IC and ESD

Carsten Wulff, carsten@wulff.no

Keywords: TempSense, Node Voltage, Ground, VDD, Clocks, Digital, Bias, RESET (POR), Package, Why ESD, CDM (Gauss, INV), HBM (01,10,02,20,12,21**), GGNMOS, Latch-up

Video is from 2024, so the plan might not be exactly the same. In addition, we're not using Caravel for the tapeout, but rather TinyTapeout.

What blocks must our IC include?

The project is to design an integrated temperature sensor.

First, we need to have an idea of what comes in and out of the temperature sensor. Before we have made the temperature sensor, we need to think what the signal interface could be, and we need to learn.

Maybe we read Kofi Makinwa's overview of temperature sensors and find one of the latest papers,

A BJT-based CMOS Temperature Sensor with Duty-cycle-modulated Output and ±0.54 °C (3-sigma) Inaccuracy from -40 °C to 125 °C [1].

At this point, you may struggle to understand the details of the paper, but at least it should be possible to see what comes in and out of the module. What I could find is in the table below, maybe you can find more?

Pin	Function	in/out	Value	Unit
VDD_3V3	analog supply	in	3.0	V
VDD_1V2	digital supply	in	1.2	V
VSS	ground	in	0	V
CLK_1V2	clock	in	20	MHz
RST_1V2	digital	out	0 or 1.2	V
I_C	bias	in	?	uA?
PHI1_1V2	digital	out	0 or 1.2	V
PHI2_1V2	digital	out	0 or 1.2	V
DCM_1V2	digital	out	0 or 1.2	V

This list contains supplies, clocks, digital outputs, bias currents and a ground. Let me explain what they are.

Supply

The temperature sensor has two supplies, one analog (3.3 V) and one digital (1.2 V), which must come from somewhere.

We're using TinyTapeout

That has ability for both 3.3 V and 1.8 V. An external low dropout regulator (LDO) provides the digital supply (1.8 V).

See for example absolute maximum ratings

Ground: Most ICs have a ground, a pin which is considered 0 V. It may have multiple grounds. Remember that a voltage is only defined between two points, so it's actually not true to talk about a voltage in a node (or on a wire). A voltage is always a differential to something. We've (as in global electronics engineers) have just agreed that it's useful to have a "node" or "wire" we consider 0 V.

Clocks: Most digital need a clock, and TinyTapeout can provide a 50 MHz clock which should suffice for most things. We could probably just use that clock for our temperature sensor.

Digital: We need to read the digital outputs. We could either feed those off chip, or use a on chip micro-controller. The TinyTapeout includes options to do both. We could connect digital outputs to the logic analyzer, and program the MCU to store the readings. Or we could connect the digital output to the I/O and use an instrument in the lab.

Bias: The TinyTapeout does not provide bias currents (that I found), so that is something you will need to make.

Conclusion: Even a temperature sensor needs something else on the IC. We need digital input/output, clock generation (PLL, oscillators), bias current generators, and voltage regulators (which require a constant reference voltage).

I would claim that any System-On-Chip will always need these blocks!

I want you to pause, take a look at the

course plan

and now you might understand why I've selected the topics.

One more thing: There is one more function we need when we have digital logic and a power supply. We need a "RESET" system.

Digital logic has a fundamental assumption that we can separate between a "1" and a "0", which is usually translated to for example 1.8 V (logic 1) and 0 V (logic 0). But if the power supply is at 0 V, before we connect the battery, then that fundamental assumption breaks.

When we connect the battery, how do we know the fundamental assumption is OK? It's certainly not OK at 30 mV supply. How about 500 mV? or 1.0 V? How would we know?

Most ICs will have a special analog block that can keep the digital logic, bias generators, clock generators, input/output and voltage regulators in a **safe** state until the power supply is high enough (for example 1.62 V).

One of the challenges with a Power On Reset (POR) is that we want to keep the system in a reset state until we're sure that the power is on. Another challenge is that the POR should not consume current.

If we make a level triggered (triggers when VDD reaches a certain level), then we need a reference, a comparator and maybe other circuits. As a result, potentially high current.

If we make a delay based POR, then we need a long delay, which means large resistors or capacitors. Accordingly, high cost.

Below is an idea for a Power-On-Reset I had way back when. The POR uses a delay based on the tunneling current in a thin oxide transistor (2), and uses a thick-oxide transistor (3) as a capacitor. The output X would go to a Schmitt trigger (5).

