

Holy City Audio Forum

Former Home of SpinCAD Designer

[Skip to content](#)

Search...	Search
-----------	--------

[Advanced search](#)

[[Moderator Control Panel](#)]

Some simple modular block ideas

Post a reply

Search this topic...	Search
----------------------	--------

16 posts • Page **1** of **1**

- [Edit post](#) (./posting.php?mode=edit&f=31&p=1308)
- [Delete post](#) (./posting.php?mode=delete&f=31&p=1308)
- [Report this post](#) (./report.php?f=31&p=1308)
- [Warn user](#) (./mcp.php?i=warn&mode=warn_post&f=31&p=1308&sid=befaf904784df501ab581b011c284453)
- [Information](#) (./mcp.php?i=main&mode=post_details&f=31&p=1308&sid=befaf904784df501ab581b011c284453)
- [Reply with quote](#) (./posting.php?mode=quote&f=31&p=1308)

Some simple modular block ideas (#p1308)

by **slacker** » Mon Jun 03, 2013 12:57 pm

Hi Gary

I said I'd have a go at writing some simple blocks in Spin asm that you could possibly adapt in blocks, here's a first attempt at a couple.

First a simple delay, this is just a delay line, any mixing and feedback to make an effect would be done with other blocks. The maximum length can be set anywhere up to 1 second. The control signal from a pot or whatever is then scaled accordingly so the range 0 - 1 always sweeps the whole delay line irrespective of it's length. The control signal is smoothed to prevent zipper noise.

```
;Single delay line dry only
equ maxlength 1000; max length of delay in milli seconds 0 - 1000
equ smooth 0.00125
equ del_read reg0
mem delay 32767*maxlength/1000
equ control pot0
```

```
rdax adcl,1
wra delay,0
clr
or 32767*256
mulx control
sof maxlength/1000,0
rdfx del_read, smooth
wra del_read,1
wra addr_ptr,0
```

```
rmpa 1
wrax dacl,0
```

The second is a short delay that can be modulated suitable for making flangers, vibrato or chorus effects.

```
;modulated delay for flanger etc
equ control pot0
mem model 8192

;set up LFOs, only at start up
skp run, START
wldr rmp0,0,4096          ;ramp0 used for modulation operations

START:

rdax adcl,1
wra model,0              ;write to head of delay
cho rdal,rmp0            ;servo ramp0 to correct position using value in control
rdax control,-0.25        ;scaled so that 0 - 1 sweeps full range
wrax rmp0_rate,0
cho rda,rmp0,reg|compc,model ;read from delay
cho rda,rmp0,0,model+1
wrax dacl,0
```

If these are any use to you let me know and I'll do a few more, I'm thinking some LFO's of different shapes would be useful.

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1310\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1310\)](#)
- [Report this post \(./report.php?f=31&p=1310\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1310&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1310\)](#)

Re: Some simple modular block ideas (#p1310)

by **Digital Larry** » Mon Jun 03, 2013 9:23 pm

Well it's getting a little late and while I'm tempted to stay up and finish this, tomorrow my coworkers will appreciate my at least having attempted to get some beauty sleep. 🙄 So I will put out a new beta release tomorrow.

I implemented the modulated delay first. It helps to point out the problems with the Ramp LFO simulation! I hope you find it amusing, but don't think this is what your patch will really sound like.

Also I wanted to let you know that oddly enough, I have to implement feedback within a code block rather than looping back around several blocks, even if the editor will allow you to draw it. It has

to do with the algorithm I use to sort the blocks and it will not be happy with a circular path in the model. So let me know how you'd like to implement a feedback control.

In general, if the control input pin of the block is not connected to a control output, you can set the controlled parameter to a default value, which you adjust on the block's control panel. In fact, you could do anything you like if the control isn't connected, including creating a different effect entirely, but setting the controlled parameter to some fixed value makes the most sense. So for example, in the case of a feedback input pin, when it is not connected I could have no feedback, and not generate any code associated with possible feedback, or it could be at 50% fixed, or it could be set via a slider on the control panel.

