

EE155/255 Green Electronics

Periodic Steady State
Embedded Software
9/27/17

Prof. William Dally
Computer Systems Laboratory
Stanford University



Logistics

- Lab group assignments
- Homework 1 due next Monday 10/2
- Sign up for Lab signoff time
 - Lab 1 must be signed off next week
- Sign up for Piazza
 - piazza.com/stanford/fall2017/ee155ee255
- Discussion sections
 - Fridays 4:30-5:50PM
 - STLC 118

Course at a Glance

No	Date	Topic	HW out	HW in	Lab out	Lab ck	Lab	HW
1	9/25/17	Intro (basic converters)	1		1		Intro to ST32F3	Periodic Steady State
2	9/27/17	Embedded Prog/Power Elect.						
3	10/2/17	Power Electronics - 1 (switches)	2	1	2	1	AC Energy Meter	Power Devices
4	10/4/17	Power Electronics - 2 (circuits)						
5	10/9/17	Photovoltaics	3	2	3	2	PV MPPT	Motor control Matlab
6	10/11/17	Feedback Control						
7	10/16/17	Electric Motors	4	3	4	3	Motor control - Lab/	Feedback
8	10/18/17	Isolated Converters						
9	10/23/17	Solar Day	5/PP	4	5	4	PS	Isolated Converters
10	10/25/17	Magnetics						
11	10/30/17	Soft Switching	6	5/PP	6	5	Magnetics	Magnetics and Inverters
12	11/1/17	Project Discussions						
13	11/6/17	Inverters, Grid, PF, and Batteries		6	P	6	Project	
14	11/8/17	Thermal & EMI						
15	11/13/17	Quiz Review				C1		
16	11/15/17	Grounding, and Debugging						
Q	11/15/17	Quiz - in the evening						
	11/20/17	Thanksgiving Break				C2		
	11/22/17	Thanksgiving Break						
17	11/27/17	Wrapup						
18	11/29/17	Guest Lecture				C3		
19	12/4/17	Guest Lecture						
20	12/6/17	No Class						
	TBD	Project presentations			P			
	12/15/17	Project webpage due						

RECENT

EV Battery Management Impacts Vehicle Design and the Grid

SEP 01, 2017

Advanced System Measures Battery Impedance While in Use

AUG 17, 2017

DEFEAT HEAT
Reliable Thermal Management Solutions



CUI INC **Digi-Key ELECTRONICS** [LEARN MORE »](#)

Book Review: The Circuit Designer's Companion

AUG 17, 2017

Researchers Look To Materials To Improve Micro-Grid Renewable Energy



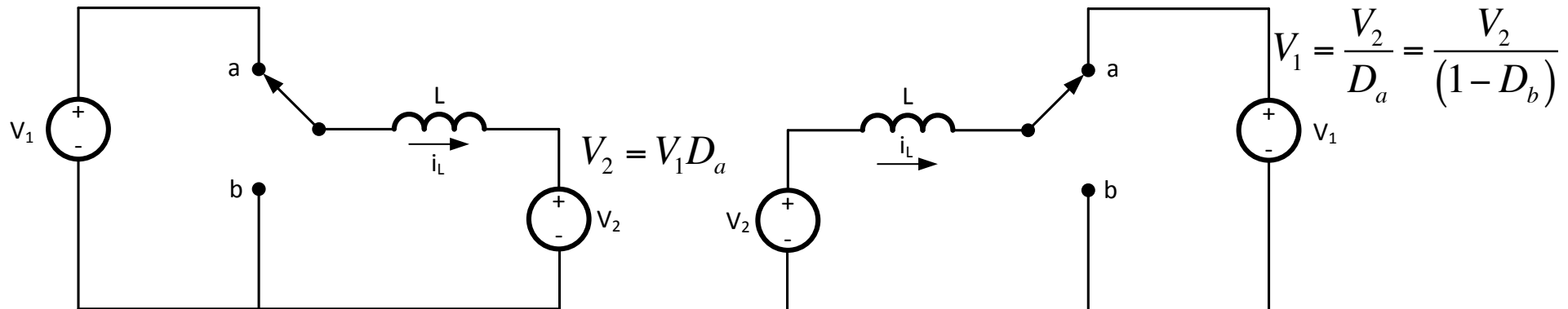
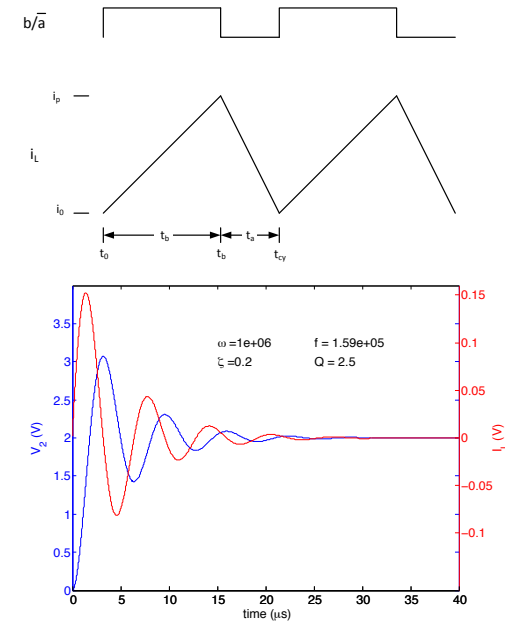
MARKETS > AUTOMOTIVE

EV Battery Management Impacts Vehicle Design and the Grid

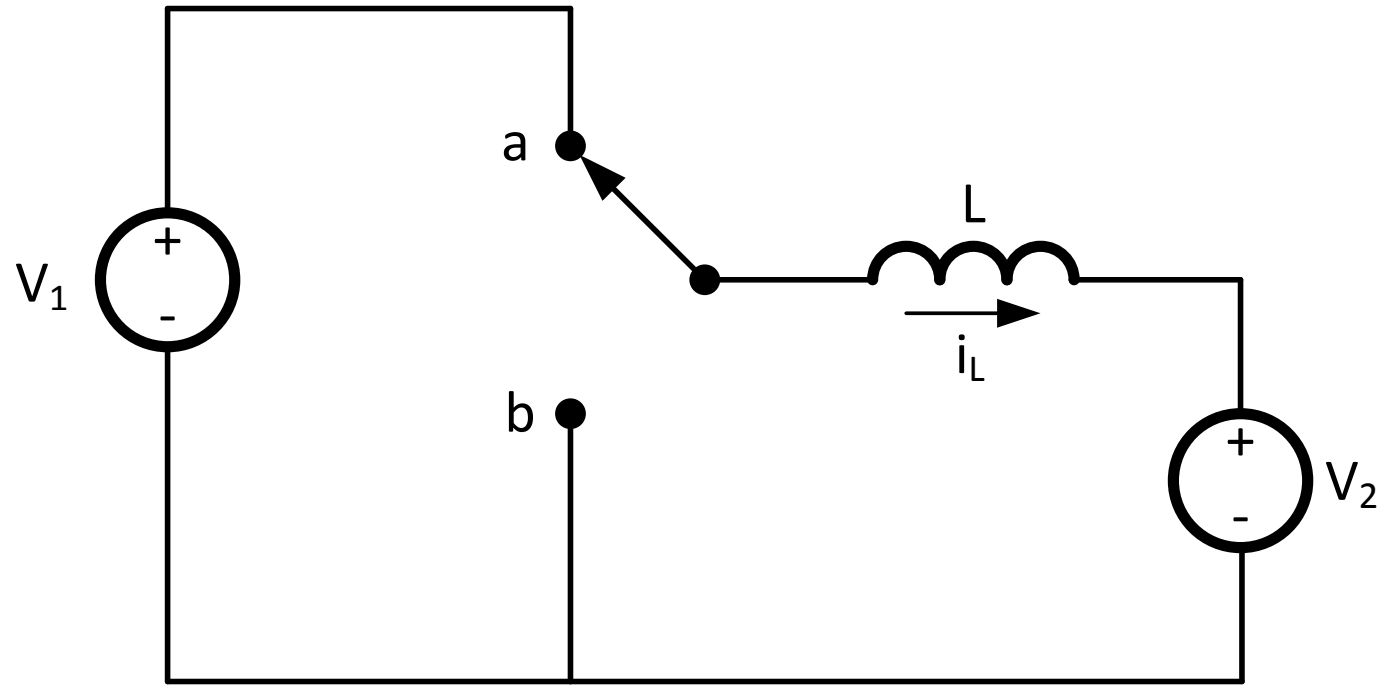
Refueling a vehicle with an internal combustion engine is very simple: You just drive in to a gas station and fill up your tank. You would hope that “refueling” an EV battery would be just as easy. That’s not exactly the case.

PSSA in One Slide

- Green energy systems are made from voltage converters
 - PV systems, electric/hybrid cars, wind power, etc...
 - Power path + intelligent control
- Separate behavior into fast and slow
 - Fast – within switching cycle
 - Slow – $f \ll f_{cy}$
- Within cycle solve for **periodic steady state**
 - State variables the same at start and end of cycle.
- For slow transient response
 - Write difference equations
- Buck/Boost Topologies



Buck Converter



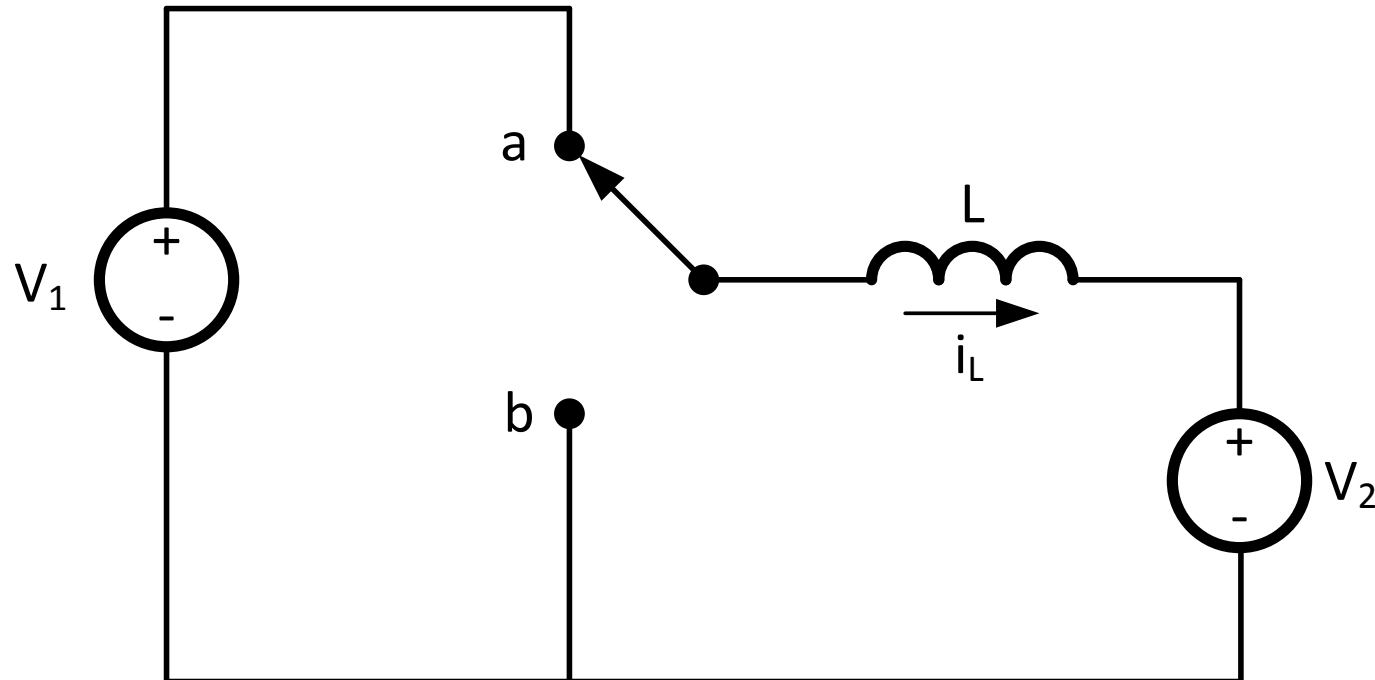
Periodic Steady State

Switch is periodic with cycle time t_{cy}

Separate behavior

Over one cycle of switch

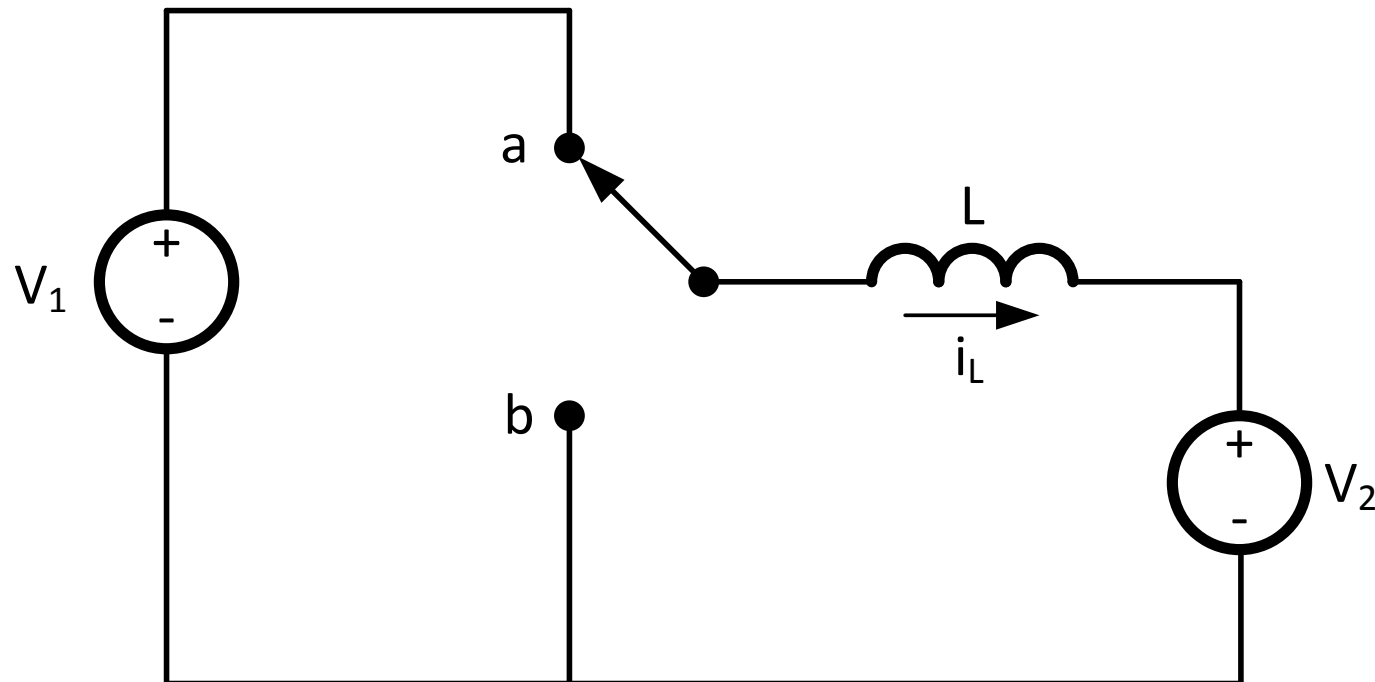
At frequencies much lower than f_{cy}



Periodic Steady State

Time is divided into cycles with period t_{cy}

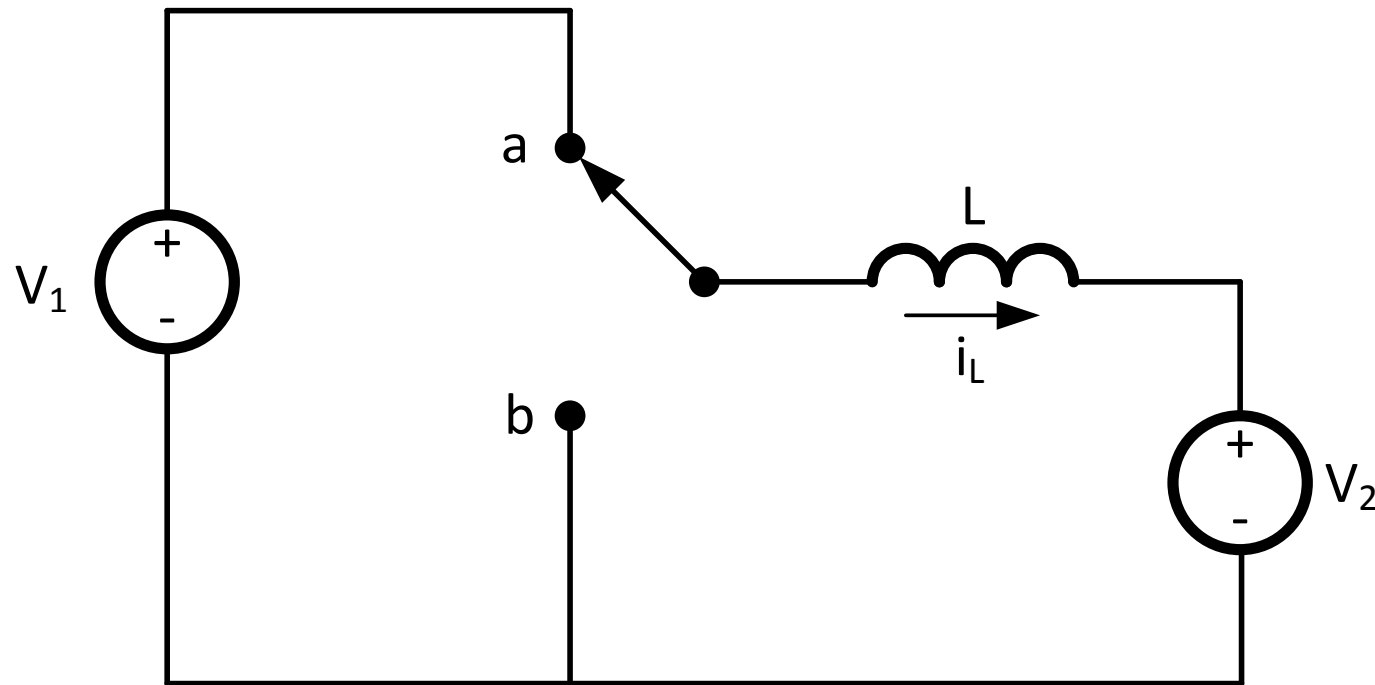
State variables are the same at the beginning and end of each cycle



Periodic Steady State

Each cycle:

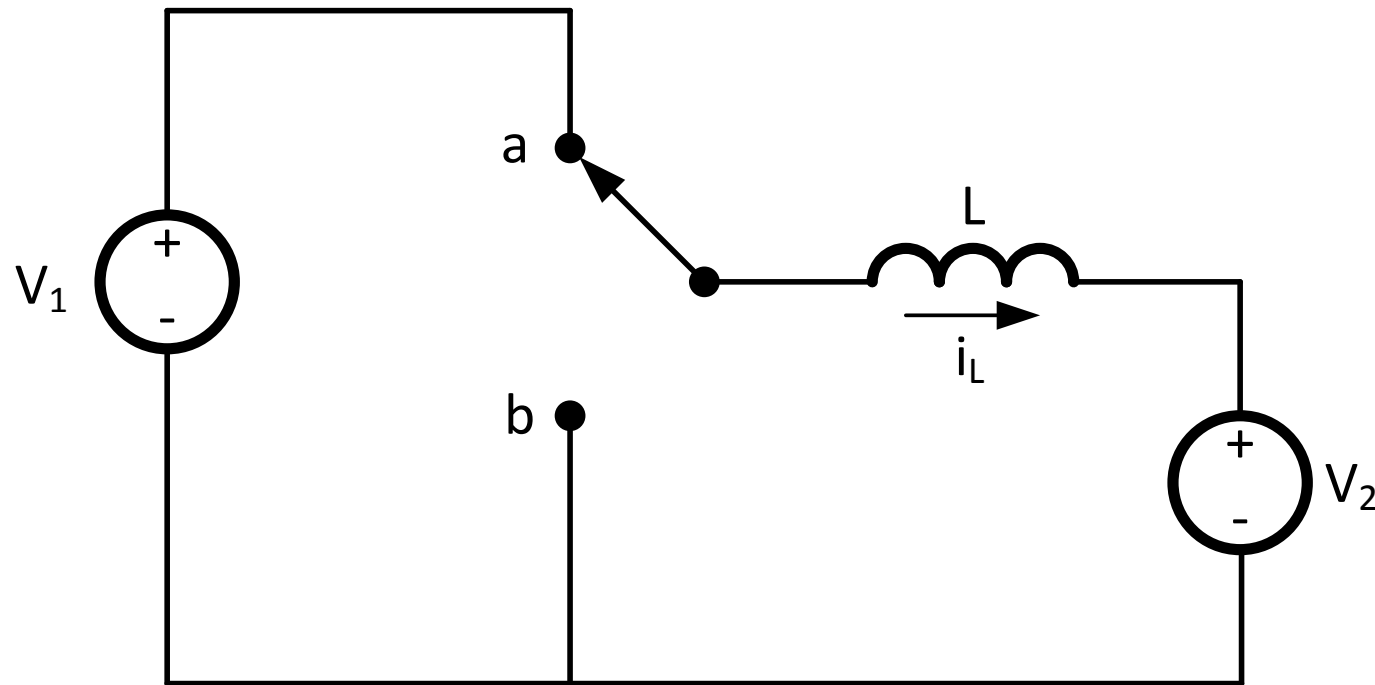
switch in position **a** for t_a , **b** for $t_b = t_{cy} - t_a$



Position a

$$V_L = V_1 - V_2$$

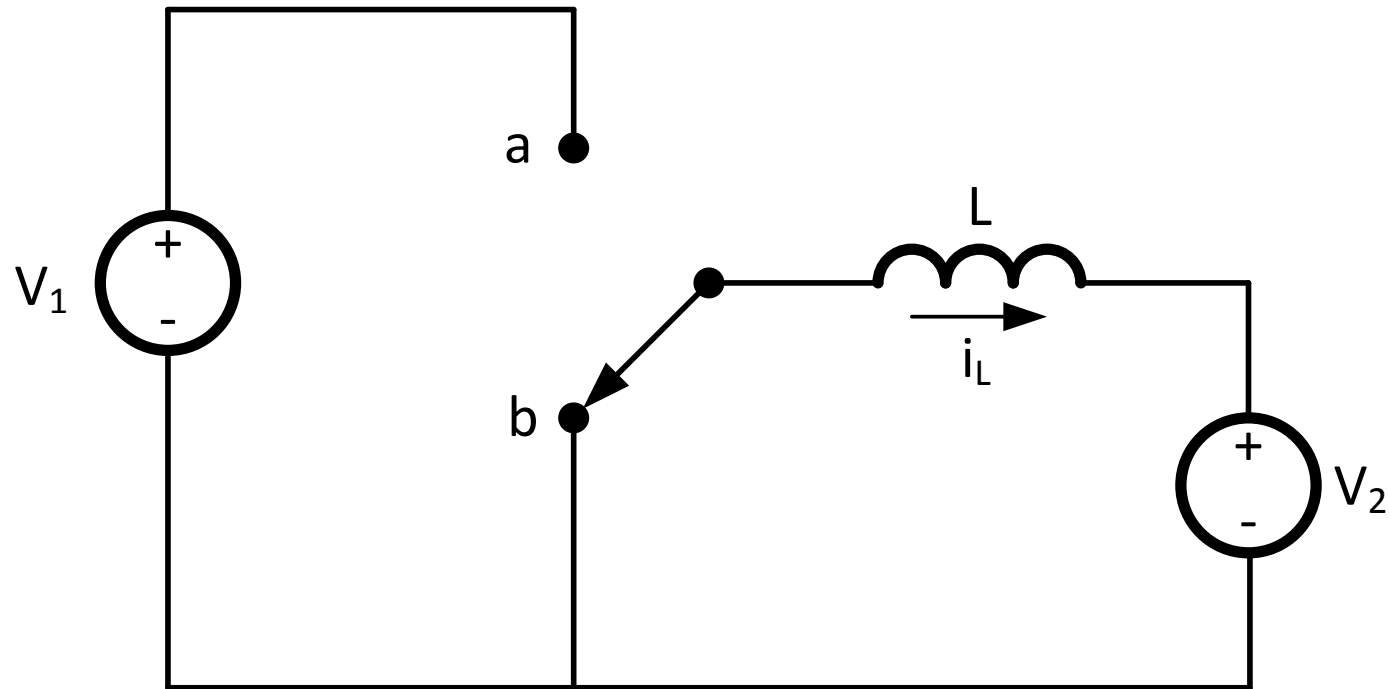
$$\Delta I_a = \frac{t_a V_L}{L} = \frac{t_a (V_1 - V_2)}{L}$$



