

EECS 452 – Lecture 7

Today:

Non-uniform quantization: companding
ADC and DAC circuit implementation
principles of DAC - oversampling and equalization
digital filter design

Announcements:

Hw4 posted and due next Thu. (Oct 2)
Proposals due on Friday Sept 26. Email to **hero**
Proposal presentations on Mon Sept 29.
Mon Sept 29, 6-10PM, 1311EECS.

References:

Please see last slide.

Last one out closes the lab door!!!! Please keep the lab clean and organized.

You see things; and you say, "Why?" But I dream things that never were; and I say, "Why not?" George Bernard Shaw

2014 project titles and teams

Team Name	Name
Acoustic Corection and Analyzer	Lacy, Charles Peter Mistaleski, Michael Robert Pomakov, Rossen Ognianov
Fitness Wristband	Meirbekova, Aida Joseph, Jaya Elizabeth Khurana, Arjun Krishan
Guitar and Music performance augmenter	Sukh, Chetan Krishna Goddard, George Kanem Ganguly, Shamik Luthra Rao, Yasodakishore Mahavratayajula
Jarvis System	Agrawal, Pragya Martel, David Thomas Calabrese, Dominic Frank Sawicki, Nathan James
Music Transcription Device	Schmidt, Skyler Lee Hung, Abben Learman, Melissa Anne
Touchless Keyboard	Lu, Cynthia You Jie, Mengmeng Sha, Yu
Weight Lifting feedback	Bretz, Aaron James Den Hollander, Bryan Jay Gansallo, Emmanuel Adetokunbo Wan, Yiyang

Deadlines over the next week

Schedule

- ▶ Fri Sept 26, by 6PM, team's project proposal due.
- ▶ Your team spokesperson should email proposal to **hero**.
- ▶ Mon Sept 29 6-10:30PM, proposal presentations.
- ▶ Your team spokesperson should sign your team up for 30 min slot.
- ▶ Come see staff during office hours to discuss any questions you may have on proposal or on presentation.

Preparation of project proposal

The project proposal must use the template available on the EE452 webpage (Homeworks/Projects)

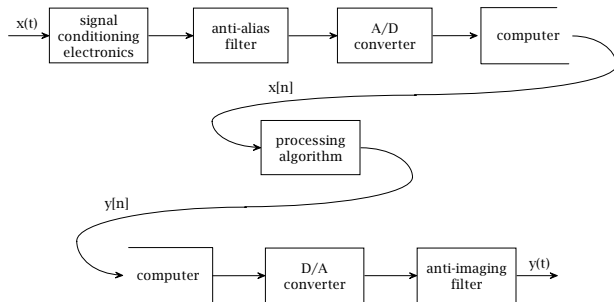
1. Introduction and overview
2. Description of project
 - 2.1 System concept, feasibility of project
 - 2.2 Describe system architecture with detailed block diagram including DSP, FPGA, peripherals as applicable.
 - 2.3 Predict what can go wrong and your contingency plan.
 - 2.4 Provide a preliminary parts list including devices available the lab and those that you wish to purchase (parts numbers, cost and links to webpages). Give a total cost projection.
3. Milestones
 - 3.1 Milestone 1 (Th Nov 6)
 - 3.2 Milestone 2 (Tu Nov 25)
4. Contributions of each member of the team.
5. References and citations

Proposal presentations: Mon Sept 29

Schedule

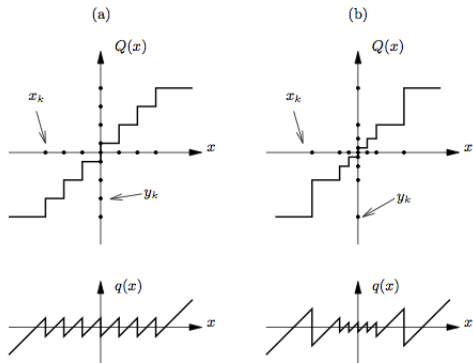
- ▶ Presentations will occur from 6PM to 10:00PM in EECS 1311.
- ▶ Your team spokesperson must sign the team up for a 30 minute slot (20 min presentation).
- ▶ All team members must take part in their team's presentation.
- ▶ You may stay for any or all other portions of the presentation meeting.
- ▶ Team should arrive at least 20 minutes before their time slot.
- ▶ Team must use powerpoint or other projectable media for your presentations.
- ▶ The presentation must cover each section of the proposal.
- ▶ You should put your presentation on a thumb drive and/or email copy to *hero* before the meeting.

