Decoupling Capacitance Estimation, Implementation, and Verification: A Practical Approach for Deep Submicron SoCs

David Stringfellow
John Pedicone

Synopsys Professional Services

David.Stringfellow@synopsys.com
John.Pedicone@synopsys.com

ABSTRACT

The problem of dynamic variations in supply voltage and the related impact on chip performance is a major issue facing today’s DSM SoC design teams. Through careful design of the power supply network, correct chip functionality can be ensured.

In order to achieve an acceptable level of voltage fluctuation in the power supply network, a sufficient amount of decoupling capacitance must be allocated. These capacitors act as local charge reservoirs for high-frequency circuits and reduce the effects of power-supply noise on neighboring circuits.

This paper addresses this critical problem head-on by describing practical implementation approaches and verification techniques developed by Synopsys Professional Services.
Table of Contents

1.0 Introduction ......................................................................................................................... 7
2.0 Estimation ............................................................................................................................... 7
2.1 Estimation Rationale .............................................................................................................. 8
2.2 Circuit Model .......................................................................................................................... 8
2.3 Capacitance Calculations ...................................................................................................... 9
2.4 ITRS Guidelines .................................................................................................................... 11
2.5 Table 2 Scaling Assumptions ............................................................................................... 13
2.6 Table 2 Results ...................................................................................................................... 14
2.7 Comparison of design characteristics .................................................................................. 15
2.8 Technical Summary .............................................................................................................. 16
3.0 Implementation ..................................................................................................................... 16
3.1 Approach 1: Structured Relative Placement ...................................................................... 17
3.2 Approach 2: Uniform Distribution ...................................................................................... 18
3.3 Comparison of approaches ................................................................................................. 19
3.3.1 Structured Relative Placement Advantages/Disadvantages ............................................ 19
3.3.2 Uniform Distribution Advantages/Disadvantages .......................................................... 20
3.4 Preferred Flow ..................................................................................................................... 20
3.5 Phased Approach ................................................................................................................ 21
3.5.1 Early Estimation and Planning ......................................................................................... 21
3.5.2 Implementation and Refinement ..................................................................................... 21
3.5.3 Signoff ............................................................................................................................ 21
3.6 Plan at the top-level .............................................................................................................. 22
3.6.1 Insert decoupling capacitors at the block-level .............................................................. 22
3.6.2 Insert decoupling capacitors at the top-level ................................................................. 22
3.7 Implementation details ....................................................................................................... 22
3.7.1 Physical Area Requirements ......................................................................................... 23
3.7.2 Incremental Refinement ................................................................................................. 24
3.7.3 Jupiter-XT Scripts ........................................................................................................... 24
3.7.4 Implementation Results ................................................................................................. 25
4.0 Analysis and Verification ..................................................................................................... 25
4.1 PrimeRail Usage Model ...................................................................................................... 26
4.2 Standard Cell Characterization .......................................................................................... 26
4.3 DWM-Lite Model Characterization .................................................................................... 27
4.4 Characterization summary .................................................................................................. 27
4.5 Analysis, debug, and design fixing ..................................................................................... 27
4.5.1 Test block verification results ......................................................................................... 27
4.5.2 Block “R” analysis results ............................................................................................... 30
4.5.3 Block “O” analysis results .............................................................................................. 33
4.6 Analysis and verification summary ..................................................................................... 36
4.7 What-if Analysis ................................................................................................................ 36
4.8 Script Summary ................................................................................................................... 37
5.0 Conclusions and Recommendations .................................................................................. 37
6.0 Acknowledgements ............................................................................................................ 38
**Table of Figures**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Switching model of global decoupling capacitor with power grid</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Decoupling Cap Circuit Model</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>Decoupling capacitance density vs. process node</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>Comparison of sample design’s power density</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>Customer-specific conceptual dcap implementation</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>Uniform distribution conceptual dcap implementation</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>Decoupling cap placement result for sample block (7.5% area utilization)</td>
<td>25</td>
</tr>
<tr>
<td>8</td>
<td>Dynamic IR drop near adjacent clock buffers</td>
<td>28</td>
</tr>
<tr>
<td>9</td>
<td>Dynamic IR drop near adjacent clock buffers (zoom)</td>
<td>29</td>
</tr>
<tr>
<td>10</td>
<td>Dynamic rail &quot;bounce&quot; on VDD near clock buffers</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>Block &quot;R&quot; localized VDD IR drop hot spot on metal1</td>
<td>31</td>
</tr>
<tr>
<td>12</td>
<td>Hard placement blockage causes VDD IR drop issues on metal1 in block “R”</td>
<td>32</td>
</tr>
<tr>
<td>13</td>
<td>Block &quot;O&quot; dynamic IR drop plot for VDD</td>
<td>33</td>
</tr>
<tr>
<td>14</td>
<td>Uniform dcap distribution in block &quot;O&quot;</td>
<td>34</td>
</tr>
<tr>
<td>15</td>
<td>Clumping of isolation buffers in block &quot;O&quot;</td>
<td>35</td>
</tr>
<tr>
<td>16</td>
<td>Flylines show two back-to-back isolation buffers in block &quot;O&quot;</td>
<td>36</td>
</tr>
</tbody>
</table>
Table of Tables

Table 1 – Technology parameters for Cu Interconnects (ITRS 2001)................................. 11
Table 2 – Sample Design Decoupling Capacitance Estimates based on ITRS 2001 benchmark designs .............................................................................................................................................. 12
Table 3 – Comparisons with ITRS 2001 guidelines ................................................................. 15
Table 4 – Capacitance and leakage power for two single-DCAP implementations ............. 24
Table 5 - Sample design characteristics ................................................................................... 26
1.0 Introduction

We describe a practical approach to the estimation, implementation, and verification of decoupling capacitors for deep submicron SoCs. The case cited uses a popular 90nm technology node and foundry. The objective of the practical method described here is to produce an accurate decoupling capacitor allocation with minimal related area overhead. We have found that the best results are obtained if the capacitors are initially placed at the floorplanning stage.

In Section 2.0, we describe our assumptions and estimation rationale, including the circuit model used. A sample decoupling capacitance calculation is provided. Industry trends in power density and design metrics are discussed.

In Section 3.0, we discuss two possible implementation approaches. A preferred approach is selected and described in detail including a sample make file target and script. Here, we proposed a simple two-step implementation methodology. The first step occurs at the floorplanning stage and the second step occurs after the clock tree is placed and routed.

Section 4.0 provides an overview of our dynamic rail analysis and verification flow, which is based on PrimeRail. To speed the analysis, we abstract the hard macros using Dynamic Whitebox Models, or DWMs. The model characterization process for both standard cells and hard macros is discussed.

We conclude with some lessons learned, best practices, and recommendations in Section 5.0.

2.0 Estimation

Estimates for an example SoC are based on available published data on power density and decoupling capacitance allocations for a range of high-performance and handheld/mobile designs. Based on a simple circuit model, rationale for selecting the amount of decoupling capacitance appropriate for early physical design planning is presented.

For our example design, estimates show that between 3% and 8% of the core physical area is required for the decoupling capacitors. This is referred to later as the “dcap density”. The exact value for dcap density varies by major block according to each block’s dynamic power density, as dynamic power density is proportional to operating frequency $f$.

For worst case planning purposes, it is assumed that the required decoupling capacitance comes entirely from dedicated Cox (gate oxide capacitance) sources. At the block level, a slight refinement is introduced to improve the estimations: at least 20% of the decoupling capacitance comes from non-dedicated Cox sources, as described in [23]. These non-dedicated sources are described in detail in Section 2.5.

Designs using DSM technologies now require the power grid voltage drop to be much less than 10% of $V_{dd}$. To achieve this goal, decoupling capacitors are added to minimize switching noise. Designs using these technologies have been shown in the literature to require a decoupling capacitor area allocation of between 5% and 12%. One design, the IA-64 is on high end of the
range and the P6C (Celeron) is on low end of the range. For high-performance circuits with a clock frequency of 400 MHz or higher, it has been reported that a minimum of 10% of the die area was needed for on-chip decoupling capacitors, and the implementation was done during the early floorplanning stages [1][11].

2.1 Estimation Rationale

There is some very good literature on decoupling cap estimation and allocation for custom and high-performance designs. The estimations for our example design are primarily based on case studies for these designs and some theory provided in [1], [2], and [4]. The rationale and methods used to produce the estimates are covered in the next sections.

2.2 Circuit Model

Initial decoupling capacitance estimates for our example design are based on the simple first-order model shown in Figure 1. Here, the localized blocks have their logic gates lumped into a simple switch model having a series resistance (Rsw). The switch can also be modeled with a large inverter. This localized model is then extended for use in the initial global decoupling capacitance calculations. The series RC circuit on the left of Figure 1 models a composite capacitor whose key parameters are Rdcap and Cdcap [1]. Great care was also taken to minimize L in the design’s power grid, as we did not attempt to model on-chip inductance. The objective was to determine the optimal amount of on-chip dcap and to simplify the analysis and signoff.

The general idea for the model is that the charge held by Cdcap is used to ensure voltage stability during the high-speed switching events that charge and discharge Csw. A simple model for charge and discharge duration of 10% of the clock cycle is used. Eventually, as the charge recovery occurs during the remainder of the clock cycle, the average current is provided by the main voltage source via the long current loop and through the inductors LVDD and LVSS.

Figure 1 – Switching model of global decoupling capacitor with power grid
2.3 Capacitance Calculations

If decoupling capacitors are placed efficiently and the series inductance to them is kept low, the voltage fluctuations will be minimized. An upper bound on the transient voltage fluctuation can be calculated by modeling the power lines behind the capacitor as an infinitely large inductor (labeled as $L_{VDD}$ and $L_{VSS}$ in Figure 1). Immediately after switching, based on the model shown in Figure 1, no current flows through this inductor, and a capacitance divider is established. The total charge in both capacitors must satisfy the charge conservation law [1]:

\[
C_{d\text{cap}} V_{dd} = (V_{dd} + \Delta V) (C_{d\text{cap}} + C_{sw})
\]

(2.1)

\[
\Delta V = -\frac{C_{sw}}{C_{d\text{cap}} + C_{sw}} V_{dd}
\]

(2.2)

where $C_{d\text{cap}}$ and $C_{sw}$ are the total decoupling capacitance and total signal capacitance, respectively, for a given design. To ensure a small $\Delta V$, $C_{d\text{cap}} \gg C_{sw}$.

For example, a design having 200 nF of on-chip decoupling cap can achieve a $\Delta V$ of less than 10% of $V_{dd}$ as long as the worst case on-chip signal capacitance switched during a clock event is less than 20 nF.

Based on the simple models in Figure 1 and Figure 2, along with key design characteristics, the following can be used to estimate the amount of decoupling capacitance required to maintain a limited voltage drop in the power and ground meshes [2]:

\[
P = \gamma (C_{sw} V_{dd}^2 f)
\]

(2.3)

where $P$ is the total dynamic power consumption, $\gamma$ is the probability that a 0-1 signal transition occurs, $V_{dd}$ is the supply voltage, $f$ is the clock frequency, and $C_{sw}$ is the total signal capacitance.
in the design. For example, a signal toggling with a 50% duty cycle (not toggle rate) has \( \gamma = 0.5 \), and this leads to the familiar equation:

\[
P = \frac{1}{2} C_{sw} V_{dd}^2 f
\]  

(2.4)

Equation 2.3 implies that all of the available signal capacitance \( C_{sw} \) is switching. But this is only for the worst case condition and very rarely occurs. To compensate for this, we can modify Equation 2.3 by defining the “effective signal capacitance, \( C_{eff} \), as:

\[
C_{eff} = \alpha \cdot C_{sw}
\]  

(2.5)

where \( \alpha \) is the average toggle rate for all signal nets. This is later referred to as the “activity factor”. Note that \( \alpha \) is normally expressed in units of “toggles per ns” when \( f \) is in units of MHz.

