Next pixel submission activities in Pisa

What Pisa did, does and will do for a pixel submission

- ***pixel/matrix readout architecture***

- ***full-custom implementation of in-pixel logic and matrix macrocell release for automatic P&R in Bologna (pins, no-routing area → LEF file)***

- ***integration of front-end block (from Pavia) with the pixel logic***

- ***where needed, final chip layout editing & checking (DRC, LVS, post-layout simulations)***

KEYWORDS
Readout architecture: scalability, timestamp, datapush, triggered.
Speed, noise and power consumption.

Submissions timeline:

After years of work on MAPS (we are not arrived yesterday) …

- first 3D Tezzaron-Chartered submission
  Old-style 8x32 pixel matrix readout architecture implemented (like previous Apsel4D: macropixels, external timestamp latches, …)

→ still waiting for some well manufactured 3D samples

Submissions timeline:

- first readout chip for a high-resistivity pixel 50x50 μm^2 sensor (**Superpix0**);
  again macropixels but implemented the **submatrices array slicing** (fundamental to sustain SuperB rates), slow control revised, pixel reset on readout (no extra operation needed)

- on **2011** submitted a **INMAPS** design containing a new fully-scalable architecture looking at big chips, now it is under test:
  - 8-bit timestamp latch on pixel
  - no more macropixels
  - two submatrices
  - power supply current peaks reduced
  - datapush or triggered readout

- **2012** 2nd 3D submission on the way, **MAPS and HR sensors**, biggest MAPS matrix ever for us:
  - 96x128 50x50 μm^2 pixel array divided into two readout-independent submatrices
  - **Superpix1**, 32x128 readout chip for HR pixel sensor
  - timestamp resolution less than 100ns
  - datapush or triggered selectable modes as in INMAPS
  - current peaks extensively reduced
  - single pixel injectable, threshold correction DAC for each pixel

Pixel, you provoked me …

From behavioral (mental) model to the smallest and fastest cmos in-pixel circuit

No crosstalk!

Scalability!

Smaller!

Faster!

Less power!

Yield!

Less metals for routing!

Minimal dead area!

"Pasta, you provoked me and I'll destroy you now, pasta! I'm going to eat you, ahmm!"

