A Continuous-Time ADC/DSP/DAC System With No Clock and With Activity-Dependent Power Dissipation

Bob Schell and Yannis Tsividis, *Fellow, IEEE*

***Abstract*—A continuous-time system that converts its analog input to a continuous-time digital representation without sampling, then processes the information digitally without the aid of a clock, is presented. Without sampling there is no aliasing, which reduces the in-band distortion power by not aliasing into band out-of-band distortion components. The 8-bit system, fabricated in a 90 nm CMOS process, utilizes continuous delay elements as part of a programmable transversal FIR filter. The input is encoded by a delta modulator without a clock into a series of non-uniformly spaced tokens, which are processed by the digital continuous-time filter and converted to an analog output using a custom DAC that guarantees there are no glitches in the output waveform. All activity is signal driven, automatically affording dynamic power scaling that tracks input activity.**

***Index Terms*—Asynchronous processing, continuous-time digital filter, continuous-time digital signal processing, delay element, delta modulator.**

I. Introduction

CONVENTIONAL digital signal processors (DSPs) operate in discrete amplitude and discrete time. The block diagram of a conventional DSP system front-end, the sampled analog-to-digital converter (ADC), is shown at the top of Fig. 1. The discretization of time is done with a sampling clock at frequency f_S that provides a finite number of samples which can then be stored in a digital medium. The discretization of amplitude, done via the finite-resolution ADC, allows for processing that involves handling only 0s and 1s, affording noise immunity and programmability. The remainder of Fig. 1 qualitatively shows the spectra at various points when the input is a single sinusoid and the sampling clock has a frequency of f_S. The spectrum of the input, seen in Fig. 1(a), contains a single frequency, while the spectrum at the ADC input, seen in Fig. 1(b), contains many frequencies at $kf_S \pm f_{INPUT}$, where k is an integer.[1] As quantization is a nonlinear operation, the output of the ADC will have intermodulation (IM) components where the IM product frequencies arise from all possible combinations of the alias frequencies. Thus the ADC output spectrum seen in Fig. 1(c) contains many frequency components, collectively considered to represent the so-called "quantization noise."

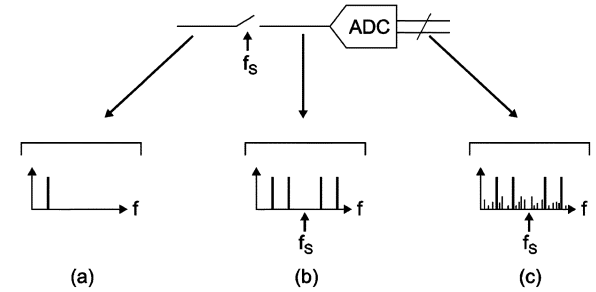


Fig. 1. Spectra of signals involved in a conventional ADC.

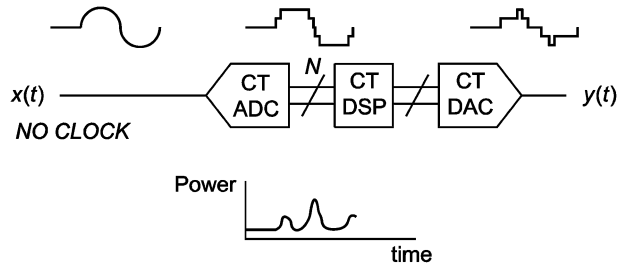


Fig. 2. General block diagram of a CT DSP system, with example waveforms.

If the sampling block in Fig. 1 were removed, then the ADC input will only have one tone in its spectrum (the input frequency). Therefore, the nonlinearity of the quantization process of the ADC will only give rise to harmonic distortion, which appears only at frequencies directly related to the input frequency. Such a construction does not discretize time but still generates a discrete-amplitude signal that may be processed with digital hardware [1], [2]. A system that operates on such a discrete-amplitude, but not discrete-time, signal is called here a continuous-time (CT) ADC/DSP/DAC system, or a CT DSP system for short, and is shown in Fig. 2. A CT DSP system operates on 0s and 1s but the conversion from analog to digital form is done without a clock, as shown. There is no clock in *any* part of a CT DSP system. From the qualitative analysis above, the spectrum of the signal in a CT DSP system will be "cleaner" and will have fewer distortion components at frequencies within the band of interest.

Further benefits that CT DSP systems have over conventional (sampled) systems include a power dissipation that depends drastically on input activity, as shown by the example waveforms in Fig. 2. The CT DSP will only receive new inputs

Manuscript received March 12, 2008; revised July 11, 2008. Current version published November 19, 2008. This material is based on work supported by the National Science Foundation under Grants CCR-02-09109 and CCF-07-01766. This material was presented in part at the 2008 IEEE International Solid-State Circuits Conference.

B. Schell was with Columbia University, New York, NY 10027 USA. He is now with Conexant Systems Inc., Red Bank, NJ 07701 USA (e-mail: rcs2002@columbia.edu).

Y. Tsividis is with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: tsividis@ee.columbia.edu).

Digital Object Identifier 10.1109/JSSC.2008.2005456

[1]Note that at the output of the sampler there is *no* filter to limit the band to $f_S/2$.

from the CT ADC when the input is active, and thus power dissipation within the CT DSP will depend directly on input activity. Thus, for signals with low activity and long periods of silence, the power consumption of a CT DSP system becomes much lower than during full activity. Also, CT DSPs have a very fast response to sudden input variations as they do not need to wait for a clock signal; this turns out to be beneficial in DC control loops, such as those in DC-DC converters [3]. See [1] and [2] for more details on the CT DSP concept.

The remainder of this paper is organized as follows. Section II describes the entire CT DSP system at a glance. The circuits used in the CT ADC, CT DSP, and CT DAC are each presented in Sections III–V, respectively. Section VII presents measured results characterizing the entire CT ADC/DSP/DAC system targeting voiceband applications.

II. Overview of the CT DSP System

The CT DSP system presented in this paper has a similar general signal flow to that of a previous system by Li *et al.* in [4]. However, the design presented in [4] did not function properly as a single system due to block-to-block interference, and their results were presented by examining individual system components one at a time. In addition, the chip area and power dissipation were very large. The CT system presented in this paper represents a major improvement at the circuit level, resulting in a fully operational, high-performance CT DSP system.

An overview of the CT DSP system is shown in Fig. 3. The CT ADC is an asynchronous delta modulator [5], [6] that has a resolution of $N = 8$ bits. The output of the CT ADC is an indication of whether the 8-bit representation of the input has moved up or down by one least significant bit (LSB). Such CT ADC output signals are easier to process through the delay taps (of the CT DSP) compared to delaying a full 8-bit representation of the input. The CT DSP is a 16-tap direct-form finite impulse response (FIR) filter with programmable 8-bit tap coefficients $h_0 \ \ldots \ h_{15}$ and continuous time delay taps of delay time $T_{\rm D}$ (these are not flip-flops; rather, they are pure continuous-time delay elements which do not involve clocking). The CT ADC outputs, indications of change, are accumulated in the UP/DN counters, whose outputs are then weighted by the tap coefficients to form the full 16-bit words representing the weighted and delayed versions of the digitized input, $\hat{x}(t)$. Other ways to do DSP involving delta modulation have been discussed elsewhere [7], [8]. The final adder, shown as a summation block in Fig. 3, outputs a 16-bit word whose value, $y(t)$, is given by

$$y(t) = \sum_{k=0}^{15} h_k \cdot \hat{x}(t - k \cdot T_{\rm D}). \tag{1}$$

The complete block diagram of the CT DSP system is shown in Fig. 4. The CT ADC has two inputs INH and INL, formed from the applied input $x(t)$ as will be made clear in Section III. The CT ADC generates delta mod signals that are passed to CT delay taps, just as in Fig. 3. However, in Fig. 3 the operations of accumulating these delta mod signals and then weighting them by the tap coefficients was shown in two steps, while in the actual hardware these two operations are done in one step, as will be described in Section IV-B. These outputs are then summed and sent to a CT DAC, which has a resolution of only 8 bits. Thus the final output of the CT DSP system is $\hat{y}(t)$, an 8-bit representation of $y(t)$. To prevent glitches and ensure reliable transmission of data, handshaking techniques as in standard asynchronous digital design are used throughout [9], [10]. Note the difference between this CT DSP system and an asynchronous DSP, as in [11]—in the former the time difference between events is maintained and is critical to the signal representation, while in the latter only their ordering is important. As there is no clock, though, the CT DSP system must be designed similarly to an asynchronous system, and conventional synchronous design techniques will not apply.

