Status Report of the SVT external layers

Lorenzo Vitale
University and INFN Trieste

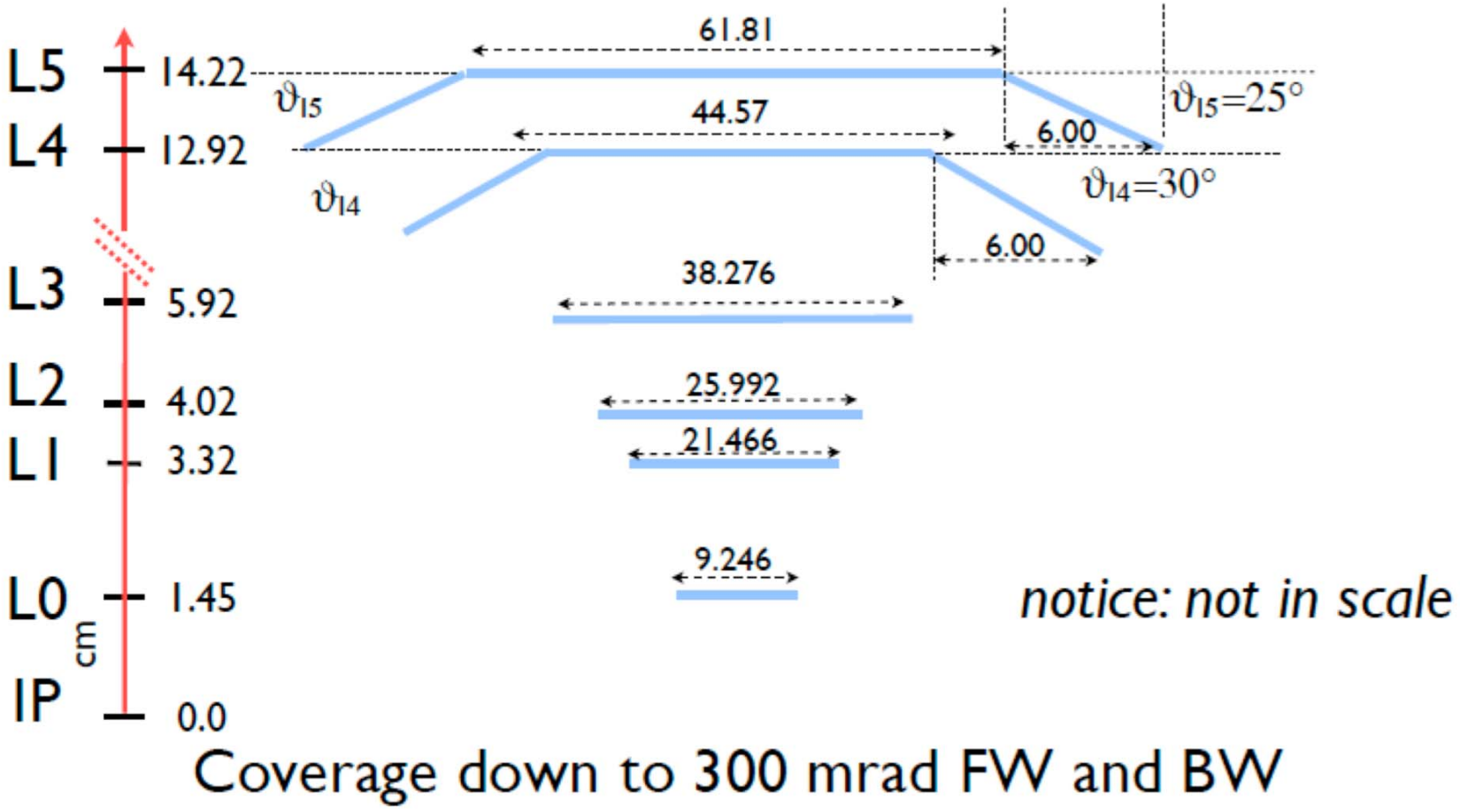
Summary

- SVT Baseline geometry
- Background studies
- Front end chip candidate: FSSR2 from BTEV
  - Can the FSSR2 stand the bkg rate? @ L1 still around 0.7-1.0 Mhz/cm^2
  - (Not in this talk: Is its signal-to-noise adequate for the external layers?)

