

Article

Design of a Parallel Sampling Encoder for Analog to Information (A2I) Converters: Theory, Architecture and CMOS Implementation

Thomas S. Murray, Philippe O. Pouliquen and Andreas G. Andreou *

Electrical and Computer Engineering Department, The Johns Hopkins University, Baltimore, MD 21218, USA; E-Mails: thomasmurray@jhu.edu (T.S.M.); philippe@jhu.edu (P.O.P.)

* Author to whom correspondence should be addressed; E-Mail: andreou@jhu.edu;
Tel.: +1-410-516-8361; Fax: +1-410-516-5566.

Received: 19 December 2012; in revised form: 9 February 2013 / Accepted: 21 February 2013 /
Published: 5 March 2013

Abstract: We discuss the architecture and design of parallel sampling front ends for analog to information (A2I) converters. As a way of example, we detail the design of a custom 0.5 μm CMOS implementation of a mixed signal parallel sampling encoder architecture. The system consists of configurable parallel analog processing channels, whose output is sampled by traditional analog-to-digital converters (ADCs). The analog front-end modulates the signal of interest with a high-speed digital chipping sequence and integrates the result prior to sampling at a low rate. An FPGA is employed to generate the chipping sequences and process the digitized samples.

Keywords: analog to information converter; sub-Nyquist sampling; compressive sensing; parallel ADCs

1. Introduction

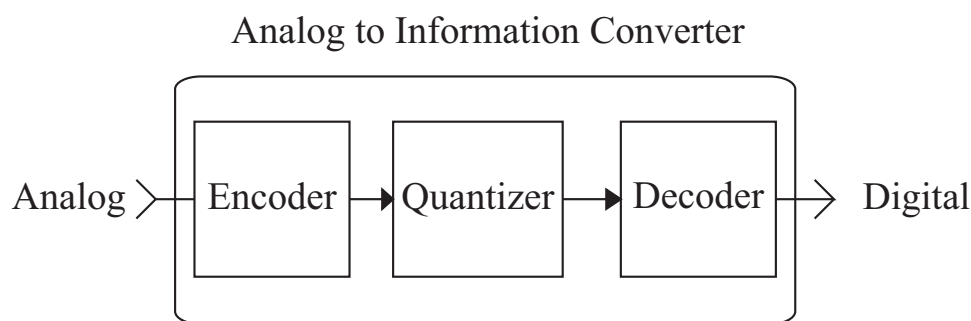
Today, the vast majority of information processing takes place in the digital domain where computation on large scales has become increasingly cheap and convenient. Unfortunately, there are still many applications that struggle to take advantage of digital computation because the signals involved exceed the capabilities of current physical hardware approaches. Analog-to-digital converters (ADCs) are the traditional method for capturing real world signals and converting the information into

digital samples. The two most critical measures of an ADC are its sampling speed, which limits the bandwidth of the signals it can accommodate, and the resolution, which limits the accuracy of the resulting samples. Conventional approaches process signals sequentially and different architectures trade sampling speed for resolution or *vice versa*. This relationship is illustrated in [1], where Walden analyzed the state-of-the-art in ADC technology in 1999, showing empirically that each additional bit in resolution decreases the sampling rate by a factor of two. Since then, these architectures have slowly increased in speed as fabrication processes have continued to improve according to Moore's law, but the basic trend endures.

More recently, the field of compressed sensing (CS) [2] has made a number of theoretical developments that leverage the structure of sparse signals into reduced sampling rates. One application where compressive sampling promises advances in the state-of-the-art is in the design of Analog to Information converters (A2I).

A typical A2I architecture consists of a front-end encoder and a back-end decoder with an analog to digital converter in between (see Figure 1). The purpose of the coder is to transform the input signal into a representation that efficiently preserves information relevant to the function of the system. To achieve this, coders may be designed to exploit prior knowledge of the signals to be processed. Exploiting such prior knowledge has been employed in the design of hardware architectures for non-uniform amplitude [3,4] and spectral quantization [5]. Modern communication systems also do this when high frequency signals are down-converted to base-band prior to low-rate sampling [6].

Figure 1. Analog to information converter architecture.



Although related sampling theories, such as non-uniform sampling [7], have also existed in the literature for many years, the CS framework has led to the appearance of several novel sampling schemes. The modulated wideband converter (MWC) [8–10] and the random demodulator (RD) [11–13] are two of the earliest architectures. The compressive multiplexer (CMUX) [14] is another more recent concept for hardware implementation. Each of these architectures achieves compressive sampling through some combination of parallelism, random modulation or sampling, and prior knowledge about the signals of interest. However, the assumptions on signal models and the details of the sampling process are different.

The RD architecture, described in [13,15], modulates the input signal with a random chipping sequence, and integrates the modulator output before digitizing with a conventional ADC. The practical details of implementing a RD architecture are described in [12]. The RD operates on analog signals composed of sparse linear combinations of known basis functions. The random demodulator concept

was demonstrated using discrete components in [16]. Since the Nyquist rate requirement is satisfied by the high frequency contents of the chipping sequence, the ADC is allowed to operate at sub-Nyquist rates, and only the generation of the chipping sequences and the modulator require high speed hardware components. From a hardware perspective, compressive sampling is desirable because the modulation of the input signal with a binary sequence is simpler and easier to implement at high sampling rates as compared with the sample and hold circuits necessary for interleaved architectures. A single chip sub-Nyquist sampling receiver architecture operating between 100 MHz and 2 GHz was recently reported in the literature [17,18].

The MWC architecture modulates the input signal with multiple chipping sequences in parallel. The modulator output is low-pass filtered before being digitized by conventional ADCs. The MWC operates on analog signals that comply with the multi-band signal model described in [8]. Note that the system modeling is primarily done in the frequency domain (whereas the RD is primarily modeled in the time domain). To the best of our knowledge it is also the only full scale system that has been realized in hardware using discrete components.

In order to process information in applications that require data conversion rates above the practical limits of traditional hardware, we need to consider alternative architectures. Parallel analog to digital converter architectures enable the design of high performance hardware systems in state-of-the-art CMOS technologies that can go beyond the bandwidth and resolution possible in single channel systems. Breaking the data conversion problem into multiple parallel sub-tasks is not a new concept, as exemplified by interleaved ADCs [19] and other random sampling architectures [20,21], which have existed in the literature and in practice for many years. Interleaved ADCs operate multiple ADCs in parallel with offset sampling times such that the resulting low-rate samples can be multiplexed back together into a higher effective sampling rate [22]. Although these architectures resolve the difficulty of designing fast ADCs, the high speed sample-and-hold circuits remain a challenge for high resolution Gsps ADCs.

No single ADC architecture can cover the whole design space. For instance, flash ADC architectures can produce a digitized sample with a non-iterative algorithm at high speeds. Unfortunately, they achieve this through massive parallelism that incurs an exponential cost in silicon real-estate, quickly limiting the achievable bits of resolution. On the other end of the spectrum, sigma-delta ADC architectures can produce high resolution samples using very little silicon area. However, it takes a relatively long time for the result to converge. Algorithmic ADC architectures often exhibit a more balanced tradeoff between bandwidth and resolution. Parallel data converter architectures look to harness the capabilities of multiple individual data converters to boost the overall system performance, generally by increasing the effective sampling rate while maintaining a resolution superior to that of the fastest flash architectures.

In this paper, we present a sampling architecture for compressed and Nyquist sampling [23] and provide experimental results from a fabricated system in 0.5 μm CMOS technology. Even though we aim for reasonably high speed and resolution, our goal is to explore hardware CS sampling architecture ideas and hence our system is not fabricated in a state-of-the-art CMOS process. The coder (front end) architecture presented in this paper is inspired by both the RD and MWC schemes. We employ the concept of chipping and integrating from the RD, and the concept of multiple parallel channels from the MWC. The time domain modeling is also derived from the RD. The hardware combines multiple

resources in parallel and is flexible enough to leverage additional signal structure, allowing information to be captured at a low data rate. The system is highly configurable and illustrates situations that mirror the behavior of interleaved ADC architectures as well as compressed sensing architectures.

