逻辑结点函数的优化覆盖*

叶以正 曾献君 喻明艳

(哈尔滨工业大学微电子中心 哈尔滨 150001)

摘要 本文提出多级组合逻辑结构中逻辑结点函数的优化覆盖算法,证明了该算法在多级逻辑优化过程中的有效性.

关键词 多级逻辑综合,逻辑优化,函数覆盖.

多级组合逻辑结构中逻辑结点函数具有不完全确定特性可表示为多级形式[1],利用已有的逻辑结点的函数优化表示其它逻辑结点函数,求取逻辑结点函数的优化表示形式可获取逻辑无冗余、结构优化的多级逻辑结构. K. A. Bartlett[1], S. Muroga[2,3]在利用逻辑结点函数的不完全确定集合优化多级逻辑结构方面已作了许多研究工作,逻辑结点函数的不完全确定集合的求取已出现了一些成熟的算法,文献[4]从基于结构的多级逻辑优化的角度出发,讨论了逻辑结点的可替代函数形式及其求取算法;K. C. Chen[5]从布尔函数的二值决定图 OBDD 表示形式出发讨论了一组逻辑函数的优化覆盖问题.

本文从布尔函数的代数表示形式出发,以文献[4,5]为基础,提出了多级组合逻辑结构中逻辑结点函数的优化覆盖算法,并证明了该算法在多级逻辑优化过程中的有效性.

1 逻辑结点函数的优化覆盖问题

组合逻辑电路的逻辑结构可表示为无有向回路的有向图 $G=(V,E,PI,PO)$,其中

PI/PO:输入/输出信号集合;

$V=V_g \cup PI \cup PO$, $v \in V_g$ 表示逻辑结点(门).

$e_{ij}=(v_i,v_j) \in E$,表示 v_i 至 v_j 的逻辑连线, $v_i,v_j \in V$.

$\forall v \in V$,以 $IP(v)/IS(v)$表示 v 的输入/输出结点集合,$P(v)/S(v)$表示 v 的前导/后继结点集合,定义如下:

$IP(v)=\{u \mid u \in V,(u,v) \in E\}$

$IS(v)=\{u \mid u \in V,(v,u) \in E\}$

$P(v)=\{u \mid u \in V, G$ 中存在由 u 至 v 的组合逻辑通路$\}$

$S(v)=\{u \mid u \in V, G$ 中存在由 v 至 u 的组合逻辑通路$\}$

---

* 作者叶以正,女,1937 年生,教授,主要研究领域为 *ASIC* 设计技术. 曾献君,1967 年生,讲师,主要研究领域为逻辑综合技术. 喻明艳,1965 年生,副教授,主要研究领域为 *EDA* 技术.

本文通讯联系人:叶以正,哈尔滨 150001,哈尔滨工业大学微电子中心

本文 1995—10—04 收到修改稿

$v\in V_g$,v 的逻辑函数具有多级表示形式[1]

$$v=f_{PI}(v)=f_{IP}(v)=f_P(v)=f_{V'}(v) \tag{1}$$

式(1)中,$V'\subseteq V\text{-}S(v)$,$f_{PI}(v)$,$f_{IP}(v)$,$f_P(v)$,$f_{V'}(v)$分别为 v 相对于 PI,$IP(v)$,$P(v)$,V'的函数.

v 的逻辑函数的不完全确定形式如下:

$$\left.\begin{aligned} &v=(v^{ON},v^{DC},v^{OFF})\\ &f_{PI}(v^{ON})=1,f_{PI}(v^{OFF})=0,f_{PI}(v^{DC})=0\text{ 或 }1\end{aligned}\right\} \tag{2}$$

多级逻辑结构 G 中 $v\in V_g$ 的逻辑函数首先表示为 $f_{IP}(v)$,v 的逻辑复杂度定义为 $f_{IP}(v)$的二级逻辑函数形式中的布尔文字数目[1],记为 $cost(v)$,G 的逻辑复杂度定义为 $cost(G)$:

$$cost(G)=\sum_{v\in V_g}cost(v) \tag{3}$$

G 的逻辑优化可描述为:对$\forall\ v\in V_g$,寻求 $f_{V'}(v)$,设 G_v 为 G 中 v 的函数 $f_{1p}(v)$被 $f_{V'}(v)$替代后的逻辑结构,使 $f_{V'}(v)$具有无冗余的极小化形式且 $cost(G_v)\leqslant cost(G)$. $f_{V'}(v)$称为 $f_{IP}(v)$的替代函数,显然,$f_{V'}(v)$具有式(2)的表示形式.

