NON-LINEAR D/A CONVERTERS FOR DIRECT DIGITAL FREQUENCY SYNTHESIZERS

By

ZHIHE ZHOU

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

AUGUST 2006

To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of ZHIHE ZHOU find it satisfactory and recommend that it be accepted.

____
Chair

____

____

ACKNOWLEDGEMENT

I would like to thank Air Fore Research Laboratory (AFRL) and Center for Design of Analog-Digital Integrated Circuit (CDADIC) for funding the research project. I would also like to thank Dr. George La Rue for his advice and support on this dissertation.

I would also like to the School of Electrical Engineering and Computer Science for assistantship in completing my research.

NON-LINEAR D/A CONVERTERS FOR DIRECT DIGITAL FREQUENCY SYNTHESIZERS

Abstract

By Zhihe Zhou, Ph.D.
Washington State University
August 2006

Chair: George S. La Rue

We investigated the design and implementation of non-linear D/A converters (DAC) for a direct digital frequency synthesizer (DDFS), which consists of a phase accumulator, a phase-to-sine look-up table and a DAC. Using a non-linear DAC can significantly reduce the look-up table size, hence, reducing the power dissipation and enabling high-speed phase-to-sine conversion. A non-linear DAC with 12-bit resolution was designed and fabricated on Honeywell MOI5 0.35 μm SOI process. It implements a 32-segment piece-wise linear approximation to a sine function. The non-linear DAC is uniquely designed to implement multiplication using a high-speed analog method. The look-up table is reduced from 11K bits to 544 bits. Test shows the non-linear DAC can operate up to 600 MHz. The spurious-free dynamic range (SFDR) is up to 72 dBc and the radiation tolerance is 200 Krad Si. Another 14-bit non-linear DAC was also designed and fabricated on a commercial 0.18 μm CMOS process. This non-linear DAC implements a 16-segment piecewise quadratic approximation and is used in a DDFS with 14-bit phase resolution, achieving even higher memory compression ratio. The target operation speed is 2 GHz and the SFDR is anticipated to be over 80 dBc.

TABLE OF CONTENTS

CHAPTER

LIST OF PUBLICATIONS

[1] Zhihe Zhou and G. S. La Rue, “A 12-bit Non-linear DAC for Direct Digital Frequency Synthesis," *IEEE Transactions on Circuits and Systems I*, Submitted May 10, 2006.

[2] Zhihe Zhou and G. S. La Rue, “A Radiation-Hard Non-linear DAC for DDFS," *12th NASA Symposium on VLSI Design*, Coeur d'Alene, ID, October, 2005.

[3] Zhihe Zhou, I. Horowitz, G. S. La Rue, “Design of a radiation-hard DDFS,” *IEEE Workshop on Microelectronics and Electron Devices*, pp 88-91, April 15, 2005

[4] Zhihe Zhou, I. Horowitz, G. S. La Rue, “Non-linear DAC implementations in DDFS,” *IEEE Workshop on Microelectronics and Electron Devices*, pp 124-125, 2004.

[5] Zhihe Zhou, D. Betowski, G. La Rue, V. Beiu, “High performance direct digital frequency synthesizers”, *Proceedings of the 15th Biennial University/Government/Industry Microelectronics Symposium*, June 2003.

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

1. INTRODUCTION

Frequency synthesizers are one of the important building blocks of modern communication systems. Their job is to translate the performance of a reference oscillator to frequencies useful to users. Frequency synthesizers can be classified as direct and indirect synthesizers. They can be implemented as either analog circuits or digital circuits. Most of the time they are implemented as a combination of both. Direct frequency synthesizers convert an input reference to the desired output frequency directly, using analog mixers or digital counters. Indirect frequency synthesizers often use some feedback methods to generate the desired output frequencies. Usually, a phase lock loop (PLL) with frequency transfer unit in its feedback path is involved. The basic parameters that character the performance of a frequency synthesizer include frequency range, frequency resolution, settling time, phase/frequency stability, and spectral purity. Frequency range and resolution are mainly defined by applications. Fast settling time and high phase/frequency stability (low phase noise) are always desirable properties for most systems.

1.1 How DDFSs work

Direct digital frequency synthesis (DDFS) is a frequency synthesis technique developed in the early 1970s. The DDFS is unique because it is digitally deterministic; the output it generates is synthesized from a digital definition of the desired result. All other synthesizers start with some oscillator, the output of which is manipulated or controlled by the synthesizer. A DDFS uses logic and memory to digitally construct the desired output signal, and a D/A converter (DAC) to convert it from the digital to the analog domain. Therefore, the DDFS method of constructing a signal is almost entirely digital, and the precise amplitude, frequency, and phase are known and controllable at all times. Figure 1 shows the architecture of a DDFS. It has three main blocks, a phase accumulator, a sine map table, and a DAC.

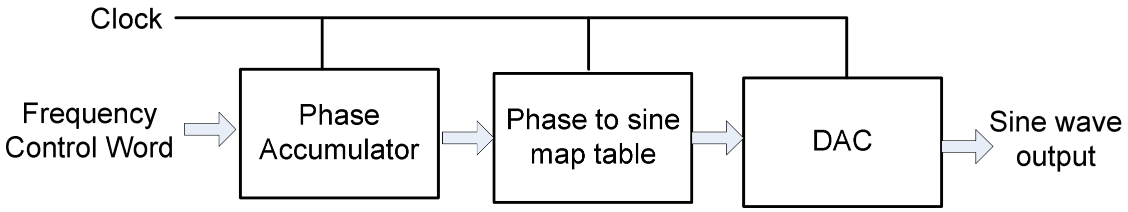


Figure 1 DDFS architecture

Every clock cycle, the overflowing phase accumulator increases the phase by a value determined by the frequency control word. The output of the phase accumulator is used to address the phase to sine map table, which converts the digital phase to digital amplitude of the sine wave. The DAC converts the digital amplitude to analog and forms the sine wave output signal. Usually, there is low pass filter after the DAC to smooth the output waveform. Given an operation clock frequency f_{CLK} and an N-bit frequency control word K (0 … 2^N-1), the output frequency f_{OUT} is given by Eq. (1).

$$f_{OUT} = \frac{K}{2^N} f_{CLK}, \quad (K\text{=0 to } 2^{\mathrm{N}}\text{-1}) \tag{1}$$

The minimum frequency step or the frequency resolution is $f_{CLK}/2^N$. By increasing the number of bits for the phase accumulator, the minimum frequency step is reduced and finer frequency resolution is obtained. Typically, DDFSs have frequency resolutions less than 1 Hz. The output signal frequency is directly related to the frequency control words.

This enables the DDFS to switch to a different frequency very quickly, although there may be some latency. It can also switch between any frequencies within its working bandwidth, which no other synthesizer can do.

1.2 Applications

DDFS provides many significant advantages such as fast frequency switching, fine frequency resolution, and good spectrum purity. Those properties make it very suitable in modern communication systems.

The DDFS is an optimum modulator because waveform manipulation is simple and inexpensive while the signal is in the digital domain. The first two blocks in a DDFS define the frequency, phase and the digital amplitude of the output signal. The addition of a few parts to the DDFS will support combinations of frequency, phase, and amplitude modulation. With DDFS technology, minor modifications of the basic architecture will support virtually all modulation schemes because the DDFS can impress the required modulation directly upon the carrier as an integrated part of the digital construction of the output waveform, without the need for a separate modulation subsystem.

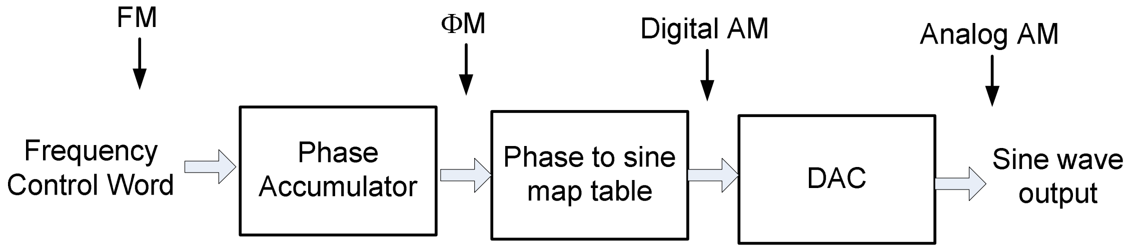


Figure 2 Use DDFS as a modulator

As shown in Figure 2, FM is achieved before the phase accumulator by varying the frequency control word; phase is modulated at the output of the phase accumulator before the conversion table; amplitude can be modulated between the conversion table and the DAC. A varying analog signal applied to the DAC output will also permit control of amplitude to achieve analog AM. When a DDFS is constructed with such control over the signal, it can be used as a generator of complex waveforms.

In addition to being used as a modulator, DDFSs can also be used in a phase-lock loop (PLL) to achieve either finer frequency stepping or higher spectrum purity. Using DDFS in a direct analog synthesizer to achieve up conversion is also possible. The DDFS output f_1 is mixed with an LO f_2 in a mixer to generate two new frequency components $(f_1 + f_2)$ and $(f_1 - f_2)$. A filer can be used to select the desired component if they are separated enough.

1.3 Limitations and challenge

Despite the advantages, DDFS does have limitations. First of all, the Nyquist sampling theory limits the DDFS operational bandwidth to half the clock rate. Practically, the output is limited to about 45% of the maximum clock rate at which the logic can be operated. Therefore, the DDFS does not play a dominant role in the applications for wide-bandwidth frequency generations such as cellular base-station hopping synthesizers and radar systems. Second, the output signal spectrum purity depends on the density and complexity of the logic circuit that is achievable at the operating clock rate. To improve the spectrum purity, more bits are required for the phase as well as the digital amplitude, which increases the circuit complexity and lowers the operating clock rate. Another limiting factor is the DAC. Although the digital signals can be very accurate, the DAC can degrade the signal by 10-20 dB or even more at higher clock rate.

The DDFS design faces two challenges: wider bandwidth and higher spectral purity. In fact, spectral purity and operating bandwidth are inversely correlated. To achieve wider bandwidth, the only solution is to increase the operating clock rate. For digital logic design, one way is to use parallelism and/or pipelining. The drawback is the increased circuit area and power consumption. Since the memory for phase-to-sine conversion table is the slowest part in the digital logic, especially when the memory size is large, another way to improve the clock rate is to use an approximation technique to compress the required memory for the mapping table. The high-speed memory tends to be power hungry too. By reducing the memory size, the total power consumed is also lowered.

Figure 3 shows the block diagram of the conventional DDFS, in which the top 12 bits of the phase accumulator output are used to address a phase-to-sine look up table stored in memory. The memory outputs 12 bits for input to the DAC. This DDFS architecture uses quarter-wave symmetry of the sine wave to reduce the memory by a factor of 4.

In Figure 3, the MSB of the phase accumulator output determines the sign of the result and the second MSB determines whether the amplitude is increasing or decreasing. The remaining 10 bits are used to address the one-quadrant sine memory. The memory stores only $\pi/2$ rad of sine amplitude information instead of 2π . Although the size is reduced by a factor of 4, the memory size is 11 K, which is still large.

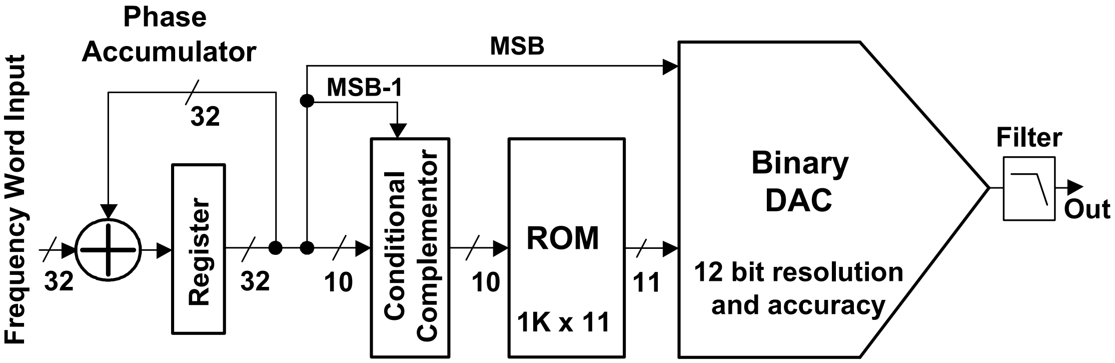


Figure 3 Diagram of the conventional DDFS using quarter-wave symmetry

To further reduce the look up table or memory size, various sine approximation methods have been proposed. A common technique used in sine approximation is that an initial guess, which can be easily computed from the directly available digital phase, is used as the approximation of the sine function. The difference between the initial guess and the sine function is stored in the memory. The dynamic range of these errors is smaller than that of the sine function and fewer bits are required for the memory.

One approach is the sine-phase difference, which uses a straight line as the initial guess [2]. The dynamic range is reduced by approximately 4.75, which saves 2 bits per memory sample. The double trigonometric approximation in [7] is a simple improvement to the sine-phase difference approximation. This technique adds a triangle wave derived from the phase information to the straight line in [2] to form an approximation. The 2-segment approximation is closer to the sine function than the one straight line approximation, saving 3 bits per memory sample. Langlois proposed a generalization to the two techniques [6]. Langlois' approach uses n straight lines to approximate the sine function and tries to exploit as much redundancy as possible between the value of a phase angle and its sine amplitude. A reduction of 4 bits per memory sample was achieved.

Other techniques that can also save 4 bits per sample include the parabolic approximation [4], Liao's technique [5] and quad line approximation [8]. Figure 4 (taken from [8]) summarizes the various sine approximation approaches and their associated approximation errors. Even with saving 4 bits per sample, the memory size is still large and 7K memory bits is required for the DDFS shown in Figure 3.

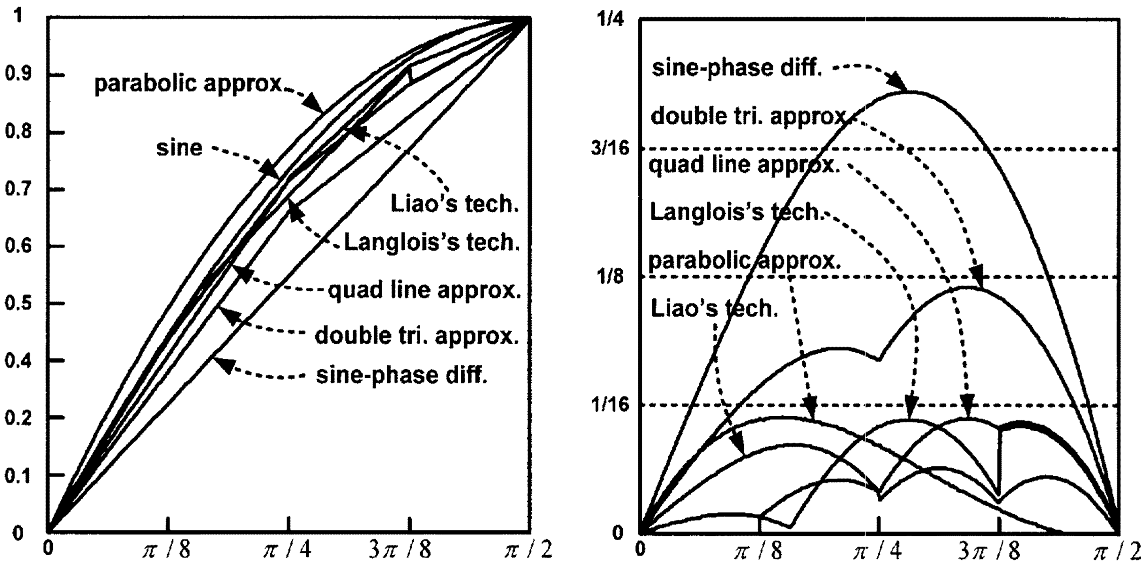


Figure 4 Sine approximation and approximation errors

The memory data can be further compressed by exploiting its redundancy and continuity. In [8], the approximation error data is further quantized by magnitude and address, and then the quantized values and their errors are stored in a separate ROM. This ROM can be much smaller. However, additional digital adders, complementors and even multiplier are used to reconstruct a digital sine wave before sending it to a DAC.

To achieve higher spectral purity, the digital signal must have increased number of bits to define the phase and amplitude. The approximation error should also be small, which may require more memory. Eventually, the DAC becomes the ultimate limiting factor and must be carefully designed.

The DAC generates spurs for a number of reasons. First, the quantization error causes harmonic outputs. Even with an ideal DAC, the actual amplitude may be a value that falls between two adjacent digital levels. However, spurs caused by quantization errors decrease by 6 dB/per bit and can be reduced by using higher resolution DACs. Second, glitches between DAC level transitions and the finite settling time results in harmonic frequencies. When the transition rate gets higher, the length of time after settling will get shorter while the length of time before settling is fixed and glitch energy dominates. The glitches will eventually corrupt the desired level and large spurious signals will occur. The last category for spur generation is RF issues such as timing, clock feed through, etc. All these issues must be dealt with carefully in design as well as layout to minimize the spur generation so as to improve spectral purity.

1.4 Proposed solution

A non-linear DAC approach for implementing high-speed high-spectral-purity DDFS is proposed. The non-linear DAC implements either a piecewise linear or a piecewise quadratic approximation to a sine function. Compared to the aforementioned works, using a non-linear DAC has many advantages.

First, a non-linear DAC enables a high memory compression ratio. For a 12-bit output DDFS, 2176 bits memory are required in [7]. Our non-linear DAC approach only needs 544 bits, which is another factor of 4 memory reduction.

Second, piecewise approximation has low approximation error when the segment number is large enough (i.e., 16 or more). The digital amplitude is more accurate so that high spectral purity can be achieved. The surveyed DDFSs with a built-in DAC have an SFDR between 50 and 65 dBc. Over 70 dBc is attainable with piecewise approximations.

Third, the non-linear DAC is capable to perform multiplication and summation in the analog domain. Shifting the computation from the digital to the analog domain simplifies the digital design and reduces power dissipation. Not only are digital multipliers and adders not needed, but they may also limit the clock rate, even with pipelining

The non-linear DAC incorporates many techniques to improve its performance. Digital calibration and common-centroid layout are used to improve the matching of the current sources in the current steering DAC. Control signal levels are carefully adjusted for the switching matrix to minimize glitches. Clock feed-through cancellation and transmission lines for clock distribution are used to improve timing and minimize performance degradation. In addition to these techniques, the piecewise quadratic design also includes top-level switch reduction and third harmonic cancellation techniques to further improve the performance. The DAC and memory are also designed for radiation tolerance.

CHAPTER TWO

2. PIECEWISE APPROXIMATION

Piecewise approximation uses a series of straight lines or curves to approximate a sine wave. This approach is very useful for high spectral purity DDFS designs. The piecewise approximation can be made accurate to within the resolution of the DAC so that no additional approximation error memory is required. However, this does not mean the memory can be totally eliminated. Certain memory is required to store other parameters used in the approximation. There are two version of piecewise approximations, namely piecewise linear approximation (PLA) and piecewise quadratic approximation (PQA). Both result in significant reduction of the memory. PQA is very suitable for higher resolution and higher spectral purity application since it uses fewer segments and less memory to achieve the same SFDR than the PLA approach. The following investigation focuses on the optimal segment numbers for a desired SFDR and memory size [11]. The result will ultimately determine an optimum design of a non-linear DAC to reach a desired SFDR.

2.1 Piecewise-linear approximation

In the PLA approach, the first 90 degrees of the sine function is broken into a number of equally spaced segments. Each segment is approximated by a straight line. PLA can be described as Equation (2).

$$\sin(x) \approx a_i + b_i(x - x_i) \qquad (i = 1 \ldots N) \tag{2}$$

where x is the phase of the sine wave and lies in the range of 0 to π/2, N is the number of segments, x_i is the starting phase of the i_{th} segment, and a_i and b_i are the offsets and gains of the i_{th} segment. Figure 5 illustrates this concept with a 4-segment PLA having a 12-bit DAC resolution. The smooth blue line is the ideal sine wave and the 4 straight red lines represent the approximation. The figure on the right side shows the zoomed in view of one segment. The difference between those two lines is the approximation error. Intuitively, the approximation error can be reduced by increasing the number of segments.

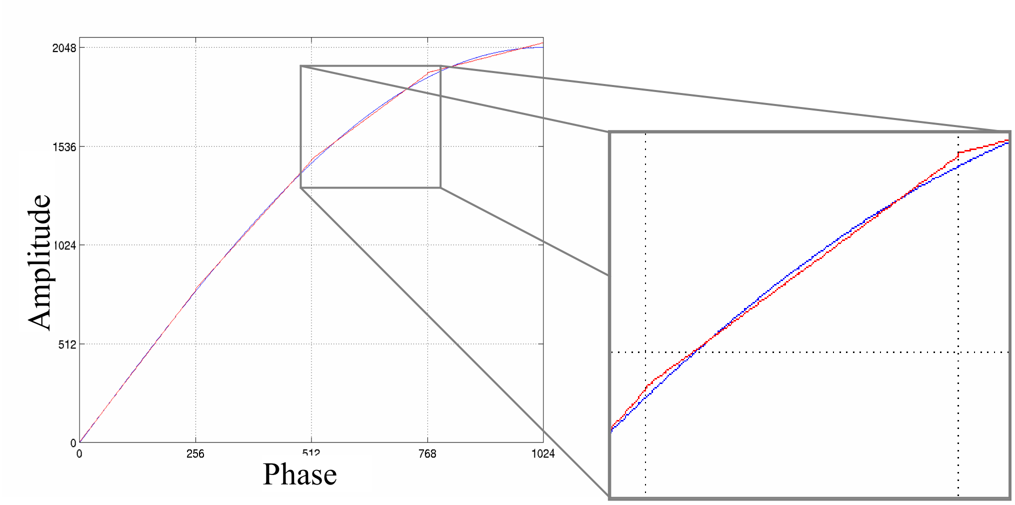


Figure 5 4-segment piecewise linear approximation

To implement the PLA, the two terms in the right hand side of Equation (2) need to be determined and summed. The first term is the segment offset. The offset comes from a ROM look-up table. The number of bits for each segment offset matches the resolution of the DAC. For example, for a 12-bit DAC resolution, 11 bits are required for the segment offset. The MSB needs not be stored since it is the sign bit and comes from the MSB of the digital phase. The resolution for the phase also matches the resolution of the DAC, i.e. the digital amplitude and the phase of one cycle of the sine wave have the same digital resolution. The second term is the multiplication of the segment phase offset and the gain. The segment gain defines the slope of the straight line in the segment. The number of bits for segment gains should be large enough to cover all possible slopes for different segments.

Note that the second term only covers the difference between offsets of the adjacent segments. The dynamic range is small and can be determined by the largest offset difference. The largest offset difference always happens at the first two segments. When using 32 segments in the first quadrant, the segment phase offset has 5 bits and the gain coefficient has 7 bits for an implementation of a 12-bit phase and DAC resolution. The largest offset difference is within 128 LSBs. The result of multiplication is a 12-bit number. When rounding to a full LSB, only the top 7 bits are significant.

The memory size is small because only the offset and gain coefficients are stored. Memory size depends on the number of segments. It decreases as the segment number is reduced. But fewer segments result in larger residual errors between the sine function and the approximation, leading to larger harmonics and lower SFDR. The relationship between SFDR and the number of segments for a 12-bit, 14-bit and 16-bit DAC resolution is shown in Figure 6, assuming ideal DACs. The “direct” approach means a binary DAC without piecewise approximation is used and only the quantization errors are present. The direct approach sets the upper limit on the achievable SFDRs.

It can be seen from Figure 6 that the number of segments limits SFDR when the number is small. The DAC resolution limits SFDR when there are an adequate number of segments. We also noticed that it is not always beneficial to increase the number of segments. For a given DAC resolution, there is an optimum number for segments to achieve a maximal SFDR. Further increasing the segments beyond this number will not improve the SFDR and may even reduce it.

