Announcements

• Upcoming Due Dates
  – Experiment 1 Report: R 1/24 (before class)
  – Experiment 2 Report: T 1/29 (end of class)
  – Experiment 3 Prelab: R 1/31 (before class)

Comments on Motor Control

EXPERIMENT 1
Motor Driver: Trapezoidal Control

Trapezoidal Control

![Graphs showing speed, input current, phase current, and torque](Image credit: Microchip Technology Inc.)
Sinusoidal Control

Microcontroller Programming

EXPERIMENT 2
Experiment 2

- Experiment 3 will build synchronous boost converter
- To operate open loop, need gate drive signals
- Experiment 2: brief introduction to MSP programming – Generate voltage-controlled PWM signals

Microprocessor: MSP430 Launchpad

- MSP430 microprocessors from Texas Instruments – MSP430G2553
- Programmable in C or ASM
- Ultra-low power (not a focus here)

- On-board USB bootloader
- Two LEDs, one switch
- Two timers, one 5-channel 10-bit ADC
- System clock up to 16 MHz
High Resolution PWM

MSP430G2553:
• 16 MHz clock
  – Max PWM resolution is 62.5ns
MSP430F5172:
• PWM 16x clock multiplier
  – Max PWM resolution is 4ns
• Final decision TBD; same programming approach applies in either case.

Notes on Launchpad

• P1.1 and P1.2 are used as part of the digital communication for the debugger
• P1.0, P1.6, P2.1, P2.3, P2.5 can be tied to on-board LEDs for visual debugging
• Do not apply power to Vcc; it is generated on-board
• Launchpad does not break out all pins on MSP
  – User guide lists all functionality in family
  – Make sure to take note of what your chip can do
• Documentation contains both assembly and C code
MSP430 Documentation

• User’s Guide

• Datasheet

• Errata
  – http://www.ti.com/lit/er/slaz440g/slaz440g.pdf

Example Today

• General Purpose I/O
• System Clock
• TimerA
• Interrupts
Pin Assignments

MSP430 Internal Block Diagram

Functional Block Diagram, MSP430G2x53

NOTE: ADC10 is available on MSP430G2x53 devices only.
NOTE: The pulldown resistors of port P3 should be enabled by setting P3REN.x = 1.
Digital I/O Registers

8.2.1 Input Register PxIN
Each bit in each PxIN register reflects the value of the input signal at the corresponding I/O pin when the pin is configured as I/O function.
- Bit = 0: The input is low
- Bit = 1: The input is high

8.2.2 Output Registers PxOUT
Each bit in each PxOUT register is the value to be output on the corresponding I/O pin when the pin is configured as I/O function, output direction, and the pullup/pulldown resistor is disabled.
- Bit = 0: The output is low
- Bit = 1: The output is high
If the pin's pullup/pulldown resistor is enabled, the corresponding bit in the PxOUT register selects pullup or pulldown.
- Bit = 0: The pin is pulled down
- Bit = 1: The pin is pulled up

8.2.3 Direction Registers PxDIR
Each bit in each PxDIR register selects the direction of the corresponding I/O pin, regardless of the selected function for the pin. PxDIR bits for I/O pins that are selected for other functions must be set as required by the other function.
- Bit = 0: The port pin is switched to input direction
- Bit = 1: The port pin is switched to output direction

8.2.4 Pullup/Pulldown Resistor Enable Registers PxREN
Each bit in each PxREN register enables or disables the pullup/pulldown resistor of the corresponding I/O pin. The corresponding bit in the PxOUT register selects if the pin is pulled up or pulled down.
- Bit = 0: Pullup/pulldown resistor disabled
- Bit = 1: Pullup/pulldown resistor enabled

8.2.5 Function Select Registers PxSEL and PxSEL2
Port pins are often multiplexed with other peripheral module functions. See the device-specific data sheet to determine pin functions. Each PxSEL and PxSEL2 bit is used to select the pin function - I/O port or peripheral module function.

Clock Module

Figure 5-1. Basic Clock Module+ Block Diagram - MSP430F2xx
DCO Frequency Select

5.2.5.2 Adjusting the DCO Frequency

After a PUC, RSElx = 7 and DCOx = 3, allowing the DCO to start at a mid-range frequency. MCLK and SMCLK are sourced from DCOCCLK. Because the CPU executes code from MCLK, which is sourced from the fast-starting DCO, code execution typically begins from PUC in less than 2 μs. The typical DCOx and RSElx ranges and steps are shown in Figure 5-6.

