Substrate noise full-chip level analysis flow from early design stages till tapeout

Hagay Guterman, CSR

Jerome Toublanc, Ansys
**Abstract**
As SOCs integrate more analogue and RF IPs together with noisy fast digital blocks and interfaces, substrate noise risks increases. Traditionally chip-level substrate noise analysis takes place just before tape-out, when it’s too late to make the required modifications without risking the schedule. We suggest a flow which utilizes standard substrate analysis tools and starts very early in the chip design stages. Even with very basic chip information, substrate noise analysis can provide valuable data. The models of noise generation and noise propagation through the substrate can develop in parallel to each other when relevant data is available.

**Authors Biography**
Hagay Guterman is a senior signal and power integrity engineer at CSR. He has prior experience as an analogue and circuit designer. He presently works on signal integrity in highly dense mixed signal ICs inside and outside of the DIE.

Jerome Toublanc is Principal Product Engineer providing technical support in Europe and driving the direction for ANSYS-Apache's Power and Noise solutions for full System Integrity. Prior experiences focused on different SoC implementation tools such as Place & Route or Power Analysis as well as analog/digital full-custom design.
Introduction

Today’s SoCs integrate analogue/RF IPs together with noisy fast digital blocks and interfaces more frequently than before. Technology and features development in many SoCs (System on a Chip) integrate large digital blocks, operating at higher frequencies than ever before together with sensitive high frequency, low noise analogue and RF IPs. Process development, high performance requirements and manufacturing, assembly and material cost constraints often drive these elements to be in close proximity on the same die. The combination of these trends introduces, or aggravates, several risks and issues which need to be addressed. One of these issues is *substrate noise*. While having been a concern in the past, it now requires greater attention.

Substrate noise analysis is traditionally handled in different levels of the design: specific structures, block/IP levels and top or chip level. Only in top/chip level can we evaluate the effects of substrate noise between different elements and blocks in the design. Since usually the IPs and blocks layout, placement and connectivity are finalized only shortly prior to the tapeout, traditionally chip level substrate noise analysis occurs only then. Many times substrate noise related issues can only be resolved by making dramatic changes: floorplanning manipulations, layout modification, isolation structures or functional modification. By the time the substrate noise is analyzed, it is too late to make such dramatic changes without risking the project’s schedule. This usually introduces a tradeoff between the project schedule and chip performance.

In this work we are suggesting a method to evaluate substrate noise risks and effects very early in the project design stages. We are introducing a flow of work, using standard and proven substrate noise analysis tools, built in phases alongside the project’s development. The first point we make is that it’s never too early to evaluate substrate noise. Even in the most preliminary design stages, evaluating substrate noise can provide useful insight. The phase structure can be flexible, according to the nature and character of the project development. From phase to phase the substrate noise picture becomes clearer and better correlates to the final design.

Another useful benefit of this flow is the ability of comparing substrate noise between different architectures or designs. It provides a way to generate a platform suitable for substrate noise evaluation without actually having a real design which requires considerable effort. This way alternative floor plan, guard structures, package connectivity, etc. can be simulated and compared in terms of substrate noise.

Background

“Substrate noise” refers to the noise conducted across the SoC through the substrate network. The switching elements, i.e. the “aggressors”, inject noise that couples to the substrate through wells connected to the power rails. The substrate conducts the noise through its wells and bulk layers. At the sensitive circuits, the substrate noise couples to the victim devices or rails.

The chip’s package has a considerable impact on substrate noise. Increasing package impedance increases aggressors’ voltage swings, thus increasing the noise injected into the substrate. It also reduces noise absorption by isolation structures and increases noise conduction. Large impedance in the package can also reduce noise absorption by wells connected to rails in the victim area. Therefore, substrate conducted
noise has to be fully analyzed in a full chip-package context. Figure 1 shows the overall Noise conducted paths.

