1. Introduction

Thank-you Yami Tuscon. So today in this tutorial we are going to talk about Vmin which has actually always been an important topic but I think probably the reason why you guys are all here is because in recent years it has become very important because of all of these challenges that we face with power decipation today.

It is also a tricky topic because it involves so many different mechanisms coming into play that all together interact to determine Vmin.

So my hope with this tutorial is to walk through at least my own perspective of all the different constraints and all of the issues that you have to think about when you look at Vmin for a given chip at the end of the day. I also want to talk through so that we understand where everything comes from and then to talk through circuit techniques so that we might be able to use them as circuit designers to try to make the situation better.

2. Outline

So first a quick outline of my talk. I want to first define Vmin so that we are all on the same page here and talk about how Vmin relates to energy efficiency, the needs that we have in our products today and also the trends going forward for our products of tomorrow. Then I want to go through them one by one and try to categorize the constraints that we might place on Vmin and so I want to talk through frequency limited constraints, functionality constraints and reliability. These are probably more traditional things that we as circuit designers might think about. I want to spend some time talking about power delivery which is something that circuit designers don’t normally think about because it plays a pretty significant role when we are talking about Vmin. Finally, I wanted to look a little bit ahead and talk about energy efficiency and whether we want to call this a constraint or not (we can argue about that), but it will impact going forward into the future and where we want to target our Vmin as it relates to the other constraints that we have on the board. Then finally I will try to conclude and I will try to do this all in 90 minutes.

3. What is Vmin

So I want to start first with a definition because Vmin, sort of minimum voltage, is something that has been used in a whole lot of different contexts. It simply says minimum voltage. In this tutorial we are mainly going to think about Vmin in the context of the minimum supply voltage that can be used for a given VLSI chip.
So if you have say a micro-processor or a micro-controller or something you want to use the lowest voltage possible to make that chip still work at your target specifications. Because That will be the lowest power operating point.

So as circuit designers if we want to make a power efficient part we have to understand where Vmin comes from and to also know how to apply techniques to improve it. So that is the purpose to this tutorial.

I will first talk to the constraints and I will work through practical mechanisms that limit Vmin. These are more engineering type things rather then fundamental physics.

And then I want to spend some time talking about optimization. So I will walk through circuit techniques and each of the different types of constraints just to give you references for when you go home and some of the big picture concepts that people are trying to do in each space.

So, the focus here will be on chip level Vmin. I want to take a big picture here and not get bogged down by looking at every little analog or IO circuit in your chip.

So the way I am going to think about this is to focus on digital logic and memory because when we look at a VLSI Chip most of the time most of the chip is going to be made up of logic and memory. So, the issues that I am going to cover in this tutorial are mainly logic and memory focused and so we are going to say that specific issues for say analog and IO circuits will be beyond the scope of this tutorial.

**4. Fundamental Vmin Limit**

So I want to start first with some physics and think about what the fundamental Vmin limit might be. So, this is not necessarily the practical Vmin limit but this is some work that Jim Meindl at Georgia tech published several years ago where he looked at physics and said “That ultimately at the end of the day what you need is for CMOS logic gain to be greater than one.” So in order to distinguish binary signals as zero and one, and in order to do binary computation, you simply need to make sure that the logic gates have enough gain. If you go through a calculation and look at the inverter transfer characteristic and you apply some transistor models to it then you can figure out the slope of the transfer characteristic and you can back calculate what the minimum voltage might need to be. And so what you figured out was that the minimum voltage was about 36 millivolts or so at room temperature which is clearly much lower than the voltages that we use today. We are probably more at 600 or 700 or above for most applications. So this 36 millivolts is obviously a very, very optimistic projection. The calculation assumed for ideal transistors and perfect electrostatics. The calculation was done on a CMOS inverter, obviously we used much more complicated gates today. It assumed no process variation which is a very big assumption there. It also had no performance targets to meet so this is only looking at CMOS logic gates that are actually doing their jobs and says nothing about what say an end user or customer might want to see. And so one way to think about this tutorial is that sure, maybe 36 millivolts is the fundamental limit, but I want to talk through all the reasons why we can’t achieve this 36 millivolt limit in practice.
5. Scaling Theory Revisited

So the reason why we can’t get there- it all relates to scaling theory and how we don’t follow scaling theory anymore and so I just want to go quickly over this because you all probably studied this in class and know this probably even better than me.

Scaling theory back in the day, as proposed, was this sort of this ideal scaling theory and the idea there was to try to maintain constant electric fields in your device as you went from technology generation to technology generation. So if you are able to scale the device dimensions down in accordance with pushing the doping concentration up and bringing the voltage down then you are able to get everything you might want out of scaling. You get your improved power and your improved performance and at a constant power density so you really have no power dissipation issues here and Robbert Dennard who actually still comes into work every day at IBM always tells me that we only really did this for a few technology generations and we deviated very quickly from this. So in the real world

6. Scaling is No Longer "Ideal"

when we tried to actually scale we realized that scaling isn't ideal and we have a lot of problems to face and deal with. I plotted here on the left the voltage that we actually used for technologies as a function of the gate length. This is actually gathered from different publications and things like that and you see that the voltage scaling has not really followed scaling theory at all especially in recent generations and it kind of flattens out around a volt or so and so in particular voltage scaling is one of our big problems. It does not fall into scaling theory because of a bunch of reasons. You have gate dialectic scaling limits due to quantum mechanical tunneling you can't make the oxide thin enough. Also you have sub-threshold leakage limits because the threshold voltage has to be a finite number, you can't go down to zero, and you have also a lot of process variations these days to deal with so you need larger voltage operating margins. So you look here at the transistor current equation you can already understand why we need to do this because at the end of the day we are always very interested in performance. And the performance is always going to be related in some way with your transistor drive current. So if you really have trouble scaling your tox and you have trouble scaling your VT. Dielectric constants and the saturation velocities are kind of material parameters that are hard to tinker with and what you are really left with is VDD. That is going to be the easiest knob to crank your current back up if scaling theory can't get you to where you want to be.

7. Increasing Power Density

And so the implication of this is that if you have a similar type of plot here, except this time we are looking at CV over I, which would be sort of an intrinsic transistor performance metric and I have also plotted the power density up top here so with CV over I you can see that it is actually following scaling theory pretty well. That is because performance has actually followed scaling theory and the reason for that is actually because we are all very good engineers.
We decided we wanted this type of performance and we have made all kinds of tradeoffs to get there and so you do see is performance improving but the tradeoff has been because we do not scale our voltages as much as we would like to.

So the power density goes way up and doesn't follow scaling theory at all. So this, I think we all know about and obviously power density cannot keep increasing because you start getting to very high temperatures and comparisons with rocket nozzles and the temperature of the sun. So obviously we need to start thinking about (specially right now) power dissipation being such an issue and how we can bring this power back down.

This is probably why you guys are all here to talk about Vmin to understand how much you might be able to bring the voltage back and so just to foreshadow on a little bit of what we will talk about at the end of this presentation- one of the arguments that I am going to make is with regards to our traditional focus that has been on CV over I types of things. We have focused on raw speed single thread performance and high frequency operation of our chips.

Going forward this focus might actually change depending on what customers will want at the end of the day so we will get into that in a lot more detail but overall we want to try to reduce the voltage as much as we can.

8. Challenges in Voltage Reduction

And so the challenges of scaling the voltage down, essentially those mechanisms that determine Vmin. I have tried to categorize a little bit here. We have performance issues in reducing the voltage and this probably has been historically the heaviest hitter here. That really raises your Vmin above that 36 millivolts by many hundreds of millivolts and that is simply because as you lower the voltage the frequency of your chip comes down and it is harder to get the performance and the drive currents that you need. Variability and operating margins are something that actually have become very important in recent years, maybe just as important in a lot of situations as performance. We all know about process skew, temperature, and layout density affects. Everything is going to vary in our process technologies today but really the fundamental limit that we are really struggling with, especially with memory, is random variation.