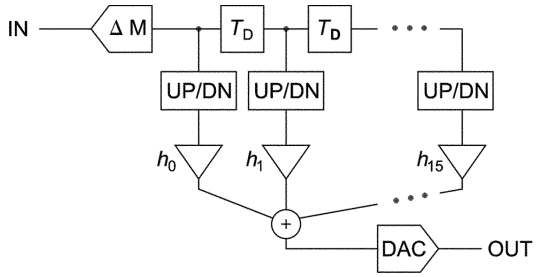


Fig. 3. Simplified block diagram of the implemented CT ADC/DSP/DAC system.

III. CT ADC

A block diagram of the CT ADC, modified from [12], is shown in Fig. 5(a), while example waveforms are shown in Fig. 5(b). The two inputs ACKA and ACKB will be discussed later. The input $x(t)$ is split off-chip into two signals, to which DC voltages are added, resulting in the signals INH and INL which feed two comparators. The purpose of the split is to allow calibration to cancel the offset of each comparator individually; this is discussed later in this section (for this chip, calibration was done externally, although it could be integrated on chip). The top comparator compares INH to a signal $V_{\rm HIGH}(t)$, while the bottom comparator compares INL to $V_{\rm LOW}(t)$. $V_{\rm LOW}(t)$ and $V_{\rm HIGH}(t)$ are consecutive levels taken from a resistor string DAC.

The summary of operations is as follows: when $x(t)$ is within the range $(V_{\rm LOW}(t),\ V_{\rm HIGH}(t))$, then both comparators have outputs of logic 0, and nothing happens. Such a state is seen at the extreme left in the waveforms of Fig. 5(b). When $x(t)$ moves outside this range [in Fig. 5(b)], $x(t)$ begins by increasing beyond $V_{\rm HIGH}(t)$), one of the comparators will detect the movement of $x(t)$ and create a logic 1 on either INC (increment) or DEC (decrement), depending on the direction of movement. When either INC or DEC becomes a logic 1, the control logic will create an indication of the change, akin to that in delta modulation, here termed a *token*. The token consists of two binary signals: CHANGE and UPDN. As seen in Fig. 5(a), the token is fed back through a DAC so that both $V_{\rm LOW}(t)$ and $V_{\rm HIGH}(t)$ are updated after a token is generated. After such an update, $x(t)$ is once again within the range $(V_{\rm LOW}(t),\ V_{\rm HIGH}(t))$. This behavior is demonstrated in the example shown in Fig. 5(b). The CHANGE signal is seen to have a negative edge every time $x(t)$ crosses *either* $V_{\rm LOW}(t)$ *or* $V_{\rm HIGH}(t)$, while UPDN

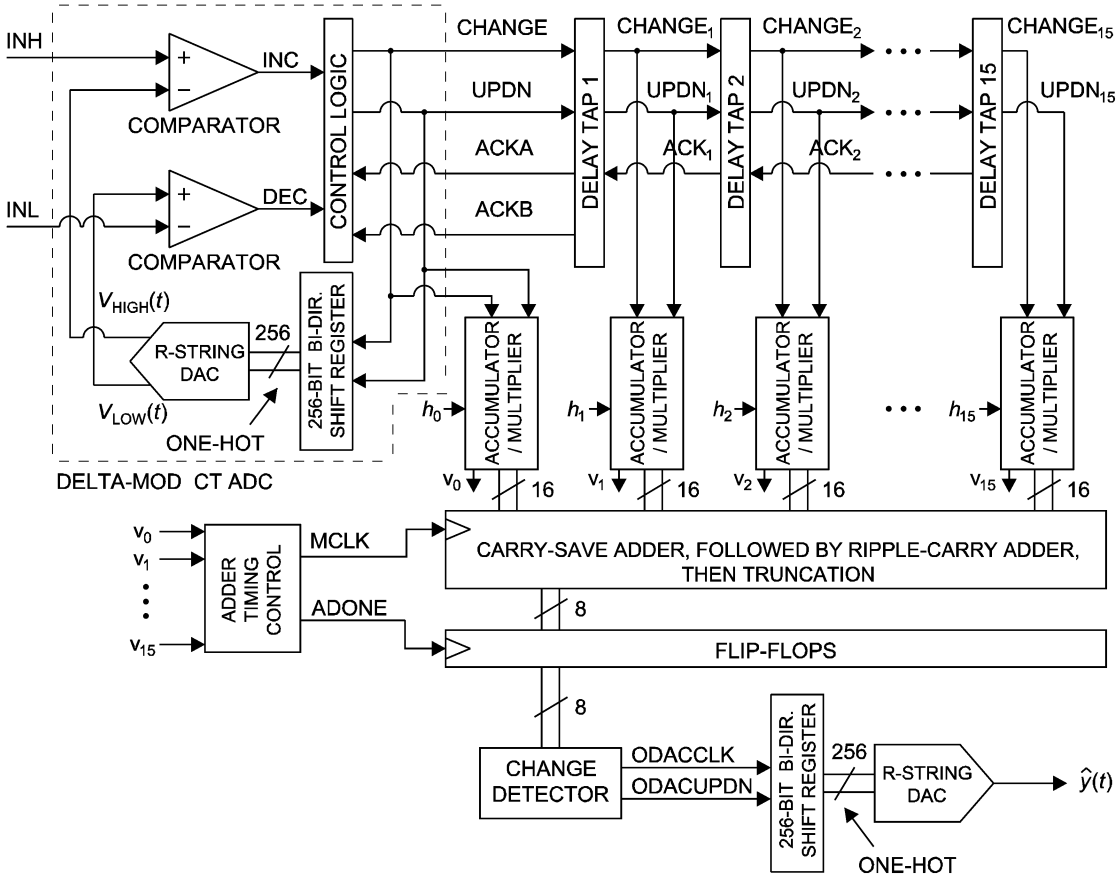


Fig. 4. Detailed block diagram of the entire CT ADC/DSP/DSP system.

