Contents

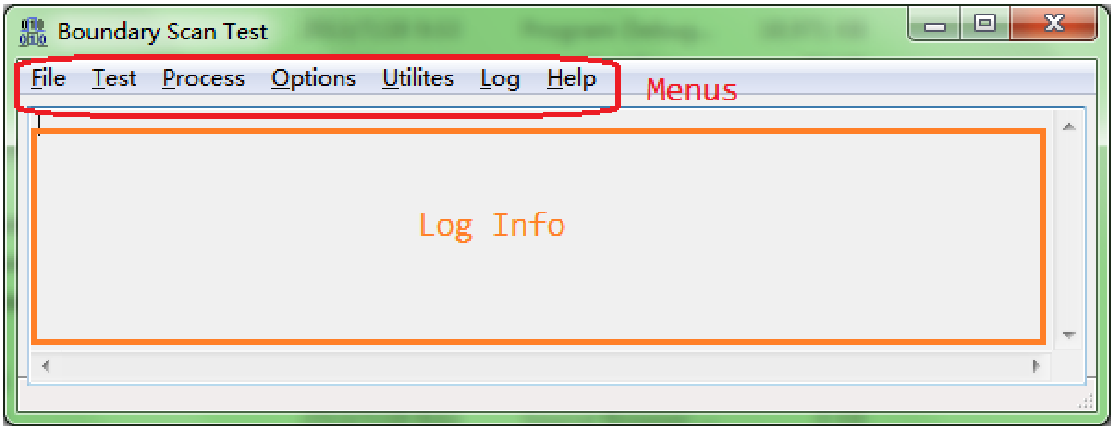
Main User Interface 3
Before Start 3
Set TCK Frequency 5
Manual Test 6
    Preparation 6
        If you have BSDL file for every device 7
        If you don’t get BSDL file for every device 8
            Prefix Instruction Length 8
            Post Instruction Length 8
    Setting 9
        Operation 10
        I/O 10
            Utility for Setting I/O 10
            Utilities 11
                Lookup BSC No. by Pin Location 11
                Lookup BSC No. by Port Name 12
            Output Setting 13
            Input Setting 13
            More Devices… 14
    Test 14
Inter-Device Test 15
    Steps 15
    About the Parameters 16
    An Example 17
        Tips 17
    Run the Test 17
    UI Difference Under Different Launching Mode 19
Single Device Test 20
    Preparation 20
        Output Setting 21
        Expected Input 22
        Save 22
        Utilities 23
            Alias 23
            Auto Set to Max Testable 23
            Batch Set with Port Name Regular Expression 23
        Edit Saved Setting File 25
    Steps 25
    Example 1 27
        Test on Good Board 27
        Short-Circuit Issue 27
        More… 28
    Example 2 28

Test on Good Board 29
Short Circuit with Power........ 29
Short Circuit with GND 30
Short-Circuit Between Pins........ 30
More… 31
Peripheral Test........ 31
Edit JFP Configuration File........ 31
Run........ 31
Batch Test........ 33
Edit Workspace File 33
Run........ 33
PLD Configuration........ 34
Get .svf File........ 34
Start 35
Set .svf File 35
Run........ 37
PseudoCLI Feature........ 37

Note: There may be some changes between versions. So, your software may look a bit different from this manual.

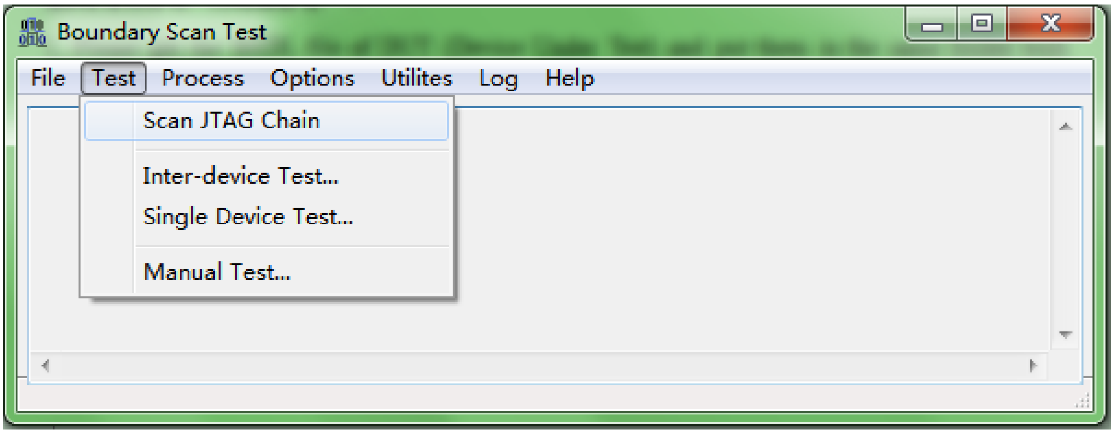
Attention: To get better test result and wider test coverage, please keep FPGA, CPLDs blank when testing, and do not program or configure them before test. And please keep CPUs in idle status, i.e. do not program their Boot ROMs or Flashs before test.

Main User Interface

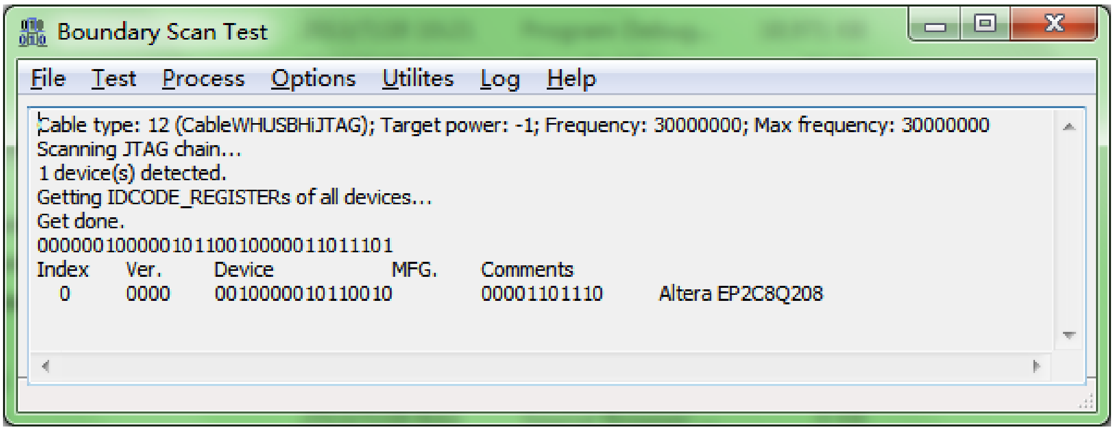


Before Start

1, Please get the BSDL file of DUT (Device Under Test) and put them in the same folder with software. If there multi devices in JTAG chain, please try to get all BSDL files. And if you could not get every BSDL file, you must know each device’s JTAG instruction length.
2, Power off the target board.
3, Connect the cables between computer and target board.
4, Power up the target board;
5, Click to run the software.
6, When you see the main UI dialog, please select menu **Test** – **Scan JTAG Chain** to check how many devices detected by the software. See screenshot below:



If any device is detected, device ID and other info will be displayed. See screenshot below:

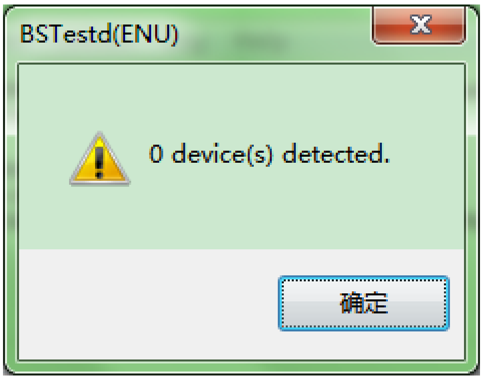


```
Cable type: 12 (CableWHUSBHiJTAG); Target power: -1; Frequency: 30000000; Max frequency: 30000000
Scanning JTAG chain…
1 device(s) detected.
Getting IDCODE_REGISTERs of all devices…
Get done.
00000010000010110010000011011101
Index     Ver.      Device                MFG.      Comments
  0       0000      0010000010110010              00001101110      Altera EP2C8Q208
```

Another log example of Xilinx ML505 reference design board.

```
Cable type: 10; Target power: -1; Frequency: 6000000
Scanning JTAG chain…
5 device(s) detected.
Getting IDCODE_REGISTERs of all devices…
Get done.
11110101000001011001000010010011
11110101000001011001000010010011
01011001011000001000000010010011
00001010000000000001000010010011
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
```

If no device detected, a message box is popped, and please refer to troubleshooting below.



BSTestd(ENU)

0 device(s) detected.

If no device detected, please check:

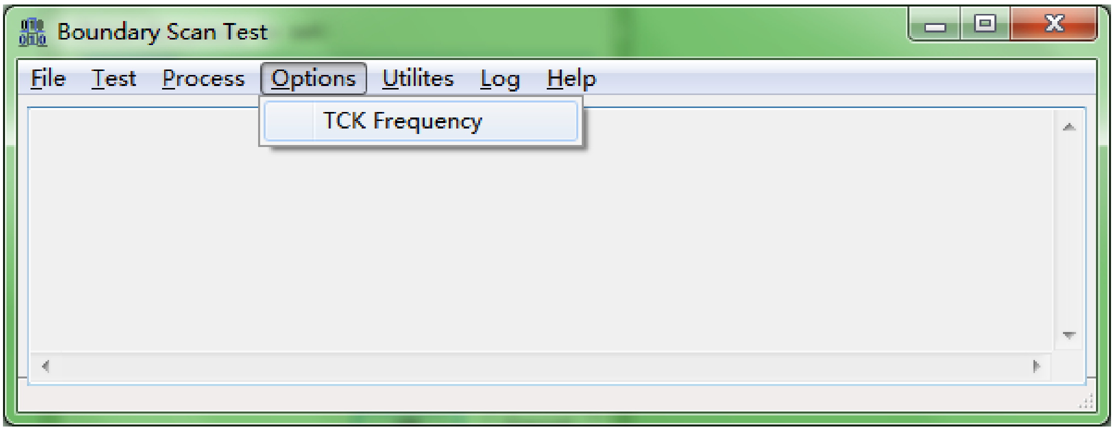
- Cable is connected correctly;
- JTAG circuit of target board works;
- Target board is powered OK;

If the device list doesn't match with the board, please read target board's manual to confirm jumper settings or configurations. If you are using WH-USB-HiJTAG cable, you could try to slow the TCK frequency, please refer to Set TCK Frequency.

Only when device list detected by the software matches with the board, you can do any test with the software.

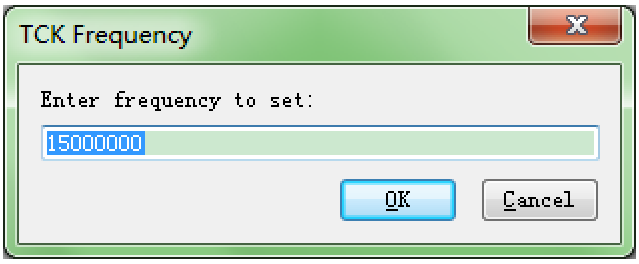
Set TCK Frequency

Select menu **Options** / **TCK Frequency**. See screenshot below:



Input the frequency you want to set. For example, if you want to change TCK frequency to 15MHz, you should input 15000000.

See screenshots below:



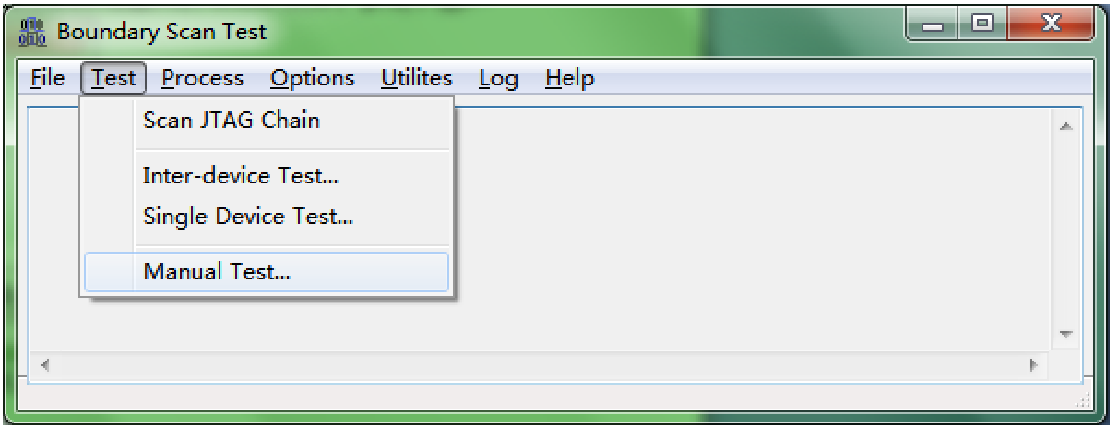
To WH-USB-HiJTAG cable, the frequency could be 30000000, 1500000, 10000000, and 6000000 and so on.
To WH-USB-JTAG cable, the frequency could be 6000000, 3000000 and so on.

Manual Test

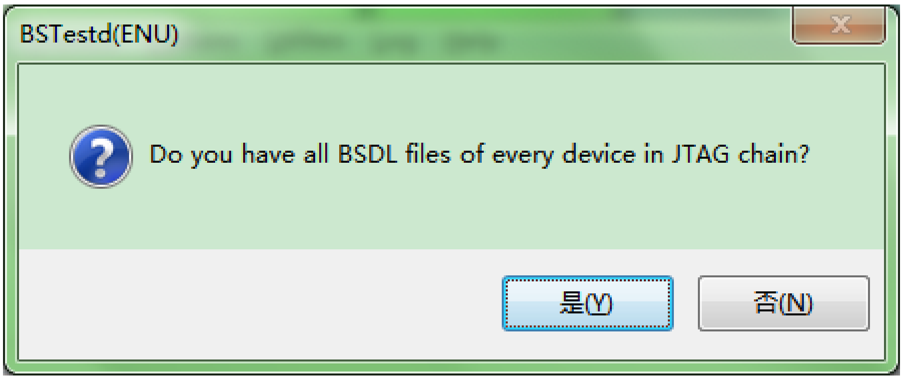
The system can display status of JTAG chip's pins (something like an oscilloscope or logic analyzer), and can control the chip to output a user defined waveform (something like a signal generator). You can find issues in welding (open, short, etc.) or PCB production by analyzing the phenomenon. Also, it can be used as debugging tool.

Preparation

Select menu **Test** – **Manual Test**. See screenshot below:



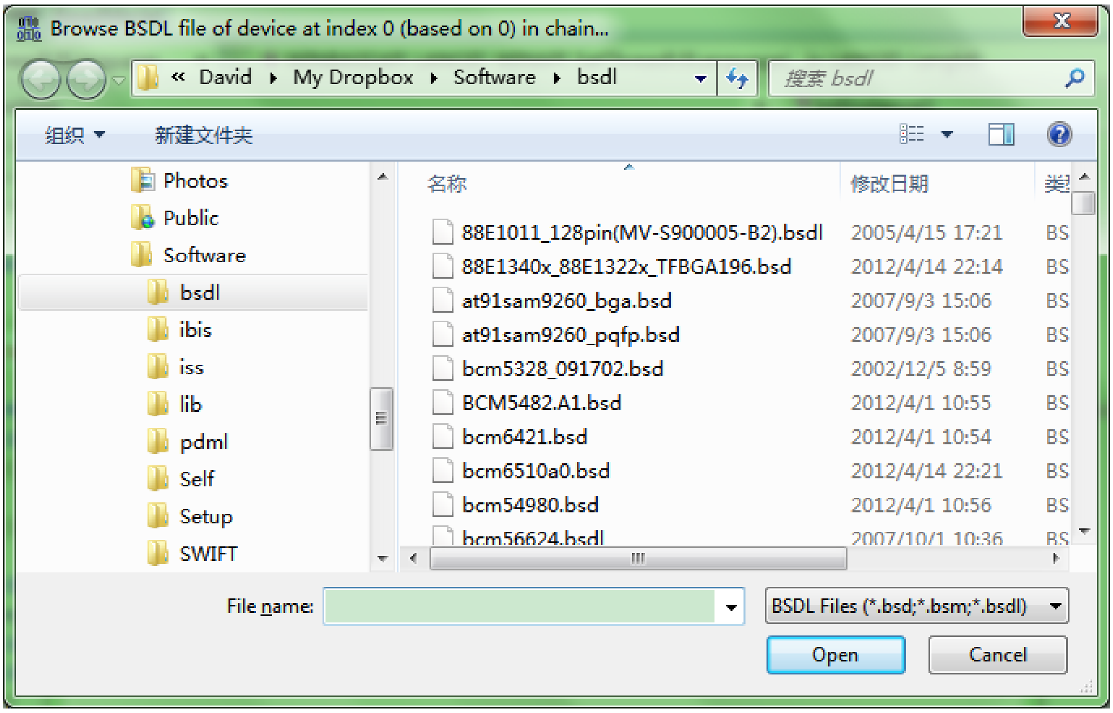
You will be asked whether you have BSDL file for every device in JTAG chain. See screenshot below:


BSTestd(ENU)

Do you have all BSDL files of every device in JTAG chain?

If you have BSDL file for every device

The software will let you choose BSDL file of device one by one. See screenshot below:


Browse BSDL file of device at index 0 (based on 0) in chain...

