

2024 中国研究生创“芯”大赛·EDA 精英挑战赛

赛题指南

版本	时间	修订内容
V0.1	2024-08-01	初版
V0.2	2024-10-28	引入强连通分量表示组合逻辑环定义； 明确每个题目输出文件格式； 更新问题 3、4 的检测内容和输出格式； 更新问题 5 内容和评分标准；
V0.3	2024-10-31	更正赛题一描述中“单个逻辑环”为“强连通分量”。

一、 赛题名称

数字电路仿真中组合逻辑环路分析

二、 命题单位

上海概伦电子股份有限公司

三、 赛题主席

祁仲冬（西安电子科技大学）

四、 赛题背景

用以实现基本逻辑运算和复合逻辑运算的单元电路称为门

电路。常用的门电路在逻辑功能上有与门、或门、非门、与非门、或非门、异或门、异或非门等几种。通常门电路为多输入、单输出结构，多个门电路按照一定规则组合在一起可实现具有特定功能的组合逻辑电路，例如下图为 2-bit 带进位全加器的门级电路图和 Verilog 代码。

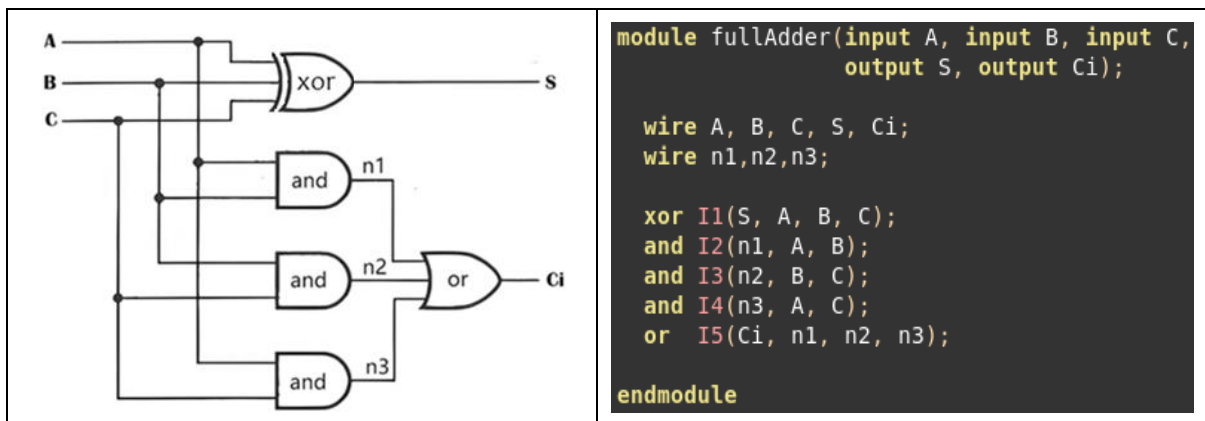


图 1 2-bit 带进位全加器门电路（左）和 Verilog 代码（右）

在组合逻辑电路中，起始于某个组合逻辑单元经过一串组合逻辑又回到起始组合逻辑单元的逻辑环路称为组合逻辑环路。组合逻辑环路分为正向反馈和负向反馈两种类型。正向反馈环路是指能够维持当前组合逻辑输出信号状态不变的反馈，常见于锁存器、寄存器和 SRAM 存储器单元等电路中。负向反馈环路是指将当前组合逻辑输出信号翻转的反馈，负向反馈环路会导致输出信号值不停翻转，常见与振荡器、伪随机数生成器等电路中。图 2 和图 3 为锁存器和振荡器门电路，其中红色信号线为锁存器和振荡器门电路中的组合逻辑环路部分。图 4 为不期望的组合逻辑

