Analog and Mixed Integrated Systems

December 6th 2022

Initial Target: High level project of an ADC

Bologna Master Degree in Electrical and Computer Engineering (MEEC)

Group 2: Diogo Matos Ferreira (96189) | Duarte Miguel de Aguiar Pinto e Morais Marques (96523)

The main objectives of this laboratory assignment include introducing the environment of Cadence DF II (version 6) with ADE L and ADE XL, understand ADCs (analog-to-digital converters) test setups using FFT (Fast Fourier Transform), understand variables and specifications in ADCs and DACs (digital-to-analog converters) and compare them with an ideal Sample and Hold.

Initially, a tutorial was followed in order to install and setup the Cadence environment. Moreover, an already designed ADC present in the cell `test_sar4b_SIA` was included in the library `eesar_SIA`. Inside this library, different cells contain the files necessary to test the SAR (Sucessive Approximation) ADC with an output of 4 bits. The schematic included in `test_sar4b_SIA` contains three different parts - a signal generator, which creates the sinusoidal input signal, the 1MHz squared wave clock and the DC voltages $\mathrm{V_{dd}}$, $\mathrm{V_{ss}}$ and $\mathrm{V_{cm}}$; the ideal SAR ADC sampling the input signal into a 4-bit word; an ideal DAC which generates an output representation of the sampled signal.

Initial results obtained with input frequency $\mathrm{F_i} = 33.203125\mathrm{kHz}$

As described in the tutorial, the aforementioned schematic can be used to run the transient simulation and obtain the graphs shown in Figure 1. As seen here, the input signal suffers a linear transformation, having a DC offset been added to it, followed by a division by 2. This is done so that the resulting wave is strictly positive. However, the general form of the wave remains the same, thus making possible the full recovery of the initial bipolar signal with no added complexity to the system.

This modulated signal is then passed through the ideal **SAR ADC** (a fairly complex system that will not be analyzed in this report). The main point to notice on this device is the output signal, the 4-bit word representing the voltage of the input signal. These bits can be seen as squared waves representing the logic values 0 or 1. The binary value of 1 corresponds to the voltage of 3.3V in this signal, whereas the binary value of 0 corresponds to 0V. From a digital point of view, there may be a range of voltages that represent each binary value - with margin for losses and noise error -, but from an ideal analog point of view, those eventual characteristics are irrelevant.

The lowest bit (bit 1) corresponds to the lowest significant bit (LSB) and provides the voltage resolution, which can be obtained with this word representation. Considering the modulated signal (red wave in Figure 1), with an amplitude of $2 \times [3.1 + 3.3 - (-3.1 + 3.3)]/2 = 3.1\mathrm{V}$, a theoretical value of $3.1/2^4 = 0.19375\mathrm{V}$ corresponding to LSB is obtained. However, by analysing the output signal from the ADC and the modulated input, it is visible that the commutation happens on average around $0.344\mathrm{V}$. Taking into account the non-modulated signal, a theoretical LSB correspondence to $6.2/2^4 = 0.3875\mathrm{V}$ is obtained. As for the most significant bit (MSB), in theory, its corresponding value should be $3.1/2 = 1.55\mathrm{V}$ for the modulated signal. By checking the output, it can be concluded that an average value for the commutation of $1.96\mathrm{V}$ is obtained.

One aspect that is clearly visible in Figure 1 is the switching frequency difference amongst the different bits. The lowest valued bit has almost double the number of changes than the second to lowest valued bit. This fact repeats through the remaining bits and is coherent with the binary logic of a normal 0 and 1 counting code system.

Something visible and worth noting is the apparent non-sequential and repetitive nature of the 4-bit word. It has to do with the "mismatch" between the binary voltage's code values and the changing value at the input with the used sampling frequency. In fact, this jumping happens much more frequently around the point with the highest derivative in the sinusoidal wave, whereas the repetitive codes appear at the highest and lowest values of the wave, where the derivative reaches 0. Only with, for instance, a triangular wave with a constant derivative equal to that of $\mathrm{LSB} \times \mathrm{F_s}$, would a sequential and non-repetitive 4-bit word be obtained.

