2022 17th Conference on Ph.D Research in Microelectronics and Electronics (PRIME) | 978-1-6654-6700-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/PRIME55000.2022.9816805

Investigation of Timing Properties for an Event Driven with Access and Reset Decoder Readout Architecture for a Pixel Array

Dominik Gorni, Grzegorz Deptuch, Sandeep Miryala
Instrumentation Division
Brookhaven National Laboratory
Upton, NY, United States
{dgorni, gdeptuch, smiryala}@bnl.gov

***Abstract*—The large number of data generating sources (data channels) on a single chip requires appropriate techniques to manage a readout from these channels. One of the main methods is sharing a medium of transmission, which requires arbitration to avoid collisions or deadlocks. Existing solutions face several problems such as a dead time, unintended prioritization or metastability. That is why we decided to create a new readout architecture named EDWARD i.e., Event Driven with Access and Reset Decoder. The EDWARD architecture gets rid of the earlier mentioned problems and mitigates the other ones. However, due to the use of logic circuits outside a standard cell library, which are hard to characterize, we were challenged to perform an additional transient analysis to validate the architecture. Here we show a methodology and the results of the simulations. Based on the results obtained we can confirm the functional correctness of the system and plan the optimization of operating conditions in order to achieve better performance. Our goal is to use the EDWARD architecture in the future radiation detectors to be built at Brookhaven National Laboratory.**

Keywords—readout, pixel, sensor, verification, arbitration

I. INTRODUCTION

The large and growing number of data sources, i.e., channels that reside on a single integrated circuit, requires the development of readout techniques that maximize throughput while minimizing the resources used. These channels may include, but are not limited to, the signals from strip radiation sensors as well as pixel radiation sensors used for example in the high energy physics [1]. The output from such channels can be either analog, digital or a combination of both. These data must then be read out and sent to peripheral processing units or off-chip for acquisition and further analysis. With a few data channels one can afford a direct and dedicated link, however with a larger number of channels one may encounter problems related to, among others, limited space, limited routing area, power consumption. But more importantly, it is a suboptimal solution, because, in most cases, these links will remain in a state of idling for a new data. Therefore, for optimization purposes, a link can be shared among several data channels. However, this requires implementation of access management to the link, or more generally to the shared resources that compose the link. The existing solutions are discussed in Section II. Also, a novel, *patent pending*, Event Driven with Access and Reset Decoder (EDWARD) readout architecture [2] is presented there. Section III describes the methodology by which the timing properties were measured for an 8x8 pixel array built based on the EDWARD architecture. In section IV we discuss and summarize the results obtained from the simulations that were conducted.

II. DIFFERENT READOUT ARCHITECTURES

A. State of the art

One of the simplest methods of readout is based on the principle of polling consecutive channels to see if they have data ready to be read out. In order to notify a channel that it has the right to use shared resources, one can introduce a so-called token, which is passed from channel to channel. The transfer can be done synchronously, to the tick of a clock, or asynchronously with a strobe signal followed by a checking of the token's location. A channel with data ready to be read out should hold the token until the readout is finished. This is known as the "Token Passing" architecture [3]. One of the main limitations when using this architecture is the length of the chain into which the channels can be grouped - the longer the chain, the longer the time required to poll all the channels. For this reason, among others, a better solution is an architecture based on event-driven readout. Such an example is the AERD architecture, i.e., "Address Encoder Reset Decoder [4], which consists of a multi-stage arbitration tree. It uses priority logic to determine which channel gets permission to use shared resources. However, this solution has two significant drawbacks - due to the purely combinatorial nature of the arbitration tree, glitches may occur when a higher priority channel makes a readout request at a time when the request from a lower priority channel has already been propagated. The second problem is the prioritization imposed from before, which makes some channels more privileged than the others. The other methods of channel access management

978-1-6654-6700-1/22/$31.00 ©2022 IEEE

can be found in a literature, but many problems persist. These include a need to use an external clock or strobe signal to check data availability, metastability problems caused by synchronous latching of asynchronously arriving data [5], the need to divide the access granting operation into phases which introduces dead time when data from any of the channels are not available. Due to the drawbacks of the existing so far solutions, we developed the new readout architecture that is called EDWARD.

