

EclaireXL - Feature #84

Add SID

11/09/2019 09:56 AM - foft

Status:	New	Start date:	11/09/2019
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0:00 hour
Target version:			
Description			
Agreed to try adding a SID to the pokeymax.			
Eclaire is a good test bed for this, so starting out with that			

History

#1 - 11/09/2019 09:56 AM - foft

- Tracker changed from Bug to Feature

#2 - 11/09/2019 09:56 AM - foft

Have a dual 8250 working, though it doesn't fit in the smaller boards. Investigating why, memory or logic...

#3 - 11/09/2019 10:00 AM - foft

Logic utilization (in ALMs) 9,808 / 9,430 (104 %)

#4 - 11/09/2019 10:03 AM - foft

600 alms each, will use mono sid I guess. Or make it a generic and enable on the larger boards?

#5 - 11/09/2019 10:05 AM - foft

Also got this potential USB sharing, that'd save 750 ALMS. Each request/reply is host initiated, so no reason can't share one. Keepalives would need to go to both ports. Not sure how complex that'd be! Perhaps I plumb the output to both, with a mux for when I send explicit in/out requests?

#6 - 11/12/2019 10:02 PM - foft

Actually the issue with fitting in v2 is block ram, not ALMs. Though its always good to save ALMs!

The sid tables take 14KB each. So for stereo sid its 28KB. Anyway I do not even have 14KB to spare on v2...

#7 - 11/12/2019 10:04 PM - foft

I loaded the tables in to octave to take a look.

Observations...

- Not a simple curve
- Lots of zeros, >80% in some cases
- Quite a lot of dupes

dupes:

```
log2(numel(unique(ps)))
```

ans = 6.5699

```
log2(numel(unique(st)))
```

ans = 4.7549

```
log2(numel(unique(p_t)))
```

ans = 5.8074

```
log2(numel(unique(pst)))
```

ans = 4.5850

zeros:

```
sum(ps==0)/numel(ps)
```

ans = 0.63379

```
sum(st==0)/numel(st)
```

ans = 0.86743

```
sum(p_t==0)/numel(p_t)
```

ans = 0.58545

```
sum(pst==0)/numel(pst)
```

ans = 0.89331

#8 - 11/12/2019 10:18 PM - foft

Wondering about some kind of bit map of unpopulated locations (reducing locations) or a 2nd lookup layer (reducing bits)

#9 - 11/12/2019 10:19 PM - foft

Also wondering if the clk frequency is important (not enable), I see some state table increment on every clock, which is reset on enable

#10 - 11/12/2019 10:19 PM - foft

It sounds good to me, but not sure what it is meant to sound like!!

#11 - 11/17/2019 09:07 PM - foft

Did some research over the weekend. Generating these isn't a terribly trivial problem!

I tried for some time to generate in matlab. I came up with a pretty good solution for wave 3: triangle + sawtooth. Actually its sawtooth + 2 sawtooth since the triangle is inverted. Anyway this worked pretty well:

```
tmp = zeros(size(a));
for k=1:12
    pre =k-4;
    post = k;
    if pre<1
        pre = 1;
    end
    if post>12
        post = 12;
    end
    idx = pre:post;
    if a(idx) == b(idx);
        tmp(k) = b(k);
    end
end
```

i.e. and and if in conflict, ground the 4 adjacent bits.

Was struggling with the pulse + triangle one, now come up with an easy answer for that one yet!

Anyway found some description of what is going on in the libsidplayfp + some functions to generate:

libsidplayfp-2.0.1/src/builders/residfp-builder/residfp/WaveformCalculator.cpp

Also the wave files are here:
<https://bel.fi/alankila/c64-sw/combined-waveforms/sidwaves/>

#12 - 11/17/2019 09:10 PM - foft
- File sidbits.tar.bz2 added

Attaching some of the octave files from investigating this for reference + the wave files + other useful bits I found online.

#13 - 01/14/2021 10:35 AM - foft
I have implemented my own better sid core now which I will back-port to Eclairé.

Files			
sidbits.tar.bz2	148 KB	11/17/2019	foft