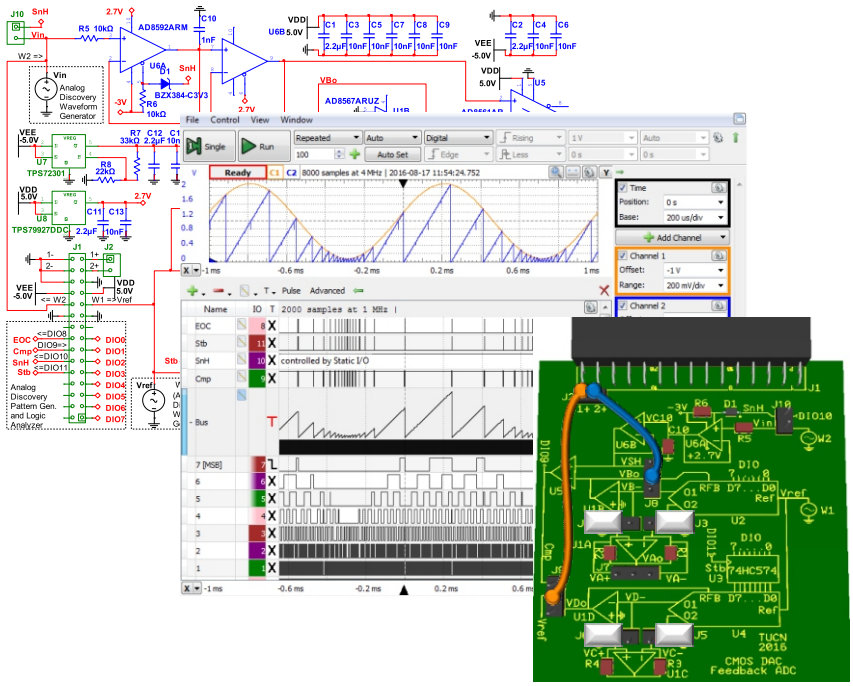
Data Acquisition Systems Fundamentals

Applications

MIRCEA ALEXANDRU DĂBÂCAN
LIVIU MARIN VIMAN
VLAD BANDE



UTPRESS
Cluj-Napoca, 2022
ISBN 978-606-737-605-0

Mircea DĂBÂCAN
Liviu Marin VIMAN
Vlad BANDE

Data Acquisition Systems Fundamentals

Applications

UTPRESS
Cluj - Napoca, 2022
ISBN 978-606-737-605-0

Editura UTPRESS
Str. Observatorului nr. 34
400775 Cluj-Napoca
Tel.: 0264-401.999
e-mail: utpress@biblio.utcluj.ro
http://biblioteca.utcluj.ro/editura

Director: ing. Dan Colțea

Recenzia: Conf.dr.ing. Septimiu Sever Pop
Șl.dr.ing. Ionel Horea Baciu

Pregătire format electronic: Gabriela Groza

ISBN 978-606-737-605-0
Bun de tipar: 06.12.2022

1 Sampling Theory

1.1 Background

1.1.1 Signal Spectrum

A periodical signal, of period *T*, can be decomposed in spectral components, as in the Fourier series (1.1):

$$f(t) = \sum_{n=-\infty}^{\infty} c_n \cdot e^{jn\omega t} \qquad \textbf{(1.1)}$$

The Fourier coefficients are:

$$c_n = \frac{1}{T} \cdot \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) \cdot e^{-jn\omega t} dt \qquad \textbf{(1.2)}$$

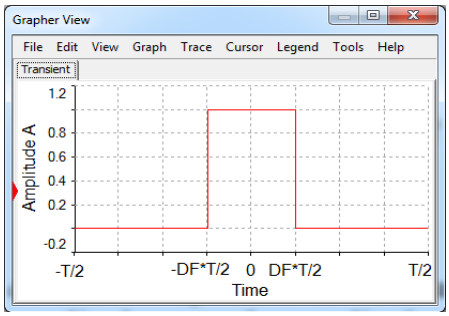


Figure 1.1 Rectangular pulse train with amplitude *A*, period *T*, duty factor *DF*

c_0 = DC component; c_1 = fundamental amplitude; c_n = the amplitude of the *n*-th harmonic.

For a rectangular signal of period *T* and duty factor *DF*, as in Figure 1.1, the Fourier coefficients are:

$$c_0 = A \cdot DF \qquad \textbf{(1.3)}$$

$$c_n = \frac{1}{T} \cdot \int_{-DF \cdot \frac{T}{2}}^{DF \cdot \frac{T}{2}} A \cdot e^{-jn\omega t} dt = \frac{A}{jn2\pi} \cdot \left(e^{jn\pi \cdot DF} - e^{-jn\pi \cdot DF}\right) = \qquad \textbf{(1.4)}$$
$$= A \cdot DF \cdot \frac{sin(n \cdot \pi \cdot DF)}{n \cdot \pi \cdot DF} = A \cdot DF \cdot sin\, c\,(n \cdot \pi \cdot DF)$$

As evident in (1.4) and Figure 1.2, all spectral components are covered by a *sinc* type function. Furthermore, *n=k/DF* harmonics are missing in the spectrum. This is:

- even harmonics for DF = 50%
- 3^{rd}, 6^{th}, 9^{th}, etc. for DF = 1/3.
- 5^{th}, 10^{th}, 15^{th},, etc. for DF = 1/5.
- 10^{th}, 20^{th}, 30^{th}, etc. for DF=1/10 or 9/10 (notice DC component = DF*A).

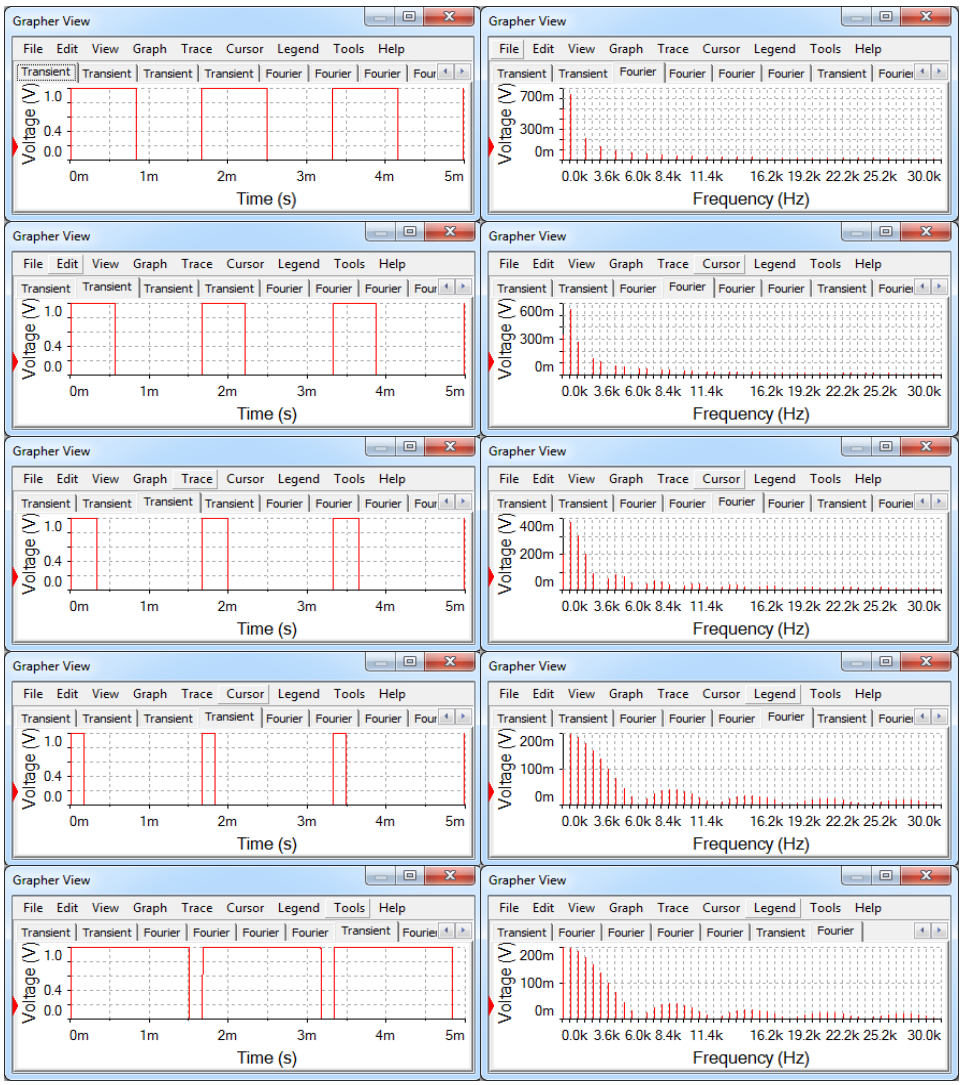


Figure 1.2 Left: 600Hz rectangular signals with various duty factors (from top to down: 1/2, 1/3, 1/5, 1/10, 9/10).

1.1.2 Sampling Rate

Analog signals are continuous, both in "amplitude" and "time" domains. Converting them to digital implies discretizing in both domains.

The amplitude discretization is called quantification. The conversion resolution is the smallest change in the analog value which modifies the digital value.

The "time" discretization is sampling, meaning that the signal is represented by a sample string. Uniform sampling uses constant the time interval between subsequent samples:

$$T_{sampling} = 1/f_{sampling} \quad \textbf{(1.5)}$$

The sampling frequency must obey the sampling theorem (Nyquist):

$$f_{sampling} \geq 2 \cdot f_{max} \quad \textbf{(1.6)}$$

where f_{max} is the highest frequency in the input signal spectrum.

Under-sampling means using a sampling frequency below that allowed by the sampling theorem, as in Figure 1.3, up. The collected samples do not correctly represent the signal of which were extracted.

Sampling a signal at the Nyquist limit frequency, still does not allow the rebuild of the initial signal. In Figure 1.3, middle, the input signal is sinusoidal, having a single spectral component. Two samples are

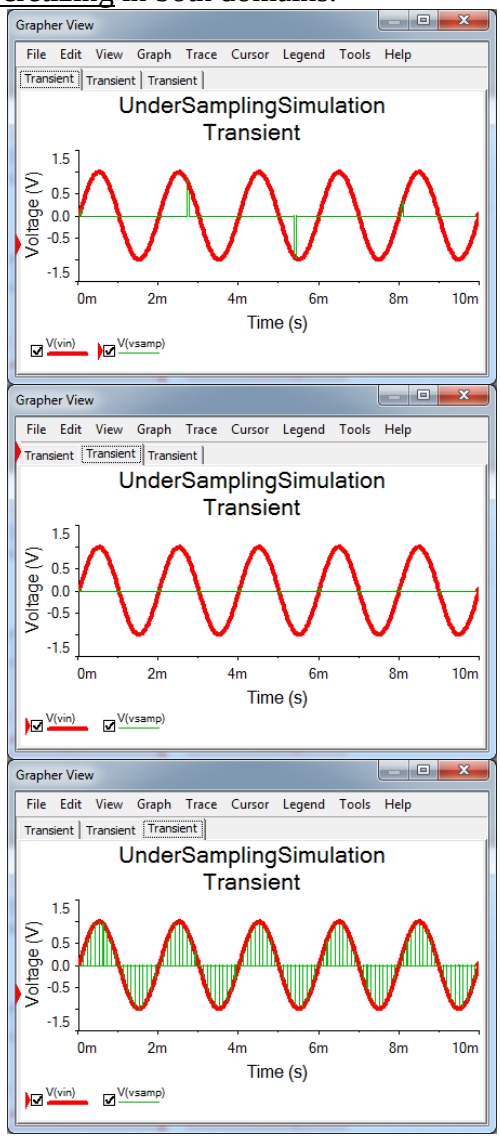


Figure 1.3 Under-sampling (up), sampling at Nyquist limit (middle) and oversampling (down) a sinusoidal signal

collected each period. Depending on the phase shift between the sinus signal and the sampling times, the sampled values are different. The extreme cases are pecking samples at:

- "zero" points of the input signal (as in Figure 1.3, middle).
- maximum, respectively minimum points.

Any intermediate situation is possible.

Choosing a sampling frequency (much) higher than required by the sampling theorem is called over-sampling. The resulting samples represent more correct the original input signal as the over-sampling rate increases. Figure 1.3, down, illustrates a sinus signal over-sampled by a factor of 10 (10 time more than required by sampling theorem, which is 20 times the highest frequency in the spectrum).

1.1.3 Alias

The sampled signal spectrum includes the original signal spectrum and mirror images of it around sampling frequency and its integer multiples.

Alias phenomenon occurs at under-sampling and consists in overlapping the mirror spectrum (most likely the mirror image built around the fundamental frequency) to the original signal spectrum. "Recovering" the initial signal spectrum out of the sampled signal one is impossible in such conditions, so rebuilding of the initial signal cannot be anymore correct.

For infinite spectrum signals (i.e.: rectangle, triangle, etc.) it is impossible to choose a sampling frequency (∞) to obey the sampling theorem. It is preferable to limit the initial signal spectral bandwidth, using an anti-alias (low pass) filter.

Figure 1.4 (up) shows 600Hz input signals and sampled signals with 10kHz, 10% rectangular pulses. The middle graphs show the spectra of the input signals, and the lower graphs show the spectra of the sampled signals. The left figures use a rectangular input signal while the right-side figures use a filtered rectangle, keeping only the fundamental, the 3^{rd} and 5^{th} harmonics.

Notice that the vertical scale in spectrum diagrams is linear, in [V].

In the left-side graphs, the sampled signal spectrum overlaps the mirrored spectral images to the initial one. Rebuilding the initial signal by filtering the sampled one is impossible, even using an ideal filter.

In the right-side graphs, the rectangular signal was anti-alias filtered, cutting out harmonics above the 5th. The filtering changes the initial signal shape, as shown in the upper-right graph, but avoids alias. The signal can be rebuilt out of its samples, using an output filter identical to the anti-alias one. The errors induced by filtering the input signal are less than the avoided alias ones.

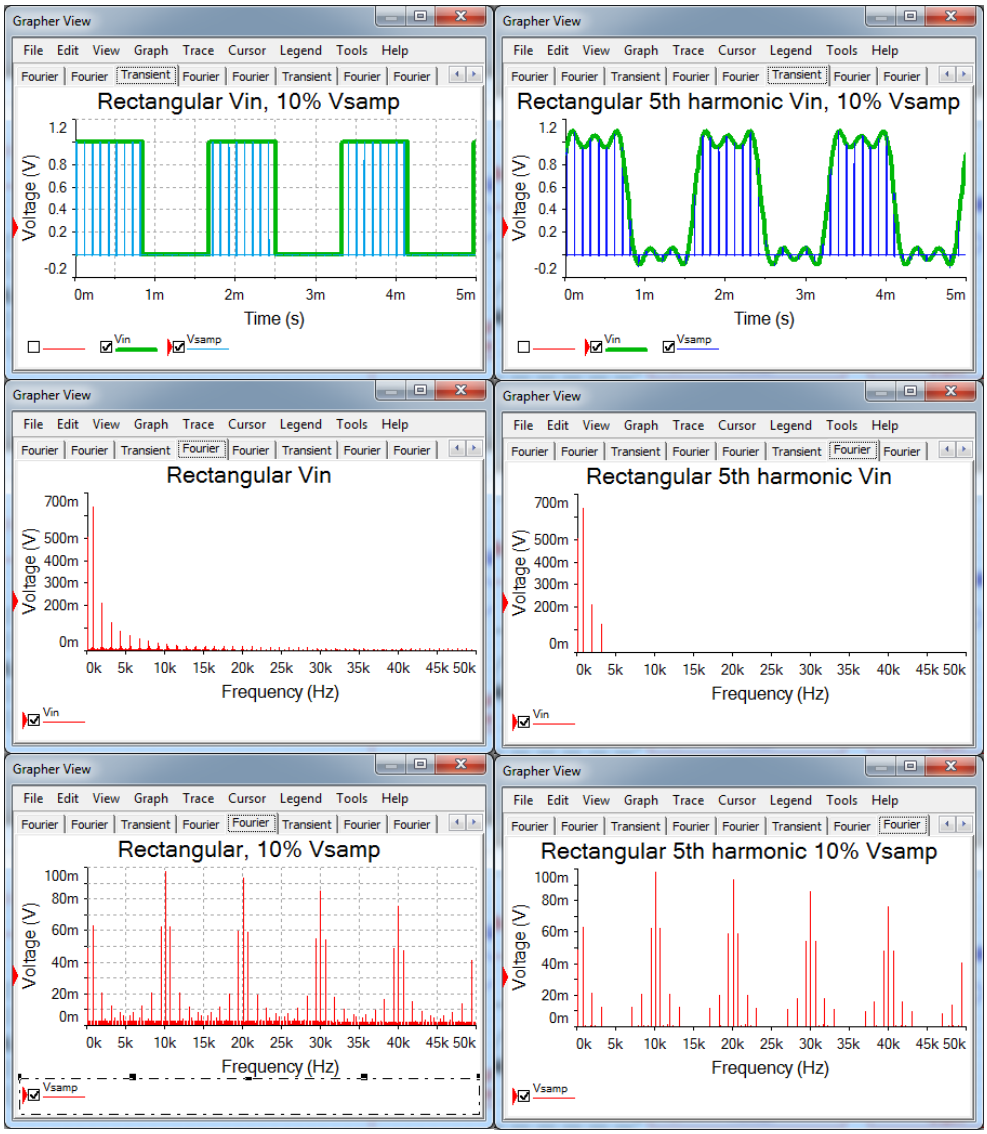


Figure 1.4 Sampling signals and spectra:

- 600Hz rectangular (left) and filtered rectangular (right)
- Time domain (up), original spectrum (middle) and sampled (down)

The ideal sampling is multiplying the input signal to a periodical string of Dirac pulses, repeating at the sampling frequency. A Dirac pulse has a time length of *0*, infinite amplitude and closes surface equal to unity. The sampled signal keeps same energy as the input signal: the sample pulses are infinitesimal short in time (*0* at limit), but very high amplitude (∞ at limit).

Real sampling circuits peck the instantaneous value of the input signal at the sampling time. That corresponds to multiplying the input signal by a string of periodical pulses, at sampling frequency, with time length as short as possible (but finite) and finite amplitude. The real sampled signal has much lower energy than the initial, analog one.

The resulting sample values are quantified, during the analog-to-digital conversion. The digital signal representation is a string of successive numbers, which can be transmitted, stored, and/or processed, taking the advantages of digital technology (speed, density, computing power, error immunity).

If the analog signal needs to be rebuilt, the digitized string, is converted back to analog and then filtered with an output filter as shown above. Before filtering, the samples are interpolated, for getting back the time continuity and for recovering the energy lost when sampling. Mostly used, the 0-degree (polynomial) interpolation holds the value of a sample until the next sample is available, similar to a "sample-and-hold" circuit.

Interpolation itself acts as *low pass filtering*, as seen in Figure 1.5 (compared to the lower-right image in Figure 1.4), the original spectral components recovered the energy.

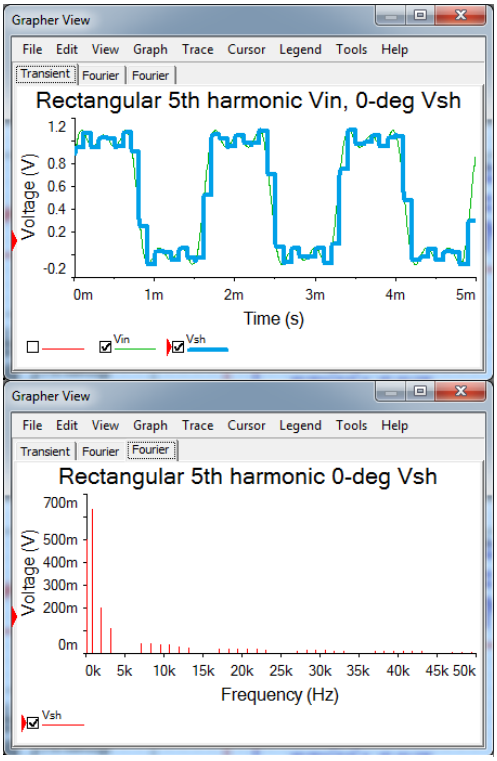


Figure 1.5 0-degree interpolated signal (up) and spectrum (down)

1.2 Experiment

Figure 1.6 shows the full schematic of the experimental board, but circuits not used in the current experiment are shaded.

U7 and U8 are linear voltage regulators, providing *-3V* and *+2.7V*, since AD8592 supports maximum *6V* differential supply voltage.

Pin 5 of AD8592 is the *Shut Down* signal for the amplifier A; when *LOW*, the output of the amplifier goes *HiZ*, insulating C_{10}, for the *Hold* period. D_1 and R_6 translate the 5V logic signal SnH to the *(-3V...+1.7V)* range, compatible with the *Shut Down* signal requirements.

The board is designed to directly connect to the Digilent Analog Discovery. Channel 2 of the *AWG* (*W2*) generates the input signal, V_{in}. The digital pin *DIO10* generates *SnH*. The scope channels 1+ and 2+ are accessible in J2, for probing V_{in} (J10-2) and V_{SH} (J8-1).

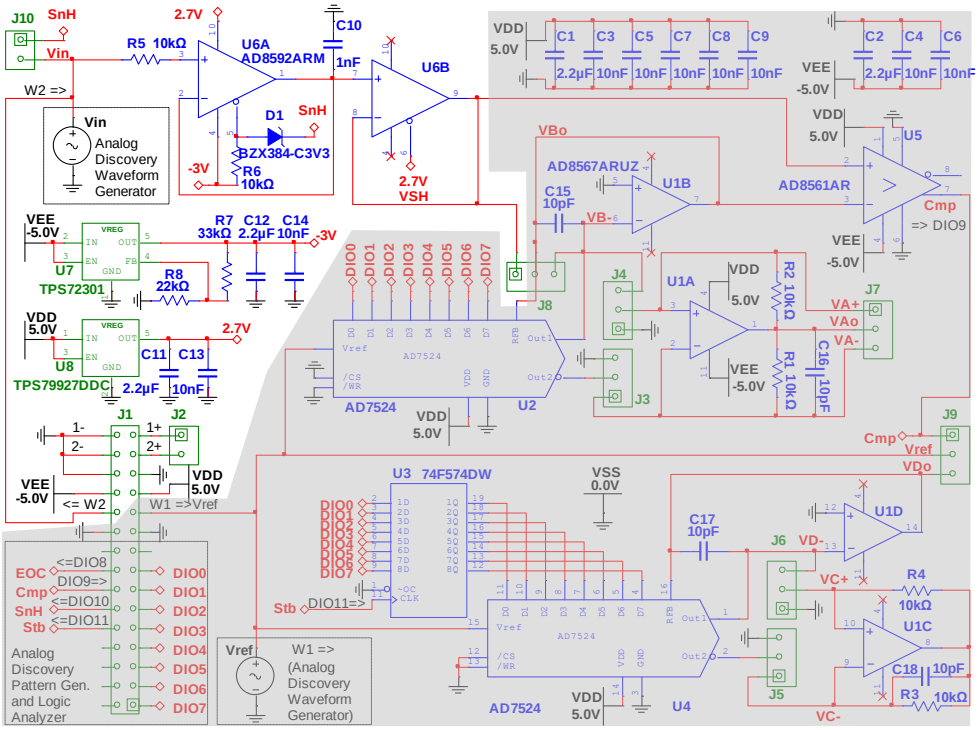


Figure 1.6 Sample and Hold experimental board schematic

Figure 1.7 shows the experimental board prepared with scope probe wires for the Sampling experiments. Jumpers on J3-J6 are irrelevant for these experiments. Notice that the same board is used for several other experiments.

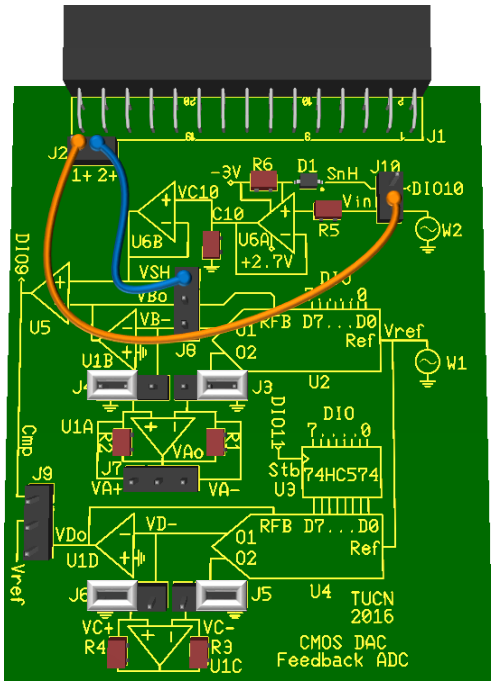


Figure 1.7 S&H experimental board

1.2.1 Signal Spectrum Experiment

Set the WaveForms instruments:

- *WaveGen* Channel 2: *Square, 600Hz, Amplitude: 0.5V, Offset: 0.5V, Symmetry: 50%*
- *Scope*:
  - Set Time, Channels and Trigger as in Figure 1.8.
  - View/FFT.
    - Click the green arrow, to see advanced options.
    - Set FFT as in Figure 1.8. Notice the settings effect.
    - Change time base and notice the change in the FFT view. The FFT needs a large number of periods in the time view (acquisition buffer) for accurate spectrum computation. At the resulting time base, the time view is difficult to read. A Zoom view is needed.
  - View/AddZoom.
    - Hoover mouse over the zoom window to highlight the zoomed area in the time view.
    - Click (left or right) and drag on the horizontal or vertical axes, to move, respectively zoom in/out. Notice the changes in the Zoom window and of the highlighted area in the main time view.
- You can undock and dock the FFT and Zoom windows.
- You can undock/dock instruments, clicking the graphical symbol in the upper-right corner of the instrument tab.

Modify the *Duty Factor* in *WaveGen*. Observe the spectrum, similar to Figure 1.2.

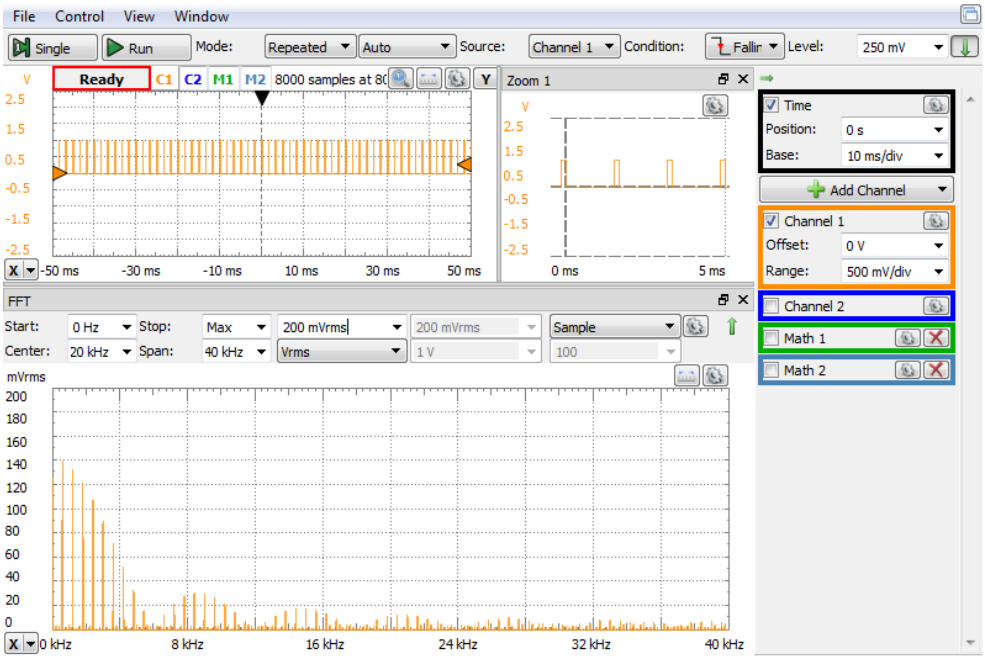


Figure 1.8 Rectangular signal: DF = 10%

1.2.2 Sampling rate experiment

- *WaveGen* Ch 2: *Sinus, 500Hz, amplitude: 1V, offset: 0V, symmetry: 50%.*
- *Patterns1:*
  - *Add/Signal/*DIO10/*Add*
  - *Output: PP (Push-Pull)*
  - *Type: Clock*
  - *Edit parameters*
    - *Frequency: 370Hz,*
    - *Duty: 1%*
    - *Phase: 0º*
- *Patterns2:* WaveForms software allows multiple instances of the same instrument. A single one can *Run* at a time; all others are *Busy*. From Patterns1:
  - *File/NewPatterns/Clone*
  - Change *Frequency: 1kHz.*
- *Patterns3*: set *Frequency*: *10kHz.*

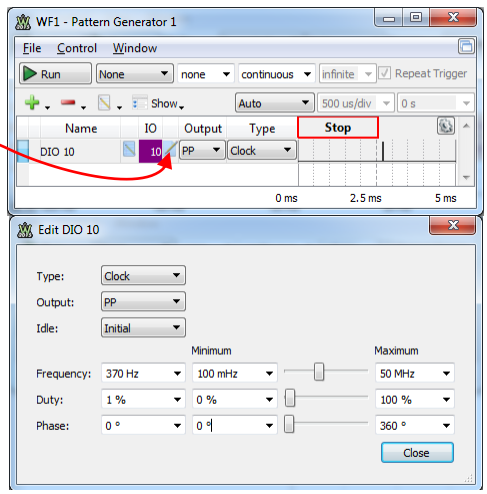


Figure 1.9 Setting DIO10 as clock

- Power *Supplies*: ON, +5V and -5V (If WaveForms reports an "Overcurrent condition" and stops the user power supplies, try starting the supplies one by one: first the positive supply and, after few seconds, the negative one).
- *Scope*:
  - Set Time, Channels an Trigger as in Figure 1.10
  - AddChannel/Digital/Signal: DIO10
  - View/FFT.
    - Click the green arrow, to see advanced options.
    - Set FFT as in Figure 1.11. Notice the settings effect.
    - Change time base for convenient image in the FFT view.
  - View/AddZoom.
    - Hoover mouse over the zoom window to highlight the zoomed area in the time view.
    - Click (left or right) and drag on the horizontal or vertical scales, to move, respectively zoom in/out. Notice the changes in the Zoom window and of the highlighted area in the main time view.

Run Patterns 1 for under-sampling (f_{samp}*=370Hz*), Patterns 2 for sampling at Nyqusit limit (f_{samp} *=1kHz*) respectively Patterns 3 for oversampling (f_{samp} *=10kHz*). Run Scope in Single mode.

Observe in the time domain that only oversampling allows rebuilding the signal from the samples (higher oversampling rate allows better reconstruction).

Observe in the frequency domain that under-sampling produces frequency components lower than the sampled signal, so these cannot be removed by a LPF to rebuild the original signal. Sampling at Nyquist limit produces a rectangle of same frequency as the input signal, but the amplitude is random (depends on the phase shift between input and sampling signals). Oversampling replicates the original signal spectrum and adds mirror images (direct and reflected) around sampling frequency and integer multiples of it.

Ideally, the rebuild filter needs to keep all frequencies in the input signal spectrum and reject all the mirror images. Higher sampling frequency and lower input frequency range both increase the frequency distance between the highest frequency to keep and the lowest frequency to reject, making the filter task easier.

Also notice that interpolation (even the simple 0-degree) accentuated the original signal spectrum over the mirror images. (A better interpolation would do even more of that, with the cost of complexity, delay and stability).

Notice that the magnitude of the spectral components is represented at logarithmic scale in Figure 1.11. Chang units from db_V to V_{rms} to see the linear ratio between original spectral components and mirror ones.

With the logarithmic amplitude scale, many other (residual) spectral lines are visible. These have various origins:

- Mixing products (of type $(f_0 \pm f_1)/2$) between the input signal (pure) frequency and all the harmonics of the sampling signal.
- Harmonic distortion due to the finite time of the "*sample*" phase in the *sample-and-hold* circuit. (ideally, "*sample*" time should be null, and the rebuilt signal should be a series of rectangles, without any moment to "follow" the input signal)
- Harmonic distortion of the physical circuits in the schematic.
- Measuring system sampling. The Analog discovery acquires the signals by sampling. The maximum frequency able to be shown in the FFT window is half of the sampling frequency. The Analog Discovery input stage bandwidth is limited, but no anti-alias filtering is done. Negative mirror images reflected by the acquisition sampling process are visible in the FFT view.

Figure 1.10 Under-sampling a sine signal

- Noise.

All these components define the THD (Total Harmonic Distortion) of the acquisition system. They are only visible on logarithmic scale, but insignificant on the linear scale.

Even small changes in the sampling frequency or duty factor significantly modify the residual spectral components. Edit *DIO10* in the Patterns. Set close *Min/Max* limits for *Frequency* (9kHz…11kHz) and *Duty Factor* (0%...2%). Drag cursors within these close limits and observe the effect on the FFT image.

1.2.3 Alias experiment

Set WaveForms instruments:

- *WaveGen1* Ch 2: *Rectangle, 600Hz, amplitude: 1V, offset: 0V, symmetry: 50%.*

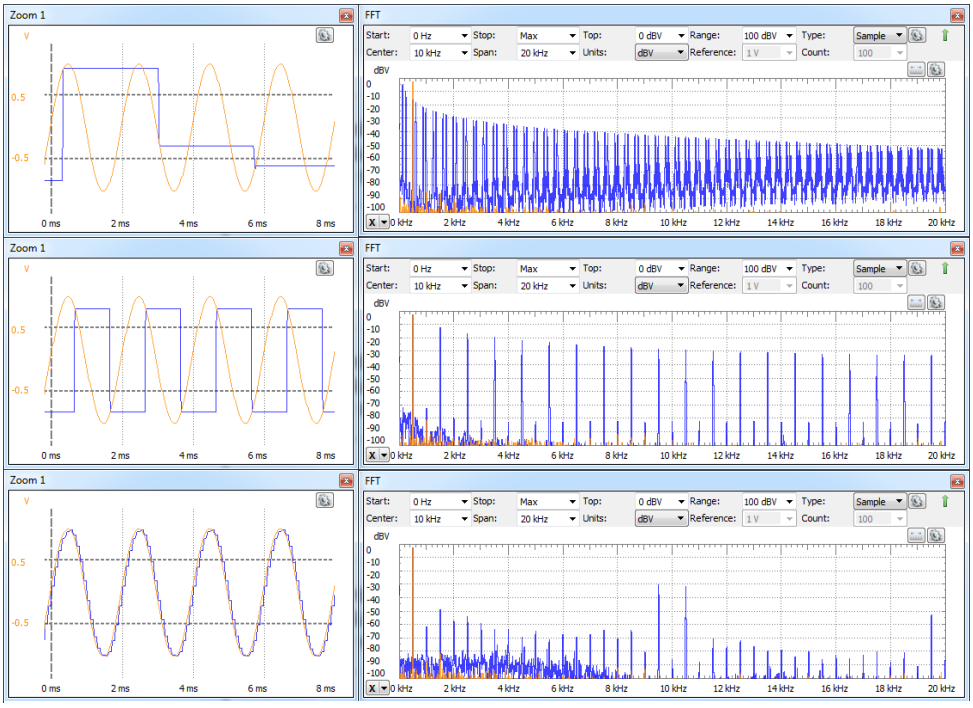


Figure 1.11 Sampling and rebuilding a signal with 0-degree interpolation: Under sampling (up), sampling at Nyquist limit (middle) and oversampling (down). Time views (left) and spectra (right)

- Use *Spectrum* analyzer, to measure the relative amplitudes of fundamental, 3rd and 5th harmonics of the WaveGen1 signal, as in Figure 1.12: fundamental @ ~ 600Hz, 3rd harmonic @ ~ 1800Hz, 5th harmonic @ ~ 3000Hz:
    - View/Components/T1
    - Units: Vrms
    - Top: 1Vrms
- *From WaveGen1: File/NewWaveGen/Empty*:
- *WaveGen2* Ch 2: Filtered version of the WaveGen1 signal: only the fundamental, the 3rd and 5th harmonics are kept. The signal is generated in the custom mode. After measuring the amplitude of the fundamental, 3rd, 5th harmonics in the FFT view of WaveGen1 signal, use them as coefficients in a Math Custom synthesized signal:
    - *Custom/Edit/Math*
    - 0.901*sin(2*PI*X) +0.301*sin(2*3*PI*X)+0.181*sin(2*5*PI*X)
    - *Generate*
    - *Save*
    - *Frequency*: 600Hz (Sample Rate is calculated automatically)
    - *Amplitude*: 1V
    - *Offset*: 0V
    - *Phase*: 0o

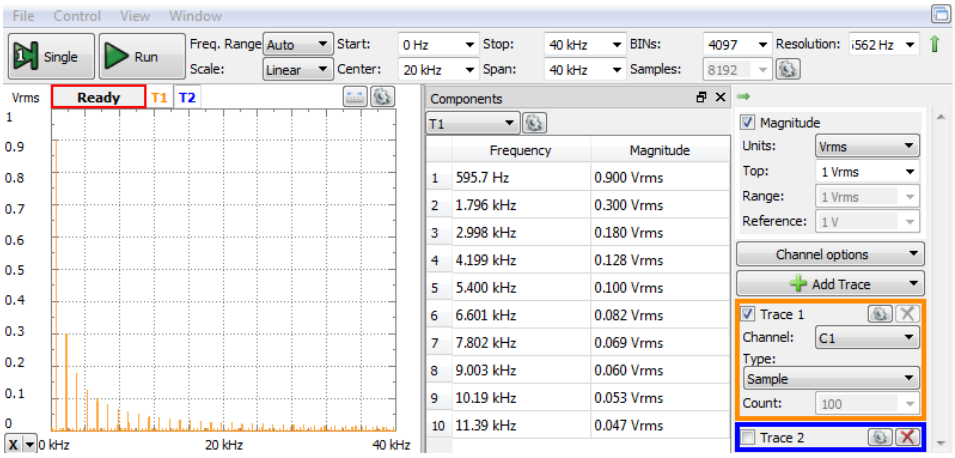


Figure 1.12 Reading the amplitudes of spectral components

- *Patterns1:*
    - *Add/Signal/*DIO10/*Add*
    - *Output: PP* (*Push-Pull)*
    - *Type: Clock*
    - *Edit parameters: Frequency: 10kHz, Duty: 10%, Phase: 0º*
- Power *Supplies*: ON, +5V and -5V
- *Scope*:
    - Set Time, Channels a Trigger as in Figure 1.10
    - View/FFT.
    - View/AddZoom; adjust zoom view as convenient.

The alias experiment is conducted in two versions:

a. With no interpolation (Figure 1.13) – closer to ideal sampling. Samples are 10µs short, with null value between samples (90µs). Scope channel 1 probes V_{in}, channel 2 probes *SnH*. The sampled signal is computed by *Math1* (AddChannel/Math-Custom) as product of V_{in} by *SnH* (divided by the voltage amplitude of *SnH* signal): *Math1 = C1 * C2 /3.3*.

b. With 0-degree interpolation (Figure 1.14) – real "sample-and-hold, closer to real reconstruction of the samples. The V_{SH} signal repeats V_{in} for 10µs, then keeps the last value for the next 90µs. Scope channel 1 probes V_{in}, channel 2 probes V_{SH}, the sampled (and interpolated) signal.

The alias experiment is conducted over two input signals:

1. Rectangular – As defined for WaveGen1 (up in both Figure 1.13 and Figure 1.14).
2. Filtered rectangular, as defined for WaveGen2 (down in both Figure 1.13 and Figure 1.14).

Change the FFT vertical scale between dB_V (logarithmic) and V_{rms} (linear). Observe the effect. Explain which are the advantages for each scale.

Figure 1.13 Ideal sampling on rectangular (up) and filtered (down) signal

Figure 1.14 0-degree interpolated rectangular (up) and filtered (down) signal

2 Sample and Hold

2.1 Background

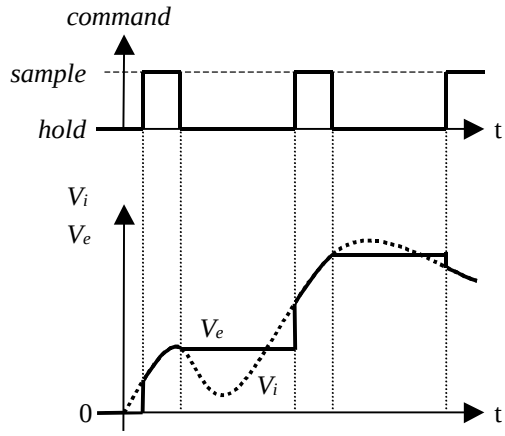


Figure 2.1 Ideal Sample and Hold circuit behavior

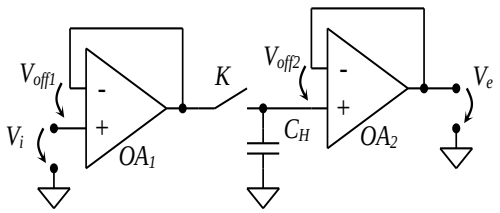


Figure 2.2 The Sample and Hold circuit principle

A *Sample and Hold* circuit follows the input signal during *sample* period and stores it for all the *hold* period. The *sample* period is usually required to be as short as possible, while the *hold* period needs to cover the AD conversion time. Figure 2.1 shows the input voltage V_i, and the output one V_e, for an ideal sample and hold circuit.

In Figure 2.2, the switch *K* is *ON* during *"sample"* state, loading capacitor C_H to voltage V_i. In *"hold"* state, *K* is *OFF* and V_e keeps the value it had at the end of *"sample"* state. OA_1 and OA_2 are impedance adapters: OA_1 provides low output impedance for fast C_H loading in *"sample"* state, the high input impedance of OA_2 prevents C_H discharge in *"hold"* state.

2.1.1 Sample and Hold Circuits Parameters and Errors

Sample and Hold circuits producing error phenomena can be classified considering the behavioral phase they manifest:

2.1.1.1 *"Sample" phase*

Sample offset - In Figure 2.2, the offset errors of OA_1 and OA_2 alter V_e value. When *"sample"* state ends:

$$V_e = V_{CH} + V_{off2} = V_i + V_{off1} + V_{off2} \quad \textbf{(2.1)}$$

Gain accuracy – due to the amplifier gain error in *"sample"* state.
Gain non-linearity – due to the operational amplifiers, but also to the non-

linearity of passive components implied in overall circuit gain.

2.1.1.2 *"Sample" to "Hold" transition*

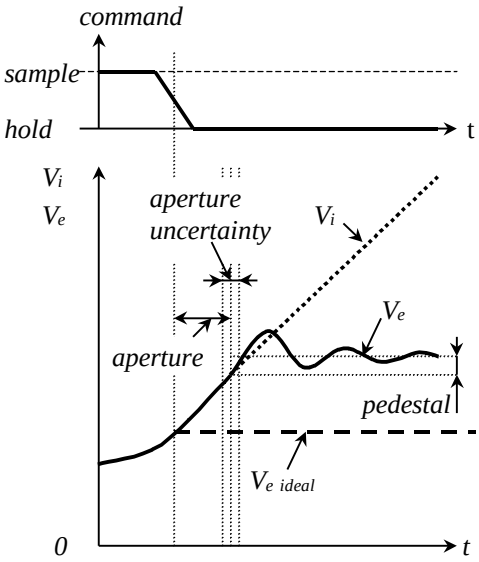


Figure 2.3 Sample to Hold transition

Aperture delay time – the time needed to switch from *"sample"* to *"hold"*. It is measured from the 50% point of the command signal *"hold"* transition until the output voltage doesn't follow any more the input voltage.

Aperture uncertainty time or *aperture jitter* – the variation of the *aperture delay time* for the same circuit at different transitions (same or different environment conditions).

Charge offset or *pedestal* – the offset of the output voltage due to the charge transfer between the storing capacitor and the switching circuit parasitic capacity.

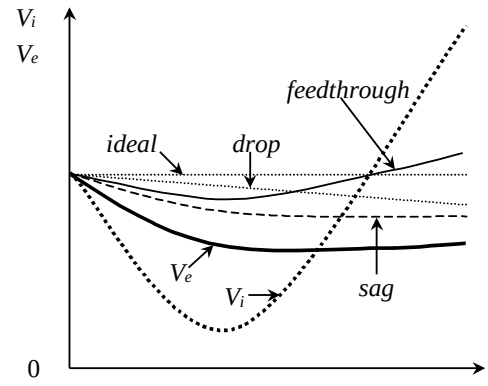


Figure 2.4 Hold phase errors

2.1.1.3 *"Hold" phase*

Figure 2.4 emphasizes the most important errors occurring in the *"hold"* phase. Each error type is shown, as it would be alone. The cumulated effect of all errors can be seen in the V_e evolution.

Droop rate – due to the leakage current of the storing capacitor, but mostly to the *"off"* switch current and to the output operational amplifier bias current. It is a linear time function, with dV_e/dt ratio.

Feedthrough – the quantum of input signal transferred to output during *"hold"* phase. It is due to the *"off"* switch finite impedance, but also to other parasitic impedances in the circuit.

Dielectric absorption in the hold capacitor or *sag*. A typical capacitor model is shown in Figure 2.5, where C_H is the nominal capacity, C_a, C_b, R_a, R_b are the parasitic capacities and resistances. C_H is charged fast during *sample* period, but the parasitic capacities need much more time and tend to redistribute the charge in the *"hold"* phase, reducing the swing by V_e in the previous *"sample"* phase.

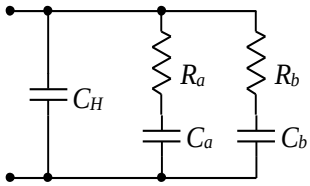


Figure 2.5 Typical capacitor model

2.1.1.4 *"Hold" to "sample" transition*

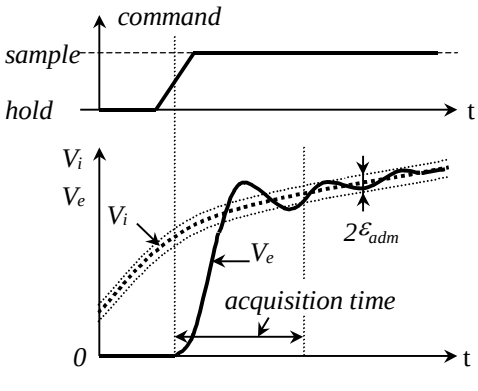


Figure 2.6 Hold to Sample transition

Acquisition time or *settling time* – the time needed by the sample and hold circuit output to enter (and stay) in an allowable error range around the ideal (final) value, when the circuit switches from *"hold"* to *"sample"*. Usually defined for a full-scale V_e jump.

Hold to sample switching transient – damped oscillation or exponential time function.

2.1.1.5 *Global defined errors*

This category includes errors summing effects from multiple sample and hold circuit behavior phases and manifesting in the global throughput function: from input signal V_i to the *"hold"* phase stored voltage V_e:

Throughput offset – the sum of *"sample-"* and *"charge-offset"*.

Throughput nonlinearity – the sum of *charge offset variation* and *gain nonlinearity*. The effect of these two errors cannot be compensated by gain and offset adjustment.

Dynamic nonlinearity - defined as total *"hold"* phase throughput function nonadjustable error. It includes the *throughput nonlinearity* and errors due to *storage capacitor losses* and temperature variations.

2.2 Simulation

Figure 2.7 shows the simulated circuit. Pin 5 of AD8592 is the *Shut Down* signal for the amplifier A; when *LOW*, the output of the amplifier goes *HiZ*, insulating C_{10}, for the *Hold* period. D_1 and R_6 translate the 5V logic signal

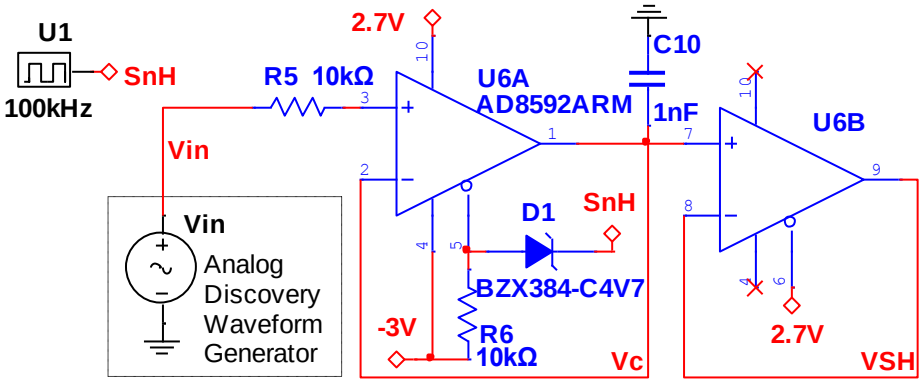


Figure 2.7 Sample and Hold simulation circuit

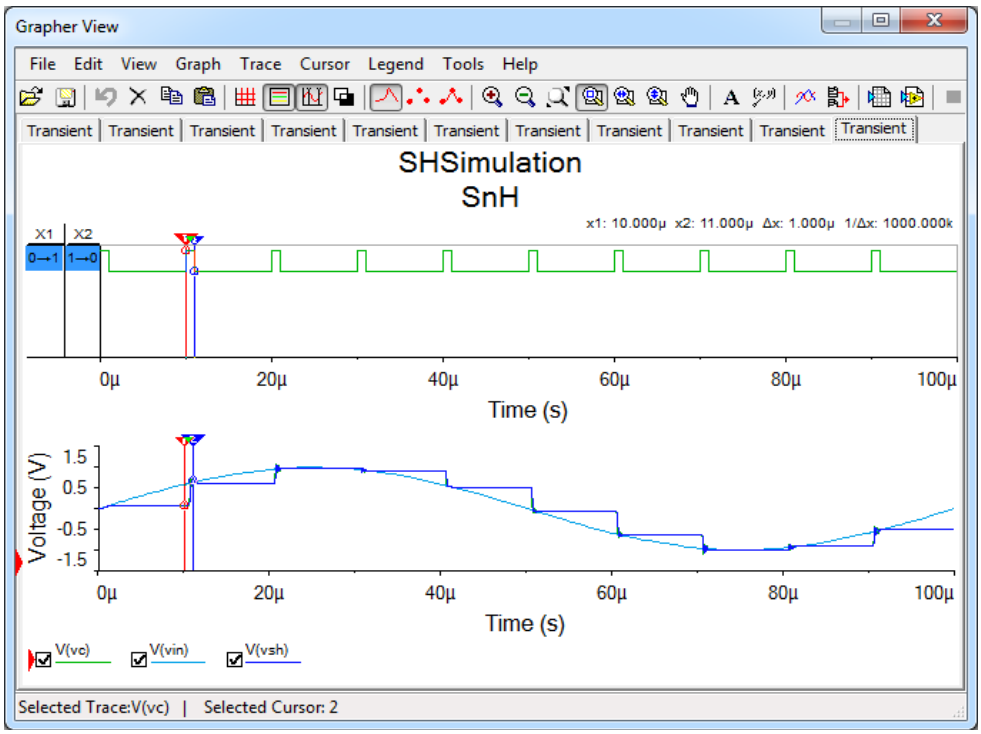


Figure 2.8 Sample and Hold general behavior simulation result

SnH to the *(-3V…+1.7V)* range, compatible with the *Shut Down* signal requirements.

2.2.1 General behavior - Simulation

Figure 2.8 shows the general behavior simulation results: V_{in} is *1V* amplitude, *10KHz* sinus. *SnH* is set at *100KHz*, with *10%* Duty Factor: the Sample period begins at every *10µs* and takes *1µs*. *10* samples are taken for each V_{in} period.

To analyze the errors and non-idealities, specific simulation conditions are set for each phase.

2.2.2 "Sample" phase - Simulation

For permanent "*Sample*", *SnH* is set continuously *High*, by replacing *U1* with a *5V* supply.

V_{in} is set *Triangular*, with $1V_{VV}$ amplitude, *20µs* period and *10µs* fall time (*50kHz*, *50%* duty factor).

Figure 2.9 shows very close shapes for V_{in}, V_C and V_{SH} (left). A postprocessor graph (right) shows the error V_{SH}-V_{in}.