By “effective” signal capacitance, we mean the maximum amount of signal capacitance that switches during a single clock cycle. It must be noted that in all cases \( C_{eff} \leq C_{sw} \). It follows that a signal that never switches has \( C_{eff} = 0 \), even though the corresponding \( C_{sw} \) may be non-zero. Thus, after replacing \( C_{sw} \) with \( C_{eff} \), Equation 2.4 becomes:

\[
P = \frac{1}{2} \sum_{i=0}^{n-1} C_i V_{dd}^2 f = \frac{1}{2} C_{eff} V_{dd}^2 f = \frac{1}{2} \alpha C_{sw} V_{dd}^2 f
\]  

(2.6)

Chang and Oscilowski documented heuristic methods for estimating semiconductor decoupling capacitance requirements in [4]. The research applied to packaging and circuit board design for high performance devices but can also be applied to our SoC design problem without loss of generality. In “Heuristic Equations for Semiconductor and Packaging Technology”, the authors deduced the amount of decoupling capacitance needed to limit the voltage swing to less than 10% of \( V_{dd} \) as

\[
C_{dcap} = \frac{9P}{fV_{dd}^2}
\]  

(2.7)

Presumably, this estimation is done by assuming \( \Delta V = 0.1V_{dd} \) in Equation 2.2 and then solving for \( C_{sw} \). This result is then used in Equation 2.3 to solve for \( C_{dcap} \). The heuristic method apparently used a value for \( \gamma \) of approximately 0.5. Independent of the derivation, we used Equation 2.7 to verify the results from the deterministic Equation 2.6.

The theory covered in Equations 2.1 through 2.6 has worked well in practice, but the total dynamic power consumption is somewhat difficult to determine at early design stages, as few design details are known. A more practical approach, covered in the next section, will allow us to characterize the design for power based on some published results for similar designs. We can make some initial estimates by “reverse engineering” some existing designs and then we can use the equations above to verify and adjust the results if necessary.
Note: Important figures of merit here are $\alpha$, $C_{on}$, and $C_{eff}$. In our design project, we tracked these as design metrics that we continually compared against our power consumption metric $P$.

2.4 ITRS Guidelines

The ITRS, or International Technology Roadmap for Semiconductors is used as a key reference for developing the decoupling capacitance requirements for our sample design. Table 1 below, from ITRS-2001, describes a representative benchmark chip as it progresses down the device scaling path. The table also highlights key technology parameters for Cu interconnects. This benchmark design, most likely a high performance processor or CPU, was used as the basis for estimating the decoupling capacitance requirements for our sample design. Selected parameters are listed below, and the assumption here is that all parameters are for the nominal case, e.g. $V_{dd}$ for the 0.1 $\mu$m node is set at 1.2 V. The complete table can be found in [2].

<table>
<thead>
<tr>
<th>Process Node (um)</th>
<th>0.18</th>
<th>0.13</th>
<th>0.1</th>
<th>0.07</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jo (A/cm²)</td>
<td>5.8e5</td>
<td>9.6e5</td>
<td>1.4e6</td>
<td>2.1e6</td>
</tr>
<tr>
<td>Chip size (mm²)</td>
<td>450</td>
<td>450</td>
<td>622</td>
<td>713</td>
</tr>
<tr>
<td>Vdd (V)</td>
<td>1.8</td>
<td>1.5</td>
<td>1.2</td>
<td>0.9</td>
</tr>
<tr>
<td>Freq (MHz)</td>
<td>1000</td>
<td>1700</td>
<td>3000</td>
<td>5000</td>
</tr>
<tr>
<td>P (W)</td>
<td>90</td>
<td>130</td>
<td>160</td>
<td>170</td>
</tr>
<tr>
<td>$P_{den}$ (W/cm²)</td>
<td>20</td>
<td>29</td>
<td>26</td>
<td>24</td>
</tr>
<tr>
<td>On-Chip $C_{dcap}$ (nF)</td>
<td>250</td>
<td>305</td>
<td>333</td>
<td>377</td>
</tr>
<tr>
<td>Tmax (°C)</td>
<td>120</td>
<td>140</td>
<td>150</td>
<td>175</td>
</tr>
<tr>
<td># P/G Pads</td>
<td>1536</td>
<td>2018</td>
<td>2018</td>
<td>2560</td>
</tr>
<tr>
<td>R-local (kΩ/m)</td>
<td>76.23</td>
<td>125.96</td>
<td>219.56</td>
<td>435.5</td>
</tr>
</tbody>
</table>

Table 1 – Technology parameters for Cu Interconnects (ITRS 2001)

This technology roadmap has served many chip design projects well and served as a good reference for our early planning phase. The design closest to our sample design and its process node is shown in the column labeled 0.1 in Table 1. Presumably these values were estimated or measured under nominal operating conditions, i.e. 25°C at the supply voltage specified in the table. In addition, we believe the devices were nominal, e.g. typical N and P transistors.

Comparing these ITRS design characteristics with those of our sample design, a baseline decoupling capacitance requirement was calculated. To further refine our model, values for Vdd and Freq were scaled to appropriate values. The results are shown Table 2 (estimated values for our sample design are highlighted). Note the following key information:

- Decoupling capacitance of 333 nF for the ITRS01-0.1 (Feature Size = 0.1 $\mu$m)
- The benchmark design includes both I/O pads and core logic
- We limited our scope to only the core, thus the entries for “% Core Area” in the table are set to 100%
- The upper bound for dynamic power consumption on our sample design is 1.762 W
- The upper bound for dynamic power density on our sample design’s core is 2.382 W/cm²
The rationale and assumptions for Table 2 are covered in detail in Section 2.5. Results are discussed in detail in Section 2.6.

<table>
<thead>
<tr>
<th>Table Column</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Upper Bound</td>
<td>Worst Case Estimate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Design's Digital Core</td>
<td>ITRS 2001</td>
<td>ITRS 2001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature Size (um)</td>
<td>0.1</td>
<td>0.07</td>
<td>0.09</td>
<td>0.09</td>
</tr>
<tr>
<td>X (mm)</td>
<td>24.94</td>
<td>26.70</td>
<td>8.6</td>
<td>8.6</td>
</tr>
<tr>
<td>Y (mm)</td>
<td>24.94</td>
<td>26.70</td>
<td>8.6</td>
<td>8.6</td>
</tr>
<tr>
<td>Area (mm²)</td>
<td>622</td>
<td>713</td>
<td>73.960</td>
<td>73.960</td>
</tr>
<tr>
<td>Freq (GHz)</td>
<td>3.000</td>
<td>5.000</td>
<td>0.400</td>
<td>0.377</td>
</tr>
<tr>
<td>VDD (V)</td>
<td>1.20</td>
<td>0.90</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>% Core Area</td>
<td>85%</td>
<td>73%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Est Power</td>
<td>160</td>
<td>170</td>
<td>1.762</td>
<td>1.762</td>
</tr>
<tr>
<td>Est Power Dens</td>
<td>25.723</td>
<td>23.843</td>
<td>2.382</td>
<td>2.382</td>
</tr>
<tr>
<td>Scaling Factors</td>
<td>Max Pwr (W)</td>
<td>160</td>
<td>170</td>
<td>3.000</td>
</tr>
<tr>
<td>Activity factor (α)</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.50</td>
</tr>
<tr>
<td>% non-dcap</td>
<td>100%</td>
<td>100%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Process factor</td>
<td>1.00</td>
<td>1.00</td>
<td>1.04</td>
<td>1.04</td>
</tr>
<tr>
<td>Dyn IR Sway</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>Results</td>
<td>Base dcap (nF)</td>
<td>333</td>
<td>377</td>
<td>30.833</td>
</tr>
<tr>
<td>Adj dcap (nF)</td>
<td>333</td>
<td>377</td>
<td>54.822</td>
<td>25.860</td>
</tr>
<tr>
<td>Adj PD (W/cm²)</td>
<td>25.723</td>
<td>23.843</td>
<td>4.056</td>
<td>4.056</td>
</tr>
<tr>
<td>Base DD (nF/cm²)</td>
<td>63.0</td>
<td>72.0</td>
<td>77.4</td>
<td>36.5</td>
</tr>
<tr>
<td>Adj DD (nF/cm²)</td>
<td>63.0</td>
<td>72.0</td>
<td>77.4</td>
<td>34.4</td>
</tr>
</tbody>
</table>

Notes:
Adj PD = Adjusted Power Density, Base DD = Baseline dcap density, Adj DD = Adjusted dcap density

Table 2 – Sample Design Decoupling Capacitance Estimates based on ITRS 2001 benchmark designs

The upper bound estimate (Column 3) assumes a maximum core power of 3 W (the upper limit allowed to maintain an acceptable worst case IR drop for our sample design), an activity factor (α) of 1.00 (every “data” signal switches once every clock), and no decoupling capacitance from any intrinsic source.

The worst case estimate (Column 4) describes a more realistic scenario where all the “data” signal capacitance switches every other clock (α = 0.50). Again, for this estimate, we assume that there is no intrinsic decoupling capacitance source in order to introduce some pessimism.

Note the power densities of approx. 4 W / cm² are identical for both scenarios. This is because a maximum power of 3 W is assumed in both cases.

The worst case estimates result in a total decoupling capacitance requirement for our sample design of approx. 26 nF, as highlighted in Table 2 (Column 4, lower-right).
Assuming a limited variation on $V_{dd}$ (less than or equal to 10%), the computed decoupling nF/cm² (for 0.18 μm and 0.07 μm, respectively, as shown in Table 2 and in more detail in [2] and [4]). Figure 3 shows the predicted “dcap density” for each process node, based on the ITRS-2001 studies. The detailed calculations are covered in [1].

![Sample design range](image)

**Figure 3 – Decoupling capacitance density vs. process node**

### 2.5 Table 2 Scaling Assumptions

The 0.1 μm ITRS benchmark design shown in Table 2 was refined to more closely match the sample design using a series of simple scaling factors, as outlined below.

