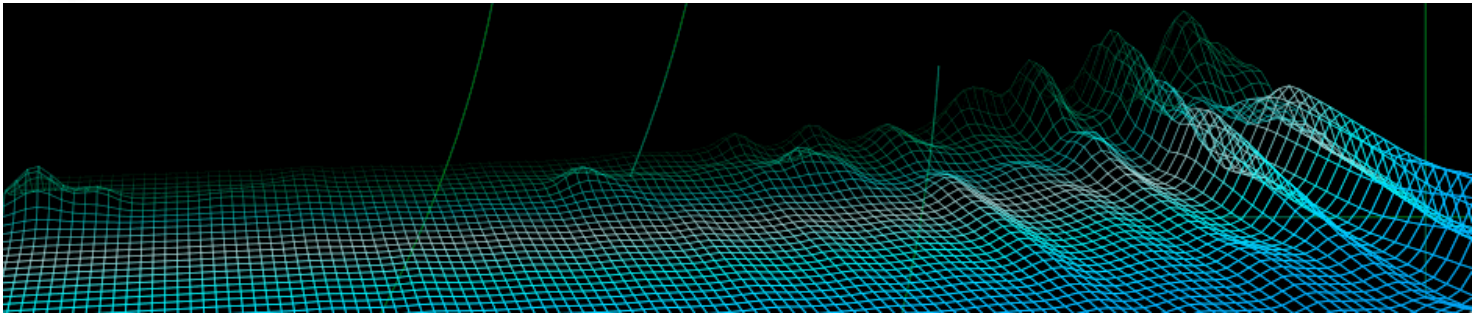




Practical digital audio signal processing

[Skip to content](#)[Home](#)[About](#)[← Oversampling](#)[What is aliasing? →](#)

What is dither?

Posted on [October 20, 1996](#) by [Nigel Redmon](#)

To dither means to add noise to our audio signal. Yes, we add noise on purpose, and it is a good thing.

How can adding noise be a good thing??!!

We add noise to make a trade. We trade a little low-level hiss for a big reduction in distortion. It's a good trade, and one that our ears like.

The problem

The problem results from something Nyquist didn't mention about a real-world implementation—the shortcoming of using a fixed number of bits (16, for instance) to accurately represent our sample points. The technical term for this is “finite wordlength effects”.

At first blush, 16 bits sounds pretty good—96 dB dynamic range, we're told. And it is pretty good—if you use all of it all of the time. We can't. We don't listen to full-amplitude (“full code”) sine waves, for instance. If you adjust the recording to allow for peaks that hit the full sixteen bits, that means much of the music is recorded at a much lower volume—using fewer bits.

In fact, if you think about the quietest sine wave you can play back this way, you'll realize it's one bit in amplitude—and therefore plays back as a square wave. Yikes! Talk about distortion. It's easy to see that the lower the signal levels, the higher the relative distortion. Equally disturbing, components smaller than the level of one bit simply won't be recorded at all.

This is where dither comes in. If we add a little noise to the recording process... well, first, an analogy...

An analogy

Try this experiment yourself, right now. Spread your fingers and hold them up a few inches in front of one eye, and close the other. Try to read this text. Your fingers will certainly block portions of the text (the smaller the text, the more you'll be missing), making reading difficult.

Wag your hand back and forth (to and fro!) quickly. You'll be able to read all of the text easily. You'll see the blur of your hand in front of the text, but definitely an improvement over what we had before.

The blur is analogous to the noise we add in dithering. We trade off a little added noise for a much better picture of what's underneath.

Back to audio

For audio, dithering is done by adding noise of a level less than the least-significant bit before rounding to 16 bits. The added noise has the effect of spreading the many short-term errors across the audio spectrum as broadband noise. We can make small improvements to this dithering algorithm (such as shaping the noise to areas where it's less objectionable), but the process remains simply one of adding the minimal amount of noise necessary to do the job.

An added bonus

Besides reducing the distortion of the low-level components, dither let's us hear components below the level of our least-significant bit! How?