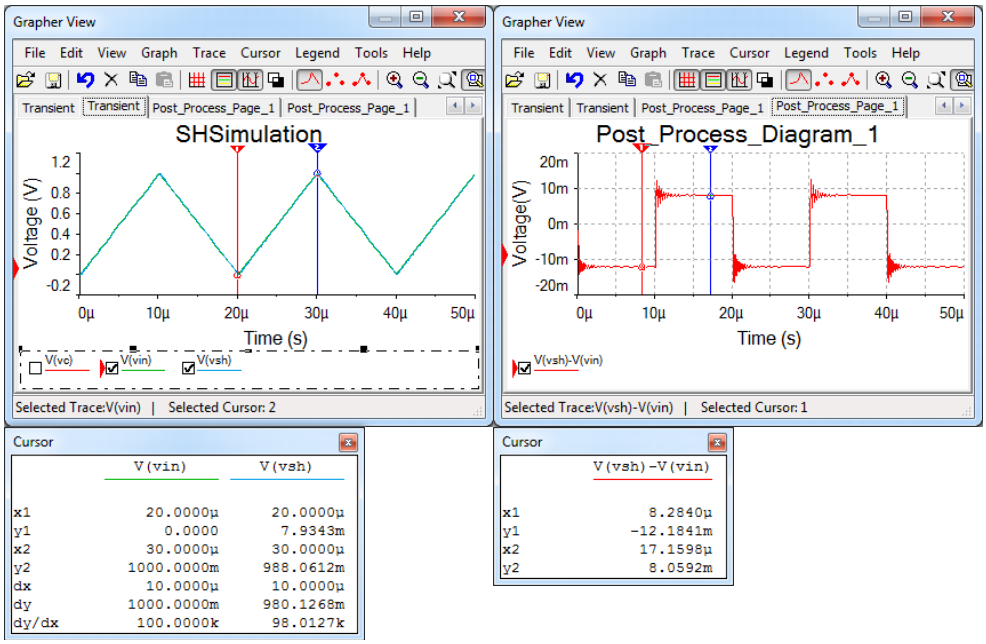


Figure 2.9 Sample phase simulation (left) and post processing (right)

The DC component (average value) of the error is the overal offset error of the Sample and Hold circuit.

$$\varepsilon_{off} = mean(V_{SH} - V_{in}) = \frac{-12.1841mV + 8.0592mV}{2} = -2.06245mV \quad \textbf{(2.2)}$$

The rectangular component in the error is given by the propagation time. During the rising slope, the apparent delay, $t_{d,r}$ is:

$$\frac{dV_{in}}{dt} = \frac{1V}{10\mu s} = \frac{-12.1841mV}{-t_{d,r}} \quad \textbf{(2.3)}$$

$$t_{d,r} = 121.841ns \quad \textbf{(2.4)}$$

During the falling slope:

$$\frac{dV_{in}}{dt} = \frac{-1V}{10\mu s} = \frac{8.0592mV}{-t_{d,f}} \quad \textbf{(2.5)}$$

$$t_{d,f} = 80.592ns \quad \textbf{(2.6)}$$

The actual propagation delay is the average of above:

$$t_d = \frac{t_{d,r} + t_{d,f}}{2} \approx 100ns \quad \textbf{(2.7)}$$

t_d is the time delay between the input- and the output signals, during the *sampling* period, as opposite to the *apperture time*, t_a, which is the time needed to switch from *hold* to *sampling*.

Since the error in the post processing diagram has no triangular component, the simulated gain error component is null: both gain accuracy and gain non-linearity are non-measurable low.

2.2.3 "Hold" to "Sample"- Simulation

In Figure 2.10, sampling frequency is *50kHz*, *50%* duty factor; *hold to sample* transitions happen at *0µs, 20µs, 40µs…* and *sample to hold* at *10µs, 30µs, …*
The input voltage is V_{in} = *1VVV, 5kHz*, symmetrical.
The left diagrams show the full simulation time. A detail of the *hold-to-sample* transition starting at time *20µs* is shown on the right.
The upper diagrams show V_{in} and V_{SH}, while the lower figures are post-processed to show the difference (error) V_{SH}-V_{in}.

In the lower-right diagram, V_{SH}-V_{in}, the *settling time*, t_s is:

$$t_s = 3.3\mu s@ \pm 1mV$$
$$t_s = 4.5\mu s@ \pm 0.2mV \qquad \textbf{(2.8)}$$

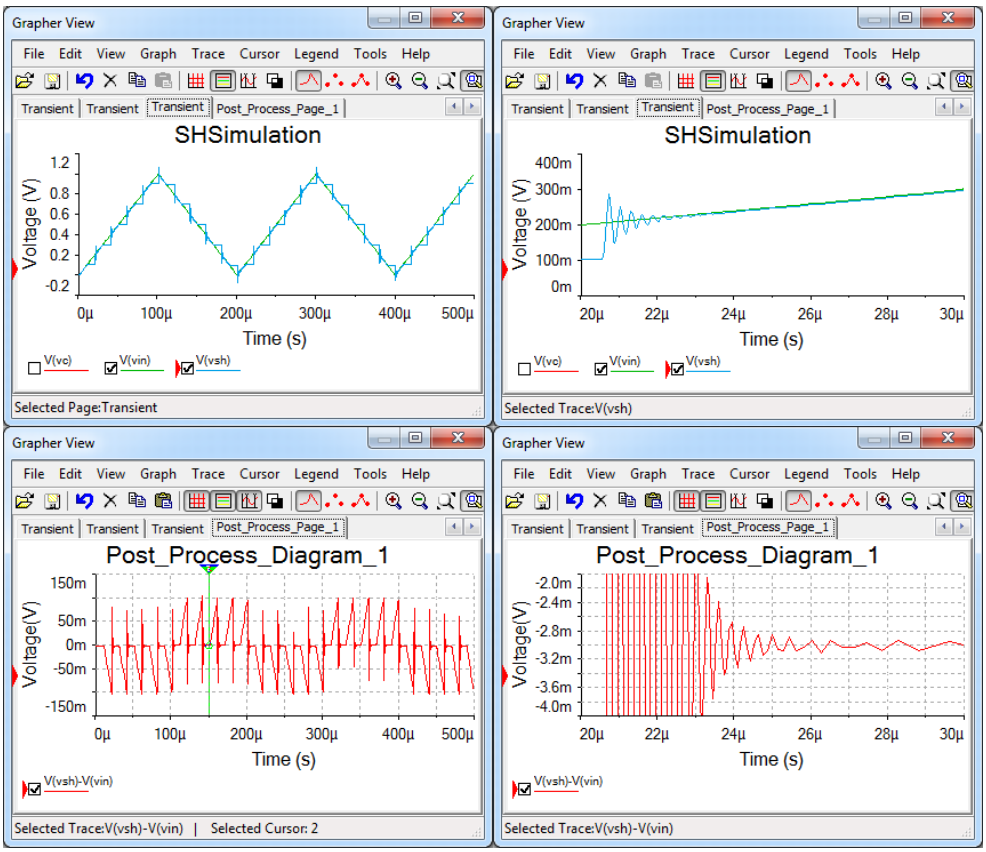


Figure 2.10 Hold to Sample simulation (up), and post processing V_{SH}-V_{in} (down), full size (left) and detail (down)

2.2.4 "Sample" to "Hold" – Simulation

Figure 2.11 shows two *sample* to *hold* details, from Figure 2.10; at time $50\mu s$, during rising slope of V_{in}, and at time $150\mu s$, during the falling slope of V_{in}.

The ideal *hold* $V_{SH,id}$ should be $500mV$ in both cases. However, this is influenced by the offset error as measured in (2.2), by the propagation time, evaluated in (2.7) and by the aperture time, t_a, resulting in:

$$V_{SH} = V_{SH,id} + \varepsilon_{off} + (t_a - t_d) \cdot \frac{dV_{in}}{dt} \tag{2.9}$$

$$V_{SH,r} = 500mV - 2mV + (t_a - 100ns) \cdot \frac{1V}{100\mu s} = 501.6mV \tag{2.10}$$

$$V_{SH,f} = 500mV - 2mV - (t_a - 100ns) \cdot \frac{1V}{100\mu s} = 494.4mV \tag{2.11}$$

Solving both (2.10) and (2.11) for t_a, independently results in:

$$t_a \cong 0.46\mu s \tag{2.12}$$

Which is confirmed by graphical reading in Figure 2.11.

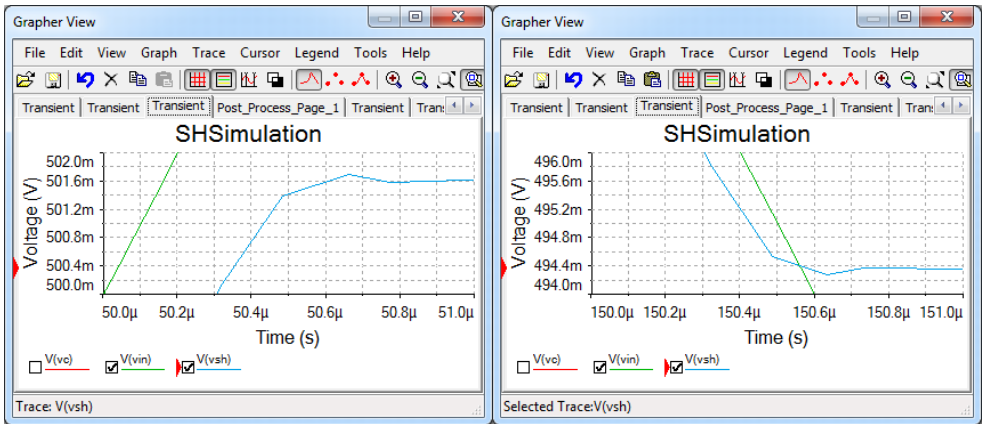


Figure 2.11 Sample to Hold detail; rising- (left) and falling slope (right)

2.2.5 "Hold" phase - Simulation

In Figure 2.12, the input signal is *20Hz*, $2V_{VV}$ *(±1V)*, *rectangular*, with *50%* duty factor. Sampling signal is set to *5Hz*, *1%* duty factor, *delay = 10ms*: the first sampling phase begins at *10ms*, the first hold period at time *12ms*. The hold period is *198ms*.

Subsequent sampling phases start at *210ms*, *410ms*, etc. All samples should theoretically pick the "*high*" value of the rectangular signal, *1V*.

In the right-side detail diagram, some errors are visible:

- the sample phase *offset*: $\varepsilon_{off} \cong 2mV$ **(2.13)**
- the *drop rate*: $\frac{dV_{SH}}{dt} \cong \frac{1mV}{200ms} = 5mV/s$ **(2.14)**
- the *Feedthrough*: $\frac{dV_{SH}}{dV_{in}} \cong \frac{0.1mV_{VV}}{2V_{VV}} = 50ppm = -86dB$ **(2.15)**

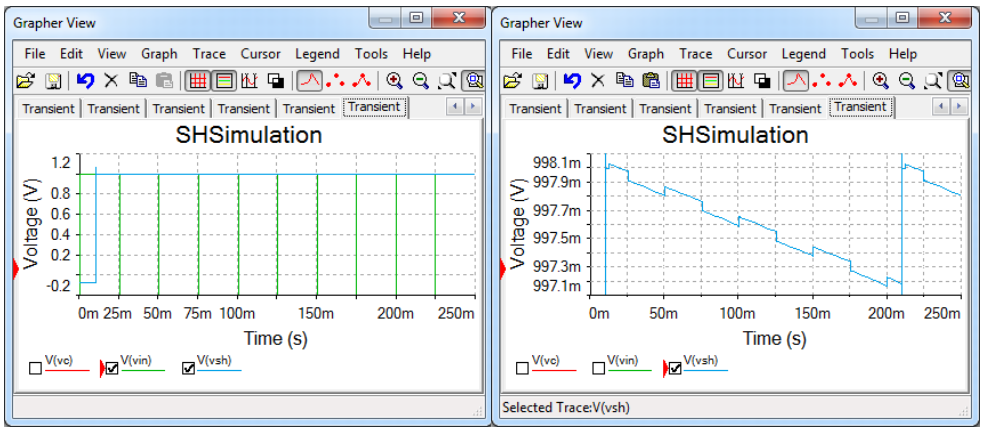


Figure 2.12 Hold phase simulation: full view (left) and V_{SH} detail (right)

2.3 Experiment

Figure 2.13 shows the full schematic of the experimental board, but circuits not used in the current experiment are shaded. U7 and U8 are linear voltage regulators, used to get *-3V* and *+2.7V* supply voltages; AD8592 supports maximum *6V* differential supply voltage. The other circuits work as explained next to Figure 2.7.

The board is designed to directly connect to the Digilent Analog Discovery. Channel 2 of the *AWG* (*W2*) generates the input signal, V_{in}. The digital pin *DIO10* generates *SnH*. The scope channels 1+ and 2+ are accessible in J2, for probing V_{in} (J10-2) and V_{SH} (J8-1).

Figure 2.14 shows the experimental board prepared with scope probe wires for the Sample and Hold experiment. Notice that the same board is used for several other experiments.

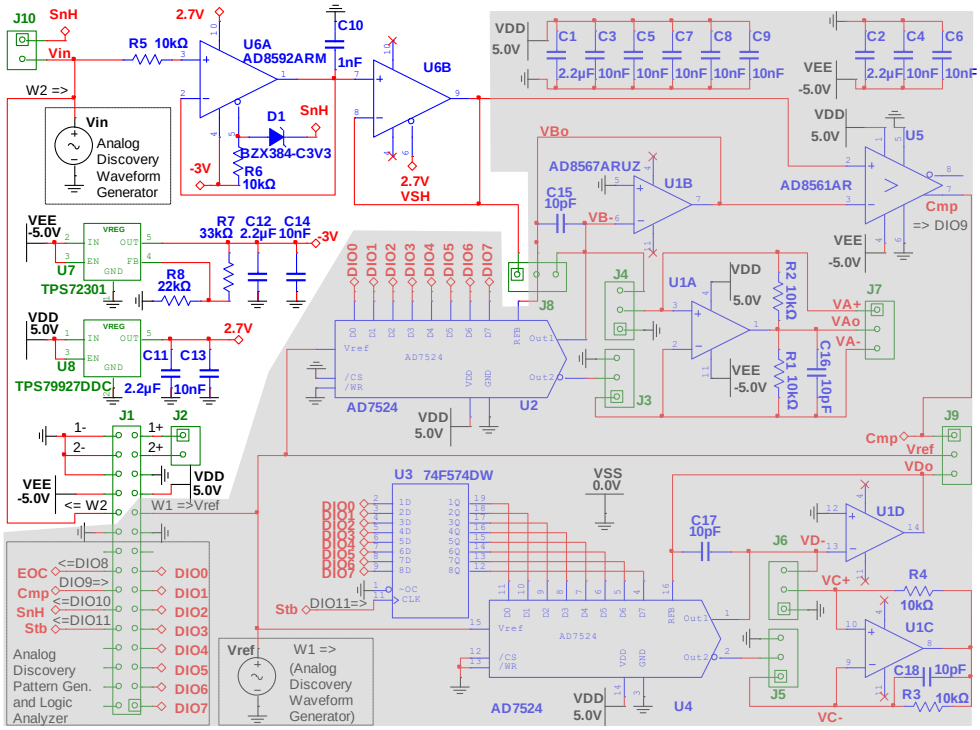


Figure 2.13 Sample and Hold experimental board schematic

2.3.1 General behavior

The WaweForms instruments are set as below (similar to Figure 2.8):

- *Supplies*: ON, +5V and -5V (If the WaveForms software reports an "Overcurrent condition" and stops the user power supplies, try starting the supplies one by one: first the positive supply and, after few seconds, the negative one).
- *WaveGen*, Channel 2: *Sinus, 10kHz, 1V amplitude, 50% symmetry*.
- *Pattern*, DIO10: *Push-Pull clock, 100kHz, 10% duty factor, 0º phase.*
- *Scope*:
  - Time base = *10µs/div.*
  - C1, C2 range = *500mV/div*
  - AddChannel/Digital/Signal: DIO10
  - Trigger: source = C1, condition = Falling

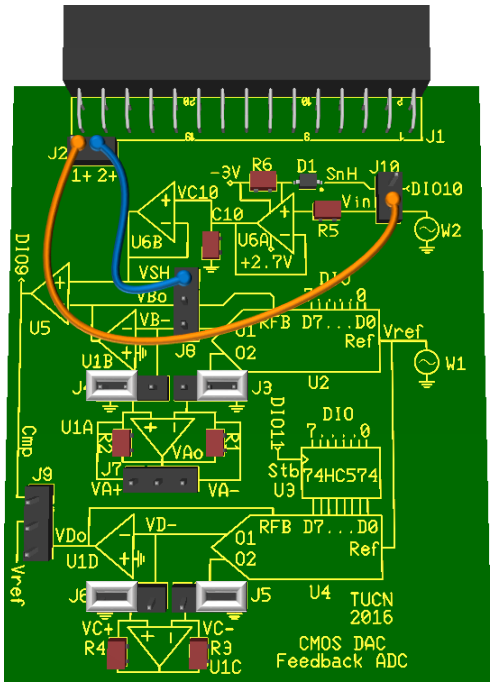


Figure 2.14 Sample and Hold experimental board

Figure 2.15 Sample and Hold general behavior experiment

To analyze the errors and non-idealities, specific simulation conditions are set for each phase.

2.3.2 "Sample" phase - Experiment

For permanent "*Sample*", *SnH* is set Constant/High in the *Pattern* generator.

V_{in} is set *Triangular*, with $1V_{VV}$ amplitude, *50kHz, 50%* symmetry.

Figure 2.16 shows very close shapes for C1 = V_{in}, and C2 = V_{SH}. Math1 (Math/Simple) channel computes the error C2-C1 = V_{SH}-V_{in}. A measurement view is open (View/Measure) and the *Average* and *Amplitude* of Math1 are displayed *(Add/DefinedMeasurement/Channel1/Vertical/…).*

The DC component (average value) of the error is the overal offset error of the Sample and Hold circuit.

$$\varepsilon_{off} \approx -4.8mV \tag{2.16}$$

The rectangular component in the error is given by the propagation time. During the rising slope, the apparent delay, $t_{d,r}$ is:

$$\frac{dV_{in}}{dt} = \frac{1V}{10\mu s} = \frac{-30mV}{-t_{d,r}} \tag{2.17}$$

$$t_{d,r} = 300ns \tag{2.18}$$

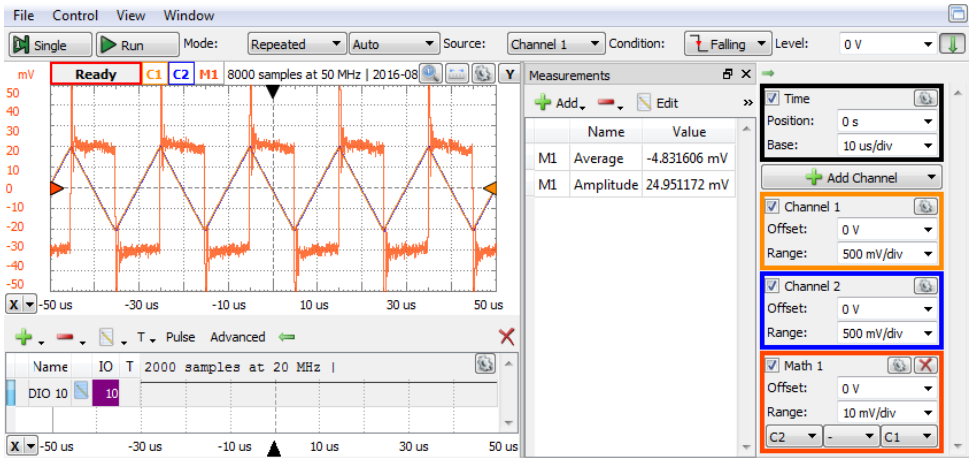


Figure 2.16 Sample phase experiment

During the falling slope:

$$\frac{dV_{in}}{dt} = \frac{-1V}{10\mu s} = \frac{20mV}{-t_{d,r}} \tag{2.19}$$

$$t_{d,f} = 200ns \tag{2.20}$$

The actual propagation delay is the average of above:

$$t_d = \frac{t_{d,r} + t_{d,f}}{2} \approx 250ns \tag{2.21}$$

t_d is the time delay between the input- and the output signals, during the *sampling* period, as opposite to the *apperture time*, t_a, which is the time needed to switch from *hold* to *sampling*.

The triangular component in the Math1 error, measures the gain error. In Figure 2.17, a Custom Math channel computes:

$$Math2 = C2 - (1 + \varepsilon_{gain}) \cdot C1 \tag{2.22}$$

ε_{gain} is manually adjusted to bring Math2 as close as possible to rectangular. In Figure 2.17, the final result is:

$$\varepsilon_{gain} = 0.2\% \tag{2.23}$$

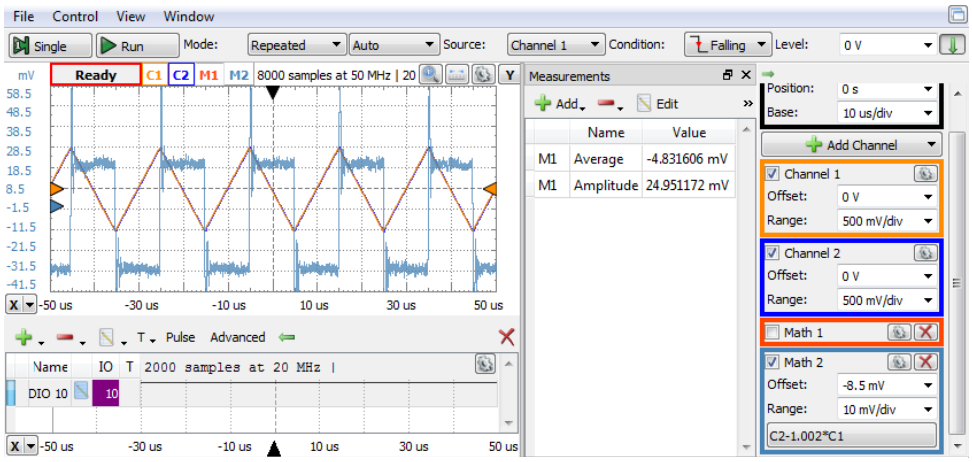


Figure 2.17 Sample phase experiment – gain error compensation

2.3.3 "Hold" to "Sample"- Experiment

In Figure 2.18, the sampling frequency is *50kHz*, *50%* duty factor; *hold to sample* transitions happen at *0µs*, *20µs*, *40µs*… and *sample to hold* at *10µs*, *30µs*, …

The input voltage is V_{in}: 0.5V Amplitude, 0.5V Offset, *5kHz*, triangle.

The upper diagram shows: *C1* = V_{in}, *C2* = V_{SH}, and the error *M1=C2-C1*. The lower diagrams show zoom details, at trigger time: V_{in} and V_{SH} (left), and *M1*= V_{SH} - V_{in} (right).

In the lower-right diagram, V_{SH}-V_{in}, the *settling time*, t_s is:

$$t_s = 4\mu s @ \pm 1mV \qquad \textbf{(2.24)}$$

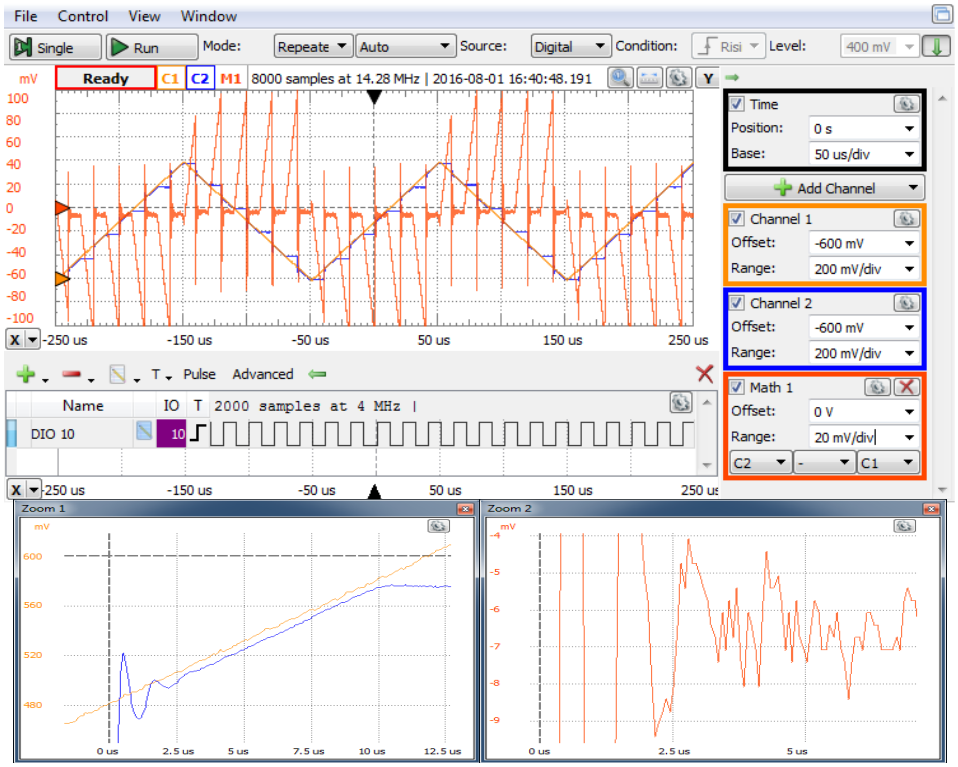


Figure 2.18 Hold to Sample experiment: full view (up) and zoom details (down): V_{in} and V_{SH} (left) and *Math1* = V_{SH}-V_{in} (right)

2.3.4 "Sample" to "Hold" – Experiment

Figure 2.19 shows two *sample* to *hold* details, from Figure 2.18 ; at time *10µs*, during rising slope of V_{in}, and at time *90µs*, during the falling slope of V_{in}.

The ideal *hold* $V_{SH,id}$ is influenced by the offset error as measured in (2.16), by the propagation time, evaluated in (2.21) and by the aperture time, t_a, resulting in:

$$V_{SH} = V_{SH,id} + \varepsilon_{off} + (t_a - t_d) \cdot \frac{dV_{in}}{dt} \qquad \textbf{(2.25)}$$

$$V_{SH,r} = 643mV - 4.8mV + (t_a - 250ns) \cdot \frac{1V}{100\mu s} = 637mV \qquad \textbf{(2.26)}$$

$$V_{SH,f} = 541mV - 4.8mV - (t_a - 250ns) \cdot \frac{1V}{100\mu s} = 537.5mV \qquad \textbf{(2.27)}$$

Solving both (2.26) and (2.27) for t_a, independently results in:

$$t_a \cong 125ns \qquad \textbf{(2.28)}$$

Which is confirmed by graphical reading in Figure 2.11(2.19).

Notice that the measurement above is at the noise limit: *1mV* of noise on top of V_{SH} or $V_{SH,id}$ would modify the computed t_a by *100ns*.

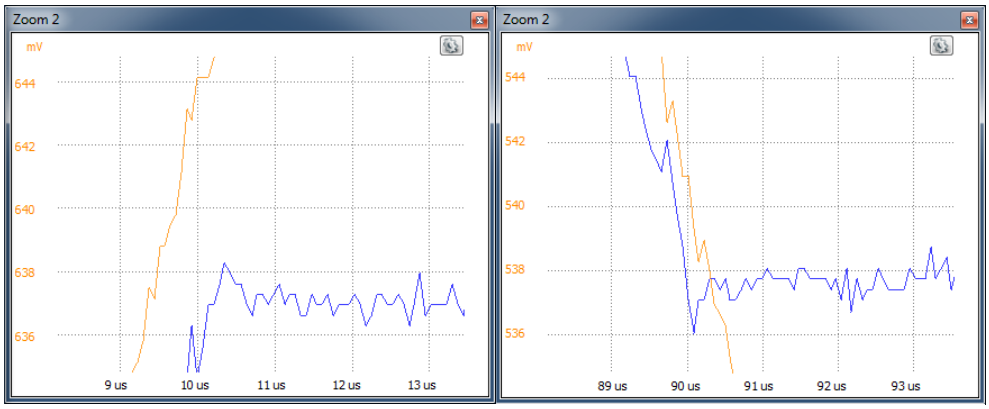


Figure 2.19 Sample to Hold details; - rising V_{in} (left) and falling V_{in} (right)

2.3.5 "Hold" phase - Experiment

In Figure 2.20, the input signal is *20Hz*, $2V_{VV}$ *(±1V)*, *rectangular*, with *50%* duty factor. Sampling signal is set to *5Hz*, *1%* duty factor: the triggered sampling phase begins at *0ms*, the first hold period at time *2ms*. The hold period is *198ms*. Subsequent sampling phases start at *200ms*, *400ms*, etc. All samples should theoretically pick the "*high*" value of the rectangular signal, *1V*.

In the diagram, some errors are visible:

- the sample phase *offset*: $$\varepsilon_{off} \cong -4mV \quad \textbf{(2.29)}$$

- the *drop rate*: $$\frac{dV_{SH}}{dt} \cong \frac{10mV}{200ms} = 50mV/s \quad \textbf{(2.30)}$$

- the *Feedthrough*: $$\frac{dV_{SH}}{dV_{in}} \cong \frac{1mV_{VV}}{2V_{VV}} = 500ppm = -66dB \quad \textbf{(2.31)}$$

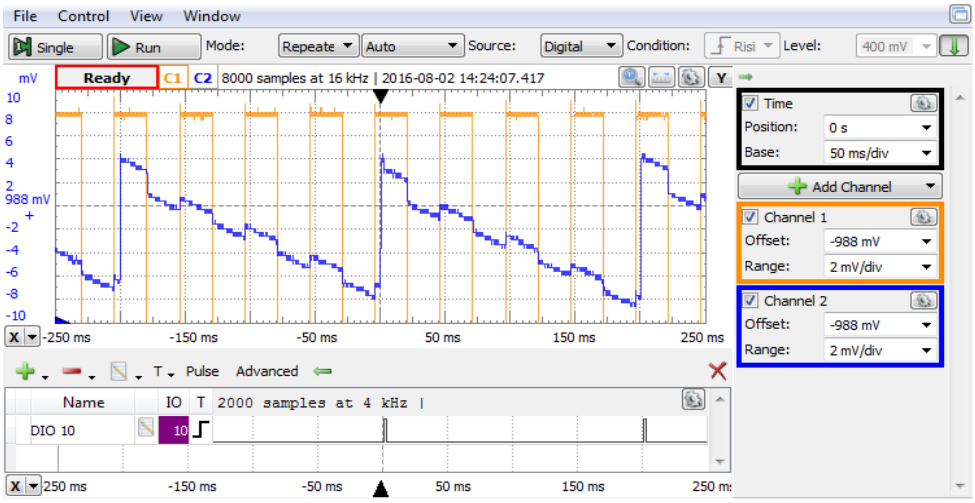


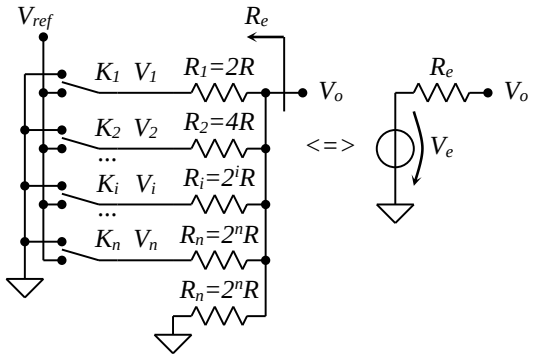
Figure 2.20 Hold phase experiment

3 Resistor Networks

3.1 Binary weighted resistors network, voltage switching, no load

3.1.1 Background

The resistor values in Figure 3.1 are power-of-two multiples of the generic value R. The dual-pole switches K_i are driven by a_i bits, which build the unipolar, binary number *{A}* as in (3.1).



$$\{A\} = 0.a_1 \ldots a_{(n-1)}a_n = \sum_{i=1}^{n} a_i \cdot 2^{-i} \qquad \textbf{(3.1)}$$

The branch voltages, V_i, are:

$$a_i = 0 \Rightarrow K_i \rightarrow GND \Rightarrow V_i = 0$$
$$a_i = 1 \Rightarrow K_i \rightarrow V_{ref} \Rightarrow V_i = V_{ref} \qquad \textbf{(3.2)}$$

Figure 3.1 Weighted resistors network Thevenin model

which can be expressed as: $V_i = a_i \cdot V_{ref}$ **(3.3)**

The Thevenin model has:

$$R_e = \frac{1}{\sum_{i=1}^{n}\frac{1}{R_i} + \frac{1}{R_n}} = \frac{R}{\sum_{i=1}^{n} 2^{-i} + 2^{-n}} = R \qquad \textbf{(3.4)}$$

$$V_e = \frac{\sum_{i=1}^{n}\frac{V_i}{R_i}}{\sum_{i=1}^{n}\frac{1}{R_i} + \frac{1}{R_n}} = \frac{\frac{V_{ref}}{R} \cdot \sum_{i=1}^{n} a_i \cdot 2^{-i}}{\frac{1}{R} \cdot (\sum_{i=1}^{n} 2^{-i} + 2^{-n})} = V_{ref} \cdot \sum_{i=1}^{n} a_i \cdot 2^{-i} = V_{ref} \cdot \{A\} \qquad \textbf{(3.5)}$$

With no load, the output voltage, V_o, is: $V_o = V_e$ **(3.6)**

The full-scale voltage, V_{FS}, the absolute resolution, R_{abs}, and the voltage corresponding to the least significant bit, V_{LSB}, (n-bit, binary, unipolar) are:

$$V_{FS} = V_o|\{A\} = 1 = V_{ref} \qquad \textbf{(3.7)}$$

$$R_{abs} = V_{LSB} = V_{FS} \cdot 2^{-n} \qquad \textbf{(3.8)}$$

3.1.2 Simulation

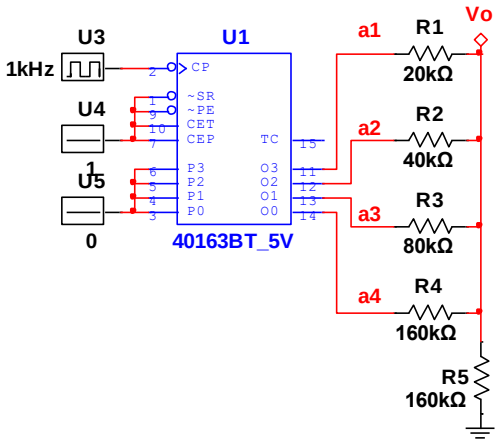


Figure 3.2 Weighted resistors network simulation schematic

3.1.2.1 *Ideal circuit*

Figure 3.2 shows the Multisim schematic file for simulation. A binary counter, with 1kHz clock, generates the *{A}* numbers. The reference voltage (5V) and the switches are simulated by the 40163BT_5V output buffers:

$$\begin{aligned} R &= 10k\Omega \\ n &= 4 \\ V_{ref} &= V_{FS} = 5V \\ R_{abs} &= V_{LSB} \\ &= 312.5mV \end{aligned} \qquad \textbf{(3.9)}$$

Figure 3.3 shows the transient simulation results. The digital graph (up) shows the bits of the number *{A}* (four-bit binary counter), the analog graph (down) shows the V_o voltage (ramp from *0V* to V_{FS}-V_{LSB} =*4.6875V*, in steps of V_{LSB}= *312.5mV*). The cursors are set at *{A}=0* and *{A}=15LSB=*, with *dy=*V_{FS}-V_{LSB}.

3.1.2.2 *Resistor mismatch induced errors*

For analyzing the resistor mismatch induced errors, a series of 5 transient simulations were done. For simulation number *k*, one resistor (R_k) was altered by the relative error of ε_R=*+1%*;

A postprocessor was set to calculate the error of each simulation.

err(R1)=V(vo)-(V(a1)/2+V(a2)/4+V(a3)/8+V(a4)/16)
err(R2)=tran02.V(vo)-(tran02.V(a1)/2+tran02.V(a2)/4+tran02.V(a3)/8+tran02.V(a4)/16)
err(R3)=tran03.V(vo)-(tran03.V(a1)/2+tran03.V(a2)/4+tran03.V(a3)/8+tran03.V(a4)/16)
err(R4)=tran04.V(vo)-(tran04.V(a1)/2+tran04.V(a2)/4+tran04.V(a3)/8+tran04.V(a4)/16)
err(R5)=tran05.V(vo)-(tran05.V(a1)/2+tran05.V(a2)/4+tran05.V(a3)/8+tran05.V(a4)/16)

$$\begin{aligned} R_k &= 2^k \cdot R \cdot (1 + \varepsilon_R) \\ R_i &= 2^i \cdot R \quad \forall i \neq k \end{aligned} \qquad \textbf{(3.10)}$$

The post processing results are shown in Figure 3.4. Some interesting, yet predictable, properties of the graphics:

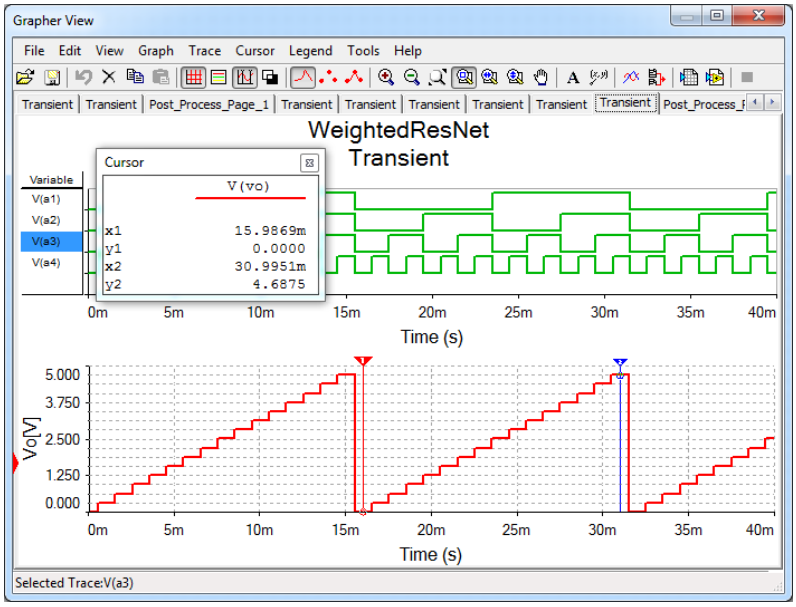


Figure 3.3 Weighted resistors network simulation results

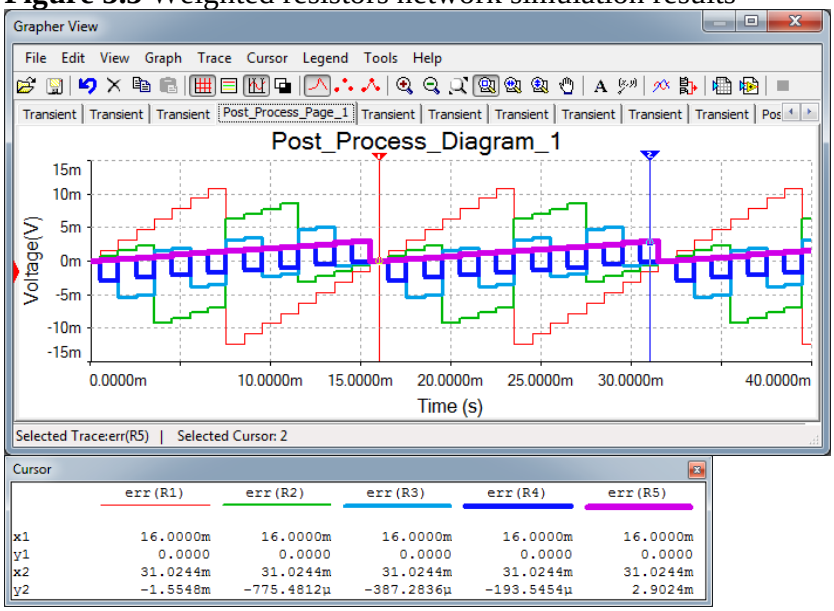


Figure 3.4 1% resistor mismatch errors

A. *err(R5)* is different compared to all others: it is linear with the input number *{A}*.
B. all other errors are non-linear and have minimum absolute values for *{A}=0* and *{A}=1-1LSB* (see cursor positions in Figure 3.3 and Figure 3.4). Indeed, conform to equation (3.5), when all bits $a_i = 0$, $V_e = 0$ and does not depend on R_k (nor on R_k errors). For *{A}= 1* (if that would be possible) $V_e=V_{ref}$, not depending on R_i or R_i errors.
C. the resistor mismatch in higher significant branches of the network propagates with higher weight in the output error.
D. the polarity of the error due to branch k changes when the a_k bit changes. That makes the errors to be "linearity errors".

3.1.2.3 *Vi induced errors*

If all V_i branch voltages are affected by the same percentage error, this is equivalent to an V_{ref} error in (3.3) and (3.5), and generates an V_o "gain error", which is easy to compensate. However, if V_i errors are different for each branch, this generates linearity errors.

For analyzing the V_i mismatch induced errors, a series of 4 transient simulations were done. For simulation number k, one branch voltage (V_k) was altered by the relative error of $\varepsilon_R=+1\%$.

$$\begin{aligned} V_k &= a_k \cdot V_{ref} \cdot (1 + \varepsilon_R) \\ V_i &= a_i \cdot V_{ref} \quad \forall i \neq k \end{aligned} \qquad \textbf{(3.11)}$$

A postprocessor was set to calculate the error of each simulation. The results are shown in Figure 3.5. Interesting graphics properties:

A. the V_i mismatch errors propagate to the output signal with the same weight as the useful signal of the branch (2^{-k}). Accordingly, when designing a DAC, the most significant branches should be designed to have minimal errors, while least significant branches have less restrictions.
B. as simulated, the error due to branch k only appears when a_k bit is 1 and is null when the a_k bit is 0. That makes the errors to be "linearity errors".

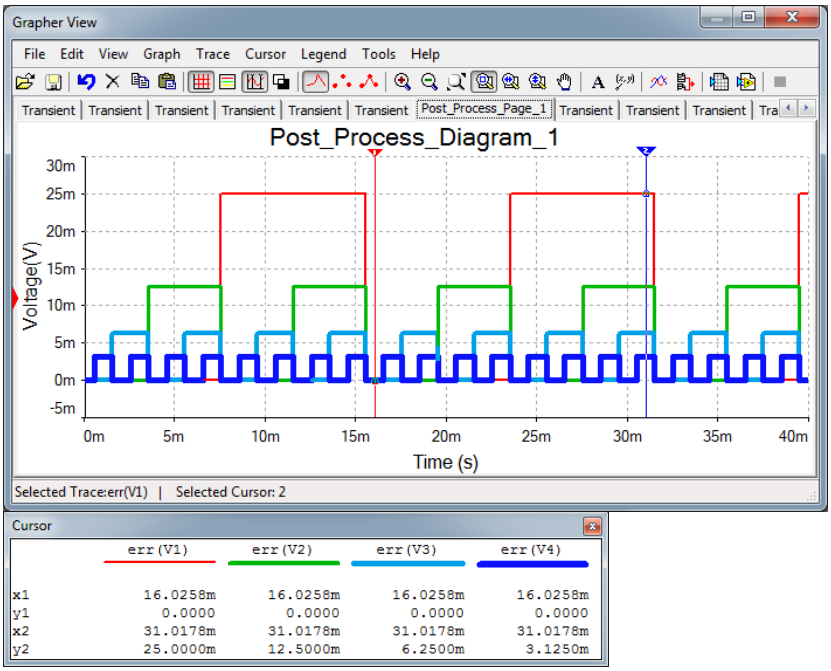


Figure 3.5 1% Vi mismatch errors

3.1.2.4 *Dynamic errors*

To simulate dynamic errors, ADG859 analog switches are used in Figure 3.6 and Figure 3.7. The SPICE model of these circuits considers finite propagation time, parasitical capacitance of switches, as well as different switching time for rising, respectively falling time.

A. *Settling time*: is measured from the change of the input number until the output signal enters and remains within the allowed error band (around the ideal value). In Figure 3.7, the transition from *{A}=15LSB* to *0LSB* happens at time *310ns*, but V_o finishes the corresponding trip (with *0.1LSB* error band) *10ns* later.
B. *Glitch*: multiple bits cannot switch absolutely at the same time. Usually, rising and falling transition times are different. In Figure 3.7, the transition from *{A}=0.0111*, to *{A}=0.1000* at time *150ns* generates an intermediate state of *{A}=0.1111*. Ideally, V_o would shortly jump at V_{FS}*-1LSB* and then back to $V_{FS}/2$. The analog limited speed reduces the glitch to the artefact seen at about *160ns.*

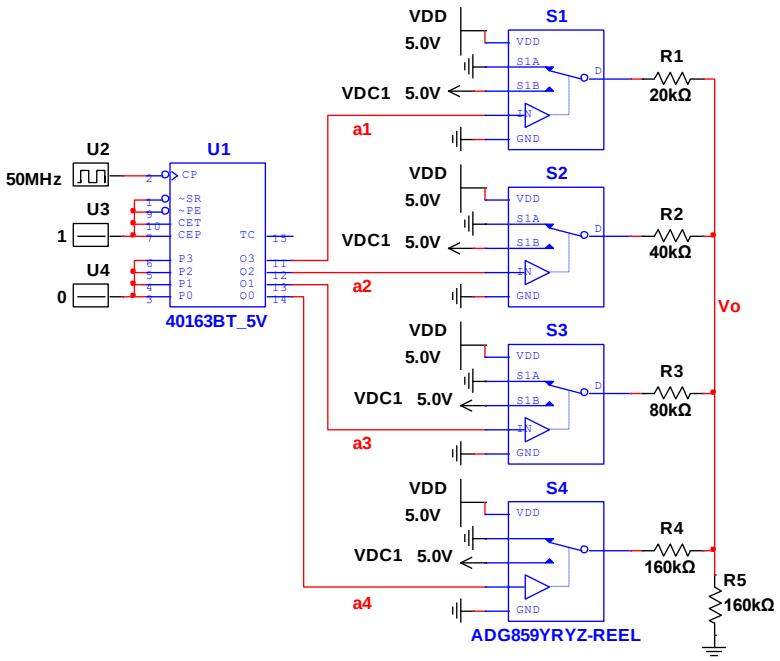


Figure 3.6 Dynamic simulation schematic

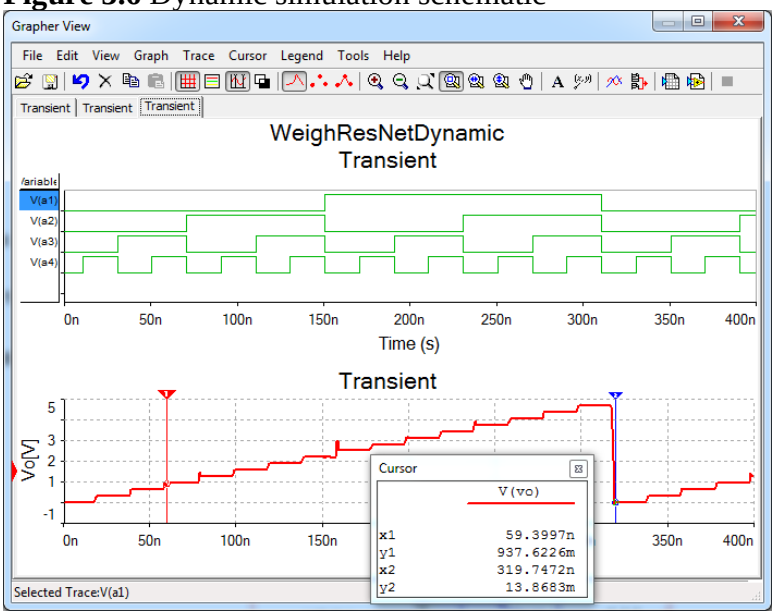


Figure 3.7 Dynamic simulation

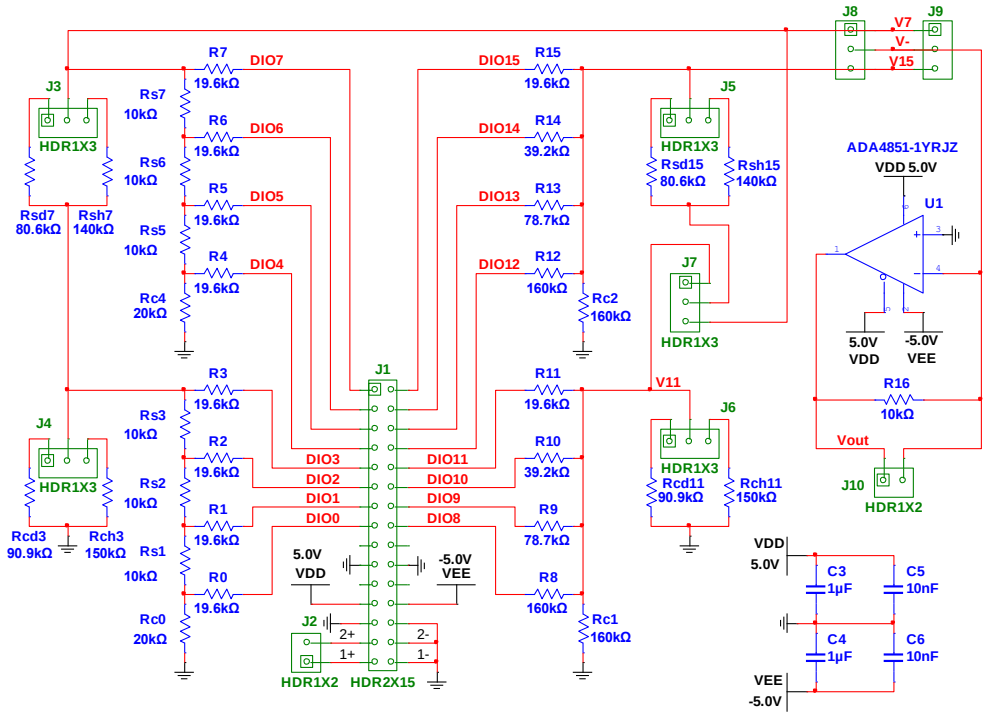


Figure 3.8 The Resistor Networks board schematic

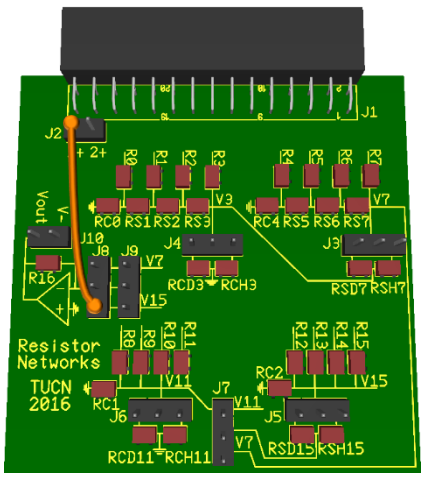


Figure 3.9 The Resistor Networks board

3.1.3 Experiment and measurements

The experiment uses the Resistor Networks board, shown in Figure 3.8 and Figure 3.9.

The network includes *R15…R12*, and *Rc2*. The branches are connected to 4 pins of Analog Discovery: *DIO15=a_1=MSB*, *DIO14=a_2*, *DIO13=a_3*, *DIO12=a_4=LSB*.

To separate the output voltage, *V15 = V_o*, no jumper should be loaded on *J5*, *J8*, *J9*.

The scope channel 1 is used to measure *V15*: pin 3 of *J8 (V15)* must be tied to pin 1 of *J2 (1+)*.

3.1.3.1 *Non-idealities*

The FPGA within the Analog Discovery drives digital signals *DIO15…DIO12* to voltages *approximating Vcc=3.3V* for a_j*=1*, and V_{GND}*= 0V*, for a_j*=0*. The approximations generate gain and linearity errors.

The digital signals, DIOx, are protected within the Analog Discovery with R_s *= 220Ω* series resistors, which add to the branch resistance. The equivalent branch resistances are:

$$R15_e = R15 + R_s; R14_e = R14 + R_s; \\ R13_e = R13 + R_s; R12_e = R12 + R_s \quad \textbf{(3.12)}$$

Since only certain resistor values are available as discrete components, the equivalent resistances of the branches are not perfectly matching the ideal power of two sequence. This generates linearity and gain errors.

3.1.3.2 *Experiment*

In the *Patterns* Generator, *Add/Bus* of DIO15…DIO12, set it as *Binary Counter*, *Push-Pull*, with frequency = *1kHz*. Run Patterns Generator.

Set the *Scope* as in Figure 3.10. Run the scope. Observe the graph. Compare to Figure 3.3. Notice that for the experimental board, the ideal values are:

$$R = 10k\Omega; n = 4; V_{ref} = V_{FS} = 3.3V; R_{abs} = V_{LSB} = 206.25mV \quad \textbf{(3.13)}$$

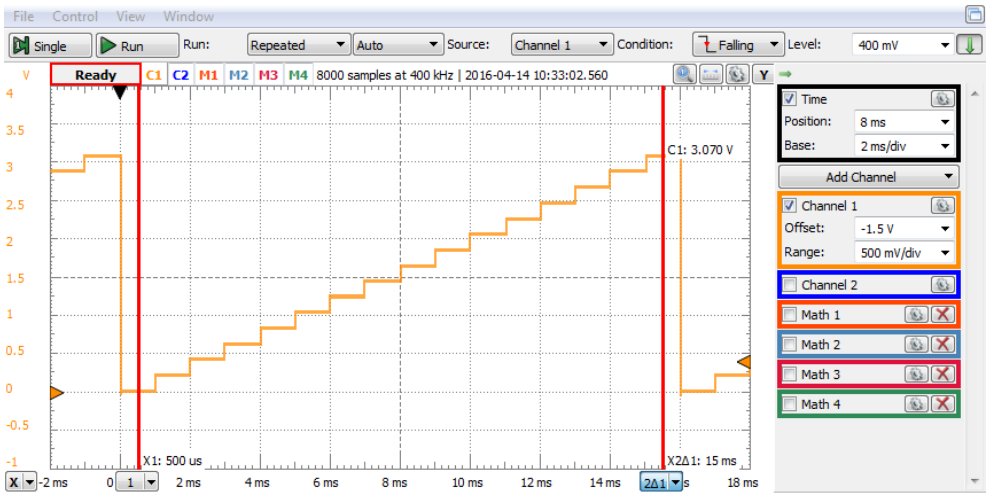


Figure 3.10 The weighted resistor network output voltage ramp

3.1.3.3 *Measurements*

Set two cursors at times *0.5ms* and *15.5ms*. Hover the mouse over the cursors to read the voltage of *C1* at the cursors.

