3-1-1996

Optimal Selection of Supply Voltages and Level Conversions During Low Power Data Path Scheduling

Mark C. Johnson
Purdue University School of Electrical and Computer Engineering

Kaushik Roy
Purdue University School of Electrical and Computer Engineering

Optimal Selection of Supply Voltages and Level Conversions During Low Power Data Path Scheduling

Mark C. Johnson
Kaushik Roy

TR-ECE 96-3
March 1996

School of Electrical
and Computer Engineering
Purdue University
West Lafayette, Indiana 47907-1285

Optimal Selection of Supply Voltages and Level Conversions During Low Power Data Path Scheduling *

Mark C. Johnson and Kaushik Roy

Electrical Engineering

Purdue University

West Lafayette

IN 47907-1285, USA.

Ph: 317-494-2361

Fax: 317-494-3371

e-mail: mcjohnso@ecn.purdue.edu, kaushik@ecn.purdue.edu

Abstract

In this paper we will consider how to select an optimal set of supply voltages and account for level conversion costs when optimizing the schedule of a resource dominated data path for minimum average power dissipation. Integer linear program (ILP) and non-linear program (NLP) formulations are presented for a minimum power schedule under latency and throughput constraints. Results are presented for several data path topologies under minimum latency constraints and under more relaxed latency constraints. The optimization demonstrated substantial benefit going from one to two supply voltages, but minimal additional benefit from any additional supplies. For example, a Kalman filter benchmark produced a power estimate of 356.7mW for a single 5V supply, 265.4mW for 4V and 5V supplies, but no additional improvement for three supplies. Increasing minimum schedule latency by 50% improved optimization results substantially for two and three supply voltages but in most cases there was no improvement at all for a single optimal supply voltage.

*This research was supported in part by ARPA (under contract F33615-95-C-1625)

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 workstirtions, 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 and discharging of circuit capacitances during switching. For static CMOS, switching power is proportional to V_{dd}^2 [15]. 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 proportional to $\frac{V_{dd}}{(V_{dd}-V_T)^2}$ [15], 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. Chandrakasan et. al. describe: these techniques in [15].

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 Chandrakasan et. al. applied to this approach.

A number of researchers have developed systems or proposed methods that incorporate architecture driven voltage scaling [4, 6, 7, 11, 5, 8, 9]. The HYPER-LP system [4] 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 of the systems [4, 6, 7, 11, 5] try to also minimize switched capacitance in the data path. Raje and Sarrafzadeh [9] take switching activities as given. They schedule the data path and assign voltages to data path operators so as to minimize power given a predetermined set of supply voltages.

The objective of this research has been to incorporate multiple supply voltage selection and level conversion costs into the low power optimization of resource dominated data paths. In this paper, ILP and NLP formulations are presented that generate a schedule with supply voltages assigned to each operation so as to minimize average power dissipation. These formulations are designed for resource dominated data paths for which area, performance, and power dissipation are dominated by the data path resources (arithmetic operators and registers). For the remainder of this paper, we will refer to our ILP formulation as MPSVS (Minirnum Power Schedule with Voltage Selection). MPSVS is closest in scope to the work

of Raje and Sarrafzadeh [9]. However, MPSVS is distinguished by the fact that it selects an optimal set of one, two, or three supply voltages from a larger set of possible supply voltages, and factors level conversion effects into the delay constraints and power estimate. The NLP formulation generates a schedule with continuous valued schedule times and unlimited supply voltages. The NLP solutions are included for comparison to ILP results.

2 ILP Model for Minimum Power Schedule With Voltage Selection

The MPSVS formulation describes a minimum power scheduling problem under latency and multiple supply voltage constraints. It is a zero-one integer linear program ILP similar in structure to data path scheduling formulations described by DeMicheli [14] and Gebotys [10]. The primary input to MPSVS is a data flow graph that specifies the operations, data flows, and latency constraints for a data path. Other inputs to MPSVS include: specification of a discrete set of permitted supply voltages, a limit on the number of supply voltages that can be selected, a minimum difference between voltages that can be selected, average switching activities for each data path operation, and nominal propagation delay and average energy dissipation values for each data path resource. Solution of the minimum power scheduling problem results in a data path schedule, selection of an optimal set of supply voltages, and assignment of a supply voltage to each operation.

MPSVS makes the following assumptions: a one-to-one relationship of operator types to module types, unlimited resources, and a predetermined clock period. Furthermore, the outputs of all operations are registered for an entire sample interval of the data path. Level converters, when needed, are always located at the inputs to an operator.

Delay and power dissipation are accounted for arithmetic operations, registers, and logic level conversions. Worst case propagation delay and average energy dissipation per input switching event were measured for each type of operator and register under nominal operating conditions. MPSVS re-scales the delay and energy estimates to be consistent with the supply voltages, switching activities, and estimated load capacitances in the data path. Energy estimates for operators and registers are scaled as $E = E_0 \times \frac{V^2}{V_0^2}$, where V_0 is the nominal supply voltage, V is the actual voltage, and E_0 is the nominal energy. Delay estimates for operators and registers scale as $t_p = t_{p0} \times \frac{V}{V_0} \times \frac{(V_0 - |V_T|)^2}{(V - |V_T|)^2}$, where t_{p0} is the nominal propagation delay and V_T is the MOS transistor threshold voltage. Delay and energy estimates for level converters are treated in a similar manner except that they are functions of two supply voltages rather than one. Details of the level conversion models are given in section 4.1. In all cases delay is scaled linearly with respect to load capacitance. Energy is scaled linearly with respect to both load capacitance and switching activity.

The objective function for MPSVS is an estimate of data path power dissipation as a function of supply voltages and the rate at which data samples are processed. The average energy of each data path operation, register operation, and level conversion is determined based on the .voltageassignments. The sum of these energies over the entire data path represents the average energy dissipated by a single execution of the data path. The total energy is divided by the time interval between data samples to calculate average power.

2.1 Definitions

Before presenting the ILP formulation, it is important to describe the manner in which a data path is specified for optimization and define the notations that will be used. Notations to be defined include set names, set indices, and parameters that characterize data path resources.

The: input to both the ILP and NLP formulations is 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 [14] except that hierarchical and conditional graph entities are not supported. Following is a brief description of the DFG definition.

The DFG is a directed acyclic graph, $G(V, \mathrm{E})$ 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: 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 value for the destination start time minus the source start time. The magnitude of a negative value specifies a maximum value for the source start time minus the destination start time.

Two types of NO-OP's are used which we will refer to as "transitive" and "non-transitive" NO-01''s. 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 logic levels are propagated through the NO-OP. Non-transitive NO-01''s are ignored in the accounting of register delays, level conversions, and voltage supply choice:.

