

A Low-Power, High-Speed Readout for Pixel Detectors Based on an Arbitration Tree

Farah Fahim¹, Siddhartha Joshi¹, *Student Member, IEEE*, Seda Ogrenci-Memik¹, *Senior Member, IEEE*, and Hooman Mohseni, *Senior Member, IEEE*

Abstract—In this article, a low-power, high-speed arbitration tree for pixel detector readout is presented. The synchronized, binary tree priority encoder establishes a position-dependent priority list at the start of every time frame. Pixels that indicate the presence of data for readout are sequentially granted access to a shared bus for data transfer to the periphery, without the use of an additional global strobe signal. It can be used for either full frame imaging or zero-suppressed readout, in which case it can simultaneously generate the pixel address. To increase the readout frame rate, the pixel array is subdivided into two halves, which allow interleaved latching of data at the output serializer. The design was implemented in a 65-nm LP-CMOS process for the readout of a 64×64 pixel array. Measurement results demonstrate a deadtimeless, full frame imaging rate of ~ 50 kfps, achieved with a dedicated output for every (32×32) 1024 pixels and for a pixel data packet of 11 bits, with no bit errors detected over 1000 frames. The measured energy per bit is 0.94 pJ.

Index Terms—Arbitration tree, data sparsification, pixel detector readout, priority encoder (PE), zero suppression.

I. INTRODUCTION

HYBRID pixel radiation detectors typically contain a pixelated sensor layer bonded to a pixelated readout integrated circuit (ROIC). They are used for measuring the properties of incoming radiation in a wide range of applications, including particle tracking in high energy physics, medical imaging, and focal plane arrays for astronomy. As pixel detectors have evolved over the last few decades, the readout of the pixels has itself evolved from simple analog readouts to fast digital readouts enabling higher data frame rates. Typical in-pixel digital measurements include counting the number of incoming photons [1]–[3], measuring the time of arrival of the photon [4], [5], or analog-to-digital conversion of accumulated photons [6]–[8] within a given time frame or integration window. It is generally desirable for these detectors to be operated continuously without any deadtime such that when

new information is being recorded in the current time frame, simultaneously the previous information is being sent off to the data acquisition system. For deadtimeless operation, the time to read out the frame should be less than or equal to the photon detection and processing time frame. Over time, the number of pixels has increased from a few hundreds to greater than a few billions [9], and the sensitive area for the entire detector has increased from few square millimeters to few square meters [10]. As the size and area of the detector increases, so does the power consumption, while the full-frame readout rate decreases. It takes more time and energy to transfer data over longer distance due to larger interconnect capacitance from the central areas of the detector to the periphery. Hence, high-speed, low-power readout architectures are required.

Applications with low pixel occupancy such as photon correlation spectroscopy [11] or quantum cryptography [12] benefit from data sparsification techniques, such as zero-suppression which eliminates zeros by not transmitting data from pixels with no acquired photons. Data-driven zero-suppressed readout is a means of reducing the data bandwidth as well as increasing frame rates, provided that the system is able to move the information off-chip at the same rate as it is being produced.

We propose a low-power, high-speed, reconfigurable readout architecture which allows both full frame as well as zero-suppressed read out, based on an arbitration tree for transferring data from pixel to periphery within a user-defined time frame. It is capable of achieving deadtimeless, full frame imaging rates of ~ 50 kfps, with a dedicated output for every 1024 pixels and for a pixel data packet of 11 bits, with no bit errors detected over 1000 frames. The measured energy per bit is 0.94 pJ.

The rest of this article is arranged as follows. In Section II, we briefly discuss pixel detector readout techniques; and in Section III, we explain the need for our approach and the optimizing potential in other designs. In Section IV, we present our design of a synchronized binary tree priority encoder (SB-PE). The detailed pixel level logic and implementation is shown in Section V. Section VI presents the test results, followed by conclusions in Section VII.

II. PIXEL DETECTOR READOUT TECHNIQUES

Pixel detector readouts using “time frames” enable periodic snapshots of data to be recorded, which allow for deadtimeless operation. Each pixel contains two sets of registers: while one set is processing data in the current time frame, another set

Manuscript received May 15, 2019; revised September 12, 2019; accepted October 8, 2019. Date of publication December 11, 2019; date of current version January 21, 2020. This work was supported by Fermi Research Alliance, LLC through the U.S. Department of Energy, Office of Science, Office of High Energy Physics, under Contract DE-AC02-07CH11359. (*Corresponding author: Farah Fahim.*)

F. Fahim is with the ASIC Development Group of the Electrical Engineering, Department of the Particle Physics Division, Fermi National Accelerator Laboratory, Batavia, IL 60510 USA, and also with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208 USA (e-mail: farah@fnal.gov).

S. Joshi, S. Ogrenci-Memik, and H. Mohseni are with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208 USA.

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVLSI.2019.2953871

is transferring data off-chip from the previous time frame. The traditional method used for full frame digital data readout is by daisy-chaining all data storage registers to create a long shift register connected to one or multiple external I/Os and clocked at a high frequency to obtain off-chip data transfer [13], [14]. However, this method has the disadvantage of high-power consumption, both from distributing the readout clock signal to every register and from clocking at high frequency. This method can be modified to allow a frame-based zero-suppressed readout by adding logic which skips a pixel with no data, and adds a “flag” which identifies the position of the skipped pixels [15].

Several methods such as data node-based, network-based or a combination of these architectures [16], [17] can be used to move packet-based, zero-suppressed data from the shift register at one end of the column or matrix to the periphery at the opposite end of the ROIC, and subsequently off-chip. The data are essentially moved from the shift register in one pixel to another, possibly with additional registers per group of pixels to alleviate congestion. Several algorithms which give priority to data that had been waiting the longest exist [18], which are easier to implement when the data to be transferred contain timestamps.

One of the simplest methods of data-driven readout is a token passing scheme, where the first pixel to grab the token is enabled to transmit data on a shared bus [19]. The main disadvantage in the token passing scheme is that the readout speed is limited by the time it takes for the token to circle through a given bank of pixels. This limitation can be overcome by adding a fast look-ahead logic [20]–[22].

In other readout methods which employ statistical time division multiplexing, transmission nodes are continuously monitoring the availability of the data bus for transmission. To avoid collisions when two nodes try to access the bus at the same time, one node is arbitrarily granted access to the bus. The other node waits for a random time duration before attempting to transmit again [23].

Neuromorphic designs which use address event representation schemes and readout constraints are even more stringent as these do not operate using time frames but instead require an event-driven approach. Since a predetermined list of pixels is not available, arbiters are required at every node to avoid race conditions when new pixels need to transmit data [24]–[26].

Although we are discussing readout techniques for pixel detectors, it is potentially applicable to other scenarios which require sparse data readouts. This includes access and transfer of data from content addressable memories [27]–[29], sparsifying output in tracking-trigger ASICs [30], population count circuits [31], and networking applications [32].