Geometry

- The baseline is to assume BaBar SVT with two main modifications:
  - Symmetrical layout
  - Angular coverage down to 300 mrad FW and BW

SVT Baseline Geometry



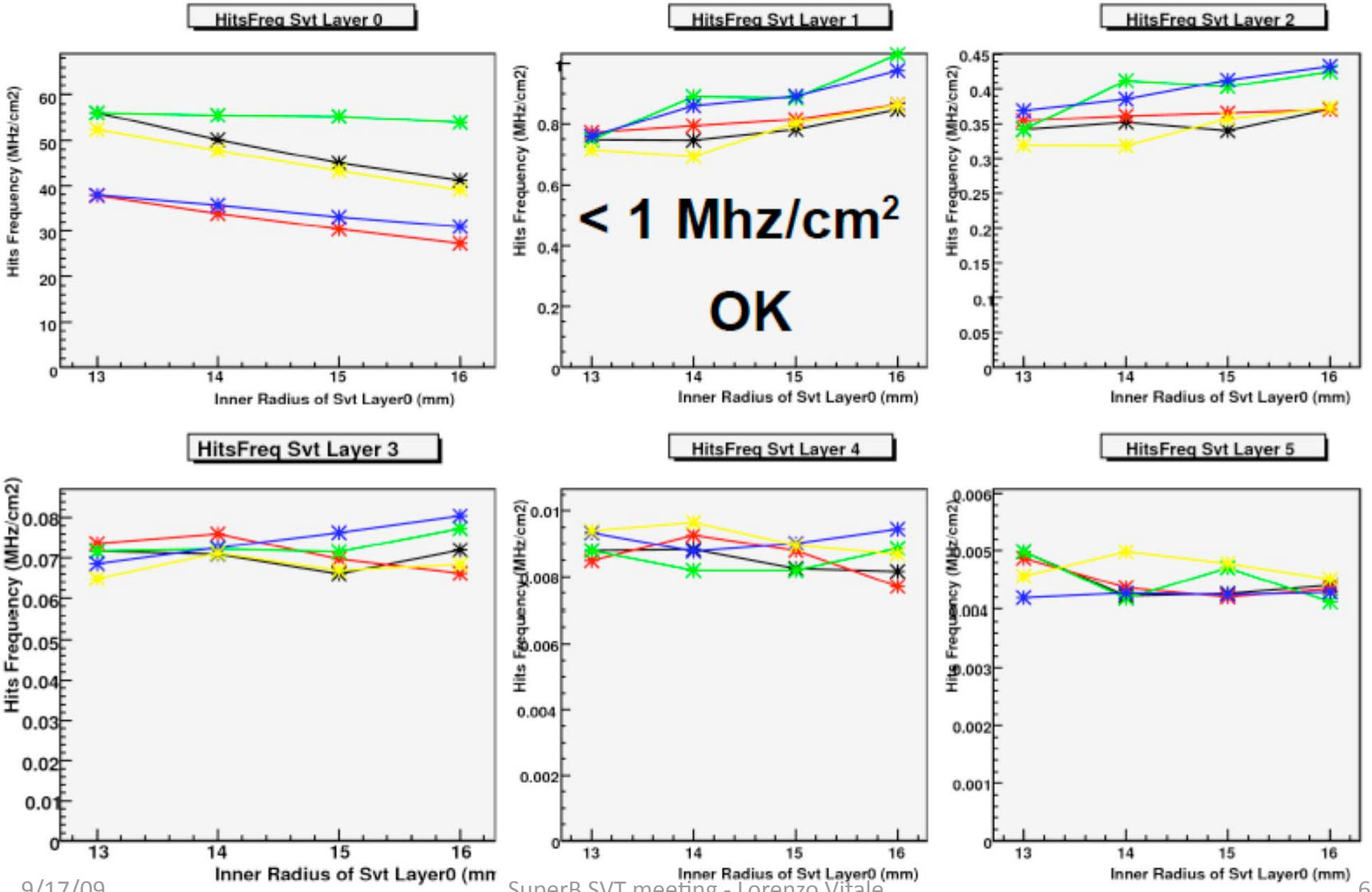
Coverage down to 300 mrad FW and BW

Plot from Nicola Neri

Background Studies

- Riccardo Cenci made a full simulation study (Bruno) with 40k events from the main know background (pair production) with this geometry and 5 configurations:
  1. BP (Beam Pipe) close to L0, BP inner radius is equal to the inner L0 one minus 3 mm (BP thickness + clearance + pin-wheel average)
  2. like black one, with the gold foil moved from BP to inner L0 surface
  3. BP inner radius fixed at 10mm
  4. like green one but the gold foil moved on inner L0 surface
  5. like black but the gold foil is both on the BP and on L0 inner surface

Hit Frequency for pair production (from Riccardo Cenci)



