Fractional-N Frequency Synthesizer – 6.776 Lab 2

Taeg Sang Cho, Tao Pan
Massachusetts Institute of Technology
Cambridge, MA

Introduction

A mixer forms a critical building block for modern communication systems due to the limited budget in the frequency spectrum and the ever-increasing demand of channel capacity. If the mixer provides a single frequency signal with no harmonic components and no phase noise, many problems such as signal interference can be circumvented. The mixers are typically implemented as Fractional-N frequency synthesizers, which can provide high resolutions in frequency and high loop bandwidth (thus faster settling time) at the same time. We will explore this circuitry in more detail throughout the report.

The system that we've developed over the latter half of the semester is shown in Figure 0-1. Discrete components, including MC145151-2 (PLL chip from Motorola), 74LS628 (VCO), were provided. The main topic of this project was to implement a Fractional-N Frequency synthesizer with sigma-delta dithering schemes. The lab was divided into 4 separate parts to ease the analysis of each building blocks.

Part 1: VCO and buffer circuitry

Making the VCO to have a range of 3MHz to 10MHz was not too hard thanks to the datasheet. We used two variable resistors to change the operational range of the VCO and the output frequency. As shown in Figure 1-1, our VCO showed very linear characteristic as we sweep the control voltage of the VCO. With the help of MATLAB, we fitted our result to a first order polynomial; the VCO characteristic follows the

$$OutputFreq = (2.091*10^6)V + 6.991*10^5$$

relation. This characteristic curve corresponds quite closely to the case where Vrng is about 2V. We also looked at the spectral content of the VCO output signal. Since the output of the VCO can be approximated as a square wave, we expect that the spectral content of the signal will contain harmonic multiples of the fundamental frequency. This is exactly what we observed when we looked at the spectral content through the spectrum analyzer. (Figure 1-2)

To implement a buffer to drive the spectrum analyzer, we chose the Class AB topology utilizing BJT's (Figure 1-3). We used the BJT's to implement this circuit because the V_{BE} is quite constant among different BJT's. If we implemented the buffer with MOSFET's, the mismatch of the MOSFET would have skewed the output voltage level to one way or another.

Because of the finite base current in BJT's, we had to make R_A slightly smaller than R_B to ensure that the output DC voltage is centered at 2.5V. If not, the output signal cannot attain the full swing possible, i.e. from the rail to rail, without distortion.

Part 2: The Phase-Locked Loop

Designing the Low-Pass Filter

In order to achieve zero steady-state phase error, we need to generate one pole at the origin in the low-pass loop filter transfer function. After a number of trials, we realized that using the recommended topology in the data sheet is a safe and efficient way to implement the loop-filter, thus we finalized design based on Figure 2-1 (a). In fact, we tried another topology, as shown in Figure 2-1(b) which is

supposed to generate the same transfer function, but it didn't work as we expected to due subtle issues. One major issue is due to the fact that PDout is floating when neither UP nor DOWN are asserted. When floating, this floating voltage should match the virtual ground voltage in order to be effectively floating so that no current from the PLL gets integrated in the filter. This is nearly impossible to do, so we switched to the topology shown in Figure 2-1 (a).

We write the LPF transfer function as $F(s)=\frac{R_2 sC+1}{R_1 sC}$. If we model the whole PLL loop as shown in Figure 2-2, we would be able to write down the closed loop transfer function,

$$H(s)=\frac{R_2 K_{PD}K_{VCO}}{R_1}\frac{s+\frac{1}{R_2C}}{s^2+\frac{K_{PD}K_{VCO}R_2}{NR_1}s+\frac{K_{PD}K_{VCO}}{NR_1C}}=\frac{R_2 K_{PD}K_{VCO}}{R_1}\frac{s+\frac{1}{R_2C}}{s^2+2\zeta\omega_n+\omega_n^2}$$

where $\omega_n=\sqrt{\frac{K_{PD}K_{VCO}}{NR_1C}}$ and $\zeta=\frac{\omega_n R_2 C}{2}$.

According to datasheet,

$$K_{PD}K_{VCO}=\frac{V_{DD}}{2\pi}\frac{2\pi\Delta f_{VCO}}{\Delta V_{VCO}}=1.05\times10^7,\ \ \omega_n=\frac{2\pi}{10}\frac{f_r}{R}=3.93\times10^5,\ \ \zeta\cong1$$

We solve these equations and component values should satisfy $\begin{cases}R_1C=8.51\times10^{-6}\\R_2C=5.09\times10^{-6}\end{cases}$

However, some concerns were raised that we shouldn't believe these equations completely. After trying different values, we finally chose

$$R_1=5K\Omega,\ \ R_2=500\Omega,\ \ C=0.22\mu F$$

and the resulting 3dB bandwidth is 9.8KHz

Measuring Closed Loop Bandwidth

In order to measure closed loop bandwidth, we came up with two approaches. As derived in REF2, the closed loop acts as a low pass for the PD-referred noise and high pass for the VCO-referred noise. If we manage to design the low pass filter so that the closed loop bandwidth is located where the phase noise from VCO and PD intersects, the total phase noise is minimized as well as that the frequency at which the phase noise begins to roll off is roughly the closed loop bandwidth (Fig 2-3). Even for not so good designs where the closed loop bandwidth is not at the optimum noise point, a peak of the phase noise that is located approximately at the closed loop bandwidth can be observable (Fig 2-4). The accuracy is enough. We thus expect to see similar flat waveform from the spectrum analyzer. Unfortunately what we observe in reality is as in Fig 2-5. It is still unexplained why this experiment result is different from theoretical result. Our guess is that there're more noise sources than PD and VCO which shaped the output phase noise as the unexplainable. Or since the carry is not perfect sinusoid, it's hard to differentiate noise and carry signal and the way we plot phase noise is not accurate.

So the other approach is the step response. As for any second order system, if we apply a step input to the system as shown in Figure 2-6(a), the system response similar to Figure 2-6(b) would appear at the output. Here both the input and output are in terms of frequency. Two justifications need to be made before we start the measurement. First, since the system is second order, the relation between the settling time and the bandwidth is somewhat complicated. But as an approximation, the first order

relation $BW = \frac{1}{\Delta t}$ is accurate enough. Also, the step input is applied at the divider – changing one bit of the divider from 0 to 1. From lecture, we learn that the block diagrams in Figure 2-7 are equivalent. Thus even though we apply the input at the divider, the close loop transfer function is unchanged from the transfer function from the reference frequency to VCO output.

Figure 2-8(a) shows the oscilloscope measurement. The time it takes to switch from frequency one to frequency two after the step input applied is 680us. The corresponding bandwidth is 1.47 kHz. Compared to the calculated value (1.56 kHz), the result is fairly accurate. To test the validity of this method, we changed the LPF component values. Figure 2-8(b) is the picture taken after a 4.7uF capacitor is used in the LPF feedback loop. The measured bandwidth is 330Hz while the hand calculation gives 680Hz. Again, there's no error in magnitude.

