An on-chip charge cluster reconstruction technique in the miniVIPIC pixel readout chip for X-ray counting and timing

J. Hoff, G. W. Deptuch, *Senior Member, IEEE*, F. Fahim, P. Gryboś, *Member, IEEE*, P. Maj, *Member, IEEE*, D. P. Siddons, *Member, IEEE*, R. Szczygieł, *Member, IEEE*, M. Trimpl, T. Zimmerman

***Abstract*– An on-chip algorithm for the allocation of a hit to a single pixel in the presence of charge sharing in a highly segmented pixel detector is presented. It has been developed to advance pixel detector technology for experiments with X-ray beams at a synchrotron facility. Its key elements are: activation of groups of pixels (*neighborhood_active),* comparisons of peak amplitudes within the active neighborhood, virtual pixels that recover composite signals, ability to create event driven strobes to control comparisons of fractional signals between neighboring pixels and finally latching of the results of these comparisons. The miniVIPIC prototype was designed in a 130 nm process, as a proof of feasibility. The chip contains an array of 32×32 100×100 μm^2 pixels. Analog and digital signals are exchanged between pixels, forming an extensive inter-pixel connection grid, whose routing to minimize parasistics, represented the major challenge. The design details of the chip are provided.**

I. Introduction

POSITION sensitivity, timing and amplitude spectroscopy are typical functions of pixelated X-ray detectors used as scientists' eyes in experiments, where samples to be studied are illuminated by brilliant synchrotron radiation beams. Charge that is liberated in interactions between photons and the detector material is typically collected with the assistance of an electric field. In highly granular systems, the charge drifting towards the detector electrodes is not collected on one electrode, but is distributed among several [1]. If a pixel pitch is significantly smaller than the thickness of a sensor, a substantial number of events result with charge shared among adjacent pixels. As a result, the pixel electronics of a classical pixel detector has only fractions of the whole charge available for processing. This leads to the registering of excessive hits from fractional signals interpreted as coming from real photons. Unfortunately threshold levels cannot be increased freely to compensate for this effect, as it would lead to loss of real hits. Measurements of energies of incident photons are also imprecise. In addition, splitting of charges between multiple electrodes may lead to inaccurate determination of photon time of arrival (ToA) measurements due to time walk. ToA measurements can be made insensitive to signal amplitudes by using, for example, the constant fraction discrimination (CFD) method [2]. However, this technique requires delay elements and extensive electronics that cannot be fit into a small pixel footprint. Assuming monoenergetic radiation incident on the detector, an alternative approach is to reconstruct the full signal from the charge fractions and use the leading-edge technique for ToA measurement [3]. A full pulse-signal with a steep slope frees discrimination from the amplitude dependent time-walk and eventually results in accurate ToA measurement.

This paper presents an idea that has been developed to advance pixel detector technology for experiments using X-rays at a synchrotron facility, by allowing simultaneous ToA measurements and allocation of a hit to a single pixel in the presence of charge sharing in a highly segmented pixel detector. The detector described here registers only one hit per photon despite the fact that fractional signals may also exceed detection thresholds.

The work has direct applicability to the detection of X-ray photons, but it can also be useful for other applications, e.g. in tracking of relativistic charged particles. Such particles traversing a detector cause generation of charge that is collected in groups of neighboring pixels. Delineating boundaries of clusters and sending minimal information off the detector, i.e. an address of a single pixel hit, saves transmission bandwidth and may be useful in triggering, where coarser spatial resolution is generally satisfactory [4].

The emphasis of this paper is on the feasibility of an on-chip implementation of the proposed algorithm. It is organized in 5 sections. The first section is an introduction. The second section provides a description of a hardware realizable algorithm, named C8P1. Distinctive features of the algorithm, terms required for understanding of the flow of action following an impact of a photon according to the algorithm

Manuscript received December 6, 2014; revised XX, 2014.

Fermilab is operated by Fermi Research Alliance, LLC under contract No. DE-AC02-07CH11359 with the U.S. Department of Energy. BNL is supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-98CH10886. AGH-UST is supported by National Science Center, under Contract DEC-2011/01/B/ST7/05155.

G. W. Deptuch, F. Fahim, J. Hoff, M. Trimpl, T. Zimmerman are with the ASIC Development Group of the Electrical Engineering Department of the Particle Physics Division at the Fermi National Accelerator Laboratory, BP 500, MS 222, Batavia, IL 60510, USA, (telephone: +1 630 840 4659, fax: +1 630 840 2950, e-mail: deptuch@ieee.org, farah@fnal.gov, jimhoff@fnal.gov, trimpl@fnal.gov, tzimmer@fnal.gov).

D. P. Siddons is with the Photon Sciences Directorate at the Brookhaven National Laboratory, BP 5000, Upton, NY 11973, USA, (e-mail: siddons@bnl.gov).

P. Gryboś, P. Maj and R. Szczygieł are with the Department of Measurement and Electronics at the Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering at the AGH University of Science and Technology, al. A. Mickiewicza 30, 30-059 Kraków, Poland, (pawel.grybos@agh.edu.pl, piotr.maj@agh.edu.pl, robert.szczygiel@agh.edu.pl).

and block-level representations of analog and digital parts of a pixel are described. The third section discusses practical aspects of forming activated groups of pixels following impacts of photons for various scenarios of photon impacts in deeper detail. The fourth section discusses the design of the miniVIPIC prototype that was designed in a 130 nm process aiming at delivery of a proof of feasibility of the hardware implementation of the C8P1 algorithm. The last section of the paper concludes and summarizes the material presented.

II. The C8P1 Algorithm

A. Motivation and Basics

The objective is to assign an individual hit to a particular pixel and to stamp it with the appropriate time. The goal, which is actually intuitively obvious, is to assign a hit to the pixel at or nearest to the center of the photon impact (or particle passage). Charge generated in a sensor falls off with radial distance from the center of the photon impact. Therefore, it is logical to assume that of a group of pixels sharing charge from an interaction, the *center pixel* would be that pixel that has the largest charge deposited upon itself. Consequently, the objective is to find the single pixel in an area with the largest charge deposition upon itself and to stamp the pixel as accurately as possible to the time of the event. To realize this objective, the algorithm must be implementable in an on-chip signal processing chain and it should use a minimum amount of circuit resources.

Typical signals on an output of a continuous time front-end chain in a radiation detector have some form of semi-Gaussian waveforms, e.g. CR-RC^n [5]. It is obvious that finding the one pixel to which a hit would be allocated requires comparing signal between pixels. Comparisons of such signals can be performed between their peak amplitudes or between their respective times-over-threshold. A comparison of time-over-threshold, particularly in the presence of noise, generally performs worse, as is illustrated in Fig. 1. Waveforms with added noise envelopes visibly overlap as they return to baseline and drop below threshold. The situation depicted in Fig. 1 is drawn for CR-RC^2 signals that peak at 250 ns and is for the sake of reference only. It can be seen that peak amplitudes can unambiguously be separated (amp2>amp1), while looking at times-over-threshold can give an incorrect result, i.e., a weaker signal may yield larger time-over-threshold than a stronger signal (tot2<tot1). Thus, contrary to other efforts [6], it was decided, to use comparisons of signal amplitudes [7], considering such a choice optimal for the work presented in this paper.

An ideal algorithm for registering hits can be easily defined by explicitly stating a start time, naming the pixels to be compared, and finally latching the result and marking the selected pixel as being hit. However, an implementation of such a theoretical algorithm into an on-chip signal processing chain faces a few challenges. First and foremost is the fact that photon arrival is generally an asynchronous process and secondly, the handling of signals in real circuit blocks suffers from delays, of offsets, and noise.

