A Placement Technique for Multiple-Voltage Design

Qi Zhu
Farhana Sheikh
Philip Chong

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-133

http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-133.html

October 19, 2006

A Placement Technique for Multiple-Voltage Design

Qi Zhu
University of California, Berkeley
Berkeley, CA 94720
zhuqi@eecs.berkeley.edu

Farhana Sheikh
University of California, Berkeley
Berkeley, CA 94720
farhana@eecs.berkeley.edu

Philip Chong
Cadence Berkeley Labs
Berkeley, CA 94704
pchong@cadence.com

Abstract

A voltage-island architecture for systems-on-chip is an effective way to reduce active and static power. For such multiple supply designs, various layout architectures exist; however, placement algorithms that take advantage of a circuit rows style of implementation are not available to designers today. This paper presents two algorithms to place standard cells in a circuit rows style of implementation for dual-supply digital designs using double-height level converting flip-flops. Our results show significant improvement in terms of wirelength over a simple bi-partitioning scheme that is currently employed in manufactured designs. On average, we show a 21% wiring overhead for multiple-supply design using our new techniques compared to an 81% overhead under the bi-partitioning scheme. This paper represents a first work quantifying the physical design overhead of a dual-supply system in the context of multiple-supply aware placement algorithms.

1 Introduction

Today, power dissipation is a limiting factor in both high-performance and mobile applications. Under this power-limited scaling regime, designers have engineered novel solutions to maintain performance while reducing power dissipation. One such solution entails the use of multiple supply voltages on a single ASIC leading to voltage-island system architectures [1]. Authors of [1] show that the voltage island concept allows for supply voltage and threshold voltage optimization of each functional block independently, thereby exploiting the unique power and performance characteristics of each block.

Numerous publications outline power savings of up to 45% resulting from the use of multiple supply voltages either at the block level or at the gate level [2, 3, 4, 5, 6]. However, multiple supply voltage design necessitates changes at the physical design level. In [7] and [8], two approaches to layout of dual-supply circuits are discussed in the context of a semi-automated cell-based design flow. Each method requires changes to partitioning, placement, and power routing algorithms, and in [7], the standard cell library must be re-designed to reflect the new double-rail cell architecture style. Additionally, level converting functions must be included in the standard cell library as either stand-alone cells or incorporated into other elements [9, 10].

Outside of experimental circuits [3, 6], there are no known manufactured designs that use multiple supplies at the gate-level. However, the use of multiple supply voltages at the block level is quite common and the majority of the designs use a voltage island style of layout where each block is routed and placed in its own domain using standard or custom place and route tools. An example is seen in Intel's most recent integer execution unit design presented in [11] where a 64-bit adder is split into two supply domains with the upper 32-bit section operating at one supply and the lower 32-bit section operating at another supply. This is an example of the commonly found *bi-partition scheme* used for dual-supply circuits where each supply domain is placed and routed independently. In the case of Intel's adder, the circuit structure lends itself well to a bi-partition style of layout.

It is advantageous to use a bi-partition layout scheme because existing place and route tools for automated physical design can be used without significant modification to their underlying placement algorithms. However, in general ASIC layout where circuit structure does not generally lend itself well to bi-partitioning, using such a layout style may result in significant routing overhead. As a result, placement algorithms must be modified to be "multiple-supply aware".

In this paper we present two placement algorithms for dual-supply voltage island architectures employing a circuit rows style of ASIC layout. Both algorithms leverage a GORDIAN-style placer described in [12] as a starting point. The first algorithm is a sequential heuristic that tries to perturb the initial placement solution as little as possible and the second is based on an Integer Linear Programming (ILP) formulation with the same objective. The results of our implementation are compared with the Generic Voltage Island style of layout using a double-rail standard cell library as described in [7].

As far as the authors are aware, no work has been published on placement algorithms specifically targeted towards multiple supply design. This was thus the motivation for developing novel algorithms to tackle this problem, as the additional placement constraints imposed by dual-supply designs lie beyond the scope of established placement techniques, e.g. [13]. This work also represents a first attempt at quantifying the overhead of dual-supply systems.

Section 2 provides a brief review of circuit design using multiple supply voltages. Section 3 outlines various layout styles that have been used in either experimental or manufactured dual-supply circuits and motivates our work. Section 5 presents details of our algorithms and Section 6 discusses our results on some ISCAS89 benchmarks. We summarize and conclude our paper in Section 7.

2 Multiple Supply Circuit Design

Power consumption in CMOS circuits is dominated by dynamic switching power which decreases quadratically as supply voltage is lowered:

$$P = \alpha \cdot C_{load} \cdot f_{clk} \cdot V_{DD}^2$$

Here, α, C_{load}, and f_{clk} denote the switching activity, load capacitance, and clock frequency, respectively [3]. However, lowering V_{DD} also increases the individual gate delays by:

$$t_d \sim \frac{C_{load} \cdot V_{DD}}{2} \left[\frac{1}{K_n \cdot (V_{DD} - V_{T_n})^2} + \frac{1}{K_p \cdot (V_{DD} - |V_{T_p}|)^2} \right]$$

where all technology and gate topology parameters are lumped into constants K_n and K_p. Additionally, V_{T_n} and V_{T_p} denote the transistor threshold voltages [14].

The increased delay results in a performance degradation only if the supply voltage is reduced for gates on the critical path. The performance loss can be recovered by either adjusting the threshold voltage V_T of these gates [4], using parallel or pipelined architectures, or by reducing the supply voltage only of gates off the critical path [2]. A reduction in threshold voltage results in increased standby leakage and thus larger static power consumption. The use of parallel or pipelined architectures causes large area penalties.

