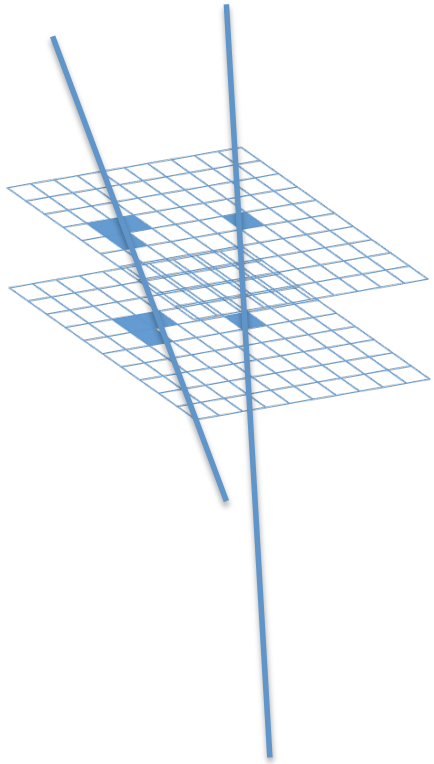
Architecture of a module with transverse momentum discrimination for the CMS Tracker Upgrade

A. Marchioro / CERN

WIT2010 - February 2010



Collaborating in this work

- CERN
  - D. Abbaneo
  - P. Aspell
  - D. Felici
  - L. Pierobon
  - A. Conde
- IC
  - G. Hall
  - M. Raymond
  - M. Pesaresi
- Cornell University
  - A. Ryd
  - L. Fields

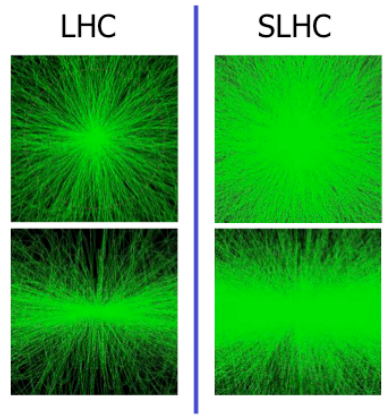
Outline

- Overview of the proposed architecture
- Details of module interconnect scheme
- Logic for Data Movement
  and Trigger Primitives
- The hardware simulation environment
- Some circuit details
- Conclusion

History of the concept in CMS

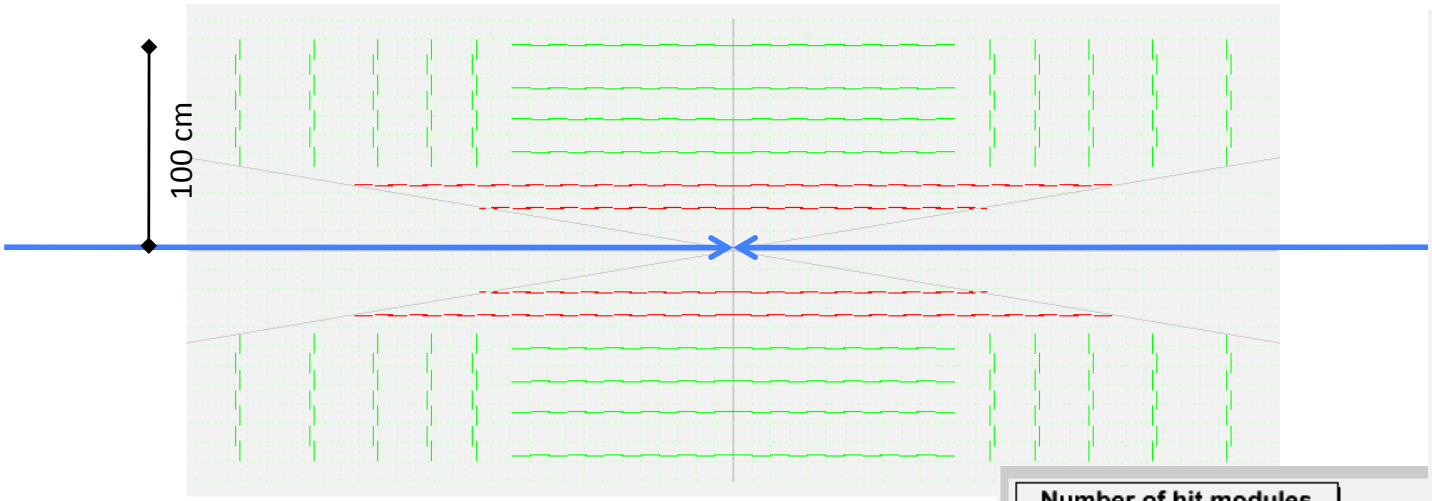
LECC 2005 Heidelberg
September 16th, 2005

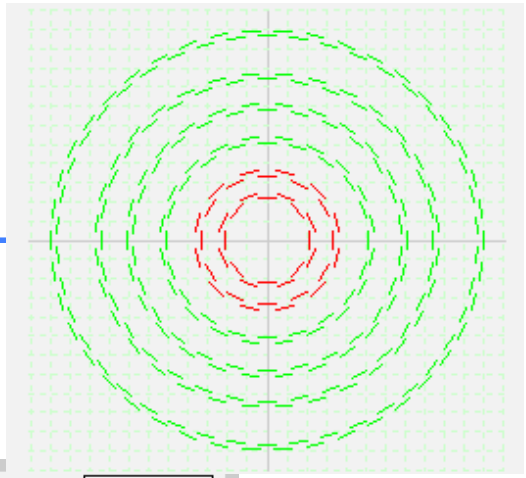
A Pixel Detector for Level-1 Triggering at SLHC

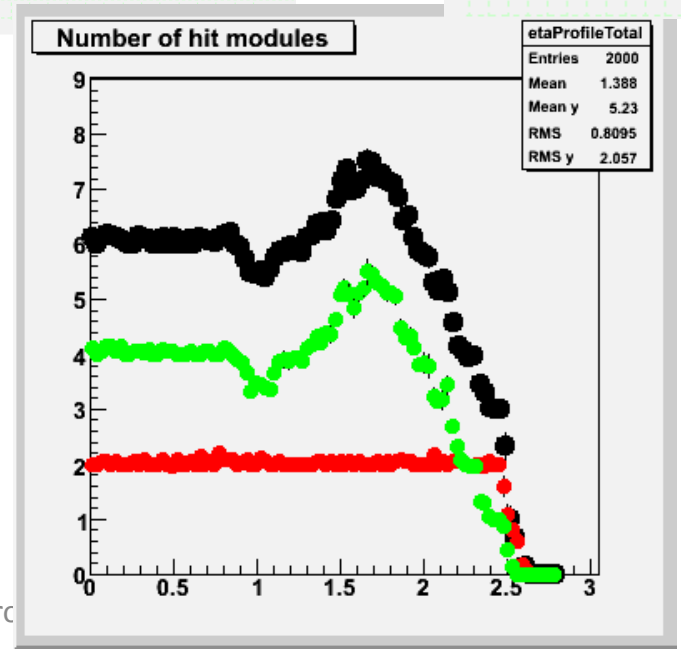


J. Jones (john.jones@imperial.ac.uk)
G. Hall, C. Foudas, A. Rose
Imperial College London

Layout model



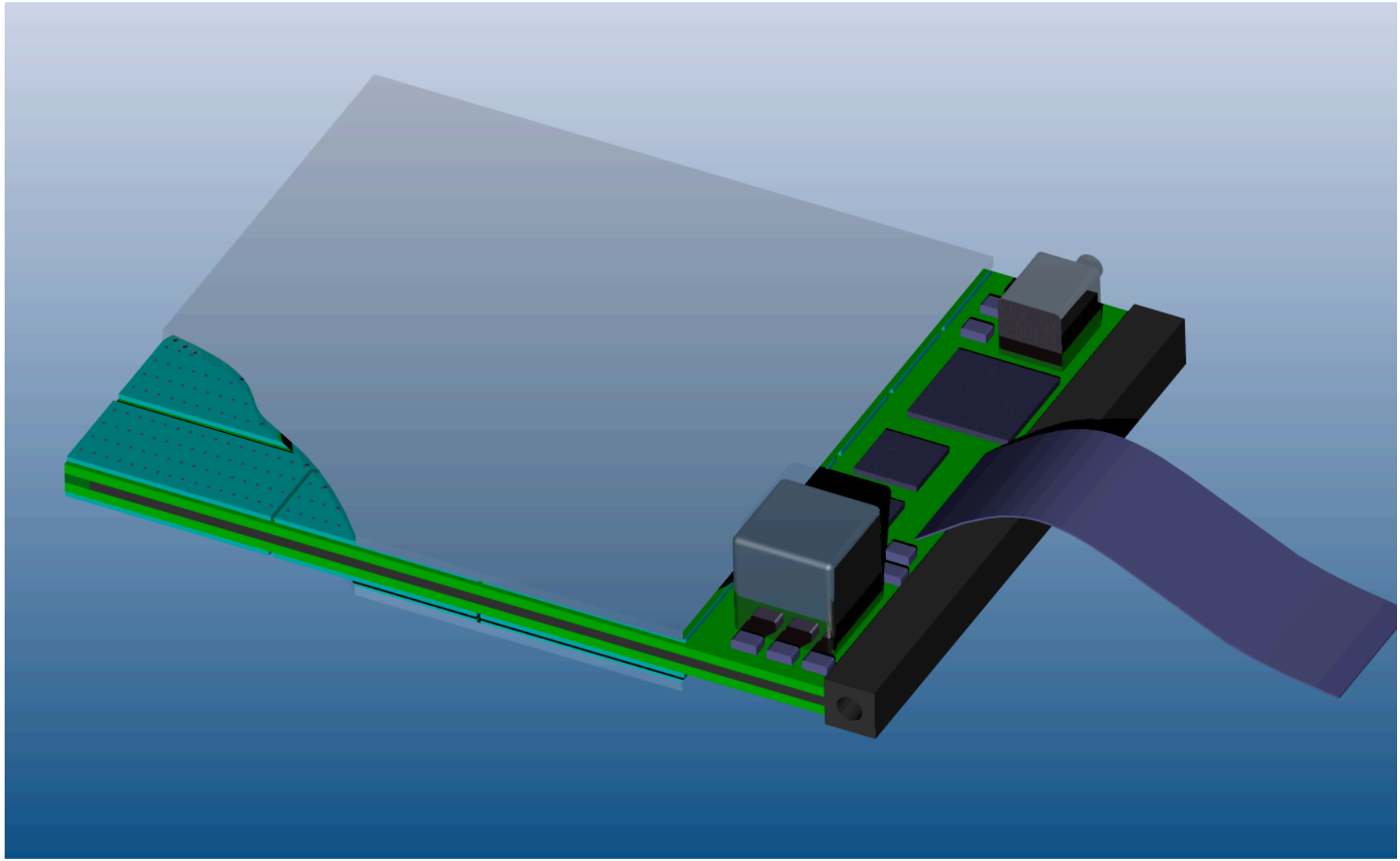




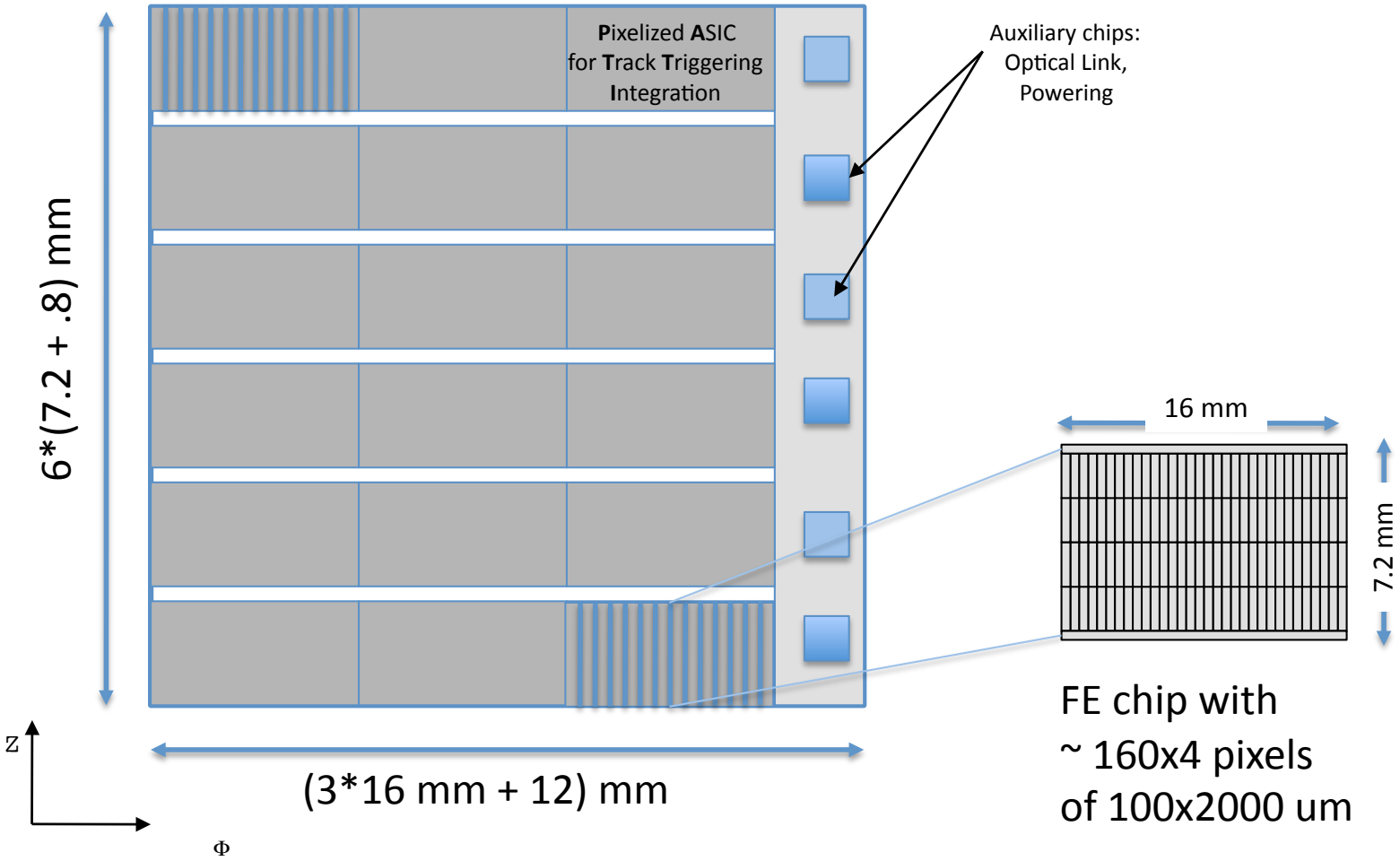
Red Modules: Trigger Layers
Green Modules: Outer Layers