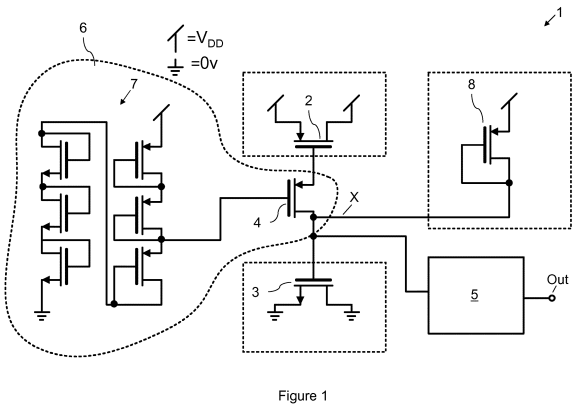

Figure 1: Power On Reset using gate tunneling

Electrostatic Discharge

If you make an IC, you must consider Electrostatic Discharge (ESD) Protection circuits

ESD events are tricky. They are short (ns), high current (Amps) and poorly modeled in the SPICE model.

Most SPICE models will not model correctly what happens to an transistor during an ESD event. The SPICE models are not made to model what happens during an ESD event, they are made to model how the transistors behave at low fields and lower current.

But ESD design is a must, you have to think about ESD, otherwise your IC will never work.

Consider a certain ESD specification, for example 1 kV human body model, a requirement for an integrated circuit.

By requirement I mean if the 1 kV is not met, then the project will be delayed until it is fixed. If it's not fixed, then the project will be infinitely delayed, or in other words, canceled.

Now imagine it's your responsibility to ensure it meets the 1 kV specification, what would you do? I would recommend you read one of the few ESD books in existence - see the reading list at the end of the chapter - and rely on your understanding of PN-junctions.

The industry has agreed on some common test criteria for electrostatic discharge. Test that model what happens when a person touches your IC, during soldering, and PCB mounting. If your IC passes the test then it's probably going to survive in volume production

Standards for testing at JEDEC

The JEDEC standard splits ESD events into Human body model, Charged device model, and System level ESD.

Once mounted on the PCB, the ICs can be more protected against ESD events, however, it depends on the PCB, and how that reacts to a current.

Take a look at your USB-A connector, you will notice that the outer pins, the power and ground, are made such that they connect first, The $D+$ and $D-$ pins are a bit shorter, so they connect some μs later. The reason is ESD. The power and ground usually have a low impedance connection in decoupling capacitors and power circuits, so those can handle a large ESD zap. The signals can go directly to an IC, and thus be more sensitive.

We won't go into details on System level ESD, as that is more a PCB type of concern. The physics are the same, but the details are different.

Human body model (HBM)

HBM is the "simple" version of ESD, a model can be seen in Figure 2. Some of the properties of HBM are:

- Models a person touching a device with a finger
- **Long** duration (around 100 ns)
- Acts like a current source into a pin
- Can usually be handled in the I/O ring
- 4 kV HBM ESD is 2.67 A peak current

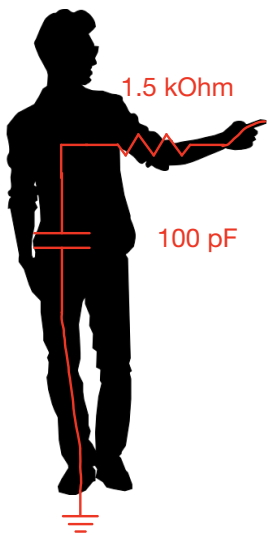

Figure 2: Human body model (HBM)

More on circuits that protect from HBM later.

Charged device model (CDM)

> An IC left alone for long enough will equalize the Fermi potential across the whole IC.

Not entirely a true statement, but roughly true. One exception is non-volatile memory, like flash, which uses Fowler-Nordheim tunneling to charge and discharge a capacitor that keeps its charge for a very, very long time.

I'm pretty sure that if you leave an SSD hardrive to the heat death of the universe in maybe $10^{10^{10^{56}}}$ years, then the charges will equalize, and the Fermi level will be the same across the whole IC, so it's just a matter of time.

Assume there is an equal number of electrons and protons on the IC. According to Gauss' law

$$\oint_{\partial\Omega} \mathbf{E} \cdot d\mathbf{S} = \frac{1}{\epsilon_0} \iiint_V \rho \cdot dV$$

Which says that the electric field through the surface is the volume integral of the charges inside the surface. If there are the same amount of protons and electrons, and the distribution is even, then there will be no field through IC surface. As such, there is no external electric field from the IC.

If we place an IC in an electric field, the charges inside will redistribute. Flip the IC on its back, place it on an metal plate with an insulator in-between, and charge the metal plate to 1 kV, as shown in Figure 3.