Task 1. Starting from the cursor measured voltages, write down the equations and calculate the **actual values** of V_{FS} and V_{LSB}**.**

In the scope instrument, AddChannel/*Math/Custom* to create channel Math 1. Edit the script shown in Figure 3.11. This describes the ideal shape of *C1*. Notice the meaning of the constants in the equation: *Time*1000* shows time in *ms*, $V_{FS}=3.3V$, $2^4=16$ is the number of possible values for $n=4$ bits. However, the script works only for one counting cycle, from 0ms to 16ms, measured from the trigger event. Ignore the graph beyond these limits.

Disable *C1*. Read the voltage of *Math 1* at the cursors.

Task 2. Starting from the cursor measured voltages, write down the equations and calculate the **ideal values** of V_{FS} and V_{LSB}.

In the scope instrument, click *AddChannel/Math/Simple* to create channel *Math 2*. Edit the math function as *C1-M1*. *Math 2* shows the error of *C1*, as in Figure 3.12. Disable *M1*. Stop the scope to freeze the image. Set the *Range* and *Offset* of *Math 2* for optimal reading. Hover the mouse over the cursors to read the voltage of *Math 2* at the cursors.

Notice that the equation used in Figure 3.11 includes quantization (*floor* function), so *Math2* does not include the quantization error. Remove quantization from *Math1* and notice the effect on *Math1* and *Math2*.

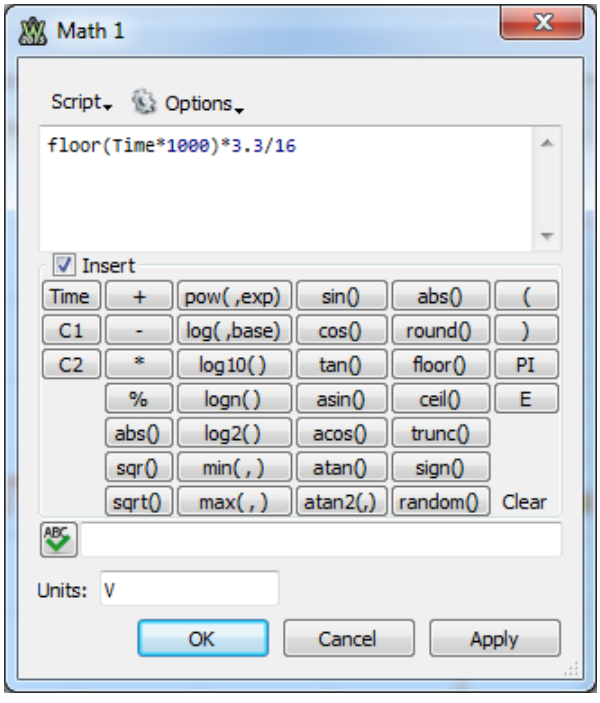


Figure 3.11 Math script for ideal 4-bit ramp

Explain the effect. Restore quantization in *Math1* for subsequent steps.

Math2 shows the global error of *C1*. It includes offset, gain, linearity and dynamic errors.
For understanding the spikes of Figure 3.12, change the time base to a low value (2us/div), as in Figure 3.13. At this scale, the dots indicate the Analog Discovery samples. Notice the sampled value of *C1* (real signal) versus the ideal *Math1*. The difference is shown in *Math2* (notice that *Math2* has another

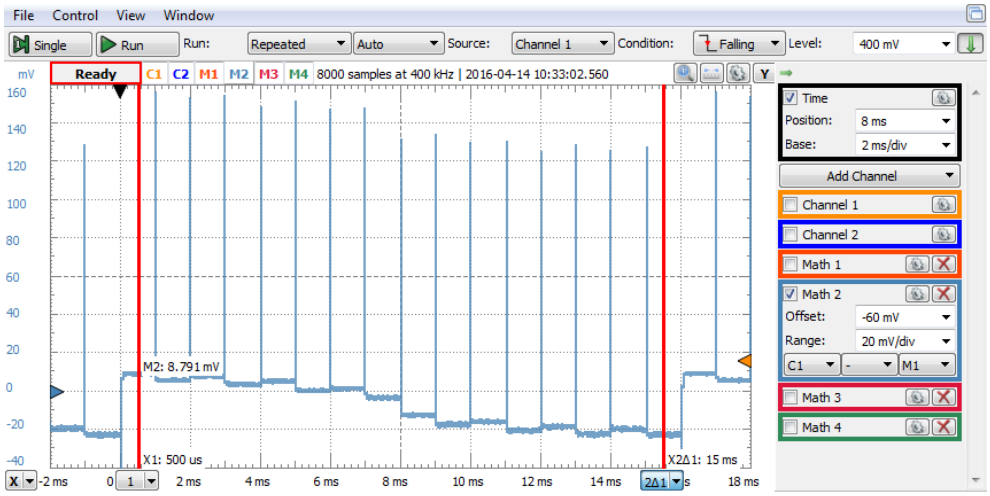


Figure 3.12 The weighted resistor network output voltage error

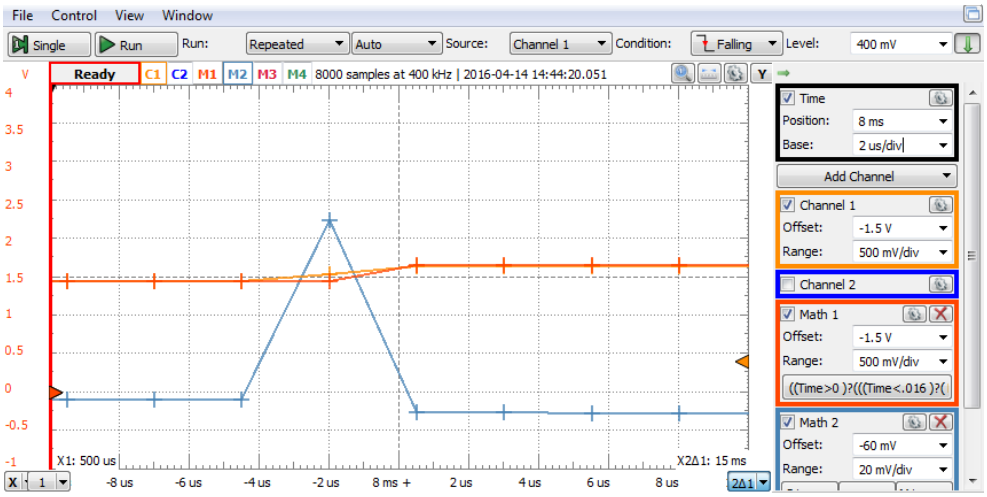


Figure 3.13 The weighted resistor network output voltage dynamic error

scope range as *C1* and *Math1*). This difference is not a dynamic error of the resistor network, rather a sampling misalignment of *C1* and *Math1*. Actual resistor network dynamic errors are measured later in this paragraph.

To remove the **offset** and **gain** errors, add custom *Math 3*: *C1*1.01+0.01*, where cor_{gain}*=1.01* is the gain correction coefficient and cor_{off}*=0.01* is the offset correction (in *mV*). Add simple *Math 4*: *M3-M1*. Edit the equation of *Math 3* to minimize *Math 4*: adjust the cor_{gain} to bring the overall slope of *Math 4* as close as possible to zero (horizontal); adjust the cor_{off} to bring the average value of *Math 4* as close as possible to zero. When offset and gain are well compensated, as in Figure 3.14, *Math 4* only includes **linearity** and dynamic errors. The initial *C1* gain and offset errors can be calculated:

$$\begin{aligned} &\mathrm{C1}*\mathrm{cor}_{\mathrm{gain}}+\mathrm{cor}_{\mathrm{off}} = Math1 \\ &\mathrm{C1} = \frac{Math1}{\mathrm{cor}_{\mathrm{gain}}} - \frac{\mathrm{cor}_{\mathrm{off}}}{\mathrm{cor}_{\mathrm{gain}}} = Math1 \cdot \left(1 + \varepsilon_{gain,rel}\right) + \varepsilon_{off,abs} \\ &\varepsilon_{gain,rel} = \frac{1}{\mathrm{cor}_{\mathrm{gain}}} - 1 = relative\ \ gain\ \ error \\ &\varepsilon_{off,abs} = -\frac{\mathrm{cor}_{\mathrm{off}}}{\mathrm{cor}_{\mathrm{gain}}} = absolute\ \ offset\ \ error \end{aligned} \qquad \textbf{(3.14)}$$

Task 3. Compute the **absolute** and **relative gain** and **offset errors** of your weighted resistor network.

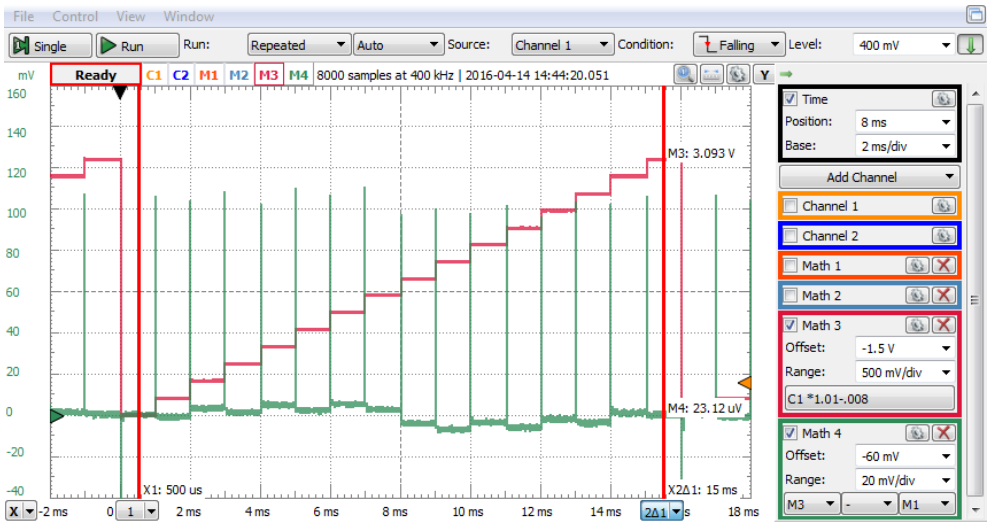


Figure 3.14 Linearity after compensating gain and offset errors

To compute the integral absolute linearity error, add two horizontal cursors (Y drop menu in the upper right corner of the plot). Make the cursors independent (in the cursor's drop menu, set *Reference* to *none*, for both cursors). Place the cursors on the highest positive, respective lowest negative values of *Math 4* (visually mediate the quantifying noise), as in Figure 3.15. The highest absolute value among the two cursors is the **maximum integral linearity error**.

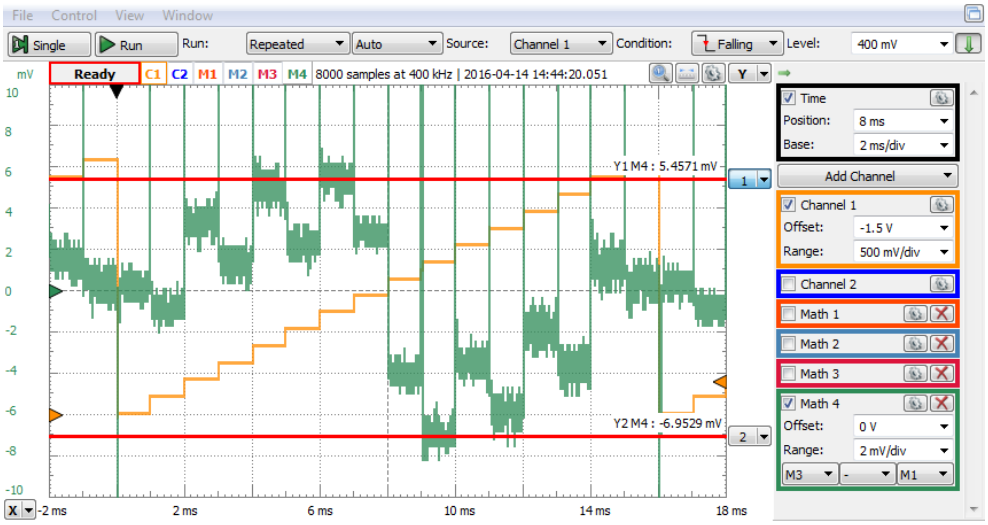


Figure 3.15 Measuring the absolute integral linearity error

To compute the differential absolute linearity error, add two horizontal cursors (Y drop menu in the upper right corner of the plot). Make cursor 2 relative to cursor 1 (in the cursor's 2 drop menu, set *Reference* to *1*). Drag cursors to catch the biggest difference between two adjacent flat levels of Math 4 (visually mediate the quantifying noise), as in Figure 3.17. Notice that Cursor 2 displays the delta relative to Cursor 1. The absolute value of the difference is the **maximum differential linearity error**. If the maximum differential linearity error is higher as V_{LSB}, the *DAC* is non-monotonic.

Task 4. Compute the **absolute** and **relative integral** and **differential errors** of your weighted resistor network. Verify if your DAC is **monotonic**.

Dynamic errors are not measurable at this time-base. Set the scope time base to *5us/div* and the time position to *15us*. *Add/Digital/Bus DIO15…DIO12*.to the Scope view. That builds a combined instrument, with synchronized analog and digital signals. Set the trigger source to *Digital*. In the digital area

of the scope, set the trigger condition to *DIO15 = falling edge*, all other *DIOx = Don't Care*. This sets the trigger event and time origin of the acquisition to the moment when the digital bus rolls over from *15* to *0*. Set scope *Channel1* range and offset as in Figure 3.17, for optimal readings.

Consider the acceptable error = $\pm 0.1 V_{LSB} = \pm 20mV$, as in paragraph 3.1.2.4, A.

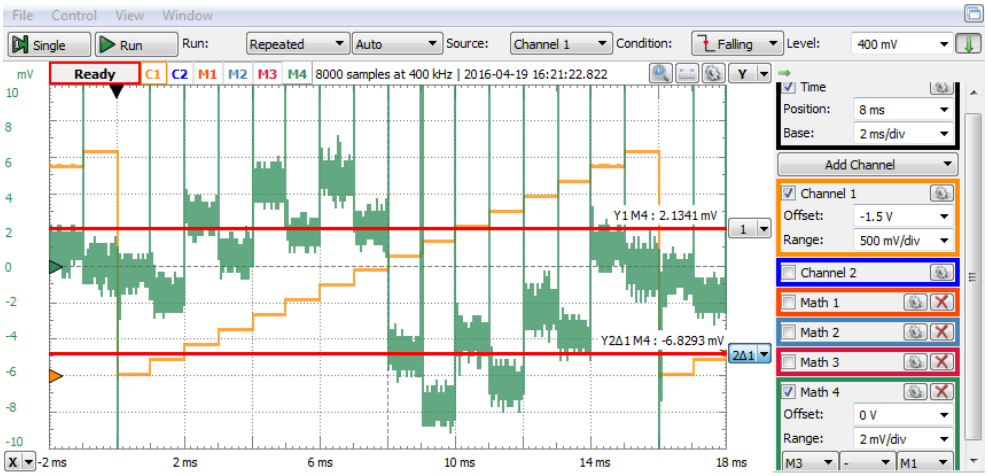


Figure 3.17 Measuring the absolute differential linearity error

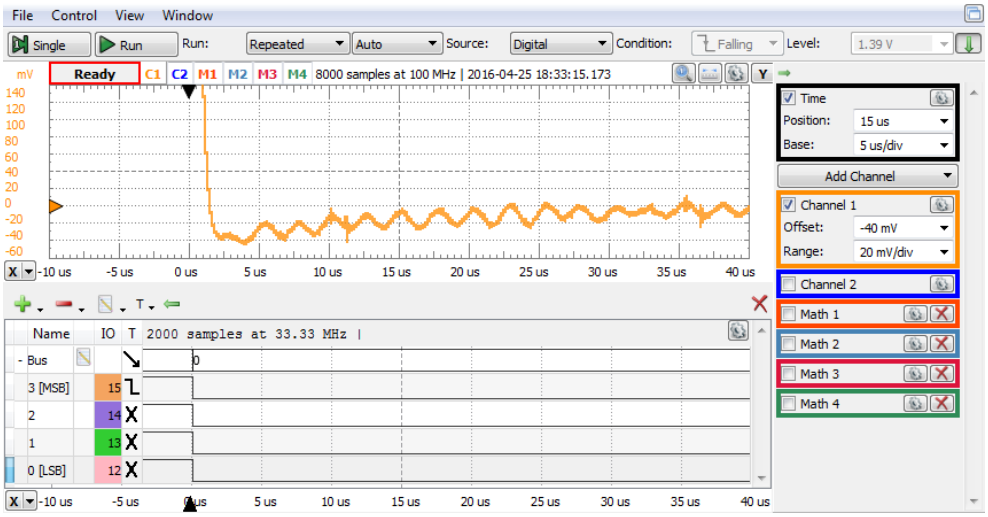


Figure 3.16 Measuring the settling time

The ideal output voltage should drop instantly at *time = 0*, from V_{FS} to *0*. However, in Figure 3.17, the real *V15*, measured by the scope *Channel1*, has an **overshoot** of about *-40mV*, has a **damped oscillation**, and enters the allowed *±20mV* error band at a **settling time** of about *25us*.

Task 5. Measure the settling time of your weighted resistor network.

The next figures show some digital transitions of amplitude 1LSB. Each of these transitions should result in a V_{LSB} rising step of *V15*.

Finite settling time can be observed. Furthermore, **glitches** can be seen as explained in paragraph 3.1.2.4, section B: instead of direct rising ramp, the transition begins with a negative pulse (showing that the effect of the falling bits in the input number *DIO15…DIO12* is faster than the effect of the rising bits). The glitch is null in Figure 3.18, since a single bit is switching, slightly observable in Figure 3.19, with just two bits switching in opposite directions, and doubles in each Figure 3.20 and Figure 3.21 with increasing number of simultaneously switching bits. The small switching time unbalance and the integration effect of parasitical capacitances in the schematic keep the glitches much smaller than the maximum theoretical level (V_{FS}/2 in Figure 3.21).

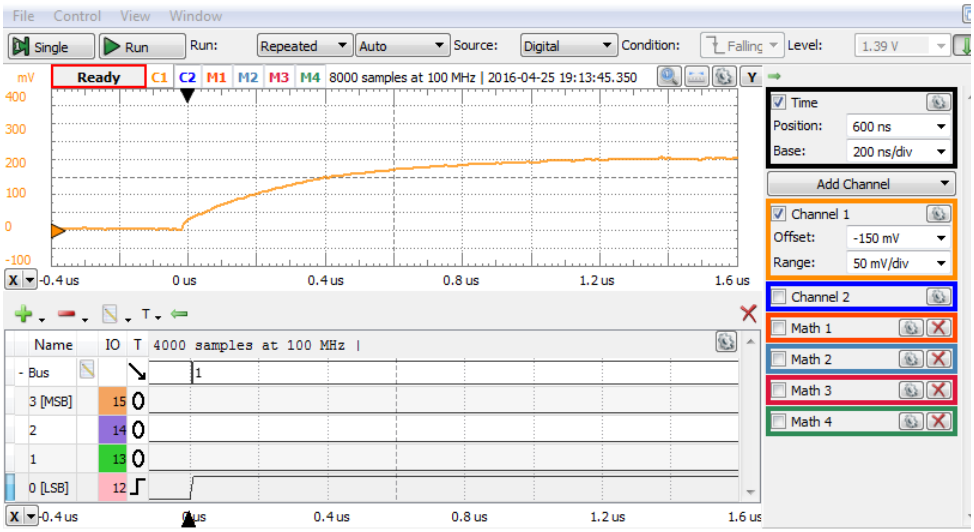


Figure 3.18 *DIO15…DIO12* transition *0000* to *0001*. No Glitch

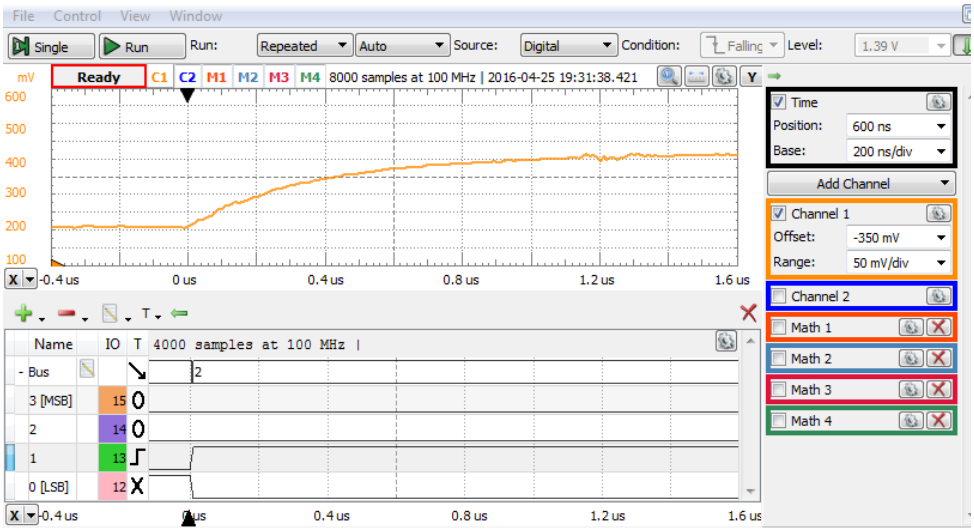


Figure 3.19 *DIO15…DIO12* transition *0001* to *0010*. Small Glitch

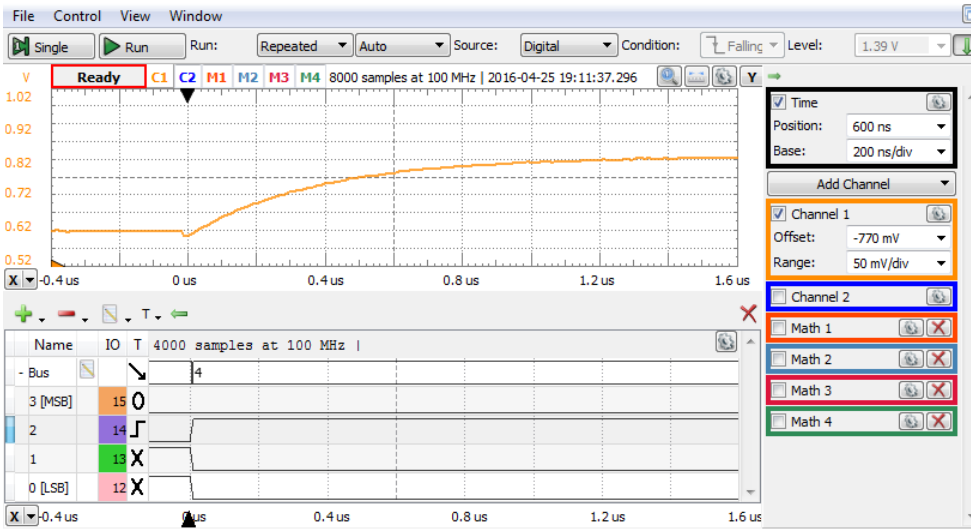


Figure 3.20 *DIO15…DIO12* transition *0011* to *0100*. Moderate Glitch

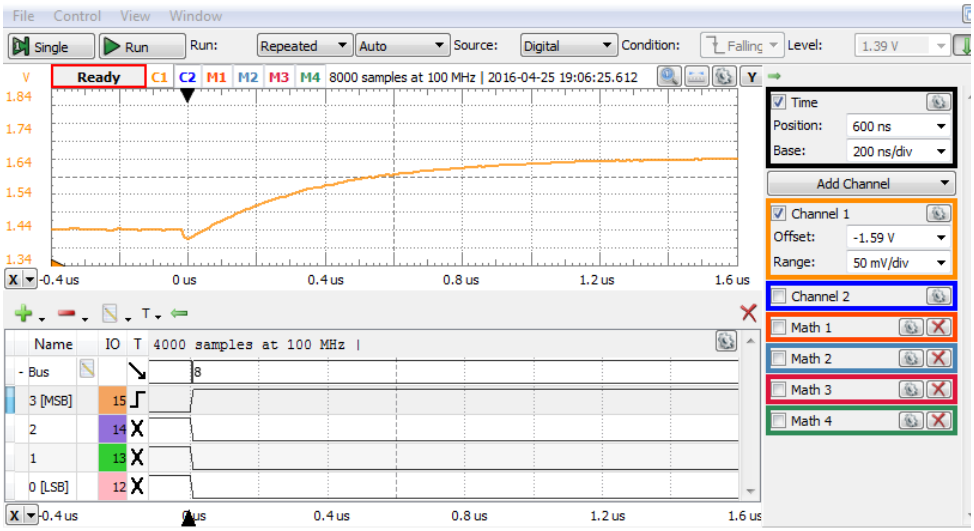


Figure 3.21 *DIO15…DIO12* transition *0111* to *1000*. Biggest Glitch

3.2 Binary ladder resistors network, voltage switching, no load

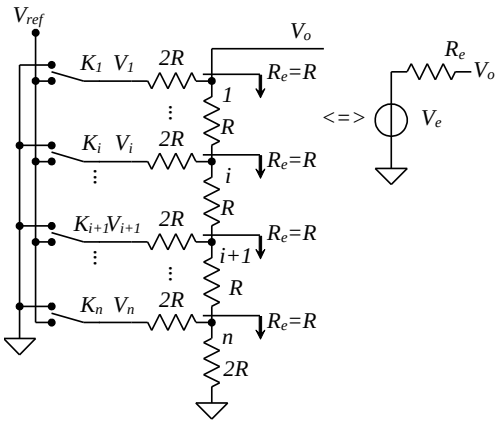


Figure 3.22 Ladder resistors network Thevenin model

3.2.1 Background

Only two resistor values are used in Figure 3.22: the generic value R and 2R. With the same conventions as in (3.1), (3.2) and (3.3), it is shown that the Thevenin model has same values as in (3.4) and (3.5):

$$R_e = R \qquad \textbf{(3.15)}$$

$$V_e = V_{ref} \cdot \sum_{i=1}^{n} a_i \cdot 2^{-i} = V_{ref} \cdot \{A\} \qquad \textbf{(3.16)}$$

With no load, the output voltage is:

$$V_o = V_e \qquad \textbf{(3.17)}$$

The full-scale voltage, V_{FS}, the absolute resolution, R_{abs}, and the voltage corresponding to the least significant bit, V_{LSB}, (n-bit, binary, unipolar) are identical to (3.7) and (3.8):

$$V_{FS} = V_o|\{A\} = 1 = V_{ref} \qquad \textbf{(3.18)}$$

$$R_{abs} = V_{LSB} = V_{FS} \cdot 2^{-n} \qquad \textbf{(3.19)}$$

3.2.2 Simulation

3.2.2.1 *Ideal circuit*

Figure 3.23 shows the Multisim schematic file for simulation. A binary counter, with 1kHz clock, generates the *{A}* numbers. The reference voltage (5V) and the switches are simulated by the 40163BT_5V output buffers:

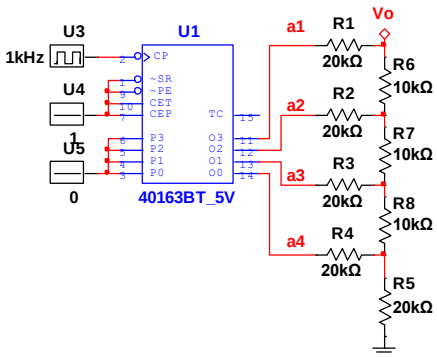


Figure 3.23 Ladder resistors network simulation schematic

$$R = 10k\Omega$$
$$n = 4$$
$$V_{ref} = V_{FS} = 5V \qquad \textbf{(3.20)}$$
$$R_{abs} = V_{LSB} = 312.5mV$$

Figure 3.24 shows the transient simulation results. This ideal graph is identical to the one in Figure 3.3 (ramp from *0V* to V_{FS}-V_{LSB}=*4.6875V*, V_{LSB}=*312.5mV*).

The cursors are set at *{A} =0* and *{A} =15LSB*, with *dy*=V_{FS}-V_{LSB}.

3.2.2.2 *Resistor mismatch induced errors*

For analyzing the resistor mismatch induced errors, a series of 9 transient simulations were done. For simulation number *k*, one resistor (R_k) was altered by the relative error of ε_R=*+1%*; The results shown in Figure 3.25 are similar to the ones for weighted resistor networks (Figure 3.5). Observations B, C and D of paragraph 3.1.2.2 apply here also.

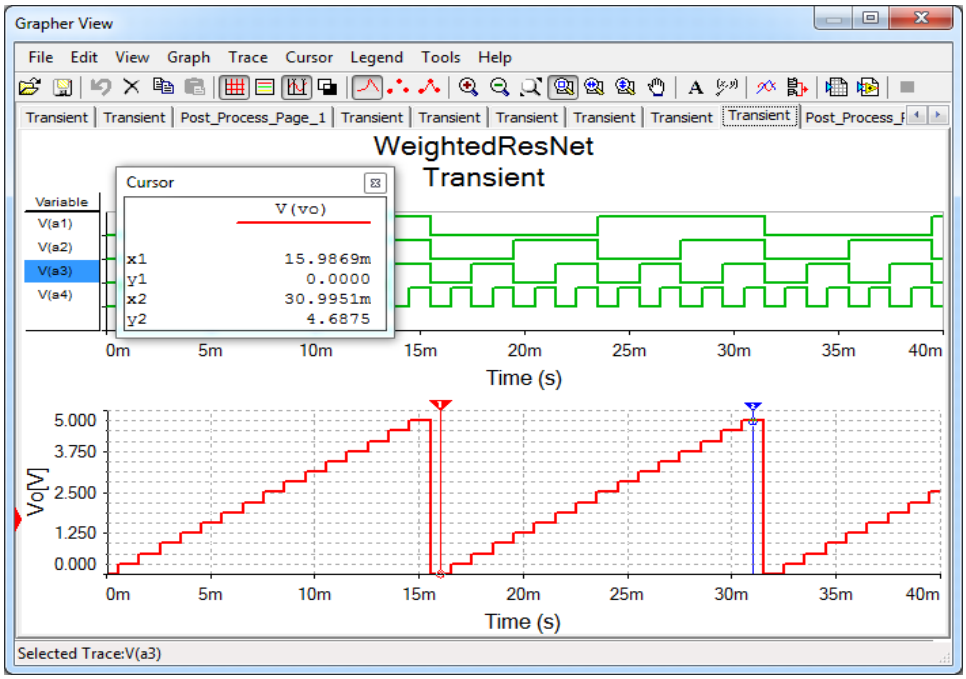


Figure 3.24 Ladder resistors network simulation results

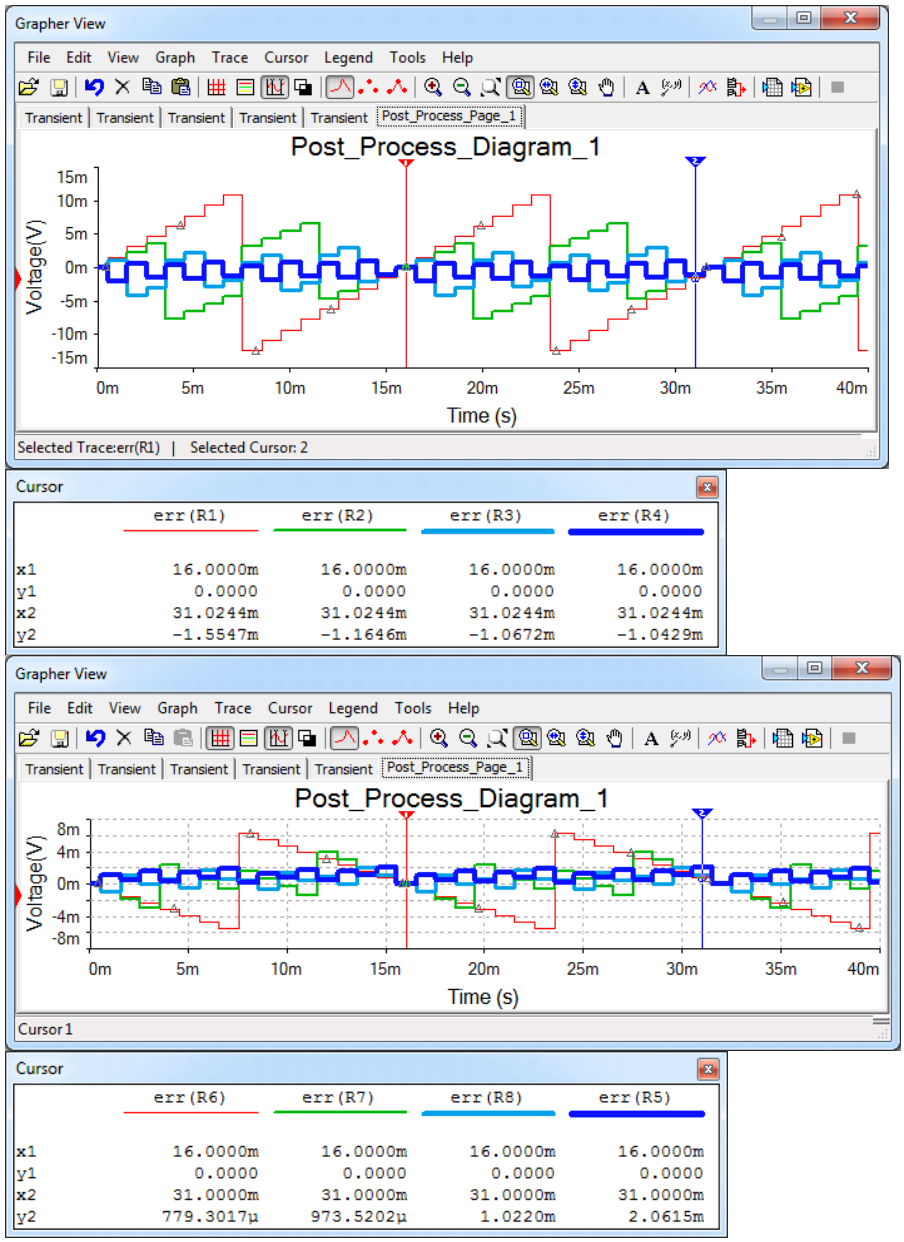


Figure 3.25 1% resistor mismatch errors (R1…R4 = up, R5…R8 = down)

3.2.2.3 *Vi induced errors*

For *Vi* induced errors, the ladder network is identical to the weighted network, so paragraph 3.1.2.3 applies here, including graphics and observations.

3.2.3 Experiment and measurements

The experiment uses the Resistor Networks lab board, as shown in Figure 3.8 and Figure 3.26.

The network to study includes *R7…R4*, *Rs7…Rs5*, and *Rc4*. The branches are connected to 4 pins of the Analog Discovery: *DIO7* = a_1=*MSB*, *DIO6* = a_2, *DIO5* = a_3, *DIO4* = a_4 = *LSB*.

To separate the output voltage, *V7* = V_o, no jumper should be loaded on *J3, J8, J9*.

The scope channel 1 is used to measure *V7*: pin 1 of *J8 (V7)* must be tied to pin 1 of *J2 (1+)*.

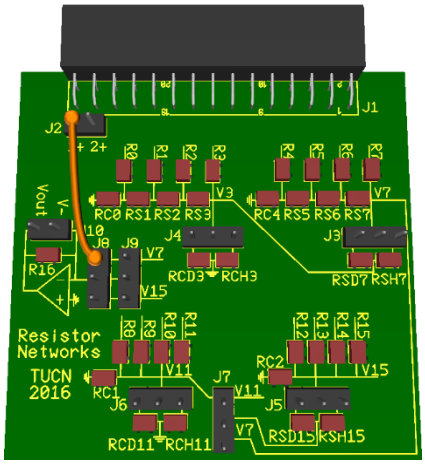


Figure 3.26 The Resistor Networks board

3.2.3.1 *Non-idealities*

Same non-idealities as in paragraph 3.1.3.1 apply here. The branch voltages approximate *Vcc=3.3V* for a_j=*1*, and V_{GND}= *0V*, for a_j=*0*. The approximations generate gain and linearity errors.

The digital signals, *DIOx*, are protected within the Analog Discovery with R_s = *220Ω* series resistors, which add to the branch resistance. The equivalent branch resistances are:

$$R7_e = R7 + R_s; \quad R6_e = R6 + R_s; \quad R5_e = R5 + R_s; \quad R4_e = R4 + R_s \tag{3.21}$$

Since only certain resistor values are available as discrete components, the equivalent resistances of the branches are not perfectly matching the ideal power of two sequence. This generates linearity and gain errors.

3.2.3.2 *Experiment*

Redo all the steps, experiments, measurements and tasks for the ladder network described in 3.2.3. Use *DIO7…DIO4* instead of *DIO15…DIO12*. Use *V7* instead of *V15*.

3.3 Combined resistors network, voltage switching, no load

3.3.1 Background

Each branch in Figure 3.27 is the Thevenin model of a subnetwork, with the equivalent impedance R_p and equivalent voltage:

$$V_i = \frac{V_{ref}}{2^m} \cdot \{d_i\} \qquad \textbf{(3.22)}$$

Each subnetwork can be considered a binary *DAC* of a digit $\{d_i\}$ of radix *r*, where d_i is an *m*-bit binary integer:

$$d_i = \sum_{j=0}^{m-1} 2^j \cdot b_{i,j} \qquad \textbf{(3.23)}$$

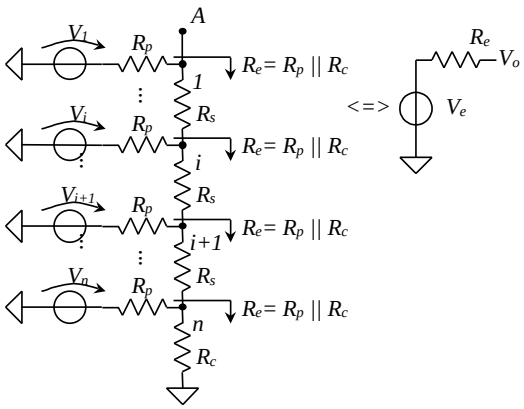


Figure 3.27 Combined resistors network Thevenin model.

Additional resistors R_s and R_c connect the subnetworks to get a radix *r* combined network. The design equations are:

A. The equivalent impedance from each node downwards is the same:

$$R_e = R_p||R_c = R_p||(R_s + R_p||R_c) <=> R_c = R_s + R_p||R_c \qquad \textbf{(3.24)}$$

B. The equivalent Thevenin voltage propagating from node *i* to node *i-1* is attenuated by factor *r*:

$$r = \frac{V_{e,i,i}}{V_{e,i-1,i}} = \frac{R_p + R_c}{R_p} \qquad \textbf{(3.25)}$$

The equation system (3.24), (3.25) can be solved for R_c and R_s:

$$R_c = (r-1) \cdot R_p; \qquad R_s = \frac{(r-1)^2}{r} \cdot R_p \qquad \textbf{(3.26)}$$

The combined network works as a radix *r* Digital to Analog Converter. The overall Thevenin model in Figure 3.27 has:

$$V_e = (r-1) \cdot \sum_{i=1}^{n} \frac{V_i}{r^i} = \frac{V_{ref}}{2^m} \cdot (r-1) \cdot \sum_{i=1}^{n} \{d_i\} \cdot r^{-i} =$$
$$= V_{ref} \cdot \frac{r-1}{2^m} \cdot \sum_{i=1}^{n} \sum_{j=0}^{m-1} r^{-i} \cdot 2^j \cdot b_{i,j} = V_{ref} \cdot \frac{r-1}{2^m} \cdot \{A\}_r \qquad \textbf{(3.27)}$$

$$R_e = R_p||R_c = \frac{r-1}{r} \cdot R_p \qquad \textbf{(3.28)}$$

Where $\{A\}_r$ is an n-digit, fractional, unipolar number written in radix r:

$$\{A\}_r = 0.d_1 \ldots d_{(n-1)}d_n = \sum_{i=1}^{n} d_i \cdot r^{-i} = \sum_{i=1}^{n}\sum_{j=0}^{m-1} r^{-i} \cdot 2^j \cdot b_{i,j} \quad \textbf{(3.29)}$$

V_{FS} and $V_{LSB} = R_{abs}$ are the V_e values for $\{A\}=1$, respectively $\{A\}=r^{-n}$:

$$V_{FS} = V_{ref} \cdot \frac{r-1}{2^m}$$
$$V_{LSB} = V_{ref} \cdot \frac{r-1}{2^m \cdot r^n} = R_{abs} \quad \textbf{(3.30)}$$

3.3.1.1 *Hexadecimal combined resistor network*

$$R_p = R; \quad m = 4; \quad r = 16$$
$$R_c = 15 \cdot R_p = 15 \cdot R$$
$$R_s = \frac{15^2}{16} \cdot R_p = 14.0625 \cdot R$$
$$R_e = \frac{15}{16} \cdot R_p = 0.9375 \cdot R$$
$$V_e = V_{ref} \cdot \frac{15}{16} \cdot \{A\}_r \quad \textbf{(3.31)}$$
$$V_{FS} = V_{ref} \cdot \frac{15}{16}$$
$$V_{LSB} = V_{ref} \cdot \frac{15}{16^{n+1}} = R_{abs}$$

3.3.1.2 *Decimal combined resistor network*

$$R_p = R; \quad m = 4; \quad r = 10$$
$$R_c = 9 \cdot R_p = 9 \cdot R$$
$$R_s = \frac{9^2}{10} \cdot R_p = 8.1 \cdot R$$
$$R_e = \frac{9}{10} \cdot R_p = 0.9 \cdot R$$
$$V_e = V_{ref} \cdot \frac{9}{16} \cdot \{A\}_r \quad \textbf{(3.32)}$$
$$V_{FS} = V_{ref} \cdot \frac{9}{16}$$
$$V_{LSB} = V_{ref} \cdot \frac{9}{16 \cdot 10^n} = R_{abs}$$

3.3.2 Simulation

3.3.2.1 *Hexadecimal*

Figure 3.28 shows the Multisim schematic file for simulating a 2-digit, *Hexadecimal* combined network. Ladder 4-bit subnetworks are used but weighted or a combination of ladder and weighted subnetworks can be equally used. A binary counter, with 1kHz clock, generates the *{A}* numbers. The reference voltage (5V) and the switches are simulated by the 40163BT_5V output buffers:

$$R = 10k\Omega; \quad n = 2; \quad V_{ref} = 5V$$
$$V_{FS} = 4.6875V \qquad \textbf{(3.33)}$$
$$R_{abs} = V_{LSB} = 18.310546875mV$$

Figure 3.29 shows the transient simulation results. The digital graph (up) shows the bits of the *{A}* number (two-digit BCD counter), the analog graph (down) shows the V_o voltage (ramp from *0V* to V_{FS}-V_{LSB} = *4.669189453125V*, in steps of V_{LSB}). The cursors are set at *{A}=0* and *{A}=255LSB*, with *dy*=V_{FS}-V_{LSB}.

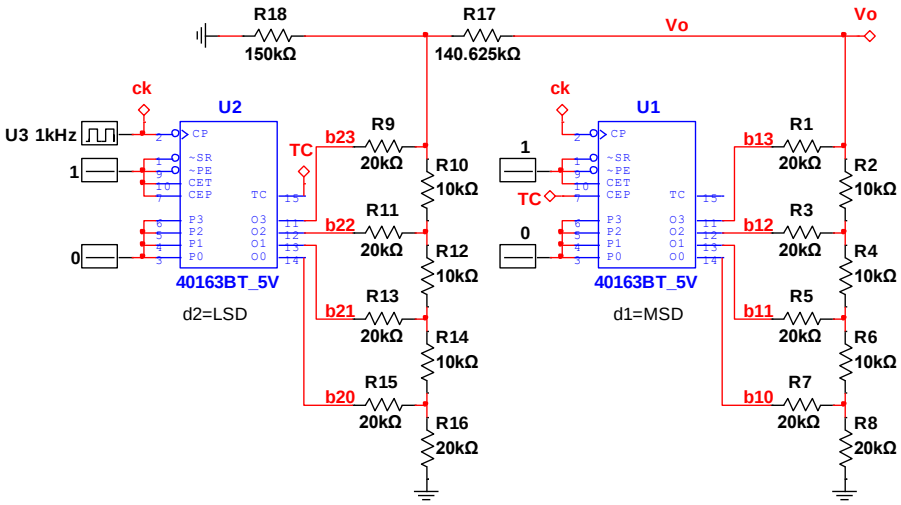


Figure 3.28 2-digit Hexadecimal Combined Network schematic

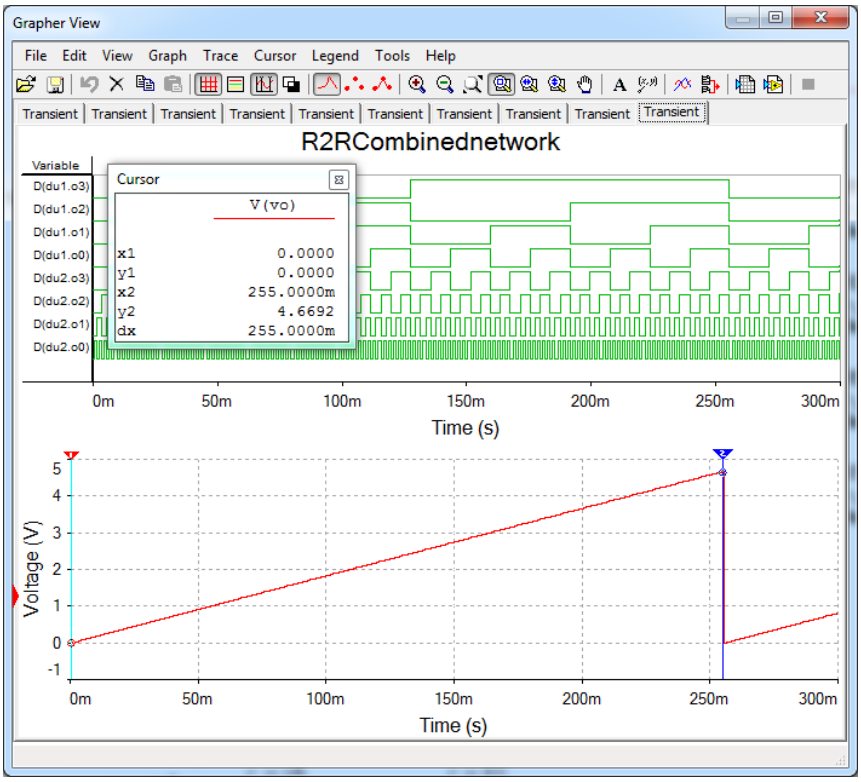


Figure 3.29 2-digit Hexadecimal Combined Network simulation results

3.3.2.2 *BCD*

Figure 3.30 shows the Multisim schematic file for simulating a 2-digit, BCD combined network. Compared to Figure 3.28, U1 and U2 are replaced by BCD counters 40162BT_5V and R17 and R18 changed values as in **(** 3.32 **)**.

$$R = 10k\Omega; \quad n = 2; \quad V_{ref} = 5V$$
$$V_{FS} = 2.8125V \qquad \textbf{(3.34)}$$
$$R_{abs} = V_{LSB} = 28.125mV$$

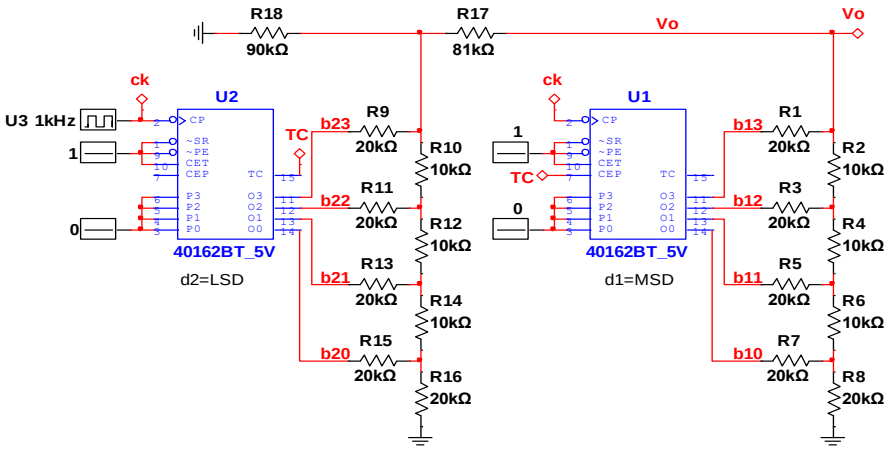


Figure 3.30 2-digit BCD Combined Resistor Network schematic

Figure 3.31 shows the transient simulation results: the bits of the *{A}* number (two-digit BCD counter) (up) and the V_o voltage (ramp from *0V* to V_{FS}-V_{LSB} *=2.784375V*, in steps of V_{LSB}*= 28.125mV*) (down). The cursors are set at *{A}=0* and *{A}=99LSB*, with $dy=V_{FS}-V_{LSB}$.

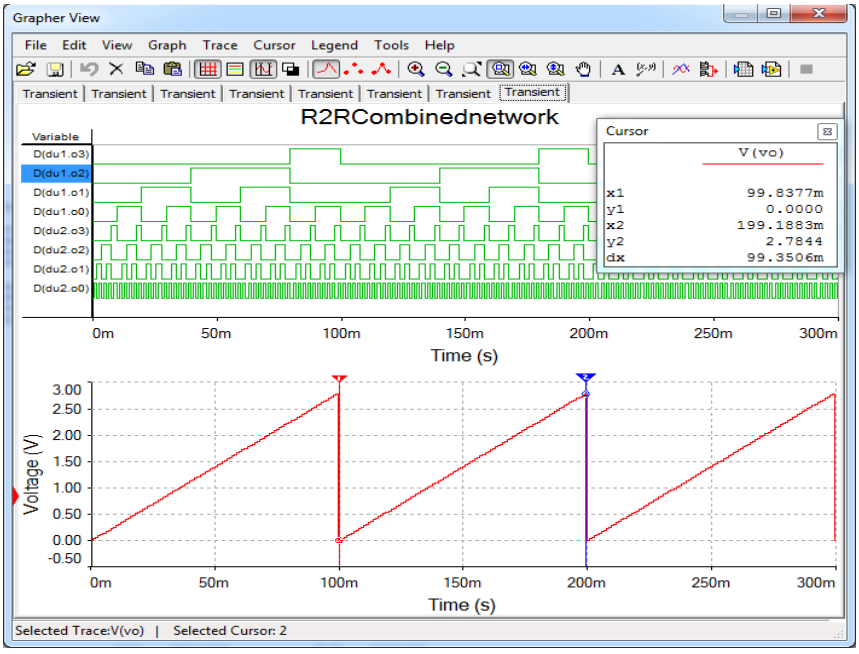


Figure 3.31 2-digit BCD Combined Resistor Network simulation results

Figure 3.32 simulates a BCD network (as in Figure 3.30), driven by hexadecimal counters (40163). This is a non-typical situation: a BCD DAC should never get input digits higher than *9*, since the digits are weighted *10:1*. However, each digit (*d2=b23…b20, d1=b13…b10*) gets hexadecimal values of *0…F*. Consequently, there are multiple input values which result in the same output voltage value:

$$\begin{matrix} V_e(0A_h) = V_e(10_h); & V_e(0B_h) = V_e(11_h); & V_e(0C_h) = V_e(12_h)\ldots \\ V_e(1A_h) = V_e(20_h); & V_e(1B_h) = V_e(21_h); & V_e(1C_h) = V_e(22_h)\ldots \end{matrix} \quad \textbf{(3.35)}$$

This explains the non-monotonic "saw-tooth" aspect of the ramp.