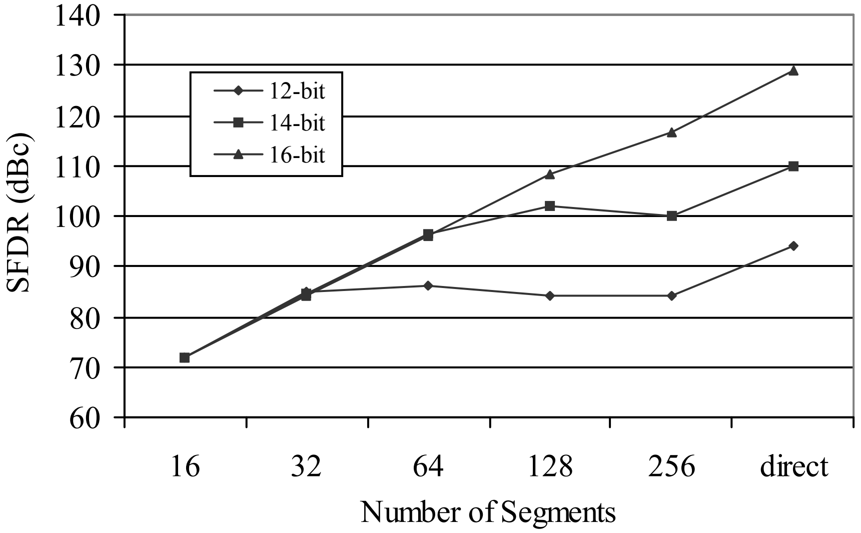


Figure 6 SFDR of piecewise-linear approximation

To explain this phenomenon, we need to start from the implementation of the second term of Equation (2), which is the multiplication of gain coefficients and segment offsets. The gain coefficients and segment phase offsets are digital words and have limited number of bits. The approximation error in a segment is directly related to how close the gain coefficient matches the actual slope of the sine wave within the segment. When the number of segments is small, the difference between the sine and linear function is large no matter how well the slope matches. As we increase the number of segments, the approximation error reduces and SFDR increases. However, the difference between the adjacent segments also reduces as more segments are used. Hence, the number of bits for the gain coefficients reduces if we keep resolution of the second term the same as that for the offset DAC, namely 1 LSB. As a result, there are less available gains from which to choose. But as the number of segments increases and each segment has its own slope, more possible gains are required. Therefore, at certain number, the available gains are not enough to provide the required gain for some segment. It results in larger approximation error and lower SFDR. The optimal number for the 12-bit and 14-bit DAC is 64 and 128, respectively. Figure 6 does not show the optimal number for 16-bit resolution, which may be 256 or higher.

Figure 7 shows the required memory sizes for 12-bit, 14-bit and 16-bit resolutions as a function of the number of segments. These results assume that the phase resolution for a complete cycle is equal to the resolution of the DAC. For comparison, the required memory size using only quarter wave symmetry for the direct case at 12, 14 and 16-bit DAC resolution is 11264, 53248 and 245,760 bits respectively. The memory size increases as the number of segment increases. For the same number of segments, higher resolution requires more bits for the gain coefficients and results in larger memory. Choosing an appropriate number for segments for a certain design requires a compromise between SFDR and memory size.

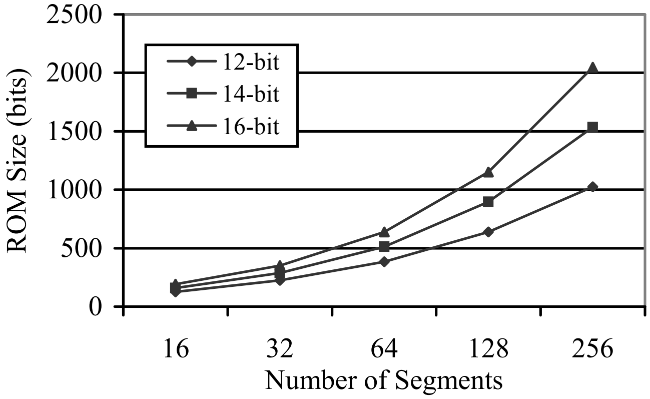


Figure 7 The required memory size of piecewise-linear approximation

For a 12-bit DAC, an SFDR of 85 dB is obtained with a 32-segment approximation and a memory size of only 192 bits (not including the 352 bits for segment offsets). Figure 8 shows the approximation error at each point. The maximum error is 1.34 LSBs. The direct approach has an SFDR of 94 dBc but requires 50 times more memory.

Increasing the number of segments to 64, only improves the SFDR by 1 dB but the required memory is almost doubled.

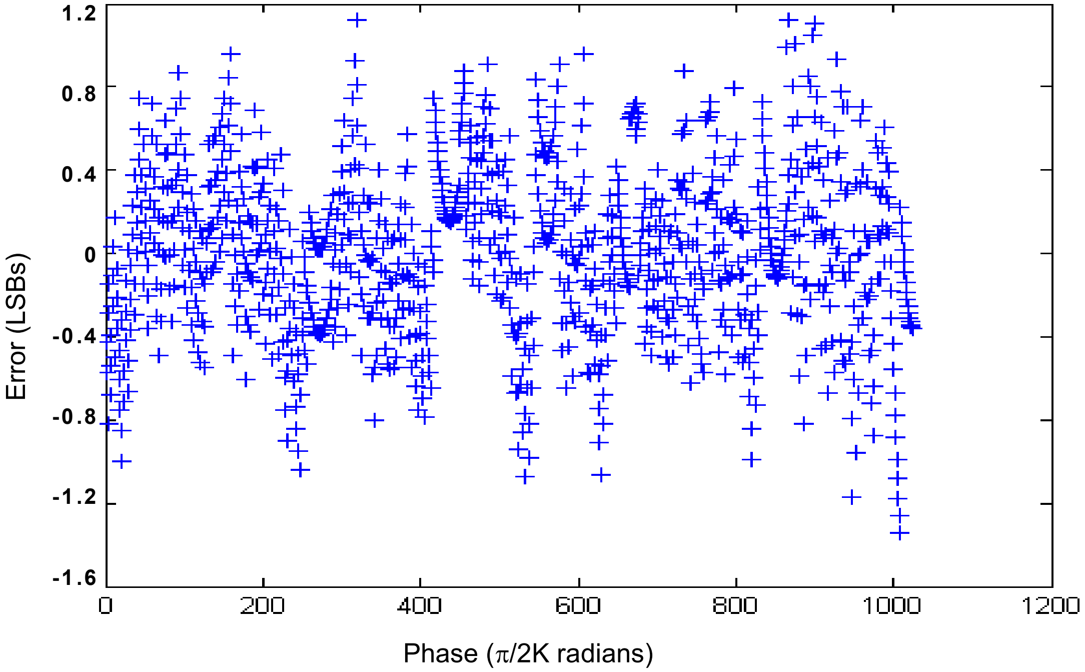


Figure 8 Approximation error of 32-segment PLA

2.2 Piecewise-quadratic approximation

The piecewise-quadratic approximation (PQA) method was investigated to reduce the memory size even more for high-resolution high-SFDR applications. The PQA can be obtained by adding a quadratic term to the PLA equation, as shown in Eq. (3).

$$\sin(x) \approx a_i + b_i(x - x_i) + c_i(x - x_i)^2 \quad (i = 1 \ldots N) \tag{3}$$

where x is the phase of the sine wave and lies in the range of 0 to π/2, N is the number of segments, x_i is the starting phase of the i_{th} segment, and $a_{i,}$ b_i and c_i are the offsets, linear gain coefficients and quadratic coefficients of the i_{th} segment.

Figure 9 shows a 4-segment PQA with 12-bit digital resolution. If compared with Figure 5, the PQA has smaller approximation error than the PLA for the same number of segments. Adding in the quadratic term significantly reduces the approximation errors.

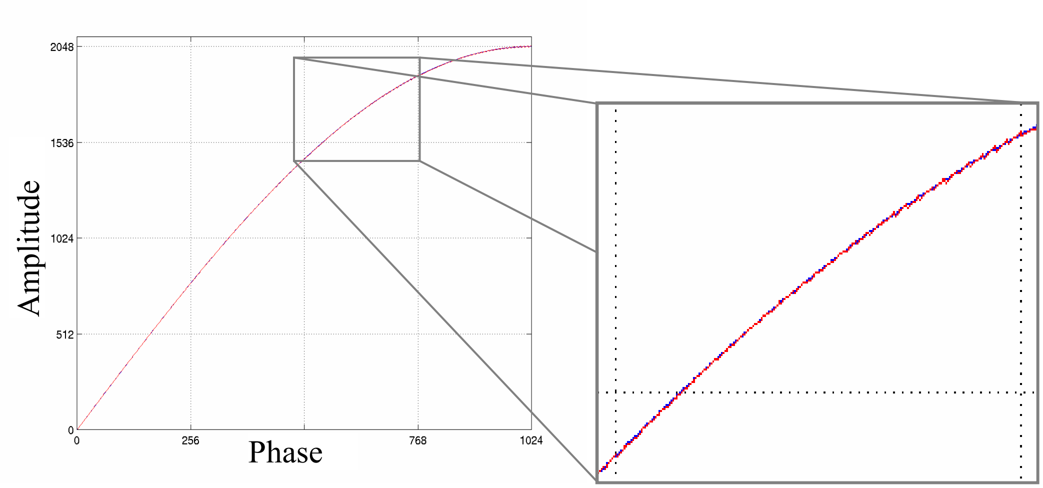


Figure 9 4-segment piecewise quadratic approximation

In the PLA approach, the number of segments must be increased to get high SFDR for high resolution DACs. The memory size nearly doubles as the number of segments doubles. With a quadratic term, the number of segments can remain low. Additional memory is required to store the small quadratic coefficients and this is less than the memory needed for a PLA with twice as many segments. Simulation results for the piecewise-quadratic approximation are shown in Figure 10.

In general, PQA approach results in better SFDR than the PLA method, especially when the segment number is low. For 16 segments, the SFDR for 12, 14 and 16-bit resolution increases by 11 dB, 22 dB and 36 dB, respectively, compared to the PLA method with the same number of segments. The SFDR curves have similar pattern as in the PLA approach in terms of their dependence on segment number. The SFDRs of the 12-bit and 14-bit resolution DDFSs have a peak at 64 and 128 segments, respectively and drops thereafter. With fewer segments required to achieve a given SFDR, the memory size can be reduced significantly when using PQA. However, the reduction in memory size comes at the cost of evaluating the third term cx^2 in Eq. (3). The linear term in the PLA/PQA is formed using an analog multiplying DAC instead of a digital multiplier to achieve high speed operation.

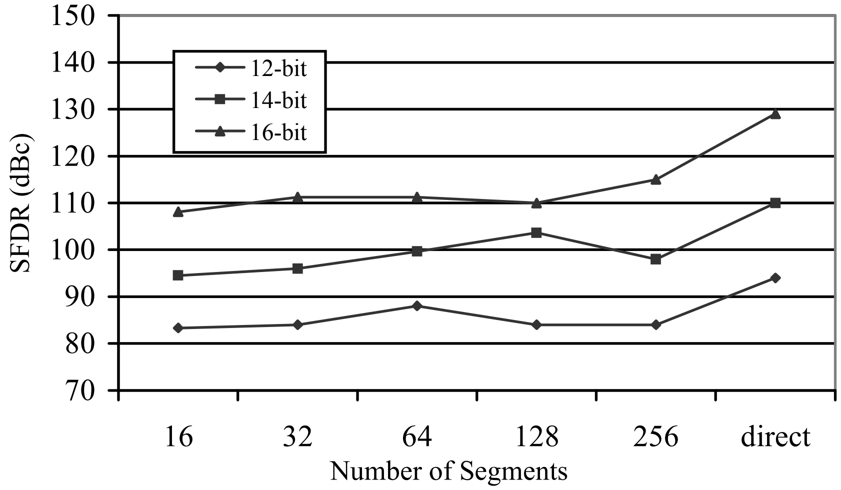


Figure 10 SFDR of piecewise-quadratic approximation

The third term is evaluated by first determining x^2 digitally and then using a multiplying DAC to form the product of c and x^2. If only the top 4 or 5 significant bits of x are used, x^2 can be implemented with only 3 to 4 levels of logic and has much lower complexity than a digital multiplier.

A 14-bit DAC with 16-segments can achieve an SFDR of 97 dBc if 7 significant bits in x are maintained. A 7-bit squaring circuit adds significant complexity however. Maintaining only 5 significant bits reduces the SFDR to 94 dBc but only a small 5-bit squaring circuit is needed. The required memory size is 256 bits, which is half the memory required for the 14-bit 64-segment PLA with an SFDR of 96 dBc. With a 32 segment PQA, an SFDR of 98 dBc can be achieved with 4 significant bits and a memory size of 416 bits. The 4-bit squaring circuit is smaller than the 5-bit circuit. The memory size is larger than the 16-segment 5-bit squaring case but is still 20% smaller than the 14-bit 64-segment PLA memory size. The approximation error for the 16-segment PQA is shown in Figure 11. The maximum approximation error is 1.9 LSBs and the SFDR is 97 dBc.

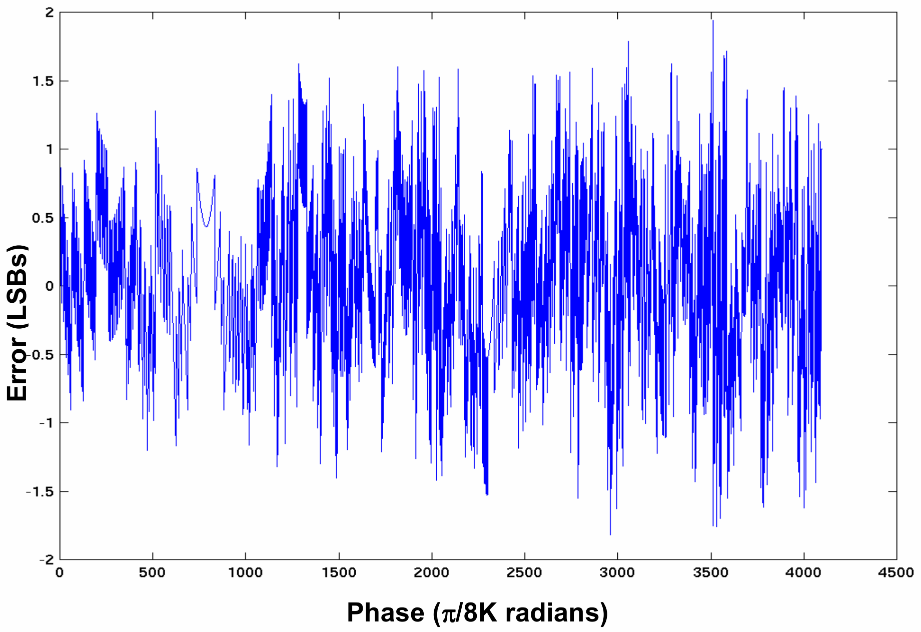


Figure 11 16-segment PQA approximation error

2.3 Conclusion

PLA and PQA approaches can significantly reduce the required memory lookup table size for DDFS systems. Compared with the existing memory compression methods, it provides the highest memory compression ratio. Instead of storing the amplitude or approximation errors for each sampling point, PLA and PQA methods only store the offset and gain coefficients for each segment in the memory table, resulting in a significant memory reduction. Compared with the PLA method, PQA is more suitable for high-resolution high-SFDR DDFS implementations. The PQA method saves more memory and has smaller approximation error.

PLA and PQA can be implemented using non-linear DACs. The number of segments is determined from the DAC resolution, the desired SFDR and the implementation complexity. Tradeoffs among those parameters must be carefully considered. The implementation of a 32-segment PLA using a 12-bit non-linear DAC is described in Chapter 3. The SFDR is anticipated to be over 60 dBc. The design of a 16-segment PQA is described in Chapter 4 using a 14-bit non-linear DAC. The SFDR is targeted to be over 80 dBc.

CHAPTER THREE

3. 12-BIT NON-LINEAR DAC

A 12-bit non-linear DAC is designed to implement a 32-segment piecewise linear approximation to the sine function. This implementation of a non-linear DAC allows the memory look-up table to be reduced from 11K to 544 bits. The DAC is implemented using a hybrid current steering architecture. It also requires relatively few switches, increasing its potential for high-speed operation. The non-linear DAC architecture in [9] requires 2^N switches where N is the resolution in bits and limits high speed operation at higher resolution. The reduction in memory size requires no digital adders or multipliers, which increase complexity and power dissipation, as in several other approaches. The chip was fabricated in a 0.35 μm SOI CMOS process. Test results show that the DAC has 12-bit accuracy, operates up to 600 MSps and provides differential outputs of 0.5 Vpp into 50 Ω loads. The SFDR is up to 72 dBc. Radiation test shows the non-linear DAC can tolerate a total ionizing dose of 200 Krad Si.

3.1 Non-linear DAC

A block diagram of the DAC and its associated control logic are shown in Figure 12. Advantage is taken of the symmetry properties of the sine function to reduce memory size using the sign of the phase and the conditional complementor, as is common in many designs. The ROM look-up table stores the segment offsets and gains. The MSB of the digital phase connects directly to the DAC. The remaining 11 bits DAC inputs come from the ROM look up table. The ROM table also outputs 6-bit gain coefficients for each segment. The gain multiplies the segment phase offset, which is the lower 5 bits of the digital phase. The result of the multiplication is summed with the offset and provides the final output. The second MSB controls a conditional complementor. Together with the MSB, which control the sign of the output, a full cycle of sine wave can be generated.

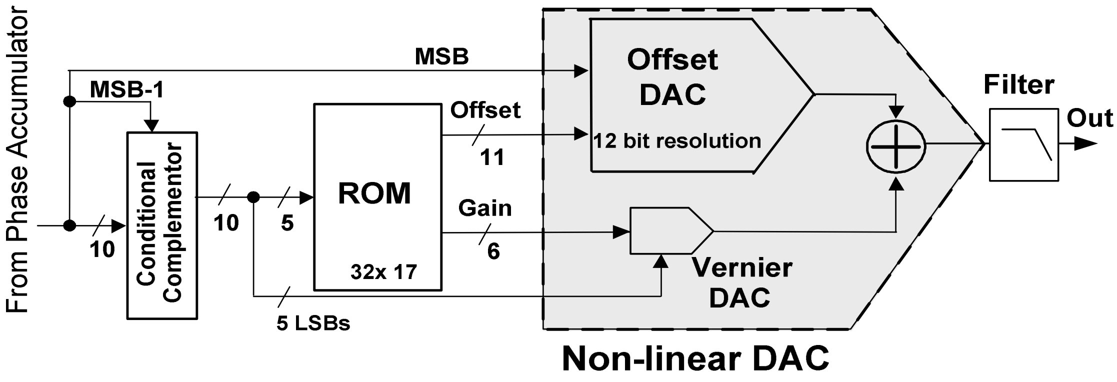


Figure 12 A 12-bit Non-linear DAC with control logic

The 12-bit non-linear DAC consists of two DACs; an offset DAC and a vernier DAC. Each DAC implements one term in Equation (2). Since the DACs are designed using current steering architecture, the addition of the two terms is performed by summing the current at the output node. No digital adder is required. The multiplication in the second term of Equation (2) is performed by summing currents corresponding to partial products instead of using a digital multiplier, which would impact the high-speed operation.

3.2 Offset DAC

The offset DAC is implemented as a 12-bit hybrid current-steering DAC. The top 5 MSBs are implemented using unary current sources and the 7 LSBs by 7 binary-weighted current sources. This hybrid architecture reduces the amplitude of glitches at the output [10]. The glitch is usually proportional to the size of the current sources. When using unary current sources for the top 5 MSBs, the largest current source is limited to 128 LSBs, which is the size of one unary current source. There are 31 such current sources controlled by the top 5 MSBs through a thermometer code decoder. Converting the binary codes to thermometer codes eliminates the possibility of turning on some current sources and turning off other current sources at the same time during code transitions, which is a major cause of glitches. The 7 binary weighted current sources do contribute to glitches but the total of the 7 binary currents is 1 LSB less than one unary current. Thus, the maximum glitch amplitude is limited to 128 LSBs. If the DAC is implemented using all binary weighted current sources, the maximum glitch would be 2048 LSBs, which is 16 times larger than that with the current implementation.

Reducing the size of unary current sources can reduce the glitches further since the glitch amplitude is directly related to the unary current. If the top 6 MSBs are implemented by unary current sources, the size of unary current source can be halved. However, the number of top level current switches will be doubled. More top level current switches not only add more capacitive loading at the output node, but also lower the output impedance. High output impedance is necessary to achieve high SFDR [20].

To compensate for the current source matching error during the fabrication process, digital calibration and trimming is used. All unary current sources and the largest of the binary-weighted current sources are digitally trimmed. Figure 13 shows a unary current source along with its digital trimming circuit. Each unary current source is implemented with a current source that is 4 LSB lower than the ideal value of 128 LSBs. *Tbias* controls 6 binary weighted PFET current sources with the smallest one being 0.125 LSB. The 6 PFET current sources connect to a current mirror through switches controlled by TB_0~TB_5. The current mirror can add an extra current of up to 8 LSBs with a resolution of 0.125 LSBs. Thus the total current can be adjusted by ±4 LSBs at a resolution of 1/8

LSB around 128 LSBs. Each trimmable current source is associated with a 6-bit control word (*TB*$_0$~*TB*$_5$), which is stored in RAM.

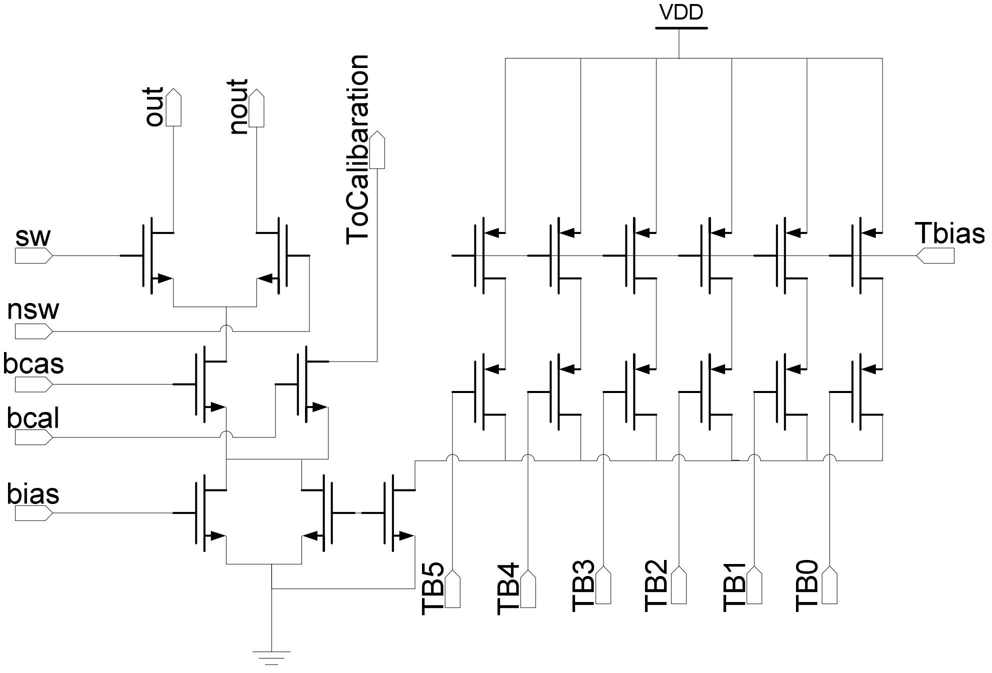


Figure 13 A current source with trimming circuit

Control signals *bcas* is the bias to the cascode FET. Under normal condition, signal *bcal* is low and all current flows through the cascode FET to output. The current switch signals *sw* and *nsw* direct all of the current to one of the output nodes. Since the current must flow to one of the output nodes, the current steering DAC inherently has differential outputs. When the current source is under calibration, signal *bcal* becomes high and the source node of the cascode FET is raised high enough to cause the cascode to be turned off. All current is switched to the calibration circuit, which contains mainly a comparator. The comparator compares the current under calibration with a reference current.

The reference current source used to calibrate the unary current sources is almost identical to the unary current sources except its current does not flow to the output nodes.

To ensure the unary current sources match the binary ones after calibration, the reference current source must be calibrated to a value that equals the sum of all binary current sources plus 1 LSB (i.e., 128 LSBs). Before calibrating the reference current source, the largest binary current source must be first calibrated to 64 LSBs using the lower 6 binary weighted current sources plus a 1 LSB current source. Therefore, three steps are required to calibrate the DAC. First, the largest binary current source is calibrated using other binary current sources and an additional 1 LSB current source. Second, the reference is calibrated using all binary current sources and the one LSB current source. Finally, all the unary current sources are calibrated to the reference.