Here's the Java code for the delay itself:

```
/* SpinCAD Designer - DSP Development Tool for the Spin FV-1
 * ModDelayCADBlock.java
 * Copyright (C)2013 - Gary Worsham
 * Based on ElmGen by Andrew Kilpatrick
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

package com.holycityaudio.SpinCAD.CADBlocks;

import com.holycityaudio.SpinCAD.SpinCADPin;
import com.holycityaudio.SpinCAD.fxblocks.basic.SpinFXBlock;

public class ModDelayCADBlock extends ModulationCADBlock {

    private static final long serialVersionUID = 695539935034103396L;
    int delayLength = 8192;

    public ModDelayCADBlock(int x, int y) {
        super(x, y);
        addControlInputPin(this, "Modulation In");
        setName("Mod Delay");
    }
}
```

```
private void modDelay(SpinFXBlock sfxb, int chorusLength) {
    int input = -1;
    SpinCADPin p = this.getPin("Audio Input 1").getPinConnection();
    if (p != null) {
        input = p.getRegister();
        int chorout = sfxb.allocateReg();

        int Control1 = -1;

        p = this.getPin("Control Input 1").getPinConnection();
        if (p != null) {
            Control1 = p.getRegister();
        }

//          ;modulated delay for flanger etc
//          equ control pot0
//          mem model 8192
        sfxb.FXallocDelayMem("model", delayLength);

//          ;set up LFOs, only at start up
//          skip run, START
        sfxb.skip(SKIP_RUN, 1);
//          wldr rmp0,0,4096          ;ramp0 used for modulation operations
        sfxb.loadRampLFO(0, 0, 4096);
//          START:

//          rdax adcl,1
        sfxb.readRegister(input, 1.0);
//          wra model,0          ;write to head of delay
        sfxb.FXwriteDelay("model", 0, 0);
//          cho rdal,rmp0          ;servo ramp0 to correct position using value
in control
        sfxb.chorusReadValue(CHO_LFO_RMP0);
//          rdax control,-0.25      ;scaled so that 0 - 1 sweeps full range
        sfxb.readRegister(Control1, -0.25);
//          wrax rmp0_rate,0
        sfxb.writeRegister(RMP0_RATE, 0);
//          cho rda,rmp0,reg|compc,model      ;read from delay
        sfxb.FXchorusReadDelay(CHO_LFO_RMP0, CHO_REG | CHO_COMPC,
"model", 0);
//          cho rda,rmp0,0,model+1
        sfxb.FXchorusReadDelay(CHO_LFO_RMP0, 0, "model", 1);
//          wrax dacl,0
        sfxb.writeRegister(chorout, 0);
        this.getPin("Audio Output 1").setRegister(chorout);
        System.out.println("Mod Delay code gen!");
    }
```

```

    }
}

public void generateCode(SpinFXBlock sfxb) {
    modDelay(sfxb, delayLength);
}

// control panel functions

public void editBlock(){
    new ModDelayControlPanel(this);
}

public int getDelayLength () {
    return delayLength;
}

public void setDelayLength (int l) {
    delayLength = l;
}
}

```

And here's the Java code for the control panel which sets the delay length:

```

/* SpinCAD Designer - DSP Development Tool for the Spin FV-1
 * ModDelayControlPanel.java
 * Copyright (C)2013 - Gary Worsham
 * Based on ElmGen by Andrew Kilpatrick
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

package com.holycityaudio.SpinCAD.CADBlocks;

```

```
import javax.swing.BoxLayout;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JSlider;
import javax.swing.SwingUtilities;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;

class ModDelayControlPanel {

    private ModDelayCADBlock mD;
    private JSlider delaySlider;
    private JLabel delayLabel;
    private JFrame frame;

    public ModDelayControlPanel(ModDelayCADBlock modDelayCADBlock) {
        this.mD = modDelayCADBlock;

        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                frame = new JFrame();
                frame.setTitle("Mod Delay");
                frame.setLayout(new BoxLayout(frame.getContentPane(),
BoxLayout.Y_AXIS));
                delaySlider = new JSlider(JSlider.HORIZONTAL, 4096, 16384,
mD.getDelayLength());
                delaySlider.addChangeListener(new bitSliderListener());
                delayLabel = new JLabel();
                frame.add(delayLabel);
                frame.add(delaySlider);
                updateDelayLabel();
                frame.setVisible(true);
                frame.pack();
                frame.setLocation(mD.getX() + 200, mD.getY() + 150);
            }
        });
    }

    class bitSliderListener implements ChangeListener {
        public void stateChanged(ChangeEvent ce) {
            if(ce.getSource() == delaySlider) {
                mD.setDelayLength(delaySlider.getValue());
                updateDelayLabel();
            }
        }
    }
}
```

```

public void updateDelayLabel() {
    // TODO Auto-generated method stub
    delayLabel.setText("Delay length: " + String.format("%d",
mD.getDelayLength()));
}

