Introduction

DC voltage conversion is a topic as old as electronics itself. As far as I can recall, there are three broad classes of DC converters. The first class is switching converters, which induce large current swings through an inductor to produce a higher or lower voltage at very high efficiency. The second class is linear regulators, which are not power efficient but are precise, small, and easy to implement. The third class, which we will examine here, are charge pumps – like switching regulators but utilizing a ladder of capacitors, diodes, and switches to "stack" voltages.

The first charge pump circuit for very large voltages was the Cockcroft-Walton generator, which was designed to amplify the AC mains voltage for particle physics research. While this circuit was (and still is) highly useful for large research equipment, it is the Dickson charge pump that is adaptable to most electronics problems today – it is small, scalable, and runs off a single DC supply. The Dickson charge pump topology requires only a power rail, two complementary clock lines, and a common ground rail.

Many Dickson circuits are adapted to low power operation. IC designers can include a Dickson onboard to provide necessary large local voltages (typically at low currents). Here we adapt the circuit for high power usage, and include a feedback monitor and large output capacitor to control its output voltage.

This circuit can provide an arbitrarily large voltage at low current for an extended period of time. A high-voltage linear regulator can be added to provide a more stable output voltage.[1] For low current the capacitor can be "topped off" by the microcontroller during operation. For large ($\geq 1A$) output currents the output voltage will remain stable on the order of milliseconds.

Hardware

The Dickson circuit relies on two out of phase clocks to successively charge and discharge capacitors. At the start (assuming CLK is low), V_{in} conducts through the diode and fills the capacitor C1. When CLK changes to high (and !CLK changes to low), the ground-referenced voltage at the D1/C1 junction increases to $V_{in} - V_D + V_C$. The diode D2 conducts to fill capacitor C2, and when CLK changes again, the voltage at the D2/C2 junction increases to $V_{in} - 2V_D + 2V_C$. Adding successive stages increases the multiplication factor linearly. After N stages the output voltage is $V_{out} = V_{in} - V_D + N(V_C - V_D)$.[2]

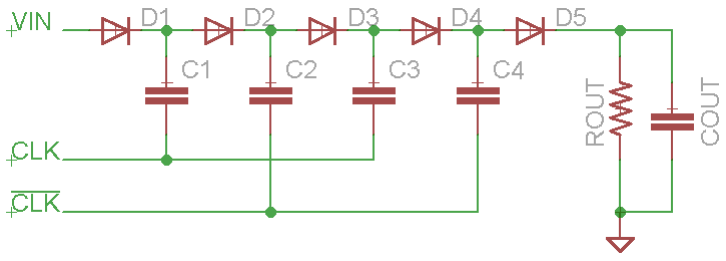

Figure 1: Four stage Dickson charge pump. D5 and Zout provide a smoothing function.

[1] http://www.linear.com/product/LT3012

[2] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.128.4085

There are nonideal factors that result in deviation from this ideal function. Stray capacitance in parallel with the boost capacitors reduces the capacitor voltage by a factor $C/(C + C_S)$. A load directly connected to the output reduces the output voltage by $I_{out}/(C \cdot f_{osc})$, where f_{osc} is the frequency of the clock. Our large output capacitor ($C_{out} \geq 1000C$) with finite parasitic resistance R_p will lead to a longer charging time and a small leakage current. To mitigate these nonidealities, we can 1) use large ($\geq 1\ \mu F$) capacitors for C, 2) use a fast ($\geq 10\ kHz$) PWM signal to keep output voltage loss low, and 3) use load switches and/or relays to keep the parasitic resistance R_p to a minimum.

Since there is no inherent limit to current draw, for the sake of safety it is wise to include a current limiting resistor between the input V_{in} and the first diode D1. This is especially true because we expect a large current spike from charging C_{out} upon enabling the circuit. Both the diodes and the resistor must be able to handle the maximum current. The diode voltage spec is less important. MUR405 and MUR410 diodes by On Semiconductor are typically sufficient. The capacitors in the circuit must be able to withstand the final output voltage V_{out}, which will vary by application. 100 volt ceramic capacitors are a safe bet.