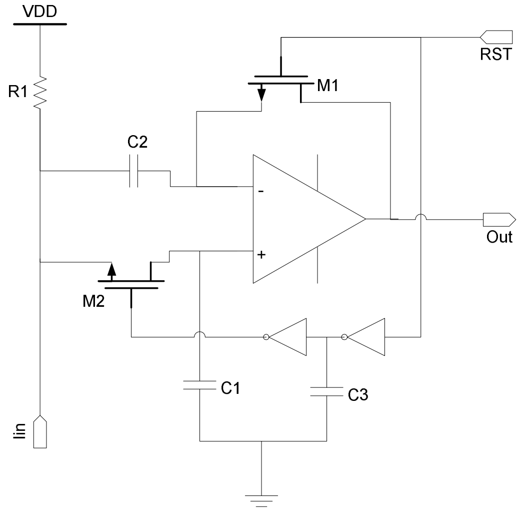


Figure 14 A high precision comparator

The calibration process involves comparing whether two currents are equal. Since one LSB current is 5 nA in this design, a resolution of 0.125 LSB requires the comparator to have the ability to differentiate a current difference of 0.625 nA. If using a 2.5 kΩ resistor to convert the current to a voltage, a comparator resolution of 1.6 mV is required. A larger resistor would reduce the comparator resolution requirements but would not allow adequate voltage to remain across the current sources. The comparator gain must be at least 80 dB to provide a full output swing of 3.3V. Figure 14 shows the high precision comparator featuring DC offset and charge injection cancellation.

The comparison operation requires two steps. First, RST signal holds high while the reference current I_{ref} is directed to I_{in}. Transistors M1 and M2 are turned on. The comparator is in reset and sampling mode. The reference voltage is sampled and stored on C1. The DC offset of the opamp is stored on C2. Then, RST signal goes low turning M1 off. After a short delay provided by two invertors and the capacitor C3, M2 turns off. Charge injection from M1 and M2 will affect the voltage of the opamp input nodes since those nodes are at high resistance. The charge injected from M1 and M2 when they turn off is equal since M1 and M2 are the same size. C1 and C2 are equal size too. Thus the injected charge will cause the same potential change at both opamp input nodes and will not affect the comparison result. After M2 turns off, the current being calibrated is directed to the input node I_{in} and the comparator output will change to either high or low, depending on whether I_{in} is greater than or less than I_{ref}.

The current under calibration can be adjusted by writing different numbers to the RAM associated with the current source. A 6-bit ripple counter provides the input to the RAM. The RAM outputs are always connected to the PFET current switch that control the amount of current added to the main current source. Initially, the counter is reset to 0. The maximum amount of current is mirrored to the main current source. The current under calibration is guaranteed to be larger than the reference current and the comparator output is high. As the counter steps through its 64 values, the current gradually decreases. When the current is slightly less than the reference current, the comparator changes its output state to low. At this time, the counter stops counting and its value is stored in the RAM. Considering comparator metastability, the counter must operate at low. The low speed operation also relaxes the bandwidth requirement for the opamp.

The comparator only uses a single resistor to convert the reference current and the calibrated current to voltage. Hence, there is not a resistor mismatch problem, which would cause errors if two resistors are used. A current mode opamp is used for the comparator. The low impedance on the internal nodes limit the voltage changes on those nodes even a large current change has occurred. This minimizes the interference to the opamp input nodes from the voltage fluctuation on the drain of the input FET. A traditional two-stage opamp was initially used, but the large voltage fluctuation on the drain node of the input FET coupled to the gate. The magnitude of the coupled signal caused errors when the input signals to be compared are very close to each other.

The performance of a current steering DAC at high speed is highly related to the impedance at the output node of the current sources [20]. To achieve high output impedance, a cascode FET is added to each current source. The current switches are also designed to operate in the saturation region. When the switch is ON, it acts as a second cascode FET and increases the output impedance. Moreover, using small size current switches helps to improve performance. The small switch has less gate-to-drain capacitance and reduces clock feed through from gate to the drain node. It also has less capacitive loading on the output node and its source nodes. The capacitance on the source node also has negative effects on SFDR.

FETs are square law devices in the saturation region. The control signals of the current switches must be carefully designed so that the source node of the current switches has minimum fluctuation. The fluctuation at those nodes degrades the overall SFDR. To minimize the fluctuation, the differential control signals not only need to have a reduced swing but also need to cross at a point that is 0.707 of the full signal swing. This is achieved by a high speed latch with adjustable cross point. The latch is shown in Figure 15 and is modified from [10].

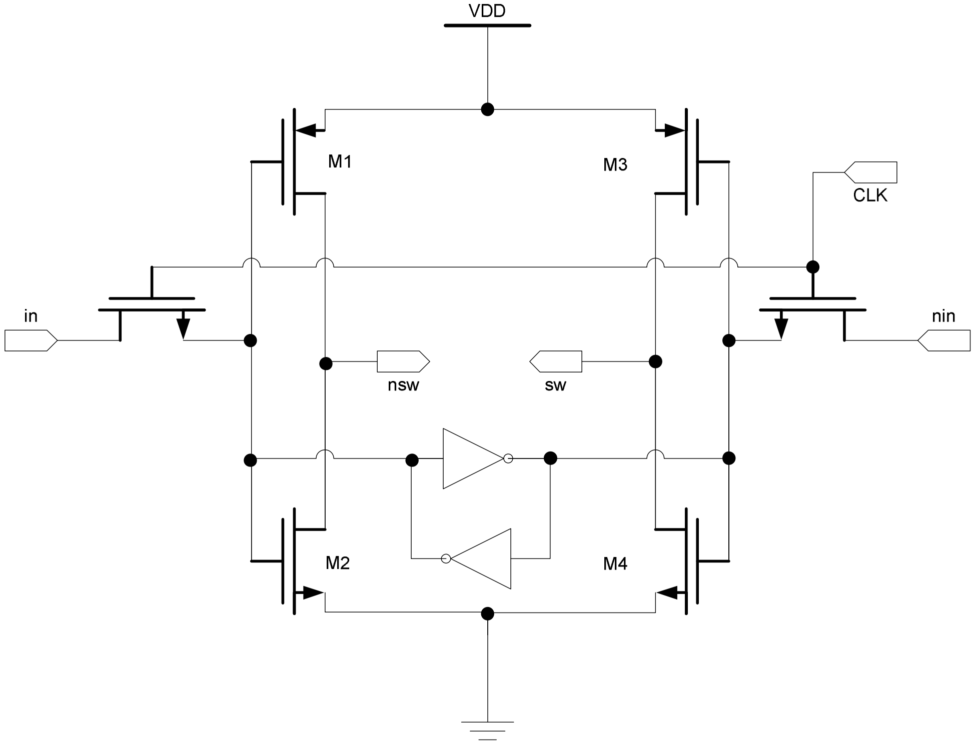


Figure 15 High speed latch with a single phase clock

The latch has differential inputs and outputs. The structure is symmetrical and only uses a single phase clock. Using a single clock eliminates timing errors from clock mismatch. The output cross point can be adjusted by sizing M1 and M2. Because of the symmetry property, M3 and M4 always take the same size as M1 and M2, respectively. Two cross coupled weak invertors make sure the latch keep its state when the input switches are off.

3.3 Vernier DAC

The offset DAC is a regular 12-bit DAC, which provides accurate segment offsets. The vernier DAC implements the second term in Equation (2) that involves multiplication. The procedure of binary multiplication is actually the summation of weighted partial products. Figure 16 shows the partial products when multiplying two 5-bit binary numbers A (A5A4A3A2A1) and B (B5B4B3B2B1).

					A5	**A4**	**A3**	**A2**	**A1**
					B5	**B4**	**B3**	**B2**	**B1**
					A5B1	**A4B1**	**A3B1**	**A2B1**	**A1B1**
			+	**A5B2**	**A4B2**	**A3B2**	**A2B2**	**A1B2**	
		+	**A5B3**	**A4B3**	**A3B3**	**A2B3**	**A1B3**		
	+	**A5B4**	**A4B4**	**A3B4**	**A2B4**	**A1B4**			
+	**A5B5**	**A4B5**	**A3B5**	**A2B5**	**A1B5**				

Figure 16 Binary multiplication

Each partial product is one bit, either 1 or 0, and can be obtained by multiplying the corresponding bits in A and B. Partial products in the same column have the same weight. The column on the left has twice the weight as the column immediately to its right. The analog implementation of binary multiplication can be achieved by using binary weighted current sources controlled by the corresponding partial product bits having the same weight. The summation of weighted partial products is done by summing the analog currents. This implementation avoids a digital multiplier, which adds complexity, increases power dissipation and may slow down the system operation speed. The only complexity introduced with the analog approach is that more current sources are required.

The vernier DAC is implemented by a multiplying DAC. It consists 7 sets of binary-weighted current sources with values shown in Table 1. X_i (i=1 to 5) denotes the i_{th} bit of the phase offset within a segment and G_k (k=1 to 7) the k_{th} bit of the gain for the segment. These bits control 26 current sources with relative sizes given in the table. Current is switched to the output when both row and column control bits are high. Otherwise the current is switched to the complementary output. The total current switched to an output node is the product of the segment gain and the phase offset. In this implementation, the minimum current is 1 LSB and all bits weighting less than 0.5 LSB in the product are rounded off. This approach reduces the number of switches required and enables the DAC to operate at high speed.

Table 1 Multiplication by summing partial products

	X_s	X_5	X_4	X_3	X_2	X_1
G7	**62**	**32**	**16**	**8**	**4**	**2**
G_6	**31**	**16**	**8**	**4**	**2**	**1**
G_5	**16**	**8**	**4**	**2**	**1**	**1**
G_4	**8**	**4**	**2**	**1**	**1**	
G_3	**4**	**2**	**1**	**1**		
G_2	**2**	**1**	**1**			
G_1	**1**	**1**				

Note: values are in LSBs

According to Table 1, partial products corresponding to 0.5 LSB are always rounded up to 1 LSB and no 0.5 LSB current is used. No matter rounding up or down, keeping the output current at full LSBs ensures that the final non-linear DAC has 12 bits resolution. Because of two or more 0.5 LSB may be rounded up, this multiplier doe not perform an ideal multiplication. However, the segment offset and gain coefficients are chosen based on the actually vernier DAC implementation. The multiplication error has been taken into consideration when fitting the sine function. We choose rounding up the 0.5 LSB current because it gives slightly better SFDR in simulation than rounding down.

The offset DAC has differential outputs and adding in the vernier DAC may corrupt the differential outputs. For example, if the gain bits are all 0, the vernier DAC should not add any current to either of the output nodes. But the implementation of the vernier DAC described above would direct all its current to the complementary output, thus disrupting the differential outputs. Keeping the outputs truly differential is important because it helps to reduce the even order harmonics of the output wave.

The problem is fixed by adding an additional column of current sources in Table 1 under X_s. X_s denotes the sign bit. The size of current sources is special in that each of them is exactly the sum of the rest of the current sources in the same row. When a gain bit, e.g. *G6*, is 0, the current in this column next to *G6* is always switched to the output irrespective of the value of X_s. Since the rest of currents in row G6 are switched to the complementary output, equal amount of current is switched to both outputs. The differential outputs are maintained. When the gain bit is 1, the current is switched to output if X_s is 1 and to the complementary output if X_s is 0.

Adding the extra column of current sources and the sign bit also enables the subtraction operation if the origin is set at the level where all gain bits are 0. Given a gain, if the sign bit is 1, the segment phase offset $X_{5\dots}$ X_1 determines the amount of current to be added to the output node. By taking the 1's complement of X_s $X_{5\dots}$ X_1, the new control code will subtract the same amount of current from the output node. The ability to use 1's complement simplifies the digital logic design to generate the sine wave in quadrants other than the first.

3.4 ROM table and control logic

3.4.1 ROM Table

The memory lookup table stores the digital sine wave. For a 12-bit DAC, the standard approach of implementing a DDFS requires 11K bits of memory. When using the non-linear DAC, the memory only stores the offset and gain for each segment instead of each point of the sine function. The size of memory can be significantly reduced, depending on the number of segments and the number of bits assigned to the segment gain. The ROM look-up table is 544 bits with 11 bits for each offset and 6 bits for each gain.

The ROM is implemented by muxes controlled by the 5-bit address code. A 32:1 mux can be used to implement one block of ROM. Each block is 32 bits, i.e., one bit for each segment. The 32:1 mux can be implemented using 5 stages of 2:1 muxes as shown in Figure 17. The total number of the 2:1 muxes would be 31. However, the first stage can be totally removed by connecting the inputs to the second stage to either 1, 0, b6 or the inverse of b6 based on the actual ROM values. This method reduces the number of 2:1 muxes to 15 saving more than half the layout area. The inverse of b6 requires only an inverter or even no additional hardware if the register has differential outputs, which is nearly always the case. 11 such blocks are made for the offset bits and 6 for the gain coefficients, giving a total of 544 bits for the ROM table. Note that the actual gain coefficient has 7 bits. The relationship between gains and segment numbers is a monotonic function. The MSB of the gain bits is implemented using simple logic gates that are much less complex than one ROM block.

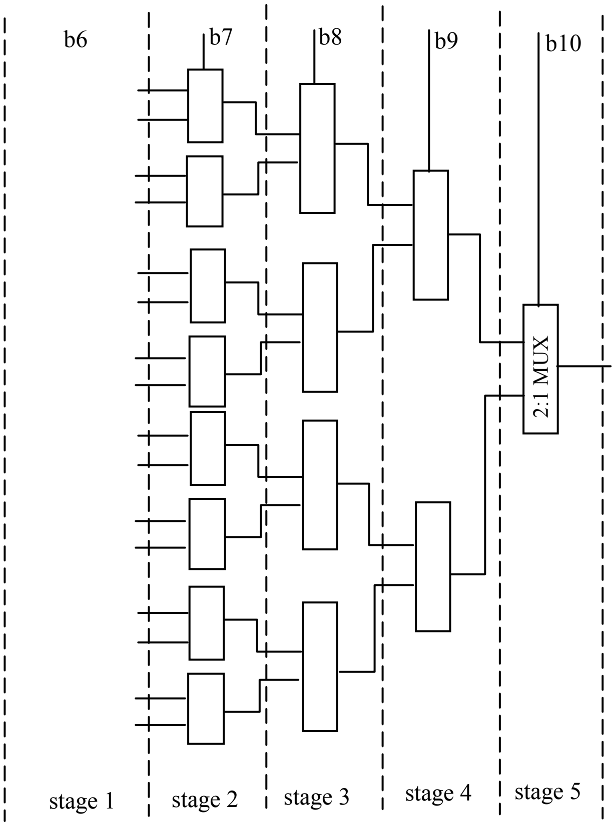


Figure 17 Block diagram of a ROM block

When determining the ROM values for offsets and gain coefficients, one must take into consideration the actual implementation of the vernier multiplying DAC. The multiplying DAC provides the difference between the desired output value and its segment offset. The implementation of the multiplying DAC must ensure that the largest difference between the adjacent segments is covered. This value is always the difference between the first two segments when approximating a sine wave and is about 100 LSBs for a 12-bit DAC. The implementation should also have enough possible gains to cover all the slopes or gains for different segments. An ideal multiplication will generate values with products less than 1 LSB. Rounding up to a full LSB is performed to be consistent with the DAC resolution. In Table 1, the minimum current is 1 LSB. The result of multiplication only produces integer numbers. The rounding errors have limited effect on the final SFDR. To determine the optimal offset and gain coefficients for the 32 segments in the first quadrant, the following algorithm is used:

i) Set current segment =1

ii) Computer the ideal offset and gain for the current segment. Quantize the offset to 11 bits and gain coefficient to 7 bits.

iii) Let offset vary from (offset-2) to (offset+2) and gain vary from (gain-2) to (gain+2)

iv) For each set of offset and gain coefficient, calculate the errors between the ideal sine wave and the regenerated one based on Table 1 for the current segment.

v) For each set of offset and gain coefficient, calculate the sum of the squared errors. Pick the set of offset and gain coefficient that give the minimum error.

vi) repeat (ii)-(v) for segment 2 to 32.

Although the vernier DAC does not perform an ideal multiplication, this algorithm ensures the offset and gain bits in use to provide a best approximation given the current multiplying DAC implementation.

3.4.2 Thermometer Code Decoder

The offset DAC requires a thermometer code decoder for the unary current sources. The decoder converts the top 5 binary encoded MSBs to 31-bit thermometer code. Starting from the first bit, the 31-bit code always contains consecutive 1s whose number equals the value of the input binary code. The schematic of the thermometer decoder is shown in Figure 18.

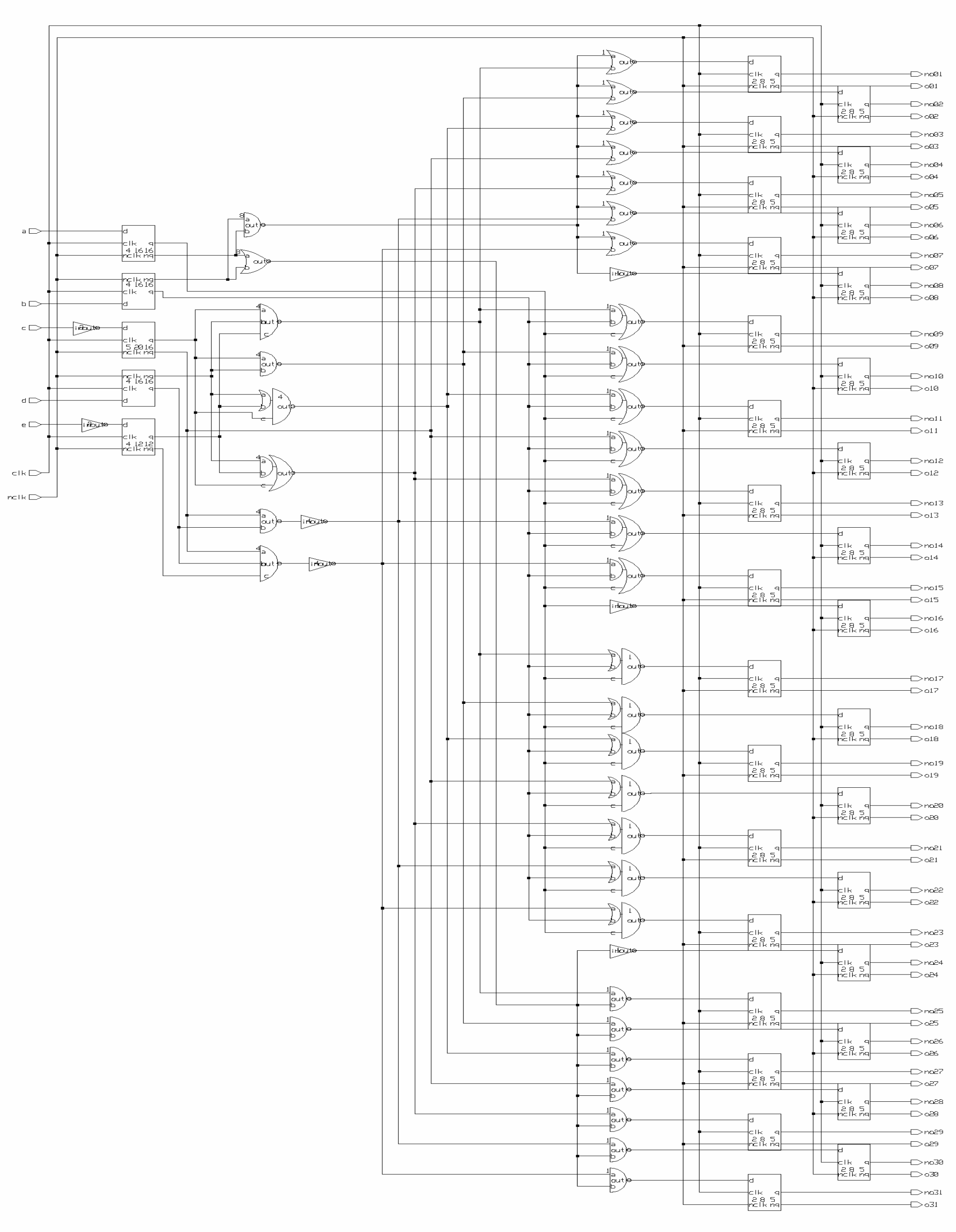


Figure 18 Schematic of the thermometer code decoder

The thermometer code decoder is organized as a two stage pipeline for high speed operation. Output latches provide differential outputs as required by the subsequent circuits. The 31 outputs control the current switches of the 31 unary current sources.

3.4.3 Control Logic Summary

The overall digital control logic is shown in Figure 19.

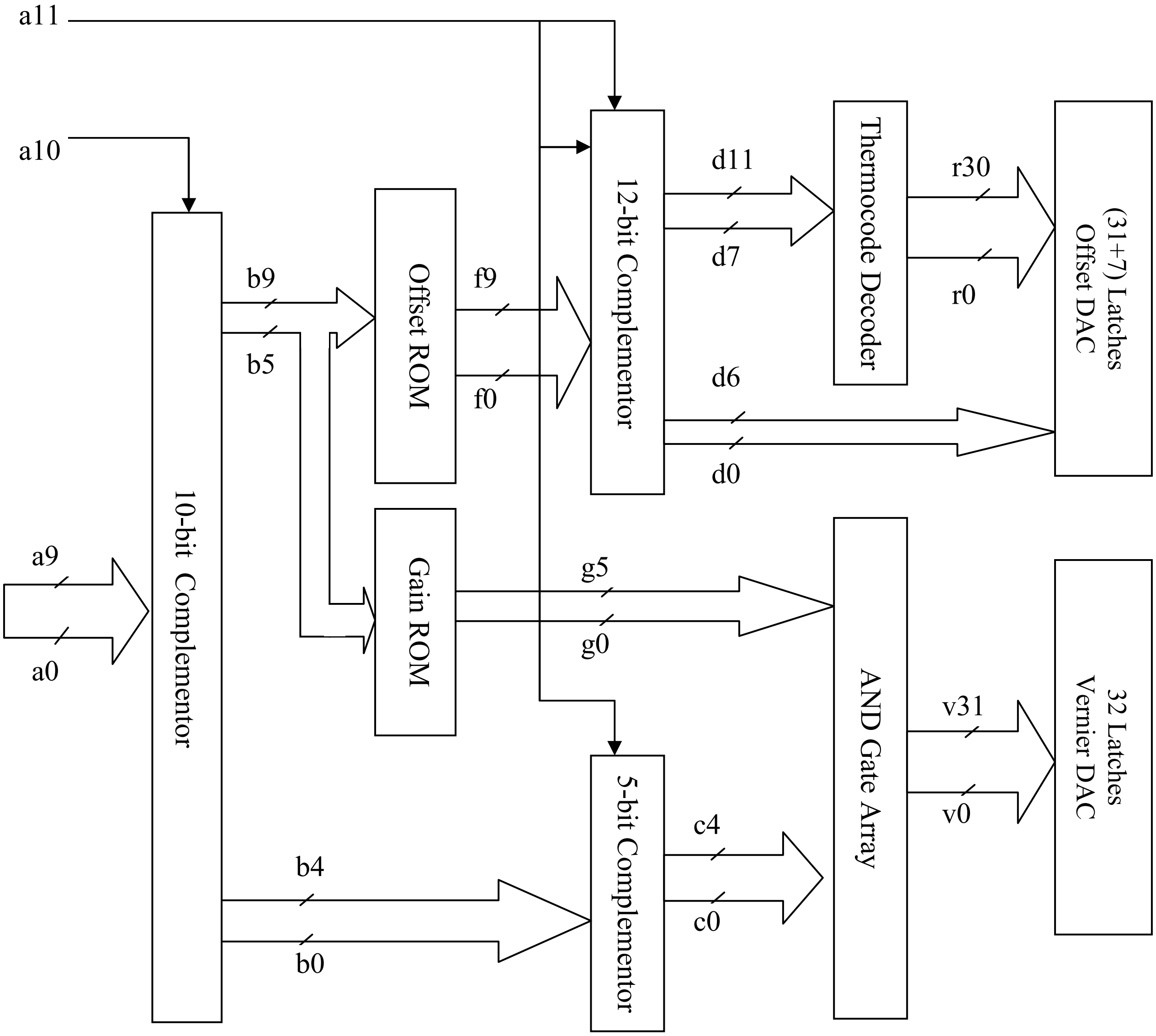


Figure 19 Block diagram of the control logic

The top two MSBs of the 12 bit phase input control several complementors, which are necessary to generate the sine wave in quadrants other than the first. Complementors are implemented using 2:1 muxes. When the control bit is 0, the output is the same as input. If the control bit is 1, the output is the inverse of the input.