2. Sampling Architecture

In the analog to information converter framework, signals are first processed in the analog domain before being quantized by analog to digital converters (ADC) [20]. Additional signal processing in the digital domain may be required to recover the original input signal in a usable form. Such a situation occurs when analog signal processing is performed to optimize the performance of the analog to digital conversion.

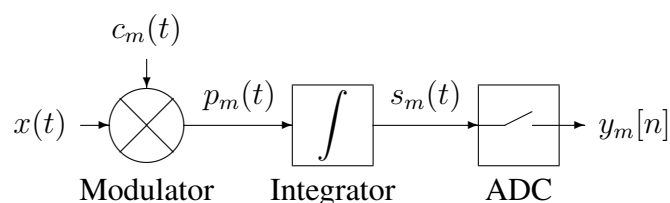
Note that conventional data acquisition systems already follow this general framework in that they almost invariably perform simple transformations of the input signal, such as gain and offset correction or non-linear amplitude compression. These transformations are reversible in the sense that they can be undone with only knowledge of the transformation that was applied. However, irreversible transformations are also performed, the most common being low-pass filtering to satisfy the Nyquist criterion for the ADC.

In the specific implementation of our sampling architecture described herein, the analog coding consists in modulating the input signal with a high-speed chipping sequence and filtering the modulator output. The filter output is then digitized by the Nyquist ADC at a relatively low rate. When multiple parallel channels are used, each channel will normally use a different chipping sequence, so that the ADC outputs are not duplicated, unless some explicit redundancy is desired.

2.1. The Analog Processing Channel

Our implementation of the analog processing channel is shown schematically in Figure 2. The input signal $x(t)$ is multiplied with a digital chipping sequence $c_m(t)$ to produce a modulated signal $p_m(t)$, which in turn is integrated over a finite time period to produce the channel output signal $s_m(t)$. The channel output is sampled and digitized by the ADC, producing a digital sample $y_m[n]$.

Figure 2. Block diagram of the channel.



For an analog front-end using M channels, the subscript m refers to the channel index and takes values from the set $\{1, \dots, M\}$. The integration period T_s is also the sampling period of the ADC, so we define the discrete time index n such that $t = nT_s$, and hence

$$y_m[n] = s_m(nT_s)$$

Since each chipping sequence $c_m(t)$ is a finite digital sequence, it is piecewise constant and can be described by a finite set of N bits. The integration period T_s does not need to be the same as the chipping sequence length, but we do require that it be an integer multiple of T_b , the duration of a chipping sequence bit. Additionally, since each integration period operates in essentially the same way, we can restrict our analysis to the first integration period ($0 \leq t < T_s$) and let $T_s = NT_b$ without loss of generality. Therefore, let

$$c_m(t) = \alpha_i^{(m)} \quad \text{for } (i-1)T_b \leq t < iT_b$$

where $\alpha_i^{(m)}$ takes values from the set $\{-1, +1\}$, and $i = 1, \dots, N$ indexes those values.

The modulator output is just the product of the input signal and the chipping sequence, so

$$p_m(t) = c_m(t)x(t)$$

Therefore, the output of a single integration period is

$$y_m = \int_0^{T_s} p_m(t)dt = \int_0^{T_s} c_m(t)x(t)dt$$

(As we are only considering the first integration period, we have dropped the discrete time index for simplicity.)

Since $c_m(t)$ is piecewise constant,

$$y_m = \sum_{i=1}^N \alpha_i^{(m)} \int_{(i-1)\frac{T_s}{N}}^{i\frac{T_s}{N}} x(t)dt \quad (1)$$

Equation 1 shows that the digitized samples can be broken down into a linear combination of the channel chipping sequence and short-term time integrations of the input signal $x(t)$ over the bit period T_b . If the duration of the bit is too long, any high-frequency components of the input signal will be lost or aliased. Therefore, the rate at which the chipping sequences alternate, $\frac{1}{T_b}$, determines the effective input signal bandwidth that can be resolved by the system.

To simplify the expression for y_m , we define the auxiliary discrete time signal $\mathbf{z}[i]$, such that

$$\mathbf{z}[i] = \int_{(i-1)\frac{T_s}{N}}^{i\frac{T_s}{N}} x(t)dt = \int_{(i-1)T_b}^{iT_b} x(t)dt$$

and

$$y_m = \sum_{i=1}^N \alpha_i^{(m)} \mathbf{z}[i] \quad (2)$$

We also define the vector notation of the bit sequence

$$\mathbf{a}_m = (\alpha_1^{(m)}, \dots, \alpha_N^{(m)})^T$$

and the set of samples

$$\mathbf{y} = (y_1, \dots, y_M)^T$$

Now we can express the set of samples obtained from the front-end after the first integration period as:

$$\underset{M \times 1}{\mathbf{y}} = \underset{M \times NN \times 1}{\mathbf{C}} \underset{N \times 1}{\mathbf{z}}, \quad (3)$$

where $\mathbf{C} = (\mathbf{a}_1, \dots, \mathbf{a}_M)^T$.

If the duration of the bit is sufficiently short, then the input signal $x(t)$ can be reasonably approximated by the discrete time estimate $\hat{x}[i]$, where

$$\hat{x}[i] = \frac{\mathbf{z}[i]}{T_b} \approx x((i - \frac{1}{2})T_b) \quad (4)$$

Recovery of the digitized input signal is therefore a matter of solving Equation 3 for $\mathbf{z}$.

2.2. System Configurations

The relationship between the output of a channel and the channel's chipping sequence expressed in Equation 1 leads immediately to several interesting and instructive system configurations.

Case 1: $N = M$. If $\mathbf{C}$ is the identity matrix, then only one channel would be sampling the input during each bit period. If the chipping sequences were composed of 1's and 0's, then it would be easy to observe that the output of the m th channel is simply the integral of the i th bit period. Moreover, because the number of channels (M) is the same as the number of bit periods (N), when the parallel samples from the channels are interleaved in the right order, the original signal can be estimated. This is essentially how an interleaved ADC arrangement operates.

To get around the fact that we have arbitrarily restricted the composition of the chipping sequences to contain only ± 1 , we can achieve the same effect by setting

$$\alpha_i^{(m)} = \begin{cases} +1 & i = m \\ -1 & i \neq m \end{cases}$$

Applying these chipping sequences to Equation 2 yields

$$y_m = 2\mathbf{z}[m] - \sum_{i=1}^N \mathbf{z}[i]$$

where we have added and subtracted an additional $\mathbf{z}[m]$ term to complete the summation over N . As $N = M$ in this case, the term $\mathbf{z}[m]$ corresponds to the $+1$ chipping coefficient, while all other $\mathbf{z}[i]$, for $i \neq m$, are modified with a -1 coefficient. Solving for $\mathbf{z}[m]$ yields,

$$\mathbf{z}[m] = \frac{1}{2}y_m + \frac{1}{2}T_s\bar{x}$$

where

$$\bar{x} = \frac{1}{T_s} \int_0^{T_s} x(t)dt = \frac{1}{T_s} \sum_{i=1}^N \mathbf{z}[i]$$

Hence, although each ADC is only sampled at a rate of $f_s = \frac{1}{T_s}$, the effective sampling rate is Nf_s .

For other cases where $N \neq M$, it is sufficient that $\mathbf{C}$ be invertible, so that $\mathbf{z}$ can be recovered according to

$$\mathbf{z} = \mathbf{C}^{-1}\mathbf{y}$$

Under these conditions, the chipping sequences form a basis set. Although $\mathbf{C}^{-1}$ can be precomputed, so that the signal recovery requires only a matrix-vector multiplication, there is very little apparent advantage to this approach over the previous interleaving method.

Case 2: $M > N$. If the number of channels M is greater than the number of bits in the chipping sequence N , then Equation 3 is over-determined. A pseudo-inverse of $\mathbf{C}$ is then needed to recover $\mathbf{z}$ from $\mathbf{y}$, but the pseudo-inverse can also be precomputed. The advantage of this approach is that the added redundancy may improve the robustness of the system.

