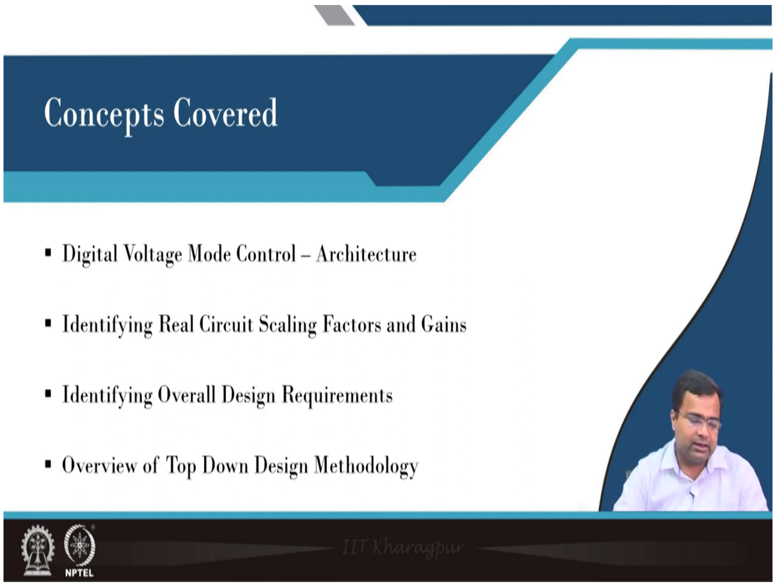
Digital Control in Switched Mode Power Converters and FPGA-based Prototyping

Prof. Santanu Kapat
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Module - 08
Digital Controller Implementation using Fixed-Point Arithmetic and Verilog HDL
Lecture - 71
Top Down Design Methodology in Digital Voltage Mode Control - I

In this lecture, we are going to talk about Top Down Design Methodology in Digital Voltage Mode Control, particularly using hardware descriptive language.

(Refer Slide Time: 00:33)



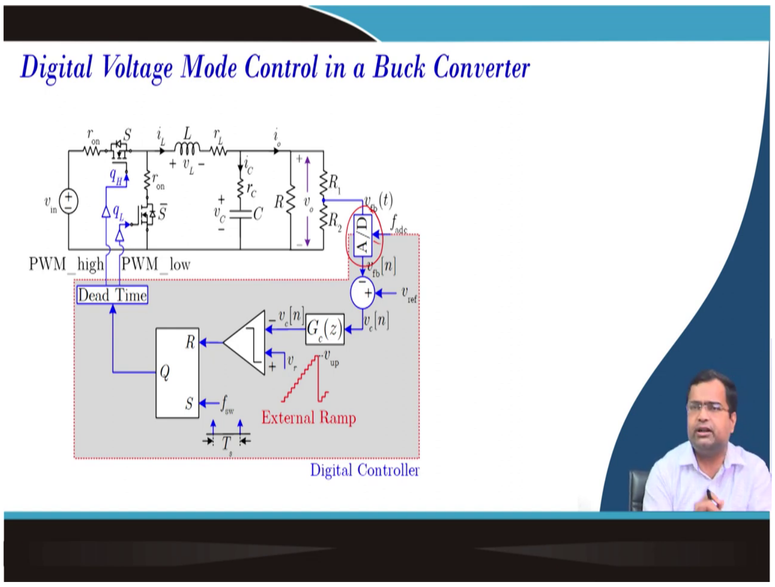
Concepts Covered

- Digital Voltage Mode Control – Architecture
- Identifying Real Circuit Scaling Factors and Gains
- Identifying Overall Design Requirements
- Overview of Top Down Design Methodology

IIT Kharagpur
NPTEL

So, here we will talk about digital voltage mode control architecture, then we need to identify what are the real circuit scaling factors, and gains. Then, we need to identify overall design requirements and some overview of top-down design methodology.

