Modeling and Simulation of a Readout Architecture for Pixel Detectors

Gustavo I. E. Cancelo*, Sergio Zimmermann**
*Univ. of La Plata/Fermilab, ** Fermilab

Abstract

This paper analyzes in detail some theoretical aspects in the modeling of a proposed readout architecture for pixel detectors. The readout architecture is designed for a chip containing about 3000 pixels of 50μm x 400μm. The main objective is to get the maximum pixel hit readout with the minimum probability of hit loss. The readout architecture is modeled as a Marcov stochastic process. The pixel front-end and readout are simulated and tested with Montecarlo data. The simulations allow to optimize the communication channel bandwidths and local buffering. The probability of system overflow of the simulated system is confronted with the one obtained by modeling.

I. INTRODUCTION

Pixels Detectors are the future for most of the inner tracker and vertex detector systems in high energy physic experiments. The resolution depends on the pixel size and whether only digital or digital plus analog information is provided by the pixel front end amplifier and discriminator cell.

The present work has been done at Fermilab, as part of the specification and design of a pixel device to meet BTeV experiment requirements [1]. Since BTeV plans to use the pixel detector as part of the trigger system the most important requirement is readout speed [2]. The primary goal is to achieve a readout rate to cope with the number of hits generated by a luminosity of $2*10^{32}$ p/cm^2 and a bunch crossing (BCO) time of 132 ns at Fermilb's Tevatron. BTeV pixel's detector consists of 93 parallel planes (31 triplets) of 10 cm by 10 cm placed perpendicul[1]arly to the direction of the beam. As shown in Figure 1, the beam passes through the center of the planes.

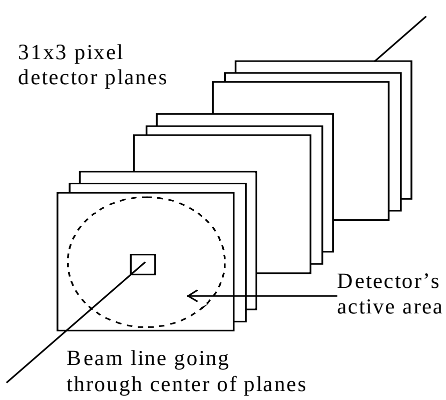

Figure 1: BTeV pixel detector

II. DESCRIPTION OF THE PIXEL READOUT ARCHITECTURE

Figure 2 shows the proposed pixel readout architecture [3]. The pixels are organized by columns. Each column has its own End of Column Logic (EOC) at the bottom. The pixel cells store hit location and a pointer to the Time Stamp (TS) information. This pointer is a two bit register which points to a set of Time Stamp Registers (TSR) in its own EOC logic. Each TSR has its own link which connects it to all the pixel cells in the column. The Pixel Readout Controllers (PRC) readout pixel hits into on chip FIFO buffers. The pixel hit readout is chronologically organized by its time stamp, facilitating the work of the trigger processor and saving time in a very time critical job. Finally, the data is readout off chip from the buffers using a high speed synchronous communication channel. The Output Data Controller (ODC) performs this task.

At readout time, all the columns start, in parallel, the readout of the pixels which match a specific TSR. A token passing mechanism is employed by the EOC logic to locate the hit pixels. A pixel grouping technique with a two level of hierarchy token passing provides a simple and very fast way of locating hit pixels during the readout cycle [3].

The purpose of the current paper is to find a general framework to design a pixel readout architecture subject to the imposed requirements: maximum readout speed and minimum data loss. Data loss is caused by overflows of the internal resources (i.e. no more TSR registers, or FIFO buffers available). An optimization of those resources is mandatory since they increase the so called "dead area" of the chip, the area which cannot be covered by pixel detectors.

The clock frequency and width of the data word provide the maximum or "*peak*" readout hit rate. However, since the hit rate in the pixel array is not a constant function of time, the mean readout hit rate is necessarily smaller. Clearly, in order to maximize the chip's throughput, the Output Data Channel throughput must be maximized. The Pixel Readout links, the TSRs and the FIFO buffers should provide the necessary channel equalization (Fig. 1). The following analysis fixes the internal clock frequency of the Pixel chip to 26.5MHz with the exception of the ODC which runs at 53MHz. The 26.5 MHz frequency was selected based on several facts: it is half the frequency of the Tevatron's master clock used to synchronize the electronics, therefore synchronized to BCOs; it is low enough to be able to manage noise problems; and it will keep the power budget reasonable low.