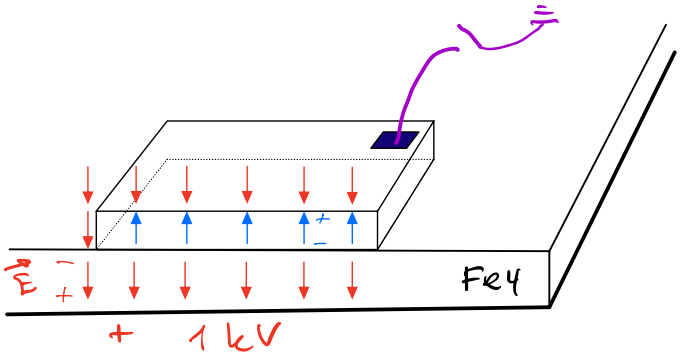

Figure 3: Charged Device Model (CDM) testing

Inside the integrated circuit, electrons and holes will redistribute to compensate for the electric field. Closest to the metal plate there will be a negative charge, and furthest away there will be a positive charge.

This comes from the fact that if you leave a metal inside an electric field for long enough the metal will not have any internal field. If there was an internal field, the charges would move. Over time the charges will be located at the ends of the metal.

Take a grounded wire, touch one of the pins on the IC. Since we now have a metal connection between a pin and a low potential the charges inside the IC will redistribute extremely quickly, on the order of a few ns.

During this Charged Device Model event the internal fields in the IC will be chaotic, but at any given point in time, the voltage across sensitive devices must remain below where the device physically breaks.

Take the MOSFET transistor. Between the gate and the source there is an thin oxide, maybe a few nm. If the field strength between gate and source is high enough, then the force felt by the electrons in co-valent bonds will be $\vec{F} = q\vec{E}$. At some point the co-valent bonds might break, and the oxide could be permanently damaged. Think of a lightning bolt through the oxide, it's a similar process.

Our job, as electronics engineers, is to ensure we put in additional circuits to prevent the fields during a CDM event from causing damage.

For example, let's say I have two inverters powered by different supply, VDD1 and VDD2. If I in my ESD test ground VDD1, and not VDD2, I will quickly bring VDD1 to zero, while VDD2 might react slower, and stay closer to 1 kV. The gate source of the PMOS in the second inverter will see approximately 1 kV across the oxide, and will break. How could I prevent that?

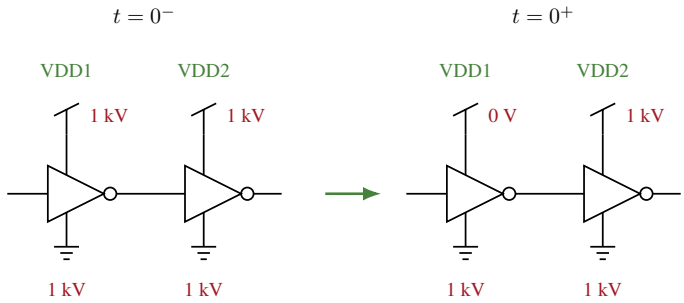

Figure 4: Cross domain voltage problem with CDM (or indeed HBM) events

Assuming some luck, then VDD1 and VDD2 are separate, but the same voltage, or at least close enough, I can take two diodes, connected in opposite directions, between VDD1 and VDD2. As such, when VDD1 is grounded, VDD2 will follow but maybe be 0.6 V higher. As a result, the PMOS gate never sees more than approximately 0.6 V across the gate oxide, and everyone is happy.

Now imagine an IC with hundreds of supplies, and billions of inverters. How can I make sure that everything is OK?

CDM is tricky, because there are so many details, and it's easy to miss one that makes your circuit break.

An HBM ESD zap example

Imagine a ESD zap between VSS and VDD. How can we protect the device?

The positive current enters the VSS, and leaves via the VDD, so our supplies are flipped up-side down. It's a fair assumption that none of the circuits inside will work as intended.

But the IC must not die, so we have to lead the current to ground somehow.

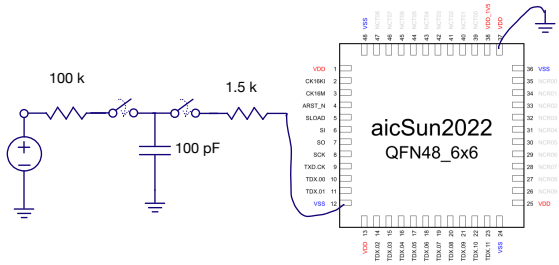

Figure 5: ESD HBM zap example

Let's simplify and think of the possible permutations, shown in Figure 6. We don't know where the current will enter nor where it will leave our circuit, so we must make sure that all combinations are covered.