文献[4]对 $f_{V'}(v)$的不完全确定形式作了深入的讨论,设:

$$g=v^{ON}+v^{DC}=\{g_1,g_2,\cdots,g_n\} \tag{4}$$

式(4)中 $g_i\in g$ 为 $V'=V\text{-}S(v)$的逻辑函数,且:

$$f_{PI}(\sum_{i=1}^{m}g_i)\geqslant f_{PI}(v) \tag{5}$$

$v\in V_g$,v 的逻辑函数的优化覆盖问题 OCP(*optimal covering problem*)描述如下:

从式(4)中选取 $g_{i1},\cdots,g_{im}\in g$,构造函数 $f_{V'}(v)=\sum_{j=1}^{m}g_{ij}$,使满足:

1)$f_{V'}(v)$与 $f_{IP}(v)$逻辑等价;

2)$f_{V'}(v)$具有无冗余的极小化形式;

3) $cost(Gv)\leqslant cost(G)$.

2 OCP 问题的求解

$\forall\ g_i\in g=v^{ON}+v^{DC}$,定义函数: $co(g_i,v)=f_{PI}(g_i)+\overline{f_{PI}(v)}$ (6)

式(6)中,$f_{PI}(g_i)$为 g_i 相对于 PI 的二级逻辑函数.

显然,$co(g_i,v)\neq 0$.

令 $H=\{h_i\mid\exists\ g_i\in g,h_i=co(g_i,v)\}$,由于 $f_{PI}(\sum_{g_i\in g}g_i)\geqslant f_{PI}(v)$,故 $\sum_{h_i\in H}h_i\equiv 1$,以 $h_{i\cdot idx}$表示 $h_i\in H$ 所对应的 $g_i\in g$.

OCP 问题可转化为对应的优化覆盖问题 CO_OCP 如下:

从 H 中选取 $h_{i1},h_{i2},\cdots,h_{im}$,使

1)$\sum_{j=1}^{m}h_{ij}\equiv 1$;

2)$\sum_{j=1}^{m}h_{ij\cdot idx}$具有极小化的无冗余形式;

3)以 $f_{V'}(v)=\sum_{j=1}^{m}h_{ij\cdot idx}$替代 $f_{IP}(v)$得 G_v,有下式成立:

$$cost(Gv)\leqslant cost(G).$$

K. C. Chen[5]证明了上述 2 个优化问题的等价性.

OCP 问题求解的基本思路如下:

欲解 *OCP* 问题,先将其转化为等价的 *CO_OCP* 问题,并求 *CO_OCP* 问题的部分解 $h_{i1},\cdots,h_{ik}$,显然 $h_{i1\cdot idx},\cdots,h_{ik\cdot idx}$为 *OCP* 问题的部分解,且知 $h_{i1\cdot idx}+\cdots+h_{ik\cdot idx}$可作为进一步构造 $f_{V'}(v)$的基础,将 *OCP* 问题的求解转化为其子问题的求解.

设 $table=\{g_i|g_i\in g,g_i$ 为 *OCP* 问题的部分解$\}$;

$h=f_{PI}(v)\#f_{PI}(\sum_{g_i\in table}g_i)$;

$fun\_list=g-table$

其中"#"表示锐积运算.[6]

OCP 问题的剩余部分解可由如下子 *OCP* 问题求出:

从 *fun_list* 选取 $g'_{i1},\cdots,g'_{im}$,使之满足:

1)$f_{V'}(v)=\sum_{j=1}^{m}g'_{ij}+\sum_{g_i\in table}g_i$ 与 $f_{PI}(v)$等价;

2)$f_{V'}(v)$为无冗余的极小化表示形式;

3)$cost(Gv)\leqslant cost(G)$.

由上可知,*OCP* 问题可采用递归方式求解,并在递归过程中采用枚举法.

