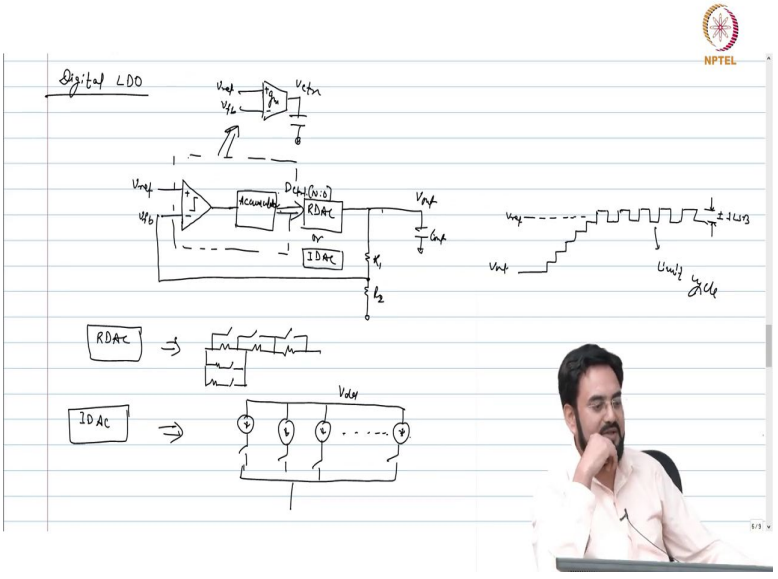
Power Management Integrated Circuits
Dr. Qadeer Ahmad Khan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture – 29
Digital LDO, Technique to Avoid Limit Cycle Oscillations in Digital LDO

(Refer Slide Time: 00:15)



Digital LDO. So far we have seen we use an error amplifier in the feedback, we need to compensate it and that error amplifier generates your control voltage which controls the gate of your pass transistor; which basically, is nothing, but your voltage-controlled current source. And depending upon what load current is, that control voltage will move up or down and will ensure that your output remains constant for that particular load.

Now, if I want to do the same thing digitally; how can I do, what do I need to implement in digital? So, any analog circuit if you want to implement in digital, what is the first thing you require?

You require ADC, first we have to digitize. So, you cannot now directly, you cannot feed your V fb now, correct? So, I will start with a very simple circuit. So, I will just tell you the concept of digital and you can make it as complicated as possible, depending upon what performance you require and all those, so, you will keep adding. But if you want to make it

more complicated, then it will require a digital filters and then it will take you to one level higher.

But the simplest method is, you just make it a first order system, it will simplify everything; but obviously, your transient response may not be that good. But sometimes if you are not concerned about your bandwidth, only regulation and you are looking for very small area and a small current, this method will always work. So, I have a comparator V ref you have a plus minus. Let us say this is my feedback voltage, so R 1 R 2, this is my V out.

So I need something in between. So, what I mean; let us consider this is your now, this is your comparator; but in the analog case, this was your error amplifier. So, that was driving your pass transistor. Now, this is comparator, which is giving me either 0 or 1 ok; what do I need here?

You need an accumulator, so this is generating your error, which is single bit error, ok. Now, you have to accumulate that error and convert to a code, ok. If you do not want to do that accumulator then use a flash; but depending upon resolution you require that many comparators, 2 to the power n minus 1, that will bloat your area. So, that is why I was saying you can make it as complicated as possible, depending upon your requirement. If you need a very fast, you just use flash and that will do your work.

So, basically this comparator combined with an accumulator is equivalent to your analog integrator with limited resolution, you can say. So, this is behaving like an integrator. So, it compares your reference with the feedback voltage and then. So, in the, this was generating V ctrl; now this will generate a code I will call it D ctrl, digital code, depending upon how many bits you have, ok, so n plus 1 bit. Now, what else I require? What do I need to, where should this go, this code?

So, let us say I use R DAC, resistive DAC, ok, this is the one way. So, this resistor can be implemented using MOSFET or you can have a resistor in series with the switches. So, depending upon what is your load current, you turn on or off these many resistors. So, if load current is high, you reduce this resistor; if load current is low. So, this is the one way, or what else I can have? You can have I DAC, current DAC also.

So, your MOSFET, the pass element was voltage-controlled current source. Now, if we put a current DAC, so, the voltage controlled current source now I DAC behaves like a digitally controlled current source that is it. So, you have parallel currents. So, your R DAC is nothing, but. So, your R DAC is basically. So, you have switches and you can have parallel combination also, whatever or you have R - 2R DAC. So, I am just drawing a very simple. So, depending upon whether you want to reduce or you want to increase; you connect in series or parallel and you can increase or decrease.