[1]Work supported by U.S. DOE under contract No DE-ACO2-76CH3000

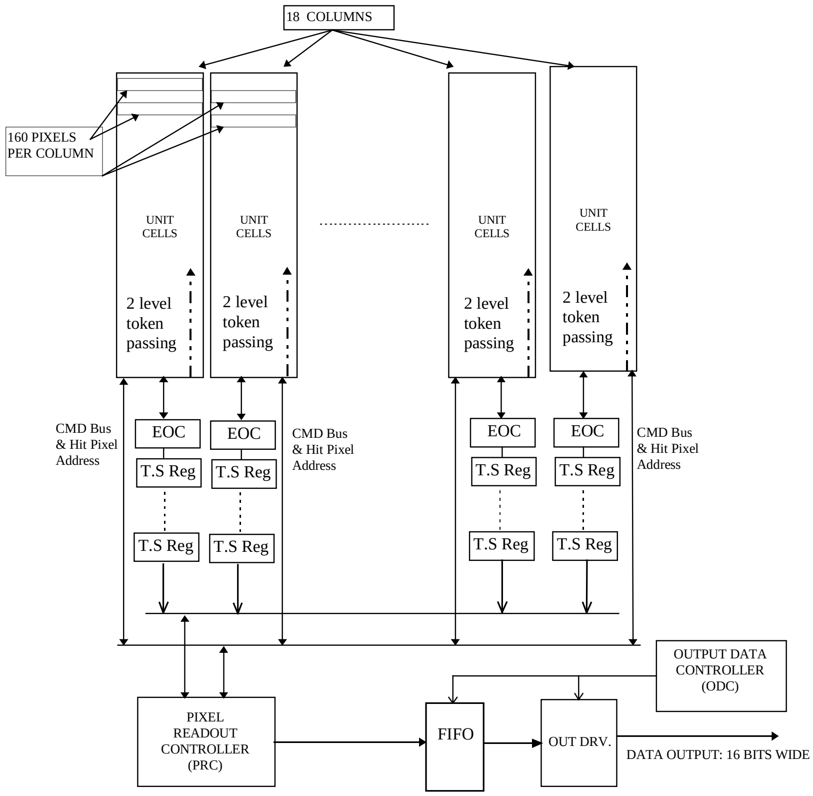

FIGURE 2: Front-end hit discriminator and readout electronics for pixels

III. MODELING OF THE ARCHITECTURE

Let's consider, without loss of generality, a pixel array of 18 columns by 160 rows. The size of the pixel is set to 50μ by 400μ. Then, the size of the chip is 7.2 mm wide by 8.0 mm tall. (Figure 3). The hit probability distribution function (pdf) in the detector planes follows, aproximately, an inverse quadratic law of the distance from the beam to the pixel (Figure 3) [4]:

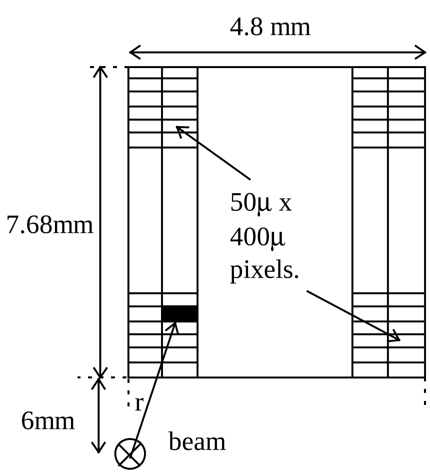

Figure 3: Pixel array and beam possition

$$p(r) = \frac{1}{r^2} \tag{1}$$

where r is the distance from the center of the beam to the pixel. Equation (1) can be expressed in rectangular form:

$$p(x,y) = \frac{1}{x^2 + y^2} \tag{2}$$

where: $r^2 = x^2 + y^2$

The column hit rate can be obtained integrating (2):

In order to measure the Pixel Readout link throughput is necessary to calculate the **pdf** per column in the pixel array:

$$p(x) = \int_{6}^{13.68} \frac{1}{x^2 + y^2}\, dy$$

$$p(x) = \frac{1}{x} \cdot \left[tg^{-1}\left(\frac{13.68}{x}\right) - tg^{-1}\left(\frac{6}{x}\right)\right] \tag{3}$$