By jiggling a signal that's not large enough to cause a bit transition on its own, the added noise pushes it over the transition point for an amount statistically proportional to its actual amplitude level. Our ears and brain, skilled at separating such a signal from the background noise, does the rest. Just as we can follow a conversation in a much louder room, we can pull the weak signal out of the noise.

Going back to our hand-waving analogy, you can demonstrate this principle for yourself. View a large text character (or an object around you), and view it by looking through a gap between your fingers. Close the gap so that you can see only a portion of the character in any one position. Now jiggle your hand back and forth. Even though you can't see the entire character at any one instant, your brain will average and assemble the different views to put the characters together. It may look fuzzy, but you can easily discern it.

When do we need to dither?

At its most basic level, dither is required only when reducing the number of bits used to represent a signal. So, an obvious need for dither is when you reduce a 16-bit sound file to eight bits. Instead of truncating or rounding to fit the samples into the reduced word size—creating harmonic and intermodulation distortion—the added dither spreads the error out over time, as broadband noise.

But there are less obvious reductions in wordlength happening all the time as you work with digital audio. First, when you record, you are reducing from an essentially unlimited wordlength (an analog signal) to 16 bits. You must dither at this point, but don't bother to check the specs on your equipment—noise in your recording chain typically is more than adequate to perform the dithering!

At this point, if you simply played back what you recorded, you wouldn't need to dither again. However, almost any kind of signal processing causes a reduction of bits, and prompts the need to dither. The culprit is multiplication. When you multiply two 16-bit values, you get a 32-bit value. You can't simply discard or round with the extra bits—you must dither.

Any for of gain change uses multiplication, you need to dither. This means not only when the volume level of a digital audio track is something other than 100%, but also when you mix multiple tracks together (which generally has an implied level scaling built in). And any form of filtering uses multiplication and requires dithering afterwards.

The process of normalizing—adjust a sound file's level so that its peaks are at full level—is also a gain change and requires dithering. In fact, some people normalize a signal after every digital edit they make, mistakenly thinking they are maximizing the signal-to-noise ratio. In fact, they are doing nothing except increasing noise and distortion, since the noise level is “normalized” along with the signal and the signal has to be redithered or suffer more distortion. *Don't normalize until you're done processing and wish to adjust the level to full code.*

Your digital audio editing software should know this and dither automatically when appropriate. One caveat is that dithering does require some computational power itself, so the software is more likely to take shortcuts when doing “real-time” processing as compared to processing a file in a non-real-time manner. So, an applications that presents you with a live on-screen mixer with live effects for real-time control of digital track mixdown is likely to skimp in this area, whereas an application that must complete its process before you can hear the result doesn't need to.


Is that the best we can do?

If we use high enough resolution, dither becomes unnecessary. For audio, this means 24 bits (or 32-bit floating point). At that point, the dynamic range is such that the least-significant bit is equivalent to the amplitude of noise at the atomic level—no sense going further. Audio digital signal processors usually work at this resolution, so they can do their intermediate calculations without fear of significant errors, and dither only when its time to deliver the result as 16-bit values. (That's OK, since there aren't any 24-bit accurate A/D convertors to record with. We could compute a 24-bit accurate waveform, but there are no 24-bit D/A convertors to play it back on either! Still, a 24-bit system would be great because we could do all the processing and editing we want, then dither only when we want to hear it.)

This entry was posted in [Digital Audio](#), [Dither](#). Bookmark the [permalink](#).


[← Oversampling](#)
[What is aliasing? →](#)

31 Responses to *What is dither?*

1.  *Michael Thomas* says:
[December 17, 2010 at 9:18 pm](#)


Thats Nice....

[Reply](#)

- o  *Vesna* says:
[April 28, 2012 at 11:42 am](#)

Finally an excellent audio demonstration of why dithering is used. I've been dithering my final mixes at the last mastering stage on the basis that it was recommended by professionals. I knew that dithering had something to do with improving the quality of the final mastered audio but I couldn't really hear why. With these audio examples you can actually hear the disturbance to the waveform in the tail of the non-dithered sine wave as it fades away. Thanks for this excellent explanation.