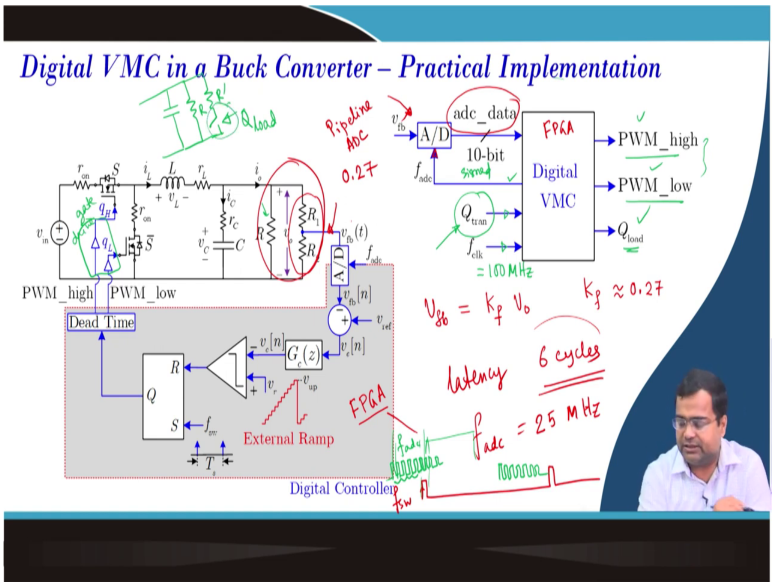
(Refer Slide Time: 00:47)



Digital Voltage Mode Control in a Buck Converter

So, in this particular architecture, we are talking about digital voltage mode control and this we have discussed multiple times in week 2 as well as week 3 in the MATLAB implementation, where we need A to D converters for the output voltage or sampling. And, then everything else in the digital domain so; that means, the data coming out of the digital that is coming, then all the reference signal, saw tooth waveform, everything we are generating inside the digital platform.

(Refer Slide Time: 01:18)



Digital VMC in a Buck Converter – Practical Implementation

So, here for practical constraint, we need to understand how is it linked. First of all when you take a practical converter for example, if we emphasize this part; so, there is you know first of all the scaling feedback gain, because we need to provide this voltage to the ADC input which must be consistent with the ADC span including the overshoot, undershoot, the startup behavior of the converter. So, the output voltage after scaling down should not go out of the range of the ADC that we have to; that means, that is one way the safety purpose.

Another is the mapping between the number, the voltage must be within the span of the ADC. And, that is why you need to select a suitable R 1, R 2 ratio and in this particular design, we are going to show hardware also in the subsequent lecture. We have considered this scaling fact scaling factor to be 0.27 which is; that means, I would say this voltage feedback which is the feedback voltage is some feedback gain by this output voltage, where feedback gain is approximately equal to 0.27 because, by the resistive divider, we are getting like this.

Now; that means if we consider this particular architecture and we are talking about now we want to develop; what we want to develop? We want to develop because this digital platform we are using is an FPGA; which means, a Field Programmable Gate Array device. This FPGA only can take digital data and it can only give rise to digital data; that means, it has I O pin which only accepts digital data in and out. So, this is the digital control where you are using FPGA.

What is coming to the input to the FPGA or the digital controller? For the data coming out of the ADC, we are talking about adc_data and here we are talking about ADC. We have discussed in the previous lecture when you talk about the signal conditioning circuit. Here, using we are using ADC, which is you know differential ADC, fully differential ADC, 10-bit ADC, 10 bit. This ADC requires a clock because we need to provide a sampling.

And, we are using a pipeline ADC from an analog device. So, it is a pipeline ADC. In pipeline ADC, the processing; means, you know the propagation delay or sampling or the conversion time depends on the number of clock cycles; instead of the absolute value you know the conversion time is several clock cycles. And, in this case, there are 6 clock cycles; that means, the throughput is the same as like you know flash ADC, but it has a latency of; so, what is the latency? That means a delay.

