Holy City Audio Forum

Former Home of SpinCAD Designer

Thoughts on reverb building blocks

Post a reply

- Edit post (./posting.php?mode=edit&f=42&p=1979)
- Delete post (./posting.php?mode=delete&f=42&p=1979)
- Report this post (./report.php?f=42&p=1979)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=1979&sid=d6d1dcc4501596550704606791e62b4c)
- Reply with quote (./posting.php?mode=quote&f=42&p=1979)

Thoughts on reverb building blocks (#p1979)

by **Digital Larry** » Sat Nov 15, 2014 7:18 am

Recently I created a block for reverbs that lets you select the # of all passes up front and the # of delay blocks in the reverb ring. One of my forum members wanted more control over all of the delay and all pass block lengths, which I didn't feel like doing, as it's a lot of coding effort (still) from what I felt was excessive complexity to present to the end user (as in "how do I know what to set these numbers to, other than just trying and seeing what happens?").

But I did have a thought on something that might help people build up unique reverb structures. Whether to make every single thing adjustable is still up in the air.

First, I could easily create a compound series all-pass block that could have, say, 2 to 6 stages, with lengths adjustable over some range.
Then, I could have a reverb ring block, with two inputs, as well as two outputs. The two inputs would allow you to take "something" into one part of the reverb ring and "something else" into a different part of the reverb ring.

All reverbs I've looked at so far take a cascade of 4 all passes prior to entering the reverb ring, but in the Spin knowledge base there is also discussion about using a multi-tap delay for early reflections, so this could be another option. All passes on one side, multi-taps on the other.

Within the established limitations of the FV-1, this would also let you do things like apply filtering, dynamic processing, or modulation to the signal between the all-passes and the reverb ring. I don't know what would happen. It might be great or it might suck.

Top

- Edit post (./posting.php?mode=edit&f=42&p=1980)
- Delete post (./posting.php?mode=delete&f=42&p=1980)
- Report this post (./report.php?f=42&p=1980)
- Warn user (./mcp.php?i=warn&mode=warn_post&f=42&p=1980&sid=d6d1dcc4501596550704606791e62b4c)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=1980&sid=d6d1dcc4501596550704606791e62b4c)
- Reply with quote (./posting.php?mode=quote&f=42&p=1980)

Re: Thoughts on reverb building blocks (#p1980)

by **Jacko** » Sat Nov 15, 2014 7:45 am

DL,

I think that basics are more important than complex blocks with lots of parameters. A basic delay block could be used to build reverbs. A basic all-pass block could be added as needed - last I looked, there was no AP block in the prog.

Pre-built complex block are great but they can be limiting, as in there is no flexibility other than the few options available that are built in it.

IMO, you need a few basic blocks and then a couple of tutorials on how to combine them into other effects units. Let's have a basic comb filter block that you can adjust the length, feedback and feedback filter... this is the basic building block of most reverbs. Combine a few comb filters with some all-pass stages and then you have built a reverb effect.

A simple delay block would be useful - just delay time for control. Nothing else. You could use it as a predelay or as a standalone delay unit.

My 2 cents worth...

JACKO

Top

- Edit post (./posting.php?mode=edit&f=42&p=1981)
- Delete post (./posting.php?mode=delete&f=42&p=1981)
- Report this post (./report.php?f=42&p=1981)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=1981&sid=d6d1dcc4501596550704606791e62b4c)
- Reply with quote (./posting.php?mode=quote&f=42&p=1981)

Re: Thoughts on reverb building blocks (#p1981)

by **Digital Larry** » Sat Nov 15, 2014 8:25 am

Hey Jacko,

I do agree with that concept. And what holds me back is the inefficient register allocation to link blocks where it is not needed. Building up reverbs from really small pieces would use a lot of blocks and a lot of registers would get wasted in the process. But ultimately I think you are right. I just need to back up a bit and try to figure out how to handle that!

