Generating Realistic Stimuli for Accurate Power Grid Analysis

P. Marques Morgado [1] morgado@algos.inesc-id.pt
Paulo F. Flores[1] pff@inesc-id.pt
L. Miguel Silveira [2] lms@inesc-id.pt

[1]INESC–ID
Dept. of Electrical and Computer Eng.
IST - Technical Univ. of Lisbon
Lisboa, Portugal

[2]INESC–ID / Cadence Laboratories
Dept. of Electrical and Computer Eng.
IST - Technical Univ. of Lisbon
Lisboa, Portugal

Abstract

Power distribution systems in integrated circuits are used to provide the voltages and currents the devices need to operate properly. As the semiconductor industry moves into deep nanometer nodes, problems like voltage drop, ground bounce and electromigration which may cause chip failures, are worsening, as more devices, operating at higher frequencies are placed closer together. Verification of a power distribution system is therefore paramount to silicon success. This type of verification is usually done by simulation, targeting a worst-case scenario, typically characterized by the almost simultaneous switching of several devices in the circuit. The definition of the worst-case situation is therefore crucial, since it influences the result of the simulation and ultimately the design target. Supposedly safe but unrealistic settings such as assuming that all signals switch simultaneously, could lead to costly over-designs in terms of die area. In this paper we describe a software tool that generates a reasonable, realistic, worst-case set of stimuli for simulation, based on timing and spatial restrictions that arise from the circuit's netlist and placement. Generating such stimuli is akin to performing a standard static timing analysis, as is done before signoff, so the tool fits well within conventional design frameworks. The resulting stimuli indicates that only a fraction of the gates change in any given timing window, leading to a more robust verification methodology, specially in the dynamic case.

Keywords: Power grid, verification, simulation, voltage drop, ground bounce

1 Introduction

The power distribution system design is an area of increasing concern in semiconductor industry. According to data in [1], more than 50% of tapeouts using 0.13-micron technology would fail, if the power distribution system were not validated beforehand. Lower operating voltages, increased device integration density and leakage currents, higher operating frequencies and the use of low power design techniques, they all tend to stress the power grid as technology evolves. The design of such systems is complex and error-prone, since there is a wide variety of aspects that must be taken into account. Of these, perhaps the four major problems that may affect power distribution systems are voltage drop, ground bounce, L di/dt noise and electromigration [4].

Voltage drop, also called IR drop, is the voltage reduction that occurs on power supply networks. The IR drop can be static or dynamic and results from the existence of non-ideal elements: the resistance within the power and ground supply wiring and the capacitance between them. While static voltage drop considers only the average currents, dynamic voltage drop considers current waveforms within clock cycles and has a RC transient behavior. Similar effects may be found in ground wiring, usually referred as ground bounce. Both effects contribute to lower operating voltages within devices (i.e. logic cells/gates in digital circuits), which in general increase the overall time response of a device and might cause a failure in its operation. The L di/dt noise is caused by current spikes on wires that will induce abrupt voltage changes on these wires and their neighboring wires, due to inductance coupling [6], [2]. Finally, electromigration in the power distribution system is the movement of metal atoms from a certain region to another, and is caused by high current densities. Electromigration can itself lead to voltage drop and/or ground bounce, as metal lines wear out.

Several approaches can be taken with the goal of minimizing or eliminating these problems such as the insertion of decoupling capacitance (decap) and/or the use of wider metal lines. Decoupling capacitance work as a "charge reservoir", in a situation where several devices become active, and the power grid is unable to deliver the total required current. The location of the decoupling capacitors within the power grid is extremely important. Badly placed decaps may have a reduced contribution for voltage drop and ground bounce elimination and are a waste of die area, since in general they consume a considerable amount of silicon [7].