In a multi-V_{DD} approach, gates off the critical path are allowed to operate at low V_{DD} (V_{DDL}) and gates on the critical path operate at high V_{DD} (V_{DDH}). This methodology allows a significant power reduction without compromising the performance of the circuit. The selective adjustment of V_{DD} can be made at either the block level or at the gate level.

When using dual V_{DD}, special attention must be given to boundaries of gates with different supply voltages. V_{DDH} gates can safely drive V_{DDL} gates. However, V_{DDL} gates cannot directly feed V_{DDH} gates without using level converters. As illustrated in Figure 1, the pull-up device of a high-voltage gate that is driven by a low-voltage gate will not completely shut off for a logical-1 input causing static current to flow.

Level converters, which are similar to sense amplifiers used in memories, must be inserted at supply boundaries from V_{DDL} to V_{DDH}. This circuitry consumes additional power and area, and contributes to delay. However, the overhead associated with level converters can be minimized by cleverly incorporating the level conversion function into either a flip-flop [9] or another gate [10]. If the functionality is combined with a synchronous element such as a flip-flop then the level converter is known as a synchronous level converter and can only be placed at synchronous boundaries. If the level conversion function is stand-alone or combined with a combinational gate it is known as an asynchronous level converter and can be placed anywhere in the design. In our paper we choose to employ synchronous level converters as their use tends to result in minimal overhead and minimal impact on robustness.

The layout of synchronous level converters can result in significant area penalty, however, a double-height cell architecture can mitigate the overhead [9]. Unfortunately, the use of double-height level converting flip-flops (LCFF or LC) with single-height combinational elements results in a difficult placement problem as detailed in Section 3.

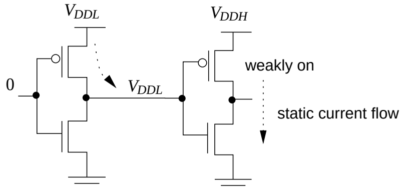


Figure 1: Static leakage current for direct connections of V_{DDL} to V_{DDH} gates [5].

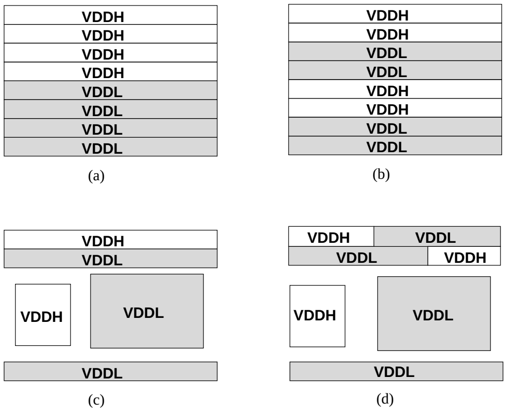


Figure 2: Dual supply layout styles: (a) Bi-partition (b) Circuit Rows (c) Voltage Islands (d) Generic Voltage Islands

3 Physical Design for Multiple-Supply Systems

In contrast to single-supply CMOS circuits, multiple-supply circuits are faced with more complications during the layout phase of their design. This is primarily due to additional overhead associated with use of multiple supplies on chip. For illustration purposes, we focus on dual-supply systems and discuss physical design issues relating to bulk CMOS dual-supply designs.

There are four different types of layout architectures that are available to ASIC designers as shown in Figure 2. The first is the bi-partition style discussed in the Introduction. The second is known as *Circuit Rows* where rows of cells operating at different supplies are interleaved. An experimental multiple supply media processor was developed at Toshiba [3] that used this style of layout. The *Voltage Island* style of ASIC architecture incorporates the Circuit Rows style with blocks that are operating at different supplies isolated from the rest of the cells in the design. As shown in [1], this style lends itself well to hierarchical multiple supply design and optimization. The last layout architecture is discussed in [15] and is known as *Generic Voltage Islands*. In this case cells and blocks operating at different supply voltages can be interleaved on the same row in the ASIC.

If conventional layout styles for row-based design are used for dual-supply circuits we are faced with situations shown in Figure 3. In Figure 3(a), required well separation for CMOS gates operating at different supply voltages prevents cell abutment resulting in poor layout density. However, if the N-Well is shared between cells operating at different supply voltages, then better

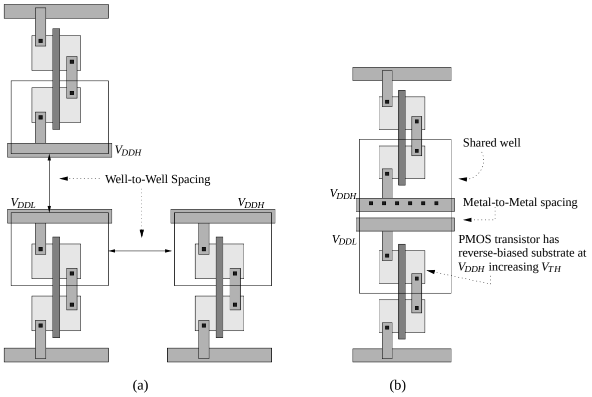


Figure 3: (a) Horizontal and vertical well-isolation issues with dual-V_{DD} circuits (b) Low supply PMOS circuits reverse-biased

area utilization is achieved at the expense of speed as illustrated in the following example.