Total area of trigger layers: 26.9 m^2
Total area of outer layers: 85.1 m^2

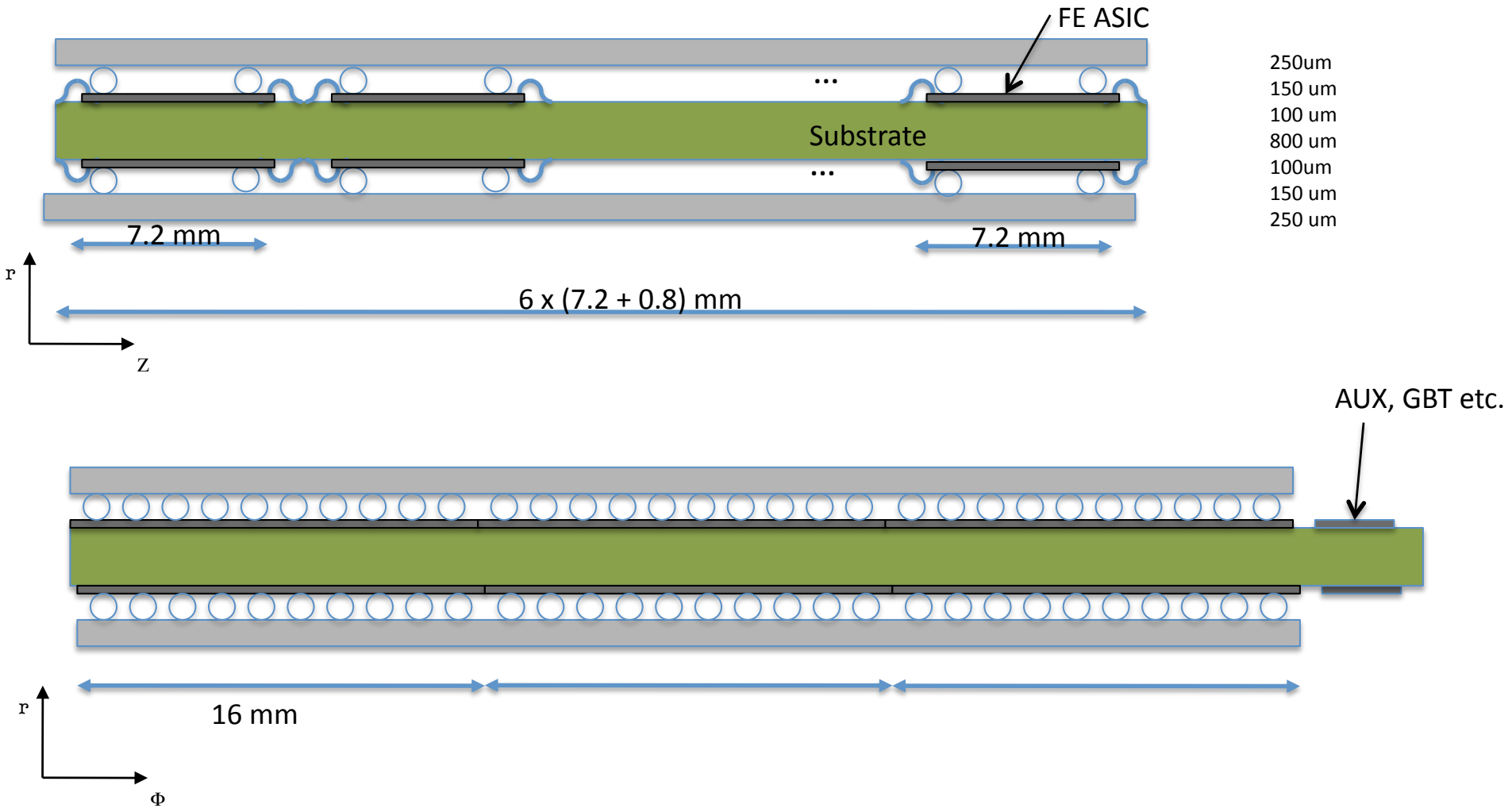
Module overview



Module Layout



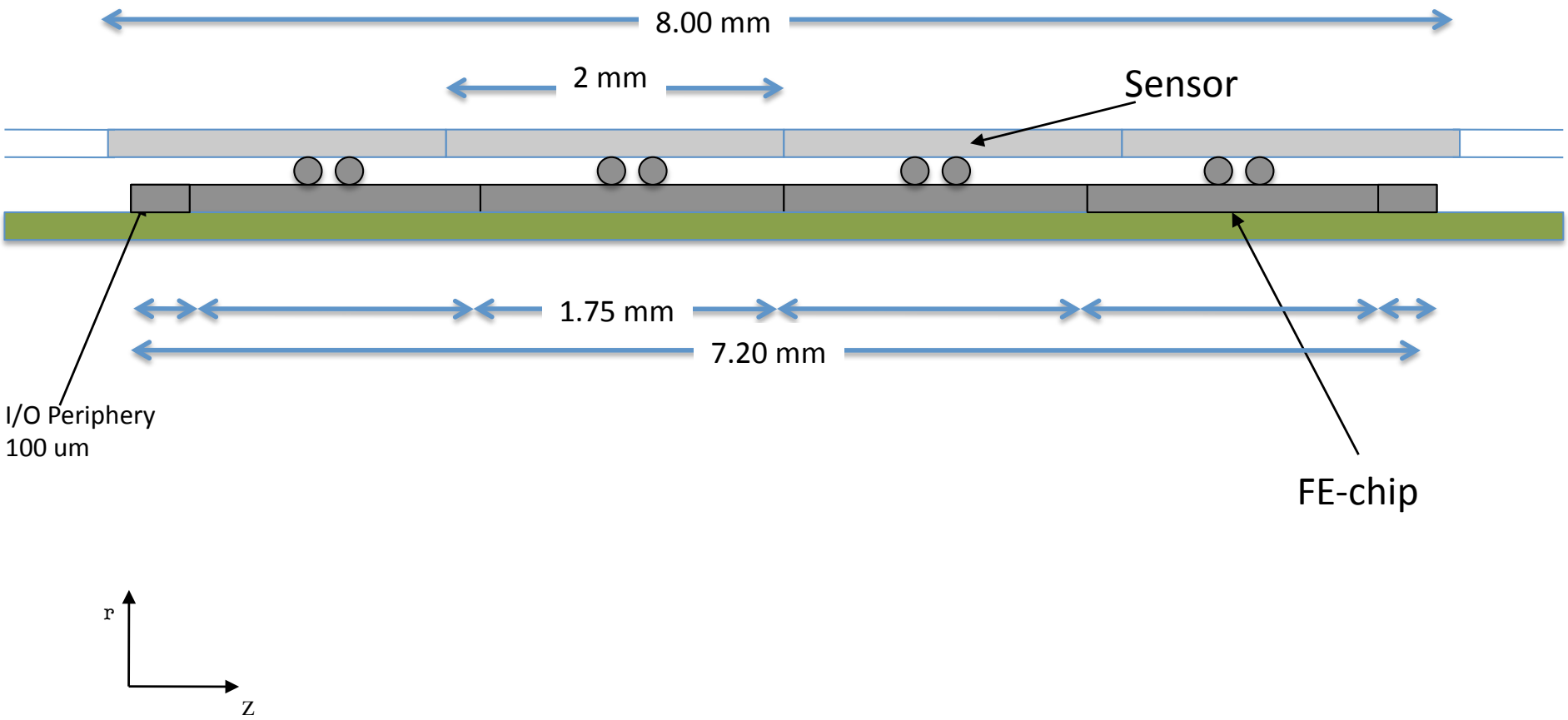
Base Option: WB + BB



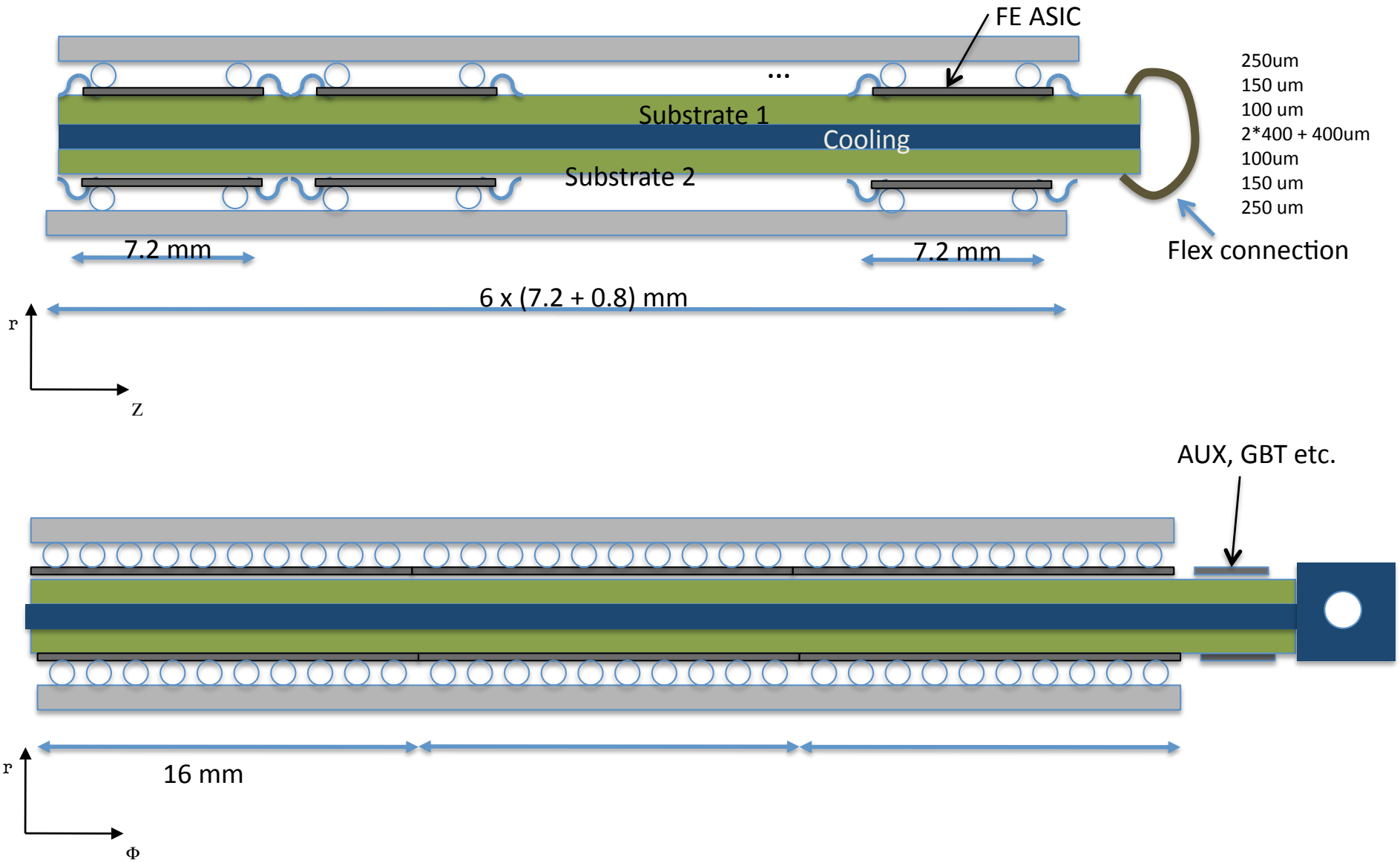
250um
150 um
100 um
800 um
100um
150 um
250 um

Module size = [6 * 8] x [3 * 16 + 12] mm

Sensor to FE chip
Z-pitch adaptation
(for WB option)