Table 1 defines the sets and indices used in the ILP formulation for MPSVS. The "Set Name" column lists the labels used to represent the entire membership of a set. "Index" and "Index Aliases" identify the index variables used to represent individual members of the corresponding set.

Table 2 describes constants and model parameters used by the ILP formulation. Indices, if any, for each parameter array will be shown enclosed in parenthesis following the parameter name.

2.2 Decision Variables

Every possible pair of i and l values define a possible assignment of start time to an operator by means of the $x_{i,l}$ variables. The results of an "as soon as possible" (ASAP) and an "as late as possible" (ALAP) schedule are used to put bounds on the range of start times allowed for each operator [2].

$$x_{i,l} = \begin{cases} 1 & \text{if operation } i \text{ is scheduled to start at cycle } l \\ 0 & \text{otherwise} \end{cases}$$

Every possible pair of i and l values also define a possible assignment of an execution time to an operator by means of the $cyc_{i,l}$ variables. The results of an ASAP and an ALAP schedule

Table 1: Set Names and Indices for ILP Formulation

Set Name	Index	Index Aliases	Description
V	i	j	Set of vertices in the DFG, each corresponding to an operator or NO-OP.
V_{anchor}	i	j	Set of vertices in the DFG for which the start time will be anchored to the ASAP schedule.
E	(i,j)		Set of edges in the DFG, each corresponding to a data or timing dependency from operator i to j.
E_{conv}	(i,j)		Set of DFG edges representing data flows from operator i to j that could require level conversion
E_{oper}	(i,j)		Set of DFG edges that do not have a NO-OP as the destination vertex.
E_{reg}	(i,j)		Set of data flows that include a register delay.
E_{tran}	(i,j)		Set of DFG edges that have a transitive NO-OP at the destination vertex.
L	l		Set of clock cycles available for scheduling of operations.
S	s	$s1, s2$	Set of possible supply voltages available for selection.
M	m		Set of operator types: MI = transitive NO-OP MO = non-transitive NO-OP M1 = adder M2 = multiplier
T	(i,m)		Mapping of operators to operator types.

Table 2: Parameters Used in ILP Formulation

Parameter Name	Description
minv	Minimum supply voltage [V]
$maxv$	Maximum supply voltage [V]
pathwidth	Width (in bits) of all data flows
$activity(i, j)$	Average switching activity on each signal in the entire data path. Ranges from 0 to 1
t_{clk}	Clock period [ns].
t_{samp}	Data introduction interval [ns].
$vspace$	Minimum voltage difference between voltage supplies that are made available [V].
$cdel(s1, s2)$	Level converter delay for each possible pairing of supply voltages [ns]
$cnrgy(s1, s2)$	Average converter energy [pJ] dissipated when inputs-to the converter switch, calculated for each possible pairing of supply voltages
$cdelmult(i, j)$	Factor by which to multiply converter delay in a data flow. Set to zero for arcs that can not have a level conversion. Otherwise, this parameter is a scale factor to adjust for input capacitance of operator j
$cnrgymult(i, j)$	Identical to $cnrgymult(i, j)$ except that non-zero values are scale factors to adjust the converter energy estimate.
$fanout(i)$	Number of operator inputs driven by the output of operator i
$lat(i, j)$	Latency constraint on arc (i, j). A positive value indicates a minimum delay from i to j. The magnitude of a negative value specifies a maximum delay from j to i. [clock cycles]
$odel(i, s)$	Propagation delay of operator i when using supply voltage s [ns]
$onrgy(i, s)$	Average energy dissipated for one execution of operator i when using supply voltage s [pJ]
$rdel(i, s)$	Propagation delay of register at output of operator i when using supply voltage s [ns]
$rnrgy(i, s)$	Average energy [pJ] dissipated when a new value is latched by the register at the output of operator i when using supply voltage s
$voltage(s)$	Voltage level [V] of supply s. These levels are determined so as to be uniformly distributed from minv to $maxv$.

are used to put bounds on the range of execution times to be considered for each operator.

$$cyc_{i,l} = \begin{cases} 1 & \text{if operation } i \text{ is allowed } l \text{ cycles to execute} \\ 0 & \text{otherwise} \end{cases}$$

Every possible pair of i and s values define a possible assignment of supply voltage to an operator by means of the $v_{i,s}$ variables. This voltage assignment also specifies the level of a logic one output from the operator.

$$v_{i,s} = \begin{cases} 1 & \text{if operation } i \text{ is powered by voltage supply } s \\ 0 & \text{otherwise} \end{cases}$$

The $vsel_s$ variables determine which of the supply voltages (and logic levels) in S will be allowed to be used.

$$vsel_s = \begin{cases} 1 & \text{if supply voltage is is available for use} \\ 0 & \text{otherwise} \end{cases}$$

The vij_{i,j,s_1,s_2} variables account for all of the possible logic level conversions required in a data path. vij_{i,j,s_1,s_2} is set to one when there is a data flow from operation i to j, supply voltage s_1 is assigned to operation i, supply voltage s_2 is assigned to operation j, and $voltage(s_1) < voltage(s_2)$. vij_{i,j,s_0,s_0} is set to one if there is a data flow from i to j for which a level conversion is not required. s_0 represents the index for the lowest defined supply voltage, but (so, so) was arbitrarily selected to represent all cases where a level converter is not required.

$$vij_{i,j,s_1,s_2} = \begin{cases} 1 & \text{if (i, j)} \in E_{conv}\text{, operator } i \text{ uses supply} \\ & s_1\text{, operator j uses supply } s_2\text{, and} \\ & voltage(s_2) > voltage(s_1) \\ 1 & \text{if } (i,j) \in E_{conv}\text{,} \\ & voltage(s_1) = voltage(s_2) = minv \\ & \text{and the supply voltage for operator i equals or} \\ & \text{exceeds that of operator } j \\ 0 & \text{otherwise} \end{cases}$$

2.3 Constraints

There can only be one start time, one execution time, and one supply voltage assigned to each data path operation. These restrictions are enforced by constraint equations 1, 2, and 3 respectively.

$$\sum x_{i,l} = 1 \qquad \forall i \tag{1}$$

$$\sum_l cyc_{i,l} = 1 \qquad \forall i \tag{2}$$

$$\sum_s v_{i,s} = 1 \qquad \forall i \tag{3}$$

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.

$$vij_{i,j,s_1,s_2} \geq v_{i,s_1} + v_{j,s_2} - 1 \qquad \forall (i,j) \in E_{conv},\ voltage(s_1) < voltage(s_2) \tag{4}$$

For each data flow (i,j), only one kind of level conversion can be specified. There must be one and only one choice of s_1 and s_2 for which vij_{i,j,s_1,s_2} will equal 1.