And I DAC is this is quite simple, you have this, ok. So, this could be your. So, you can have a binary weighted or linear whatever, ok. See if a binary weighted, then you have a I, then we have next will be 2*I, 4*I, 8*I; the weights will be like that. If you have linear, then each elements could be LSB, 1 I. So, the only drawback with the R DAC is your PSRR, one way good, because it is a resistive, so, any ripple on that will not get filtered and it will appear with the resistor divider. But if you use a current DAC, then you know that R ds of these current DAC's, these will be operating in saturation, then you can get a better PSRR. So, that is a main difference between the two. One advantage with the resistive DAC is, you can operate your LDO at very low dropout voltage, ok, because it's already in linear, so it does not matter what dropout you. But this will go in linear, if you operate below certain voltage.

So, and these resistors can be replaced with the simple MOS switches also, on off switches and those R ds on of those switches will define the. So, when you turn on those switches fully that will be linear actually. And in linear triode region whatever the resistance is that, will define the unit resistance of that and you connect them in parallel and series, you get the value depending upon your. So, this loop will automatically take care of that. So, the when your V f b becomes V ref; what will happen to the comparator output?

So, what will happen actually. So, let us say you are starting with a 0. Your V out you starting with 0, it will keep stepping up, ok. And let us say my V ref is here somewhere. My V ref is here, ok. So, now, this was still low; but if you put the next, then it crosses. So, the comparator output will be higher, here your comparator output will be low ok; then in the feedback this is low, again it will turn high. So, it will keep doing actually this; high low high low, ok. And what will be the size of this? What will be the size of this?

Plus minus 1 LSB, worst case, ok. So, if you are at the very boundary actually, you can jump up by 1 LSB and you can go down by 1 LSB. So, you can always have plus minus 1 LSB error. So, you can have either 1 LSB or if it is moving around mid court; let us say around the middle, then average may lie exactly at V ref, but you will still see a plus minus 1 LSB. So, this can be plus 1 LSB or minus 1 LSB depending upon where it is settled.

So, this can only be reduced, this ripple can only be reduced by increasing your resolution, and this is called limit cycle, and this amplitude increases if you. So, what happens actually, you in order to get this, this is the best you can achieve here; but in order to get this, your bandwidth should be very low. Why? So, if you move around the loop, what you are seeing; your accumulator is very slow, ok.

So, which means your accumulator has to wait until your output settles, before you take any decision; which means, your accumulator has to be very slow. So, your dominant pole should be at the accumulator, because this is an integrator, that is why it becomes a first order system; which means, your output pole should be outside your UGB. If you bring your output pole inside UGB, what would happen?

So, your let us say output has changed. So, when you change your a accumulator changes the code, your output will change and that output has not settled and you have changed the code of the accumulator and there is some lag due to the pole. So, what you are doing actually; your accumulator will keep counting up and up and it will increase the code, before the output settles and when the output starts going low, in that it will realize I have increased too much now I have to go down, so your.

Student: Sine wave.

It will not be sine, it will look like a triangular waveform actually. So, instead of plus minus 1 LSB, this amplitude will, limit cycle amplitude will start increasing, it may go to plus minus 2 LSB, 3 LSB, 4 LSB. So, it will keep increasing as you bring your output pole near UGB. And will you bring it inside the UGB, it may become almost unstable and you will get very large oscillations at the output; that is why you have to make sure your output pole is outside

your accumulator output, which means update rate of the accumulator should be slower than, much much slower than R C time constant of the output.

And we know that it usually takes 4 to 5 tau's to settle the output nearly 100 percent. So, you have to wait for 5 tau. So, you know whatever the output pole is, and at least wait for 5 tau before you update the accumulator; because otherwise it will increase your oscillations, limit cycle oscillations, ok. So, it is just a first order system that is it. If you want to make it faster; then instead of single bit ADC, which is your comparator, you use the higher order oh sorry, they use the flash basically.

But that will come at the cost of area and power. So, if I want to avoid this, let us say. So, I have ensured that my output pole is outside your UGB; which means, accumulator is very slow, that I have ensured. But I will still get plus minus 1 LSB that oscillation; if I want to reduce this oscillation, then what can I do?

Student: Reduce the ripple voltage ripple.