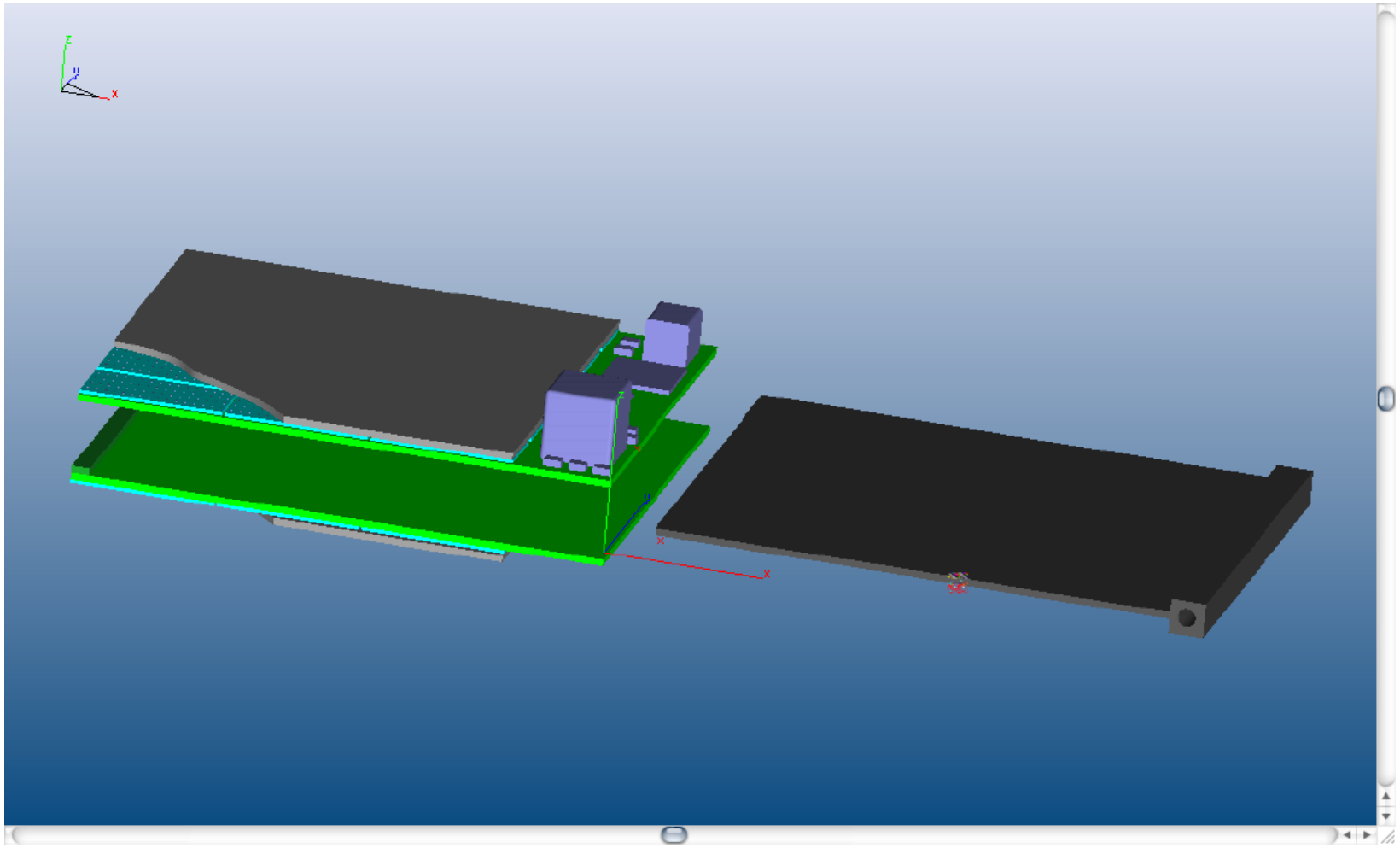


Base Option++: WB & BB

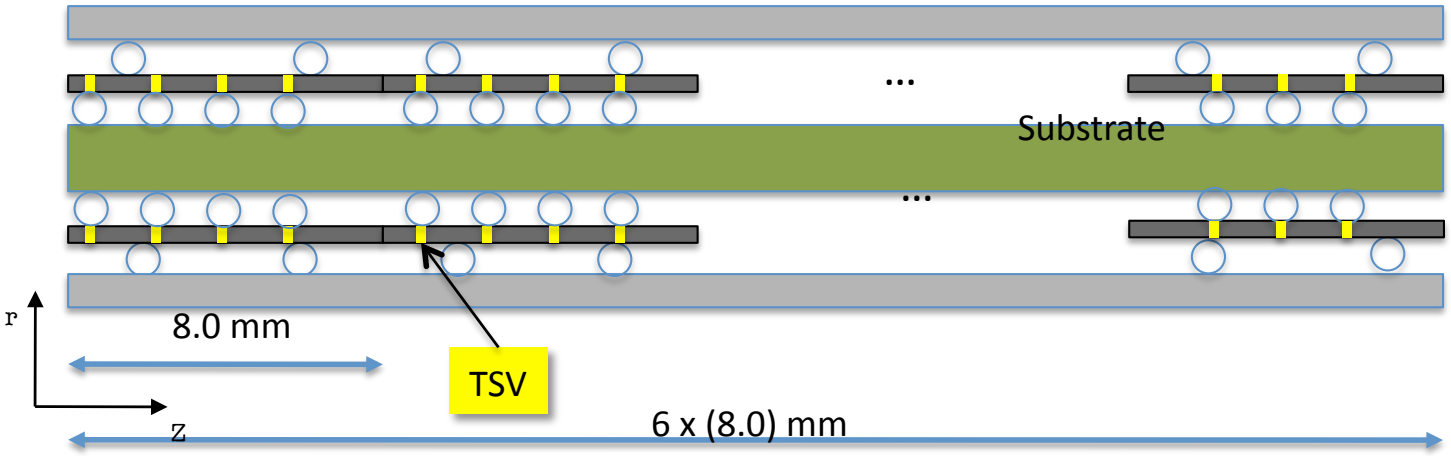


Module size = [6 * 8] x [3 * 16 + 12] mm

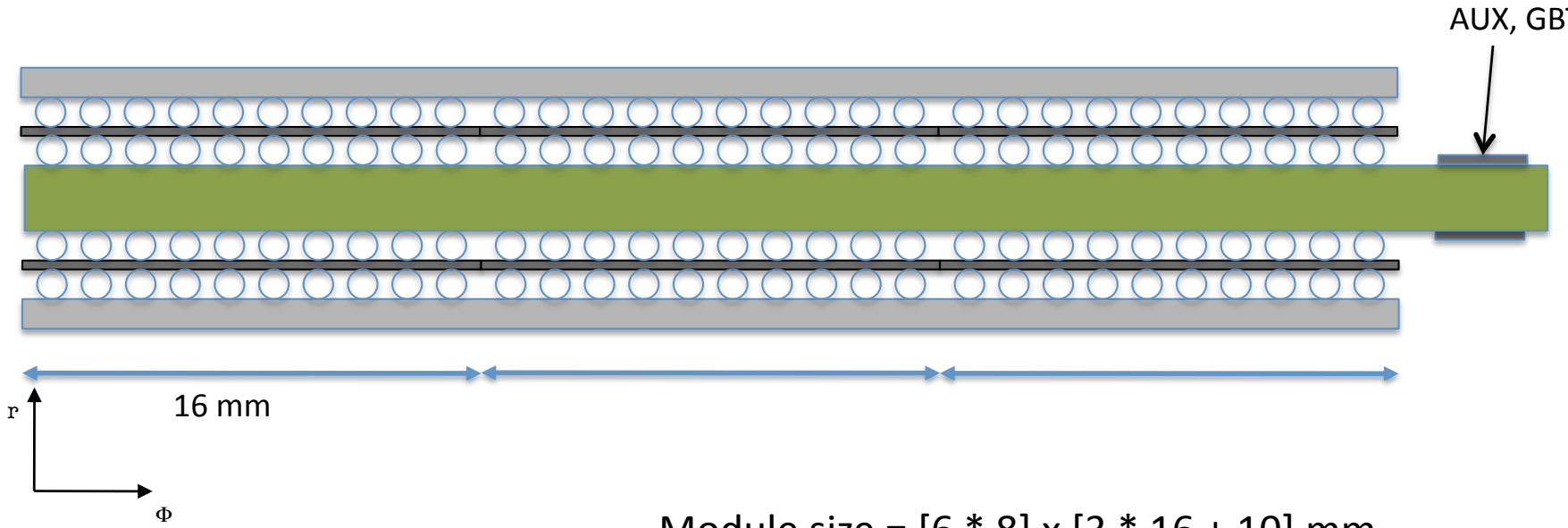
Module Assembly



Advanced Option: TSV & BB

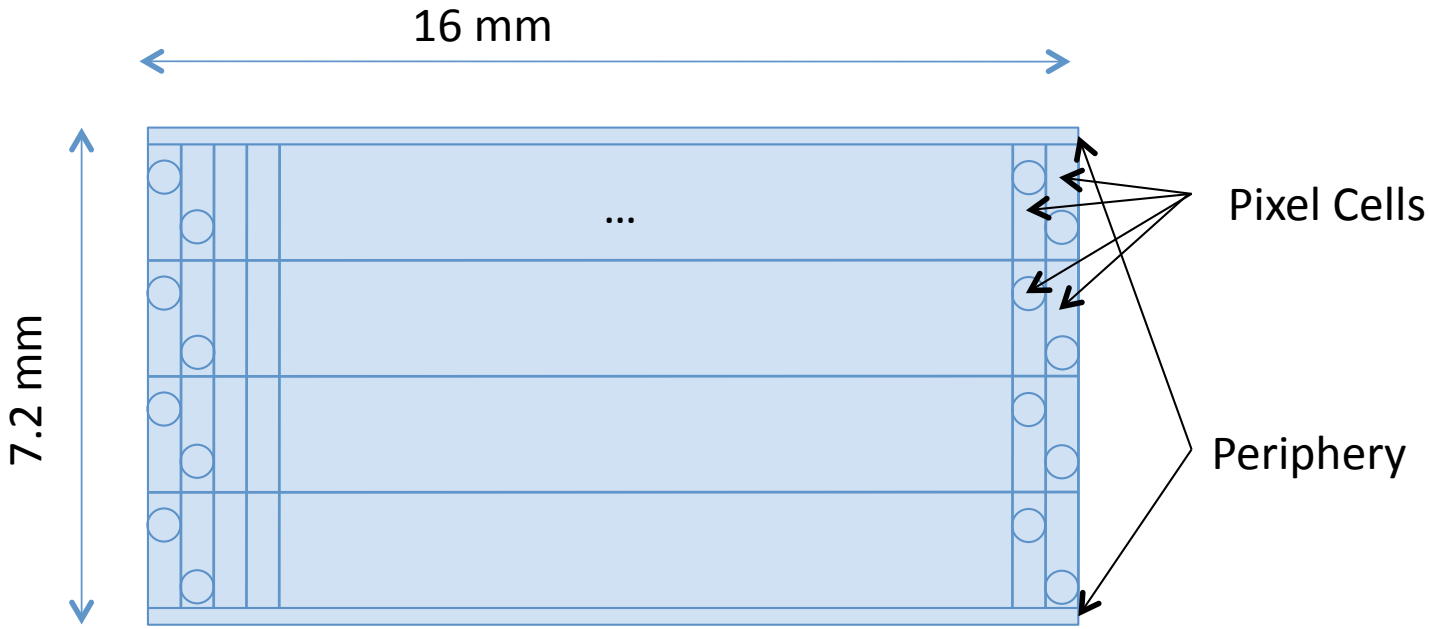


Sensor	250um
C4	100 um
ASIC	100 um
C4	100 um
Substrate	700 um
C4	100 um
ASIC	100um
C4	100 um
Sensor	250 um



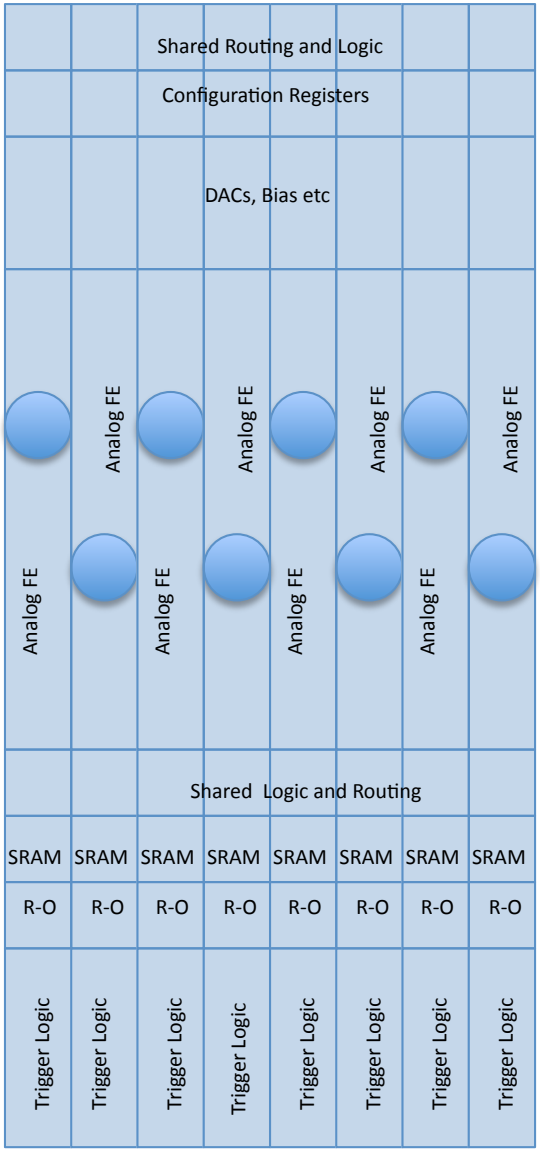
Module size = [6 * 8] x [3 * 16 + 10] mm

FE chip floorplan



ASIC size:
7.2 x 16 mm2
Sensor size: ~ 100 um x 2000 um
Channel size: 100um x 1750um
channels: 160 x 4 = 640 channels

Pixel block Floorplan



Single Channel size: 100 x 1750 um

Analog circuit: 100 x 750 um2

Bias, DACs etc: 100 x 200 um2

C4 Bump-bond pad: 90 x 90 um2

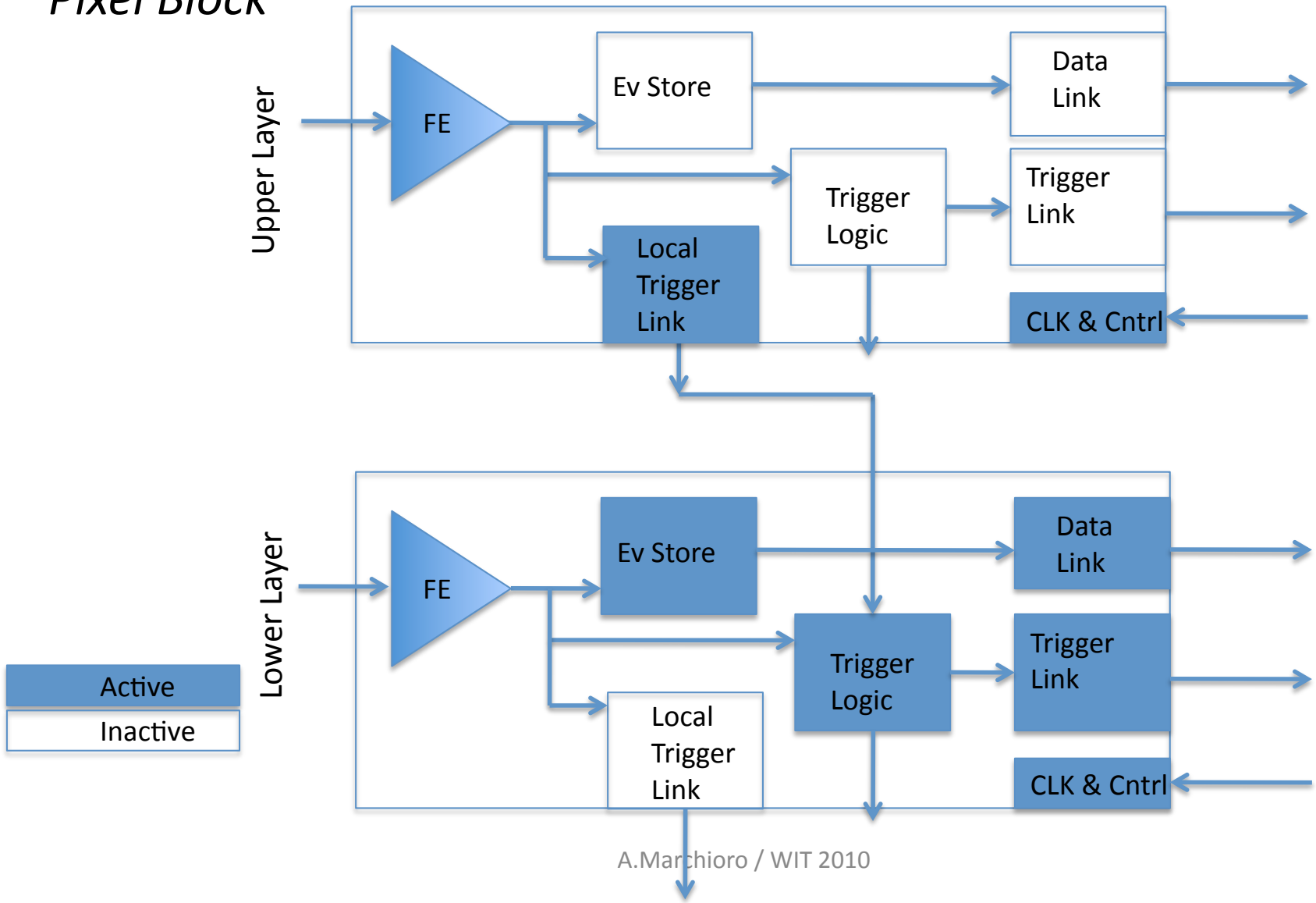
Event Memory depth: 6 usec
-> memory length: 40 x 6 ~ 256 bits
-> cell size: 2.1 um2
-> block size: 750 um2 ~ 28x28 um2
(rounded to 100x100 um2)