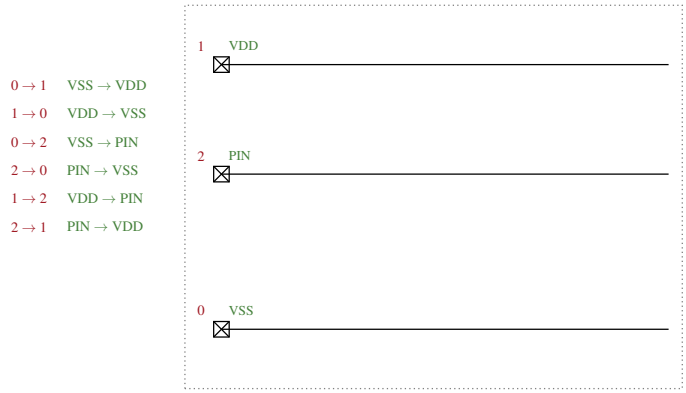

Figure 6: The six ways a zap can cross a three terminal chip. Three pads means six ordered pairs, and ESD protection is the promise that every one of them has somewhere for the current to go. The chip is empty here on purpose; the next three figures fill it in.

When the current enters VSS and must leave via VDD, then it's simple, we can use a diode, as shown in Figure 7.

Under normal operation the diode will be reverse biased, and although it will add some leakage, it will not affect the normal operation of our IC.

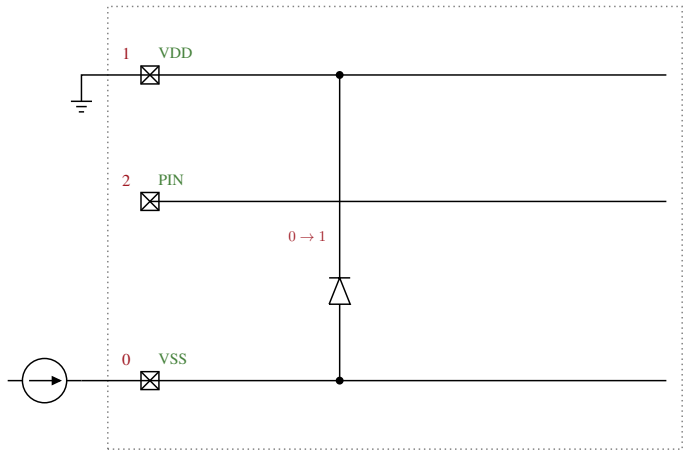

Figure 7: Protection for a zap from ground to VDD. One diode, reverse biased whenever the chip is running, forward biased the moment VSS goes above VDD. The column crosses the pin rail without a junction dot, which is a statement that nothing is connected there yet.

The same is true for current in on VSS and out on PIN. Here we can also use a diode, as shown in Figure 8.

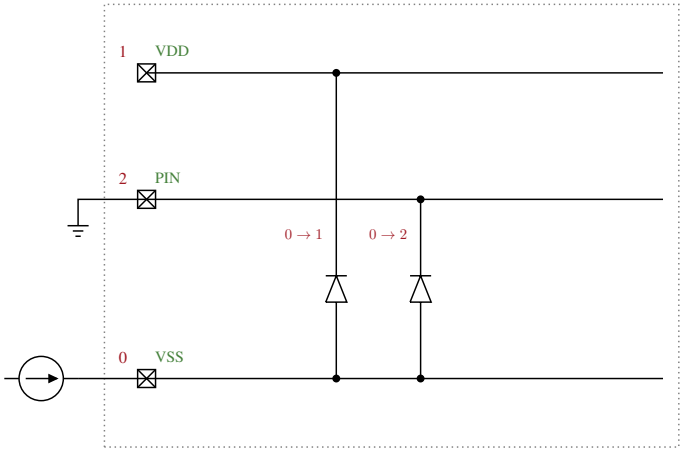

Figure 8: Protection for a zap from ground to a pin. Nothing new happens, and that is the point: a signal pin sits above VSS in normal operation for the same reason VDD does, so the same reverse biased diode covers it.

For a current in on VDD and out on VSS we have a challenge. That's the normal way for current to flow.

For those from Norway that have played a kids game Bjørnen sover, that's a apt mental image. We want a circuit that most of the time sleeps, and does not affect our normal IC operation. But if a huge current comes in on VDD, and the VDD voltage shoots up fast, the circuit must wake up and bring the voltage down.

If the circuit triggers under normal operating condition, when you're watching a video on your phone, your battery will drain very fast, and your phone might even catch fire.

As such, ESD design engineers have a "ESD design window". Never let the ESD circuit trigger when VDD < normal, but always trigger the ESD circuit before VDD > breakdown of circuit.

A circuit that can sometimes be used, if the ESD design window is not too small, is the Grounded-Gate-NMOS in Figure 9.

