The Design of a Stitched, High-dynamic Range CMOS Particle Sensor

Master of Science Thesis

Jiaqi Zhu (4233476)

July 2014

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

The project of this thesis was in cooperation with Caeleste CVBA.
Their support and assistance are hereby gratefully acknowledged.

The Design of a Stitched, High-dynamic Range CMOS Particle Sensor

Master of Science Thesis

Jiaqi Zhu (4233476)

Supervisor: Prof. dr. ir. Albert J.P. Theuwissen

In Partial Fulfillment of the Requirements

For the Degree of Master of Science in Electrical Engineering at Delft University of Technology

July 2014

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

Delft University of Technology

Department of

Electrical Engineering

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled

The Design of a Stitched, High-dynamic Range CMOS Particle Sensor

by

Jiaqi Zhu

in partial fulfillment of the requirements for the degree of

Master of Science Microelectronics

Date: _____

Supervisor(s):

Prof. dr. ir. Albert J.P. Theuwissen

Dr. ir. Benoit Dupont

Committee member(s):

Prof. dr. ir. Albert J.P. Theuwissen

Dr. ir. André Bossche

Dr. ir. Benoit Dupont

Abstract

This thesis presents the design of a stitched, high-dynamic range particle sensor in 0.18 µm technology, which has three different configurations of 4096*4096 (4K), 8192*8192 (8K) and 12288*12288 (12K) pixels. It can be operated at 65 MHz with a target of 5 e- RMS noise and 92 dB dynamic range.

The architecture of the whole sensor is explained with a functional illustration of every block. The pixel is based on a 4T pinned photodiode structure, and the high dynamic range is realized with an extra MIM capacitor in the pixel. The pixel Y-axis addressing and control are accomplished by a block called pixel row driver. For the pixel readout, in readout columns, correlated double sampling is implemented which significantly reduces the reset or kTC noise. The signal in the columns will be multiplexed and sent to 8 readout lines. Those signals on 8 lines are sent to single-ended to differential converters, and finally are read out via buffers. The digital control signals, such as the address of pixel array, signals to switch between different modes, are created from a block called digital control block, which works as an interface between an FPGA and the internal circuits.

The functional simulation with parasitic resistance and capacitance and the layout of every block and the whole sensor is illustrated. The noise simulation and result analysis also are given.

Key words: high dynamic range, stitched design, CMOS image sensor

Table of Contents

List of Figures

List of Tables

Acknowledgements

Experts say that to be a master in one field, it needs 10,000 hours study and work on the average, which is equal to 5 years. So after 5 years learning of microelectronics, I finally had the chance to do a real silicon design. Here, I want to express my gratitude to all those people who helped me during this thesis work. Without them, I wouldn't experience the real fun of image sensor design.

First of all, I would like to give my sincere thanks to Prof. Albert Theuwissen, my supervisor at TU Delft. It is his lively and humorous teaching in the image sensor course that attracted me to choose this direction. His responsible guidance and instructive suggestions help me in every step in the thesis writing.

Moreover, I am particularly grateful to my daily supervisor Dr. Benoit Dupont at Caeleste. It is his trust on my ability that makes me get this opportunity to join this project. He showed me what the qualities of a mature designer. His patient instructions and encouragement helped me to overcome many difficulties.

I would also like to wholeheartedly thank Prof. Dr. Bart Dierickx, CTO of Caeleste and Dr. Arnaud Defernez, senior designer of Caeleste. Bart gave me many inspiring instruction during this design, and reminds me things need to be paid attention to as I am just a junior designer. Arnaud is my partner in this design. He taught me what the right steps are in a real design, discussed with me to solve the problems together and encouraged me to implement my ideas in the design.

Additionally, I want to thanks all my colleagues at Caeleste: especially Patrick Henckes, Bert Luyssaert, Peng Gao, Qiang Yao, Dirk Uwaerts and many others, who have provided suggestions and helps during my work.

I would like to thanks all the other teachers taught me for their patience and kindness to make me reach present stage.

Finally, I would like to express my love to my parents and friends supporting and understanding me.

Chapter 1 Introduction

With the fast development of image sensors, the trends of increasing fields image sensor can be applied into, such as scientific research, medical equipment, aerospace detector, etc., gives demands on image sensors' specifications. For instance, medical X-ray sensors for CT application always need a large pixel area [1.1], space application requires low noise and wide dynamic range [1.2] and for real-time motion detection, advanced integrated signal-processing circuits are necessary to achieve high speed [1.3]. Other applications like sensitivity to other wavelengths or particles sometimes need a combination of the specifications all above. Thus, a custom design sensor is the call of the market.

In this thesis project, a large-area high-dynamic range (HDR) particle sensor is introduced. Thanks to stitching technique, very high resolution is realized and the sensor can be arranged into 4096*4096 (4K), 8192*8192 (8K) and 12288*12288 (12K) configurations. This project work is executed at Caeleste CVBA, Mechelen, Belgium.

In this introductory chapter, the motivation of this project is discussed. Then the challenges in this design are analyzed and a comparison of the state of the art HDR sensors is illustrated. Furthermore, an overview of this thesis project is presented. The last part is the organization of the thesis.

1.1.Motivation and challenges

The human eye has a wide dynamic range of about 140 dB. In contrast, most solid-state image sensors have a dynamic range in the order of 50 to 70 dB [1.4]. Thus, stretching the dynamic range in various ways to combine both low and high illumination in one image arouses the interests of sensor designer.

It is commonly known that in a standard CMOS manufacturing process, the chip size is limited to the available field size of the reticle [1.5]. However, with stitching, a single image sensor can be built from a sequence of exposures to produce a device that is many times larger than the size of a single mask [1.6]. Facing the need of market and maturity of the technology, an extremely high resolution HDR sensor is worth designers' effort.

Here lists some key specifications of HDR sensors in recent years. Since the sensor in this work has three different configurations, we take 4K configuration as an example. From the comparison of this work with others, we can see that our sensor has a competitive higher resolution and a considerable dynamic range in the middle-level.

Table 1-1: Key specification of state of the art HDR sensor and this work

	Pixel number	Frame rate [fps]	HDR [dB]	Pixel size [μm]	Technology [μm]
Commercial available	1024*512	30	154	5.6	N/A
	752*480	60	100	6.0	N/A
	1280*1080	60	120	4.2	N/A
Paper available [1.7]; [1.8]; [1.9]	320*240	37	80	2.25	0.13
	320*240	36	91	2.25	0.13
	1000*1028	30	87.5	7.1	0.18

Caeleste present design	4096*4096	39	92	6.5	0.18

There are many challenges in this design: the organization of the architecture of the sensor to make it stitch-able, the balance between power consumption and speed and the design-for-yield when the chip size goes up.

1.2.Thesis project overview

In this project, a 0.18 µm image sensor CMOS process is used. The high dynamic range is achieved by a novel pixel based on 4T pinned photodiode structure with an extra capacitor.

Taking 4K configuration as an example, the sensor's whole architecture is illustrated in Figure 1.1.

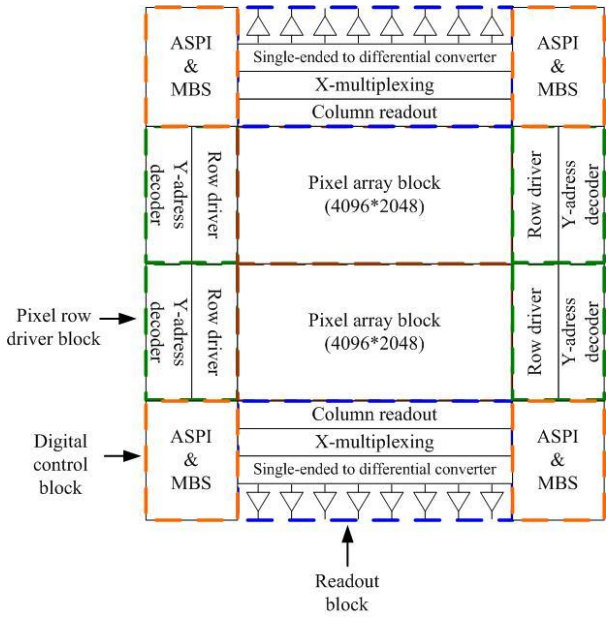


Figure 1.1: Overview of sensor architecture (4K configuration)

To realize a stitched design, the sensor is divided into four blocks and the details of each block are discussed in Chapter 3.

- Pixel array block: One pixel array contains 4096*2048 pixels. Pixel size is 6.5 µm* 6.5 µm.
- Pixel row driver block: It contains the Y-axis addressing decoder for pixel row selection and row driver to drive the transistors in the pixel.
- Digital control block: It contains addressable serial parallel interface (ASPI), mixed boundary scan (MBS) and other interface and housekeeping sub-circuits.
- Readout block: It contains the column readout, X-axis multiplexing, single-ended to differential signal converters and the final output buffers.

Table 1-2 illustrates some key specifications, the details of those specifications will be explained in Chapters 3 and 4.

Table 1-2: Specifications of the sensor

Specifications	Value
Pixel size [µm]	6.5

Charge conversion [μV/e-]	High gain: 100 Low gain: 4
Dynamic range [dB]	92
RMS readout noise	High gain measurement: 5 e- Low gain measurement: 50 e-
Frame rate [fps]	Low dynamic range: 4K: 58; 8K: 59; 12K: 39 High dynamic range: 4K: 39; 8K: 29; 12K: 20
Shutter mode	4K: Global shutter 8K & 12K: Rolling shutter
Correlated double sampling (CDS)	Global shutter: off-chip Rolling shutter: on-chip
Averaging	2*2
ADC	Off-chip
Power consumption [W]	4K: about 2.2 8K: about 4.5

1.3.Thesis organization

This thesis consists of 5 chapters. Besides the first introductory chapter, the rest is arranged as follows.

Chapter 2 explains the principle of conventional pixel operation, the non-idealities in an image sensor and the concepts of the methods to increase the dynamic range. Features as binning and different shutter modes are discussed as well.

Chapter 3 analyses the sub-circuit implementation in each block. The functional simulation and layout of each block are given.

Chapter 4 shows the top sensor functional validity, noise analysis and simulation and top layout. Meanwhile, some extra structures help for chip testing are also mentioned.

Chapter 5 contains the conclusions of the thesis and a discussion of the future research.

1.4.References

[1.1] Turchetta, R., "Large area CMOS Image Sensor for X-ray imaging", IMAGIC seminar days 2010-Frontiers in IR and X-ray Imaging, Nov.2010.

[1.2] Takeda, A., Arai, Y., Tsuru, T., Tanaka, T., Nakashima, S., Matsumura, H., Imamura, T., Ohmoto, T., and Iwata, A., "Development of New Circuit with CSA for X-Ray Astronomical SOI Pixel Detector", IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), pp. 1-4, Oct. 2013.

[1.3] Zhu, H. and Shibata, T., "A Real-Time Motion-Feature-Extraction VLSI Employing Digital-Pixel-Sensor-Based Parallel Architecture", IEEE Transaction on Circuits and Systems for video Technology, No.99, Mar. 2014.

[1.4] El Gamal, A., 392b Lectures and Classnotes, Stanford University, 2001.

[1.5] "Stitching for CMOS Image Sensors", available at http://www.advancedimagingpro.com/print/Advanced-Imaging-Magazine/Stitching-for-CMOS-Image-Sensors/1$126, Jul. 2008.

[1.6] "The Art of Making Large Image Sensors (or Quilting with Silicon)", available at http://blog.teledynedalsa.com/2012/02/the-art-of-making-large-image-sensors-or-quilting-with-silicon/, Feb. 2012.

[1.7] Bae, J., Kim, D., Hwang, I. and Song, M., "A High Dynamic Range CMOS Image Sensor with a Digital Configurable Logarithmic Counter", 2013 European Conference on Circuit Theory and Design (ECCTD), pp. 1-4, Sep. 2013.

[1.8] Lee, J., Baek, I., and Yang, K., "Memoryless Wide-Dynamic-Range CMOS Image Sensor Using Nonfully Depleted PPD-Storage Dual Capture", IEEE Transaction on Circuits and Systems-II: Express Briefs, Vol. 60, No. 1, Jan. 2013.

[1.9] Seo, M., Sawamoto, T., Akahori, T., Iida, T., Takasawa, T., Yasutomi, K. and Kawahito, S., "A Low Noise Wide Dynamic Range CMOS Image Sensor With Low-Noise Transistors and 17b Column-Parallel ADCs" IEEE Sensors Journal, Vol. 103, No. 8, pp. 2922-2929, Aug. 2013.

Chapter 2
Background of CMOS image sensors

A brief introduction of most common-used pixel structures is present in section 2.1. An overview of non-idealities in image sensor, the definition of dynamic range and signal-to-noise ratio, and common methods of HDR are discussed in section 2.2. Moreover, the special characters as different binning methods and shutter modes used in image sensor are discussed in section 2.3 and section 2.4.

2.1.Basic pixel structures

2.1.1. Active pixel [2.1]

Active pixel sensor (APS) is the mainstream of modern imager sensors, because it solves a lot of noise problems with a reasonable performance to cost ratio. One basic 3T (3 transistors) structure of the photodiode APS pixel is shown in Figure 2.1. There is an in-pixel buffer amplifier implemented which can be configured as a source follower. From Figure 2.1, we can see in every pixel operation cycle, the photo-generated electrons are collected on the reverse-biased photodiode and decrease the voltage across. This voltage will be read out through a source follower when the row select transistor (RS) closed. At reset, the photodiode is reset by transistor RST, and the reverse voltage across the photodiode will reach its maximum value. The small voltage drop on the diode after the reset is due to the cross-talk of the reset pulse. The timing diagram of APS pixel is shown in Figure 2.2.

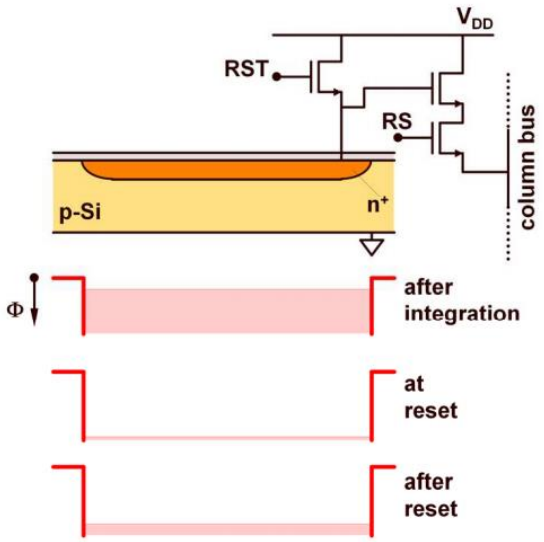


Figure 2.1: Schematic and principle of 3T APS pixel [2.1]

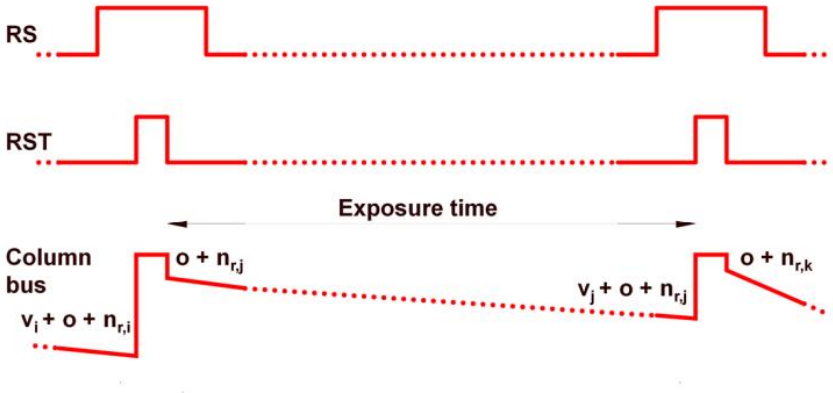


Figure 2.2: Timing diagram of 3T APS pixel [2.1]

Though the APS pixel is popular in modern CMOS design, it suffers from a large reset or kTC noise. Thus, a new pixel structure using pinned photodiode (PPD) which offers better performance in cancelling the kTC noise of reset starts to get people's attention.

2.1.2. Pinned photodiode [2.1]

The biggest difference between a PPD pixel and a general APS pixel is that a pinned photodiode replaces the conventional photodiode. Figure 2.3 shows a 4T PPD pixel structure. A pinned photodiode is a fully depleted buried diode. The diode is shield form SiO_2/Si interface by p+ shallow implantation, which results in a lower dark current. The diode is coupled with a transfer gate (TX), allowing complete charge transfer from the PPD to the floating diffusion node (FD). The use of intra-pixel charge transfer offers the opportunity to reduce the noise by correlated double sampling (CDS).

According to Figure 2.3 and Figure 2.4, the working principles of the PPD pixel are 4 steps as follows. At first, the FD is reset via transistor RST, the voltage across the FD reaches its maximum value. Then, the small voltage drop on the diode after the reset is due to the cross-talk of the reset pulse and the reset value V_{reset} is read out by closing the RS switch. Next, the photo-generated electrons are transferred from the PPD to the FD when TX is closed and the signal value V_{signal} is read out. CDS will compute V_{reset} - V_{signal}, which almost cancels out the reset noise, as well as off-set variations from pixel to pixel. Finally, the next integration will start.

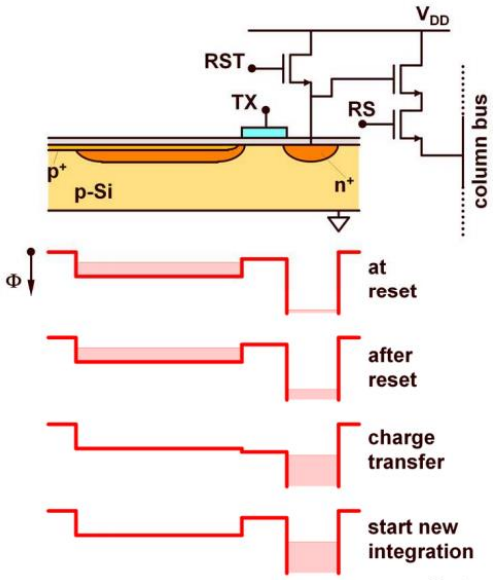


Figure 2.3: Schematic and principle of 4T PPD pixel [2.1]

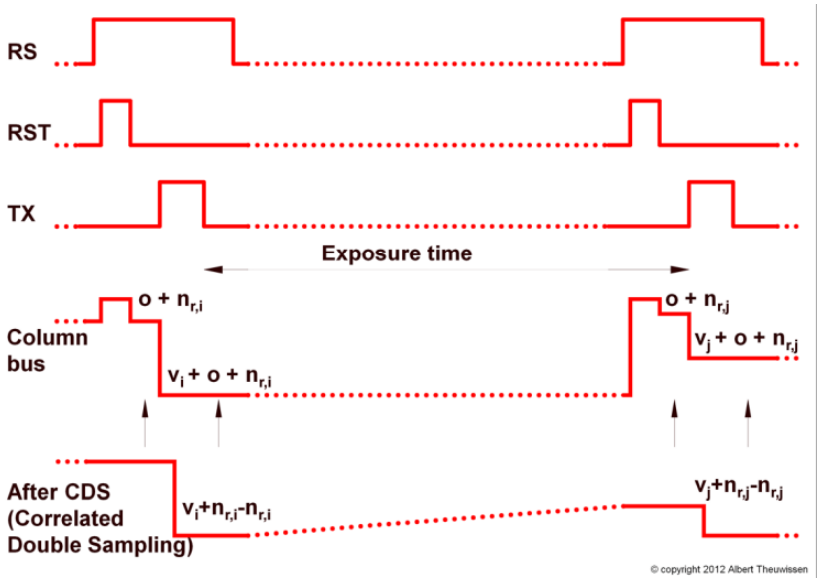


Figure 2.4: Timing diagram of 4T PPD pixel [2.1]

CDS can be realized in many ways, like [2.2] contains a two-stage analog CDS, [2.3] does CDS in digital domain. This design also implements column CDS, the detail is discussed in Chapter 3.

2.2.Overview of non-idealities in image sensor

2.2.1. Non-idealities in image sensor

The non-idealities in an image sensor can be briefly categorized into three things [2.4]: temporal noise, fixed pattern noise (FPN) and dark current.

Temporal noise consists of thermal, shot, and 1/f noise. Three main sources are pixel shot noise; pixel reset noise, and readout noise.

Fixed pattern noise (FPN) is the spatial variation in pixel outputs, due to device and interconnects mismatches. Two most general mismatches are offset and gain mismatch. In some literature, only the offset mismatch is called FPN, the gain mismatch is called the photo response non-uniformity (PRNU) [2.5].

Dark current is the leakage current at the integration node. It is a current not induced by photo generation, but due to junction and transistor leakages. It limits the image sensor dynamic range by introducing dark integration noise, which is also called dark current shot noise (DCSN). It varies widely across the image sensor array causing dark signal non-uniformity (DSNU) that cannot be easily removed [2.5].

2.2.2. Dynamic range and signal to noise ratio [2.1]

Dynamic range (DR) quantifies the ability of a sensor to adequately image both high lights and black shadow in one scene. It is defined as

$$DR = 10log(\frac{P_{SAT}}{P_{MIM}}) = 20log(\frac{N_{SAT}}{N_{MIM}})\ dB \tag{2-1}$$

Where

P_{SAT} is the maximum signal power and P_{MIM} is the minimum detectable signal power. Since signal power is proportional to the number of carriers per second, the equation can also be expressed as a ratio between the maximum number of photo charges at saturation N_{SAT} over the minimum number of detectable photo charges N_{MIM}. N_{MIM} also can be understood as the pixel noise level without illumination represented in electrons.

Signal to noise ratio (SNR) is the ratio of the signal power to the noise power at a given input level, which is defined as

$$SNR = 10log(\frac{P_{signal}}{P_{noise}}) = 20log(\frac{N_{signal}}{N_{noise}})\ dB \tag{2-2}$$

Where

P_{signal} is the signal power and P_{noise} is the noise power at the given signal level. N_{Signal} is the number of photo charges of the signal and N_{noise} is the number of charges represented by noise.

2.2.3. General methods to increase DR

According to equation (2-1), the DR is limited by the minimum detectable photons and the number of saturated photons in the photodiode. So to increase N_{SAT}, or decrease N_{MIM} or do both will help to enlarge the dynamic range. The most common ways in HDR sensors are all based on that concept.

The common-used HDR methods can be classified into three categories [2.6]: nonlinear response, multiple capture and time to saturation detection. Figure 2.5 illustrates the output transfer function curves of these methods.

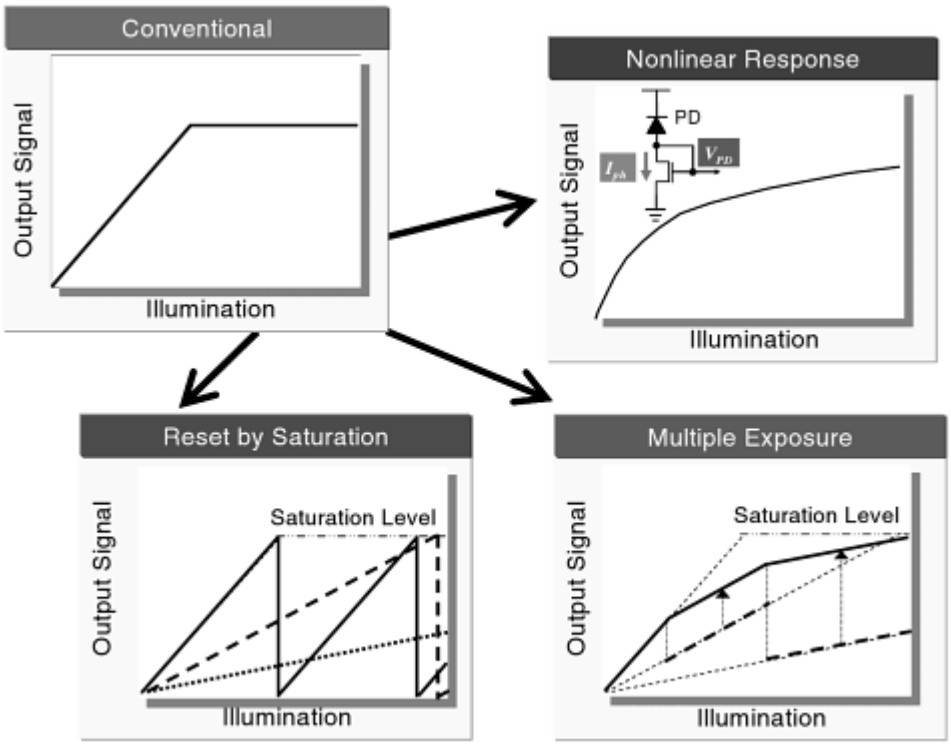


Figure 2.5: Transfer functions of different HDR methods [2.4]

Nonlinear response is usually implemented by a logarithmic response photodiode [2.7] or by well capacity adjusting method.

Well capacity adjusting is a method to control the well depth in the charge accumulation region during the integration. Figure 2.6 is an example of well capacity adjusting [2.8]. An overflow gate is used, whose gate voltage is a function represented as B(t). M3 is a charge spill gate whose gate voltage is held at a constant value of 1 V, and the photo-generated electrons in the photodiode will flow through M3 into the sense diffusion node (SD). As shown in Figure 2.6, the SD is first reset by pulling the overflow gate voltage B(t) high. The previous integrated electrons are dumped into the drain. After reset, the voltage of overflow gate abruptly raises a small amount. Photo-generated electrons begin to accumulate in the photodiode. During the integration, the overflow gate is adjusted with time to increase the potential barrier gradually. Eventually, the potential barrier will be high enough to retain all the electrons entering the well.

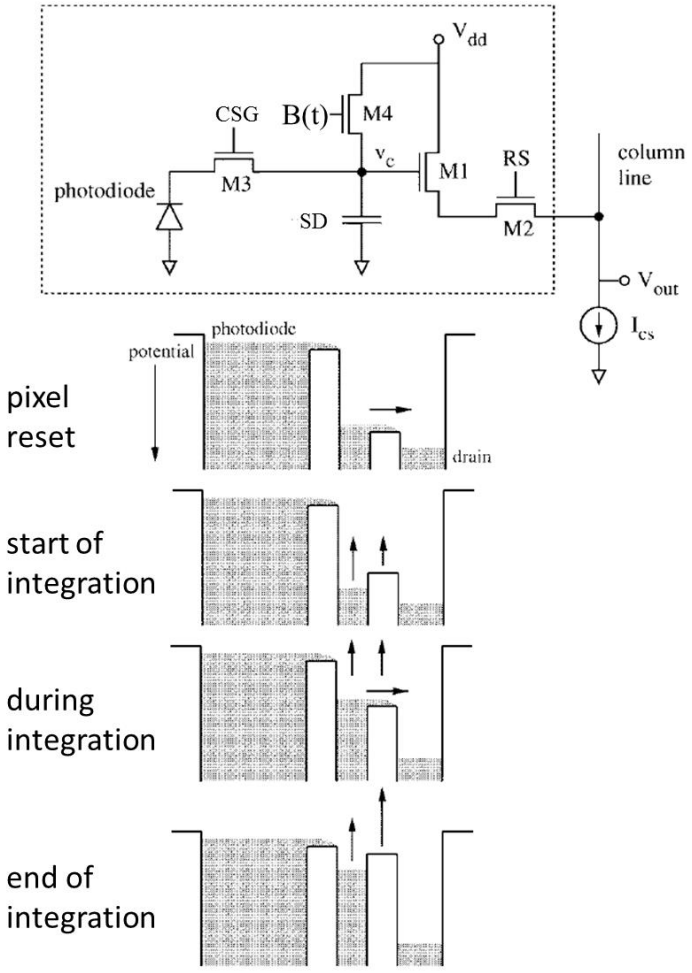


Figure 2.6: One example of well capacity adjustment [2.8]

Multiple capture, as well as addressed like multiple exposure, is a method where the signal charges are read several times [2.6]. For example [2.9], short integration time images capture high light regions and long integration time images capture low light regions. Those two kinds of images are synthesized together so that both scenes can be displayed in one image.