Trigger logic and Lookup SRAM
-> 300x100 um2

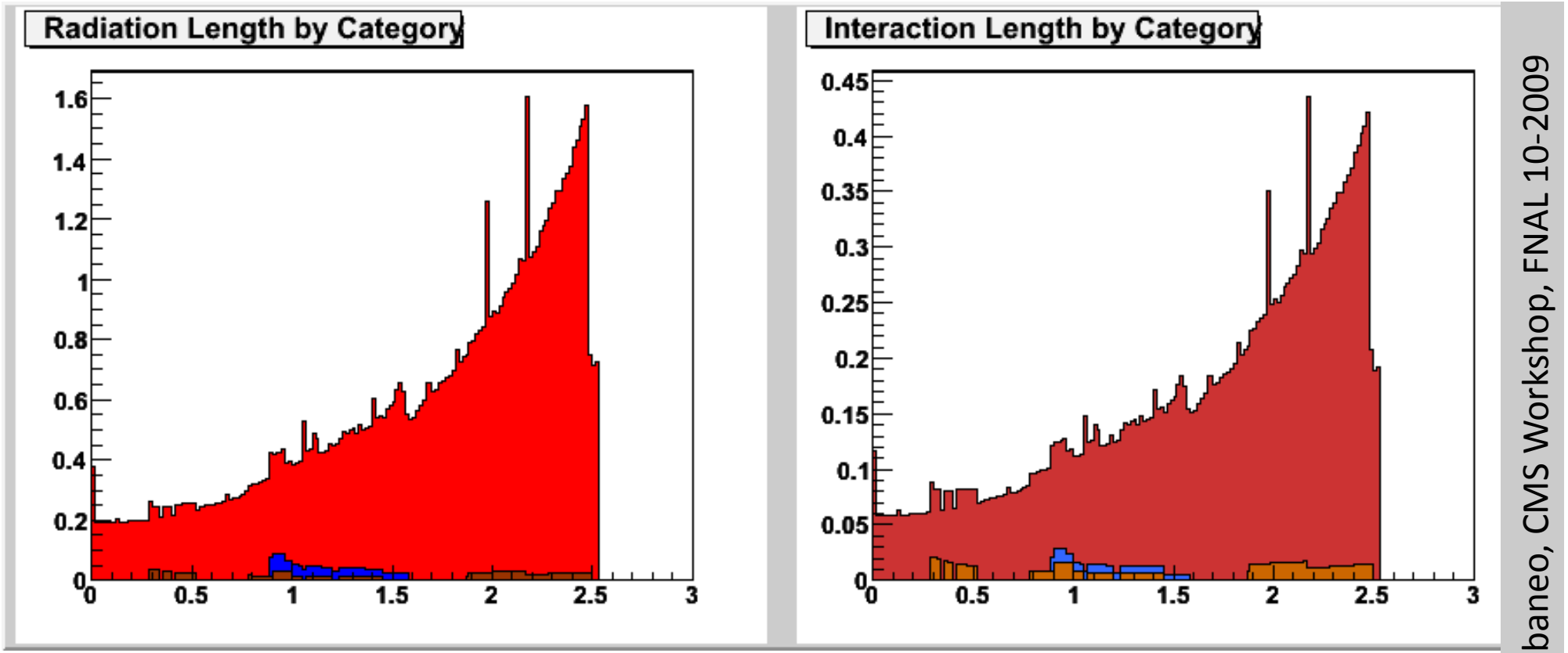
Read-Out Logic: 80 x 80 um2
Channel Configuration regs: 100 x 100 um2

Functional block diagram

Pixel Block



Material



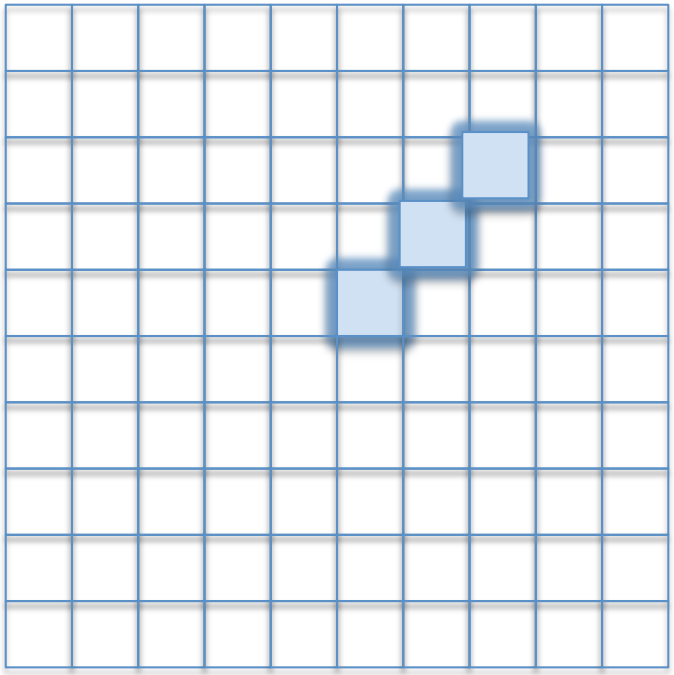
From D. Abbaneo, CMS Workshop, FNAL 10-2009

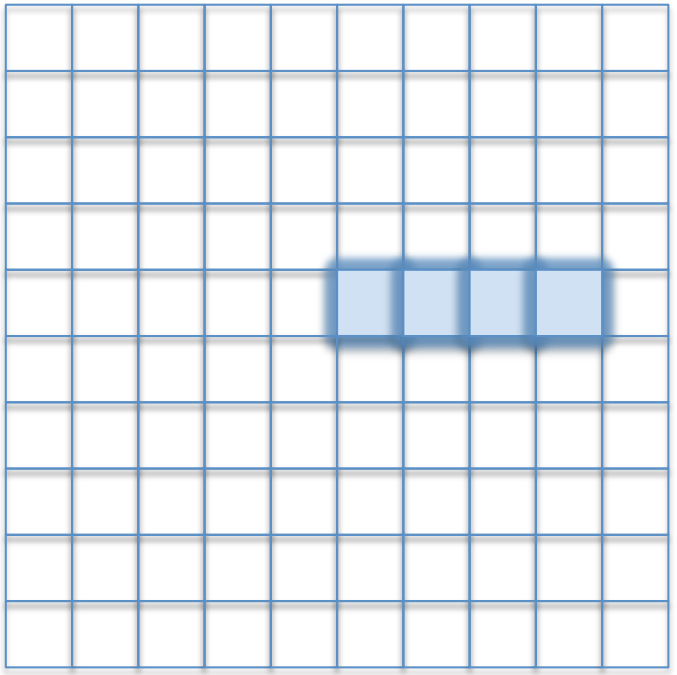
- Average radiation length in tracking volume [0, 2.4]: 55%
  - Max of ≈155%
- Average interaction length in tracking volume [0, 2.4]: 16%
  - Max of ≈42%
- Material entirely dominated by the two P_T layers

Trigger Logic Algorithm

- Step 1:
  - Elimination of large clusters
- Step2:
  - Transfer from “top” chip to “bottom” chip
  - Z alignment
  - Φ alignment
- Step 3:
  - Coincidence between planes

Step 1:
Rejection of large clusters





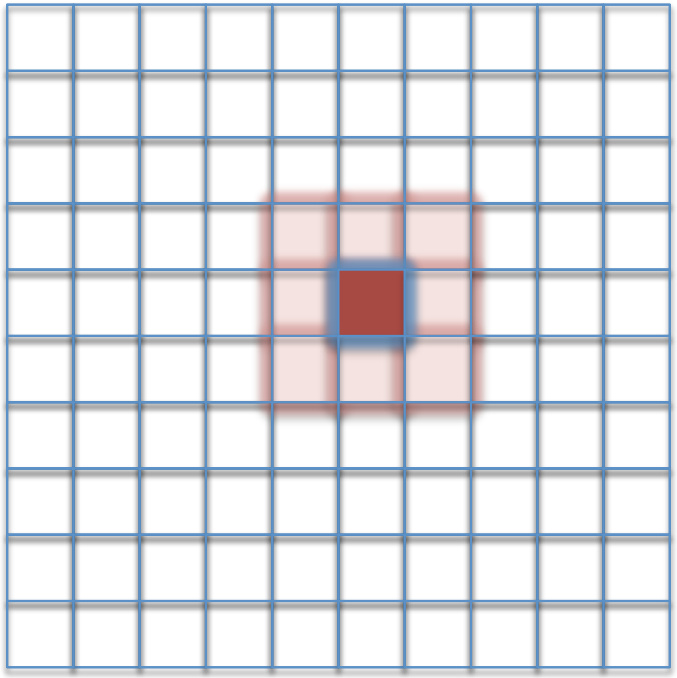
- All these combinations to be eliminated before transferring pattern to master layer.
- Algorithm for clean-up:

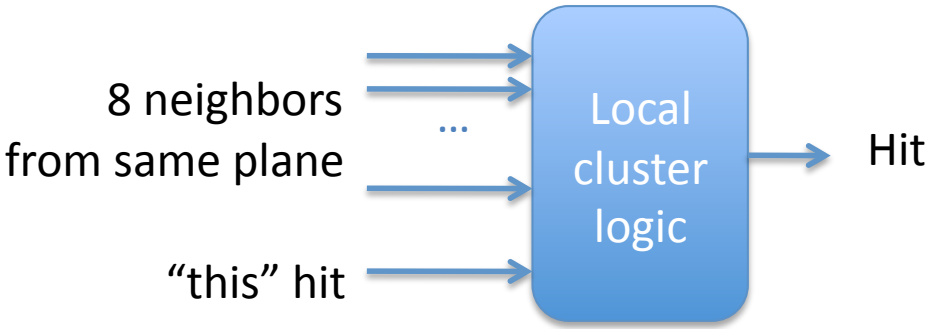
  if any pixel has more than one neighbor turned on in a + - 1 vicinity,
  all are turned off

Step 1: Cluster reject

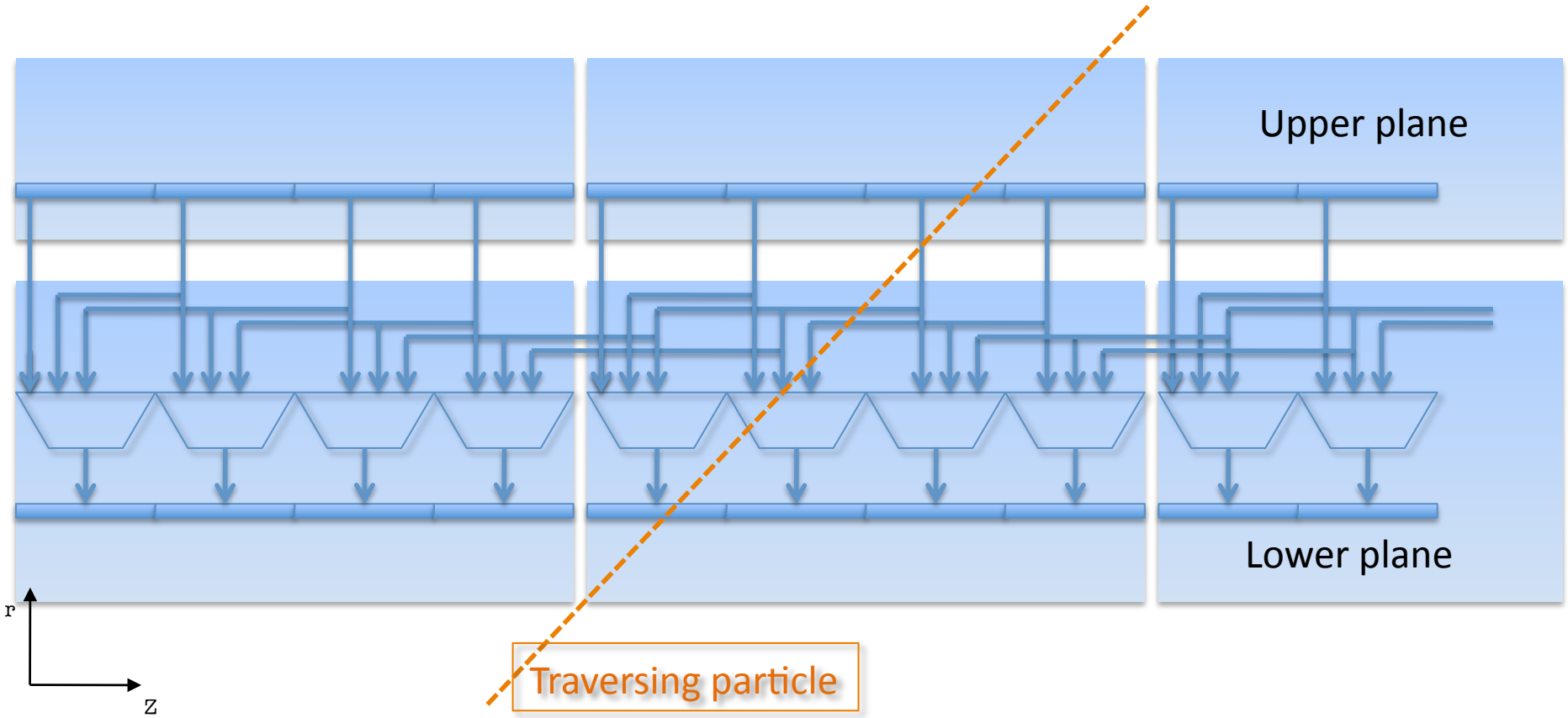
For each pixel hit:

- Look in all adjacent pixels for clusters of at least 3 pixels
- eliminate all clusters of 3 or more



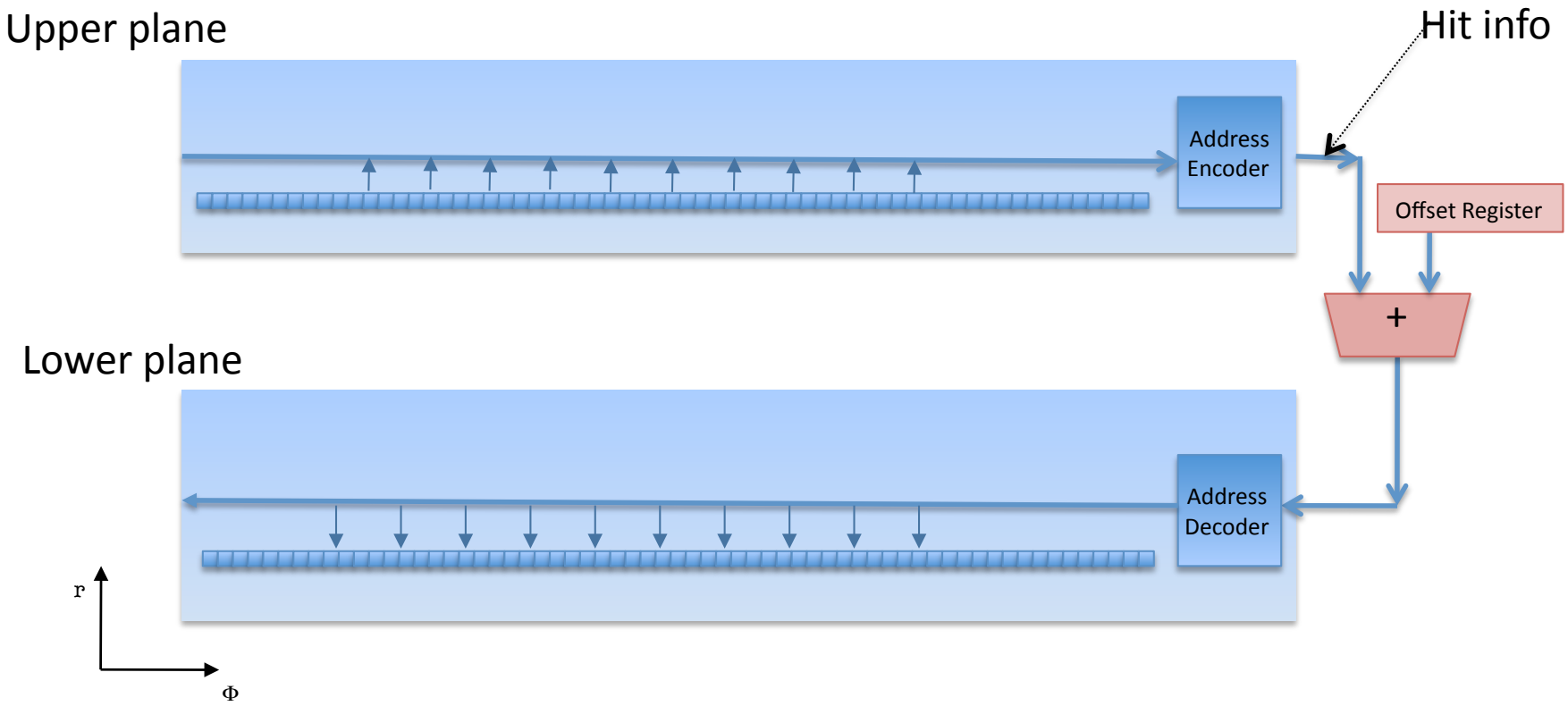


Step 2: 2+2mm strips
Simplified Z alignment



Notice:
Maximum Z shift for 2 mm spacing: ~ 11 mm (i.e. ~6 pixels) at η=2.5
but only some connections shown

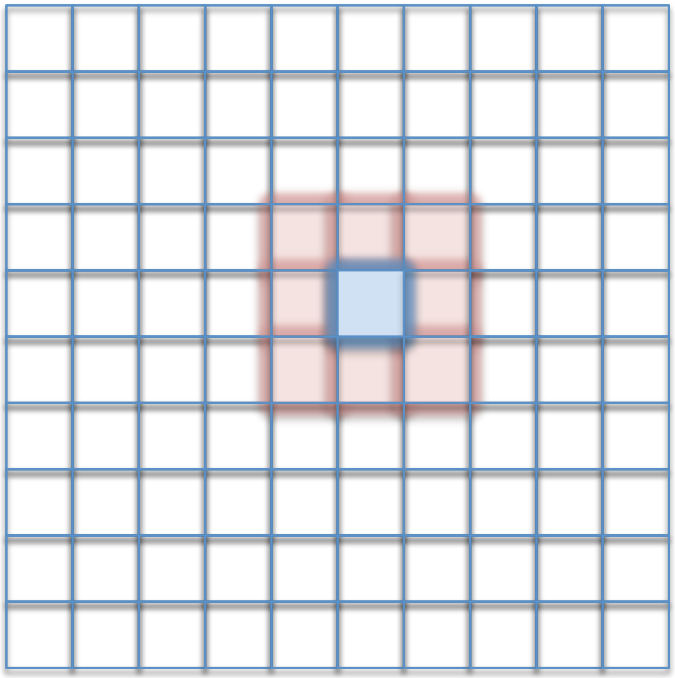
Step 2: Φ Alignement

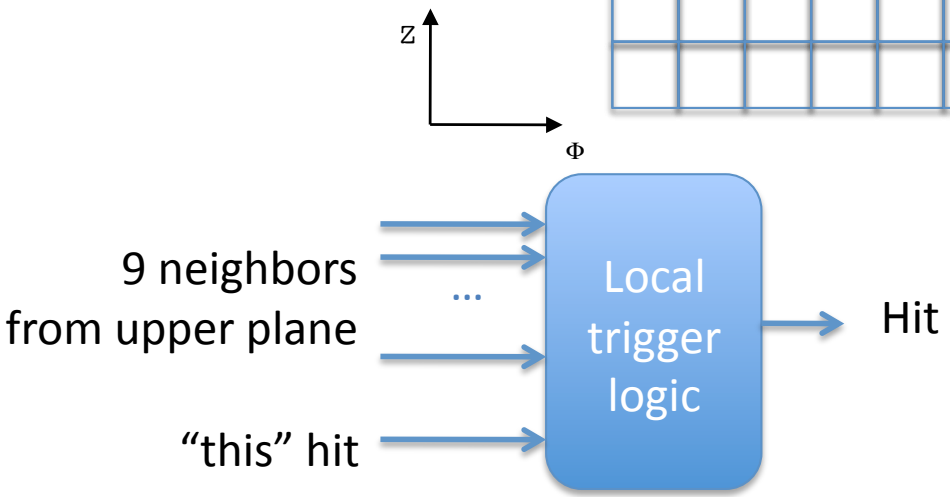


Having an “electronic” alignment scheme can save significant manufacturing cost

Step 3, option 1: coincidence between planes

Each pixel in the lower plane receives the "image" of the 9 adjacent (and re-aligned) pixels in the upper plane

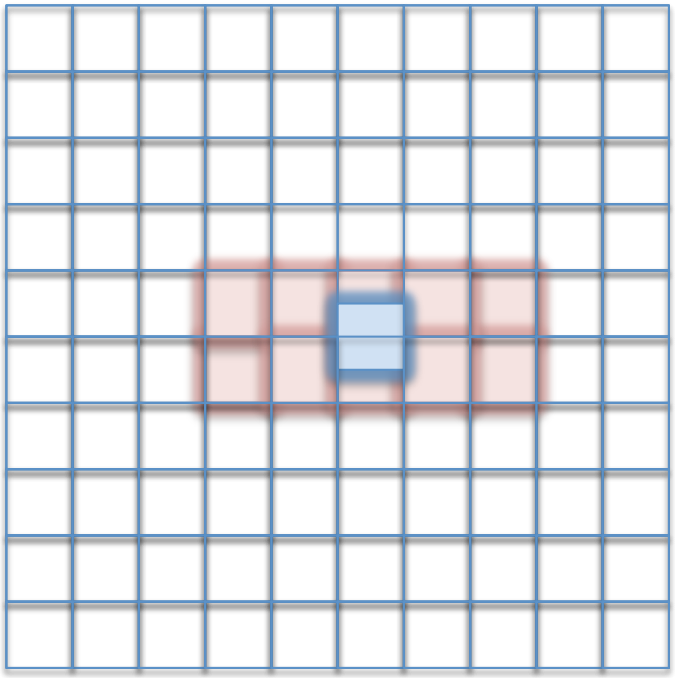


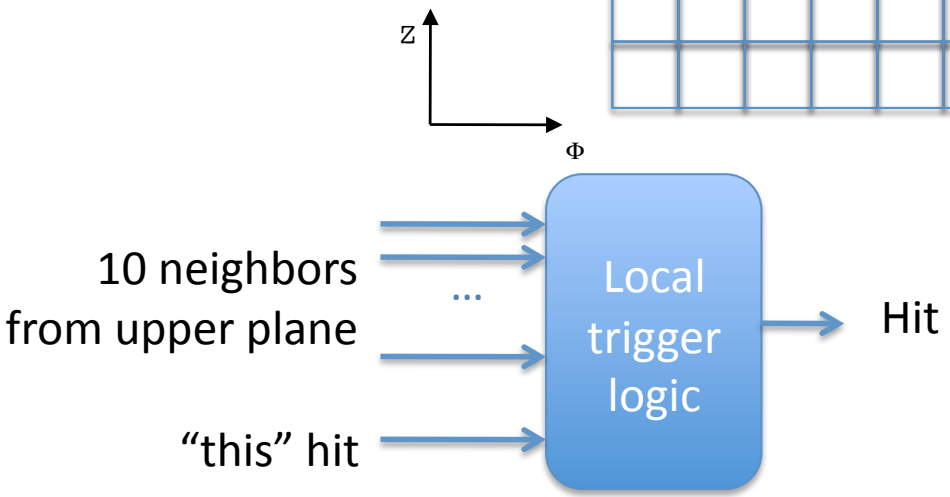


Step 3, option 2: coincidence between planes

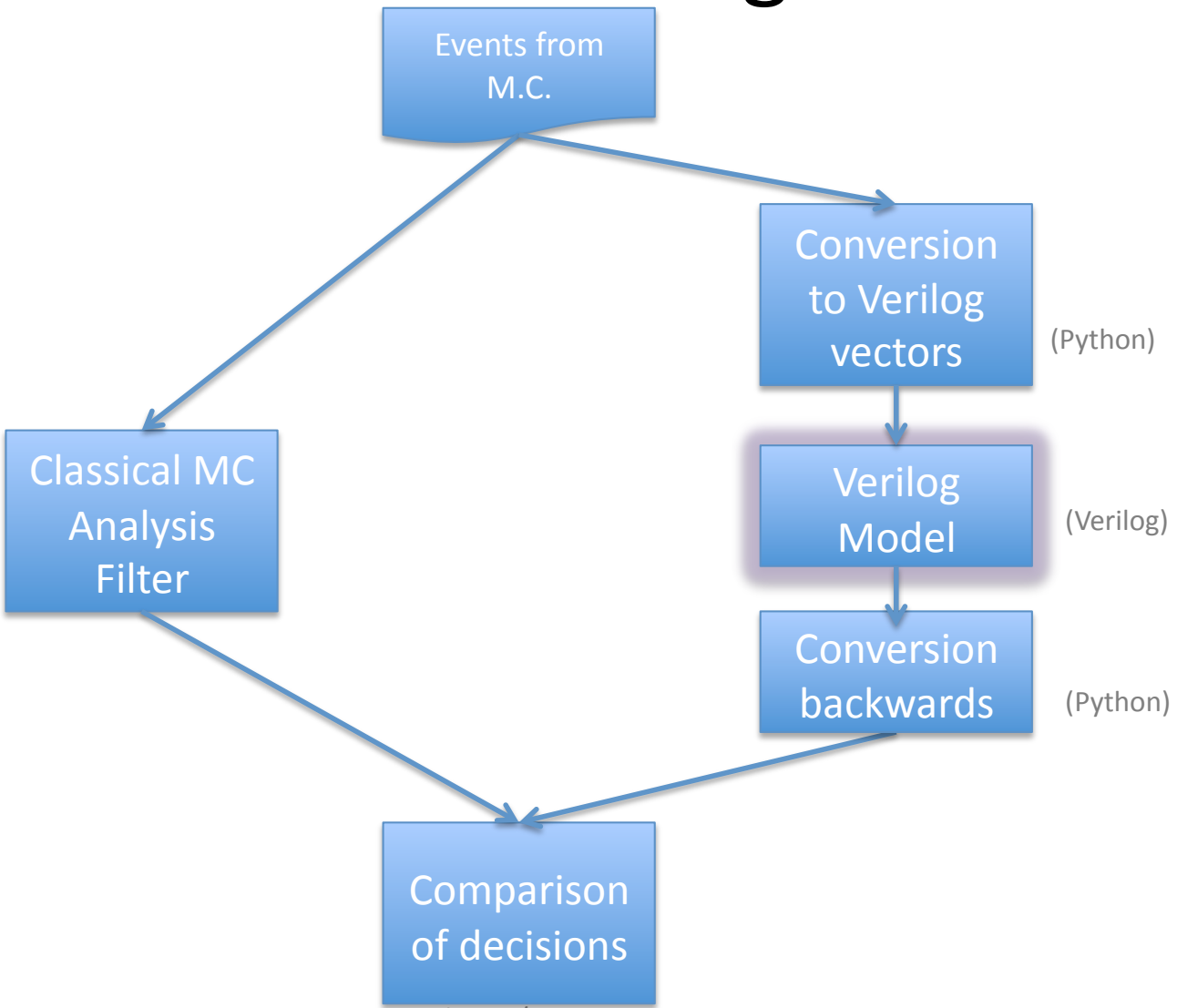
If matching at low pT necessary, then a larger Φ mask is necessary