[Reply](#)

2.  *Torben@orevox.dk* says:
[December 21, 2010 at 6:07 am](#)


Very nice straight forward explanation,
thank you.

[Reply](#)

- o  *jeslapriya* says:
[July 31, 2011 at 4:21 am](#)


u say in audio signal what is dither but in digital image processing, means how its suitable 16 bit can reduce 8bit please mail me
what is dither in digital image processing

[Reply](#)

-  *Nigel Redmon* says:
[August 7, 2011 at 10:01 am](#)


In image processing, the bit depth controls color; dither is used to approximate the correct color over over several pixels when reducing bit depth to the point where the target bit depth can't represent the image's true colors accurately. We see this most often in print media—take a close look at an image in a newspaper. An extreme example of reducing bit depth is to convert a full color image to black and white (not just gray scale, but pure black or white pixels).

[Reply](#)

3.  *tut tut* says:
[December 28, 2010 at 2:48 pm](#)


That's all good and i understand now ..But Does The internet radio stations do dithering such as shout cast or microsoft station guide or radio that is pumping the net too???

[Reply](#)

4.  *Nigel Redmon* says:
[December 28, 2010 at 11:49 pm](#)

I can't say, but in general, it should be done when there is a reduction in resolution. I can't say specifically, not knowing those formats, where or if dithering is done. For instance, such a format might allow compression of 24-bit fixed, or 32-bit float source, and play it back on a computer system that can accept those bit depths, and no dithering is needed. If the format itself reduces the bit depth, then the encoder should dither; if the playback system require a smaller bit depth, then the decoder should dither.

[Reply](#)

5.  *Mark* says:
[February 11, 2011 at 3:38 am](#)

Ehh, is this info still accurate, i mean: it was posted in 1996?
I am pretty shure I have a 24Bit 96Khz interface:S.

I'm not a pro or something so i'm just asking.

thx

[Reply](#)

- o  *Nigel Redmon* says:
[February 11, 2011 at 5:02 pm](#)


Yes, Mark, it's true that we have 24-bit converters now. But not really.

That is, each additional bit gives us half of what the previous bit gave us. By the time you get to the 24th bit, its contribution is so small, that it's now below the thermal noise of any real circuit we can put it in. For instance, if one bit gives us one volt, the 24th bit would give us one ten-millionth of a volt. Considering that the background noise of your circuit needs to be quieter than that in order for it to be effective, you'll need to resort to cryogenics.

But that's not to say that we might as well use 20-bit converters. Converters are typically spec'd based on the accuracy of their least-significant bit. So, the 20th bit of a 24-bit converter is likely to be more accurate (maybe 16 times more accurate, at the same spec) than the 20th bit of a 20-bit converter. So we might as well use 24-bit converters, whether we think we can hear the last couple of bits or not.

Other than that, we're at the limits of physics here—the thermal noise of atoms isn't going to change any time soon. So if someone tries to sell you a 32-bit converter 5 years from now, claiming a breakthrough in knowledge, just have a good chuckle about it.

[Reply](#)

6.  *Shay* says:
[August 5, 2011 at 10:30 am](#)

I think a great example of dither is that of determining whether a die is fair by rolling it many times and examining the distribution. Each roll alone tells you absolutely nothing about whether it's fair; you have no expectation of any particular side coming up. But since the process is random (uncorrelated), multiple rolls can tell you to high accuracy just how fair it is, and which sides come up every so slightly more often than others.

[Reply](#)

7.  *Shamefully Ignorant* says:
[March 30, 2012 at 7:20 am](#)

Nigel- I'm still a little confused. I am pretty sure that your explanation is really thorough and easy to understand for most people who are reading this, but I spent too much time in my life obsessing over learning instruments and not enough time trying to understand physics, so please forgive my total obtuseness. Simply put, I am merely trying to determine whether or not it is beneficial to use one of the dithering options when I bounce a completed track from Logic Pro 8. My resolution is 24 bit and the sample rate is 44100. The program defaults to no dithering. Should I change that to get a higher quality result?

