SVT-Pixel layer 0
Readout Architectures

Filippo Maria Giorgi - INFN and University of Bologna

XIII SuperB General Meeting
Isola d’Elba, May 30th - June 5th 2010

Summary

- Boundary target conditions
- Matrix architecture comparisons
- Matrix scan logic
- Sparsification and readout scheme
- Triggered Architecture
- Integration achievements
- Simulations results
- Improvements
- Conclusions

- Rate on Area: **100** MHz/cm^2.
- Matrix area ~ **1.2-1.3** cm^2.
- Pixel pitch ~ **50** µm
- Matrix dimension **256x192** pixels
- Architecture tailored for **hybrid /3DMAPS** sensor
- Output bus bandwidth ~ **20**bit@**200**MHz (**4**Gbps)

Target Conditions

Matrix architectures comparison

Previous matrix architectures (2D MAPS):

- **Simple** in pixel digital logic (competitive N-Well)
- **Time labeling** of hits relayed to **external logic**
- **No hit information** from every **single pixel** (scalability limits)

→ group of 16 pixels: **Macro Pixel (MP)** with 1 single **Fast-OR**

→ **freezing logic** (avoid hits belonging to different Time Windows (BC clock) to populate the same MP)

→ **increase of dead area** proportional to MP area

→ **trade off scalability vs efficiency**

- **Moreover**: time ordered hit extraction from the matrix requires **great** amounts of **memory** to store maps of MPs to be scanned for a determined TS (*Scan Buffer*).

New matrix architecture (Hybrid or 3D MAPS):

- **Dense in pixel digital logic** (Time labeling, arbitrary TS comparator for time ordered readout, auto pixel latch reset…)
- Still no hit information from every single pixel (*same 2D scalability limits*)

→ Column fast-OR **BUT… NOW the TS is at pixel level**

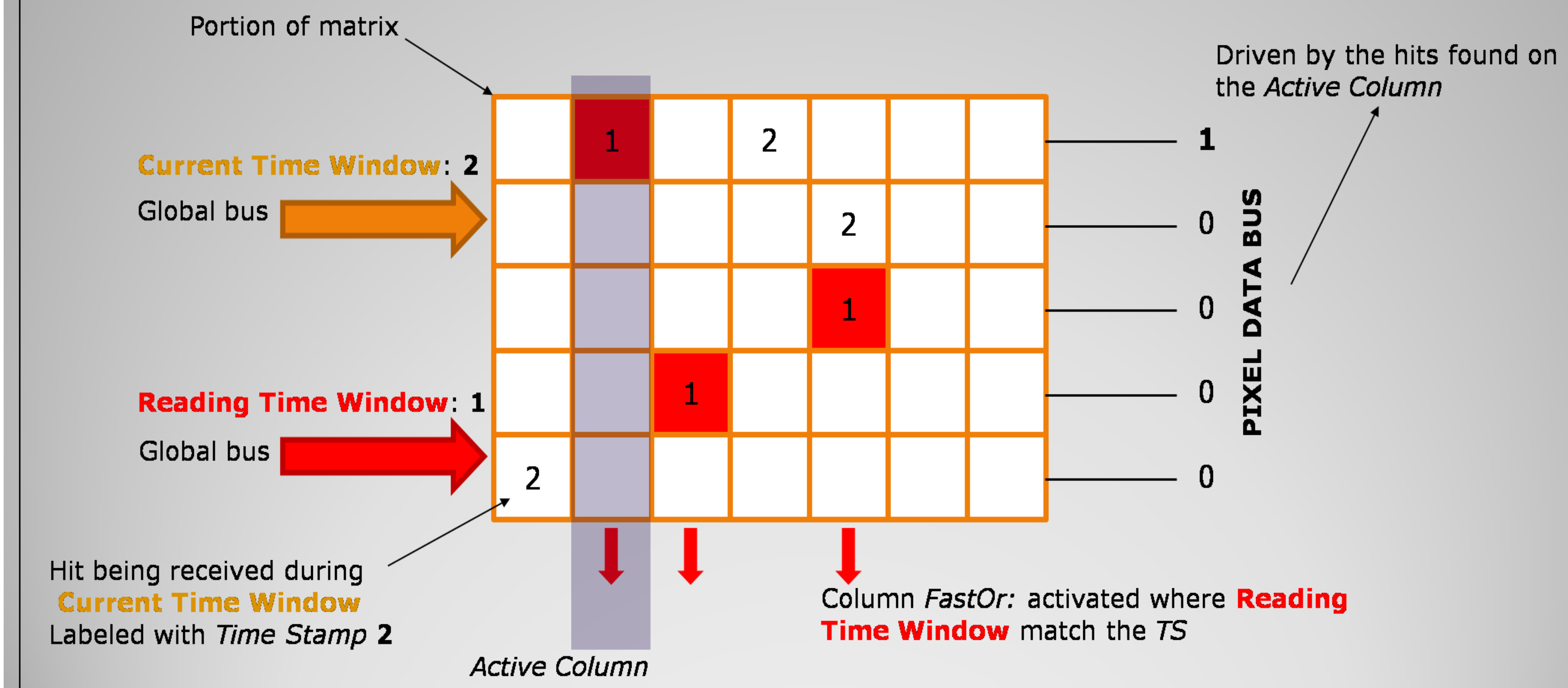
→**NO FREEZING required → much less dead area**
→**NO memory required (*Scan Buffer*)** to perform time ordered matrix scans
→**Smaller BC periods allowed** (no scan buffer overflows, single col. Sweep..)
→**Polyvalent Triggered & Data-push arch. using** MATRIX as buffer element.