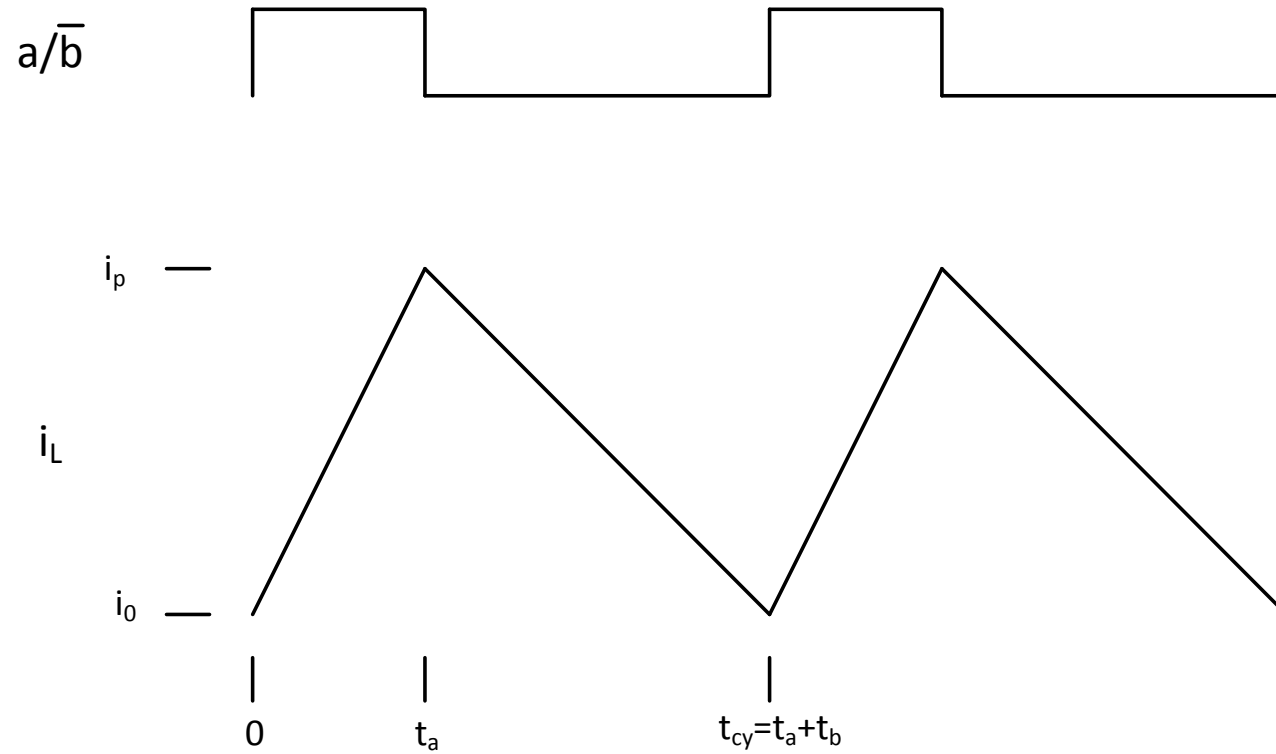
Position b

$$V_L = -V_2$$

$$\Delta I_b = \frac{-t_b V_2}{L}$$

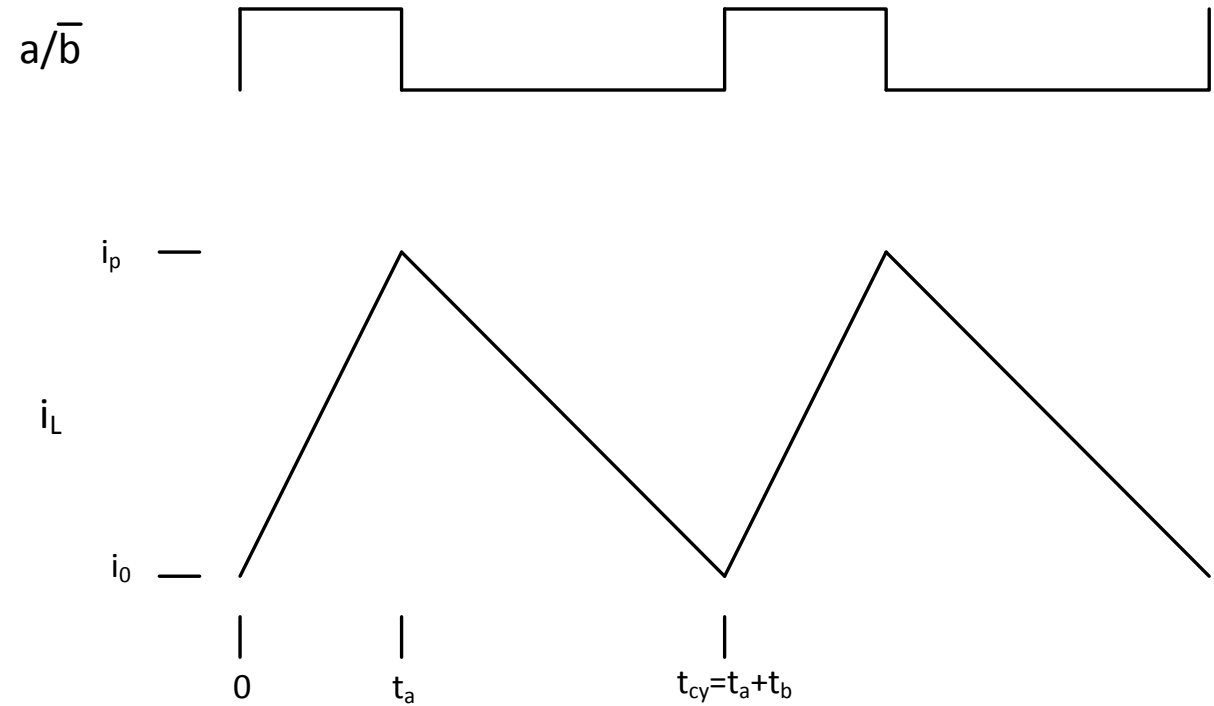
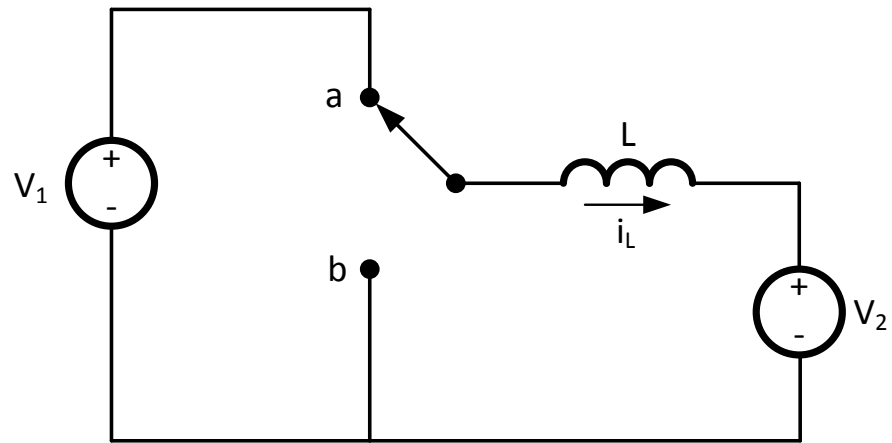


Waveforms



Peak Current

$$i_p = t_a(V_1 - V_2)/L + i_0$$



Periodic Steady State

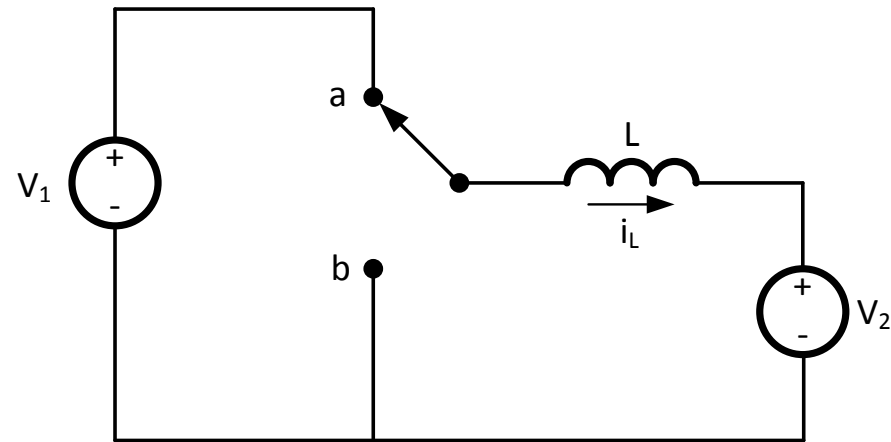
$$\Delta I_a + \Delta I_b = 0$$

$$\frac{t_a (V_1 - V_2)}{L} + \frac{t_b (-V_2)}{L} = 0$$

$$t_a (V_1 - V_2) + t_b (-V_2) = 0$$

$$t_a V_1 = (t_a + t_b) V_2$$

$$V_2 = \left(\frac{t_a}{t_a + t_b} \right) V_1 = DV_1$$

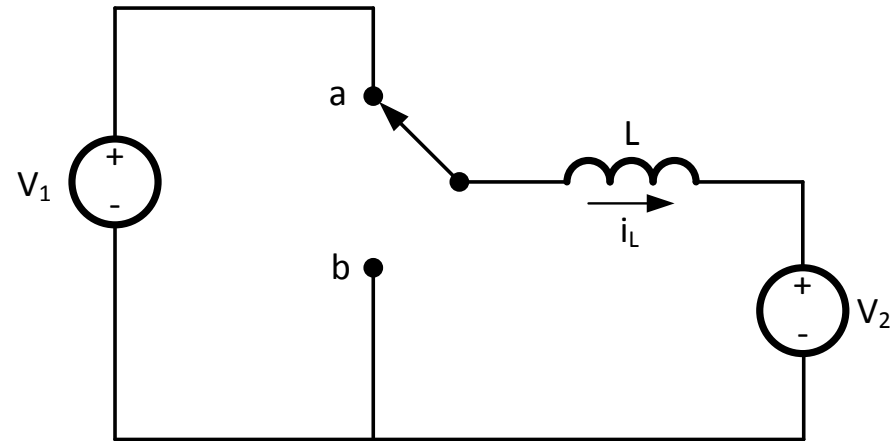


Periodic Steady State

- $V_2 = DV_1$
- Only depends on duty factor D
 - Not on cycle time t_{cy}
 - Not on inductor value L
 - They determine the “ripple”

Simpler Approach

- Look at net volt-seconds on each side of the inductor
- Must be equal in periodic steady state
- DV_1 on left side
- V_2 on right side
- $DV_1 = V_2$

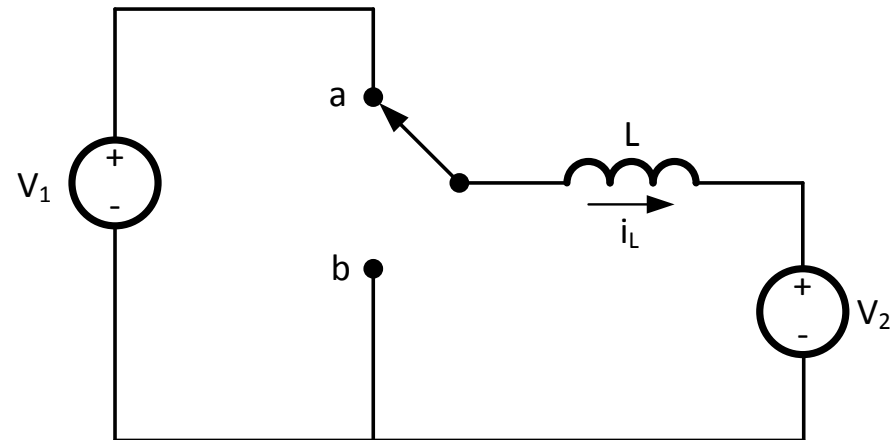


What if $V_2 \neq DV_1$?

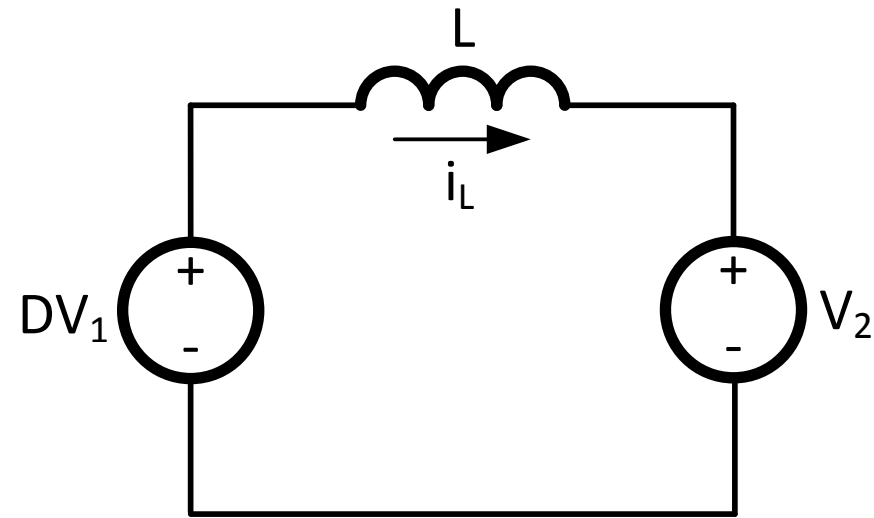
$$\Delta I = \Delta I_a + \Delta I_b$$

$$\Delta I = \frac{t_a (V_1 - V_2)}{L} + \frac{t_b (-V_2)}{L}$$

$$\Delta I = \frac{t_{cy}}{L} (DV_1 - V_2)$$



Equivalent Circuit



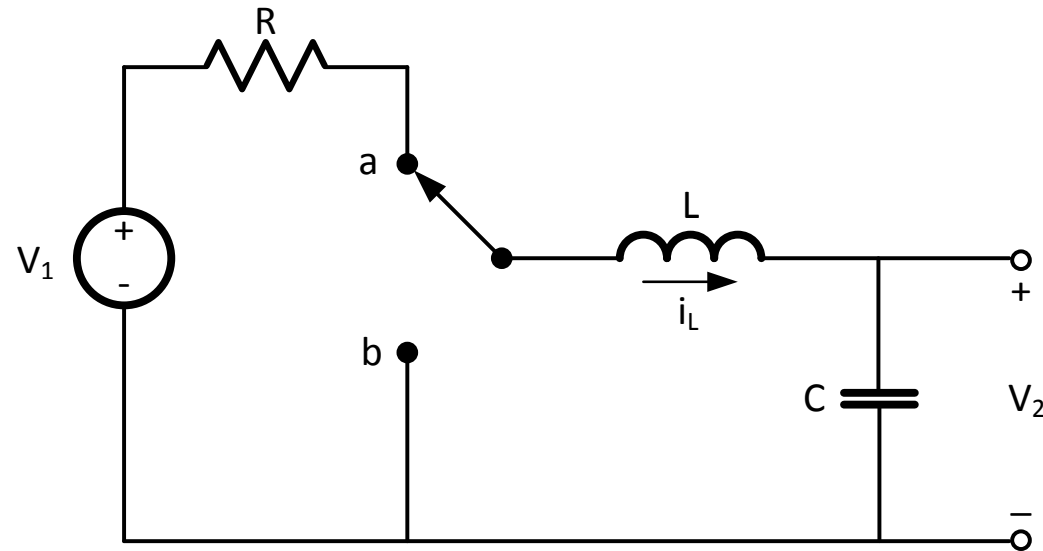
Transient Response

$$V_2(0) = 0, I_L(0) = 0 .$$

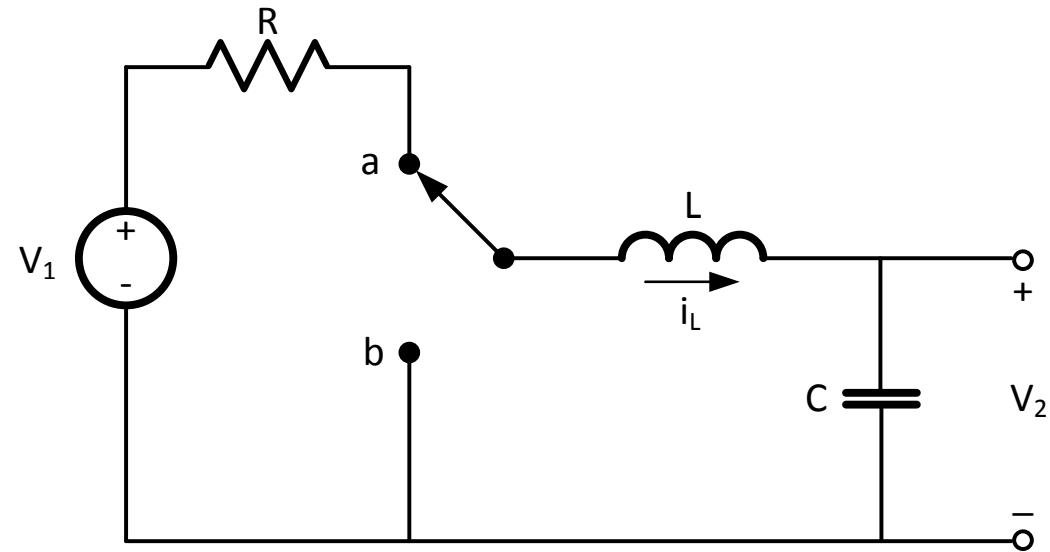
$$L = 10\mu\text{H}, C = 0.1\mu\text{F}, R = 8\Omega, D = 0.5$$

What is $V_2(t)$? If $V_1(t > 0)$ is 4V.

Switching frequency is high enough you can ignore it.



Transient Response



$$\Delta I = \Delta I_a + \Delta I_b$$

$$\Delta I = \frac{t_a (V_1 - IR - V_2)}{L} + \frac{t_b (-V_2)}{L}$$

$$\Delta I = \frac{t_{cy}}{L} (DV_1 - DIR - V_2)$$

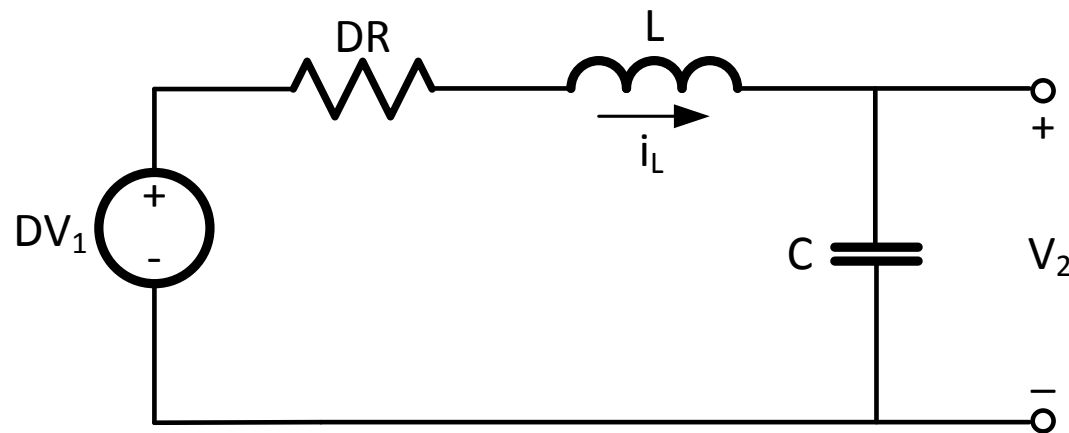
Equivalent Circuit

$$\omega = \frac{1}{\sqrt{LC}} = 1\text{MHz}$$

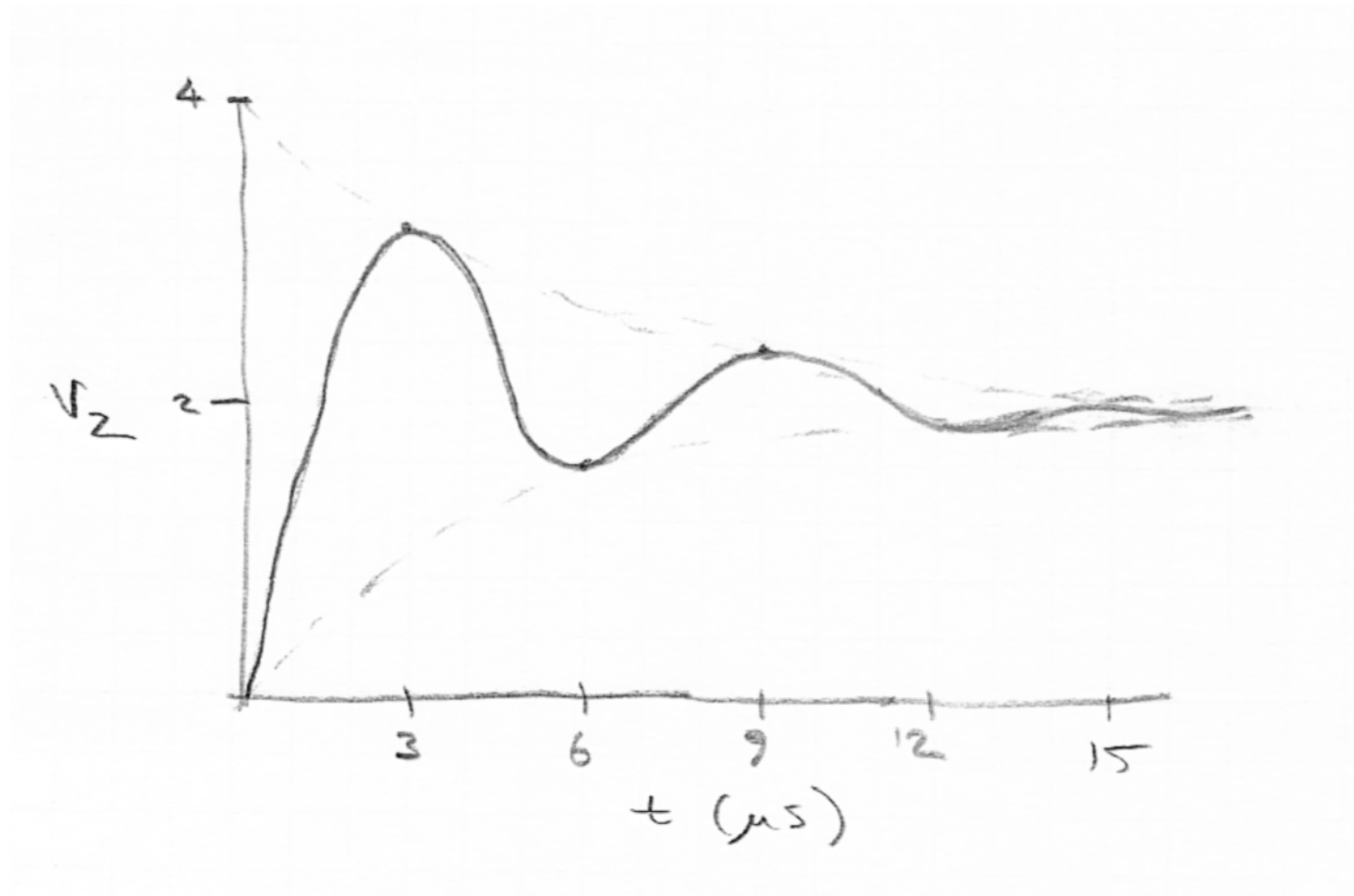
$$f = \frac{\omega}{2\pi} = 159\text{kHz}$$

