

第四届（2022）集成电路 EDA 设计精英挑战赛

赛题指南

- 一、 **赛题名称**: Verilog 中 case 功能的综合优化
- 二、 **命题企业**: 上海安路信息科技股份有限公司、深圳鸿芯微纳技术有限公司
- 三、 **赛题 Chair**: 杨帆 复旦大学、储著飞 宁波大学
- 四、 **赛题描述**

数字电路逻辑综合对象分为数据通路和控制逻辑，而多路选择器（MUX）兼具数据通路和控制逻辑的特征。多路选择器的时延、面积优化是影响 ASIC/FPGA 逻辑综合质量的重要环节。针对多路选择器的综合优化是所有逻辑综合工具中必不可少的重要环节。

本赛题针对 Verilog 中常用的 case 语句所描述的多位数据输入输出、多位选择信号的多路选择器，要求综合优化算法输出功能等价的、由 2:1 MUX/AND/NOT 组成的网表，并优化逻辑级数和面积。

赛题程序的主体结构如图（1）所示，各个模块的详细内容将在后文阐述。参赛者可以依据算法构思和工作量评估，任选以下 2 个方案之一，编写参赛代码。赛题指定的编程语言为 C++。

方案一：利用赛题提供的输入解析模块、网表输出模块以及网表数据库的应用程序接口（API），参赛者只需编写参赛算法的代码。参赛者提交代码、可执行文件以及必要的文档说明。

方案二：参赛者自行编写所有模块的代码，包括输入解析模块、网表输出模块、网表数据库以及参赛算法。参赛者提交代码、可执行文件以及必要的文档说明。

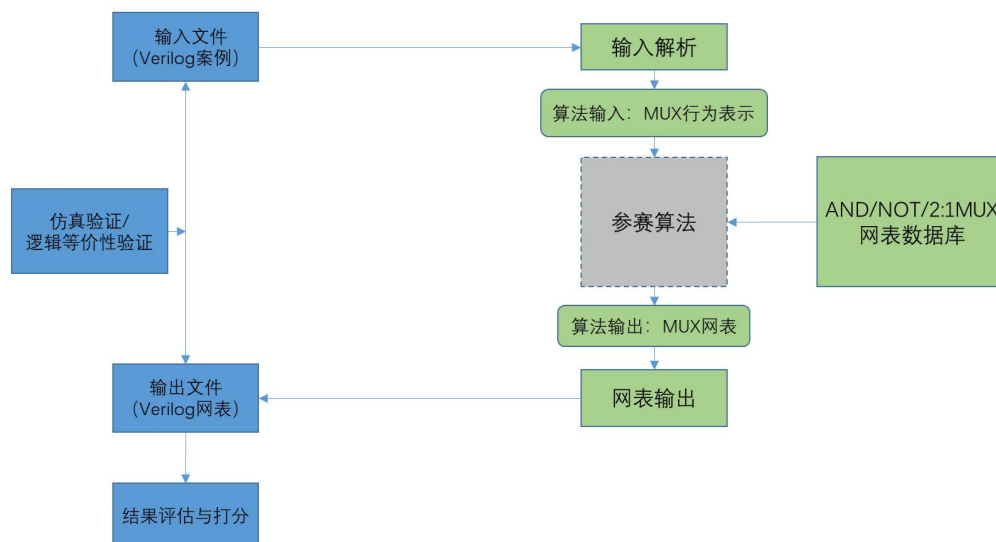


图 (1)

五、 输入文件

赛题的输入文件为以 case 语句描述 mux 行为的 Verilog。典型的输入文件如下：

```

1 module mux (
2   input [2:0] sel,
3   input [5:0] din,
4   output [2:0] dout
5 );
6
7 reg [2:0] dout_t;
8 always @(*)
9   casez (sel)
10    3'b111: dout_t = {din[5], ~din[4], 1'b0};
11    3'b10?: dout_t = {1'b1, din[1], din[3]};
12    3'b001: dout_t = {din[4], 1'b1, 1'b0};
13    3'b000: dout_t = {din[4], ~din[3], din[2]};
14    default: dout_t = {din[2], din[1], din[0]};
15  endcase
16 assign dout = dout_t;
17
18 endmodule
  
```

图 (2)