In the time to saturation methods, when the integration signal is observed to reach a threshold value, it is reset and the reset number is counted. By repeating that process, the final output signal is the residual charge signal and reset counting number [2.6].

Besides those three methods, another method is also popular, called dual sensitivity. It is more similar with human visual system. In a CCD sensor, it needs two types of photodiodes with different sensitivities integrated into one pixel [2.10]. Under high illumination, low sensitivity is used and under low illumination, high sensitivity is used. The same principle happened in CMOS sensor, but the job of two photodiodes in a CCD is replaced by an extra floating diffusion [2.11]. It is a direct method and no latency of capture occurs, in contrast with multiple capture method [2.6]. In this design, the dual sensitivity is achieved by one FD and one extra in-pixel capacitor.

2.3.Overview of binning methods

Nowadays, there are many additional changes to improve camera performance without adding extra product cost. Binning is one example. It can be achieved both in hardware and software. Software binning also can be called as an exposure balancing, it takes the same image at different exposure times and adds the pixel values together to calibrate the exposure. For the sensor design, hardware binning is what needs to be focused on. There are different implementations for hardware binning in CCD sensor and CMOS sensor [2.12].

2.3.1. Hardware binning in CCD sensor

In CCD sensor, binning is to bin the pixel directly before reading the pixels. One example is illustrated in Figure 2.7. If 4 pixels are read out individually, they will be associated with 4 separated readout noise events. By vertical and horizontal shifting the pixel into one register then reading the binned signal, there will be only one readout noise event. So binning in CCD can add up the photon generated electrons, as well as reduce the readout noise contribution. Using the equation (2-2), if we name the signals from 4 pixels N_{S1}, N_{S2}, N_{S3}, N_{S4}, then without binning, the SNR is

$$SNR = 20log(\frac{N_{S1,S2,S3,S4}}{N_{noise}})\ dB \tag{2-3}$$

With 2*2 binning, we get new SNR as

$$SNR = 20log(\frac{N_{S1}+N_{S2}+N_{S3}+N_{S4}}{N_{noise}})\ dB \tag{2-4}$$

Thus, we can see binning in this case results in a better SNR.

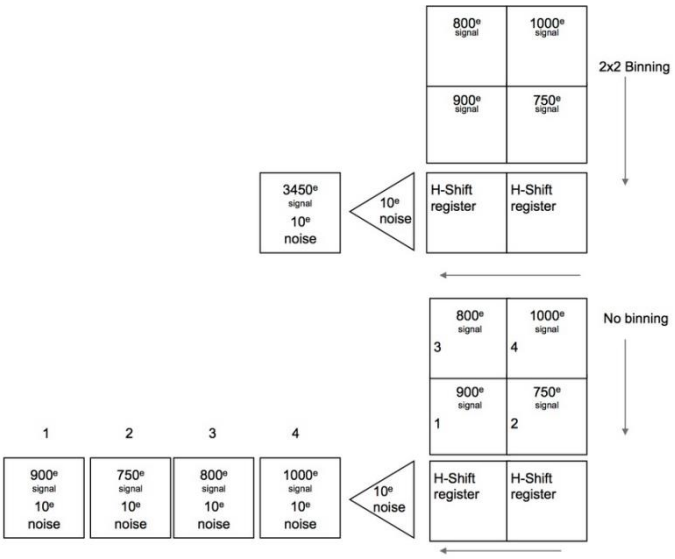


Figure 2.7: Binning of pixel in CCD sensor [2.12]

2.3.2. Hardware binning in CMOS sensor

In CMOS sensor, binning is realized by shared pixel architecture. One example is shown in Figure 2.8 [2.13].

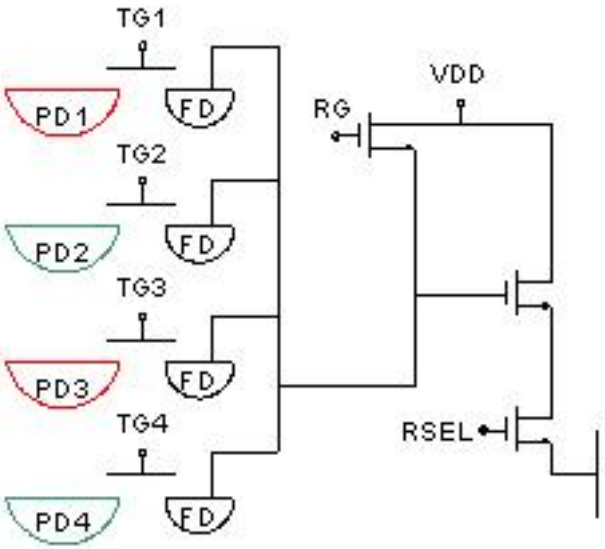


Figure 2.8: Binning in CMOS sensor with shared pixel architecture [2.13]

One shared pixel contains 4 pixels. Each of them has individual PPD, transfer gate (TG1-TG4) and FD, while they share the same reset transistor (RG), source follower and select transistor (RSEL). The charge from those 4 pixels can be measured by one single readout operation. The same concept as binning in CCD, it reduces the contribution of pixel readout noise. Moreover, it increases the frame rate, since 4 pixels only need one time readout operation.

However, as one shared pixel actually contains 4 FD, it has a conversion gain smaller than one single pixel due to the bigger FD capacitance. In this design, to keep the advantage of a higher frame rate as well as a high conversion gain in-pixel, the off-array averaging is used. Binning can be understood as adding up N signals together. Averaging, which has a similar concept, is to add up N signals and then the sum is divided by N. In this design, the signals belong to 4 adjacent pixels are averaged. Without averaging, we need to read 4 signals, while with averaging only one time readout is needed, which reduces the total number of signals so as to increase the frame rate.

In this design, user can choose mode without averaging function to keep a high resolution or averaging mode for higher frame rate. The implementation of averaging will be discussed in detail in Chapter 3.

2.4.Overview of shutter mode

Since the images are converted electronic signals from light, the exposure time or integration time influences the amount of signals. Thus, what type of shutter fits the sensor requirements to control the exposure time is an issue in design. To minimize the camera system volume, an external mechanical shutter is less used in nowadays design. The control of the integration time depends on an on-chip electronic shutter [2.14].

In CMOS image sensors, there are two mainly used shutters: global shutter and rolling shutter.

Global shutter mode

Global shutter also can be called snapshot mode. The entire pixel array is reset before exposure, and then all photodiodes accumulate charges for a same period of time. Finally, the pixels are read out row by row. Figure 2.9 illustrated the timing diagram of global shutter.

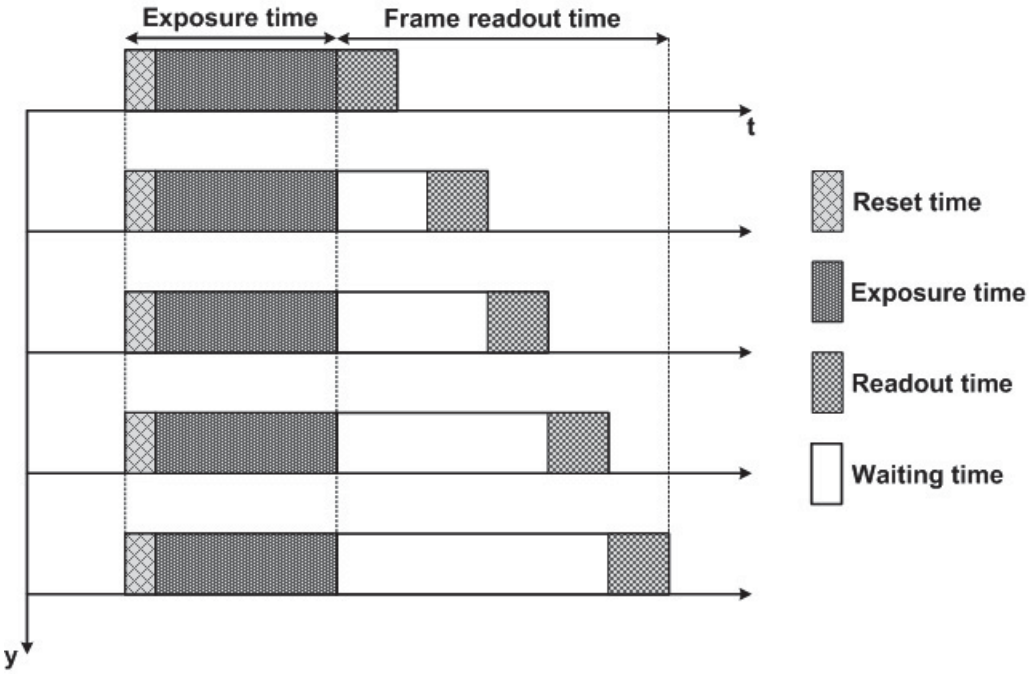


Figure 2.9: General timing diagram of global shutter [2.14]

Rolling shutter mode

Different from global shutter, in rolling shutter, the pixels do not collect information at the same time. All pixels in one row are reset, then exposed for a same period of time and read out sequentially. Those operations are completed row by row. When the first row is under exposure, the second row can start reset, which means there is a constant delay between rows, as well as the exposure time for each row is exactly the same. Figure 2.10 illustrated the timing diagram of rolling shutter.

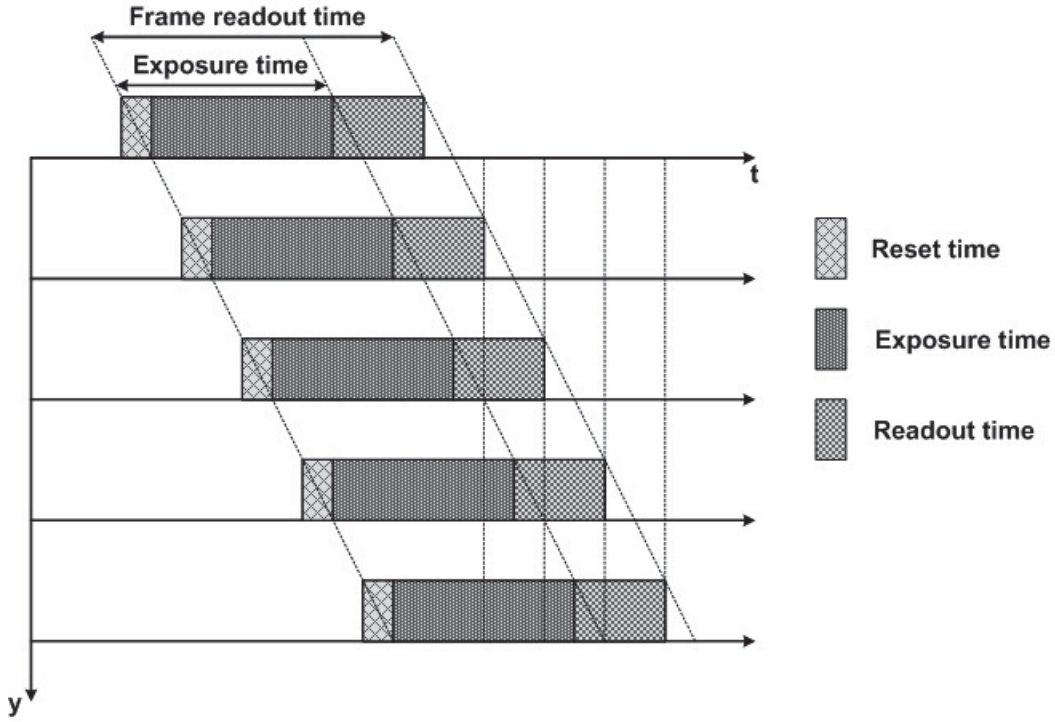


Figure 2.10: General timing diagram of rolling shutter [2.14]

The working principle of global shutter determines that the image is ''frozen'' in time. Provided the exposure time is short enough, there will be no motion blur [2.15]. Figure 2.11 is the comparison of global shutter and rolling shutter when taking a photo of a working fan. We can see clearly motion blur problem in rolling shutter, which means unless a rolling shutter is enough fast, it cannot compete with global shutter in capture image of fast moving objects.

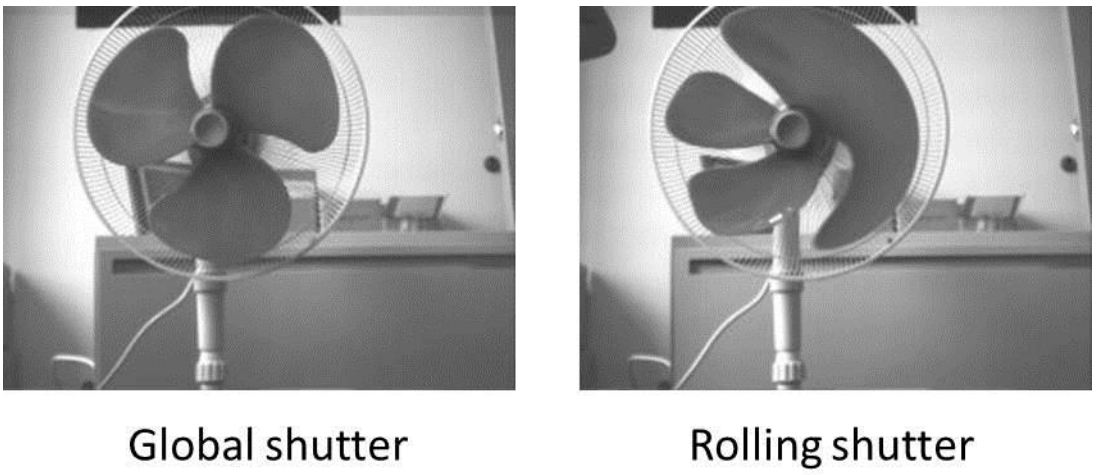


Figure 2.11: Motion blur in rolling shutter mode [2.15]

Additional, if there are some mismatches in the integration time of rows in rolling shutter, it will also generate distortion on the image.

However, global shutter is faced with peak current problems when operating a relatively large pixels array, especially doing global reset.

2.5.Chapter conclusion

In this chapter, in Section 2.1, an overview of conventional pixel structures and their working principles are introduced. In Section 2.2, the non-idealities in image sensor and the general ways to improve DR are given. In section 2.3 and 2.4, features of image sensor like binning and different shutter modes are introduced.

In the next chapter, the detailed design of sub-circuits in every block will be explained.

2.6.References

[2.1] Theuwissen, A. J. P. and Charbon, E., ET4390 Imaging sensors Lectures, Delft University of Technology, 2013.

[2.2] Takahashi, H., Noda, T., Matsuda, T. and Watanabe, T., "A 1/2.7-in 2.96 MPixel CMOS Image Sensor With Double CDS Architecture for Full High-Definition Camcorders" IEEE Journal of Solid-state Circuits, Vol. 42, No. 12, Dec. 2007.

[2.3] Takahashi, T., Ui, H., Takatori, N. and Sanada, S., "A Digital CDS Scheme on Fully Column-Inline TDC Architecture for An APS-C Format CMOS Image Sensor", 2011 Symposium on VLSI Circuits (VLSIC), pp. 90-91, Jun. 2011

[2.4] El Gamal, A., 392b Lectures and Classnotes, Stanford University, 2001.

[2.5] Dierickx, B., “High Dynamic Range the pixel standpoint,” ISSCC Forum on Image sensors, www.caeleste.be, Feb. 2008.

[2.6] Ohta, J., Smart CMOS Image Sensors and Applications, CRC Press, pp.100-104, 2007.

[2.7] Lai, L., Lai, C. and King, Y., “A Novel Logarithmic Response CMOS Image Sensor with High Output Voltage Swing and In-Pixel Fixed-Pattern Noise Reduction”, IEEE Sensors Journal, Vol. 4, No. 1, pp. 122–126, Feb. 2004.

[2.8] Decker, S., Daniel McGrath, R., Brehmer, K. and Sodini, C.G., "A 256*256 CMOS Imaging Array with Wide Dynamic Range Pixels and Column-Parallel Digital Output", IEEE Journal of Solid-state Circuits, Vol. 33, No. 12, Dec. 1998.

[2.9] Mase, M., Kawahito, S., Sasaki, M., Wakamori, Y. and Furuta, M., "A Wide Dynamic Range CMOS Image Sensor with Multiple Exposure-Time Signal Outputs and 12-bit Column-Parallel Cyclic AD Converters". IEEE Journal of Solid-state Circuits, Vol. 40, No. 12, Dec. 2005.

[2.10] Oda, K., Kobayashi, H., Takemura, K., Takeuchi, Y. and Yamda, T., "The Development of Wide Dynamic Range Image Sensor", ITE Tech. Report, Vol 27, No. 25, pp. 17–20, 2003.

[2.11] Wang, Y., Barna, S.L., Campbell, S. and Fossum, E.R., "A high dynamic range CMOS APS image sensor", In IEEE Workshop on Charge-Coupled Devices & Advanced Image Sensors, pp. 137–140, Jun. 2001.

[2.12] "Binning to increase SNR and frame rate with CCD and CMOS industrial cameras", available at http://info.adimec.com/blogposts/bid/104547/Binning-to-increase-SNR-and-frame-rate-with-CCD-and-CMOS-industrial-cameras, Jan. 2014.

[2.13] Chu, F., "Improving CMOS image sensor performance with combined pixels", available at http://www.embedded.com/design/prototyping-and-development/4013011/Improving-CMOS-image-sensor-performance-with-combined-pixels, Nov. 2005.

[2.14] Ge, X., "The Design of a Global Shutter CMOS Image Sensor in 110nm Technology", Master Thesis, Delft University of Technology, Electrical Engineering, Mathematics and Computer Science, Microelectronics & Computer Engineering, The Netherlands, 2012.

[2.15] "Key differences between rolling shutter and frame (global) shutter", available at http://www.ptgrey.com/support/kb/index.asp?a=4&q=115, Jul, 2012.

Chapter 3
Design of the sensor sub-circuits

This chapter focuses on the sub-circuits implementation in the sensor design. Since it is a stitched design, the sensor is divided into 4 blocks: pixel array block, pixel row driver block, digital control block and readout block. Their circuitry analysis, functional simulation and layout are discussed in each section.

3.1.Pixel overview

3.1.1. Pixel architecture

The work we do is based on an image sensor using the pixel architecture shown in Figure 3.1.

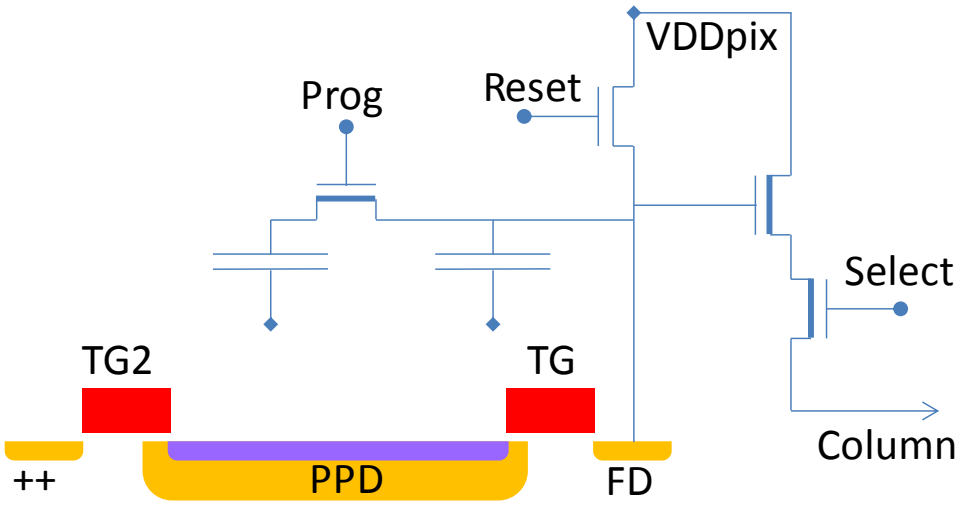


Figure 3.1: Pixel architecture

The pixel is based on a 4T pinned photodiode (PPD) pixel, and it allows simultaneous acquisition of dual sensitivity regions with the same integration time.

The second range is obtained using an extra MIM capacitor C_{MIM} to increase the full well charge (Q_{FW}). The photo charges are first accumulated in the photodiode and read out via a floating diffusion node (FD) when operating the transfer gate TG. This is what we call the high gain (HG) measurement range [3.1].

Photo charges that do not fit in the PPD will flow into the C_{MIM} over a potential barrier, in series with a transistor called SERIES. Those overflow charges can be read out when we close the transistor MERGE, which is called low gain (LG) measurement range [3.1].

A specific implant on the source follower and Select (SEL) transistors results in a lower V_{th}, which provides a larger signal range. The buried channel moves the charge away from the interface to reduce the 1/f noise.

The second transfer gate TG2 is used to adjust the integration time from a few μs to the entire frame time. Moreover, one can operate TG2 to let part of the charge in the PPD flow into the VPIX to realize an anti- blooming function.

3.1.2. Dynamic range calculation

The tentative specifications of this pixel are listed in the following Table 3-1.

Table 3-1: Pixel specification

Mode	Pixel size (μm)	Charge conversion (μV/e^-)	Readout noise (e^-)	Q_{FW} (e^-)
High Gain	6.5*6.5	100	5	10k
Low Gain		4	50	200k

The minimum number of detectable electrons depends on readout noise and dark current. The dark current value based on Caeleste's experimental results with the technology is about 100 e-/s in room temperature (21 ℃). Since we implement a cooling system to the sensor, which effectively reduces dark current by a factor of 2 every 7 ℃, so we can assume that under the real usage condition (-40℃), the dark current is almost negligible.

Based on the equation (2-1) in Chapter 2, we can assume the tentative dynamic range of our pixel is

$$DR = 20log(\frac{2\times10^5}{5}) = 92\ dB \quad (3\text{-}1)$$

3.1.3. Pixel kernel for readout

To increase the readout speed, we read out a kernel of 2*8 pixels at the same time, and equally split the pixel array to be read out from south (S) and north (N) in parallel [3.2], which is illustrated in Figure 3.2.

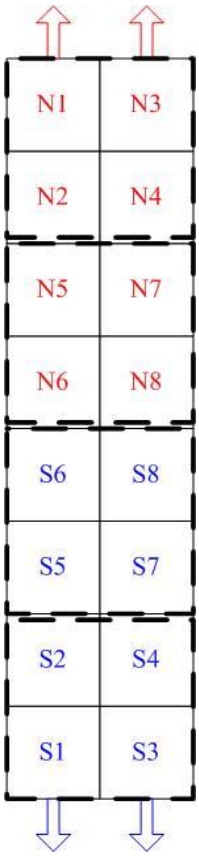


Figure 3.2: Pixel readout order for a kernel of 2*8 pixels

The four adjacent pixels highlighted by the black bold dashed line in Figure 3.2 are the 4 pixels whose output signals are averaged to increase the frame rate in averaging mode, which will be explained in Section 3.4.4.

3.1.4. Pixel timing diagram and standalone simulation

The timing diagrams of the pixel in different modes are illustrated in Figure 3.3, Figure 3.4 and Figure 3.5.

Since we read out a kernel of 2*8 pixels at one time, we define 8 adjacent rows of pixels to be one kernel row. Figure 3.3 shows the timing diagram in rolling shutter mode for one kernel row. In rolling shutter mode, one does CDS on chip. The operations of pixels in rolling shutter mode are performed kernel row by kernel row. One “pass” here refers to one time readout from pixel to output bond pad. The readout of every kernel row from one HDR frame needs two passes. The first pass is to read out the high gain signal, and the second is the low gain signal. We call the readout time of one kernel row “row time”.

In every pixel operation cycle, at first the FD and MIM capacitor are both reset when Merge switch is on. After reset, Select transistor is on and the reset value on the FD will be sent out. Then the transfer gate (TG) is on to transfer the electrons from the PPD to the FD. Another Select pulse follows to read out the HG signal on the FD. With Merge switch closed again, the third Select pulse is to read out the LG signal on the MIM capacitor.

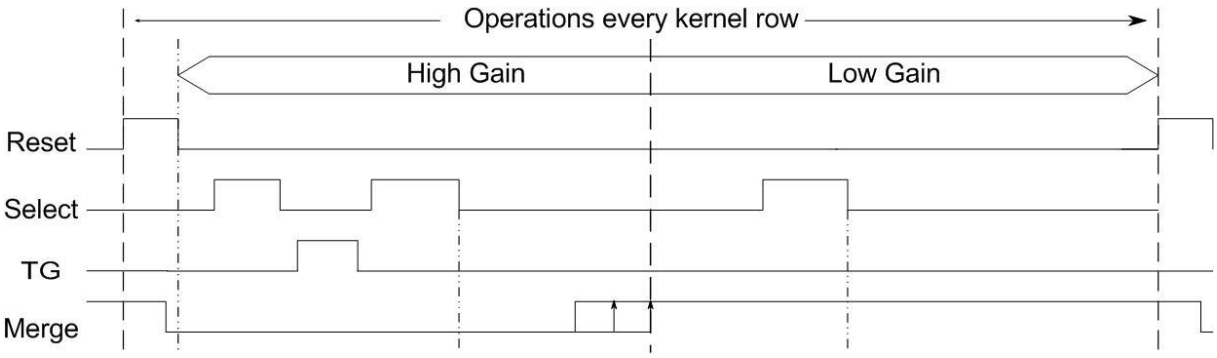


Figure 3.3: Pixel timing diagram in rolling shutter mode

Figure 3.4 shows the timing diagram in snapshot HDR mode. In snapshot mode, one does CDS off-chip. The operations of pixel are performed frame by frame. Only the Select signal is still operated every kernel row since it is the control signal for pixel readout.

A snapshot HDR frame needs three passes. First is to read out the pixel reset signals, then the high gain signals, and finally the low gain signal.

In every pixel operation cycle, first a global reset is performed. Since a current limiter is implemented to reduce the peak current when doing global reset, the time of reset in snapshot mode will be longer (about 50 μs). The purpose and detail of current limiter will be discussed further in this chapter. Then the Select transistors are on kernel row by kernel row to read out the reset signal. In Figure 3.4, 4K configuration is taken as an example, so we have 512 kernel rows. After the reset value is read out, transfer gate (TG) is on to transfer the electrons from the PPD to the FD. Then the Select transistors complete the HG signals readout. After the Merge switch is on, the readout of the LG signals is realized by the third time Select pulses.

In HDR mode, while reading out the signal voltage from the FD, we can stop the integration in the photodiode by turning TG2 on. If not, the photodiode would keep on integrating and charges could possibly overflow the barrier into the series capacitance C_{MIM}, merging 2 frames together. We can see from Figure 3.4, the integration time is defined as the time interval between the falling edge of TG2 pulse and TG pulse.

