gen\_multi\_cc 工具使用说明

brain.liu

## 1功能说明

本工具用于在多CC链路测试中，将多个CC的CASE中的向量文件合并成一个多CC的case的向量文件，例如FDD/CASE0000 和TDD/CASE0000 这两个不同的CASE 需要合并成一个2CC的case. 3CC也一样操作。

## 2使用前提

### 2.1 使用工具的要求

gen\_multicc\_gen.exe是免安装，可直接使用的.exe可执行文件, 但是由于内网机的win7系统存在系统问题，因此如果直接点开exe 可能会出现如下错误：

**应用程序正常初始化(0XC0000007)失败**

如果没有上述错误，则可忽略此小节。

如果存在上述错误，则解决上述错误的方法如下：

1：联系IT人员，使用管理员权限安装win7 系统补丁:windows6.1-kb2999226-x64，此补丁为C运行库补丁

2：联系IT人员，使用管理员权，使用DirectX Repair V3.5 修复工具，检查系统，如下图所示：



点击“检查并修复”，自动完成.dll的补齐。

上述工具存放路径：P:\PHY\brain.liu\代码注释工具补丁

3：通过上述两个步骤之后，如果系统正常，则此时可以正常使用了。

4）如果第3步之后还有问题，有可能是系统缺少程序运行所需.dll导致，建议使用depends 工具进行解析，查找缺少的.dll，然后补齐即可。

### 2.2 合并CASE的要求

要求在要合并的CASE文件夹下已经存在要合并的.bin文件或者.case文件。

以2CC的合并为例，在fpga\_test\_case\_list-CA.txt中设置要和并FDD/CASE0000 和TDD/CASE0000 这两个CASE ,那么如下的要求要满足：

1. FDD/CASE0000和TDD/CASE0000文件夹中要存在一个名为“ONL”的子文件夹，
2. 在ONL文件夹下已经提前生成了各自的.case,以DP的case为例，在两个ONL文件夹中已经各自生成了onl\_cg0\_c0\_dp\_fpga.case文件。 注意：两个文件夹中的向量文件名必须一样，便于后续工具搜索文件，如果不一样，则需要修改为一样。

说明：因为复用的是前端TFC/CEP等向量和并的配置文件fpga\_test\_case\_list-CA.txt，在这个文件中，路径中没有携带“ONL”，所以这里专门说明，以免误解。工具是在ONL子文件夹中去寻找要合并的.case 或者.bin文件。详情见3.1节的说明。

## 3 合并多个CASE中的某一种向量操作步骤

需要将两个文件夹中的2个onl\_cg0\_c0\_dp\_fpga.case 合并成一个向量文件。输出到指定目录下，并输出多CC的case list.

### 3.1 设置fpga\_test\_case\_list-CA.txt

在工具目录下的fpga\_test\_case\_list-CA.txt中，按行写入要合并的case.

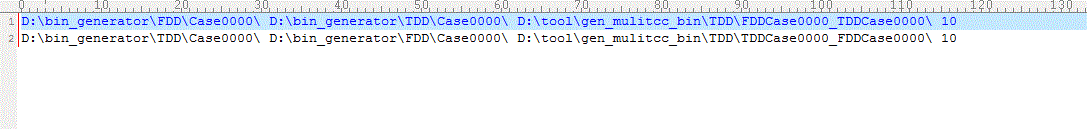
例如要合并如下两个CASE，

D:\tool\gen\_mulitcc\_bin\FDD\Case0000\;

D:\tool\gen\_mulitcc\_bin\TDD\Case0000\

注意：这里每个CASE的路径要指向向量文件所在目录的上级目录，注意不要遗漏“\”，原始向量在“D:\git\_clone\small\_tools\_develop\bin\_generator\FDD\Case0000\\*.bin“目录里;

在fpga\_test\_case\_list-CA.txt中，按行写入，在每一行，包含要合并的CASE以及输出case的路径，用1个**空格**隔开,并在末尾附带一个数字。如下图所示：