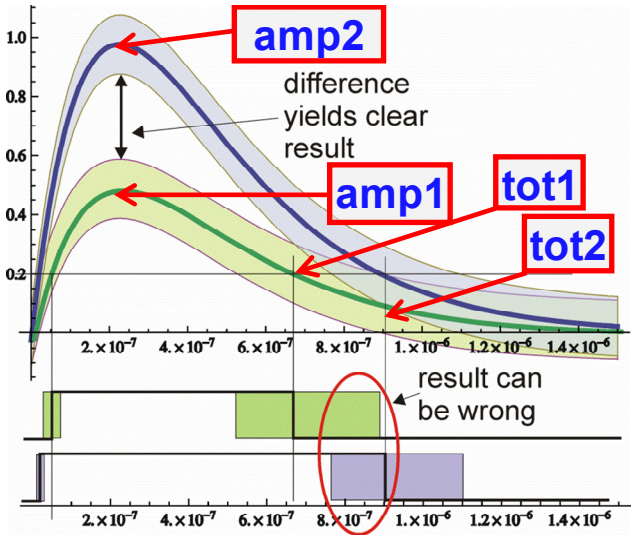

Fig. 1. Contrasting comparisons of portions of collected charge using CR-RC^n shaped signals based on their amplitudes to their durations of time-over-threshold in presence of noise.

An internally generated control strobe signal is necessary to initiate and to conclude the comparisons among adjacent pixels. This control strobe must be derived directly and as quickly as possible from the asynchronous arrival of a photon. The fractional charges collected on adjacent pixels do not guarantee timely stamping of the photon arrival times. Thus, such strobes are created from signals recomposed from the fractions coming from neighboring pixels [7]. This composite signal, shaped in a filter with a fast response time, leads to the prompt firing of the discriminator and the generation of the strobe. The rising edge of the strobe signals ToA, while the falling edge latches a status of comparisons of signals between neighboring pixels. Adjusting the timing properties of the composite signal and of the filtering of fractional signals in such a way that latching occurs at maximum amplitudes of the latter fulfils the requirement of optimal comparing of peak amplitudes. Fig. 2 shows the temporal relationship between the strobe signal, resulting from discrimination of the composite signal, which is shaped in a filter with a fast time response, and two fractional A and B signals from neighboring pixels. The A and B signals are shaped in a filter with a relatively slow time response. The comparators operating on composite and fractional signals from neighbors need to have their offsets removed. The solution proposed consists of using offset an trimmed digital-to-analog converter (DAC) for the discriminator operating on the composite signal, and auto-zeroed comparators for the weighting of signals from neighbors.

It is worth highlighting that reconstruction of full signals, for ToA measurements needs to occur with minimum offsets, avoiding any DC level or amplitude trimming and respecting the temporal nature of the fractional signals. For example adding signals in a DC current mode comes naturally, but achieving high speed operation requires large bias currents, whose offsets may be larger than the processed signal. AC-coupled injections of fractional signals from a first stage amplifier onto an active integrator or band pass filter,

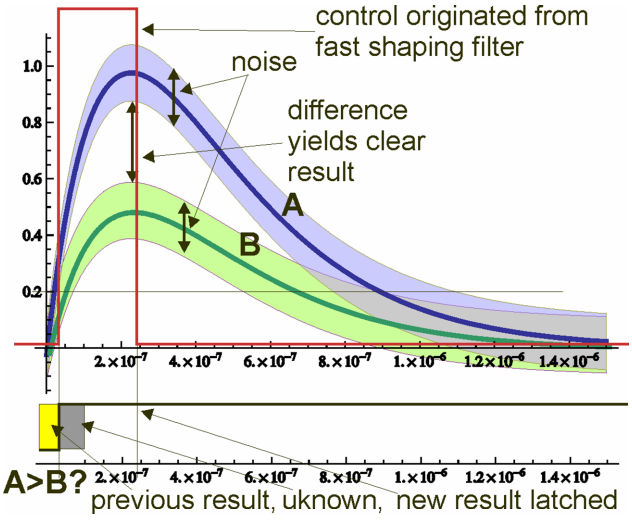

Fig. 2. Temporal relationship between the discriminator signal (strobe), resulting from discrimination of the composite signal, shaped in a filter with a fast time response, and two exemplary fractional A and B signals from neighboring pixels, shaped in a filter with a slow time response.

producing a voltage response, across series capacitors was chosen as the best solution for the attempted implementation.

B. Definition of Terms for the C8P1 Algorithm

In order to present the details of the flow of the C8P1 algorithm the following terms are introduced first: *neighborhood* (activation and evaluation), *virtual pixel*, P1 (activation) and C8 (evaluation).

The C8P1 algorithm was developed specifically for charge center location in the presence of charge sharing [7][8]. The algorithm was developed for square pixels; however other pixel layouts are possible. The C8P1 name is shorthand for "compare 8 if 1 *virtual pixel* is above threshold" and this name is a direct reference to the operation of the algorithm itself. Decisions are based on eight comparisons between the charge deposited in one pixel and the charge deposited in each of its eight neighbors. At the same time a virtual pixel must be above a threshold. This virtual pixel creates the aforementioned composite signal that triggers the comparisons.

1) Neighborhood

In the presence of charge sharing the determination of the *center pixel* is not performed by an individual pixel alone, but rather is accomplished by information exchange among pixels in an activated group. The concept of the activation of a group of pixels that participate in the evaluation of a photon impact event is referred to as *neighborhood*. The size and shape of the *neighborhood* is not known *a priori* and the size and shape of the *neighborhood* may not necessarily be the same from one event to the next. In short, a *neighborhood* of some form and size is activated by a particle's passage and the pixels in the *neighborhood* exchange information and from the pixels in that activated *neighborhood* a *center pixel* (receiving a hit to store) is evaluated.

2) Virtual Pixels

The other ingredient central to the C8P1 algorithm is that of a *virtual pixel* – that entity that creates the composite signal and generates the strobe to activate the *neighborhood*. For the algorithm to work properly, the number of *virtual pixels* must be equal to the number of real pixels; each *virtual pixel* must be a unique composition of real pixels and every real pixel must have a *virtual pixel* associated with it. Real pixels can be part of several *virtual pixels*, but there is one and only one *virtual pixel* that each real pixel owns. In fact, an array of *virtual pixels* can be seen as forming a super-layer above the real pixels.

The composite signal of a *virtual pixel* is used to create the *activation* for the *neighborhood* and is used to generate ToA measurement. The ToA result is latched together with the hit to the real pixel that owns the *virtual pixel* after the *evaluation*.

Since the *virtual pixel* is a solution to problems that arise from charge sharing, it is obvious that the specific definition of a *virtual pixel* must be dependent on the extent of charge sharing. Therefore, the specific definition of a virtual pixel must depend on pixel size and detector thickness. In the present implementation of the C8P1 algorithm, a *virtual pixel* is defined as a 2×2 block of real pixels formed from the set of four real pixels that surround each real pixel's lower right-hand corner, i.e. the pixel itself, the pixel to the west (left), the pixel to the northwest (above-left) and the pixel to the north above), as shown in Fig. 3.

3) P1 - Activation

For each *virtual pixel* whose composite signal exceeds the user defined threshold, the real pixel corresponding to the *virtual pixel* and the eight pixels surrounding become *active*. The *virtual pixel* notifies its associated real pixel and that pixel's neighbors through the *request_to_evaluate* signal. A *virtual pixel* may *activate* a *neighborhood* because its composite signal is above threshold. Conversely a pixel may be *activated* because the composite signal of a neighbor's *virtual pixel* is above threshold and that neighbor has issued its *request_to_evaluate*. In fact, a pixel may be activated by more than one *request_to_evaluate*. Regardless of the source of the *request_to_evaluate*, that pixel is involved in activity in its *neighborhood*.