Matrix scan Logic

EXAMPLE

During Time Window **2** :

- Some pixels getting fired and labeled with *Time Stamp (TS)* =**2**
- The readout queries the columns containing hits labeled with *TS*=1 (**Reading Time Window** → *FastOr* activation)
- The readout moves the *Active Column* over the columns with an active *FastOr*.



Portion of matrix

Current Time Window: **2**

Global bus

Reading Time Window: **1**

Global bus

Hit being received during **Current Time Window** Labeled with *Time Stamp* **2**

Active Column

Column *FastOr:* activated where **Reading Time Window** match the *TS*

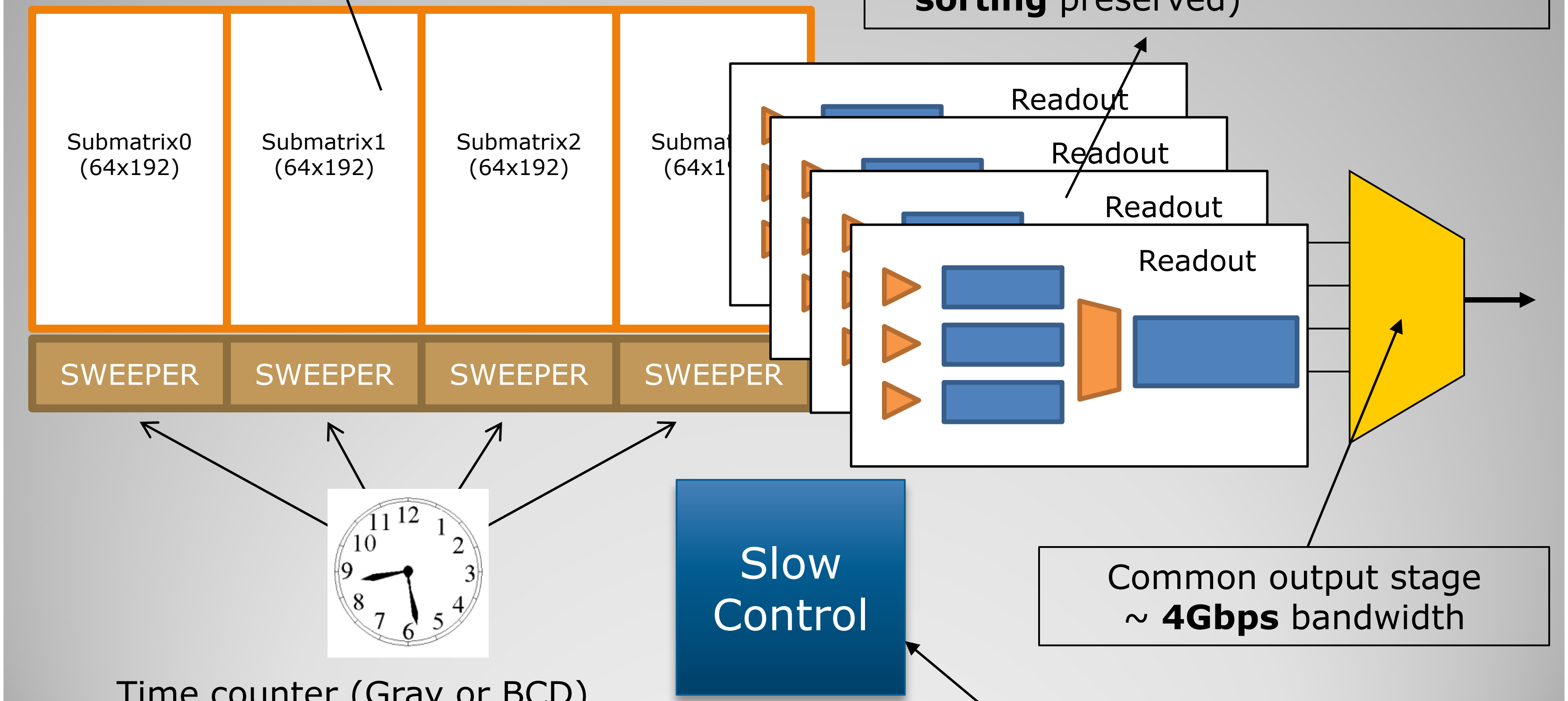
Driven by the hits found on the *Active Column*