Here, *p(x)* represents the hit probability per column and per hit. If hit events are considered independent the total probability per column can be calculated as a binomial process:

$$P(k) = \binom{n}{k} p^k \cdot q^{n-k} \tag{4}$$

where *p*: hit probability in the column of interest and *q=1-p* is the probability of no hit. Also, *n* is the number of total hits and *k* the number of hits in the column under consideration.

$$prob\_hit = 1 - prob\_no\_hit: \quad p_h = 1 - p_{no\_hit} \tag{5}$$

$$P_{no\_hit} = P(0) = \binom{n}{0} p^0 . q^n = q^n \quad (6)$$

For instance, for n=5 hits, the hit probability (P(k>0)) in column 1 is 0.3.

The occupancy of the TSR registers in each column and each FIFO buffer set can be modeled as Marcovian stochastic processes [7]. The modeling is performed separately for every column. Unfortunately, the TSR process and the FIFO process are coupled and influence each other to some extent. To introduce the analysis of Marcov chains, we will first analyze only one of them, unconstrained by the other process. Figure 4 shows a five state homogeneous Marcov chain representing the possible occupancy states of four TSR registers. S0 means that all 4 TSRs are empty and S4 that they are all full.

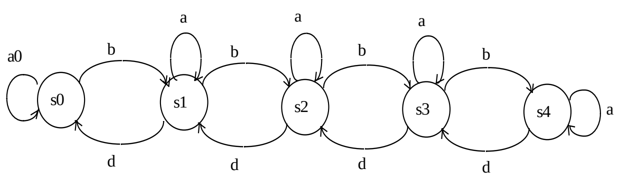

Figure 4: Marcov chain model of 4 TSR registers

b, *d*, *a* and a_0 are the probabilities of jumping to a neighboring state or staying in the current state. The goal is to predict the long term probability density vector and the overflow probability. The current Marcov chain model of the Pixel Readout system is aperiodic and all its states are recurrent. If we define $p_j(n)$ as the probability of being in state *j* at time *n*, we can calculate the long term probability density vector v_i as:

$$v_i = \lim_{n\to\infty} p_j(n) \quad (7)$$

The solution is based on the fact that a stable state must accomplish:

$$v = P.v \quad (8)$$

$$P = \begin{bmatrix} p_{00} & p_{01} & & p_{04} \\ & & & \\ & & & \\ & & & \\ p_{40} & p_{41} & & p_{44} \end{bmatrix} = \begin{bmatrix} a_0 & b & 0 & 0 & 0 \\ d & a & b & 0 & 0 \\ 0 & d & a & b & 0 \\ 0 & 0 & d & a & b \\ 0 & 0 & 0 & d & a \end{bmatrix} \quad (9)$$

where $\boldsymbol{v}$ is the probability density vector of the long term and *P* is the Marcov chain's transition matrix. We see that $\boldsymbol{v}$ is an eigenvector of the *P* matrix. Then, solving for that system:

$$v_i = \left(\frac{b}{d}\right)^i v_0 \quad i = 0,1,..,4 \quad (10)$$

$$v_0 = \frac{1}{\sum_{i=0}^{4}\left(\frac{b}{d}\right)^i} = \frac{1-\left(\frac{b}{d}\right)}{1-\left(\frac{b}{d}\right)^{m+1}} \quad (11)$$

The probability of overflowing is given by making a transition to an imaginary state S5 from S4. This probability is simply $b.v_4$. Then, can be calculated as:

$$P_{ov} = b.v_4 = b.\left(\frac{b}{d}\right)^m . \frac{1-\left(\frac{b}{d}\right)}{1-\left(\frac{b}{d}\right)^{m+1}} = \frac{b^{m+1}(d-b)}{d^{m+1}-b^{m+1}} \quad (12)$$

To find out the probability of overflowing in the particular case of the pixel detector we must find out the values of *a*, *b* and *d*. However, since *p(x)* is monotone decreasing with *x* it suffices to analyze column 1 which gets the highest hit rate.

b represents the probability of having one or more hits in column 1, hence, is a function of the number of hits per BCO (Eq.(5)). Since the pixels are readout in groups of 4 pixels, *b* depends on the group hit rate per Pixel Readout time cycle (37.7ns).

$$P(k)_{37.7ns} = g_h = 1 - g_{no\_hit} \quad (13)$$

The probability of a *d* transition can be calculated by taking conditional probabilities of the columns associated with the PRC, which is encharged of reading column 1. The probability of making a *d* transition is the conditional probability of reading the last word of a particular TSR in column 1 given that PRC is reading column 1 times the probability that the PRC is reading column 1. This can be expressed by:

$$P(d) = P\left(L_g | c_1\right) . P\left(c_1\right) \quad (14)$$

$P\left(c_1\right)$ can be calculated based on the column probability and the number of columns feeding the Pixel Readout controller #1. This probability depends on the hit group rate and the column distribution.