B. The EDWARD architecture

The EDWARD readout architecture is shown in Fig. 1. It distinguishes four major functional blocks: an in-channel logic, an arbitration tree, a synchronization and management circuitry, and a shared data bus with the default state setting circuitry. These four blocks work together to

- receive an information that the data are ready to be read out (rdy signal),
- send a request (req signal) to access the shared bus,
- asynchronously, without direct distribution of the clock signal to each channel, transmit the request signal to the synchronization unit (rqo signal) with simultaneous arbitration if there are multiple requests,
- transmit an acknowledge token (acki signals) to the channel (ack signal) that wins the arbitration, while granting permission to this channel for uninterrupted and exclusive access to the bus,
- drive the data to the bus,
- define an access time frame to the channel and, if necessary, to drive several data packets from the same channel in an uninterrupted manner with the multiple time frames,
- switch immediately, without the dead time, between channels if there is still at least one readout request after completing the readout,
- establish a default bus state when no channel is currently being read out in order to ensure bit stream continuity when using e.g., a serializer to send data off-chip and distinguish between data coming from a channel and data at idle state.

The arbitration tree is built based on a binary tree structure, in which the building block is an arbitration cell. There are several variants of the arbitration cell and one of them is shown in Fig. 2. The presented variant is the one that was selected for the development of pixel detector at BNL. In this variant, two arbitration stages and a commuting circuit can be distinguished. The first one prevents glitches that could trigger the readout of multiple channels simultaneously causing a conflict in the bus access. The second stage is the actual arbitration between readout requests and allows a request to be selected based on the time it reached the arbitration cell. The commuting circuit establishes a path for the arbitration token in the direction from which the successful request comes and passes the request up the tree. A second request made after the first request has already been processed has no capability to break the path for the acknowledge token as long as the first request remains active.

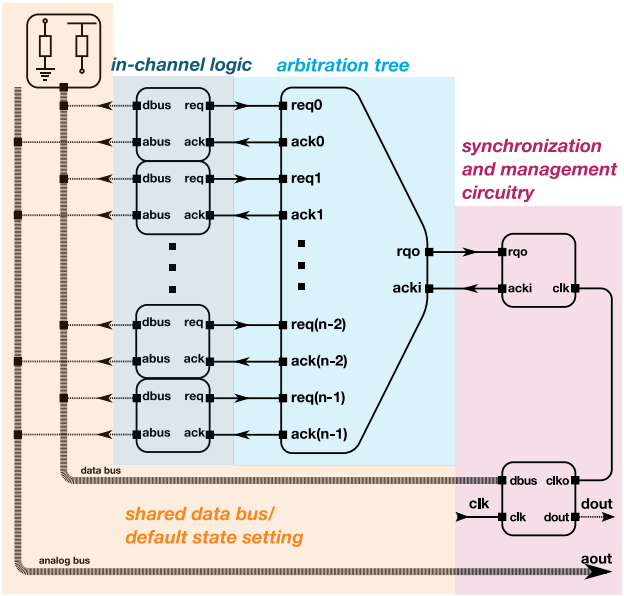

Fig. 1. EDWARD readout architecture overview

This has been achieved by using an arbiter cell based on the so-called Seitz arbiter [6]. The arbiter cell, depicted in Fig. 3, consists of an SR latch and a filter circuit for glitches that may occur when two readout requests are received almost at the same time. The two implementations using different logic states for the active state are possible, allowing them to be used alternately in the arbitration tree without the need for additional inverters (also the commuting circuit must be accordingly adapted to the logic states). Using the SR latch in a circuit where the inputs can change asynchronously carries the possibility of the glitches as well as a delay. Both result from the possibility of a metastable state at the output of the latch. The first problem was solved using the previously mentioned filter, while the second should not interfere with the functional operation of the readout circuit except for the additional time required to read data out from a channel. To test the correctness of this statement, we decided to perform additional transient simulations and check the timing properties of the EDWARD readout architecture implemented using a 65 nm CMOS process.

