

# 一种加速 FPGA 布线的不可满足子式求解算法

张建民, 黎铁军, 马柯帆, 肖立权

(国防科学技术大学计算机学院, 湖南长沙 410073)

**摘要:** 随着 VLSI (Very Large Scale Integrated) 芯片设计的规模越来越大, 功能越来越复杂, 在 FPGA (Field Programmable Gate Array) 上实现或进行原型验证时, 往往会出现布线拥塞或无法布通的情况. 而不可满足子式能够迅速诊断 FPGA 无法布通的原因, 并且精确定位关键线网. 针对如何加速 FPGA 详细布线过程, 提出了一种基于消解否证的启发式局部搜索算法, 能够快速从布尔公式中提取不可满足子式. 基于典型的 FPGA 布线测试集, 与两种求解最小不可满足子式效率最高的算法进行了比较, 结果表明局部搜索算法在运行效率方面显著优于分支限界算法与贪心遗传算法, 而局部搜索算法也能得到最小不可满足子式; 并且深入分析了不可满足子式在 FPGA 详细布线中的作用, 能够加速芯片的设计与验证过程.

**关键词:** FPGA 布线; 布线约束; 布尔可满足性; 不可满足子式; 局部搜索; 消解否证

**中图分类号:** TP391 **文献标识码:** A **文章编号:** 0372-2112 (2021)06-1210-07

**电子学报 URL:** <http://www.ejournal.org.cn> **DOI:** 10.12263/DZXB.20200670

## An Unsatisfiable Subformula Computing Algorithm to Accelerate FPGA Routing

ZHANG Jian-min, LI Tie-jun, MA Ke-fan, XIAO Li-quan

(School of Computer, National University of Defense Technology, Changsha, Hunan 410073, China)

**Abstract:** With the growing scale and complexity of VLSI (Very Large Scale Integrated) chip designs, the FPGA (Field Programmable Gate Array) detailed routing process generally meets the congestion or unroutable problems during the FPGA implementation or prototype verification. The unsatisfiable subformulas can quickly diagnose the FPGA unroutable root cause, and accurately localize the critical nets. In order to accelerate the FPGA routing process, we have proposed a heuristic local search algorithm based on resolution refutation, to derive the unsatisfiable subformulas from the Boolean formulas. On the typical FPGA routing benchmarks, the local search algorithm has been compared to the two optimal minimum unsatisfiable subformula extraction algorithms. The experimental results show that the local search algorithm strongly outperforms the branch-bound algorithm and the greedy generic algorithm, and it also obtains the minimum unsatisfiable subformula. Furthermore, the unsatisfiable subformula plays an important role in FPGA routing, and it can improve the efficiency of design and verification of the VLSI chips.

**Key words:** FPGA routing; routing constraint; Boolean satisfiability; unsatisfiable subformula; local search; resolution refutation

### 1 引言

对于功能复杂且规模日益增加的微处理器芯片, 在进行 FPGA 实现时由于布线资源受限, 经常会出现布线拥塞甚至无法布通的情况. 这时就需要查找无法布通的

原因, 定位导致拥塞的线网, 从而帮助 FPGA 自动设计工具快速完成布线过程, 加速芯片设计与验证. 布尔可满足 (Boolean Satisfiability, SAT) 是计算机理论领域中的经典问题. 基于 SAT 的 FPGA 详细布线方法<sup>[1]</sup>就是将 FPGA 布线约束规约为布尔公式, 转换为可满足问题来解决, 该

公式是可满足的当且仅当 FPGA 是可布通的. 如果公式不可满足, 表明待求解的问题中包含错误与不一致, 那么需要移除公式中与不可满足原因无关的信息, 只保留反映真实错误的一部分短句, 即求解不可满足子式. 不可满足子式在诸如硬件与软件的形式化验证、集成电路 EDA 与人工智能等众多领域中, 都具有非常重要的理论研究意义与实际应用价值, 因此不可满足子式的求解方法成为近几年来学术界的研究热点.

面向 FPGA 的布线方法是不可满足子式的典型应用, 而 FPGA 详细布线问题可以规约为包含 2 个文字的短句构成的布尔公式, 从而转换为 2-SAT 问题求解. 局部搜索算法具有搜索效率高, 运算速度快的优点, 尤其针对某些类型的问题求解效率很高, 例如 2-SAT 与 3-SAT 问题, 因此 FPGA 布线问题非常适合采用局部搜索算法来解决. 但是自从融合冲突学习机制等启发式方法的 DPLL (Davis-Putnam-Logemann-Loveland) 算法出现之后, SAT 求解器得到了飞速地发展, 所以目前求解布尔不可满足子式的算法大多都是基于 DPLL 回溯搜索过程的完全方法<sup>[2-24]</sup>, 而采用局部搜索方法提取不可满足子式的相关研究却非常少.

