LOW POWER SCHEDULING WITH RESOURCES OPERATING AT MULTIPLE VOLTAGES

Wen-Tsong Shiue and Chaitali Chakrabarti

Center for Low Power Electronics
Department of Electrical Engineering
Arizona State University, Tempe, AZ 85287-5706, USA

ABSTRACT

This paper presents (i) a resource constrained scheduling scheme and (ii) a latency-constrained scheduling scheme that minimize power consumption for the case when the resources operate at multiple voltages. The resource-constrained scheduling reduces the power consumption by maximally utilizing resources operating at reduced voltages and at the same time reducing the latency. The latency-constrained scheduling scheme reduces the power consumption by assigning as many nodes (of the data flow graph) as possible to the resources operating at reduced voltages. Two cases have been studied: one in which the possible operating voltages are 5V and 3.3V, and the other in which the operating voltages are 5V, 3.3V and 2.4V. Experiments with some HLS benchmark examples show that the proposed schemes achieve significant power reduction.

1. INTRODUCTION

In the past, the primary considerations in VLSI design were performance, area, cost, and reliability. In recent years, the demand on portability has compelled low power to be one of the primary design considerations. It has been well established that to design a low power system, power has to be optimized at all levels of the design process - system, algorithm (behavior), architecture, logic, circuit and processing technology [1]. The work presented in this paper focuses on behavioral level power optimization. Specifically, we address the problem of scheduling a data-flow graph (i) under resource constraint and (ii) under latency constraint for the case when the resources operate at multiple voltages. The resource constraint thus consists of not only the type and number of resources but also the corresponding operating voltages. The proposed resource-constrained algorithm is list based, and tries to achieve a balance between reducing the number of control cycles and maximally utilizing the resources operating at reduced voltages. In comparison, the algorithm in [2] uses a combination of dynamic programming and geometric scheduling is computationally more complex. Recently, another dynamic programming technique has been proposed in [5] for the multiple voltage scheduling problem. This algorithm has pseudo-polynomial complexity and produces optimal results for trees (and not for general directed acyclic graphs).

The latency-constrained scheduling algorithm proposed in this paper also has polynomial time complexity and minimizes power consumption for the case when the resources operate at multiple voltages (5V, 3.3V and 2.4V). Since there is no restriction on the number of available resources, the bias is towards operating the resources at low voltages without violating the timing constraint. Experiments with some benchmark examples show that the average reductions for tight timing constraints are ≈ 39% for a 2 voltage system and ≈ 58% for a 3 voltage system. The problem of latency-constrained scheduling with multiple voltage units has also been addressed in [3]. The algorithm in [3] iteratively computes the voltage assignment of each node and is computationally more complex compared to the one-pass algorithm presented here. A comparison of the results on benchmark examples show that our method achieves greater power reduction.

2. PRELIMINARIES

The input to both our algorithms is a data flow graph and resources that operate at multiple voltages. We assume that (i) the functional unit can be operated at 3 voltages: 5V, 3.3V and 2.4V and (ii) if the delay of the functional unit at 5V is Δ, then the delay of the unit at 3.3V is 2Δ and that at 2.4V is 3Δ. We assume that multiple power lines are available. Thus there is no need for on-chip level shifters for the power lines. On-chip level shifters are however required in the data path, since data is transferred between resources operating at different voltages. The proposed algorithms make a conscious effort to reduce the number of instances in which data is passed through a level shifter. The first scheme assigns lower voltage to the nodes with smaller ASAP value. This is because for $x < y$, a $xV \rightarrow yV$ level shifter consumes less power compared to a $yV \rightarrow xV$ level shifter [5]. The second scheme assigns xV to nodes whose parent or children have been assigned to xV (if possible).

3. RESOURCE-CONSTRAINED CHEDULING ALGORITHM

In resource-constrained scheduling, the problem is given a DFG and resource constraints, assign nodes to resources such that the power consumption is minimum and that the latency is as small as possible. In order to reduce the power consumption, the nodes have to be assigned to low voltage resources. However, low voltage resources have larger delays. The proposed scheduling algorithm tries to achieve a balance between maximally utilizing resources operating at reduced voltages and reducing the latency.