Part 3: The ΣΔ Modulator

For this part, we use the first-order digital integrator as shown in Figure 3-1. The open loop transfer function $H(z) = \frac{z^{-1}}{1 - z^{-1}}$ gives a closed loop transfer function of z^{-1} and an error transfer function of $1 - z^{-1}$. The corresponding frequency spectrum is shown in Figure 3-2.

We choose to use two's complement to realize the arithmetic part. Since we have 4 dip switches input, the input value ranges from -8 to 7. Ideally, we could obtain 16 different output sequences with different duty cycles if we manage to feedback 7(0111) (rather than 0) if the value before the comparator is above or equal to 0 and -8(1000) if the value before the comparator is below 0. However, in order to avoid the overflow problem, we would need at least 5 bits to perform the math which would increase the circuit complexity greatly. As a trade-off, we decided to feedback 0 instead of 7 when the output of the integrator is a non-negative number and we limit the input range within -8 and -1. With the input ranges from -8 to -1, we obtain outputs with duty cycles of 100%, 87.5%, 75%, 62.5%, 50%, 37.5%, 25% and 12.5% respectively. (shown in Figure 3-3)

Part 4: Putting it all together

With the sigma-delta modulator, we dithered the LSB of the internal counter in the PLL chip. In order to clock the sigma-delta modulator, we used the Fv signal of the PLL chip so that we don't miss out some of the VCO rising edges due to the changes in the divider, thus counter, values. Because our sigma-delta modulator outputs a sequence of one-bit value in the period of 8, we could attain the resolution of 0.625Megahertz/8 = 78.125 KHz. Although the signal showed some jitter (Figure 4-1), in general, the spectrum analyzer showed clear peaks at right frequencies (Figure 4-2). Here is a table of measurement results that shows the phase noise characteristics and attainable output frequencies.

Center Frequency	5.001MHz (Figure 4-2)
Carrier Power	12.1dBm (Figure 4-2)
Phase noise at 10kHz Offset	82.8dBc/Hz (Figure 4-3)
Phase noise at 100kHz Offset	100.25dBc/Hz (Figure 4-4)
Digital Input to the Sigma-delta Modulator	**Attainable Frequency**
0000	5.001MHz
0001	5.078MHz
0010	5.156MHz
0011	5.237MHz
0100	5.313MHz
0101	5.393MHz

0110	5.469MHz
0111	5.55MHz

As you can see, we can increment the digital input bit by bit, and increase the frequency in 1/8 of the resolution in the integer PLL. The measured output frequency is compared with ideal output frequency in Figure 4-5.

To delve into the spur characteristics, we measured the primary spur power and its offset from the center frequency. The following table shows the measurement result.

Center Frequency	Carrier Power	Spur Offset from the CF	Spur Power
5.078MHz	7dBm	0.078MHz	2.6dBm
5.156MHz	9.8dBm	0.157MHz	-7dBm
5.312MHz	9.98dBm	0.311MHz	-6.7dBm
5.549MHz	8dBm	0.078MHz	1.3dBm

An interesting observation is that by changing the voltage that controls the frequency range of the VCO, we can control the spur and the phase noise in the output signal spectrum. This is perhaps due to the fact if the output of the loop filter has some ripples that are not totally filtered out when the PLL is locked, high gain in the VCO (Kv) can actually induce a lot of phase noise. This signifies that when the loop-filter is sloppy, there is a certain trade-off between high Kv to attain high loop-gain and low Kv to attain small phase noise. In other words, we need to set the range of the VCO appropriately to accommodate the necessary range of frequency that we design to synthesize, attain high enough Kv to have high loop-gain to decrease the phase difference between the reference clock signal and the output of the VCO, while low enough Kv to be able to tolerate the ripples from the output of the loop filter. Thus, designing a good loop-filter is a major concern from the PLL system perspective.

OPTIMIZATIONS

Added Digital Circuitry to randomize the output of the sigma-delta modulator

Because loop-filters pose serious problems in the phase noise of the output signal, we wanted to implement an adaptive loop-filter that adapts its bandwidth based on the band-width and noise performance requirements, as in Figure 4-6. By varying the loop dynamics based on the input digital words that signify the frequency to be synthesized, we could make better trade-offs in terms of settling time and phase-noise performance. However, many non-idealities with off-the-shelf capacitors, resistors and OPAMP resulted in a hassle with the loop-filter in part 2 of the lab, and this discouraged us a great deal in pursuing this path. Under a more controlled process (such as IC implementations), variable loop-filter could be a very interesting topic to consider.

After all, as an optimization, we decided to decrease the spurs by employing the digital dithering scheme in the sigma-delta modulator. Spectrum analyzer shows that we have many spurs around the center frequency if we dither the internal counter in the PLL chip with the first-order sigma-delta modulator. This is due to the idle tones that are neglected in the derivation of the sigma-delta modulator model. The introduction of the spur can be referenced back to the dithering between two frequencies that are available from the integer part of the PLL. When the output frequency can be denoted as $f_{out} = M * f_{ref} + \frac{k}{N} * f_{ref}$, where there is no common denominator for k and N, the spur spacing is given as $\frac{f_{ref}}{N}$. Thus, when generating the output frequency of 1/8, 3/8, 5/8, 7/8 of the

reference frequency, we get the worst spurs. These can be denoted as the idle tones present in the system. To exacerbate the situation, these spurs will actually interact with one another again due to non-linearity present in the system to increase the power.

Furthermore, another problem is that when the input of the sigma-delta modulator is DC, then the modulator encounters limit-cycle problems. When the limit cycle occurs, a very strong tone will be present at the output of the sigma-delta modulator, which will prohibit the noise-shaping functionality of the modulator. On top of that, this strong tone will be very hard to low-pass filter through the PLL dynamics. Therefore, we thought we can get rid of some of these spurs by adding more randomness in front of the comparator, which will effectively smear out the strong tone present for DC inputs.

In order to achieve this, we use the GLFSR (Galois Linear Feedback Shift Registers) to generate the random numbers. The conceptual visualization of the GLFSR is shown in Figure 4-6. What is different from the LFSR (Linear Feedback Shift Registers) is that the GLFSR does not concatenate every register to the next. Instead, the GLFSR employs a configuration in which some of the register outputs are tapped to produce an XOR with the LSB of the GLFSR to produce the following bit, as shown in Figure 4-7. In doing so, we can guarantee more randomness in the number generated through GLFSR. However, there are several conditions to satisfy in order to guarantee the generation of every possible combination of numbers with N registers. To summarize, we should tap the output of the registers at prime numbered position. In doing so, we can satisfy the necessary condition to achieve largest possible variations in the number we get from the GLFSR (this maximum number of variations we get from GLFSR is 2^N-1, where N is the number of registers in GLFSR). Also, the number of registers to tap depends on the number of registers you have. In general, larger the number of registers, larger the number of tapped outputs is. Interested readers can refer to REF 1 for more information.

