
说明.....	28
举例 2.....	28

正常情形.....	29
管脚与电源短路.....	29
管脚与地短路.....	30
信号管脚之间短路.....	30
说明.....	31
外设测试.....	31
编辑 JFP 配置文件.....	31
运行.....	31
批量测试.....	33
编辑工作空间文件.....	33
运行.....	33
PLD 加载.....	34
获取.svf 文件.....	34
启动软件.....	35
选择.svf 文件.....	35
执行.....	37
PseudoCLI 特性.....	37

说明：各个版本的软件可能存在某些差异，因此您的软件(界面/操作)可能会与本文档略有不同。

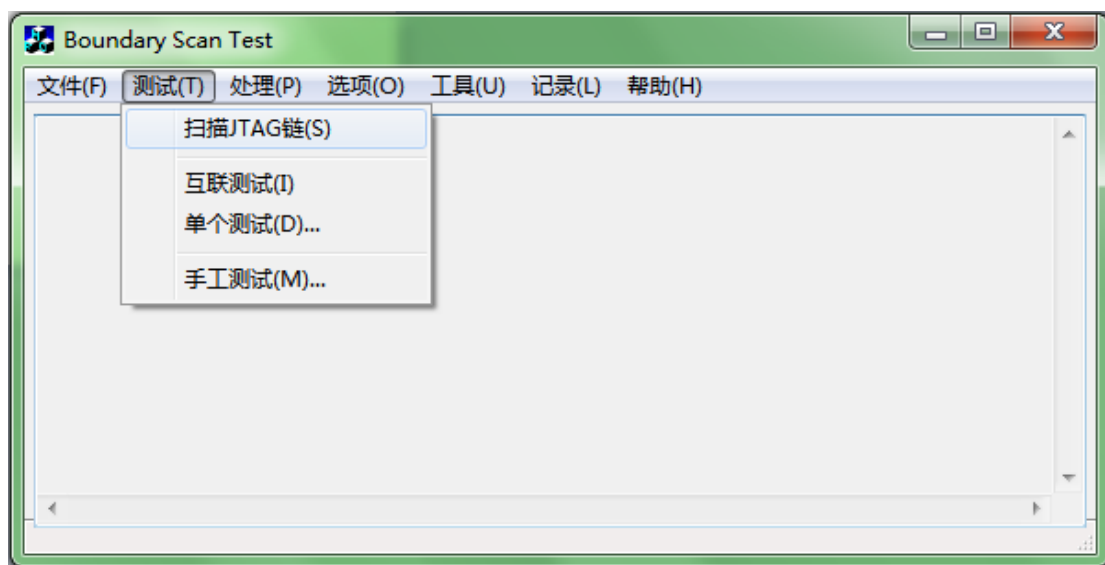
注意：为了保证最佳测试效果和最大测试覆盖率，请尽量保持 FPGA 和 CPLD 在未编程状态，在测试前不要对其进行烧写；类似的，保持 CPU 为未运行状态，JTAG 测试前不要对其 Boot ROM/Flash 进行烧写。

## 主界面说明

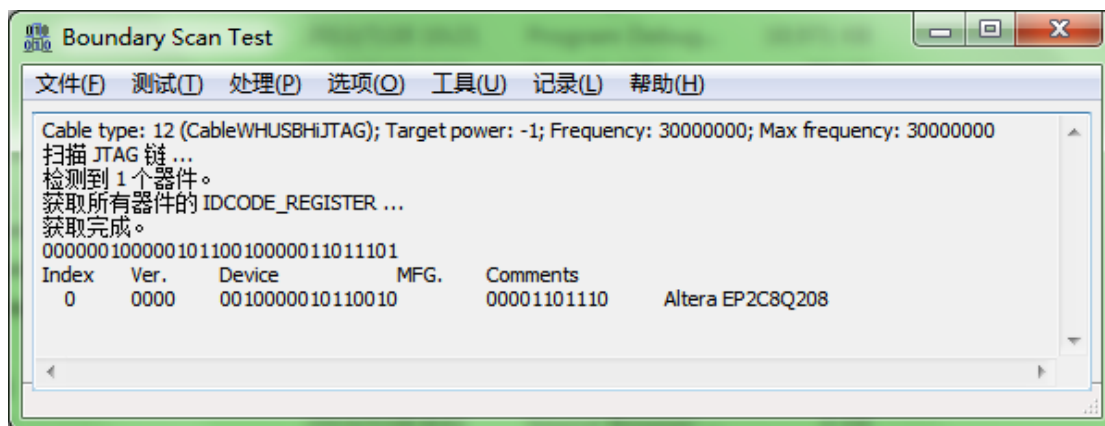


## 操作准备

- 1, 请先获取 JTAG 链上待测器件的 BSDL 文件，放到与程序同一目录下；对于链上的其它器件，如果有 BSDL 文件最好，如果没有则需要知道每个器件的 JTAG 指令长度。
- 2, 单板断电；
- 3, 连接加载线；
- 4, 单板上电；
- 5, 运行程序；
- 6, 在程序主窗口对话框出现后，点击菜单 **测试(Test) - 扫描 JTAG 链(Scan JTAG Chain)**，看能不能扫描到器件。参见下图：



如果能扫描到器件，程序会显示相关信息（比如器件 ID 等）。参见下图：



另一个例子，在 Xilinx ML505 板上实际扫描结果 log 文件：

```
Cable type: 10; Target power: -1; Frequency: 6000000
```

```
扫描 JTAG 链 ...
```

```
检测到 5 个器件。
```

```
获取所有器件的 IDCODE_REGISTER ...
```

```
获取完成。
```

```
11110101000001011001000010010011
```

```
11110101000001011001000010010011
```

```
01011001011000001000000010010011
```

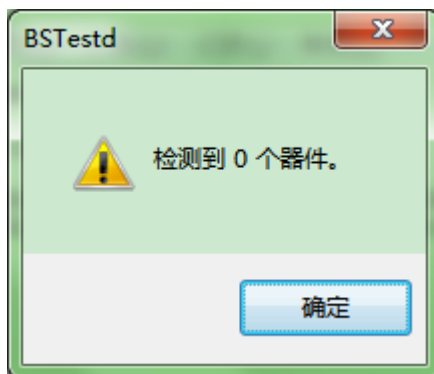
```
000010100000000000001000010010011
```

```
11000010101010010110000010010011
```

Index	Ver.	Device	MFG.	Comments
0	1111	0101000001011001	00001001001	Xilinx XCF32P
1	1111	0101000001011001	00001001001	Xilinx XCF32P
2	0101	1001011000001000	00001001001	Xilinx XC95144XL
3	0000	1010000000000001	00001001001	Xilinx XCCACE

4 1100 0010101010010110 00001001001 Xilinx XC5LX50T

如果不能扫描到任何器件，程序会有对话框提示，需要按照下文要求进一步核对。提示信息参见下图：



如果不能扫描到任何器件，请检查加载电缆是否正常、目标板加载头的电路是否正确。

如果可以扫描到器件，请您确认扫描到的器件数量跟目标板 JTAG 链上的实际器件数量是否一致。如不一致，则要检查。

程序扫描到的器件列表跟 JTAG 链上的实际器件一致以后，才可以进入下列步骤。

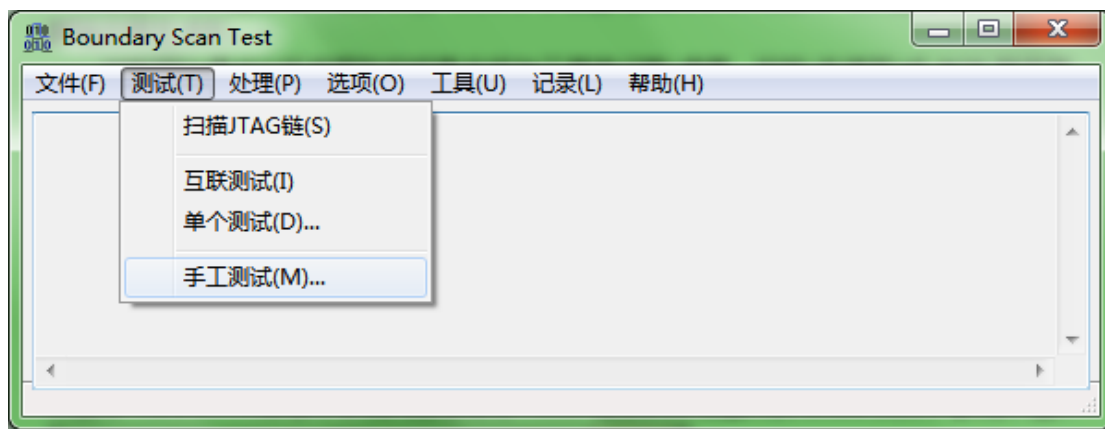
## 手工测试

本系统可以显示芯片管脚状态（类似于示波器/逻辑分析仪功能），可以通过控制芯片管脚输出特定波形。

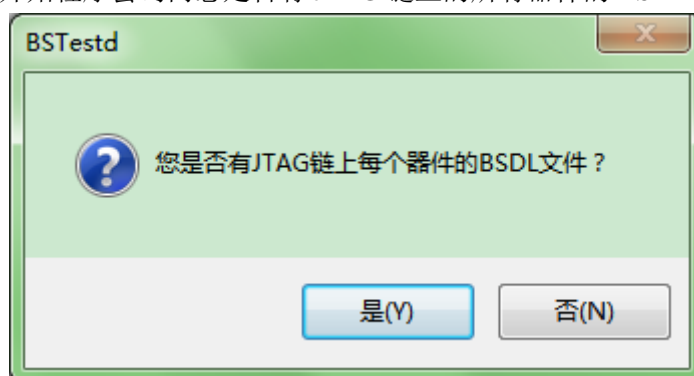
这样可以通过分析观察到的现象发现加工焊接问题(虚焊、短路/连锡等)或 PCB 制作等问题，同时也可以作为单板调试的辅助手段。

## 准备

点击菜单 **测试(Test)** – **手工测试(Manual Test)** 。参见下图：



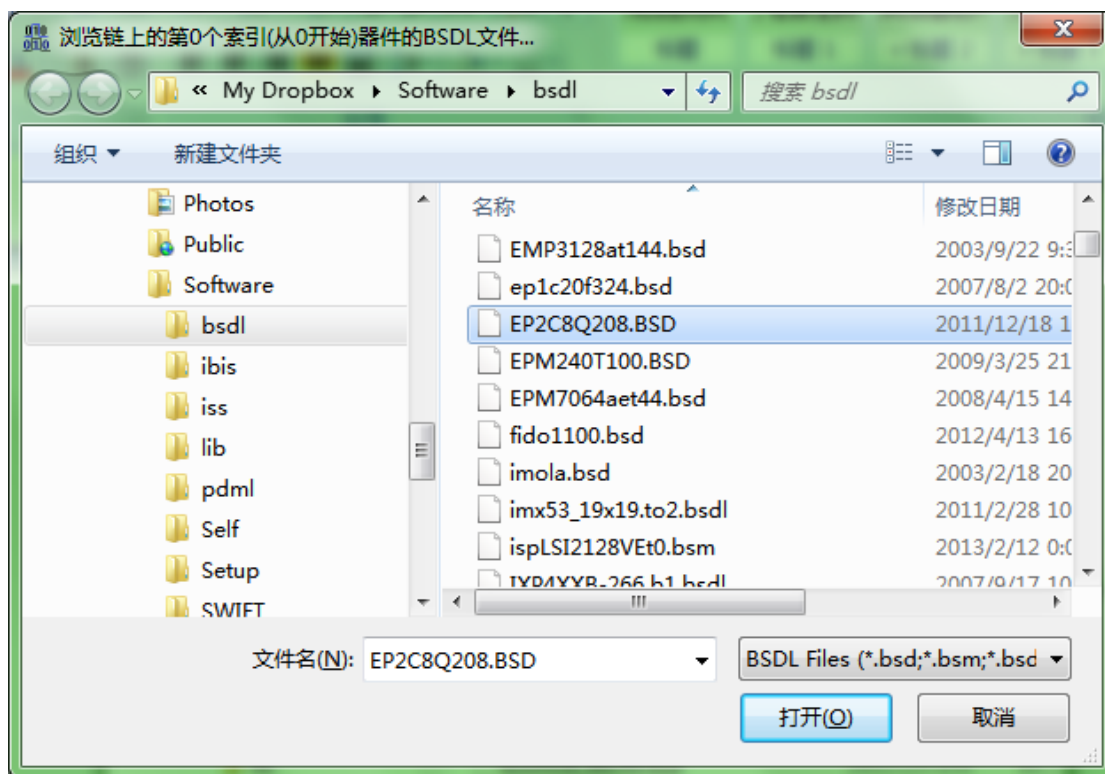
从版本 V2.0.0.3 开始程序会询问您是否有 JTAG 链上的所有器件的 BSDL 文件，参见下图：