SuperB Front end chip candidate

In the CDR we identified the second release of the Fermilab Silicon Strip Readout chip (FSSR2) as a good candidate for the L1-5 silicon strip detectors (also L0 striplets).

- FSSR2 is completely data driven and data-push (zero suppression) and it can also allow direct use of the detector info in the trigger
- 128 analog channels, but no analog storage, only digital output for fast data output: address, time, and 3 bit amplitude information for all hits above threshold

FSSR2 characteristics

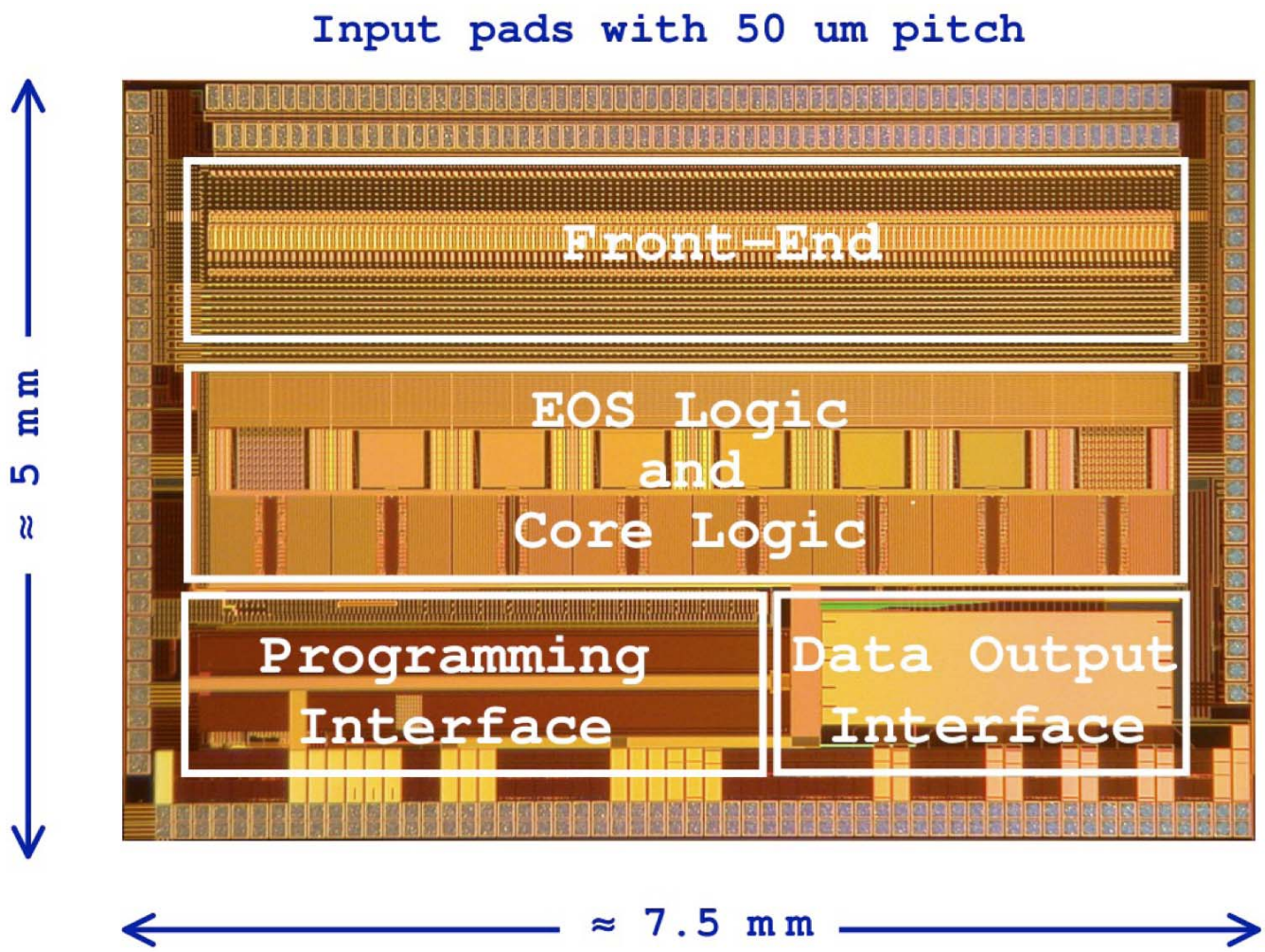
- Settable Thresholds and masking channels
- 3 bit flash ADC
- Internal pulser for calibration
- Tolerance to total ionizing dose (5 Mrad) and specially designed registers to mitigate single event upset (SEU) effects
- Power ~ 4 mW/channel (In BTeV front end electronics cooled to ~-5°C)

