

# 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：

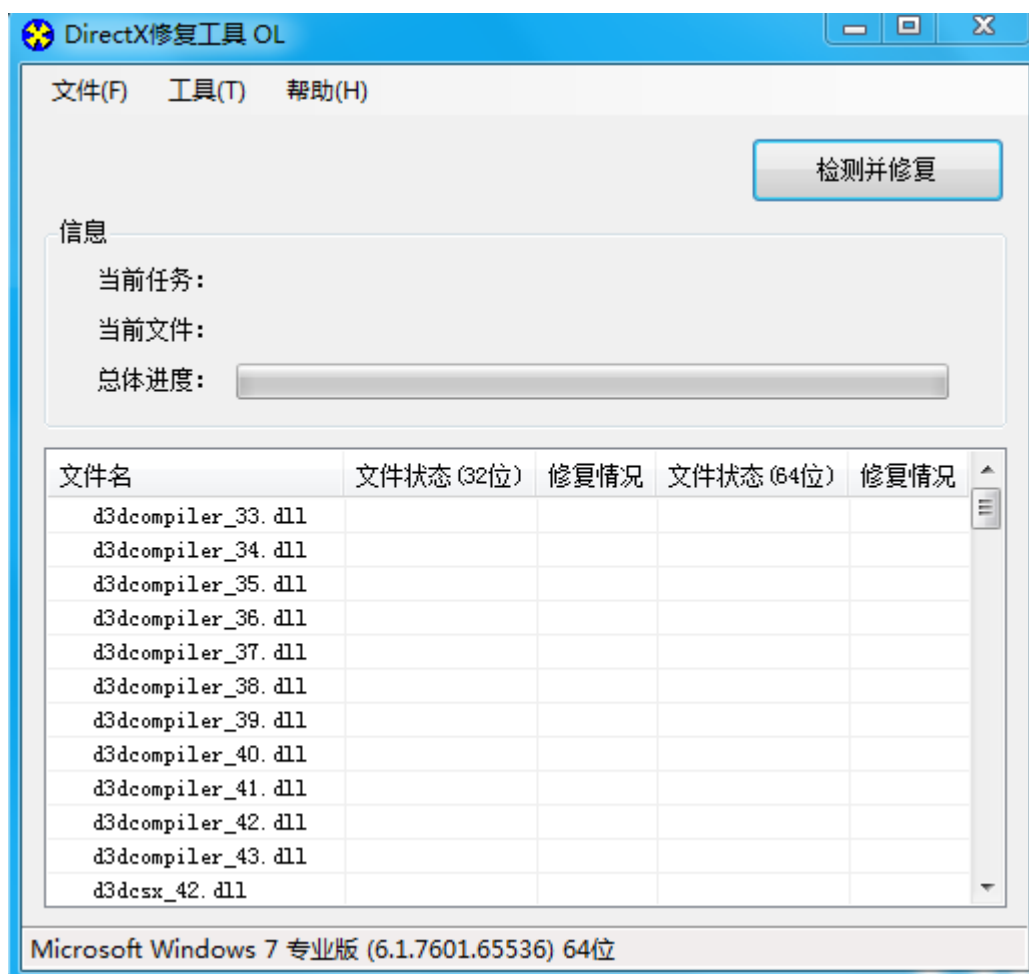
### 应用程序正常初始化(0XC0000007)失败

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 FDD/CASE0000 和 TDD/CASE0000 这两个文件夹中分别已经生成了各自的.case,以 DP 的 case 为例，在两个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。

## 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\_multicc\_bin\FDD\Case0000\;  
D:\tool\gen\_multicc\_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：

```
1 D:\bin_generator\FDD\Case0000\ D:\bin_generator\TDD\Case0000\ D:\tool\gen_multicc_bin\TDD\FDDCase0000_TDDCase0000\ 10
2 D:\bin_generator\TDD\Case0000\ D:\bin_generator\FDD\Case0000\ D:\tool\gen_multicc_bin\TDD\TDDCase0000_FDDCase0000\ 10
```