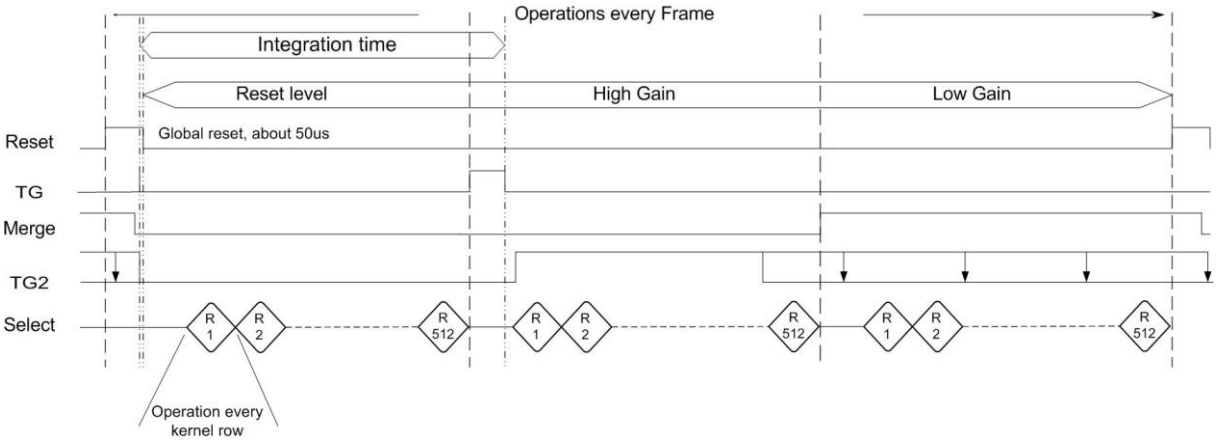


Figure 3.4: Pixel timing diagram in snapshot HDR mode

Figure 3.5 shows the timing diagram in snapshot a low dynamic range (LDR) mode. The pixel operation is similar to the one in HDR mode. For a LDR frame, we only need two passes: for the reset signal and the high gain signal. Since a PPD is a natural pipelined device, we can immediately start integrating a new frame after extracting the data of the previous frame.

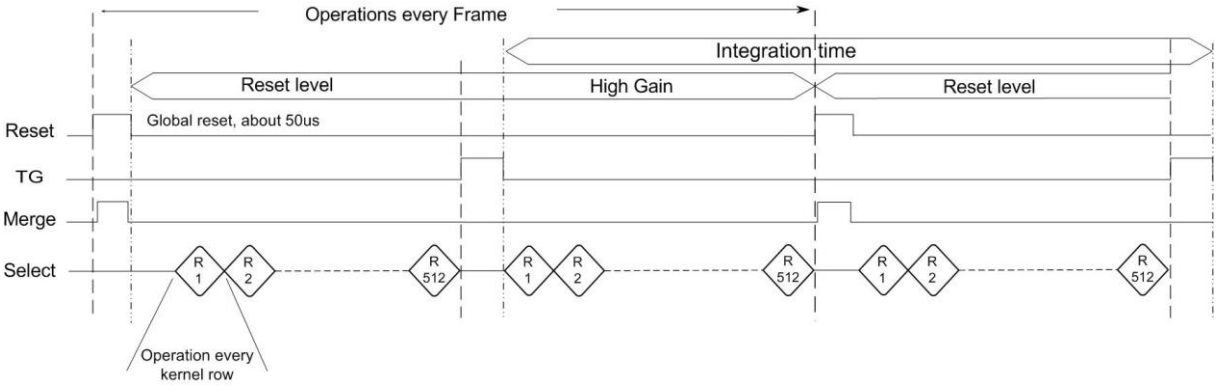


Figure 3.5: Pixel timing diagram in snapshot LDR mode

The pixel model we used in our simulation is shown in Figure 3.6.

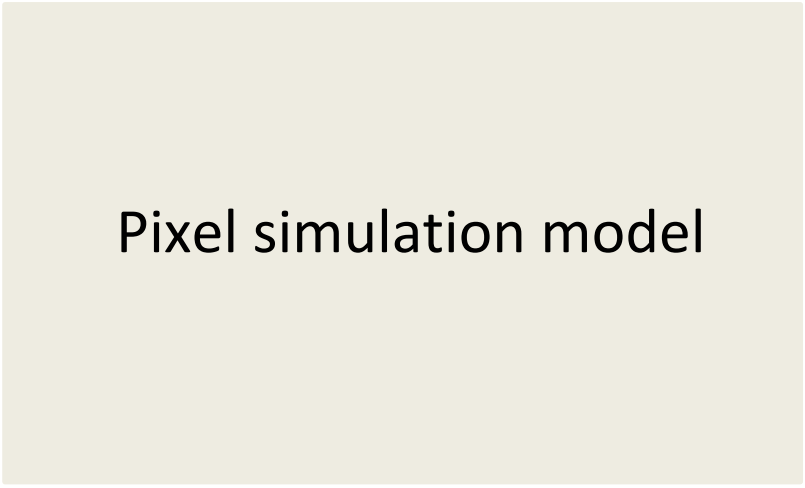


Figure 3.6: Pixel model schematic for simulation

A small capacitor C_{FD} is added to FD node, which together with the transistor parasitic capacitance models the real floating diffusion capacitance. We use a current pulse to model the photodiode charge packet transferred by TG.

Since we have a large pixel array, the parasitic resistance on the column bus for pixel readout varies based on the pixel position in the array. So when we check the pixel function, we also take the closest and the farthest pixel as examples. Figure 3.7 are the simulation results for both HG and LG mode.

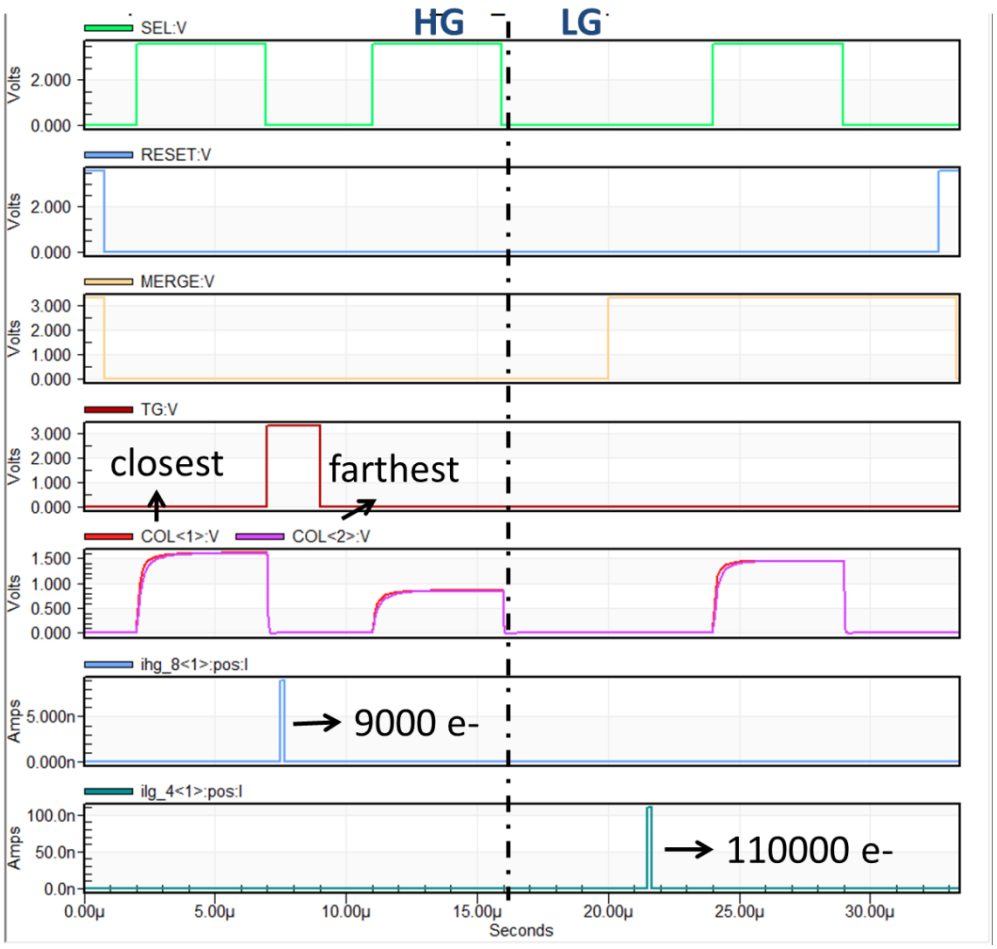


Figure 3.7: Pixel simulation with different wire parasitic

In Figure 3.7:

- The first 4 waveforms are the pixel operation signals: Select, Reset, Merge and TG.
- The traces in the fifth waveform are the outputs COL<1> and COL<2> of two pixels in one kernel. The operations on those two pixels are the same, the only difference between them is the output COL<1> connects to a smaller parasitic resistor, and another one COL <2> connects to a bigger parasitic resistor. The resistor value is proportional to the length of the metal wire measured from the layout.
- The last two waveforms are the current pulses which model the photocurrent. In HG measurement, here we give 9000 e- while in LG measurement, 110000 e- is taken as an example.

From the simulation result, we can see with a Select pulse width wider than 4 µs, the output of the pixels have enough time to settle, and the influence of the different parasitic resistance is negligible.

3.1.5. Pixel Layout

Figure 3.8 is the layout of the pixel. The size of one pixel is 6.5 µm *6.5 µm. The MIM capacitor using metal5 layer (M5) and top metal layer (topM) is on top of the pixel. This is allowed as the device is intended to be backside thinned.

Figure 3.8: Pixel layout

3.1.6. Pixel power supply concern

A huge peak current will flow through the VPIX supply line when we do global reset. Meanwhile, due to the large parasitic capacitance of the column load wires, it also results in a significant peak current when we select one pixel to send the reset value on a column, which refers to “first select” in Table 3-2.

We do a simulation with one kernel of pixels to check the peak current. Table 3-2 shows the results and the estimated total peak current in one time for the whole sensor. According to the design plan, global shutter mode is only applied in 4K configuration, but rolling shutter for all three configurations, the worst case is listed as follow.

Table 3-2: Peak current on VPIX in different situation

Situation	I_{peak} on one kernel of pixels	I_{peak} on VPIX
Global reset (4K configuration)	2 µA	2A
First select (12K configuration)	1.25 mA	7.6A

To avoid the huge peak currents and the risk of any damage due to electro-migration or melting the bond wires, a current limiter is implemented. The input of the current limiter is the master VPIX from the bond pad, and then it is split into several parallel local VPIXs connected to the pixel array, which can be current-tuned by the user. The location of the current limiter is at the north and south of the sensor, beginning in the readout block. The circuit of current limiter in detail will be discussed in Section 3.4.2.

The final routing of the power supply of the pixel array is briefly illustrated in Figure 3.9. We add extra pads for VPIX at the bottom and top of the sensor to avoid gradients in the middle of the sensor, especially in 12K configuration.

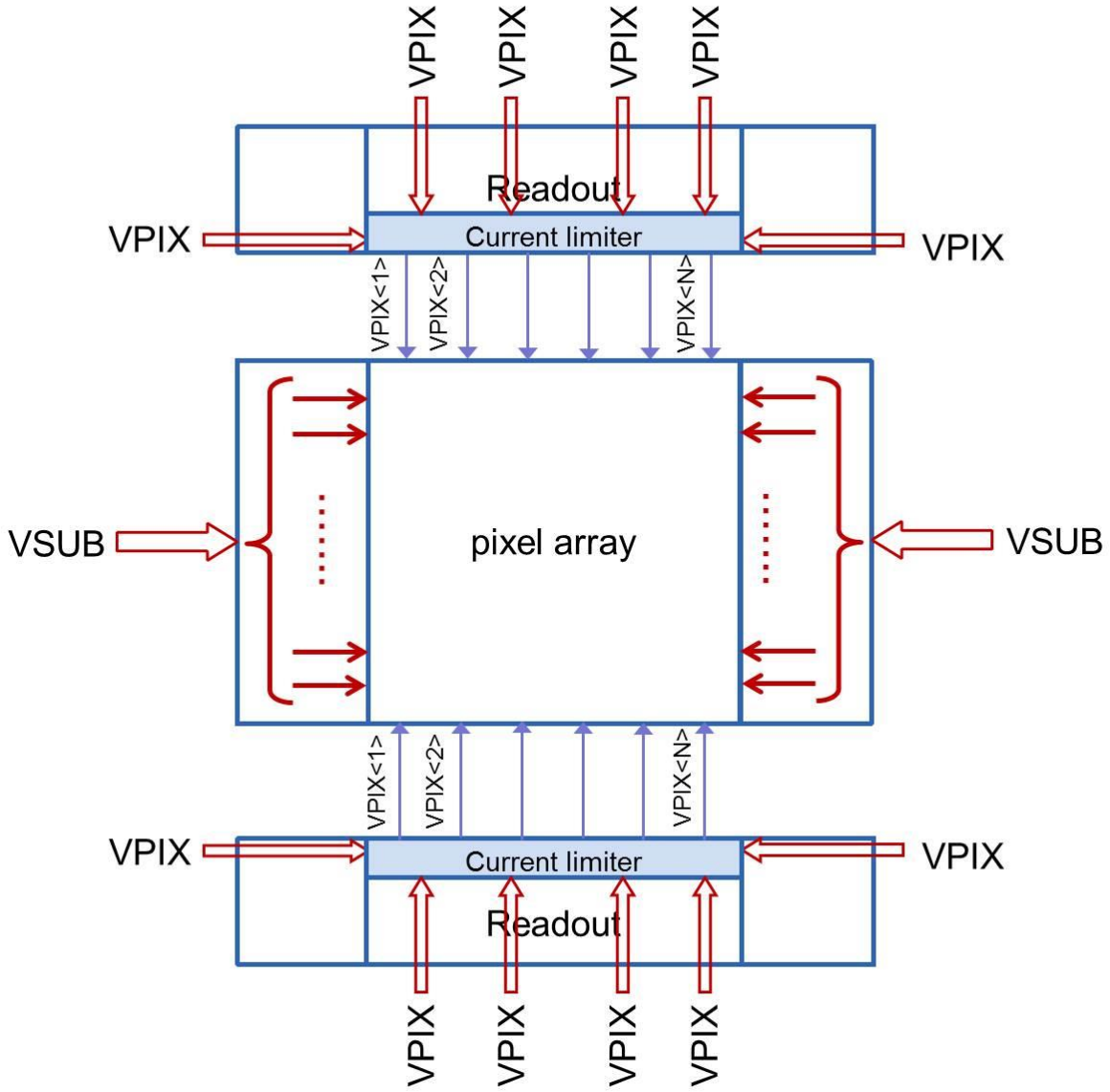


Figure 3.9: The arrangement of the power supplies for the pixel array

3.2.Pixel row driver

3.2.1. Pixel row driver overview

Pixel row driver is needed to realize the Y-axis scanning of pixels and provide all the operation signals. The overview of the pixel driver block is shown in Figure 3.10.

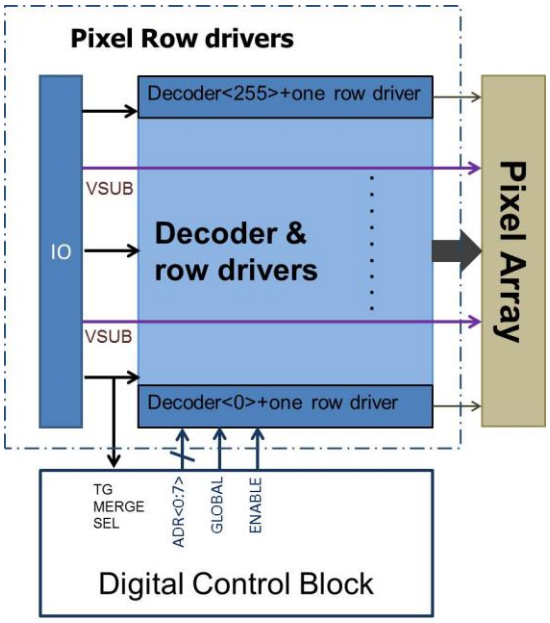


Figure 3.10: Overview of the pixel row driver block

It consists of a decoder for the row addressing (Y-axis scanning), logic gates to achieve the right pixel operational function, level shifter to isolation digital power supply with analog ones and drivers to drive the transistors in pixel array.

According to Figure 3.10, the address code and some other signals are sent from the addressable serial parallel interface (ASPI) in the corner digital control block. And also the pixel ground power supply VSUB is from the bond pad at the side of the row driver block, routing through the block to the pixel array.

In this sensor, both a global shutter mode and a rolling shutter mode are available, so our pixel row driver is able to be switched between those two different modes.

3.2.2. Row addressing

We aim to read out pixel kernel by kernel (one kernel = 2 columns*8 rows) at one time, so in rolling shutter mode, we address 8 rows of pixels simultaneously. 8 adjacent rows of pixels are defined as one kernel row. Therefore, every 8 rows of pixels share the same address. Since in one pixel row driver block (driving 2048 rows of pixels), we have 256 kernels of drivers, an 8 bits decoder is required in this design. Moreover, as it is a stitched design, different numbers of the pixel row driver block will be implemented based on which configuration (2 blocks in 4K, 4 blocks in 8K and 6 blocks in 12K), but they indeed share the same address lines. Thus, we add an ENABLE signal to the decoder, to enable or disable them to meet the usage in different configurations.

We implement the decoder in a NAND gate way [3.3], which is shown in Figure 3.11.

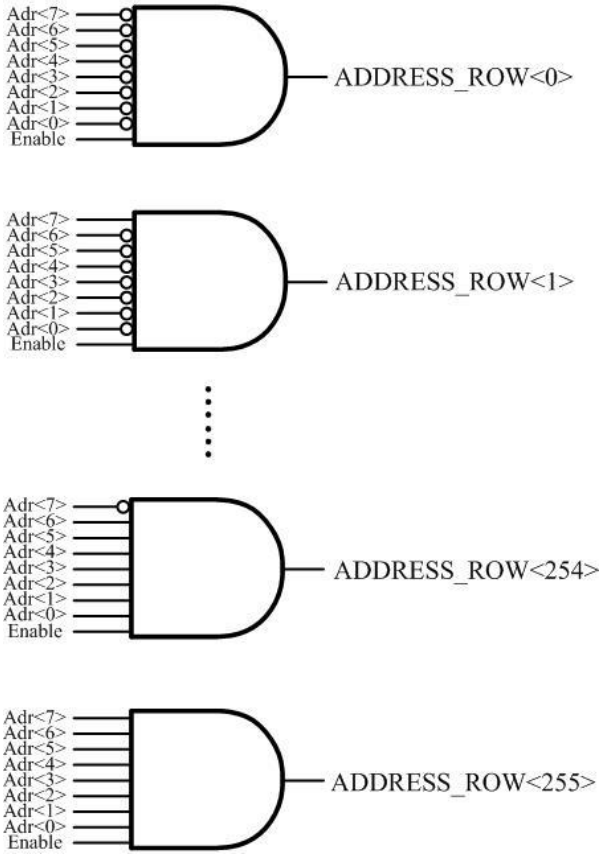


Figure 3.11: Structure of the 8-bit decoder

The 8 Adr<0:7>, 8 inverted Adr<0:8> and Enable signals are provided from the ASPI.

3.2.3. Pixel driver

As discussed in the Section 3.1, one pixel contains 6 control transistors: TG, MERGE, RESET, SEL, SERIES and TG2. Each of them needs individual logic control to be functional. For TG, MERGE, RESET, SERIES and TG2, they should be activated by either the address code in rolling shutter mode or the global code in snapshot mode. For the SEL signal, as it works for pixel readout, it should only follow the address code. The default value (when both address and global is 0) for RESET and SERIES is high, for the other control signals is low. It means we always keep the FD being reset and the extra MIM accumulating charges when the pixel is not accessed.

So the driver schematic is designed as Figure 3.12.

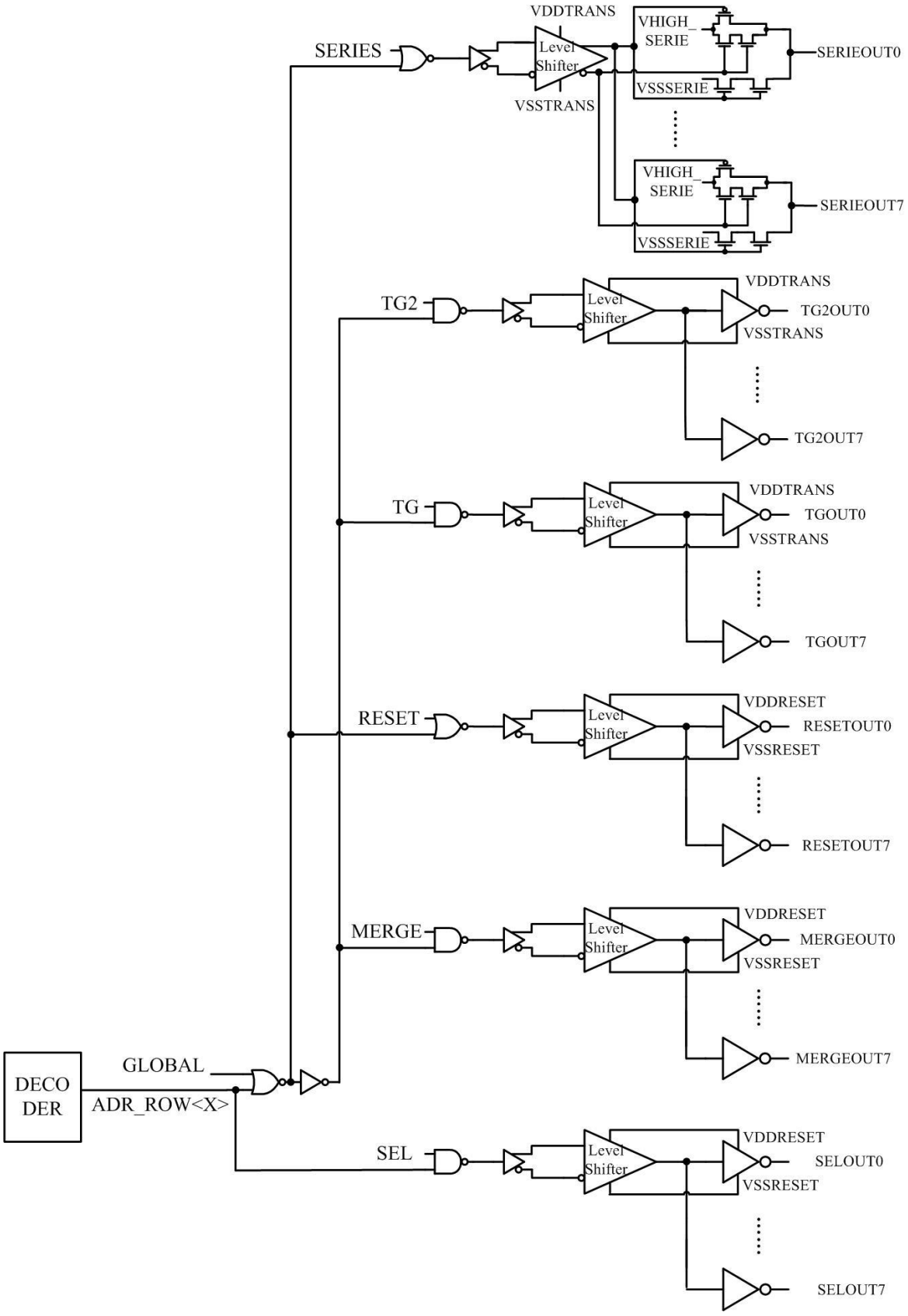


Figure 3.12: Schematic of row driver unit

Because we operate pixels kernel by kernel, so every 8 rows of pixels share one such row driver unit.

The inverters act as buffers to drive the transistors in the pixel row. We use clean analog power supply VDDRESET and VDDTRANS for the buffers and use level shifters to isolate the different kinds of power supplies between the logic gates and the inverter buffers. Moreover, to avoid the shorts happening on one row might influence another row, we have one independent buffer per row, yet 8 buffers are controlled by the same logic gates in one row driver unit.

Also, SERIES and TG2 need specific buffers to achieve their function.

For the SERIES switch in the pixel, as it is used to control a barrier which lets excess charges overflow into the MIM capacitor, the high level of SERIES output varies from 0.7V to 3.3V. Thus, we use a CMOS transfer gate to pass a voltage called Vhigh_series instead of using an inverter as a buffer. The value for Vhigh_series is powered via a bond pad.

For the TG2 switches, concerning the anti-blooming feature, the lower value should be equal to or higher than VSS (e.g. 0.2 V). So we separate the substrate and source of the NMOS in the TG2 inverter, and using a dedicated VSS, called VSSTG2, which is also powered via a bond pad.

Figure 3.13 is the functional simulation of the pixel row driver.

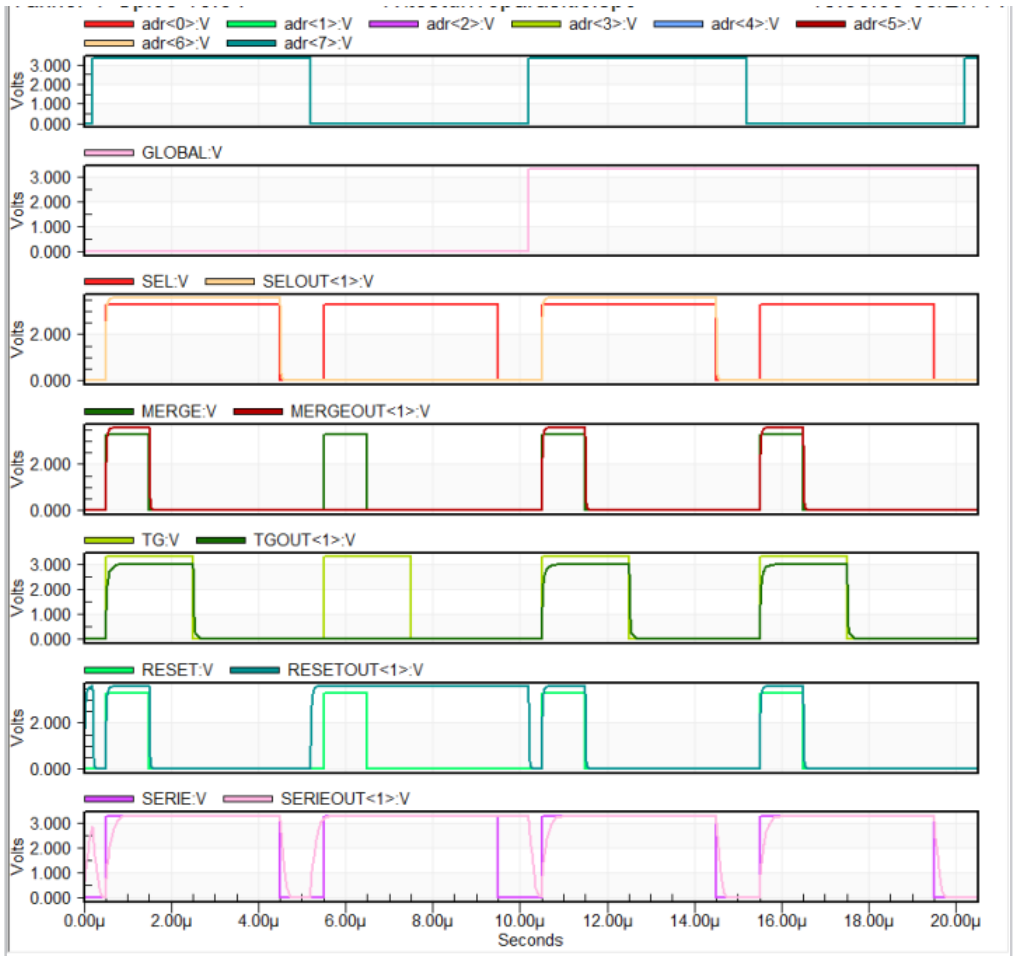


Figure 3.13: Functional simulation of pixel row driver

- In Figure 3.13, the traces in the first waveform are 8 address codes.
- The second waveform is the Global signal.