The top 5 output bits from the 10-bit complementor are the segment number. They are used to address the ROM look-up table. The offset ROM and gain ROM output the coefficients corresponding to the current segment. The offset bits go through another complementor then the top 5 MSB outputs are fed into the thermometer decoder. The lower 7 LSBs, together with the 31 decoder outputs, control the current switches in the offset DAC.

The lower 5 output bits from the 10-bit complementor are segment offsets. They are first input to another complementor (5-bit) and then, together with the gain coefficients, are fed into the AND-gate array. This array has 32 outputs, which are the control signals for the current switches of the vernier DAC.

There are flip-flops that pipeline the progress of signals. The flip-flops are not shown in Figure 19. They are used to synchronize the digital signals. Furthermore, dividing the complex logic into pipeline stages enables high speed operation.

3.5 Layout

The design is laid out as compact as possible for fabrication. The digital and analog circuits use different layout strategies. A library of standard digital cells is created before using tools to automatically place and route the cells for each digital block.

Interconnections between the digital blocks are routed manually. The analog blocks and the critical digital cells such as the high-speed latch are laid out manually.

The high speed latches directly control the current switches. Any timing error caused by the different delay from its inputs to outputs may result in glitches at the output. The latch has differential inputs and outputs and is highly symmetrical in design as well as in layout. Figure 20 shows the layout of the high speed latch. The left half is symmetrical to its right half. The clock input is in the middle. The FETs and input/output ports are also placed symmetrically. This layout ensures the same delay from the input to the output for the differential control signals.

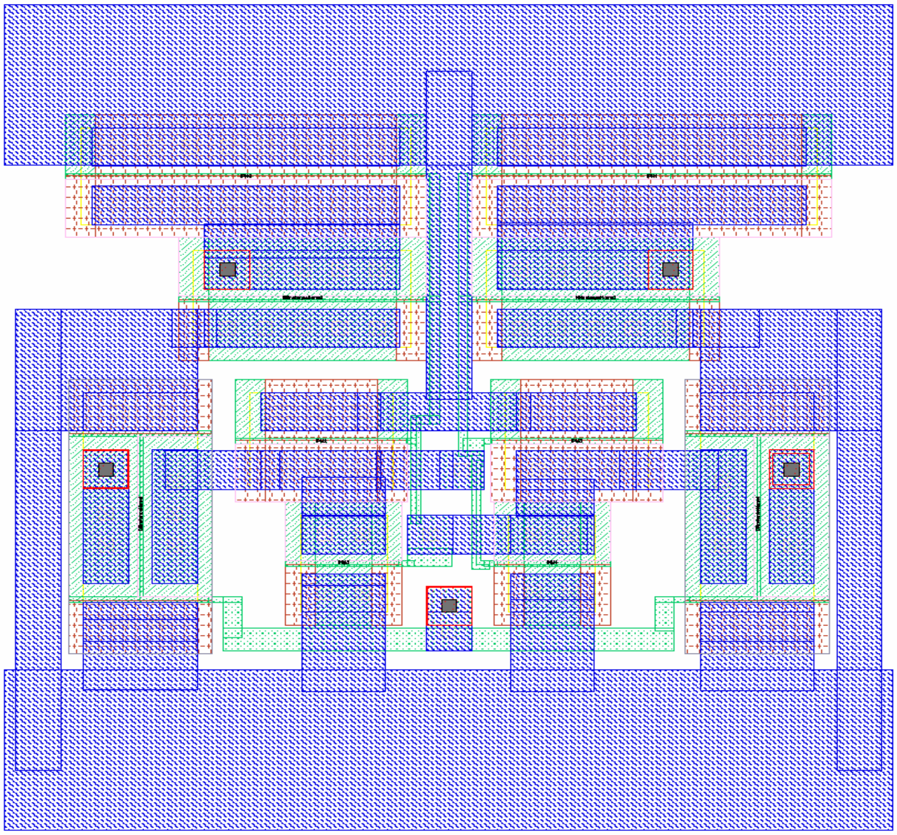


Figure 20 Layout of the high speed latch

The current switches controlled by the high-speed latch are also symmetrical in layout. When they are connected together, the differential outputs of the latch have equal delay to the current switches' gate. The current switches are placed in one row and the high speed latches are another row above of the switches.

All currents, including those for the offset DAC, vernier DAC and the reference, are implemented using 2-LSB current sources except for the 1 LSB and 2 LSB current. The 1 and 2 LSB currents are implemented using 0.5 LSB current sources. Although digital calibration is used to compensate the matching errors, common centroid layout strategy is still applied. Common centroid layout is very common in current steering DACs to improve the matching of the current sources [10]. The 0.5-LSB current sources are placed in the center of the 2-LSB current source array. At least two current sources must be used to independence to gradients introduced the gradient induced in fabrication. The 2 LSB current is obtained by combining 4 0.5-LSB current sources instead of using a single 2-LSB current source. The transistor size for each 2-LSB current source is reasonably large to improve matching [21].

When picking the small current sources to implement a larger one, one must make sure the small current sources to have a common center: the center of the current source array. The best choice is that the small current sources are symmetrical vertically, horizontally and diagonally. Since the 7 binary weighted currents for LSBs are not digitally trimmable, the current sources for them are picked closer to the center. The trimmable current sources are 4 LSBs less than the nominal current. Hence, one set of 2-LSB current sources are removed from the upper left and bottom right corner.

The main current sources are implemented by NFETs. The trimming current that is added to the main current is implemented by PFET current sources. Each set of trimming current sources is laid out symmetrically and is associated with a 6-bit RAM. Different sets of trim current sources do not need to match each other. Therefore, common centroid layout is not applied. The layout of the whole chip is shown in Figure 21.

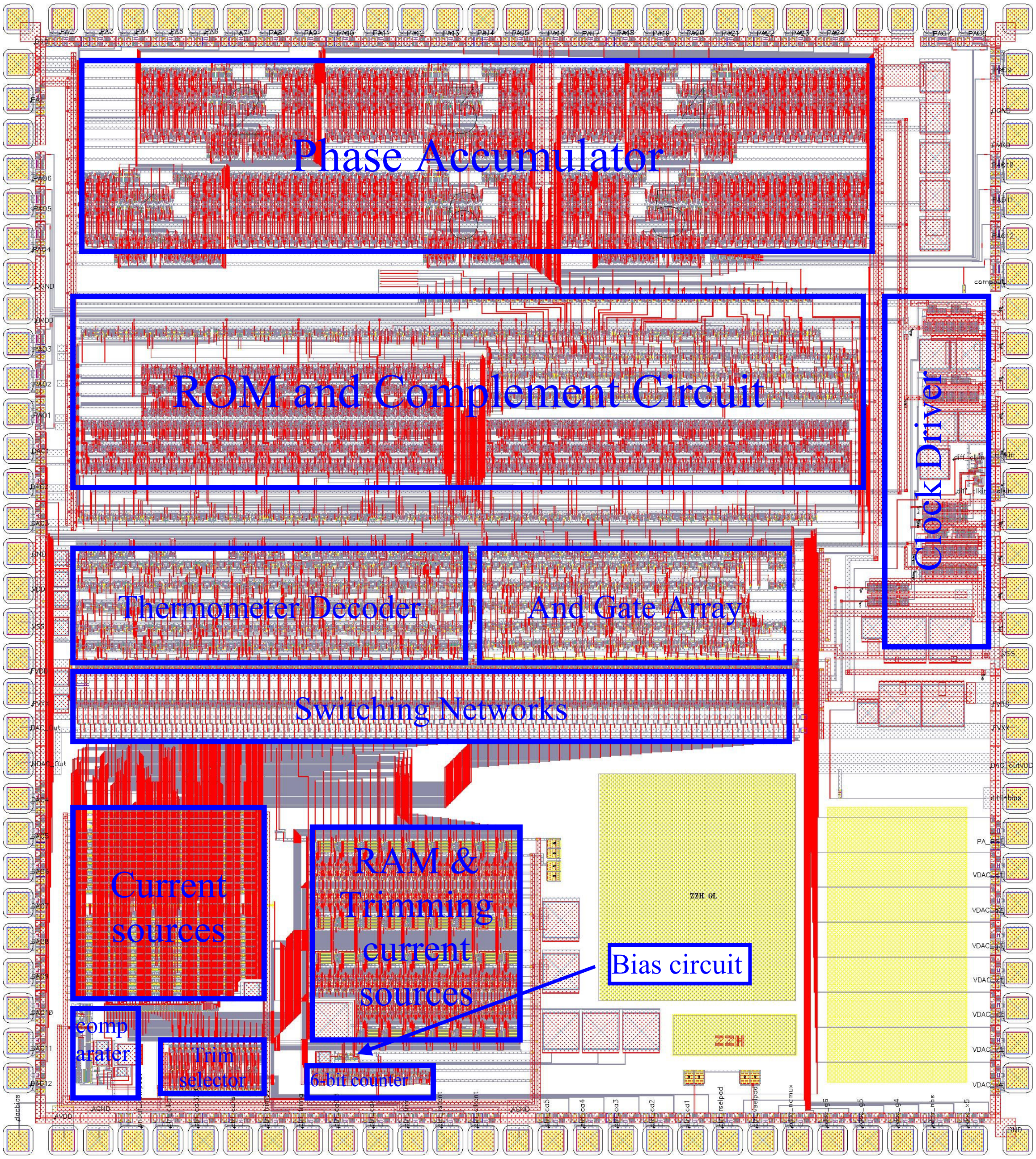


Figure 21 Chip layout

The switching network consists of high speed latches and current switches. The drain nodes of the current switches connect to the output lines. At high operation speed, RF effects must be taken into account. The differential output lines are laid out as 50 ohm transmission lines. One end of the transmission line is terminated with an on-chip 50 ohm resistor. The other end connects to the output pad that will eventually be connected to a 50 ohm load. The clock signal to the latch is also critical. A properly terminated 50 ohm transmission line is also used to deliver the timing critical clock signals.

3.6 Measurement results

The non-linear DAC was fabricated on Honeywell's 0.35 μm MOI5 SOI CMOS process. The chip is packaged in a 128 pin leaded quad flat pack. Figure 22 shows the die photo. The test setup is shown in Figure 23.

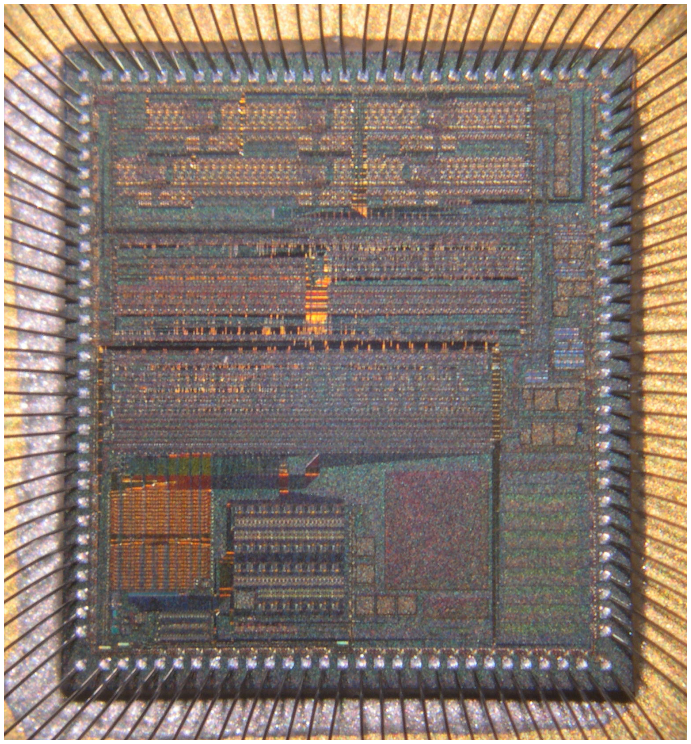


Figure 22 Die photograph of the chip

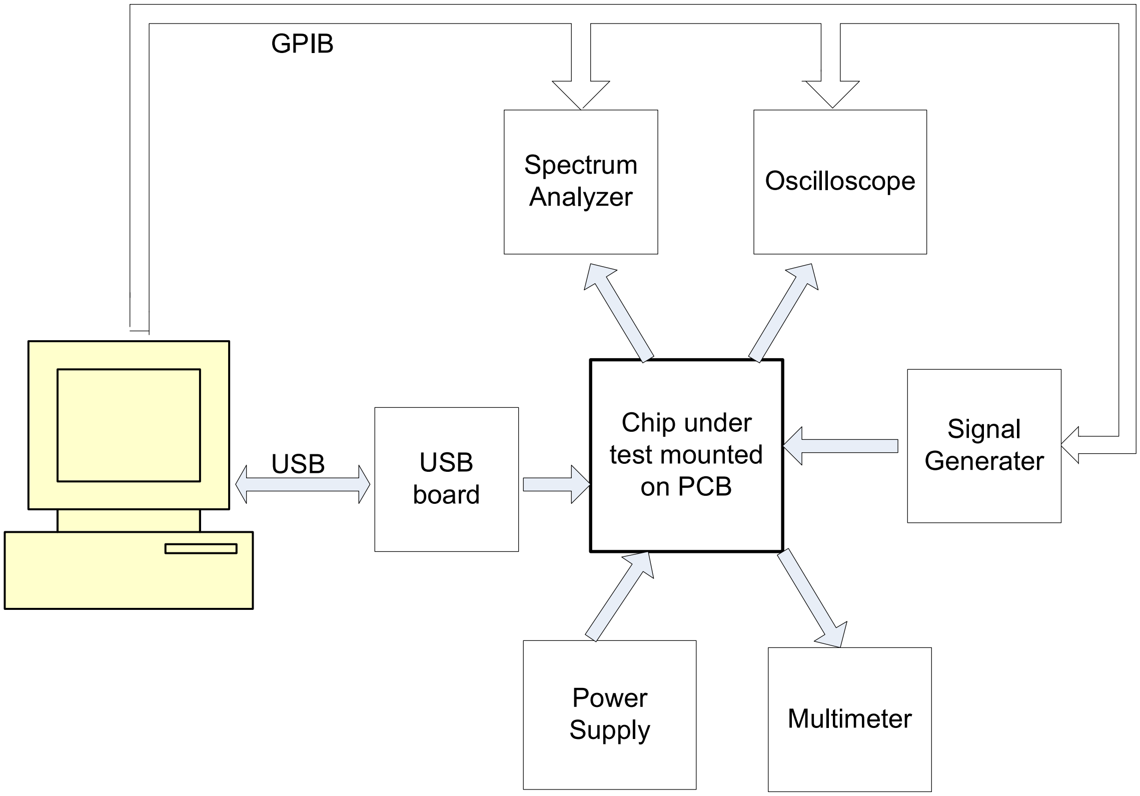


Figure 23 Test set up

The packaged chip was mounted on the test PCB. A USB board was purchased, which enables us to use a computer to control the chip. The USB board has 8 bits digital outputs. Additional circuitry is built to convert the 8 bit output to 48 bits of control signals. Other equipments such as power supplies, multimeters, signal generators, a spectrum analyzer and oscilloscopes, are also controlled by the computer through GPIB. The non-linear DAC was characterized for DC and high speed performance along with radiation testing for total ionizing dose.

3.6.1 DC characterization

The power supply for the non-linear DAC is 3.3 V. The DAC consume 120 mW at DC. Most of it is from the current sources. The total current is 25 mA, including the trimming current and reference. The nominal output current is 20.48 mA. It generates an output swing of 0.5 Vpp on a 50 ohm load that is in parallel with a 50 ohm on chip resistor.

The differential non-linearity (DNL) and integral non-linearity (INL) of the 12-bit offset DAC was measured. The initial result showed very poor matching for the unary current sources. The INL for the 31 unary current sources is within +12 LSBs and -20 LSBs, which is shown in Figure 24.

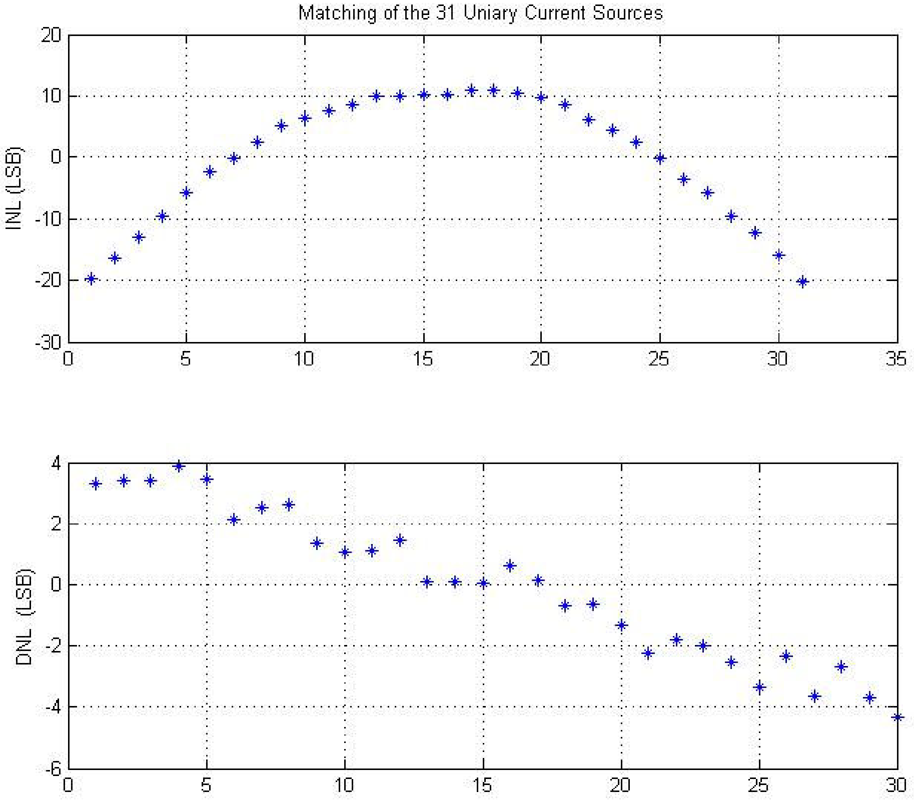


Figure 24 DAC DNL/INL with on-chip output resistors

The 6 binary weighted current sources showed very good matching as shown in Figure 25. The INL error is within +0.15 LSB and -0.12 LSB. The DNL is within +0.16

LSB and -0.1 LSB. It confirms that the small binary current sources do not need to be calibrated.

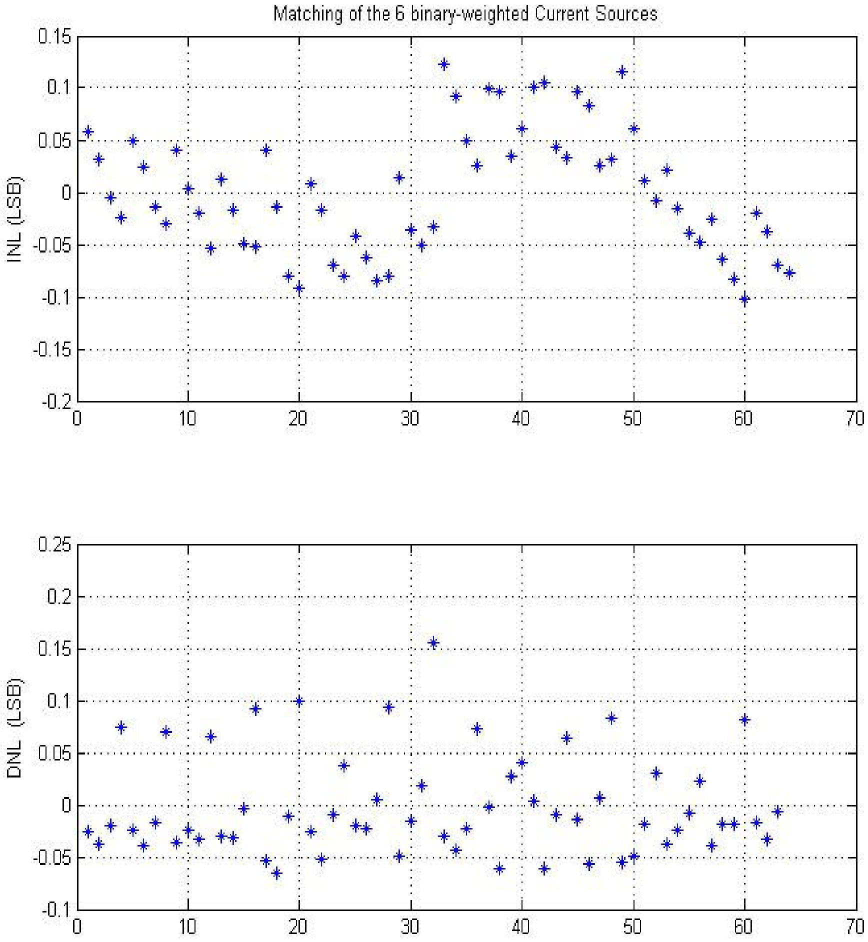


Figure 25 INL/DNL for the binary current sources

The problem for the unary current source is identified to be the on-chip output resistors. After removing the on-chip output resistors using laser cutter, the INL is reduced to within +2 LSBs and -2 LSBs. The DNL is within +1.2 LSB and -1.2 LSB. Figure 26 show the INL/DNL errors for the unary current sources with the on-chip resistor removed. That result is obtained before calibration and is within the calibration range, which is ±4 LSB for each unary current source.

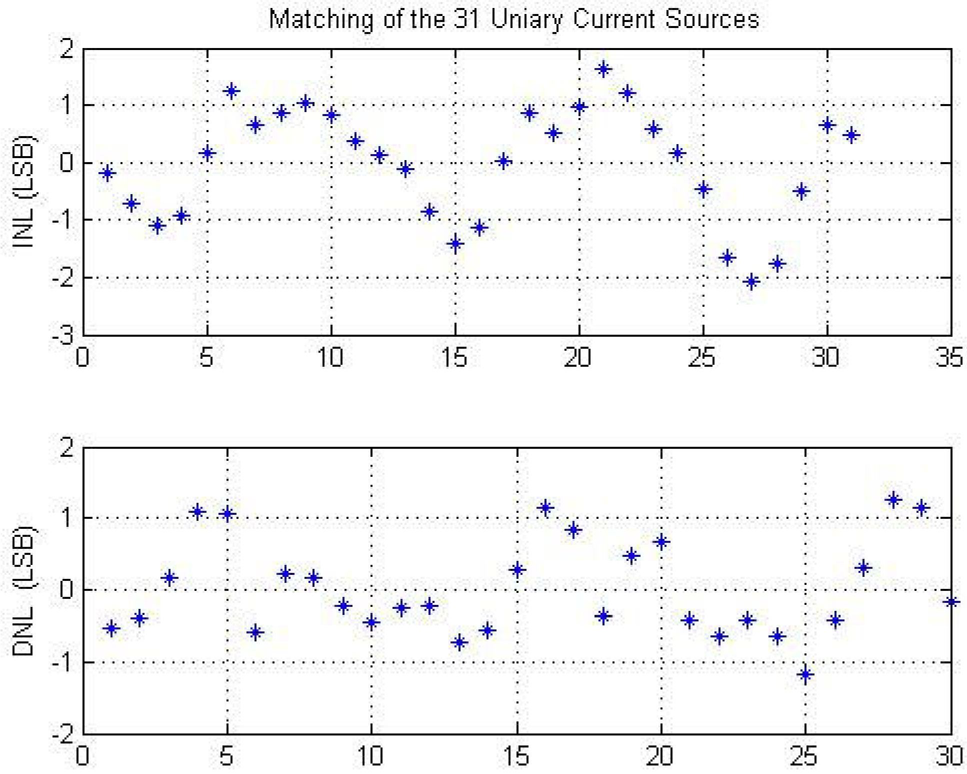


Figure 26 INL/DNL with the on-chip resistor removed

After performing digital calibration, the final DAC DNL/INL is measured. The result includes both unary and binary weighted current sources and is shown in Figure 27. The overall DNL/INL errors are within ±0.5 LSB. The DAC is confirmed to have 12-bit accuracy. The digital trimming approach is effective in compensating for process induced matching errors.

