EECS192 Lecture 3
Jan. 30, 2018

Notes:

- Handouts: lab rules
- 2/2 project proposal – upload to bcourses by Fri 5 pm
- 2/10: Motor drive/stall, steering servo
- Quiz 1: motor behavior Tues 2/6 (See Motor Worksheet week 2)
- CalDay Sat April 4/21

Topics

- MCUXpresso hint: timing using systime or PIT cval0
- Motor electrical model conclusion
  - Motor electromechanical behavior
- Driving MOSFETs and motor
  - Op amp TLC084
  - MC33883 MOSFET driver
  - H bridge
- PWM issues for motor
- (Back EMF measurement)

Timing from MCUXpresso Tools

- Global Systime (every 100 us)
- Periodic Interrupt Timer PIT-CVAL0 (count down)
- internal module clock frequency (60 MHz)

Step over

MCUXpressoProj - Develop - frdmk64f_skeletonrtos/source/task1.c - MCUXpresso IDE

File Edit Source Refactor Navigate Search Project Run FreeRTOS Window Help

Quick Access

Qui... Glo... Var... Bre... Out...

Variable	Type	Value
log_queue	QueueHandle_t	0x200011c8
sqrt_array	float [1000]	0x20000094
systime	volatile uint32_t	30076
ulIdleCycleCount	uint32_t	95938
xTickCount	volatile TickType_t	3000
ZSum	double	95789.89952

Debug

write_task_10 at task1.c:75 0x15ea

startup_mk6... main.c idletask.c task1.c

```
70      tick_start = xTaskGetTickCount();
71      for (i = 0; i < 9; i++)
72      {
73          systime_start = systime;
74          for (j=0; j < 1000; j++)
75              sqrt_array[j]= sqrtf((float)j);  // sqrt ~ 100 u
76          sprintf(log, "Tsk1#%d, 1000 sqrt time (us) %d\n\r",
77                  (int)i, 100*(long)(systime-systime_start));
```

Monitors

PIT [MK64FN1M0x

PIT: 0x40037000 [MK64FN1M0xxx12] New Renderings...

Register	Address	Value
PIT-CVAL0	0x40037104	0x16d3
TVL	[31:0]	0x16d3
PIT-CVAL1	0x40037114	0x0
PIT-CVAL2	0x40037124	0x0
PIT-CVAL3	0x40037134	0x0

NXP MK64FN1M0xxx12 (frdmk64f...etonrtos)

Timing from MCUXpresso Tools

~38 clocks for sqrt (hardware floating point) ~0.6 us
~1689 clocks for cos() ~28 us

MCUXpressoProj - Develop - frdmk64f_skeletonrtos/source/task1.c - MCUXpresso IDE

File Edit Source Refactor Navigate Search Project Run FreeRTOS Window Help

Quick Access

Qui... Glo... Var... Bre... Out...

Name	Type	Value
pvParameters	void *	<optimized out>
log	char [50]	0x20002374 <uch
tick_start	TickType_t	0
tick_end	TickType_t	<optimized out>
systime_start	uint32_t	30075
i	int	3
j	int	6

```
Name : log
    Details:"Tsk1#2, 1000 sqrt time (us) 
    Default:0x20002374 <ucHeap+4536>
    Decimal:536879988
    Hex:0x20002374
    Binary:1000000000000000100011011110100
```

Debug

write_task_10 at task1.c:74 0x15ee

startup_mk6... main.c idletask.c task1.c

```
70      tick_start = xTaskGetTickCount();
71      for (i = 0; i < 9; i++)
72      {
73          systime_start = systime;
74          for (j=0; j < 1000; j++)
75              sqrt_array[j]= sqrtf((float)j);  // sqrt ~ 100 u
76          sprintf(log, "Tsk1#%d, 1000 sqrt time (us) %d\n\r",
77              (int)i, 100*(long)(systime-systime_start));
```

Monitors

PIT [MK64FN1M0x

PIT: 0x40037000 [MK64FN1M0xxx12] New Renderings...

Register	Address	Value
PIT-CVAL0	0x40037104	0x16d0
TVL	[31:0]	0x16d0
PIT-CVAL1	0x40037114	0x0
PIT-CVAL2	0x40037124	0x0
PIT-CVAL3	0x40037134	0x0

NXP MK64FN1M0xxx12 (frdmk64f...etonrtos)

Timing from MCUXpresso Tools

```
/* Set timer period for channel 0 */
PIT_SetTimerPeriod(PIT, kPIT_Chnl_0, USEC_TO_COUNT(100U, PIT_SOURCE_CLOCK));
// 100 us timing
    /* Enable timer interrupts for channel 0 */
    PIT_EnableInterrupts(PIT, kPIT_Chnl_0, kPIT_TimerInterruptEnable);
    /* Enable at the NVIC */
    EnableIRQ(PIT_IRQ_ID);
```

```
void PIT0_IRQHandler(void)
{
systime++; /* hopefully atomic operation */
    PIT_ClearStatusFlags(PIT, kPIT_Chnl_0, kPIT_TimerFlag);
/* Clear interrupt flag.*/
    pitIsrFlag = true;
}
```

Clock Choice for K64 CPU

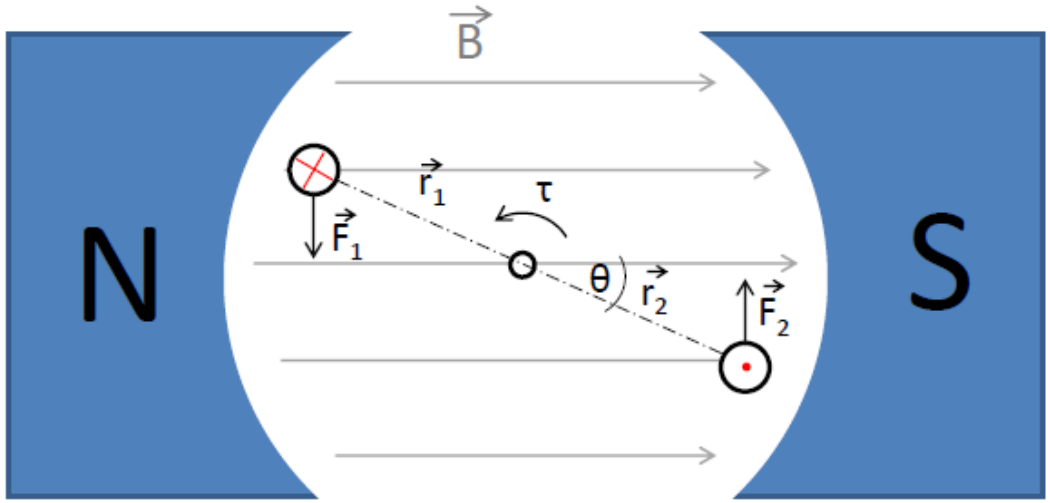
- CPU and system clocks = 100 MHz
- Bus clock = 50 MHz
- FlexBus clock = 50 MHz
- Flash clock = 25 MHz