说明:上图中每一行末尾附带一个数字,这是因为前端TFC/CEP等向量的生成工具使用的配置文件就是这样的格式，所以DP/FEC的工具沿用了这个格式。所以DP/FEC的合并工具可以直接使用前端CEP等向量生成的配置文件。

每一行都表明要合并的一条多CC的CASE，如果有3个路径，则是2CC合并，最后一个路径是输出向量的目录， 如果有4个路径，则是3CC合并，最后一个路径是输出向量的目录。

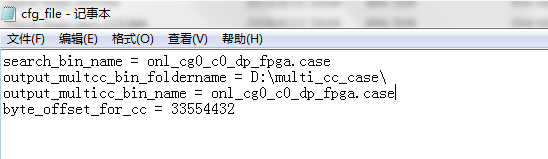
在每一行中的路径顺序表明合并的CC顺序，写在前面的路径是要合并的第一个CC，写在后面的路径是要合并的第二个CC，以此类推。

### 3.2 设置cfg\_file.txt

在工具目录下的cfg\_file.txt中，设置如下四个参数

|  |  |  |
| --- | --- | --- |
| 变量名 | 设置值 | 备注 |
| search\_bin\_name | 要合并的向量名，例如要合并onl\_cgo\_c0\_dp\_fpga.case | 在“=”号后面输入要合并的向量名 |
| output\_multcc\_bin\_foldername | 生成的多CC case list的txt 的输出目录 | 如果是当前目录，则直接输入”.\”即可，如果是指定绝对目录，则填入绝对目录，例如d:\ |
| output\_multicc\_bin\_name | 合并生成的向量名，建议跟原文件名一样，也可以另行设置 |  |
| byte\_offset\_for\_cc | 不同CC的向量之间的偏移，以Byte为单位,可根据各自向量的大小自行设置。下图示例中设置的是32Mbyte 。 | Cc0的向量在0地址，CC1的向量在32Mbyte地址，CC2 的向量在64Mbyte地址，依次类推 |
|  |  |  |

一个设置完整的cfg\_file，如下图所示：



### 3.3 点击gen\_multicc\_bin.exe

如果正常处理，cmd窗口会无消息输出，没有消息就是好消息。

如果处理过程中存在异常或者错误，则会在CMD 窗口上打印出错误消息。 错误消息中，会打印出linenum,表示的是fpga\_test\_case\_list-CA.txt中的行号，表示处理这一行时存在错误或者异常。

根据具体的输出信息，可以参考3.4.2节去检查单个CASE的处理错误。

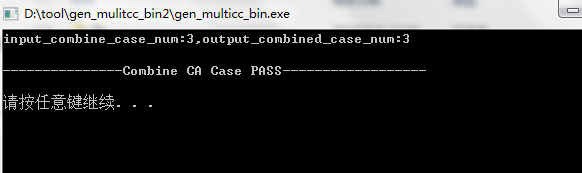
在所有case和并完成后，会在CMD 窗口上出现PASS/FAIL的打印。具体的图像见3.4.1节。

### 3.4 检查结果

#### 3.4.1 多CASE整体性检查

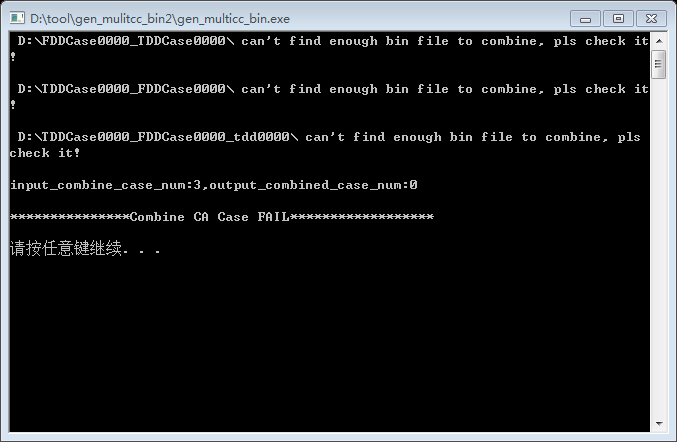
在gen\_multicc\_bin的cmd 窗口上，会显示整个case 的合并结果，