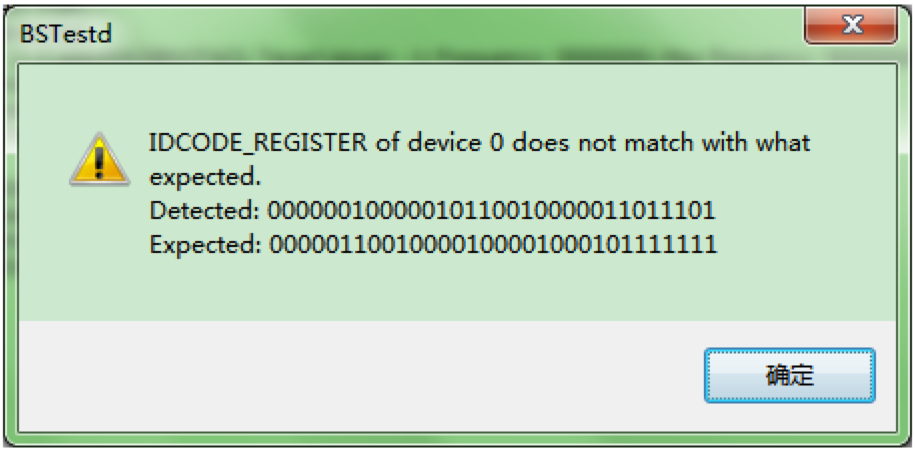
« David ▸ My Dropbox ▸ Software ▸ bsdl

名称	修改日期	类
88E1011_128pin(MV-S900005-B2).bsdl	2005/4/15 17:21	BS
88E1340x_88E1322x_TFBGA196.bsd	2012/4/14 22:14	BS
at91sam9260_bga.bsd	2007/9/3 15:06	BS
at91sam9260_pqfp.bsd	2007/9/3 15:06	BS
bcm5328_091702.bsd	2002/12/5 8:59	BS
BCM5482.A1.bsd	2012/4/1 10:55	BS
bcm6421.bsd	2012/4/1 10:54	BS
bcm6510a0.bsd	2012/4/14 22:21	BS
bcm54980.bsd	2012/4/1 10:56	BS
bcm56624.bsdl	2007/10/1 10:36	BS

File name:

BSDL Files (*.bsd;*.bsm;*.bsdl)

The software will parse each device’s BSDL, and compare IDCODE in BSDL with detected IDCODE. If mismatch found, a message box is popped like this:


BSTestd

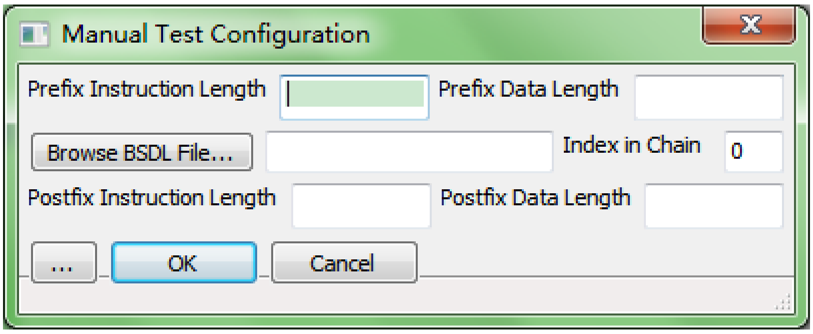
IDCODE_REGISTER of device 0 does not match with what expected.
Detected: 00000010000010110010000011011101
Expected: 00000110010000100001000101111111

Note: Some JTAG device doesn't implement IDCODE instruction, and no IDCODE register in BSDL. So, the software could not judge whether BSDL file and device is matched or not. It's just a reminder.

If check passed, you will come to **Manual Test** dialog.

If you don't get BSDL file for every device

You will see a configuration dialog:



Manual Test Configuration

Prefix Instruction Length

Prefix Data Length

Browse BSDL File...

Index in Chain 0

Postfix Instruction Length

Postfix Data Length

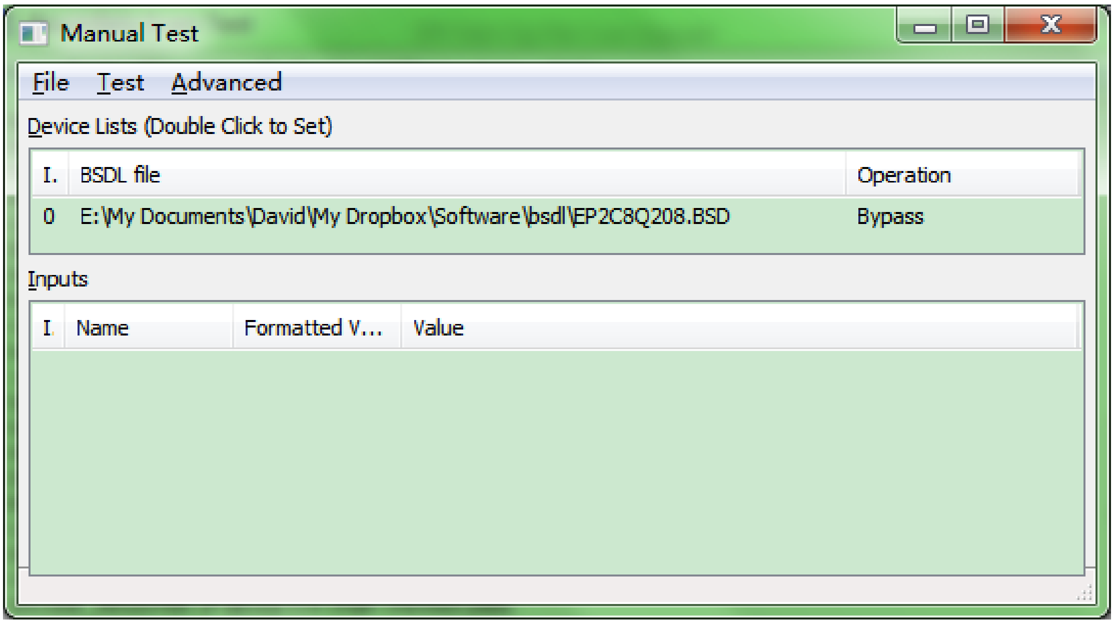
Prefix Instruction Length

Please input JTAG instruction length of all other devices ahead of DUT. If there are more than one devices, please split them with '|' character. For example, '4|5' means there are two devices ahead of DUT in JTAG chain, and device at index 0 has a 4-bit JTAG instruction while device at index 1 has a 5-bit JTAG instruction. And you could find that the DUT is at index 2. Must left blank when no prefix device.

Post Instruction Length

Please input JTAG instruction length of all other devices behind of DUT. Refer to Prefix Instruction Length. Must left blank when no postfix device.

Setting

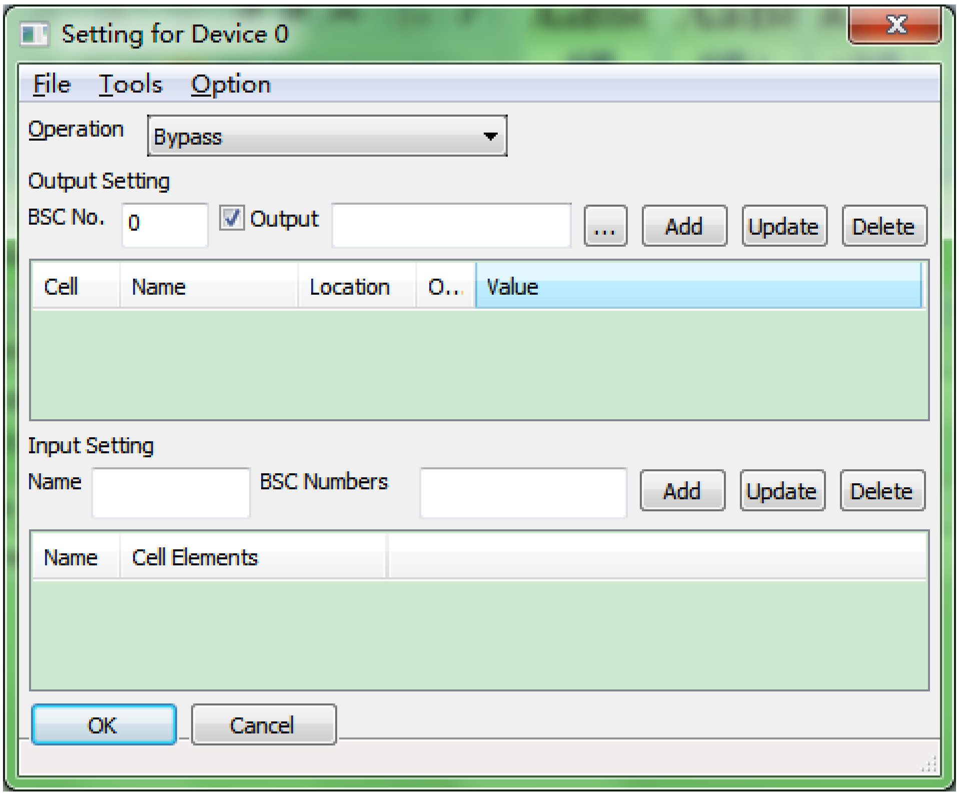


I.	BSDL file	Operation
0	E:\My Documents\David\My Dropbox\Software\bsdl\EP2C8Q208.BSD	Bypass

I.	Name	Formatted V...	Value

By default the software will set operation to **Bypass**.

- Double click DUT in **Device Lists** to edit settings. You will see **Setting for Device n** dialog (n means index in JTAG chain and index begins with 0).



Operation: Bypass

BSC No.: 0 ☑ Output

Cell	Name	Location	O...	Value

Name: BSC Numbers:

Name	Cell Elements

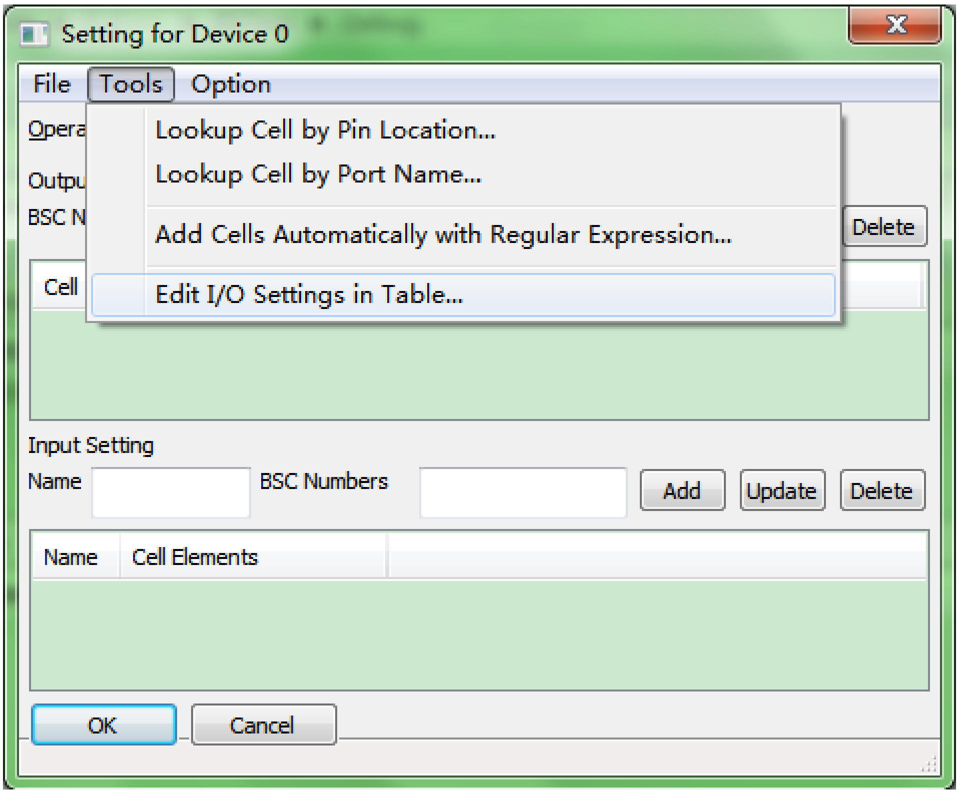
Operation

Change **Operation** to **Test** (for those devices that will not be tested, please keep '**Bypass**').

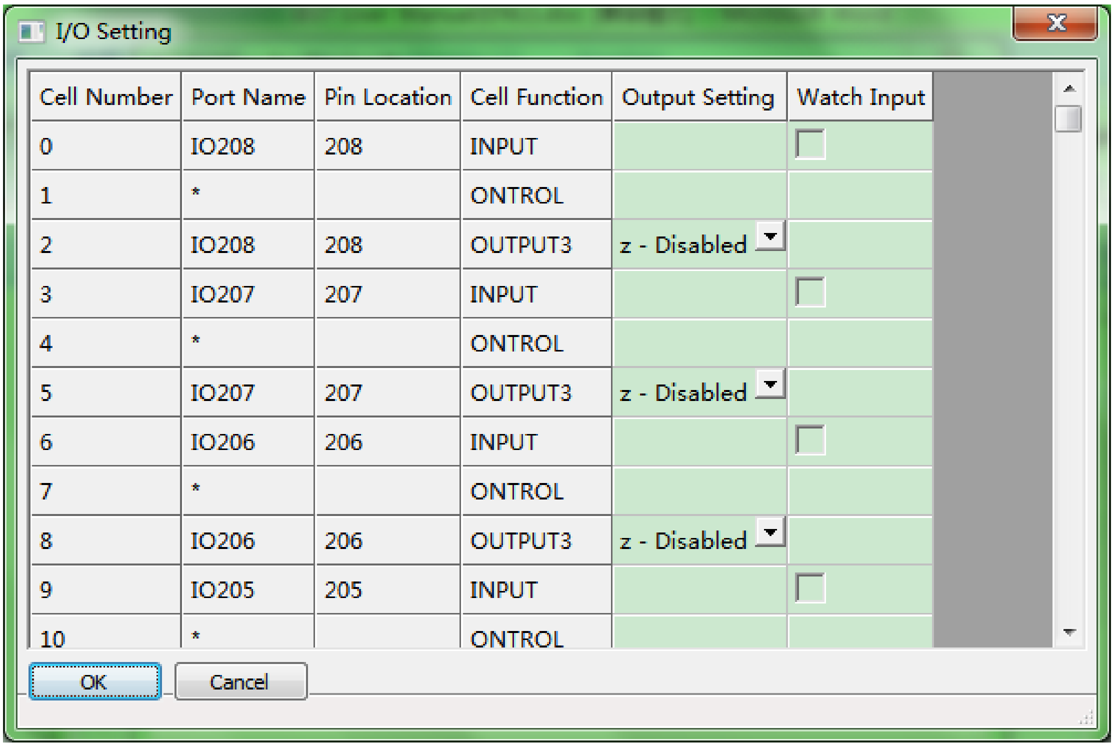
I/O

Utility for Setting I/O

Select menu **Utilities** / **Edit I/O Settings in Table**… . See screenshot below:


Setting for Device 0
File
Tools
Option
Lookup Cell by Pin Location...
Lookup Cell by Port Name...
Add Cells Automatically with Regular Expression...
Edit I/O Settings in Table...
Opera
Outpu
BSC N
Delete
Cell
Input Setting
Name
BSC Numbers
Add
Update
Delete
Name
Cell Elements
OK
Cancel

You will see **I/O Setting** dialog popped:



Cell Number	Port Name	Pin Location	Cell Function	Output Setting	Watch Input
0	IO208	208	INPUT		□
1	*		ONTROL		
2	IO208	208	OUTPUT3	z - Disabled	
3	IO207	207	INPUT		□
4	*		ONTROL		
5	IO207	207	OUTPUT3	z - Disabled	
6	IO206	206	INPUT		□
7	*		ONTROL		
8	IO206	206	OUTPUT3	z - Disabled	
9	IO205	205	INPUT		□
10	*		ONTROL		

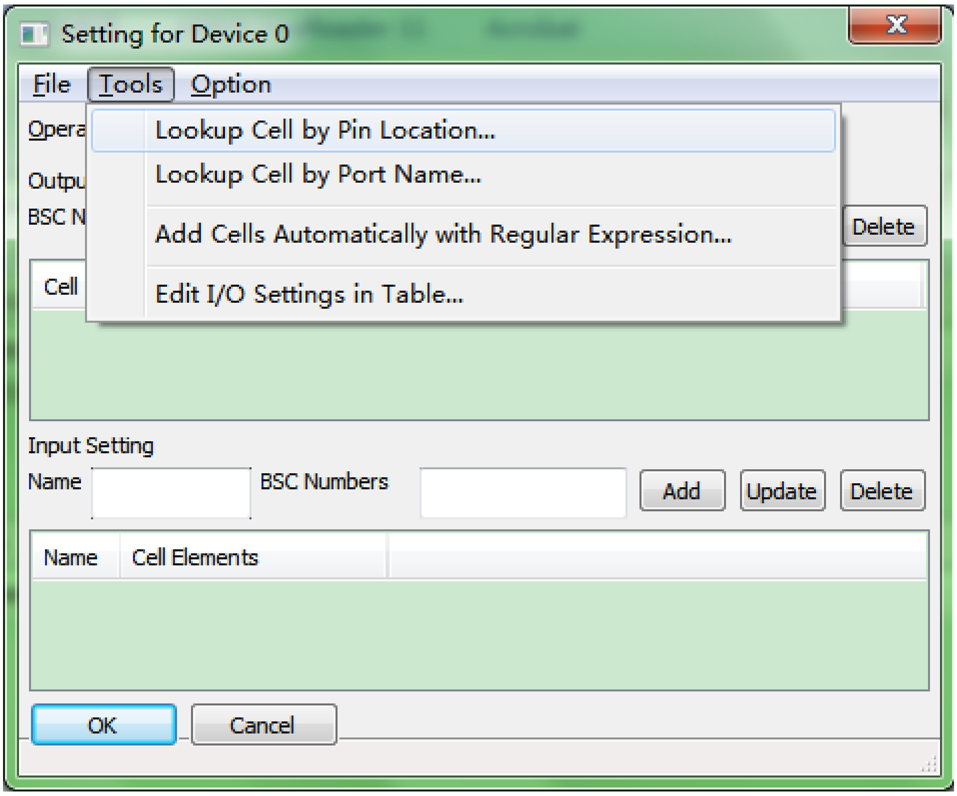
If you want to control a pin to output, please type the output pattern string in Edit box of the cell, or please choose output value from Comboboxo of the cell.
If you want to observe input value of a pin, please tick the checkbox.

