

9-1-1996

Datapath Scheduling with Multiple Supply Voltages and Level Converters

Mark C. Johnson

Purdue University School of Electrical and Computer Engineering

Kaushik Roy

Purdue University School of Electrical and Computer Engineering

Follow this and additional works at: <http://docs.lib.purdue.edu/ecetr>

Johnson, Mark C. and Roy, Kaushik, "Datapath Scheduling with Multiple Supply Voltages and Level Converters" (1996). *ECE Technical Reports*. Paper 92.

<http://docs.lib.purdue.edu/ecetr/92>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

DATAPATH SCHEDULING WITH
MULTIPLE SUPPLY VOLTAGES AND
LEVEL CONVERTERS

MARK C. JOHNSON
KAUSHIK ROY

TR-ECE 96-16
SEPTEMBER 1996



SCHOOL OF ELECTRICAL
AND COMPUTER ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

Datapath Scheduling with Multiple Supply Voltages and Level Converters

Mark C. Johnson and Kaushik Roy
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, 47907-1285, USA

`mcjohnso@ecn.purdue.edu, kaushik@ecn.purdue.edu`

This research was supported in part by ARPA (F33615-95-C-1625), NSF CAREER award (9501869-MIP), and ASSERT program (DAAH04-96-1-0222)

ABSTRACT

We present an algorithm called MOVER (Multiple Operating Voltage Energy Reduction) to minimize datapath energy dissipation through use of multiple supply voltages. In a single voltage design, the critical path length, clock period, and number of control steps limit minimization of voltage and power. Multiple supply voltages permit localized voltage reductions to take up remaining schedule slack. MOVER initially finds one minimum voltage for an entire datapath. It then determines a second voltage for operations where there is still schedule slack. New voltages can be introduced and minimized until no schedule slack remains. MOVER was exercised for a variety of DSP datapath examples. Energy savings ranged from 0% to 50% when comparing dual to single voltage results. The benefit of going from two to three voltages never exceeded 15%. Power supply costs are not reflected in these savings, but a simple analysis shows that energy savings can be achieved even with relatively inefficient DC-DC converters. Datapath resource requirements were found to vary greatly with respect to number of supplies. Area penalties ranged from 0% to more than 150%. Implications of multiple voltage design for IC layout and power supply requirements are discussed.

1. INTRODUCTION

A great deal of current research is motivated by the need for decreased power dissipation while satisfying requirements for increased computing capacity. In portable systems, battery life is a primary constraint on power. However, even in non-portable systems such as scientific workstations, power is still a serious constraint due to limits on heat dissipation.

One design technique that promises substantial power reduction is voltage scaling. The term "voltage scaling" refers to the trade-off of supply voltage against circuit area and other CMOS device parameters to achieve reduced power dissipation while maintaining circuit performance. The dominant source of power dissipation in a

conventional CMOS circuit is due to the charging and discharging of circuit capacitances during switching. For static CMOS, the switching power is proportional to V_{dd}^2 [Rabaey 1996]. This relationship provides a strong incentive to lower supply voltage, especially since changes to any other design parameter can only achieve linear savings with respect to the parameter change. The penalty of voltage reduction is a loss of circuit performance. The propagation delay of CMOS is approximately proportional to $\frac{V_{dd}}{(V_{dd}-V_T)^2}$ [Rabaey 1996], where V_T is the transistor threshold voltage.

A variety of techniques are applied to compensate for the loss of performance with respect to V_{dd} including reduction of threshold voltages, increasing transistor widths, optimizing the device technology for a lower supply voltage, and shortening critical paths in the data path by means of parallel architectures and pipelining.

Data path designs can benefit from voltage scaling even without changes in device technologies. Algorithm transformations and scheduling techniques can be used to increase the latency available for some or all data path operations. The increased latency allows an operation to execute at a lower supply voltage without violating schedule constraints. "Architecture-Driven Voltage Scaling" is a name applied to this approach.

A number of researchers have developed systems or proposed methods that incorporate architecture driven voltage scaling [Chandrakasan et al. 1995; Raghunathan and Jha 1994; Raghunathan and Jha 1995; Goodby et al. 1994; Kumar et al. 1995; SanMartin and Knight 1995; Rajee and Sarrafzadeh 1995; Gebotys 1995a]. HYPER-LP [Chandrakasan et al. 1995] is a system that applies transformations to the data flow graph of an algorithm to optimize it for low power. Other systems accept the algorithm as given and apply a variety of techniques during scheduling, module selection, resource binding, etc. to minimize power dissipation. All of the systems mentioned above try to exploit parallelism in the algorithm to shorten critical paths so that reduced supply voltages can be used. Most systems [Chandrakasan et al. 1995; Raghunathan and Jha 1994; Raghunathan and Jha 1995; Goodby et al. 1994; Kumar et al. 1995; Gebotys 1995a] also minimize switched capacitance in the data path.

Most voltage scaling approaches require that the IC operate at a single supply voltage. Although substantial energy savings can be realized with a single minimum supply voltage, one cannot always take full advantage of available schedule slack to reduce the voltage. Non-uniform path lengths, a fixed clock period, and a fixed number of control steps can all result in schedule slack that is not fully exploited. Figure 1 provides examples of each type of bottleneck. When there are non-uniform path lengths, the critical (longest) path determines the minimum supply voltage even though the shorter path could execute at a still lower voltage and meet timing constraints. When the clock period is a bottleneck, some operations only use part of a clock period. The slack within these clock periods goes to waste. Additional voltages would permit such operations to use the entire clock period. Finally, a fixed number of control steps (resulting from a fixed clock period and latency constraint) may lead to unused clock cycles if the sequence of operations does not match the number of available clock cycles. This could even occur in the critical path. Consider the control step bottleneck illustrated in figure 1. Decreasing the supply voltage would cause the datapath latency to increase from three to six clock

cycles. Unless the clock period can be changed, the datapath cannot be scaled to four clock cycles. Additional voltages would allow specific operations to be slowed down to take up unused cycles. It should be noted that in some cases these bottlenecks can be alleviated by restructuring the datapath specification or choosing alternate circuit implementations for some operations.

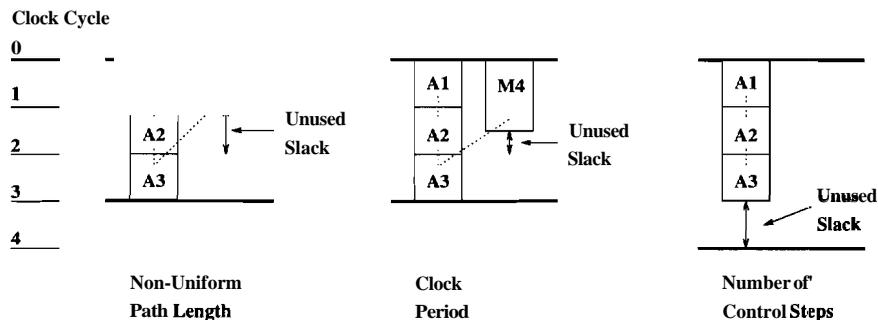


Fig. 1. Examples of scheduling bottlenecks

Literature on multiple voltage synthesis is limited, but this is changing. Publications that address the topic include [Raje and Sarrafzadeh 1995], [Gebotys 1995a], and [Johnson and Roy 1996]. Raje and Sarrafzadeh [Raje and Sarrafzadeh 1995] schedule the data path and assign voltages to data path operators so as to minimize power given a predetermined set of supply voltages. Logic level conversions are not explicitly modeled in their formulation. Gebotys [Gebotys 1995a] used an integer programming approach to scheduling and partitioning a VLSI system across multiple chips operating at different supply voltages. Johnson [Johnson and Roy 1996] used an integer program to choose voltages from a list of candidates, schedule datapath operations, model logic level conversions, and assign voltages to each operation.

The integer linear program (ILP) presented in [Johnson and Roy 1996] led to the MOVER algorithm to be discussed in this paper. The purely ILP approach was useful because it allowed us to test the problem formulation and obtain provably optimal solutions using a general purpose branch and bound ILP solver. Execution times varied from minutes to days. However, for certain well defined problems, ILP can in fact be very efficient. Gebotys [Gebotys 1992] has shown that for the general precedence constrained scheduling problem, one can specify linear constraints on continuous variables that very closely approximate the boundary of the set of integer solutions. This is a very desirable property because it allows a branch and bound algorithm to finish in a small number of iterations. A difficulty with the ILP approach is that there may be subproblems for which it is very difficult to obtain such tight linear constraints. This often leads to very large execution times. Modeling of logic level conversions proved to be especially difficult in terms of decision variables and constraints.

MOVER attempts to use ILP only to solve those subproblems for which an efficient formulation is known. particular, ILP is used to partition operations into high

and low voltage groups and to evaluate schedule feasibility for particular choices of supply voltages. MOVER searches a user specified range of supply voltages, calling the ILP formulation as needed to evaluate schedule feasibility and obtain all energy estimate. In the remainder of this paper, we will describe the MOVER algorithm, explain the delay and energy dissipation models, discuss IC layout and power supply considerations, present scheduling results for several datapath specifications, make observations and draw conclusions regarding multiple voltage datapaths and the applicability of this algorithm.

2. DATAPATH SPECIFICATIONS

A datapath is specified in the form of a data flow graph (DFG) where each vertex represents an operation and each arc represents a data flow or latency constraint. This DFG representation is similar to the "sequencing graph" representation described by DeMicheli [DeMicheli 1994] except that hierarchical and conditional graph entities are not supported.

The DFG is a directed acyclic graph, $G(V, E)$, with vertex set V and edge set E . Each vertex corresponds one-to-one with an operator in the data path. Each edge corresponds one-to-one with a dependency between two operators: a data flow, a latency constraint, or both. Associated with each vertex is an attribute that specifies the operator type such as adder, multiplier, or null operation (NO-OP). Associated with each edge is an attribute that indicates a latency constraint between the start times of the source and destination operations. A positive value indicates a minimum delay between operation start times. The magnitude of a negative value specifies a maximum allowable delay from the destination to the source. Figure 2 provides a simple example of a datapath specification and defines elements of the DFG notation.