Readout Scheme

- Pixel organized into **4 sub-matrices**
- **Each** sub-matrix has an **independent** scan logic

→Increase horizontal parallelization
→The shorter the scans the greater the effi.

- One readout for **each** sub-matrix
- Vertical parallel sparsification (**one entire column** per clock cycle)
- Hit encoding
- Hit de-queuing system (**time sorting** preserved)



Time counter (Gray or BCD)

Common output stage
~ **4Gbps** bandwidth

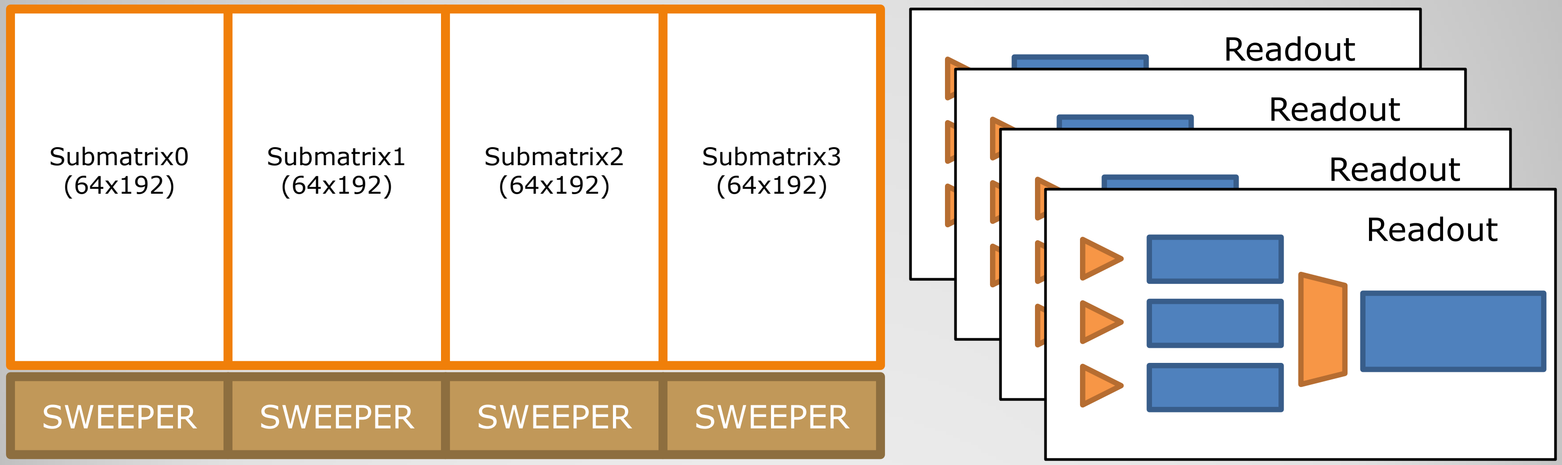
I²C-like external interface:
- 2 pad per chip
- 2 lines for entire module

Triggered Architecture

A pixel data push architecture for Layer0 requires a lot of available bandwidth. (all data must be sent)

Some modifications, involving the sweeper architecture only, make possible to exploit the matrix itself as a hit buffer for a triggered architecture.