In Figure 3(b), two inverters are shown that operate at different voltages but share the same well. In this situation, the power rails cannot be shared across both cells since they operate at different supplies. They must be separated according to metal spacing rules resulting in some area loss as compared to single supply layouts. More importantly, the PMOS transistor is now reverse-biased as the well is tied to high supply resulting in a lower effective threshold voltage. This slows down the PMOS transistor causing a performance degradation for the entire design. In [6], the authors report a speed degradation of 18% at 1.2V compared to a conventional non-reverse-biased V_{DDL} circuit.

Another complication that occurs in the physical design of multiple supply systems is due to level converters. The elements that incorporate level conversion functionality must include two supply nets in addition to the standard ground supply. This complicates both power routing algorithms and power grid generation. Both V_{DDH} and V_{DDL} supplies must now be routed to a single cell. The placement is additionally complicated if a double-height cell architecture, such as the one described in [9], is employed. The double-height cells can only be placed where two rows of different supplies are adjacent, adding new constraints to placement.

Our work focuses on placement algorithms that can adapt to a Circuit Rows layout architecture and use of double-height LCFF cells. The reasons for taking this direction are two-fold: (1) Circuit Rows is a key feature in three of the layout styles presented in Figure 2, and (2) double-height LCFF cells have minimal area overhead over other LCFF cell architectures. This is a deliberate design tradeoff on our part. We are willing to incur extra constraints during placement by adopting the Circuit Rows placement style, than, say, using a Generic Voltage Island style, in order to avoid the area and/or performance penalties the latter would incur.

4 Problem Statement

Formally, our placement problem is stated as follows. We are given following inputs.

- A list of cells in the circuit that need to be placed: $C = \{c_1, c_2, \cdots, c_Q\}$.

- A list of double-height LCFF cells: $LC = \{lc_1, lc_2, \cdots, lc_R\},\ LC \subseteq C$.

- Two supply voltages, V_{DDL} and V_{DDH}, and a voltage assignment to every cell but LCFF cells: $v(c_i) = \{V_{DDL}, V_{DDH}\},\ c_i \notin LC$.

- Chip geometry, I/O pads, circuit connectivity, area of cells.

Our objective is to find a voltage assignment to circuit rows and determine the location of each cell c_i such that routing overhead (measured in terms of wirelength) is minimized. Special attention must be given to the placement of LCFF cells. These double-height cells must be put on two adjacent rows assigned different supplies.

5 Algorithms

The overview of our approach to the problem is shown in Figure 4. The first part of our placement is a GORDIAN-like analytical placer [12], which is composed of alternating global optimization and partitioning phases. The global quadratic programming (QP) solver is interleaved with a partitioner using a bi-partitioning scheme. Regions are recursively partitioned until every region has no more than k cells (where k is predetermined by the user). During each level of partition, constraints which enforce the center of gravity for each partition are added to the quadratic programming formulation.

There is no legalization step in our pseudo-GORDIAN placer. Legalization is performed after global placement. The results of this QP-based placement are used as a guide to determine voltage assignment to rows. They are also used to legalize the placement based on the row voltage assignment and supply assignment to cells. The objective is to perturb the locations of the cells as little as possible. Usually a minimal perturbation of the initial placement results in a minimal increment in wirelength. Once a feasible row assignment is found that allows all cells to fit in, cell locations are legalized with respect to both y- and x-directions, in that order, with the minimal perturbation objective in mind. Special attention is given to LCFF cells as the algorithms must ensure that there is enough space to feasibly place them across rows with different supplies. Row supply assignment and y- and x-legalization take this constraint into account. We present two algorithms for y-direction legalization (Sections 5.3 and 5.4). The x-direction legalization algorithm is described in Section 5.5.

Our work is dependent on a circuit netlist that contains gates that have already been assigned a V_{DDH} or V_{DDL} supply voltage. Unfortunately, all freely available benchmarks today do not have dual-supply circuit netlists. Thus, our work includes a simple algorithm, based on one presented in [3], that assigns supply voltages to cells in our benchmark designs.

The following subsections describe each of our algorithms for row supply assignment and legalization. We start with a brief summary of our approach to voltage assignment for cells in our netlist.

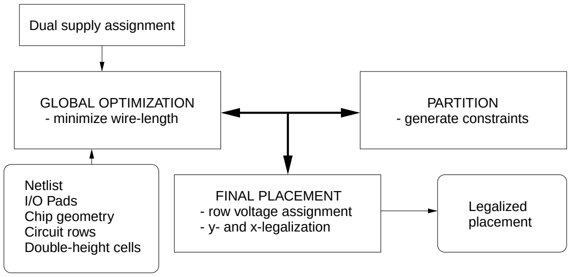


Figure 4: Algorithm overview

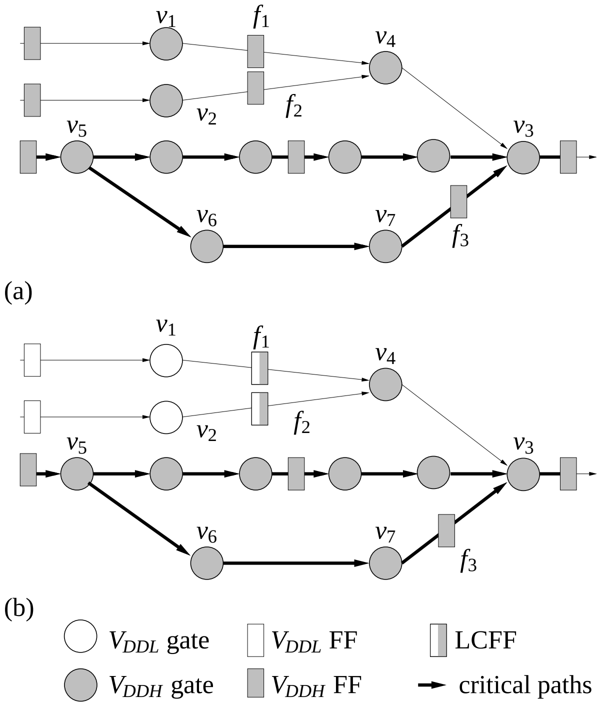


Figure 5: Example for voltage assignment: (a) original circuit with all gates at V_{DDH}, (b)voltage assignment with two gates at V_{DDL} based on available timing slack.