Example to choose clock frequency

```
#FTM_SOURCE_CLOCK CLOCK_GetFreq(kCLOCK_BusClk)
```

```
kCLOCK_CoreSysClk,     /*!< Core/system clock  */
kCLOCK_PlatClk,        /*!< Platform clock       */
kCLOCK_BusClk,         /*!< Bus clock          */
kCLOCK_FlexBusClk,     /*!< FlexBus clock      */
kCLOCK_FlashClk,       /*!< Flash clock      */
```

DC Motor Physical Model-review

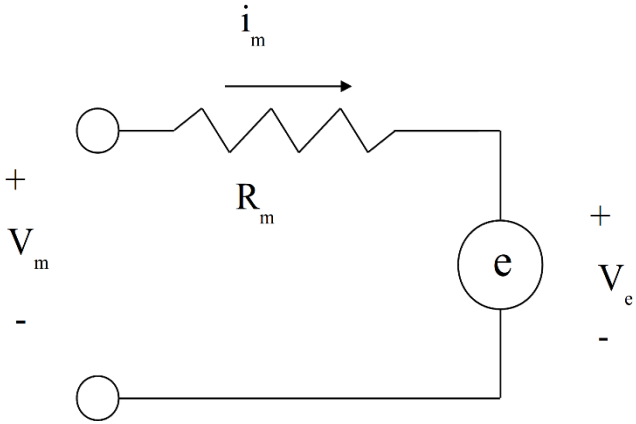


$$\vec{F} = i\vec{l} \times \vec{B}$$

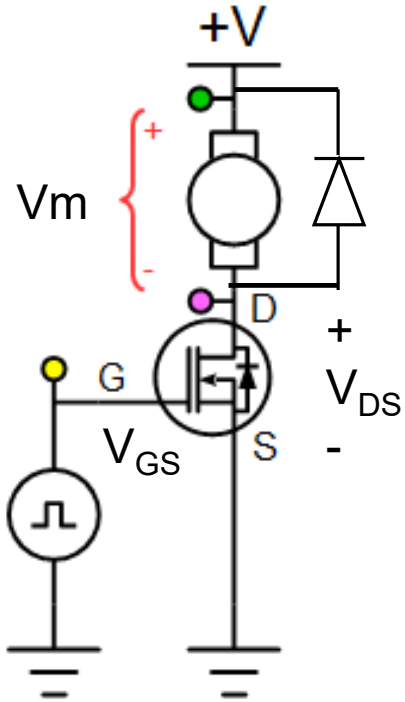
$$\tau = \vec{r}_1 \times \vec{F}_1 + \vec{r}_2 \times \vec{F}_2$$

Motor Electrical Model

Motor Electrical Model
Back EMF
Motor electromechanical behavior



Continued on board
Also- see motor worksheet……

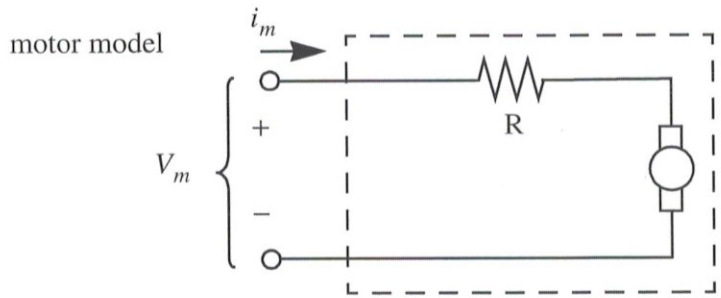


Note: missing e-stop!

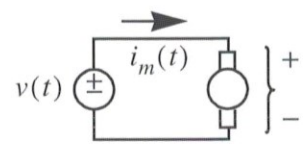
Motor model

For this problem, consider a DC permanent magnet motor (as used in your car). The car is on a carpet and moves in a straight line with no slip between the wheels and the carpet. The car is initially moving at a speed of 2 meters per second.

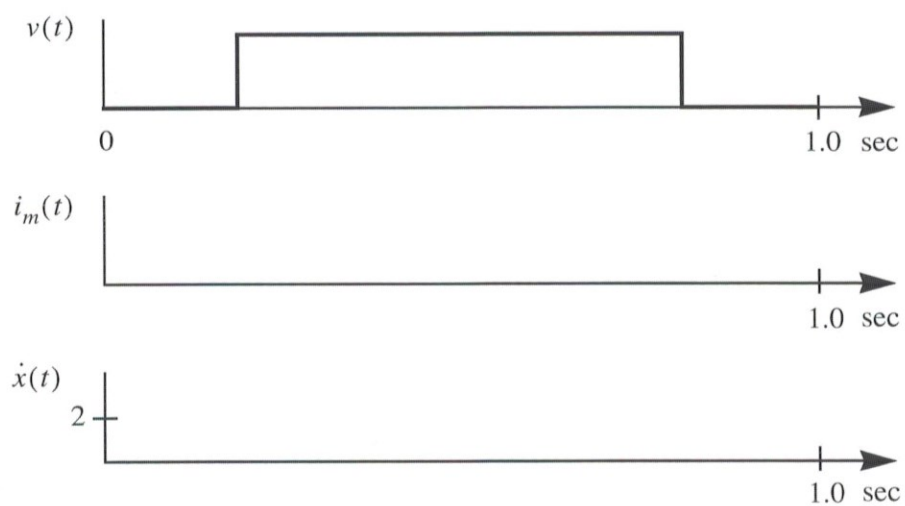
You can assume a motor model as shown below. The qualitative shape of the curves is more important than magnitudes.



[4 pts.] **a)** Consider the motor driven from a voltage source with voltage $v(t)$, as shown. Sketch car velocity $\dot{x}(t)$ and motor terminal current for the time indicated.