针对如何加速 FPGA 详细布线过程, 提出了一种基于消解否证的启发式局部搜索算法, 快速从布尔公式中提取不可满足子式. 该算法首先采用局部搜索方法, 构造证明公式不可满足性的消解否证. 搜索过程融合了多种启发式的逻辑推理技术, 包括单元短句传播、二元短句消解、等价约简以及纯文字处理等. 当产生的新短句添加到公式中时, 利用蕴含关系消除公式中的冗余短句. 同时采用否证剪枝方法来动态地减小消解序列. 最后通过一个递归过程遍历公式的消解否证, 提取不可满足子式. 基于典型的 FPGA 详细布线测试集, 将局部搜索算法与支限界算法<sup>[2]</sup>以及贪心遗传算法<sup>[3]</sup>进行了对比实验, 二者是当前最小不可满足子式求解方法中效率最高的两种. 结果表明, 局部搜索算法在运行时间方面显著优于分支限界算法与贪心遗传算法, 而局部搜索算法的结果也为最小不可满足子式; 并且深入分析了不可满足子式在 FPGA 详细布线中的作用, 以及如何加速芯片的设计与验证过程.

## 2 基于 SAT 的 FPGA 布线算法

基于 SAT 的 FPGA 布线方法是将 FPGA 布线约束表示为布尔公式, 通过 SAT 求解器求解该公式的可满足性. 如果公式是可满足的, 表示 FPGA 是可布通的; 如果公式是不可满足的, 表示当前的布线策略无法布通, 那么需要查找无法布通的原因, 定位哪些线网导致拥塞, 那么布线时需要优先考虑这些关键线网. 因此, 该方法有两个显著的优点: 一个是能够同时验证所有线网

的可布通性; 另一个是可以根据布线约束公式的不可满足性查找无法布通的原因.

为了更具体的说明布尔不可满足子式在 FPGA 详细布线中所发挥的作用, 通过一个 FPGA 自动布线的示例描述不可满足子式进行错误诊断与定位的过程. 图 1 给出了一个 FPGA 详细布线的示例. 其中 FPGA 包含二维的可配置逻辑模块 CLB 阵列、互连模块 C-blk、以及交叉开关模块 S-blk. 图 1 包含 5 条线网需要在 FPGA 中布通, 分别命名为  $n_0, n_1, n_2, n_3, n_4$ ; 每两个逻辑模块之间存在两条布线通道, 分别命名为通道 0 与通道 1. 将线网进行如下方式的编码: 以线网  $n_2$  为例, 它可以编码为布尔变元  $n_{(2,0)}$  和  $n_{(2,1)}$ , 那么这两个布尔变元的取值包含下列 4 种情况:

- (1)  $n_{(2,0)} = 1$  表示线网  $n_2$  被置于通道 0;
- (2)  $n_{(2,0)} = 0$  表示线网  $n_2$  不被置于通道 0;
- (3)  $n_{(2,1)} = 1$  表示线网  $n_2$  被置于通道 1;
- (4)  $n_{(2,1)} = 0$  表示线网  $n_2$  不被置于通道 1.

通过这种编码方式, FPGA 的布线要求可以被规约为两类约束: 活跃性约束表示每条线网至少要分配到一个布线通道中; 排斥性约束保证每个布线通道至多分配一条线网.

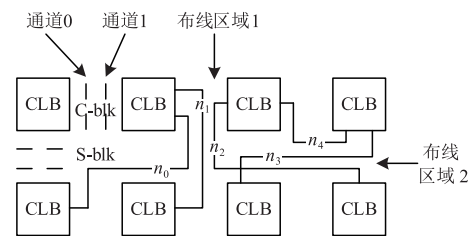


图1 一个FPGA布线示例

表 1 中列出了 FPGA 的布线约束及其所有极小不可满足子式. 如表 1 所示, 该 FPGA 布线示例包含对应于线网的 5 个活跃性约束, 以及布线区域 1 和 2 的 12 个排斥性约束. 这些约束可以构成一个布尔公式:  $\varphi = L_0 \wedge L_1 \wedge L_2 \wedge L_3 \wedge L_4 \wedge E_0 \wedge E_1 \wedge \dots \wedge E_{10} \wedge E_{11}$ . 求解  $\varphi$  的不可满足子式, 可以得到表 1 中列出的 4 个, 其中极小不可满足子式 1 表示将 3 条线网  $n_0, n_1, n_2$  同时置于只有两个布线通道的区域 1 内而导致的冲突. 类似地, 极小不可满足子式 2 指出不可能将线网  $n_2, n_3, n_4$  同时布于区域 2 中. 而极小不可满足子式 3 与 4 表示同时布下 5 条线网时, 必定导致某些排斥性约束的违反. 另外通过分析 4 个极小不可满足子式的构成, 可以发现一些关键的线网, 例如  $n_2$ , 由于它在所有极小不可满足子式中出现的频率最高, 对于区域 1 与 2 的布线最敏感, 因此这类线网在进行自动布线时需要优先考虑. 这个 FPGA 布线示例说明了不可满足子式如何在诊断错误原因与消除布线失败方面发挥作用.