We chose to have 12 registers to generate the random number due to area constraints, and we tapped the output of the registers at 3rd, 5th, and 11th position in the register array. It is proven that by tapping the outputs from the above registers we can get the full range of number available from 12 registers. In order to add the randomness in the output of the sigma-delta, we added the LSB of the random number in front of the comparator, as shown in Figure 4-8. This seems to add non-zero DC component in the sigma-delta modulator output, but the dynamics of the sigma-delta modulator operates such that the DC component of the sigma-delta modulator is suppressed. In other words, we are randomly adding either 1 or 0 in front of the comparator to randomize the accumulated error term, which the accumulator will try to pin down to zero with the feedback. Thus, even if there is certain DC component we are adding with the GLFSR, we will eventually get the help from the sigma-delta modulator dynamics to suppress the DC component. If the DC component is still a problem due to stability or dynamic range issues (such as overflow), we could choose to assert -1 when the LSB is 0 and assert 1 when the LSB is 1.

The measured output spectrum of the PLL before and after dithering is shown in Figure 4-9. As you can see here, the noise floor in the spectrum rose significantly, while the spur is suppressed quite a bit. The rise in the noise floor is intuitive because we are effectively introducing more random noise into the system with the GLFSR. From the measurement, the peak-power in the largest spurs decreased by 7dBm. The measured phase-noise of our system is illustrated in the table below.

Phase noise at 10kHz Offset	30.16dBc/Hz
Phase noise at 100kHz Offset	44.21dBc/Hz

Although there is a visible enhancement, the improvement is rather small. Furthermore, the improvements in spur characteristics came about at the expense of two-fold increase in background noise. Whether the introduction of the random noise into the system is still beneficial is thus questionable based on the application of this system. The possible reason for only a small improvement could be due to the fact that even if you use GLFSR that generates the full range of numbers with N registers, it doesn't guarantee that the sequence of bits you get from the LSB is as

random (as white) as possible. Therefore, in order to increase the effectiveness of our dithering scheme, we should come up with a better scheme to generate the random sequence of bits.

Because we couldn't get rid of spurs as much as we wanted, we decided to push forth implementing a second-order sigma delta modulator. In pursuing this path, we thought that more noise shaping would lead us to better noise performance and also smaller spurs.

Second Order Sigma-Delta Modulator

We tried three different topologies as shown in Figure 4-10. The reason we added the shaded delay block in topology (a) is to block the direct feedback and make the system stable. The closed loop signal gain G(z) and error gain E(z) are expressed as follows:

Topology (a) $G(z) = \frac{(2 - z^{-1})z^{-1}}{(1 - z^{-1})^2 + (2 - z^{-1})z^{-1}}$, $E(z) = \frac{(1 - z^{-1})^2}{(1 - z^{-1})^2 + (2 - z^{-1})z^{-1}}$

Topology (b) $G(z) = \frac{1}{(1 - z^{-1})^2 + 1}$, $E(z) = \frac{(1 - z^{-1})^2}{(1 - z^{-1})^2 + 1}$

Topology (c) $G(z) = z^{-1}$, $E(z) = (1 - z^{-1})^2$

The frequency response of the three LPF is plotted as in figure 4-11 with the first-order LPF response included as a reference. The solid line is signal and the dotted line is error.

We see that topology (c) is the second-order loop filter in the common sense while (a) and (b) shows some interesting characteristics such as increasing the signal or suppressing the error. But all shape the noise and push it to high frequency as we expect. To see what is really happening in the time domain, we simulated all circuits in matlab. As in part 3, we again choose -8 and 0 as feedback factors and 1/8 as the frequency resolution to make the comparison with first-order fair.

However, topology (a) and (b) give only 7 outputs with the duty cycle 100% and 62.5% missing respectively. We figure it might be due to the irregularity of the frequency response. One thing to note though, the valid outputs of (a) and (b) do bear a more random fashion. We list first-order and second-order (c) output sequence in the table below.

DC	first-order	second-order (c)
1	1	1
87.5%	11111110	1101111111101111
75%	1110	10110111
62.5%	11011010	0110101101011011
50%	10	0011
37.5%	10100100	0001010010100011
25%	1000	00000011
12.5%	10000000	0000000000000011

We can see that output from 2nd order LPF is more random and scattered which means its high frequency component is magnified while the low frequency part is suppressed.

As for implementation, since the maximum and minimum numbers encountered are 28 and -12, 6 bits are enough to give a resolution of 1/8 the original minimum frequency. The measured result is shown in

Figure 4-12.

Conclusion

On top of the fractional-N synthesizer utilizing a first-order sigma-delta modulator, we have devised a method to enhance the performance of the system by increasing the randomness of the sigma-delta modulator output and introducing a second-order sigma-delta modulator in the dithering scheme. Although we have achieved noticeable enhancement in performance, several points need to be studied further. First of all, a better random-vector generator should be studied in the digital dithering scheme in order to decrease the spur at the output of the synthesizer. Also, different schemes to add the random noise into the system should be studied to increase the randomness of the sigma-delta modulator output.

There are also a number of ways to improve our design second-order sigma-delta modulator in future. Due to the wiring complexity, we limited the number of effective ALU bits to be 6. This gives 80kHz resolution which is fairly small. However, we later realized that in order to achieve 40kHz resolution, only one more bit is needed. It's a pity we didn't try to push ourselves harder for a finer resolution. Although we were able to generate random sequence through random number generator and second-order sigma-delta, we weren't able to reduce the phase noise visibly. We see VCO output waveform dithering between two frequencies on the oscilloscope, which is worse than the first-order simple case. At first, we figured it might be the loop filter's problem. But we weren't able to solve it by changing component values. A more detailed calculation of the loop filter is required in order to filter out high frequency noise effectively. Since the second-order part is a really big digital module, noise injected into ground might influence the analog part negatively, which is another possible reason for the observed noisy waveform.

The future generations of the project can delve into these issues to further enhance the performance of the system.

REF1. D. K. Pradhan et al, "GLFSR—A New Test Pattern Generator for Built-in-Self-Test", IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 18, NO. 2, FEBRUARY 1999

FIGURES

Figure 0-1

System Block Diagram

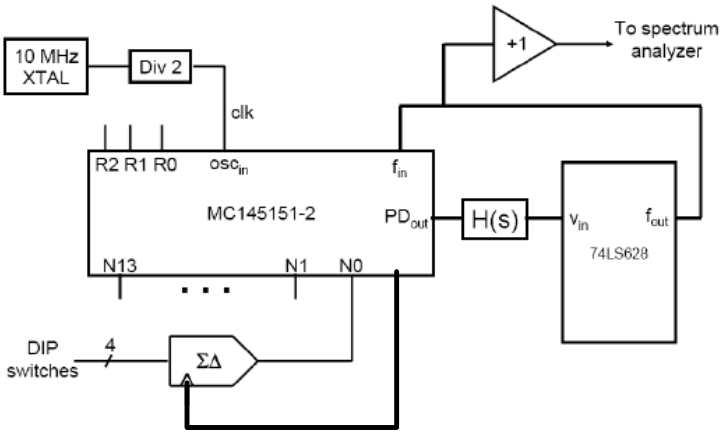


Figure 1-1.