So we will talk a lot about doping fluctuation and line edge roughness here. I will talk about reliability because it is important and non-negligible margins have to be applied. I will talk a little bit about things like bias temperature and stability but I think I will focus the discussion actually on soft error rates as an actual Vmin limitation. Then I want to as I mentioned talk a little bit about power delivery because you know as designers usually when you run on spice?? you usually just draw a Vdd in a ground symbol and you assume that those are perfect rails. Obviously they are not and that will figure in pretty heavily into Vmin discussions. So hopefully I will convince you that practical issues actually dominate over the 36 millivolts of that fundamental limit.

9. Technology Dependencies
So most of this talk will be focused towards circuit design. But I want to emphasize that we really have to talk to our friends. We really have to talk to our technology friends in particular because there are dependencies on what they do and how we need to deal with Vmin. For example, for things like High-k/Metal Gate and FinFET, which is all the rage right now, provides devices that should be intrinsically higher performance. So if we are able to do that then obviously things like performance limited Vmin could become more relaxed. You can start dialing down the voltage a lot more because the intrinsic device is higher performance. These guys also usually give us reduced variability so it improves our variability margins there and I will also point out that we should always count on our technology friends (hopefully) to every generation and every couple of years to improve process control so while we as circuit designers worry a lot about variability it is reasonable to expect that they will come up with new ideas for example things like directed self assembly that may actually help to reduce the global and local variation. So we have to keep track of what our technology friends are doing because their progress will probably help us further reduce Vmin.

10. Application Dependencies

On the flip side we also have to talk to our systems friends. We have to talk to those guys who are actually dealing with software people and the customers because there are strong application dependencies of what we need to think about for the challenges and the solutions that we might apply.

For example, if you are worrying about a very low cost and low power part- let's say a wireless sensor node or something like that- where performance may not be that big of a deal but you really need high density, low Vmin and you need low leakage. You will probably play very different tradeoffs as compared to say a high performance server microprocessor where performance is really the key.

So the issues that we are going to talk about today are probably common across all different types of applications. But I will try to point out as we go along where different types of applications will emphasize different types of solutions and I will try to show you some examples of the different solutions and where they are actually used.

11. Factors That Determine Vmin

So before I go into the different factors that determine Vmin I wanted to draw a little power point chart here just to show you how it all fits together in context. At the end of the day with Vmin you simply need it to be sufficiently large to 1- maintain your functionality margin. You have to get your zeros and ones into the right place under variability and reliability. You also have to add some margins to that and you have to also maintain your performance target under variability and reliability. So, I am drawing here sort of a logic like example where I have shown that performance is actually more of a limitation then functionality. Also I will show you in the next chart how this might change for other blocks on your chip. In addition to that you also need to have some sort of a power delivery overhead to compensate for your delivery losses so that the voltage drops IR and supply noise due to inductance parasitics, LdI/dt, We have to put a little
margin there when you are thinking about $V_{\text{min}}$ and as I mentioned that this cartoon is related to something more like a logic block.

12. Factors That Determine $V_{\text{min}}$

If we start thinking instead about say a memory block then memory is usually these days much more limited by functionality than by performance. I have actually drawn them both to be the same in this particular case because we are good engineers and we improve functionality as much as we need to. So that it is not the limiting factor but basically the point here is that if we are looking at an overall VLSI chip most likely we are going to have both logic and memory on this chip. If the logic is limited by performance and the memory is limited by functionality then we are going to have to think about all the issues and we will be talking about these issues in the tutorial and actually bring the overall chip level in $V_{\text{min}}$ down to where we need it to be.

13. Outline

So enough of motivation and background I want to get started here first talking about the class of $V_{\text{min}}$ constraints that surrounds frequency and performance.

14. Frequency vs. Voltage

So I have plotted here, I grabbed basically from last year’s ISSCC from oracle guys, Oracle paper, a shmoo plot. This how we would normally would think about frequency versus voltage dependence. When we are looking for $V_{\text{min}}$ and after we have designed the chip and fabricated the chip, we are just going to look at it after the fact right. And on the shmoo plot we might have regions in this frequency versus voltage characteristic where the chip is functional seen here in green and nonfunctional seen here in red. The very interesting part is where this boundary lies and that is probably where we want to operate our chip from because that allows us to hit our frequency target at our lowest voltage possible. And so if we have say a given frequency target we can look at where this boundary is and we basically can calculate our $V_{\text{min}}$. So if all we cared about was looking at $V_{\text{min}}$ after the fact I could probably stop right here and we could move on. This is because it is just measuring it but what I want to try to give you in the next couple of charts is a little more intuition as to where this relationship comes from and what we are going to do when we care about a population of chips because we usually just don’t care about one we have to yield multiple.

So I want to try to talk through intuition on frequency versus voltage and talk through how we handle it in considering variability and in projecting what our $V_{\text{min}}$ might be as limited by frequency. So the first bit of intuition that I want to point out is that this particular boundary, which is really what we care about here, is kind of a straight line and I will admit that everything kind of looks like a straight line if we squint the right way and if you zoom in enough.

15. Frequency vs. Voltage: Some Intuition
But it actually works out pretty well in a lot of our work at IBM. We actually use a very simple linear approximation here and it gets us kind of where we need to be just in the first order of approximation. So we might assume sort of a slope alpha and we also usually have an intercept here- a V zero value- that seems to be technology dependent and it is around a quarter volt. But will depend on what sort of circuit topologies you use and how your transistors behave and what your Vt is and this is obviously an approximation. In reality you probably have more of a curve that say bends over a little bit at the high voltage because you have the law of diminishing returns. Then here at low voltage usually things fall off the cliff because you just have weird things going on with your timing. But if you think of it as a straight line it probably gives you enough intuition to at least, in the practical regime of operation, to think through how frequency is going to depend on voltage so that you can think about your Vmin. Obviously voltage is very related to power and so if we are looking for power minimization, which is probably what we are all trying to do at the end of the day, we kind of can keep in mind that powers may be a V to the third power-ish dependence. So getting that Vmin down is obviously going to be a good thing for improving the power.

16. Variability Impact on Vmin

So if that is the frequency versus voltage relationship and the intuition that we want to have for a given chip, variability is actually probably is the more difficult one to understand. We do obviously as circuit designers spend a lot of time thinking about how to project the impact of variability to capture it properly. So I want to spend a few charts talking about that. So obviously if you have a distribution of chips, you know say a gaussian distribution, which is what most things end up being then variability, say as coming from global process skew, so your L gate or Vt will vary from chip to chip from wafer to wafer, you have to make sure that all these chips together as a population yield. Also you might want to know the Vmin of that population with random local variations probably because you have a single path or a couple of paths that will dominate performance because randomly something is going to be slowest. But if you plot out a distribution here you have to make sure that the worst case chip does yield and what happens in general is that the Vmin will increase.

17. Vmin Dependency on Yield Target

And I will note that the particular yield target that you have will impact exactly how much that Vmin might increase and this is one of our first examples of application dependents. If you are looking for example for a high volume part where you need 99 percent yield then you have to make sure that you know this entire distribution works. So that Vmin is probably going to be higher in that the case where maybe you only need 50% yield. Maybe you are only looking at a low volume super high performance parts where you only need to make a couple of them to satisfy your customers and this variability maybe you don't care quite as much about and your Vmin can actually come down just to your yield target.

18. Variability Analysis Techniques
So now we have gone through a little bit of intuition on frequency versus voltage and some intuition on what it is we need to do to capture variability into that calculation.

I want to go through some variability analysis techniques. Traditionally, I think what we have all done is we used fast/typical/slow models in our simulations. So you build your circuit simulated it different corners and you say that that is the variability distribution. Now these days that works reasonably well if you are not too worried about variability. However, necessarily those types of models will over estimate the impact of variability because all process parameters will probably be skewed to their worse case points. It is not a statistical calculation it is just sort of a reference point that will kind of get you in the right ballpark. But if we are really trying to shave down the variability margin and in this case we are trying to reduce $V_{\text{min}}$ as much as we can we have to a better job.