ADC and DAC - Recall basic DSP paradigm



Physical signal \rightarrow **Digital signal** \rightarrow **Physical signal**

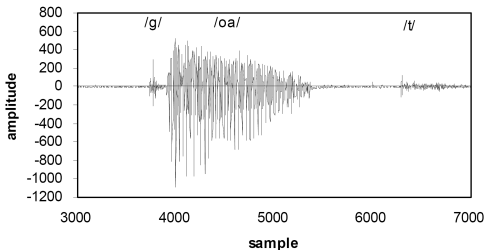
Quantizer functions and their errors



Source: "Memoryless scalar quantization," Phil Schniter, Connexions module
<http://cnx.org/content/m32058/latest/>.

- ▶ Top: uniform quantizer (left) non-uniform quantizer (right).
- ▶ Bottom: quantizer errors as a function of x .

Non-uniform quantization



Source: TMS320C5554 application note SPRA 163A

- ▶ Shown are the speech samples of the word "goat"
 - ▶ speech has large dynamic range - high amplitudes rarer than low
 - ▶ unvoiced and plosive phonemes carry most of the information
 - ▶ The amplitude distribution is concentrated about zero
 - ▶ Human ear has low (logarithmic) sensitivity to high amplitudes
 - ▶ ... therefore we should allocate more bits for low amplitudes!
- ▶ For speech and audio non-uniform quantization gives lower error.

Non-uniform quantization: mu-law companding

COMpression and exPANDING of digitized waveforms (companding).

- ▶ Developed for digital speech transmission in the telephone system.
 - ▶ International Telecommunication Union ITU-T Recommendation G.711.
 - ▶ Uses 8 bits to represent each sample
 - ▶ 8000 samples/sec is sampling rate
 - ▶ Speech transmitted at 64 Kbps
- ▶ Idea: progressive taxation
 - ▶ Instead of using equal step sizes, use smaller steps in certain ranges of input values.
 - ▶ Why? Non-uniform amplitude distribution.
 - ▶ Can accomplish this by applying memoryless non-linearity followed by uniform quantizer.

μ -law transfer function

How to implement?

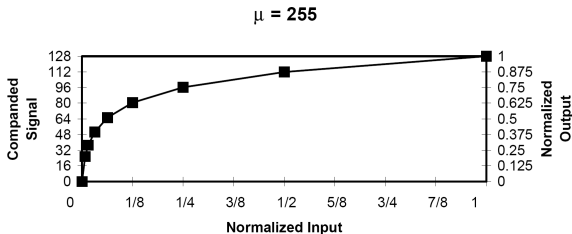
- ▶ Idea one: come up with unequal step sizes for the entire range.
- ▶ Idea two: let's skew the signal instead.
 - ▶ “stretch” the small values and “compress” the large values.
 - ▶ Now apply a 8-bit uniform quantizer.
 - ▶ This has an equivalent effect of assigning small steps when input is small, and large steps when input is large.

$$y = Q \left(\operatorname{sgn}(x) \frac{\ln(1 + \mu|x|/V)}{\ln(1 + \mu)} \right), \quad \mu = 255.$$

where $Q(x)$ is the uniform quantizer and $\mu > 0$ is a design parameter.

- ▶ Mu-255 Law is commonly used in North America and Japan telephone networks. Another (A-Law) compander is used in Europe and elsewhere.
- ▶ Performance roughly equivalent to 14-bit uniform (linear) encoding.

μ -law transfer function



- ▶ 4 bits (3 amplitude bits plus 1 sign bit) of resolution shown for illustration
- ▶ The samples along μ -law curve show how the signal is encoded (compressed)
- ▶ When decoded (expanded) an inverse μ -law could be applied after ADC
- ▶ In actual waveform coding PCM systems, the decoding is done before the ADC.