环的示例图，当 $w1$ 为 0 时， $w2$ 为 0， $w3$ 为 1；当 $w1$ 为 1 时， $w2$ 和 $w3$ 将产生震荡。

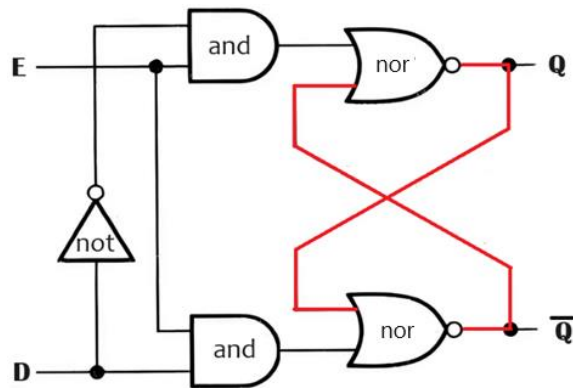


图2 锁存器门电路

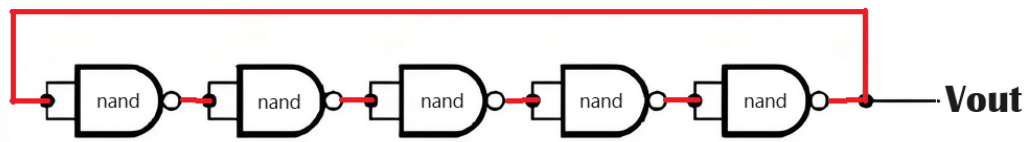


图3 由五个与非门构成的震荡器

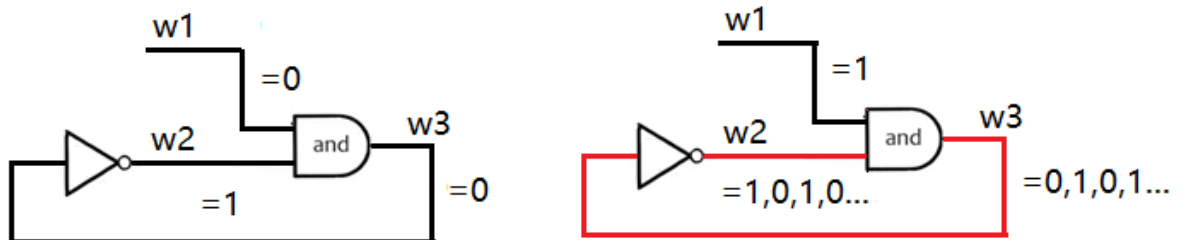


图4 不期望的组合逻辑环示例

组合逻辑环路常见于基本的门电路模块中，但在大部分数字电路设计应用场景中都不需要使用组合逻辑环路。不符合预期的组合逻辑环路可能导致多重驱动或信号震荡，导致电路功耗增加和功能错误。组合逻辑环难以被静态时序分析工具分析和计算，

可能导致仿真器进入死循环。在实践中，组合逻辑环可能相互嵌套，形成强连通分量（在有向图的数学理论中，如果一个图的每一个顶点都可从该图其他任意一点到达，则称该图是强连通的。在任意有向图中能够实现强连通的部分，称其为强连通分量）。简单的组合逻辑环也符合强连通分量的定义。在实际设计中，IC 设计人员无意引入的强连通分量可能规模巨大且结构复杂，无法人工识别，因此自动检测数字门电路中的不符合预期的组合逻辑环，分析触发条件和断开环路，帮助设计人员发现设计问题，是 EDA 工具需要解决的问题。

检测数字门电路中的组合逻辑环路需要获取门电路的拓扑结构和逻辑门信息，可通过 Verilog 仿真器的 VPI 接口实现。Verilog 语言标准支持使用 VPI（Verilog Procedural Interface）编程语言接口与 C 语言程序交互，VPI 接口提供了一套 C 语言函数，通过这些 C 语言函数可以获取 Verilog 网表的器件连接关系和行为逻辑，并且可访问和修改仿真阶段的信号值。所有 Verilog 仿真器都支持 VPI 接口，在仿真器仿真阶段，通过 VPI 接口可获取网表中逻辑门及其信号连接信息，通过这些信息可构建完整逻辑门电路信息。

五、 赛题描述

给定一个 Verilog 门级网表测试用例，在测试用例中的逻辑

门器件中，非门为单输入单输出端口，其他逻辑门均为双输入单输出端口。按 Verilog 语句定义的从左到右顺序，第一个端口为逻辑门的输出端口，其他端口均为逻辑门的输入端口。测试用例中可能包含多个强连通分量。参赛队需编写程序，通过仿真器 VPI 接口获取测试用例中逻辑门及连接信息，通过自定义数据结构构建网表信息，按照赛题要求，运行程序并输出结果。第一至四题的结果需由参赛队提供程序根据输入的测试用例输出结果文件，第五题为开放式题目，将可能震荡的强连通分量改造为不会震荡且功能基本等价的电路结构。