5.1 Supply Voltage Assignment to Cells

The voltage assignment heuristic presented in [3] was implemented here. Initially, all gates are assigned V_{DDH}. Then the algorithm works backward through the netlist from primary outputs to primary inputs. If timing slack is available to a gate, then the gate is assigned V_{DDL} as long as it does not feed directly into a gate at V_{DDH}. At synchronous boundaries where the gate prior to the flip-flop is at V_{DDL} and the gate after the register is at V_{DDH}, the flip-flop is replaced by a LCFF. An example is shown in Figure 5.

In the example, v_1, v_2, and v_4 lie off the critical path so they may be assigned V_{DDL}. However, v_4 must remain at V_{DDH} since it directly feeds into v_3 which operates at V_{DDH}. Registers f_1 and f_2 are replaced with LCFFs.

5.2 Supply Voltage Assignment to Rows

The first stage of the row supply voltage assignment algorithm calculates the number of rows available based on cell height. After this, there are two schemes available to the user: (a) use all the rows in the chip to place cells, or (b) use as few rows as possible by incrementally adding rows until we obtain a feasible placement. Choosing the latter scheme leads to a more dense design but may result in a larger perturbation of the initial QP placement result. Depending on the benchmark, scheme (a) may be be better than (b) or vice-versa. Our algorithm has an option to choose either or the one that results in the least perturbation. If the algorithm cannot accommodate all the cells within the chip dimensions, it adds a minimal set of rows that allow a feasible placement. Pseudo-code for the core of the row assignment algorithm is given in Figure 6.

The number of high and low supply rows are calculated based on the total distribution of V_{DDL} and V_{DDH} cells after the QP placement. In a bi-partition scheme, the assignment of supply voltage to rows is simple: the top part of the design is designated V_{DDH} (or V_{DDL}) and the bottom part V_{DDL} (or V_{DDH}). There are only two rows where LCFFs can be placed, so if the total row width does not accommodate all the LCFFs, then the row width is adjusted to place all LCFFs followed by an adjustment of the total number of high and low supply rows based on the new row width.

In the Circuit Rows style placement, rows are assigned V_{DDH} based on rows with the maximum area sum of V_{DDH} cells. Once all V_{DDH} cells are accommodated (based on area), the remainder of the rows are assigned V_{DDL}. Then, the assignment is checked to ensure a feasible placement. If the assignment cannot accommodate all the LCFF cells, supply assignment to rows is adjusted by an exchange mechanism until all LCFFs can be feasibly placed within the chip dimensions: V_{DDH} rows with the highest V_{DDL} area are assigned to V_{DDL} and V_{DDL} rows with the highest V_{DDH} area are assigned to V_{DDH}. Subsequently, if the assignment is still infeasible, the number of V_{DDH} and V_{DDL} rows are adjusted until a feasible assignment is found.

5.3 Sequential Row Legalization

This section describes the first algorithm used for y-direction legalization. It is a sequential heuristic that we term *Sequential Row Legalization*. Figure 7 shows the algorithm pseudo-code. The rows are divided into four categories: pure high (low) rows, which only can accommodate V_{DDH} (V_{DDL}) cells; mixed high (low) rows, which can accommodate V_{DDH} (V_{DDL}) cells and LCFF cells. We sort V_{DDH} cells, V_{DDL} cells and LCFF cells, respectively, by their y-coordinates, and then sequentially place these cells to proper rows.

There are two schemes to place LCFF cells. The first starts placing LCFF cells from a specified position in each row; the second starts placing LCFF cells after all V_{DDH} and V_{DDL} cells have been placed. The first scheme places the LCFF cells evenly across the chip which leads to a smaller perturbation of the initial QP placement. However, sometimes this leads to more wasted area and thus not all cells can be accommodated within the chip boundary. There is an option in our implementation for choosing between these two schemes. Another option is to initially use the

```
// Assign supply voltages to rows
Algorithm AssignRowSupply(cellView, option) {
   // Calculate number of VDDH and VDDL rows
   $VDDH\_area = VDDH\_cell\_total\_area + LC\_cell\_total\_area/2$;
   $VDDL\_area = VDDL\_cell\_total\_area + LC\_cell\_total\_area/2$;
   $VDDH\_rows = total\_rows * VDDH\_area/total\_area$;
   $VDDL\_rows = total\_rows * VDDL\_area/total\_area$;
   if area for VDDH and VDDL cells is not sufficient then
      adjust number of VDDH and VDDL rows;
   // Compute VDDH cell area for each row
   for each cell at VDDH do {
      find the row it belongs to after the pseudo-GORDIAN,
      add cell area to total VDDH cell area for that row;
   }
   // Assign supply voltages to rows
   if option == bi-partition then {
      // Can only fit LCs in two rows
      if $LC\_area > row\_width * 2 * std\_cell\_height$ then {
         increase width of row to accommodate LCs;
         adjust number of VDDH and VDDL rows;
      }
      calculate which portion(top or bottom) has more VDDH
      cell area;
      assign top/bottom(depends on the above calculation)
      rows to VDDH until all VDDH cells fit, assign
      bottom/top rows to VDDL until all VDDL cells fit;
   } else {
      // Row assignment for circuit rows
      sort rows by decreasing VDDH cell area;
      assign row i($i \leq VDDH\_rows$) to VDDH;
      assign row i($VDDH\_rows + 1 \leq i \leq total\_rows$) to VDDL;
      find rows for LCs by finding high-low boundaries;
      if there is enough space for LCs then return;
      // Adjust voltage assignments to accommodate LCs
      while not enough space for LCs do {
         assign VDDH row with highest VDDL area to VDDL;
         assign VDDL row with highest VDDH area to VDDH;
         ensure that exchange accommodates more LCs;
      }
      if assignment is still infeasible then {
          adjust number of VDDH and VDDL rows;
          redo row assignment;
      }
   }
```