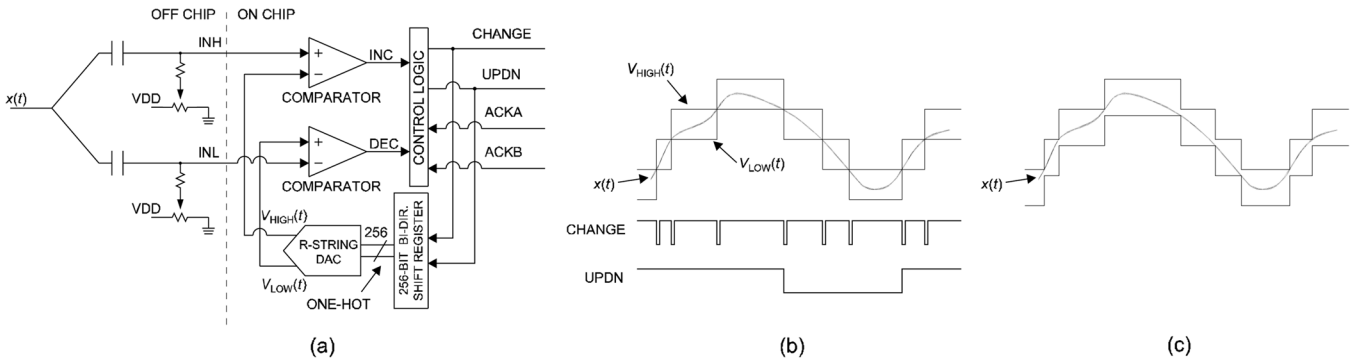


Fig. 5. (a) Block diagram of CT ADC. (b) Example waveforms showing how the CT ADC operates. (c) With added hysteresis, exaggerated for clarity.

is a logic 1 when $V_{\rm HIGH}(t)$ has been crossed, and a logic 0 when $V_{\rm LOW}(t)$ has been crossed. Such an ADC scheme has been termed "level-crossing sampling" or "time code modulation" [12]–[15].

As the CT ADC output tokens always indicate a change (either up or down) by one LSB, the 8-bit feedback resistor string DAC in Fig. 5(a) is not controlled by an 8-bit input word but by a simple bi-directional shift register, as seen in Fig. 6. This simplifies the design of the feedback DAC and prevents glitches in the $V_{\rm LOW}(t)$ and $V_{\rm HIGH}(t)$ signals, as it is not possible for either signal to change by more than one step of the resistor string at a time.

The speed required by the entire CT ADC is determined by how quickly $x(t)$ can cross consecutive boundaries $V_{\rm LOW}(t)$ and $V_{\rm HIGH}(t)$. If the full-scale amplitude covering half of the the non-overload region of the CT ADC is $A_{\rm MAX}$, then the spacing of quantization level boundaries for an N-bit CT ADC will be

$$V_\Delta = \frac{2 \cdot A_{\rm MAX}}{2^N} = \frac{A_{\rm MAX}}{2^{N-1}}. \tag{2}$$

Suppose $x(t)$ is rising; once it crosses $V_{\rm HIGH}(t)$, $V_{\rm HIGH}(t)$ will be updated to the value of the next point of the resistor string DAC (higher by V_Δ). Thus the new value of $V_{\rm HIGH}(t)$ must settle before $x(t)$ can reach it. If $x(t)$ is a sinusoidal input of frequency $f_{\rm INPUT}$ with an amplitude of $A = A_{\rm MAX}$, the minimum time it will take for $x(t)$ to cross V_Δ is

$$T_{\rm GRAN} \equiv \frac{V_\Delta}{\max.\,\text{rate of change}} = \frac{V_\Delta}{\frac{d(x(t))}{dt}\big|_{\max}} = \frac{V_\Delta}{2\pi A_{\rm MAX} f_{\rm INPUT}} = \frac{1}{2^N \pi f_{\rm INPUT}}. \tag{3}$$

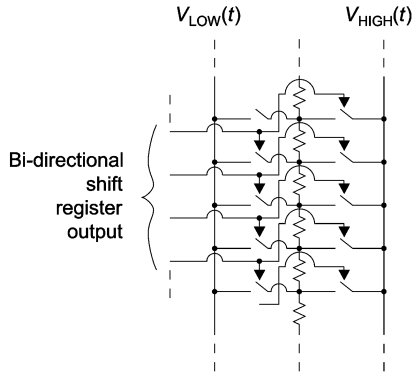


Fig. 6. Schematic of resistor string DAC seen in Fig. 5(a).

To process voiceband signals (200–3400 Hz), which is the goal of this work, we assume that the CT system must be able to process signals for frequencies up to 20 kHz (including spectral components that must be rejected[2]). This leads to $T_{\mathrm{GRAN}} \approx 60$ ns. The CT ADC is designed so that it updates in a time shorter than T_{GRAN}, so that the actual value of the update time does not affect the performance. Thus, the CT ADC can track an input with a maximum slope of $V_{\Delta}/T_{\mathrm{GRAN}} \simeq 32$ V/ms for this design.[3]

A. Control Logic in CT ADC

The control circuitry within the CT ADC, shown in Fig. 7, will be described next. The circuitry takes the comparator outputs INC and DEC [Fig. 5(a)] and forms a token made up of CHANGE and UPDN as previously described. The token, the CT ADC output, is passed with handshaking to the first delay tap, governed by the ACKA and ACKB signals that come from the first delay tap. The generation of ACKA and ACKB will be described in detail in Section IV-A, but for now it should be kept in mind that ACKB will always arrive after ACKA.

INC and DEC enter on the left in Fig. 7. Consider the situation where DEC is a logic 0 and INC changes from a logic 0 to a logic 1 when INH crosses $V_{\mathrm{HIGH}}(t)$ in Fig. 5. INC passes through a switch and sets a NOR-based set-reset (SR) latch (SR_2, labeled no. 2), which eventually causes CHANGE to flip from a logic 1 to a logic 0 (as SR_3 gets set). SR_4 will also get set, disconnecting INC and DEC from the circuitry (by opening the input switches). Also, the output of SR_4 triggers the D flip-flop at the bottom of the figure, which latches in the output of SR_2 to be the UPDN signal. The circuit remains in this state until ACKA resets SR_3, then ACKB resets SR_4 (this order of operations is guaranteed, as will be made clear in Section IV-A). Only when SR_4 is reset are INC and DEC reconnected to the rest the circuitry. In this manner, the CT ADC cannot generate a new token until ACKB resets SR_4. The operation when INL crosses $V_{\mathrm{LOW}}(t)$ in Fig. 5 is similar, with DEC playing the role of INC and SR_1 playing the role of SR_2.

[2]For example, a digital LPF with a passband of 3.4 kHz with an effective $f_S/2$ of 4 kHz would repeat its passband starting at 4.6 kHz, which is undesirable.

[3]If the input moves faster than this, distortion will arise due to slope overload similar to the case of conventional delta modulation. It is the slope of the input, not the input spectrum, that determines proper tracking of the CT ADC.

B. Comparators in CT ADC

The schematic of each comparator in Fig. 5(a) is shown in Fig. 8. The comparator, with no clock, is effectively an amplifier with rail-to-rail (logic level) swings at the output. The first stage is a preamplifier (differential pair), the second is a high-gain stage, and the third is a very-wide-common-mode-range differential amplifier (VCDA) [16]. The design achieves a low static power consumption (20 μW per comparator on a 1 V supply) and a high gain (over 80 dB). However, due to the non-clocked usage of the comparators, their offsets cannot be stored and cancelled while the signal is being processed. In this experimental chip, these offsets are cancelled by setting the DC levels of INH and INL through the use of the potentiometers shown in Fig. 5(a). In fact, following offset cancellation, these levels are on purpose made to differ by a slightly larger amount than V_{Δ}, as shown in Fig. 5(c), thus adding hysteresis to the loop and ensuring noise robustness. Metastable states can lead to an overly long comparison time, but not to multiple "decisions" due to this hysteresis. In a complete implementation, the setting of INH and INL DC levels would be done through an on-chip digital calibration cycle.[4]

As there is no clock, the decision time of the comparator will depend on the slope of the input as it crosses the comparator threshold. To keep power low, the comparators are allowed to have a comparison time which significantly depends on the input rate of change (20, 30, and 150 ns, simulated, for input slopes of 2000 V/ms, 32 V/ms, and 2 V/ms, respectively). Since this time is smallest when the token distance is minimum (shorter comparator delay for faster moving input), the resulting effect is small, consistent with intended performance.