When finished, please click **OK** to return **to Setting for Device n** dialog.

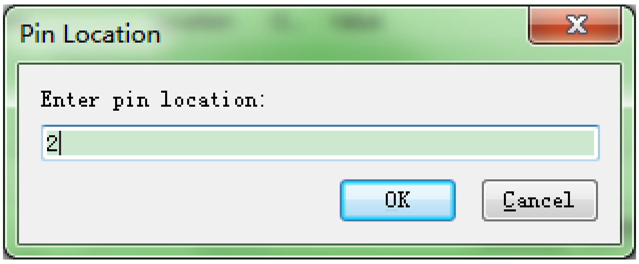
Utilities

Lookup BSC No. by Pin Location

Select menu **Utilities –Lookup cell by pin location…**:


Setting for Device 0
File
Tools
Option
Lookup Cell by Pin Location...
Lookup Cell by Port Name...
Add Cells Automatically with Regular Expression...
Edit I/O Settings in Table...
Opera
Outpu
BSC N
Delete
Cell
Input Setting
Name
BSC Numbers
Add
Update
Delete
Name
Cell Elements
OK
Cancel

Input pin location. See screenshot below:

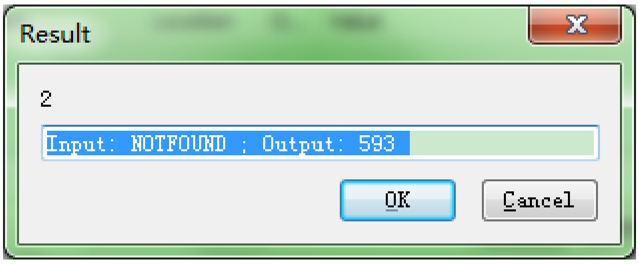


Enter pin location:

2

Click **OK** button.

The software will lookup BSDL file for the cell, and display search result.



2

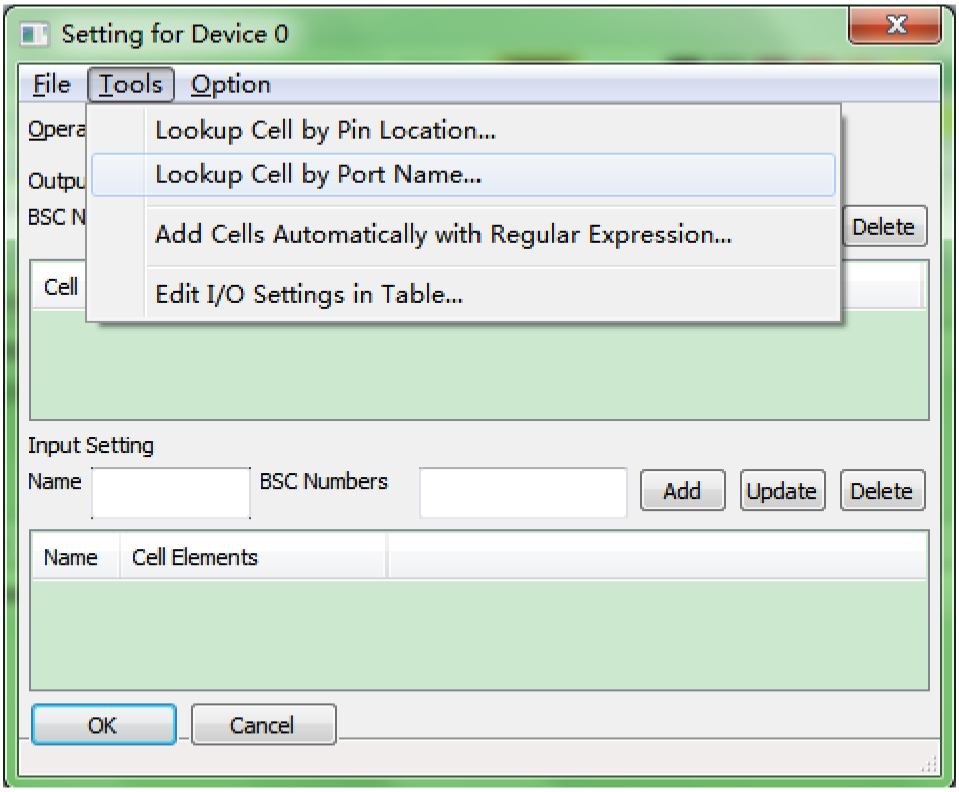
Input: NOTFOUND ; Output: 593

Now, you could copy corresponding cell number to **BSC No.** field for input or output setting.

Lookup BSC No. by Port Name

Similarly, you could lookup a cell by port name.

Select menu **Utilities –Lookup cell by port name…** .


Setting for Device 0
File
Tools
Option
Opera
Outpu
BSC N
Lookup Cell by Pin Location...
Lookup Cell by Port Name...
Add Cells Automatically with Regular Expression...
Edit I/O Settings in Table...
Delete
Cell
Input Setting
Name
BSC Numbers
Add
Update
Delete
Name
Cell Elements
OK
Cancel

Output Setting

Input one in edit box behind **BSC No.** then input value ('0' or '1') in edit box after **Outpu**t.

Note: In fact, the software could output any pattern. That's to say, you could input a binary string in the Output edit box. For example, if you set Output to 01, the software will output a 1:1 square waveform. If the Output pattern is 1000, the duty-cycle of output waveform will be 1:3.

Do above steps repeatedly until you have set all outputs.

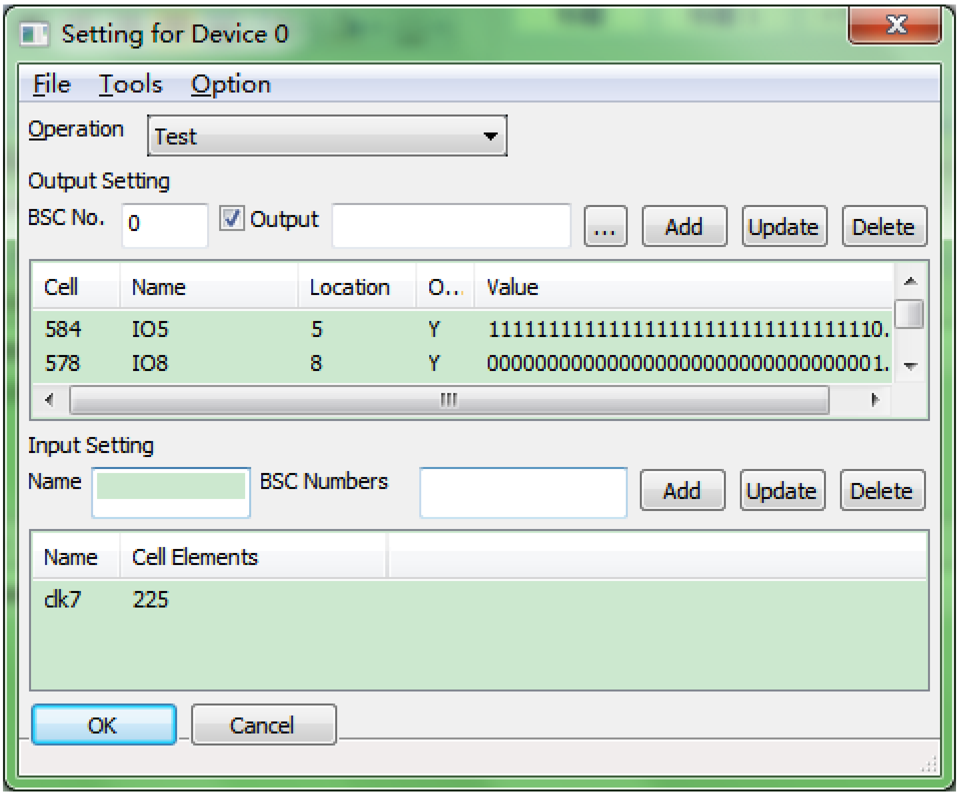
Input Setting

Input a name in edit box after 'Name', and input a cell number in edit box after **BSC number**, and click **Add** button.

Note: You can watch many input pins together and combine them in a group. Just input multi cell numbers and split them with ','.

Do above steps repeatedly until you have set all inputs.

Here is an example:



Operation: Test

Output Setting

BSC No. 0 ☑ Output

Cell	Name	Location	O...	Value
584	IO5	5	Y	11111111111111111111111111111110.
578	IO8	8	Y	00000000000000000000000000000001.

Input Setting

Name BSC Numbers

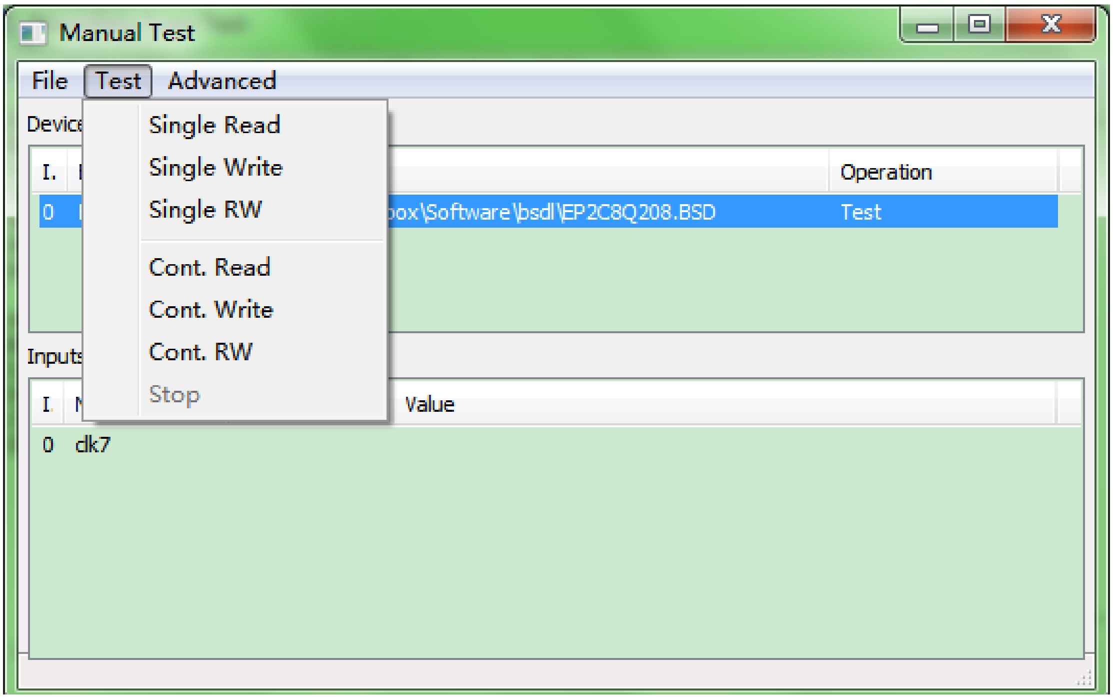
Name	Cell Elements
dk7	225

More Devices…

- Click **OK** button to return to **Manual Test** dialog.
- Do above steps repeatedly until you have set all devices.

Test

Test submenus:



Single Read: The software will read input value to pins one time of every device whose operation is in **Test**.

Single Write: The software will drive output value to pins one time of every device whose operation is in **Test**.

Single RW: **Single Write** and **Single Read**.

Cont. Read: The software will loop doing **Single Read** until menu **Test – Stop** is clicked.

Cont. Write: The software will loop doing **Single Write** until menu **Test – Stop** is clicked.

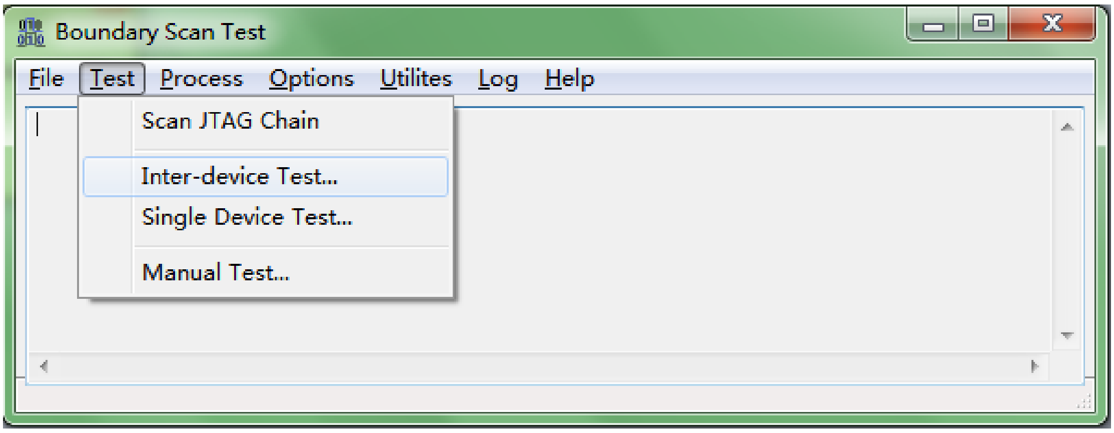
Cont. RW: The software will loop doing **Single RW** until menu **Test – Stop** is clicked.

Inter-Device Test

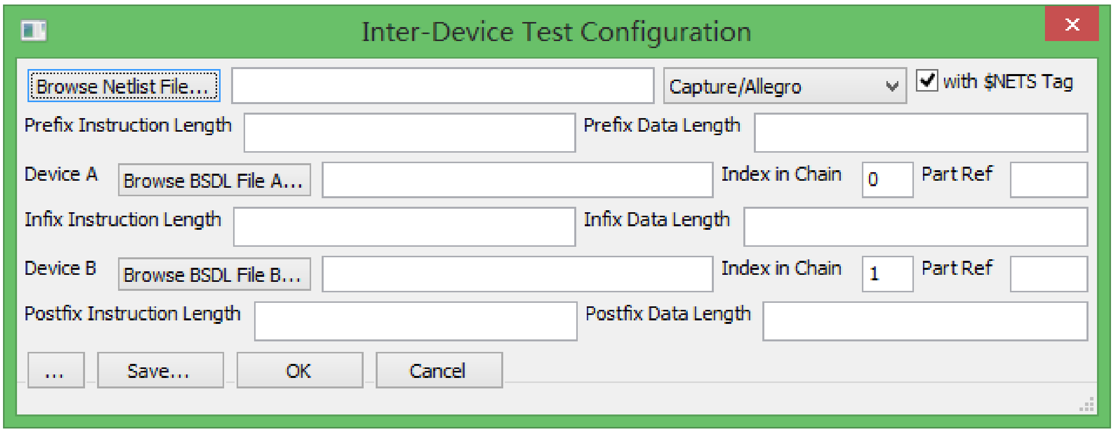
The system reads schematic netlists, and analyzes the device inter-connection, then generates test patterns. Inter-device test can automatically find short (pin to pin, pin to power supply or ground), open and other issues.

