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
• 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/cm².
- Matrix area ~ 1.2-1.3 cm².
- Pixel pitch ~ 50 μm
- Matrix dimension 256x192 pixels
- Architecture tailored for hybrid /3DMAPS sensor
- Output bus bandwidth ~ 20bit@200MHz (4Gbps)
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.
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**.

**Matrix scan Logic**
- 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.

<table>
<thead>
<tr>
<th>Submatrix0</th>
<th>Submatrix1</th>
<th>Submatrix2</th>
<th>Submatrix3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(64x192)</td>
<td>(64x192)</td>
<td>(64x192)</td>
<td>(64x1)</td>
</tr>
</tbody>
</table>

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

- **Slow Control**

- **Common output stage**
  - ~ **4Gbps** bandwidth

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

**Readout Scheme**
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.
• All the readout architecture coded in **synthysizeable 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-queueing logic).
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 $\rightarrow$ no readout bottlenecks taken into account.

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

<table>
<thead>
<tr>
<th>RDclk(ns)</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
<th>300</th>
<th>350</th>
<th>400</th>
<th>450</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>89.31</td>
<td>89.20</td>
<td>88.78</td>
<td>87.92</td>
<td>88.07</td>
<td>91.18</td>
<td>93.89</td>
<td>96.34</td>
<td>98.70</td>
<td>99.69</td>
<td>99.61</td>
<td>99.51</td>
</tr>
</tbody>
</table>

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

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

Mean Sweeping Time (MST) > BC
### 1. Matrix+Sweeper simulations

#### Linear BC span

<table>
<thead>
<tr>
<th>RDclk(ns)</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
<th>300</th>
<th>350</th>
<th>400</th>
<th>450</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
<th>BC (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td><strong>93.95</strong></td>
<td><strong>92.73</strong></td>
<td><strong>95.86</strong></td>
<td>99.72</td>
<td>99.85</td>
<td>99.85</td>
<td>99.84</td>
<td>99.83</td>
<td>99.83</td>
<td>99.72</td>
<td>99.63</td>
<td>99.55</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td><strong>89.31</strong></td>
<td><strong>89.20</strong></td>
<td><strong>88.78</strong></td>
<td><strong>87.92</strong></td>
<td><strong>88.07</strong></td>
<td><strong>91.18</strong></td>
<td><strong>93.89</strong></td>
<td><strong>96.34</strong></td>
<td><strong>98.70</strong></td>
<td>99.69</td>
<td>99.61</td>
<td>99.51</td>
<td></td>
</tr>
</tbody>
</table>

#### Mean Sweeping Time (MST) > BC

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

- **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

Respect to previous matrix architectures:

1. **Wider margin** on MST>BC condition (no scan buffer)
2. **Higher efficiencies** (no freezing)
### Linear BC span

<table>
<thead>
<tr>
<th>RDclk(ns)</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
<th>300</th>
<th>350</th>
<th>400</th>
<th>450</th>
<th>500</th>
<th>1000</th>
<th>1500</th>
<th>2000</th>
<th>BC (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>89,31</td>
<td>89,20</td>
<td>88,78</td>
<td>87,92</td>
<td>88,07</td>
<td>91,18</td>
<td>93,89</td>
<td>96,34</td>
<td>98,70</td>
<td>99,69</td>
<td>99,61</td>
<td>99,51</td>
<td></td>
</tr>
</tbody>
</table>

**Mean Sweeping Time (MST) > BC**

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

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

**1. Matrix+Sweeper simulations**
1. Matrix + Sweeper simulations

NO MST > BC points plotted
Simulation 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 simulations of the whole architecture (DATA-PUSH):
   - New matrix
   - New sweeper
   - “OLD” SuperPX0 readout AS IS (sparsification and de-queuing logic).
2. Full architecture simulations

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

<table>
<thead>
<tr>
<th>RD (MHz)</th>
<th>200</th>
<th>250</th>
<th>300</th>
</tr>
</thead>
<tbody>
<tr>
<td>66.7</td>
<td>99.93</td>
<td>99.92</td>
<td>99.92</td>
</tr>
<tr>
<td>55.6</td>
<td>99.93</td>
<td>99.92</td>
<td>99.91</td>
</tr>
<tr>
<td>50</td>
<td>99.92</td>
<td>99.92</td>
<td>99.90</td>
</tr>
</tbody>
</table>

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
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/cm² 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.
Conclusions

- 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)