Power grid analysis and verification is, from a practical standpoint, one of the most important steps in the design flow, yet computationally a very complex one. Power grid verification is usually accomplished by simulation [5], [9]. The disadvantage of simulation is that stimuli must be generated very carefully such that the relevant scenarios are accounted for. Only settings corresponding to the chosen stimuli are simulated and thus verified, so they should be chosen appropriately and should be representative of relevant situations. Since the power grid encompasses the whole die area, its description is rather large and the simula-

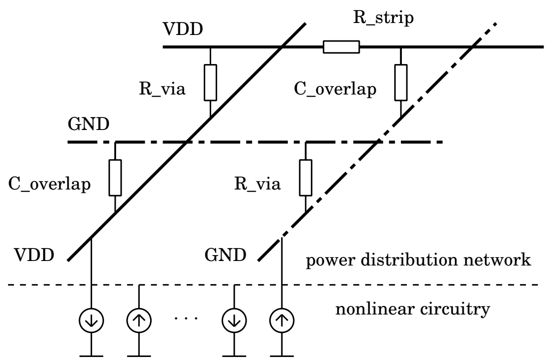

Figure 1. Simplified power grid model.

tion process is slow and highly complex. This results from the necessity to take into account a huge number of power grid parameters (RLC non-idealities) and all the devices that take current from it, as shown in Fig. 1. Simulating the power grid with all the devices might be impossible for VLSI circuits, as it would consume too many resources. Furthermore, simulating for all possible device settings is also impossible, as it would take too long. In addition, given the size of current designs, it is also impossible to assume that designer intervention will be sufficient to generate appropriate sets of stimuli for the grid verification. Typically no single designer knows enough of the circuit behavior, as a whole, to perform such task. Even if that were possible, it would require considerable effort from designers and might delay significantly the design process. Therefore, an automatic way of generating realistic sets of stimuli given the knowledge of the actual circuit implementation is necessary. Hopefully, this task can be integrated in a standard design & verification framework and accomplished efficiently in an acceptable time.

Timing and spatial constraints arising from the circuit netlist and placement prevent most of the devices from being active at the same time. Therefore, power grid simulation that considers all the circuit devices as simultaneously active is clearly unnecessary. Voltage drop and ground bounce may occur if there is a significant number of devices becoming active in a short period of time and drawing current from close regions of the power grid. We propose a technique to determine, within a time frame, how many devices become active on nearby regions of the power grid. The higher the number of devices in this situation, the greater the possibility of voltage drop and/or ground bounce effects be felt in the power grid (assuming the grid has a regular structure and that all devices take an equal amount of current). Combining timing and spatial information obtained in a traditional design flow it is possible to know when and which devices are active in order to generate a set of more realistic worst-case stimuli for grid verification.

The outline of this paper is as follows: in Section 2, we present the traditional design flow with emphasis on the power grid design and analysis. In Section 3 a new design flow is proposed in order to determine a set of more realistic power grid stimuli. A more detailed explanation of our method to identify problematic devices is presented in Section 4. The results obtained using the proposed method are presented in Section 5, and the conclusions drawn from these results are presented in Section 6.

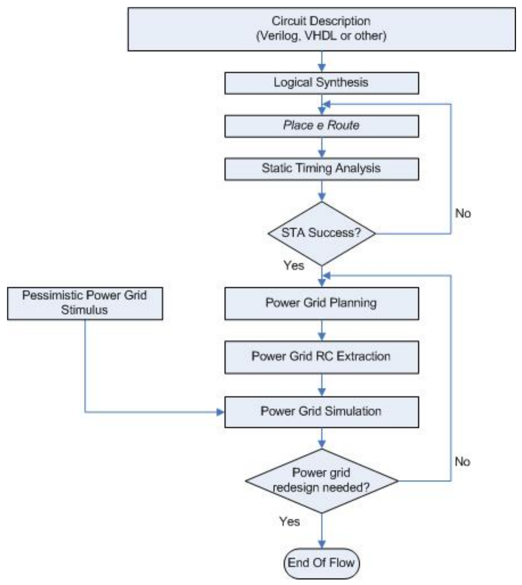

Figure 2. Traditional design flow (with emphasis on power grid design).

2 Traditional Design Flow

Simulation is the most commonly used method to validate the power grid. It enables one to verify if the power grid is suited for a given design, that is, if it is robust enough to deal with problems such as voltage drop and ground bounce. In Fig. 2 a simplified version of a standard design flow, with emphasis in power grid design and analysis, is presented. As can be seen in Fig. 2 a typical flow starts with a circuit description in VHDL or Verilog. This description is converted into a gate-level netlist of a given technology library during logical synthesis. After synthesis, place & route of circuit cells is done. To ensure the circuit timing sign-off, static timing analysis (STA) is usually done afterwards. If STA fails, a new place & route should be performed. Then, power grid planning is done based on the knowledge of power distribution along the circuit [3]. However, this knowledge is, of course, rather limited.