$$\sum_{s_1}\sum_{s_2} vij_{i,j,s_1,s_2} = 1 \qquad \forall (i,j) \in E_{conv} \tag{5}$$

vij_{i,j,s_0,s_0} will equal 1 if no level conversion is used in the data flow from operator i to j. This is necessary in order to satisfy equation 5 which requires that exactly one level conversion is always specified for each data flow. s_0 is the index of the minimum supply voltage, but (s_0, s_0) is used here as a way to indicate that no level conversion is required..

$$vij_{i,j,s_0,s_0} \leq 1 \qquad \forall (i,j) \in E_{conv} \tag{6}$$

If operator j is a transitive NO-OP, force the supply voltage for operator j to match the supply voltage for operator i.

$$v_{j,s} = v_{i,s} \qquad \forall (i,j) \in E_{trans} \tag{7}$$

Restrict the number of supply voltages actually used to a specified number.

$$\sum_{s} vsel_s = \text{number of supplies allowed} \tag{8}$$

A voltage supply can only be assigned to operator i if that supply is available as indicated by $vsel_s$.

$$v_{i,s} \leq vsel_s \qquad \forall i \text{ and } s \tag{9}$$

Equation 10 guarantees that not more than one supply voltage will be selected in an interval of $vspace$ volts.

$$\sum_{s \leq s_1 \leq s+vspace} vsel_{s_1} \leq 1 \qquad Vs \tag{10}$$

For each data flow from operator i to j, the execution time allocated to operator j must meet or exceed the sum of the propagation delay of operator j, the register at the output of operator i, and the level conversion (if any). Equation 11 represents this constraint as follows:

$$\sum_{l} l \times cyc_{j,l} \geq \tag{11}$$

$$\left(\sum_{s} v_{j,s} \times odel(j,s) + \sum_{s} v_{i,s} \times rdel(i,s) + \right.$$

$$\left. \sum_{s_1}\sum_{s_2} cdelmult(i,j) \times cdel(s1,s2) \times v_{i,j,s_1,s_2} \right) \times \frac{1}{T_{clk}}$$

$$\forall (i,j) \in E_{oper}$$

For every forward arc in the DFG, equation 12 ensures that the start time of operator j must exceed the start time of operator i by at least the execution time assigned to operator i. This guarantees that data flow dependencies in the data path are satisfied.

$$\sum_l l \times x_{j,l} - \sum_l l \times x_{i,l} - \sum_l l \times cyc_{i,l} \geq 0 \tag{12}$$

$$\forall (i,j) \in E \text{ where } lat_{i,j} \geq 0$$

For every arc with a non-zero latency constraint specified by parameter $lat(i,j)$, the start time of operator j must exceed the start time of operator j by the amount $lat(i,j)$. If $LAT(i,j) < 0$, equation 13 has the effect of enforcing a maximum latency constraint of $|lat(i,j)|$ clock cycles from the start time of operator j to i.

$$\sum_l l \times x_{j,l} - \sum_l l \times x_{i,l} - lat_{i,j} \geq 0 \tag{13}$$

$$\forall (i,j) \in E \text{ where } lat_{i,j} \neq 0$$

2.4 Objective Function

An estimateof power 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 $rnrgy(i,s)$ contain estimates of the energy expended to perform operation i and store the result for a single change of input values. $cnrgymult(i,j) \times cnrgy(s_1,s_2)$ gives the energy dissipation of the level conversion 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 assignment variables $v_{i,s}$ and vij_{i,j,s_1,s_2} are used to select one energy estimate from the parameter arrays for each aperator, register, and level converter. Finally, the total energy is divided by the data introduction interval T_{SAMP} to give an estimate of average power dissipation.

$$pwr = \frac{1}{t_{samp}} \times \sum_i \sum_s v_{i,s} \times (onrgy(i,s) + rnrgy(i,s)) + \tag{14}$$

$$\frac{1}{t_{samp}} \times \sum_{i,j \in E} \sum_{s_1} \sum_{s_2} cnrgymult(i,j) \times cnrgy(s_1,s_2) \times v_{i,j,s_1,s_2}$$

A different objective function is needed for "as soon as possible" (ASAP) and "as late as possible" (ALAP) schedule formulations that are used to set bounds on operator start times and execution times. The ASAP and ALAP objective function (equation 15) is simply the sum of the start times for all vertices in the DFG. The ASAP schedule minimizes the objective while the ALAP schedule maximizes the objective.

$$z = \sum_i \sum_l l \times x_{i,l} \tag{15}$$

2.5 Solution Strategy

The ILP formulation was implemented using GAMS [13] (General Algebraic Modeling System) and solved using the CPLEX 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 problenns using the previous results to set bounds and initial conditions. Here is the sequence of modeling and optimization phases used to finally obtain a minimum power schedule.

1. Specify the DFG and timing constraints for the data path to be optimized.
2. Obtain the ASAP schedule.
3. Obtain the ALAP schedule. The ASAP results provide the initial conditions. Start times of source nodes in the DFG are anchored to the ASAP values before running the ALAP schedule.
4. Use the ASAP and ALAP results to set bounds on the start times and execution times of each operator.
5. Obtain a minimum power schedule where the number of voltage supplies is limited to one. The ASAP results provide a starting condition and an upper bound on the power objective.
6. Obtain a minimum power schedule where the number of voltage supplies is limited to two. The single supply voltage solution provides the starting state and an upper bound on the power objective.
7. Obtain a minimum power schedule for three voltage supplies using the two supply solution for the starting state and for the upper bound on the power objective.

NLP Formulation

The NLP formulation is a continuous variable realization of the constraints and objective function that have been described for the ILP formulation. The NLP formulation should produce an optimized power dissipation lower than or equal to the ILP formulation since the ILP solution has to be a feasible solution to the continuous variable problem.