VCO Characteristic

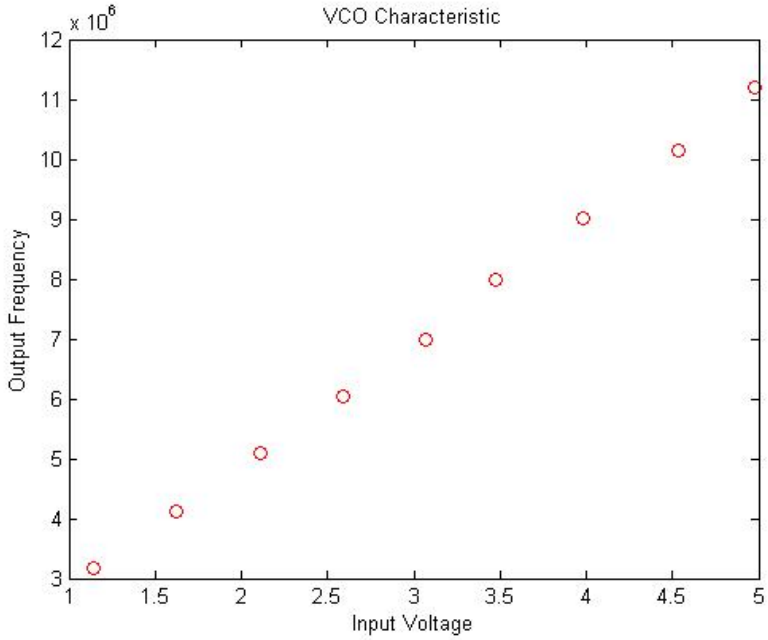


Figure 1-2

Spectral Content of the Output of the VCO at 5.5MHz

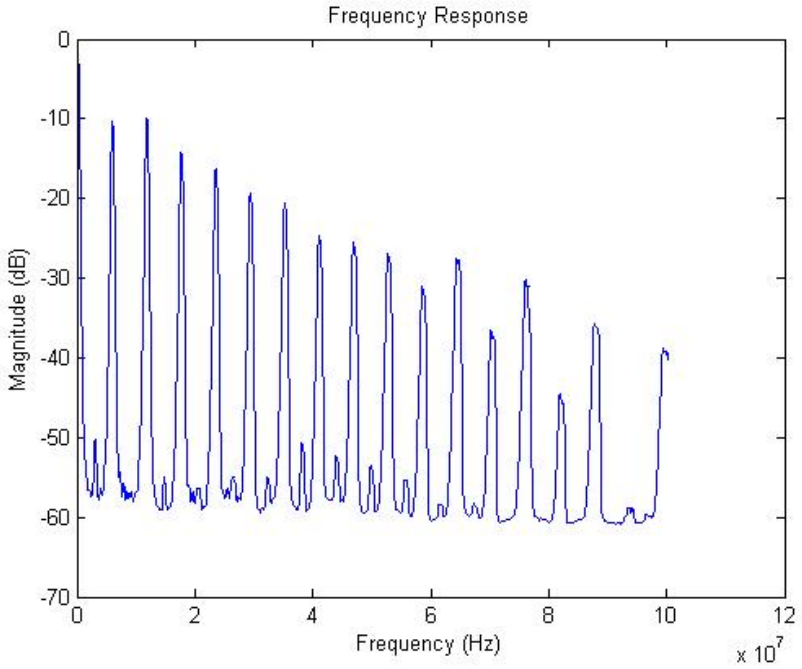


Figure 1-3

Buffer Schematic

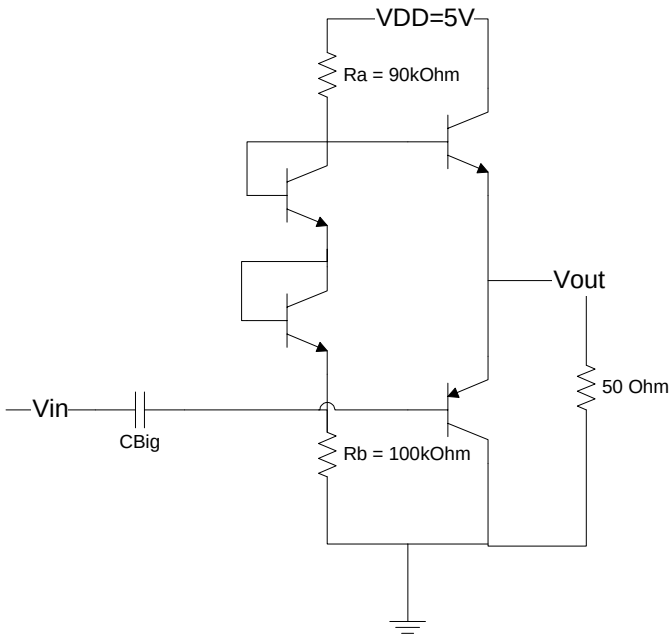


Figure 2-1
Loop-filter Schematic

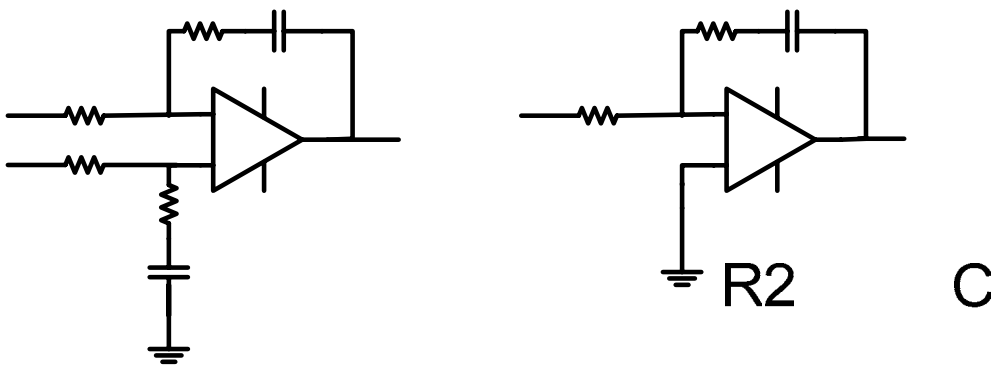


Figure 2-2
Linearized PLL Model

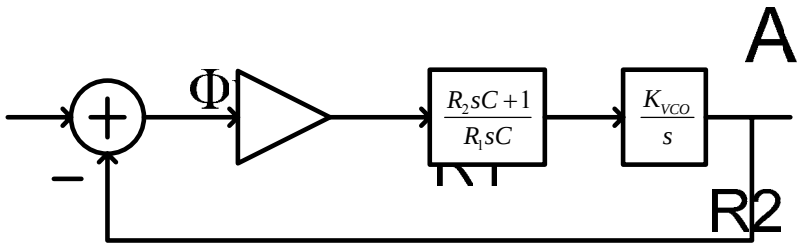
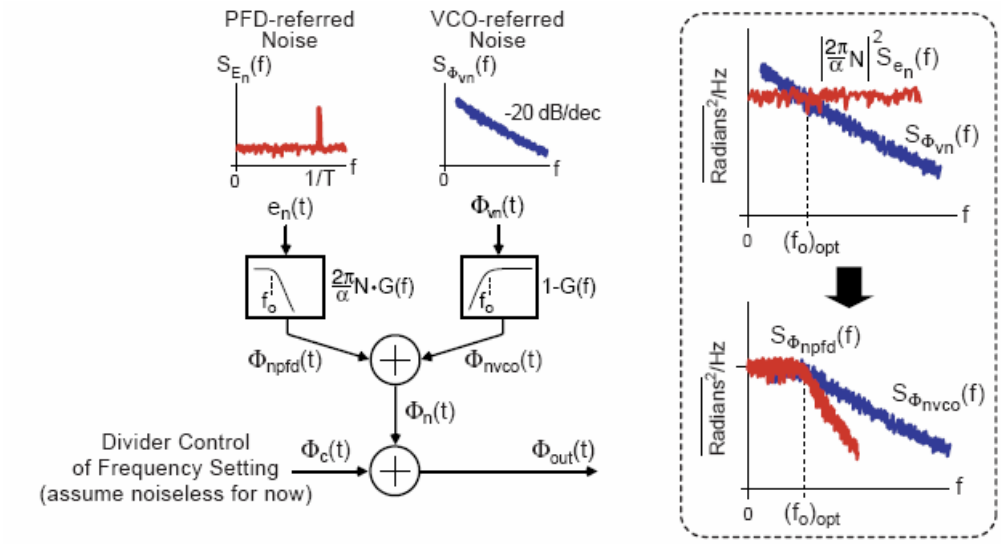


Figure 2-3
Relation between 3dB bandwidth of phase noise and closed loop bandwith



Courtesy: Mike Perrott

Figure 2-4
Relation between 3dB bandwidth of phase noise and closed loop bandwith for bandwidth too small and too big

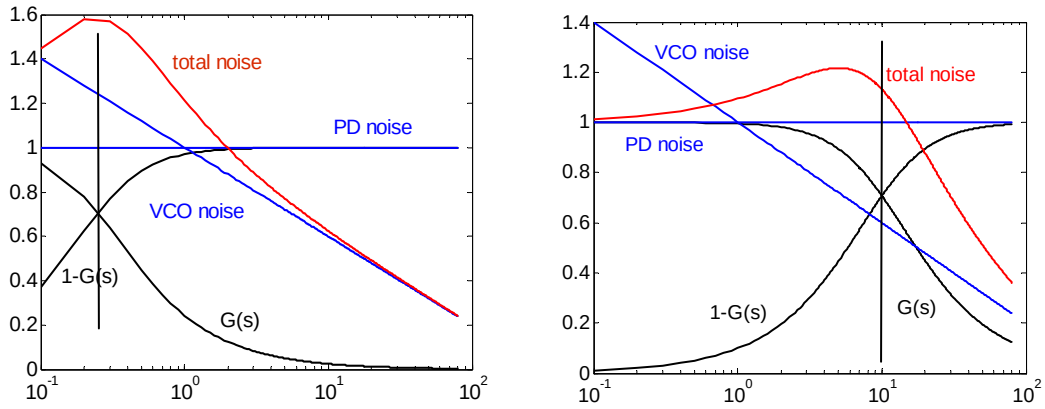
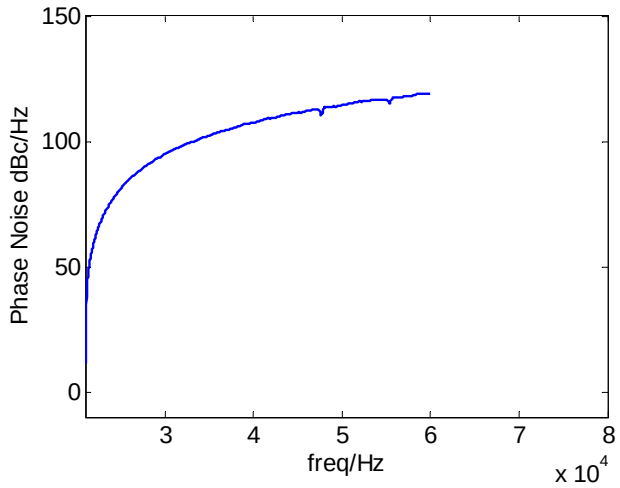