IV. CT DSP

A. CT Delay Tap

As part of the operation of the transversal FIR filter, the tokens output from the CT ADC are delayed by CT delay taps of delay time T_{D}. The value of T_{D} is chosen based on the desired frequency response of the system. An example of the input $x(t)$ and the tokens generated by the CT ADC at the input and output of a delay tap is shown in Fig. 9 (only the timing signal CHANGE of the token is plotted for simplicity). The input rises from a low to a high value, then remains constant. Such movement generates a number of unevenly spaced tokens (each generated at the times $x(t)$ crosses a threshold) by the CT ADC. As seen in Fig. 9, the delay tap must be able to accomodate many tokens at any given time. Eight bits of resolution is typical for voiceband processing and thus is the target for this design. If higher resolution were desired, a smaller T_{GRAN} would have to be used, requiring the delay taps to handle a higher density of tokens. The ability of the delay taps to process tokens is a limiting factor in this design.

The structure of the kth delay tap is shown in Fig. 10. The token, consisting of two signals CHANGE_{k-1} and UPDN_{k-1}, enters the delay tap and emerges after a delay time T_{D} as CHANGE_k and UPDN_k. As the delay tap must have within it many tokens at a given time, the delay tap is formed by concatenating many delay cells in series, as shown in the

[4]For such startup calibration, any on-chip oscillator can be used.

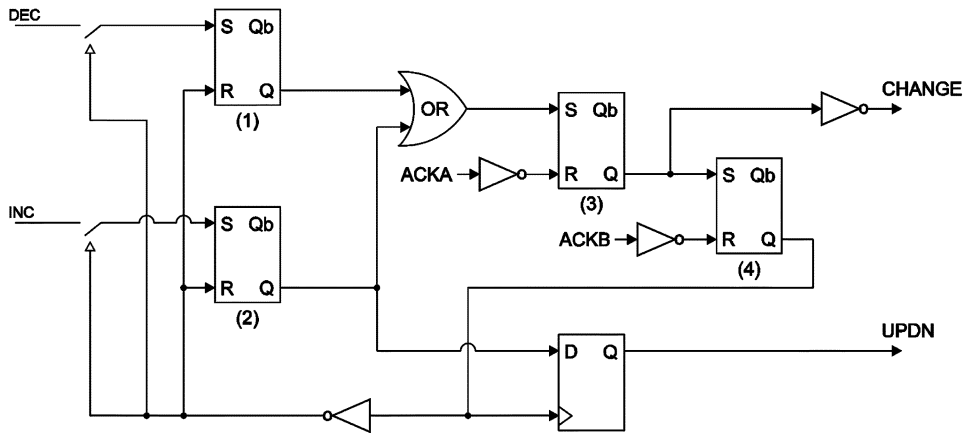


Fig. 7. Schematic of CONTROL LOGIC block shown in Fig. 5(a).

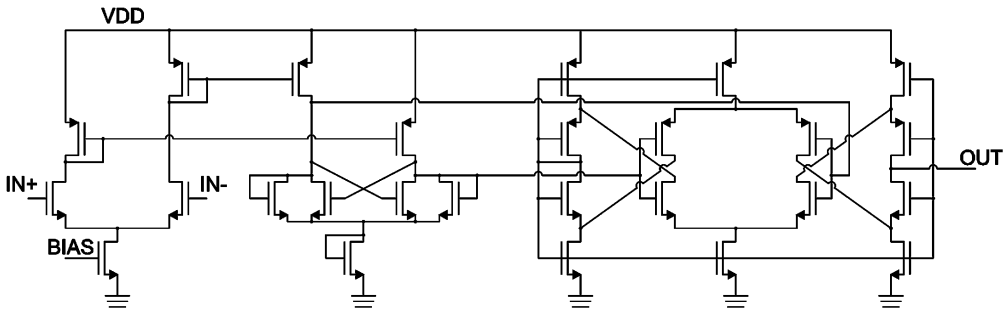


Fig. 8. Schematic of comparator.

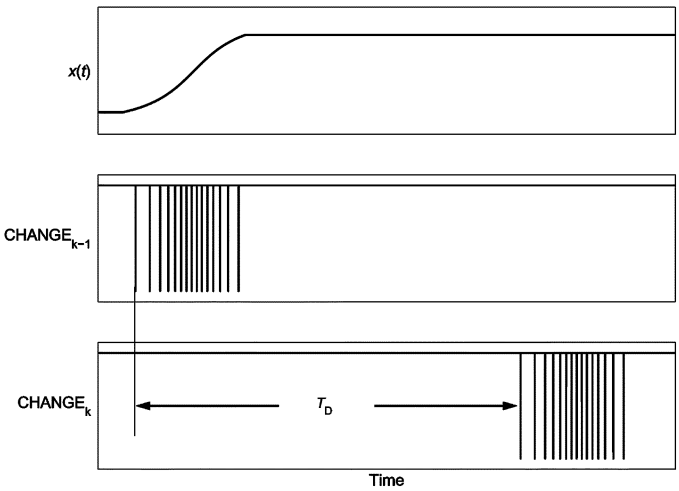


Fig. 9. Example waveforms showing how a rising input (top) can generate many unequally spaced tokens (middle—showing only CHANGE signal at input to the kth delay tap). (bottom) The output of the kth delay tap is a delayed version of its input.

middle of Fig. 10. Tokens are passed between each delay cell via handshaking.

The details of a single delay cell are shown at the bottom of Fig. 10. This delay cell is very similar to that in our previous work [17]—the reader is referred to that work for details of its operation and the tuning of a collection of cells, as only a brief summary is given here. The delay cell receives a token (CHANGE, UPDN) on its IN and D ports, while the delayed token emerges on the OUT and Q ports. The ACK and REQ ports are used for handshaking. When IN has a negative edge, a current source I_{CHARGE} begins charging capacitor C_{CHARGE}. The capacitor is charged in this manner until the positive feedback circuit [18], formed by the remaining drawn transistors in the bottom of Fig. 10, flips state. This causes OUT to change from a logic 1 to 0, completing the delay.

The UPDN signal is carried along with the CHANGE signal by successively latching in the signal in each delay cell. The delay time of the delay cell is tunable by controlling the bias voltage V_{BIAS}. This allows for setting the value of the total delay through digital calibration using a delay-locked loop (not included on this experimental chip) which will also track supply and temperature variations as is done, for example, in [19].

The first delay tap, which receives the tokens generated by the CT ADC, handshakes with the CT ADC via the ACKA and ACKB signals as indicated in Fig. 5(a). ACKA and ACKB are simply the ACK signals from the first two delay cells, respectively. ACKA tells the CT ADC that the first delay tap has received the token, and ACKB tells the CT ADC that that token has advanced to the second delay cell. With the token in the second delay cell, the first delay cell is now free, and thus the CT ADC is free to generate a new token (when required by the input).