After the power grid design, a simulation (at electrical level) of the grid along with the "circuit" is performed. For that purpose, a RC extraction is done as well as the definition of power grid stimuli. This mainly consists on the definition of the circuit cells, which must be considered throughout the simulation, and their correspondent current waveforms. This, or similar methodologies are commonly used and they all share the same problem: the definition of power grid stimuli for an accurate power grid simulation.

Some of the simulation tools consider all the circuit devices as stimuli to the power grid. Others allow users to define which stimuli should be applied, i.e., which circuit cells are going to be considered during simulation. Most of the times this definition is based on user experience and knowledge. However, both options may deteriorate the quality and resulting accuracy of power grid simulation. A critical region may be neglected if the user misses the combination of grid stimuli that will cause the worst voltage drop or ground bounce (a false negative). Results from a simulation obtained on the assumption that all cells need to be

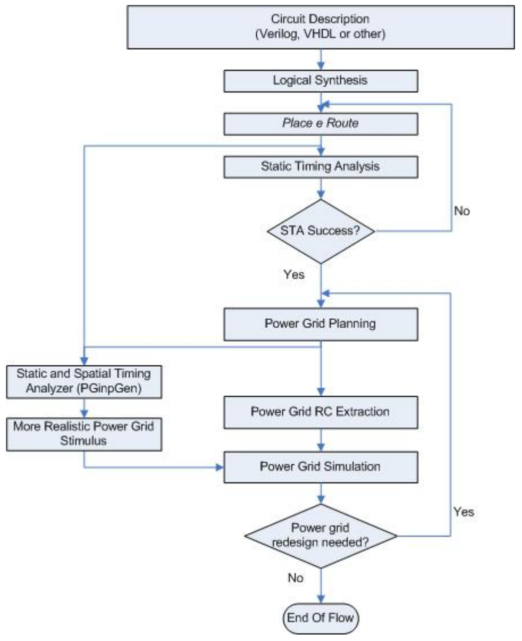

Figure 3. Proposed design flow (with emphasis on power grid design).

accounted for, may also identify invalid critical regions of the power grid that are supposedly affected by voltage drop or ground bounce (a false positive). This occurs because in normal working conditions all cells in the circuit can not draw current from the power grid at the same time. Moreover, this type of simulation may also increase total runtime and memory requirements from simulators. After this simulation procedure, the designer will try to solve IR-drop problems, usually by placing decoupling capacitance inside those critical regions. If those regions are non-critical, from a voltage drop and ground bounce point of view, the insertion of decoupling capacitance will only increase the overall static power consumption and it will be a waste of silicon area. This circuit changes can itself cause voltage drop and ground bounce to appear in other circuit regions.

3 Proposed Design Flow

In this section we present an alternative power grid design flow, which uses placement and timing information to determine a reasonable realistic worst-case stimuli for simulation of power grids. This new flow is presented in Fig. 3. As shown in this figure, we introduce an additional task in the flow: the Static and Spatial Timing Analysis. This task merges the spatial information available after cell placement and power grid planning, and the timing relationships that exist between cells. These timing relationships, which are a consequence of circuit topology and cell delays, can be easily obtained using a modified version of a static timing analyzer. The objective of this task in the flow is to find a set of power grid simulation stimuli that are more realistic than those obtained by traditional design flows.

Assuming the power grid has a regular structure, which is generically valid and certainly in a context of standard cells, the critical region of the power grid will be the determined by high levels of local cell activity. Specifically it will be where there is a high number of nearby cells all drawing current from the power grid in a short period of time. The switching of the cells demands from the power grid an amount of current which may not be available in that period of time. In this situation, the integrity of the power grid is affected, and the typical circuit behavior may also be jeopardized. Note that it is important not only that all cells draw current at nearby instants of time, but also that the current is drawn from the same region of the power grid. For example, consider the output of a cell with a fan-out of 7. When the output of the cell changes, 7 other cells might become active and draw current from the power grid. If these cells are placed far from each other, the power grid is able to handle this situation which therefore does not cause much concern. A similar situation occurs if we consider a chain of 10 inverters placed closer together. One logic change in the input to the first inverter will force all other inverters to become active. However, due to the propagation delay of each cell, the maximum current drawn from the power grid does not occur simultaneously, and may not cause a significant problem to the power grid.