Case 3: $M < N$. In general, this corresponds to an under-determined system and we would not be able to recover $\mathbf{z}$ uniquely. In this setting, $\mathbf{z}$ can still be recovered if additional information about the input signal is known. For example, if $x(t)$ meets appropriate sparse structure assumptions, then we can use pseudo-random chipping sequences, and the situation reduces to the well studied L_1 optimization problem addressed in [2,12].

2.3. System Extensions

In practice, the chipping sequences $c_m(t)$ are periodic so that they can be stored in a finite amount of space in a digital controller. However, the chipping sequence period T_c can be different from the integration period T_s , so that $\mathbf{C}$ changes from one integration interval to the next. All that is required for reconstruction of the signal in a given integration interval is for the $\mathbf{C}$ of that integration interval to be known to the digital controller.

Note that both T_s and T_c must be integer multiples of the bit period T_b , and that the chipping sequence is also periodic in $2T_c, 3T_c$, etc. Therefore, without loss of generality, we can redefine the chipping sequence period such that $T_c = kT_s$ for some integer k , where k is the number of distinct versions of $\mathbf{C}$.

Thus far we have assumed that the signal in a given time interval is reconstructed solely from the ADC outputs of a single integration interval. However, for some applications, it may be advantageous to reconstruct the signal using the ADC outputs from several (e.g., k) integration intervals. For instance, suppose that the input signal is known to be periodic in T_s . Then, the M chipping sequences can be applied by a single channel sequentially over M integration intervals, and the same set of M ADC outputs needed for reconstruction will be obtained but with a substantially reduced amount of hardware.

In general, we can reduce the number of channels by concatenating chipping sequences over several integration intervals and using the ADC outputs from these integration intervals in the reconstruction. However, the properties of the signal must not change substantially over these integration intervals, and the reconstruction algorithm must take into account the time shift between ADC outputs obtained from different integration intervals. This allows the reduction of the physical complexity of the sampling architecture at the cost of constraining how quickly the input signal characteristics are permitted to evolve in time.

3. Analog Hardware Implementation

A complete sampling system can be built with discrete components, using discrete ADCs and a field programmable gate array (FPGA) to implement the digital control and chipping sequence generation, as shown in Figure 3. The problem with this approach is that the analog front-end channels require many components to implement them with discrete parts.

To address this problem, we have designed and fabricated an analog front-end custom integrated circuit in a 0.5 μm CMOS process. The chip contains all the analog front-end components except for

the ADCs (which we plan to integrate in future versions). The front-end was designed to operate from 20 kHz up to 200 MHz. The current design has 8 channels, but multiple chips can be operated in parallel for cases where $M > 8$. A micrograph of the fabricated chip is shown in Figure 4. The active chip area is 0.66 mm^2 (the remaining area is consumed by the I/O pads, wiring and fill).

Figure 3. System block diagram.

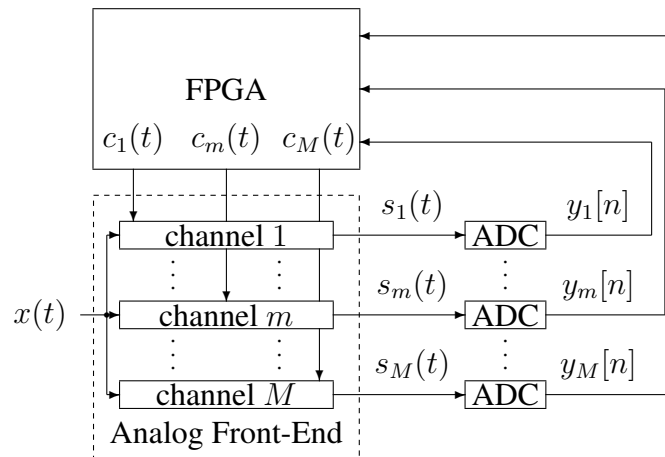
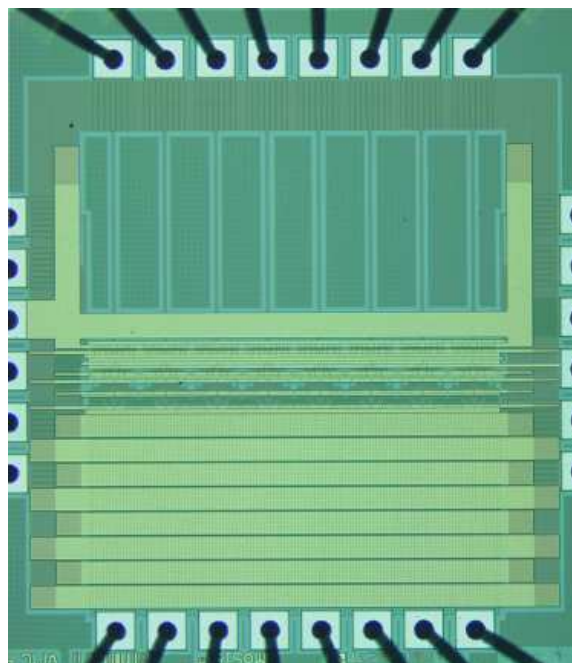


Figure 4. A micrograph of the fabricated $1.5 \text{ mm} \times 1.5 \text{ mm}$ chip.

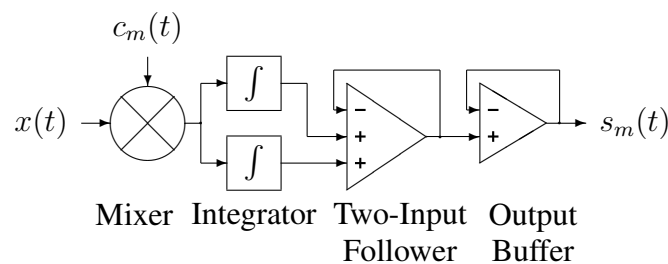


3.1. Channel Pipeline

The proper operation of the front-end requires that the integration periods follow one another without interruption. In practice the integration time interval needs to be followed by a hold interval to allow the ADC's sampling circuit to acquire the integrator's output, and then a reset interval to reset the integrator to its starting value. The hold interval in particular can be quite long, because commercial ADCs normally include track-and-hold circuits whose performance is on par with the conversion circuits.

We therefore designed our front-end channel using two integrators operating in tandem. While one integrator is integrating the input signal, the other is holding its value for the external ADC. Each integrator is reset just prior to switching back to integration mode, so that the sum of the reset interval and the hold interval is equal to the integration interval. This allows the output of one of the integrators to be available to the external ADC for almost a full integration period. A two-input voltage follower is used to select which integrator output is sent to the ADC. This arrangement is shown schematically in Figure 5.

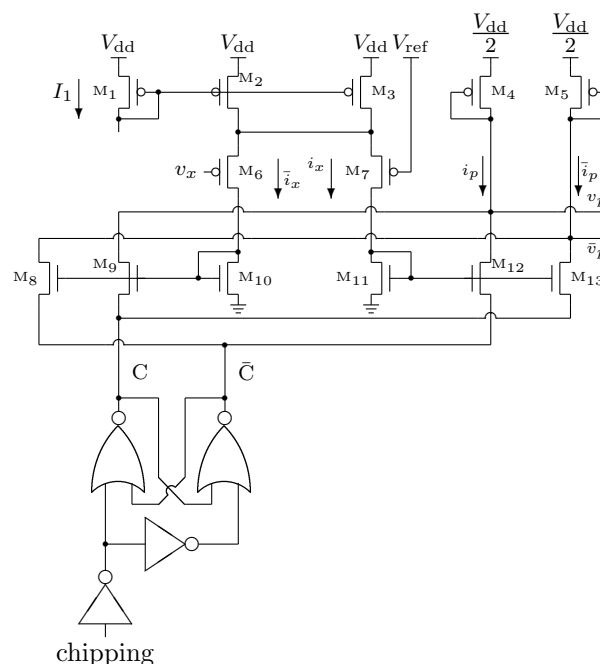
Figure 5. Channel component block diagram.



3.2. Modulator Design

Since the input signal is analog, but the chipping sequence is binary, the modulator does not need to be implemented with a full analog multiplier (e.g., Gilbert multiplier). Rather, it is sufficient to invert $x(t)$ and select $+x(t)$ or $-x(t)$ according to whether $c_m(t)$ is 1 or 0 (0 being the binary representation for a chipping sequence value of -1).