$$\zeta = \frac{DR}{2} \sqrt{\frac{C}{L}} = 0.2$$

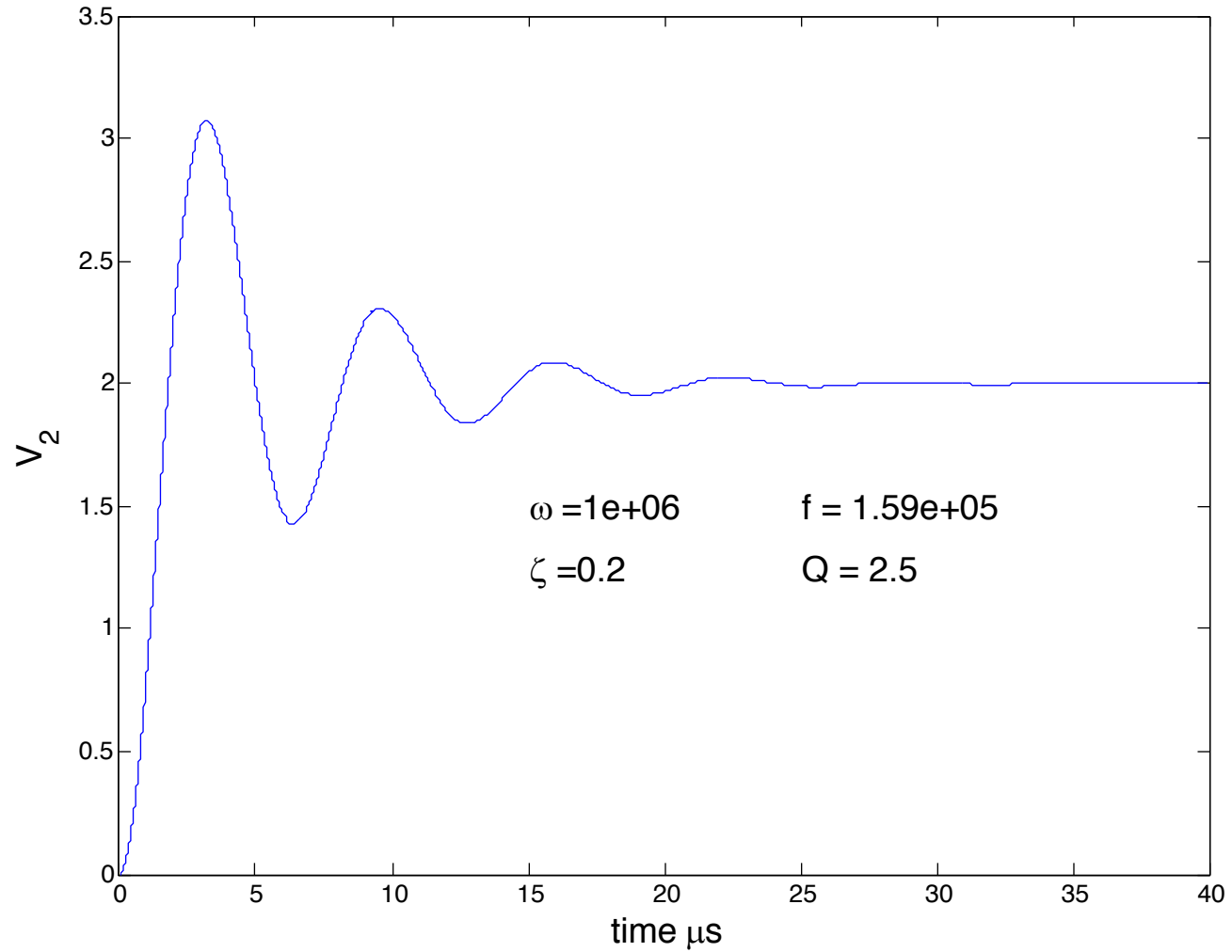
$$Q = \frac{1}{2\zeta} = 2.5$$



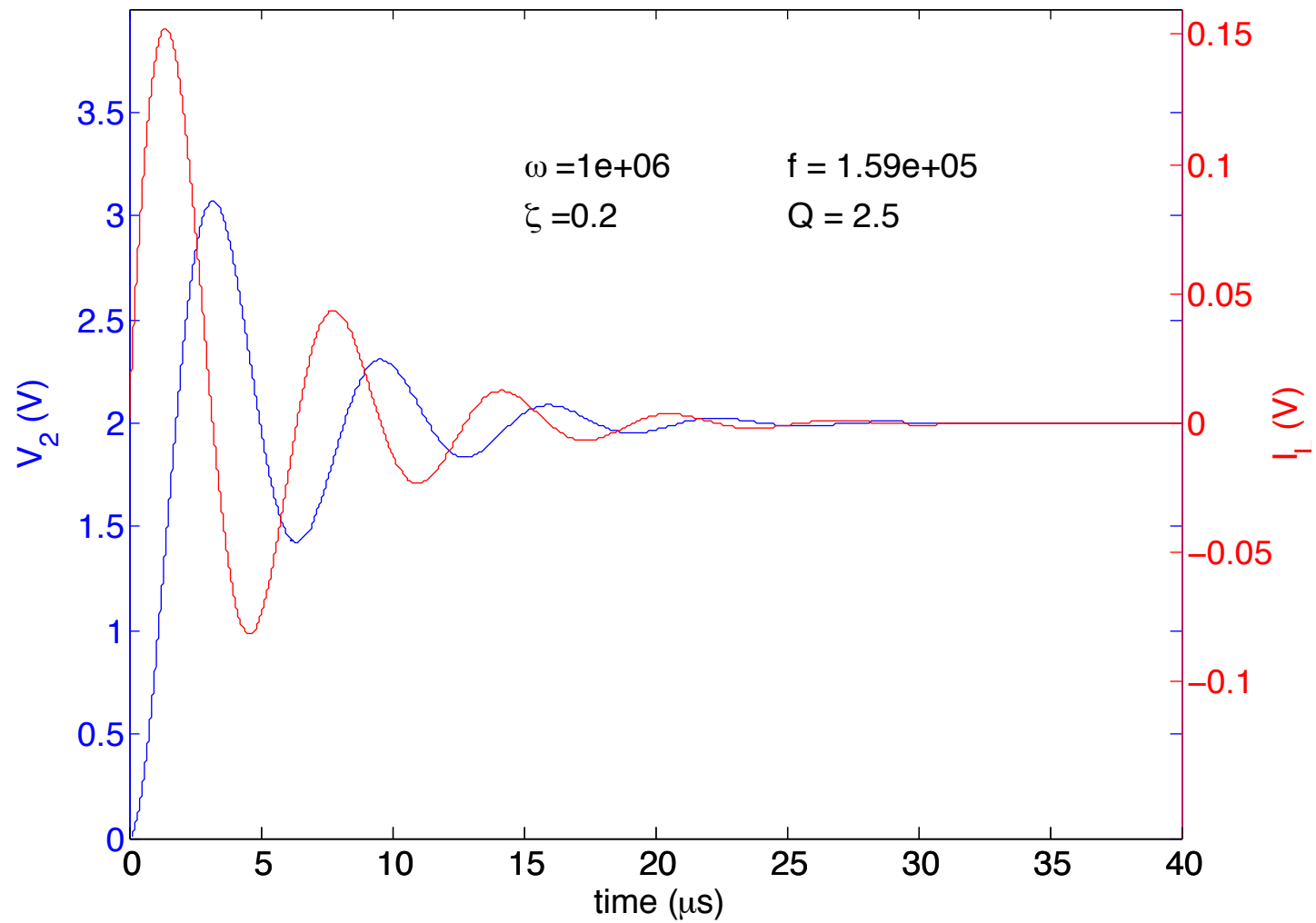
Sketch Response from f and Q



Verify With Simulation



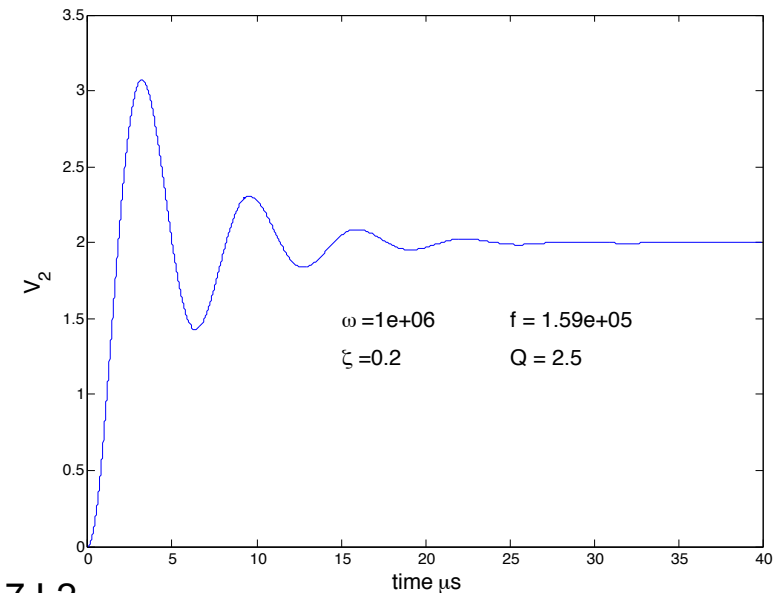
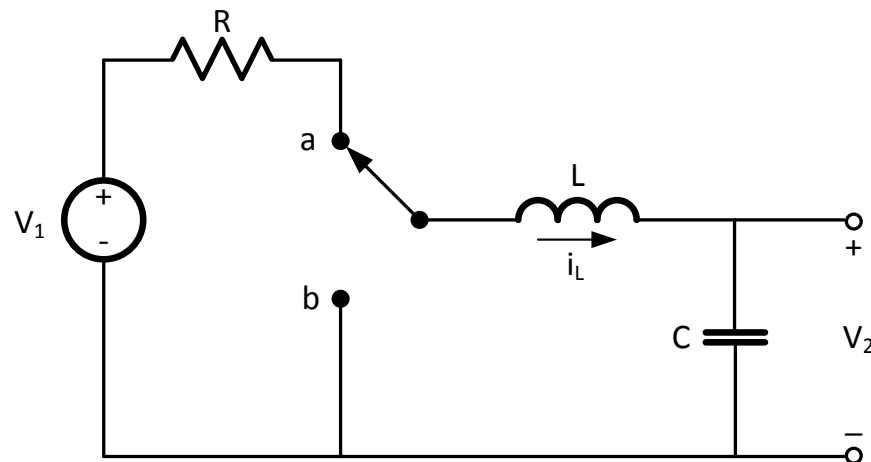
With Current



50% Overshoot

Is this OK?

What if your 5V power supply went to 7.5V on turn-on?



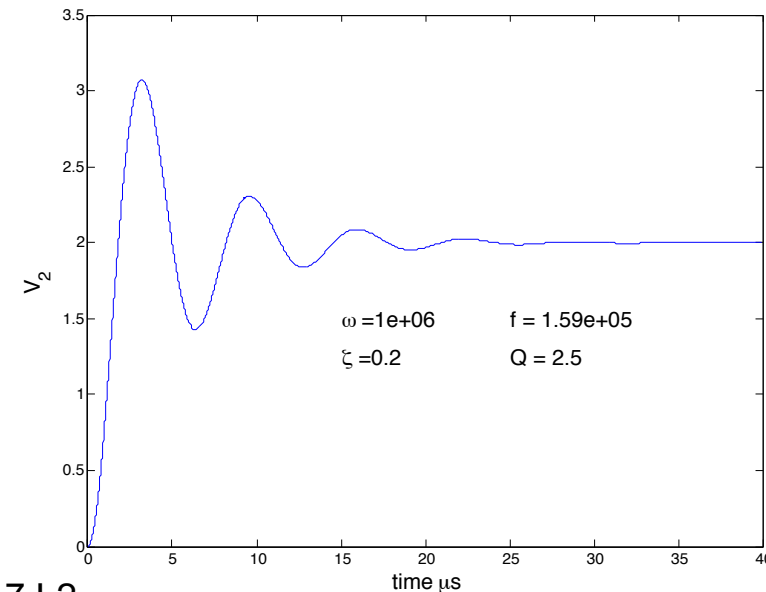
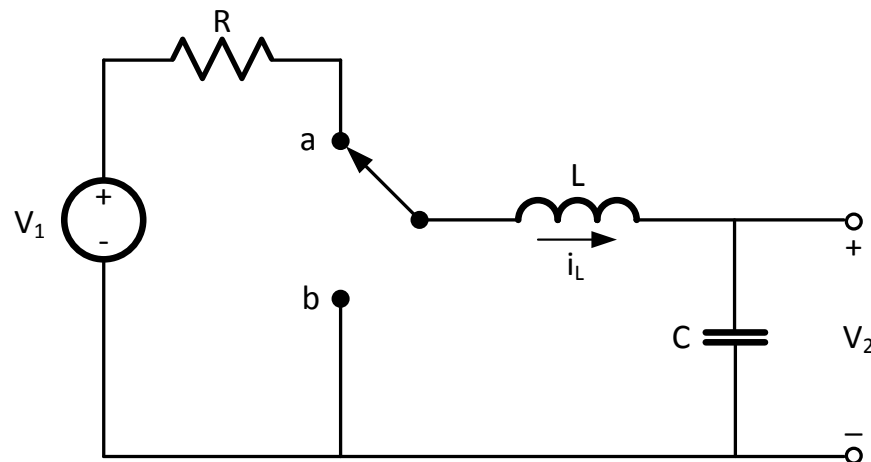
50% Overshoot

Is this OK?

What if your 5V power supply went to 7.5V on turn-on?

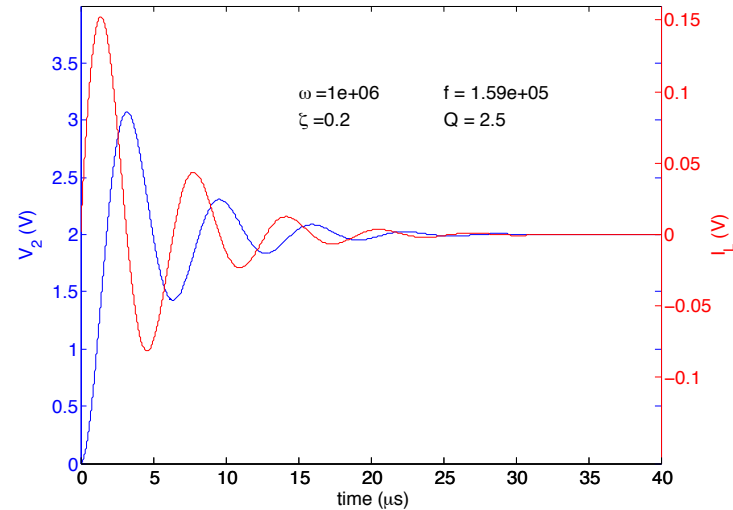
It would fry all of your logic chips

Control of switch-mode circuits prevents overshoot



Matlab Model

```
L=1e-5 ;  
C=1e-7 ;  
R=4 ; % DR  
VI=2 ;  
  
dt=1e-8 ;  
tsim=4e-5 ;  
steps = ceil(tsim/dt) ;  
  
t=0 ;  
vc=0 ;  
il=0 ;
```



```
% preallocate result arrays to speed simulation  
tx=zeros(1,steps) ; vx=zeros(1,steps) ; ix = zeros(1, steps) ;  
  
for i=1:steps  
    vl = VI-vc-il*R ;  
    vc = vc+il*dt/C ;  
    il = il+vl*dt/L ;  
    tx(i) = t ; vx(i) = vc ; ix(i) = il ; % log variables  
    t=t+dt ;  
end
```

Matlab Model

```
L=1e-5 ;
C=1e-7 ;
R=4 ; % DR
VI=2 ;

dt=1e-8 ;
tsim=4e-5 ;
steps = ceil(tsim/dt) ;

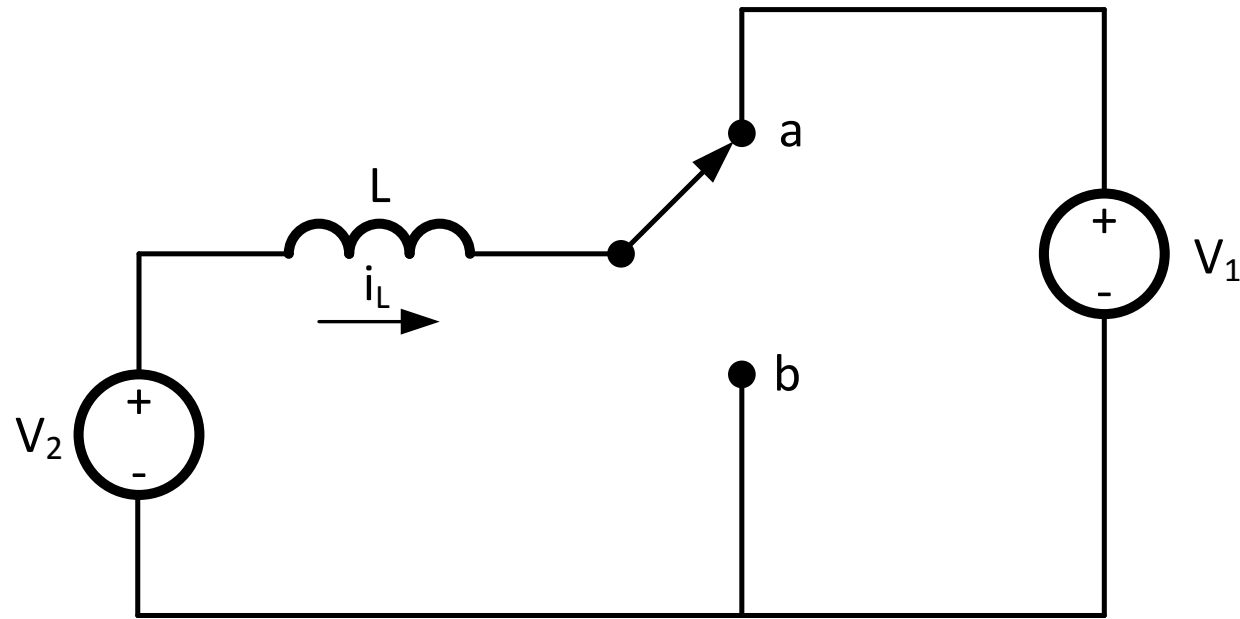
t=0 ;
vc=0 ;
il=0 ;

% preallocate result arrays to speed simulation
tx=zeros(1,steps) ; vx=zeros(1,steps) ; ix = zeros(1,
steps) ;

for i=1:steps
    vl = VI-vc-il*R ;
    vc = vc+il*dt/C ;
    il = il+vl*dt/L ;
    tx(i) = t ; vx(i) = vc ; ix(i) = il ;
    t=t+dt ;
end
```

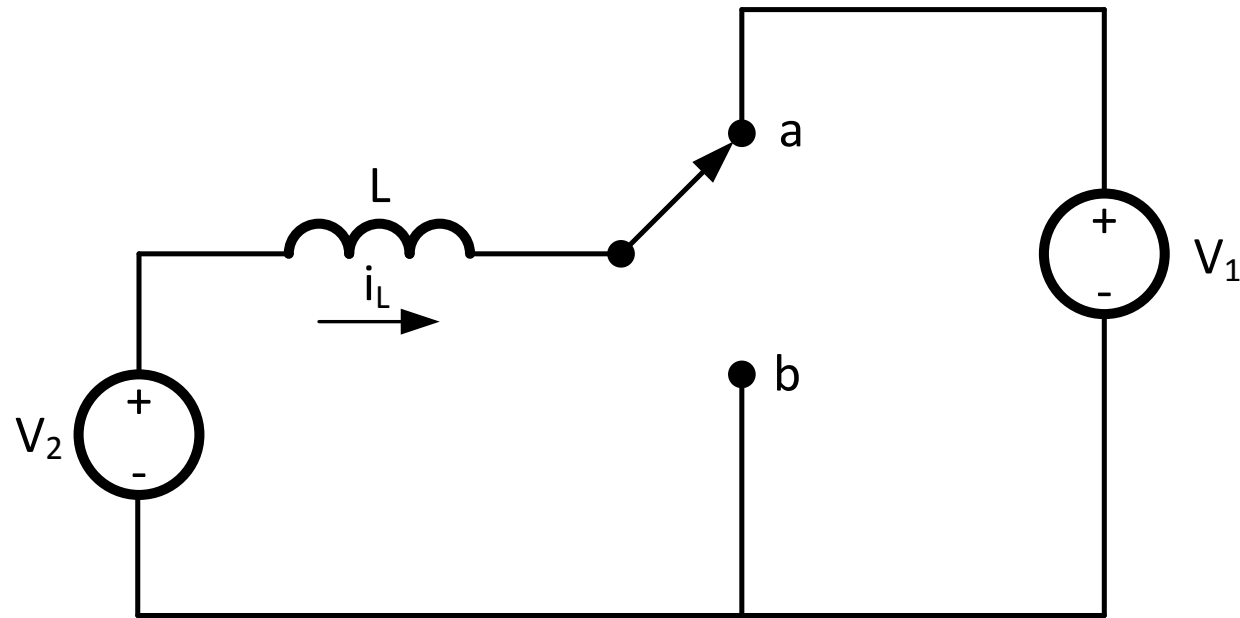
Caution!
**Simulation is not a substitute for
understanding**

Can the Buck Run Backwards?

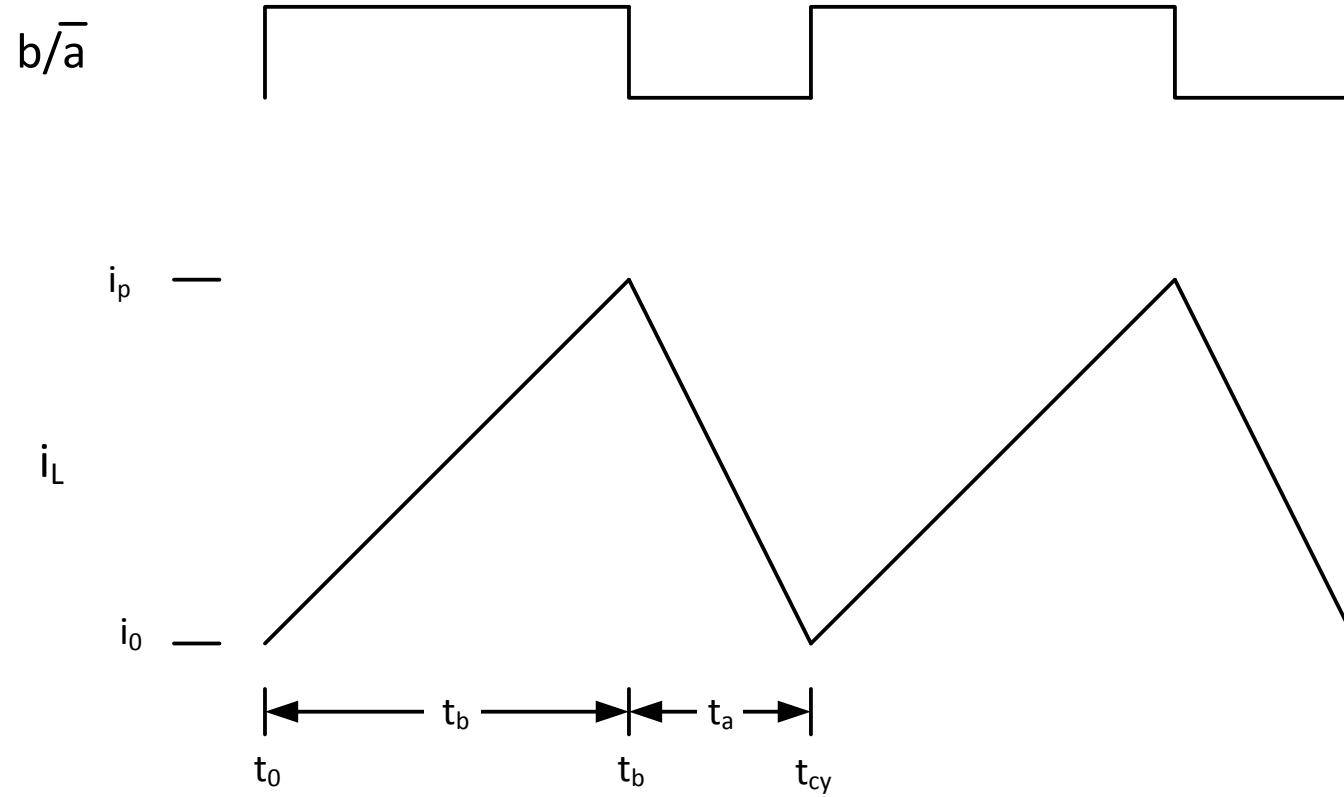


Boost Converter

$$V_1 = \frac{V_2}{D_a} = \frac{V_2}{(1 - D_b)}$$

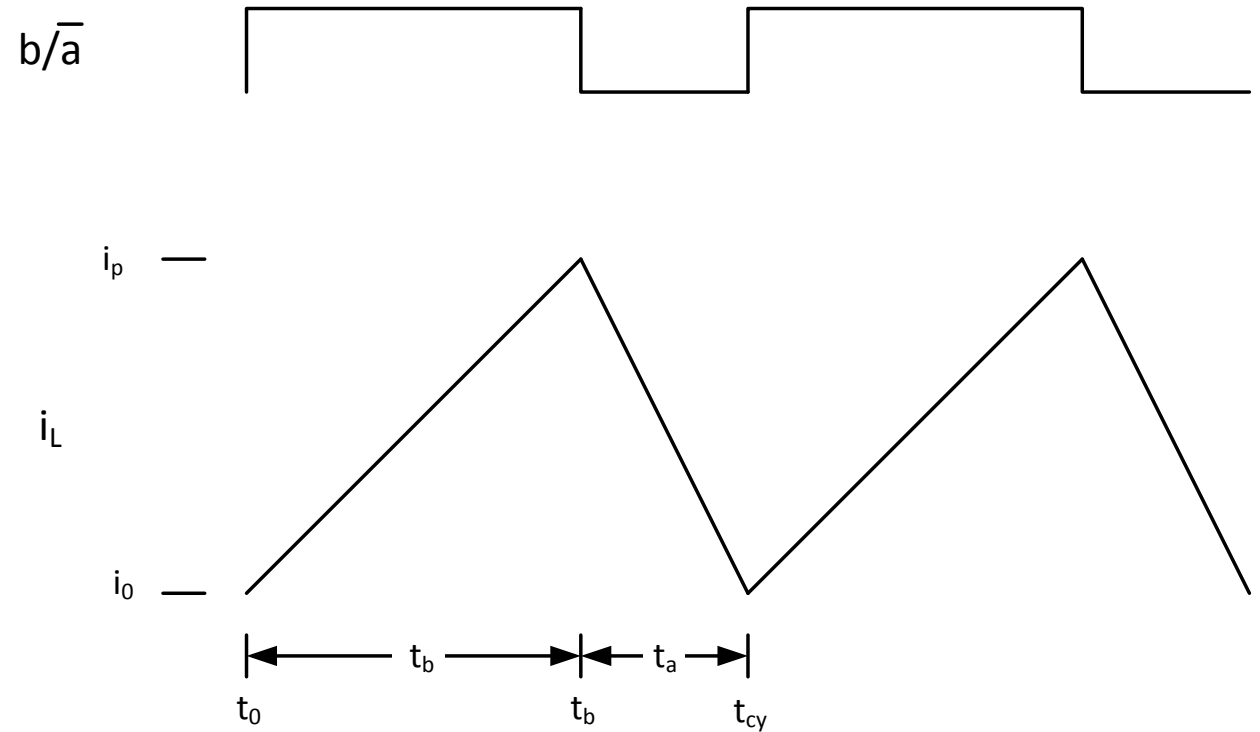
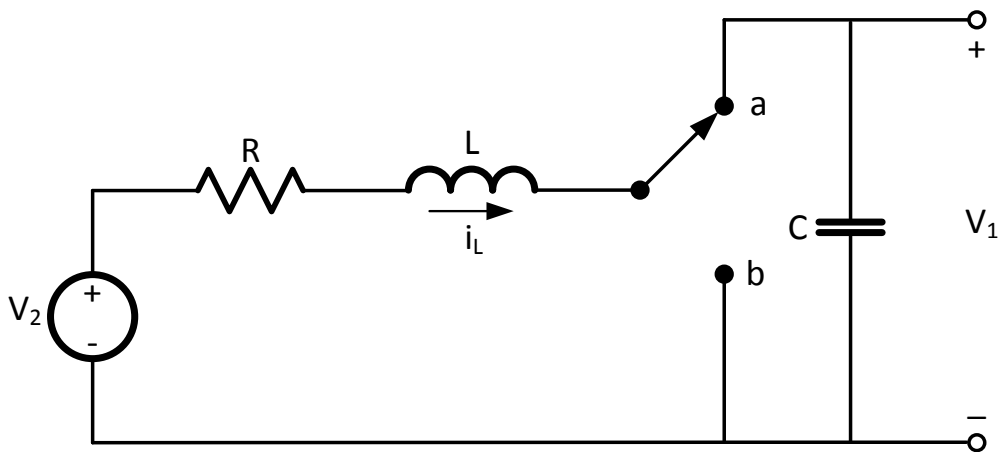