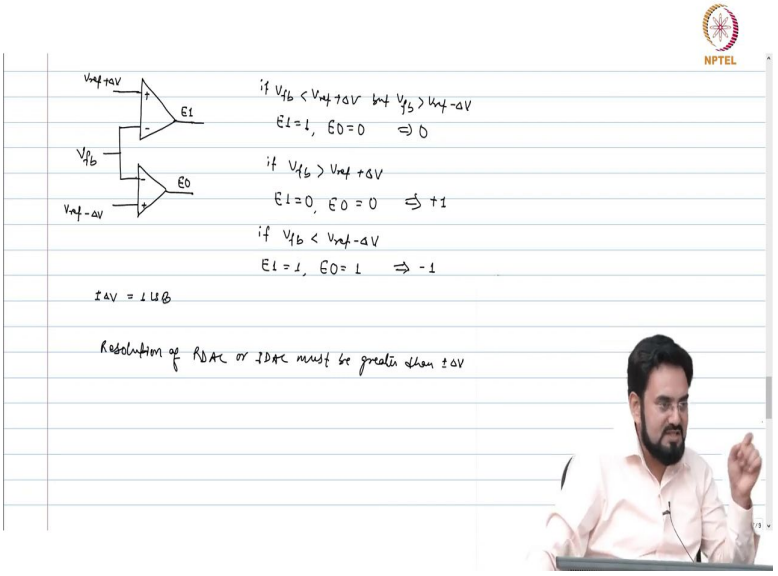
I want to just completely eliminate this ripple, can I do this?

Student: We can use 2 LDOs with like.

Two LDOs why?

You just use two comparators, windowed comparators you use and if your resolution of your R DAC or I DAC is more than that window; then it will settle. This problem is happening, because this is a single bit, it is a 0 or 1, it has only two levels to go. Now, if you have two comparators, which is 2 bit ADC.

(Refer Slide Time: 15:35)



So, if this is your V ref plus delta V and then you have V ref minus delta V. If this is your V fb, so, what will happen? Let us see the code. So, if I call it E 0, not E 0; E 1 and E 0. So, if your V fb is higher than V ref plus delta V; what will be the code? So, V fb is greater than V ref plus delta V, E 1 equal to 1 and E 0 equal to 1.

Sorry, V fb is less than, and if V fb is greater than V ref plus delta V, then, then what will happen? This will go to 0. So, let us change this polarity, let us make it minus and this plus. So, this E 0 will go to … , correct, but … , correct? So, this is more than your lower threshold, but less than the upper threshold. So, you get one 0. Now, if V fb is greater than V ref plus delta V; then this will go to 0, E 1 will be 0 and E 0 will be what? 0. If it is greater than V ref plus delta V, then it will be greater than V ref minus delta V also. If V fb is less than V ref minus delta V, then what would happen?

E 1 equal to 1 and E 0 equal to 1. This is less than this. So, plus is more, so this will be one; in both the side then negative is less than positive. So, what does it mean? If I define this window. So, if this plus minus delta V equal to 1 LSB, correct? Plus minus delta V means, you define two delta V as 1 LSB. So, if you know 1 LSB, then you break that into two halves and one make V ref plus 1 LSB by 2, other make other V ref minus half LSB. So, now, what would happen?

We know my comparator output will only change. So, my accumulator I want to change only when it crosses that window; when my V fb remains within this window, what I will get at 1 0. So, if I make sure my accumulator code only changes, when I get both 0 or both 1. So, when it is both 0, I treat it as minus 1 ok; both 0 means low, ok. You have to increase it.

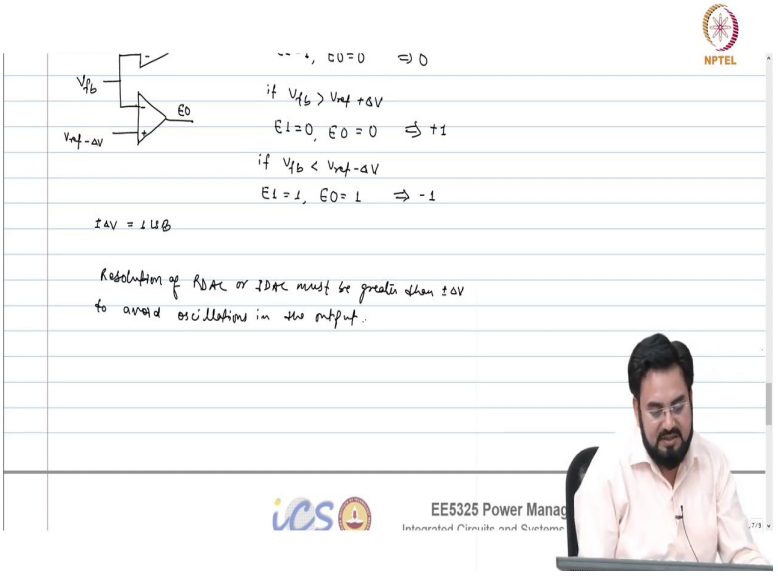
So, polarity you can decide whatever you want actually, you have to just make sure; when both are 0 you have to increase the code, when both are 1, I have to reduce the code, correct; and when it is in between, I freeze the code. So, now, you have three levels. In fact, you have four levels here, but you will get three levels here. So, these are three possible outputs you can have. So, now you have, you treat it as 0, you treat it as plus 1, you treat it as minus 1, ok.