So we need more accurate statistical multi-variable analysis techniques to properly capture the impact of variability the traditional way there is to do brute force Monte Carlo stimulation so just run a heck of a lot of simulations but what I want to talk about in the next two charts are some techniques that improve upon that and improve upon the efficiency of that analysis.

So I want to talk about important sampling and also a more analytic technique that we call the most probable worst case specter.

19. Importance Scaling

First with importance sampling, the idea here is to recognize that in traditional Monte Carlo analysis, you take you know samples at random. So when you look at the distribution of the simulations that you might get you actually end up with a lot of samples that are totally fine. They are in a good part of the distribution but since we are looking for failure points and we are trying to understand where the tail of this distribution is we actually don't get very many samples over here in that tail, just due to statistics. If you expect to have one fail in one part per million then obviously you have to do about a million simulations just to get one data point and to sort of make sure that you are not looking at things in error you need a lot more than that. So that takes a long time and with the importance sampling the idea is actually to use a distorted sampling function. So something that may actually be different from the distribution itself and you can set up if you work the math properly in a case where you have a lot more failing samples in your simulation. You should be able to get actually some pretty extraordinary speedups over traditional Monte Carlo. The trick here is to do the math correctly so you have to do a lot of mathematical manipulation to create the sampling function and then also to un-bias the results at the end. So I will simply in this talk refer you to some references that you can see how at least in one example how they did that.

20. Most Probable Worst-Case Vector

Another technique that we like to use if we are trying to do more, not quite back of the envelope, but pretty close to that type of analysis is a little more analytic and it is very fast and reasonably accurate for a lot of things. We call it the most probable worst-case vector method and it is just
an analytic technique to try to statistically combine a couple of Gaussian distributions to essentially convolute them together.

The way I try to explain it is with this little graph here on the left. So what I plotted; this is a two variable example. So you can imagine how you might extend this to many more variables but if you plot the 2 variables on the 2 axis I want to draw 2 sets of contours. Those are the ones I am going to look at so maybe in this example we can imagine that the two variables are say the N and P threshold voltages. The contours I am going to draw; one are the blue ones here which are contours of constant probability. On this set of axis you would imagine this should be ellipses. And so at every single point on this ellipse here that I labelled one sigma these are all one sigma yield points. And on the other contours I am going to draw are of the contours of the function to be minimized.

So if I am interested in frequency, for example, I might draw contours of constant frequency.

And so what I am looking for is the worst one sigma design, the worst 2 sigma design and I don't want the one sigma fast case in this. I am thinking about Vmin so I want the one sigma slow case. What you find graphically at least is that the point at which the 2 sets of contours are tangent to each other define that point that you care about.

So if you map out essentially a vector going from one to two to three sigma that is the most probable worst case vector and that defines all the points that we care about for this worst case frequency.

Alright, so intuitively hopefully you can see how this is going to be related to the partial derivatives of the function to be minimized with respect to the 2 variables. So you can work out a sum of squares type of math here to describe the distribution of the overall metric that you care about, say the frequency as a function of the individual sigmas and the individual variables that come into play. So this is something that you can actually do pretty quickly in simulation. Just assuming linear slopes and things like that and you can pretty quickly back calculate what frequency distributions are and use that and your frequency versus voltage relationship to figure out and what Vmin to project and what Vmin for your design will be.

21. Improving Frequency-Limited Vmin

So now that we understand hopefully, how to analyze variability and nominal performance in thinking about Vmin I want to talk about some circuit techniques to improve frequency limited Vmin.

The most obvious thing to do of course is to design faster circuits. If you have say a Finfet device that works better than the last generation you should be able to bring Vmin down simply because the device performance is good. But, I am going to postulate that because we are all very good circuit designers and we are given a fixed function to implement there is probably not too much we are going to be able to do to improve the nominal circuit performance.
So what I want to talk about here is ways in which we can improve the frequency limited Vmin by really targeting variability. And so the first thing we can try to do is reduce the variability itself in the circuits that we design. There was a nice paper here at the ISSCC a couple of years ago from Intel where they talked about how knowing that you want to bring the Vmin down, you can design using gates with smaller stacks, lower fan-ins, wider transistors that vary less and lower VT's that maintain larger VGS under variability. These are all techniques that should be able to allow you to use a lower voltage on your chip.

The caveat is that I will point out, each of these techniques actually degrade your power you end up with more gates, with more capacitance and more leakage. Therefore you have to think at the end of the day is this the best thing to do if my overall goal is actually to reduce power?

The other set of techniques that I will spend a couple of charts talking about is whatever variability you have in your circuit. You must try to reduce the margin that you have to apply. This family of techniques that I was thinking about is active compensation so using things like critical path monitors and also timing speculation. This sort of apply the lowest possible Vmin margin that you can to fight variability.

### 22. Critical Path Monitors

And so with critical path monitors the idea is kind of simple at the end of the days. It is basically taking your circuit and knowing where the critical paths are and making some replicas of those critical paths and using them as sensors. So if you figure out a way, an infrastructure around which you can take these replicas paths, measure the delay, sort of constantly of what you are getting on the chip. You can do some calculations afterwards and make sure that you are always using the lowest Vmin that you can get away with. This obviously captures not just variability but reliability and any sort of workload dependence or temperature changes or anything like that.

This particular type of concept is how things are implemented in our IBM power processors. Here on top of just these basic critical path sensors we put a bunch of them and we actually mux them together. So this will allow you to reconfigure things and look at more wire loaded situations or more gate loaded situations and different types of gates.

So you can imagine that you can make this as complex as you are willing to make it. To get as much information as you think you need to in order to sense the critical path delay properly and then feed it back into a voltage regulator so you can also apply this onto the frequency of the PLL that you have on the chip. You can basically adjust things properly and in our context with think about Vmin, so you should be able to bring the Vmin down as much as you can all the time.

### 23. Razor: Timing Speculation

A similar type of end goal is that to use timing speculation. And this is work that has been championed by the Michigan guys for a while now. They call it razor and the idea is to shave off as much of a margin as possible. Just like the critical path monitors but the difference is in the
implementation in that they actually come in and they change every single flip-flop in the system. So instead of just the basic flip-flop you add a shadow latch in parallel with that flip-flop and you run that off of a delayed clock. So for example as you start decreasing your voltage, the flip-flop will just pass the data to the next logic pipeline stage. But you are always kind of checking whether that data is correct with that shadow latch. The moment you fall off the cliff and you start getting a timing error then this shadow latch will have a different data than this flip-flop. It will generate an error and in their architecture they basically figure out a way to deal with that. But this is again kind of a sensor type of concept where if you start lowering the voltage you can start figuring out exactly where the Vmin is and where things start to fail.

24. Outline

So that is all I wanted to say about frequency limited Vmin and circuits techniques that we can apply to try to fight that and what I want to move onto now is functionality limited Vmin.

25. Noise Margins in CMOS Logic

And as we learned in class noise margins in CMOS logic; these days we use static CMOS for almost everything because of the inherent robustness. Also static CMOS because of the complimentary nature actually has very, very large noise margins. It is usually not that much of an issue if we just look at inverter transfer characteristics here. I hope you will all remember that the noise margins have very, very large voltage ranges in order to maintain stage to stage functionality. A CMOS logic gate is very good at cleaning up signals and bringing things back to rail. It has very, very large noise margins and the only time you will run into any sort of problem is if you get extreme mismatch between these 2 transistors, that might take this transfer characteristic function and skew it way to the right or way to the left, that is the only time when you are going to get an inverter that doesn't actually function. And this doesn't happen very often. I might actually put out there that this ideal 36 millivolt that we talked about at the beginning of the talk I think it is not entirely outrageous. So for digital logic we don’t usually worry so much about functionality limits to Vmin but we do actually have to worry very much about this in memory circuits because we do actually get extreme mismatch.