The conditional probability $P\left(L_P | c_1\right)$ depends on the number of hit groups in column 1 for a particular TS, hence, can be expressed as:

$$P\left(L_g | c_1\right) = P(1g) + \frac{P(2g)}{2} + \frac{P(3g)}{3} + \frac{P(4g)}{4} + \ldots \quad (15)$$

where $P(1g)$, $P(2g)$, $P(3g)$ ..., represent the probability of having 1, 2, 3... hit groups of a particular TSR, in column 1. The coefficients 1/2, 1/3, 1/4, etc. represent the probability that the PRC is reading the last pixel of a TSR in column 1, since all the pixels are equally probable.

It is worth noting that for low hit rates $P(4g) << P(3g) << P(2g) << P(1g)$, hence, equation (8) can be very well approximated with the first 2 terms. For instance, for a hit rate of 5 hits/BCO:

$$P(2g) = 0.15.\,P(1g) \text{ and } P(3g) = 0.011.\,P(1g)$$

Combining the values obtained for $P(c_1)$ and $P(L_P|c_1)$ for 5 hits/BCO:

$$P(d) = P(L_g|c_1).\,P(c_1) = 0.94 * 0.2154 = 0.2028$$

Replacing $P(b)$ and $P(d)$ in equation (6):

$$P_{ov} = 0.31\%$$

As said before, the TSR and the FIFO systems are coupled and influence each other. There are two ways of overcame this problem, to find an expression for the modified probability transition matrices of each system or to represent the complete coupled systems in one. The last solution is preferred when the total number of states of the Marcov process is not too big. Figure 5 shows the state transition probability scheme of the complete system. The horizontal transitions represent a change in the state of the TSRs and a vertical transition represent a change in the in the state of the FIFOs.

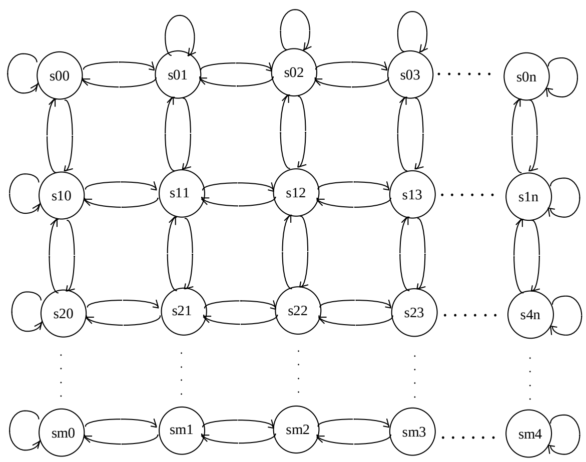

Figure 5: Marcovian model of the complete system

The probability transition matrix has $(n.m).(n.m) = n^2.m^2$ states. Unfortunately, there is not an analytical expression to represent the overflow situations. However, it can solved numerically finding the eigenvectors of the steady state equation (Eq.8) and computing:

$$P_{ov} = b.\sum_{k=1}^{m} P_{kn} \qquad (16)$$

Figure 6 shows the probability of overflow as a function of hit rate for a system with 4 TSRs and 2, 3 and 4 FIFOs.

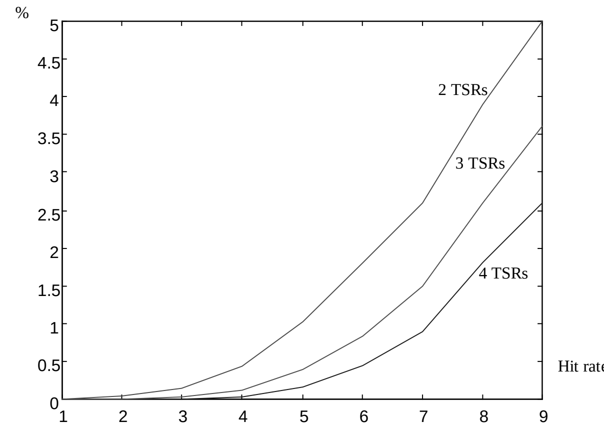

Figure 6: Modeled system overflow

IV. SIMULATION OF THE ARCHITECTURE

The described architecture has been simulated in Matlab [5] in order to quantisize the behavior of the internal variables. The principal parameters under study are: the PRC and ODC communication link throughputs, the token passing latency, the number of hit pixels in the array, the performance of the TSRs and FIFOs and the hit overflow. Two different cases have been simulated.