这主要是为了解决如果您仅仅有某些器件的 BSDL 文件从而无法在低版本进行测试的问题。请根据实际情况选择。

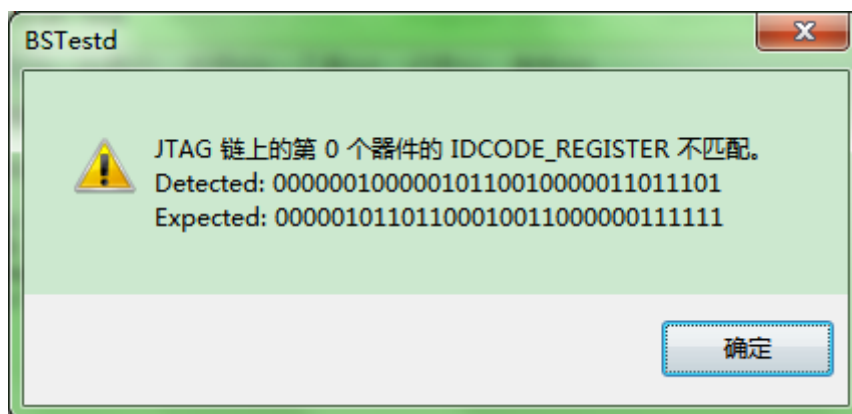
## 如果有 JTAG 链上的所有器件的 BSDL 文件

程序则要求依次为 JTAG 链上的每个器件选择正确的 BSDL 文件。参见下图：

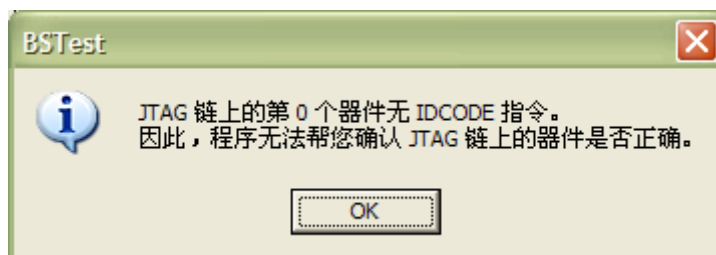


程序将逐个分析 BSDL 文件，全部分析完成后，程序将运用 BSDL 文件给予的信息依次检查每个器件的 ID。

如果任意一个器件的 ID 检查失败，程序将提示，用户需要依照前面的指导检查，直到链上每个器件的 ID 全部通过检查。下图是 ID 检查失败的一个例子：



说明：有的器件没有 IDCODE 指令，程序会提示，但不是错误信息，参见下图：



这些步骤成功完成以后，将进入 **手工测试 (Manual Test)** 对话框。

## 如果没有

程序会弹出一个设置对话框，参见下图：



## 前置器件指令长度(Prefix Instruction Length)

整个 JTAG 链上在待测 JTAG 器件之前的其他器件的指令长度，如果有多个这样的器件，长度用“|”符号分割，如“4|5”表示待测 JTAG 器件之前有两个器件，它们的 JTAG 指令长度分别 4 和 5。如果没有前置器件请留空。

## 后置器件指令长度(Postfix Instruction Length)

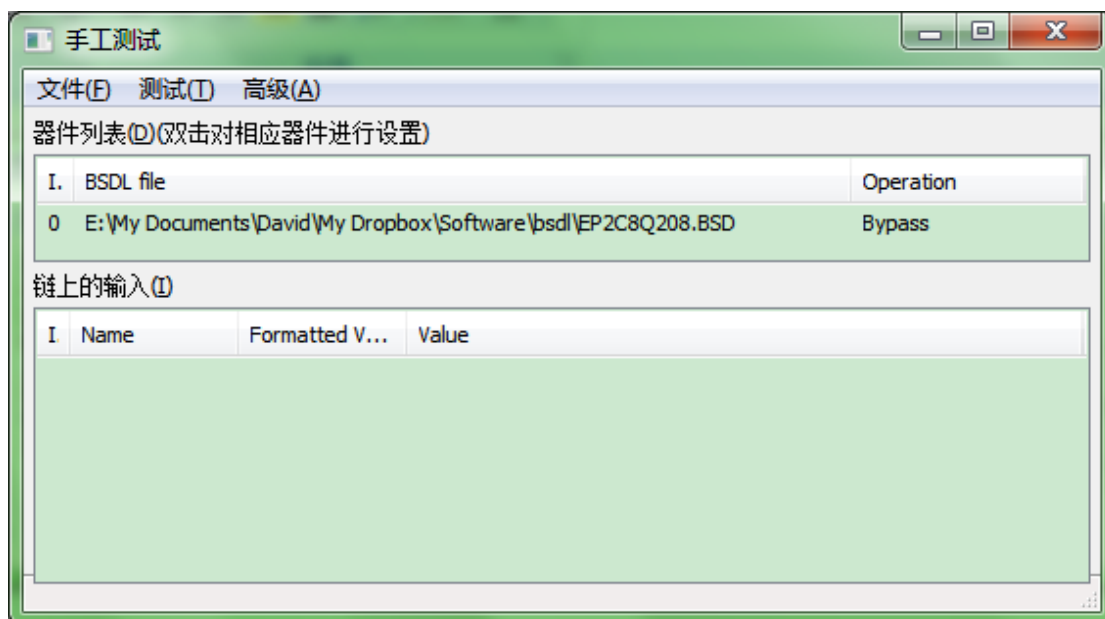
整个 JTAG 链上在待测 JTAG 器件之后的其他器件的指令长度。其他同“前置器件指令长度”。如果没有后置器件请留空。

## 设置

程序缺省把所有器件设为 **Bypass**。

- 根据需要，双击 **器件列表(Device Lists)** 中要进行测试的器件，将出现 **Setting for Device n** 对话框。(说明：n 是器件在链中的位置，从 0 开始，下同。)





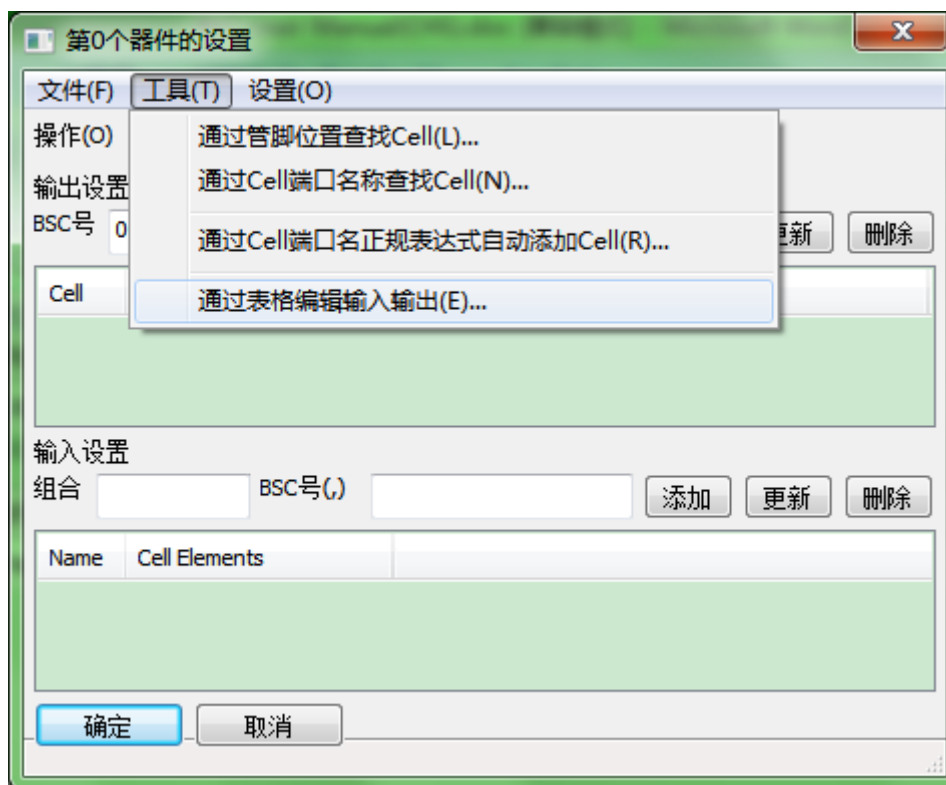
## 操作(Operation)

首先，将器件的 **操作(Operation)** 选为 **Test**（如果确实不需要测试的器件，请保持操作为 Bypass）

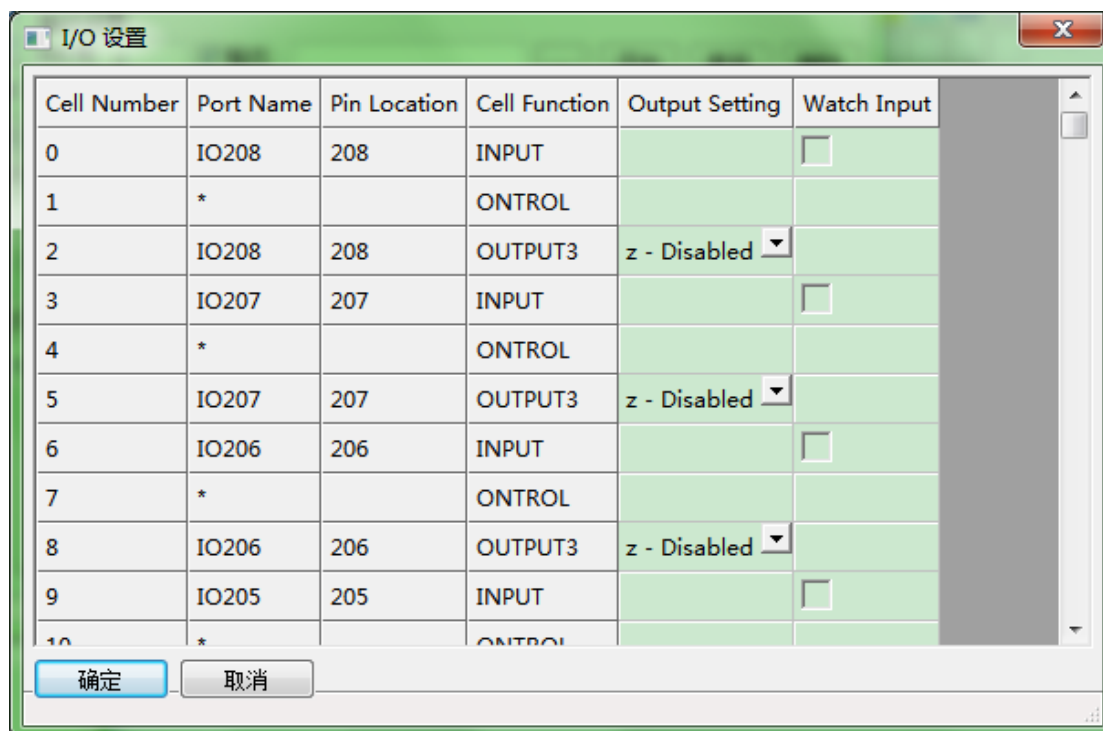
## 设置输入输出(I/O)

### 快捷设置

菜单 **工具(Utilities)** / **通过表格编辑输入输出(Edit I/O Settings in Table)...**，参见下图：



然后将弹出 **I/O 设置(I/O Setting)** 对话框，参见下图：



如果要某个管脚输出，请直接填写要输出的图案字符串，或者在点击相应的表格单元的 Combobox 的下拉按钮，选择要输出的值；

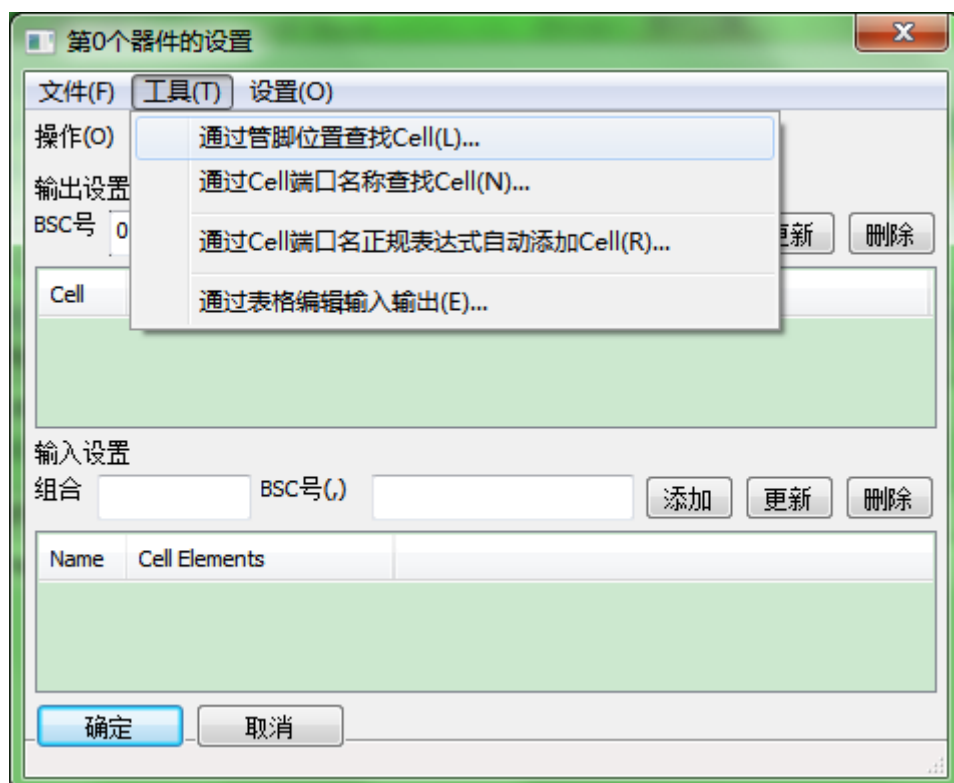
如果要观察某个管脚的输入，请把相应的表格单元的 Checkbox 选上；

操作完成后，请点击 **确定(OK)** 按钮返回 器件设置(Setting for Device n) 对话框。

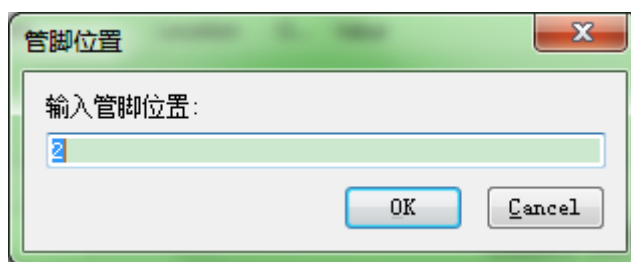
## 设置工具(Utilities)

