

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]=  $\left\{ \left\{ V_{\text{in}} \rightarrow \frac{v_{\text{inFS}} (i_{\text{refFS}} (t_{\text{Neg}} - t_{\text{Pos}}) - C_{\text{int}} \Delta v_{\text{ADC}})}{i_{\text{inFS}} \Delta t_{\text{int}}} \right\} \right\}$ 
```

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; trunup = 0.001; vrunupFS = -5000; nPatters = 100;
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.}
```

Timing

Calculate the mean time a reference is swichted in, with the 3458A patterns. 10% at each end is without any v_{ref} activated, so only use 80% of the pattern time

```
In[8]:= Dynamic[ {tPattern = tRunup / nPatters, tPos = tRunup / nPatters * 0.8, tNeg = tRunup / nPatters * 0.8} ]  
Out[8]= {0.00001, 8. × 10-6, 8. × 10-6}
```

Design choice: At least 100 clock cycles for the shortest time interval, which is 10% of a pattern time with the 3458A patterns:

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

```
In[9]:= Dynamic[ fMin = 10 / (tRunup / nPatters * 0.10) ]  
Out[9]= 1. × 107
```

Required ADC resolution

```
In[10]:= Dynamic[ adcVolts = -vRunupFS / 2500000 ]  
Out[10]= 1  
500
```

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

```
In[11]:= Dynamic[ adcCounts = 2 * vSat / adcVolts ]  
Out[11]= 11000
```

Add some margin

```
In[12]:= Dynamic[ adcRealCounts = adcCounts * 1.25 ]  
Out[12]= 13750.
```

```
In[13]:= {2^14, 2^16, 2^18}  
Out[13]= {16384, 65536, 262144}
```

Compute the component values and constraints

Integration Capacitor: With the maximum slope (V_{in} positive and negative reference switched in), the integrator shall rise to 70% V_{sat} within 45% of the time the reference is switched in:
old and wrong:

```
capacitor = NSolve[  
 1 / (rMes * cInt) * vInFS * (tPos * 0.45) + 1 / (rRefPos * cInt) * vRefPos * (tPos * 0.45) == (0.70 * vSat),  
 cInt]
```

```
In[14]:= Dynamic[ capacitor = NSolve[ 1 / (rMes * cInt) * vInFS * tRunup == -vRunupFS, cInt] ]  
Out[14]= {cInt → 1. × 10-10}
```

Calculate the **slew rate** of the rising signal with fullscale input and corresponding reference:

```
In[15]:= Dynamic[slewrateVus =
  1/(rMes * Cint) * vInFS * 0.000001 + 1/(rRefPos * Cint) * vrefPos * 0.000001 /. capacitor]
Out[15]= {11.25}
```

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[16]:= Dynamic[tSettle = tPattern * 0.1]
Out[16]= 1. \times 10^{-6}
```

```
In[17]:= Dynamic[vSettle = adcVolts * 0.1]
Out[17]= 0.0002
```

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[18]:= Dynamic[iErrorOpAmpMax = iInFS * 0.0000005]
Out[18]= 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[19]:= 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[20]:= Dynamic[{NSolve[VerrJitter[tJitter, 13] == 0.00000075, tJitter],
  NSolve[VerrJitter[tJitter, 13] == 0.0000001, tJitter]}]
Out[20]= {{tJitter \rightarrow 1.15398 \times 10^{-7}}, {{tJitter \rightarrow 1.53849 \times 10^{-8}}}}
```