III. TRANSIENT SIMULATION

To investigate the timing properties of the EDWARD readout architecture, we implemented, using digital circuit synthesis and physical implementation tools, an 8x8 pixel array. We then performed a parasitic extraction and created a testbench

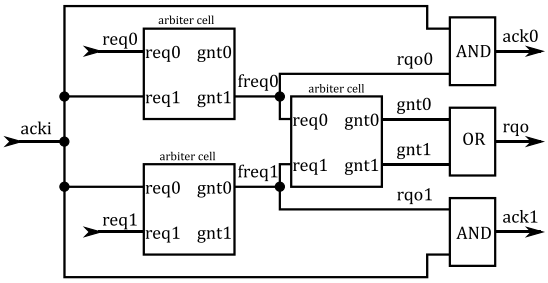

Fig. 2. Arbitration cell – a building element of the arbitration tree

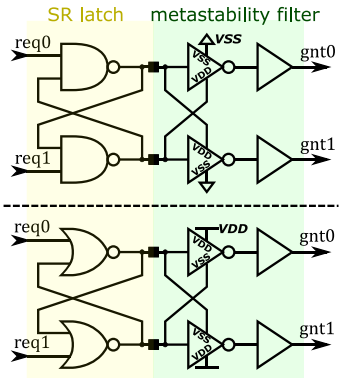

Fig. 3. Implementation of the arbitration cell using a) NAND b) NOR gates

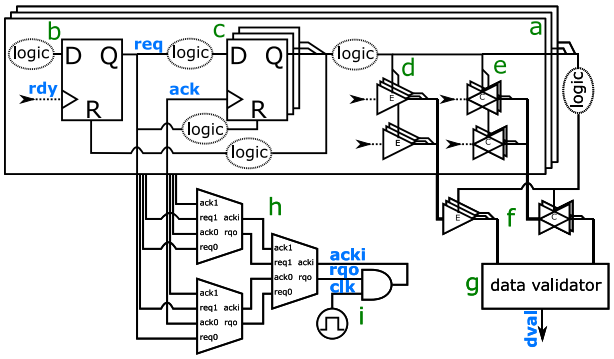

Fig. 5. Simplified structure of the readout system

including the resulting design. The pixel array is intended to be use as a building block for a larger array of the radiation detector circuit by tiling this small block in a bigger structure. A simplified structure of the readout system based on the EDWARD architecture is shown in Fig. 5. It consists of the pixel array along with the in-channel logic (*a*). The logic includes a flip-flop that sends a readout request in response to a data ready signal (*b*). An additional logic allows, among other things, to disable the requests and to clear the request after a completed readout operation. A group of flip-flops sensitive to changes in the acknowledge token signal (*c*) is responsible for the internal state of the in-channel logic. They indicate that the data bus can be accessed, count down the number of data packets to be read out, and indicate that all data has been sent and the readout should complete, and the token be returned. Groups of tri-state (*d*) and transmit (*e*) gates are used to load data from the pixel onto the data buses - digital and analog, respectively. An additional buffering stage (*f*) reduces the effective capacity seen by each buffer. A data validator (*g*) is a circuit that checks if the digital data lines have reached a level of 70% of the final value, which means that the data can be considered loaded on the bus and then sends a data valid (dval) signal. The system also poses the arbitration tree (*h*) and a circuit that generates acknowledge tokens based on the provided clock (*i*). The expected time dependencies between signals during the operation of the readout system are depicted in Fig 4. The time intervals that can be used to characterize the readout architecture are also marked. The most important of these are t_s and t_h, i.e., the setup and hold times, respectively, on the digital data bus. They determine the time frame, relative to the clock signal, at which the data should be latched in peripheral circuits such as a serializer. In addition, the propagation time of the token in the arbitration tree is marked as t_t and the redistribution time of the token from one channel to another as t_{rd}. The latter can be used to optimize the duty cycle for the clock that generates the tokens. Such operation can maximize the data bus access time, defined as an inactive state (the high logic state in the presented case) on the acknowledge line. Measuring both times also allows to check on potential arbitration tree malfunctions, as values that deviate significantly from the mean values may indicate that the readout circuit locks up. The last time is the pixel reset time *tr* which determines the minimum time after which the chip can process the next event.

