

Voltage Integrator Calculation

Definitions and Equations - Multislope Continuous Runup with periodic ADC measurements

```
In[1]:= voutIn = -  $\frac{1}{r_{Mes} * C_{int}}$  * vin * Δtint;  
In[2]:= voutRefPos = -  $\frac{1}{r_{RefPos} * C_{int}}$  * vrefPos * tPos;  
In[3]:= voutRefNeg = -  $\frac{1}{r_{RefNeg} * C_{int}}$  * vrefNeg * tNeg;
```

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

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

Component value calculations and some error calculation

Design Choices:

```
In[48]:= iinFS = 0.0005; vinFS = 2.5; vrefPos = 5.0; vrefNeg = -5.0;  
vsat = 11; vth = 5; trunup = 0.001; vrunupFS = -5000;  
In[6]:= irefFS = iinFS * 1.25;
```

Resistor Values:

```
Dynamic[ { rMes =  $\frac{V_{inFS}}{i_{inFS}}$ , rRefPos =  $\frac{V_{refPos}}{i_{refFS}}$ , rRefNeg =  $\frac{-V_{refNeg}}{i_{refFS}}$  } ]  
{5000., 8000., 8000.}
```

Integration Capacitor:

```
In[21]:= Dynamic[ capacitor = NSolve[  $\frac{1}{r_{Mes} * C_{int}}$  * vinFS * trunup == -vrunupFS, Cint ] ]  
Out[21]= { {Cint → 1. × 10-10} }
```

Timing

Calculate the maximum allowable pattern time, to not overshoot V_{th} by more than 0.5V (we want the integrator to stabilize around the zero position)

```
In[56]:= Dynamic[ptime = NSolve[-(1/(rMes * Cint) * vInFS * tPattern) - (1/(rRefNeg * Cint) * vRefNeg * 0.8 * tPattern) - (1/(rRefPos * Cint) * vRefPos * 0.1 * tPattern) == -2 * (vTh + 0.25), tPattern]] /. capacitor
```

Out[56]= $\{ \{ t_{pattern} \rightarrow 0.0000168 \} \}$

Compute the minimum allowable number of patterns, due to the pattern time:

```
In[68]:= Dynamic[nPatterns = Ceiling[tRunup / . ptime]]
```

Out[68]= {60}


```
In[63]:= Dynamic[{tPos = tRunup / . nPatterns * 0.8, tNeg = tRunup / . nPatterns * 0.8, tMin = tRunup / . nPatterns * 0.1}] /. ptime
```

Out[63]= $\{ \{ 0.0000133333, 0.0000133333, 1.66667 \times 10^{-6} \} \}$

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[64]:= Dynamic[fMin = 10 / . ptime]
```

Out[64]= $\{ 6. \times 10^6 \}$

Required ADC resolution

```
In[11]:= Dynamic[adcVolts = -vRunupFS / 2500000]
```

Out[11]= $\frac{1}{500}$

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

```
In[12]:= Dynamic[adcCounts = 2 * vSat / adcVolts]
```

Out[12]= 11 000

Add some margin

```
In[13]:= Dynamic[adcRealCounts = adcCounts * 1.25]
```

Out[13]= 13 750.

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

Out[14]= {16384, 65536, 262144}

Compute the component values and constraints

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

```
In[15]:= Dynamic[slewrateVus =
  
$$\frac{1}{r_{\text{Mes}} * c_{\text{int}}} * v_{\text{inFS}} * 0.000001 + \frac{1}{r_{\text{RefPos}} * c_{\text{int}}} * v_{\text{refPos}} * 0.000001 / . \text{capacitor}]$$

Out[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[67]:= Dynamic[t_settle =  $\frac{t_{\text{runup}}}{n_{\text{Patters}}} * 0.1$ ] /. ptime
Out[67]= {{1.66667 \times 10^{-6}}}
```

```
In[34]:= Dynamic[v_settle = adcVolts * 0.1] /. ptime
Out[34]= {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[i_errorOpAmpMax = i_inFS * 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[65]:= VerrJitter[tJitter_, ΔvADC_] :=
  
$$(- (r_{\text{Mes}} ((t_{\text{Neg}} + t_{\text{Jitter}}) r_{\text{RefPos}} v_{\text{refNeg}} + r_{\text{RefNeg}} ((t_{\text{Neg}} + t_{\text{Jitter}}) v_{\text{refPos}} +
    c_{\text{int}} r_{\text{RefPos}} \Delta v_{\text{ADC}})))) / (r_{\text{RefNeg}} r_{\text{RefPos}} (t_{\text{runup}} + t_{\text{Jitter}})) +
  (r_{\text{Mes}} (t_{\text{Neg}} r_{\text{RefPos}} v_{\text{refNeg}} + r_{\text{RefNeg}} (t_{\text{Pos}} v_{\text{refPos}} + c_{\text{int}} r_{\text{RefPos}} \Delta v_{\text{ADC}}))) /
  (r_{\text{RefNeg}} r_{\text{RefPos}} t_{\text{runup}}) / . \text{capacitor};$$

In[66]:= Dynamic[{NSolve[VerrJitter[tJitter, 13] == 0.00000075, tJitter],
  NSolve[VerrJitter[tJitter, 13] == 0.0000001, tJitter]}]
Out[66]= {{tJitter \rightarrow 1.15398 \times 10^{-7}}}, {{tJitter \rightarrow 1.53849 \times 10^{-8}}}
```