The DFG used to specify a data path is identical for the ILP and NLP formulations. Assumptions regarding the structure of the data path to be scheduled are also the same. The specifications of variables and some constraints are different in the NLP formulation. Each quantiity to be determined by the optimization is represented by a single continuous valued variable. For example, the start time of operation i is represented by a single continuous valued variable rather than a collection of zero/one variables. The execution times of operators, excluding NO-OPs, are constrained to be a least one clock cycle in duration. No restrictions are applied to the number of different supply voltages that can be selected. The supply voltages

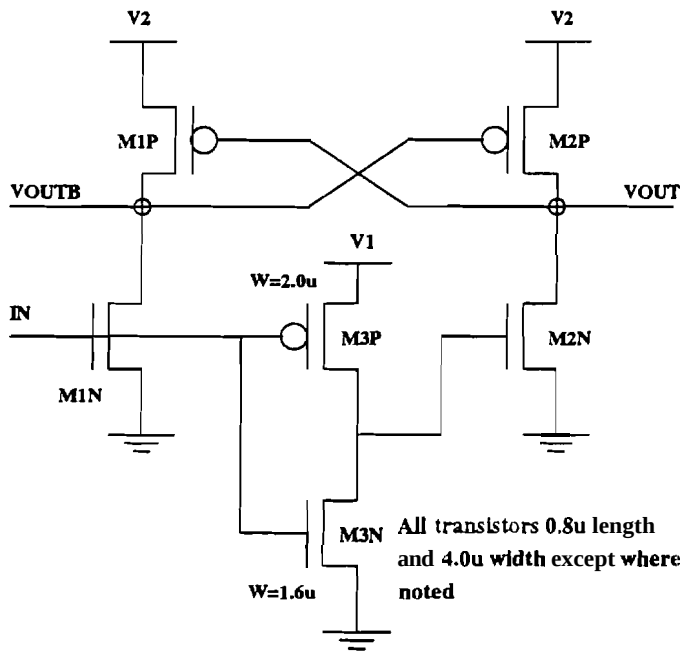

Figure 1: DCVS Logic Level Converter

can be selected from a continuum of values between a user specified minimum and maximum. Quantities such as the propagation delay of each operator are calculated dynamically by the NLP solver rather the being taken from look-up tables.

The: NLP formulation was using the GAMS input language and solved using CONOPT.

4 Level Conversion

Four alternatives were considered for implementing logic level conversions in CMOS:

1. Omit the logic level converter
2. Use a chain of inverters
3. Use an active or passive pullup
4. DCVS level converter

We omit the level converter for step-down conversions and use the DCVS circuit for step-up conversions. The DCVS circuit, shown in figure 1 was derived from a chip to chip level converter described by Chandrakasan [1]. More recently, Usami and Horowitz discussed the use of this circuit as a level converter in [3]. As long as the transistors are sized appropriately for the level conversion to be performed, 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.

Another option is to combine the register and level converters together. This approach was documented by Usami and Horowitz [3]. The combined register and level conversion was found to dissipate only 10% more power than the register alone.

4.1 Converter Modeling Approach

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 [19] 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 that closely match curves produced in HSPICE. The resulting model is valid for V_1 ranging from $1.5V$ to 5V and $V_1 + V_T \leq V_2 \leq 5V$. This correspontls to the range of supply. voltages for which a level converter is needed.

4.2 Power Model

The power model is separated into three factors. The first factor calculates the power consumption for $V_1 = \&$. 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.

$$DCVSPWR(V_2, V_2) = (a3 \times {V_2}^3 + a2 \times {V_2}^2 + a1 \times V_2 + a0) \quad (16)$$

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.

$$DCVSPWR(V_1, V_2) = DCVSPWR(V2, V2) \times (b0 + b1 \times \frac{V_2 - V_T}{(V_1 - V_T)^2}) + b2 \times {V_1}^2 \quad (17)$$

$b0$ 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 ($pwr0$) at the nominal load. ***dpdc*** is the slope of power versus capacitance for the values of V_1 and V_2 for which $pwr0$ was measured. The following expression models this dependence on load capacitance.

$$\mathrm{DCVSP}WR(V_1, V_2, C_L) = DCVSPWR(V_1, V_2) \times (1 + dpdc \times \frac{(C_L - C_{LO})}{pwr0}) \quad (18)$$

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 if extended. To take advantage of this behavior, a polynomial curve fit to $1 \div$ 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.

$$DCVSDEL(V_1, V_1 + V_T) = \frac{1}{d3 \times V_1^3 + d2 \times V_1^2 + d1 \times V_1 + d0} \tag{19}$$

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.

$$DCVSDEL(V_1, V_2) = \frac{DCVSDEL(V_1, V_1 + V_T) - del0}{V_1 + V_T - V_0} \times (V_2 - V_0) + del0 \tag{20}$$

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

5 Results

5.1 Data Path Examples

The ILP and NLP scheduling formulations were run for four example data paths: two toy benchmarks based on a four point FFT (FFT4a and FFT4b) [17], the 5th order elliptic wave filter benchmark (ELLIP) [16], a 6th order Auto-Regressive Lattice filter (LATTICE) based on the topology documented in [17], and the Kalman filter benchmark (KALMAN) [18]. Data flow graphs for each example are shown in figures 3 through 7. Figure 2 defines the notations used in each DFG. In the FFT data path, complex signal paths are split into real and imaginary data flows. The FFT4a example uses a seperate adder to implement each 2's complement inversion. FFT4b lumps any 2's complement operations into the next adder input. 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%, ie., the probability of a transition on any selected 1 bit signal is 50% in any one sample interval.

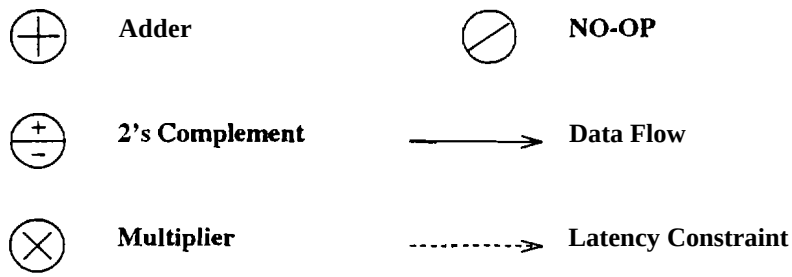

Figure 2: Key to DFG Notation

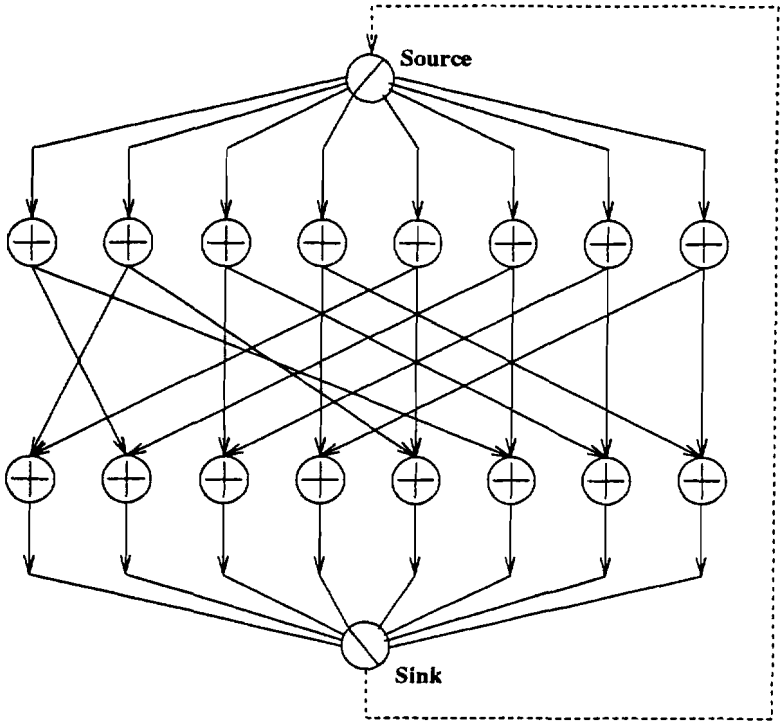

Figure 3: FFT4a - 4 Point FFT with Balanced Paths

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 and output nodes in the DFG connected by a backwards 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. The total execution time of the data path is permitted to exceed the data introduction interval as long as the outputs generated after the maximum latency are only used in the next iteration. In that situation, a maximum latency constraint is applied between the output node and any inputs that; use the output value.