As stated before, one of the components used in this testing is a DAC, which takes as input the 4-bit word from the ADC and outputs a virtual and ideal representation of the voltage it represents. The working mode is fairly simple, as it only sums the products of the bits by their ideal voltage values of $[1/16 \times 3.3,\ 2/16 \times 3.3,\ 4/16 \times 3.3,\ 8/16 \times 3.3]$, respectively.

In Figure 1, it is also possible to easily compare the modulated input signal with the output wave. They are very similar, but the distinction between continuous input and discretization of the output is visible. There is, in fact, the introduction of errors along the signal due to the approximation of certain input voltages to their closest sampled values. The biggest errors (i.e., differences between the modulated signal and the output of the DAC) occur at the values right before commutation, having an average difference of about 0.41V been obtained for these points, with a maximum of about 0.58V around the points in the sinusoid with the highest derivatives. Furthermore, the curves are separated by what seems to be a delay of around $0.86\mu\mathrm{s}$. Because of this, the previously mentioned errors are not entirely representative of the respective bits. They do not represent the actual errors committed by the ADC, due to the time shift in the output signal regarding the input. For different frequencies, different results would be obtained, considering a constant delay. In order to neglect this delay effect and get the intrinsic errors for each bit, a DC analysis would have to be conducted.

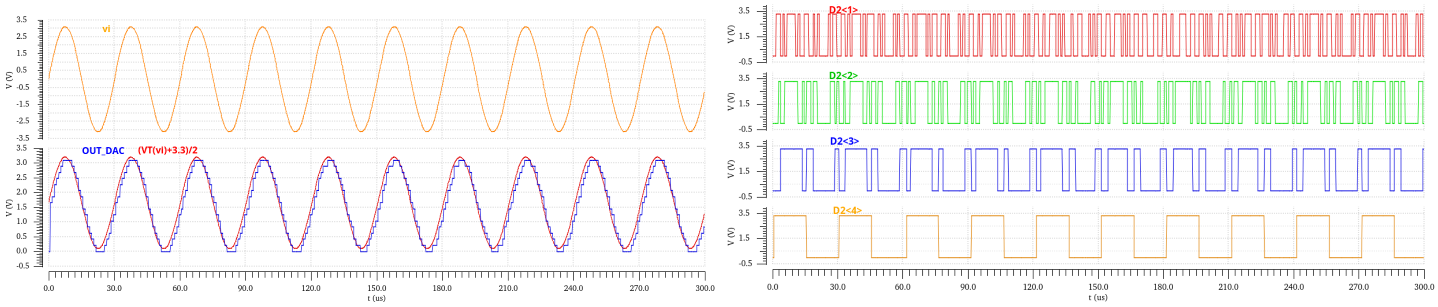

Figure 1: Signals obtained by running a transient analysis (liberal, stop time $600\mu\mathrm{s}$) with the schematic in `test_sar4b_SIA`, using a sampling frequency of $\mathrm{F_s} = 1\mathrm{MHz}$. On the left, input `vi` (amplitude 3.1V, frequency 33.203125kHz), output signal `OUT_DAC` and wave obtained from `((VT("/vi") + 3.3) / 2)`; on the right, inputs of the adder (`adder_4` from `ahdlLib`). The graphs are shown here in the initial $300\mu\mathrm{s}$ for convenience.

1) Explaining used setup values (input and sampling frequencies)

Coherent sampling for FFT is given by the relation shown in (1), where $\mathrm{f_{in}}$ is the input frequency, $\mathrm{f_s}$ is the sampling frequency, N is a power of two and M is a prime integer:

$$\frac{f_{in}}{M} = \frac{f_s}{N} \tag{1}$$

A **Fast Fourier Transform (FFT)** can be performed to obtain a frequency spectrum that represents the signal. Bearing in mind that the input signal is a perfect sinusoidal wave, the expected FFT should be a continuous spectrum with null magnitude in every frequency, apart from the frequency of the signal - where an isolated spike with a much higher magnitude should be observed. Even though there will be noise throughout the entire spectrum, a more significant isolated spike should still remains. Since time discrete signals are being used in this assignment, the corespondent discrete analyses should be performed - the **Discrete Fourier Transform (DFT)**. In order to do that, some analyses parameters must be selected; in this case, the following values were selected:

- Initial time: $6\mu\mathrm{s}$
- Final time: $518\mu\mathrm{s}$
- Number of samples: 512
- Window type: "Rectangular", "Hamming" or "Blackman-Harris"
- Smoothing factor: 1
- Coherent gain: default
- ADC span: 1

Some immediate conclusions can be directly obtained from the parameters shown above. Firstly, since the time interval is $512\mu\mathrm{s}$ and the number of samples is 512, the time resolution is $1\mu s$; the simulations are likely being considered from $\mathrm{t} = 6\mu\mathrm{s}$ onwards in order to avoid initial transient errors. Moreover, since the sampling frequency is 1MHz, the Nyquist theorem states that it is possible to fully sample frequencies up to $\mathrm{F_s/2 = 500kHz}$and have a resolution of $\mathrm{f_s/N = 1MHz/512 = 1.953125kHz \equiv \Delta f}$. Taking into account this frequency resolution, the use of an input frequency of $\mathrm{F_i = 33.203125kHz}$ can be understood, as it corresponds to $\Delta\mathrm{f} \times 17$. The 17th sample in frequency will therefore be able to fully sample the input frequency, thus achieving no spread spectrum.

2) Rectangular, Hamming and Blackman-Harris windows with different input frequencies

In these simulations, the values of the Signal-to-noise ratio (**SNR**), Signal-to-noise and Distortion (**SINAD**), Spurious Free Dynamic Range (**SFDR**) and Effective Number of Bits (**ENOB**), given by the equations below - where N is the number of bits - will be obtained.

$$\mathrm{SNR[dB] = 6.02N + 1.76}\ , \quad \mathrm{SINAD[dB]} = 10\log\left(\frac{\mathrm{A^2_{bin(f_{in})}}}{\sum_{\mathrm{n=2}}^{\mathrm{Size/2}} \mathrm{A_n^2} - \mathrm{A^2_{bin(f_{in})}}}\right)$$

$$\mathrm{SFDR[dB]} = 10\log\left(\frac{\mathrm{A^2_{bin(f_{in})}}}{\mathrm{A^2_{bin\_max(\neq f_{in})}}}\right), \quad \mathrm{ENOB} = \frac{\mathrm{SINAD} - 1.76}{6.02}$$

Considering an effective number of bits of ENOB=4, a corresponding signal-to-noise-ratio of $\mathrm{SNR[dB]} = 6.02 \times 4 + 1.76 = 25.84$ is obtained. These are the theoretical maximum values that this such device should be able to achieve.

Similar simulations were then performed for an input frequency of $\mathrm{F_i = 28kHz}$, which is no longer a multiple of the frequency resolution $\Delta\mathrm{f}$, since $\mathrm{28kHz}/\Delta\mathrm{f} = 14.336$. This means that the frequency of the sinusoidal wave will not be represented by a single frequency value, thus spreading throughout the spectrum; this will have obvious impacts in the SINAD and ENOB values, as seen in Tables 1 and 2. The values of these parameters are not as easily calculated analytically, but smaller values were to be expected.

		ENOB		
Sampling frequency, $\mathrm{F_s}$ [MHz]	Input frequency, $\mathrm{F_i}$ [kHz]	Rectangular	Hamming	Blackman-Harris
1	33.203125	3.981	3.999	4.046
	28	0.271	3.455	3.972

Table 1: Effective Number of Bits (ENOB) obtained by using the schematic included in the cell `test_sar4b_SIA` presented in the tutorial. For this purpose, a transient analysis (liberal, stop time $600\mu s$) was run, in which the sampling frequency in the instance `vpulse` and the input frequency in `vsin` were defined as shown above, whereas ENOB was obtained from `spectrumMeas(VT("/OUT_DAC") 6e-06 0.000518 512 0 0 nil "Rectangular" 0 "enob")` for the rectangular window (using `"Hamming"` or `"Blackman"` for the other two, respectively).