表 1 FPGA 布线示例的约束及其不可满足子式

| 约束与不可满足子式     | 布尔表示   |
|---------------|--|
| 活跃性约束         | $L_0 = n_{(0,0)} \vee n_{(0,1)}$ $L_1 = n_{(1,0)} \vee n_{(1,1)}$ $L_2 = n_{(2,0)} \vee n_{(2,1)}$<br>$L_3 = n_{(3,0)} \vee n_{(3,1)}$ $L_4 = n_{(4,0)} \vee n_{(4,1)}$  |
| 布线区域 1 的排斥性约束 | $E_0 = \neg n_{(0,0)} \vee \neg n_{(1,0)}$ $E_1 = \neg n_{(0,0)} \vee \neg n_{(2,0)}$ $E_2 = \neg n_{(1,0)} \vee \neg n_{(2,0)}$<br>$E_3 = \neg n_{(0,1)} \vee \neg n_{(1,1)}$ $E_4 = \neg n_{(0,1)} \vee \neg n_{(2,1)}$ $E_5 = \neg n_{(1,1)} \vee \neg n_{(2,1)}$       |
| 布线区域 2 的排斥性约束 | $E_6 = \neg n_{(2,0)} \vee \neg n_{(3,0)}$ $E_7 = \neg n_{(2,0)} \vee \neg n_{(4,0)}$ $E_8 = \neg n_{(3,0)} \vee \neg n_{(4,0)}$<br>$E_9 = \neg n_{(2,1)} \vee \neg n_{(3,1)}$ $E_{10} = \neg n_{(2,1)} \vee \neg n_{(4,1)}$ $E_{11} = \neg n_{(3,1)} \vee \neg n_{(4,1)}$ |
| 极小不可满足子式 1    | $L_0 \wedge L_1 \wedge L_2 \wedge E_0 \wedge E_1 \wedge E_2 \wedge E_3 \wedge E_4 \wedge E_5$  |
| 极小不可满足子式 2    | $L_2 \wedge L_3 \wedge L_4 \wedge E_6 \wedge E_7 \wedge E_8 \wedge E_9 \wedge E_{10} \wedge E_{11}$  |
| 极小不可满足子式 3    | $L_0 \wedge L_1 \wedge L_2 \wedge L_3 \wedge L_4 \wedge E_0 \wedge E_4 \wedge E_5 \wedge E_6 \wedge E_7 \wedge E_{11}$   |
| 极小不可满足子式 4    | $L_0 \wedge L_1 \wedge L_2 \wedge L_3 \wedge L_4 \wedge E_1 \wedge E_2 \wedge E_3 \wedge E_8 \wedge E_9 \wedge E_{10}$   |

### 3 一种快速局部搜索算法

布尔公式的合取范式 (Conjunctive Normal Form, CNF) 表示规则为: 文字是命题变元本身或命题变元的非, 而若干个文字的析取构成短句, 若干个短句的合取组成公式。

**定义 1** 给定一个 CNF 公式  $\varphi: \varphi = \bigwedge_{i=1}^n C_i$ , 其中短句  $C_i = \bigvee_j x_j \vee \neg x_k, V = \{x \mid x \in \varphi\}$  表示命题变元的集合. 可满足问题是指给定一个赋值  $A, A$  表示从  $\varphi$  的变元集合  $V$  到真值  $\{0, 1\}$  的映射:  $V \rightarrow \{0, 1\}^{|V|}$ . 如果存在一个赋值  $A$ , 使得  $\varphi$  的真值为 1, 即  $A \models \varphi$ , 那么称公式  $\varphi$  是可满足的; 若对于任意的赋值  $A$ , 都有  $A \not\models \varphi$ , 则称公式  $\varphi$  是不可满足的。

**定义 2** 给定一个不可满足公式  $\varphi, \eta$  是公式  $\varphi$  的不可满足子式, 当且仅当  $\eta$  是不可满足的, 并且  $\eta \subseteq \varphi$ .

**定义 3** 给定不可满足公式  $\varphi$  的一个不可满足子式  $\eta, \eta$  是极小不可满足子式当且仅当  $\forall \Psi \subset \eta$ , 使得  $\Psi$  是可满足的。

**定义 4** 给定一个不可满足的公式  $\varphi$ , 以及  $\varphi$  的所有不可满足子式构成的集合:  $\{\eta_1, \eta_2, \dots, \eta_j\}$ . 那么  $\eta_k \in \{\eta_1, \eta_2, \dots, \eta_j\}$  是最小不可满足子式, 当且仅当  $\forall \eta_i \in \{\eta_1, \eta_2, \dots, \eta_j\}, 1 \leq i \leq j$ , 使得  $|\eta_k| \leq |\eta_i|$ .

