

Converter-Free Multiple-Voltage Scaling Techniques for Low-Power CMOS Digital Design

Yi-Jong Yeh, Sy-Yen Kuo, and Jing-Yang Jou

Abstract—Recent research has shown that voltage scaling is a very effective technique for low-power design. This paper describes a voltage scaling technique to minimize the power consumption of a combinational circuit. First, the converter-free multiple-voltage (CFMV) structures are proposed, including the p-type, the n-type, and the two-way CFMV structures. The CFMV structures make use of multiple supply voltages and do not require level converters. In contrast, previous works employing multiple supply voltages need level converters to prevent static currents, which may result in large power consumption. In addition, the CFMV structures group the gates with the same supply voltage in a cluster to reduce the complexity of placement and routing for the subsequent physical layout stage. Next, we formulated the problem and proposed an efficient heuristic algorithm to solve it. The heuristic algorithm has been implemented in C and experiments were performed on the ISCAS85 circuits to demonstrate the effectiveness of our approach.

Index Terms—CFMV, clustered voltage scaling, converter-free, CVS, low-power, multiple-voltage, voltage scaling.

I. INTRODUCTION

A. Background and Related Work

Power dissipation has become one of the most significant parameters in very large scale integration design due to the trend toward portable computing and communications systems. For portable devices, power dissipation limits the battery life and the available time. Even for non-portable devices, power dissipation affects the cost of packaging and cooling equipment.

Total power dissipation in a digital CMOS circuit can be obtained from the sum of three components: static dissipation, dynamic dissipation, and short-circuit dissipation [1]. In general, the total power dissipation is dominated by the dynamic dissipation and may be estimated by $P_d = a \cdot f_{\text{clk}} \cdot C_L \cdot V_{DD}^2$, where a is the activity factor, f_{clk} the switching frequency, C_L the total node capacitance, and V_{DD} the supply voltage. This formula is the basis of previous researches in low-power CMOS digital design [2]–[11].

As the dynamic power dissipation is proportional to the square of the supply voltage, voltage scaling is evidently the most effective technique to minimize the power dissipation. Moreover, the conclusion of [8] provides us a clear goal in minimizing the power dissipation, i.e., operate the circuits as slowly as possible, with the lowest possible supply voltage.

The most popular voltage scaling technique is to operate all the gates in a circuit with a reduced supply voltage, which is limited by the critical paths. However, the gates that are not on the critical paths could operate slower with lower supply voltages. This motivated some researchers to operate gates with two or more supply voltages in a circuit [10]–[12].

Manuscript received September 17, 1998; revised July 10, 2000. This work was supported by the National Science Council, Taiwan, R.O.C., under Grant NSC 89-2219-E-002-003. This paper was recommended by Associate Editor M. Redram.

Y.-J. Yeh and S.-Y. Kuo are with the Department of Electrical Engineering, National Taiwan University, Taipei Taiwan, R.O.C. (e-mail: yeh@lion.ee.ntu.edu.tw; sykou@cc.ee.ntu.edu.tw).

J.-Y. Jou is with the Department of Electronics Engineering, National Chiao Tung University, Hsinchu Taiwan, R.O.C. (e-mail: jjjou@bestmap.ee.nctu.edu.tw).

Publisher Item Identifier S 0278-0070(01)00362-1.

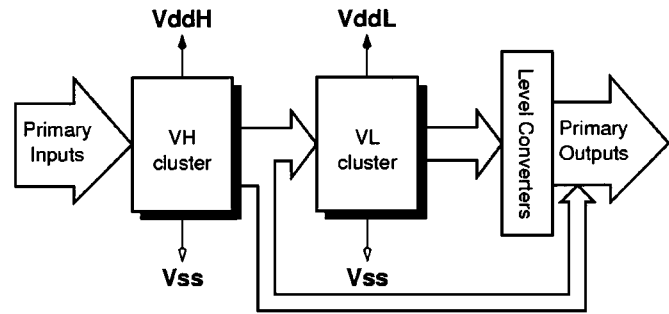


Fig. 1. Block diagram of the CVS structure.

Usami *et al.* proposed a clustered voltage scaling (CVS) technique to reduce the power consumption with two supply voltages [11]. The block diagram of CVS is shown in Fig. 1. They arranged supply voltages such that the voltage swings of all paths are in decreasing order. Then level converters and latches with the level-conversion function are inserted before the primary outputs to prevent the static current.

B. Motivation and Goals

In previous works, level converters and latches with the level-conversion function were inserted to prevent the static current in circuits with multiple supply voltages. However, there exist some overheads with level converters.

First, the power consumption of level converters is not negligible. From our circuit simulation, the power consumption of a level converter is about four times that of an inverter. Next, the insertion of level converters introduces extra delays into the circuits. The rising delay of a level converter is about four times that of an inverter, too. Finally, the insertion of level converters changes the topology of circuits.