26. Random Variation

This extreme mismatch is not because you end up with N to P skew on the wafer or gate length variation, at least in a global scale, what we worry about today is random variation. Individual device to device variation; nothing you can control that is just what you get, 2 devices sitting right next to each other and looking very different. And this comes from 2 major sources these days. One is line edge roughness and that is just because we have such small transistors these days and small line widths with any wiggle or bump that comes from litho or the process, the litho on the etch and ends up with some variation of the gate length; probably the more fundamental issue that we have to deal with is that of random dopant fluctuation and that is because you know transistors are actually atomistic in nature and if you go home and do the calculation and you think through how many dopant atoms are actually in the channel of an advanced transistor these days, you will find that that number is not very high simply because these devices are so small. So statistically speaking with the dopant concentration you are going
to end up with 1 or 2 extra atoms here or there with some probability and you are going to realize that 1 or 2 atoms is actually a very big deal. So the net impact of a dopant fluctuation; and you can also capture line edge roughness in a similar way; is that you get a Vt variation in the transistor itself so every single device on the chip is going to have some random Vt applied to it. You can describe it based on a lot of work that people have been doing. The sigma of that distribution as related to the W and L of the device also the tox and the doping concentration.

27. Functional Vmin Limitation: Memory

I am going to talk in this section primarily about memory and I am going to focus my discussion on SRAM and as a memory sub-committee here, as an SRAM guy, I will say that this is entirely what we talk about. We think all about Vmin; how we bring the SRAM Vmins down and circuit techniques to target Vmin, everything else is secondary to this.

So this is a very very important topic and that is what I am going to focus this section on and the reason why we worry so much about this is SRAM is particularly sensitive to random variation. It has a lot of functionality problems with respect to Vmin as a result it is because as one SRAM you obviously want to get as many bits as you can on the chip to make the cell as small as possible. The transistors are usually the smallest ones on the chip. So they vary the most, one because they are small but also because they have high VT generally to try to keep the leakage down so that actually makes the sigma VT even worse.

SRAM operation fundamentally is dependent on very delicately balanced transistor strengths, it is not like your inverter transfer characteristics. It is much less robust the\an that and so that actually makes things very difficult and on top of that we have to remember that every cell has to work. We apply some error correction here but most of the cells do need to work. It is not like the logic path where you may have 20 logic gates before you get to the flip-flop. So there is some averaging going on between those logic stages. With this, every single cell we have to be able to read and write properly. If you put all of these together; where you have lots of random variations with a circuit that is very sensitive to random variation and some extreme statistics that you actually need to get to work; you can probably already see why SRAM is the biggest deal in functionality limited Vmin.

Here I will mention that there are other circuits of concern. You do have register files and latches on the chip which do also face some functionality limited Vmin issues. I am just going to not talk about them and say that they are like large SRAMS. I will also mention that embedded DRAM which is becoming more and more popular these days, I will say that is beyond the scope of the tutorial, it has similar types of issues to SRAM because it is memory but it has it's own can of worms. For analog I am just going to mention; because yes there will be functionality limited Vmin for a lot of analog circuits, but again I am taking a big picture view so we are not going to talk about them.

28. SRAM Scaling Challenge

So SRAM you are going to have a bunch of charts on this. The basic challenge that we have is in SRAM Vmin and thus chip level Vmin is rooted in the cell itself in the way that we actually
operate it. Traditionally, we used 6 transistor cells where we have cross couple inverters and then we use 2 pass gates to access the cell for both a read and a write and the problem is that it is the same devices that use both for the read and the write. You can certainly imagine that in order to write the cross couple inverters you actually want these cascades to be very, very strong and to read the cell you want them to be weak because you want to look at the contents of the cell but you don't want to ruin and destroy the data. So you have these conflicting requirements here and if you think about what variability is going to do well; I kind of didn't have that margin to begin with because I am fighting between the read and the write, then I apply the variability then everything goes to pots and you severely degrade your reading and write margins. As a result you always end up with very low Vmin.

29. DC Cell Read Stability Margin

So you really have to take this into consideration and you have to design your circuits to deal with this. So in the next couple of charts I want to give you a little bit more historical intuition. And I am going to put the disclaimer out there right now that this is actually not quite how we think about it today but historically; especially in our classes we learn about DC read and write stability margins.

Our first discover read margin is what people normally like to talk about and they normally like to plot these curves here and they call them butterfly curves. That is basically looking at the two halves of the cell in the read condition and looking at the transfer characteristics and plotting them back to back. So during a read we turn on the word line and we put both bit lines to 1 and the 0 node of the cell. It will start to conduct current but what happens there is that the 0 node of the cell will get disturbed. This pass-gate will pull up to VDD and sort of bring that 0 up a little bit and that is actually the root of our problem; that is the disturb that we always talk about and so that distorts the inverter transfer characteristic up a little bit. What we are looking for read stability is we actually want 3 intersection points. We want the distance between these 2 sets of curves to be as large as possible. I will say right off the bat here and I will discuss it in a later chart that this metric is actually pessimistic so we actually usually do better than this in the actual operation.

30. DC Cell Write Margin

We can do the same thing with the write margin; except that in this case a cell in the write condition, we have one of the bit lines pulled down to zero and that actually you are counting on this guy to really start to write the data into the cell. The distortion of the inverter transfer characteristic is very severe because you have a past gate pulling down to zero and nfet is very good at that and so you can plot the same types of butterfly curves and in this case, because you are writing the cell, you actually want the cell not to be stable. You want to overwhelm the cell with the new data and so you can look again at the distance between these two curves and define that as your write margin. Now I will say just to note very quickly that this metric is generally too optimistic when you think about it in actual operation and we will talk about that in the subsequent chart.
31. 6T-SRAM Cell Device Ratios

So historically the way we used to deal with SRAM Vmin is that we keep those DC read and write margins in our heads and then we try to ratio our cell devices so that we maximize or get enough margin for both the read and the write. We have to hit that 5 sigma design point to make sure that all the cells will work and then after that we try to make tradeoffs to get the performance, the leakage and the cell size that we really need to do. This hasn't changed, it is basically a big trade off game and you are trying to figure out what type of design point works; when you are juggling all these different constraints. Historically, it is up to about the 90 nanometer node or so where we did a very good job getting to where we needed to be just by changing all the ratios of the devices, changing W's and L's and VT's and things like that to try to maintain both our read stability and our write margin and so generally what you ended up with is this sort of Pfet here that was kind of a minimum sized. You need to size the pass gate to be strong enough to overwhelm the Pfet for your write margin and then you would size after that the pull down to be stronger then your pass-gate in order to make sure that your read stability margin was sufficient.

32. 6T-SRAM Write Operation

So I said that this worked very well thinking about things in DC by just looking at transistor strength and things like that. Today we don't really, we still do that but that is not enough. So what I want to do in the next 2 charts is show AC operation of the read and write operations and then use these waveforms to basically show you what we are actually doing in products today to bring Vmin down for SRAM. I am going to start with the write and basically as we discussed you set the bit lines to the different values and you pulse the word line up; so as long as you set up your word lines and your bit lines properly; then the cell will flip at some point. You can already see hopefully that it does take some time for this cell to flip and actually the pulse width of the word line, basically the amount of time that you spend trying to write this cell can actually limit your cell write margin. That is why I said that the DC margin was actually too optimistic.

33. 6T-SRAM Read Operation

The read operation you are doing is a similar thing except you pre-charge the 2 bit lines to the high voltage and then you count on the zero node of the cell to say pull down one of the bit-lines; so what happens when you look at the wave-forms of the cell itself is that you disturb the low-node of the cell and due to feedback you also kind of disturb the upper node of the cell; at some point the bit line starts to come down, this disturb goes away. But your hope is that these two nodes don't crisscross and end up doing a write. You want to maintain the cell read state. So you can also probably see that in addition to adjusting the Fet ratios; you must make sure you have enough read stability here, and the timing again plays a pretty big role because the amount of time under which the word line and the bit line are on is important. Also the bit line is up at high so it determines just how much of a disturb you get on the cell and just how much your read margin is.