![Combined Noise Model](image)

Figure 1: Combined Noise Model

Analyzing full chip substrate noise naturally involves on-die substrate network modeling. The solution requires comprehensive noise injection capabilities, noise propagation simulation and layout-based results exploration.

The substrate modeling depends on the process. For instance, image sensors, automotive SoC, lighting-control or any high-voltage analogue/mixed-signal (A&MS) will use different processes such as bipolar, CMOS, DMOS, BCD. Each of these requires specific extraction rules based on foundry’s specifications.

Many designs with sensitive analogue circuits use a triple-well CMOS process. Figure 2 illustrates a simplified substrate cross-section for the typical p-bulk process. “Triple-well” refers to the three main well types: shallow n-well, shallow p-well and deep n-well. Same-type wells connect resistively; opposite-type wells connect capacitively. Deep n-well is primarily used for DC isolation of nFETs from bulk, but it also acts as a substrate isolation structure. It reduces noise injected by an aggressor by shielding the p-wells containing its nFETs from substrate bulk. It provides additional decoupling capacitance between aggressor power and ground rails.
Additional isolation structures, such as guard bands, usually surround sensitive circuits to break surface conduction paths, absorbing or even blocking some of the noise. Figure 3 illustrates the main isolation structures. These should be part of the simulation to validate their efficiency.

Analyzing Substrate noise also implies complete Chip-Package modeling. The on-die voltage transients depend on the parasitic (resistivity, inductivity and decoupling) of the complete power and ground delivery network. Bond wires and packages contribute with their series inductance between on-die nets and AC grounds. The resistivity of the on-die grid relies on the topology and the number of dedicated layers for the SoC meshing. The overall decoupling mainly comes from capacitances in the package and intrinsic standard cells decoupling, such as decap cells.

From a data input point of view, the solution has to be comprehensive to import a mix of standard formats: victims IPs description usually comes in GDSII format, whereas the top level is described in a LEF/DEF format. While the GDSII file includes all the layout layers (metal, substrate, contact, diffusion), this is very often not the case for the LEF/DEF data; most of the time, the digital Nwell and Pwell structures are missing and require extra setup to be added in the simulation.
Early Substrate noise analysis and Flow

Analysis data inputs

Signal and power integrity analysis simulations usually require knowledge of the specific structures and layout used in the design. This is obviously very true when discussing on-die signal and power integrity in top chip level. Substrate noise analysis can be divided into three parts, each with its own inputs: Noise generation, noise propagation and the impact on the victim. The victim is typically a sensitive analogue or RF IP, often integrating a ground-bounce sensitive circuit (e.g. VCO). The level and profile of noise that the IP can tolerate depend both on its design and the specifications it is aiming for. It is therefore preferable that the analogue designer is provided with a noise waveform at predefined locations within the IP. The analogue designer can then simulate the effects in a circuit simulator to evaluate the performance under these noise conditions. The flow in this paper focuses on the two other stages in the analysis; substrate noise generation and noise propagation throughout the chip. One major outcome of the flow is a waveform of substrate noise at the victim, to be forwarded to the analogue designers for performance analysis, in order to indicate what risk the expected substrate noise poses towards the chip’s performance.

Some of the data required as input for the analysis is generic and relates to both portions of the analysis. The main required pieces of information include the technology used (process), the metal stack and the environmental conditions (temperature). Knowing the process is an elementary piece of information, as each process flavor carries its own substrate characteristics for noise attenuation, as well as for noise generation at the source elements (aggressors). The metal stack indicates the amount, type and order of metals used in the die and also their thickness. Much of the noise generated at the substrate of the aggressor, or reaching the victim’s substrate, actually propagates through these metal rails. Each project also has its own technology settings which indicate the types of vias used, metal widths and spaces, etc.

Beyond this point the inputs split between Noise-Generation requirements and requirements of the Noise-Propagation throughout the chip.