mmm … I must build my own gates

and well pack them in functional blocks

Transparent latch

Smaller XOR

8-bit TS latch

8-bit comparator

Hit DFF

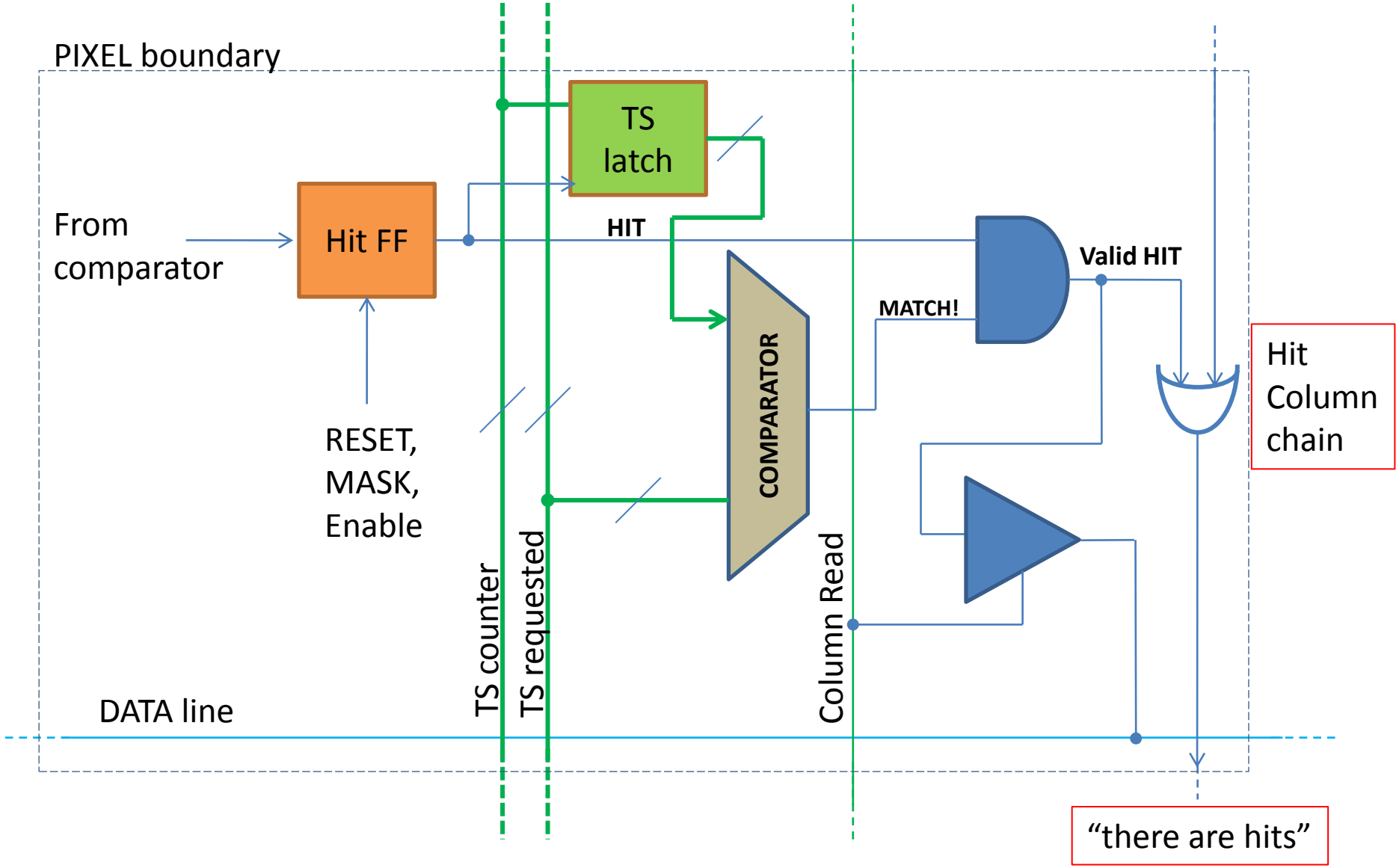
Data line drivers

Efficient Hit OR Column chain

16 delays

Relax moment,
check your email now …

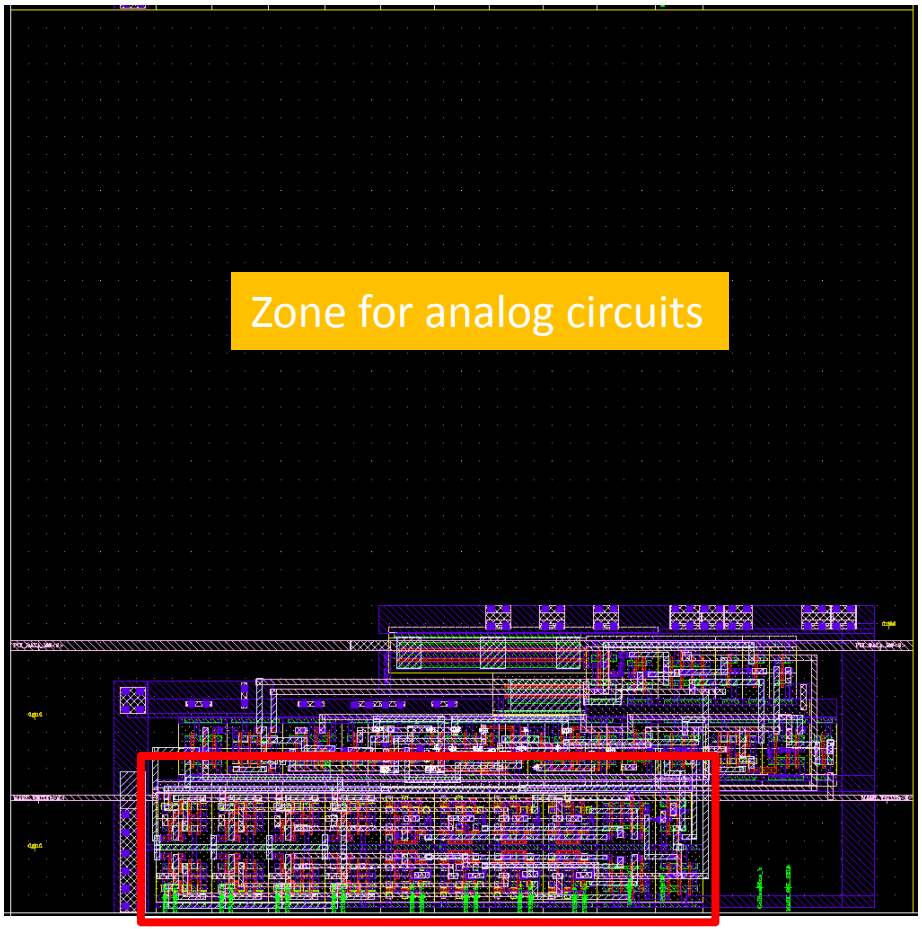
Logic inside a pixel (not detailed)



Different processes, different layouts

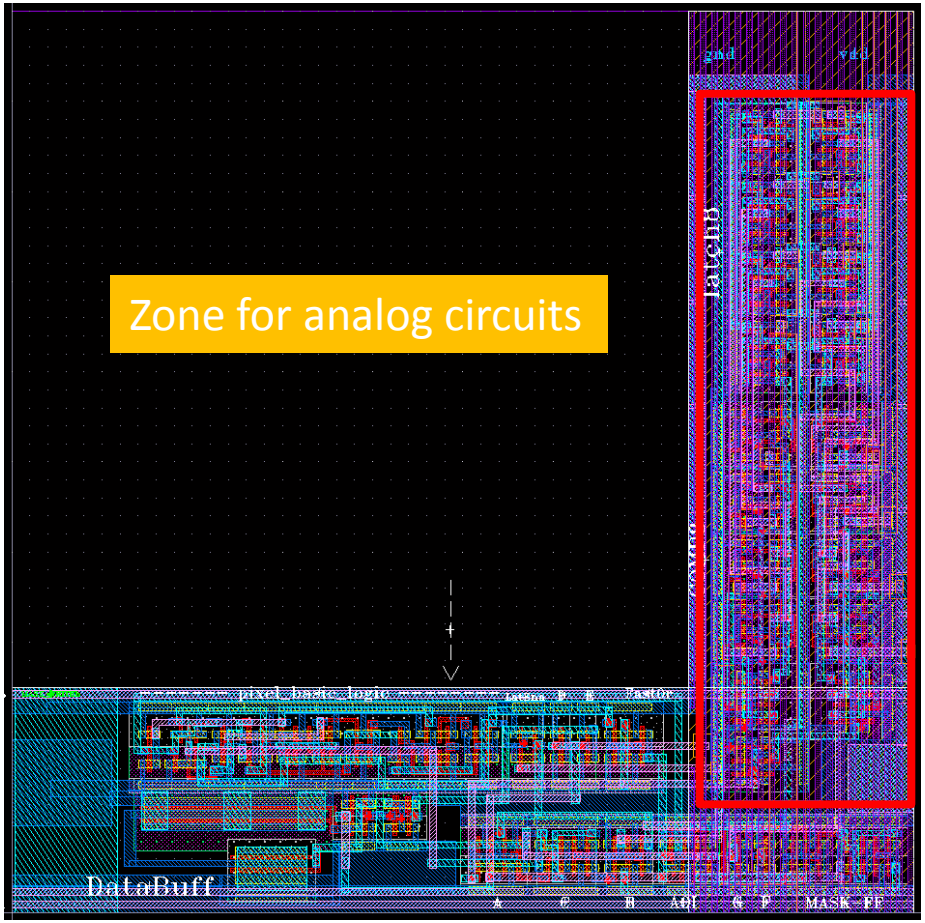
(same circuit of ~240 trans., no power and vertical common signal rails shown)

