High efficiency readout architecture for a large matrix of pixels

RD09 – Firenze, 10/02/2009

A. Gabrielli, **F.M. Giorgi**, M. Villa for the **VIPIX** collaboration

INFN & University of Bologna

Outline

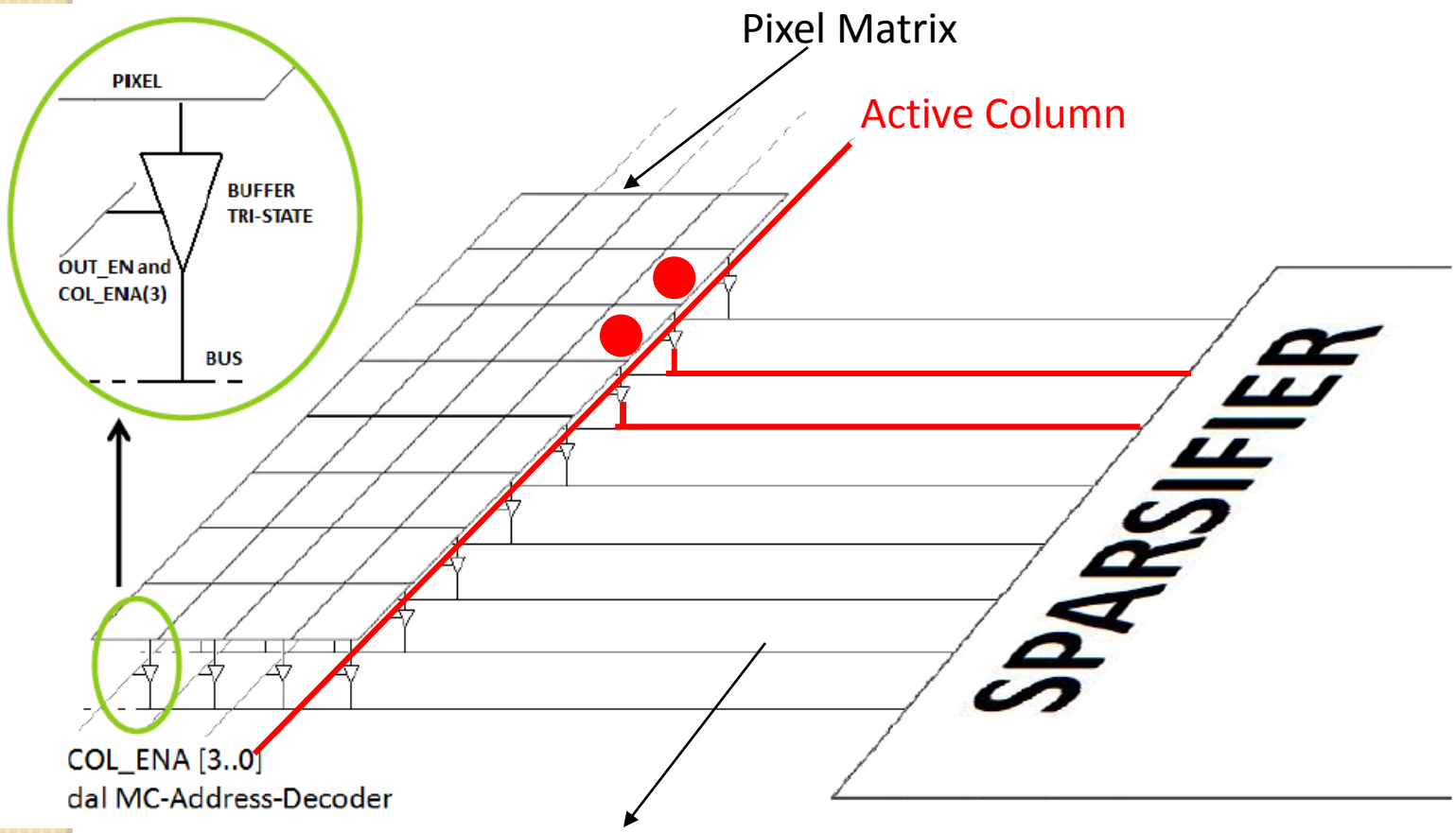
- Expected Conditions
- Matrix overview
- Readout architecture
- Slow control interface
- Simulations
- Efficiencies
- Test Chip submission Sept. 09

Expected conditions

- 100 MHz/cm^2 hit rate
- 0.25 – 2.0 µs BCO clock:
  - Time Counter clock, represents the time granularity of the events.
- 60-100 MHz Matrix Read Clock
- 3 Gbit/s data bus bandwidth per chip

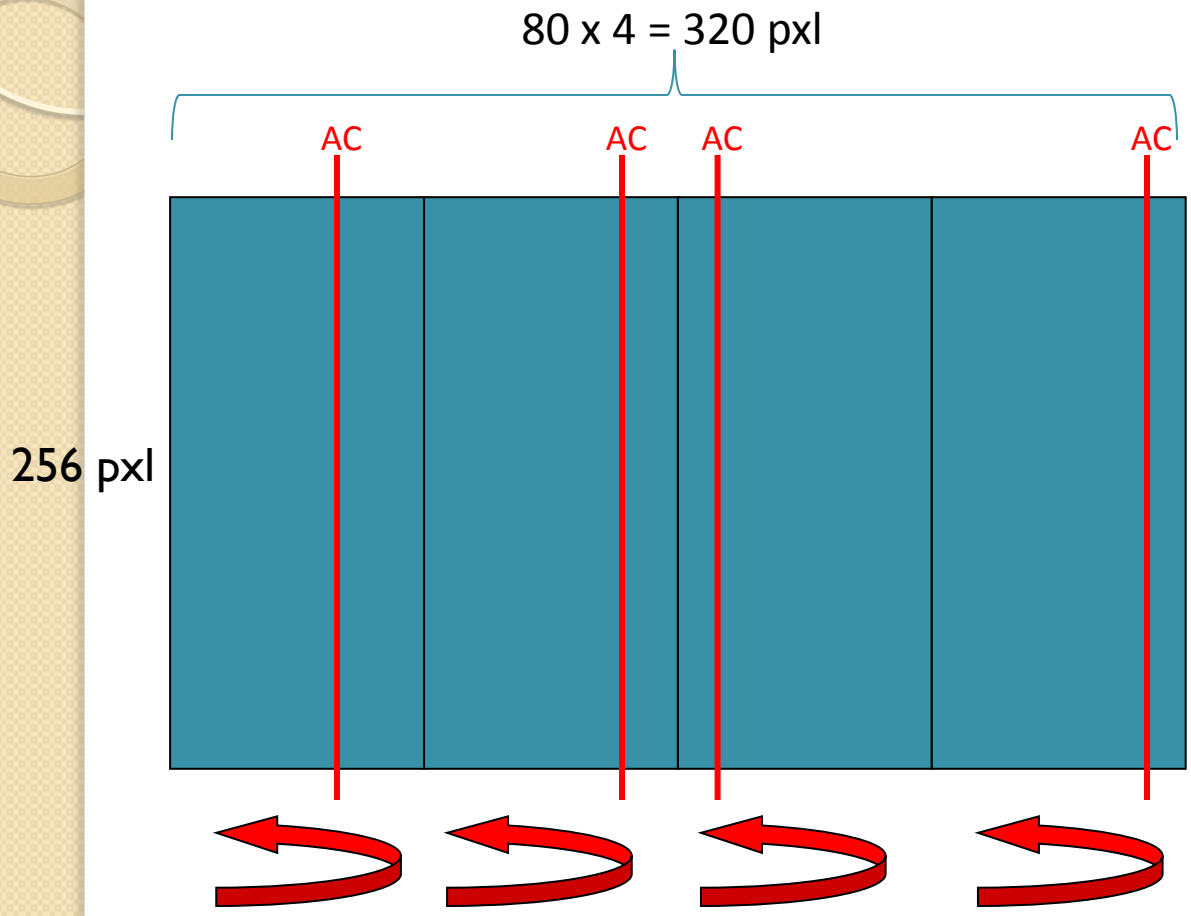
Matrix overview

- Binary **pixels matrix**
- Hit readout through a column-wide **shared data-bus**



Common Pixel Data Bus – Active 1 column of pixel at a time

The Matrix 320x256



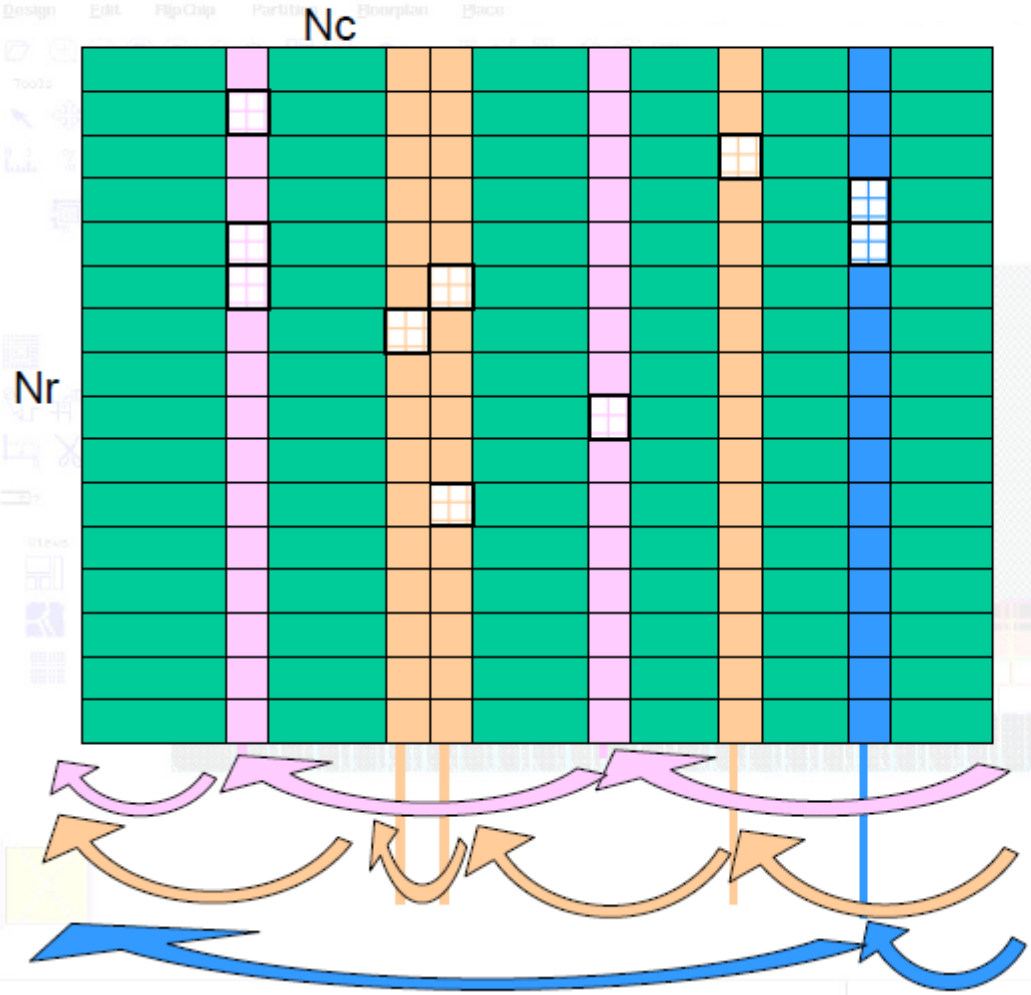
- 80K pixel matrix
- Total area ~ 1.3 cm^2
- 130 Mhit/s

