OPTIMIZATION OF A READOUT ARCHITECTURE FOR PIXEL DETECTORS

Gustavo I. E. Cancelo*
*Fermilab

Abstract

This paper analyzes in detail some theoretical aspects in the modeling of a readout architecture for pixel detectors. In fact, these problems are common to the design of data acquisition systems and other processes containing buffers and where the input and output signals can be expressed by probability density functions. It is the purpose of this paper to point out that the same type of analysis can be extended to other systems with the benefit of saving time in long Montecarlo simulations and prototype design. The example case in which this paper is based on is the readout architecture of a column-based pixel detector amplifier and discriminator chip containing more than 3000 pixels of 50μ x 400μ. Several readout strategies are compared searching for an optimal design, which minimizes data loss and maximizes throughput. In particular, the probability of loosing pixel hits by overflowing the readout system is minimized studying the behavior of the stochastic Markov process. Also, the communication channel bandwidths and local buffering are optimized.

I. INTRODUCTION

Pixels Detectors are the future for most of the inner tracker and vertex detector systems in high energy physic experiments. They provide position information in the μm range and a very good signal to noise ratio. The current 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[1] 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 $2*10^{32}\,\mathrm{p/cm^2}$ and a bunch crossing (BCO) time of 132 ns at Fermilb's Tevatron. The current paper is organized as follows. Section II describes the pixel readout architecture. Section III analyses the problem of the buffers used to Time Stamp the pixel hits. Section IV describes the problem of using FIFOs for data equalization and optimization of the data output channel. Section V discuses the Output Data Channel optimization. Finally, Section V summarizes the results.

II.READOUT ARCHITECTURE

Pixel detectors must provide spatial and temporal information of a particle going through. The particle's trajectory is calculated based on the information provided by the hit pixels. If the pixel provides only a binary output, the spatial resolution is directly proportional to the pixel size. Instead, if the energy collected in the group of pixels turned on by a single particle is also measured, the hit can be calculated based on the center of mass of that measurement. The later method improves the pixel's spatial resolution. The current example assumes a chip of 18 columns by 160 pixels. The pixel cell size is kept constant at 50μ x 400μ. The pixel cell provides a digitized value of the collected charge.

As said, 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. registers and buffers available). An optimization of those resources is mandatory since they increase the so-called "dead area" of the chip, the area that cannot be covered by pixel detectors [3].

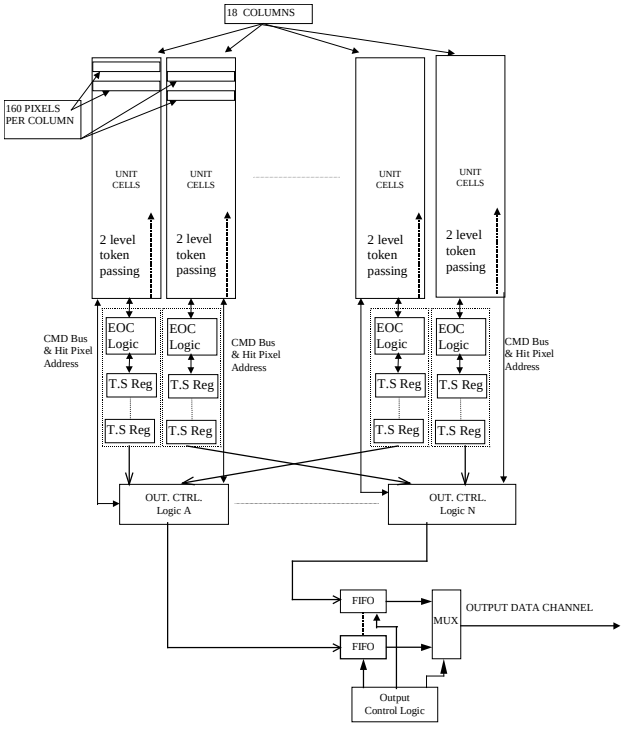

Figure 1: Front-end hit discriminator and readout electronics for pixels

Figure 1 shows one possible readout architecture. This architecture is not guarantied to be optimal but depicts component blocks and points of analysis. The Pixel Array readout is organized in columns. Each column has its own End of Column Logic (EOC) at the bottom. The pixel cells store hit location, a 2 or 3 bit value of the input, and a pointer to the Time

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

Stamp (TS) information. This pointer 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.