Mask pattern: 5x2

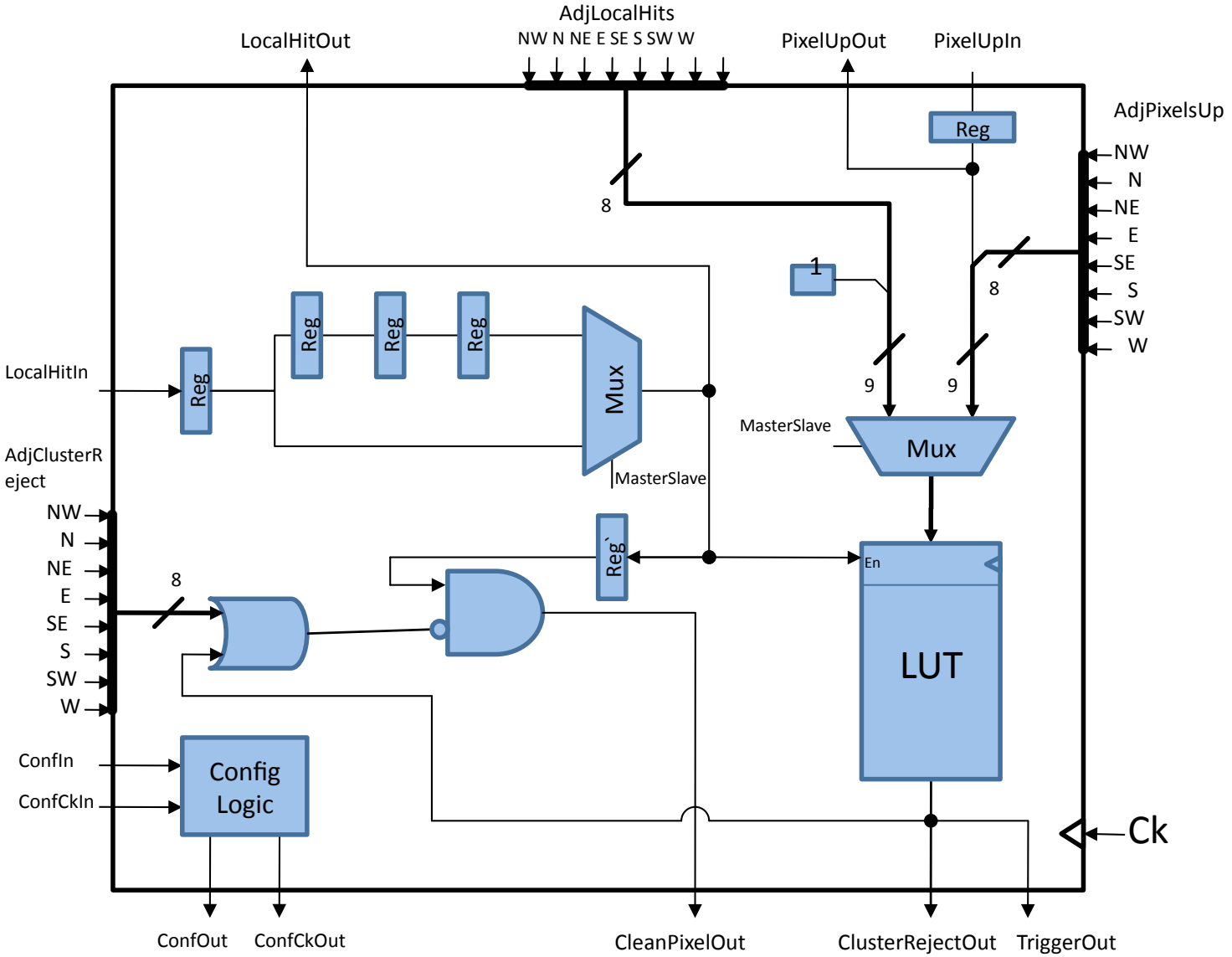




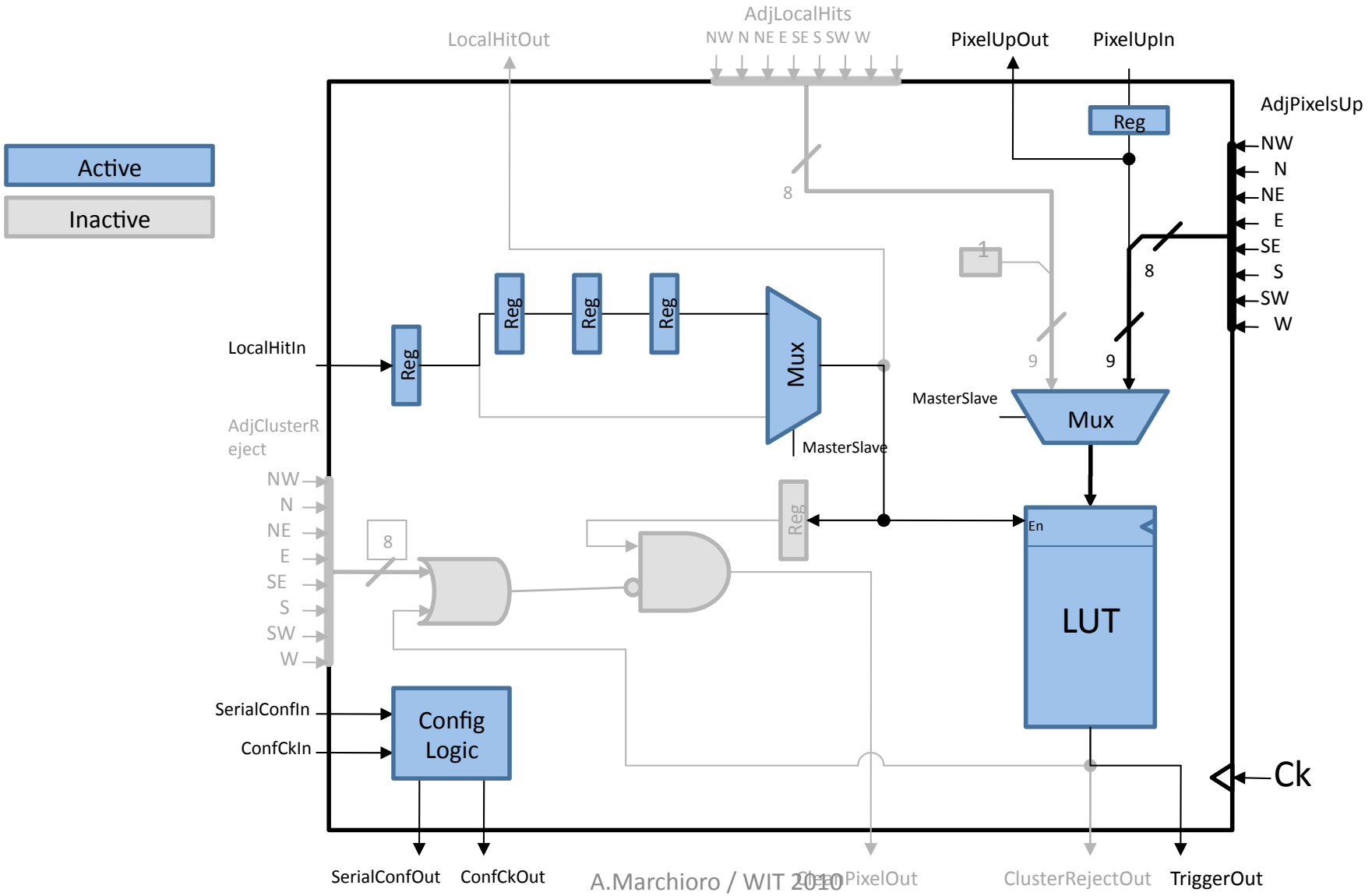
HW (Simulated) vs Software event filtering



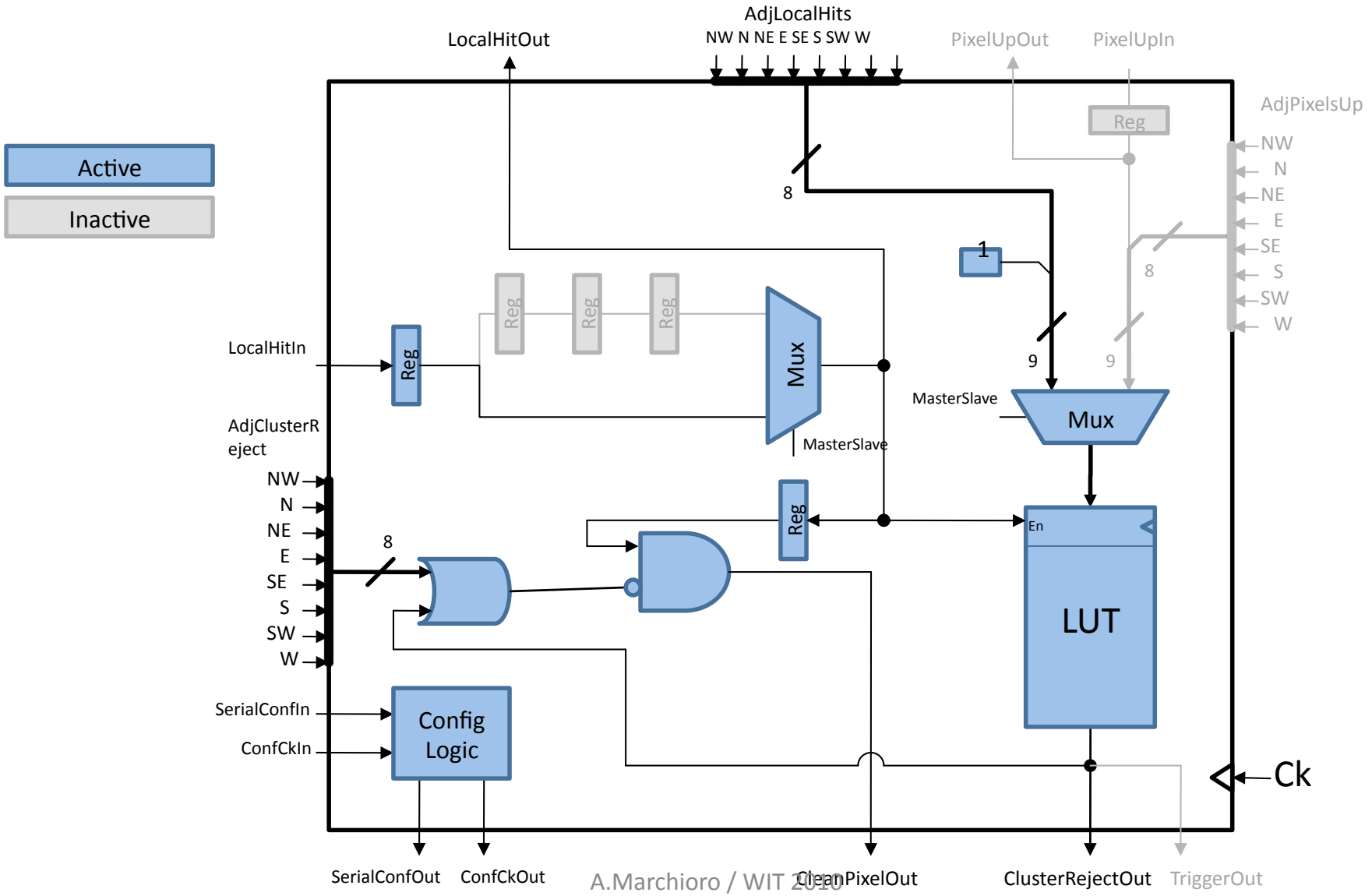
Step 3: Verilog Model of Pixel Logic



Pixel (Master at bottom)



Pixel (Slave on top)



Algorithm efficiency

pT	Total Events	Trigger miss	Efficency
2 GeV/c	118	4	96.61%
5 GeV/c	120	0	100%
50 GeV/c	112	2	98.21%

Remarks:

- few events in each MC sample
- full Z and Φ alignment logic not yet implemented
- simple trick to improve Φ coverage at chip boundaries not yet implemented