[Reply](#)


- o  *Nigel Redmon* says:
[April 8, 2012 at 2:09 pm](#)

No problem—I know this is a confusing issue...

No dithering is needed on a 24-bit result—it would be a waste of CPU cycles. While some may be inclined to argue that *any* word-length reduction should be dithered, at a point it's just silly to add noise that you won't hear to avoid distortion that you won't hear. At 24 bits—144 dB dynamic range—distortion in the lowest bit will be buried under the noise floor of any electronics that you will play it through.


To put it another way: When you shorten the word length, you create a distortion; we dither to change the nature of the distortion to one that's less irritating to us. So, if the distortion level is too low to hear, why bother changing its form?

[Reply](#)

8.  *M.* says:
[April 25, 2012 at 2:54 pm](#)

This is seriously a great response to a question I've had for a long time. Thank you!

[Reply](#)

9.  *Mark Heath* says:
[January 2, 2013 at 6:18 am](#)

Thanks for this, I've only recently discovered your blog and I'm working my way through it. Very helpful stuff. In my audio code I sometimes convert from floating point (32 bit IEEE float in range -1 to 1) to short by multiplying by 32767 and casting to a 16 bit int. If I wanted to add dither to this, I'd add in some random noise in the range +/- 1.0 after multiplying but before truncating? Or should the amplitude of the dither noise be +/- 0.5?

[Reply](#)

- o  *Nigel Redmon* says:
[January 2, 2013 at 10:33 am](#)


Hi Mark,

Yes, the noise should be ± 1.0 —two times the LSB size, peak to peak. This is TPDF (triangular probability density function) noise, which you can make by adding two random numbers of size ± 0.5 LSB (note that you need only generate a single random number per sample period, and add it to the previous one for TPDF*). Another way to look at it is that you need noise of ± 0.5 LSB to jog in-between values to the next higher or lower bit level in a statistical manner, before truncation, but this alone would leave the output subject to noise modulation; you need a second ± 0.5 noise to decouple that, resulting in a ± 1.0 TPDF dither.

* That actually gives you lowpass-filtered noise; subtracting the previous noise sample gives a highpass-filtered response and is probably better, perceptually, but you should really use more sophisticated noise shaping if that's what you're after. Generating a second random number instead of the delay gives a flat response. Ideally, you'll use noise shaped dither for the final bit-reduced output to make the noise least noticeable. You'll use flat dither if you need to do further significant processing, though it's rare that you would need to dither anywhere except the final output, since most processing chains maintain 24 bits of more of resolution.

Nigel

[Reply](#)

10.  [Ian Shepherd](#) says:
[February 15, 2013 at 7:56 am](#)

Great post !

I disagree with your final point, though. Even at 24-bit, the cumulative effects of truncation distortion are much more audible than multiple layers of dither – so I recommend people use dither all the time, to avoid this pitfall. Of course if they only export once from a 32-bit floating point DAW, the practical effects may not be that different, but I still think it's the best advice.

[Reply](#)

o  [Nigel Redmon](#) says:
[February 15, 2013 at 10:01 am](#)

Hi Ian,

Sure, it's a valid point—and I would have been disappointed if not one person disagreed with me.

Whether anyone could really hear the difference could only be settled by defining a particular case in which you could hear it. It's my opinion that such a case would be so far removed from practical use that it's not worth considering. If not, we're already in a lot of trouble because for most systems in use today, the pipe is either 24-bit (legacy Pro Tools) or 25-bit (32-bit float host processing), and few plug-ins dither their output. So, you already have a lot of accumulated truncation, and you could manufacture a situation where you had a lot of extremely low-level outputs (tails for instance) getting dithered at once.

If this wasn't already enough, lack of correlation might make the accumulated truncation sound indiscernible from dither, so you'd want to do this with essentially the same signal multiplied many times...the point is that the closer you get to manufacturing a case that you could hear, the farther you get from any situation that you would ever find yourself in.