Let peak speed = 5 m/sec
Accel = 5 m/s^2
k_e = 1 v/(m/sec)
On board



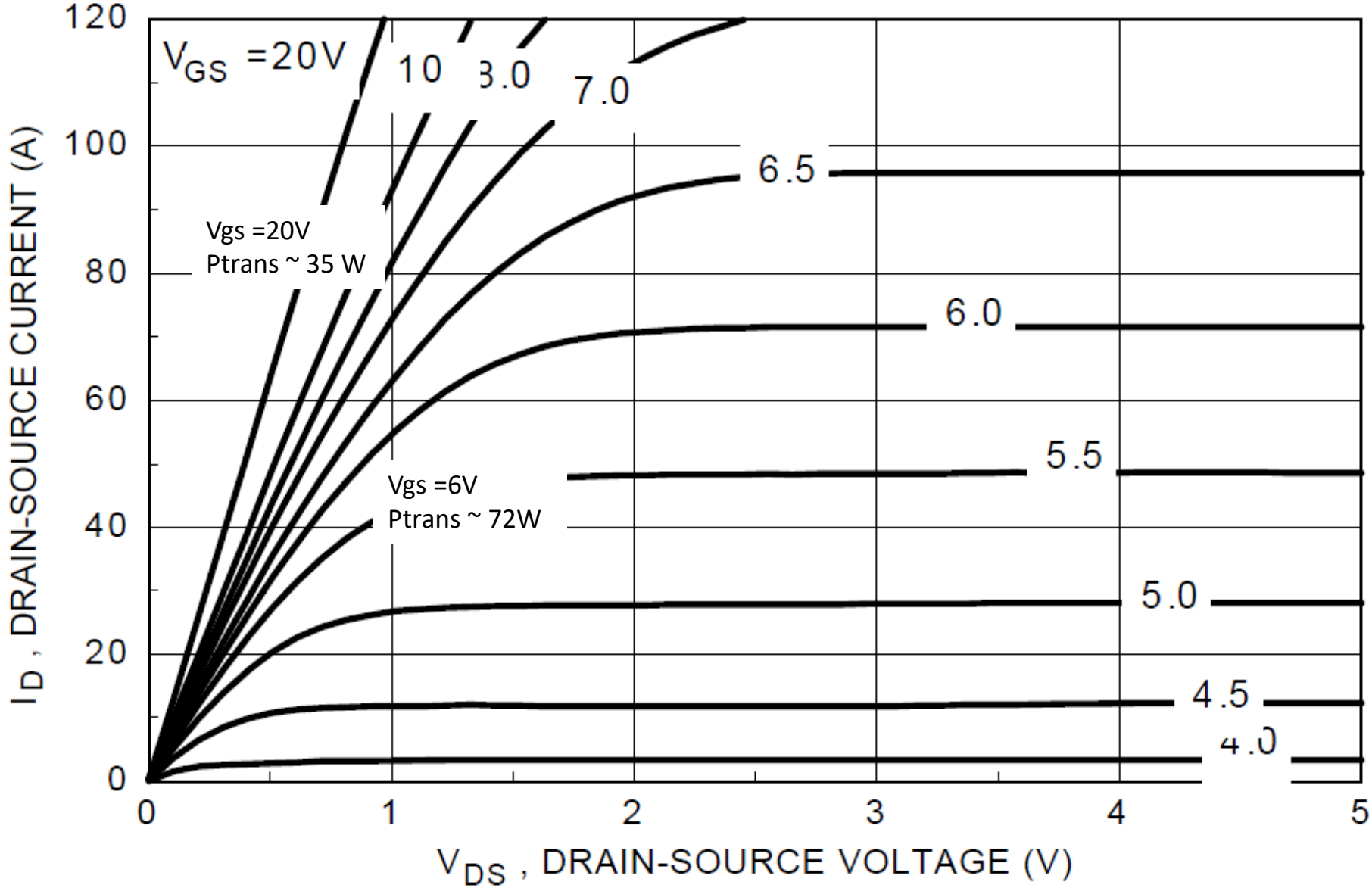
(for answer
see sp99 final solution)

MOSFETs and motor drive

Given: Rm = 0.1 ohms, Vbatt = 7.2 V, Rbat = 0.

Vds = ? ➔ Ids = ? amps

(LiPo 11 V!)