B. Digital Core

The remainder of the digital core is next described. As mentioned in Section II, tokens are accumulated and weighted by the tap coefficients in a single step in an *accumulator/multiplier* block as shown in Fig. 4. Instead of accumulating tokens and

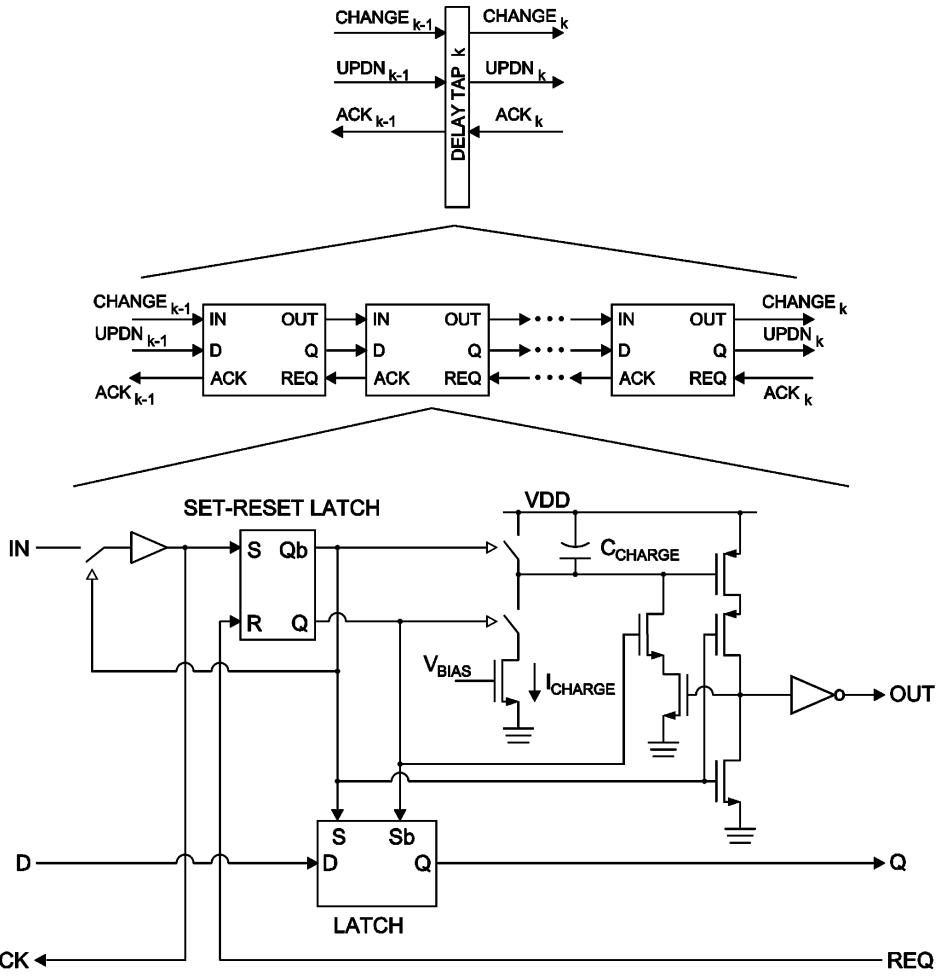


Fig. 10. Architecture of a delay tap.

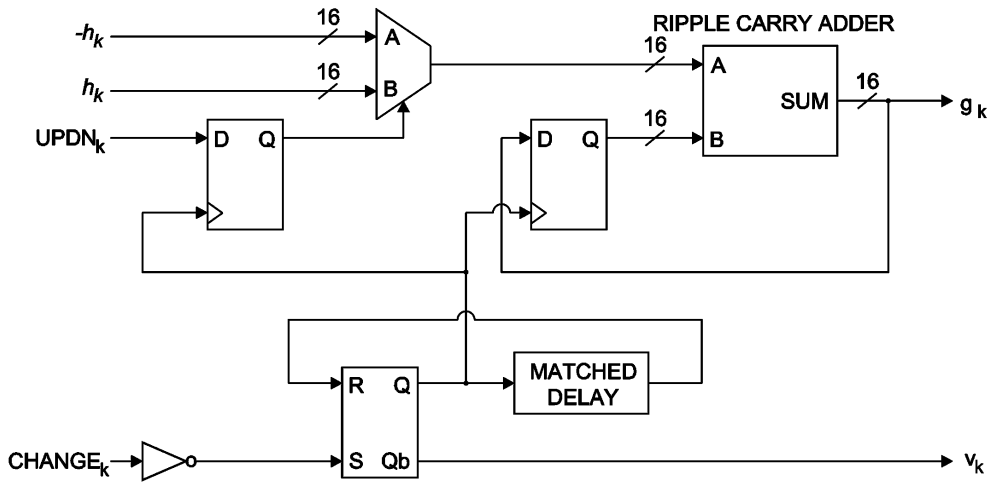


Fig. 11. Block diagram of an accumulator/multiplier block. The multiplexer output is chosen from the A (B) port when the selection signal is a logic 0 (1).

then multiplying the 8-bit word by an 8-bit tap coefficient h_k to form a 16-bit word, the accumulator/multiplier block starts with the 16-bit word that represents its current output value, and either adds $(\mathrm{UPDN} = 1)$ or subtracts $(\mathrm{UPDN} = 0)$ h_k from that value. The circuitry is shown in Fig. 11. The benefit of processing delta mod tokens in this manner is that a 16-bit output signal, g_k, is formed without any digital multipliers; only digital adders are used. As tokens cannot arrive at a given accumulator/multiplier block spaced in time less than $T_{\mathrm{GRAN}} = 60$ ns (see (3)), a simple adder (ripple-carry) is used as even for this adder the addition time is much less than 60 ns.

The adder in the digital core that completes the FIR filter is the combination of a carry-save adder followed by a ripple-carry adder, and the total completion time is measured to be near

4 ns. The inputs to the adder are the 16 output words from each accumulator/multiplier block. Note that two accumulator/multiplier blocks may update at times arbitrarily close to each other. In such a case, two addition requests from different accumulator/multiplier blocks need to be arbitrated. The arbitration routine operates as follows: when an addition request arrives, all the adder input words are latched in and addition begins. If another addition request arrives while adding, the original addition is continued, but as soon as it is complete, all the adder inputs are latched in and a new addition begins immediately.

V. CT DAC

The final adder discussed in Section IV-B outputs a 16-bit output word. However, the output DAC has an 8-bit resolution, and thus the 16-bit adder output word is truncated to 8 bits as mentioned in Section II.

The final 8-bit word is not passed to the output DAC as a parallel word. Instead, the 8-bit word is examined for changes in its LSB to create a token consisting of two signals ODACCLK and ODACUPDN which have similar meanings to the CT ADC output token counterparts CHANGE and UPDN, respectively. A justification of this approach is given in the next paragraph. The hardware of the DAC, with an 8-bit resolution, only handles a delta mod input, meaning it will only have to change its output voltage by one step at a time (either up or down, depending on the value of ODACUPDN). There is already a DAC in this design that operates by changing its output value by one step at a time, the DAC used in feedback in the CT ADC [see Fig. 5(a)]; the output DAC is a copy.

The output DAC as described above creates tokens by examining the LSB of the truncated 8-bit adder output. This scheme will work properly if the final adder's 16-bit word (of value $y(t)$), when truncated to 8-bits (of value $\hat{y}(t)$), changes from addition to addition by no more than one 8-bit LSB at a time. Given that $y(t)$ is a signal that results from filtering the input $x(t)$, and the 8-bit representation of $x(t)$, $\hat{x}(t)$, has been shown to vary through consecutive levels no faster than once every T_{GRAN}, the 8-bit representation of $y(t)$ will also not vary through consecutive levels faster than once every T_{GRAN}.

VI. CT DSP System Power Consumption

When tracing the path of a single token (CHANGE, UPDN) as generated by the CT ADC all the way through the CT DSP system, it is seen that each token generates the same amount of circuit processing as every other token. Let the amount of energy expended on processing a single token be E_{TOKEN} (delay through all 15 delay taps, updating of all 16 accumulator/multiplier blocks, and additions from 16 addition requests). The average power consumption (over a finite sliding time window) of the entire CT system will therefore be a function of the number of tokens generated per second. There will also be static power consumption involved in the resistor string DACs (in the CT ADC and output DAC) and the comparators in the CT ADC, as well as leakage power throughout. Let all the static power be termed $P_{\mathrm{BIAS+LEAK}}$.