III. BINARY TREE PE FOR ADDRESS GENERATION AND PIXEL DATA TRANSFER

In zero-suppressed readout techniques the address of the pixel along with its data is required to reconstruct a position-dependent map. The significant advantage of a binary tree PE over the techniques discussed in Section II is that it can simultaneously generate the pixel address while transferring

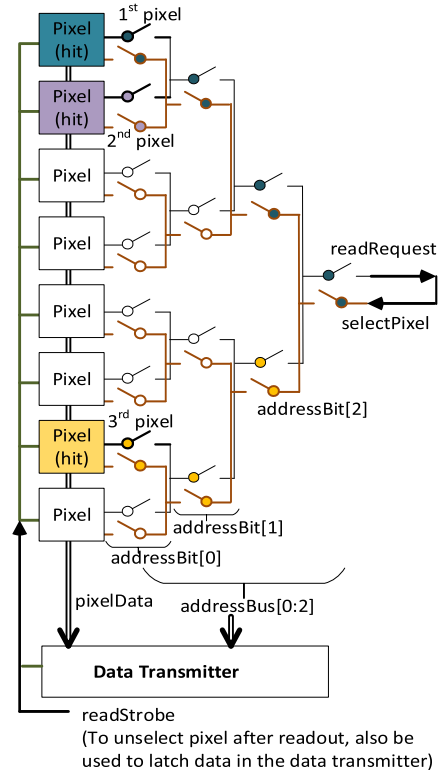


Fig. 1. Pixels with valid data assert a request signal (*readRequest*), which propagates through the binary tree. An acknowledge signal (*selectPixel*) on its reverse path selects the pixel with the highest priority for readout and simultaneously creates the pixel address. A global, independent strobe signal (*readStrobe*) tells the periphery to latch the contents of the selected pixel to the data-output register and disables the *readRequest* signal to remove the pixel’s access to the bus.

data with minimal additional circuitry. A binary-tree PE, first implemented in [33], was originally used as an address generator. The concept can be extended to allow the selected pixel to access a common bus for transferring its data to a peripheral data transmitter, as shown in Fig. 1. The binary tree operates on an address-based priority, whereby pixels with higher address have higher priority. When a pixel has valid data, it asserts a request signal *readRequest*, which propagates through the binary tree, cascading down from the pixel to the periphery. After reaching the root node of the tree, the same signal is propagated back on the reverse path as an asynchronous acknowledge signal *selectPixel* to select the pixel with the highest priority and simultaneously creates the pixel address. However, typically a global, independent, strobe signal *readStrobe* is required by the periphery to latch the contents of the selected pixel and its address to the data-output register. It subsequently disables the *readRequest* signal, removing the pixel’s access to the bus.

The binary-tree PE has several advantages; the high-speed output data transfer clock is localized to a short serial data-output register in the data transmitter, instead of being distributed across the matrix of thousands of pixels. However, this method has several limitations. First, it uses two entirely independent paths, one for selection of pixel for readout and the other to latch data and deselect the pixel. The delays

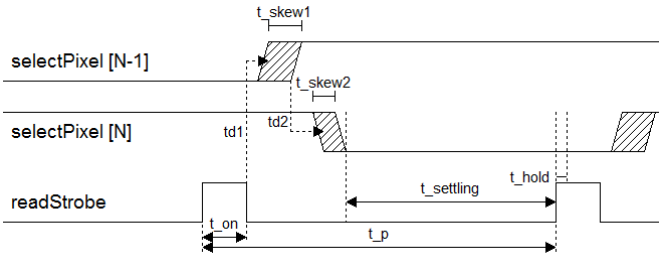


Fig. 2. Timing diagram showing data of pixel $[N - 1]$ being latched by *readStrobe* and subsequently pixel $[N]$ being enabled by *selectPixel [N]* showing the various delays and pixel position-dependent skew. Data are latched at the rising edge of *readStrobe* and the pixel is subsequently disabled at the falling edge. After the current pixel is disabled, the *selectPixel* signal enables the next pixel based on its priority.

through these paths are different, hence to maintain data integrity the *readStrobe* period needs always to be based on the worse case skews and delays. Second, there is a large capacitance on the common bus, requiring a long time for data to settle before being valid for readout.

With reference to Fig. 2, data are latched at posedge of *readStrobe*, and at its negedge, the current pixel's access to the readout bus is terminated. The next pixel is then selected automatically by the combinational logic of the binary tree. For a given readout speed, based on all the delays through the two paths, once a pixel is given access to the bus, the maximum time available for data to settle is

$$t_{settling} = t_p - t_{on} - t_{d1}(\text{readStrobe}) - t_{skew1} - t_{d2}(\text{selectPixel}) - t_{skew2}$$

where t_p is the *readStrobe* period, t_{on} is when *readStrobe* is high (typically much longer than a register's hold time, t_{hold}), t_{d1} is the propagation delay for the independent *readStrobe* from the periphery to the pixel, with t_{skew1} the pixel position-dependent uncertainty in resetting the pixel, t_{d2} is the propagation delay of the *selectPixel* through the arbitration tree, t_{skew2} is the pixel position dependent uncertainty in selecting the next pixel. t_{d1} and t_{d2} are using two different signal networks with different propagation delays. The maximum readout rate is therefore limited by the worst case propagation delay and skew. Moreover, *readStrobe* and *selectPixel* are not derived from a single on-chip clock, and use two entirely different propagation paths. For data to be valid, the data bus settling time should be less than or equal to $t_{settling}$.

The implementation presented in [33] was subsequently optimized in [34] by minimizing the number of transistors required for the logic. The address-encoder and reset-decoder (AERD) proposed in [35] also uses a binary tree arbiter to generate pixel addresses. Instead of the *selectPixel* signal, a *sync* signal is generated by ANDing the *readRequest* signal with a synchronous clock signal used for pixel selection. The positive edge of the clock is used to latch the address at the periphery. At the negative edge of the clock the pixel is disabled and the next pixel is selected. The disadvantage of this scheme is its clock duty cycle inefficiency, half the clock period is used for pixel selection and the other half for generating the address.

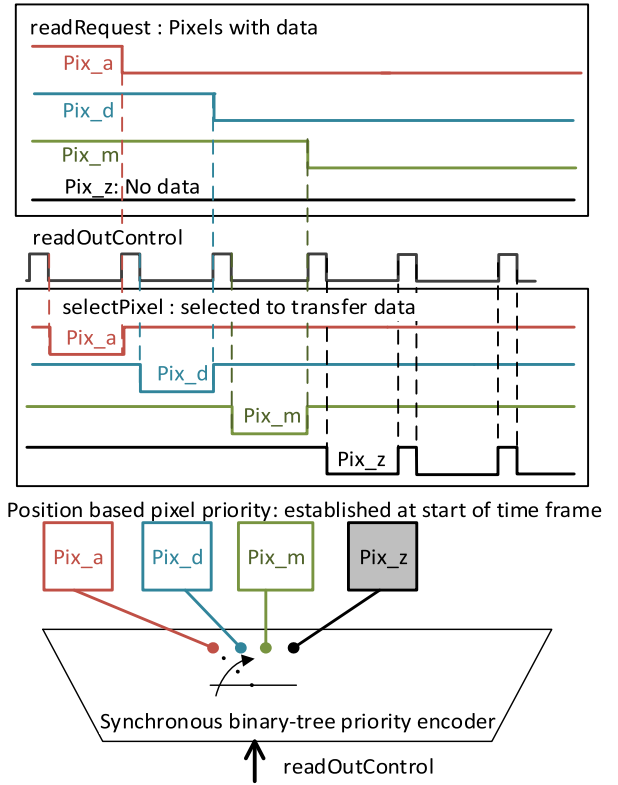


Fig. 3. PE behaving as a commutator switch and splitting the *readOutControl* signal into pulses for each individual pixel based on the priority list. When all pixels with data have been read out, the switch continuously chooses *Pix_z* (the last pixel in the array).