Tezzaron-GF, 130nm



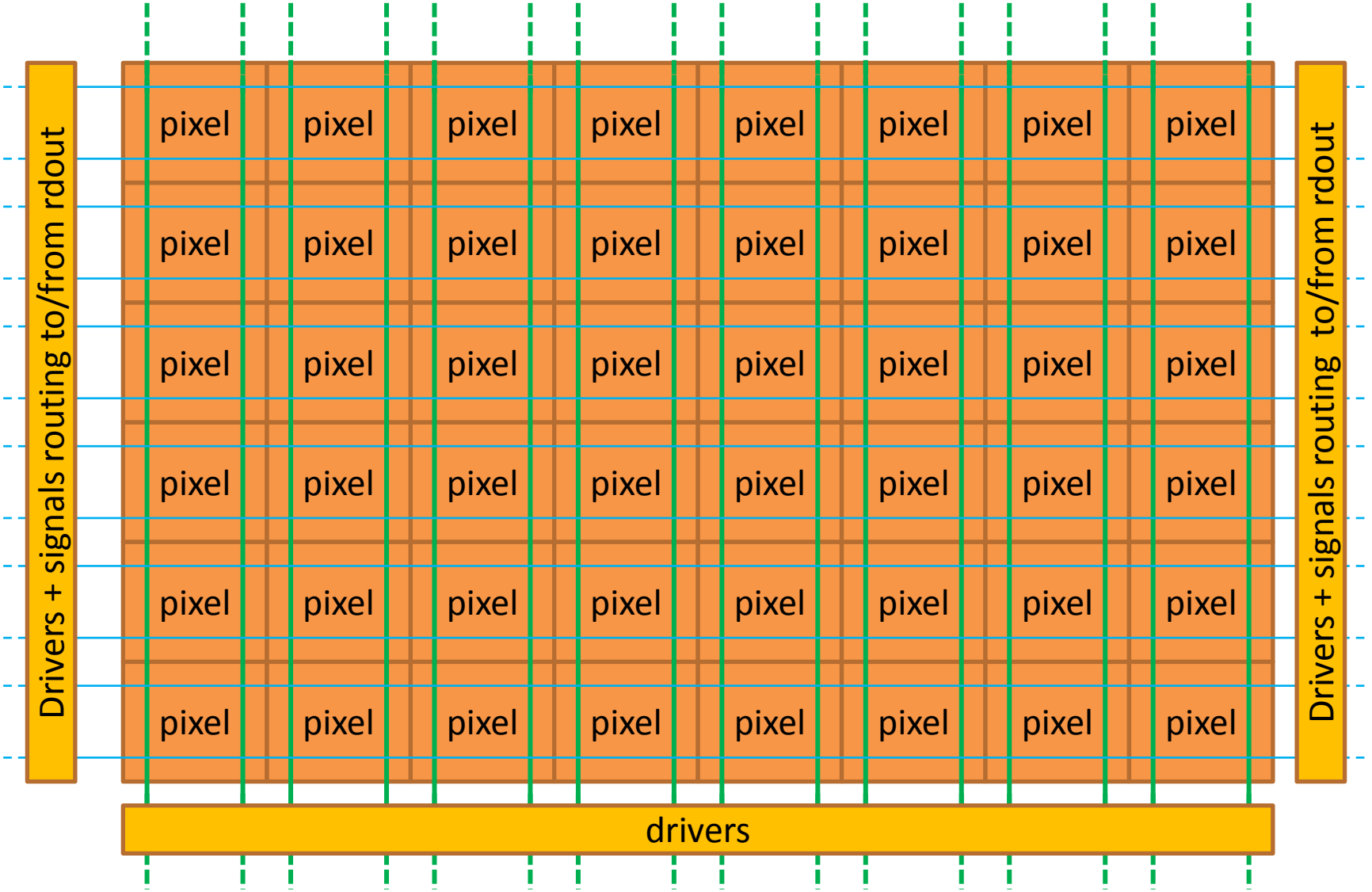
8-bit TS latch + comparator

INMAPS, 180nm



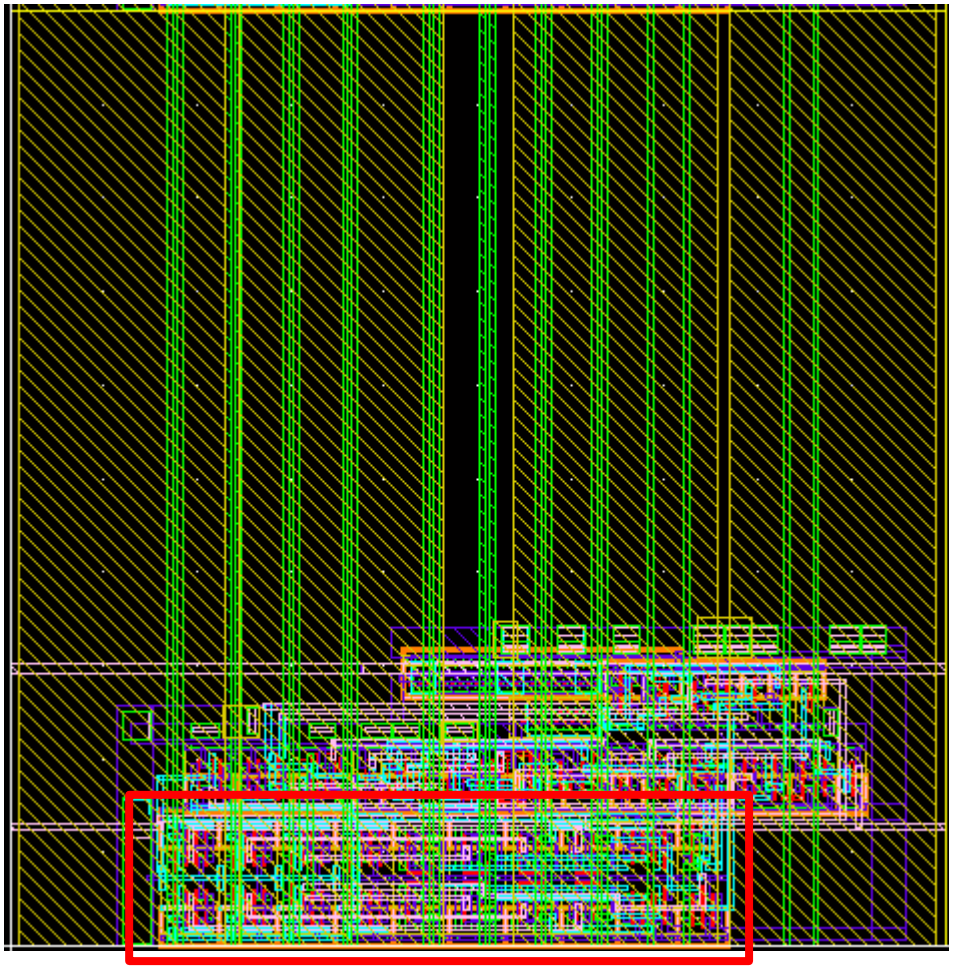
How to build the matrix.