4) C8 - Evaluation

Following the *activation* of a pixel in a *neighborhood*, its signal must be compared to all eight of that of that pixel's

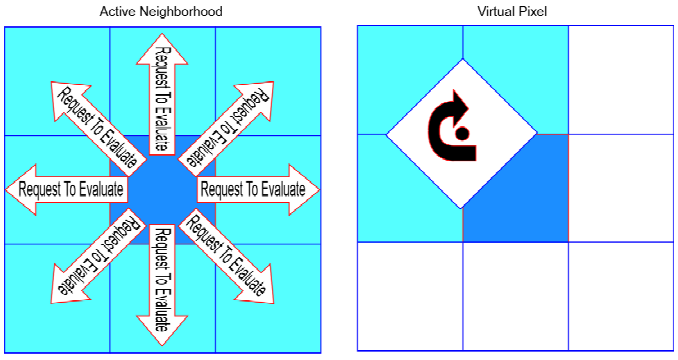

Fig. 3. Concept of *activation* of *neighborhood* and *virtual pixel*.

nearest neighbors – i.e. the pixel immediately to the north (above – N) the north-east (above-right – NE), the east (right – E), the south-east (below-right – SE), the south (below – S), the south-west (below-left –SW), the west (left – W), and the north-west (above-left – NW). Each comparison is performed independently and simultaneously. A pixel is chosen as a *center pixel* if all eight of its comparisons indicate that it has the largest charge. If any comparison shows that a neighbor's signal is larger, then another pixel is chosen as *center pixel*. The *evaluation* of comparisons is combinatorial and performed independently in every pixel. Until the *evaluation* concludes the comparison can change freely with no consequences. Upon conclusion, the state of the *evaluation* of the comparisons is latched and if the *evaluation* is true, then that pixel is a *center pixel*. Choosing the time to actually conclude the *evaluation* of an event is decided by at least one *virtual pixel*. It is worth remembering that the activating of a *neighborhood* may be triggered by one or more *virtual pixels*. *Virtual pixels* do not necessarily withdraw their *request_to_evaluate* signals simultaneously and the last active *request_to_evaluate* becomes responsible for ending the *evaluation* and latching the results of the comparisons. Since *neighborhood activation* can occur from any neighboring *virtual pixel*, *neighborhoods* can be extensive, especially in the presence of multiple, coincident photons. However, comparisons are still only performed between an active pixel and its eight nearest neighbors. Therefore, it is possible that more than one active pixel in a *neighborhood* can be evaluated to be a *center pixel.*

Finally, it should be underscored that while *activation* comes from a *virtual pixel* and its composite signal*, evaluation* is performed on fractional signals deposited on real pixels. Moreover, these fractional signals are filtered in such a way that their peak amplitudes are used for *evaluation.*

C. Realization of C8P1 Algorithm in a Circuit Network

The details of the implementation are presented for a single pixel. However the description differentiates between the real pixel and the *virtual pixel*. Analog and digital sections of the circuits are also addressed separately for clarity. Special attention is given to the exchange of signals between pixels. The inter-pixel communication occurs at both levels, i.e. the analog and digital level, and every real pixel receives some signals from its neighbors and broadcasts others to its neighbors.

1) The Real and Virtual Pixel Analog Sections

The analog pixel section is shown in Fig. 4. The first stage of the signal processing is a charge-sensitive preamplifier (CSA). It performs active charge integration and converts the charge collected by the pixel electrode into a voltage step. The signal from the preamplifier is split into five branches. Four of the branches go to *virtual pixels* through capacitive couplings. The last branch goes to a shaping filter amplifier that is marked *slow* in Fig. 4. The *virtual pixels* are represented by diamonds located in the corners of a real pixel. The *virtual pixel* that is owned by the real pixel is located in the north-west corner of the real pixel. Each *virtual pixel* constructs a composite signal from the outputs of four CSAs. This composite signal is passed to a shaping-filter amplifier that is marked *fast* in Fig. 4. For the sake of clarity and for the duration of the paper, the fast shaper will be referred to as the *trigger shaper* and the slower shaper will be called the *compare shaper*.

The trigger shaper is responsible for the generation of the *request_to_evaluate* signal upon exceeding a programmable threshold at the discriminator that follows the trigger shaper. The *request_to_evaluate* signal from the *virtual pixel* is sent from the real pixel to its eight neighbors. It can be seen in Fig. 4 that a *virtual pixel* is formed from capacitive addition of the outputs of the preamplifiers from four real pixels that share a corner. Thus, each real pixel contributes to four *virtual pixels*, and each *virtual pixel* is responsible for issuing the *request_to_evaluate* signal that contributes to each pixel's *neighborhood activation*. The *neighborhood_active* signal is the internally generated control strobe signal introduced earlier. This signal is necessary to process a photon impact, i.e. to initiate and to conclude the comparisons of fractional signals among adjacent pixels. All pixels in an *active neighborhood* will have their *neighborhood_active* signal set.

The processing of signals in each pixel requires eight comparisons. Four of the comparators are located internally. Looking at Figure 4, the output of the compare shaper is driven to four local (internal) comparators, which also receive the analogous compare shaper signals from the pixels to the northwest, north, west and southwest directions. Similarly, the same compare shaper signal is also driven to four external comparators located in the pixels to the north-east, east, south-east and south, whose outputs are returned. In this manner, all eight comparisons of the real pixel signal with the corresponding signals from the eight neighbors are achieved in the *evaluation* process. Thus, eight comparisons are performed for every pixel, while each pixel only houses four comparators. This optimizes the use of circuit resources, and is particularly important for keeping the footprint of the pixel small and avoiding redundancy. To further the optimization, the comparators, operating on signals from compare shapers, are implemented with auto-zeroing of their offsets [9]. Implementation of auto-zeroing eliminates the valuable real-estate otherwise occupied by the digital-to-analog convertors that are classically used for trimming the offsets of comparators.

As depicted in Fig. 4, the comparison process is controlled by the *neighborhood_active* signal in every pixel. Active comparisons occur when *neighborhood active* is set. On the falling edge of *neighborhood_active*, the results of the comparisons are latched and the comparators are auto-zeroed. From the above discussion, it comes evident that time responses of the trigger and compare shapers are related. The trigger shaper is considerably faster than the compare shaper ideally the relationship between them should be such that the response of the trigger shaper goes below threshold when the response of compare shaper reaches its peak amplitude.

2) The Digital Pixel Section

The digital section of the pixel determines to which *activated* pixel a processed hit should be attributed. This decision is based on the eight required analog comparisons, which are synchronized by *neighborhood_active*. A simplified diagram of the digital pixel section is shown in Fig. 5. The main components of the digital pixel circuitry are: the neighbor logic, the C8 logic and the frame logic.

The C8 logic is effectively a large AND gate that is enabled when *neighborhood_active* is set. The pixel is considered a possible *center pixel* when all eight comparisons are in its favor. Ultimately, possible *center pixels* become genuine, latched *center pixels* with the falling edge of the *neighborhood_active* signal.