We propose a solution that also does not require a global strobe signal based on a different paradigm [36], [37] by optimally utilizing the clock period, which achieves faster operation as explained in Section IV.

IV. SYNCHRONIZED IMPLEMENTATION OF BUS ARBITRATION

An SB-PE has been developed to overcome the challenges presented in Section III. In this implementation, the pixel's access to a shared bus is "synchronized" and entirely controlled by the output data transmitter without requiring a global *readStrobe* signal.

The concept of the PE is shown in Fig. 3. The SB-PE behaves like a commutator switch selecting the pixels one after another based on their address-dependent priority. A series of synchronous pulses (*readOutControl*) are sent through the binary tree: the first pulse reaches the pixel with highest priority *Pix_a*, the falling edge enables this pixel and the rising edge disables it, the commutator opens the switch with *Pix_a* and closes the switch to *Pix_d*, therefore the next pulse reaches *Pix_d*, and so on. When all the pixels with data have been readout, the commutator switch defaults to the last pixel in the tree, *Pix_z*. However, since it no longer has any valid data, its data output will be "0." *Pix_z* is continuously read out till a frame change occurs and a new priority list is created. Alternatively, it could be gated to stop readout. This eliminates the problem of the two independent data paths *selectPixel*

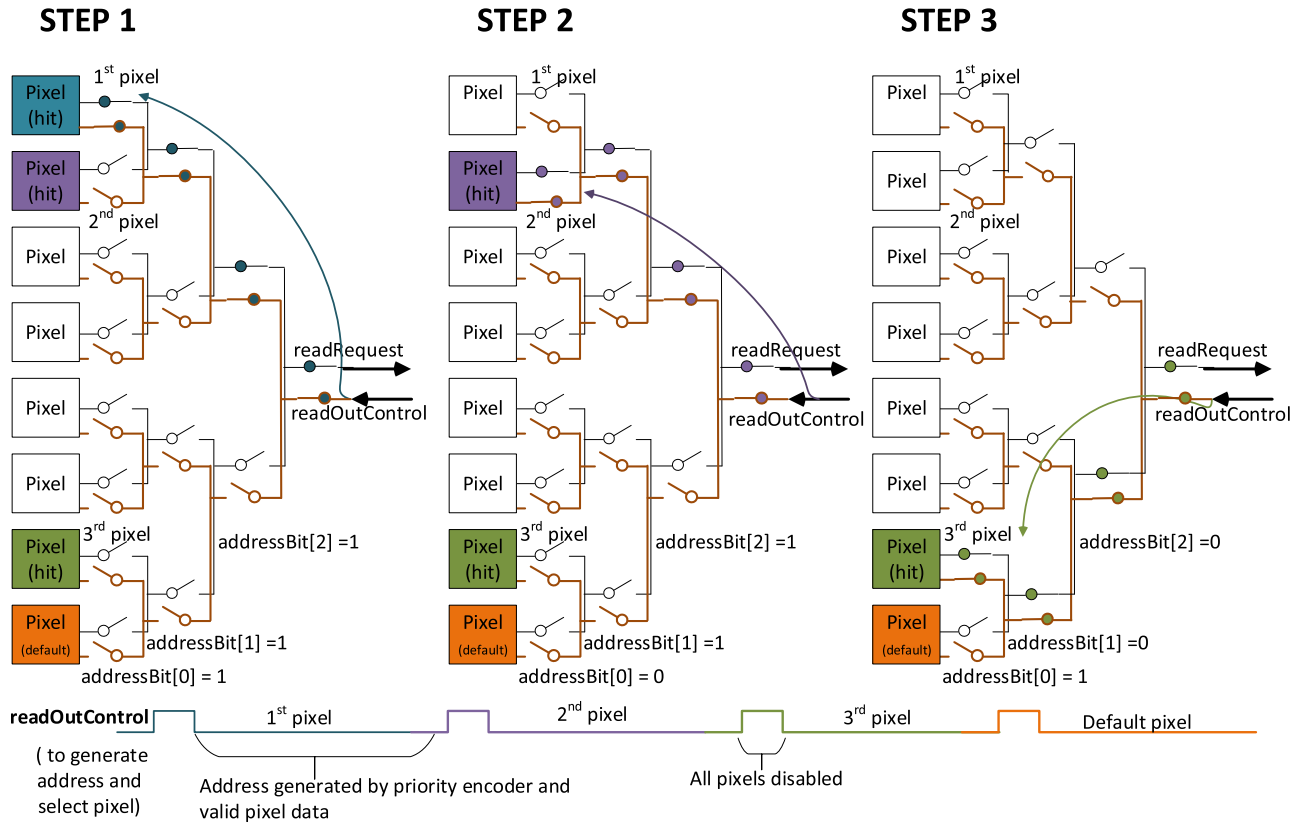


Fig. 4. Binary tree PE is shown for eight pixels. Pulses generated from the data transmitter (*readOutControl*) are broadcast only to the selected pixel by using the path enabled by *readRequest* from the pixel of highest priority to the data transmitter in the opposite direction. The next pulse reaches the pixel with the next highest priority, and so on.

and *readStrobe* selecting and transferring the data from the pixel to the data transmitter. Furthermore, the duty cycle of *readOutControl* signal can be optimized to achieve higher operating speeds.

The reverse path of the binary tree, which is typically used to create the pixel address and allocate the access of the bus to the pixel (*selectPixel*), is also used to send the *readOutControl* signal. This further reduces the switching activity across the matrix by propagating one strobe pulse at a time only to the relevant pixel. The uncertainty in pixel selection time is position dependent. The shortest propagation delay is between two adjacent pixels which share the same parent node of the binary tree. Conversely, pixels with no shared path exhibit the longest delay. With reference to Fig. 4, the selection between the first and second pixels with hits is an example of shortest delay, while the subsequent selection of the third pixel is an example of the longest delay.