Figure 2-5
Phase noise spectral density from spectrum analyzer



(a) bandwidth smaller than minimum

Figure 2-6
Step response of second order system

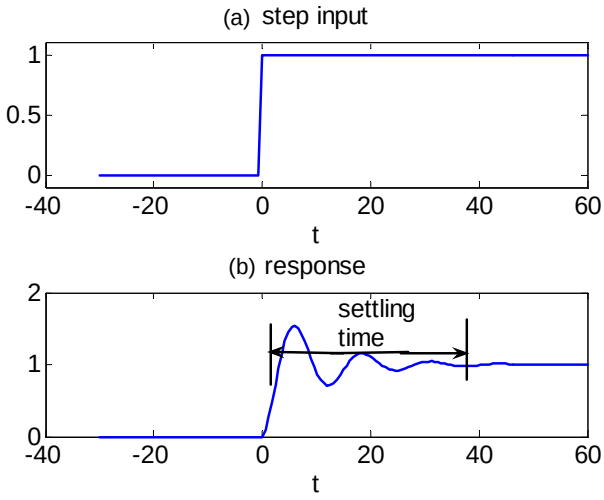


Figure 2-7
Equivalent block diagram of PLL

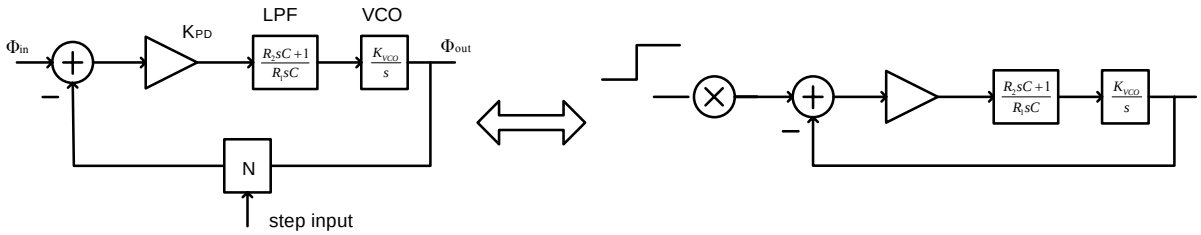


Figure 2-8
Measured step response for different LPF cap values

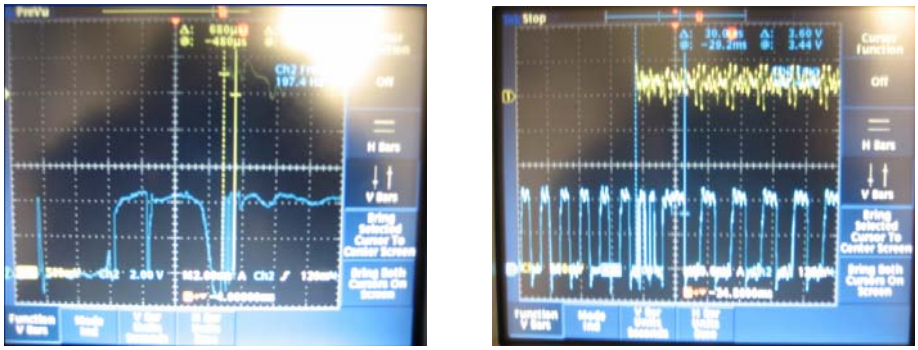


Figure 3-1
Block diagram of first-order ΣΔ modulator

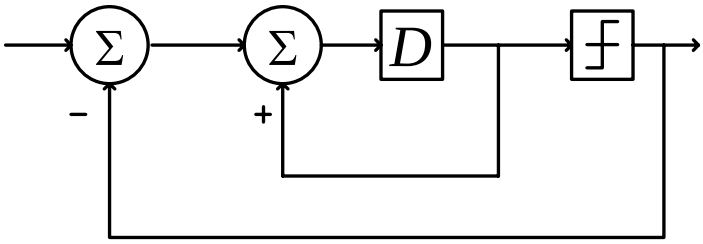


Figure 3-2
Frequency response of first-order ΣΔ modulator

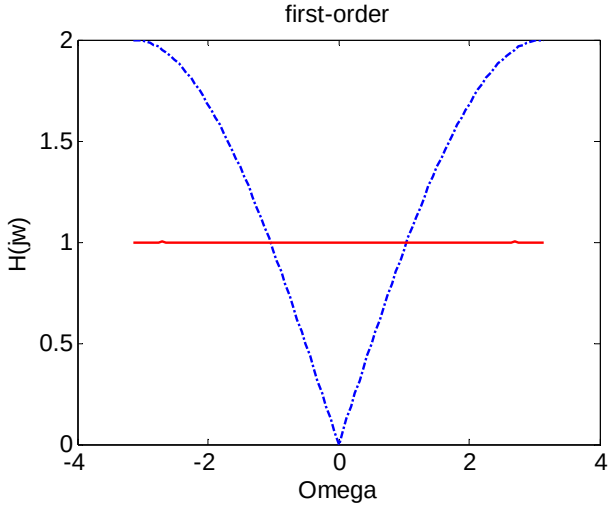


Figure 3-3
Output sequence for different input

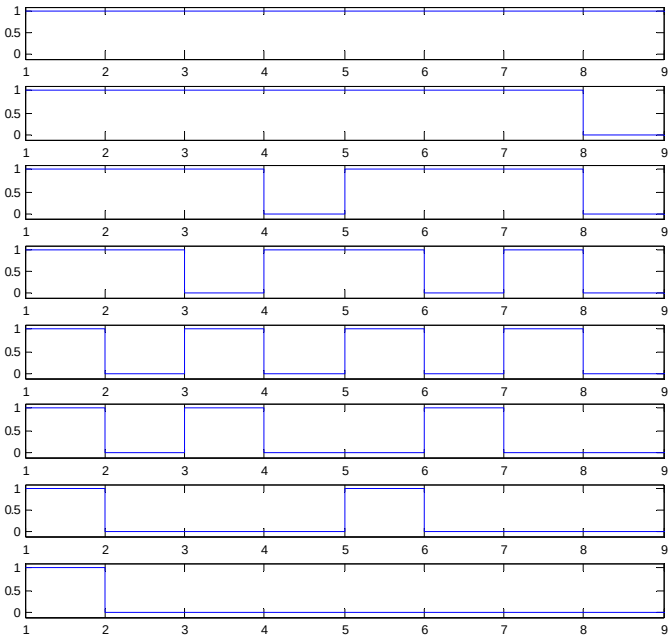


Figure 4-1.

Time domain signal of the output of the Fractional N.

Figure 4-2.

Spectrum Analyzer plot of the above signal.