The resource-constrained scheduling algorithms are list based. In each cycle, the algorithm first assigns nodes with high freedom to low voltage resources and nodes with low freedom to high voltage resources. If at the end of this step, there are still some unassigned nodes, then the algorithm assigns these nodes to whatever resources are available (high voltage or low voltage). Such a procedure results in a schedule with maximum utilization of low voltage resources with little increase in the computational latency. We have developed algorithms for the case when the resources operate at 5V and 3.3V and for the case when the resources operate at 5V, 3.3V and 2.4V. The resources (multipliers and adders) can have different delays and can be pipelined.

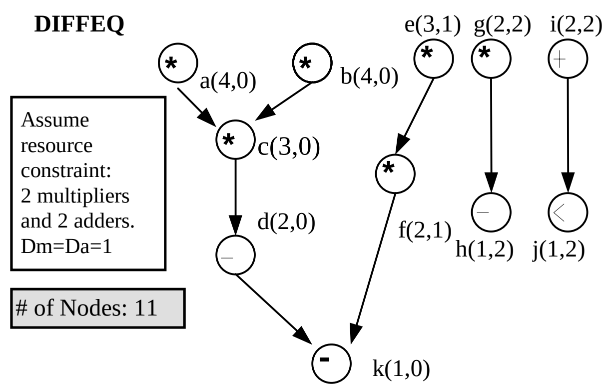
We describe briefly the algorithm for the case when the resources are non-pipelined and operate at 5V and 3.3V. The algorithm first assigns the nodes with high priority to the available 3.3Vmultipliers and adders. The priority of a node is a function of its depth (primary), freedom, and the possibility of requiring a level shifter. Next, the remaining nodes are assigned to the 5V operators. The nodes are sorted according to their priority and nodes with high priority are assigned to the 5V multipliers and adders. Here too the priority is a function of the depth (primary), freedom and level shifter requirements. However, now, if two nodes have the same depth, then the node with the lower freedom has higher priority. The algorithm is shown below.

---

1. *Determine the ASAP, ALAP, depth, and freedom value of each node.*
2. *Add nodes in the ready set of cycle 1 to Queue.*
3. *while Queue not empty*

- *for nodes with freedom >=1*
  - *Assign nodes of higher depth to the available 3.3V operators.*
  - *Mark the corresponding 3.3V operators as busy in the next cycle.*
  - *Assign nodes of higher depth to the available5V operators.*
  - *Assign the remaining nodes to 3.3V operators if possible.*
  - *Append the queue with the ready nodes of the next control cycle.*

The algorithm for the case when the resources operate at 3 voltages is an extension of this algorithm. Here, first, the nodes with freedom >= 2 are assigned to the available 2.4V resources, then the nodes with freedom >=1 are assigned to the available 3.3V resources, and finally, the remaining nodes are assigned to the available 5V resources. If, there are still same unassigned nodes in the ready set, then these are assigned to the available 2.4V or 3.3V resources.



(a) DFG of DIFFEQ example. Each node is labeled with the tuple (depth, freedom)

Cycle	Mult. 3.3V	Mult. 5V	Adder 3.3V	Adder 5V
1	e	a	i	-
2	e'	b	i'	-
3	g	c	j	-
4	g'	f	j'	d
5	-	-	h	k
6	-	-	h'	-

(b) the final scheduling results

Figure 1. Resource-constrained scheduling. Example DIFFEQ for the case when the number of voltage levels is 2.

Example1 : We explain this algorithm with the help of the DIFFEQ example (Figure1). The data flow graph of Example 1 consists of both additions and multiplication as shown in Figure 1(a). Our objective is to schedule the nodes among 2 multipliers and 2 adders. Here the delay of the multipliers and adders is the same (Dm=Da=1). The final scheduling results are given in Figure 1(b). From the results, we see that since 5 nodes have been assigned to 3.3V resources, power consumption has been significantly reduced. The number of level shifters used here is one (i.e. $LS_{3.3V \rightarrow 5V}$).

We present the results obtained by running our algorithm on some high-level synthesis benchmarks. The algorithms were implemented in MATLAB. Since [3] deals with the case when both the latency and the resource constraints are specified and our algorithm deals with only the resource constraints, we designed a small module to take latency constraints into consideration. The comparisons with AR lattice and EW filter are listed in Figure 2. In these two examples, it was found that our procedure yielded a higher reduction in power. For instance, for the EW filter with M_5V(2EA), A_3.3V(2EA) and A_5V(2EA) and a timing constraint of 22, we achieved a 26.56% reduction compared to the 23.24% reduction in [3]. Figure 3 describes the power savings results when exact values of power consumption of the resources (multipliers, adders and level shifters) are used. The power consumption values have been obtained from [5]. Note that, while for the DIFFEQ and AR filter example the % reductions with the level shifter are comparable to that without the level shifter, for the EW filter, the difference in the % reduction is significant.