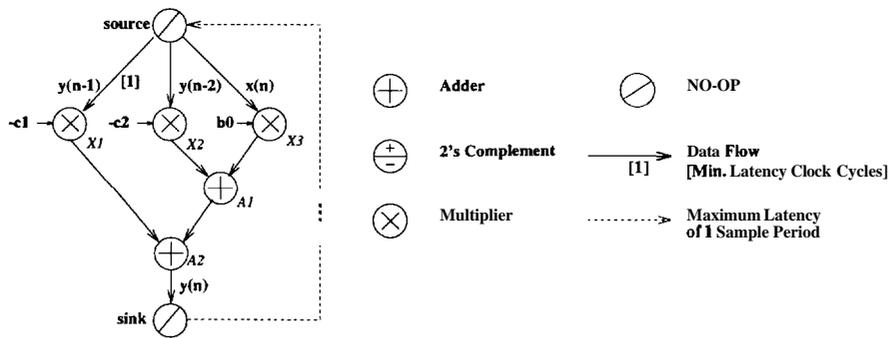


Fig. 2. Sample datapath specification and key to notation

Two types of NO-OP's are used which we will refer to as "transitive" and "non-transitive" NO-OP's. The term "transitive" is used to indicate that a NO-OP propagates signals without any delay or cost. Neither type of NO-OP introduces delay or power dissipation. Both types serve as vertices in the DFG to which latency constraints can be attached. The transitive NO-OP is treated as if signals and their

Table I. Sample datapath constraints

Maximum Multipliers	
Maximum Clock Cycles	$V_{max} = 5V$
	$V_{min} = 1.5V$
Convergence threshold	$V_{conv} = 0.1V$

logic levels are propagated through the NO-OP. Non-transitive NO-OP's and the arcs entering or leaving a non-transitive NO-OP are ignored in the accounting of register delays, level conversions, and voltage supply choices.

3. MOVER SCHEDULING ALGORITHM

MOVER will generate a schedule, select a user specified number of supply voltage levels, and assign voltages to each operation. MOVER uses an ILP method to evaluate the feasibility of candidate supply voltage selections, to partition operations among different power supplies, and to produce a minimum area schedule under latency constraints once voltages have been selected. The algorithm proceeds in several phases. First, MOVER determines maximum and minimum bounds on the time window in which each operation must execute. It then searches for a minimum single supply voltage. Next, MOVER partitions datapath operations into two groups: those which will be assigned to a higher supply voltage and those which will be assigned to a lower supply voltage. The high voltage group is initially fixed to a voltage somewhat above the minimum single voltage. MOVER then searches for a minimum voltage for the lower group. The voltage of the lower group is fixed. A new minimum voltage for the upper group is sought. To find a three supply schedule, partition the lower voltage group and search for new minimum voltages for bottom, middle, and upper groups.

Let us use the datapath shown in figure 2 to illustrate the process. Let the scheduling constraints be as specified in table I. Maximum clock cycles indicate the user specified maximum number of control steps. The convergence threshold, V_{conv} , determines when the voltage search mechanism will accept a candidate voltage; the candidate must be known to be within one threshold of the minimum voltage.

Table II describes how MOVER would typically process this simple example. V_1 is the minimum single supply voltage. V_{2h} and V_{2l} are the minimum voltages given two supplies. V_{3h} , V_{3m} , and V_{3l} are the minimum voltages given three supplies. Please note that the voltage search shown in step two is simplified somewhat from the actual search process, but it conveys the concept. A more precise description of the voltage search is given in section 3.6. Figure 3 presents examples of the type of schedules that would be available at the completion of steps 2, 6, and 12.

3.1 ILP Formulation

At the core of MOVER is an integer linear program (ILP) that is used repeatedly to evaluate possible supply voltages, partition operations between different power supplies, and produce a schedule that minimizes resource usage. A single ILP

Table II. MOVER Scheduling Example

1.	Determine maximum range of start times for each operation by generating an as soon as possible (ASAP) schedule and an as late as possible (ALAP) schedule.	$X1 \in [1, 4], X2 \in [0, 3], X3 \in [0, 3]$ $A1 \in [1, 4], A2 \in [2, 5]$
2.	Search for minimum single supply	V_{test} feasible? V_{hi} V_{lo}
	Initial condition	5V 1.5V
	1st Candidate voltage	3.3V No 5V 3.3V
	Infeasible, so try higher	4.1V Yes 4.1V 3.3V
	Feasible, so try lower	3.7V Yes 3.7V 3.3V
	Feasible, so try lower	3.5V No 3.7V 3.5V
	Infeasible, try higher	3.6V Yes 3.6V 3.5V
	$V_{hi} - V_{lo} < V_{conv}$ So let $V_1 = 3.6V$	
3.	Partition operations between two power supplies	High voltage operations: A1,A2 Low voltage operations: X1,X2,X3
4.	Insert logic level conversions into delay and energy model.	Level conversions required between X1 and A2, X2 and A1, X3 and A1
5.	Temporarily fix high voltage	$V_{2h} = \frac{1}{2} \times (V_{max} + V_1)$
6.	Search for minimum lower supply in same manner as step 2 and then fix that voltage	$V_{min} \leq V_{2l} \leq V_1$ Result: $V_{2l} = 2.4$
7.	Search for minimum higher supply and fix voltages	$V_{2l} \leq V_{2h} \leq \text{previous } V_{2h}$ Result: $V_{2h} = 3.7V$
8.	Partition operations from lower group into middle (V_{3m}) and bottom (V_{3l}) voltage groups.	Operations in top group (A1,A2) unchanged. Middle group: X1,X2. Bottom Group: X3.
9.	Insert logic level conversions into delay and energy model	No new logic level conversions required in this example.
10.	Temporarily fix top voltage	$V_{3h} = \frac{1}{2} \times (V_{max} + V_{2h})$
11.	Temporarily fix middle voltage	$V_{3m} = V_{2h}$
12.	Search for minimum low supply	Result: $V_{3l} = 1.9V$
13.	Search for min. middle supply	Result: $V_{3m} = 2.5V$
14.	Search for minimum top supply	Result: $V_{3h} = 3.8V$

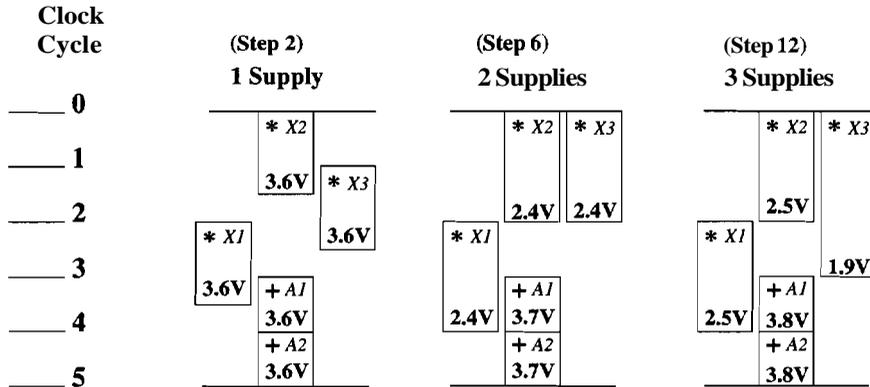


Fig. 3. Sample Schedules

formulation serves all three purposes. In each case, MOVER analyzes the DFG and generates a collection of linear inequalities that represent precedence constraints, timing constraints, and resource constraints for the datapath to be scheduled. A weighted sum of the energy dissipation for each operation is used as the optimization objective when partitioning operations or evaluating the feasibility of a supply voltage. A weighted sum of resource usage serves as the optimization objective when minimizing resources. The inequalities and objective function are packed into a matrix of coefficients that are fed into an ILP program solver (CPLEX). MOVER interprets the results from CPLEX and annotates the DFG to indicate schedule times and voltage assignments.

The architectural model assumed by MOVER is depicted in Figure 4. All operator outputs have registers. Each operator output feeds only one register. That register operates at the same voltage as the operator supplying its input. All level conversions, when needed, are performed at operator inputs.

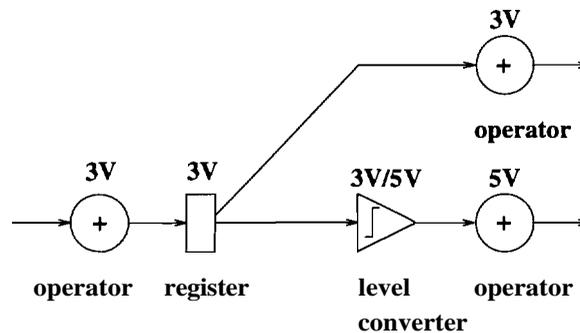


Fig. 4. MOVER architectural model

MOVER's ILP formulation works on a DFG where voltage assignments for some operations may already be fixed. For operations not already fixed to a voltage,

the formulation chooses between two closely spaced voltages so as to minimize energy. The voltages are chosen to be close enough together that level conversions from one to the other can be ignored. Consequently, level conversions only need to be accounted between operations fixed to different voltages and on interfaces between fixed and unfixed operations. Figure 5 gives examples of situations in which MOVER would or would not insert level conversions. Question marks in the figure represent operations that have not yet been fixed to a voltage.

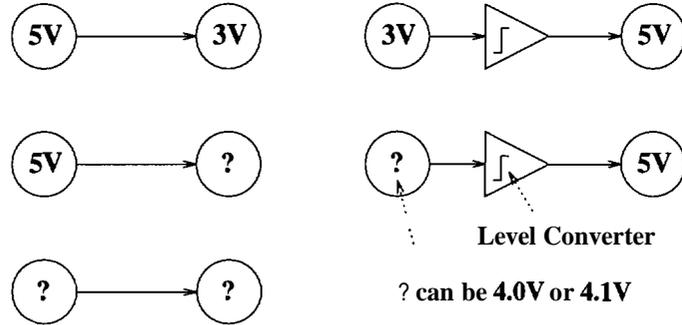


Fig. 5. Where MOVER inserts level converters

3.2 ILP Decision Variables

Three categories of decision variables are used in the MOVER ILP formulation. One set of variables of the form $x_{i,l,s}$ indicates the start time and supply voltage assignment for each operator that has not already been fixed to a particular supply voltage. $x_{i,l,s} = 1$ indicates that operation i begins execution on clock cycle l using supply voltage s . Under any other condition, $x_{i,l,s}$ will equal zero. The supply voltage selection is limited to two values where $s = 1$ selects the lower and $s = 2$ selects the higher candidate voltage. Another set of variables, $x_{i,l}$, indicates the start time of operations for which the supply voltage has been fixed. $x_{i,l} = 1$ indicates that operation i starts at clock cycle l . Under any other condition, $x_{i,l}$ will equal zero. The last group of variables, $a_{i,m}$, indicates the allocation of operator resources to each possible supply voltage. $a_{i,m}$ will be greater than or equal to the number of resources of type m that are allocated to supply voltage s . In this case, s can be an integer in the range $(1, \# \text{ fixed supplies} + 2)$. $s \in (1, 2)$ corresponds to the new candidate supply voltages. $s > 2$ corresponds to supply voltages that have already been fixed.