		SINAD and SNR [dB]			**SFDR [dB]**		
Sampling frequency, $\mathrm{F_s}$ [MHz]	Input frequency, $\mathrm{F_i}$ [kHz]	Rectangular	Hamming	Blackman-Harris	Rectangular	Hamming	Blackman-Harris
1	33.203125	25.73	25.83	26.12	35.56	35.56	35.60
	28	3.391	22.56	25.67	6.169	24.17	34.12

Table 2: SINAD, SNR and SFDR values obtained by using the schematic included in the cell `test_sar4b_SIA` presented in the tutorial. For this purpose, a transient analysis (liberal, stop time $600\mu s$) was run, in which the sampling frequency in the instance `vpulse` and the input frequency in `vsin` were defined as shown above. The results were obtained from `spectrumMeas(VT("/OUT_DAC") 6e-06 0.000518 512 0 0 nil "[window type]" 0 "[parameter]")`, where `[parameter]` is replaced by `sinad`, `snhr` or `sfdr(db)` (respectively) and `[window type]` is replaced by `Rectangular`, `Hamming` or `Blackman`.

In Figure 2, the DFT for both input frequencies can be seen for different **window types**. The concept of a window has to do with the consideration of certain parts of the signal as being either part of the actual signal to be analyzed or as noise. An FFT/DFT works by taking a set of points and virtually re-propagating them through time as if they correspond to a full period of a periodic wave. If not used with proper resolution and/or a non-periodic signal, this process will result in leakage errors. A way to correct these errors is to use windowing. The windowing processes provide a certain ponderation to each point; even if leakage occurs, it will not have an as relevant impact, as the noise parts will be attenuated and the main signal part will be somewhat boosted. It many also happen that this windowing will consider parts of the noise as part of the actual signal, therefore inflating the real value of the signal, while also reducing the noise value.

The windows considered in this report are the following: Rectangular, Hamming and Blackman-Harris, whose effects on the frequency domain can be seen in Figure 2. For the input sinusoidal wave with $\mathrm{F_i = 33.203125kHz}$, the coherent sampling by the isolated spike on the 17th sample when using a simple rectangular window can be confirmed. The SINAD (shown in Table 2) is also very close to the theoretical maximum of 25.84, thus it can be proved that this represents an ideal sample and that the ADC and DAC are working as expected, while introducing minimal errors. This results in ENOB=3.981, quite close to the value of 4 bits.

For the same input and with the other windowing processes, the DFTs give off the idea that there is some leakage and therefore spread spectrum. However, the SINAD and ENOB values are higher - with a Blackman-Harris window, the maximum value of 4 bits is even surpassed, as seen in Table 1. The fact that the sampling was coherent implies that the use of these windowing processes results in noise close to the signal frequency now being considered as part of the signal.

When it comes to the 28kHz sinusoidal input, the mentioned leakage is visible, as the energy of this frequency appears disperse throughout the neighboring frequencies of the highest pike, at 27.3438kHz (the closest sampled frequency to 28kHz). Using the Rectangular window, it is obvious that smaller values for the SINAD and ENOB are obtained, as the signal energy is smaller; the neighboring frequencies are now of much larger values, but will contribute to the noise. In this case, the use of windowing is significantly more beneficial, as not only the magnitude of frequencies further from the largest peak will be attenuated, but also the components contributing to noise are reduced - since the frequencies next to the largest peak will also count for the signal energy. Using the Blackman-Harris window, the SINAD and ENOB values are almost as good as if coherent sampling had been made.

In conclusion, it can be postulated that the best way to get an accurate measurement is to sample with a resolution which is multiple of the input signal's frequency. In case this is not possible (for instance, due to not knowing the input frequency or having a signal with various non-multiple frequencies), windows should be used in order to minimize leakage effects and obtain better results. Another alternative to try and minimize the bitter consequences of windowing on coherent signal (for instance, higher ENOB and SINAD values than expected) is to increase the number of samples.