That is the sending the sample command and the data ready that takes around 6 cycles ok. So, now, since it takes 6 cycles, we have to be very careful about the frequency of the ADC because, if we give a lower frequency, it will take 6 cycles. So, the total delay of the conversion will be pretty large and that will create a lot of problems in your control loop; first of all the transient response. So, we need to provide a sufficiently high-frequency clock; so, that with 6 cycles, your delay will not be too large.

So, in our case, we are using f adc to be 25 megahertz, but another practical aspect because it requires 5, and 6 cycles to get the data ready from the start of the conversion to the end of the conversion. But, after this 6 cycle, we may not want to continue this sampling of the ADC because we do not need the output voltage, we need it again the next. For example, you know we have this switching frequency clock.

Let us say this is my switching frequency clock and we need to sample the ADC let us say just before it starts; that means, we are giving a high-frequency clock of this kind of clock 6 data. So, data will be kind of ready here. So, as if the data was captured as you know here, but we may or may not continue this clock, this is my f adc. If you continue then ADC is unnecessarily burning power when we do not need data. So, we have to be only careful, the 6 cycles are needed which will end close to that switching point. So, the data should be ready; that means, that is the first requirement.

So, after that you can do some kind of windowing operation where the ADC clock will be forced to 0; that means, there is no ADC clock going on. There is no activity, there is no sampling and you again start doing something somewhere here. Because, you need another 6 cycle data for the data to be ready, again you turn that way. But, for simplicity in this experiment, we are continuing 25 megahertz to the ADC continuously, but in real implementation, we can minimize the power consumption by this.

So, the adc_data is 10-bit data 10-bit signed data, and signed data. Now, here is the transient; that means, we are also making a transient; that means, for load here; that means, if we take the output capacitor, we are using a continuous load and we are also using a transient load. So, in this, we gate signal we are talking about Q load and that is the output; that means if this is high then this resistance. So, this is your continuous resistance, this is another resistance like R dash, they will come in parallel.

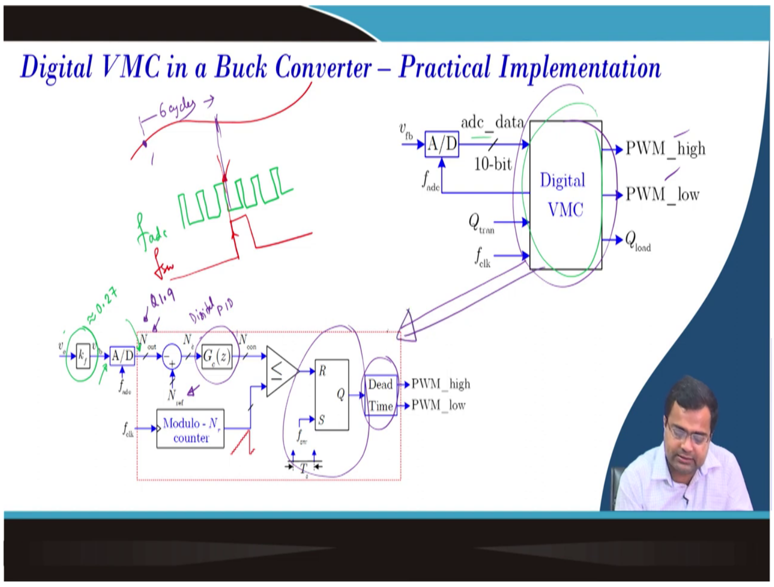
So, the effective resistance will be down, and smaller. So, as if it will look like a step-up transient. So, the net current will increase. So, it is a step-up transient and this PWM_high and PWM_low are going to the gate drive circuit and there is there will be a gate drive circuit, gate drive.

So, the gate drive will generate the actual gate signal for the MOSFET, but we are providing the high and low of this gate drive signal from the FPGA. Some, IC gate drive circuit also has it only accept only 1 PWM signal, and internally it generate the dead time for all other signals.