Four of the total eight comparisons are performed by comparators located inside a given pixel. These give the

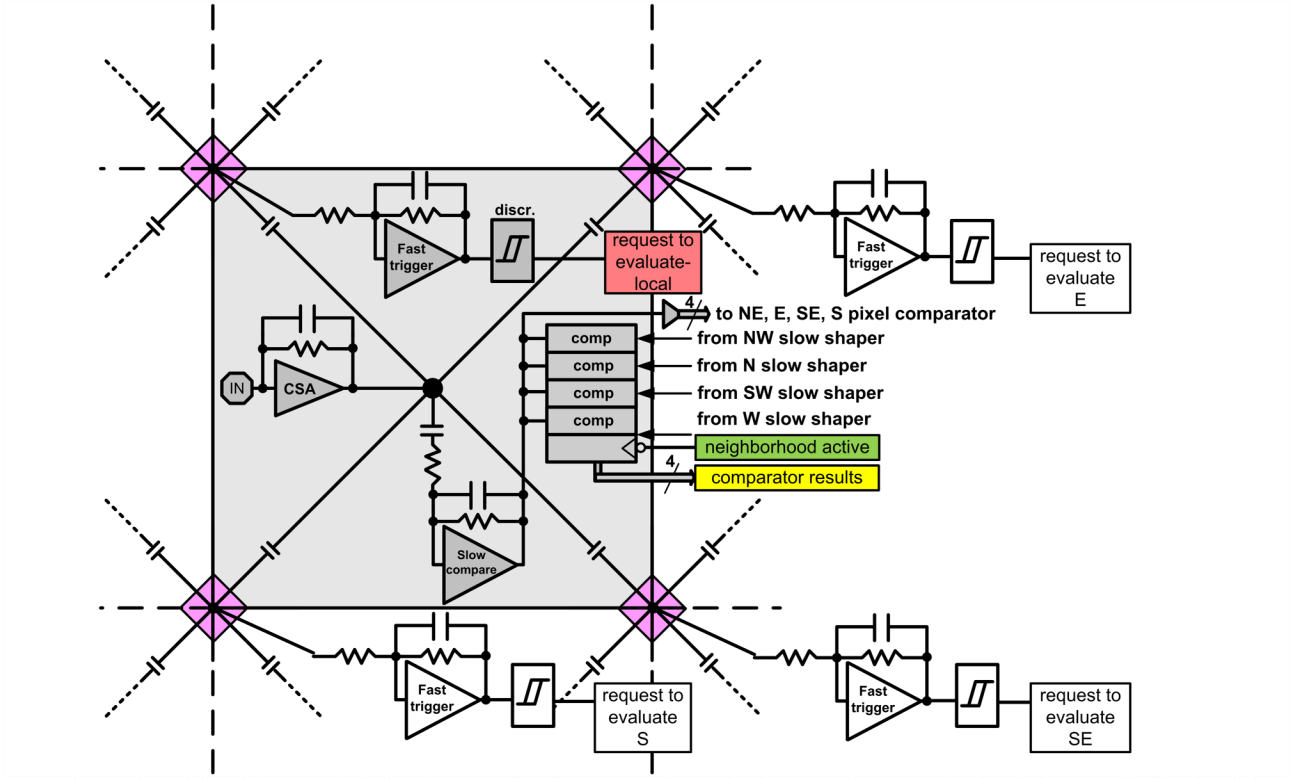

Fig. 4. Analog portion of the miniVIPIC pixel, showing *virtual pixels* (magenta diamonds) that create the composite signals from 2×2 blocks of real pixels.

The neighbor logic decides how a self-defining and event dependent neighborhood actually grows. It takes the local *request_to_evaluate* signal and processes it with some of the neighbor *request_to_evaluate* signals. Selecting which neighbor *request_to_evaluate* signals ought to be used is crucial to the results. As is shown in Fig. 5, the *neighborhood active* is set when at least one of the gated *request_to_evaluate* signals comes to the pixel activated.

Performing a simple logical OR operation on *request_to_evaluate* signals from all eight directions is one of the possibilities. This is implemented as the Full Neighbor Mode. Unfortunately the definition of the *virtual pixel* and its position relative to the real pixel that owns it introduces an asymmetry to the system. It was seen that the Full Neighbor Mode may exhibit some inefficiencies in the presence of this asymmetry. Therefore, in order to alleviate the effects related to the asymmetry, the neighbor logic implements directional restrictions on how *request_to_evaluate* signals are used by a pixel. This led to two additional modes in the neighbor logic. These two modes, called Minimally Modified Neighborhood Mode and Symmetric Neighborhood Mode, are best illustrated by the example and comparison with the Full Neighborhood Mode that is provided in Section III.

results directly to the pixel logic, while the results of four remaining comparisons come from neighbors. Obviously, the results of comparisons performed locally in a pixel are broadcast to the appropriate neighbors. As it is shown in Fig. 5 these forwarded results are inverted and gated with the *neighborhood_active* signal. All comparators in all pixels are wired so that their output is True if the charge collected by the local pixel is larger than the charge collected by the neighbor.

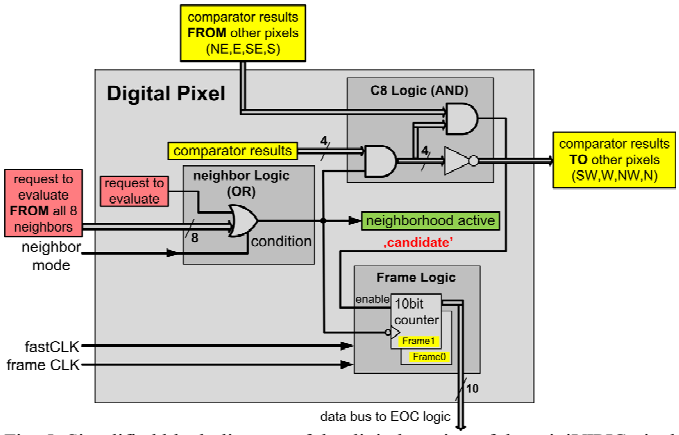

Fig. 5. Simplified block diagram of the digital portion of the miniVIPIC pixel with the neighborhood logic, C8 logic and frame logic.

Therefore, a True output from a neighbor's comparator means that the charge collected by the neighbor is larger – i.e. neighbor comparator outputs are implicitly inverted and must be inverted again for use in the C8 logic. Additionally, when a pixel is not part of an active *neighborhood*, it cannot become a center pixel and it must passively indicate a state equivalent to losing any comparison to any of its neighbors. This is achieved by gating of the comparators outputs with the *neighborhood active* signal before broadcasting them.

The frame logic is illustrated in Fig. 5 only as a dual 10-bit counter, where hits are counted. In reality, this frame logic is much more complicated. It is built for selectable operation of the chip, both in timing (precise ToA measurements of photon impacts) and counting mode (counting of photon impact events in a given time frame). The frame logic is also responsible for correct dead-time-less operation of the detector in synchronization with the external frame control signal. This includes toggling between one counter, used for acquisition, and a second counter, from which the data is transmitted via a data bus.

D. Inter-pixel Connections

The communication between neighboring pixels in the C8P1 algorithm requires a dense network of analog and digital connections between the pixels. The degree of inter-pixel connectivity in the C8P1 algorithm is significant, but considering the density of connection, such a need can be satisfied by modern fabrication processes. The exact nature of the problem lies in the fact that both the analogue and digital signals must be transmitted over relatively long distances. Clearly, adequate shielding against interference is essential as is a minimization of stray capacitance. The goal is to maintain the power consumption at an acceptable level, keeping in mind that a detector with the C8P1 algorithm should not differ in benchmarking parameters from conventional pixel detectors. In practice, dedicated, appropriately shielded routing channels of about 10 µm width have to be reserved horizontally and vertically across every pixel to host these interconnects. The diagrams of inter-pixel connections are shown in Fig. 6 and Fig. 7. In both figures, only signals serving one pixel are depicted in order to preserve clarity. In the actual pixel array, all pixels are accordingly connected with each other. For the sake of intelligibility, all incoming inter-pixel connections are shown in Fig. 6, while all the outgoing inter-pixel connections are shown in Fig. 7. It can be seen that interconnects are symmetric, and all signals, coming to a pixel, are analogous to those that are broadcast by a pixel.

1) Incoming Connections

A *virtual pixel* (a diamond in Fig. 6) receives signals from charge-sensitive preamplifiers from pixels in the N, NW and W directions and adds them with a signal from the local charge-sensitive preamplifier.