OCP 问题求解的递归过程 $OCP(G,v,g,table,fun_list,h)$从 *fun_list* 中选取子函数以覆盖 h,构造 $f_{IP}(v)$的可替代函数 $f_{V'}(v)$,根据 G_v,G 的逻辑复杂度变化情形,对 $f_{IP}(v)$作逻辑函数替代变换.递归过程描述如下:

step 1. 将 *OCP* 问题转化为 *CO_OCP* 问题,构造 $H=\{h_i|\exists\ g_i\in fun\_list,h_i=co(g_i,h)\}$;

step 2. 求

$$A=\{h_i|h_i\in H,h_i\equiv 1\}$$

$$Ag=\{g_i|\exists\ h_i\in A,g_i=h_{i\cdot idx}\}$$

step 3. 若 $A\neq\Phi$,,则$\forall\ g_i\in A_g,g_i\equiv h_i$,且 $f_{V'}(v)=g_i+\sum_{g_i\in table}g_i\equiv f_{IP}(v)\equiv f_{PI}(v)$成立. 根据 G_v,G 的逻辑复杂度变化情形作必要的逻辑结点函数替代变换,即若 $cost(G_v)\leqslant cost(G)$,对 G 作逻辑结点函数替代变换,执行(1)~(3)直至 $A_g=\varnothing$:

(1)选取 $g_i\in Ag,g_i$ 满足: $cost(g_i)=\min_{g_j\in A_g}\{cost(g_j)\}$;

(2)构造 $f_{V'}(v)=g_i+\sum_{g_j\in table}g_j$;

(3)$repl\_node(G,v,f_{V'}(v))$;

其中逻辑结点函数替代变换过程 *repl_node*()将在算法后描述.

step 4. 令:

$$H=H-A$$

$$fun\_list=fun\_list-A_g$$

step 5. 若$\sum_{h_i\in H}h_i\neq 1$,则不可能从 *fun_list* 中选取一组子函数覆盖函数 h,返回;否则:

step 6. $\sum_{h_i\in H}h_i\equiv 1$，可从 H 中选取 $h_{i1},\cdots,h_{im}$，使 $\sum_{j=1}^{m}h_{ij}\equiv 1$，$\sum_{j=1}^{m}h_{ij\cdot idx}\equiv h$，且有：

$$\sum_{j=1}^{m}h_{ij\cdot idx}+\sum_{g_i\in table}g_i\equiv f_{PI}(v)\equiv f_{IP}(v)$$

令
$$B=\{h_i\mid h_i\in H,\sum_{h_j\in H-\{h_i\}}h_j\not\equiv 1\}$$
$$B_g=\{g_i\mid\exists\ h_i\in B,g_i=h_{i\cdot idx}\}$$

step 7. 若 $B_g\neq\varnothing$，则 B_g 中的子函数是由 *table* 及 *fun_list* 进一步构造 $f_{V'}(v)$ 的必选子函数，即若存在 $g_{i1},\cdots,g_{im}\in fun\_list$，使：

$$\sum_{j=1}^{m}g_{ij}+\sum_{g_i\in table}g_i\equiv f_{PI}(v)$$

则有
$$B_g\cap\{g_{i1},\cdots,g_{im}\}=B_g$$

分下述 2 种情形分别加以处理：

(1)若 $\sum_{h_i\in B}h_i\equiv 1$，则 $f_{V'}(v)=\sum_{g_i\in B_g}g_i+\sum_{g_i\in table}g_i\equiv f_{PI}(v)\equiv f_{IP}(v)$

有 $f_{V'}(v)$ 为由 *fun_list* 及 *table* 构成的唯一最简化的可替代函数形式；调用 *repl_node*$(G,v,f_{V'}(v))$ 对 G 作必要的逻辑结点函数替代变换；

(2)若 $\sum_{h_i\in B}h_i\not\equiv 1$，令
$$h=h\#(\sum_{g_i\in B_g}g_i)$$
$$table=table\cup B_g$$
$$fun\_list=fun\_list-B_g$$

从而将 *OCP* 问题的剩余部分解求取转化为从 *fun_list* 选取一组子函数覆盖 h，并使之满足 *OCP* 问题的约束条件，并进行递归调用.

step 8. 若 $B_g=\varnothing$，这表明以 *table* 为基的 $f_{IP}(v)$ 的可替代函数有多种构成形式，则依次从 *fun_list* 中选取子函数以作为 $f_{V'}(v)$ 的构成部分.

