

# Simultaneous Partitioning and Scheduling Algorithm for Clustered Architecture

Wang Lei and Wei Shaojun

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

**Abstract:** Clustered architecture is selected for high level synthesis, and a simultaneous partitioning and scheduling algorithm are proposed. Compared with traditional methods, circuit performance can be improved. Experiments show the efficiency of the method.

**Key words:** high level synthesis; scheduling; partitioning; clustered architecture

**EEACC:** 1130B; 1265A

**CLC number:** TN402

**Document code:** A

**Article ID:** 0253-4177(2004)04-0383-05

## 1 Introduction

DSM (deep submicron) makes chip designers face many new problems, among which are design complexity and interconnect effects. With feature size shrinking, more and more transistors can be integrated into a single chip, and earlier design space explorations become indispensable. On another aspect, wire width shrinks and routing density increases, physical effects bring on great mismatch on circuit performance between physical implementation and prediction in high level synthesis. How to avoid design iteration is really a challenge.

Besides researching on pure high level<sup>[1]</sup> or physical level<sup>[2]</sup>, lots of researches have been done in exploring simultaneous high level synthesis and physical design. Fang *et al.* introduced floorplan information to influence synthesis decision<sup>[3]</sup>. Pedram proposed incremental synthesis method<sup>[4]</sup>. All above adopted traditional register transfer architecture where global wire delays still contribute high percentage to clock period.

An in-depth research of design methodology in DSM era was made by Sylvester<sup>[5]</sup>, with the help of extensive experiments, conclusion can be drawn that even when pessimistic noise consideration were introduced, design method and tools can be evolve to make precise timing analysis for systems less than 50k gates. Hierarchical method is a promising and in future entire system consists of macros or subsystems in such size. With the similar point of view, clustered architecture has been introduced in ASIP design. Clock periods of clustered architecture are much less than that of non-clustered architecture with same number of functional units and registers<sup>[6]</sup>.

In our research, clustered architecture is selected for high level synthesis. Datapath is composed of several clusters and global interconnect delays are eliminated from critical path in clock period. Different from ASIP design, there is no pre-determined architecture and resources in high level synthesis and every cluster can have its distinctive configuration. Designer can explore hardware resource capability to the best in a custom design. We

Wang Lei male, was born in 1976, PhD candidate. His research interests are high level synthesis for digital systems.

Wei Shaojun male, was born in 1958, professor. His research interests are EDA methodology and communication ASIC design.

Received 31 July 2003, revised manuscript received 20 November 2003

©2004 The Chinese Institute of Electronics

give formal description of the synthesis tasks and propose a heuristic algorithm. Experiment results show efficiency of this method.

## 2 Simultaneous partition and schedule algorithm

Figure 1 is the RTL (register transfer level) architecture for synthesis in our approach. First, according to technology parameters, maximum size of cluster is determined so that it is possible to make precise timing analysis within a cluster with consideration of wire delays. Second, after floor-planning clusters, number of clock cycles for communication between clusters are extracted, it will be the additional data dependency constraints for data transfers between operators that are not partitioned into the same cluster. Then, operators are scheduled to control steps and partitioned into clusters simultaneously to minimize total steps of control.

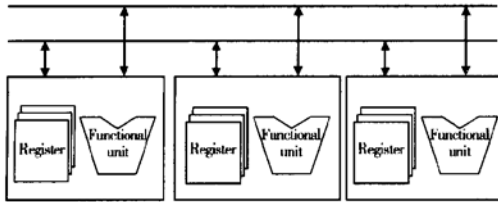


Fig. 1 Object architecture in our approach

Table 1 Symbols

$G(V, E)$	DFG (data flow graph)
$V$	Set of operators
$E$	Set of data transfers
$TN$	Number of resource types
$AP(i)$	Area of the $i$ th resource
$AT_{max}$	Area constraint
$MS$	Set of clusters
$P$	Number of clusters
$AM(i)$	Area of the $i$ th cluster
$AM_{max}$	Area constraint for cluster
$VS_i$	Set of operators in the $i$ th cluster
$cs(v)$	Control step for operator $v$
$CS$	Total number of control steps
$ms(v)$	Cluster operator $v$ belong to
$gc(i)$	Global communication in control step $i$
$CP$	Number of clock cycles for inter-cluster communication

Symbols used in this paper are listed in Table 1.

We define the simultaneous partitioning and scheduling problem below.

The input is DFG (data flow graph) and area constraints, including total area constraint  $AT_{max}$ , area constraints for cluster  $AM_{max}$ . Data transfers between clusters are realized through multi-cycle global communication mechanism,  $CP$  is determined by technology and the scale of the design. The outputs are the scheduling results and partition results.

