FDR: Logic changes & simulations

FDR Simulations

Dialogue

This document summarises any relevant simulation or design changes for the FDR; and is primarily formed by outstanding actions from the two IDR sessions.

Row Controller Modifications

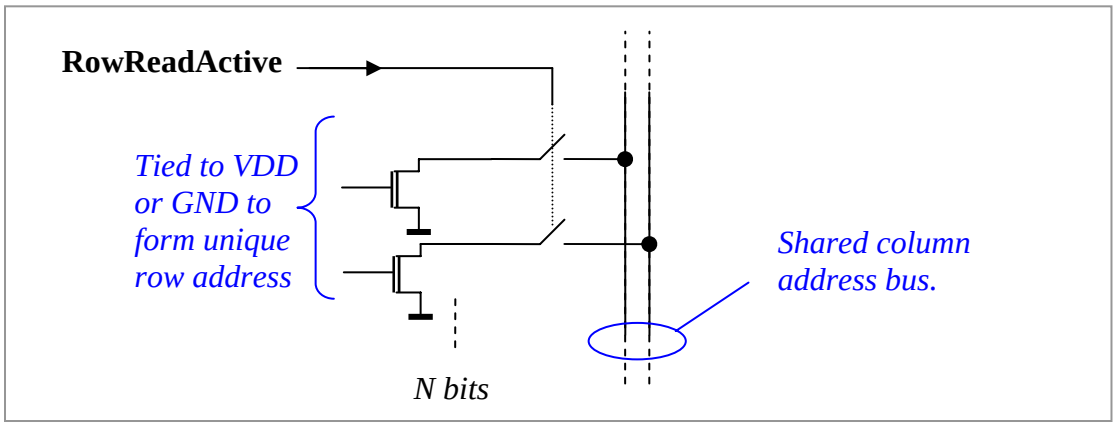
During layout a number of modifications were made to the row control logic schematics. These are listed below:

a) Latch hold circuit updated to include a "safe-mode" circuit.
b) In the mux6 cell, the spare NOR gate was removed, wiring OVERRIDE directly into the NAND to save space. (this inverts the polarity of the override signal).
c) The bidirectional SRAM cells were modified to use the 3 phase clocking scheme rather than two as requested at IDR.
d) Clock buffers added for phi3
e) In the mux6x7 schematic an error was corrected: The pull-down for DataCodeB had incorrectly been wired after the buffer, not before. (The DataCode signals a left tristate by the mix_6 block unless there is valid data.)
f) An extra buffer for the DataValid signal was added to drive local signals in the mux_6x7 block to reduce the load on the main DataValid output
g) New bit ordering in the 20-input or gate (schematic sram_bifir_shift20.5). For efficient layout it was necessary to change the bit order; provided D<19> is present in the output or6 gate the order of the other bits has no effect on functionality.
h) The shift-in transmission gate (schematic sram_bifir_shift20.5) was removed, and a special input-version of the sram_bidir_shift cell was created to perform the same operation – this helped particularly in achieving a compact layout.
i) A new row encoder to span 168 unique addresses was created as requested in the IDR.
j) Timecode distribution was adjujsted to suit isolated columns of logic rather than "back-to-back". Original design had one timecode buffer driving two SRAM banks on two "back-to-back" rows. New design has one timecode buffer driving four SRAM banks on four rows.
k) Control signal buffering modified so that each column re-buffers independently from the next column across (before, just the left hand version of each control signal was used to re-generate the onward buffered signal for a pair of columns, but running these control signals across 2mm to the neighbour column {parallel to bias lines} seemed unnecessary.)
l) Additional current-limiting transistors were added to the NOR gate in the monostable circuit (same bias lines as current limited inverter).
m) Power-on-reset for monostable circuits will be driven from off chip. Pull-down resistor should be added to the circuit board to ensure correct operation on power-up. (the pull-down pad showed odd behaviour in simulation so will not be used)
n) The preSample pixel schematic was modified such that external resets are separated for sample and preamp reset (originally shared as one Rst_Ext signal). This is necessary to correctly reset the pixel from off-chip, ie before the start of a bunch train.
o) "Slice" level buffers for reset signals are now powered by digital, not pixel power supply (unnecessary since they drive logic gates in the pixel rather than analog circuits directly).

p) The bus ordering for SRAM write was changed to facilitate layout; Time-code now occupies the more significant end of the binary word; Data-code located in the LSBs.
q) Configuration “slow” clocks are driven independently along each row, rather than the one-between-four scheme presented at IDR.

Logic Simulation: New Row Encoder

The basic structure of the row encoder is illustrated below. The function of this block is to report at the column base the row address of the cell that is currently being read.