设选取 $g_i\in fun\_list$，令：
$$table=table\cup\{g_i\}$$
$$h=h\#g_{ij}$$
$$fun\_list=fun\_list-\{g_i\}$$

将 *OCP* 问题的求解转化为其子 *OCP* 问题的求解，并进行递归调用 *OCP*$(G,v,g,table,fun\_list,h)$.

递归调用开始时，$g=v^{ON}+v^{DC}$，$table=\varnothing$，$fun\_list=v^{ON}+v^{DC}$，$h=f_{PI}(v)$.

递归调用过程 *OCP*()中逻辑结点函数替代过程 *repl_node*$(G,v,f_{V'}(v))$ 根据 $f_{V'}(v)$ 对 G 中逻辑结点 v 的函数 $f_{IP}(v)$ 替代后逻辑结构的逻辑复杂度的变化情形，对 G 作必要的逻辑结构替代变换，以获取优化的多级逻辑结构，逻辑结点函数的替代过程描述如下：

repl_node$(G,v,f_{V'}(v))$：

(1) $tag=0$; /* *tag* 为替代标志 */

(2) if$(cost(f_{V'}(v)\leqslant cost(v))$ $tag=1$;

(3) else {

(4) $Sup(f_{V'}(v))=\{u\mid u,\bar{u}$ 为 $f_{V'}(v)$ 布尔文字$\}$;

(5) for *each* $u\in IP(v)$\{
(6) if ($|IS(u)|==1\wedge u\overline{\in}Sup(f_{V'}(v))$\{
(7) $cost(f_{V'}(v)=cost(f_{V'}(v))-cost(u)$;
(8) $cost(f_{V'}(v))=cost(f_{V'}(v))-cost\_del(u)$;
(9) }\};
(10) if($cost(f_{V'}(v))\leqslant cost(u)$) $tag=1$;
(11) if($tag==1$)将 $f_{IP}(v)$用 $f_{V'}(v)$替代;
(12) return($table$);

替代过程 $repl_node(G,v,f_{V'}(v))$首先估测利用 $f_{V'}(v)$替代 $f_{IP}(v)$对逻辑结构的逻辑复杂度的影响,当替代变换有利于消除 G 中的逻辑冗余,降低 G 的逻辑复杂度,则进行替代变换,否则保持原有的逻辑结构不变;步骤(8)估测从 G 中删除单元 $u\in IP(v)$对 G 的逻辑复杂度的影响. $cost_del(u)$是一个递归过程,描述如下:

int $cost\_del(u)$\{
c1 $cost=0$;
c2 for *each* $v_i\in IP(u)$
c3 if ($|IS(v_i)|==1$) $cost=cost+cost\_del(v_i)$;
c4 return($cost$);\}

由上述描述的 OCP 问题的递归过程可知,递归调用 $OCP(G,v,v^{ON}+V^{DC},\varnothing,v^{ON}+v^{DC},f_{PI}(v)$可解决 $v\in V_g$ 的逻辑函数的优化覆盖问题. 设G'_v为递归调用后得到的逻辑结构,则有 $cost(G'_v)\leqslant cost(G)$;
$cost(G'_v)$满足最小化条件.

定理. $G=(V,E,PI,PO)$为多级逻辑结构,$v\in V_g$,$g=v^{ON}+v^{DC}$,若 g 中存在一组子函数 $g_{i1},g_{i2},\cdots,g_{im}$,使

(1)$g_{i1}+\cdots+g_{im}\equiv f_{PI}(v)\equiv f_{IP}(v)$;

(2)以 $f_{V'}(v)=g_{i1}+\cdots+g_{im}$作为 $f_{IP}(v)$的替代函数可使作替代变换后的逻辑结构 G_v 满足 $cost(G_v)\leqslant cost(G)$,则经递归过程 $OCP(G,v,v^{ON}+v^{DC},\varnothing,v^{ON}+v^{DC},f_{PI}(v))$调用后的逻辑结构$G'_v$满足 $cost(G'_v)\leqslant cost(G_v)\leqslant cost(G)$;
且G'_v中 $f_{IP}(v)$具有无冗余的极小化函数形式.