This is made possible by the low trigger latency (few us). Efficiency should not drop drastically.



Integration achievements

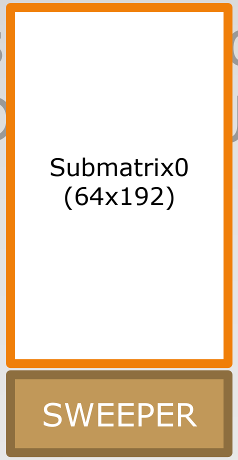
- All the readout architecture coded in **synthsizable VHDL**.
- **Sweeper** for new matrix architecture rewritten **from scratch**.
  - Work in progress for the modifications that allow a **triggerable architecture**.
- **Full architecture entirely integrated** reusing the same readout components from ***SuperPX0*** alias *FE4D32x128*.
- We want to recycle as much as possible of them:
  - Sparsification algorithms (zone sparsification)
  - Barrel architecture (dynamic asymmetric FIFOs: variable input width)
  - Concentrators with time sorting preserving algorithms.

1. High statistic simulations with **Matrix** and **Sweeper** ONLY (DATA-PUSH):
   - The evaluated inefficiency depends **only** on how long it takes to extract a hit from the matrix.
   - No readout → no readout bottlenecks taken into account.
2. High statistic simulations of the whole architecture (DATA-PUSH):
   - **New matrix**
   - **New sweeper**
   - **"OLD" SuperPX0 readout AS IS** (sparsification and de-queuing logic).

Simulations overview

Simulations overview

1. High statistic simulations with **Matrix** and **Sweeper** ONLY (DATA-PUSH):
   - The evaluated inefficiency depends **only** on how long it takes to extract a hit from the matrix.
   - No readout → no readout bottlenecks taken into account.
2. High statistic s ons of the whole architecture (D USH):
   - **New matrix**
   - **New sweeper**
   - **"OLD" SuperPX0** **AS IS** (sparsification and de-queuing logic).



1. Matrix+Sweeper simulations

Linear BC span

	100	150	200	250	300	350	400	450	500	1000	1500	2000	BC (ns)
10	99,97	99,95	99,95	99,93	99,91	99,90	99,88	99,87	99,87	99,78	99,70	99,64	
12	99,96	99,95	99,95	99,92	99,91	99,89	99,88	99,86	99,86	99,77	99,70	99,63	
15	99,95	99,95	99,94	99,91	99,91	99,89	99,87	99,86	99,85	99,76	99,68	99,61	
18	99,93	99,93	99,92	99,91	99,89	99,88	99,86	99,86	99,85	99,74	99,67	99,59	
20	95,40	99,91	99,91	99,90	99,89	99,87	99,86	99,85	99,84	99,73	99,66	99,58	
22	94,33	99,07	99,89	99,89	99,88	99,87	99,85	99,84	99,84	99,73	99,65	99,57	
25	93,95	92,73	95,86	99,72	99,85	99,85	99,84	99,83	99,83	99,72	99,63	99,55	
30	89,31	89,20	88,78	87,92	88,07	91,18	93,89	96,34	98,70	99,69	99,61	99,51	
RDclk(ns)													

Mean Sweeping Time (MST) > BC

Since we perform an independent sweep for each BC period, this is an **UNAFFORDABLE WORKING CONDITION**

Respect to previous matrix architectures :

- **Wider margin** on MST>BC condition (no scan buffer)
- **Higher efficiencies** (no freezing)

NO sensor efficiency
NO pixel reset dead time
ONLY SWEEPING DEAD TIME

Linear BC span

	100	150	200	250	300	350	400	450	500	1000	1500	2000	BC (ns)
10	99,97	99,95	99,95	99,93	99,91	99,90	99,88	99,87	99,87	99,78	99,70	99,64	
12	99,96	99,95	99,95	99,92	99,91	99,89	99,88	99,86	99,86	99,77	99,70	99,63	
15	99,95	99,95	99,94	99,91	99,91	99,89	99,87	99,86	99,85	99,76	99,68	99,61	
18	99,93	99,93	99,92	99,91	99,89	99,88	99,86	99,86	99,85	99,74	99,67	99,59	
20	**95,40**	99,91	99,91	99,90	99,89	99,87	99,86	99,85	99,84	99,73	99,66	99,58	
22	**94,33**	**99,07**	99,89	99,89	99,88	99,87	99,85	99,84	99,84	99,73	99,65	99,57	
25	**93,95**	**92,73**	**95,86**	99,72	99,85	99,85	99,84	99,83	99,83	99,72	99,63	99,55	
30	**89,31**	**89,20**	**88,78**	**87,92**	**88,07**	**91,18**	**93,89**	**96,34**	**98,70**	99,69	99,61	99,51	
RDclk(ns)													

