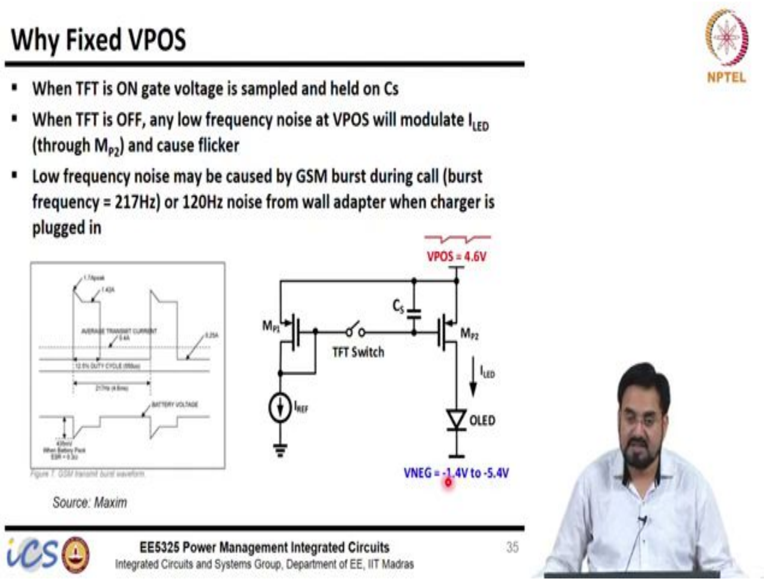
**Power Management Integrated Circuits**

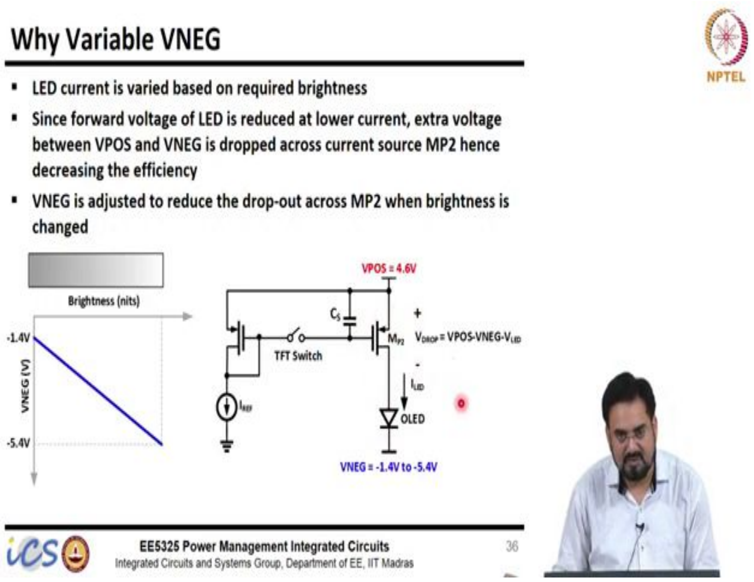
**Dr. Qadeer Ahmad Khan**
**Department of Electrical Engineering**
**Indian Institute of Technology, Madras**

**Lecture - 100**
**LCD/AMOLED Display Drivers – Part 3**




The main reason in order to get the constant or fixed VPOS of 4.6 volts is that we cannot afford any noise here and since we are refreshing this OLED pixel at a particular frequency or refresh rate when we are not sampling the current any changes in this VPOS will basically cause a change in the current. If it's a high frequency it's quite possible that this capacitor will act as a short and it may try to suppress some of the noise, but any low frequency may not get suppressed.

So, we may get a flicker, and if it's the order of let us say 100 Hertz or below we can easily get a flicker which will be visible to our eyes. So, that is why we need a very clean 4.6 volt and VNEG is programmable from minus 1.4 volts to minus 5.4, as I mentioned some of the newer devices require as low as minus 0.8 volts.