### 根据管脚位置(Pin Location)查找 BSC 号(BSC No.)

点击菜单 工具(Utilities) – 通过管脚位置查找 Cell (Look up cell by pin location...):

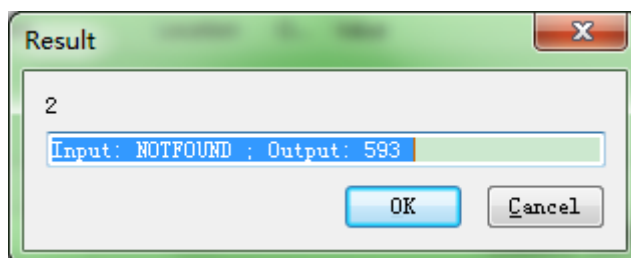


然后在弹出的对话框中输入管脚位置，参见下图：



然后点击 **确定(OK)** 按钮；

程序会从 BSDL 文件中找到该管脚的 Cell，然后判断该 Cell 是否可以输入或输出，然后通过对话框显示出来。

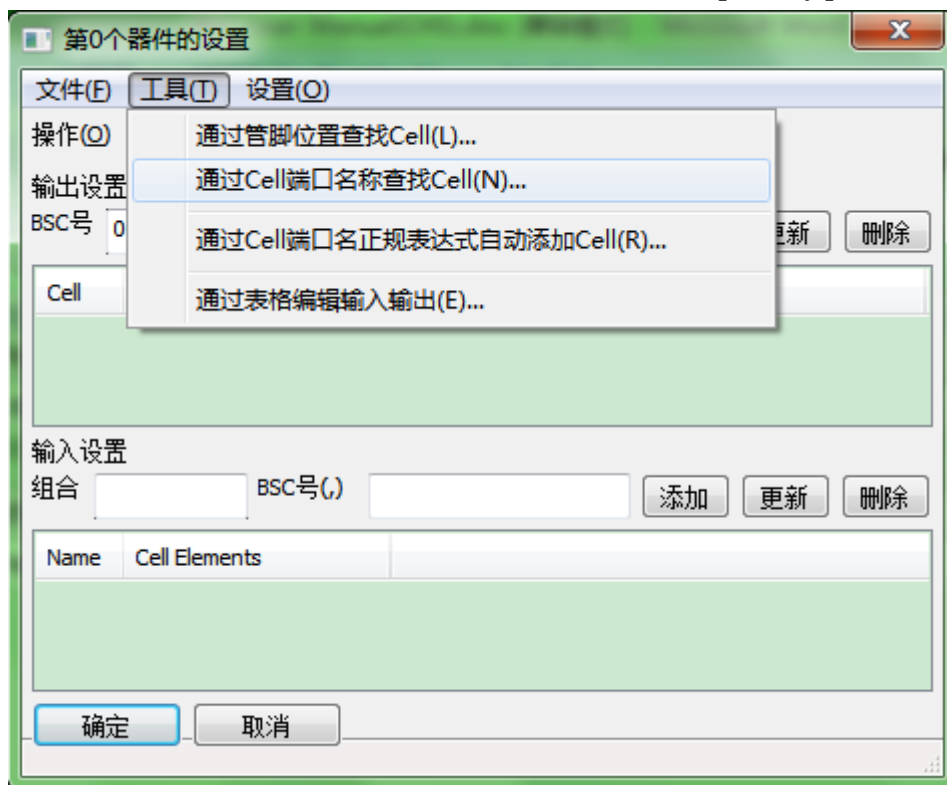


这时，您可以根据查询结果和您的需要，将相应结果拷贝添加到输入或输出的 BSC 号(BSC No.)后的文本框内。

## 根据端口名称(Port Name)查找 BSC 号(BSC No.)

类似的，您还可以根据端口名称(Port Name)查找 Cell 号(Cell Number)。

方法是点击菜单 工具(Utilities) – 通过端口名称查找 Cell (Lookup cell by port name...)



## 输出设置

在 BSC 号(BSC No.)后的文本框输入 1 个 Cell 号，再在 **输出(Output)** 复选框后的 **输出内容** 文本框内填上要输出的电平（0 或 1），然后点击 **添加(Add)** 按钮。

说明：实际上，可以输出任意序列，也就是说，**输出内容** 文本框内可以填入字符串。比如输出内容是“01”则输出占空比 1:1 的方波，而“0100”则输出占空比是 1:3 的波形。

重复上述步骤，直到所有输出设置完成。

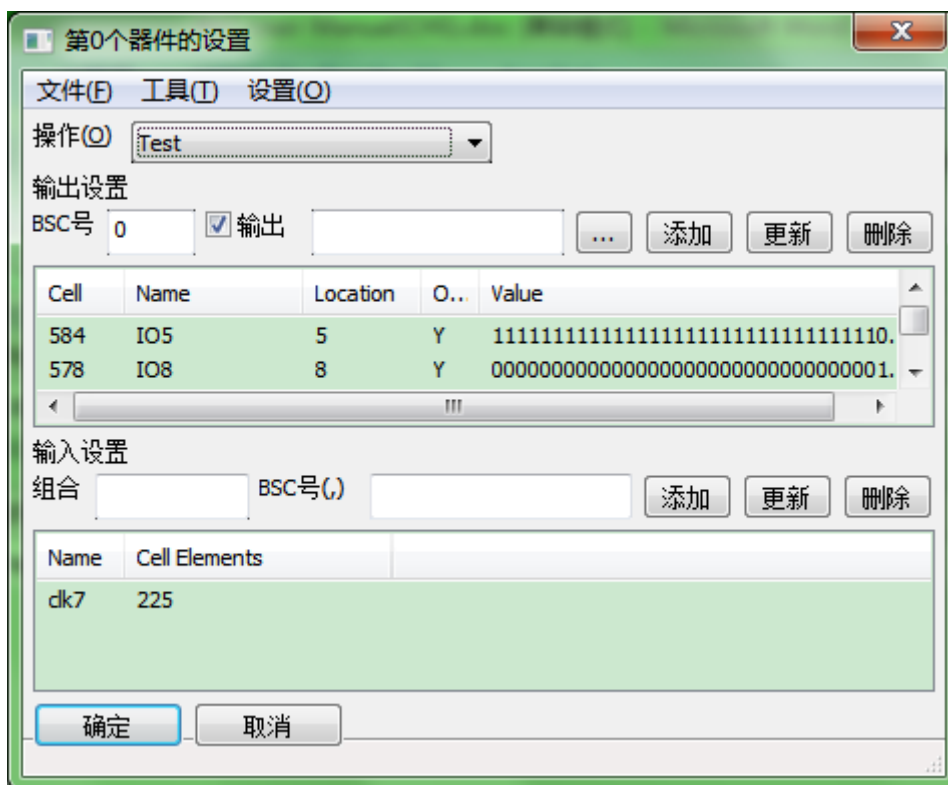
## 输入设置

给输入起个名字，在 组合 后的文本框内填上名字，再在 BSC 号 后的 **输入 Cell 号(BSC numbers)** 文本框内填上 1 个 Cell 号，然后点击 **添加(Add)** 按钮。

（说明：实际上，可以把多个 Cell 的输入组合到一起观察。这时，多个 Cell 号之间用英文逗号分隔。）

重复上述步骤，直到所有输入设置完成。

下图是一个例子：



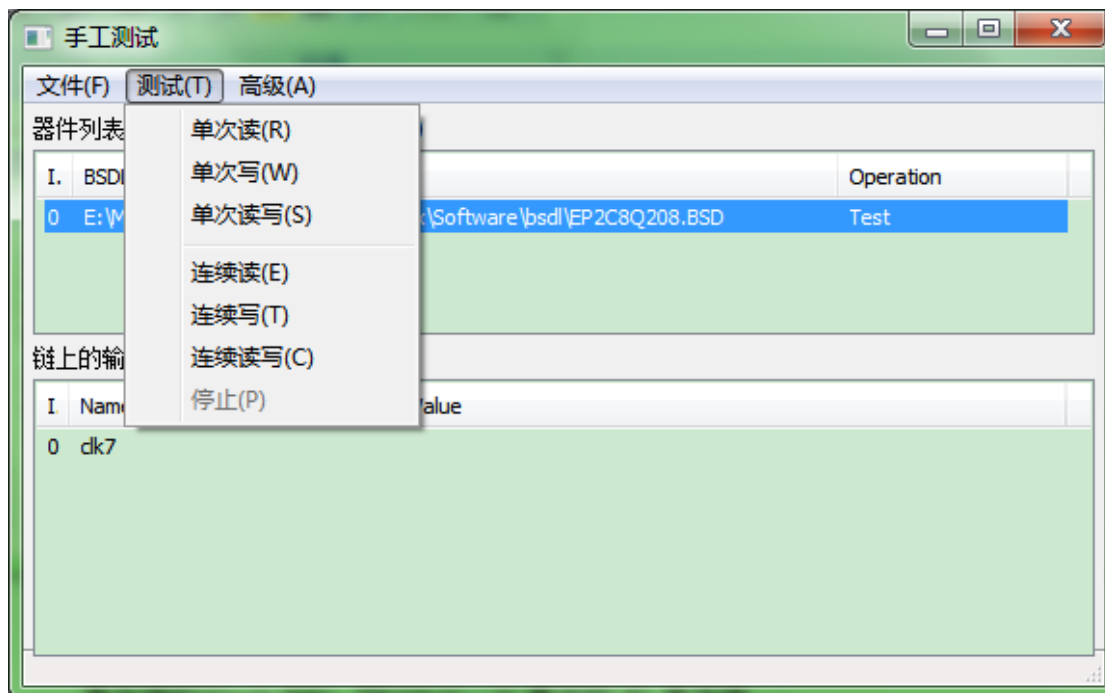
## 更多器件及完成

- 对该器件的所有设置完成后，点击 **确定(OK)** 按钮，返回到 **手工测试(Manual Test)** 对话框。
- 重复上述步骤，直到都需要设置的器件都设置完毕。
- 返回 **手工测试(Manual Test)** 对话框。

## 测试

在 **手工测试(Manual Test)** 对话框中测试。

手工测试(Manual Test) 对话框的菜单 **测试(Test)** 下的几个测试功能说明如下:



**单次写(Single Write):** 程序把链上所有 操作设成了 Test (不是 Bypass) 的器件的输出 (如果有) 送到管脚上, 执行 1 次。

**单次读(Single Read):** 程序把链上所有 操作设成了 Test (不是 Bypass) 的器件的输入 (如果有) 读取并显示出来, 执行 1 次。

**单次读写(Single RW):** 同时执行一次 **单次写** 和 **单次读**。

**连续写(Cont. Write):** 不停执行 **单次写** 直到 **停止**(菜单 **测试(Test)** – **停止(Stop)**)。

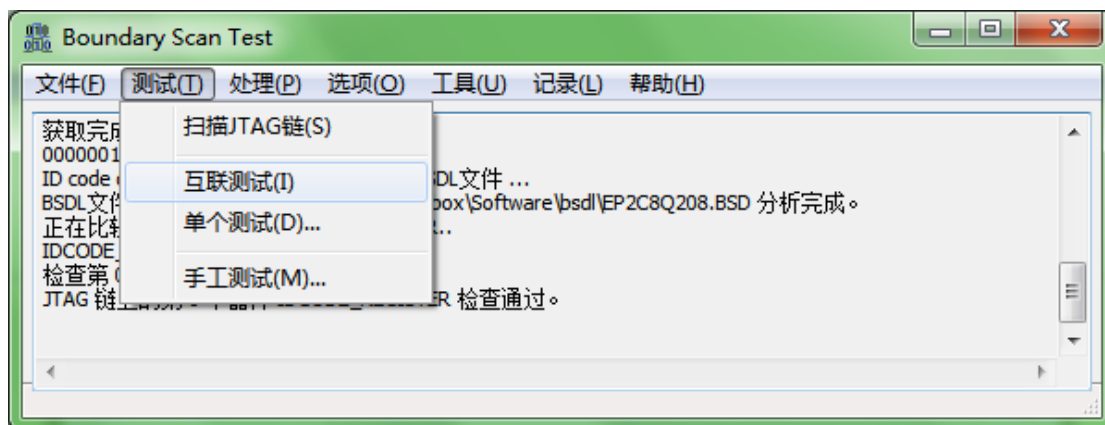
**连续读(Cont. Read):** 不停执行 **单次读** 直到 **停止**(菜单 **测试(Test)** – **停止(Stop)**)。