So, as you know it can be added up to dead time, but in this particular driver we are using where we are dedicatedly giving low side and high side gate signals and the driver circuit makes it compatible with this MOSFET to turn on and off. So that means, this high side and low side at the gate signal and in between, there should be some dead time. This Q tran we are giving is to select whether we want to make a load transient or whether we want to make a reference transient. So, that is also added.

Now, that means, FPGA input to the FPGA is the adc_data, input to the FPGA is the transient type that you detect, and the clock of the digital controller and this we are taking 100 megahertz, 100 megahertz clock and that will be the crystal oscillator outside. The output of the FPGA is a clock signal, and one is the high side and low side gate signal going to the gate drive circuit. Another is the load transient command whether you want to turn on this switch or not; that means, it is linked with this ok.

(Refer Slide Time: 09:08)



Digital VMC in a Buck Converter – Practical Implementation

So, now we want to design this particular thing and what does it look like? So, again I discuss this k actual output voltage you have a resistive divider, and this gain we are taking is around 0.27. After that scaled it is coming to the input to the ADC and now the adc_data is going out. As we have discussed this adc_data is adc_data, we are assigning this because this data is coming at a higher rate. After all, we are using a 25 megahertz clock.

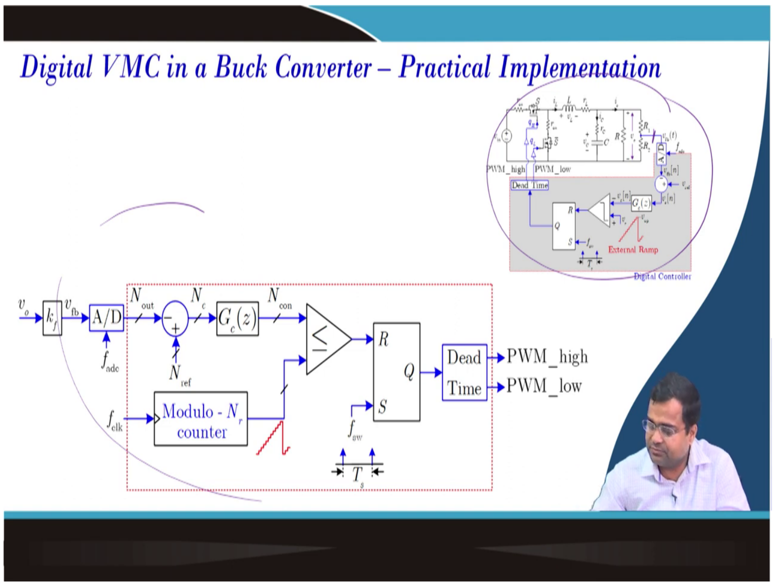
So, continuously this data is coming at the rate of 25 megahertz because the throughput of the ADC is 25 megahertz, but the latency is 6 cycle clock cycle. Well, that means, our samples are available of all this ADC if this is my f adc, but let us say we want to capture this sample somewhere at this point because this is continuous though this is my f switching frequency.

So, we are calling the data which is captured right here; that means, there will be output voltage. So, as if these samples were captured here and this sample is available here; that means, this to this we are assuming 6 cycle latency. So, this data actually will be captured here and this is I in out; that means, the inside what is the output voltage number corresponding to the output voltage is nothing but a sample voltage which was sample 6 ADC cycle earlier; that means, there is a sample first and then you switch.

Then, this is available and this is captured; that means, we need a clock to synchronize data access and that will be in and out and we will go to the Verilog code. Now, once it is captured that is the number because we have discussed the Q format and this is in the format of Q 1.9 format.