Figure 6: Pseudo-code for row voltage assignment

first scheme; if the cells cannot be placed feasibly within the chip boundary, then the second scheme is employed.

Furthermore, a windowing option is used to improve the result of sequential legalization. Within each small window, a locally optimal solution is found by using an ILP-based legalization formulation which is explained in Section 5.4. The size of the window can be configured in our implementation.

After windowing, we apply the x-direction legalization; that is, legalize the placement within rows, which is explained in Section 5.5.

5.4 ILP-Based Row Legalization

Another method for y-direction legalization is to use an ILP formulation. We use S_H to denote the set of indices for all V_{DDH} cells; use S_L to denote the index set of V_{DDL} cells; use S_{LC} to denote the index set of $LCFF$ cells. We use R_H to denote the index set of V_{DDH} rows; use R_L to denote the index set of V_{DDL} rows; use R_{LC} to denote the index set of rows which satisfy: $R_{LC} = \{ j \mid \text{row } j+1 \text{ has different voltage supply than row } j \}$. We use $x_{i,j}$ to denote whether to place the i_{th} cell into row j, $x_{i,j} = \{0,1\}$.

```
Algorithm SequentialLegalization(cellView) {
   sort VDDH, VDDL, LC cell by increasing y-coordinate;
   if using LC start point then
       compute the start point for LC within one row;
   // Place cells into rows
   for each row $i$ do {
      if $i$ is pure VDDH row(only can place VDDH cell) {
         place VDDH cells in sorted order until
         no VDDH cells left or no more space in the row;
      }
      if $i$ is pure VDDL row(only can place VDDL cell) {
         place VDDL cells in sorted order until
         no VDDL cells left or no more space in the row;
      }
      if $i$ is mixed VDDH row(can place VDDH and LC cell) {
         if using LC start point then {
            place VDDH cells in order until no VDDH
            cells left or reached LC start point;
            place LC cells in order from LC start point
            until no LC cells left or no more space in
            the row;
         } else {
            if VDDH cells left then
               place VDDH cells in order until no VDDH
               cells left or no more space in the row;
            else
               place LC cells in order until no LC
               cells left or no more space in the row;
         }
      }
      if $i$ is mixed VDDL row(can place VDDL and LC cell) {
         similar as mixed VDDH row;
      }
   }
   windowing;
   legalization within rows;
   return;
}
```

Figure 7: Pseudo-code for sequential row legalization

The objective function for the ILP is given by:

$$\begin{aligned} \min F \quad = \quad & \sum_{i \in S_H, j \in R_H} x_{i,j} \cdot f(i,j) \\ & + \sum_{i \in S_L, j \in R_L} x_{i,j} \cdot f(i,j) \\ & + \sum_{i \in S_{LC}, j \in R_{LC}} x_{i,j} \cdot f(i,j) \end{aligned}$$

And we have the following constraints:

- The total area of all cells in row j must be less than or equal to the available area A_j of that row:

$$\begin{aligned} & H_j \left(\sum_{i \in S_H} x_{i,j} \cdot A(i) \right) + L_j \left(\sum_{i \in S_L} x_{i,j} \cdot A(i) \right) \\ & \quad + LC_j \left(\sum_{i \in S_{LC}} x_{i,j} \cdot A(i)/2 \right) \\ & \quad + LCP_j \left(\sum_{i \in S_{LC}} x_{i,j-1} \cdot A(i)/2 \right) \leq A_j; \quad \forall\ j \end{aligned}$$

where
$H_j = 1$ when $j \in R_H$, otherwise $H_j = 0$
$L_j = 1$ when $j \in R_L$, otherwise $L_j = 0$

$LC_j = 1$ when $j \in R_{LC}$, otherwise $LC_j = 0$
$LCP_j = 1$ when $j-1 \in R_{LC}$ $(j \geq 1)$, otherwise $LCP_j = 0$

Note that we divide $A(i)$ by 2 for LCFF cells because they are double-height, and will occupy two adjacent rows. This is also the reason we need to consider not only row j, but also row $j-1$.

- Each V_{DDH} cell must be assigned to a row:

$$\sum_{j \in R_H} x_{i,j} = 1; \;\; \forall\, V_{DDH} \text{ cell } i \in S_H$$

- Each V_{DDL} cell must be assigned to a row:

$$\sum_{j \in R_L} x_{i,j} = 1; \;\; \forall\, V_{DDL} \text{ cell } i \in S_L$$

- Each LCFF cell must be assigned to a pair of rows j and $j+1$:

$$\sum_{j \in R_{LC}} x_{i,j} = 1; \;\; \forall\, LCFF \text{ cell } i \in S_{LC}$$

Here, $A()$ is the area function of cells; A_j is the area of row j; $f()$ is a cost function representing the cost of moving cells to rows. Currently in our implementation, the cost function is computed as the difference between current position and potential new position multiplied by the area of cell, i.e., $f(i,j) = A(i) \cdot d(i,j)$ where $d(i,j)$ is the position difference. We include the area here due to the fact that cells with greater connectivity are more costly to wirelength when they are perturbed. As a heuristic measure we assume that larger cells generally have greater connectivity and hence penalize their movement more in the cost function.