## 器件互联测试

器件互联测试在对网表文件进行分析的基础上, 自动测试相关器件的连接关系, 从而发现测试短路(管脚连锡、跟电源/地短路)、开路(虚焊、PCB 铜箔断裂)等情形。

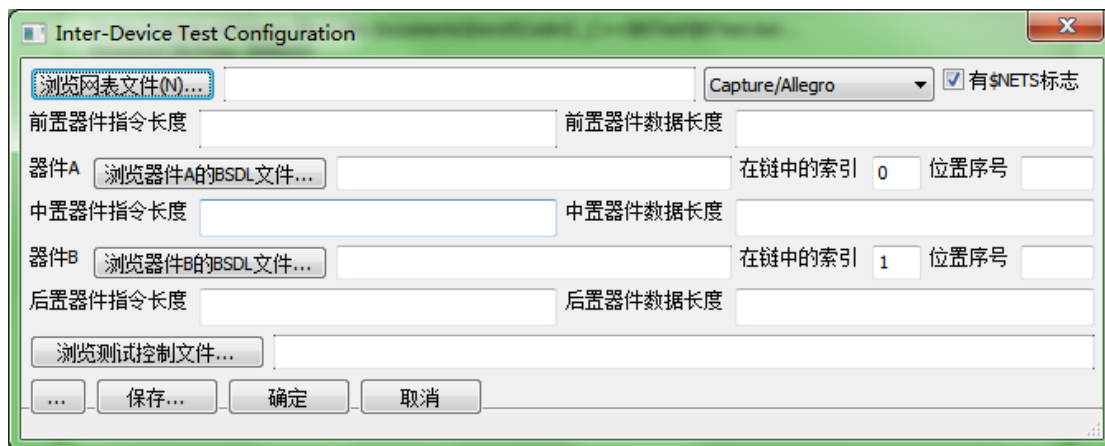
## 启动测试

运行程序, 点击菜单 **测试(Test)** – **器件互联测试(Inter-device Test)**, 参见下图:



## 测试设置

然后在弹出的 **器件互联测试配置(Inter-Device Test Configuration)** 对话框中设置参数:



## 参数释义

指定原理图网表文件、网表格式、是否有\$NETS 标志。

分别指定两个待测器件（称为器件 A、B）的 BSDL 文件、在 PCB 上位置序号(Part Ref)、在 JTAG 链上的位置(Index in JTAG Chain)(索引基于 0)。

**前置器件指令长度(Prefix Instruction Length):** 整个 JTAG 链上在待测 JTAG 器件之前的其他器件的指令长度，如果有多个这样的器件，长度用“|”符号分割，如“4|5”表示待测 JTAG 器件之前有两个器件，它们的 JTAG 指令长度分别 4 和 5。如果没有前置器件请留空。

**中置器件指令长度(Infix Instruction Length):** 整个 JTAG 链上在待测 JTAG 器件 A 和待测 JTAG 器件 B 之间的其他器件的指令长度。其他同“前置器件指令长度”。如果没有中置器件请留空。

**后置器件指令长度(Postfix Instruction Length):** 整个 JTAG 链上在待测 JTAG 器件之后的其他器件的指令长度。其他同“前置器件指令长度”。如果没有后置器件请留空。

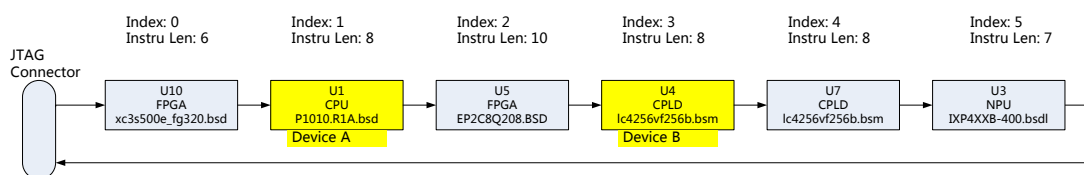
数据长度不需要填写。

**测试控制文件(Test Control File):** 对于 IDT 测试中其他非互联管脚需要进行额外控制时使用。文本文件，一行一个设置。语法：**位置序号<TAB>管脚位置<TAB>输出值。**

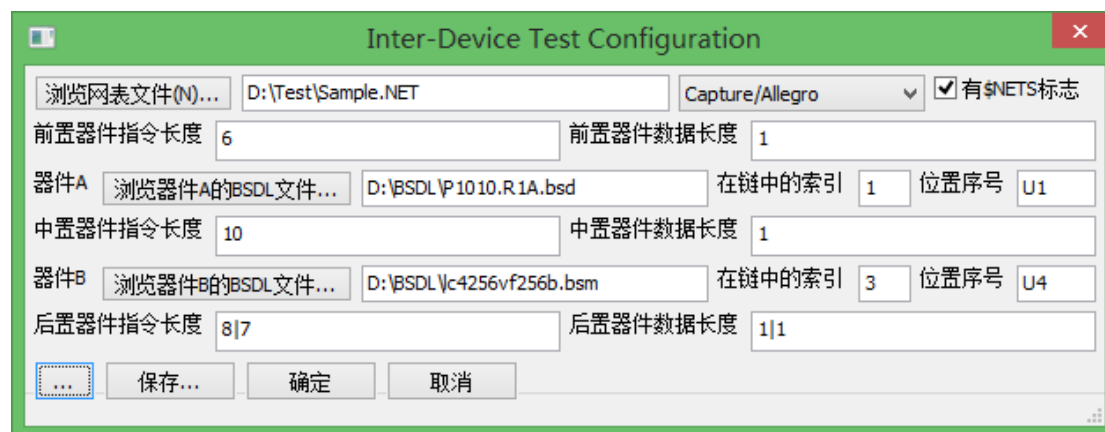
例如：IC11 3 1

## 一个实例

为了便于理解，下面举一个实例：假设某板卡上有 6 个器件在一个 JTAG 链上，我们需要对 U1 和 U4 进行互联测试。如下图所示：



那么应该设置如下：




说明：

- 所有索引基于 0 开始。
- 因为除了被测器件 U1 和 U4 外，其他器件都是 Bypass 的，所以在此互联测试场景下，这些 Bypass 器件在 JTAG 链上的数据长度都是 1。这个参数也可以不填写。

## 技巧

为了方便，您可以点击 **保存(Save)...** 按钮把这些设置保存一个扩展名为.ini 的文本文件，

然后下次可以通过对话框左下角的  按钮打开这个.ini 文件，这样就不用一个个手工去设置了。



## 运行测试

设置正确后，点击“**确定(OK)**”按钮。

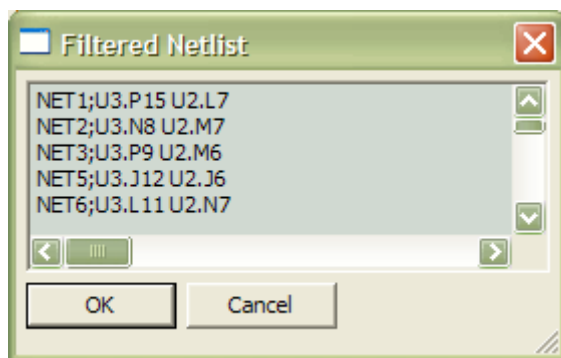
程序将扫描 JTAG 链，并显示所有器件的信息。

程序将依次分析链上所有器件的 BSDL 文件。

分析完成以后，程序将对链上的每个器件进行 ID 检查。

全部检查通过以后，程序将分析网表文件。

分析结束之后，程序将根据指定的器件的位置序号，过滤网表，忽略其他网表，显示跟指定两个器件有关系的网表，弹出 **Filtered Netlist** 对话框。



这时，用户可以自己手工添加/删除网表。

说明：

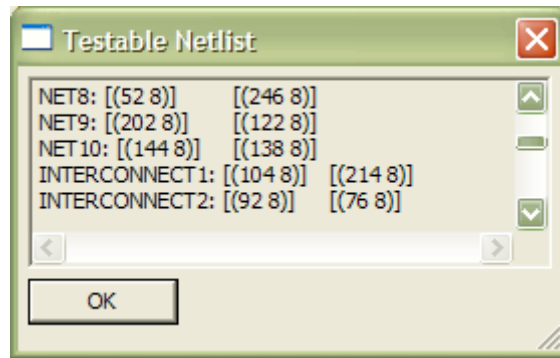
- 有些不可以测试网表，比如 GND VCC TDI TDO 之类的，程序会予以剔除，无需理会。
- 有些不想测试的网表，可以手工删除。
- 同时可以添加一些网表：比如有的网表虽然没有直接相连，但是是通过电阻或者驱动器甚至是 logic cluster 连起来了（在网表里面没有直接体现），这还是可以测试的。

用户编辑以后，点击“**确定(OK)**”；如果不编辑，点击“**取消(Cancel)**”即可。

之后，程序将进行可测试性网表分析。

分析的依据是管脚是否相连，以及管脚是否互为输入输出等。

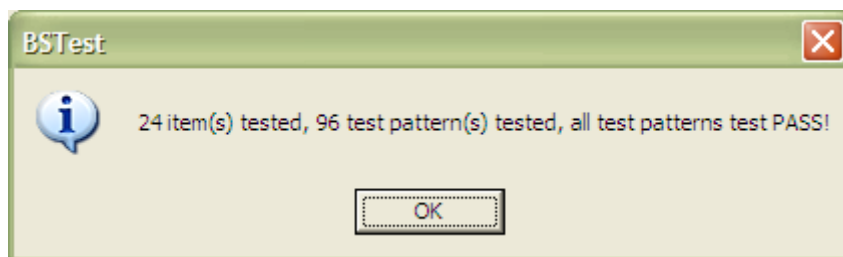
分析的结果在弹出 **Testable Netlist** 对话框中显示。



点击“确定(OK)”。

之后，程序将开始测试。

最后显示测试结果。

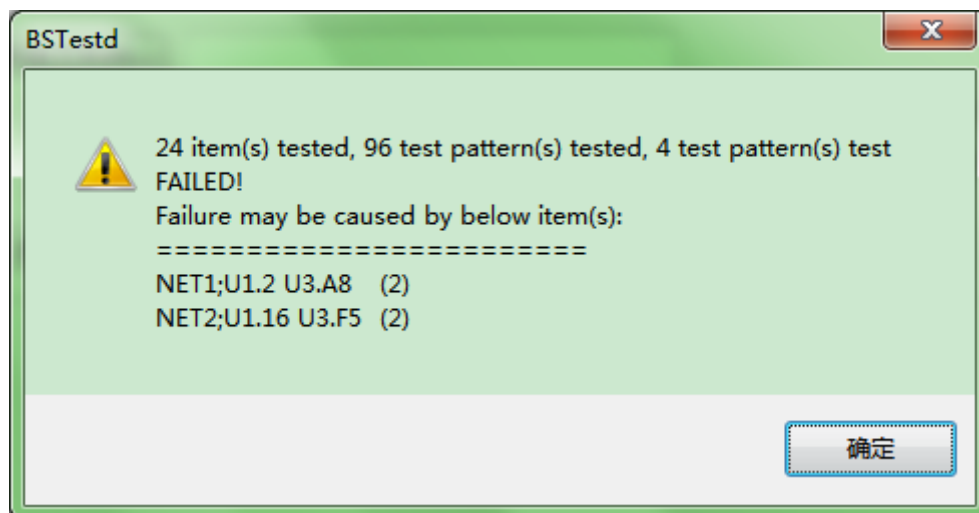


同时在程序窗口将显示详细测试情况。

V2.2.2.1 之后的程序，会在失败时显示失败的网表项目。

V2.3.0.0 之后的程序，会在显示失败网表名称的同时，显示该网表连接的被测器件的序号和管脚位置。

参见下图：



## 不同方式启动测试时界面区别

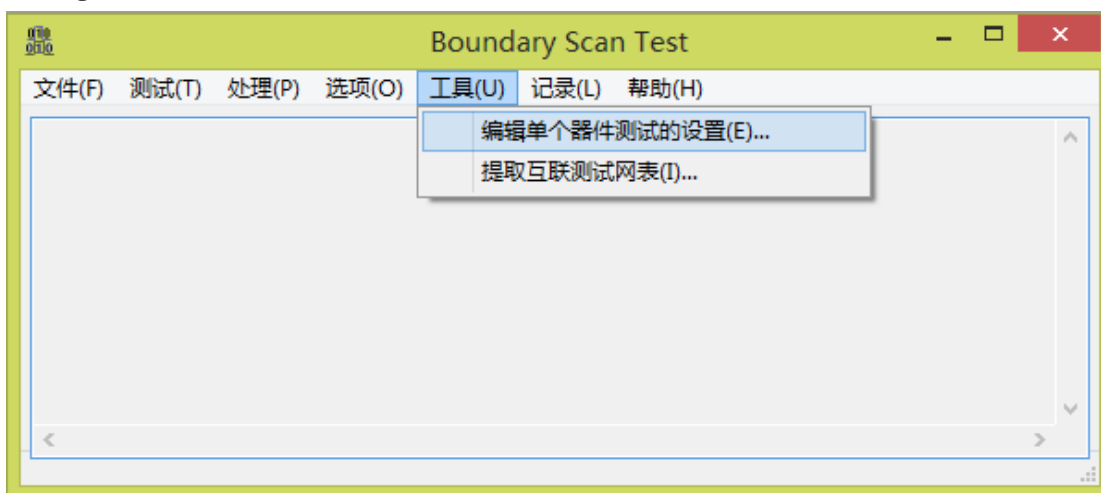
运行方式 界面	菜单 器件互联测试	菜单 批量测试	程序启动自动加载
是否显示编辑网表对话框	是	否	否
是否显示确认可测试网表对话框	是	否	否
是否显示错误信息	是	是	是
是否显示成功信息	是	是	否

## 单个器件测试

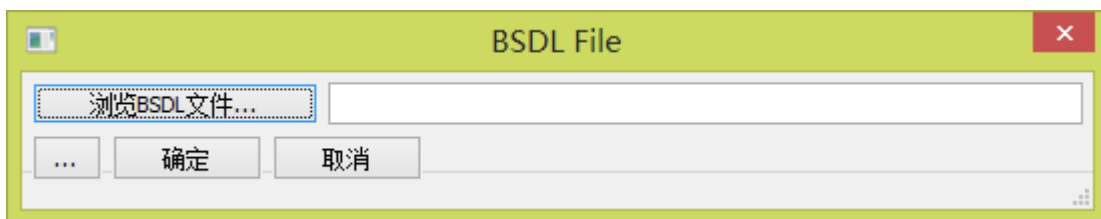
该功能通过读取测试设置文件，自动运行，可以发现管脚（不仅仅是被测器件的管脚，还包括与被测器件管脚相连的其他器件）连锡、跟电源或地短路等情形。