- The rest 5 waveforms are the input of pixel control signals (SEL, MERGE, TG, RESET, SERIES) and corresponding output of the control signal from pixel row drivers (e.g. SEL is the input and SELOUT<1> is the corresponding output).

We choose 4 different working condition of the pixel row driver (Address & Global=00, 01, 10, 11), to show the validity of pixel row driver. We can see that SEL signal always follow 'address', while other control signal follow 'address' or 'global'. Moreover, when one kernel row is not selected (Address & Global=00), the default value of Reset and Series output is high, which meets the requirement of the function.

Figure 3.14 shows the simulation of the driving capability of the buffers. In the test bench, we model the large parasitic resistor and capacitor (RC) from the long and thin metal wires in the pixel array, especially in 12K configuration, and those control signals pulse width is limited by the row time.

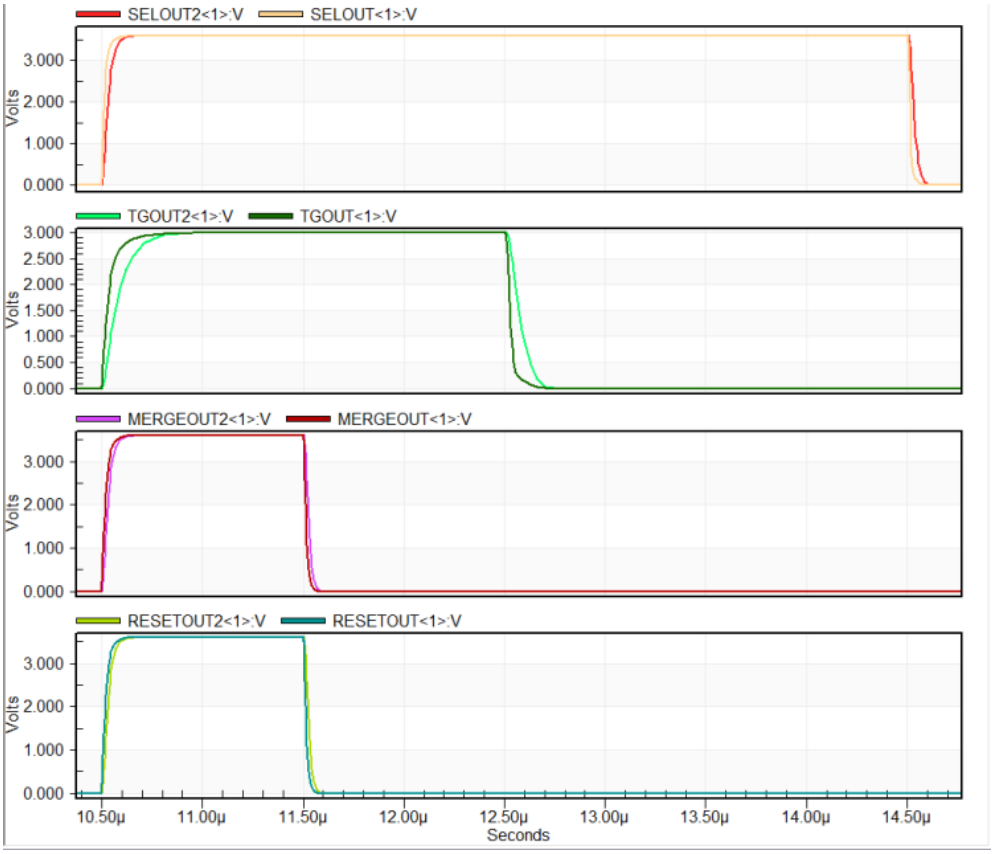


Figure 3.14: Simulation of driving ability with parasitic RC

In Figure 3.14, the outputs of the pixel row driver of four fast operation signals (SEL, TG, MERGE, RESET) are shown. Each waveform contains two outputs, one is the closest to the pixel row driver (at the edge of the pixel array), another is the farthest to the pixel row driver (in the middle of the pixel array). Take the first two traces as an example, the trace SELOUT in pink is the closest and the trace SELOUT2 in red is the farthest. The farthest outputs are connected to larger parasitic resistors and capacitors which are calculated from the length of metal wires.

From the simulation result, we can see the difference between the closest and farthest output signals are negligible, which prove the driving ability of the pixel row driver block.

3.2.4. Layout consideration

In the layout, most space limited is the buffers to send control signal to the pixels since our pixel size is only 6.5 um. The layout of one unit row driver is shown in Figure 3.15. The buffers for one pixel row is exactly 6.5 um height and duplicated 8 times in one driver unit, the other MOSFETs are relaxed in space. That driver unit is duplicated 256 times in one pixel row driver block. The metal wires at the left part in Figure 3.15 are the control signal buses and address buses.

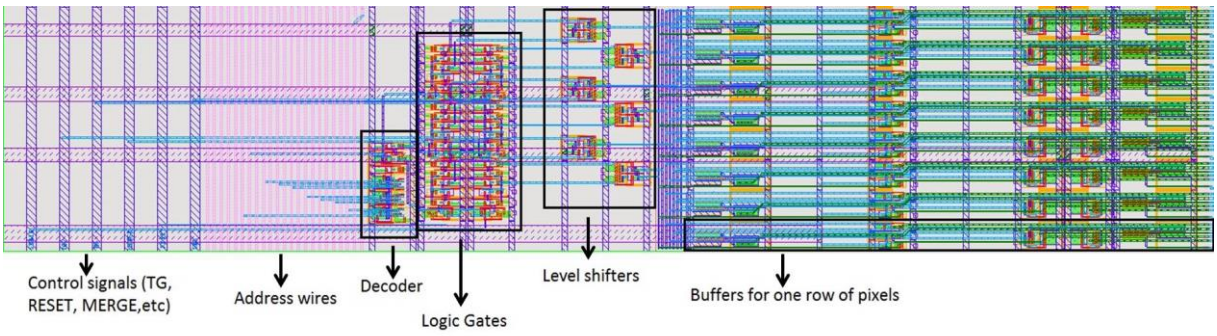


Figure 3.15: Layout of row driver unit

Since we need large inverters in the pixel driver, it will consequentially generate a substantial amount of peak currents in global mode when all the inverters work at the same time. We run the simulation to estimate the peak current on VDDTRANS and VDDRESET in 4K configuration, the result is shown in Table 3-3.

Table 3-3: Peak current on VDDRESET/VDDTRANS in different situation

	I_{peak} (A)
VDDRESET/VSSRESET	1.7
VDDTRANS/VSSTRANS	2.1

To reduce the parasitic resistance on power supply wires, as well as to avoid the peak current to burn the pad, we give more than one pad for one power supply as there is quite a lot space in the pixel row driver block. Meanwhile we make the power supplies as wide as possible, especially VDDTRANS and VDDRESET, the estimated parasitic resistance from one bond pad to inside is less than 2 ohm.

The overview of the layout of the pixel driver block is shown in Figure 3.16. We take the pixel row driver block in the west of the sensor as an example.

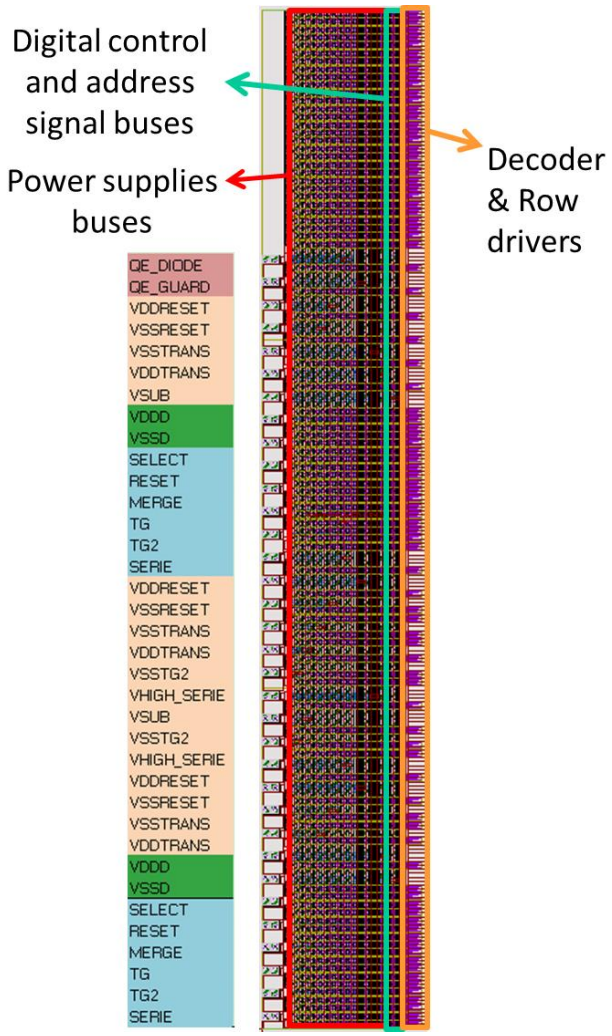


Figure 3.16: Layout of pixel row driver block

The right part in Figure 3.16 is the layout of the whole pixel row driver block. We can see that the power supplies buses cover the most area in the layout to reduce the parasitic resistance as much as possible. The left part in Figure 3.16 illustrates the IO position in the pixel row driver block. The analog power supplies VDDTRANS/VSSTRANS, VDDRESET/VSSRESET, VSSTG2, VHIGH_SERIES and VSUB are labelled in pink. Digital power supplies VDDD/VSSD are in green. Blue is the digital control signals, such as TG, RESET, MERGE, etc. Two brown ones are pads for extra quantum efficiency test structure, which will be introduced in Chapter 4.

3.3.Digital Control Block

3.3.1. Digital control block overview

The digital control block provides various logic control signals to different parts of the sensor, and it is located at every corner of the sensor. Figure 3.17 shows an overview of the digital control block.

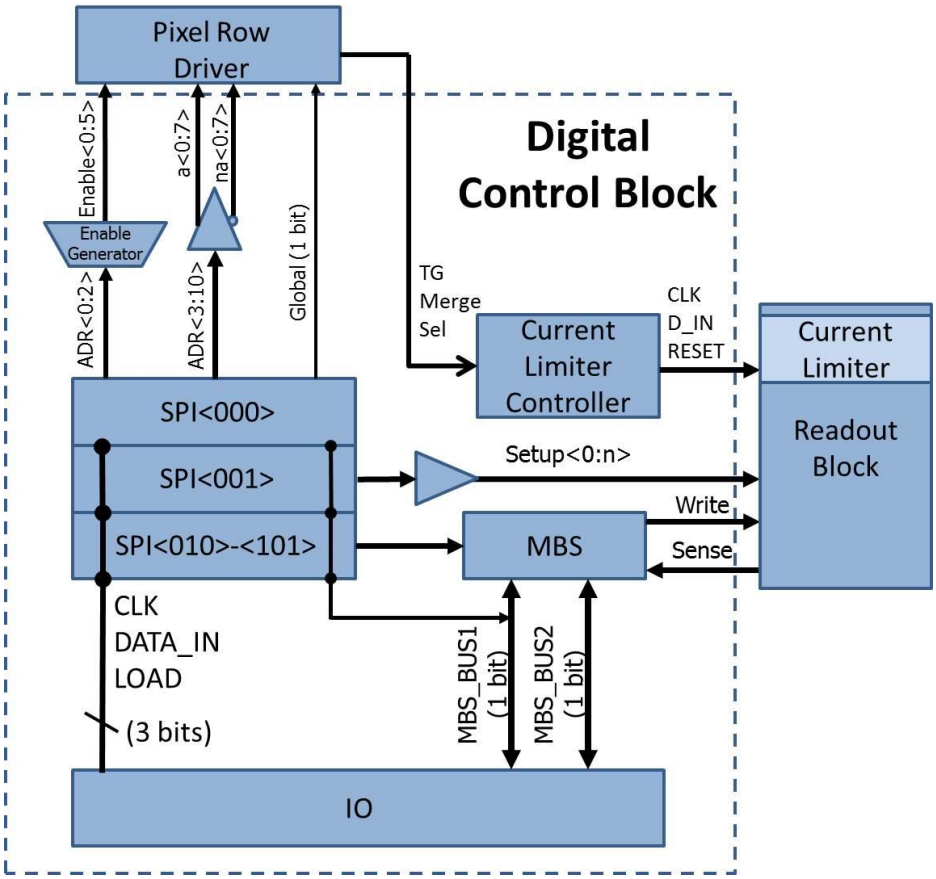


Figure 3.17: Overview of the digital control block

It mainly contains:

The ASPI (Addressable Serial Parallel Interface) for sending digital codes to different parts of the sensor;

The MBS (Mixed Boundary Scan) which is used to sense and write some important nodes in the sensor for testing consideration;

A small circuit named “current limiter controller” utilizing 3 signals (TG, MERGE and SEL) from the pixel driver block to generate signals to control the current limiter.

3.3.2. ASPI

There are 2 common ways to implement a SPI [3.4]:

- An architecture based on a long shift register chain.
- An architecture called ASPI (addressable SPI).

The drawback of a SPI consisting of a single long shift register is that when the SPI has a lot of data to send, and the users only want to change few of them, they still need to clock through the whole register chain. It is time-consuming and not efficient.

The ASPI is based on a set of short SPI registers, all individually addressable. This system allows changing only a part of the register data without affecting the other shift registers.

In the design, 6 ASPI registers are in use:

- One ASPI whose address is <000> provides the row address code and the enable signal to pixel row driver, and define the snapshot mode or rolling shutter mode;
- One ASPI whose address is <001> provides signals for a general setup of the sensor, e.g. averaging mode or non-averaging mode;
- Four ASPIs whose address are from <010> to <101> provide the setup for the MBS sense and drive function.

Each ASPI has 12 bits of data. And to address these 6 ASPIs, 3-bit address code is needed.

The schematic of one ASPI unit is shown in Figure 3.18.

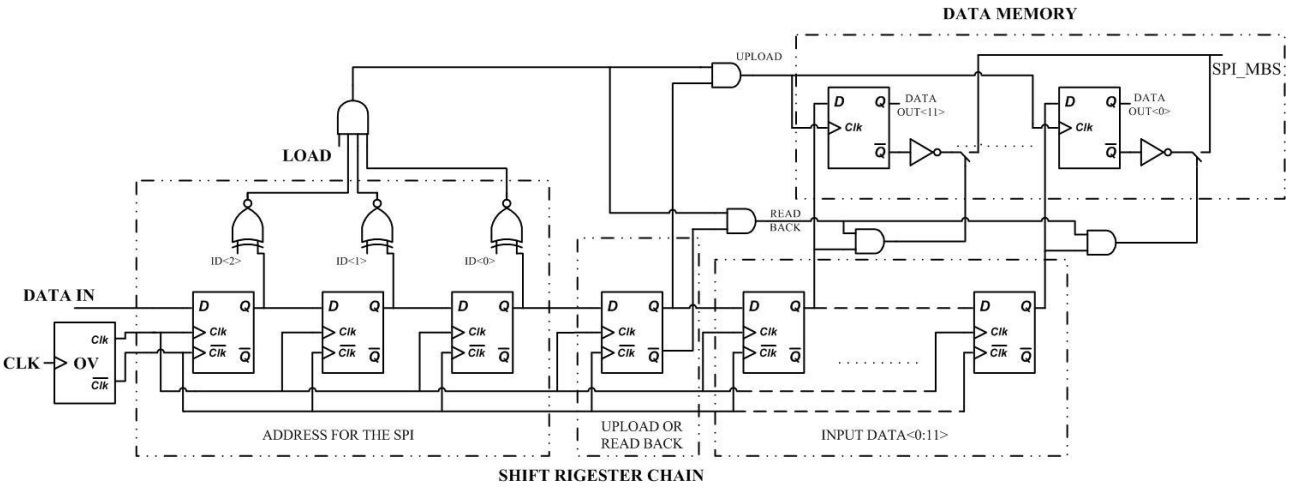


Figure 3.18: Schematic of ASPI

It uses three control signals:

- SPI_CLK: it provides the clock for the shift register.
- SPI_DATA_IN: it sends the input data for the ASPI in serial.
- SPI_LOAD: it is the control signal which makes the input data loaded into the memory register in ASPI and then they will be sent into the corresponding place in the sensor.

There is also an output pin called SPI_MBS, which is used for read-back the signals in the memory registers. It is important to make sure that the DFFs in the ASPI are still alive under heavy radiation.

The bit arrangement of the ASPI is shown in Table 3-4.

Table 3-4: ASPI content

Bit order in time	15	14	13	12	11	10		1	0
Content	adr<2>	adr<1>	adr<0>	1=Upload 0= Read back	data<11>	data<10>	……	data<1>	data<0>

3 DFFs are used to form the address control, 1 DFF is for switch between data load mode or readback mode and 12 DFFs are for data sending.

The timing diagram of ASPI is shown in Figure 3.19.

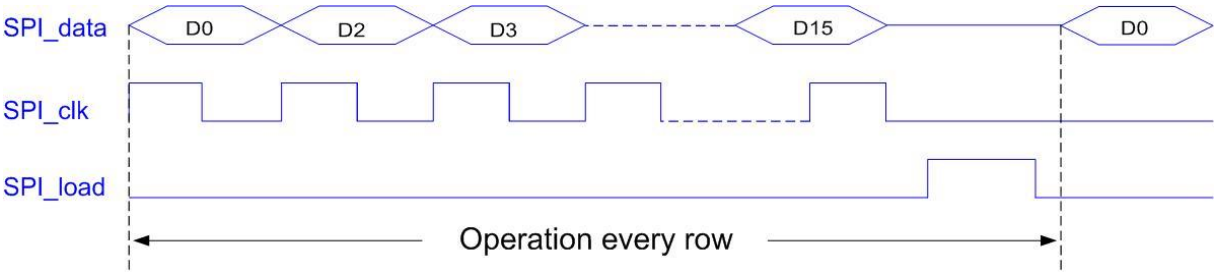


Figure 3.19: Timing diagram of the ASPI

In one transmission cycle, 16 clock cycles are need to send the data into SPI, then with a SPI_load pulse, the data is stored in the meory register and sent to the corresponding part of the sensor.

3.3.3. MBS

MBS is a test and diagnostic tool. It can “sense” and “write” nodes in the sensor.

“MBS sense” will allow us to access important nodes in the sensor, especially in the readout circuits, when the sensor does not give us the presumed results. “MBS write” provides us the possibility that we impulse a value on some nodes in the sensor to check that a specified part in the sensor has the correct function.

The schematics of MBS sense and MBS write are shown in Figure 3.20 and Figure 3.21. Figure 3.22 shows that one node both can be written or sensed.

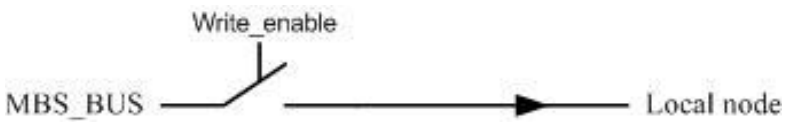


Figure 3.20: Schematic of MBS write

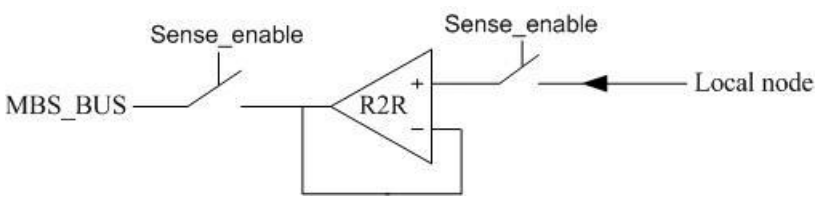


Figure 3.21: Schematic of MBS sense

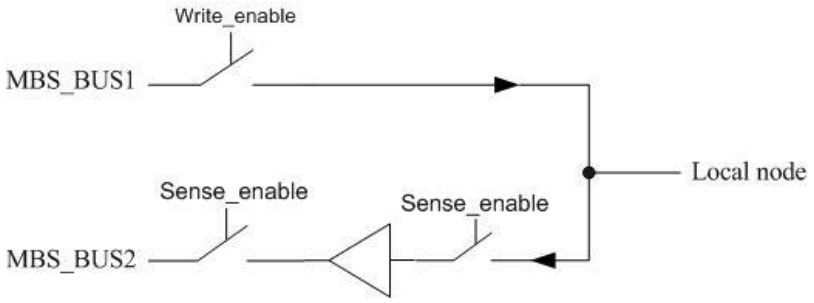


Figure 3.22: Schematic of MBS sense and write connected to the same node

The Write_enable and Sense_enable controls are provided by the ASPI.

3.3.4. Enable generator

It is a small decoder which using the same principle as decoder in the row driver. With 3 bits from the ASPI, it generates 6 individual ENABLE signals to select the pixel row driver blocks. In rolling shutter mode, we need only one row driver block working at one time.

3.3.5. Current limiter controller

The current limiter in the readout block needs four signals to work, input data (data_in), synchronized reset (sync), clock signal (clock) and inverted clock signal (clock_inv). Since the setup of the current limiter only happened at the beginning of the sensor initialization, it does not need a very fast signal to drive. As well as to save some bond pads, we take three signals from pixel row driver block, which are TG, MERGE, SEL. TG is used to generate the reset signal of the DFFs in the current limiter, MERGE is used to generate data_in signal of the current limiter and SEL is used to generate clock for the DFFs. The schematic of the current limiter controller is shown in Figure 3.23.

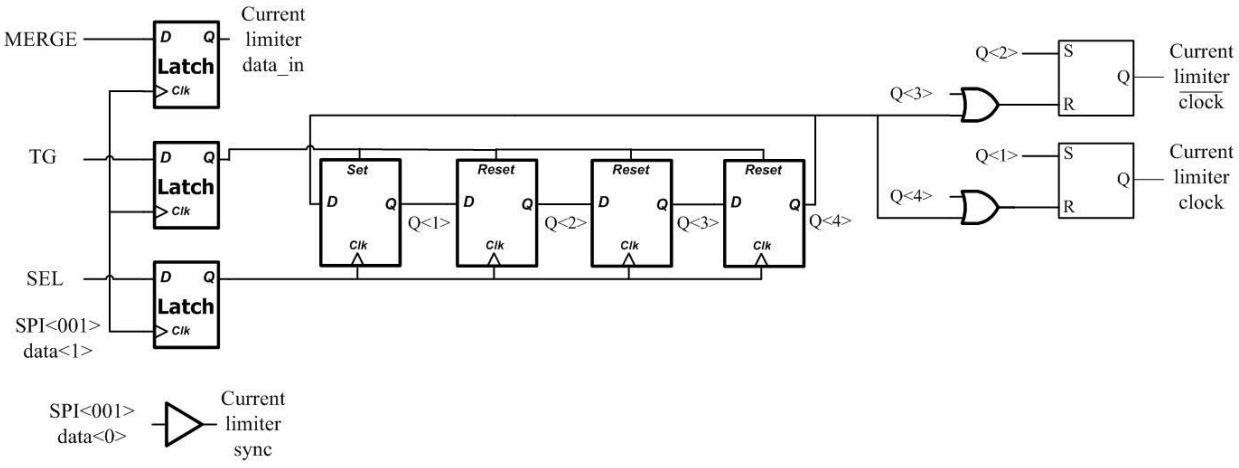


Figure 3.23: Schematic of current limiter controller

3.3.6. Functional simulation

Since there are thousands of ASPI code combination, we take one sample in ASPI upload mode and one sample in readback mode. The simulation results are shown in Figure 3.24 and Figure 3.25 respectively.

In the simulation for upload mode, ASPI<000> sends address, enable and global signals to the pixel row driver block. The SPI content for pixel row driver is shown in Table 3-5.

Table 3-5: ASPI content for control pixel row driver block

Bit order in time	15	14	13	12	11	10	.....	3	2	1	0
Content	SPI_adr<2>	SPI_adr<1>	SPI_adr<0>	upload/ read back	Global	Decoder Address			Enable		
Detail	0	0	0	1=upload 0=readback	1=snapshot 0=rolling shutter	LSB		MSB	LSB		MSB

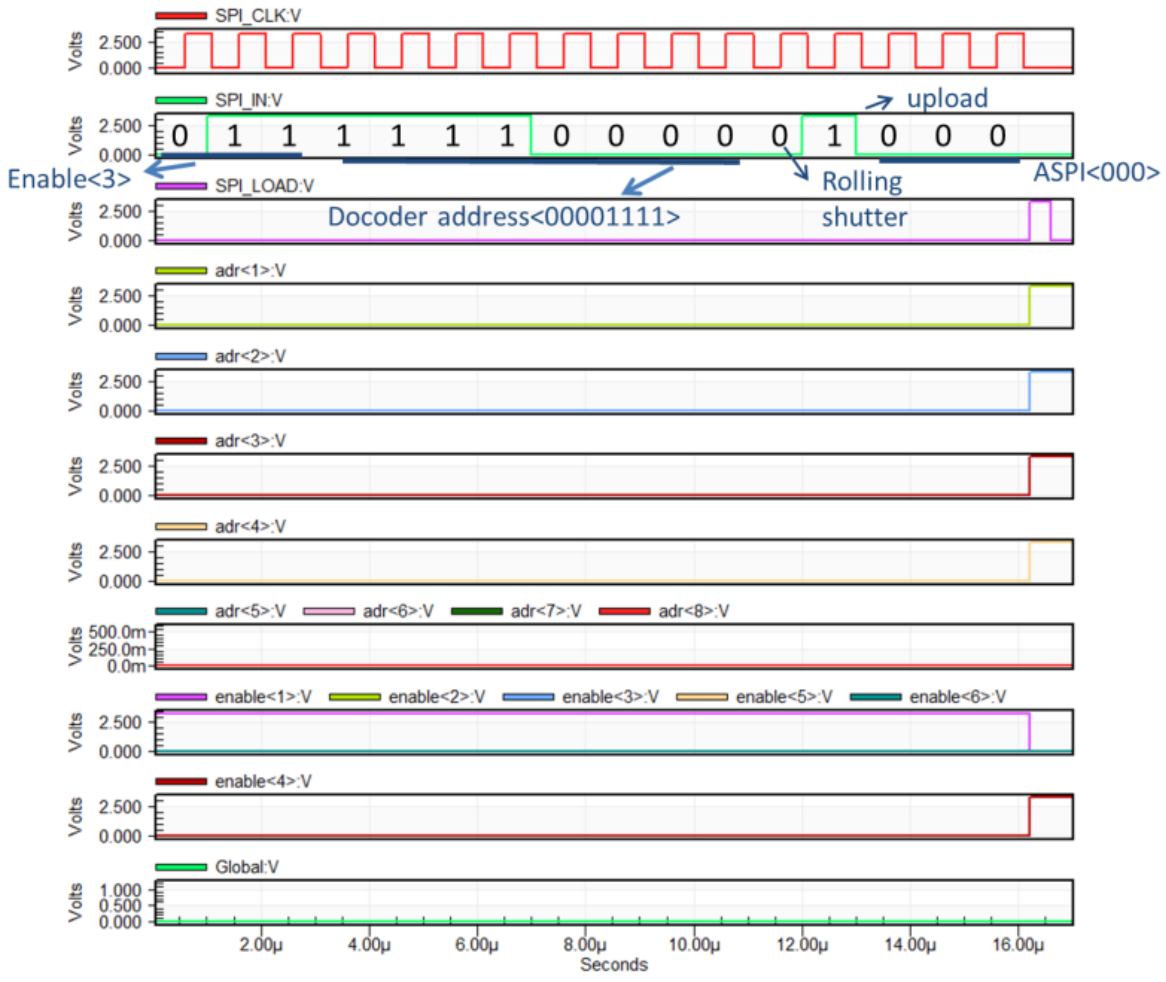


Figure 3.24: Simulation of the upload mode

Figure 3.24 shows the simulation of the upload mode.