3.3 Lookup tables for delay and energy

Delay and energy estimates are tabulated as a function of supply voltage prior to solving the ILP formulation. Section 4 describes the delay and energy calculations used to fill the lookup tables. The functions $onrg()$, $rnrng()$, and $cnrg()$ were defined to look up energy values from those tables and scale the result as a function of load capacitance and switching activity. $del_{i,s}$, gives the delay of each operation i as a function of supply voltage s .

$onrg(j, s_j, c_{load})$ returns the energy estimate for operation j , using supply voltage s_j , with a load capacitance of c_{load} at the output. $rnrg(s_j, fanout_j)$ returns the energy estimate for a register using supply voltage s_j and an output load capacitance of $fanout_j$. $fanout_j$ reflects the level of fanout from operation j in the DFG. $cnrg(s_i, s_j, c_j)$ returns the energy estimate for a level conversion from a block operating at supply voltage s_i to a block operating at supply voltage s_j . c_j is the input capacitance of operation j . $deli,,$ gives the delay of operation i including register propagation and level conversion delays.

3.4 Objective Functions

The objective function (equation 1) estimates the energy required for one execution of the data path as a function of the voltage assigned to each operation. Consider the energy expression split into two parts. The first nested summation counts the total energy contribution associated with operations not already fixed to a supply voltage. The second nested summation counts the total energy contribution of operations that are already fixed to a particular supply voltage.

For each operation j that has not been fixed to a supply voltage (e.g., $j \in V_{free}$), the first nested summation accumulates the energy of operation j ($onrg(j, s_j, c_{reg})$), the register at the output of operation j ($rnrg(s_j, fanout_j)$), and any level conversions required at the input to j ($cnrg_{free}(j, s)$). The decision variables $x_{j,l,s}$ are used to select which lookup table values for operator, register, and level conversion energy are added into the total energy. We must sum over both candidate supply voltages s_j and all clock cycles l in the possible execution time window R_j of operation j . E_{conv} is the set of DFG arcs that may require a level conversion, depending on voltage assignments. V_{oper} is the set of DFG vertices that are not NO-OPs. V_{fix} is the set of DFG vertices (operations) that have been fixed to a particular voltage. V_{free} is the set of vertices that have not previously been fixed to a voltage.

For each operation j that has been fixed to a supply voltage, we again accumulate the energy of each operation, register, and level conversion. The only difference from the expression for free operations is that now all voltages in the expression are constants determined prior to solving the ILP formulation. Consequently, the index s_j can be removed from the summation and the decision variable x .

$$\begin{aligned} \text{Energy} = & \sum_{j \in V_{free} \cap V_{oper}} \sum_{l \in R_j} \sum_{s=1}^2 x_{j,l,s} \times (onrg(j, v_s, c_{reg}) + rnrg(v_s, c_{fanout(i)}) + cnrg_{free}(j, s)) \\ & + \sum_{j \in V_{fix} \cap V_{oper}} \sum_{l \in R_j} x_{j,l} \times (onrg(j, v_j, c_{reg}) + rnrg(v_j, c_{fanout(i)}) + cnrg_{fix}(j)) \end{aligned} \quad (1)$$

$cnrg_{free}(j, s)$ and $cnrg_{fix}(j)$ represent the level conversion energy at the input of free and fixed operations respectively.

$$cnrg_{free}(j, s) = \sum_{i|(i,j) \in E_{conv} \text{ and } i \in V_{fix}} cnrg(v_i, v_s, c_{in,i}) + \sum_{i|(i,j) \in E_{conv} \text{ and } i \in V_{free}} cnrg(v_1, v_s, c_{in,i}) \quad (2)$$

$$cnrg_{fix}(j) = \sum_{i|(i,j) \in E_{conv} \text{ and } i \in V_{fix}} cnrg(v_i, v_j, c_{in_j}) + \sum_{i|(i,j) \in E_{conv} \text{ and } i \in V_{free}} cnrg(v_1, v_j, c_{in_j}) \quad (3)$$

Equation 4 is the objective function used when minimizing resource usage. Here, $a_{m,s}$ indicates the minimum number of operators of type m with supply voltage s needed to implement a datapath. Each operation of type m is considered to have an area of $area_m$. M_{oper} represents the set of all operation types excluding NO-OPs. The summation accumulates an estimate of the total circuit resources required to implement a datapath.

$$area = \sum_{m \in M_{oper}} \sum_{s \in S} area_m \times a_{m,s} \quad (4)$$

3.5 ILP Constraint Inequalities

Equation 5 guarantees that only one start time l is assigned to each operation i for which the supply voltage is already fixed. Equation 6 guarantees that only one start time l and supply voltage s can be assigned to each operation i that does not have a supply voltage assignment.

$$\sum_{l \in R_i} x_{i,l} = 1 \quad \forall i \in V_{fix} \quad (5)$$

$$\sum_{s=1}^L \sum_{l \in R_i} x_{i,l,s} = 1 \quad \forall i \in V_{free} \quad (6)$$

Equation 7 guarantees that the voltage of a transitive NO-OP j matches the voltage of all operations supplying an input to the transitive NO-OP. V_{trnoop} is the set of vertices in the DFG corresponding to transient NO-OP's. E is the set of all arcs in the DFG.

$$\sum_l (x_{i,l,s} - x_{j,l,s}) = 0 \quad \forall j \in V_{trnoop} \quad \forall i|(i,j) \in E \quad (7)$$

Equations 8 through 11 enforce precedence constraints specified in the DFG. All are adaptations of the structured precedence constraint shown by Gebotys [Gebotys 1992] to produce facets of the scheduling polytope. Each arc (i,j) with a latency $lat_{i,j} \geq 0$ specifies a minimum latency from the start of operation i to the start of operation j . Equation 8 defines the set of precedence constraint inequalities corresponding to DFG arcs where the source and destination operations are both free (not fixed to a voltage). The remaining equations are simplifications of equation 8. Equation 9 handles the case where the source operation is free and the destination is fixed. Equation 10 handles fixed source operations with free destination operations. Equation 11 handles the case where both operations are fixed.

$$\sum_{s_j=1}^{del_{i,s_i}+l} \sum_{l_1=0}^{maxclk} x_{j,l_1,s_j} + \sum_{l_2=l}^{maxclk} x_{i,l_2,s_i} \leq 1 \quad (8)$$

$$\forall i, j \in V_{free}, \forall s_i \in (1, 2), \forall l \in L, \forall (i, j) \mid lat_{i,j} \geq 0$$

$$\sum_{l_1=0}^{del_{i,s_i}+l} x_{j,l_1} + \sum_{l_2=l}^{maxclk} x_{i,l_2,s_i} \leq 1 \quad (9)$$

$$\forall i \in V_{free}, \forall j \in V_{fix}, \forall s_i \in S_{cand}, \forall l \in L, \forall (i, j) \mid lat_{i,j} \geq 0$$

$$\sum_{s_j=1}^2 \sum_{l_1=0}^{del_{i,s_i}+l} x_{j,l_1,s_j} + \sum_{l_2=l}^{maxclk} x_{i,l_2} \leq 1 \quad (10)$$

$$\forall i \in V_{fix}, \forall j \in V_{free}, \forall s_i \in S_{cand}, \forall l \in L, \forall (i, j) \mid lat_{i,j} \geq 0$$

$$\sum_{l_1=0}^{del_{i,s_i}+l} x_{j,l_1} + \sum_{l_2=l}^{maxclk} x_{i,l_2} \leq 1 \quad (11)$$

$$\forall i, j \in V_{fix}, \forall l \in L, \forall (i, j) \mid lat_{i,j} \geq 0$$

Equations 12 through 15 enforce maximum constraints specified in the DFG. Each arc (i, j) with a latency $lat_{i,j} < 0$ specifies a maximum delay from operation j to operation i . Equation 12 defines the set of maximum latency constraint inequalities corresponding to arcs where the source and destination operations are both free (not fixed to a voltage). The remaining equations are simplifications of equation 12. Equation 13 handles the case where the source operation is free and the destination is fixed. Equation 14 handles fixed source operations with free destination operations. Equation 15 handles the case where both operations are fixed.

$$\sum_{s_j=1}^2 x_{j,l,s_j} + \sum_{s_i=1}^2 \sum_{l_2=l-lat_{i,j}+1}^{maxclk} x_{i,l_2,s_i} \leq 1 \quad (12)$$

$$\forall i, j \in V_{free} \forall l \in L, \forall (i, j) \mid lat_{i,j} < 0$$

$$x_{j,l} + \sum_{s_i=1}^2 \sum_{l_2=l-lat_{i,j}+1}^{maxclk} x_{i,l_2,s_i} \leq 1 \quad (13)$$

$$\forall i \in V_{free}, \forall j \in V_{fix} \forall l \in L, \forall (i, j) \mid lat_{i,j} < 0$$

$$\sum_{s_j=1}^2 x_{j,l,s_j} + \sum_{l_2=l-lat_{i,j}+1}^{maxclk} x_{i,l_2} \leq 1 \quad (14)$$

$$\forall i \in V_{fix}, \forall j \in V_{free} \forall l \in L, \forall (i,j) \mid lat_{i,j} < 0$$

$$x_{j,l} + \sum_{l_2=l-lat_{i,j}+1}^{maxclk} x_{i,l_2} \leq 1 \quad (15)$$

$$\forall i, j \in V_{fix}, \forall l \in L, \forall (i,j) \mid lat_{i,j} < 0$$

Equations 16 and 17 ensure that resource usage during each time step does not exceed the resource allocation given by $a_{m,s}$. The expressions on the left computes the number operations of type m with supply voltage s that are executing concurrently during clock cycle l . $a_{m,s}$ indicates the number of type m resources that have been allocated to supply voltage s . Equation 16 enforces the resource constraint for free operations. Equation 17 enforces the constraint for fixed operations. Free operations are allowed to take on one of two candidate voltages.