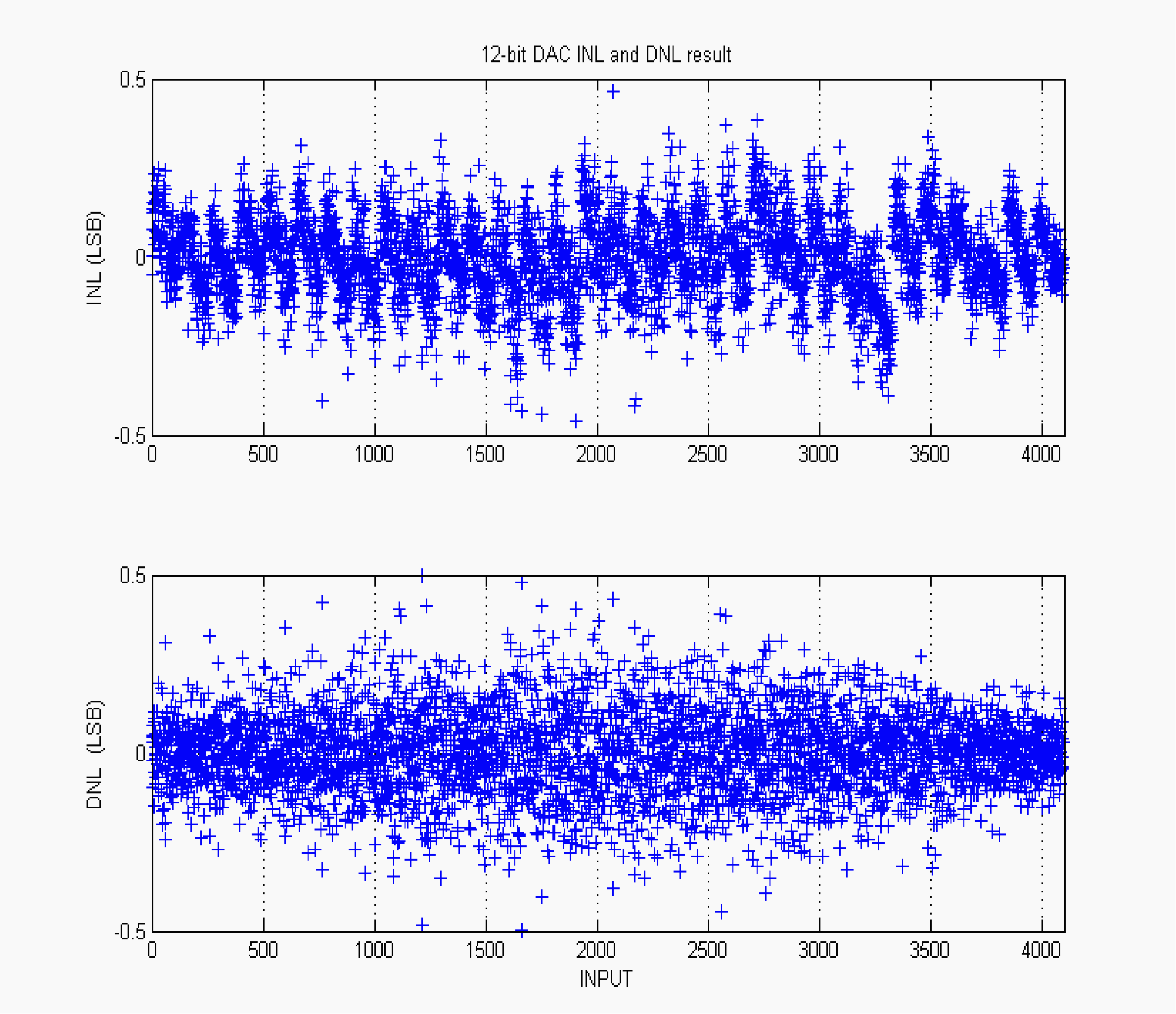


Figure 27 The Measured DAC INL/DNL after calibration

Operation of the vernier DAC was tested by providing external gain bits and segment offset bits. The output was measured to check the generation of different slopes. Figure 28 shows 16 of the possible gains. The figure was obtained by sweeping the segment offset from -31 to +31 for each of the 16 gains. The straight lines with different slopes show that the multiplying vernier DAC is working as expected.

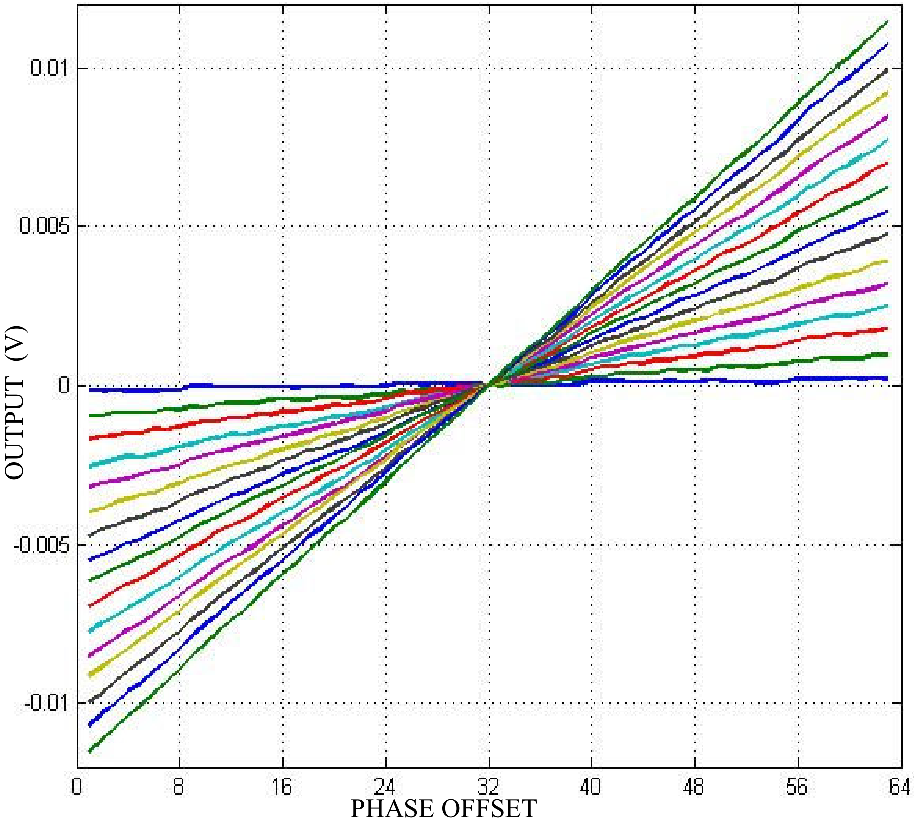
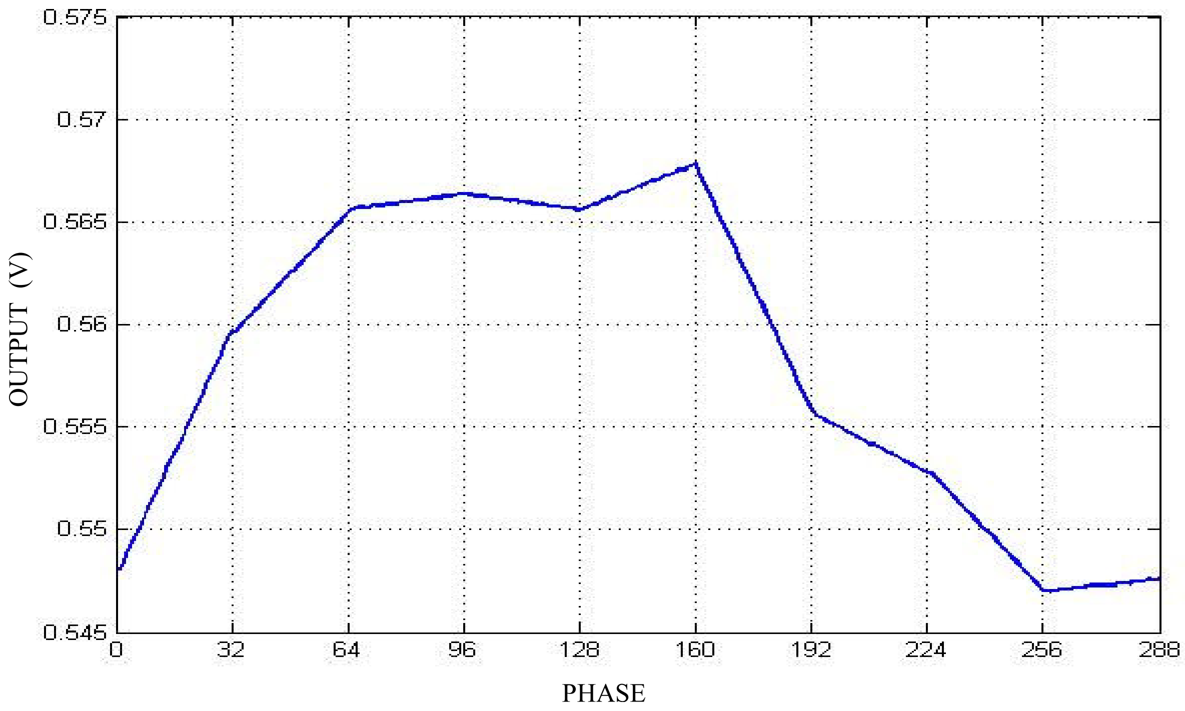


Figure 28 Variable gains of the vernier DAC

The non-linear DAC can generate a piecewise linear function and is not limited to just generating sine waves. As an illustration, the combined output of the offset DAC and vernier DAC is shown in Figure 29 with various offsets and gains to form a continuous waveform. Each segment consists of 32 points. The offset and gain words for each segment are shown in the table below the graph. The fully differential design of the vernier DAC enables the implementation of negative gain. When the gain is negative, the gain bits are set to the absolute value of the desired gain and the segment phase offset bits are inverted and the sign bit is set to 0.



Offset	2048	2172	2234	2234	2234	2134	2103	2041	2041
Gain	63	31	4	-4	12	-63	-15	-31	3

Figure 29 Output wave with different offsets and gains

3.6.2 High speed measurements

The PCB board for high speed test is shown in Figure 30. The packaged chip is soldered onto the board. The high speed differential clock signals connect to the two SMA connectors to the right of the chip. The two SMA connectors on the left side are the differential outputs. The small U.FL connectors are connected to the high speed digital inputs. The low speed control signals use simple SIP connectors. Decoupling capacitors are also used on the front as well as back on the board. All traces for the high speed signals are designed as 50 ohm transmission lines and are properly terminated. Unused pins are connected to ground.

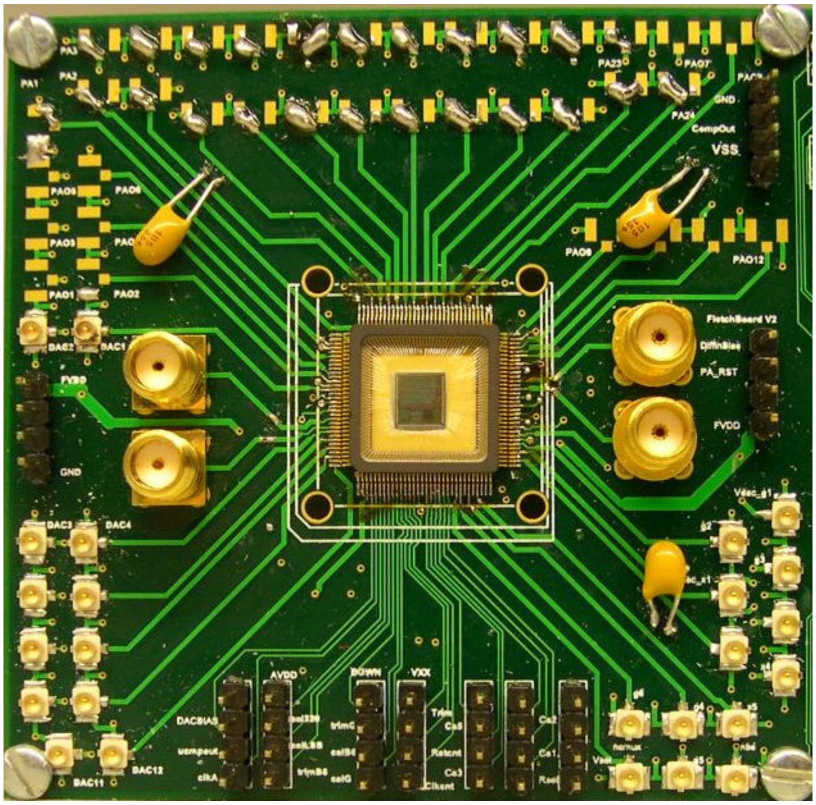


Figure 30 Test board with chip

Testing showed the non-linear DAC can be clocked up to 600 MHz. The power dissipation at 600 MHz clock is about 480 mW. The spurious-free dynamic range (SFDR) at low output frequency is up to 72 dBc and rolls off at high output frequencies. Figure 31 shows an output waveform of 6.77 KHz where the DAC is clocked at 20 MHz. The waveform confirms that the ROM lookup table and control logic are working correctly. Figure 32 to Figure 34 show output waves of 27 MHz, 57 MHz and 166 MHz, respectively, where the DAC is clocked at 600 MHz. The spectrum of output signals at 1.28 MHz, 13.5 MHz, 98 MHz and 166 MHz are shown in Figure 35 to Figure 38, respectively. These outputs are obtained at a data rate of 600 MSps. The spectra of 98 MHz output in Figure 37 have a second harmonic of -48.33 dBm while the fundamental is -1 dBm. So the SFDR at 98 MHz is 47.33 dBc. At 1.25 MHz output, the third harmonic is dominant and the SFDR is 69.17 dBc. The second harmonic becomes dominant at 13.5 MHz output and the SFDR is 64 dBc. Figure 39 shows the relationship between the SFDR and output frequencies.

