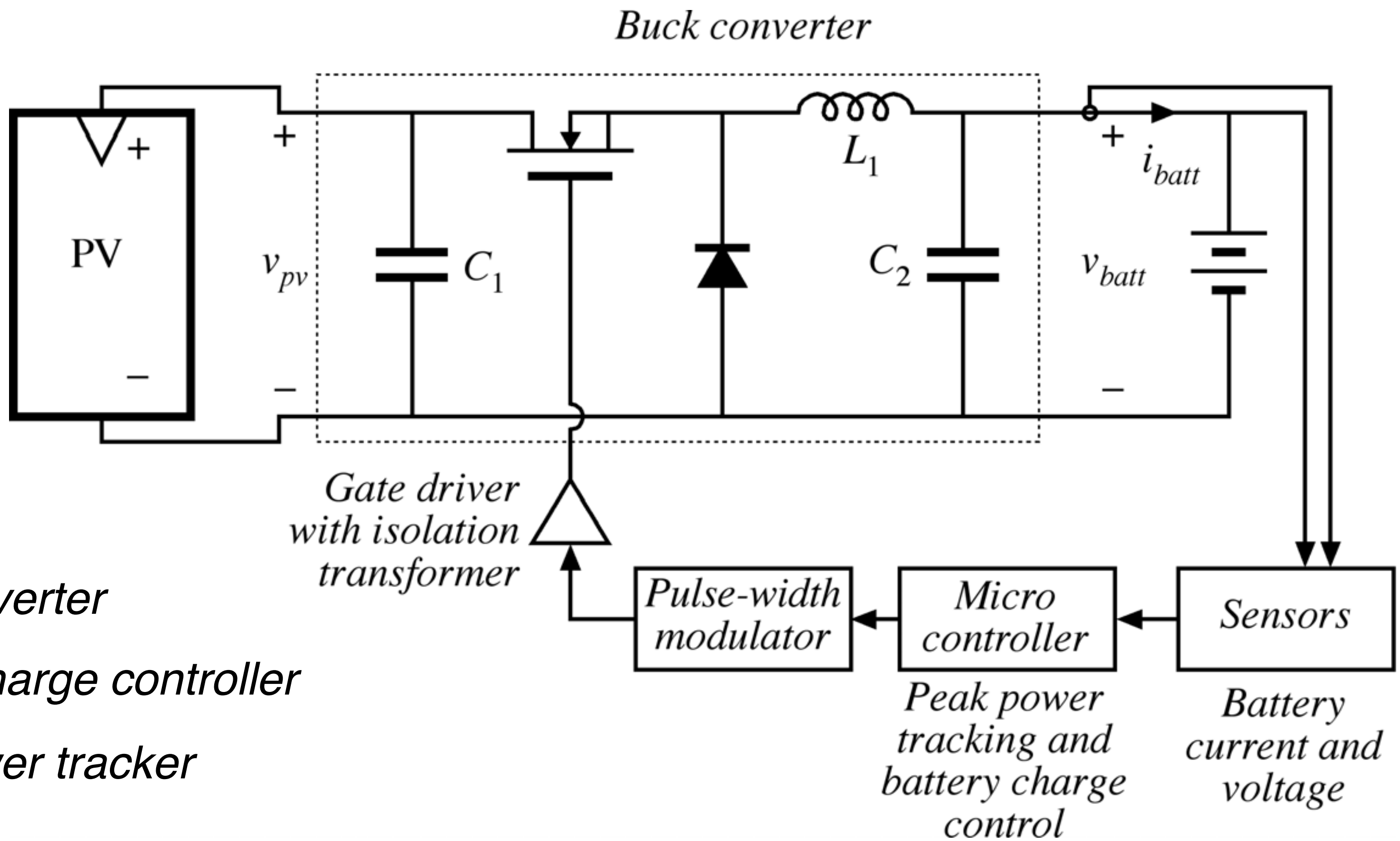
Lecture 5

ECEN 4517/5517

Experiment 3



Buck converter

Battery charge controller

Peak power tracker

Due dates

Next week:

Exp. 3 part 2 prelab assignment: MPPT algorithm

Late assignments will not be accepted.