4 independent Submatrix

1 000 pixels sparsified in 1 CLK cycle

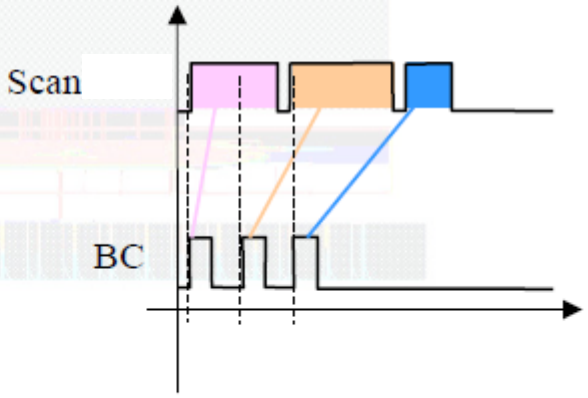
4 horizontal **parallel** scans for the Active Column
80 x 256 pxl, **40** microns pitch

Submatrix Scan Policy



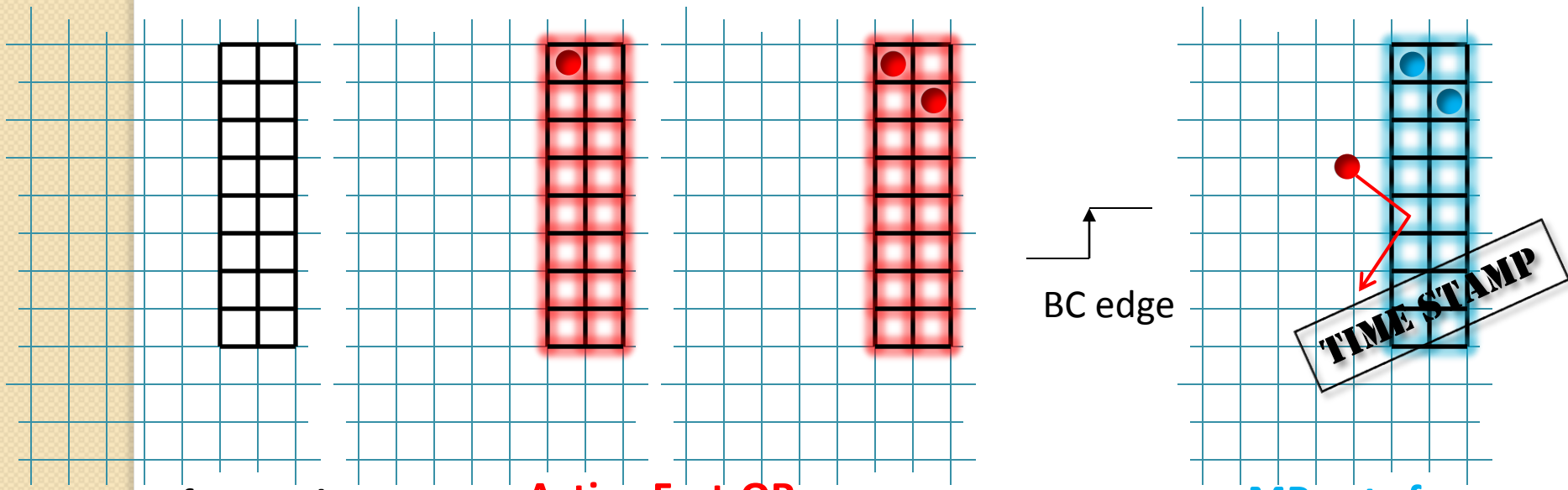
Hit-TS1
Hit-TS2
Hit-TS3

Column scan time oriented:
1 scan for each Time Stamp
→ The hits are time sorted



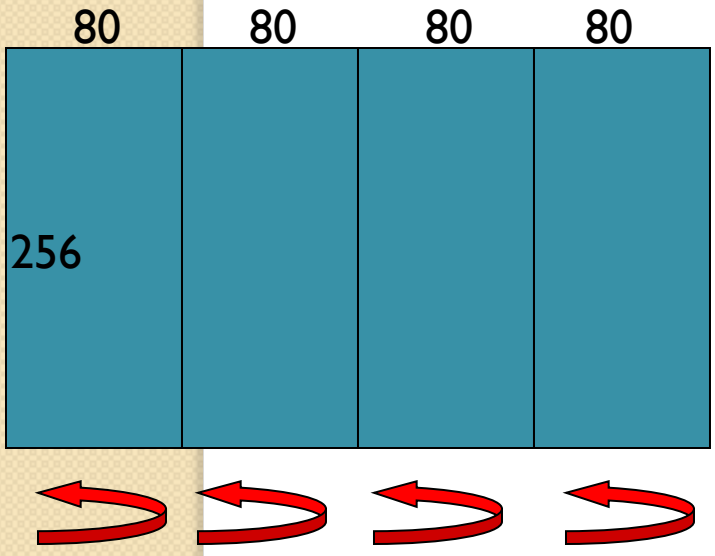
The Macro Pixels

- Matrix divided into MPs: **group of pixels** (2x8)
  - MP global lines:
    - **Fast-OR line:** (MP output) inclusive OR of all pixel latches.
    - **Freeze line:** (MP input) disable the reception of new hits.
  - On BCO clock edge all MPs **with active fast-OR** :
    - Gets frozen
    - Are associated to the current value of BCO counter (Time Stamp)
    - Waits to be scanned and reset

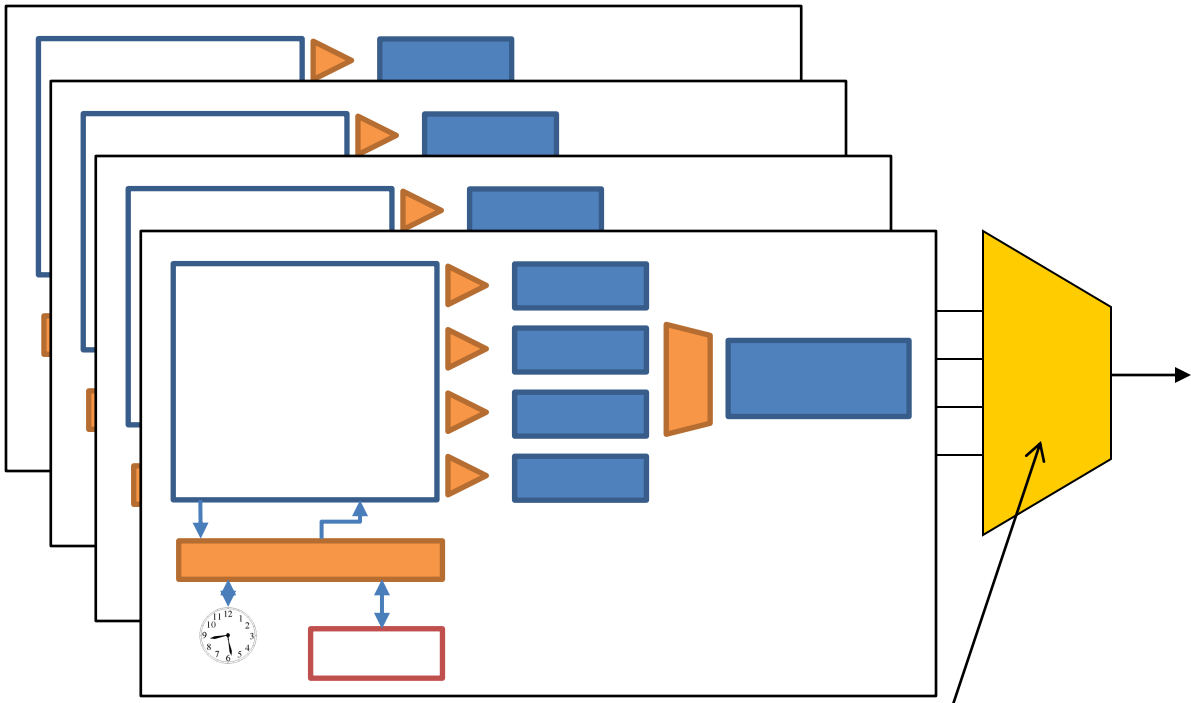


Matrix readout architecture

- Each sub-matrix scan has its own readout & scan logic
- All readout working in parallel
- Common final output stage