Next, let the input $x(t)$ be a sinusoid of amplitude A and frequency f_{INPUT}, and let the CT ADC's maximum non-overload amplitude be A_{MAX} ($A_{\mathrm{MAX}} = 250$ mV in this design). The resolution of the CT ADC is $N = 8$ bits, and if all 2^N levels are used, then there will be $2^N - 1$ threshold crossings as the input moves from the bottom to the top of the non-overload region, leading to $2^{N+1} - 2$ tokens generated per period. The power consumption of the entire CT system can be written as

$$P = P_{\mathrm{BIAS+LEAK}} + \frac{A}{A_{\mathrm{MAX}}}(2^{N+1} - 2) \cdot E_{\mathrm{TOKEN}} \cdot f_{\mathrm{INPUT}}. \tag{4}$$

Thus the power consumption of the CT system is automatically signal driven.

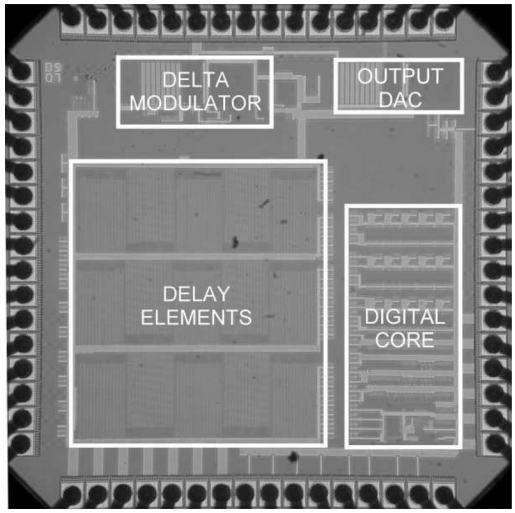


Fig. 12. Chip photograph.

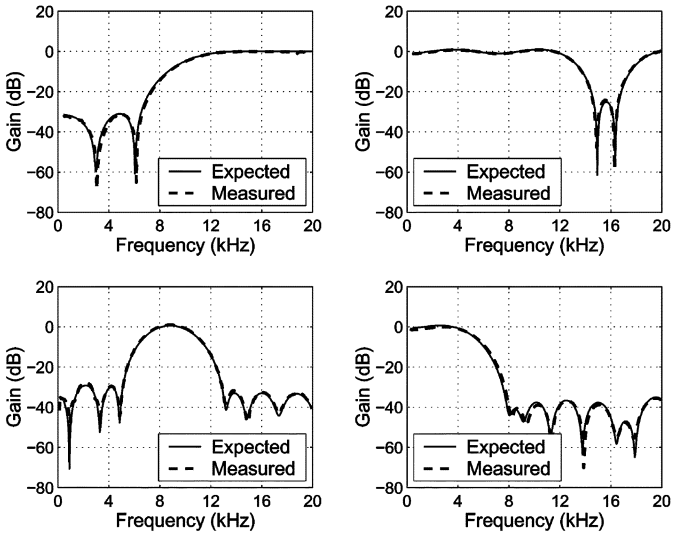


Fig. 13. Four different frequency responses (measured versus expected magnitudes) achieved by programming the tap coefficients h_k. Clockwise from upper left: high-pass, bandstop, low-pass, and bandpass filters.

VII. Measurement Results

This section presents measurement results on the whole CT DSP system, shown in Fig. 12 [20]. A summary of the chip is given in Table I. Fig. 13 shows measured magnitude frequency responses of the CT DSP attained by setting the tap delays to

TABLE I
SUMMARY OF CT ADC/DSP/DAC CHIP

Process	90 nm CMOS
Supply voltage	1 V
Full-scale signal amplitude	0.5 V_{pp}
CT ADC resolution	8 bits
Tap coefficient resolution	8 bits
CT DAC resolution	8 bits
Total area (inc. pads and ESD)	1.3 mm × 1.3 mm
Active area (circuits only)	0.64 mm^2
CT ADC	0.06 mm^2
Delay elements	0.42 mm^2
Digital core	0.13 mm^2
CT DAC	0.03 mm^2

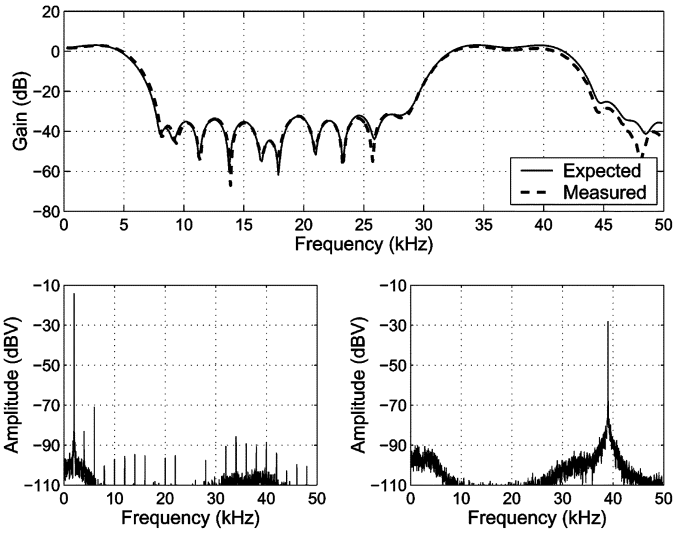


Fig. 14. (Top) Expanded frequency response view of the low-pass filter. (Bottom left) Output spectrum given a full-scale 2 kHz input. (Bottom right) Output spectrum given a −14 dBFS 39 kHz input.

$T_D = 27\ \mu$s (using a test port, with 460 delay cells per tap (on average) with a ±8% spread) and programming the tap coefficients $h_0 \ \ldots \ h_{15}$. Different responses are attained, achieving high-pass, band-stop, bandpass, and low-pass filters. Each frequency response is also compared to an expected curve, which represents the frequency response when accounting for the 8-bit resolution of the tap coefficients as well as the actual measured delay tap delay times.

By configuring the CT DSP as the low-pass filter on the lower right in Fig. 13, some interesting properties of a CT DSP system are investigated next. Fig. 14 expands the view of the low-pass filter frequency response to higher frequencies (top) while examining the output spectrum for different input sinusoids (bottom). As is expected for a transversal FIR filter using tap delays of $T_D = 27\ \mu$s, the passband of the low-pass filter is repeated, centered at a frequency of $1/T_D = 37$ kHz [1]. The bottom of Fig. 14 shows two output spectra given a full-scale 2 kHz input sinusoid (left) and a −14 dBFS 39 kHz input sinusoid (right). In each case, there is no aliasing seen, either of a 2 kHz signal appearing in the second lobe, or from the 39 kHz signal down to the first passband. This is because there is no sampling in the CT DSP system; only continuous time processing is present.

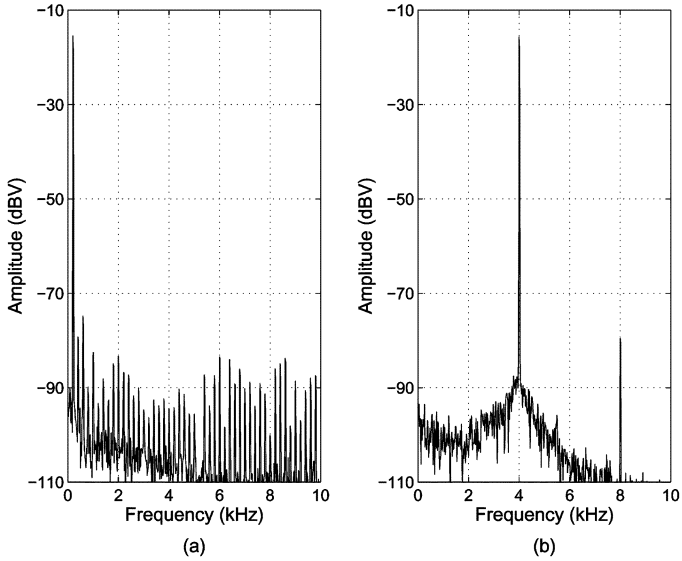


Fig. 15. Output spectra when the CT DSP configured as a low-pass filter, given (a) a full-scale 200 Hz input, and (b) a full-scale 4 kHz input.

