Accurate Resource Estimation Algorithms for Behavioral Synthesis

Srinivas Katkoori
CSE Department
University of South Florida
Tampa, FL 33620-5399
katkoori@csee.usf.edu

Ranga Vemuri
ECE&CS Department
University of Cincinnati
Cincinnati, OH 45221-0030
ranga@ececs.uc.edu

Abstract

Given a scheduled data flow graph the functional, storage, and interconnect (multiplexors) resources are analytically estimated taking into account the effects of post-scheduling tasks. Complexity of the controller implementation is also estimated. The novelty of this work lies in predicting the effects of the post-scheduling tasks on the final amount of resources, the effects of data path resource optimization on the controller complexity. Experimental results show high correlation between estimated and actual numbers.

1 Introduction

Techniques exist in literature [1, 2, 3] that attempt to estimate tight lower and upper bounds on the resources. We adopt a similar approach. The novelty of this work lies in predicting the effects of the subsequent tasks (register allocation, register optimization, and binding) on the final amount of resources. The amount of resource optimization in the data path affects the controller complexity. We also account for such a dependency. The algorithms are implemented in Profile-Driven Synthesis System [4].

2 Datapath Resource Estimation

Functional Resources: The CDFG is scheduled by employing a resource-constrained Force Directed List Scheduling (FDLS) Algorithm. Before performing scheduling, an ASAP (As Soon As Possible) schedule is generated and an analysis of the schedule for parallelism is carried out to generate a set of valid module bags. The estimated functional resources are same as that in the module bag.

Storage Resources: We adopt a bound-based approach to estimate the number of registers. From the scheduled CDFG, we compute the lower-bound and upper-bound on the storage resources appearing in the synthesized data path. The average of the two bounds gives us a simple estimate of the number of registers.

The lower bound on the storage resources, R_{lbound}, is equal to the minimum number of registers needed to implement a correct data path. Which in turn, is equal to the maximum number of edges (carriers) crossing the control step boundary. The upper bound on the storage resources, R_{ubound}, is equal to total number of declared and temporary carriers, as clearly we will never require registers more than the number of carriers in the CDFG.

Interconnect Resources: We assume point-to-point (multiplexor-based) interconnections. From the operation binding information we can estimate the number of interconnect units arising due to operator sharing. For each functional unit f_{op} in the module set, the operation binding information provides a list of operations L_{op} that are bound to the functional unit. If more than one n-ary operation is associated with a functional resource f_{op}, then we need n multiplexors at the inputs of f_{op}. The input size of each multiplexor so introduced is $|L_{op}|$ and the maximum operation bit-width in L_{op}.

The interconnect introduced due to register sharing is estimated based on the storage resource estimates. Each carrier appearing in the CDFG must be bound to some register. Thus, the R_{ubound} number of carriers are bound to R_{est} number of registers. Assuming that the register sharing is equally distributed amongst R_{est}, the number of inputs to the multiplexor introduced is equal to $\lceil R_{ubound}/R_{est} \rceil$. The bit-width of the multiplexor is equal to the bit-width of the register, at whose input the multiplexor is introduced.

3 Controller Complexity Estimation

We assume a PLA-based implementation. The *inputs* to the controller are classified into (1) a set of flags from the data path; and (2) feedback lines, $\mathcal{F}$. The flags are derived from the control branch points in the CDFG such as a conditional expression. Each flag is latched and fed to the controller. Depending on its value, the controller generates the control signals for the selected branch. The latches introduced for the flags are optimized. We estimate the number of latches after the optimization in the same way as we did for the storage resources in the data path as discussed in Section 2.

The *outputs* of the controller, $\mathcal{O}$, consists of (1) enable lines to the storage resources and select lines to the interconnect resources in the entire design; and (2) feedback lines, $\mathcal{F}$. $\mathcal{O}$ can be estimated accurately provided we can estimate the number of storage units and the number and size of interconnect units accurately.

0-7695-0104-4/99 $10.00 © 1999 IEEE

Design	FU Resources		Storage resources			
	Est	Act	L	U	Est	Act
	$\|F_{est}\|$		R_{lb}	R_{ub}	$\|R_{est}\|$	
Compress	2	2	1	2	2	2
Decompress	2	2	1	2	2	1
Elliptic	4	4	30	30	30	30
FIFO	10	10	4	23	14	11
Scalar	9	9	8	16	12	16
TLC	5	5	3	7	5	4

(I) Functional And Storage Resource Estimates

Estimation of storage units and interconnect units was discussed in the previous section.

