

Voltage Integrator Calculation

Definitions and Equations - Multislope Continuous Runup with periodic ADC measurements

```
In[1]:= voutIn = -  $\frac{1}{r_{\text{Mes}} * C_{\text{int}}} * v_{\text{in}} * \Delta t_{\text{int}};$ 
In[2]:= voutRefPos = -  $\frac{1}{r_{\text{RefPos}} * C_{\text{int}}} * v_{\text{refPos}} * t_{\text{Pos}};$ 
In[3]:= voutRefNeg = -  $\frac{1}{r_{\text{RefNeg}} * C_{\text{int}}} * v_{\text{refNeg}} * t_{\text{Neg}};$ 
```

Solve for the unkown input voltage V_{in} we want to measure:

```
In[4]:= Solve[voutIn + voutRefPos + voutRefNeg == ΔvADC, vin] // FullSimplify
Out[4]=  $\left\{ \left\{ v_{\text{in}} \rightarrow 0. - \frac{1. C_{\text{int}} v_{\text{inFS}} \Delta v_{\text{ADC}}}{i_{\text{inFS}} \Delta t_{\text{int}}} \right\} \right\}$ 
```

Component value calculations and some error calculation

Design Choices:

```
In[5]:= iinFS = 0.0005; vinFS = 2.5; vrefPos = 5.0; vrefNeg = -5.0;
vsat = 11; vth = 5; trunup = 0.001; vrunupFS = -2750;
In[6]:= irefFS = iinFS * 1.25;
```

Resistor Values:

```
In[7]:= Dynamic[ $\left\{ r_{\text{Mes}} = \frac{v_{\text{inFS}}}{i_{\text{inFS}}}, r_{\text{RefPos}} = \frac{v_{\text{refPos}}}{i_{\text{refFS}}}, r_{\text{RefNeg}} = \frac{-v_{\text{refNeg}}}{i_{\text{refFS}}} \right\}$ ]
Out[7]= {5000., 8000., 8000.}
```

Integration Capacitor:

The integrator capacitor is determined by the virtual runup alone. We solve for it's value by assuming a one-input integrator that charges to v_{runupFS} within t_{runup} and has infinite output voltage capability, due to the fact that we later on subtract charge during the runup (multislope-runup).

```
In[8]:= Dynamic[capacitor = NSolve[ $\frac{1}{r_{\text{Mes}} * c_{\text{int}}} * v_{\text{inFS}} * t_{\text{runup}} == -v_{\text{runupFS}}, c_{\text{int}}]$ ]]
Out[8]=  $\{c_{\text{int}} \rightarrow 1.81818 \times 10^{-10}\}$ 
```

Timing

Design choice: Calculate the maximum allowable pattern time, to stay below V_{th} by 0.1V (we want the integrator to stabilize around the zero position):

```
In[9]:= Dynamic[pTime = NSolve[- $\frac{1}{r_{\text{Mes}} * c_{\text{int}}} * v_{\text{inFS}} * t_{\text{pattern}} - \frac{1}{r_{\text{RefNeg}} * c_{\text{int}}} * v_{\text{refNeg}} * 0.8 * t_{\text{pattern}} - \frac{1}{r_{\text{RefPos}} * c_{\text{int}}} * v_{\text{refPos}} * 0.1 * t_{\text{pattern}} == -2 * v_{\text{th}} - 1 + 0.1, t_{\text{pattern}}] /. capacitor]
Out[9]=  $\{t_{\text{pattern}} \rightarrow 0.0000317091\}$$ 
```

Compute the minimum allowable number of patterns within one runup/measurement period, due to the just computed pattern time:

```
In[10]:= Dynamic[nPatterns = Ceiling[ $\frac{t_{\text{runup}}}{t_{\text{pattern}}}$ ]]
Out[10]=  $\{32\}$ 
```

Calculate some more resulting timing figures for the positive and negative reference voltage slopes during multislope runup, and the minimum time length of the system state, regarding the switch and the integrator:

```
In[11]:= Dynamic[ $\{t_{\text{Pos}} = \frac{t_{\text{runup}}}{n_{\text{Patterns}}} * 0.8, t_{\text{Neg}} = \frac{t_{\text{runup}}}{n_{\text{Patterns}}} * 0.8, t_{\text{min}} = \frac{t_{\text{runup}}}{n_{\text{Patterns}}} * 0.1\} /. pTime]
Out[11]=  $\{\{0.000025\}, \{0.000025\}, \{3.125 \times 10^{-6}\}\}$$ 
```