Using an STA-like method we can obtain the time instants where each cell has the possibility of drawing current from the grid, i.e., the instants where each cell may become active/switch. The placement information allows us to know the location of each cell along the die, as well as their connection to the power grid. Combining this timing information with the spatial information, obtained from the placement, we know which regions of the power grid may suffer the strongest impact in terms of drawn current in a short time span. Note that most of the information needed for the new task may already be available by the normal execution of the traditional flow, but it is discarded in the power grid analysis tasks.

4 Static and Spatial Timing Analyzer

The static and spatial timing analyzer task uses information available after place & route operation to generate a more realistic set of simulation stimuli. To determine which devices may become active in a time frame, the circuit propagation delays are obtained using techniques similar to those employed by static timing analysis (STA) tools. In order to determine the region of the power grid that is most affected by voltage drop or ground bounce, placement information is used. In the next sections we describe the models and the algorithms used to compute the timing and spatial information and how both are combined by the PGinpGen tool to identify the cells and the corresponding region of the power grid in which a severe voltage drop or ground bounce may occur. A complexity analysis of the algorithm proposed is also included.

4.1 Timing Information

Timing information regarding device activation enables one to know the time frame where the highest number of devices switching can be. In order to determine the time instants where each cell can become active, an operation similar to a static timing analysis (STA) is performed.

Traditional STA traverses the circuit computing the worst arrival time of all input signals for each cell. Since STA only considers the inverting or non-inverting function of cells, this type of analysis is almost cell-independent and can be done several times faster than logic simulation. However, from the power-grid analysis perspective, traditional STA can not be used to determine the cells switching activity, since it only keeps the maximum delay for each cell (potentially the minimum delay can also be computed if early mode analysis is done, but in any case only information on one transition is kept). Unfortunately, nothing prevents a significant number of cells from switching prior to their maximum delay and almost at the same time, causing a seemingly worst-case impact on the power grid. For example, consider an AND gate with two inputs ans consider that the time instants they can switch are $3ns$ and $6ns$, respectively. Assume that the output may change within $2ns$ delay (typical propagation delay of the gate) after an input change. So, we need to account for a possible output switch at instants $5ns$, and $8ns$, as each of these may propagate to other gates leading to different switching activity. Therefore, we need to calculate and propagate all possible activation instants throughout the circuit, not only maximum delays.

Fortunately, computing all activation instants in a circuit can be done with some additional processing while traversing the circuit for STA. This operation may seem daunting since larger circuits will have a huge number of possible activation instants. However, in general most of the activation instants are close in time so, we can combine them into intervals with a relatively small loss of precision. In the case of the previously mentioned AND gate, we would say that the output could be active in an interval from $5ns$ to $8ns$ since the gate can switch any time during that interval. The creation of intervals is controlled by a parameter that can be changed by the user, trading precision for performance.

Having determined all possible activation instants/intervals along the circuit, we can determine the worst timing window (i.e. where there is the largest number of cells with the possibility of switching, i.e., becoming active). Conceptually this is achieved by computing the maximum number of intervals that fall inside a sliding window with a user defined size. To improve performance, the instants/intervals are placed in four linked lists, where two of the lists contain the minimum and maximum values of the rise transitions sorted in ascending order, and the other two the fall transition information sorted in a similar way. The first list of each type of transition allows us to know when each cell can become active. The other lists tells us when each cell is no longer active. Dividing real time in several virtual timing windows, it is then possible to compute the timing frame where there is the largest number of cells with the possibility of becoming active.

4.2 Spatial Information

As mentioned in Section 3, the worst-case situation from a power perspective corresponds to having a significant number of devices becoming active and drawing current from close regions of the power grid. This introduces a necessity of not only having timing but also spatial information, namely the spatial information describing the connection between the devices and the power grid. To account for the impact that active cells have upon the power grid, we create a discrete model of the circuit power grid. This model is composed by grid points that result from the intersection of the several power grid structures (intersections between

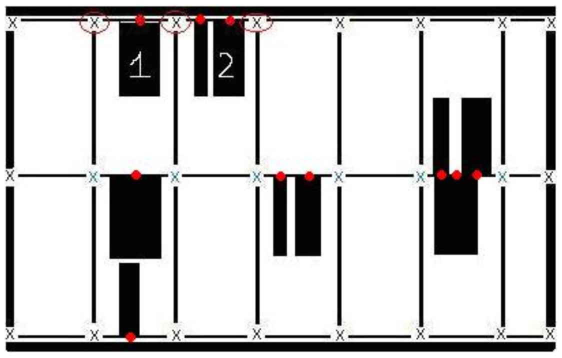

Figure 4. Discrete grid model for power lines.

the outer rings, the grid stripes and the standard cell wires in standard cell technologies). Fig. 4 shows a mock-up power grid and the corresponding discretized grid points marked with an $\times$.