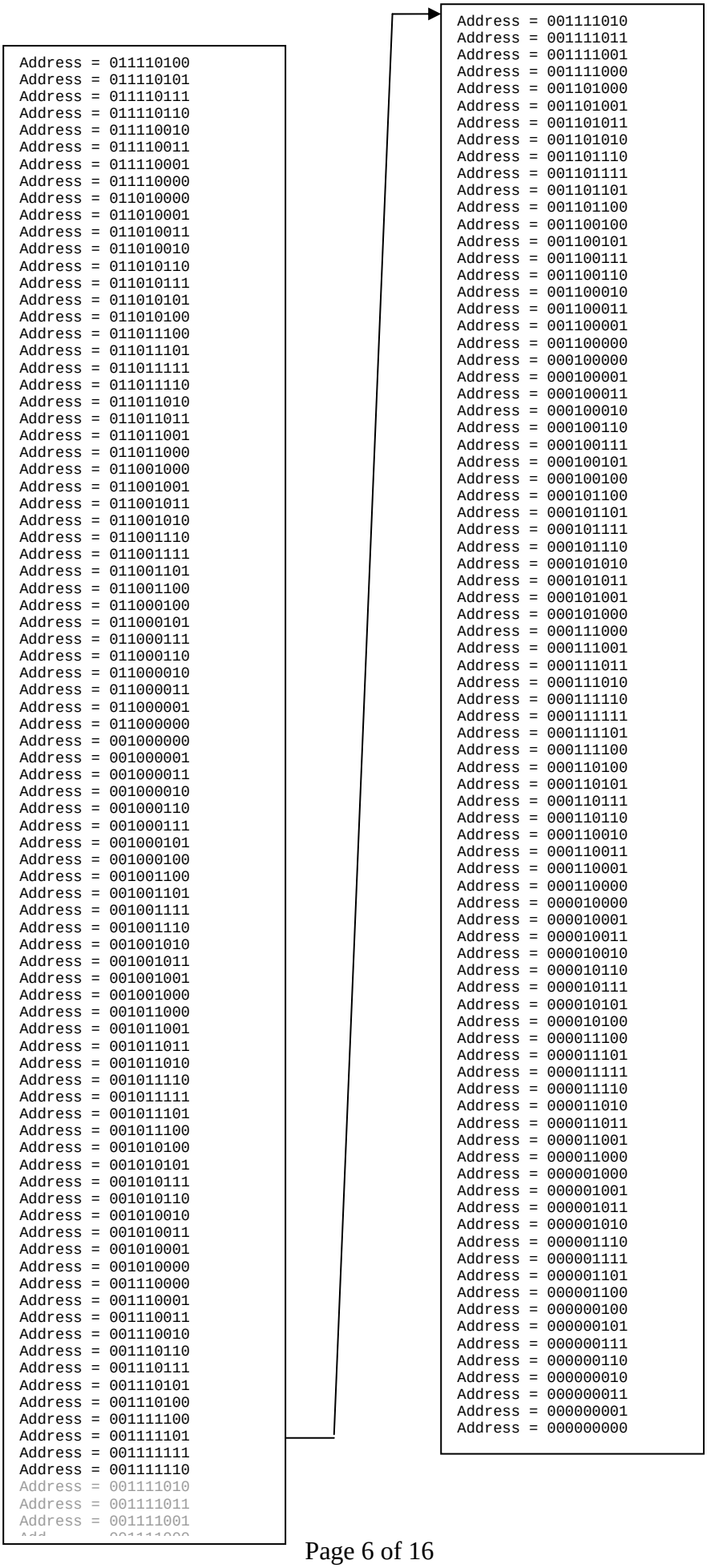
The active cell drives the column bus with its own unique code. The pull-to-ground scheme is compatible with the sense amplifiers used to read the SRAM cells.

Simulations

A row-encoder of 168 cells is simulated.