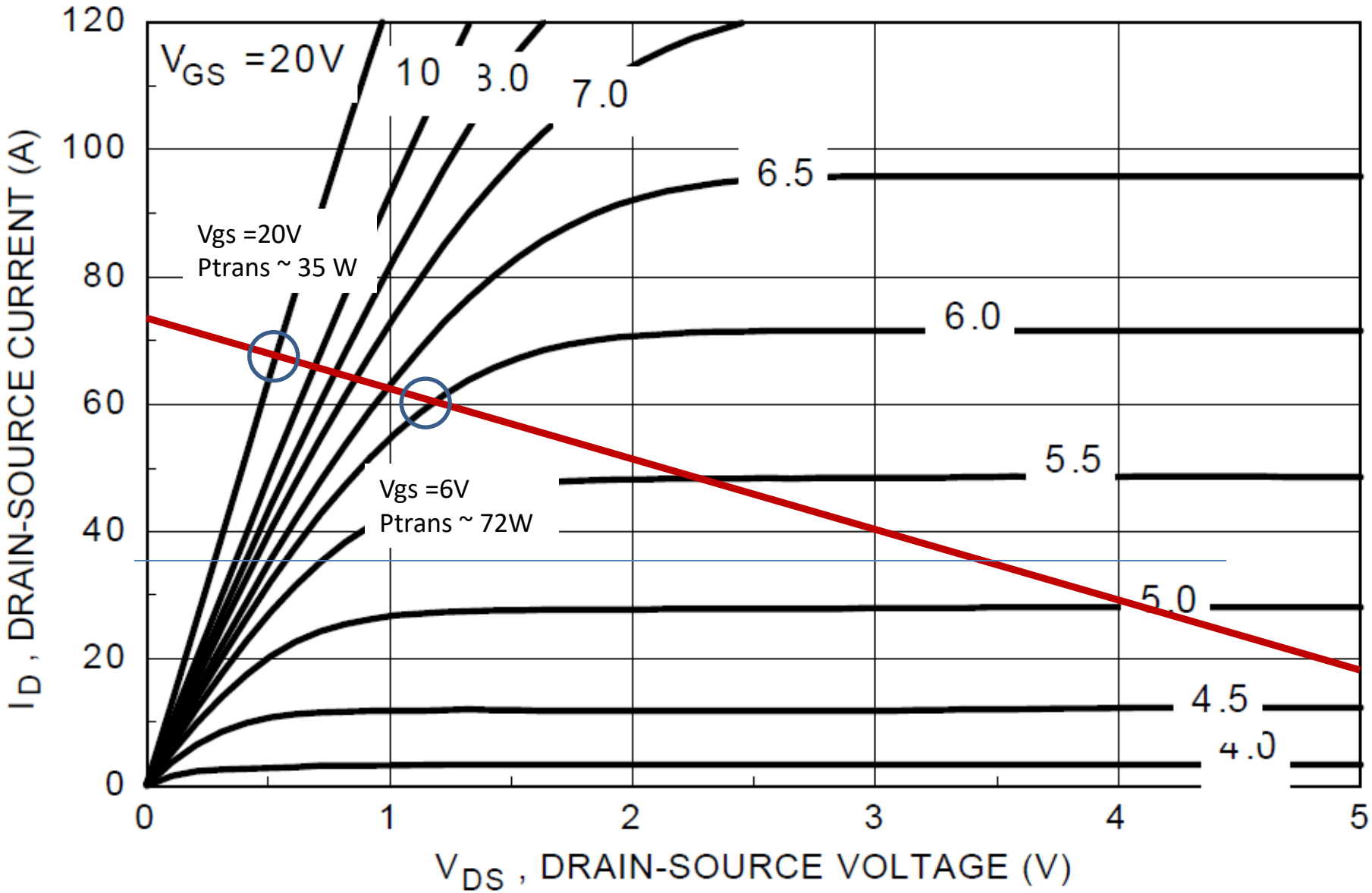
Driving MOSFETs and motor

Rm = 0.1 ohms, Vbatt = 7.2 V, Rbat = 0.

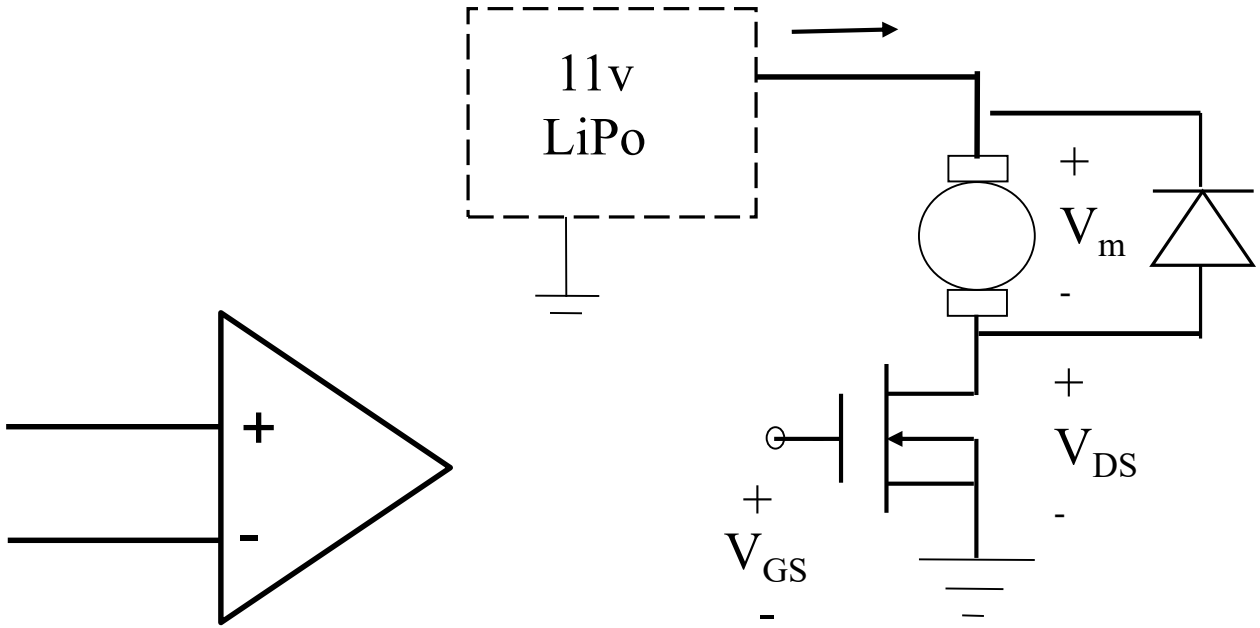
Vds = 3.6V ➔ Ids = (7.2-3.6V)/(0.1 ohm) = 36 amps

Key design points:

1) High Vgs better than low Vgs
2) Switch quickly
3) Make sure Vs=0 (big ground)



Motor Drive Quick and Dirty w/ Op Amp

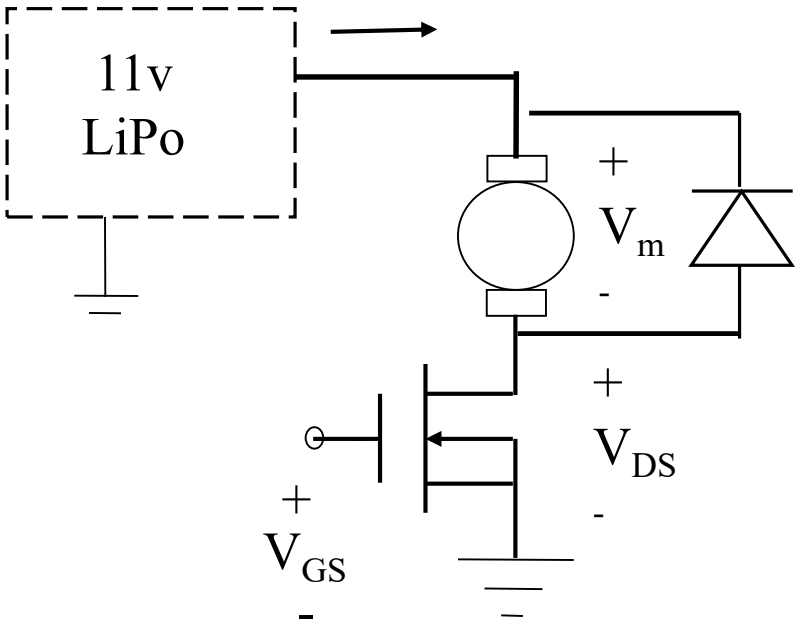


Details On board….

TLC084 or TLC074 quad op amp

Checklist:
1) Emergency stop
2) Reset Protection
3) Snubbing

Motor Drive Quick and Dirty w/ Op Amp



- IRFB3006: C_{GS}= 9000 pF
- TL084: 50 mA
- i = C dV/dt
- dV/dt = i/C ~ 5 V/usec ➔ 2us on + 2 us off, PWM < 100 us period
- TL084 slew rate: 10 V/usec ➔ exceed current limit
- Series resistor on output of op amp of 200 ohms, limits current to 50 mA (10V/200 ohms = 50 mA)

Driving MOSFETs and motor

How to choose PWM frequency?