如果一个不可满足子式是不可约的, 即它的所有真子集都是可满足的, 那么它是极小不可满足子式. 而最小不可满足子式是公式的所有不可满足子式中长度最小的, 即所包含的短句数最少。

**定义 5** 假设  $C_i$  与  $C_j$  为两个短句, 若  $l \in C_i$  与  $\neg l \in C_j$  是一对互补的文字, 则  $(C_i \setminus l) \vee (C_j \setminus \neg l)$  称为  $C_i$  与  $C_j$  的消解式, 其中  $l$  或  $\neg l$  称为消解基,  $C_i$  和  $C_j$  称为消解母式。

**引理 1**<sup>[25]</sup> 若短句  $C$  为  $C_i$  与  $C_j$  的消解式, 则  $(C_i, C_j) \models C$ .

**引理 2**<sup>[25]</sup> 若  $C_i = l$  与  $C_j = \neg l$  为两个单元短句, 显然  $l$  和  $\neg l$  是一对互补的文字, 则  $C_i$  与  $C_j$  的消解式为为空短句, 即  $(C_i, C_j) \models \perp$ .

**定义 6** 假设  $T$  为短句集, 且  $C$  为短句. 若存在短句的有穷序列  $C_0, C_1, \dots, C_n$ , 满足  $C_n = C$ ; 且令  $0 \leq i \leq n$ , 则短句  $C_i$  至少满足下列两个条件之一:  $C_i \in T$ ; 或  $\exists j, k, 0 \leq j, k \leq i$ , 使得  $(C_j, C_k) \models C_i$ . 那么短句  $C$  称为  $T$  的消解结果, 表示为  $T \vdash \neg C$ , 并将  $C_0, C_1, \dots, C_n$  称为由  $T$  导出  $C$  的消解序列。

**引理 3**<sup>[25]</sup> 设  $T$  为短句集, 且  $C$  为短句, 若  $T \vdash \neg C$ , 则  $T \models C$ .

**引理 4**<sup>[25]</sup> 短句集  $T$  为不可满足的当且仅当  $T \vdash \neg \perp$ .

**定义 7** 给定一个不可满足公式  $\varphi, M = \{C_0, C_1, \dots, C_n\}$  为  $\varphi$  的一个有穷消解序列. 若  $M$  的最终消解式为短句, 即  $C_n = \perp$ , 构造一个由消解步骤构成的集合  $R = \{S_i \mid S_i \text{ 为 } (C_j, C_k) \models C_i, \text{ 其中 } C_j, C_k, C_i \in M\}$ , 则  $R$  称为  $\varphi$  的一个消解否定。

**定理 1** 给定一个 CNF 公式  $\varphi, \varphi$  是不可满足的当且仅当  $\varphi$  至少包含一个消解否定  $R$ . 若令集合  $N = P(\varphi) \cap P(R)$ , 其中  $P(\varphi) = \{C \mid C \text{ 为 } \varphi \text{ 中短句}\}, P(R) = \{C \mid C \text{ 为 } R \text{ 中短句}\}$ , 那么  $\eta = \bigwedge_{C \in N} C \models \perp$  成立, 且  $\eta$  是  $\varphi$  的不可满足子式。

**证明** 首先证明第一个结论. 必要性是显然的, 若  $\varphi$  包含一个消解否定  $R$ , 根据定义 7,  $R$  的最终消解步骤的消解式为短句, 因此  $\varphi$  是不可满足的。

下面证明充分性.  $P(\varphi)$  表示公式  $\varphi$  中所有短句构成的集合, 那么  $P(\varphi)$  就为一个有限短句集。

根据引理 4 的消解原理, 若  $\varphi$  是不可满足的, 则  $P(\varphi) \vdash \neg \perp$ . 根据定义 7, 将短句集导出  $\perp$  的所有消解步骤析出, 就可以构成一个消解否定  $R$ , 因此结论成立。

下面证明第二个结论. 令  $N = P(\varphi) \cap P(R)$ , 表示消解否定  $R$  中属于原始公式  $\varphi$  的短句, 构成一个短句

集  $N$ ;

根据定义 7, 可以从消解否定  $R$  中提取出一个短句序列  $M = \{C_0, C_1, \dots, C_n\}$ , 使得  $C_n = \perp$ , 且  $\forall C_i \in M$ , 则要么  $C_i \in N$ , 要么  $(C_j, C_k) \models C_i$ , 其中  $C_j, C_k \in M$ .