The spectral investigation of the output given sinusoidal inputs is continued in Fig. 15 with the CT DSP always a low-pass filter. The output spectrum given a full-scale 200 Hz (4 kHz) sinusoid is shown on the left (right) of Fig. 15. As can be seen, only harmonic distortion is present; no tones at frequencies unrelated to the input frequency are generated. This is in contrast to a sampled system where distortion components at frequencies unrelated to the input frequency arise due to sampling and aliasing. Considering a bandwidth up to 10 kHz as plotted in Fig. 15, the in-band signal-to-noise-and-distortion ratio (SNDR) will depend on how many harmonics fall in-band. On the left (right) of Fig. 15, fifty (two) harmonics fall in-band. Thus, the SNDR increases as the input frequency increases: the measured in-band SNDR is 47 dB given a 200 Hz input signal, but rises to 62 dB for a 4 kHz input signal.

The output of the CT DSP system given a two-tone input is investigated in Fig. 16. The input is a full-scale signal consisting of two equal-amplitude tones, and the CT DSP is again a low-pass filter. The total intermodulation distortion (IMD) remains relatively constant as the input frequencies increase. This is because the IM3 components are among the largest distortion components and they remain in-band as the input tones are swept in frequency. Also, unlike for single sinusoidal inputs, the number of IMD components remains relatively constant as the input frequencies are increased. Thus the total in-band IMD does not decrease as drastically as it did for sinusoidal inputs, measuring −50 dB for both cases. Noting that the SNDR of an ideal 8-bit conventional system for a full-scale input is 50 dB, the lack of improvement by the CT system is partially attributed to imperfect linearity of the blocks, as well as jitter in the delay taps.

Fig. 17 shows plots of the power consumption of each of the main blocks in the CT DSP as well as the total power consumption versus the frequency of a sinusoidal input. The amplitude of the sinusoid is full-scale ($A = A_{MAX}$) for the results shown in Fig. 17(a), and half-scale ($A = (1/2)A_{MAX}$) for the results

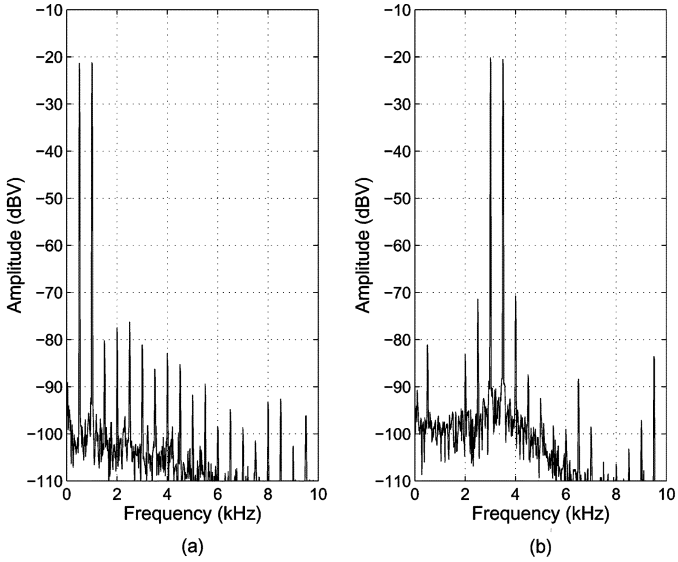


Fig. 16. Output spectra when the CT DSP configured as a low-pass filter, given (a) a full-scale input consisting of equal amplitude 0.5 kHz and 1 kHz tones. (b) The same, given 3 kHz and 3.5 kHz tones.

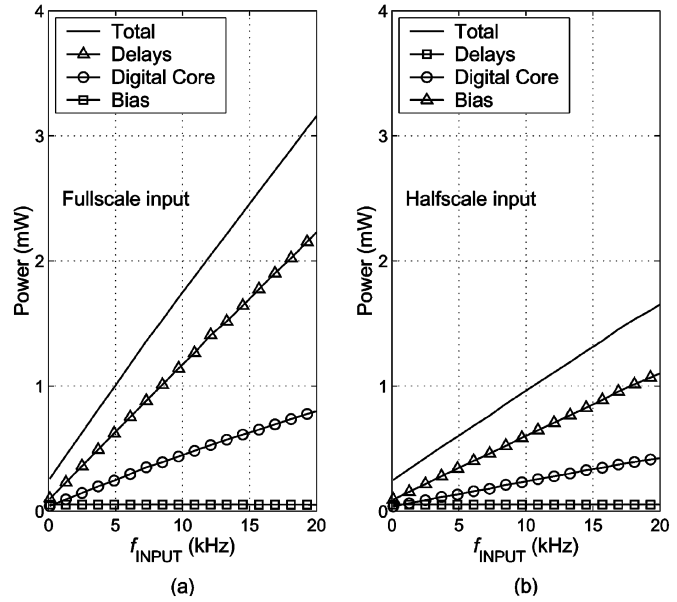


Fig. 17. Plot of power consumption, broken down into the main components. (a) Full-scale sinusoidal input. (b) Half-scale sinusoidal input.

shown in Fig. 17(b). As predicted by (4), the total power consumption varies linearly with the input frequency. Comparing Fig. 17(a)–(b), the reduction in amplitude by half is also seen to reduce the slope of the total power curve by half, also agreeing with (4). From these results, we deduce that $E_{\mathrm{TOKEN}} = 280$ pJ in (4).

In Fig. 18, the top of the figure shows a speech signal over time, and the bottom is the measured instantaneous total power consumption. As can be seen, when the input signal is quiet, the power consumption drops to a quiescent level of 250 μW, which is the value of $P_{\mathrm{BIAS+LEAK}}$ in (4). When the input becomes more active, the power consumption increases automatically due to the processing of more tokens per second. The peak

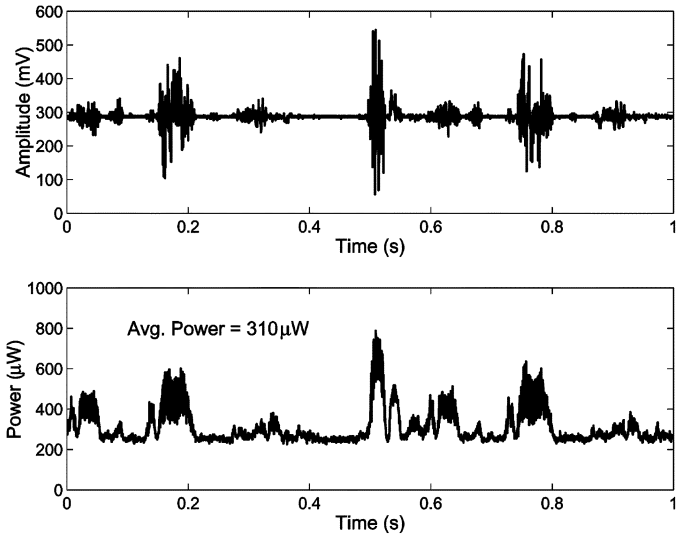


Fig. 18. Plot of input speech signal (top) and the instantaneous power consumption (bottom) of the CT ADC/DSP/DAC.