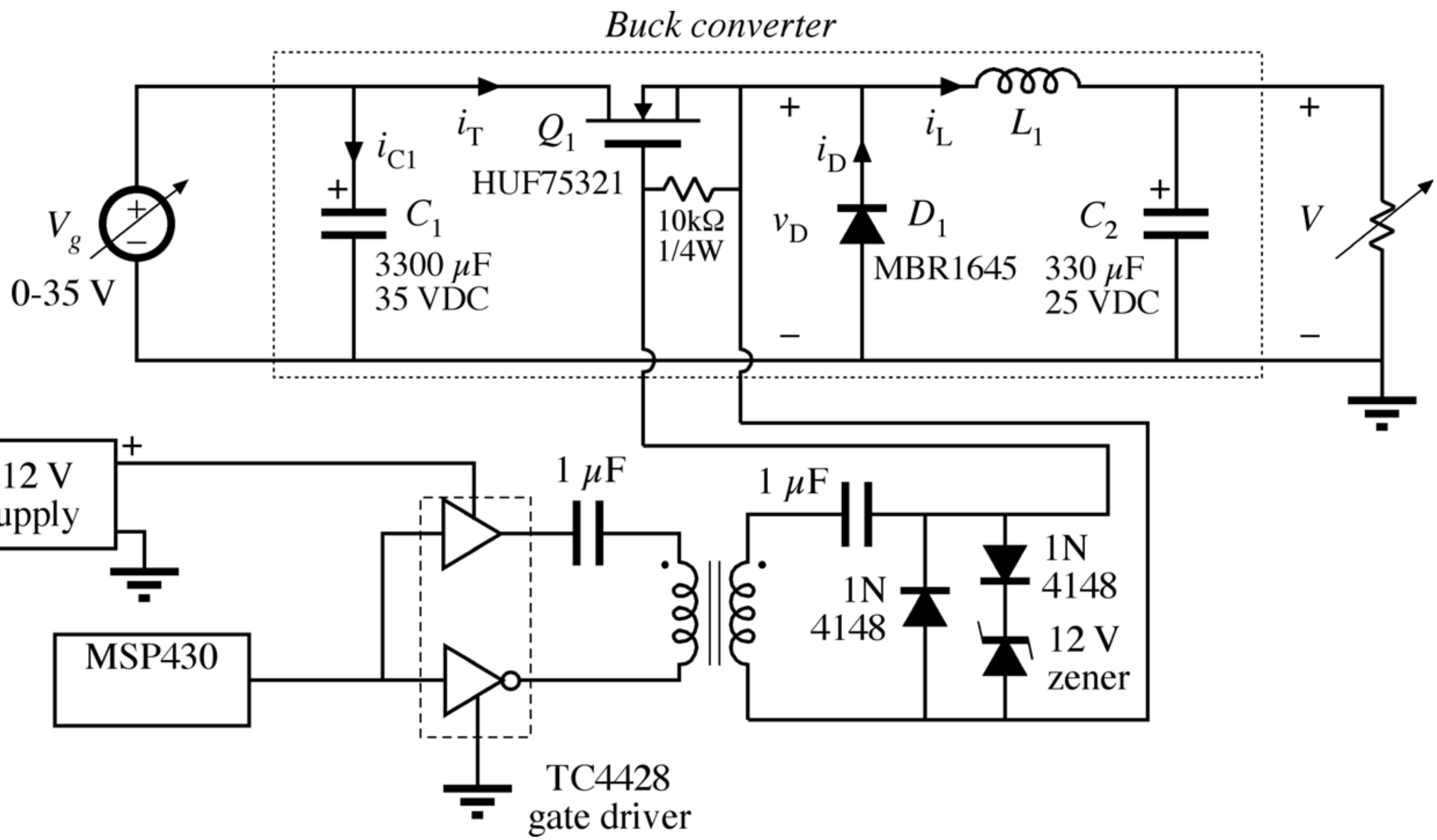
Due at noon next Tuesday in D2L

This week:

Finish Exp. 3 part 1!

Exp. 3, Part 1

Demonstrate buck power stage



Heatsinks

The power semiconductors generally require heatsinks. Example—from the HUF35371 (our 55 V, 34 mΩ MOSFET) datasheet:

THERMAL SPECIFICATIONS						
Thermal Resistance Junction to Case	$R_{\theta JC}$	(Figure 3)	-	-	1.6	$^oC/W$
Thermal Resistance Junction to Ambient	$R_{\theta JA}$	TO-220, TO-263	-	-	62	$^oC/W$

Multiply thermal resistance by power loss to find temperature rise

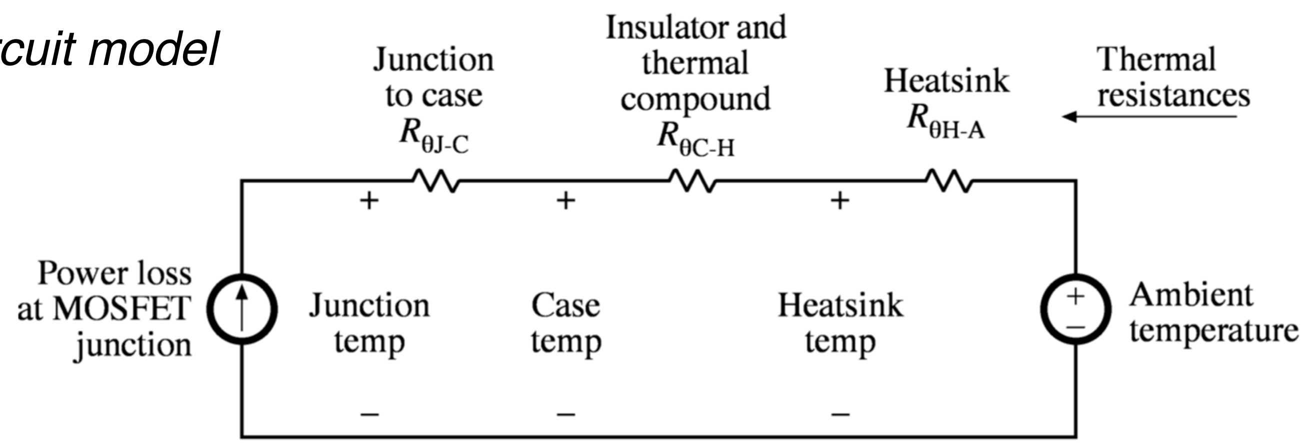
With no heatsink, the thermal resistance is quite high (62˚C/W)
With a 25˚C ambient temperature and no heatsink, this device will reach the rated limit of 175˚C if its power dissipation is

$$P_{loss} = (175^\circ C - 25^\circ C)/(62^\circ C/W) = 2.4\ W$$

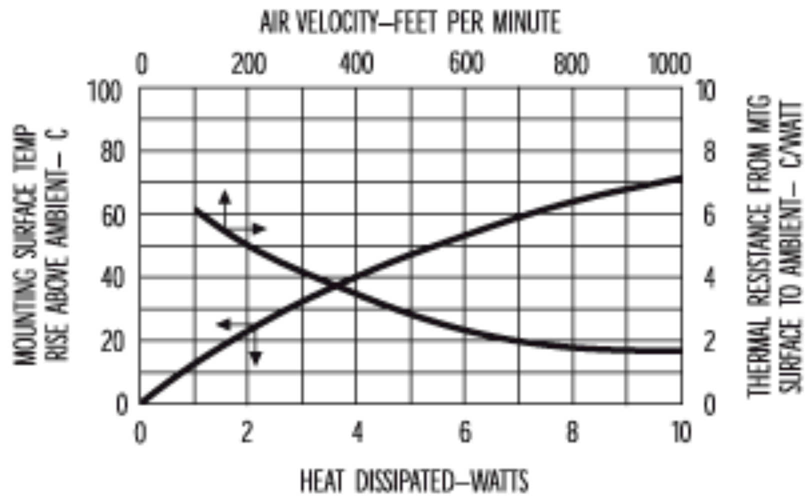
A heatsink can lower this temperature rise considerably. The junction-to-case thermal resistance is only 1.6 ˚C/W.

