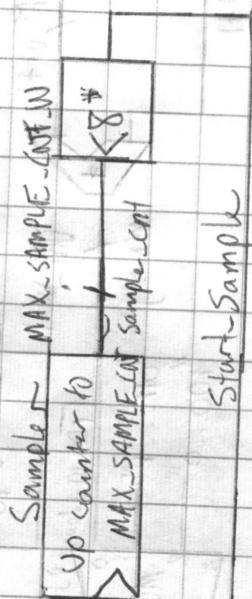


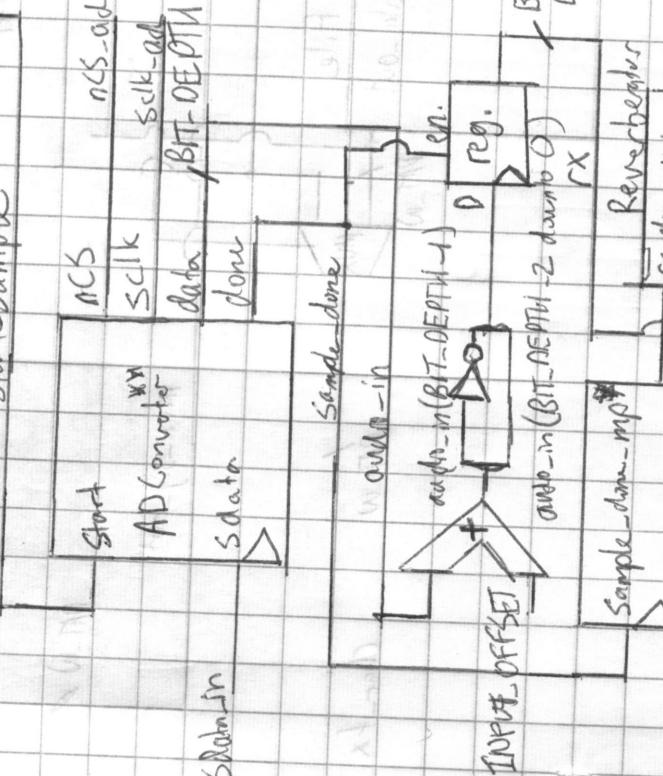
F.12.2: FPCA Top level

Notes:

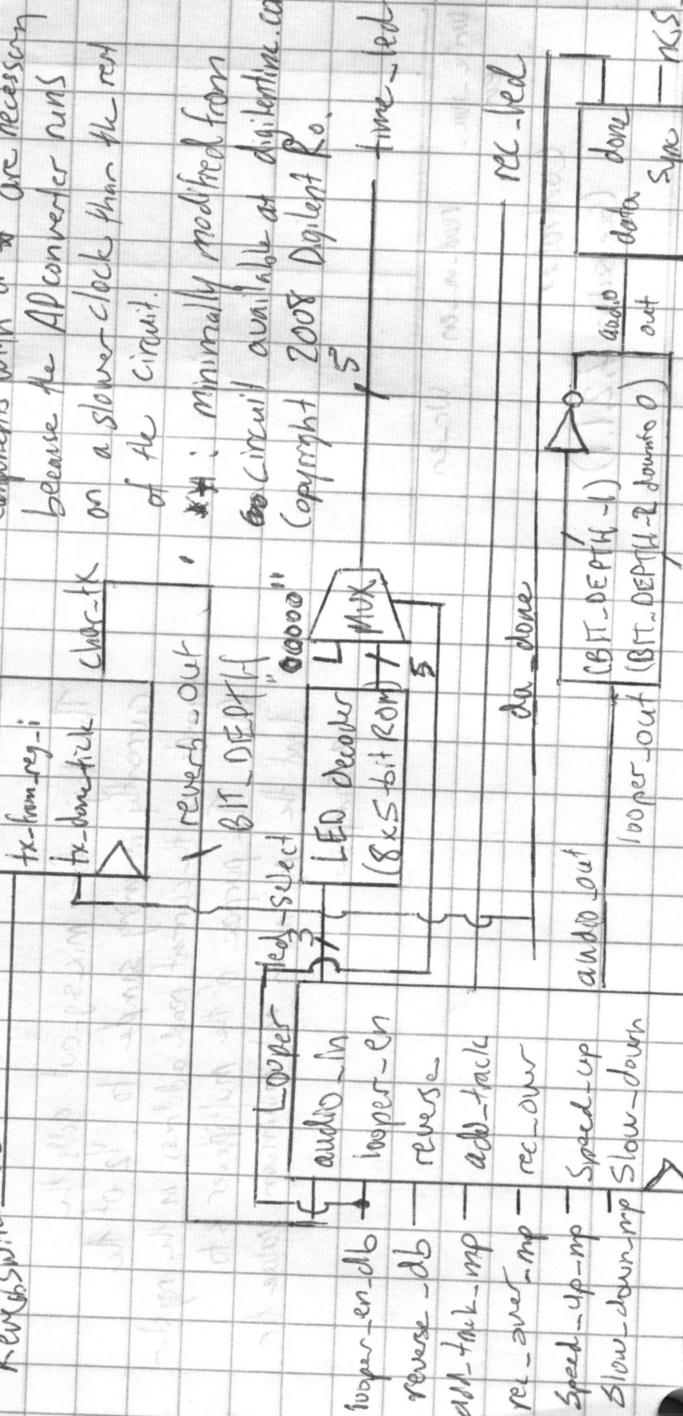
- Switches "Sound SW" "Reverb Switch", "looper on" and "reverse" pass through a debouncer (not shown) indicated by an appended " - db".
- Buttons "Add track", "re-aver", "speed up" and "slow down" each pass through a debouncer and a monopole (not shown), indicated by an appended " - m".