Until now we have assumed ideal clocks with low output impedance. In a typical design scenario where the signals are provided by microcontroller PWM outputs, it is unlikely the clocks will have sufficiently low output impedance. This will slow the capacitor discharge time ($\tau = R_{out} \cdot \frac{1}{2}NC$), which will conflict with the need for a fast PWM frequency. A simple cascode inverter tied to V_{in} solves this problem, increases the capacitor discharge current, and sets V_C to V_{in} for the sake of simplicity.

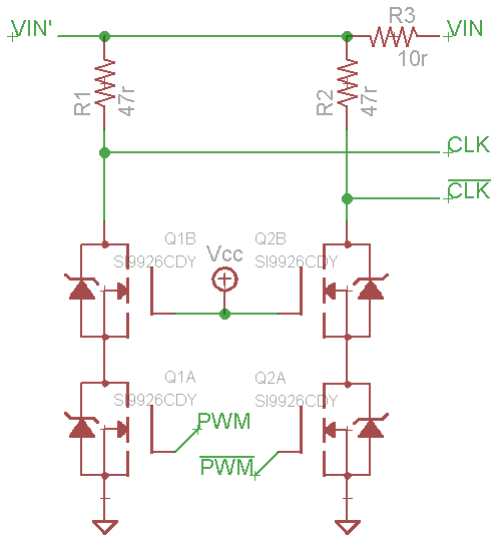

Figure 2: Cascode inverter and current limiting resistors.

Finally a buffered voltage divider provides the necessary feedback for ADC monitoring. Since our large output capacitor will charge slowly, the microcontroller will have plenty of time to compare the feedback to its set value and disable the charging process when appropriate. The microcontroller can also be set to “top off” the capacitor as parasitic resistance reduces its voltage.

Firmware

The firmware control for this circuit is comparatively simple. Two complementary PWM signals need to be generated, and one ADC needs to be active as part of a while loop check. Sample PWM setup code for an Atmega32U4 would look like the following:

```
void PWMsetup(void) // 16 MHz clock, 8-bit PWM6, set at OCR4A match, clear at TOP = OCR4C
{
	DDRC = 0xFF;  PORTC = 0x00; // default switch state off -> no current
	TCCR4A = 0x42; // enable OC4A, disable OC4B
	TCCR4C = 0x40; // shadow enable OC4A, disable OC4B
	TCCR4D = 0x02; // PWM6 single slope mode
	TCCR4E = 0x03; // enable OC4A and !OC4A
	TC4H = 0x00; // don't need the extra bits
	OCR4A = 0x7F; // this is the set point
	OCR4C = 0xFF; // this is the clear point
	TIMSK4 = 0x00; TIFR4 = 0x00; // no interrupts
}
void PWMenable(void) // scales down 16 MHz to 15.625 kHz and enables
	TCCR4B = 0x08;
void PWMdisable(void) // shuts off clock and disables PWM
	PORTC = 0x00; CCR4B = 0x00;
```

The functionality is readily visible – PWM is enabled and the output capacitor is charged to the desired level, at which point the PWM signals are stopped by being fixed to 0. (The cascode inverter converts this to a 1, which will deplete the voltage across all the capacitors.)

After enabling the ADC, a simple loop can be set up to charge and monitor the capacitor voltage:

```
void chargeCap(void) // charge capacitor to specified global value in VMAX
{
	ADCenable();
	PORTD |= 1<<CHARGE; // closes switches for charge circuit
	PWMenable();
	do
	{
		V = getADC();
		_delay_us(1000);
	}
	while (V < VMAX); // loop exits on charge complete
	PWMdisable();
	PORTD &= ~(1<<CHARGE);
	ADCdisable();
}
```

This function can also be called when needed to “top off” the output capacitor.

Performance

A Dickson circuit with $N = 8$ was built and evaluated. The charge rate can be compared to simulation and calculation to determine the equivalent charging resistance and deviation from the ideal case.