$$\sum_{i \mid i \in V_{free}, type(i)=m} \sum_{l_1=l-del_{i,s_i}+1}^l x_{i,l_1,s_i} \leq a_{m,s_i} \quad (16)$$

$$\forall m \in M_{oper}, \forall l \in L, \forall s_i \in (1, 2)$$

$$\sum_{i \mid i \in V_{fix}, type(i)=m, supply(i)=s_i} \sum_{l_1=l-del_{i,s_i}+1} x_{i,l_1} \leq a_{m,s_i} \quad (17)$$

$$\forall m \in M_{oper}, \forall l \in L, \forall s_i > 2$$

Equation 18 enforces the user specified resource constraints. $maxres(m)$ represents the total number of resources of type m (regardless of voltage) that can be permitted. The left side expression accumulates the number of resources of type m that have been allocated to all supply voltages. The total is not allowed to exceed the user specified number of resources.

$$\sum_{s \in S} a_{m,s} \leq maxres(m) \quad \forall m \in M_{oper} \quad (18)$$

3.6 Voltage search

MOVER searches a continuous range of voltages when seeking a minimum voltage one, two, or three power supply design. The user must specify a convergence

Table III. Voltage search algorithm

1.	Choose starting voltages V_2 and $V_1 = V_2 - V_{conv}$ where $V_{max} \leq V_2 \leq V_{min}$
2.	Create matrix of ILP constraint inequalities.
3.	Obtain minimum energy solution to inequalities. The solution will provide a schedule, a mapping of V_1 or V_2 to each operator, an energy estimate, and an area estimate for the datapath.
4.	If a solution was found, then
4a.	If most operations were assigned to V_1 , then Choose new candidate voltages midway between V_1 and V_{lo} . Set $V_{hi} = V_2$. Go to step 2.
4b.	else There must be little or no benefit to assigning operations to V_1 Fix all operations to V_2 DONE!
4c.	else (if the problem was infeasible) Choose new candidate voltages midway between V_2 and V_{hi} . Set $V_{lo} = V_2$ Go to step 2.

threshold V_{conv} that is used to determine when a voltage selection is acceptably close to minimum. Let V_{hi} and V_{lo} represent the current upper and lower bound on the supply voltage (as in table II). The initial values of V_{hi} and V_{lo} will be set as described in table II.

When searching for a minimum single supply voltage, all operations are initially considered to be free (not fixed to a voltage). When searching for a minimum set of two or three supply voltages, MOVER considers one power supply at a time. The voltage will be fixed for any operations not allocated to the supply voltage under consideration. Table III outlines the voltage search algorithm.

3.7 Partitioning

Partitioning is the process by which MOVER takes all free operations in the DFG and allocates each to one of two possible power supplies. Partitioning is not performed until a single minimum supply voltage is known for the group of operations. Let V_1 represent the minimum supply voltage for the free operations. Choose two candidate supply voltages (V_a and V_b) one slightly above V_1 and the other, slightly below.

$$V_a = V_1 - \frac{V_{conv}}{2} \tag{19}$$

$$V_b = V_1 + \frac{V_{conv}}{2} \tag{20}$$

Set up the ILP constraint inequalities. Obtain a minimum energy schedule. Operations will only be assigned to V_a if there is schedule slack available. There may be several ways that the operations can be partitioned. In such a case, the optimal ILP solution will maximize the energy dissipation of the lower voltage group (i.e., put the most energy hungry operations in the lower voltage group). This will tend to maximize the benefit from reducing the voltage of the lower group.

Given a successful partition, operations assigned to V_a will be put into the lower supply voltage group and operations assigned to V_b will be put into the higher supply voltage group. Let $count(V_x)$ represent the number of operations allocated to voltage V_x . Let $MinPartition$ represent the user specified minimum allocation ratio for a successful partition. Then the partition is considered successful if the following two conditions are satisfied:

$$\frac{count(V_a)}{count(V_a) + count(V_b)} \geq MinPartition \quad (21)$$

$$\frac{count(V_a)}{count(V_a) + count(V_b)} < 1 \quad (22)$$

The partition can fail at least three ways.

- (1) All operations were allocated to the lower supply voltage.
- (2) All or nearly all operations were allocated to the higher supply voltage.
- (3) The ILP solver exceeded some resource or time limit.

The first situation indicates that the minimum single voltage could have been a bit lower. In this event, MOVER lowers the values of V_a and V_b by $\frac{V_{conn}}{2}$ and tries the partition again. Lowering V_a and V_b too far will lead to a completely infeasible ILP problem. The second situation indicates that there is not enough schedule slack available for any operations to bear a further reduction in voltage. In this case, MOVER terminates. The only remedies for the third situation are to either increase resource and time limits on the ILP solver or make the problem smaller.

4. CHARACTERIZATION OF DATAPATH RESOURCES

The results presented in this paper make use of four types of circuit resources: an adder, multiplier, register, and level converter. MOVER requires models of the energy and delay of each type of resource as a function of supply voltage, load capacitance, and average switching activity. The input capacitance of each resource type is required in order to determine load capacitances within a datapath design. For each type of resource, an HSPICE netlist was created. 0.8 micron MOSIS library models were used with the level 3 MOS transistor model. Energy dissipation and worst case delays were measured from simulation results. Energy dissipation is assumed to scale proportionally to input switching activity. Input capacitance in each case was determined by inserting a series resistance at input nodes and then measuring input rise time in response to a step function. The results of all measurements were used to compute model parameters provided as

Table IV. Nominal energy and delay values used by MOVER

Resource Type	Energy [pJ]	$\frac{d}{dC}$ Energy [pJ/pF]	Delay [ns]	$\frac{d}{dC}$ Delay [ns/pF]	C_{in} [pF]
ADDER	84	200	12.0	3.5	0.021
MULTIPLIER	2966	200	18.5	3.33	0.095
REGISTER	312	200	0.48	2.25	0.045

input to MOVER. In this section we will discuss the particulars of how the delay and energy characteristics of each resource type were measured and modeled with respect to supply voltage and load capacitance.

4.1 Datapath operators and registers

16 bit adders and multipliers were simulated with a supply voltage of 5V, average input switching activities of 50% and a nominal load capacitance of 0.1pF on each output pin. Total average power dissipation was measured. The average energy per clock cycle was then computed and provided as input to MOVER. Registers were characterized in a similar manner, except that a single bit register was simulated for a few clock cycles. The register energy dissipation was then scaled to represent 16 bit, 50% switching activity conditions. Worst propagation delays through the adder and multiplier were measured at 5V supply and 0.1pF load on each output. Delays were also measured at 0.2pF load in order to measure the scaling of delay with respect to load. Delay is modeled as scaling linearly with respect to the load capacitance.

Power dissipation (E) for each operator and register scales with respect to supply voltage as

$$\frac{E}{E_0} = \frac{V^2}{V_0^2} \quad (23)$$

where E_0 is the energy dissipation of the operator or register measured at the nominal supply voltage V_0 .

Delay (t_p) for each operator and register scale with respect to supply voltage as

$$\frac{t_p}{t_{p_0}} = \frac{V}{V_0} \times \frac{(V_0 - |V_T|)^2}{(V - |V_T|)^2} \quad (24)$$

where t_{p_0} is the propagation delay measured at the nominal supply voltage V_0 . The power and delay scaling factors were derived directly from the CMOS power and delay equations described by Rabaey [Rabaey 1996].

Table IV gives the model parameters used by MOVER for each type of resource. Note that the register delay given here is just the propagation time relative to a clock edge. Register setup time is treated as part of the datapath operator delays. The nominal values are for $V_{DD} = 5V$, $C_{Load} = 0.1pF$, 16 bit wide operations, and input switching activities of 50%. Energy values are given as the average per clock cycle.

4.2 Level conversion

Whenever one resource has to drive an input of another resource operating at a higher voltage, a level conversion is needed at the interface. Four alternatives were considered to accomplish this: omit the level converter, use a chain of inverters at successively higher voltages, use an active or passive pullup, or use a dual cascode voltage switch (DCVS) circuit as a level converter [Chandrakasan et al. 1994; Usami and Horowitz 1995]. We omit the level converter for step-down conversions and use the DCVS circuit for step-up conversions. Given appropriate transistor sizes, this circuit exhibits no static current paths and it can operate over a full 1.5V to 5.0V range of input and output supply voltages.

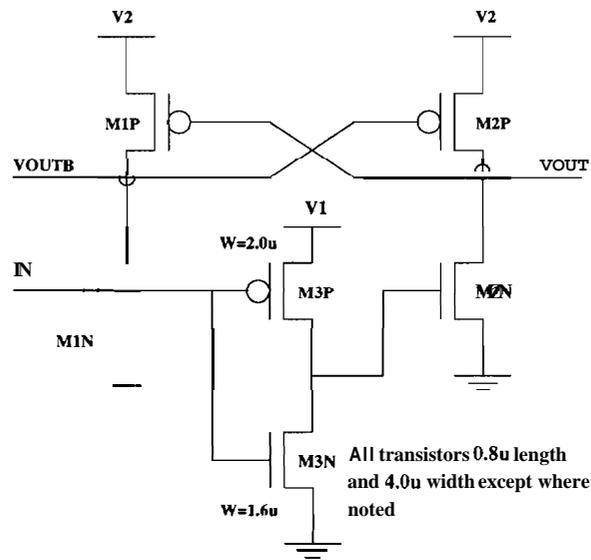


Fig. 6. DCVS Level Converter

Another option is to combine the register and level converters together. This approach was documented by Usami and Horowitz [Usami and Horowitz 1995]. The combined register and level conversion was found to dissipate only 10% more power than the register alone.