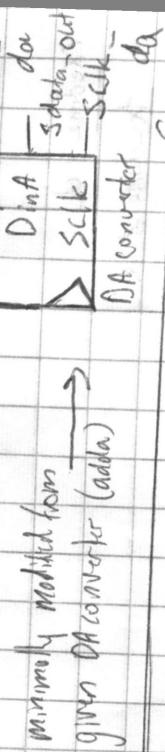
Start-Sample



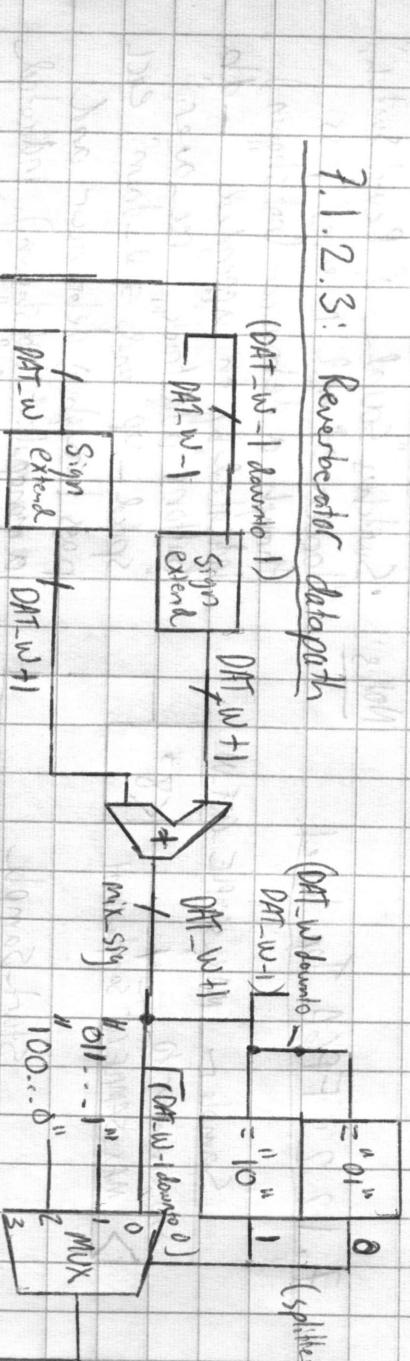
四百一十一



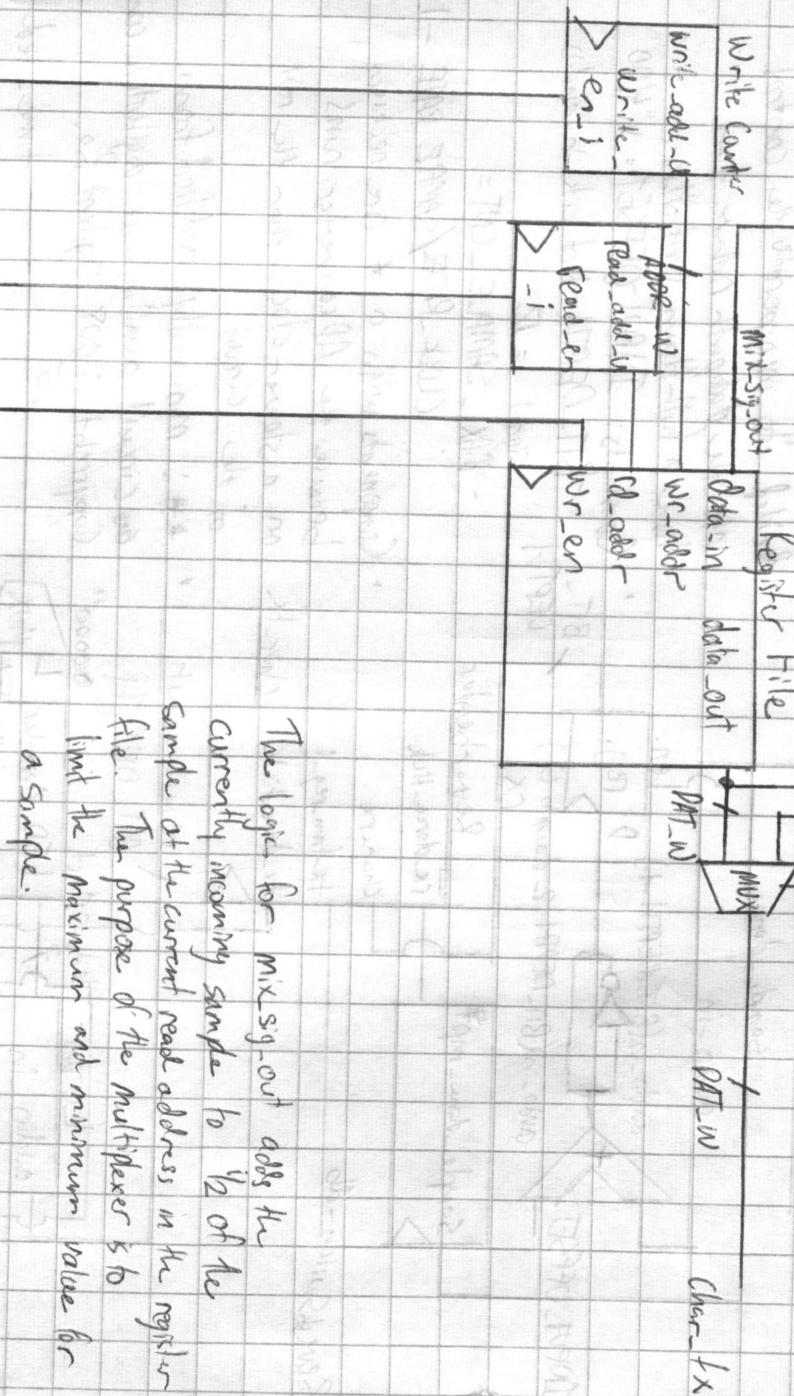
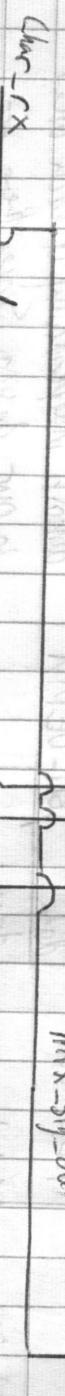
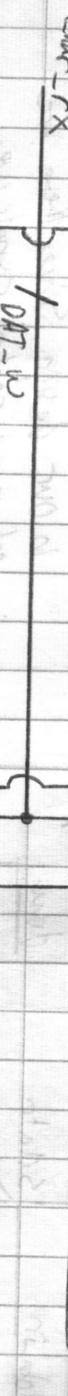
11



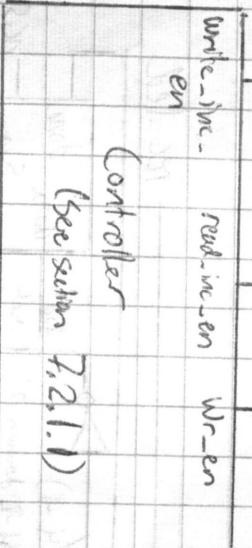
7.1.2.3: Reverbator datapath



Char-RX
Mix-Sig-out



The logic for Mix-Sig-out adds the currently incoming sample to y_2 of the sample at the current read address in the register file. The purpose of the multiplexer is to limit the maximum and minimum value for a sample.



Controller
(See section 7.2.1.1)

X-done-hick
tx-done-hick

2.2.1.1 : Reverbator FSM

$\text{tx_done_tick}!$

$\text{if } \text{tx_done_tick} = '1'$
 $\text{read_inc_en} = '1'$



tx_done_tick
 $\text{if } \text{tx_done_tick} = '1'$
 tx_force_tick



$\text{wr_en} \Leftarrow '1'$
 $\text{if } \text{tx_ready} = '1'$
 then
 $\text{tx_start} \Leftarrow '1'$

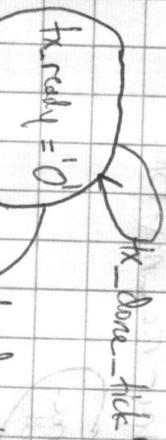


$\text{if } \text{tx_done_tick} = '1'$
 then

$\text{write_inc_en} \Leftarrow '1'$

$\text{read_inc_en} \Leftarrow '1'$

Helper tx Ready FSM:



tx_start

$\text{tx_ready} = '0'$

tx_done_tick

$\text{tx_start}^{'}$

7.2.1.2: Looper FSM



```

if sample_t = '1'
if sample_en = '1'
  if sample_en_clr = '1'
    end if;
    if looper_en = '0' then
      addr_reset <= '1'
    end if;
  end if;
end if;
  
```

Signals outside the FSM:

- Out_combine <= '0' when looper_en' or recording else '1'.
- rec_lead <= '1' when recording or adding else '0'.

* add-next_n, rec-next_n,

adding_n and recording_n update the values of their corresponding flip-flops (without the '_n') on each clock rise. However, the rec_over and add-back buttons override this, setting rec_next and add_next to 1, respectively.

```

if add_tc = '1' then
  Recording_n <= '0'
  Adding_n <= '0'
  if rec_next = '1' then
    Recording_n <= '1'
    Rec_next_n <= '1'
  else if add_next = '1' then
    Adding_n <= '1'
    Add_next_n <= '1'
  end if;
end if;
  
```

7. 1. 2. 4: Looper datopath