The frequency of DCOCCLK is set by the following functions:

- The four RSElx bits select one of sixteen nominal frequency ranges for the DCO. These ranges are defined for an individual device in the device-specific data sheet.
- The three DCOx bits divide the DCO range selected by the RSElx bits into 8 frequency steps, separated by approximately 10%
- The five MODx bits, switch between the frequency selected by the DCOx bits and the next higher frequency set by DCOx+1. When DCOx = 0, the MODx bits have no effect because the DCO is already at the highest setting for the selected RSElx range.

![Figure 5-6. Typical DCOx Range and RSElx Steps](image)

Each MSP430F2xx device (and most MSP430G2xx devices; see device-specific data sheets) has calibrated DCOCCTL and BCSCCTL1 register settings for specific frequencies stored in information memory segment A. To use the calibrated settings, the information is copied into the DCOCCTL and BCSCCTL1 registers. The calibrated settings affect the DCOx, MODx, and RSElx bits, and clear all other bits, except XT2OFF which remains set. The remaining bits of BCSCCTL1 can be set or cleared as needed with BIS.B or BIC.B instructions.

Clock Registers (1/2)

5.3.1 DCOCCTL, DCO Control Register

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCOx</td>
<td>nw-0</td>
<td>nw-1</td>
<td>nw-1</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
</tr>
<tr>
<td>MODx</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
</tr>
</tbody>
</table>

DCOx
- Bits 7-5: DCO frequency select. These bits select which of the eight discrete DCO frequencies within the range defined by the RSElx setting is selected.

MODx
- Bits 4-0: Modulator selection. These bits define how often the f_{dcoc} frequency is used within a period of 32 DCOCCLK cycles. During the remaining clock cycles (32-MOD) the f_{dcoc} frequency is used. Not useable when DCOx = 7.

5.3.2 BCSCCTL1, Basic Clock System Control Register 1

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>XT2OFF</td>
<td>nw-(1)</td>
<td>nw-(0)</td>
<td>nw-(0)</td>
<td>nw-(0)</td>
<td>nw-0</td>
<td>nw-1</td>
<td>nw-1</td>
<td>nw-1</td>
</tr>
<tr>
<td>XTS</td>
<td>nw-(0)</td>
<td>nw-(0)</td>
<td>nw-(0)</td>
<td>nw-0</td>
<td>nw-1</td>
<td>nw-1</td>
<td>nw-1</td>
<td>nw-1</td>
</tr>
<tr>
<td>DIVAx</td>
<td>nw-(0)</td>
<td>nw-(0)</td>
<td>nw-(0)</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
</tr>
<tr>
<td>RSElx</td>
<td>nw-(0)</td>
<td>nw-(0)</td>
<td>nw-(0)</td>
<td>nw-(0)</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
<td>nw-0</td>
</tr>
</tbody>
</table>

XT2OFF
- Bit 7: XT2 off. This bit turns off the XT2 oscillator.
  - 0: XT2 is on
  - 1: XT2 is off if it is not used for MCLK or SMCLK.

XTS
- Bit 6: LFXT1 mode select.
  - 0: Low-frequency mode
  - 1: High-frequency mode

DIVAx
- Bits 5-4: Divider for ACLK
  - 00: 1
  - 01: 2
  - 10: 4
  - 11: 8

RSElx
- Bits 3-0: Range select. Sixteen different frequency ranges are available. The lowest frequency range is selected by setting RSElx = 0. RSEL3 is ignored when DCOx = 1.
## Clock Registers (2/2)

### 5.3.3 BCSCCTL2, Basic Clock System Control Register 2

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>SELMx</td>
<td>Bits 7-6 Select MCLK. These bits select the MCLK source.</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>00 DCOCCLK</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>01 DCOCCLK</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>10 XT2CLK when XT2 oscillator present on-chip. LFXT1CLK or VLOCLK when XT2 oscillator not present on-chip.</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>11 LFXT1CLK or VLOCLK</td>
</tr>
<tr>
<td>2</td>
<td>DIVMx</td>
<td>Bits 5-4 Divider for MCLK</td>
</tr>
<tr>
<td>1</td>
<td>SELS</td>
<td>Bit 3 Select SMCLK. This bit selects the SMCLK source.</td>
</tr>
<tr>
<td>0</td>
<td>DIVSx</td>
<td>Bits 2-1 Divider for SMCLK</td>
</tr>
<tr>
<td></td>
<td>DCOR</td>
<td>Bit 0 DCO resistor select. Not available in all devices. See the device-specific data sheet.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 Internal resistor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 External resistor</td>
</tr>
</tbody>
</table>