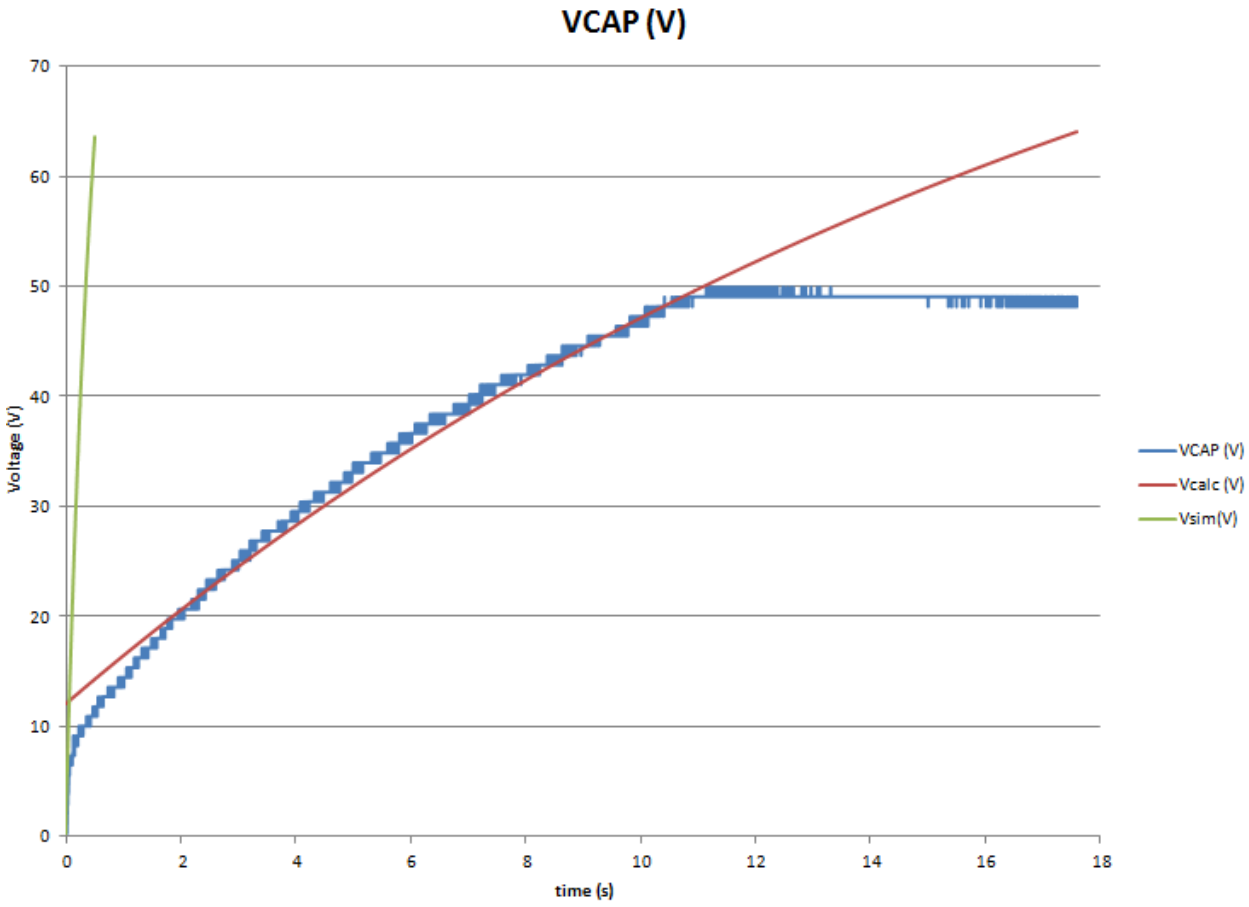

Figure 3: Measured (blue) and curve fit (red) for capacitor charging. Circuit was programmed to stop charging at 50 V.

The closest exponential curve fit (in red) yields an effective charging resistance of 8.9 kΩ. This is much higher than the expected effective charging resistance, which yields longer charge times and a lower efficiency. A simulation of the same circuit (green) shows the output voltage hitting 50 V in 0.5 seconds, with the limiting resistance being only the 10Ω current limiting resistor.

The leakage rate can be measured as well. Using the first-order exponential curve fit, we can determine the parasitic resistance from the leakage curve (given a 2200 µF capacitor) to be 191.6 kΩ. This parameter can be fed back into a simulation for testing potential applications.