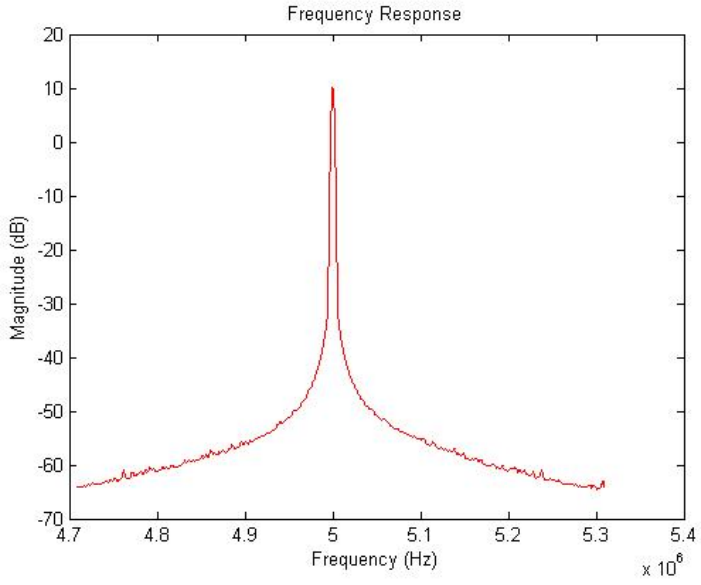


Figure 4-3.
Phase noise at 10kHz offset

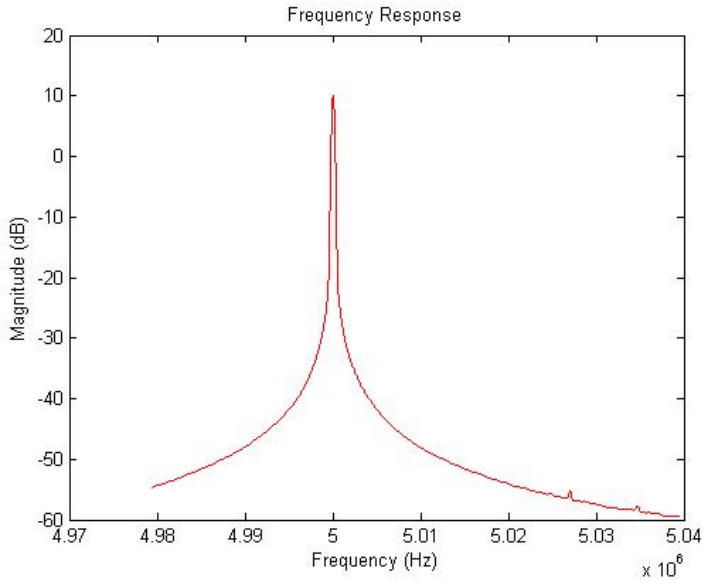


Figure 4-4.
Phase noise at 100kHz Offset

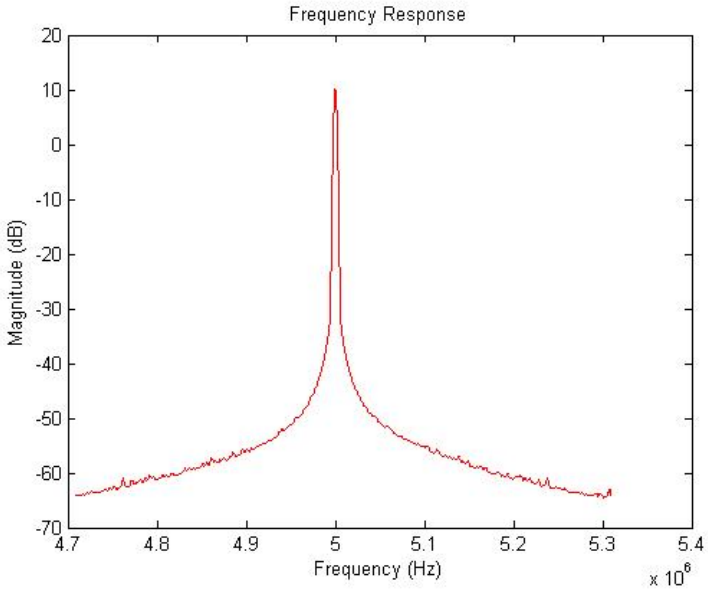


Figure 4-5.
Measured output frequency versus ideal output frequency

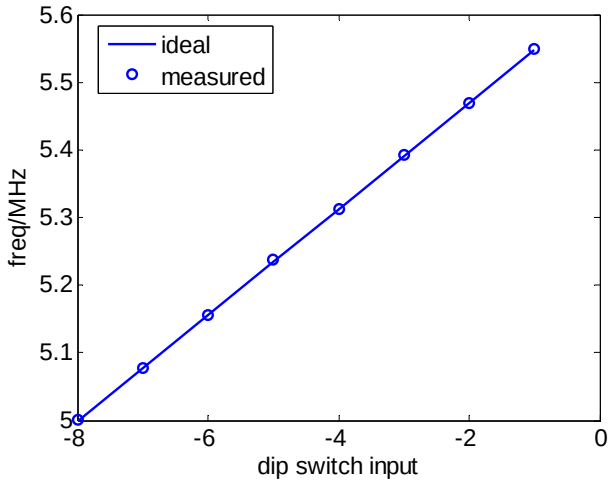


Figure 4-6.