In order to model noise Generation, it is required to know the layout of the aggressor (whichever internal block it is). Since a specific block may generate different levels and profiles of noise depending on its operation, it is also important to know its activity mode. Dealing with digital aggressors is quite different than dealing with analogue or I/O cells as aggressors. For digital aggressor, one would need the layout of the digital block (DEF, LEF views usually), as well as its post place-and-route netlist (e.g. Verilog). The block’s netlist dictates its behavior while the activity is defined by a set of vectors (vcd). For an analogue, RF or I/O aggressor the layout and netlist are also required (GDS + CDL), but the activity is extracted using circuit simulations.

Noise Propagation inputs include Floorplan, Power Mesh, Tap location, Power Pads Location, Package characteristics, Netlist and layout of hard macros and I/O cells. ‘Floorplan’ refers to information indicating where each internal block is located inside the chip and its dimensions. A floorplan is specific for each chip and is the basis for any overall chip analysis. The Power Mesh structure corresponds to the metal rails of the power delivery network in the die, indicating for each metal layer the rails’ width, space, offset and orientation. This specifies how the blocks and standard cells connect to power
throughout the chip. The Tap location indicates where the power connects (taps) into the substrate or wells. Both these parts of the Power delivery network in the die are important elements of the substrate noise, since noise rises where power connectivity is weak. “Good” power connection usually means that noise coming from the periphery of the local cell via the substrate has a smaller influence and may even die out at the edge of the cell. Since the power source usually originates outside of the chip, it is also important to have information on the power delivery network into the die. Power pads location and Package characteristics will provide the required information.

When the above information on the environment is available the next step would be to describe the path in which the noise propagates through. Detailed description of the die is given by its layout, from which the parasitic electrical properties can be extracted. However, for simulations we also require the netlist in order to correctly connect the supplies. For the I/O cells, Analogue and RF IPs we require the layout and connectivity as well, as noise may propagate through them. It is doubly required for the victim Analogue IP.

**Traditional Chip level Analysis**

Normally much of the input information described above becomes available in a usable format (e.g. DEFs, LEFs, GDS, package extraction, pad location, etc.) only during the final stages of the chip design. Therefore, traditionally top-level chip substrate noise analysis is done shortly prior to tape-out, often even after tape-out if it is not considered a “gating” mile stone for tape-out. Many issues of substrate noise can be resolved by changes in the design. Some require floorplan changes, e.g. putting more distance between aggressor and victim. Some issues can be resolved by block design modification, e.g. changing RF blocks to be more robust towards substrate noise. Sometimes this includes modification of the power scheme and adding additional ground or power nets to isolate certain devices or blocks. Layout modifications are also a common solution. These may include isolation structures like guard bands, addition of deep N-wells, etc. In some cases the solution might actually come from changing the architectural operation of the system. For example, preventing a digital aggressor from operating while a sensitive RF receiver is active, modifying the operational frequencies of blocks so that their harmonics do not coincide with each other, spreading the spectrum of clocks artificially to even out energy across the band, etc. Some of these modifications require a considerable amount of work and may require a re-spin through many of the design changes. If this is discovered only during the final stages of the project design, it poses a dilemma of either risking the project’s schedule or its performance. The flow described in this work intends to minimize some of these risks, by making the analysis earlier in the design stages.

**Early Analysis**

The purpose of substrate noise early analysis, as introduced above, is simply to evaluate substrate noise analysis much earlier in the design stages. More precisely, the goal is to be able to face decisions concerning substrate noise risks in early stages, while the design modifications required to solve issues cost much less in time and effort.

The main challenge lies in the generation of a platform for substrate noise simulation of the chip, before the actual layout of the chip exists. We are taking advantage of the fact that even though in early stages the design teams have yet to
produce usable views of the inputs required for the analysis, much of it is known in theory. For example, the floorplanning of the chip is set by the architects quite early, but there’s no layout description for it (e.g. DEF). Another example is the power mesh architecture, which is usually chosen before work on the blocks starts. There is no layout of it, but the guidelines exist. Layout of the analogue/RF macros may be available even before the project design starts, at least in a preliminary version. Much of the remaining required inputs can be evaluated based on previous experience. This is especially true if there’s a previous chip with similar size, type and package characteristic. Package RLC, power numbers, number of supply pads, etc. can be extrapolated from previous designs.