IRFB 3006 Power MOSFET

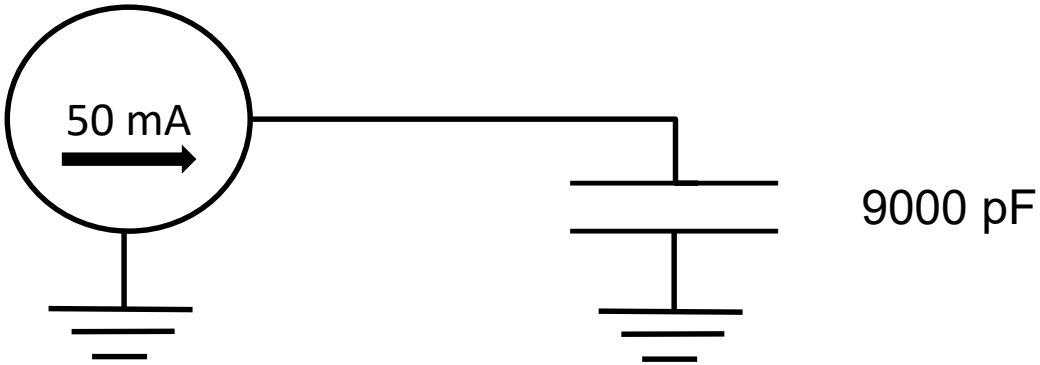
Symbol	Parameter	Min.	Typ.	Max.	Units	Conditions
gfs	Forward Transconductance	280	—	—	S	V_{DS} = 25V, I_D = 170A
Q_g	Total Gate Charge	—	200	300	nC	I_D = 170A
Q_{gs}	Gate-to-Source Charge	—	37	—		V_{DS} =30V
Q_{gd}	Gate-to-Drain ("Miller") Charge	—	60			V_{GS} = 10V ⑤
Q_{sync}	Total Gate Charge Sync. (Q_g - Q_{gd})	—	140	—		I_D = 170A, V_{DS} =0V, V_{GS} = 10V
$t_{d(on)}$	Turn-On Delay Time	—	16	—	ns	V_{DD} = 39V
t_r	Rise Time	—	182	—		I_D = 170A
$t_{d(off)}$	Turn-Off Delay Time	—	118	—		R_G = 2.7Ω
t_f	Fall Time	—	189	—		V_{GS} = 10V ⑤
C_{iss}	Input Capacitance	—	8970	—	pF	V_{GS} = 0V

TLC084 Op amp (same pinout as TL974)

V_{OH}	High-level output voltage	V_{IC} = 2.5 V	I_{OH} = –1 mA	25°C	4.1	4.3		V
				Full range	3.9			
			I_{OH} = –20 mA	25°C	3.7	4		
				Full range	3.5			
			I_{OH} = –35 mA	25°C	3.4	3.8		
				Full range	3.2			
			I_{OH} = –50 mA	25°C	3.2	3.6		
				–40°C to 85°C	3			
V_{OL}	Low-level output voltage	V_{IC} = 2.5 V	I_{OL} = 1 mA	25°C		0.18	0.25	V
				Full range			0.35	
			I_{OL} = 20 mA	25°C		0.35	0.39	
				Full range			0.45	
			I_{OL} = 35 mA	25°C		0.43	0.55	
				Full range			0.7	
			I_{OL} = 50 mA	25°C		0.45	0.63	
				–40°C to 85°C			0.7	
I_{OS}	Short-circuit output current	Sourcing		25°C		100		mA
		Sinking		25°C		100		
I_O	Output current	V_{OH} = 1.5 V from positive rail		25°C		57		mA
		V_{OL} = 0.5 V from negative rail		25°C		55		

How to choose PWM frequency (if using op amp)?

(on board)



Driving MOSFETs and motor

H-Bridge Gate Driver IC

The 33883 is an H-bridge gate driver (also known as a full-bridge pre-driver) IC with integrated charge pump and independent high and low side gate driver channels. The gate driver channels are independently controlled by four separate input pins, thus allowing the device to be optionally configured as two independent high side gate drivers and two independent low side gate drivers. The low side channels are referenced to ground. The high side channels are floating.

The gate driver outputs can source and sink up to 1.0 A peak current pulses, permitting large gate-charge MOSFETs to be driven and/or high pulse- width modulation (PWM) frequencies to be utilized. A linear regulator is incorporated, providing a 15 V typical gate supply to the low side gate drivers.

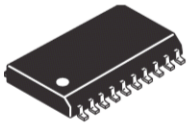
This device powered by SMARTMOS technology.

Features

- V_{CC} operating voltage range from 5.5 V up to 55 V
- V_{CC2} operating voltage range from 5.5 V up to 28 V
- CMOS/LSTTL compatible I/O
- 1.0 A peak gate driver current
- Built-in high side charge pump
- Under-voltage lockout (UVLO)
- Over-voltage lockout (OVLO)
- Global enable with <10 μA Sleep mode
- Supports PWM up to 100 kHz

33883

H-BRIDGE GATE DRIVER IC



EG SUFFIX (PB-FREE)
98ASB42343B
20-PIN SOICW

ORDERING INFORMATION		
Device (Add R2 Suffix for Tape and Reel)	**Temperature Range (T_A)**	**Package**
MC33883HEG	-40 °C to 125 °C	20 SOICW