34. 6T-SRAM Read. BL Length
So right here I want to point out that there are 2 schools of thought in designing SRAMS today and that actually changes the timing of that bit-line tremendously. This also tremendously impacts what your Vmin is and the difference is basically short versus long bit line. That is basically how many cells you might put and share on that particular bit line, a short bit line design might be say 16 cells on that bit line and a long design might be 256 and so the 256 guy is going to have a lot more capacitance on the bit line and that bit line will move very slowly and thus not move very far. And so what happens is that long bit-line is probably going to be low swinging where as the short bit line is not much capacitance so it is going to come all the way down to the ground. That is actually going to influence just how much of a disturb you impart to the cell. So if you can make this bit-line swing tremendously fast 16- bits per bit line is pretty quick, you can actually reduce the amount of disturb that the cell sees just because of the time constant. And so different applications, which I will foreshadow now, will choose one design or the other because there is some density in speed trade-offs here you can also see that a long bit line doesn't move very much. You probably need a full on sense amp whereas with a short bit-line you might just put an inverter at the output because you are swinging the bit-line rail to rail. And so there are some trade-offs in how you would choose it but there is already some difference in how hard you are going to have to work on Vmn just fundamentally based on how you design the bit-line length.

35. Read/Write Assist Techniques

So that is an example of pretty much what we are doing today. Everything is sort of an AC type of situation and the name of the game in SRAM, Vmin for the last several years has been read and write assist techniques. So the cell is still going to operate the way you normally think about it operating but what we are going to try to do is to play with all the peripheral circuits so essentially dynamically modulate the device strengths. So we are going to have different device strengths during the read, the write and the standby conditions and we are going to do that usually by generating voltages locally. So we will create a driven rail and then try to use these rails to say reduce the pass-gate disturb in a read assist by making the pass gate weaker. In a write assist we will either make the pass-gate stronger or we will make the Pfet weaker. The key issue here with all these techniques people publish paper after paper about ways to do this the key is trying to figure out the best way that minimizes your area of power and timing overheads so you can meet the specs that you need to meet.

36. Read Assist: Lowering WL/BL Voltage

So for read assist I won't show you circuit diagrams here I will just show you what people are trying to achieve with the different circuits. You can always go to the references to look at how they actually implement them but the idea is pretty simple in that you are just trying to generate either a word line voltage or a bit line voltage below VDD. So you are going to generate that with a driven rail of some sort whether through resistant dividers or some very simple regulators. You are going to try to amortize the penalty in the area in the power across as many columns and rows, subareas or whatever as you can, in order to generate this voltage and clearly this will reduce the strength of the pass-gate device and thus reduce the re-distribute and improve your read limited Vmin. It does degrade your read current because this pass-gate is now weaker and it
also actually does degrade your write margin. The pass-gate strength is weaker in the write so this is usually when people apply read assist they also need to apply in tandem

### 37. Write Assist: WL/BL Boosting

Some sort of write assist which as you can imagine is just the opposite; whereby you might boost the word line voltage above VDD, or you might actually boost the bit-line voltage below ground to essentially increase the strength of that pass-gate device, to make it stronger so that you can write more easily. There are trade-offs; because obviously you can imagine there are timing delays that you have to take care of here, and with the voltages that are above VDD you have to worry about Vmax reliability limits in addition to cell leakage.

If you think about all the neighboring cells you are going to, for example, take a bit line and bring it below ground and some of the other cells might actually see now a positive VGS and might leak a little bit more.

So there are a lot of issues you have to think about and you can go to these references to look at the details. But this technique, at least this read and write technique that we have used has been used very successfully and they are pretty much critical to delivering the last several generations of SRAM.

### 38. Write Assist: Cell Supply Collapse

Now more aggressive write assist technique; that is a little bit different from the last chart, is something that Intel has been talking about lately and they have been using this in 22 SRAM and I would imagine they would be using this going forward. Instead of playing with the strength of the pass-gate they are playing now with the strength of the Pfet and so they are actually going to take the cell supply and collapse it as much as say down to zero. So obviously that makes the cell incredibly weak while you are trying to write and you are going to have a very nice write margin on that particular cell. So that is why it is a write assist, the issue that you are going to come up with is that cell supply line is usually shared across a bunch of other cells and so there is going to be another cell on that line that you don't want to write. And so that cell is going to have to retain his state, you don't want to lose this data. If you bring that cell supply down to zero it is going to be below any sort of data retention voltage that you are normally concerned about and what they are actually counting on is that they are not going to pulse the supply down for too long, they are only talking about a few nanoseconds or so. So in that time you are going to count on dynamic retention so this SRAM cell is now a DRAM. You are going to hope that the leakage doesn't destroy the cell data and that you still manage to retain the state on this cell. So basically you are trying to improve the write margin on this cell so you must make sure that you don't mess up the neighbor.

### 39. Dual-Supply SRAM

So as we go I am going to talk about a couple even more exotic techniques here; the technique that IBM likes to use in contrast to Intel on the last chart is what we call dual supply SRAM. It is
now going away from local voltage generation to just paying the penalty of bringing in a separate supply. This separate supply is a little bit higher than VDD we call it VCS and it is up to say 200 millivolts higher and the idea is to use VCS on the word line and actually also on the cell and to use VDD which is a slightly lower voltage on the bit-line. So you can argue, well we are kind of cheating the Vmin because we are bringing a higher voltage, there is that part. But in addition to that this differential between VDD and VCS actually makes a big difference in all cell margins and the bit-line voltage is a little lower so that it reduces pass-gate disturbs and the word line voltage is higher so you get stronger write margin and faster cells and everything.

So for IBM types of applications this idea makes a lot of sense so obviously you have to figure out what your system is; the system power planes you have room for that, you can deliver this extra voltage, the higher voltage actually leads to extra cell leakage. For certain applications this technique makes a lot of sense and actually eliminates the complication of the circuits and it pushes all the complication up into the power delivery system.

**40. 8T-SRAM**

Another exotic technique is that of 8 Transistor SRAM. So instead of trying to deal with the voltages and the peripheral circuits around the cell this is a technique that says well let's just mess with the cell circuit itself and try to make the rest of the peripheral circuits a little bit easier, though you could also combine this technique with all the other techniques we talked about before. But the idea here is to just say, okay I will just make my cell bigger I will add a couple of transistors, recognizing that the fundamental trade-off that we had with the 6 transistor cell was that we used the same devices for read and write. Here we can de-couple the read and write mechanisms and thus you can optimize the read and write as much as you want to. So this cell is probably the one that is going to give you the lowest VMIN at the end of the day because there is no conflict between read and write and it has 2 extra transistors. So it is going to be necessarily a little larger in the 6 transistor cell but it can be very viable for very high performance application say L 1 or L 2 cash.

**41. Technology Dependence: FinFET**

Those are all the circuit techniques that I want to talk about and that is pretty much getting us to where we are today at sort of 22/14 nanometer node and how SRAMS are designed for low Vmin. I want to mention FinFET because that is all the rage today and it actually makes a pretty big difference on SRAM Vmin and there are some pluses and some minuses, at least that is the case with everything in the world. The minuses are probably that we actually have more limited read write margin tuning with a finFET and that is because the Fin has quantized width, you either have 1 fin or 2 fins, and in an SRAM you are not going to have a tremendous number of fins because you are trying to keep the cell size, transistor size, down but you can't have 1 and 1/2 fins. So you can't tune things quite as well as you want. In addition the Pfet, because of the vertical side wall of the structure, the Pfet is usually much stronger as it actually degrades our write margin and makes it more difficult for us to balance the 2 and so necessarily it probably is just due to these 2 factors. The SRAM Vmin will probably be degraded if we switch over to a FinFET but on the flip side the FinFET structure has the potential for reduced sigma VT. It requires some advancement from the technology guys, they need the right material to set the gate
work function, they need to get the fin to be small enough so you know very small line widths and things like that. But if we count on our technology guys to deliver those 2 then sigma VT can be dramatically reduced because you don't have to have any dophine in the channel. So in that case you can have no dopant fluctuation so sigma VT can be very, very small and clearly that is going to improve your Vmin tremendously. Now, we do believe that the first implementations and incarnations of Finfet don't actually have this undoped channel but we are hoping we get there with the more advanced FinFET technologies. So we are going to have some pluses and minus here with FinFET. As circuit designers it is probably not going to be the case that the Finfet is just going to solve all of our problems, so we will need to be aware of the benefits and the trade-offs that come along with this new technology.

