

A Synthesis Scheme for Low Power Designs with Multiple Voltages Under Timing Constraints

Ling Wang, Yingtao Jiang, Henry Selvaraj and Venkatesan Muthukumar
Department of Electrical & Computer Engineering
University of Nevada, Las Vegas
Las Vegas, NV 89154, USA
{lwang, yingtao, selvaraj, venkim}@egr.unlv.edu

Abstract

This paper presents a time-constrained synthesis scheme to minimize power consumption with resources operating at multiple voltages. The input to our scheme is an unscheduled data flow graph, and the timing constraints. The proposed scheme, with polynomial time complexity, has following four steps. 1) The initial resource running at the lowest possible voltage is selected for each operation from a design library. 2) Operations are then scheduled with the objectives to minimize the power consumption due to the resources. 3) Operations are clustered to form voltage islands to minimize the power consumption due to the interconnections. Note that in both Steps 2 and 3, some operations may have to be reassigned to the resources to satisfy the timing constraints and reduce the interconnections, respectively. 4) Operations are finally bound to the resources. Experiments with a number of DSP benchmarks show that the proposed algorithms achieve the power reduction by an average of 46.5%.

Keywords

Low power, multiple supply voltages, partitioning, scheduling

1. Introduction

Power considerations have become an increasingly dominant factor in the design of both portable and desktop systems that demand high-speed computations and complex functionalities with low power consumption. An effective way to reduce the power consumption is to lower the supply voltage level of a circuit due to the quadratic proportional relationship between power consumption and supply voltage. For this reason, more recently, the use of multiple supply voltages on the chip has attracted attention [2]-[4][6]-[9]. This scheme has the

advantage of allowing modules on the critical paths to use the highest voltage level (thus meeting the required timing constraints) while allowing modules on non-critical paths to use lower voltages (thus reducing the energy consumption)[2].

Across the literature, various multiple voltage synthesis schemes under either timing constraints [2][3][7][9], or resource constraints[3][9], or both timing and resource constraints [3][4][6][8], have been proposed. In [7], a scheduling technique is given by utilizing multiple supply voltages to reduce the power under timing constraints. In [2] a dynamic programming scheme is presented for assigning voltage levels to the modules in nonpipelining and functionally pipelined datapaths. In [3], an integer linear programming (ILP) model is developed, and then a heuristic scheduling algorithm that has exponential time complexity is presented. However, these time-constrained multiple voltage synthesis algorithms [2][3][7] are mainly focused on minimizing the power consumption due to the resources. As a matter of fact, there is a large opportunity available for further decreasing power consumption through interconnection power reduction [5]. In [5], such interconnection power reduction is achieved by partitioning to preserve the locality in the assignment of operations to hardware units. However, the target chip in [5] operates at a single supply voltage. Although the list-based time-constrained multiple voltage algorithm [9] reduces the power due to the interconnections by applying pattern matching to explore the structure regularity, the algorithm is not very effective for irregular designs.

To address above problems, in this paper, we present a time-constrained multiple voltage synthesis scheme to minimize the power consumed by both the resources and the interconnections. The power reduction in our scheme has been irrespective of the hardware structures.

2. Background

2.1 Notations

A DFG (Data Flow Graph) is a directed acyclic graph $G = (V, E)$, where V is a set of nodes, and E is a set of edges between nodes. Here each node represents an operation, and a directed edge from node v_i to node v_j means execution of v_i must precede that of v_j .

Timing constraint is the time available to execute the nodes in a DFG.

The mobility of a node is the difference between its ALAP (As Late As Possible) schedule time and its ASAP (As Soon As Possible) schedule time. For time-constrained scheduling, the ASAP and ALAP times are computed using the specified latency.

A voltage cluster $C(V_i)$ is a group of nodes that operate at the same supply voltage V_i .

Two nodes are tightly connected if the shortest distance between them, in terms of the number of edges traversed, is small.

The gain of a node v with respect to a cluster $C(V_i)$, denoted as $g(v, C(V_i))$, is the weighed sum of the edges between v and nodes in $C(V_i)$.

2.2 The Timing Model

Let c denote a control step (clock cycle), the basic unit of time used in the DFG in behavioral level. For a given length of c -step, an operation may thus become a multi-cycle operation.