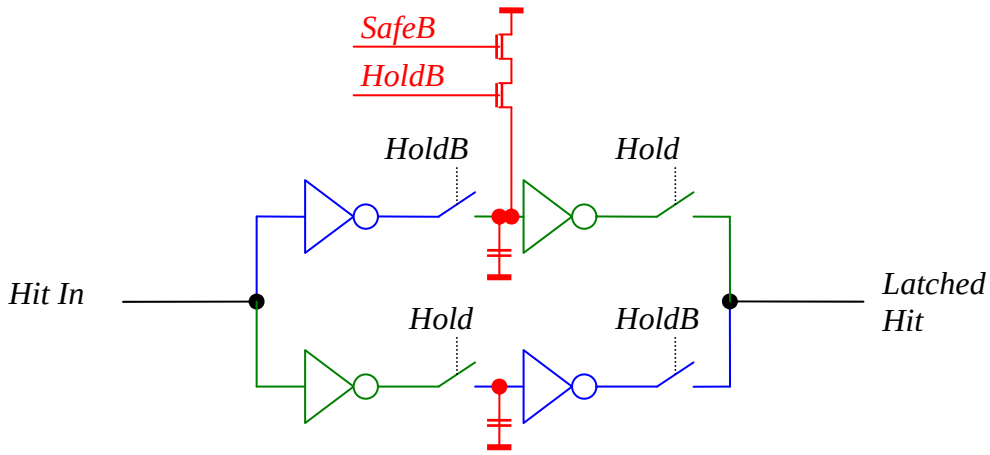
A new Verilog stimulus is used to individually excite each of the 138 enable (RowReadActive) signals and print to screen the address code that is seen at the base of the column. This text file is imported into excel and processed to check that no repeated codes are seen in the 84-code set.

Below: Mixed-mode simulation output log excerpt.


Address = 011110100
Address = 011110101
Address = 011110111
Address = 011110110
Address = 011110010
Address = 011110011
Address = 011110001
Address = 011110000
Address = 011010000
Address = 011010001
Address = 011010011
Address = 011010010
Address = 011010110
Address = 011010111
Address = 011010101
Address = 011010100
Address = 011011100
Address = 011011101
Address = 011011111
Address = 011011110
Address = 011011010
Address = 011011011
Address = 011011001
Address = 011011000
Address = 011001000
Address = 011001001
Address = 011001011
Address = 011001010
Address = 011001110
Address = 011001111
Address = 011001101
Address = 011001100
Address = 011000100
Address = 011000101
Address = 011000111
Address = 011000110
Address = 011000010
Address = 011000011
Address = 011000001
Address = 011000000
Address = 001000000
Address = 001000001
Address = 001000011
Address = 001000010
Address = 001000110
Address = 001000111
Address = 001000101
Address = 001000100
Address = 001001100
Address = 001001101
Address = 001001111
Address = 001001110
Address = 001001010
Address = 001001011
Address = 001001001
Address = 001001000
Address = 001011000
Address = 001011001
Address = 001011011
Address = 001011010
Address = 001011110
Address = 001011111
Address = 001011101
Address = 001011100
Address = 001010100
Address = 001010101
Address = 001010111
Address = 001010110
Address = 001010010
Address = 001010011
Address = 001010001
Address = 001010000
Address = 001110000
Address = 001110001
Address = 001110011
Address = 001110010
Address = 001110110
Address = 001110111
Address = 001110101
Address = 001110100
Address = 001111100
Address = 001111101
Address = 001111111
Address = 001111110
Address = 001111010
Address = 001111011
Address = 001111001
Address = 001111000
Address = 001101000
Address = 001101001
Address = 001101011
Address = 001101010
Address = 001101110
Address = 001101111
Address = 001101101
Address = 001101100
Address = 001100100
Address = 001100101
Address = 001100111
Address = 001100110
Address = 001100010
Address = 001100011
Address = 001100001
Address = 001100000
Address = 000100000
Address = 000100001
Address = 000100011
Address = 000100010
Address = 000100110
Address = 000100111
Address = 000100101
Address = 000100100
Address = 000101100
Address = 000101101
Address = 000101111
Address = 000101110
Address = 000101010
Address = 000101011
Address = 000101001
Address = 000101000
Address = 000111000
Address = 000111001
Address = 000111011
Address = 000111010
Address = 000111110
Address = 000111111
Address = 000111101
Address = 000111100
Address = 000110100
Address = 000110101
Address = 000110111
Address = 000110110
Address = 000110010
Address = 000110011
Address = 000110001
Address = 000110000
Address = 000010000
Address = 000010001
Address = 000010011
Address = 000010010
Address = 000010110
Address = 000010111
Address = 000010101
Address = 000010100
Address = 000011100
Address = 000011101
Address = 000011111
Address = 000011110
Address = 000011010
Address = 000011011
Address = 000011001
Address = 000011000
Address = 000001000
Address = 000001001
Address = 000001011
Address = 000001010
Address = 000001110
Address = 000001111
Address = 000001101
Address = 000001100
Address = 000000100
Address = 000000101
Address = 000000111
Address = 000000110
Address = 000000010
Address = 000000011
Address = 000000001
Address = 000000000

Logic simulation: New Latch-Hold circuits

The structure below uses a ping-pong architecture to sample the hit signals on both rising and falling edges of the Hold signal. This means the sampling "Hold" signal can run at the ~6Mhz. Additional circuits are shown in RED: This new addition ensures that the circuit can be held in a safe (zero current) state when not in use and does not draw extra current if driven incorrectly.