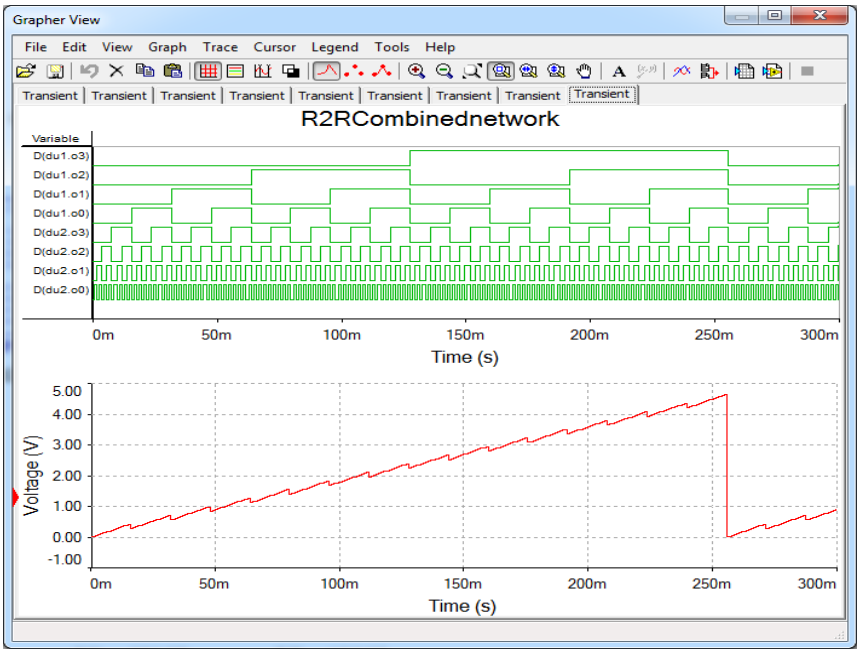


Figure 3.32 BCD combined network driven by hexadecimal code

3.3.3 Experiment and measurements

The experiment uses the Resistor Networks lab board, as shown in Figure 3.8 and Figure 3.33.

The network to study includes *R7…R0*, *Rs7…Rs1*, *Rc4* and *Rc0*. The branches are connected to 8 pins of the Analog Discovery: *DIO7* = a_1=*MSB, … DIO0* = a_8 = *LSB.*

To separate the output voltage, *V7* = V_o, no jumper should be loaded on headers *J8, J9.*

The scope channel 1 is used to measure *V7*: pin 1 of *J8 (V7)* must be tied to pin 1 of *J2 (1+).*

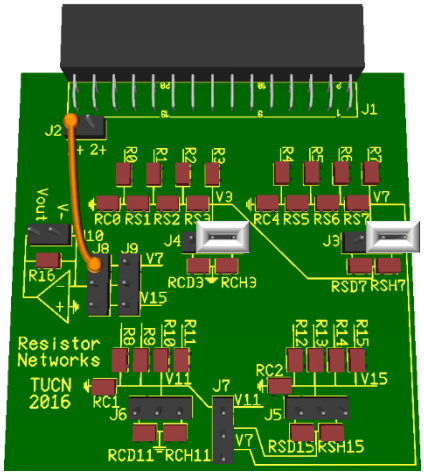


Figure 3.33 The Resistor Networks board – two-digit hexadecimal with ladder subnetworks

3.3.3.1 *Non-idealities*

All the non-idealities shown in paragraph 3.1.3.1 are present here also and generate similar errors.

3.3.3.2 *Hexadecimal experiment*

Load jumpers on J3 and J4 as shown in Figure 3.33, to select RSH7 as series resistor and RCH3 as closing resistor. That builds a two-digit hexadecimal DAC from the two ladder subnetworks.

In the *Patterns* Generator, add a bus of bits DIO7…DIO0, set it as *Binary Counter, Push-Pull*, with frequency = *1kHz*. Run Patterns Generator.

Set the scope as in Figure 3.34. Run the scope. Observe the graph. Compare to Figure 3.29. Notice that for the experimental board, the ideal values are:

$$R = 10k\Omega; \quad n = 4; \quad V_{ref} = 3.3V$$
$$V_{FS} = 3.09375V \qquad \textbf{(3.36)}$$
$$R_{abs} = V_{LSB} = 12.0849609375mV$$

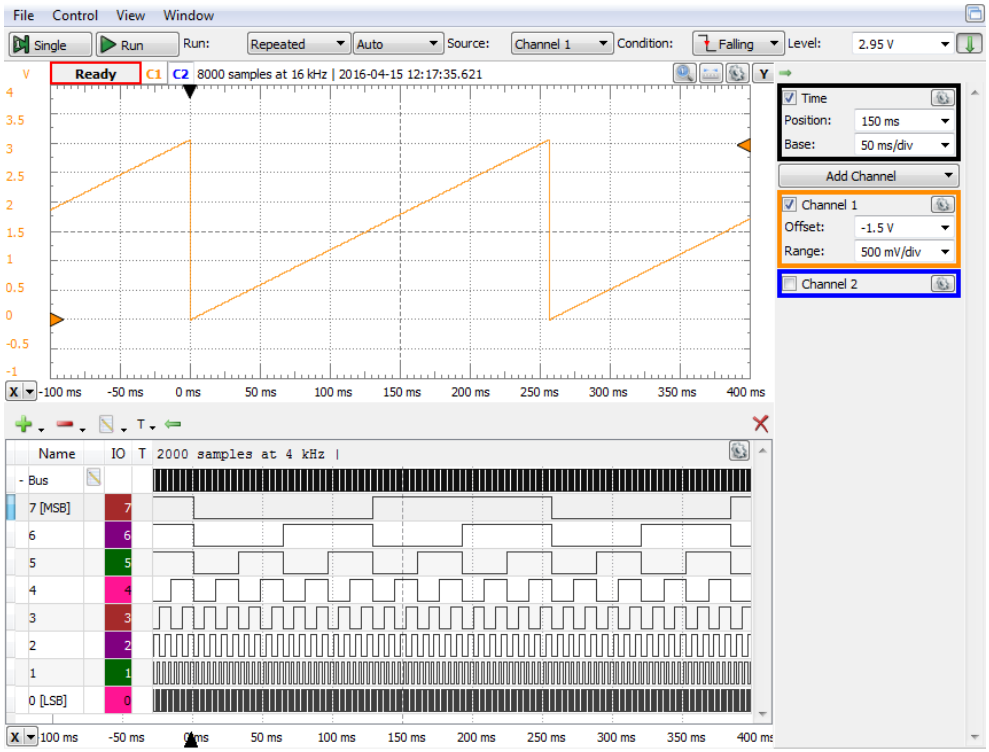


Figure 3.34 The weighted resistor network output voltage ramp

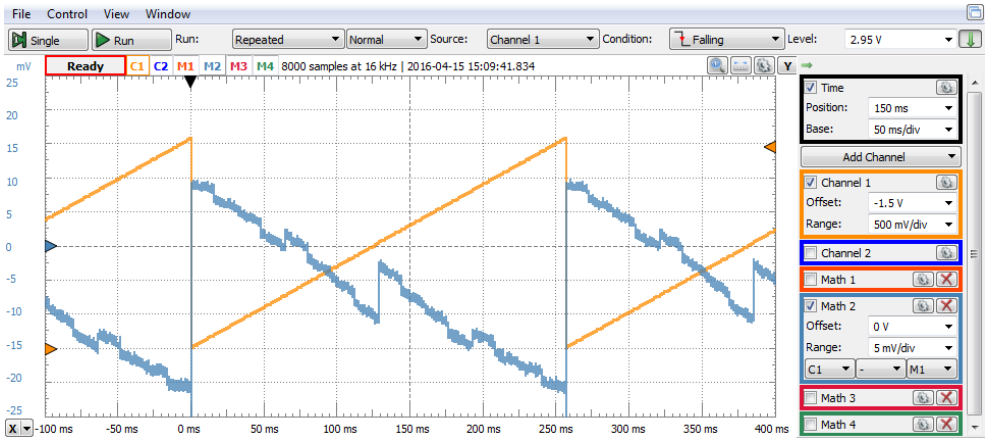


Figure 3.35 The hexadecimal combined resistor network error

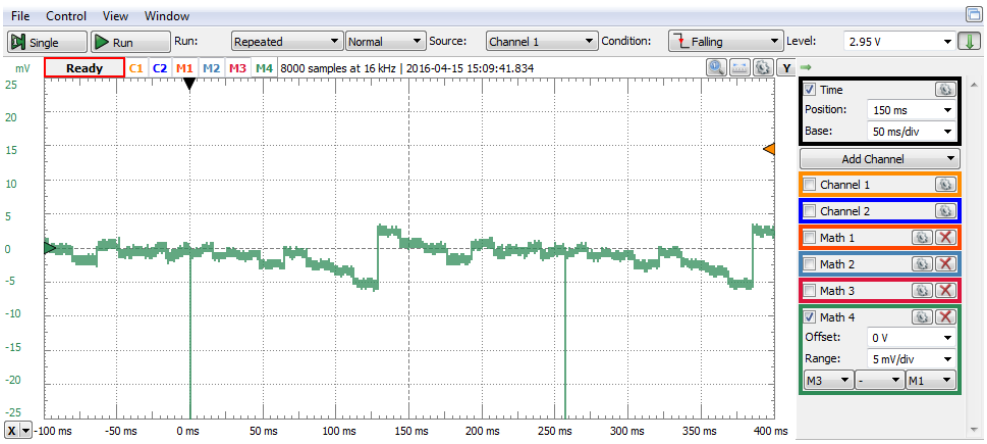


Figure 3.36 Linearity after compensating gain and offset errors

Task 6. Compute the **absolute** and **relative integral** and **differential errors** of your combined resistor network. Verify if your DAC is **monotonic**.

Hint 1: build Math channels 1…4, similar to paragraph 3.1.3.3. Modify the constant values to match the current experiment.

Hint 2: compared to Figure 3.12…Figure 3.17, in Figure 3.35 and Figure 3.36 there are many more transitions in a single saw tooth period: *256* transitions in *256ms*. Each transition generates a dynamic spike, as explained in Figure 3.13. The magnitude and density of the spikes make *Math 2* and *Math 4* look "noisy" signals, and their actual DC level is difficult to estimate. To overcome this effect, you can do either or both workarounds below:

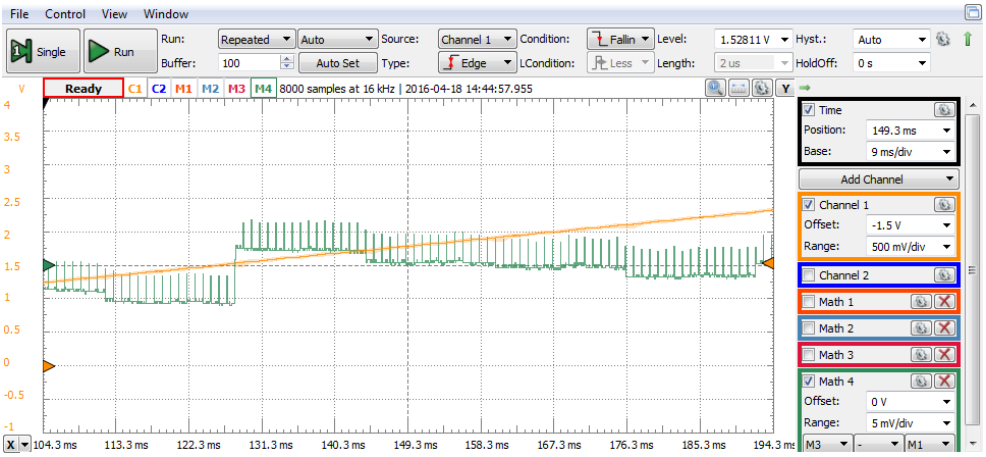


Figure 3.37 Identifying transition-generated spikes in the error graph

A. Run the scope in "Single" mode. Repeat "Single" runs, to get a clean (cleaner) image of Math 2 and Math 4.
B. After running scope in "Single" mode, change the scope time base to enlarge the image, as in Figure 3.37. You will be able to identify the dynamic spikes. Coming back to the original time base, you will be able to ignore the spikes in your measurements/adjustments.

3.3.3.3 *Decimal experiment*

Load jumpers on J3 and J4 as shown in Figure 3.38, to select *RSD7* as series resistor and *RCD3* as closing resistor. That builds a two-digit BCD DAC from the two ladder subnetworks.

Set an 8-bit *BCD Counter* on DIO7…DIO0. Connect the scope as in Figure 3.38. The voltage *V7* will be shown on channel 1 of the Scope.

Set the scope as in Figure 3.39. Run the scope. Observe the graph. Compare to Figure 3.31. Notice that for the experimental board, the ideal values are:

$$R = 10k\Omega; n = 4; V_{ref} = 3.3V$$
$$V_{FS} = 1.85625V \qquad \textbf{(3.37)}$$
$$R_{abs} = V_{LSB} = 18.5625mV$$

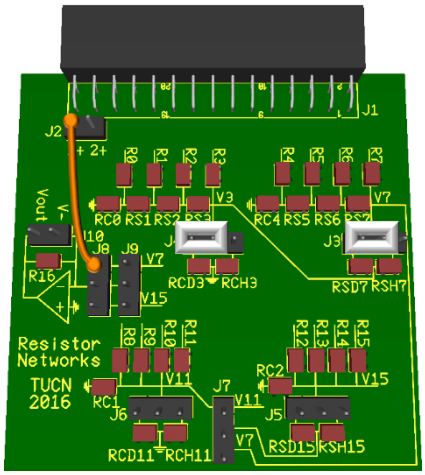


Figure 3.38 The Resistor Networks board – two-digit BCD, with ladder subnetworks

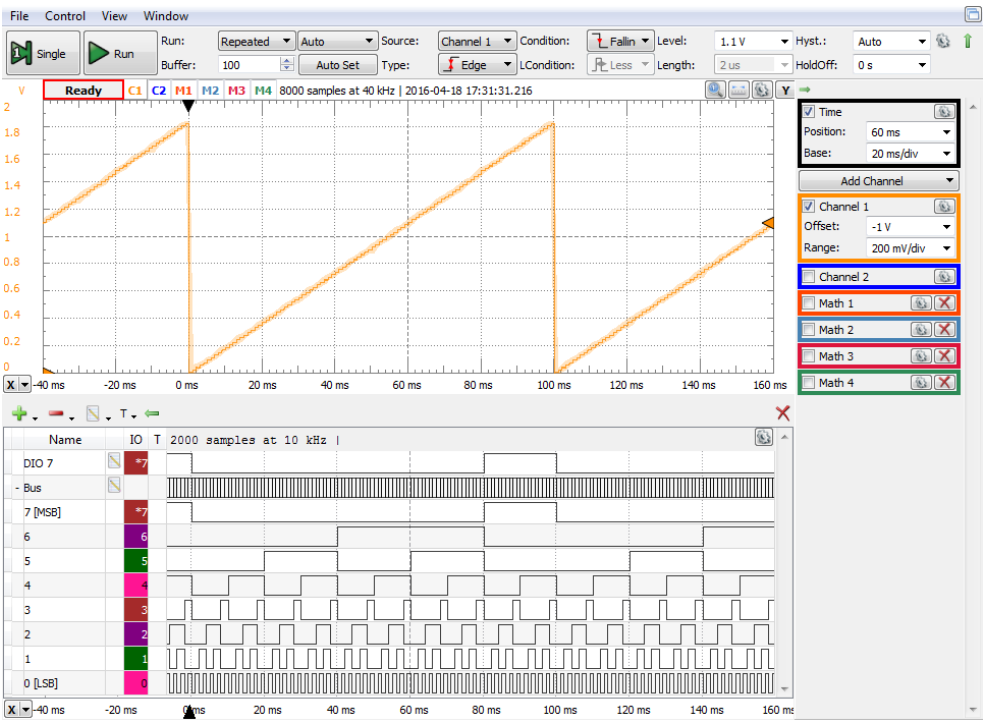


Figure 3.39 The weighted resistor network output voltage ramp

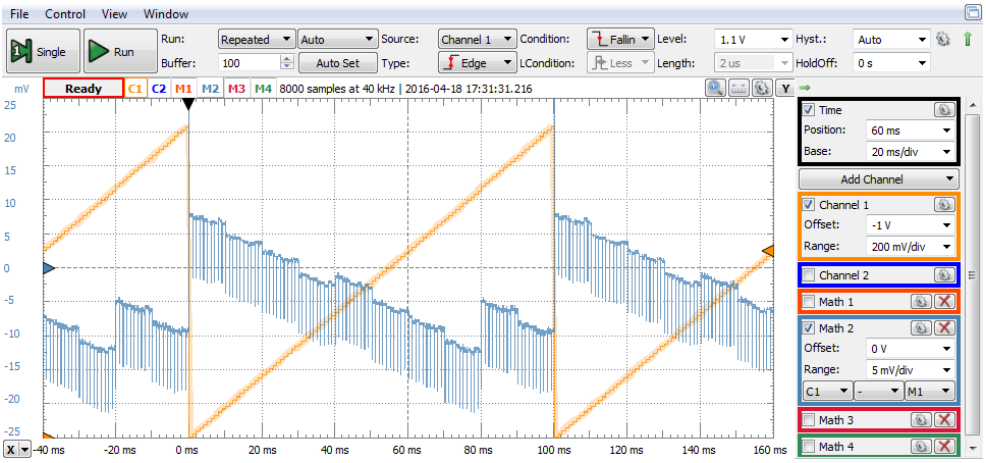


Figure 3.40 The BCD combined resistor network error

Figure 3.41 Linearity after compensating gain and offset errors

Task 7. Compute the **absolute** and **relative integral** and **differential errors** of your weighted resistor network. Verify if your DAC is **monotonic**.

For the task above, use Figure 3.40 and Figure 3.41 as references. Consider the Hints in the previous paragraph.

3.3.3.4 *Decimal network forced to hexadecimal number*

Load jumpers on J3 and J4 as shown in Figure 3.38, to build a two-digit BCD DAC, as in the previous paragraph. However, set the Pattern Generator to a hexadecimal counter DIO7…DIO0, as in paragraph 3.3.3.2. This combination generates the non-typical situation simulated in Figure 3.32. The scope image is shown in Figure 3.43.

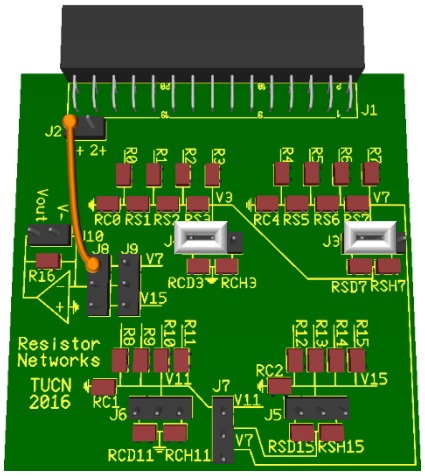


Figure 3.42 The Resistor Networks board – two-digit BCD, with ladder subnetworks

Figure 3.43 BCD combined network driven by hexadecimal code experiment

3.3.3.5 *Combined network with weighted resistor subnetworks*

All experiments in 3.3.3.2 … 3.3.3.4 can be repeated with weighted subnetworks *R15…R12, Rc2* and *R11…R8, Rc1*.

Load jumper on J7, as in Figure 3.44, to cascade the two weighted subnetworks in a combined network. Connect *V15* to scope channel 1 (pin 1+ of J2).

For hexadecimal experiments, load jumpers on J5 and J6 as shown in Figure 3.44, to select *RSH15* as series resistor and *RCH11* as closing resistor. That builds a two-digit hexadecimal DAC from the two weighted subnetworks. Set the Pattern Generator to drive *DIO15…DIO8* as a two-digit hexadecimal counter (in fact, a single 8-bit binary counter).

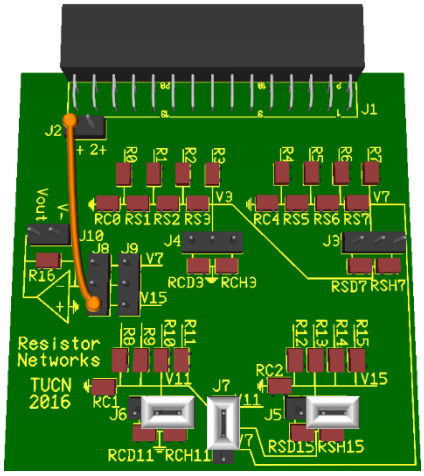


Figure 3.44 The Resistor Networks board – two-digit hexadecimal, with weighted subnetworks

For BCD experiments, move the jumpers on J5 and J6 to left, to select *RSD15* as series resistor and *RCD11* as closing resistor. That builds a two-digit BCD DAC from the two weighted subnetworks. Set *DIO15…DIO8* as a two-digit BCD counter.

3.3.3.6 *3-digit combined network with mixed resistor subnetworks*

To build a 3-digit combined network, load a jumper on J7, as shown in Figure 3.45. This connects *R15…R12, Rc2* as the most significant subnetwork, in front of *R7…R4* and *R3…R0*.

For 3-digit hexadecimal (12-bit binary) network, place jumpers on J3, J4 and J5, as shown in Figure 3.45, to set *RSH15* and *RSH7* as series resistors, respectively *RCH3* as closing resistor.

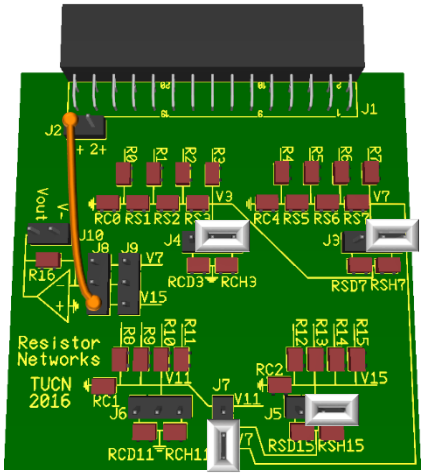


Figure 3.45 The Resistor Networks board – three-digit hexadecimal

Set a 12-bit binary counter in the Pattern Generator, with the bits in the following order (from MSB to LSB): *DIO15, DIO14, DIO13, DIO12, DIO7, DIO6, DIO5, DIO4, DIO3, DIO2, DIO1, DIO0*, as shown in Figure 3.46.

Set the scope as in Figure 3.47. Modify the *Math1* script for the current time base and number of states/ramp period:

*floor(Time*1000)*3.09375/4096*

Observe the output voltage *V15* and the total error as in Figure 3.47. Compensate the offset and gain errors in *Math3* and observe the linearity error in *Math4*, in Figure 3.48.

Build a 3-digit BCD network, by moving to right the jumpers on J3, J4 and J5, to set *RSD15* and *RSD7* as series resistors, respectively *RCD3* as closing resistor. In the *Patterns Generator*, set a *BCD* counter to drive this circuit.

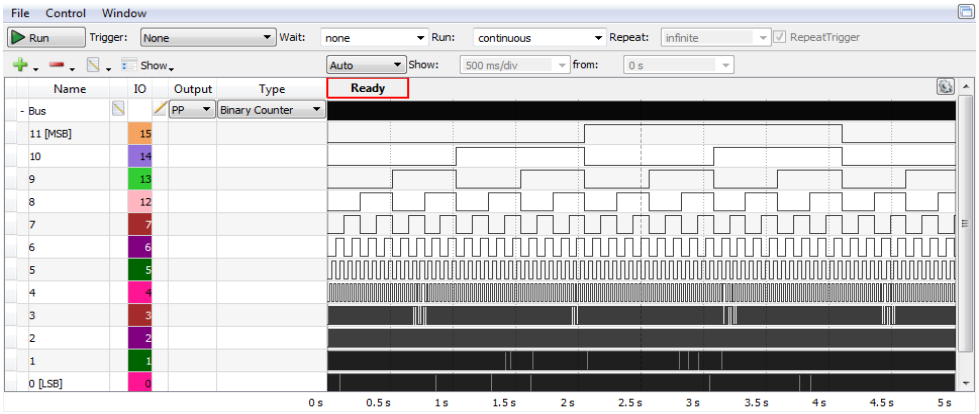


Figure 3.46 Pattern Generator 12-bit binary counter

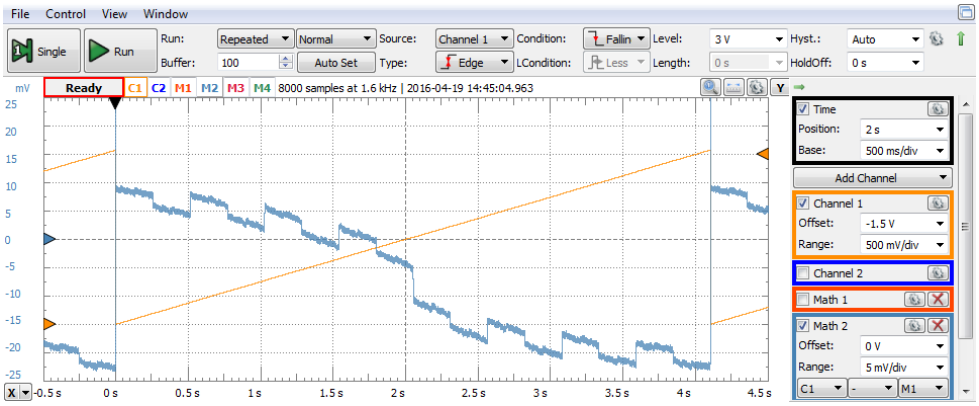


Figure 3.47 3-digit hexadecimal network output voltage and errors

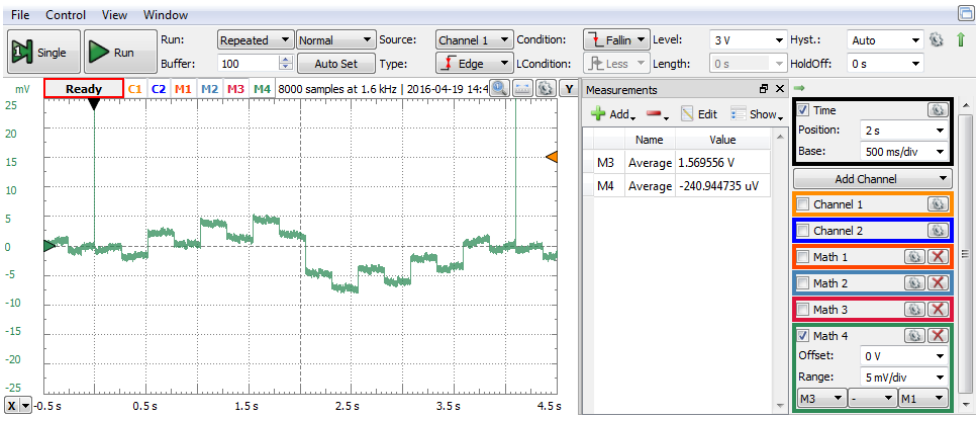


Figure 3.48 3-digit hexadecimal network linearity error

3.4 Operational Amplifier Output Stage for Voltage Switching Resistor Networks

3.4.1 Background

An inverting operational amplifier output stage can be attached to any of the resistor networks analyzed above, as in Figure 3.49. Due to the negative feedback, the network output node is shortcut to the virtual ground, V_-:

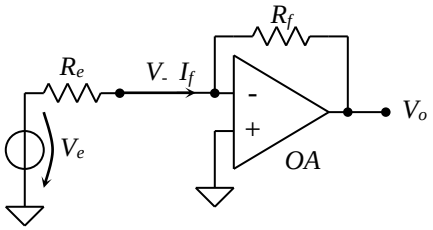


Figure 3.49 Operational amplifier output stage for voltage switching resistor networks

$$V_- = 0 \qquad \textbf{(3.38)}$$

$$I_f = \frac{V_e}{R_e} \qquad \textbf{(3.39)}$$

$$V_o = -V_e \cdot \frac{R_f}{R_e} \qquad \textbf{(3.40)}$$

The operational amplifier provides low impedance for the output signal V_o, but brings additional non-idealities, errors and potential issues:

A. Offset error (operational amplifier offset voltage multiplied by the stage gain).
B. Gain error (mostly due to R_f/R_e ratio, less due to finite operational amplifier gain).
C. Bandwidth limitations (usually a constant gain times bandwidth product).
D. Slew Rate limitations.
E. Input and output voltage range limitations (usually related to the supply voltages).
F. Stability (depending on the stage gain and load).
G. Noise (own thermal noise and propagated from supply voltages and adjacent signals).
H. Drifts (the parameters change with temperature and age).

For both weighted and ladder, *n-bit*, unipolar networks, from (3.4), (3.5), (3.15), (3.16):

$$R_e = R; \quad V_e = V_{ref} \cdot \sum_{i=1}^{n} a_i \cdot 2^{-i} = V_{ref} \cdot \{A\} \qquad \textbf{(3.41)}$$

$$V_o = -V_{ref} \cdot \frac{R_f}{R} \cdot \sum_{i=1}^{n} a_i \cdot 2^{-i} = -V_{ref} \cdot \frac{R_f}{R} \cdot \{A\} \qquad \textbf{(3.42)}$$

$$V_{FS} = -V_{ref} \cdot \frac{R_f}{R}; \quad V_{LSB} = -\frac{V_{ref}}{2^n} \cdot \frac{R_f}{R} \qquad \textbf{(3.43)}$$

For *n-digit* combined unipolar networks of radix *r*, from (3.27), (3.28):

$$V_e = V_{ref} \cdot \frac{r-1}{2^m} \cdot \{A\}_r; \quad R_e = \frac{r-1}{r} \cdot R_p \qquad \textbf{(3.44)}$$

$$V_o = -V_{ref} \cdot \frac{r}{2^m} \cdot \frac{R_f}{R_p} \cdot \sum_{i=1}^{n} d_i \cdot r^{-i} = -V_{ref} \cdot \frac{r}{2^m} \cdot \frac{R_f}{R_p} \cdot \{A\}_r \qquad \textbf{(3.45)}$$

$$V_{FS} = -V_{ref} \cdot \frac{r}{2^m} \cdot \frac{R_f}{R_p}; \quad V_{LSB} = -\frac{V_{ref}}{r^n} \cdot \frac{r}{2^m} \cdot \frac{R_f}{R_p} \qquad \textbf{(3.46)}$$

For *n-digit* unipolar hexadecimal combined networks, *m=4, r=16*:

$$V_e = V_{ref} \cdot \frac{15}{16} \cdot \{A\}_r; \quad R_e = \frac{15}{16} \cdot R_p \qquad \textbf{(3.47)}$$

$$V_o = -V_{ref} \cdot \frac{16}{16} \cdot \frac{R_f}{R_p} \cdot \sum_{i=1}^{n} d_i \cdot 16^{-i} = -V_{ref} \cdot \frac{R_f}{R_p} \cdot \{A\}_r \qquad \textbf{(3.48)}$$

$$V_{FS} = -V_{ref} \cdot \frac{R_f}{R_p}; \quad V_{LSB} = -\frac{V_{ref}}{16^n} \cdot \frac{R_f}{R_p} \qquad \textbf{(3.49)}$$

For *n-digit* unipolar BCD combined networks, *m=4, r=10*:

$$V_e = V_{ref} \cdot \frac{9}{16} \cdot \{A\}_r; \quad R_e = \frac{9}{10} \cdot R_p \qquad \textbf{(3.50)}$$

$$V_o = -V_{ref} \cdot \frac{10}{16} \cdot \frac{R_f}{R_p} \cdot \sum_{i=1}^{n} d_i \cdot 10^{-i} = -V_{ref} \cdot \frac{10}{16} \cdot \frac{R_f}{R_p} \cdot \{A\}_r \qquad \textbf{(3.51)}$$

$$V_{FS} = -V_{ref} \cdot \frac{10}{16} \cdot \frac{R_f}{R_p}; \quad V_{LSB} = -\frac{V_{ref}}{10^n} \cdot \frac{10}{16} \cdot \frac{R_f}{R_p} \qquad \textbf{(3.52)}$$

3.4.2 Simulation

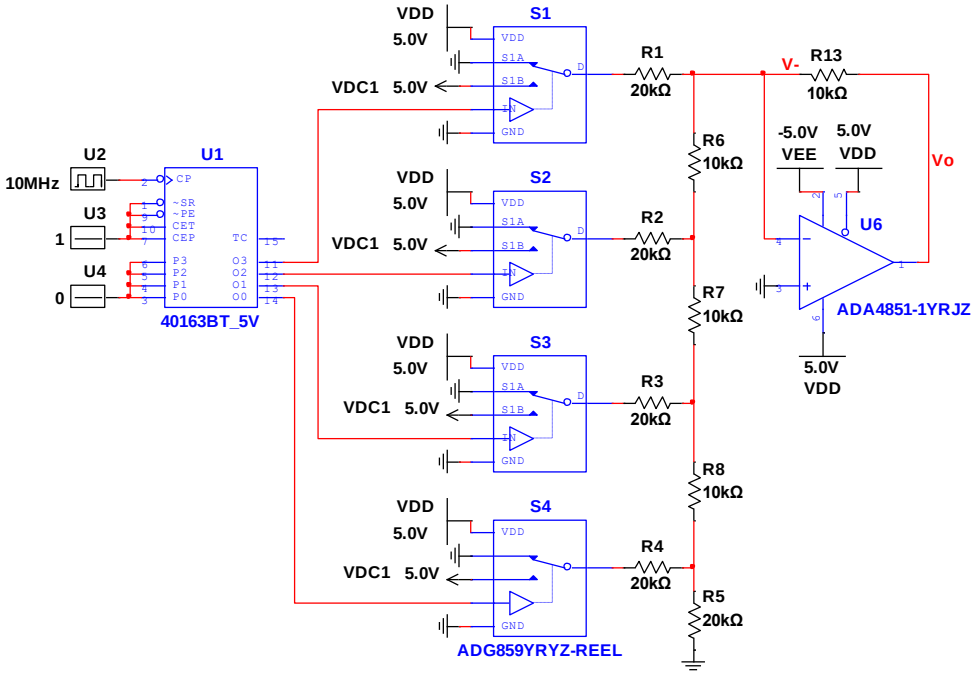


Figure 3.50 Ladder resistor network with operational amplifier output stage

Figure 3.50 shows the Multisim schematic file for simulation. A weighted or combined network could be as well used instead of the shown ladder network. A 10MHz clock and a fast time base are used to observe fast dynamic artefacts.

The noticeable parameters of ADA 4851 OpAmp are:

- V_{supp}: $+3 \ldots \pm 5V$
- *Output swing: 60mV to either rail*
- *Input common mode:* V_{suppN} *- 0.2V…*V_{suppP} *- 2.2V*
- *Slew Rate: 375V/us (specific conditions)*
- *Bandwidth: 130MHz*

The simulation results are shown in Figure 3.51. V_o is negative since the output stage is inverter. The settling time and slew rate limit the speed of V_o, for every V_{LSB} step. The biggest delay happens when the input number rolls over and V_o jumps from almost V_{FS} to *0*.

V_-, the voltage in the inverter pin of the operational amplifier is not constant null, as expected in a negative feedback stage.

$$V_- = V_o + R_f \cdot I_f \qquad \textbf{(3.53)}$$

The operational amplifier in negative feedback should generate the needed value of V_o to keep V_- null in (3.53). However, the slew rate and settling time limitations do not allow V_o to change fast enough, and generate the pulses on V_-, visible in Figure 3.51. Glitches are too fast to be seen on V_o, however they exist in I_f and can be seen in V_-.

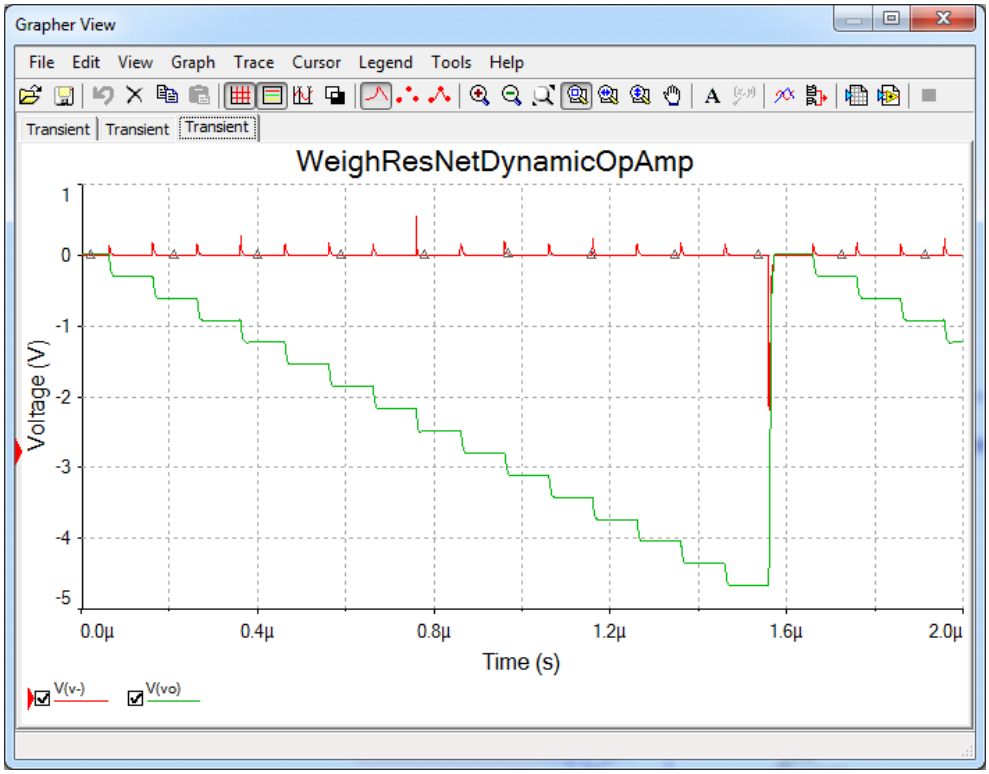


Figure 3.51 Weighted resistor network with operational amplifier output stage simulation results

3.4.3 Experiment and measurements

The 4-bit ladder resistor network *R7…R4, Rs7…Rs5, Rc4* is connected in Figure 3.52 to the inverting operational amplifier stage, with a jumper on *J9*, position *V7*. V_{out} and *V-* are probed by channels *1* and *2* of the scope.

The Pattern Generator drives *DIO7…DIO4* as a 4-bit binary counter with a clock of 10MHz.

The signals in Figure 3.53 are not identical to the simulation above. The Analog Discovery scope probe impedance (1MΩ || 24pF), the probe wires inductance and the crosstalk from digital signals were not considered in simulation but do affect the experiment.

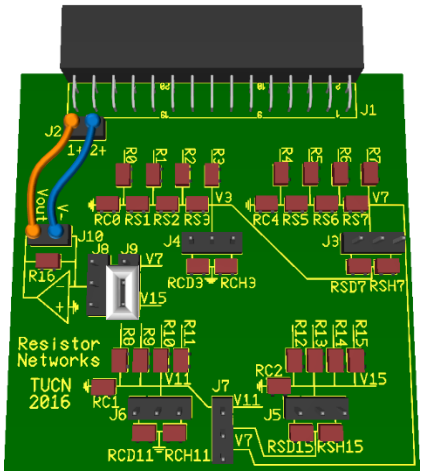


Figure 3.52 The Resistor Networks board – weighted resistor network with OpAmp output stage

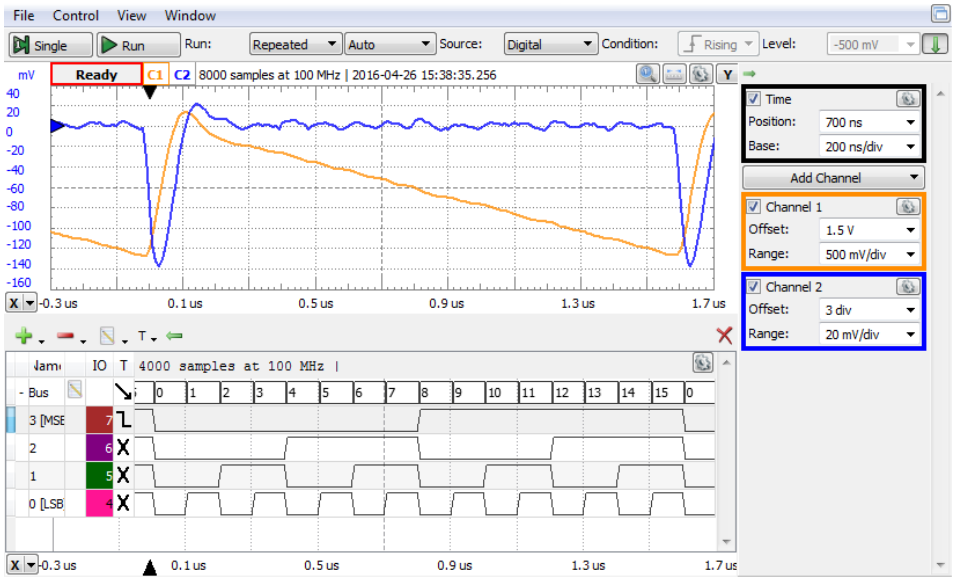


Figure 3.53 Operational amplifier output stage experiment

The scope probes not only modify the probed signal shapes, but also influence the overall circuit. Removing the scope probe from V_- circuit node, increases the overall V_{out} bandwidth, as in Figure 3.54.

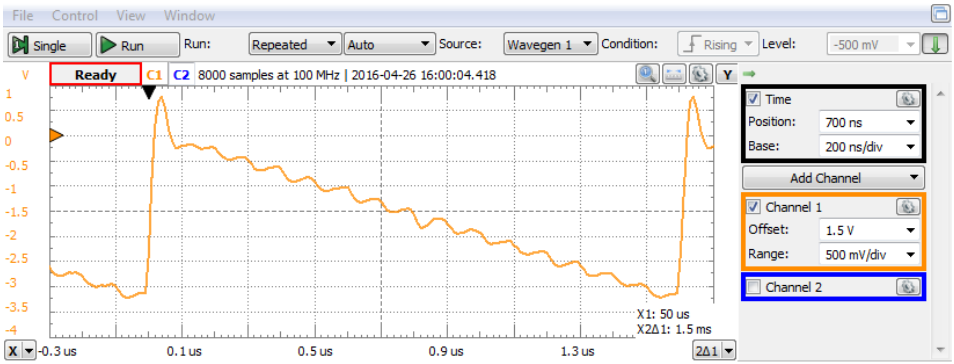


Figure 3.54 Operational amplifier output stage experiment (V_- probe removed)

Task 8. Based on the acquisition in Figure 3.54, measure the apparent slew rate of the operational amplifier.

Hint: modify the scope time base and horizontal position, to enlarge the steepest slope (at digital rollover). Place vertical cursors across it. Click *View/X Cursors*. Read $\Delta y/\Delta x$ in the *X Cursors* Window. Notice that the actual Slew Rate, measured for a particular set of conditions, might be significantly different compared to the standard measured value in the Operational amplifier data sheet.

At low frequency, the Analog Discovery probe and input stage dynamic limitations are less visible, as in Figure 3.55. The operational amplifier offset error increases the overall offset, compared to Figure 3.12.

Offset, gain and linearity errors can be computed in Figure 3.55, similarly to paragraph 3.1.3.3, Figure 3.14 and Figure 3.15.

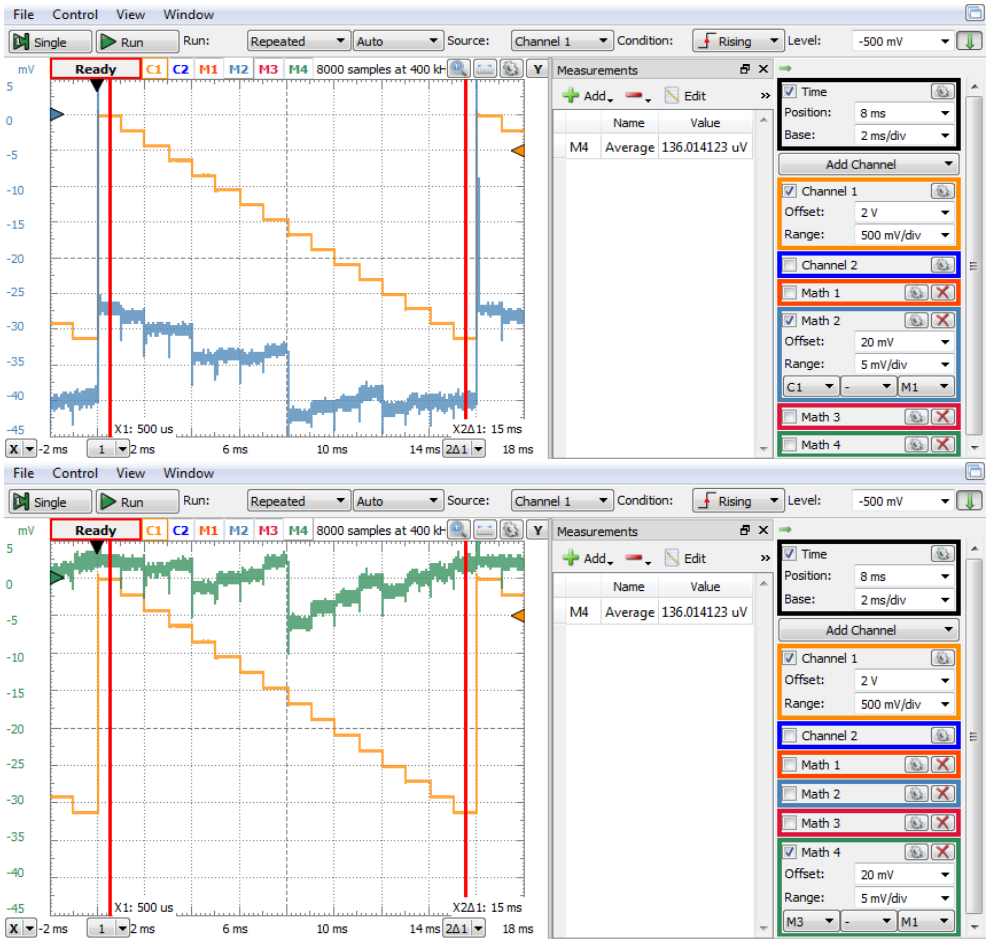


Figure 3.55 Operational amplifier output stage – total error (up) and linearity error (down).

Task 9. Use jumpers on J3, J4, J5, J7 headers to build a 12-bit combined hexadecimal resistor network. Set up a 12-bit binary counter in the *Patterns* Generator to drive the network above. Use the scope channel 1 to visualize the saw tooth signal *V15* on J8. Extend the time base and set the horizontal position to observe "big" glitches at ½ of the ramp, smaller glitches at ¼ and ¾ of the ramp. Notice that the Analog Discovery probes modify the wave

shapes: the probe parasitical capacity (24pF) builds a Low Pass Filter with the equivalent Resistor Network Resistance (9.375kΩ).

Task 10. Use jumpers on J3, J4, J5, J7 headers to build a 12-bit combined decimal resistor network. Set up a 3-digit (12 bit) BCD counter in the Patterns Generator to drive network above. Use the scope channel 1 to visualize the saw tooth signal *V15* on J8. Extend the time base and search for glitches. Explain the glitch location.

Task 11. Use a jumper on J9 to add the operational amplifier output stage. Use scope channel 1 to visualize V_{out} and channel 2 to see V_-. Repeat for both hexadecimal and decimal networks. Observe same glitches as above. Notice that the glitches are larger in amplitude; in fact, the operational amplifier stage makes the circuit less sensitive to the Analog Discovery scope probe: the operational amplifier output resistance is small, so the influence of the same parasitical capacitance of the probe is almost negligible. A scope probe on V_- affects the shape of V_{out} (remove the probe on V_- to see the difference).

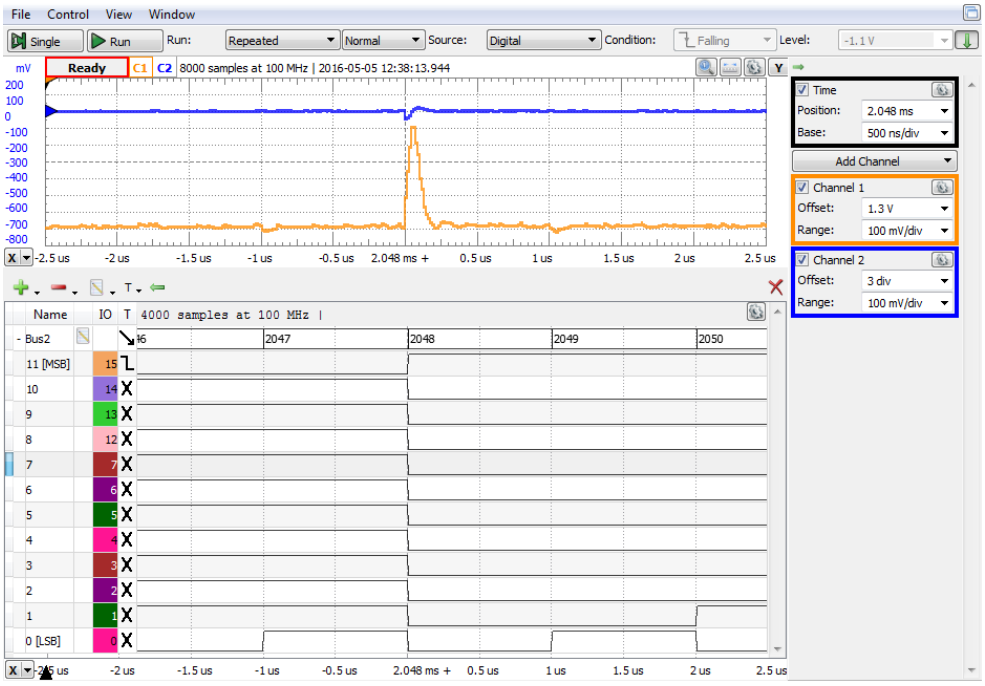


Figure 3.56 12-bit hexadecimal network with operational amplifier output stage detail: V_{out} and V_- glitch

Task 12. Use the decimal network with a hexadecimal counter and reverse. Explain the wave shapes.

Task 13. Use a 2-digit hexadecimal network with an 8-bit custom sinus Pattern Generator sequence.
Hint: use Xcell to build a .csv file with sinus samples:

- fill column A, rows 1 to 1024 with values 0, 1,…1023.
- write the equation =INT(127*SIN(2*PI()*A1/1024))+128 in cell B1. Copy cell B1 to cells B2 to B1024. That will generate 1024 sinus samples for one sinus period. The samples are shifted half of amplitude up, scaled to 0…256 range and rounded to integer
- save the file in csv format.

Hint: in the Pattern Generator, set an 8-bit custom bus with the sinus samples:

- in the pattern Generator, add the 8-bit custom bus DIO7…DIO0.
- in the Edit window, Import the sinus sample .csv file.
- In the import window, choose column 2 as source for Bus values.

Hint: set jumpers for DIO7…DIO0 two-digit hexadecimal combined resistor network. Repeat experiments for both configurations: with and without operational amplifier output stage.

