

基于网络芯片功能验证效率的探析*

杨飞, 蒋林

(西安邮电学院 电子工程学院, 陕西 西安 710061)

摘要: 讨论了分配问题匈牙利法在网络芯片功能验证中的应用, 在建立了网络芯片模型的基础上提出验证需求, 针对网络芯片验证的实际问题对匈牙利法进行了扩展和改进, 提出了适应于分级流水芯片提高其功能验证效率的有效方式, 改进后的匈牙利法具有更好的适用性。芯片验证结果证明了其可行性及其优越性。

关键词: 包交换; 功能验证; 匈牙利法; 覆盖率

中图分类号: TP391

文献标识码: A

文章编号: 1674-7720(2011)22-0055-04

The research on the efficiency of network chip functional verification

Yang Fei, Jiang Lin

(School of Computer Science, Xi'an University of Posts and Telecommunications, Xi'an 710061, China)

Abstract: The Hungary law of distribution was discussed in this article on the functional verification of the network chip in application. With the basis of the establishment of a network model, the requirements of verification were proposed for and the Hungary law was extended and improved in the practical problems on the network chip. Finally, the effective way in verification was put forward to in order to improve the efficient in verification. The improved Hungary law is better in applicability. The feasibility and superiority are proved in the verification results.

Key words: packet switching; functional verification; Hungary law; coverage

包交换又称分组交换, 是一种传送数据的方法, 将用户传送的数据包划分成一定长度的单元, 每个单元称为一个分组。在每个分组的前面加上一个分组头, 用以指明该分组发往何地址, 然后由交换机根据每个分组的地址标志, 将他们转发至目的地, 这一过程称为分组交换。包交换处理分组的一般过程是: 将收到的数据包先放入缓存, 再查找转发表, 找出到某个目的地址应从哪个端口转发, 然后由交换机构将该数据包传递给适当的端口转发出去^[1]。由于包交换功能复杂, 其硬件设计达到百万门级, 而验证百万门的设计是一件非常难的事情。目前进行功能验证有效的策略主要有两种^[2]: 一种是传统的功能验证方法, 包括模拟验证、形式验证、基于断言的验证三种方法, 其特点是基于特定测试向量的针对特定功能的定向功能验证; 另一种是基于伪随机向量的针对覆盖率的随机功能验证。当然两种验证方法又有交

叉。但针对一般的设计, 其验证策略通常先进行定向功能验证, 保证设计满足设计规范的功能要求, 但其很难避免设计之外的缺陷及错误, 也很难满足代码覆盖率的要求, 所以需要进一步用随机测试向量进行随机验证^[3]。但无论哪种策略在进行验证时, 在顶层 testbench 加载激励时, 不同的数据必然有自己特定的数据通道, 但不同的数据通道在进行功能验证时, 其效率是不同的, 因而有必要从数据通道方面针对验证进行研究, 以提高功能验证效率。

本文首先根据以上包交换工作过程构造定向功能验证模型, 根据验证需求提出如何在满足需求的基础上高效地进行定向功能验证, 并对结果进行分析。

1 模型概述

本文将整个包交换系统划分成三个部分, 分别是输入缓存管理、查找管理和输出管理。模块之间数据通道

* 基金项目: 陕西省“13115”科技创新工程重大科技专项(2009ZDKG-43); 陕西省教育厅科研计划项目(2010JK840)

网络与通信

Network and Communication

采用分级流水管理的方式,假定输入缓存管理模块分为两个独立的功能模块,以分别处理不同类型的输入数据。数据处理完成后存入缓存,查找管理模块从缓存中读出数据包头信息进行分类管理,假定其同样有两个独立的功能模块。输出管理模块从查找模块读出包头信息,再从缓存模块读出输出的数据包信息,并将此数据包信息转发出去,在此假定模块也有两个独立的功能模块,此时建立系统模型输出管理如图1所示。

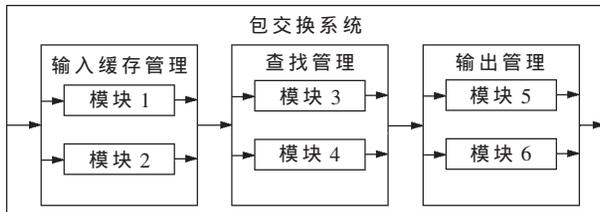


图1 包交换系统结构图

2 验证需求

针对系统技术规范要求,在系统顶层需要定向测试6个模块的功能,这样在顶层从数据包进入系统到数据包输出系统,根据遗传学数据从输入到输出共有8条通路,每条通路均可进行3个模块的功能测试。根据实际测试来看,要想完全测试一个模块的功能,必然很难保证其余模块测试功能的完整性,由于每条验证通路针对某一模块的测试效率是不同的,这样有必要针对某一模块的功能选择一条最高效的验证通路来测试此模块的功能,此条通路主要保证测试模块功能的完整性,可以辅助测试其他两个模块的部分功能。以下将介绍如何来选取最为高效的验证通路的方法。