For reliability reasons, we like to limit temperature rises to much lower values— perhaps a few tens of ˚C

Thermal equivalent circuit model



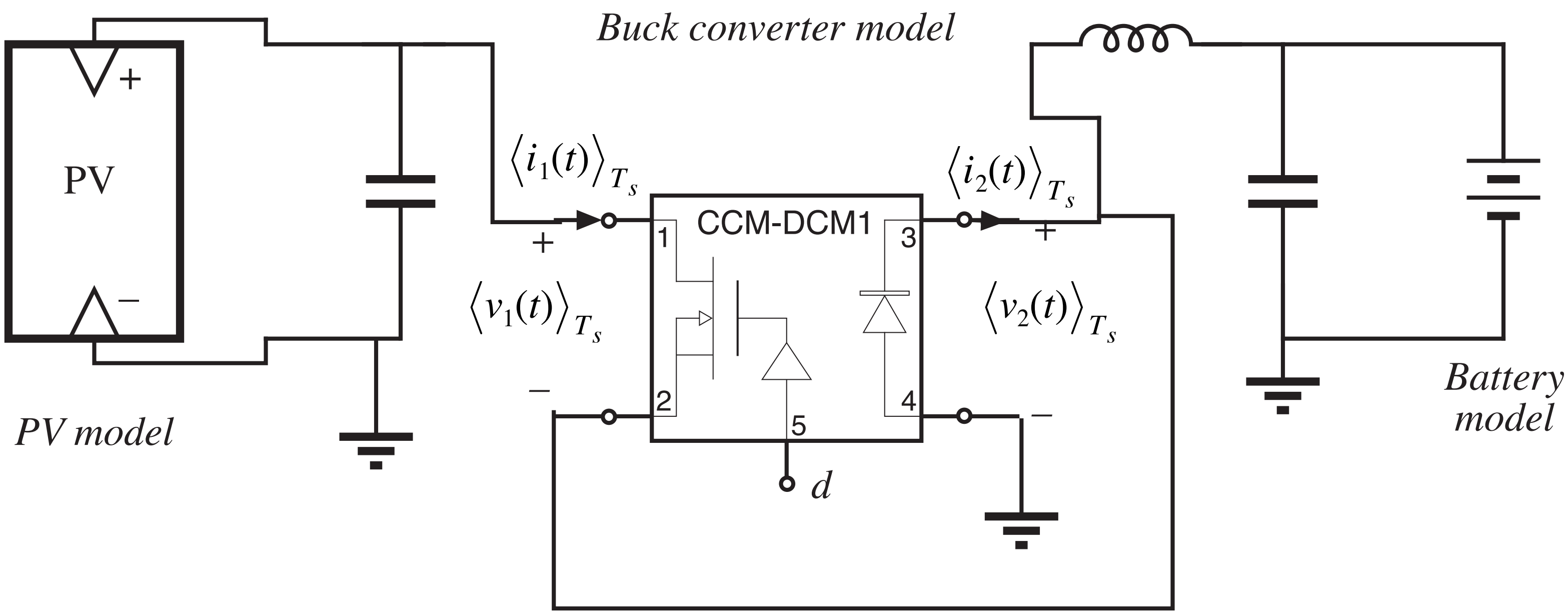
The parts kit heatsinks:



From the graph, 2.4 W of loss causes a 30 ˚C rise, which would make the heatsink operate at 55˚C for a 25˚C ambient.
Plus junction-to-case temperature rise of (1.6˚C/W)(2.4 W) = 4˚C