Let t_i be the starting time of node v_i (the schedule time of v_i), $v_{i1}, v_{i2}, \dots, v_{ij}$ the parent nodes of node v_i , t_{ij} the starting time of node v_{ij} , d_{ij} the execution time (delay) of node v_{ij} , then we have

$$t_i = \max\{(t_{i1} + d_{i1}), (t_{i2} + d_{i2}), \dots, (t_{ij} + d_{ij})\} \quad (1)$$

$$t_i^e = t_i + d_i \quad (2)$$

$$T = \max\{t_i^e\}, i \in \{1, 2, \dots, n\} \quad (3)$$

where d_i is the delay of node v_i , t_i^e is the ending time of node v_i , T is the total schedule time, n is the number of nodes in the DFG.

Let P be a critical path, and T can be calculated by traversing the critical path P which is

$$T = \sum_{i=1}^n x_i d_i \quad (4)$$

$$x_i = 1 \text{ if node } v_i \text{ is on the critical path } P; \text{ otherwise, } x_i = 0 \quad (5)$$

Lemma 1 The delay of an operation is inversely proportional to its supply voltage.

3. Problem Formulation

The time-constrained multiple voltage schedule problem is defined as follows:

Given a DFG and the timing constraints, find a schedule that satisfies the given timing constraints with minimum power consumption.

We consider total power consumption consisting of two components, P_{res} and P_{int} , the power consumed by the resources and the interconnections, respectively. We concentrate our attention on the inter-cluster interconnections (between macroblocks, e.g., different datapaths). Let N be the total number of interconnections among clusters C_1, C_2, \dots, C_r , r the total number of clusters, and n the total number of nodes in the DFG, then our objective function is given as:

$$\text{Min}\left\{\sum_{j=1}^N P_{int}^j + \sum_{i=1}^n P_{res}^i\right\} \quad (6)$$

Subject to

$$T \leq T_C \quad (7)$$

$$S_k \leq S_C, k = 1, \dots, r \quad (8)$$

where T is the total schedule time, T_C is the timing constraint, S_k is the number of nodes in the cluster C_k , S_C is the allowed maximum number of nodes in each cluster.

4. ALGORITHM DESCRIPTION

In this section, we present an algorithm for the above problem. The inputs to the proposed algorithm consist of the DFG representation of a circuit, the timing constraints, and a design library with fully characterized resources, and an allowed maximum cluster size. The proposed scheme has the following four steps.

4.1 Step 1 Initial Resource Assignment

For a given DFG, ASAP and ALAP are performed with the resources running at the highest voltage. Then, the mobility of each node is calculated. The initial resource is searched for each node v in the design library such that it has the lowest possible voltage and its delay is less than or equal to m plus d^M , where m is the mobility of node v and d^M is the delay of node v when node v is assigned to the highest voltage resource (i.e. the timing constraint is satisfied at the node level).

The complexity of ALAP and ASAP is $O(n)$, and the complexity of searching for the initial resources is $O(l \cdot n)$, where l is the number of types of resources in the design library. In a meaningful design, we have $l \ll n$. Therefore, the complexity of this step is $O(n)$, where n is the number of nodes.

4.2 Step2 Scheduling

Lemma 2: If the schedule time of any node v_i surpasses its ALAP schedule time, that is,

$$t_i > l_i$$

where t_i is the schedule time of node v_i , l_i is the ALAP schedule time of node v_i , then the timing constraints are violated.

(Proof: see Appendix)

After the initial resource is assigned to each node, the timing constraint is satisfied at the node level. However, the timing constraints may not be satisfied at the DFG level (i.e., the resource assignment may not be feasible).

The feasibility of the resource assignment is evaluated by the schedule time of each scheduled node while the scheduling is performed. If the schedule time of any node surpasses its ALAP schedule time, the timing constraints are not satisfied (see Lemma 2). In above case:

- The node assigned to the resource consuming less power has higher priority. The nodes with the highest priority are selected among the scheduled nodes.
- Selected nodes are reassigned to the higher voltage resources.

The scheduling is iteratively performed until the resource assignment is feasible. The scheduling algorithm is given as follows.

```

Schedule_Nodes_To_Minimize_Func_Power (G (V, E),
T){
while (all nodes are not scheduled) {
  put all the nodes that do not have parent nodes into
  the Ready List;
  while (!Ready List) {
    find node  $v_i$  in the ready list with the least
    mobility;
    // Schedule the node  $v_i$  ;
    if (Schedule_Node( $v_i$ )=False)
      //timing constraints are violated.
      break;
    add all the child nodes of  $v_i$  into the ready list;
    delete  $v_i$  from the ready list;
  }
}
Schedule_Node( $v_i$ ){
   $t = t_i$  ; //  $t_i$  is denoted in Eq.(3) in section 2.2
  if ( $t > l_i$ ) { //  $l_i$  is the schedule time of  $v_i$  in ALAP
  scheduling
   $D = t - l_i$ ; //  $D$  is denoted as lag time.
  while ( $D \neq 0$ ) {
    select the highest priority node  $v_j$  among  $v_i$  and
     $v_i$ 's parent nodes;
    increase  $v_j$ 's voltage by one level;
     $v_i = v_j$ ;
    update  $D$ ;
  }
  return False;
}
else {
  schedule node  $v_i$  at the schedule time  $t$ ;
  return True;
}
}