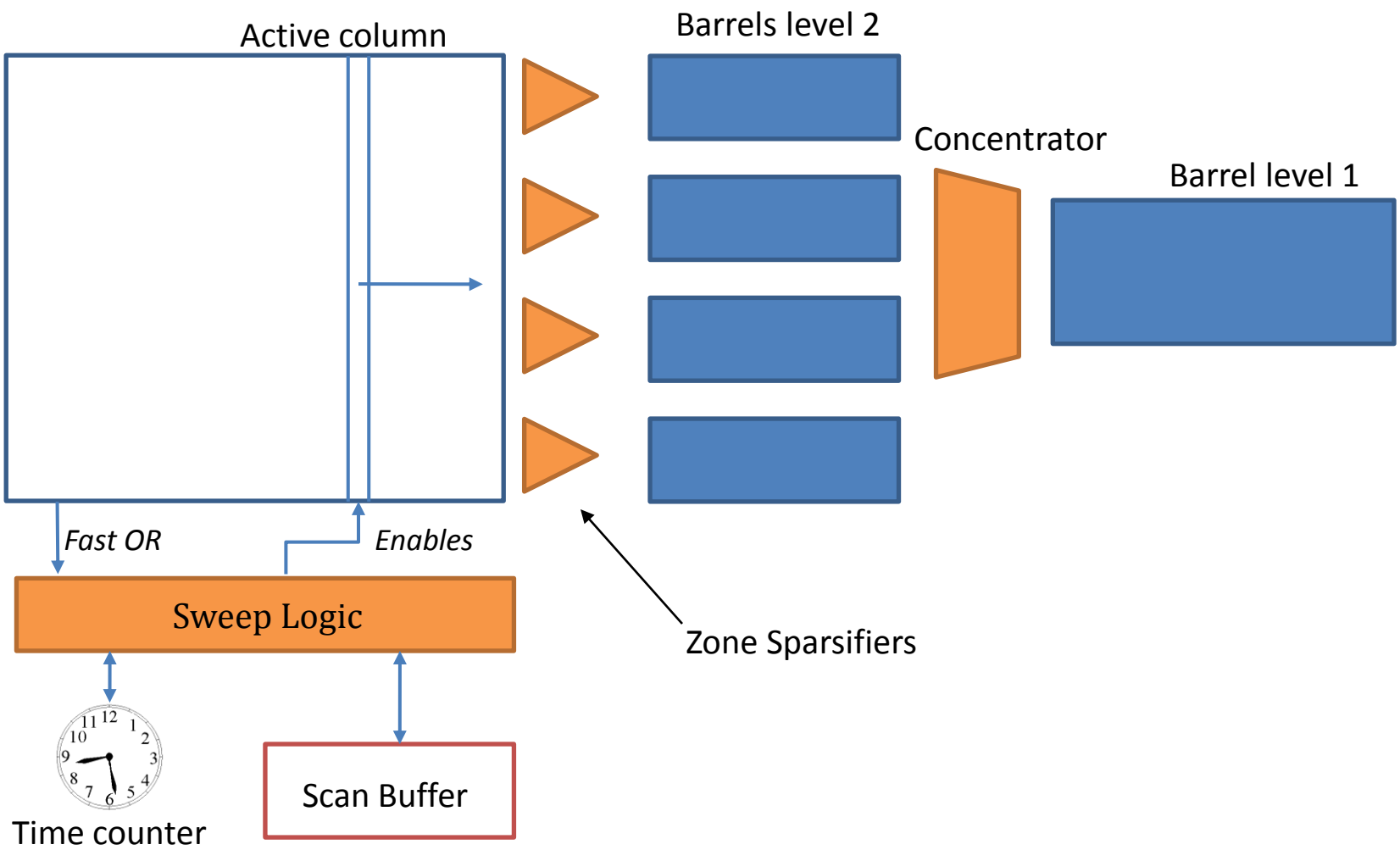


4 parallel scan

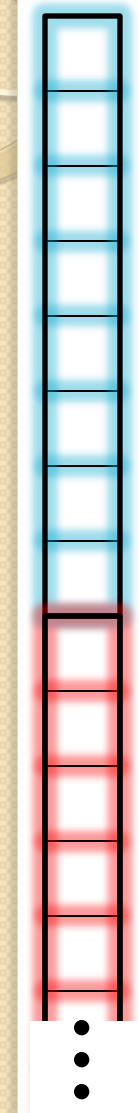


Common output stage

Sub-matrix readout architecture

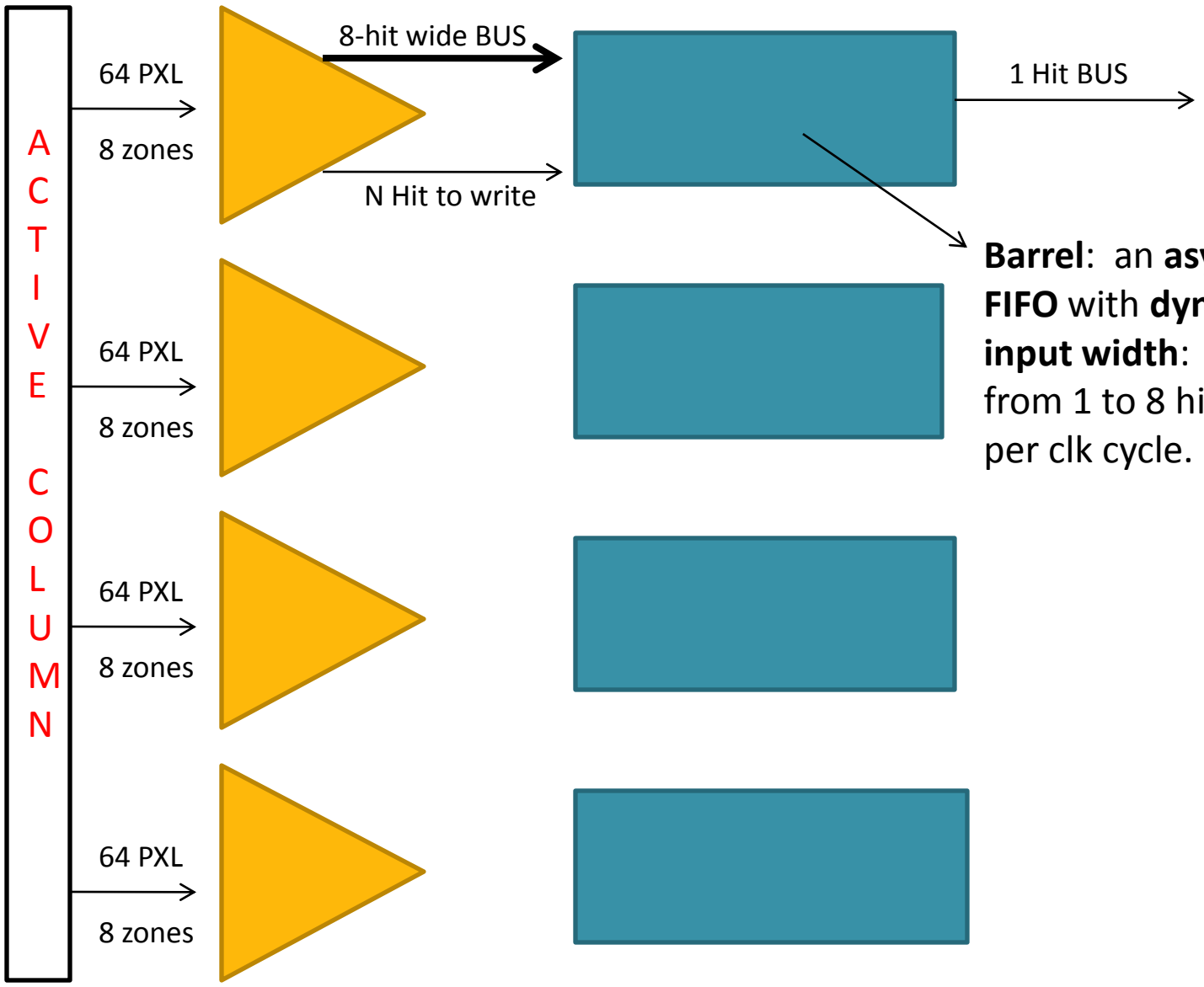


Zone sparsification



- The **256 pixels** of the Active Column are **divided into 32 vertical zones (1x8 pixels each)**
- Every **HIT** stored contains the information of a zone, **not of a single pixel**:
  - HIT= (**Zone address**+ **Zone pattern**)
    - X zone address = Column address for 80 columns →7 bit
    - Y zone address : 32 vertical zones → 5 bit
    - Zone pattern: 8-pixel zone → 8 bit
- **Time Stamp**: since the hits are time sorted, the relative TS word is stored at the beginning of each hit sequence

The sparsifiers and barrels



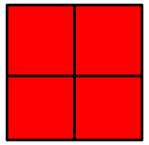
Barrel: an **asymmetric FIFO** with **dynamic input width**:
from 1 to 8 hits stored per clk cycle.

Output stage solutions

- FULL resolution Hit + time stamp
  - 8 bit TS (modulo 256 BCO counter)
  - 9 bit X address (320 pixels)
  - 8 bit Y address (256 pixels)
  - TOT **25 bit**
  - → **expected rate 130 MHit/s per chip = 130MHz x 25bit = <u>3.2 Gbps</u>**

- Zone sparsification & time sorting of the hits (TS heading the relative hits, 1 MHz BC clock) lead to:
  - 2 bit Barrel L2 address (→1/4 of submatrix: 80x64 pxl)
  - 2 bit Barrel L1 address (1 submatrix: 80x256 pxl)
  - 7 bit X address (80 pixels)
  - 3 bit zone Y address (8 vertical zones for each L2 barrel)
  - 8 bit zone pattern
  - TOT 22 **bit**
  - → **expected rate: 130 (+1 TS) * 22 = <u>2.8 Gbps</u>**

BUT: assuming a x4 cluster factor of the form 2x2:
in 87.5% of cases 2 hits only & in 12.5% are required 4 hits



  - → **[(22*2)* 0.875 + (22*4)*0.125] *25 Mtrack s^{-1} cm^{-2} * 1.3 cm^2**
    - **Weighted average ~ <u>1.6 Gbps</u>**