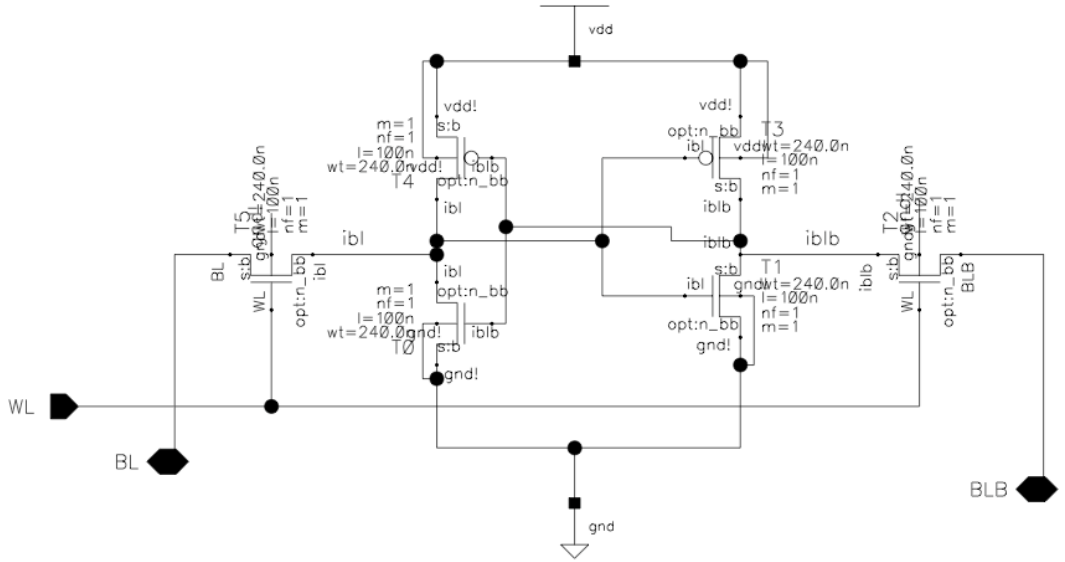
Missed event

Power Estimation for SRAM

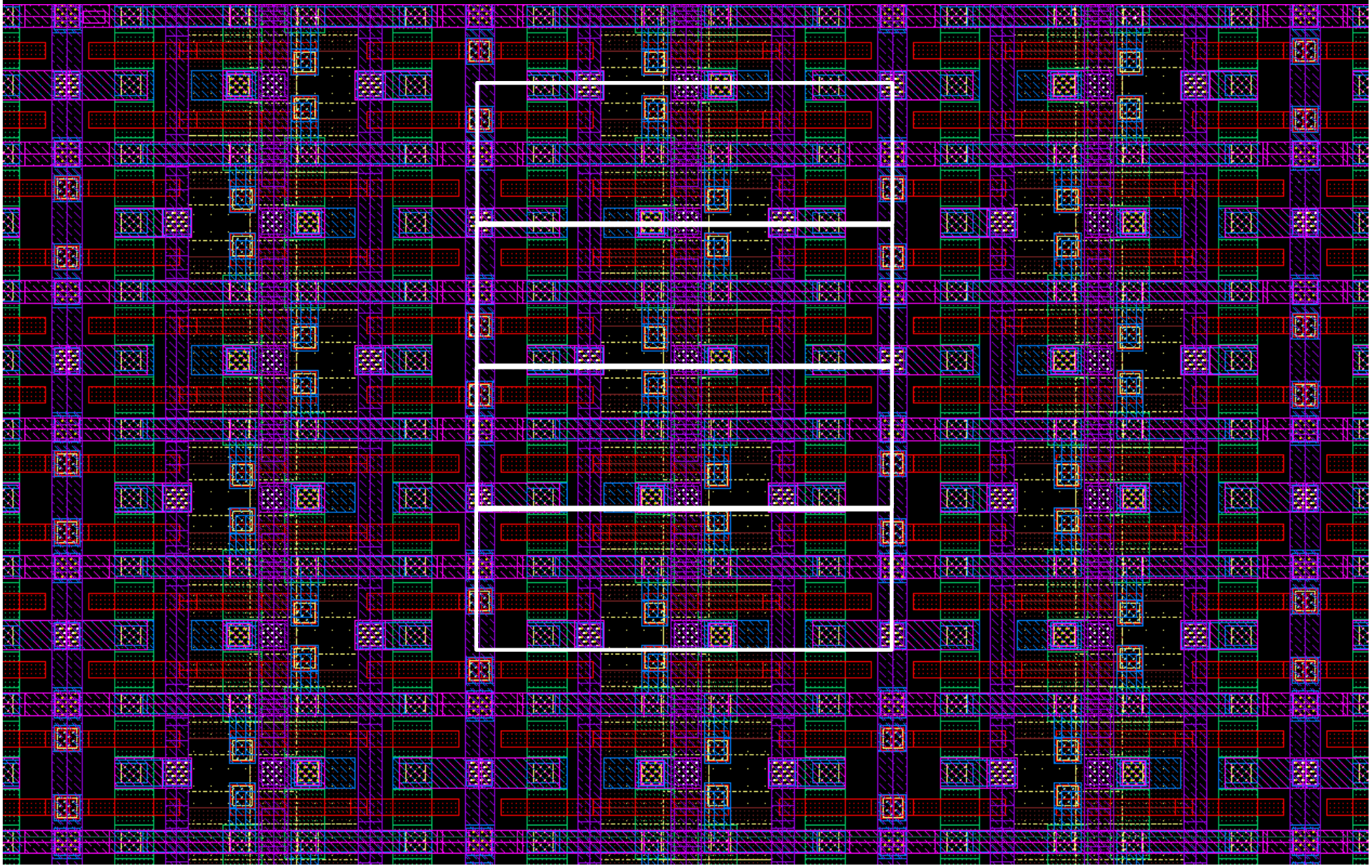
- Identical memory block is used for cluster rejection (upper chip) and trigger matching (lower chip)

- Power reduced by:
  - Optimized manual layout of all blocks (no generator used)
  - Enabling bitline PC only on active column during read
  - Slow rise-time on WL enables using minimum size transistors in cells

- 512x1 SRAM power:
  - Full from circuit simulation (including address decoding etc.) for 90nm technology @1 V and operating at 40 MHz is 13 uW
  - Cell size 2.52 x 0.84 um



SRAM Array detail



Conclusion

- Final detailed design still awaits for confirmation from MC refinements and early LHC cross-section data for final word on occupancies
- Moderately aggressive technological choices with predictable cost
- Verilog model of entire Front-End Trigger Logic and interconnectivity developed:
  - Verilog capable of processing data from real MC events
- Preliminary floorplanning
  - Preliminary power estimations gives ~ 100 uW/pixel for a 100x2000 μm^2 pixel size
- Design of critical blocks in 90 nm CMOS has started

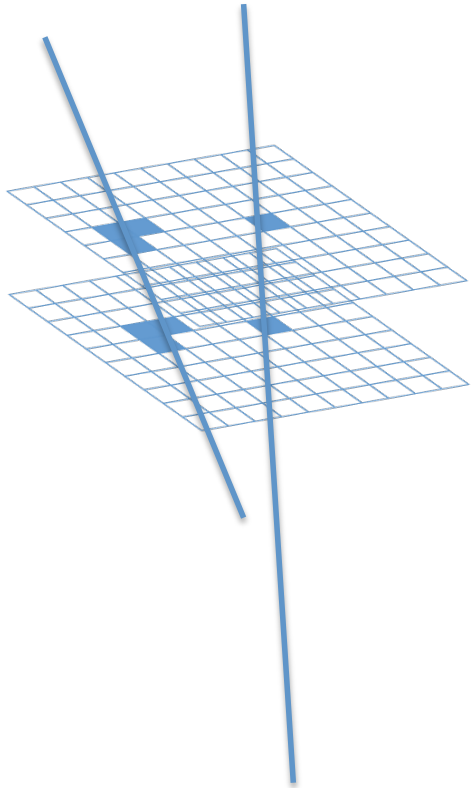
Thank you!

SPARE SLIDES

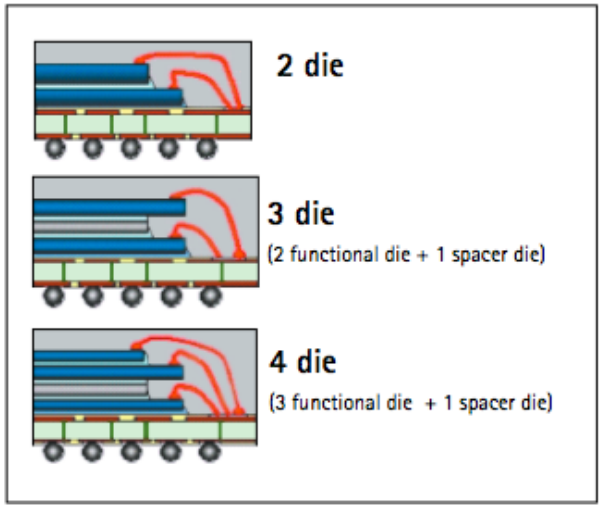
Measuring transverse momentum (p_T)