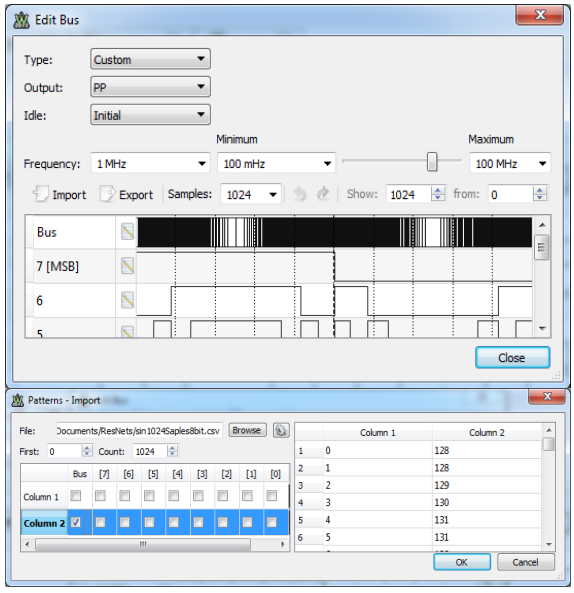


Figure 3.57 Edit Bus and Import windows

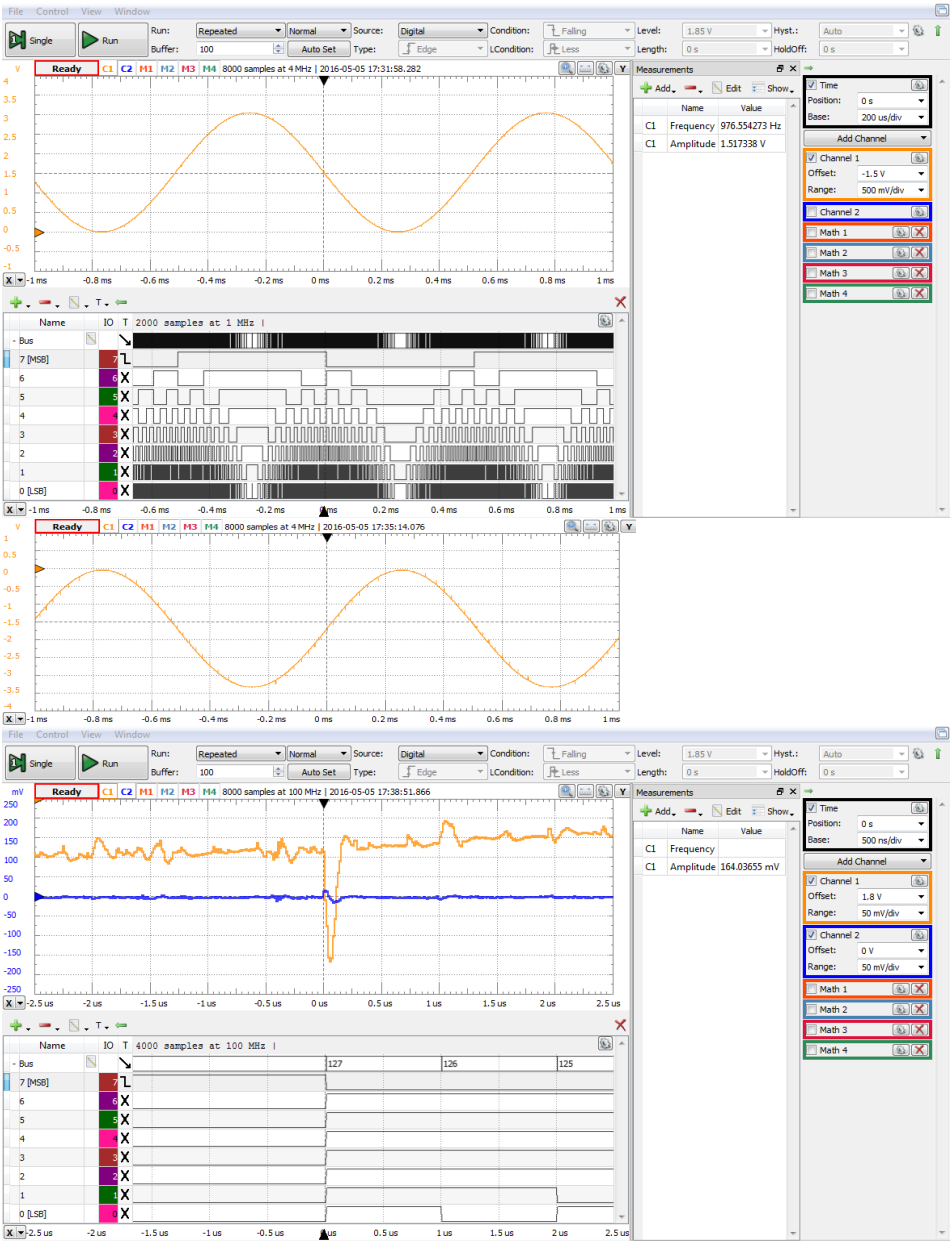


Figure 3.58 8-bit sinus by two-digit hexadecimal network without (up) or with (middle and down) operational amplifier output stage. Glitch detail (down)

Task 14. Measure dynamic parameters for the signal above.

Hint: open a Spectrum Analyzer instrument in *WaveForms*, and set:

- Click *View/Measure*;
- *Add/Trace1/Dynamic/ENOB*. This shows the equivalent number of bits, computed as:

$$ENOB = (SNR - 1.76)/6.02 \qquad \textbf{(3.54)}$$

- *Add/Trace1/Harmonics/FF*. That will show the amplitude and frequency of the fundamental component. Add some harmonics.
- In *Channel Options*, change *Sample Mode* between *Average* and *Decimate*. Notice the influence on *ENOB*. (Averaging the samples at acquisition reduces the noise and improve the *ENOB*. However, this improvement is done on the acquired image of the signal, not on the real signal).

Figure 3.59 Using the Spectrum Analyzer for Dynamic parameters measurement (8-bit sinus, decimate sampling mode)

Task 15. Repeat Task 13 and Task 14 with a 3-digit hexadecimal network and a 12-bit custom sinus Pattern Generator sequence.
Hint: in the Excell file, change the equation for cell B1 to =INT(2047*SIN(2*PI()*A1/1024))+2048 and copy cell B1 to cells B2…B1024. Save the file in csv format.
Hint: in the Patterns generator, add the 12-bit custom bus *DIO 15…DIO12, DIO7…DIO0*. In the Edit window, *Import* the sinus csv file.
Hint: set jumpers to configure the 3-digit hexadecimal combined network *R15…R12, R7…R0*. Repeat experiments for both configurations: with and without operational amplifier output stage.

4 DAC CMOS

4.1 Background

4.1.1 Unipolar operation

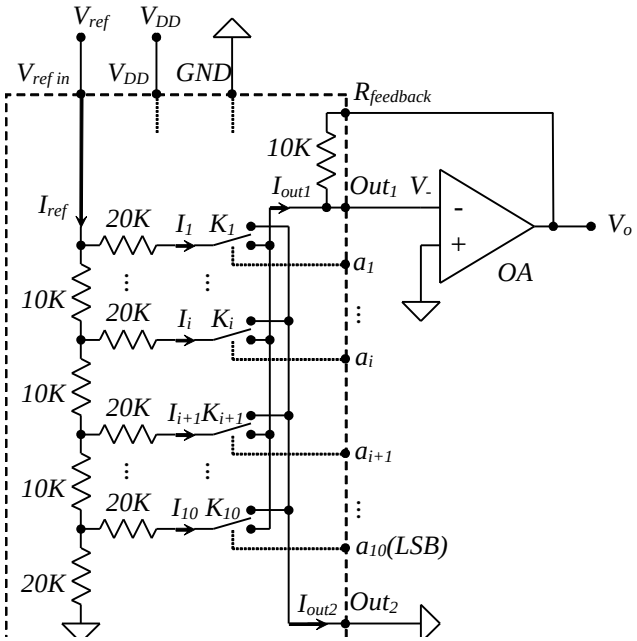


Figure 4.1 The simplified internal *AD 7520* structure (in dashed contour) and typical connection

Figure 4.1 shows a unipolar multiplying D/A circuit.

The dashed line includes the simplified internal structure of the10 bit, CMOS integrated AD7520 D/A converter. ICs with similar internal structure (with various resolutions and eventual additional features) are manufactured by Analog Devices and Texas Instruments.

The core of the circuit consists in an R-2R resistor network and the associated CMOS switches.

For correct behavior, the two current output pins Out_1 and Out_2 need to be driven at 0V by external circuitry. The typical application circuit connects Out_2 to GND and uses an operational amplifier with negative feedback for generating a virtual GND node in Out_1, as in Figure 4.1. The internal 10kΩ feedback resistor is matched with the R-2R network in regard to initial value, and drifts.

Each switch is driven by a bit of the input number: K_1 by the *MSB*…K_{10} by the *LSB.*

The unipolar binary number at the DAC input can be understood either as "fractional", composed of bits a_i, with $i=1\ldots n$, $a_1=MSB\ldots a_n=LSB$, *(n=10 for AD7520)*:

$$\{A\} = 0.a_1 \ldots a_{(n-1)}a_n = \sum_{i=1}^{n} a_i \cdot 2^{-i}; \quad \{A\} \in [0\ldots 1) \tag{4.1}$$

Or, as "integer", built of bits b_i, with $j=0\ldots n\text{-}1$, $b_{n-1}=MSB\ldots b_0=LSB$ *(n=10 for AD7520)*:

$$\{N\} = b_{n-1} \ldots b_1 b_0 = \sum_{j=0}^{n-1} b_j \cdot 2^{j}; \quad \{N\} \in [0\ldots 2^n) \tag{4.2}$$

where: $b_{n-i} = a_i \ \forall i=1\ldots n$. The two representations are tied by:

$$\{N\} = 2^n \cdot \{A\} \tag{4.3}$$

The equivalent resistance of pin V_{ref} to GND is *R;* the current in pin V_{ref} is:

$$I_{ref} = \frac{V_{ref}}{R} \tag{4.4}$$

The current halves in each node of the *R-2R* network; the current in branch *i* is:

$$I_i = \frac{I_{ref}}{2^i} \tag{4.5}$$

I_{out1} collects all the currents from branches with the corresponding bit $a_i=1$ and I_{out2} from branches with the corresponding bit $a_i=0$:

$$I_{out_1} = I_{ref} \cdot \sum_{i=1}^{8} a_i \cdot 2^{-i} = I_{ref} \cdot \{A\} \tag{4.6}$$

$$I_{out_2} = I_{ref} \cdot \sum_{i=1}^{N} \overline{a_i} \cdot 2^{-i} = I_{ref} \cdot \left\{\overline{A}\right\} \tag{4.7}$$

where$\{\overline{A}\}$is the complement of *{A}*:

$$\{\overline{A}\} = 0.\overline{a_1}\ldots\overline{a_{(n-1)}a_n} = \sum_{i=1}^{n}\overline{a_i}\cdot 2^{-i}; \quad \{\overline{A}\} \in [0\ldots1) \tag{4.8}$$

and:

$$\{A\} + \{\overline{A}\} + 1LSB = 1 \tag{4.9}$$

$$I_{out1} + I_{out2} + I_{LSB} = I_{ref}; \quad I_{LSB} = I_n = \frac{I_{ref}}{2^i} \tag{4.10}$$

I_{out1} flows through the feedback resistor and generates an output voltage of:

$$V_{out} = -I_{out1}\cdot R = -I_{ref}\cdot R\cdot\sum_{i=1}^{8} a_i\cdot 2^{-i} = -V_{ref}\cdot\{A\} = V_{FS}\cdot\{A\} \in [0\ldots V_{FS}) \tag{4.11}$$

In terms of integer representation:

$$V_{out} = -V_{ref}\cdot\frac{\{N\}}{2^n} = \frac{V_{FS}}{2^n}\cdot\{N\} = V_{LSB}\cdot\{N\} \in [0\ldots V_{FS}) \tag{4.12}$$

The CMOS switches in the DA7520 (and similar circuits) allow both polarities for I_{ref}, V_{ref}, I_1, I_2, V_{out}, etc. The circuit in Figure 4.1 can work as a two-quadrant multiplier.

4.1.2 Bipolar operation

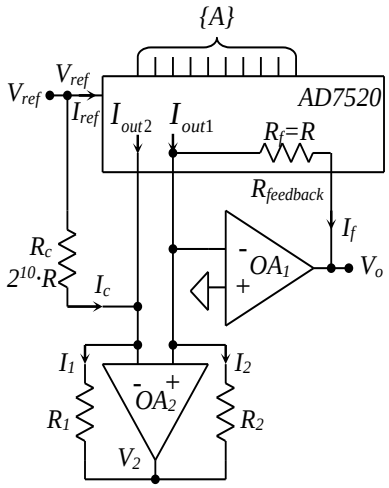


Figure 4.2 Bipolar DAC using AD7520

Figure 4.2 shows a bipolar *DAC* schematic. OA_2 has negative feedback; even if both feedback types coexist, the "positive" one is "reduced" by the OA_1 negative feedback, which "forces" the noninverting OA_2 input voltage to GND. As result, OA_2 input voltages are equal and null, due to virtual ground in "- "OA_1 input.

The circuit around OA_2, with $R_1 = R_2$, works as "current mirror":

$$I_1 = \frac{V_2}{R_1} = I_2 = \frac{V_2}{R_2} \tag{4.13}$$

Kirchhoff equations for OA_2 input nodes:

$$\begin{matrix} I_{out2} + I_c - I_1 = 0 \\ I_{out1} - I_f - I_2 = 0 \end{matrix} \quad \textbf{(4.14)}$$

And, also considering (4.10):

$$I_f = I_{out1} - I_{out2} - I_c; \quad I_c = \frac{V_{ref}}{2^n \cdot R} = I_{LSB} \quad \textbf{(4.15)}$$

$$I_f = I_{ref}(2 \cdot \{A\}_U - 1) \quad \textbf{(4.16)}$$

$$\begin{matrix} V_o = -I_f \cdot R_f = -I_{ref} \cdot R_f \cdot (2 \cdot \{A\}_U - 1) \\ = -V_{ref} \cdot (2 \cdot \{A\}_U - 1) \end{matrix} \quad \textbf{(4.17)}$$

Were $\{A\}_U$ is a unipolar, fractional number, as in (4.1). Notating $\{A'\}_{BO}$, the *Binary Offset* number:

$$\{A'\}_{BO} = 2 \cdot \{A\}_U - 1 \in -[1 \ldots 1) \quad \textbf{(4.18)}$$

Equation (4.17) becomes the definition of a *Binary Offset* DAC:

$$V_o = -V_{ref} \cdot \{A'\}_{BO} = V_{FS} \cdot \{A'\}_{BO} \in [-V_{FS} \ldots V_{FS}) \quad \textbf{(4.19)}$$

In terms of integer numbers, with $\{N'\}_{BO}$, the integer *Binary Offset* number (equivalent of moving the fractional dot over *n-1* bits to the right):

$$\{N'\}_{BO} = \{A'\}_{BO} \cdot 2^{n-1} \in [-2^{n-1} \ldots 2^{n-1}) \quad \textbf{(4.20)}$$

Equation (4.17) becomes the definition of a *Binary Offset* DAC:

$$V_o = -V_{ref} \cdot \frac{\{N'\}_{BO}}{2^{n-1}} = V_{LSB} \cdot \{N'\}_{BO} \in [-V_{FS} \ldots V_{FS}) \quad \textbf{(4.21)}$$

The circuit is a bipolar DAC, meaning the input number is bipolar, in *Binary Offset* code. The most significant bit is now the Sign Bit (0 for *"negative"*, 1 for *"positive"*).
Complementing the MSB would turn the circuit into a *2's Complement* DAC. With both possible polarities for I_{ref}, V_{ref}, I_1, I_2, V_{out}, and also for the input number, the circuit in Figure 4.2 can work as a four-quadrant multiplier.

4.2 Experiment and measurements

The AD7524 CMOS DAC is used on the experimental board. It has a similar structure with AD7520, except:

- It has only 8-bit of resolution.
- Data Latches are provided for compatibility to a microprocessor data bus. If not used, *ChipSelect* and *Write* pins must be tied to GND.

4.2.1 Unipolar Operation

Figure 4.3 shows the full schematic of the experimental board, but circuits not used in the current experiment are shaded. Notice that *J3* is loaded with a jumper on pins *2-3*, to tie Out_2 of *U2* to *GND*, and *J4* is loaded with a jumper on pins *1-2*, to insulate *U1A*.

U2 and U1B build a circuit equivalent to Figure 4.1, with:

- V_{ref} = *W1*, the AWG channel 1 of the Analog Discovery.
- *Data bits = DIO7 (MSB)…DIO0(LSB)* of Analog Discovery.

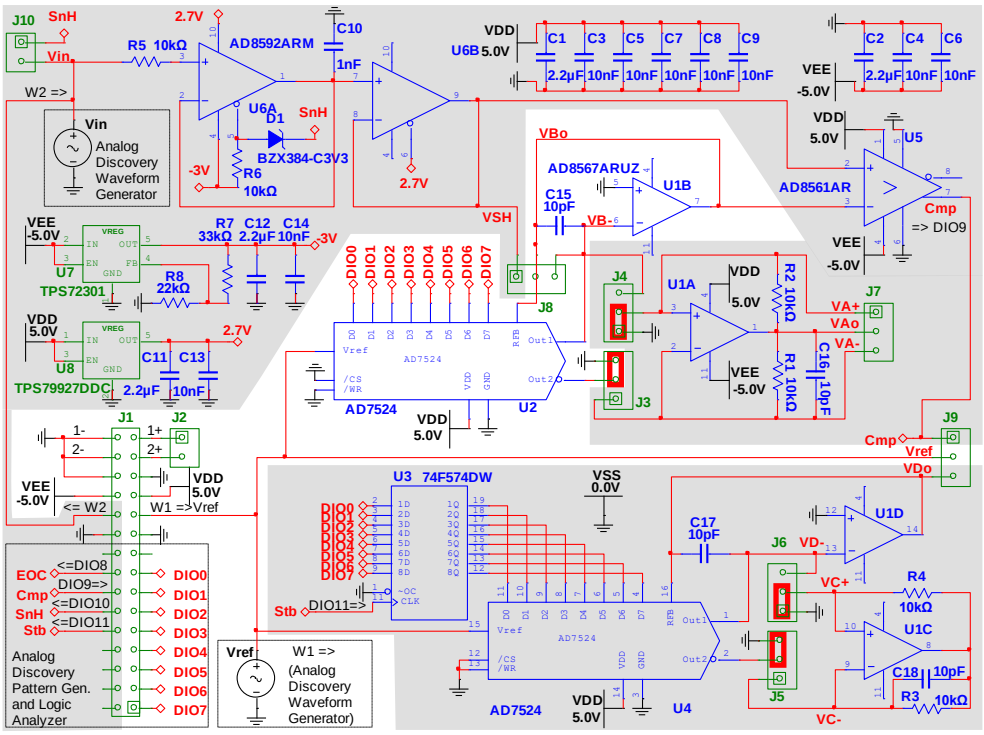


Figure 4.3 DAC CMOS, unipolar operation – experimental board schematic

Notice C_{15}, C_{16}, C_{17} and C_{18}, added to compensate U1B, U1A, U1D and U1C respectively. They reduce ringing and oscillation risk for the high speed AD8567 operational amplifiers.

Figure 4.4 shows the experimental board prepared with jumpers and scope probe wires for the Static DAC experiment. Notice that the same board is used for several other experiments.

4.2.1.1 *Static DAC experiment*

On the experimental board:

- Place jumpers at:
  - *J3 = 2-3* – shorts I_{out2} to GND.
  - *J4 = 1-2* – disconnects U1A.
- Place scope probe wires:
  - Channel 1 to *J9-2 = W1 =* V_{ref}.
  - Channel 2 to *J8-2 = VBo =* V_{out}.

In the WaveForms software:

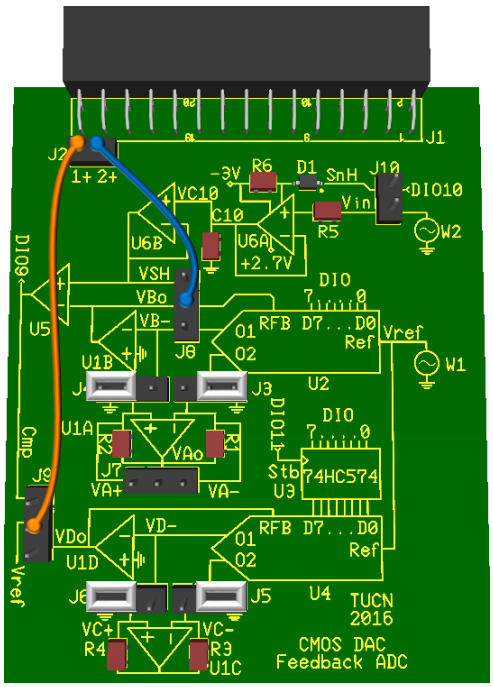


Figure 4.4 DAC CMOS experimental board

- In *Wavegen*, W1, set a constant, positive or negative voltage for V_{ref}.
- In *StaticIO*, set a slider for bits *DIO7…DIO0*.
- In *Supplies*, activate both +/-V user voltage supplies. (If the WaveForms software reports an "Overcurrent condition" and stops the user power supplies, try starting the supplies one by one: first the positive supply and, after few seconds, the negative one)
- In *Scope*, open *Measurements* for channel 1 *Average* and channel 2 *Average*. Set a low sampling rate (i.e., 100ms/div). This results in averaging more samples for stable measured values.
- Modify the input number in the *StaticIO* and read the output voltage in the scope measurements pane.

Task 1. Starting from the measured voltages, write down the equations and calculate the **actual values** of ε_{offset}, ε_{gain}, V'_{FS} and V'_{LSB}.

Hint: The ideal DAC equation is:

$$V_o = V_{FS} \cdot \{A\} \tag{4.22}$$

Considering the offset and gain errors, the DAC equation becomes:

$$V_o = \varepsilon_{off,abs} + V'_{FS} \cdot \{A\} \tag{4.23}$$

$$\begin{aligned} V_o &= \varepsilon_{off,abs} + (V_{FS} + \varepsilon_{gain,abs}) \cdot \{A\} \\ &= \varepsilon_{off,abs} + (1 + \varepsilon_{gain,rel}) \cdot V_{FS} \cdot \{A\} \end{aligned} \tag{4.24}$$

4.2.1.2 *Dynamic DAC experiment*

On the experimental board:

- Place jumpers at:
  - *J3 = 2-3* – shorts I_{out2} to GND.
  - *J4 = 1-2* – disconnects U1A.
- Place scope probe wires:
  - Channel 1 to *J8-3 = VB- = U1B* inverting input (virtual *GND*).
  - Channel 2 to *J8-2 = VBo =* V_{out}.

In the WaveForms software:

- In *Wavegen*, W1, set a constant, positive voltage for V_{ref}.
- Make sure *StaticIO* is closed or set a LEDs for bits *DIO7…DIO0*.
- In *Patterns*, set a binary counter on DIO7…DIO0. Set the clock frequency to 1MHz.
- In *Supplies*, activate both +/-V user voltage supplies. (If the WaveForms software reports an "Overcurrent condition" and stops the user power supplies, try starting the supplies one by one: first the positive supply and, after few seconds, the negative one).
- In Scope, *Add/Digital/Bus DIO7…DIO0*. Set *Trigger Source* to *Digital* and set Trigger condition to DIO7 *Falling Edge*.
- *In scope, Add/Math/Simple: C2-C1, Math1* shows the voltage drop over R_{FB}.
- In *Scope*, set appropriate time base to see several saw-tooth periods of V_{out}.

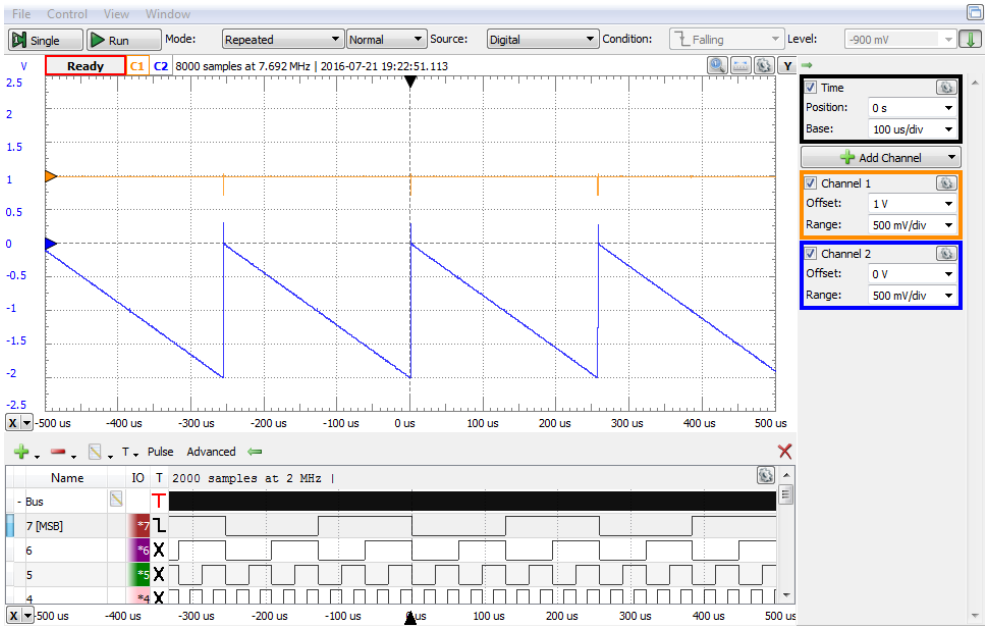


Figure 4.5 Dynamic DAC experiment (unipolar operation) – scope view

- Play with the values of V_{ref} (in *Wavegen1*), clock *Frequency* (in *Patterns*), *Time/Base* (in *Scope*). Observe the effects and understand the mechanisms.
- Identify the non-idealities and errors in the scope image: settling time, overshoot, glitches. Prepare to change the time base and channel range for more detailed observation and measurements.

Settling time, t_{set}, is the time from changing the digital input value to a stable value of the output voltage, within the acceptable error band (accuracy).

Task 2. Measure the **actual value** of t_{set}, at ±10mV accuracy.

Hint: The worst case is at the maximal voltage trip; for a saw tooth signal, from V_{FS} to 0, when the input number rolls over. This is easy to synchronize with the falling edge of the MSB in the digital bus on the scope. Change the time base and position, respectively the range and offset to optimize the view, as in Figure 4.6. Eventually use zoom windows in the scope (*View/Add Zoom* and use right and left drag over the Zoom window axes to change the zoom factor and position). Change the frequency in the *Pattern generator* (if

needed), to make the period longer than the settling time. Notice the ringing due to the large voltage trip and understand that, for this particular circuit, the big component of the t_{set} is given by the incompletely compensated operational amplifier. Notice that the scope wire probes influence the signal shapes (independent/ribbon/twisted wires).

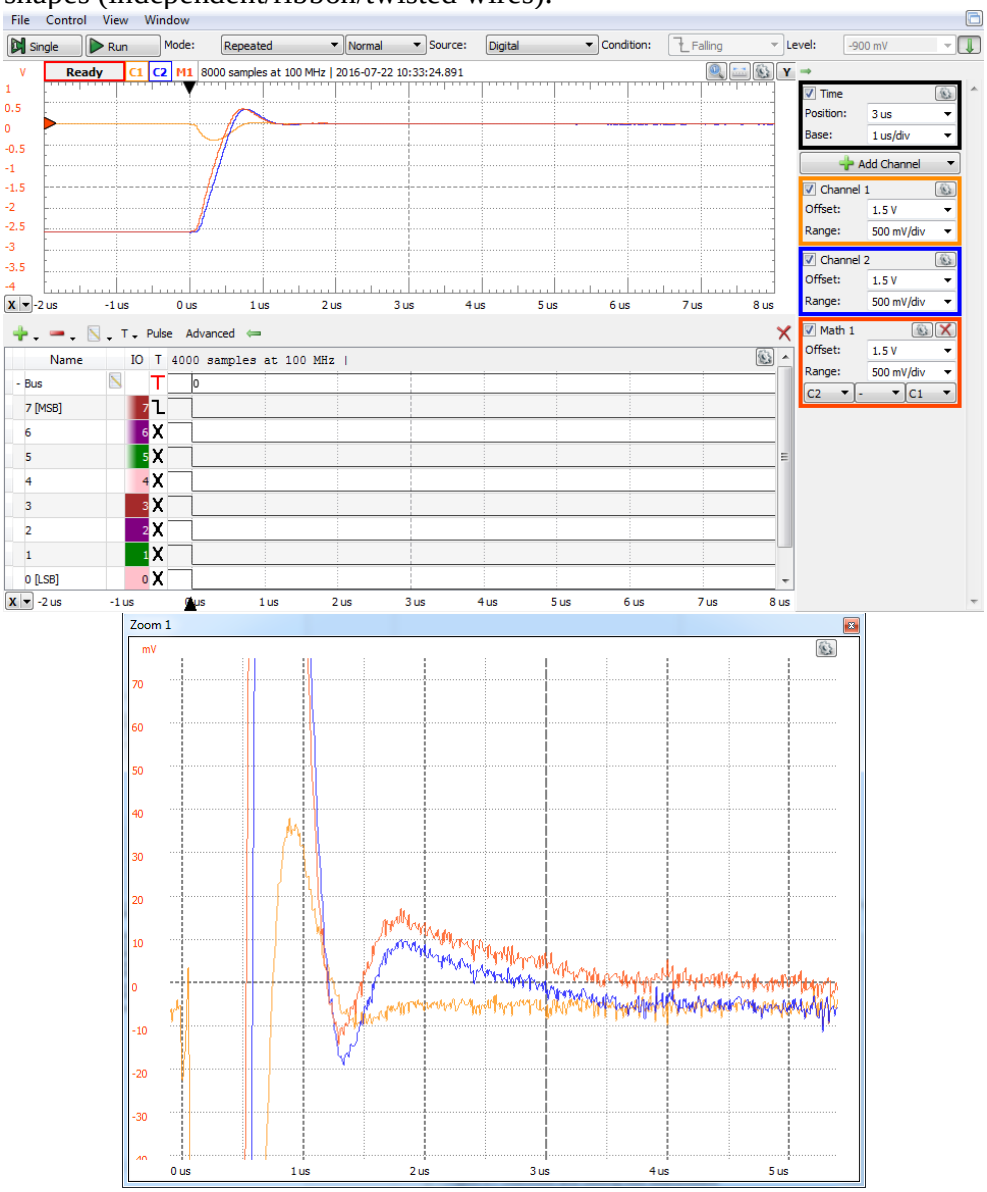


Figure 4.6 Overshoot and settling time measurement

Ringing is the damped oscillation following a step of the output voltage. The amplitude **(overshoot)** and the frequency of the ringing depends on the circuit frequency characteristics. Ringing denotes insufficient phase compensation, while overcompensation would lead to an exponential time characteristic, as in Figure 4.7, where a 39pF capacitor was placed parallel to C_{15}.

Overshoot, V_{set}, is the time period from changing the digital input value to a stable value of the output voltage, within the acceptable error band.

Task 3. Measure the **actual value** of the **overshoot** and **ringing frequency**.

Notice the glitch on V_{B-} (channel 1 of the scope). Theoretically, this should be virtual *GND*, due to the negative feedback of U1B. The negative feedback should force V_{Bo} to such a voltage that makes V_{B-} null:

$$V_{B-} = V_{Bo} - I_{FB} \cdot R_{FB} = 0V \qquad \textbf{(4.25)}$$

However, when V_{Bo} cannot have the required value, V_{B-} is not zero; in this case, V_{Bo} would be required to change faster than the operational amplifier *Slew Rate*, which is impossible, generating a pulse on V_{B-}.

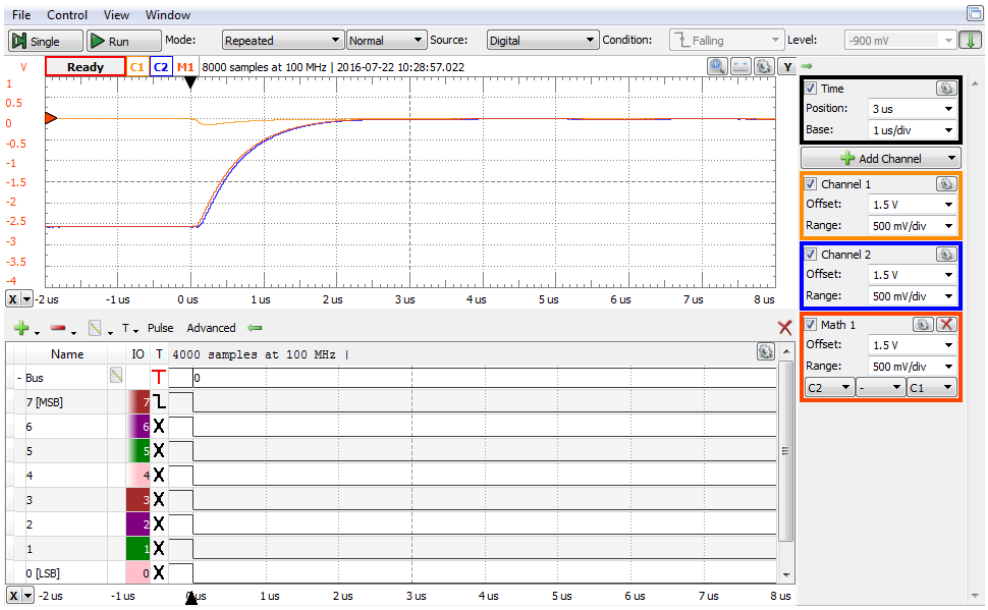


Figure 4.7 Overcompensated output stage behavior

Glitch is the V_{out} pulse due to mismatched switching time of multiple bits changing at the same time. The glitch is measured either as amplitude or as "energy"- the area closed by the glitch pulse in $\mu V*sec$.

The worst situation is when all bits switch at once, from 0111… to 1000… (for example at midscale of a binary counter), or reverse. This is easy to synchronize on the rising edge of the MSB, in the digital bus on the scope. Change the time base and position, respectively the range and offset to optimize the view, as in Figure 4.8. Notice V_{Bo} should theoretically change from $127V_{LSB}$ to $128V_{LSB}$, with a delta of V_{LSB}. The glitch amplitude is much bigger. Notice again that the V_{Bo} variation should theoretically be faster than the operational amplifier Slew Rate, so the V_{Bo} speed is limited to Slew Rate and the difference is taken by V_{B-}.

Task 4. Measure the **actual value** of the **glitch amplitude** and visually approximate the **glitch energy**.

Hint: for glitch energy, approximate the area underneath the glitch pulse to a triangle. Change the time base and position, respectively voltage range and offset as convenient.

Task 5. Identify the glitch(es) with the second amplitude. Explain.

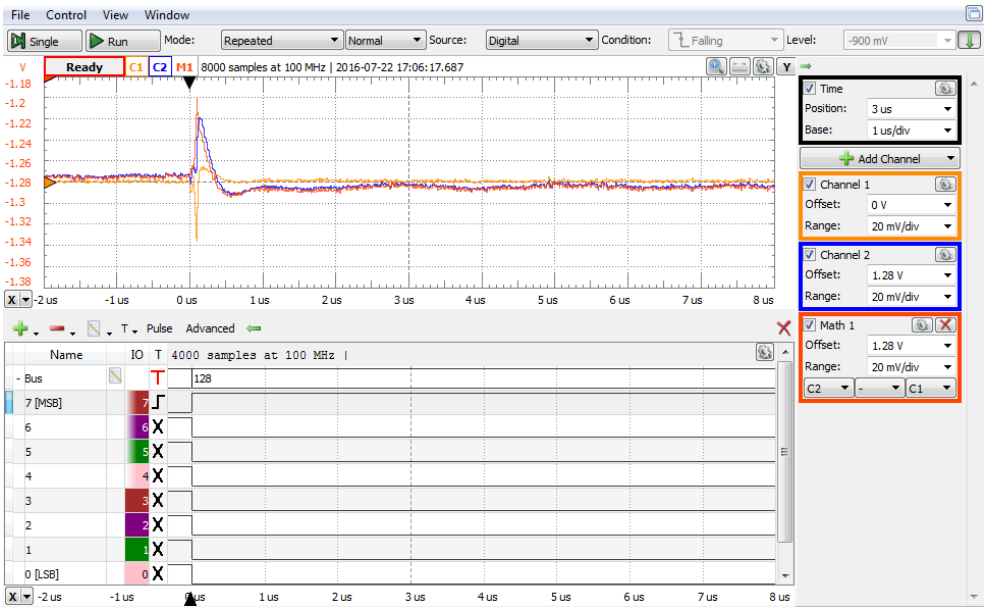


Figure 4.8 Glitch

4.2.1.3 *Multiplying DAC experiment*

On the experimental board:

- Place jumpers at:
  - *J3 = 2-3* – shorts I_{out2} to *GND*.
  - *J4 = 1-2* – disconnects U1A.
- Place scope probe wires:
  - Channel 1 to *J9-2* = V_{ref} = *W1*.
  - Channel 2 to *J8-2* = *VBo* = V_{out}

In the WaveForms software:

- In *Wavegen*, W1, set a 3kHz sinus voltage for V_{ref}, with null offset and 2V amplitude.
- Make sure *StaticIO* is closed or set a LEDs for bits *DIO7…DIO0*.
- In *Patterns*, set a binary counter on *DIO7…DIO0*. Set the clock frequency to 100kHz. This will generate a saw-tooth digital signal with the frequency 100kHz/2^8.
- In *Supplies*, activate both +/-V user voltage supplies.
- In *Scope*, *Add/Digital/Bus DIO7…DIO0*. Set *Trigger Source* to *Digital* and set Trigger condition to *DIO7 Falling Edge*. Expand the Bus line in the digital view of the scope, to see the graphical

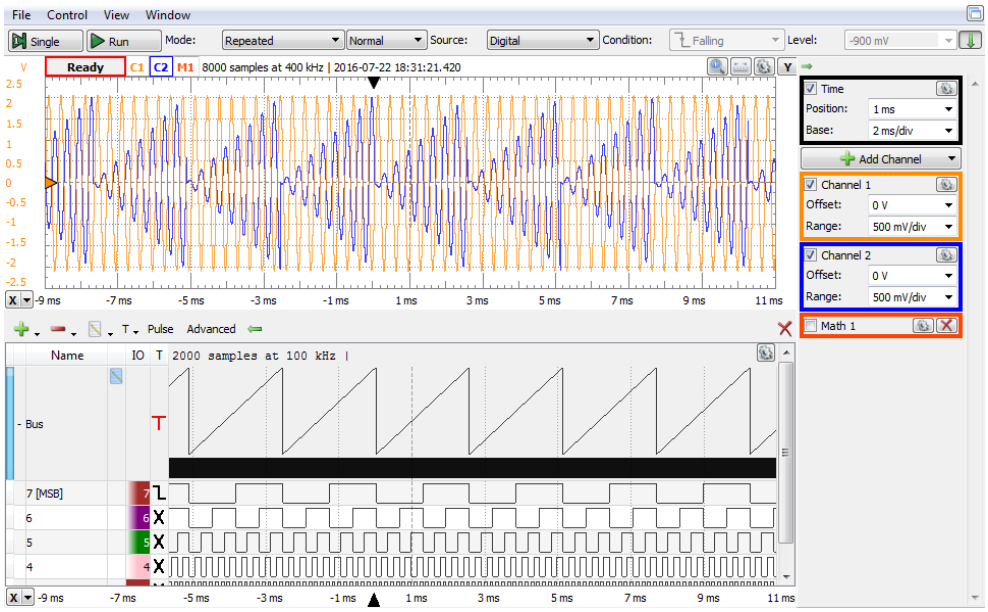


Figure 4.9 Multiplying unipolar DAC (two quadrant multiplier)

representation of the bus value, as in Figure 4.12. Notice that the graphical representation depends on the *Format* chosen for the bus (*Binary/Vector/Signed/Ones Complement/2sComplement…*). Choose any *Unipolar* representation since the DAC is unipolar.

- In *Scope*, set appropriate time base and voltage Range.

Notice that the output voltage is the product between the bipolar V_{ref} and the unipolar input number. The circuit works with both polarities of V_{ref} since the CMOS switches allow current flow in both directions.

Task 6. Produce a product of two sinus signals: 3kHz $\boldsymbol{V_{ref}}$ and 100Hz input number.

Hint: generate a 100Hz sinus with custom samples in the *Pattern Generator*:

- Prepare a .csv file with 1000 lines. In column A, place integer values from to 0…999. In the cell B1, write the following formula: =128+INT(127*SIN(2*PI()*A1/1000)). Copy cell B1 to cells B2…B1000. This calculates 1000 integer 8-bit samples of a sinus period, with 2*π/1000 phase resolution. Save the .csv file. When saving the file as .csv, the formulas are replaced by numerical values.
- In *Patterns*, change the bus type to *Custom* and edit it. Click Import and select the .csv file prepared above. Select column 2 to be loaded for the whole bus, as in Figure 4.10. This way, the decimal values in column B of the .csv file are converted to binary, and each bit is loaded into the appropriate signal of the bus.
- Make sure, there are exactly 1000 samples in the buffer.

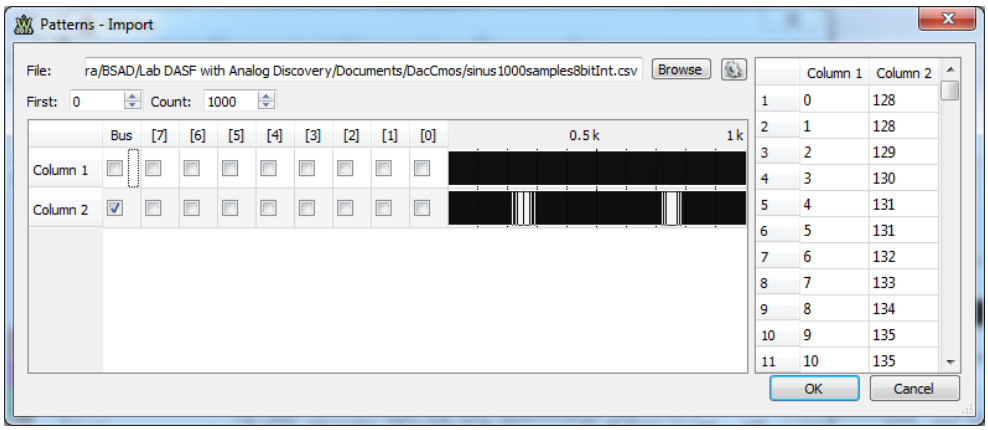


Figure 4.10 Importing a .csv file to *Patterns* generator

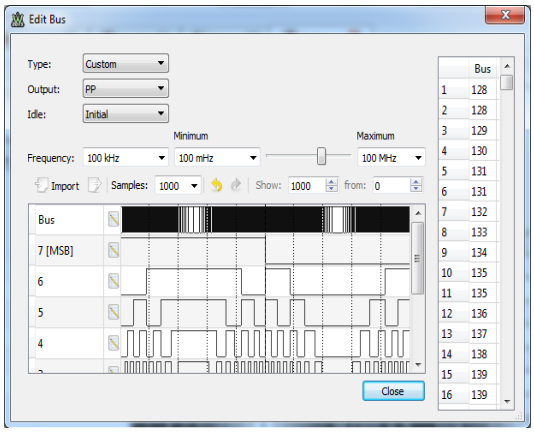


Figure 4.11 Edit a bus in *Patterns*

- Set the frequency to 100kHz. The sinus frequency will be 100kHz/1000samples = 100Hz.
- Run all instruments and observe the scope image. Notice that the output voltage has opposite polarity to V_{ref}, due to the inverting output stage.
- Slightly change the V_{ref} frequency (ex. to 3.003kHz). Notice the effect and explain.

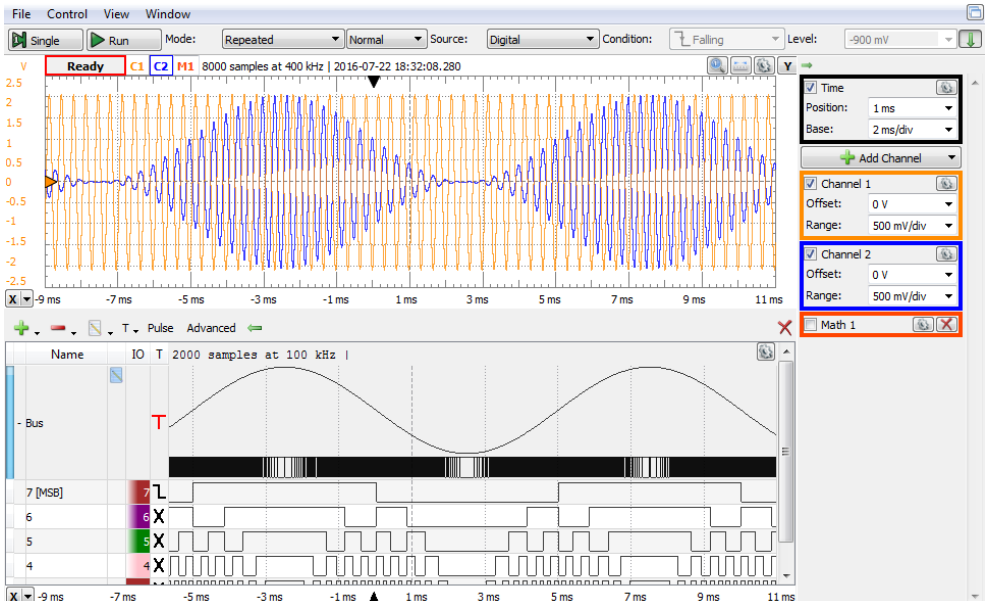


Figure 4.12 Multiplying unipolar DAC (two quadrant multiplier) - sinus

4.2.2 Bipolar Operation

In Figure 4.13, J3 and J4 jumpers were moved to connect U1A in the circuit. The schematic is similar to the one paragraph 4.1.2. The circuit is a *7+1-bit*, *bipolar*, *Binary Offset DAC* and can work as a *four quadrant multiplier*, with:

- V_{ref} = *W1*, the AWG channel 1 of the Analog Discovery.
- *Data bits = DIO7 (MSB=Sign Bit)…DIO0(LSB)* of Analog Discovery

4.2.2.1 *Static DAC experiment*

On the experimental board:

- Place jumpers at:
  - *J3 = 1-2* – connects I_{out2} to U1A current mirror.
  - *J4 = 2-3* – connects U1A in the circuit.
- Place scope probe wires:
  - Channel 1 to *J9-2 = W1 =* V_{ref}.
  - Channel 2 to *J8-2 = VBo =* V_{out}.

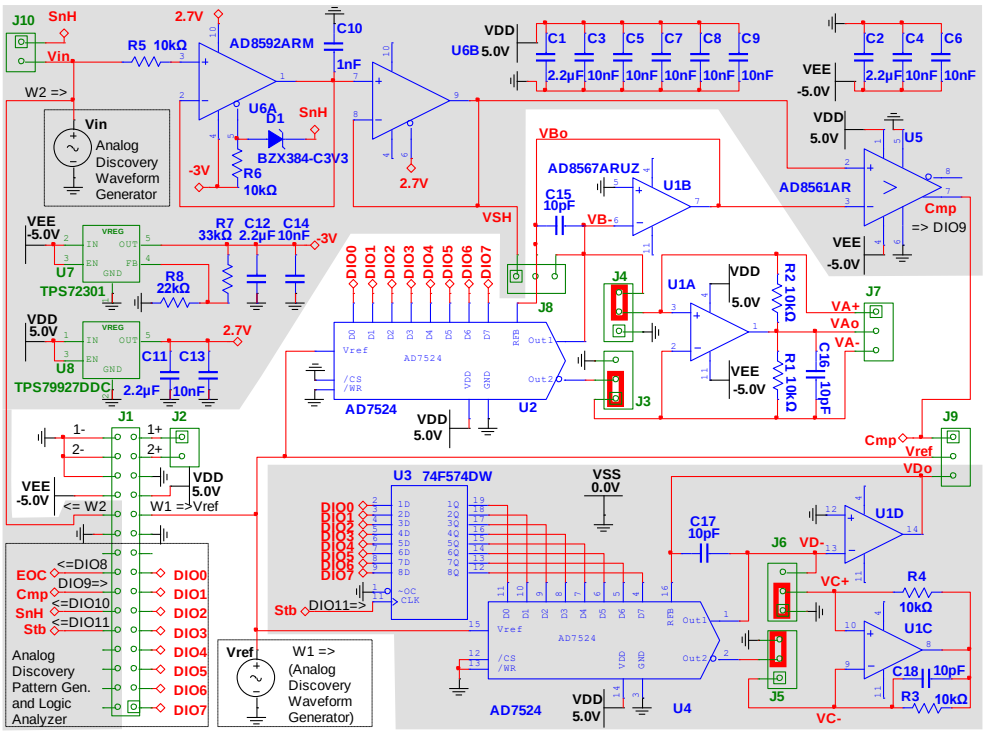


Figure 4.13 DAC CMOS, unipolar operation – experimental board schematic

In the WaveForms software:

- In *Wavegen*, W1, set a constant, positive or negative voltage for V_{ref}.
- In *StaticIO*, set a *slider* for bits *DIO7…DIO0*. The static IO instrument in WaveForms encodes the *8 bits* above as an *unsigned integer*. However, in this configuration, the experimental board encodes the same bits as a *7+1 bipolar, binary offset integer*. For the *Binary Offset* value, always subtract 128 from the number displayed on the right side of the slider.
- In *Supplies*, activate both +/-V user voltage supplies. (If the WaveForms software reports an "Overcurrent condition" and stops the user power supplies, try starting the supplies one by one: first the positive supply and, after few seconds, the negative one).
- In *Scope*, open *Measurements* for channel 1 *Average* and channel 2 *Average*. Set a low sampling rate (i.e. 100ms/div). This results in averaging more samples for stable measured values.
- Modify the input number in the *StaticIO* and read the output voltage in the scope measurements pane. Notice that the output voltage polarity changes when the input number polarity changes.

4.2.2.2 *Dynamic DAC experiment*

On the experimental board:

- Keep jumpers at:
  - *J3 = 1-2* – connects I_{out2} to U1A current mirror.
  - *J4 = 2-3* – connects U1A in the circuit.
- Place scope probe wires:
  - Channel 1 to *J8-3 = VB- = U1B* inverting input (virtual *GND*).
  - Channel 2 to *J8-2 = VBo =* V_{out}.

In the WaveForms software:

- In *Wavegen*, W1, set a constant, positive or negative voltage for V_{ref}.
- Make sure *StaticIO* is closed or set a LEDs for bits *DIO7…DIO0*.
- In *Patterns*, set a binary counter on DIO7…DIO0. Set the clock frequency to 1MHz.
- In *Supplies*, activate both +/-V user voltage supplies. (If the WaveForms software reports an "Overcurrent condition" and stops the user power supplies, try starting the supplies one by one: first the positive supply and, after few seconds, the negative one).

- In Scope, *Add/Digital/Bus DIO7…DIO0*. Set *Trigger Source* to *Digital* and set Trigger condition to DIO7 *Falling Edge*.
- *In scope, Add/Math/Simple: C2-C1, Math1* shows the voltage drop over R_{FB}.
- In *Scope*, set appropriate time base to see several saw-tooth periods of V_{out}.
- Play with the values of V_{ref} (in *Wavegen1*), clock *Frequency* (in *Patterns*), *Time/Base* (in *Scope*). Observe the effects and understand the mechanisms.
- Notice that the V_{out} amplitude is double compared to the unipolar operation, for the same value of V_{ref}.
- Identify the non-idealities and errors in the scope image: settling time, overshoot, glitches. Change the time base and channel range for more detailed observation and measurements.

4.2.2.3 *Multiplying DAC experiment*

On the experimental board:

- Keep jumpers at:
  - *J3 = 1-2* – connects I_{out2} to U1A current mirror.
  - *J4 = 2-3* – connects U1A in the circuit.
- Place scope probe wires:
  - Channel 1 to *J9-2* = V_{ref} = *W1*.
  - Channel 2 to *J8-2 = VBo* = V_{out}.

In the WaveForms software:

- In *Wavegen*, W1, set a 3kHz sinus voltage for V_{ref}, with null offset and 2V amplitude.
- Make sure *StaticIO* is closed or set a LEDs for bits *DIO7…DIO0*.
- In *Patterns*, set a binary counter on *DIO7…DIO0*. Set the clock frequency to 100kHz. This will generate a saw-tooth digital signal with the frequency 100kHz/2^8.
- In *Supplies*, activate both +/-V user voltage supplies.
- In Scope, *Add/Digital/Bus DIO7…DIO0*. Set *Trigger Source* to *Digital* and set Trigger condition to DIO7 *Falling Edge*. Expand the Bus line in the digital view of the scope, to see the graphical representation of the bus value, as in Figure 4.12. Notice that the graphical representation depends on the *Format* chosen for the bus (*Binary/Vector/Signed/Ones Complement/2sComplement…*). Choose

any unipolar representation since *Binary Offset* uses the same code sequence.

- In *Scope*, set appropriate time base and voltage range.

Notice that the output voltage is the product between the bipolar V_{ref} and the bipolar input number. The circuit works with both polarities of V_{ref} since the CMOS switches allow current flow in both directions. Since the input number is also bipolar, the circuit works in all 4 quadrants.

Task 7. Produce a product of two sinus signals: 3kHz $\boldsymbol{V_{ref}}$ and 100Hz input number.

Hint: follow the same exact steps as for the unipolar experiment. Since *Binary Offset* use the same code sequence as unipolar, the same sample string will work in this configuration as a correct *Binary Offset* one.

- Run all instruments and observe the scope image. Output voltage polarity depends both on the input number and V_{ref} polarities.
- Slightly change the V_{ref} frequency (ex. to 3.003kHz). Notice the effect and explain.