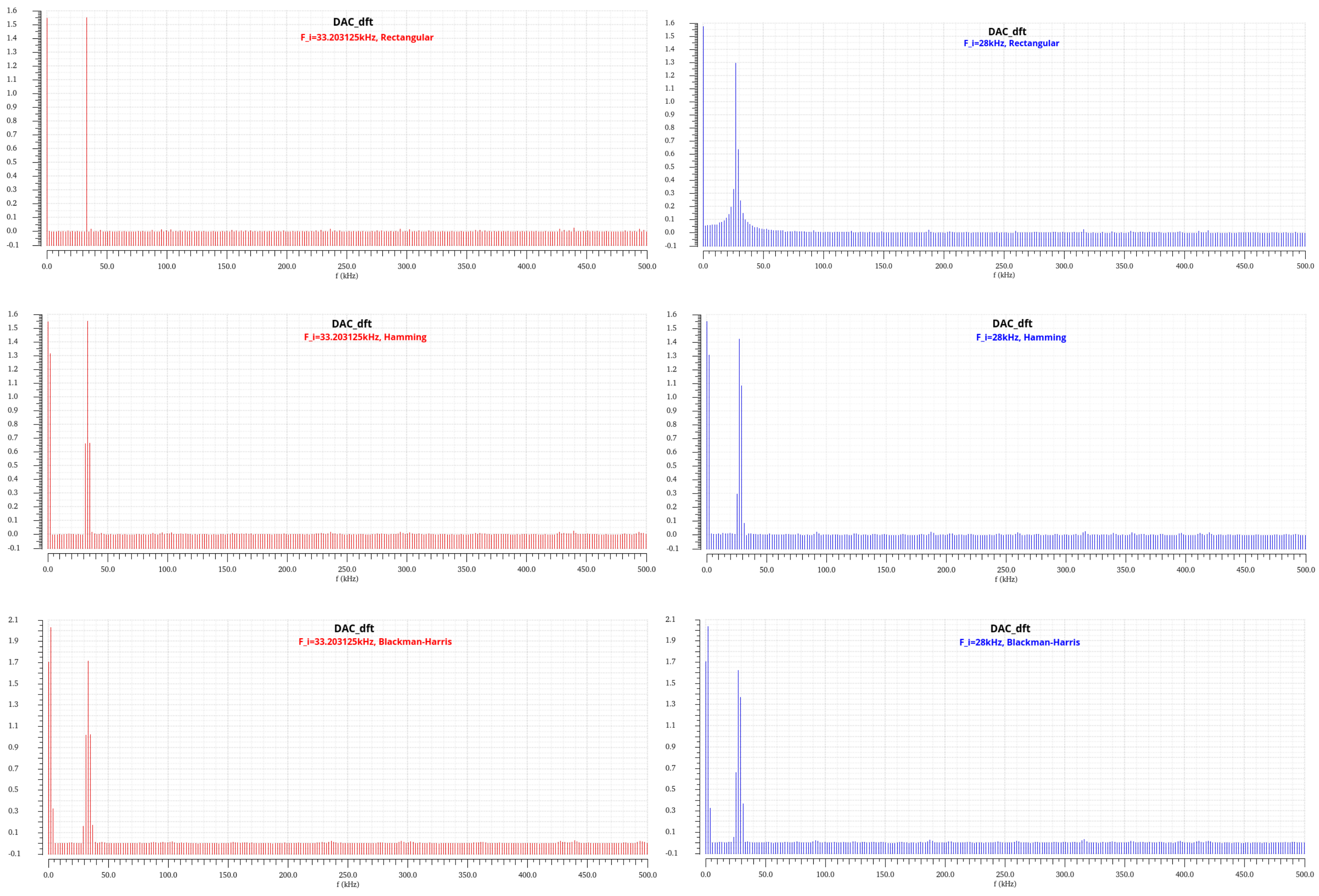

Figure 2: Discrete Fourier Transform (DFT) results obtained with the schematic in `test_sar4b_SIA`, using a sampling frequency of $\mathrm{F_s} = 1\mathrm{MHz}$ and the code lines `dft(VT("/OUT_DAC") 6e-06 0.000518 512 "[window type]" 1 "default")`, where `[window type]` is replaced by `Rectangular` (graphs shown above), `Hamming` (in the middle) or `Blackman` (below). On the left, results obtained with an input frequency of $\mathrm{F_i} = 33.203125\mathrm{kHz}$; on the right, same results for $\mathrm{F_i} = 28\mathrm{kHz}$.

3) Sample and Hold (S&H)

Finally, an ideal sample and hold schematic was designed as shown in Figure 3, in order to run similar simulations and compare the respective results with those obtained from the ADC. This device can almost be seen as using an infinite number of bits, thus it is only limited by mathematical errors of the computer and simulation limitations, since the results are obtained with continuous voltage values and not using a certain (finite) number of bits. The working principle of a typical sample and hold device relies on the signal passing through a controlled switch connected to a capacitor. Upon triggering of the circuit, the switch turns ON and the capacitor charges. When the trigger turns OFF, the switch opens and the voltage at the capacitor is kept the same. There is no quantization of values, and present errors are due to the discretization at a finite frequency.