The idea of early analysis is to generate the layout description views of the chip (DEF, LEF) based on theoretical information. This can be done either manually using a design tool or automated. Automation is mainly beneficial for the repetitive structures found in the digital area and for accuracy and consistency purposes. Repetitive patterns can handle power mesh generation, standard cells internal structure, tap locations, etc. The automation can be implemented by proprietary scripts or standard analysis tools. Once the theoretical data is collected it must be formed as rules or elaborate instructions for the automation tool. For example, the power mesh generation rule indicates to the tool the width, space, offset and orientation of the rails in each metal. The automation tool will then execute the rules to deliver a mesh structure in a generated DEF file. This principle can be applied to most of the required information described above. The outcome is an artificial design which emulates the actual chip from substrate noise analysis perspective.

Although during the first stages of the design the information required for the analysis input is incomplete or uncertain and much needs to be estimated, a lot of conclusions can still be deduced from the early simulations. For example, let’s assume that at the beginning of the design stages already the locations and layout of the analogue victim is known, and there’s a basic idea of the floorplan and the types of blocks around the chip, but there no more inputs are available. Even with this little amount of information, an artificial chip, which includes a rough image of the floorplan, can be generated. The digital blocks can be filled with wells according to the standard cell’s row structure and a power mesh with tap points can be added according to the known guidelines. Noise can be injected “manually” in a location in the design where the plan is to place a noisy aggressor. The noise can be any waveform of interest. For example sine, step, piecewise linear (PWL), etc. This simulation can shed light on the magnitude of noise levels which can be expected to reach the victim from the aggressor and the attenuations per frequency in the path. It can also allow comparison of noise levels at different locations to optimize the floorplanning.

**Analysis flow**

As mentioned above, the maturity of the input data determines the capability of the early analysis to emulate the chip reliably. The more the database becomes comprehensive and stable, the better its simulation results will correlate to the final design, up until the point when it is complete. It is therefore useful to make the substrate noise analysis in a flow of phases. In each phase the database used is a step closer to the final design and fewer assumptions or generalizations are required. As indicated above, the inputs split between the two parts of the analysis, noise propagation and noise generation.
For the noise generation, it usually takes some time into the project cycles before there are enough details on the aggressor in order to simulate it fiducially. However, the location of the aggressor block is known much earlier than that, and so are its frequencies and rough structure. This is often sufficient for injecting noise “manually” at a location of interest and set its frequency and profile according to expectations. Analysis of the impact of this noise may provide early indication on noise robustness for this floorplan and aggressor. Later on, the aggressor layout information might be available but it is not yet placed in the chip top level. For example, an I/O cells layout is known, but is not yet placed in the design. The I/O cell can be simulated stand-alone to evaluate the noise it generates which can be in turn injected at its intended location into the artificial environment created beforehand. This would emulate more reliably the noise profile at the source than a sine or PWL waveform. Further down the line, when the aggressor design is complete (the entire I/O interface in our example), it can be fully instantiated in the analysis environment to produce an even more exact representation of the noise source.

Noise propagation related input data tends to arrive in a similar manner as the generation related inputs. At the beginning, the entire chip needs to be artificially generated as described previously. Stage by stage, whatever is known regarding the specific chip is directly implemented. The rest of the data needs to be generated either based on past experience or by analytic estimations. As the project develops, more information regarding the chip becomes available and can be fed into the analysis environment to replace the earlier rough models. For example, the number and location of supply pads can be estimated fairly early based on the die dimensions and the power estimations. Later on the pad list becomes more stable and can replace the initial estimations.