μ -law coding and decoding tables

Biased Input Values													Compressed Code Word						
													Chord			Step			
bit: 12	11	10	9	8	7	6	5	4	3	2	1	0	bit: 6	5	4	3	2	1	0
0	0	0	0	0	0	0	1	a	b	c	d	x	0	0	0	a	b	c	d
0	0	0	0	0	0	1	a	b	c	d	x	x	0	0	1	a	b	c	d
0	0	0	0	0	1	a	b	c	d	x	x	x	0	1	0	a	b	c	d
0	0	0	0	1	a	b	c	d	x	x	x	x	0	1	1	a	b	c	d
0	0	0	1	a	b	c	d	x	x	x	x	x	1	0	0	a	b	c	d
0	0	1	a	b	c	d	x	x	x	x	x	x	1	0	1	a	b	c	d
0	1	a	b	c	d	x	x	x	x	x	x	x	1	1	0	a	b	c	d
1	a	b	c	d	x	x	x	x	x	x	x	x	1	1	1	a	b	c	d

E

Compressed Code Word									Biased Output Values												
Chord			Step																		
bit: 6	5	4	3	2	1	0	bit: 12	11	10	9	8	7	6	5	4	3	2	1	0		
0	0	0	a	b	c	d	0	0	0	0	0	0	0	1	a	b	c	d	1		
0	0	1	a	b	c	d	0	0	0	0	0	0	1	a	b	c	d	1	0		
0	1	0	a	b	c	d	0	0	0	0	0	1	a	b	c	d	1	0	0		
0	1	1	a	b	c	d	0	0	0	0	1	a	b	c	d	1	0	0	0		
1	0	0	a	b	c	d	0	0	0	1	a	b	c	d	1	0	0	0	0		
1	0	1	a	b	c	d	0	0	1	a	b	c	d	1	0	0	0	0	0		
1	1	0	a	b	c	d	0	1	a	b	c	d	1	0	0	0	0	0	0		
1	1	1	a	b	c	d	1	a	b	c	d	1	0	0	0	0	0	0	0		

D

Some common A/D circuits

There are several ways in which the conversion of an analog waveform into a series of numbers can be implemented.

Flash

Slope integration.

Successive approximation.

Delta-sigma modulation.

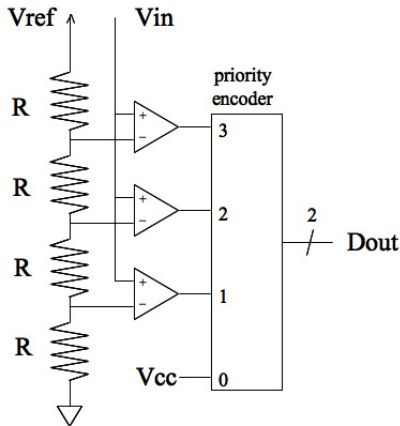
Implementation with Priority encoder

4-to-2 priority encoder
outputs two bit word

Input 3 has highest priority while input 1 has lowest

Input 0 has no effect on output.

I3	I2	I1	I0	O1	O0
0	0	0	x	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1



8-bit Flash A/D

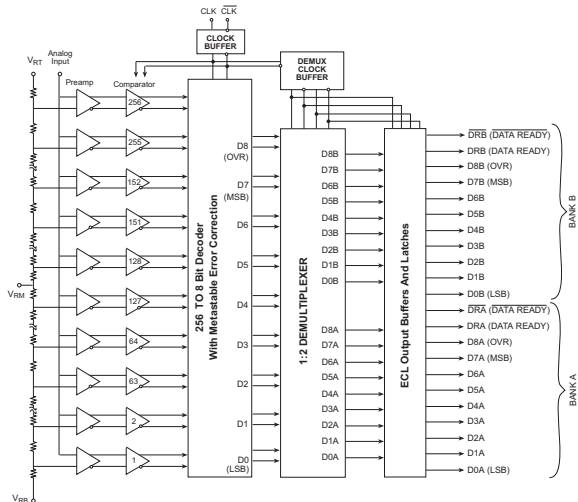


Figure from Fairchild Semiconductor SPT7750 data sheet.