Figure 6. Modulator circuit schematic.



The schematic of the modulator is shown in Figure 6. The input signal $x(t)$ is first converted from a single-ended voltage v_x to a differential current $i_x - \bar{i}_x$ by the PMOS source-coupled pair (M_6 – M_7).

source-coupled pair (M_{19} – M_{20} , or M_{21} – M_{22}). The resulting current i_s is fed through a switch to the appropriate integration capacitor (C_L or C_R , respectively).

This results in two output signals, v_{sL} and v_{sR} , one from each integration capacitor. The signals controlling the switches are arranged to achieve a ping-pong integration scheme such that only one of the two integrators is integrating at any given moment. The other integrator is in the hold mode and can be reset by asserting the chip's reset input without affecting the operation of the first integrator.

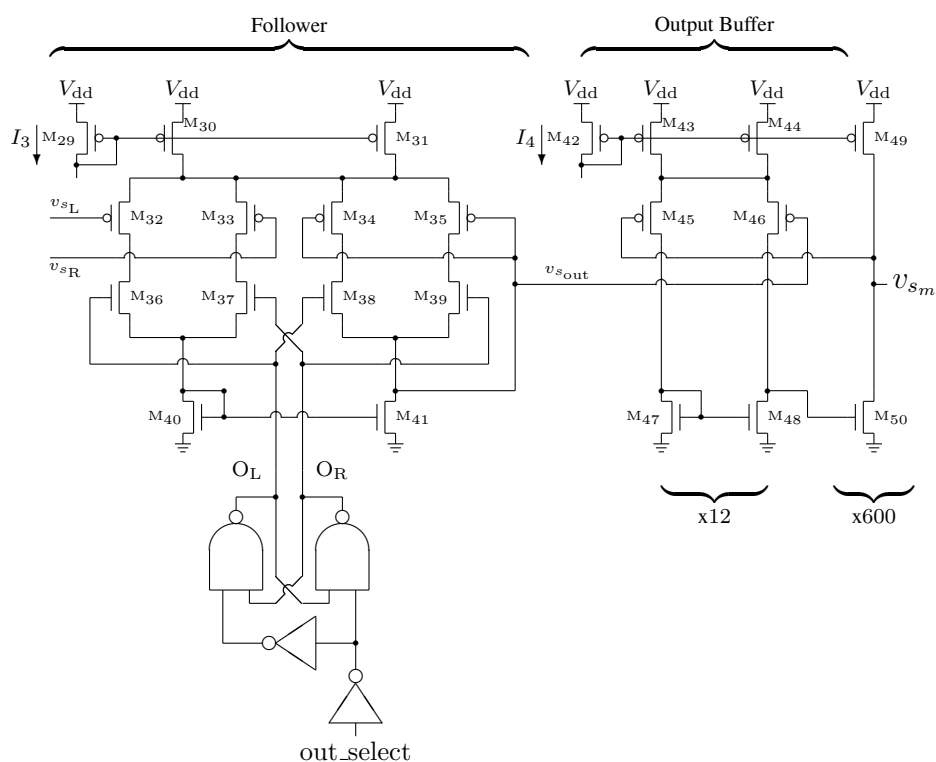
Due to the inverting nature of the PMOS control switches (M_{23} – M_{26}), the integrator digital control logic employs a set-reset latch configured so that the internal select signals S_L and S_R are never overlapping in the low state. This ensures that charge is being deposited on at least one capacitor at all times so that there are no interruptions in the sampling process.

Similar to the modulator circuit, two tail transistors were used for the integrator circuits so that the average current density in each transistor in the integrator is the same. The bias current I_2 is used to control the integration range of the front-end. For long integration times or large amplitude input signals, I_2 can be reduced to prevent overflowing the charge capacity of the integrators. For short integration times or small amplitude input signals, I_2 can be increased to maximize the integrators' output amplitude. The combined die area consumed by the mixer and integrator for each channel is $14,000 \mu\text{m}^2$.

3.4. Two-Input Follower and Output Buffer Design

The next stage in the signal chain is the two-input follower shown in Figure 8. This circuit is derived from the five transistor operational transconductance amplifier (OTA). When the output select input is high, v_{sR} is copied to the output v_{sout} , otherwise v_{sL} is copied out.

Figure 8. Two-input follower and output buffer circuit schematics.



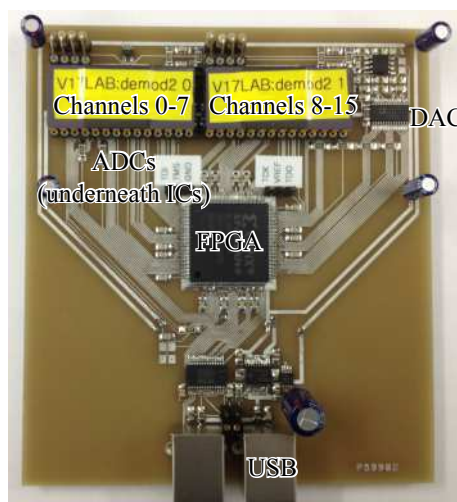
The digital control logic for the follower is also configured so that the internal select signals O_L and O_R are never overlapping in the low state. These signals interface with a set of NMOS switches (M_{36} – M_{39}), ensuring that only one capacitor node at a time is selected. The die area consumed by the follower for each channel is $7600 \mu\text{m}^2$.

In order to achieve a stable output and drive the integrated signal off-chip to be sampled by discrete ADCs, an operational amplifier (OPAMP) was used to buffer the signal. The second section of Figure 8 shows the circuit schematic for the OPAMP. The output of the analog channel is v_{sm} and was designed to drive a load of approximately 50 pF at 50 MHz. This load is the combined capacitance of the output pad, the chip package and the input of an ADC (e.g., AD7276). In order to achieve this drive capability, each transistor in the first (input) stage of the OPAMP (M_{42} – M_{48}) is composed of 12 individual transistors in parallel, and each transistor in the second (output) stage (M_{49} – M_{50}) is composed of 600 individual transistors in parallel. The resulting capacitive load of the output stage is large enough that the OPAMP does not require any additional compensation. The OPAMP consumed $61,000 \mu\text{m}^2$ of die area for each channel. As the capacitors begin their integration after being reset to ground, the OTA and OPAMP circuits were selected for their ability to accurately drive the channel output as close to ground as possible at the required operating frequency.

4. Testing

A circuit board with all of the components necessary to support and test the fabricated front-end chip at low speed was developed and is shown in Figure 9. This test board supports a maximum of two front-end chips for a total of sixteen analog channels.

Figure 9. Picture of the sampling system prototype.



The board also contains 16 individual ADCs (located under the front-end chips), a single DAC, and an FPGA. Each ADC is a 12-bit 3 Msps converter (AD7276) and is dedicated to digitizing the output of a single analog channel. The DAC is a 14-bit 200 Msps (AD9744 with AD8041 OPAMP) converter that supplies a programmable analog input signal to the front-end chips. The FPGA (XC3S50AN) coordinates the operation of the front-end chips and the data converters. It is responsible for supplying

chipping sequences to the front-end chips as well as the input signal in digital form to the DAC. It is also responsible for collecting the ADC samples and relaying them to the host computer via a USB interface.

