

应用遗传算法进行低功耗状态编码

朱 宁,周润德,羊性滋

(清华大学微电子所,北京 100084)

摘 要: 本文研究了用遗传算法进行有限状态机(FSM)的低功耗状态编码问题. 基于 FSM 的概率模型,对编码空间进行并行搜索;通过在适应性度量中引入面积和状态翻转信息,实现了面积和功耗之间的折衷. 对一些 FSM 的实际测试表明此方法平均能达到 20% 的功耗优化.

关键词: 遗传算法;有限状态机;低功耗;编码;适应性

中图分类号: O157.4 **文献标识码:** A **文章编号:** 0372-2112 (2000) 08-0124-03

Low-power State Encoding Using Genetic Algorithm

ZHU Ning, ZHOU Run-de, YANG Xing-zi

(Institute of Microelectronics, Tsinghua University, Beijing 100084, China)

Abstract: This paper discusses about low-power state encoding of finite state machines(FSM s) using genetic algorithm(GA). Based upon the probabilistic model of FSM s, GA based parallel searching is implemented in the encoding space. Moreover, area and state transition information may be both included in the 'fitness' measurement to realize some trade-off between area and power. The test results for some FSM s show that this approach may lead to 20% power reduction.

Key words: genetic algorithm; finite state machine; low-power; encoding; fitness

1 引言

随着集成电路特征线宽的不断缩小,集成电路芯片的集成度和工作频率都得到了较大提高,但是芯片的功耗也不可避免地随之增加了. 各种便携式设备、高速处理器的设计者必须面对功耗优化的问题,这就使得对于低功耗综合工具的需求变得更加迫切起来^[1].

我们研究了对有限状态机(Finite-State-Machine,简称FSM)进行低功耗状态编码的问题. 对FSM进行状态编码属于NP问题. 如果假设FSM中的状态数为 N ,编码长度为 M ,则可能的编码方案为 P_{2M}^N 种. 由此可见随着状态数的增加,可能的编码方案将迅速增加,以至采用简单的穷尽搜索方法无法进行. 因此必须采用一些并行的搜索方法,迅速而有效地求得问题的最优解或接近最优解. 这里我们采用了遗传算法.

本文的其余部分安排如下:第二部分介绍FSM的概率模型;第三部分介绍遗传算法;第四部分提出将遗传算法应用于状态编码的算法;第五部分给出实验结果;最后得出结论.

2 FSM的概率模型

FSM一般以状态转移图(State Transfer Graph,简称STG)或者状态转移表(State Transfer Table,简称STT)的形式表示. 图1所示为一个两输入、两输出、包含四个状态的FSM,其STG和相应的STT表示分别如图1(a)和图1(b)表示(STT中的四列

分别代表当前输入,当前状态,下一状态和当前输出. 由于状态编码问题与FSM的初始状态和结束状态无关,因此图中没有特别标明). 当状态发生变化时,状态位的翻转会使电路产生功耗. 因此在对状态 $S_1 \sim S_4$ 进行编码时,如果使相互间翻转比较频繁的状态的编码也比较接近,就可以降低状态翻转时的电路开关行为,从而减少功耗. 为此必须首先计算状态之间的翻转概率.

一般地,一个FSM可以近似为一个Markov过程. 因此,STG中的状态转移过程可以用Markov链来描述. 设STG中由状态 S_i 到状态 S_j 的条件转移概率为 $p_{i,j}$,即

$$p_{i,j} = \text{Prob}(\text{Next state} = S_j | \text{Present state} = S_i) \quad (1)$$

且系统处于状态 S_i 的概率为 P_i ,则状态 S_i, S_j 之间的状态转移概率为:

$$P_{ij} = p_{i,j}P_i + p_{j,i}P_j \quad (2)$$

状态转移概率反映了电路在工作时状态之间相互跳转的频率. 在进行状态编码时,应使得 P_{ij} 较大的状态的编码尽量接近,这样就能减少由于状态翻转所导致的组合逻辑门的翻转,从而降低功耗.

$p_{i,j}$ 可以由STG或STT直接计算得到. 以 $p_{1,2}$ 为例,假定各个输入不相关且信号概率都为0.5,则 $p_{1,2} = P(01) + P(00) + P(10) = 3/4$. 状态概率 P_i 是由一步转移概率矩阵 P_T 得到的. P_T 的形式为:

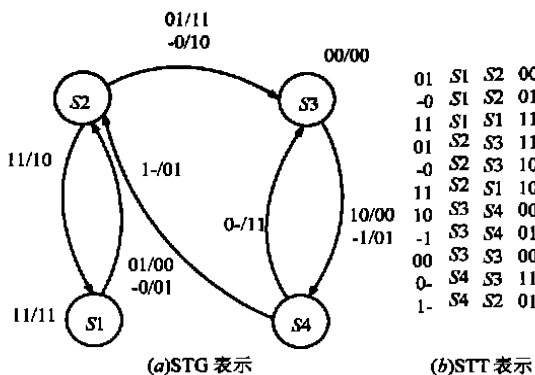


图 1 FSM 的 STG 和 STT 表示

$$P_T = \begin{bmatrix} P_{S_1, S_1} & \dots & P_{S_1, S_N} \\ \vdots & \ddots & \vdots \\ P_{S_N, S_1} & \dots & P_{S_N, S_N} \end{bmatrix} \quad (3)$$