The first case tests the column architecture using events as similar as possible to the data expected during BTeV experiment's run. These data have been generated by Montecarlo simulation of the BTeV pixel detector [6]. The data simulate 5000 events with 2 and 4 minimum bias particles per BCO, b-quark and c-quark events respectively. Two minimum bias particles per BCO are equivalent to a luminosity of $2 * 10^{32}$ p/cm^2. The collected charge are electrons and the threshold to generate a hit is 2000e-

As described in the next subsection, the column architecture is capable of processing higher data rates than the ones provided by the simulated BTeV events. As a consequence, a second simulation experiment tests the architecture to the limit of its capacity. For this purpose, the data is generated following the basic characteristics of the beam but controlling the hit rate production. The hit distribution has a probability distribution function which follows an inverse quadratic law of the distance between the pixel and beam. Both constant and random number of hits per BCO were simulated at various hit rates.

A. Column architecture simulation results using Montecarlo's input data:

The 5000 event run showed that 6 hits were lost out of a total of 11642 hits. This represents an overflow of 0.05%. Neither the TSRs nor the FIFOs were overflowed. The 6 lost hits were caused by the same pixel hit twice in two consecutive BCOs before the first hit could be readout. In fact, this effect is extremely rare and 0.05% is negligible number. Figure 7 shows the behavior of the PRC and ODC communication links. The utilization of the ODC link is about 35% of its maximum capacity. The architecture is fast enough to read the Pixel hits from the array which is empty 60% of the time.

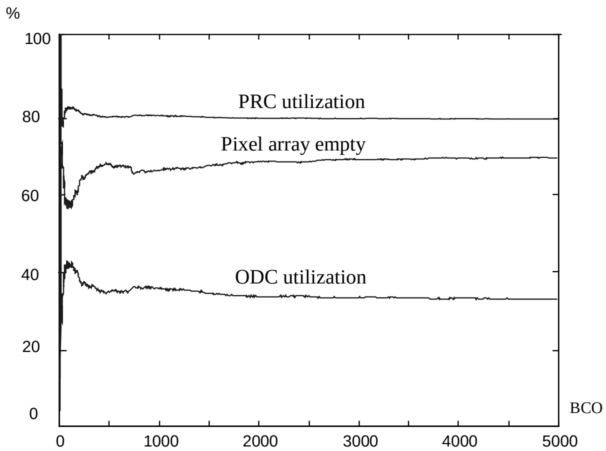

Figure 7: Column architecture channel utilization

The average token passing latency along a column is 16 ns which represents the mean value of the token process by one group times one half of the number of groups in the token passing sequence.

Figures 8 a and b show the TSR and FIFO occupancy. In particular, Figure 8a plots the maximum number of TSR registers used in each column during the 5000 event run. The TSR register occupancy does not exceed 4 registers in any column, which is the upper bound after which the system starts loosing hits. The maximum number of FIFO registers used in this run is 2 which is very low.