Input:  $G, AT_{max}, AM_{max}$

Output:  $cs(v), \forall v \in V, MS, VS_i, i = 1, 2, \dots, P$

Minimize:  $CS$

Constraints:

$$cs(v_j) - cs(v_i) \geq 1, \text{ if } (v_i, v_j) \in E \text{ and } ms(v_i) = ms(v_j) \quad (1)$$

$$cs(v_j) - cs(v_i) \geq CP + 1, \text{ if } (v_i, v_j) \in E \text{ and } ms(v_i) \neq ms(v_j)$$

$$\sum_{i=1}^P AP(i) \leq AT_{max} \quad (2)$$

$$AP(i) \leq AP_{max}, \forall 1 \leq i \leq P \quad (3)$$

$$\bigcup_{i=1, 2, \dots, P} VS_i = V \quad (4)$$

$$VS_i \cap VS_j = \emptyset, \forall 1 \leq i, j \leq P, i \neq j \quad (5)$$

Equation (1) presents data dependency constraints, if  $v_i$  and  $v_j$  do not belong to the same cluster, the difference of control step of them should be long enough for global data transfer completion. Equations (2) and (3) present area constraints, total area and cluster area respectively. Equation (4) as well as Eq. (5) presents constraints for partition. It means that each operator should be partitioned to one and only one cluster.

For ILP (integer linear program) formulation above is NP problem. In practice, we adopt heuristic algorithm to solve the problem. Following general modeling methods of high level synthesis, area constraints are expressed as constraints for number of functional units. The key procedure is to construct  $r$ -list and  $c$ -list to preferentially schedule operators in critical path and localize data transfer as much as possible.

---

```

1. Control step  $t = 1$ ;
2. while(there exist unprocessed operators)
{
3. Construct priority list  $r\_list$  of ready operator;
4. while(there exist unprocessed operator in  $r\_list$ )
{
5. Current operator is set to head elements in  $r\_list$ ;
6. Construct priority list  $c\_list$  of clusters for current operator;
7. Search  $c\_list$  orderly to find cluster with available resource
   for current operator. if success, schedule it to control step  $t$ 
   and partition it to the current cluster, if fail, continue search
    $c\_list$  until to the end;
8. current operator is removed from  $r\_list$ ;
}
9.  $t = t + 1$ ;
}

```

---

$r\_list$  is a set of operators that are ready for current control step. Notice should be taken that if the operator  $op_i$  has predecessors whose scheduled step is more than  $(t - CP - 1)$ , it is in  $r\_list$  only on condition that all these predecessors have been partitioned into the same cluster. Because if it is not true, there is no enough clock cycles for inter-cluster communication. The priority used in  $r\_list$  construction is computed according to:

$$prio(op_i) = asapMaxTime(G) - alap(op_i)$$

$asapMaxTime(G)$  is the total control steps of  $G$  in ASAP scheduling and  $alap(op_i)$  is the control step of  $op_i$  in ALAP scheduling. The higher  $prio(op_i)$  is, the more urgent  $op_i$  should be scheduled because at least  $prio(op_i)$  steps are needed for completing all operations in  $G$  even if  $op_i$  is scheduled to current control step. Priory function for  $c\_list$  construction can be mathematically expressed as

$$prio(c_i) = s_1 \times succ(OP\_SET1(c_i)) - s_2 \times test(OP\_SET2(c_i)) + s_3 \times inter(c_i, op_i)$$

$OP\_SET1(c_i)$  is a set of scheduled operators in  $r\_list$  which have already been partitioned into  $c_i$ .  $OP\_SET2(c_i)$  is unscheduled operators in  $r\_list$  whose only feasible cluster are  $c_i$ .  $succ$  will encourage to partition operators with common successors into the same cluster.  $test$  returns 0 if  $OP\_SET2$  is empty, else returns 1. In order to schedule more operators in current step, clusters which are feasible for operators in  $OP\_SET2$  are not preferable

to be selected for current operator  $op_i$ .  $inter$  returns the number of data transfers between current operator and operators previously partitioned into  $c_i$ . This item encourages allocating more data transfer locally.  $s_1$ ,  $s_2$ , and  $s_3$  are weighed coefficients, in practice,  $s_1 > s_2 > s_3$ .

### 3 Experiments and analysis

Compared with traditional architecture for high level synthesis, inter-cluster communication puts more constraints for scheduling. There will be more cycles for scheduling DFG. But clock period could be reduced greatly, for example in Ref. [4], over 50% of period could be reduced. The main tasks of methods based on clustered architecture are to have inter-cluster communication in uncritical path of the DFG and minimize additional number of cycles. We take EWF (ellipse wave filter) and DCT (discrete cosine transform) benchmark in experiments. Figures 2 and 3 are results of control steps versus resource for EWF and DCT respectively. Non-cluster presents the traditional architecture and clustered presents architecture proposed in this paper with parameters {Number of cluster = 2, CP = 1}. Resource configurations sampled throughout design space, from minimum resource requirement to that more resources will not reduce control steps due to the original data dependency constraints. Table 2 gives the details, (M, A) means the number of multipliers and adders in one cluster. The additional control steps are controlled within 12% in all circumstances and in many cases even zero.