对 P_T 连续自乘即可收敛至状态概率,其依据是如下的定理^[2]:

定理:一个不可约的、非周期的、有限状态的马尔可夫链一定是遍历的,并且存在极限分布 $\lim_{n \rightarrow \infty} P_T^n$, 其中 P_T 的各行相同,且 i, j 即为 FSM 处于状态 S_i 的概率 P_i . 由 P_T 矩阵计算的迭代公式如下:

$$P_T^{n+1} = P_T^n * P_T^n \quad (4)$$

图 2 为根据式(1), (2)计算得到的状态之间转移概率的加权图. 为计算方便,图中各条边上的权重都取了整数,其值正比于状态转移概率. 图 2 的加权无向图即为低功耗状态编码的起点.

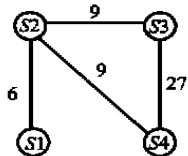


图 2 用于状态编码的加权图,各条边上的权重正比于总的状态转移概率

3 遗传算法介绍

遗传算法是一种通过模拟自然进化过程搜索最优解的方法,它具有全局优化和并行性两大特点^[3]. 在准备应用遗传算法求解问题时,要先完成以下四个主要步骤:(1)确定表示方案;(2)确定适应值度量;(3)确定控制算法的参数和变量;(4)确定指定结果的方法和停止运行的准则. 表示方案是指把问题的搜索空间中每个可能的点表示为确定长度的特征串. 二进制串是遗传算法中常用的表示方法. 适应值度量为群体中每个可能的确定长度的特征串指定一个适应值,它经常是问题本身所具有的. 适应值度量必须有能力计算搜索空间中每个确定长度的特征串的适应值. 控制遗传算法的主要参数有群体规模 N 和算法执行的最大代数 M ,次要参数有复制概率 p_r ,杂交概率 p_c 和变异概率 p_m 等参数.

停止准则有时表示成系统进化的最大代数,对那些最优解一出现就能识别的问题,算法可以在这样的个体找到时就停止执行.

一旦这些准备步骤完成,就可以执行遗传算法. 其主要步骤如下:(1)随机产生一个由特定长度的特征串组成的初始群

体,(2)对串群体迭代地执行下面的步骤(a)和(b),直到满足收敛准则:(a)计算群体中每个个体的适应值;(b)应用复制、杂交和变异算子产生下一代群体.(3)把在任一代中出现的最好的个体串指定为遗传算法的执行结果,这个结果可以表示问题的一个近似解.

4 用遗传算法进行 FSM 的低功耗状态编码

其步骤为:

(1) 确定个体的特征串表示:

对于状态编码而言,其个体即为一种编码方案,将各个状态的编码串联起来即构成了一个编码方案的特征串表示. 以图 1 的 FSM 为例,假定一种编码方案为 $S_0:00; S_1:11; S_2:10; S_3:01$,则“00111001”即为一个特征串. 初始群体的构成可以随机产生. 设 FSM 的状态数为 N ,编码码长为 M ,则产生一个特征串的过程如下:

```
for (i = 0; i < N; i++)
    generate M-bit random number c[i];
for (i = 1; i < N; i++)
    if (exists [0, i), s, t. c[i] = c[j]) c[i] = c[i] ⊕ 1;
```

(2) 适应值(fitness)计算:

如果仅从功耗的角度考虑,则个体的适应值可以表示为:

$$f_P = \left[\sum_{i,j=1}^N P_{i,j} * H(S_i, S_j) \right]^{-1} \quad (5)$$

其中 $P_{i,j}$:状态 S_i, S_j 之间发生状态转移的概率(见式(2)); $H(S_i, S_j)$:状态 S_i 与 S_j 的编码之间的汉明距离.

实际上由于 FSM 中组合逻辑部分的大小也与状态编码密切相关,而且此部分的功耗又很难预先估计,这就使得面向功耗优化的状态编码比直接面向面积优化的编码具有更高的复杂度. 即既要考虑编码对于 FSM 的功耗的影响,又要考虑其对于最终电路的组合逻辑模块规模的影响,否则很难达到预期的功耗优化效果. 为此在适应值公式中还应加上面积估计因子. 修正后的适应值为:

$$f = f_P + (1 - \alpha) f_A \quad (6)$$

其中 $0 \leq \alpha \leq 1$ 为可调的经验参数.

(3) 复制、杂交和变异过程

复制过程采用轮盘赌方法,适应值大的个体在下一代中出现的也多;杂交过程在两个特征串之间进行子串的交流,它可以产生具有更高适应值的个体;变异过程则是改变串中的某些位,它具有增加群体多样性的效果. 为了使这些过程更具有普遍性,复制、杂交和变异都按照一定的概率进行. 在每次复制、杂交和变异后都要对特征串进行调整,以使各个串中对状态的编码没有重复. 调整的方法与在生成初始个体特征串时的相同.