In order to perform the four comparisons, outputs of the compare shapers from pixels in the N, NW, W, SW directions are sent to a pixel. At the same time, a pixel receives the results of four comparisons, remaining for the completion of the C8P1 algorithm, from pixels in the S, SE, E and NE

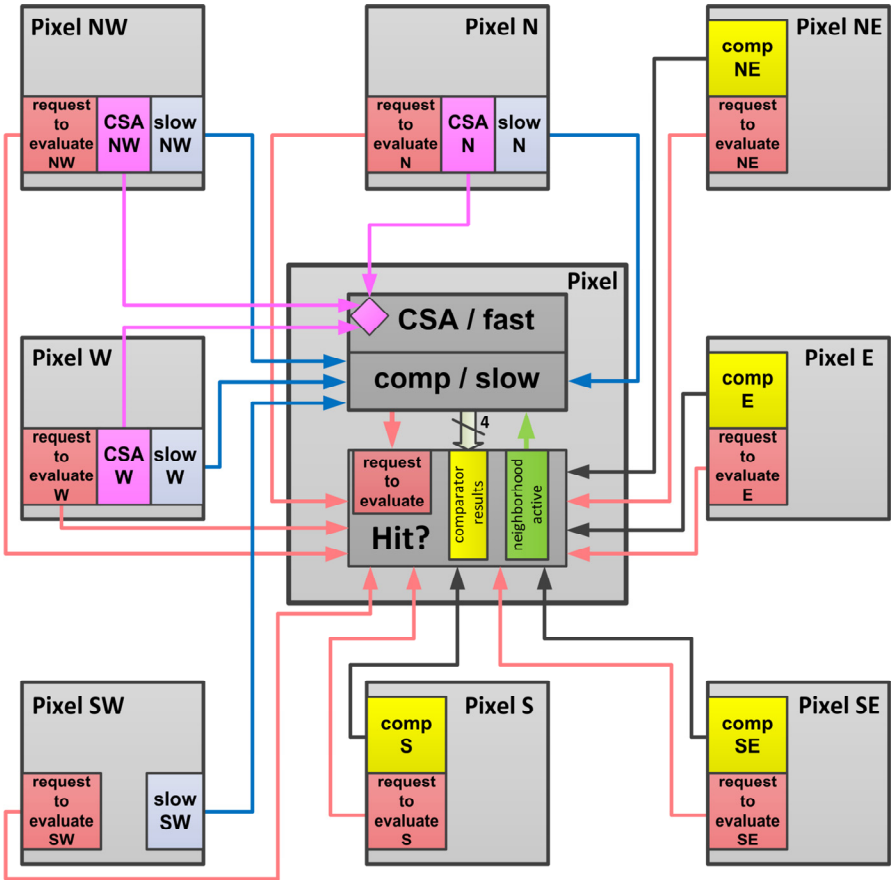

Fig. 6. Incoming inter-pixel connections to a pixel in the C8P1 algorithm.

directions. All the comparison results are passed to the digital section and, by then, a pixel has complete information to determine if it should record a hit.

The local *request_to_evaluate* signal is processed in the *neighborhood* logic together with eight *request_to_evaluate* signals that are received from the neighbors in all directions.

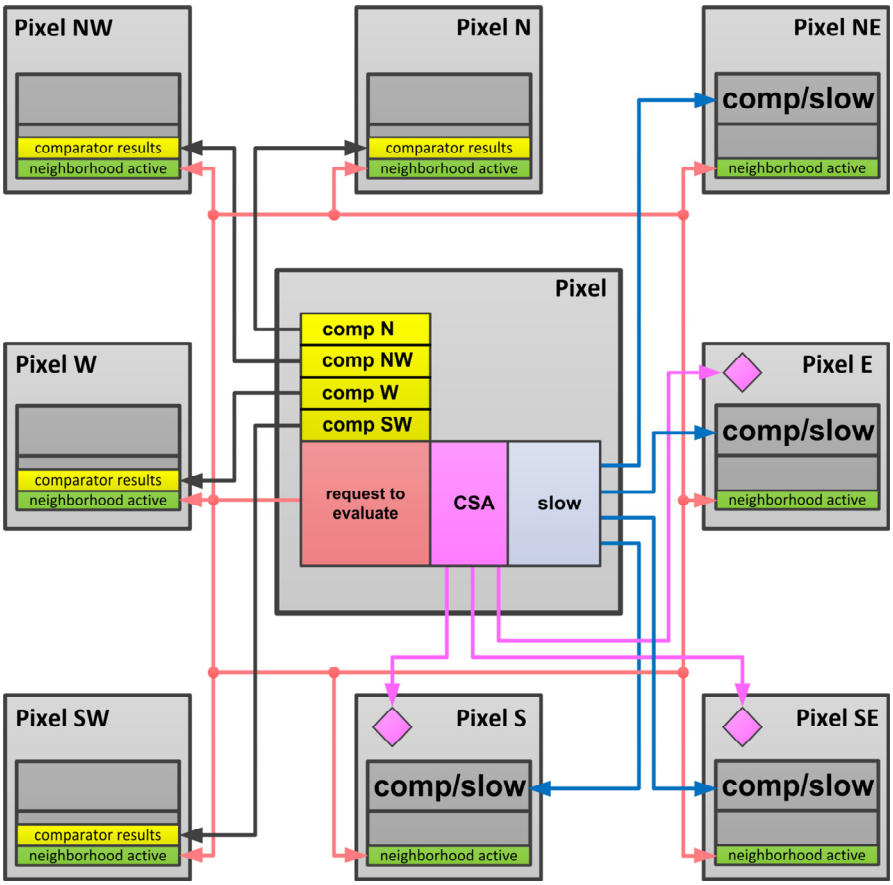

Fig. 7. Outgoing inter-pixel connections from a pixel in the C8P1 algorithm.

2) *Outgoing Connections*

The output signal of the CSA is sent to *virtual pixels* associated with the neighboring pixels located in the E, S and SE directions.

The compare shaper output is sent to the neighboring pixels in the NE, E, SE and S directions. The digital results of four internal comparisons are passed back to the relevant neighboring pixels located in the N, NW, W and SW directions.

Last but not least, the locally generated *request_to_evaluate* signal is passed to all eight neighbors to be used in their neighborhood logic blocks.

III. Processing of Events Following the C8P1 Algorithm

In this section, the C8P1 algorithm and its implementation are illustrated using two examples. In both cases, a hypothetical 5×5 array of C8P1 pixels is used and a charge of 2200 e^- is liberated in the detector, which is equivalent to an 8 keV X-ray photon absorbed in silicon. The Full Neighbor Mode is used for simplicity. In the first example, Example A, the entire charge is collected by one real pixel. In the second example, Example B, the charge is spread across four adjacent pixels. These examples are illustrated in three figures. Fig. 8 shows a time progression of signals appropriate to either example. Fig. 9 shows the algorithm states of Example A and Fig. 10 shows the corresponding states for Example B.

First, consider Fig. 8, which shows, as a function of time, the output of a trigger shaper along with the user definable threshold setting, a *neighborhood_active* signal, two different compare shaper outputs, and finally a potential *center pixel* signal labeled “candidate allocation”. At the very top of Fig. 8

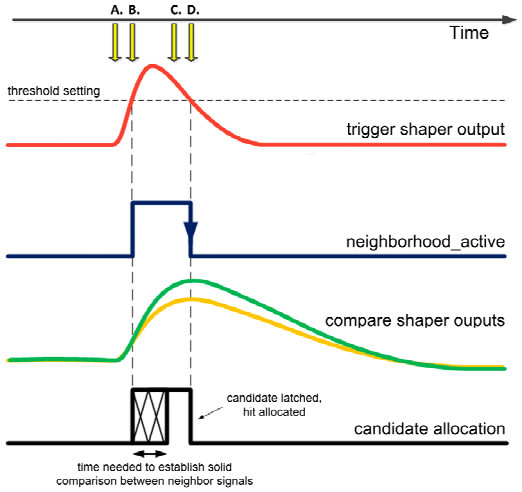

Fig. 8. Signals used in the course of the execution of the C8P1 algorithm.

is a horizontal time axis with four labels, A, B, C and D. Time A corresponds to the hit arrival time. This also corresponds roughly to Graph 1 in either Fig. 9 or Fig. 10 where it is labeled "Hit Arrival". Time B corresponds to the point at which the trigger shaper goes above the user defined threshold. Again, this corresponds to the states depicted in Graphs 2, 3 and 4, where they are labeled "Virtual Pixel" (Graph 2), "Requests to Evaluate" (Graph 3) and "Neighborhood Active" (Graph 4). The pixel broadcasting *request_to_evaluate* signals to their neighbors are highlighted by the label "rE" in Graph 3. The pixels that are included in *neighborhood_active* are highlighted by the label "nA" in Graph 4. Time C, in fact, labels the *evaluation* phase that extends from B to D. It corresponds to Graph 5. Arrows in Graph 5 of either Fig. 9 or Fig. 10 point in the directions of the comparison winners. Finally Time D corresponds to the point at which the trigger shaper goes below the user defined threshold and matches to Graph 6, where it is labeled "Hit Decision".