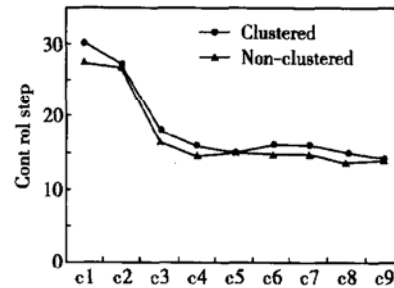


Fig. 2 Experiment results for EWF

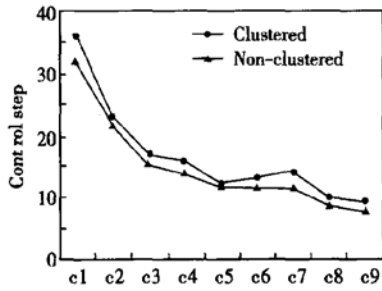


Fig. 3 Experiment results for DCT

Table 2 Resource configurations for EWF and DCT experiments

	(M, A) : (M, A)
c1	(1, 0) : (0, 1)
c2	(1, 1) : (0, 1)
c3	(1, 1) : (1, 1)
c4	(1, 2) : (1, 1)
c5	(2, 2) : (1, 1)
c6	(1, 2) : (2, 1)
c7	(1, 2) : (1, 2)
c8	(2, 2) : (2, 2)
c9	(3, 3) : (3, 3)

Next, we compare our methods with PARAS<sup>[7]</sup>. PARAS is a concurrent partitioner and scheduler for multi-ASICs. Limitations are that the resource configuration of different ASICs should be the same and the only feasible number of partitions is two. In order to compare the results, we take the benchmarks and resource configurations used in Ref. [5], multiplier occupies two control steps in this experiments. Table 3 shows that our algorithm

Table 3 Experiments comparing with PARAS

Benchmark	(M, A) : (M, A)	PARAS	OURS
ARF	(1, 1) : (1, 1)	19	18
	(2, 1) : (2, 1)	14	14
	(3, 1) : (3, 1)	14	14
EWF	(1, 1) : (1, 1)	19	19
	(1, 2) : (1, 2)	18	18
	(2, 2) : (2, 2)	17	17
FIR	(1, 1) : (1, 1)	12	12
	(1, 2) : (1, 2)	12	11
	(2, 2) : (2, 2)	11	11

could get the same or better results in the scope, so Reference [7] could take effect. Actually, our method is more flexible. Clusters could have their distinctive resource configurations and number of partitions could be any one according to the application.

## 4 Conclusion

In this paper, clustered architecture is selected for high level synthesis. A formal description of simultaneous partitioning and scheduling algorithm is proposed and an efficient heuristic algorithm is formulated in details. Compared with traditional methods, clock period is reduced while additional number of cycles could be well controlled. As a result, circuit performance could be improved. Experiments show the efficiency of our method.

## References

- [1] Zhou Haifeng, Lin Zhenghui, Cao Wei. Graph clustering algorithm for RT level ALU technology mapping. Chinese Journal of Semiconductors, 2002, 23(11): 1162
- [2] Li Zhuoyuan, Wu Weimin, Hong Xianlong, et al. Incremental placement algorithm for standard-cell layout. Chinese Journal of Semiconductors, 2002, 23(12): 1338
- [3] Fang Y M, Wong D F. Simultaneous functional unit binding and floorplanning. IEEE Trans ICCAD, 1994: 317
- [4] Pedram M. Logical-physical co-design for deep sub-micron circuits: Challenges and solutions. ASPDAC, 1998: 137
- [5] Sylvester D, Keutzer K. Getting to the bottom of deep submicron. ICCAD, 1998: 203
- [6] Sanchez J, Gonzalez A. Instruction scheduling for clustered VLIW architectures. Proceedings of the 13th Conference on International Symposium on System Synthesis, 2000: 41
- [7] Wong W H, Jain R. PARAS: System-level concurrent partitioning and scheduling. IEEE/ACM International Conference on Computer-Aided Design, 1995: 440

## 一种同时实现算子调度与数据流图划分的高层次综合算法

王 磊 魏少军

(清华大学微电子学研究所, 北京 100084)

**摘要:** 选择分模块的数据通道作为高层次综合的目标结构, 完整地定义了同时实现算子调度和数据流图划分的高层次综合算法, 并提出一种有效的启发式求解方法. 与传统的结构相比, 由于在关键路径中消除了全局连线的延时, 分模块的结构可以有效地减小时钟周期、优化电路性能. 实验结果验证了该方法的有效性.

**关键词:** 高层次综合; 调度; 划分; 分模块结构

**EEACC:** 1130B; 1265A

**中图分类号:** TN 402

**文献标识码:** A

**文章编号:** 0253-4177(2004)04-0383-05

---

王 磊 男, 1976 年出生, 博士研究生, 研究领域是深亚微米集成电路设计方法学和数字系统的高层次综合方法.

魏少军 男, 1958 年生, 教授, 博士生导师, 研究领域是电子设计自动化方法和通信专用集成电路设计.

2003-07-31 收到, 2003-11-20 定稿

©2004 中国电子学会