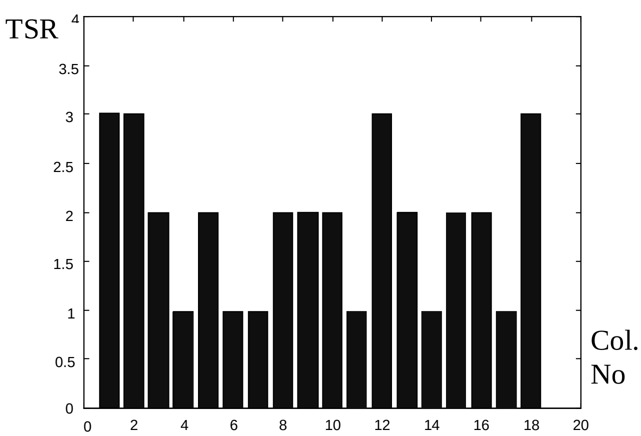

Figure 8: a) Maximum TSR occupancy

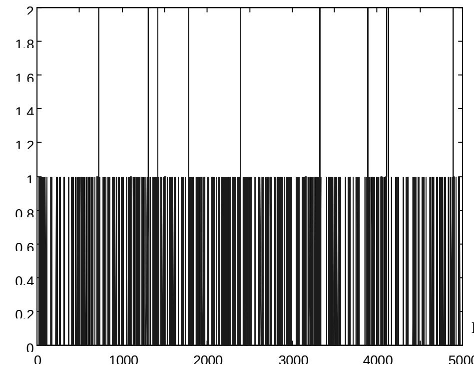

Figure 8: b) FIFO occupancy

Figure 9 shows the number of Pixel hits inside the Pixel Array as a function of BCO. It can be seen that the column architecture does not let accumulate a large number of pixel hits. The large accumulations, like the one at BCO No 685, are caused by large events firing a large number of pixels. The column architecture's response to this impulsive type of input is very good reading out the events in an acceptable number of BCOs.

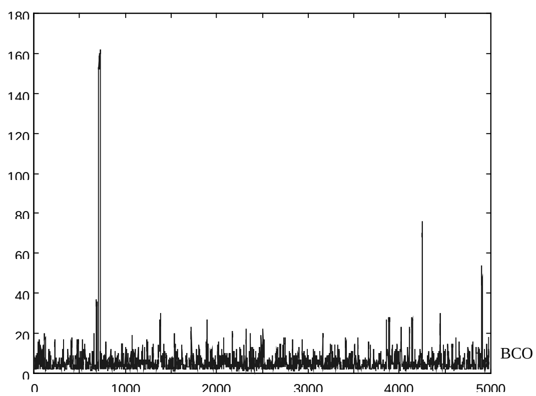

Figure 9: Number of pixels with hits in the Pixel Array

B. Column architecture simulation results using higher hit rate input data:

The most critical parameter is the ODC utilization since this is, by design, the system's bottleneck. The data must be pumped into the FIFO at a rate that can keep the output busy all the time and, in this way, using the full bandwidth of the data channel. Even when the RPC channel is only half the speed of the ODC channel, the data is transferred in a long word including the Pixel group's Column and Row address, and the 4 Pixels' digitized pulse height. The ODC, then, breaks it in two words and sequence them on the output channel along with the Time Stamp information. Figure 10 represents the data throughput of the internal channels, as the cumulative percentage of channel utilization. The input hit rate in this simulation run averages 6 Pixel hits every BCO. As shown, the ODC utilization reaches 100%. In other words, whenever the number of pixel hits is high enough to keep the pixel array not empty, the RPC pumps data into the FIFO faster than the ODC readout speed, and the ODC channel utilization reaches 100% of utilization. A similar result can be observed in The Output Idle column in Table I which shows the % of time the ODC channel is idle.

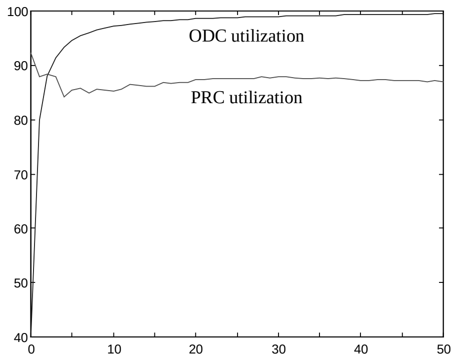

Figure 10: ODC and PRC channel utilization

As important as the data throughput is the ability for the column architecture not to loose hits. As said in Section III, the architecture will loose hits if the TSRs overflow. TableI shows various cases with hit rates between 2 and 3 hit groups per BCO. The architecture can handle up to 2.9 groups/BCO without loosing data. Probabilistic and simulation studies show an average of 2.4 pixel hits per group with hits. This implies an average hit readout rate of 7 hits/BCO. This number certifies why the architecture works at only 35% of its capacity when processing the BTeV simulated data which averages 2.32 hits/BCO. The column architecture's peek hit readout rate, assuming maximum number of pixel hits per group and zero header information in the stream-out is 14 hits/BCO.

The last two rows of the table simulate a random hit rate and cluster size. They show that even at the same average hit rate, large events will increase the readout latency increasing the probability of overflow on posterior events. This is due to the fact that large events increase the instantaneous or short term hit rate. Since the TSRs and FIFO buffers work as a short term equalization, they are sensitive to fast changes in the data rate.