---

## Timer A Block Diagram

![Timer A Block Diagram](image.png)
Timer A Operation – Up/Down Mode

Figure 12-14. Output Example—Timer in Up/Down Mode

Timer A Registers (1/2)

12.3.1 TACTL, Timer_A Control Register

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TASSELx</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>rw(0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>rw(0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>rw(0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>rw(0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>rw(0)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>rw(0)</td>
<td></td>
</tr>
</tbody>
</table>

Unused

|    |    |    |    |    |    |    |    |

TASSELx

<table>
<thead>
<tr>
<th>9-8</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>TACLK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>ACLK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>SMCLK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>INCLK (INCLK is device-specific and is often assigned to the inverted TBCLK) (see the device-specific data sheet)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IDx

<table>
<thead>
<tr>
<th>7-6</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>/2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>/4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>/8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MCx

<table>
<thead>
<tr>
<th>5-4</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Stop mode: the timer is halted.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>Up mode: the timer counts up to TACCR0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Continuous mode: the timer counts up to 0xFFFF, then down to 0000h.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Updown mode: the timer counts up to TACCR0 then down to 0000h.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Unused

| 3   |    |    |    |    |    |    |    |

TACLR

| 2   |    |    |    |    |    |    |    |

Bit 2 Timer_A clear. Setting this bit resets TAR, the clock divider, and the count direction. The TACLR bit is automatically reset and is always read as zero.

TAIE

| 1   |    |    |    |    |    |    |    |

Bit 1 Timer_A interrupt enable. This bit enables the TAIIFG interrupt request.

0 | Interrupt disabled |
1 | Interrupt enabled |

TAIFG

| 0   |    |    |    |    |    |    |    |

Bit 0 Timer_A interrupt flag

0 | No interrupt pending |
1 | Interrupt pending |
### Timer A Registers (2/2)

#### 12.3.4 TACCTLx, Capture/Compare Control Register

<table>
<thead>
<tr>
<th>Bit</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>CMx</td>
</tr>
<tr>
<td>14</td>
<td>CCISx</td>
</tr>
<tr>
<td>13</td>
<td>SCS</td>
</tr>
<tr>
<td>12</td>
<td>SCCI</td>
</tr>
<tr>
<td>11</td>
<td>Unused</td>
</tr>
<tr>
<td>10</td>
<td>CAP</td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

#### CMx
- Bit 15-14: Capture mode
  - 00: No capture
  - 01: Capture on rising edge
  - 10: Capture on falling edge
  - 11: Capture on both rising and falling edges

#### CCISx
- Bit 13-12: Capture/compare input select. These bits select the TACCx input signal. See the device-specific data sheet for specific signal connections.
  - 00: CCIxA
  - 01: CCIxB
  - 10: GND
  - 11: Vcc

#### SCS
- Bit 11: Synchronize capture source. This bit is used to synchronize the capture input signal with the timer clock.
  - 0: Asynchronous capture
  - 1: Synchronous capture

#### SCCI
- Bit 10: Synchronized capture/compare input. The selected CCI input signal is latched with the EQUx signal and can be read via this bit

#### Unused
- Bit 9: Unused. Read only. Always read as 0.

#### CAP
- Bit 8: Capture mode
  - 0: Compare mode
  - 1: Capture mode

#### OUTMODx
- Bits 7-5: Output mode. Modes 2, 3, 6, and 7 are not useful for TACCR0, because EQUx = EQU0.
  - 000: OUT bit value
  - 001: Set
  - 010: Toggle/reset
  - 011: Set/reset
  - 100: Toggle
  - 101: Reset
  - 110: Toggle/set
  - 111: Reset/set

### Interrupts

![Interrupts Diagram](image-url)
Example Codes From Class

Setting I/O

#include <msp430.h>

int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

    // Set P1.0 to output (high)
    P1DIR |= BIT0;
    P1OUT |= BIT0;

    while(1)
    {
        __no_operation();
    }
}
Pulsing I/O

int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

    // Set P1.0 to output (high)
    P1DIR |= BIT0;
    P1OUT |= BIT0;

    while(1)
    {
        P1OUT ^= BIT0;
        __no_operation();
    }
}

Setting Up Clocks

int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

    // Set P1.0 to output (high)
    P1DIR |= BIT0;
    P1OUT |= BIT0;