3 验证策略

以上讨论了模型的建立以及验证需求,下面将建立验证问题的数学模型并计算出最优策略。

3.1 构造模型

首先根据系统结构图,得出系统的验证通路共有8条,如表1所示。

假定针对某一模块每条通路的测试效率系数,即每条测试通道完成某一模块功能测试所需的时间(h),

由于一条通道并不能覆盖所有模块,所以对于不能测试的模块认为其效率系数无限大,用字母 a 表示。其效率系数分布如表2所示。

设 C_{ij} 为第 i 条通道测试第 j 个模块的效率:

$$x_{ij} = \begin{cases} 1, & \text{第 } i \text{ 条通道测试第 } j \text{ 个模块} \\ 0, & \text{第 } i \text{ 条通道不测试第 } j \text{ 个模块} \end{cases}$$

则完成6个模块的测试最优效率为:

$$\min \sum_{i=1}^8 \sum_{j=1}^6 C_{ij} x_{ij}$$

表2 测试效率系数分布

模块	通路							
	1	2	3	4	5	6	7	8
1	5	3	2	4	a	a	a	a
2	a	a	a	a	3	4	2	5
3	7	8	a	a	1	2	a	a
4	a	a	5	7	a	a	0	4
5	2	a	3	a	7	a	9	a
6	a	3	a	8	a	4	a	9

从而其模型如下:

$$\begin{aligned} \max z &= \sum_{i=1}^8 \sum_{j=1}^6 C_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^m x_{ij} = 1 \quad j=1, 2, \dots, 6 \\ & \sum_{j=1}^m x_{ij} = 1 \quad i=1, 2, \dots, 8 \end{aligned}$$

而标准型分派问题的数学模型如下:

设第 i 个人完成第 j 项工作所需要的时间或资源为 C_{ij} , 称为效率系数, 引入 0-1 变量 x_{ij} :

$$x_{ij} = \begin{cases} 1, & \text{表示分派第 } i \text{ 个人完成第 } j \text{ 项工作} \\ 0, & \text{表示不分派第 } i \text{ 个人完成第 } j \text{ 项工作} \end{cases}$$

从而其数学模型如下:

$$\begin{aligned} \max z &= \sum_{i=1}^n \sum_{j=1}^m C_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{i=1}^m x_{ij} = 1 \quad j=1, 2, \dots, n \\ & \sum_{j=1}^m x_{ij} = 1 \quad i=1, 2, \dots, n \\ & x_{ij} = 0 \text{ 或 } 1 \quad i, j=1, 2, \dots, n \end{aligned}$$

3.2 解决策略

由以上的对比可以发现此问题非标准形式的分派问题,解决的思想是:将成本矩阵化为方阵, $n \times m$ 矩阵化为 $n \times n$ 或 $m \times m$ 矩阵。

(1) 匈牙利法的基本原理^[4]

定理1: 设分派问题的效率矩阵为 $C=(C_{ij})_{n \times n}$, 若将该矩阵的某一行或某一列的各元素都减去同一常数, 得矩阵 $C'=(C'_{ij})_{n \times n}$ 。则以 C' 为效率矩阵的分派问题与原分派问题的最优解相同。

定理2: 若效率矩阵 C' 中每行、每列中至少有一个零元素, 且存在位于不同行、不同列的 n 个零元素(称独立零元素), 则令 n 个独立 0 元素位置的决策变量为 1, 其余决策变量为 0, 得到一个最优方案。

(2) 参考文献[5]针对最大化问题的收益问题不是方阵的情况提出补 0 转换为方阵, 而本文要针对最小化问题不是方阵的情况, 因而提出补 ∞ 转换为方阵的策略。具体如下:

① 当模块数 m 多于测试通道数 n 时, 增添虚构的

网络与通信 Network and Communication

$m-n$ 行,补成方阵,但对应的 $C_{ij}=\infty$,即这些虚构的通道测试模块的效率系数取为 ∞ ,仍可按标准式采用匈牙利法来求解,只是在最后的结果中应去掉虚行的解。

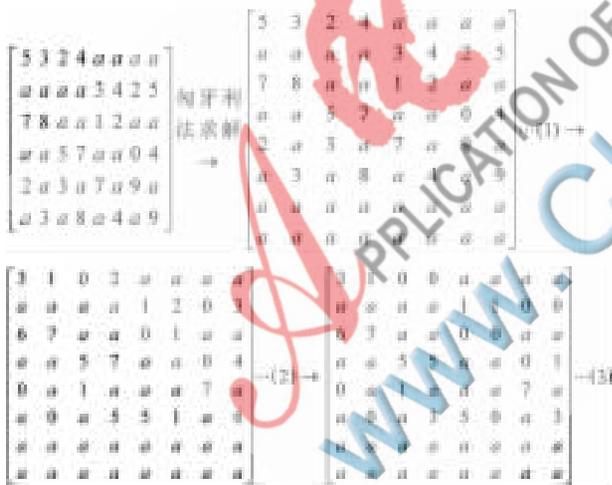
②当模块数 m 少于测试通道数 n 时,增添虚构的 $n-m$ 列,补成方阵,但对应的 $C_{ij}=\infty$,即这些虚构的模块被测试通道测试的效率系数取为 ∞ ,此时同样只是在最后的结果中应去掉虚列的解。

(3)以上针对实际问题不是方阵进行了增补,以便和标准的分派问题统一,但在此文中实际测试时,由于一条通道实际覆盖路径的不同,形成了以上表2的情况,即不仅其不是方阵而且原矩阵某些位置无效率系数,考虑实际问题,此时通道没有覆盖某些模块,其不能进行某些模块的测试,因而可以认为其效率系数为无穷大,这样在运用标准模型求解时,就始终不会取到此通路。

3.3 验证实例

根据表2设 $A_i(i=1, \dots, 6)$ 为需要测试的功能模块, $B_j(j=1, \dots, 8)$ 为8条验证通路,表中 ∞ 均用字母 a 表示,则有:

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
A_1	5	3	2	4	a	a	a	a
A_2	a	a	a	a	3	4	2	5
A_3	7	8	a	a	1	2	a	a
A_4	a	a	5	7	a	a	0	4
A_5	2	a	3	a	7	a	9	a
A_6	a	3	a	8	a	4	a	9



以上式(1)~式(3)为适用标准型用匈牙利法求解过程,但此处又有不同于匈牙利法的地方,有如下两点:

(1)求解过程遵照匈牙利法则,匈牙利法中未出现对无穷大数据的处理,而本例中效率系数出现无穷大数据。针对无穷大数据,其处理方式为其减去任何一个有限的数保持其不变仍为无穷大数据。

(2)对于虚构的行或列在适用法则时不用减去本身,因为减去自身则成为0,这样就会出错。如以上式(2)在进行行减,即将系数矩阵 C 的各行元素依次减去所在行的最小元素时,不适用于最后两行。如果适用则进行列

减,即将系数矩阵 C 的各列元素依次减去所在列的最小元素时,不能得出有效的0元素,因为最后两行虚构行所有元素为0元素。

由式(3)可以看出此式中有有效行中独立零元素的个数不等于矩阵的行数,不满足定理2,而式(3)此时也不适用于匈牙利法的计算步骤,针对此情况解决方案如下:

(1)对本文中构造的矩阵独立零元个数只需等于虚构前行、列中的小者即可得到一个最优方案;

(2)用圈0法找 C' 中独立的零元素;

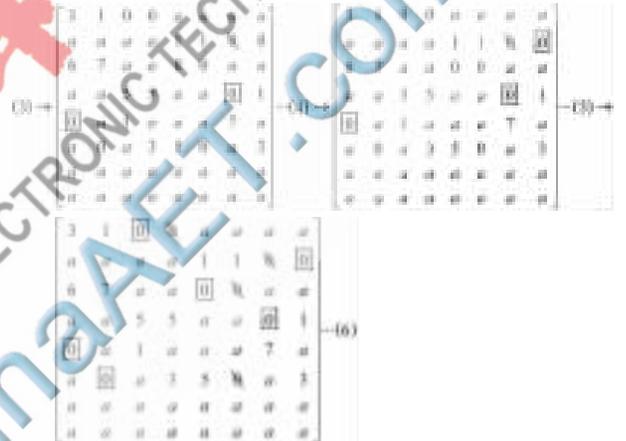
(3)找 C' 中含未加标记的零元素个数最少的行(列),从该行(列)中选效率系数最小的一个零元素加圈;

(4)把该圈0元素所在的行、列中的其他零元素划去。划去的0元素不计入式(1)统计每行、每列的0元素;

(5)反复执行式(1)~式(2)直到所有行、列的零元素全部被加了标志;

(6)去掉虚构的行或列,此时便得到最优解决方案。

具体如下所示:



所以最优的测试效率是: $C_{13}+C_{28}+C_{35}+C_{47}+C_{51}+C_{62}=2+5+1+0+2+3=13$

4 结果分析

以上从系统的角度分析和提出如何最优地针对分级流水单一模块功能的验证方法,如果不进行最优化估算则会浪费大量资源进行无用的重复工作。以下将从验证效率、验证效果两个方面对比随机分配的验证和最优的验证效率之间的差别。验证效果主要从功能覆盖率和代码覆盖率两个方面进行说明。因为本文针对功能验证因而首先保证功能覆盖率,其次代码覆盖率主要包括以下五个方面:

(1)语句覆盖率:指验证程序能覆盖代码的行数与代码的全部行数之比。

(2)路径覆盖率:指验证程序通过的 if...else 或 case 结构的可能路径与设计中的所有可能路径之比。

(3)表达式覆盖率:执行过的 if...else 分支或 case 分支与其所有分支之比。

(4)触发覆盖率:分析敏感变量中的信号是否唯一触发一个过程。

网络与通信 Network and Communication

(5) 自动机覆盖率: 分析仿真用例是否覆盖了所有的状态。

为方便计算, 以下假定系统六大模块语句均等、路径均等、表达式均等、敏感变量均等、自动机状态均等, 系统模块间数据流共有 8 条路径, 只有验证所有路径才能保证以上代码覆盖率的 100% 覆盖。

方案 1: 最优方案保证了验证效率的同时最大地覆盖了验证通路, 假定每条验证通路的验证效率取 3 个模块中最大的一个, 这样在同时覆盖 6 大模块和 8 条路径后的效率为: $C_{13}+C_{28}+C_{35}+C_{47}+C_{51}+C_{62}+l_4+l_6=2+5+1+0+2+3+8+4=25$ 。

在综合考虑验证效率和验证路径时, 对于模块数大于等于路径数的情况依然适用, 但对模块数小于路径数的情况要得到最优的验证方案需对解决方案中用圈 0 法找 C' 中独立的零元素的步骤做一定修正:

找 C' 中含未加标记的零元素个数最少的行(列), 若该行(列)中只有一个独立零元素则直接选择, 有两个及以上独立零元素时, 则将零元素行两两组合, 其中一个取零元素对应的效率系数, 另一个取该零行最大的效率系数(不包括无穷大), 将两者相加, 零元素在其中最小的组合中取得。

方案 2: 根据修正后方法有:

其在覆盖 6 大模块和 8 条路径后的效率为:

$$C_{14}+C_{28}+C_{35}+C_{47}+C_{51}+C_{62}+l_3+l_6=4+5+1+0+2+3+5+4=24$$

方案 3: 随机分配:

$$C_{11}+C_{25}+C_{32}+C_{43}+C_{57}+C_{64}+l_6+l_8=5+3+8+5+9+8+4+9=51$$

以上三种方案效率对照如表 3 所示。

从表 3 明显可以看出修正后的效率较之未作修正和随机方案有明显的提高和改善。

本文从网络数据交换芯片的系统级分析了分级

流水式芯片功能测试, 建立了系统级模型, 针对模型提出验证需求, 在满足需求的基础上提出了验证策略, 结合实例论证了验证方法的正确性及优势。其结果证明有效的分配策略对整个定向功能验证的效率影响很大。本文只是建立了一般的简化的系统模型, 但验证策略和方法却具有普遍适用性, 针对一般的系统均可建立此种模型, 通过预算提出最优的验证策略。

参考文献

- [1] 谢希仁. 计算机网络(第 4 版)[M]. 北京: 电子工业出版社, 2003: 2-8.
- [2] 杜慧敏, 李宥谋, 赵全良. 基于 Verilog 的 FPGA 设计基础[M]. 西安: 西安电子科技大学出版社, 2006: 134-136.
- [3] 任宇, 王以伍. VLSI 设计中一种新型的功能验证方法[J]. 微计算机信息, 2006, 12(2): 285-287.
- [4] 冯晓慧, 刘三阳. 工程设计中的最优化计算方法[M]. 西安: 西安电子科技大学出版社, 2006: 219-222.
- [5] 张良, 陈全润, 张明晖, 等. “一人多事”的分派问题[R]. 山东: 教育部精品课山东大学运筹, 2002: 8-9.

(收稿日期: 2011-06-24)

作者简介:

杨飞, 男, 1986 年生, 硕士生, 主要研究方向: 数字集成电路系统设计。

蒋林, 男, 1970 年生, 教授, 硕士生导师, 主要研究方向: 专用集成电路设计。