

文章编号 :1001-3857(2002)02-0055-04

基于 OBDD 时序电路设计的验证

刘建元

(西安邮电学院计算机科学系,陕西 西安 710061)

摘 要:依据有序二叉判定图(OBDD)和计算树逻辑(或称分支时态逻辑)CTL (Computational Tree Logic)的基本原理,分析了基于 OBDD 和 CTL 的验证数字电路设计的基本原理,并在此基础上,给出了时序电路等价验证的方法。

关键词:有序二叉判定图;分支时态逻辑;等价性检验;符号模型检验

中图分类号:TP273 **文献标识码:**A

有序二叉判定图 OBDD^[1]是用于表示布尔函数的一种数据结构,是布尔函数有效的符号表示.与布尔函数其它表示方法(如立方方法、最小项法等)相比,它所用的存储空间较少.这种表示是在二叉判定图 BDD 的基础上加上一些限制,使得最终形式是正则的,即布尔函数与它的 OBDD 具有一一对应关系.因此,布尔函数的比较就变成 OBDD 指针的比较.近年来,出现了许多种基于 OBDD 形式化验证电路设计的方法,如等价性检验、符号仿真、符号模型检验^[2~4]、符号轨迹计算等.文献[5]介绍了如何用基于二叉判定图 OBDD 和分支时态逻辑 CTL^[6]的验证和综合系统 VIS^[7]验证 8 位微处理器 PIC 的设计以及 PIC 设计中部分时序电路的等价性.本文主要研究基于二叉判定图 OBDD 和分支时态逻辑 CTL 验证数字设计的原理,并进一步给出时序电路等价性验证的方法.

1 OBDD 的概念

定义 1 二叉判定图(BDD)是一个具有有限个结点的有根无环图 $G, G = \langle V, E \rangle$, 其中 V 是 G 所有结点的集合, E 是 G 所有边的集合. V 中的结点分为端结点(用方框表示)和非端结点(或称中间结点,用圆框表示).每个端结点 v_T 用 0 或 1 标识,无射出边;每个非端结点 v_N 用变量 $v_{ar}(v_N)$ 标识,并且有两个射出边:指向右孩子 $h_i(v_N)$ 的 then 边用实线表示,表示 $v_{ar}(v_N)$ 被赋值 1;指向左孩子 $l_o(v_N)$ 的 else 边用虚线表示,表示 $v_{ar}(v_N)$ 被赋值 0 (图 1).

定义 2 一个 OBDD 是有序的二叉判定图 $G, G = \langle V, E \rangle$, 如果在变量集合上有全序关系 $<$, 满足对任意非端结点 $u \in V$, 如果 $v \in V$ 是 u 的左或右

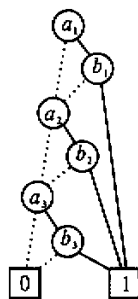


图 1 六变量的 ROBDD

Fig. 1 ROBDD of six variables

收稿日期 2001-10-10

基金项目 国家自然科学基金资助项目(69473017)

作者简介 刘建元(1964—),男,陕西西安人,西安邮电学院讲师,硕士

万方数据

孩子,且 v 也是非端结点,那么必有 $v_{ar}(u) < v_{ar}(v)$.

定义 3 每个 OBDD 上的结点 v 表示了一个布尔函数 $f(v):B^n \rightarrow B$ 满足:

- (1) 若 v 是端结点,则 $f(v) = v_{ar}(v)$;
- (2) 若 v 是非端结点,则

$$f(v) = v_{ar}(v) \cdot f(h_i(v)) + \overline{v_{ar}(v)} \cdot f(l_o(v))$$

这里“ \cdot ”表示逻辑乘;“ $+$ ”表示逻辑加;“ $\overline{}$ ”表示逻辑补.

定义 4 一个 OBDD 是简化的 OBDD(ROBDD) $G = (V, E)$ 满足下述条件:

- (1) 对任意的 $u, v \in V$, 若 $u \neq v$, 则 $f(u) \neq f(v)$;
- (2) 对任意的非端结点 $u \in V$, $h_i(u) \neq l_o(u)$.

例如图 1 中的 OBDD,变量的顺序为 $a_1 < b_1 < a_2 < b_2 < a_3 < b_3$. 对于给定的变量顺序,一个函数 ROBDD 表示是正则的. 研究表明,一个布尔函数的 OBDD 的结构和大小依赖于变量的顺序. 在最不理想的情况下,OBDD 的大小(即非端结点数)随变量数的增加呈指数增长(例如组合乘法). 近年来,人们对 OBDD 的排序进行了大量研究^[8-9],将其大致上可分为 3 种类型:一是启发式变量排序;二是动态变量排序;三是优化变量排序,其时间复杂度为 $O(n^2 3^n)$, n 为变量个数,只适应于变量数较少的电路.