● 赛题一：找出测试用例中所有强连通分量。

每个强连通分量应包含信号名称和逻辑门名称两部分，请将结果输出到名为“result_1.txt”文件中。

信号名输出格式以“Loop Signals: ”为行首标识，按字典升序输出强连通分量中全部信号名称，格式为：

Loop Signals: *SignalNameA, SignalNameB, ..., SignalNameN*

逻辑门名称输出格式以“Loop Gates: ”为行首标识，按字典升序输出强连通分量中全部逻辑门名称和输入端口，格式为：

Loop Gates: *GateName1.port#, GateName2.port#, ..., GateNameN.port#*

其中“#”为逻辑门的端口序号，按 verilog 语句从左到右顺序从 0 开始编号。通常逻辑门只有一个输出端口，因此无需指定

输出端口。

每个强连通分量需要按从 1 开始递增编号，格式为如下：

1)

Loop Signals: ...

Loop Gates: ...

2)

Loop Signals: ...

Loop Gates: ...

3)

Loop Signals: ...

Loop Gates: ...

...

具体输出格式可参考文档第八节的测试用例和参考答案。

- **赛题二：找出不可能发生持续震荡(震荡发生的条件是负反馈)的强连通分量。**

强连通分量的输出格式与 1) 相同，请将结果输出到名为“result_2.txt”文件中。

- **赛题三：找出可能产生持续震荡的强连通分量，并分析发生震荡的条件。**

输出可能产生持续震荡的强连通分量(格式与赛题一相同)，

并给出要产生持续震荡时强连通分量外部输入端口信号的值。请将结果输出到名为“result_3.txt”文件。

逻辑门输入端口逻辑值的格式以“Loop Condition”为行首标识，按字典升序输出。输出格式为：

Loop Condition: GateName1.port#=0|1, GateName2.port#=0|1, ...

，其中“#”为逻辑门的端口序号。第 0 号端口是输出端口，因此#值为 1 或 2。

如果产生震荡的输入端口值有多种组合，则需要依次列出所有情况，格式为：

Loop Condition: GateName1.port#=0|1, GateName2.port#=0|1, ...

Loop Condition: GateName1.port#=0|1, GateName2.port#=0|1, ...

...

每个强连通分量和输入端口逻辑值需要按从 1 开始递增编号，格式为：

1)

Loop Signals: ...

Loop Gates: ...

Loop Condition: ...

...

2)

Loop Signals: ...

Loop Gates: ...

Loop Condition: ...

...

● **赛题四：插入最少的寄存器破坏所有强连通分量。**

为了避免组合逻辑环震荡所导致的副作用，EDA 工具往往通过插入寄存器来打断这些可能震荡的组合逻辑环。请找出插入最少寄存器的解决方案，破坏掉可能发生震荡的强连通分量。输出内容为需要断开的信号名，请将结果输出到名为“result_4.txt”文件中。

假设 GateName1.port0 与 GateName2.port#之间通过信号 SignalNameA 连接，要通过插入寄存器断开这两个门的连接，则输出格式为：

Loop Breaker: SignalNameA

如果需要插入多个寄存器，则按字典顺序列举信号名：

Loop Breaker: SignalNameA, SignalNameB, ...

每个结果需要按从 1 开始递增编号。同一个强连通分量在本题结果的编号要与在赛题三结果的编号相同，格式为：

1)

Loop Breaker: ...

2)

Loop Breaker: ...

...

- **赛题五：对组合逻辑环进行改造，并提供一个信号能够检测原始组合逻辑环电路是否发生震荡。**

对于可能发生持续震荡的组合逻辑环，请对组合逻辑环电路设计一种改造方法，使得改造后的电路满足以下三个条件：1) 在原组合逻辑环电路不发生震荡的输入条件下，改造后的电路与原电路的功能是等价的；2) 在原组合逻辑环电路会发生震荡的输入条件下，改造后的电路不会发生震荡；3) 提供一个检测是否发生震荡的信号（在改造后的测试用例中增加一个名为 OscFlag 的 wire 类型信号），当原电路发生震荡时，OscFlag 值为 1；当原电路不震荡时，OscFlag 值为 0。本题会提供一个仅包含强连通分量的 Verilog 代码测试用例，参赛队基于这个测试用例进行改造，生成一个符合语法规则的 Verilog 代码新测试用例。提交作品时需提交改造后的测试用例（名为 result_5.v），并在同级目录下附加 pdf 文档说明改造方法及原理，文档名为“result_5.pdf”。最终比赛时通过枚举所有组合逻辑环的输入条件，利用 VeriSim 仿真电路功能和指示震荡信号值，确定改