    // Set System Clock to 16 MHz; Set ACLK to VLO
    DCOCTL = DCO0 + DCO1 + DCO2;
    BCSCCTL1 = DIVA0 + DIVA1 + RSEL0 + RSEL1 + RSEL2 + RSEL3;
    BCSCCTL2 = SELM_0 + DIVM_0;
    BCSCCTL3 = LFXT1S_2;

    while(1)
    {
        P1OUT ^= BIT0;
    }
}
Problems with CPU PWM

```c
int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

    int i;

    // Set P1.0 to output (high)
    P1DIR |= BIT0;
    P1OUT |= BIT0;

    // Set System Clock to 16 MHz; Set ACLK to VLO
    DCOCTL = DCO0 + DCO1 + DCO2;
    BCSCCTL1 = DIVA0 + DIVA1 + RSEL0 + RSEL1 + RSEL2 + RSEL3;
    BCSCCTL2 = SELM_0 + DIVM_0;
    BCSCCTL3 = LFXT1S_2;

    while(1)
    {
        P1OUT ^= BIT0;
        for (i = 0; i < 50; i++)
        {
            __no_operation();
        }
    }
}
```

Using TimerA

```c
int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

    // Set P1.0 to output (high)
    P1DIR |= BIT0;
    P1OUT |= BIT0;

    // Set P1.6 to TA0.1; Set P1.0 to TA0CLK; Set P1.1 to TA0.0
    P1DIR |= BIT0 + BIT6 + BIT1;
    P1SEL |= BIT0 + BIT6 + BIT1;
    P1SEL2 &= ~(BIT0 + BIT6 + BIT1);

    TA0CTL = ID_3 + MC_3 + TASEL0; // 8x divider, up/down mode, ACLK source \rightarrow 12kHz/8/8 = 187.5 Hz
    TA0CCR0 = 93; // ~1Hz period
    TA0CCTL1 = OUTMOD_2; // toggle/reset
    TA0CCTL1 = 46; // 50% duty

    // Set System Clock to 16 MHz; Set ACLK to VLO
    DCOCTL = DCO0 + DCO1 + DCO2;
    BCSCCTL1 = DIVA0 + DIVA1 + RSEL0 + RSEL1 + RSEL2 + RSEL3;
    BCSCCTL2 = SELM_0 + DIVM_0;
    BCSCCTL3 = LFXT1S_2;

    while(1)
    {
        __no_operation();
    }
}
```
#include <msp430.h>

int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

    //Interrupt Section
    TA0CCTL0 |= CCIE;
    __BIS_SR(GIE);

    while(1) {
        for (i = 0; i<50; i++) {
            __no_operation();
        }
    }

    // TA0_A1 Interrupt vector
#pragma vector=TIMER0_A0_VECTOR
__interrupt void Timer_A(void) {
    TA0CCR1 = TA0CCR1 + 5;
    if (TA0CCR1 > 93) {
        TA0CCR1 = 5;
    }
}

EXPERIMENT 3
Prelab Assignment
Experiment 3
ECE 482

Fig. 1 shows the power stage of the drivetrain boost converter to be assembled in experiment 3. For all parts of this prelab, consider operation of the converter at an operating point around which:

- $V_{bat} = 25 \, \text{V}$
- $V_{bus} \leq 50 \, \text{V}$
- $5 \, \text{kHz} \leq f_s \leq 1 \, \text{MHz}$
- $\Delta V_{out} \leq 1 \, \text{V}$

**Figure 1**: Open loop boost converter (implementation shown with MOSFET devices)
Design Assessment

In experiment 3, a portion of your grade will be the performance of the design that you choose to build. A 20% segment of the lab grade will be determined by the following formula, which rewards designs with small size, high efficiency, and high power capability:

$$\text{Grade} \, [\%] = 25 - \kappa_{core} - 100 \cdot (0.98 - \eta_{P_{\text{max}}=100}) - \left| \frac{P_{\text{max}} - 250}{50} \right|,$$

where

$$\kappa_{core} = \begin{cases} 0, & ETD29 \lor EFD25 \\ 3, & ETD39 \\ 6, & ETD44 \\ 9, & ETD49 \end{cases}$$

According to the inductor core you have chosen for your design. $P_{\text{max}}$ is the maximum power tested, which must be at least 100W, and may be as high as 250W.