1. "Est Power" is computed by scaling the ITRS01-0.1 "Power" value by “VDD” (squared), “Freq”, and “Area”.
2. “Est Power Density” is computed as (Est. Power) / Area.
3. "Max Power" is an adjustment from the computed "Est Power" value. It reflects the maximum allowable power for the package (in the case of the sample design) or the maximum estimated power for a realistic operating scenario.
4. “Activity Factor” is a fine adjustment used to derate “Est. Power” ($\alpha = C_{eff}/C_{sw}$).
5. "Process Factor" is a linear interpolation between ITRS01-0.1 and ITRS01-0.07. Used as an adjustment from the ITRS01-0.1 values, results in increased dcap requirement.
6. "Dyn IR Sway" is the allowable VDD dynamic droop, expressed as a percentage of VDD. All ITRS numbers assume a maximum allowable dynamic voltage swing of 10% of VDD. As design requirements change, this figure may need to be adjusted.
7. “% non-Dcap” describes the relative amount of dcap coming from non-dedicated Cox (gate oxide capacitance) cells. “Dedicated Cox gate cap” cells are those found in our 90nm library (DCAP*, DCAP*4, DCAP*8, DCAP*32, etc.).
8. “% Core Area” is an adjustment factor used to calibrate the dcap density figures gleaned from [3] with the ITRS results provided in [2] by subtracting out the effects of the I/O ring. In particular, we mean the “Base DD” (for “Baseline dcap density”) table entry. For example, a value of 85% is set for the ITRS01-0.1 table entry to produce a “Base DD” value of 63 nF/cm². This implies that 92.2% of each die edge is consumed by the core itself ($0.922^2 = 0.8501$). Note for the sample design entries in the table, this is set to 100% as we are limiting our scope to only the core.

With respect to Note #5 above, decoupling capacitance can come from four primary sources: 1) n-well junction capacitance of inactive devices 2) built-in capacitance, such as the Miller capacitance, of inactive devices, 3) sidewall/fringing capacitance of the P/G mesh, and 4) dedicated Cox gate capacitance. For *early planning purposes at the block-level*, we used a value of 20%. A value of 20% means that *at least one-fifth* of the on-chip decoupling capacitance is supplied by normal functional circuits that are not switching at any given time. These remaining inactive circuits act as decoupling capacitors as explained in [1] and [12]. To introduce some additional pessimism or margin the “upper bound” and “worst case” estimates selected a value of 0% in Table 2.

A detailed Excel spreadsheet covering assumptions, results, and estimations is available in [23].

### 2.6 Table 2 Results

This section explains some of the calculations found in the “Results” section of Table 2.

“Base dcap” (Baseline decoupling capacitance) is computed by scaling the ITRS-0.1 value of 333 nF by the following:

a) Scale by operating frequency $f$
b) Scale by $V_{dd}^2$

Note that this first calculation follows from Equation 2.7 and that no additional scaling is required for device area or power. This is because Equation 2.7 is not a function of area but $P$ is already a function of $f$ and $V_{dd}$ and thus implicitly scales as frequency and voltage are scaled.

“Adj dcap” (Adjusted decoupling capacitance) is computed by further scaling “Base dcap” by the following:

a) Max Pwr: Compares whether $P(\text{actual}) > P(\text{estimated})$. If true, the decoupling capacitance requirement increases.
b) Activity Factor: An adjustment based on block activity ($\alpha = C_{\text{eff}}/C_{\text{sw}}$). Note this must be less than or equal to 1.00. In all but the pathological case, the decoupling capacitance requirement decreases.

c) Process Factor: Pre-accounts for geometry scaling by increasing decoupling capacitance requirement, i.e. when we use the 0.1 µm as the starting point and transition to a 90 nm node.
d) Dyn IR Sway: Worst swing as a % of VDD; standard ITRS setting is 10%. Distinct from max static IR drop/rise

e) % non-Dcap: Decoupling cap from other intrinsic sources See Note #7 in Section 2.5.

Given the assumptions outlined above, the estimated worst-case power density of the sample design’s core is approx. 4 W/cm². Table 3 compares the sample design’s results directly with the appropriate ITRS guidelines found in Table 2. Note for the sample design case, we are choosing “Freq” to be \( f_{\text{max}} \) for the entire design, or 400 MHz.

<table>
<thead>
<tr>
<th>Process Node (um)</th>
<th>0.1</th>
<th>0.09</th>
<th>0.07</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ITRS</td>
<td>Sample</td>
<td>ITRS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>design core</td>
<td></td>
</tr>
<tr>
<td>Jo (A/cm²)</td>
<td>1.4e6</td>
<td>n/a</td>
<td>2.1e6</td>
</tr>
<tr>
<td>Chip size (mm²)</td>
<td>622</td>
<td>74</td>
<td>713</td>
</tr>
<tr>
<td>Vdd (V)</td>
<td>1.2</td>
<td>1.0</td>
<td>0.9</td>
</tr>
<tr>
<td>Freq (MHz)</td>
<td>3000</td>
<td>400</td>
<td>5000</td>
</tr>
<tr>
<td>P (W)</td>
<td>160</td>
<td>3</td>
<td>170</td>
</tr>
<tr>
<td>P_dens (W/cm²)</td>
<td>26</td>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>On-Chip C_dcap (nF)</td>
<td>333</td>
<td>27</td>
<td>377</td>
</tr>
<tr>
<td>Tmax (°C)</td>
<td>150</td>
<td>105</td>
<td>175</td>
</tr>
<tr>
<td># P/G Pads</td>
<td>2018</td>
<td>n/a</td>
<td>2560</td>
</tr>
<tr>
<td>R-local (kΩ/m)</td>
<td>219.56</td>
<td>n/a</td>
<td>435.5</td>
</tr>
</tbody>
</table>

Table 3 – Comparisons with ITRS 2001 guidelines

2.7 Comparison of design characteristics

A comparison of the sample design’s estimated power density with published data from Intel processor chips is included below [5]. This is important in that it sets the context of the sample design versus other industry standard designs. This comparison helped us to converge on a good estimate for our design’s decoupling capacitance requirements, which are largely based on power density figures.
2.8 Technical Summary

In summary, initial estimates indicate that a total of 25.680 nF of decoupling cap is required for the our sample design’s core, as described in Sections 2.4 through 2.7. For planning purposes, the decoupling capacitance was at first uniformly distributed across the child blocks based on their respective areas. As more information became available (e.g. when the top-level floorplan was better defined, more accurate gate counts were created, block level power analysis results were generated, etc.), the model was refined to improve accuracy and the dcap density varied slightly from block to block. This detail was captured and maintained in [23].

As a final note, the estimation methodology outlined above has shown to be accurate to first-order over several technology nodes, as described by the ITRS 2001 Report.

3.0 Implementation

Two implementation, or insertion, schemes are compared in this section. The first approach is a structured relative placement scheme based on design specific floorplan scripts. The second approach uses uniform distribution. This preferred scheme, highlighted below, uses a two-step iterative approach based on two TCL scripts. The first script runs in the Jupiter-XT tool at the floorplanning stage and the second script runs in an iterative manner in the Astro place and route tool at the post-CTS design stage. The dcap cells placed in the floorplan step ensure a uniform distribution throughout the design. The second script in the placement phase allows additional dcaps to be added after standard cell placement. Dcap cells can be added incrementally as needed.
3.1 Approach 1: Structured Relative Placement

This method is based on strict customer requirements (maximum distance between adjacent dcaps, co-placement with high-drive clock tree buffers, etc.). A relative placement approach, using a design-specific script, was used. Here, dcaps are inserted on alternating rows under the M6 power straps, as shown in Figure 5. M6 connects directly to M1 and the decoupling capacitors are spaced 100 μm apart.

Special requirements and implementation details

1) Request from the customer to uniformly distribute decoupling capacitors using very specific maximum distances between the them
2) Joint Customer-SNPS team stuck w/ the customer proposal, but altered location of the dcap cells slightly
3) Original power structure specified by the customer (originally m7 and m8 only) was changed to add vertical straps to m6, connect w/ via stacks; congestion point resulted from the via stacks, dcap cells connected directly to metal6
4) A specific pattern (using FILLCAP8 cells) resulted in an approx. 7% utilization of core. Specifically,
   - uniformly distributed dcap cells in arrays of four (4)
   - dcap cells placed every other row, with "exceptions" for rams (& other blockages)
   - dcap requirements were based on previous designs
   - those designs were unsuccessful, but the issues were not related to dcap strategies
5) Team had to define custom vias to avoid congestion between power plans
   - available contacts in via are staggered
   - no stacked vias is a requirement, so they are offset by a wiring track as you go up in metal layers
   - results in improve manufacturability as well
6) Post-route, they backfill with other filler dcaps as needed
7) Did some experiments w/ and w/o dcaps placed and noticed a significant change in routing congestion
8) Developed a related clock-buffer placement method: In Astro, combined with overlapping sites, implemented a clock buffer "region" near the dcap cells
9) In the end, utilization was in the 80%-85% range
3.2 Approach 2: Uniform Distribution

This method is loosely based on the TSMC Reference Flow 7.0, tested on our current design, and implemented in a future Synopsys Pilot release. The dcap cells are placed throughout the design as normal standard cells between row power and ground rails (see Figure 6). The decoupling capacitors are placed in a statistically uniform distribution throughout the design. The cells are arranged as equally through the design as possible with the decoupling capacitors composing the pre-determined percentage of the cell area.

The second approach is made up of the following two step strategy for placing the decoupling caps in an ASIC.

Step 1. Pre-placement using uniform dcap distribution

Details are outlined in Section 3.7. The recommended area allocation for dcap cells, on average, is 5%.

Step 2. Incremental dcap placement at the post-route stage

Here, the number of dcaps can be estimated and then placed based on intermediate power density analysis results and IR drop analysis results. This will most likely be done at the ECO cell and filler insertion stage. We recommend budgeting 1%-3% for this post-route step. In some cases for the blocks, due to their exact operating mode and conditions, some dcap cells may actually be removed at the post-route stage.

So in the worst case, final area allocation for an ASIC would be, on a global basis, between 6% and 8% of the core physical area. Due to the dynamic nature of power optimization and analysis,
this will understandably be an iterative process. As such, these numbers may require adjustment as you proceed toward design closure.

The diagram in Figure 6 implies a regular grid structure. This is for illustrative purposes only. Our actual implementation used a uniform random distribution based on a percentage of the total available floorplan area. This results in a distribution that isn’t so regular.

3.3 Comparison of approaches

Each approach has distinct advantages and disadvantages. The first differences are in the strict even distribution of decap cells versus the concentration of standard cells and macro cells. Another major defining feature of these approaches is the impact on the insertion of these cells on routing and placement congestion. An additional factor is the effect of these cells on chip area. Design complexity is also a major factor.

3.3.1 Structured Relative Placement Advantages/Disadvantages

With structured relative placement, you are guaranteed to have an equal distribution of decoupling capacitors throughout the design. This approach is useful when there is a relatively equal density of standard cells and macros in the design. The down side of this feature is that if timing requires a higher placement density of cells, no additional decap cells will be added in areas that may need it. Decoupling capacitors may need to have a higher concentration in areas of the ASIC where current densities are higher.
Placement and routing congestion effects are significant for this approach. Since the placement of dcap cells is a function of the power grid vias and via stacks, the impact to standard cell placement congestion is minimal. Since the location of the dcap cells is fixed relative to standard cells, it’s possible that routing congestion could be caused due what is effectively a matrix of routing blockages. For example, routing congestion was caused by the customer requirement that dcap cells be directly connected to M6. In the ASIC that this approach was tried, routing congestion was greatly increased when dcap cells were inserted when compared with no dcap inserted (see Section 3.1)

The impact to overall ASIC area with this approach should be minimal. Since the decoupling capacitors are essentially embedded in the same spot as the power grid vias, little additional standard cell area would be required to add these cells.

A negative for this approach is additional complexity in the implementation. The major complication was the requirement to create custom vias to prevent congestion between power planes.