造电路是否满足题目要求。本题目的分数仅由改造后测试用例的仿真结果确定，pdf 文档仅用作答辩材料。

赛题提供资料包括：

1. 用于获取门级网表电路信息的 VPI 接口示例代码。参赛队在比赛初期可基于开源 Verilog 仿真器 iverilog 的 VPI 接口或其他解析 Verilog 网表套件方式进行开发，最终比赛环境为基于 VeriSim 数字仿真器的 VPI 接口运行程序。

2. 为方便参赛队开发和测试，提供多个逻辑门数量不同的测试用例示例。最终比赛时使用的测试用例(hidden testcase)仅包含示例测试用例的器件类型，在比赛阶段提供。

六、 评分标准

1. 赛题一至四题各为 15 分，赛题五为 20 分，性能分为 20 分，总分为 100 分。

$$FinalScore = \sum_t (AccuracyScore_t * wa_t) + Q5_Score + PerformScore$$

$$AccuracyScore_t = \sum_{q=1}^4 \left(15 * \frac{C_q}{S_q} - 5 * \frac{E_q}{S_q} \right)$$

$$Q5_Score = 20 * \frac{C_5}{S_5}$$

$$PerformScore = \begin{cases} \frac{BestTotalRuntime}{TotalRuntime} * 20, & TotalAccuracyScore \geq 30 \\ 0, & TotalAccuracyScore < 30 \end{cases}$$

其中, t 为测试用例序号,

wa_t 是第 t 个测试用例的得分权重;

q 为问题编号;

N_q 为第 q 个问题的满分值;

S_q 、 C_q 和 E_q 分别为第 q 题全部正确结果数量、参赛作品输出的正确结果数量和输出的错误结果数量;

$Q5_Score$ 是第五题分值, C_5 是改造后测试用例仿真结果符合题目要求的输入条件的数量, S_5 是改造后测试用例输入条件的总数量。

$TotalAccuracyScore$ 是参赛作品第一至四题的总分数。

$BestTotalRuntime$ 是第一至四题总分数超过 30 分的所有参赛作品中净运行时间最短的时间, $TotalRuntime$ 是参赛作品运行所有测试用例的净运行时间。净运行时间是运行全部测试用例 VeriSim 总运行时间减去 VeriSim 构建数据库时间后的时间。参赛作品需要将当前测试用例的答案全部输出到指定文件中, 确保数据完整后, 通过调用函数 `vpi_control(vpiFinish)` 通知仿真器任务完成并结束运行。VeriSim 在退出前会计算耗时, 并将结果文件复制到特定位置, 该复制文件作为参赛作品的最终结果, 请确保 VeriSim 退出前所有结果文件数据是完整的。

2. 每个测试用例的程序运行时间超过 1 小时后将被强制终

止，成绩按最终程序输出内容计分。

3. 比赛用测试用例按规模分为大、中、小三类，得分权重分别为 $0.7/N$ 、 $0.2/N$ 和 $0.1/N$ ，其中 N 为该规模测试用例的数量。比赛用测试用例只需完成第一至四题。第五题仅有一个提前发放的测试用例，该测试用例仅需完成第五题。

七、 参考资料

[1] [http://www.asic-world.com/verilog/pli6.html#Verilog_Procedural_Interface_\(VPI\)](http://www.asic-world.com/verilog/pli6.html#Verilog_Procedural_Interface_(VPI))

[2] <https://www.runoob.com/w3cnote/verilog2-pli-intro.html>

八、 附件

附件提供了一个示例门级网表测试用例和赛题参考答案。

```
1 module combLogic( w_0_0, w_0_1, w_0_2, w_0_3, w_5_0, w_5_1, w_5_2 );
2   inout w_0_0, w_0_1, w_0_2, w_0_3;
3   output w_5_0, w_5_1, w_5_2;
4   wire w_0_0, w_0_1, w_0_2, w_0_3;
5   wire w_1_0, w_2_0, w_4_0;
6   wire w_5_0, w_5_1, w_5_2;
7   not I1_0(w_1_0, w_0_0);
8   or I1_2(w_1_1, w_0_3, w_0_1);
9   or I2_0(w_2_0, w_1_0, w_0_1);
10  nand I2_1(w_2_1, w_0_2, w_1_1);
11  nand I4_0(w_4_0, w_2_0, w_0_1);
12  and I4_1(w_0_3, w_2_1, w_0_2);
13  nand I5_0(w_5_0, w_4_0, w_0_2);
14  nand I5_1(w_5_1, w_2_0, w_4_0);
15  not I5_2(w_5_2, w_0_3);
16  not I6_0(w_0_0, w_5_0);
17 endmodule
```