Bench	Resource Constraint				Timing Constraint		% Red. 5V, 3.3V	
mark	M 3	M 5	A 3	A 5	Ours	Raje's	Ours	Raje's
EW Filter	0	2	2	2	22	22	**26.56**	23.24
					25	25	31.54	31.54
	0	2	1	2	21	21	13.28	13.28
					24	25	18.26	18.26
AR Filter	0	2	2	2	**16**	17	20.16	20.16
	0	2	1	2	24	24	**25.14**	16.13

Figure 2 : Resource-Constrained Scheduling Results for EW and AR filter when the resources operate at 2 voltage levels (5V, 3.3V). Where M3 / (A3) is 3.3V multiplier / (adder) and M5 / (A5) is 5V multiplier / (adder).

Bench	Resource Constraint				Timing	% Red. (5V, 3.3V)	
mark	M 3	M 5	A 3	A 5	k	without LS	with LS
DIFF	1	1	1	1	6	19.38	18.72
EQ	2	2	2	2	5	28.43	27.77
AR	1	1	1	1	16	10.54	9.7
Filter	2	2	2	2	11	13.95	13.45
EW	1	1	1	1	20	14.54	8.26
Filter	2	2	2	2	17	14.54	8.26

Figure 3 : Resource-Constrained Scheduling Results for DIFFEQ, AR and EW filter when exact values of power consumption are used.

4. LATENCY-CONSTRAINED SCHEDULING ALGORITHM

In latency-constrained scheduling, the problem is given a DFG and a timing constraint, assign voltages to the nodes such that the power consumption is minimum. The proposed algorithms try to assign low voltages to as many nodes of the DFG as possible so that $\sum C_i V_i^2$ is minimum. The algorithms also try to reduce the number of level shifters using the two heuristics described in Section 2. A summary of the algorithm is given below.

---

1. *Compute the ASAP and ALAP values of each node. Freedom = ALAP - ASAP.*
2. *Form groups with nodes having the same ASAP and ALAP values.*
3. *Form sets with groups having the same freedom. Assign priorities to the members of each set, where the priority =* $\sum_i n_i C_i$.
4. *For each freedom, do step 5 and step 6.*
5. *List the possible assignments in each set. Member priority is used to prune this list.*
6. *Compute the power for each assignment and choose the one with minimum power.*

Since the groups are ordered according to their priority, it can be shown that for each freedom *j*, there are only *O(j)* possible assignments that need to be checked. Thus the complexity of this algorithm is polynomial.

#	ASAP	ALAP	F	set
a	1	1	0	G_1
b	1	1	0	
c	2	2	0	G_2
d	3	3	0	G_3
k	4	4	0	G_4
e	1	2	1	G_5
f	2	3	1	G_6
g	1	3	2	G_7
i	1	3	2	
h	2	4	2	G_8
j	2	4	2	

(a) ASAP, ALAP and freedom value of each node.
Node {e, g, h, i, j} are assigned to 3.3V.
Node {a, b, c, d, f, k} are assigned to 5V.
(b) Final assignment.

Figure 4: Latency -constrained scheduling. Example DIFFEQ for the case when the number of voltage levels is 2.

Example2 : We explain this algorithm with the help of the DIFFEQ example of Figure 1. Assume that the computation time of each node is the same and that the latency is constrained to 4 (length of the critical path). In Step 1, the ASAP time, the ALAP time and the degree of freedom is computed for each node (Figure 4a). In step 2, groups G_1 through G_8 are formed. Note that groups G_1, G_7 and G_8 have a cardinality of 2 while the other groups have a cardinality of 1.

In step 3, sets are formed and the elements in a set are ordered according to their priority. The nodes of the groups corresponding to freedom = 0 are assigned to 5V. In the set corresponding to freedom = 1, there are two groups *G*5 and *G*6. We assign 3.3V to G5 and 5V to G6 since it results in the data passing through only are level shifters (3.3V→5V). Finally, the nodes in the set corresponding to freedom = 2 are assigned to 3.3V. If the capacitance of a multiplier is 4 times that of the an adder, the power is proportional to $\sum 4*(4*5^2+2*3.3^2)+\sum(2*5^2+3*3.3^2)=569.8$ compared to 725 for the assignment with 5V operating voltage.