Design choice: At least 10 clock cycles for the shortest time interval, which is 10% of a pattern time with the 3458A patterns (t_{min}):

All operations on the FPGA must complete within one clock cycle. The minimum required **Clock frequency** is:

```
In[25]:= Dynamic[ $\{f_{\text{min}} = \frac{10}{t_{\text{min}}} /. pTime, T_{\text{freq}} = \frac{1}{f_{\text{min}}}\}]
Out[25]= \{3.2 \times 10^6, 3.125 \times 10^{-7}\}$ 
```

Required ADC resolution

```
In[13]:= Dynamic[adcVolts =  $\frac{-v_{\text{runupFS}}}{2500000}$ ]
Out[13]=  $\frac{11}{10000}$ 
```

Compute the minimum required number of **ADC counts**

```
In[14]:= Dynamic[Ceiling[adcCounts =  $\frac{2 * v_{\text{sat}}}{adc_{\text{volts}}}\right]\]
Out[14]= 20000$ 
```

Add some margin

```
In[15]:= Dynamic[adcRealCounts = adcCounts * 1.25]
```

```
Out[15]= 25000.
```

```
In[16]:= {2^12, 2^14, 2^16, 2^18}
```

```
Out[16]= {4096, 16384, 65536, 262144}
```

Calculate the maximum number of adc samples possible with our 1.6MSPS ADC @ 625ns per Sample:

```
In[17]:= Dynamic[adcSamples =  $\frac{t_{\min}}{0.000000625}$ ]
```

```
Out[17]= {{5.}}
```

Theoretically gained resolution in bit and reduction in noise power / increase in SNR by a factor of:

```
In[18]:= Dynamic[{resgained =  $\frac{\log[\text{adcSamples}]}{2 \log[2]}$ , noisePower =  $\frac{1}{\text{adcSamples}}$ }]
```

```
Out[18]= {{{{1.16096}}}, {{{0.2}}}}
```

Compute the component values and constraints

Calculate the **slew rate** of the integrator opamp. The maximum slewrate is observed, when equal polarity input and reference signals are switched in:

```
In[19]:= Dynamic[slewrateVμs =  $\frac{1}{r_{\text{Mes}} * C_{\text{int}}} * v_{\text{inFS}} * 0.000001 + \frac{1}{r_{\text{RefPos}} * C_{\text{int}}} * v_{\text{refPos}} * 0.000001 /. \text{capacitor}]$ 
```

```
Out[19]= {6.1875}
```

Calculate the **settling time** - since we measure the second voltage for ΔV_{adc} at the end of the last pattern, the opamp has to settle to less than 0.1 least significant digits after the last switch within:

```
In[20]:= Dynamic[tSettle =  $\frac{t_{\text{runup}}}{n_{\text{Patters}}} * 0.1$ ] /. ptime
```

```
Out[20]= {{{{3.125 \times 10^{-6}}}}}
```

```
In[21]:= Dynamic[vSettle = adcVolts * 0.1] /. ptime
```

```
Out[21]= {{0.00011}}
```

Design choice: The maximum **current leakage** into the opamp should be less than 5 least significant counts (drift of bias current might become relevant with temperature):

```
In[22]:= Dynamic[iErrorOpAmpMax = iInFS * 0.0000005]
```

```
Out[22]= 2.5 \times 10^{-10}
```

Compute the maximum acceptable phase noise / **timing uncertainty** to not affect the measurement more than 0.75 of a least significant digit on a full-scale input:

First, define the voltage error as a function of the jitter/phase noise and the ΔV_{ADC}

```
In[23]:= VerrJitter[tJitter_, ΔvADC_] :=
  (- (rMes ((tNeg + tJitter) rRefPos vrefNeg + rRefNeg ((tNeg + tJitter) vrefPos +
    Cint rRefPos ΔvADC)))) / (rRefNeg rRefPos (trunup + tJitter)) +
  (rMes (tNeg rRefPos vrefNeg + rRefNeg (tPos vrefPos + Cint rRefPos ΔvADC))) /
  (rRefNeg rRefPos trunup)) /. capacitor;

In[24]:= Dynamic[{NSolve[VerrJitter[tJitter, 13] == 0.00000075, tJitter],
  NSolve[VerrJitter[tJitter, 13] == 0.0000001, tJitter]}]

Out[24]= {{tJitter → 6.34656 × 10-8}, {tJitter → 8.46161 × 10-9}}
```