So, whatever the accumulator code is you add 0 or you add plus 1 or you add minus 1, in the accumulator. So, if you are adding 0, your accumulator output will not change; if you are adding plus 1, it will increase; if you are adding minus 1, it will decrease. So, which means, if I ensure that my resolution of my DAC, R DAC or I DAC is more than this window; then when it freezes or when you get a. So, you will always get a 0; but if your resolution of R DAC or I DAC is less than this window, then it will always cross that boundary and your code will be jumping between 00 and 11 and you will get the same thing - what you are getting with the single comparator.

So, that is why the requirement here is resolution of R DAC or I DAC must be greater than let us say greater than plus minus delta V. Your accuracy you are looking for 1 millivolt, then you keep your I DAC or R DAC resolution half a millivolt or so. So when you change one resistor here or one current, the output should not change by more than half a millivolt; then your output will settle.

So, it can settle to any value within that 1 LSB, and you will always get a error, may get an error of, may or may not get actually; it may settle at a 0 error also depending upon where it lies. So, but in the worst case, you will get a 1 LSB error; but your output will not oscillate. So, that is the advantage with this.

(Refer Slide Time: 23:51)



To avoid oscillation in the output, ok. So, how will you define the worst case LSB here or resolution; at a maximum load or minimum load? Your LSB will be.

Student: At minimum load.

Because at minimum load your R out is very high. So, let us say you are using an I DAC.

Student: I DAC minimum load will be R DAC maximum load.

R DAC maximum load.

Student: Now because of output load.

So, for I DAC your R out, in both the cases your R out is very high at light load, you know, which means 1 LSB change in the code or in your DAC should not make more than 1 LSB change in the output, the delta V whatever it is ok, it should be less than that. So, in order to ensure that, so one thing and I know if R out is high even a small change in the current will make a large change in the output. So, your minimum size of the current will be decided by the lightest load.

Similarly, a higher value resistor; if you add a higher value resistor or remove the higher value resistor, it will change the current with a smaller value, ok. So, and your LSB the

accuracy will be decided by the highest value resistor you are choosing, that will define your LSB size, the highest value resistor ok; because that will cause a minimum change in the current. So, then you have to decide what will be the resistor based on your lightest load. When you go to the highest load, that resolution will further improve, ok.

So, if your wider is the load range, you may require a very higher accuracy here; the number of bits may be larger, because I have to start with the lightest load and when I have to cover the maximum load also. So, let us say your step size a lightest load is 10 microampere and you are looking for let us say, roughly about uh, 1 percent kind of accuracy. So, at 10 microampere, you are looking at a 10 bit kind of accuracy at 1 micro, at 10 microamp load ok; which means, when full all the currents are on, you will be supplying 10 microamp current.

Now, you have to supply a 10 milliamp current. So, how many, you require, 10,000 of these roughly. So, which means your, this current, the resolution requirement here might be very high; not just a 7 bit, but you may require 12 bit, 13 bit kind of, to cover the entire range of the load, ok. But if your load is fixed then it is not a problem, but if your load is varying a lot, then the requirement for resolution would be very high in that case, ok.

If it is let us say 1 milliampere fixed load you are designing, then it is very straightforward; 1 milliamp divide by 100, you know it is a 100 microamp, no 10 microamp, ok. So, 10 microamp is the smallest size, you just use uh, 100 uh, currents in parallel of 10 microamp and you are done. But this will not work if your load is varying, so you have to cover both higher and lower. So, you have to maintain the accuracy as well as maintain the highest load current.

So, there is one more technique we use; we use hybrid LDO, where you have a digital code and in the parallel you add a analog loop also. So, what do you do actually? So, let us say you have a same thing and add, take this feedback and add a error amplifier everything same as analog; but that pass transistor will be very small. So, what do you do? Whatever your load requirement is, you cater through this digital code and 1 LSB or 2 LSB load, you make, shift to the analog. In that case whatever error is left by this, that will be corrected by the analog loop.

So, you will get a resolution as good as analog, but you will get even more like a mixed signal kind of technique where digitally. So, and since that pass transistor of that analog is very small. So, you can make that high bandwidth. So, your digital will be low bandwidth and analog will be high bandwidth, it is like that, ok.

So, your, because we know that resolution will be limited by your LSB size. So, you can relax the LSB size here and get it corrected through an analog loop by making a small analog regulator. So, that is also a technique sometimes we use in hybrid LDO.