Steps

Select menu **Test** – I**nter-device Test**. See screenshot below:



You will see **Inter-device Test Configuration** dialog:



Capture/Allegro ☑ with $NETS Tag

Prefix Instruction Length
Prefix Data Length

Device A Index in Chain 0 Part Ref

Infix Instruction Length
Infix Data Length

Device B Index in Chain 1 Part Ref

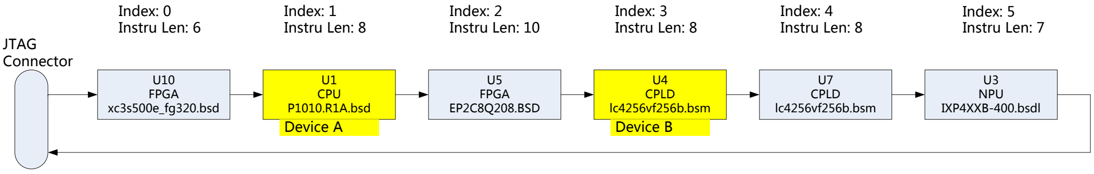
Postfix Instruction Length
Postfix Data Length

About the Parameters

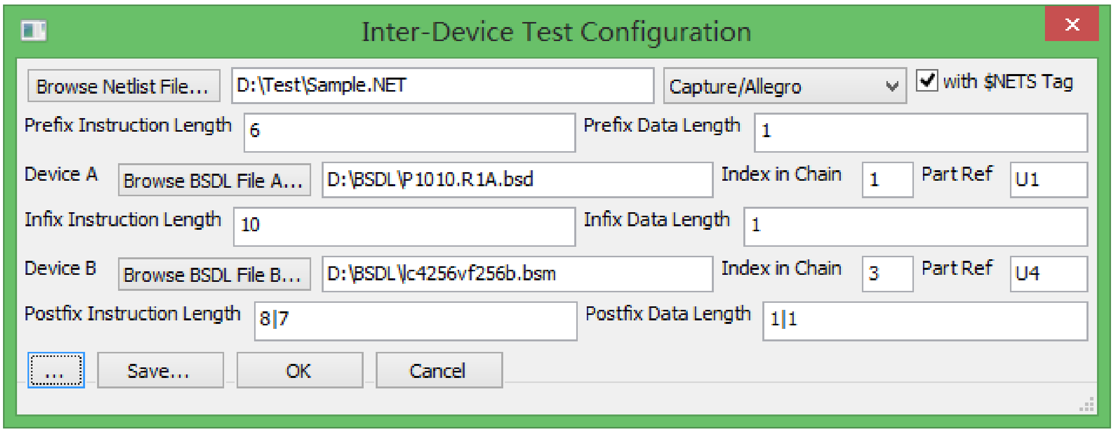
There are two DUTs. Let’s call them DUT A / **Device A** and DUT B / **Device B**.
Set schematic netlist file, and choose the netlist file format.
Set **BSDL file**, **Part Ref** and **Index in JTAG Chain** for DUT A and B.
Prefix Instruction Length: Please input JTAG instruction length of all other devices ahead of DUT A. If there are more than one devices, please split them with “|” character. For example, ‘4|5’ means there are two devices ahead of DUT in JTAG chain, and device at index 0 has a 4-bit JTAG instruction while device at index 1 has a 5-bit JTAG instruction. And you could find that the DUT is at index 2. Must left blank when no prefix device.
Infix Instruction Length: Please input JTAG instruction length of all other devices between DUT A and DUT B. Refer to Prefix Instruction Length. Must left blank when no infix device.
Postfix Instruction Length: Please input JTAG instruction length of all other devices behind of DUT B. Refer to Prefix Instruction Length. Must left blank when no postfix device.
Test Control File: This file is used to control other pins which are not inter-connected. Text file. Write one setting per line. Syntax: Part_Ref<TAB>Pin_Location<TAB>Output_Value.

An Example

Let's suppose a board with 6 devices in a single JTAG chain, and we are going to do inter-device test of U1 and U4.



The setting should be:



D:\Test\Sample.NET

Capture/Allegro

☑ with $NETS Tag

Prefix Instruction Length 6

Prefix Data Length 1

Device A D:\BSDL\P1010.R1A.bsd Index in Chain 1 Part Ref U1

Infix Instruction Length 10

Infix Data Length 1

Device B D:\BSDL\lc4256vf256b.bsm Index in Chain 3 Part Ref U4

Postfix Instruction Length 8|7

Postfix Data Length 1|1

Notes:

- □ All indexes are zero based.
- □ Except U1 and U4, other device are Bypass. The data length of the Bypass device is 1, so the data length is not required.

Tips

For you convenience, you could click **Save…** button to save the parameters to a text file with extension .ini. Next time just click the ... button to load saved parameters.

Run the Test

Click **OK** after all parameters are set.

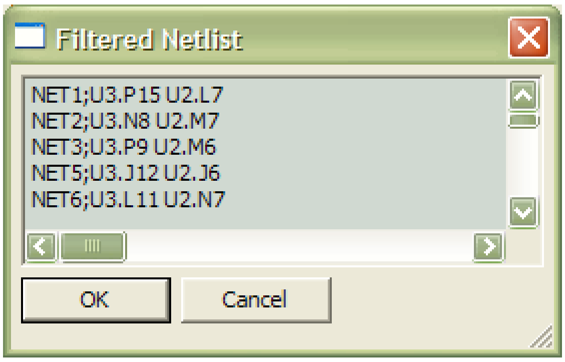
The software will do a JTAG chain scan and show result in log.

Then the software will parse BSDL files of devices in chain.

Then the software will check IDCODE.

After IDCODE checking passed, the software will parse netlist file.

The software will filter connection from all netlist according to Part Ref. You will see **Filtered Netlist** dialog.



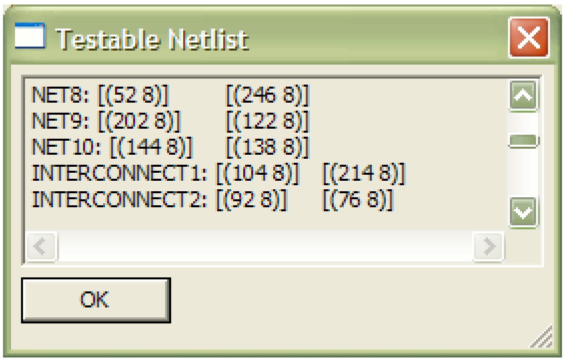
```
NET1;U3.P15 U2.L7
NET2;U3.N8 U2.M7
NET3;U3.P9 U2.M6
NET5;U3.J12 U2.J6
NET6;U3.L11 U2.N7
```

It’s a simple editor. You can edit the connections if you want.
That’s because sometimes the pins are connected through a resistor or logic clue, but they will not be filtered automatically since they have same netlist name. But they are testable. You can add them manually.

Click **OK** to continue.

The software will analyze to find testable netlists.
You know, only netlist connected between DUT A and DUT B could be tested, and those connections that one device could output and the other device could input could be testable.

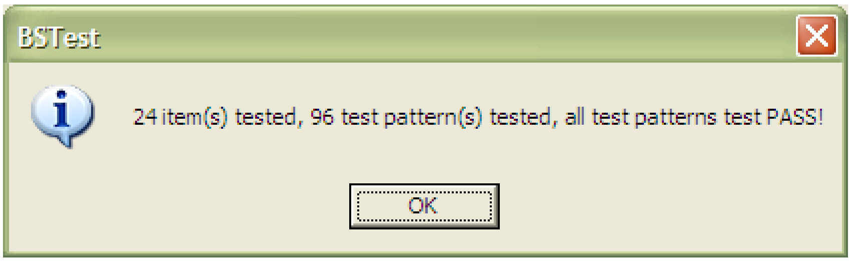
Testable Netlist dialog will show testable netlists.



```
NET8: [(52 8)]       [(246 8)]
NET9: [(202 8)]      [(122 8)]
NET10: [(144 8)]     [(138 8)]
INTERCONNECT1: [(104 8)]   [(214 8)]
INTERCONNECT2: [(92 8)]    [(76 8)]
```

Click **OK** button.

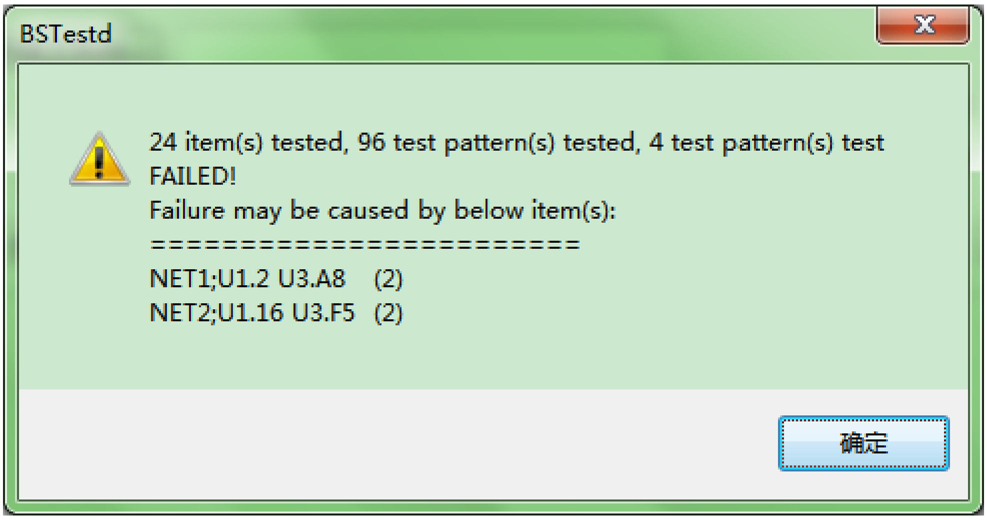
The software will run test and show result.



24 item(s) tested, 96 test pattern(s) tested, all test patterns test PASS!

You could check detailed info in log.

The software will tell you failure items detected since V2.2.2.1.
The software will show pin location at the same time since V2.3.0.0.
An example screenshot:



24 item(s) tested, 96 test pattern(s) tested, 4 test pattern(s) test FAILED!
Failure may be caused by below item(s):

```
========================
NET1;U1.2 U3.A8    (2)
NET2;U1.16 U3.F5   (2)
```

UI Difference Under Different Launching Mode

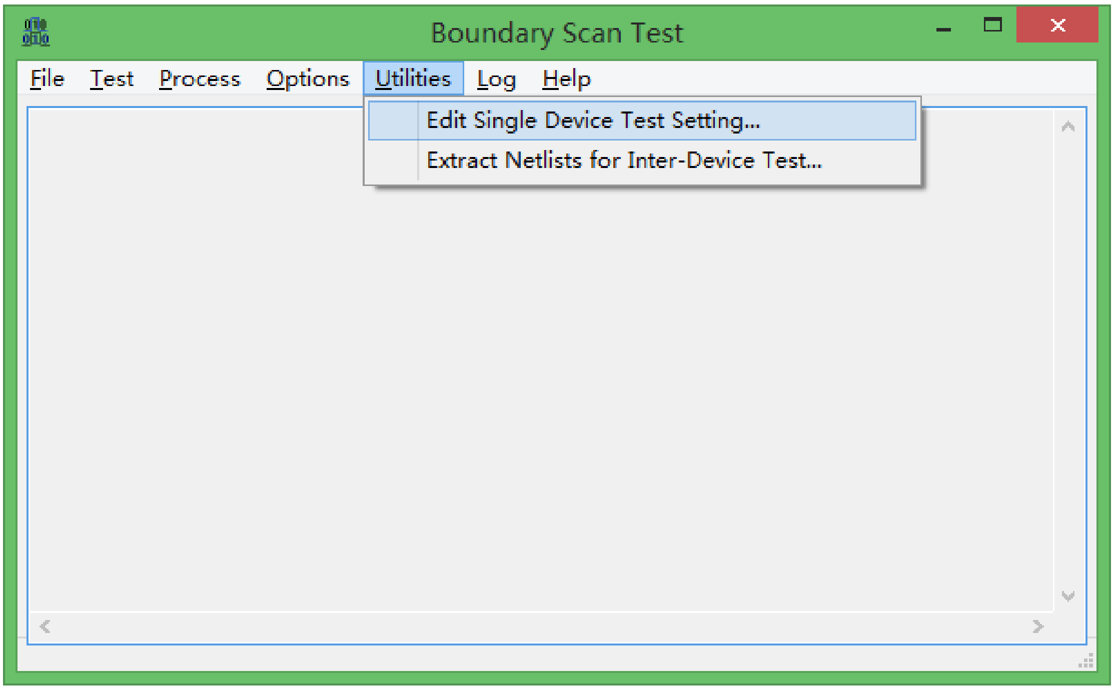
Launching UI Display	Menu: Inter-Device Test	Menu: Batch Test	Automatically launched when startup
Display ‘Edit Netlists’ Dialog	Yes	No	No
Display ‘Testable Netlists’ Confirmation Dialog	Yes	No	No
Show Error Message	Yes	Yes	Yes
Show Success Message	Yes	Yes	No

Single Device Test

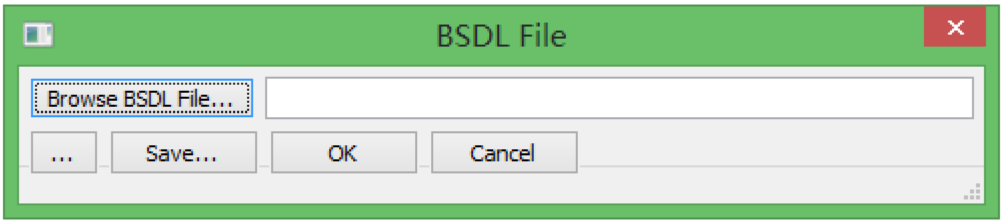
This automatic operation can detect short between pins (in fact, not only pins of JTAG device, but also pins of other devices connected to JTAG device), or short between I/O pins and power supply or ground.

Preparation

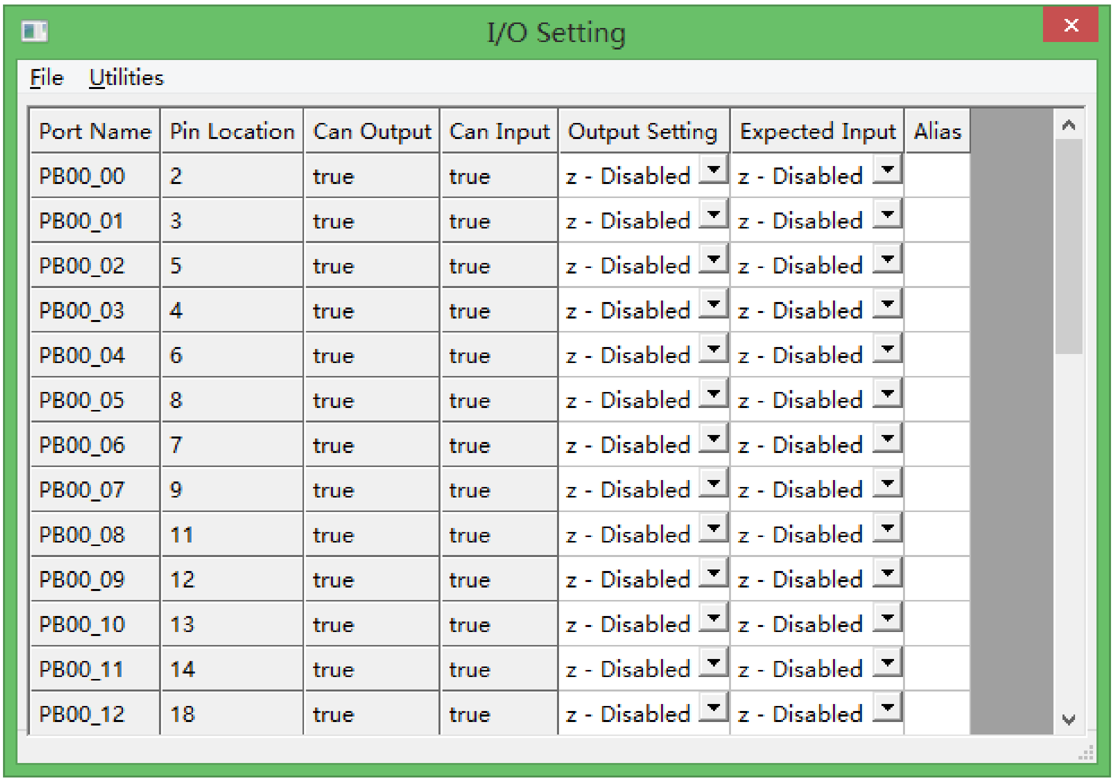
Before testing, please create a setting file by selecting menu **Utilities –Edit Single Device Test Setting…**:


Boundary Scan Test
File
Test
Process
Options
Utilities
Log
Help
Edit Single Device Test Setting...
Extract Netlists for Inter-Device Test...

A ‘**BSDL File**’ dialog will popup. Select correct BSDL and click **OK**:

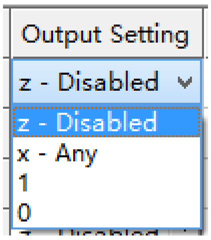


Now the ‘**I/O Setting**’ dialog is shown:



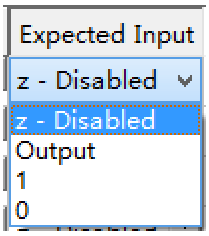
Port Name	Pin Location	Can Output	Can Input	Output Setting	Expected Input	Alias
PB00_00	2	true	true	z - Disabled	z - Disabled	
PB00_01	3	true	true	z - Disabled	z - Disabled	
PB00_02	5	true	true	z - Disabled	z - Disabled	
PB00_03	4	true	true	z - Disabled	z - Disabled	
PB00_04	6	true	true	z - Disabled	z - Disabled	
PB00_05	8	true	true	z - Disabled	z - Disabled	
PB00_06	7	true	true	z - Disabled	z - Disabled	
PB00_07	9	true	true	z - Disabled	z - Disabled	
PB00_08	11	true	true	z - Disabled	z - Disabled	
PB00_09	12	true	true	z - Disabled	z - Disabled	
PB00_10	13	true	true	z - Disabled	z - Disabled	
PB00_11	14	true	true	z - Disabled	z - Disabled	
PB00_12	18	true	true	z - Disabled	z - Disabled	

Output Setting



Option	Output Status	Comments
z – N/A	This cell could not output.	
z – Disabled	This cell is outputable, but we don’t test it.	
x – Any	Output anything.	*Pin under test should select this option.*
1	Output ‘1’ always.	The pin will stay on a certain status. It could be used to control some circuit.
0	Output ‘0’ always.	The pin will stay on a certain status. It could be used to control some circuit.