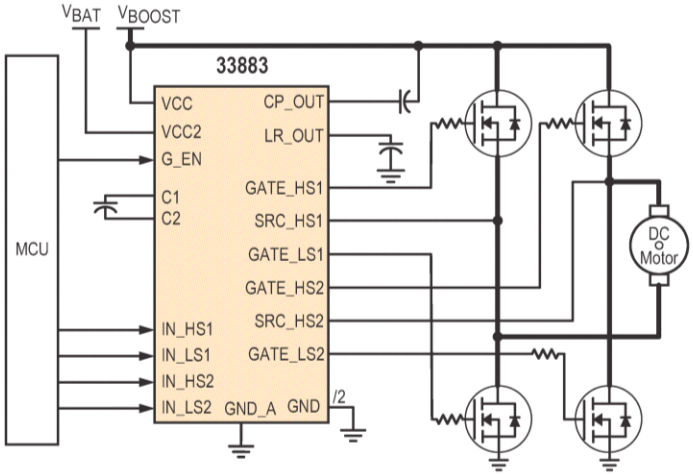


Figure 1. 33883 Simplified Application Diagram

MC3383 + H bridge

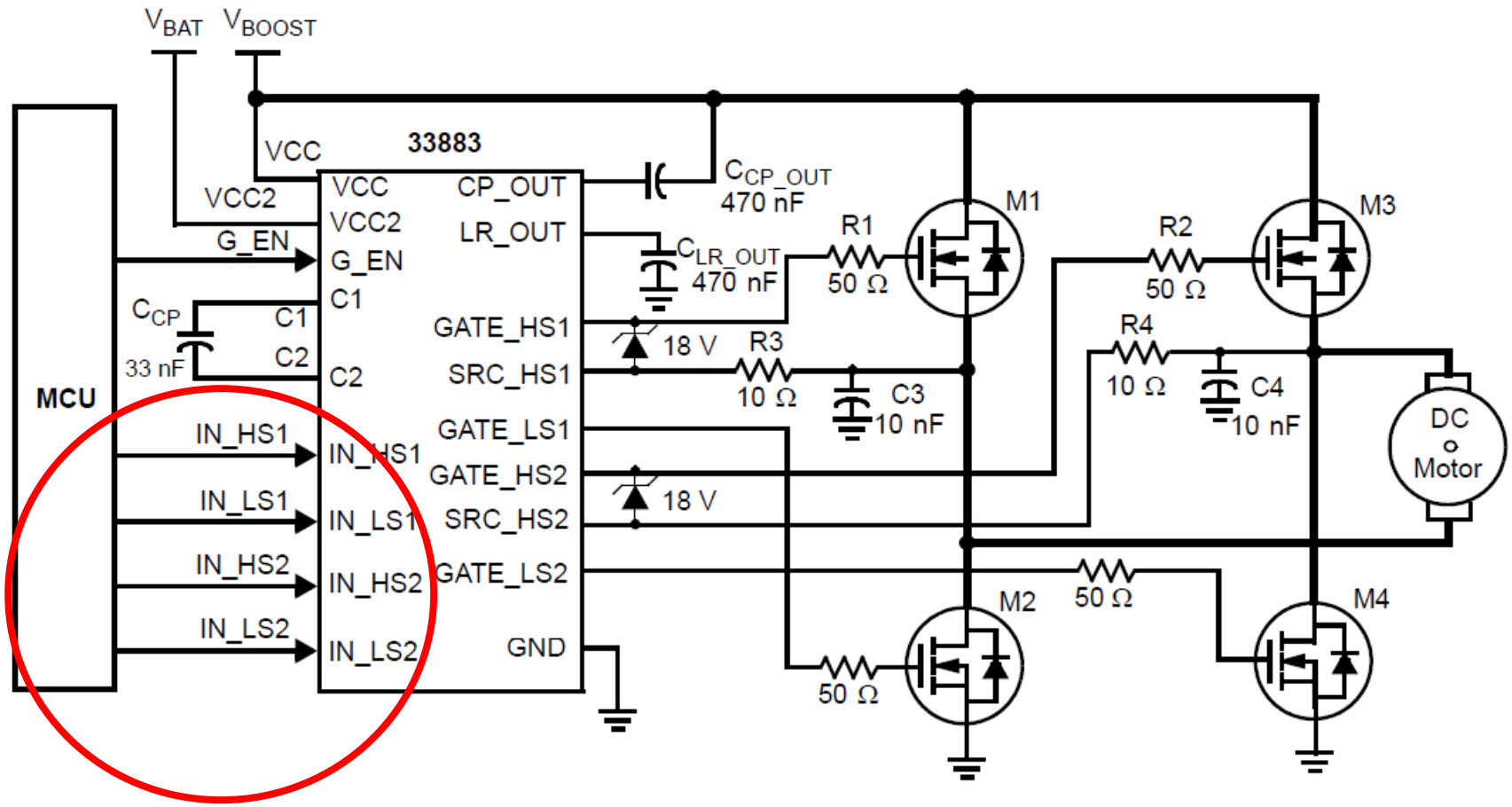


Figure 14. Application Schematic with External Protection Circuit

!!!!CAUTION!!!!
Software fries hardware….
Need protection logic- 74HCxxx
Estop?