Boost Waveforms



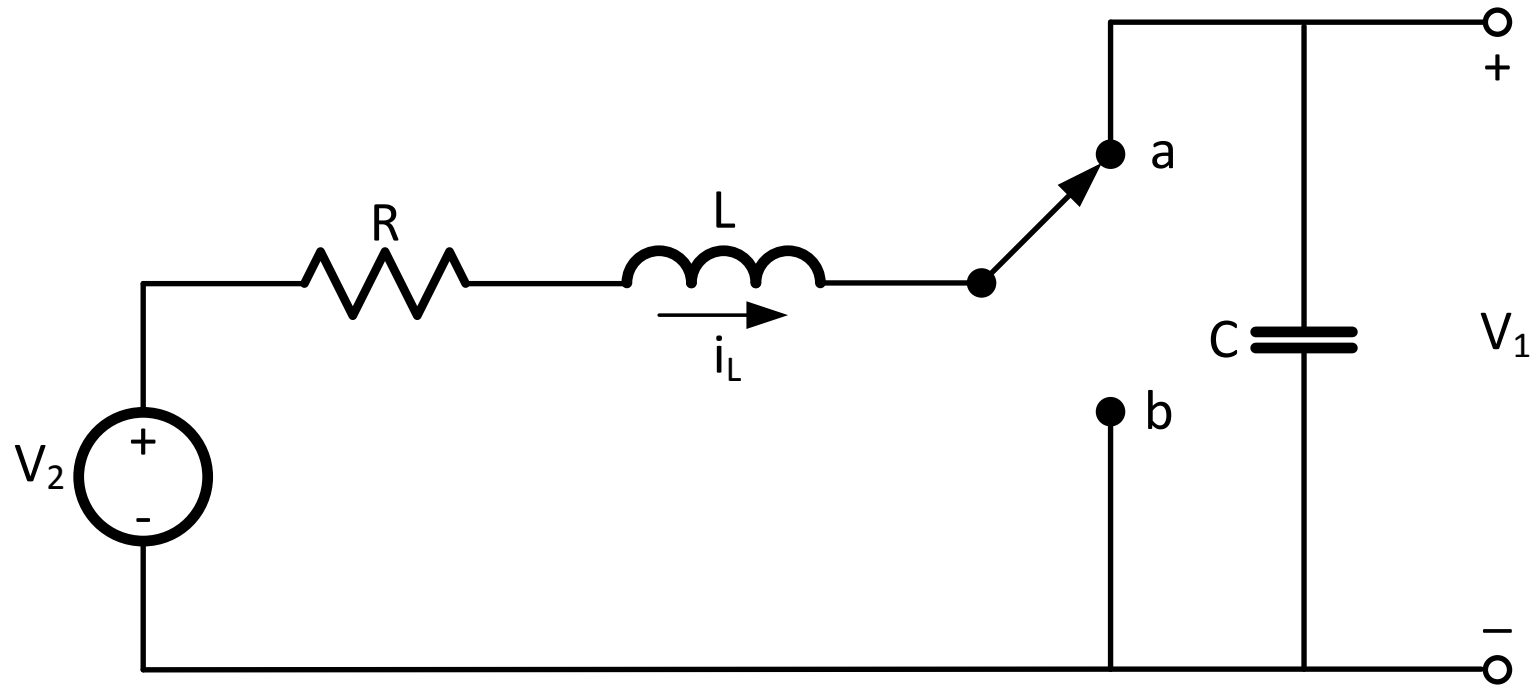
Peak Current

$$i_p = V_2 t_b / L + i_0$$



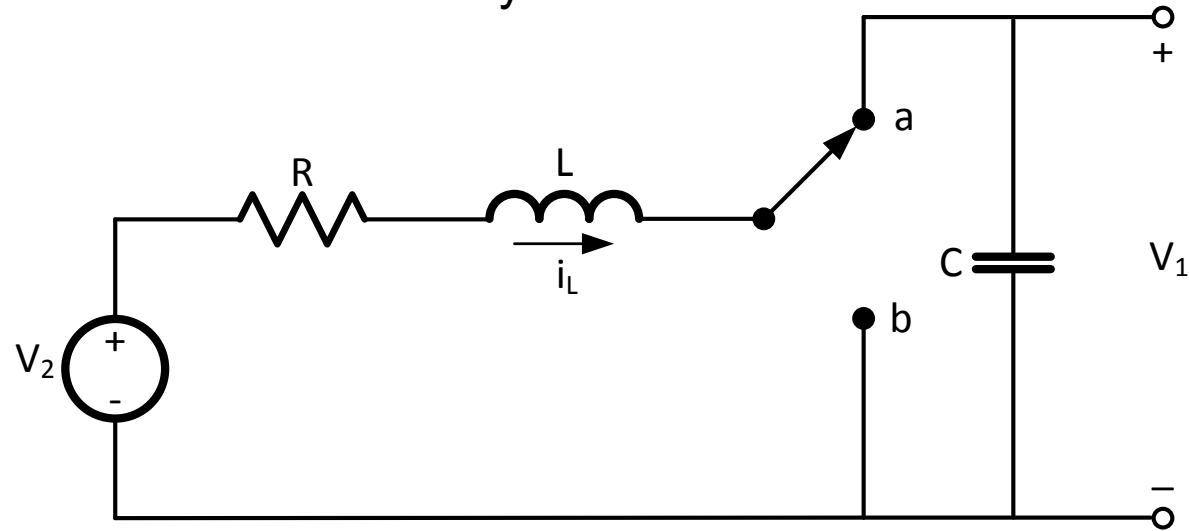
Boost Transient Response

(for $f \ll f_{cy}$)



Boost Transient Response

(for $f \ll f_{cy}$)



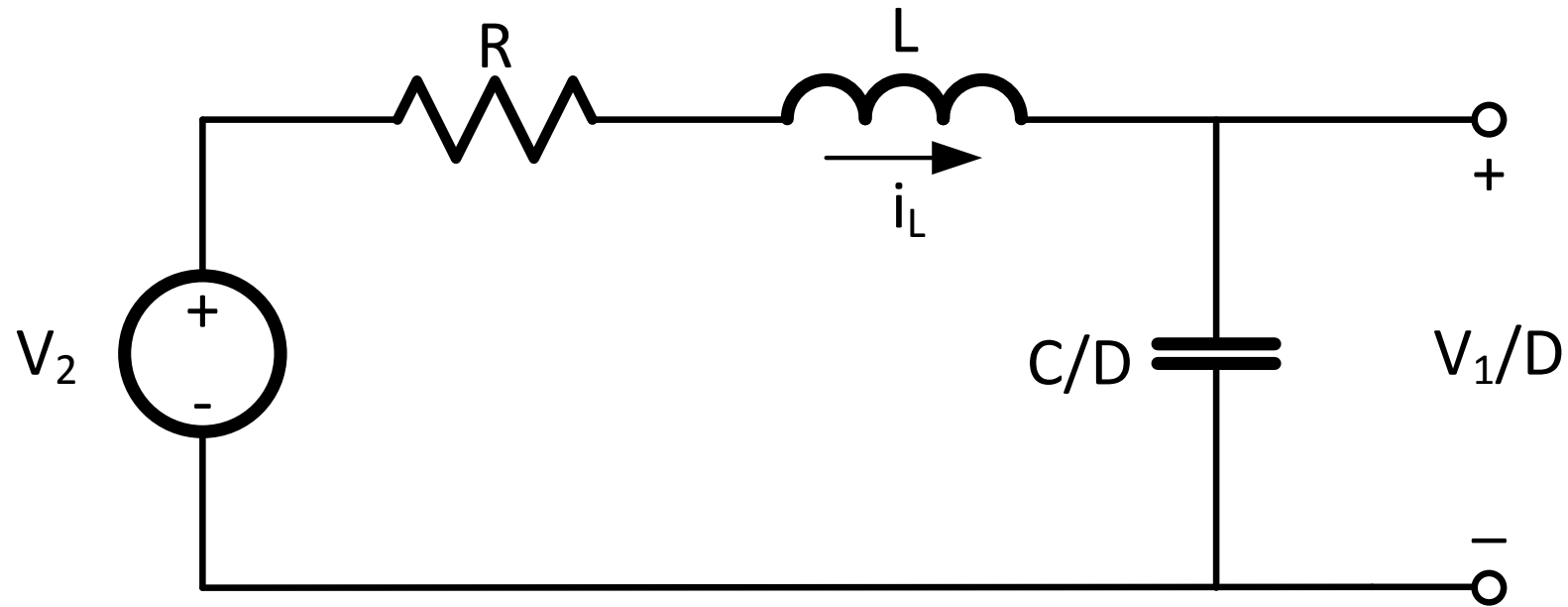
$$\Delta I = \Delta I_a + \Delta I_b$$

$$\Delta I = \frac{t_a (V_2 - IR - V_1)}{L} + \frac{t_b (V_2 - IR)}{L}$$

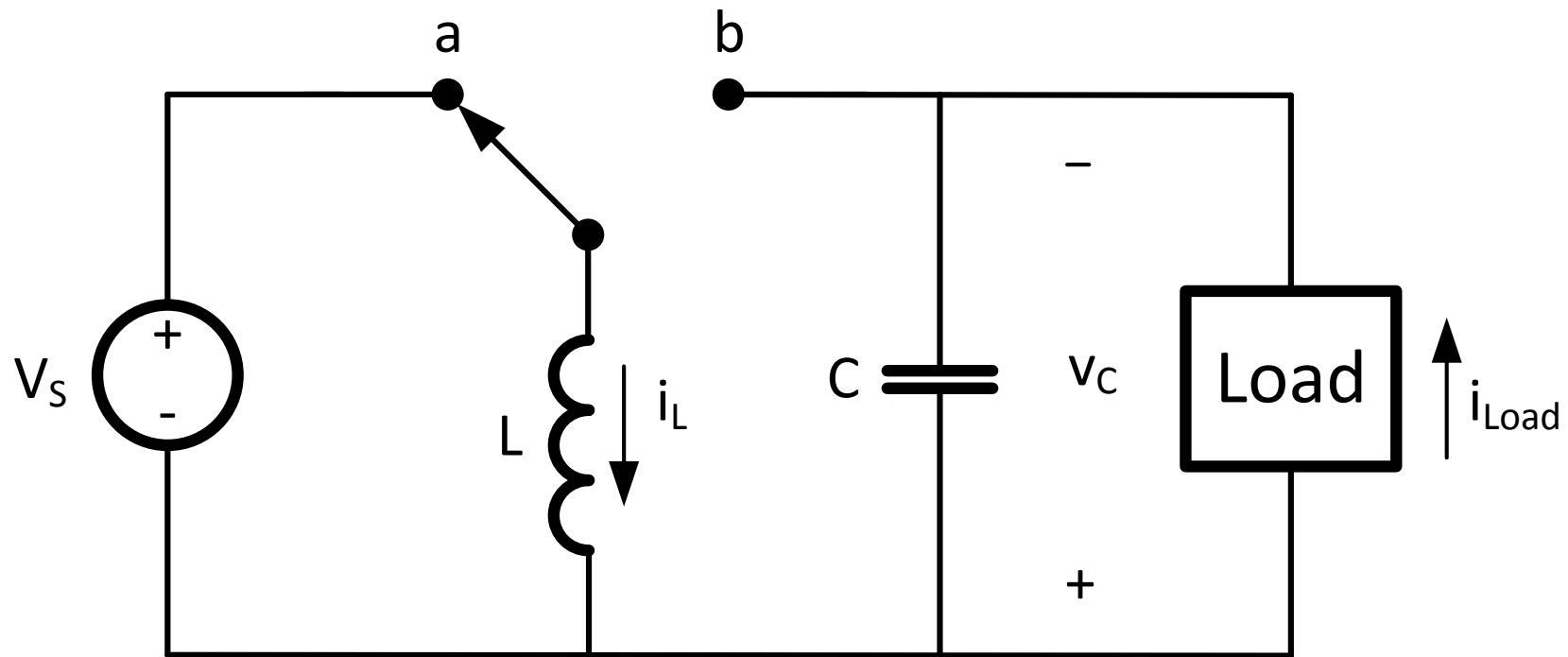
$$\Delta I = \frac{t_{cy}}{L} (V_2 - IR - D_a V_1)$$

$$\Delta V = \frac{t_a I}{C} = \frac{t_{cy} D I}{C}$$

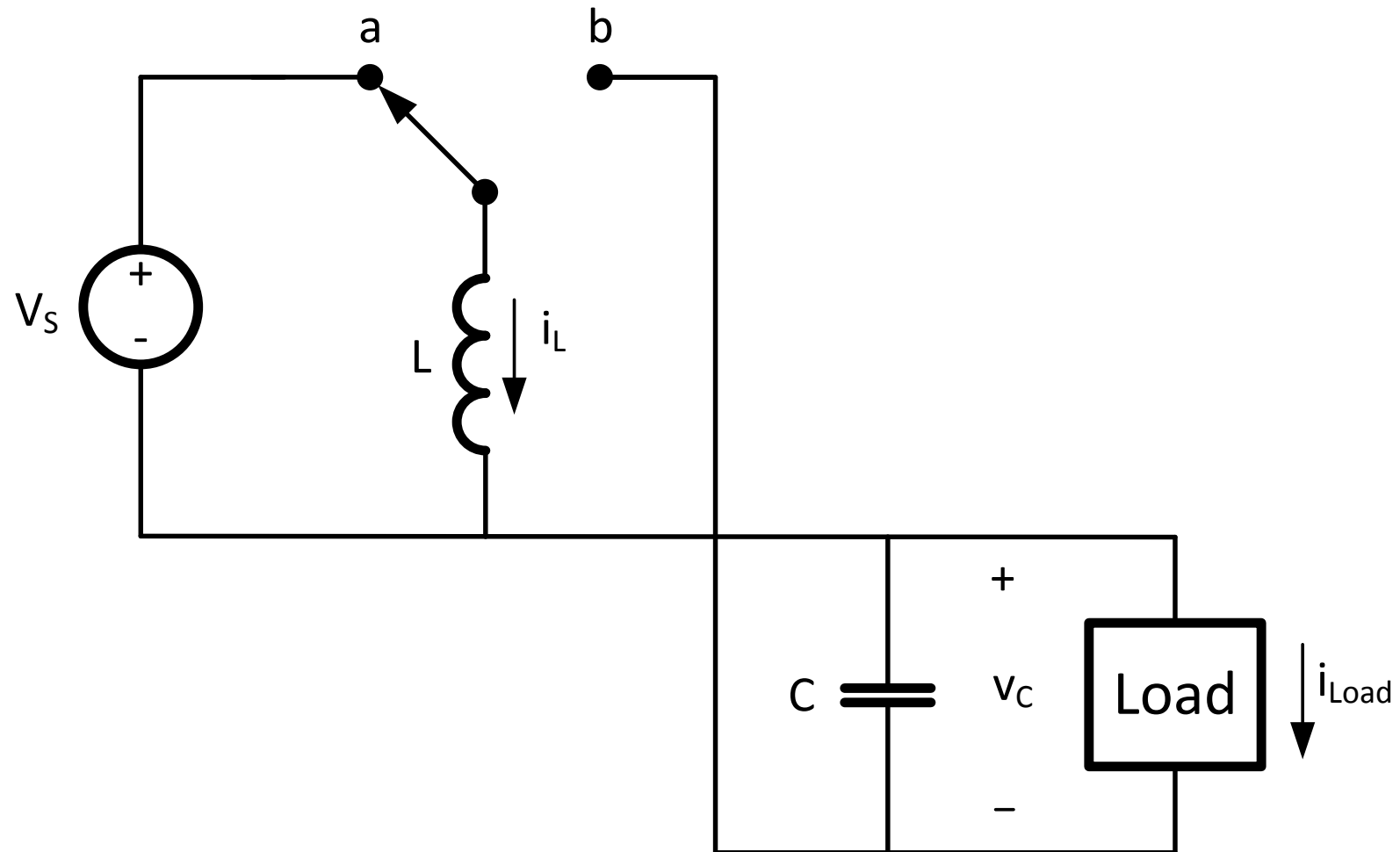
Equivalent Circuit



How Does This Work?

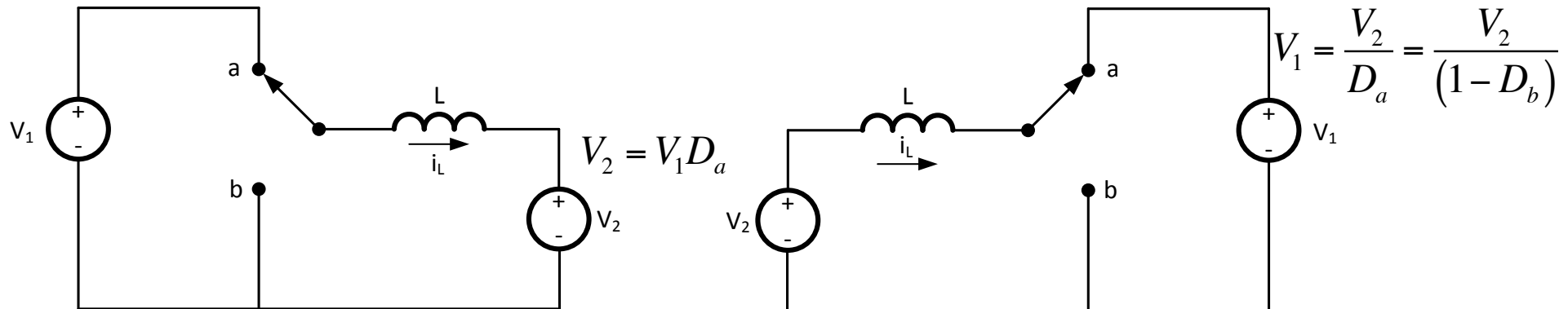
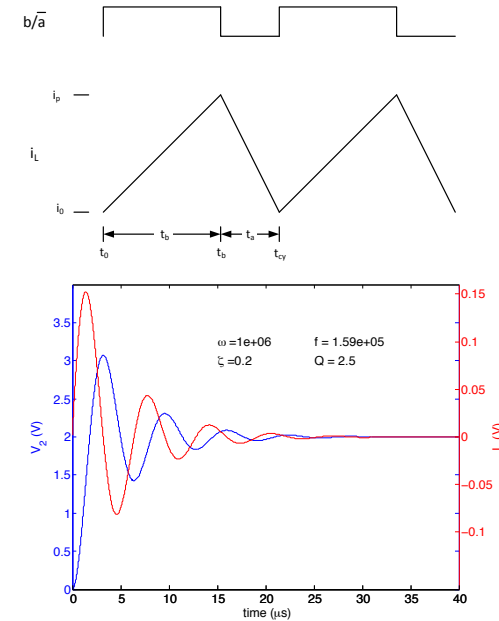


Another View



PSSA in One Slide

- Green energy systems are made from voltage converters
 - PV systems, electric/hybrid cars, wind power, etc...
 - Power path + intelligent control
- Separate behavior into fast and slow
 - Fast – within switching cycle
 - Slow – $f \ll f_{cy}$
- Within cycle solve for **periodic steady state**
 - State variables the same at start and end of cycle.
- For slow transient response
 - Write difference equations
- Buck/Boost Topologies

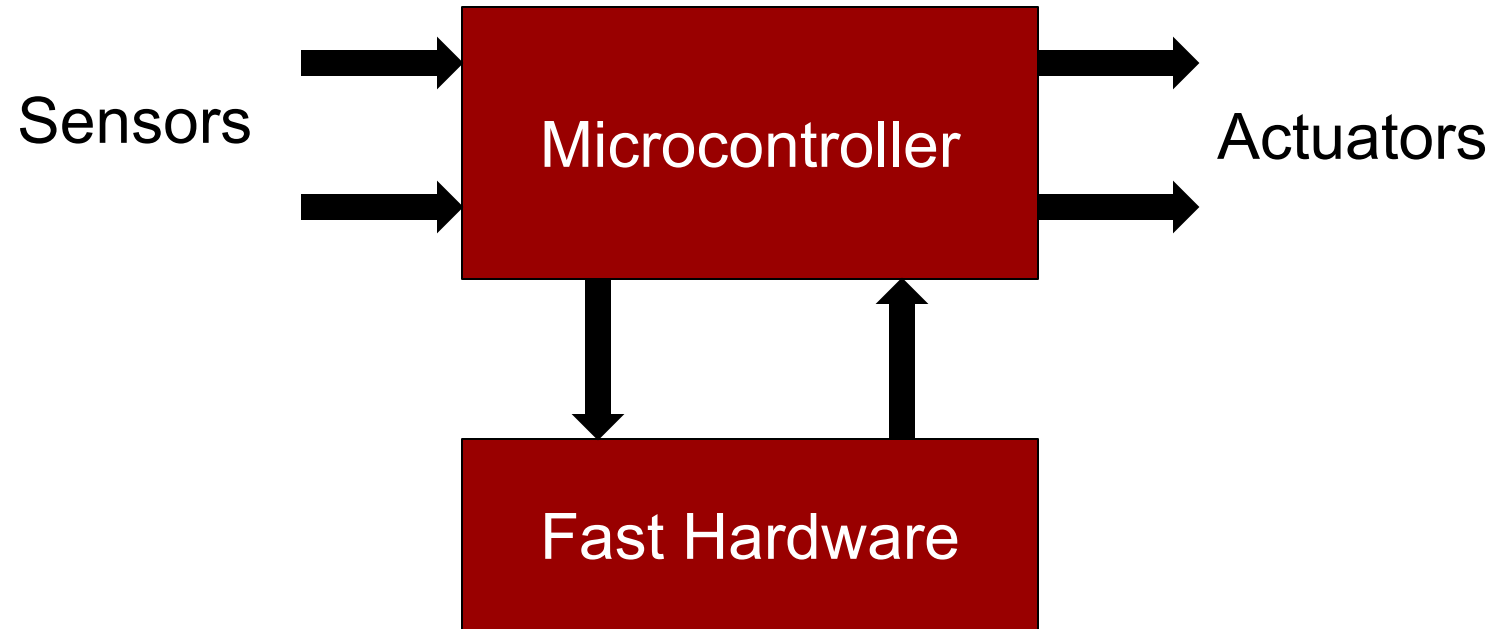


Embedded Software

Embedded Software

- Most software is embedded
 - 50 microprocessors in the average car
 - Several in many appliances
 - At the heart of most green electronic systems
- Embedded software forms the “slow” but “complex” part of most control loops
 - The fast, simple parts are in hardware (analog or digital)