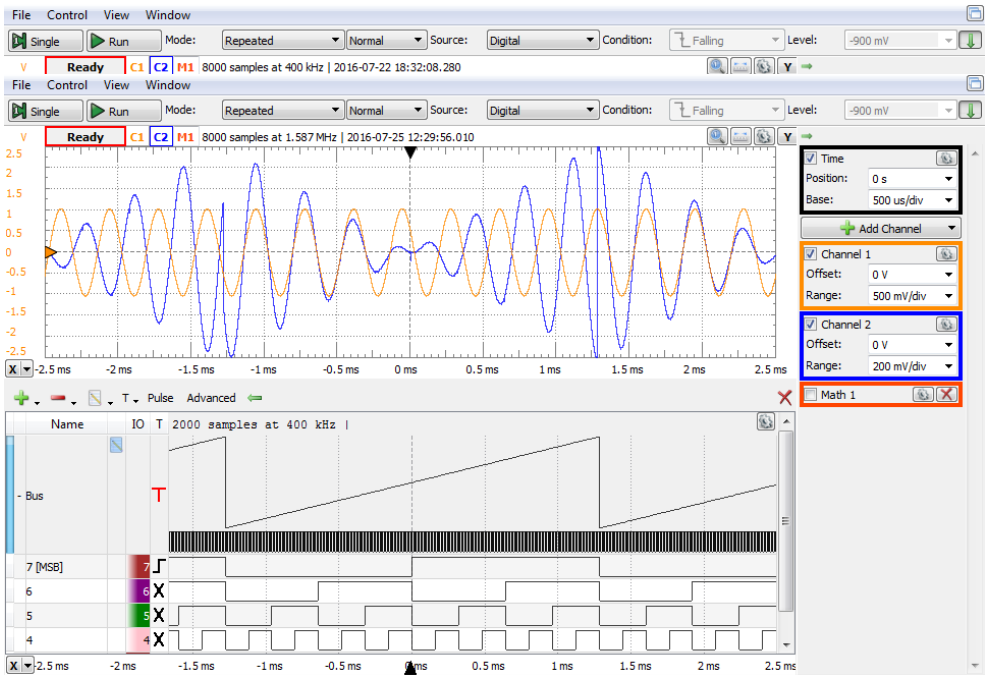


Figure 4.14 Multiplying Bipolar DAC (four quadrant multiplier)

5 Digitally controlled filter

5.1 Background

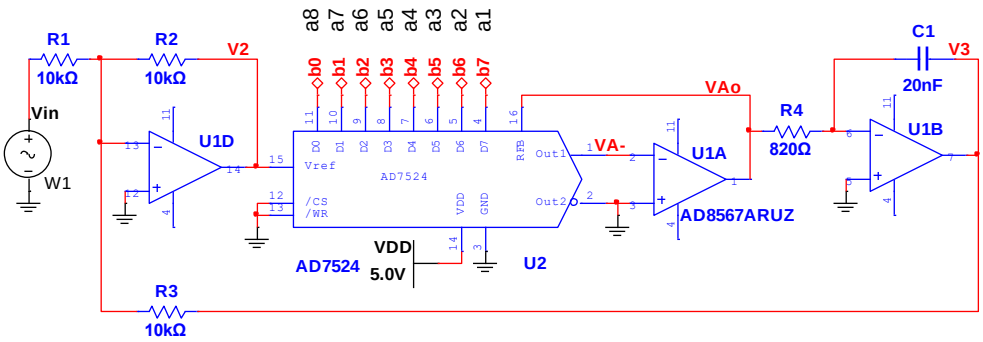


Figure 5.1 Digitally controlled filter schematics

In Figure 5.1, *U1D* is an inverting adder, *U2* and *U1A* build a typical inverting, voltage output, multiplying DAC stage, *U1B* is an inverting integrator.

The unipolar binary number at the DAC input can be understood either as "fractional", built of bits a_i, with $i=1...n$, $n=8$, $a_1=MSB...a_8=LSB$:

$$\{A\} = 0.a_1 \ldots a_{(n-1)}a_n = \sum_{i=1}^{n} a_i \cdot 2^{-i} \tag{5.1}$$

Or, as "integer", built of bits b_i, with $i=0...n-1$, $n=8$, $b_7=MSB...b_0=LSB$:

$$\{N\} = b_{n-1} \ldots b_1 b_0 = \sum_{i=0}^{n-1} b_i \cdot 2^{i} \tag{5.2}$$

Where: $b_{n-i} = a_i \ \forall i=1...n$. The two representations are tied by:

$$\{N\} = 2^n \cdot \{A\} \tag{5.3}$$

The Laplace complex equations are:

$$V_2(s) = -V_{in}(s) - V_3(s) \tag{5.4}$$

$$V_{Ao}(s) = -\{A\} \cdot V_2(s) = -\frac{\{N\}}{2^n} \cdot V_2(s) \tag{5.5}$$

$$V_3(s) = -\frac{V_{Ao}(s)}{s \cdot R_4 \cdot C_1} \quad \textbf{(5.6)}$$

The Laplace transfer functions:

$$\frac{V_3(s)}{V_{in}(s)} = -\frac{1}{1 + s \cdot \frac{R_4 \cdot C_1}{\{A\}}} \quad \textbf{(5.7)}$$

$$\frac{V_1(s)}{V_{in}(s)} = -\frac{s \cdot \frac{R_4 \cdot C_1}{\{A\}}}{1 + s \cdot \frac{R_4 \cdot C_1}{\{A\}}} \quad \textbf{(5.8)}$$

Noting f_o a circuit constant and f_c depending on $\{A\}$ and reducing the Laplace variable s to the Fourier variable, $j\omega$ (for periodical signals only):

$$f_0 = \frac{\omega_0}{2 \cdot \pi} = \frac{1}{2 \cdot \pi \cdot R_4 \cdot C_1} = 9.704kHz \quad \textbf{(5.9)}$$

$$f_c = \frac{\omega_c}{2 \cdot \pi} = \frac{\omega_0 \cdot \{A\}}{2 \cdot \pi} = \frac{\{A\}}{2 \cdot \pi \cdot R_4 \cdot C_1} = f_0 \cdot \{A\} \quad \textbf{(5.10)}$$

$$s = j \cdot \omega = 2 \cdot \pi \cdot f \cdot j \quad \textbf{(5.11)}$$

The transfer functions become:

$$\frac{V_3(s)}{V_{in}(s)} = -\frac{1}{1 + j \cdot \frac{f}{f_0 \cdot \{A\}}} = -\frac{1}{1 + j \cdot \frac{f}{f_c}} \quad \textbf{(5.12)}$$

$$\frac{V_1(s)}{V_{in}(s)} = -\frac{j \cdot \frac{f}{f_0 \cdot \{A\}}}{1 + j \cdot \frac{f}{f_0 \cdot \{A\}}} = -\frac{j \cdot \frac{f}{f_c}}{1 + j \cdot \frac{f}{f_c}} \quad \textbf{(5.13)}$$

Equation (5.12) describes a Low Pass Filter while (5.13) shows a High Pass Filter, both with the corner frequency of f_c.

Indeed, for V_3, the integrator is in the direct path, setting the LPF character of the transfer function. For V_1 instead, the integrator is in the feedback path, which complements the behavior of the transfer function to a HPF.

5.2 Simulation

The AD7524 spice model is not available. For simulation, the DAC is modeled in Figure 5.2 with R_{15}, and R_{16}. R_{16} is the direct image of the feedback resistor inside of AD7524. R_{15} is the equivalent resistance between V_{ref} and Out_1 pins of AD7524. Its value depends on the input number $\{A\}$:

$$R_{16} = 10k\Omega \tag{5.14}$$

$$R_{15} = \frac{10k\Omega}{\{A\}} \tag{5.15}$$

When $\{A\}=0$, the gain of U1A stage is null. With $\{A\}=1$ (Full Scale, impossible), the gain of U1A stage would be unity.

Figure 5.3 shows the AC simulation results with Parameter Sweep: R_{15} sweeps in octave mode, from 20kΩ to 2.56MΩ, with 3 points/octave. The 8 swept values correspond to a walking 1 among $\{A\}$ bits, from MSB to LSB:

$$R_{15,i} = 10k\Omega \cdot 2^i \Leftrightarrow \{A\} = 2^{-i}; i = 1\ldots8 \tag{5.16}$$

The magnitude characteristics family include 8 pairs of LPF/HPF. Each pair has a unique corner frequency. The lowest corner frequency (at cursor 1) corresponds to $\{A\}_{min}=2^{-8}$ (LSB=1, all others = 0), $R_{15,max}=10k\Omega*2^8=2.56M\Omega$ and the highest corner frequency (at cursor 2) to $\{A\}_{max}=2^{-1}$ (MSB=1, all others = 0) $R_{15,min}=10k\Omega*2^1=20k\Omega$.

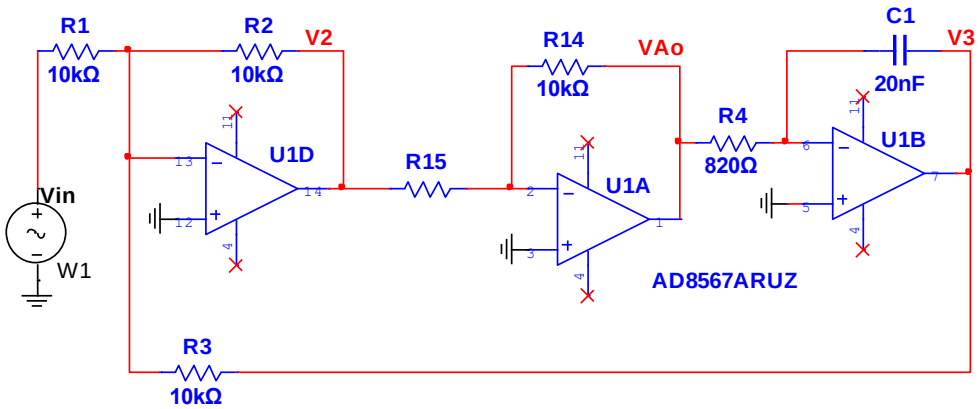


Figure 5.2 Digitally controlled filter simulation schematic

$$f_{c,min} = f_0 \cdot 2^{-8} = 9.704 \cdot 2^{-8} = 37.9Hz \quad \textbf{(5.17)}$$

$$f_{c,max} = f_0 \cdot 2^{-1} = 9.704 \cdot 2^{-1} = 4.852KHz \quad \textbf{(5.18)}$$

The 8 corner frequencies are equidistant in exponential scale.

The phase characteristics correspond to the 8 LPF (180º to 90º) and 8 HPF (-90º to -180º).

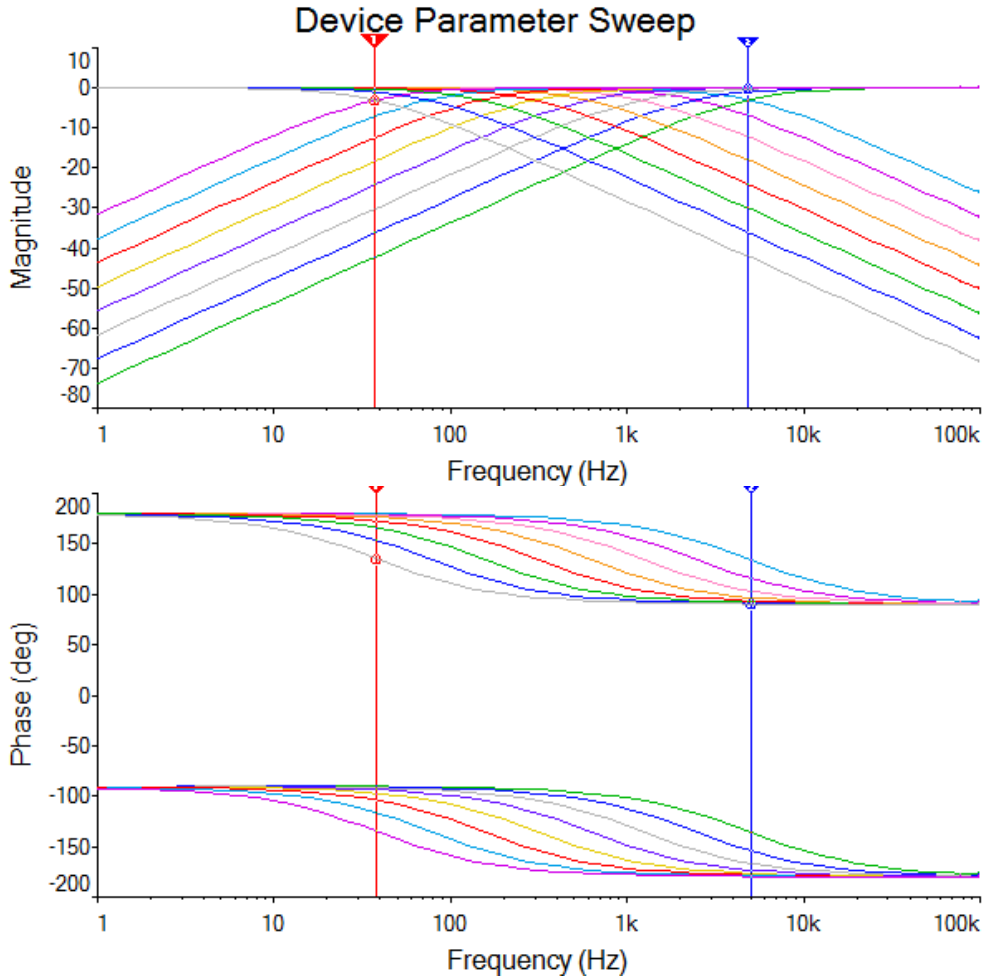


Figure 5.3 Filter spectral characteristics: amplitude (up) and phase (down)

5.3 Experiment

Figure 5.4 shows the experimental board. J2 provides access to the scope inputs 1+ and 2+. The negative nodes of the differential scope inputs (1- and 2-) are hard tied to GND.

J3 and J4 expose schematic nodes. Use wires to connect these nodes to the scope inputs.

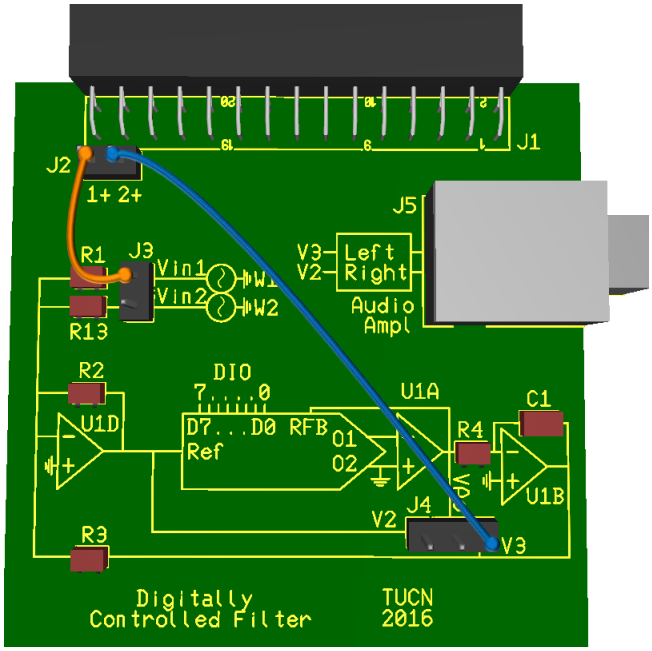


Figure 5.4 Digitally controlled filter PCB

5.3.1 Filter characteristics

The wires in Figure 5.4 probe W_1 and V_3. The WaweForms instruments are set as below:

- Power *Supplies*: ON, +5V and -5V.
- *Network*: Scale: Logarithmic, Start: 20Hz, Stop: 20kHz, Samples: 100, Channel1: Use as reference. Using Channel 1 as reference means:
  - Channel 1 probes the input node in the network (V_{in}=W_1).
  - The network analyzer displays the ratio of Channel2/Channel1 magnitudes, and the phase difference between Channel 2 and Channel1 (the actual network transfer characteristics).
- *StaticIO*: Slider. Click the "multiple windows" icon in the upper-right corner of the Static IO, to open in a separate window.

Run *Supplies*, *Network*, and *Static IO* instruments.

- Observe the network transfer characteristic.
- Identify the filter type (LPF or HPF).

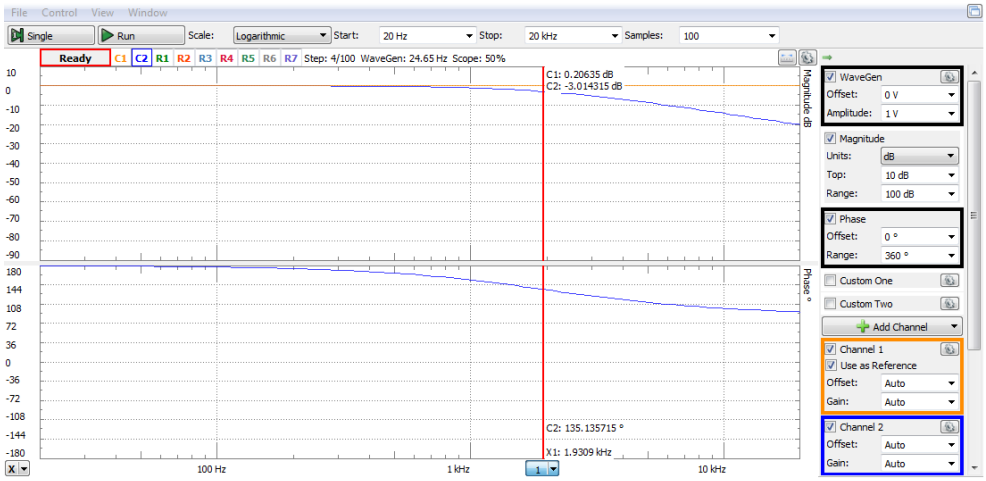


Figure 5.5 The filter transfer characteristics: magnitude (up) and phase (down)

- Use a cursor, to identify the corner frequency of the filter (Magnitude = -3dB, Phase = 135º).
- Move the slider in the Static IO and observe the change of the filter corner frequency.
- Move the Scope Channel2 probe to V_2 and observe the complementary filter, with the same corner frequency.
- In the phase characteristics, notice that the phase for the Pass Band is 180º, due to the inverting structure of the circuit.

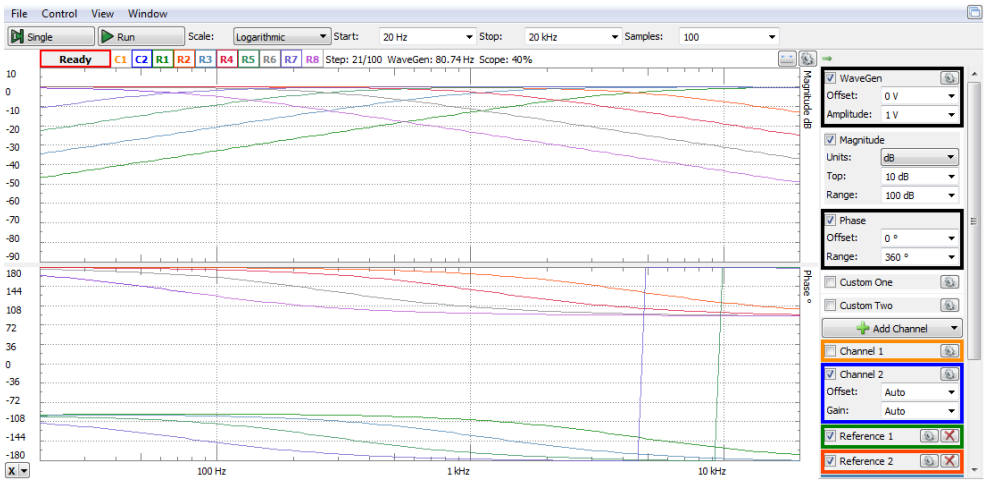


Figure 5.6. Characteristics family for different input numbers.

`To get a family of transfer characteristics, similar to the Figure 5.3 simulation:

- In the *Static IO*, set DIO7-0 as: BitIO/Switch/PushPull. In each step, set a single bit *High*, and all others *Low*. Do the step sequence: 7, 5, 3, 1.
- For each step, wait for the network instrument to complete the characteristics, then click *AddChannel/(Reference)Channel2*. Then swap channel2 probe between V_2 and V_3 and add another reference.
- In 4 steps, add characteristics for both LPF and HPF, for all the input numbers 128, 32, 4, 1 (as set in the Static IO).
- The figure shows half of the Figure 5.3 simulated characteristics (the network analyzer can support max 8 reference channels).

To see both LPF and HPF at once, as in Figure 5.7, place the scope probes on V2 and V3 and uncheck Channel1/Use as reference. This results in:

- Two magnitude separate characteristics are shown for Channels 1 and 2 respectively.
- The magnitude characteristics are not related to a "reference" channel.
- The phase characteristic cannot be computed and displayed.

Task 1. Measure the corner frequency corresponding to the input number which is your birthday.

5.3.2 Audio experiment

Plug a high-quality headset in J5. Set the *Network* Analyzer as:

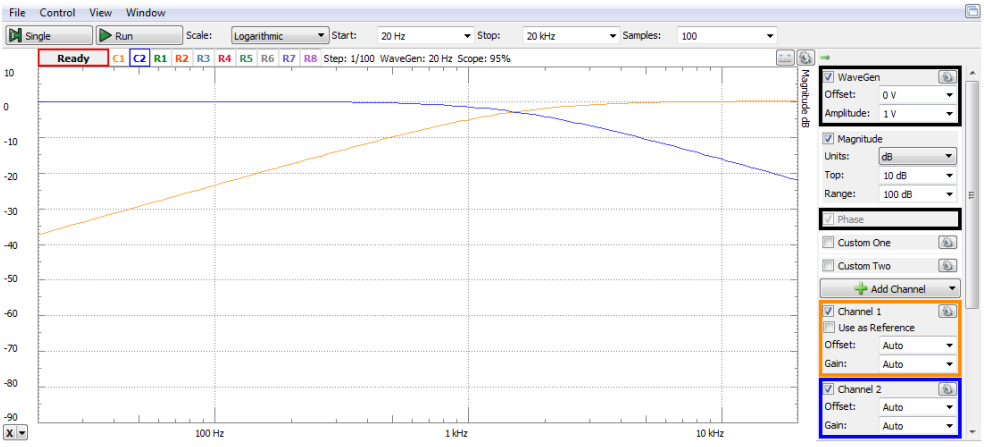


Figure 5.7 Magnitude characteristics

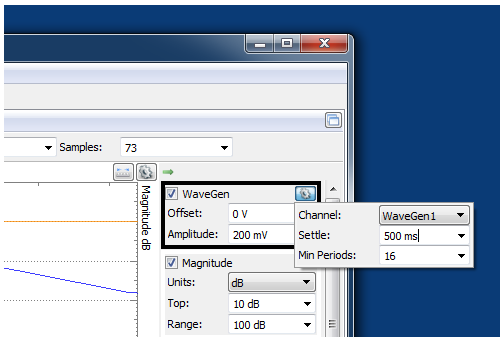


Figure 5.8 WaveGen Settle time

- *Start*: 55Hz, Stop: 3.52kHz. This covers 6 octaves, centered to 440Hz, which is the frequency of A tone.
- *Samples*: 73. That divides the range above in 72 logarithmic-equidistant intervals: 6 octaves X 12 semitones/octave. The frequencies generated during measurement will match all the tones and semitones in the range.

- in the *WaveGen* settings, set *Settle time* at 500ms, as in Figure 5.8. This will keep each frequency long time enough for observation.
- in the *WaveGen* box, set the amplitude for a reasonable sound intensity (ex. 200mV).
- Launch the *Network* analyzer.

Task 2. In the *Static IO*, adjust the DIO7-0 number to get the filters' corner frequency at 440Hz (audio A tone, the logarithmic center of the chosen frequency range). Write the number in your report.

Run the *Network* analyzer and listen to the tones. Observe that the lower frequencies are louder in the left speaker, while the higher frequencies are louder in the right one. Play with the input number and notice the effect.

Notice that human senses are logarithmically scaled: exponentially equidistant stimuli parameters produce linear equidistant feeling. Ex: every musical octave corresponds to doubling the frequency of the sound. Same observation is true for sound intensity.

5.3.3 Music filtering

Plug a high-quality headset in J5. Play a song in *WaveGen*:

- Set *WaveGen* in *Play* mode.
- Click *Import* and choose the .wav file to play.
- Click *Play*.

Set the scope:

- Probe V_2 and V_3 with scope channels 1 and 2, respectively.
- Open an FFT view (*View/FFT*).

- Set the appropriate scope *time base* for displaying the audio spectrum in the FFT: i.e., 20ms/div => 40kHz sampling rate => 1Hz…20kHz FFT range.

Listen the audio signal and observe the wave shape in the scope, as in Figure 5.9. Notice that V_2 has predominant low frequency components (grave sounds), while V_3 has more high frequency components (acute sounds):

- The scope time view shows lower frequencies on V_2.
- The FFT view shows higher amplitude for low frequencies on V_2 and for high frequencies on V_3.
- The grave sounds are louder in the V_2 speaker and the acute ones in the V_3 speaker.

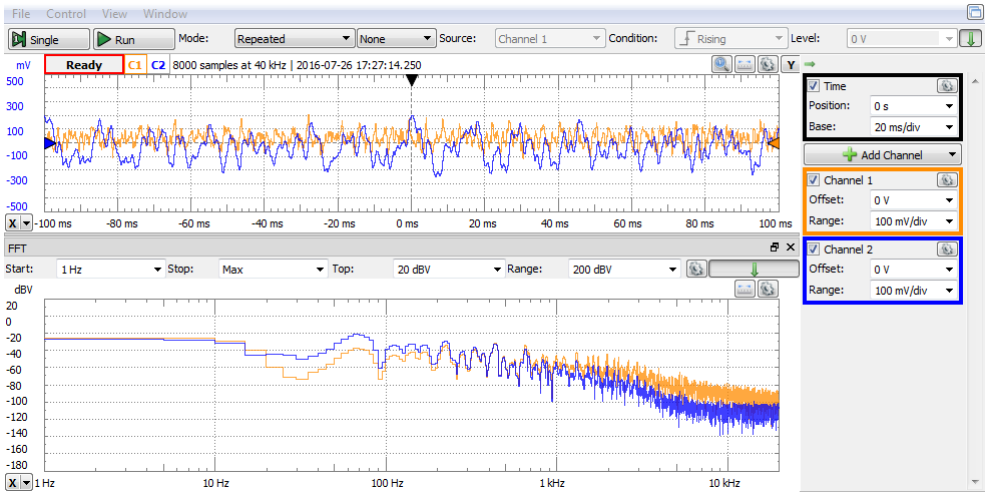


Figure 5.9 Music time signal (up) and spectrum (down) separated by the filter

6 Feedback ADC – voltage comparison

6.1 Background

6.1.1 Unipolar operation

The unipolar binary number in Figure 6.1 can be understood either as "fractional", composed of bits a_i, with $i=1...n$, $a_1=MSB...a_n=LSB$:

$$\{A\} = 0.a_1 \dots a_{(n-1)}a_n = \sum_{i=1}^{n} a_i \cdot 2^{-i}; \quad \{A\} \in [0\dots1) \tag{6.1}$$

Or, as "integer", built of bits b_j, with $j=0...n-1$, $b_{n-1}=MSB...b_0=LSB$:

$$\{N\} = b_{n-1} \dots b_1 b_0 = \sum_{j=0}^{n-1} b_j \cdot 2^{j}; \quad \{N\} \in [0\dots 2^n) \tag{6.2}$$

Where: $b_{n-i} = a_i \ \forall i=1...n$.

$$\{N\} = 2^n \cdot \{A\} \tag{6.3}$$

The equivalent resistance of pin V_{ref} to GND is R; the current in pin V_{ref} is:

$$I_{ref} = \frac{V_{ref}}{R_{ech}} \tag{6.4}$$

In Figure 6.1, the CMOS DAC with Operational Amplifier output stage generates a comparison voltage:

$$V_{cmp} = -I_{out1} \cdot R_f = -V_{ref} \cdot \frac{R_f}{R_{ech}} \cdot \{A\} \tag{6.5}$$

$$= V_{FS} \cdot \{A\} = V_{LSB} \cdot \{N\}$$

$$V_{cmp}, V_{in} \in [0\dots +V_{FS}) \tag{6.6}$$

$$V_{LSB} = \frac{V_{FS}}{2^n} \tag{6.7}$$

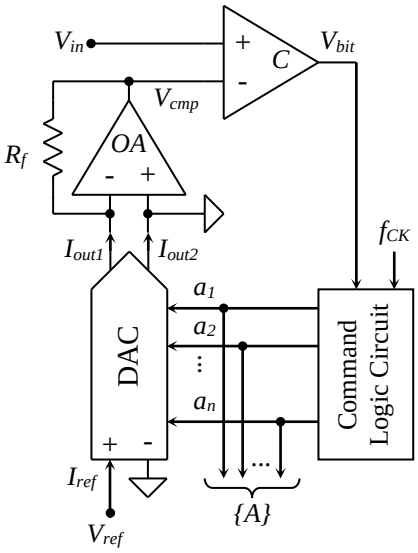


Figure 6.1 Feedback *ADC*, voltage comparison (example)

The Command Logic Circuit implements a multi-step searching algorithm, to find the best digital representation of the instantaneous value of V_{in}, i.e. a number which, converted back to analog, generates a

comparison voltage close to V_{in} (with less than $1V_{LSB}$ difference). At the end of a conversion:

$$V_{cmp,final} = V_{FS} \cdot \{A\}_{final} = V_{LSB} \cdot \{N\}_{final} \cong V_{in}$$
$$\left|V_{cmp} - V_{in}\right| < 1V_{LSB} \quad \textbf{(6.8)}$$

The search algorithm defines the command logic behavior.

6.1.1.1 Counter *ADC*

In Figure 6.2, the command logic core is a counter. Figure 6.3 explains the behavior of the *ADC*. The thin line is V_{in}, the thick line shows both V_{cmp} and $\{A\}$. The same graphic unit represents V_{LSB}, for voltages, respectively *1LSB*, for number $\{A\}$.

Before the conversion begins, the counter is reset. In step $N=0$, the *DAC* generates $V_{cmp,0}=0$. At the end of step *0* ($t=T_{CK}$):

$$V_{in}(T_{CK}) > V_{cmp,0} = 0 = > V_{bit} = 1 \quad \textbf{(6.9)}$$

The counter is enabled, so each *CK* active edge increments its content by *1LSB*, so V_{cmp} grows by V_{LSB}.

The counter is disabled and the conversion ends at the clock period *N*, when:

$$V_{in}\big((N+1)\cdot T_{ck}\big) < V_{cmp,N} = \{N\} \cdot V_{LSB} = \{A\} \cdot V_{FS} \rightarrow V_{bit} = 0 \quad (6.10)$$

$\{N\}_{final}$ is the current content of the counter and is the final result of the AD conversion, proportional to V_{in} at the end of the conversion. V_{bit} can be used as *EOC (End Of Conversion)*, to validate $\{N\}_{final}$.

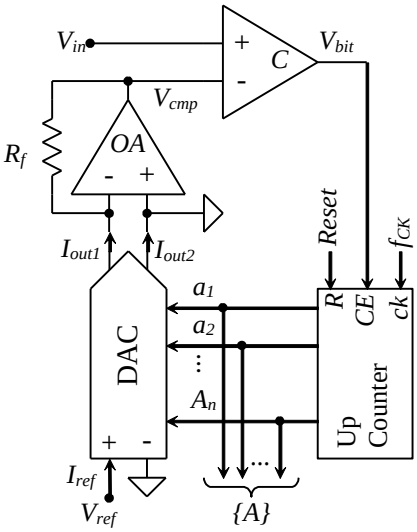


Figure 6.2 Counter *ADC*, voltage comparison (example)

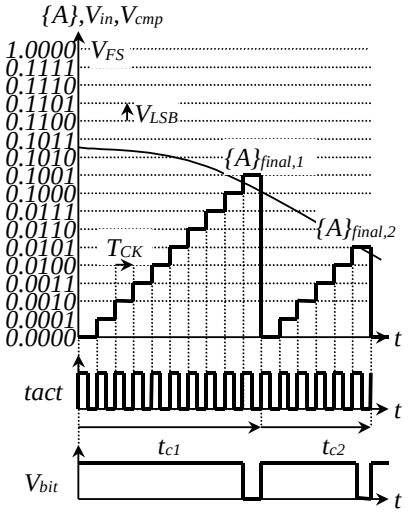


Figure 6.3 4-bit counter *ADC* behavior. Voltage comparison

$$\{N\}_{final} \cong \frac{V_{in}((N+1)\cdot T_{CK})}{V_{LSB}} \qquad \textbf{(6.11)}$$

$$\{A\}_{final} \cong \frac{V_{in}((N+1)\cdot T_{CK})}{V_{FS}} \qquad \textbf{(6.12)}$$

The schematic is simple, but the conversion time is large and variable with V_{in}. In the worst case (V_{in}=V_{FS}):

$$tn_{CK\,c,max} \qquad \textbf{(6.13)}$$

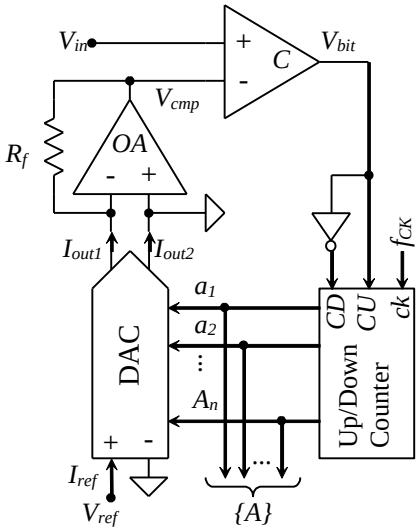


Figure 6.4 Up-Down Counter *ADC*, voltage comparison

6.1.1.2 <u>*Up-Down Counter (Delta) ADC*</u>

In Figure 6.4, the counter is reversible, with separate *Count Up* and *Count Down* enable pins. When V_{in}>V_{cmp}, *CU* is enabled and the counter increments the content at each *ck* rising edge, otherwise *CD* is enabled and the counter decrements at *ck* edges. As result, V_{cmp} always tend to follow V_{in}, in steps of $\pm V_{LSB}$, each clock period.

The circuit generates a new correct numerical value each clock period, as long as the input voltage does not change faster than the maximum V_{cmp} speed:

$$\left(\frac{dV_{in}}{dt}\right) < \left(\frac{dV_{cmp}}{dt}\right)\frac{V_{LSB}}{T_{CK\,max}} \qquad \textbf{(6.14)}$$

If V_{in} slope exceeds the limit in (6.14), V_{cmp} cannot "track" it, and digital values are not accurate, until "catching" back.

The Up-Down counter works as an integrator. Controlled by the V_{bit} signal, it adds or subtracts a quantum (*delta = 1LSB*) to the previous accumulated value.

For this reason, the circuit can be also labeled as *Delta ADC*.

V_{bit} and f_{CK} signals can be sent to a remote system, which replicates the local command

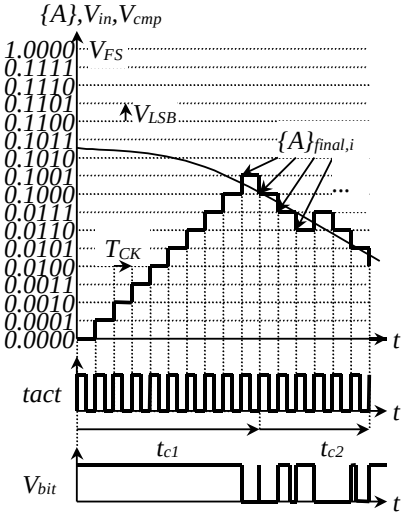


Figure 6.5 4-bit Up-Down counter *ADC* behavior. Voltage comparison

logic structure. Working synchronously, the local and remote counters always have the same content. V_{bit} is the *delta-modulated* representation of V_{in}.

6.1.1.3 *Successive Approximation Register ADC*

It is the most used feedback *ADC*. The command logic implements a *Successive Approximation Register (SAR)*, (as in Figure 6.7) which also names the whole converter.

V_{cmp} (and number *{A}*) "searches" the V_{in} value using a halving algorithm: each conversion step, the *SAR* generates a *partial result {A}*, which approximates better and better the final result $\{A\}_{final}$. Figure 6.6 exemplifies the principal behavior for a 4-bit *SAR ADC*.

Two presumptions are accepted:

- V_{in} belongs to its' definition range (initial searching interval):

$$V_{in} \in [0 \ldots V_{FS}) \qquad \textbf{(6.15)}$$

- V_{in} is constant during each conversion:

$$V_{in}(t) = cst \ \forall t \in [0 \ldots t_c] \qquad \textbf{(6.16)}$$

In Figure 6.6, the (6.16) presumption is violated, by purpose.

It takes *n* steps (*ck* periods) to complete an *n*-bit binary conversion, one bit per step, starting with the *MSB*. In the first step, all bits of *{A}* are set "*0*", but *MSB* is set *"1"*. This generates $V_{cmp}=VFS/2$, halving the initial searching interval. At the end of the first step, *MSB* remains *"1"* if V_{bit} ="1", otherwise is set to *"0"*. The next bit is set *"1"*, so the searching interval is halved by the appropriate value of V_{cmp}. At the end of each step, the current bit takes the value of

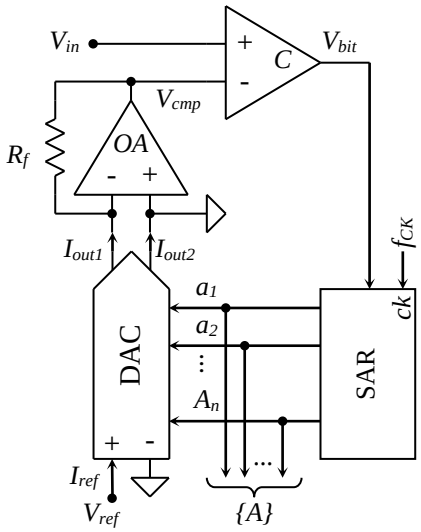


Figure 6.7 *SAR ADC* example. Voltage comparison

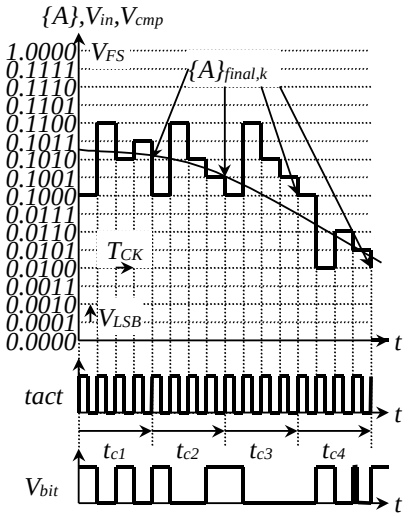


Figure 6.6 4-bit *SAR ADC* behavior. Voltage comparison

V_{bit}, the next bit becomes "*1*" and all the other bits stay unchanged, as shown in the Figure 6.8 state diagram.

The bolded transitions correspond to the first conversion shown in Figure 6.6. The state variables (also used as output signals) are the bits of number *{A}*.

The conversion time is constant, an *n*-bit conversion takes *n* clock periods:

$$t_c = n \cdot T_{CK} \qquad \textbf{(6.17)}$$

An additional state might be added to deliver the final conversion result.

The *SAR ADC* needs a *Sample and Hold* circuit, in order to keep the input voltage constant during the whole conversion, as shown in the (6.16) presumption.

In the third conversion of Figure 6.6, the (6.16) presumption does not hold:
- during the first conversion step, V_{in} is in the upper half of the definition range, *MSB* is assigned $a_1=1$, leading to a final result $\{A\}_{final} \geq 1/2$.
- during all the subsequent steps, V_{in} is "searched" in the $[V_{FS}/2 ... V_{FS})$ sub-range (or sub-sub-ranges of it).
- during the conversion, V_{in} decreases below $V_{FS}/2$.
- the final result, $\{A\}_{final} = 1/2$ reflects neither the value V_{in} had at the conversion begin, nor the one at the end.

If a *Sample and Hold* circuit is added, it samples the V_{in} signal before the conversion begins. The final result then represents the V_{in} value at the sampling moment.

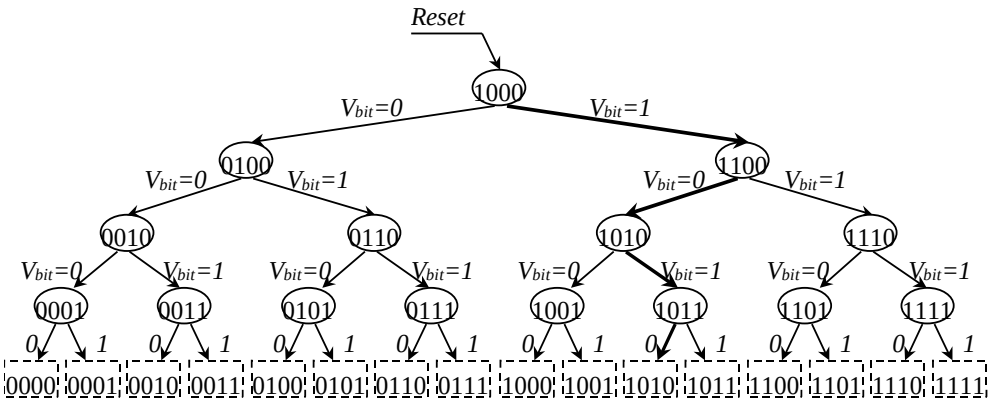


Figure 6.8 4-bit Successive Approximation Register state diagram

6.1.2 Bipolar operation

OA_2 in Figure 6.9 is a current mirror, so:

$$I_2 = I_1 = I_{out2} \tag{6.18}$$

$$I_f = I_{out1} - I_2 = I_{out1} - I_{out2} = I_{ref} \cdot \left(\{A\}_U - \{\overline{A}\}_U\right) \cong \tag{6.19}$$

$$\cong I_{ref} \cdot (2 \cdot \{A\}_U - 1) = I_{ref} \cdot \{A'\}_{BO}$$

$$\{A'\}_{BO} = (2 \cdot \{A\}_U - 1) \tag{6.20}$$

$$R_f = R_{ech} = \frac{V_{ref}}{I_{ref}} \tag{6.21}$$

$$V_{cmp} = -I_f \cdot R_f = -V_{ref} \cdot \{A'\}_{BO} \tag{6.22}$$

V_{cmp} is generated by an *n*-bit *Binary Offset*, bipolar DAC. The whole circuit in Figure 6.9 is an *n*-bit *Binary Offset*, bipolar ADC.

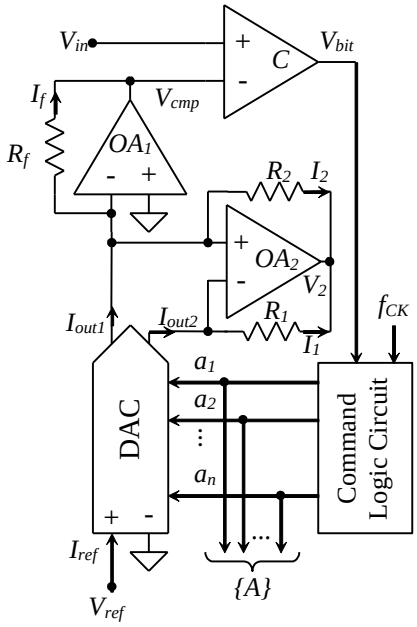


Figure 6.9 Binary Offset bipolar Feedback *ADC*, voltage comparison (example)

$$V'_{FS} = V_{FS} = -V_{ref} \tag{6.23}$$

$$\{A'\}_{BO,final} = \frac{V_{in}}{V_{FS}} \in [-1; 1) \tag{6.24}$$

$$V_{in}, V_{cmp} \in [-V_{FS}; V_{FS}) \tag{6.25}$$

$$\{N'\}_{BO,final} = \frac{V_{in}}{V'_{LSB}} \in [-2^{n-1}; 2^{n-1}) \tag{6.26}$$

$$\{N'\}_{BO} = 2^{n-1} \cdot \{A'\}_{BO} \tag{6.27}$$

$$V'_{LSB} = \frac{2 \cdot V_{FS}}{2^n} = 2 \cdot V_{LSB} \tag{6.28}$$

Since the *Binary Offset* code uses the same bit combinations, in the same

order (from lowest to highest values) as the *Unipolar Code*, the Command Logic Circuit is identical to the unipolar one, for all the three cases above (counter, up-down counter, RAS).

If the final result is needed in *2's Complement* code, the *MSB* of number $\{A'\}_{BO}$ needs to be complemented.

6.2 Experiment and measurements

The AD7524 CMOS DAC is used on the experimental board. It has a similar structure with AD7520, except:

- It has only 8-bit of resolution.
- Data Latches are provided for compatibility to a microprocessor data bus. If not used, *ChipSelect* and *Write* pins must be tied to GND.

6.2.1 Unipolar Operation

Figure 6.10 shows the schematic of the experimental board. Notice that *J3* is loaded with a jumper on pins *2-3*, to tie Out_2 of *U2* to *GND*, and *J4* is loaded with a jumper on pins *1-2*, to insulate *U1A*. Similarly, *J5* is loaded with a jumper on pins *2-3*, to tie Out_2 of *U4* to *GND*, and *J6* is loaded with a jumper on pins *1-2*, to insulate *U1C*. This way, both *U2* and *U1B*, respectively *U4* and *U1D* work as *unipolar DA* convertors.

U5, *U2* and *U1B* build a circuit equivalent to Figure 6.1, with:

- V_{ref} = *W1*, the AWG channel 1 of the Analog Discovery.
- V_{in} = *W2*, the AWG channel 2 of the Analog Discovery. V_{in} passes through the *Sample-And-Hold* circuit (*U6* and associate circuitry) and generates V_{SH} for the actual feedback ADC.
- *Command Logic Circuit* is included in the Analog Discovery *Patterns* generator, as *ROM Logic*. It uses:

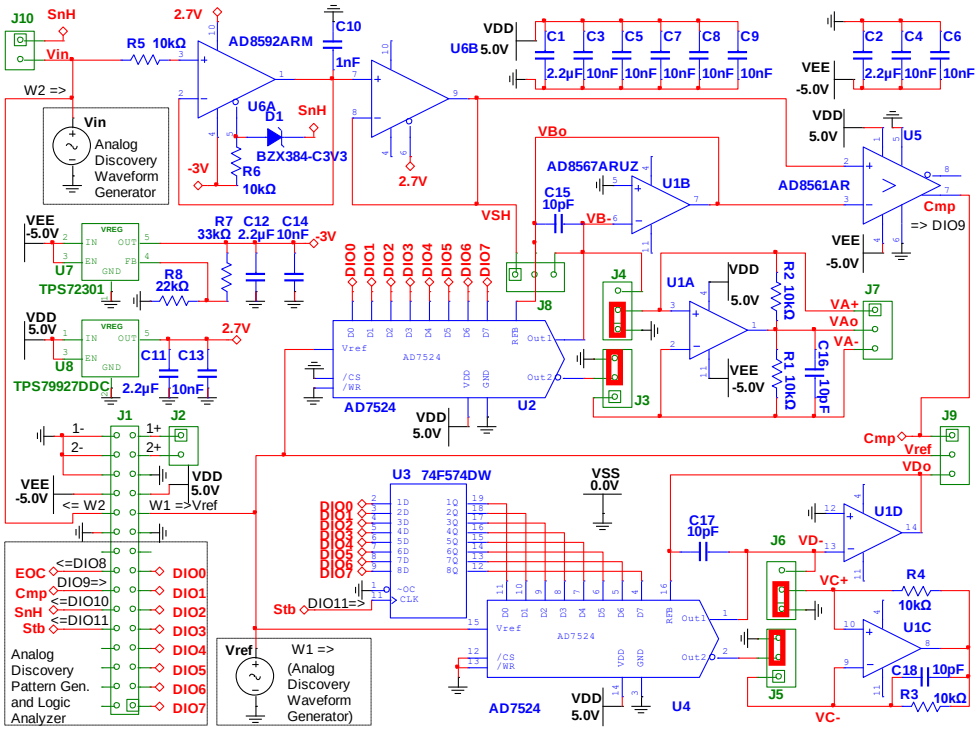


Figure 6.10 Feedback ADC, unipolar operation – experimental board schematic

- *Cmp = DIO9*; input from the comparator *U5*.
- *Data bits = DIO7 (MSB)…DIO0 (LSB)*; outputs for *U2*.
- *SnH = DIO10*, Sample/Hold command; output for *U6*.
- *Stb = DIO11*, Strobe to latch the conversion result; output for *U3*.
- *EoC = DIO8*, End of Conversion; output.

Additional to the circuits in Figure 6.1, the experimental board includes:

- the *Sample and Hold* stage, *U6* and associate circuitry.
- the latch for the final conversion results, *U3*.
- the DAC circuitry to rebuild the acquired signal, *U4*, *U1D* and *U1C*.

Notice *C15*, *C16*, *C17* and *C18*, added to compensate *U1B*, *U1A*, *U1D* and *U1C* respectively. They reduce ringing and oscillation risk for the high speed AD8567 operational amplifiers.

Figure 6.11 shows the experimental board prepared with jumpers and scope probe wires for the unipolar experiment. Notice that the same board is used for several other experiments.

Counter ADC
On the experimental board:

- Place jumpers at:
  - *J3 = 2-3* – ties I_{out2} to GND.
  - *J4 = 1-2* – disconnects *U1A*.
  - *J5 = 2-3* – ties I_{out2} to GND.
  - *J6 = 1-2* – disconnects *U1C*.
- Place scope probe wires:
  - Channel 1 to *J10-2 = W1 =* V_{in}
  - Channel 2 to *J8-2 = VBo =* V_{cmp}

Scope probes will be moved during the experiments, to see various signals.

In the WaveForms software, open workspace *FeedbackADC_Sinus*. It has all instruments prepared for beginning the experiment:

- *Supplies*: both +/-V user voltage supplies active. (If the WaveForms software reports an "Overcurrent condition" and stops the user power

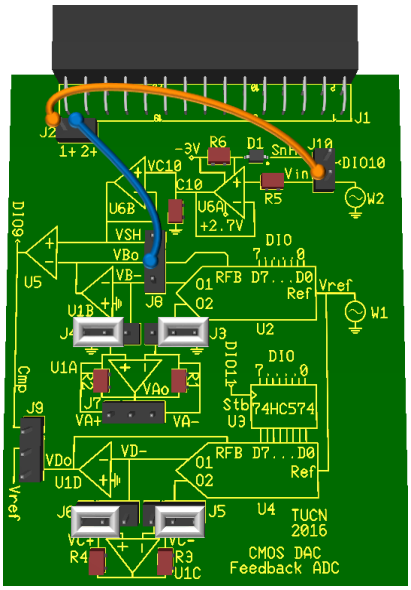


Figure 6.11 DAC CMOS experimental board

supplies, try starting the supplies one by one: first the positive supply and, after few seconds, the negative one)

- *Wavegen*:
  - W1= *DC, Offset: -2V* (constant, negative voltage) for V_{ref}.
  - W2=*Sinus, Frequency: 1kHz, Amplitude: 0.9V, Offset: 1V, Symmetry: 50%, Phase: 0º*.
- *Patterns1*:
  - ROM Logic; State Machine for *Counter ADC*; *1MHz* clock.
    - Inputs: *Cmp* (DIO9).
    - StateBits/Outputs:
      - *EOC* (DIO8) = End of Conversion.
      - *s7…s0* (*DIO7÷DIO0*) = intermediate results of the conversion.
  - ROM Logic for generating *SnH*; *2MHz* clock.
    - Inputs: *EOC* (*DIO8*).
    - Outputs: *SnH* (*DIO10*) = Sample/Hold command for the SH circuit. 0.5µs delayed version of *EOC*.
  - ROM Logic for generating *Stb*; *2MHz* clock.
    - Inputs: *EOC* (*DIO8*).
    - Outputs: *Stb* (*DIO10*) = Strobe command for the final result Latch. 0.5µs delayed version of *EOC*.
- *StaticIO*: set *SnH* (*DIO10*) *Open-Source Switch = "1"*, to override the *SnH* generated by the ROM logic, and force *VSH* to permanently follow V_{in}. ("*Z"* would allow normal behavior of *SnH* ROM logic).
- *Scope*:
  - *AddChannel/Digital/Signal*: *Stb=DIO11, SnH=DIO10, Cmp=DIO9*
  - *AddChannel/Digital/Bus DIO7…DIO0*. Expand the Bus line in the digital view, to see the graphical representation of the bus values.
  - *Time: 500µs/div, Ch1, Ch2 Range: 500mV/div.*
- *Spectrum*: *ChannelOptions*/both channels: *Offset: 0V, Range: 5V*

Patterns2 and *Patterns3* will be used in later experiments.