Mean Sweeping Time (MST) > BC

Since we perform an independent sweep for each BC period, this is an **UNAFFORDABLE WORKING CONDITION**

Respect to previous matrix architectures :

- **Wider margin** on MST>BC condition (no scan buffer)
- **Higher efficiencies** (no freezing)

NO sensor efficiency
NO pixel reset dead time
ONLY SWEEPING DEAD TIME

1. Matrix+Sweeper simulations

1. Matrix+Sweeper simulations

Linear BC span

	100	150	200	250	300	350	400	450	500	1000	1500	2000	BC (ns)
10	**99,97**	99,95	99,95	99,93	99,91	99,90	99,88	99,87	99,87	99,78	99,70	99,64	
12	99,96	99,95	99,95	99,92	99,91	99,89	99,88	99,86	99,86	99,77	99,70	99,63	
15	99,95	99,95	99,94	99,91	99,91	99,89	99,87	99,86	99,85	99,76	99,68	99,61	
18	99,93	99,93	99,92	99,91	99,89	99,88	99,86	99,86	99,85	99,74	99,67	99,59	
20	**95,40**	99,91	99,91	99,90	99,89	99,87	99,86	99,85	99,84	99,73	99,66	99,58	
22	**94,33**	**99,07**	99,89	99,89	99,88	99,87	99,85	99,84	99,84	99,73	99,65	99,57	
25	**93,95**	**92,73**	**95,86**	99,72	99,85	99,85	99,84	99,83	99,83	99,72	99,63	99,55	
30	**89,31**	**89,20**	**88,78**	**87,92**	**88,07**	**91,18**	**93,89**	**96,34**	**98,70**	99,69	99,61	**99,51**	
RDclk(ns)													