Issues with Flash ADCs

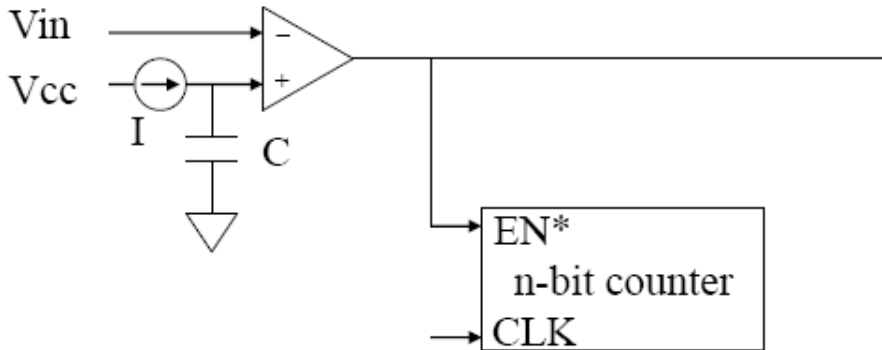
Pros

- ▶ Simple architecture that is highly parallelizable.
- ▶ Very fast since mapping is done on all bits in parallel.
- ▶ Speed is only limited by switching capacitance of opamps and priority encoder bandwidth.

Cons:

- ▶ Offset error and saturation? Shift and scale V_{in} into active region of quantizer.
- ▶ Non-linearity? Improve resistor network reliability.
- ▶ Space requirements grow exponentially in number of bits?
Sequential circuits.

Slope integration ADC



Start: Counter is reset and C is discharged

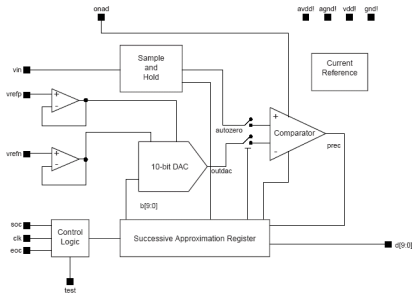
Run: Charge C at fixed current I until $V_{cap} > V_{in}$

Final counter value is D_{out}

Successive approximation A/D

Sampling: uses a track and hold to capture a voltage on a capacitor.

Quantization: uses a local voltage reference bits of a D/A converter; they are successively switched so as to match the D/A output to the captured voltage.

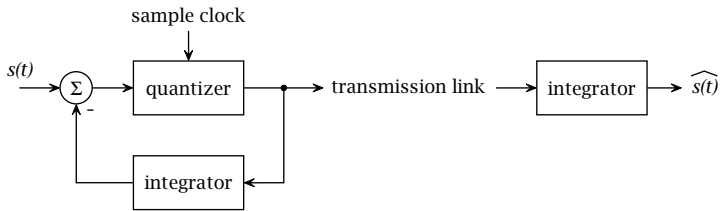


High speed, high MHz is possible. Common accuracies range from 8 to 16 bits.

Successive approximation converters can also be pipelined to develop a few bits at a time as values pass through the pipeline. This provides high speed at the cost of a small of delay.

Figure from Atmel AD023 data sheet.

Delta-sigma A/D



Basic idea is to save bandwidth by sending the changes (often encoded using a single bit) in a waveform rather than the full waveform. Samples very fast. Exploits the cheap availability and small size of today's digital logic. More details to come in a later lecture.

Digital to analog conversion (reconstruction)

Digital signal is sampled in time and quantized in amplitude

- ▶ DAC consists of two stages:
 - ▶ Reversing the quantization:
 - ▶ convert 2s comp digital word to an analog (voltage) value

$$0x77FF \rightarrow 0.9735V$$

- ▶ Reversing the sampling:
 - ▶ Convert discrete time samples to continuous time signal

$$\{x[n] : n = 0, 2, \dots, 1023(\text{samples})\} \rightarrow \{x(t) : 0 < t < 1023T_s(\text{secs})\}$$

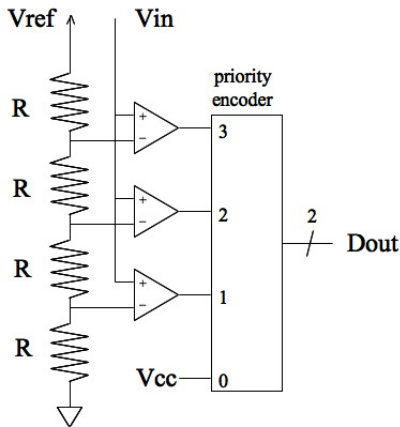
Recall: Quantization w/ VDivider-priority encoder

4-to-2 priority encoder outputs two bit word

Input 3 has highest priority while input 1 has lowest

Input 0 is has no effect in output.