42. Application Dependence: SRAM

So I want to end this section with an example looking at application dependents of SRAM Vmin type solutions. I grabbed 2 papers from last year’s conference one is a sort of SOC platform SRAM from TSMC which is operating less than a gigahertz, the other is sort of a 7 gigahertz type of SRAM from IBM for our server micro-processor L1 Cache and you can already see from the cell size there are some differences here even if you normalize for the technology node. This is a bigger cell and because you are willing to use more area to get performance but just to link into the techniques that we discussed; the TSMC guys used a long bit line because density was very important for them, and they used a lot of assist circuits because there is time in the cycle to do all of these things. It also makes sense to use these techniques to get your Vmin; whereas the IBM guys, they just used a short bit line, and they used dual supply here. With the short bit line you actually have to dot together multiple levels in the hierarchy to get your final addressing right so they really just did it a totally different way. Hopefully this gives you just an example of how the design choices that you might choose in these SRAM VMIN techniques will depend very heavily on your application target.

43. Outline

So from here I want to move onto reliability.

44. Reliability Margins

And I am going to try to use one chart to cover most of the reliability mechanisms that we might traditionally think about because I want to make the case that most reliability mechanisms actually improve at low voltage- that is because reliability generates an electric field, so a low-voltage is going to only make reliability better, and if we are thinking about Vmin constraints there is not really a constraint per say from reliability on Vmin.

We do have to remember we have to have enough margin for frequency and functionality and all those things because reliability is essentially like variability and sometimes we might operate in say an application with say power management. You might operate sometimes at Vmax and other times at Vmin. You do have to consider the degradation that happens at Vmax of course so you have to capture the degradation and the reliability margins properly but I might categorize
most of what we need to think about as temporal variation. It is just over time that the transistor characteristics change and you can model luckily most of the FET reliability mechanisms such as bias temperatures, instability or hot carriers. You can model it as a VT shift and so you can just apply that to the same techniques we talked about for variability and frequency and things like that and pretty much capture what you need to capture in a reliability.

So I could have spent this section sort of talking again about what we talked about in the last section but I want to move on from there and actually point out some of the more interesting reliability issues that do actually limit Vmin. I am going to mention that a lot of reliability issues that we might worry about are; I might consider to be catastrophic failures, and that would be electric migration, dielectric break down, and things like that. I am going to say that these problems are not for us VLSI designers to solve. I think our technology friends need to be able to take care of these; so that leaves one mechanism that is very important that does not follow into the temporal variation or catastrophic failure category and that actually doesn't impose a very strong Vmin limitation and that is a soft error.

45. Soft Errors

So we are going to spend this section talking about that. So a soft error occurs if you have a particle from the atmosphere whether it be an alpha particle or a neutron coming into your silicone sub-straight and generating a heck of a lot of charge. And so it is a problem if this amount of charge is actually collected, so depending on your device structure and your layout you might collect some of this charge and if it actually ends up overwhelming the stored charge on your circuit node then that is a problem because you just lost your data. The stored charge is certainly set by CV and we might call it Qcrit in the literature and as of course you start dialing the voltage down then V comes down and Qcrit comes down and thus you are more susceptible to soft errors.

Now some of the soft-error mechanisms can be pretty well controlled by using the proper materials here in your chip and your package but there is at least one mechanism; cosmic rays which is probably actually our biggest concern these days that you actually can't do too much about, you just have to make sure that your devices and your circuits are sufficiently robust. So hopefully this will convince you that we as circuit designers actually need to think about soft errors as we start bringing voltages down.

46. Soft Errors. Logic, Latches, & SRAM

what we need to think about, we need to break down circuit impacts of soft errors in 3 categories. We might think about logic latches and SRAM on the one side logic; it is actually only an issue if the air gets latched in a big long logic path. If you accidentally flip one of the nodes- if it never makes it to the latch then no harm, no foul, no big deal. And so we usually don't need to worry very much about logic soft errors. On the other side of the coin, SRAM is actually a big deal because the SRAM has very small devices, very small node capacitance and if you flip any of the bits there you just lost your value. So that is a big problem, however with SRAM we usually are able to apply a lot of error correction codes whether parity checks or full on error correction so these large arrays are generally protected so we can recover from any sort of single bit soft error.
fail. And so what we are left with is latches in the middle and this is usually the limitation and then we think about it if we are looking at V\text{min} and that is because in a latch if you do get an error you are flipping the store value so you actually have a problem and it is also too expensive in general to apply ECC to your latches. This is what we have to think about as circuit designers in terms of a reliability limited V\text{min} due to soft errors, we have to focus on our latches.

47. Some Technology Relief?

So I do want to mention that we might have some technology relief ahead as we transition to advanced transistor structures such as SOI and FinFETs. We actually might get a big boost and improvement in soft-error rates, that is because these devices; the SOI one is the easier one to think about here at least in 2D, and with this isolation and this buried oxide region when you get charge coming in any charge generated below the silicon substrate doesn't make it to your device and so you actually only collect a little bit of charge up here. So with finFET you can also imagine a similar type of deal and so we do expect with these new technologies that soft error rates will improve. It probably won’t improve to the point where we can ignore the circuit designers but I do want to put it out there that in some recent data from Intel; at least where they do actually show soft error rates for their 22 Tri-gate, it actually comes down pretty significantly.

48. SER-Tolerant Latch Topologies

So what we as circuit designers need to do; whatever device we get from the technology guys, probably the biggest thing that we have to work with is actually the latch circuit itself. There have been a lot of soft error tolerant latch topologies talked about in the literature and they mostly depend on some type of redundancy, so it is about adding extra devices so that if you hit a single node in the cell you can still recover from this.

The classic circuit is the dice-latch, the dual interlocked storage cell whereby you actually have 2 latches that are kind of interweaved and cross-coupled, so you should be able to have much more tolerance to soft errors. More recently the IBM guys we are talking about were stacking our SOI devices to reduce soft-error sensitivity and again it is basically a form of redundancy that if you hit one of the devices in this stack you don't flip both. So that is the trade-off, some type of area and also power trade-off, in order to make the latches more tolerant to soft-errors.

49. Further Latch SER Mitigation

Beyond that of course there are things you can do even just in the layout to reduce soft error sensitivity, using larger transistors increases node capacitances, smaller junction areas decreases the charge collection volume. Also, where you put the transistors in the cell layout itself is actually very important because you can generally protect from a single bit fail but if 2 nodes actually fail at the same time that is a problem and it is hard to recover from that. So where you put your transistors and your circuit nodes and your layout actually makes a big difference.

There are even techniques where I will just point you to the reference; where you might even put extra devices in to try to balance things out and to try to essentially play with the nodes in the
cell to reduce the chance of a multi-bit fail or even a single bit fail in this case. In addition to layout things there are also circuits and architectural techniques that you can apply to deal with latch soft-error rate. There are techniques like Biser where essentially you try to block error propagation so trying to make latches a little more like the logic issues with SCR so that you aren’t as sensitive to a soft error fail. Razor as we discussed in the first set of charts can actually be modified, you can modify that latch design to detect soft errors and architecturally sense them and then replay your pipe-line stage. And the fail safe method, the most extreme case that I just want to mention that people do talk about using in cases where you are very soft error sensitive, is triple modular redundancy and that is basically just saying I am going to put 3 latches where I only need 1 and I am going to use 2 of them to vote because single bit fail; 2 of them should still be right. So that is obviously very expensive and the most extreme technique here but I was just trying to give you guys the range of things that you can do as a circuit designer and also as an architect to fight latch soft error rates so that you can deal with the circuit of the latch and you can deal with some layout as well as using some architectural things as well.

50. Outline

So I want to move on now to power delivery which I mentioned is something

51. Power Delivery is Important!

that circuit designers normally don't care very much about- we just assume that the rails come in magically, this is the chart I like to use to try to emphasize that this is not the case. We all have to think about power delivery and what I plotted here is the break-down of power dissipation in a couple of representative systems and you can see that the red bars, which is the power sub-system, actually is pretty significant across the board and in some cases may actually be more power than the processor and cache which is what we normally think about as VLSI designers.