3.3.2 Uniform Distribution Advantages/Disadvantages

With uniform distribution, you are not guaranteed to have an equal distribution of dcap cells throughout the design in an absolute relative placement between each DCAP cell. In this case, the dcap cells are evenly distributed throughout the design on an average over the entire ASIC. Clumping and less dense concentrations of dcap cells may occur due to high placement and low placement congestion respectively. This could have the effect of having less dcap cells in areas with the greatest current density. On average, this effect should not be significant except for in extremely high utilization areas.

Placement congestion effects could be significant for this approach in high congestion areas. Placing these cells in standard cell regions could cause additional placement congestion. This could cause cells in critical paths to not be placed in optimal locations.

The impact to overall ASIC area with this approach could be measurable. Since the cells are placed in standard cell rows, this could impact the size of a block.

A positive for this approach is there is relatively little complexity in the implementation. DMUX cells need to be placed as fixed placed standard cells. The complete flow will be described below.

3.4 Preferred Flow

Regardless of the approach taken, decoupling capacitors are inserted in a two step approach. The first step consists of inserting dcap cells before the standard cells are placed. This will ensure proper distribution of dcap cells in the design before timing driven placement cause clumping that can prevent even distribution of dcap cells. The second step is done post-route to add decoupling capacitors where necessary based on power rail analysis. This flow is used in a phased approach that corresponds to the stages of the ASIC development.
3.5 Phased Approach

Over the course of the project, a phased approach was used in developing the decoupling capacitance requirements and the implementation methodology. The initial phase was based on early estimation and planning. Phase 2 consisted of refinement based on block and top level floorplanning. The final Phase 3 is the signoff phase that is based on detailed rail analysis.

3.5.1 Early Estimation and Planning

The first phase, early estimation and planning, occurred during the initial planning phase of the project. Requirements were based on general ITRS guidelines. The implementation scripts were developed using a sample design as well as a couple of representative hierarchical blocks.

Estimated DCAP requirements were also determined by selecting the uniform distribution approach and from library and parasitic updates from the library vendor. Some minor tweaks in assumptions were required in following phases.

3.5.2 Implementation and Refinement

The next phase, refinement, consists mainly of estimating the amount of GDCAP* cells that will be needed in addition to the floorplanning-based DCAP* cells. GDCAP cells are special gate array ECO cells that can be wired as decoupling caps when they are not required for an ECO. This approach is an alternative to a conventional spare gate insertion methodology. As stated earlier, we targeted between 1% and 3% for this. The decoupling_cap.tcl script also handles this phase.

PrimePower results for mapped gates will allow us to dynamically refine the values for block-level power densities and to tweak the dcap allocation and distribution as needed.

3.5.3 Signoff

The final phase, signoff, requires detailed dynamic IR drop analysis with one or more vector-based activity (.vcd) files that come from gate-level simulations (e.g. VCS). A dynamic rail analysis tool like PrimeRail should be used for this purpose. The analysis scope in our approach is limited to the digital core (top-level plus all underlying major blocks). A static analysis tool like AstroRail should also be used to verify power grid integrity and static IR drop. The PrimePower-to-AstroRail and PrimePower-to-PrimeRail interfaces (using binary files) allow the use of instanced-based power information.

It is sometimes difficult, time-consuming, and disk-space intensive to generate these .vcd files, so as a backup you can use the same statistical switching estimates that have been used to constrain the design through synthesis and dynamic rail analysis. The caveat here is that the results will likely be pessimistic and you will have to qualify them as such.

Initially, the verification can be done using A/B comparisons and the goal will be to show the relative improvement in the worst case voltage droop. This will also allow us to determine if the tool is producing the expected results and if general setup/configuration changes are required. Once that milestone is achieved, we will use the results to determine whether dcap density
estimate modifications are required and to determine if additional characterization data is needed.

We plan to use the PrimeRail virtual dcap “what if” feature to quickly determine where additional decoupling cap cells are required. This will speed the iteration process. In some cases, we believe some decoupling caps densities will be too high (over design at the planning stage) and thus some cells will be either removed or moved to less congested areas.

### 3.6 Plan at the top-level

As mentioned above, decoupling capacitance planning was done at the top level during early design stages. It has been shown in the literature [6] and on similar internal projects that an optimal result will be achieved.

#### 3.6.1 Insert decoupling capacitors at the block-level

Although the planning was done at the top level and a uniform distribution was assumed to simplify the calculations, the decoupling capacitor insertion will be done at the block level (note our top-level design “sample_design_top” is also considered a “block level” for this discussion).

Each block will have uniform distribution, but the actually dcap densities will vary from block to block. This non-uniform distribution across the die (when viewed from the top level) is done to satisfy the block-level power density requirements. The same script and algorithm used for dcap insertion at the block level will be used at the top-level (sample_design_top). This is possible because the algorithm was enhanced for use with rectilinear placement regions. Detailed analyses are available in [23].

For example, the ARM1176 core, whose nominal clock frequency is 350 MHz is estimated in [23] to consume 104 mW at $V_{dd} = 1.0V$ under “typical” operating conditions (Temperature = 25C). With an estimated physical area of 4.989 mm$^2$, this module clearly has a higher power density (and thus high required dcap density) when compared with, say, our sample design’s “Video Output” sub-module operating under the same conditions but with maximum clock frequency of 267 MHz and estimated area of 11.630 mm$^2$ [23]. Inserting decoupling capacitors at the block level, after the global power grid has been pushed down to the respective block, will allow the global dcap density to be maintained while increasing the local density as required.

#### 3.6.2 Insert decoupling capacitors at the top-level

Insertion of decoupling capacitors at the top level is performed similar to inserting dcaps at the block level. The dcap density was adjusted to reflect the estimated IR drop requirements at the top level. Any dcap requirements caused by the coupling of blocks at the top level are also analyzed and fixed with a top level run.

### 3.7 Implementation details

Decoupling caps will be uniformly distributed using the native Milkyway `axgSpreadGroupCells` command. Distances between the caps will vary once the subsequent placement legalization is done. The dcap insertion will be done just after the detailed
power grid information is pushed down from the top-level design. The script to control this was adopted from an internal SPS repository and is called decoupling_cap.tcl. The exact distance between caps will vary from block to block depending on power density characteristics.

As in other Pilot™-based flows, the insertion is invoked via the following gmake target:

```
# Decoupling Capacitance Insertion - target=dcap_insertion
# --------------------------------------------------------------
$(LOG_DIR)/045_dcap_insertion/decoupling_cap.pass: \n   $(LOG_DIR)/040_power_insertion/power_insertion.pass
   $(MAKE_CMD) jxttcl \n   GEV_SRC=040_power_insertion \n   GEV_DST=045_dcap_insertion \n   GEV_SCRIPT=$(GEV_GSCRIPT_DIR)/fp/decoupling_cap.tcl \n   TEV_JXT_DCC_PERCENT_AREA=0.050 \n   TEV_JXT_DCC_CELL=DCAPHT32.FRAM \n   $(CHECK_LOG) \n   -must_not_have 'Error:' \n   -must_not_have 'ERROR'
```

This example specifies the DCAPHT32 cell as the dcap cell master and tells the script to allocate 5% of the placeable area to these dcap cells (“placeable area” comprehends a rectilinear placement area and excludes area of the macros, blockages, preroutes, and other fixed cells).

### 3.7.1 Physical Area Requirements

As summarized in Section 2.0, between 3% and 8% of our sample design’s physical area should be pre-allocated for decoupling cap cells.

Consider an example generic implementation that requires 5% of the core area to be reserved for decoupling capacitors. To make the example simple, assume a simple rectangular floorplan and that only DCAP32 cells are used. To simplify further, assume a hypothetical case in which the decoupling capacitors are placed in regular vertical columns and have a pitch that matches that of the METAL6 vertical straps. The distribution conceptually takes the form of Figure 6.

Note the “DCAP32” entry on the left side of Table 4. For this example, a total of 134726 cells can be placed uniformly to supply 20.519 nF of dedicated decoupling capacitance. Increasing the allocation to 8% using this same cell increases the total decoupling capacitance to 32.830 nF. Note that these values correlate well with the estimate described in Section 2.4.
Table 4 includes a rough calculation for the number of vertical columns needed to implement the required cells. In this case, 140 columns would be required as shown on the far right column of the table. However, this is for comparison purposes only; the actual implementation is done as shown in Figure 7 below.

For this implementation example, the total leakage power contribution of the dcap cells ranges from 16.900 mW to 27.040 mW, which is less than 1% of the total core power budget of 3 W. In the table above, the leakage figures are for the TC condition (ff,1.1V,25C). However, we need to recognize that the total leakage of the dcap cells an exponential function of temperature, as is the case with any other cell. Because of this, some reallocation may be required.

### 3.7.2 Incremental Refinement

As shown in some the implementation scenarios above, roughly 5% at the floorplan stage would provide the necessary decoupling cap to limit the dynamic $V_{dd}$ droop. Incremental amounts of dcap cells of between 1% and 3%, perhaps a mixture of DCAP* and GDCAP* cells, can be added during the ECO and filler cell insertion stages of the physical design for a total allocation of 6% to 8%. We believe this will satisfy the voltage droop targets but not have a negative impact on the overall placement and routing of our sample design. If problems arise and this recommended area allocation is not realistic, we will re-evaluate and adjust the block floorplans.

Refer to the Appendix Section 8.2 for an example script that can be run in Astro or Jupiter and can serve as the basis for this incremental post-route refinement step.

### 3.7.3 Jupiter-XT Scripts

The decoupling capacitors will be inserted at the block level using Jupiter-XT floorplanner TCL scripts. The scripts are based on the Pilot™ 1.2 release and can be easily customized to handle special design requirements.
3.7.4 Implementation Results

Figure 7 shows an example for a section of the “Video Output” block after the dcap insertion step. The decoupling capacitor cells are highlighted in red. This example specified TEV_JXT_DCC_PERCENT_AREA=0.075 in the dcap_insertion target.

![Figure 7 – Decoupling cap placement result for sample block (7.5% area utilization)](image)

4.0 Analysis and Verification

The implementation was verified using the PrimeRail dynamic rail analysis tool along with a set of related TCL scripts and procedures that automate the reporting and plotting of results and aid in the interactive analysis and debug process.

Prior to the design analysis, the standard cell libraries were characterized using PrimeRail’s built-in pgLibCharacterize and pgLinkPGSpec commands. The memories were modeled using the poTxGenDWM command to create an abstraction called Dynamic White-Box Model, or DWM.
4.1 PrimeRail Usage Model

Our sample design consisted of standard cells, memories, some mixed-signal IP, and I/O including two DDR2 channels running at 266 MHz. It used a peripheral I/O approach and was designed for a wirebond package. The die size was approx. 81 mm², and with a staggered I/O implementation, this left the digital core size at approx. 77.6 mm². As such, the core was more susceptible to static IR drop issues near its center.