- Design spec's: ENC < 1000 e rms @ CD=20 pF Threshold dispersion < 500 e rms

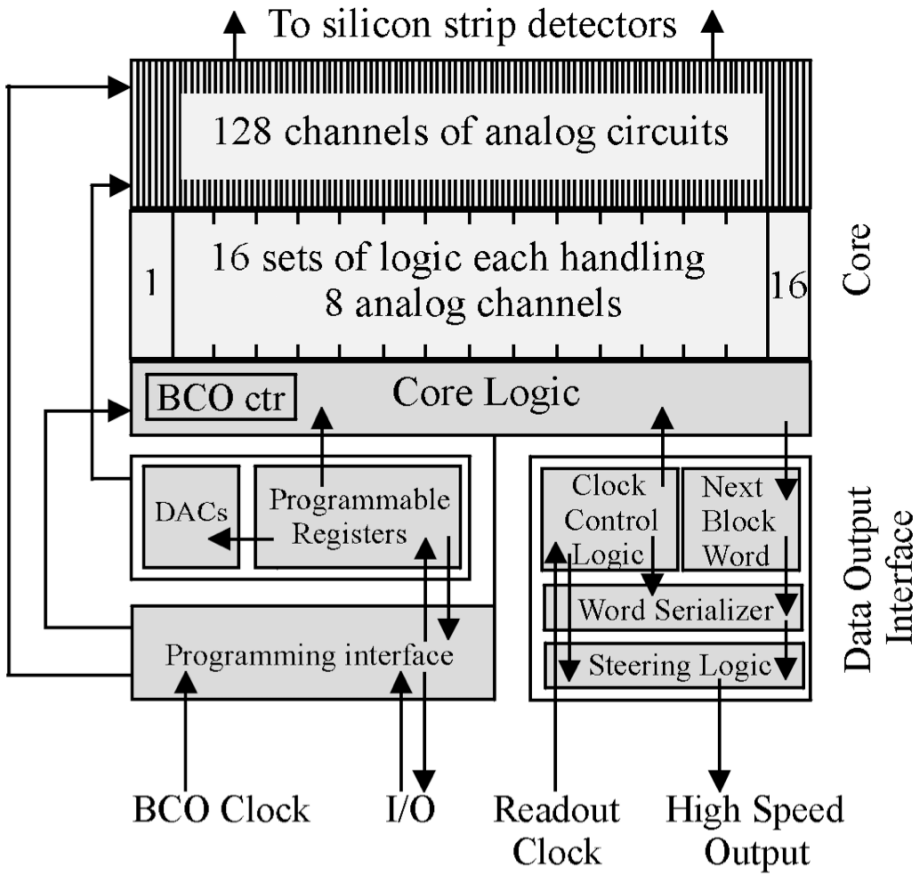
FSSR2 programmable options

- Shaping time (65, 85, 100, 125 ns)
- Gain (two values)
- Internal Baseline restorer selectable
- Data to be read can be split in 2, 4 or 6 lines