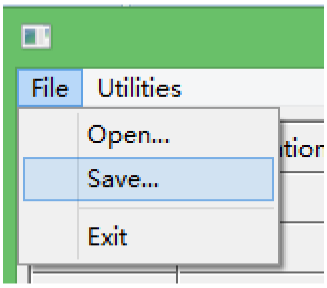
Expected Input



Option	Input Status	Comments
z – N/A	This cell could not input.	
Z – Disabled	This cell is inputable, but we don’t check it.	
Output	Changes with the Output.	Input should be the same as output. *Pin under test should select this option.* Attention: The output should not be z – N/A nor z – Disabled.
1	Input is ‘1’.	This pin should always input ‘1’ normally, and other status when failure. It happens to the pin whose status is fixed.
0	Input is ‘0’.	This pin should always input ‘0’ normally, and other status when failure. It happens to the pin whose status is fixed.

Save

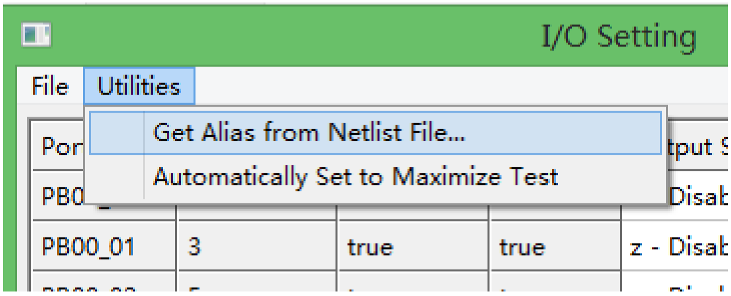
Menu **File** – **Save…**:



Utilities

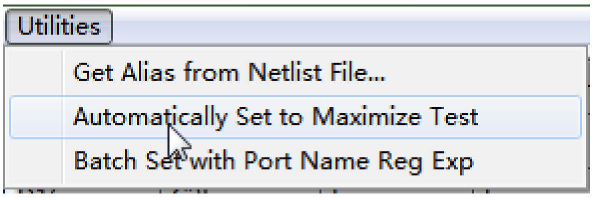
Alias

To improve the readability of test result, you could add alias to the pins. Select menu **Utilities – Get Alias from Netlist File…**:



Auto Set to Max Testable

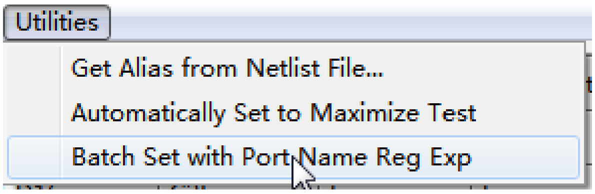
Select menu **Utilities – Get Alias from Netlist File…**:


Utilities
Get Alias from Netlist File...
Automatically Set to Maximize Test
Batch Set with Port Name Reg Exp

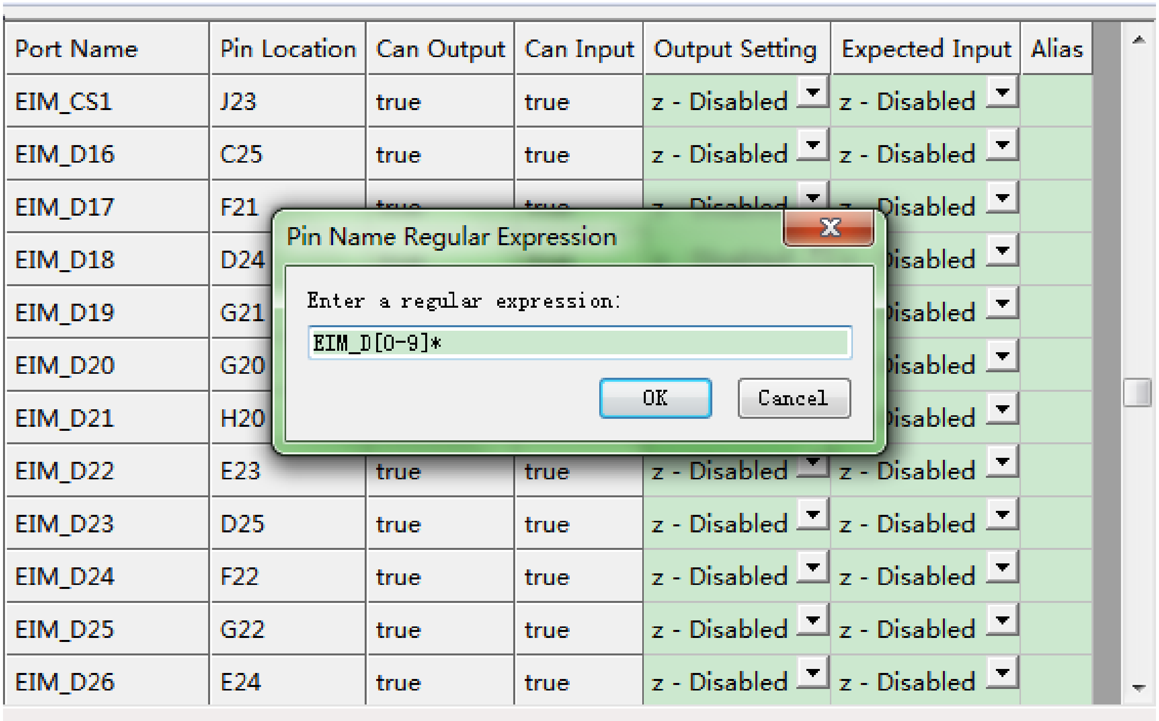
Attention: This utility automatically generate an general and maximized-possible-testable setting, but it should be revised based on your board design since the pin connections vary on each board.

Batch Set with Port Name Regular Expression

Select menu **Utilities –Batch Set with Port Name Regular Expression…**:


Utilities
Get Alias from Netlist File...
Automatically Set to Maximize Test
Batch Set with Port Name Reg Exp

Then input a regular expression:



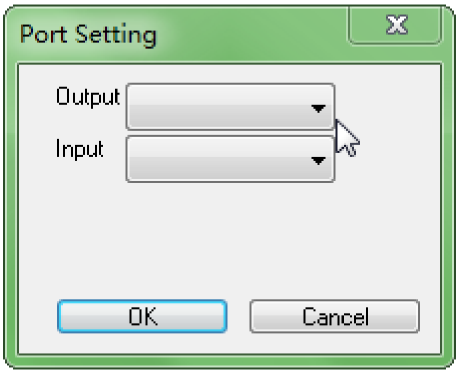
Port Name	Pin Location	Can Output	Can Input	Output Setting	Expected Input	Alias
EIM_CS1	J23	true	true	z - Disabled	z - Disabled	
EIM_D16	C25	true	true	z - Disabled	z - Disabled	
EIM_D17	F21	true	true	z - Disabled	Disabled	
EIM_D18	D24				isabled	
EIM_D19	G21				isabled	
EIM_D20	G20				isabled	
EIM_D21	H20				isabled	
EIM_D22	E23	true	true	z - Disabled	z - Disabled	
EIM_D23	D25	true	true	z - Disabled	z - Disabled	
EIM_D24	F22	true	true	z - Disabled	z - Disabled	
EIM_D25	G22	true	true	z - Disabled	z - Disabled	
EIM_D26	E24	true	true	z - Disabled	z - Disabled	

Pin Name Regular Expression

Enter a regular expression:

EIM_D[0-9]*

The setting dialog appears after ‘OK’ button is clicked:

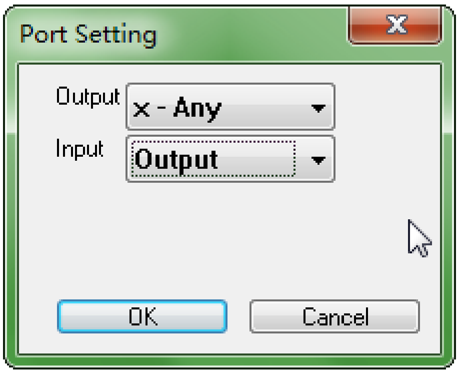


Port Setting

Output

Input

Please select output and input settings for these ports/pins. In this case, we hope output both 1 and 0, and input should be same as input. (Please refer to Output Setting and Expected Input for more info.) So:

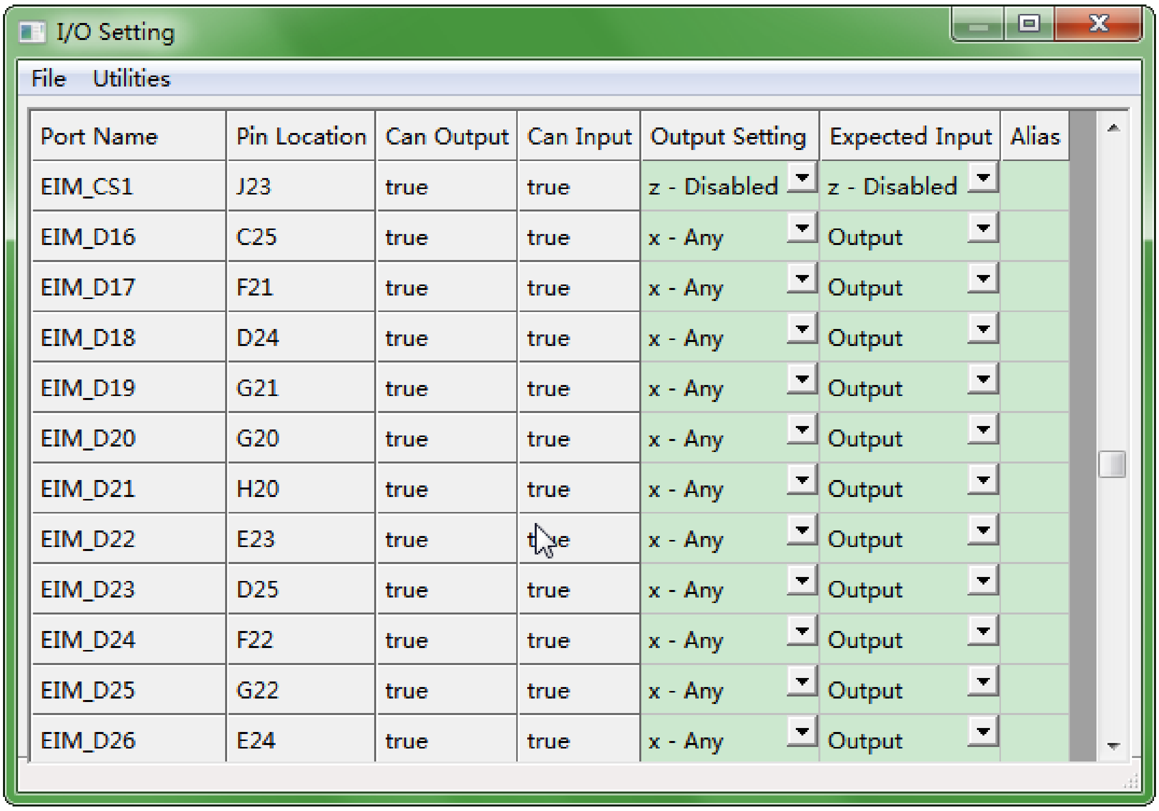


Port Setting

Output x - Any

Input Output

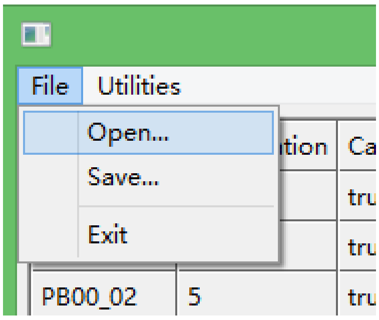
Click OK, ports/pins met the regular expression will all be set:



Port Name	Pin Location	Can Output	Can Input	Output Setting	Expected Input	Alias
EIM_CS1	J23	true	true	z - Disabled	z - Disabled	
EIM_D16	C25	true	true	x - Any	Output	
EIM_D17	F21	true	true	x - Any	Output	
EIM_D18	D24	true	true	x - Any	Output	
EIM_D19	G21	true	true	x - Any	Output	
EIM_D20	G20	true	true	x - Any	Output	
EIM_D21	H20	true	true	x - Any	Output	
EIM_D22	E23	true	true	x - Any	Output	
EIM_D23	D25	true	true	x - Any	Output	
EIM_D24	F22	true	true	x - Any	Output	
EIM_D25	G22	true	true	x - Any	Output	
EIM_D26	E24	true	true	x - Any	Output	

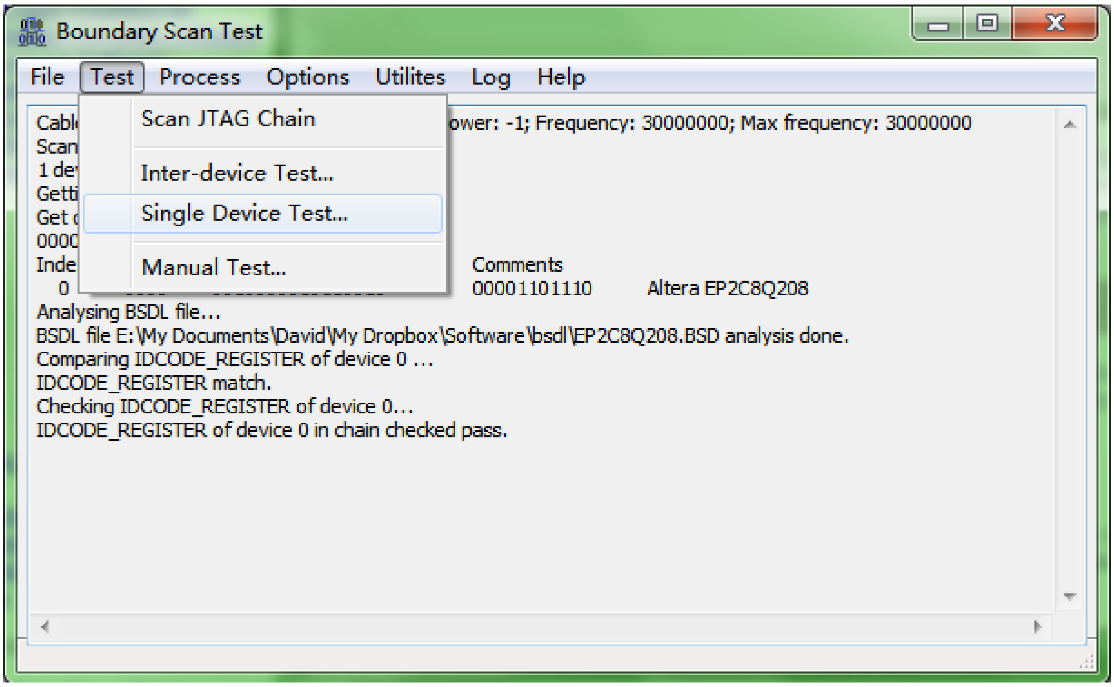
Edit Saved Setting File

Menu **File – Open…**



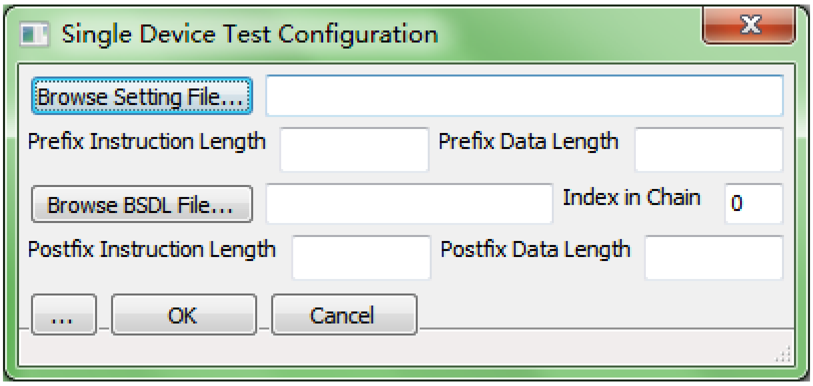
Steps

Select menu **Test** – **Single Device Test**:



```
Cabl                                   ower: -1; Frequency: 30000000; Max frequency: 30000000
Scan
1 de
Getti
Get c
0000
Inde                                   Comments
  0                                    00001101110        Altera EP2C8Q208
Analysing BSDL file...
BSDL file E:\My Documents\David\My Dropbox\Software\bsdl\EP2C8Q208.BSD analysis done.
Comparing IDCODE_REGISTER of device 0 ...
IDCODE_REGISTER match.
Checking IDCODE_REGISTER of device 0...
IDCODE_REGISTER of device 0 in chain checked pass.
```

You will see **Single Device Test Configuration** dialog:



Browse Setting File: Select the saved setting file in above section.

Prefix Instruction Length: Please input JTAG instruction length of all other devices ahead of DUT. If there are more than one devices, please split them with "|" character. For example, '4|5' means there are two devices ahead of DUT in JTAG chain, and device at index 0 has a 4-bit JTAG instruction while device at index 1 has a 5-bit JTAG instruction. And you could find that the DUT is at index 2. Must left blank when no prefix device.

Post Instruction Length: Please input JTAG instruction length of all other devices behind of DUT. Refer to Prefix Instruction Length. Must left blank when no postfix device.

When **Setting File**, **BSDL file of DUT** and **Index of DUT** are set, click **OK** button.

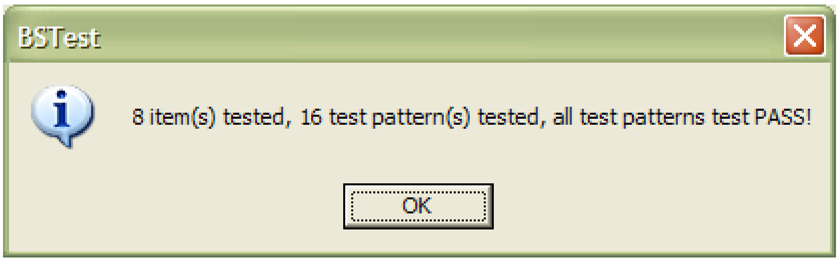
The software will run rest according to **Setting File** and show testing result.