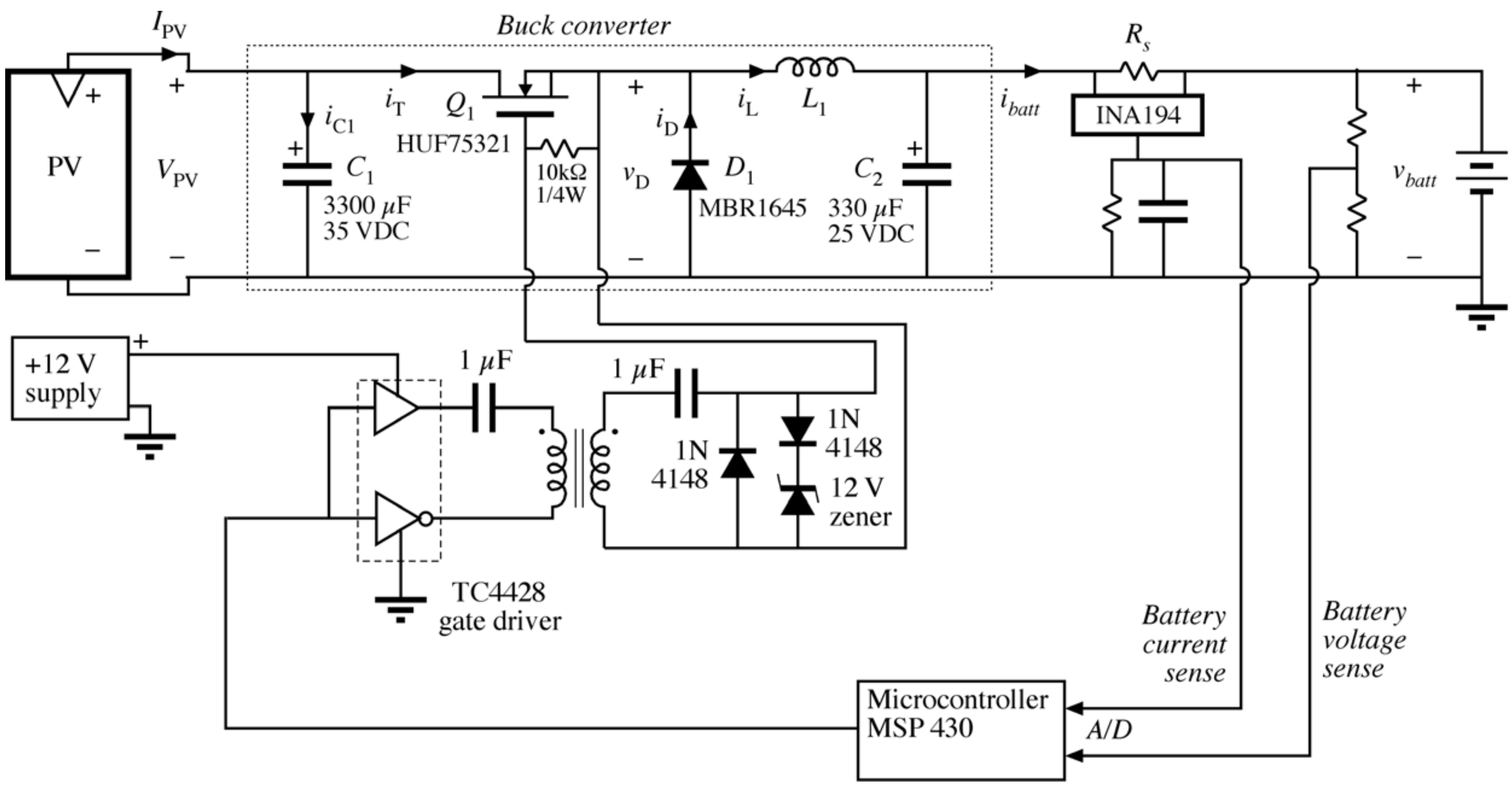
PSPICE simulation

Exp. 3 Part 1: open loop



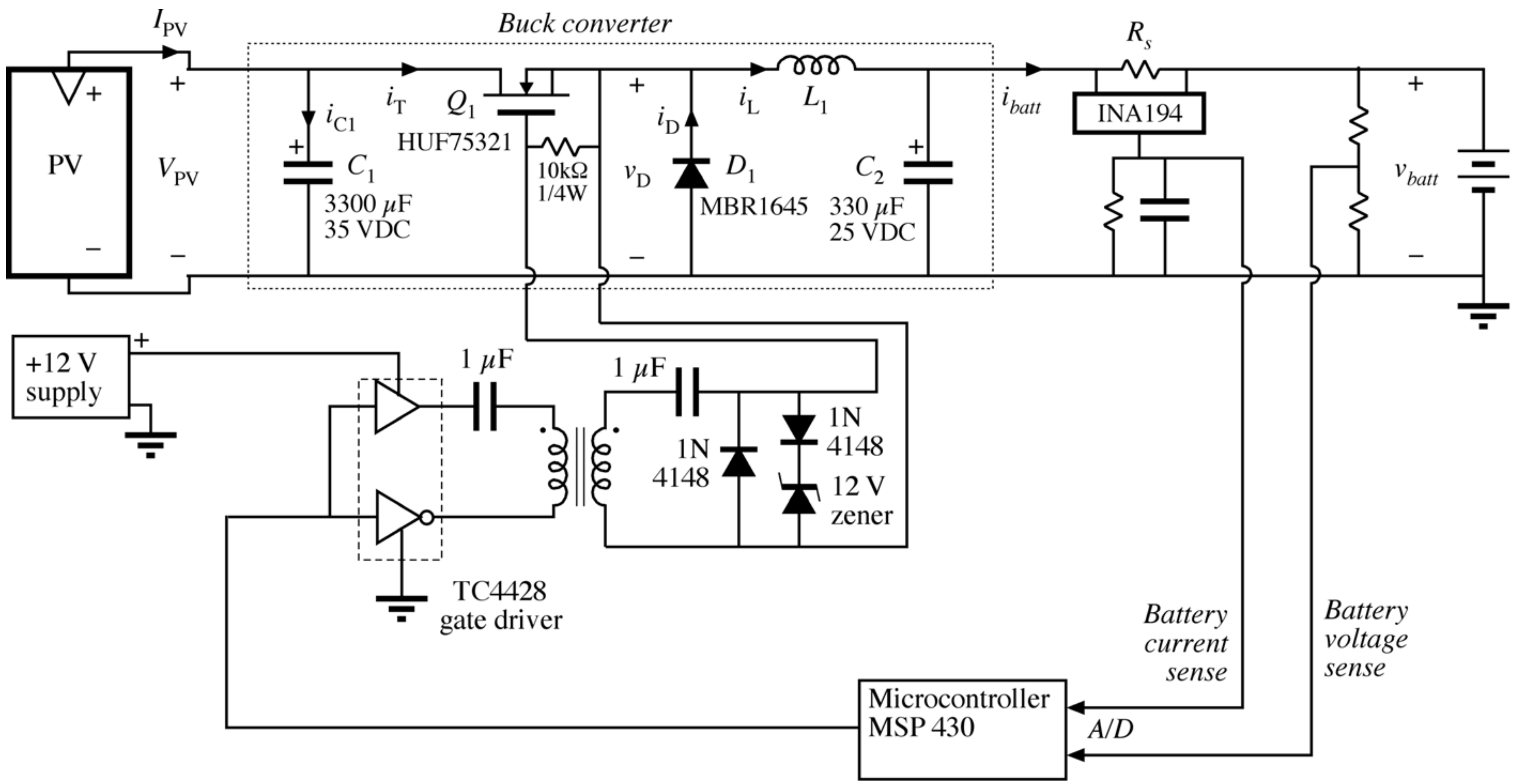
- Use your PV model from Exp. 1
- Replace buck converter switches with averaged switch model
- CCM-DCM1 and other PSPICE model library elements are linked on course web page