Common lines signals and powers are vertically (24) and horizontally (2) shared



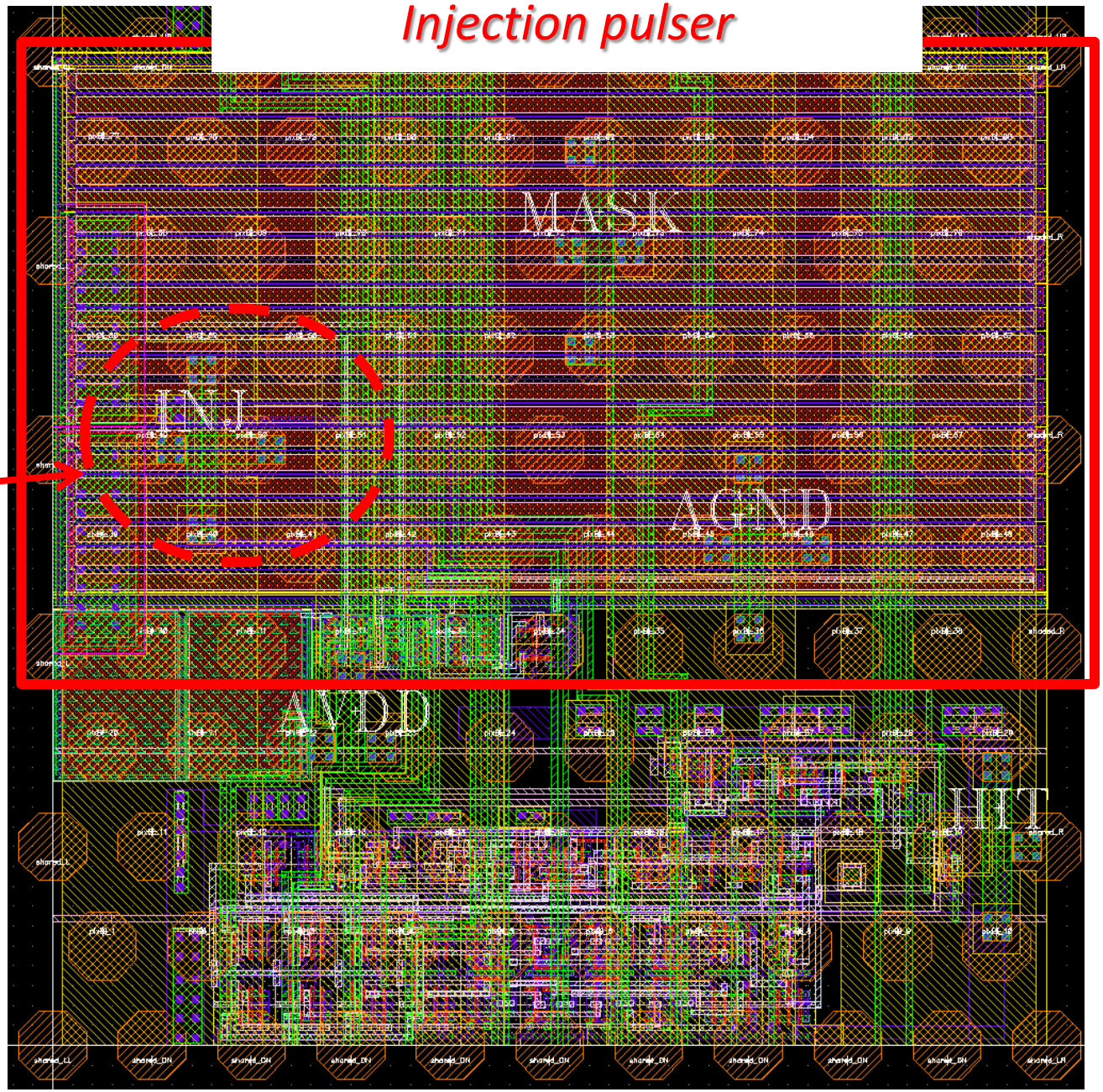
The pixel with my logic only, before analog part integration