Note the “practical” in the website tagline, not “ideal”. I think that it's worth doing internal processing in double precision. And if the developer feels they want to dither before sending it back through a single-precision pipe, fine. But I think that there are almost always practical tradeoffs already being made in the processing that far outweigh the minute error passed along by truncating to single-precision floating point.


Ultimately, I expect everyone might move to 64-bit pipes, if for no other reason than the competition did. So, by the time it could matter, it won't matter.

Again, I consider your point valid—I'm just giving my rationale here, not proclaiming you “wrong”.

Nigel

PS—Everyone should check out Ian's blog—I've just read a few posts so far, but it's excellent. Nice work, Ian!

[Reply](#)

■  [Ian Shepherd](#) says:
[February 17, 2013 at 3:48 pm](#)

Thanks, Nigel.

To be honest, I think the vast number of legacy Pro Tools systems out there is justification enough !

And, I'd actually say that it would be easier to create an audible example than you suggest – truncation distortion is quite easily audible at 16-bits, even at normal listening levels and not just with reverb tails etc – just as different dither flavours are, too. I'm constantly amazed by how sensitive our ear is to “un-natural” sounds.

For what it's worth, Paul Frindle agrees – all the Oxford DSP is dithered at 24 bits, even in a 32-bit float host. I was talking to him on Facebook a while back and he put it very well:

“The thing is that there is actually no difference between digital and analogue signals – all have a dynamic range set by the ratio between the max level and noise. The difference is that analogue comes with it's own noise (caused by the reality of signal in the physical world) whereas any digital representation in math requires us to re-insert the physical random component the math does not provide us.

It is a theoretical requirement of the system, it doesn't mask the distortion – it removes it... ANY digital data representation of a signal in the real world has artificial certainty (which reality doesn't) and it has to be removed for the signal to be harmonically accurate – i.e. like a signal in the real world... It's a deep subject that shows our math is an artificial human approximation of reality – but the approximation has too much certainty. Fascinating implications to that concept...”

For me, lack of dither is a fault, so we should always use it. And truncation distortion sounds so heinous that dither is always preferable, even at 24-bit.

(I made a video demonstrating the sound of truncation distortion, your reader might like to check it out ? It's here:

<http://productionadvice.co.uk/dither-or-distort/>

[Reply](#)



- *Nigel Redmon* says:
[February 20, 2013 at 11:29 am](#)

Hi Ian,

You bring up enough good points that I think I'll revisit the topic in a new article. But briefly...

I think you misunderstood my “legacy ProTools” comment. The point was that most DAWs and plug-ins work at 24- or 25-bit resolution—legacy ProTools probably being the most common example of 24-bit. And the vast majority of systems and plug-ins truncate to that bit depth. So, the comment was simple that if that isn't good enough, then we're already in a lot of trouble (which isn't to imply that everything is satisfactory, but that you'd have to replace about everything you have now in order to compare).

A few quick points:

- I like your youtube video very much, but you imply that 24-bit truncation is bad, but 32-bit float is fine—bear in mind that there is only 6 dB difference between the two (25-bit mantissa for 32-bit float).
- The good news: Pretty much anything you record will come with its own dither signal for free, because electronics of any kind generates more thermal noise than the 24-bit floor.
- You can't hear the error in a signal truncated to 24 bits because it's buried in the thermal noise of the 24-bit converters and subsequent electronics. That does still leave your contention that those signals can add up if we're talking about summing many truncated 24-bit signals. But you'd be hard-pressed to show that mathematically, because you'd need to do something extremely artificial, such as many copies of the same sine wave, so the errors add and are not simply distributed statistically.

Again, I'm not against it—plain dither is not very costly. I'm just trying to put perspective on its value. There are many higher priorities.

Nigel

PS—I see that you acknowledged the “self-dither” of recorded material in your blog article. This is already too long, so I'll address subsequent dither in an article...