OBDD 最显著的应用就是验证两个组合电路是否等价,即在电路设计中,验证优化前后电路功能是否等价. 基本的算法是根据两个电路输入构造 OBDD,两个电路是等价的当且仅当它们具有相同的 OBDD. 可以通过“符号解释”组合逻辑门网络来构造网络输出函数的 OBDD 表示,即把每个原始输入都表示为一个单变量的 OBDD. 由此出发,根据门操作和所求出的门输入 OBDD,用 OBDD 构造算法对每个门求输出的 OBDD,依此遍历整个网络,其构造如图 2 所示.

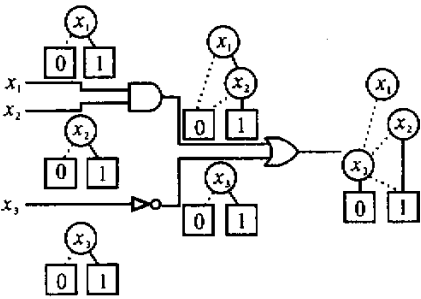


图 2 组合电路的 OBDD 构造

Fig. 2 Constructing OBDD of combinational circuit

2 分支时态逻辑(CTL)

2.1 Kripke 结构

定义 5 Kripke 结构是一个五元组 $K = (S, Q, R, AP, L)$, 其中 S 为状态有限集合; $Q \subseteq S$ 为初始状态集合; $R \subseteq S \times S$ 为转移关系; AP 为原子命题及其否定的集合; $L: S \rightarrow 2^{AP}$ 为标号函数,该函数返回在 $s \in S$ 的所有状态中为真原子命题集合,该集合是原子命题集合 AP 的一个子集.

2.2 CTL 的句法和语义

在数字设计的形式化验证中,常用时态逻辑表示所设计电路的特性.用于表示系统时间特

性的时态逻辑有许多种. CTL 公式由两部分构成, 一部分是路径量词 A (对于所有的路径) 和 E (存在某些路径); 另一部分是线性命题时态逻辑 \neg (总是) X (有时) X (下一时刻) 和 U (直到).

定义 6 CTL 公式规定为

(1) 每个原子公式是 CTL 公式;

(2) 如果 f, g 是 CTL 公式, 则 $\neg f, (f \wedge g), AXf, EXf, A(fUg), E(fUg)$ 是 CTL 公式.

除了定义形成 CTL 语言的公式之外, 在实际中经常用到其它 CTL 公式, 这些公式可以按下列规则得到:

$$f \vee g = \neg(\neg f \wedge \neg g); AFg = A(\text{true}Ug); EFg = E(\text{true}Ug);$$

$$AGf = \neg E(\text{true}U\neg f); EGf = \neg A(\text{true}U\neg f).$$

可以在上面定义的 Kripke 结构 K 中解释 CTL 的语义. 如果对于任意 $i \in \{0, 1, 2, \dots\}$, $(s_i, s_{i+1}) \in R$, 则称一个无限的状态序列 $\pi = [s_0, s_1, s_2, \dots]$ 是从状态 s_0 开始的一条路径. 用 $K, s \models f$ 表示 CTL 公式 f 在 Kripke 结构 K 的 s 状态为真. 在不产生歧义的情况下, $K, s \models f$ 可以简写为 $s \models f$.

定义 7 CTL 的语义:

$s \models p$ iff $p \in L(s)$ 这里 p 是原子命题;

$s \models \neg f$ iff $s \not\models f$;

$s \models f \wedge g$ iff $s \models f \wedge s \models g$;

$s_0 \models AXf$ iff $\forall \pi (\pi = [s_0, s_1, s_2, \dots] \rightarrow s_1 \models f)$;

$s_0 \models EXf$ iff $\exists \pi (\pi = [s_0, s_1, s_2, \dots] \wedge s_1 \models f)$;

$s_0 \models A(fUg)$ iff $\forall \pi (\pi = [s_0, s_1, s_2, \dots] \rightarrow \exists i (s_i \models g \wedge \forall (j < i) \rightarrow s_j \models f))$;

$s_0 \models E(fUg)$ iff $\exists \pi (\pi = [s_0, s_1, s_2, \dots] \wedge \exists i (s_i \models g \wedge \forall (j < i) \rightarrow s_j \models f))$.