Note the green (M4) straight vertical lines: Timestamps, control lines. The two horiz. white lines (M3): data out, control line.



8-bit TS latch + comparator

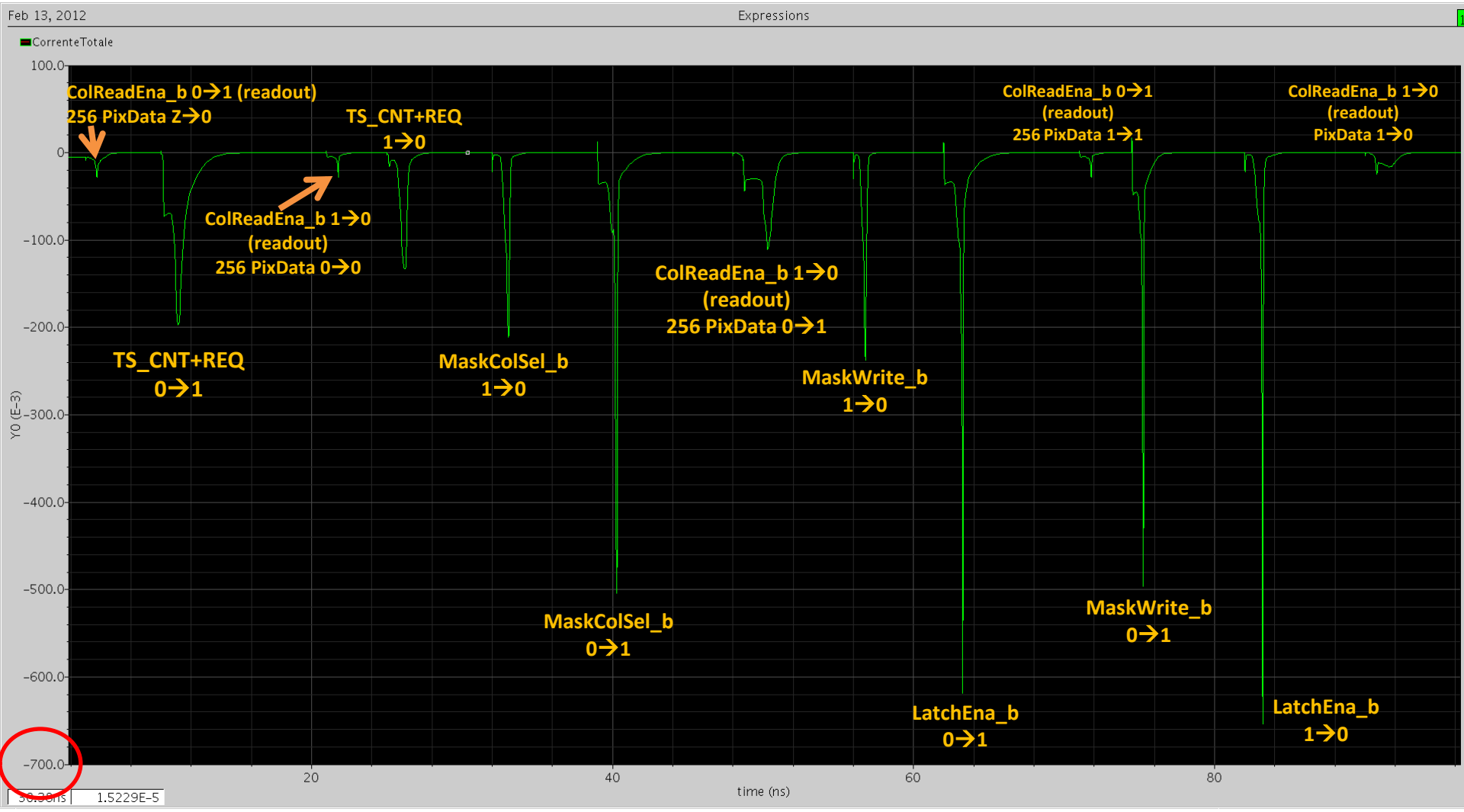
Final pixel, digital layer with both digital and analog parts.



crosstalk

Note how the green (M4) straight vertical lines have been "distorted"