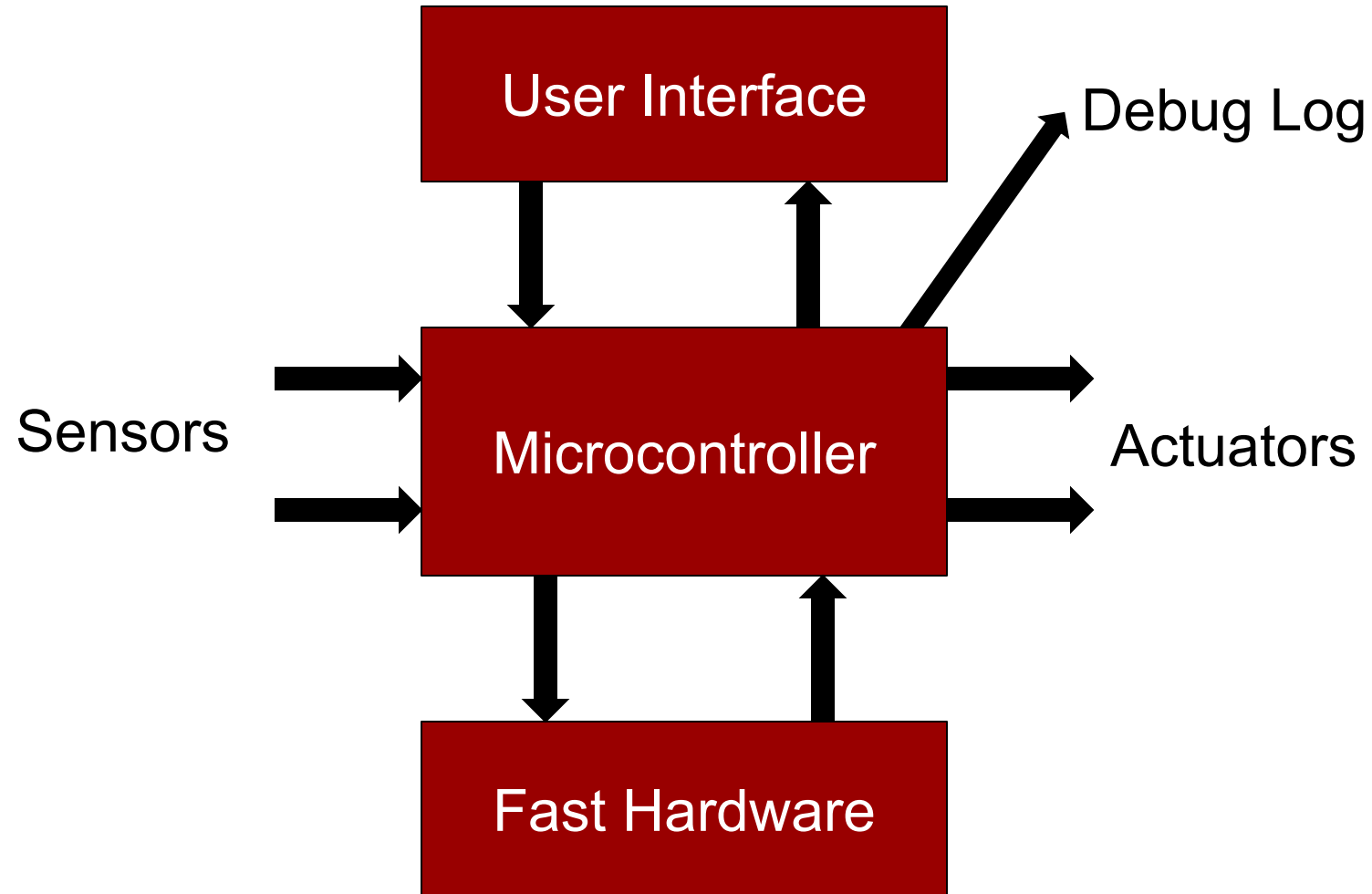
Embedded System



Example: PV Maximum Power Point Tracker



Sometimes Embedded Systems Interact with Humans



UI Design

- It should be obvious how it works
- Minimum number of “modes” (1 is best)
- Display clearly indicates
 - Current mode
 - Relevant data
 - Progress indicator if “waiting”
 - “Button” labels
- If the user needs to read the manual it's a bad UI
- Refresh display every 50ms
- Check buttons and update “state machine” every 50ms

Event Driven

- Embedded systems execute code in response to **events**
 - A real-time clock interrupt (e.g., every 1ms)
 - An I/O device completes an operation (e.g., A/D converter)
 - An input transitions (logic low-high, or analog threshold)
- Code is of the form
 - When **trigger**, do **action**
 - When **1ms clock tick**, **refresh display**
 - When **A/D completes**, **process data & start next conversion**
 - When **current over limit**, **shut down PWM**

Embedded vs Conventional Software

Embedded

- Event driven
 - Most code runs in response to interrupts
- Real-world I/O
 - A/D in, D/A and PWM out
- Real-time
 - Update PWM within one cycle

Conventional

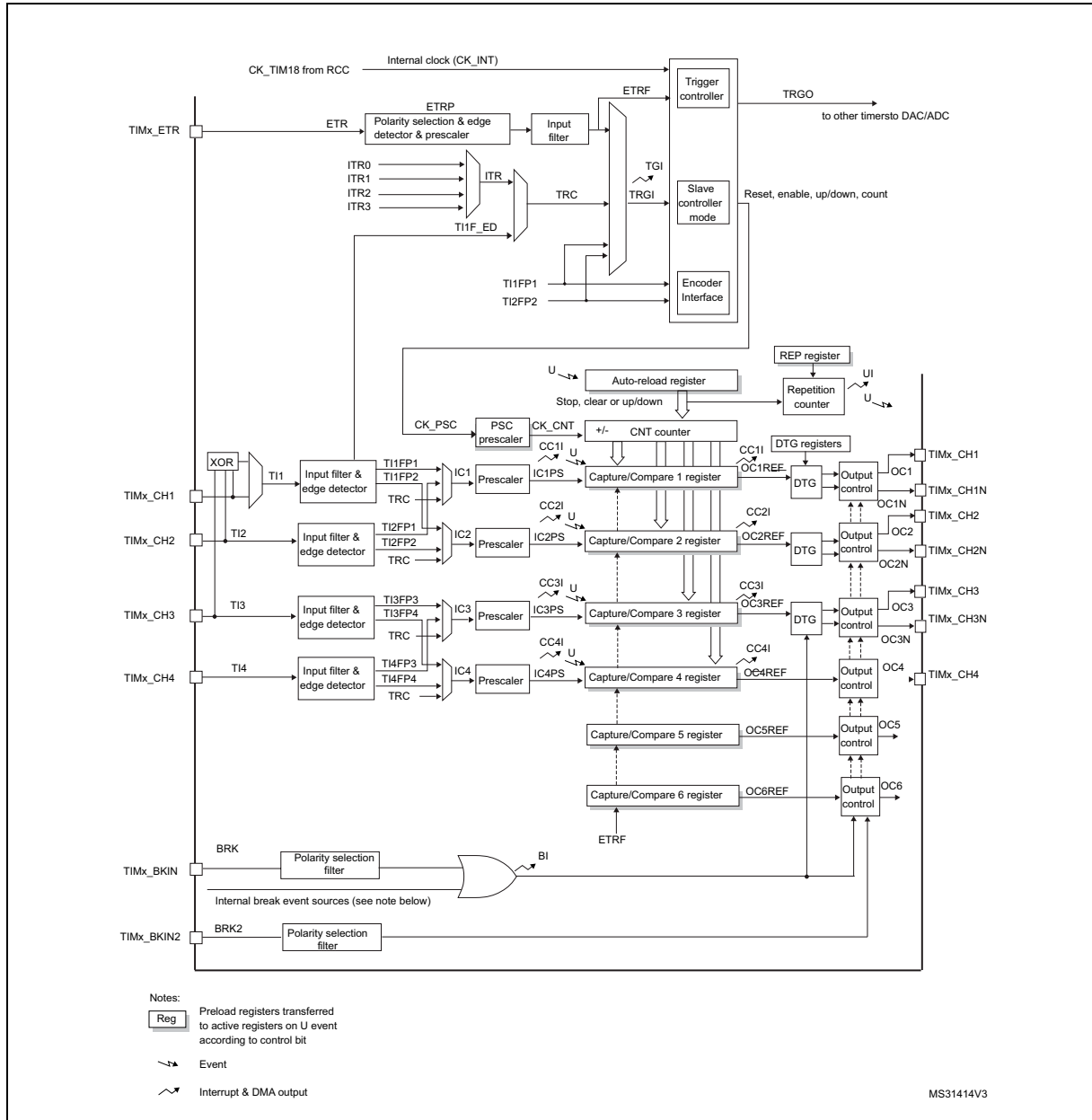
- Run code in program order
 - In response to user input (e.g. mouse click)
- User Interface I/O
- No hard time constraints

Real-Time Clock

- Microcontroller has a counter-timer that can be programmed to interrupt the processor periodically (e.g., every 1ms).

STM32 Timers

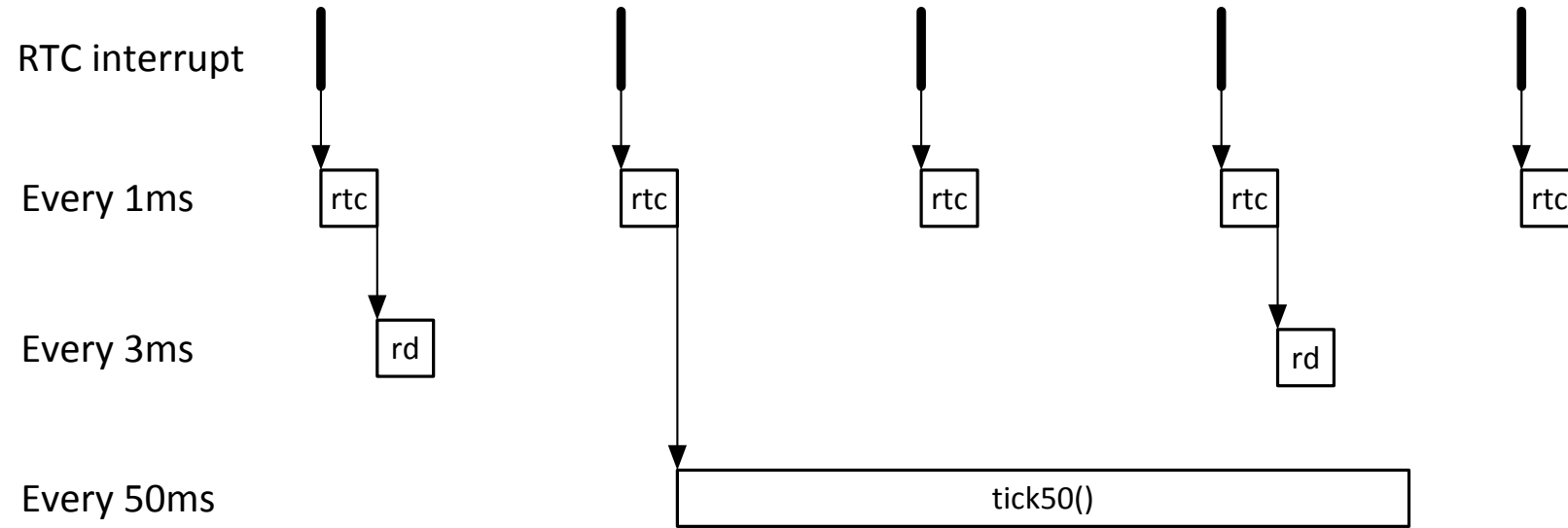
Figure 106. Advanced-control timer block diagram



Library Code

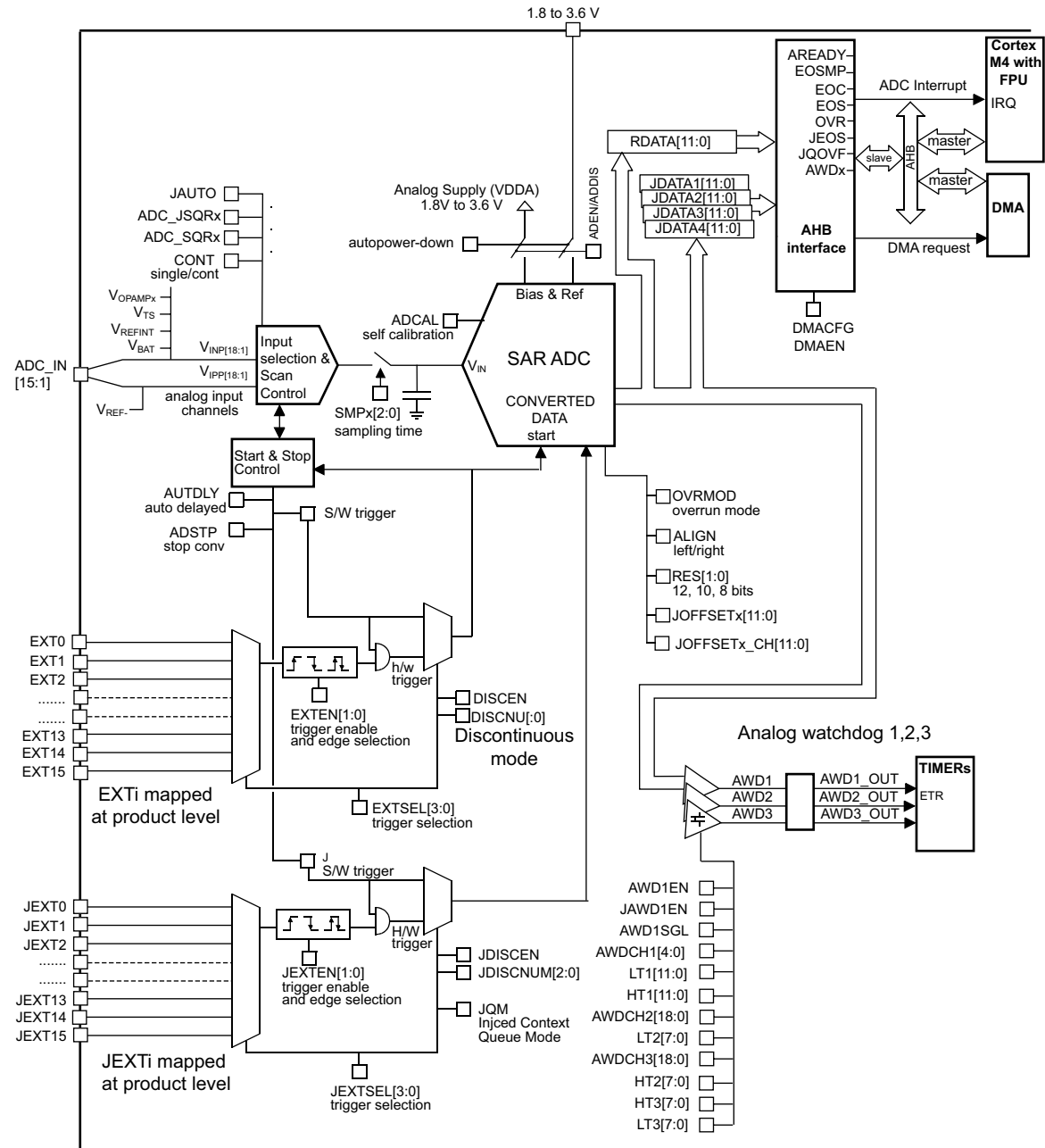
```
void my_callback1();  
timer_init();  
timer_id_t my_timer1 = timer_register(1, &my_callback1,  
                                     GE_PERIODIC);  
timer_start(my_timer1);
```

Timer Interrupt



I/O Completion

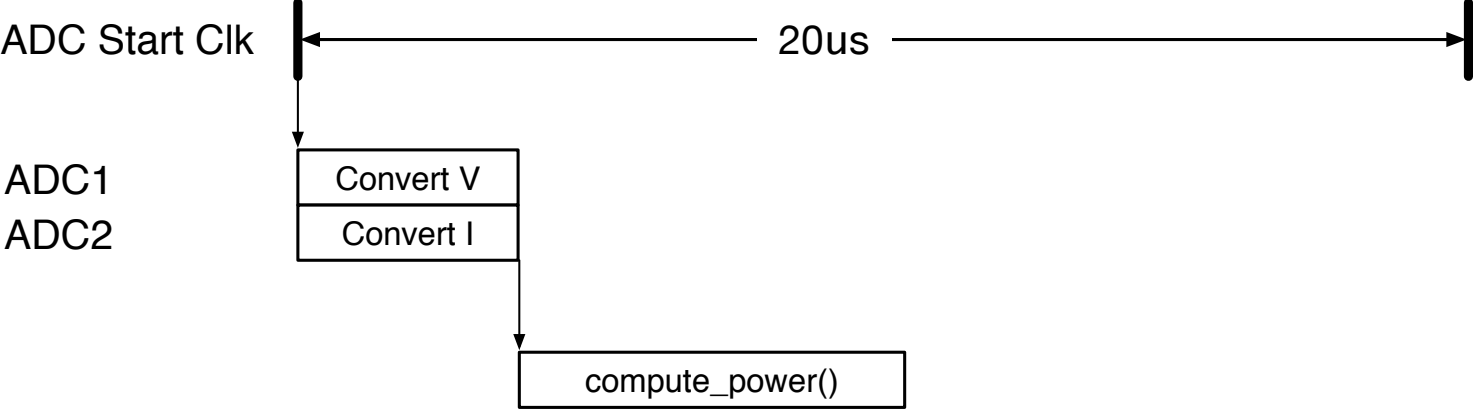
- I/O devices interrupt when they complete an operation
- Handler should
 - Process input data (if an input device)
 - Start next operation



A/D

```
adc_init();  
adc_set_fs(50000); // 50kHz sample frequency  
adc_callback(1, &calculate_power); // callback  
adc_enable_channel(1); // channels 1.1 and 2.1  
adc_start(); // start the 50kHz  
  
void calculate_power(uint32_t data) {  
    uint16_t chan1 = data & 0x00ff;  
    uint16_t chan2 = (data & 0xff00) >> 16;  
  
    power_result = (float) chan1 * V_PER_DIV * (float) chan2 *  
A_PER_DIV;  
}
```

A/D Timing



- What happens if timer interrupt handler is still running when ADC interrupt occurs?

- Make sure your interrupt driven code runs quickly!!!
- You can toggle a GPIO pin at entry and exit of your code to see how long it takes on a scope.

Volatile Variables

- If you run code with “interrupts enabled” other code may run “at the same time”.
 - And long running code (more than 200us) should not disable interrupts
 - Enable interrupts after “critical section”.
- That “other code” may change variables
- Don’t expect variables to be stable while “background” code runs!

Integer Arithmetic

- Much embedded code uses fixed-point arithmetic.
- Floating-point takes too long (on an integer uC)
 - Some modern uCs (e.g., STM32F3) have fast FP
- Fixed point has plenty of precision for typical control values
 - **int** (16 bits) [-32,768, 32,767]
 - **long** (32 bits) [-2,147,483,648, 2,147,483,647]
- But you need to scale your variables and check for overflow
 - Example – 16b represents voltage in mV, current in mA
 - Multiply mV and mA to get 32b power in uW
 - Divide by 1,000 and truncate to get 16b power in mW
 - Shift and multiply – don't really divide – its expensive
 - Multiply by 1.024 and shift right 10

Fixed-Point Arithmetic

- Scale numbers to be in $[-1, 1)$
- Binary point is just to the right of the sign bit
 - s.bbbbbbb...
- On multiply retain high half of result
- General fixed-point representation
 - saaaaaaaaa.bbbbbbb...
 - Represents a value to $1/64$
 - Need to shift after a multiply to put binary point in the correct place
- Need to think about the range and precision of your variables

Floating Point

- ST32F3 has fast 32-bit floating point
- Max value of $\sim +/-3 \times 10^{38}$
- Min non-zero value of $\sim 1.2 \times 10^{-38}$
- Represents numbers to better than six sigfigs

- Scale inputs and outputs

```
float_volts = (float) int_volts * VOLTS_PER_DIV ;
```

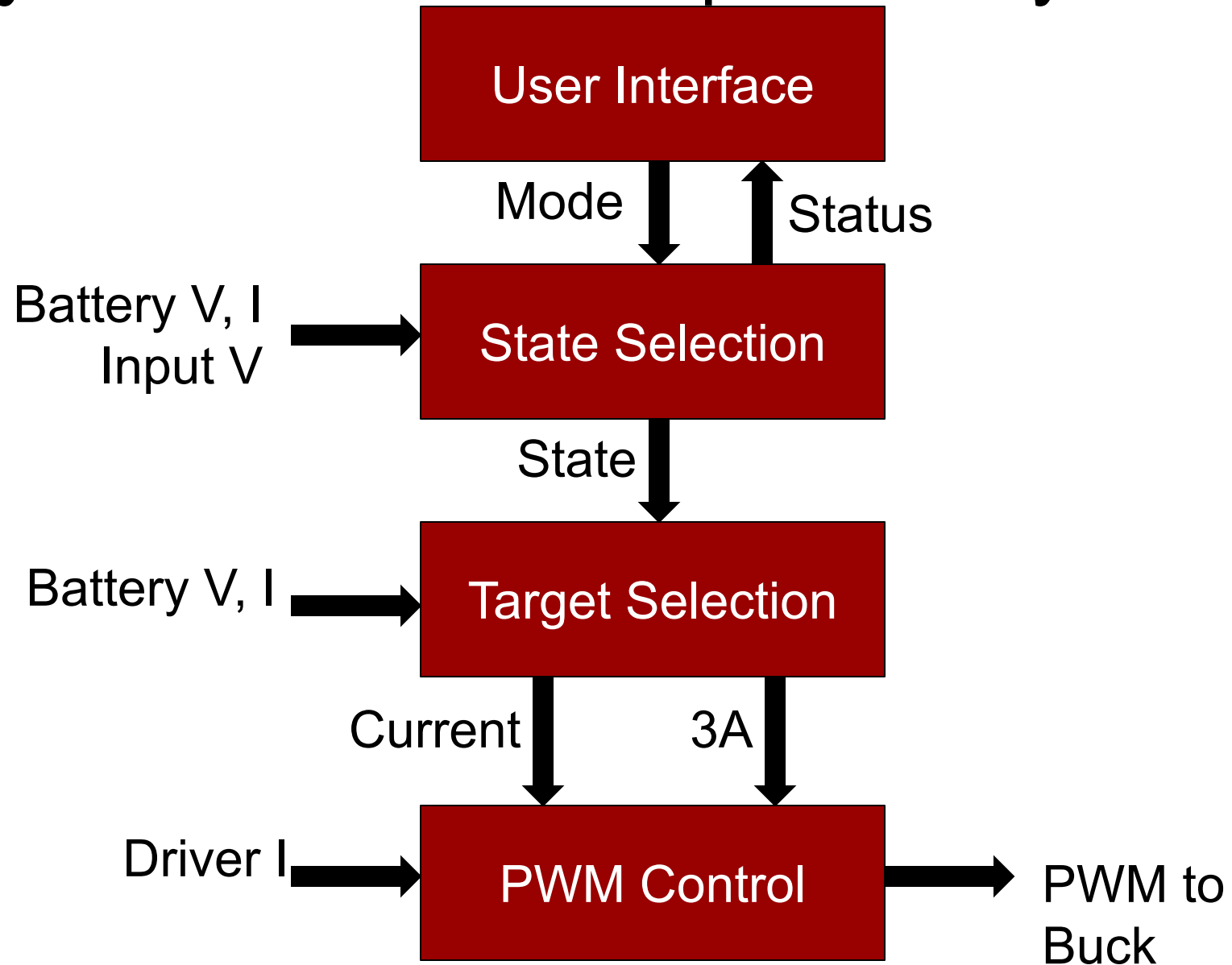
– 12b ADC with 3V range – $3V/4096 = 0.000732$ volts/div

- Always check for range, never equality

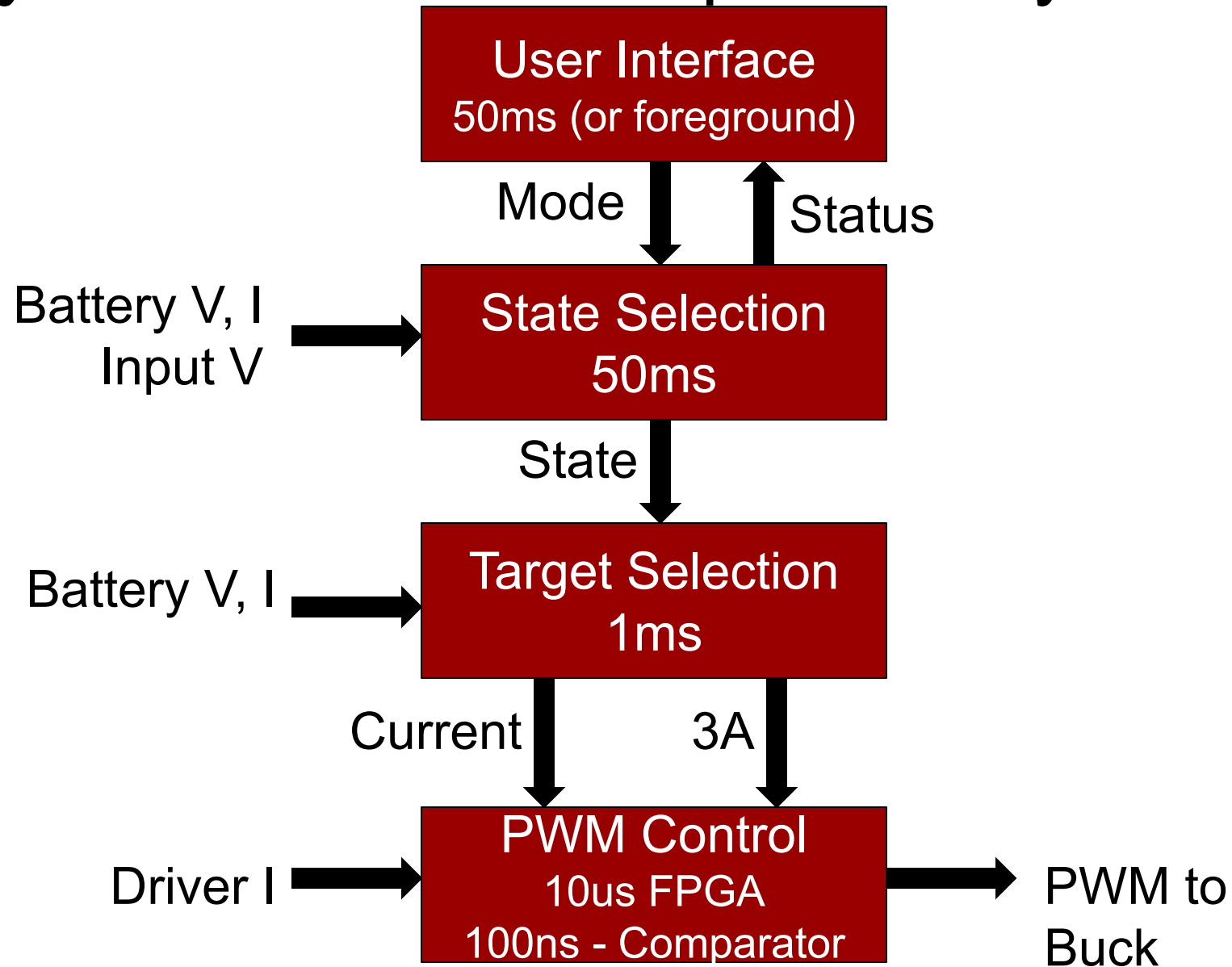
Organization of Code

- Control
- User Input
- Display
- Monitoring
- Housekeeping

Layered Control: Example Battery Charger



Layered Control: Example Battery Charger



Example Photovoltaic Controller

- Mode selection
 - Line test, Panel test, MPPT, etc...
- Monitors
 - Line (current, voltage, frequency, ...) (anti-islanding)
 - Internal operation (voltages, currents, temps)
 - Panels (current, voltage)
- Maximum Power-Point Tracking
 - Search algorithm for peak power point
- Inverter control
 - Convert 400V DC to 240V AC – in sync with line
- User interface
 - Display status, serial output, network connections, ...