The total instance count was approximately 2.8 M, not including the 261 memories (96 unique memories). The gate count in the digital core (including memories) was approximately 13.3 M. The design was partitioned into six major blocks, with the top-level design also considered a block. The basic design characteristics were as follows:

<table>
<thead>
<tr>
<th></th>
<th>Total Instances (incl memories)</th>
<th>Total gate equivalents (incl memories)</th>
<th>Memory Instances</th>
<th>Total clock domains</th>
<th>Fmax (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top only</td>
<td>665K</td>
<td>2.6M</td>
<td>11</td>
<td>39</td>
<td>400</td>
</tr>
<tr>
<td>B</td>
<td>492K</td>
<td>1.8M</td>
<td>16</td>
<td>5</td>
<td>266</td>
</tr>
<tr>
<td>R</td>
<td>586K</td>
<td>3.1M</td>
<td>85</td>
<td>18</td>
<td>377</td>
</tr>
<tr>
<td>O</td>
<td>564K</td>
<td>2.6M</td>
<td>62</td>
<td>5</td>
<td>150</td>
</tr>
<tr>
<td>D</td>
<td>159K</td>
<td>0.9M</td>
<td>22</td>
<td>1</td>
<td>133</td>
</tr>
<tr>
<td>E</td>
<td>163K</td>
<td>1.3M</td>
<td>32</td>
<td>4</td>
<td>350</td>
</tr>
<tr>
<td>A</td>
<td>125K</td>
<td>1.0M</td>
<td>33</td>
<td>3</td>
<td>350</td>
</tr>
<tr>
<td>Total</td>
<td>2.8 M</td>
<td>13.3 M</td>
<td>261</td>
<td>75</td>
<td>400</td>
</tr>
</tbody>
</table>

Table 5 - Sample design characteristics

The analysis using PrimeRail was limited to the digital core. We analyzed each major block after its timing closure began to converge. On average, three analysis passes were required to identify and correct all IR-drop related issues.

Run time for the blocks, again using the network described above, varied from 10 min to 40 min, mainly depending on the block size, the number of clock domains, and the frequencies of the clocks.

4.2 Standard Cell Characterization

The PrimeRail standard cell characterization process has been well documented, so the details of this process will not be included here. In summary, the PrimeRail pgLibCharacterize command was used to generate a current waveform profile for each standard cell’s power and ground pin and store the results in the Milkyway database.

It should be noted, however, that the characterization process is compute-intensive. During the course of our project we completed two full characterization regressions on approx. 800 standard cells. Three threshold variations (LVT, NVT, and HVT) were selected for characterization.

Using a network of approx. 30 multi-CPU Opteron servers and depending on system load, the total regression run time varied from approx. 2 days to just under 4 days. The first regression
was the faster of the two, as it saw a lower system load. For each regression, a total of approx.
10,000 spice simulations were required. The number of Spice runs per cell varied according to
cell complexity. The inverter cell Spice runs completed in a few seconds while the most
complex flip-flop and arithmetic cells completed in 3-4 minutes.

4.3 DWM-Lite Model Characterization

The full DWM is the most accurate model for running full-chip analysis as it is generated using
transistor-level parasitic extraction and circuit simulation results [27].

PrimeRail also provides a simplified model called DWM-Lite that is based on timing and power
parameters easily obtainable from a memory datasheet. Using this method, the user can still
create a model having acceptable accuracy but avoid running a detailed and time-consuming
transistor-level flow.

Due to number of memories in our sample design, the compute-intensive nature of creating full
DWMs, and project schedule pressures, we chose to create DWM-Lite models for our memories.

4.4 Characterization summary

In summary, it took one engineer just under two weeks to learn and understand the
characterization flow and to complete the first set of characterization runs (both standard cells
and DWM-Lite models). This included PrimeRail training and writing a small scripted
environment that allowed the second characterization run to run in batch mode completely
unattended. This scripted environment is available on request.

4.5 Analysis, debug, and design fixing

This section provides a few examples of how we verified the proposed method and also found
some real dynamic IR drop issues at the same time.

4.5.1 Test block verification results

We initially verified our flow using a small “test” block consisting of approximately 10K
standard cells and no hard macros (memories). In this case, the worst dynamic rail IR drop
occurred near two large clock buffers, as highlighted in red in Figure 8.
In the figure, all standard cells except for two clock buffers highlighted in white are hidden from view. The horizontal metal1 row straps are also hidden. The vertical VDD straps are highlighted in red. The vertical VSS straps are highlighted in blue. The dcap cells are highlighted in green.

A more detailed picture including the waveform probe points is shown in Figure 9. The PrimeRail probe points are marked as X’s in the figure. A corresponding waveform is shown in Figure 10.

Figure 9 and Figure 10 are labeled as follows:

- A is the VDD pin shared by the two clock buffers
- B is a standard cell VDD pin just to the right of a vertical metal2 VDD strap
- C is a point similar to B but very close to the next vertical metal2 VDD strap
The block had a sufficient number of dcap cells and their distribution was good (note the two decap cells just below the clock buffers), so the cause of the problem was initially unclear.

In Figure 9, the horizontal line under the probe pins shows the IR drop gradient along the metal1 row strap. Notice how the behavior shown in Figure 10 improves along the metal1 row strap as it gets closer to a lower-resistance metal2 (vertical) row strap near probe C.

Using PrimeRail’s “what-if analysis” (described in Section 4.7), we surrounded the clock buffers with “virtual” dcap cells, but this showed only a minor improvement. Figure 10 shows a comparison point “D” that shows the improvement after adding a large virtual dcap literally on top of the clock buffers. While this is not physically possible, it does show the very best case that could be achieved by adding another dcap cell near the clock buffers.
Figure 10 - Dynamic rail "bounce" on VDD near clock buffers

Additional study revealed the root cause of the droop in VDD: the two large clock buffers were in the same clock tree, they were switching almost simultaneously, and they straddled a horizontal metal1VDD row strap.

Ultimately we decided to fix the problem by adding placement halos around the large clock buffers. This would create an additional minimum space between them and ensure adjacent clock buffers would not straddle a rail as they were placed. It would also leave room for dcap cells to be placed using our proposed flow.

To reiterate, the dcap placement is done at the floorplanning stage and before clock tree synthesis. The fix we implemented will prevent this problem from occurring during initial placement, clock tree synthesis, and even post-placement optimization.

In summary, this issue was not a problem that our proposed dcap flow could prevent. However, our verification process uncovered a weakness in an unrelated point in the flow (initial placement, before clock tree synthesis).

4.5.2 Block “R” analysis results

Next we analyzed the largest high-performance block in our design. This block implemented an MPEG-2 decoder and had well over 500K placeable instances including 85 memories. It also had a high number of clock domains and a maximum frequency of 377 MHz.
The dynamic IR drop plot for VDD is shown in Figure 11. Note the “hot spot” at the left. This problem is localized to metal1 row straps between tightly placed RAMs.

Using our analysis scripts, we determined the cause of this issue to be due to the absence of decap cells between memories, as shown in Figure 12. In this case, a hard placement blockage had been added to the floorplan prior to decap insertion, and this prevented any cell, let alone a decap cell, from being placed there.
Our analysis here uncovered a real IR drop issue that could be fixed using a small number of dcap cells. In addition, it revealed another weakness in our overall flow. To fix the issue, the dcaps could have been manually placed in an incremental ECO and placement step. However, we decided to revise the flow to temporarily remove the hard placement blockage just before the decap insertion step and then replace it immediately thereafter.

We were curious why a similar problem did not occur in the small “P1” placement area below the top RAM and investigated further, as this area was also covered with a hard placement blockage. What we found was very interesting. There were several high-speed buffers in placement area “P2” that communicated with other logic placed above the top RAM, as shown by the net flyline. There were no such buffers in the “P1” placement area. The hot spot in the
metal1 row straps was further exacerbated by the lack of sufficient metal4 connections between the RAMs (“P1” and “P2” in Figure 12).

4.5.3 Block “O” analysis results

The video output stage in our sample design was the largest block in the design, consisting of almost 600K placeable instances including 62 memories, for a total equivalent gate count of about 2.6M. Its maximum frequency was relatively low and the clock structure was considerably simpler than that of the MPEG2 decoder block “R”.

![Figure 13 - Block "O" dynamic IR drop plot for VDD](image)

This block exhibited a new set of IR drop problems. The IR drop plot for VDD is shown in Figure 13. Note the very small hot spot to the upper left. At first glance, we did not understand the cause of the problem because the dcap distribution looked very good, as shown in Figure 14 (only dcap cells are shown in the standard cell placement area).
Drilling further, we discovered a problem similar to our test block but having to do with the set of I/O isolation buffers that were placed on the left side of the design between two very closely placed memories. The buffers with the worst IR drop characteristics were connected to input ports that had relatively high switching activity.

In contrast with this first case, the isolation buffers (sometimes called “snuggle buffers”) were tightly clumped, preventing any dcap cells from being placed amongst them. This is shown in more detail in Figure 15. The isolation buffers are highlighted in blue.

A complicating factor was that our placement flow fixed the placement of the isolation buffers prior to dcap cell placement. This problem could have been corrected using dcap cells, but the fixed placement nature of the cells prevented this.
In many cases, there were multiple back-to-back isolation buffers on these high activity inputs. One two-buffer path is highlighted with flylines in Figure 16. Although not shown in the figure, there was a hard placement blockage to the left of the RAMs that made the clumping worse.

For this situation, we spread the RAMs apart slightly and removed the hard placement blockage to the left of the RAMs so the isolation buffers could be placed in a more optimal fashion. In addition, we spread the memories apart slightly and allowed isolation buffers to be placed between them.
Again, our analysis revealed a weakness elsewhere in the flow that impacted our power and ground rail integrity. The flow was changed to prevent this type of situation in subsequent placement iterations.

4.6 Analysis and verification summary

In summary, we found no major issues with our proposed dcap flow. In the process of verifying our method, we analyzed several blocks and uncovered many real IR drop issues. We then implemented fixes for each of them, as described above.

4.7 What-if Analysis

PrimeRail was recently enhanced to allow the user to experiment with dcap cell placements “in place” using a concept called “virtual decoupling capacitors”. Similar to user-defined elements (UDEs) already supported in PrimeRail, this feature is built into the pgMap menu and allows the user to identify the problem area in the layout and then virtually place various combinations of dcap cells to quickly determine their effect on the dynamic rail drop signature. The user can place them on an instance basis using point-and-click or in a physical region using the mouse.
Although a similar “virtual resistance” what-if feature is available, we experimented only with the “virtual dcap” method and found it very easy to use.

4.8 Script Summary

While analyzing and debugging dynamic IR drop issues is both an interactive and visual process, we found a couple of areas where we could be more efficient by writing some helpful TCL procs. These scripts primarily speed up the generation of the IR drop maps and quickly identify the quality of the dcap distribution.

These procs are summarized below. Some selected procs are included in Appendix 8.3.

```
sps_power_pr_clear_pgmap      # Clear PrimeRail IR/EM/ER map
sps_power_pr_clear_pgnets     # Clear highlighting on power and ground nets (all layers)
sps_power_pr_create_pgmap     # Create PrimeRail IR-drop or Effective resistance map
sps_power_pr_hilite_decaps    # Highlight decap cells
sps_power_pr_hilite_pgnets    # Highlight power and ground nets (on visible layers only)
sps_power_pr_init_decap_info  # Initialize decap information from technology look up table
sps_power_pr_print_decap_info # Print decap look up table # technology information
sps_power_pr_setup_window     # Setup window display for visual debug # with pgMap
```

5.0 Conclusions and Recommendations

From our experience and review of available literature, we concluded that design teams find it difficult to determine the proper amount of decoupling capacitance required for a given design. We described a practical approach to the estimation, implementation, and verification of decoupling capacitors for deep submicron SoCs based on experience with a recent design at Synopsys Professional Services. The initial estimate for decoupling capacitance should be based on industry experience and guidelines from recent ASICs.