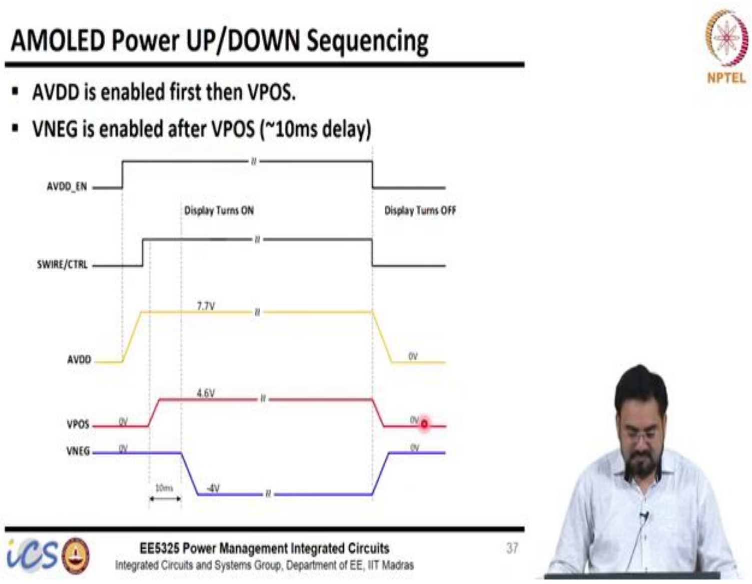



Let us see why we require a variable negative voltage. So, when we are changing the brightness what we are doing is we are changing the current, we know LED requires constant current as we saw in the backlight LED driver. See your brightness can be controlled by simply changing the current. Let us say we are reducing the current which means we are reducing the brightness. We know the forward voltage is depending on your current, the higher the current more the forward voltage this OLED will have. So, when I reduce the current this LED or organic LED will require a lesser forward voltage.

If I do not change and let us say I keep it fixed to minus 5.4 and this is fixed to 4.6 volts so, the total voltage from positive to the negative terminal will be 10 volts and we know this is the current source so it will require some headroom in order to keep in saturation and that may not be very large depending upon these are high voltage devices. Obviously, it cannot be 100 millivolts. It will be slightly larger so let us say roughly 500 millivolts.

So, You require 500 millivolts minimum $V_{ds}$ here. Out of this 4.6 volt, 500mV will be dropped here. This LED will see 4.1 volts and in order to simplify, let us say you will get 4 volts here. So, you are getting 9.4 volts across this correct. When I reduce the current let us say the forward voltage is reduced to 5 volts. So, what does it mean, you need to reduce this by 4 volts. So, when you have 4 volts here then you require minus 1 volt only at the other end, not minus 5 volts. Your current will still be constant. What would happen if you keep this fix at minus 5 volts? This LED forward voltage will not change and depending upon the current it will adjust itself, because it's having a constant current.

So, whatever the extra voltage we have will be absorbed by this but the problem here is since this current is constant so drain voltage has to drop. We are having a constant current so we know that this voltage has to go low when you are reducing the current because this LED has to maintain its forward voltage. The $V_{ds}$ across this MOSFET will increase, which means the dropout voltage is increased. If the dropout voltage is more then there will be higher losses. It will impact your efficiency. So, do not misunderstand that this voltage is controlling the brightness, brightness is still controlled by the current. So, this voltage has nothing to do as such with controlling the brightness, but mainly due to efficiency reasons we want to control this voltage when we are controlling the current in order to change the brightness. So that we can maintain the dropout voltage across this MOSFET.

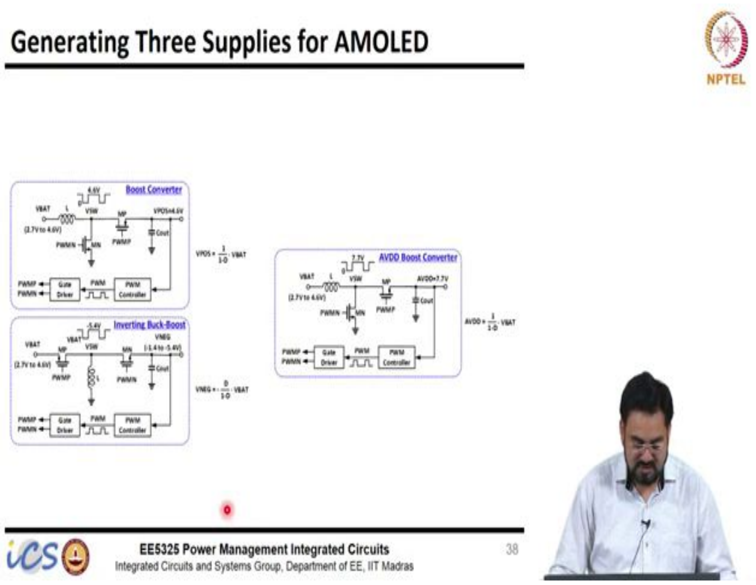



Just like in LCD, there are certain power up and power down sequencing requirements here as well. AVDD is enabled first and then we enable positive voltage which is VPOS and then VNEG. The VPOS and VNEG follow a similar sequence as we had in LCD. AVDD is analog voltage and it comes up first so it settles all the bias voltages before we turn on these LEDs.