As done previously, the maximum error between the input and output signals can be obtained. This maximum error of each sample happens right before the commutation to the next value occurs, thus a maximum difference of 0.067V and an average of 0.038V as the maximum error for each bit was obtained. Something worth noting has to do with the capacitor - the sample and hold (shown in Figure 3) used for these simulations has no physical capacitor, which should lead to an instantaneous and lossless device. However, in a real physical circuit, this would be impossible and would depend on the capacitor size. A small capacitor would be faster to charge, but would be extremely prone to lose charge; on the other hand, a large capacitor would not lose charge as easily, but would take longer to charge.

Because the introduction of error is almost none, the signal-to-noise ratio reaches unrealistically high values, as shown in Table 3. To get the same precision in a real device, an ADC would need 47 bits, having a resolution of $1/(2^{47}) \approx 7.1\mathrm{pV}$. Similarly, significantly high SFDR values were obtained.

In the DFT graphs shown in Figure 4, it can be seen there are no peaks at $\mathrm{f} = 0\mathrm{Hz}$, as opposed to what happened with the previous schematic (as shown in Figure 2). This is due to the fact that, with the DAC, an offset was introduced in the modulated signal, while that procedure does not occur here. Moreover, the plots obtained for the S&H have peaks with much higher magnitudes (similar to or higher than 500), whilst the DAC always led to magnitudes lower than 2.1. This corroborates the fact that the errors in this case are much less significant, as previously concluded with the values shown in Table 3. Finally, a zoomed in portion of the time plots obtained with the S&H is shown in Figure 5, in which a time interval for when the circuit is in Hold mode (i.e., when the input voltage is not sampled, thus the output voltage remains the same) is indicated.

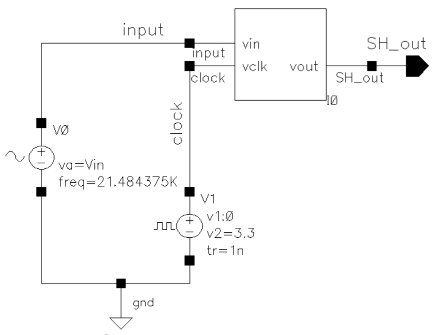

Figure 3: Schematic with implements an ideal Sample and Hold (S&H), by using an instance `sah_ideal` from `ahdlLib`.

$\mathrm{F_s}$ (MHz)	$\mathrm{F_i}$ (kHz)	**Window**	**SINAD** and SNR [dB]	ENOB	SFDR [dB]
1	21.484375	Rectangular	286.3	47.27	292.3
		Hamming	288.9	47.70	297.5
		Blackman-Harris	289.4	47.79	297.8

Table 3: SINAD, SNR, ENOB and SFDR values obtained by using the Sample and Hold schematic shown in Figure 3. For this purpose, a transient analysis (liberal, stop time $600\mu s$) was run, in which the sampling frequency in the instance `vpulse` and the input frequency in `vsin` were defined as shown above and an amplitude of $\mathrm{V_i} = 500\mathrm{mV}$ was selected for the sine wave. The results were obtained from `spectrumMeas(VT("/SH_out") 6e-06 0.000518 512 0 0 nil "[window type]" 0 "[parameter]")`, where `[parameter]` is replaced by `sinad`, `snhr`, `enob` or `sfdr(db)` (respectively) and `[window type]` is replaced by `Rectangular`, `Hamming` or `Blackman`.