Consequently, Usami *et al.* used level converters with care and just inserted level converters in front of the primary outputs to minimize the number of level converters [11]. Instead of using level converters with care, we try to find a voltage scaling technique without level converters in this paper.

To reduce the complexity of placement and routing when multiple supply voltages are used in physical layout, gates with the same supply voltage should be placed in a cluster. This is especially important for a standard-cell design since the gates in a standard-cell design are arranged in rows and their power lines are connected directly. Hence, we would like to preserve the clustering property in this paper as Usami *et al.* did in the CVS technique.

Finally, the logic structure discussed in this paper is CMOS complementary logic. Other logic structures may have more sophisticated effects at the interface of different supply voltages, which is beyond the scope of this paper.

The rest of this paper is organized as follows. In Section II, we propose the converter-free multiple-voltage (CFMV) structures, which need no level converters, make use of multiple voltages, and have gates with the same supply voltage in a cluster. In Section III, we give some definitions, formulate our problem, and propose a heuristic algorithm for the problem. Then, experimental results are shown in Section IV and are compared to the results of previous work. Finally, concluding remarks and future works are provided in Section V.

II. CFMV STRUCTURES

A. Elimination of Level Converters

When multiple supply voltages are applied in a CMOS digital circuit, there might exist a static current flowing from the supply voltage to the

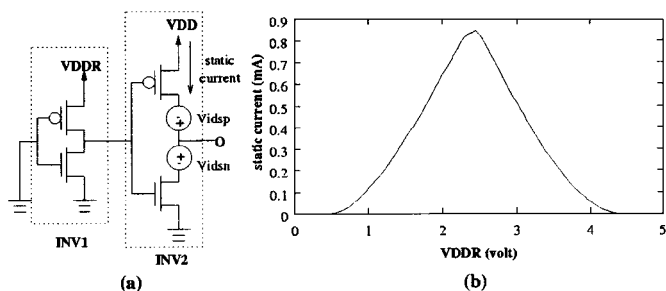


Fig. 2. (a) Inverter with VDDR drives another inverter with VDD. (b) Static currents of various reduced supply voltages with $|V_{tp}| = 0.9$ V, $V_{tn} = 0.75$ V, and $V_{DD} = 5$ V.

ground at the interface of gates with different supply voltages. Level converters are usually used at the interface to prevent the static current. To avoid using level converters in a CMOS circuit with multiple supply voltages, we put constraints on the voltage differences between adjacent gates with different supply voltages.

A simple analysis with the first-order MOS model predicts that there will be no static current if the supply voltage of a driver gate is higher than the subtraction of the threshold voltage of a PMOS from the supply voltage of a driven gate. Take Fig. 2(a) as an example where a CMOS inverter INV1 with the reduced supply voltage V_{DDR} is connected directly to another CMOS inverter INV2 with the unreduced supply voltage V_{DD} . Then, the output voltage of INV1 will be V_{DDR} , which is also the input voltage of INV2. If $V_{DDR} > V_{DD} - |V_{tp}|$, i.e., V_{SG} of the PMOS in INV2 is less than $|V_{tp}|$, the PMOS in INV2 will be OFF and there will be no static current flowing from the supply voltage to the ground through INV2.

However, the subthreshold effect makes the above prediction imprecise [13]. As we can see from Fig. 2(b), when V_{DDR} equals 4.1 V ($V_{DD} - |V_{tp}|$), the static current is about 31.4 μ A,¹ which seems large and unacceptable. Therefore, the best way to determine the reduced voltage is by a circuit simulator, such as HSPICE, when the acceptable value of the static current is given. For example, we could use HSPICE to simulate the circuit in Fig. 2(a). If the acceptable value of the static current is 1 μ A, we can determine V_{DDR} to be 4.4 V with which the static current will be less than 1 μ A.

B. Arrangement of Supply Voltages

From the above section, we know that if the voltage difference of a driven gate and its driver gate is less than a specific value, level converters are not necessary at the interface of gates with different supply voltages. We call such value a *safe threshold voltage*, denoted as V_{st} . Next, we'll discuss how to arrange multiple supply voltages in a circuit to eliminate level converters.

Assume that we have a set of n supply voltages, $\{V_{dd0}, V_{dd1}, \dots, V_{dd_{n-1}}\}$, such that

- 1) $V_{dd0} > V_{dd1} > \dots > V_{dd_{n-1}}$,
- 2) $V_{ddi} - V_{dd_{i+1}} < V_{st}$ for $i = 0, 1, \dots, (n-2)$,
- 3) $V_{ddi} - V_{dd_j} > V_{st}$ for $i - j > 1$.