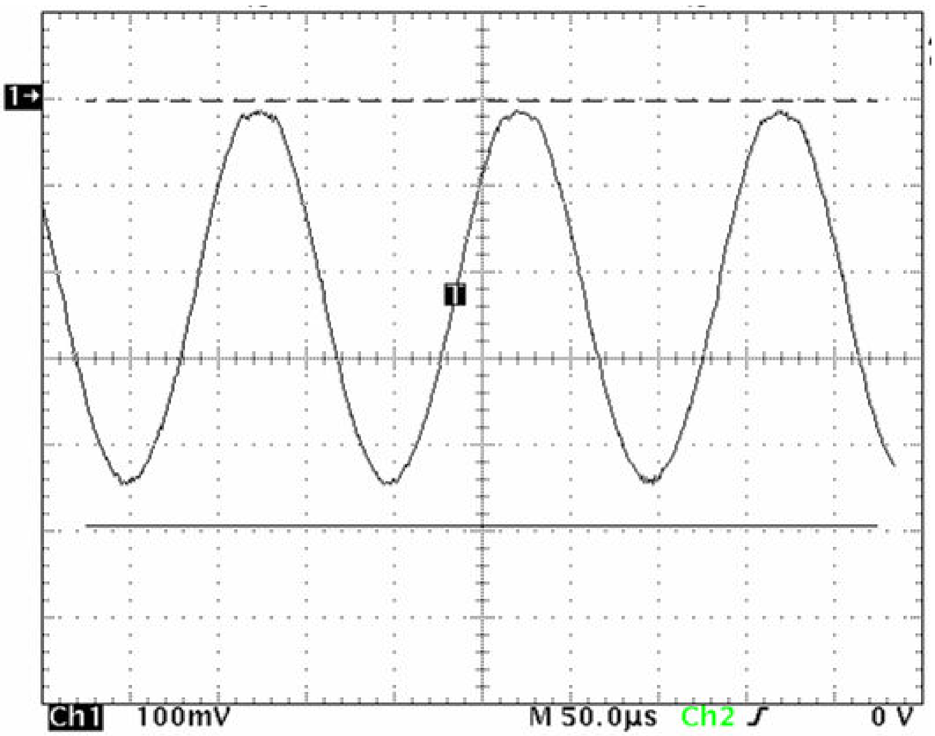


Figure 31 Sine wave output of 6.77 KHz at 20 MSps

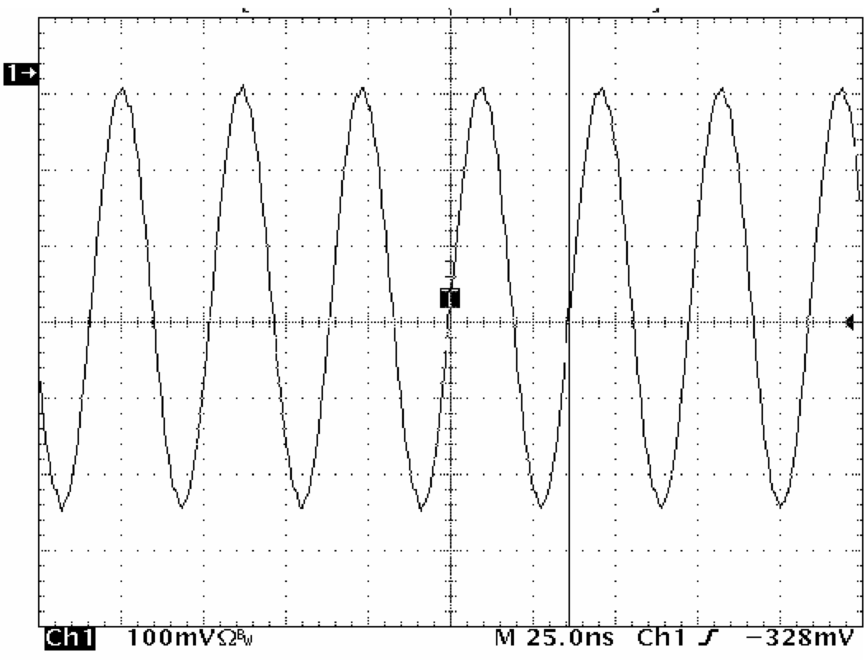


Figure 32 Output waveform at 27 MHz @ 600 MSps

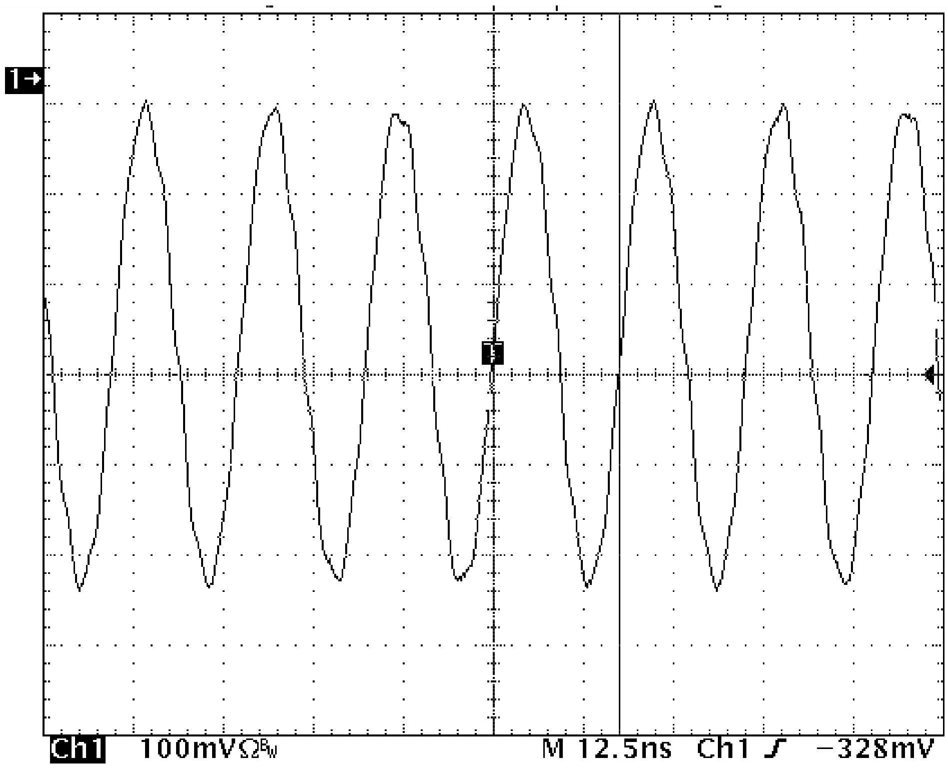


Figure 33 Output waveform at 57 MHz @ 600 MSps

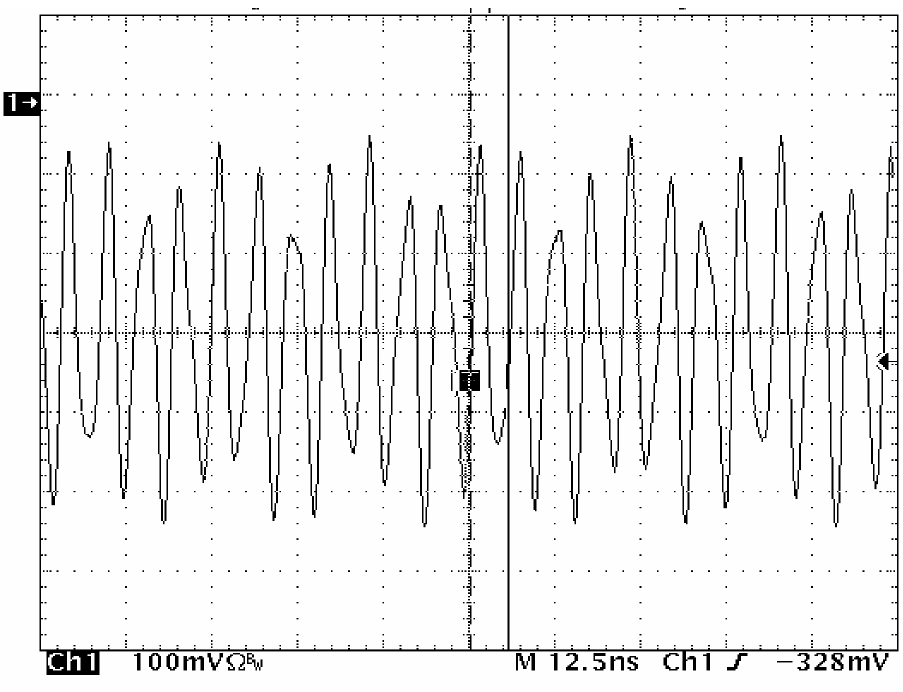


Figure 34 Output waveform at 166 MHz @ 600 MSps

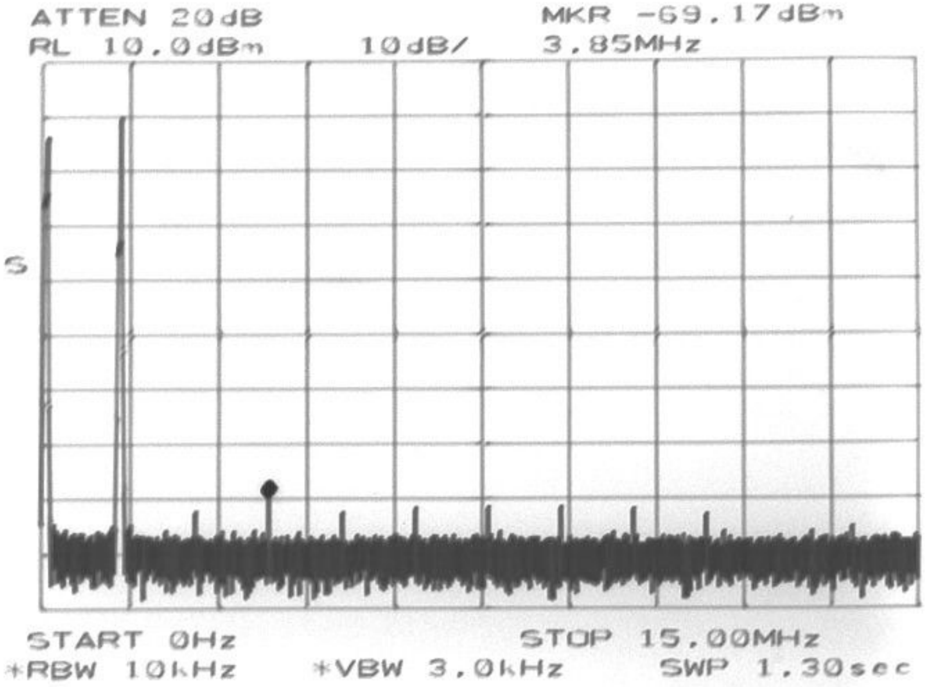


Figure 35 Spectrum of a 1.28 MHz output

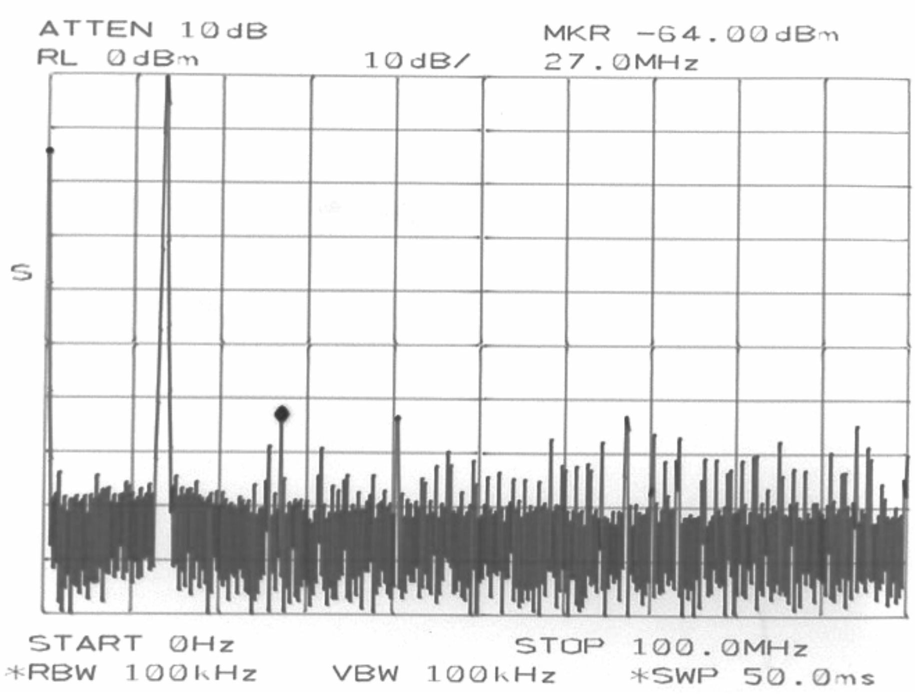


Figure 36 Spectrum of 13.5 MHz output

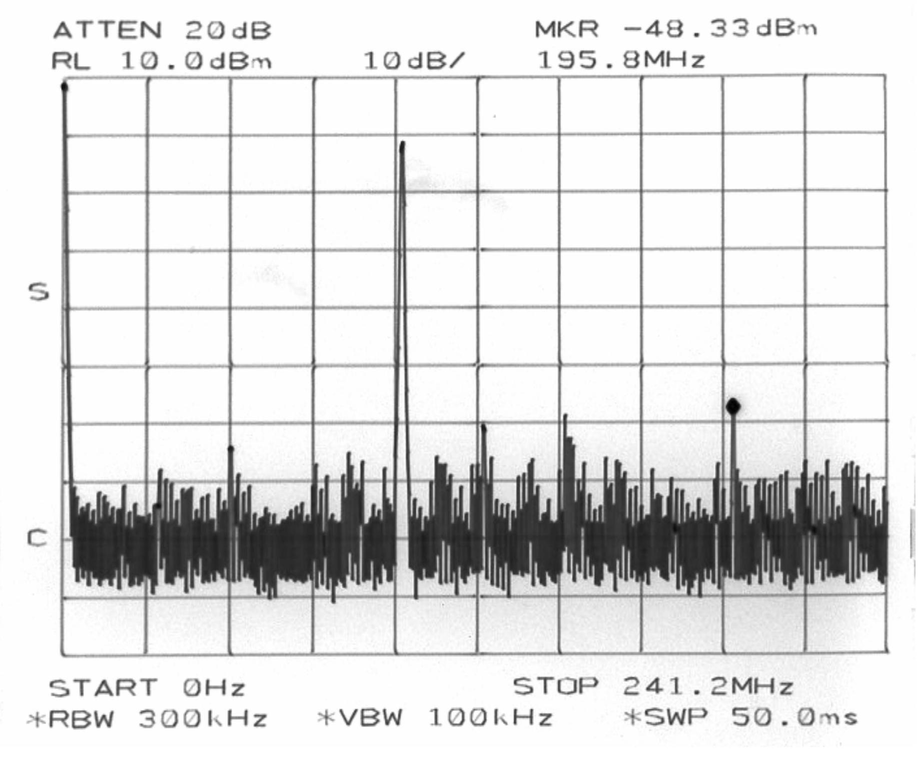


Figure 37 Spectrum of 98 MHz output

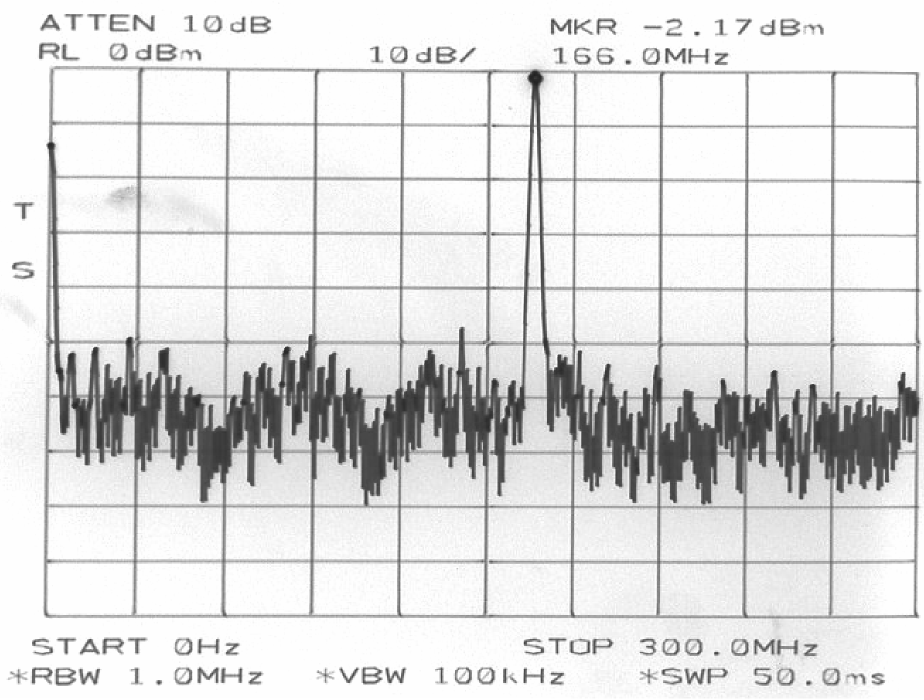


Figure 38 Spectrum of 166 MHz output

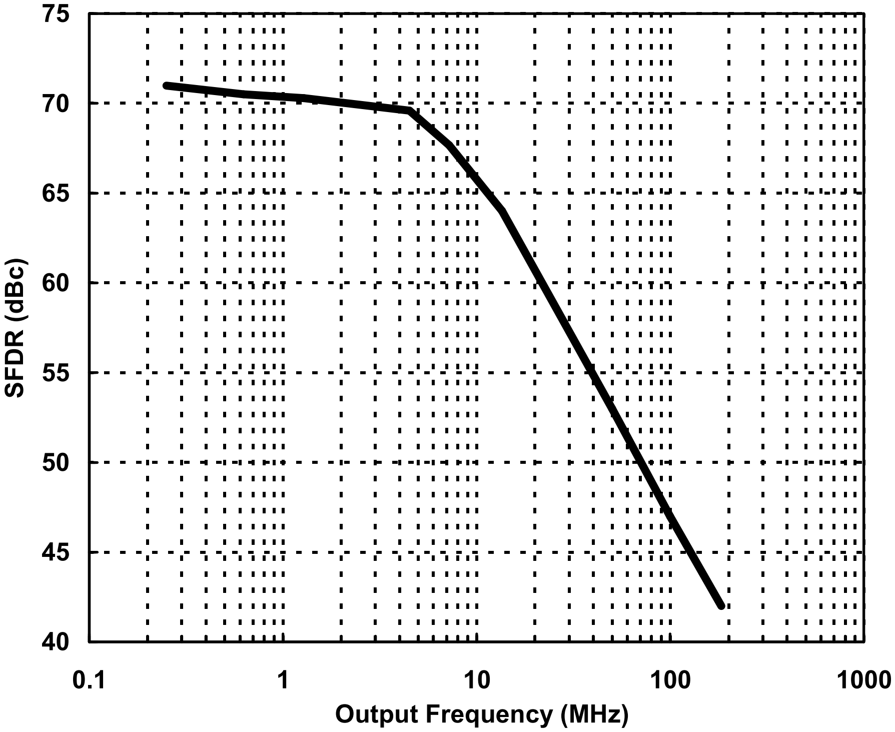


Figure 39 SFDR vs. output frequencies at 600 MSps data rate

3.5.3 Radiation testing

Total ionizing dose radiation testing was performed at Boeing Radiation Effects Laboratory. The INL and DNL errors were measured before and after radiation. Figure 40 through Figure 43 show the measured results for the lower 6 binary weighted current sources. The operation of digital logic and the power supply current were also monitored during the test.

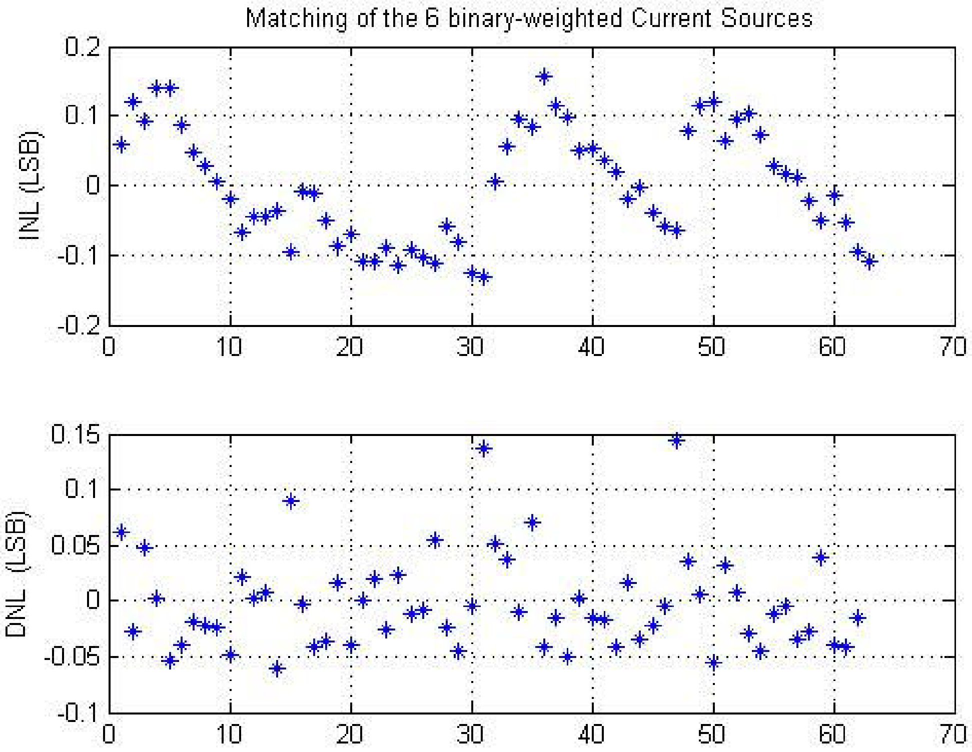


Figure 40 INL/DNL errors before radiation

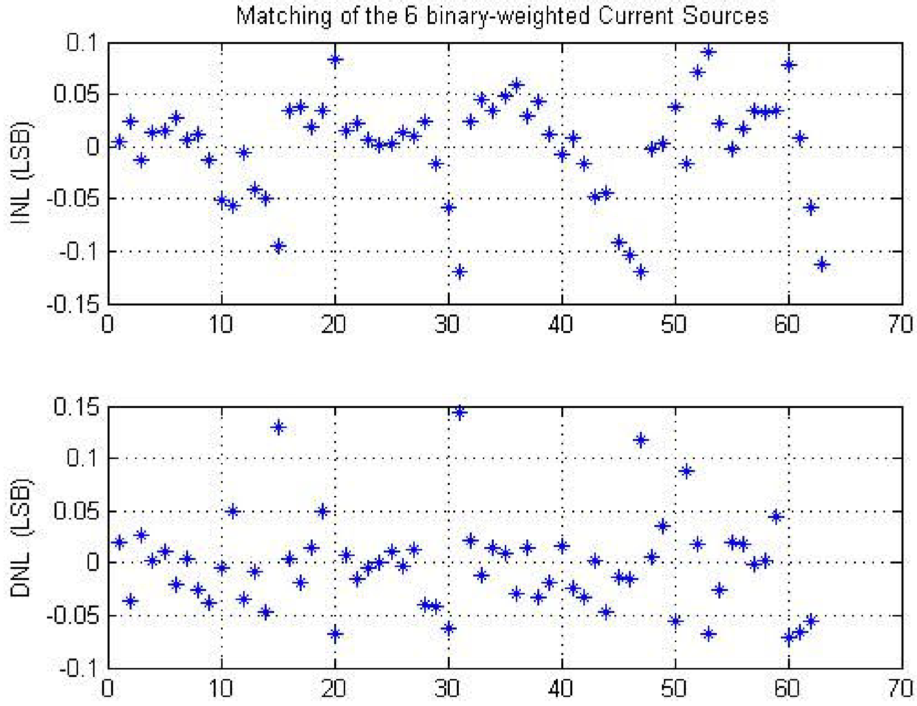


Figure 41 INL/DNL errors after 100 Krad Si

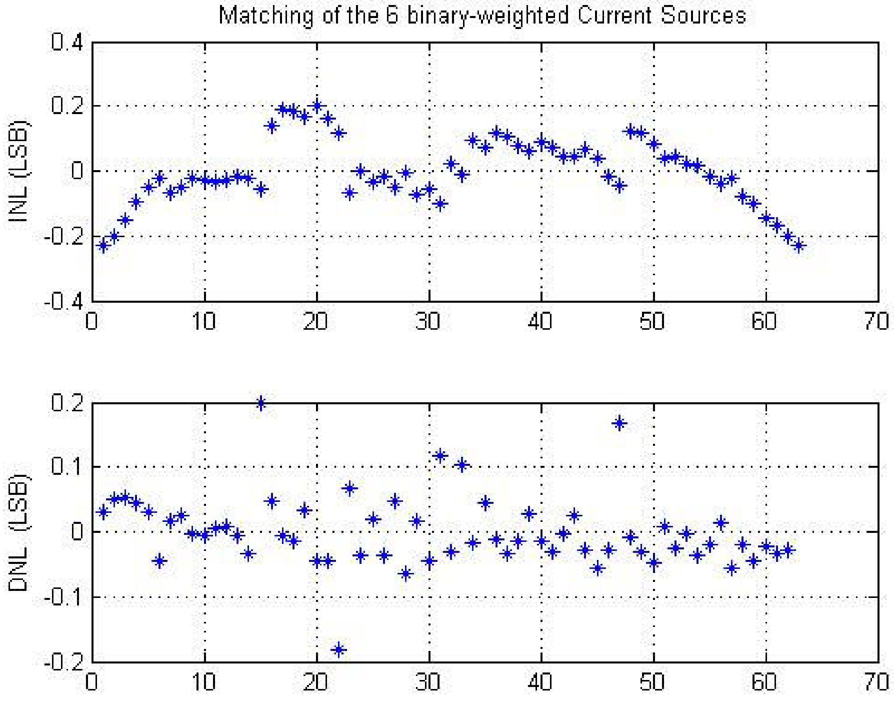


Figure 42 INL/DNL errors after 200 Krad Si

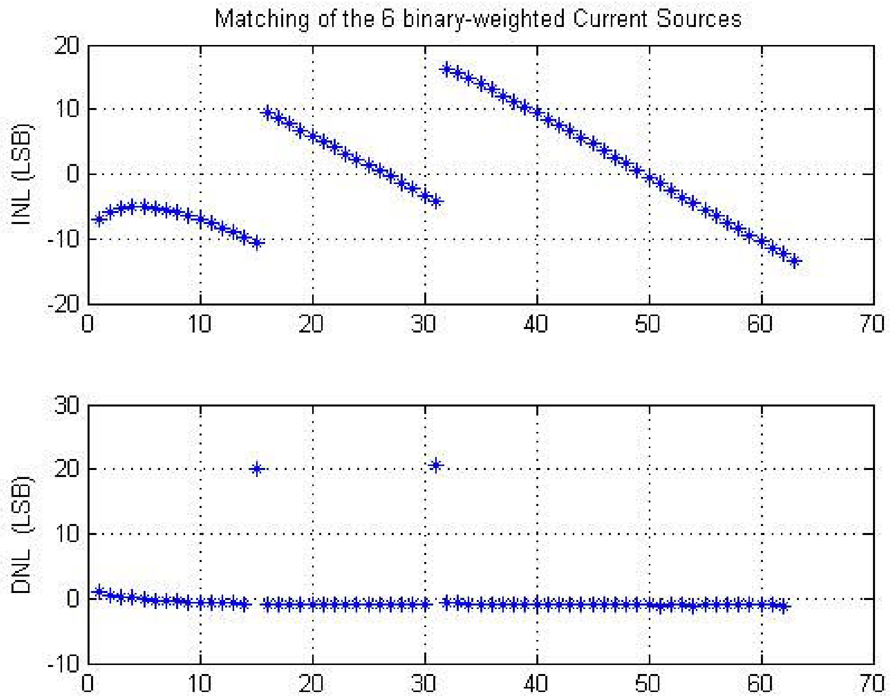


Figure 43 INL/DNL errors after 300 Krad Si

The DAC current sources showed only minor changes up through 150 Krad Si total ionizing dose (TID). The INL error of the lower 6 bit binary DAC increased 30% at 200 Krad. At 250 Krad Si, two current sources in the lower 6 bit DAC turned on hard, causing very poor DNL and INL. For the 5-bit unary DAC, only minor changes occurred up through 200 Krad but 2 current sources started leaking significantly at 250 Krad. The power supply currents had shown no changes up to 250 Krad but changed slightly at 300 Krad. The digital logic worked up to 300 Krad but after another 100 Krad dose, it stopped working.

3.7 Summary of the performance measurements

The performance of the non-linear DAC is summarized in the Table 2.

Table 2 Performance summary of the non-linear DAC

Technology	0.35 μM CMOS SOI
Core area	5mm^2
Power supply	3.3V
Power dissipation	120 mW at DC, 480 mW at 600 MHz
Resolution	12 bit
Maximum Clock Frequency	600 MHz
Output swing	0.5 Vpp
Output Resistance	50 Ω
SFDR	72 dBc at low output frequency 47 dBc at 100 MHz output
Radiation tolerance	Up to 200 krad Si total ionizing dose

Table 3 Comparison with other DDFS DACs

Chip	Technology	Core area	Power supply	Clock	Resolution	SFDR (dBc)
Yang [8]	0.35 um CMOS	1.3 mm^2	3.3 V	800 MHz	9 bits	55
Vankka [12]	0.8 um BiCMOS	3.9 mm^2	5 V	170 MHz	10 bits	60
Yamagishi [7]	0.5 um CMOS	15.0 mm^2	2V	80 MHz	10 bits	69
Edman [13]	0.8 um CMOS	3.8 mm^2	5 V	350 MHz	8 bits	50
Kosunen [14]	0.5 um CMOS	9.0 mm^2	3.3 V	150 MHz	10 bits	57
Jiang [15]	0.25 um CMOS	1.4 mm^2	2.5 V	300 MHz	11 bits	62
This work	0.35 um CMOS SOI	5.0 mm^2	3.3 V	600 MHz	12 bits	72

The non-linear DAC chip is compared with other DDFS DACs in Table 3. Since the DAC is the limiting factor of the DDFS performance given the digital processing is accurate enough, the DAC determines the overall performance of a DDFS. Compared with other DDFS DACs, our non-linear DAC achieves the highest SFDR aan operates at high clock rates. The non-linear DAC performs multiplication in the analog domain and eliminates the requirement of reproducing the digital sine wave. Hence, it simplifies the digital circuit design and increases the potential to operate the DDFS at higher speed.

A radiation hard 12-bit non-linear DAC was successfully designed, fabricated and tested. When used in DDFSs, the DAC can significantly reduce the ROM look up table size by a factor of 20, owing to its ability to implement a 32-segment piecewise linear approximation to a sine function. The SFDR is up to 72 dBc, which is close to the theoretical limitation for a 12-bit DAC. To further improve the SFDR, higher resolution DAC is needed.

CHAPTER FOUR

4. DESIGN OF A 14-BIT NON-LINEAR DAC

A 14-bit non-linear DAC was designed to improve the SFDR performance. This DAC implements a 16-segment piecewise-quadratic approximation to the sine function. Compared with the 12-bit non-linear DAC, the 14-bit DAC can reduce the look-up table size even more and still maintain high SFDR. The 14-bit DAC incorporates a top level switch reduction technique to reduce the number of top-level switches, which potentially allows for increased operating speed. In addition to digital calibration, other techniques, such as third-harmonic cancellation and reducing control signal levels on current switches, are used to improve the performance. Radiation hardness is achieved by design since a commercial non-SOI CMOS process is used. Special layout technique and hardware redundancy are used to achieve radiation tolerance. The ROM look-up table is replaced with RAM for reconfigurability.

4.1 Non-linear DAC Configuration

Figure 44 shows the block diagram of the 14-bit non-linear DAC. The non-linear DAC consists of three DACs; an offset DAC, a linear term DAC and a quadratic term DAC. Each DAC implements one term in Equation (3). The DACs are also designed using current steering architecture and the addition of the 3 terms is performed by summing the current at the output node. The Matlab code for determining the look-up table values are attached in Appendix A. The offset, linear and quadratic coefficient values are attached in Appendix B.

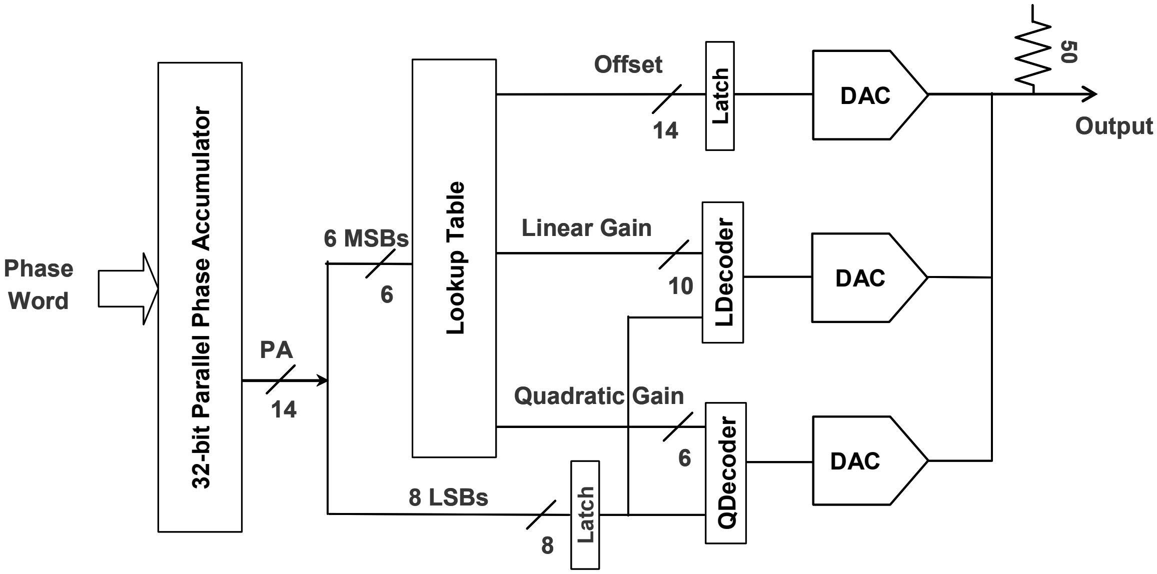


Figure 44 14-bit non-linear DAC with 16 segments PQA

4.2 Offset DAC

The offset DAC is implemented as a 14-bit hybrid current-steering DAC. The top 5 MSBs are implemented using unary current sources in the same way as the 12-bit non-linear DAC. The 9 LSBs consists of 9 binary-weighted current sources. Comparing with the 12-bit non-linear DAC, the 14-bit DAC has 2 more bits hence 2 additional binary current sources are added to form the 14-bit offset DAC.