根据定义 6, 短句集  $N \setminus \{C_n\} = \perp$ .

根据引理 4,  $N$  为不可满足的, 所以  $\eta = \bigwedge_{C \in N} C \models \perp$  成立. 并且  $\eta \subseteq \varphi$ , 那么  $\eta$  是  $\varphi$  的不可满足子式.

证毕

基于定理 1 的结论, 采用局部搜索策略, 提出了一种从消解否定中提取不可满足子式的算法. 算法 1 给出了一种快速局部搜索算法 FLSA 的伪代码. 该算法的输入是 CNF 格式的布尔公式, 目标函数是消解式为更短的短句或空短句. 算法启发式或随机地选择两个短句进行消解, 直到产生一个消解否定, 或者循环次数达到上限. 局部搜索算法首先通过单元短句传播过程判定公式是否不可满足, 这是因为公式不可满足的充要条件是两个单元短句消解出空短句. 如果当前的公式包含二元短句, 算法则采用一些布尔推理技术启发式地选择短句进行消解, 包括二元短句消解与等价约简. 若公式中不存在二元短句, 则随机地选择两个包含互补文字的短句进行消解; 两个短句中相同的文字越多, 那么被选择的概率就越高.

当产生的消解式加入到公式中时, 蕴含消除过程的功能是删除公式中被蕴含的冗余短句. 删除重言式函数移除包含一对互补文字的短句; 纯文字处理函数移除公式中包含纯文字的短句; 而消除重复短句函数剔除公式中重复的短句. 随着公式与消解序列的不断增大, 将会降低搜索效率, 因此采用一种否定剪枝方法避免出现内存溢出的问题, 通过踪迹剪枝函数与踪迹更新函数实现. 当公式长度大于某个预定义的常数时, 随机地选择一个短句从公式中删除, 其中较长的短句会以较大的概率被选中; 与此同时, 将该短句在消解否定中冗余的父结点以及相应的消解步骤删除. 在局部搜索过程中, 算法记录产生空短句的消解步骤, 构造消解否定. 而后通过递归函数从消解否定中求解不可满足子式.

算法 1 快速局部搜索算法 FLSA

输入: CNF 公式 formula

输出: 不可满足子式 unsatcore

```

1.  refuted = false
2.  iteration = 0
3.  while ((iteration < MAXITER) and! refuted) do
4.      if (Unit-Clause-Propagation() return UNSAT) then
5.          refuted = true
6.          break
7.      else if (there exist binary clauses) then

```

```

8.      Binary-Clause-Resolution()
9.      Equality-Reduction()
10.     else
11.         Random-Resolve(resolvent)
12.     Subsumption-Elimination(resolvent)
13.     Trace-Updating(resolvent)
14.     Non-Tautology()
15.     Pure-Literal-Handler()
16.     No-Same-Clause()
17.     if(formula.size > MAXSIZE) then
18.         Random-Remove(C)
19.         Trace-Pruning(C)
20.     iteration ++
21.     if (refuted == true) then
22.         print "unsatisfiable"
23.         unsatcore = ComputeUC(sequence)
24.     else
25.         print "unresolved"
26.     return unsatcore

```

## 4 结果对比与分析

为了验证求解不可满足子式的快速局部搜索算法在 FPGA 布线应用中的效果, 采用一组典型的 FPGA 布线测试集作为基准测试向量. 基于该测试集, 将求解最小不可满足子式效率最高的分支限界算法 (BaBA)<sup>[2]</sup>, 以及求解近似最小不可满足子式效率最高的贪心遗传算法 (CGGA)<sup>[3]</sup> 作为对比算法. 三种算法的输入都是 DIMACS CNF 格式的公式, 运行的时限设置为 1800s. 实验在一台 Intel Xeon 6126 主频为 2.6GHz 的 12 核 CPU, 内存 512GB, 操作系统为 Linux 的服务器上进行.