The objective of the practical method described herein produces an accurate decoupling capacitance allocation with minimal related area overhead. We compared structured relative placement versus uniform placement and concluded that uniform placement had a smaller impact on routing and placement congestion and is part of a standard flow that does not require custom structures. The method we adopted is now part of Synopsys` Pilot™ 1.3 design flow.

A phased approach with early estimation and implementation of dcap insertion is a key feature of our approach. The earlier in the project dcaps can be inserted, the sooner the impact to the design in the rest of the flow can be discovered (e.g. routability, area, power consumption). As these impacts are uncovered, the design flow can be altered to take these into account early instead of waiting to the end of the project when a problem can cause a tape out delay. We have found that the best results are obtained if the decoupling capacitors are initially placed at the floorplanning stage. This approach allows early discovery of routing and area impacts. It
also allows corrections to be easily made post-placement if rail analysis indicates additional decoupling capacitor cells are needed.

Dynamic rail analysis is essential for 90nm or smaller designs. Static analysis provides a picture of IR drop problems but it is incomplete for DSM designs. Flows exist to examine dynamic IR drop problems and also correct problems with the insertion of additional decoupling capacitor cells. Although we described some lessons learned for dynamic IR drop problems in this paper, a more in-depth paper covering best practices and detailed lessons learned is planned.

It may seem that allocating 7-10% of design area to dcap cells is excessive. For 90 nm and below, dynamic and static IR drop is a significant problem. Industry examples and rail analysis indicate that DSM ASICs can suffer from performance issues and even functional failures unless the proper amount of decoupling capacitance is allocated. Designers must plan for and implement enough dcaps as early as possible in order to ensure a working and timely delivered ASIC.

6.0 Acknowledgements

We would like to thank Ken Umino, Bill Holstein, and Julio Hernandez of Synopsys for reviewing our document and provided helpful feedback. Our reviewer, Mr. Mark Sprague of AMD, also provided valuable input that improved the quality of this paper.

7.0 References

Appendix

8.1 Pilot floorplanning implementation script [25]

```tcl
# DESCRIPTION:
# * This task is used to insert decoupling capacitors into all
# * levels of hierarchy. A degree of configuration is available
# * via optional flags.
# ---------------------------------------------------------------

tproc_source $GEV(gscript_dir)/fp/fp.common.tcl

# initialization
# ---------------------------------------------------------------

set configuration_error 0

set TEV_JXT_DCC_INSERTION_MODE [ tproc_read_TEV \   
-variable_name TEV_JXT_DCC_INSERTION_MODE \   
-variable_type oos \   
-oos_value "TOP HIER" \   
]
set TEV_JXT_DCC_PERCENT_AREA [ tproc_read_TEV \
-variable_name TEV_JXT_DCC_PERCENT_AREA \
-variable_type float \
-min_value 0.0 \
-max_value 1.0 \
-default_value 0.075 
]

set TEV_JXT_DCC_CELL [ tproc_read_TEV \
-variable_name TEV_JXT_DCC_CELL \
-variable_type string \
-default_value "DCAPHVT32.FRAM" 
]

set TEV_JXT_DCC_INST_PREFIX [ tproc_read_TEV \
-variable_name TEV_JXT_DCC_INST_PREFIX \
-variable_type string \
]

set TEV_JXT_GDCC_PERCENT_AREA [ tproc_read_TEV \
-variable_name TEV_JXT_GDCC_PERCENT_AREA \
-variable_type float \
-min_value 0.0 \
-max_value 1.0 \
-default_value 0.005 
]

set TEV_JXT_GDCC_CELL [ tproc_read_TEV \
-variable_name TEV_JXT_GDCC_CELL \
-variable_type string \
-default_value "GDCAPHVT10.FRAM" 
]

set TEV_JXT_GDCC_INST_PREFIX [ tproc_read_TEV \
-variable_name TEV_JXT_GDCC_INST_PREFIX \
-variable_type string \
]

set TEV_JXT_INSERT_GDCAP [ tproc_read_TEV \
-variable_name TEV_JXT_INSERT_GDCAP \
-variable_type integer \
-default_value "0" 
]

if {$configuration_error > 0} {
    tproc_msg -error "The task is incorrectly configured with $configuration_error errors."
    tproc_msg -error "Please correct and re-run."
    exit
}

## open database and design
tproc_copyMDB -src $GEV(mw_lib_src) -dst $GEV(mw_lib_dst)
tproc_openMDB -lib $GEV(mw_lib_dst) -design $GEV(block)

## insert decoupling capacitors to aid in IR issues

```
tproc_msg -info "Configuration for decoupling capacitor insertion:"
tproc_msg -info " percent area          = $TEV_JXT_DCC_PERCENT_AREA"
tproc_msg -info " dcap cell             = $TEV_JXT_DCC_CELL"
tproc_msg -info " dcap prefix           = $TEV_JXT_DCC_INST_PREFIX"
if {$TEV_JXT_INSERT_GDCAP} {
    tproc_msg -info "GDCAP ECO placeholder insertion enabled."
    tproc_msg -info "Configuration for GDCAP ECO placeholder insertion:"
    tproc_msg -info " percent area          = $TEV_JXT_GDCC_PERCENT_AREA"
    tproc_msg -info " gdcell                = $TEV_JXT_GDCC_CELL"
    tproc_msg -info " gdcap prefix          = $TEV_JXT_GDCC_INST_PREFIX"
}
tproc_source $GEV(tscript_dir)/fp/decoupling_cap_lut.tcl
```

## In TOP mode just insert at the top-level
## iterate over all designs if in HEIR mode

```
set DCC(designs) "$GEV(block)"
if {($TEV_JXT_DCC_INSERTION_MODE == "HIER")} {
    tproc_msg -info "Inserting DCAP cells hierarchically"
    foreach elem [tproc_get_macros -soft] {
        lappend DCC(designs) $elem
    }
} else {
    tproc_msg -info "Inserting DCAP cells in top-level only"
}
close_design -verbose $GEV(block)
```

```python
tforeach _design_name $DCC(designs) {
    
    ## misc. initialization
    
    open_design $_design_name
    set DCC(vertices) [ tproc_get_vertices ]
    set DCC(num_vertices) [ llength $DCC(vertices) ]
    tproc_unplace_std_cells
    
    ## insert one cap
    
    set DCC(inst_name) "${TEV_JXT_DCC_INST_PREFIX}0"
    create_cell -from_design $TEV_JXT_DCC_CELL -origin "0 0" "$DCC(inst_name)"
    -ignore_eco
    set DCC(cap_area) [ get_attribute [ get_cells -all $DCC(inst_name) ] area]
}
## compute how many caps to insert

```tcl
set hm_area [sps_fp_get_macro_area]
set total_area [get_attribute [get_design $_design_name] "area"]
set total_floorplan_area [sps_fp_get_floorplan_area]
set total_placeable_area [sps_fp_get_placeable_area]

tproc_msg -info " Total area is $total_area"

tproc_msg -info " Total floorplan area is $total_floorplan_area"

tproc_msg -info " Total macro area is $hm_area ([format %.2f [expr 100 * ($hm_area / $total_floorplan_area)]% of total floorplan area)"

tproc_msg -info " Total placeable area for standard cells is $total_placeable_area"

set DCC($_design_name,number_of_caps) [expr $total_placeable_area * $TEV_JXT_DCC_PERCENT_AREA ]
set DCC($_design_name,number_of_caps) [expr round($DCC($_design_name,number_of_caps) / $DCC(cap_area) ) ]
```

## insert the cap

```tcl
## insert the cap
##
## tproc_msg -info " Inserting $DCC($_design_name,number_of_caps) decoupling caps into $_design_name"
for {set i 1} {$i < $DCC($_design_name,number_of_caps)} {incr i} {
    set DCC(inst_name) \"${TEV_JXT_DCC_INST_PREFIX}$i\"
    create_cell -from_design $TEV_JXT_DCC_CELL -origin "0 0"
    \"DCC(inst_name)\"
}
```

## create a group

```tcl
## create a group
## aprCmdCreateHierGroup
formDefault create_group
setFormField create_group group_name dcc_$_design_name
setFormField create_group pattern \${TEV_JXT_DCC_INST_PREFIX}.*
#setFormField create_group group_type "Floor Plan"
formOK create_group
```

## spread a group

```tcl
## spread a group
##
## set _fname [open "my.axgSpreadGroupCells.$_design_name.scm" w]

puts $_fname "axgSpreadGroupCells"
puts $_fname "setFormField spread_group_cells group_name dcc_$_design_name"
for {set i 0} {$i <= $DCC(num_vertices)} {incr i} {
    puts $_fname "addPoint 1 \$\{lindex $DCC(vertices) 0\}\"
    set DCC(vertices) \$\{lrange $DCC(vertices) 1 end\} \$\{lindex $DCC(vertices) 0\}\"
}
puts $_fname "endEnterPoints"
puts $_fname "formCancel spread_group_cells"
close $_fname
```

tproc_source "my.axgSpreadGroupCells.$_design_name.scm"
## legalize_placement

## fix place caps

for {set i 0} {$i < $DCC($$_design_name,number_of_caps)} {incr i} {
  set DCC(inst_name) "${TEV_JXT_DCC_INST_PREFIX}$i"
  set_object_fixed $DCC(inst_name) true
}

## clean up - plan group not needed in subsequent steps

remove_plan_group -plan_group_names dcc_$_design_name
dbForceRecompCellBBox [geGetEditCell]

if {$TEV_JXT_INSERT_GDCAP} {
  tproc_overlap_rows -unit_tile gaunit
  ## insert one gdcap
  ## -> determine it's area
  set GDCC(inst_name) "${TEV_JXT_GDCC_INST_PREFIX}0"
  create_cell -from_design $TEV_JXT_GDCC_CELL -origin "0 0"
  "$GDCC(inst_name)" -ignore_eco
  set GDCC(cap_area) [get_attribute [ get_cells -all $GDCC(inst_name) ] area]

  ## compute how many gdcaps to insert
  set GDCC($$_design_name,number_of_caps) [expr $total_placeable_area * $TEV_JXT_GDCC_PERCENT_AREA ]
  set GDCC($$_design_name,number_of_caps) [expr round( $GDCC($$_design_name,number_of_caps) / $GDCC(cap_area) ) ]

  tproc_msg -info "Set unitTile to gaunit for the GDCAP10 ECO cells"
  scheme {axUseTileName (geGetEditCell) "gaunit"}
  scheme {axSetStringParam "place" "legalUnitTiles" "gaunit"}

  ## insert the cap
  tproc_msg -info " Inserting $GDCC($$_design_name,number_of_caps) gdcaps into $$_design_name"
  for {set i 1} {$i < $GDCC($$_design_name,number_of_caps)} {incr i} {
    set GDCC(inst_name) "${TEV_JXT_GDCC_INST_PREFIX}$i"
    create_cell -from_design $TEV_JXT_GDCC_CELL -origin "0 0"
    "$GDCC(inst_name)"
  }