5.2 Characterization of Data Path Resources

A 16 bit ripple carry adder, a 16 bit carry-save multiplier, and a 16 bit register were simulated in HSPICE to obtain propagation delay and power dissipation values under the following conditions. The level 3 MOS model was used with parameter values for a 0.8μ MOSIS process. A load capacitance of 0.1pF was applied to each output signal. Power supplies were set to 5V. Input signals were generated for which an average of 50% of the one bit signals would switch simultaneously every 20ns. Worst case delay was measured and used in the adder delay model. The register setup time requirement is lumped into the adder delay. Multiplier delay

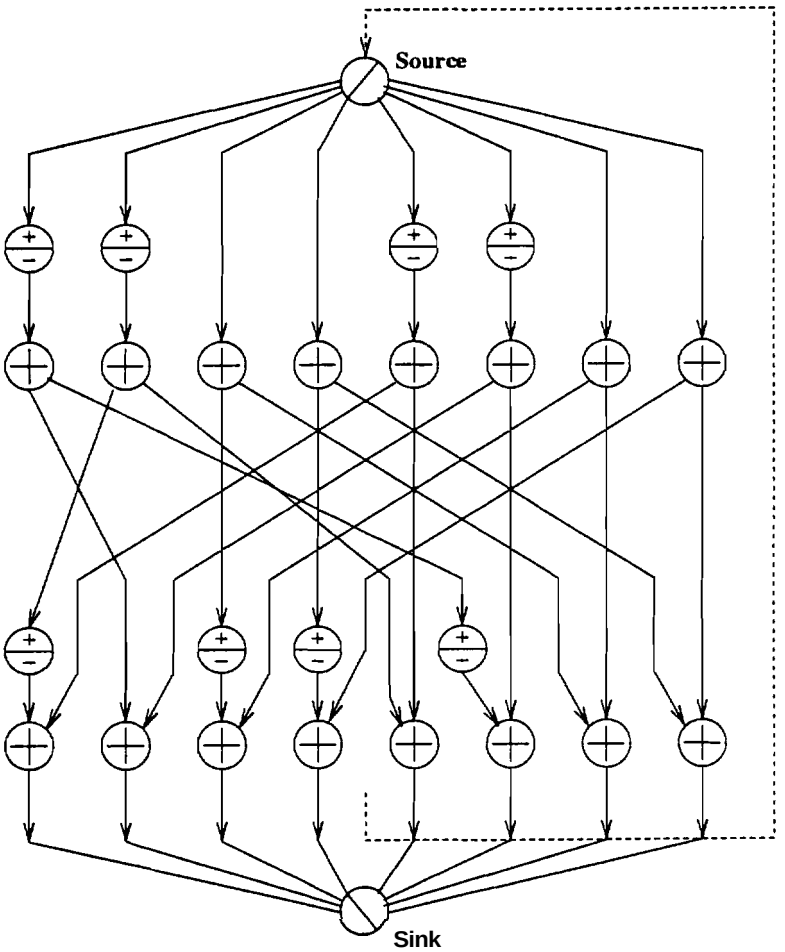

Figure 4: FFT4b - 4 Point FFT with Imbalanced Paths

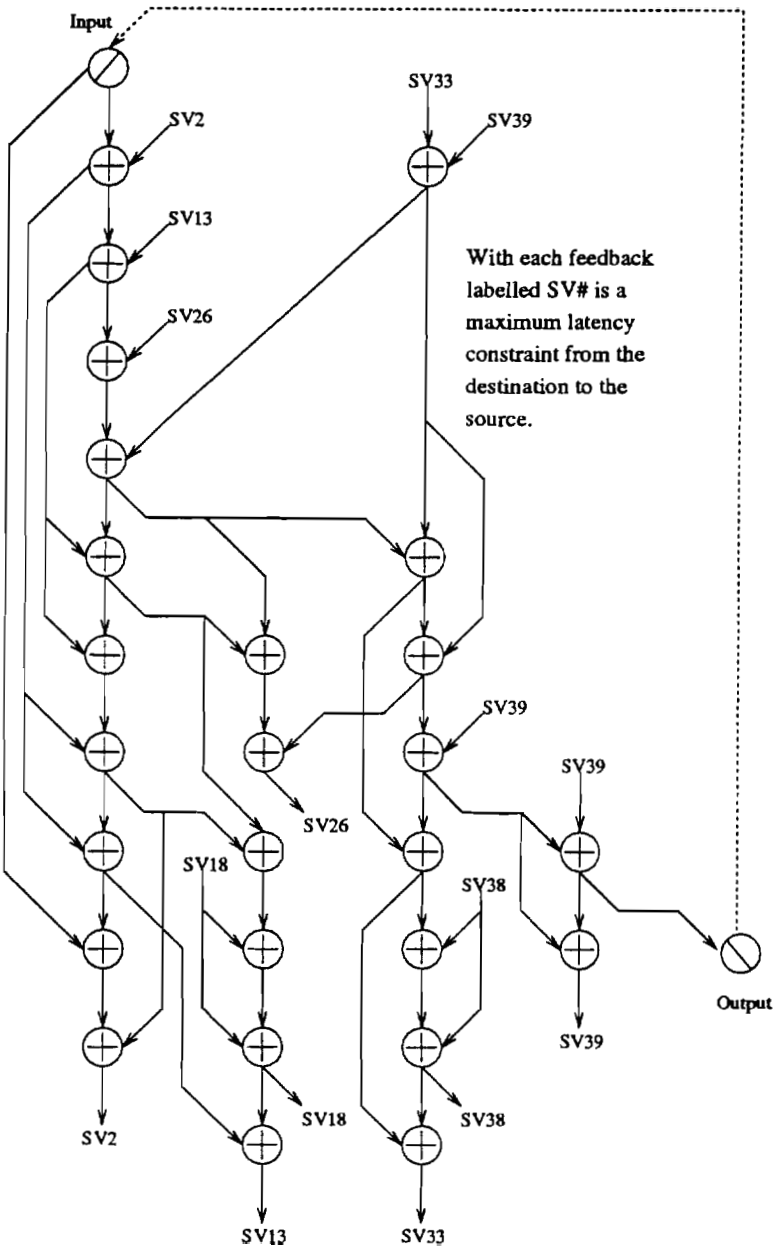

Figure 5: ELLIP - 5th Order Elliptic Wave Filter

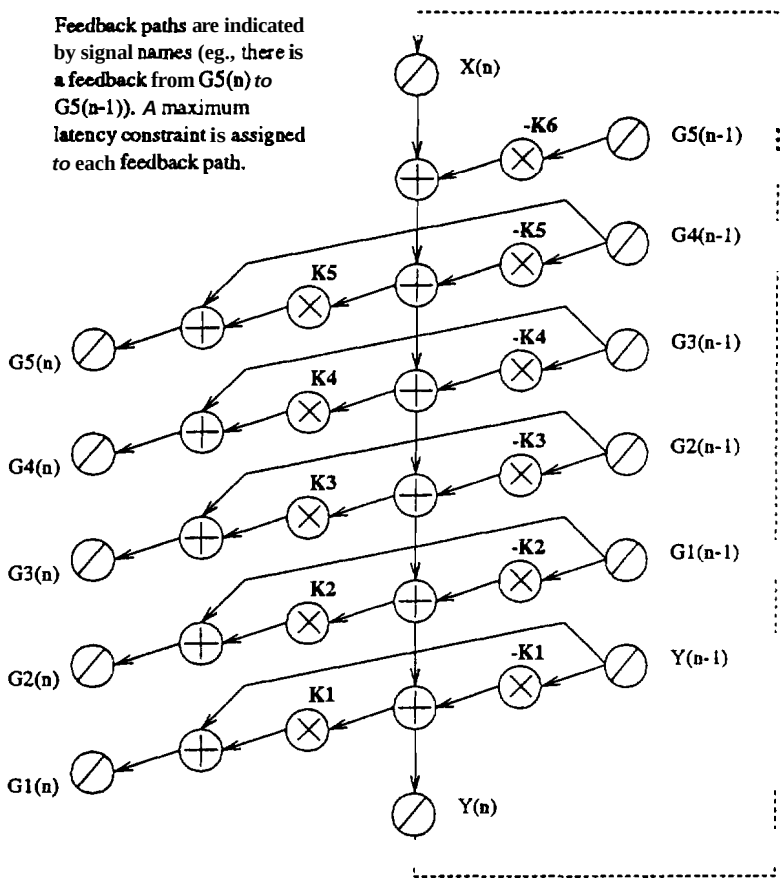

