

峰值功耗优化的力引导调度算法

孙 强

(牡丹江师范学院计算机科学与技术系 黑龙江 牡丹江 157012)

【摘要】提出了基于单周期操作和多周期操作的峰值功耗优化的力引导调度算法。该算法运用传统力引导调度算法的基本思想,通过对力相关参数的重新设置,实现了调度过程对周期功耗的平衡分布,从而达到了峰值功耗最小化的目标。实验结果显示,在控制步数目和资源数目相同的情况下,该算法在峰值功耗优化方面比传统的力引导调度算法有所改进,与基于整数线性规划的算法基本相当。

关键词 功耗控制; 峰值功耗; 综合; 调度算法

中图分类号 TP302.1

文献标识码 A

doi:10.3969/j.issn.1001-0548.2010.01.031

Force-Directed Scheduling Algorithm for Optimized Peak Power

SUN Qiang

(Department of Computer Science and Technology, Mudanjiang Normal University Mudanjiang Heilongjiang 157012)

Abstract A force-directed scheduling algorithm for optimized peak power is proposed based on single cycle operation and multi-cycle operation. The balanced distribution of cycle power in the schedule process is realized through utilizing the basic idea of traditional force-directed scheduling algorithm and resetting the related force parameters, thus the goal of minimum peak power is achieved. Experimental results show that with the same number of control steps and resources, the algorithm proposed in this paper achieves performance in peak power optimization better than traditional force-directed scheduling algorithms, and equivalent to integer linear programming methods.

Key words power control; peak power; synthesis; scheduling algorithms

集成电路集成度和时钟频率的不断提高,使得功耗成为电路设计中日渐突出的问题^[1-3]。当前,国内外对于高层次功耗优化问题有很多研究。文献[4]提出了一种高层次综合中基于整数线性规划的算法,并对峰值周期功耗和峰值模块功耗进行优化。文献[5]通过调度和互联两种手段降低功耗。文献[6]采用多电压方法降低功耗,利用分区方法减小连线复杂性。

传统的功耗优化方法力求降低电路的总功耗,而现在越来越多的研究把功耗优化目标转移到对电路峰值功耗和平均功耗的优化。电路的峰值功耗就是指电路调度的所有控制步中周期功耗的最大值。文献[7]提出了力引导(force-directed scheduling, FDS)调度算法,该算法通过平衡操作并发性实现功能单元数目最小化目标。最近有研究对FDS进行改进,实现周期功耗优化的目标。文献[8]提出一种启发式的力引导调度算法实现峰值功耗最小化。文献[9]通

过力引导调度算法降低模块内部的开关功耗,实现动态功耗的优化。文献[10]运用改进的力引导调度算法和再调度算法实现峰值功耗和平均功耗的优化。

本文通过对传统的力引导调度算法和现有功耗优化的力引导调度算法的研究,提出了基于单周期和多周期操作的峰值功耗优化的力引导调度算法。该算法运用传统力引导调度算法的基本思想,通过对力相关参数的重新设置,实现调度过程周期功耗的平衡分布,并考虑每次调度导致的功耗对全局的影响。

1 基本的力引导调度算法

FDS算法是一种构造型、基于时间约束的调度算法,其基本思想是尽可能均匀地在硬件功能单元之间安排操作,提高功能单元的总使用效率。该算法分为3步^[7]:

(1) 根据ASAP和ALAP算法确定某个操作 v 的

灵活度:

$$\text{Mobility}(v) = \text{ALAP}(v) - \text{ASAP}(v) \quad (1)$$

假设所有操作都是单控制步操作, 倒数为该操作 v 在当前控制步 i 的执行概率:

$$\text{Pro}(v, i) = \frac{1}{\text{Mobility}(v)} \quad (2)$$

(2) 确定每一控制步中每一种类型操作的执行概率之和:

$$\text{DG}_k(i) = \sum_{v \in k} \text{Pro}(v, i) \quad (3)$$

概率分布图表示每一种类型操作的并行性。

(3) 力引导调度算法迭代计算每一个操作分配到某一个控制步的“力”, 选择最小力的操作控制步作为当前迭代的结果保存, 每次迭代只完成一个操作的调度, 直到所有操作调度完成。力($\text{Force}(v, i)$)计算包括自身力($\text{Self}(v, i)$)和间接力($\text{Ps}(v, i)$), 自身力为当前操作在不同控制步调度对并发性的影响, 间接力是当前操作的调度结果对其前导和后继操作调度的控制步造成的影响。各种力的计算公式如下:

$$\text{Force}(v, i) = \text{Self}(v, i) + \text{Ps}(v, i) \quad (4)$$

$$\text{Self}(v, i) = \sum_{j=\text{ASAP}(v)}^{\text{ALAP}(v)} (\text{DG}_k(j) \cdot X(v, i, j)) \quad (5)$$

$$\text{Ps}(v, i) = \sum_{w \in \text{ps}(v)} \frac{\sum_{l=\text{newASAP}(w)}^{\text{newALAP}(w)} \text{Self}(w, l)}{\text{newALAP}(w) - \text{newASAP}(w) + 1} \quad (6)$$

