



最具性价比的芯片内存调 适环境：**EZ-Debug**

芯测科技(上海芯复瑞)

前瞻技术暨云端服务部 经理 张耀宗

在日益蓬勃发展的集成电路产业中，确保集成电路是否符合设计规格和功能已是开发与制造过程中不可或缺的重要环节。在这关键的任务下，IC产业广泛运用自动测试装置 (Automatic Test Equipment, ATE) 来执行测试工作。然而，过去，在处理小批量或是非量产的芯片测试时，常常仅能藉由使用ATE机台进行，但这却往往伴随需要额外的成本投入以及漫长的测试时间。

而芯测科技(上海芯复瑞)近期研发一款基于计算机平台的JTAG转USB的调适环境: EZ-Debug，可以针对非量产芯片以及开发中的芯片进行快速且高效率的进行测试，不仅可以省下ATE测试的成本，同时还可以及时的知晓结果。

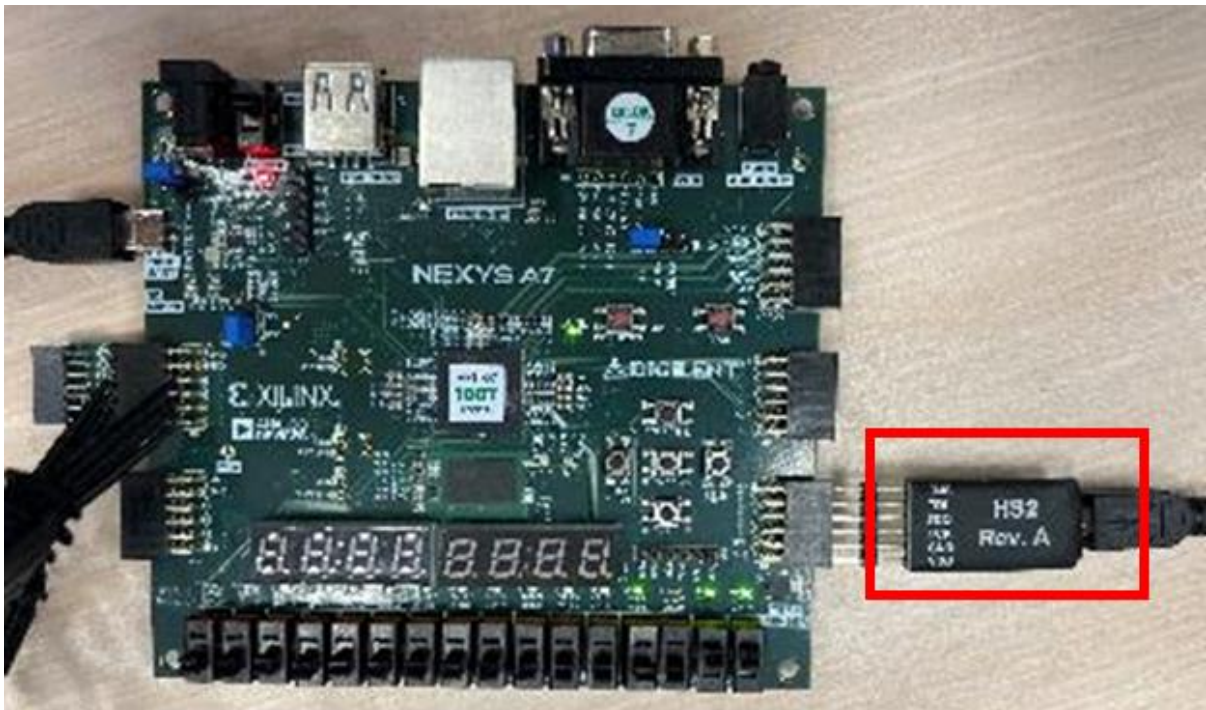
一、EZ-Debug架构

EZ-Debug可以快速且方便进行调适，EZ-Debug主要是透过转接板进行计算机与FPGA/EVB board之间的沟通。转接板与整体工具的规格示意图如下图所示。计算机与转接板是用USB做连接，转接板配合工具会将讯号转换成JTAG，再送给FPGA/EVB board进行调适。



Architecture Diagram

下图为实际使用示意图，红色框起来的是转接板，转接板右侧是FPGA，左侧是计算机端，此FPGA是用于模拟非量产chip或是开发中的chip。



二、EZ-Debug实际使用

在计算机端安装完EZ-Debug以及转接版的驱动程序后，即可开始使用调适。EZ-Debug提供两种测试模式：(1)auto test、(2>manual test。

```
Test mode Selection
(1)auto test
(2>manual test
Select an option:
```

1. EZ-Debug测试模式 - auto test

```
Select an option: 1  
set Jtag frequency(MHz): 10  
set the file of test bench(.v): test.v  
set the file of integ spec(.integ or .f): test.f
```

使用时需设定JTAG的频率、START™ v3产生的test bench file以及START™ v3产生的integ spec file，设定完成后EZ-Debug即会自动进行BIST的测试，

Jtag frequency：JTAG在FPGA/EVB的使用频率。

file of test bench：START在BII时产生可提供模拟需要的档案(ex: integ_tb.v)

file of integ spec：START在BFL时产生BII整合需要的档案(ex: [ctr_name]_spec.integ)

进行auto test时，EZ-Debug会对design里的每一个controller进行BIST的测试，并将测试结果显示在屏幕上，若所insert的BIST电路有支持LATCH_GO功能，EZ-Debug同样也会一并显示LATCH_GO的结果。

当测试结果是PASS时，则会显示Test Pass!并且LATCH_GO的结果为 '1'。

[Test Pass]

```

Test result of Controller 1 : Pass!
LATCH_GO Results of top_default_1 : 1

Test result of Controller 2 : Pass!
LATCH_GO Results of top_default_2 : 11

Test result of Controller 3 : Pass!
LATCH_GO Results of top_default_3 : 1

Test result of Controller 4 : Pass!
LATCH_GO Results of top_default_4 : 11

Test All result: Pass!