The worst case uncertainty can be estimated based on the number of levels in the binary tree. Defining a single propagation step between nodes as a “hop,” the worst case selection delay for an array of 2^N pixels requires $2N$ hops. For a matrix of 2^{10} pixels and a gate delay of ~ 100 ps, this translates to 20 hops, requiring few nanoseconds. Since the *readOutControl* signal is generated by the data transmitter which also latches the data, the two signals use the same data paths. Moreover, *selectPixel* is derived from *readOutControl* and hence their skews are similar. With reference to Fig. 5,

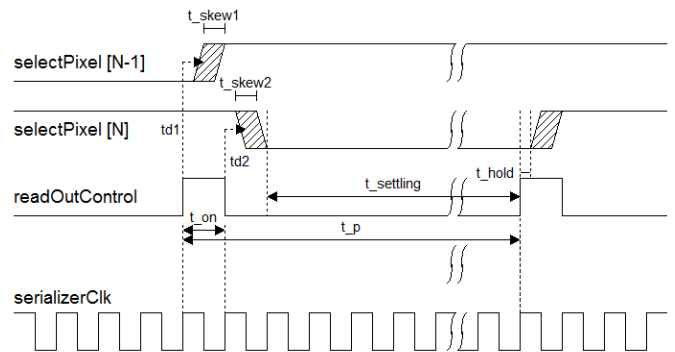


Fig. 5. Timing diagram of the proposed pixel selection scheme. Pixel $[N-1]$ is latched and subsequently disabled at the posedge of *readOutControl* and the pixel $[N]$ is selected at the negedge. The *readOutControl* is generated from *serializerClk*, based on the number of bits in a data packet. t_{d1} and t_{skew1} are now absorbed in t_{on} , so that there is longer time for data to settle for a given readout rate.

the maximum time available for the readout bus to settle is

$$t_{\text{settling}} = t_p - t_{\text{on}}^* - t_{d2}(\text{selectPixel}) - t_{\text{skew2}}$$

where t_p is the *readOutControl* period, t_{on} is when *readOutControl* is “high”; t_{d1} and t_{d2} are the propagation delays of *readOutControl* to arrive at two successive pixels through the arbitration tree, t_{skew1} and t_{skew2} are the corresponding position-dependent uncertainties in releasing

TABLE I

COMPARISON BETWEEN READOUT SCHEMES BASED ON POWER CONSUMPTION AND AREA, FOR 65- AND 130-nm TECHNOLOGY NODES FOR 1024 PIXELS EACH WITH 10 bit OF DATA TRANSMITTING AT A RATE OF 400 Mb/s. TOTAL PER PIXEL AREA REPORTED IS BASED ON 10-bit STORAGE/SHIFT REGISTER

Readout Type:	Serial shift register	Priority encoder	Synchronized priority encoder
Power Consumption 65 nm	51.20 mW	~1 mW	0.63 mW
Power Consumption 130 nm	146.98 mW	~6 mW	3.24 mW
Total Area - 65 nm	211.7 μm^2	~240.0 μm^2	229.5 μm^2
Total Area - 130 nm	600.0 μm^2	~710.3 μm^2	674.1 μm^2

and enabling a pixel, respectively. Data are latched at posedge of *readOutControl*. t_{d1} and t_{skew1} do not appear in this equation, provided $t_{on} > t_{d1} + t_{skew1}$, so that a given settling time can sustain a higher readout rate. t_{d1} must be greater than t_{hold} , but this is easily achieved in modern processes.

readOutControl is derived from the data output *serializerClk*, and its period t_p is set equal to the time it takes to transfer a data packet off-chip. Since the data are latched at the periphery of any of the posedge of the *serializerClk*, which occur before the posedge of *readOutControl*, it can also be used to latch the data for additional safety margin at the cost of shorter $t_{settling}$.

To evaluate the advantages of the proposed architecture, it was compared with a serial shift register as well as with the original PE [33]. The comparison was performed at the register transfer logic (RTL) implementation level using area and power information from the normal V_t standard cell libraries for both 65- and 130-nm CMOS processes for the readout of 1024 pixels, each with 10 bits of data to be read out at 400 Mb/s. The analysis is approximate but conservative, since the power consumption for the PE does not account for the increased number of buffers required for the clock tree distribution of the *readStrobe* signal, since the pixel area is larger than the synthesized area of the readout. Table I clearly highlights the benefit of the SB-PE in terms of both power and area. As expected, irrespective of the readout scheme, both power consumption and area decrease as the technology scales. For an 8% marginal increase in area the SB-PE has around 50–80 times lower power consumption than the serial shift register. The elimination of buffers required for clock tree synthesis of the *readStrobe* signal effectively reduces the power of the SB-PE by almost a factor of 2 compared to the standard PE.

V. PIXEL LEVEL IMPLEMENTATION

The pixel level implementation of the logic which interfaces with the readout is shown in Fig. 6. The operating time frame is defined by a global, user-defined *frameClk*. As mentioned earlier for deadtimeless operation, the pixel includes two sets of data registers, one for storing data from the current frame

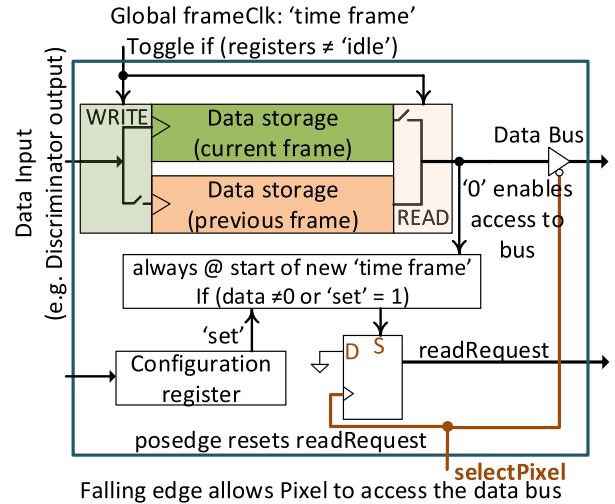


Fig. 6. Conceptual circuit level implementation of a pixel with two sets of data registers for current and previous time frames. A *readRequest* signal is activated (high) at the start of a frame if pixel has data or in frame-based readout. The falling edge of the *selectPixel* signal allows the pixel to access the common data bus and the rising edge disables the pixel and resets (low) the *readRequest* signal. A global signal changes the time frame, which cycles the data storage registers for readout.

and the other for sending data out from the previous frame. At the rising edge of the *frameClk*, the registers are toggled by changing switch positions if new data are available for readout. Simultaneously, the *readRequest* signal is asserted to indicate that the pixel has data.

For full frame imaging additional pixel logic is used to “set” the *readRequest* signal for all pixels. When the *selectPixel* signal is “low,” the tristate buffer is enabling pixel access to the data bus. Subsequently, the rising edge of the *selectPixel* signal resets and cancels the *readRequest*.

For triggered zero suppressed readout applications, such as those implemented for front-end readout of particle detectors at the LHC, the scheme as shown in Fig. 6 should be complemented by a memory buffer of sufficient depth to accommodate trigger latency. Depending on the pixel size and trigger latency, such memory can be placed in the periphery or in each pixel. The selection and transfer of data from memory would require additional logic for matching the trigger bunch crossing ID with the time stamp of the stored data.

The switches in Fig. 4 are implemented as combinational logic gates as shown in Fig. 7. The circuit associated with a PE for four pixels shows that the higher order address bits can be computed and settle faster than the lower order bits.