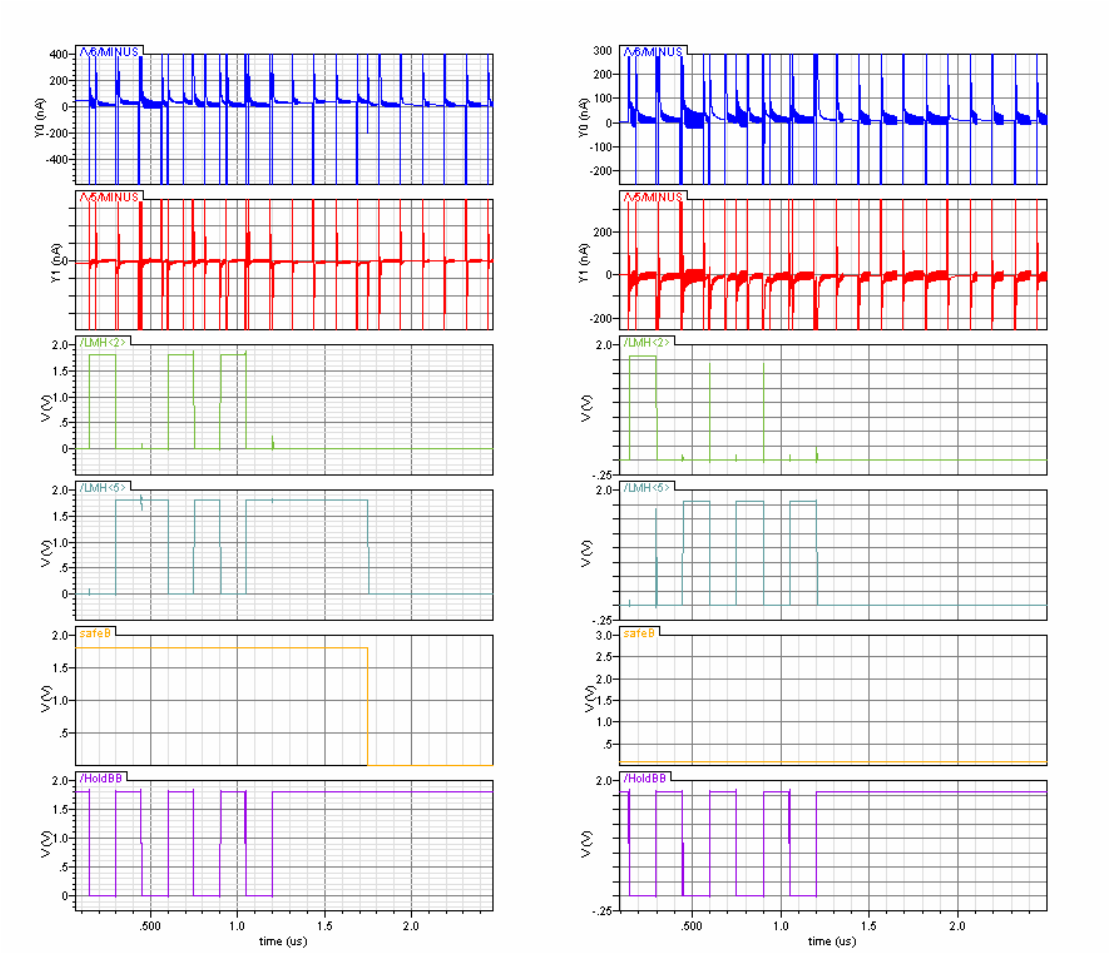


The "hit" states are stored on the gate capacitance of the second inverters. This capacitance is small, but the state must be held for only 150ns. Additional capacitance is added to the storage nodes: These help reduce charge injection and will also improve data hold times at higher temperatures.