The ILP solver packages GLPK [16] and IBM OSL [17] were used in our implementation. Since the runtime on large benchmarks was found to be very slow, and setting timeouts was not a scalable solution, we implemented a windowing scheme. An ILP-based legalization is performed in a small window where locally optimal solutions are found. By moving the window around the whole placement, the global result can be improved. As mentioned earlier, this windowing scheme is also used for improving the result of sequential legalization.

After the ILP legalization is completed in the y-direction, x-direction legalization is performed using the scheme explained in next section.

5.5 Legalization within Rows

During x-direction legalization, cells are sorted by increasing x-coordinate and are considered one by one. For each cell, a feasible region of placement is found. The left boundary of the feasible region is decided by the cells placed earlier, and the right boundary is decided by the area sum of cells left to be placed; that is, enough area must be available for cells which have not been placed. Within this feasible region, the cells are placed in the position which is closest to its original x-position.

Occasionally, the expansion of some heavily-overlapped cells will perturb the original positions significantly, especially for relatively sparse rows. In order to improve the results, the cells are divided into sets prior to sorting; each set includes some consecutive overlapped cells. After sorting the sets, the overlapped cells are expanded within each set. Then each cell set slides within its feasible region. In this feasible region, the positions which give the minimal sum of perturbations of the cells in the set are chosen.

Design	**Total cells**	**LCFFs**	V_{DDL} **Cells**	V_{DDH} **Cells**
s208	76	4	16	56
s298	124	12	18	94
s344	153	10	36	107
s349	160	12	20	128
s382	203	18	50	135
s1196	920	16	427	477
s1298	954	17	274	663
s1423	882	67	388	427
s1488	720	3	196	521
s1494	604	2	149	553
s13207	3108	465	1610	1033

Table 1: Voltage assignment results for the ISCAS89 benchmarks

Legalization of LCFF cells causes further issues. In our current implementation, we place them either on the right or left side of the row, depending on the comparison of perturbation. This can be improved by adopting a more sophisticated scheme which supports arbitrarily placement of LCFF cells within rows.

6 Results

This section presents results of our work. Our benchmarks are circuits in the ISCAS89 set with a mapping to a generic 0.25μm technology. The standard cell library was augmented with double-height LCFF cells.

The distribution of V_{DDL} and V_{DDH} cells after voltage assignment was run on each benchmark is shown in Table 1.

After voltage assignment, the benchmarks were run through four different placers each with a GORDIAN-style placer core and a variation on the legalization and style of layout. Table 2 shows estimated wirelengths for each different type of legalization algorithm and style. As detailed routing is not available at this stage, here the widely-used half-perimeter bounding box metric is taken as an estimate for wirelength.

The first column in Table 2 shows the best possible placement obtained by using a Generic Voltage Island implementation. It uses a standard cell library with dual rails [7] which allows a mix of $LCFF$ cells, V_{DDH} and V_{DDL} cells on a single row. The legalization used in the generic placement is based on sequential legalization in the y-direction followed by the same x-direction legalization used in the other algorithms described earlier in this paper. Practically, it would require a double-rail standard cell library. In some sense, this is an idealized placement; we accept the ability to place cells in any row, regardless of voltage assignment, but we do not include here the area and/or performance penalties such an approach would incur, as described in Section 3. Although this placement is physically unrealizable, we use it as an ideal case for comparison with the other techniques. The second column of the table shows the results of using the simple, commonly-used bipartition style of layout. The third and fourth column are both for circuit rows style of layout. The third column shows the results of the sequential row legalization heuristic. The fourth column shows the results from using the ILP-based legalization.

Figure 8 shows two sample placements, one using the sequential legalization technique and the other using the ILP-based legalization. Note that in both placements all double-height LCFF cells span both V_{DDH} and V_{DDL} standard cell rows as required.

Design	Generic	Bi-partition	Circuit Rows (Sequential)	Circuit Rows (ILP)
s208_1	2.520	2.873(*)	2.875	2.763
s298	4.361	9.427(*)	5.654(*)	5.902
s344	5.312	8.333(*)	6.207	5.883
s349	5.360	10.393(*)	7.380	6.693
s382	6.967	13.472(*)	9.585	9.280
s1196	58.232	73.294(*)	63.548	64.670
s1238	61.642	77.353(*)	64.610	68.437
s1423	45.118	214.782(*)	66.454	74.315
s1488	49.953	53.553	51.654	51.013
s1494	46.401	49.023	47.890	47.729
s13207	260.187	2918.829(*)	490.458(*)	432.767(*)

Table 2: Total estimated wirelength (mm) after each type of placement of selected ISCAS89 benchmarks; (*) indicates that placement did not fit within the original chip dimensions, so more area must be added.