If only full frame imaging is required, the readout architecture uses a cascade of alternating *leafNOR* and *leafNAND* cells without any address generation circuitry. Hence, for a matrix of 1024 pixels, 512 *leafNAND* cells and 511 *leafNOR* cells are required, no additional buffers are added.

A. Distribution of Signals and Placement of Cells

A digital-on-top design methodology is used to assemble the readout architecture of the ROIC.

The distribution of the *readRequest* and *readOutControl* signals is done by the place-and-route (P&R) tool to minimize the

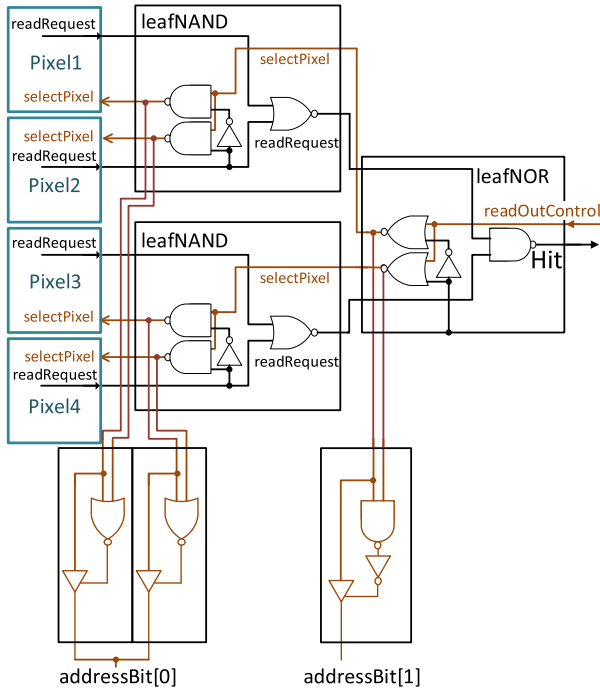


Fig. 7. PE for four pixels showing the propagation of *readRequest* from pixel to the periphery, the pixel selection using the *readOutControl* signal uses the opposite path through the arbitration tree to reach only one pixel at a time. The address generation logic is also shown, which can be omitted if implementation is only for full frame readout.

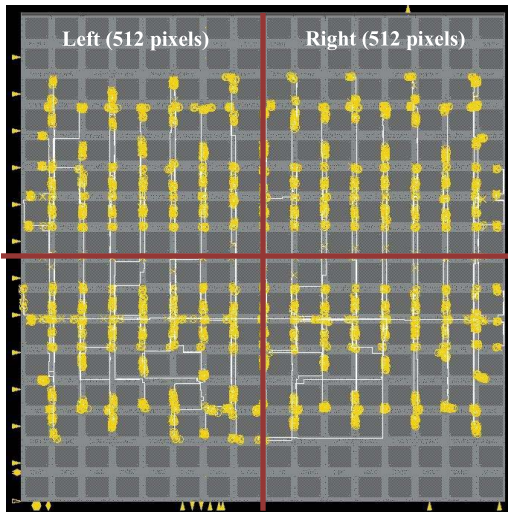
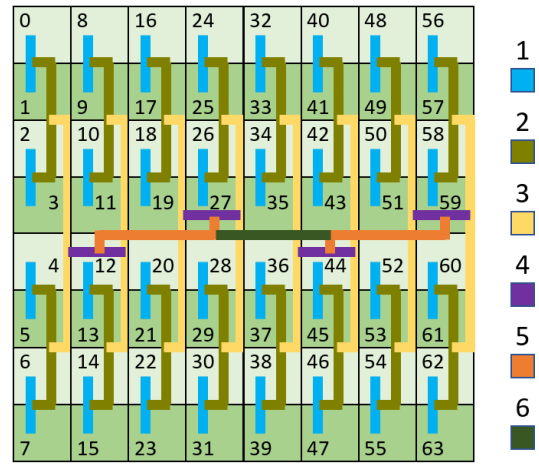


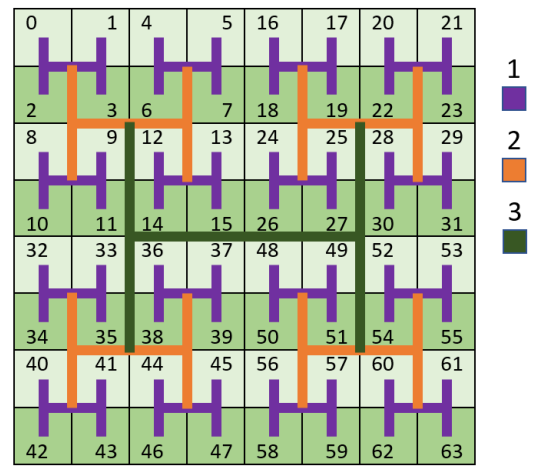
Fig. 8. Network of *readRequest* signal from each pixel cascades through the PE for requesting bus access across the matrix of 32×32 pixels. The P&R tool distributes the readout logic of *leafNOR* and *leafNAND* cells symmetrically across the matrix as shown by the yellow nodes.

skew across the ROIC as shown in Fig. 8. The distribution and the placement of the *leafNAND* and *leafNOR* is symmetrical across the matrix using a clone placement strategy.

The *readOutControl* signal requires a maximum of nine gate delays and six physical hops to reach from the output serializer controller to the pixel. The first three hops are through the central row and the next three are within every double column. The reduced number of hops is achieved



(a)



(b)

Fig. 9. (a) Implemented solution results in a column-based 1-D tree with fan-out only in the horizontal or vertical direction. (b) “2-D H-tree” approach results in the least number of hops. Changing the pixel numbering pattern automatically allows the P&R tool to use a 2-D H-tree for distribution.

by physically clustering the data registers of four pixels. Owing to the column-wise pixel numbering, the P&R tool uses a 1-D tree to distribute the signal across the matrix as shown in Fig. 9(a). However, to further reduce skew and the trace length of these signals a “H-Tree” distribution must be implemented [38]. If the pixel numbering is redefined as shown in Fig. 9(b), then the P&R tool automatically generates a “H-tree.” Fig. 9 illustrates the concept for 64 pixels but the actual implementation is for two independent 512 pixels corresponding to Fig. 10.

B. Increasing Speed Using the Ping-Pong Approach

As the number of pixels in an array sharing the common data bus increases, so does the bus capacitance. The higher the bus capacitance the longer it takes for data to settle and be valid before it is latched by the peripheral data transmitter.