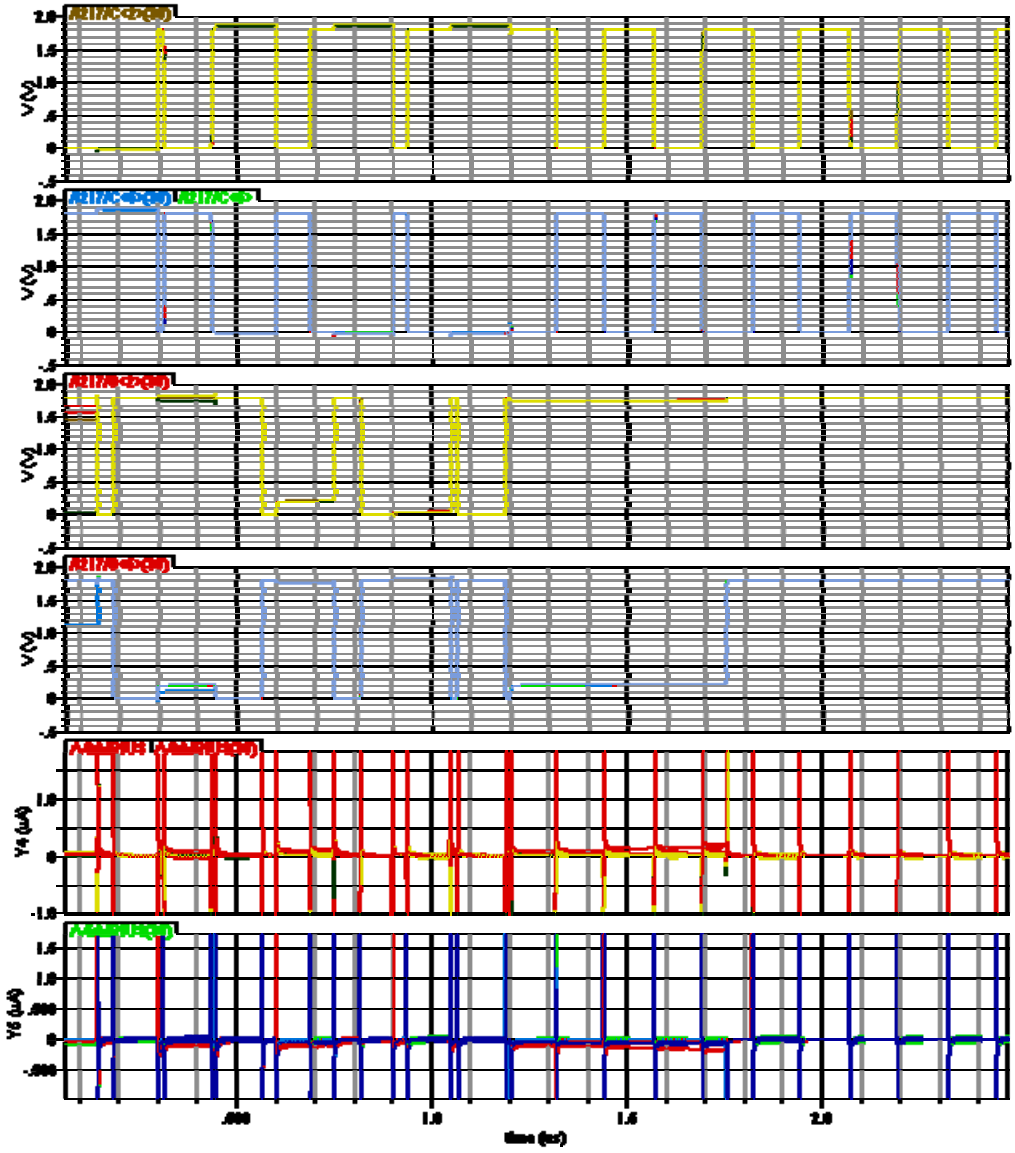
Circuit States

Hold	**HoldB**	**SafeB**	**Valid / Correct operation?**	**Static current flow?**
0	1	0	✗ Functionality incorrect	0 ✓
0	1	1	✓ Latched	0 ✓
1	0	0	✓ Safe power-down state	0 ✓
1	0	1	✓ Latched	0 ✓

Results Waveforms



Above: Typical operation is demonstrated in correct and incorrect modes of operation. When the safeB signal is active at the same time as the hold signal the circuit operation is compromised, but the high current flow due to conflict is inhibited (as seen on the right).



Above: Simulation in all process corners is shown to monitor the critical storage nodes. No excessive current flow conditions are seen.

Mixed-Mode Simulation: Master Controller

The final schematic for the master controller (that has been used for LVS) is simulated again with the original Verilog stimulus. The following modifications were necessary due to design changes:

a) Polarity of OVERRIDE inverted (now active low)
b) Hold signal now toggles every 150ns rather than fast pulses every 150ns
c) Third clock signal specified for bidir-sram cell

Simulation log

Colour highlighting is used to show that the correct hits are reported during readout.