```

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1314\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1314\)](#)
- [Report this post \(./report.php?f=31&p=1314\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1314&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1314\)](#)

Re: Some simple modular block ideas (#p1314)

by **Digital Larry** » Tue Jun 04, 2013 9:03 pm

New beta release is up. 😊

[viewtopic.php?f=29&t=1274&p=1313#p1313](http://holycityaudio.com/forum/viewtopic.php?f=29&t=1274&p=1313#p1313) (<http://holycityaudio.com/forum/viewtopic.php?f=29&t=1274&p=1313#p1313>)

A few comments on the blocks:

- 1) For the first one, single delay, there is a control panel setting which sets the maximum delay time. Then the control input (which you'd typically attach to a pot) scales down from this setting.
- 2) I notice also that you used control smoothing on the delay time instead of the more usual control quantization. Any particular reason?
- 3) As I mentioned in my [post on block design philosophy \(http://holycityaudio.com/forum/viewtopic.php?f=31&t=1266\)](http://holycityaudio.com/forum/viewtopic.php?f=31&t=1266), these blocks include special control processing such as scaling and control smoothing. This probably results in a more usable block at the expense (possibly) of some flexibility. I'm still on the fence as to which approach I prefer.

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1319\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1319\)](#)
- [Report this post \(./report.php?f=31&p=1319\)](#)
- [Warn user \(./mcp.php?i=warn&mode=warn_post&f=31&p=1319&sid=befaf904784df501ab581b011c284453\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1319&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1319\)](#)

Re: Some simple modular block ideas (#p1319)

by **slacker** » Wed Jun 05, 2013 12:13 pm

Nice work, I tried the delay out and it seems to do what it should, like you say the modulated one doesn't sound right in the simulator I'll make a patch using it and try in on real hardware.

I used the smoothing on the delay rather than quantising purely because I like the sound it makes when you adjust the delay time, sounds a bit like an analogue delay.

I don't quite follow what you mean about not being able to add feedback, I've tried it with the

delay block and using mixers before and after the delay block works in the simulator. Here's a quick patch with feedback to show what I've tried, the two EQs are just to provide some gain to the signal so that it will cause multiple repeats.

The idea of having feedback within the block sounds good though.

Attachments

feedback.jpg (94.96 KiB) Viewed 194 times

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1320\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1320\)](#)
- [Report this post \(./report.php?f=31&p=1320\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1320&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1320\)](#)

Re: Some simple modular block ideas (#p1320)

by **Digital Larry** » Wed Jun 05, 2013 12:22 pm

Hah, maybe I was wrong then! It takes one willing to break the rules. Good work.

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1321\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1321\)](#)
- [Report this post \(./report.php?f=31&p=1321\)](#)
- [Warn user \(./mcp.php?i=warn&mode=warn_post&f=31&p=1321&sid=befaf904784df501ab581b011c284453\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1321&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1321\)](#)

Re: Some simple modular block ideas (#p1321)

by **slacker** » Wed Jun 05, 2013 12:44 pm

Probably helps if you don't know what the rules are 😊

It doesn't seem to like it if you take the output straight from the delay block, sometimes when I started the simulator it worked other times there was no feedback. Seems to work consistently using a mixer after the delay block.

Replacing the EQs with the overdrive block works better and allows control over the number of repeats.

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1322\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1322\)](#)
- [Report this post \(./report.php?f=31&p=1322\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1322&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1322\)](#)

Re: Some simple modular block ideas (#p1322)

by **Digital Larry** » Wed Jun 05, 2013 4:24 pm

My general concept is that the inputs (audio data) of each block are assumed to be calculated for

the current cycle by the time you use them. When you write a full block in ASM, you calculate all the results from the previous cycle first, then finally you add the input and then shift whatever stored values into place for the next time around.

Making a loop in the connection breaks this assumption because audio data will actually have to go through the following blocks in order to be available as an input in an earlier block. Not to say that delaying some data by 1 sample time wouldn't work for some cases, and a delay line would be a good example of that. But it may cause some unintended results. I can't really explain the results you've seen and I'm not sure I'm going to try!