证明:只需证明若 g 中存在一组子函数集合$g'_v=\{g_{i1},\cdots,g_{im}\}$,$g'_v$ 满足:

a)$g_{i1}+\cdots+g_{im}\equiv f_{PI}(v)$;

b)$\forall\ g_i\in g'_v,\overline{g_i}+\sum_{g_i\in g'_v-\{g_i\}}g_i\neq 1$,

则调用过程 $OCP()$可求出这样的集合 g'_v,

施归纳于g'_v中的子函数数目 m;

设 $A^{(i)},A_g^{(i)},B^{(i)},B_g^{(i)},table^{(i)}$分别表示在求解$g'_v$过程中第 i 层递归调用时所求出的集合 $A,A_g,B,B_g,table$.

显然,$|table^{(i)}|\geqslant i$,故求g'_v最多需进行 m 层递归调用.

(1)$m=1$ 时,显然可设$g'_v=\{g_i\}$,$g_i\equiv f_{PI}(v)$,$g'_v=A_g^{(0)}$,结论正确;$m=2$ 时,不妨设g'_v

$=\{g_{i1},g_{i2}\}$，则必有$g'_v\cap A_g^{(0)}=\varnothing$，由于$g'_v\cap B_g^{(0)}=B_g^{(0)}$，可分下述情形：

i) $|B_g^{(0)}|=2$ 时，显然有 $B_g^{(0)}=g'_v$，在第 0 层递归调用时可求出g'_v；

ii) $|B_g^{(0)}|=1$，则易由算法知，g'_v 在第 1 层递归调用可求出；

iii) $|B_g^{(0)}|=0$，g'_v 也可由 Step 8 转化为子 $OCP()$递归调用过程，在第 1 层递归调用时可求出.

故 $m=2$ 时结论正确.

(2)设g'_v 中含有 $m=k(k\geqslant 2)$个元素时结论成立，当 $m=k+1$ 时，设求解g'_v需进行 p 层递归调用，$p\leqslant m$，故有 $A_g^{(i)}\cap g'_v=\varnothing \quad i<p$；

$$g'_v\cap\left(\bigcup_{i=0}^{p}B_g^{(i)}\right)=\bigcup_{i=1}^{p}B_g^{(i)}$$

若 $B_g^{(0)}=\varnothing$，则由 Step 8 必可选取 $g_{ij}\in g'_v$，并作递归调用，将 OCP 问题的求解转化为子 OCP 问题的求解，此时由于只需求出 k 个元素即可. 由假设知，结论正确.

若 $B_g^{(0)}\neq\varnothing$，则当 $|B_g^{(0)}|=k+1$，结论显然正确，若 $k\leqslant|B_g^{(0)}|<k+1$ 时，则可由 Step 7 将其转化为子递归问题，由于子递归问题至多需求出 k 个元素. 由假设知结论正确.

由(1)、(2)及数学归纳法知，递归过程可求 $g=v^{ON}+v^{DC}$中不含冗余子函数的 $f_{IP}(v)$的可替代函数形式.

由算法描述知，一但求出 $f_{IP}(v)$的可替代函数$f'_v(v)$，即调用 $repl\_node(G,v,f_{V'}(v))$，整个递归过程对逻辑结构的替代过程是迭代进行的，故最终获得的多级逻辑结构G'_v必是 OCP 问题的最优解对 G 的逻辑结构替代变换的结果，故：

$$cost(G'_v)\leqslant cost(G_v)\leqslant cost(G)$$ □

决定递归调用过程 $OCP()$的计算时间复杂性有 3 个方面：(1)$v^{ON}+v^{DC}$中子函数的数目；(2)重言式判定过程的计算时间复杂性；(3)$f_{PI}(v)$计算及其极小化过程.

最坏情况下计算时间复杂性为 $O(m^2n^2k^2+m\cdot log_2p)$，其中 $m=|v^{ON}+v^{DC}|$，$n=|PI|$，$p=|P(v)|$，k 为递归调用中 h 的最大项数目，k 的大小由具体的函数而定，最坏情况下与 n 呈指数增长的趋势. 一般情况下，k 较大，但呈指数增长的机会很少，故 $OCP()$递归调用可快速完成.

3 逻辑结点函数替代与逻辑结构优化

给定多级逻辑结构 $G=(V,E,PI,PO)$，$\forall\ v\in V_g$，求取 $f_{IP}(v)$的优化可替代函数 $f_{V'}(v)$并作逻辑函数替代变换有利于优化 G 的逻辑结构，降低 G 的逻辑复杂度 $cost(G)$.