We also consider that there is a unique point connecting the cell to the power grid (despite the fact that these standard cells are connected to the power grid through a small region of abutment). This unique feeding point, for each cell, is located in the middle of the abutment region and is shown in Fig. 4 by a round point at the top/bottom of each cell. It is also assumed that each cell that becomes active, impacts the discrete power grid at only two points. These two grid points are on the left and right side of the cell feeding point. Note that the cell position between these two points is also important, since the closer the cell is to one of the points, the higher the impact it causes over that particular point and the lesser impact is caused over the other. In Fig. 4 we see that cells 1 and 2 both contribute to the same grid point between them. So, if these two cells are active the impact caused in this grid point is higher than in any other.

In order to determine the region with the most impact, a spatial window, with user defined dimensions, is moved along the die. In each window position, all the grid points covered by that spatial window are considered in order to identify the region that has suffered the most impact from active cells. Associating the spatial information to the time instants/intervals, determined in the previous section, one is able to identify the worst timing/spatial region from the power grid perspective.

4.3 The PGinpGen Tool

A software tool that implements the additional task described in Fig. 3 was created and named PGinpGen. It was written in C/C++ for efficiency and was built over OpenAccess Gear Timer [8]. This tool receives the circuit information from the OpenAccess database, reads the technology library, .lib file, and accepts constraints from a .sdc file. PGinpGen then generates a report describing the result of the analysis of the circuit power grid.

The algorithm implemented by PGinpGen has the following steps:

1. Create a discretized model of the circuit power grid using physical information from the OpenAccess database.
2. Determine the time instants when each cell has the possibility of becoming active and inactive, for each

type of output transition (rise and fall). To that end, PGinpGen uses information from the .lib library characterization file to calculate the specific delay for each cell and uses a STA traversal algorithm to compute all the relevant instants/intervals.

3. Sort all the four lists of instants obtained in the previous step.
4. Determine the current timing window. Use the sorted information to know which cells may be active within this timing window. This is done incrementally within the lists by analyzing only the cells that changed their active/inactive status.
5. For each cell that becomes active during a timing window, its impact over the grid points is added (incremented). For each cell that became inactive during this timing window, its impact over the two closer grid points is subtracted (decremented).
6. For a timing window, the power grid is analyzed using a spatial window. While this spatial window moves along the die, the impact of active cells over the power grid points inside the window is observed. For a contribution with a maximum impact, the information about the corresponding timing and spatial window is updated.
7. While not all activation instants have been considered, make the timing window slide through time and go to step 4.
8. As a final result PGinpGen determines which timing window has the greatest impact over a certain region of the power grid. It also knows which cells are responsible for that impact.

Note that PGinpGen reads the netlist and placement information from the OpenAccess database. So before running the tool, the design must have been previously placed & routed, and must have a regular power grid defined. Generating the associated stimuli for simulation simply requires translating each possible cell switching within the given interval into an appropriated stimulus for the simulator.

4.4 Complexity Analysis

This section analyzes the computational costs for the algorithm implemented in the PGinpGen tool.

PGinpGen starts by reading the power grid description from the OpenAccess database. It converts the physical description into a discretized model, as mentioned in step 1 of the algorithm described on the previous subsection. All the points of the discretized model result from the intersection of the several power grid physical structures. Therefore this operation takes a run-time that is proportional to the number of discretized points:

$$O(\#stripes \cdot \#standardCellWires) \tag{1}$$

where $\#stripes$ is the number of vertical power grid lines and $\#standardCellWires$ the number of horizontal power grid lines, according to the orientation shown in Fig. 4.

Circuit traversal and creation of time instants/intervals for each cell (step 2) can be estimated as:

$$O(\#cells) \tag{2}$$

Table 1. PGinpGen vs normal STA run-times.