My preference would be to understand what options you would like in a feedback loop (as I understand that putting filters into the feedback path can be fun) and try to design that into a block from the beginning. May require some additional thought (and I do have to prepare myself for these mega thinking marathons). 😊

Top

- [Edit post](#) (./posting.php?mode=edit&f=31&p=1323)
- [Delete post](#) (./posting.php?mode=delete&f=31&p=1323)
- [Report this post](#) (./report.php?f=31&p=1323)
- [Information](#) (./mcp.php?i=main&mode=post_details&f=31&p=1323&sid=befaf904784df501ab581b011c284453)
- [Reply with quote](#) (./posting.php?mode=quote&f=31&p=1323)

Re: Some simple modular block ideas (#p1323)

by **Digital Larry** » Thu Jun 06, 2013 7:51 am

The input gain of the 2-1 mixer block in build 367 was limited to 0.5. There's also a bug where adjusting slider 2 sets input 1's level, not input 2. I've fixed that issue and also made the maximum gain for each input 1.0. This does open up the possibility of signal levels getting too hot, so please mix responsibly. 😊

I also noticed that your feedback path takes additional dry input.

I'll probably release this in a few days after I fix or create some additional bugs. If you have another block you'd like me to toss in, let me know.

Thanks!

Top

- [Edit post](#) (./posting.php?mode=edit&f=31&p=1324)
- [Delete post](#) (./posting.php?mode=delete&f=31&p=1324)
- [Report this post](#) (./report.php?f=31&p=1324)
- [Warn user](#) (./mcp.php?i=warn&mode=warn_post&f=31&p=1324&sid=befaf904784df501ab581b011c284453)
- [Information](#) (./mcp.php?i=main&mode=post_details&f=31&p=1324&sid=befaf904784df501ab581b011c284453)
- [Reply with quote](#) (./posting.php?mode=quote&f=31&p=1324)

Re: Some simple modular block ideas (#p1324)

by **slacker** » Thu Jun 06, 2013 9:25 am

Digital Larry wrote:

There's also a bug where adjusting slider 2 sets input 1's level, not input 2.

Yes I'd noticed that, I was going to raise it as an issue once I'd figured out if it was just a control panel error or whether it actually did that.

I also noticed that your feedback path takes additional dry input.

Yeah I just chucked that in there so I could hear the dry signal, you wouldn't normally want that in a delay, you'd only take feedback from the delayed signal.

I'll try and do a few more blocks, I'll have a read up on your design philosophy first try and make ones that fit in with it.

I understand what you mean about feedback breaking your algorithm now. I've looked at the output produced by SpinCad whilst repeatedly running the simulator and it doesn't always put the blocks in the same order, occasionally it happens to put them in the order that matches what you've drawn and you get the effect you expected. I must have just been lucky before and made something that worked more often than it didn't.

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1490\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1490\)](#)
- [Report this post \(./report.php?f=31&p=1490\)](#)
- [Warn user \(./mcp.php?i=warn&mode=warn_post&f=31&p=1490&sid=befaf904784df501ab581b011c284453\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1490&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1490\)](#)

Re: Some simple modular block ideas (#p1490)

by **disasterarea** » Wed Mar 19, 2014 5:48 am

How about a block that feeds a constant into a control? There are a lot of existing blocks that require some kind of control input (chorus, delay, etc) and if you want to use one without a user-facing control there's no way to tell that control input what to do. Can we use the SOF block for that? As in input 0, output 1?

Also, this is probably a question for another part of the forum but how do I create a depth-and-width control for an LFO? Normally I would use one pot to change the LFO rate and the other one to MULX the output with the pot value. The standard chorus block has rate and depth controls (although the depth seems really sensitive) but I'd almost prefer to control it myself. A chorus really only has one parameter, which is the delay time. The delay time is modulated with a carrier whose frequency is the chorus rate and whose amplitude is the chorus depth. Additionally, the amplitude can have an offset that sets the maximum and minimum delay times - that's the "manual" parameter on a flanger. I'd love to see Slacker's mod delay block implemented, along with a way to tweak the LFO output.

While we're at it, how about a block that implements an oscillator without using one of the chorus

sin / cos or ramp LFOs? There's an example in the "Battling LFOs" program on the Spin site.

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1491\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1491\)](#)
- [Report this post \(./report.php?f=31&p=1491\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1491&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1491\)](#)

Re: Some simple modular block ideas (#p1491)

by **Digital Larry** » Wed Mar 19, 2014 8:23 am

Hello disasterarea and welcome to the forum!

