Slide 1

\*\*\*

Variant 1

The pixel contains the charge sensitive amplifier and the analog buffer (source follower).

Every pixel has its digital pixel cell at the periphery

The periphery contains

**CR filter**

Poly capacitor and an NMOS as resistance

(This circuit has been successfully implemented on HVStripV1)

**Comparator**

Can be based on the comparator used in HVStripV1

**Tune DAC**

We have tested several tune DACs at HVCMOS chips done in Heidelberg/KIT

We need a PMOS TDAC for better radiation tolerance. Such a PMOS-based TDAC was used in HVPixel2 chip. We need a layout modification (see below)

**Edge detector (diff)**

Based on the design used in mu3e chip

**Synchronizer**

SR latch, D-latch, OR

This block generates a pulse ResB synchronized with the Ck=0 after the hit.

**Address ROM and Hit bit**

ResB connects address ROM and Hit Bit to the bus

**Logic Block**

The logic block generates overflow signal and will be presented by Herve.

Very similar scheme has been implemented on HVStripV1 chip and on Mu3e chip.

The height of the digital cell can be 21u the width 80um

The vertical signal lines for two segmented strips are placed above the part with comparator, TDAC and CR filter

We would need two metal layers for the lines (M2 and M3). The internal connections must be done mainly with M1.

There are no analog lines above the CMOS logic, which reduces crosstalk.

Slide 2

\*\*\*

The slide shows the floor-plan or the segmented strips and the digital cells

Part C denotes comparator and TDAC. Part L denotes the remaining logic blocks.

Slide 3

\*\*\*

Here a modified scheme is shown. The differential pair of the comparator, the TDAC and the CR filter are placed in the pixel itself.

This simplifies the periphery – we do not need analog components on the periphery. Only a current-source CS (that biases the differential pair (comparator)) and a simple low-voltage to CMOS convertor are needed. This convertor occupies much smaller area than 40um x 21um – it can be easily done only with M1.

Slide 4

\*\*\*

The modified scheme would probably allow a smaller digital pixel part.

Slide 5

\*\*\*

There will be 128 groups of 32 digital cells. In other words, we have 128 strixels.

For each strixel we have one “Strixel RO cell”.

We will group the Strixel RO cells in 8 groups – each 16 of them.

The purpose of all the RO cells is to multiplex the strixel data to one of the eight output busses Add0 – Add7. Each bus is 13 bit wide.

The working principle is as follows:

For every Strixel RO cell ( “cell”) there is a local sum.

If there is a hit in a cell the local sum is increased.

The hit data are then sent to the data bus that corresponds to the local sum.

There are eight such data buses, to cope with eight simultaneous hits.

In the simplest realization, we would need a chain of 128 3-bit adders. The sum-calculation would take in this case longer than 25ns. To increase the speed, we calculate the total sum in two stages (“pipeline”) – the first for every 16 cell group and then for 8 groups.

Slide 6

\*\*\*

The principle or the data multiplexing and pipeline is shown in slide 6.

**First clock period**

The 5bit pixel address, the overflow bit and the hit flag coming from the strixels are stored into flip flops of the first stage.

The adders (a) after the first stage calculate output = input + 1 if there is a hit in this RO cell. The adders (a) calculate the sum for the local 16-cell group.

**Second clock period**

The hit data and the local sum are stored in flip flips (stage 2) for further processing (pipeline).

Adder (b) calculates the sum of all local adder chain outputs. There are 8 adders (b) one at the output of every cell group.

Adder (c) adds the local sum (stored in flip flop) to the global sum.

**Third clock period**

The final result of the addition and the hit data are stored in flip flips (stage 3). The stored adder output diverts the stored pixel address, the overflow bit and the strixel address (ROM) to one of eight busses. Hit bit enables the data transfer to the bus.

The whole readout chain has been implemented as schematic and tested using analog simulator.

In the tested implementation, the flip flops were realizes as dynamic flip flops.

The advantage of the dynamic flip flops is that they are smaller and rely mostly on PMOS transistors. There are only two NMOS transistors per flip flop. NMOSTs can be made enclosed. This improves radiation tolerance.

However, we are planing to implement the first version of the digital block with the “normal” standard-cell flip-flops (for safety reasons). If there is time, we might implement a smaller block with dynamic flip-flops.