图 1(a)为一多级组合逻辑结构 G，$cost(G)=37$，对图 1(a)中的结点 v_8，易求出：

$$v_8=f_{PI}(v_8)=v_5+v_6;$$

$$f_{PI}(v_8)=x_1x_2+\bar{x}_1\bar{x}_2;$$

$g=v_8^{ON}+v_8^{DC}=\{v_5,v_6,\bar{v}_{11},v_{18},v_1v_2,x_1x_2,\bar{x}_1\bar{x}_2\}$；由递归调用过程 $OCP(G,v_8,v_8^{ON}+v_8^{DC},\varnothing,v_8^{ON}+v_8^{DC},x_1x_2+\bar{x}_1\bar{x}_2)$可求得：

$A_g^{(0)}=\{v_{18},\bar{v}_{11}\}$，利用 $f_{V'}(v_8)=\bar{v}_{11}$作替代变换得图 1(b)的逻辑结构，进一步利用 $f_{V'}(v_8)=v_{18}$作逻辑结构变换得图 1(c)的逻辑结构.

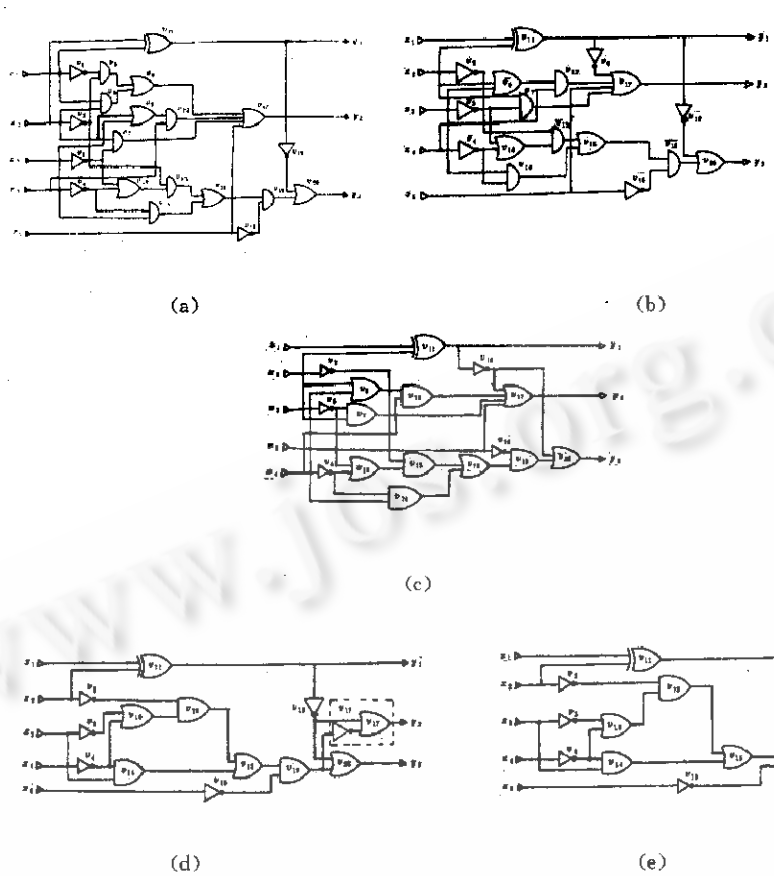

图 1 逻辑结点函数的优化替代与逻辑优化

由于 $fun\_list = g - A_g^{(0)} = \{v_5, v_6, v_1v_2, x_1x_2, \bar{x}_1\bar{x}_2\}$, $B_g^{(0)} = \varnothing$,可选取 v_5, v_6 之一作为构造 $f_{V'}(v)$的基,设 $table^{(0)} = \{v_5\}$,由递归调用可知 $A_g^{(1)} = \{v_6, x_1x_2\}$,故 $v_5 + v_6, v_5 + x_1x_2$ 可作为 $f_{IP}(v_8)$的可替代函数,但 $v_5 + v_6, v_5 + x_1x_2$ 作替代函数不能得到比图 1(c)更优化的逻辑结构.