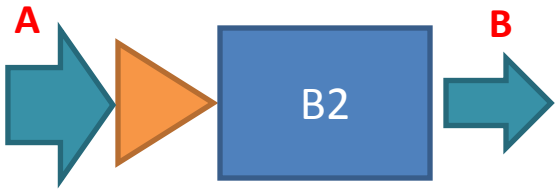
VHDL model simulations

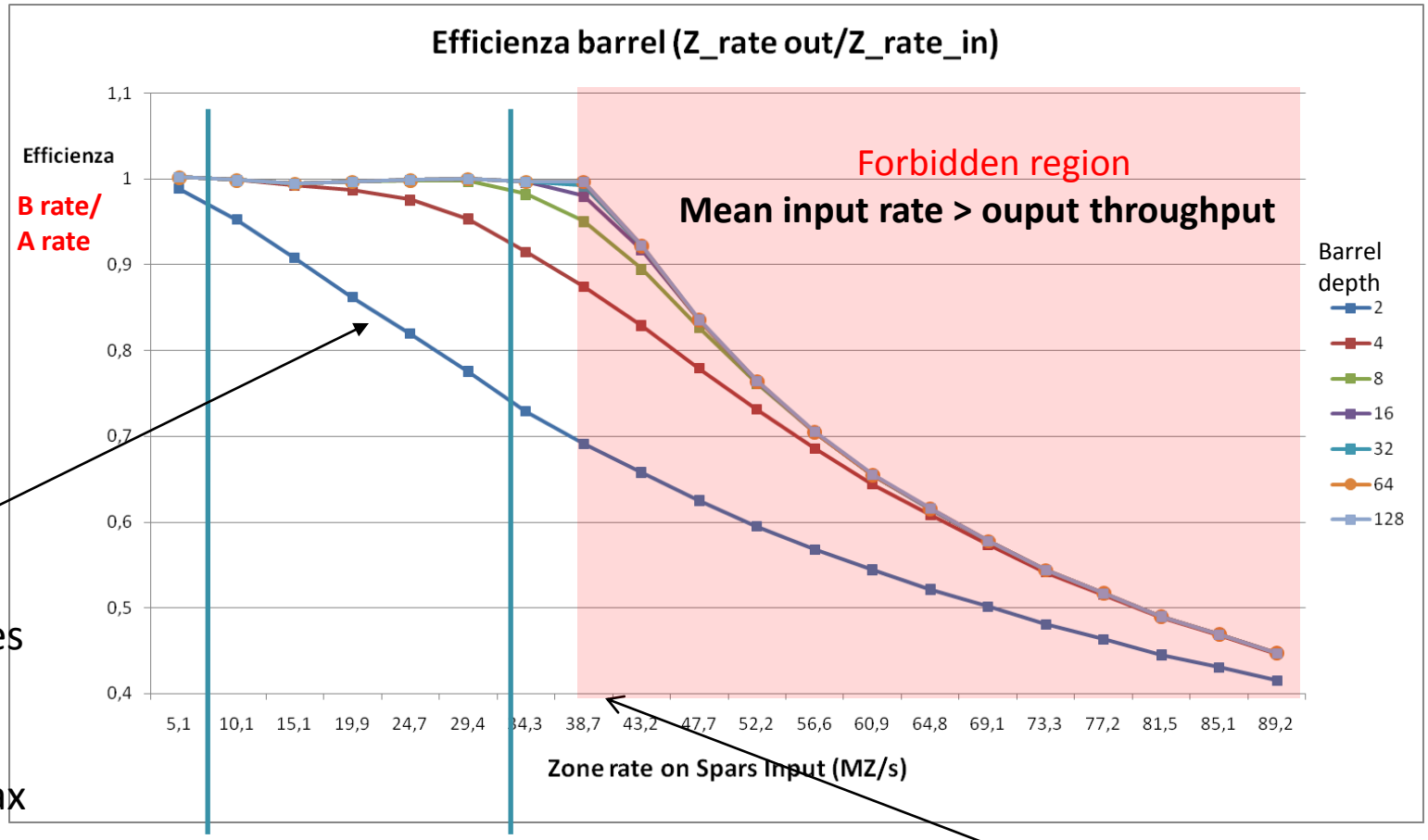
- Models verification
- Estimation of the optimal parameters for the expected working conditions (hit rate, clock frequency …)
  - Barrels depth
  - Zone width
- Efficiencies estimation

SIMULATIONS: the infrastructure

- Realistic VHDL model of a Sub-matrix for behavioral simulation.
  - 2D array of MP entities, each one with uniform random hit generation. User-defined hit rate.
  - NO pixel dead time taken into account. (pixel immediately reset after read)
- VHDL test bench
  - Integrated data integrity check.
  - Efficiencies evaluation.
  - File logs
    - Simulation runs e-log (storing the whole parameter set for each simulation)
    - Frozen hit log (once a MP gets frozen, the fired pixels within are stored in absolute x-y format)
    - B2 Readout log (stores the hits read out from any of the B2 decoded in absolute x-y format)
    - B1 Readout log (stores the hits read out from B1 decoded in absolute x-y format)
    - Output log (stores the hit read out from the final output stage decoded in absolute x-y format)
- Hit controller program: a C++ tool that checks the correspondence between the frozen hit log and output log.

Study on Barrel optimal Depth:





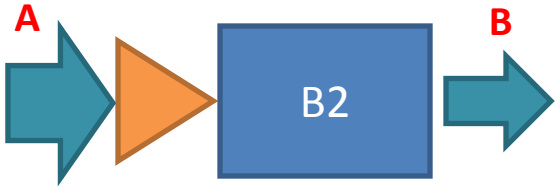
Barrel inefficiencies due to fluctuation over the max throughput

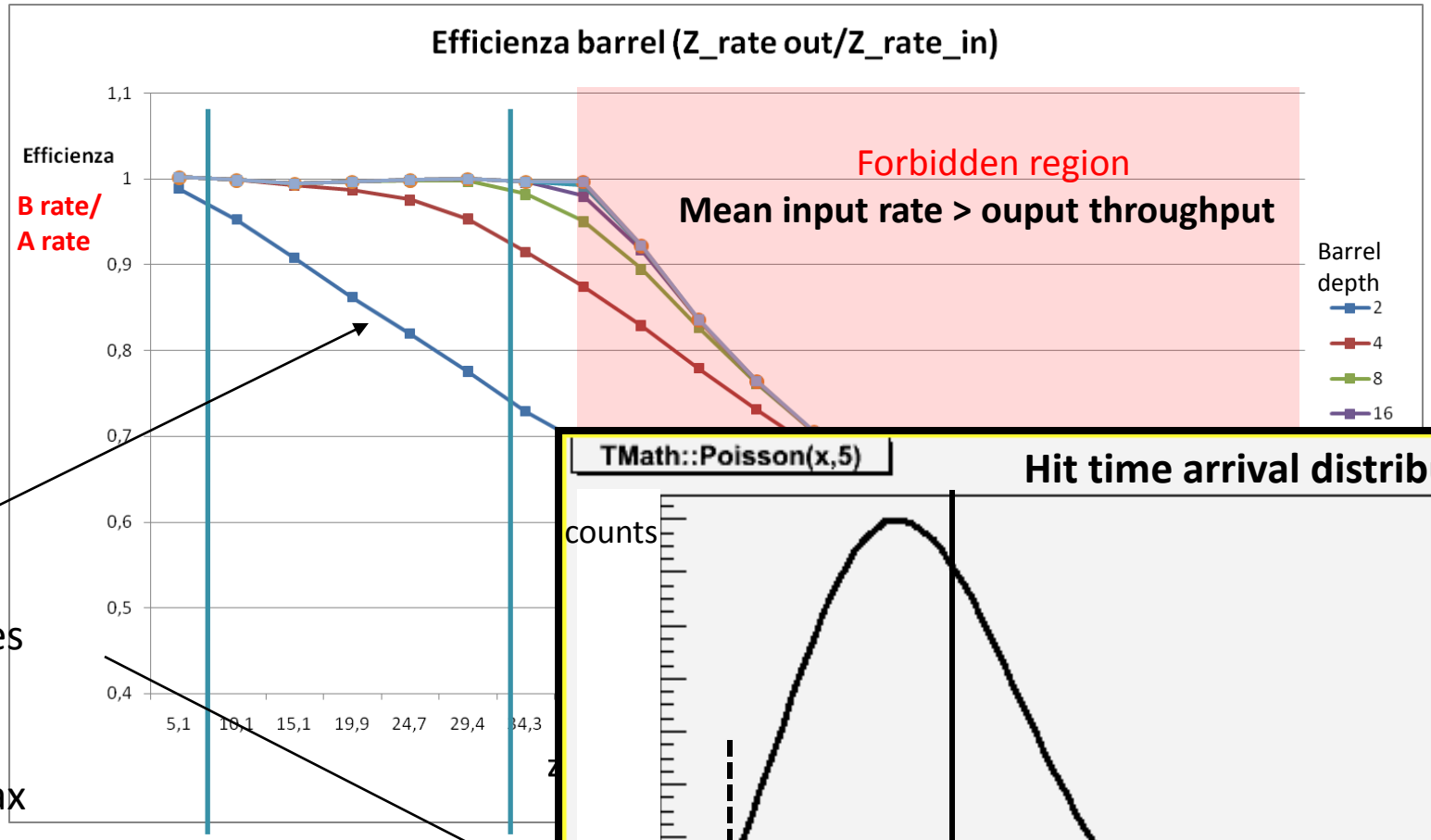
Expected rate @**B2**: 8,2 MPxl/s

Expected rate @**B1**: 32,8 MPxl/s

Barrel clock 40 MHz

Study on Barrel optimal Depth:





Barrel inefficiencies due to fluctuation over the max throughput

Expected rate @**B2**: 8,2 MPxl/s

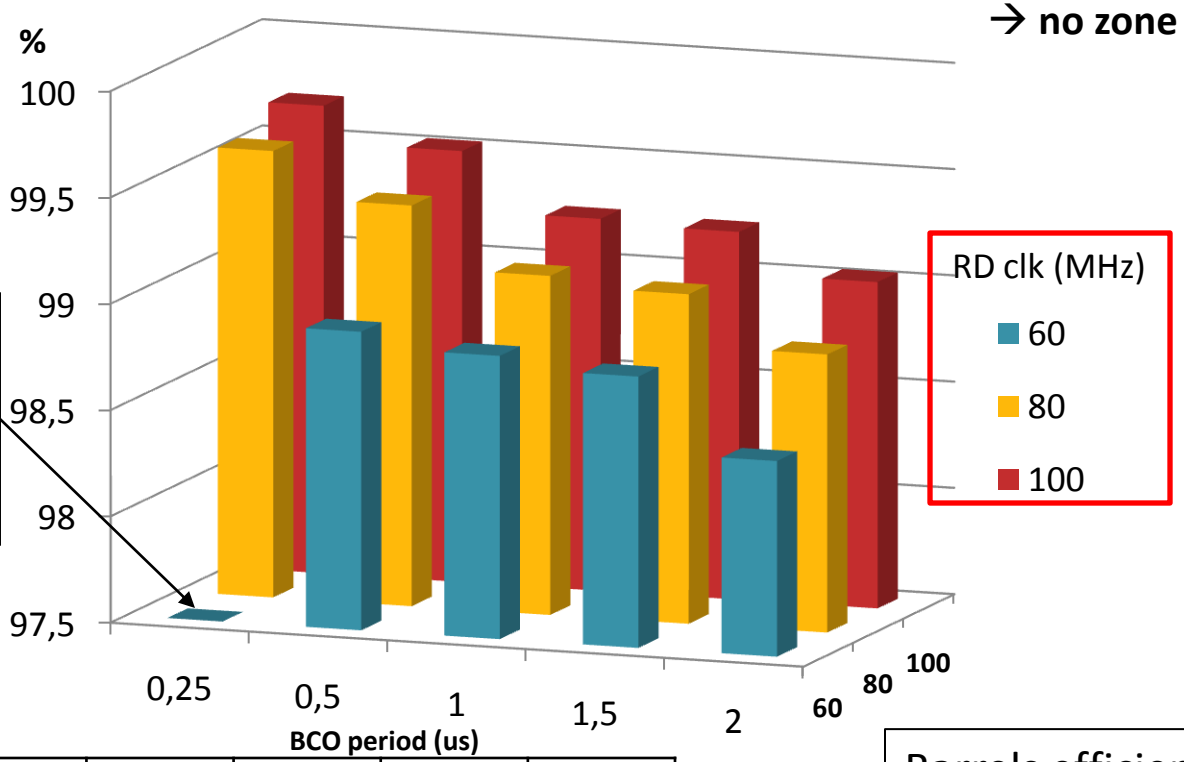
Expected rate @ 32,8 MPxl/s

Efficiencies

- Two sources of inefficiency due to digital readout:
  - **Frozen MP inefficiency**: hits generated on a frozen MP are lost.
  - **Overflow inefficiencies**: when a buffer goes full it looses all the incoming hits.

Frozen MP Efficiency @ 100 MHz/cm^2

(Random and no clusters → **no zone benefits** at this rate)



Efficiency drop induced by the overflow of *Scan Buffer*: Longer freezing time

frozen effi.						
				BCO (us)		
		0,25	**0,5**	**1**	**1,5**	**2**
	100	99.7	99,5	99,3	99,2	99,0
Rd_clk (MHz)	**80**	99,6	99,4	99,1	99,1	98,8
	60	97,5	98,9	98,8	98,8	98,4

Barrels efficiency **100%**
No barrel overflow
(B2 and B1)

Fast read clock and narrow BC edges

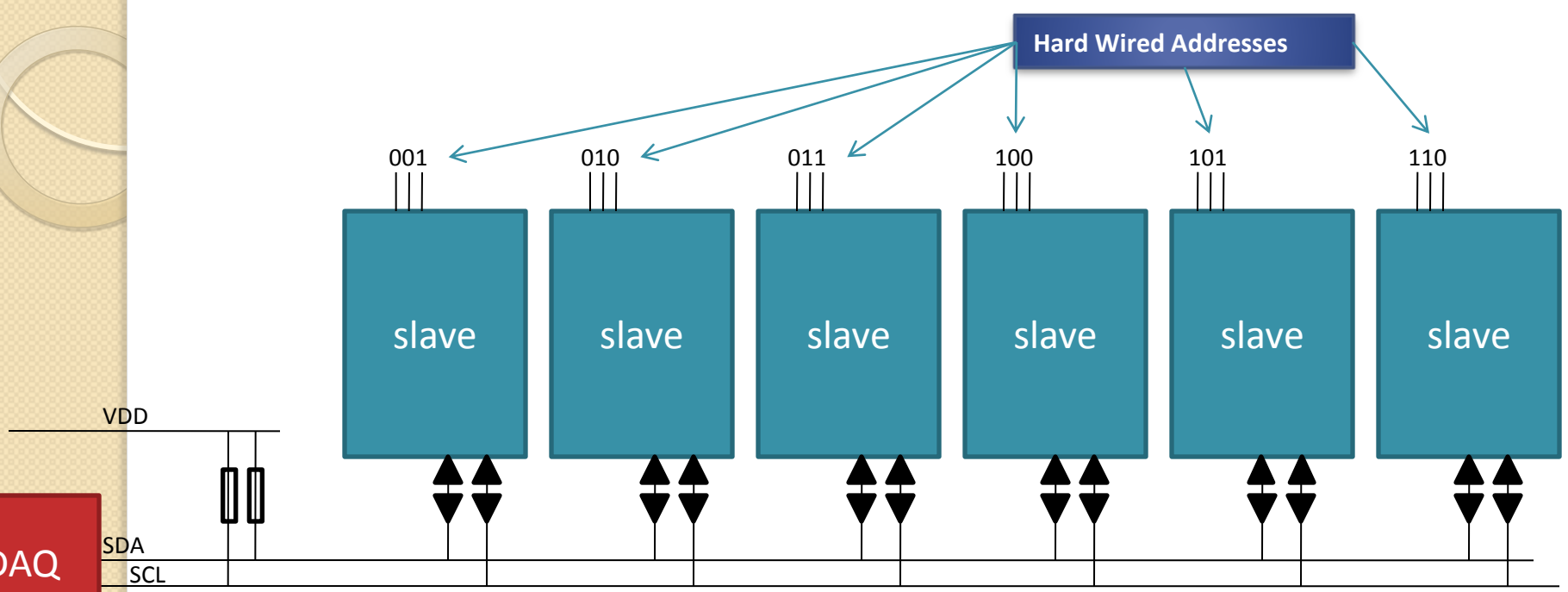
NB: for 200 MHz/cm^2 with Rdclk 80MHz and BC=1 us → efficiency 97.6%

Sub-matrix readout Efficiency table

Hit rate 100 MHz/cm^2

RN	sim DURATION (us)	**RDclk (MHz)**	**BCO (us)**	Mean Sweeping time (us)	global hit rate (MHz)	rate on area (MHz/mm2)	B2 depth	B1 depth	Scan buffer overflow	Already hit effi (%)	**Frozen MP effi (%)**	**Overflow effi B2 (%)**	**Overflow effi B1 (%)**
107	1	**60**	**0,5**	0,45	33,8	1.03	8	32	0	99,96	**98,90**	**100**	**100**
108	1	**80**	**0,5**	0,34	33,8	1.03	8	32	0	99,95	**99,39**	**100**	**100**
109	1	**100**	**0,5**	0,27	33,8	1.03	8	32	0	99,96	**99,53**	**100**	**100**
110	1	**60**	**1**	0,75	33,8	1.03	8	32	0	99,91	**98,83**	**100**	**100**
111	1	**80**	**1**	0,56	33,8	1.03	8	32	0	99,91	**99,10**	**100**	**100**
112	1	**100**	**1**	0,45	33,8	1.03	8	32	0	99,91	**99,25**	**100**	**100**
113	1	**60**	**1,5**	0,95	33,8	1.03	8	32	0	99,86	**98,78**	**100**	**100**
114	1	**80**	**1,5**	0,71	33,8	1.03	8	32	0	99,86	**99,05**	**100**	**100**
115	1	**100**	**1,5**	0,57	33,8	1.03	8	32	0	99,86	**99,23**	**100**	**100**
116	1	**60**	**2**	1,08	33,8	1.03	8	32	0	99,84	**98,42**	**100**	**100**
117	1	**80**	**2**	0,81	33,8	1.03	8	32	0	99,83	**98,81**	**100**	**100**
118	1	**100**	**2**	0,65	33,8	1.03	8	32	0	99,83	**99,04**	**100**	**100**

The Slow Control bus: I²C-like system



Registers R/W access communication type

I²C : two bidirectional open-drain lines.

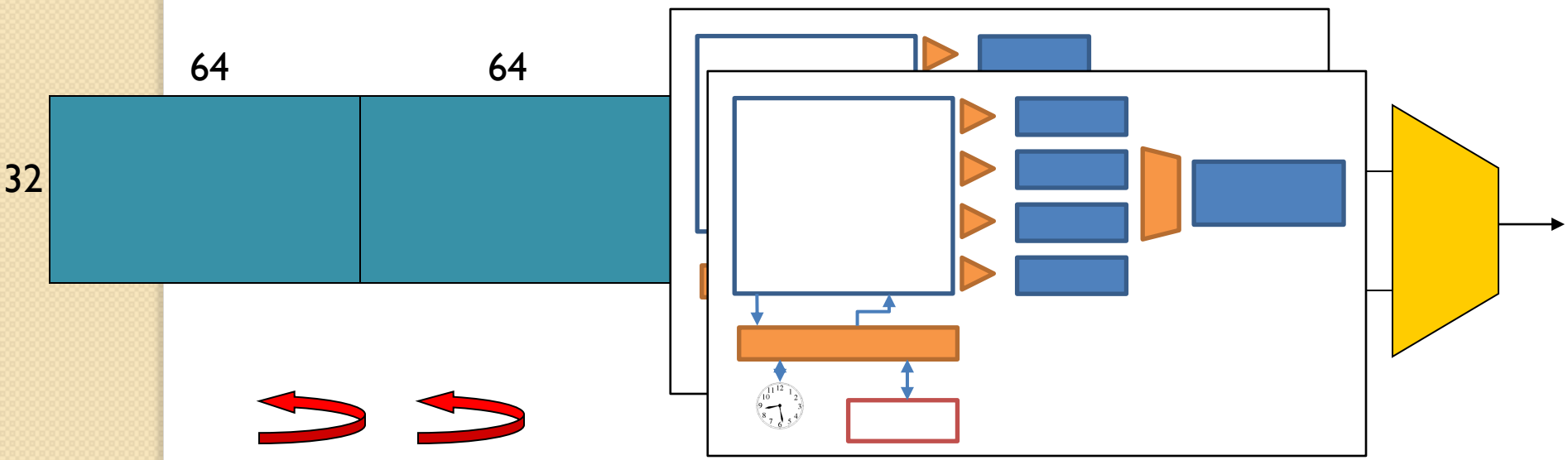
- Serial Data (SDA)
- Serial Clock (SCL), pulled up with resistors.