Figure 6: LATTICE - 6th Order Lattice Filter

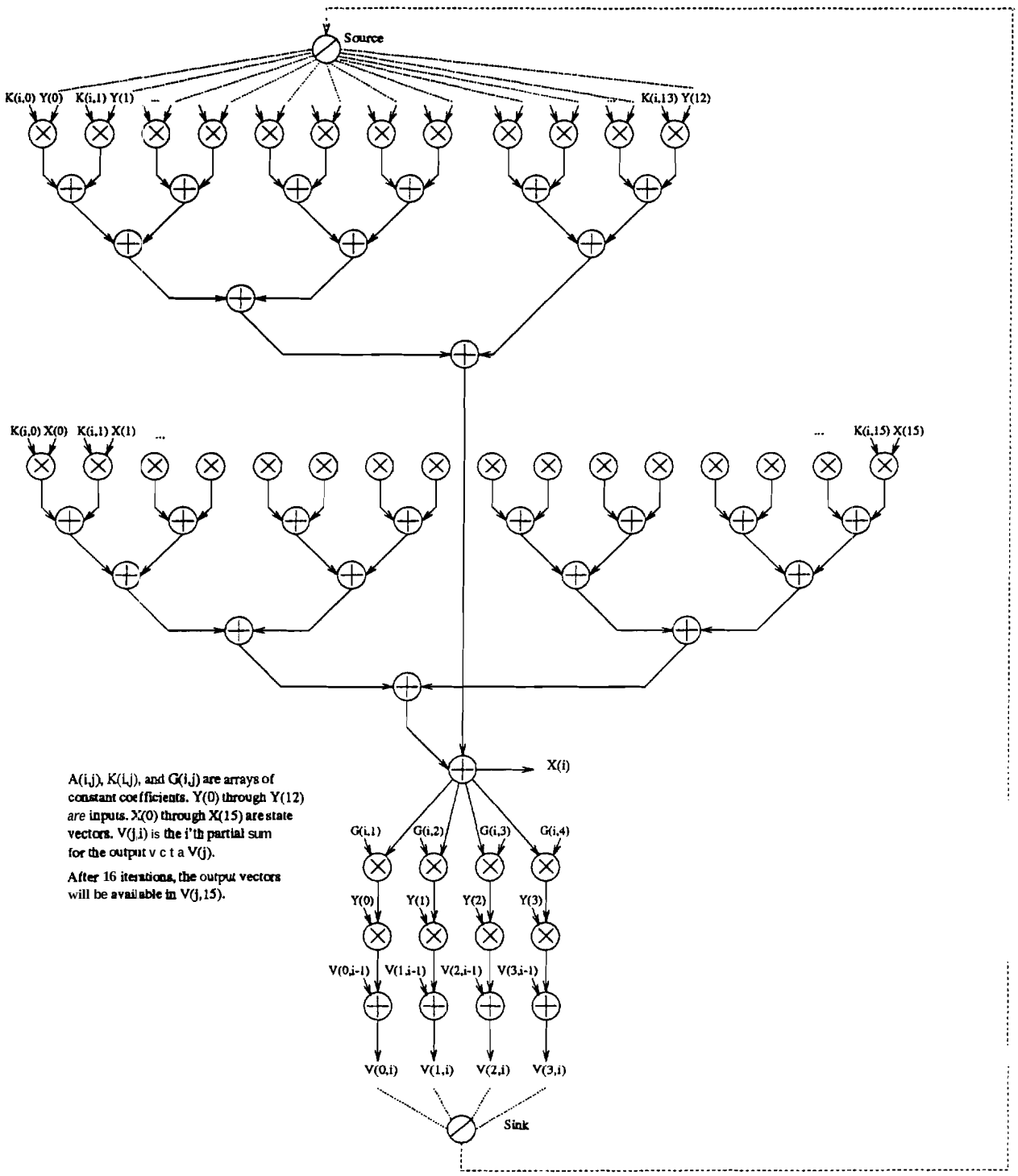

Figure 7: KALMAN - Kalman Filter Benchmark

Table 3: Nominal Power and Delay Measurements

Resource	Power	Energy	Delay
[redacted]	[mW]	[pJ]	
[redacted]	[redacted].3	2965.6	18.5
[redacted]		[redacted]	

was taken to be the largest delay observed for the entire sequence of random input signals. Power dissipation was taken to be the average power dissipation for the sequence of random input values. Nominal operating conditions for the level conversion are the same as for the other resources, except for power supplies and load capacitance. The converter requires two power supplies: the nominal lower supply level was taken to be 3.3V, the higher supply was 5V. Load capacitance was 0.1pF on both sides of the differential output for a total load of 0.2pF. Table 5.2 gives the nominal power, energy and delay values that were measured for each type of resource.