- The first three waveforms are SPI_CLK, SPI_DATA_IN, SPI_LOAD signals.
- The rest waveforms are the 8 address codes, 6 enable codes selecting pixel row driver blocks and Global signal uploaded by ASPI.

Based on Table 3-5, focusing on the second trace SPI_DATA_IN in Figure 3.24, it select the ASPI<000> to let Global to be 0 to choose the rolling shutter mode, enable the 4th pixel row driver block and send the address code <00001111> to the chosen block. After the arrival of the pulse of the third trace SPI_LOAD, we can see that address <1:4> become high and address<5:8> keep low. Only Enable<4> is high while other enable codes are low and Global signal also keep low. That means we get the right output signals with those input codes.

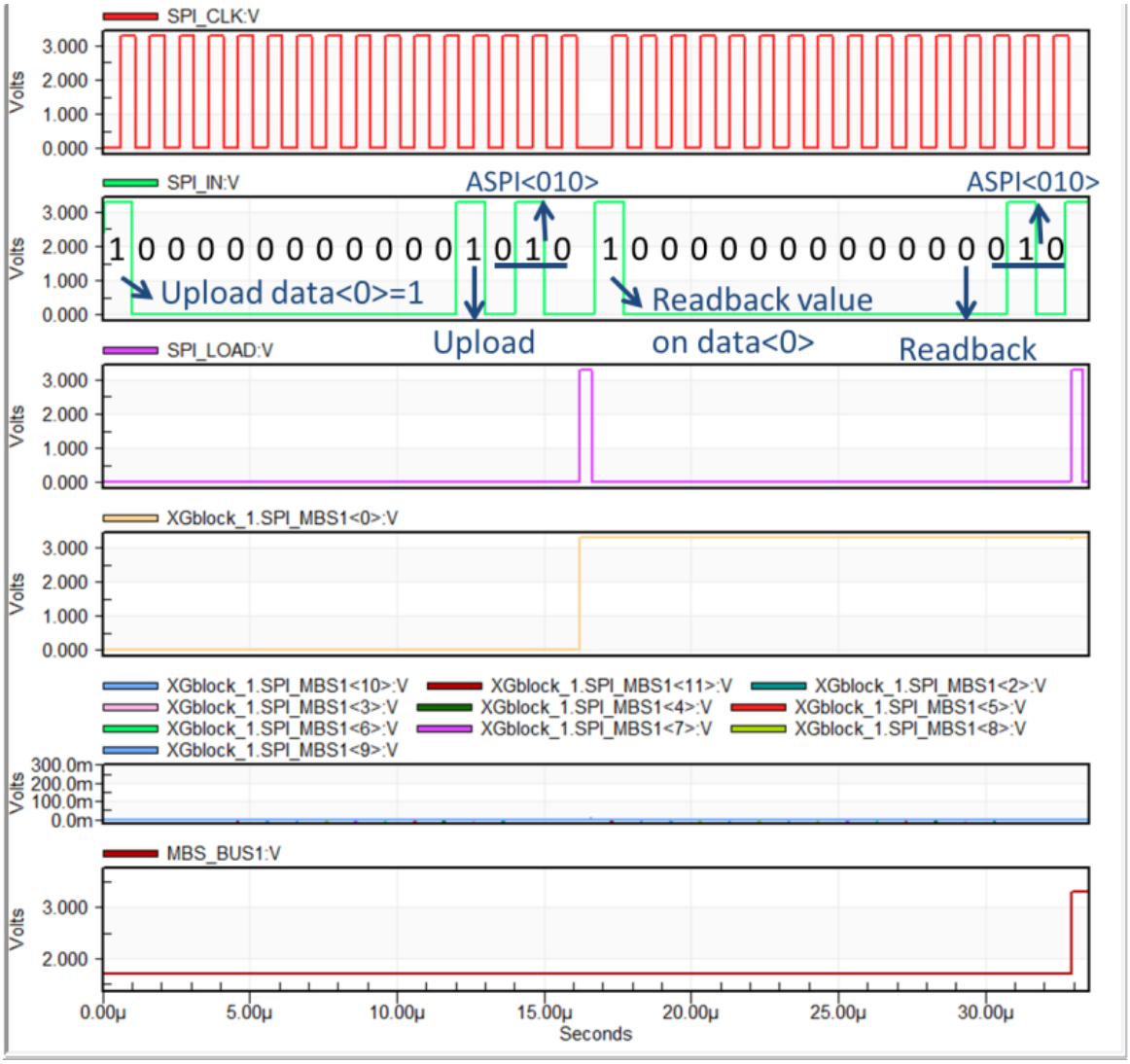


Figure 3.25: Simulation result of the readback mode

Figure 3.25 shows the simulation of the readback mode.

- The first three waveforms are SPI_CLK, SPI_DATA_IN, SPI_LOAD signals.
- The next 12 traces are the outputs of the ASPI whose address is <010>.
- The trace in the last waveform is the SPI_MBS.

From the first trace in red which is SPI_CLK, we can see there are two transmission cycles in this simulation. In the first cycle, we upload 1 into the 1st register in ASPI<010>, and let the data in the rest registers to be 0. Then, in the second cycle, we choose to readback the data in the 1st register in

ASPI<010>, which give us the right value '1' on the SPI_MBS node after the pulse of SPI_LOAD arrives. The validity of the readback function of out ASPI is proved.

3.3.7. Layout consideration

In Figure 3.26 and Figure 3.27, we show the bottom left corner block as an example of our layout. VPIX need to horizontally propagate into the current limiter in the readout block at the north and south side of the sensor. So the VPIX bond pad needs to be located at the top left corner. The PMOS tuner for current limiter follows VPIX for the ESD protection consideration. Then there are the VDDD and VSSD digital power supply. At the bottom of the corner are the digital input signal bond pads and two MBS bond pad for sense and write respectively.

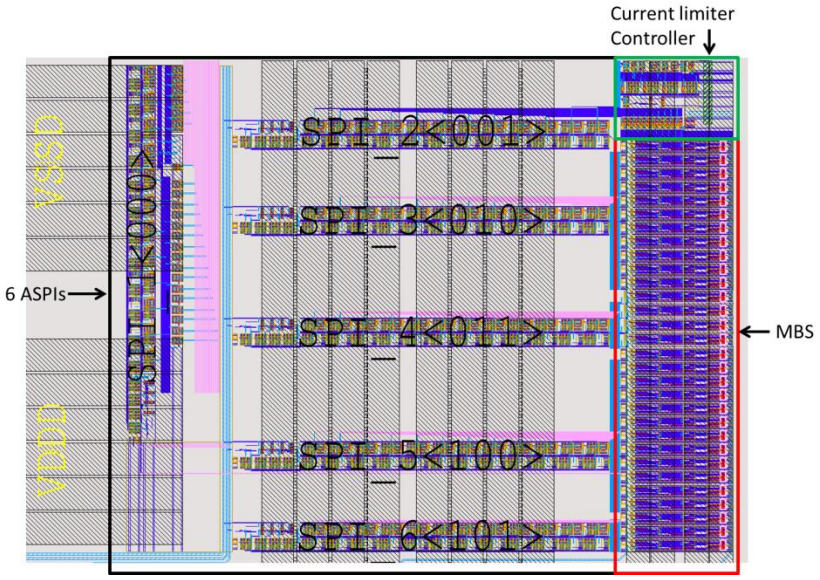


Figure 3.26: Layout of 6 ASPIs and MBS

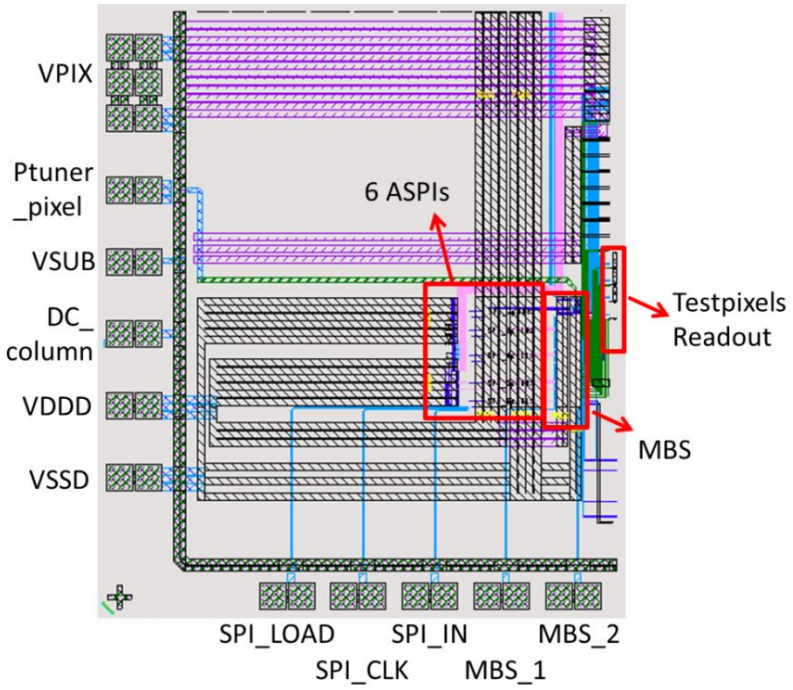


Figure 3.27: Layout of the whole digital control block

3.4.Readout Block

3.4.1. Overview of the readout block

The readout block is mainly for the X-axis scanning and pixel signal readout. Figure 3.28 shows the overview of readout block topology.

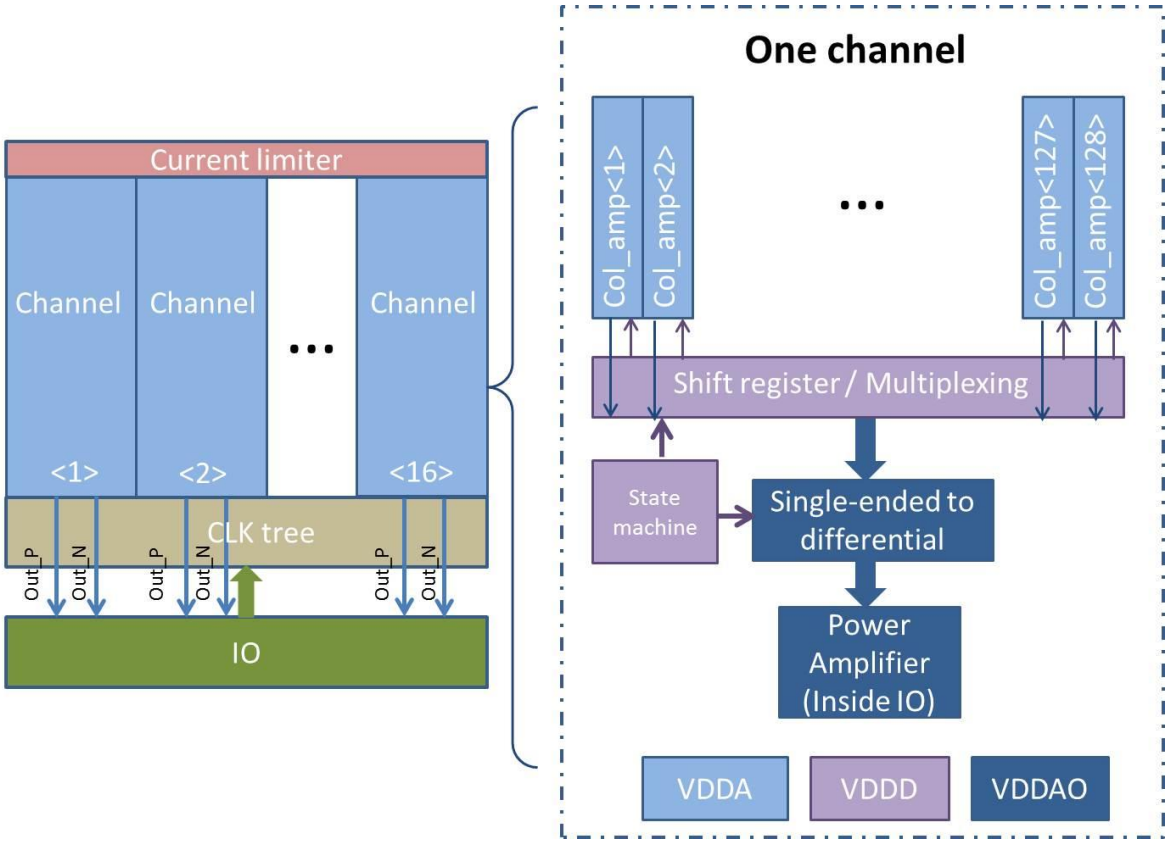


Figure 3.28: Overview of readout block

In one readout block, it contains:

- Current limiter for limiting the VPIX peak current
- 16 readout channels working at the same time.
- 1 clock tree for split a master clock or other fast digital signals into 16 identical signals and sending them into every channel.

Every readout channel contains:

- Column readout – the readout of pixel signal, doing CDS and send them to channel readout
- Readout multiplexing – X-axis scanning, which realizes the multiplexing of columns to readout lines
- State machine – generating pipeline clock and sync signals
- Channel readout – containing single-ended to differential signal converter (S2D) and final signal readout buffer to bond pads.

We have three different power supplies VDDA for the column readout, VDDD for the digital multiplexing and state machine, and VDDAO for channel readout. Different filling color of the rectangle shows their power supply domain.

3.4.2. Current limiter

The purpose to implement a current limiter has been discussed in Section 3.1.6. The current limiter is a current source controlled by a shift register. Figure 3.29 shows the schematic. Figure 3.30 shows how it is implemented into the sensor.

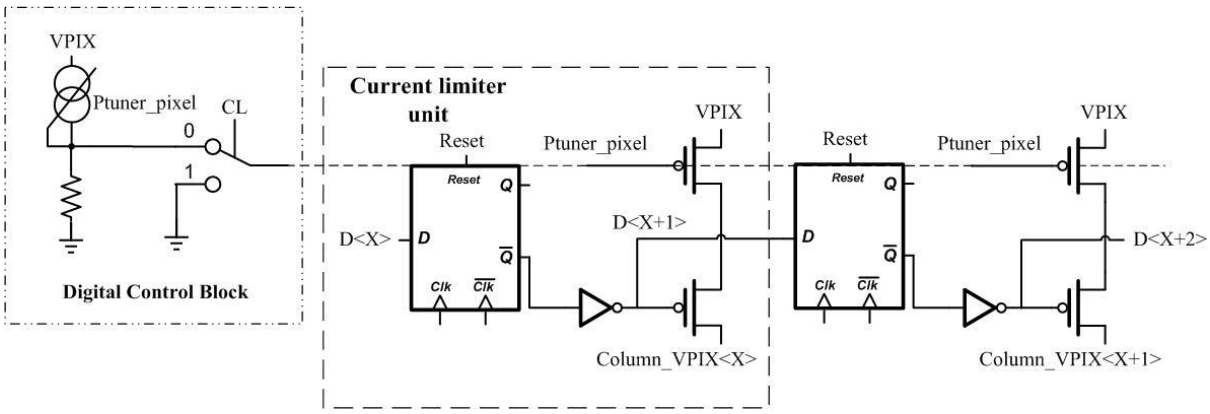


Figure 3.29: Schematic of current limiter

In Figure 3.29, the CL signal which controls the biasing of current limiter is provided by one ASPI bit. When CL is 0, it means current limiter will work, and the current flow over one unit cell of the current limiter is a mirror of the Ptuner_pixel. When CL is 1, it means the bias is connected to VSS, so the PMOS in current limiter works as closed switches. The current limiter no more limits the current on VPIX. The reason to choose CL=0 to activate the current limiter is because the default output value of the register of ASPI is 0 when it is powered on, and there is a huge peak current on VPIX as well when VPIX is powered on. Thus, the current limiter is activated as soon as we power on the sensor and supposed to avoid the peak current on VPIX during power on.

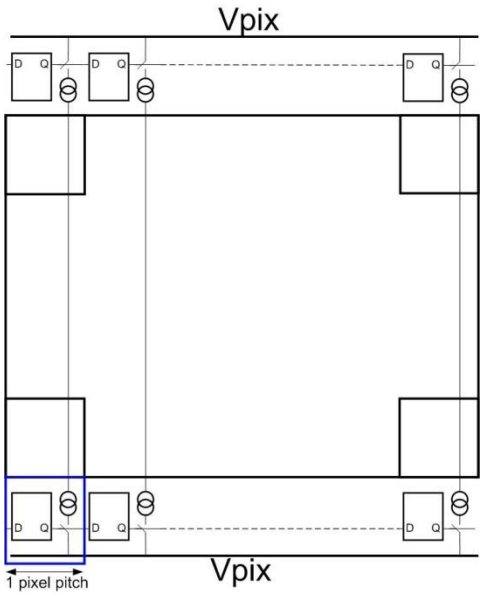


Figure 3.30: Current limiter implementation

In Figure 3.30, since VPIX is routed vertically column by column, the output of each unit cell of the current limiter is connected to every column VPIX. And for one column it is powered both from north and south.

The current limiter also provides us with a possibility that when one column VPIX is short with another column VPIX, then we can disable those VPIX by setting the shift register to be 1, since the sensor can still be functional without one or two columns of pixels.

3.4.3. Column readout

A brief schematic of column readout is shown in Figure 3.31.

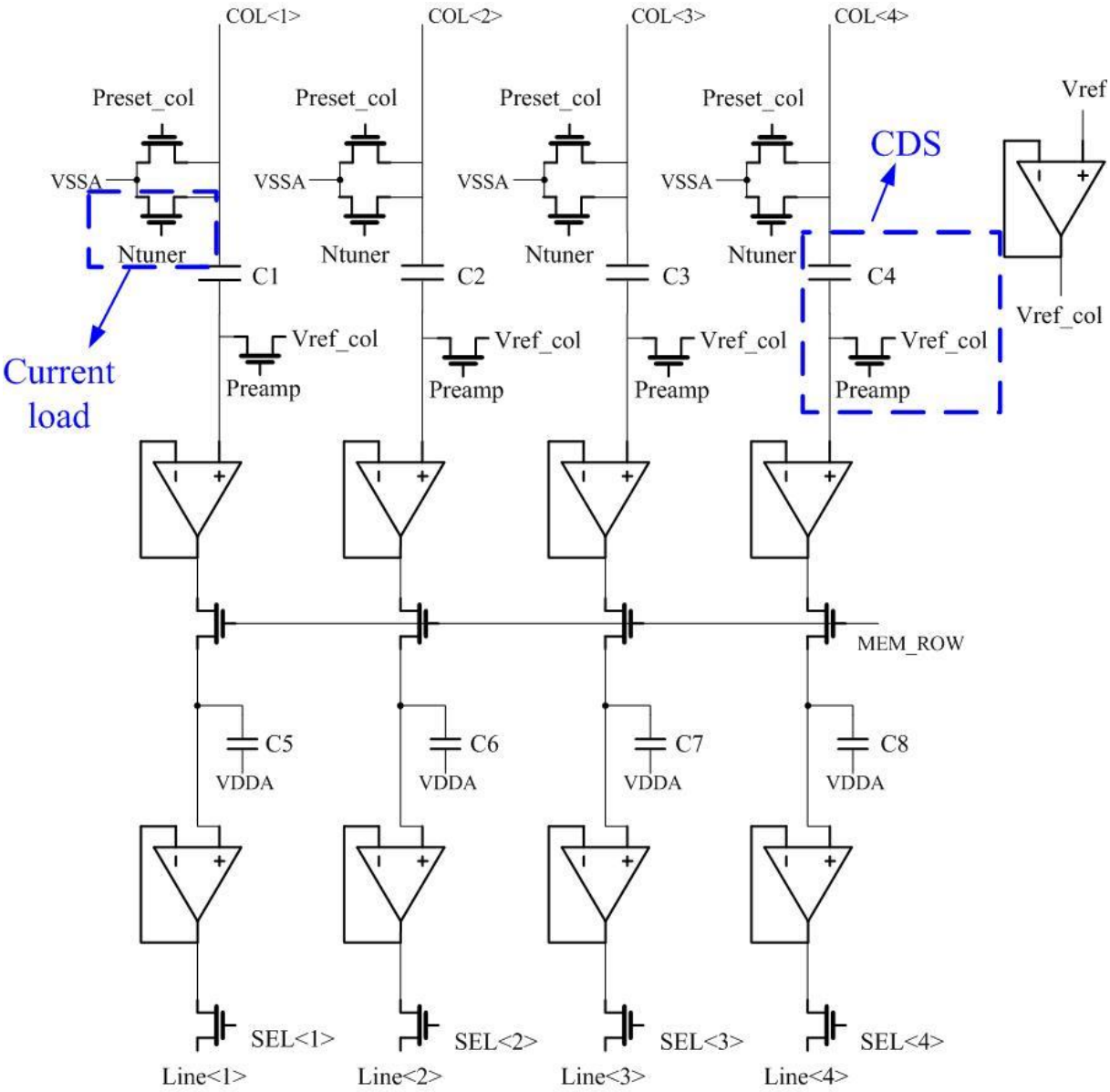


Figure 3.31: Schematic of column readout

It contains column current load for pixel readout, also CDS to cancel the reset noise and buffers to send the signals on capacitors to channel readout stage. Every time a kernel of 8 pixels is read out simultaneously. To simplify the drawing, only 4 columns are shown. 8 columns share one buffer for the reference voltage to avoid crosstalk on the V_{ref} signal. The 8 outputs of one column readout unit

will be sent to 8 readout lines called Line<1:8> when SEL<1:8> switches are closed as Figure 3.31 shows. The SEL<1:8> are controlled by X-axis scanning (readout multiplexing) shift registers.

CDS is implemented by capacitors C_1-C_4. After the CDS, the signals are stored on capacitors C_5-C_8 when MEM_ROW switches are closed. C_5-C_8 are accumulation capacitors. For the sake of better linearity, we choose one node of them to be VDD since our signals values are much closer to VSS.

3.4.3.1. CDS implementation

The CDS is realized with a MIM capacitor and a reference voltage V_{ref}. The steps to complete CDS are shown in Figure 3.32.

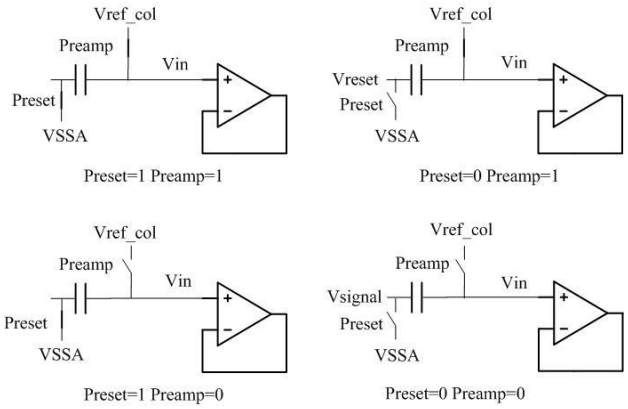


Figure 3.32: CDS implementation

First we clamp the input node to VSS, and then set V_{reset} on one side of the MIM capacitor, while another side is the V_{ref}. So the voltage difference V_{ref}-V_{reset} will be stored on the MIM capacitor. After that, the "Preamp" switch is open and we preset the input node again. The pixel signal V_{signal} is set on the MIM capacitor. Finally, it will give us:

$$V_{in} = V_{ref} - V_{reset} + V_{signal} \quad (3\text{-}2)$$

3.4.3.2. Averaging implementation

Averaging mode here is for increase frame rate while sacrificing resolution. As explained in Section 3.1.3, we choose to average the signals from 4 adjacent pixels from one kernel of pixels, and then only read out one of four to make readout multiplexing faster.

There are three available positions (position A, B, C) for us to implement the averaging in the column readout, which is illustrated in Figure 3.33.

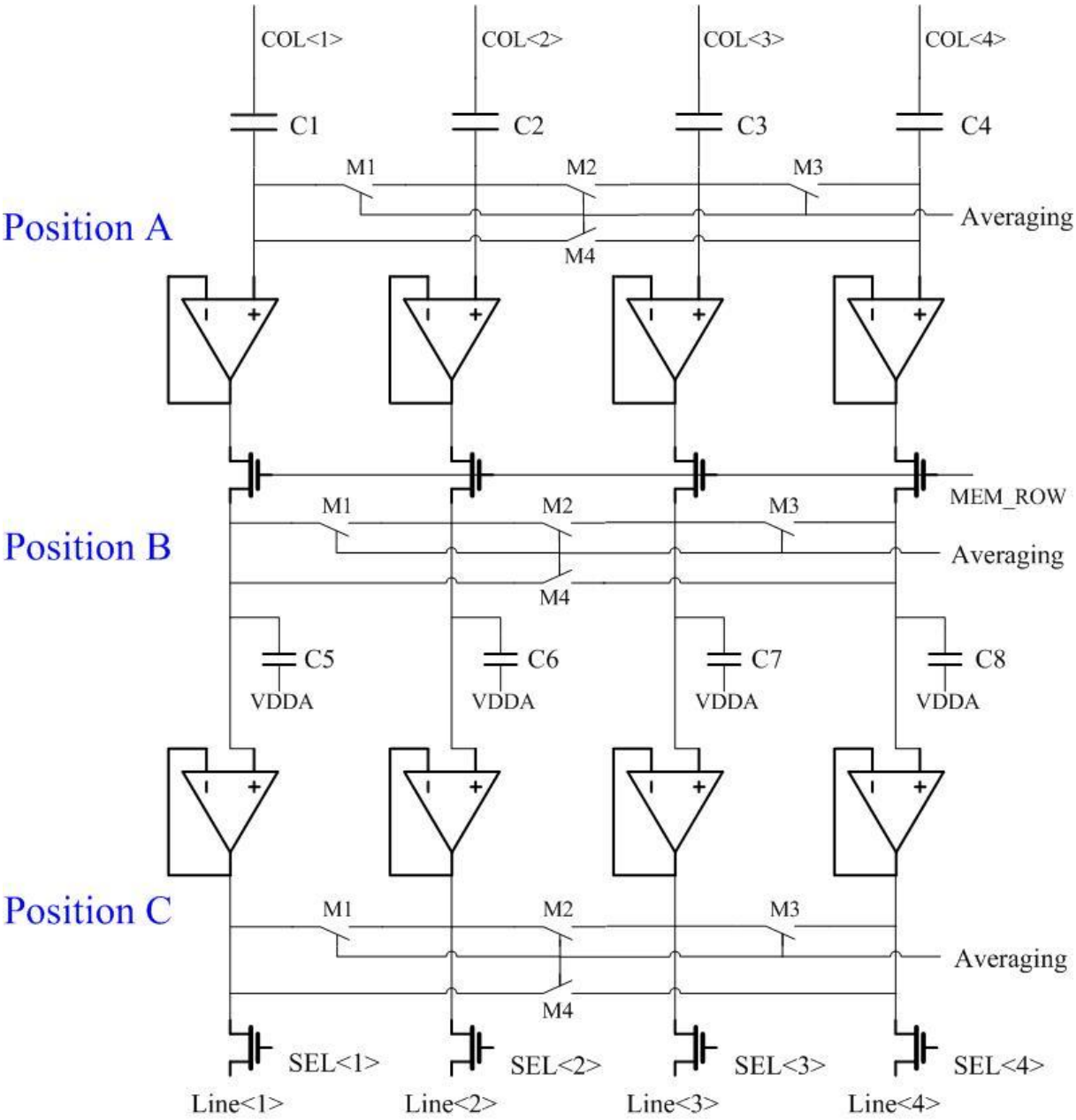


Figure 3.33 Possible averaging positions in column

The averaging is realized by closing the switches M_1-M_3 between 4 column signals, due to the charge sharing, the average value of those 4 signals will be stored on the capacitors. To make the parasitic environment of every column equal to ensure the linearity, we add an extra switch M_4 between the first and last column, so each column sees two switches.