Also, we have a set of n clusters of gates $\{C_0, C_1, \dots, C_{n-1}\}$, where the gates in different clusters are supplied with different voltages and each cluster has two adjacent clusters at most. If we like to assign these supply voltages to the clusters such that there is no static current and no level converters, the only solution will be as shown in Fig. 3.

We call the structure shown in Fig. 3 a *p-type CFMV structure* if all clusters have the same V_{ss} . Similarly, if all clusters have the same V_{dd} and $V_{ss0}, V_{ss1}, \dots, V_{ss_{n-1}}$ are in increasing order, we call

¹All the circuit simulations of this paper used the level-3 SPICE models of the TSMC 0.8- μ m single-poly double-metal process.

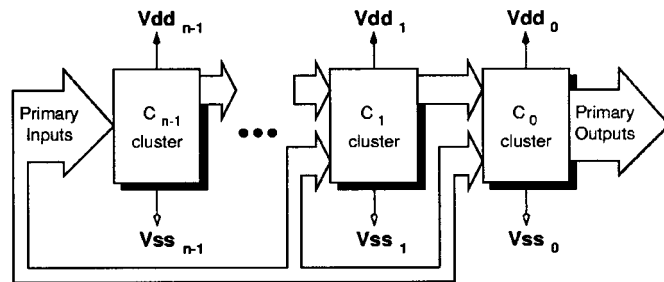


Fig. 3. Block diagram of the CFMV structure.

it an *n-type CFMV structure*. If both V_{dd} and V_{ss} are scalable, it is called a *two-way CFMV structure*. Whichever CFMV structure is used, the voltage swings along all paths are in increasing order. The set of supply voltages $\{(V_{dd0}, V_{ss0}), (V_{dd1}, V_{ss1}), \dots\}$ is called the *feasible supply voltage set*.

III. ALGORITHMS FOR THE CFMV STRUCTURES

A. Preliminaries

A combinational circuit can be represented as a *directed acyclic graph* $G = (V, E)$ consisting of two sets: a finite set V of elements called *vertices* and a finite set E of elements called *edges*. Each vertex $v \in V$ is in one-to-one correspondence with a gate in the circuit and is associated with a *delay* $d(v)$, which is the delay of the gate. There is an edge denoted by an ordered pair $(u, v) \in E$ if the output of gate $u \in V$ is connected to an input pin of gate $v \in V$.

Definition 1 (Fan-Out Set): For any vertex $v \in V$, the *fan-out set* $\Gamma^+(v) = \{w | (v, w) \in E\}$.

Definition 2 (Stable Time): The time when the output of vertex u becomes stable is called the *stable time* of vertex u , denoted as $T_s(u)$.

Definition 3 (Required Time): The *required time* of vertex u , denoted as $T_r(u)$, is the latest time when the output of vertex u has to be stable to meet the timing constraint of the circuit.

Definition 4 (Slack): The *slack* of vertex u , denoted as $s(u)$, is the maximal delay increase which vertex u may have under the timing constraint. When the stable time and the required time of vertex u are computed, the slack of vertex u can be obtained by $s(u) = T_r(u) - T_s(u)$.

Definition 5 (Depth): Similar to the definition of *level* in [14], we can define the *depth* of a vertex u in a graph G to be the number of edges in the longest path from u to a sink of G . The depth of a sink is defined to be zero and the depth of u , denoted as $\text{dep}(u)$, can be determined by $\text{dep}(u) = 1 + \max_{v \in \Gamma^+(u)} \text{dep}(v)$. In addition, the *depth* of a graph $G = (V, E)$ is defined by $\max_{v \in V} \text{dep}(v)$.

Definition 6 (Reachable): If there is a path p from u to v , we say that v is *reachable* from u via p and is denoted by $u \xrightarrow{p} v$.

Definition 7 (Reachable Set): For any vertex $v \in V$, the *reachable set* of v is

$$R(v) = \{w | \text{there exists a path } p \text{ such that } v \xrightarrow{p} w\}.$$

Definition 8 (Cut): Let V_1 and V_2 be two mutually disjoint subsets of V such that $V = V_1 \cup V_2$; i.e., V_1 and V_2 have no common vertices and together contain all the vertices of V . Then the set of all those edges of G having one end vertex in V_1 and the other in V_2 is called a *cut* of G . This is denoted as $\langle V_1, V_2 \rangle$. The removal of $\langle V_1, V_2 \rangle$ partitions G into two graphs G_1 and G_2 , which are the induced subgraphs of G on the vertex sets V_1 and V_2 .

Definition 9 (Directed Cut): A cut $\langle V_1, V_2 \rangle$ whose edges are all from V_1 to V_2 is called a *directed cut*, which is denoted by $[V_1, V_2]$.