### 测试准备

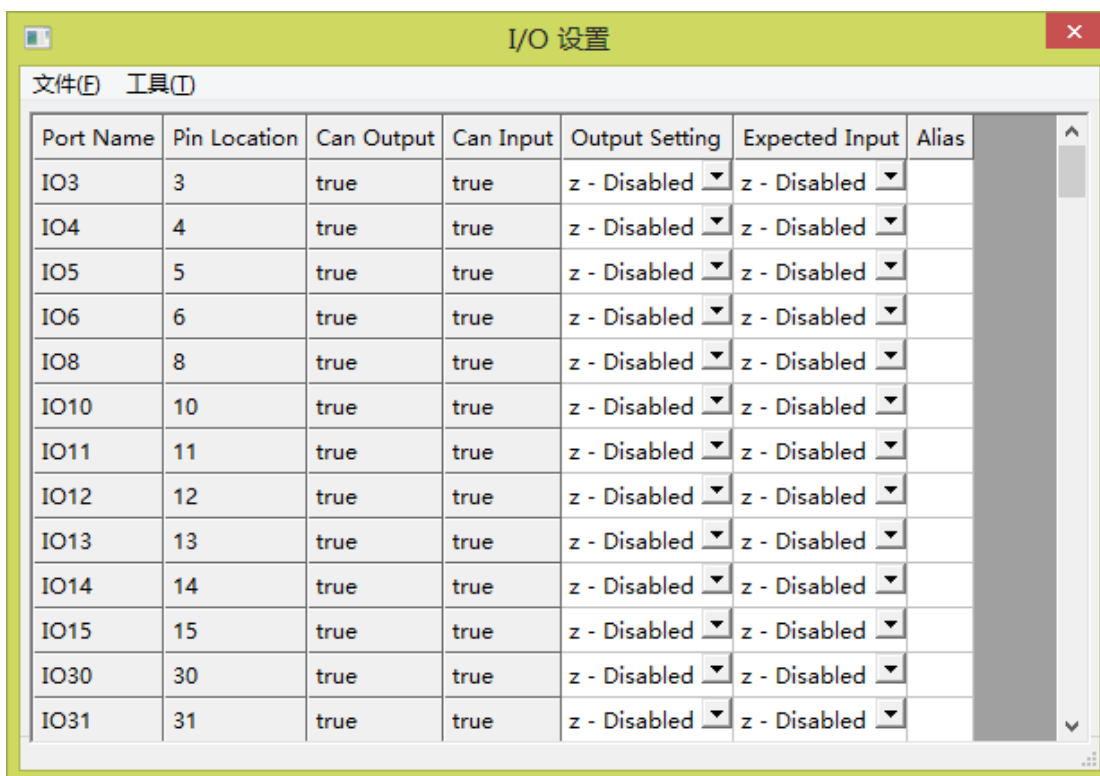
在测试之前，请通过菜单 **工具(Utilities) – 编辑单个器件测试的设置(Edit Single Device Test Setting)...**:



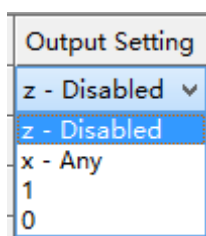
然后弹出 BSDL File 对话框，选择待测器件的 BSDL 文件，点击 **确定(OK)** 按钮：



现在看到 **I/O 设置** 对话框：

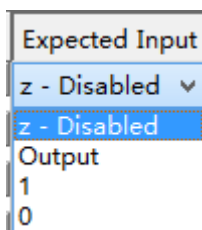


## 输出设置(Output Setting)说明



输出选项	输出设置	备注
z - N/A	不支持输出	
z - Disabled	Cell 可以输出，但是不测试	
x - Any	输出任意值	待测普通管脚应该选择这个选项
1	固定输出 1	让某个管脚维持某个状态，可以用来控制相关电路
0	固定输出 0	让某个管脚维持某个状态，可以用来控制相关电路

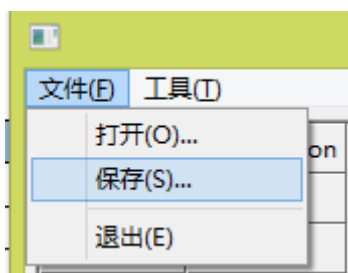
## 期望输入(Expected Input)说明



输入选项	输入设置	备注
z - N/A	Cell 不支持输入	
z - Disabled	Cell 可以输入，但不测试	
Output	跟输出一致	输入应该跟输出一致，待测普通管脚应该选择这个选项。 注意：此时输出必须为输出状态，而不能是 z - N/A 或 z - Disabled。
1	输入恒为 1	该管脚应该一直为 1，其他状态认为有问题。这常见于一些固定状态脚。
0	输入恒为 0	该管脚应该一直为 0，其他状态认为有问题。这常见于一些固定状态脚。

## 保存

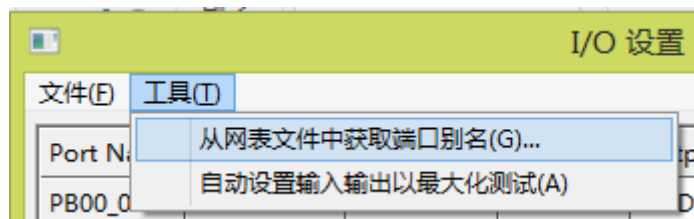
菜单 文件(File) – 保存(Save)...



## 工具

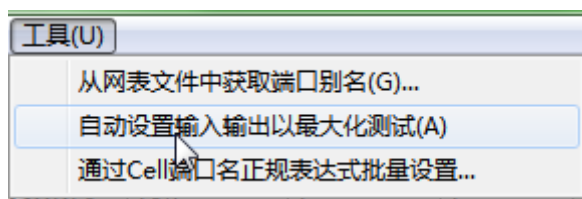
## 别名

为了提供可识别性，可以对每个关键增加一个别名(Alias)。方法菜单 工具(Utilities) - 从网表文件中获取端口别名(Get Alias from Netlist File)...。参见下图：



## 自动设置

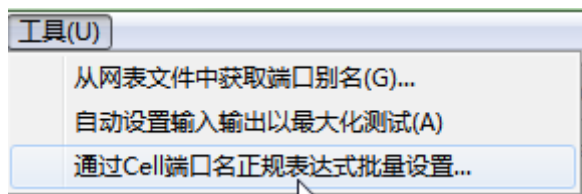
该功能将最大程度的设置可测试的项目。方法菜单 **工具(Utilities) – 自动设置输入输出以最大化测试...**。参见下图：



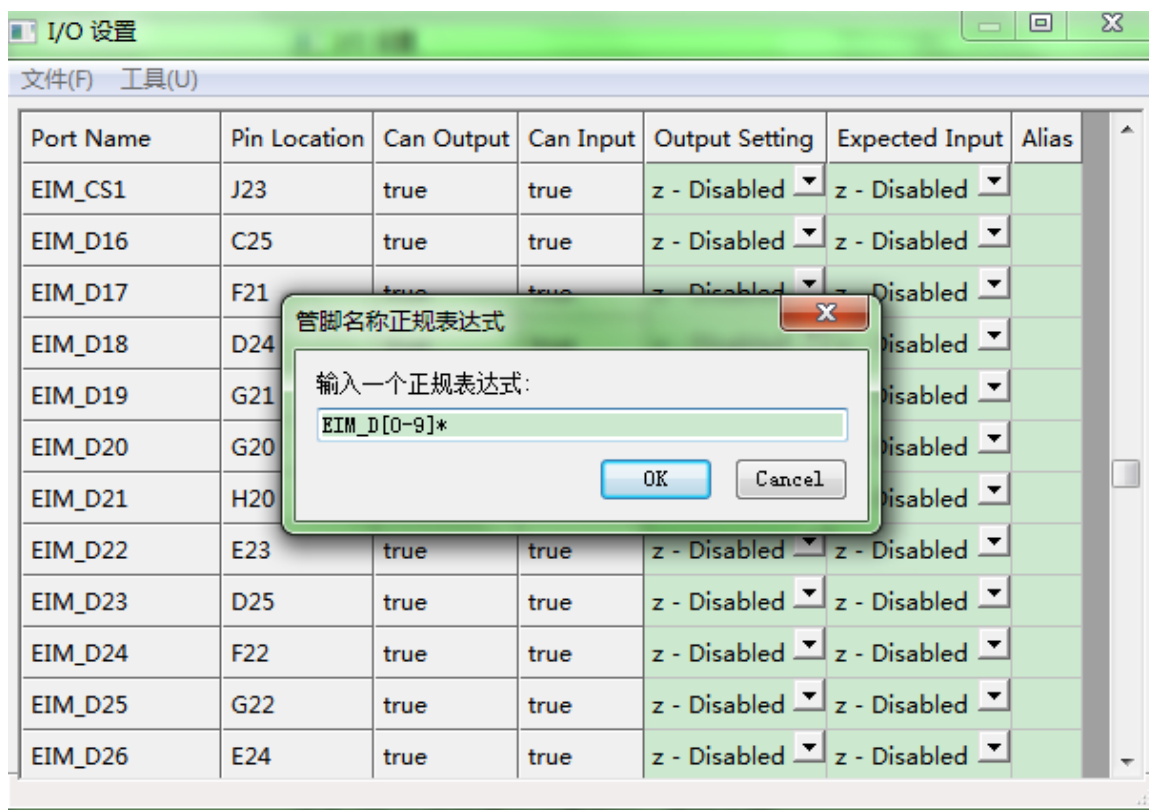
注意：因为每个板卡的设计不同，所以该功能得到的测试设置并不意味着直接可用，请务必根据板卡实际情况逐条确认后方可用于测试。

## 通过正规表达式批量设置

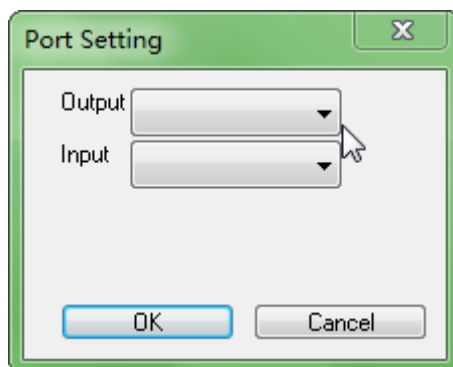
方法菜单 **工具(Utilities) – 通过 Cell 端口名正规表达式批量设置...**。参见下图：



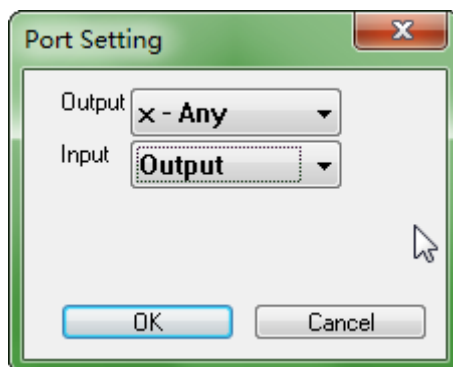
之后请输入要设置的某类端口(管脚)的正规表达式，参见下图：



点击 OK 以后将弹出设置窗口，参见下图：

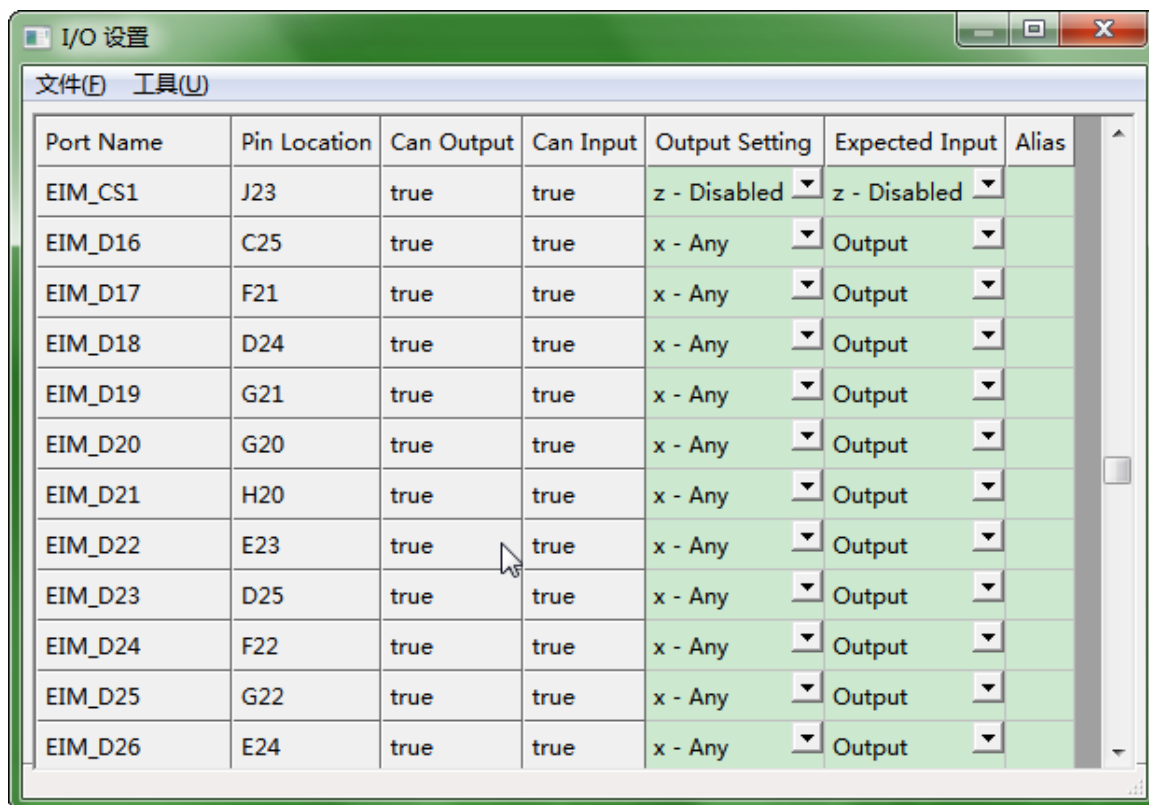


根据实际情况选择为这类端口(管脚)输出和输入设置。本例中，既要输出 0 也要输出 1，输入应该和输出一样，因此设置如下：(请参阅[输出设置\(Output Setting\)说明](#)和[期望输入\(Expected Input\)说明](#)章节)