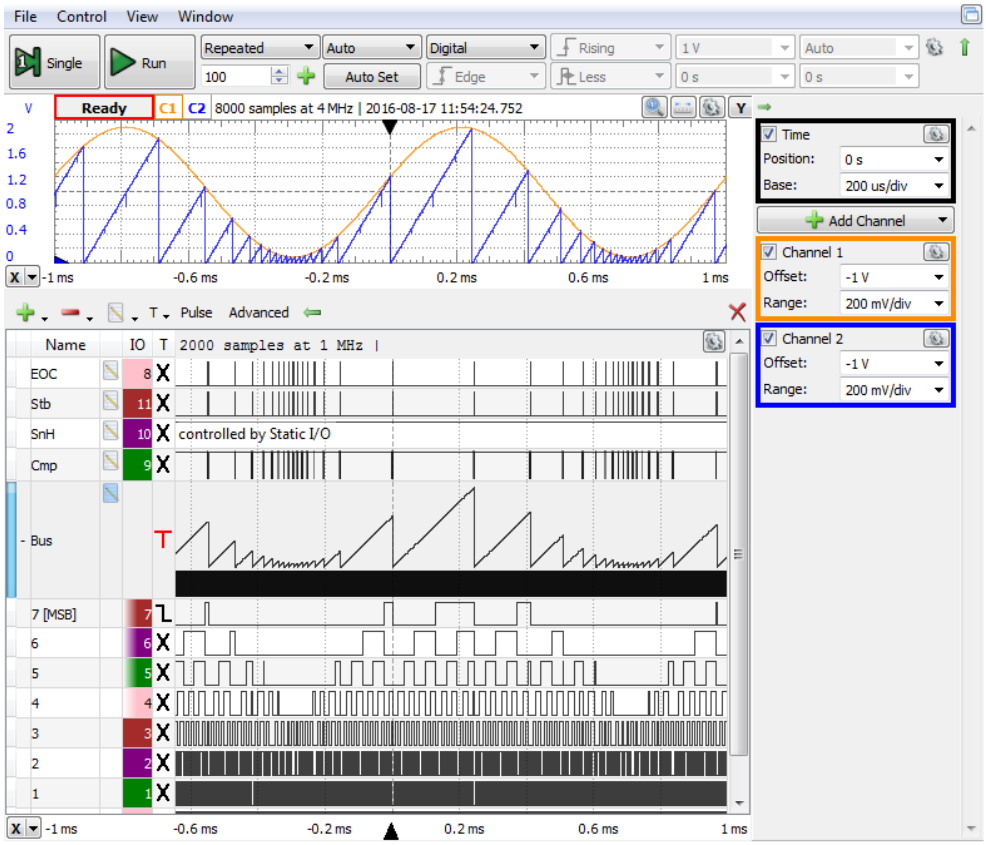


Figure 6.12 Unipolar Counter ADC experiment

Run *Supplies, Wavegen1, Patterns1, Static IO, and Scope.* Observe *Scope*, as in Figure 6.12:

- In the analog Time view:
  - V_{in} is scaled to almost all the definition range ($0.1V \div 1.9V$).
  - $VBo = V_{cmp}$ starts from zero in each conversion and rises to V_{in}, with constant slope V_{LSB}/T_{ck}.
- In the digital Time view:
  - Individual bits *Cmp, EOC, Stb* and the bus. Observe the graphical representation of the bus values, similar to V_{cmp}.

Task 1. Change the Scope Time Base to detail the *End of Conversion* moments. Use cursors to read V_{in}, V_{cmp} and $\{N\}$ at the start of *EOC* pulse. Verify the equation (6.11). Repeat for 3 successive conversions.

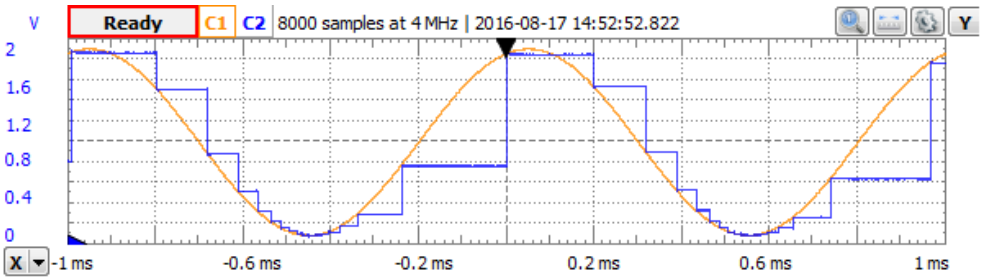


Figure 6.13 Unipolar Counter ADC – Re-constructed signal

Task 2. Identify the beginning and the end of the conversions above. Use cursors to read the conversion time, t_c. Verify that t_c is proportional to V_{in} and *{N}*. Write the t_c equation.

Move scope channel 2 probe to J9-3, to observe the re-constructed, 0-degree interpolated *VDo* signal, as in Figure 6.13. Identify the beginning and the end of the conversions.

Run the *Spectrum Analyzer* as in Figure 6.14. Measure Channel 2 *THD*, *ENOB* and *SNR* (*View/Measure/Add/Trace2/*). Observe:

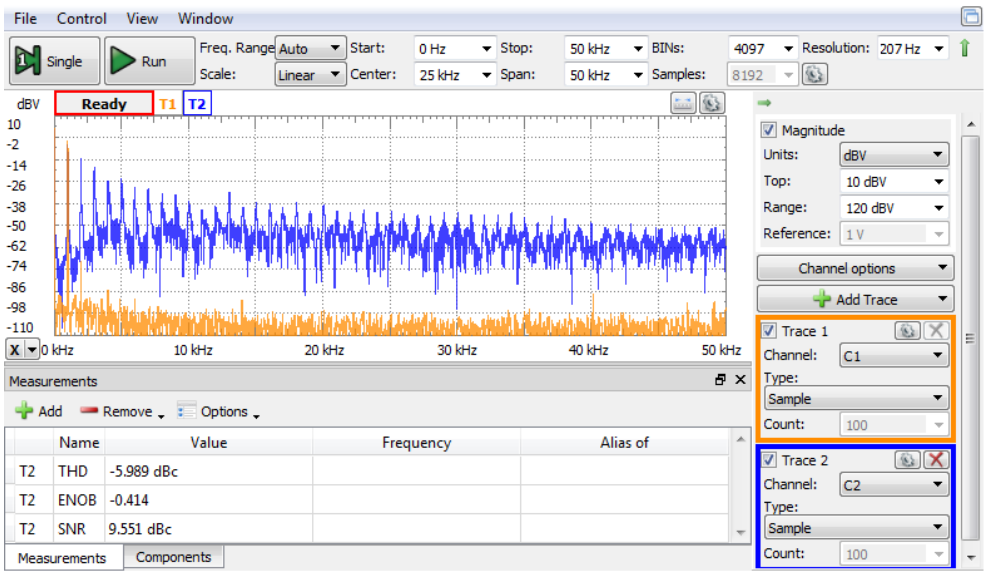


Figure 6.14 Unipolar Counter ADC – Spectrum of the re-constructed signal

- The sampling frequency is higher than required by the Nyquist theorem, but too low for high quality acquisition; see THD, ENOB, SNR values.

- Non-uniform sampling generates multiple spectral components of *VDo*.

6.2.1.1 *Up/Down Counter ADC*

- *Patterns2* (WaveForms allows more instruments of same type, but only one can Run at a time):
  - ROM Logic; FSM for *Up/Down Counter ADC*; *1MHz* clock.
    - Inputs: *Cmp* (DIO9).
    - StateBits/Outputs:
      - *EOC* (DIO8) = *End of Conversion* = permanently active.
      - *s7…s0* (*DIO7…DIO0*) = intermediate results of the conversion
  - ROM Logic for generating *SnH*; *2MHz* clock.
    - Inputs: *EOC* (*DIO8*).
    - Outputs: *SnH* (*DIO10*) = *Sample/Hold* command for the SH circuit. Permanent active, for continuous *Sampling* (no *Hold*).
  - Clock for generating *Stb*; *2MHz* clock. Generates pulses for continuously update the output latch.

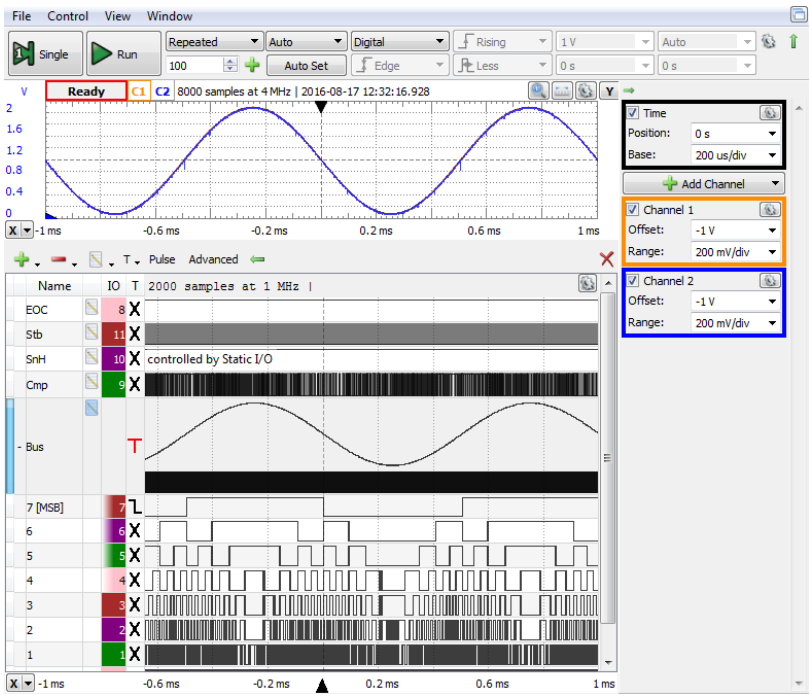


Figure 6.15 Unipolar Up/Down Counter ADC experiment

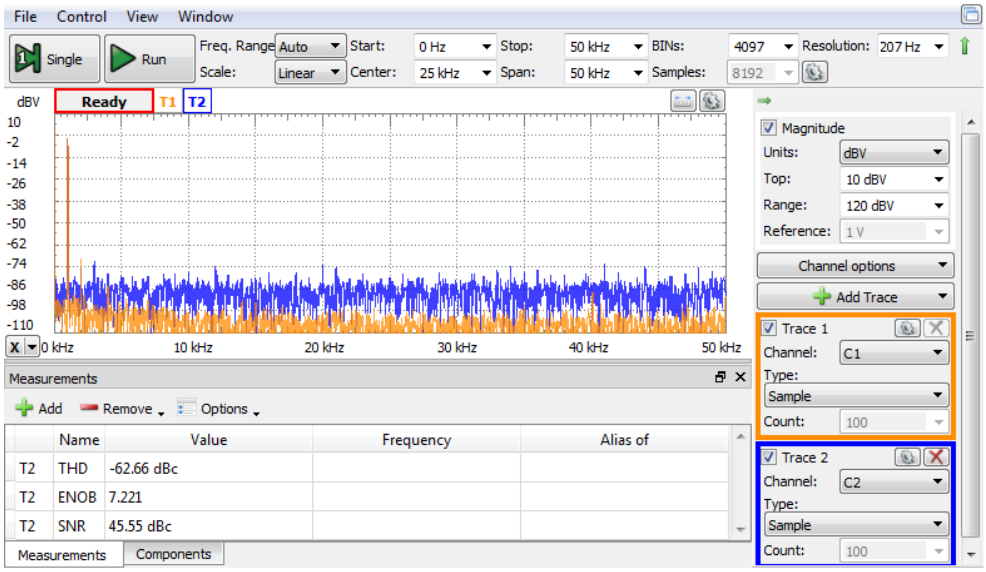


Figure 6.16 Unipolar Up/Down Counter ADC – Spectrum of the re-constructed signal

Run *Supplies, Wavegen1, Patterns2, Static IO, and Scope.* For comparison, both V_{in} and ADC clock are same as in the previous experiment. Observe *Scope*, as in Figure 6.15:

- *VDo* (re-constructed signal) is identical to $VBo = V_{cmp}$.
- *VDo* looks very close to V_{in}. The spectrum and measurements in Figure 6.16 show much higher quality acquisition, compared to the Counter ADC.
- Doubling the frequency of V_{in} shows the Up/Down Counter ADC limitation, as in Figure 6.17: V_{in} is changing faster than (6.14), so V_{cmp} cannot follow it effectively. Due to condition (6.14), the Up/Down Counter ADC performance is close to the regular Counter ADC.

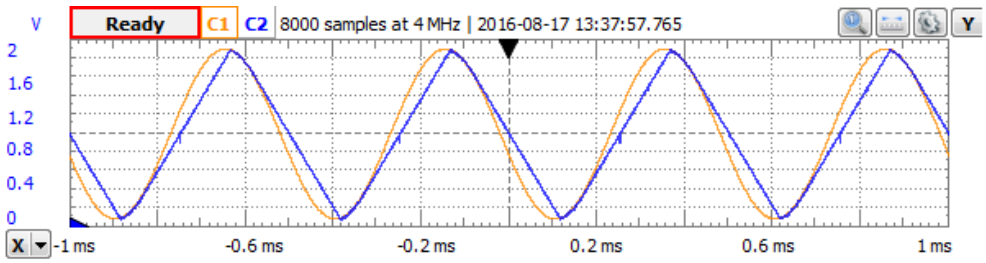


Figure 6.17 Unipolar Up/Down Counter ADC – 2kHz re-constructed signal

6.2.1.2 *Successive Approximation Register ADC*

- *Wavegen*: return to 1kHz V_{in} for comparison under the same conditions.
  - *Patterns3*:
    - ROM Logic; State Machine for *SAR ADC*; *1MHz* clock.
      - Inputs: *Cmp* (DIO9).
      - StateBits/Outputs:
        - *EOC* (DIO8) = End of Conversion.
        - *s7÷s0* (*DIO7…DIO0*) = intermediate results of the conversion.
    - ROM Logic for generating *SnH*; *2MHz* clock.
      - Inputs: *EOC* (*DIO8*).
      - Outputs: *SnH* (*DIO10*) = Sample/Hold command for the SH circuit. 0.5µs delayed version of *EOC*.
    - ROM Logic for generating *Stb*; *2MHz* clock.
      - Inputs: *EOC* (*DIO8*).
      - Outputs: *Stb* (*DIO11*) = Strobe command for the final result Latch. 0.5µs delayed version of *EOC*.
  - *StaticIO*: set *SnH* (*DIO10*) *Open-Source Switch = "1"*, to override the *SnH* generated by the ROM logic, and force *VSH* to permanently follow V_{in}. ("*Z"* would allow normal behavior of *SnH* ROM logic).

Run *Supplies, Wavegen1, Patterns3, Static IO, and Scope.* V_{in} and ADC clock are the same as in the previous experiment. Observe *Scope*, as in Figure 6.18.

Use an extended Time Base (2µs/div) to analyze the steps within an analog-to-digital conversion. Observe:

- A new conversion always begins with $V_{cmp} = ½V_{FS} = 1V$.
- The second conversion step brings V_{cmp} to either $¼V_{FS} = 0.5V$ or to $¾V_{FS} = 1.5V$ to halve the lower half or the upper half of the V_{in} range.
- Next steps halve successive subranges of V_{FS}, to finally get a V_{LSB} range.
- The largest V_{cmp} trip happens in the first step: from the old conversion result to $½V_{FS}$. The worst cases are from 0 to $½V_{FS}$ or from V_{FS} to $½V_{FS}$. The V_{cmp} trip speed is limited by the *U1B Slew Rate*. The *ck* frequency should be low enough for this trip to finish in one *ck* period (step).
- A new conversion begins every *9 ck* periods: *8* periods for conversion, the 9^{th} one (shown by *EOC*) to register the result (on *Stb* rising edge).
- the SAR searching algorithm:
  - bit by bit – bits 7…0 of the bus, in the Digital view
  - the number *{N}* – hexadecimal and graphic, in the Digital view.
  - V_{cmp} – in the analog view.

Task 3. Calculate the sampling frequency.

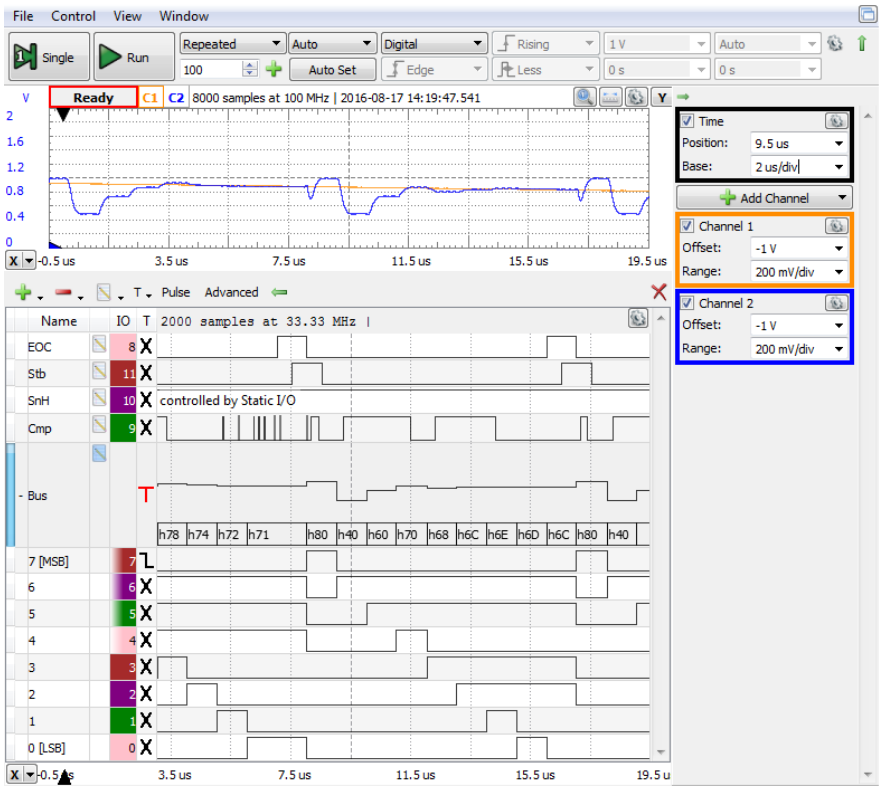


Figure 6.18 Unipolar SAR ADC experiment

Set the scope *Time Base* to *200µs/div*, to compare the sampling rate in Figure 6.19 to the *Counter ADC* (for same *ck* frequency), in Figure 6.12.

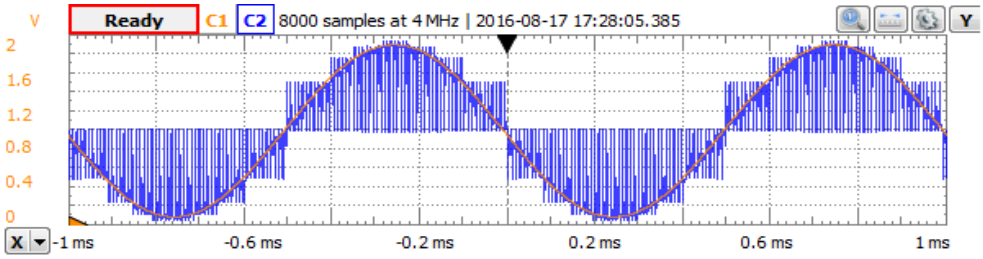


Figure 6.19 Unipolar SAR ADC – V_{in} and V_{cmp}

Move scope Channel 2 to probe the re-constructed signal *VDo*, as in Figure 6.20. Compare with the *Counter ADC*, in Figure 6.13.

The frequency range in the spectrum analyzer was changed in Figure 6.21 to 0…500kHz, to fit the sampling frequency and mirror images. Even so, the spectral image and measured parameters show a much higher acquisition quality compared to the Counter ADC.

For more visible spectral components, change the V_{in} frequency to 10kHz, as in Figure 6.22. Observe the V_{in} mirror images at ±10kHz around the sampling frequency and its 2nd, 3rd and 4th harmonics.

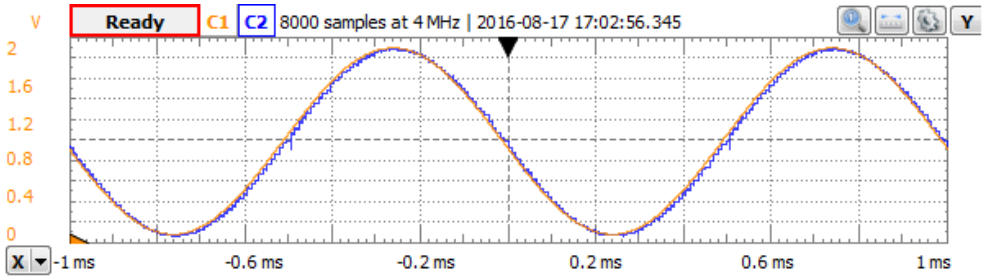


Figure 6.20 Unipolar SAR ADC – reconstructed signal

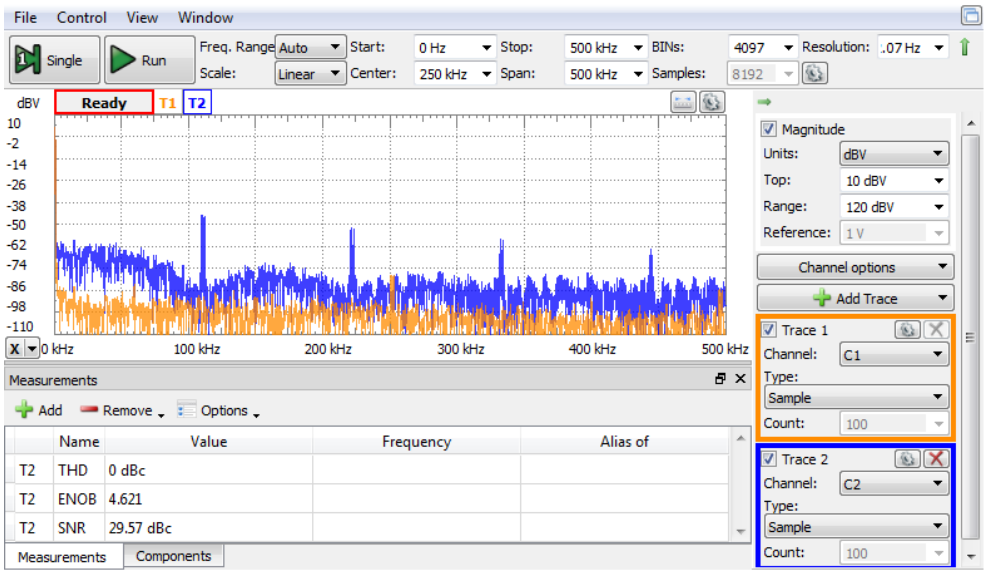


Figure 6.21 Unipolar SAR ADC – 1kHz reconstructed signal spectrum

Figure 6.22 also emphasizes a phenomenon which is difficult to observe in the time domain: the *SAR ADC* needs a *Sample and Hold* stage to provide constant V_{in} during the conversion, as explained in paragraph 6.1.1.3. Both spectral image and measured parameters are better if an *SH* stage is active (in the *Static IO*, switch DIO10 to *Z*, to allow *Patterns 3* to generate *SnH*).

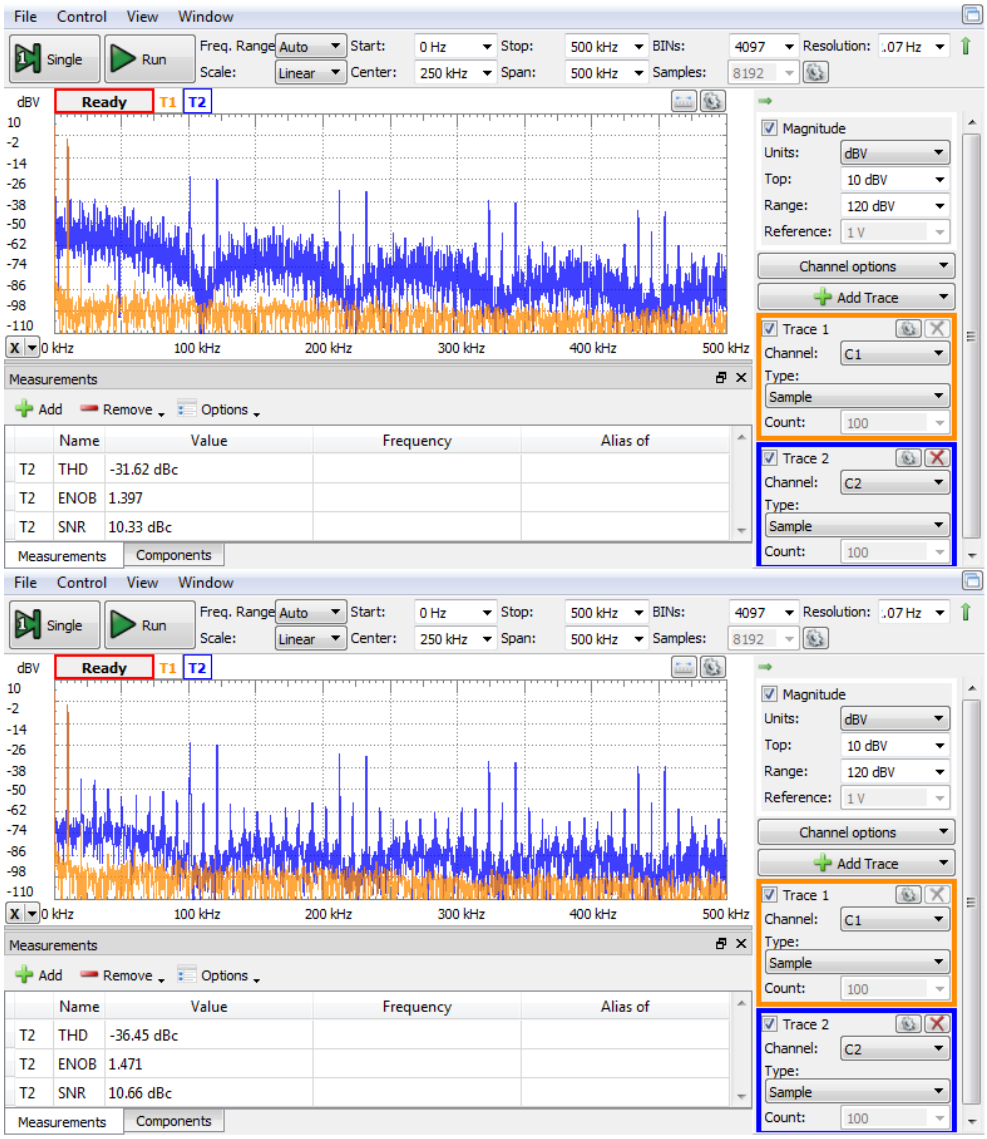


Figure 6.22 Unipolar SAR ADC – 10kHz reconstructed signal spectrum: without *Sample and Hold* (up) and with *Sample and Hold* (down)

6.2.2 Bipolar operation

On the experimental board:

- Place jumpers at:
  - *J3 = 1-2* – ties I_{out2} to *U1A* inverting input.
  - *J4 = 2-3* – connects *U1A* to the I_{out1} node.
  - *J5 = 1-2* – ties I_{out2} to *U1C* inverting input.
  - *J6 = 2-3* – connects *U1C* to the I_{out1} node.

This transforms both *U2* and *U4* to *Binary Offset bipolar DACs*.

In the WaveForms software:

- *Wavegen*:
  - W1= *DC, Offset: -2V* (constant, negative voltage) for V_{ref}. V_{FS}= -V_{ref}.
  - W2=*Sinus, Frequency: 1kHz, Amplitude: 1.9V, Offset: 0V, Symmetry: 50%, Phase: 0°*.
- *Scope* (both channels): *Offset: 0V, Range: 500mV/div.*

As explained in paragraph 6.1.2, since the *Binary Offset* code uses the same bit combinations, in the same order (from lowest to highest values) as the *Unipolar Code*, the Command Logic Circuit is identical to the unipolar one, for all the three cases above (*counter, up/down counter, RAS*).

As result, all the experiments performed in paragraph 6.2.1 can be repeated as bipolar, with similar observations and conclusions.

The following figures are bipolar replica of unipolar similar figures in the previous paragraph. Same order (Counter, Up/Down Counter, SAR) is used. Most of the experiment parameters are also unchanged, except:

- WaveGen, Ch2. amplitude and offset (for a bipolar range *[-1.9V…1.9V)*
- Scope amplitude and offset (for same reason)

Most of the text comments for unipolar converters hold true for bipolar ones, so they are not written again. Just the differences are emphasized below.

$$V_{FS} = -V_{ref} = 2V \quad \textbf{(6.29)}$$

$$V'_{LSB} = \frac{2 \cdot V_{FS}}{2^8} = 15.625mV \quad \textbf{(6.30)}$$

Observe in all figures that the V_{in} and V_{cmp} ranges are double, compared to the unipolar experiments [-V_{FS};V_{FS}). V'_{LSB} is also double, since one of the 8 bits became "sign" bit.

6.2.2.1 *Counter ADC*

Move scope channel 2 probe to J9-3, to observe the re-constructed, 0-degree interpolated *VDo* signal, as in Figure 6.23**.**

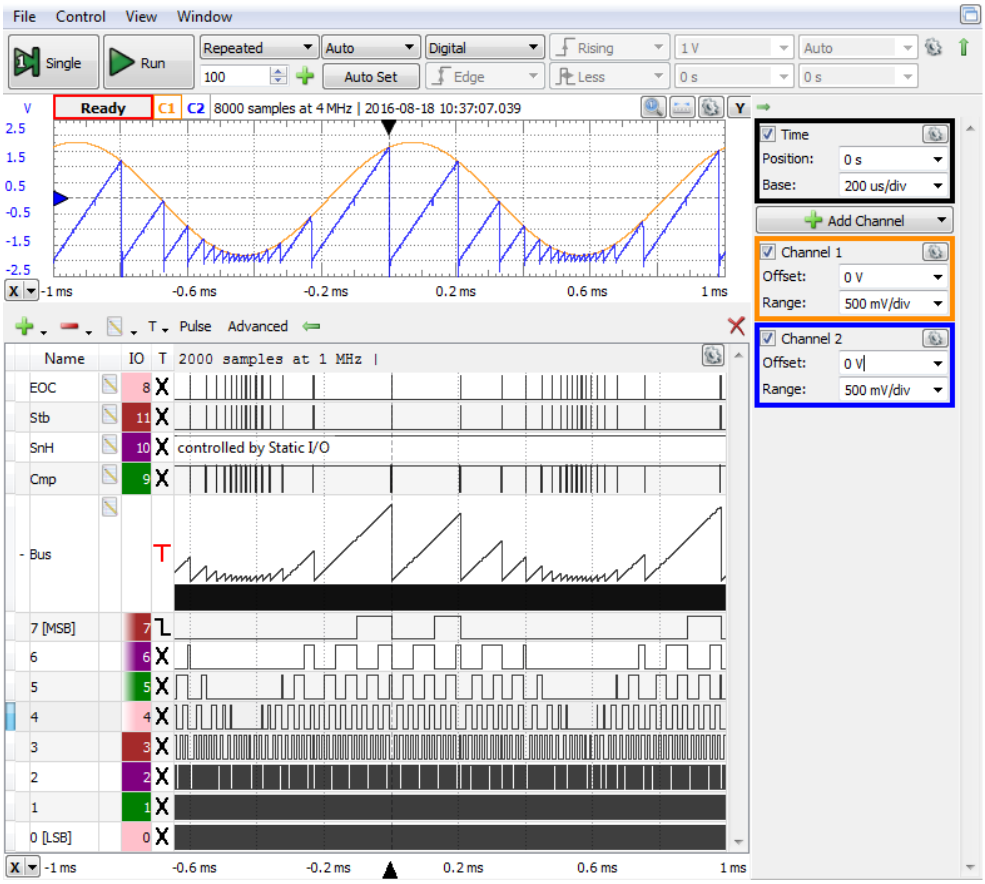


Figure 6.24 Bipolar Counter ADC experiment

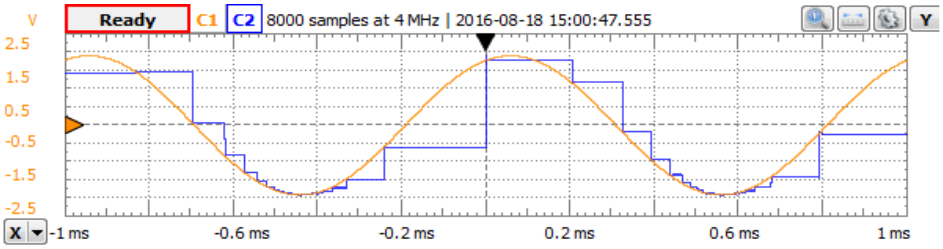


Figure 6.23 Bipolar Counter ADC – reconstructed signal

6.2.2.2 *Up/Down Counter ADC*

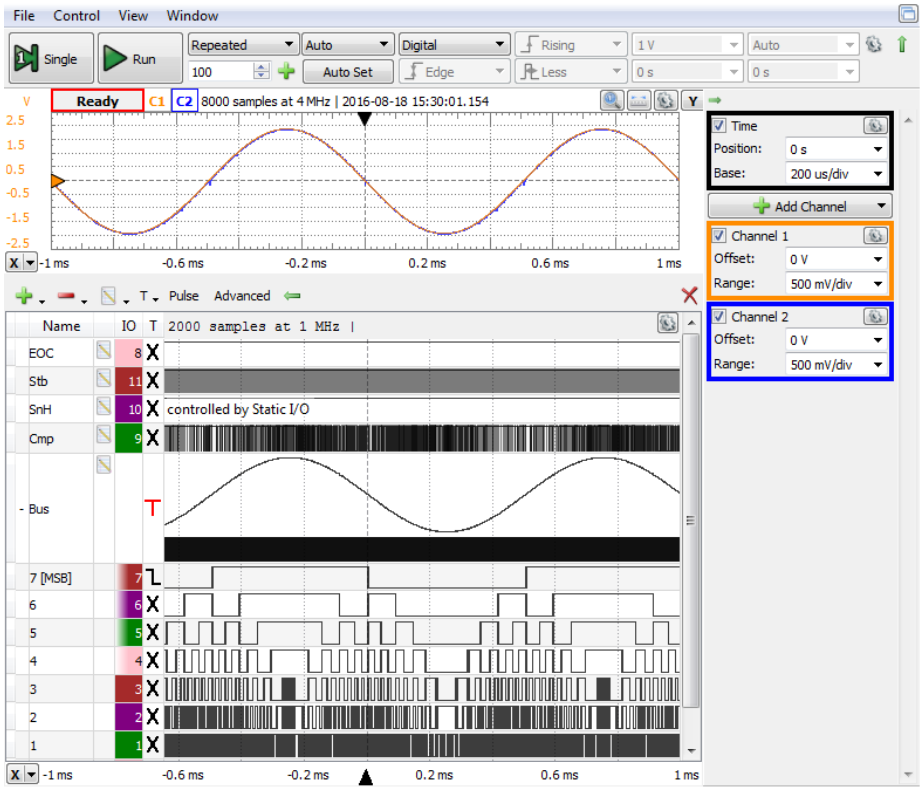


Figure 6.25 Bipolar Up/Down Counter ADC experiment

Double the frequency of V_{in} to see the Up/Down Counter ADC limitation, as in Figure 6.17.

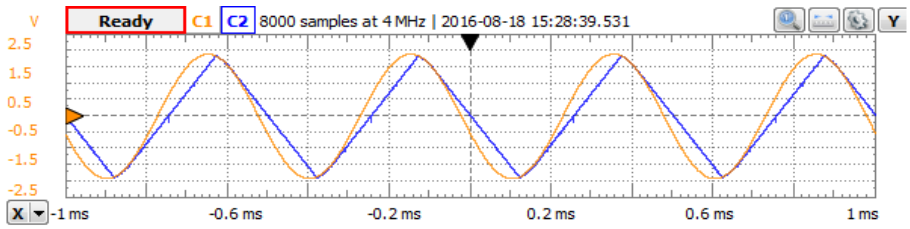


Figure 6.26 Bipolar Up/Down Counter ADC – 2kHz re-constructed signal

6.2.2.1 *Successive Approximation Register ADC*

Use an extended Time Base (2µs/div) to analyze the steps within an analog-to-digital conversion. Observe:

- A new conversion always begins with $V_{cmp} = 0V$, which is V_{in} *midrange*.
- The second conversion step brings V_{cmp} to either $-½V_{FS}=-1V$ or to $½V_{FS}=1V$ to halve the lower or upper half of the V_{in} range.
- Next steps halve successive subranges, to finally get a V'_{LSB} range.

The largest V_{cmp} trip happens in the first step: from the old conversion result to *0V*. The worst cases are from $-V_{FS}$ to *0V* or from V_{FS} to *0V*. The V_{cmp} trip speed is limited by the *U1B Slew Rate*. The *ck* frequency should be low enough for this trip to finish in one *ck* period (step).

Task 4. Use Cursors to measure the *U1B Slew Rate*. Calculate the max *ck* frequency. Calculate the max sampling frequency

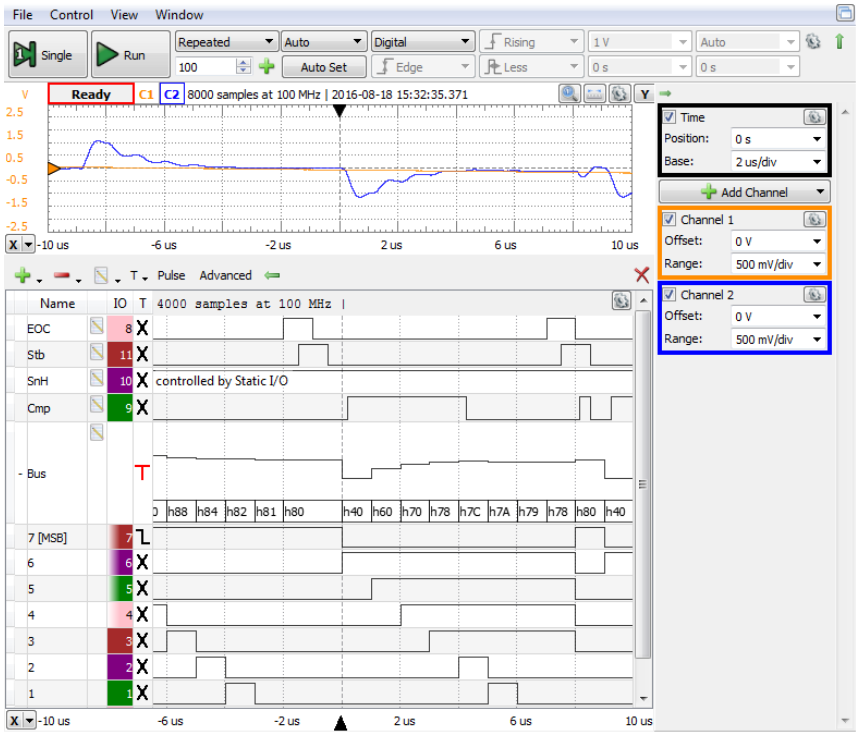


Figure 6.27 Bipolar SAR ADC experiment

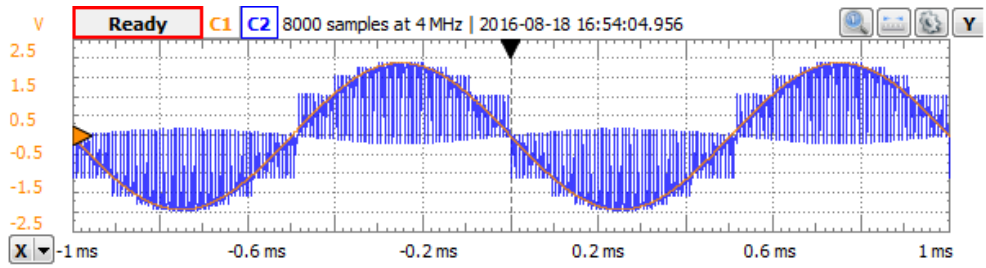


Figure 6.30 Bipolar SAR ADC – V_{in} and V_{cmp}

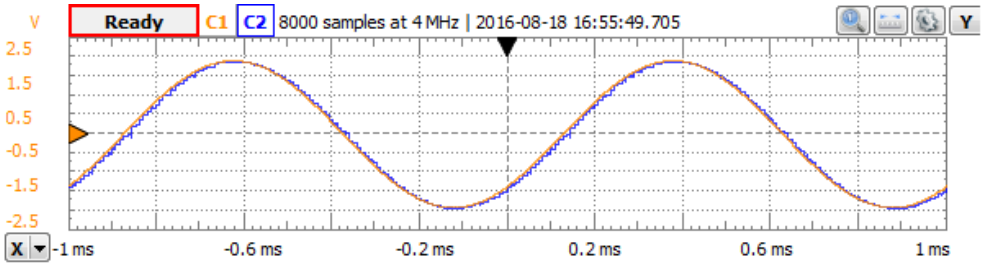


Figure 6.29 Bipolar SAR ADC – reconstructed signal.

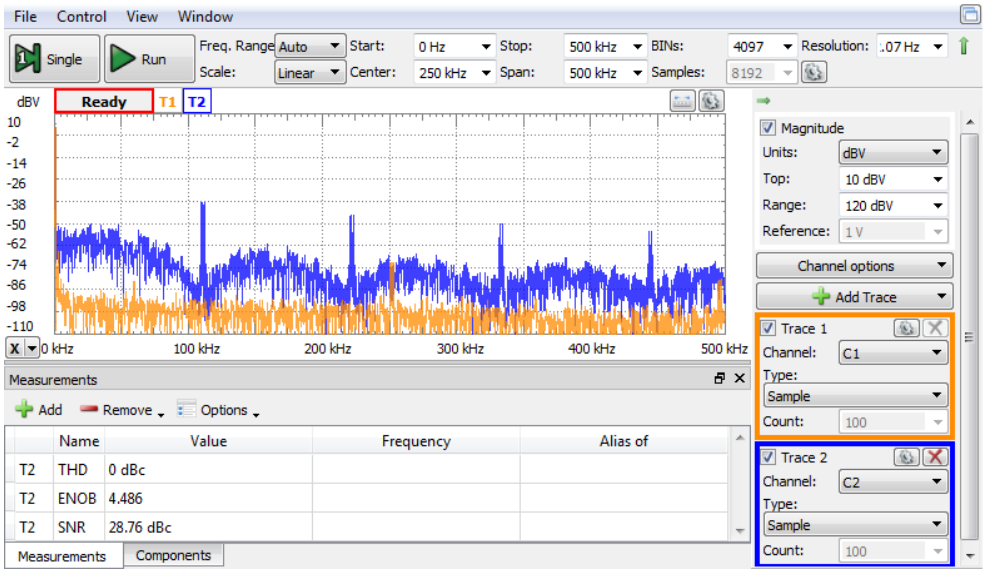


Figure 6.28 Unipolar SAR ADC – 1kHz reconstructed signal spectrum

Observe the *SAR* sampling rate as in Figure 6.30. Observe the re-constructed signal *VDo*, as in Figure 6.29. Observe the 1kHz reconstructed signal

spectrum, as in Figure 6.28. Observe the 10kHz reconstructed signal with and without a Sample and Hold stage, in Figure 6.31.

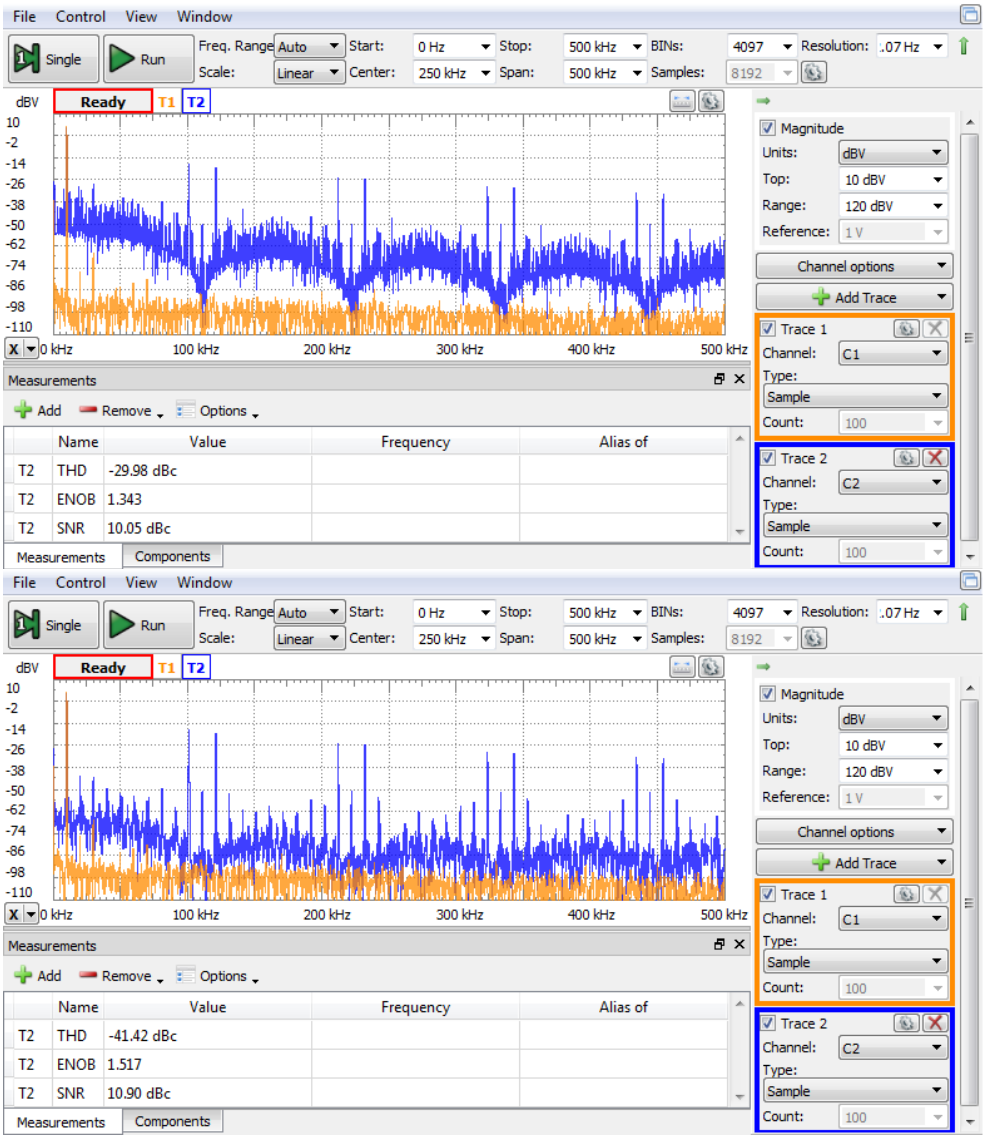


Figure 6.31 Bipolar SAR ADC – 10kHz reconstructed signal spectrum: without *Sample and Hold* (up) and with *Sample and Hold* (down).

7 Voltage Controlled Oscillator

7.1 Background

Figure 7.1 shows a voltage to frequency converter (VCO) example. U_{1C} is a repeater, U_{1B} is an inverter (with $R_1 = R_2$):

$$V_1 = V_{in}; \quad V_2 = -V_{in} \tag{7.1}$$

U_{1A} is a non-inverting *Schmidt Trigger*. AD8567 is a rail-to-rail operational amplifier. Conform to the data sheet, V_5 can trip as close as 20mV to both positive and negative power voltages, for I_{load} = 1mA.

$$\begin{aligned} V_{5,H} &= V_{sup+} - 20mV \approx V_{sup+} = +5V \\ V_{5,L} &= V_{sup-} + 20mV \approx V_{sup-} = -5V \end{aligned} \tag{7.2}$$

The Schmidt trigger thresholds and hysteresis are:

$$\begin{aligned} V_{4,H} &= -V_{5,L} \cdot \frac{R_4}{R_5} \\ V_{4,L} &= -V_{4,H} = -V_{5,H} \cdot \frac{R_4}{R_5} \\ \Delta V_4 &= V_{4,H} - V_{4,L} \end{aligned} \tag{7.3}$$

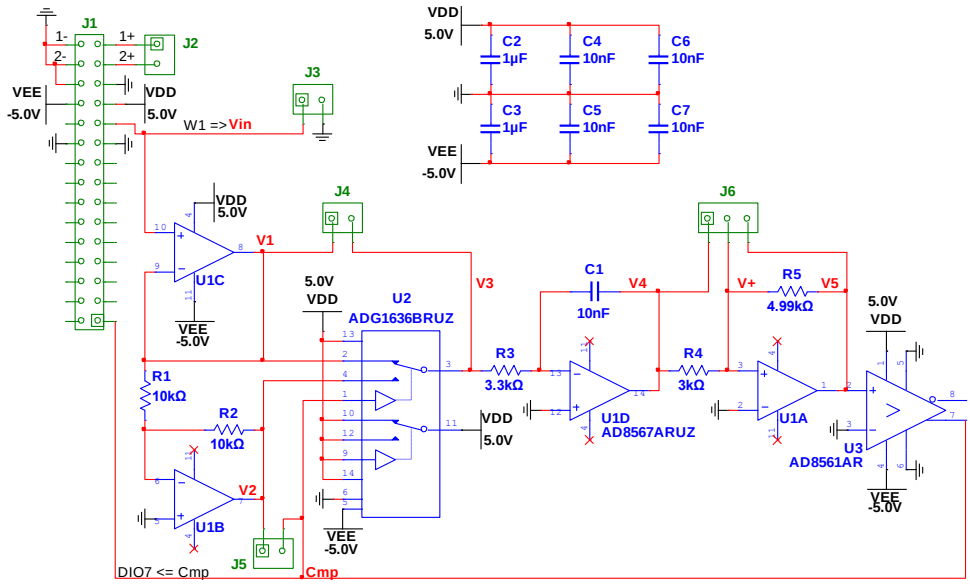


Figure 7.1. Voltage Controlled Oscillator schematic

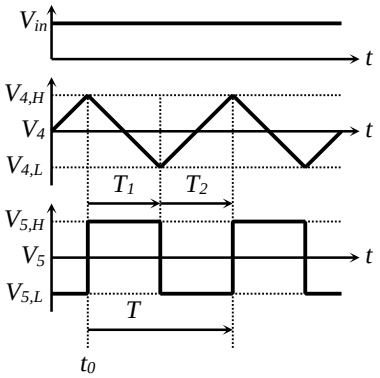


Figure 7.2. VCO behavior. Constant input voltage

The *Cmp* signal drives the U2 switch:

$$\begin{aligned} Phase\ 1{:}\ Cmp = High\ &=> \\ V_3 = V_1\ &= V_{in} \\ Phase\ 2{:}\ Cmp\ = \ Low\ &=> \\ V_3 = V_2\ &= -V_{in} \end{aligned} \quad \textbf{(7.4)}$$

U_{1D} is used in an inverting integrator stage.

Phase 1 begins at time t_0, with $V_4(t_0)=V_{4,H}$.

$$V_4(t_0 + t) = V_{4,H} - \frac{1}{R_3 \cdot C_1} \cdot \int_{t_0}^{t} V_{in}(t) \cdot dt \quad \textbf{(7.5)}$$

Phase 1 ends at time t_0+T_1, with $V_4(t_0+T_1)=V_{4,L}$:

$$V_4(t_0 + T_1) = V_{4,L} = V_{4,H} - \frac{1}{R_3 \cdot C_1} \cdot \int_{t_0}^{t_0+T_1} V_{in}(t) \cdot dt \quad \textbf{(7.6)}$$

$$\Delta V_4 = V_{4,H} - V_{4,L} = \frac{1}{R_3 \cdot C_1} \cdot \int_{t_0}^{t_0+T_1} V_{in}(t) \cdot dt \quad \textbf{(7.7)}$$

Phase 2 begins at time t_0+T_1, with $V_4(t_0+T_1)=V_{4,L}$:

$$V_4(t_0 + T_1 + t) = V_{4,L} + \frac{1}{R_3 \cdot C_1} \cdot \int_{t_0+T_1}^{t} V_{in}(t) \cdot dt \quad \textbf{(7.8)}$$

Phase 2 ends at time $t_0+T_1+T_2$, with $V_4(t_0+T_1+T_2)=V_{4,H}$:

$$V_4(t_0 + T_1 + T_2) = V_{4,H} = V_{4,L} + \frac{1}{R_3 \cdot C_1} \cdot \int_{t_0+T_1}^{t_0+T_1+T_2} V_{in}(t) \cdot dt \quad \textbf{(7.9)}$$

$$\Delta V_4 = V_{4,H} - V_{4,L} = \frac{1}{R_3 \cdot C_1} \cdot \int_{t_0+T_1}^{t_0+T_1+T_2} V_{in}(t) \cdot dt \quad \textbf{(7.10)}$$

A long conversion time, t_c, includes $\{N\}$ pulses of V_4 (or V_5). For variable V_{in}, the pulses are not strictly periodical and successive instances of T_1 and T_2 are not equal to each other. However, $\{N\}$ successive instances of (7.7) and (7.10) can be concatenated to get a continuous integral from t_0 to t_0+t_c:

$$t_c = \sum_{i=1}^{N} \left(T_{1,i} + T_{2,i}\right) \quad \textbf{(7.11)}$$

$$2 \cdot \{N\} \cdot \Delta V_4 = \frac{1}{R_3 \cdot C_1} \cdot \int_{t_0}^{t_0+t_c} V_{in}(t) \cdot dt \tag{7.12}$$

$$f = \frac{\{N\}}{t_c} = \frac{1}{2 \cdot R_3 \cdot C_1 \cdot \Delta V_4} \cdot \frac{1}{t_c} \int_{t_0}^{t_0+t_c} V_{in}(t) \cdot dt \tag{7.13}$$

Output frequency is proportional to the integral average of V_{in}, over the conversion time t_c.