AVDD is regulated at 7.7 volts and this is control voltage SWIRE. Just like in the backlight we get a separate enable signal for backlight there is a separate enable signal for AVDD, but your positive and negative supply is controlled by a single enable signal which we called

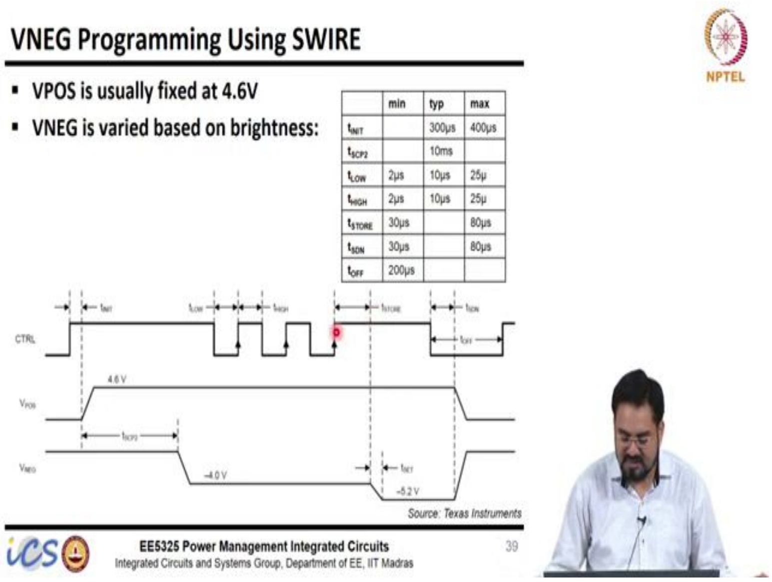
SWIRE or CTRL signal. And the same signal is used to program this negative supply also so it is a single wire control that enables as well as controls the level of this negative supply.

So, by default VPOS will go to 4.6 volts. It will remain there and by default, this VNEG goes to minus 4 volts. If you do not do any programming which means you just turn on then by default it will go to minus 4 volts.




How do we generate these 3 supplies? The concept remains the same just like in LCD we require one boost converter to generate 7.7 volts and another boost converter is required to generate 4.6 volts because we know the battery voltage is 2.7 to 4.6 volt. Most of the time it is below your output voltage so you need to boost, in this case, it is always higher than your inputs so you need to boost here as well and for negative supply, we use inverting buck boost. So, we cannot use a charge pump here, the reason is the current requirement is higher like 300 milliamp or so. It is not easy to build a high current charge pump.

Another issue is its programmable. So, we need to program it and with the charge pump, we need a fixed voltage that is discrete and not continuous. You can use an LDO, use a charge pump, and then use an LDO to drop the voltage, but it will be highly inefficient. So, that is why we always use an inverting buck boost here.