Definition 10 (Boundary Vertex): Let $[V_1, V_2]$ be a directed cut of G , and G_1, G_2 be the induced subgraphs of G on V_1 and V_2 . Then, a

```

DFS( $m$ )
1  For (each vertex  $v$  with voltage level  $m$ ) Do
2    DFS-Visit( $v, m$ );

DFS-Visit( $v, m$ )
1  If ( $v$  is marked) Then
2    return;
3  If ( $v$  is a sink or a boundary vertex) Then
4    Mark  $v$ ;
5  Else
6    For (each fanin vertex  $u$  of  $v$ ) Do
7      DFS-Visit( $u, m$ );
8  If (all the voltage levels of  $v$ 's fanins are  $(m + 1)$ ) Then
9    set  $v$ 's voltage level to  $(m + 1)$ ;
10   If (there exists negative slack) Then
11     set  $v$ 's voltage level back to  $m$ ;
12   Mark  $v$ ;

```

Fig. 4. Algorithm for two supply voltages.

vertex $v \in V_1$ is called a *boundary vertex* of G_1 if there exists a vertex $u \in V_2$ such that $(v, u) \in [V_1, V_2]$.

Next, we present a lemma before the definition of *proper-directed cut*.

Lemma 1: Let $[V_1, V_2]$ be a directed cut of G and v be a vertex in V_2 . Then, $R(v) \subset V_2$.

From Lemma 1, we know that if a vertex is in V_2 , then all the vertices in its reachable set must be included in V_2 as well. Now we can define the *proper-directed cut*, which will be used to partition graphs in the following subsections.

Definition 11 (Proper-Directed Cut): $[V_1, V_2]$ is called a *proper-directed cut* of G if V_2 contains all the sinks of G , all the boundary vertices of G , and all the vertices in their reachable set. $\text{pdc}(G)$ denotes a set consisting of all the proper-directed cuts of G . Take Fig. 7 as an example, where C1 is a proper-directed cut but C2 is not because the vertex d is a sink but not in the right hand side of C2.

B. Problem Formulation

Now we can formulate the problem that we would like to solve in this paper as the following. Given a circuit with timing constraints and a feasible supply voltage set, scale the supply voltages of a subset of gates with positive slacks to minimize the total power consumption for the CFMV structure.

Note that the formulated problem is for the CFMV structure and then has a much smaller solution space than a generic multiple-voltage scaling problem since the voltage sequence in the CFMV structure is a continuous subsequence of the feasible supply voltage set.

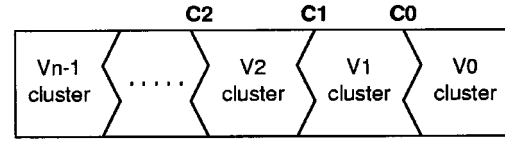
When there are only two elements in the given feasible supply voltage set, the optimal solution can be easily obtained by a depth-first search algorithm shown in Fig. 4. However, when there are more than two elements in the given feasible supply voltage set, the problem becomes much more difficult. Thus we next give an asymptotic bound on the solution space of the formulated problem.

Theorem 1: Assume that a graph $G = (V, E)$ is partitioned into n clusters as shown in Fig. 5 as a solution of the formulated problem. Let $\Upsilon_i = \bigcup_{k=i}^{n-1} V_k$, G_i be the induced subgraph of G on Υ_i , and C_i be the cut between Υ_{i+1} and V_i . Then, C_i is a proper directed cut of G_i , for $i = 0, 1, \dots, (n - 2)$.

Theorem 2: The number of elements in $\text{pdc}(G)$ is $\Omega(2^{n/(p+1)})$, where n is the number of vertices in the graph G and p is the depth of G .

C. Heuristic Algorithm

Since the number of proper-directed cuts of a graph is exponentially proportional to its number of vertices, it is impractical to search all the

Fig. 5. Solution of the formulated problem with n voltage levels.

proper-directed cuts for the optimal solution. Therefore, we propose a heuristic algorithm to search only a subset of the solution space.

Let G' be the induced subgraph of G on the vertices whose voltage level is m . After $\text{DFS}(m)$ is applied, G' is partitioned by a proper-directed cut $[V_1, V_2]$, where the voltage level of the vertices in V_1 is $(m + 1)$ and in V_2 is m . Next, we give some theorems from which our heuristic algorithm is derived from.

Theorem 3: Let $[V'_1, V'_2]$ be a proper-directed cut of G' and $V'_1 \supset V_1$, where $[V_1, V_2]$ is obtained from $\text{DFS}(m)$. Then, there must exist a negative slack if G' is partitioned by $[V'_1, V'_2]$.