5.3 Optimization Results

Tables 4 and 5 present the results of running the ILP and NLP formulations for each example data path under a variety of voltage supply restrictions. Table 4 reports the results when maximum latency is set equal to the latency of the ASAP schedule. Table 5 reports the results when .maximum latency is set equal to the ASAP latency plus 50%. In both tables, NLP formulation results are reported for all supplies fixed at 5V and for an unlimited selection of supply voltages between 1.5V and 5V. ILP formulation results are reported for 5V fixed supplies, a single optimal supply voltage, an optimal choice of two supply voltages, and an optimal choice of three supply voltages. The ILP formulation was permitted to choose from voltages ranging from 1.5V to 5V in 0.5V increments. Selected voltages were: required to differ by at least 1V. A clock period of 20ns was used for all examples.

Vallues reported in tables 4 and 5 are meant to be interpreted in the following manner. The power estimate is the average energy per switching event divided by the sample period for the data path. "Min Voltage" and "Max Voltage" report the smallest and largest supply voltages assigned to at least one operator by the NLP formulation. "# Converters" indicates the number of level converters needed as a result of the way voltages were assigned to operators. "Voltage", "Voltage 1", etc. all report supply voltages selected by an ILP solution. Next to each supply voltage is an indication of the number of operations of each type to which that voltage was assigned. For example, "(5*)" next to a voltage indicates that five multiplications were assigned to that supply voltage.

Table 4: Power Dissipation and Voltage Selection Results for No Critical Path Slack

Formulation Supply Restrictions	All latencies in clock cyc.	FFT4a Latency = 2	FFT4b Latency = 4	ELLIP Latency = 10	LATTICE Latency = 6	KALMAN Latency = 9
NLP, all 5V	Energy [pJ]	2955	4427	4814	19913	64200
	Power [mW]	73.9	55.3	24.1	165.9	356.7
NLP,	Energy [pJ]	1436	1885	1675	7587	32160
	Power [mW]	35.9	23.6	8.3	63.2	178.7
No Voltage	Max Voltage	3.52	3.5	3.6V	4.9V	4.8V
Restrictions	Min Voltage	3.46	1.9V	1.5V	1.8V	1.8V
	# Converters	0	7	7	8	22
ILP, all 5V	Energy [pJ]	2955	4427	4814	19913	64200
	Power [mW]	73.9	55.3	24.1	165.9	356.7
ILP,	Energy [pJ]	1890	2833	3081	19913	64200
	Power [mW]	47.3	35.4	15.4	165.9	356.7
1 Supply	Voltage	4.0V	4.0V	4.0V	5V	5V
ILP,	Energy [pJ]	1890	2461	2544	10145	47769
	Power [mW]	47.3	30.7	12.7	84.5	265.4
2 Supplies	Voltage 1	4.0V (16+)	2.5V (6+)	2.5V (8+)	3.0V (4+,9*)	4.0V (10+,19*)
	Voltage 2	unused	4.0V (18+)	4.0V (18+)	5.0V (7+,2*)	5.0V (21+,17*)
	# Converters	0	8	7	7	29
ILP,	Energy [pJ]	1800	2461	2508	9817	47769
	Power [mW]	47.3	30.7	12.5	81.8	265.4
3 Supplies	Voltage 1	4.0V (16+)	2.5V (6+)	2.5V (10+)	2.0V (1+)	4.0V (10+,19*)
	Voltage 2	unused	4.0V (18+)	4.0V (15+)	3.0V (4+,8*)	5.0V (21+,17*)
	Voltage 3	unused	unused	5.0V (1+)	5.0V (7+,2*)	unused
	# Converters	0	8	13	6	29

Table 5: Power Dissipation and Voltage Selection:1 Results for a 50% Increase in Critical Path Latency

Formulation Supply Restrictions	All latencies in clock cyc.	FFT4a Latency = 3	FFT4b Latency = 6	ELLIP Latency = 15	LATTICE Latency = 9	KALMAN Latency = 14
NLP, all 5V	Energy [pJ]	2955	4427	4814	19913	64200
	Power [mW]	49.2	36.9	16.0	110.6	229.3
NLP,	Energy [pJ]	871	1076	1052	5189	12305
	Power [mW]	14.5	9.0	3.5	28.8	43.9
No Voltage	Max Voltage	2.72V	3.0V	2.9V	4.9V	3.7V
Restrictions	Min Voltage	2.71V	1.7V	1.5V	1.7V	1.5V
	# Converters	0	2	7	9	36
ILP, all 5V	Energy [pJ]	2955	4427	4814	19913	64200
	Power [mW]	49.2	36.9	16.0	110.6	229.3
ILP,	Energy [pJ]	1890	2833	3081	19913	64200
	Power [mW]	31.5	23.6	10.3	110.6	229.3
1 Supply	Voltage	4.0V	4.0V	4.0V	5V	5V
ILP,	Energy [pJ]	1401	1406	1447	7111	44636
	Power [mW]	23.4	11.7	4.8	39.5	159.4
2 Supplies	Voltage 1	2.5V (8+)	2.5V (20+)	2.5V (23+)	2.5V (5+,10*)	3.0V (9+,18*)
	Voltage 2	4.0V (8+)	4.0V (4+)	4.0V (3+)	5.0V (6+,1*)	5.0V (22+,18*)
	# Converters	16	2	5	7	20
ILP,	Energy [pJ]	1401	1406	1447	6851	38870
	Power [mW]	23.4	11.7	4.8	81.8	138.8
3 Supplies	Voltage 1	2.5V (8+)	2.5V (20+)	2.5V (23+)	2.5V (5+,10*)	2.0V (5+,12*)
	Voltage 2	4.0V (8+)	4.0V (4+)	4.0V (3+)	4.0V (4+)	4.0V (22+,12*)
	Voltage 3	unused	unused	unused	5.0V (2+,1*)	5.0V (4+,12*)
	# Converters	16	2	5	9	26

5.4 Observations

In the preceding results, we are able to observe the effect of the number of supply voltages, data path topology, and latency constraints on our ability to minimize povver dissipation by appropriate scheduling and selection of supply voltages.