Finally, Table I shows how the system starts overflowing when the hit group rate is raised higher that 2.82 groups/BCO. However, even when the system starts loosing data, it fails gracefully rejecting some events but still working overloaded at its maximum capacity. As the hit rate increases the pixel readout latency will also increase. A large event like the one at BCO No 685 of the current simulation run may take so long that next incoming events start accumulating and overflow the system.

Tracks	Cluster Size	Max.FIFO depth	Max. No hits in array	Max. TSRs Occup.	Overflow (%)	Avrg. No group/BCO	Avrg. No pix/BCO	OutPort Idle (%)
1	5	2	10	1	0	2.		0
1	6	4	12	1	0	2.24	5.96	0
1	7	5	14	1	0	2.51	6.95	0
1	8	10	40	2	0	2.71	7.95	0
1	9	10	120	4	0	2.73	8.94	0
1	10	10	240	5	0.4	2.76	9.93	0
2	2	10	8	1	0	2.49	3.97	0
2	3	10	100	4	0.3	2.74	5.92	0
2	4	10	250	8	1.2	2.79	7.95	0
2	5	10	450	10	2.3	2.82	9.93	0
3	1	10	40	1	0	2.73	2.98	0
Random mean=2	Random mean=3	12	25	2	0	2.08	4.14	4
Random mean=3	Random mean=4	10	38	2	0	2.38	5.15	3

Table I

The results of pixel overflow show a great deal of consistency with the theoretical approach. Figure 11 shows the percentage of hit overflow versus hit rate for both the modeled and the simulated system. The overflow for the average BTeV hit rate is 2.32 hits/BCO is almost negligible and is lower than 0.5% for a hit rate of 6 hits/BCO.

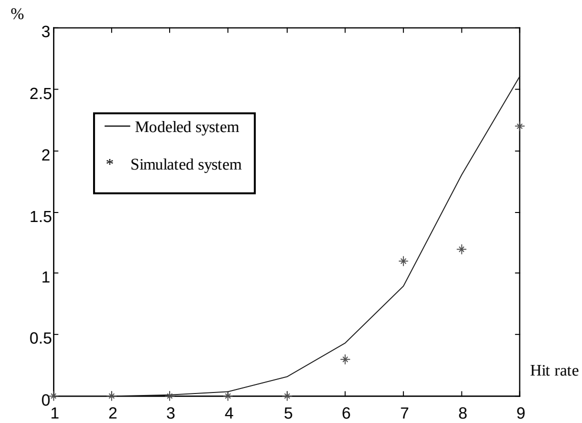

Figure 11: Column architecture system overflow

V. CONCLUSIONS

General aspects in the modeling of a proposed column based readout architecture for pixel detectors have been developed. The readout of this architecture has been modeled as a Marcov stochastic process. Furthermore, the pixel front-end and readout were extensively simulated and tested with various types of data. The readout architecture containing about 3000 pixels of 50µm x 400µm showed to be capable of delivering a higher data rate than the one expected for BTeV experiment. The modeling shows a general path to the analysis of compound Marcovian processes. The simulations provide a good knowledge on the evolution of the internal variables associated with the proposed column architecture. Finally, the comparison between both approaches shows a great deal of consistency on the probability of overflow as a function of hit rate.

References:

[1] Santoro, A., *et al.*, "An Expression of Interest for a Heavy Quark Program at C0," BTeV pub. note, Fermilab, May 1997.

[2] Husby, D., *et al.*, "Design of a secondary-vertex trigger system for a hadron collider," Nucl. Instrum. Meth. A383, pp. 193-198, 1996.

[3] Cancelo, G., *et al.*, "High readout speed pixel chip development at Fermilab," 4th Workshop on Electronics for LHC Experiments, Sept. 21-25, Rome, 1998.

[4] Butler, J., "An initial study of fluence and occupancy in the BTeV pixel detector," BTeV int. Note, Fermilab, Nov. 1997.

[5] Matlab, The MathWorks, Inc., Natick, MA.

[6] Kasper, P., pixel hit data generated from Monte Carlo simulations for minimum bias and b-quark and c-quark events, Fermilab, Mar. 1998.

[7] Trivedi K., Probability and Statistics with reliability queuing and computer science applications, Prentice-Hall, NJ, 1982.