To overcome the bus settling time, an array of pixels (e.g., 32×32) can be divided into two halves (512), each with its own arbitration logic as shown in Fig. 10. This allows for interleaved latching of data from the two pixel banks into

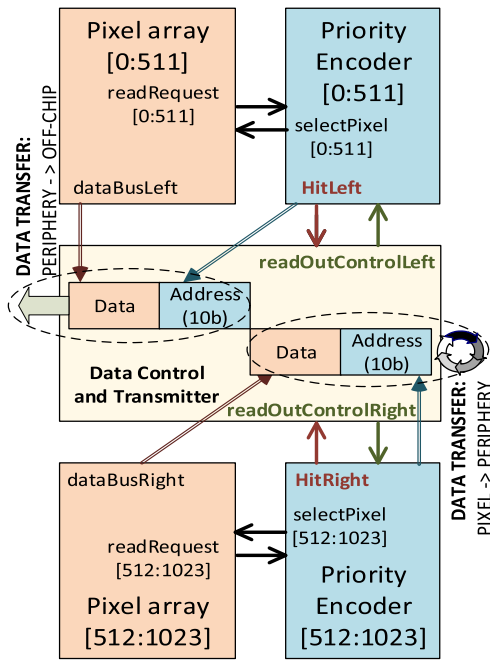


Fig. 10. Pixel array divided into two halves for interleaved ping-pong pipelining of data from pixel to the central data control and transmitter block, which contains an output serializer for serial data transfer and also generates the *readOutControl* signal derived from the *serializerClk* for pixel selection.

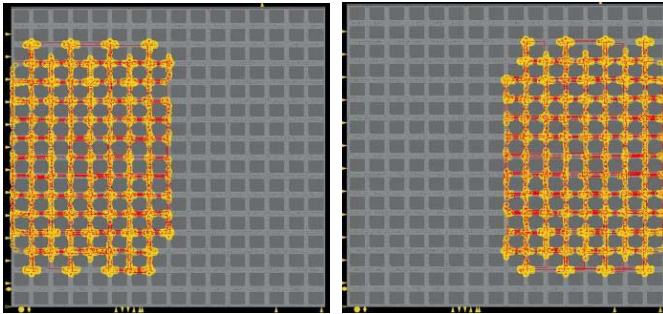


Fig. 11. Highlighting of the physical placement of the output data buses from two halves of the matrix. Interleaving the two allows the data bus a longer time to settle ensuring data integrity, while also reducing the total bus capacitance by a factor of 2.

two parts of an output serializer within the data transmitter, which operates as a one-stage pipeline, increasing the time available for settling by a factor of 2. When one bank is being readout the other bank is being latched, increasing the speed without compromising settling time.

The ping-pong pipelining ensures sufficient settling time for the next valid data to be transferred from a pixel to the output serializer, which is less than the time it takes to readout the current data packet. Additionally, it reduces the parasitics on the shared address and data output buses by a factor of 2 [39]. The independent output data buses on the left and right half of the matrix is shown in Fig. 11.

VI. EXPERIMENTAL RESULTS

The SB-PE has to date been implemented in two ROICs. The first is a 192×192 pixel array [39] in 130-nm LP-CMOS,

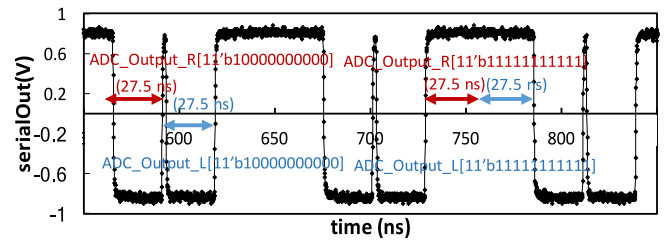


Fig. 12. Serial output data corresponding to pixel output ADC bus switching from $11'b1000000000$ to $11'b1111111111$ at 400-MHz serializer clock.

with both full frame as well as zero-suppressed readout capabilities. The second is a 64×64 pixel array in 65-nm LP-CMOS [7], exclusively with full-frame readout, which does not require the address generation logic. The test results presented in this section are for this second ROIC, which is the only one available so far. For deadtimeless operation a full frame readout speed of ~ 40 kfps or higher needs to be demonstrated which is equal to the time it takes for a single ADC conversion cycle. This full-frame rate is achievable if the output serializer is operating at 400 MHz. The goal of testing was to determine the speed, reliability, and power consumption of the readout architecture.

A high-Vt standard cell library was used for this implementation to reduce the power consumption by a factor of 2 compared to the normal-Vt library. The chip is divided into four quadrants with one dedicated output pin for every 1024 pixels, with one SB-PE per 512 pixels to interleave pixel readout from the left and the right halves of the matrix, as presented in Fig. 10.

The data controller and transmitter block serially transmits 1-bit range and 10-bit of ADC output data per pixel, alternating between the left and right halves.

When considering the speed limitations of the data readout architecture, the worst case occurs when all the lines of the 11 bit shared parallel data output bus, which transfers data from the pixel to the periphery, transition from “0 to 1” or “1 to 0” for every consecutive data transfer cycle.

For this test, the ROIC is configured such that 12.5% of the pixels contain ADC data corresponding to all 1’s, with each of them followed in the full-frame readout sequence by a pixel with no ADC data. With the range bit set to 1, this causes the output bus to switch at every readout cycle from $11'b1111111111$ to $11'b1000000000$ for 25% of the frame. Fig. 12 captures such transitions at the output of the ROIC, for a serializer clock frequency of 400 MHz.

The power consumption of the ROIC power supplies for analog, digital, and LVDS sections can be independently monitored. The digital power consumption includes the readout power as well as ADC digital control logic. The power consumption of the readout is measured by recording the power during the full-frame readout and by subtracting the dynamic power corresponding to the ADC conversion (which can be independently measured by turning off the data readout *serializerClk*).

Power consumption is plotted against supply voltage shown in Fig. 13 as well as serializer clock frequency shown in Fig. 14, which both demonstrate good agreement between

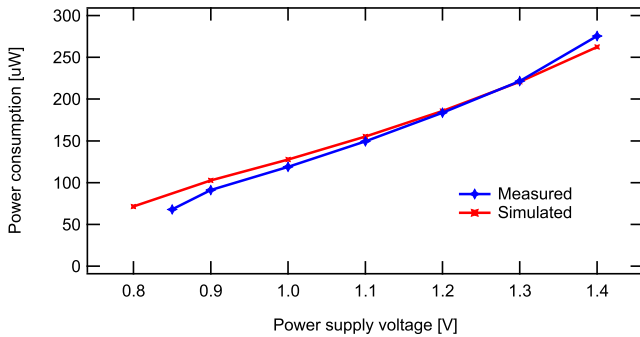


Fig. 13. Readout power consumption versus power supply voltage for 1024 pixels with the serializer clock operating at 200 MHz.

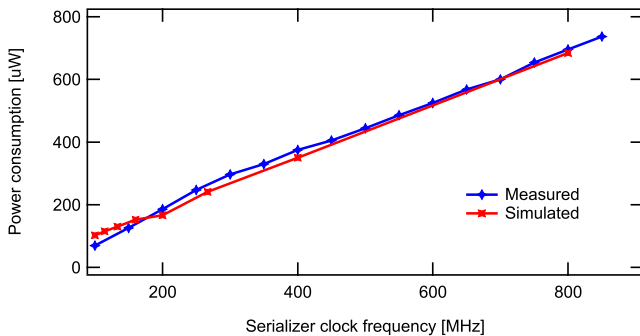


Fig. 14. Readout power consumption versus serializer clock frequency for 1024 pixels for the nominal digital power supply of 1.2 V.

measurements and simulations. Simulations include post-layout parasitic capacitance per address line of up to 1 pF. The nominal operating voltage is 1.2 V. The readout fails progressively when lowering supply voltage: at 0.85 V only half of the chip is working, with only one active matrix of 1024 pixels, while no output is observed at 0.8 V.