A model was needed that could accurately indicate the power dissipation and propagation delay of the DCVS level converter as a function of the input logic supply voltage V_1 , output logic supply voltage V_2 , and load capacitance. The circuit was studied both analytically and from HSPICE simulation results to determine a suitable form for the model equations. Coefficients of the equations were then calibrated so that the model equations would produce families of curves closely matching simulation results for V_1 ranging from 1.5V to 5V and $V_1 + V_T \leq V_2 \leq 5V$. These are the ranges of supply voltages for which a level converter is needed. Typical energy dissipation of the level converter was found to be on the order of 5 to 15pJ per switching event per bit, given a 0.1pF load. Typical propagation delays range were approximately 1ns for level conversions such as 3.3V to 5V or 2.4V to 3.3V.

Propagation delays become large as the input voltage of the level converter falls towards $2V_T$. A 2.5V to 5V conversion had a delay of about 2.5ns. A 2V to 5V conversion had a delay of nearly 5ns.

4.2.1 Power Dissipation Model. The power dissipation model is separated into three factors. The first factor calculates the power consumption for $V_1 = V_2$. Charging and discharging of the load capacitance contributes a V_2^2 term to the power. The short circuit current on the paths through M1P/M1N and M2P/M2N contribute power as a third order polynomial of V_2 .

$$DCVSPWR(V_2, V_2) = \tag{25}$$

$$(a3 \times V_2^3 + a2 \times V_2^2 + a1 \times V_2 + a0)$$

The coefficients $a3$ through $a0$ are obtained by means of a polynomial curve fit to a plot of circuit power vs. V_2 .

The next factor estimates the ratio of increase in power consumption due to V_1 being less than V_2 .

$$DCVSPWR(V_1, V_2) = \tag{26}$$

$$DCVSPWR(V_2, V_2) \times (b0 + b1 \times \frac{V_2 - V_T}{(V_1 - V_T)^2}) + b2 \times V_1^2$$

$b0$ represents the portion of power dissipation not affected by V_1 . The fractional expression models the effect of $V_1 < V_2$. When $V_1 < V_2$, M2N is in saturation until V_{OUT} drops to $V_2 - V_T$. Shortly thereafter, the cross-coupled circuit switches and M2P turns off. The fractional expression in $DCVSPWR(V_1, V_2)$ models the effect of saturation current in the pull-down transistors on the duration of short circuit current. The final term represents the power consumption in the inverter.

The power model is scaled linearly for load capacitance. All of the analytical expressions for DCVS power dissipation showed a linear dependence on load capacitance. Plots of power dissipation versus load capacitance showed an almost perfect linear dependence on the load. Furthermore, if one chooses a nominal load capacitance (C_{L0}) to evaluate power dissipation, the slope of power versus capacitance is found to be proportional to the power dissipation ($pw0$) at the nominal load. $dpdc$ is the slope of power versus capacitance for the values of V_1 and V_2 for which $pw0$ was measured. The following expression models this dependence on load capacitance.

$$DCVSPWR(V_1, V_2, C_L) = \tag{27}$$

$$DCVSPWR(V_1, V_2) \times (1 + dpdc \times \frac{(C_L - C_{L0})}{pw0})$$

4.3 Delay Model

The delay model hinges on the following observation of delay versus V_2 for fixed values of V_1 . For $V_2 > V_1 + V_T$, delay increases almost linearly with respect to V_2 . More importantly, the delay versus V_2 lines all intersect at nearly the same point on a graph. To take advantage of this behavior, a polynomial curve fit to $\frac{1}{delay}$ was used to estimate the position of a point on the linear portion of each delay versus V_2 curve. In particular, data points corresponding to $V_2 = V_1 + V_T$ were used. The expression for $DCVSDEL(V_1, V_1 + V_T)$ estimates these data points.

$$DCVSDEL(V_1, V_1 + V_T) = \tag{28}$$

$$\frac{1}{d3 \times V_1^3 + d2 \times V_1^2 + d1 \times V_1 + d0}$$

The expression for $DCVSDEL(V_1, V_2)$ models the radial behavior of the delay versus V_2 curves. $(V_0, del0)$ specifies the point from which the lines radiate.

$$DCVSDEL(V_1, V_2) = \tag{29}$$

$$\frac{DCVSDEL(V_1, V_1 + V_T) - del0}{V_1 + V_T - V_0} \times (V_2 - V_0) + del0$$

Delay scales with respect to load capacitance in a manner identical to that described for power versus capacitance.

5. RESULTS

5.1 Datapath examples

ILP schedule optimization results are presented for six example data paths: a four point FFT (FFT4), the 5th order elliptic wave filter benchmark (ELLIP) [Rao 1992], a 6th order Auto-Regressive Lattice filter (LATTICE), a frequency sampled filter (FSAMP) with three 2nd order stages and one 1st order stage, a direct form 9 tap linear phase FIR filter (LFIR9), and a 5th order state-space realization of an IIR filter (SSIIR). In the FFT data path, complex signal paths are split into real and imaginary data flows. For all other data paths, the signals are modeled as non-complex integer values. All data flows were taken to be 16 bits wide. Switching activities at all nodes were assumed to be 50%, i.e., the probability of a transition on any selected 1 bit signal is 50% in any one sample interval.

Each example was modeled for one sample period with data flow and latency constraints specified for any feedback signals. No conditional operations were modeled. Any loops that start and finish within the same sample period were completely unrolled. Any loops spanning multiple sample periods were broken. A data flow passing from one sample period to the next was represented by input and output nodes in the DFG connected by a backward arc to specify a maximum latency

constraint from the input to the output. A 20ns clock was specified for all examples. Latency constraints were specified so that the data introduction interval equals the maximum delay from the input to the output of the data path.

5.2 MOVER Results

The MOVER algorithm was exercised for each datapath topology (FFT4, ELLIP, LATTICE, FSAMP, LFIR9, and SSIIR) under a variety of latency and resource constraints.

Figure 7 presents energy reduction results. The left-most column identifies the particular datapath topology and indicates the number of operations (additions, multiplications, and sample period delays) performed in one iteration of the datapath. "Max Lat/Clks" specifies the maximum latency (equal to the data sample rate) and the maximum number of control steps (Clks), both given in terms of the number of clock cycles. "Max +/-" specifies the maximum numbers of adder and multiplier circuits permitted in the design. Values of "-/-" indicate that unlimited resources were permitted. The columns headed by "Voltages 1 2 3" indicate the supply voltages selected by MOVER. A "-" is used to fill voltage columns "2" or "3" in those cases where a one or two supply voltage result is presented. The string "NR" in voltage columns "1" and "2" indicates that a solution with two supply voltages could not be obtained. "NR" in all three columns indicates that a solution with three supply voltages could not be obtained. The "Exec" column reports the minutes of execution time (Real, not CPU) required to obtain the result. The number in parenthesis identifies the type of machine used to obtain the result. "(1)" indicates a SPARCserver 1000 with 4 processors and 320MB of RAM. "(2)" indicates a Sparc 5 with 64MB of RAM.

The bar graph down the center represents the normalized energy consumption of each test case. Each energy result is divided by the single supply voltage, unlimited resource, minimum latency result to obtain a normalized value. Single supply voltage results are shown with black bars. All other results are shown in gray. This style of presentation is intended to visually emphasize the effect of different latency, resource, and supply voltage constraints on the energy estimate. The right-most column presents the absolute energy estimate in units of 10⁻¹² Joules (pJ).

Figure 8 presents area penalty results. All but two columns have the same meaning as the corresponding columns in figure 7. The only exceptions are the bar graph and the "area" column on the right. The "area" value is a weighted sum of the minimum circuit resources required to implement the datapath schedule. The resources (all 16 bits wide) were weighted as follows: adder=1, multiplier=16, register=0.75, and level converter=0.15. These weights are proportional to the transistor count of each resource. Each area value was divided by the area estimate for the corresponding single voltage result. Each single voltage result is shown as a black bar. Two and three voltage results are shown in gray.

5.3 Observations

The preceding results permit several observations to be made regarding the effect of latency, circuit resource, and supply voltage constraints on energy savings, area costs, and execution time. Because our primary objective has been to minimize energy dissipation through use of multiple voltages, we are especially interested in