Design	Generic	Bi-partition	Circuit Rows (Sequential)	Circuit Rows (ILP)
s208_1	1	1.14(*)	1.14	1.10
s298	1	2.16(*)	1.30(*)	1.35
s344	1	1.57(*)	1.17	1.11
s349	1	1.94(*)	1.38	1.25
s382	1	1.93(*)	1.38	1.33
s1196	1	1.26(*)	1.10	1.11
s1238	1	1.25(*)	1.05	1.11
s1423	1	4.76(*)	1.47	1.65
s1488	1	1.07	1.03	1.02
s1494	1	1.06	1.03	1.03
s13207	1	11.2(*)	1.89(*)	1.66(*)
Average	1	2.67	1.27	1.25
Avg. excl. s13207	1	1.81	1.21	1.21

Table 3: Comparison of placement style and legalization scheme; (*) indicates that the cells did not fit within the chip boundary

First of all, our results show that the circuit rows style of implementation has a significant improvement over bi-partition style in terms of wirelength. Both sequential legalization and ILP-based legalization have a better wirelength than bi-partition in all benchmarks. Furthermore, the bi-partition scheme often cannot fit all the cells within the chip dimensions due to the constrained placement of the LCFFs. It gives relatively good results for test cases s1488 and s1494 because these benchmarks have very few LCFFs.

The results of sequential legalization and ILP-based legalization are close in most cases. For small benchmarks, the ILP-based legalization is better, while in some larger benchmarks the sequential legalization is better. The reason is that when the size of problem increases, the complexity of ILP formulation increases and we cannot find an optimal solution within a reasonable time. So we implemented the windowing scheme to improve the results. But since the windowing is used for both sequential and ILP-based legalization, the results of sequential legalization are still better in some cases. Another option would be to employ a more powerful ILP solver, such as CPLEX [18]. This is part of our future work. In terms of run time, the sequential legalization is very fast, while the ILP-based legalization needs much more time. Currently, we set the time out limit to 500 seconds.

In terms of overall comparison, the bi-partition scheme produced results that were 1.81× worse than the Generic placement. The sequential legalization was only 1.21× worse and the ILP was also 1.21× worse than the Generic placement. We exclude the s13207 benchmark because, in this case, to accommodate the LCFF cells, the bi-partition scheme would need a significant reshape of the original chip dimension. This would make the comparison of wirelength unreasonable. Table 3 presents a comparison of all the schemes in relation to the Generic layout. Our results show that on average, there is a 21% to 81% additional wirelength overhead for multiple-supply design using circuit rows, depending on the type of legalization and layout scheme used.

7 Conclusions and Future Work

In this paper we presented a novel work on multiple-supply-aware placement for voltage islands. We devised two algorithms that use the result of a GORDIAN-style placement as a starting point for row supply assignment and legalization of placement in a circuit rows paradigm. We compared our results to common bi-partition layout, and show that both circuit rows layout with sequential legalization and circuit rows layout with ILP-based legalization are better in terms of wirelength. We also compared our two schemes and bi-partition to a Generic Voltage Island placement based on GORDIAN-style placement and sequential y- and x-direction legalization. The results show that our schemes do not increase the wirelength of Generic placement much, while the common bi-partition scheme results in highly degraded performance in terms of wirelength, especially for the benchmarks with high-percentage LCFFs.

In the future, we hope to enhance our algorithms to be more efficient in terms of run-time and include more sophistication in the QP placer by adding constraints to prevent large cell overlap in the x-direction; we also hope to include better partitioning algorithms. Alternatively, we may experiment with applying our legalization techniques on initial placements obtained through other global placement techniques besides a GORDIAN-style method. For example, modern recursive bisectioning-based placement tools such as that of [19], or non-partitioning techniques such as [20], may provide good initial solutions for us to work with, and there is certainly much room for experimentation in this domain. Other future enhancements include using a more powerful ILP solver such as CPLEX to improve scalability.

We recognize the small size of the current set of benchmarks. The prime difficulty in this area is obtaining real, openly-available designs which are either already realized in a multi-V_{DD} technology, or have timing information available so that the supply voltage assignment technique described in Section 5.1 can be applied to them. It should be noted that the standard placement benchmarks such as [21] do not provide any timing or functional information and so cannot be realistically transformed into a multi-V_{DD} design. We hope that as multiple-supply design continues to gather interest among power-aware circuit designers, larger and more suitable benchmarks will become available in future.

Finally, it should be noted that timing is not considered during the placement phase which may lead to a timing closure problem. There is a large possibility that this will happen as there is much less timing slack available after dual-supply voltage assignment. An iterative placement and supply voltage assignment loop may mitigate the timing closure problem. We hope to address this in future versions of our implementation.

8 Acknowledgments

References

[1] D. Lackey, P. S. Zuchowski, T. R. Bednar, D. W. Stout, S. W. Gould, and J. M. Cohn, “Managing power and performance for system-on-chip designs using voltage islands,” in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 195–202, 2002.

[2] K. Usami and M. Horowitz, “Clustered voltage scaling technique for low-power design,” in *Proceedings ISPLD*, pp. 3–8, April 1995.

[3] K. Usami, M. Igarashi, F. Minami, T. Ishikawa, M. Kanzawa, M. Ichida, and K. Nogami, “Automated low-power technique exploiting multiple supply voltages applied to media processor,” *IEEE Journal of Solid-State Circuits*, vol. 33, no. 3, pp. 463–472, 1998.

[4] T. Kuroda and T. Sakurai, “Low-power circuit design techniques for multimedia CMOS VLSIs,” *Electronics and Communications in Japan, Part 3*, vol. 81, no. 9, 1998.

[5] K. Usami and M. Igarashi, “Low-power design methodology and applications utilizing dual supply voltages,” in *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 123–128, January 2000.

[6] Y. Shimazaki, R. Zlatanovici, and B. Nikolić, “A shared-well dual-supply voltage 64-bit ALU,” in *Proceedings of IEEE International Solid State Circuits Conference*, (San Francisco, CA), pp. 104–105, February 2003.