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。

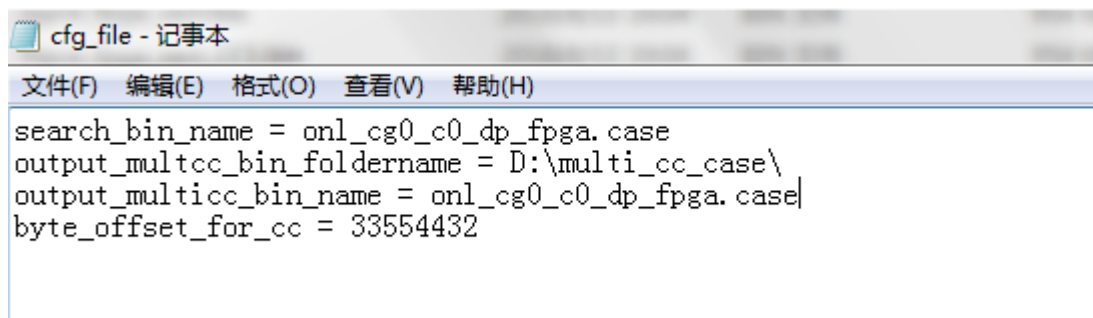
### 3.2 设置 cfg\_file.txt

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_multicc_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	Cc0 的向量在 0 地址，CC1 的向量在 32Mbyte 地址，CC2 的向量在 64Mbyte 地

	示例中设置的是 32Mbyte 。	址，依次类推

一个设置完整的 `cfg_file`，如下图所示：



```

search_bin_name = onl_cg0_c0_dp_fpga.case
output_multicc_bin_foldername = D:\multi_cc_case\
output_multicc_bin_name = onl_cg0_c0_dp_fpga.case
byte_offset_for_cc = 33554432

```

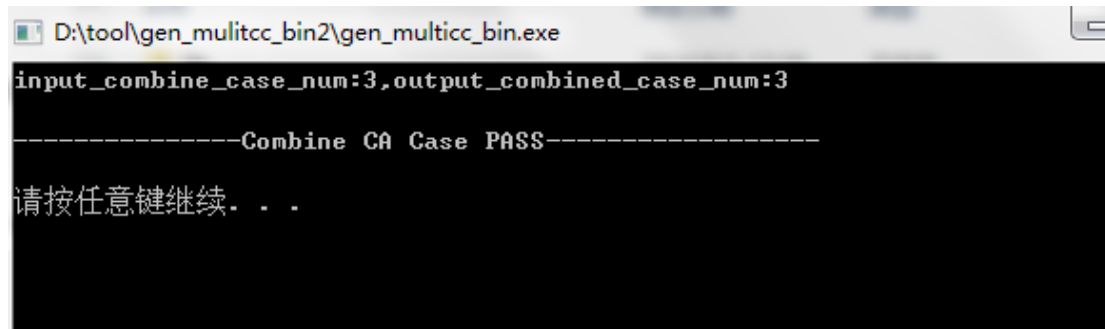
### 3.3 点击 `gen_multicc_bin.exe`

等待处理完成。

## 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：



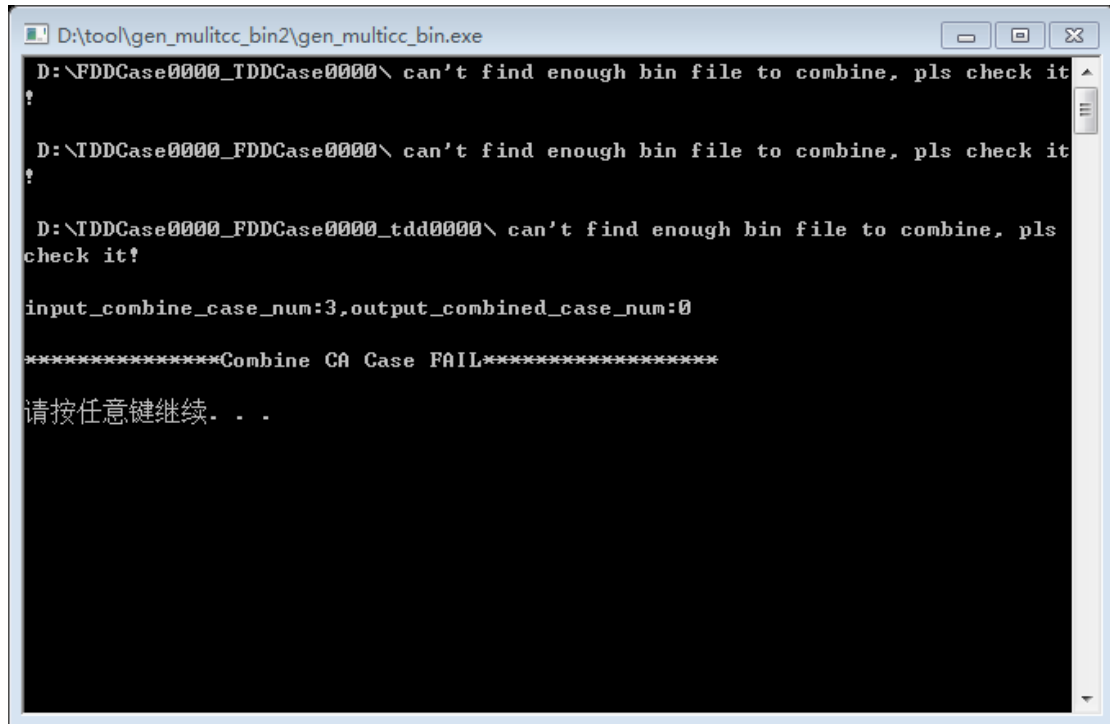
```

D:\tool\gen_multicc_bin2\gen_multicc_bin.exe
input_combine_case_num:3,output_combined_case_num:3
-----Combine CA Case PASS-----
请按任意键继续. . .