$Circuit$	$\#\ Cells$	$STA\ (s)$	$PGinpGen\ (s)$
s27	23	0.15	0.16
s1196	945	0.70	0.87
s5378	2680	2.47	3.08
s13207	6084	14.83	16.37
s35932	24732	139.59	142.33
s38417	20872	77.19	82.02
s38584	27150	68.17	73.90

where $\#cells$ is the number of logic gates in the circuit. Note that each cell is visited only once per traversal and we are considering that the number of intervals processed per cell is limited by a constant number (in particular if we consider the merging of instants/intervals as described before). This is clearly an approximation, but is a fairly acceptable one assuming the circuit was well designed.

The sorting operation over the four rise and fall transition lists (step 3) is done using quicksort and takes (in the average case):

$$O(\#intervals \cdot \log(\#intervals)) \tag{3}$$

where $\#intervals$ is the total number of intervals that need to be sorted, which given the above approximation, is proportional to the number of cells in the circuit.

The intervals processing time (steps 4 to 6) is proportional to:

$$O(\#timingWindows \cdot \#spatialWindows) \tag{4}$$

where $\#timingWindows$ and $\#spatialWindows$ are the number of timing and spatial windows that will be analyzed.

Note that the user can influence step 2, 4 and 6 by changing some PGinpGen input parameters. However, considering all steps of the algorithm, the time PGinpGen takes to analyze a single circuit is bound by the sorting operation and is proportional to:

$$O(\#cells \cdot \log(\#cells)) \tag{5}$$

Observe that the proposed algorithm has a small computation overhead when compared with traditional STA algorithms that run in time proportional to $O(\#cells)$.

5 Results

PGinpGen was run on a Pentium IV at 3Ghz, with 1GB of RAM memory. It was tested over 7 benchmark circuits from ISCAS89 (these circuits were mapped into a standard cell library with three types of cells, namely AND, NOT and D-type FlipFlop). Table 1 presents the number of cells of each benchmark ($\#\ Cells$) and the run-times of the OpenAccess Gear Timer static timing analyzer (STA) and our tool ($PGinpGen$), when using a timing window size of $0.1ns$ and a timing increment of $0.01ns$ (the latter is used to discretize real time). The size of the spatial window was chosen for each circuit in order to have approximately a total of 40 spatial windows covering the die. As expected, we conclude from Table 1 that the modification of the static timing analysis procedure along with the use of spatial information has a small impact regarding run-time.

For each circuit, the worst timing/spatial window detected is shown in Table 2. The $Grid\ size$ column indicates the upper right indexes of the discretized grid model, the

Table 2. Critical region for each of the benchmark circuits.

Circuit name	*Grid size*	*Worst timing/spatial wind.* *Time (ns)*	*Region*	*Active Cells*
s27	[5,2]	1.8 to 1.9	[2,1] to [2,2]	43%
s1196	[14,14]	3.3 to 3.4	[5,0] to [5,4]	7%
s5378	[28,31]	4.6 to 4.7	[4,18] to [6,27]	8%
s13207	[43,51]	6.7 to 6.8	[19,34] to [22,51]	10%
s35932	[92,85]	2.1 to 2.2	[43,44] to [70,72]	18%
s38417	[93,85]	8.8 to 8.9	[57,58] to [84,85]	22%
s38584	[95,85]	2.9 to 3.0	[59,15] to [87,43]	15%

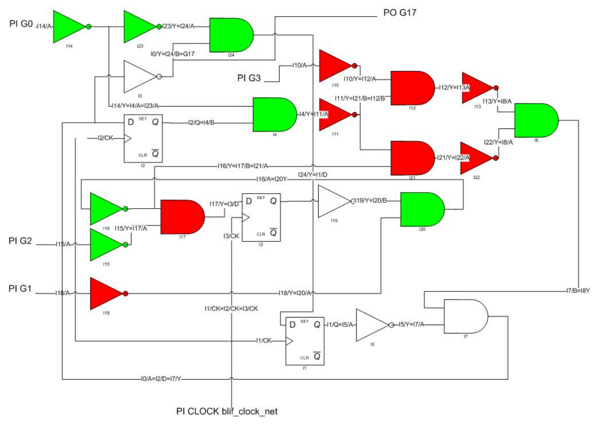

Figure 5. Logical description of the mapped s27 circuit.

lower left indexes being [0,0] for all circuits. The *Interval* and *Region* column indicate the critical timing/spatial region identified by the tool. Finally, the *Active Cells* column refers to the maximum percentage of active devices inside the critical region. Note that, in general only a small percentage of cells (less then 25% for larger circuits) are responsible for the worst case scenario. For the smaller circuit (s27) the analysis results are presented and discussed. In Fig. 5 the logical description of the s27 circuit is shown. After analyzing the s27 circuit power grid, the critical spatial/timing window is shown in Fig. 6.