Slow Control

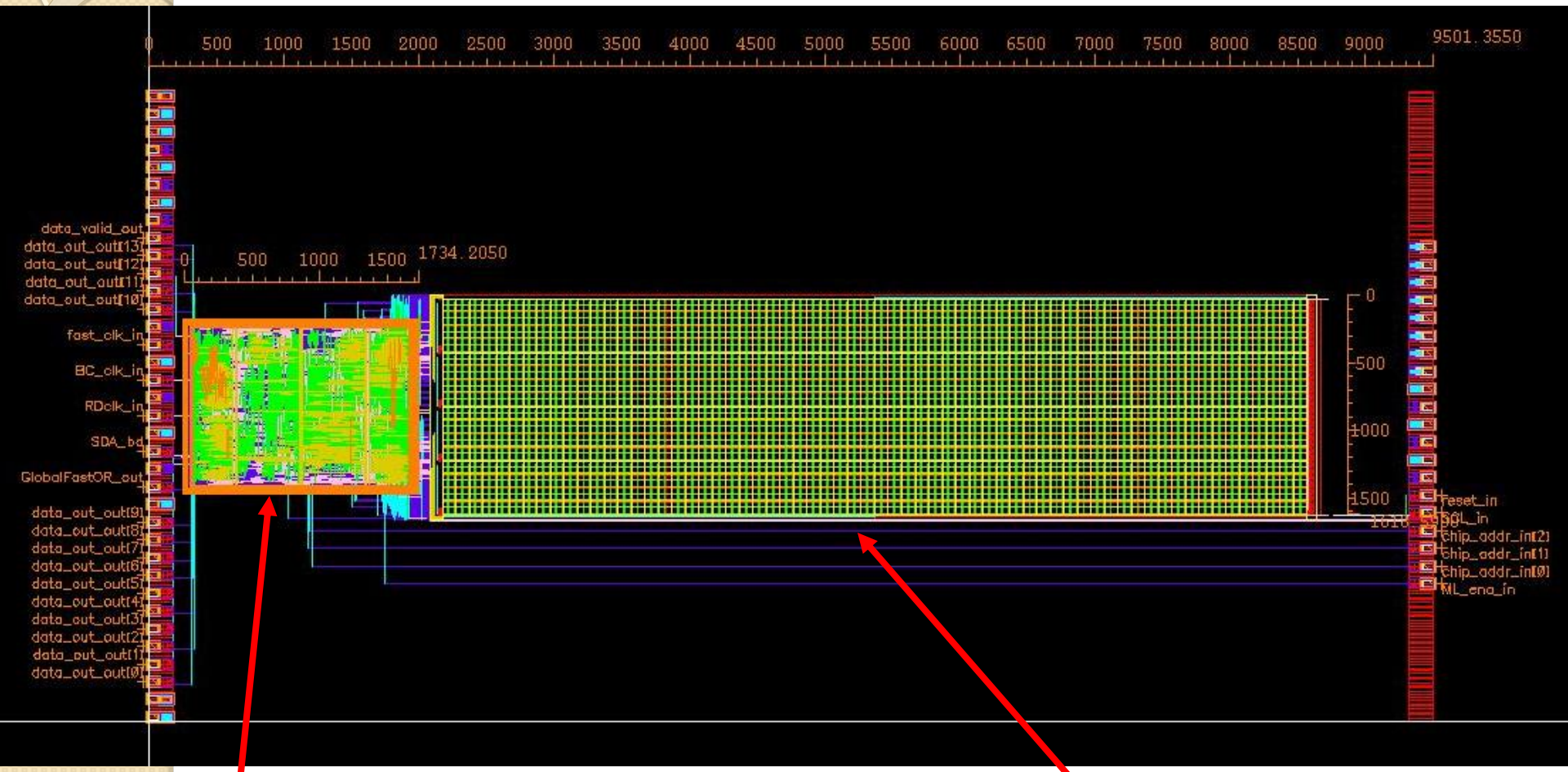
- 1 Set of Read/Write registers
  - Chip settings
  - MP masks
- 1 set of Read Only registers
  - Acquisition flags
  - Rate counters
  - Error flags

The Test Chip

- Submitted Sept. 2009
- Technology STM 130 nm
- Hybrid Pixels Matrix 128x32 pixels, 50 µm pitch (1/20 of the target matrix area)
- Only 2 readout instances implemented
  - The readout instances are oversized respect to the matrix height (32 vs 256 pixels), but the connections implemented allows to stimulate all the components.



The Test Chip layout – ST130nm



Readout

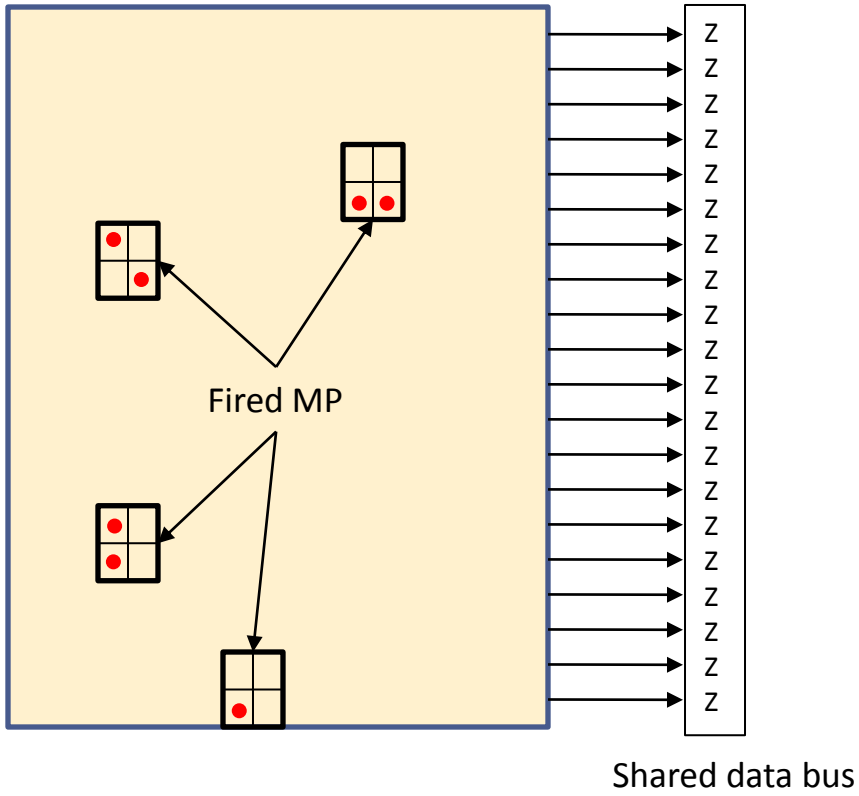
- TOT 58 636 logic cells
- Area < 2 mm^2

Pixel matrix 128x32
Area ~ 10 mm^2

Backups

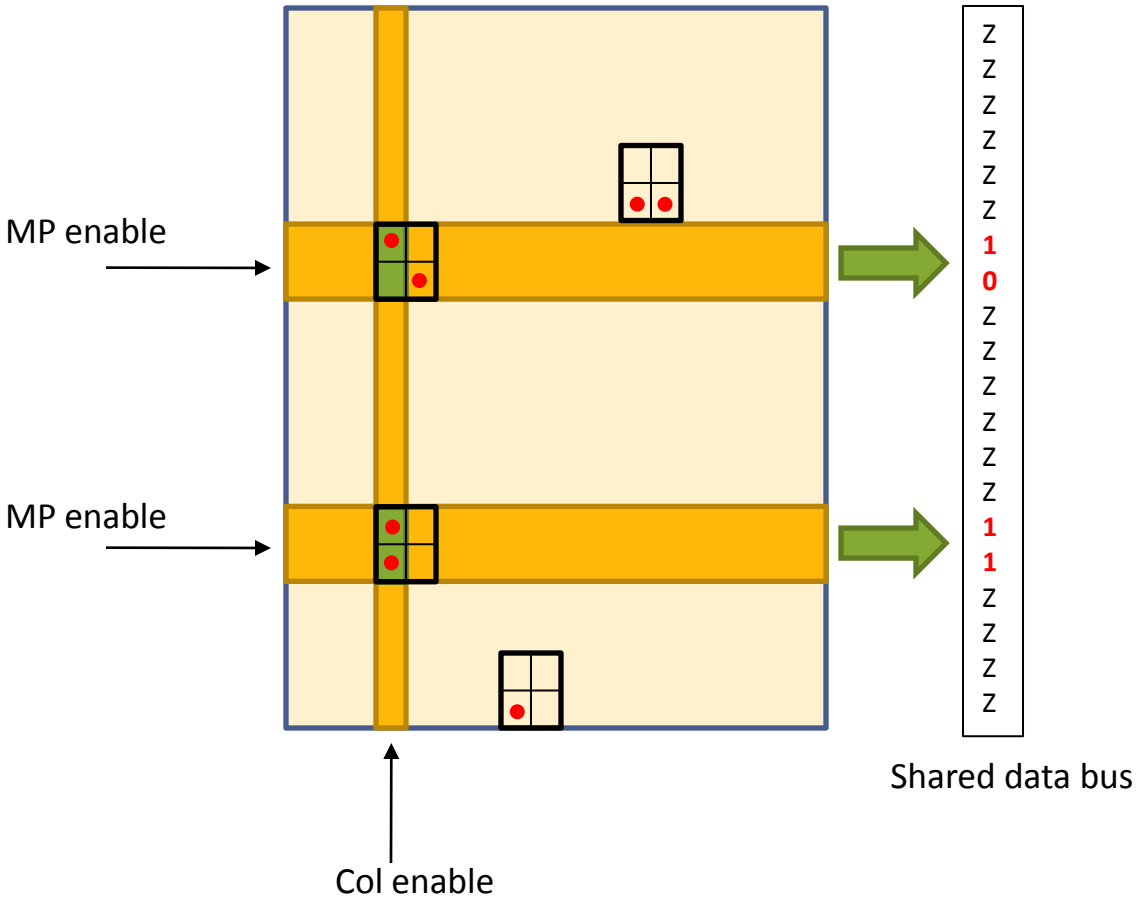
Matrix overview (2)