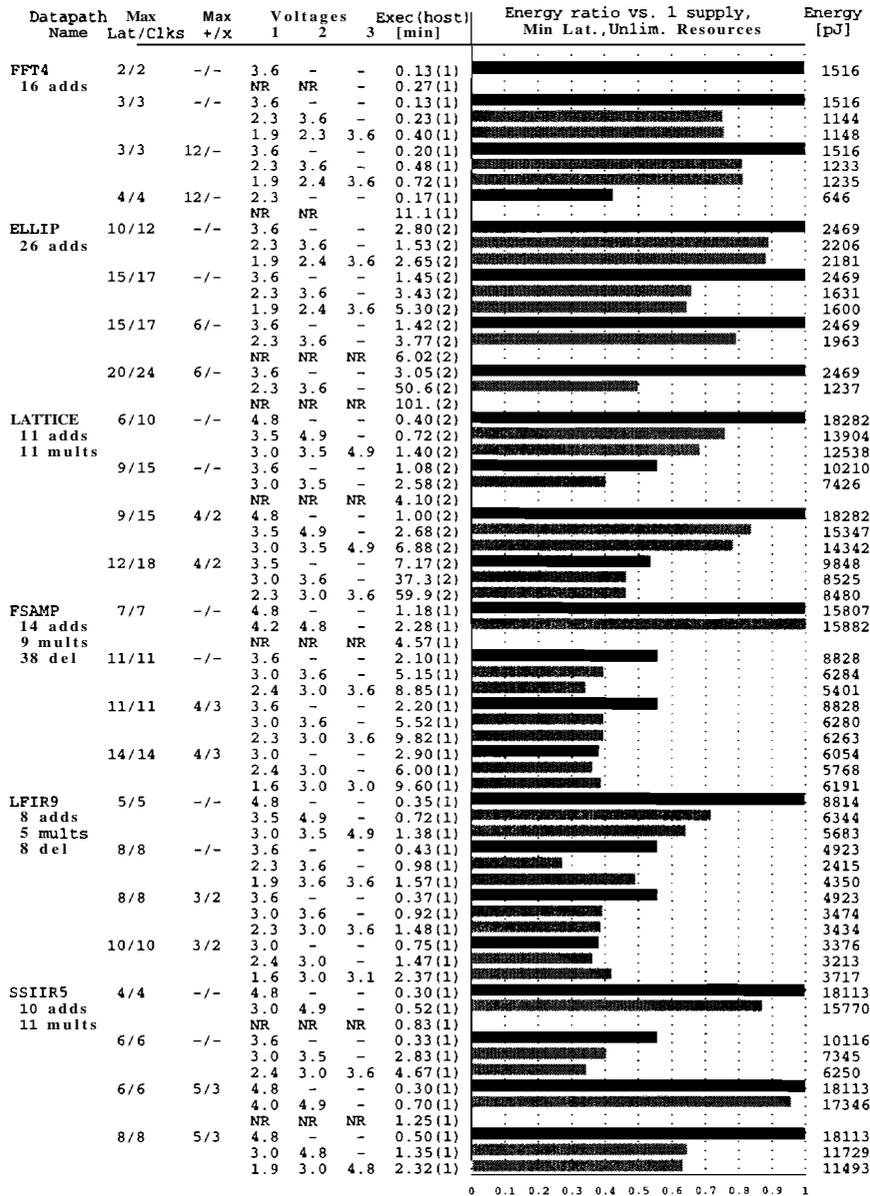


Fig. 7. Multi-voltage Energy Savings

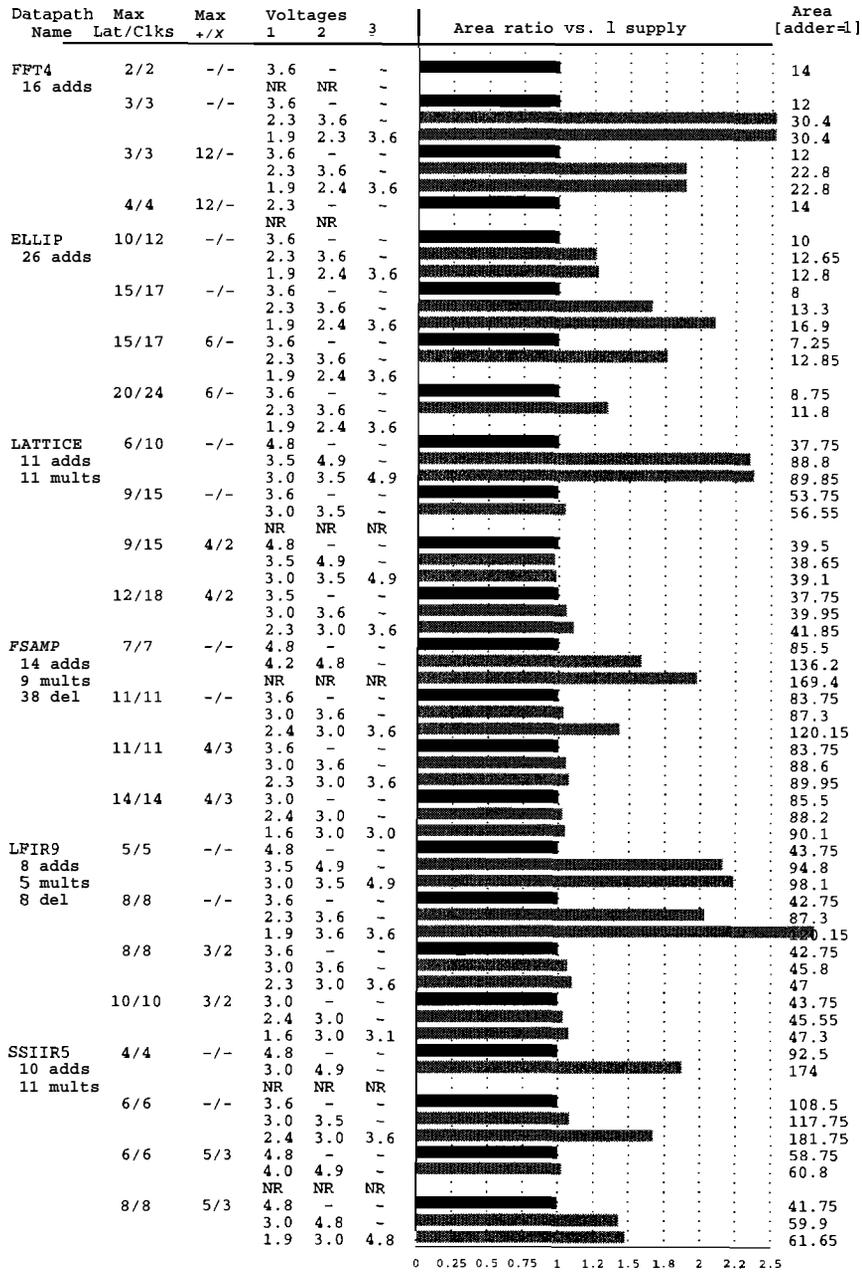


Fig. 8. Multi-voltage Area Penalties

the comparison of multiple supply voltage results to minimum single supply voltage results. Energy savings ranging from 0% to 50% were observed when comparing multiple to single voltage results. Estimated area penalties ranged from a slight improvement to a 170% increase in area. Actual area penalties could be higher, since our estimate only considers the number of circuit resources used. There is not a clear correlation between energy savings and area penalty when looking at the complete set of results. Sometimes a substantial energy savings was achieved with minimal increased circuit resources, other times even a small energy savings incurred a large area cost.

If we consider the impact of latency constraints alone, effects on area and energy are easier to observe. In most cases, multiple voltage area penalties were greatest for the minimum latency unlimited resource test cases. We can also observe that increasing latency constraints always led to the same or lower energy for a given number of supply voltages. However, the effect of latency constraints on the single vs. multiple voltage trade-off varied greatly from one example to another. Results for multiple voltages are most favorable in situations where the single supply voltage solution did not benefit from increased latency, perhaps due to a control step bottleneck such as illustrated earlier in figure 1.

The effect of resource constraints on energy savings are also relatively easy to observe. Not surprisingly, resource constraints tended to produce the lowest area penalties. The only reason for any area penalty at all in the resource constrained case is that sometimes the minimum single supply solution does not require all of the resources that were permitted. Tightening resource constraints always led to energy estimates that were either the same or worse than the corresponding unlimited resource case.

Program execution time was affected most by the latency, control step, and resource constraints. 40% of the minimum voltage (1, 2, and 3 supply) schedules were obtained in a minute or less. **93%** of the results were obtained in 10 minutes or less. The remaining 7% took anywhere from 37 to 101 minutes. All of execution times less than a minute occurred for test cases with 10 or fewer control steps. The largest execution times occurred for test cases where resource constraints were applied and a much larger number of control steps were available. The impacts of latency and control steps are likely due a greatly increased number of decision variables and precedence constraint inequalities. The resource constraints can cause the linear solution space to not fit the integer solution space quite so tightly. This can lead to a much larger integer solution search tree for the branch and bound ILP solver.

6. DESIGN ISSUES

There are several design issues that can be taken into account in order to make MOVER results more useful. In particular, the effects of multiple voltage operation on IC layout and power supply requirements should to be considered during design optimization. In the following sections we will identify some of the impacts and consider how MOVER might be enhanced to take them into account.

6.1 Layout

Following are some ways that multiple voltage design may affect IC layout.

- (1) If the multiple supplies are generated off-chip, additional power and ground pins will be required.
- (2) It may be necessary to partition the chip into separate regions, where all operations in a region operate at the same supply voltage.
- (3) Some kind of isolation will be needed between regions operated at different voltages.
- (4) There may be some limit on the voltage difference that can be tolerated between regions.
- (5) Protection against latch-up may be needed at the logic interfaces between regions of different voltage.
- (6) New design rules for routing may be needed to deal with signals at one voltage passing through a region at another voltage.

Some of these issues can be considered during multiple voltage scheduling. Perhaps the greatest impact will be related to grouping operations of a particular supply voltage into a common region. It may also be necessary to limit voltage differences on logic interfaces in order to avoid latch-up. Closely intermingled operations at different voltages could lead to complex routing between regions, increased need for level conversions, and increased risk of latch-up. Grouping operations logically and temporally could not only improve routing, but should also lead to fewer voltage regions on the chip, less space lost to isolation between voltage regions, less interfaces where latch-up might be a problem, and fewer signals passing between regions operating at different voltages.

Another synthesis task to be affected by multiple voltages is resource bincling, i.e., determining exactly which instance of a circuit resource will be used to implement each datapath operation. Grouping of operations into voltage regions actually constitutes a form of binding decision. Grouping decisions made without regard to binding are likely to lead to violations of resource constraints. Binding results are also needed in order to estimate the effects of scheduling decisions on switched capacitance.

6.2 Circuit Design

There are some circuit design issues that still need to be addressed by IMOVER including alternative level converter designs, multiplexer design, and control logic design.

Alternative level converter designs such as the combined register and level converter should be considered. The DCVS converter design considered in this paper doesn't exhibit static power consumption, but short circuit energy is a problem. Delays and energy also increase greatly as the input voltage to the level converter becomes small.

MOVER does not presently consider the area or delays associated with multiplexers needed to share interconnect and circuit resources. The architectural model assumed by MOVER should be extended to consider how resource sharing will be implemented. In particular, it needs to be decided where multiplexers should be

inserted and at what supply voltage. An appropriate multiplexer must be selected and characterized for delay and energy dissipation characteristics.

MOVER makes assumptions about datapath control and clocking that are convenient for scheduling and energy estimation, but will require support from the control logic. It is assumed that the entire control of the datapath is accomplished through selective clocking of registers and switching of multiplexers. This will require specially gated clocks for each register.

6.3 Power Supplies

Before implementing a multiple voltage datapath, some decisions must be made regarding the voltages that can be selected and the type of power supply to be used. Regarding voltage selection, we must decide how many supplies to use and determine whether or not non-standard voltages are acceptable. Regarding the type of power supply, we will only consider the choice between generating the voltage on-chip or off-chip. All of these choices will depend largely on the application. Possible scenarios include the following:

- (1) The datapath is used in an ASIC where heat dissipation within the chip is the over-riding concern.
- (2) The datapath is the critical element (both in terms of power and speed) in a battery powered system where it might be possible to run the other components at some reduced non-standard voltages.
- (3) The datapath is used in a battery powered system where one or more standard voltages (e.g., 5V, 3.3V, 1.5V, etc.) are required for other components in the system.