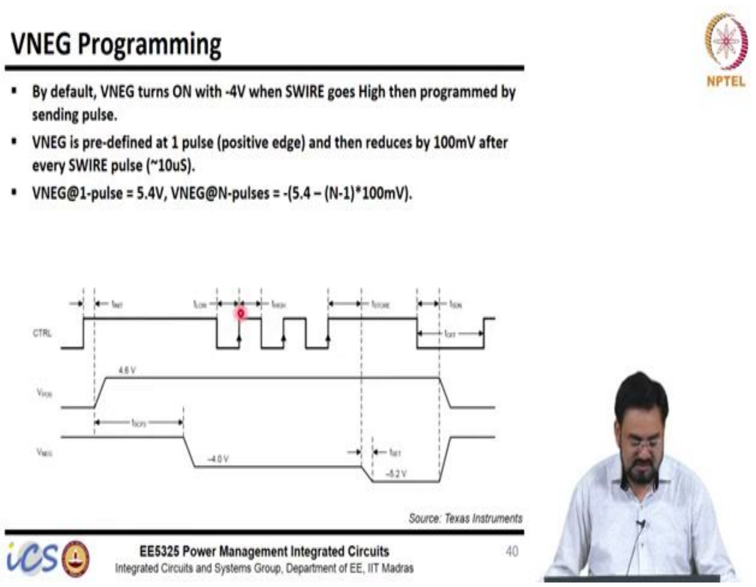

So, let us see how we program this negative voltage using SWIRE or the control signal(CTRL). There are some timing requirements, initial timing($t_{init}$) timing is 300 microsecond typically. $t_{init}$ is the time delay which is given before you even turn on this 4.6 volt or positive supply and then after that you have the short circuit protection. So, the 10-millisecond wait is given before you turn on the negative supply in order to make sure that the short circuit is not there.

Let us say somebody has shorted. Your supplies will not ramp up. You can say, it's more like a time out signal. When the 10 milliseconds are passed and still your supply is not ramping up. We know that we need to turn on positive supply first and negative later. We do not want any instance where negative is turned on and positive is not there. So, let us say there is a short circuit at the positive voltage so it will not ramp up at all. If you are not monitoring VPOS and you just give a 1-millisecond delay and after that, you turn on negative supply. So, what will happen is positive supply will not come up and negative will go high and there is a risk of damaging your panel here. That is why we give this much time so we wait for 10 milliseconds and keep monitoring this positive supply if it does not come up we flag the short circuit, if it has already come up within ten milliseconds then we turn on the negative supply.

Then this is your $t_{low}$ which is nothing but off time. The way it is programmed you sent pulses. The first positive edge will be acting as an enable and after that, any positive edges will act as a programming signal. And there is some timing requirement such as these pulses should not be wider than 25 microseconds and shorter than 2 microseconds. Similarly, we have $t_{high}$. It has the same timing as t_low.

What happens once you send the pulses. So, let us say I send this 10 microsecond ON and 10 microseconds OFF pulse. In order to get the first pulse, you have to first pull it down and then wait for the minimum pulse width required. Let us say I fixed it 10 microseconds so you keep it low for 10 microseconds and then keep it high. So, this will be treated as the first stage of the programming, and then after that, it will keep counting, and the reason we have a maximum of 25 microseconds because 30 microsecond is the STORE. So, after 30 microseconds if no pulses come then it will be acting as STORE. So, within 30 microseconds if you get let us say one more pulse then it will keep programming and after 30 microseconds only it will latch the output. So, let us say you have a counter and you keep counting this. So, here we have 3 rising edges. So, after the last edge, you wait for 30 microseconds and you see if there are any edges. If no edges are coming then you latch the output and then set the negative voltage.

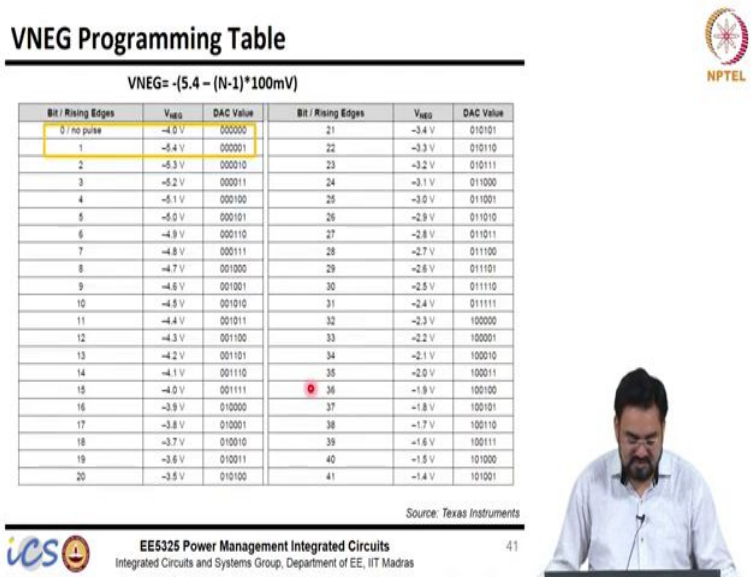
Then when you have a negative edge then again you wait for 30 microseconds and if it remains low for more than 30 microseconds then it will be treated as shutdown. Because this is acting as both enable and disable.