For all but one of the data path examples that were evaluated, two appeared to be the optimal number of voltage supplies. Three supply voltages provided little or no reduction in power and sometimes increased the number of level converters required. This may be a consequence of evaluating data paths that are all comparable in size and complexity.

Data path topology seems to have the greatest impact on schedules with minimum latency constrirints. By "minimum latency", we mean that no schedule slack was available on the critical path. In the single supply voltage case, the only usable slack is the difference between the execution time of each operation in the critical path and the nearest multiple of the clock period that is larger, regardless of the topology. For multiple supply voltages, a minimum latency data path with most signal paths of similar length still offers relatively little opportunity for voltage reduction. FFT4a and FFT4b were tailored to demonstrate this effect. The paths in FFT4a are identical in length. The only available schedule slack is the difference between the clock cycle time and the time for an addition. Multiple voltages are of no use in FFT4a. FFT4b has three different path lengths and is able to take advantage of multiple voltages. Among the less trivial examples, the signal paths through the DFG for the KALMAN benchmark were most nearly of similar length. Consequently, the KALMAN benchmark derived less benefit from multiple supply voltages than ELLIP or LATTICE. The LATTICE filter DFG had the greatest variation in signal path lengths and also derived the greatest benefit from multiple supply voltages for the minimum latency constraint case.

Increasing schedule latency by 50% doesn't improve the results much for a single supply voltage, but multiple voltage results are enhanced, and the influence of topology is reduced. In the single supply voltage minimum latency case, a lowering of voltage increases the delay of all data path operations. If the data path was already voltage scaled for minimum latency, a small voltage drop may cause the delay of many data path operators to exceed the next multiple of a clock cycle and cause the data path latency to increase by more than 50%. In the multiple supply voltage case, voltage reductions can be selectively applied to individual operators to take up just the amount of schedule slack that is available. Topology becomes less important with increased latency, since unbalanced signal paths are not needed to provide slack for voltage scaling.

6 Conclusions

In this paper we have presented a method, MPSVS, for using integer programming to optimize the schedule and supply voltage levels for a mixed voltage data path design. The primary benefit of MPSVS is to obtain a data path schedule and supply voltage assignments that minimize data path power dissipation. However, there are some beneficial side effects. Use of level conversions should be lower than for multiple voltage scheduling algorithms that ignore level conversion costs. Fewer level conversions should result in larger portions of the data path that operate with a single supply voltage, simplifying layout and routing. Lowering supply

voltages to signal paths with relatively large schedule slack will balance the delay paths and should help reduce glitching activity.

Running MPSVS on a variety of data path examples resulted in the followvving observations. In all but a perfectly balanced data path example, the optimal number of supply voltages turned out to be two. When minimum latency constraints are applied, no voltage scaling can be applied unless there are signal paths shorter than the critical path. Loosening the latency constraints allowed lower voltages to be selected, but the optimal number of supply voltages still appeared to be two.

There are a number of extensions to MPSVS that would improve the quality of the optimization results and bring them closer to a realistic specification of a data path. Useful extensions would include the addition of data path resource constraints, module selection and binding, support for retiming, support for conditional operations and loops, and including the effect of schedule changes on switching. Problems of particular interest are the incorporation of resource constraints and module binding which will have to be reformulated to account for assignment of a voltage to each module.

References

[1] A.P. Chandrakasan, R. Allmon, A. Stratakos, R.W. Brodersen, "Design of Portable Systems," ***IEEE*** 1994 ***Custom Integrated Circuits Conference,*** San Diego, CA, pp.259-266.

[2] S. Chaudhuri, R.A. Walker, and J.E. Mitchell, "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,*** Dec. 1994, pp.456-471.

[3] K. Usami and M. Horowitz, "Clustered Voltage Scaling Technique for Low-Power Design", ***Proceedings of the International Symposium on Low Power Design*** 1995, New York, pp.3-8.

[4] A.P. Chandrakasan, et. al., "Optimizing Power Using Transformations," ***IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*** Vol.14, No.1, Jan. 1995, pp.12-31.

[5] N. Kumar, S. Katkoori, L. Rader, and R. Vemuri, "Profile-Driven Behavioral Synthesis for Low-Power VLSI Systems," ***IEEE Design & Test of Computers,*** Fall 1995, pp.70-84.

[6] A. Raghunathan and N.K. Jha, "Behavioral Synthesis for Low Power," ***Proceedings - IEEE International Conference on Computer Design: VLSI in Computers and Processors*** 1994, pp.318-322.

[7] A. Raghunathan and N.K. Jha, "An Iterative Improvement Algorithm for Low Power Data Path Synthesis," ***Proceedings of the Internaiional Conference on Computer Aided Design*** 1995, pp.597-602.

[8] R. San Martin and J.P. Knight, "Power-Profiler: Optimizing ASICs Power Consumption at the Behavioral Level," ***Proceedings 32nd Design Automation Conference,*** June 1995, San Francisco, pp.42-47.

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

[10] C.H. Gebotys and M.I. Elmasry, "A Global Optimization Approach for Architectural Synthesis", IEEE Transactions on *CAD/ICAS*, Vol.12, No.9, pp.1266-12'78, Sep 1993.

[11] L. Goodby, A. Orailoglu, and P.M. Chau, "Microarchitectural Synthesis of Performance-Constrained, Low-Power VLSI Designs," Proceedings - IEEE International Conference on Computer Design: *VLSI* in Computers and Processors 1994, pp.323-326.

[12] J. Rabaey, Digital Integrated Circuits: A Design Perspective, Prentice Hall, Englewood Cliffs, N.J., to be published.

[13] Anthony Brooke, David Kendrick, and Alexander Meeraus, GAMS A User's Guide, The Scientific Press, 1992.

[14] G. DeMicheli, Synthesis and Optimization of Digital Circuits, McGraw-Hill, Inc., 1994.

[15] A.P.Chandrakasan, S.Sheng, and R.W.Brodersen, "Low-Power CMOS Digital Design," Journal of Solid-State Circuit, Vol.27, No.4, April 1992, pp.473-483.

[16] D. S. Rao, "The Fifth Order Elliptic Wave Filter Benchmark" benchmark set: HLSynth92 http://www.cbl.ncsu.edu/www/CBL_Docs/Bench.htm

[17] J.G. Proakis, D.G. Manolakis, Digital Signal Processing Principles, Algorithms, and Applications Macmillan Publishing Company, Inc., 1992.

[18] C. Ramachandran, "Kalman Filter Benchmark", benchmark set: HLSynth92 http://www.cbl.ncsu.edu/www/CBL_Docs/Bench.htm

[19] *HSPICE* User's Manual, Meta-Software, Inc., 1995.