Scenario 1 is the most favorable to multiple voltages because we are willing to bear the cost of off-chip power supplies for non-standard voltages if it will cool the chip down. In this case, we must determine that the amount of heat reduction achieved is enough to merit the increased layout complexity, more supply pins on the ASIC, and non-standard power supplies. Scenario 2 may favor using a single minimum non-standard voltage. However, we would have to determine if the energy savings of two or three supplies justify increased layout complexity and the overhead of additional power supplies on or off the chip. Scenario 3 would tend to favor a multiple standard voltage, provided that we can accept the increased layout complexity. Non-standard voltages might be worth using if the energy savings substantially exceeds the energy cost of the additional power supplies.

A simple analysis provides some insight into the conditions under which a new supply voltage could be justified. In a battery powered system, we would need a DC to DC converter to obtain the new voltage. Let λ represent the efficiency of the DC to DC converter. The efficiency can be most easily described as the power output to the datapath divided by the power input to the DC-DC converter.

$$\lambda = \frac{P_{output}}{P_{input}} \quad (30)$$

This model does not explicitly represent the effect of the amount of loading or choice of voltages on converter efficiency. For now, we are only trying to determine

the degree of converter efficiency needed in order to make a new supply voltage viable. Conversely, given a DC-DC converter of known efficiency, we want to know how much voltage reduction is needed to justify use of the converter.

Let α represent the fraction of switched capacitance in the datapath that will be allocated to the new supply voltage. V_1 represents the primary supply voltage. V_2 represents the new reduced supply voltage under consideration. E_1 represents the energy dissipation of the datapath operating with the single supply voltage V_1 . The energy E_1 can be split into a portion, αE_1 , representing the circuitry that will run at voltage V_2 , and a remaining portion $(1 - \alpha) E_1$ that will continue to run at voltage V_1 .

$$E_1 = \alpha E_1 + (1 - \alpha) E_1 \quad (31)$$

When the new supply voltage V_2 is introduced, the first term in equation 31 will be scaled by the factor $\frac{V_2^2}{V_1^2}$. The new datapath energy dissipation (ignoring DC-DC conversion) becomes:

$$E_2 = \alpha \frac{V_2^2}{V_1^2} E_1 + (1 - \alpha) E_1 \quad (32)$$

We can now determine the energy savings.

$$E_{saved} = E_2 - E_1 = (1 - \frac{V_2^2}{V_1^2}) \times \alpha E_1 \quad (33)$$

However, the energy lost in the DC-DC converter equals the energy of the circuitry operating at V_2 divided by the efficiency of the converter.

$$E_{lost} = \frac{E_{output}}{\lambda} = \frac{\alpha \frac{V_2^2}{V_1^2} E_1}{\lambda} \quad (34)$$

A bit of algebraic manipulation will reveal the system energy savings (including converter losses) as a function of α , λ , V_1 , and V_2 .

$$\% \text{ Savings} = 100 \times \frac{E_{saved} - E_{lost}}{E_1} = 100 \times \alpha \times (1 - \frac{V_2^2}{\lambda V_1^2}) \quad (35)$$

Consider a simple example. Let $V_1 = 3.3V$, $V_2 = 2.1V$, and efficiency $\lambda = 0.75$. Suppose 60% of the circuit can operate at voltage V_2 . Given an ideal DC-DC converter, the energy savings would be 36%. However, when the converter efficiency is considered, the savings drops more than a half to 17%. The break-even point occurs when $\lambda = \frac{V_2^2}{V_1^2}$. For the last example, the converter efficiency has to be at least 41% to avoid losing energy. In practice, the break-even point will be somewhat higher due to logic level conversions that will be required within the datapath.

The preceding analysis suggests that a DC to DC converter doesn't have to be exceedingly efficient in order to achieve energy savings. Had the voltage reduction been merely from 3.3V to 3.0V, DC-DC converter efficiency would have to be at least

83%. Converter designs are available that easily exceed this efficiency requirement. Stratakos et al. [Stratakos et al. 1994] designed a DC-DC converter that achieves better than 90% efficiency for a 6V to 1.5V voltage reduction.

7. CONCLUSIONS

In this paper we have presented MOVER, a tool which reduces the energy dissipation of a datapath design through use of multiple supply voltages. An area estimate is produced based on the minimum number of circuit resources required to implement the design. One, two, and three supply voltage designs are generated for consideration by the circuit designer. The user has control over latency constraints, resource constraints, total number of control steps, clock period, voltage range, and number of power supplies. MOVER can be used to examine and trade-off the effects of each constraint on the energy and area estimates.

MOVER iteratively searches the voltage range for minimum voltages that will be feasible in a one, two, and three supply solution. An exact ILP formulation is used to evaluate schedule feasibility for each voltage selection. The same ILP formulation is used to determine which operations are assigned to each power supply.

MOVER was exercised for six different datapath specifications, each subjected to a variety of latency, resource, and power supply constraints for a total of 70 test cases. The test cases were modest in size, ranging from 13 to 26 datapath operations and 2 to 24 control steps. 40% of test cases completed in less than one minute; 93% in less than 10 minutes. The results indicate that some but not all datapath specifications can benefit significantly from use of multiple voltages. In many cases, energy was reduced substantially going from one to two supply voltages. Improvements as much as 50% were observed, but 20-30% savings were more typical. Adding a third supply produced relatively little improvement over two supplies, 15% improvement at most. Results from MOVER are comparable and in many cases better than results obtained using the MESVS (Minimum Energy Scheduling with Voltage Selection) ILP formulation presented in [Johnson and Roy 1996]. Behavior with respect to latency, resource, and supply voltage constraints is similar between MOVER and MESVS. The improvement relative to a pure ILP formulation is due to the fact that ILP formulation could only select from a discrete set of voltages, whereas MOVER can select from a continuous range of voltages.

Several opportunities exist to help MOVER address a broader range of datapath design problems. One area for development is to integrate resource binding into the scheduling process. The bindings can have a significant effect on switched capacitances, layout, and routing. Furthermore, multiple voltage requirements will place new constraints on the binding process, especially if circuit resources at a particular voltage are clustered together. The delay models also need to reflect the effects of multiple voltage binding and IC layout. Finally, the architectural model used by MOVER should be extended to account for multiplexing of signals and support conditional execution, functional pipelining, and chaining.

Appendix: MESVS ILP Formulation

The MESVS (Minimum Energy Scheduling with Voltage Selection) formulation [Johnson and Roy 1996] is an ILP formulation that solves nearly the same problem as MOVER. The only difference between the problem definitions is that MESVS

selects supply voltages from a user specified discrete set, whereas MOVER selects voltages from a continuous range of values. The big difference between MESVS and MOVER is in the implementation. MESVS defines a single ILP problem to simultaneously solve the scheduling, voltage selection, level conversion, voltage assignment, and resource allocation problems. The MESVS formulation is useful for seeing what can be achieved with multiple voltages. It could also be useful for some design problems of moderate size (up to 20 or 30 operations), provided that the designer does not mind running MESVS on a general purpose ILP solver several times while adjusting problem constraints and ILP solver controls to obtain a solution. MESVS results were used as benchmarks against which MOVER results were compared. MOVER results were consistently as good or better than MESVS results and were obtained orders of magnitude more quickly with very little manual intervention. The MESVS formulation is present here for reference.

The MESVS formulation is a zero-one integer linear program (ILP) that adapts and extends data path scheduling formulations described by DeMicheli [DeMicheli 1994] and Gebotys [Gebotys 1995b; Gebotys and Elmasry 1993]. Inputs, outputs, and architectural assumptions are all nearly identical between MESVS and MOVER, so we will not repeat them here. MESVS decision variables, constraint inequalities, objective functions, and solution strategies will be presented in the remainder of this appendix.

Decision variables

Decision variables are defined for five types of design parameters: operation start time and supply voltage ($x_{i,l,s}$), operation completion time and supply voltage ($z_{i,l,s}$), supply voltage availability ($usel_s$), insertion of level conversions (vi_{i,j,s_1,s_2}), and allocation of resources to each available supply voltage ($aq_{m,s}$). $x_{i,l,s} = 1$ indicates that operation i is scheduled to start on clock cycle l and use supply voltage s . $z_{i,l,s} = 1$ indicates that operation i is scheduled to complete by clock cycle l and uses supply voltage s . $usel_s = 1$ indicates that supply voltage s is available for use by the data path. $vi_{i,j,s_1,s_2} = 1$ for $voltage(s_1) < voltage(s_2)$ indicates that a level converter is required in the signal path from operation i using voltage s_1 to operation j using voltage s_2 . $vi_{i,j,s_0,s_0} = 1$ is used to indicate that no level conversion is required on the path from operation i to j . s_0 is arbitrarily chosen to be the index of the lowest supply voltage. $aq_{m,s}$ indicates the number of resources of type m (e.g., adder, multiplier, etc.) that are allocated to supply voltage s .