考虑 $f_{IP}(v_{17})$的优化替代过程:

$$f_{IP}(v_{17}) = v_{18} + v_{12} + v_7 + x_3;$$

$$f_{PI}(v_{17}) = x_1x_2 + \bar{x}_1\bar{x}_2 + x_2x_4 + x_3x_4 + x_2\bar{x}_3 + x_5;$$

$$g = v_{17}^{ON} + v_{17}^{DC}$$

$$= \{x_5, v_{18}, v_{12}, v_7, \bar{v}_{11}, x_4v_9, x_2v_3, x_2\bar{x}_3, x_2x_4, x_3x_4, \bar{v}_{19}, \bar{v}_{15}, \bar{v}_{13}\bar{v}_{14}, \bar{v}_{15}, \bar{v}_4\bar{v}_{10}, \bar{x}_3\bar{v}_2, \bar{v}_3\bar{v}_4\}.$$

递归调用可求得: $A_g^{(0)} = \varnothing$, $B_g^{(0)} = \varnothing$

由 Step 8 选取 $g_i \in g$,不妨取 $g_i = \bar{v}_{19}$,可得:

$$h = f_{PI}(v_{17}) \# \bar{v}_{19} = x_1x_2 + \bar{x}_1\bar{x}_2;$$

递归调用可求得: $A_g^{(1)} = \{v_{18}, \bar{v}_{11}\}$

故 $f_{V'}(v_{17}) = v_{18} + \bar{v}_{19}$, $f_{V'}(v_{17}) = \bar{v}_{11} + \bar{v}_{19}$为 $f_{IP}(v_{17})$的可替代函数. 取 $f_{V'}(v_{17}) = v_{18} + \bar{v}_{19}$作替代变换得图 1(d)的逻辑结构,进一步取 $f_{V'}(v_{17}) = \bar{v}_{11} + \bar{v}_{19}$难于得到更优化的逻辑结构;进一步对图 1(a)作局部逻辑结构变换,可得图 1(e)的优化逻辑结构 G',此时 $cost(G') = 24$.

给定多级逻辑结构 G,可采用自 PI 开始的广度优化次序遍历 G 中的逻辑结点 $V \in V_g$,寻求 $f_{IP}(v)$ 的优化可替代函数 $f_{V'}(v)$,尽量利用已有的逻辑结点函数优化表示 $f_{IP}(v)$,降低 G 的逻辑复杂度.

4 结 论

多级逻辑结构中逻辑结点函数的优化覆盖问题可通过递归算法 OCP()求取,递归算法 OCP()在求取逻辑结点函数的优化覆盖时考虑了结点函数的优化替代对逻辑结构复杂度的影响,可求取逻辑复杂度极小化的多级逻辑结构,并可去除逻辑结构中存在的逻辑冗余;基于逻辑结点函数的优化覆盖算法的多级逻辑优化过程具有较好的逻辑函数优化及逻辑结构重构能力.

参考文献

1 Barlett K A, Brayton R K, Hachtel G D *et al*. Multilevel logic minimization using implicant don't cares. IEEE Trans., on: CAD. 1988,CAD-7(6):723～740.

2 Muroga S, Kambayashi Y, Lai H C *et al*. The transduction method-design of logic networks based on permissible functions. IEEE Trans., on: Computers, 1989,38(10):1404～1423.

3 Muroga S, Xiang X Q, Limqueco J *et al*. A logic network synthesis system: SYLON. Proc. of the Inter. Conf., on: Computer Design(ICCD-89), 1989. 329～333.

4 曾献君. 基于结构的 ASIC 逻辑综合技术研究(第二章)[博士论文]. 哈尔滨工业大学,1994 年.

5 Chen K C, Matsunage Y, Muroga S. A resynthesis approach for network optimization. Proc. of the 28th ACM/IEEE Design Automation Conf., 1991. 458～463.

6 刘明业. 计算机辅助逻辑综合理论. 北京:科学出版社,1985.

FINDING AN OPTIMAL COVER OF THE NODE'S FUNCTIONS

Ye Yizheng Zeng Xianjun Yu Mingyan

(*Microelectronics Center Harbin Institute of Technology Harbin* 150001)

Abstract This paper presents an algorithm to find an optimal cover of the logic node's functions in multilevel logic structure, and proves the efficiency of the algorithm in optimizing the multilevel logic structure.

Key words Multilevel logic synthesis, logic optimization, function covering.