  ## create a group
  aprCmdCreateHierGroup
formDefault create_group
setFormField create_group group_name gdcc_$_design_name
setFormField create_group pattern ${TEV_JXT_GDCC_INST_PREFIX}.*
#setFormField create_group group_type "Floor Plan"
formOK create_group

## spread a group
##
set _fname [open "my_gdcap.axgSpreadGroupCells.$_design_name.scm" w]
puts $_fname "axgSpreadGroupCells"
puts $_fname "setFormField spread_group_cells group_name
gdcc_$_design_name"
for {set i 0} {$i <= $DCC(num_vertices)} {incr i} {
    puts $_fname "addPoint 1 \{{[lindex $DCC(vertices) 0]}\}"
    set DCC(vertices) "[lrange $DCC(vertices) 1 end] [list [lindex $DCC(vertices) 0]]"
}
puts $_fname "endEnterPoints"
puts $_fname "formCancel spread_group_cells"

close $_fname
tproc_source "my_gdcap.axgSpreadGroupCells.$_design_name.scm"

## legalize_placement
##
legalize_placement

## fix place gdcaps
##
for {set i 0} {$i < $GDCC($_design_name,number_of_caps)} {incr i} {
    set GDCC(inst_name) "${TEV_JXT_GDCC_INST_PREFIX}$i"
    set_object_fixed $GDCC(inst_name) true
}

## clean up - plan group not needed in subsequent steps
##
remove_plan_group -plan_group_names gdcc_$_design_name
dbForceRecompCellBBox [geGetEditCell]
tproc_msg -info "Set unitTile back to unit after the GDCAP10 ECO cells are placed"
scheme {axUseTileName (geGetEditCell) "unit"}
scheme {axSetStringParam "place" "legalUnitTiles" "unit"}
}

##
## seriously, do a low effort placement to provide a good starting point for subsequent physical optimizations
##
tproc_unplace_std_cells
create_placement -effort Low -legalize
## create metrics for step, assumes LUT info available.
## Note: DCAP_TOT metric is in units of nF
## Note: DCAP_PER_AREA metric is in units of nF/cm^2
##
set all_instances [get_cells -all -hier *]

set dcap_instances [filter $all_instances "full_name=~*${TEV_JXT_DCC_INST_PREFIX}*$"]
set total_dcap_cap 0
foreach_in_collection inst_i $dcap_instances {
  set inst_ref_name [get_attr $inst_i "ref_name"]
  regsub {.FRAM} $inst_ref_name {} dcap_cell_name
  set total_dcap_cap [expr $total_dcap_cap + $dcap_cap_in_pf($dcap_cell_name)]
}
set dcap_cap_per_unit_area [expr 1e5 * ($total_dcap_cap / $total_placeable_area)] ;# pF/u^2 -> nF/cm^2

set gdcap_instances [filter $all_instances "full_name=~*${TEV_JXT_GDCC_INST_PREFIX}*$"]
set total_gdcap_cap 0
foreach_in_collection inst_i $gdcap_instances {
  set inst_ref_name [get_attr $inst_i "ref_name"]
  regsub {.FRAM} $inst_ref_name {} dcap_cell_name
  set total_gdcap_cap [expr $total_gdcap_cap + $dcap_cap_in_pf($dcap_cell_name)]
}
set gdcap_cap_per_unit_area [expr 1e5 * ($total_gdcap_cap / $total_placeable_area)] ;# pF/u^2 -> nF/cm^2

tproc_msg -info "Total of [format %0.3f $total_dcap_cap] pF of decoupling capacitance added in $_design_name"
tproc_msg -info "Decap per unit area is [format %0.3f $dcap_cap_per_unit_area] nF/cm^2 in $_design_name"
tproc_msg -info "Total of [format %0.3f $total_gdcap_cap] pF of ECO decoupling capacitance added in $_design_name"
tproc_msg -info "ECO Decap per unit area is [format %0.3f $gdcap_cap_per_unit_area] nF/cm^2 in $_design_name"

# this attribute gets set incorrectly by create_placement on some cells.
it is a bug.
set_attribute [ get_cells * -hier -filter "is_soft_fixed==true" ] is_soft_fixed false
if {{ [llength [get_cells * -hier -filter "is_soft_fixed==true"]] == 0} { tproc_msg -info "Checked for soft fixed cells, found none." } else { tproc_msg -error "Error: Soft fixed cells found in design."
  get_cells * -hier -filter "is_soft_fixed==true"
}}

## save and close design
##
save_design -verbose
close_design -verbose $_design_name

##
## do a final hierarchy preservation repair; for unknown reasons
## this is required after using create_cell to add the decap cells,
## otherwise you will see the following error later in the fp flow
## (specifically, in the 110_htv_and_budget task):
##
## astDumpHierVerilog
## Error: Hierarchy preservation information not in sync with flat netlist.
(MWNL-056)
## astDumpHierVerilog failed.
## Fail to execute command
##
## repair_hierarchy_preservation $_design_name
##
}
tproc_msg -info "... done.\n"

## ---------------------------------------------------------------------------
## End of file
## ---------------------------------------------------------------------------

8.2 Incremental cap insertion script [26]

## Fill areas around IP's and std_cell logic
axgAddFillerCellByArea
formDefault "Add Filler Cell By Area"
setFormField "Add Filler Cell By Area" \"Master Cell Name(s) With Metal" SNVT_FILLER_CELLS_WITH_CAP_LIST
setFormField "Add Filler Cell By Area" \"Connect to Power Net (optional)" "VDD"
setFormField "Add Filler Cell By Area" \"Connect to Ground Net (optional)" "VSS"
addPoint [geGetEditCell] [list 567.290000 7200.355000]
addPoint [geGetEditCell] [list 1380.330000 5777.535000]
addPoint [geGetEditCell] [list 4058.445000 7204.095000]
addPoint [geGetEditCell] [list 4960.215000 6338.515000]

## Fill one vertical stripe 64 micron wide with DECAPS and skip three
set xorigin 606.000
set yorigin 7165.14
set offset 64
for {set x 0} {$x< 28} {incr x} {
    addPoint [geGetEditCell] [list [expr $xorigin + $offset*4*$x] [expr $xorigin ]]
    addPoint [geGetEditCell] [list [expr $xorigin + $offset*(4*$x+1)] [expr $yorigin ]]
}
formCancel "Add Filler Cell By Area"

## Fill all the rest with fillers
axgAddFillerCell
formDefault "Add Filler Cell"
setToggleField "Add Filler Cell" "No Filler Under Mx" "M1" 0
8.3 PrimeRail Dynamic Rail Analysis script

```tcl
# DESCRIPTION:
# * This script performs a number of power-related functions using PrimeRail.
# * - Dynamic power analysis
# * - Dynamic rail analysis

# Check TEV variables for this script.

set configuration_error 0

set TEV_OP_MODE [tproc_read_TEV \
    -variable_name TEV_OP_MODE \
    -variable_type oos \
    -oos_values ( OP_WC OP_BC OP_TYP OP_BC0 OP_TL OP_ML ) \
    -default_value OP_WC ]

set TEV_BASE_NAME [tproc_read_TEV \
    -variable_name TEV_BASE_NAME \
    -variable_type string \
    -default_value NULL_BASE_NAME ]

set TEV_RAILOUT_FILE [tproc_read_TEV \
    -variable_name TEV_RAILOUT_FILE \
    -variable_type filename \
    -default_value NULL_RAILOUT_FILE ]

set TEV_PR_IRDROP_PEAK_THRESHOLD_PERCENT [tproc_read_TEV \
    -variable_name TEV_PR_IRDROP_PEAK_THRESHOLD_PERCENT \
    -variable_type float \
    -default_value $TVAR(config,irdrop,peak,percent) ]

set TEV_PR_REPORT_SIZE [tproc_read_TEV \
    -variable_name TEV_PR_REPORT_SIZE \
    -variable_type integer \
    -min_value 1 \
    -max_value 100 \
    -default_value 10 ]

if { $configuration_error } {
    tproc_msg -error "The task is incorrectly configured with $configuration_error errors."
}
```
tproc_msg -error "Please correct and re-run."
exit

## Local variables for this script.

set report_basename $GEV(rpt_dir)/$GEV(block).$GEV(step)_pr

set PP_CFG_FILE $GEV(work_dir)/$GEV(block).power_analysis_pr.$TEV_BASE_NAME.pp_vf.cfg
set FSDB_FILE $GEV(work_dir)/$GEV(block).power_analysis_pr.$TEV_BASE_NAME.fsdb
set RPT_POWER_INFO $report_basename.$TEV_BASE_NAME.poDumpPowerInfo
set RPT_POWER_SPICE_FILE $report_basename.$TEV_BASE_NAME.poRailAnalysis.power.spi
set RPT_POWER_WAVEFORM_FILE $report_basename.$TEV_BASE_NAME.poRailAnalysis.power.wdb
set RPT_GROUND_SPICE_FILE $report_basename.$TEV_BASE_NAME.poRailAnalysis.ground.spi
set RPT_GROUND_WAVEFORM_FILE $report_basename.$TEV_BASE_NAME.poRailAnalysis.ground.wdb
set RPT_RAIL_POWER_VIOLATIONS $report_basename.$TEV_BASE_NAME.pgMap.rail_power_violations
set RPT_RAIL_GROUND_VIOLATIONS $report_basename.$TEV_BASE_NAME.pgMap.rail_ground_violations
set RPT_EM_POWER_VIOLATIONS $report basename.$TEV_BASE_NAME.pgMap.em_power_violations
set RPT_EM_GROUND_VIOLATIONS $report basename.$TEV_BASE_NAME.pgMap.em_ground_violations
set PWR_MASTER_FILE $GEV(dst_dir)/$GEV(block).power_analysis_pr.$TEV_BASE_NAME.power_master_file
set POWER_NAME [lindex $TVAR(tech,power_net_list) 0]
set GROUND_NAME [lindex $TVAR(tech,ground_net_list) 0]

## Standard load

tproc_copyMDB -src $GEV(mw_lib_src) -dst $GEV(mw_lib_dst)
tproc_openMDB -lib $GEV(mw_lib_dst) -design $GEV(block)

## Include decap cells (marked as filler) in the C-intrinsic extraction

scheme { define poIncludeFiller 1 }

## In certain cases, if a resistance model is missing then the MW tech
## file information can be used. However, the user needs to turn on
## following switch

scheme { define poOrgRExtraction 1 }

## Set the number of "maximum powers" and "peak currents" reported in the log

scheme { (define pgStatListSize (getTclVar "TEV_PR_REPORT_SIZE")) }
## Power setup : Purge all power information.

```
poPurgePowerInfo
if { [file isdirectory $GEV(mw_lib_dst)/RAIL] } {
    poPurgeRail
    formOK purge_rail_view
} else {
    tproc_msg -info "$GEV(mw_lib_dst)/RAIL does not exist. Skipping 'poPurgeRail'."
}
```