FSSR2


Input pads with 50 um pitch
Front-End
EOS Logic
and
Core Logic
Programming
Interface
Data Output
Interface
≈ 5 mm
≈ 7.5 mm

FSSR2 Block diagram



Core circuitry:

- 128 analog channels
- 16 sets of logic, each handling 8 channels (16 “Column State Machines” handling 8 pixel Cells)
- Core logic with BCO counter (time stamp) Command State Machines (4 per Column?)

Programming Interface (PI):

- Programmable registers
- DACs

Data Output Interface:

- Communicates with core logic
- Formats data output
- Same as BTeV FPIX2 chip

FSSR2 digital readout: two State Machines handled by two external Clocks

- BCO clock
  - 396ns BCO* period (2.5MHz), but 132ns (7.5MHz) was also fully supported
  - To program registers and in “Command State machine” logic

*@BTeV BCO true time between beam crossings

- Readout Clock (untied)
  - up to 70MHz + up to 6 parallel LVDS lines allows data output rate up to 840 Mbit/s
  - in “Column State machine” logic (1 column = 8 channels/strips)

FSSR2 Performance Studies

The present knowledge about FSSR2 is based on:

- BTeV documentation: design, simulation studies, laboratory tests in Milano/Pavia

- Experience in Trieste for SuperB, starting in 2007 with the INFN SLIM5 collaboration (Telescope and Striplets modules, used already in 2 Beam tests), unfortunately not yet occupancy studies

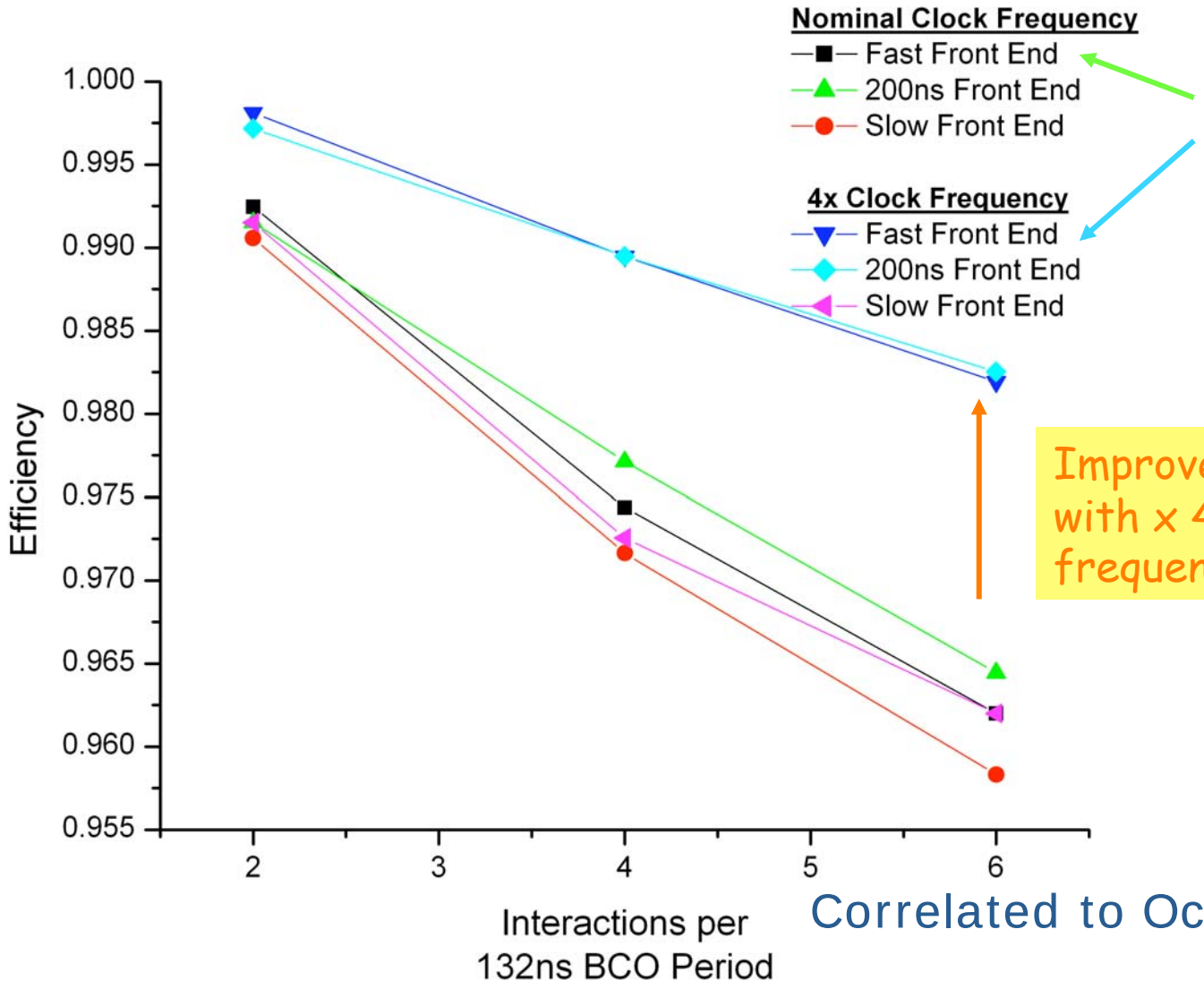
Can the FSSR2 stand the SuperB bkg rate expected in Layer 1?