```

test result

而测试结果为FAIL时，则会显示Test Fail!并且LATCH_GO的结果为'0'。

若insert的BIST电路有支持Diagnosis功能时，EZ-Debug会进行Diagnosis的测试，并且将Diagnosis测试完成的信息显示在屏幕上。

[Test Fail]

```

Test result of Controller 1 : Fail!
LATCH_GO Results of top_default_1 : 0

Test result of Controller 2 : Fail!
LATCH_GO Results of top_default_2 : 00

Test result of Controller 3 : Fail!
LATCH_GO Results of top_default_3 : 0

Test result of Controller 4 : Fail!
LATCH_GO Results of top_default_4 : 00

Test All result: Fail!

```

test result↵

```

----- Start to Diagnosis Memory of top_default_1 -----
----- Memory of top_default_1 SEQ 1 GRP 1 MEB 1 Fail! -----
-----Diagnosis Information-----
Algorithm : 1
Sequencer ID : 1
Group ID : 1
Element : 001
Operation : 0011
Faulty Address : 0000000001
Faulty Bits : 00000000010000000000000000000000
-----
----- Memory of top_default_1 SEQ 1 GRP 1 MEB 1 Fail! -----
-----Diagnosis Information-----
Algorithm : 1
Sequencer ID : 1
Group ID : 1
Element : 001
Operation : 1000
Faulty Address : 0000000001
Faulty Bits : 00000000010000000000000000000000
-----
----- Memory of top_default_1 SEQ 1 GRP 1 MEB 1 Fail! -----
-----Diagnosis Information-----
Algorithm : 1
Sequencer ID : 1
Group ID : 1
Element : 010
Operation : 0011
Faulty Address : 0000000001
Faulty Bits : 00000000010000000000000000000000
-----

```

diagnosis result←

2. EZ-Debug测试模式 - manual test

首先可以要参阅做完整整合的INTEG testbench 中bist_testing这个task，在这个task中我们可以找到下图的CMD_DATA信息，依照下图信息即可填写设定input binary value去使用JTAG的TDI做send_command，即可开始测试。

```

top_default_CMD_DATA = {top_default_DIAG, top_default_ALG,
                        top_default_SEQ_ID, top_default_GRP_ID,
                        top_default_MEB_ID, top_default_MEN};

```

[ctr_name]_DIAG：是否要执行Diagnosis，当设定为1为启动。

[ctr_name]_ALG：当BFL选项中有开启algorithm_selection时，在testbench中就产生Controller_name_ALG的指令可以来控制想要测试的algorithm。

[ctr_name]_SEQ_ID、Controller_name_GRP_ID、Controller_name_MEB_ID：

都是用来指定目前要做测试的memory的ID。

[ctr_name]_MEN：Controller BIST enable的指令，当设定为1为启动。

JTAG的TDO会产出capture_commad，而capture_commad可参考下图INTEG

testbench中的test_result的讯号排列，即可解读capture_commad的内容。

```
{top_default_MGO, top_default_MRD,
top_default_SRD, top_default_LATCH_GO} = top_default_test_result;
```

讯号解释：

[ctr_name]_MGO：为BIST测试结果，当BIST测试fail时，为0。

[ctr_name]_MRD：为BIST测试完成时，会为1。

[ctr_name]_SRD：当Diagnosis Data已准备好时，会为1，即可进行capture
Diagnosis Data。

[ctr_name]_LATCH_GO:此讯号的width是依照START所产出的meminfo檔中的
memory数量而定，当LATCH_GO的每个讯号从1变到0时，代表此颗memory测试
fail。

使用时输入controller的个数、测试的command，capture result的长度，以及
MGO/MGD的bit number，EZ-Debug即会进行测试，并将测试结果的row-data显
示在屏幕上。

[输入测试command]

```
Select test:
(1)bist test
(2)diagnosis test

Select an option: 1

set the number of controller: 4
set the cmd data(MSB->LSB): 00110100110010011010011001
set the size of capture data: 18
```

[输入MGO/MRD bit number]

```
controller #1
set the bit number of MGO in test result: 17
set the bit number of MRD in test result: 16
controller #2
set the bit number of MGO in test result: 13
set the bit number of MRD in test result: 12
controller #3
set the bit number of MGO in test result: 8
set the bit number of MRD in test result: 7
controller #4
set the bit number of MGO in test result: 4
set the bit number of MRD in test result: 3
```


[测试结果]

```
Test result(MSB->LSB)
0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0
Test result of Controller 1 : Fail!
Test result of Controller 2 : Fail!
Test result of Controller 3 : Fail!
Test result of Controller 4 : Fail!
```