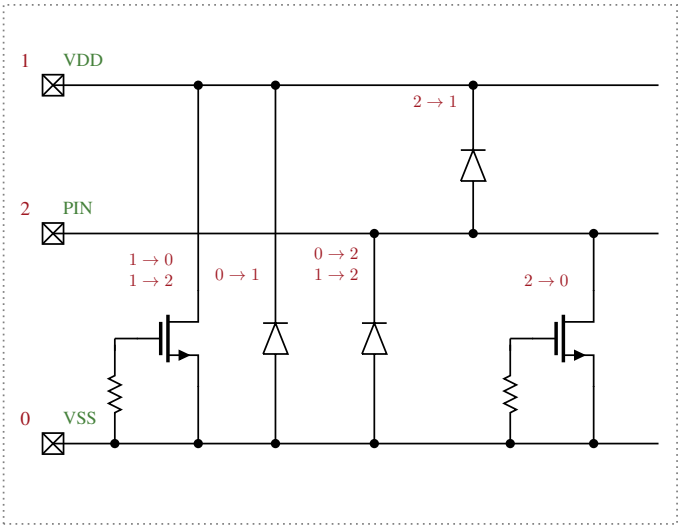

Figure 9: All six permutations covered by four devices. Each device is labelled with the zap it carries, and two of them carry one jointly: a strike from VDD to the pin goes down the left grounded gate NMOS to VSS and back up through the pin diode, which is why $1 \to 2$ appears twice. That is the real structure of an ESD network. It is not one path per permutation but a small set of devices arranged so every permutation has a series combination available, which is why adding a pin costs two diodes rather than six.

The grounded gate NMOS

If you try the circuit in Figure 10 with the normal BSIM spice model, it will not work. The transistor model does not include that part of the physics.

We need to think about how electrons, holes PN-junctions and bipolars work. Let's refresh quantum mechanics a bit.

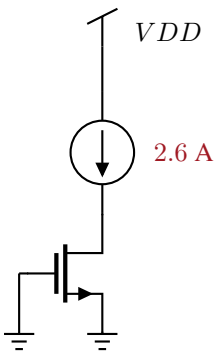

Figure 10: The grounded gate NMOS (GGNMOS)

Electrons sticking to atoms (bound electrons), can only exist at discrete energy levels. As we bring atoms closer to each-other the discrete energy levels will split, as computed from Schrodinger, into bands of allowed energy states. These bands of energy can have lower energy than the discrete energy levels of the atom. That's why some atoms stick together and form molecules through co-valent bonds, ionic bonds, or whatever the chemists like to call it. It's all the same thing, it's lower energy states that make the electrons happy, some are strong, some are weak.

For silicon the energy band structure is tricky to compute, so we simplify to band diagrams that only show the lowest energy conduction band and highest energy valence band.

Electrons can move freely in the conduction band (until they hit something, or scatter), and electrons moving in the valence band act like positive particles, nicknamed holes.

How many free charges there are in a band is given by Fermi-Dirac distribution and the density of states (allowed energy levels).

If an electron, or a hole have sufficient energy (accelerated by a field), they can free an electron/hole pair when they scatter off an atom. If you break too many bonds between atoms, your material will be damaged.

Assume a transistor like the one in Figure 11. The gate, source and bulk is connected to ground. The drain is connected to a high voltage.

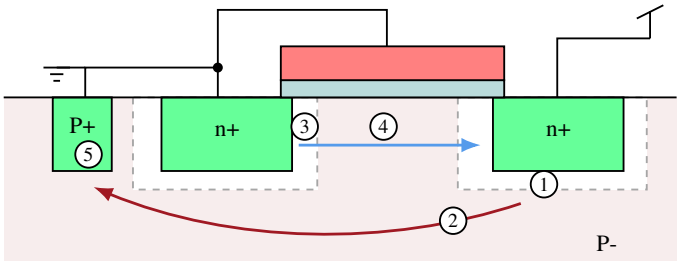

Figure 11: Cross section of the grounded gate NMOS

The process of a GGNMOS will be (1) Avalanche, (2) hole accumulation, (3) forward bias of PN-junction, and (4) direct electron current from source to drain.

The first thing that can happen is that the field in the depletion zone between drain and bulk (1) is large, due to the high voltage on drain, and the thin depletion region.

In the substrate (P-) there are mostly holes, but there are also electrons. If an electron diffuses close to the drain region it will be swept across to drain by the high field.

The high field might accelerate the electron to such an energy that it can, when it scatters off the atoms in the depletion zone, knock out an electron/hole pair.

The hole will go to the substrate (2), while the new electron will continue towards drain. The new electron can also knock out a new electron/hole pair (energy level is set by impact ionization of the atom), so can the old one assuming it accelerates enough.

One electron turn into two, two to four, four to eight and so on. The number of electrons can quickly become large, and we have an avalanche condition. Same as a snow avalanche, where everything was quiet and nice, now suddenly, there is a big trouble.