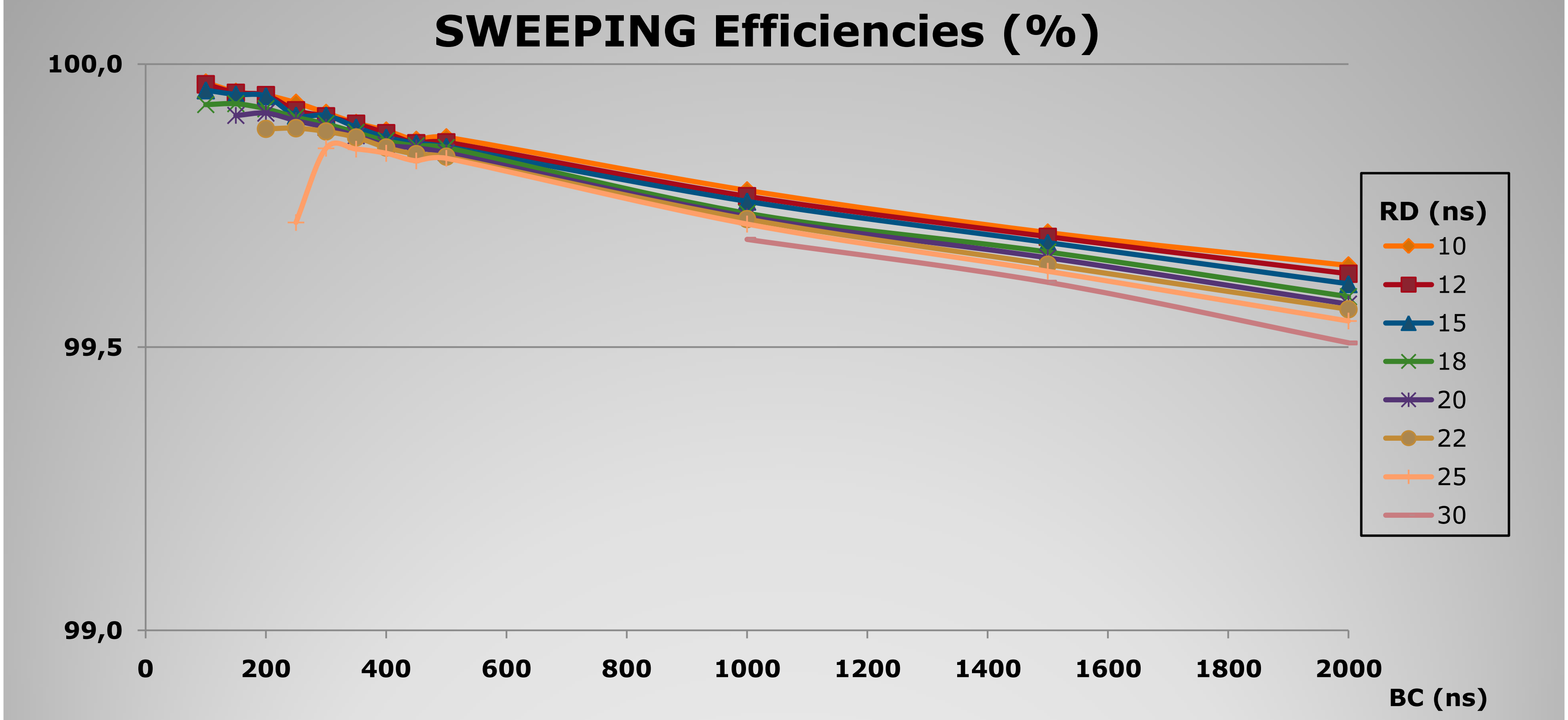
Mean Sweeping Time (MST) > BC

Since we perform an independent sweep for each BC period, this is an **UNAFFORDABLE WORKING CONDITION**

Respect to previous matrix architectures :

- **Wider margin** on MST>BC condition (no scan buffer)
- **Higher efficiencies** (no freezing)

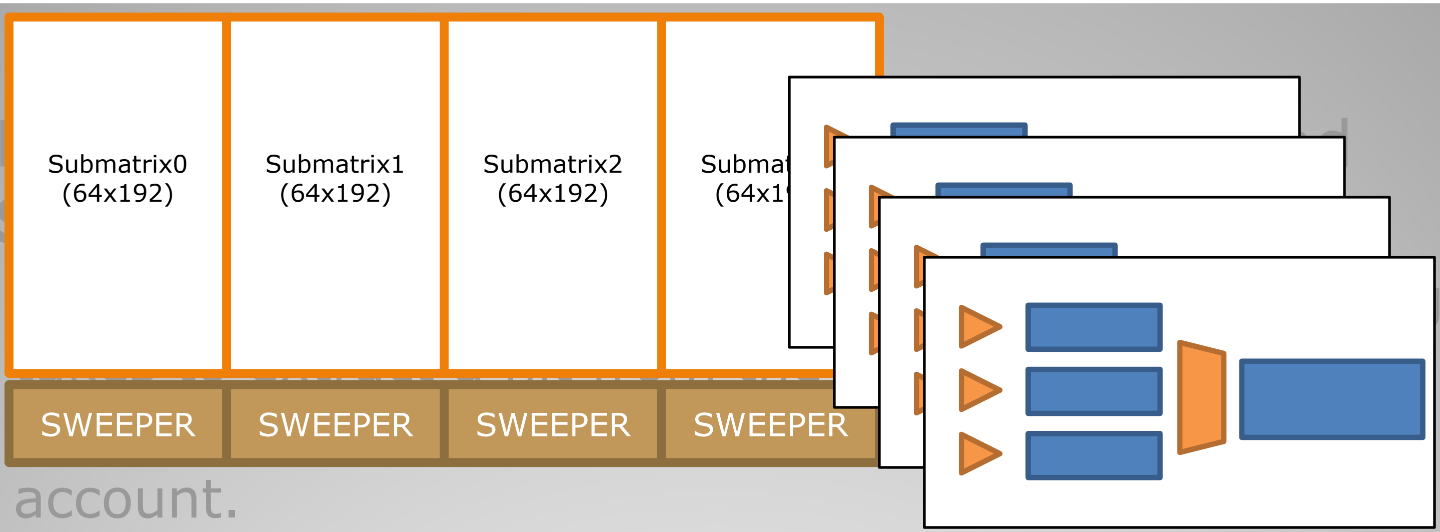
NO sensor efficiency
NO pixel reset dead time
ONLY SWEEPING DEAD TIME



NO MST>BC points plotted

1. Matrix+Sweeper simulations

1.



account.