Example 1

The **Setting File** contains 8 items to be tested. They are `D0` to `D7`.

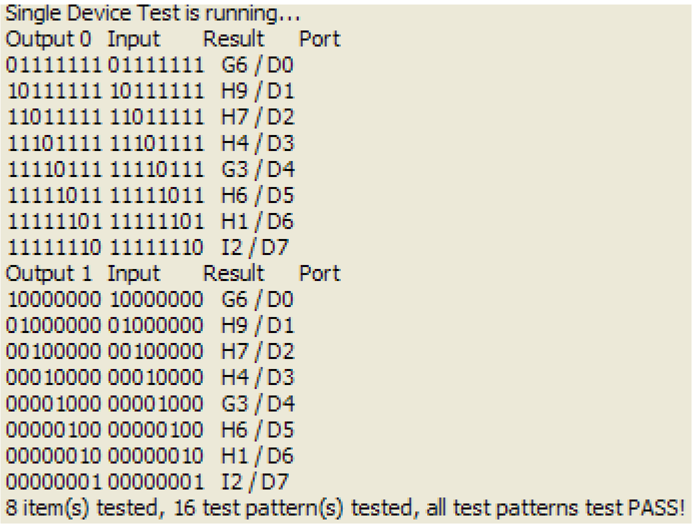
Test on Good Board

No error found.



8 item(s) tested, 16 test pattern(s) tested, all test patterns test PASS!

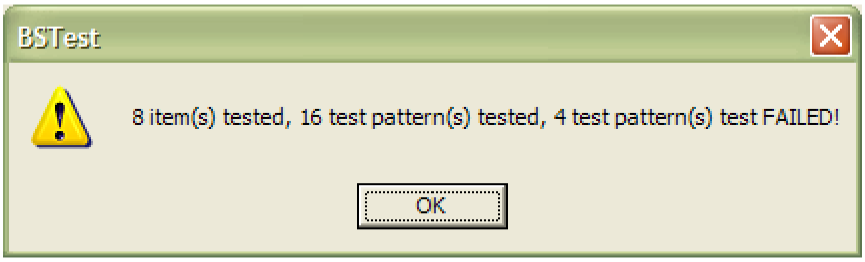
The log will show detailed info of test.



```
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!
```

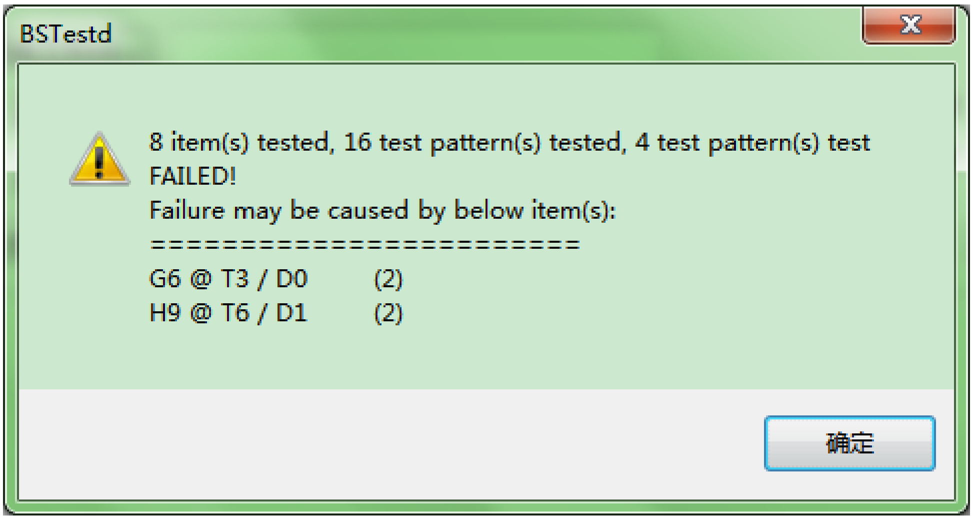
Short-Circuit Issue

We manually connect pin `D0` and `D1` with a wire. This time we'll be told failure found in test.



8 item(s) tested, 16 test pattern(s) tested, 4 test pattern(s) test FAILED!

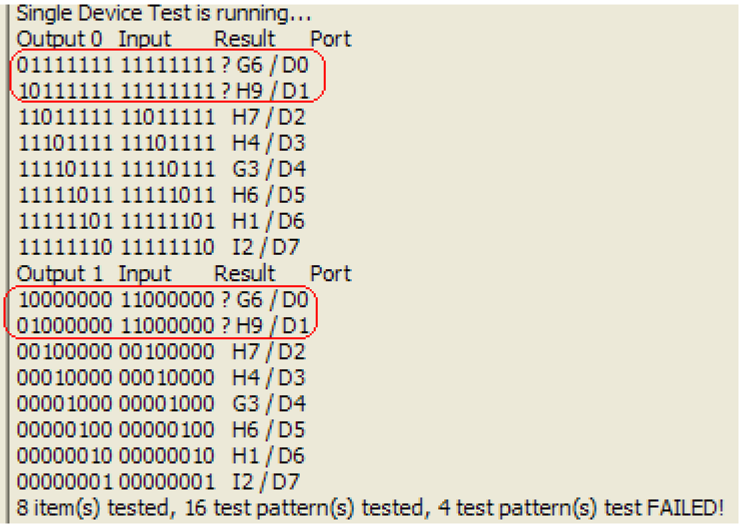
The software will tell you failure items detected since V2.2.2.1. Furthermore, the software will show pin location of failed item since V2.3.0.0. Display format: **Port Name @ Pin Location / Alias**. Please note pin location or alias will not be displayed is it’s empty. See screenshot below:



8 item(s) tested, 16 test pattern(s) tested, 4 test pattern(s) test FAILED!
Failure may be caused by below item(s):

```
========================
G6 @ T3 / D0       (2)
H9 @ T6 / D1       (2)
```

Log showed:



```
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!
```

It’s easy to find `D0` and `D1` didn’t pass the test since we shorted them.

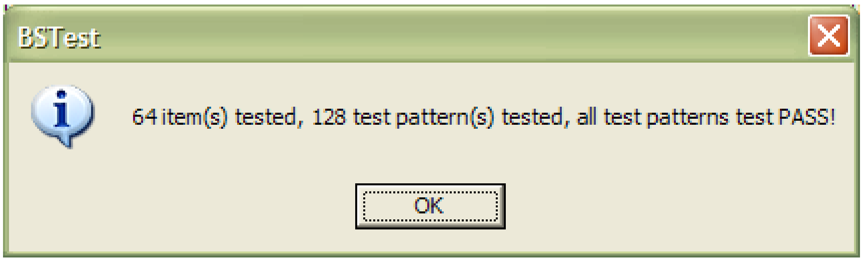
More…

In fact, not only short circuit between pins of JTAG device, but also short circuit between pins of other devices connected to JTAG device could be found.

Example 2

The **Setting File** contains 64 items to be tested. They are `d(0)` to `d(63)`.

Test on Good Board

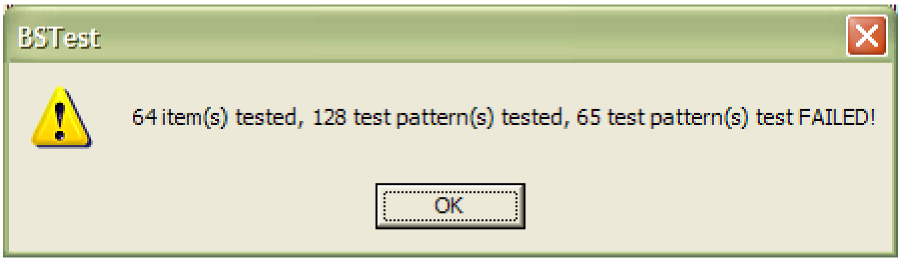


64 item(s) tested, 128 test pattern(s) tested, all test patterns test PASS!

No error found.

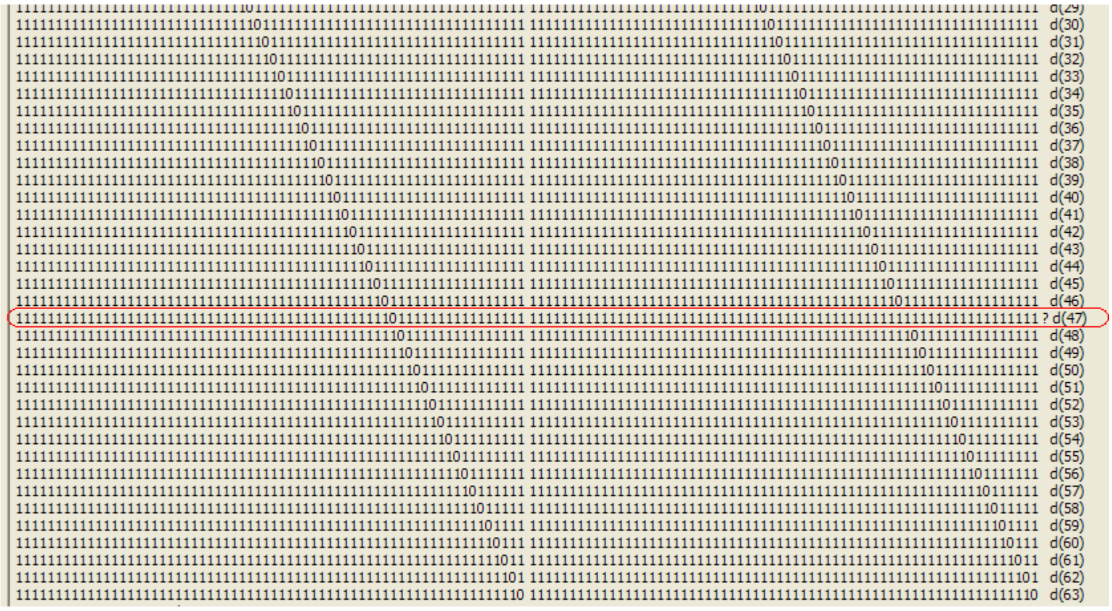
Short Circuit with Power

We manually connect pin `d(47)` and `3.3V` with a wire.



64 item(s) tested, 128 test pattern(s) tested, 65 test pattern(s) test FAILED!

Detailed log:

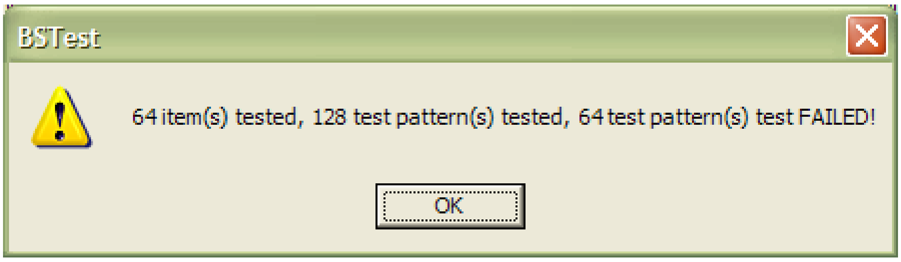


```
1111111111111111111111111111101111111111111111111111111111111111 1111111111111111111111111111101111111111111111111111111111111111  d(29)
1111111111111111111111111111110111111111111111111111111111111111 1111111111111111111111111111110111111111111111111111111111111111  d(30)
1111111111111111111111111111111011111111111111111111111111111111 1111111111111111111111111111111011111111111111111111111111111111  d(31)
1111111111111111111111111111111101111111111111111111111111111111 1111111111111111111111111111111101111111111111111111111111111111  d(32)
1111111111111111111111111111111110111111111111111111111111111111 1111111111111111111111111111111110111111111111111111111111111111  d(33)
1111111111111111111111111111111111011111111111111111111111111111 1111111111111111111111111111111111011111111111111111111111111111  d(34)
1111111111111111111111111111111111101111111111111111111111111111 1111111111111111111111111111111111101111111111111111111111111111  d(35)
1111111111111111111111111111111111110111111111111111111111111111 1111111111111111111111111111111111110111111111111111111111111111  d(36)
1111111111111111111111111111111111111011111111111111111111111111 1111111111111111111111111111111111111011111111111111111111111111  d(37)
1111111111111111111111111111111111111101111111111111111111111111 1111111111111111111111111111111111111101111111111111111111111111  d(38)
1111111111111111111111111111111111111110111111111111111111111111 1111111111111111111111111111111111111110111111111111111111111111  d(39)
1111111111111111111111111111111111111111011111111111111111111111 1111111111111111111111111111111111111111011111111111111111111111  d(40)
1111111111111111111111111111111111111111101111111111111111111111 1111111111111111111111111111111111111111101111111111111111111111  d(41)
1111111111111111111111111111111111111111110111111111111111111111 1111111111111111111111111111111111111111110111111111111111111111  d(42)
1111111111111111111111111111111111111111111011111111111111111111 1111111111111111111111111111111111111111111011111111111111111111  d(43)
1111111111111111111111111111111111111111111101111111111111111111 1111111111111111111111111111111111111111111101111111111111111111  d(44)
1111111111111111111111111111111111111111111110111111111111111111 1111111111111111111111111111111111111111111110111111111111111111  d(45)
1111111111111111111111111111111111111111111111011111111111111111 1111111111111111111111111111111111111111111111011111111111111111  d(46)
1111111111111111111111111111111111111111111111101111111111111111 1111111111111111111111111111111111111111111111111111111111111111 ? d(47)
1111111111111111111111111111111111111111111111110111111111111111 1111111111111111111111111111111111111111111111110111111111111111  d(48)
1111111111111111111111111111111111111111111111111011111111111111 1111111111111111111111111111111111111111111111111011111111111111  d(49)
1111111111111111111111111111111111111111111111111101111111111111 1111111111111111111111111111111111111111111111111101111111111111  d(50)
1111111111111111111111111111111111111111111111111110111111111111 1111111111111111111111111111111111111111111111111110111111111111  d(51)
1111111111111111111111111111111111111111111111111111011111111111 1111111111111111111111111111111111111111111111111111011111111111  d(52)
1111111111111111111111111111111111111111111111111111101111111111 1111111111111111111111111111111111111111111111111111101111111111  d(53)
1111111111111111111111111111111111111111111111111111110111111111 1111111111111111111111111111111111111111111111111111110111111111  d(54)
1111111111111111111111111111111111111111111111111111111011111111 1111111111111111111111111111111111111111111111111111111011111111  d(55)
1111111111111111111111111111111111111111111111111111111101111111 1111111111111111111111111111111111111111111111111111111101111111  d(56)
1111111111111111111111111111111111111111111111111111111110111111 1111111111111111111111111111111111111111111111111111111110111111  d(57)
1111111111111111111111111111111111111111111111111111111111011111 1111111111111111111111111111111111111111111111111111111111011111  d(58)
1111111111111111111111111111111111111111111111111111111111101111 1111111111111111111111111111111111111111111111111111111111101111  d(59)
1111111111111111111111111111111111111111111111111111111111110111 1111111111111111111111111111111111111111111111111111111111110111  d(60)
1111111111111111111111111111111111111111111111111111111111111011 1111111111111111111111111111111111111111111111111111111111111011  d(61)
1111111111111111111111111111111111111111111111111111111111111101 1111111111111111111111111111111111111111111111111111111111111101  d(62)
1111111111111111111111111111111111111111111111111111111111111110 1111111111111111111111111111111111111111111111111111111111111110  d(63)
```

We could see easily that pin `d(47)` doesn’t pass the test.

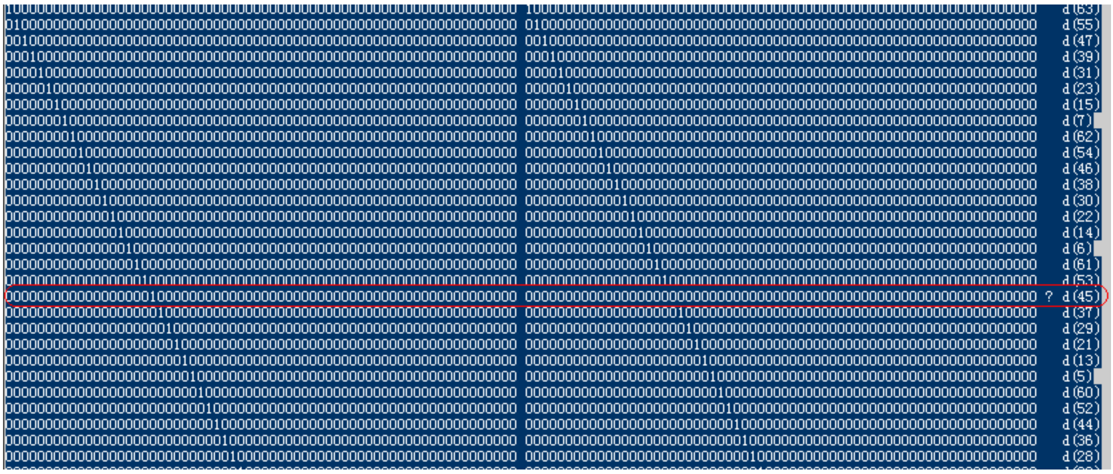
Short Circuit with GND

We manually connect pin `d(45)` and `GND` with a wire



64 item(s) tested, 128 test pattern(s) tested, 64 test pattern(s) test FAILED!