式中 $X(v, i, j)$ 表示当操作 v 被调度到控制步 i 时, 控制步 j 中类型为 k 的操作概率的变化值; $\text{newASAP}(w)$ 和 $\text{newALAP}(w)$ 表示当操作 v 被调度到控制步 i 时, 导致前导和后继灵活度的变化。

2 峰值功耗优化的力引导调度算法

2.1 单周期操作峰值功耗优化的力引导调度算法

算法迭代进行, 每一次迭代完成一个操作的调度, 操作对控制步的选择基于峰值功耗平衡的前提进行。计算力的目的是为了在所有控制步中平衡峰值功耗, 并考虑每次调度对全局峰值功耗的影响。力分为两部分: (1) 自身力考虑一个操作 v 调度到控制步 i 时, 对峰值功耗的影响; (2) 前导后继力考虑当操作 v 调度到控制步 i 时, 改变其控制步所导致的峰值功耗影响。

峰值功耗优化的力引导调度算法的操作步骤为:

(1) 应用式(1)计算每个节点的灵活度, 再运用

式(2)计算每一个操作在其灵活度内的执行概率。

(2) 建立操作功耗概率分布图(本文不区分操作的类型), 每个控制步的功耗概率分布为该控制步上所有操作执行的功耗与其执行概率乘积的和, 每一个周期的功耗概率分布为:

$$\text{DG}_{\text{power}}(i) = \sum_{v \in C(i)} (\text{Pro}(v, i) \cdot \text{Power}(v)) \quad (7)$$

(3) 运用峰值功耗相关的力完成候选操作到指定控制步的调度。由于功耗大的类型操作对功耗概率分布和峰值功耗平衡影响最大, 首先对功耗最大类型的操作节点进行调度, 并计算峰值功耗相关的力。力也是由自身力和前导后继力构成, 其计算公式为:

$$\text{Force}_{\text{sp}}(v, i) = \text{Self}_{\text{sp}}(v, i) + \text{Ps}_{\text{sp}}(v, i) \quad (8)$$

$$\text{Self}_{\text{sp}}(v, i) = \sum_{j=\text{ASAP}(v)}^{\text{ALAP}(v)} (\text{DG}_{\text{power}}(j) \cdot X(v, i, j)) \quad (9)$$

$$\text{Ps}_{\text{sp}}(v, i) = \sum_{w \in \text{ps}(v)} \frac{\sum_{l=\text{newASAP}(w)}^{\text{newALAP}(w)} \text{Self}_{\text{sp}}(w, l)}{\text{newALAP}(w) - \text{newASAP}(w) + 1} \quad (10)$$

(4) 选择一个最小力的操作, 完成该操作到控制步调度。如果存在未被调度节点, 返回第一步, 否则结束。

假设以DiffEq作为调度对象, 调度的总控制步为4, 给出单周期操作峰值功耗优化力引导调度的实例。各功能模块在5 V电压下的延迟、能耗和功耗如表1所示。

表1 功能模块在5 V电压下的延迟、能耗和功耗

模块	延迟/ns	功耗/mW
Mult16	100	25.04
Add16, Sub16	20	9.05

单周期操作调度控制步概率分布如图1所示。首先计算出一个初始调度控制步概率分布, 如图1a所示。根据图1a中各控制步操作的分布概率和表1中各模块的功耗, 运用式(7)计算出各控制步的功耗概率分布, 如图2所示。由于乘法操作的功耗最大, 所以首先对所有乘法操作的力进行计算, 完成它们的调度。力的计算结果如表2所示。从该表中可以看出, 将*4调度到控制步3的力最小, 因此将*4调度到控制步3。此外, 由于+9是*4的后继操作, 所以+9只能调度到控制步4。