Bank assignment codes =	100	101	111	110	010	011	001

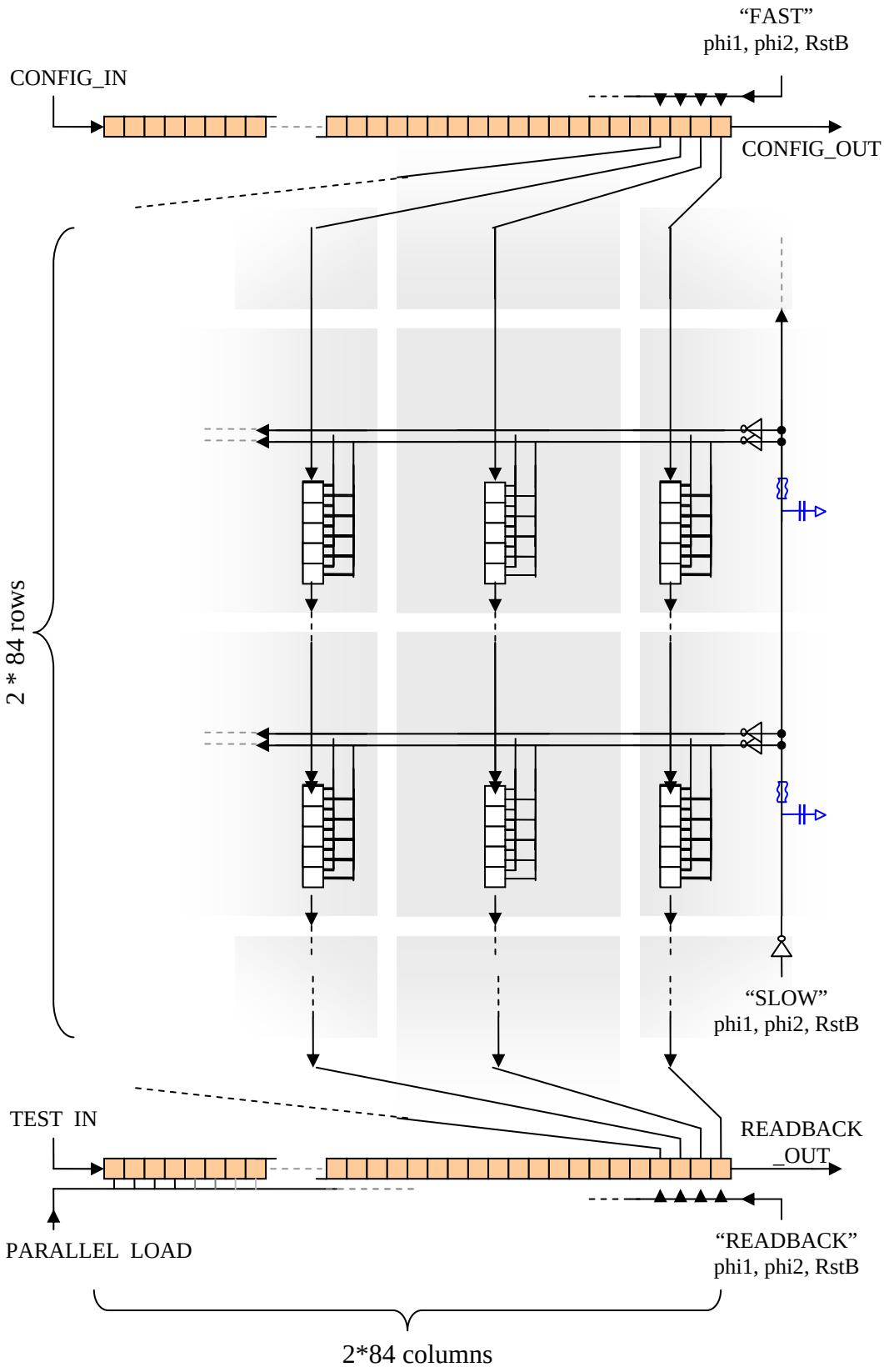
Time stamp

```
Hit pattern   = 1000010000001100110000000000000000000000000 at initialisation
Reset the SR control registers
SR Reset complete
Latch current Hit Pattern:
Hit pattern   = 1000010000001100110000000000000000000000000
Latch current Hit Pattern: Round 2
Hit pattern   = 0001110001100010100010000011000010000001
Latch current Hit Pattern: Round 3 (1 hit)
Hit pattern   = 1000010000000000000000000000010010000000
Readout phase:
SR Initialisation complete
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >> HIT INFO: TIME = 0000000000011
     >>           BANK = 100
     >>           PATT = 100001
     >>           DATA = 011100001111111111111100
     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >> HIT INFO: TIME = 0000000000011
     >>           BANK = 011
     >>           PATT = 010010
     >>           DATA = 100010010111111111111100
     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >> HIT INFO: TIME = 0000000000010
     >>           BANK = 100
     >>           PATT = 000111
     >>           DATA = 011000111111111111111101
     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >> HIT INFO: TIME = 0000000000010
     >>           BANK = 101
     >>           PATT = 000110
     >>           DATA = 010000110111111111111101
     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >> HIT INFO: TIME = 0000000000010
     >>           BANK = 111
     >>           PATT = 000101
     >>           DATA = 000000101111111111111101
     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >> HIT INFO: TIME = 0000000000010
     >>           BANK = 110
     >>           PATT = 000100
     >>           DATA = 001000100111111111111101
     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >> HIT INFO: TIME = 0000000000010
     >>           BANK = 010
     >>           PATT = 000011
     >>           DATA = 101000011111111111111101
     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