Thx,

DL

Top
- Edit post (./posting.php?mode=edit&f=42&p=1982)
- Delete post (./posting.php?mode=delete&f=42&p=1982)
- Report this post (./report.php?f=42&p=1982)
- Warn user (./mcp.php?i=warn&mode=warn_post&f=42&p=1982&sid=d6d1dcc450159655070460679 1e62b4c)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=1982&sid=d6d1dcc450159655070460679 1e62b4c)
- Reply with quote (./posting.php?mode=quote&f=42&p=1982)

Re: Thoughts on reverb building blocks (#p1982)

by **Jacko** » Sun Nov 16, 2014 6:43 am

I had been thinking about this previously... how about some reserved registers for use with combinations of blocks?

Start with a 1-4 Mixer that sets up a series of parallel strings. Each of the outputs is followed by a series of blocks that inherit the register to be used from the block preceding it. For example, Out1 of the mixer writes to Reg32. Then the next block in series inherits the Reg32 as its signal path... read Reg32, process it however and then write Reg32 as its output too. The next block gets its register to use from the block before (Reg32) so it also reads Reg32, processes then writes Reg32.

A parallel signal path from Out2 of the mixer could use Reg31 for the same series process. Read, process, save.

Series combinations of blocks would only use one register for their main signal path.

You could also reserve several blocks as temporary registers. They would be used internally in a block but their value from block to block would not be guaranteed since they are temporary and reset/reused in the next block.

8 or 10 registers could be dedicated to this whole process.

I'm sure you have probably considered something like this, or have even better ideas! Sometimes you just need to hear someone say it to get you started down the path...

Best Regards, JACKO

Top
- Edit post (./posting.php?mode=edit&f=42&p=1983)
- Delete post (./posting.php?mode=delete&f=42&p=1983)
- Report this post (./report.php?f=42&p=1983)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=1983&sid=d6d1dcc450159655070460679 1e62b4c)
- Reply with quote (./posting.php?mode=quote&f=42&p=1983)

Re: Thoughts on reverb building blocks (#p1983)

by **Digital Larry** » Sun Nov 16, 2014 7:42 am

Hi Jacko,

Thanks for the discussion.

In the software world there are indeed many ways to solve problems. And in my world, since I am again working full time (not on SpinCAD) then I have to slice my improvement efforts into small pieces.

I've written some ideas down elsewhere around here about coding patterns that SpinCAD generates which could be identified and optimized by running a second pass through the code. The whole issue boils down to this:

All SpinCADBlocks place their output (in the accumulator) into a register which was allocated for that block's use.
Lots of times, that register simply gets read back into the accumulator for processing by the next block.
So, why not just leave it in the accumulator and skip putting into the register?

As far as I can tell, you have to render the entire model first to determine whether a given block routes to only one or multiple destinations.

The source code is available to anyone who wants to dive in and improve my klutzy brute force implementations of things. I'm not even being humble or sarcastic here.

However, prior to any of this, I want to deal with the LFO allocation much as I have the registers, instructions, and delay memory. For example, some reverbs have Sin LFOs in them for a smoothing function. If you try to use one of those at the same time using another block with the same LFO allocated, you'll get unpredictable results and will have a hard time figuring out why unless you are really good at reading Spin ASM and have a suspicious nature (that is suspicious that SpinCAD COULD have bugs or problems - which it certainly does).

So THAT I am having a little trouble figuring out. But my thought is that I simply keep count of every time the WLDR or WLDS instruction is referenced during code generation. For each LFO (sin/Ramp 0/1) that should only happen once.

I have also tried as hard as I can to maintain a separation in functionality between the lower level ElmGen code and SpinCAD. So, to keep that separation, I might not go into ElmGen to keep a count of LFO setup references, I'd rather scan the rendered program after I'm done with code generation and count the WLDR/WLDS references there. See, just by having this little chit chat I came to a point of greater clarity about how I should approach things.

Thanks!

DL

Top
Display posts from previous: All posts Sort by Post time Ascending Go
Post a reply
5 posts • Page **1** of **1**
Return to Developer's Corner
Jump to: Developer's Corner Go
Quick-mod tools: Lock topic Go

Who is online

Users browsing this forum: Digital Larry and 0 guests

Administration Control Panel