Usually the avalanche process does not damage anything, at least initially, but it does increase the hole concentration in the bulk. The number of holes in the bulk will be the same as the number of electrons freed in the depletion region.

The extra holes underneath the transistor will increase the local potential. If the substrate contact (5) is far away, then the local potential close to the source/bulk PN-junction (3) might increase enough to significantly increase the number of electrons injected from source.

Some of the electrons will find a hole, and settle down, while others will diffuse around. If some of the electrons gets close to the drain region, and the field in the depletion zone, they will be accelerated by the drain/bulk field, and can further increase the avalanche condition.

For a normal transistor, not designed to survive, the electron flow (4) can cause local damage to the drain. Normally there is nothing that prevents the current from increasing, and the transistor will eventually die.

If we add a resistor to the drain region (unsilicided drain), however, we will slow down the electron flow, and we can get a stable condition, and design a transistor that survives.

Turns out, that every single NMOS has a sleeping bear. A parasitic bipolar. That's exactly what this GGNMOS is, a bipolar transistor, although a pretty bad one, that is designed to trigger when avalanche condition sets in and is designed to survive.

A normal NMOS, however, can also trigger, and if you have not thought about limiting the electron current, it can die, with IC killing consequences. Specifically, the drain and source will be shorted by likely the silicide on top of the drain, and instead of a transistor with high output impedance, we'll have a drain source connection with a few kOhm output impedance.

Take a look at New Ballasting Layout Schemes to Improve ESD Robustness of I/O Buffers in Fully Silicided CMOS Process [2] for the pretty pictures you'll get when the drain/source breaks.

But I just want a digital input, what do I need?

Even if it's only a digital input, you still need to consider ESD events.

Below is a complete digital input network.

Assuming we have a QFN package package there will be a bond-wire from the package metal, to our die pad.

Right after the die pad, sometimes under, there will be a primary ESD protection that can conduct, in all directions, between input, supply and ground.

From the input it's common to have a resistor to reduce the probability of currents going towards the core area.

Before we get to a transistor gate oxide it's common to have a set of secondary protection circuits. A resistor further reduces the current, and two local clamps (GGPMOS and GGNMOS) ensure that the voltage across the transistor gate does not go to breakdown levels.

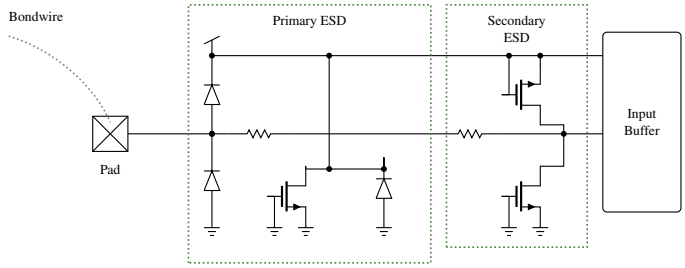

Figure 12: Full protection of an input including secondary protection

Input buffer

An input buffer can be seen below. I like to include a RC low-pass filter to filter out the RF frequencies (I don't want my input to toggle if a phone is on top of my circuit).

After the RC filter we need a Schmitt trigger, you can find a Schmitt trigger at JNW_TR_SKY130A.

The Schmitt trigger must be with thick oxide gates and with IO supply (for example 3.0 V).

The first inverter must also be a thick oxide inverter, however, the supply of the inverter will be core supply (for example 1.2 V). The thick oxide inverter provides a level-shift to core supply.

The last inverter is just to get the polarity of the TO_CORE signal the same as the input.

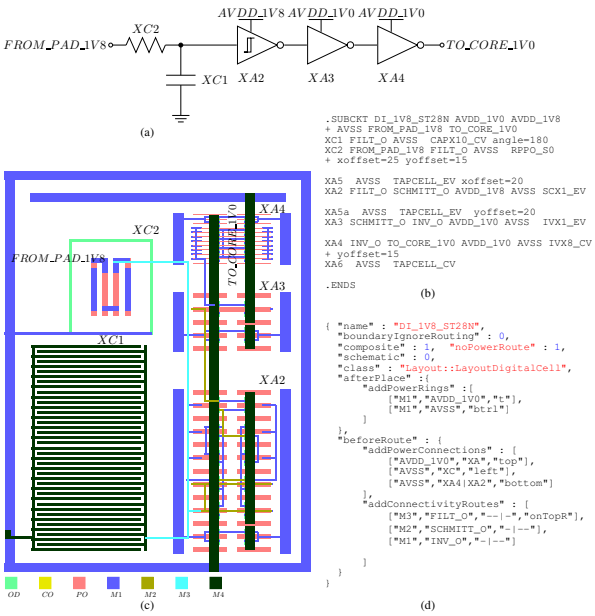

Figure 13: Full digital input including RC filter, Schmitt trigger and level shifters: (a) schematic, (b) netlist, (c) the compiled layout, and (d) the ciccreator recipe that generated it

Latch-up

Another fun physics problem can happen in digital logic that is close to an electron source, like a connection to the real

world, what we call a pad. A pad is where you connect the bond-wire in a QFN type of package with wire-bonding

Assume we have the circuit in Figure 14. Under certain conditions we can get a short from VDD to ground.