4.1. Channel Calibration

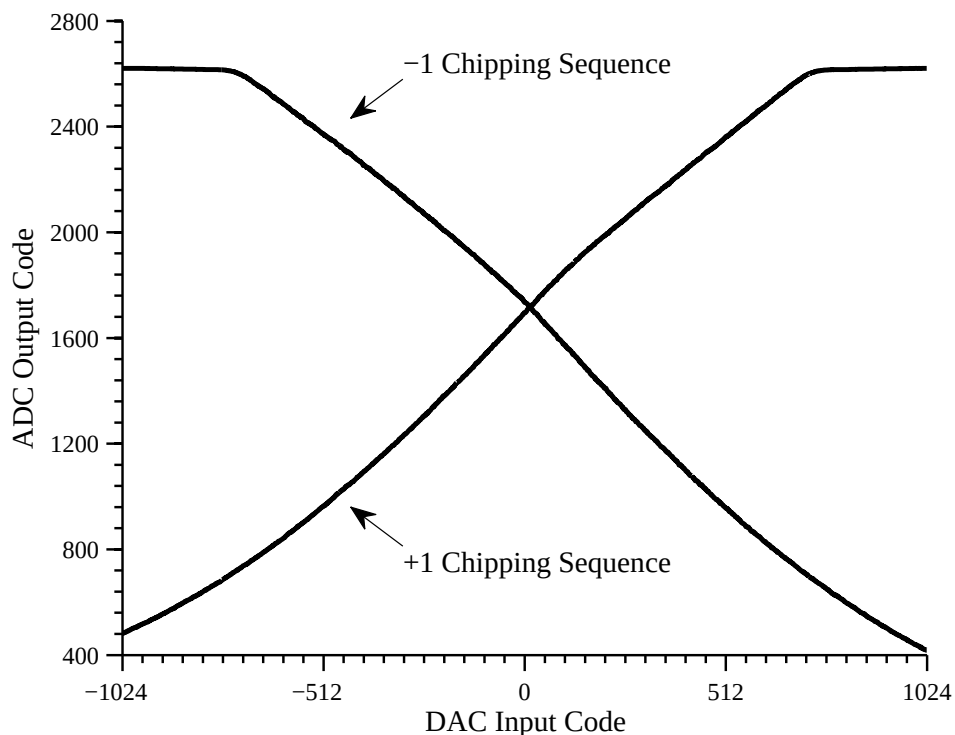
Before applying actual analog input signals, we first calibrated both integrators (left and right) in each of the sixteen channels. These calibration steps were all performed using constant analog inputs and chipping sequences.

4.1.1. Bias Current Adjustment

The front-end chip uses four bias currents, I_1 through I_4 , to tune the operation of the mixer, integrators, follower and output buffer respectively. I_1 , I_3 and I_4 were left at their nominal value of $10\ \mu\text{A}$, while I_2 was tuned to the integration period T_s of $1\ \mu\text{s}$.

To tune I_2 , analog inputs and chipping sequences must be applied in such a way as to obtain maximum integrator output. This was accomplished using a constant maximal DAC output (DAC set to +8191) with a constant chipping sequence of +1, as well as a constant minimal DAC output (DAC set to −8192) with a constant chipping sequence of −1. The bias current was then progressively reduced until the integrators were no longer saturating. The final value of I_2 was $0.1\ \mu\text{A}$.

Figure 10. Channel transfer characteristic.



4.1.2. Input Dynamic Range

The front-end chip uses a source-coupled pair to convert the analog input to a current. Since the circuit does not make use of source resistors, the linear region is fairly narrow. The linear region was

characterized by sequentially applying constant analog input signals with constant chipping sequences (both +1 and −1) for all 2^{14} possible DAC codes (−8192 to +8191).

The result, reported in Figure 10, shows that the input saturates at the edges of the range −1024 to +1024. Therefore we chose the input range −512 to +512, which represents an input dynamic range of approximately ± 30 mV, and avoids the saturation region.

4.1.3. Mixer Symmetry

Ideally, the operation of the front-end should be symmetric, in that if an analog input of v is applied with a +1 chipping sequence, the resulting integrator output should be the same as for an analog input of $-v$ and a chipping sequence of −1. However, we noted during the input dynamic range test that the operation of the front-end was not symmetric due to transistor mismatch in the mixer. Two causes for the asymmetry were identified.

The first cause is the mismatch between the two transistors forming the source-coupled pair. This introduces an input offset ϕ_{in} , which can be compensated by adjusting the V_{ref} for each channel. Since this was not an option for the current front-end implementation, we instead corrected for the input offset by adding a constant to the DAC values. This prevented us from operating multiple channels in parallel, and was not by itself sufficient to eliminate all the asymmetry in many channels.

The second cause is the mismatch between the transistors composing the current mirrors. These cause the current gain G_{+1} for a +1 chipping sequence to be different from the current gain G_{-1} for a −1 chipping sequence. Unfortunately, these mismatches cannot be trivially corrected.

The input offset and current mirror gains were estimated for each integrator by applying a constant maximal or minimal analog input and constant chipping sequence of +1 or −1. The four corresponding integrator outputs are given by

$$\begin{aligned} y_{min_1} &= \phi_{out} - G_{-1}(\phi_{in} + 512) \\ y_{min_2} &= \phi_{out} + G_{+1}(\phi_{in} - 512) \\ y_{max_1} &= \phi_{out} + G_{+1}(\phi_{in} + 512) \\ y_{max_2} &= \phi_{out} - G_{-1}(\phi_{in} - 512) \end{aligned}$$

where ϕ_{out} is the output offset. From these relationships, we can extract the offsets and current mirror gains using

$$\begin{aligned} G_{+1} &= \frac{y_{max_1} - y_{min_2}}{1024} \\ G_{-1} &= \frac{y_{max_2} - y_{min_1}}{1024} \\ \phi_{in} &= 512 \frac{y_{max_1} - y_{max_2} + y_{min_2} - y_{min_1}}{y_{max_1} - y_{min_1} + y_{max_2} - y_{min_2}} \\ \phi_{out} &= \frac{1025(y_{max_2} - y_{min_1}) + 1023(y_{max_1} - y_{min_2})}{4096} \end{aligned}$$

In practice however, due to the non-linearity of the source-coupled pair, we first obtained the input offset ϕ_{in} through an iterative process, in which we computed an input offset from the above equation, and then reapplied the test patterns with the new input offset. This process converged after three to four iterations.

The values shown in Table 1 are for channel 4L (the left capacitor of the fourth channel) after convergence of the input offset. For this integrator, $G_{+1} = 1.36$, $G_{-1} = 1.37$ and $\phi_{in} = 36$. Several other

integrators exhibited near perfect symmetry, but many of the remaining integrators exhibited pronounced differences in the current mirror gains.

Table 1. Symmetry Characterization Table.

DAC input	Chipping	Output	Channel 4L
+512	−1	$y_{\min_1}$	995
−512	+1	$y_{\min_2}$	998
+512	+1	$y_{\max_1}$	2405
−512	−1	$y_{\max_2}$	2411

The transfer characteristics for channel 4L (after input offset correction) is shown in Figure 10. The curve with the positive slope was measured with a constant chipping sequence of +1, while the curve with the negative slope was measured with a chipping sequence of −1. Close observation shows that the transfer characteristics are slightly non-linear so that the two curves do not cross at a DAC input of zero.

4.2. Chipping Sequence Sensitivity

With a constant analog input signal, the integrator output should be insensitive to the order of the chipping sequence bits. For instance, any chipping sequence with zero mean should produce the same integrator output. However, when we switched from using the two possible constant chipping sequences to other chipping sequences, we found that the front-end did not behave as expected.

Figure 11. Channel 4L output for two circularly shifted chipping sequences.

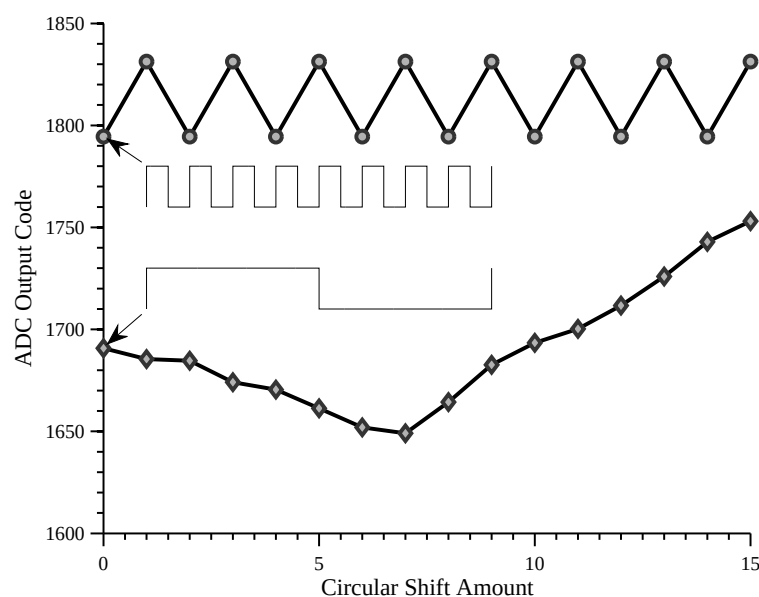


Figure 11 shows the output of channel 4L for two chipping sequences and a zero constant analog input signal (with input offset correction applied). The first chipping sequence is a square wave with period equal to the integration period ($T_s = 1 \mu s$), and the second is a square wave with a period of one

eighth of the integration period ($\frac{T_s}{8}$). Each chipping sequence was circularly shifted by one sixteenth of the integration period ($\frac{T_s}{16} = T_b$ or one bit period of the fast square wave).