Exp. 3 Part 2



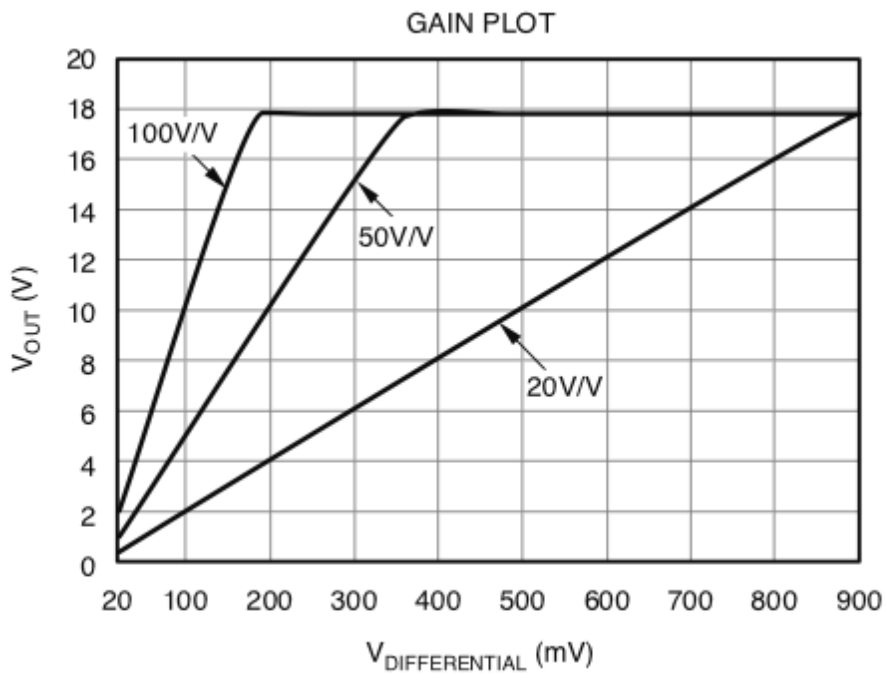
- Implement maximum power point tracking algorithm
- Demonstrate on PV cart outside

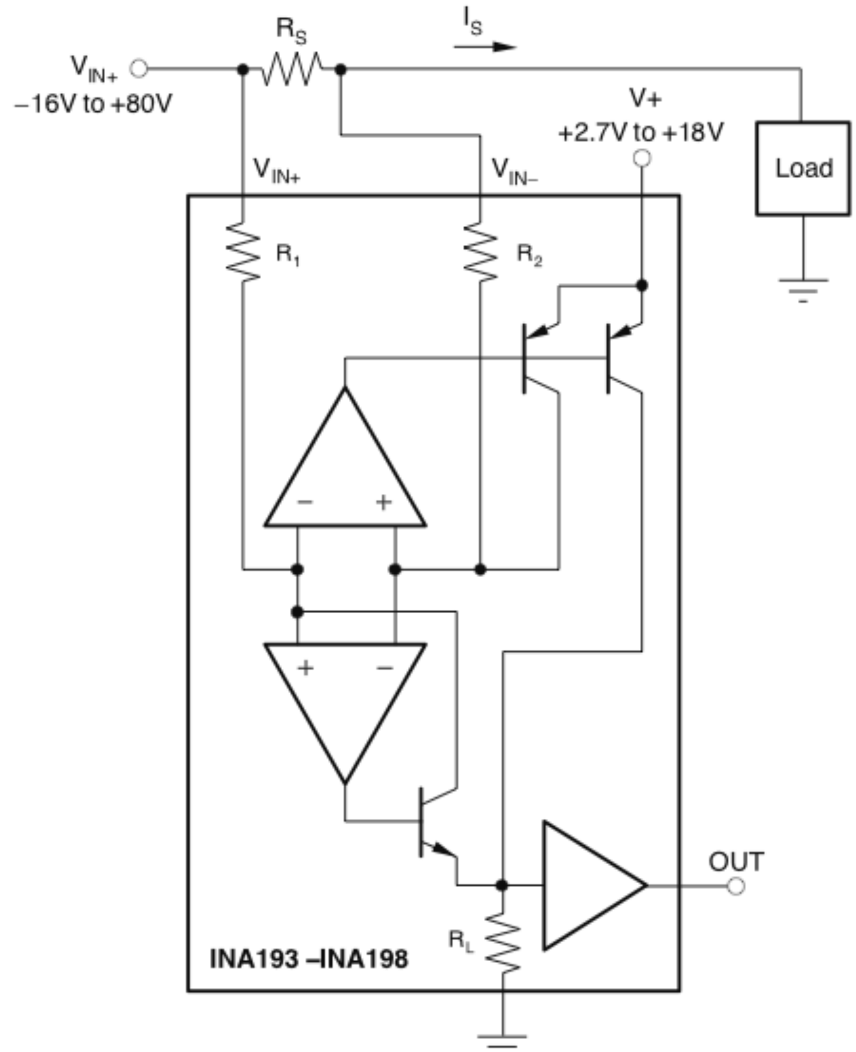
Sensing the battery current and voltage
Exp. 3 Part 2