表 2 给出了 BaBA、CGGA 与 FLSA 三种算法基于 FPGA\_Routing 测试集的实验结果. 表 2 中的第二列和第三列分别给出了每个公式的变元数与短句数. 第四列数据表示每个公式所包含的全部极小不可满足子式的数目, 采用 CAMUS 算法<sup>[6]</sup> 求解. 第五列与第六列分别是分支限界算法的运行时间与所求解不可满足子式的短句数. 接下来两列分别是贪心遗传算法的运行时间与所提取不可满足子式的短句数. 随后两列分别表示快速局部搜索算法的运行时间与不可满足子式的短句数. 表中所有算法的运行时间都是以秒为单位. 最后一列给出了 FLSA 算法所得到的不可满足子式的短句数占原始公式短句数的百分比. 在 FPGA 测试集中, 有 9 个公式在 1800s 的时限内没有计算出全部极小不可满足子式, 在第四列中以“TO”表示.

根据表 2 的实验结果, 对于所有公式, 快速局部搜索算法的求解速度优于分支限界算法与贪心遗传算法, 而贪心遗传算法快于分支限界算法. 其中有 9 个公式 fpga\_routing7 ~ fpga\_routing15, 分支限界算法与贪心

遗传算法未能在预设的时限内得到不可满足子式,表中以“time out”表示;而快速局部搜索算法能够在时限内求解出不可满足子式.而对于测试集 fpga\_routing1 与 fpga\_routing2,由于公式的规模较小,以及有效位的缘故,因此三种算法的运行时间函数都输出为 0,对于 fpga\_routing3,快速局部搜索算法的运行时间输出也为 0,即 <0.001.另外,对于 FPGA 布线测试集,分支限界算法与局部搜索算法都能够求解出最小不可满足子式,而贪心遗传算法只给出近似最小不可满足子式.快速局部搜索算法优于两种对比算法的主要原因包括以下

三点:(1)FLSA 算法中不可满足子式的提取过程与公式可满足性判定过程紧密结合,在进行公式的可满足性判定时,就以简洁的数据结构保存了求解不可满足子式所需的消解否定,因此提取不可满足子式的过程非常高效;(2)FLSA 算法的可满足性判定与提取不可满足子式的过程实现简单,运行效率高,因此相同时间内执行的循环次数更多;(3)FLSA 算法中融合了很多诸如二元短句消解与等价约简等针对二元短句的启发式优化技术,而 FPGA 布线应用测试集中包含了很多二元短句.

表 2 三种算法在 FPGA Routing 测试集上的实验结果

| Benchmarks     | vars | clas | MUSes | BaBA     |      | CGGA     |      | FLSA    |      | Per(%) |
|----------------|------|------|-------|----------|------|----------|------|---------|------|--------|
|                |      |      |       | time     | size | time     | size | time    | size |        |
| fpga_routing1  | 10   | 17   | 4     | <0.001   | 9    | <0.001   | 11   | <0.001  | 9    | 52.9   |
| fpga_routing2  | 14   | 25   | 11    | <0.001   | 9    | <0.001   | 12   | <0.001  | 9    | 36.0   |
| fpga_routing3  | 18   | 33   | 26    | 0.03     | 9    | 0.02     | 10   | <0.001  | 9    | 27.3   |
| fpga_routing4  | 22   | 41   | 57    | 0.39     | 9    | 0.25     | 14   | 0.13    | 9    | 21.9   |
| fpga_routing5  | 26   | 49   | 120   | 2.26     | 9    | 1.74     | 15   | 1.05    | 9    | 18.4   |
| fpga_routing6  | 30   | 57   | 247   | 84.27    |      | 62.70    |      | 32.00   | 9    | 15.8   |
| fpga_routing7  | 34   | 65   | TO    | time out |      | time out |      | 60.05   | 9    | 13.8   |
| fpga_routing8  | 38   | 73   | TO    | time out |      | time out |      | 106.20  | 9    | 12.3   |
| fpga_routing9  | 42   | 81   | TO    | time out |      | time out |      | 178.10  | 9    | 11.1   |
| fpga_routing10 | 46   | 89   | TO    | time out |      | time out |      | 282.00  | 9    | 10.1   |
| fpga_routing11 | 50   | 97   | TO    | time out |      | time out |      | 434.50  | 9    | 9.3    |
| fpga_routing12 | 54   | 105  | TO    | time out |      | time out |      | 640.02  | 9    | 8.6    |
| fpga_routing13 | 58   | 113  | TO    | time out |      | time out |      | 926.50  | 9    | 8.0    |
| fpga_routing14 | 62   | 121  | TO    | time out |      | time out |      | 1309.00 | 9    | 7.4    |
| fpga_routing15 | 66   | 129  | TO    | time out |      | time out |      | 1541.00 | 9    | 7.0    |

表 2 的最后一列给出了快速局部搜索算法所得到的不可满足子式的短句数占公式总短句数的比例.通过分析得出结论:对于测试集中所有公式,快速局部搜索算法提取的不可满足子式所包含的短句非常少,对于大多数公式仅占总短句数的 7%~30%左右.因此,在 FPGA 详细布线的过程中,通过求解不可满足子式能够快速找出无法布通的原因,并精确定位关键线网,将关键线网重新布线,消除导致公式不可满足的原因,大大加速 FPGA 自动布局布线工具的布线过程,缩短 FPGA 芯片实现与原型验证的时间,从而说明了不可满足子式在诊断错误原因与消除布线失败方面具有重要的作用.