I3	I2	I1	I0	O1	O0
0	0	0	x	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1



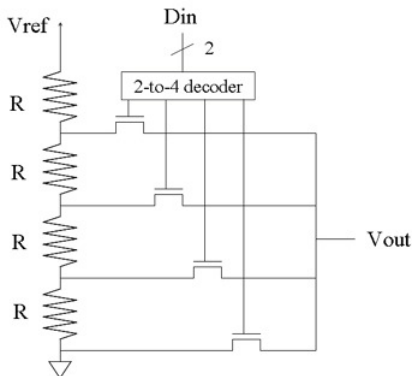
Reversing the quantization process

Reversing the quantization process is easy.

One of many ways: voltage divider.

What's the input-output mapping?

Can change the output voltage levels by modifying R in voltage divider.



Reversing the sampling: interpolation

Assume that bandwidth B signal $x(t)$ has been sampled at $F_s > 2B$, giving samples $x[n]$.

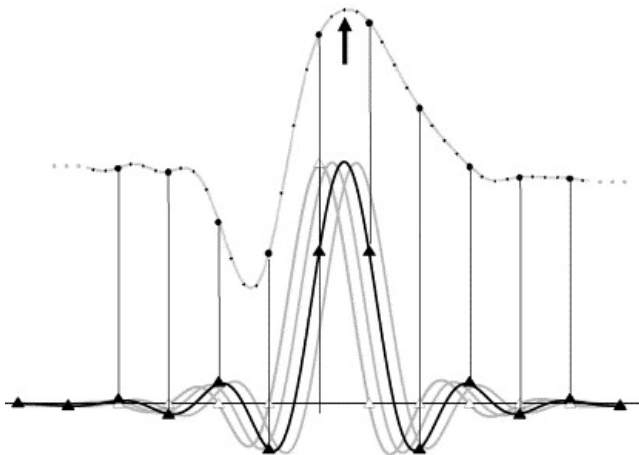
Interpolate the samples $x[n] = x(nT_s)$ using the cardinal series (FT of ideal low-pass filter (LPF) with BW $F_s/2$):

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \cdot \text{sinc}(\pi F_s(t - nT_s))$$

- ▶ This is the cardinal series expansion of $x(t)$ ^a
- ▶ This perfectly recovers the input signal $x(t)$, due to Nyquist sampling theorem.

^aHere $\text{sinc}(x) = \sin(x)/x$. Sometimes you will see the sinc function defined differently: $\text{sinc}_{\pi}(x) = \sin(\pi x)/(\pi x)$

Cardinal series: sinc-interpolation



<http://dx.doi.org/10.1016/j.jneumeth.2005.12.031>

Reversing the sampling process: CardSeries

Derivation of cardinal series reconstruction (neglect quantization)

Represent

$$x(t) = \mathcal{F}^{-1}\{\mathbb{X}(F)\} = \mathcal{F}^{-1}\left\{H(F) \sum_n \mathbb{X}(F + nF_s)\right\}$$

where $H(F) = 1$, $|F| \leq F_s/2$, is ideal lowpass filter of bandwidth $F_s/2$.

Apply fundamental relation: $\sum_n \mathbb{X}(F + nF_s) = T_s \sum_n x[n]e^{-j2\pi FnT_s}$,

$$\begin{aligned}x(t) &= \mathcal{F}^{-1}\left\{H(F)T_s \sum_n x[n]e^{-j2\pi FnT_s}\right\} \\&= \sum_n x[n]\mathcal{F}^{-1}\left\{H(F)T_s e^{-j2\pi FnT_s}\right\} \\&= \sum_n x[n]\text{sinc}(\pi F_s(t - nT_s))\end{aligned}$$

Implementation issues with cardinal series

Perfect reconstruction by sinc interpolation

$$x(t) = \sum_n x[n] \text{sinc}(\pi F_s(t - nT_s))$$

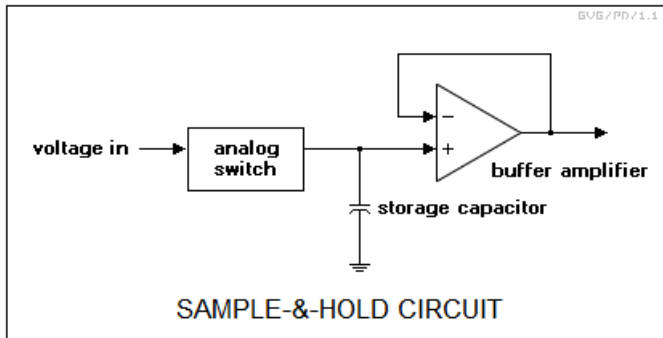
Where $F_s = 1/T_s$ is at least the Nyquist sampling rate $2B$.