为方便程序解析，赛题的所有输入 Verilog 均以指定的、有限的 Verilog 语法描述：

- 1) 只包含端口定义和 case 语句 2 个部分；
- 2) 输入输出端口只定义名为 sel, din 和 dout 共 3 个向量变量，单个向量的位宽 dw 可以是任意值，且位宽定义必须为[dw-1 : 0]；
- 3) case 语句的条件值不能重复，且必须带有 default 条件；条件值必须由 2 进制表示，每个 bit 位可以取值为 0,1 或? ；
- 4) 对 dout_t 的赋值表达式，支持变量、变量取反、常数 0 和常数 1，且必须以单个 bit 位来描述，不允许用向量。

六、 算法输入：MUX 行为表示

程序需要解析 Verilog 输入文件并组织为算法的输入数据。赛题提供相关的解析接口。本赛题解析 Verilog 后的输入数据结构描述如下：

Verilog case 语句描述的 mux 行为，可以以一个地址矩阵加上一个数据矩阵表示。

根据选择端每一个 bit 的取值，用整型数据描述地址矩阵：

```
std::vector< std::vector<int> > addrMat ; // address matrix
```

用 '0' 表示选择端 sel 信号的某个 bit 取值 '0'，用 '1' 表示取值 '1'，用 '-1' 表示取值 '?'。则图 (2) 中案例的地址矩阵表示为：

```
[ [1,1,1],  
  [1,0,-1],  
  [0,0,1],  
  [0,0,0] ]
```

根据数据端每一个 bit 的取值和变量名，用整型数据描述数据矩阵：

```
std::vector< std::vector<int> > dataMat; // data matrix
```

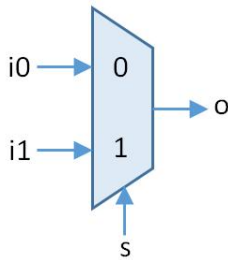


根据赛题 Verilog 指定的 din 表示，可以采取以下标识方案：用 0 表示 1' b0，用 1 表示 1' b1。用 2,4,...(2n+2)分别标识 din[0],din[1]...din[n]，用 3,5,...(2n+3)分别标识 din[0],din[1]...din[n]的取反。则图 (2) 中案例的数据矩阵表示为：

```
[ [12,11,0],  
  [1,4,8],  
  [10,1,0],  
  [10,9,6],  
  [6,4,2] ]
```

地址矩阵与数据矩阵的关系为：地址矩阵共 i 行，数据矩阵共 i+1 行；地址矩阵与数据矩阵的前 i 行一一对应，数据矩阵的第 i+1 行（最后一行）表示 default 对应的数据。

七、网表数据库

赛题指定用如下 3 种网表元件搭建算法的输出网表，典型描述如下：

性质/网表元件	2:1 MUX	AND	NOT
图例			
逻辑表达式	$o = (i0 \& \sim s) (i1 \& s)$	$o = a \& b$	$o = \sim a$
面积	2.5~4	1	0.1
时延	2	1	0.1

赛题提供对上述元件的增加/删除/查找/连接等接口。

对于每个测试案例，上述元件的面积/时延不尽相同，赛题会给出网表元件信息表（以文本文件描述），并由赛题程序解析（若选择方案二，则由参赛代码自行解析）。

八、算法参考文献

- 1) S Mitra, Avya, L.J. Efficient Multiplexer Synthesis Techniques
- 2) J Cong, Y Ding FlowMap: an optimal technology mapping algorithm for delay optimization in lookup-table based FPGA designs
- 3) P. Metzgen; D. Nancekievill Multiplexer restructuring for

FPGA implementation cost reduction

九、 算法输出：MUX 网表

由参赛算法输出的，代表 mux 功能的网表，由网表数据库中的 3 种网表元件，及其连接关系构成。

例如以下的案例：

```

case(s)
  8'b11101111: dout = di[5];
  8'b11001011: dout = di[5];
  8'b00101010: dout = di[4];
  8'b11010001: dout = di[4];
  8'b10111000: dout = di[3];
  8'b01011110: dout = di[3];
  8'b10001001: dout = di[2];
  8'b00000011: dout = di[1];
  default: dout = di[0];

```

图 (3)