```
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >> HIT INFO: TIME = 0000000000010
     >>           BANK = 011
     >>           PATT = 000010
     >>           DATA = 1000000101111111111101
     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >> HIT INFO: TIME = 0000000000010
     >>           BANK = 001
     >>           PATT = 000001
     >>           DATA = 1100000011111111111101
     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >> HIT INFO: TIME = 0000000000001
     >>           BANK = 100
     >>           PATT = 100001
     >>           DATA = 0111000011111111111110
     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
     >> HIT INFO: TIME = 0000000000001
     >>           BANK = 111
     >>           PATT = 110011
     >>           DATA = 0001100111111111111110
     <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
-
-
-
-
-
-
```

no change *

Review: Configuration programming and Verification



Simulation results

Mixed mode verification simulation implements a fast SR, a few lines of pixel SR cells and a fast “READBACK” SR. The pattern shifted out should match that shifted in.

Current status: 95% certainty: Log file (below) shows valid data appearing at readback of top fast shift register. Readback from the bottom fast shift register is currently appearing two cycles out of synchronisation with the expected arrival time. The fact that recognisable data patterns are seen read out from the array is positive, and essentially indicates the programming and readback are functioning correctly. The precise explanation for the loss of synchronisation is currently unknown, but probably a simulation/setup issue, not an inherrant design problem, since valid data is seen at the output.

```
Reset all the shift registers...
Reset complete at                  210.
Fast load Trim0 for row 4
Readback errors ============================           0
   READ BACK 00000000000000000000000000000000000000000000
    EXPECTED 0000000s
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim1 for row 4
Readback errors ============================           0
   READ BACK 10000000000000000000000010000000000000000
    EXPECTED 10000000000000000000000010000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim2 for row 4
Readback errors ============================           0
   READ BACK 10000000000000000000000000000000000000000000
    EXPECTED 10000000000000000000000000000000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim3 for row 4
Readback errors ============================           0
   READ BACK 00000000000000000000000000000000000000000000
    EXPECTED 00000000000000000000000000000000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load MASK for row 4
Readback errors ============================           0
   READ BACK 10000000000000000000000000000000000000000000
    EXPECTED 10000000000000000000000000000000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim0 for row 3
Readback errors ============================           0
   READ BACK 10000000000100000000000001000000000000000001
    EXPECTED 10000000000100000000000001000000000000000001
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim1 for row 3
Readback errors ============================           0
   READ BACK 00000000000000000000000010000000000000000
    EXPECTED 00000000000000000000000010000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim2 for row 3
Readback errors ============================           0
   READ BACK 00000000000000000000000000000000000000000000
    EXPECTED 00000000000000000000000000000000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
```