```

In the scheduling process, the worst case is that the schedule time of each node exceeds its ALAP schedule time so that the scheduling needs to be performed n times. Hence, the time complexity of Step 2 is $O(n^2)$, where n is the number of nodes.

4.3 Step 3 Partitioning

Initial partitioning is performed to assign the nodes with the same voltage into the same cluster. If the number of nodes in one cluster exceeds the user-defined threshold size, extra clusters will be created using KL partitioning [10] to obtain the optimal partition.

5. An illustrative Example

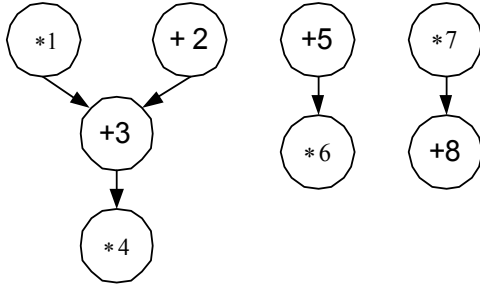


Figure. 1 Data flow graph of example

Table. 1 Power consumption (pJ), normalized delay ($c=20\text{ns}$) of functional units in [2]

Module	5V		3.3V		2.4V	
	Delay	Power	Delay	Power	Delay	Power
Adder	1	118.0	2	51.4	3	27.2
Substracter	1	118.0	2	51.4	3	27.2
Multiplier	5	2504	9	1090	15	576.9

In this section, an example (Fig.1) is provided to illustrate the proposed algorithm. We assume that the timing constraint is $11c$, where c is a 20 ns control step. The delay and power of each resource is given in Table 1 adopted from [2].

In Step 1, the minimum voltage resource is assigned to each node according to the mobility. Initial voltage of each node is reported in the Column V1 in Table2.

In Step 2, at the beginning, the *Ready List* is $\{1, 2, 7, 5\}$. Nodes are sorted according to their motilities in the *Ready List*. As initial voltage of node 1 is 5V, node 1 is firstly assigned to the 5V multiplier at the control step from 1 to 5. Next, node 2 is assigned to the 2.8V multiplier at the control step from 1 to 3. After nodes 1 and 2 are scheduled, the *Ready List* is updated to $\{3, 7, 5\}$ since node 3 is the child node of node 1 and node 2.

After node 3 is assigned to the 5V adder at the control step 7, the ready list is $\{4, 7, 5\}$. Node 4 is assigned to the 5V multiplier at the control step from 8 to 11. Nodes 1, 2, 3 and 4 are scheduled without changing the initial resource since their control steps meet the timing constraints.

Now the *Ready List* is $\{7, 5\}$. Node 7 is assigned to the 3.3V multiplier and its delay is $9c$.

After node 8 is added to the *Ready List*, the *Ready List* becomes $\{8, 5\}$. Node 8 is assigned to the 2.4V adder at the control step 10, 11, 12. Timing constraint is

violated since the time constraint is $11c$. Scheduling is interrupted. Lag time D is equal to 1.

As node 8 is not on the critical path and its operation is *add*, it is reassigned to the 3.3V adder without searching for its parent nodes. Lag time D reduces to 0, and then scheduling is restarted.

Nodes 7 and 8 are rescheduled at control step 1 and 3 without breaking timing constraint.

Then, the *Ready List* is $\{5\}$. Node 5 is assigned to the 2.8V adder and scheduled at the control step 1.

The *Ready List* is $\{6\}$. Node 6 is assigned to 3.3V multiplier and scheduled at the control step from 4 to 12. Node 6 violates timing constraints, therefore scheduling is stopped. Lag time D is $1c$.

As the operation of node 6 is *multiply*, searching is initiated. We search for parents nodes of node 6 and find out node 5. Then node 5 is reassigned to the 3.3V adder. Lag time D is updated to 0 and searching ends. Scheduling is restarted.

After node 5 is rescheduled at control step 1, node 6 is rescheduled at control step 3 and assigned to the 3.3V multiplier. Timing constraints is satisfied. Scheduling ends at this point. The resource required by each node obtained from Step 2 is shown in the Column V2 in the table 2.