Every EOC logic controls a token passing mechanism to locate the hit pixels. A pixel grouping technique with a two level of hierarchy token passing is able to locate the next hit pixel within one clock cycle, during the readout cycle of the previous hit pixel [3].

III. ONE BUFFER OR MULTIPLE BUFFERS?

The readout architecture showed in Figure 1 has one TSR buffer and EOC logic per column. However, any number of TSR and EOC sets can be implemented. The columns can be grouped to Time Stamp and to be readout using any number of TSR and EOC sets. The first part of this section considers the case of a single TSR buffer for the entire chip. The second part considers multiple TSR buffers.

II.1 One buffer

The TSR buffers contribute to the chip's dead area. Hence, its number must be minimized. However, the random nature of the pixel hit rate will make the number of required TSRs to fluctuate in time. If all the TSRs are being used, the incoming hits are lost until a TSR becomes available. This stochastic process can be modeled as a Markov process. A single TSR buffer can be modeled as a *birth-death* process as shown in Figure 2. The TSR buffer's input and output are stochastic variables with Poisson distributions with parameters λ and μ respectively [4]. The TSR registers form a M/M/1 queue.

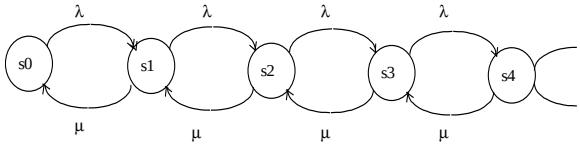

Figure 2: Markovian *birth-death* model for a single TSR buffer

The state equation of an M/M/1 queue must satisfy the Chapman-Kolmogorov equation of state transition. This leads to a complex differential-difference equation. However, the most important system characteristics can be obtained analyzing the steady state of the queue. The balance equations are:

$$(\lambda+\mu).p_k = \lambda.p_{k-1} + \mu.p_{k+1} \quad (1)$$

$$\lambda.p_0 = \mu.p_1 \quad (2)$$

Substituting (2) in (1):

$$p_k = p_0.(\frac{\lambda}{\mu})^k = p_0.\rho^k \quad \rho = \frac{\lambda}{\mu} \quad (3)$$

$$p_0 = 1-\rho$$

It is clear that the system is stable for values of $\rho<1$. That is, when the average input rate is smaller than the average output rate, otherwise the buffer size grows unbounded. ρ also represents the utilization factor, that is the proportion of time the output is busy. The M/M/1 first and second moments can be easily derived:

$$E[N] = \frac{\rho}{1-\rho} \quad (4)$$

$$Var[N] = \frac{\rho}{(1-\rho)^2} \quad (5)$$

Where E[N] is the expectation and Var[N] the variance of the process. According to Little's formula[4]. The average response time can be expressed as: $E[R] = \frac{E[N]}{\lambda} = \left[\mu.(1-\rho)\right]^{-1}$

II.2 Multiple buffers

On the other hand, a multiple buffer TSR set can be analyzed as follows. Figure 3 shows the state transition diagram of a 2 buffer system. The solution can be generalized for m-buffers.

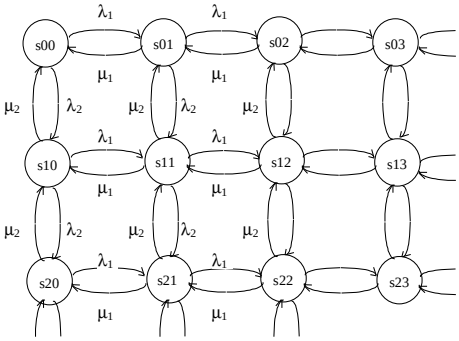

Figure 3 State transition diagram of a 2 buffer system.