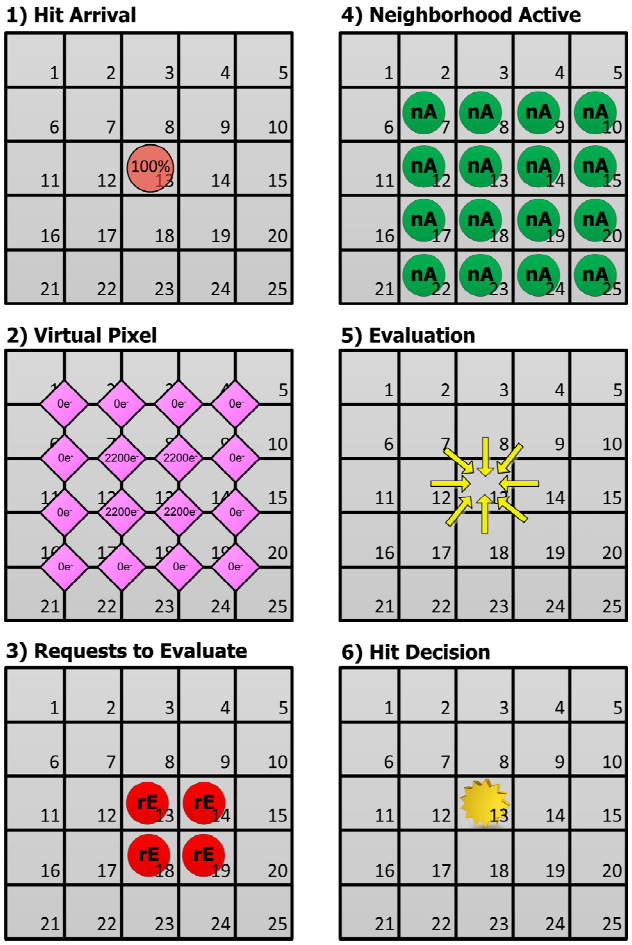

Fig. 9. Illustration of the processing of a photon impact according to the C8P1 algorithm for collection of the generated charge entirely by one pixel (13).

A. *Example A – Isolated Hit on One Pixel*

At Time A, an incident photon deposits all its charge on Pixel 13. This is shown in Fig. 9 Graph 1. This means that the *virtual pixel* associated with Pixel 13 as well as the *virtual pixels* associated with Pixel 14, 18, and 19 all have signals equivalent to 2200 e^- presented to their trigger shapers. This is shown in Fig. 9 Graph 2.

At Time B, the output of the trigger shapers in *virtual pixels* owned by Pixels 13, 14, 18, and 19 are above the user defined threshold. This causes the *virtual pixels* to issue *request_to_evaluate* signals to its associated real pixel as well as its eight neighbors. This is shown in Fig. 9 Graph 3.

This means that the *request_to_evaluate* from Pixel 13 is broadcast to Pixels 7, 8, 9, 12, 13, 14, 17, 18, and 19. The *request_to_evaluate* from Pixel 14 is broadcast to Pixels 8, 9, 10, 13, 14, 15, 18, 19, and 20. The *request_to_evaluate* from Pixel 18 is broadcast to Pixels 12, 13, 14, 17, 18, 19, 22, 23, and 24. Finally, the *request_to_evaluate* from Pixel 19 is broadcast to Pixels 13, 14, 15, 18, 19, 20, 23, 24 and 25. Looking differently at the same information, Pixel 13 has the

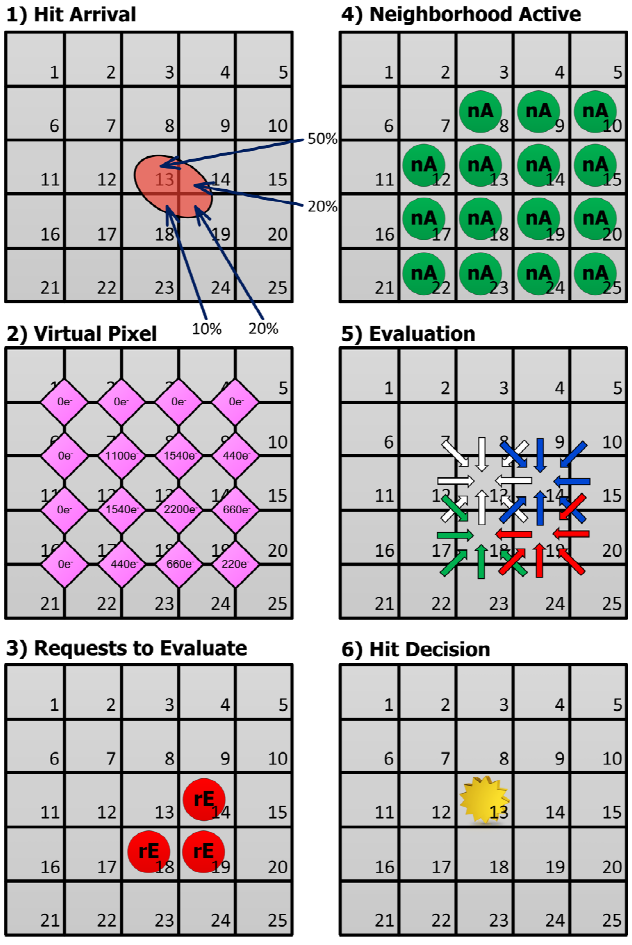

Fig. 10. Illustration of the processing of a photon impact according to the C8P1 algorithm for shared collection of the generated charge between 4 pixels (13 – 50%, 14 – 20%, 18 – 10% and 19 – 20%).

entire charge collected locally, and its own *virtual pixel* is above threshold, and it receives *request_to_evaluate* from its own virtual pixel as well as from Pixel 14, 18 and 19. Pixel 19 has no charge collected locally, but its own *virtual pixel* is above threshold and it receives *request_to_evaluate* from its own *virtual pixel* as well as from Pixel 13, 14, and 18. Finally, Pixel 7 has no charge collected locally, and its *virtual pixel* is not above threshold, and it receives only one *request_to_evaluate* from Pixel 13. Regardless of the different manner in which these pixels receive their *request_to_evaluate*, the neighbor logic in Pixels 13, 19, and 7

all evaluate equivalently and activate their local *neighborhood_active* signal. In fact, Pixels 7, 8, 9, 10, 12, 13, 14, 15, 17, 18, 19, 20, 22, 23, 24 and 25 all set their local *neighborhood_active* signal and join the active neighborhood for evaluation. This is shown in Fig. 9 Graph 4.