If V_{in} is constant:

$$T = T_1 + T_2 = 2 \cdot T_1 = \frac{2 \cdot R_3 \cdot C_1 \cdot \Delta V_4}{V_{in}};$$
$$f = \frac{\{N\}}{t_c} = \frac{V_{in}}{2 \cdot R_3 \cdot C_1 \cdot \Delta V_4} \tag{7.14}$$

7.2 Experiment

Figure 7.3 shows the experimental board built on schematic in Figure 7.1. The board is designed for direct connection to Analog Discovery, via J1.

Several test points in the schematic can be connected with wires to the scope inputs, 1+ and 2+. The digital signal *Cmp* is hard connected to DIO7 pin of Analog Discovery.

Figure 7.4 shows the scope with the Schmidt trigger signals. Both power supplies are enabled (+5V and -5V). The AWG channel W1 is set with a constant voltage of 2.5V. The scope shows V_4 on channel 1+ and V_+ on channel 2+. In the first step, V_5 was probed with channel 2+, and stored as a reference (*AddCannel/Channel2*). The digital signal *Cmp* is added in the scope time view (DIO7). An XY view shows the hysteresis cycle of the Schmidt trigger. To understand which part in the time diagram corresponds to which segment in the XY view, stop the acquisition

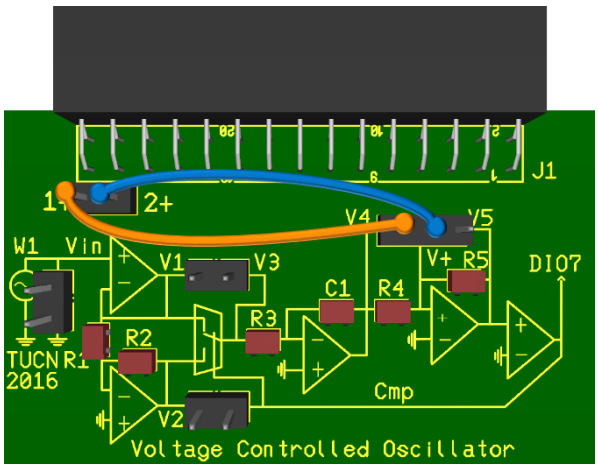


Figure 7.3. VCO experimental board

and then drag the time view image left and right, as in the lower image of Figure 7.4.

Task 1. Based on V_4 and V_5 amplitudes, determine the ratio of R_4/R_5.

Task 2. Use V_5 to determine the Slew Rate of the operational amplifier.
Hint: add two X-cursors and use View X-cursors, Show C2ΔY/ΔX.

Task 3. Use V_3 to determine the switching time of the analog switch.
Hint: add two X-cursors and use View X-cursors, Show C2ΔX.

Task 4. Determine the polarity of V_{in} for correct behavior. Determine the Full Scale values $V_{in,\,FS}$ and f_{FS}.

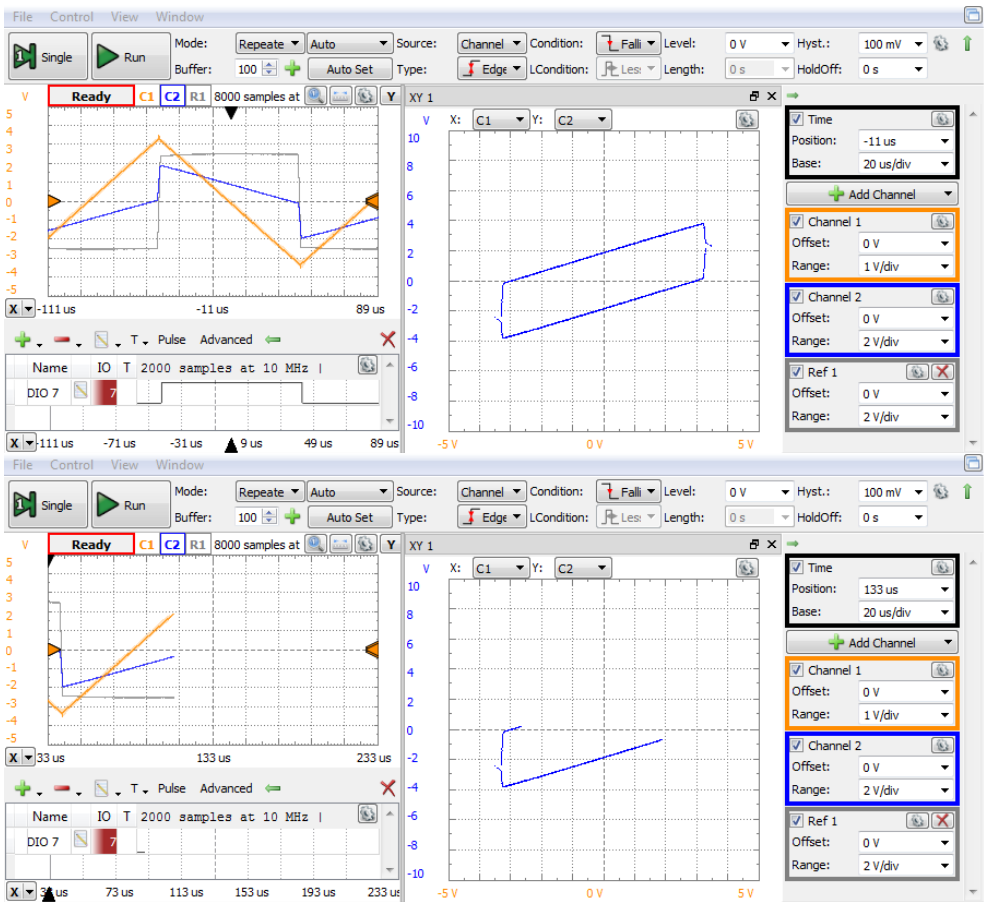


Figure 7.4 Schmidt trigger: full view (UP) and shifted view (down)

Hint: Browse V_{in} and observe the behavior. Use View/Measure/Add/DefinedMeasurement/Channel2/Horizontal/Frequency.

Task 5. Set V_{in} to be a 200Hz sinus from 0 to 5V. Visualize V_{in} on Channel 2, and V_4 on Channel 1 of the scope, as in Figure 7.5. Explain the figure. Explain the V_4 shape for V_{in} close to zero.

Hint: Measure the min and max values of V_{in}. Notice that the calibration of Analog Discovery is not ideal: even if not supposed, V_{in} gets negative values. (if $\boldsymbol{V_{in}}$ never gets negative value on your Analog Discovery, do force that by slightly decreasing the DC component of V_{in}).

Task 6. Play with V_{in} shape, frequency, amplitude, offset and with scope time base, to get nice Lissajoux figures on the XY view.

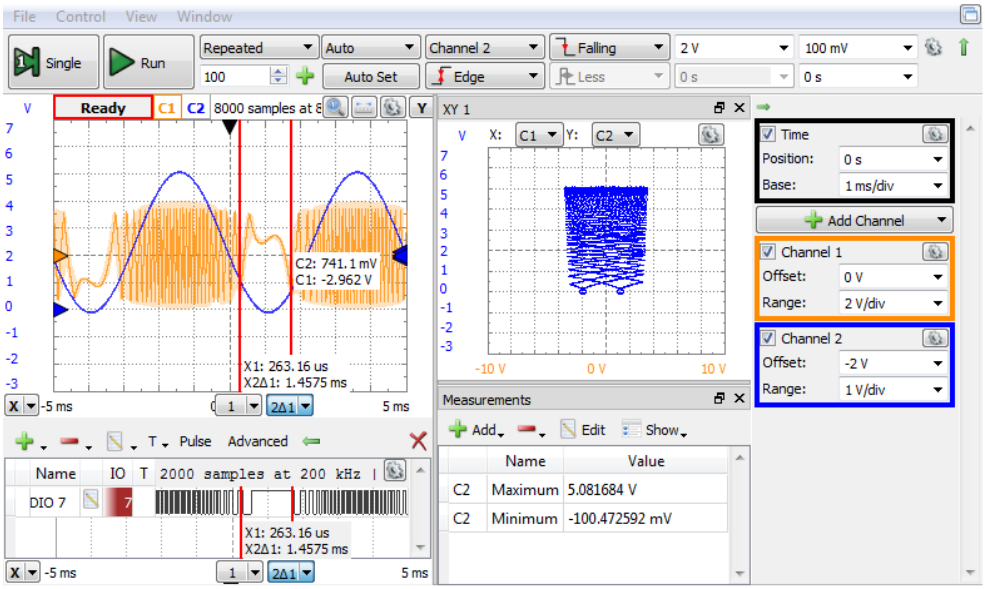


Figure 7.5. VCO with variable input voltage

8 Charge Compensated Modulator

8.1 Background

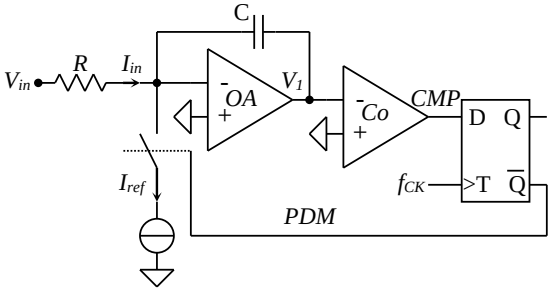


Figure 8.1 Charge compensated modulator

In Figure 8.1, *OA* is an integrator, *Co* compares the V_1 signal to 0V. The *D* flip-flop stores the *CMP* value only at active (rising) clock edges. *PDM* controls the switch, which intermittently connects the I_{ref} current.

Figure 8.2 shows the time diagrams. Required conditions:

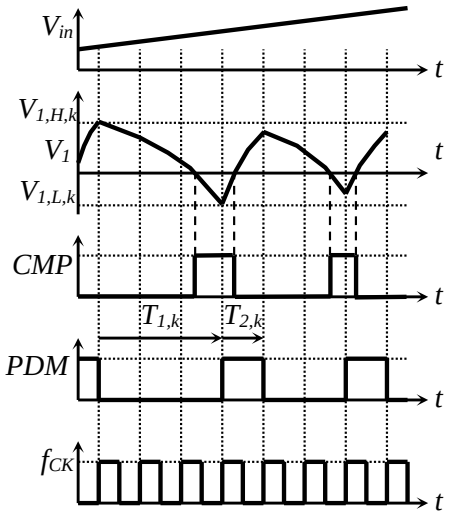


Figure 8.2 Time diagrams

$$V_{in}(t) \geq 0 => I_{in}(t) \geq 0 \quad \textbf{(8.1)}$$

$$I_{ref} = I_{in,FS} = \frac{V_{in,FS}}{R} \geq \frac{V_{in}}{R} = I_{in} \quad \textbf{(8.2)}$$

Phase 1 (*switch = OFF*), begins at time $t_{0,k}$, with $V_1(t_{0,k})=V_{2,H,k}>0$. $V_1(t)$ is falling:

$$V_1(t) = V_{1,H,k} - \frac{1}{C}\int_{t_{0,k}}^{t} I_{in}(t)dt \quad \textbf{(8.3)}$$

When $V_1(t)$ becomes negative, *CMP* becomes *HIGH*, but the flip-flop only changes *PDM* at the next active edge of the clock. Phase 1 ends at time $t_{0,k}+T_{1,k}$, with $V_1(t_{0,k}+T_{1,k})=V_{1,L,k}<0$, where $T_{1,k}$ is an integer multiple of T_{CK}.

Phase 2 (*switch = ON*, $V_1(t)$ rising) begins at $t_{0,k}+T_{1,k}$, with $V_1(t_{0,k}+T_{1,k})=V_{1,L,k}$.:

$$V_1(t) = V_{1,L,k} - \frac{1}{C}\int_{t_{0,k}+T_{1,k}}^{t} \left(I_{in}(t) - I_{ref}\right)dt \quad \textbf{(8.4)}$$

When $V_1(t)$ becomes positive, *CMP* gets *LOW*, but the flip-flop only changes *PDM* at the next active edge of the clock. Phase 2 ends at $t_{0,k}+T_{1,k}+T_{2,k}$ time, with $V_1(t_{0,k}+T_{1,k}+T_{2,k}) = V_{1,H,k+1}>0$, where $T_{2,k}$ is an integer multiple of T_{CK}.

The capacitor accumulates electrical charge as result of two opposite currents:
- the input current I_{in}, which flows continuously and
- the reference current I_{ref}, which flows during phase 2 periods only.

A long measuring time, $t_c >> T_{CK}$ is considered. An *n*-bit conversion takes:

$$t_c = 2^n \cdot T_{CK} \tag{8.5}$$

During t_c, the switch was open for *{N}* clock periods (cumulated Phase 2 periods). At the end of t_c, the integrator output voltage is about the same as the initial voltage, meaning the electrical charge injected by the input current was balanced by the electrical charge extracted by the reference current:

$$All\ phases\ 1\ and\ 2: Q_{in} = \int_{t_0}^{t_0+t_c} I_{in}(t)\cdot dt = \frac{1}{R}\cdot \int_{t_0}^{t_0+t_c} V_{in}(t)\cdot dt \tag{8.6}$$

$$All\ phases\ 2: Q_{ref} = \{N\}\cdot I_{ref}\cdot T_{CK} = \{N\}\cdot \frac{V_{FS}}{R}\cdot \frac{t_c}{2^n} = \{N\}\cdot \frac{V_{LSB}}{R}\cdot t_c \tag{8.7}$$

$$Q_{in} = Q_{ref} \tag{8.8}$$

$$\{N\} = \frac{1}{V_{LSB}}\cdot \frac{1}{t_c}\cdot \int_{t_0}^{t_0+t_c} V_{in}(t)\cdot dt = \frac{V_{in,med}}{V_{LSB}} \tag{8.9}$$

$$\{A\} = \frac{\{N\}}{2^n} = \frac{V_{in,med}}{V_{FS}} \tag{8.10}$$

{N} (n-bit unsigned binary integer) is the cumulated number of clock periods with *PDM = HIGH*, during t_c and is proportional to the mean V_{in} value over t_c. This defines a *DF intermediate quantity integrative ADC. {A} (n-bit unsigned binary fractional)* is the normalized representation of *{N}*. Equation (8.5) shows the conversion time for such an n-bit resolution ADC.

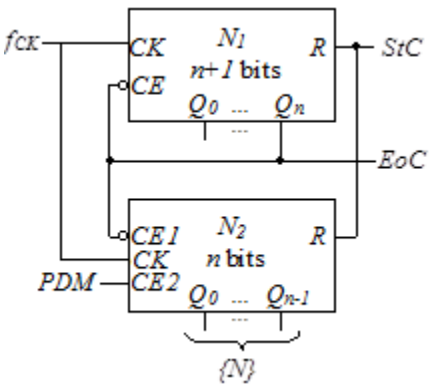


Figure 8.3. DF measuring

The circuit in Figure 8.3 measures *{N}*.

A *StC (Start Conversion)* pulse resets both counters. The Q_n of N_1 enables both counters, for $t_c = 2^n \cdot T_{ck}$. N_2 only counts the clock pulses for which *PDM = HIGH*.

After t_c, both counters freeze: N_1 has Q_n = *HIGH (End of Conversion)* and all other bits *LOW*, N_2 has the value of $\{N\}$, as defined in (8.9).

Variations of the schematic in Figure 8.1 can be used in sigma-delta ADCs. In this case, f_{ck} is the oversampling frequency, which is decimated within a digital filtering algorithm down to the final sampling frequency.

The average duty factor of the PDM signal is:

$$DF_{med} = \frac{\{N\} \cdot T_{CK}}{t_c} = \frac{\{N\} \cdot T_{CK}}{2^n \cdot T_{CK}} = \{A\} = \frac{V_{in,med}}{V_{FS}} \qquad \textbf{(8.11)}$$

Defining DF as the instantaneous probability of the PDM signal to be HIGH, equation (8.11) can be re-written for instantaneous values of DF and V_{in} defining the same circuit as a PDM modulator:

$$DF = \frac{V_{in}}{V_{FS}} \qquad \textbf{(8.12)}$$

In this approach, V_{in} is the modulator signal, *PDM* is the *DF modulated signal*, and the carrier is a spread spectrum rectangular signal, with a frequency range from *0Hz* to $f_{CK}/2$*:*

$$V_{in} = 0 => \ DF = 0 => \ f_{PDM} = 0 \qquad \textbf{(8.13)}$$

$$V_{in} = V_{FS} => \ DF = 1 => \ f_{PDM} = 0 \qquad \textbf{(8.14)}$$

$$V_{in} = 0.5 \cdot V_{FS} => \ DF = 0.5 => \ f_{PDM} = 0.5 \cdot f_{CK} \qquad \textbf{(8.15)}$$

$$V_{in} \in (0 \dots V_{FS}) => f_{PDM} \in (0 \dots 0.5 \cdot f_{CK}) \qquad \textbf{(8.16)}$$

The simplest demodulator is a Low Pass Filter. An ideal LPF should remove the carrier frequencies and recover the unaffected modulator. A real filter attenuates the carrier frequencies and keeps the modulator almost unaffected. The lowest carrier frequency must be (much) higher than the modulator frequency, which implies limiting V_{in} to less than in (8.16).

$$V_{in} = V_{FS}/k => \ DF = 1/k => \ f_{PDM} = f_{CK}/k \qquad \textbf{(8.17)}$$

$$V_{in} = (1 - 1/k) \cdot V_{FS} => DF = (1 - 1/k) => f_{PDM} = f_{CK}/k \qquad \textbf{(8.18)}$$

$$V_{in} \in (V_{FS}/k \dots (1 - 1/k) \cdot V_{FS}) => f_{PDM} \in (f_{CK}/k \dots 0.5 f_{CK}) \qquad \textbf{(8.19)}$$

$$f_{PDM,min} = f_{CK}/k \gg f_{in} \qquad \textbf{(8.20)}$$

8.2 Simulation

Figure 8.4 shows the Multisim schematic for simulation. The signal generator W1, the clock generator ck, and the D flip-flop are simulation-only components. Both CMP and PDM are 5V logic signals. All the other components and connections are realized on the Charge Compensated Modulator experimental board.
When the analog switch ADG601 is ON, it connects the resistor R_3 to – 5V, which acts as a current supply of:

$$I_{ref} = \frac{0 - VEE}{R3} = \frac{5V}{1k\Omega} = 5mA \qquad \textbf{(8.21)}$$

The full-scale voltage of the modulator is:

$$V_{in,FS} = I_{ref} \cdot R1 = 5mA \cdot 1k\Omega = 5V \qquad \textbf{(8.22)}$$

R_2 and C_2 build a simple Low Pass filter which re-constructs a copy of V_{in}, based on the *PDM* signal.

Figure 8.5 shows the transient simulation results for the input signal, V_{in}, set at 90%, 50% and 10% of the V_{FS}. The *ck* frequency is f_{ck}=*200kHz*.

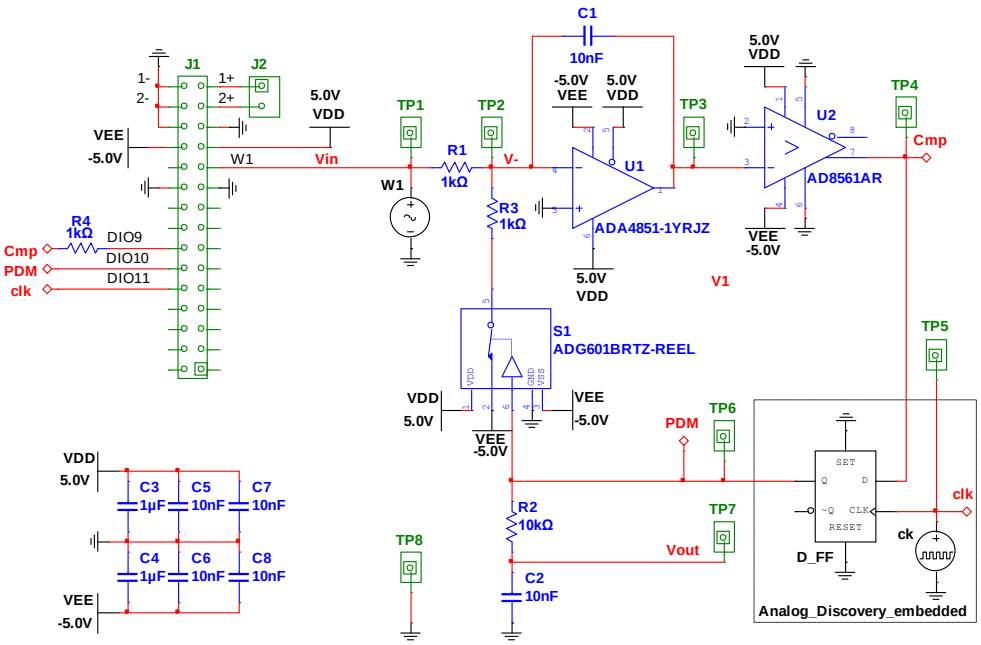


Figure 8.4. Charge Compensated Modulator schematic

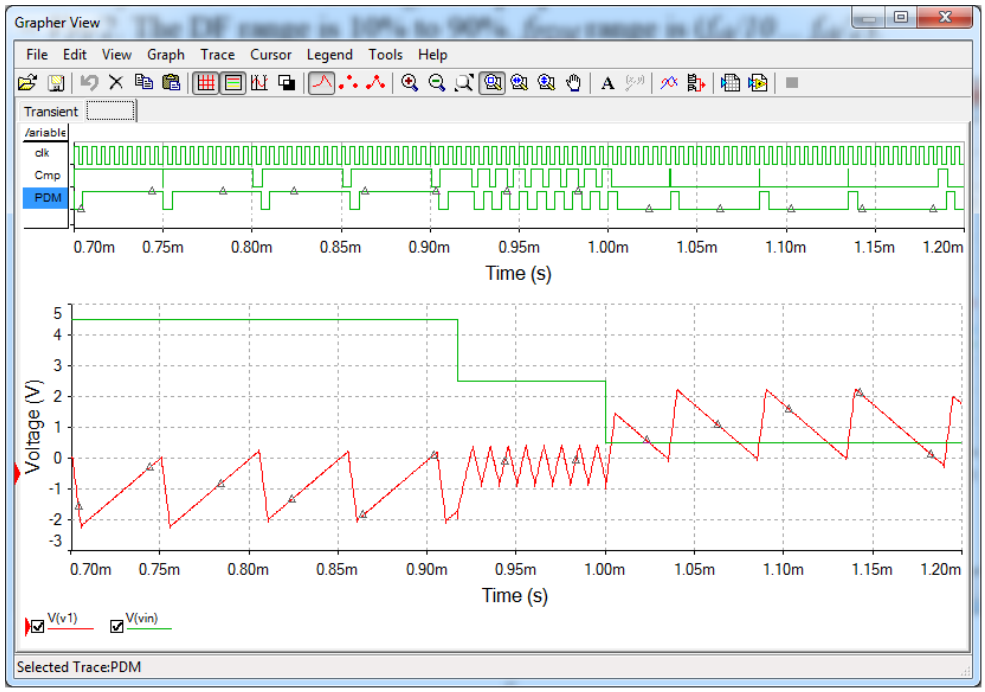


Figure 8.5. Charge Compensated Modulator simulation – constant V_{in}

V1 falls slower and rises faster when *Vin* is higher, reverse when *Vin* is closer to zero, and has symmetrical slopes when $V_{in} = V_{FS}/2$. CMP is the asynchronous output of the comparator, while PDM is the synchronized version of it; both LOW (phase 1) and HIGH (phase 2) pulses take integer number of clock periods. DF is equal to the V_{in}/V_{FS} ratio, as shown by (8.12). In Figure 8.6, the input signal, V_{in}, is a $f_{in}=2kHz$ sinus, with a DC component of $2.5V = V_{FS}/2$ and $2V$ amplitude; V_{in} range is 10% to 90% of the V_{FS}.

$$V_{in} = 2.5V + 2V \cdot \sin(2\pi f_{in} t) \qquad \textbf{(8.23)}$$

The duty factor of the PDM signal is proportional to V_{in}: DF = 50% when $V_{in} = V_{FS}/2$. The DF range is 10% to 90%. f_{PDM} range is $(f_{ck}/10 \ldots f_{ck}/2)$.

V_{out} is the re-construction of the V_{in}, using the output filter R_2C_2. An ideal output filter would completely eliminate the carrier frequencies and not affect the demodulated signal. The ideal range of V_{out} results from the PDM amplitude (5V) and PDM DF range:

$$V_{out,ideal} \in (10\% \cdot 5V \ldots 90\% \cdot 5V) = (0.5V \ldots 4.5V) \qquad \textbf{(8.24)}$$

$$V_{out,ideal} = 2.5V + 2V \cdot \sin(2\pi f_{in} t) \qquad \textbf{(8.25)}$$

R_2C_2 is a first order low-pass filter, with the corner frequency below f.

$$f_{corner} = \frac{1}{2\pi R_2 C_2} = 1.59kHz \qquad \textbf{(8.26)}$$

The DC component is not affected, the AC amplitude is attenuated, and the carrier frequencies are incompletely removed:

$$V_{out,ampl} = \frac{V_{PDM,ampl}}{|1 + 2j\pi f_{in} R_2 C_2|} = \frac{2V}{1.6} = 1.24V \qquad \textbf{(8.27)}$$

$$V_{out,real} \approx 2.5V + 1.24V \cdot sin\,(2\pi f_{in} t) \qquad \textbf{(8.28)}$$

The ratio between the $f_{PDM,min}$ and f_{in} frequency is far too low: it allowed waveforms visualization for a detailed analysis of the circuit behavior, but resulted in a low quality re-constructed signal.

For a high quality PDM modulation which can serve for better re-construction of V_{in}, a much higher ratio should be used: either higher $f_{PDM,min}$ or lower f_{in} or both. Higher $f_{PDM,min}$ means either higher f_{ck}, lower k (lower fraction of V_{FS} used), or both in (8.19) and (8.20). The re-construction filter should be higher order and have the f_{corner} between $f_{in,max}$ and $f_{PDM,min}$.

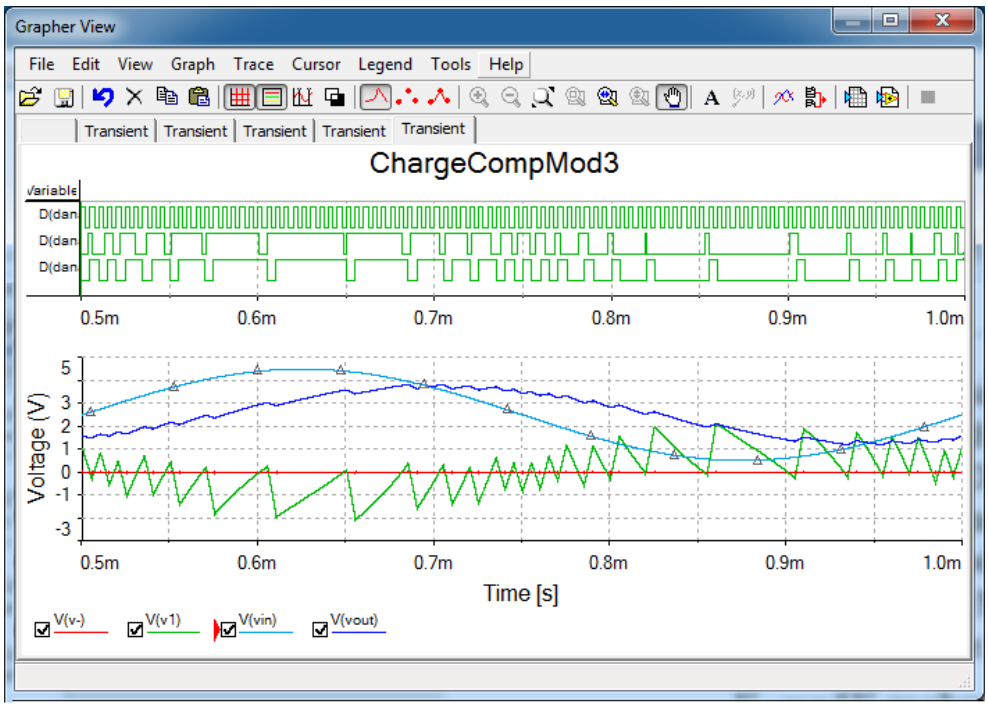


Figure 8.6. Charge Compensated Modulator simulation – sinusoidal V_{in}

8.3 Experiment

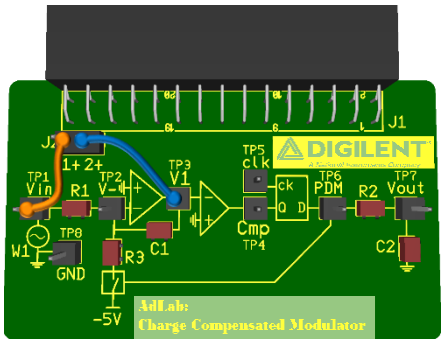


Figure 8.7 Charge Compensated Modulator experimental PCB

Figure 5.4Figure 8.7 shows the experimental board. J2 header provides access to the Analog Discovery scope inputs 1+ and 2+. The negative nodes of the differential scope inputs (1- and 2-) are hard connected to GND.

TP0…TP7 headers are schematic test points. Use wires to connect these nodes to the scope inputs.

The wire setup shown in Figure 8.7 probes V_{in} and $V_.$

In Figure 8.9, the WaweForms instruments are set as below:

- Power Supplies: ON, +5V and -5V.
- WaveGen: *constant (DC) = 4.5V*.
- Patterns Generator:
  - *DIO11: clock, 200kHz, Duty = 50%, Phase = 0*.
  - *DIO10: ROM logic, 200kHz, DIO10 <= DIO9* (see Figure 8.8).

This configuration synchronizes *DIO9* (*Cmp*) by clock *clk*, to get *PDM* (*DIO10*). It simulates a D Flip-Flop with the falling edge active clock *DIO11*.

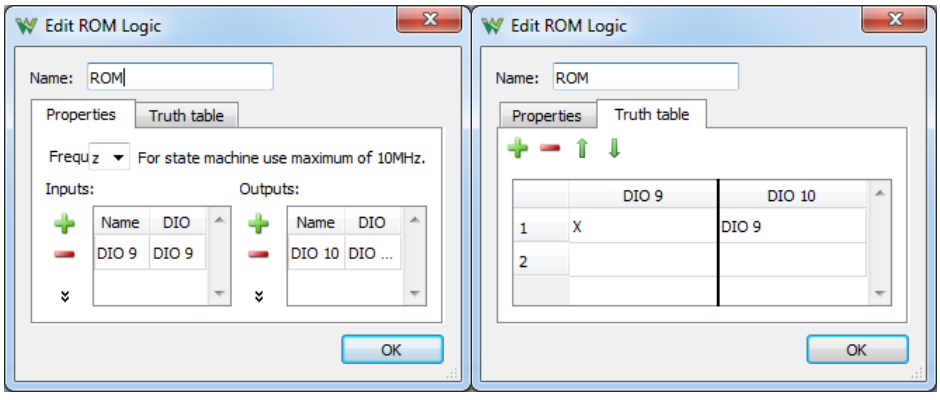


Figure 8.8 ROM logic for PDM

The scope is set as shown in Figure 8.9. Ref 1 shows V_{in}, Channel 1 shows V_1, Channel 2 shows V_{out}. A noise band around scope signals suggests that an oscillation happens in the circuit; the input capacitance of the scope probe is

too large for the OA, which becomes unstable. To avoid oscillation, a series 100Ω resistor was added in the tip of the V_1 scope probe, which insulated the scope capacitance and removed the oscillation.

Three digital signals are added in the scope window: *DIO9=Cmp*, *DIO10=PDM* and *DIO11=clk*. Although *DIO11* is not actually the clock of the ROM logic D flip-flop, it has the same frequency and phase with it, so it is shown as reference. Notice that the ROM logic synchronizes to the end of each clock period, which is the falling edge of *DIO11*. This is a difference compared to the Multisim simulation, where the D Flip-Flop is active on the rising clock edge.

$V_{in,FS}=5V$, as shown in (8.22). The Analog Discovery DIO signals are CMOS 3V3 compatible, so $V_{out,FS}=3.3V$ (the V_{out} value when DF = 1).

In Figure 8.9, $V_{in}=4.5V=90\%\cdot V_{in,FS}$; the rising slopes of V_1 are 9 time slower than the falling ones, the *PDM DF=90%*, $f_{PDM}=f_{ck}/10$ and $V_{out}=2.92V\approx 0.9\cdot V_{out,FS}$. In Figure 8.11, $V_{in}=0.5V=10\%\cdot V_{in,FS}$; the V_1 slopes are reversed, the *PDM DF=10%*, $f_{PDM}=f_{ck}/10$ and $V_{out}=0.3V\approx 0.1\cdot V_{out,FS}$.

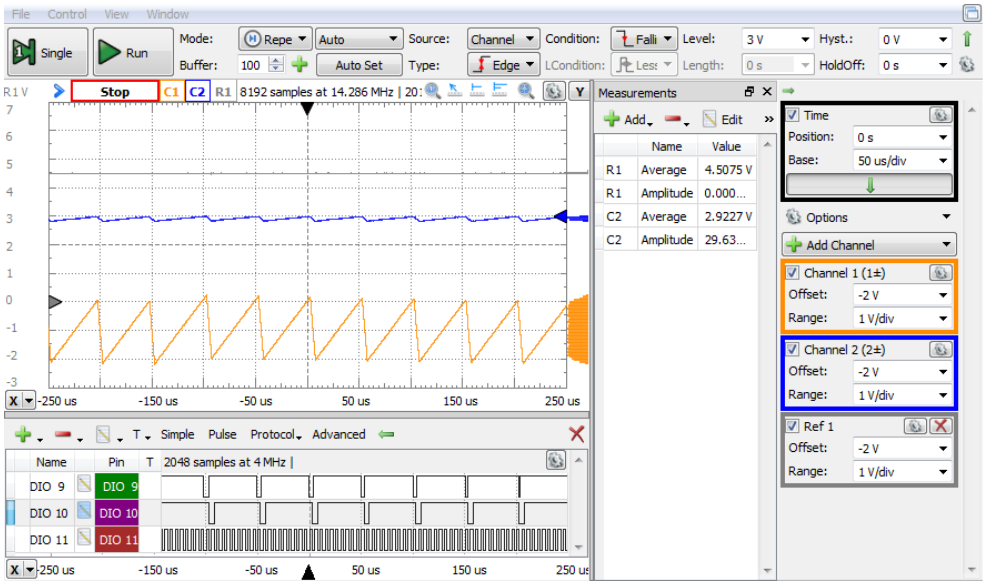


Figure 8.9 Charge Compensated Modulator experiment–$V_{in}=0.9\cdot V_{in,FS}$

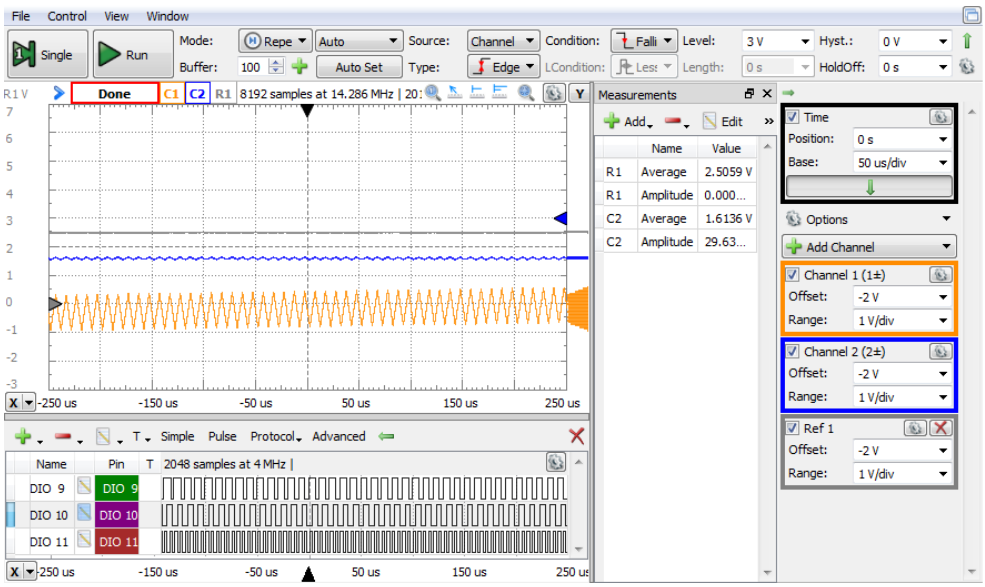


Figure 8.10 Charge Compensated Modulator experiment–$V_{in}=0.5\cdot V_{in,FS}$

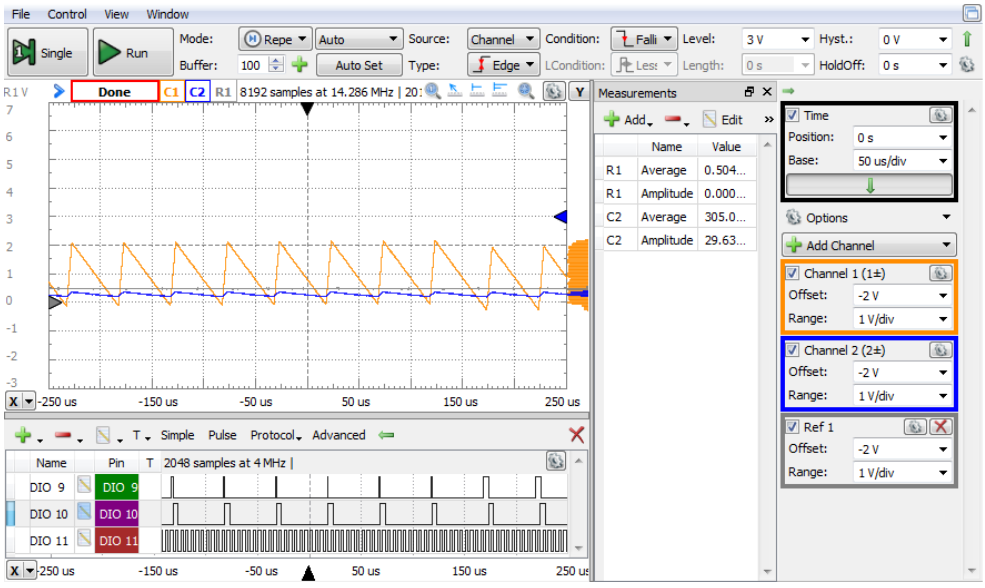


Figure 8.11. Charge Compensated Modulator experiment–$V_{in}=0.1\cdot V_{in,FS}$

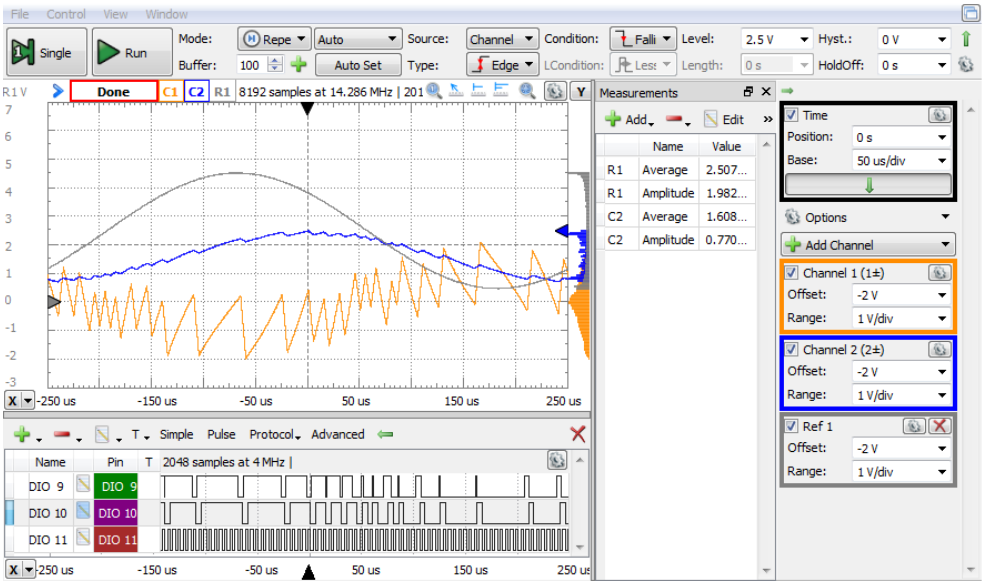


Figure 8.12 Charge Compensated Modulator experiment– sinusoidal V_{in}

In Figure 8.10, V_{in} = 2.5V = 50%·$V_{in,FS}$; the V_1 slopes are symmetrical, the PDM DF = 50%, $f_{PDM} = f_{ck}/2$ and $V_{out}=1.61V\approx 0.5\cdot V_{out,FS}$.

In Figure 8.12, V_{in} = 2.5V+2V·$sin(2\pi f_{in}t)$, with $f_{in}=2kHz$; the V_1 rising slopes are slower when V_{in} is higher, faster when V_{in} is lower. The falling slopes are reversed. The PDM DF changes between *10%* and *90%*, f_{PDM} changes between $f_{ck}/10$ and $f_{ck}/2$.

Equations (8.23)…(8.28) can be written with $V_{out,FS}=3.3V$:

$$V_{in} = 2.5V + 2V \cdot \sin(2\pi f_{in} t) \quad \textbf{(8.29)}$$

$$V_{out,ideal} \in (10\% \cdot 3.3V \ldots 90\% \cdot 3.3V) = (0.33V \ldots 2.97V) \quad \textbf{(8.30)}$$

$$V_{out,ideal} = 1.65V + 1.32V \cdot \sin(2\pi f_{in} t) \quad \textbf{(8.31)}$$

$$f_{corner} = \frac{1}{2\pi R_2 C_2} = 1.59kHz \quad \textbf{(8.32)}$$

$$V_{out,ampl} = \frac{V_{PDM,ampl}}{|1 + 2j\pi f_{in} R_2 C_2|} = \frac{1.32V}{1.6} = 0.822V \quad \textbf{(8.33)}$$

$$V_{out,real} \approx 1.65V + 0.822V \cdot sin(2\pi f_{in} t) \quad \textbf{(8.34)}$$

Task 1. Increase the AWG signal amplitude to cover more of the definition range. Explain the wave shape V_1. Export to your report document the scope

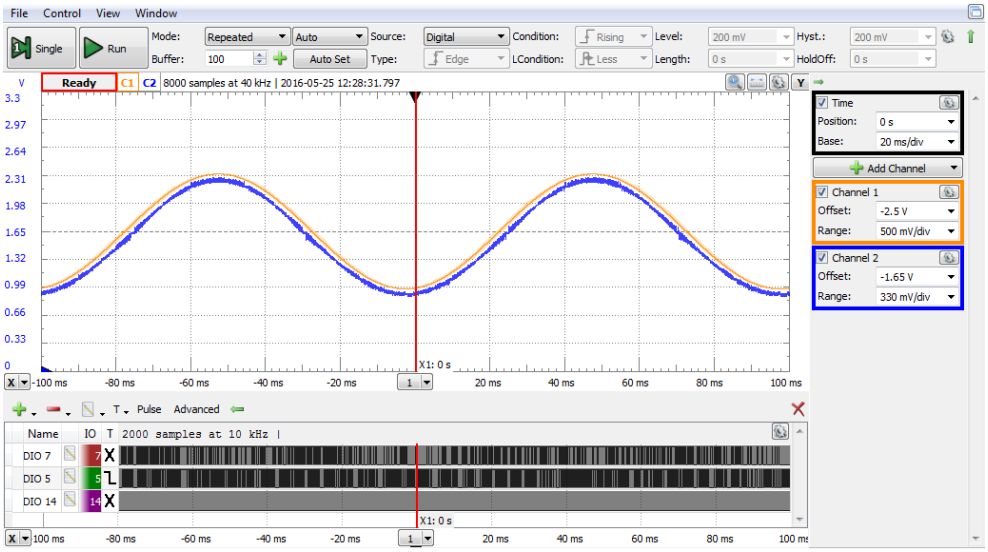


Figure 8.13 Signal reconstruction

image for V_{in} covering the whole definition range.

As in the simulation, the ratio of the clock frequency to the V_{in} frequency is far too low to allow high quality V_{out} re-build. This ratio allows visualizing the signals and understanding the basics of the circuit. In a real application, the clock (oversampling) frequency is chosen to be much higher than the highest spectral component of V_{in}.

A better approximation is shown in Figure 8.13. Here, the clock frequency is *200kHz*, and the V_{in} frequency is *10Hz*.

However, if the clock period becomes comparable to the switching time for I_{ref}, this generates linearity errors, and the re-built signal is distorted, as shown below.

8.4 Design considerations

8.4.1 V_{in} dynamic range

V_1 swing range needs to be within the OA output range, which depends on the supply voltages and rail margins of the OA.

Lowest negative value of V_1 happens during phase 1, when:

- V_1 becomes negative immediately after an active edge of *ck* (see $V(V_1)$ in Figure 8.5, the slope beginning at time 750µs), and
- $V_{in} = V_{in,FS}$;

From (8.3):

$$V_{1,\min} = -\frac{V_{in,FS}}{R \cdot C} \cdot T_{ck} \qquad \textbf{(8.35)}$$

Similarly, highest positive value of V_1 happens during phase 2, when:

- V_1 becomes positive immediately after an active edge of *ck* (see $V(V_1)$ in Figure 8.5, the slope beginning at time ~1.08ms), and
- $V_{in} = 0$;

From (8.4):

$$V_{1,\max} = \frac{I_{ref}}{C} \cdot T_{ck} = \frac{V_{in,FS}}{R \cdot C} \cdot T_{ck} \qquad \textbf{(8.36)}$$

Equations (8.35) and (8.36) need to be considered when designing *R, C,* T_{ck}, in conjunction with the OA output voltage range.

8.4.2 Clock frequency

For high bandwidth, short t_c and high resolution, T_{ck} should be as short as possible. However, if T_{ck} becomes comparable to the switching time of I_{ref}, the linearity of the ADC is affected.

Figure 8.14 to Figure 8.16 use triangular V_{in} shape, for linearity estimation. The clock frequency is f_{ck}*=100kHz* in Figure 8.14, f_{ck}*=1MHz* in Figure 8.15, f_{ck}*=10MHz* in Figure 8.16. The scope vertical scaling and vertical position are set to overlap V_{out} to V_{in} (same rate as $V_{in,FS}/V_{out,FS}$*=5V/3.3V*).
When the clock frequency is low, the V_{out} LPF cannot effectively remove the carrier frequencies, visible in V_{out}.

At medium clock frequency, the LPF works much better except at extreme values of V_{in} (*0* and V_{FS}), where the modulator frequency is low. A small linearity error is visible for V_{in} around midrange: the I_{ref} switching frequency is highest there ($f_{ck}/2$) and the switching period becomes comparable with the switching time of the *S1* switch. The integrated value of I_{ref} is affected.

At high clock frequencies, the LPF works even better, but the linearity error increases.

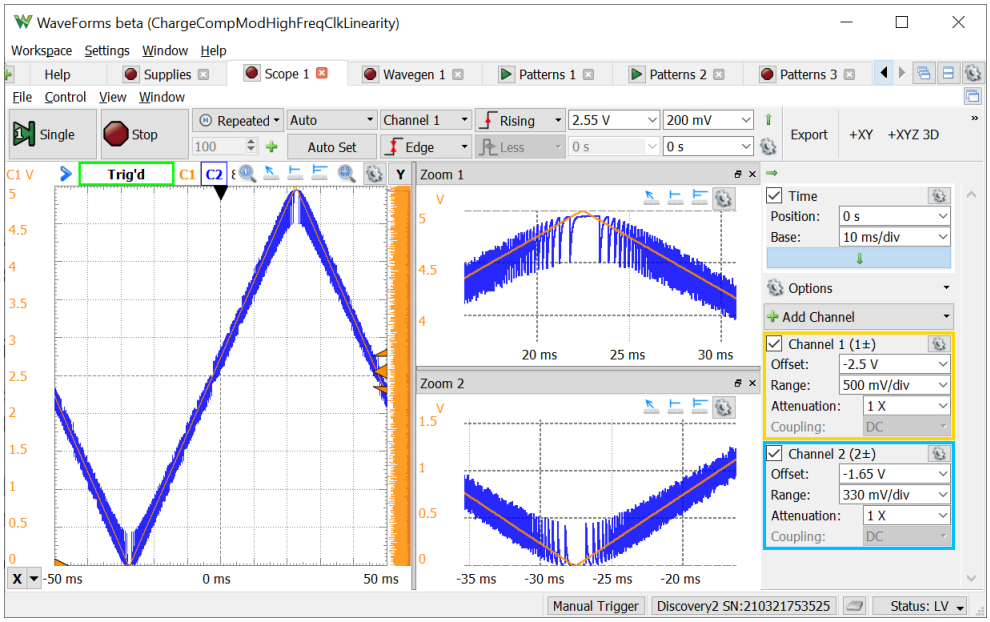


Figure 8.14 *10Hz, 5*V_{vv} ($V_{in,FS}$) triangular V_{in} $\boldsymbol{f_{ck}}$***=100kHz***

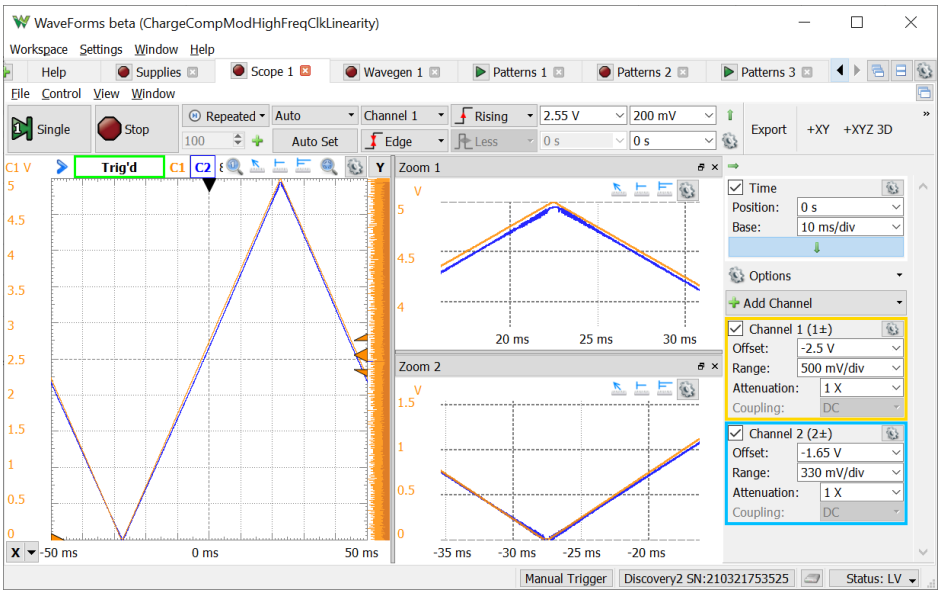


Figure 8.16 *10Hz, 5*V_{vv} ($V_{in,FS}$) triangular V_{in} f_{ck}***=1MHz***

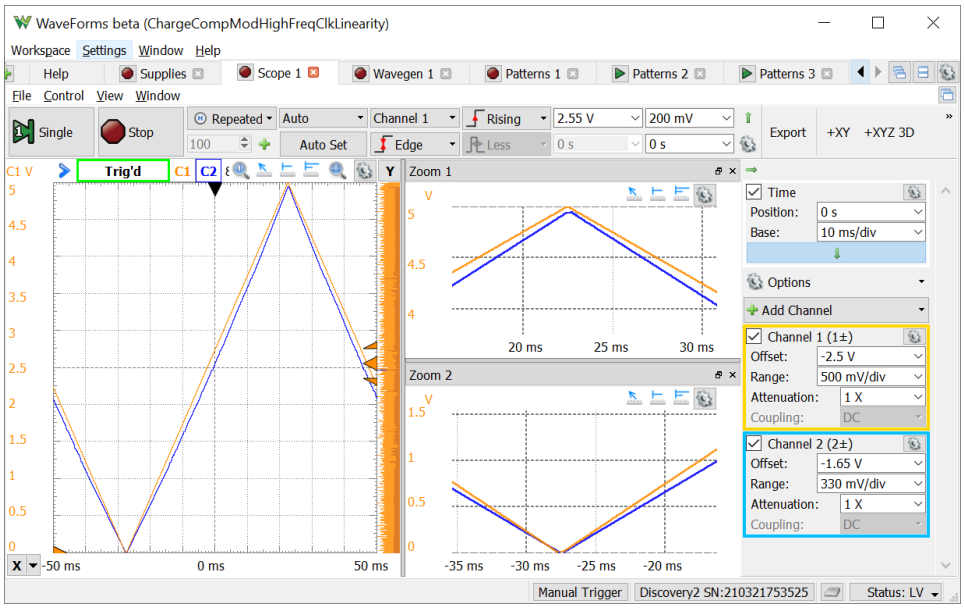


Figure 8.15 *10Hz, 5*V_{vv} ($V_{in,FS}$) triangular V_{in} f_{ck}***=10MHz***