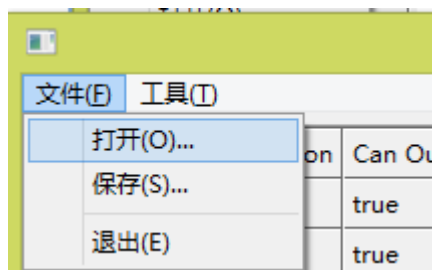
点击 OK，程序将自动对符合正则表达式的端口(管脚) 完成设置，参见下图：





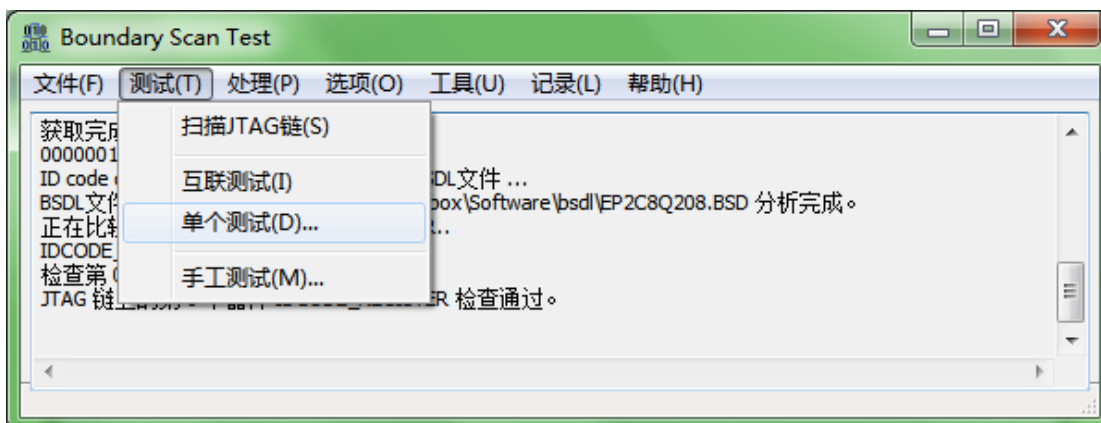
## 修改已经保存的设置

菜单 文件(File) – 打开(Open)...



## 测试步骤

连接好电缆，运行程序，选择菜单 **测试(Test) – 单个测试(Single Device Test)**:



然后将弹出测试设置对话框，参见下图：



**浏览设置文件(Browse Setting File):** 选择在上文 [测试准备](#) 步骤中保存下来的单个器件测试设置文件。

**前置器件指令长度(Prefix Instruction Length):** 整个 JTAG 链上在待测 JTAG 器件之前的其他器件的指令长度，如果有多个这样的器件，长度用“|”符号分割，如“4|5”表示待测 JTAG 器件之前有两个器件，它们的 JTAG 指令长度分别 4 和 5。如果没有前置器件请留空。

**后置器件指令长度(Postfix Instruction Length):** 整个 JTAG 链上在待测 JTAG 器件之后的其他器件的指令长度。其他同“前置器件指令长度”。如果没有后置器件请留空。

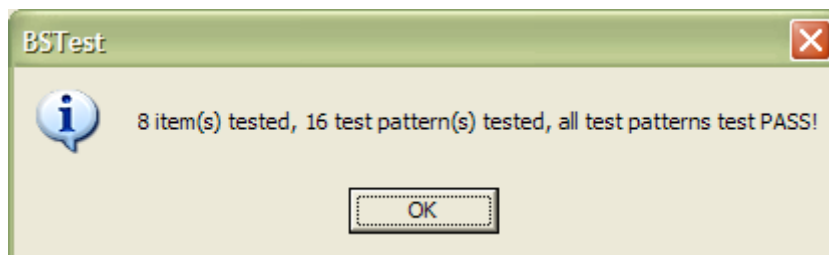
指定好设置文件、待测器件的 BSDL 文件、在 JTAG 链上的位置，然后点击 **确定(OK)** 按钮。

程序测试结束将显示测试结论。

## 举例 1

设置文件包含了某器件的 8 个管脚，对应于数据线 D0 到 D7 共 8 个项目。

## 正常情形

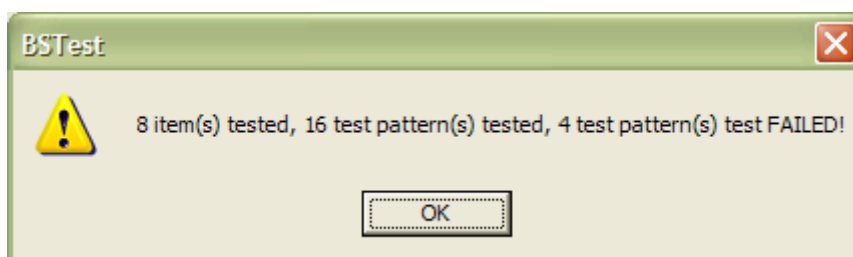


可以看到详细 Log，没有任何错误。

```
Single Device Test is running...
Output 0 Input Result Port
01111111 01111111 G6 /D0
10111111 10111111 H9 /D1
11011111 11011111 H7 /D2
11101111 11101111 H4 /D3
11110111 11110111 G3 /D4
11111011 11111011 H6 /D5
11111101 11111101 H1 /D6
11111110 11111110 I2 /D7
Output 1 Input Result Port
10000000 10000000 G6 /D0
01000000 01000000 H9 /D1
00100000 00100000 H7 /D2
00010000 00010000 H4 /D3
00001000 00001000 G3 /D4
00000100 00000100 H6 /D5
00000010 00000010 H1 /D6
00000001 00000001 I2 /D7
8 item(s) tested, 16 test pattern(s) tested, all test patterns test PASS!
```

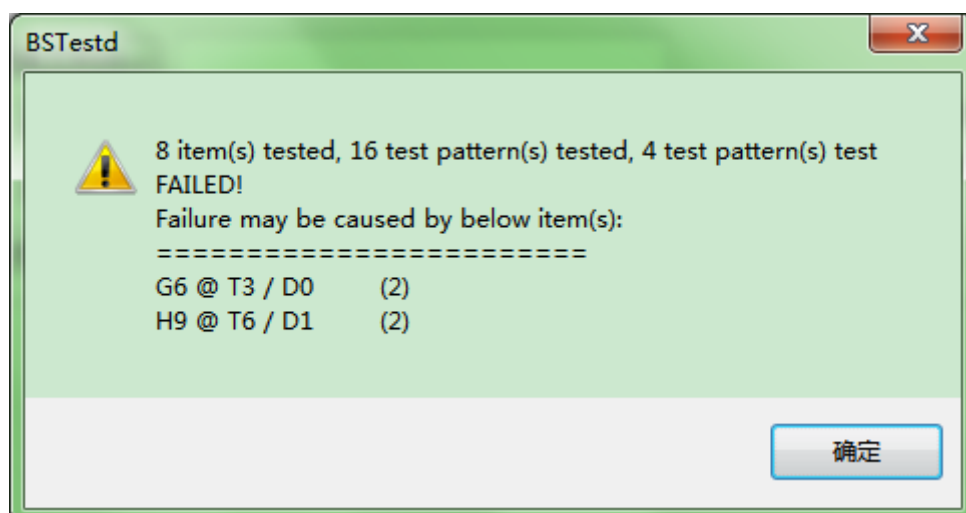
## 管脚之间短路

把 D0 和 D1 短路，再次测试，结果如下：



V2.2.2.1 之后的版本，会显示分析参考结果。

进一步的，V2.3.0.0 之后版本会同时显示管脚位置。格式为：**端口名称 @ 管脚位置 / 别名**。说明：如果管脚位置或者别名为空，则不会显示。参见下图：



查看 Log:

```
Single Device Test is running...
Output 0 Input Result Port
01111111 11111111 ? G6 / D0
10111111 11111111 ? H9 / D1
11011111 11011111 H7 / D2
11101111 11101111 H4 / D3
11110111 11110111 G3 / D4
11111011 11111011 H6 / D5
11111101 11111101 H1 / D6
11111110 11111110 I2 / D7
Output 1 Input Result Port
10000000 11000000 ? G6 / D0
01000000 11000000 ? H9 / D1
00100000 00100000 H7 / D2
00010000 00010000 H4 / D3
00001000 00001000 G3 / D4
00000100 00000100 H6 / D5
00000010 00000010 H1 / D6
00000001 00000001 I2 / D7
8 item(s) tested, 16 test pattern(s) tested, 4 test pattern(s) test FAILED!
```