The representation above is exact (no error)

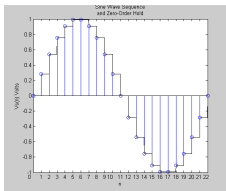
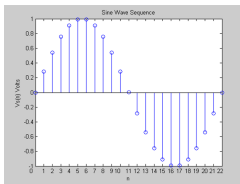
The problem is that this solution suffers from high complexity.

- ▶ Requires infinite number of samples.
- ▶ Non causal. Need future and past samples to compute $x(t)$.
- ▶ Not implementable in real-time streaming data applications.
- ▶ Sinc function non-linearity is difficult to synthesize.

Reversing the sampling process: SaHold



Reversing the sampling process: SaHold



What can we do to minimize the effect of this step-like distortion?
Mathematically represent the sample and hold output signal as

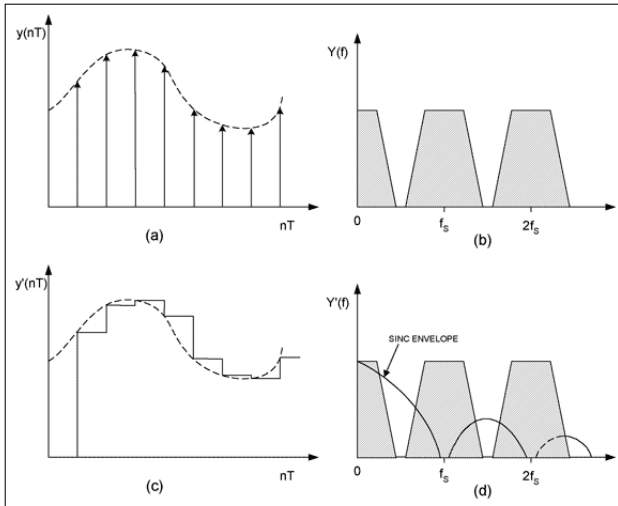
$$y(t) = \sum_n x[n]p(t - nT_s)$$

where $T_s = 1/F_s$ is the sampling period and $p(t)$ is a rectangular pulse

$$p(t) = \begin{cases} 1, & t \in [0, T_s] \\ 0, & o.w. \end{cases}$$

What is spectrum of the DAC signal $y(t)$?

Reversing the sampling process: SaHold



Source: MAXIM, Application Note 3853, 2006

Reversing the sampling process: derivation

Spectrum $\mathbb{Y}(F)$ of $y(t)$ is easy enough to derive:

$$\begin{aligned}\mathbb{Y}(F) &= \mathcal{F}\{y(t)\} = \int_{-\infty}^{\infty} y(t)e^{-j2\pi Ft} dt \\ &= \sum_n x[n]\mathcal{F}\{p(t - nT_s)\} \\ &= \mathbb{P}(F) \sum_n x[n]e^{-j2\pi FnT_s}\end{aligned}$$

with $\mathbb{P}(F)$ the fourier transform of $p(t)$.

Ok, but how is this related to the spectrum of $x(t)$?

Again, recall relation: if $x[n] = x(nT_s)$ where $x(t)$ is a cts time signal with spectrum $\mathbb{X}(F)$ then

$$\sum_n x[n]e^{-j2\pi FnT_s} = \frac{1}{T_s} \sum_n \mathbb{X}(F + nF_s) \quad (1)$$

Reversing the sampling process: derivation

Use fundamental relation to obtain

$$\mathbb{Y}(f) = \mathbb{P}(F) \frac{1}{T_s} \sum_n \mathbb{X}(F + nF_s)$$

It remains to find $\mathbb{P}(F)$. Simple exercise:

$$\mathbb{P}(F) = e^{-j2\pi FT_s/2} T_s \text{sinc}(\pi FT_s)$$

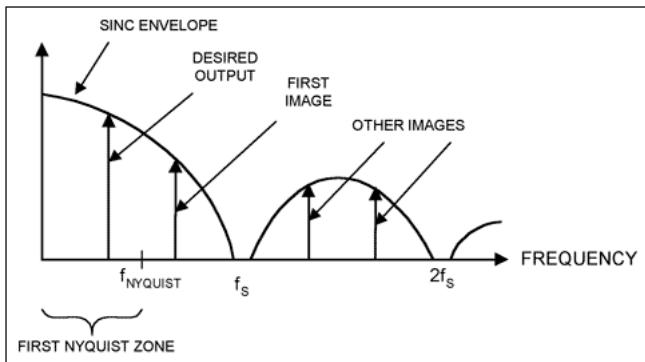
So, plugging back into expression for $\mathbb{Y}(F)$