So this maybe marginalizes us a little bit but I want to convince you that we can think about power delivery and actually contribute a lot to keeping this amount of power loss down and so generally it is about 25% loss in just bringing the power from the wall plug all the way through to your chip. So that is all the conversion steps to drop down the voltage and the distribution losses due to resistance. It is also supply noise tolerances due to parasitic conductances. So that is a pretty big deal which is why we need to think about this, for example if I can improve the $V_{\text{min}}$ just at the chip level. But if I don't think about the rest of it then, this is actually a very big deal in that I haven't really solved my problem. The other thing that power delivery makes really interesting is that the $V_{\text{min}}$ definition gets very confusing. There are many ways in which we can

52. Power Delivery Impact on Vmin

think about what $V_{\text{min}}$ actually is because the way we deliver power in a system today is that we usually have some type of external voltage regulator module, a VRM, that say steps down the high voltage whether 12, 48 volts or whatever type of system you are talking about down to approximately the VDD that we want to use, and then it actually has to be distributed through resistive and inductive parasitics all the way down to our circuit. So we have to think about what
goes on in the chip itself with the C4s. We have to think about packages and we have to think about board parasitics; all of this matters and so depending on whether you are a circuit guy or say a chip guy or a system guy you might think about Vmin at these different points and so I am not going to try to say who is right because it just depends on your perspective but I am just going to emphasize that since our ultimate goal is probably to reduce system level power we need to think about Vmin; not just at the circuit level but we need to think about how to bring down the voltage at these other levels as well.

53. Typical Chip Package Model

So in a typical chip package model we have to think about all of these parasitics. We have our resistances, we have our inductances, and what we usually try to do is we put de-coupling capacitors to cancel out in particular the inductive parasitics. So what we end up with usually is we have on the board the package and the chip level different levels of capacitances that we might apply, the values of the inductances and the capacitances that we put there sort of end up attacking different regions of the frequency spectrum in terms of the noise that we might see at the board level we might be talking about a micro second scale of low frequency noise- the package level we might be talking about 100's of nanoseconds, or mid-frequency noise and the highest frequency noise is the stuff that is on chip.

So the issue that we are concerned about here is the LdI/dt noise that I am going to talk about in the next couple of charts that might occur when the load current in the circuit changes so if you are say turning on a component, a circuit block that you had in sleep mode before you will get an in-rush of current and then you will have a problem or if you start any work load and you start executing instructions when you didn't before you are going to get this type of supply noise

54. Typical Supply Response to Load Step

and in general what it looks like, at least in a cartoon form, is something like this where you get maybe 3 distinct types of droops occurring at different frequencies. So this corresponds very nicely to what we talked about in the last chart here

with your board, your package and your chip parasitics and so you will see

3 sorts of characteristic types of droops going on. In general the 2nd and 3rd droop will share more of these chip and packages level parasites. We can usually for a lot of applications put enough external de-caps so that these are minimized so I kind of drew this one cartoon wise. At least the first droop is the most important one, the one that we have to worry about, and as circuit designers that are normally thinking about the circuit layout and area and things like that, the issue here is on chip de-cap resource and how much area we are willing to dedicate to that.

55. On-Chip Decoupling Capacitors

And so the traditional way in which we put de-cap on our chip is we might use say a MOS-capacitor. Maybe we have a thick oxide device available in our technology menu and we put
those everywhere in our unused lay-out light space. We try to create as much capacitance as we
can on our supply rails but I want to point out that more recently IBM and Intel, for example,
have started to integrate dedicated structures for capacitance to have more de-coupling
 capacitance. IBM developed the trench process in their high performance SOI CMOS.
Originally for imbedded DRAM but I think that we have realized that it is also just as important,
if not more important, for on-chip de-coupling because the 3D nature of this trench gives you a
whole lot of surface area to get a lot of capacitance per unit area, and then Intel in their 22
nanometer technology have showed a back end MIM cap, probably with some sort of high K
electric up there, but it is again added process steps just to attack de-coupling and so hopefully
this convinces you that

56. Application Dependence: Packaging

people are realizing that it actually very important to power delivery and de-coupling capacitance
in particular that they are going to throw extra dollars at getting special structures to improve the
de-coupling. Hopefully this makes you think a little bit about application dependencies because
the importance of all these power delivery overheads varies a lot depending on the target for your
application. The severity of your IR and your LdI/dt issues; they are clearly related to the amount
of current that you are going to draw in your chip, whether it is a micro-controller that is drawing
milli-amps or a high performance server or a multi-chip module that is drawing a kilo amp. The
amount of I makes a big difference in how much IR and LdI/dt you have .

In addition to package costs which maybe compensates that a little bit you must ask how many
dollars are you willing to throw at the package? This will determine what the R's and the L's are
so the micro-controller might have a much higher R and L than the very high performance and
expensive server part.

In addition, the availability of the de-coupling capacitance is also going to make a big deal. This
is also clearly a cost consideration; both in terms of the external capacitors and also the on-chip
technology that you might have. So you kind of have to put all of these together and for different
applications you can have a different range of parameters to work with so it is going to affect not
just how much you have to deal with these problems but also the solutions that you are able to
apply.

But, I would like to think that all of the issues across different applications are basically the same
it is just differences in magnitudes and thus what you are going to do from a circuit perspective.

57. Circuits to Mitigate Supply Noise

So I want to talk a little about circuits that people are using to try to attack these power delivery
over heads. I put supply noise here but it actually does also potentially improve your resistance
drops as well and so I am going to first talk about active supply regulation using linear regulator
techniques and then I will talk about more exotic techniques that people are working on at least
in the research phase right now specifically integrated step down conversion on the chip and so I
already said that Vmin the definition is kind of sketchy, it is hard to figure out well are you
looking at the chip level or the C4 level or the VRM output. Well if we are going to apply
regulators and convertors now and actually further blurs the definition of Vmin because you have an input and an output to the regulator and so I am not going to touch what we should actually define as V min but I am going to point out that it is something that you are going to have to keep track of.

58. Linear Regulators

So linear regulators what we are generally trying to do here is minimize any sort of supply droop at the load. So with Vmin of the load we are trying to really bring that down as much as possible. We are counting on essentially a tunable resistor that we are going to change very quickly to compensate for any load, any supply droop, so you are just modulating and strengthening this resistor. You do incur some loss across this resistor and there is definitely a voltage drop, across this resistor. But the idea here is that that loss is sort of linearly proportional to the volt,… It is linearly proportional to the voltage that you have here whereas if you had to apply let’s say this extra margin of voltage directly to the load itself it is more of a V to the 2.5 or 3 power. At the end of the day with linear versus super linear you usually win by reducing the power here by putting a linear regulator in series with your load.

Now the challenge here and I will simply refer you to a publication wherein they implemented this; and there is the follow on actually at the conference this year where they applied it to a power-A processor, basically what you are able to do is minimize the power lost in this resistor. Pfet usually works pretty well to minimize the voltage lost, the voltage drop there, and thus the power loss, and you also have to make sure that you have a very, very fast control loop on this linear regulator in order to compensate the very, very high frequency noise that we are worried about.

59. Push-Pull Series-Shunt Regulator

You can improve upon the basic linear regulator by putting 2 of them into your circuit and you can use say 1 to push current into the load and a 2nd one to pull current from the load. This is an improvement just upon the single linear regulator because what you are limited by is the average load variation rather then the peak so if you have a particular current wave form here, a signature, you will sometimes turn on the push and you will sometimes turn on the pull so you are not always drawing current from these extra supplies that you need to bring in. So this is an improvement but obviously it adds some complication. You are dealing with 2 different control loops here that could potentially be fighting each other and so it is just extra complexity but you do potentially get a benefit of improved regulation losses.

