

Re: Multiplexing and mixing

Source: <http://sci.tech--archive.net/Archive/sci.electronics.design/2009-07/msg00173.html>

- *From:* "Bob.Jones5400@xxxxxxxx" <bob.jones5400@xxxxxxxx>
 - *Date:* Wed, 1 Jul 2009 12:36:43 -0700 (PDT)
-

On Jun 29, 11:13 am, whit3rd <whit...@xxxxxxxx> wrote:

On Jun 27, 1:44 pm, "Bob.Jones5...@xxxxxxxx"

<bob.jones5...@xxxxxxxx> wrote:

I want to design a system, unless something similar exists, that multiplexes digital audio signals but also allow for mixing of those signals in arbitrary ways. This should be relatively cheap because I want to use a lot of them.

The signal is arranged Sample1_audio1, sample1_audio2, etc...

This is the domain of sound programs (like Logic Pro on a Macintosh). Smaller numbers of channels are do-able with the nearly-free software, like GarageBand and iDVD.

In any case it should not be difficult to "mix" to signals. I would just extract a sample, multiply it by a the "mix" value, and add it to a some a buffer.

But how difficult will this be to implement using discrete components (IC's but not fpga's or uP's)? The number of signals is fixed around 20(the more the better). The audio rate is ~200khz @ 24-bit.

Ahh, you don't want to use a standard audio encoding. If your samples are all the same, it's easier than if they aren't, but you WILL have to reinvent the wheel. It'll be harder if you insist on 'using discrete' than if you stick with massproduced computing machinery.

Re: Multiplexing and mixing

If it were an analog signal, you could buffer all the inputs with high output impedance gain-controlled amplifiers (LM13700 is a dual of one such amp), and just connect all the outputs together (which is allowed because they have high output impedance). The 'control' function would then just be generating slow waveforms to program all the gain-control pins, which is relatively easy.

This is an embedded solution and does not work with a pc or similar device. Because it is actually a mixture of digital and analog I need some way to pass around signals easily as I do not want to hardwire the signal chain.

.