To compare which position is better, we run a simulation under the same test bench with all three options. Providing different sets of 4 signals that have the same average value can help us to check the performance of the averaging. The results are listed in Table 3-6.

In Table 3-6, the signal deviation means after averaging, the difference between 4 output signals with their average value under one set.

The linearity deviation here means the difference between the average values got from different sets.

Table 3-6 Averaging performance at different positions

Situation	Time needed for signal to stable [ns]	Signal deviation [V]	Linearity deviation [V]
Position A	160	Almost 0	Almost 0
Position B	10	Almost 0	Almost 0
Position C	5	0.04	0.12

Because at position A, the signals are not directly buffered, so the averaging takes longer time, which adds extra time for the readout in averaging mode than in non-averaging mode.

For the situation at position C, the averaging is happen on the output impedance of the buffers, not directly on the capacitors. The signals store on the capacitors will beat the buffer to against the averaging, and result in bad performance.

So, averaging at position B becomes our choice which is fast and gives excellent average results.

Figure 3.34 is the functional simulation for the averaging at position B.

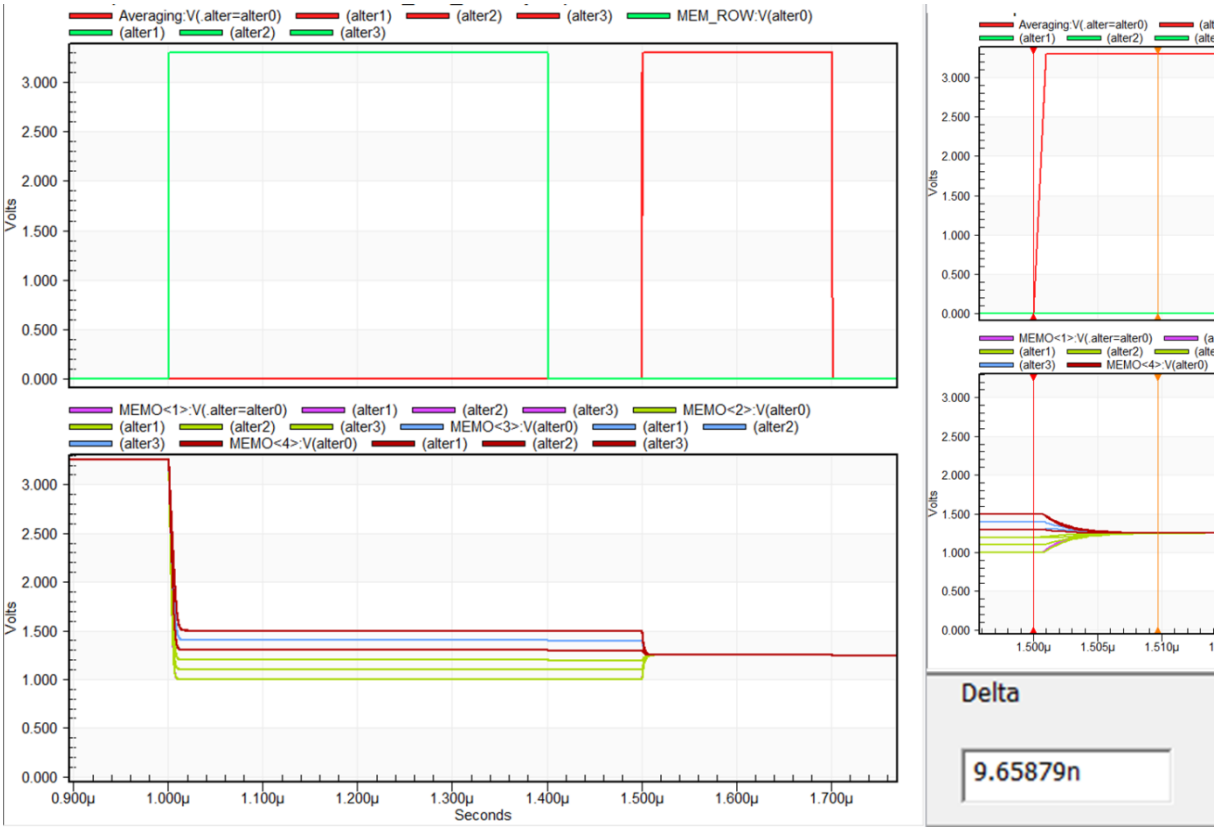


Figure 3.34: Simulation result of averaging

In Figure 3.34:

- The traces in the top waveform are MEM_ROW signal in red and Averaging signal in green. According to Figure 3.33, MEM_ROW signal controls the switches to store the voltages on capacitor C_5-C_8. Averaging signal controls the switches M_1-M_4 to average the 4 voltages stored on C_5-C_8.

- The traces in the bottom waveform are the outputs of the readout columns named MEMO<1:4>. As mentioned before, in the simulation, there are 3 sets of those 4 signals which share the same average value (3 variations are named alter0, alter1, alter2 in Figure 3.34).

From the bottom traces in Figure 3.34, after the arrival of the pulse of the Averaging signal, the output of those signals completely overlap with each other and become one single line, so it proves that the averaging function meets the requirement.

The right part of Figure 3.34 shows the measurement of the time interval for averaging. We can see the time needed to average the signals is less than 10 ns. That means the averaging function will not add extra time to read out the line.

3.4.4. Readout multiplexing

3.4.4.1. Line Multiplex

The sensor is required to achieve 65 MHz readout frequency. To realize high speed readout while optimize the memory effect and power consumption, a pipeline method is applied to separate 1 line for readout to be 8 paralleled lines that each of them works at 65/8 ≈ 8 MHz. That bandwidth reduction also helps to reduce the readout noise. The signals on the 8 lines in Figure 3.35 are sent to 8 single-ended to differential signal converters (S2D). After an 8 to 1 multiplexing, final differential signals are buffered to the output pads.

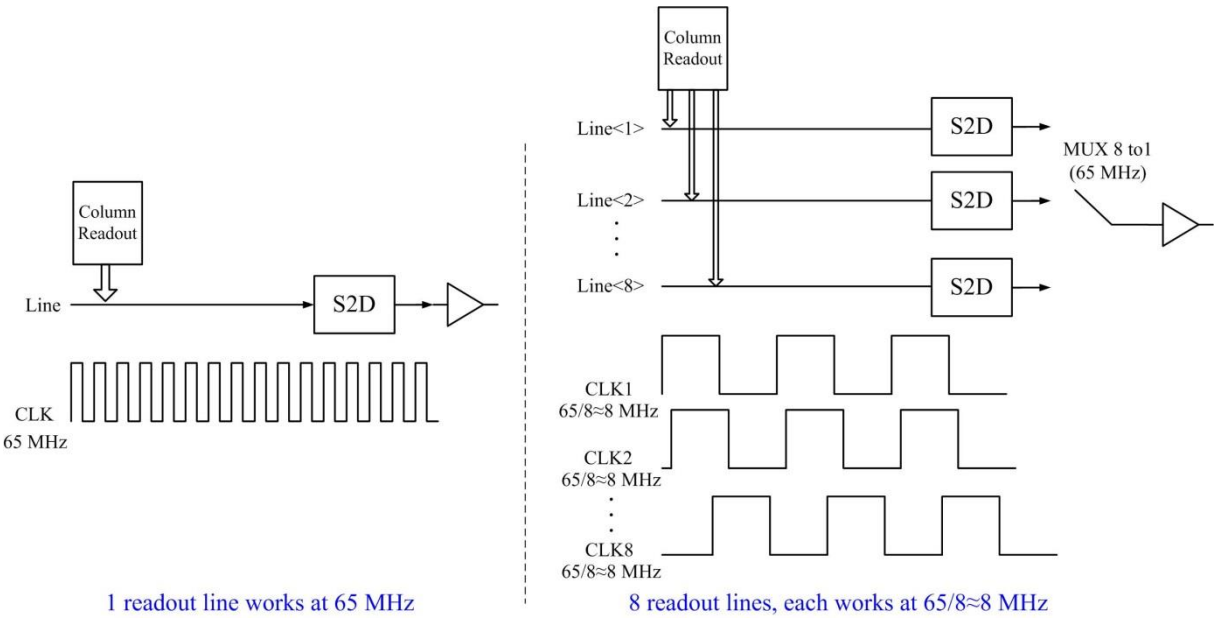


Figure 3.35: One line readout vs. 8 lines multiplexing readout

This sensor can work in non-averaging mode and averaging mode. To realize a speed advance in averaging mode, the multiplexing is different. A brief diagram of different readout sequences in non-averaging and averaging mode is shown in Figure 3.36 and Figure 3.37.

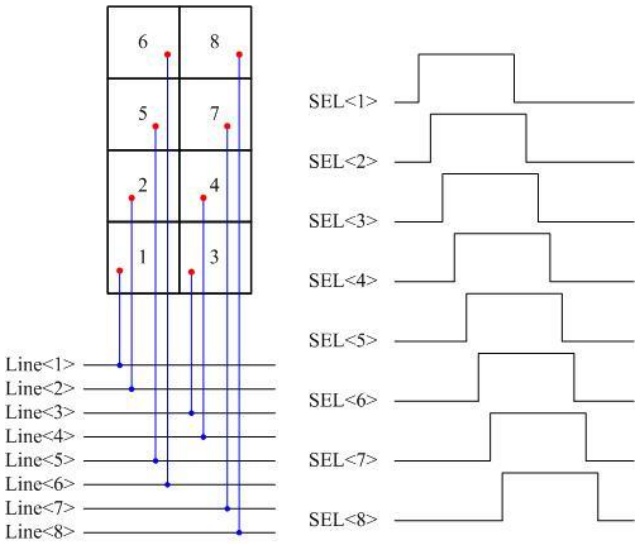


Figure 3.36: Multiplexing in non-averaging mode

As shown in Figure 3.36, in non-averaging mode, the 8 signals of one pixel kernel is read out and sequentially sent to the 8 lines in order. The SEL<1> to SEL<8> signals are the select signals in readout columns control the output of the readout culumns to be sent on the corresponding readout lines.

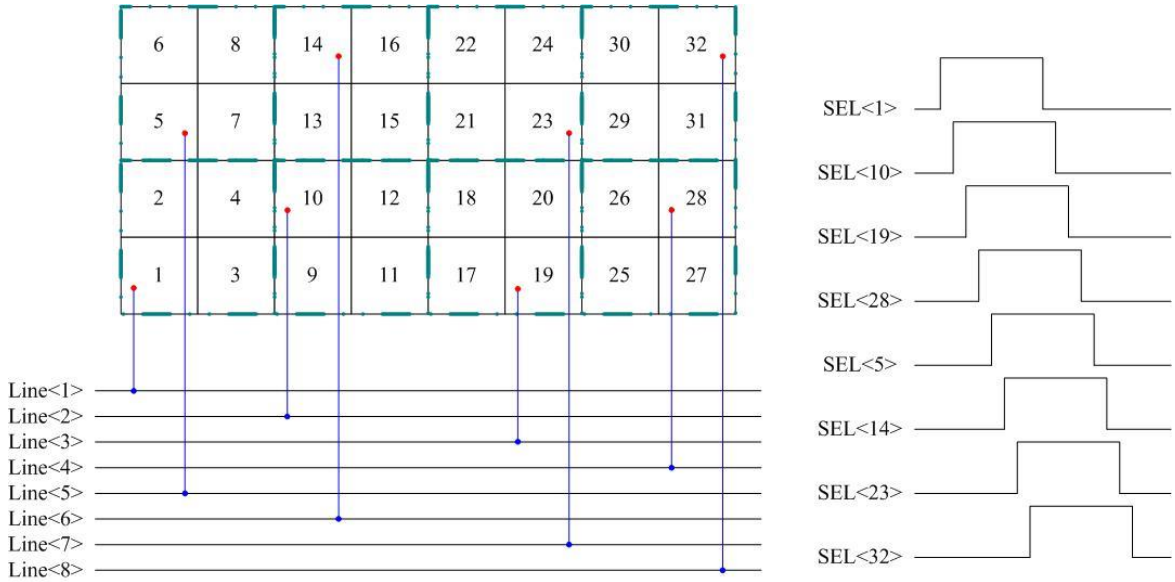


Figure 3.37: Multiplexing in averaging mode

In averaging mode, as explained in section 3.4.3.2, the output signal of 4 adjacent pixles (labelled by light blue dashed lines in Figure 3.37) are averaged. So choosing to read out one of them can reduce the total number of signals to read out. In Figure 3.37, we can see the multiplexing sequence are different from the one shown in Figure 3.36. The reason is that the connection from column output to readout line are always the same in both modes, if we read out the signals belonging to the same position in one group of 4 adjacent pixels, for example, No.1 and No.9 in Figure 3.37, since they are both connected to Line<1>, it will generate a signal conflicts. So in one cycle, to read out pixels at different positions belonging to different readout lines is neccessary for increasing readout speed.

For example, as shown in Figure 3.37, we read out No.1 on Line<1>, No. 10 on Line<2>, No. 19 on Line<3> and No.28 on Line<4>.

The select signals or X-axis scanning signals which control the outputs of readout column to be loaded on the 8 readout lines are provided by shift register chain.

Figure 3.38 and Figure 3.39 illustrate how the signals in the shifter registers propagate in non-averaging mode and averaging mode.

Those shift registers are initialized by 8 signals called BIT <1:8> in non-averaging mode, or by 8 signals called BIN_BIT <1:8> in averaging mode. There is a circuit in the state machine generating BIT<1:8> or BIN_BIT<1:8> by utilizing 8 slow sync signals, which will be discussed in the next section. A signal called Enable_Averaging is used to switch between the averaging and non-averaging modes. This signal is provided by the ASPI.

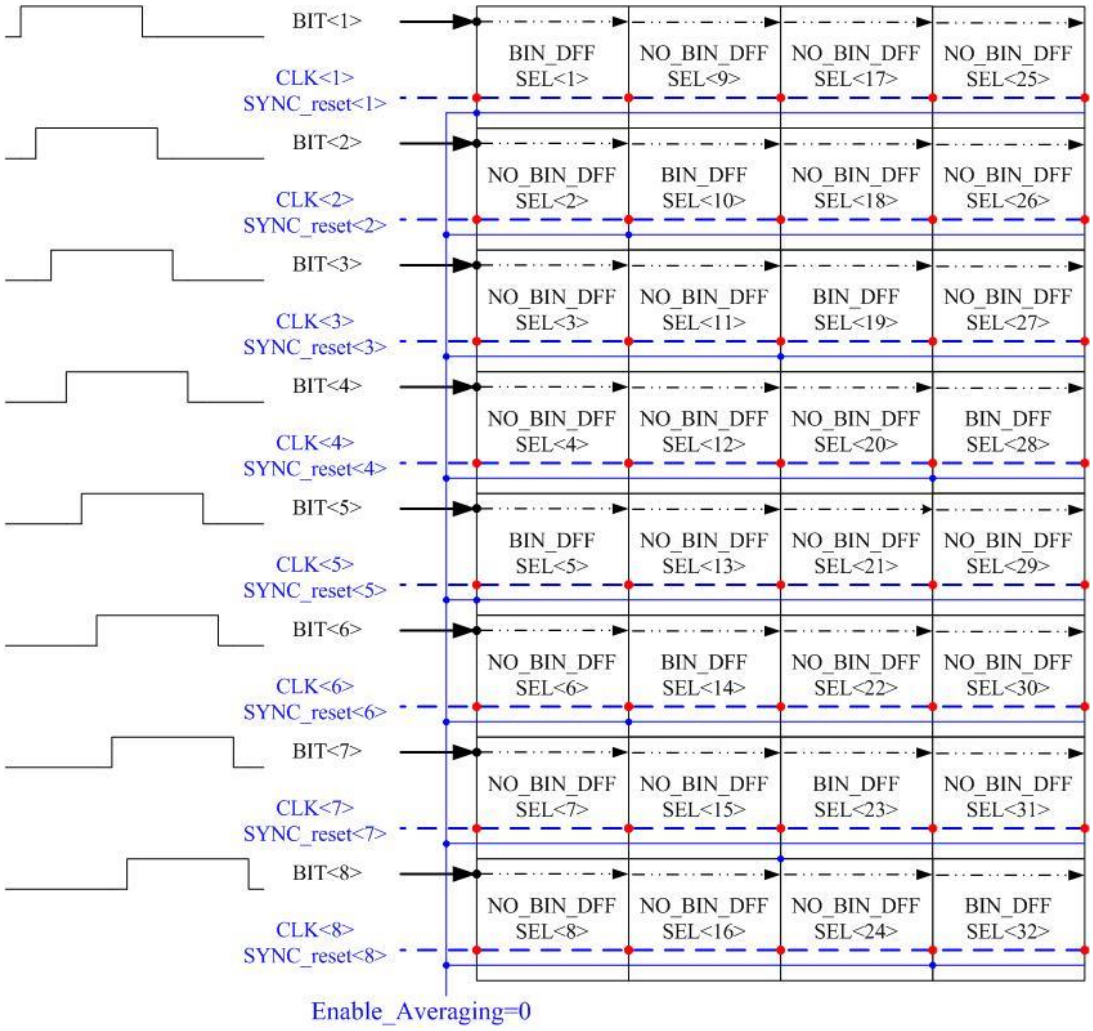


Figure 3.38: Multiplexing propagation in non-averaging mode

In Figure 3.38, there are 8 paralleled shift registers, and the 8 signals BIT<1:8> are the inputs of those 8 shift registers. There is a time interval equal to one clock cycle between BIT<N> pulse and BIT<N+1> pulse. The 8 clock signals CLK<1:8> and 8 synchronized reset signals SYNC_reset<1:8> are generated by state machine by dividing a fast master clock signal. It will be explained in the next section. In non-

averaging mode, the 8 BIT<1:8> signals will sequentially propagate through every DFFs in the 8 shift registers, which generate select signals SEL<1> to SEL<N> in order.

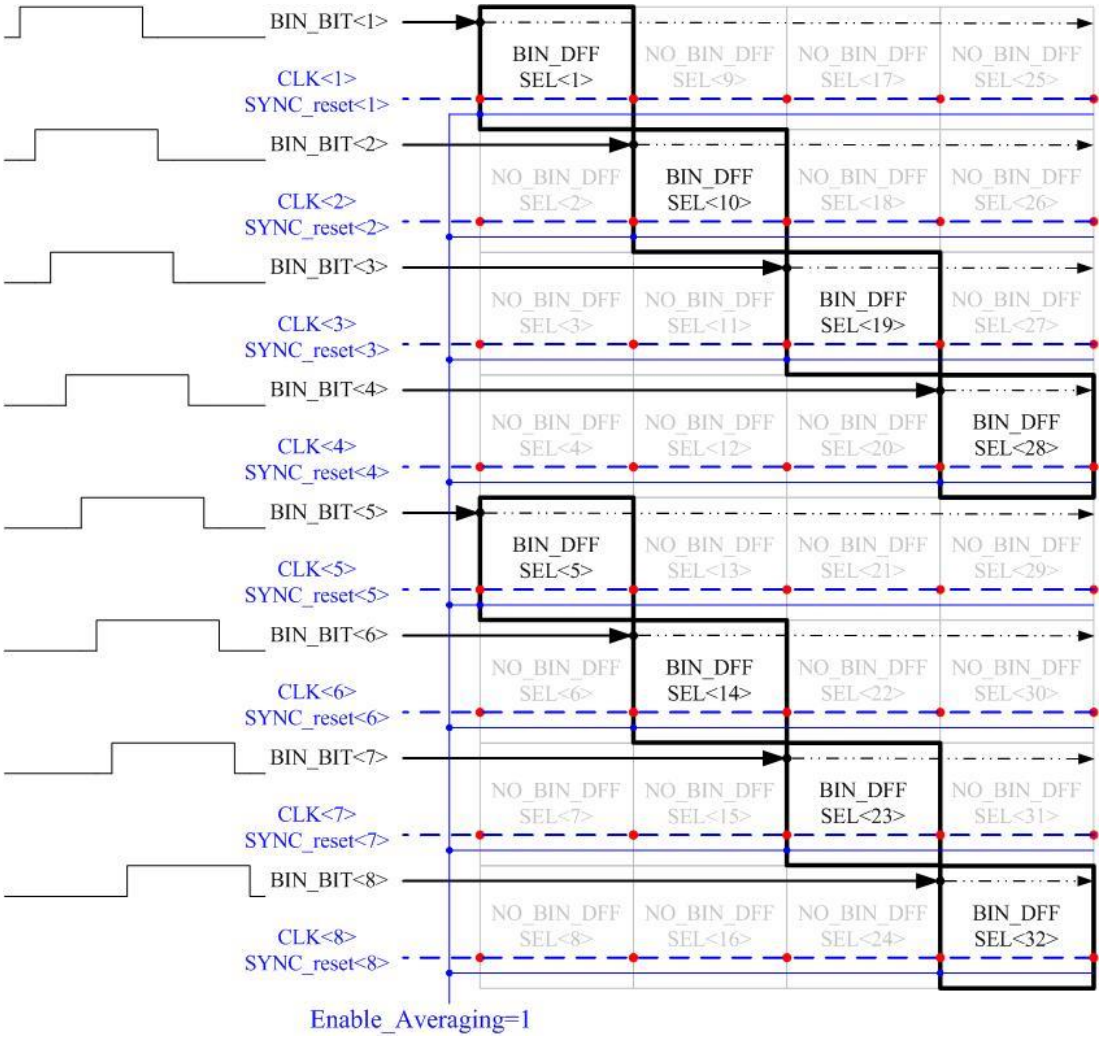


Figure 3.39: Multiplexing propagation in averaging mode

In Figure 3.39, similar to the non-averaging mode, the 8 signals BIN_ BIT<1:8> are the inputs of the 8 shift registers. As discussed earlier, to reach a speed advance in averaging mode, only one signal from a group of 4 signals will be read out. So, only one corresponding DFF is working while the rest three DFFs are bypassed. In Figure 3.39, the DFF which works in averaging mode is highlighted with black bold lines and the DFFs which are bypassed is in dim grey.

3.4.4.2. State machine

The state machine is a digital block which contains frequency dividers and the generator of BIT<1:8> and BIN_BIT<1:8> signals and other signals needed for single-ended to differential signal converter.

The frequency dividers provide 8 pipelined 8 MHz clock and synchronized reset (SYNC_reset) signals which are divided from 65MHz master signals. To achieve the divide of 1 fast 65 MHz master clock into 8 slow 8 MHz signals, first we have a shifter register which contain N DFFs. (N varies according to how many signals we need), which is shown in Figure 3.40. An OR gate with SR flip-flop is used to get the frequency we want.

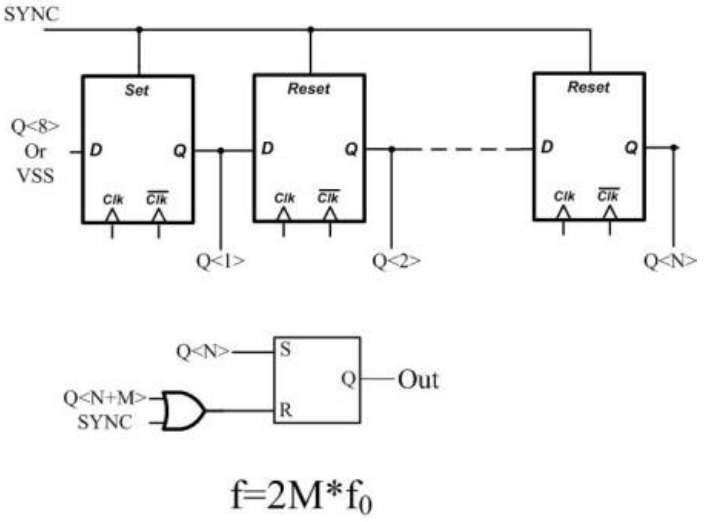


Figure 3.40: Schematic of frequency divider

From Figure 3.40, we suppose the master clock working at frequency f_0, and then the frequency f of the output signal of the SR flip-flop got after division is

$$f = 2 \times M \times f_0 \tag{3-3}$$

Moreover, the BIT<1:8> and BIN_BIT<1:8> discussed in the former section are generated by the circuit shown in Figure 3.41. It can switch to generate BIN_BIT<1:8> or BIT<1:8> by the Enable_Averaging signal which is provided by one bit from ASPI.

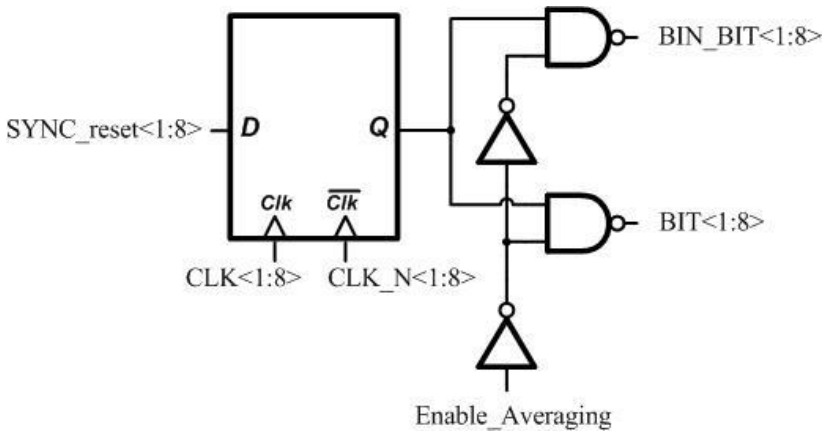


Figure 3.41: Schematic of BIT<1:8> and BIN_BIT<1:8> generator

We take advantage of 8 “SYNC_reset” signals to generate BIT and BIN_BIT, which make the multiplexing happen followed by the reset signals as we want. Thus there will be no blank time in-between.