The inputs of the two portions of the analysis may develop independently of each other since they depend on separate bits of information. Constructing the analysis phases so that each portion advances on a separate track, with its own rate, adds a measure of flexibility to the flow. This way the flow becomes “opportunistic”, i.e. a new phase can start as soon as a fresh piece of data turns up to fine-tunes the analysis a step further. The transition from phase to phase can include either progress in both portions or only one. It often makes sense to attach the analysis phases to the project design milestones. Often project milestones indicate certain levels of maturity of design elements required for the analysis. This assists in creating a realistic schedule for the analysis as well. For example, the dry-run milestone traditionally requires some stability of the I/O wall, hence supply pad location is available. It also means that the digital blocks are starting to achieve their final shape, area and location. The analogue blocks’ abstracts should have been finalized by then and usually at least a preliminary layout of them is available. The planning of the flow phases should be done by considering both availability of the inputs, analysis goal, required schedule and resources.

Figure 4 demonstrates an example for phase construction and its flexibility along the project development. The plan (left columns) is constructed soon after the project kickoff. During the kickoff a schedule is set for the project’s milestones, indicating, among others, the due dates of the first netlist, Dry-run netlist and the final netlist. These construct the timeline for the noise generation portion. A separate timeline for the noise generation determines when the aggressor (I/O interface in this case) is available; stand-
alone cell, complete interface and activity circuit simulation. Once the two timelines are set, separation for phases of analysis can be defined. In this plan 5 phases are defined plus one optional. The first phase, titled initial, is the basic early analysis. The next phase (A) was chosen to include improvement in the propagation portion; the integration of the layout of the I/O cells. Phase B is a modification in the noise generation modeling, where the PWL initially used to emulate the aggressor noise is replaced by a stand-alone I/O pad simulation output. Phase C includes modifications in both portions. The generation is progresses to become a complete interface circuit simulation and the propagation platform advances to be constructed of real PNR design, based on the first netlist. Phase D is a placeholder in case there’s enough time to conduct analysis on the dry-run database, before the final database is ready for the final phase.

![Figure 4: Phase planning and flexibility in execution example](image)

During the project development some changes to the schedule and plan may occur. In this example (right columns of Figure 4), by the time the I/O pad simulation was available the layout of the cells was ready for the propagation as well, so phase A became obsolete. The simulation of the full interface became ready before the first netlist arrived, so it was decided to add a phase between B and C. After phase C the work on the dry-run was delayed, until it became redundant to simulate it, and instead the flow proceeded directly to the final design.

Some of the phases can be done with a few sub-revisions, i.e. though the inputs from the project remain the same, the manipulation in the analysis is slightly different. By taking advantage of the fact that in the early stages the design is artificial, we can make modifications to the design that have not occurred (or yet occurred) in the actual design,
this experimenting in alternative designs. The next chapter demonstrates some possible usages of this principle with several examples.

To explain the flow further let’s examines an example from an actual SOC. The majority of the chip hosts digital logic including a large CPU. In addition the chip houses several analogue IPs, an RF IP and a noisy fast digital I/O interface. The main substrate noise risk is assumed to be the one generated by the noisy digital I/O interface and received by the RF IP. The concern was mainly due to one of the interface’s harmonics overlapping the RF IP’s band.

The initial phase was entirely conducted in an ‘early analysis’ fashion. The information acquired from the architects included the process flavor, metal stack, chip dimensions, general description of the floorplan, power mesh architecture, blocks’ power estimations, standard cells library and I/Os to be used. The RF block layout was available at the get-go. The number of supply pads was extrapolated given the power estimations and the experience from previous chips. Package RLC figures were taken from a previous similarly sized chip. The noise generation for the initial run was a sine wave injected at the approximate location of the fast digital I/O interface, close to the chip south wall. Figure 5-A illustrates the chip’s maximal IR drop map on the Pwell and Nwell layers.