The target sensor chip is to consist of 100k pixels, and readout is to be done through 8 serial data outputs at 250 Mbps rate. Hence, the clock used to generate the acknowledge tokens is 17.86 MHz. The data ready signal for each channel was simulated as a Poisson process with an average interval value of 2µs. The last value was chosen to observe both the overlap of several readout requests and the idle state. The simulation time was 256 µs, during which each pixel was read out several times, as shown in Fig. 6. The graph shows the total number of readouts from each individual channel. The resulting simulation data were processed, and the interested time values calculated using a program written in LabVIEW. An example of a histogram for the setup time is shown in Fig. 7. It shows the distribution of measured values for the setup time from all channels. A more detailed view is provided by the error graph with marked standard deviation for each channel as shown in Fig. 8. Due to the unrealistic task of examining all possible redistribution paths and collecting multiple samples a graph of this type cannot be drawn for the redistribution time. A good insight into the timing properties in this case is given by the intensity plot shown in

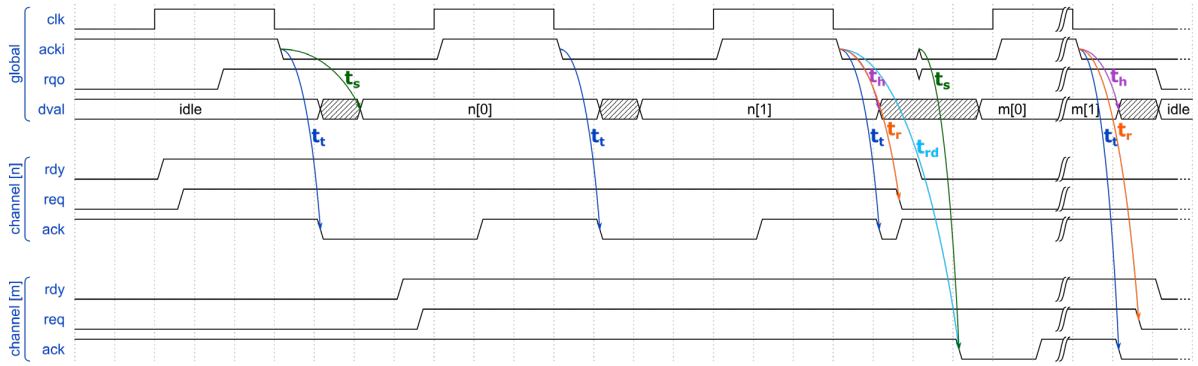

Fig. 4. Expected waveforms of the signals and time dependencies of the system built based on the EDWARD architecture