What I have done with some of the blocks with control inputs is this: If there is no input connected, then a default value is used instead. The details of what that value is, is hidden inside the Java code presently. The most obvious example of this is in the 2-1 mixer. With no controls connected, the gain on each side is set to 0.5 (I think). Or the default value can be set using the control panel via right-click (though not all blocks have control panels).

Your other question is a bit more complicated. The chorus presets I've turned into blocks were mostly just taken from the Spin Forum or Knowledge Base/supplied code. I don't think the current chorus patch is very flexible.

If you've spent much time looking at the supplied patch examples from Spin, which it seems you have, then you can tell that many of them are really fine tuned in terms of what the controls do in a given patch. Off the top of my head (as I have not been spending a lot of time on SpinCAD lately), you could start with a simple delay line with modulation input and then do all the control processing with Scale/Offset, Gain, and other control manipulation blocks, along with a mixer control at the output to simultaneously control the blend. Or, you could simply create a new block which was a chorus whose control inputs worked in such and such a way (though that seems a little antithetical to SpinCAD's concept). You've really hit on one of my biggest conceptual challenges! 🍷

I'm pretty sure that Slacker's delay is implemented - it's called "Straight Delay" or "Mod Delay" or something. It needs an adjustment capability for the max delay time as currently it's quite long, and not really suitable for chorusing.

Tell you what - I get motivation from interest in the program. Keep posting and it'll help me stop programming my Raspberry Pi and back to the FV-1. It will also help to make a new topic for each request as I find it easier to deal with things one at a time.

Thanks again for the input.

DL

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1492\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1492\)](#)
- [Report this post \(./report.php?f=31&p=1492\)](#)
- [Warn user \(./mcp.php?i=warn&mode=warn_post&f=31&p=1492&sid=befaf904784df501ab581b011c284453\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1492&sid=befaf904784df501ab581b011c284453\)](#)

[sid=befaf904784df501ab581b011c284453\)](#)

- [Reply with quote \(./posting.php?mode=quote&f=31&p=1492\)](#)

Re: Some simple modular block ideas (#p1492)

by **disasterarea** » Wed Mar 19, 2014 1:35 pm

Here's a question - does the Volume block just implement a straight MULX with the control input? Can we use it to process control signals? I'm assuming it just goes zero to one, like the POT values, so in theory it could be used to allow control over the amplitude of control sources much like the control parameters for the LFO.

I think one nice way to do the max delay time would just be to let us choose it in the right-click menu on the delay block. Honestly, a simple delay block with a max time that we can set and an input for the time is all you need for 90% of delay stuff.

Regarding feedback in a delay line (or even in a reverb algorithm) the signal from THIS pass needs to be stored in order to use it in the NEXT pass. You can stash the value in a register or leave them in the accumulator (but that's nasty.) I think a "carryover" or "storage" block would be really useful. Let us place it in the signal path somewhere and it stores a value to use for later. Then you have a "retrieval" block that pulls from the storage block to use on the current pass. This would let you have feedback anywhere in the signal path, not just immediately around the delay. The code would be trivial, I think, just write the ACC to the storage and then pass it on, or read it from storage and do something with it.

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1493\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1493\)](#)
- [Report this post \(./report.php?f=31&p=1493\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1493&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1493\)](#)

Re: Some simple modular block ideas (#p1493)

by **Digital Larry** » Wed Mar 19, 2014 3:24 pm

disasterarea wrote:

Regarding feedback in a delay line (or even in a reverb algorithm) the signal from THIS pass needs to be stored in order to use it in the NEXT pass. You can stash the value in a register or leave them in the accumulator (but that's nasty.) I think a "carryover" or "storage" block would be really useful. Let us place it in the signal path somewhere and it stores a value to use for later. Then you have a "retrieval" block that pulls from the storage block to use on the current pass. This would let you have feedback anywhere in the signal path, not just immediately around the delay. The code would be trivial, I think, just write the ACC to the storage and then pass it on, or read it from storage and do something with it.

This is another case where the current structure of the software isn't as flexible or optimized as it might be.

Each block reserves a register to save its output value. An input connected to an output figures out

what the output's register number is and then reads from that register. I'm also pretty sure that I expect the ACC to be zero at the end of each block. So there is some inefficiency, for example if in Spin ASM you can keep a relevant value in the accumulator so that it can flow through to the next block, SpinCAD still wastes a register to hold it. The way I'd choose to do it would be simply to incorporate the feedback right into the block design itself, and if feedback is not used, don't bother to implement those instructions. This is similar to the way the 6 band EQ works. When you set a band to zero, those instructions drop out. The decision is made in the Java code that builds the EQ block based on the values from the control panel.