So, by default VNEG turns on with minus 4 volts when SWIRE goes high and then it is programmed by sending those pulses where you count the positive edges of each pulse. So, VNEG is predefined at 1 pulse (positive edge) and then reduces by 100 millivolts. So, what it does basically, 1 pulse is internally programmed by default. We know it will go to minus 4 volts and there is one more thing actually. We know the difference between edges of positive voltage and a negative voltage is roughly 10 millisecond. So, before a positive edge comes if you do any programming then it will by default go to minus 4 volts, but let us say you send some pulses here in between before it comes up then it will take the new value.

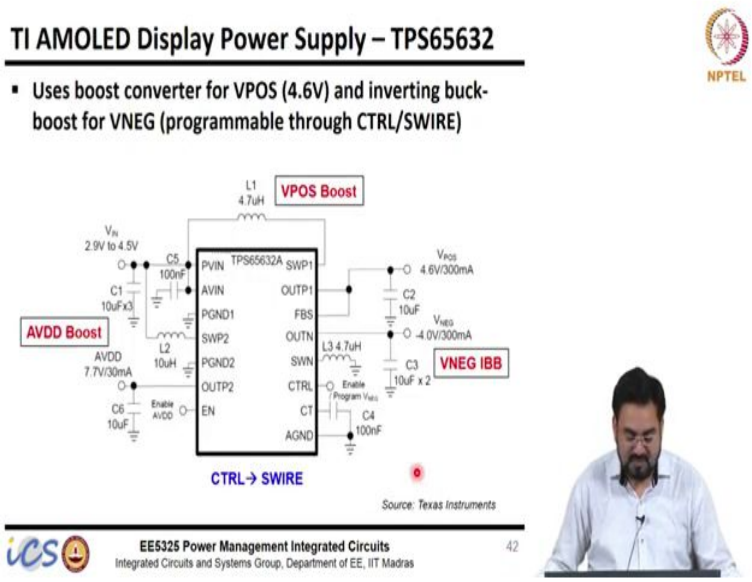
If you send 1 pulse then 1 pulse is already prefixed. I mean in hardware this programming is done. 1 pulse is already programmed to minus 5.4 volts. When you send only 1 pulse and let us say you keep it high after that which means you basically lap this 30 microseconds time and it will go to minus 5.4 volts. And then after that, every pulse will be reduced by 100 millivolts. So, this number of pulses will always be counted from a single pulse reference. We know 1 pulse is minus 4 volt. If I send 2 pulses then it will be reduced by 100 millivolts so you will get minus 5.3.

Whatever the one pulse means may vary from part to part actually. So, you have to look at the datasheet for what is 1 pulse they have set in there. It is not necessary that all devices will have minus 5.4 volts. One may have minus 4 volt other may have minus 5.4 volts. So, then after that whatever number of pulses you send and based on that you program. Let us say if I want this voltage to go to minus 5 volts so we have to send 5 pulses.