To measure the output data bit error rate (BER), 1000 frames were analyzed. Each frame consists of 11286 bits of data, starting with a 22-bit header [10101010100000000000], followed by 11-bit data per pixel from 1024 pixels. The output serializer is operated for a range of frequencies from 100 to 800 MHz, and the number of errors for 10 million bits is determined. These measurements do not use a pseudorandom bit stream (PRBS) but instead are based on the worse case test pattern generated by toggling ten lines of the pixel data bus to min/max values for consecutive data transfer cycles. No bit errors were observed up to 550 MHz, at 1.2-V power supply, after which BER increases as shown in Fig. 15. These measurements were also performed for a fixed clock frequency of 200 MHz, and varying the digital power supply voltage from 900 mV to 1.4 V. No bit errors were observed for supply voltages above 1.0 V.

In this implementation for full-frame readout, which does not include address generation, the energy per bit measurement at the nominal operating voltage of 1.2 V and at a serializer output frequency of 400 MHz is 0.94 pJ/bit. This design reads 1024 pixels of $50 \times 50 \mu\text{m}^2$, distributed across an area of $1.6 \text{ mm} \times 1.6 \text{ mm}$ optimized by the digital-on-top, P&R design assembly. It compares favorably with other binary tree readouts [35].

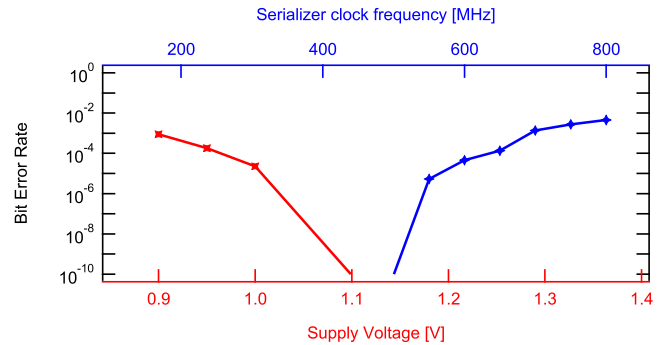


Fig. 15. BER versus output serializer clock frequency, performed at 1.2-V power supply. No bit errors were detected up to 550 MHz. BER versus power supply voltage, performed at 200-MHz output serializer clock frequency. No bit errors were detected above 1-V power supply. The BER test is based on min/max switching of the data bus used to transfer data from pixel to periphery.

VII. CONCLUSION

In this article, the SB-PE readout for pixel detectors was presented. It is a low-power, high-speed bus arbitration technique for data transfer from pixels to periphery. It is suitable for both full frame readout as well as zero-suppressed readout. This technique minimizes switching activity across the matrix by allowing only one single pixel to be active at a time. Furthermore, it eliminates the need to synchronize between the two independent signals for pixel selection and deselection. Simulations and measurements show the viability of the design and demonstrate significant power saving. Unlike mentioned in [35], this readout is not organized as a thin long column which is susceptible to large sidewall capacitance from closely packed metal tracks but instead by subchips [39]. The subchip approach is particularly suited to take advantage of data transfer through the backside of the chip utilizing low capacitance, small diameter through silicon vias [40] for future large chip implementation. The full chip assembly process of segmenting subchips based on resource optimization is elaborated in [41]. If several parallel output ports are not available, a hierarchical nested approach could be used. Encoding the address value at each level can further decrease the power consumption and save area [42]. Furthermore, the digital-on-top P&R assembly can easily scale the architecture for implementation to different pixel geometries and technologies.

Although we demonstrated our technique for pixel detectors, it is potentially applicable to other scenarios which require sparse data readouts. This includes access and transfer of data from content addressable memories, sparsifying output in tracking-trigger ASICs, population count circuits and networking applications.

ACKNOWLEDGMENT

The authors would like to thank Fermilab colleague and Group Leader G. Deptuch for suggesting the improved form of a priority-encoder-based readout controller for use in pixel detectors. They would also like to thank the Fermilab ASIC Design Group Staff and Particle Physics Division Management for supporting this article. They would also like to thank S. Holm, Fermilab, and L. Kadłubowski, AGH-UST

Cracow for developing the readout system to test the chip described in this article. This document was prepared by using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359.