INA194 High-side current sense IC

INA194: *gain* = 50V/V





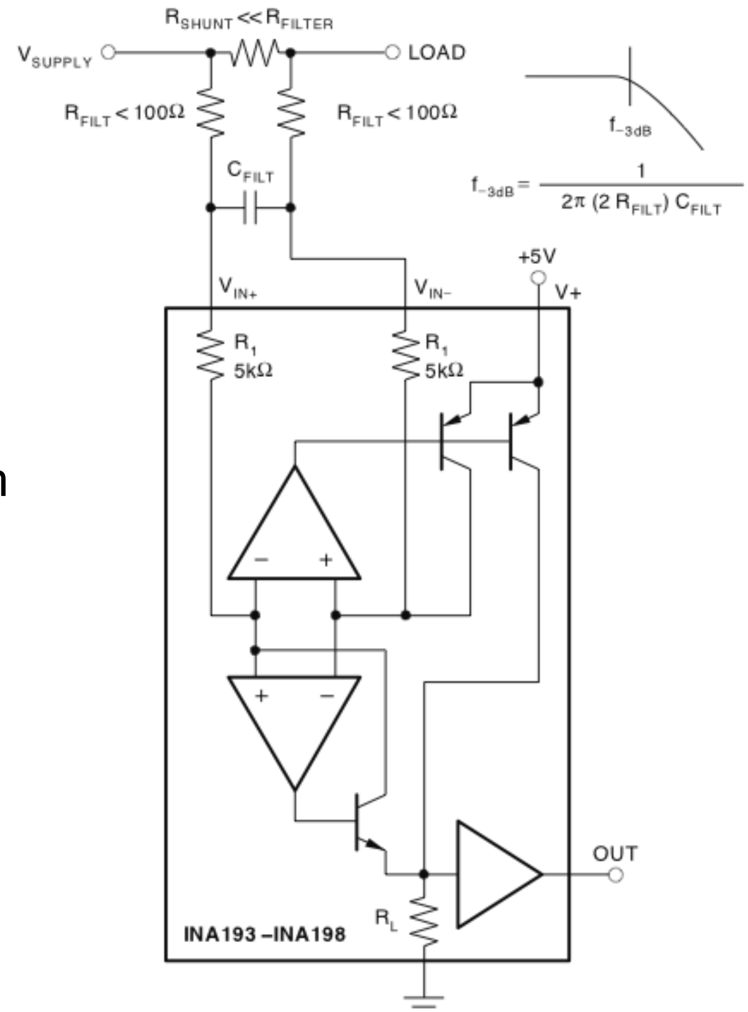
About the INA194

Must bypass power supply pins!

Filtering the waveforms:

$$\text{Gain Error\%} = 100 - \left(100 \times \frac{5k\Omega}{5k\Omega + R_{FILT}}\right)$$

- Use twisted pair to transmit signal from INA194 output to your MSP430 board
- An *R-C* filter will likely be necessary at A/D input of MSP430



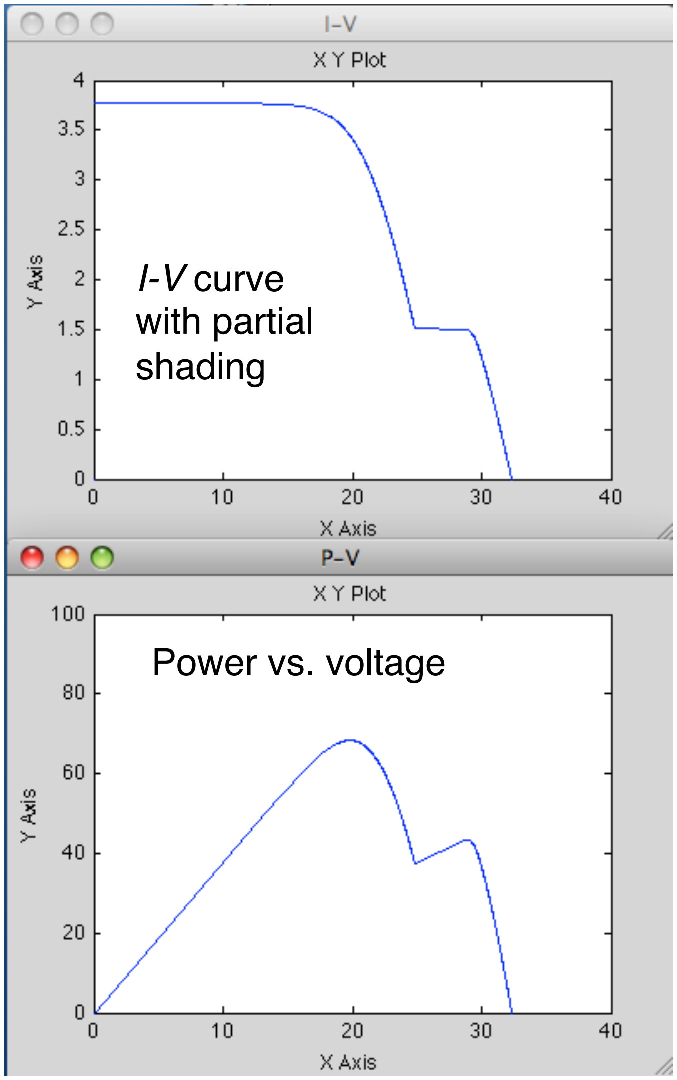
Maximum Power Point Tracking

Automatically operate the PV panel at its maximum power point

Some possible MPPT algorithms:

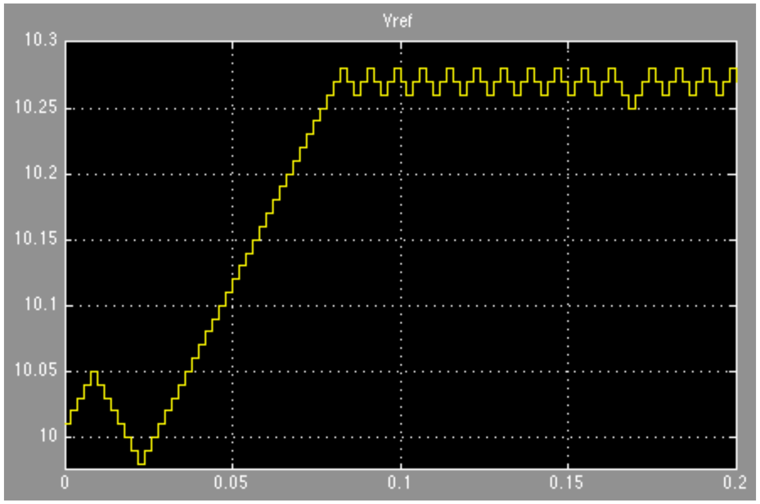
- Perturb and observe
- Periodic scan
- Newton's method, or related hill-climbing algorithms
- What is the control variable? Where is the power measured?

Next week's prelab assignment: propose a MPPT algorithm, submit flowchart/block diagram



Example MPPT: Perturb and Observe

- A well-known approach
- Works well if properly tuned
- When not well tuned, maximum power point tracker (MPPT) is slow and can get confused by rapid changes in operating point
- A common choice: "control" is switch duty cycle



Basic algorithm

Measure power
Loop:

- Perturb the operating point in some direction
- Wait for system to settle
- Measure power
- Did the power increase?

  Yes: retain direction for next perturbation

  N: reverse direction for next perturbation

Repeat

Example MPPT: Sweep

Start at V = minimum PV voltage. Set Pmax = 0.

Loop:

 Wait for system transients to settle

 Measure power P. Is P > Pmax?