[7] C. Yeh and Y.-S. Kang, “Cell-based layout techniques supporting gate-level voltage scaling for low power,” *IEEE Transactions on VLSI Systems*, vol. 9, pp. 983–986, December 2001.

[8] M. Igarashi, K. Usami, K. Nogami, F. Minami, Y. Kawasaki, T. Aoki, M. Takano, S. Sonoda, M. Ichida, and N. Hatanaka, “A low-power design method using multiple supply voltages,” in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 36–41, 1997.

[9] F. Ishihara, F. Sheikh, and B. Nikolić, “Level conversion for dual-supply systems,” *IEEE Transactions on VLSI*, vol. 12, no. 2, pp. 185–195, 2004.

[10] S. H. Kulkarni and D. Sylvester, “Fast and energy-efficient asynchronous level converters for multi-VDD design,” in *Proceedings of the IEEE International SOC Conference*, pp. 169–172, September 2003.

[11] S. Mathew, M. Anders, B. Bloechel, T. Nguyen, R. Krishnamurthy, and S. Borkar, “A 4GHz 300mW 64b integer execution ALU with dual supply voltages in 90nm CMOS,” in *Proceedings of IEEE International Solid State Circuits Conference*, (San Francisco, CA), pp. 162–163, February 2004.

[12] J. M. Kleinhans, G. Sigl, F. M. Johannes, and K. J. Antreich, “GORDIAN: VLSI placement by quadratic programming and slicing optimization,” *IEEE Transactions on Computer-Aided Design*, vol. 10, no. 3, pp. 356–365, 1991.

[13] K. Doll, F. Johannes, and K. Antreich, “Iterative placement improvement by network flow methods,” *IEEE Transactions on Computer-Aided Design*, vol. 13, pp. 1189–1200, Oct. 1994.

[14] J. M. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective, Second Edition*. New Jersey: Prentice-Hall, 2003.

[15] R. Puri, L. Stok, J. Cohn, D. Kung, D. Pan, D. Sylvester, A. Srivastava, and S. Kulkarni, “Pushing ASIC performance in a power envelope,” in *Proceedings of the 40th ACM/IEEE Design Automation Conference*, (Anaheim, CA), pp. 788–793, June 2003.

[16] `http://www.gnu.org/software/glpk/`.

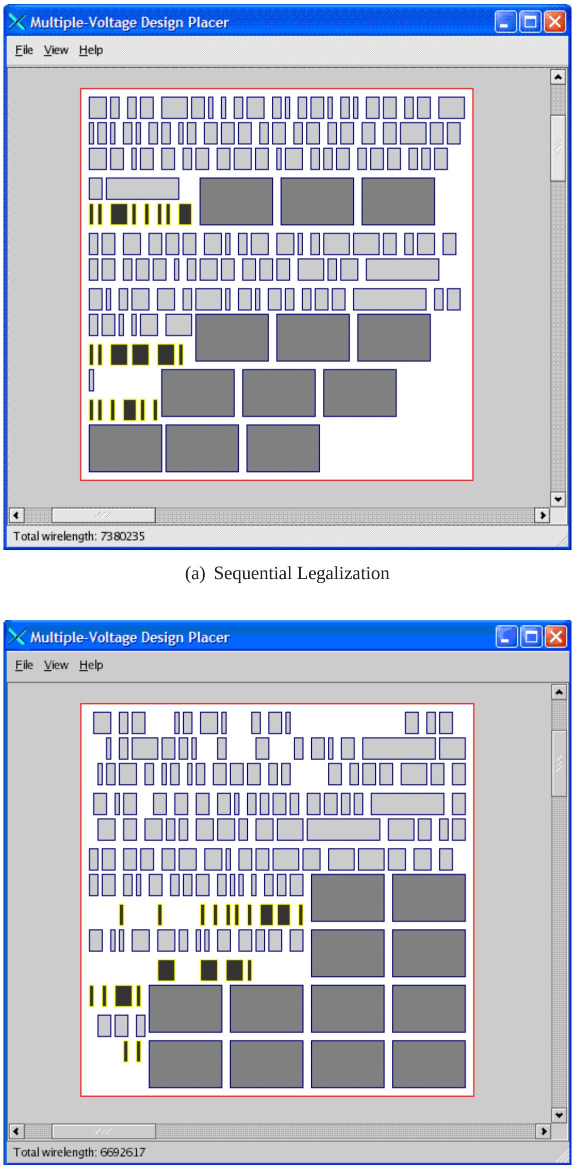
[17] `http://www-306.ibm.com/software/data/bi/osl/`.

[18] `http://www.ilog.com/products/cplex/`.

[19] A. E. Caldwell, A. B. Kahng, and I. L. Markov, “Can recursive bisection alone produce routable placements?,” in *Design Automation Conference*, pp. 477–482, 2000.

[20] B. Hu and M. Marek-Sadowska, “FAR: Fixed-points addition & relaxation based placement,” in *International Symposium on Physical Design*, pp. 161–166, 2002.

[21] C. J. Alpert, “The ISPD98 circuit benchmark suite,” in *International Symposium on Physical Design*, pp. 80–85, 1998.



Multiple-Voltage Design Placer
File
View
Help

Total wirelength: 7380235

(a) Sequential Legalization

Multiple-Voltage Design Placer
File
View
Help

Total wirelength: 6692617

(b) ILP Legalization

Figure 8: Example placements. Light gray indicates V_{DDH} cells, dark gray indicates V_{DDL} cells; double-height cells are LCFF cells.