$$\mathbb{Y}(F) = \text{sinc}(\pi FT_s) \sum_n \mathbb{X}(F + nF_s)$$

The sinc function is a shading function with zeros at $F = k/T_s$ for integer $k \neq 0$.

The summation generates "images" of the spectrum $\mathbb{X}(F)$ centered at the zeros of the shading function.

Reversing the sampling process: shading



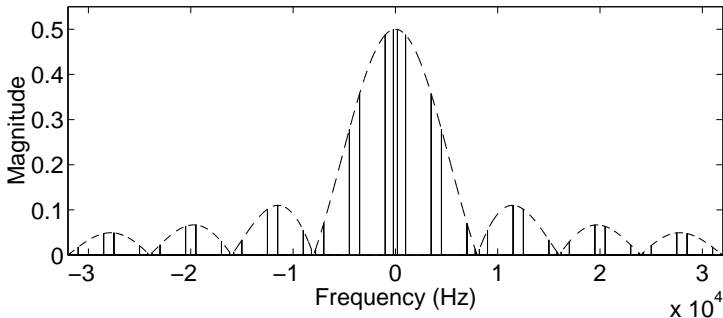
- ▶ Illustration of shading and image frequencies for sample-and-hold of a single tone (note f_s denotes sampling freq in Hz).

$$x(t) = A \sin(2\pi F_0 t + \theta)$$

- ▶ Nyquist criterion is satisfied here: $F_0 < F_s/2$.

Reversing the sampling process: real example

Spectrum of the reconstructed waveform



- ▶ Real example:
 - ▶ Three tones: 200 Hz, 1000 Hz, and 3500 Hz, $F_s = 8000$ Hz.
 - ▶ Using a zero-order hold D/A converter. The dashed line shows the effects of the zero-hold on the spectrum magnitude.
 - ▶ It possesses images of all three tones.

Reversing the sampling process

Require a reconstruction filter to filter out images at higher frequencies

Zero-order hold has weighted (shaded) the spectrum.

Low pass (anti-image) filter needed to attenuate images.

Can also correct for the zero-order hold amplitude shading.

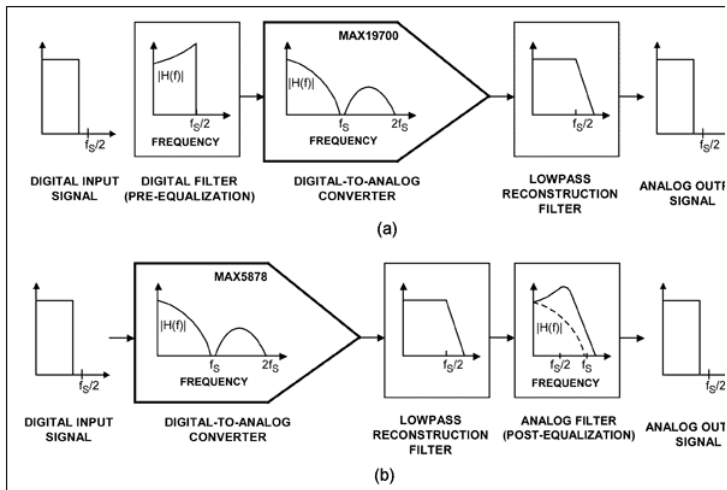
Low pass cutoff will nominally be somewhat below $F_s/2$.

May have concerns about phase distortion and delay.

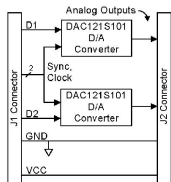
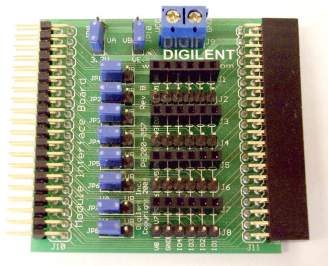
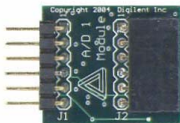
Approaches to compensate for shading:

- ▶ Digital oversampling. Really is just interpolation in digital domain: insert an interpolated sample between each pair of data samples.
- ▶ Apply equalization filter to "invert" the shading frequency function.