Boost Design

**Magnetics Library**

<table>
<thead>
<tr>
<th>Core Geometry</th>
<th>Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFD25</td>
<td>Ferromax 3C90</td>
</tr>
<tr>
<td></td>
<td>Ferromax 3F3</td>
</tr>
<tr>
<td></td>
<td>Ferromax 3F4</td>
</tr>
<tr>
<td>ETD29</td>
<td>Ferromax 3C90</td>
</tr>
<tr>
<td></td>
<td>Ferromax 3F3</td>
</tr>
<tr>
<td>ETD39</td>
<td>Ferromax 3C90</td>
</tr>
<tr>
<td></td>
<td>Ferromax 3F3</td>
</tr>
<tr>
<td>ETD44</td>
<td>Ferromax 3C90</td>
</tr>
<tr>
<td></td>
<td>Ferromax 3F3</td>
</tr>
<tr>
<td>ETD49</td>
<td>Ferromax 3C90</td>
</tr>
<tr>
<td></td>
<td>Ferromax 3F3</td>
</tr>
</tbody>
</table>

**Core Loss Parameters**

<table>
<thead>
<tr>
<th>Wire Gauge</th>
<th>Diameter [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWG 10</td>
<td>0.267</td>
</tr>
<tr>
<td>AWG 12</td>
<td>0.213</td>
</tr>
<tr>
<td>AWG 14</td>
<td>0.171</td>
</tr>
<tr>
<td>AWG 16</td>
<td>0.137</td>
</tr>
<tr>
<td>AWG 20</td>
<td>0.0874</td>
</tr>
</tbody>
</table>

Full AWG table

**Power Semiconductors**

<table>
<thead>
<tr>
<th>Part No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOT2500L</td>
<td>150 V, 150 A High Voltage Trench MOSFET</td>
</tr>
<tr>
<td>FDP083N15A</td>
<td>150 V, 117 A PowerTrench MOSFET</td>
</tr>
<tr>
<td>IPP200N15N3</td>
<td>150 V, 50 A OptiMOS Power MOSFET</td>
</tr>
<tr>
<td>irfb4615pbf</td>
<td>150 V, 35 A HEXFET Power MOSFET</td>
</tr>
<tr>
<td>CSD1953S5KCS</td>
<td>100 V, 150 A NexFET Power MOSFET</td>
</tr>
<tr>
<td>IPP023N10N5</td>
<td>100 V, 120 A OptiMOS Power MOSFET</td>
</tr>
<tr>
<td>FGPF50N33BT</td>
<td>330 V, PDP Trench IGBT</td>
</tr>
<tr>
<td>ISL9V3040D</td>
<td>400 V, N-Channel IGBT</td>
</tr>
</tbody>
</table>
## Supplemental Lectures

<table>
<thead>
<tr>
<th>Device</th>
<th>Loss Mechanism</th>
<th>ECE 481</th>
<th>ECE 581</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOSFET</td>
<td>$R_{on}$</td>
<td>Lecture 7-8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$C_{oss}$</td>
<td>Lecture 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overlap</td>
<td>Lecture 5-6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$P_{g}$</td>
<td>Lecture 5</td>
<td></td>
</tr>
<tr>
<td>Diode</td>
<td>$V_F$</td>
<td>Lecture 7-8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$R_d$</td>
<td>Lecture 7-8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$t_d$ cond</td>
<td>Lecture 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$C_d$</td>
<td>Lecture 7 (see: $C_{oss}$)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Reverse-Recovery</td>
<td>Lecture 11</td>
<td></td>
</tr>
<tr>
<td>Inductor</td>
<td>$R_{dc}$</td>
<td>Lecture 38</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Skin Effect</td>
<td>Lecture 39</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Core Loss</td>
<td>Lecture 39</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fringing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Proximity</td>
<td>Lecture 39</td>
<td></td>
</tr>
</tbody>
</table>


ECE481: [http://web.eecs.utk.edu/~dcostine/ECE481/Fall2017/schedule.php](http://web.eecs.utk.edu/~dcostine/ECE481/Fall2017/schedule.php)