  Yes: set Pmax = P, Vopt = V

 Increase V by one step

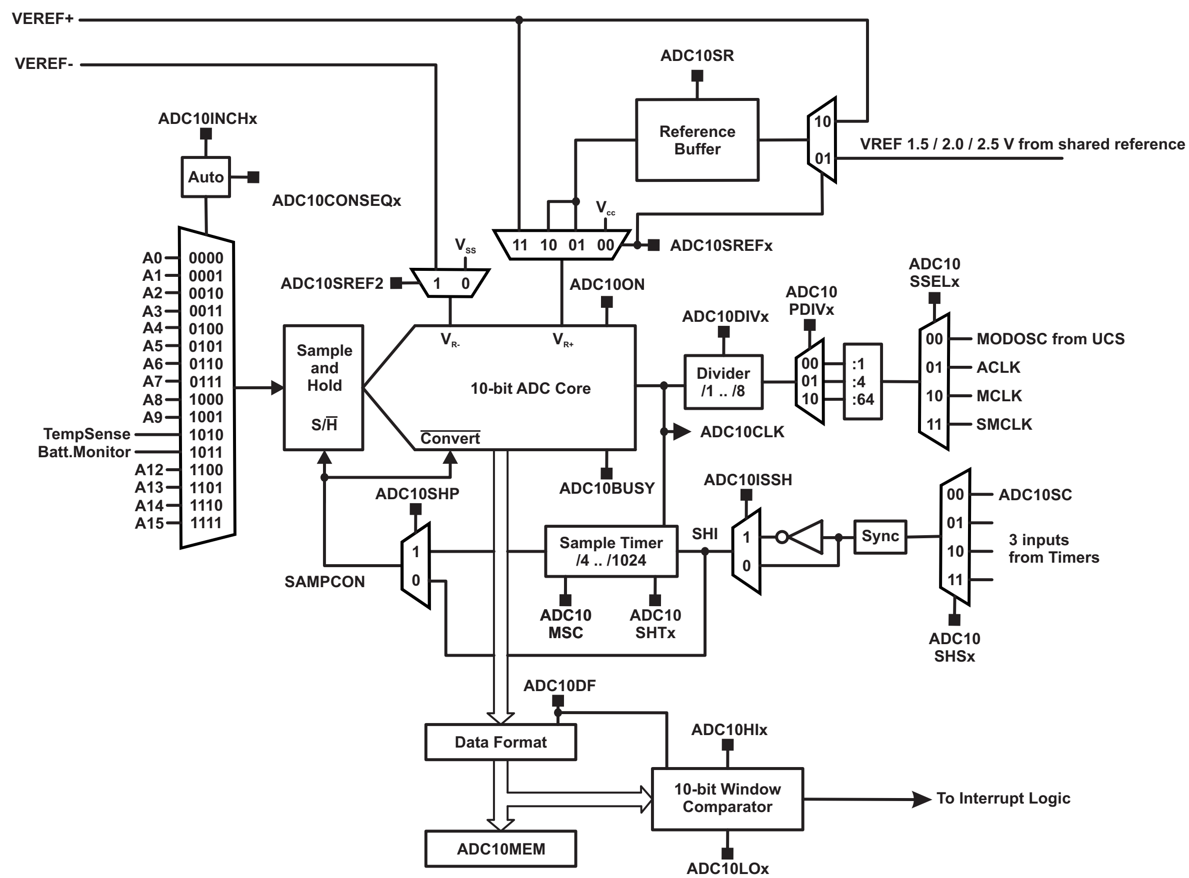
 Repeat until V = Voc

Set V = Vopt. Wait some time, then sweep again.

ADC10: The 10-Bit A/D Converter
of the MSP430

Key features:

- Multiplexed inputs
- Sample and hold circuit
- Successive approximation register, driven by selectable clock
- Selectable reference sources
- Buffered output memory
- 10 bit or 8 bit conversion



Successive Approximations

- After the input signal has been sampled, the 10-bit SAR requires 11 clock cycles to generate an output
- Compare analog input with references
- The MSP430 uses a switched capacitor scheme to perform the comparisons
- See *MSP430x5xx Family User's Guide*, Ch. 27

Reference: John H. Davies, *MSP430 Microcontroller Basics,* Elsevier, 2008, ISBN 987-7506-8276-3.

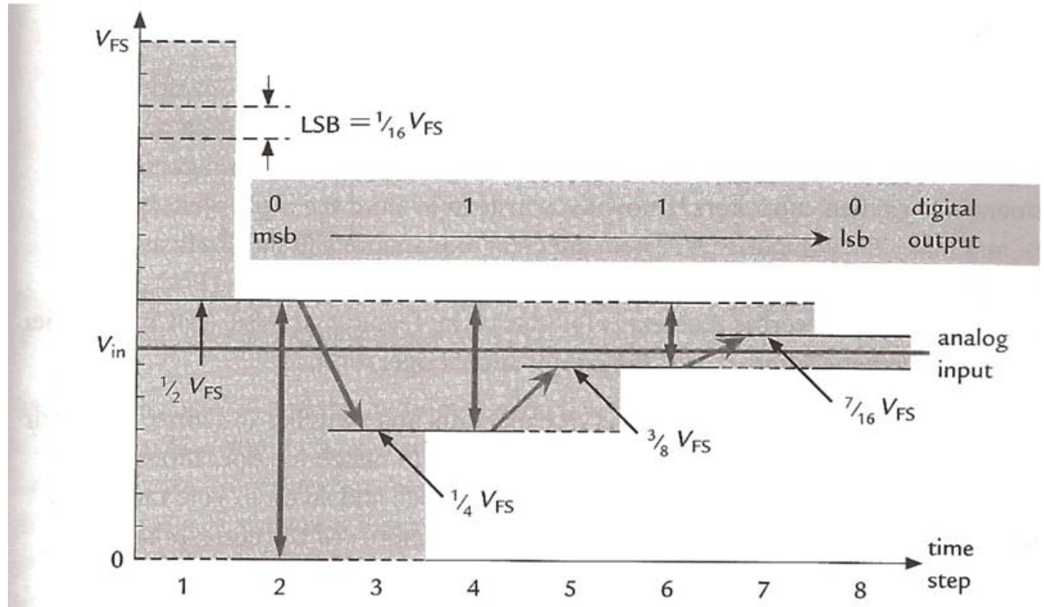


Figure 9.13: Operation of a 4-bit successive-operation ADC with an input of $V_{in} = 0.4V_{FS}$.