The balance equations can be inferred from the transition diagram:

$$(\lambda_1+\lambda_2+\mu 2).p(k_1,k_2) = \\ = \lambda_1.p(k_{1-1},k_2) + \lambda_2.p(k_1,k_{2-1}) + \mu.\left[p(k_{1+1},k_2) + p(k_1,k_{2+1})\right] \quad (6)$$

$$(\lambda_1+\lambda_2+\mu).p(k_1,0) = \\ = \lambda_1.p(k_{1-1},0) + \mu.\left[p(k_{1+1},0) + p(k_1,1)\right] \quad (7)$$

$$(\lambda_1+\lambda_2+\mu).p(0,k_2) = \\ = \lambda_2.p(0,k_{2-1}) + \mu.\left[p(0,k_{2+1}) + p(1,k_2)\right] \quad (8)$$

$$(\lambda_1+\lambda_2).p(0,0) = \mu.\left[p(0,1) + p(1,0)\right] \quad (9)$$

And the normalization factor is $\sum_{\forall k_0,k_1} p(k_0,k_1) = 1 \cdot \quad (10)$

The solution of this system can be found by direct substitution (see [5]) and is expressed as:

$$p(k_1,k_2) = (1-\rho_1).\rho_1^{k_1}.(1-\rho_2).\rho_2^{k_2} \quad (11)$$

Where,

$$\rho_1 = \frac{\lambda_1}{\mu} \quad and \quad \rho_2 = \frac{\lambda_2}{\mu}$$

Since k1 and k2 are independent, this system of two buffers can be generalized to an m-buffer system in the following way:

$$p(k_1,k_2,...,k_m) = \prod_{j=1}^{m}(1-\rho_j).\rho_j^{k_j} \quad (12)$$

Then, the Poisson random variables describing the TSR buffer input and output are statistically independent. The expectation and variance for each TSR buffer can be calculated as:

$$E_i[N] = \frac{\rho_i}{1-\rho_i} \quad where \quad \rho_i = \frac{\lambda_i}{\mu} \quad (13)$$

$$Var_i[N] = \frac{\rho_i}{(1-\rho_i)^2} \quad (14)$$

Several important conclusions can be drawn from the comparison of (4) and (5) with (13) and (14). First, we are interested in the average number of occupied TSR buffers. If a system is designed with only one TSR buffer common to all the columns, then the average number of occupied buffers is given by equation (4). The values of λ and μ can be calculated based on the average number of hits and the average number of TSR released every cycle. λ = p.n/T where p is the probability of having a hit during a certain time interval ΔT, and n is the number of ΔT intervals in T. In the current example, λ depends on the pixel hit rate which for a fixed pixel chip is directly proportional to the luminosity of the accelerator's beam generating the events. The maximum value that λ can reach in the single TSR buffer architecture is given by the use of one TSR every cycle of the accelerator's beam (BCO). A system demanding one TSR every BCO has a λ equal to 7.57. Accordingly, μ is calculated based on the pixel-hit distributions along the pixel chip. These distributions are obtained from simulations. In our case $\mu = p_\mu .n/T = 20.41$

Then, $E[N] = 0.58$

In the multibuffer approach, the expectation values can be calculated based on the input's probability distribution function (pdf). For instance, let's assume one TSR buffer per pixel column as shown in Figure 1. According to [6][7], the hit rate distribution in the pixel detector planes follows an inverse quadratic law respect to the proximity of the pixel to the center of the beam:

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

Where r, x, and y are the coordinates from a pixel to the center of the beam. Since the pixels are handled by column readout, we are interested in all the hits affecting an entire column. Hence integrating along the y-axis:

$$p(x) = \int_6^{13.68} \frac{1}{x^2+y^2}dy \qquad p(x) = \frac{1}{x}\cdot\left[tg^{-1}\left(\frac{13.68}{x}\right) - tg^{-1}\left(\frac{6}{x}\right)\right]$$

The resulting pdf is shown in Figure 4.

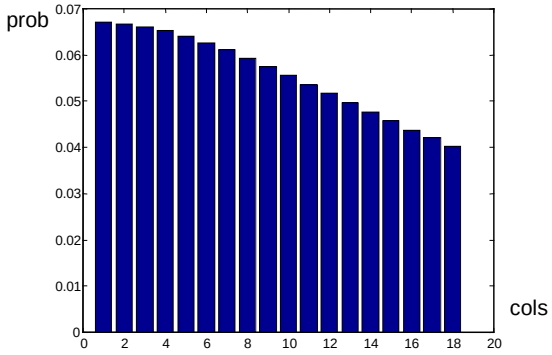

Figure 4 Probability density function of an 18-column pixel chip

It is clear that the total number of TSR buffers used in the multibuffer approach is the sum of the expectation in every column. That is:

$$E_{tot}[N] = \sum_{\forall i} E_i[N] = \sum_{\forall i} \frac{\rho_i}{1-\rho_i} \quad where \quad \rho_i = \frac{\lambda_i}{\mu} \quad (15)$$