As can be seen by comparing Fig. 5 and Fig. 6, almost all the cells that are detected by PGinpGen as capable of becoming active in the critical region are closely positioned in space (inside the spatial window) and in time (the cells are connected to each other). The cells that lay inside the critical region are represented in a darker colour in Fig. 5. Only the coloured cells may be active inside the timing window considered (1.8ns to 1.9ns).

6 Conclusions

We have proposed an extended design flow to include a more realistic method for generation of power grid stimulus. This method uses placement and netlist information to compute the stimuli for a power grid simulation flow, as shown in Fig. 3. A software tool that implements the method was developed and the result of its analysis over seven benchmark circuits was presented. The results showed that several assumptions done during power grid simulation are clearly pessimistic and may lead to costly over-design. Our tool was able to identify on which region of the power grid at a given interval of time there may be considerable impact

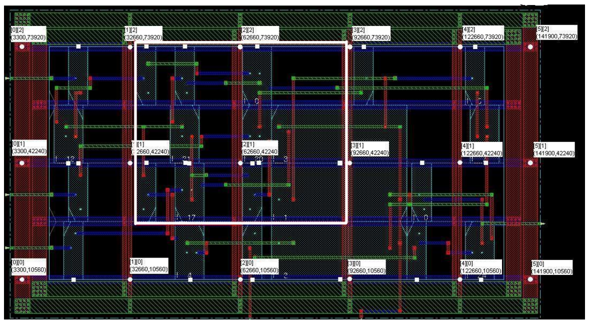

Figure 6. Critical region in s27 circuit power grid.

of simultaneously switching cells over the grid. It was also shown that this type of analysis is almost as fast as regular STA. These results should help the circuit designer to get acquainted with the circuit power grid and should be used priorly to power grid simulation. As future work we intend to use this information to generate the appropriate stimuli that can directly feed a power grid simulator, such as VoltageStorm.

7 Acknowledgements

This research was supported in part by the Portuguese FCT under program POSI, ref.EEA-ESE/61528/2004.

References

[1] Power grid verification. Whitepaper, Cadence Design Systems, Inc., 2001.

[2] S. H. Choi, B. C. Paul, and K. Roy. Dynamic noise analysis with capacitive and inductive coupling. In *Proceedings of the Asian and South-Pacific Design Automation Conference (ASP-DAC)*, pages 65–70, January 2002.

[3] T. H. Krodel. Powerplay-fast dynamic power estimation based on logic simulation. In *ICCD '91: Proceedings of the 1991 IEEE International Conference on Computer Design on VLSI in Computer & Processors*, pages 96–100, Washington, DC, USA, 1991. IEEE Computer Society.

[4] S. Lin and N. Chang. Challenges in power-ground integrity. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 651–654, San Jose, California, U.S.A., November 2001.

[5] S. R. Nassif and J. N. Kozhaya. Fast power grid simulation. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)*, pages 156–161, Las Vegas, Nevada, U.S.A., June 2000.

[6] T. Sato, D. Sylvester, Y. Cao, and C. Hu. Accurate in-situ measurement of peak noise and signal delay induced by interconnect coupling. In *Proceedings of the International Solid-State Circuits Conference Technical Papers (ISSCC)*, pages 226–227, February 2000.

[7] H. Su, S. S. Sapatnekar, and S. R. Nassif. Optimal decoupling capacitor sizing and placement for standard-cell layout designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits (TCAD)*, 22(4):428–436, April 2003.

[8] Z. Xiu, D. A. Papa, P. Chong, C. Albrecht, A. Kuehlmann, R. A. Rutenbar, and I. L. Markov. Early research experience with openaccess gear: an open source development environment for physical design. In *ISPD '05: Proceedings of the 2005 international symposium on Physical design*, pages 94–100, New York, NY, USA, 2005. ACM Press.

[9] Y. Zhong and M. D. F. Wong. Fast algorithms for ir drop analysis in large power grid. In *ICCAD '05: Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 351–357, Washington, DC, USA, 2005. IEEE Computer Society.