Variable Loop-filter (Courtesy: SeungHwan Cho, MIT Ph.D. Thesis)

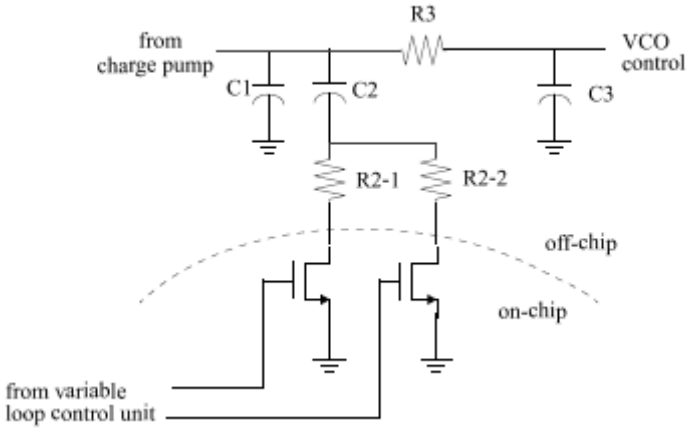


Figure 4-7.
Galois LFSR Scheme

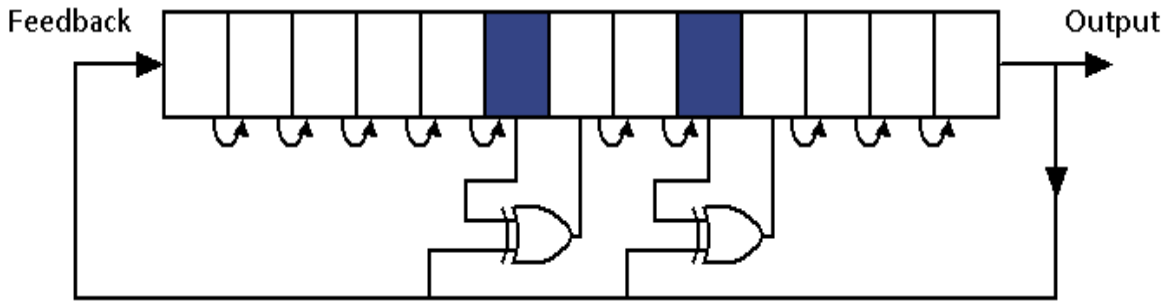


Figure 4-8.

Sigma-Delta Block diagram with digital dithering

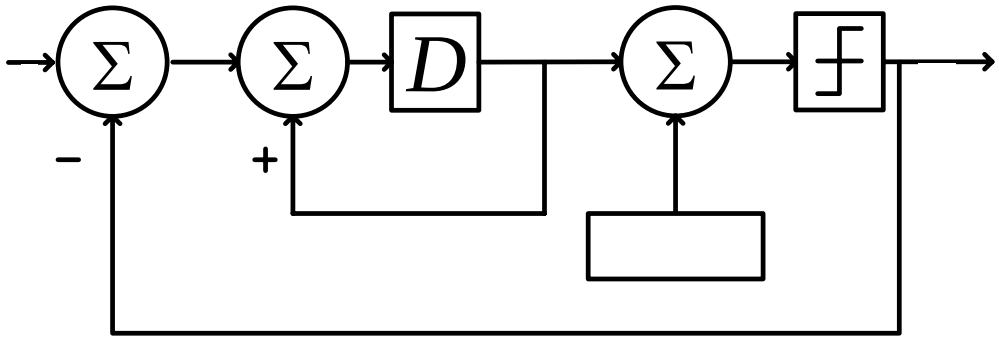
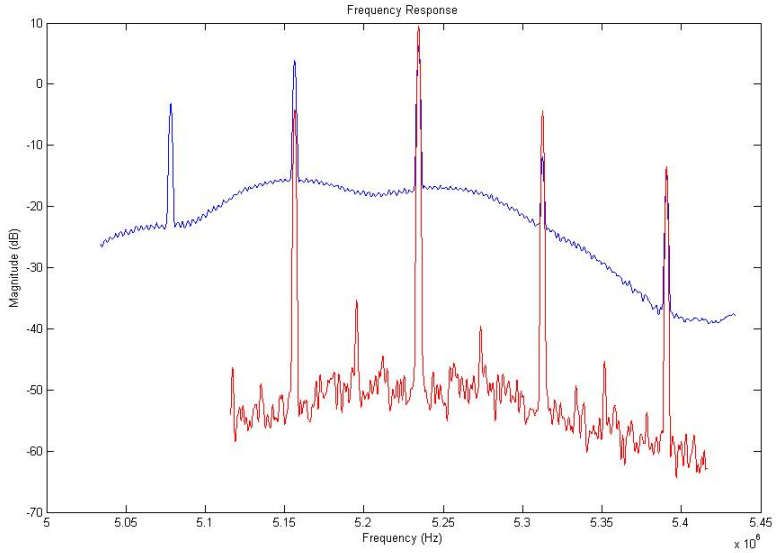
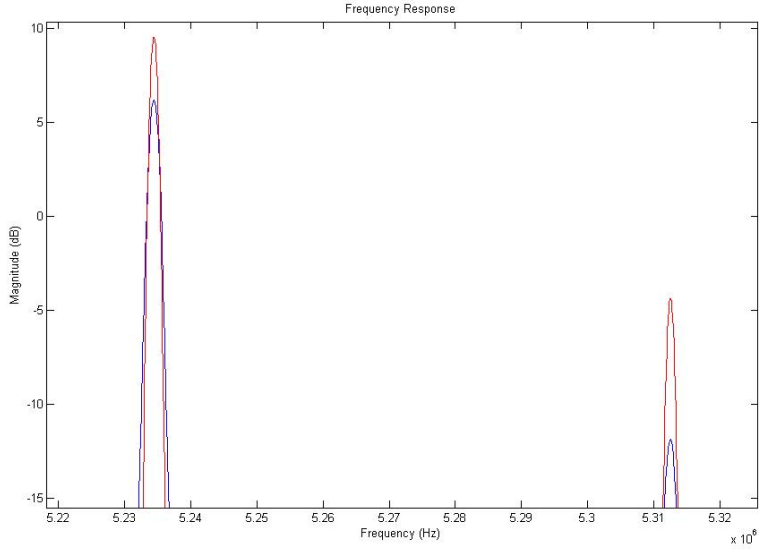