In Layer 1 we can expect a FSSR2 chip occupancy up to 7.5%

- Assuming bkg rate of 1MHz/cm^2, 2 hit/track, 10.7cm strip length, 50µm pitch, including a safety factor 5 and using in the denominator 7.5Mhz (the maximum BCO clock foreseen fro FSSR2)
  - Is FSSR2 fast enough for SuperB?

- FSSR2 chip was optimized for BTeV operation. Simulations showed:
  - With 2 interactions/bunch crossing (132 ns), BTeV FSSR2 average (?) occupancy 2%
  - Verilog Simulation performed for BTeV with 2 interactions/bunch crossing indicates: FSSR2 can handle 2% occupancy with efficiency > 99% even with a readout clock = BCO clock
  - with 6 interactions/bunch (~6% occupancy?) efficiency can be kept high enough (~ 98.5%) with a readout clock = 30 MHz

FSSR Efficiency Verilog Simulation (Jim Hoff – Fermilab May 2003)



Green and light blue lines corresponds to performance with 65 ns peaking time

Improve efficiency with x 4 BCO clock frequency (30 MHz)

Correlated to Occupancy (%)

Personal remarks

Actually the FSSR2 Readout Clock can be operated even faster than 30MHz (up to 70MHz).

Is it possible to have a faster BCO too?

On the other hand:

- I'm not sure that the quoted 2% occupancy is an average occupancy and not a peak occupancy
- I guess that in a BTeV beam crossing the peak occupancy was dominated by 1 high multiplicity bb event. In this case the peak occupancy is not directly proportional to the number of interactions per beam crossing.

In the BTeV literature I found only other two occupancy studies.

Occupancy studies in BTeV TDR (Sept. 2004)

"BTeV is designed to operate at a luminosity of $2*10^{32}cm^{-2}s^{-1}$ with beam-crossing intervals of 132ns or 396ns".