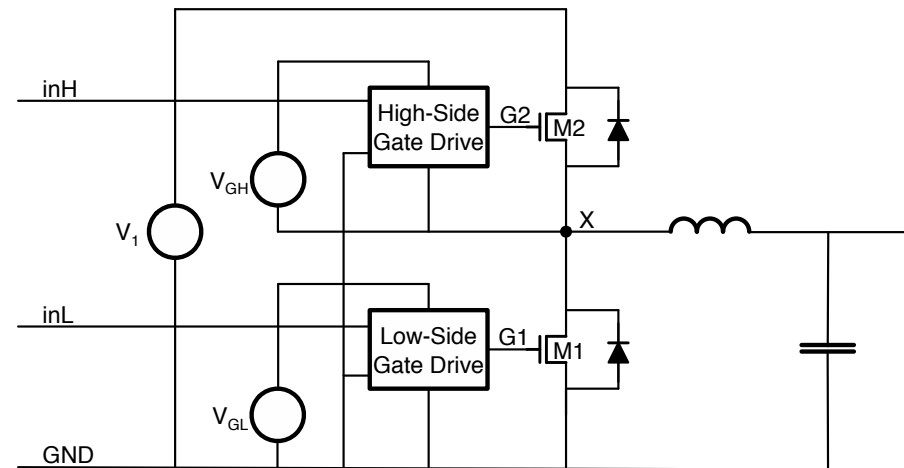
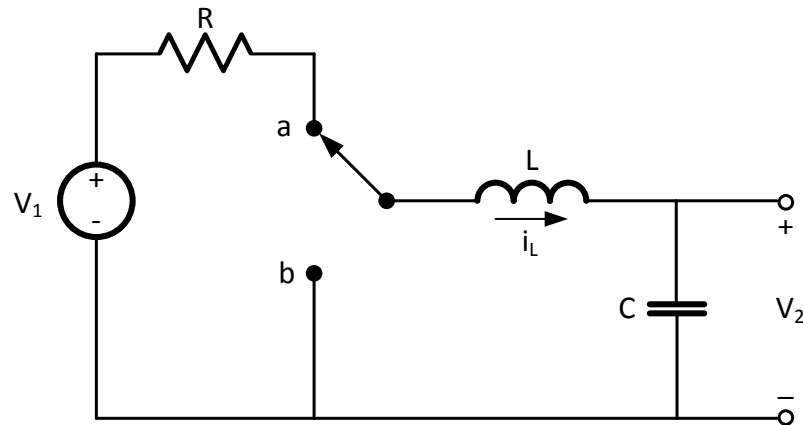
Summary – Embedded Software

- Embedded software consists of event handlers
 - When **trigger** do **action**
- Timer is used to perform periodic tasks
 - Schedule A/D converter every 20us
 - Report status on serial port every 10ms
 - Update mode every 50ms
 - Etc...
- I/O devices trigger handler on completion
 - Process input data
 - Start next I/O operation
- Layered Control
 - High-level control loop sets parameters of lower level control loops
 - Fastest loops may be in hardware

Power Electronics

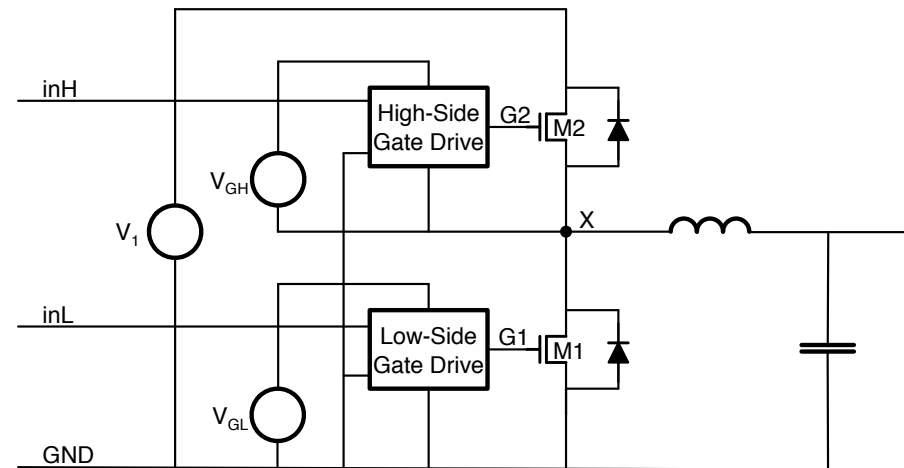
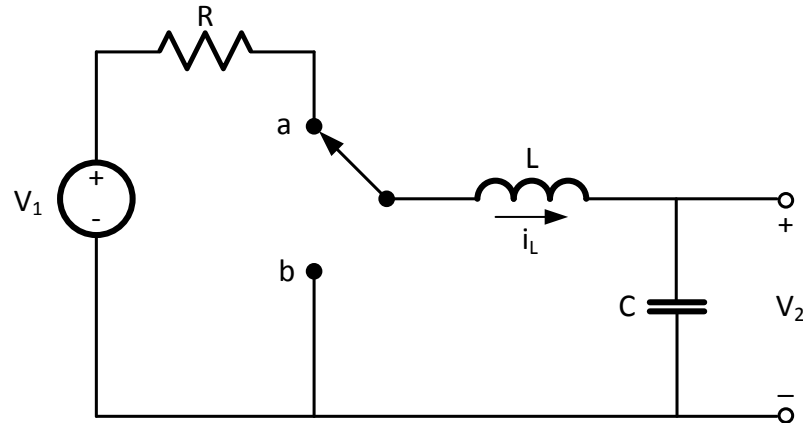
Part 1 – Power Devices

Real Switches



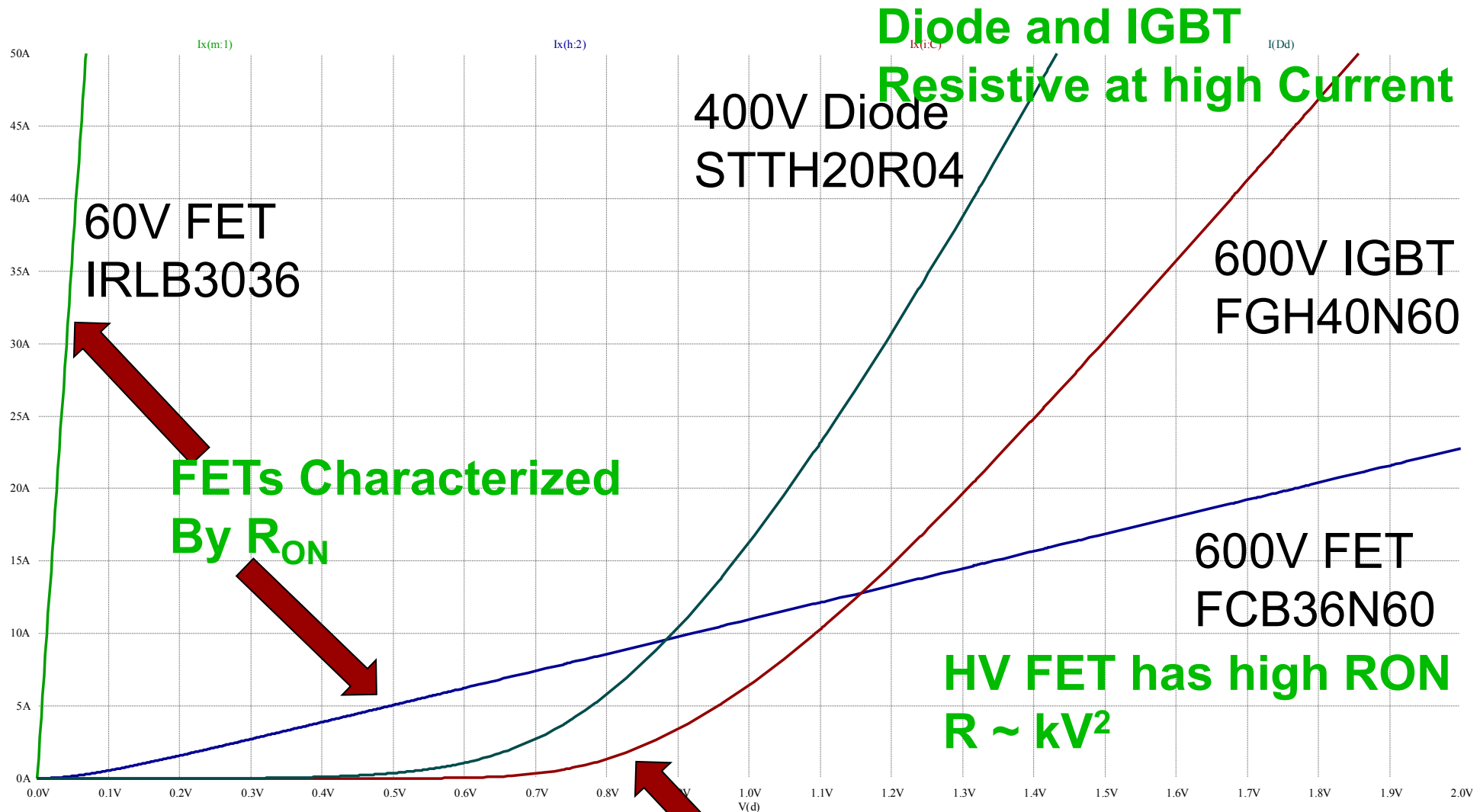
Real Switches

- Finite switching time
- Finite voltage blocking
- Non-zero on resistance
- Parasitic L and C