2. High statistic simulations of the whole architecture (DATA-PUSH):
   - **New matrix**
   - **New sweeper**
   - **“OLD” SuperPX0 readout AS IS** (sparsification and de-queuing logic).

Simulation overview

Efficiency results:

Compare with *SuperPX0* data push arch.

Improvements mainly due to:

- Reduced pixel dead time (no x16 factor due to MP freezed area)
- No more *Scan Buffer* overflows

Effi.(%)

BC (ns)

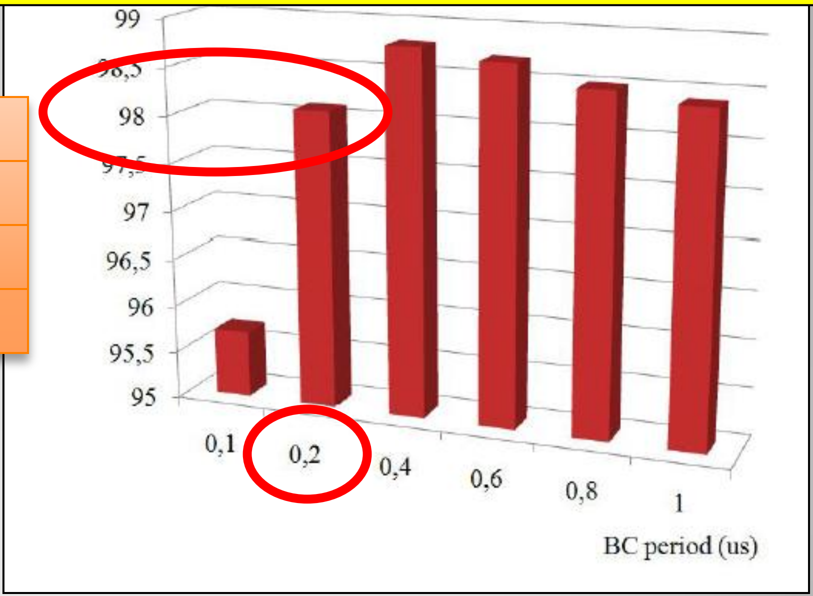
RD (MHz)	**200**	**250**	**300**
66.7	99.93	99.92	99.92
55.6	99.93	99.92	99.91
50	**99.92**	99.92	99.90

Again **NOT** taken into account:

- sensor efficiency (assumed 100%)
- pixel reset dead time (assumed few ns)

- **Consistent** with sweeper+matrix only simulations
- **Readout de-queuing efficiency 100%** (no barrel overflows)
- **Hit check** results: **100 % match**.
- Fast_clock **4 x** RDclk (output bus frequency)

SUPERPX0 - RDclk **66.67 MHz** – Fast_clk **200 MHz (3x)**



efficiency results from similar simulations of *SuperPX0* readout

2. Full architecture simulations

Improvements….

Simulations shows that

for **even smaller BC period (150 →100 ns):**

→ Time sorting de-queuing algorithm suddenly slows down.
(more time windows to manage → more complexity)

→ *Barrel* overflow more frequent.

Steps already taken to reach the BC=100 ns working point:

REINFORCEMENT for critical components (barrels, concentrators…)

IMPROVEMENTS and **OPTIMIZATION** in other areas

Good chances to reach **100 ns BC** with few modifications.

Anyway consider that:

- The Rate 100Mhz/cm^2 we are trying to sustain should include a x4 cluster factor.
- The architecture is strongly optimized for clustered events
- In the simulations shown hit dispersion is UNIFORM → NO clusters (rate increased x4 "for free" with no cluster benefits)

AND Remember that with **triggered architecture** reaching 100ns of time resolution is no longer an issue.

- New sweeper logic implemented for DATA-PUSH.
  - Under development the triggered sweeper.
- Sweeper connected to an improved SuperPX0 readout.
  - Simulations showed excellent results down to 200 ns of BC even recycling SuperPX0 de-queuing system *AS IS* (it was designed for BC down to 1 us).
- Further improvements under investigation in order to reach even smaller time windows and wider margins on *Barrel* overflows.
- New functional simulations in sight for the triggered operating mode of the sweeper. (followed by efficiency estimations)

Conclusions