The total expectation is calculated based on the λ_i values obtained from the column pdfs functions (Figure 4) and the already calculated μ.

$$E_{tot}[N] = 2.08$$

It is easy to see that in the case of the pixel detector chip, if every column is assigned to its own TSR set, when a beam event hits more than one column, the same Time Stamp will be stored in as many registers as columns are hit. In average, the $E_{tot}[N]/E[N]$ ratio equals the average column hit rate per bunch crossing of the accelerator. In other words it is a function of the accelerator's luminosity. This ratio can easily reach 5 for the expected luminosity of the Tevatron at Fermilab ($2*10^{32}$ p/cm²).

As important as the average number of occupied buffers is its dynamic fluctuation. It was said that if the number of TSR registers is overflowed all new data is lost until the readout system releases a TSR. A good measure to this problem is provided by the Variance of the process. The number of required TSR buffers can be calculated based on the 1st and 2nd moments and the maximum data loss allowed in the system. Let's allow a maximum data loss of 10exp(-5). This is equivalent to an entry of 3.12 in the Gaussian error function. In other words the buffer must be at least 3.12*σ+Ei[N] deep. Where σ is the standard deviation. This value can be used to calculate the maximum ρ value (ρ=λ/μ) allowed. In the current example, since the ρ values are already given, we can calculate the minimum buffer size to keep the data loss less than a certain value (p.e. 10exp(-5)). In the single TSR buffer approach the minimum buffer depth (MBD) is:

$$MBD[N] = inv\_erf(mx\_error)*\sigma + E[N] = 3.12*\sigma + 0.58 = 3.4 \quad (16)$$

In the multiple TSR approach, the buffers must be considered individually since they may overflow separately. However, since the hit distribution is monotonic decreasing with the distance to the beam it suffices to analyze the busiest column (i.e. column 1). Then,

$$MBD_1[N] = inv\_erf(mx\_error) * \sigma_1 + E_1[N] = 3.12 * \sigma_1 + 0.15 = 1.4 \quad (17)$$

IV. THE FIFO BUFFERS:

The pixel data is readout off chip using a single high-speed synchronous communication channel, the Output Data Controller (ODC). In order to optimize the overall pixel readout system's throughput the ODC utilization must be as close to 100% as possible. This can be achieved having one or more FIFO buffers inside the pixel chips to equalize the random pixel hit data flowing from the pixel array. Different schemes can be analyzed based on the two-buffer tandem system. The simplest system contains one TSR buffer and a FIFO buffer. Other systems with multiple TSRs and FIFOs can be calculated using the corresponding λ_i and μ_i.

The two-stage tandem network is shown in Figure 5a. The state diagram is shown in Figure 5b. The system has some similarities with the one described by (11). This system has 2 buffers in series instead of parallel.

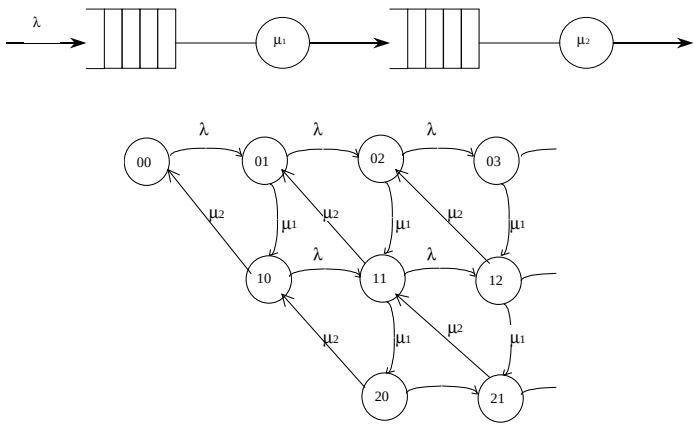

Figure 5a) Two-stage tandem network. b) State transition diagram

The balance equations can be inferred from the transition diagram:

$$(\mu_1 + \mu_2 + \lambda).p(k_1,k_2) = \\ = \mu_1.p(k_1+1,k_2-1) + \mu_2.p(k_1,k_2+1) + \lambda.p(k_1,k_2+1) \qquad k_1 > 0 \quad k_2 > 0$$

$$(\mu_1 + \lambda).p(k_1,0) = \mu_2.p(k_1,1) + \lambda.p(k_1-1,0)$$

$$(\mu_2 + \lambda).p(0,k_2) = \mu_1.p(1,k_2-1) + \lambda.p(0,k_2+1)$$

$$\lambda.p(0,0) = \mu_2.p(0,1)$$

And the normalization factor is $\sum_{\forall k_0 k_1} p(k_0,k_1) = 1$.

The solution of this system can be expressed as:

$$p(k_1,k_2) = (1-\rho_1).\rho_1^{k_1}.(1-\rho_2).\rho_2^{k_2} \quad where \quad \rho_1 = \frac{\lambda}{\mu_1} \quad \rho_2 = \frac{\lambda}{\mu_2} \qquad (18)$$

Again, we find that ρ_1 and ρ_2 are independent in the steady state. The expected number of registers occupied in each buffer can be obtained analyzing the marginal probability $P(N_1 = k_1) = p(k_1) = (1-\rho_1)\rho_1^{k_1}$. The last equation does not reflect the overflow problem in the TSR buffers since the state diagram was not bounded. It is worth to mention that no overflow occurs in the FIFO buffers since the data can be held in the pixels until the FIFO full condition goes away. The overflow can be calculated limiting the size of the buffers. Let's assume that the sizes of the TSR and FIFO buffers are **n** and **m** respectively. Then the overflow is:

$$P_{ov} = P(k_1 > n) = \sum_{k_1 > n} \sum_{k_2=1}^{m} p(k_1,k_2)$$

$$= \lambda.p(n,1) + \lambda.p(n,2) + ... + \lambda.p(n,m)$$

$$= \lambda.\sum_{k_2=1}^{m} (1-\rho_1)\rho_1^n (1-\rho_2)\rho_2^{k_{21}}$$

$$= \lambda.(1-\rho_1)\rho_1^n (1-\rho_2)\sum_{k_2=1}^{m} \rho_2^{k_{21}}$$

$$= \lambda.(1-\rho_1)\rho_1^n (1-\rho_2)\frac{1-\rho_2^{m=1}}{1-\rho_2}.$$

$$P_{ov} = \lambda.(1-\rho_1)\rho_1^n (1-\rho_2^{m=1}) \qquad (19)$$

Figure 6 shows the probability of overflowing the TSR buffer as a function of the number of registers in that buffer. The number of registers in the FIFO buffer is used as a parameter. The plots show that the Pov decreases when the number of registers in either buffer is increased.

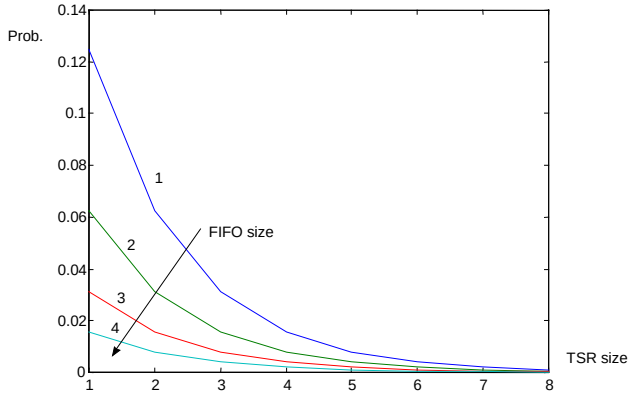

Figure 6 TSR overflow probability function

The TSR and FIFO buffer system overflow (19) was obtained analytically from equation (18). However, sometimes