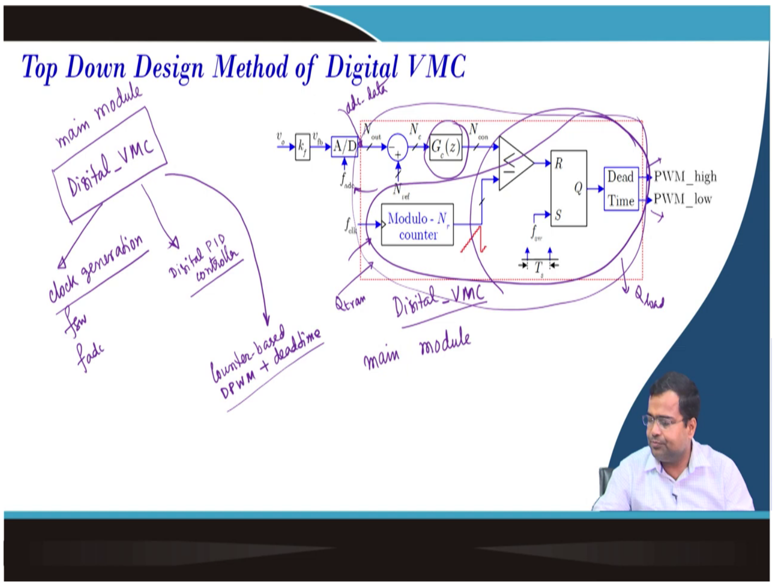
So, we need to provide the N ref accordingly, because there is a scaling factor. So, it is acting like a reference V reference, reference voltage in the digital command that is why it is N ref, and then the N error is going to the controller. So, we are using a PID, digital PID controller and that control output is going and there is a saw tooth waveform. And, then this is a trailing edge modulator and after this modulator will have a deadtime circuit to generate Q high and Q low.

(Refer Slide Time: 11:56)



So, eventually, this whole block is shown in this red line; that means, we need to design. This is a practical implementation. So, this circuit if you take from the feedback to the gate signal, will look like this ok, what we have discussed.

(Refer Slide Time: 12:08)



Now, top-down design methodology; so, we want to design, this we call it as a digital underscore voltage mode control because we want to make a Verilog module. We have already discussed Verilog in detail. So, you want to create a module which is the main module is the main module and that name is digital VMC. The main module will interface with the external world and the first is the adc_data.

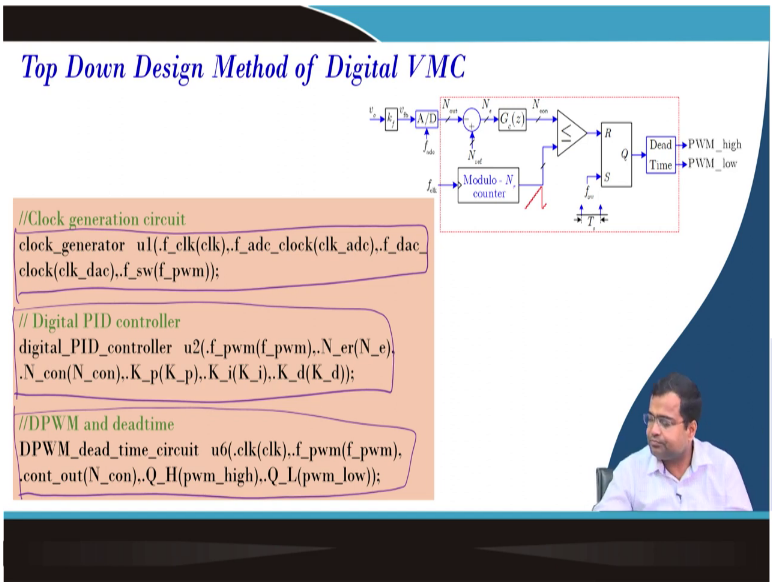
Second, it is taking going out the clock signal, the clock coming in, the PWM going out and there will be one more which will be Q load and there will be one more input which will be Q transient type. So, this interface is taking place inside this controller. So, the external wall it is dealing with all digital numbers either gate signals or adc_data or clock, but it is only dealing with 0 and 1. It can be scalar data or vector data.

So, we need to design this main module, we need to subdivide it into multiple parts. So, the top down, our digital voltage mode control which will interface with the external world; this is my main module. So, as if I am designing a digital IC, you can imagine because FPGA is emulating like an IC that interfaces with the world with digital data from ADC, sending gate signals, and so on.