Figure 3.42 and Figure 3.43 are the figures showing the simulation results for the state machine and multiplexing in both non-averaging and averaging mode.

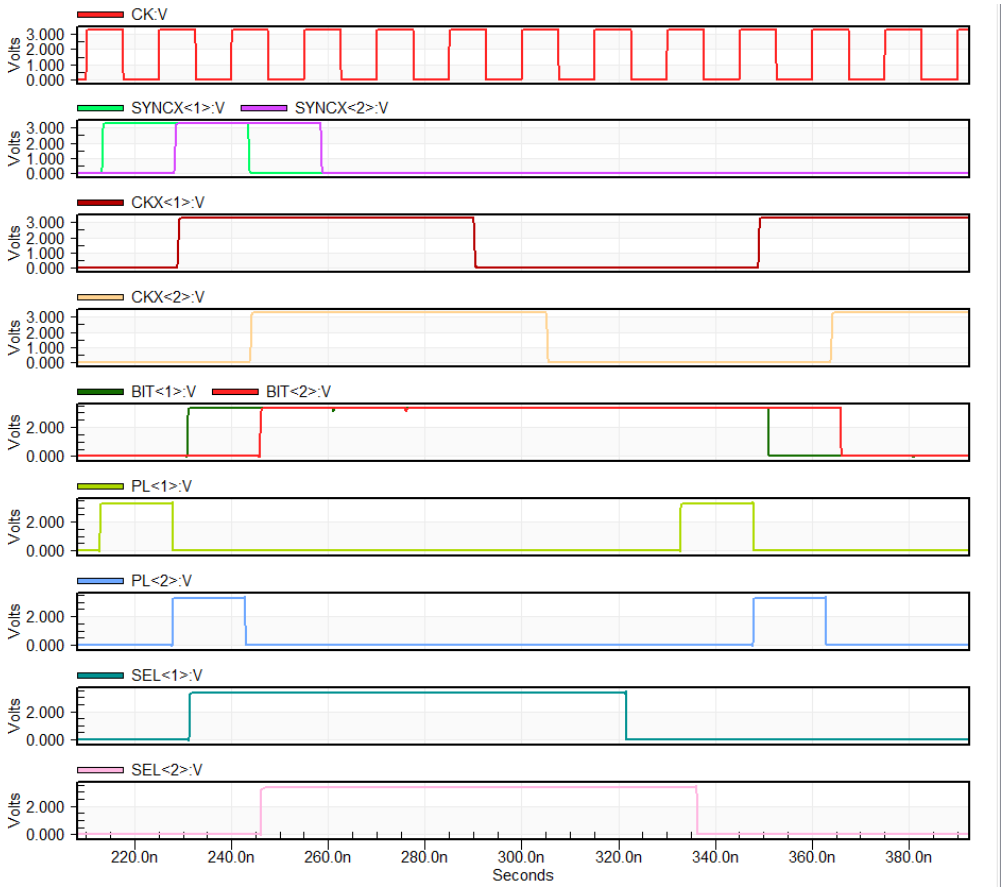


Figure 3.42: Simulation result of state machine and multiplexing in non-averaging mode

In Figure 3.42:

- The first red waveform is the 65 MHz master clock.
- The traces in the second and third waveform are the synchronized reset SYNC_reset<1:2> and slow clock signals CLK<1:2>.
- The two traces in the fifth waveform are the inputs of the X-axis scanning shift registers BIT<1:2>.
- The next two traces are preset line signal PL<1:2>. PL<N> will set the voltage on Line<N> to be dark level voltage which will be explained in Section 3.4.6.
- The last two traces are the select signal in readout column SEL<1:2>.

We can see from Figure 3.42, taking CLK<1> as an example, in one multiplexing cycle of CLK<1>, the line is first preset by PL <1> when SYNC_reset<1> arrives. Then the SEL<1> sends the output of one column on the readout line Line<1> when the pulses of BIT<1> and CLK<1> arrive.

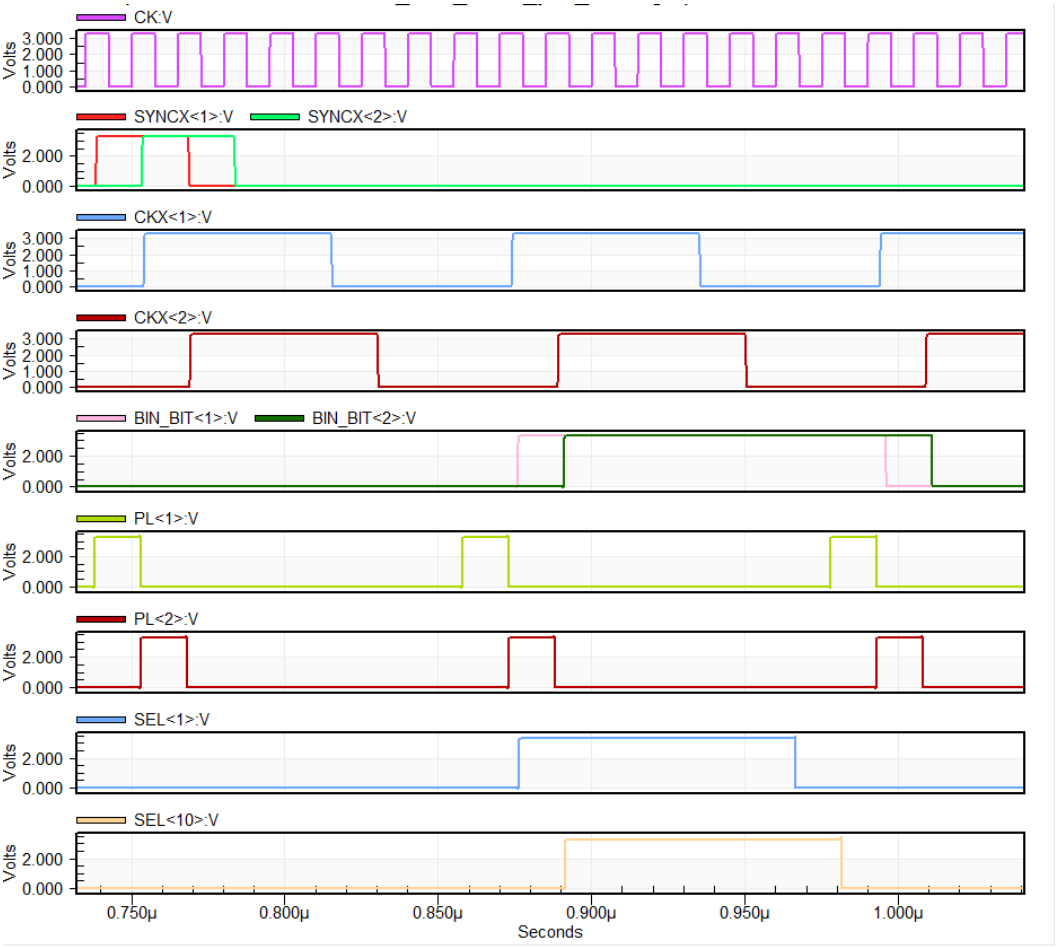


Figure 3.43: Simulation result of state machine and multiplexing in averaging mode

In Figure 3.43, similar to Figure 3.42:

- The first red trace is the 65 MHz master clock.
- The traces in the second and third waveform are the synchronized reset SYNC_reset<1:2> and clock signals CLK<1:2>.
- The two traces in the fifth waveform are the inputs of the X-axis scanning shift registers BIN_BIT<1:2>.
- The next two traces are preset line signal PL<1:2>.
- The last two traces are SEL<1> and SEL<10>, we can see the sequence here is different from the one in Figure 3.42 which has been explained in Figure 3.37.

3.4.5. Clock tree

Since one readout block contains 16 channels working at the same time, but the bond pads number is limited. We only can give one bond pad for one fast digital signal, e.g. clock, reset, averaging, etc. So a clock tree is needed in this sensor. It is to make sure those digital signals arrive at the same time in different channels to avoid a variation of clock skew from channel to channel. To separate one master signal into 16 uniform signals, a simple binary tree architecture is chosen as shown in Figure 3.44. It has strict symmetry of the 16 split signals, yet it gave challenges during layout.

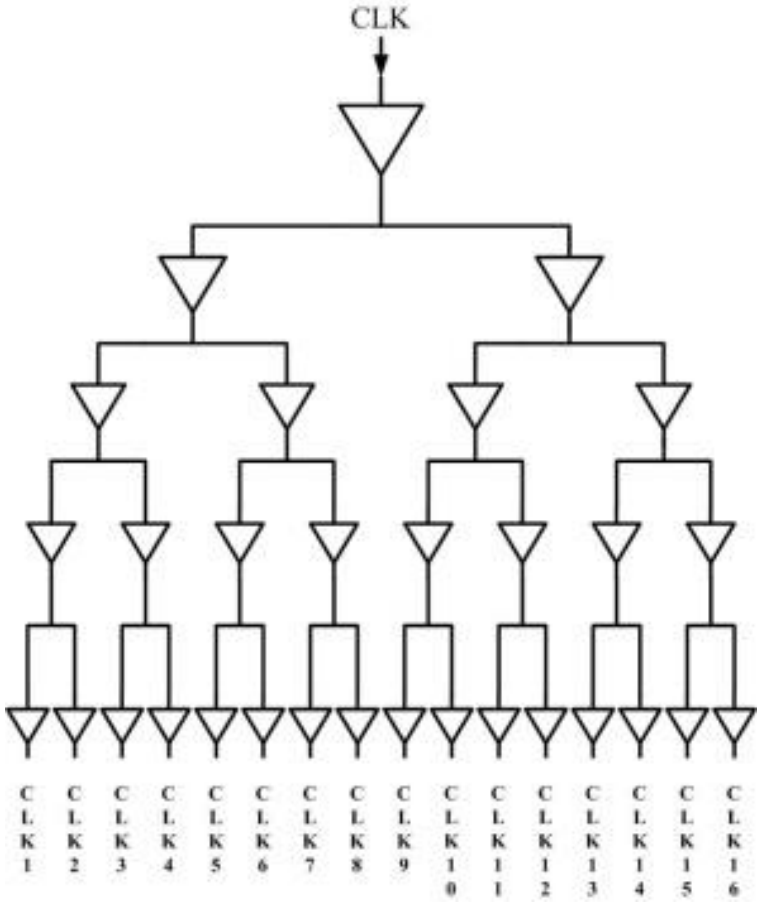


Figure 3.44: Structure of clock tree

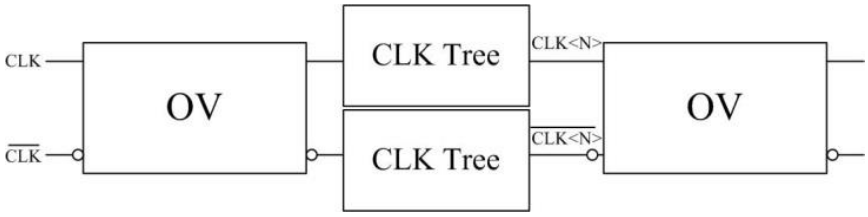


Figure 3.45: Diagram of how to maintain overlapping clock

We need nice overlapping clocks propagating through the state machine, so to avoid non-overlapping is happening due to the long metal wires parasitic, we add overlapping clock generator, labelled as “OV”, at both the input and output of the clock tree as shown in Figure 3.45 [3.2, 3.5].

3.4.6. Channel readout

The topology of channel readout is shown in Figure 3.46.

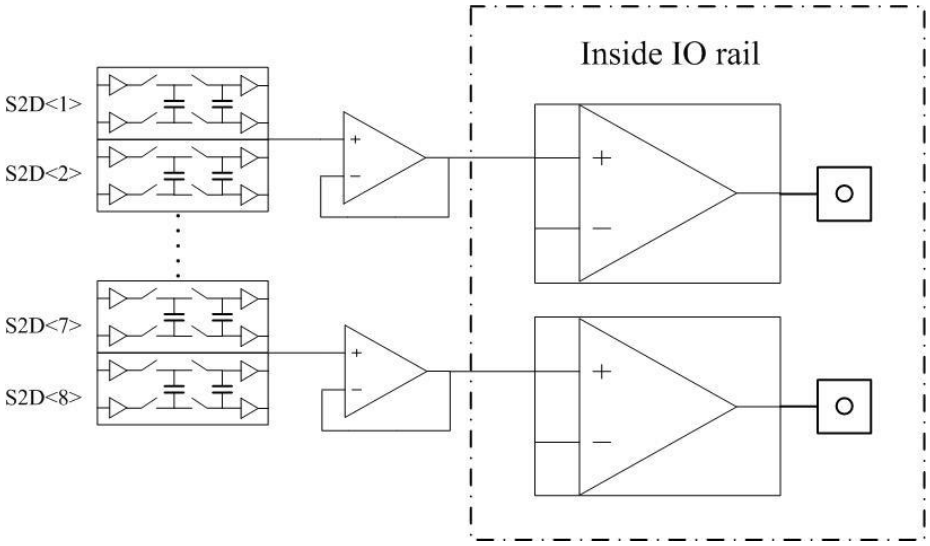


Figure 3.46: Overview of channel readout

After the signals are send on the 8 lines, we have 8 paralleled single-ended to differential signal converters (S2D) process the signals in pipeline and then send two differential signals to two intermediate buffers respectively. Those buffers connect to final buffers to the output bond pad. Due to the reason that we have heavy load from external transmission lines, the final buffers need to be powerful enough, but that will make their input capacitance also increase with the dimension. Thus, we add intermediate buffers in-between to reduce the load for the buffers in single-ended to differential stage.

3.4.6.1. Single-ended to differential converter

The sample and hold stage in the single-ended to differential signal converter is implemented as Figure 3.47 shows. An alternative of gain or no-gain mode are offered by two switches controlled by one bit from the ASPI called “gain” in Figure 3.47. The gain here means a gain of the output signal range compared with the pixel signal range. Since we use a differential output, so the final output will be

$$V_P - V_N = GAIN \times V_{pixel\ output} \tag{3-4}$$

Ideally in the gain mode, the gain is 2 since we double the signal range after the positive output minus the negative output, but there is a gain loss in the buffers through the readout path, so the final gain will be less than 2. From simulation result, the final gain is around 1.75.

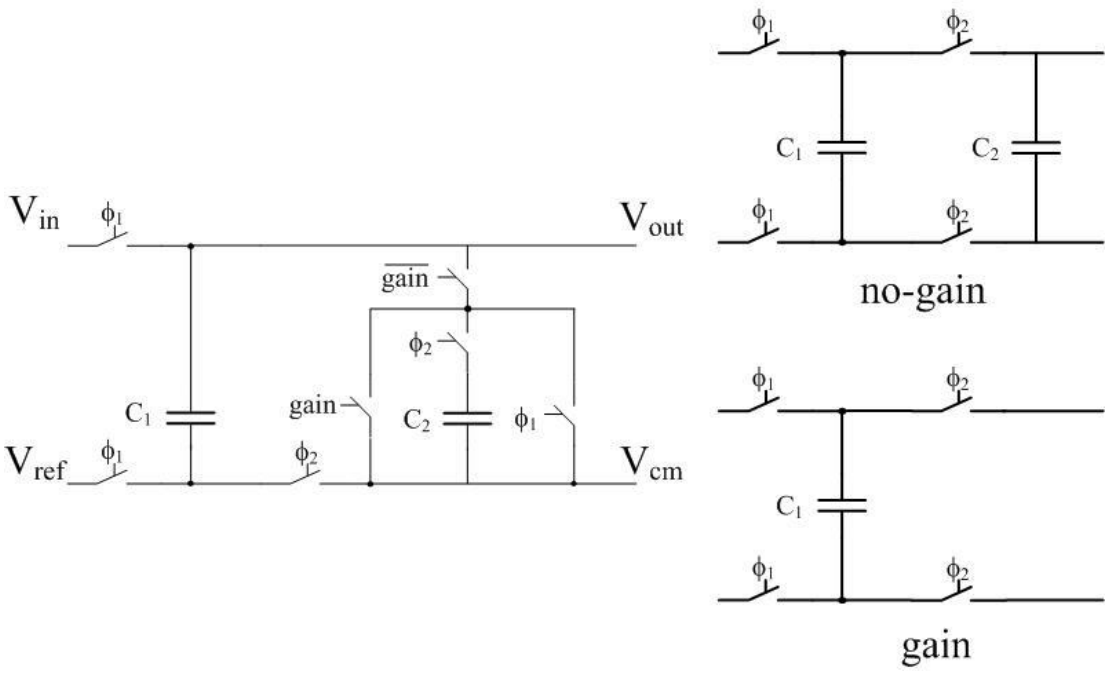


Figure 3.47: Schematic of sample and hold stage

As shown in Figure 3.47, if the user chooses no-gain mode, then when ϕ_1 switches are closed, the voltage difference between V_{in} and V_{ref} is stored on the capacitor C_1. When ϕ_2 switches are closed, charge conversion forces the V_{out} to be:

$$Q = C_1 \times (V_{in} - V_{ref}) = (C_1 + C_2) \times (V_{out} - V_{cm}) \quad (3\text{-}5)$$

Thus

$$V_{out} - V_{cm} = \left(\frac{C_1}{C_1 + C_2}\right)(V_{in} - V_{ref}) \quad (3\text{-}6)$$

Here we choose C_1 to be 600 fF, C_2 to be 500 fF, so the factor $C_1/(C_1+C_2)$ become 0.55. With a compensation of the gain loss, it will result in a gain close to 1 from the system view.

The single-ended to differential converter is implemented as Figure 3.48 shows.

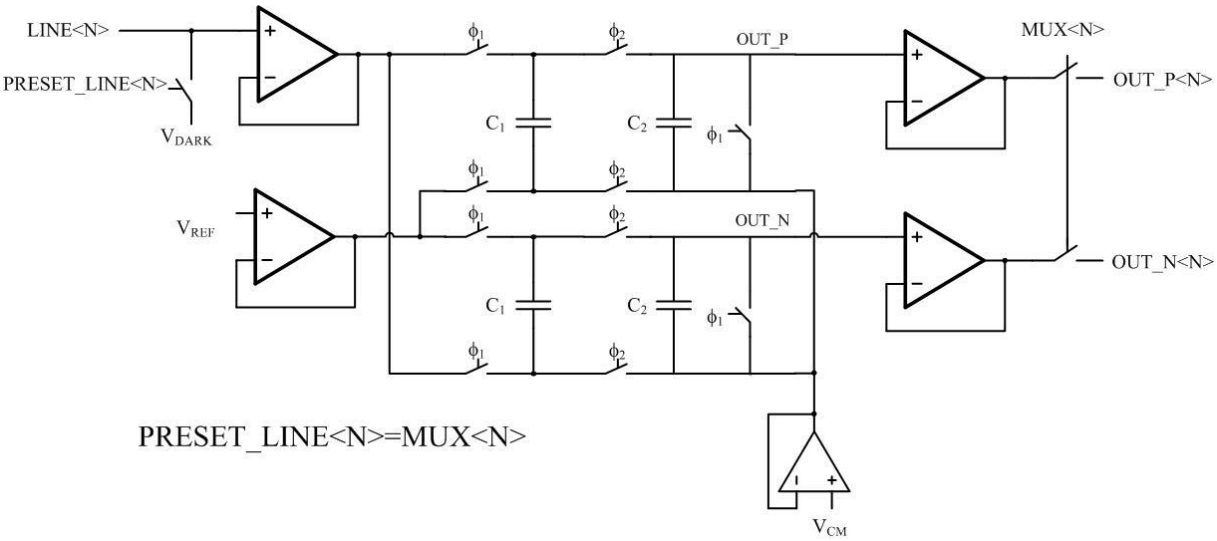


Figure 3.48: Schematic of single-ended to differential converter

The PRESET_LINE switch will set the voltage on the readout line to be the dark level signal V_{DARK}. The dark level signal voltage is the column output voltage of the pixel in dark. That preset happen at the beginning of every readout cycle to remove the former signal store on the line. The timing of PRESET_LINE (PL) signals is illustrated in Figure 3.42 and Figure 3.43. The MUX<N> signal here is equal to PL<N> signal, so every time when the Line<N> is preset to V_{DARK}, the output signals of former transmission cycle will be sent out to the next stage.

According to Figure 3.48, we will get V_P and V_N at the output respectively.

$$V_P = V_{cm} + \left(\frac{C_1}{C_1+C_2}\right)(V_{in} - V_{ref}) \quad (3\text{-}7)$$

$$V_N = V_{cm} - \left(\frac{C_1}{C_1+C_2}\right)(V_{in} - V_{ref}) \quad (3\text{-}8)$$

3.4.6.2. Mirrored Biasing

Since we have 16 channels working at the same time in one readout block, if we directly connect the bias of all the buffers together, the kickback of the signal on one bias will significantly influence another. A bias mirror circuit is implemented to avoid crosstalk on the bias [3.6]. We use a NMOS tuner as a master one, which is mirrored in every single channel to bias a PMOS tuner locally. The circuit is shown in Figure 3.49.

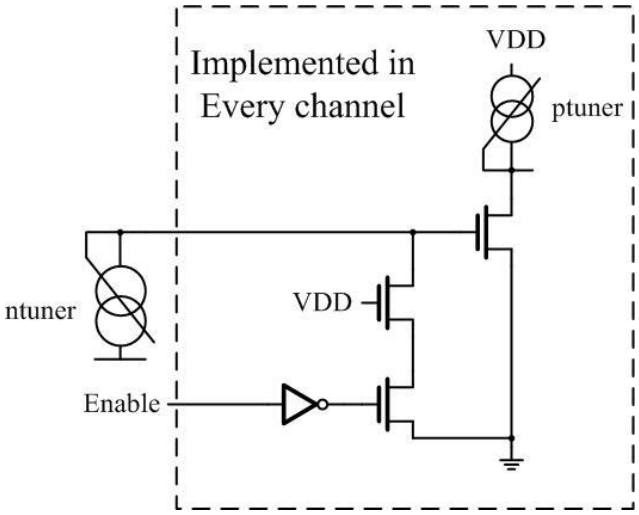


Figure 3.49: Schematic of bias mirror

With an Enable switch, we can simply disable the bias, consequentially disable the readout buffers. The Enable signal is provided by the ASPI. An additional NMOS is added in series with the Enable switch is to avoid large V_{DS} voltage generating photons. The function of the disabled bias is for power saving concern in half frame rate mode, which will be discussed in next chapter.

Due to the reason that the bond pad pitch is restricted to 300 um, it will result in that some channel's output can be close to the bond pad, while others are far away. As shown in Figure 3.50, the length of the wires in channel<1> is different from the one in channel<8>. To reduce the mismatch comes from the metal parasitic resistance and capacitance (RC), we put the final output buffers inside the IO rail, meaning the final buffer location will follow the same pitch with the bond pads. Then the mismatch will happen between the intermediate buffers and the final output buffers.

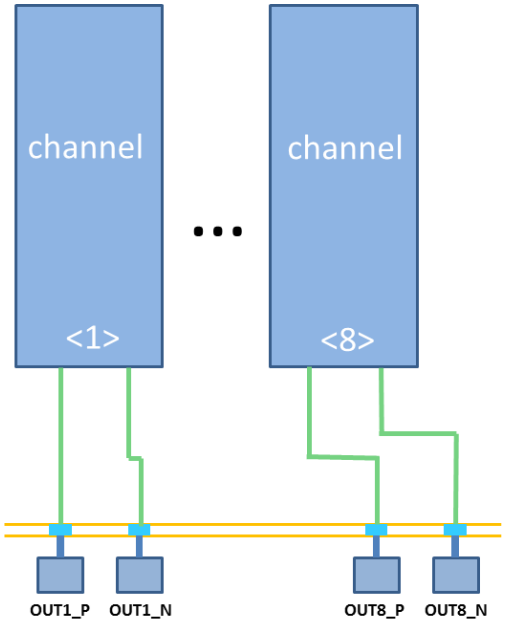
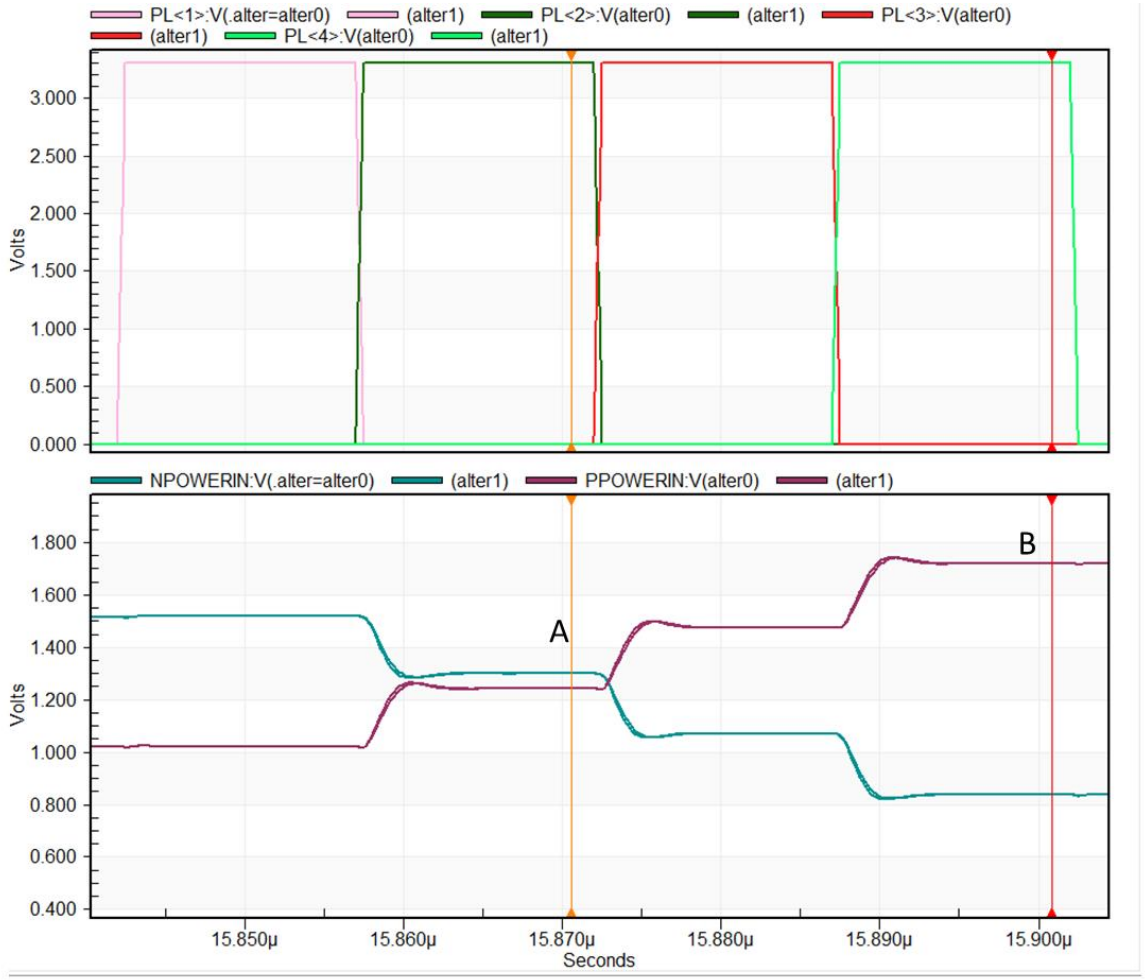


Figure 3.50: Mismatch of wires in channel readout

To prove the validity, we run the simulation for different pixel signals with both shortest wires and longest wires. The simulation result is shown in Figure 3.51.