MC3383 protection

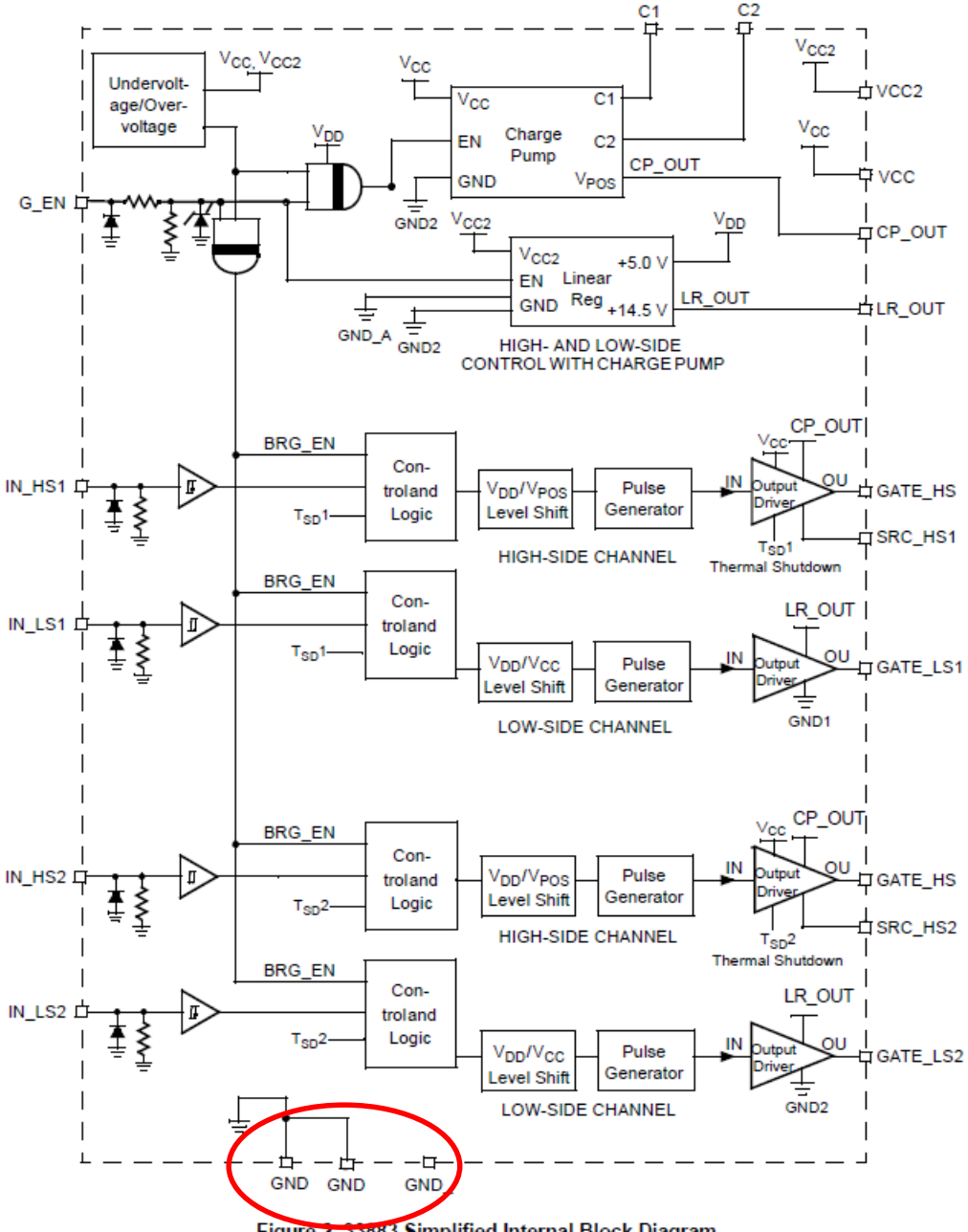


Figure 2. 33883 Simplified Internal Block Diagram

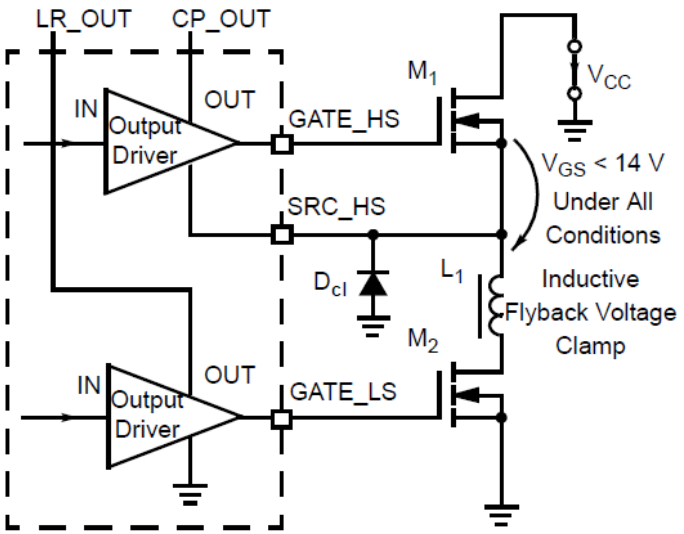


Figure 12. Gate Protection and Flyback Voltage Clamp

Caution: don't run motor current through here

PWM Skeleton/Example

https://github.com/ucb-ee192/SkeletonMCUX/tree/master/frdmk64f_pwm

```
FTM_UpdatePwmDutycycle
(BOARD_FTM_BASEADDR, BOARD_FTM_CHANNEL,
        kFTM_CenterAlignedPwm, updated_duty_cycle);
```

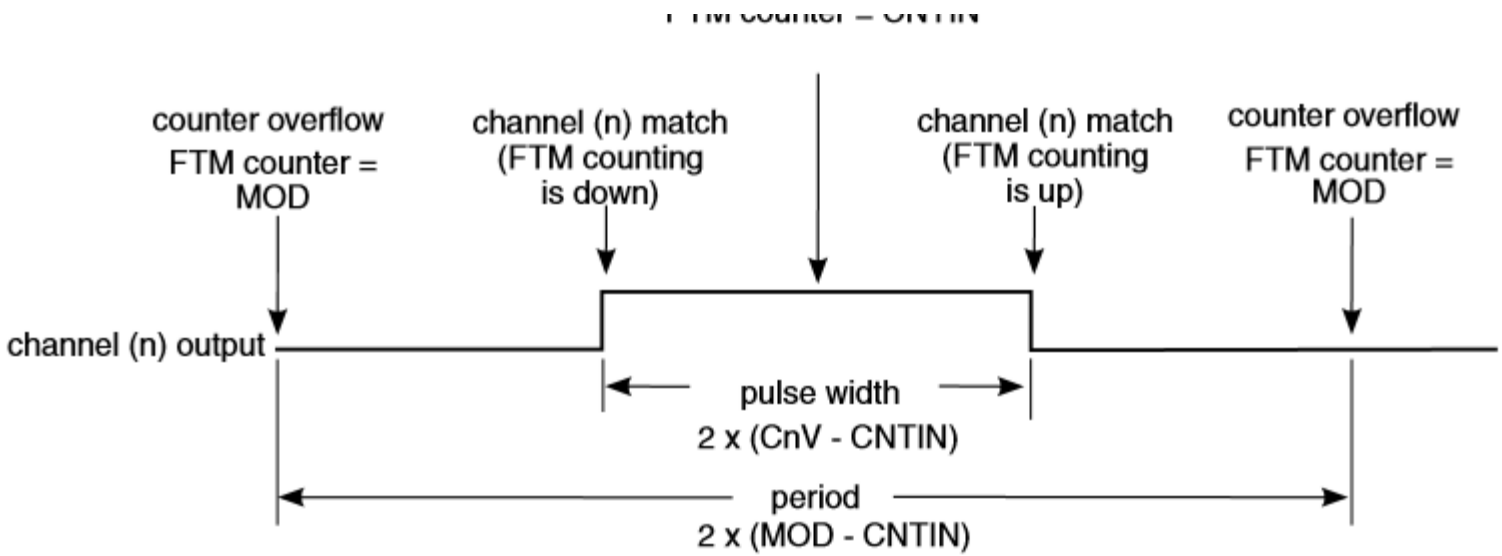
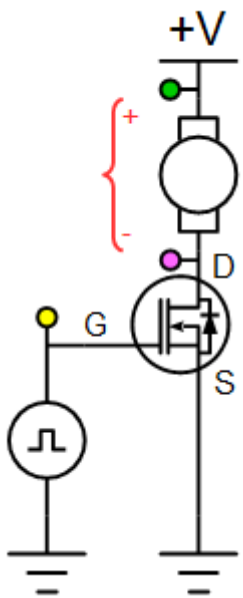
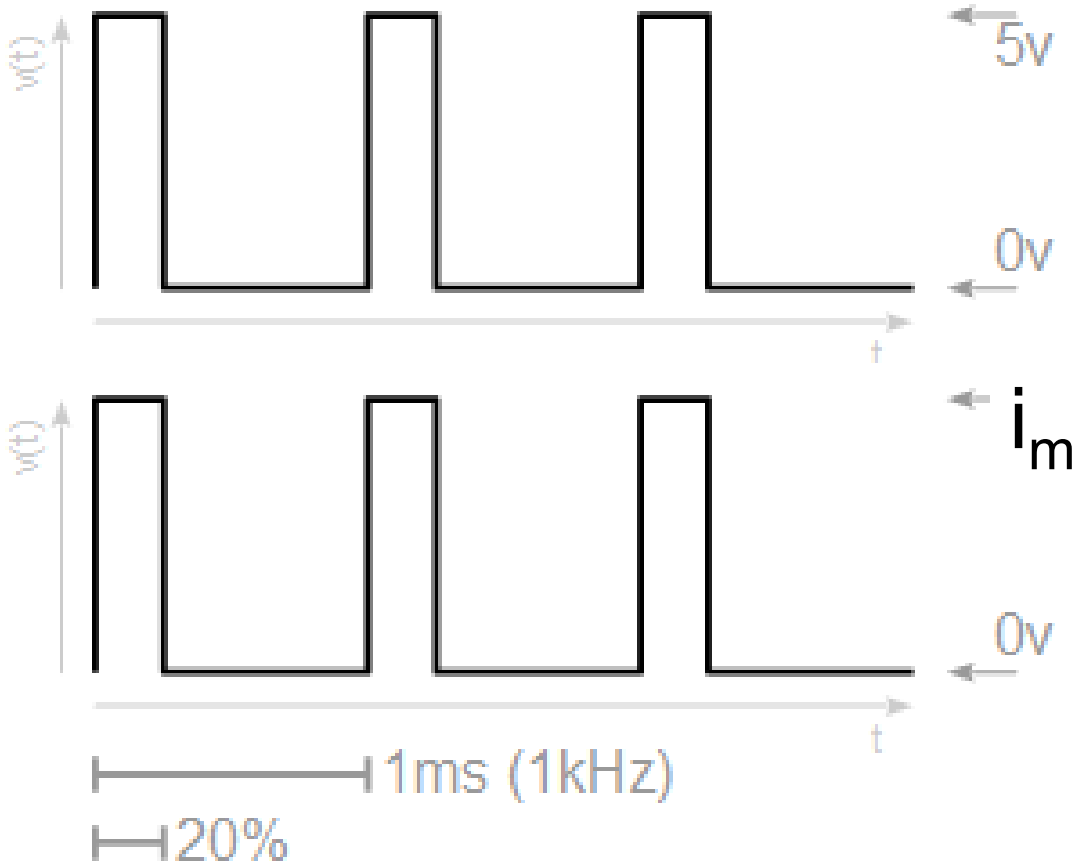