Similar digital calibration techniques are used. The trimming current sources are also extended to include 8 binary weighted current sources. The smallest one is 1/8 LSB. Note that 1 LSB now equals ¼ LSB of the 12-bit DAC. Each unary current source is implemented with a current source that is 16 LSBs lower than the ideal value of 512 LSBs. A current mirror adds in an extra current of up to 32 LSBs from 8 binary-weighted trimming current sources that are controlled by an 8-bit word stored in RAM. The current can be varied from 496 LSBs to +528 LSBs with a resolution of 0.125 LSB by writing different words to the RAM. The largest binary current source has a value of 256 LSBs and also needs to be calibrated. Other current sources that need calibration include the large current sources in the linear term DAC, namely the current sources with sizes of 510, 256 and 255 LSBs. The calibration circuit is different from that of the 12-bit DAC and is shown in Figure 45.

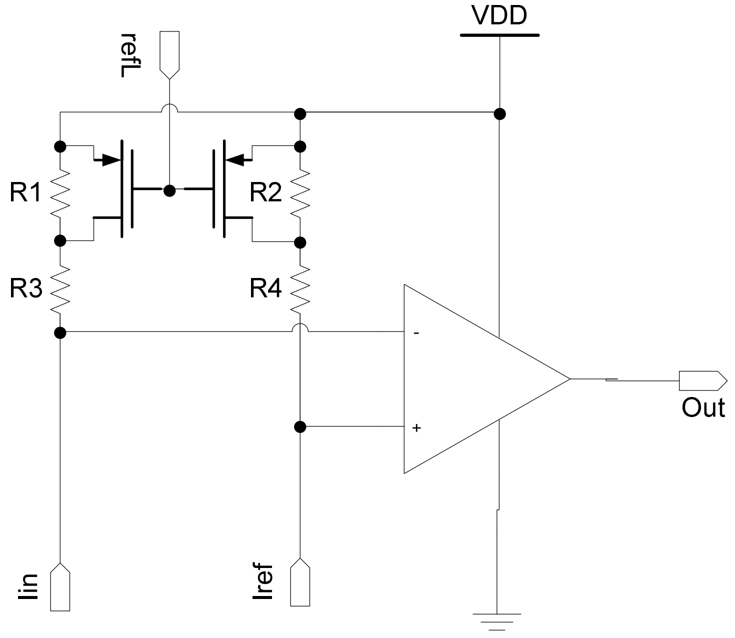


Figure 45 Calibration circuit

The four resistors R1, R2, R3 and R4 have the same values. They convert the current to a voltage for input to an opamp. The opamp acts as a comparator. The reference current I_{ref} t is always flowing . The current under calibration is connected to the I_{in} input node. The control signal refL controls two PFETs, which can short the two resistors R1 and R2 when on, i.e., when refL is low.

Compared with the calibration circuit in the 12-bit DAC, this design does not use capacitors and FET switches that inject charges to high impedance nodes. In the 12-bit

DAC, leakage of the capacitor forces the calibration circuit to operate at reasonably high speed. The reference voltage held on the capacitor has to be constantly refreshed. In the new design, although there is no charge injection problem, opamp DC offset and resistor mismatch do pose possible problems. However, further study shows that these problems can be easily solved.

Note that the purpose of calibration is to make sure the current sources under calibration match each other. They are not required to have the same current as the reference. If all calibrated current sources have a fixed offset to the reference current, they still match each other. Since the reference current is always connected to the positive input of the opamp, DC offset and resistor mismatch only cause a fixed difference between the reference current and the current under calibration. Thus, they pose no problem as long as the reference is always connected to the same opamp input.

Another concern would be how to make sure the calibrated current match the non-trimmable binary current sources. This problem is solved by making the reference current trimmable. Before using the reference current to calibrate a certain current, e.g. 256 LSB, the reference current is calibrated with the sum of all low binary weighted currents (128, 64, 32, 16, 8, 4, 2, 1 LSB) plus 1 LSB current. This procedure finds out the required reference current to match the desired current. The two currents will have a fixed difference but this not an issue as discussed above.

The current under calibration takes two values, 512 and 256 LSBs. The reference current is controllable and half the current can be turned off when calibrating current of 256 LSBs. The control signal refL is used to double the resistance for calibrating 256 LSB current so that the voltages at the opamp input nodes are always around half way between the power supply and ground. To calibrate the 510 LSB or 255 LSB currents, an additional 2 LSB or 1 LSB current is added to make the current under calibration to be 512 LSB or 256 LSB. This additional current only connects to the I_{in} of the calibration circuit and is turned off in normal operation.

4.3 Linear term DAC

The linear term DAC is implemented in the same way as the vernier DAC in the 12-bit DAC, but contains more current sources. It is a current steering DAC and consists 10 sets of binary-weighted current sources with sizes in LSBs shown in Table 4.

Table 4 Linear term DAC implementation

	X_s	X_8	X_7	X_6	X_5	X_4	X_3	X_2	X_1
G_{10}	510	256	128	64	32	16	8	4	2
G_9	255	128	64	32	16	8	4	2	1
G_8	128	64	32	16	8	4	2	1	1
G_7	64	32	16	8	4	2	1	1	
G_6	32	16	8	4	2	1	1		
G_5	16	8	4	2	1	1			
G_4	8	4	2	1	1				
G_3	4	2	1	1					
G_2	2	1	1						
G_1	1	1							

The multiplication in the linear term DAC is also performed by summing partial products the same way as in the 12-bit DAC design. Table 4 shows how this is implemented. X_i (i=1 to 8) denotes the i_{th} bit of the phase offset within a segment and G_k (k=1 to 10) the k_{th} bit of the gain for the current segment. X_s is the sign bit. These bits control 62 current sources with relative sizes given in the table. Current is switched to the output when both row and column control bits are high. Otherwise the current is switched to the complementary output. The LDecoder in Figure 44 is implemented using an array of AND gates. The column under X_s ensures the differential outputs not to be disrupted.

4.4 Quadratic term DAC

The quadratic term DAC was designed using a similar approach for multiplication. Note that quadratic gain coefficients are 6-bit words. Thus, only the lower 6 rows in Table 4 (shaded area) are implemented for the quadratic term DAC. The quadratic term contains the square of phase offset so a squaring circuit [8] is needed.

The QDecoder in Figure 44 has a 5-bit squaring circuit in addition to AND gates forming the partial products. The top 5 significant bits of the segment phase offset are fed into the squaring circuit, the output of which is then fed to the partial product AND gates. Only the top 6 bits of the squared result are needed to provide enough resolution, simplifying the implementation. The squaring circuit is implemented using combinational logic requiring 50 logic gates instead of a fully digital multiplier. The simplified implementation enables high speed operation.

Table 5 shows the truth table for the 5 bit squaring logic. The inputs and outputs are divided into 8 groups. Each group is associated with a variable d. d_i=1 (i=0 to7) means the current input and output are in the i_{th} group. The input is 5 bits (i4 i3 i2 i1 i0) and the output is 10 bits (o9o8…o0). Only the top 6 bits of the output are implemented in the logic.

Table 5 Truth table for 5-bit squaring logic

		X										X^2						d
		i4	3	2	1	0		o9	8	7	6	5	4	3	2	1	0	d0
00	->	0	0	0	0	0	-->	0	0	0	0	0	0	0	0	0	0	
01	->	0	0	0	0	1	-->	0	0	0	0	0	0	0	0	0	1	
02	->	0	0	0	1	0	-->	0	0	0	0	0	0	0	1	0	0	
03	->	0	0	0	1	1	-->	0	0	0	0	0	0	1	0	0	1	
04	->	0	0	1	0	0	-->	0	0	0	0	0	1	0	0	0	0	d1
05	->	0	0	1	0	1	-->	0	0	0	0	0	1	1	0	0	1	
06	->	0	0	1	1	0	-->	0	0	0	0	1	0	0	1	0	0	
07	->	0	0	1	1	1	-->	0	0	0	0	1	1	0	0	0	1	
08	->	0	1	0	0	0	-->	0	0	0	1	0	0	0	0	0	0	d2
09	->	0	1	0	0	1	-->	0	0	0	1	0	1	0	0	0	1	
10	->	0	1	0	1	0	-->	0	0	0	1	1	0	0	1	0	0	
11	->	0	1	0	1	1	-->	0	0	0	1	1	1	1	0	0	1	
12	->	0	1	1	0	0	-->	0	0	1	0	0	1	0	0	0	0	d3
13	->	0	1	1	0	1	-->	0	0	1	0	1	0	1	0	0	1	
14	->	0	1	1	1	0	-->	0	0	1	1	0	0	0	1	0	0	
15	->	0	1	1	1	1	-->	0	0	1	1	1	0	0	0	0	1	
16	->	1	0	0	0	0	-->	0	1	0	0	0	0	0	0	0	0	d4
17	->	1	0	0	0	1	-->	0	1	0	0	1	0	0	0	0	1	
18	->	1	0	0	1	0	-->	0	1	0	1	0	0	0	1	0	0	
19	->	1	0	0	1	1	-->	0	1	0	1	1	0	1	0	0	1	
20	->	1	0	1	0	0	-->	0	1	1	0	0	1	0	0	0	0	d5
21	->	1	0	1	0	1	-->	0	1	1	0	1	1	1	0	0	1	
22	->	1	0	1	1	0	-->	0	1	1	1	1	0	0	1	0	0	
23	->	1	0	1	1	1	-->	1	0	0	0	0	1	0	0	0	1	
24	->	1	1	0	0	0	-->	1	0	0	1	0	0	0	0	0	0	d6
25	->	1	1	0	0	1	-->	1	0	0	1	1	1	0	0	0	1	
26	->	1	1	0	1	0	-->	1	0	1	0	1	0	0	1	0	0	
27	->	1	1	0	1	1	-->	1	0	1	1	0	1	1	0	0	1	
28	->	1	1	1	0	0	-->	1	1	0	0	0	1	0	0	0	0	d7
29	->	1	1	1	0	1	-->	1	1	0	1	0	0	1	0	0	1	
30	->	1	1	1	1	0	-->	1	1	1	0	0	0	0	1	0	0	
31	->	1	1	1	1	1	-->	1	1	1	1	0	0	0	0	0	1	

All the d values are generated using a 3-8 decoder with i4 i3 i2 as its inputs. The final outputs can be produced with at most 3 stages of logic gates. The logic to generate the outputs follows:

$$d0=\overline{i4}\cdot\overline{i3}\cdot\overline{i2},\quad d1=\overline{i4}\cdot\overline{i3}\cdot i2,\quad d2=\overline{i4}\cdot i3\cdot\overline{i2},\quad d3=\overline{i4}\cdot i3\cdot i2,$$

$$d4=i4\cdot\overline{i3}\cdot\overline{i2},\quad d5=i5\cdot\overline{i3}\cdot i2,\quad d6=i4\cdot i3\cdot\overline{i2},\quad d7=i4\cdot i3\cdot i2$$

$$o9=i4\cdot i3+d5\cdot i1\cdot i0$$

$$o8=d4+d7+d5\cdot\overline{i1\cdot i0}$$

$$o7=d3+d5\cdot\overline{i1\cdot i0}+(d6+d7)\cdot i1$$

$$o6=d2+(d3+d4)\cdot i1+d5\cdot i1\cdot\overline{i0}+d6\cdot\overline{i1\cdot\overline{i0}}+d7\cdot i0$$

$$o5=(d1+d2)\cdot i1+(d3+d4)\cdot i0+(d5+d6)\cdot(i1\oplus i0)$$

$$o4=(d1+d5)\cdot\overline{i1\cdot\overline{i0}}+(d2+d6)\cdot i0+(d3+d7)\cdot\overline{i0+i1}$$

4.5 Radiation hard by design

Circuits designed for space application must be able to tolerate the effects of radiation. Ionizing radiation creates electron-hole pairs that can be trapped in the field oxide. Where the gate overlaps the field oxide a parasitic channel can form even with no gate bias applied. The leakage between source and drain using standard FET layouts causes increased power dissipation and circuit failure at high enough doses. In modern sub-micron processes the gate oxide is thin enough to avoid trapped charge in the gate oxide above the channel from turning on the channel.

A widely used radiation-hard-by-design (RHBD) layout technique is to use enclosed or annular FETs [17] and is shown in Figure 46. This enclosed layout has no channel over field oxide between source and drain and the radiation tolerance of the circuit improves significantly. An accurate approach has been developed to generate SPICE models for annular FETs and other arbitrary gate geometry FETs useful for RHBD design [18]. Annular FETs and variations, such as gate-around-source (GAS) and gate-around-drain (GAD) with smaller width-to-length ratios, are used to provide total dose radiation tolerance to the non-linear DAC. Radiation also can cause single-event upsets which can lead to transient errors. The on-chip RAM is protected against SEUs using the dual interlocked memory cell [19]. The dual interlocked memory cell is shown in Figure 47.

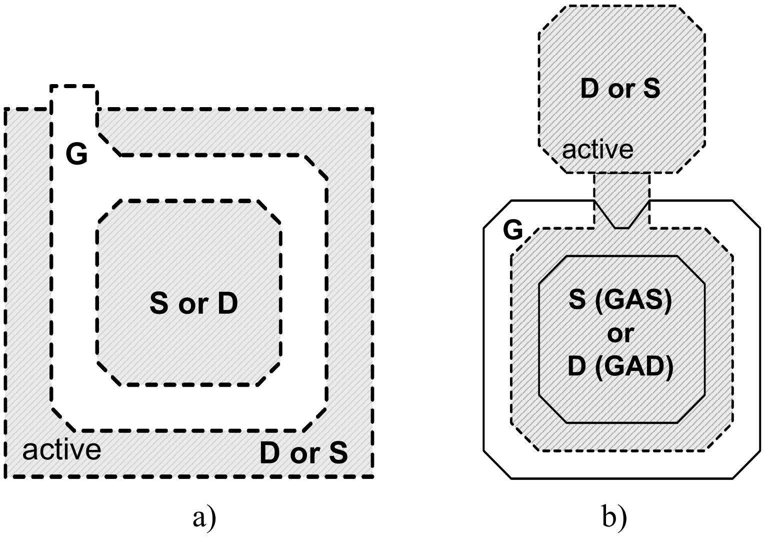


Figure 46 a) Annular FET layout and b) gate-around-source (GAS) or gate-around-drain (GAD) layout

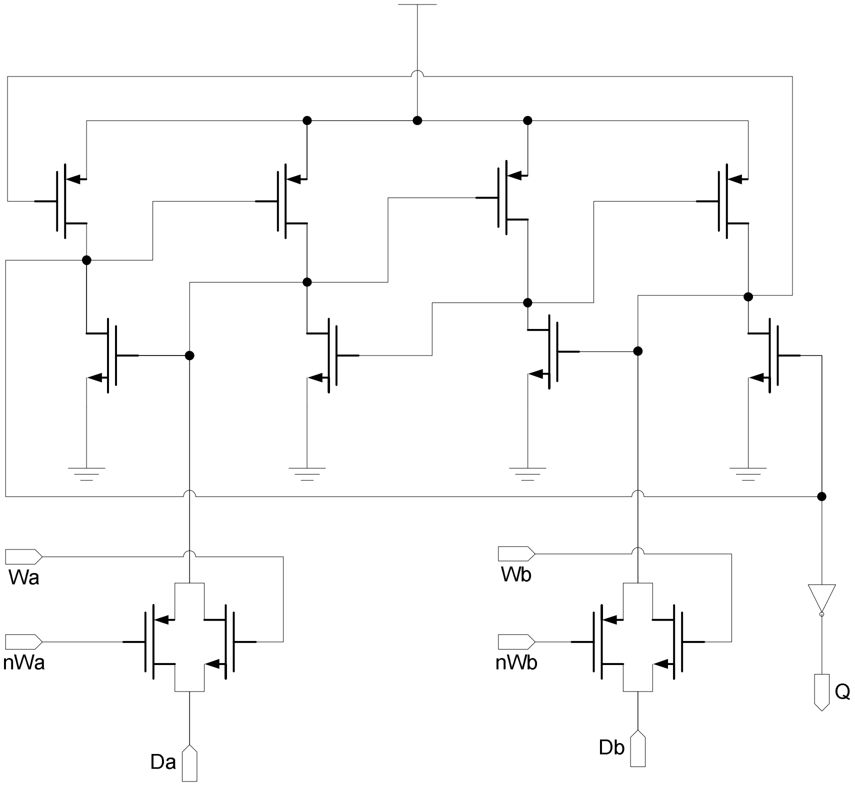


Figure 47 Dual interlocked memory cell

The dual interlocked memory cell has two inputs Da and Db and two sets of write signals. The state of the memory can only be changed by simultaneously changing the state of two internal nodes. Two sets of data and control signals are required to write the RAM. If one node temporally changes state due to being hit by ions, the cell restores to the original state and the output does not change. The possibility that two internal nodes get hit is low if the cell is carefully laid out so that the critical internal nodes are separated sufficiently.

4.6 Techniques to improve the high speed performance

The speed of the non-linear DAC suffers if all of the switches in the offset, linear and quadratic term DACs are implemented at the top level due to the large number of total switches. One way to reduce the number of switches at the top level is to group small current sources together. Each group only uses one top level switch and each member in the group has a lower level switch as shown in Figure 48. This technique is referred to as top level switch reduction. However, timing errors between the first and second level switches will adversely affect SFDR. To solve this problem, we use an approach that alternately uses two pairs of linear and quadratic term DACs operating at half rate. The top level switches alternately select different pairs of DACs at full speed to allow the second-level switched currents to settle for a clock cycle. This keeps the number of switches at the top level to fewer than 60, 41 of which are from the offset DAC. Since the linear term and quadratic term DACs operate at half speed, the time constraint for the digital control logic, such as the squaring circuit, is relaxed. The cost for this approach is that two sets of linear term DACs and quadratic term DACs are needed.

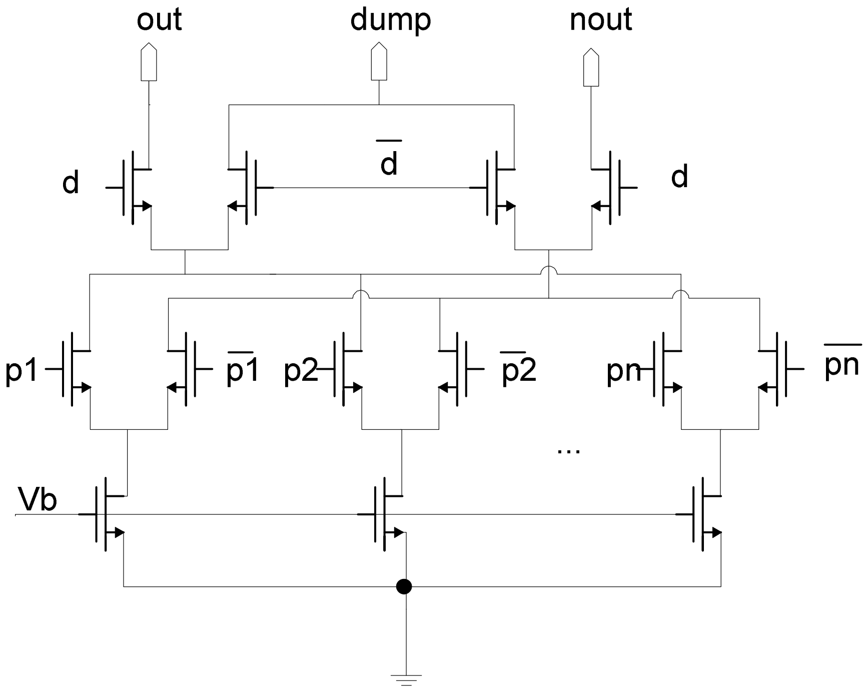


Figure 48 Top level switch reduction

The maximum current of each group must be less than 256 LSBs, which is half the unary current size. Otherwise, large glitches can be introduced at the output when alternating the switches. Note that the linear term DAC contains current source as large as 510 LSBs. It is beneficial to break the linear term DAC into two parts. Currents equal to or larger than 128 LSBs are implemented using only top level switch in the same way as the offset DAC. Currents less than 128 LSBs are grouped into sets and each set has total current no more than 256 LSBs. One top level switch is required for each set.

The high speed performance is not only affected by the current switch size but also by the switch control signal. Using smaller size switches usually results in better SFDR. The differential control signals to the square-law FET switches need to have a cross point at 0.707 of the signal swing to minimize the voltage fluctuation at the source of the current switch FETs. For the binary weighted current sources, the current switches should also have binary weighted sizes. However, achieving this is difficult because the process limits minimum transistor sizes. The minimum size transistor may be larger than desired for the lower LSB current switches and the capacitance between gate and source or drain may introduce output glitches. At high clock frequencies, the glitch energy may exceed the signal energy and degrade performance. Adding one more bit to the DAC with large switch FETs does not necessarily improve SFDR performance.

SPICE simulations of the DDFS using non-radiation hard rectangular FETs show an SFDR of 80 dB. An SFDR of 80 dB was also obtained after substituting radiation hard FETs for all of the transistors except the lower 4 LSB current switches. The SFDR degraded significantly when very small width to length ratio GAS or GAD FETs were substituted. Width to length ratios of these devices are as low as rectangular FETs but have much higher capacitance between the gate and the surrounded node leading to large glitches.

The swing of the control signal was reduced in an attempt to improve SFDR. Figure 49 shows the circuit used for the lower 4 LSBs. The output from the high speed latch drives a resistor divider. The level lowered signal then controls the current switch M1 and M2. This circuit provides an improvement of 6 dB but the overall SFDR was still lower than the desired 80 dB. An additional technique, third harmonic cancellation, is used to increase the SFDR to 80 dB at high frequencies.

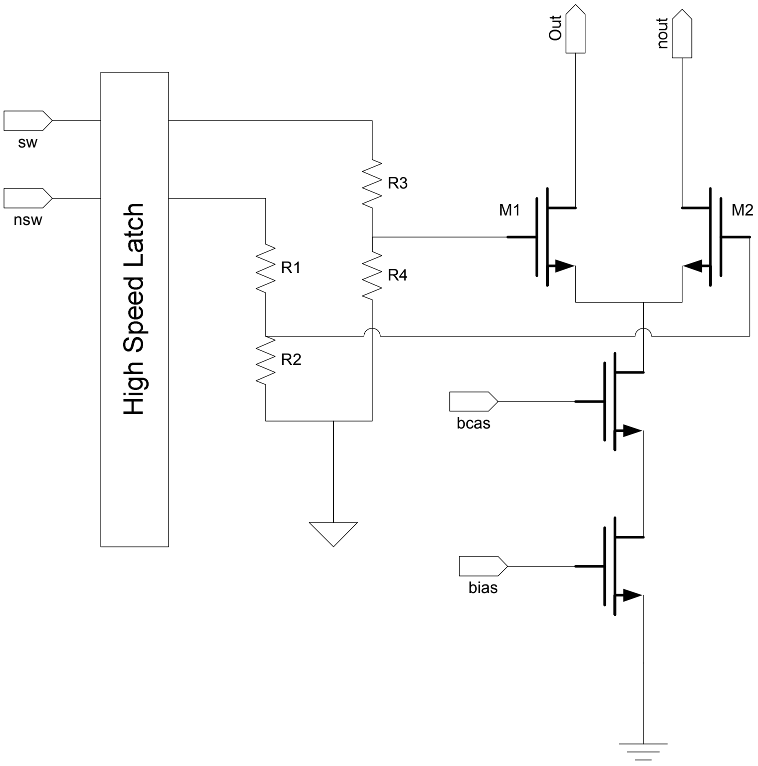


Figure 49 The current switch control signal for lower binary current sources

4.7 Third harmonic cancellation

The offset DAC, linear term DAC and quadratic DAC are all designed to have differential outputs so that all even order harmonics are cancelled. However, the odd order harmonics are unaffected and the third order harmonic is often the dominant one. A third harmonic cancellation technique was introduced to improve the SFDR for high frequency outputs. Figure 50 shows the block diagram of the circuit.

A look-up table is often used in DDFS for sine wave generation. We introduce an additional table driving a few additional current sources, which can similarly be used to generate a third harmonic. If the amplitude and phase are correct, the third harmonic can be cancelled. SPICE simulations show the phase and amplitude of the third harmonic depend heavily on the output frequency. An improvement of 14 to 19 dB is achieved at high frequency. The magnitude of the required correction signal decreases for low frequency outputs and the correction must be turned. Figure 51 shows the simulated result for canceling the 3rd harmonic at 125 MHz and 250 MHz output frequencies.