Theorem 4: Let $[V'_1, V'_2]$ be a proper-directed cut of G' and $V'_1 \subset V_3$, where $[V_3, V_4]$ is a proper-directed cut such that there is no negative slack. Then, there is no negative slack when G' is partitioned by $[V'_1, V'_2]$.

Theorem 5: Let $[V_3, V_4]$ be a proper-directed cut of G' such that there is no negative slack and $S = \{v | \Gamma^+(v) \subset V_4, \forall v \in V_3\}$. Then, the cut $(V_3 - S, V_4 \cup S)$ is a proper-directed cut of G' such that there is no negative slack.

Theorem 3 shows that $\text{DFS}()$ can give us a bound of feasible solutions. Theorem 4 shows how to find other feasible solutions when a feasible solution is given. Though the solution space can be narrowed down by $\text{DFS}()$, the remaining solution space is still exponentially proportional to the number of the remaining vertices. Hence, we use Theorem 5, which is implemented as $\text{Fwd} - \text{One} - \text{Layer}()$ in this paper, to search for potential solutions with a practical complexity.

Based on Theorems 3–5, we propose a heuristic algorithm $\text{CFMV}()$, shown in Fig. 6, to solve the formulated problem. Given n elements in the feasible supply voltage set, we first initialize the voltage level of each vertex to zero, and then apply $\text{CFMV}(n - 1, 0)$ to a graph $G = (V, E)$. By way of illustration, let's take a look at how $\text{CFMV}(2, 0)$ works.

Initially, the voltage level of each vertex is zero, $ML = \phi$ and $MP = \infty$. Then, $\text{CFMV}(2, 0)$ calls $\text{DFS}(0)$ to assign the voltage levels of $\{j, k, \dots, r\}$ to one. Next, $\text{CFMV}(0, 0)$ is called such that $L1 = \{(a, 0), (b, 0), \dots, (i, 0)\}$ and $P1 = 9.00$. Then, $\text{CFMV}(2, 1)$ is called and it calls $\text{DFS}(1)$. Since there is no positive slack, $\text{DFS}(1)$ does not update the voltage level of any vertex. When $\text{CFMV}(2, 1)$ returns, $L2 = \{(j, 1), (k, 1), \dots, (r, 1)\}$ and $P2 = 5.76$. Since $(P1 + P2 = 14.76) < (MP = \infty)$, $MP = 14.76$ and $ML = \{(a, 0), \dots, (i, 0), (j, 1), \dots, (r, 1)\}$.

Next, $\text{Fwd} - \text{One} - \text{Layer}(0)$ is called to assign the voltage levels of $\{j, k, l\}$ to zero. Then $\text{CFMV}(0, 0)$ is called again such that $L1 = \{(a, 0), (b, 0), \dots, (l, 0)\}$ and $P1 = 12.00$. Next, $\text{CFMV}(2, 1)$ is called and it calls $\text{DFS}(1)$, which assigns the voltage levels of $\{p, q, r\}$ to two. The ML of $\text{CFMV}(2, 1)$ is then $\{(m, 1), (n, 1), (o, 1), (p, 2), (q, 2), (r, 2)\}$. Before $\text{CFMV}(2, 1)$ returns, the voltage levels of $\{p, q, r\}$ are assigned back to one. When $\text{CFMV}(2, 1)$ returns, $L2 = \{(m, 1), (n, 1), (o, 1), (p, 2), (q, 2), (r, 2)\}$ and $P2 = 3.00$. Since $(P1 + P2 = 15.00) > (MP = 14.76)$, MP and ML remain unchanged.

Next, $\text{Fwd} - \text{One} - \text{Layer}(0)$ is called again to assign the voltage levels of $\{m, n, o\}$ to zero. Then, $\text{CFMV}(0, 0)$ is called such that $L1 = \{(a, 0), (b, 0), \dots, (o, 0)\}$ and $P1 = 15.00$. Next, $\text{CFMV}(2, 1)$ is called again such that $L2 = \{(p, 1), (q, 1), (r, 1)\}$ and $P2 = 1.92$.

```

CFMV(b,a)
1  ML =  $\phi$ ; MP =  $\infty$ ;
2  If (b==a) Then
3    For each vertex v Do
4      If (v.vl==b) Then ML = ML  $\cup$  {(v,b)};
5      MP = power of ML;
6      Return (MP,ML);
7  If (b==(a + 1)) Then
8    DFS(a);
9    For each vertex v Do
10     If (v.vl==a) Then ML = ML  $\cup$  {(v,a)};
11     If (v.vl==b) Then
12       ML = ML  $\cup$  {(v,b)};
13       v.vl = a;
14     MP = power of ML;
15     Return (MP,ML);
16  DFS(a);
17  While (there exists any vertex whose voltage level is (a + 1))
18    (P1,L1)=CFMV(a,a);
19    (P2,L2)=CFMV(b,a + 1);
20    If ((P1 + P2);MP) Then
21      MP = P1 + P2;
22      ML = L1  $\cup$  L2;
23      Fwd-One-Layer(a);
24    L3 =  $\phi$ ;
25    For each vertex v Do
26      If (v.vl==a) Then L3 = L3  $\cup$  {(v,a)};
27    P3 = power of L3;
28    If (P3;MP) Then
29      MP = P3;
30      ML = L3;
31  Return (MP,ML);

```

Fig. 6. Heuristic algorithm for the formulated problem.