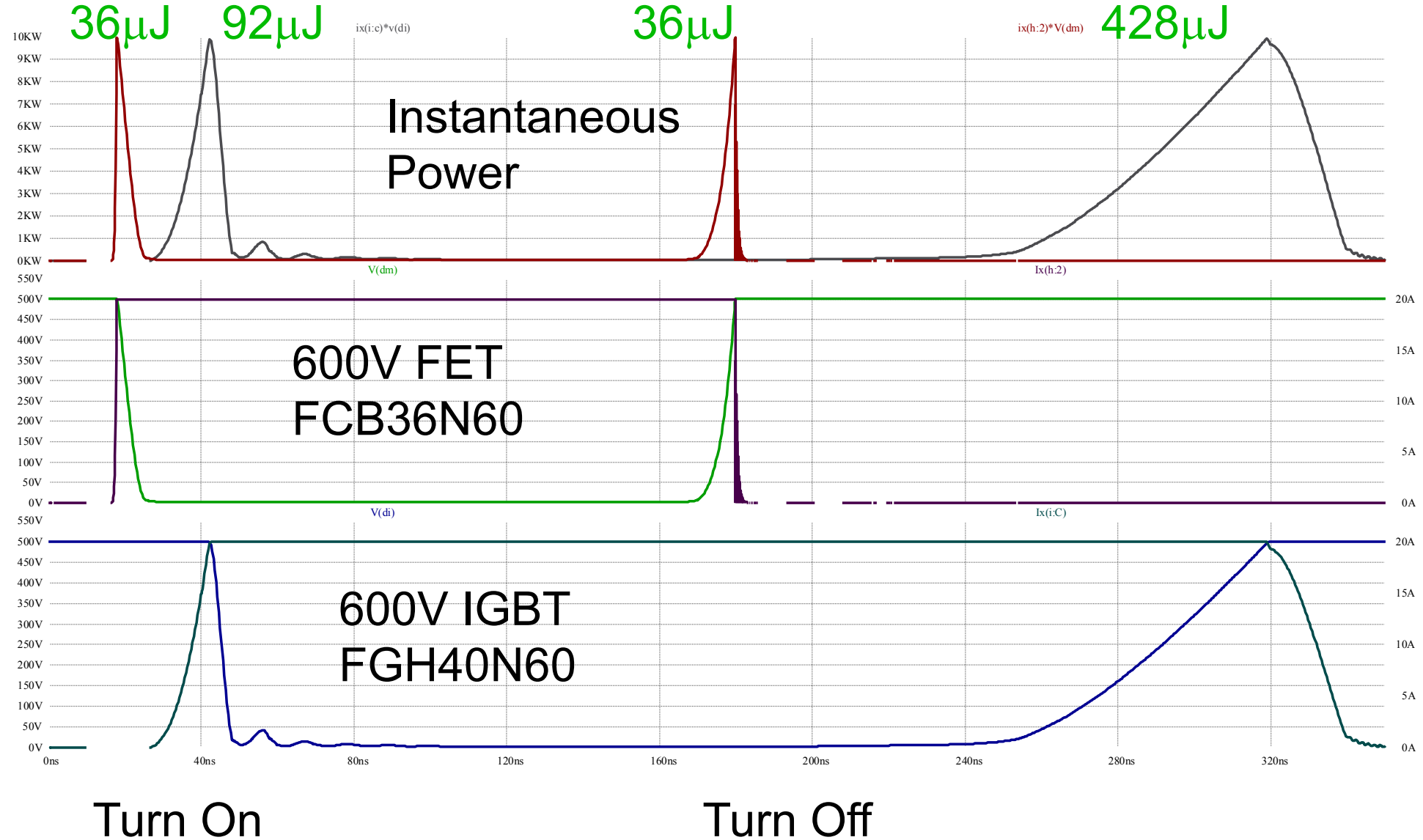


Quick Summary in a Few Pictures

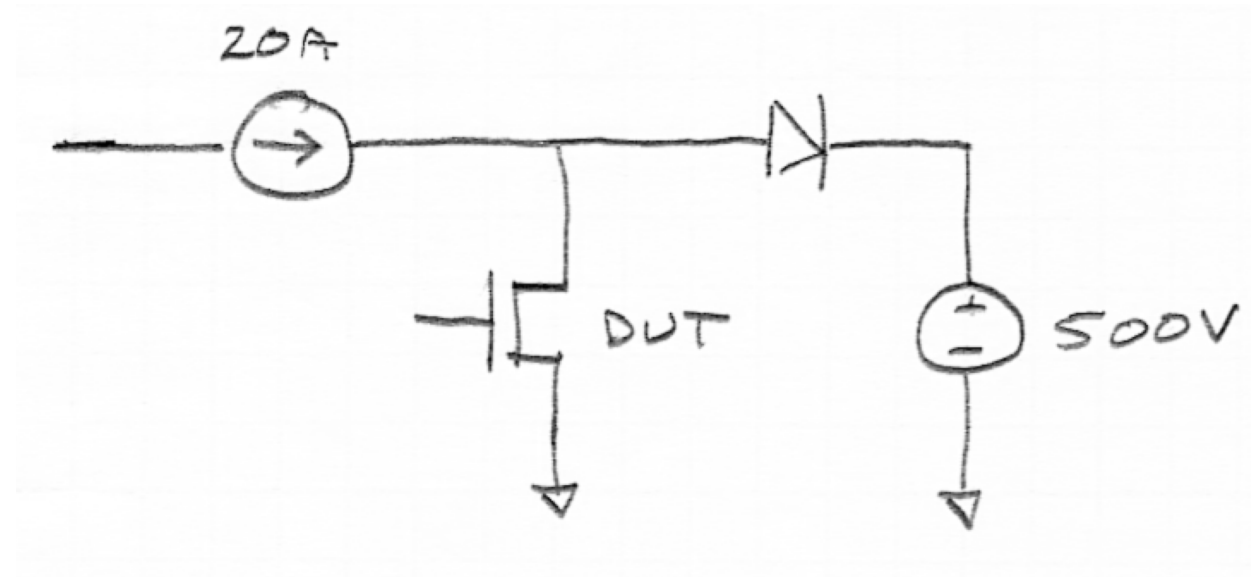
DC I-V Characteristics of On Switches



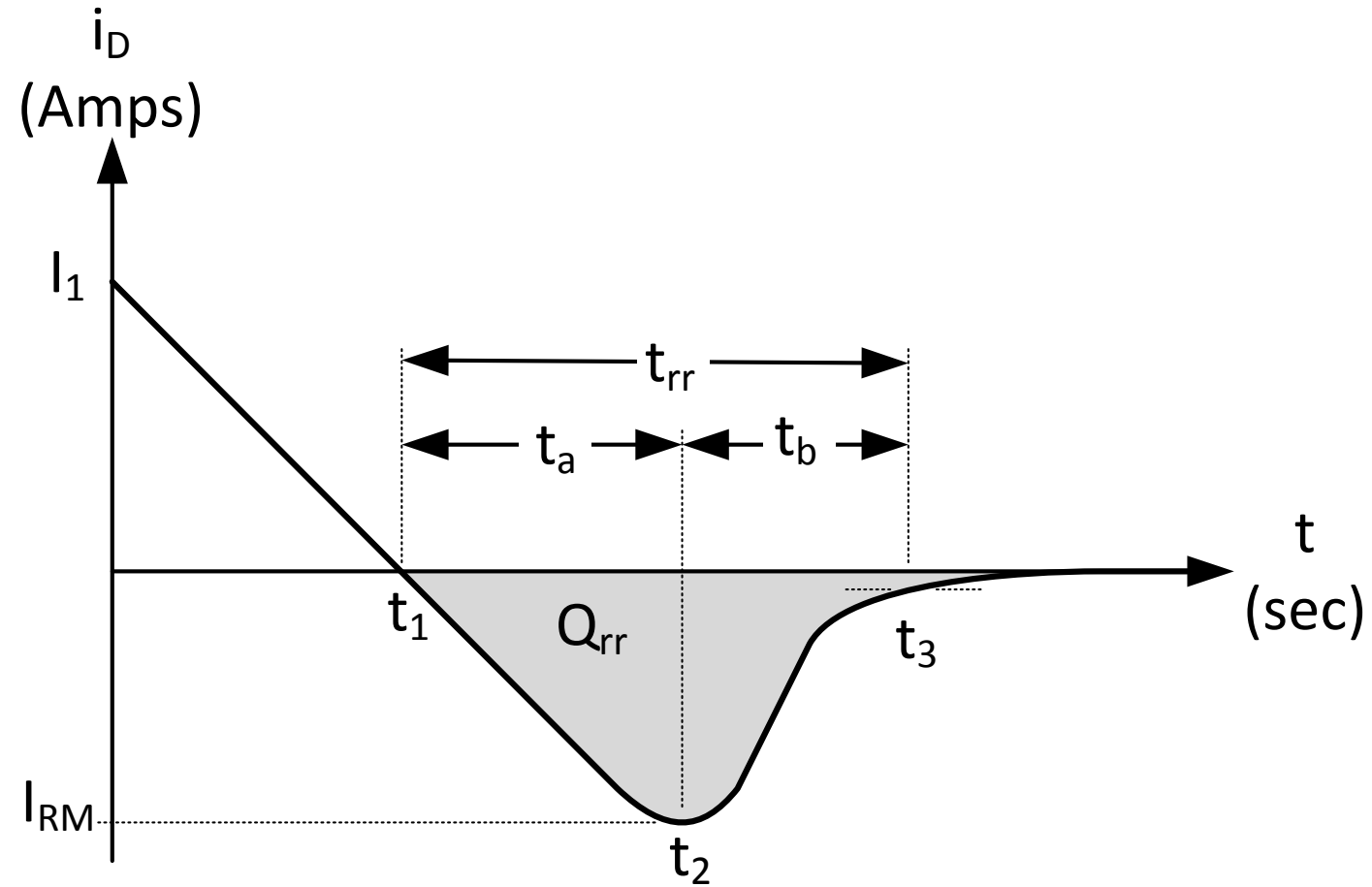
Transient Response of FET and IGBT



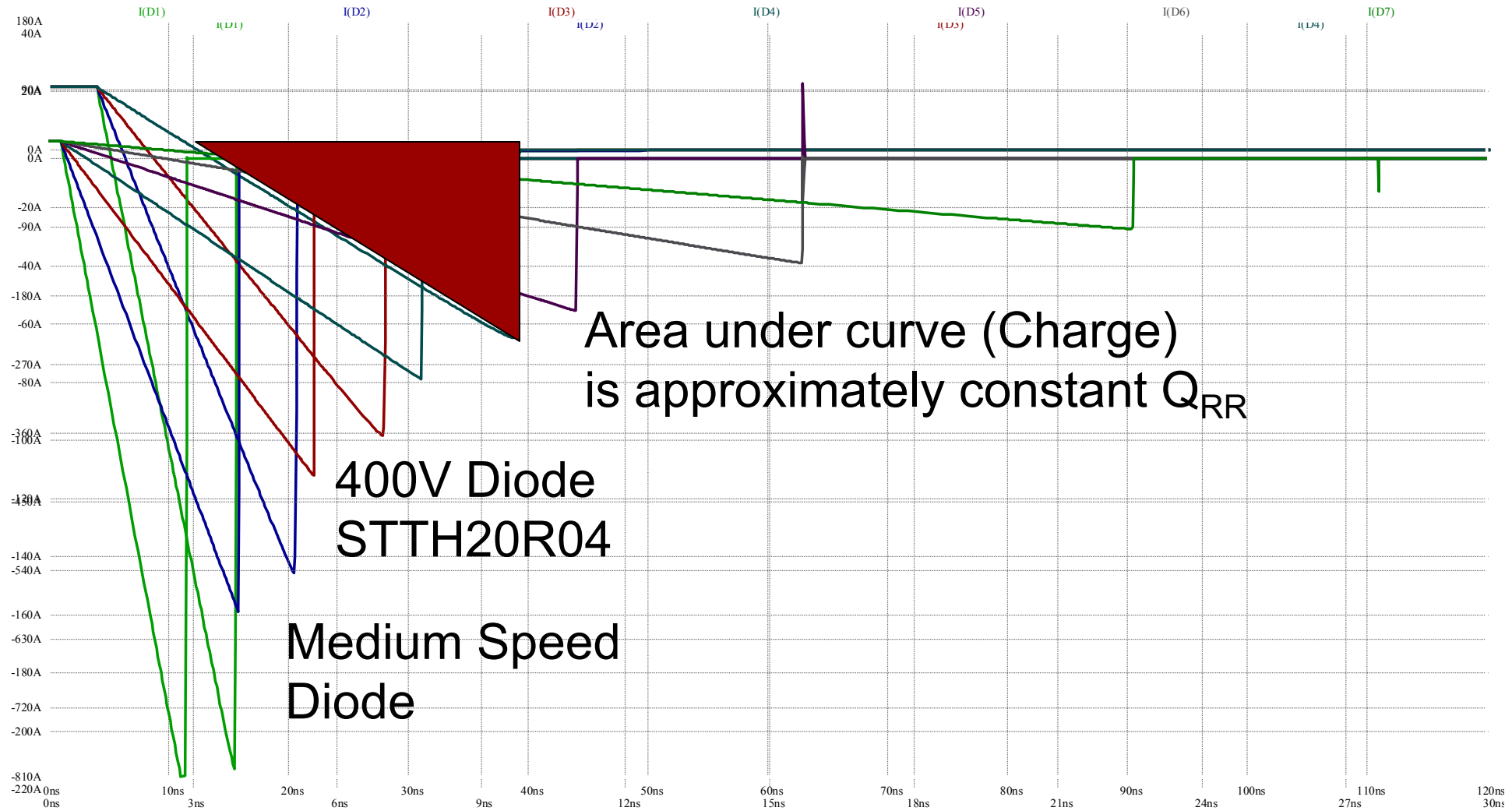
Boost Configuration for Transient Test



Diode Reverse Recovery

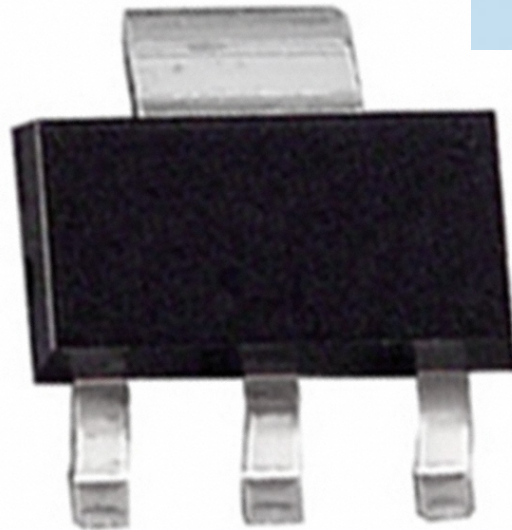
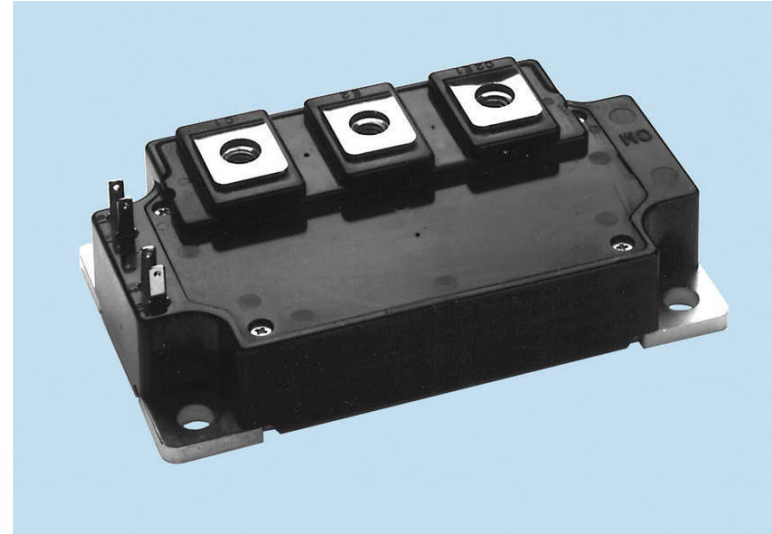
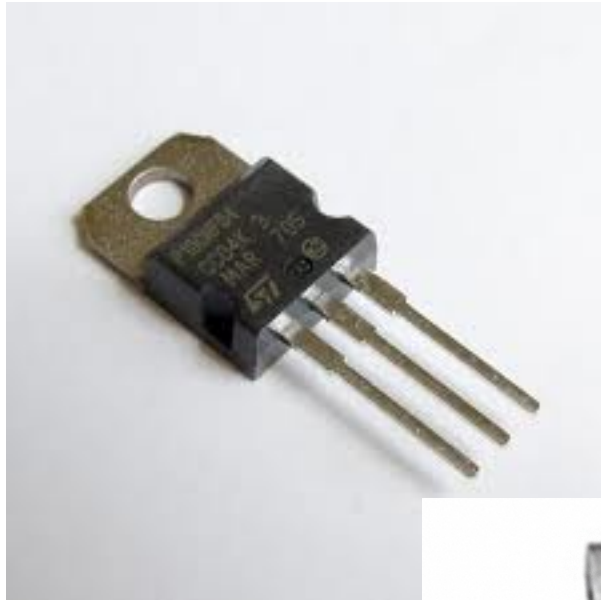


Diode Reverse Recovery



Power MOSFETs

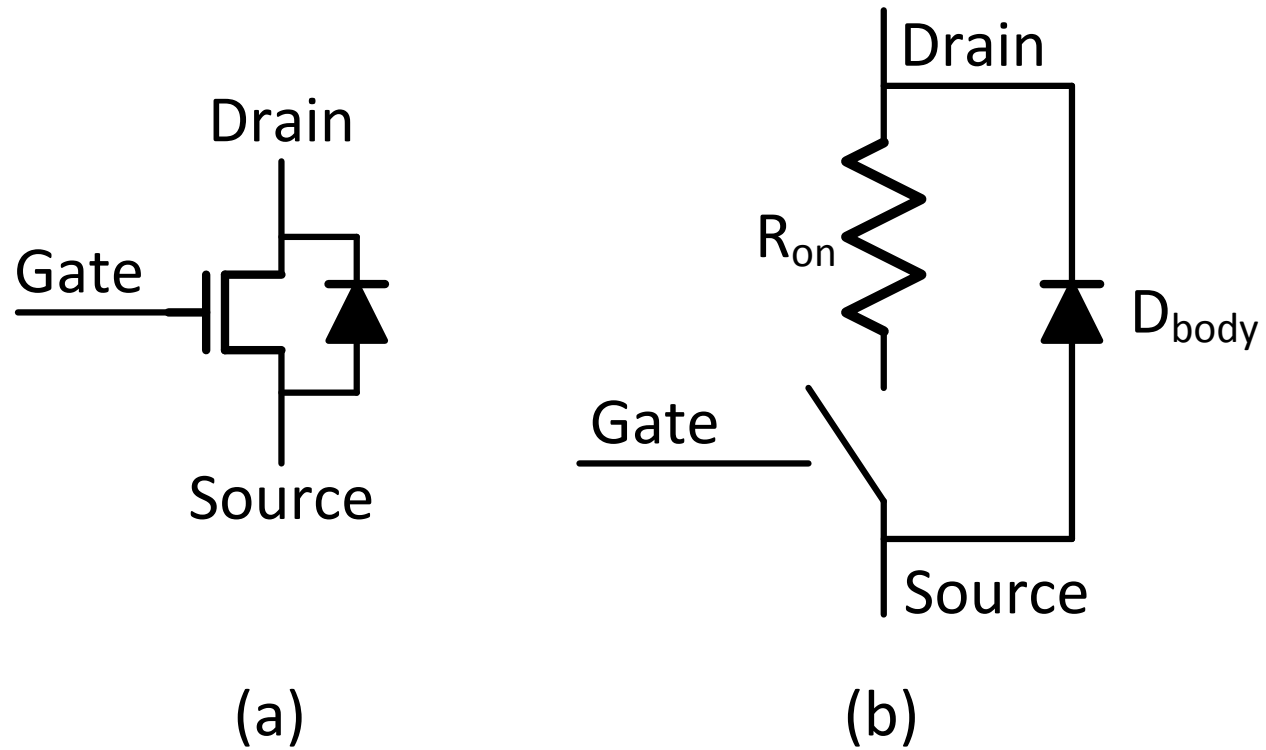
Power MOSFETs are your friends



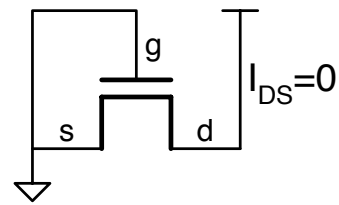
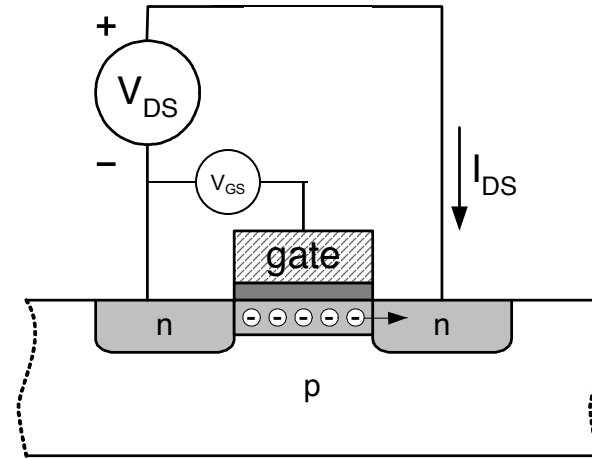
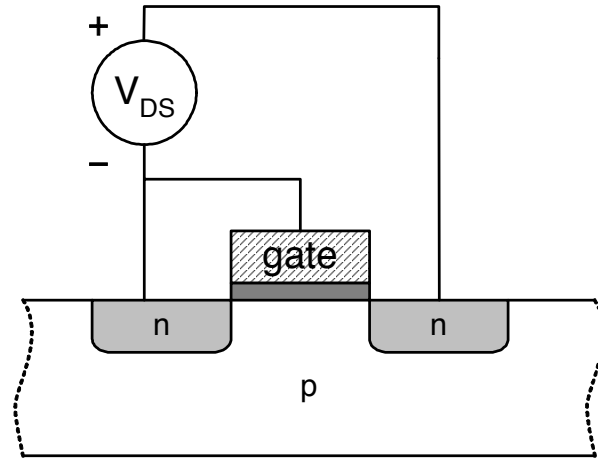
MOSFET Properties

- Fast switching time 10-50ns
 - Low switching losses
- Low conduction losses at low voltages
 - V^2/R of 1-2MW
 - e.g., at 20V $R = 0.4\text{m}\Omega$ (4mV drop at 10A)
 - Typically better than IGBT up to ~400V
- Easily paralleled for lower R_{on} , More I, or easier cooling
- Integral body diode in DMOS FET
- Avalanche breakdown can be used to “snub” overshoot

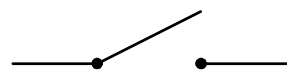
MOSFETs are Switches



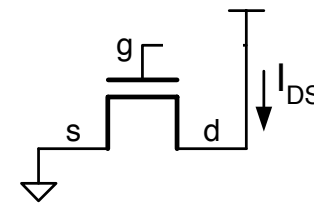
How do they Work



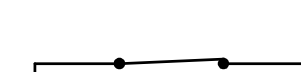
$|g=0$



(a)

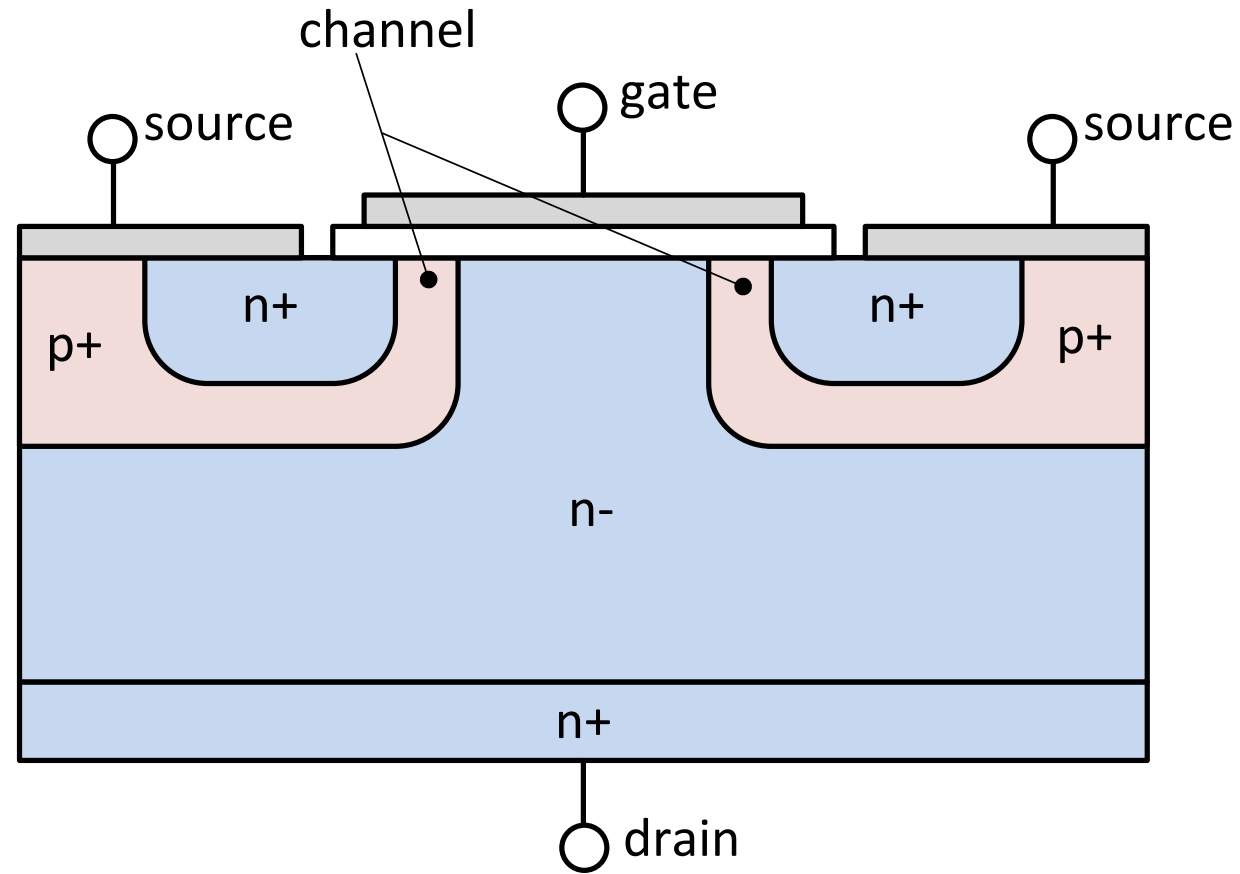


$|g=1$

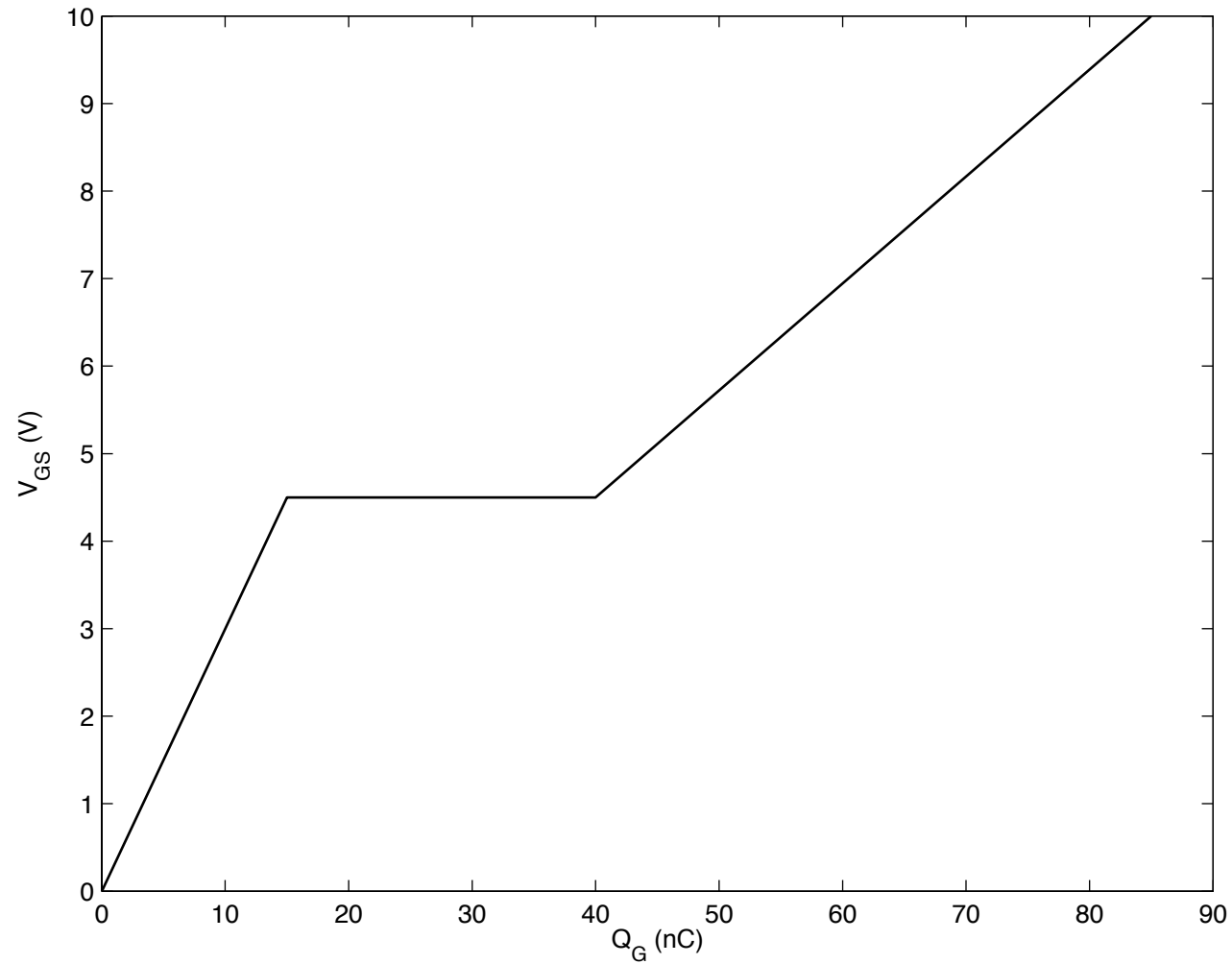


(b)

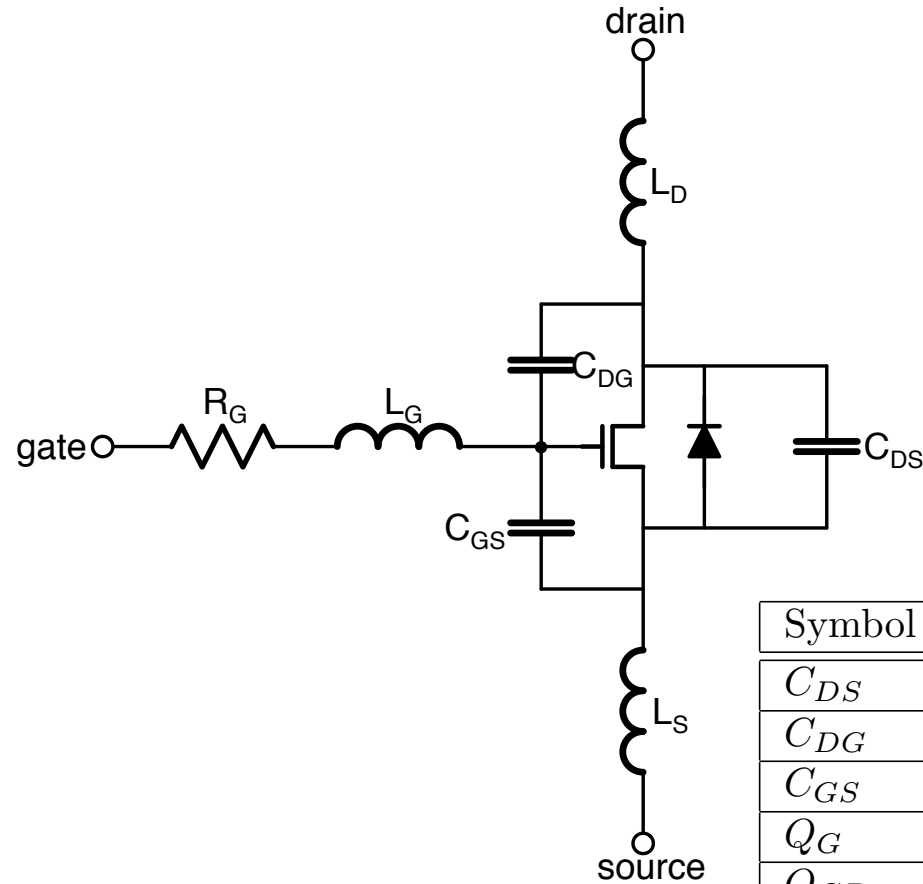
Power MOSFET (DMOS) Structure



Gate Charge vs V_{GS}



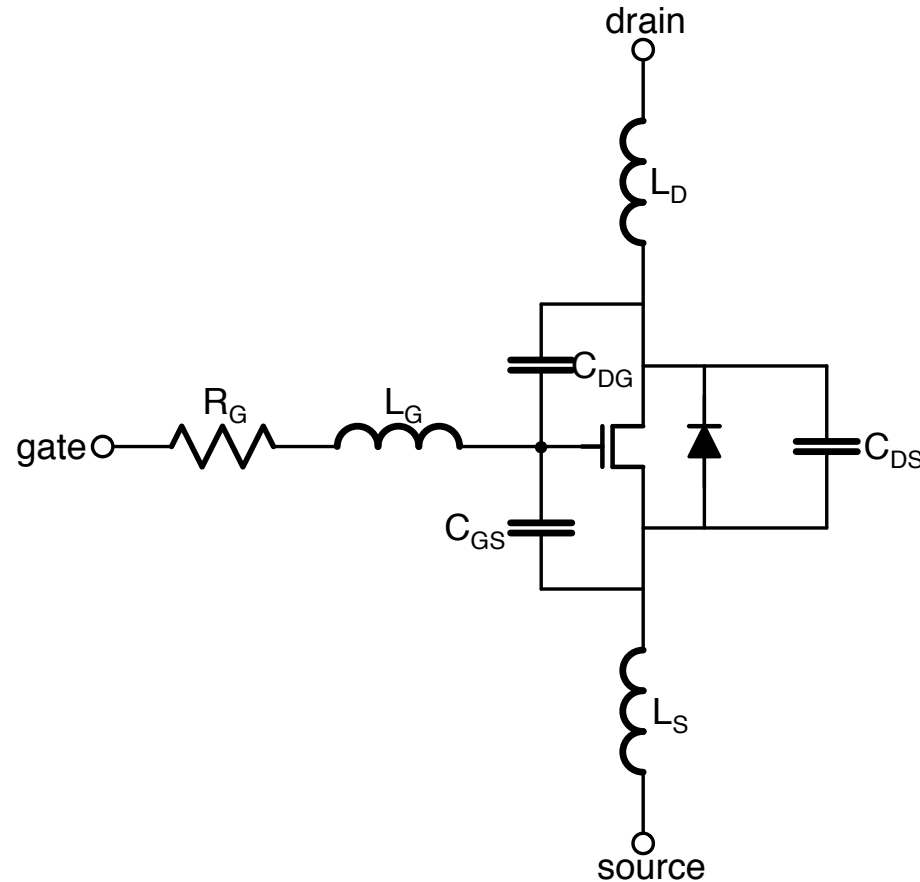
Parasitics



600V 0.1Ω FET
TO220 Package

Symbol	Value	Units	Description
C_{DS}	200	pF	Drain-source capacitance
C_{DG}	70	pF	Drain-gate (Miller) capacitance
C_{GS}	3600	pF	Gate-source capacitance
Q_G	86	nC	Total gate turn-on charge
Q_{GD}	35	nC	Gate-drain turn-on charge
L_S	7	nH	Source inductance
L_D	3	nH	Drain inductance
L_G	7	nH	Gate inductance
R_G	1.5	Ω	Gate resistance

Parasitics



C_{DS} – CV^2 energy

L_D, L_S – I^2L energy

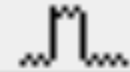
slows switching,
overshoot,

corrupts gate drive

C_{DG} – slows turn on

R_G, L_G – slow device turn-on

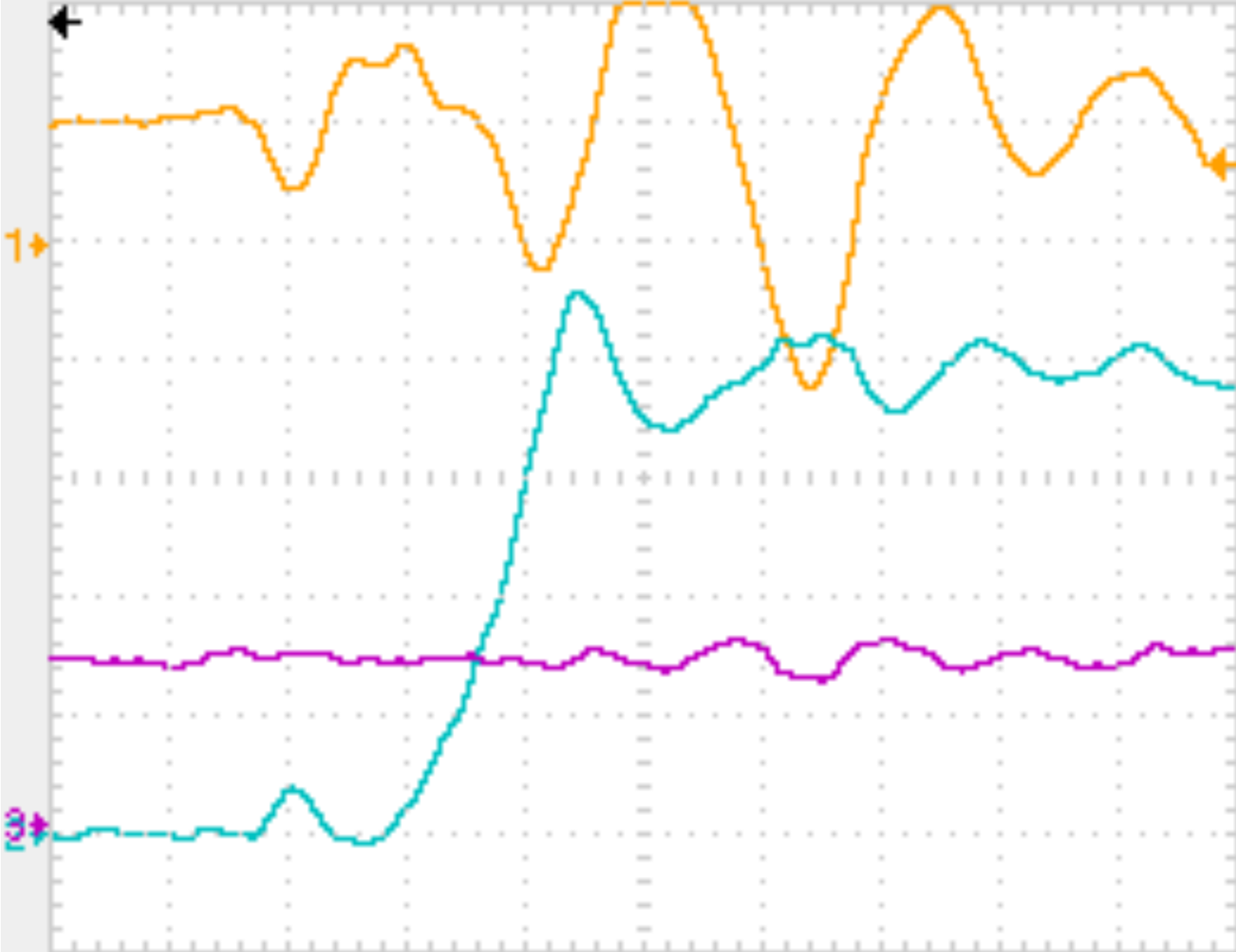
Tek



● Acq Complete

M Pos: 91.11µs

MEASURE



CH2
Rise Time
5.220ns

CH2
Fall Time
?