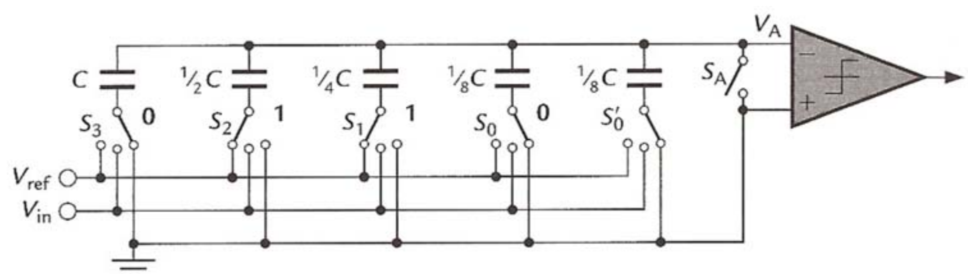
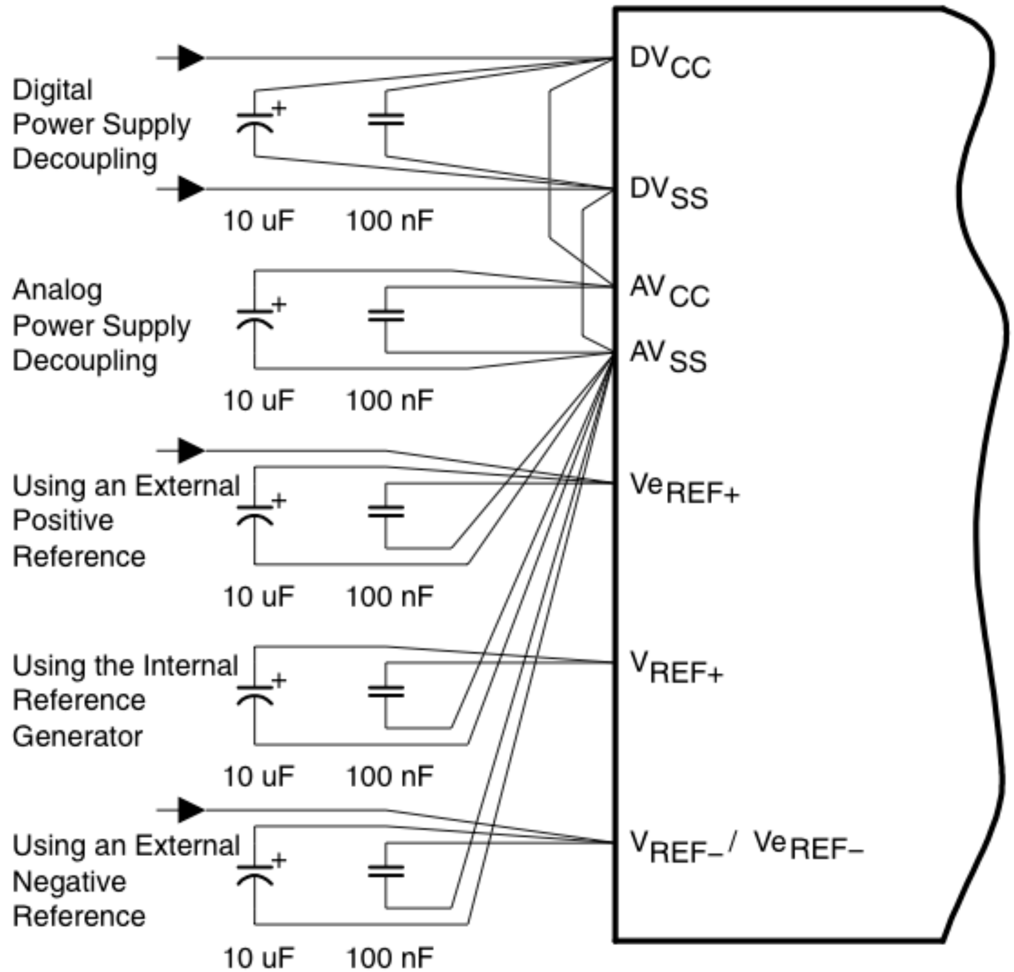


Figure 9.15: The SAR ADC with the switches in the final positions for an input of $V_{in} = 0.4V_{FS}$ and a binary output of 0110.

Capacitor bypassing is required

What the User's Guide recommends:

Also need capacitance at analog input pin



Setting up the A/D Converter ADC10

```
// Configure ADC10
ADC10CTL0 = ADC10SHT_2 + ADC10ON;            // sample time of 16 clocks, turn on
                                             // use internal ADC 5 MHz clock
ADC10CTL1 = ADC10SHP + ADC10CONSEQ_0;        // software trigger to start a sample
                                             // single channel conversion
ADC10CTL2 = ADC10RES;                        // use full 10 bit resolution
ADC10MCTL0 = ADC10SREF_1+ADC10INCH_5;        // ADC10 ref: use VREF and AVSS
                                             // input channel A5 (pin 10)
// Configure internal reference VREF
while(REFCTL0 & REFGENBUSY);                 // if ref gen is busy, wait
REFCTL0 l= REFVSEL_0 + REFON;                // select VREF = 1.5 V, turn on
_delay_cycles(75);                           // delay for VREF to settle
```

The above code sets up the 10-bit ADC with A5 as its only input, with 1.5 V giving a reading of $2^{10} - 1$, and 0 V giving a reading of 0. Each reading will employ a sampling window of 16 ADC clocks = 3.2 μsec.

Sampling the ADC input

```
ADC10CTL0 |= ADC10ENC + ADC10SC;        // sampling and conversion start
while(ADC10CTL1 & ADC10BUSY);           // wait for completion
X = ADC10MEM0;                          // ADC10MEM0 contains result
```

The above code is simple and a good start. See CCS5 code examples for use of interrupts that do not require the processor to wait during the conversion time.