如果成功，则如下所示：



CMD 窗口中，会显示输入的要合并的CASE 数量，以及输出的合并完成的CASE 数量，两者相等则整个合并过程PASS。

如果失败，则如下所示：



上图中，表示的是：某一个合并的CASE找不到足够的bin文件来合并的错误，例如要合并afc.bin,但是找不到这个bin文件。其它错误也会打印在CMD窗口上，根据实际情况显示。

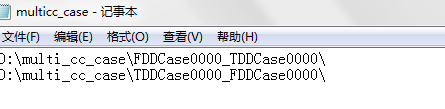
在遇到错误情况时，根据cmd窗口上的指示信息指明是哪条CASE有问题，则去该case的输出目录下检查输出的log,在log中会有相应的信息。详情见3.4.2 节。

在cfg\_file.txt中指定输出目录D:\multi\_cc\_case\，如果正常完成，输出目录下会出现生成的multicc\_case.txt, 如下图所示：



在multicc\_case.txt中，输出了成功合并的case 名。

如果发现输出的合并case的个数不符合要求，则参加3.4.2 节进行单个CASE的检查



##### 3.4.1.1 多CC case 输出路径

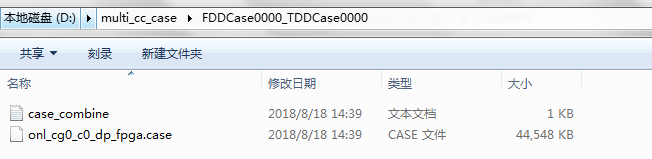
在fpga\_test\_case\_list-CA.txt中的每一行的最后一个路径，在这个路径下，存在一个“ONL”子目录，在这个子目录下会输出合并完的.case文件或者.bin文件。

#### 3.4.2单个case的检查

在每个多CC的case 文件夹下面，可以看到如下文件：

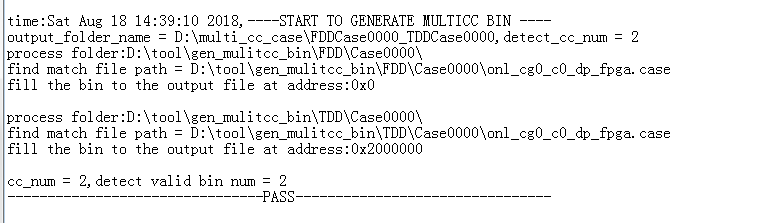
1. case\_combine.log
2. 已经合并完成的.case 文件。

如下图所示：



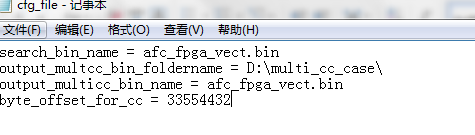
##### 3.4.2.1 case\_combine.log

在case\_combine.log中，会输出合并过程中的log,包括处理时间，处理目录，处理的向量文件，每个向量的偏移地址，并最终输出三种整体结果：分别是PASS/FAIL/EXCEPTION(异常) ，如下图所示是PASS的log：



## 4 多种向量需要合并

如果一个CASE中所需的向量有多种，则只需要修改cfg\_file.txt中的search\_bin\_name, output\_multicc\_bin\_name, byte\_offset\_for\_cc 这三个变量，输出目录可以保持不变，则就可以在之前的输出目录下合并输出新种类的向量了。

例如，之前已经合并了dp的case,现在需要合并afc的case,之需要修改cfg\_file.txt， 

然后再次运新工具，在输出目录下的各个CASE中，就生成了合并的afc的case,如下图所示：



这时打开case\_combine.log ，在文件的末尾可以看到刚刚处理afc的向量的log:

