



## (12) 发明专利申请

(10) 申请公布号 CN 101821737 A

(43) 申请公布日 2010.09.01

(21) 申请号 200880100011.X

代理人 周建秋 王凤桐

(22) 申请日 2008.07.23

(51) Int. Cl.

(30) 优先权数据

G06F 17/50 (2006.01)

60/951,436 2007.07.23 US

12/177,867 2008.07.22 US

(85) PCT申请进入国家阶段日

2010.01.22

(86) PCT申请的申请数据

PCT/US2008/008998 2008.07.23

(87) PCT申请的公布数据

W02009/014731 EN 2009.01.29

(71) 申请人 新思公司

地址 美国加利福尼亚州

(72) 发明人 K·S·麦克尔文 B·勒莫尼耶

B·哈尔平

(74) 专利代理机构 北京润平知识产权代理有限公司

公司 11283

权利要求书 7 页 说明书 18 页 附图 11 页

(54) 发明名称

构架物理综合

(57) 摘要

本发明公开了用于设计集成电路的方法和设备。根据一方面,本发明的电路设计公开了综合与布局的迭代过程,其中每一次迭代均提供对集成电路的设计的增量式改变。之后,利用来自布局的精确时序信息进行综合转换,且所述过程是朝着设计的最终时序闭合来进行增量式迭代的。本发明的增量式迭代方法可通过综合到布线以及布线到综合来提供连续进步,具有在知晓当前实例布局的情况下进行综合的增量式改进,以及在知晓当前电路逻辑的情况下进行布局的增量式改进。

1. 一种设计集成电路的方法,该方法包括:  
在所述集成电路的高层设计表示上迭代且增量式地改变设计的状态。
2. 根据权利要求1所述的方法,其中迭代过程朝着以下各项中的至少一者推进:  
增量式地降低所述集成电路中器件表示的抽象化水平;  
使RTL网表增量式地成熟化;以及  
使布局配置增量式地成熟化。
3. 根据权利要求1所述的方法,其中所述高层设计表示包括至少一个组件,该组件不具有为该组件选择的芯片本原中的实施形式。
4. 根据权利要求1所述的方法,其中迭代过程朝着以下各项中的至少一者推进:期望时序、每层的拥塞的减少、区域利用率的改善以及功耗的改善。
5. 根据权利要求1所述的方法,其中所述集成电路的设计的状态包括具有以下各项中至少一者的网表:时序数据、资源信息、布局信息、布线信息以及功率数据。
6. 根据权利要求1所述的方法,其中改变过程为综合转换和布局转换中的至少一者。
7. 一种包含多个可执行指令的机器可读介质,当所述可执行指令在数字处理系统上被执行时,促使所述数字处理系统执行设计集成电路(IC)的方法,该方法包括:  
在所述集成电路的高层设计表示上迭代且增量式地改变设计的状态。
8. 根据权利要求7所述的介质,其中迭代过程朝着以下各项中的至少一者推进:  
增量式地降低所述集成电路中器件表示的抽象化水平;  
使RTL网表增量式地成熟化;以及  
使布局配置增量式地成熟化。
9. 根据权利要求7所述的介质,其中所述高层设计表示包括至少一个组件,该组件不具有为该组件选择的芯片本原中的实施形式。
10. 根据权利要求7所述的介质,其中迭代过程朝向以下各项中的至少一者推进:期望时序、每层的拥塞的减少、区域利用率的改善以及功耗的改善。
11. 根据权利要求7所述的介质,其中所述集成电路的设计的状态包括具有以下各项中至少一者的网表:时序数据、资源信息、布局信息、布线信息以及功率数据。
12. 根据权利要求7所述的介质,其中改变过程为综合转换和布局转换中的至少一者。
13. 一种设计集成电路的方法,该方法包括:  
迭代地选择并执行综合转换和布局转换中的至少一者,以在所述集成电路的高层设计表示上增量式地改变设计的状态。
14. 根据权利要求13所述的方法,其中所述高层设计表示包括至少一个组件,该组件不具有为该组件选择的芯片本原中的实施形式。
15. 根据权利要求13所述的方法,其中迭代过程朝着以下各项中的至少一者推进:  
增量式地降低所述集成电路中器件表示的抽象化水平;  
使RTL网表增量式地成熟化;以及  
使布局配置增量式地成熟化。
16. 根据权利要求15所述的方法,其中如果存在以下情况中的至少一者,则网表是更加成熟的:  
所述网表由较低层RTL实例以及更多门级实例构成;以及

所述网表具有较低的抽象化水平。

17. 根据权利要求 15 所述的方法,其中如果布局配置满足更多对所述集成电路的资源使用进行管理的规则,则该布局配置是更加成熟的。

18. 根据权利要求 13 所述的方法,其中迭代过程朝着以下各项中的至少一者推进:期望时序、每层的拥塞的减少、区域利用率的改善以及功耗的改善。

19. 根据权利要求 13 所述的方法,其中所述集成电路的设计的状态包括具有以下各项中至少一者的网表:时序数据、资源信息、布局信息、布线信息以及功率数据。

20. 一种包含多个可执行指令的机器可读介质,当所述可执行指令在数字处理系统上被执行时,促使所述数字处理系统执行设计集成电路(IC)的方法,该方法包括:

迭代地选择并执行综合转换和布局转换中的至少一者,以在所述集成电路的高层设计表示上增量式地改变设计的状态。

21. 根据权利要求 20 所述的介质,其中所述高层设计表示包括至少一个组件,该组件不具有为该组件选择的芯片本原中的实施形式。

22. 根据权利要求 20 所述的介质,其中迭代过程朝着以下各项中的至少一者推进:

增量式地降低所述集成电路中器件表示的抽象化水平;

使 RTL 网表增量式地成熟化;以及

使布局配置增量式地成熟化。

23. 根据权利要求 22 所述的介质,其中如果存在以下情况中的至少一者,则网表是更加成熟的:

所述网表由较低层 RTL 实例以及更多门级实例构成;以及

所述网表具有较低的抽象化水平。

24. 根据权利要求 22 所述的介质,其中如果布局配置满足更多对所述集成电路的资源使用进行管理的规则,则该布局配置是更加成熟的。

25. 根据权利要求 20 所述的介质,其中迭代过程朝着以下各项中的至少一者推进:期望时序、每层的拥塞的减少、区域利用率的改善以及功耗的改善。

26. 根据权利要求 20 所述的介质,其中所述集成电路的设计的状态包括具有以下各项中至少一者的网表:时序数据、资源信息、布局信息、布线信息以及功率数据。

27. 一种设计集成电路的方法,该方法包括:

生成所述集成电路的设计的初始状态;

针对多个潜在转换,迭代地评估价值函数;以及

执行提供最佳价值函数的转换;

其中每一次迭代在所述集成电路的高层设计表示上增量式地改变所述设计的状态。

28. 根据权利要求 27 所述的方法,其中所述高层设计表示包括至少一个组件,该组件不具有为该组件选择的芯片本原中的实施形式。

29. 根据权利要求 27 所述的方法,其中迭代过程朝着以下各项中的至少一者推进:

增量式地降低所述集成电路中器件表示的抽象化水平;

使 RTL 网表增量式地成熟化;以及

使布局配置增量式地成熟化。

30. 根据权利要求 29 所述的方法,其中降低抽象化水平和所述网表的成熟化中的至少

一者包括综合转换。

31. 根据权利要求 29 所述的方法,其中布局配置的成熟化包括对布局转换和综合优化中的至少一者的第二次迭代。

32. 根据权利要求 29 所述的方法,其中如果存在以下情况中的至少一者,则网表是更加成熟的:

所述网表由较低层 RTL 实例以及更多门级实例构成;以及  
所述网表具有较低的抽象化水平。

33. 根据权利要求 29 所述的方法,其中如果布局配置满足更多对所述集成电路的资源使用进行管理的规则,则该布局配置是更加成熟的。

34. 根据权利要求 29 所述的方法,其中全局布局被用于对成熟度较低的网表进行布局,并用于布局配置。

35. 根据权利要求 34 所述的方法,其中所述全局布局为力导向布局方法。

36. 根据权利要求 27 所述的方法,其中迭代过程朝着以下各项中的至少一者推进:期望时序、每层的拥塞的减少、区域利用率的改善以及功耗的改善。

37. 根据权利要求 27 所述的方法,其中所述价值函数包括以下各项中的至少一者:时序、每层的拥塞、区域利用率以及功率。

38. 根据权利要求 27 所述的方法,其中所述价值函数基于所述集成电路的设计的状态。

39. 根据权利要求 27 所述的方法,其中所述价值函数基于收敛性判别标准。

40. 根据权利要求 27 所述的方法,其中所述集成电路的设计的状态包括具有以下各项中至少一者的网表:时序数据、资源信息、布局信息、布线信息以及功率数据。

41. 根据权利要求 40 所述的方法,其中基于时序由多个中性优化来对所述网表进行优化。

42. 根据权利要求 41 所述的方法,其中所述中性优化包括以下各项中的至少一者:可逆的区域恢复、加法树分解、明显的资源分配、网表平化以及电子复用器分解。

43. 根据权利要求 40 所述的方法,该方法还包括针对粗略的全局布局来估算所述网表中每个元件的形状和资源。

44. 根据权利要求 27 所述的方法,其中所述潜在转换包括以下各项中的至少一者:撤销/进行资源共享、加法树分解、与/或门分解、逻辑复制、香农扩展、比特拼接、复用器/可编程复用器合并及时序驱动分解、绕行移除、对离散资源的分配、因素分解以及布局。

45. 根据权利要求 27 所述的方法,其中所述潜在转换包括以下各项中至少一者:布局、更新的布局、布局优化、资源分配、资源利用、逻辑优化、逻辑分解、逻辑映射以及逻辑综合。

46. 根据权利要求 45 所述的方法,其中所述资源利用过程包括基于质量度量进行等级评定并移除低等级的资源。

47. 根据权利要求 45 所述的方法,其中所述质量度量包括距资源使用的距离。

48. 根据权利要求 45 所述的方法,其中所述资源利用过程包括计算每一个电路实例上的力,该力为根据多个范围所计算的力的加权和。

49. 根据权利要求 48 所述的方法,其中短程力的权重与所述实例的密度成正比,且远程力的权重与所述实例的密度成反比。

50. 根据权利要求 45 所述的方法,其中所述资源分配包括混杂资源分配。

51. 根据权利要求 45 所述的方法,其中所述混杂资源分配包括将具有混杂资源的非本原实例分配给所有层,以使得该实例具有资源,且在该实例上的力为来自所有层的力的加权和。

52. 根据权利要求 45 所述的方法,其中所述混杂资源分配包括将具有混杂资源的非本原实例分配给所有层,以使得该实例具有资源,且在该实例上的力为来自所有层的力之中具有最小幅值的力。

53. 根据权利要求 45 所述的方法,其中所述混杂资源分配包括将具有混杂资源的非本原实例的资源分配给在不考虑所述实例的资源布局的情况下当所述实例被布置在该层上时提供最低破坏性的层。

54. 根据权利要求 27 所述的方法,其中所述潜在转换包括以下各项中至少一者:综合映射、综合优化、用于降低所述集成电路的设计中器件表示的抽象化水平并增加 RTL 网表的成熟化水平的综合。

55. 根据权利要求 27 所述的方法,其中所述潜在转换还包括第二次迭代过程。

56. 一种包含多个可执行指令的机器可读介质,当所述可执行指令在数字处理系统上被执行时,促使所述数字处理系统执行设计集成电路(IC)的方法,该方法包括:

生成所述集成电路的设计的初始状态;

针对多个潜在转换,迭代地评估价值函数;以及

执行提供最佳价值函数的转换;

其中每一次迭代在所述集成电路的高层设计表示上增量式地改变所述设计的状态。

57. 根据权利要求 56 所述的介质,其中所述高层设计表示包括至少一个组件,该组件不具有为该组件选择的芯片本原中的实施形式。

58. 根据权利要求 56 所述的介质,其中迭代过程朝着以下各项中的至少一者推进:

增量式地降低所述集成电路中器件表示的抽象化水平;

使 RTL 网表增量式地成熟化;以及

使布局配置增量式地成熟化。

59. 根据权利要求 58 所述的介质,其中降低抽象化水平和所述网表成熟化中的至少一者包括综合转换。

60. 根据权利要求 58 所述的介质,其中布局配置的成熟化包括对布局转换和综合优化中的至少一者的第二次迭代。

61. 根据权利要求 58 所述的介质,其中如果存在以下情况中的至少一者,则网表是更加成熟的:

所述网表由较低层 RTL 实例以及更多门级实例构成;以及

所述网表具有较低的抽象化水平。

62. 根据权利要求 58 所述的介质,其中如果布局配置满足更多对所述集成电路的资源使用进行管理的规则,则该布局配置是更加成熟的。

63. 根据权利要求 58 所述的介质,其中全局布局被用于对成熟度较低的网表进行布局,并用于布局配置。

64. 根据权利要求 63 所述的介质,其中所述全局布局为力导向布局方法。

65. 根据权利要求 56 所述的介质,其中迭代过程朝着以下各项中的至少一者推进:期望时序、每层的拥塞的减少、区域利用率的改善以及功耗的改善。

66. 根据权利要求 56 所述的介质,其中所述价值函数包括以下各项中的至少一者:时序、每层的拥塞、区域利用率以及功率。

67. 根据权利要求 56 所述的介质,其中所述价值函数基于所述集成电路的设计的状态。

68. 根据权利要求 56 所述的介质,其中所述价值函数基于收敛性判别标准。

69. 根据权利要求 56 所述的介质,其中所述集成电路的设计的状态包括具有以下各项中至少一者的网表:时序数据、资源信息、布局信息、布线信息以及功率数据。

70. 根据权利要求 69 所述的介质,其中基于时序由多个中性优化来对所述网表进行优化。

71. 根据权利要求 70 所述的介质,其中所述中性优化包括以下各项中的至少一者:可逆的区域恢复、加法树分解、明显的资源分配、网表平化以及电子复用器分解。

72. 根据权利要求 69 所述的介质,还包括针对粗略的全局布局,估算所述网表中每个元件的形状和资源。

73. 根据权利要求 56 所述的介质,其中所述潜在转换包括以下各项中的至少一者:撤销/进行资源共享、加法树分解、与/或门分解、逻辑复制、香农扩展、比特拼接、复用器/可编程复用器合并及时序驱动分解、绕行移除、对离散资源的分配、因素分解以及布局。

74. 根据权利要求 56 所述的介质,其中所述潜在转换包括以下各项中至少一者:布局、更新的布局、布局优化、资源分配、资源利用、逻辑优化、逻辑分解、逻辑映射以及逻辑综合。

75. 根据权利要求 74 所述的介质,其中所述资源利用过程包括基于质量度量进行等级评定,并移除低等级的资源。

76. 根据权利要求 74 所述的介质,其中所述质量度量包括距资源使用的距离。

77. 根据权利要求 74 所述的介质,其中所述资源利用过程包括计算每个电路实例上的力,该力为根据多个范围所计算的力的加权和。

78. 根据权利要求 77 所述的介质,其中短程力的权重与所述实例的密度成正比,且远程力的权重与所述实例的密度成反比。

79. 根据权利要求 74 所述的介质,其中所述资源分配包括混杂资源分配。

80. 根据权利要求 74 所述的介质,其中所述混杂资源分配包括将具有混杂资源的非本原实例分配给所有层,以使得该实例具有资源,且该实体上的力为来自所有层的力的加权和。

81. 根据权利要求 74 所述的介质,其中所述混杂资源分配包括将具有混杂资源的非本原实例分配给所有层,以使得该实例具有资源,且该实体上的力为来自所有层的力中具有最小幅值的力。

82. 根据权利要求 74 所述的介质,其中所述混杂资源分配包括将具有混杂资源的非本原实例分配给在不考虑所述实例的资源布局的情况下当所述实例被布置在层上时提供最低破坏性的层。

83. 根据权利要求 56 所述的介质,其中所述潜在转换包括以下各项中至少一者:综合映射、综合优化、用于降低所述集成电路的设计中器件表示的抽象化水平并增加 RTL 网表

的成熟化水平的综合。

84. 根据权利要求 56 所述的介质,其中所述潜在转换还包括第二次迭代过程。

85. 一种设计集成电路的方法,该方法包括:

在高层电路表示上确定设计的综合转换;以及

在所述高层电路表示的综合仍待完成之前,确定布局决策。

86. 根据权利要求 85 所述的方法,其中所述高层电路表示为包括至少一个组件的表示,所述组件不具有为该组件选择的芯片本原中的实施形式。

87. 根据权利要求 85 所述的方法,该方法还包括:

对之前所综合的高层电路表示的一部分重复执行综合转换。

88. 根据权利要求 85 所述的方法,其中所述综合转换为以下各项中的一者:

加法树分解;

与/或门分解;

网表平化;

电子复用器分解;

逻辑优化;

逻辑分解;

撤销/进行资源共享;

逻辑复制;

逻辑因素分解;

香农扩展;

比特拼接;

复用器/可编程复用器合并及时序驱动分解;以及

绕行移除。

89. 根据权利要求 85 所述的方法,其中所述综合转换对根据所述高层电路表示所编译的 RTL 组件进行汇编。

90. 根据权利要求 85 所述的方法,其中所述布局决策是增量式的。

91. 根据权利要求 85 所述的方法,其中所述布局决策是针对所述电路设计的一部分的少量对象的部分布局。

92. 一种设计集成电路(IC)的方法,该方法包括:

确定不完整的布局信息;以及

通过使用所述不完整的布局信息,在高层电路表示上确定设计的综合转换,

其中所述不完整的布局信息包括与所述 IC 的一部分相关的布局决定。

93. 根据权利要求 92 所述的方法,其中所述高层电路表示为包括至少一个组件的表示,所述组件不具有为该组件选择的芯片本原中的实施形式。

94. 根据权利要求 92 所述的方法,该方法还包括:

对之前所综合的高层电路表示的一部分重复执行综合转换。

95. 根据权利要求 92 所述的方法,该方法还包括:

在所述高层电路表示的综合仍待完成之前,确定布局决定。

96. 根据权利要求 92 所述的方法,其中所述综合转换为以下各项中的一者:

加法树分解；  
与 / 或门分解；  
网表平化；  
电子复用器分解；  
逻辑优化；  
逻辑分解；  
撤销 / 进行资源共享；  
逻辑复制；  
逻辑因素分解；  
香农扩展；  
比特拼接；  
复用器 / 可编程复用器合并及时序驱动分解；以及  
绕行移除。

97. 根据权利要求 92 所述的方法,其中所述综合转换对根据所述高层电路表示所编译的 RTL 组件进行汇编。

98. 根据权利要求 92 所述的方法,其中所述综合转换是增量式的。

99. 根据权利要求 92 所述的方法,其中所述综合转换是针对所述电路设计的一部分的少量对象的部分布局。

## 构架物理综合

[0001] 相关申请

[0002] 本申请要求于 2007 年 7 月 23 日提交的美国临时申请 No. 60/951,436 (档案号: 02986.P059Z) 的优先权, 该临时申请作为参考而被结合于此。本申请还与于 2008 年 7 月 22 日提交的名称为“Architectural Physical Synthesis”的申请 no. 12/177,869 (档案号: 02986.P1117) 相关, 并与该申请于同一日提交。

### 技术领域

[0003] 本发明主要涉及集成电路设计领域, 并且尤其涉及通过高层描述 (highlevel description) 的综合 (synthesis) 过程来进行集成电路设计。

### 背景技术

[0004] 对于 VLSI (超大规模集成) 级的数字电路设计工艺而言, 设计者常常采用计算机辅助技术。已开发了诸如硬件描述语言 (HDL) 的标准语言来描述数字电路, 以辅助复杂数字电路的设计和模拟。诸如 VHDL 和 Verilog 的多种硬件描述语言已逐渐成为行业标准。VHDL 和 Verilog 是通用硬件描述语言, 其允许使用抽象数据类型来定义芯片原始级、寄存器传输级 (RTL) 或行为级的硬件模型。随着器件工艺继续进步, 已开发了各种产品设计工具以使 HDL 适合用于新的器件和设计风格。

[0005] 在利用 HDL 代码设计集成电路时, 首先编写代码, 并由 HDL 编译器对所编写的代码进行编译。HDL 源代码在某一层描述电路元件, 且编译程序通过所述编译而产生 RTL 网表 (nestlist)。RTL 网表由多个 RTL 对象或组件以及多个网络 (net) (该网络为所述组件之间的信号连接) 构成。所述 RTL 网表通常是与技术无关的网表, 因为其与特定厂商的集成电路 (例如, 现场可编程门阵列 (FPGA) 或专用集成电路 (ASIC)) 的技术或架构无关。所述 RTL 网表对应于电路元件的示意性表示 (与行为表示相比)。然后执行映射操作, 以从与技术无关的 RTL 网表转换到可以用来创建厂商技术或架构中的电路的技术专用网表, 所述映射操作包括对实例 (instance) 进行布局以及对互连 (interconnection) 进行布线, 以使得电路满足给定的时序、间隔和功率约束。

[0006] 如图 1 所示, 早期的电子设计自动化 (EDA) 是完全将 HDL 综合与布局 / 布线过程相分离的。在操作 11, 准备 HDL 代码。在操作 13, 对操作 11 中所准备的 HDL 进行编译并综合, 以产生网表, 通常通过执行逻辑优化对该网表进行优化。在此之后, 映射过程将所述网表映射至特定目标技术 / 架构。在操作 13 结束之后, 便已完成了综合, 且现在可提供专用于厂商的 IC 中使用的技术 / 架构的网表。该网表有效地位于门级, 且基于预布局信息 (例如, 扇出计数、或所连接的组件类型及尺寸) 通过使用互连性质统计模型来对时序分析进行估算。在综合之后, 可在操作 15 对逻辑电路执行传统的布局操作, 并在操作 17 对网表进行局部改变 (仅在芯片原始级、单元级或门级上), 以满足时序性能。之后, 在操作 19 执行传统布线操作, 以创建每一 IC 中的电路设计。如果存在一些未满足的约束, 则所述过程通过回路迭代 (iteration) 进行修改。

[0007] 以前,当在早期综合工具中实例延迟较为突出时,基于统计模型的时序估算是足够精确的,从而综合与布局的分离需要相对较少次迭代来返回至 HDL 和综合阶段。

[0008] 然而,利用收缩技术节点,互连延迟变得非常显著,超过了门延迟。这导致综合操作中的延迟估算与布局和布线操作之后的实际延迟之间的关联越来越小,致使后综合(post-synthesis)与后布置(post-layout)结果之间缺乏时序预测能力。因此,在很多情况下,在布局和布线过程之后,电路物理布置无法满足电路设计标准,且通常设计必须从综合步骤重新开始,并重复综合/布局/布线过程。

[0009] 为对综合进行改善,在综合过程期间考虑与设计相关的物理特性(例如,布局)是非常重要的。现已采用了一系列技术来将布局信息引入综合过程,例如元件平面布置、区域内最佳化(IPO)以及物理综合。

[0010] 在元件平面布置(floorplanning)技术中,设计被划分成芯片上的多个区,并在通过使用统计模型来估算区内的互连的同时,针对区间互连使用基于布局的互连估算。可在 RTL 阶段早期或初始综合运行之后使用元件平面布置。元件平面布置可被扩展为将 RTL 组件划分、复制以及切割成多个区,并与 RTL 级时序和区域模型相组合。之后,可使用来自区间时序的改善后的时序来更精确地驱动 RTL 级优化。手动产生高质量的元件平面布置是非常困难的,且要求用户拥有熟练的技术。自动元件平面布置器(例如,太拉系统(Tera System)(美国专利 6,145,117 和 6,360,356)中的元件平面布置器)可创建多个区,并将 RTL 组件分配给多个区。因为综合被去耦合(decouple),且在自动元件平面布置之后,因此在元件平面布置期间,时序和区域信息的准确度很低。

[0011] 一种称为区域内最佳化(IPO)的技术提供了从布局和布线延迟至综合域的逆向注解。仅因为细节布局未被更新,关键路径会被重新优化,经修改的网络的互连延迟会返回到统计模型。如果做出了许多改变,则后续的最终网表的合法化(legalization)可能需要将实例移动至远离他们初始位置的位置,导致巨大的延迟估算误差。基于此原因,在需要显著改变以实现时序闭合(enclosure)时,IPO 被视为是不稳定的。

[0012] 另一技术是物理综合,该技术是对 IPO 技术的改进,其中对被映射网表的少量优化与增量式(incremental)重新合法化相交织,以保持延迟和资源度量的精确性。该技术的限制在于,个别改变受限于资源的适度增加,或者 IPO 技术的不稳定问题可能会再次出现。目前存在多种用于物理综合的算法。图 2 显示了一种通过使用基于被布局实例邻近性(proximity)的时序估算而提供物理综合引擎的算法。在操作 23 中对被映射网表初始布局之后,在操作 24(仅在芯片原始级执行)中物理综合操作选择电路中用于增量式优化及重新布局的部分。

[0013] 从上述描述可知,亟需对电子设计自动化进行算法改进。

[0014] 现有专利还涉及或描述了芯片综合,这些专利包括:美国专利 6,519,754、6,711,729、7,010,769、6,145,117 以及 6,360,356。论文(Bo Hu, Timing-Driven Placement for Heterogeneous Field Programmable Gate Array, IEEE/ACM International Conference on Computer-Aided Design, 2006 年 11 月 (ICCAD' 06), 383-388 页 (ISSN: 1092-3152; ISBN 1-59593-389-1)) 中最近描述了一种布局算法。

## 发明内容

[0015] 本发明公开了用于设计集成电路的方法和设备。在示例性实施方式中,本发明的电路设计公开了综合与布局的迭代过程,该过程起始于 RTL 或行为级,其中每一次迭代均通过集成电路设计的转换而提供增量式改变。在某些方面,所述转换可以是综合或布局转换。综合转换对网表中的对象和 / 或形成该对象之间的连接的网络进行修改。布局转换对网表中一个或多个对象的位置进行修改。在本发明至少某些实施方式中的增量式迭代方法可通过使用由设计度量 (design metric) (例如,当前电路网表、布局、时序、资源可用性以及功率) 所确定的适当的综合与布局转换,来提供不断的改进。在某些方面,在每次转换之后,对受到影响的设计度量进行更新,以使得将来的转换决策可基于精确的设计统计。所述过程朝向设计的最终时序资源和功率闭合 (enclosure) 而增量式迭代。

[0016] 本发明至少某些实施方式的关键方面在于,在识别高层 (high level) 组件的特定资源类型之前进行布局。例如,对具有期望权重以及相关的组件资源总量的可选实施形式进行分类,且布局器对布局进行演变,以移动组件靠近期望实施形式的资源类型。

[0017] 在优选实施方式中,本发明起始于表示 RTL 或行为设计 (电路) 的图以及芯片资源物理图。执行迭代转换,其中每次转换可对电路或电路中对象的布局进行优化或完善。

[0018] 在一种实施方式中,转换由高层优化构成。该转换通过规则或数学转换而将组件或多个组件优化成功能等效且具有较优特性 (例如,时序、功率或资源消耗) 的可选的一组组件。此类转换的一示例为重新组织算术表达式,以降低树高从而改善延迟。另一示例为资源共享或不共享。

[0019] 在另一实施方式中,高层优化转换将一组或多组电路对象从抽象的形式完善为更为具体的形式。完善转换的一示例为将算术表达式映射至芯片上的 DSP 资源。当抽象形式被完善之后,通常存在多个实施形式选择。例如,算数表达式可通过芯片上的专用算数功能 (DSP 块)、存储器内的表查找来实施,或可用芯片上的较低层的逻辑组件 (LUT 或门和触发器) 来构建。行为综合流程中的组件可具有基于可选规划和资源共享所记录的多个实施形式。行为组件的此类可选形式还可基于当前可用资源以及互连延迟来动态生成。

[0020] 在另一实施方式中,所述完善转换还具有基于可选实施形式的质量的紧急性度量,且可根据紧急性而被选择。可根据设计目标 (例如,区域消耗、功耗或时序) 来测量实施形式的质量。还可包括其他更为深奥的目标,例如单一事件扰乱难度 (single event upset hardness)。例如,如果设计包含一个大的存储器以及若干个小存储器,且大存储器在被逻辑结构实施时具有相对较差的实施质量,则相比于将设计中的中型存储器与芯片上稀有的专用存储资源相关联,将大存储器与芯片上稀有的专用存储资源相关联则相对更为重要一些。因此,大存储器的紧急性度量远高于小存储器的度量。一旦组件被映射至特定实施形式并与芯片上的特定资源相关联,则到这些组件的连接便用作电路剩余部分的布局的锚 (anchor),从而改善了时序质量以及可用资源的估算。

[0021] 在一种实施方式中,所述布局转换可以是一个或多个可布局对象的位置的完善,以改善布局度量,例如,实例拥塞、可布线性 (routability) 以及电路性能。可布局对象可由行为综合组件、未被映射的逻辑的 RTL 块、被映射的逻辑、或这些项的任意组合构成。

[0022] 在一实施方式中,所述布局转换能够修改不同抽象化水平的对象。例如,一些可布局对象可以是 RTL 块,而其他可布局对象可以是被映射的门。

[0023] 在另一实施方式中,可在所述布局被演化得足够局部化以致于可确定可用资源并估算布线延迟时,触发完善转换。

[0024] 根据本发明另一方面,用于设计集成电路的示例性方法提供一种增量式转换迭代,其中综合与布局转换并没有一定的顺序,仅根据他们的功能而被选择。电路设计自动化基于选择函数来选择下一转换(综合或布线)。在每一次迭代时,计算预定的转换列表的成本。该成本可包括预测(look ahead)其他转换中的成本变化。例如,如果将算术运算映射至 ROM,则会移除用于其它运算的实施形式的 ROM 选项,从而增加了该其他运算的成本。可基于成本收敛标准(例如当前布局、网表、资源、时序或功率)来选择最佳转换。

[0025] 下一转换可以是布局更新、资源分配、综合优化、布局优化或布线更新。因此,IC 设计的状态可增量式地朝向最终的电路规格和布置(layout)推进。

[0026] 在另一实施方式中,迭代地执行所述布局转换,直至关键路径开始成形,或直至根据预定的拥塞阈值资源足够分散。迭代性能的标准为时序、每资源层(per resource layer)的拥塞、区域利用率以及功率。

[0027] 所述每资源层的拥塞可通过使用资源层来确定。针对芯片上每一不同原始类型的资源,存在一资源层。例如,如今的 FPGA 和结构化 ASIC 均引入了原始芯片资源的不规则布置。这些原始类型包括逻辑(LUTs)、触发器、特定 I/O 单元(例如用于高速串行互连的串化器/解串化器(SERDES))、各种具有不同容量的存储组件以及用于加速 DSP 算法的高速算术块。除逻辑和触发器之外,通常,这些资源均以稀疏且可能不规则的形式被包括。许多 FPGA 具有有限数量的 RAM、DSP 以及以稀疏的列布置在芯片上的其他专用逻辑块。例如,仅在芯片布局中的两列有 DSP 算术块。资源层是针对每一原始类型所创建的分布图,并记录了针对所述类型的可用资源位置以及所述类型的每一本原(primitive)的布局。如果局部物理区供不应求,则称该层存在拥塞。

[0028] 在所述方法的典型示例中,用具有时序约束以及布局约束(例如 IO 管脚、现有的元件平面布置或现有的布局)的高层表示来生成所述集成电路的设计的初始状态。该高层表示可为硬件描述语言(HDL)代码或对硬件描述语言(HDL)代码进行编译之后的与技术无关的 RTL 网表。

[0029] 在一实施方式中,首先基于时序由一系列中性优化(neutral optimization)来对所述集成电路设计的初始状态的网表进行优化。所述中性优化可以是很容易撤销的任何区域恢复,例如资源共享或不共享;优选基于扇出(fanout)表时序的加法树分解;资源分配;有助于跨层级优化的网表的平化(flattening);复用器提取或重构。

[0030] 在一实施方式中,集成电路设计的基本流程从 RTL 网表发展至分解、因素分解(factorization)、以及被映射及布线之后的网表。在该流程中,可执行布局修改、资源分配以及区域或时序优化。

[0031] 在一实施方式中,可重复对布局和电路架构进行完善的过程,直至所有高层组件均已被给定一个特定的实施形式且资源分配及布局已在芯片上展开以使得每个组件附近均具有足够的实施资源。基于此观点,可使用更多的传统物理综合流程来完成整个实施。

[0032] 在另一实施方式中,记录所应用的转换及其潜在的可选转换。可重复所述流程,并可应用可选转换来实现更佳的效果。

[0033] 本发明还公开了包括软件媒体的设备,所述软件媒体可用于设计集成电路。例如,

本发明包括数字处理系统,该系统能够根据本发明来设计集成电路,而且本发明还提供机器可读媒体,当该机器可读媒体在数字处理系统(例如计算机系统)上执行时,可促使该数字处理器系统执行用于设计集成电路的方法。

[0034] 通过附图以及以下详细描述,本发明的其他特征将显而易见。

## 附图说明

[0035] 附图中的图示以示例的方式而非限定性地示出了本发明,其中相同的参考标记表示相类似的元件。

[0036] 图 1 显示了现有技术的用于设计集成电路的方法;

[0037] 图 2 显示了现有技术的物理综合的示例性方法;

[0038] 图 3 显示了根据本发明一种实施方式的设计集成电路的方法的流程图;

[0039] 图 4 显示了根据本发明一种实施方式的设计集成电路的另一方法的流程图;

[0040] 图 5A 和图 5B 显示了根据本发明某些实施方式的设计集成电路的方法的细节;

[0041] 图 6 显示了根据本发明一种实施方式的设计集成电路的方法的流程图;

[0042] 图 7 显示了对形状和资源进行的示例性估算;

[0043] 图 8 显示了针对资源类型的示例性映射;

[0044] 图 9A 和图 9B 为存储资源的示例性映射;

[0045] 图 10A 和图 10B 为示例性的资源共享实施形式;

[0046] 图 11 显示了加法树分解的示例;

[0047] 图 12 显示了门级树 (gate tree) 分解的示例;

[0048] 图 13A 和图 13B 显示了切片优化的示例;

[0049] 图 14 显示了复制优化的示例;

[0050] 图 15 显示了香农扩展的示例;

[0051] 图 16A 和图 16B 显示了复用器 / 可编程复用器合并 (collapsing) 及时序驱动分解的示例;

[0052] 图 17 显示了可与本发明一起使用的数据处理系统的框图示例。

## 具体实施方式

[0053] 在此描述了用于设计集成电路或多个集成电路的方法和设备。在以下描述中,出于说明的目的,列出了大量特定细节,以提供对本发明的透彻理解。然而,对于本领域技术人员而言很明显的是,本发明可在缺少这些特定细节的情况下实施。在其他实例中,公知的结构、过程和装置以框图形式被显示、或简要地引用,以使得解释不至过于繁琐。

[0054] 本发明公开了设计集成电路的方法和设备,在一种实施方式中,所方法和设备将布局 and 综合整合到单个通道 (pass) 中。本发明的实施方式公开了一种物理综合过程(称为架构物理综合),其中综合与布局之间的相互作用出现在架构层。这使得可利用集成电路基板表示上的实际物理布局进行综合,可为综合提供可用的局部资源以及与布局的实际电路时序密切相关的延迟估算,从而可同时考虑综合与布局之间的相互影响。此外,这可通过考虑布局、拥塞估算以及目标芯片架构的特性(该特性包括但不限于不同资源的物理分布、组件延迟以及互连延迟)的方式来提供作出高层架构决策、映射高层组件或进行高层

电路转换的自动化方法。根据本发明的一方面,在给定电路设计或 HDL 代码表示的情况下,存在大量对综合与布局进行连结 (interlink) 的可选实施形式,尤其是对于具有给定分布资源的现有元件平面布置。为获得最佳设计实施形式,能够基于当前通过布局收集的可用电路数据 (例如,时序或功率) 来回溯跟踪早期的综合决策是很重要的。

[0055] 因此,在本发明的一方面,在综合周期的早期 (例如,在电路架构层、高层设计或行为表示中) 执行布局,以允许精确评估各种设计实施形式的适用性。这对于并未在芯片上均匀分布资源的预扩散芯片 (例如, FPGA 和结构化 ASIC) 而言是极其重要的。在预扩散芯片中,资源的位置和资源类型被预先定义,且以稀疏形式被分布。例如,如今的 FPGA 和结构化 ASIC 已引入了芯片资源不规则布置。这些组件可包含逻辑、触发器、特定 I/O 单元 (例如,用于高速串行互联的串化器 / 解串化器)、各种具有不同容量的存储组件以及用于加速 DSP 算法的高速算术块。许多 FPGA 具有有限数量的 RAM、DSP 以及布置在芯片上的稀疏列上的其他专用逻辑块。例如,可仅在芯片布置中的两列中有 DSP 算术块。

[0056] 一方面,本发明提出了芯片架构演进中的这一变化,以在综合流程起始处对物理布局和架构选择进行整合。该需求可位于 RTL 层或确定所需不同类型资源数量的行为综合层。

[0057] 当前知晓资源布局信息以及在综合过程早期 (例如,设计中的许多组件并未具有所选的实施形式时) 对布局和综合进行整合可提供资源的优化利用。例如,并不知晓资源布置信息的 RTL 综合过程可产生一中间网表,该中间网表过度使用了一些资源类型,而其他资源类型并未得到充分利用。此外,资源类型决策可能与资源的物理位置并不协调。例如,在芯片的局部部分所需的 DSP 资源可能超过了可获得的资源。本套综合方法可通过知晓资源在芯片上的分布而提供这些资源的有效利用,且不仅可知晓是否存在足够的特定资源,而且还可知晓附近是否存在足够的特定资源。因此,由于将信号布线至位于较远处的资源从而可避免大的互连延迟。

[0058] 根据本发明的一方面,可在综合仍处于高层电路表示 (例如,设计中的许多组件还不具有所选实施形式)、或门级描述仍待确定时,确定各种布局决策。该布局决策可使得对电路参数 (例如,时间延迟或功耗) 进行精确评估成为可能,且允许了一条实现最佳设计实施形式的增量式路径。在一种实施方式中,如图 3 所示,在操作 30 中,过程起始于 IC 设计的初始状态,该初始状态可包括 ESL 或 HDL 语言、行为抽象化、或编译后的 HDL 代码至 RTL 网表的高层抽象化,以及时序、元件平面布置、功率和布局约束。在操作 31,执行综合转换,在该过程的早期阶段,该转换可为高层转换。所述综合转换可不仅仅针对部分所述设计。在操作 32,在现有的电路表示上执行布局转换,且在早期阶段,该转换可以是架构层上的布局。所述布局转换可不仅仅针对部分所述设计。在此操作处,布局决策可能需要各种假设和估算,因为在该早期阶段可能缺少详细信息。之后在操作 34,评估 IC 设计状态的准备程度,如果该准备程度满足了所设计和规范的目标,则进行至操作 48 中的传统物理综合。如果所述目标并未被满足 (这在该早期阶段是很常见的),则循环回到另一轮综合。

[0059] 综合的下一代 (当前操作 31) 可对设计表示进行改进,尤其是在拥有物理布局信息 (之前的操作 32) 之后。而且类似的,在拥有综合的改进之后,布局的下一代 (当前操作 32) 可改良其对电路参数的估算。利用此类紧密循环,综合与布局可一起密切地工作,以提供一条实现最佳设计表示的路径,且不需要大量的返工 (rework)。

[0060] 在一实施方式中,所述综合操作提供了电路设计表示的各种实施形式,且所述布

局操作可执行电路参数分析,以助于缩小选择范围。例如,如果实施形式 #1 明显较优,则其将被选择,从而将潜在实施形式的数量缩小为一。可选地,如果实施形式 #2 明显位于设计约束的范围之外,则其可被排除,从而可使潜在实施形式的数量减一。

[0061] 根据本发明的一方面,用于设计多个集成电路的示例性方法根据抽象机器规格给出了经整合的、相互影响且迭代的综合与布局。在一种实施方式中,所述设计集成电路的示例性方法增量式地改变 IC 设计的状态。起始于 IC 设计的初始状态(该初始状态包括 ESL 或 HDL 语言、行为抽象化、或编译后的 HDL 代码至 RTL 网表的高层抽象化,以及时序、元件平面布置、功率和布局约束),所述示例性方法迭代地增量式地改变 IC 设计状态,直至达到了最佳设计状态。该最佳状态优选为满足时序和布局约束的芯片原始级网表,之后该网表可被传递至传统的布局和布线过程,而不需要大规模的修正工作。

[0062] 因此,根据一方面,本发明公开了综合与布局的迭代过程,其中每一次迭代对集成电路的设计提供增量式改变。现参考附图 4 给出本发明某些实施方式的主要示例。图 4 的方法起始于操作 40,其中生成 IC 设计的初始状态。该 IC 设计的初始状态包括行为表示或高层 RTL 网表,该网表可由描述电路和逻辑的 HDL 源代码编译得到。

[0063] 所述与技术无关的 RTL 网表通常是设计的更高层的行为表示。这保存了抽象信息,该抽象信息可用于最终映射步骤之前的过程。这与传统综合工具的不同之处在于,在进行语言编译之后,立即将设计分段为微小的低层(门级)表示。通过保留更高层的行为表示,综合工具可在更为全局的水平上执行优化、划分和元件平面布置,且通常可给出较好的结果。通过对抽象数据进行操作,所述综合工具还可运行得更快,并可处理更大型的设计。所述高层 RTL 网表包含高层抽象化(例如,电路块表示),该网表与任何特定厂商的技术或架构无关。

[0064] 所述 IC 设计的初始状态还包括时序约束、功率约束以及布局约束,例如,IO 管脚位置、现有的元件平面布置或现有的布局(例如,IC 芯片的尺寸和形状、IP 块)。在操作 42 中,增量式地改变所述 IC 设计的状态。集成电路设计的状态主要包括网表、时序数据、资源信息、布局信息、布线信息和功率数据。设计状态中的增量式改变可以是综合或布局修改,且这将在以下进行描述。在本发明的一方面,所述改变是增量式的,这意味着设计优化通常通过对所有当前信息(例如,时序估算和布局约束)进行少量修改来进行。所述增量式改变允许所述设计满怀信心地稳步推进。在一方面,所述增量式改变涉及增量式全局布局算法,例如力导向(force directed)方法。在另一方面,所述增量式改变涉及全局优化算法,例如模拟退火法(simulated annealing)。在操作 44 中,评估所述 IC 设计的状态,并在操作 46 做出是通过回到操作 42 来继续进一步的迭代还是在操作 48 完成设计流程的决策。

[0065] 本电路设计方法在集成电路设计的两个基本步骤之间设置高度集成且迭代的过程,该过程为综合和物理设计(例如,布局和布线)。由于在没有布局的情况下无法在综合中对设计约束进行准确估算,且在没有综合的情况下无法执行布局,利用综合与布局强烈地相互依赖的理念,本发明的设计方法通过增量式迭代方法有效地将综合与布局合并成一个步骤过程。

[0066] 在一种实施方式中,本发明提供综合/布局转换的迭代。该迭代过程的主体可以是布局转换、综合转换或者综合与布局转换的组合。无论在何种情况下,集成电路设计的状态均增量式且迭代地朝向满足设计目标的芯片原始级的综合或布局改变。图 5A 和图 5B 示

出了用于设计 IC 的部分流程的两个示例,在图 5A 所示方法的情况下,首先进行布局转换,之后进行综合转换,而图 5B 中则相反。综合、布局或综合 / 布局的增量式且迭代的转换提供了在设计在任何状态下综合与布局之间的连续互动。综合与布局的增量式且迭代的进展确保了所述综合转换总是具有最新且最精确的设计状态信息(该信息包括延迟信息以及来自布局转换的局部资源可用性),且其中所述布局转换总是基于最新的综合网表,提供对物理布局和连线信息的最好估算。布局与综合转换持续进行,直至所述网表仅由芯片级本原所构成、满足了设计目标、且布局拥塞被减小至零件布局器(detail placer)可很容易独立地使得任何微小局部区域合乎规范的水平。该流程之后,可进行传统物理综合流程,以完成整个实施。

[0067] 图 6 显示了本发明的用于所述 IC 设计状态的增量式改变的实施方式。本发明可同时设置(place)各层抽象化。相比于后期迭代(其中,设计几乎由芯片本原所构成),在早期迭代期间,较高层的抽象化对象更为普遍。芯片本原实例通常为最低层表示。综合转换逐步修改所述网表,将高层抽象化的对象变为更具体的对象。这些具体对象具有更为特别的资源需求,该需求可在后续的综合与布局转换中被考虑。布局转换确定网表实例(RTL 实例、未被映射的实例、被映射后的实例、或芯片原始级实例)的位置,从而确定电路中除布线器(router)之外的网络的长度及延迟。所述布局转换可朝着规范布局(其中,规范的布局意味着满足对 IC 芯片资源的使用进行管理的规则)逐步迭代电路参数。一般而言,在早期迭代中,布局与规范相差较远。由于所述布局转换可对对象位置进行增量式改变,因此布局转换的单次迭代并不能创建合乎规范的布局。通过反复的布局转换,所述布局将变得合乎规范。在本实施方式中,所述布局转换对于本电子设计自动化而言极其重要。

[0068] 在每一次迭代中,迭代的标准可以是时序数据、每资源层的拥塞、区域利用率、功率水平或他们的任意组合。所述方法可进一步包括可能的内部循环迭代以优化所述设计、形成关键途径、或根据预定阈值散布所述资源。

[0069] 通过本发明的综合与布局转换的增量式迭代方法的实施方式,在设计的任何阶段,总是可以在综合转换时获得物理设计信息。因此,所述综合中的优化和转化在时序、区域以及对可布线性的影响方面总是最新的。在综合中所作出的有关电路结构的决策与所述布局充分配合。

[0070] 本发明的综合与布局转换的增量式迭代方法有效地将综合与布局转换组合起来,从而可同时优化逻辑结构以及电路的空间布局。在该方法的普通示例中,集成电路的设计状态增量式地向最终的电路规格和布置进行改进。

[0071] 迭代的布局转换的改进可以是网表或布局配置的成熟化的增加程度。设计的成熟化通过以下各项的程度来度量:网表仅由芯片级本原所构成、满足了设计目标、以及布局拥塞被减小至零件布局器(detail placer)可很容易独立地使得任何微小局部区合乎规定的水平。

[0072] 所述迭代的综合转换的改进可以是综合优化,例如对对象或实例进行重构或复制以满足时序约束。综合优化包括但不限于:电路优化、抽象组件分解、算术映射、不进行 / 进行资源共享、加法树分解、基于布局的与 / 或门分解、路径复制、路径绕行移除(path detour removal)、对离散资源(例如,RAM 或 DSP)的分配、逻辑因素分解、复用器重构、或有助于跨层级优化的网表平化。

[0073] 图6显示了该方法的实施方式,该实施方式起始于操作61,在操作61中生成IC设计的初始状态。该IC设计的初始状态可以是具有相关状态信息(例如,时序数据、资源信息、布局信息、布线信息和/或功率数据)的RTL网表。一般而言,所述IC设计的状态包含足够的信息来指明电路需求,例如功能、时序、功率和元件平面布置。

[0074] 高层RTL网表包括一网表,该网表中大多数对象均为低层芯片本原的抽象表示。多组相关本原可被表示为具有高层表示(表示经RTL编码的功能)的对象。集成电路的高层表示或抽象表示可以是逻辑对象,表示RTL代码或部分RTL代码。每个对象通常表示多个芯片本原,例如更为复杂的功能(例如,加法器、乘法器、复用器及顺序逻辑)以及AND功能、OR功能。具有高层表示的对象还可以包括存储块或私有(知识产权块或IP)块。其他逻辑对象可为部分RTL代码,以提供支持功能,例如胶合逻辑(提供缓冲器或接口功能)、时序逻辑、控制逻辑或存储逻辑。一些高层RTL对象还可以为芯片级本原。由对象构成的网表还可包括与每个对象相关的用于连线和布局的信息。所述对象可以包括用于变回至(map back to)相应RTL代码的信息。

[0075] 此外,RTL代码可包含多个层级,其中多个功能被组合在一起。在一些情形中,多个组件可从一个层级重新分组到另一层级,以优化时序、布线、区域或功率需求。在其他情形中,可在增量式迭代过程期间,整体或部分地对功能RTL层级进行平化。

[0076] 首先,设计的初始状态可包括约束,例如时序约束、功率约束和/或布局约束。例如,布局约束可包括IO管脚的位置、现有的元件平面布置或现有的布局数据。

[0077] 在示例性实施方式中,可基于时序由一系列中性优化来对所述设计的初始状态进行首次优化。所述中性优化包括很容易撤销的任何区域恢复,例如,撤销/进行资源共享;基于扇出表时序的加法树分解;明显的资源完善(refinement),例如如果设计中存在巨大的RAM且仅存在一个可用的RAM块资源,则所述RAM必须位于所述RAM块;有助于跨层级优化的网表平化;以及复用器结构的提取和重构。

[0078] 在操作62,基于当前设计状态(当前布局、网表、资源、时序、功率和布线)选择下一转换,以增量式地改变所述IC设计的状态。操作63-70为根据本发明一种实施方式的普通转换,包括布局或更新布局(63)、分配资源(64)、因素分解(65)、映射(66)、优化逻辑(67)、创建/完善实施形式(68)、更新布线(69)以及其他综合(70)。所述转换通常是小的增量式操作,以允许布局与综合准确无误的整合,即,在知晓布局的情况下执行综合,且在知晓综合的情况下执行布局。

[0079] 因此,迭代和增量式转换63-70包括布局与综合操作,包括优化转换以及布局转换,所述优化转换例如为撤销/进行资源共享、加法树分解、AND/OR门分解、逻辑复制、比特拼接(bit splicing)、绕行移除、因素分解,所述布局转换例如为对离散资源(RAM、DSP等)的分配以及布线。

[0080] 在示例性实施方式中,在每一次迭代时,在操作62基于价值函数对各种潜在转换进行评估。所述价值函数被设计为用于选择最佳转换来进行首次操作,从而包括设计状态信息,例如时序、布局拥塞、布线拥塞、区域利用率以及功率。在评估之后,执行最佳转换,且继续进行迭代,直至满足了设计约束。在一方面,所述设计之后可进行到传统的门级布局和布线。

[0081] 在每一次迭代时,所述方法经过一系列选择,并基于价值函数选择最佳转换。例如,

布局转换与综合转换之间的选择是基于时序收敛判别标准的。在关键路径上,如果可能的话,布局可试着缩短临界网络(critical net)。如果临界网络不能被缩短,则该网可用于物理综合优化。

[0082] 根据本发明另一方面,用于设计集成电路的示例性方法提供转换迭代,其中综合与布局转换没有一定的顺序,其中在迭代中,下一转换是基于集成电路设计的状态而被选择的,以向具有时序和布局约束的最终配置改进。在一种实施方式中,所述方法提供转换选择算法,其中下一转换是基于某一标准(例如,时序、每资源层的拥塞、区域利用率以及功率)而被选择的。所述下一转换可以是布局的更新,其中电路将经历一次迭代以进行布局改变,使得当前网表具有更少的资源拥塞或更好地满足设计目标。所述下一转换可已是综合优化,例如,因素分解、优化或分解。所述下一转换可以是综合优化,例如分割、重构或复制,以满足时序或关键路径需求。所述下一转换可以是综合,其中当前网表可朝着芯片原始级网表被映射至较低的抽象水平,以定下电路规格和布置、或更新布线。

[0083] 所述下一转换可以是布局优化,例如元件平面布置划分、资源分配、用于满足时序或关键路径需求的逻辑重构或复制、或更新实例布局的布线。所述下一转换可以是综合操作,其中当前网表可朝着芯片原始级网表被映射至较低的抽象水平,以定下电路规格和布置。

[0084] 利用增量式转换,设计状态信息(例如,时序和功率)是最新的,从而可通过精确考虑对目标的影响来执行优化。

[0085] 在可选实施方式中,可选择若干个转换。之后应用每个所选转换来测量对设计状态的影响,并在之后进行恢复或撤销。之后,选择和应用最佳转换。

[0086] 在一实施方式中,本发明的关键步骤为操作 68,该操作针对网表中的每个 RTL 对象创建或完善可能的实施形式选择。相关函数执行对每个可选实施形式所需的形状和资源的估算。在另一实施方式中,操作 68 还可以为每个实施形式分配权重,以表示优选实施形式。在架构层合并了综合与布局的本发明的一个关键优点在于,允许对不同架构实施形式进行评估。在没有本架构物理综合的情况下,一旦在 RTL 综合阶段选择了实施形式,则在门级布局阶段,将不可能对高层信息进行恢复。如果其他实施形式已是优选的,这可能造成次优选。因此,如果利用物理信息在 RTL 层作出了实施形式决策,则可获得更佳的时序效果。在电路已被映射用于布局和布局阶段之后,该转化将很难被执行。

[0087] 随着迭代的进行以及设计状态的完善,操作 68 可利用下级特性(inferiorproperty)来排除实施形式选项。下面将利用函数示例 F 来解释操作 68,该函数 F 实现为  $F = S \& (A * C) \mid \mid \sim S \& (B * C)$ 。如果选择信号 S 为 1,则 F 为 A 与 C 之积,而如果选择 S 为 0,则 F 为 B 与 C 之积。操作 68 针对该函数确定可行的可选实施形式。图 10A 和图 10B 示出了两个可行的可选实施形式,其中所述创建/完善实施形式操作可创建所述函数。图 10A 显示了使用两个乘法器以及一个复用器的实施形式,如果输出 F 对时序要求较为严格且选择信号 S 具有最新的到达时间,则该实施形式可能是所需的。图 10B 显示了使用单个乘法器和复用器的实施形式,如果输入 C 为最新到达的信号、或者如果输出 F 对时序要求并不严格且需要减小区域,则该实施形式可能是所需的。这两个可选实施形式示出了资源共享/不共享。在没有有关时序和功能布局的明确信息的情况下,普通高层综合算法通常无法评估可选实施形式(例如,图 10A),因为其使用资源以用于两个成本非常高的乘法器。

这即使在传统流程的布局将未被利用的专用乘法器资源附近的该函数替换为其关键输出和在 A、B 和 C 之后到达的选择信号 S 的情况下亦是如此。在本发明中,操作 68 可创建这两个实施形式以及可能创建排除明显较差的实施形式之后的其他实施形式。例如,随着迭代的进行,可清楚了解到,输出 F 并不关键。在此情况下,操作 68 可将实施形式选项精简至仅图 10B 这一种情况,因为该实施形式使用更少的资源。可选地,如果 F 和选择信号 S 非常关键且附近存在实施乘法器的可用资源,则操作 68 可排除图 10B 中的实施形式。

[0088] FPGA 芯片通常具有多个预扩散存储资源(例如,触发器)以及多个具有可变比特大小的块(例如 512、4K 以及 MRAM)。设计所需要的存储组件的大小亦可改变。一般而言,不清楚应该如何实施这些存储组件。例如,可利用触发器、512 的资源或者甚至 4k 的资源来实现 2 至 512 比特之间的适当大小的 RAM。而且,芯片上用于较大存储尺寸的资源站点(resource site)通常是非常稀少的。在之前的 EDA 工具中,在存储器实施阶段无法得到布局信息。因此,实施形式决策是在没有局部使用和精确时序信息的情况下作出的。该限制可导致严重的性能下降。如果适当大小的 RAM 被实施为 512 的资源且仅有的 512 站点位于与所述 RAM 所连接至的逻辑相远离的位置,则使所述 RAM 成为 512 将导致很长的内部连线,且会使在起伏实施方式(flip-flow implementation)上使用 512 站点的延迟效果失效。即使使用触发器的实施形式的延迟可能较长,但如果该实施形式允许 RAM 的触发器与该 RAM 所连接至的逻辑之间的互联线路更短,则可能造成更高速的设计。可选地,如果所述 RAM 的连接逻辑附近存在 4K 可用资源,则实施为 4K 可能是有益的。因此,存储器实施形式决策应该在考虑各种可用存储资源以及连接至存储器的组件的位置的情况下作出。

[0089] 图 9A 示出了存储器实施形式决策的示例。该图显示了示例性芯片,该芯片具有位于该芯片的顶部和底部的存储资源。4 比特的 RAM 连接至位于芯片右侧的垫脚以及 AND 门。如果该 RAM 被实施为存储器且被置于芯片的顶部,则可能导致该存储器与芯片的垫脚输入以及芯片所驱动的 AND 门之间的互联线路非常长。图 9B 显示了同一逻辑的可选映射。通过使用附近的逻辑来实施所述 RAM,因此使得互联线路和延迟更短。

[0090] 与操作 68 密切相关的是估算实施形式所需形状和资源的函数。在一种实施方式中,该函数执行映射,以估算用于 RTL 组件的资源。在另一实施方式中,所述映射是针对目标芯片架构的。这些资源估算是基于综合的,该综合被设计为估算逻辑需求以及特定组件的输入/输出需求从而在目标架构中对模块进行实施。此外,在一种实施方式中,所述函数还估算所述组件的时序转换(timing transition)。

[0091] 图 7 示出了将两条总线 A[31:0] 与 B[31:0] 相加以得到第三条总线 O[31:0] 的加法器的示例。实施该加法器所需的逻辑区域是通过转换、估算所述实施形式、确定所需资源以及从其输入至其输出的内部转变延迟而被估算的。在某些方面,例如,所述加法器可通过使用两个逻辑阵列块(LAB)(每个逻辑阵列块由 16 个查找表(LUT)构成)来实施的。

[0092] 操作 65-67 及操作 70 为示例性综合转换,例如逻辑因素分解(操作 65)、逻辑映射(操作 66)、逻辑优化(操作 67)以及抽象化(操作 70),其中可修改 RTL 网表所表示的组件和连接,以产生功能等效电路,该电路改善了设计状态,例如时序、功率。这些转换可添加或移除组件及其内部连接。转换示例包括执行组件复制、或分割单一的 RTL 组件。

[0093] 所述示例性实施方式展现了一大类实施形式选项(这些实施形式选项可用于 I/O、不同大小的存储器、CPU 以及 DSP)中的一种非常简单的情况。不同的设计可能需要以不

同的方式使用这些资源。本发明的抽象化转换（操作 70）能够基于时序信息、被连接组件的位置、每个资源类型的利用率以及布线利用来改变实施形式。所述抽象化转换与所述创建 / 完善转换（操作 68）相类似。虽然操作 68 创建了多个可选实施形式（该多个可选实施形式被保存并在后续的迭代中被评估），但是，所述抽象化操作可从更为具体的实施形式抽象化出抽象组件。考虑抽象组件的各种实施形式，并选择最佳实施形式来替代原始实施形式。该技巧可避免选择，该技巧可列举出所有可能的架构映射选项，并通过映射、布局以及布线而运行所有这些选项。

[0094] 图 11 给出了抽象化转换的示例，该图显示了加法树分解操作。加法树分解将  $n$  输入加法器分为  $m$  输入加法树。在没有从布局得到延迟信息的情况下，该优化不具有关于至加法器的输入位于何处的信息，且仅基于对输入到达时间的粗略估算来形成加法树。在本示例中，如果所有的输入均来自寄存器，则他们大致上具有相同的到达时间。所述分解将为叶节点选择  $(a, b)$ 、 $(c, d)$  以及  $(e, f)$  组合。然而，输入  $b$  和  $d$ 、 $a$  和  $c$  可靠近放置。利用布局信息，最好是为叶节点选择  $(a, c)$ 、 $(b, d)$ 、 $(e, f)$  组合。这将在输出端产生更佳的时序。

[0095] 图 12 显示了门级树分解的另一抽象示例。综合流程中的关键步骤为将大型门（具有许多输入，例如 32 个输入 AND 门）分解成树表示。该阶段一般在流程早期执行一次，以及对树分解的决策并不包括有关所述大型门的驱动器位置的任何信息。本发明包括作为转换的门级树分解和重排，所述转换是布局与时序识别。最不关键且最早到达的输入信号被置于所述树的叶级，且与其他附近的次关键输入分为一组。当时序并非要素时，输入信号根据该信号的驱动器的位置而被分组。

[0096] 操作 67 的优化逻辑转换对网表进行改变，以针对设计目标（例如，时序或功率）来进行优化。所述优化转换的示例为切片操作，如图 13A 所示。如果宽的本原的输入或输出间隔很远，则对该本原进行分割是有益的。该优化只能基于布局信息被执行。以下示例显示了此类情况，该示例针对 2 比特的存储器  $a[1:0]$ ，其输出间隔非常远。所述存储器可被分割成 2 个触发器，该两个触发器之后可被置于靠近他们的输出。

[0097] 在另一示例中，基于组件的扇出或扇入信号而对该组件进行划分。例如，图 13B 所示的示例显示了存储器，该存储器基于该存储器扇出的位置而被分为 3 个簇。从而，原始组件（被示为单个框）被划分，以创建三个新的组件，该三个组件根据其相应的负载而被切片。可基于组件的输入信号来应用类似的划分。该优化是通用的，并不限于存储器。

[0098] 图 14 示出了另一示例性操作，即逻辑复制。复制的情形非常类似于分割。对于具有多个相互远离的输入或输出的组件，制作该组件的副本并将其置于靠近关键负载是非常有益的。该优化可仅基于布局信息而被执行。以下示例显示了此类情况，该示例针对其输出相隔非常远的组件。该组件可被分割为两个实例  $a1$  和  $a2$ ，该两个实例之后可被置于非常靠近他们的输出。这在驱动器的扇出很高时是很常见的。仅所述实例的一个副本被保持在给定物理范围内。

[0099] 图 15 显示了另一示例性操作，即香农扩展。对于具有很大延迟的 RTL 元件的输入锥（input cone）处的逻辑（例如，加法器或乘法器），可将关键输入网络“置前（pull ahead）”，以改善时序。所述逻辑被复制，且关键网络被替换为恒定输入 0 和 1，以及复用器被用于选择两个运算符的输出，且关键网络选择哪个运算符副本是输出。可基于恒定输入来进一步简化所述两个逻辑副本。此外，这是优选在知晓逻辑以及驱动该逻辑的关键网络

驱动器的位置的情况下而被执行的优化。。

[0100] 图 16A 和 16B 显示了再一示例性操作,即复用器 / 可编程复用器 (该可编程复用器被定义为具有独热编码选择的复用器) 合并及时序驱动分解。大型复用器在商用电路中是很常见的。对复用器进行分解非常类似于之前提到的加法树及 AND/OR 树分解。只是选择逻辑使得复用器分解更加困难,因为在树内移动较晚到达的输入不仅会影响树结构,而且还会影响所述选择逻辑。对于其他的分解而言,本发明包括基于布局与布线的时序信号,以确定合适的分解。

[0101] 操作 69 为更新布线。本增量式迭代方法为集成电路提供了较佳的可布线性,以改善设计的性能、噪声灵敏度、产量、区域以及功率。增量式迭代过程可逐步改善芯片上的线路拥塞,该线路拥塞为每单位区域所需的线路资源密度。

[0102] 所提到的许多转换均对 FPGA 的功耗存在影响。例如,存储器被分解的方式 (列相对行的形式) 会影响该存储器的功耗。行分解使用更少的功率,但需要附加复用,该附加复用引入附加延迟。在本发明中,可确定用于优化功耗的行相对列的分解,因为利用综合与布局之间的密切联系,可得到精确的延迟信息。

[0103] 操作 63 是布局转换、或更新布局转换。所述布局转换对网表实例 (例如,RTL 对象、未被映射的实例或芯片原始级实例) 的位置进行修改,从而可确定电路中除布线器操作之外的网络的长度和延迟。

[0104] 所述布局转换可根据网表与布局的成熟化来使用各种布局方法。在示例性实施方式中,本布局器采用增量式算法。增量式算法为响应于输入的微小改变而对算法输出产生增量式改变的算法。例如,可使用全局布局 (例如,力导向布局) 来放置成熟度较低的网表和布局。所述力导向布局 (FDP) 方法是本发明全局布局的优选选择之一,因为其是增量式方法,其中 FDP 的迭代可生成增量式布局改变。一般而言,FDP 使用二次规划技术来对网络进行建模,并确定重叠的实例应该如何展开。

[0105] 在一实施方式中,第一步 FDP 解决了不受约束的二次规划问题,该步骤仅对用于对实例进行互连的网络进行建模。该初始方案一般具有很高的拥塞。之后,FDP 迭代地构建扩张力 (spreading force),以将实例从过度拥塞 (很高的实例利用率) 的区域移至不拥塞 (很高的资源可用性) 的区域。这是所述用于使得 FDP 成为增量式算法的迭代步骤的属性。可在所述迭代之间对网表或其他设计状态数据进行改变。当这些状态改变是增量式的时,FDP 中的最终改变对于并未作出设计状态改变的 FDP 而言也应该是增量式的。

[0106] 存在各种 FDP 算法,但所有这些算法均共享一基本理念,即对实例应该移向的方向进行计算,以消除过度拥塞的区域。在给定布局中,假定网络所连接的实例对彼此施加吸引力,该吸引力与实例之间的二次距离成正比。在之前的工作过程中,所有实例均相互排斥,且被吸引至所有布局站点,即使该站点并不适合于该实例。之后,实例被移动,直至系统实现了最低能量状态的平衡。之后,所述 FDP 方法基于施加于实例上的总力的方向来移动该实例。

[0107] 一方面,本发明提供了用于对许多现代可再编程芯片的混杂 (heterogeneous) 资源进行寻址的新式混杂资源布局以及一些 ASIC 设计流程。例如,大多数 FPGA 具有各种预定芯片资源,例如 IO、DSP、RAM、LUT、FF 等,这些预定芯片资源仅存在于特定站点。这些预定资源是 FPGA 芯片的预扩散属性的结果。每个资源站点对于可置于该站点的实例数量存在

限制。例如,对于 Altera Stratix-II 芯片,可将 16 个或更少的 LUT 和 FF 置于 LAB 站点,且存在 3 个不同的 RAM 站点,该三个 RAM 站点分别拥有 512 比特、4K 比特以及 64K 比特。

[0108] 在示例性实施方式中,本增量式布局解决了混杂资源问题。在 FPGA、结构化 ASIC 以及一些 ASIC 芯片中,资源可仅被置于某些站点,该站点通常并未均匀分布在布局区域上。大多数全局布局器(包括所有之前的 FDP)均假定了混杂资源,其中任何实例(无论其类型如何)均可被置于芯片边界内的任何有效区域。该先前方法简化了布局问题,因为所有实例均被视为简单的直线型对象(rectilinear object),且只要这些对象没有重叠并位于芯片边界内,布局便是合乎规定的。该简单矩形模型可允许特定类型的实例被置于资源不够充足的邻域。该假设忽略了这一事实,即对于混杂资源而言,每个资源均具有实例必须被置于的特定组的站点。虽然该“组合的”布局可能不存在任何重叠,但当考虑实际资源类型时,该布局可能远远超出了规定范围。模拟退火布局器中的一些先前工作过程已考虑了资源信息,但这些布局器仅被用于放置静态映射网表,而非 RTL 对象。此外,模拟退火可用于非常小型的设计,并且对于大型设计而言,模拟退火会由于运行时间而变得非常困难。

[0109] 一方面,本发明对每个不同的资源站点分别进行了建模,以使得在所有布局转换中,均可由布局器对资源需求进行优化。一方面,本发明对任意数量的站点类型(称之为“层”)进行了建模。这些层被用于确定每一实例上的扩张力。在一种实施方式中,所述层创建于初始阶段。针对芯片上存在的每个资源类型,创建层。层的资源站点记录于该层的供应分布的相应位置。分布为类似于矩阵的二维数据结构,其中一位置处的值给出了该位置处的供应的值。

[0110] 每个实例均被分配给为其消耗资源的层。消耗单个资源类型的实例被称之为原始实例,而消耗多个资源的实例被称之为非本原实例。非本原实例的示例可以是状态机,该状态机消耗了 LUT 和 FF 站点类型。分配给层的每个实例所利用的资源均被记录在层使用分布中。本方法规定了一种非本原实例,通过将所述非本原实例的区域记录在具有该非本原实例所需资源的所有层上来进行处理所述非本原实例。这些使用所起的作用会反过来影响每个非本原实例的层的力的计算。

[0111] 对于层而言,其使用与供应分布之差即为该层的拥塞分布。针对先前的 FDP 方法,可使用该拥塞分布来计算层上每个实例的力。

[0112] 可通过考虑来自其每个资源层的力的加权平均值、或基于那些资源的局部拥塞来计算非本原实例的力。应用至每个层的权重可以是统一权重(uniform weight)或取决于层资源的相对离散性的权重。资源的离散性可通过以下各项来描述:资源所处位置间隔多远、资源的稀疏程度如何、或资源分布的均匀程度或非均匀程度如何。

[0113] 在一实施方式中,可类似于非本原实例的情况计算具有多个可行实施方式的组件的力。可通过考虑来自其每个实施形式的资源层的力的加权平均值来计算所述力。应用至每个实施形式的资源的权重可以是统一权重或取决于选中给定实施形式的概率的权重。

[0114] 本发明的优点在于,实例的力仅取决于使用相同资源类型的其他实例和对于该资源类型的资源供应。例如,如果实例 A 和 B 均具有使用资源 C 的部分,则实例 A(或实例 A 中使用资源 C 的部分)上的力取决于实例 B 中使用资源 C 的部分,而且还取决于可用于布局的资源 C。不同层上的实例的扩张力并不会相互影响。

[0115] 一方面,当全局布局器终止时,每个实例均位于适用于其类型的有效站点处或附

近,因此可利用很小的移动使得布局合乎规定。相比于先前的 FDP(要求所有的实例均被建模为单个类型,所有资源区域均被组合,且实例之后被散布在组合后的区域上)而言,该方法是全新的。

[0116] 在示例性实施方式中,本发明的架构物理综合可改善资源利用问题。芯片资源超出了电路需求的情况是很常见的。例如,在 FPGA 设计中,将被实施的电路可能需要 150 个 LUTs,而于其中实施该电路的芯片或部分具有 256 个 LUTs。该问题被称之为资源利用问题。当忽略该资源利用问题时,布局器通常将电路实例均匀散布于可用资源之上,即使可通过具有可变资源密度的布局而实现更佳的结果。先前的布局器要么忽略该问题,要么插入额外的“填充”实例。填充实例是不会给电路添加连通性的额外实例。“填充”实例的使用也是存在问题的,因为必须确定这些实例的位置。

[0117] 在示例性实施方式中,本发明采用区域移除方法来解决资源利用问题。对于力的生成而言,每个资源层均被单独考虑。在所述区域移除方法中,资源基于他们的质量而被利用,低质量的资源被移除。首先确定质量度量,之后分析资源供应,以基于他们的质量来确定资源的等级。之后,将低质量的部分移除到布局器的布局站点考虑之外。由于布局中的改变会影响资源的质量,因此可在布局过程期间多次执行等级评定和移除。因此,该过程非常适用于本发明对设计状态的迭代增量式改善。

[0118] 在一实施方式中,用于形成所述等级的质量度量是基于资源的使用距离的。一种计算力的方法的副产物是得到层的密度分布与格林函数的褶积(convolution)。该褶积的结果可被视为拓扑图,其中较高的点指示需要资源,较低点表示缺少需求。由于分布是由离散框构成的,因此可基于褶积结果对这些离散框进行分类。可按照褶积顺序,通过以具有最低值的资源为起始点来遍历供应并移除资源直至所需的资源被移除,以确定将被移除的资源。一方面,所述方法可留下足够的资源,以使得存在足够的资源来满足实例对该层的需求,且使得芯片是可布线的。

[0119] 可选地,在其他示例性实施方式中,本发明采用力程(force range)方法来解决资源利用问题。在所述力程方法中,作用于每个实例上的力均为来自多个力程的力的加权平均值。一方面,短程加权因子与该短程区域内实例的密度成正比,在所述短程区域中较高的局部密度产生较强的力。因此,该比例性可增强实例扩散,以减少重叠。

[0120] 利用所述力程方法,应用至实例的力取决于该示例邻域内的实例密度。主要思想在于,实例的扩张力应该取决于在其邻域内对该实例进行合法化所需的区域。在大多数极度拥塞的情况(所有实例均重叠在很小的邻域内)中,可基于所有实例及所有资源的位置来计算每个实例上的力。在最小拥塞的情况(其中实例在其邻近并无其他实例,且直接位于资源之上)中,实例将不具有任何力。对于这两种极端情况之间的情况,所述力取决于对实例进行合法化所需的区域中的实例和资源。

[0121] 在一实施方式中,所述力程可分为近程、中程以及远程力。在其他实施方式中,可使用更多或更少的力程。一般而言,在计算与存储资源之间进行折中来确定领域的合法区域以及每个合法范围的力。一方面,可通过改变格林函数的大小来对力进行计算。远程格林函数覆盖整个布局区域;小型格林函数覆盖圆形区域,该区域的半径例如为平均实例区域的五倍;以及中程格林函数具有的半径例如为平均实例区域的 10 倍。实例上的力为该实例的近程、中程以及远程力的加权和。所应用的权重由所述实例邻域内的密度所决定。如

果所述邻域非常稠密,则所述长程力将具有非常高的权重,且近程权重为 0。低密度区域内的实例将具有零长程权重以及很高的近程权重。

[0122] 本方法的另一方面在于,能够确定很重要的架构决策,该决策可确定哪些资源应被用于实施架构结构。在架构层,存在多个决策,例如在 FPGA 上,应将小的 RAM 映射至 512 比特的 RAM 资源还是 4k 比特的 RAM 资源。其他示例包括对乘法器实施形式及之前所提到的诸如加法树分解的情况的决策。但是本发明并不限于这些特定示例。利用布局信息的可用性,本发明可对满足设计目标的重要架构实施形式决策进行完善。示例性的示例是这样的情况,即 1k 比特的存储器可被分配给两个 512 比特的资源或单个 4k 比特的资源。如果所述 1k 比特的存储器所连接的逻辑位于非常靠近 512 比特或 4k 比特站点的位置,则所述实施形式对于能否成功实施是非常关键的。在所述 1k 存储器的连接逻辑非常靠近 512 比特的资源且所述 4k 比特的资源较远的情况下,不佳地映射至 4k 资源将导致电路性能大幅减低。使用布局信息来作出此类及其他架构决策是非常重要的。

[0123] 在示例性实施方式中,可通过将可变层实例的部分区域包含在该实例可被映射至的每个层的使用中来进行实施方式的完善。在我们的 1k 比特的示例的情况下,所述实例的区域可部分包含于 512 层和 4k 层。该实例上的力可通过利用其潜在层的力的加权和、或利用具有最小幅值的力来确定。利用具有最小幅值的力背后的原因在于与该力相关联的层应该具有较低邻域密度。

[0124] 在其他示例性实施方式中,通过不将具有多个可能的资源实施形式的实例包含于任何层的使用中来开始资源实施。在针对所有层执行区域移除操作之后,可考虑这些可变实施形式实例。对于可变实施形式实例而言,其每个可能的层的潜在供应均被考虑。所述潜在供应是由所述区域移除操作从全部供应(full supply)移除的区域。每一实施层上的潜在供应均被检查,以确定哪个具有移除区域的层在实例被置于该移除区域上时破坏性最低。之后,所述实例被分配给所述最低破坏性的层。

[0125] 所述分配资源转换(操作 64)负责将实例分配给其特定芯片资源。可使用各种布局算法来进行该操作,包括力导向布局、模拟退火、混杂(Mongrel)、最小切割布局、数字优化布局、基于评估的布局以及其他特选布局算法。

[0126] 虽然本发明的大多数实施方式欲用于 HDL 设计综合软件程序中,但是本发明并非必须限于此类应用。虽然可使用其他语言和计算机程序(例如,可编写计算机程序来对硬件进行描述,从而所述计算机程序可被视为 HDL 形式的表达且可被编译,或者在一些实施方式中,本发明可分配和重新分配在不使用 HDL 的情况下被创建的逻辑表示,例如,网表),但本发明的多个实施方式是在用于 HDL 综合系统尤其是那些为用于具有特定厂商技术/架构的集成电路而设计的系统的情况下而被描述的。所公知的是,目标架构通常由可编程 IC 供应商来确定。目标架构的一种示例为集成电路(该集成电路为加利福尼亚州圣何塞市 Xilinx 公司的现场可编程门阵列)的可编程查找表(LTUS)以及相关逻辑。目标架构/技术的其他示例包括现场可编程门阵列和来自厂商(例如,阿尔特拉、朗讯科技、超微设备以及莱迪思半导体)的复杂可编程逻辑器件中的那些公知架构。对于某些实施方式,本发明还可用于专用集成电路(ASIC)。

[0127] 本发明的一种实施方式可以是电路设计和综合计算机辅助设计软件,该软件被实现为存储于机器可读媒介(例如,CD ROM、磁性硬盘或光盘或各种其它可选储存装置)中的

计算机程序。此外,可用数字处理系统(例如,传统的通用计算机系统)来执行本发明的许多方法。还可以使用被设计或编程为仅执行一种功能的专用计算机。

[0128] 图 17 显示了可与本发明一起使用的普通计算机系统的一种示例。可使用该计算机系统来执行以 HDL 代码进行描述的设计的逻辑综合。注意,虽然图 17 示出了计算机系统的各种组件,但其并非意欲表示所有的特定架构或对这些组件进行互联的方式,因为此类细节与本发明并非确切相关。应该注意的是,图 17 的架构仅是出于解释说明的目的而提供的,可与本发明结合使用的计算机系统或其他数字处理系统并不限于该特定架构。还应该理解的是,具有更少、或可能更多组件的网络计算机以及其他数据处理系统也可与本发明一起使用。图 17 的计算机系统 17 可为例如苹果麦金塔 (Macintosh) 计算机。

[0129] 如图 17 所示,作为一种数据处理系统的计算机系统 101 包括总线 102,该总线 102 耦合到微处理器 103、ROM 107、易失性 RAM 105 和非易失性存储器 106。微处理器 103(该微处理器 103 可以是来自英特尔、摩托罗拉公司或 IBM 的微处理器)耦合到缓冲存储器 104。总线 102 将所述各种组件互连在一起,并且将所述组件 103、107、105 和 106 互连到显示器控制器和显示器设备 108 以及诸如输入/输出(I/O)设备的外围设备,所述输入/输出(I/O)设备可为鼠标、键盘、调制解调器、网络接口、打印机、扫描仪、视频摄像机以及本领域中众所公知的其它设备。通常,输入/输出设备 110 通过输入/输出控制器 109 耦合到系统。易失性 RAM 105 通常被实现为需要连续供电以刷新或保持存储器中的数据的动态 RAM(DRAM)。非易失性存储器 106 通常为磁性硬盘驱动器、磁性光驱、光驱、DVD RAM 或即使系统掉电之后也保持数据的其它类型的存储器系统。通常,所述非易失性存储器还可以为随机存取存储器,虽然对此并无要求。虽然图 17 显示了非易失性存储器是直接耦合到数据处理系统中的其它组件的本地设备,但可以理解的是,本发明可以利用远离系统的非易失性存储器,例如通过网络接口(例如,调制解调器或以太网接口)耦合到数据处理系统的网络存储设备。如本领域中所公知的那样,总线 102 可以包括通过各种桥接器、控制器和/或适配器来彼此相互连接的一个或多个总线。在一种实施方式中,I/O 控制器 109 包括用于控制 USB 外围设备的 USB(通用串行总线)适配器和/或用于控制 IEEE-1394 外围设备的 IEEE-1394 总线适配器。

[0130] 通过本说明书显而易见的是,本发明的各方面可以至少部分地在软件中体现。也就是说,可以在计算机系统或其它数据处理系统中响应于其处理器(例如,微处理器)执行包含在存储器(例如,ROM 107、易失性 RAM 105、非易失性存储器 106、高速缓冲存储器 104 或远程存储设备)中的指令序列来实现所述技术。在各种实施方式中,可以与软件指令相结合地使用硬线连接电路来实现本发明。因此,所述技术不限于硬件电路和软件的任何特定组合,也不限于由数据处理系统执行的指令的任何特定源。另外,通过本说明书,各种功能和操作被描述为由软件代码来执行或引起以简化说明。然而,本领域技术人员可意识到,此类表达的意思是所述功能是由处理器(例如,微处理器 1103)执行代码而引起的。

[0131] 可使用机器可读介质存储软件和数据,当该软件和数据由数据处理系统执行时,可促使该系统执行本发明的各种方法。该可执行软件和数据可存储于各种位置,包括例如 ROM 107、易失性 RAM 105、非易失性存储器 106 和/或缓冲存储器 104。部分所述软件和数据可存储于这些存储设备中的任一者中。

[0132] 因此,机器可读介质包括以机器(例如,计算机、网络设备、个人数字助理、制造工

具、具有一组一个或多个处理器的任何设备等等)可访问的形式提供(即,存储和/或发送)信息的任何机构。例如,机器可读介质包括可记录/非可记录媒介(例如,只读存储器(ROM)、随机存取存储器(RAM)、磁盘储存媒介、光学储存媒介、闪存设备等等)以及信号传播的电学、光学、声学或其它形式(例如,载波、红外信号、数字信号等)等。

[0133] 在上述说明书中,已参考本发明的特定示例性实施方式描述了本发明。很显然,在不脱离所附权利要求的所提出的更为宽泛的精神和范围的情况下,可对本发明作出各种修改。因此,说明书和附图应被视为说明性,而不是约束性的。

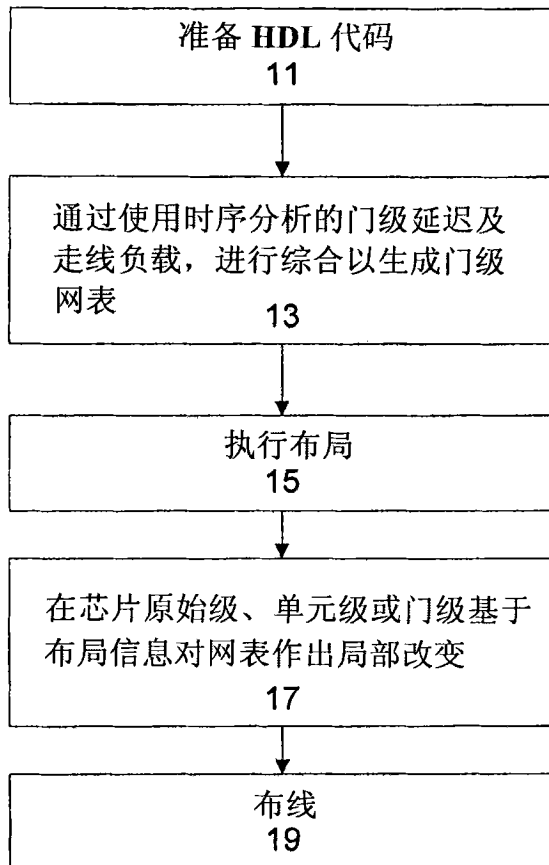


图 1

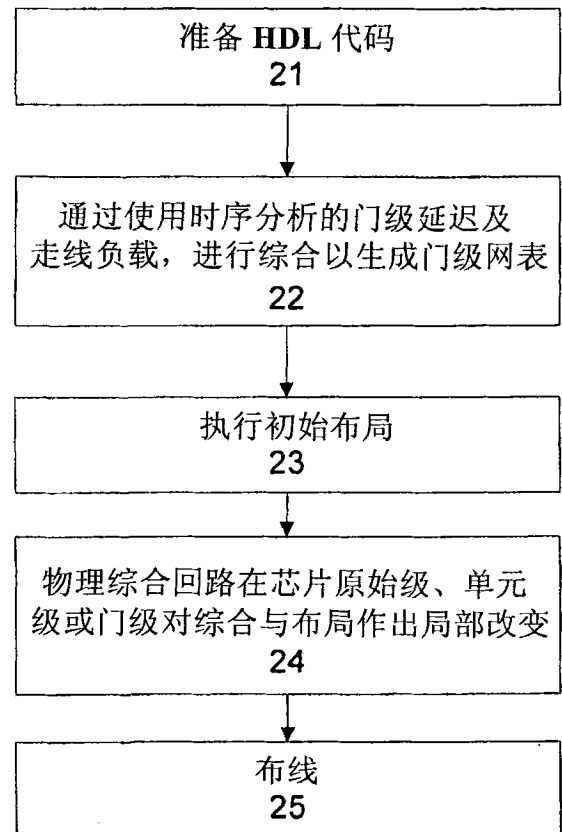


图 2

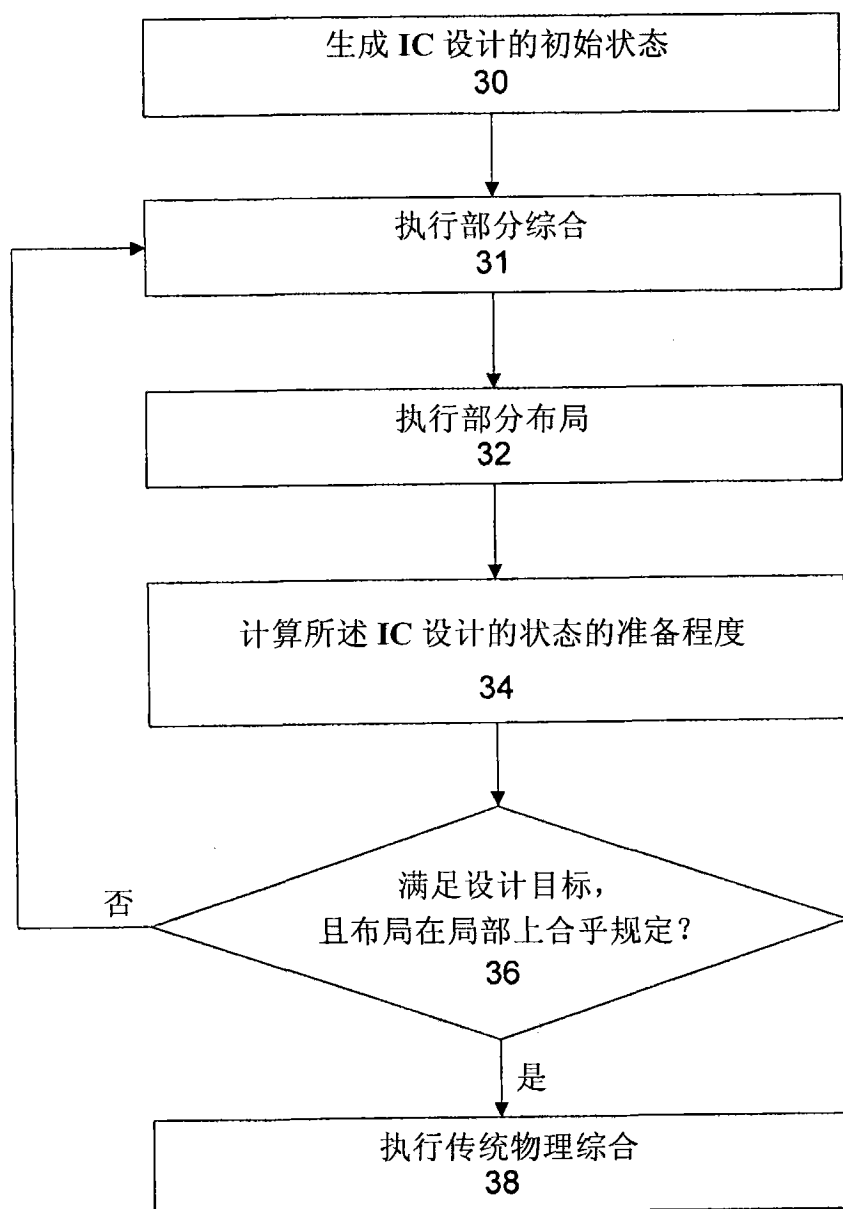


图 3

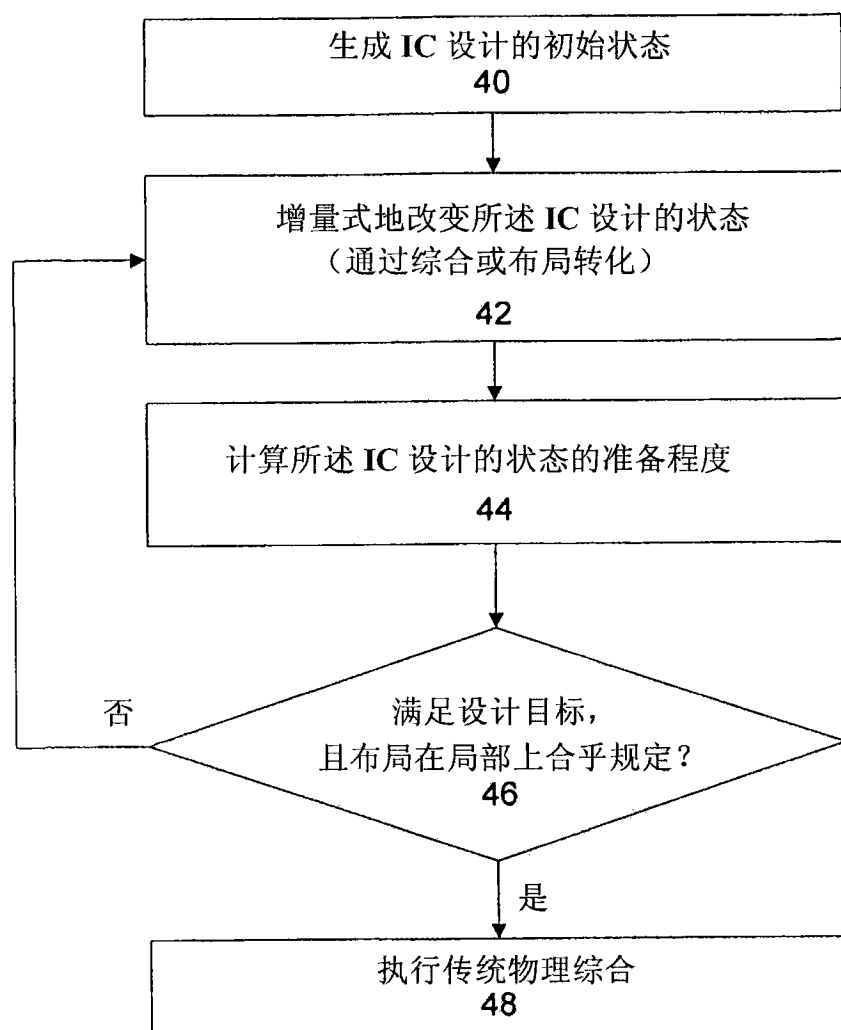


图 4

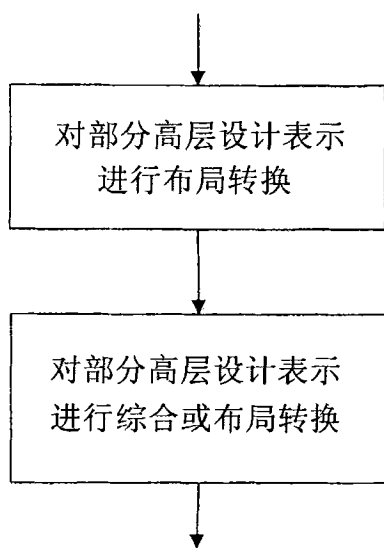


图 5A

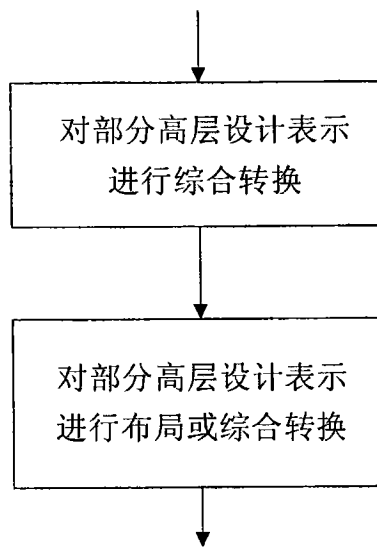


图 5B

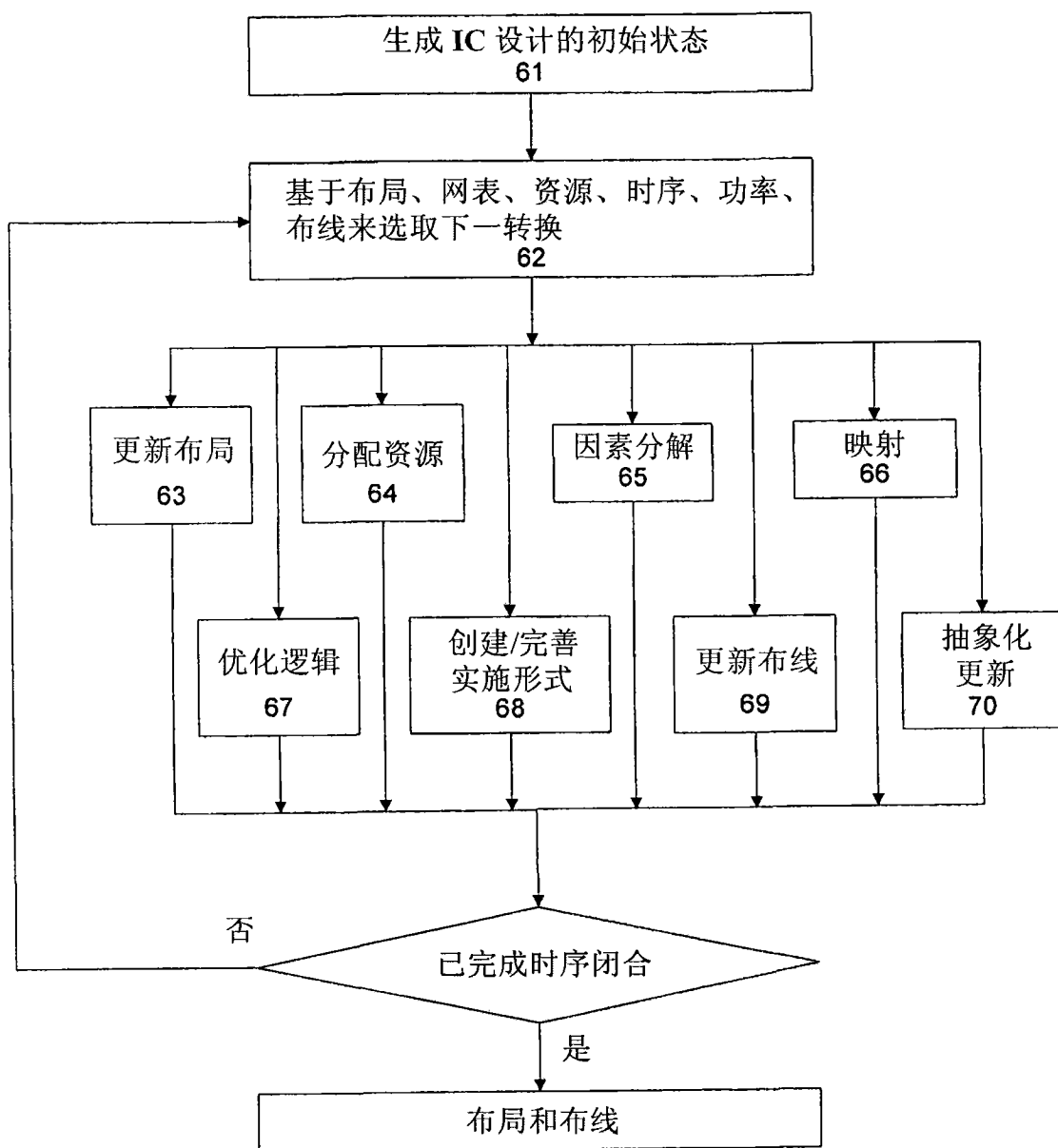


图 6

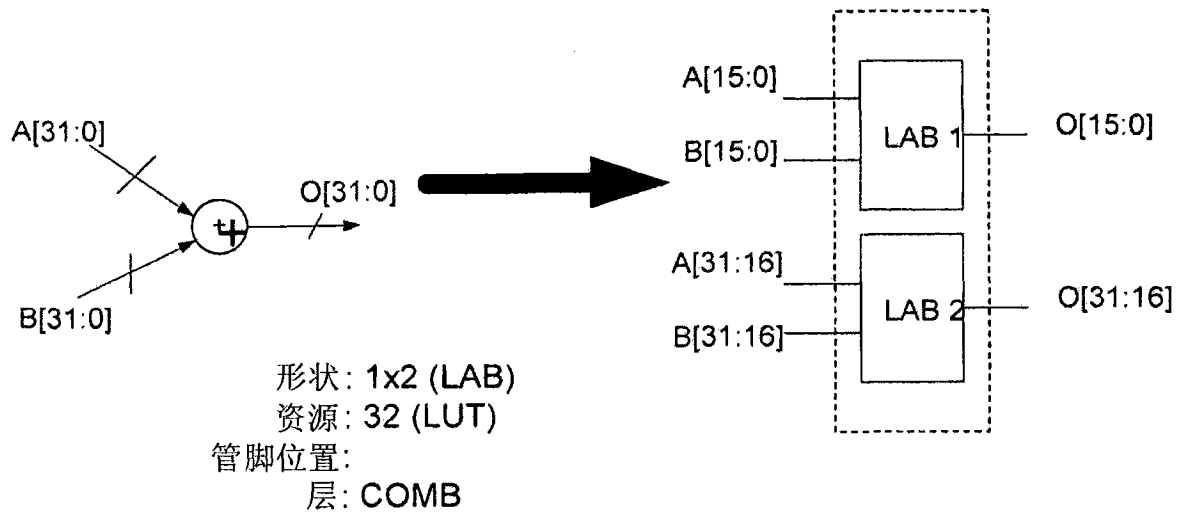


图 7

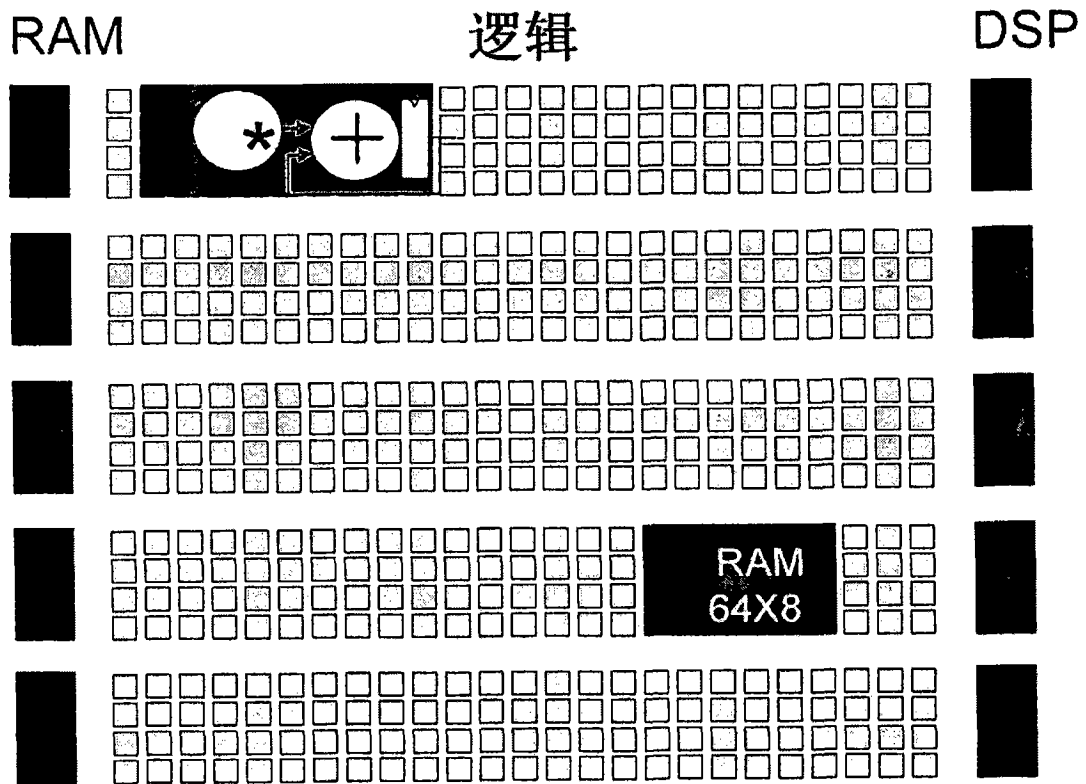


图 8

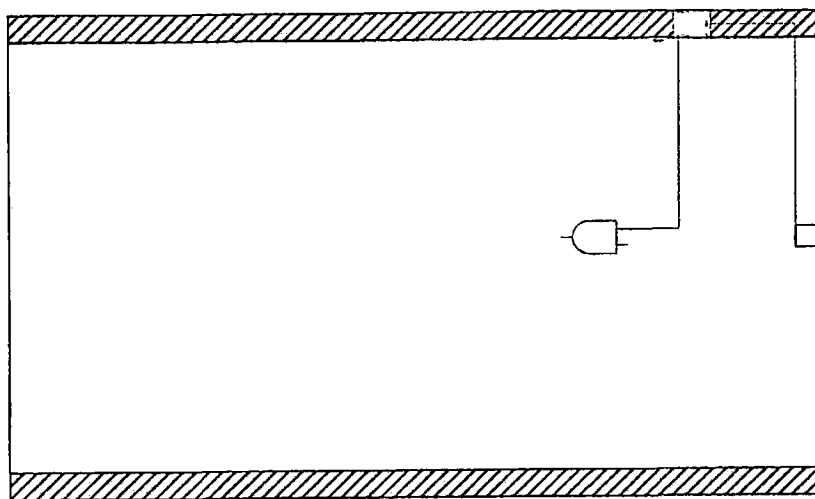


图 9A

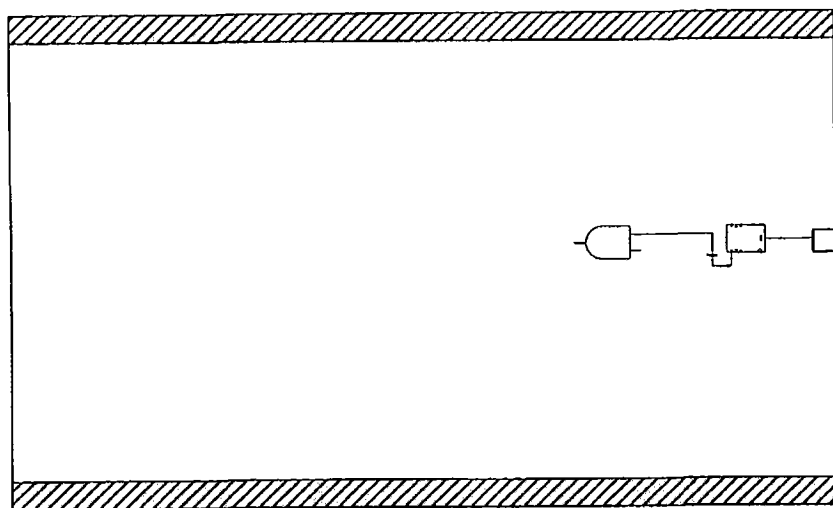


图 9B

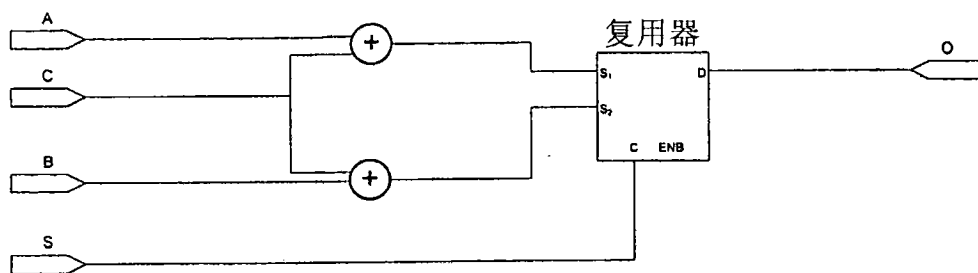


图 10A

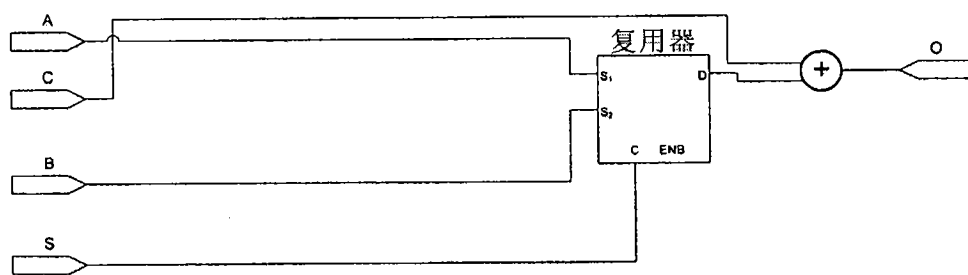


图 10B

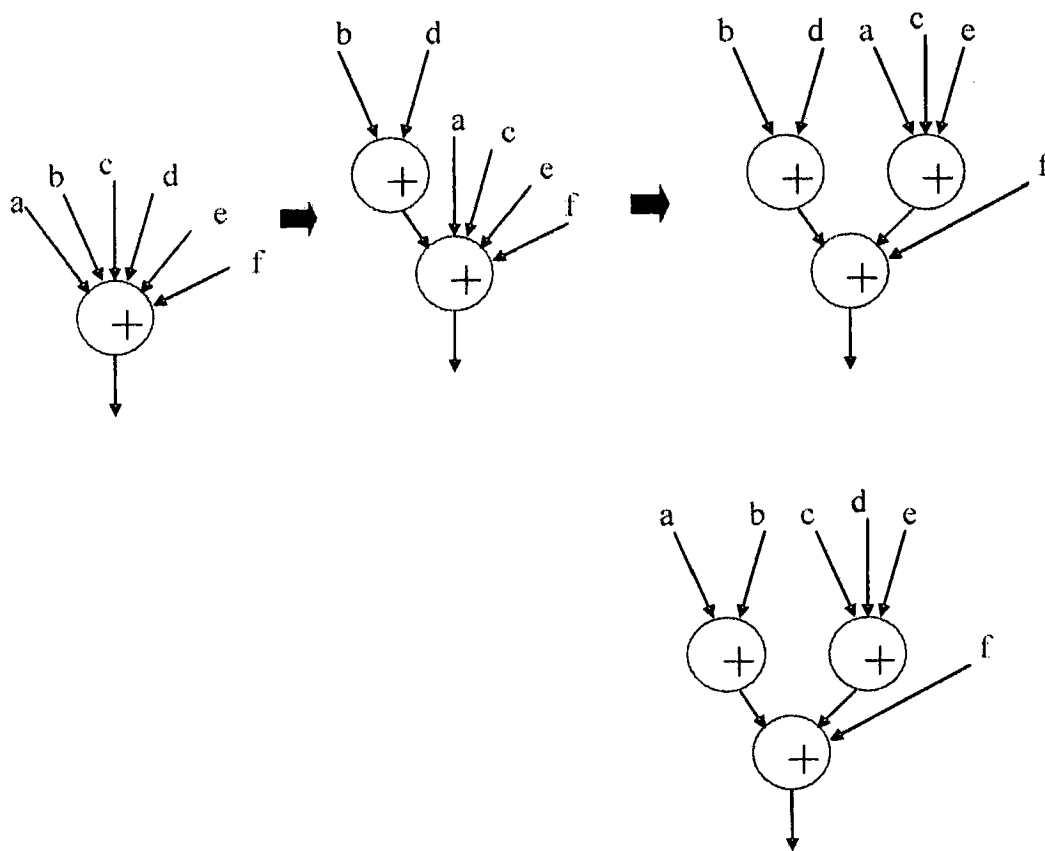


图 11

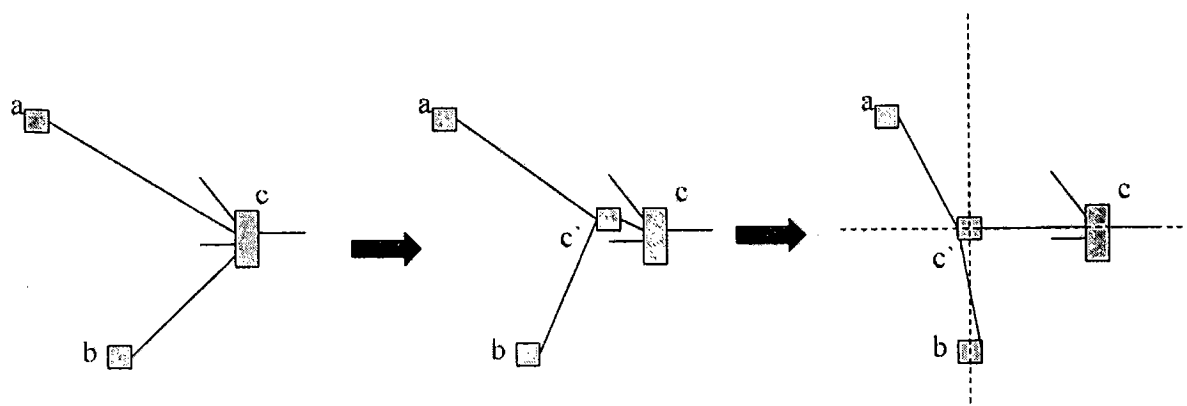


图 12

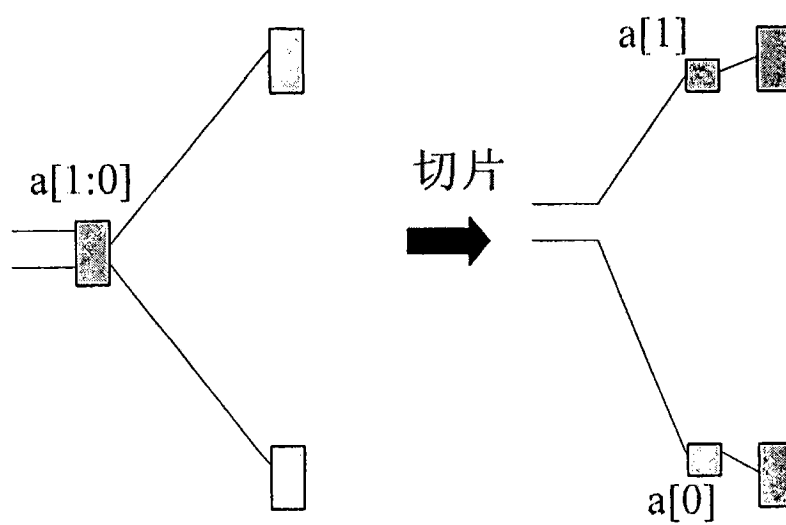


图 13A

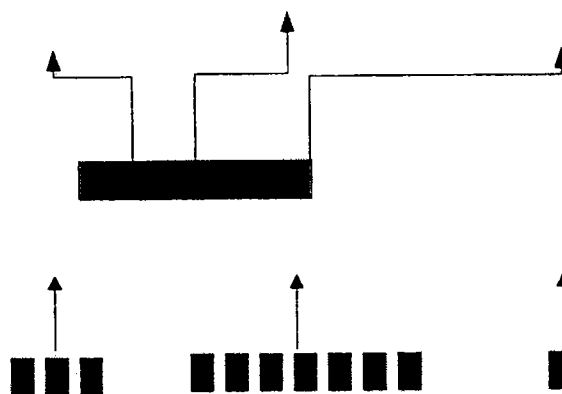


图 13B

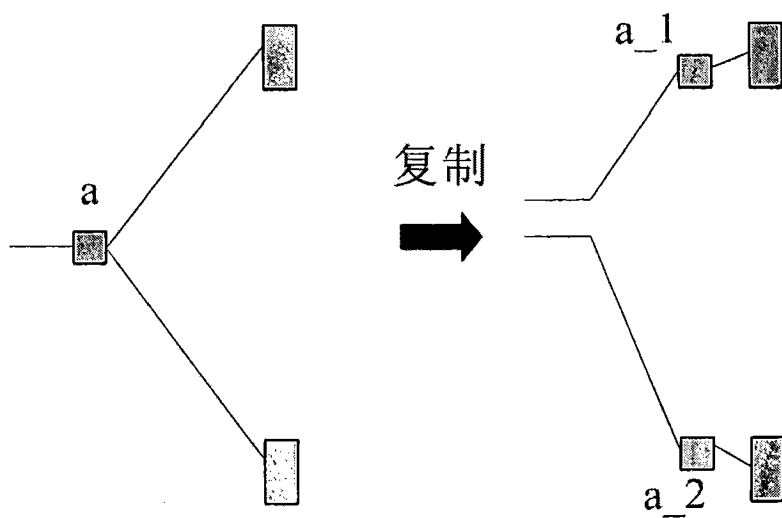


图 14

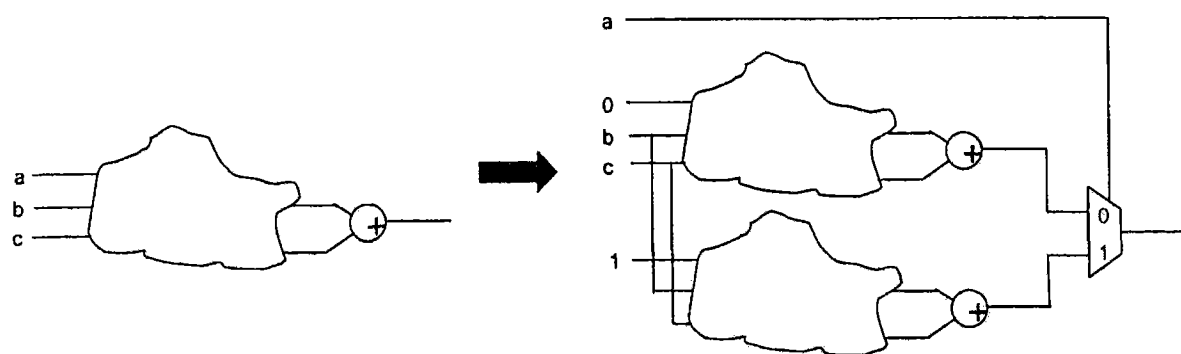


图 15

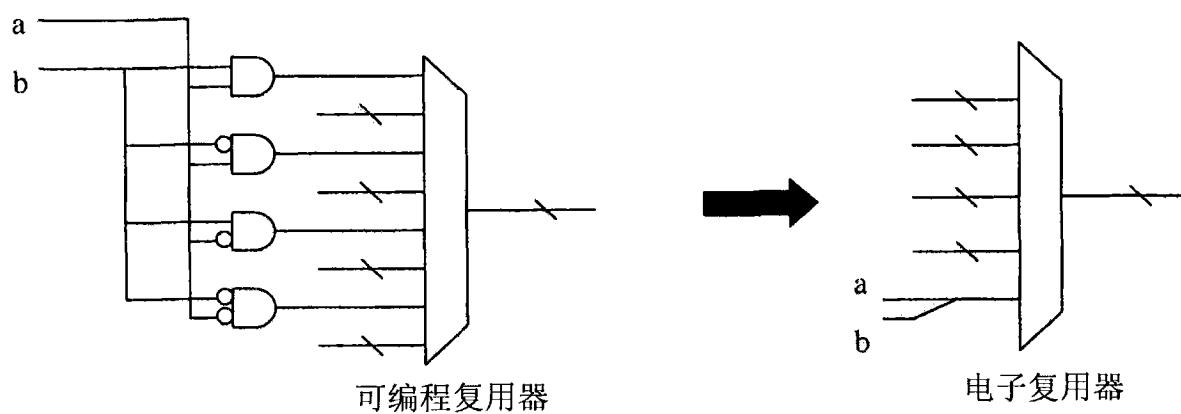


图 16A

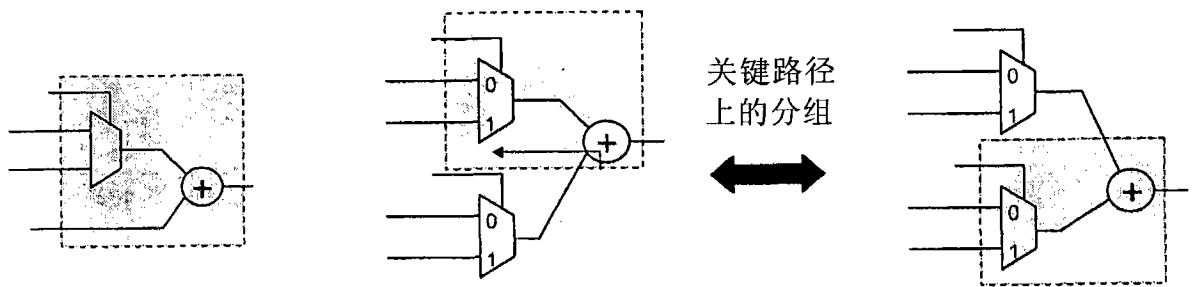


图 16B

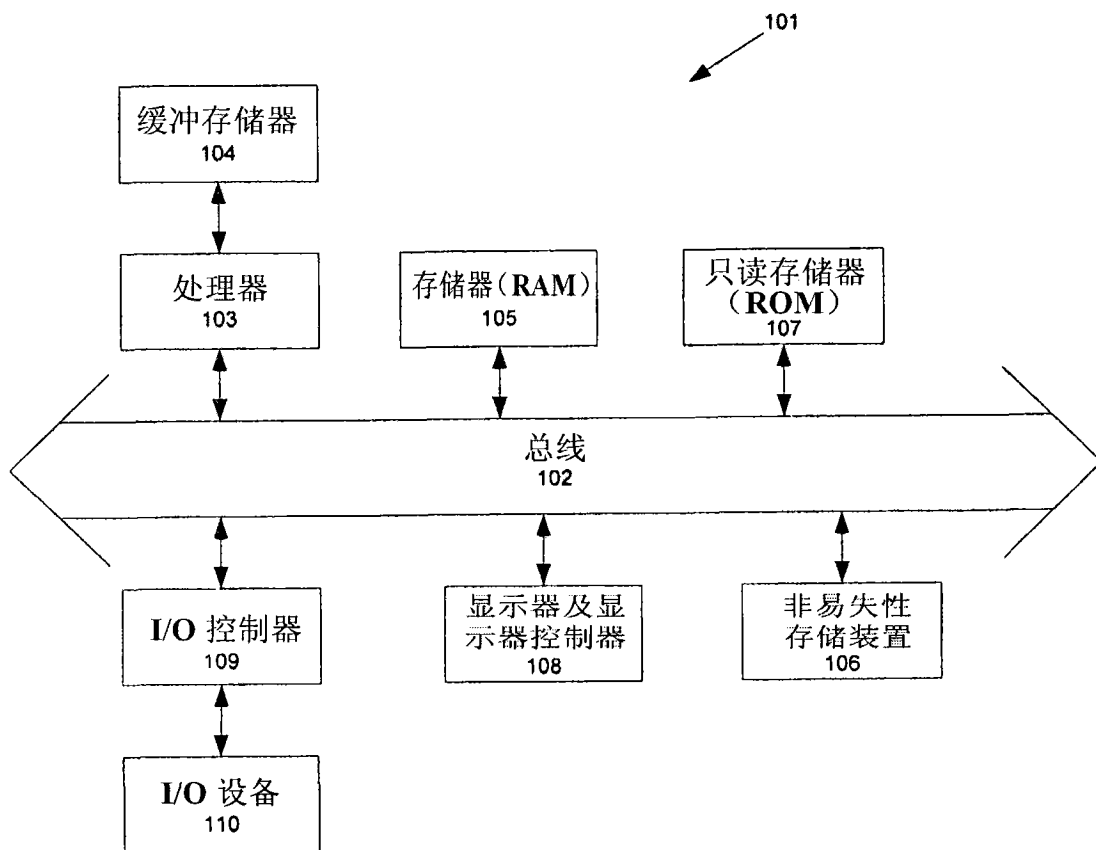


图 17