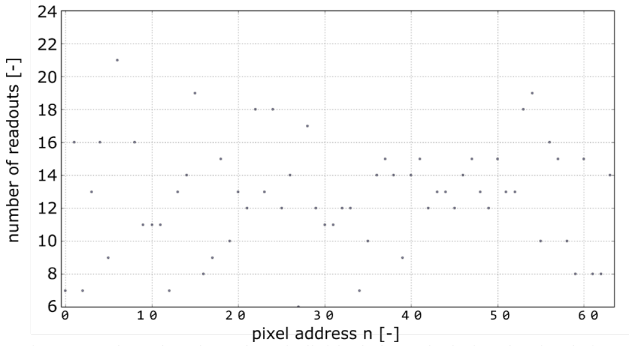

Fig. 6. Number of readouts from individual channels during the simulation

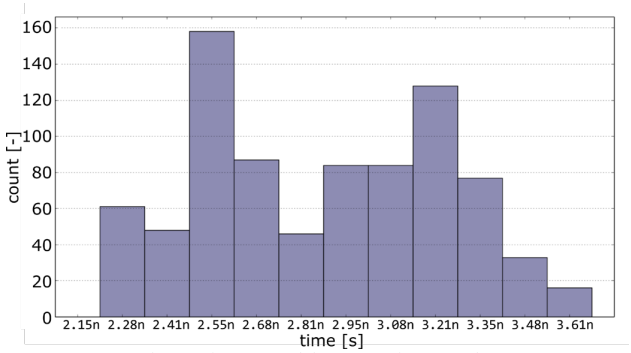

Fig. 7. Histogram of data setup time samples

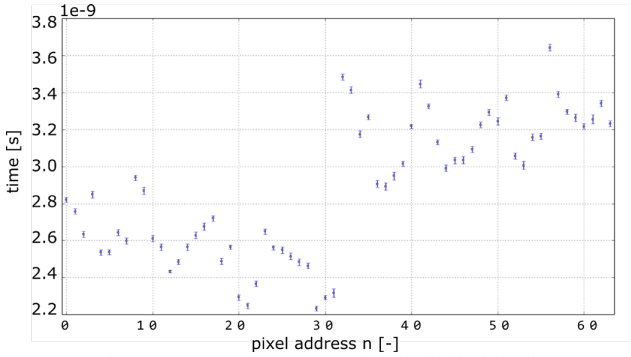

Fig. 8. Data setup time error graph for individual pixels

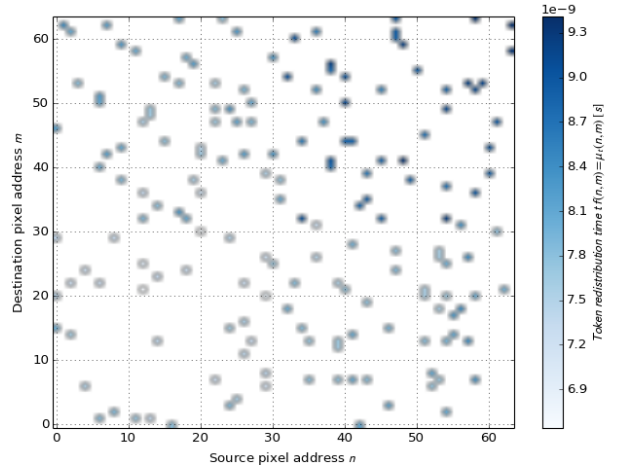

Fig. 9. Intensity graph of the token redistribution time from channel n to m

Fig. 9. The intensity indicates the time required to redistribute the token from channel *n* (*x*-axis) to channel *m* (*y*-axis).

IV. Summary

Analysis of the histograms and graphs for all the timing properties investigated shows no malfunction in the readout system. There are no outliers, and the standard deviation for each individual pixel is comparable. It is also possible to notice a certain pattern in which the mean time values for each pixel are arranged, which is visible mainly at the transition from pixel number 31 to pixel 32. This has its origin in the binary tree structure and is expected. For the same reason, larger values for the redistribution time can be observed in the upper right part of the intensity plot and the smallest values for the transitions are located in the lower left corner. Summary statistics for all the studied parameters of the analyzed system are collected in TABLE I. Based on the collected data, especially the mean value, the standard deviation as well as the maximum redistribution time, the maximum duty cycle of the clock from which the acknowledge tokens are generated can be determined. For the presented system, with clock frequency equal to 17.86 MHz, duty cycle, defined as a ratio of the time the token is inactive to the clock period, should be ≤ 82%. We are fully satisfied with the obtained results. The performed analyses allow us to establish a general procedure for designing chips containing the readout logic based on the EDWARD architecture.

TABLE 1. SUMMARY OF THE TIMING PROPERTIES

Name	***Number of samples***	***Minimum value***	***Maximum value***	***Mean value μ***	***Standard deviation σ***
Data setup time	822	2.21ns (29)[a]	3.68ns (56)	2.88ns	366ps
Data hold time	819	2.50ns (29)	3.97ns (56)	3.23n	352ps
Pixel reset time	821	3.54ns (21)	5.29ns (56)	4.33ns	416p
Token propagation time	2464	0.79ns (20)	1.66ns (56)	1.19ns	237pa
Token redistribution time	307	6.64ns (12 -> 21)	9.60ns (56 -> 61)	8.05ns	608ps

[a.] values in brackets indicate for which channel the given value was observed

References

[1] P. Grybos et al., “Ultra fast X-ray detection systems in nanometer and 3D technologies”, in 2014 Proceedings of the 21st International Conference MIXDES, 2014, pp. 38–41

[2] D. Gorni et al., "Event driven readout architecture with non-priority arbitration for radiation detectors", *Journal of Instrumentation*, vol. 17, no. 04, p. C04027, 2022.

[3] G. Deptuch et al., “A Vertically Integrated Pixel Readout Device for the Vertex Detector at the International Linear Collider*”, IEEE Transactions on Nuclear Science*, vol 57, no 2, pp. 880–890, 2010.

[4] P. Yang et al., “A Monolithic Active Pixel Sensor prototype for the CEPC vertex detector”, *Nucl. Instrum. Methods in Phys. Res.*, vol 924, pp. 82–86, 2019.

[5] F. Lin et al., “ASIC implementation of a data push architecture for silicon pixel readout”, in *1994 Meeting of the American Physical Society, Division of Particles and Fields (DPF 94)*, 10 1994, pp. 1948–1952.

[6] C. L. Seitz, “Ideas about arbiters”, *Lambda*, no 1, pp. 10–14, 1980.