```
Fast load Trim3 for row 3
Readback errors ============================           0
   READ BACK 10000000000000000000000000000000000000000
    EXPECTED 10000000000000000000000000000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load MASK for row 3
Readback errors ============================           0
   READ BACK 00000000000100000000000000000000000000000
    EXPECTED 00000000000100000000000000000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim0 for row 2
Readback errors ============================           0
   READ BACK 10000000000100000000000001000000000000001
    EXPECTED 10000000000100000000000001000000000000001
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim1 for row 2
Readback errors ============================           0
   READ BACK 00000000000100000000000000000000000000000
    EXPECTED 00000000000100000000000000000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim2 for row 2
Readback errors ============================           0
   READ BACK 00000000000100000000000001000000000000000
    EXPECTED 00000000000100000000000001000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim3 for row 2
Readback errors ============================           0
   READ BACK 10000000000000000000000000000000000000000
    EXPECTED 10000000000000000000000000000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load MASK for row 2
Readback errors ============================           0
   READ BACK 00000000000100000000000000000000000000000
    EXPECTED 00000000000100000000000000000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim0 for row 1
Readback errors ============================           0
   READ BACK 10000000000100000000000001000000000000001
    EXPECTED 10000000000100000000000001000000000000001
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim1 for row 1
Readback errors ============================           0
   READ BACK 10000000000000000000000000000000000000001
    EXPECTED 10000000000000000000000000000000000000001
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim2 for row 1
Readback errors ============================           0
   READ BACK 10000000000100000000000001000000000000001
    EXPECTED 10000000000100000000000001000000000000001
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load Trim3 for row 1
Readback errors ============================           0
   READ BACK 00000000000000000000000001000000000000001
    EXPECTED 00000000000000000000000001000000000000001
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Fast load MASK for row 1
Readback errors ============================           0
   READ BACK 00000000000000000000000000000000000000000
    EXPECTED 00000000000000000000000000000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Slow load through pixels
Pad with zeros
Readback errors ============================           0
   READ BACK 10000000000100000000000000000000000000000
    EXPECTED 10000000000100000000000000000000000000000
Readback errors ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

```
Slow load through pixels
Configuration complete at time                26124
Begin full readback...
Slow load through pixels
Readback errors ============================           1
   READ BACK 1000000000000000000000000000000000000000000000
    EXPECTED 0000000s
Slow load through pixels
Readback errors ============================           2
   READ BACK 0000000000000000000000000000000000000000000000
    EXPECTED 1000000000000000000000000100000000000000000000
Slow load through pixels
Readback errors ============================           0
   READ BACK 1000000000000000000000000000000000000000000000
    EXPECTED 1000000000000000000000000000000000000000000000
Slow load through pixels
Readback errors ============================           4
   READ BACK 1000000000010000000000000100000000000000000001
    EXPECTED 0000000000000000000000000000000000000000000000
Slow load through pixels
Readback errors ============================           2
   READ BACK 0000000000000000000000000100000000000000000000
    EXPECTED 1000000000000000000000000000000000000000000000
Slow load through pixels
Readback errors ============================           4
   READ BACK 0000000000000000000000000000000000000000000000
    EXPECTED 1000000000010000000000000100000000000000000001
Slow load through pixels
Readback errors ============================           2
   READ BACK 1000000000000000000000000000000000000000000000
    EXPECTED 0000000000000000000000000100000000000000000000
Slow load through pixels
Readback errors ============================           1
   READ BACK 0000000000010000000000000000000000000000000000
    EXPECTED 0000000000000000000000000000000000000000000000
Slow load through pixels
Readback errors ============================           3
   READ BACK 1000000000010000000000000100000000000000000001
    EXPECTED 1000000000000000000000000000000000000000000000
Slow load through pixels
Readback errors ============================           0
   READ BACK 0000000000010000000000000000000000000000000000
    EXPECTED 0000000000010000000000000000000000000000000000
Slow load through pixels
Readback errors ============================           2
   READ BACK 0000000000010000000000000100000000000000000000
    EXPECTED 1000000000010000000000000100000000000000000001
Slow load through pixels
Readback errors ============================           2
   READ BACK 1000000000000000000000000000000000000000000000
    EXPECTED 0000000000010000000000000000000000000000000000
Slow load through pixels
Readback errors ============================           1
   READ BACK 0000000000010000000000000000000000000000000000
    EXPECTED 0000000000010000000000000100000000000000000000
Slow load through pixels
Readback errors ============================           3
   READ BACK 1000000000010000000000000100000000000000000001
    EXPECTED 1000000000000000000000000000000000000000000000
Slow load through pixels
Readback errors ============================           3
   READ BACK 1000000000000000000000000000000000000000000001
    EXPECTED 0000000000010000000000000000000000000000000000
Slow load through pixels
Readback errors ============================           1
   READ BACK 0000000000010000000000000100000000000000000001
    EXPECTED 1000000000010000000000000100000000000000000001
Slow load through pixels
Readback errors ============================           2
   READ BACK 0000000000000000000000000100000000000000000001
    EXPECTED 1000000000000000000000000000000000000000000001
Slow load through pixels
Readback errors ============================           3
   READ BACK 0000000000000000000000000000000000000000000000
    EXPECTED 0000000000010000000000000100000000000000000001
Slow load through pixels
Readback errors ============================           4
```

```
   READ BACK 1000000000010000000000000000000000000000000
    EXPECTED 0000000000000000000000000100000000000000001
Slow load through pixels
Readback errors ============================          0
   READ BACK 0000000000000000000000000000000000000000000
    EXPECTED 0000000000000000000000000000000000000000000
Slow load through pixels
Readback errors ============================          2
   READ BACK 0000000000000000000000000000000000000000000
    EXPECTED 1000000000010000000000000000000000000000000
```