## 5 结论

基于 SAT 的 FPGA 详细布线方法是不可满足子式的典型应用之一,而局部搜索算法对于可以转换为 2-

SAT 问题的 FPGA 布线应用非常高效.本文提出了一种启发式局部搜索算法,从布尔公式的消解否定中提取不可满足子式.通过在典型 FPGA 详细布线测试集上的实验,与两种最小不可满足子式的求解算法进行了比较,结果表明快速局部搜索算法在求解效率方面优于分支限界算法与贪心遗传算法,而局部搜索算法也得到了最小不可满足子式;并且深入分析了布尔不可满足子式在 FPGA 详细布线中所发挥的作用,能够加速芯片的原型验证过程.下一步的工作是一方面继续优化局部搜索算法的实现,融入更多的启发式技术,另一方面将在更大规模的测试集上进行实验对比与分析.

## 参考文献

- [1] Nam G J, Aloul F, Sakallah K, et al. A comparative study of two Boolean formulations of FPGA detailed routing constraints[A]. Proceedings of the 2001 International Sympos-

- sium on Physical Design [ C ]. Sonoma County, CA, USA: ACM, 2001. 222 – 227.
- [ 2 ] Liffiton M H, Mneimneh M N, Lynce I, et al. A branch and bound algorithm for extracting smallest minimal unsatisfiable formulas [ J ]. *Constraints*, 2009, 14(4) : 415 – 442.
- [ 3 ] 张建民, 沈胜宇, 李思昆. 最小布尔不可满足子式的求解算法 [ J ]. *电子学报*, 2009, 9(5) : 993 – 999.  
Zhang Jianmin, Shen Shengyu, Li Sikun. Algorithm for deriving minimum unsatisfiable subformulae [ J ]. *Acta Electronica Sinica*, 2009, 9(5) : 993 – 999. ( in Chinese )
- [ 4 ] Oh Y, Mneimneh M N, Andraus Z S, et al. AMUSE: a minimally-unsatisfiable subformula extractor [ A ]. *Proceedings of the 41st Design Automation Conference [ C ]*. San Diego, CA, USA: ACM, 2004. 518 – 523.
- [ 5 ] Li Xiaowei, Li Guanghui, Shao Ming. Formal verification techniques based on Boolean satisfiability problem [ J ]. *Journal of Computer Science and Technology*, 2005, 20(1) : 38 – 47.
- [ 6 ] Liffiton MH, Sakallah KA. Algorithms for computing minimal unsatisfiable subsets of constraints [ J ]. *Journal of Automated Reasoning*, 2008, 40: 1 – 30.
- [ 7 ] 陈振宇, 徐宝文, 周从华. 一种基于消解的变量极小不可满足子公式的提取方法 [ J ]. *计算机研究与发展*, 2008, 45( s1 ) : 43 – 47.  
Chen Zhenyu, Xu Baowen, Zhou Conghua. A resolution-based approach for extracting variable minimal unsatisfiable subformulas [ J ]. *Journal of Computer Research and Development*, 2008, 45( s1 ) : 43 – 47. ( in Chinese )
- [ 8 ] 赵相福, 欧阳丹彤. 使用 SAT 求解器产生所有极小冲突部件集 [ J ]. *电子学报*, 2009, 37(4) : 804 – 810.  
Zhao Xiangfu, Ouyang Dantong. Deriving all minimal conflict sets using satisfiability algorithms [ J ]. *Acta Electronica Sinica*, 2009, 37(4) : 804 – 810. ( in Chinese )
- [ 9 ] Nadel A. Boosting minimal unsatisfiable core extraction [ A ]. *Proceedings of the 10th International Conference on Formal Methods in Computer Aided Design [ C ]*. Lugano, Switzerland: IEEE, 2010. 221 – 229.
- [ 10 ] Marques-Silva J, Lynce I. Onimproving MUS extraction algorithms [ A ]. *Proceedings of the 14th International Conference on Theory and Applications of Satisfiability Testing [ C ]*. Ann Arbor, MI, USA: Springer, 2011. 159 – 173.
- [ 11 ] Belov A, Lynce I, Marques-Silva J. Towards efficient MUS extraction [ J ]. *Journal AI Communications*, 2012, 25(2) : 97 – 116.
- [ 12 ] Nadel A, Ryvchin V, Strichman O. Efficient MUS extraction with resolution [ A ]. *Proceedings of the 13th International Conference Formal Methods in Computer Aided Design [ C ]*. Portland, OR, USA: IEEE, 2013. 197 – 200.
- [ 13 ] Marques-Silva J, Previti A. On computing preferred MUSes and MCSes [ A ]. *Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing [ C ]*. Vienna, Austria: Springer, 2014. 58 – 74.
- [ 14 ] Bacchus F, Katsirelos G. Using minimal correction sets to more efficiently compute minimal unsatisfiable sets [ A ]. *Proceedings of the 27th International Conference on Computer Aided Verification [ C ]*. San Francisco, CA, USA: Springer, 2015. 70 – 86.
- [ 15 ] Bacchus F, Katsirelos G. Finding a collection of MUSes incrementally [ A ]. *Proceedings of the 13th International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems [ C ]*. Banff, AB, Canada: Springer, 2016. 35 – 44.
- [ 16 ] Zhao W, Liffiton M H. Parallelizing partial MUS enumeration [ A ]. *Proceedings of IEEE 28th International Conference on Tools with Artificial Intelligence [ C ]*. San Jose, CA, USA: IEEE, 2016. 464 – 471.
- [ 17 ] Gregoire E, Izza Y. Boosting MCSes enumeration [ A ]. *Proceedings of the 27th International Joint Conference on Artificial Intelligence [ C ]*. Stockholm, Sweden: ACM, 2018. 1309 – 1315.
- [ 18 ] Narodytska N, Bjorner N, Marinescu M C, et al. Core-guided minimal correction set and core enumeration [ A ]. *Proceedings of the 27th International Joint Conference on Artificial Intelligence [ C ]*. Stockholm, Sweden: ACM, 2018. 1353 – 1361.
- [ 19 ] Liu Shaofan, Luo Jie. FMUS2: An efficient algorithm to compute minimal unsatisfiable subsets [ A ]. *Proceedings of 2018 International Conference on Artificial Intelligence and Symbolic Computation [ C ]*. Suzhou, China: Springer, 2018. 104 – 118.
- [ 20 ] Luo Jie, Liu Shaofan. Accelerating MUS enumeration by inconsistency graph partitioning [ J ]. *Science China Information Sciences*, 2019, 62: 212104.
- [ 21 ] Mencia C, Kullmann O, Ignatiev A, Marques-Silva J. On computing the union of MUSes [ A ]. *Proceedings of 22nd International Conference on Theory and Applications of Satisfiability Testing [ C ]*. Lisbon, Portugal: Springer, 2019. 211 – 221.
- [ 22 ] Jaroslav B, Kuldeep MS. Approximate counting of minimal unsatisfiable subsets [ A ]. *Proceedings of the 32nd International Conference on Computer Aided Verification [ C ]*. Los Angeles, CA, USA: Springer, 2020. 1 – 23.
- [ 23 ] Jaroslav B, Cerna I. Rotation based MSS/MCS enumeration [ A ]. *Proceedings of the 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning [ C ]*. Alicante, Spain: EasyChair, 2020. 120 – 137.