During the process of Step 3, three clusters are formed. Cluster 1 consists of nodes 1, 3, 4. Cluster 2 consists of nodes 5, 6, 7, 8. Cluster 3 has node 2. Node 2 has maximum gain to 5V Cluster 1. Thus the node 2 is reassigned to the 5V adder and moved into Cluster 1. Then, nodes 1, 2, 3, and 4 are in the 5V Cluster 1. Nodes 5, 6, 7 and 8 are in the 3.3V Cluster 2. The voltage of resource assigned to each node obtained from Step 3 is shown in the Column V3 in the table 2.

Table. 2 ASAP, ALAP, mobility, voltage of each node of (Figure.1)

Node #.	ASAP	ALAP	Mobility	V1	V2	V3
1	1	1	0	5	5	5
2	1	5	4	2.8	2.8	5
3	6	6	0	5	5	5
4	7	7	0	5	5	5
5	1	6	5	2.8	3.3	3.3
6	2	7	5	3.3	3.3	3.3
7	1	6	4	3.3	3.3	3.3
8	6	11	1	2.8	3.3	3.3

In Step 4, node 2 is rescheduled at the control step 1. 5V Multiplier 1 is assigned to node 1 and 4. After 5 V Adder 1 is assigned to node 2 and 3, 3.3V Multiplier 2 is assigned to node 6 and 7. In the end, 3.3V Adder 2 is assigned to node 5 and 8. The final scheduling result is shown in the Fig. 2. The binding result is shown in the Fig. 3.

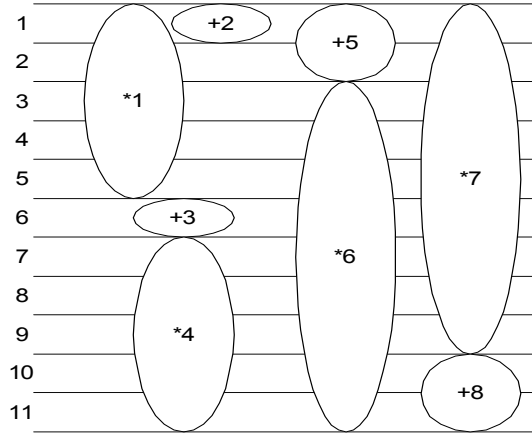


Fig.2 Scheduling result of example Figure.1

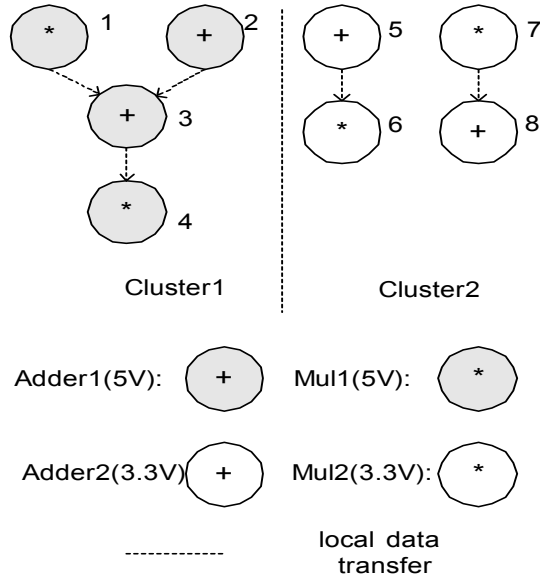


Fig 3 Partitioning and binding results of example Figure.1

6. Experimental Results

A library with fully characterized resources is adopted from [2] to evaluate our algorithm. The switching activity of nodes is assumed to be 0.5. The number of interconnections among clusters is directly proportional to

power consumption, and thus it serves as an indicator of the power reduction.

In Table 3, T_C is the timing constraint, and control step c equals to 20ns. P_{fun}^5 is the resource power consumption correspond to the supply voltage of 5V. P_{fun} is the resource power consumption obtained by our algorithm. Rn_{fun} is the reduction from P_{fun}^5 to P_{fun} , and Rn_{int} is the reduction of the number of interconnections. The average resource power reduction is 38.75% and 60% and the average interconnection reduction is 7.75% and 12.25%, when timing constraint is 1.5 times and 2 times of ASAP total schedule time, respectively.