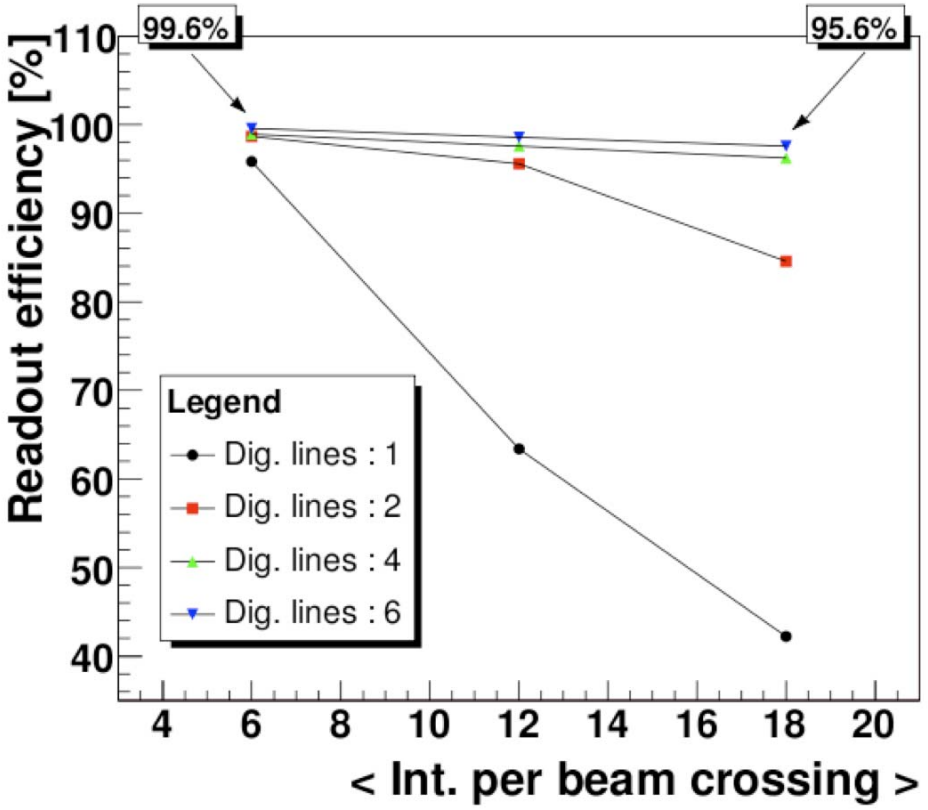
"Forward Silicon Tracker system has a segmentation of 100 µm to handle the high hit multiplicities that are expected when bb events are produced. Indeed, the peak occupancy value in the Forward Silicon Tracker as predicted by BTeV GEANT is only about 2.4%, for a bb event produced at the design luminosity of $2*10^{32}cm^{-2}s^{-1}$, accompanied by an average of six minimum bias events at 396 ns bunch spacing."

From Dinardo Thesis (Dec. 2005)

From Verilog simulations* with:

FSSR2 readout clock 70MHz

BCO period to 396ns (2.5 MHz)



* The input to the Verilog program is a file generated by the simulation framework of BTeV. The file is a list of channels that have been hit by a particle.
Taking the chip with the highest occupancy and chip analog section considered just as a delay in the signal processing chain.

Conclusions

- FSSR2 should be able to operate at >98.5% efficiency with ~6% occupancy, (still ok for layer 1)
- We must operate FSSR2 at its limits: highest BCO clock (132ns), Readout clock at 70MHz and 6 lines, 65ns, BLR
- This aspects should be kept in mind in case of redesign

Now we need:

- Ad hoc simulations (help from experts?)
- Laboratory tests, that we plan to do in Trieste