```

`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：



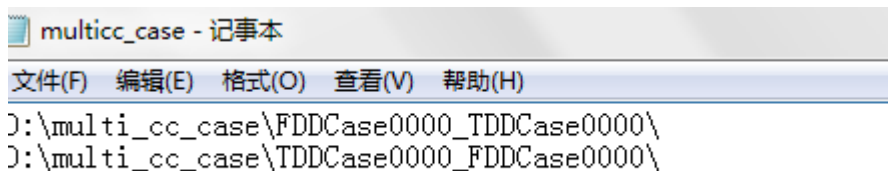
```
D:\tool\gen_multicc_bin2\gen_multicc_bin.exe
D:\FDDCase0000_TDDCase0000\ can't find enough bin file to combine, pls check it?
D:\TDDCase0000_FDDCase0000\ can't find enough bin file to combine, pls check it?
D:\TDDCase0000_FDDCase0000_tdd0000\ can't find enough bin file to combine, pls check it?
input_combine_case_num:3,output_combined_case_num:0
*****Combine CA Case FAIL*****
请按任意键继续. . .
```

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 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



```
multicc_case - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
D:\multi_cc_case\FDDCase0000_TDDCase0000\
D:\multi_cc_case\TDDCase0000_FDDCase0000\
```

### 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，就是每个多 CC case 的输出目录

### 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：

名称	修改日期	类型	大小
case_combine	2018/8/18 14:39	文本文档	1 KB
onl_cg0_c0_dp_fpga.case	2018/8/18 14:39	CASE 文件	44,548 KB

#### 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：

```
time:Sat Aug 18 14:39:10 2018, ----START TO GENERATE MULTICC BIN ----
output_folder_name = D:\multi_cc_case\FDDCase0000_TDDCase0000,detect_cc_num = 2
process folder:D:\tool\gen_multitcc_bin\FDD\Case0000\
find match file path = D:\tool\gen_multitcc_bin\FDD\Case0000\onl_cg0_c0_dp_fpga.case
fill the bin to the output file at address:0x0

process folder:D:\tool\gen_multitcc_bin\TDD\Case0000\
find match file path = D:\tool\gen_multitcc_bin\TDD\Case0000\onl_cg0_c0_dp_fpga.case
fill the bin to the output file at address:0x2000000

cc_num = 2,detect valid bin num = 2
-----PASS-----
```

## 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,

```
ctg_file - 记事本
文件(F)  编辑(E)  格式(O)  查看(V)  帮助(H)
search_bin_name = afc_fpga_vect.bin
output_multicc_bin_foldername = D:\multi_cc_case\
output_multicc_bin_name = afc_fpga_vect.bin
byte_offset_for_cc = 33554432
```

然后再次运行工具，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：

名称	修改日期	类型	大小
afc_fpga_vect.bin	2018/8/18 15:06	BIN 文件	33,021 KB
case_combine	2018/8/18 15:06	文本文档	2 KB
onl_cg0_c0_dp_fpga.case	2018/8/18 15:05	CASE 文件	44,548 KB

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:

```
time:Sat Aug 18 15:06:04 2018, ---START TO GENERATE MULTICC BIN ---
output_folder_name = D:\multi_cc_case\TDDCase0000_FDDCase0000, detect_cc_num = 2
process folder:D:\tool\gen_multicc_bin\TDD\Case0000\
find match file path = D:\tool\gen_multicc_bin\TDD\Case0000\afc_fpga_vect.bin
fill the bin to the output file at address:0x0

process folder:D:\tool\gen_multicc_bin\FDD\Case0000\
find match file path = D:\tool\gen_multicc_bin\FDD\Case0000\afc_fpga_vect.bin
fill the bin to the output file at address:0x2000000

cc_num = 2, detect valid bin num = 2
-----PASS-----
```