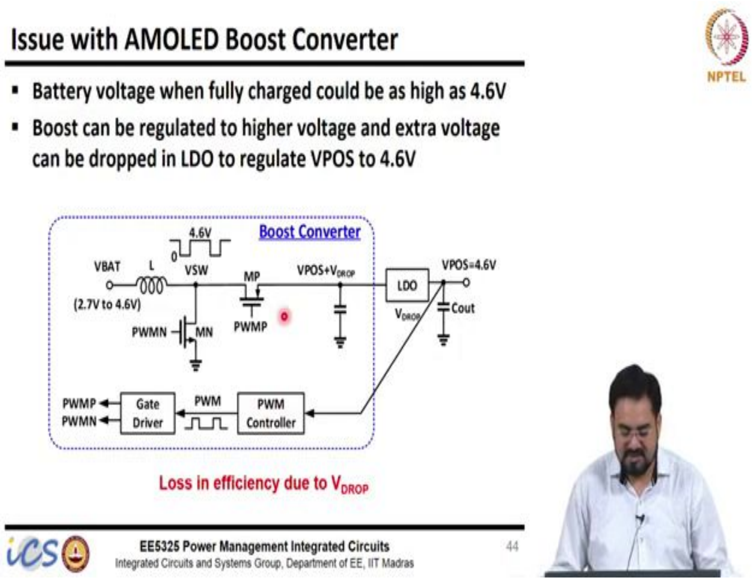

| Bit / Rising Edges | $V_{NEG}$ | DAC Value | Bit / Rising Edges | $V_{NEG}$ | DAC Value |
|---|---|---|---|---|---|
| 0 / no pulse | –4.0 V | 000000 | 21 | –3.4 V | 010101 |
| 1 | –5.4 V | 000001 | 22 | –3.3 V | 010110 |
| 2 | –5.3 V | 000010 | 23 | –3.2 V | 010111 |
| 3 | –5.2 V | 000011 | 24 | –3.1 V | 011000 |
| 4 | –5.1 V | 000100 | 25 | –3.0 V | 011001 |
| 5 | –5.0 V | 000101 | 26 | –2.9 V | 011010 |
| 6 | –4.9 V | 000110 | 27 | –2.8 V | 011011 |
| 7 | –4.8 V | 000111 | 28 | –2.7 V | 011100 |
| 8 | –4.7 V | 001000 | 29 | –2.6 V | 011101 |
| 9 | –4.6 V | 001001 | 30 | –2.5 V | 011110 |
| 10 | –4.5 V | 001010 | 31 | –2.4 V | 011111 |
| 11 | –4.4 V | 001011 | 32 | –2.3 V | 100000 |
| 12 | –4.3 V | 001100 | 33 | –2.2 V | 100001 |
| 13 | –4.2 V | 001101 | 34 | –2.1 V | 100010 |
| 14 | –4.1 V | 001110 | 35 | –2.0 V | 100011 |
| 15 | –4.0 V | 001111 | 36 | –1.9 V | 100100 |
| 16 | –3.9 V | 010000 | 37 | –1.8 V | 100101 |
| 17 | –3.8 V | 010001 | 38 | –1.7 V | 100110 |
| 18 | –3.7 V | 010010 | 39 | –1.6 V | 100111 |
| 19 | –3.6 V | 010011 | 40 | –1.5 V | 101000 |
| 20 | –3.5 V | 010100 | 41 | –1.4 V | 101001 |

So, this is your TI datasheet. They have a single pulse voltage. For no pulse, it is minus 4 volt and for 1 pulse it is minus 5.4 after that as the number of pulses increases this voltage will keep reducing. You can see that at 41 pulses it goes to minus 1.4 volts. These are the DAC values actually. So, these pulses will be counted and then will be latched and we know that these voltages can be programmed using DAC. Let us say we have a feedback resistor. You can just program the feedback resistor and get different values.




This is again a TI part TPS65632 which is used in most of the phones. It has 3 supplies. We can see here the left hand side AVDD boost of 7.7 volt and 30 milliamps. This only supplies analog supplies, it does not require very high current, and then right hand side you can see we have this positive boost of 4.6 volts and 300 milliamps and then this one is your negative buck boost which is minus 4 volt and 300 milliamps.

Minus 4 volt is default but it can be programmed using this control signal CTRL.




So, there is an issue with this positive AMOLED boost converter and we know that it requires a fixed 4.6 volt and we know that battery voltage is going from 2.7 to 4.6 volt. So, what would happen when you are very close to your maximum battery voltage then you can hardly regulate the output of 4.6 volts.

Even if you make let us say 0 percent duty cycle then you just bypass this. It is a boost converter so 0 percent duty cycle means $V_{in}$ equal to $V_{out}$. There will be some IR drop so it will be always less than your input. So, we can hardly regulate at 4.6 volts. So, that is the main problem and can be fixed by again boosting this voltage. If it is 4.6 volt, we can easily achieve more than 4.6 volts. Let us say you make it 4.7 volts and you drop 100 millivolts in this LDO just like we are doing for LCD. You can get 4.6 volts easily. We will obviously have some loss in efficiency due to this dropout voltage, but you have to live with it because that is the only way to regulate the output.

There is one more technique. Let us say this MOSFET is $M_P$ replaced with the diode so it will become non synchronous and in non synchronous the diode drop will be higher compared to the IR drop of this MOSFET because the ON resistance of this is much smaller. What we can do, when voltage $V_{in}$ is very high and we know that it's going out of regulation or your duty cycle is saturated to 0 percent or near 0 percent. You fully turn off this MOSFET and it will conduct through the body diode so it will enter into non synchronous mode and the drop will be more. You are dropping the voltage across that diode and you will get a similar functionality that you're achieving with this LDO. So, without adding this LDO you can still regulate the output by entering into non synchronous mode and conducting the current through the body of this MOSFET.