60. Step-Down Voltage Conversion

The more exotic technique that people are working on, at least in research in academia in particular right now, is step down voltage conversion and this is basically what we do when we bring power to our homes. We deliver power at a very high voltage because for a given amount of power P equals IV so the current that you have to deliver comes way down. So if we can do this in chip level power delivery we might, for example, deliver all of our power all around the
system at say a high voltage at say a factor N- larger then VDD. It will go through all the resistance inductive parasitics at this higher voltage so at the lower current, which reduces the IR, and the LdI/dt by sort of an N factor and then what we need is this magic N to 1 step-down converter on the chip or very close to the chip. And thus the net impact of IR and LdI/dt is actually improved by the square of the conversion factor so you have now basically taken all your IR's and all the LdI/dt’s and you brought them down tremendously and hopefully they are not an issue anymore.

The key challenge here is of course this magic step down convertor. If you just use traditional passive components, capacitors and inductors to do switching regulators or perform this function the efficiency is quite low, maybe in the 70 percent range. We really probably need to be up around 85% or 90% to make this type of technique viable because you don't want to give up all your power gains by losing it again in the conversion.

61. Integrated Step-Down Conversion

So what people have been working on to try to attack this is integrated step-down conversion using advanced passive devices and so recognizing that traditional capacitors are not high enough Q there is too much parasitic, not dense enough, you actually need a lot of capacitance to make this work. Also recognizing that you actually need traditional inductors say even high QRF inductors, the Q is not high enough, it is also the wrong frequency range and so people have to develop technology to make this work. There has been work on switch capacitor convertors using that deep trench structure which actually provides very low parasitics and high capacitance density so you can get nice high efficiency here in your step-down conversion and buck converters I will call out some of the work from Intel, as they have been doing a great job with Finfilm magnetics, and also package integrated inductors to try to get high enough Q for that inductor and in order to get high conversion efficiency.

62. Outline

So that is it for power delivery. I want to talk a little bit about energy efficiency and the limits that it may impose on Vmin going forward and this is something that may be more influencing how I might want to target say frequency in particular. Energy efficiency is something we will think about and we will debate

63. Vmin for Energy Efficiency

whether it is a constraint specifically or not but it is something that we definitely have to think a lot about as we are designing for Vmin; where we want that Vmin to sit and so I want to start the discussion with a lot of the academic work that has gone on not that recent but in probably the last 10-15 years or so. My team in Michigan has shown these types of plots a lot where they identify that there actually is a Vmin for energy efficiency and there is actually one particular voltage. If you plot the energy per operation of any sort of circuit as a function of the voltage, you start bringing the voltage down and you see that well yes the lower voltage gives me lower CV^2 and so the active energy comes down but as you reduce it beyond a certain point you see
that the energy per operation actually increases again and that is because when you bring the
voltage down so far the frequency and the performances come down so much that the clock
period is really, really long and so when you think about the energy consumed by leakage that
actually starts to dominate. So this Vmin is basically related to where you have your cross over
between your active and your leakage power and so this value for most circuits is around 0.3
volts or so. It is basically probably in the sub-threshold regime and so academically, and I think
we were very interested in this point for a long time, especially when we are thinking about say
sensor nodes and things like that that are just ultra-sensitive to energy per operation it makes a
lot of sense. But for those of us in industry we kind of look at this and we say, well the
frequencies that we are talking about here, you are talking about kilo Hertz frequencies so we
can't sell products based on that. So for main stream applications this particular Vmin constraint,
that of energy per operation is actually not very relevant because you end up with a performance
that is just way too low

64. Sub-vs. Near-vs. Super-vt Design

and so where a lot of us have been thinking in recent years we have been thinking about near
threshold operation. So if I take this curve that you saw from the last chart and just break up the
voltages into 3 regimes where sub-threshold might be this point where you do get your minimum
energy per operation but it is very, very slow. And I might call super threshold operation these
traditional 1 volt types of technologies where you get a lot of speed but the energy efficiency
isn't great. I might simply define the middle as near threshold, so near threshold is probably
above VT because we need some amount of performance. But this middle regime is where we
think there might be a much better balance between energy savings and performance loss
because you get a large energy reduction, as related to super threshold operation, but you don't
incur the huge delay increase that you might otherwise get in sub-threshold and so there has been
a lot of debate as to what the voltage is that you should be targeting actually is.

Actually interestingly enough across a lot of technologies and things that you might try to do I
think half a volt is a nice round number to think about and half a volt is already very different
from the sort of 1 volts types of applications that we normally think about. So it is actually a
pretty stark departure from our traditional voltage regime, it means that we have to push much
harder on all the Vmin techniques that we talked about in this tutorial to make sure you know
variability is appropriately compensated for, and to make sure your SRAM still operates at sort
of half a volt or if you consider power delivery losses and things like that maybe it goes down to
0.4 volts. It really pushes hard on all of the techniques that we have talked about today in this
tutorial.

But on the flip side it also may leverage things like finFET a lot because finFETs these types of
devices usually have much better power performance especially at low voltage. So you might
argue that there is actually an optimistic outlook here as well. There is definitely If we go to near
threshold design there is definitely a lot of work for us as circuit designers to do to make sure
that the Vmins are going to be within the right range.

65. System Performance Metrics
Now the question is, you know it is not a given, that near threshold operation is going to be something that we are going to have to worry about as circuit designers because it depends a lot on what systems we will need going forward, and the way I like to think about it is the balance between single thread performance and through put performance. This is debatable out there right now and we are going to have to see how this plays out, but the traditional metric and performance, if you ever have a system and you talk about the performance you actually have to talk about 2 performance metrics, the traditional one is single thread performance, which is essentially raw speed, the time to complete a given job. You just have to reduce the latency to improve the frequency and things like that. It works pretty much and it is optimized in that super threshold in kind of one volt-ish, a little bit lower type of regime versus through put performance which is the one that is becoming more and more important as we go forward. That is the number of jobs completed in a given amount of time so it is sensitive to latency but it is also parallelizable so you can add a bunch of parallel compute engines and you get more through put.

So as our software friends are learning how to parallelize all their work loads and they are working very hard on this we can't improve our frequencies anymore so this is maybe one of the remaining ways to really improve the performance. What we find is that this near threshold regime is actually very good for throughput performance so if you say that that is where we are going that is probably where we want to start thinking about near threshold computing. The issue is really going to be that at the end of the day in a system you probably need a balance because if you remember Amdahl's law

you are probably not going to be able to parallelize everything that a computer has to do so you are going to have some amount of single thread work that you need to do. So you are going to need some amount of both single thread and throughput performance probably from here till the end of time and so you are going to have to figure out how you are going to build a system whether you can optimize from one or the other or if you have to sit in the middle which thus means it is not optimized for either one. Or maybe you build a hybrid system where you have some super thresholds single thread centric circuits and you have some near threshold centric circuits. We will have to see how this plays out but I want to bring this up in the context of the Vmin discussion because whether we want to target super thresholds or near threshold that is going to influence very much what it is we have to do to get Vmin to be sufficient.

**66. Conclusions**

So that is all I had. Just to conclude I hope I have convinced you that understanding Vmin limits is really the key to maximizing power efficiency in VLSI design today across the whole range of applications. Vmin, as we have defined it here is the lowest chip voltage that we can possibly use to still achieve our target specifications so we really need to think about things like performance functionality and apply all of our variability and reliability margins. I hope that I have also pointed out the power delivery and the overheads that we incur there are also very important to think about.

There is a strong, strong application dependence whether you are talking about a low power or a high performance part because it influences the severity of the limits. It influences the acceptability of all the solutions because anything we do, it is engineering right, we had a trade-off and as the technology progress such as the finFET, which is all the rage right, now is likely to
help overall but that probably doesn't mean it is not a Panacea. We are still going to have to work on these types of issues but we have to be aware of how things are changing with the technology and there is a heck of a lot of work for all of us to do.

67. Related Papers

The rest of the charts in here I have listed a couple of papers at the conference for the next couple of days that are related to the topics that we have talked about so to hopefully help guide you in choosing what papers to look at and the rest

68. References

are references that you can go to, to read more about any of the topics that we discussed so that is all I have.

69. Conclusions

Thank-you very much.