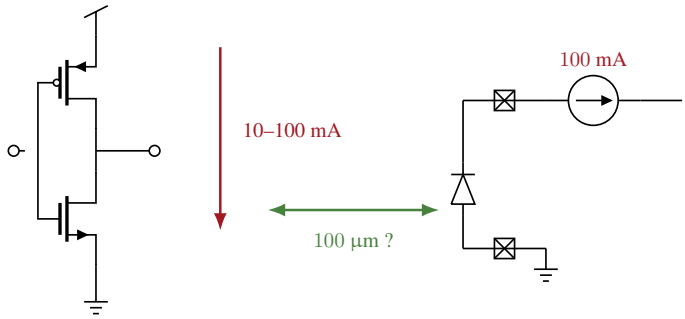

Figure 14: Inverter that suddenly shorts from VDD to ground located close to a PAD

Consider the cross section of the inverter in Figure 15. The latch-up process starts with electron injection (1), then forward bias of PMOS source/drain junction (2), forward bias of NMOS source/drain junction (3) , and finally positive feedback .

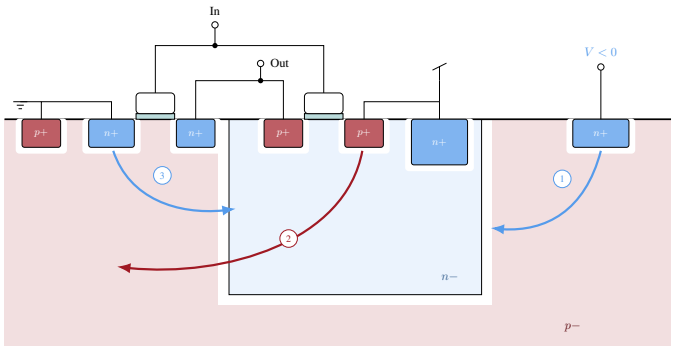

Figure 15: Cross section of an inverter

Electron injection

Assume that we have an electron source, for example a pad that is below ground for a bit. This will inject electrons into the substrate/bulk (1) and electrons will diffuse around.

If some of the electrons comes close to the N-well depletion region (2) they will be swept across by the built-in field. As a result, the potential of the N-well will decrease, and we can forward bias the source or drain junction of a PMOS.
Forward biased PMOS source or drain junction: With a forward biased source/bulk junction (2), holes will be injected into the N-Well, but similarly to the GGNMOS, they might not find a electron immediately.

Some of the holes can reach the depletion region towards our NMOS, and be swept across the junction.
Forward biased NMOS source or drain junction: The increase in hole concentration underneath the NMOS can forward bias the PN diode between source (or drain) and bulk. If this happens, then we get electron injection into bulk. Some of those electrons can reach the N-well depletion region, and be swept across (3).
Positive-feedback: Now we have a condition where the process accelerates, and locks-up. Once turned on, this circuit will not turn off until the supply is low.

This is a phenomenon called latch-up. Similar to ESD circuits, latch-up can short the supply to ground, and make things burn.

That is why, when we have digital logic, we need to be extra careful close to the connection to the real world. Latch-up is bad.

We can prevent latch-up if we ensure that the electrons that start the process never reach the N-wells. We can also prevent latch-up by separating the NMOS and PMOS by guard rings (connections to ground, or indeed supply), to serve as places where all these electrons and holes can go.

Maybe it seems like a rare event for latch-up to happen, but trust me, it's real, and it can happen in the strangest places. Similar to ESD, it's a problem that can kill an IC, and make us pay another X million dollars for a new tapeout, in addition to the layout work needed to fix it.

Latch-up is why you will find the design rule check complaining if you don't have enough substrate connections to ground, or N-well connections to power close to your transistors.

Similar to the GGNMOS, this circuit, a thyristor can be a useful circuit in ESD design. If we can trigger the thyristor when the VDD shoots too high, then we can create a good ESD protection circuit.

See low-leakage ESD for a few examples.

A model with the parasitic bipolars can be seen in Figure 16. The resistors in the picture is to emulate what happens when there is a current injected into the base of the NPN or PNP. I would recommend that you think through the physics instead of using the parasitic bipolar circuits. I've found the parasitic bipolar leads you down the wrong path when you actually want to understand the physics of latch-up.