Constraints

There can only be one assignment of a start time, completion time, and supply voltage to each operation. These restrictions are enforced by equations 36 and 37. Equation 38 guarantees that the supply voltages indicated by $x_{i,l,s}$ and $z_{i,l,s}$ are consistent. S is the set of possible supply voltages.

$$\sum_l \sum_s x_{i,l,s} = 1 \quad \forall i \in V \quad (36)$$

$$\sum \sum z_{i,l,s} = 1 \quad \forall i \in V \quad (37)$$

$$\forall i \in V, \forall s \in S \quad (38)$$

If there is a data flow from operator i to j , operator i uses voltage supply s_1 , operator j uses supply s_2 , and $voltage(s_1) < voltage(s_2)$, then $vij(i, j, s_1, s_2)$ is forced to a value of 1. E_{conv} indicates the set of arcs that correspond to signal paths.

$$vij_{i,j,s_1,s_2} \geq \sum_l (x_{i,l,s_1} + x_{j,l,s_2}) - 1 \quad (39)$$

$$\forall (i, j) \in E_{conv}, \quad voltage(s_1) < voltage(s_2)$$

For each data flow (i, j) , only one level conversion can be specified. Equation 40 requires that there be one and only one choice of s_1 and s_2 for which $vij_{i,j,s_1,s_2} = 1$. Equation 41 allows $vij_{i,j,s_0,s_0} = 1$ so that there is a way to account for signal arcs that do not use a level conversion.

$$\sum_{s_1} \sum_{s_2} vij_{i,j,s_1,s_2} = 1 \quad \forall (i, j) \in E_{conv} \quad (40)$$

$$vij_{i,j,s_0,s_0} \leq 1 \quad \forall (i, j) \in E_{conv} \quad (41)$$

If operator j is a transitive NO-OP, force the supply voltage for operator j to match the supply voltage for operator i . S is the set of user specified permissible supply voltages. E_{trans} is the set of arcs ending at a transitive NO-OP.

$$\sum_l x_{i,l,s} = \sum_l x_{j,l,s} \quad (42)$$

$$\forall (i, j) \in E_{trans}, \forall s \in S$$

Equation 43 restricts the number of supply voltages actually used to a specified number. Equation 44 can be used to keep the ILP solution from selecting more than one supply voltage in any range of $vspace$ volts.

$$\sum_s vsel_s = \text{number of supplies allowed} \quad (43)$$

$$\sum_{v(s) \leq v(s_1) \leq v(s) + vspace} vsel_{s_1} \leq 1 \quad \forall s \in S \quad (44)$$

Five similar inequalities are used to enforce precedence relationships and latency constraints among the start and completion time variables for each operator. All are derived from the structured precedence constraint shown by Gebotys [Gebotys 1992] to be facets of the scheduling polytope. The first inequality 45 requires the

start time of a null operation to not exceed the completion time. The inequality 46 requires the completion time of a non-null operation to exceed the start time by del_{i,j,s_1,s_2} . del_{i,j,s_1,s_2} is the sum of the register propagation and level conversion delay from operation i to j and the propagation delay of operation j , given that i uses voltage s_1 and j uses voltage s_2 . Inequality 47 enforces any minimum latency constraints, $lat(i,j) > 0$. Inequality 48 enforces maximum latency constraints from operation j to i in the event that $lat(i,j) < 0$. Inequality 49 requires that for each data flow (i,j) , the completion time of operation i must not exceed the start time of operation j . L is the set of available clock steps.

$$\sum_s \left(\sum_{lx < l} z_{i,lx,s} + \sum_{lx \geq l} x_{i,lx,s} \right) \leq 1 \quad (45)$$

$$\forall i \in V_{NOOP}, \forall l \in L, \forall s \in S$$

$$\sum_{lx < l + del_{i,j,s_1,s_2}} z_{j,lx,s_2} + \sum_{lx \geq l} x_{j,lx,s_2} + \sum_{lx} x_{i,lx,s_1} \leq 2 \quad (46)$$

$$\forall i \in V, j \in V_{oper}, s_1, s_2 \in S, \forall l \in L$$

$$\sum_s \left(\sum_{lx < l + lat_{i,j}} z_{j,lx,s_2} + \sum_{lx \geq l} x_{j,lx,s_2} \right) \leq 1 \quad (47)$$

$$\forall lat(i,j) > 0, \forall l \in L$$

$$\sum_s \left(x_{j,l,s} + \sum_{lx > l - lat(i,j)} x_{i,lx,s} \right) \leq 1 \quad (48)$$

$$\forall lat(i,j) < 0, \forall l \in L$$

$$\sum_s \left(\sum_{lx < l} x_{j,lx,s} + \sum_{lx \geq l} z_{i,lx,s} \right) \leq 1 \quad (49)$$

$$\forall lat(i,j) \geq 0, \forall l \in L$$

Three inequalities are used to enforce resource and voltage supply allocation constraints. Equation 50 requires the number data path resources $aq_{m,s}$ of type m allocated to each supply voltage s , to not add up to more than the resource constraint for each resource type m . Equation 51 specifies that resources can only be allocated for a supply voltage, s , that has been selected by variable $vsel_s$. In-

equality 52 states that the number of operations of type m ($i \in V_m$) using supply voltage s that are active during clock cycle l can not exceed the number of type m resources allocated to supply voltage s .

$$\sum_s aq_{m,s} = maxres(m) \quad (50)$$

$$\forall m \in M_{oper}$$

$$aq_{m,s} \leq vsel_s \times maxres(m) \quad (51)$$

$$\forall m \in M_{oper}, \forall s \in S$$

$$aq_{m,s} \geq \sum_{lx \leq l} \sum_{i \in V_m} (x_{i,lx,s} - z_{i,lx,s}) \quad (52)$$

$$\forall l \in L, \forall m \in M_{oper}, \forall s \in S$$

Objective function

An estimate of energy dissipation serves as the objective function to be minimized when scheduling and assigning supply voltages to resources in the data path. The estimate is obtained by first taking the average total energy dissipated to process one input sample, i.e., one execution of the data path. The parameter arrays $onrgy(i, s)$ and $rnrngy(i, s)$ contain estimates of the energy expended to perform operation i and store the result for a single change of input values at voltage s . $cnrgymult(i, j) \times cnrgy(s_1, s_2)$ gives the energy dissipation of the level conversion from voltage s_1 to s_2 applied to a single change in the output of operation i destined for operation j . The parameter arrays give energy estimates for each possible choice of supply voltages. The voltage assignments indicated by $x_{i,l,s}$ and v_{ij,i,j,s_1,s_2} are used to select one energy estimate from the parameter arrays for each operator, register, and level converter.

$$\text{energy} = \quad (53)$$

$$\begin{aligned} & \sum_i \sum_l \sum_s (x_{i,l,s} \times (onrgy(i, s) + rnrngy(i, s))) \\ & + \sum_{i,j \in E} \sum_{s_1} \sum_{s_2} (cnrgymult(i, j) \times \\ & \quad cnrgy(s_1, s_2) \times v_{ij,i,j,s_1,s_2}) \end{aligned}$$

Solution strategy

The ILP formulation was implemented using GAMS (General Algebraic Modeling System) and solved using the CPLEX linear and integer program solver. The solution strategy taken was to start with a formulation that is relatively easy to solve

and then solve successively more difficult problems using the previous results to set bounds and initial conditions. First, lower bound schedule times are determined based on resource constraints [Chaudhuri et al. 1994]. An ASAP (As Soon As Possible) schedule is generated to update the lower bounds. An ALAP (As Late As Possible) schedule is run to obtain upper bounds on schedule times. The upper bounds are improved by taking into account resource constraints. A single voltage minimum energy schedule is generated, given the ASAP schedule as a starting point and a 5V energy estimate as an upper bound on the objective. A dual voltage schedule is then generated using the single voltage solution as a starting point and upper bound. A triple voltage schedule is generated using the dual voltage solution as starting point and upper bound.

References

- CHANDRAKASAN, A. P. ET AL. 1995. Optimizing power using transformations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 14, 1 (January), 12–31.
- CHANDRAKASAN, A. P., ALLMON, R., STRATAKOS, A., AND BRODERSEN, R. W. 1994. Design of portable systems. In *IEEE Custom Integrated Circuits Conference* (1994). pp 259–266.
- CHAUDHURI, S., WALKER, R. A., AND MITCHELL, J. E. 1994. Analyzing and exploiting the structure of the constraints in the ILP approach to the scheduling problem. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 2, 4 (December), 456–471.
- DEMICHELI, G. 1994. *Synthesis and Optimization of Digital Circuits*. McGraw-Hill, Inc.
- GEBOTYS, C. H. 1992. *Optimal VLSI Architectural Synthesis: Area, Performance, and Testability*. Kluwer Academic Publishers, Boston, MA.
- GEBOTYS, C. H. 1995a. An ILP model for simultaneous scheduling and partitioning for low power system mapping. Technical report (April), University of Waterloo, Department of Electrical and Computer Engineering, VLSI Group.
- GEBOTYS, C. H. 1995b. An optimal methodology for synthesis of DSP multichip architectures. *Journal of VLSI Signal Processing* 11, 1-2 (Oct.-Nov.), 9–19.
- GEBOTYS, C. H. AND ELMASRY, M. I. 1993. A global optimization approach for architectural synthesis. *IEEE Transactions on CAD/ICAS* 12, 9 (Sep.), 1266–1278.
- GOODBY, L., ORAILOGLU, A., AND CHAU, P. M. 1994. Microarchitectural synthesis of performance-constrained, low-power vlsi designs. In *Proceedings - IEEE International Conference on Computer Design: VLSI in Computers and Processors* (1994). pp. 323–326.
- JOHNSON, M. C. AND ROY, K. 1996. Optimal selection of supply voltages and level conversions during data path scheduling under resource constraints. In *Proceedings, International Conference on Computer Design* (1996). To be presented at ICCD, Oct. 1996, Austin TX.
- KUMAR, N., KATKOORI, S., RADER, L., AND VEMURI, R. 1995. Profile-driven behavioral synthesis for low-power VLSI systems. *IEEE Design & Test of Computers* 12, 3 (Fall), 70–84.
- RABAEY, J. 1996. *Digital integrated circuits : a design perspective*. Prentice Hall, Englewood Cliffs, NJ.
- RAGHUNATHAN, A. AND JHA, N. K. 1994. Behavioral synthesis for low power. In *Proceedings - IEEE International Conference on Computer Design: VLSI in Computers and Processors* (1994). pp. 318–322.
- RAGHUNATHAN, A. AND JHA, N. K. 1995. An iterative improvement algorithm for low power data path synthesis. In *Proceedings of the International Conference on Computer Aided Design* (1995). pp. 597–602.
- RAJE, S. AND SARRAFZADEH, M. 1995. Variable voltage scheduling. In *Proceedings of the International Symposium on Low Power Design* (1995). pp. 9–14.
- RAO, D. S. 1992. The fifth order elliptic wave filter benchmark. Benchmarkset: HLSynth92, <http://www.cbl.ncsu.edu/www/CBLDocs/Bench.htm>.

- SANMARTIN, R. AND KNIGHT, J. P. 1995. **Power-profiler: Optimizing ASICs power consumption at the behavioral level.** In *Proceedings 32nd Design Automation Conference (1995)*. pp. 42-47.
- STRATAKOS, A. J. ET AL. 1994. **High-efficiency low-voltage DC-DC conversion for portable applications.** In *Proceedings, International Workshop on Low Power Design (1994)*.
- USAMI, K. AND HOROWITZ, M. 1995. **Clustered voltage scaling technique for low-power design.** In *Proceedings of the International Symposium on Low Power Design (1995)*. pp. 3-8.