During Time C, the active *neighborhood* is in *evaluation.* All members of the active neighborhood, regardless of how they became members of the active *neighborhood* evaluate the results of eight comparisons. Pixel 7 compares itself with Pixels 1, 2, 3, 6, 8, 11, 12, and 13. The comparison between Pixel 13 and Pixel 7 clearly indicates that Pixel 7 has less charge deposited on it, so Pixel 7 evaluates as not being a *center pixel.*

It should be remembered that *evaluation* is performed on signals generated from the actual charge deposited on the pixel and processed through the compare shaper. The fact that Pixel 19's *virtual pixel* is above threshold has no bearing on *evaluation.* Regardless, the comparison between Pixel 13 and Pixel 19 similarly indicates that Pixel 19 has less charge deposited on it, so Pixel 19 evaluates as not being a *center pixel.*

At Time D, all of the trigger shapers drop back below the user defined threshold. As each trigger shaper in *virtual pixel*s owned by Pixels 13, 14, 18, and 19 drops below threshold, they withdraw their *request_to_evaluate*. This causes each pixel to deactivate its *neighborhood_active* signal. The falling edge of this signal latches the state of the local *evaluation.* Pixel 13 is the only pixel that has all eight of its comparisons in its favor. This is shown in Fig. 9 Graph 5. Therefore, the Hit Decision at Time D indicates that Pixel 13 is the *center pixel*. This is shown in Fig. 9 Graph 6.

B. Example B – Shared Charge

At Time A, an incident photon generates 50% of 2200 e^- charge on Pixel 13, 20% of charge on Pixel 14, 10% of charge on Pixel 18, and 20% of charge on Pixel 19, as shown in Fig. 10 Graph 1. This means there is a distribution of charge from as low as 220 e^- to 2200 e^- across *virtual pixels* associated with Pixels 13, 14, 15, 18, 19, 20, 23, 24 and 25. This is shown in Fig. 10 Graph 2.

For the purpose of this example, it is assumed that the user defined threshold is equivalent to 1500 e^-. It is evident from the example that the majority of the charge is collected in Pixel 13. However, the *virtual pixel* owned by Pixel 13 does not fire, because of the manner in which *virtual pixels* construct composite signals. Therefore, at Time B, only three of the pixels own *virtual pixels* that have trigger shaper outputs above the threshold, viz. Pixels 14, 18, and 19. This is shown in Fig. 10 Graph 3. These *request_to_evaluate* result in an active neighborhood as shown in Fig 10 Graph 4.

During Time C, only one pixel, viz. Pixel 13, has all eight of its comparisons in its favor. Therefore, at Time D, Pixel 13 is assigned as the *center pixel*, when the last *request_to_evaluate* drops and *neighborhood_active* falls.

C. Neighborhood Modes

In certain classes of events, such as Example A, Pixel 25 does not have any charge deposited upon itself nor do any of its immediate neighbors. Yet, it is part of an active *neighborhood* and because of this lack of charge, it can erroneously evaluate as a *center pixel*.

Virtual pixels owned by 13, 14, 18 and 19 activate symmetrical neighborhoods around themselves for evaluations. However, the charge is actually collected entirely in real Pixel 13. With respect to real Pixel 13, this activated neighborhood is actually asymmetrical, leading to Pixel 25 attempting to evaluate itself. It is for this reason that the modes were added to the neighbor logic. The three modes available are as follows:

1) Full Neighborhood Mode, as depicted in Fig. 9, which allows a pixel to use all nine *request_to_evaluate*.

2) Minimally Modified Neighborhood Mode, which does not allow a pixel to use the *request_to_evaluate* signal from the NW direction. This would eliminate Pixel 25 in Example A from the active *neighborhood*.

3) Symmetric Neighborhood Mode which does not allow a pixel to use the *request_to_evaluate* signals from NW, NE, N and W directions. This would eliminate Pixel 10, 15, 20, 22, 23, 24, and 25 in Example A from the active *neighborhood* and would result in a perfectly symmetrical neighborhood around Pixel 13.

Again, Examples A and B presented above use the Full Neighborhood Mode. For both presented examples, changing the modes would result only in a change of Graph 4 in Fig. 9 and Fig. 10, respectively. The analysis is straightforward. Empirical verification of the best *neighborhood* method is awaited in tests of the miniVIPIC prototype.

IV. The miniVIPIC chip

The miniVIPIC chip is a test vehicle for the C8P1 algorithm using the GlobalFoundries CMOS 130 nm process. The process was chosen with an eye towards future 3D-IC implementation [9][10]. The miniVIPIC chip is a part of the broader VIPIC (Vertically Integrated Photon Imaging Chip) project, whose target is to deliver a new camera for X-ray Photon Correlation Spectroscopy experiments [10][11]. A variant of the C8P1 algorithm, in which the property governing the latching (duration of the *request_to_evaluate*) of the comparisons is defined with a user adjustable delay, was implemented in the TSMC CMOS 40 nm process [12]. The key component of the two designs is a highly dense interconnection between the pixels. These two projects going in parallel target two optimization directions bearing on future intelligent pixel detectors. On the one hand, the density of interconnects can be managed with less effort in a future 3D-IC implementation, and on the other hand the 40 nm experience showed that pixel dimensions can be significantly decreased despite hosting fairly complex circuitry.

The full layout of the miniVIPIC prototype chip, measuring 4.8×5.4 mm^2, is shown Fig. 11. It contains an array of 32×32 pixels and permits complete evaluation of the C8P1 algorithm, including all modes. The neighborhood modes can be selected using two signals available at the pads.

The readout of data is organized column-wise by grouping

every four columns into one readout node, resulting in eight output channels. The readout is not zero suppressed, which does not represent a limitation due to the small size of the matrix but allows unconstrained evaluation of the functioning of the algorithm. The information stored in every pixel is sent to the end-of-column logic, where the data is serialized and sent off chip by LVDS ports. Every pixel presents the contents of its 10-bit long counter for readout. Each row of the array is addressed after another in the readout process and pixels from each group are available for readout simultaneously. The chip can either be operated in the timing mode or in the counting mode, which changes the way in which in-pixel counters are used. In the timing mode, the counter stores a number of counts of a high speed clock ticks that pass between the detected photon arrival and the end of the observation frame. In the counting mode, the counter stores just the number of photons detected in the observation frame. The observation frame in the miniVIPIC chip is defined as the time needed to have all pixels in the matrix read out. In both modes, dead-time-less operation is provided by two interleaved counters in each pixel, one accessed for readout and one used for acquisition of new data. The serializer and the LVDS links can operate at the frequency of up to 200 MHz. This results in a minimal achievable frame readout time of about 7 µs. Taking into account the latter, the miniVIPIC device should yield measurements of photon arrivals with a 7 ns timing resolution. It is also worth mentioning at this point that the miniVIPIC chip can be operated reading only parts of the pixel matrix, thus frame durations shorter than 7 µs are also achievable.

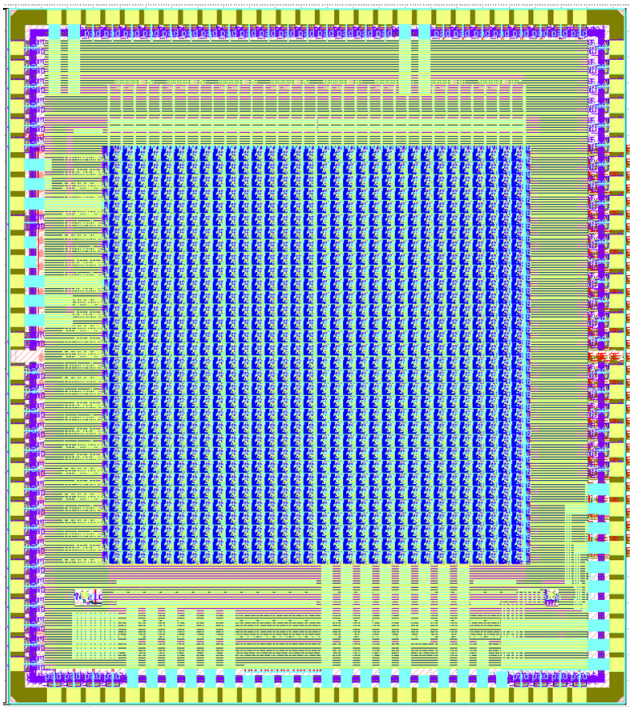

Fig. 11. Layout of the 4.8×5.4 mm^2 miniVIPIC chip with an array of 32×32 pixels.