## Power setup : Load power supply information.

```
switch $TEV_OP_MODE {
    OP_WC  { set POWER_SUPPLY_TDF_FILE $TVAR(tech,power_supply_tdf_file_op_wc) }
    OP_TYP { set POWER_SUPPLY_TDF_FILE $TVAR(tech,power_supply_tdf_file_op_typ) }
    OP_BC  { set POWER_SUPPLY_TDF_FILE $TVAR(tech,power_supply_tdf_file_op_bc) }
    OP_BC0 { set POWER_SUPPLY_TDF_FILE $TVAR(tech,power_supply_tdf_file_op_bc0) }
    OP_TL  { set POWER_SUPPLY_TDF_FILE $TVAR(tech,power_supply_tdf_file_op_tl) }
    OP_ML  { set POWER_SUPPLY_TDF_FILE $TVAR(tech,power_supply_tdf_file_op_ml) }
}
```

```
poLoadPowerSupply
formDefault Load_Power_Supply
setFormField load_power_supply database Dynamic
setFormField Load_Power_Supply TDF_File_Name $POWER_SUPPLY_TDF_FILE
formOK Load_Power_Supply
```

## Create config file for poTransientPowerAnalysis

```
set config_id [open $PP_CFG_FILE w]
puts $config_id "Power_DB_Type : Deterministics"
puts $config_id "Power_Source : PrimePower"
puts $config_id "PrimePower_Analysis_Mode : Vectorless"
puts $config_id "Current_Waveform_Alignment_Mode : Start"
puts $config_id "Current_Waveform_Point_Number : 20"
puts $config_id "Prime_Power_Reports : $TEV_RAILOUT_FILE"
close $config_id
```

## Create cell instance profiles from primepower report and library charact.

```
poTransientPowerAnalysis "$PP_CFG_FILE"
```

## Dump waveforms to file so they can be viewed

```
poDumpPowerDBToFsdb "$FSDB_FILE"
```
poRailAnalysis
formDefault P/G_Rail_Analysis
setFormField P/G_Rail_Analysis Power_Net_Name $POWER_NAME
setFormField P/G_Rail_Analysis Ground_Net_Name $GROUND_NAME
if {[sizeof_collection $all_power_pads] > 0} {
    setFormField P/G_Rail_Analysis Top-Level_Design_Pad 1
    setFormField P/G_Rail_Analysis Master_Or_Instance Master
    setFormField P/G_Rail_Analysis Pad_Name_File $PWR_MASTER_FILE
} else {
    setFormField P/G_Rail_Analysis Top-Level_Design_Pad 0
}
setFormField P/G_Rail_Analysis Top-Level_Design_Pin 1
setFormField P/G_Rail_Analysis User-Defined_Tap 0
setFormField P/G_Rail_Analysis Consider_Boundary_Conditions 0
setFormField P/G_Rail_Analysis Hierarchical_Option "Flatten Hierarchical Cells"
setFormField P/G_Rail_Analysis Extract_Net "Extract Power Net"
setFormField p/g_rail_analysis analysis_option Transient
setFormField P/G_Rail_Analysis Delay_Scaling "Store Current Values"
setFormField p/g_rail_analysis res-cur_file $RPT_POWER_SPICE_FILE
setFormField p/g_rail_analysis transient_waveform_file $RPT_POWER_WAVEFORM_FILE
formApply P/G_Rail_Analysis
setFormField P/G_Rail_Analysis Extract_Net "Extract Ground Net"
setFormField p/g_rail_analysis res-cur_file $RPT_GROUND_SPICE_FILE
setFormField p/g_rail_analysis transient_waveform_file $RPT_GROUND_WAVEFORM_FILE
formOK P/G_Rail_Analysis

## -------------------------------------
## Read the TDF file to get desired supply voltage so that ...
## -------------------------------------

set tdf [open $POWER_SUPPLY_TDF_FILE r]
while {{gets $tdf line} >= 0} {
    if {[lindex $line 0] == "tdfSetPowerSupply"} {
        if {[lindex $line 1] == $POWER_NAME} {
            set SUPPLY_VOLTAGE [lindex $line 2]
            tproc_msg -info "The supply voltage for $POWER_NAME at conditions $TEV_OP_MODE is $SUPPLY_VOLTAGE"
        }
    }
}
close $tdf

## -------------------------------------
## ... we can define the allowable IR drop as a perc. of the supply voltage.
## -------------------------------------

set PERCENT_DROP $TEV_PR_IRDROP_PEAK_THRESHOLD_PERCENT
set LOWER_BOUND_IN_V [expr 0.0 - (($PERCENT_DROP/100.0) * $SUPPLY_VOLTAGE)]
set LOWER_BOUND_IN_MV [expr 1000.0 * $LOWER_BOUND_IN_V]
set UPPER_BOUND_IN_MV [expr -1.0 * $LOWER_BOUND_IN_MV]
if { $GEV(gui) } {
    sps_power_pr_setup_window
}
## Create the IR report for the Power net

```plaintext
if { ! $GEV(gui) } {
    # Text reports will not be generated unless we are in gui-mode
    start_gui
}

pgMap
formDefault display_map
setFormField display_map results_net-mode Default:Default:$POWER_NAME
formButton display_map load
setFormField display_map display_option "Voltage Drop"
setFormField display_map auto_range 0
setFormField display_map (u)upper_bound 0.000
setFormField display_map (l)lower_bound $LOWER_BOUND_IN_MV
formButton display_map reportSetup
setFormField display_map vd_error_file_name $RPT_RAIL_POWER_VIOLATIONS
subFormHide display_map 5
formApply display_map
formButton display_map report
formOK display_map
```

## Create the IR report for the Ground net

```plaintext
pgMap
formDefault display_map
setFormField display_map results_net-mode Default:Default:$GROUND_NAME
formButton display_map load
setFormField display_map display_option "Voltage Drop"
setFormField display_map auto_range 0
setFormField display_map (u)upper_bound $UPPER_BOUND_IN_MV
setFormField display_map (l)lower_bound 0.000
formButton display_map reportSetup
setFormField display_map vd_error_file_name $RPT_RAIL_GROUND_VIOLATIONS
subFormHide display_map 5
formApply display_map
formButton display_map report
formOK display_map
```

sps_power_pr_clear_pgmap

## Rail EM&IR analysis: Perform EM analysis

```plaintext
# Rail EM&IR analysis : Perform EM analysis
```

## Create the EM report for the Ground net

```plaintext
pgMap
formDefault display_map
setFormField display_map results_net-mode Default:Default:$POWER_NAME
formButton display_map load
setFormField display_map display_option EM
formButton display_map EMReport
```
8.4 Selected PrimeRail debug procs

```tcl
proc sps_power_pr_clear_pgmap { args } {
    parse_proc_arguments -args $args options

    # global variables
    global env GEV TVAR TEV_BASE_NAME TEV_OP_MODE
    global sh_product_version synopsys_program_name
    global link_library

    current_design $GEV(block)

    # local variables
}```
define_proc_attributes sps_power_pr_clear_pgmap \
    -info "Clear PrimeRail IR/EM/ER map" \
    -define_args {

## -------------------------------------
## Main
## -------------------------------------

pgMap
formDefault display_map
formButton display_map remove
formCancelButton display_map
setView 1 "fitplus"

return 1

}

define_proc_attributes sps_power_pr_setup_window \
    -info "Prepare window for IR and EM plots by removing some detail" \
    -define_args {

## -------------------------------------
## Main
## -------------------------------------

parse_proc_arguments -args $args options

# global variables
#-------------------------------------
# global env GEV TVAR TEV_BASE_NAME TEV_OP_MODE
global sh_product_version synopsys_program_name
global link_library

current_design $GEV(block)

# local variables
#-------------------------------------

# Prepare window for IR and EM plots by removing some detail
#-------------------------------------

gewWindowOption
setToggleField window_options visible_object std/_module_cell_instance 0
setToggleField window_options visible_object contact 0
setToggleField window_options visible_object path 0
setToggleField window_options visible_object polygon 0
formOK window_options
setView 1 "fitplus"

return 1

}

define_proc_attributes sps_power_pr_setup_window \

## sps_power_pr_create_pgmap

```
# sps_power_pr_create_pgmap
proc sps_power_pr_create_pgmap { args } {
    #-------------------------------------
    # args
    #-------------------------------------
    set map_type ir # map_type=ir|em|er;default=ir
    set options(-show_legend) 0 # map_type=ir|em|er;default=ir
    set options(-pnet_name) "VDD"
    set options(-max_threshold) 10
    set options(-min_threshold) 2
    set options(-pg_metal_list) [list m1 m2]
    set options(-analysis_type) max ;# analysis_type=max|ave

    parse_proc_arguments -args $args options

    #-------------------------------------
    # Do some basic error checking on args
    #-------------------------------------
    if { [llength $pg_metal_list] <= 0 } {
        tproc_msg -info "Found invalid pg_metal_list '$pg_metal_list'
        incr args_error
    } else {
        redirect /dev/null {
            set pg_net_type [get_attr -quiet -all $pnet_name] net_type
        }
    }
    if { $pg_net_type == 0 } {
        tproc_msg -error "Cannot find pg net '$pnet_name'
        incr args_error
    }
```
```bash
if { $args_error } {
    tproc_msg -error "Invalid argument list, exiting."
} else {
    tproc_msg -info "Detected net_type of '$pg_net_type' for net '$pnet_name'"
    if { $pg_net_type == "Power" } {
        set max_threshold [expr -1.000 * $max_threshold]
        set min_threshold [expr -1.000 * $min_threshold]
    }
    sps_power_pr_clear_pgmap
    tproc_msg -info "User specifies max threshold of [format %0.2f $max_threshold]"
    tproc_msg -info "User specifies min threshold of [format %0.2f $min_threshold]"
    switch -glob -- $map_type {
        ir* {
            set map_type_str "Voltage Drop"
        }
        em* {
            set map_type_str "EM"
        }
        default {
            tproc_msg -info "Selecting default map type of 'ir'"
            set map_type_str "Voltage Drop"
        }
    }
    switch -glob -- $analysis_type {
        max* {
            set analysis_type_str "MAX"
        }
        ave* {
            set analysis_type_str "AVG"
        }
        default {
            set analysis_type_str "MAX"
            tproc_msg -info "Selecting default analysis type of '$analysis_type_str'"
        }
    }
    pgMap
    formDefault display_map
    setFormField display_map results_net-mode Default:Default:$pnet_name
    formButton display_map load
    setFormField display_map transient_stats $analysis_type_str
    setFormField display_map display_option $map_type_str
    formButton display_map V_pgMap_mapConfig
    formButton display_map V_pgMap_dspLayer_offmetal
    foreach metal_i $pg_metal_list {
        setFormField display_map $metal_i 1
    }
```
define_proc_attributes sps_power_pr_create_pgmap \
- info "Create PrimeRail IR, EM, or Effective resistance map" \ 
- define_args { 
  (-pnet_name "Power/Ground net name (default=VDD)" "" string optional) 
  (-analysis_type "Analysis type (mode) (default=max)" 
   "[max|ave]" string optional) 
  (-pg_metal_list "Specify P/G metals for analysis 
   (default=[list m1 m2])" "" string optional) 
  (-max_threshold "Filter analysis with max threshold in mV 
   (default=10)" "" float optional) 
  (-min_threshold "Filter analysis with min threshold in mV 
   (default=2)" "" float optional) 
  (-pg_metal_list "Specify P/G metals for analysis 
   (default=[list m1 m2])" "" string optional) 
  (-show_legend "Show legend on pgMap" "" boolean optional) 
  )}