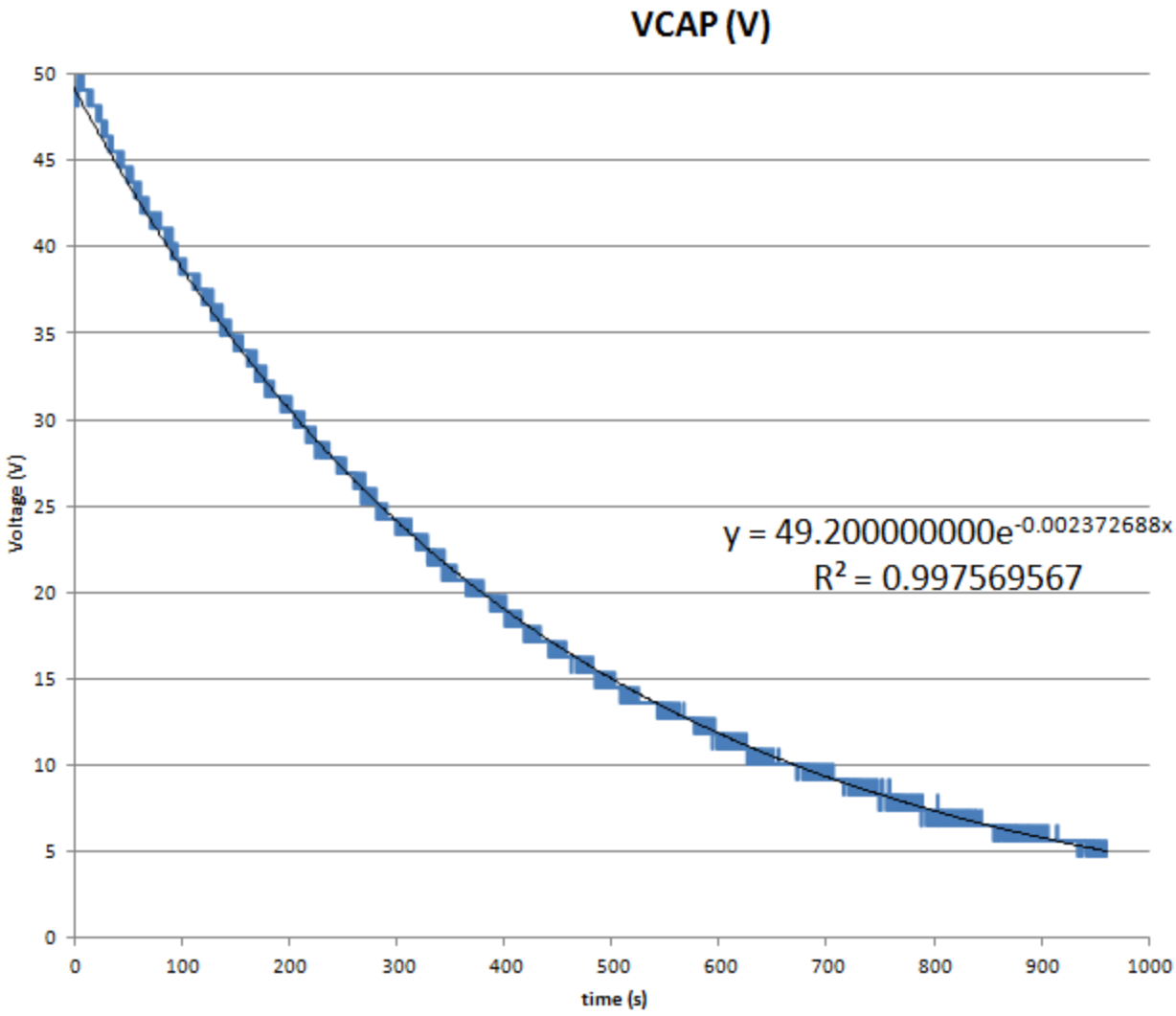

Figure 4: Long term decay of capacitor voltage over time. This yields an effective parasitic resistance.