[24] Jaroslav B, Cerna I. MUST: minimal unsatisfiable subsets enumeration tool [ A ]. Proceedings of the 26th International Conference on Tools and Algorithms for the Construction and Analysis of Systems [ C ]. Dublin, Ireland; Springer,

2020. 135 – 152.

[25] Gallier J H. Logic for Computer Science: Foundations of Automatic Theorem Proving [ M ]. New York: Harper & Row Publishers Inc, 2003.

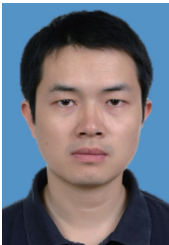
#### 作者简介



张建民 男, 1979 年生于山西晋中. 现为国防科技大学计算机学院副研究员. 主要研究方向为可满足求解和 VLSI 形式化验证.  
E-mail: jmzhang@nudt.edu.cn



马柯帆 男, 1985 年生于湖南永州. 现为国防科技大学计算机学院助理研究员. 主要研究方向为可满足求解和 FPGA 加速.  
E-mail: makefan14@nudt.edu.cn



黎铁军 男, 1977 年生于湖北十堰. 现为国防科技大学计算机学院研究员. 主要研究方向为 VLSI 功能验证和微处理器体系结构.  
E-mail: tjli@nudt.edu.cn



肖立权 男, 1969 年生于湖北孝感. 现为国防科技大学计算机学院研究员, 博士生导师. 主要研究方向为高性能计算机体系结构.  
E-mail: lqxiao2015@163.com