图 5 门级网表 Verilog 代码

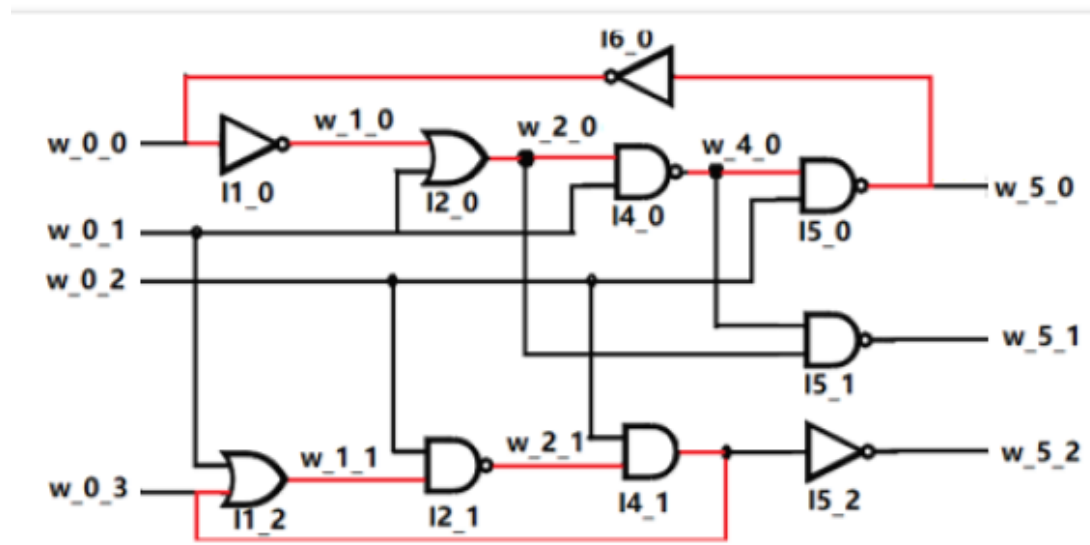


图 6 门级网表电路图

图 5 和图 6 分别为示例门级网表的 Verilog 代码和电路图，赛题参考答案如下。

● 赛题一：找出所有的组合逻辑环。

文件 result_1.txt 中内容如下：

1)

Loop Signals: w_0_0, w_1_0, w_2_0, w_4_0, w_5_0

Loop Gates: I1_0.port1, I2_0.port1, I4_0.port1, I5_0.port1, I6_0.port1

2)

Loop Signals: w_0_3, w_1_1, w_2_1

Loop Gates: I1_2.port1, I2_1.port2, I4_1.port1

● 赛题二：找出不可能发生持续震荡(震荡发生的条件是负反馈)的组合逻辑环。

文件 result_2.txt 中内容如下：

1)

Loop Signals: w_0_0, w_1_0, w_2_0, w_4_0, w_5_0

Loop Gates: I1_0.port1, I2_0.port1, I4_0.port1, I5_0.port1, I6_0.port1

- **赛题三：找出可产生持续震荡的组合逻辑环的组合逻辑电路输入信号值。**

文件 result_3.txt 中内容如下：

1)

Loop Signals: w_0_3, w_1_1, w_2_1

Loop Gates: I1_2.port1, I2_1.port2, I4_1.port1

Loop Condition: I1_2.port1=0, I2_1.port1=1, I4_1.port1=1

- **赛题四：插入最少寄存器断开组合逻辑环。**

文件 result_4.txt 中内容如下：

1)

Loop Breaker: w_1_1

- **赛题五：对组合逻辑环进行改造，并提供一个信号能够检测原始组合逻辑环电路是否发生震荡。**

略。

* 本赛题指南未尽问题，见赛题 Q&A 文件