[Reply](#)



- *Nigel Redmon* says:
[February 20, 2013 at 3:00 pm](#)

Ian—I read your other article (“When should you use dither?”), reference by your link, and I think we're pretty close to the same thoughts on the subject. The only difference is that my broader comments are from the point of view of writing DSP code (plug-ins, etc.), and your broader comments focus on final output, such as files.

Reading more closely, you say, “I do agree you should only use it once within a single DAW application, on the main stereo output – so you don't need it on every channel, or in-between plugins, for example – but you do need it once when saving your file before mastering, and then again when exporting the final 16-bit master.”

Personally, I work with 32-bit float sound files, which is the native format of my DAW (which is to say that for me it's worth using 33% more storage to avoid the conversions, audible or not). So when I say it's probably not worth dithering to 32-bit float or 24-bit fixed, I'm talking about within or between plug-ins. And while I believe you'd ever hear the difference between a 24-bit truncated final file or dithered, it certainly doesn't hurt to dither it and it's at no cost, basically, so why not.

Nigel

[Reply](#)



■ [Ian Shepherd](#) says:

[February 20, 2013 at 4:38 pm](#)

Hi Nigel,

Thanks for the detailed replies.

Your point about "self-noise" is fair enough, but of course it stops being of any benefit as soon as fixed-point arithmetic is used to process it in a fixed-point DAW. So if your assertion that most plugins truncate to 24-bit in legacy Pro Tools is correct, it's users *are* already in trouble, IMO. Or at least they are losing quality unnecessarily.

Imagine a signal comes in "self-dithered", but the fader is not at zero – truncation. Then it goes through three or four plugins – truncated. Then it gets sub-mixed to a buss – truncated. That buss has it's level tweaked (and truncated). This buss is then passed through several more plugins, all of which truncate. And finally all the buses are summed (and truncated) and go through master buss plugins – which truncate.

Now imagine that for a 64-track mix, and consider that many mixers insert a default "set" plugins on every single track as standard... that's one hell of a lot of truncation distortion accumulating !

Happily most other DAWs have been floating-point for a long time, of course. But Pro Tools is still the "industry standard"...

I think Pro Tools HD was 48-bit fixed, which will be far less of a problem, but there are a lot of non-HD systems out there.

I also have a concern that some plugins may essentially work in fixed point internally, and just interface with the DAW using floating-point – but there's no way of knowing for sure.

The reason I say dither is not as important for floating-point is because of the exponent scaling, which should keep mantissa truncation so far below the signal level that we can stop worrying about it. I have read that there may be an issue with the noise-floor modulating in a signal-correlated manner as this happens, but my maths starts to get shady at this point !

So to summarise – assuming correctly implemented floating-point maths in the DAW and all it's plugins, I agree – dither is probably optional apart from the final fixed-point file. But since we KNOW this isn't the case with legacy Pro Tools, which is incredibly widespread, and it MAY not be something we can rely on for all plugins, and since simple dither is cheap to implement and benign in sound, I still think the best practise is to use it wherever possible...!

Ian



■ [Nigel Redmon](#) says:

[February 20, 2013 at 5:54 pm](#)

You bring up a number of good points that illustrate how complex the topic is, Ian...


Pro Tools HD has a 48-bit mix engine, but the plug-in chain is 24-bit. Yes, I expect that most don't dither.

Regarding fixed point, yes the exponent scaling buys you more dynamic range without losing the bits of interest. But consider the context: If you save it to a 24-bit file, truncation. If you add it to another signal, truncation. If you change the gain, even slightly, truncation. In terms of whether or not to dither, 32-bit float is essentially 25-bit fixed.

Since 24-bit Pro Tools hardware has 56-bit accumulation, the essential difference between it an host-based floating point is one bit, and the optimal way to order your calculations. (In fact, the convenience factor of floating point probably lulls many into adopting poor architectural choices. Floating point does not save you from the treachery of adding big to small, for instance. Although if you do everything in double precision you can usually get away with it.)

About error accumulation: It's not something I've done research on, but do bear in mind that the accumulation is not correlated. The more you get, the more it will likely approach noise...