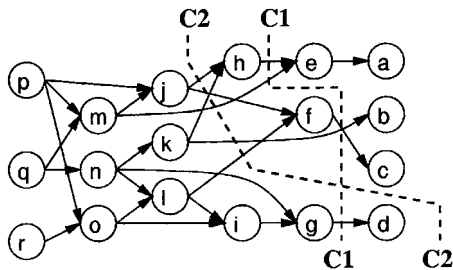


Fig. 7. Graph for the illustration of CFMV(2, 0).

Since $(P1 + P2) > MP$, MP and ML remain unchanged. When Fwd - One - Layer(0) is called, the voltage levels of $\{p, q, r\}$ are assigned zero.

Now the voltage level of each vertex is zero. So, $L3 = \{(a, 0), \dots, (r, 0)\}$ and $P3 = 18.00$. Since $P3 > MP$, MP and ML remain unchanged. Finally, (MP, ML) is the best solution found by CFMV(2, 0).

In the following, we give an asymptotic bound on the computation complexity of CFMV.

Theorem 6: Let n be the number of vertices in the graph G and l be the number of elements in the feasible supply voltage set. Then, the computation complexity of CFMV on G is $O(n^{l-1})$, for $l = 2, 3, \dots$

IV. EXPERIMENTAL RESULTS

We have implemented our heuristic algorithm in C on a Pentium II 450 PC running Linux (RedHat 6.0) with 128-MB memory, and performed experiments on all the ISCAS85 circuits. In addition, we implemented the CVS technique for comparison.

The experiment environment is shown in Fig. 8. The control file provides the feasible supply voltage set. In our experimental cell library, the length of each MOS is $0.8 \mu\text{m}$, the width of each PMOS is 16.8

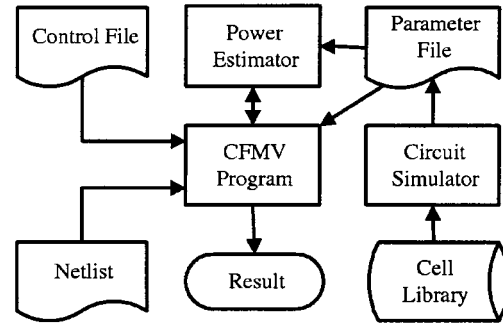


Fig. 8. Experiment environment.

TABLE I
FEASIBLE SUPPLY VOLTAGE SETS USED IN THE EXPERIMENT

types	(V_{dd0}, V_{ss0})	(V_{dd1}, V_{ss1})	(V_{dd2}, V_{ss2})	(V_{dd3}, V_{ss3})
P-type	(5.0, 0.0)	(4.4, 0.0)	(3.8, 0.0)	(3.2, 0.0)
N-type	(5.0, 0.0)	(5.0, 0.4)	(5.0, 0.8)	(5.0, 1.2)
2-way	(5.0, 0.0)	(4.4, 0.4)	(3.8, 0.8)	(3.2, 1.2)

μm and the width of each NMOS is $8 \mu\text{m}$. Using HSPICE to simulate each gate in the cell library, we obtained the parameters for timing and power analysis.

From [1], the rising delay T_{dLH} of a gate v is estimated by

$$T_{dLH} = (\text{rise } a0) + (\text{rise } a1) \times C_{\text{out}} \quad (1)$$

where C_{out} is the sum of the output capacitance of gate v and the input capacitances of its fan-outs. The falling delay is estimated similarly. If the supply voltage of a gate is scaled to (V'_{dd}, V'_{ss}) , its rising delay is estimated by

$$T'_{dLH} = T_{dLH} \times \frac{V'_{dd} - V'_{ss}}{V_{dd} - V_{ss}} \times \frac{(V_{dd} - V_{ss} - V_{thp})^2}{(V'_{dd} - V'_{ss} - V_{thp})^2}. \quad (2)$$

For the power analysis, the activity factor of each primary input is assumed to be 0.5 and the activity factors of other gates are computed accordingly. Then, the power consumption P_d of a gate v with supply voltages (V'_{dd}, V'_{ss}) , can be estimated by

$$P_d = \frac{1}{2} \times f \times \alpha \times (V'_{dd} - V'_{ss})^2. \quad (3)$$

The feasible supply voltage sets used in our experiments are shown in Table I. When n voltage levels are used, the feasible supply voltage set is $\{(V_{dd0}, V_{ss0}), \dots, (V_{dd(n-1)}, V_{ss(n-1)})\}$.