But, in this module, if you go inside there will be multiple blocks. The one block will be clock generation because we need to generate a switching frequency clock, we need to generate an ADC clock, and here we do not need a DAC clock. So, this clock generation will generate this clock.

The second one requires a digital PID controller because this controller is nothing but ok. So, this will generate PID, then the other one will be digital DPWM; DPWM. So, it is an I would say counter-based DPWM. So, it is a counter-based DPWM plus deadtime. So, these are included because we are talking about this entire block, this includes all this entire block. So, there are multiple such sub-modules, which will make the full overall module ok.

(Refer Slide Time: 15:21)



v_o
k_f
v_{fb}
A/D
f_{adc}
N_{out}
N_e
N_{ref}
$G_c(z)$
N_{con}
f_{clk}
Modulo - N_r counter
R
S
Q
f_{sw}
T_s
Dead Time
PWM_high
PWM_low

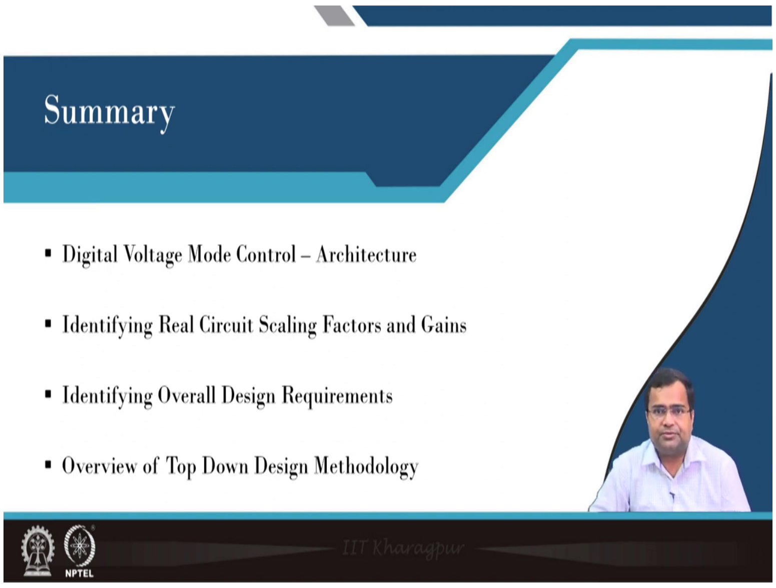
Top Down Design Method of Digital VMC

```
//Clock generation circuit
clock_generator  u1(.f_clk(clk),.f_adc_clock(clk_adc),.f_dac_
clock(clk_dac),.f_sw(f_pwm));

// Digital PID controller
digital_PID_controller  u2(.f_pwm(f_pwm),.N_er(N_e),
.N_con(N_con),.K_p(K_p),.K_i(K_i),.K_d(K_d));

//DPWM and deadtime
DPWM_dead_time_circuit  u6(.clk(clk),.f_pwm(f_pwm),
.cont_out(N_con),.Q_H(pwm_high),.Q_L(pwm_low));
```

So, I am going to discuss more detail Verilog coding in the subsequent lecture. But, in summary, we have one module which you know that in the main module, we can instantiate another module. So, I have not shown the main module yet, I will come to that part in the subsequent lecture.

But, for the main module, we are instantiating another module which is a clock generator, as I just told you. We are instantiating another is the digital PID controller and we are instantiating another which is a DPWM plus deadtime. So, these are the three sub-modules and on top of that, there is a main module.

(Refer Slide Time: 16:02)



Summary

- Digital Voltage Mode Control – Architecture
- Identifying Real Circuit Scaling Factors and Gains
- Identifying Overall Design Requirements
- Overview of Top Down Design Methodology

NPTEL
IIT Kharagpur

So, in summary, we have discussed digital voltage mode control architecture. We have we try to identify what are the real scaling factors and gains. Then, we talked about overall design requirements and just an overview of the top-down design methodology, that is it for today.

Thank you very much.