A key point: Dither only changes the correlation of the quantization error.

11.  *Steve Kralik* says:
[September 9, 2013 at 1:00 am](#)

This thread is a fantastic source of information, however I am a complete beginner in my exploration of the digital audio field. I do not understand most of the conversation, particularly nearer and nearer to the bottom. I have ten basic areas of question. Please answer anything you wish, opinions are just fine. I would make this a separate thread, but I feel like my questions are too closely related to this article and its many contributing comments. Please be specific to indicate which of my questions you may be answering:

1:

What levels of experience and education in the audio field are necessary to debate in the way that the both of you have?

2:

I have an example question to help me understand this better...

Files purchased on iTunes open in Adobe Audition CS6 as "QuickTime: MPEG-4 Audio" (lossy M4A), and "32-bit (float)" at 256 kbps, 44.1 kHz. Is the noise floor that I hear on virtually every existent iTunes-purchased song added dithering? If so, isn't that dithering unnecessary on systems that natively play (or force) 32-bit float audio? On systems that do not play 32-bit float?

3:

If iTunes (ONLY as an example) does add dithering like most professional masters do, can it theoretically be safely removed via selective noise reduction in order to create a collection of music (32-bit float or higher) that plays with no audible errors, distortion, or the original static?

4:

By removing that static, even if I were extremely careful, will I remove anything I shouldn't? I know that amplitude and frequency range are not the same thing, but I wonder if after removing that static I will also be removing parts of the music? If so, are the parts that would be removed the parts that would've caused the errors that would occur without dithering?—The type of errors discussed throughout and below the article?

5:

Would it ever be wise to output files as 32/64-bit Integer or Floating Point (IEEE) since I can do so with Adobe Audition CS6 or CC?

6:

Is there a "best of both worlds" approach as to when to dither, and integer vs. float?

7:

In the end, is the best all-purpose master a minimally-dithered, lossless, 16-bit, 44100-Hz file? Would this be due to it's often native-decoding capability on 16-bit equipment? Does the average CD (if well-encoded) truly provide the most logical source of audio files for all equipment?

8:

As a proponent of high-end audio hardware (amplifiers, speakers, etc.), I generally prefer the amplifiers made by McIntosh Laboratories, Inc. "Mac Amps" have patented "PowerGuard" and "SentryMonitor" circuits designed to jointly remove clipping, and apply noise reduction by analyzing differences in the waveforms in order to cancel out those differences. With only that information in mind, would you estimate that audio output from McIntosh amplifiers is compromised (excluding possible damage to the amps, speakers, and other hardware)?

9:


What is the best audio software (or group of programs) for the purposes of remastering (regardless of cost)? What about for mixing? Music creation? Notation? Conversion? Analysis? I noticed Pro Tools was mentioned, what about programs by a company known as "Channel D"?

10:

I love audio, but only as perhaps my most passionate hobby, not a career... with that in mind, where should I start if I want to start learning all of this on my own?

Thank you so very much for any answers you may provide me with.


[Reply](#)

- o  *Nigel Redmon* says:
[September 25, 2013 at 2:06 pm](#)

For many of these general questions, I refer you to sites such as kvraudio and gearslutz for discussion. You'll find a lot of passionate people eager to discuss and debate many of these topics.

3: No... We add dither (noise) to randomize the error created when we throw away information (the lesser bits). Removing the noise won't get back what we discarded.

[Reply](#)

12.  *Tom* says:
[October 22, 2013 at 10:38 pm](#)


What I don't understand is, say you have a 16 bit depth system, and you record a tone at full bit depth, OK, so little distortion there. But now, record a tone, that is very small in amplitude, and only using say 1 or 2 bits. Have you not lost resolution there.... what is the result with and without dither. I would suspect that distortion has to go up as you capture smaller signals vs larger ones.