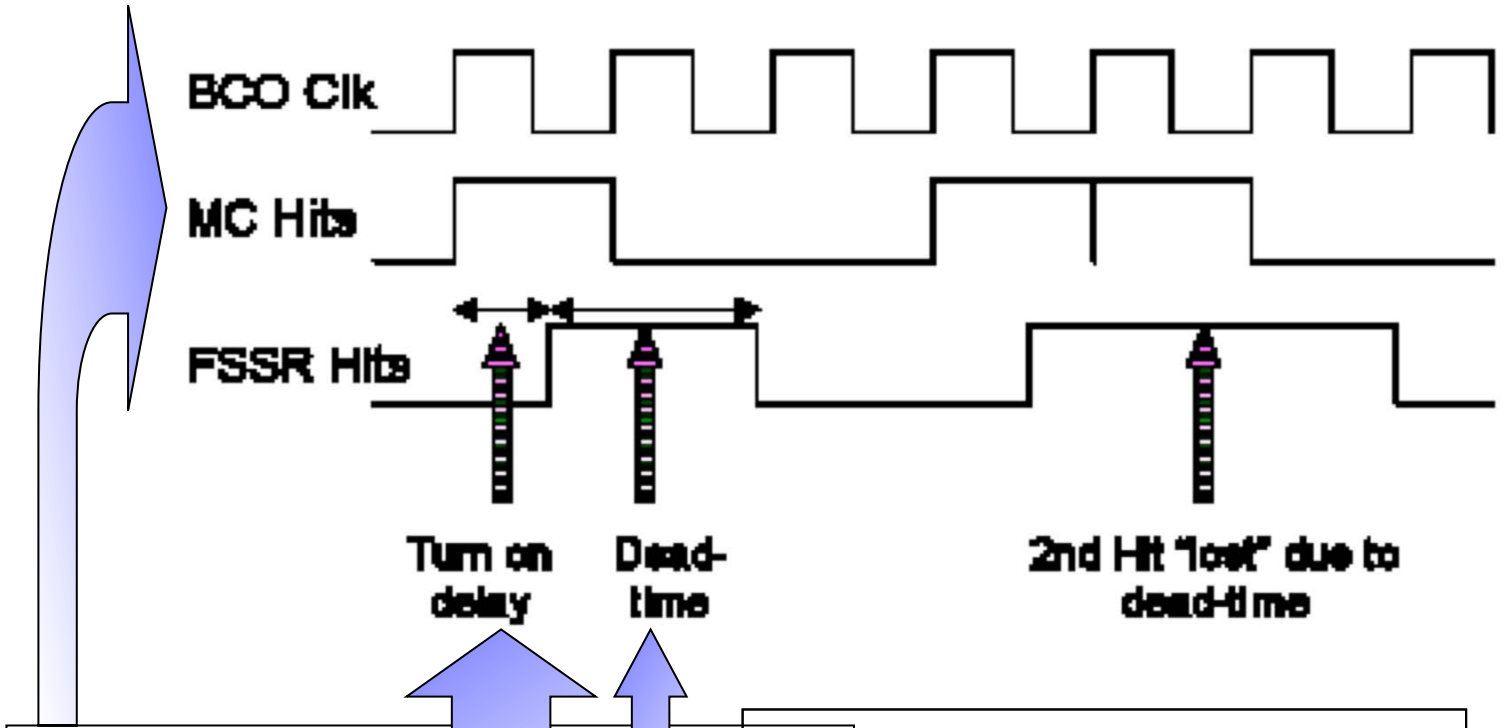
Backup slides

FSSR2 main source of inefficiency: Delays, latency and Dead Time

If I well understand, there are two sources:

- Analog delays (amplification chain). To improve, for high occupancy and short BCO periods, one can choose short shaping time 65ns and BLR.
- Digital logic. The elapsed time between the occurrences of an hit and the output of the data is the sum of latency time for BCO clock period completion, plus one BCO clock period, plus three readout clock periods. During this time one of the state machines is not available.

Silicon Strip Detector Model used in the BTeV simulations (J. Hoff)



Other sources of inefficiency? Big events?

Since the “End of Column” Logic handles 8 channels/strips (technically called “Pixel Cells”), and there are 4 sets of “Command State Machines” per Column, not clear for me how to handle more than 4 channels within one column.
Are the 8 channels within a column contiguous?

Bibliography

- J. Hoff Verilog studies

http://www-btev.fnal.gov/cgi-bin/public/DocDB/ShowDocument?docid=1718

- Silicon chapter in BTeV TDR

http://www-btev.fnal.gov/DocDB/0037/003752/003/silicon-chapter.pdf

- Dinardo PhD. Thesis

http://lss.fnal.gov/archive/thesis/fermilab-thesis-2005-66.shtml

- J. Hoff FPIX/FSSR2 Core architecture

http://www-ppd.fnal.gov/EEDOffice-W/ASIC.../Conf-00-260-E.pdf

Output data

- A readout word consists of 24 bits. It can be either the status/sync word, or the data word.
- Data sent from the chip is not time ordered. The BCO number is used off line.
- The output data bit format changes when different numbers of output lines are used.