First of all, we compare the results of CFMV with those of CVS to show the effectiveness of the CFMV technique. Since the CVS technique uses two supply voltages, we compare it with the CFMV technique with two voltage levels. The comparison results are shown in Table II. We can find that the CFMV technique is better than the CVS technique in most cases, except in c880 and c5315. On average, the power reduction of CVS (5 V, 4 V) is 7.17%, CVS (5 V, 3 V) is 8.99%, and two-way CFMV (2 levels) is 13.65%.

Next, we perform experiments on three types of CFMV structures with more voltage levels to find the effect of voltage levels as shown in Table III. We find that the more voltage levels are provided, the more power reduction we can obtain. For example, from Table III, the average power reduction of a two-way CFMV with two voltage levels is 13.65%, three voltage levels is 18.05%, and four voltage levels is 18.73%. Though more power reduction can be obtained with more voltage levels, the increment of power reduction is less with more voltage levels. It is a tradeoff between the power reduction and the cost of voltage levels.

TABLE II
COMPARISON RESULTS OF CVS AND CFMV

circuit name	CVS (5V, 4V)		CVS (5V, 3V)		2-way (2 Levels)	
	Power Red.	CPU time	Power Red.	CPU time	Power Red.	CPU time
c432	0%	0.010	0.11%	0.010	4.18%	0.020
c499	0%	0.010	0%	0.010	8.97%	0.080
c880	16.25%	0.100	17.08%	0.070	14.25%	0.100
c1355	0%	0.020	0%	0.030	8.20%	0.100
c1908	7.15%	0.300	6.53%	0.160	17.36%	0.410
c2670	9.14%	0.900	18.58%	0.880	21.36%	1.590
c3540	3.54%	0.490	5.67%	0.460	16.23%	1.960
c5315	19.78%	5.660	29.66%	4.820	21.72%	5.620
c6288	0.62%	0.460	1.69%	0.440	8.63%	1.970
c7552	15.21%	11.54	10.57%	5.860	15.59%	10.45

TABLE III
EXPERIMENTAL RESULTS OF CFMV ALGORITHM WITH TWO, THREE, AND FOUR VOLTAGE LEVELS

circuit name	P-type (2 Levels)		P-type (3 Levels)		P-type (4 Levels)	
	Power Red.	CPU time	Power Red.	CPU time	Power Red.	CPU time
c432	2.62%	0.030	3.25%	0.020	3.25%	0.020
c499	5.62%	0.070	8.98%	0.160	11.31%	0.210
c880	8.93%	0.100	14.11%	0.380	17.93%	1.090
c1355	5.14%	0.110	8.58%	0.310	11.01%	0.520
c1908	11.52%	0.440	16.05%	1.130	16.87%	1.370
c2670	13.39%	1.590	20.03%	6.350	22.94%	14.90
c3540	11.27%	2.110	16.94%	7.410	19.46%	14.32
c5315	15.19%	6.550	21.74%	24.25	26.09%	58.03
c6288	6.74%	2.650	7.58%	12.86	7.65%	26.46
c7552	9.95%	11.16	13.41%	43.57	13.72%	64.09

circuit name	N-type (2 Levels)		N-type (3 Levels)		N-type (4 Levels)	
	Power Red.	CPU time	Power Red.	CPU time	Power Red.	CPU time
c432	1.78%	0.010	2.23%	0.020	2.23%	0.020
c499	3.83%	0.070	6.23%	0.150	8.01%	0.210
c880	6.08%	0.100	9.79%	0.350	12.69%	1.080
c1355	3.50%	0.100	5.96%	0.290	7.81%	0.500
c1908	8.11%	0.440	11.60%	1.060	12.83%	1.550
c2670	9.11%	1.620	13.87%	6.050	16.43%	14.17
c3540	7.86%	2.150	12.63%	7.930	14.79%	14.73
c5315	11.53%	7.010	16.35%	24.82	19.73%	62.84
c6288	5.33%	3.000	6.48%	26.34	6.68%	101.41
c7552	6.77%	10.68	9.38%	44.28	10.24%	82.72

circuit name	2-way (2 Levels)		2-way (3 Levels)		2-way (4 Levels)	
	Power Red.	CPU time	Power Red.	CPU time	Power Red.	CPU time
c432	4.18%	0.020	5.07%	0.020	5.07%	0.020
c499	8.97%	0.080	13.15%	0.150	13.51%	0.200
c880	14.25%	0.100	21.39%	0.370	22.25%	0.850
c1355	8.20%	0.100	13.10%	0.290	13.10%	0.380
c1908	17.36%	0.410	20.56%	0.730	20.69%	0.760
c2670	21.36%	1.590	27.56%	5.290	29.46%	12.61
c3540	16.23%	1.960	23.09%	6.480	24.04%	10.11
c5315	21.72%	5.620	30.91%	21.35	33.39%	48.47
c6288	8.63%	1.970	9.11%	4.440	9.11%	5.330
c7552	15.59%	10.45	16.53%	19.13	16.63%	21.85