TABLE II
SUMMARY OF PERFORMANCE OF FIR LPF OVER A 10 KHZ RANGE

Parameter	Value
Total power dissipation for 1 kHz (10 kHz) full-scale sinusoidal input	0.35 (1.71) mW
Bias	0.05 (0.05) mW
Delay elements	0.21 (1.18) mW
Digital core	0.09 (0.48) mW
Total in-band noise power (input -34 dBFS)	25 n(V^2)
SNDR for full-scale sinusoidal input (200 Hz to 4 kHz)	47 to 62 dB
IMD for full-scale dual tone (tones 0.5 kHz apart, swept through passband)	-50 to -51 dB

power consumption is 800 μW for the specific input shown, but due to the quiet times the average power consumption is only 310 μW.

A summary of the performance of the CT DSP system when configured as a low-pass filter (lower-right in Fig. 13) is given in Table II.

VIII. CONCLUSION

We have presented a chip that converts an analog signal to the digital domain, processes the resulting digital signal, and converts the output to analog form, all without sampling or any clock. The result, a CT DSP system, is shown to only generate harmonic distortion for single sinusoid inputs, or intermodulation distortion for multi-tone inputs. As there is no sampling, there is no aliasing, which results in fewer in-band distortion components and a higher in-band SDR compared to conventional (Nyquist-rate sampled) systems. The power consumption is strongly dependent on input activity due to the per-token based energy consumption of the CT DSP system. Handshaking is used throughout the entire path of a token to prevent glitches from occurring. Our results indicate that continuous-time digital signal processing is complementary to sampled signal processing, with certain interesting and attractive properties of its own.

ACKNOWLEDGMENT

The authors would like to thank Prof. K. Shepard for helpful discussions and UMC for chip fabrication.

REFERENCES

[1] Y. Tsividis, "Digital signal processing in continuous time: a possibility for avoiding aliasing and reducing quantization error," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Montreal, Canada, 2004, vol. II, pp. 589–592.

[2] B. Schell and Y. Tsividis, "Continuous-time digital signal processing," in *Proc. 2007 IEEE Int. Symp. Circuits and Systems (ISCAS'07)*, New Orleans, LA, May 2007, pp. 2232–2235.

[3] Z. Zhao, V. Smolyakov, and A. Prodic, "Continuous-time digital signal processing based controller for high-frequency DC-DC converters," in *Proc. IEEE Applied Power Electronics Conf.*, Anaheim, CA, Feb. 2007, pp. 882–886.

[4] Y. Li, K. Shepard, and Y. Tsividis, "A continuous-time programmable digital FIR filter," *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2512–2520, Nov. 2006.

[5] H. Inose, T. Aoki, and K. Watanabe, "Asynchronous delta-modulation system," *Electron. Lett.*, vol. 2, no. 3, pp. 95–96, Mar. 1966.

[6] R. Steele, *Delta Modulation Systems*. New York: Wiley, 1975.

[7] G. Lockhart, "Digital encoding and filtering using delta modulation," *Radio Electron. Eng.*, vol. 42, no. 12, pp. 547–551, Dec. 1972.

[8] N. Kouvaras, "Operations on delta-modulated signals and their applications in the realizations of digital filters," *Radio Electron. Eng.*, vol. 48, no. 9, pp. 431–438, Sep. 1978.

[9] S. Hauck, "Asynchronous design methodologies: an overview," *Proc. IEEE*, vol. 83, no. 1, pp. 69–93, Jan. 1995.

[10] M. B. Josephs, S. M. Nowick, and C. H. van Berkel, "Modeling and design of asynchronous circuits," *Proc. IEEE*, vol. 87, no. 2, pp. 234–242, Feb. 1999.

[11] T. H.-Y. Meng, R. Broderson, and D. G. Messerschmitt, "Asynchronous design for programmable digital signal processors," *IEEE Trans. Signal Process.*, vol. 39, pp. 939–952, Apr. 1991.

[12] E. Allier, G. Sicard, L. Fesquet, and M. Renaudin, "A new class of asynchronous A/D converters based on time quantization," in *Proc. 9th IEEE Int. Symp. Asynchronous Circuits and Systems*, Vancouver, Canada, May 2003, pp. 196–205.

[13] J. Mark and T. Todd, "A nonuniform sampling approach to data compression," *IEEE Trans. Commun.*, vol. COM-29, no. 1, pp. 24–32, Jan. 1981.

[14] J. Foster and T.-K. Wang, "Speech coding using time code modulation," in *Proc. IEEE SoutheastCon*, Williamsburg, VA, 1991, vol. 2, pp. 861–863.

[15] N. Sayiner, H. V. Sorensen, and T. R. Viswanathan, "A level-crossing sampling scheme for A/D conversion," *IEEE Trans. Circuits Syst. II*, vol. 43, no. 4, pp. 335–339, Apr. 1996.

[16] M. Bazes, "Two novel fully complementary self-biased CMOS differential amplifiers," *IEEE J. Solid-State Circuits*, vol. 26, no. 2, pp. 165–168, Feb. 1991.

[17] B. Schell and Y. Tsividis, "A low power tunable delay element suitable for asynchronous delays of burst information," *IEEE J. Solid-State Circuits*, vol. 43, no. 5, pp. 1227–1234, May 2008.

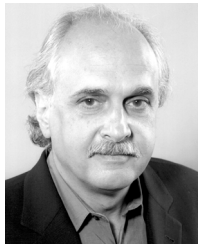
[18] G. Kim, M. K. Kim, B. S. Chang, and W. Kim, "A low-voltage, low-power CMOS delay element," *IEEE J. Solid-State Circuits*, vol. 31, no. 7, pp. 966–971, Jul. 1996.

[19] G. Taylor, S. Moore, S. Wilcox, and P. Robinson, "An on-chip dynamically recalibrated delay line for embedded self-timed systems," in *IEEE Asynch. Symp.*, Eilat, Israel, Apr. 2000, pp. 45–51.

[20] B. Schell and Y. Tsividis, "A clockless ADC/DSP/DAC system with activity-dependent power dissipation and no aliasing," in *IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, 2008, pp. 550–551.

Bob Schell received the B.S.E. degree in electrical engineering from Princeton University, Princeton, NJ, in 2001, and the M.S. and Ph.D. degrees from Columbia University, New York, NY, in 2004 and 2008, respectively.

From 2001 to 2004 he worked at MultiLink Technology, later Vitesse Semiconductor, in the applications division supporting optical drivers and serial links. He has since worked for Analog Devices and Conexant Systems working on mixed-signal blocks for various applications. His current research interests include analog and mixed-signal circuits.



Yannis Tsividis (F'86) received the B.S. degree from the University of Minnesota, Minneapolis, and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 1972, 1973, and 1976, respectively.

He is the Charles Batchelor Memorial Professor of Electrical Engineering at Columbia University, New York, NY. He has worked at Motorola Semiconductor and AT&T Bell Laboratories, and has taught at the University of California at Berkeley, the Massachusetts Institute of Technology, and the National Technical University of Athens, Greece.

Dr. Tsividis is the recipient of the 1984 IEEE W. R. G. Baker Best Paper Award, the 1986 European Solid-State Circuits Conference Best Paper Award, and the 1998 IEEE Circuits and Systems Society Guilllemin-Cauer Best Paper Award. He is a co-recipient of the 1987 IEEE Circuits and Systems Society Darlington Best Paper Award and the 2003 International Solid-State Circuits Conference L. Winner Outstanding Paper Award. He received the IEEE Gustav Robert Kirchhoff Award in 2007, the Presidential Teaching Award from Columbia University in 2003, and the IEEE Undergraduate Teaching Award in 2005.