- Pixels grouped into **Macro Pixels:**
  - Minimum entities addressable by readout logic
  - Minimum entities for time tagging



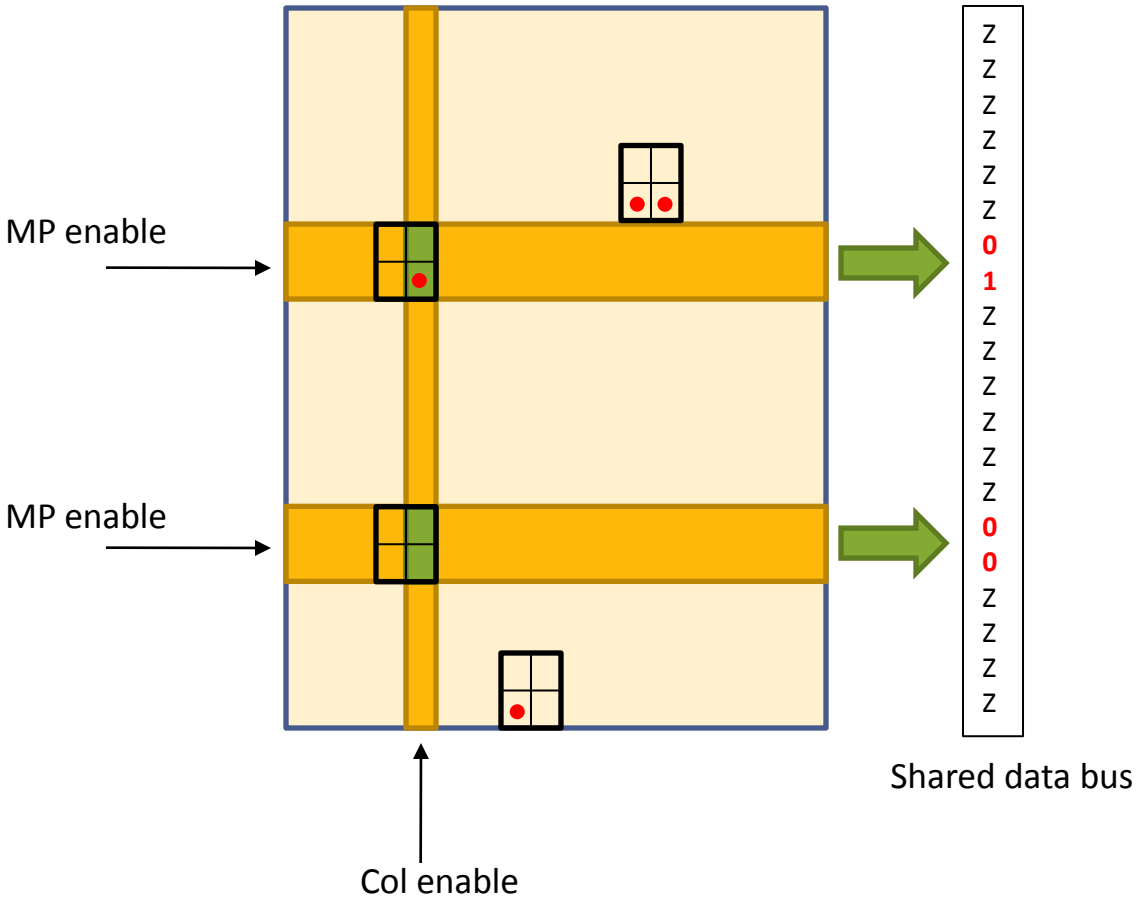
Matrix overview (2)

- Pixels grouped into **Macro Pixels:**
  - Minimum entities addressable by readout logic
  - Minimum entities for time tagging



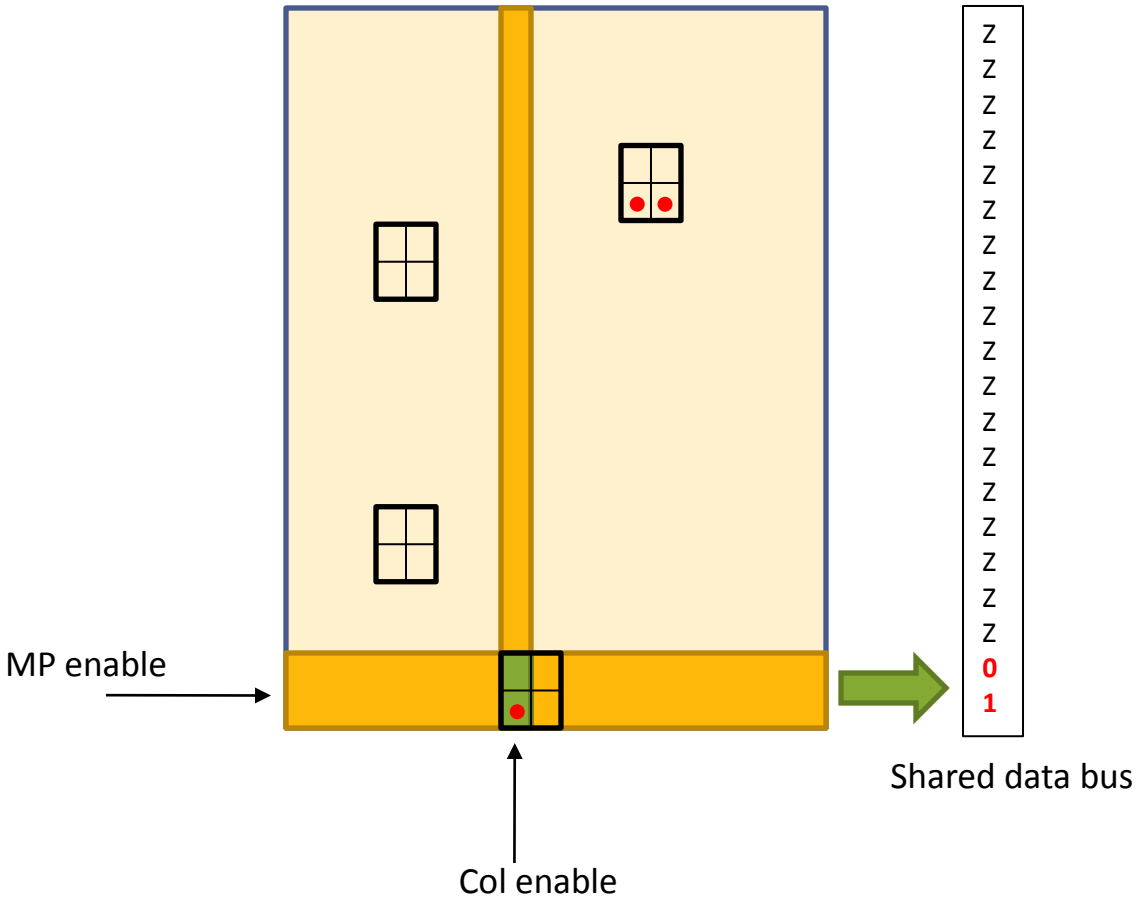
Matrix overview (2)

- Pixels grouped into **Macro Pixels:**
  - Minimum entities addressable by readout logic
  - Minimum entities for time tagging



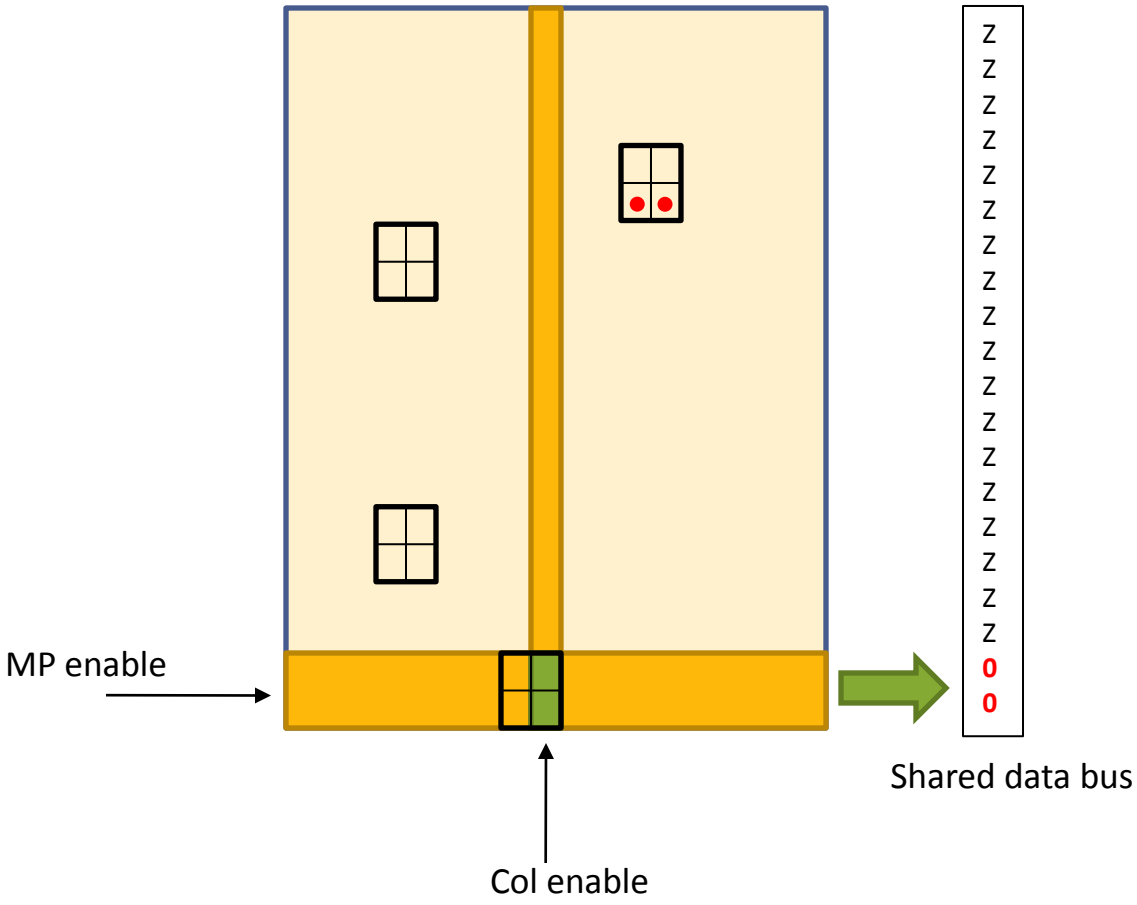
Matrix overview (2)