V. CONCLUSION AND FUTURE WORK

Voltage scaling with multiple supply voltages is a very challenging problem since the size of its solution space is $O(l^n)$, where l is the

number of supply voltages and n is the number of gates. In this paper, we have proposed a multiple-voltage scaling technique to minimize the power consumption of a combinational circuit.

We put constraints on the voltage differences between connected gates to eliminate the necessity of level converters, which were used in previous works to prevent static currents. With such constraints, we formulated the problem and found that the size of its solution space is $\Omega(2^{n/(p+1)})$, where p is the depth of a graph. Though the solution space of the formulated problem is much smaller than that of the generic multiple-voltage scaling problem, it is still exponentially proportional to the number of gates.

Therefore, we tried to find a practical solution and proposed a heuristic algorithm for the formulated problem. The complexity of our heuristic algorithm is shown to be $O(n^{l-1})$. Furthermore, we implemented the heuristic algorithm in C, performed experiments on all the ISCAS85 circuits, and compared the results with those of the CVS technique. From the experimental results, we can find that the CFMV technique can reduce the power consumption by up to 33.39%. On average, 9–18% power reduction can be obtained using the CFMV technique.

In this paper, we used the maximum voltage difference allowed in the CFMV structures. In future work, we will find what the voltage differences should be to obtain maximum power reduction.

In addition, we used monotonously increasing supply voltages in the CFMV structures to have both the converter-free and the clustering features. If the clustering constraint is released, it is not necessary for the voltage sequences to increase monotonously. In the future, we will also explore the solution of such formulation.

Last but not least, if the converter-free constraint is released, it becomes the generic problem and has the largest solution space. This is really a challenging research topic.

REFERENCES

- [1] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design—A Systems Perspective*, 2nd ed. Reading, MA: Addison-Wesley, 1992.
- [2] M. Alidina, J. Monteiro, S. Devadas, A. Ghosh, and M. Papaefthymiou, "Precomputation-based sequential logic optimization for low power," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 426–436, Dec. 1994.
- [3] L. Benini, P. Siegel, and G. De Micheli, "Saving power by synthesizing gated clocks for sequential circuits," *IEEE Design Test Comput.*, vol. 11, pp. 32–41, Winter 1994.
- [4] C.-L. Su, C.-Y. Tsui, and A. M. Despain, "Saving power in the control path of embedded processors," *IEEE Design Test Comput.*, vol. 11, pp. 24–30, Winter 1994.
- [5] R. Hossain, L. D. Wronski, and A. Albicki, "Low power design using double edge triggered flip-flops," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 261–265, June 1994.
- [6] D.-S. Chen and M. Sarrafzadeh, "An exact algorithm for low power library-specific gate resizing," in *Proc. 33rd Design Automat. Conf.*, June 1996, pp. 783–788.
- [7] O. Coudert, R. Haddad, and S. Manne, "New algorithms for gate sizing: A comparative study," in *Proc. 33rd Design Automat. Conf.*, June 1996, pp. 734–739.
- [8] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473–484, Apr. 1992.
- [9] L. S. Nielsen, C. Niessen, J. Sparsø, and K. van Berkel, "Low-power operation using self-timed circuits and adaptive scaling of the supply voltage," *IEEE Trans. VLSI Syst.*, vol. 2, pp. 391–397, Dec. 1994.
- [10] S. Raje and M. Sarrafzadeh, "Variable voltage scheduling," in *Proc. ISLPD*, Apr. 1995, pp. 9–14.
- [11] K. Usami and M. Horowitz, "Clustered voltage scaling technique for low-power design," in *Proc. ISLPD*, Apr. 1995, pp. 3–8.
- [12] J.-M. Chang and M. Pedram, "Energy minimization using multiple supply voltages," in *Proc. ISLPD*, 1996, pp. 157–162.
- [13] P. Antognetti, D. D. Caviglia, and E. Profumo, "CAD model for threshold and subthreshold conduction in MOSFETs," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 454–458, June 1982.
- [14] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*. Rockville, MO: Computer Science, 1990.