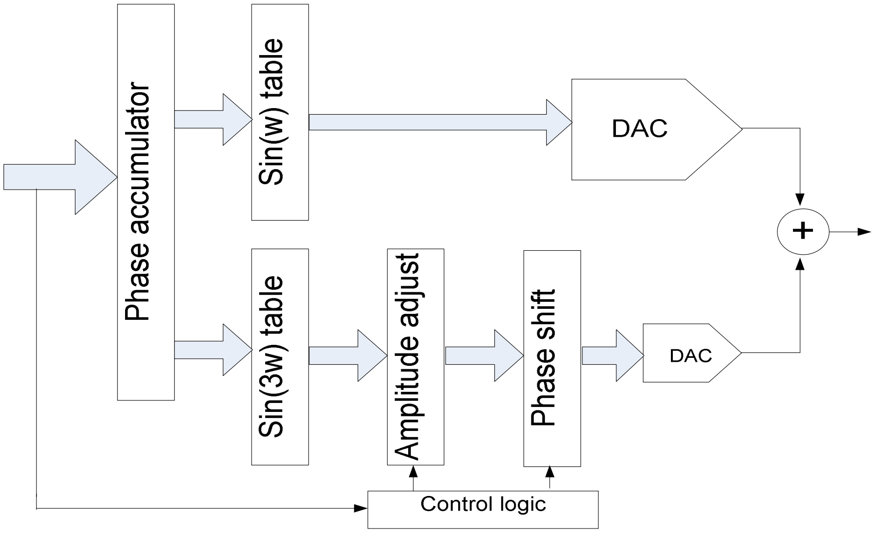
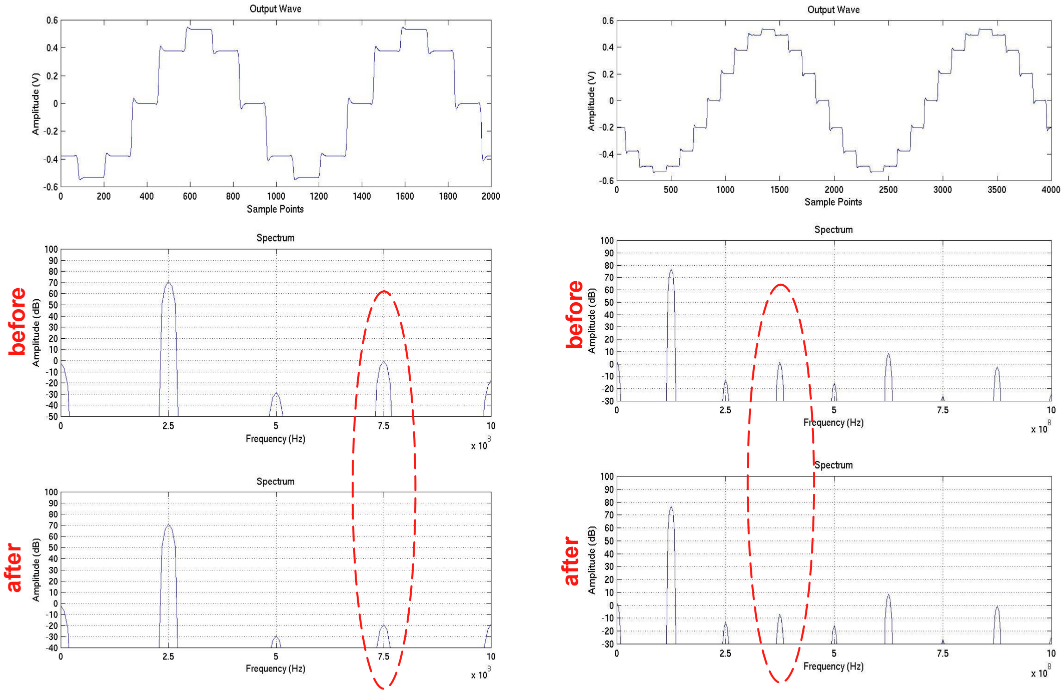


Figure 50 3rd harmonic cancellation



a) b)

Figure 51 Cancel the 3rd harmonic for a) 250 MHz and b) 125 MHz output at 2 GSps

Figure 52 show the simulated SFDR before (dashed) and after (solid) third harmonic cancellation as a function of frequency at a data rate of 2 GSps. The improvement in SFDR is 18 dB, 16.5 dB and 14 dB at output frequencies of 250 MHz, 167 MHz and 125 MHz, respectively. The cancellation is turned off at 62.5 MHz.

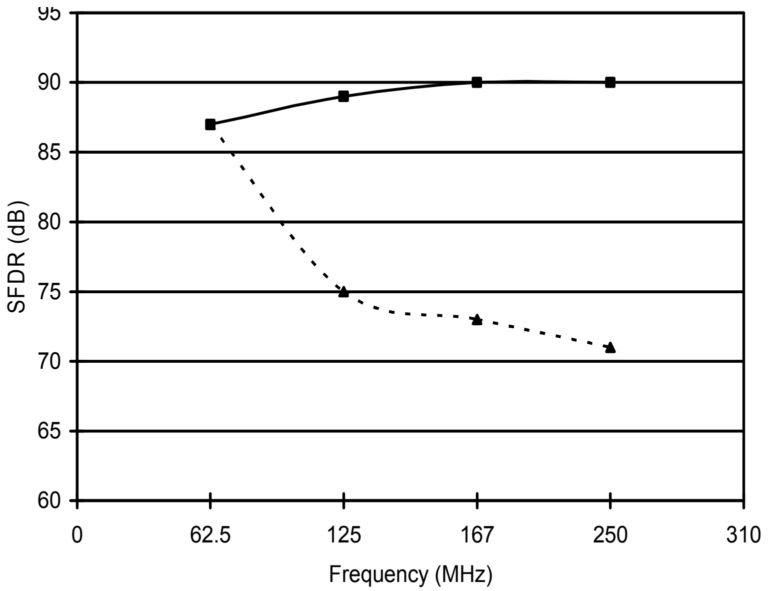


Figure 52 SFDR before (dashed) and after (solid) 3rd harmonic cancellation

4.8 Chip layout

The layout of the 14-bit non-linear DAC and a 32-bit phase accumulator is shown in Figure 53.

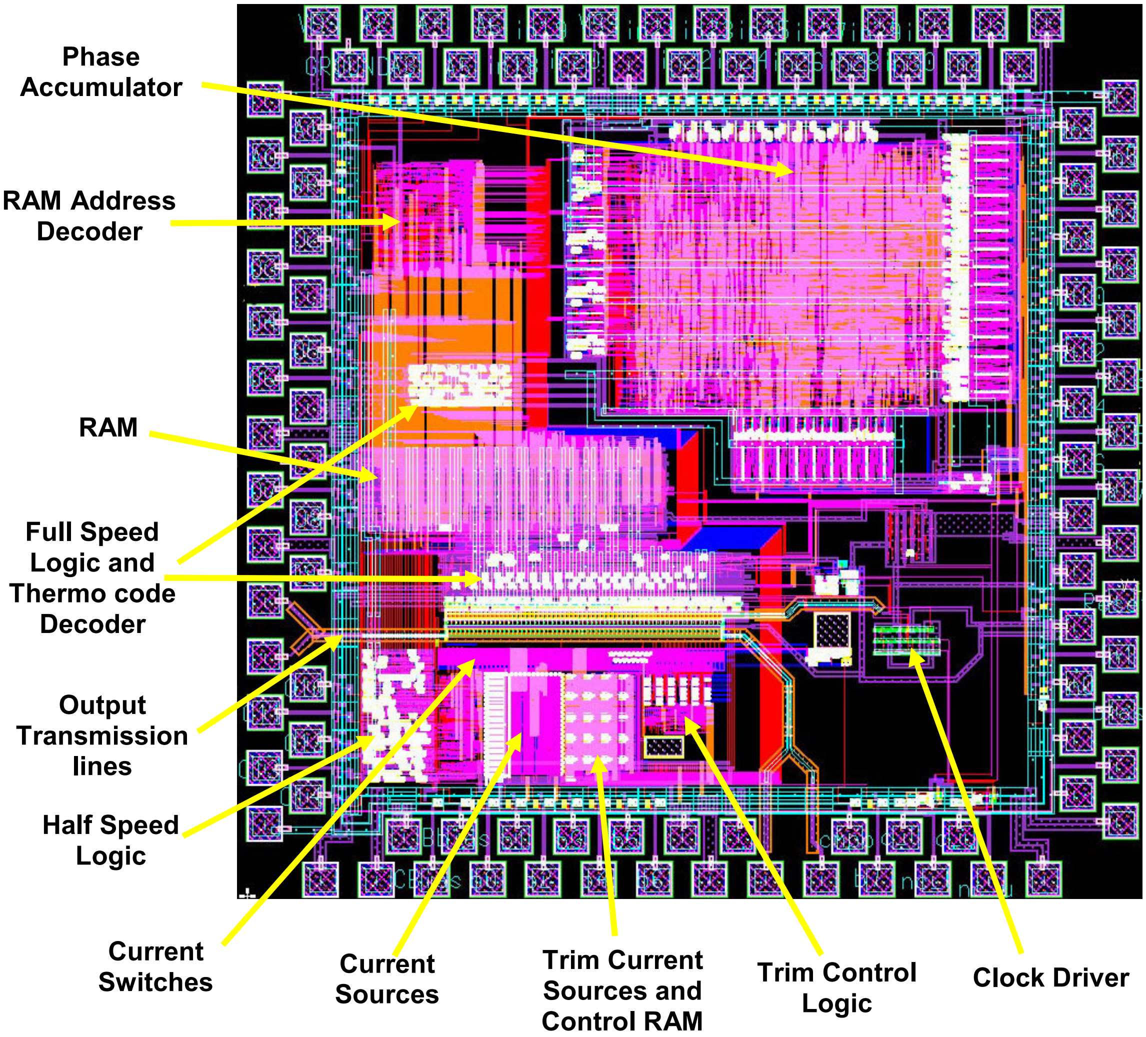


Figure 53 Layout of the second chip

CHAPTER FIVE

5. CONCLUSIONS AND FUTURE STUDY

A 12-bit non-linear DAC was designed, fabricated and tested. Results show the DAC has true 12 bit accuracy and can substantially reduce the required memory look up table size and hence the power dissipation when used in DDFS applications. The non-linear DAC can be used to implement a piecewise linear approximation to a sine function. The DAC operates up to 600 MSps and has high SFDR and spectral purity. The non-linear DAC was fabricated in a 0.35 μm SOI process and can tolerate radiation up to a total ionizing dose of 200 Krad Si.

The design of 14-bit non-linear DAC was also completed. The DAC, together with a 32-bit parallel phase accumulator, is fabricated on Jazz 0.18 um CMOS process. Simulations show that the SFDR is 80 dB up to 250 MHz output frequency. The 14-bit DAC can implement a 16-segment piecewise quadratic approximation and reduces the look-up table size and power even further compared to a piecewise linear approximation. The non-linear DAC uses dual interlocked memory cell and other radiation hard layout techniques to achieve radiation hardness. Top level switch reduction and third harmonic cancellation techniques are also used to improve the high speed performance.

The previous designs use a regular offset DAC to generate the analog levels for the segment offsets. The segment offsets take a large portion in the look up table. The offset DAC only provides a limited number of output values: 64 for the 12-bit DAC and 32 for the 14-bit DAC in a full cycle of sine wave. Note that the digital control signals of the offset DAC current switches can be derived using combinational logic from the input phase bits. This would enable us to totally remove the memory for storing the digital offset bits and achieve an even higher memory compression ratio. If the logic is much simpler than the saved memory, higher speed and lower power dissipation would result.

BIBLIOGRAPHY

[1] J. Tierney, C.M. Rader, and B. Gold, “A digital frequency synthesizer,” *IEEE Trans. on Audio and Electroacoustics*, vol. 19, no. 1, pp. 48-57, January 1971.

[2] J. Vankka, “Methods of mapping from phase to sine amplitude in direct digital synthesis,” IEEE *Trans. Ultrason., Ferroelectr. Freq. Contr.*, vol. 44, pp. 526-534, Mar. 1997

[3] A. Bellaouar, M. S. O’brecht, A. M. Fahim, and M. I. Elmasry, “Low-power direct digital requency synthesis for wireless communications,” *IEEE J. Solid-State Circuits*, vol. 35, pp. 385-390, Mar. 2000

[4] A. M. Sodagar and G. R. Lahiji, “Mapping from phase to sine-amplitude in direct digital frequency synthesizers using parabolic approximation,” *IEEE Trans. Circuits Syst. II*, vol. 47, pp. 1452-1457, Dec. 2000

[5] S. Liao and L. G. Chen, “A low-power low-voltage direct digital frequency synthesizer,” in *Proc. Int. Symp. VLSI Technology, System, and Applications*, June 1997, pp. 265-269

[6] J. M. P. Langlois and D. Al-Khalili, “ROM size reduction with low processing cost for direct digital frequency synthesis,” in *Proc. IEEE Pacific Rim Conf. Communications, Computers and Signal Processing*, Aug. 2001, pp. 287-290

[7] A. Yamagishi, M. Ishikawa, T. Tsukahara and S. Date “A 2-V, 2-GHz low power direct digital frequency synthesizer chip-set for wireless communication,” in *IEEE Journal of Solid-State Circuits*, vol. 33, February 1998, pp. 210-217

[8] B. D. Yang, J. H. Chois, S.H. Han, etc., "An 800-MHz low-power direct digital frequency synthesizer with an on-chip D/A converter*", IEEE J. Solid-State Circuits*, vol 39, no. 5, May 2004, pp. 761-774

[9] Mortezapour, S., and Lee, E.K.F., "Design of low-power ROM-less direct digital frequency synthesizer using nonlinear digital-to-analog converter," *IEEE J. Solid-State Circuits*, vol. 34, no. 10, pp. 1350-1359, Oct. 1999.

[10] A. Van den Bosch, M. A. F. Borremans, M. S. J. Steyaert, and W. Sansen*,* "A 10-bit 1 GSample/s Nyquist Current-Steering CMOS D/A Converter", *IEEE J. Solid-State Circuits*, 36(3), pp. 315–324, 2001.

[11] Z. Zhou, I. Horowitz, G. S. La Rue, "Non-linear DAC implementations in DDFS," *IEEE Workshop on Microelectronics and Electron Devices*, pp 124-125, 2004.

[12] J. Vankka, M. Waltari, M. Kosunen, and K. A. I. Halonen, "A direct digital synthesizer with an on-chip D/A-Converter," in *IEEE Journal of Solid-State Circuits*, vol. 33, February 1998, pp. 218-227

[13] A. Edman, A. Bjorklid, and I. Soderquist, "A 0.8 μm CMOS 350 MHz quadrature direct digital frequency synthesizer with integrated D/A converters," in *Symp. VLSI Circuits Dig. Papers*, June 1998, pp.54-55

[14] M. Kosunen, J. Vankka, M. Waltari, L. Sumanen, K. Koli and K. Halonen, "A CMOS quadrature baseband frequency synthesizer/modulator," in *Analog Integrated Circuits Signal Processing*, vol. 18. pp. 55-67, Jan. 1999

[15] J. Jiang, and E. K. F. Lee, "A Low-Power Segmented Nonlinear DAC-Based Direct Digital Frequency Synthesizer," in *IEEE Journal of Solid-State Circuits*, vol. 37, October 2002, pp. 1326-1330

[16] R. C. Lacoe, J. V. Osborn, R. Koga, S. Brown, and D. C. Mayer, “Application of hardness-by-design methodology to radiation-tolerant ASIC technologies,” *IEEE Trans. Nucl. Sci*., vol. 47, pp. 2334–2341, Dec. 2000

[17] R. C. Lacoe, J. V. Osborn, R. Koga, S. Brown, and D. C. Mayer, “Application of hardness-by-design methodology to radiation-tolerant ASIC technologies,” *IEEE Trans. Nucl. Sci.*, vol. 47, pp. 2334–2341, Dec. 2000

[18] C. Champion, “Modeling of FETs with arbitrary gate geometries for radiation hardening,” Thesis, Washington State University, Aug. 2000

[19] M. J. Myjak, D. R. Blum, J. G. Delgado-Frias, “Enhanced fault-tolerant CMOS memory elements,” *47th Midwest Symp. on Circuits and Systems*, pp. 453-456 July 2004

[20] A. Bosch, M. Steyaert, and W. Sansen, "SFDR-bandwidth limitations for high speed high resolution current steering CMOS D/A converters," *Proc. IEEE ICECS*, pp. 1193-1196, Sept. 1999.

[21] M. Pelgrom, and A. P. G. Welbers, “Matching properties of MOS transistors,” in *IEEE Journal of Solid-State Circuits*, vol. 24, October 1989, pp. 1433-1440

APPENDIX

A. THE MATLAB CODE FOR LOOKUP TABLE VALUE DETERMINATION

```
% 14-bit DAC with 16 segment approximation
clear all
N=2^14;  % number of phase point in one cycle
x=1:N/4; % phase point in pi/2
num=256; % number of points in each segment
global y;
finalwave=zeros(4096, 1);
tfinalwave=zeros(256, 1);

% set up tables
ltab=[256 128 64 32 16  8  4  2;
      128  64 32 16  8  4  2  1;
       64  32 16  8  4  2  1  1;
       32  16  8  4  2  1  1  0;
       16   8  4  2  1  1  0  0;
        8   4  2  1  1  0  0  0;
        4   2  1  1  0  0  0  0;
        2   1  1  0  0  0  0  0;
        1   1  0  0  0  0  0  0;
        1   0  0  0  0  0  0  0];

stab=[ 16 8 4 2 1 1;
        8 4 2 1 1 0;
        4 2 1 1 0 0;
        2 1 1 0 0 0;
        1 1 0 0 0 0;
        1 0 0 0 0 0];
```

```
  % compute the ideal sine wave
  ideal=(N/2)*sin((x-0.5)*2*pi/N); %ideal value of sine wave
for 0-pi/2
  ideal=ideal';

  %read offsets lcoef scoef
  load 'offsets16' % ideal offsets
  load 'lcoef16'   % ideal coefficients for linear terms
  load 'scoef16'   % ideal square coefficients

  % max current 512 LSB for x and 16 LSB for x^2
  % search for best coef.
  newoffsets=zeros(32, 1); % to store optimum offsets
  newlcoef=zeros(32, 1);   % optimum linear gains
  newscoef=zeros(32, 1);   % optimum quadratic coefficients
  for i=1:N/4/num
    n1=(i-1)*num+1;
    n2=i*num;
    x1=0:num-1;
    x2=floor(x1/8)*8; % set 3 of the 8 lsbs to 0
    x2=x2.*x2;
    gl=zeros(10,1);
    gs=zeros(6,1);
    bl=zeros(1,8);
    bs=zeros(1,6);
    err=1e200;
    % get gains offsets
    toffset=offsets16(i)-5;
    if toffset < 0
      toffset=0;
    end
    for l=1:11
```

```
     tlcoef=lcoef16(i)-5;
     if tlcoef <0
       tlcoef=0;
     end
     for m=1:11
      tscoef=scoef16(i)-5;
      if tscoef<0
        tscoef=0;
      end
      for k=1:10
        gl(11-k)=bitget(tlcoef, k);
      end
      for n=1:11
       for k=1:6
         gs(7-k)=bitget(tscoef, k);
       end
       for j=1:num
        for k=1:8
         bl(9-k)=bitget(x1(j), k);
        end
        for k=1:6
         bs(7-k)=bitget(x2(j),k+10);
        end
        % perform AND
        ml=and([gl gl gl gl gl gl gl gl], [bl; bl; bl; bl;
bl; bl; bl; bl; bl; bl]);
        ms=and([gs gs gs gs gs gs], [bs; bs; bs; bs; bs;
bs]);

        % calculate vernier currents
        vl=sum(sum(ml.*ltab, 1)); %linear term
        vs=sum(sum(ms.*stab, 1)); % quadratic term
        tfinalwave(j)=toffset+vl-vs;%final=offset+linear-
quadratic
```

```
     end
     resi=tfinalwave-ideal(n1:n2);
     terr=max(abs(resi));
     if terr < err
         err=terr;
         newoffsets(i) = toffset;
         newlcoef(i) = tlcoef;
         newscoef(i) = tscoef;
         finalwave(n1:n2)=tfinalwave;
     end
    tscoef = tscoef +1;
    end
   tlcoef=tlcoef+1;
   end
  toffset = toffset+1;
  end
end

% make one cycle of sine wave
center=(N-1)/2;
outwave=finalwave';% use actual output wave
q1=outwave+center; % first 90 degree
q2=q1(N/4:-1:1);   % 90-180 degree
q3=center-outwave; % 180-270 degree
q4=q3(N/4:-1:1);   % 270-360 degree
y=[q1 q2 q3 q4];   % one cycle
y1=y;

% repeat 32 cycles and compute SFDR
for i=1:5
 y1=[y1 y1];
end
M=32*N;
```

```
global wi;
wi=blackman(M);
z=fft(y1'.*wi, M);
zl=20*log10(abs(z)+1e-200);
subplot(2,1,1);
plot(y1(1:2*N));
subplot(2,1,2);
plot(zl(1:300));
grid on;
s=sprintf('sfdr=%6.3fdB\n',max(zl(20:50))-max(zl(60:110)));
disp(s);
```

B. RAM VALUES FOR THE 14-BIT NON-LINEAR DAC

The first line is for the first segment.

```
Offsets
 dec              bin
0002 -->0 0 0 0 0 0 0 0 0 0 0 1 0
0805 -->0 0 0 1 1 0 0 1 0 0 1 0 1
1601 -->0 0 1 1 0 0 1 0 0 0 0 0 1
2380 -->0 1 0 0 1 0 1 0 0 1 1 0 0
3136 -->0 1 1 0 0 0 1 0 0 0 0 0 0
3864 -->0 1 1 1 1 0 0 0 1 1 0 0 0
4552 -->1 0 0 0 1 1 1 0 0 1 0 0 0
5199 -->1 0 1 0 0 0 1 0 0 1 1 1 1

5793 -->1 0 1 1 0 1 0 1 0 0 0 0 1
6333 -->1 1 0 0 0 1 0 1 1 1 1 0 1
6812 -->1 1 0 1 0 1 0 0 1 1 1 0 0
7226 -->1 1 1 0 0 0 0 1 1 1 0 1 0
7570 -->1 1 1 0 1 1 0 0 1 0 0 1 0
7840 -->1 1 1 1 0 1 0 1 0 0 0 0 0
8034 -->1 1 1 1 1 0 1 1 0 0 0 1 0
8153 -->1 1 1 1 1 1 1 0 1 1 0 0 1

linear coefficients
0803 -->1 1 0 0 1 0 0 0 1 1
0799 -->1 1 0 0 0 1 1 1 1 1
```

```
0783 -->1 1 0 0 0 0 1 1 1 1
0767 -->1 0 1 1 1 1 1 1 1 1
0742 -->1 0 1 1 1 0 0 1 1 0
0705 -->1 0 1 1 0 0 0 0 0 1
0670 -->1 0 1 0 0 1 1 1 1 0
0619 -->1 0 0 1 1 0 1 0 1 1

0570 -->1 0 0 0 1 1 1 0 1 0
0510 -->0 1 1 1 1 1 1 1 1 0
0447 -->0 1 1 0 1 1 1 1 1 1
0374 -->0 1 0 1 1 1 0 1 1 0
0303 -->0 1 0 0 1 0 1 1 1 1
0229 -->0 0 1 1 1 0 0 1 0 1
0158 -->0 0 1 0 0 1 1 1 1 0
0076 -->0 0 0 1 0 0 1 1 0 0

quadratic coefficients
0000 -->0 0 0 0 0 0
0005 -->0 0 0 1 0 1
0004 -->0 0 0 1 0 0
0012 -->0 0 1 1 0 0
0015 -->0 0 1 1 1 1
0017 -->0 1 0 0 0 1
0025 -->0 1 1 0 0 1
0025 -->0 1 1 0 0 1

0031 -->0 1 1 1 1 1
0032 -->1 0 0 0 0 0
0035 -->1 0 0 0 1 1
0031 -->0 1 1 1 1 1
0034 -->1 0 0 0 1 0
0035 -->1 0 0 0 1 1
0041 -->1 0 1 0 0 1
0038 -->1 0 0 1 1 0
```