CH2
Pk-Pk
464V

CH3
Mean
280V

CH2
Pos Width
?

CH1 5.00V

CH2 100V

M 5.00ns

CH1 / 3.40V

CH3 200V

12-Nov-14 00:50

<10Hz

Typical MOSFETs

Device	20V				GaN	SiC	Units
		IRLB3036	IRFB4227	FCB36N60N	EPC2010	C2M0025120	
V_{DSmax}	20	60	200	600	200	1200	V
R_{on}		1.9	20	81	18	25	m Ω
Q_G		91	70	86	5	161	nC
C_{oss}		1020	460	80	270	220	pF
I_{Dmax}		195	65	36	12	90	A
I_{DM}		1100	260	108	60	250	A
E_{AS}		290	140	1800			mJ
P_{max}		380	330	312		463	W
V^2/R		1.9	2.0	4.4	2.2	58	MW
V^2/RQ_G		21	29	52	440	360	mV/s

Power MOSFETs should be ON or OFF

They are not happy in between

- IRLB3036
- Can handle 60V (when its off)
- Can handle 195A (when its on – if you can cool it)
 - $I^2R = (195)^2(0.002) = 76W$
- But it can't handle 60V and 195A at the same time
 - $P = VI = (60)(195) = 11.7kW$
 - At least not for very long
- Turn them on and off quickly
- Best circuits are “soft switching”
 - Zero-current switching (ZCS) or zero-voltage switching(ZVS) or both.

Power Diodes

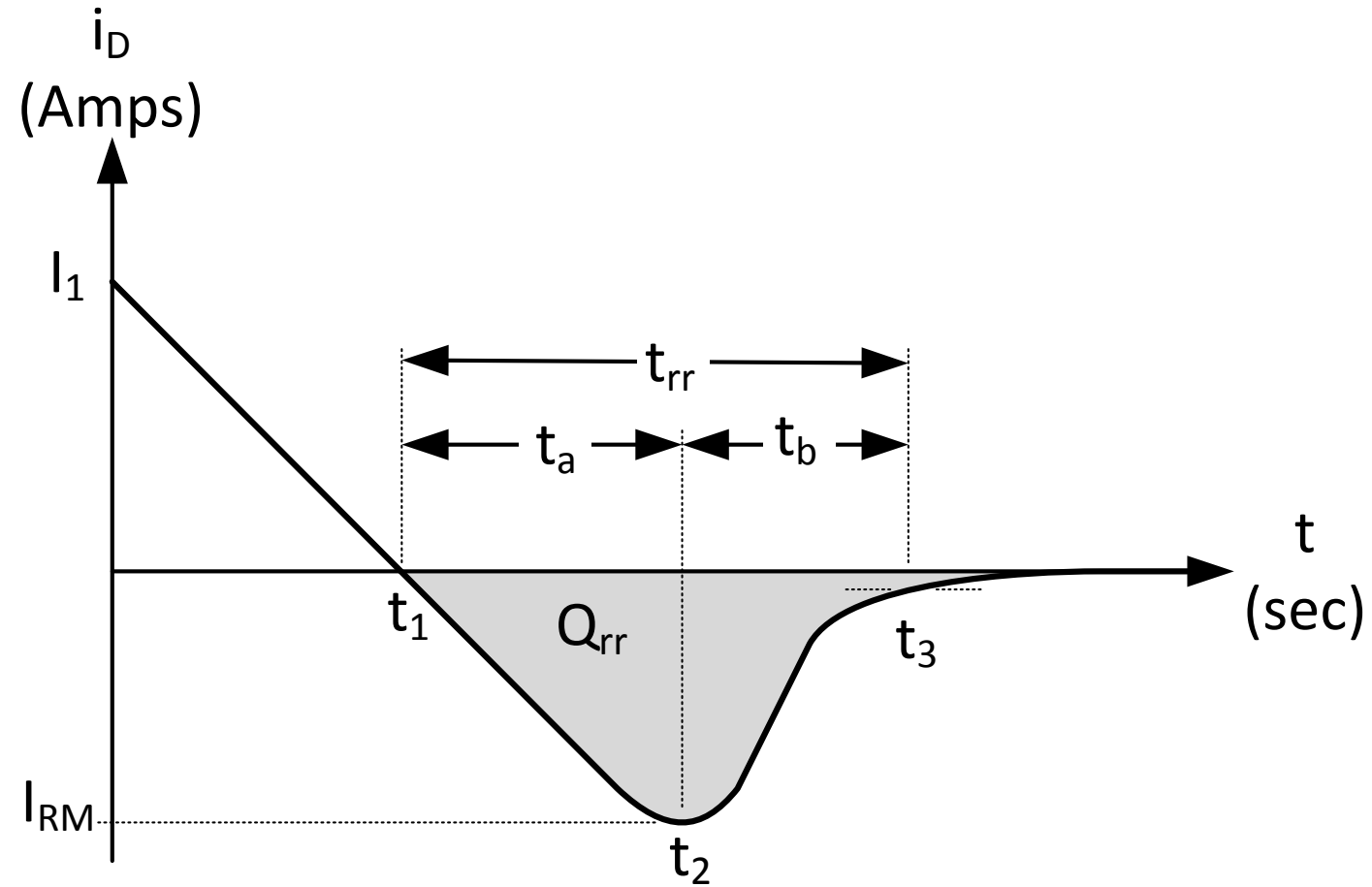
Diode Properties

- Self-controlling switch
 - Allows current in one direction
 - Turns off when current reaches zero (in theory)
- Relatively fixed voltage drop independent of current
 - 0.5 to 2.0V
 - High losses at low voltages
- Care required to operate in parallel
 - Current hogging
- Turn-off delay
 - Must clear space charge out of junction
- Turn-on delay
 - Negligible for most fast diodes, but some are problematic

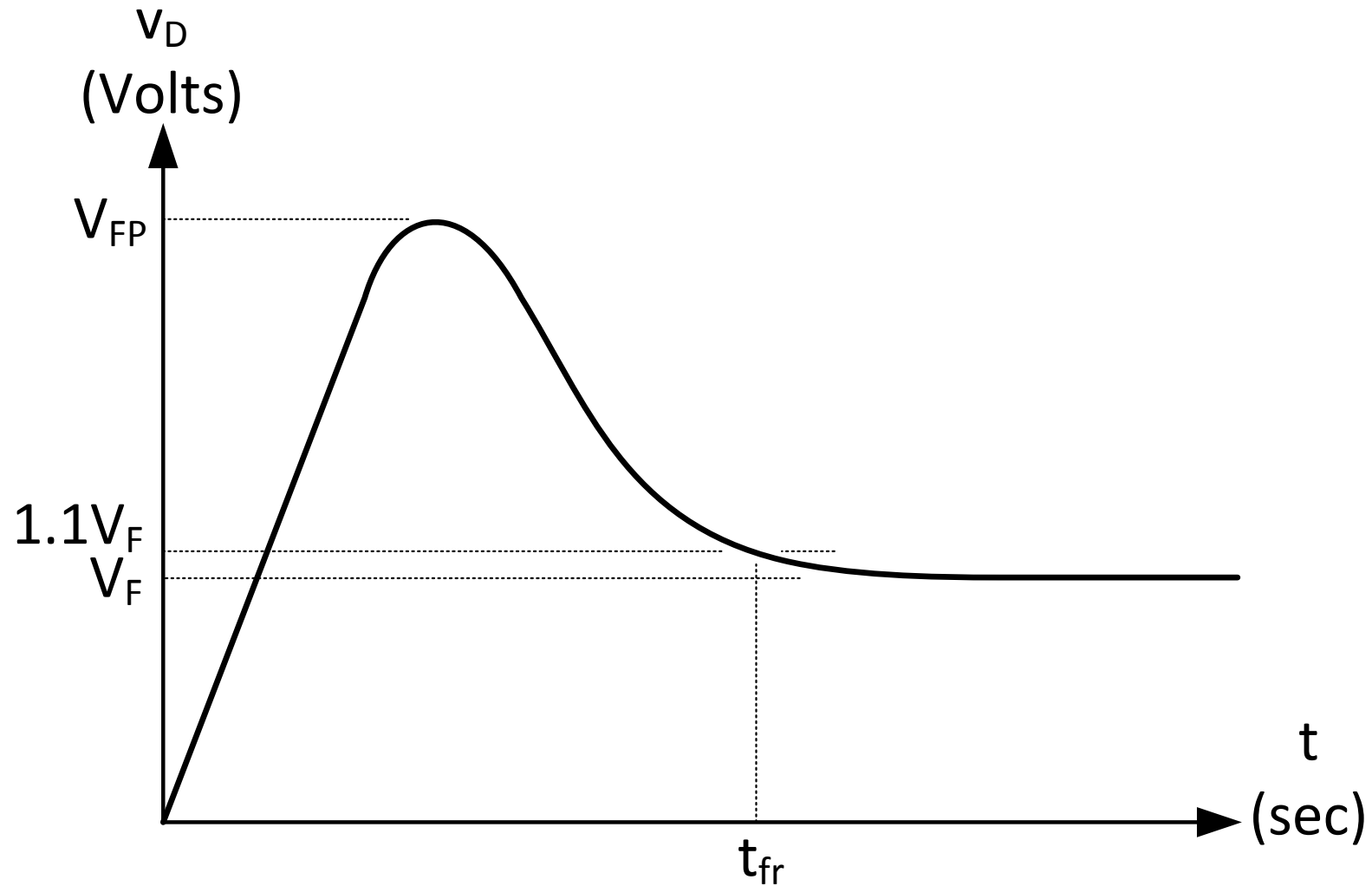
Key Parameters

- Reverse breakdown voltage
- Maximum current
- Reverse recovery time
- Junction capacitance

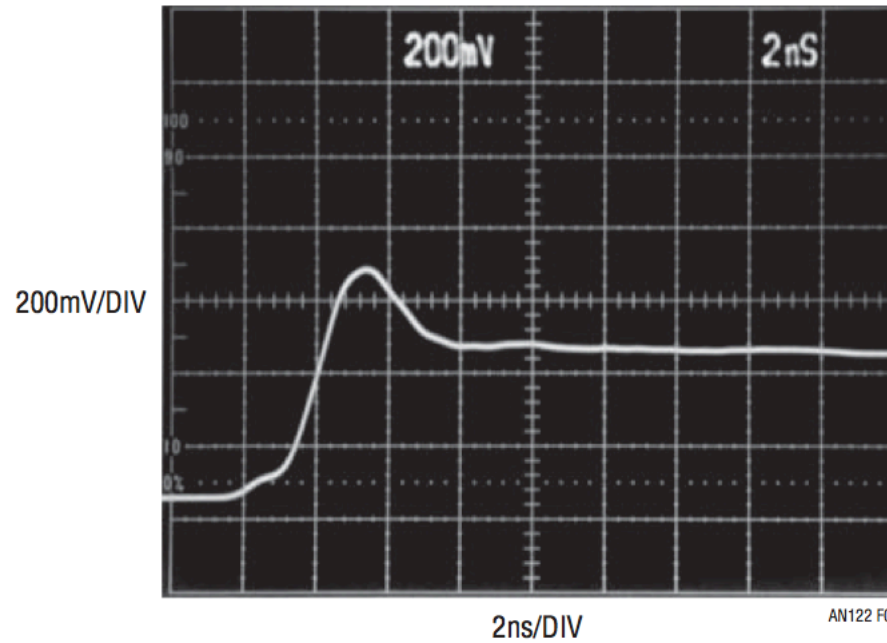
Diode Reverse Recovery



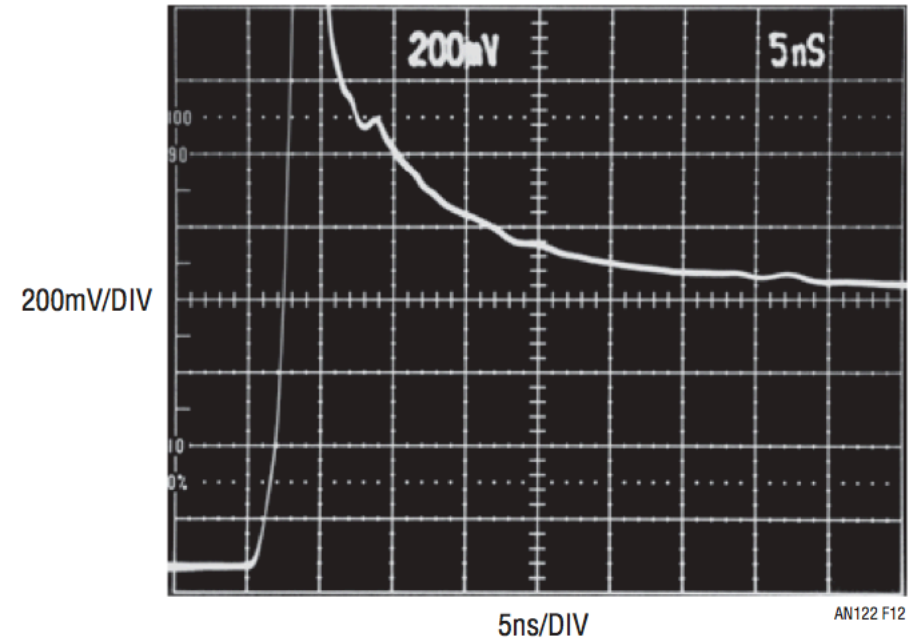
Diode Forward Recovery



Diode Forward Recovery

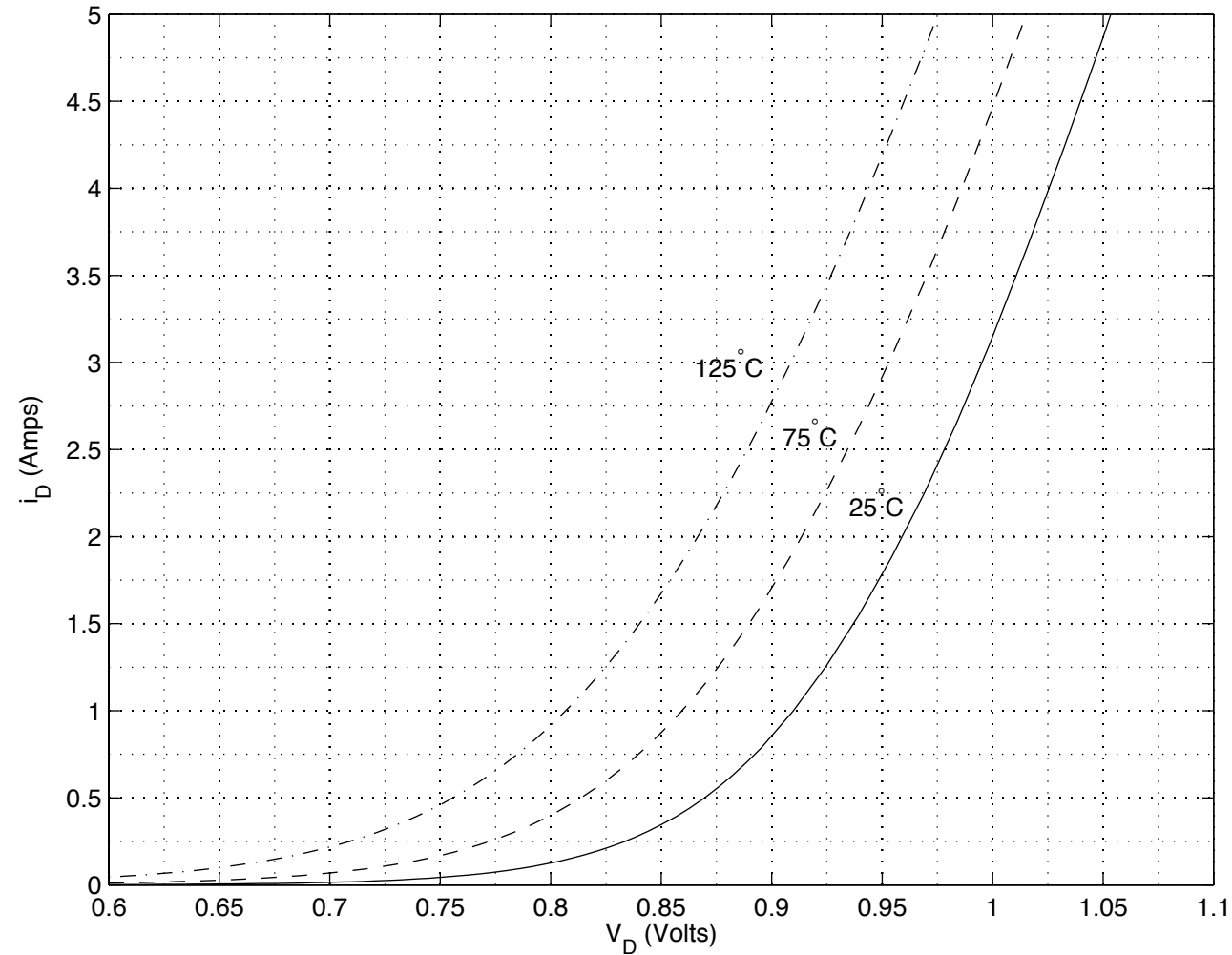


Good Diode



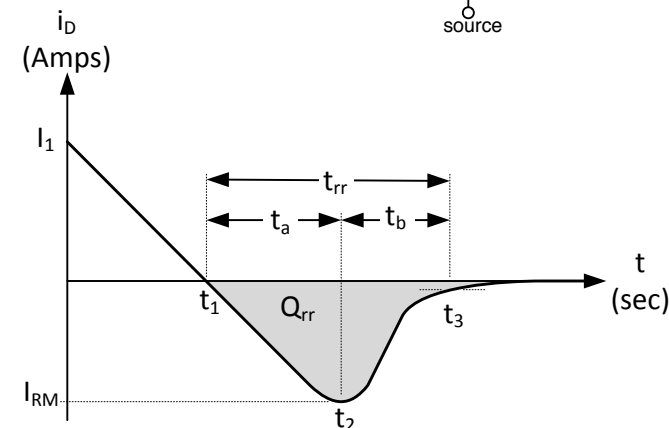
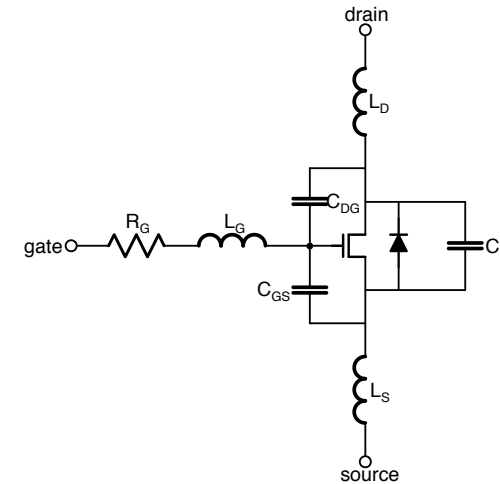
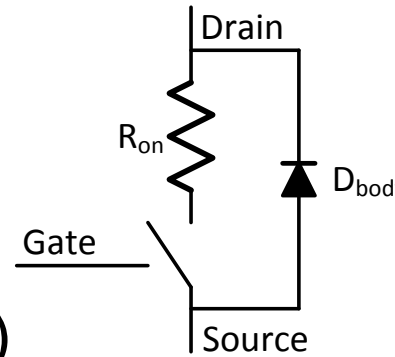
Bad Diode

Two 2A Diodes < One 4A Diode

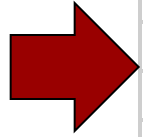


Summary – Power Devices

- Finite, non-zero
 - Switching time
 - Blocking voltage
 - On-voltage (resistance)
- Parasitics – L and C
- MOSFETs – switches
 - Turn on/off as fast as gate can be charged
 - $R = kV^2$
- Diodes
 - Self-controlled switches
 - Reverse recovery loss Q_{RR}



Course at a Glance



No	Date	Topic	HW out	HW in	Lab out	Lab ck	Lab	HW
1	9/25/17	Intro (basic converters)	1		1		Intro to ST32F3	Periodic Steady State
2	9/27/17	Embedded Prog/Power Elect.						
3	10/2/17	Power Electronics - 1 (switches)	2	1	2	1	AC Energy Meter	Power Devices
4	10/4/17	Power Electronics - 2 (circuits)						
5	10/9/17	Photovoltaics	3	2	3	2	PV MPPT	Motor control Matlab
6	10/11/17	Feedback Control						
7	10/16/17	Electric Motors	4	3	4	3	Motor control - Lab/	Feedback
8	10/18/17	Isolated Converters						
9	10/23/17	Solar Day	5/PP	4	5	4	PS	Isolated Converters
10	10/25/17	Magnetics						
11	10/30/17	Soft Switching	6	5/PP	6	5	Magnetics	Magnetics and Inverters
12	11/1/17	Project Discussions						
13	11/6/17	Inverters, Grid, PF, and Batteries		6	P	6	Project	
14	11/8/17	Thermal & EMI						
15	11/13/17	Quiz Review				C1		
16	11/15/17	Grounding, and Debugging						
Q	11/15/17	Quiz - in the evening						
	11/20/17	Thanksgiving Break				C2		
	11/22/17	Thanksgiving Break						
17	11/27/17	Wrapup						
18	11/29/17	Guest Lecture				C3		
19	12/4/17	Guest Lecture						
20	12/6/17	No Class						
	TBD	Project presentations			P			
	12/15/17	Project webpage due						