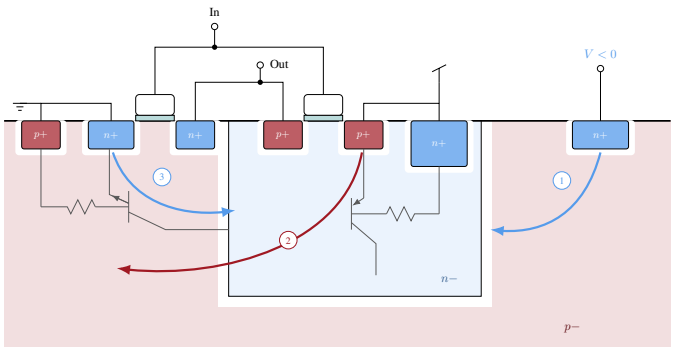

Figure 16: Cross section of an inverter including the parasitic bipolars

You must **always handle ESD** on an IC

- Do everything yourself

- Use libraries from foundry
- Get help www.sofics.com

Summary

The one-page version of this chapter:

- An ESD zap is a real event with a name: HBM is a charged person (kilovolts, amps through 1.5k), CDM is the chip discharging itself (nanoseconds, brutal)
- Protection is a promise about every ordered pair of pads: each of the six permutations on a three-pad chip needs somewhere for the current to go
- Two diodes per pin plus one rail clamp cover all permutations - adding a pin costs two diodes, not six paths
- The grounded gate NMOS is off by every model you were taught; the parasitic lateral bipolar is what sinks the amperes
- Thin gate oxide cannot take the leftover voltage, so real inputs add secondary protection behind a resistor
- Latch-up is the same parasitic bipolars firing in normal operation: keep well taps close, keep injectors away from wells

Would you like to know more?

ESD (Electrostatic Discharge) Protection Design for Nanoelectronics in CMOS Technology [3]

Overview on Latch-Up Prevention in CMOS Integrated Circuits by Circuit Solutions [4]

Overview on ESD Protection Designs of Low-Parasitic Capacitance for RF ICs in CMOS Technologies [5]

[1] Z. Huang, Z. Tang, X.-P. Yu, Z. Shi, L. Lin, and N. N. Tan, "A BJT-based CMOS temperature sensor with duty-cycle-modulated output and ±0.5°c (3σ) inaccuracy from −40 °c to 125 °c," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 8, pp. 2780–2784, 2021, doi: 10.1109/TCSII.2021.3068283.

[2] M.-D. Ker, W.-Y. Chen, W.-T. Shieh, and I.-J. Wei, "New ballasting layout schemes to improve ESD robustness of i/o buffers in fully silicided CMOS process," *IEEE Transactions on Electron Devices*, vol. 56, no. 12, pp. 3149–3159, 2009, doi: 10.1109/TED.2009.2031003.

[3] M.-D. Ker, "ESD (electrostatic discharge) protection design for nanoelectronics in CMOS technology," in *2006 advanced signal processing, circuit and system design techniques for communications*, 2006, pp. 217–279, doi: 10.1109/ASPCAS.2006.251127.

[4] M.-D. Ker and Z.-H. Jiang, "Overview on latch-up prevention in CMOS integrated circuits by circuit solutions," *IEEE Journal of the Electron Devices Society*, vol. 11, pp. 141–152, 2023, doi: 10.1109/JEDS.2022.3231822.

[5] M.-D. Ker, C.-Y. Lin, and Y.-W. Hsiao, "Overview on ESD protection designs of low-parasitic capacitance for RF ICs in CMOS technologies," *IEEE Transactions on Device and Materials Reliability*, vol. 11, no. 2, pp. 207–218, 2011, doi: 10.1109/TDMR.2011.2106129.

Carsten Wulff received the M.Sc. and Ph.D. degrees in electrical engineering from the Department of Electronics and Telecommunication, Norwegian University of Science and Technology (NTNU), in 2002 and 2008, respectively. During his Ph.D. work at NTNU, he worked on open-loop sigma-delta modulators and analog-to-digital converters in nanoscale CMOS technologies. In 2006-2007, he was a Visiting Researcher with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. Since 2008 he's been with Nordic Semiconductor in various roles, from analog designer, to Wireless Group Manager, to currently Principle IC Scientist. From 2014-2017 he did a part time Post.Doc focusing on compiled, ultra low power, SAR ADCs in nanoscale technologies. He's also an Adjunct Associate Professor at NTNU. His present research interests includes analog and mixed-signal CMOS design, design of high-efficiency analog-to-digital converters and low-power wireless transceivers. He is the developer of Custom IC Compiler, a general purpose integrated circuit compiler, and makes the occational video on analog integrated circuits at https://www.youtube.com/@analogicus. For full CV see https://analogicus.com/markdown-cv/.