Table 3. Results

Benchmark	T_C (c)	P_{fun}^5 (pJ)	P_{fun} (pJ)	Rn_{fun} %	Rn_{int} %
DCT	25	43132	36112.2	16.2	25
	34	43132	16712.6	61.3	25
Lattice	36	23598	17182.8	27.2	6
	48	23598	16635.8	29.5	21
lir7	30	39212	16953.5	56.8	0
	40	39212	9909.1	74.7	3
Wavelet	31	73180	33095.2	54.8	0
	42	73180	18676.4	74.5	0

7. Conclusions

In this paper, we have presented a time-constrained multiple voltage synthesis scheme to minimize the power consumption due to both the resources and the interconnections. The algorithm has time complexity of $O(n^2 \log n)$, where n is the number of nodes in the DFG. Experiments with a number of DSP benchmarks show that the proposed algorithms achieve significant power reduction by average of over 46.5%.

8. References

- [1] P. Chandrakasan, S. Sheng, R. W. Brodersen, "Low Power CMOS Digital Design," *IEEE J. Solid-State Circuits.* vol. 27, pp.473-483, Apr. 1992.
- [2] J.-M. Chang, and M. Pedram, "Energy Minimization Using Multiple Supply Voltages," *IEEE Trans. VLSI*, vol.5, pp.157-162, Dec. 1997.

- [3] Y.-R. Lin and C.-T. Hwang, "Scheduling Techniques for Variable Voltage Low Power Designs," *ACM Trans. Design Automation Electronic Systems*, vol.2, pp. 81-97, Apr. 1997.
- [4] A. Manzak and C. Chakrabarti, "A Lower Power Scheduling Scheme with Resources Operating at Multiple Voltages," *IEEE Tran. VLSI Systems*, vol.10, pp. 6-14, Feb. 2002.
- [5] R. Mehra, L. M. Guerra, and J. M. Rabaey, "Low Power Architectural Synthesis and the Impact of Exploiting Locality," *J. VLSI. Signal Processing*, vol.13, pp.239-258, 1996.
- [6] M.C. Johnson and K. Roy, "Datapath Scheduling with Multiple Supply Voltages and Level Converters," *ACM Trans. Design Automation Electronic Systems*, vol.2, pp. 227-248, July 1997.
- [7] S. Rajee and M. Sarrafzadeh, "Variable Voltage Scheduling," *Proc. Int. Symp. Low Power Designing*, pp.9-14, Apr 1995.
- [8] S. Rajee and M. Sarrafzadeh, "Scheduling with multiple voltages," *Integr. VLSI. J.*, pp.37-60, Oct.1997.
- [9] W.T. Shiue and C. Chaitali, "Low Power Scheduling with Resources Operating at Multiple Voltages," *IEEE Trans. CASII*, vol.47, pp.536-543, June 2000.
- [10] B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," *Bell System Technical Journal*, Vol. 49, pp.291-307, 1970.

APPENDIX

Proof:

- (i) It is easy to show when v_i is on the critical path.
- (ii) We assume that v_i is not on the critical path. Let P be a critical path, then we define

$$z_{i,j}^1 = \begin{cases} 1 & \text{if node } v_j \text{ is on the critical path } P \text{ and is executed before node } v_i \\ 0 & \text{otherwise} \end{cases}$$

$$z_{i,j}^2 = \begin{cases} 1 & \text{if node } v_j \text{ is on the critical path } P \text{ and is executed after node } v_i \\ 0 & \text{otherwise} \end{cases}$$

From Eq.(4), we have

$$T = \sum_{j=1}^n x_j d_j = \sum_{j=1}^n z_{i,j}^1 d_j + \sum_{j=1}^n z_{i,j}^2 d_j$$

where T is total schedule time, d_j is the delay of node v_j .

From $t_i > l_i$ and $t_i^e = t_i + d_i$, we have

$$t_i^e > l_i + d_i^M$$

where t_i^e is the ending time of node v_i , and d_i^M is the delay of node v_i when v_i is assigned to the highest voltage resource.

Since $t_i^e < \sum_{j=1}^n z_{i,j}^1 d_j$ and $d_j \geq d_j^M, j = 1, \dots, n$, we have

$$T \geq t_i^e + \sum_{j=1}^n z_{i,j}^2 d_j = t_i + d_i + \sum_{j=1}^n z_{i,j}^2 d_j \quad (*)$$

$$> l_i + d_i^M + \sum_{j=1}^n z_{i,j}^2 d_j \geq l_i + d_i^M + \sum_{j=1}^n z_{i,j}^2 d_j^M$$

We define

$$T_C = L = l_i + d_i^M + \sum_{j=1}^n z_{i,j}^2 d_j^M \quad (**)$$

where L is the ALAP total schedule time, T_C is the timing constraint. From (*) and (**), we have $T > T_C$.

Thus, the timing constraints are violated.