We present the results obtained by running our algorithm on some high-level synthesis benchmarks. In order to compare our results with [3], (i) the power quoted in the table is $\sum_{i \in nodes} V_i^2$, where *Vi* is the voltage assignment of node *i*, (ii) the possible operating voltages are 5V and 3V (instead of 3.3V), (iii) it is assumed that the capacitance of a multiplier and an adder are the same and (iv) the power consumed by the level shifters is not taken into account. Figure 5 lists the results for DIFFEQ, AR Lattice filter and EW filter. In all three examples, our procedure yielded a higher reduction in power. For instance, for the DIFFEQ three voltage assignment example, we achieved a 54.5 % reduction compared to 49.6% reduction in [3]. It is to be noted that for large timing constraints, our results are the same as that in [3]. This is to be expected, since for large constraints, all the nodes can be operated on at a reduced voltage. Figure 6 describes the power consumption results when the actual power consumed by the multipliers, adders, and level shifters are taken into account. We used the power consumption values listed in [5]. We assumed that the delay of a multiplier is 5 times that of an adder and that the control cycle is 20 ns long as in [5]. Note that the difference in the % power reduction with and without the level shifter power being considered is 4-7%.

5. CONCLUSION

In this paper we present (i) a resource-constrained scheduling scheme and (ii) a latency constrained scheduling scheme for the case when the resources operate at multiple voltages. Both schemes try to reduce the overall power consumption (including power consumed by the level shifters). We study two cases: one in which the possible operating voltages are 5V and 3.3V and the other in which the operating voltages are 5V, 3.3Vand 2.4V. Experimental results show that the average reductions for both the systems are quite high. However, the 3 voltage system has a larger power reduction and a larger design overhead compared to the 2 voltage system.

Bench mark	Timing	Avg. % red. (5V, 3V)		Timing	Avg. % red. (5V, 3V, 2.4V)	
	k	Ours	Raje's	k	ours	Raje's
DIFF	4-7	46.6	40.7	4-8	54.5	49.6
EQ	>=8	64	64	>=12	76.9	76.9
AR	8,9,10, 12	28.6	27.4	8,9,10, 12	31.4	30.2
Filter	>=16	64	64	16,24	71.9	71.9
EW	14-27	41.3	-	14-41	58.3	-
Filter	>=28	64	-	>=42	76.9	-

Figure 5 : Power consumption results for 2 and 3 voltage levels for latency constrained scheduling. Here the capacitance and delay of a multiplier and adder are the same (i.e. Cm=Ca & Dm=Da).

Bench mark	Timing	Avg. % red. (5V, 3.3V)		Timing	Avg. % red. (5V, 3.3V, 2.4V)	
	ns	wt/LS	w/LS	ns	wt/LS	w/LS
DIFF	240-300	44.38	43.05	260-380	66.7	65.5
EQ	>=320	56.44	56.44	>=400	76.96	76.96
AR	420-540	49.68	41.71	440-620	66.05	59.66
Filter	>=560	56.44	56.44	>=720	76.96	76.96
EW	580-760	39.13	33.65	620-920	55.95	50.43
Filter	>=800	56.44	56.44	>=1080	76.96	76.96

Figure 6 : Latency constrained scheduling results when exact values of power consumption are used and assume control cycle is 20ns and Cm=21.2Ca.

6. REFERENCE

[1] A. Chandrakasan, S. Sheng and R. Brodersen, "Low power CMOS digital design", IEEE journal of Solid State Circuits, pp. 473-484, 1992.

[2] S. Raje and M. Sarrafzadeh, "Scheduling with two voltages under resource constraints", Technical Report. Department of Electrical Engineering and Computer Science, Northwestern University

[3] S. Raje and M. Sarrafzadeh, "Variable Voltage Scheduling", Proceedings of the International Symposium on Low Power Design, pp. 9-14, 1995.

[4] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low power", Proceedings of the International Symposium on Low Power Design, pp. 3-8, 1995.

[5] Jui-Ming Chang and Massoud Pedram, "Energy minimization using multiple supply voltages", Proceedings of the International Symposium on Low Power Design, Monterey CA, 1996.