一种可行的输出网表的电路图表示如下

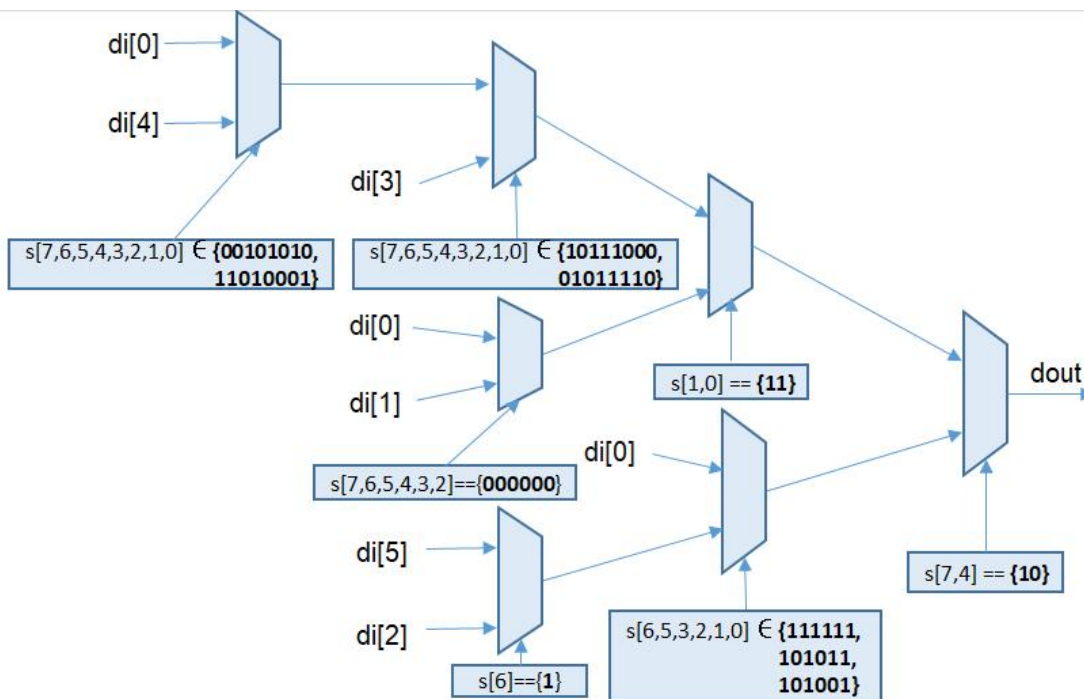


图 (4)

其主要的电路结构为 2:1 MUX, 其中方框内的为 2:1 MUX 的控制逻辑。控制逻辑例如 $s[6,5,3,2,1,0] \in \{111111, 101011, 101001\}$ 可用门级逻辑表达式表示为

$$(s[6] \& s[5] \& s[3] \& s[2] \& s[1] \& s[0]) |$$

$$(s[6] \& \sim s[5] \& s[3] \& \sim s[2] \& s[1] \& s[0]) |$$

$$(s[6] \& \sim s[5] \& s[3] \& \sim s[2] \& \sim s[1] \& s[0])$$

上述控制逻辑的一种平衡型的门级网表实现方案如下。该网表实现的面积为 13.7 (13 个 AND 和 7 个 NOT), 最大时延为 5.3 (从输入端 $s[5]$ 至输出端经过 5 个 AND 和 3 个 NOT)。