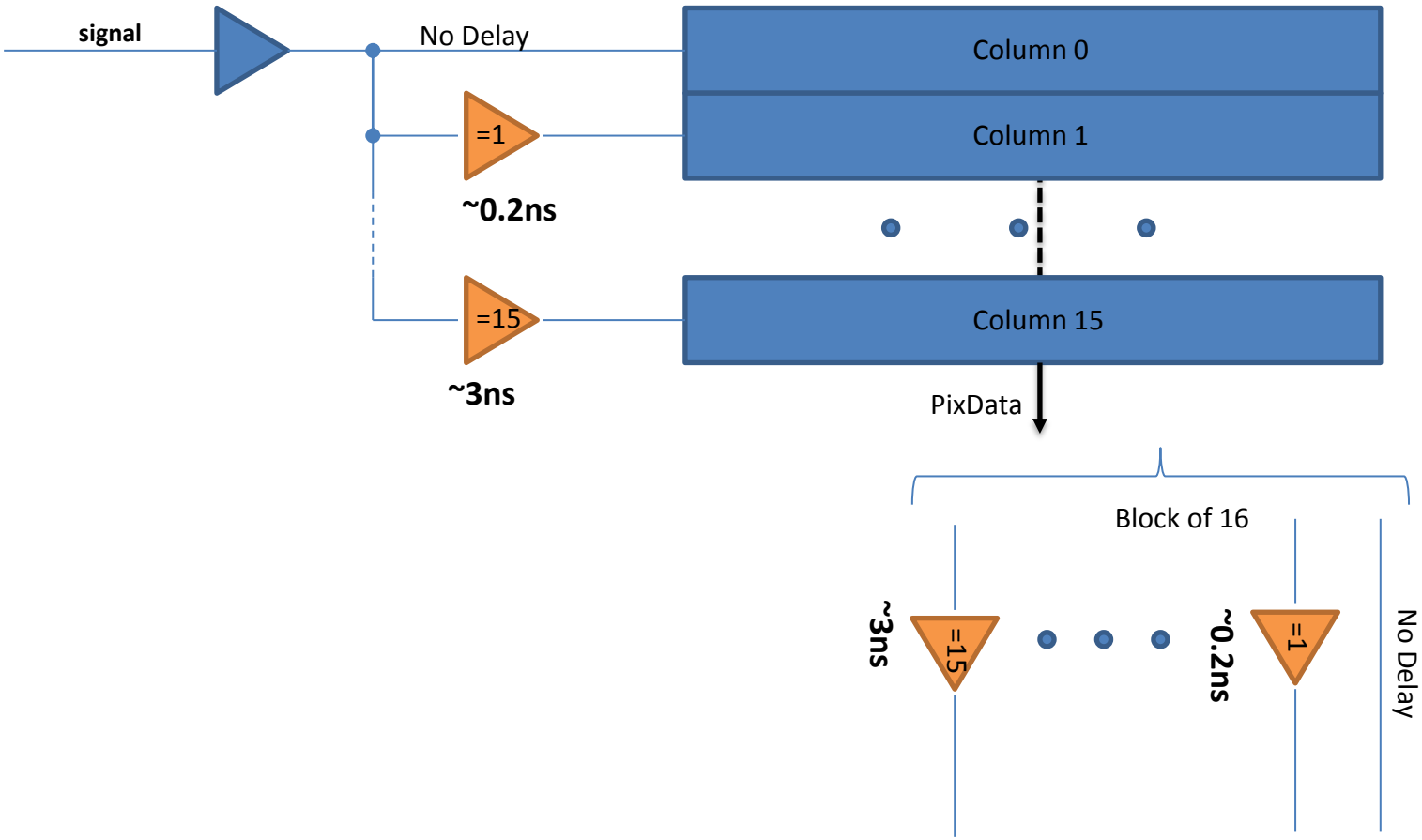
Current peaks in a 96x128 matrix



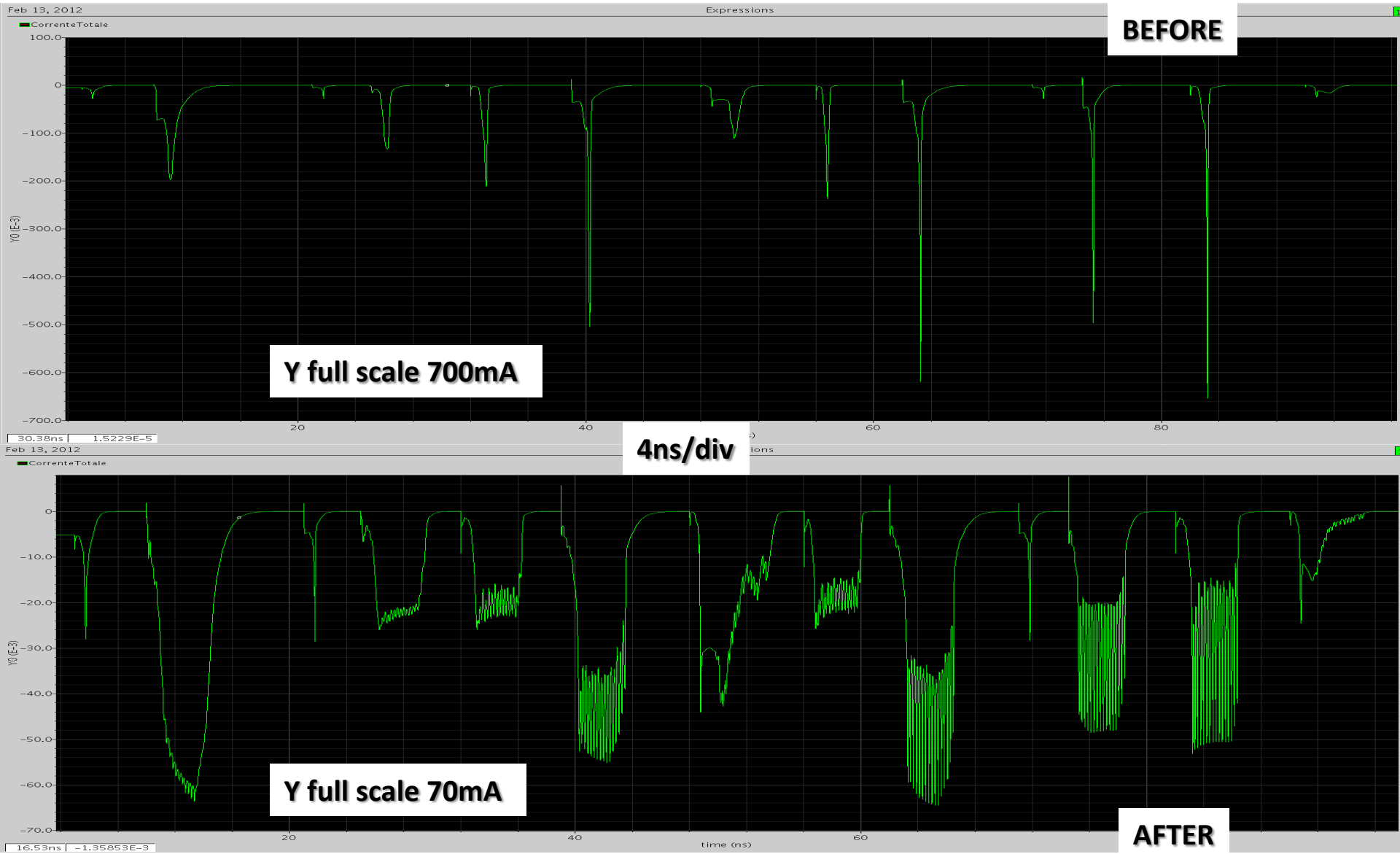
Worst case: ~ 700mA

The cure

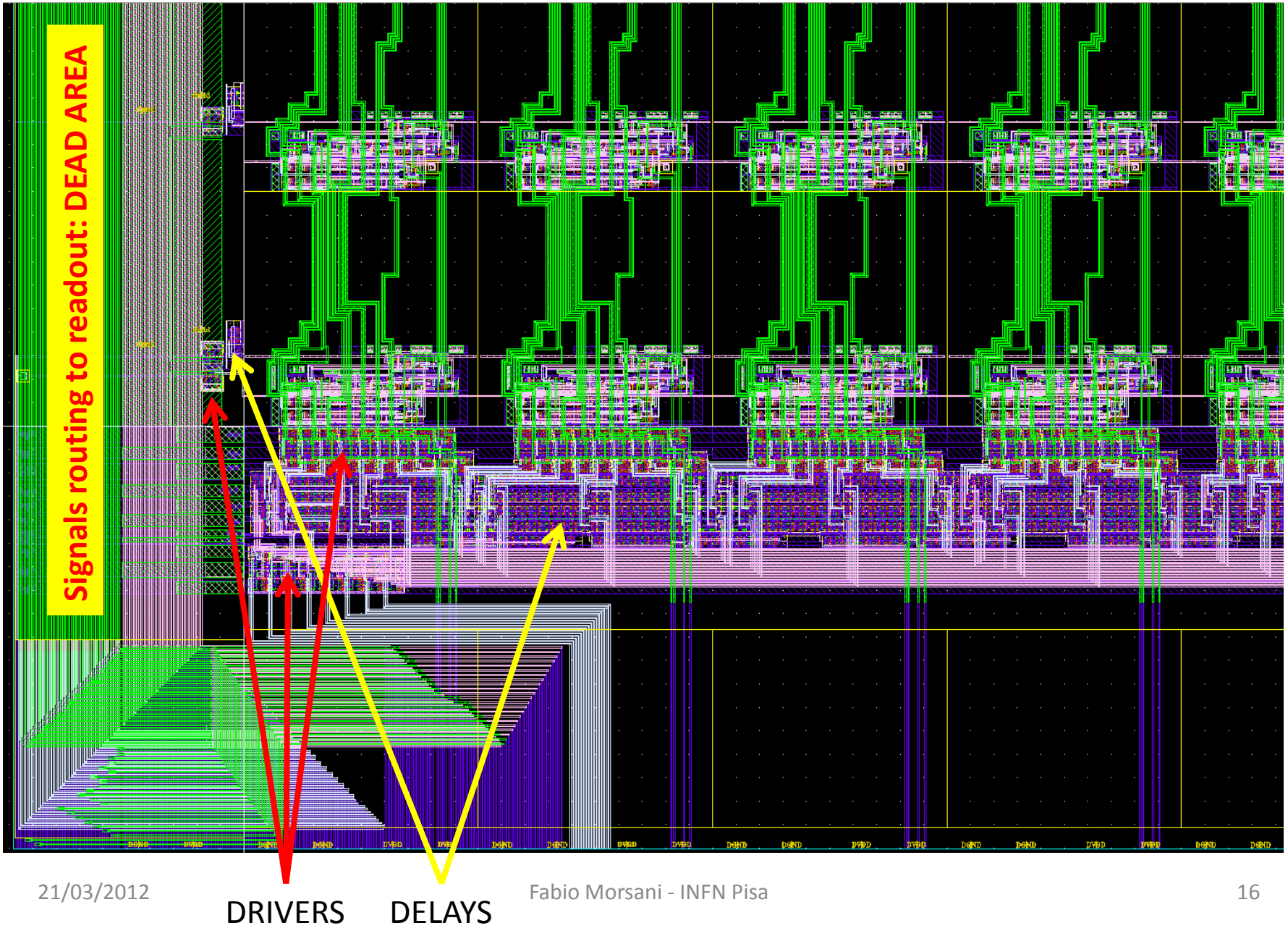
Each column-shared input and row-shared input and output signal is delayed from 0 to ~3ns in blocks of 16, as shown.
Each data output (PixData, row-shared output signal) is delayed in the same way before to reach the readout block.