$p(k_1,k_2,\ldots,k_n)$ does not have an analytical expression and must be represented by its transition matrix:

$$P = \begin{bmatrix} p_{00} & p_{01} & \cdot & \cdot & p_{0n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ p_{m0} & p_{m1} & \cdot & \cdot & p_{mn} \end{bmatrix}$$

then, the probability distribution function of the steady state can be found taking

$$v = \lim_{n\to\infty} p(n) \tag{20}$$

Where *v* represents the steady state of the system and *P(n)* is the n-iterated transition matrix. It evident that for $n\to\infty$ the solution to this system must accomplish:

$$v = P.v \Rightarrow (I - P).v = 0 \tag{21}$$

The last equation can be solved analytically or numerically. Then, the overflow can then be obtained computing:

$$P_{ov} = \lambda.\sum_{k=1}^{m} P_{kn} \tag{22}$$

V. MAXIMIZATION OF THE OUTPUT CHANNEL UTILIZATION

The last point in the analysis of this architecture concerns the output data channel (ODC). It is clear, that from the system's performance point of view the utilization of the output channel must be maximized. The utilization is maximized by reading-out data from the pixels, as soon as possible and keeping the FIFO(s) not empty, unless the pixel array is empty. The analysis of this system is similar to the one of Section III. The system can be designed with one or more FIFOs. The equations describing the probability distribution function, and moments are the same as (3), (4) and (5) for the single buffer and (12), (13), and (14) for the multi FIFO approach. It is obvious that in order to maximize the utilization of the output channel, the speed of the data input to the buffers must be higher than the speed of the output. Then, the output is the bottleneck of the system and the FIFOs are loaded unless the pixel array is empty. This condition makes the data output work close to 100% of utilization. The Utilization (U) of a single FIFO system (M/M/1 queue) is equal to ρ. If $U=\rho\geq 1$(i.e $\lambda\geq\mu$) the system becomes marginally stable or unstable. Since the FIFOs do not loose data, as is the case with the TSR buffers, this is not an undesired condition.

The output data channel is usually designed based on system constrains such as data path width and data output clock rate. This parameters are imposed by the mechanical and electrical characteristics of the inter-connective system, and they define the μ of the Poisson distribution of the output channel. The λ will depend on the input's architecture. That is the number of FIFO buffers. The convenience of having one or more FIFO buffers has a similar analysis to the one in Section III. A single buffer system with a data input rate λ equivalent to the sum of the individual input rates λ_i of the multi-buffer system, will minimize the buffer size and, hence, the dead area.

In practice, a high speed and small size (i.e. 8-bits, 100MHz clock) data output is preferred due to system integration issues. A single buffer can keep $\rho\cong 1$ by having an internal wider data bus, transferring all the information corresponding to one pixel, or a group of pixels, at once (i.e. pixel coordinates, Time Stamp, and digitized input value).

VI. CONCLUSIONS

This paper analyzed some theoretical aspects in the modeling of a readout architecture for pixel detectors. Most of this analysis can be extended to data acquisition systems where buffers are used to equalize data flows coming from different sources. The analysis of systems as random processes can be very advantageous, avoiding long Montecarlo simulations and costly prototype designs. The particular case study in which this paper is based on, the readout architecture of a column-based pixel detector amplifier and discriminator chip, has shown that a single TSR buffer will minimize the dead area, minimizing the total number or buffer registers, while providing the same performance. A measure of relative buffer size can be defined as:

$$BSR = \frac{\left\lceil \sum_i MBD_i[N] \right\rceil}{\lceil MBD[N] \rceil},$$

where |MBDi| and |MBD| are integer numbers representing the minimum buffer size needed. The BSR of one TSR buffer per column versus a single TSR buffer is 9. Section IV showed how to design a system to minimize overflow and what is the influence of TSR and FIFO buffer size in the overflow measure. It was also shown that a single FIFO system can keep very small overflow rates with small size buffers Finally, Section V showed that a saturated system is convenient to maximize the output channel utilization.

REFERENCES:

[1] BTeV: An Expression of Interest for a Heavy Quark Program at C0, BTeV collaboration, Fermilab, May 18, 1997.

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

[3] FPIX1 Pixel Readout Chip, A.Baumbaugh, D.Christian, G.Cancelo, J.Hoff, A.Mekkaoui, R.Yarema, S.Zimmerman, Fermilab Internal Document, Feb. 11, 1988.

[5] Cancelo, G. “Stochastic models for a Pixel Readout architecture”, Fermilab report, July, 1999.

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

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