4.2.1. Circular Shift Sensitivity

For both chipping sequences, the integrator output is sensitive to the circular shift, even though the circular shift does not affect the proportion of +1 to −1 bits. This is probably caused by leakage from the integration capacitor.

4.2.2. Frequency Sensitivity

Although both chipping sequences have zero mean, the integrator output is always lower for the chipping sequence with fewer transitions. This is probably due to charge injection when the chipping input toggles.

4.2.3. Output Offset Correction

The chipping sequence dependent behavior of the front-end chip is particularly problematic with respect to the output offset ϕ_{out} . Normally, ϕ_{out} can be measured directly by applying a zero constant analog input, and recording the integrator output for each chipping sequence of interest. The recorded output offset can then be subtracted from data subsequently collected with the analog input signals of interest.

Alternatively, the chipping sequence and its complement can be applied with the analog input signal of interest, and the two integrator outputs subtracted. Since the offset being corrected is a property of a given channel, the chipping sequence and the its complement sequence must be applied sequentially to the same channel (this is the $k = 2$ case described at the end of Section 2.3). Although this requires that the input signal be applied twice, in practice, this gave better results than using the recorded output offsets.

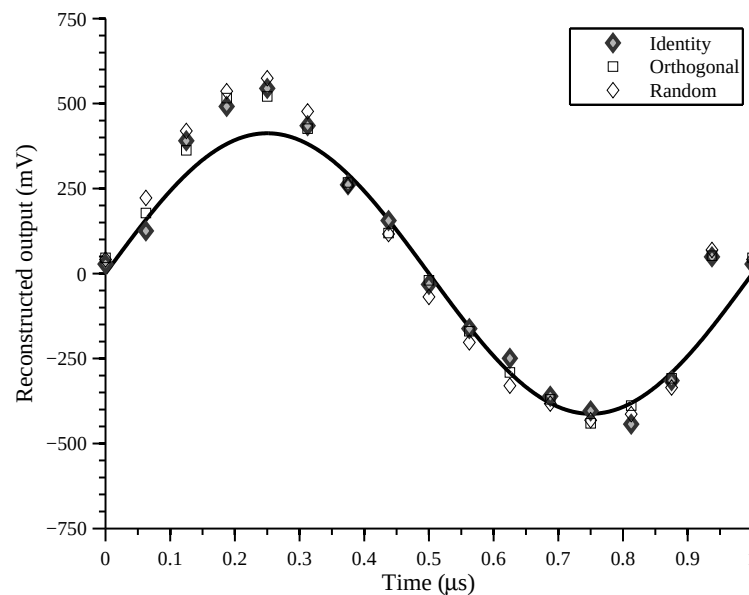
5. Signal Sampling and Reconstruction

Due to the channel dependent offsets, the system was emulated by sequentially applying the chipping sequences to channel 4L. To match the implemented system limitation of sixteen channels, the chipping sequence sets were constrained to at most 16 distinct chipping sequences of 16 bits each. Since the external ADCs only operated reliably at a maximum of 1 MHz, this resulted in a maximum chipping sequence frequency of 16 MHz even though the chips were successfully tested at frequencies up to 256 MHz.

5.1. Reconstruction with Complete Chipping Sequences

Figure 12 illustrates the reconstruction of a 1 MHz sine wave using three different sets of chipping sequences. Each of the three sets includes a full sixteen chipping sequences (and their complements) such that the matrix can be directly inverted to recover the input signal.

Figure 12. 1 MHz sine wave input with output offset corrected using the full chipping sequence complement.



The first set of chipping sequences $\mathbf{C}$ is the identity matrix in which the zero entries have been replaced with -1 . The second $\mathbf{C}$ consists of square waves with periods of 0 , T_s , $\frac{T_s}{2}$, $\frac{T_s}{4}$ and $\frac{T_s}{8}$. All possible phase shifts are represented to form an orthogonal basis. The third $\mathbf{C}$ consists of sixteen random chipping sequences. Since each chipping sequence set in this case is invertible, the reconstructed input is fairly close to the original as shown in Figure 12, and these sequences all perform about the same.

The total reconstruction signal-to-noise ratio (SNR) is reported in the second column of Table 2 and was estimated by comparing the power of input signal to the power of the reconstructed signal in the time domain. However, the error is not entirely attributable to the front-end chip. For instance, a portion of the error is due to the approximation Equation 4. Another portion of the error is due to the piece-wise constant nature of the DAC output, which was operating at 16MHz.

To eliminate some of these extraneous sources of reconstruction error, we computed the ideal value of the ADC samples y given a quantized piece-wise constant input signal representing the DAC output. The simulation was parametrized by gain, offset and phase, and the maximum SNR obtained is reported in the third column of Table 2. The gain and offset parameters account for the frequency response effect shown in Figure 11, and the phase parameter accounts for the small time difference between when the DAC updates its output and when the chipping sequences change state.

Table 2. 1 MHz Sine Wave Performance.

Chipping Sequence	Total recon- struction SNR	Front-end chip SNR
Identity	9.64 dB	10.18 dB
Orthogonal	11.92 dB	17.22 dB
Random	11.54 dB	11.65 dB

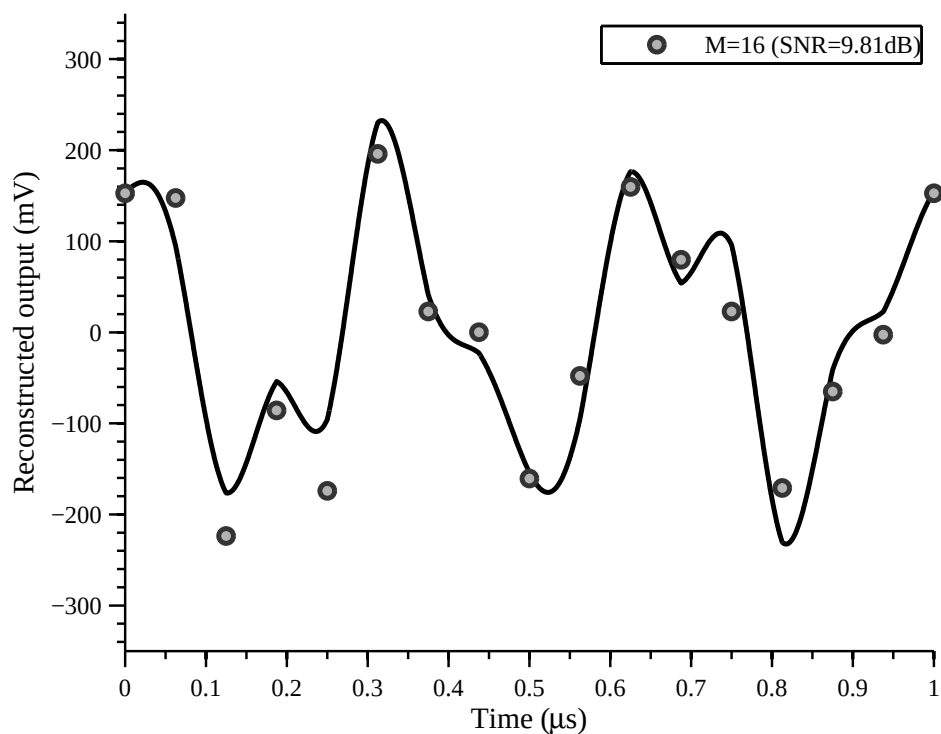
5.2. Reconstruction with Reduced Chipping Sequences

To explore the ability of the sampling system to reconstruct sparse analog signals, we considered a basis set $\Psi = \{\psi_i(t)\}$ composed of both sine and cosine elements:

$$\psi_i(t) = \begin{cases} \cos(2\pi i \frac{t}{T_s}), & \text{for } i = 0, \dots, \frac{N}{2} - 1 \\ \sin(2\pi(i - \frac{N}{2}) \frac{t}{T_s}), & \text{for } i = \frac{N}{2} + 1, \dots, N \end{cases}$$

The following results were generated using an input signal created from the sum of two randomly chosen basis elements, each given a random amplitude ($N = 16$). The resulting signal was programmed into the DAC (at 16 Msps) and the output of the sampling system was used to reconstruct the signal. Figure 13 shows the input signal and the resulting reconstruction achieved with the random chipping sequence set described previously.