The number of controller *states* is estimated as follows. Two operations are said to be *mutually exclusive* iff they belong to two distinct branches of a conditional or a case statement. Informally, the two operations are such that they can never be executed in the same clock cycle and therefore two distinct states in the controller have to be created. For each control step in the CDFG, we partition the scheduled operations into disjoint subsets of operations. Each subset, S, has the following properties: (1) Each operation in S and every operation belonging to a subset other than S, are mutually exclusive; and (2) No two operations in a subset are mutually exclusive with each other. The number of states introduced in any control step is equal to the number of disjoint subsets. Assuming binary state encoding, the number of feedback lines, $\mathcal{F}$, is given by $\lceil \log_2(\text{Total number of disjoint sets}) \rceil$.

Minterm estimation is difficult as it is dependent on lower level design factors such as amount of logic level optimization and state-encoding style. By synthesizing various benchmarks we derived an empirical formula for the number of min-terms as a function of state-size: $T = 1.45 * S$. That is, for every two states in the controller description, we have approximately three min-terms appearing in the optimized boolean equations. This scaling factor is dependent on the design domain, the logic optimization, and the synthesis tools being used.

4 Experimental Results

The functional and storage resource estimates for six benchmarks are given in Table (I). In Table (II), estimates of interconnect due to operator sharing and register sharing are shown. Since the operator binding information is known during the resource estimation phase, we expect the estimation error in case of operator sharing to be reasonably smaller. In case of register sharing, the interconnect estimation error is high due to: (1) the simplistic estimation of number of registers which tends to introduce large estimation error; and (2) the interconnect estimation error is directly proportional to the storage estimation error. Further, we assume that all the bounded edges are equally distributed amongst the estimated storage elements. In some cases, the edge distribution is non-uniform and is dictated by the clique sizes. Table (III) tabulates the latch estimation results. Table (IV) presents the controller complexity parameter estimates. As the output size of the controller is estimated from the storage and interconnect resources, its estimation error is directly affected by the resource estimation error.

Design	Op. Sharing		Reg. Sharing		Total	
	Est	Act	Est	Act	Est	Act
Compress	1	1	1	2	2	3
Decompress	1	1	1	2	2	3
Elliptic	3	5	29	29	32	34
FIFO	6	10	25	13	31	23
Scalar	0	0	12	6	12	6
TLC	3	3	15	7	18	10

(II) Interconnect (Mux) Estimates

Design	Lbound	Ubound	Est	Act
	$\|L_{lb}\|$	$\|L_{ub}\|$	$\|L_{est}\|$	
Compress	8	8	8	8
Decompress	8	8	8	8
Elliptic	1	1	1	1
FIFO	2	6	4	4
Scalar	1	1	1	1
TLC	5	8	6	3

(III) Estimation of Latches

Design	$\mathcal{I}$		$\mathcal{O}$		$\mathcal{S}$		$\mathcal{T}$	
	Est	Act	Est	Act	Est	Act	Est	Act
Compress	8	8	26	27	33	35	48	54
Decompress	8	8	26	25	32	34	47	50
Elliptic	1	1	132	143	35	44	51	54
FIFO	4	4	79	75	34	41	49	56
Scalar	1	1	60	39	8	13	12	17
TLC	6	3	53	41	33	39	48	60

(IV) Controller Complexity Estimates

References

[1] S. Chaudhuri and R. A. Walker. "Computing lower bounds on functional units before scheduling". *IEEE Transactions on Very Large Scale Integrated Systems*, 4(2), June 1996.

[2] J. M. Rabaey, C. Chu, P. Haong, and M. Potkonjak. "Fast prototyping of datapath-intensive architectures". *IEEE Design & Test of Computers*, pages 40–51, June 1991.

[3] A. Sharma and R. Jain. "Estimating architectural resources and performance for high-level synthesis applications". *IEEE Transactions on Very Large Scale Integrated Systems*, 1:175–190, June 1993.

[4] S. Katkoori, N. Kumar, and R. Vemuri. "High Level Profiling Based Low Power Synthesis Technique". In *Proceedings of International Conference on Computer Design*, pages 759–765, October 1995.