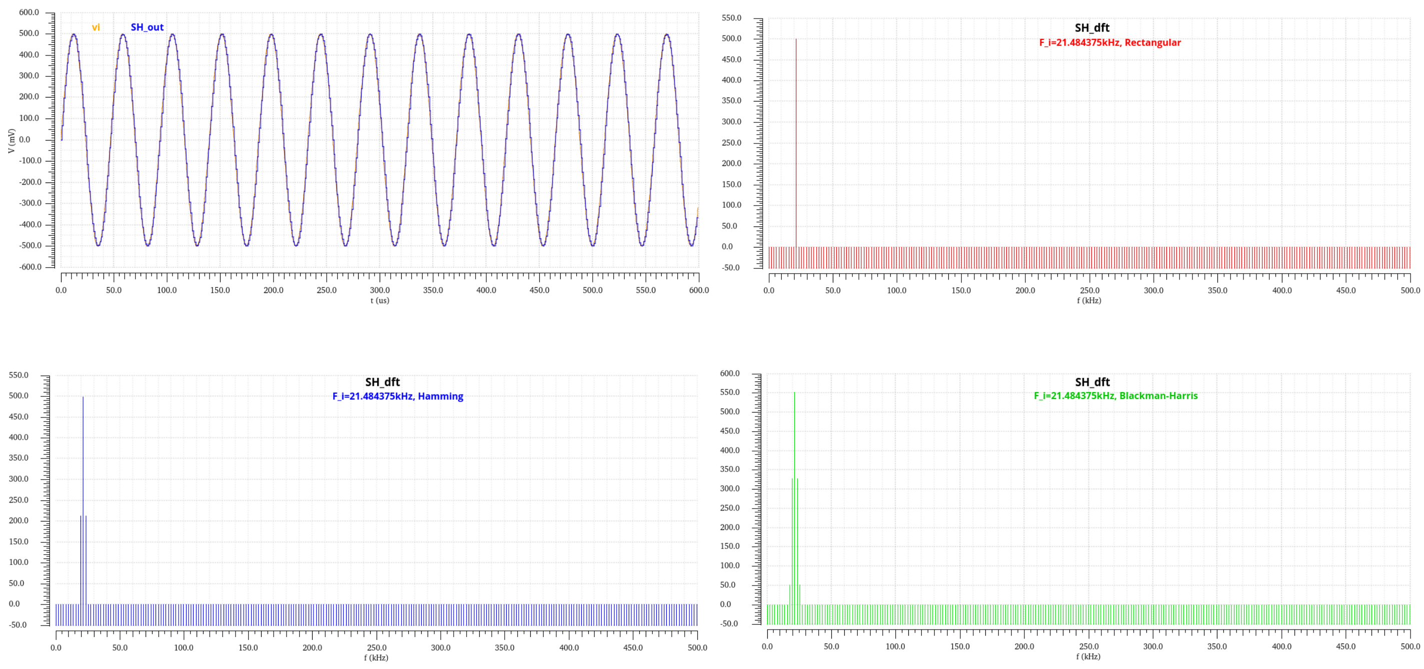

Figure 4: Results obtained by running a transient analysis (liberal, stop time $600\mu s$) with the schematic shown in Figure 3, using a sampling frequency of $\mathrm{F_s} = 1\mathrm{MHz}$ and input frequency $\mathrm{F_i} = 21.484375\mathrm{kHz}$. On the top left, input signal (sine wave with amplitude $\mathrm{V_i} = 500\mathrm{mV}$) and output signal `SH_out`. The graph is shown here in the initial $300\mu s$ for convenience. The remaining graphs correspond to the Discrete Fourier Transform (DFT) results obtained using code lines `dft(VT("/SH_out") 6e-06 0.000518 512 "[window type]" 1 "default")`, where `[window type]` is replaced by `Rectangular` (top right), `Hamming` (bottom left) or `Blackman` (bottom right).

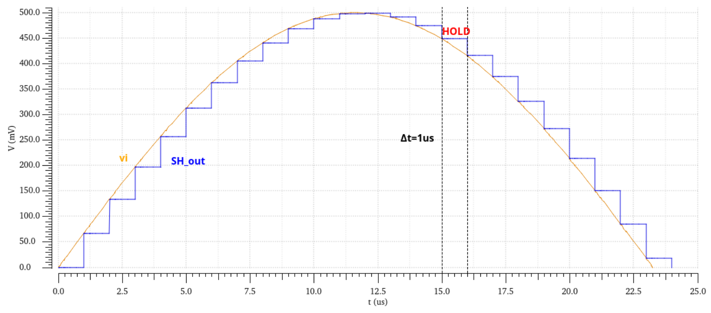

Figure 5: Zoomed in detail of the top left graph shown in Figure 4, in order to clearly identify points for when the circuit is in hold mode - for instance, between $\mathrm{t} = 15\mu s$ and $\mathrm{t} = 16\mu s$.