- Pixels grouped into **Macro Pixels:**
  - Minimum entities addressable by readout logic
  - Minimum entities for time tagging



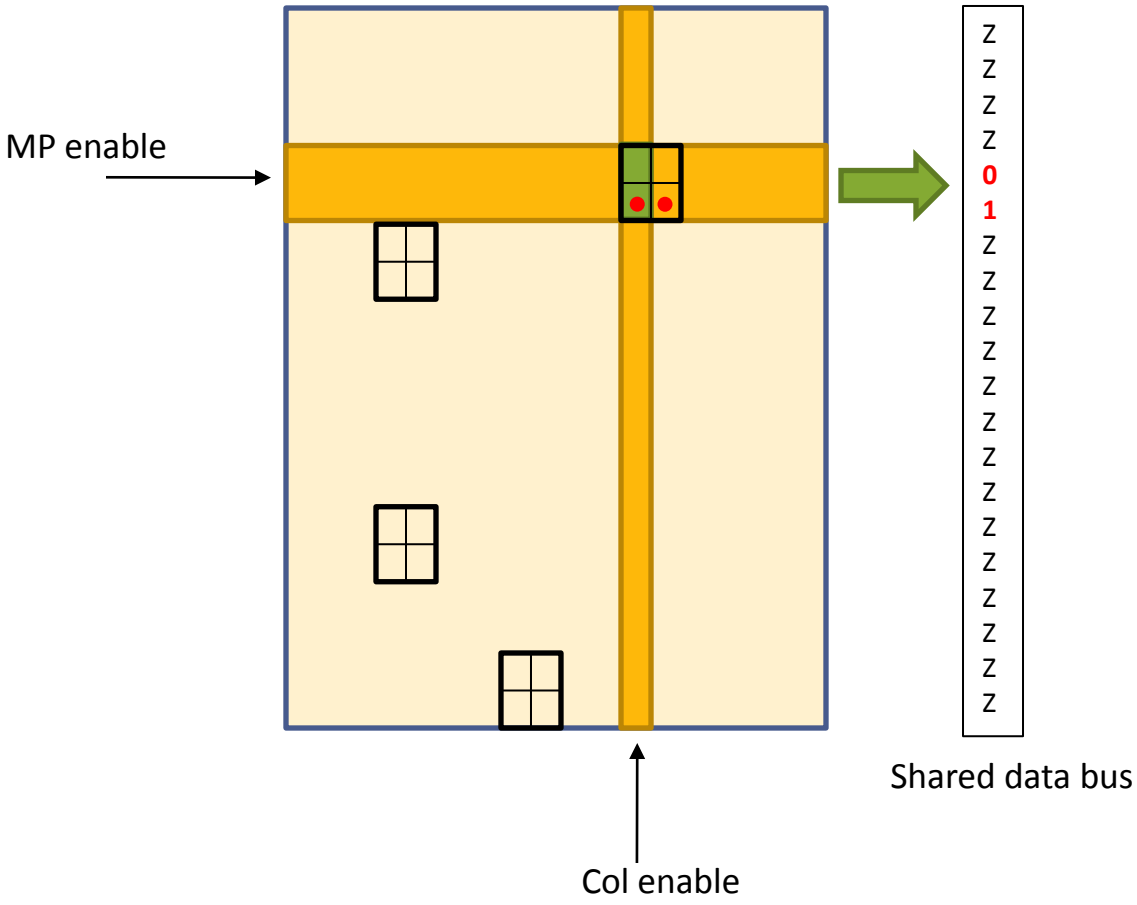
Matrix overview (2)

- Pixels grouped into **Macro Pixels:**
  - Minimum entities addressable by readout logic
  - Minimum entities for time tagging



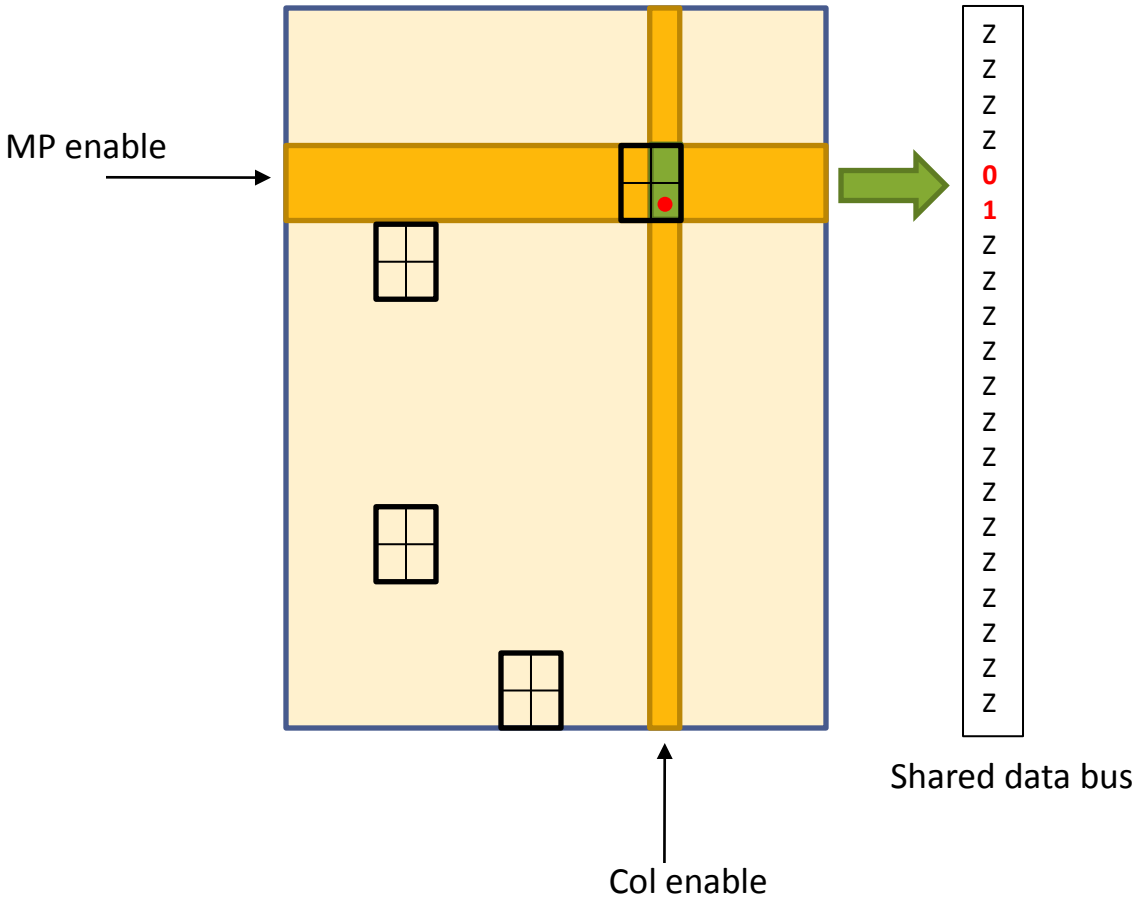
Matrix overview (2)

- Pixels grouped into **Macro Pixels:**
  - Minimum entities addressable by readout logic
  - Minimum entities for time tagging



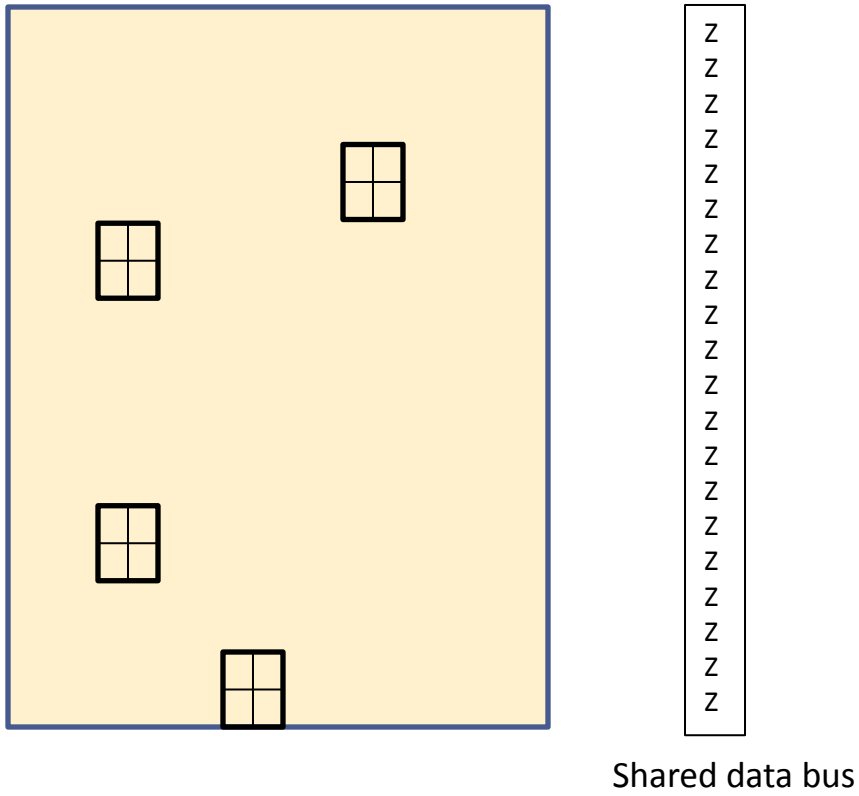
Matrix overview (2)

- Pixels grouped into **Macro Pixels:**
  - Minimum entities addressable by readout logic
  - Minimum entities for time tagging



Matrix overview (2)

- Pixels grouped into **Macro Pixels:**
  - Minimum entities addressable by readout logic
  - Minimum entities for time tagging



Components synthesis

Components	Flip Flop registers	Logic gates
B2	~140	~1400
B1	~1000	~6700
Concentrator	~230	~1000
Concentrator out	~120	~370
I2C interface	~130	~600
Mask register	~520	~2300
Scan buffer	~4700	~9600
Register file	~1000	~2700
Sparsifier	~160	~1000
Sweeper	~7200	~16200