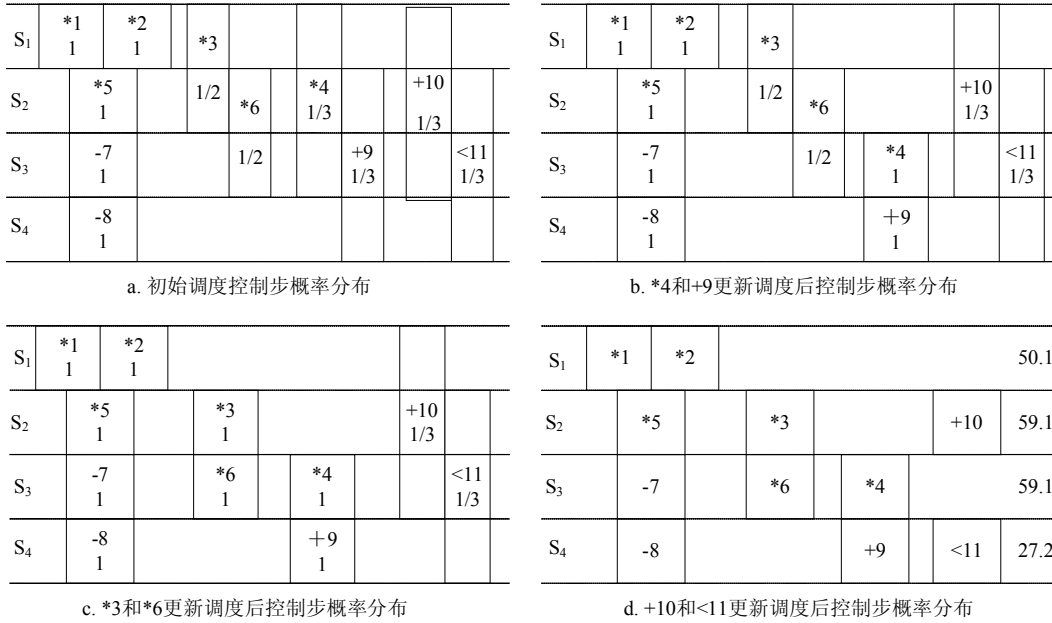


图1 单周期操作调度控制步概率分布图

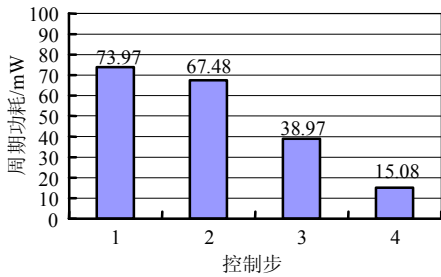


图2 单周期操作功耗概率分布图

表2 所有乘法操作的力计算结果

调度操作	自身力	前导后继力	总的力
(*3, 1)	3.25	0	3.25
(*3, 2)	-3.25	-14.26	-17.51
(*6, 2)	14.26	3.25	17.51
(*6, 3)	-14.26	0	-14.26
(*4, 1)	13.83	0	13.83
(*4, 2)	7.34	-13.49	-6.15
(*4, 3)	-21.17	-25.43	-46.60

表3 *3和*6操作的力计算结果

调度操作	自身力	前导后继力	总的力
(*3, 1)	4.76	0	4.76
(*3, 2)	-4.76	-1.74	-6.50
(*6, 2)	1.74	4.76	6.50
(*6, 3)	-1.74	0	-1.74

表4 +10和<11操作的力计算结果

调度操作	自身力	前导后继力	总的力
(+10, 1)	-5.02	0	-5.02
(+10, 2)	-2.01	-4.32	-6.33
(+10, 3)	7.04	-26.34	-19.30
(<11, 2)	8.65	-5.02	3.63
(<11, 3)	17.7	-3.52	14.18
(<11, 4)	-26.34	0	-26.34

更新的调度控制步概率分布如图1b所示。继续对剩余的乘法操作进行调度，将*3操作调度到控制步2的力最小，如表3所示。因此，应将*3调度到控制步2。此外，由于*6是*3的后继操作，并且根据*6的灵活度约束，只能将*6调度到控制步3。完成*3和*6调度后的更新调度控制步概率分布如图1c所示。由于此时所有的乘法操作均已调度完成，需对+10和<11进行调度。将<11操作调度到控制步4的力最小，如表4所示。所以将<11调度到控制步4。最后只剩下+10时，该操作可以调度到控制步1和控制步2(本文中，将+10调度到控制步2)。最终的单周期操作调度结果和周期功耗分布如图1d所示。

2.2 多周期操作峰值功耗优化的力引导调度算法

将峰值功耗优化力引导调度算法扩展到存在多周期操作的调度过程，算法操作步骤如下。

(1) 用式(1)计算每个节点的灵活度，对于一个执行周期为*c*的操作，执行概率为：

$$\text{Pro}(v, i) = \frac{c}{\text{Mobility}(v)} \quad (11)$$

(2) 建立操作功耗概率分布图，每个控制步的功耗概率分布为该控制步上所有操作执行的功耗与其执行概率乘积的和，用式(7)计算每一个周期功耗概率分布 $DG_{\text{power}}(i)$ 。

(3) 运用峰值功耗相关的力完成候选操作到指定控制步的调度。对每个操作节点计算其与峰值功

耗相关的力,此时的力也是由自身力和前导后继力构成。需要探讨的是多周期操作的调度,一个多周期操作要被调度到一组连续的控制步,需要的控制步数目为多周期操作的执行周期数 c 。对于一个多周期操作可以被调度到控制步组的每一种可能,自身力的计算公式为:

$$\text{Self}_{\text{mp}}(v,i) = \sum_j (\text{DG}_{\text{avgpower}}(j) \cdot X(v,i,j)) \quad (12)$$