3 基于 OBDD 技术的时序电路的等价验证

近年来, 有人已成功地把 OBDD 技术应用到检验两个时序电路是否等价上, 即证明两个模型在功能上是否等价. 这两个模型通常一个是描述系统行为的设计说明, 另一个则是具体结构电路. 在设计中, 一般用 FSM 表示时序电路. 因此, 比较两个时序电路是否等价将转化为比较两个 FSM 是否等价. 基本的算法是给定两个要验证的电路, 把它们的输入连到一起, 把输出送到比较器中, 用同一时钟同步两个时序电路 (图 3). 增加虚线的连接使两个电路变成一个电路. 两个原始的电路是等价的当且仅当按上述连接的电路对所有的可达状态都有相同的输出^[8-9].

为了用 OBDD 计算可达状态, 需要实现下面 3 个基本的要求 (1) 用 OBDD 表示状态集合. 由于 OBDD 表示一个布尔函数, 因此可把 OBDD 看成是使该函数为真的变量赋值的集合, 即 OBDD 表示的函数对给定的赋值为真, 那么这组赋值应在这个集合中, 如果函数为假, 那么这组赋值不在这个集合中. (2) 计算映像. 用 OBDD 表示状态

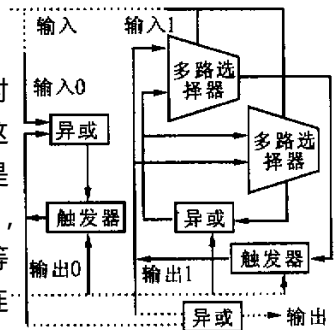


图 3 两个时序电路等价验证

Fig. 3 Equivalence checking of two sequential circuits

集合,那么 OBDD 的映象是一个新 OBDD,新 OBDD 表示在一个时钟周期后,电路所到达的新状态.第一步用 OBDD 表示电路的转移关系,用 X 表示当前状态变量集合, Y 表示下一状态变量集合, X 中的每一变量都与 Y 中一个变量对应.那么序偶 $\{x, y \mid x \in X \wedge y \in Y\}$ 表示电路的转移关系 $R(X, Y)$,这个关系可用 OBDD 表示.第二步合取各个转移关系.(3) 计算可达状态.重复使用映象计算所有从初始状态可达的状态.从初始状态集合出发,利用映象计算以获得所有可达状态的集合.

参考文献:

- [1] Brayant R E. Graph-Based algorithm for Boolean function manipulation[J]. IEEE Trans on Computers, 1996, C-35(8): 34~36.
- [2] McMillan K L. Symbolic model checking: an approach to avoid state explosion[M]. New York: Ph D Carnegie Mellon University, 1992. 20~26.
- [3] Clarke E M, Grumberg O, Long D E. Model checking and abstraction[R]. New York: In Proceedings of the 19th ACM Symposium on Principles of Programming Languages, 1992. 30~33.
- [4] Burch J R, Clarke E M, Long D E, et al. Symbolic model checking for sequential circuit verification[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1994, 13(4): 401~424, 20~24.
- [5] 杜慧敏,刘建元.用 VIS 验证微处理器 PIC[J].计算机辅助设计与图形学学报, 2000, 5: 36~40.
- [6] Hafer T, Thoms W. Computation tree logic CTL* and path quantifier in the monadic theory of the binary tree [A]. In Proc 14th Internat. Coll. on automata, languages and programming[C]. Berlin: Lecture Notes in computer Science, Vol. 267 Springer, 1987. 269~279.
- [7] Brayton R K. VIS: A system for verification and synthesis[A]. In T. Henzinger and R. Alur, Editors, Rutgers University, 1996. LNCS 1102. Eighth Conference on Computer Aided Verification (CAV '96) [C]. New York: Springer-Verlag, 428~432.
- [8] Fujita M, Fujisawa H, Matsunaga Y. Variable ordering algorithms for ordered binary decision diagrams and their evaluation[J]. IEEE Transactions on CAD of Integrated Circuits and System, 1993, 12(1): 6~12.
- [9] Friedman S J, Supowit K J. Finding the optimal variable ordering for decision diagram[J]. IEEE Transactions on Computers, 1990, 39(5): 710~713.

〔责任编辑 强志军〕

Verification of sequential circuit design based on OBDD

LIU Jian-yuan

(Department of Computer Science, Xi'an Institute of Posts and
Telecommunications, Xi'an, 710061 Shaanxi, China)

Abstract: Ordered binary decision diagram is a data structure of Boolean function canonical representation. OBDD can be used to check some properties of Boolean function such as satisfy ability, equivalence, etc. This paper introduces the basic conception of OBDD and CTL, mainly the principle based on OBDD and CTL of verification of digital design. The method of sequential circuits equivalence checking is given.

Key words: ordered binary decision diagram; computational tree logic; equivalence checking; symbolic model checking