

gen_multi_cc 工具使用说明

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

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 添加要合并的 case 路径

在工具目录下的 multi_cc_bin_list.txt 中，按行写入要合并的 case.
例如要合并如下两个 CASE，

D:\tool\gen_multitcc_bin\FDD\Case0000\;

D:\tool\gen_multitcc_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 multi_cc_bin_list.txt 中，按行写入，CASE 之间用**分号**隔开。如下图所示：



每一行都表明要合并的一条多 CC 的 CASE，如果一行中有 2 个路径，则是 2CC 合并，如果有 3 个路径，则是 3CC 合并。

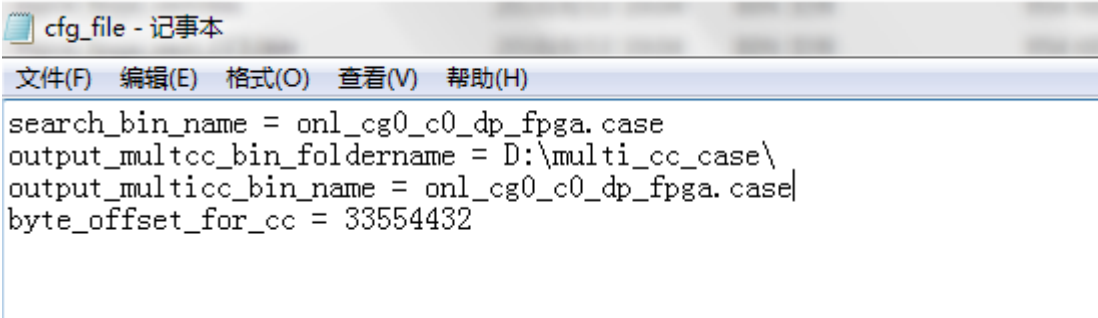
上面的配置表明要合并 3 条 CASE，其中前 2 条是 2CC 的 case,最后一条是 3CC 的 case

3.2 设置要合并的向量以及控制信息

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
<code>search_bin_name</code>	要合并的向量名,例如要合并 <code>onl_cgo_c0_dp_fpga.case</code>	在“=”号后面输入要合并的向量名
<code>output_multicc_bin_foldername</code>	多 CC 向量的输出目录	注意不要遗漏 “\”
<code>output_multicc_bin_name</code>	合并生成的向量名,建议跟原文件名一样，也可以另行设置	
<code>byte_offset_for_cc</code>	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：



```
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`

3.4 检查结果

3.4.1 整体性检查

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多 CC case 文件和 `multicc_case.txt`，如下图所示：



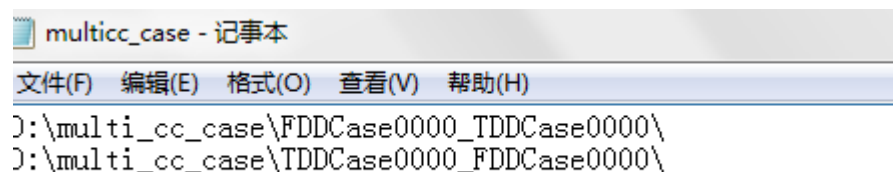
3.4.1.1 多 CC case 目录的命名

其中，每个多 CC 的 case 的文件夹命名规则是将要合并的 CASE 的名字级联而成，例如 FDD/CASE0000 和 TDD/CASE0000 合并，其生成的文件名就是 FDDCASE0000_TDDCASE0000，之所以要添加前缀，是为了区分 FDD/CASE0000 和 TDD/CASE0000 分别为主小区的 CASE，如果不加前缀，则无法区分是哪个 CC 是主小区。

3.4.1.2 多 CC 的 case list

在 multicc_case.txt 中，s 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





3.4.2 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：

本地磁盘 (D:) > multi_cc_case > FDDCase0000_TDDCase0000				
共享 ▾ 刻录 新建文件夹				
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_multicc_bin\FDD\Case0000\
find match file path = D:\tool\gen_multicc_bin\FDD\Case0000\onl_cg0_c0_dp_fpga.case
fill the bin to the output file at address:0x0

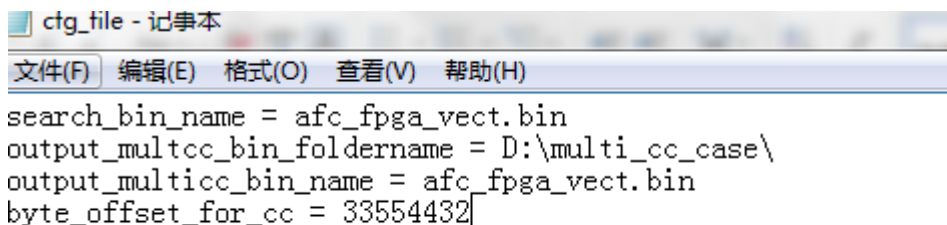
process folder:D:\tool\gen_multicc_bin\TDD\Case0000\
find match file path = D:\tool\gen_multicc_bin\TDD\Case0000\onl_cg0_c0_dp_fpga.case
fill the bin to the output file at address:0x2000000

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,



```
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:

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_multitcc_bin\TDD\Case0000\
find match file path = D:\tool\gen_multitcc_bin\TDD\Case0000\afc_fpga_vect.bin
fill the bin to the output file at address:0x0

process folder:D:\tool\gen_multitcc_bin\FDD\Case0000\
find match file path = D:\tool\gen_multitcc_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-----
```