Detailed log:

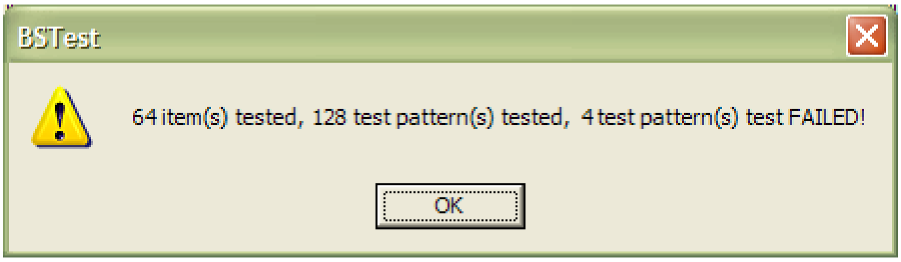


```
1000000000000000000000000000000000000000000000000000000000000000 1000000000000000000000000000000000000000000000000000000000000000   d(63)
0100000000000000000000000000000000000000000000000000000000000000 0100000000000000000000000000000000000000000000000000000000000000   d(55)
0010000000000000000000000000000000000000000000000000000000000000 0010000000000000000000000000000000000000000000000000000000000000   d(47)
0001000000000000000000000000000000000000000000000000000000000000 0001000000000000000000000000000000000000000000000000000000000000   d(39)
0000100000000000000000000000000000000000000000000000000000000000 0000100000000000000000000000000000000000000000000000000000000000   d(31)
0000010000000000000000000000000000000000000000000000000000000000 0000010000000000000000000000000000000000000000000000000000000000   d(23)
0000001000000000000000000000000000000000000000000000000000000000 0000001000000000000000000000000000000000000000000000000000000000   d(15)
0000000100000000000000000000000000000000000000000000000000000000 0000000100000000000000000000000000000000000000000000000000000000   d(7)
0000000010000000000000000000000000000000000000000000000000000000 0000000010000000000000000000000000000000000000000000000000000000   d(62)
0000000001000000000000000000000000000000000000000000000000000000 0000000001000000000000000000000000000000000000000000000000000000   d(54)
0000000000100000000000000000000000000000000000000000000000000000 0000000000100000000000000000000000000000000000000000000000000000   d(46)
0000000000010000000000000000000000000000000000000000000000000000 0000000000010000000000000000000000000000000000000000000000000000   d(38)
0000000000001000000000000000000000000000000000000000000000000000 0000000000001000000000000000000000000000000000000000000000000000   d(30)
0000000000000100000000000000000000000000000000000000000000000000 0000000000000100000000000000000000000000000000000000000000000000   d(22)
0000000000000010000000000000000000000000000000000000000000000000 0000000000000010000000000000000000000000000000000000000000000000   d(14)
0000000000000001000000000000000000000000000000000000000000000000 0000000000000001000000000000000000000000000000000000000000000000   d(6)
0000000000000000100000000000000000000000000000000000000000000000 0000000000000000100000000000000000000000000000000000000000000000   d(61)
0000000000000000010000000000000000000000000000000000000000000000 0000000000000000010000000000000000000000000000000000000000000000   d(53)
0000000000000000001000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000000000000000 ? d(45)
0000000000000000000100000000000000000000000000000000000000000000 0000000000000000000100000000000000000000000000000000000000000000   d(37)
0000000000000000000010000000000000000000000000000000000000000000 0000000000000000000010000000000000000000000000000000000000000000   d(29)
0000000000000000000001000000000000000000000000000000000000000000 0000000000000000000001000000000000000000000000000000000000000000   d(21)
0000000000000000000000100000000000000000000000000000000000000000 0000000000000000000000100000000000000000000000000000000000000000   d(13)
0000000000000000000000010000000000000000000000000000000000000000 0000000000000000000000010000000000000000000000000000000000000000   d(5)
0000000000000000000000001000000000000000000000000000000000000000 0000000000000000000000001000000000000000000000000000000000000000   d(60)
0000000000000000000000000100000000000000000000000000000000000000 0000000000000000000000000100000000000000000000000000000000000000   d(52)
0000000000000000000000000010000000000000000000000000000000000000 0000000000000000000000000010000000000000000000000000000000000000   d(44)
0000000000000000000000000001000000000000000000000000000000000000 0000000000000000000000000001000000000000000000000000000000000000   d(36)
0000000000000000000000000000100000000000000000000000000000000000 0000000000000000000000000000100000000000000000000000000000000000   d(28)
```

Obviously, `d(45)` could not pass the test.

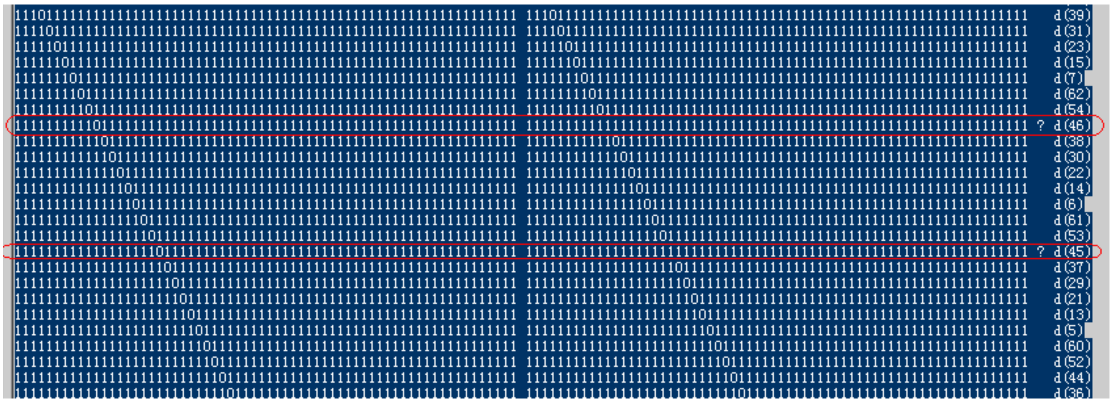
Short-Circuit Between Pins

We manually connect pin `d(45)` and `d(46)` with a wire.:



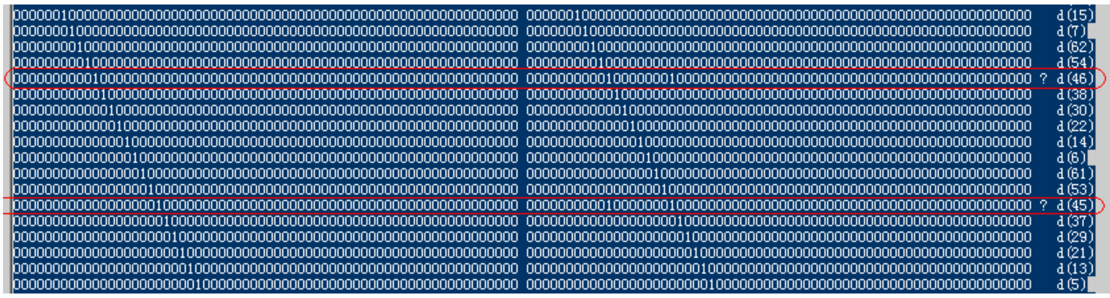
64 item(s) tested, 128 test pattern(s) tested, 4 test pattern(s) test FAILED!

Detailed log of test when outputting 0:



```
1110111111111111111111111111111111111111111111111111111111111111 1110111111111111111111111111111111111111111111111111111111111111   d(39)
1111011111111111111111111111111111111111111111111111111111111111 1111011111111111111111111111111111111111111111111111111111111111   d(31)
1111101111111111111111111111111111111111111111111111111111111111 1111101111111111111111111111111111111111111111111111111111111111   d(23)
1111110111111111111111111111111111111111111111111111111111111111 1111110111111111111111111111111111111111111111111111111111111111   d(15)
1111111011111111111111111111111111111111111111111111111111111111 1111111011111111111111111111111111111111111111111111111111111111   d(7)
1111111101111111111111111111111111111111111111111111111111111111 1111111101111111111111111111111111111111111111111111111111111111   d(62)
1111111110111111111111111111111111111111111111111111111111111111 1111111110111111111111111111111111111111111111111111111111111111   d(54)
1111111111011111111111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111111111111111111111 ? d(46)
1111111111101111111111111111111111111111111111111111111111111111 1111111111101111111111111111111111111111111111111111111111111111   d(38)
1111111111110111111111111111111111111111111111111111111111111111 1111111111110111111111111111111111111111111111111111111111111111   d(30)
1111111111111011111111111111111111111111111111111111111111111111 1111111111111011111111111111111111111111111111111111111111111111   d(22)
1111111111111101111111111111111111111111111111111111111111111111 1111111111111101111111111111111111111111111111111111111111111111   d(14)
1111111111111110111111111111111111111111111111111111111111111111 1111111111111110111111111111111111111111111111111111111111111111   d(6)
1111111111111111011111111111111111111111111111111111111111111111 1111111111111111011111111111111111111111111111111111111111111111   d(61)
1111111111111111101111111111111111111111111111111111111111111111 1111111111111111101111111111111111111111111111111111111111111111   d(53)
1111111111111111110111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111111111111111111111 ? d(45)
1111111111111111111011111111111111111111111111111111111111111111 1111111111111111111011111111111111111111111111111111111111111111   d(37)
1111111111111111111101111111111111111111111111111111111111111111 1111111111111111111101111111111111111111111111111111111111111111   d(29)
1111111111111111111110111111111111111111111111111111111111111111 1111111111111111111110111111111111111111111111111111111111111111   d(21)
1111111111111111111111011111111111111111111111111111111111111111 1111111111111111111111011111111111111111111111111111111111111111   d(13)
1111111111111111111111101111111111111111111111111111111111111111 1111111111111111111111101111111111111111111111111111111111111111   d(5)
1111111111111111111111110111111111111111111111111111111111111111 1111111111111111111111110111111111111111111111111111111111111111   d(60)
1111111111111111111111111011111111111111111111111111111111111111 1111111111111111111111111011111111111111111111111111111111111111   d(52)
1111111111111111111111111101111111111111111111111111111111111111 1111111111111111111111111101111111111111111111111111111111111111   d(44)
1111111111111111111111111110111111111111111111111111111111111111 1111111111111111111111111110111111111111111111111111111111111111   d(36)
```

Detailed log of test when outputting 1:



```
0000001000000000000000000000000000000000000000000000000000000000 0000001000000000000000000000000000000000000000000000000000000000   d(15)
0000000100000000000000000000000000000000000000000000000000000000 0000000100000000000000000000000000000000000000000000000000000000   d(7)
0000000010000000000000000000000000000000000000000000000000000000 0000000010000000000000000000000000000000000000000000000000000000   d(62)
0000000001000000000000000000000000000000000000000000000000000000 0000000001000000000000000000000000000000000000000000000000000000   d(54)
0000000000100000000000000000000000000000000000000000000000000000 0000000000100000001000000000000000000000000000000000000000000000 ? d(46)
0000000000010000000000000000000000000000000000000000000000000000 0000000000010000000000000000000000000000000000000000000000000000   d(38)
0000000000001000000000000000000000000000000000000000000000000000 0000000000001000000000000000000000000000000000000000000000000000   d(30)
0000000000000100000000000000000000000000000000000000000000000000 0000000000000100000000000000000000000000000000000000000000000000   d(22)
0000000000000010000000000000000000000000000000000000000000000000 0000000000000010000000000000000000000000000000000000000000000000   d(14)
0000000000000001000000000000000000000000000000000000000000000000 0000000000000001000000000000000000000000000000000000000000000000   d(6)
0000000000000000100000000000000000000000000000000000000000000000 0000000000000000100000000000000000000000000000000000000000000000   d(61)
0000000000000000010000000000000000000000000000000000000000000000 0000000000000000010000000000000000000000000000000000000000000000   d(53)
0000000000000000001000000000000000000000000000000000000000000000 0000000000100000001000000000000000000000000000000000000000000000 ? d(45)
0000000000000000000100000000000000000000000000000000000000000000 0000000000000000000100000000000000000000000000000000000000000000   d(37)
0000000000000000000010000000000000000000000000000000000000000000 0000000000000000000010000000000000000000000000000000000000000000   d(29)
0000000000000000000001000000000000000000000000000000000000000000 0000000000000000000001000000000000000000000000000000000000000000   d(21)
0000000000000000000000100000000000000000000000000000000000000000 0000000000000000000000100000000000000000000000000000000000000000   d(13)
0000000000000000000000010000000000000000000000000000000000000000 0000000000000000000000010000000000000000000000000000000000000000   d(5)
```

We could see that there are always two pins test failed no matter what value is outputting (0 or 1). The two pins are d(45) and d(46).

More…

Examples above just list part of pins. You can add as more as you want in the test setting file.

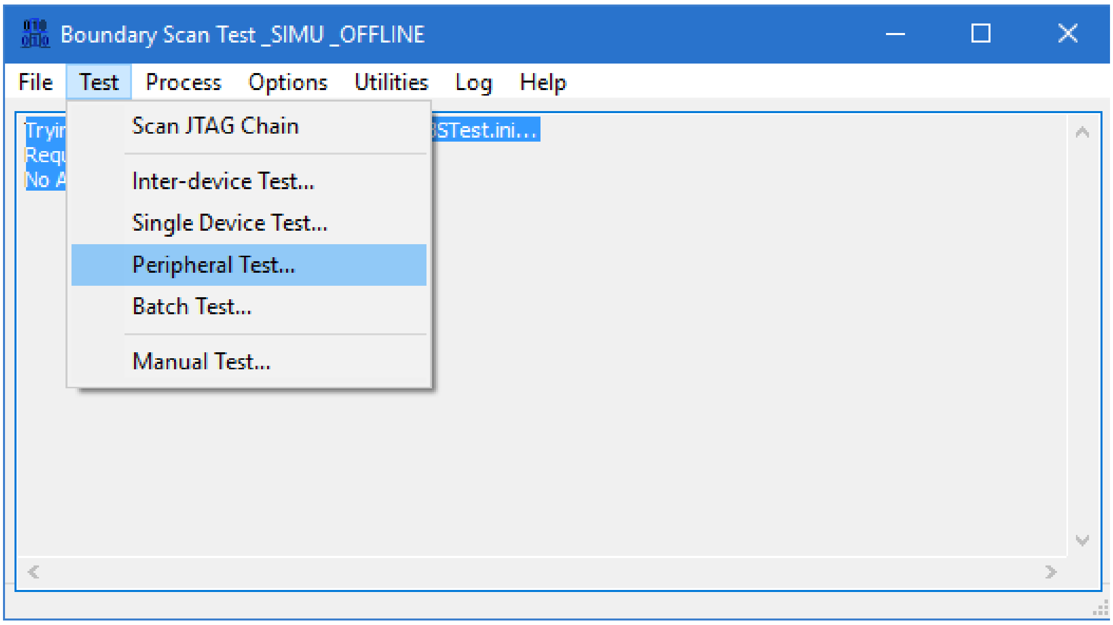
Peripheral Test

Edit JFP Configuration File

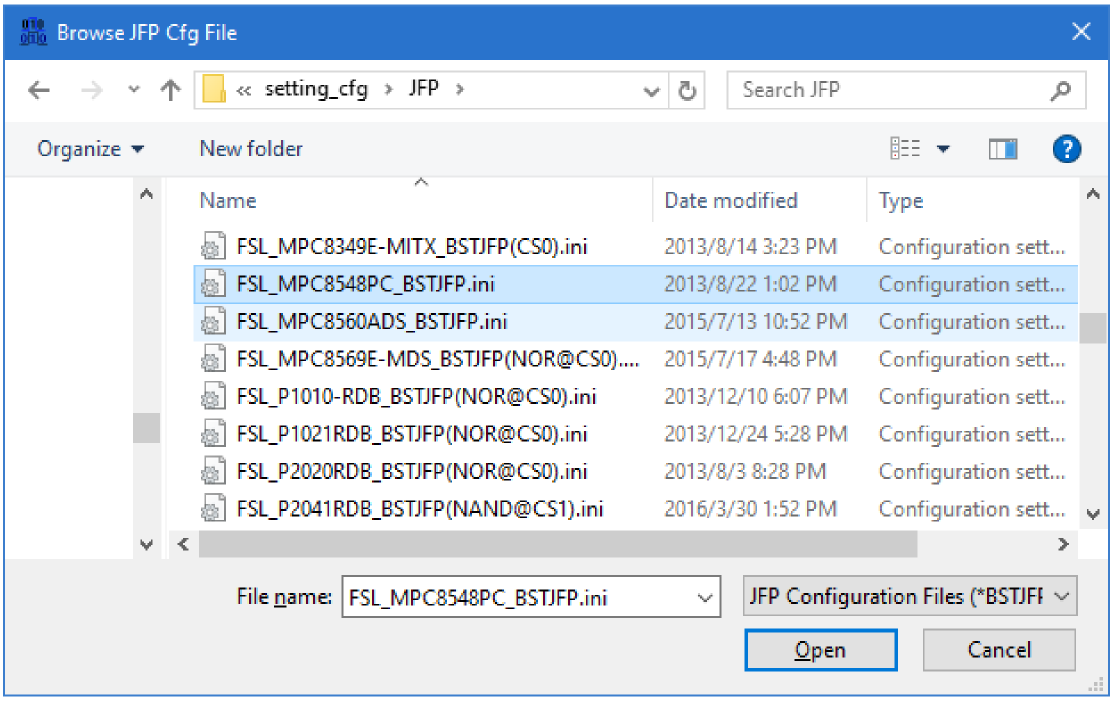
Follow “JFP Edit Cfg File User Manual(ENU).pdf” to get a JFP configuration file.