It is visible how the noise is attenuated as it propagates across the chip towards the RF block in the middle of the north wall. It is also clear how the power pad connections serve to damp the oscillations and serve as a current source for the disturbance on the expense of voltage drop. The noise attenuation calculated from voltage probes’ amplitude between the PW area at the noise source and a point of interest at the victim is -72dB. At this point the absolute levels of the noise are not accurate and serve more as reference for comparison and displaying trends, as portrayed in the next chapter.

The next phase included pad numbers and locations from the actual design, together with more realistic package figures. The number of pads decreased from the initial plan, and the effect is visible in Figure 5-B. At this point a review of the results vs. the cost made the compromise reasonable. This is an example of how the early analysis flow can be integrated into the decision making process during the chip design, while it’s still possible to make modifications.

The next phase included an improvement in the noise generation by injecting a waveform which was the result of a circuit simulation of an I/O cell. Figure 6-A presents the waveform of the noise injected at the aggressor. It can be observed how the voltage

![Figure 5: Substrate noise induced Voltage Drop given different supply pad locations](image-url)
drop is highest in the digital signal transitions, where the current consumption is high and in the high harmonics. The inductance on the package doesn’t allow enough current to reach the well, and the drop is high. Figure 6-B is the noise at the victim, after it propagated through the complex impedance network of the chip. It can be observed how the waveform has been distorted, since the attenuation is frequency dependent. Figure 6-C presents the noise inside a deep NWell in the victim. Figure 7 presents the approximate transfer functions of the propagation through the chip (A) and the DNW interface (B). They are mere approximations in this case, since they are based on a time domain simulation with a specific input (rather than a real frequency domain analysis). These two figures illustrate how unlike the path through the chip, the deep NWell attenuates high frequencies less than low frequencies. This is expected, since the impedance of the deep Nwell interface is mostly capacitive in nature. This sort of research provides a better understanding of the noise profile which may be observed in the victim in reality.

The next phase in this case was the final phase, which included the real tapeout database.

**Applications**

The advantages of the early analysis are not confined to the suggested flow. The ability to easily generate platforms for chip level substrate noise analysis allows comparing different design options and evaluating their substrate noise robustness in comparison.

A chief example is comparing alternative floorplans. Using the early analysis method described above one can generate alternative floorplans and compare the
substrate noise profiles between them. Using the same design of the examples in the previous chapter, this can be demonstrated. Figure 8-A displays the substrate noise induced voltage drop map in the original floor plan, where the RF IP was placed in the middle of the North wall. Figure 8-B displays a floor plan where the RF block was placed on the East wall.

Figure 8: Victim location variations

It can be observed that the noise levels increased. The attenuations calculated from probes in the victim and the source show deterioration from -68dB to -63dB, i.e. 5dB difference. This comparison provides a sense of the effect of the victim’s position, in case alternative floorplans are evaluated.

Figure 9: Aggressor location variation

In another experiment the aggressor is shifted. Figure 9 displays the map when the aggressor is located differently in each. The noise injected is identical, and the effect of the location and distance to the victim can be observed by the waveforms of the noise in the same location within it.
Another possible usage for the method is comparing alternative substrate isolation architectures. For example, Figure 10 displays the noise levels in the original design, while in Figure 10 guard bands, composed of P-diffusion and metals tied to ground, were added on the boundary of the RF block.

The guard bands caused the noise to equalize along the IP’s boundary and lowered the noise levels which penetrated it. In Figure 10-A the RF block has no deep N- Wells in its design, while in Figure 10-B deep-NWells were added beneath the sensitive areas and tied through N diffusions to a supply.

As expected, it is visible that the Deep-Nwells added considerable attenuation between the aggressor on the outside of the deep NWells and the sensitive circuits within (marked in an ellipse). Calculating the attenuations from probes at the aggressor and the victim shows an improvement of additional -14dB. Although the noise levels measured in the aggressor are still far from accurate at this time to real account for absolute levels, the comparison between the alternative isolation options correlate quite well to the actual figures (as is conveyed in the next chapter).