(4) 确定算法控制参数和停止准则

如前所述,需要确定的算法参数有群体规模,演化代数以及复制、杂交、变异概率. 一般地说,对于规模在 50 到 100 的群体,经过 10 代到 20 代的演化,遗传算法都以很高的概率找到最优或近似最优解. 另外,变异概率只要在 0.001 的数量级,就足以能防止搜索陷入局部最优^[3].

5 实验结果

我们实现了相应的用遗传算法进行低功耗状态编码的软件 GENECODE,其步骤如下:(1)输入 FSM 的 STT 表示,计算出 P_T , $P_{i,j}$ 和 P_i ;(2)应用遗传算法,根据式(6)中的适应值度量进行状态编码,选出适应值最大的编码方案,它即为最后的编码结果。

我们用 GENECODE 对一些 FSM 进行了测试,其中群体规模定为 100,演化代数 50,复制概率为 0.8,交配概率为 0.6,变异概率为 0.001。为 0.8。编码长度取为最小比特数。

为了检验状态编码的效果,用 SYNOPSIS 的 DesignCompiler 逻辑综合工具对完成状态编码后的 FSM 进行了逻辑综合,然后统计综合后的电路面积和功耗。为了进行比较,还用随

机生成的方法 (RANDOM) 和 JEDI 状态编码算法对这些 FSM 进行了状态编码和逻辑综合,并且同样得到最终电路的功耗和面积。实验所用硬件为 ULTRASPAC2 工作站,主频为 143MHz,内存为 64M。

表 1 列出了 GENECODE、JEDI 和 RANDOM 三种状态编码的实现结果(其中 RANDOM 的结果为三次随机编码中的最优值)。表中面积计算的单位为一个标准反相器的面积,功耗计算时假定电源电压为 5V,时钟频率为 20MHz。

由表中可看出,GENECODE 方法与 RANDOM 方法相比,平均面积改善了近 10%,功耗改善了近 20%。与 JEDI 的结果相比,平均面积几乎没有增加,而平均功耗改善了 6.79%。而且所花费的 CPU 时间也是可以接受的。这说明了应用遗传算法进行低功耗状态编码的有效性和应用潜力。

表 1 GENECODE 的结果与 JEDI 和 RANDOM 的结果比较

	GENECODE		RANDOM		积 改善 %	功耗 改善 %	JEDI		面积 改善 %	功耗 改善 %
	$P(\mu w)$	A	$P(\mu w)$	A			$P(\mu w)$	A		
s208	1.60	126	1.96	128	1.56	18.4	1.70	130	3.08	5.88
s386	2.40	154	3.08	170	9.41	22.1	2.30	157	1.91	- 4.34
s820	3.66	311	4.53	329	5.47	19.2	4.07	295	- 5.42	10.1
s1494	4.00	578	5.78	682	15.2	30.8	4.90	579	0.17	18.4
dk14	2.17	128	2.43	141	9.22	10.7	2.36	128	0.00	8.05
dk15	1.74	88	1.62	85	- 3.53	- 7.4	11.74	88	0.00	0.00
bbara	1.09	88	1.61	110	20.0	32.3	1.25	93	5.38	12.8
sse	1.96	128	3.25	167	23.4	39.7	1.93	135	5.19	- 1.55
planet	9.36	550	11.38	599	8.18	17.8	10.88	496	- 10.9	14.0
s420	1.51	129	1.92	132	2.27	21.4	1.57	120	- 7.50	3.82
tma	3.16	169	4.75	206	18.0	33.5	3.47	173	2.31	8.93
sand	8.18	511	8.63	565	9.56	5.21	8.65	507	- 0.79	5.43
s298	33.8	2961	37.9	3172	6.65	10.8	-	-	-	-
面积、功耗改善平均值 %					9.65	19.6			- 0.55	6.79

6 结论

本文应用遗传算法进行 FSM 的低功耗状态编码,并实现了相应的状态编码软件 GENECODE。对一些 FSM 的编码结果表明此方法具有较好的效果和实际应用潜力。

参考文献:

- [1] A. Chandrakasan, S. Sheng, and R. Broderson. Low-power CMOS digital design [J]. IEEE J. Solid-State Circ., April 1992, 27(4): 473 - 484.
- [2] 陆大. 随机过程及其应用 [M]. 北京:清华大学出版社, 1986.
- [3] 刘勇, 康立山, 陈毓屏. 非数值并行算法-遗传算法 [M]. 北京:科学出版社, 1997.

作者简介:



朱 宁 1994 年毕业于清华大学电子工程系,获学士学位。现在清华大学微电子所攻读博士学位,主要研究方向为 CMOS 数字电路的低功耗设计方法,以及嵌入式系统设计。



周润德 1968 年毕业于清华大学自动化系。1982 年和 1986 年于美国波士顿大学分别获得硕士学位和博士学位。现为清华大学微电子所教授,博士生导师。主要研究方向为微处理器和微控制器设计, RISC 体系结构,嵌入式系统设计以及低功耗电路设计。