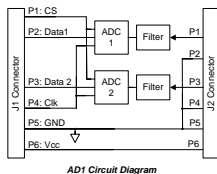
Reversing the sampling process: Equalization



Digilent D/A, A/D and MIB

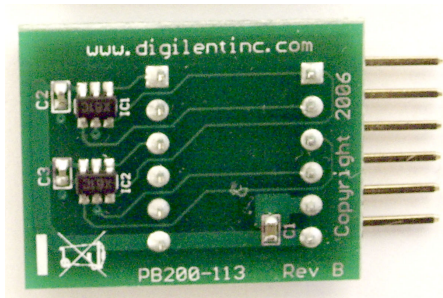
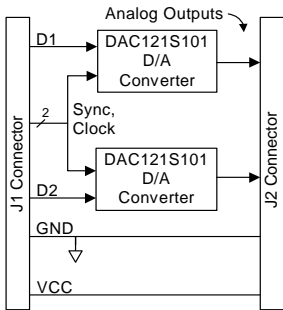


From Digilent data sheets.



We have modified our MIB boards to have sockets in positions J1, J3, J5, J7 and pins in the other positions. The UCF naming is J1→pmod_a, J3→pmod_b, J5→pmod_c and J7→pmod_d.

The Digilent PMod-DA2 module

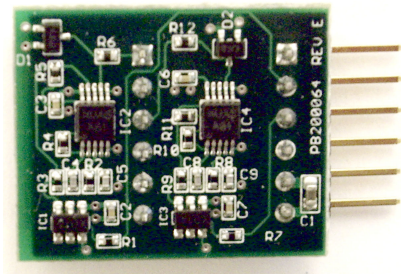
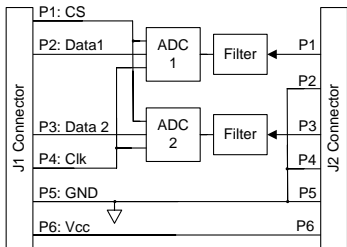


The PMod-DA2 uses two National Semiconductor DAC121S101 12-bit digital-to-analog converters.

Uses a bit-serial interface. Maximum serial clock rate is 30 MHz.
Operates using supply voltages in the range 2.7V to 5.5V.

Figure from the PMod Digilent data sheet.

The Digilent PMod-AD1 module



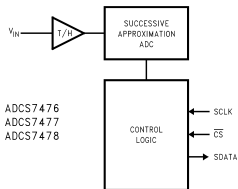
That was D/A. Now let's take a look at the A/D PMod.

The PMod-AD1 uses two National Semiconductor ADCS7476 12-bit analog-to-digital converters.

Uses a bit-serial interface. Maximum serial clock rate is 20 MHz.
Operates using supply voltages in the range 2.7V to 5.25V.

Figure from the PMod Digilent data sheet.

The ADCS7476 A/D



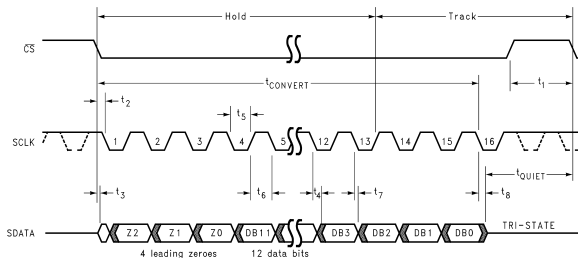
Max serial clock : 20 MHz

Max sample rate: 1 MHz

Data uses offset binary.

Input switches from track to hold on falling edge of the sync signal.

From the National Semiconductor data sheet.



Summary of what we covered today

- ▶ ADC/DAC theory and practice
- ▶ Reconstruction, image frequency filtering, equalization,

References

"Equalizing Techniques Flatten DAC Frequency Response," MAXIM, Application Note 3853, 2006.

<http://www.maxim-ic.com/app-notes/index.mvp/id/3853>

Freescale Semiconductor AN2438/D application notes,

TI's *Understanding Data Converters*, SLAA013, July 1995.

"Digital signal processing," Proakis and Manolakis, 3rd Edition.

"Understanding digital signal processing," R. Lyons, 2004.