可以看出 D0、D1 没有通过测试，因为这两个管脚被短路了。

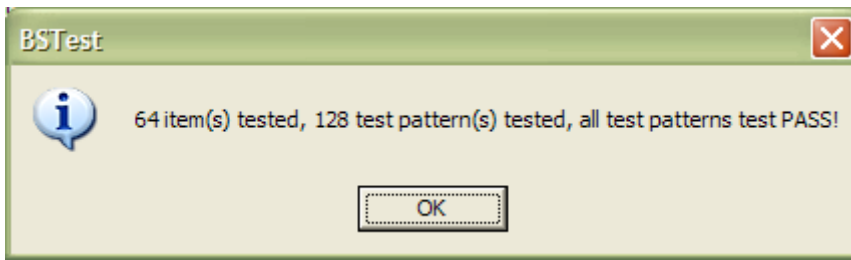
## 说明

不论是在芯片管脚处发生了短路，还是在和被测芯片信号所连接的其他器件（比如排阻、驱动器）发生短路，程序都可以测试发现。

## 举例 2

设置文件包含了某芯片数据线 d(0)到 d(63)共 64 个项目。

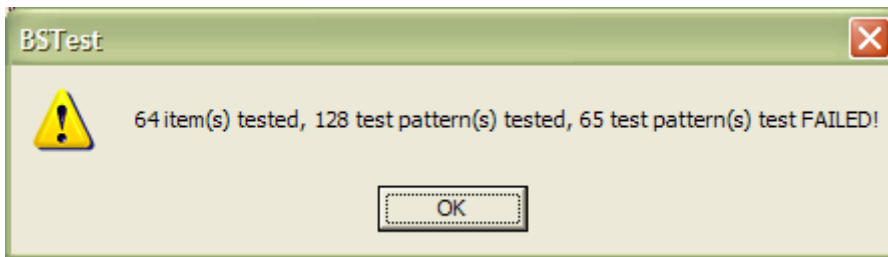
### 正常情形



可以看到详细 Log，没有任何错误。

### 管脚与电源短路

如果我们把某单板的数据线 d(47)跟 3.3V 电源短路，我们再次进行测试。



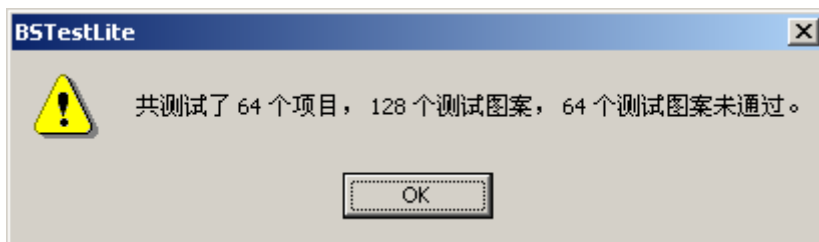
详细 Log 的节选



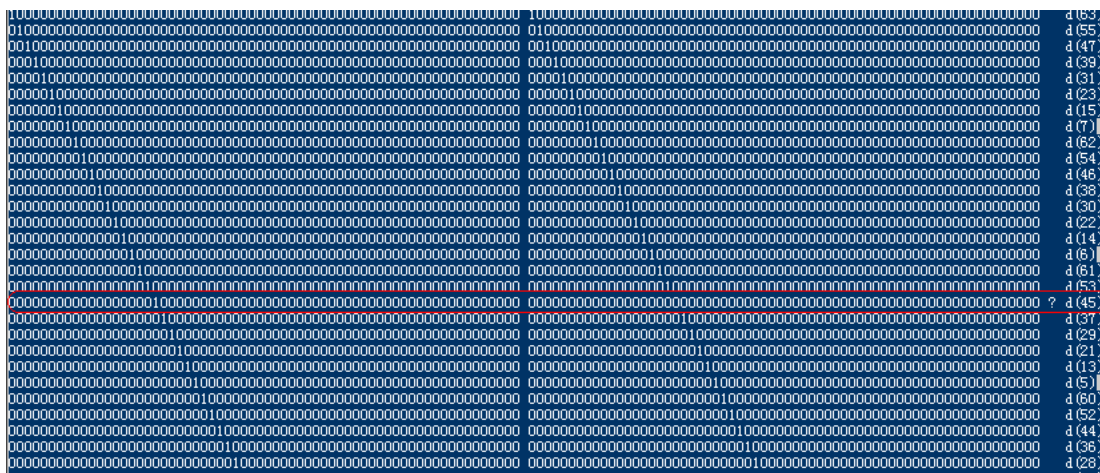
可以清楚的看到，d(47)在输出 0 测试中没有通过，因为跟电源短路了。

## 管脚与地短路

我们把数据线 d(45)跟地短路，我们再次进行测试。结果如下：



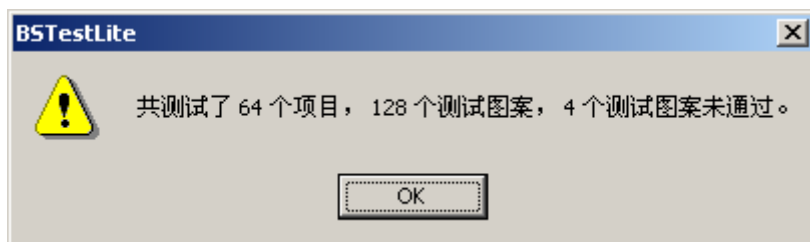
详细 Log 的节选



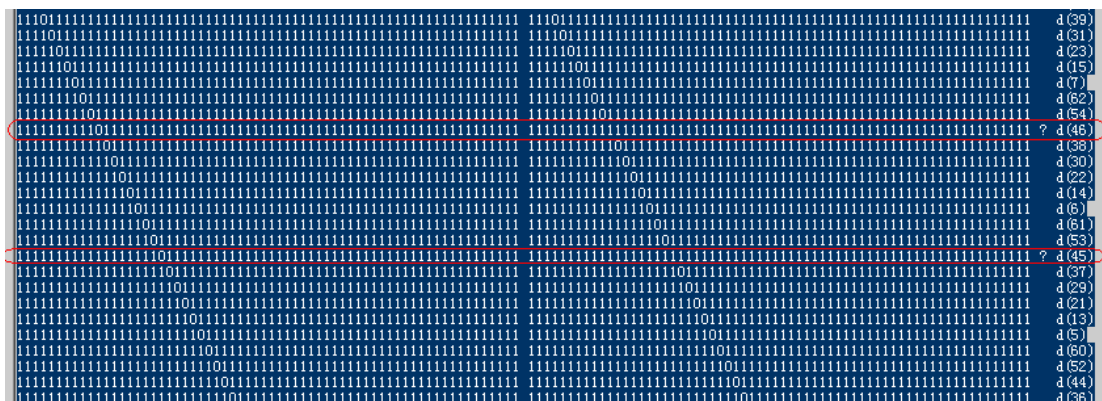
可以清楚的看到，d(45)在输出 1 测试中没有通过，因为跟地短路了。

## 信号管脚之间短路

如果我们把数据线 d(45)和 d(46)短路，我们再次进行测试。结果如下：



输出 0 时详细 Log 的节选



输出 1 时详细 Log 的节选



我们可以看到，不论输出 0 还是输出 1，都有两个没有通过，就是 d(46)跟 d(45)。

## 说明

以上的例子中，每个器件只测试了少部分管脚，实际上，根据具体情况，设置文件可以包含几乎所有的管脚，从而最大的程度对器件进行测试。

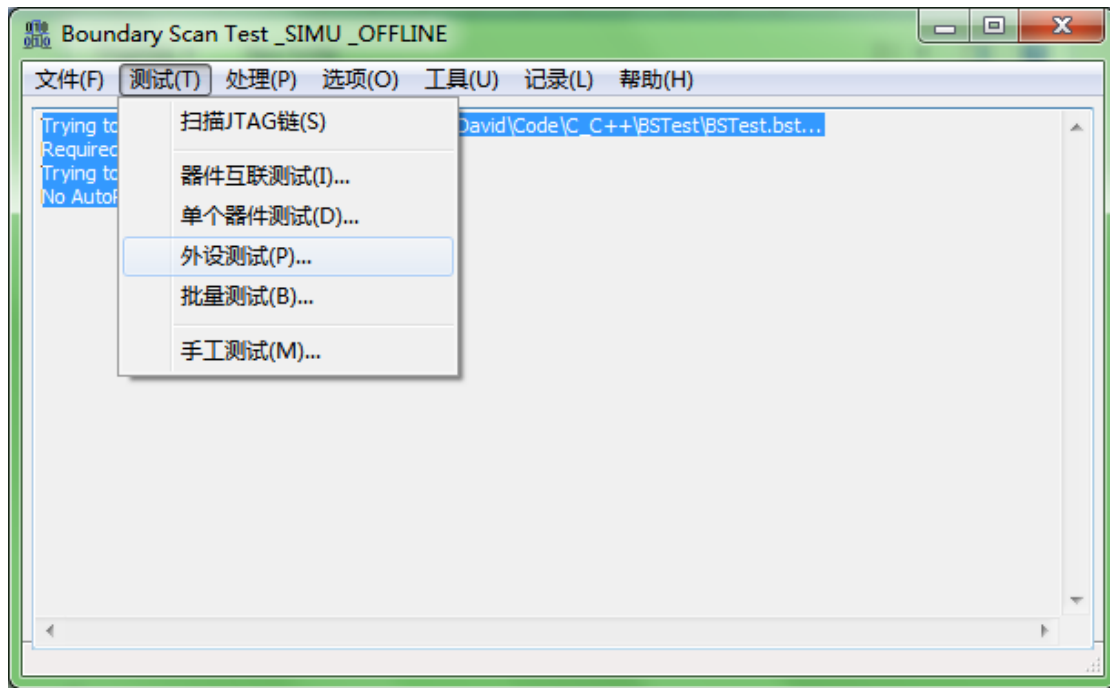
## 外设测试

### 编辑 JFP 配置文件

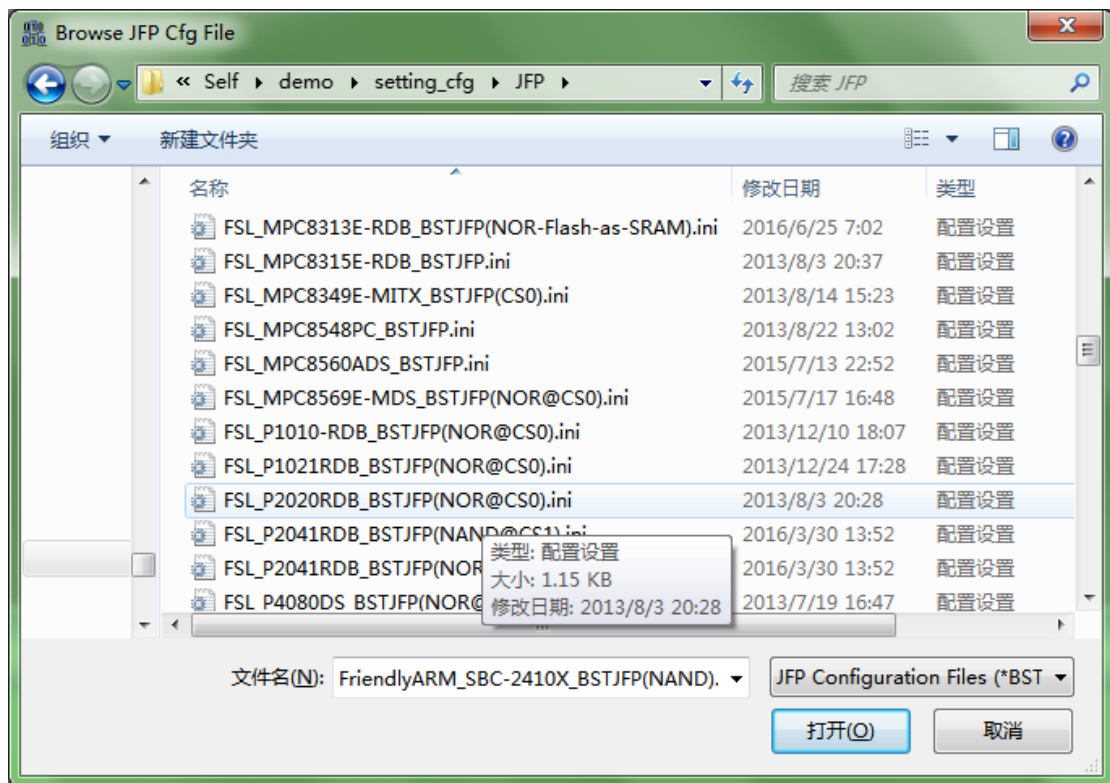
请按照《JFP Edit Cfg File User Manual(CHS).pdf》步骤生成 JFP 配置文件。

## 运行

菜单 **测试(Test)** – **外设测试(Peripheral Test)**，参见下图：



选择 JFP 配置文件。参见下图：



程序将显示执行结果。



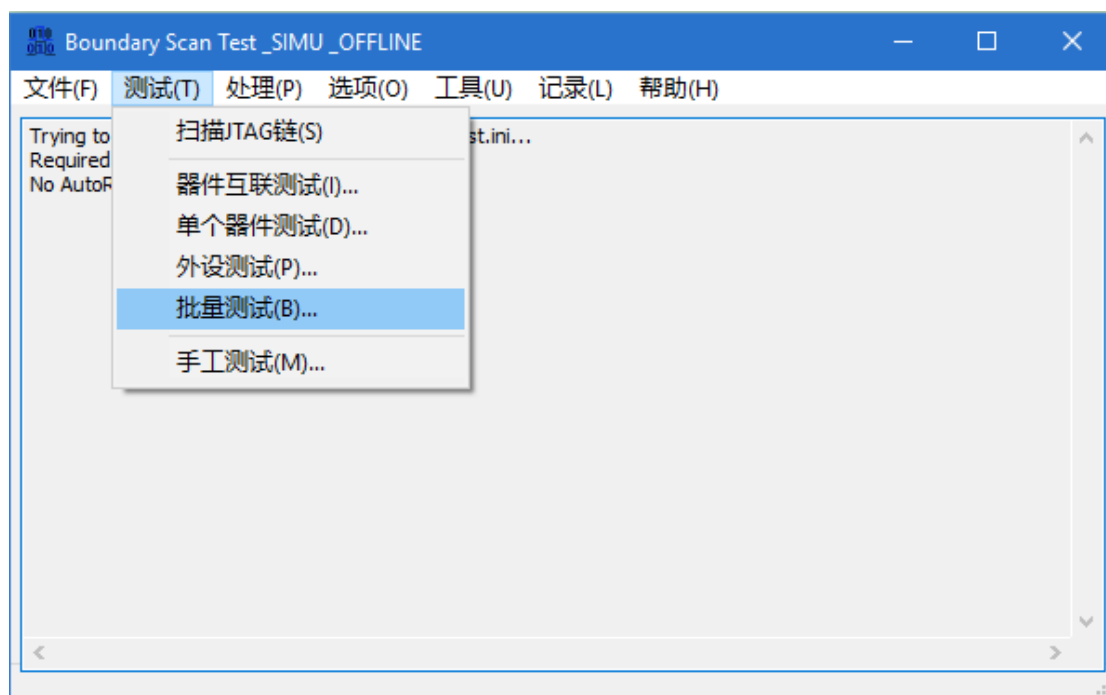
# 批量测试

## 编辑工作空间文件

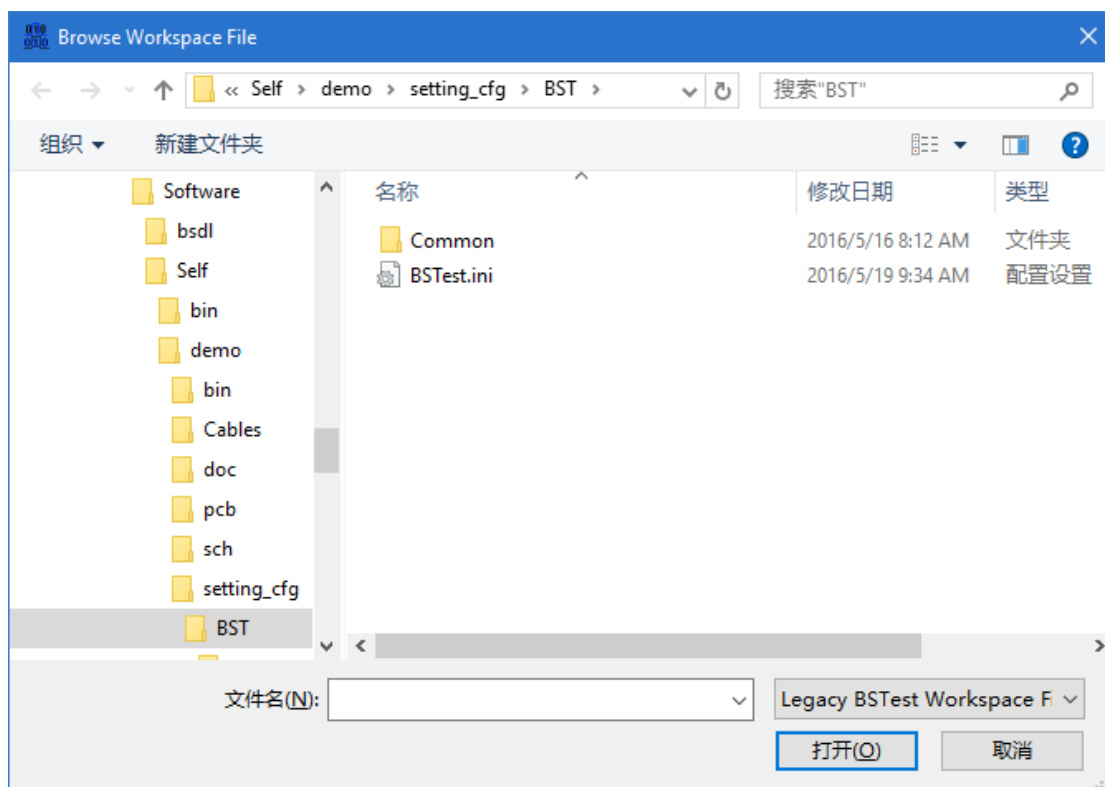
请按照《FileHelper User Manual(CHS).pdf》步骤生成工作空间文件。