Run

Select menu **Test** – **Peripheral Test**…. See screenshot below:



Select configuration file. See screenshot below:



Name	Date modified	Type
FSL_MPC8349E-MITX_BSTJFP(CS0).ini	2013/8/14 3:23 PM	Configuration sett...
FSL_MPC8548PC_BSTJFP.ini	2013/8/22 1:02 PM	Configuration sett...
FSL_MPC8560ADS_BSTJFP.ini	2015/7/13 10:52 PM	Configuration sett...
FSL_MPC8569E-MDS_BSTJFP(NOR@CS0)....	2015/7/17 4:48 PM	Configuration sett...
FSL_P1010-RDB_BSTJFP(NOR@CS0).ini	2013/12/10 6:07 PM	Configuration sett...
FSL_P1021RDB_BSTJFP(NOR@CS0).ini	2013/12/24 5:28 PM	Configuration sett...
FSL_P2020RDB_BSTJFP(NOR@CS0).ini	2013/8/3 8:28 PM	Configuration sett...
FSL_P2041RDB_BSTJFP(NAND@CS1).ini	2016/3/30 1:52 PM	Configuration sett...

File name: FSL_MPC8548PC_BSTJFP.ini

JFP Configuration Files (*BSTJFI

The software will show the run result.

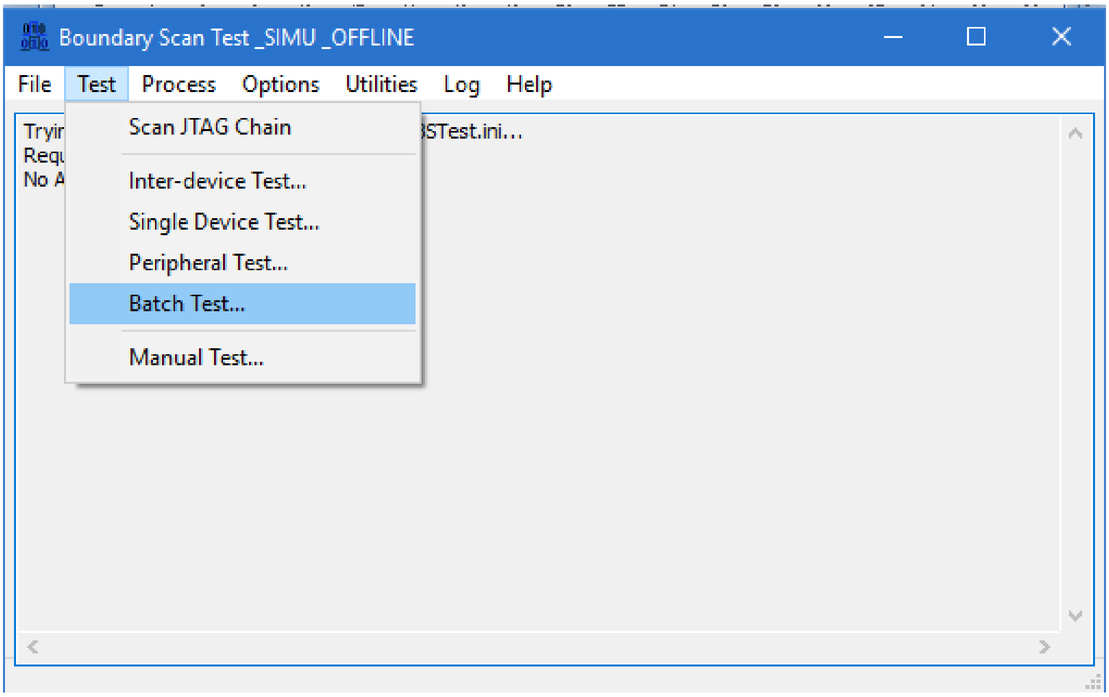
Batch Test

Edit Workspace File

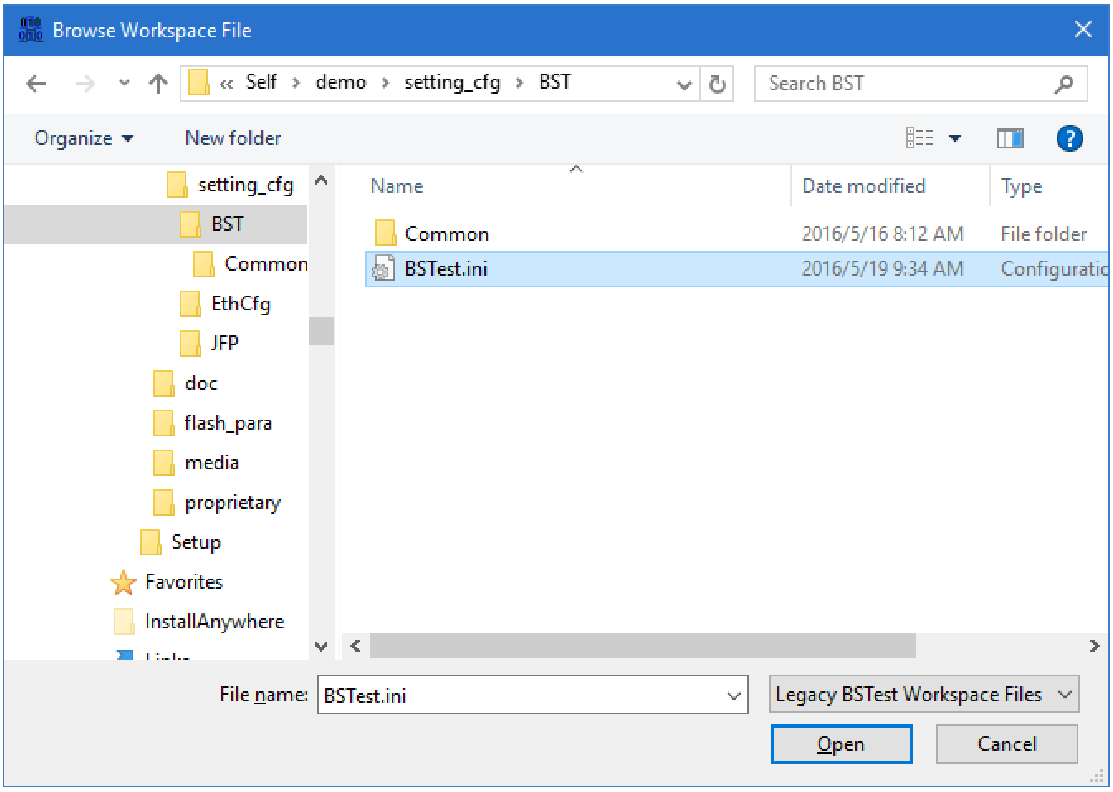
Follow “FileHelper User Manual(ENU).pdf” to create or edit a workspace file.

Run

Select menu **Test** – **Batch Test**…. See screenshot below:



Select workspace file. See screenshot below:



Name	Date modified	Type
Common	2016/5/16 8:12 AM	File folder
BSTest.ini	2016/5/19 9:34 AM	Configuratio

File name: BSTest.ini

Legacy BSTest Workspace Files

The software will show the run result.

PLD Configuration

Currently .svf and .vme files are supported.

Note:

- *A PLD maybe a CPLD/EPLD, and could also be a FPGA.*
- *.svf file could be used to perform a test besides configuration.*

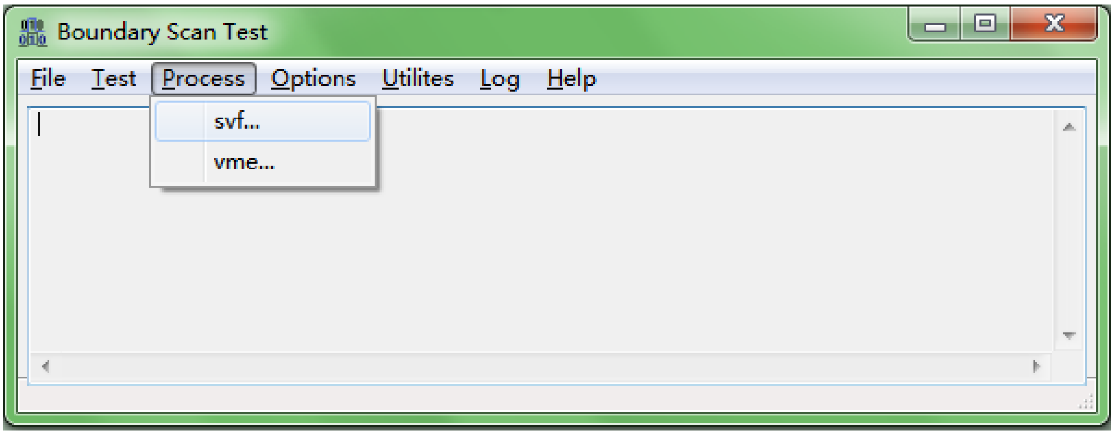
We show example of .svf in this manual. To .vme file, it’s very similar.

Get .svf File

Almost every EAD software could convert programming file from other format to .svf format. Ask help from your PLD vendor if you have difficulty.

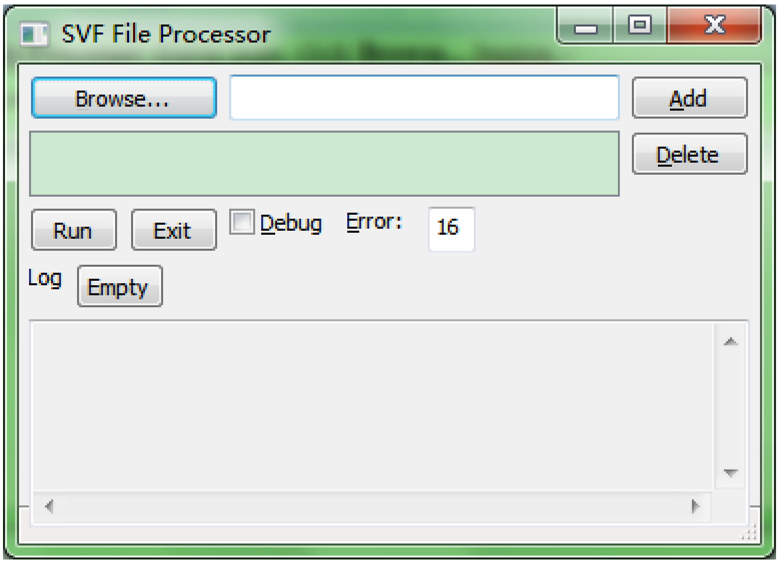
Start

Select menu **Process - .svf File**. See screenshot below:

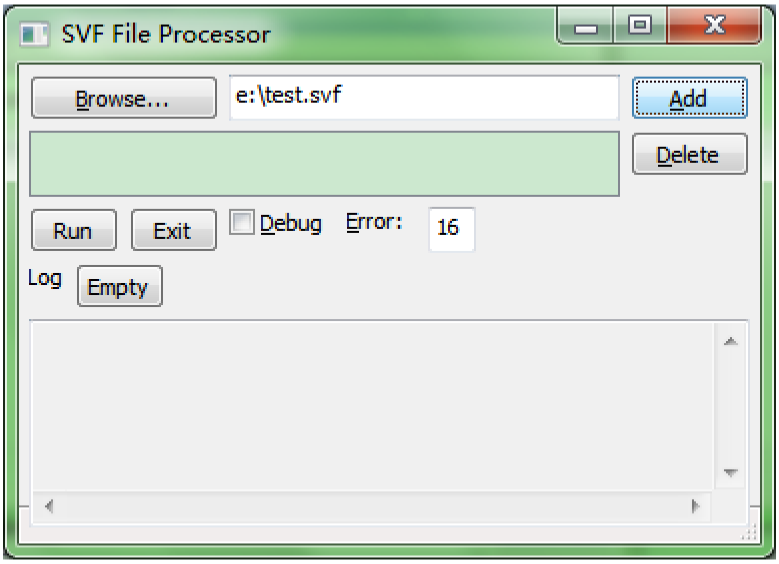


Set .svf File

When **SVF File Processor** dialog pops, click **Browse…** button.



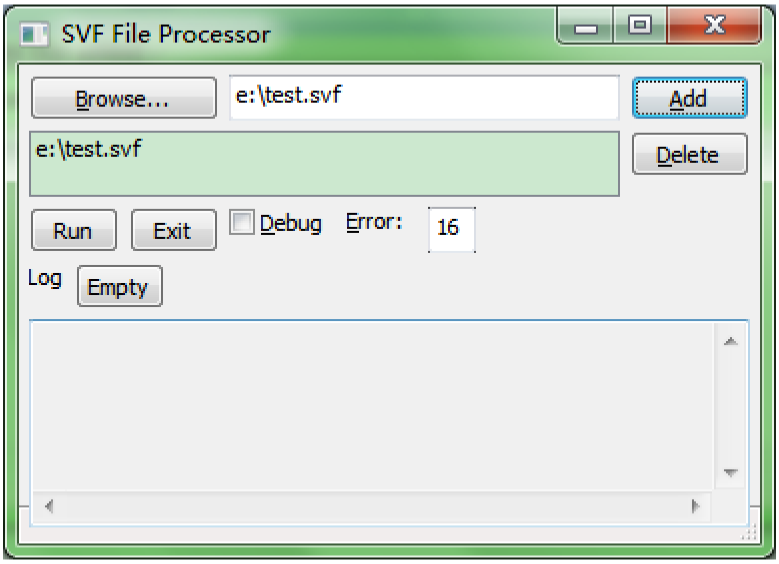
Click **Add** button after file is selected.



e:\test.svf

16

Screenshot after file added:



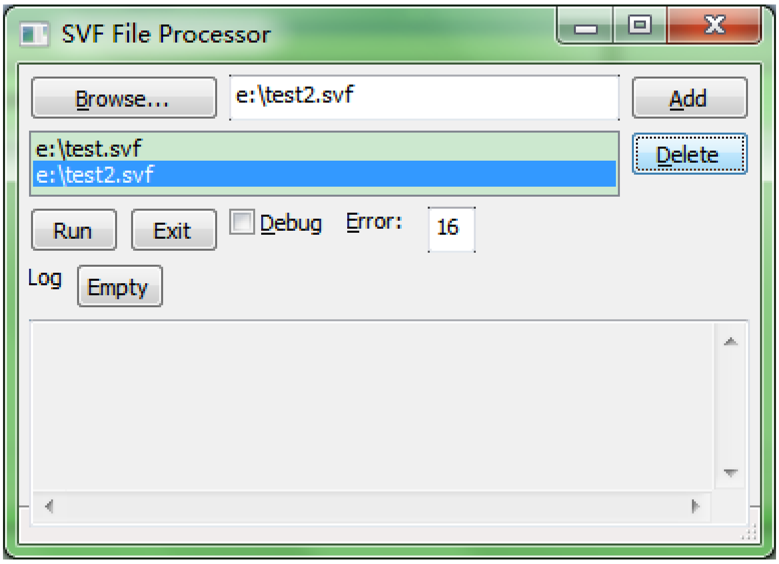
e:\test.svf

e:\test.svf

16

You may add multi files. The software will process them one by one.

If you want to remove one svf file, just select it and click **Delete** button. See screenshot below:



e:\test2.svf

e:\test.svf
e:\test2.svf

□ Debug Error: 16

Run

Click **Run** button when setting is done.
The software will show the run result.

PseudoCLI Feature

Fully automatic operation without any mouse clicking could be implemented with additional PseudoCLI feature. This feature will automatically close GUI and provide return code to caller.
Return code:

- -1: License error;
- 0: OK;
- n>0: The *n*th operation failed.

Please note: The GUI will pop up a message box when error occurs. Turn off this by setting [Option] ExitWhenFail=true in the workspace file. The caller must check the return value.

Revision History

Date	Version	Author	Changes
2020/4/19			Add PseudoCLI feature;
2019/8/18			Add output constraint to Expected Input for Single Device Test;
2019/1/31			Add Batch Set with Port Name Regular Expression
2016/8/12			Add Section “Batch Test”;
2016/8/2			Add IDT Test Control File syntax;
2016/7/20			● Add Section “Peripheral Test”; ● Re-organize sections, current order: Manual Test, Inter-device Test, Single Device Test, Peripheral Test, PLD Configuration;
2016/5/27			● Add: Attention: To get better test result and wider test coverage, please keep FPGA, CPLDs blank when testing, and do not program or configure them before test. And please keep CPUs in idle status, i.e. do not program their Boot ROMs or Flashs before test. ● The output cell of IO Setting table for Manual Test becomes Edit box instead of Combobox since V2.4.0.2, so add ‘type the output pattern string in Edit box of the cell’;
2014/12/2			● Move V2.2.2.1 improvement to first failure case; ● Add V2.3.0.0 feature;
2014/11/20			● Add section UI Difference Under Different Launching Mode;
2014/11/14			● Add screenshot of V2.2.2.1 which will show detected failure pins besides failure counter;
2014/11/13			● Add Preparation section in Single Device Test; ● Add An Example in Inter-Device Test; ● Add Tips in About the Parameters section in Inter-Device Test; ● Update text and screenshot because more netlist file formats are supported;
2014/6/9			● Add line above footer;
2013/7/30			● Format optimization;
2013/7/28			● First Release