The main specifications and features of the miniVIPIC chip are summarized in Table 1. The pixel layout is shown in Fig. 12. The main components of the analog section, occupying the left side of the pixel, and of the digital section, occupying the right side of the pixel are highlighted. The pixel size is 100×100 µm^2. A pad for the bump-bonding connection to a silicon sensor has a conservative diameter of 60 µm. The digital and analog sections are separated by having the digital section placed in a deep-nwell in addition to having both of them in different parts of the pixel. Care was taken to have all digital control signals run vertically over the digital section only. Fitting the digital circuitry into the available space was possible thanks to a highly compact, custom digital standard cell library that was developed in-house. Approximately 10 µm-wide routing channels were reserved horizontally and vertically as well around the pixel border to allow the dense interconnections required to implement the C8P1 algorithm, as shown in Fig. 12.

TABLE I. SPECIFICATIONS OF THE MINIVIPIC CHIP

Specification	
Fabrication process	GlobalFoundries 130nm, 1.5V, CMOS LP, 1P8M
Submission	September 2014
Chip size	4.8 x 5.4 mm^2
Number of pixels	32 x 32 in 8 groups
Pixel size	100×100 µm^2
Peaking time and gain of fast shaper	40 ns and 37 µV/e$^-$
Peaking time and gain of slow shaper	85 ns and 25 µV/e$^-$
Analog power consumption	45 µW / pixel
ENC noise (fast shaper)	110 e$^-$ @ C_{DET} = 70 fF
Target photon energy	8 keV X-rays (2200 e$^-$)
Injection capacitance (selectable)	0, 1.25, 2.5, 3.75, 5 fF
Trimming DAC (for trigger discriminator)	7 bits
Deadtime less operation	yes
Sparsified readout	no
Energy resolution	no
Operation modes	counting or timing
In-Pixel counter	2 x 10 bit (interleaved)
Frame readout time	7 µs at 200 MHz LVDS clock

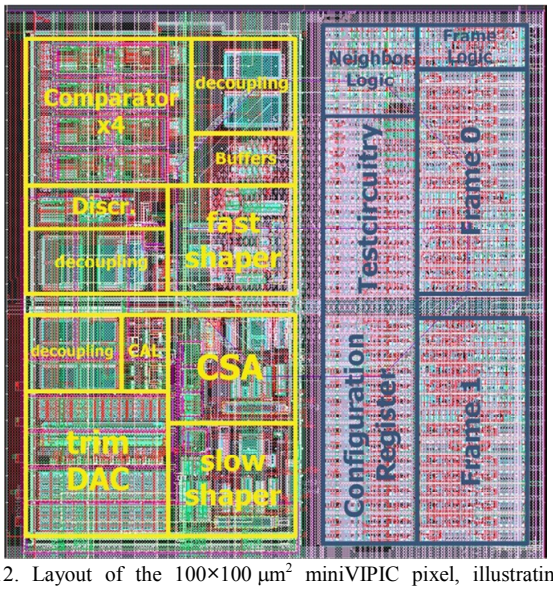

Fig. 12. Layout of the 100×100 µm^2 miniVIPIC pixel, illustrating the components of the analog section (left) and of the digital section (right).

V. Conclusions

The C8P1 algorithm for the allocation of a hit to a single pixel in the presence of charge sharing in a highly segmented pixel detector was presented. The key elements of the algorithm were described as was the method of implementation. A thorough analysis of the performance of the algorithm was presented through two examples of photon impacts. The miniVIPIC prototype was designed in a 130 nm process as a proof of feasibility of the hardware implementation of the C8P1 algorithm.

The C8P1 algorithm was developed to advance pixel detector technology for experiments with X-ray beams at a synchrotron facility. It is a part of the broader VIPIC project, whose target is to deliver a new camera for X-ray Photon Correlation Spectroscopy experiments.

Acknowledgments

We would like to acknowledge Dr. Rafael Ballabriga and Dr. Michael Campbell from CERN, Geneva Switzerland for discussions related to the MEDIPIX devices that allowed better understanding of the issues related to the charge sharing and led us to the conclusions in the form of the algorithm presented in this paper.

References

[1] C.Ponchut, “Correction of charge sharing in photon-counting pixel detector data“, Nucl. Instr. and Meth. Phys. Res. A 591, pp. 311-313, 2008

[2] D.M.Binkley, B.S.Puckett, BK.Swann, J.M.Rochelle, M.S.Musrock, M.E.Casey, “A 10-mc/s, 0.5-µm CMOS constant-fraction discriminator having built-in pulse tail cancellation“, IEEE Trans. Nucl. Sci., vol. 49, no. 3, pp. 1130-1140, 2002

[3] ORTEC, “Principles and Applications of Timing Spectroscopy”, application note AN42 [online] http://www.ortec-online.com/download/Application-Note-AN42-Principles-Applications-Timing-Spectroscopy.pdf.

[4] T.Akesson, E.Arik, K.Assamagan, K.Baker, E.Barberio, D.Barberis, et al. “Particle identification using the time-over-threshold method in the ATLAS Transition Radiation Tracker”, Nucl. Instr. and Meth. Phys. Res. A 474, pp. 172–187, 2001

[5] Z.Y.Chang, W.M.C.Sansen, “Low –Noise Wide-Band Amplifiers in Bipolar and CMOS Technologies”, Kluwer Academic Publishers, 1991

[6] R.Ballabriga, “The Design and Implementation in 0.13µm CMOS of an Algorithm Permitting Spectroscopic Imaging with High Spatial Resolution for Hybrid Pixel Detectors”, PhD thesis, Universitat Ramon Llull, 2009

[7] G.Deptuch, “Speckle Detectors”, BES Neutron and Photon Detector Workshop, August 1-3 2012, Gaithersburg, MD USA [online] https://portal.slac.stanford.edu/sites/conf_public/nxd2012/presentations/DOEBES_deptuch_white.pdf

[8] A.Baumbaugh, G.Carini, G.Deptuch, P.Grybos, J.Hoff, P.Maj, et al., “Analysis of full charge reconstruction algorithms for X-ray pixelated detectors”, 2011 IEEE NSS & MIC Conference Record, Valencia, Spain pp. 660-667, 2011

[9] G.Deptuch, D.Christian, J.Hoff, R.Lipton, A.Shenai, M.Trimpl, et al., “A Vertically Integrated Pixel Readout Device for the Vertex Detector at the International Linear Collider“, IEEE Trans. Nucl. Sci, vol. 57, no. 2, pp. 880-890, 2010

[10] G.Deptuch, M.Demarteau, J.Hoff, R.Lipton, A.Shenai, M.Trimpl, et al., “Vertically Integrated Circuits at Fermilab“, IEEE Trans. Nucl. Sci., vol. 57, no. 4, pp. 2178–2186, 2010

[11] G.Deptuch, M.Trimpl, R.Yarema, P.Siddons, G.Carini, P.Grybos, et al., “VIPIC IC - Design and test aspects of the 3D pixel chip“, 2010 IEEE IEEE NSS & MIC Conference Record, Orlando, FL, USA, 2010, pp. 1540-1543, 2010

[12] P.Maj, P.Gryboś, R.Szczygieł, P.Kmon, A.Drozd, G.Deptuch, “A pixel readout chip in 40 nm CMOS process for high count rate imaging systems with minimization of charge sharing effects”, 2013 IEEE NSS & MIC Conference Record, Seoul, South Korea, 2013