Figure 13. Reconstruction of input signal composed of two randomly chosen basis elements with random amplitudes.



The reconstruction shown in Figure 14 was obtained from a reduced set of $M = 10$ chipping sequences. It is important to realize that the elements of the basis set are continuous functions of time, and therefore, so is $x(t)$. However, because the structure of $x(t)$ is constrained to a finite set of basis elements $\psi_i(t)$, $x(t)$ exhibits sparsity with respect to the basis Ψ , which is key to recovery when $M < N$. A basis pursuit optimization algorithm, described in [24,25], was used to recover the original signal once the samples were transmitted to the host computer.

Figure 14. Compressive reconstruction of input signal composed of two randomly chosen basis elements with random amplitudes.

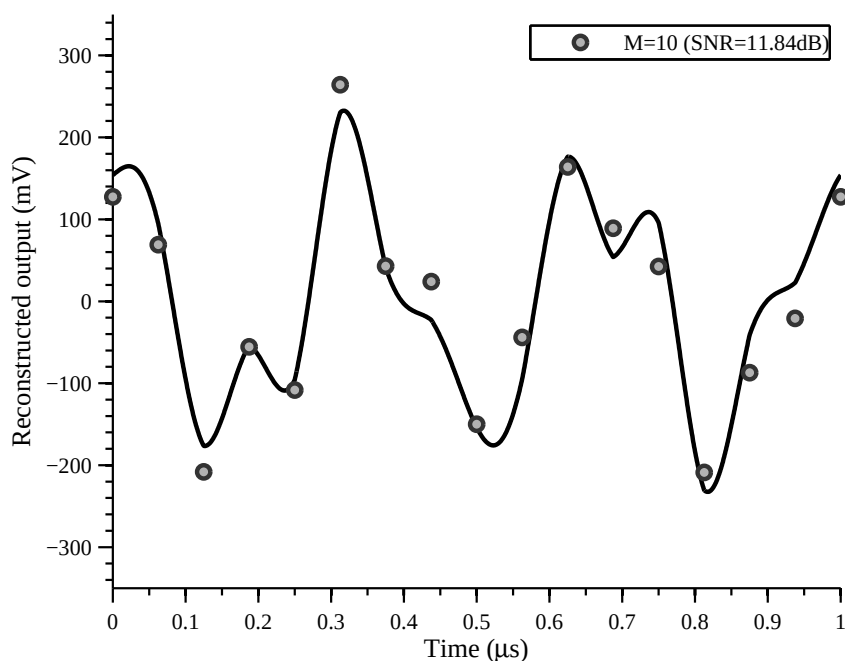
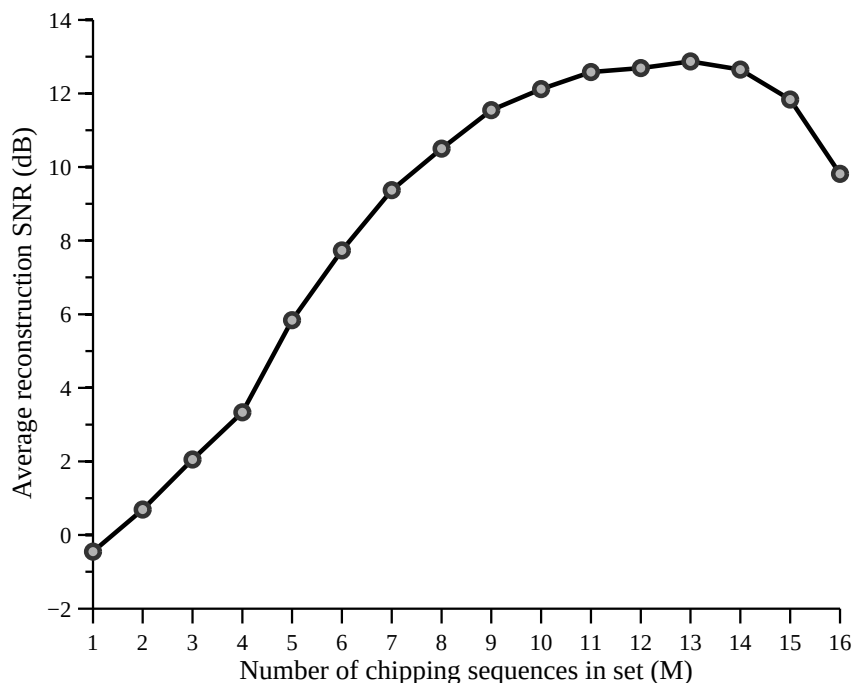


Figure 15. Average reconstruction signal to noise ratio for different numbers of samples.



While a conventional Nyquist rate ADC would require 16 samples to achieve similar results, our architecture demonstrates reconstruction of the signal with similar quality using as few as eight samples. Figure 15 reports the reconstruction SNR averaged over a hundred trials where the signal is reconstructed from 1 to 16 chipping sequences randomly selected from a full set of 16 sequences.

6. Discussion

Today, the vast majority of information processing takes place in the digital domain where computation on large scales has become increasingly cheap and convenient. Unfortunately, there are still many applications that struggle to take advantage of digital computation because the signals involved exceed the capabilities of current physical hardware approaches.

Our performance analysis of the analog front-end provides valuable insights for improvements to be made in the next generation of hardware. Integrating additional discrete components like the channel ADCs into the chip will solve a number of practical issues. Analog signals typically require far more power to drive off chip than digital signals. Replacing the large analog drivers in the existing chip with dedicated ADCs will greatly reduce the complexity of the overall system and provide a more controlled connection between a channel's integrating node and the ADC. This in turn will allow for smoother operation of the ping-pong sampling scheme.

Technology related practical issues that affect the performance of analog to information converters were discussed in previous papers on the implementation of integrated systems [26]. Channel matching is clearly a major obstacle for the highly parallel architecture. Furthermore, our current system model assumes that the channel components are linear, limiting its ability to account for the nonlinear behavior exhibited by many of the fabricated components. Nonetheless, our work in this paper is the first experimental realization of an integrated sampler for an analog to information converter where fabrication and implementation related issues are evident in the experimental data. Further work and development of implementation models [27] for the non-idealities in the system needs to be done before these fabricated systems can achieve their theoretical limits. In future designs, it may be necessary to provide expanded calibration capabilities in addition to redesigning the integrators to provide a more uniform result. It may also be necessary to augment the model to account for some of these nonlinear effects in order to fully explain the reconstruction errors.

Although the prototype system is limited in speed by the fabrication process used and its reliance on external ADCs and digital processing components, it provides an important first step towards a fully integrated system. All the hardware components were designed to allow for the possibility of building data converters in commercial processes with higher effective bandwidths.

Note that the relative simplicity of the circuit elements in the mixer and integrator was intentional and is an important practical consideration for this architecture. Indeed, in order for the architecture to retain its broad applicability, it must necessarily operate near the maximum speed of the fabrication process, independent of which fabrication process is used. This implies that complex circuit elements, such as OPAMPs, cannot be used in the mixer and integrator, because they would limit the speed to such an extent that Nyquist ADCs could be used instead to directly digitize the input signal.