REFERENCES

- [1] R. Ballabriga *et al.*, "Review of hybrid pixel detector readout ASICs for spectroscopic X-ray imaging," *J. Instrum.*, vol. 11, Jan. 2016, Art. no. P01007.
- [2] Z. Yu *et al.*, "Evaluation of conventional imaging performance in a research whole-body CT system with a photon-counting detector array," *Phys. Med. Biol.*, vol. 61, no. 4, p. 1572, 2016.
- [3] X. Kleczek, P. Grybos, R. Szczygiel, and P. Maj, "Single photon-counting pixel readout chip operating up to 1.2 Gcps/mm² for digital X-ray imaging systems," *IEEE J. Solid-State Circuits*, vol. 53, no. 9, pp. 2651–2662, Sep. 2018, doi: [10.1109/JSSC.2018.2851234](https://doi.org/10.1109/JSSC.2018.2851234).
- [4] X. Llopart, R. Ballabriga, M. Campbell, L. Tlustos, and W. Wong, "Timepix, a 65k programmable pixel readout chip for arrival time, energy and/or photon counting measurements," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 581, nos. 1–2, pp. 485–494, 2007.
- [5] C. Zhang, S. Lindner, I. M. Antolović, J. M. Pavia, M. Wolf, and E. Charbon, "A 30-frames/s, 252×144 SPAD flash LiDAR with 1728 dual-clock 48.8-ps TDCs, and pixel-wise integrated histogramming," *IEEE J. Solid-State Circuits*, vol. 54, no. 4, pp. 1137–1151, Apr. 2019.
- [6] S. Kleinfelder, S. Lim, X. Liu, and A. El Gamal, "A 10000 frames/s CMOS digital pixel sensor," *IEEE J. Solid-State Circuits*, vol. 36, no. 12, pp. 2049–2059, Dec. 2001.
- [7] G. A. Carini *et al.*, "Hybridized MAPS with an in-pixel A-to-D conversion readout ASIC," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 935, pp. 232–238, Aug. 2019.
- [8] T. Kishishita, T. Hemperek, H. Krüger, M. Koch, L. Germic, and N. Wermes, "A 10 MS/s 8-bit charge-redistribution ADC for hybrid pixel applications in 65 m CMOS," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 732, pp. 506–510, Dec. 2013.
- [9] O. S. Cossairt, D. Miao, and S. K. Nayar, "Camera systems and methods for gigapixel computational imaging," U.S. Patent 9473 700, Oct. 18, 2016.
- [10] ATLAS—Inner Detector. Accessed: May 1, 2019. [Online]. Available: <https://atlas.cern/discover/detector/inner-detector>
- [11] O. G. Shpyrko, "X-ray photon correlation spectroscopy," *J. Synchrotron Radiat.*, vol. 21, no. 5, pp. 1057–1064, 2014.
- [12] R. H. Hadfield, "Single-photon detectors for optical quantum information applications," *Nature Photon.*, vol. 3, no. 12, pp. 696–705, 2009.
- [13] F. Fahim, G. Deptuch, S. Holm, A. Shenai, and R. Lipton, "Monolithic active pixel matrix with binary counters ASIC with nested wells," *J. Instrum.*, vol. 8, Apr. 2013, Art. no. C04008.
- [14] Q. Zhang *et al.*, "Submillisecond X-ray photon correlation spectroscopy from a pixel array detector with fast dual gating and no readout dead-time," *J. Synchrotron Radiat.*, vol. 23, no. 3, pp. 679–684, 2016.
- [15] T. S. Poikela, "Readout architecture for hybrid pixel readout chips," Ph.D. dissertation, Turku Centre Comput. Sci., Finland, 2015. Accessed: May 1, 2019. [Online]. Available: <https://cds.cern.ch/record/2042198/files/CERN-THESIS-2015-111.pdf>
- [16] T. Poikela *et al.*, "Architectural modeling of pixel readout chips Vexipox and Timepix3," *J. Instrum.*, vol. 7, no. 1, 2012, Art. no. C01093.
- [17] T. Poikela *et al.*, "Digital column readout architectures for hybrid pixel detector readout chips," *J. Instrum.*, vol. 9, no. 1, 2014, Art. no. C01007.
- [18] N. McKeown, A. Mekikittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999.
- [19] I. Perić *et al.*, "The FEI3 readout chip for the ATLAS pixel detector," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 565, no. 1, pp. 178–187, 2006.
- [20] K. Einsweiler *et al.*, "Dead-time free pixel readout architecture for ATLAS front-end IC," *IEEE Trans. Nucl. Sci.*, vol. 46, no. 3, pp. 166–170, Jun. 1999.
- [21] A. Himmi, A. Doziere, O. Torheim, C. Hu-Guo, and A. Winter, "A zero-suppression micro-circuit for binary readout CMOS Monolithic sensors," in *Proc. Top. Workshop Electron. Part. Phys. (TWEPP)*, 2009.
- [22] C. Hu-Guo *et al.*, "CMOS pixel sensor development: A fast read-out architecture with integrated zero suppression," *J. Instrum.*, vol. 4, no. 4, 2009, Art. no. P04012.
- [23] A. Quayum, Y. Kakishima, H. Minn, and H. Papadopoulos, "Non-orthogonal pilot designs with collision detection capability for grant-free access," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
- [24] K. A. Boahen, "A burst-mode word-serial address-event link-I: Transmitter design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 7, pp. 1269–1280, Jul. 2004.
- [25] K. A. Boahen, "A burst-mode word-serial address-event link-II: Receiver design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 7, pp. 1281–1291, Jul. 2004.
- [26] S. Furber, "Large-scale neuromorphic computing systems," *J. Neural Eng.*, vol. 13, no. 5, 2016, Art. no. 051001.
- [27] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 712–727, Mar. 2006.
- [28] J. R. Hoff, G. W. Deptuch, S. Joshi, T. Liu, J. Olsen, and A. Shenai, "VIPRAM_L1CMS: A 2-tier 3D architecture for pattern recognition for track finding," in *Proc. IEEE Nucl. Sci. Symp., Med. Imag. Conf. Room-Temp. Semiconductor Detector Workshop (NSS/MIC/RTSD)*, Oct./Nov. 2016, pp. 1–6.
- [29] S. Joshi *et al.*, "Multi-V_{dd} design for content addressable memories (CAM): A power-delay optimization analysis," *J. Low Power Electron. Appl.*, vol. 8, no. 3, p. 25, 2018.
- [30] C.-C. Wang, J.-S. Wang, and C. Yeh, "High-speed and low-power design techniques for TCAM macros," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 530–540, Feb. 2008.
- [31] L. Frontini, V. Liberali, and A. Stabile, "A very compact population count circuit for associative memories," in *Proc. 7th Int. Conf. Mod. Circuits Syst. Technol. (MOCAS)*, May 2018, pp. 1–3.
- [32] D. H. Summerville, J. G. Delgado-Frias, and S. Vassiliadis, "A flexible bit-pattern associative router for interconnection networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 7, no. 5, pp. 477–485, May 1996.
- [33] P. Fischer, "First implementation of the MAPHISTO binary readout architecture for strip detectors," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 461, nos. 1–3, pp. 499–504, 2001.
- [34] G. W. Deptuch *et al.*, "Design and tests of the vertically integrated photon imaging chip," *IEEE Trans. Nucl. Sci.*, vol. 61, no. 1, pp. 663–674, Feb. 2014.
- [35] P. Yang *et al.*, "Low-power priority address-encoder and reset-decoder data-driven readout for monolithic active pixel sensors for tracker system," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 785, pp. 61–69, Jun. 2015, doi: [10.1016/j.nima.2015.02.063](https://doi.org/10.1016/j.nima.2015.02.063).
- [36] F. Fahim and G. W. Deptuch, "Edgeless large area ASIC," U.S. Patent 20170023405 A1, Jan. 26, 2017.
- [37] G. W. Deptuch. (2019). *Fermilab Technical Note*. [Online]. Available: <http://lss.fnal.gov/archive/test-tm/2000/fermilab-tm-2709-ppd.pdf>
- [38] S. Fok and K. Boahen, "A serial H-tree router for two-dimensional arrays," in *Proc. 24th IEEE Int. Symp. Asynchronous Circuits Syst.*, May 2018, pp. 78–85.
- [39] F. Fahim, G. W. Deptuch, J. R. Hoff, and H. Mohseni, "Design methodology: Edgeless 3D ASICs with complex in-pixel processing for Pixel Detectors," *Proc. SPIE, Opt. Sens., Imag., Photon Counting, Nanostruct. Devices Appl.*, vol. 9555, Aug. 2015, Art. no. 95550M, doi: [10.1117/12.2188153](https://doi.org/10.1117/12.2188153).
- [40] G. W. Deptuch *et al.*, "Fully 3-D integrated pixel detectors for X-rays," *IEEE Trans. Electron Devices*, vol. 63, no. 1, pp. 205–214, Jan. 2016.
- [41] F. Fahim, "Assembly of edgeless four side tileable ROICs for a wafer scale, deadzone-less camera," *IEEE Trans. Nucl. Sci.*, to be published.
- [42] J. Georgiou and A. G. Andreou, "High-speed, address-encoding arbiter architecture," *Electron. Lett.*, vol. 42, no. 3, pp. 170–171, Feb. 2006, doi: [10.1049/el:20063914](https://doi.org/10.1049/el:20063914).