I'll answer your other question later when I have a chance to look at the code.

Thanks,

DL

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1495\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1495\)](#)
- [Report this post \(./report.php?f=31&p=1495\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1495&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1495\)](#)

Re: Some simple modular block ideas (#p1495)

by **Digital Larry** » Wed Mar 19, 2014 8:05 pm

OK I started a new topic in this subforum on "The Volume Block". Check it out.

Back to whether you could have an external block be a "feedback" signal holder, that will not work reliably the way the program is currently written. When the code is generated, there is a routine that gets all the blocks in order from first to last. If you have a loop in the topology, this routine can get confused. Sometimes it works and sometimes it doesn't. So that more than anything else is why I recommend that feedback be implemented within a block rather than outside it.

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1498\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1498\)](#)
- [Report this post \(./report.php?f=31&p=1498\)](#)
- [Warn user \(./mcp.php?i=warn&mode=warn_post&f=31&p=1498&sid=befaf904784df501ab581b011c284453\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1498&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1498\)](#)

Re: Some simple modular block ideas (#p1498)

by **disasterarea** » Thu Mar 20, 2014 11:45 am

Digital Larry wrote:

OK I started a new topic in this subforum on "The Volume Block". Check it out.

Back to whether you could have an external block be a "feedback" signal holder, that will not work reliably the way the program is currently written. When the code is generated,

there is a routine that gets all the blocks in order from first to last. If you have a loop in the topology, this routine can get confused. Sometimes it works and sometimes it doesn't. So that more than anything else is why I recommend that feedback be implemented within a block rather than outside it.

That's exactly what I was hoping for with regards to the Volume Block.

I was thinking of making two blocks for feedback - Feedback Send and Feedback Return. Feedback Send would just put any value passed in a register. It would probably need a CLR at boot to make sure it was clean before use. Feedback Return would go earlier in the chain, and it would just be the same register number as the Send. Its only job would be to pass the last "sent" value somewhere else in the program. There wouldn't be any loops within the program, just a place to store a register and retrieve it later.

We might need the ability to make more than one pair of send / return, so a slider that would let you pair them up would be good. Send 1 / Return 1, Send 2 / Return 2, etc.

I don't know how you're generating ASM from these blocks or how you're making the blocks, even. But what I'm thinking of is literally just a register that we stuff before program end, and then read back again & clear on the next pass. Stuff again, end, read again, clear.

You could SOF it to invert its polarity for negative feedback, or to amplify it. Feedback in a filter makes it deeper / higher "q." Lots of applications for a feedback setup.

Top

- [Edit post \(./posting.php?mode=edit&f=31&p=1500\)](#)
- [Delete post \(./posting.php?mode=delete&f=31&p=1500\)](#)
- [Report this post \(./report.php?f=31&p=1500\)](#)
- [Information \(./mcp.php?i=main&mode=post_details&f=31&p=1500&sid=befaf904784df501ab581b011c284453\)](#)
- [Reply with quote \(./posting.php?mode=quote&f=31&p=1500\)](#)

Re: Some simple modular block ideas (#p1500)

by **Digital Larry** » Thu Mar 20, 2014 12:51 pm

disasterarea wrote:

I was thinking of making two blocks for feedback - Feedback Send and Feedback Return. Feedback Send would just put any value passed in a register. It would probably need a CLR at boot to make sure it was clean before use. Feedback Return would go earlier in the chain, and it would just be the same register number as the Send. Its only job would be to pass the last "sent" value somewhere else in the program. There wouldn't be any loops within the program, just a place to store a register and retrieve it later.

Now THAT is an interesting idea. 😊

Top

Display posts from previous: All posts Sort by Post time Ascending Go

Post a reply

16 posts • Page **1** of **1**

[Return to SpinCADBlocks](#)

Jump to:	<input type="text" value="SpinCADBlocks"/>	<input type="button" value="Go"/>
Quick-mod tools:	<input type="button" value="Lock topic"/>	<input type="button" value="Go"/>

Who is online

Users browsing this forum: **Digital Larry** and 0 guests

Powered by phpBB® Forum Software © phpBB Group

Administration Control Panel