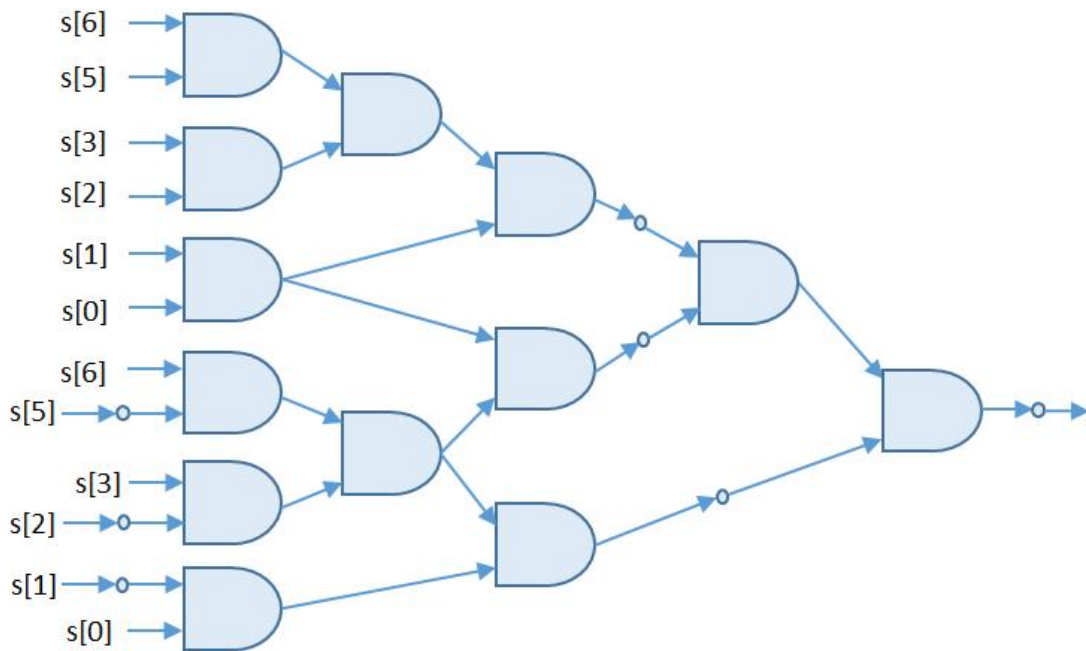


图 (5)

对于图 (4) 的电路实现, 如果对所有 2:1 MUX 的控制逻辑采用图 (4) 的平衡型网表实现:

$s[7,6,5,4,3,2,1,0] \in \{00101010, 11010001\}$: 面积 16.1 (15 个 AND 和 11 个 NOT), 最大时延 4.3 (经过 4 个 AND 和 3 个 NOT)

$s[7,6,5,4,3,2,1,0] \in \{10111000, 01011110\}$: 面积 15.9 (15 个 AND 和 9 个 NOT), 最大时延 4.3 (经过 4 个 AND 和 3 个 NOT)

$s[6,5,3,2,1,0] \in \{111111, 101011, 101001\}$: 面积 13.7 (13 个 AND 和 7 个 NOT), 最大时延 5.3 (经过 5 个 AND 和 3 个 NOT)

$s[7,6,5,4,3,2] \in \{000000\}$: 面积 6.9 (6 个 AND 和 9 个 NOT), 最大时延 3.3 (经过 3 个 AND 和 3 个 NOT)

$s[1,0] \in \{11\}$: 面积 1, 最大时延 1

$s[7,4] \in \{10\}$: 面积 1.1, 最大时延 1.1

$s[6] \in \{1\}$: 面积 0, 最大时延 0

那么图 (4) 电路的总面积为 $16.1+15.9+13.7+6.9+1+1.1+0+2.5*7$ (有 7 个 2:1 MUX, 并假设每个 2:1MUX 的面积为 2.5) = 72.2; 最大时延为 $4*1+3*0.1+4*2$ (从 $s[7]$ 至输出最多经过 4 个 AND、3 个 NOT 以及 4 个 2:1MUX) = 12.3

十、 输出文件

根据算法输出的 mux 网表, 由网表输出模块输出为网表文件, 可供仿真、逻辑等价验证或人工参考。

为方便赛题的打分系统解析, 赛题要求输出文件按以下指定的 Verilog 语法描述:

- 1) 仅使用 assign 语句对组合逻辑信号赋值;
- 2) 仅使用按位与 (&)、按位或 (|) 和按位非 (~) 描述组合逻辑表达式。

对于一个 2:1 mux 元件, 可以用类似 $(i0 \& \sim s) | (i1 \& s)$ 的逻辑表达式来表示。

例如, 对于图 (6) 中的 mux 网表, 输出文件格式如图 (7) 所示:

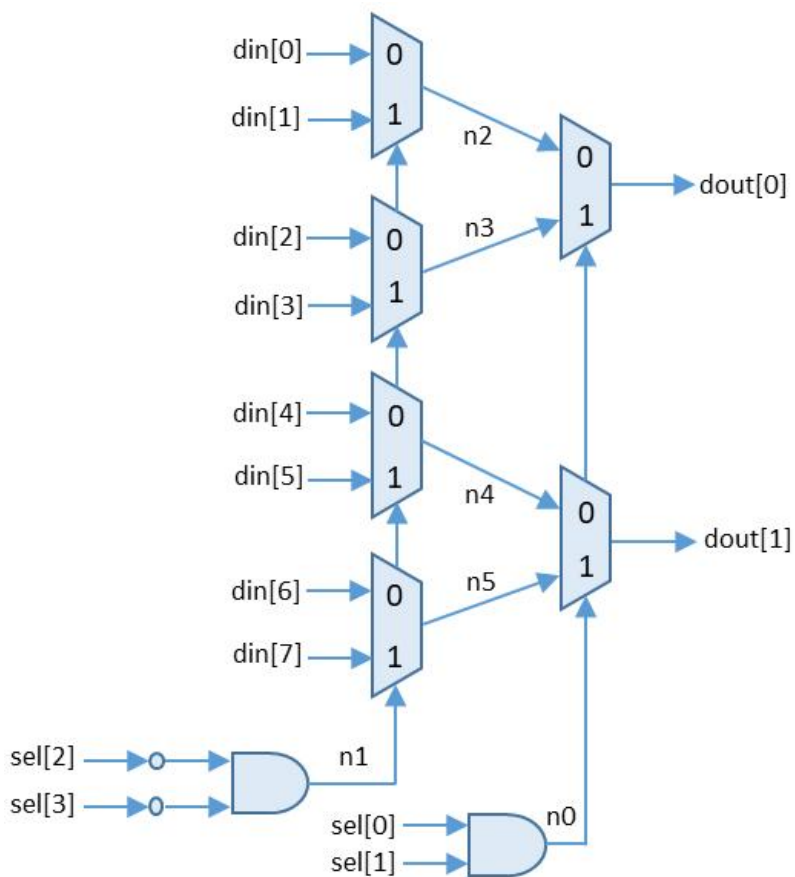


图 (6)

```
1 module mux_out (  
2     input [3:0] sel,  
3     input [7:0] din,  
4     output [1:0] dout  
5 );  
6  
7 wire n0,n1,n2,n3,n4,n5;  
8  
9 assign n0 = sel[0] & sel[1];  
10 assign n1 = ~sel[2] & ~sel[3];  
11 assign n2 = (din[0] & ~n1) | (din[1] & n1);  
12 assign n3 = (din[2] & ~n1) | (din[3] & n1);  
13 assign n4 = (din[4] & ~n1) | (din[5] & n1);  
14 assign n5 = (din[6] & ~n1) | (din[7] & n1);  
15 assign dout[0] = (n2 & ~n0) | (n3 & n0);  
16 assign dout[1] = (n4 & ~n0) | (n5 & n0);  
17  
18 endmodule
```

图 (7)

十一、 结果评估

赛题的所有测试案例根据输入规模分为大、中、小三类测试案例。赛题从 3 类测试案例中筛选并提供一部分案例给参赛者评估算法的质量；其余案例只作评分用途，不开放给参赛者。

赛题的每个测试案例包含如下文件：

- 1) 输入 Verilog
- 2) 网表元件信息表
- 3) 供仿真用的 testbench
- 4) 质量参考表（定义了该案例的总面积参考值、最大时延参考值和限定运行时间）

对于每个测试案例，有以下几个方面的结果质量评估：

- 1) 通过逻辑等价验证工具，证明输出 Verilog 文件与输入 Verilog 文件逻辑

辑等价；或者通过仿真工具和提供的 testbench，证明输出 Verilog 文件与输入 Verilog 文件逻辑等价。

- 2) 输出网表的总面积与最大时延，两者越小越好，可以结合参考值作评估。
- 3) 算法的运行时间在限定运行时间范围内。

十二、 评分方式

每个测试案例有独立的评分，遵循同样的评分标准：

1) 若案例的逻辑等价性验证失败，或仿真失败，或者超时，则该案例的评分为 n (n 为参赛队伍的数量)。

2) 案例通过条件 1) 的情况下，假设参赛队伍有 n 支，分别按总面积和最大时延的结果质量排序，第 1 名得分 1，第 2 名得分 2，以此类推，最后一名得分 n ，然后按 $0.6 * \text{最大时延得分} + 0.4 * \text{总面积得分}$ 的公式计算案例得分。例如某支队伍在某个案例上总面积得分 1，最大时延得分 5，则案例得分为 $0.6 * 5 + 0.4 * 1 = 3.4$ 。

3) 案例通过条件 1) 的情况下，如果存在结果质量相同的情况，则按分数相同，名次递增的规则评分。例如有 2 支队伍都是总面积结果最佳，则这 2 支队伍在总面积的评分上得分都是 1，第 3 个次优队伍得分 3。

对评分案例集中的每个案例评分后，总得分为单个案例得分的加权求和，其中单个案例的权重系数与该案例的规模和复杂度正相关。总得分越低，排名越高。