## 运行

菜单 **测试(Test)** – **批量测试(Batch Test)**，参见下图：



选择 JFP 配置文件。参见下图：



程序将显示执行结果。

## PLD 加载

加载 PLD 是通过 .svf 或 .vme 格式文件的处理实现的。

需要说明的是：

- PLD 包括 CPLD/EPLD 和 FPGA。
- .svf 不仅可以实现加载，还可以实现测试或者其他功能。

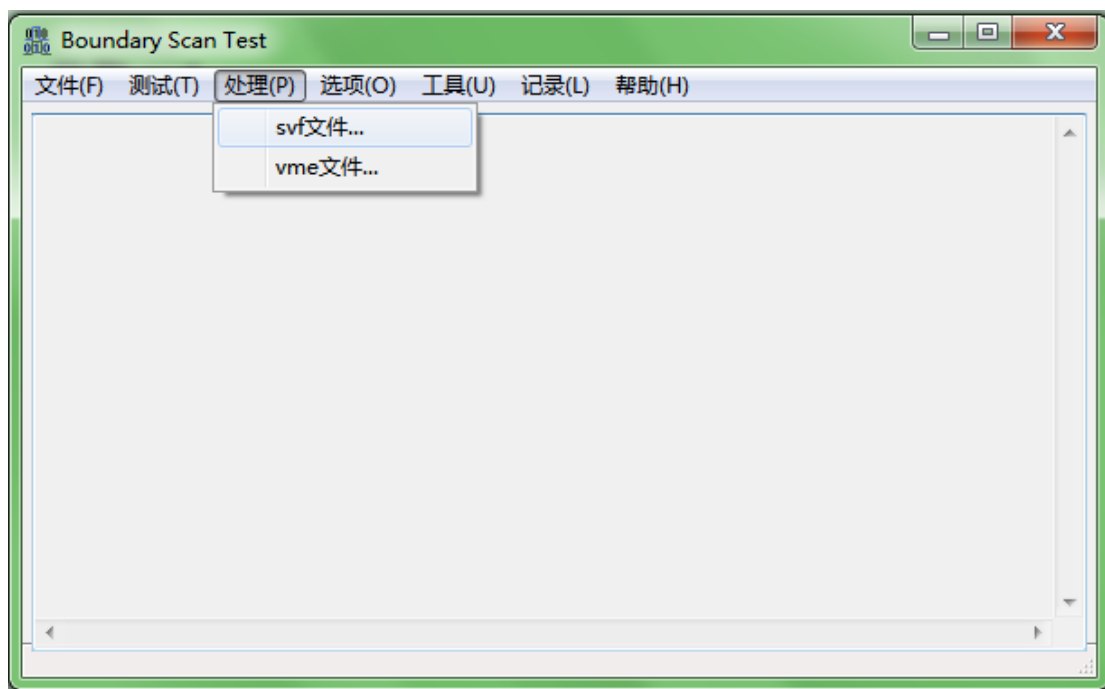
本章节将以加载 PLD 为例说明软件对 .svf 文件的处理。 .vme 文件类似，不赘述。

## 获取.svf 文件

不同 PLD 厂家的软件最终生成的编程文件不同(如 .pof 或者 .jed)，但是都会提供转换为 .svf 的方法。

## 启动软件

软件启动后，选择菜单 **处理(Process) - .svf 文件(.svf File)**，参见下图：

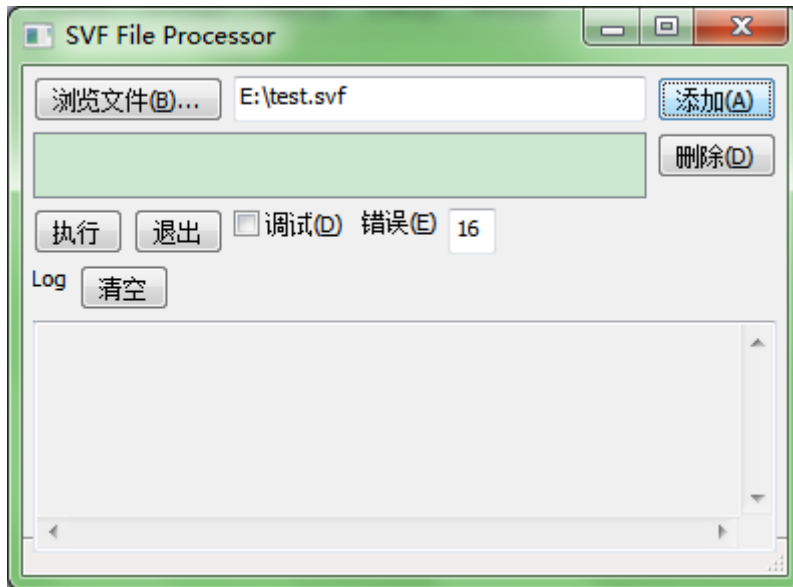


## 选择.svf 文件

在弹出的 **SVF File Processor** 对话框中，点击 **浏览文件...(Browse...)** 按钮选择.svf 文件，参见下图：



选择好.svf 文件后，点击 **添加(Add)** 按钮，参见下图：

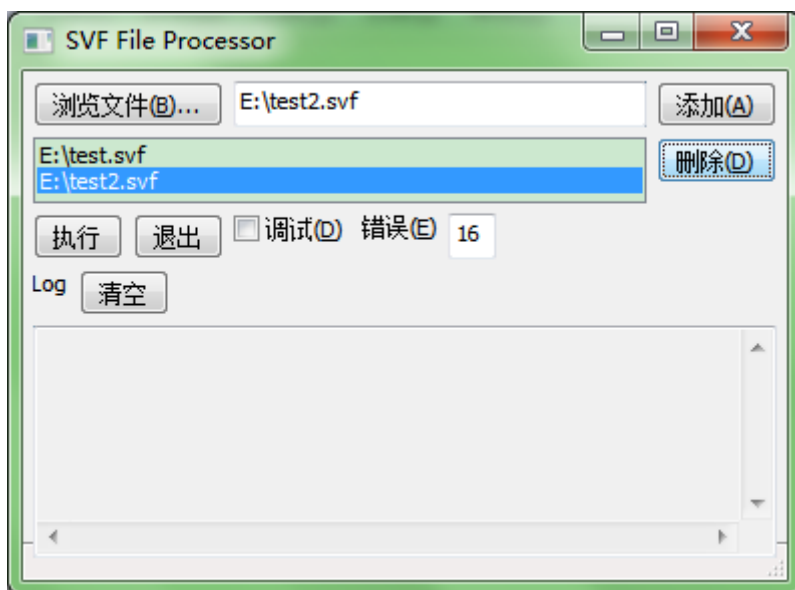


添加后:



可以重复添加多个.svf 文件，程序将依次处理。

如果需要移除某个.svf 文件，则选择该.svf 文件，点击 **删除(Delete)** 按钮，参见下图：



## 执行

全部文件设置正确后，点击 **执行(Run)** 按钮。  
程序将显示执行结果。

## PseudoCLI 特性

如果需要全程自动操作，则可以购买 PseudoCLI 特性。该特性将在自动执行 AutoRun 完成后无需用户干预 GUI 并退出，同时提供返回值供调用者判断。

返回值说明：

- -1: License 出错；
- 0: 所有操作成功完成；
- 大于 0 的整数 n: 第 n（从 1 开始）个 AutoRun 的操作失败；

需要注意的是：程序默认在遇到错误时会弹出错误提示并停在 GUI，因此无人值守的应用，请通过工作空间文件中设置[Option] ExitWhenFail=true 来关闭这一特性。后续操作必须对返回值进行检查。

## Revision History

Date	Version	Author	Changes
2020/4/19			增加 PseudoCLI 特性说明；
2019/8/18			增加单个器件测试 期望输入为 Output 时对输出设置的要求。
2019/1/31			增加 <a href="#">通过正规表达式批量设置</a> 章节；
2016/8/12			增加“批量测试”章节；
2016/8/2			增加互联测试时测试控制文件语法说明；
2016/7/20			<ul style="list-style-type: none"> <li>● 增加“外设测试”章节；</li> <li>● 调整章节顺序为手工测试、互联测试、单个器件测试、外设测试、加载 PLD；</li> </ul>
2015/2/27			<ul style="list-style-type: none"> <li>● 增加：注意：为了保证最佳测试效果和最大测试覆盖率，请尽量保持 FPGA 和 CPLD 在未编程状态，在测试前不要对其进行烧写；类似的，保持 CPU 为未运行状态，测试前不要对其 Boot ROM/Flash 进行烧写。</li> <li>● 因为 V2.4.0.2 开始，手工测试设定 IO 的表格中，输出空间默认为 Edit 而非 Comobox，所以增加“直接填写要输出的图案字符串”说明；</li> </ul>
2014/12/2			<ul style="list-style-type: none"> <li>● 把 V2.2.2.1 的说明放到第一个测试失败的地方；</li> <li>● 增加 V2.3.0.0 新特性；</li> </ul>
2014/11/20			● 增加 <a href="#">不同方式启动测试时界面区别</a> ；
2014/11/14			● 增加了 V2.2.2.1 之后的错误结果（原来仅仅有数量）提示截图；
2014/11/8			● 支持选择网表格式，修改屏幕截图和参数说明；删除配置文件截图；
2014/11/7			<ul style="list-style-type: none"> <li>● 程序增加配置保存功能，在<a href="#">技巧</a>章节说明；</li> <li>● 删除了原来配置的文本和手工修改说明；</li> </ul>
2014/11/6			<ul style="list-style-type: none"> <li>● 撤销 2014/11/5 的提醒，改为说明“有些不可以测试网表，比如 GND VCC TDI TDO 之类的，程序会予以剔除，无需理会。”；</li> <li>● 同时增加说明可以删除不想测试的网表；</li> <li>● 原文增加网表从正文调整为列表；</li> </ul>
2014/11/5			● <a href="#">互联测试</a> 过滤网表时增加提醒“注意：必须剔除掉不可以测试网表，比如 GND VCC TDI TDO 之类的。”
2014/11/4			● <a href="#">互联测试</a> 中 <a href="#">测试设置</a> 增加了一个实例；
2014/11/3			● 增加 <a href="#">单个器件测试</a> 的 <a href="#">测试准备</a> ；
2014/6/9			● 页脚上方增加横线；
2013/7/30			● 格式微调；
2013/7/28			<ul style="list-style-type: none"> <li>● 将部分英文界面的截图替换为中文界面的截图；</li> <li>● 修改了部分笔误；</li> </ul>
2012/11/16			● 页脚增加主页链接；
2012/9/25			● 修改笔误（“参加”应为“参见”）；

			<ul style="list-style-type: none"> <li>● 文字修改以更通顺;</li> <li>● 修正了 <a href="#">互联测试 使用方法</a> 章节的 <b>中置器件指令长度</b> 的说明;</li> <li>● “请先获取 JTAG 链上所有器件的 BSDL 文件, 放到与程序同一目录下” 修改为“请先获取 JTAG 链上待测器件的 BSDL 文件, 放到与程序同一目录下; 对于链上的其它器件, 如果有 BSDL 文件最好, 如果没有则需要知道每个器件的 JTAG 指令长度。”</li> </ul>
2012/9/25			<ul style="list-style-type: none"> <li>● 增加“<a href="#">PLD 加载</a>”章节;</li> </ul>
2012/5/4			<ul style="list-style-type: none"> <li>● 软件升级, 更新截图, 增加“前置器件指令长度”“中置器件指令长度”“后置器件指令长度”的说明;</li> </ul>
2012/3/28			<ul style="list-style-type: none"> <li>● <a href="#">输出设置</a>增加说明: 比如输出内容是“01”则输出占空比 1:1 的方波, 而“0100”则输出占空比是 1:3 的波形。</li> </ul>
2012/3/19			<ul style="list-style-type: none"> <li>● 增加主界面说明;</li> <li>● 调整目录, 提升原有的标题级别;</li> <li>● 将原来的手工测试中的步骤和准备放到新的<a href="#">测试准备</a>章节;</li> <li>● 将手工测试的输入输出设置放到正常步骤中而不是留在最后;</li> </ul>
2012/3/8			<ul style="list-style-type: none"> <li>● 增加 Xilinx ML505 板上实际扫描结果 log 文件;</li> </ul>
2012/2/22			<ul style="list-style-type: none"> <li>● 修改错别字;</li> </ul>
2012/2/21			<ul style="list-style-type: none"> <li>● 增加页眉页脚;</li> </ul>
2012/2/7			<ul style="list-style-type: none"> <li>● 因为不同版本的程序的文件名可能不同, 故删除具体程序文件名;</li> </ul>
2012/2/1			<ul style="list-style-type: none"> <li>● 删除具体程序文件名;</li> </ul>
			<ul style="list-style-type: none"> <li>● 初始版本</li> </ul>