[Reply](#)

- o  *Nigel Redmon* says:
[October 22, 2013 at 11:27 pm](#)

Yes, Tom, that's the problem—a threshold effect at low bit levels. Consider a similar problem if you try to convert a grayscale image to black and white using a 50% threshold: areas that are slightly less than 50% appear as white, and slightly more appear as black, obscuring the fact that both are about half way between. Now, if you add a small random value to each pixel, the areas of the lighter gray will be mixed, but with more white pixels than black, and the darker gray with more black pixels than white. The result is that you've preserved some detail by adding noise.

[Reply](#)

13.  *David Buitrago* says:
[February 22, 2014 at 7:13 pm](#)


Hola
Great Explanation!, could you please give me some bibliography to dive into this topic? Thanks.

[Reply](#)

- o  *Nigel Redmon* says:
[February 26, 2014 at 10:38 pm](#)


Glad you enjoyed it, David. At this point (wow, 1996, time flies), I'd have to do the same thing you'd do in looking for good material on dither—google. But I do have a video on dither that I just need to finish up some charts for, and it should be posted soon...

[Reply](#)

14.  *Bob Dylan* says:
[March 28, 2015 at 11:08 am](#)

Wow, very clear thank you!!!

[Reply](#)

15.  *Gabe* says:
[April 16, 2015 at 9:29 pm](#)

So if a master recording already has a fairly high noise floor (say -75dBFS from tape hiss), is dither still necessary at the final stage or is the noise inherent in the recording serving the purpose of dither?

[Reply](#)

- o  *Nigel Redmon* says:
[April 16, 2015 at 10:29 pm](#)

Hi Gabe—I hope you've seen my latest video on dither, but the gist is this: Yes, tape hiss of that level would certainly dither at 16-bit, as dither is basically trying to mimic that level of randomness. However, you might as well dither without question for 16-bit, because the dither noise at around -90 dBFS isn't going to be heard over that much louder tape hiss anyway.

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website



CAPTCHA Code *

Comment

You may use these HTML tags and attributes: <abbr title=""> <acronym title=""> <blockquote cite=""> <cite> <code> <del datetime=""> <i> <q cite=""> <strike>

• Search for:

• Recent Posts

- [Dither—The Naked Truth video](#)
- [Dither widget](#)
- [Audio Dither Explained video](#)
- [More about source code](#)
- [Pole-Zero placement v2](#)
- [Biquad calculator v2](#)
- [Envelope generators—ADSR widget](#)
- [Envelope generators—ADSR video](#)

• Archives

- [January 2015](#)
- [March 2014](#)
- [January 2014](#)
- [October 2013](#)
- [June 2013](#)
- [May 2013](#)
- [April 2013](#)
- [March 2013](#)
- [February 2013](#)
- [December 2012](#)
- [November 2012](#)
- [May 2012](#)
- [March 2012](#)
- [January 2011](#)
- [December 2010](#)
- [November 2010](#)
- [July 2007](#)
- [March 2003](#)
- [February 2003](#)
- [August 2002](#)
- [January 1997](#)
- [November 1996](#)
- [October 1996](#)
- [August 1996](#)

- [July 1996](#)

- **Categories**

- [Digital Audio](#) (42)
 - [Aliasing](#) (8)
 - [Convolution](#) (3)
 - [Dither](#) (7)
 - [Filters](#) (17)
 - [DC Blocker](#) (1)
 - [FIR Filters](#) (5)
 - [IIR Filters](#) (11)
 - [Biquads](#) (7)
 - [Fourier](#) (2)
 - [FFT](#) (2)
 - [Impulse Response](#) (5)
 - [Jitter](#) (2)
 - [Oscillators](#) (8)
 - [Wavetable Oscillators](#) (8)
 - [Phase](#) (2)
 - [Reverb](#) (2)
 - [Sample Rate Conversion](#) (8)
 - [Sampling Theory](#) (1)
- [Envelope Generators](#) (5)
- [MIDI](#) (2)
- [Source Code](#) (7)
- [Synthesizers](#) (7)
- [Uncategorized](#) (2)
- [Video](#) (4)
- [Widgets](#) (8)