Differences: note the scales and the peaks spread over time



Matrix bottom-left corner, analog part not shown



Simulations and next steps

Simulations:

Spectre for small circuits, Verilog and mixed Verilog-Ultrasim for the whole matrix (about 3M trans. in a 96x128 matrix)

Verilog models for all gates present on the circuit. This allows the circuit import in the readout simulation environment (Bologna)

Next steps:

-Analog part integration (with Pavia, in progress)

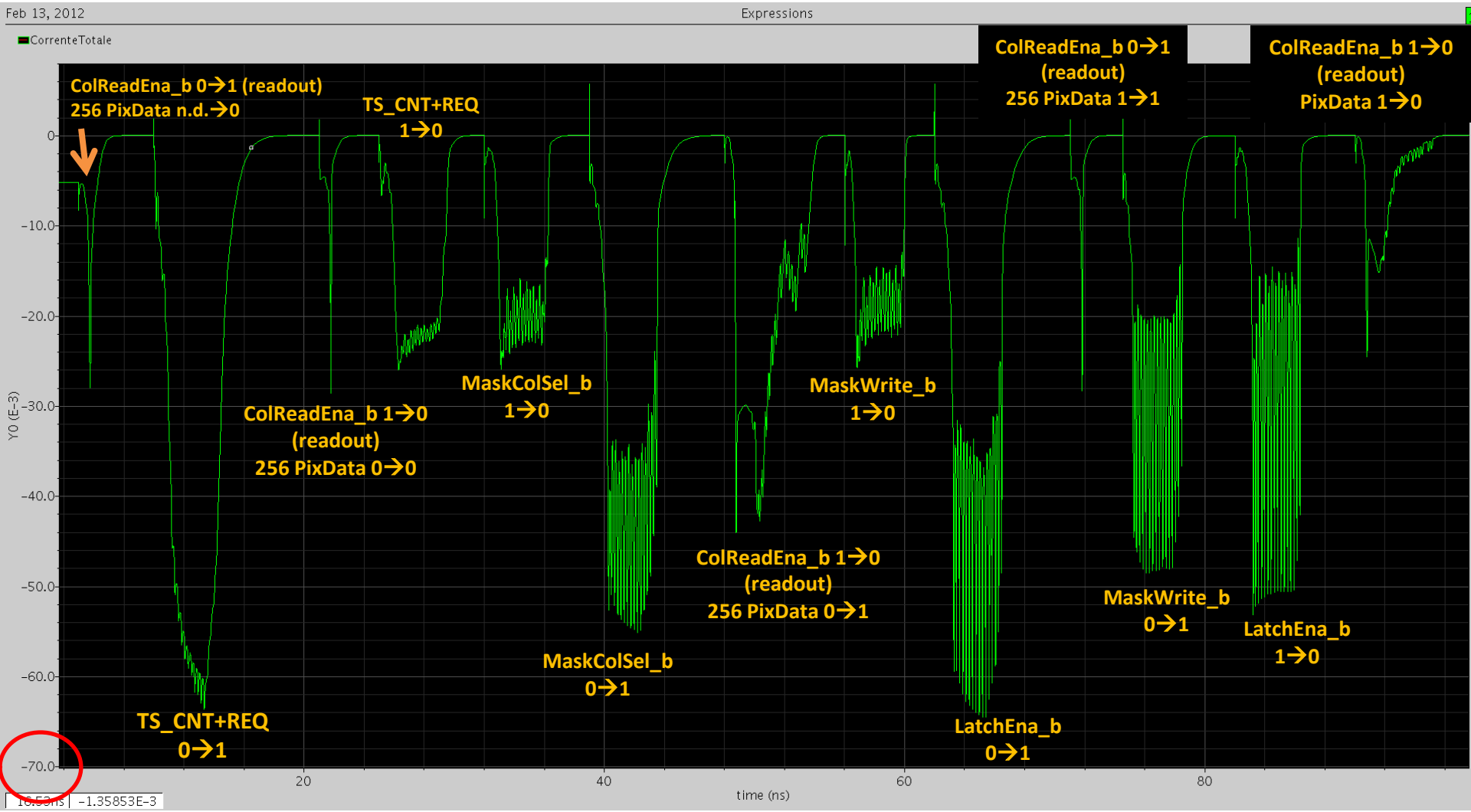
-File for automatic P&R to Bologna

-P&R, back complete chip from Bologna, checks and possible re-iteration

Submission in July?

… and finally we will have a working chip!

Current peaks after the cure



TS + readout = 110mA[310], GlobalPixelEnable(LatchEna_b) = 65mA[620], MatrixReset(MaskWrite_b) = 48mA[500], TS = 65mA[200], Readout = 45mA[110]