Figure 4-9.

Output Spectrum of the PLL Before/After Dithering

M



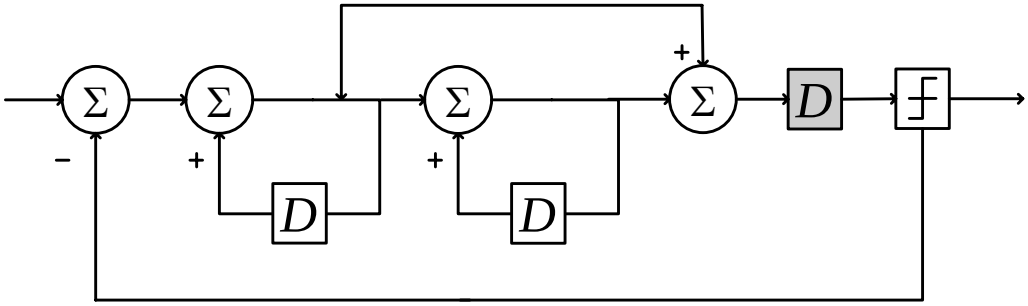


RED Graph = No Dithering

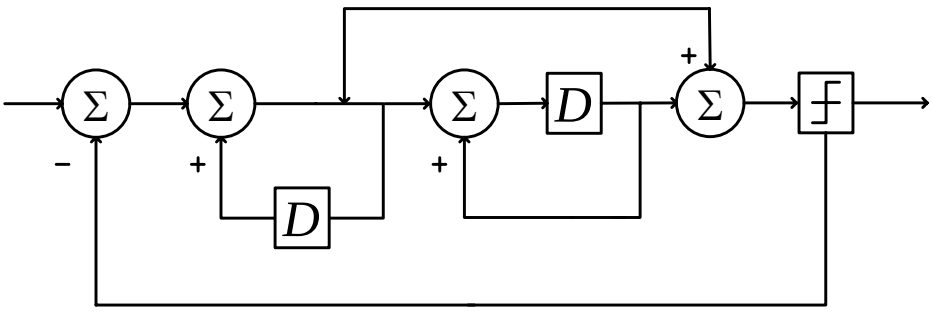
BLUE Graph= With Digital Dithering

Figure 4-10

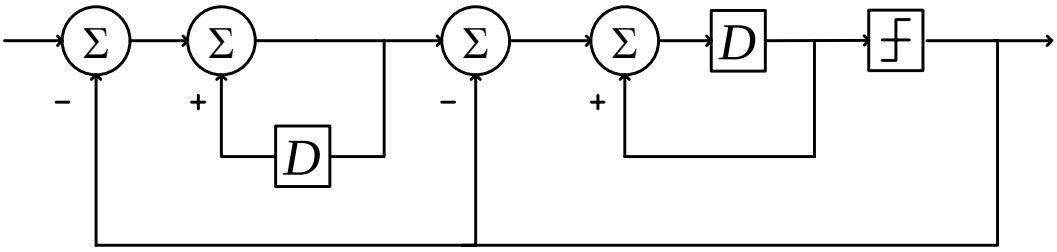
Block diagram of second-order ΣΔ modulator



(a)



(b)



(c)

Figure 4-11
Comparison of frequency response of second-order ΣΔ modulator

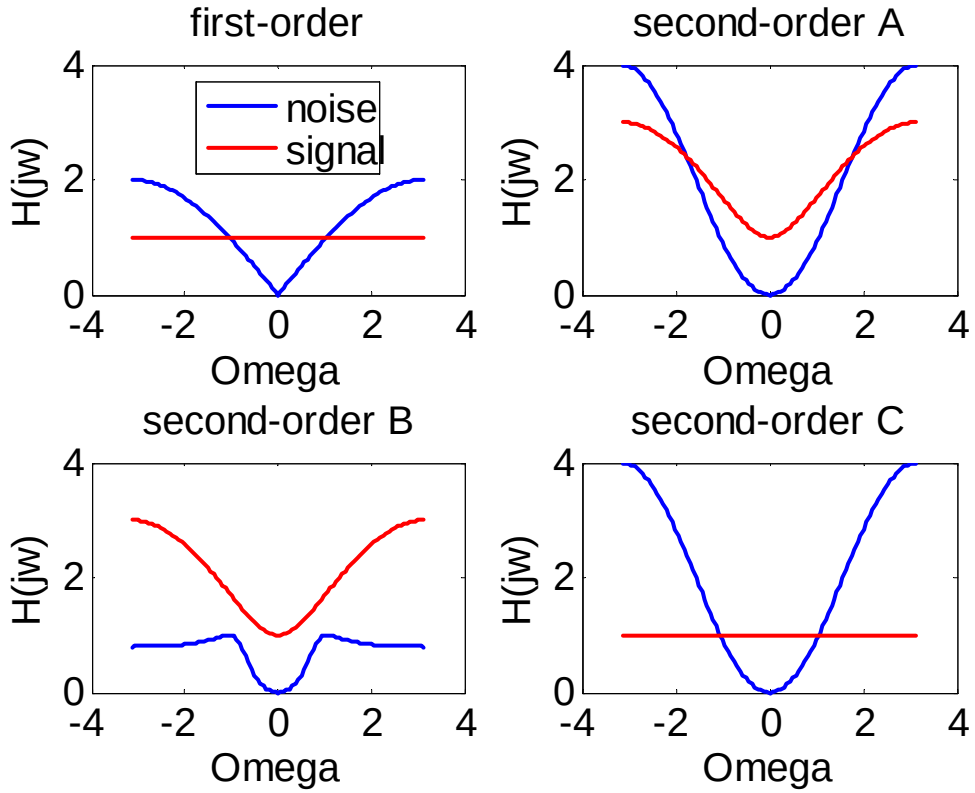


Figure 4-12

Measured result with second-order sigma-delta modulator