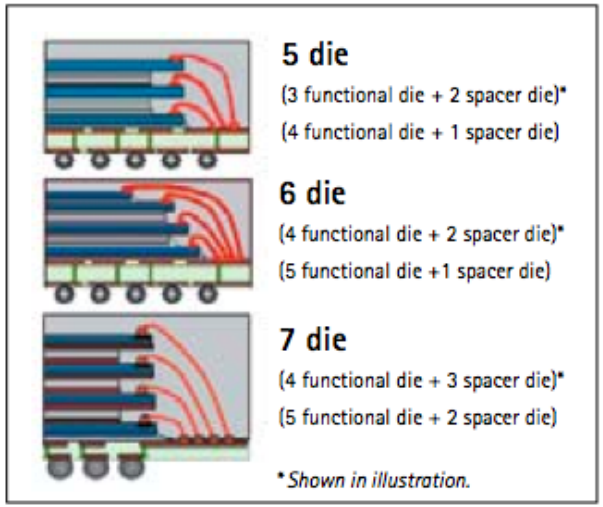
OK, small bend

Not OK, too large bend



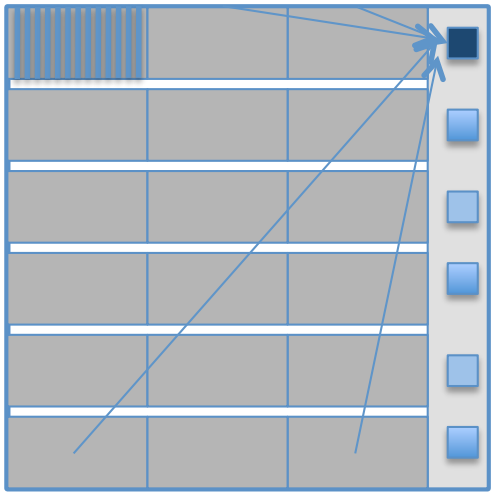
Stacking in Commercial Devices





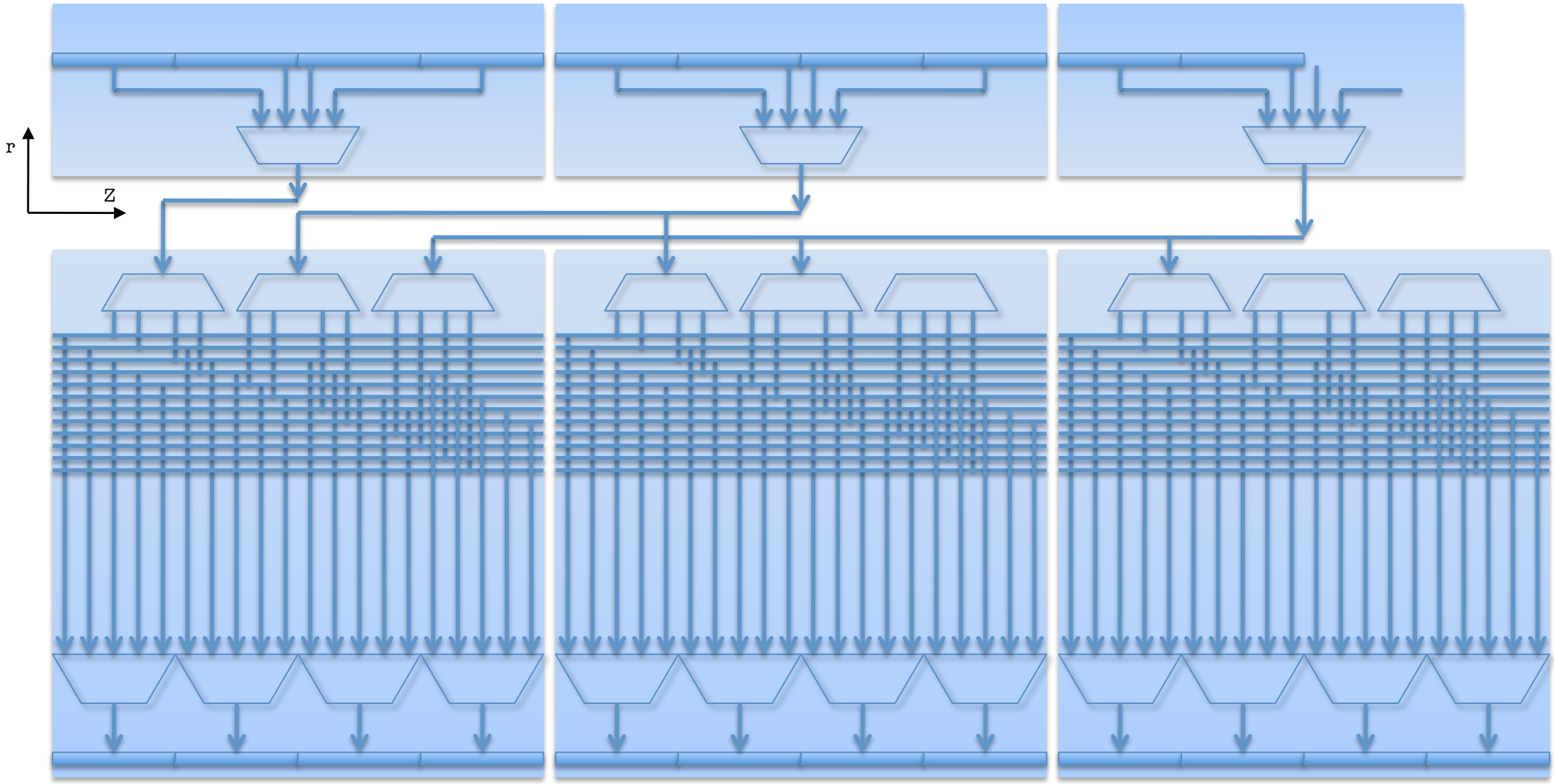
Front-End to Opto-link connectivity

Opto to Trigger



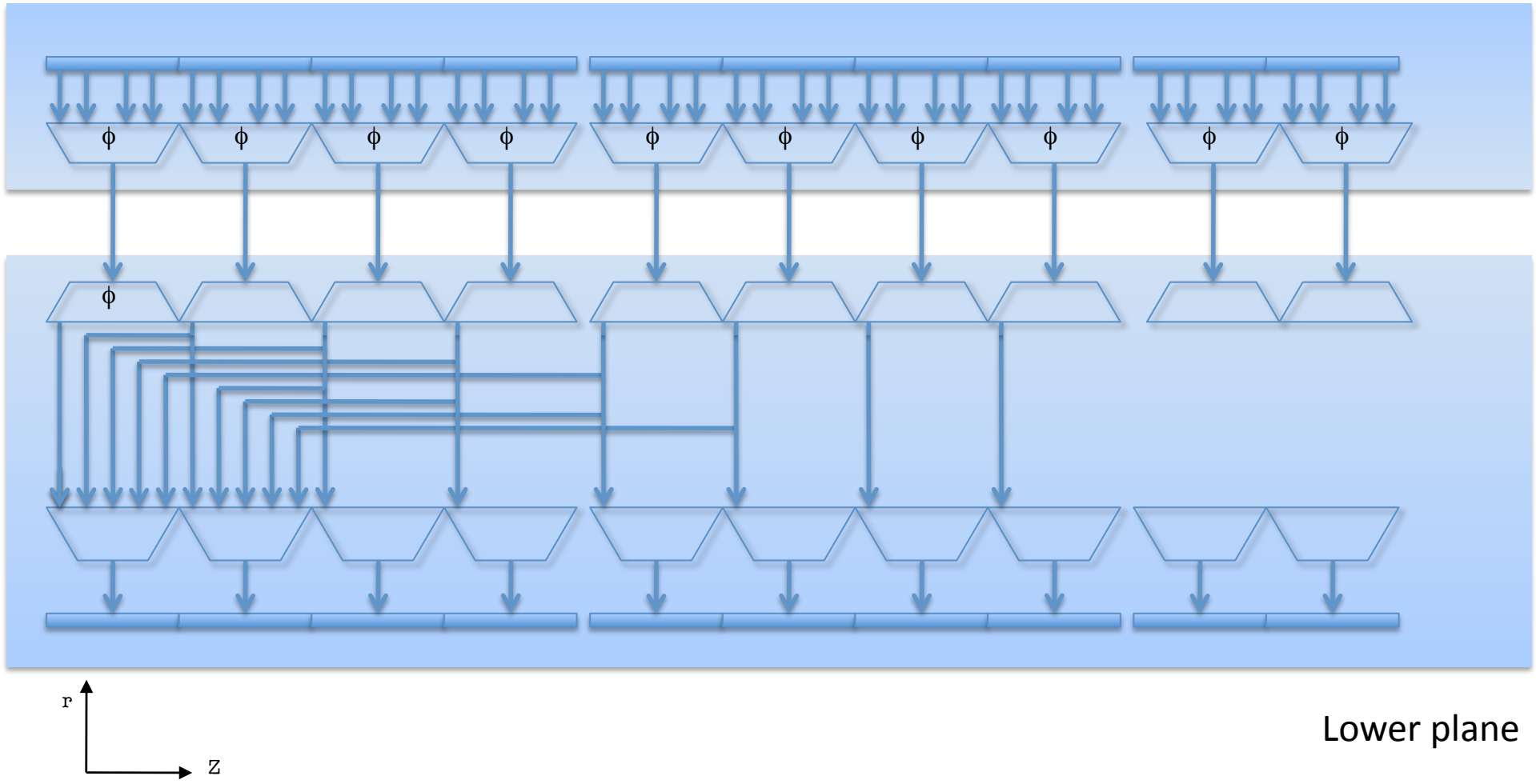
- 18 chips/side
- 18 inputs to Opto-link
  - Only master provide trigger info
- Synchronous decision
  10+5 ??? bit/chip @ 40 MHz
  to
- Overhead per module: 8 bit
- Total BW/ MCM = 10 * 18 * 40 MHz = 10.8 Gbit/sec

Step 2':
Z alignment with (some) details



Maximum Z shift for 2 mm spacing: ~ 11 mm at η=2.5

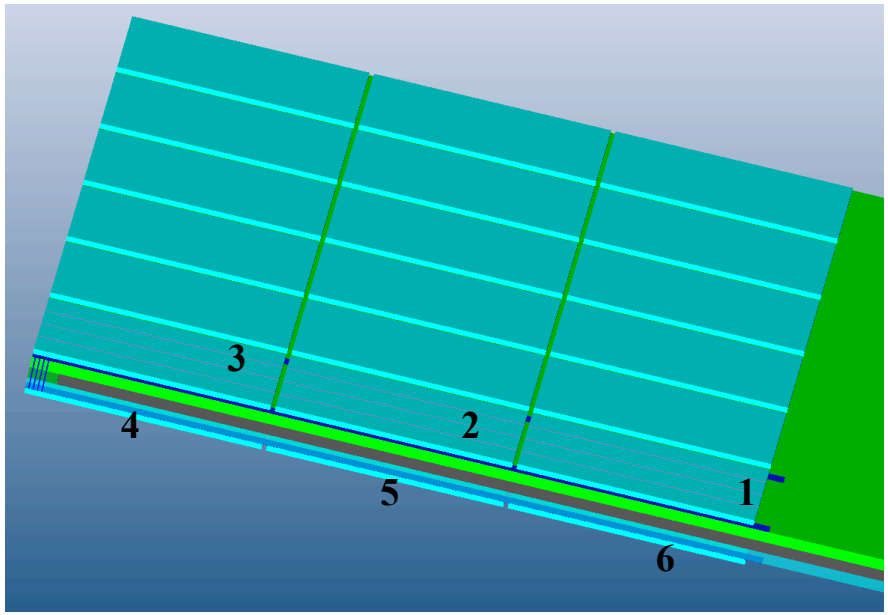
Step 2 (option):
Z alignment with (some) details

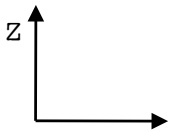


Maximum Z shift for 2 mm spacing: ~ 11 mm at η=2.5

Transmission between planes

- Each row of each chip from upper plane has to transfer data to corresponding chip in lower plane (chip 1 to chip 6, 2 to 5, 3 to 4)
- One 12 bit word per chip row, 48 bits per chip to go through substrate
- @160 MHz, four cycles are available:
  - Cycle1: 1 talks to 6
  - Cycle2: 2 talks to 5
  - Cycle3: 3 talks to 4
- Using regular CMOS drivers @1V swing, estimated power is:
  - 48 * ½ * 10 pF * (1V)^2 * 20 MHz = 4.8 mW (total power budget for entire chip: 64 mW)





Connectivity per module (high granularity, 1')

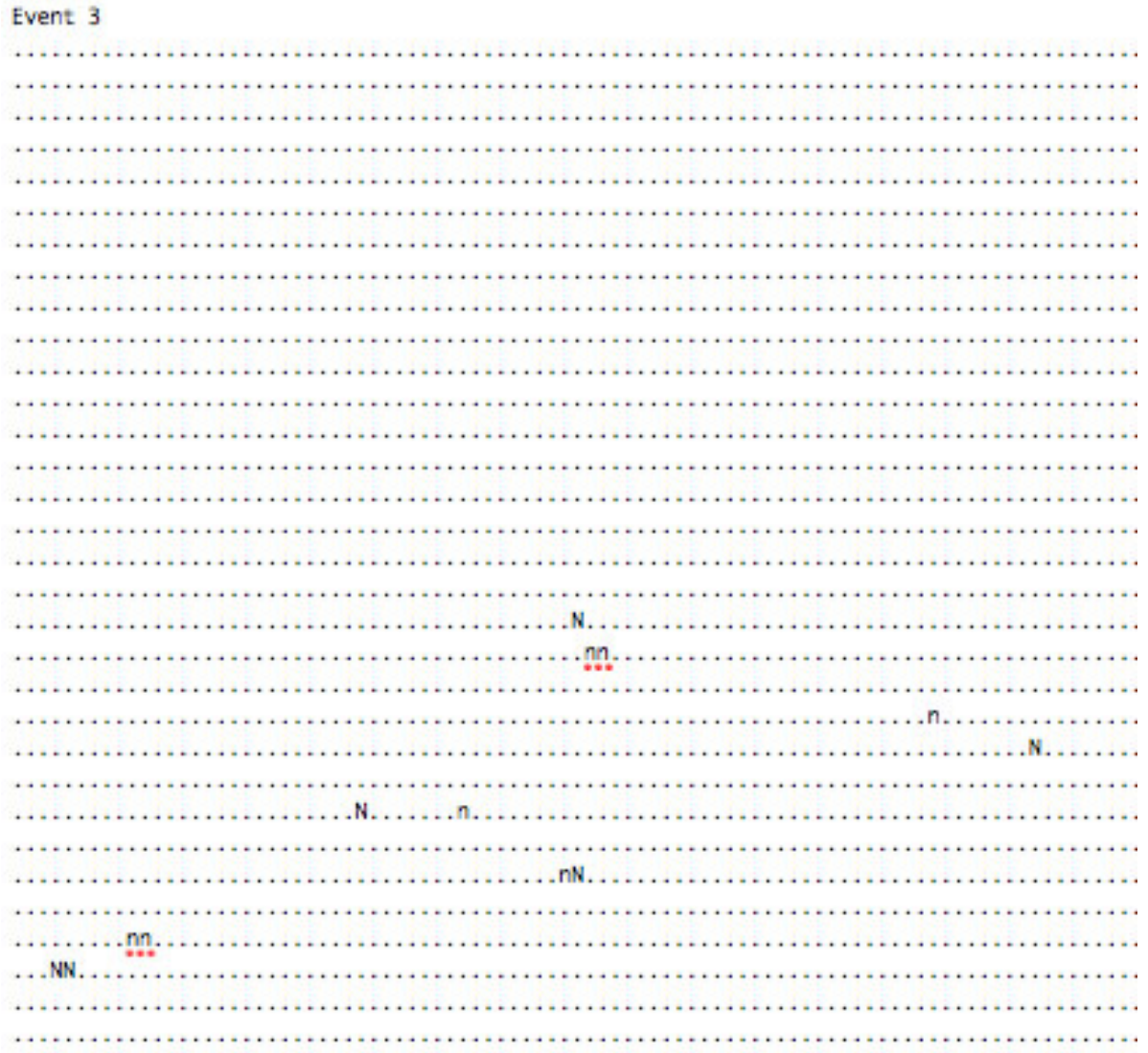
- Each MCM has 18 chips per side , i.e. 3 rows of 6 chips

- Assuming that each chip with 640 pixels multiplexes one column of four pixels into one line, then:
  - For each column:
    - First 4 rows of chips have 3 inputs
    - Fifth row of chips has 2 inputs
    - Sixth row of chips has 1 input
  - Total connectivity per Z-column: 17 lines
- Total module connectivity: 17*480=8,160 lines

Step 1: Event from MC

Event:2
Module Position(r,phi,z) : 25.2684 : 0.901326 : -0.19
High
28 28 12164 0.243744
2 36 18855 0.573319
3 66 30925 0.163432
3 67 30925 0.163432
3 68 30925 0.163432
25 89 19225 1.23674
28 93 16268 0.424487
21 98 34681 0.170382
20 99 34681 0.170382
20 100 34681 0.170382
16 167 22324 0.236007
2 192 37920 0.210922
18 192 21067 0.919657
2 193 37920 0.210922
8 193 0 50
16 203 25168 0.703147
26 216 21064 0.518388
20 222 20445 0.174685
Low
28 20 12164 0.243744
3 34 30925 0.163432
3 35 30925 0.163432
3 36 30925 0.163432
15 60 50120 0.142625
15 61 50120 0.142625
15 62 50120 0.142625
15 63 50120 0.142625
15 64 50120 0.142625
15 65 50120 0.142625
15 66 50120 0.142625
21 75 34681 0.170382
21 76 34681 0.170382
24 87 19225 1.23674
28 89 16268 0.424487
16 175 22324 0.236007
2 182 37920 0.210922
18 190 21067 0.919657
7 193 0 50
20 198 22033 0.493651
19 199 20445 0.174685
19 200 20445 0.174685
16 205 25168 0.703147
End

Step 2: Conversion to Verilog



Step 3’: Verilog Model

```
pixel pixel12 (TriggerOut[12], ClusterRjectOut12, PixelUpOut12, LocalHitOut12, CleanPixelOut12, SerialConfOut12
, ConfCkOut12, Ck, ConfCkOut11, SerialConfOut11, Reset, MasterSlave, PixelUpIn12, LocalHitIn[12],
{3'b000,PixelUpOut11,PixelUpOut13,PixelUpOut171,PixelUpOut172,PixelUpOut173}
, {3'b000,LocalHitOut11,LocalHitOut13,LocalHitOut171,LocalHitOut172,LocalHitOut173}
, {3'b000,ClusterRejectOut11,ClusterRejectOut13,ClusterRejectOut171,ClusterRejectOut172,ClusterRejectOut173});
pixel pixel13 (TriggerOut[13], ClusterRjectOut13, PixelUpOut13, LocalHitOut13, CleanPixelOut13, SerialConfOut13
, ConfCkOut13, Ck, ConfCkOut12, SerialConfOut12, Reset, MasterSlave, PixelUpIn13, LocalHitIn[13],
{PixelUpOut-148,PixelUpOut-147, PixelUpOut-146,PixelUpOut12,PixelUpOut14,PixelUpOut172,PixelUpOut173,PixelUpOut174}
,
{LocalHitOut-148,LocalHitOut-147,LocalHitOut-146,LocalHitOut12,LocalHitOut14,LocalHitOut172,LocalHitOut173,LocalHit
Out174}
,
{ClusterRejectOut-148,ClusterRejectOut-147,ClusterRejectOut-146,ClusterRejectOut12,ClusterRejectOut14,ClusterReject
Out172,ClusterRejectOut173,ClusterRejectOut174});
pixel pixel14 (TriggerOut[14], ClusterRjectOut14, PixelUpOut14, LocalHitOut14, CleanPixelOut14, SerialConfOut14
, ConfCkOut14, Ck, ConfCkOut13, SerialConfOut13, Reset, MasterSlave, PixelUpIn14, LocalHitIn[14],
{PixelUpOut-147,PixelUpOut-146, PixelUpOut-145,PixelUpOut13,PixelUpOut15,PixelUpOut173,PixelUpOut174,PixelUpOut175}
,
{LocalHitOut-147,LocalHitOut-146,LocalHitOut-145,LocalHitOut13,LocalHitOut15,LocalHitOut173,LocalHitOut174,LocalHit
Out175}
,
{ClusterRejectOut-147,ClusterRejectOut-146,ClusterRejectOut-145,ClusterRejectOut13,ClusterRejectOut15,ClusterReject
Out173,ClusterRejectOut174,ClusterRejectOut175});
pixel pixel15 (TriggerOut[15], ClusterRjectOut15, PixelUpOut15, LocalHitOut15, CleanPixelOut15, SerialConfOut15
, ConfCkOut15, Ck, ConfCkOut14, SerialConfOut14, Reset, MasterSlave, PixelUpIn15, LocalHitIn[15],
{PixelUpOut-146,PixelUpOut-145, PixelUpOut-144,PixelUpOut14,PixelUpOut16,3'b000}
, {LocalHitOut-146,LocalHitOut-145,LocalHitOut-144,LocalHitOut14,LocalHitOut16,3'b000}
, {ClusterRejectOut-146,ClusterRejectOut-145,ClusterRejectOut-144,ClusterRejectOut14,ClusterRejectOut16,3'b000});
```