Our D.C. coupled system is complementary to the radio frequency architectures recently reported in [17,18]. We expect that further optimization of the circuit components combined with fabrication in a state-of-the-art process will allow our system to span both ranges of operating frequencies. In the future we envision such systems as solid platforms on which to develop a new generation of efficient data converters that improve upon the bandwidth and resolution constraints of conventional ADCs.

7. Conclusions

We have presented an integrated sampling sub-system fabricated in 0.5 μm CMOS that uses flexible D.C. coupled analog components and programmable digital components to create a highly versatile architecture capable of implementing a wide array of sampling schemes that can sample signals from a few kHz to many hundreds of megahertz. This architecture can be employed in a conventional ADC architecture or a compressive sampling ADC.

There are many applications involving signals that should be digitized but are not digitized, either because they are too fast to sample or because the application requires a higher resolution than existing hardware solutions can provide. The recent developments in the field of compressed sensing suggest that more efficient sampling schemes exist for many classes of signals, but the transition of these ideas to physical systems is still in a nascent state.

Acknowledgment

The late Dennis M. Healy, program manager for the DARPA A2I program has sparked our interest in analog to information converters and compressive sampling. This work was supported by a Johns Hopkins University Whiting School of Engineering and Applied Physics Laboratory collaborative research grant. We would also like to thank Yonina Eldar for a stimulating discussion during her recent visit to JHU.

References

1. Walden, R.H. Analog-to-digital converter survey and analysis. *IEEE J. Sel. Areas Commun.* **1999**, *17*, 539–550.
2. Candès, E.J.; Wakin, M.B. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30.
3. Max, J. Quantizing for minimum distortion. *IRE Trans. Inf. Theory* **1960**, *6*, 7–12.
4. Paschalidis, N.P.; Andreou, A.G.; Sarris, E.T. A CMOS analog-digital integrated circuit for charged particle spectrum measurements. *IEEE Trans. Nucl. Sci.* **1991**, *1*, 634–638.
5. Furth, P.M.; Andreou, A.G. A design framework for low power analog filter banks. *IEEE Trans. Circ. Syst. I: Fundam. Theory Appl.* **1995**, *42*, 966–971.
6. Eldar, Y.C.; Oppenheim, A.V. Filterbank reconstruction of bandlimited signals from nonuniform and generalized samples. *IEEE Trans. Signal Process.* **2000**, *48*, 2864–2875.
7. Jerri, A.J. The Shannon sampling theorem—Its various extensions and applications: A tutorial review. *Proc. IEEE* **1977**, *65*, 1565–1596.
8. Mishali, M.; Eldar, Y.C. Blind multiband signal reconstruction: Compressed sensing for analog signals. *IEEE Trans. Signal Process.* **2009**, *57*, 993–1009.
9. Mishali, M.; Eldar, Y.C. From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals. *IEEE J. Sel. Top. Signal Process.* **2010**, *4*, 375–391.
10. Mishali, M.; Eldar, Y.C.; Dounaevsky, O.; Shoshan, E. Xampling: Analog to digital at sub-Nyquist rates. *IET Circ. Devices Syst.* **2011**, *5*, 8–20.

11. Kirolos, S.; Laska, J.N.; Wakin, M.; Duarte, M.A.; Baron, D.; Ragheb, T.S.; Massoud, Y.; Baraniuk, R.G. Analog-to-information Conversion via Random Demodulation. In Proceedings of the IEEE Dallas Circuits and Systems Workshop (DCAS), Dallas, USA, 29–30 October 2006.
12. Kirolos, S.; Massoud, Y.; Baraniuk, R.G.; Gilbert, A. Random Sampling for Analog-to-information Conversion of Wideband Signals. In Proceedings of the IEEE Dallas Circuits and Systems Workshop (DCAS), Dallas, USA, 29–30 October 2006.
13. Tropp, J.A.; Laska, J.N.; Duarte, M.F.; Romberg, J.K.; Baraniuk, R.G. Beyond Nyquist: Efficient sampling of sparse bandlimited signals. *IEEE Trans. Inf. Theory* **2010**, *56*, 520–544.
14. Slavinsky, J.P.; Laska, J.N.; Davenport, M.A.; Baraniuk, R.G. The Compressive Multiplexer for Multi-Channel Compressive Sensing. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011.
15. Laska, J.N.; Kirolos, S.; Duarte, M.F.; Ragheb, T.S.; Baraniuk, R.G.; Massoud, Y. Theory and Implementation of an Analog-to-Information Converter Using Random Demodulation. In Proceedings of the 2007 IEEE International Symposium on Circuits and Systems (ISCAS), New Orleans, USA, 27–30 May 2007.
16. Pfetsch, S.; Ragheb, T.S.; Laska, J.N.; Nejati, H.; Gilbert, A.; Strauss, M.; Baraniuk, R.G.; Massoud, Y. On the Feasibility of Hardware Implementation of Sub-Nyquist Random-sampling Based Analog-to-information Conversion. In Proceedings of the 2008 IEEE International Symposium on Circuits and Systems (ISCAS), Seattle, USA, 18–21 May 2008.
17. Yoo, J.; Becker, S.; Loh, M.; Monge, M.; Candès, E.J.; Emami-Neystanak, A. A 100 MHz–2 GHz 12.5x Sub-Nyquist Rate Receiver in 90 nm CMOS. In Proceedings of the IEEE Radio Frequency Integrated Circuits Symposium (RFIC), Montreal, Canada, 16–19 June 2012.
18. Yoo, J.; Becker, S.; Monge, M.; Loh, M.; Candès, E.J.; Emami-Neystanak, A. Design and Implementation of a Fully Integrated Compressed-Sensing Signal Acquisition System. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012.
19. Black, W.C.; Hodges, D.A. Time interleaved converter arrays. *IEEE J. Solid-State Circuit* **1980**, *15*, 1022–1029.
20. Jenq, Y.C. Digital spectra of nonuniformly sampled signals: Fundamentals and high-speed waveform digitizers. *IEEE Trans. Instrum. Meas.* **1988**, *37*, 245–251.
21. Jenq, Y.C. Digital Spectra of Non-uniformly Sampled Signals: Theories and Applications; IV. Measuring Clock/aperture Jitter of an A/D System. In Proceedings of the 7th IEEE Instrumentation and Measurement Technology Conference (IMTC-90), San Jose, California, 13–15 February 1990.
22. Nathawad, L.Y.; Urata, R.; Wooley, B.A.; Miller, D.A.B. A 40-GHz-bandwidth, 4-bit, time-interleaved A/D converter using photoconductive sampling. *IEEE J. Solid-State Circuit* **2003**, *38*, 2021–2030.
23. Murray, T.; Pouliquen, P.O.; Lauritzen, K.; Andreou, A.G. Design of a CMOS A2I Data Converter: Theory, Architecture and Implementation. In Proceedings of the 45th Annual Conference on Information Sciences and Systems (CISS), Baltimore, USA, 23–25 March 2011.
24. van den Berg, E.; Friedlander, M.P. Probing the Pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.* **2008**, *31*, 890–912.

25. Friedlander, M.P.; van den Berg, E. SGPL1: A solver for large-scale sparse reconstruction (version 1.7), 2009. Available online: <http://www.cs.ubc.ca/labs/scl/spgl1/> (accessed on 5 September, 2009).
26. Kirolos, S.; Ragheb, T.S.; Laska, J.N.; Duarte, M.F.; Massoud, Y.; Baraniuk, R.G. Practical Issues in Implementing Analog-to-information Converters. In Proceedings of the 6th International Workshop on System On Chip for Real-Time Applications, Cairo, Egypt, 27–29 December 2006.
27. Ragheb, T.S.; Kirolos, S.; Laska, J.N.; Gilbert, A.; Strauss, M.; Baraniuk, R.G.; Massoud, Y. Implementation Models for Analog-to-information Conversion via Random Sampling. In Proceedings of the 50th Midwest Symposium on Circuits and Systems (MWSCAS), Montreal, Canada, 5–8 August 2007.

© 2013 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).