Figure 40-21. CPWM period and pulse width with ELSnB:ELSnA = 1:0

PWM for Main Motor control

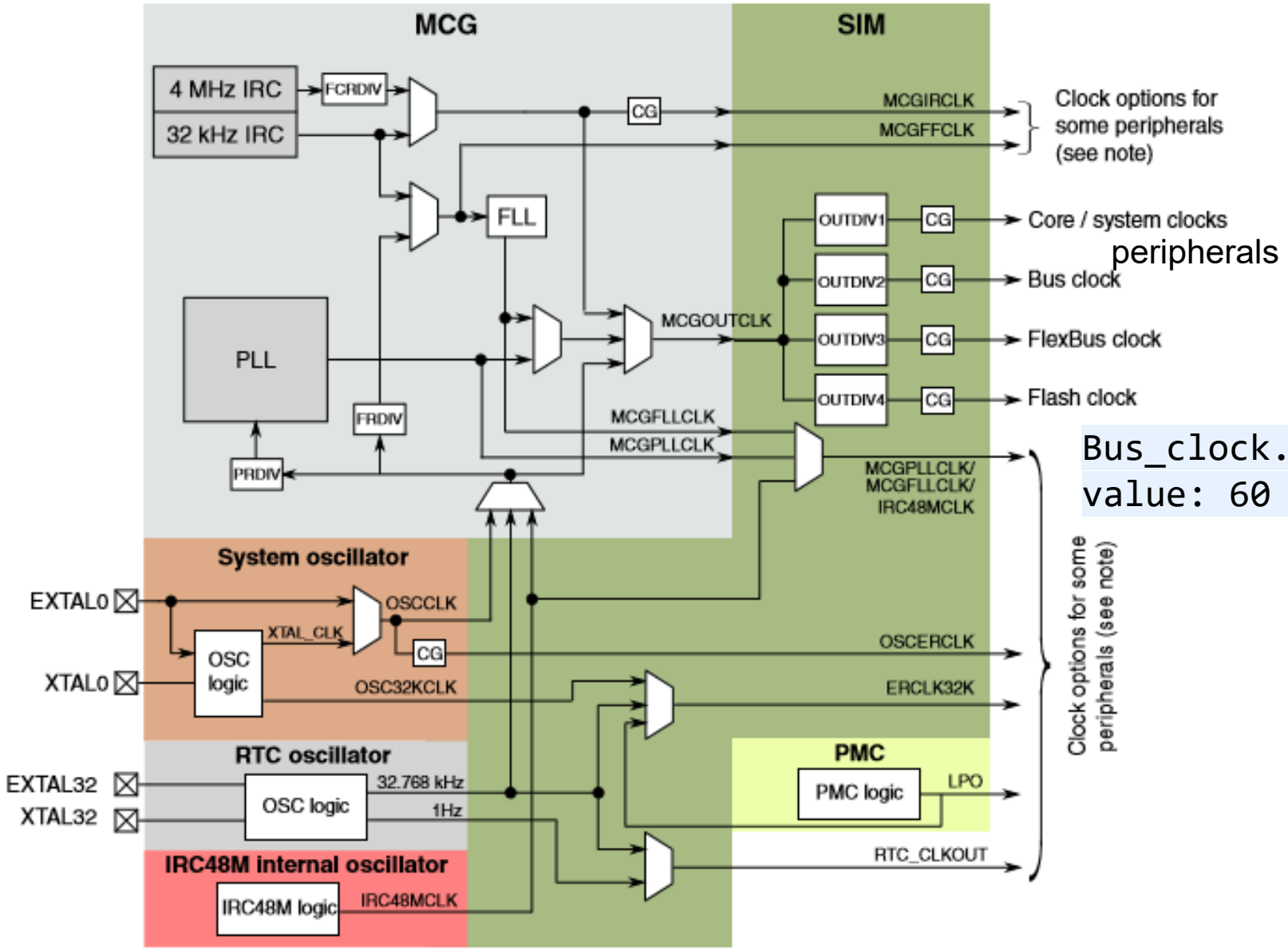




$$\langle i_m \rangle = (T/T_o)\, i_{max}$$

Is i_{max} constant?

Extra Slides



Bus_clock.outFreq, value: 60 MHz}

CG — Clock gate
Note: See subsequent sections for details on where these clocks are used.

Figure 5-1. Clocking diagram