式中 j 为多周期操作在其灵活度范围内所有合法的候选调度控制步组; $\text{DG}_{\text{avgpower}}(j)$ 为在一个候选调度控制步组 j 中所有控制步的功耗概率分布 $\text{DG}_{\text{power}}(i)$ 的平均值; $X(v,i,j)$ 为当操作 v 被调度到控制步组 i 时,控制步组 j 中操作概率的变化值。多周期操作调度导致前导后继力的计算公式为:

$$\text{Ps}_{\text{mp}}(v,i) = \sum_{w \in \text{ps}(v)} \frac{\sum_l \text{Self}_{\text{mp}}(w,l)}{\text{newALAP}(w) - \text{newASAP}(w) + 1} \quad (13)$$

式中 l 表示在操作 v 的调度到控制步组 i 后,前导后继操作在新的灵活度范围内的所有合法的候选调度控制步组。

(4) 选择一个最小力的操作,完成该操作的调度。如果存在未被调度的节点,返回步骤(1),否则结束。

3 实验结果

通过实验验证本文方法的有效性,实验过程中选择了4个基准电路。假设在单周期调度中所有操作都是单控制步操作,在多周期调度中乘法操作是两个控制步操作,加法、减法和比较操作都是单控制步操作。所有操作的供应电压都为5V。

实验中,在控制步约束和资源消耗相同的情况下,将本文方法(MFDS)调度结果的峰值功耗与传统的力引导调度算法(FDS)和文献[11]的ILP方法进行比较。

(1) 单周期操作峰值功耗的比较如表5所示。本文方法比FDS算法在峰值功耗上的平均改善程度为12.95%,与ILP方法基本相当。

(2) 多周期操作峰值功耗的比较如表6所示。本文方法比FDS算法在峰值功耗上的平均改善程度为9.78%,与ILP方法相当。

从实验结果可以看出,本文所提出的算法在峰值功耗优化方面比传统的FDS算法有所改进,与ILP

方法基本相当。

表5 单周期操作调度峰值功耗比较

基准电路	控制步数	资源约束		峰值功耗/mW			改善程度(%)	
		*	+	FDS	MFDS	ILP	MFDS	ILP
DiffEq	4	2	2	59.13	59.13	59.13	0	0
	5	2	2	59.13	50.08	50.08	15.31	15.31
Real	6	1	2	34.09	34.09	34.09	0	0
	8	1	2	34.09	25.04	25.04	26.55	26.55
EWF	16	2	3	77.23	68.18	59.13	11.71	23.44
	17	2	2	68.18	43.14	43.14	36.73	36.73
AR-Filter	10	2	2	68.18	68.18	59.13	0	13.27
	11	2	2	68.18	59.13	59.13	13.27	13.27

表6 多周期操作调度峰值功耗比较

基准电路	控制步数	资源约束		峰值功耗/mW/			改善程度(%)	
		*	+	FDS	MFDS	ILP	MFDS	ILP
DiffEq	6	3	3	75.12	75.12	75.12	0	0
	8	2	2	59.13	50.08	50.08	15.31	15.31
	10	2	1	34.09	25.04	25.04	26.55	26.55
Real	9	1	2	34.09	34.09	34.09	0	0
	13	1	2	34.09	25.04	25.04	26.55	26.55
EWF	17	2	3	77.23	77.23	77.23	0	0
	19	2	2	68.18	68.18	68.18	0	0
AR-Filter	15	3	2	93.22	84.17	84.17	9.71	9.71
	18	2	1	59.13	59.13	59.13	0	0

4 结束语

传统的力引导调度算法的思想是通过平衡操作的并发性,实现功能单元的最小化。本文通过对传统的力引导调度算法的研究,提出基于单周期操作和多周期操作的峰值功耗优化的力引导调度算法,利用力引导调度算方法的思想,通过对力相关参数的重新设置,实现调度过程对周期功耗的平衡以及峰值功耗最小化目标。

本文研究工作得到牡丹江师范学院博士科研启动基金(MSB200901)资助,在此表示感谢。

参考文献

- [1] MEI F C, PHON S, ALIAS M Y. Adaptive GA: An essential ingredient in high-level synthesis[C]//IEEE Congress on Evolutionary Computation. Hong Kong: IEEE Press, 2008: 3837-3844.
- [2] WANG F, SUN G Y, XIE Y. A variation aware high level synthesis framework[C]//The Conference on Design, Automation and Test in Europe. Munich: ACM Press, 2008: 1063-1068.

(下转第148页)