Many other comparisons and feasibility studies in terms of substrate noise can be conducted this way. It’s important to keep in mind that ‘the output is always as good as
the input’, i.e. the prediction is as good as the measure of accuracy in which the setup resembles the final design. If the final design turns out to greatly differ from the early analysis platform, so will the absolute results. However, comparative measurements always show great accuracy.

**Correlation to final analysis**

Previous works have shown how traditional full-chip substrate noise analysis correlates to post-tapeout measurements [1]. We therefore assume in this work that there is a good correlation of the noise at RF frequencies (dBµA) between measurement and full-chip simulations.

The focus here is to examine the correlation between the early analysis and final layout database based analysis. For this task we used the same SoC used in the examples above and compared early analysis to final place and route database. For the early analysis we generated the floorplan, power mesh and the wells in the standard cell designated areas from scratch, based on our knowledge from the actual design. The RF block’s actual layout was placed in its location and the power pad locations were made as similar as possible to the final design’s. The noise in both platforms for this correlation work is a sine waveform, injected close to the south wall, where the high speed digital interface (suspected aggressor) is placed. The sine waveform was chosen simply because of simplicity in visibility and performance calculations, though the same performance was observed using more authentic waveforms.

Figure 12 displays the noise map (voltage amplitude) of the full chip in the early analysis (A) and the real P&R dB (B). From these maps we can see that the form of the noise propagation and it’s profile correlate quite well, though the absolute noise levels have some offset. The main reason for these changes is because some of the digital design is not composed of standard cells, but rather from memory cells (RAMs and ROMs). These cells have a different profile of wells, which cause differences in the impedances along the noise propagation path. The internal well structure of the memories introduces additional elements into the path, mostly capacitive. The amount, location, size and internal topology of the memory blocks are usually unknown during the early design stages, so we did not add them to the early analysis.

![full chip noise maps](image-url)

**Figure 12:** Full chip Noise maps (same scale 0-10mV)
Figure 13 is a zoomed window of the voltage map, with refined scale legend. The offset of 10% (0-0.9mV for early dB versus 0-10mV for P&R dB) produces similar image. It is visible that given this offset, the noise’s profile within the RF block in the early analysis correlates quite well to the final design’s. The offset is again due to the impedance differences of the path between source and victim.

![Figure 13: Voltage Noise maps: zoom in victim area (adjusted scales)](image)

The offset in noise level at the victim can be observed in the waveforms in Figure 14, which shows the voltage over time at a victim’s ground node in the early analysis versus the final design. This point of interest is a PW node located within a Deep-Nwell area. The shape of the waveform is the same in both, while there amplitude offset can be clearly seen.

![Figure 14: PW Voltage Waveform in DNW area within victim IP](image)

The table in Figure 15 is an overview of the differences between the two environments. The effects of the differences in the impedance of the top level path between the injection point and the victim can be observed. Even with the offsets, the ability of the early analysis to predict the noise voltage amplitude and its attenuation is quite good.
It is important to note that the offset, related to memory cells, may not be the same in a different chip. Each chip has a different area percentage and location for the memories, and thus the changes in the noise propagation path may differ from chip to chip. It is also possible to integrate the well alterations caused by the memory cells in an early analysis platform if there is information describing them. For example, a second step of a chip, with limited modification to its memory setup.

**Summary**

Early substrate noise analysis is essential in order to acquire understanding of the substrate noise issues and profile, while there’s enough time to make modifications in the design. Implementing the suggested flow provides a constantly updated picture of the substrate noise status as the chip develops and the design matures. This technique also allows evaluation of alternative designs (floorplan, isolation, and package) from substrate noise point of view without the need for constructing a costly real test design. The validity of the results in the early stages is useful mostly to understand trends, but if the actual design parameters are known, the technique can provide results with good correlation to an actual real layout based analysis.

**References**


Figure 15: Some details of Noise amplitude and attenuation comparisons between Early and P&R analysis