		A	B
Shortest	NPOWERIN:V(.alter=alter1)	1.298871938056381	836.81407458921020m
Longest	NPOWERIN:V(.alter=alter0)	1.298870389643812	836.81244178621750m
Longest	PPOWERIN:V(.alter=alter0)	1.241720966909556	1.71862521727711
Shortest	PPOWERIN:V(.alter=alter1)	1.241720674039387	1.71862000271379

Figure 3.51: Simulation of channel readout with different parasitic RC

In Figure 3.51, there are two cases of simulation (alter0 and alter1).

- In alter0, the outputs of the channel are connected to the largest parasitic RC which calculated from the longest metal wire in layout.
- In alter1, the outputs are connected to the smallest parasitic RC.
- The traces in the top waveform are preset line signals (PL), the traces in the bottom waveform are the differential output signals of one channel.

The measured values at the bottom of Figure 3.51 prove that our buffers are strong enough to make the signals settle in time. The mismatch caused by wires is negligible.

3.4.7. Layout consideration

The layout of the readout block is most critical part since it is more space limited and there are high speed analog signals together with digital signals propagating through the whole block. A brief screenshot Figure 3.52 shows how the layout of one readout channel is arranged.

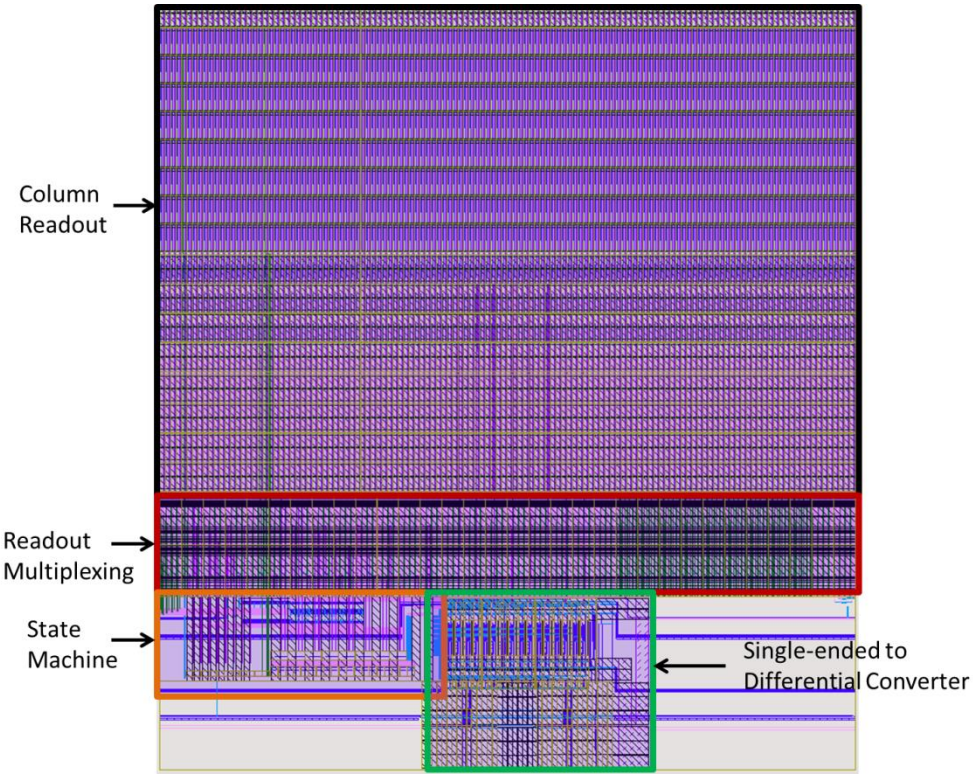


Figure 3.52: Layout of one readout channel

Another critical issue is the power routing of VDDA and VDDAO. We do the simulation to estimate the average current of them in 4K configuration, which is shown in Table 3-7. Results in 8K and 12K configurations are simply increased by a factor of 2 and 3.

Table 3-7: Average current of VDDA and VDDAO in readout block

Power supply	I_{DC} (mA)
VDDA/VSSA	327
VDDAO/VSSAO	313

From the table we can see that those power supplies have comparative high current flow. To avoid IR drops caused by metal parasitic resistance, we give priority to provide as many bond pads of power supplies as possible.

The width of the whole readout block is 26624 µm, which allows 88 bond pads maximum. Deducting the 32 output signals, digital signals and digital power supplies, the rest are 36 pads. For symmetry consideration, every two channels are chosen to share one set of VDDIO, VSSIO and VDDA, VSSA. The rest 4 pads for VPIX are also distributed symmetrically in the whole block.

So the finally layout overview of the readout block is shown in Figure 3.53. We take the readout block in the south as an example.

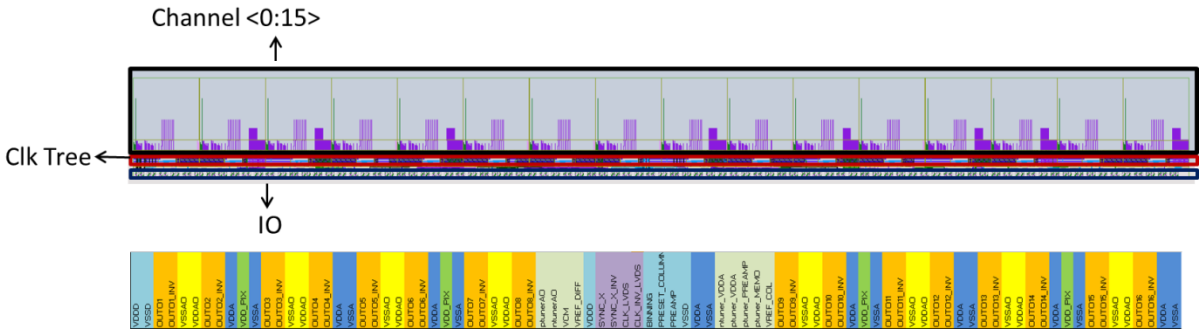


Figure 3.53: Layout of readout block

In Figure 3.53, at top is the layout of one readout block, it has 16 readout channels, one clock tree and IO. At bottom, the content of the IO is illustrated. The VDDIO/VSSIO is labelled in yellow. The VDDA/VSSA is in deep blue. The VPIX is in deep green. The differential output signals are in orange. The analog reference voltage and bias signals are in light green. The VDDD/VSSD is in light blue and the digital signals are in purple.

3.5.Chapter conclusion

In this chapter, the working principle, schematic and layout of the sub-circuits involved in every block are given. The validity of their function is proved by the results of functional simulations with parasitic RC. Some potential failures may happen in the sensor are also discussed.

In the next chapter, the top sensor system overview, functional simulation, noise analysis and top layout will be given. Moreover, some test structures will also be introduced.

3.6.References

[3.1] Dierickx, B., “High Dynamic Range the pixel standpoint,” ISSCC Forum on Image sensors, available at www.caeleste.be, Feb. 2008.

[3.2] Dierickx, B.; Dupont, B.; Defernez, A.; Fryer, M.; Jorden, P.; Walker, A.; Pike, A.; Jerram, P. and Pratlong, J., “Backside thinned, 2.5 e- RMS, BSI, 700fps, 1760x1760 pixels wave-front imager with 88 parallel LVDS output channels,” International Image Sensor Workshop, available at www.caeleste.be, Jun. 2011.

[3.3] “Address decoder”, available at http://en.wikipedia.org/wiki/Address_decoder, Jul. 2013.

[3.4] "Serial Peripheral Interface Bus", available at http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus, May. 2014.

[3.5] Downing, M.; Kolba, J.; Balarde, P.; Dierickx, B.; Defernez, A.; Dupont, B.; Feautrierd, P.; Fingera, G.; Fryer, M.; Gache, J.; Guillaumef, C.; Hubina, N.; Iwerta, O.; Jerram, P.; Jorden, P.; Pike, A.; Pratlong, J.; Reyesa, J.; Stadlerd, E. and Andrew Walker. A., “Backside-Illuminated, high QE, 3e- RoN, fast 700fps, 1760x1680 pixels CMOS Imager for AO with highly parallel readout,” Proc. SPIE 8453, High Energy, Optical, and Infrared Detectors for Astronomy V, 84530C, available at www.caeleste.be, Sep. 2012.

[3.6] "Chapter 11: The Current Mirror", available at http://wiki.analog.com/university/courses/electronics/text/chapter-11, Jan, 2014.

Chapter 4
Top Sensor architecture

In this chapter, the top view of the sensor arrangement is presented. First, the system timing diagrams and frame rate calculation are explained. The system functional and noise transient simulation is illustrated and some specifications calculation is discussed. Additionally, the test structures involved in the sensor are explained. Finally, the top layout of the whole sensor is given.

4.1.Top level design issues

4.1.1. Stitched blocks arrangement

Three different sensor formats can be realized with 4 “stitch” blocks. The three variations are illustrated in Figure 4.1.

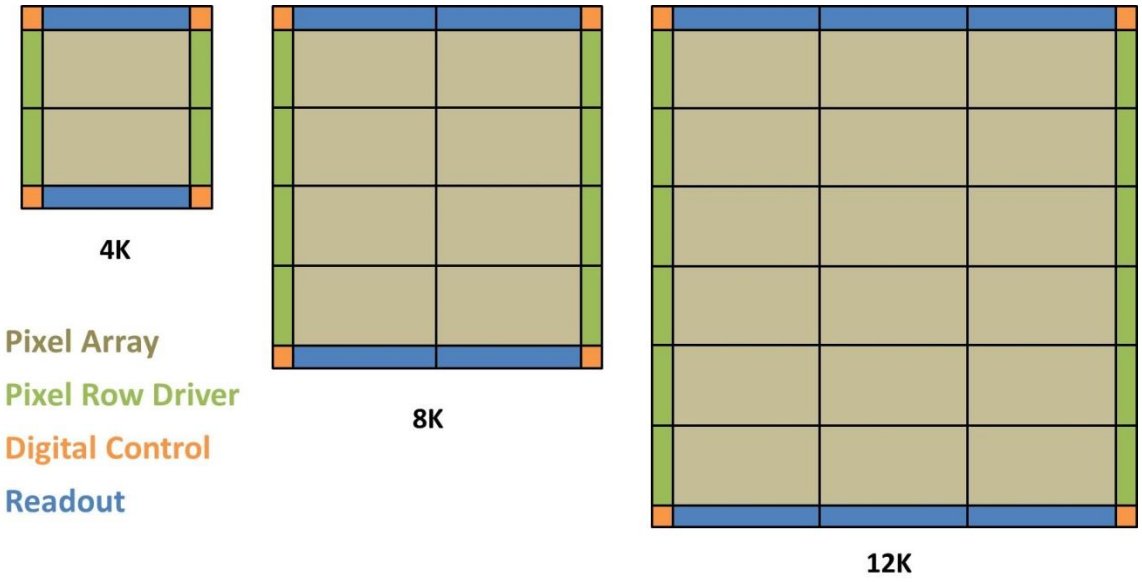


Figure 4.1: Stitched block arrangement in different configurations

In Figure 4.1, from left to right are the 4K, 8K and 12K configurations. The block in grey is the pixel array block and each one contains 4096*2048 pixels. The block in green is the pixel row driver block and it is located at the east and west side of the sensor. The block in orange is the digital control block and it is located at the 4 corners of the sensor. The block in blue is the readout block and it is located at the north and south side of the sensor.

4.1.2. Sensor operation timing diagrams and frame rate calculation

The timing diagram of the whole system in rolling shutter mode is shown in Figure 4.2 and the one in snapshot mode is shown in Figure 4.3. Note that those drawings are based on 4K configuration in HDR mode.

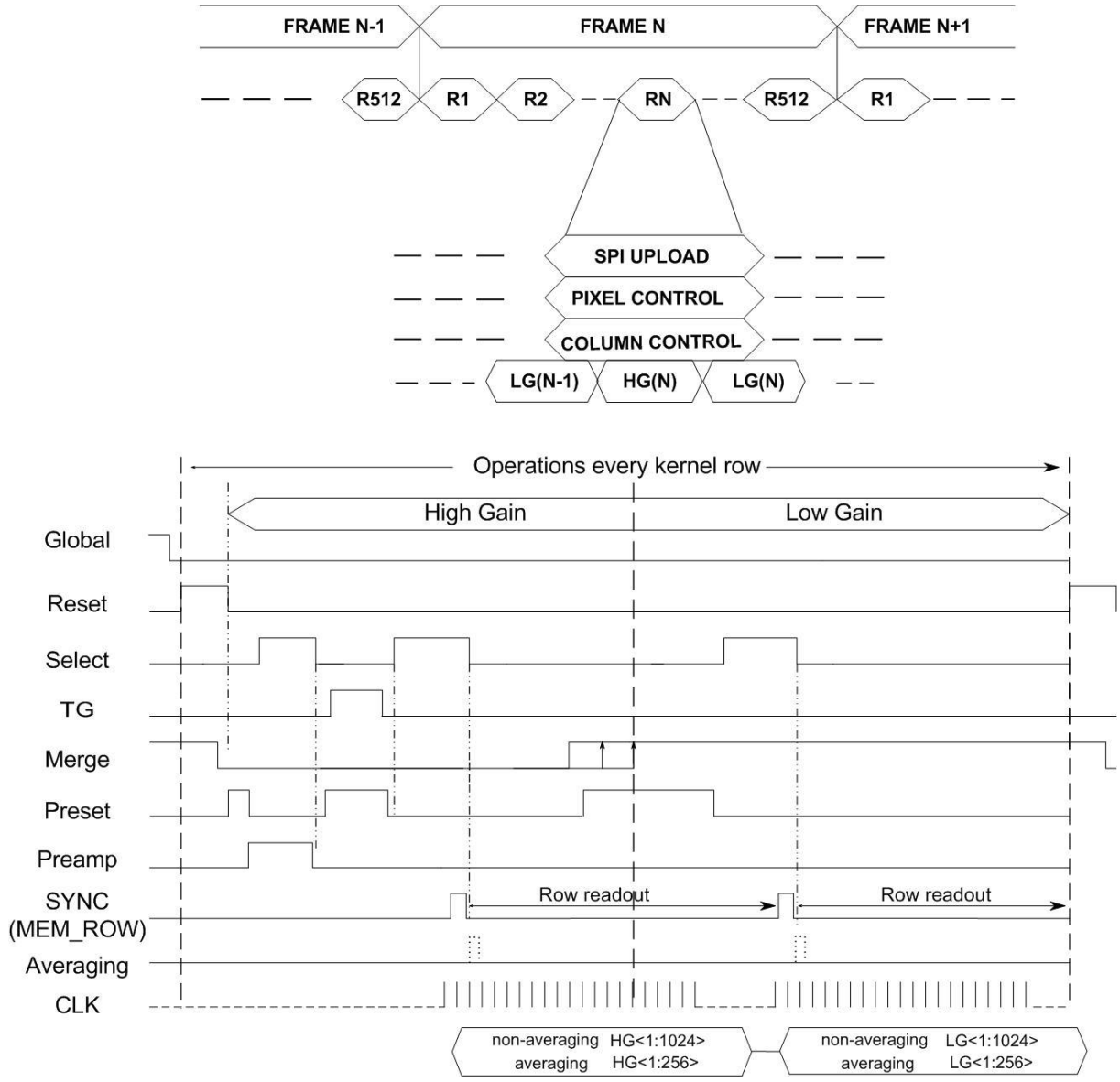


Figure 4.2: System timing diagram in rolling shutter mode

In Figure 4.2, at top, there are the operations in one frame. Taking 4K configuration as an example, it has 512 kernel rows (one kernel row has 8 rows). In rolling shutter mode, operation is based on every kernel row. So in the timing of every kernel row, we upload the SPI code to select one kernel row, and then do the pixel operation and readout operation. CDS is performed on-chip, so to get a HDR frame we need to read out both the high gain signals and the low gain signals. In LDR mode, we only read out the high gain signals. At the bottom of Figure 4.2 are the detailed pixel and readout operations during the timing of every kernel row.

The timing of pixel operation has been explained in Section 3.1.4. The Preset and Preamp signal are the signal to realize CDS on-chip, which has been discussed in Section 3.4.3.1. There is always one Preset pulse before one Select pulse, which aims to clean the former signal store on the pixel readout column bus. The SYNC signal is the sensor master synchronized reset signal, which is equal to the MEM_ROW signal in column readout. Between two MEM_ROW pulses, is the time for the signals belong to one kernel row being readout, which is labelled as “Row readout” in Figure 4.2.

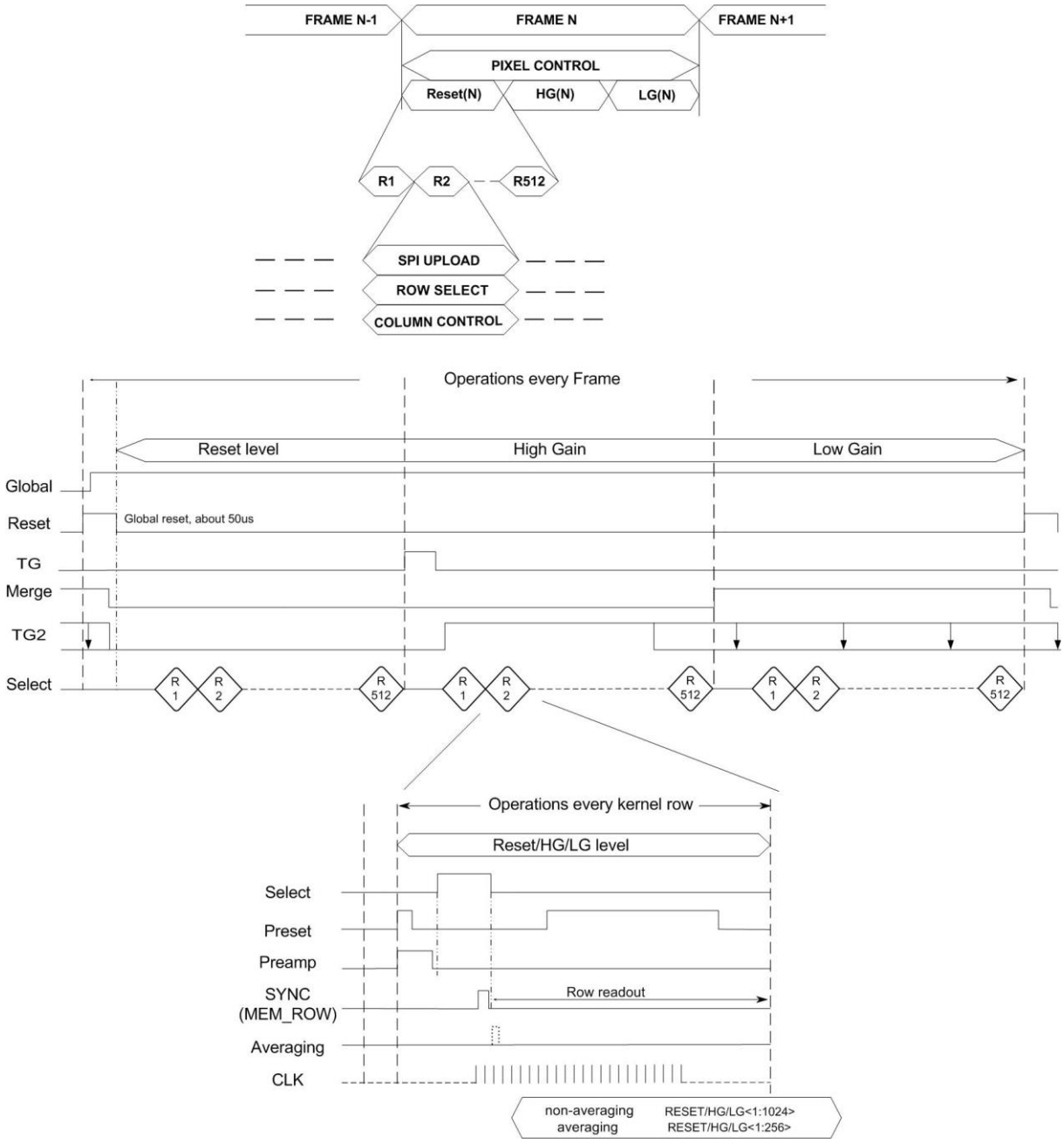


Figure 4.3: System timing diagram in snapshot mode

In Figure 4.3, at the top, there are the operations in one frame. In snapshot mode, pixel operation is based on every frame except row selection of pixel readout. Since CDS is done off-chip, to get a HDR frame, we need read out the reset frame, the high gain frame and the low gain frame. The SPI still need to upload every kernel row to complete the row selection. The details of the timing of the pixel operation are shown in the middle of Figure 4.3. After one kernel row has been selected, the readout operation is done as shown at the bottom of Figure 4.3. As the pixel operation in LDR snapshot mode has been discussed in Chapter 3, it is not included here again.

The calculation of frame rate in various modes and configurations are shown in Table 4-1 and Table 4-2.

In Table 4-1 and Table 4-2, the “pass” refers to one time read out from pixel to output bond pad. Since in global shutter mode we always do CDS off-chip and we need to readout the reset signals, there are always one pass more than the one in rolling shutter mode.

The frame overhead time (FOT) here means an extra time needed every frame. It is due to that when we do global reset with current limiter on, limited current makes the reset takes longer time which is around 50 µs.

The line blanking time (LBT) also can be understand as row dead time, meaning we cannot do column readout at that time. It is equal to the time when the MEM_ROW switch is closed, which stores signals on the accumulation capacitors in the columns.

The extra multiplex kernels are some test pixels. To avoid breaking the pixel block symmetry for the stitching concern, we put them in the left and right pixel row driver block, next to the boundary of the real pixel array, but isolated by guard pixels in-between.

The minimum row time is 7.5 µs which is determined by the sum of the pulse widths of the pixel operation signals.

The row time is defined as

$$MAX\{MIN(row\ time), LBT + \frac{(\#\ kernels\ X \times kernel\ size\ X)}{f_{readout}}\} \quad (4\text{-}1)$$

The frame time is

$$Frame\ time = (FOT + row\ time\ \times \#\ kernels\ Y) \times \#\ passes \quad (4\text{-}2)$$

$$Frame\ rate = \frac{1}{Frame\ time} \quad (4\text{-}3)$$

Based on these equations, we can get the frame rate calculation results in Table 4-1 and Table 4-2.

Table 4-1: Frame rate calculation of LDR mode

configuration	4K	4K	8K	8K	12K	12K
shutter	global	global	rolling	rolling	rolling	rolling
averaging	N	Y	N	Y	N	Y
x pixels	256	128	256	128	256	128
y pixels	2048	1024	4096	2048	6144	3072
kernel size x	1	1	1	1	1	1
kernel size y	4	2	4	2	4	2
extra multiplex kernels	7	7	7	7	7	7
# kernels x	263	135	263	135	263	135
# kernels y	512	512	1024	1024	1536	1536
pass(es)	2	2	1	1	1	1
LBT[µs]	0.5	0.5	0.5	0.5	0.5	0.5
$f_{readout}$ [MHz]	65	65	65	65	65	65
row time[µs]	16.7	7.5	16.7	7.5	16.7	7.5
FOT [µs]	50	50	0	0	0	0
frame time[s]	1.72E-02	7.73E-03	1.71E-02	7.68E-03	2.56E-02	1.15E-02

frame rate	**58**	**129**	**59**	**130**	**39**	**87**

Table 4-2: Frame rate calculation of HDR mode

configuration	4K	4K	8K	8K	12K	12K
shutter	global	global	rolling	rolling	rolling	rolling
averaging	N	Y	N	Y	N	Y
x pixels	256	128	256	128	256	128
y pixels	2048	1024	4096	2048	6144	3072
kernel size x	1	1	1	1	1	1
kernel size y	4	2	4	2	4	2
extra multiplex kernels	7	7	7	7	7	7
kernels x	263	135	263	135	263	135
kernels y	512	512	1024	1024	1536	1536
pass(es)	3	3	2	2	2	2
LBT[µs]	0.5	0.5	0.5	0.5	0.5	0.5
$f_{readout}$ [MHz]	65	65	65	65	65	65
row time	16.7	7.5	16.7	7.5	16.7	7.5
FOT [µs]	50	50	0	0	0	0
frame time[s]	2.57E-02	1.16E-02	3.42E-02	1.54E-02	5.13E-02	2.30E-02
frame rate	**39**	**86**	**29**	**65**	**20**	**43**

4.1.3. System simulation

4.1.3.1. Functional simulation

To make the size of the test bench acceptable for our software environment, we need to reduce the schematic netlist. We keep the pixel row driver unit and digital control block. The pixel array is reduced to contain only one kernel of pixels, and the readout block is shrunk to be 8 readout columns, one readout channel and the state machine. A brief drawing of the test bench constitutions is shown in Figure 4.4.

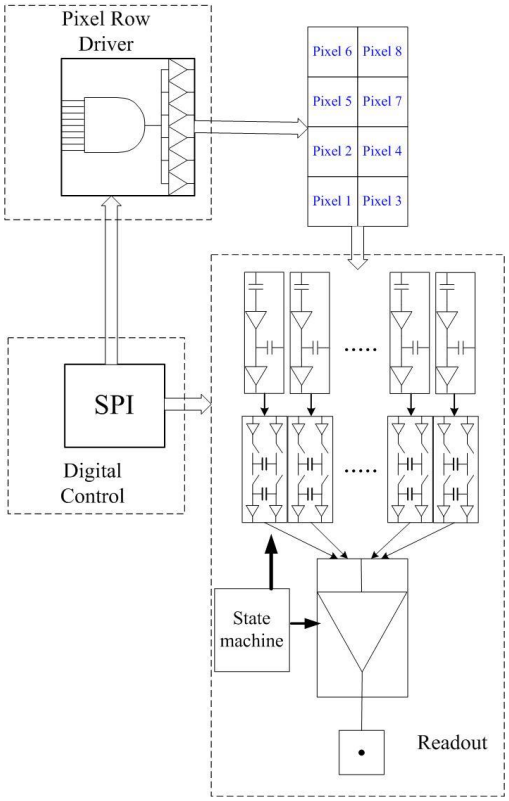


Figure 4.4: Diagram of system after netlist reduced

There are some reference voltages we need to consider in the simulation.

- V_{ref_column}: reference voltage for CDS in the column readout.
- V_{dark}: dark level voltage. It is the voltage value used to preset 8 readout lines. The same value as V_{ref_column}.
- V_{ref_s2d}: reference voltage for single-ended to differential converter.
- V_{cm_s2d}: Common mode voltage for the S2D output differential signals, value must fit the ADC input range requirement.

The value of those four reference voltages are list in Table 4-3.

Table 4-3: Reference voltage values used in simulations

Name of Reference	Voltage [V]
V_{ref_column}	1.8
V_{dark}	1.8
V_{ref_s2d}	1.3
V_{cm_s2d}	1.3

In the simulation, we first clock SPI to select one kernel row. Then do the pixel operation, readout operation and readout multiplexing to get the final differential output signals. The system simulation results are shown in Figure 4.5, Figure 4.6, Figure 4.7 and Figure 4.8.

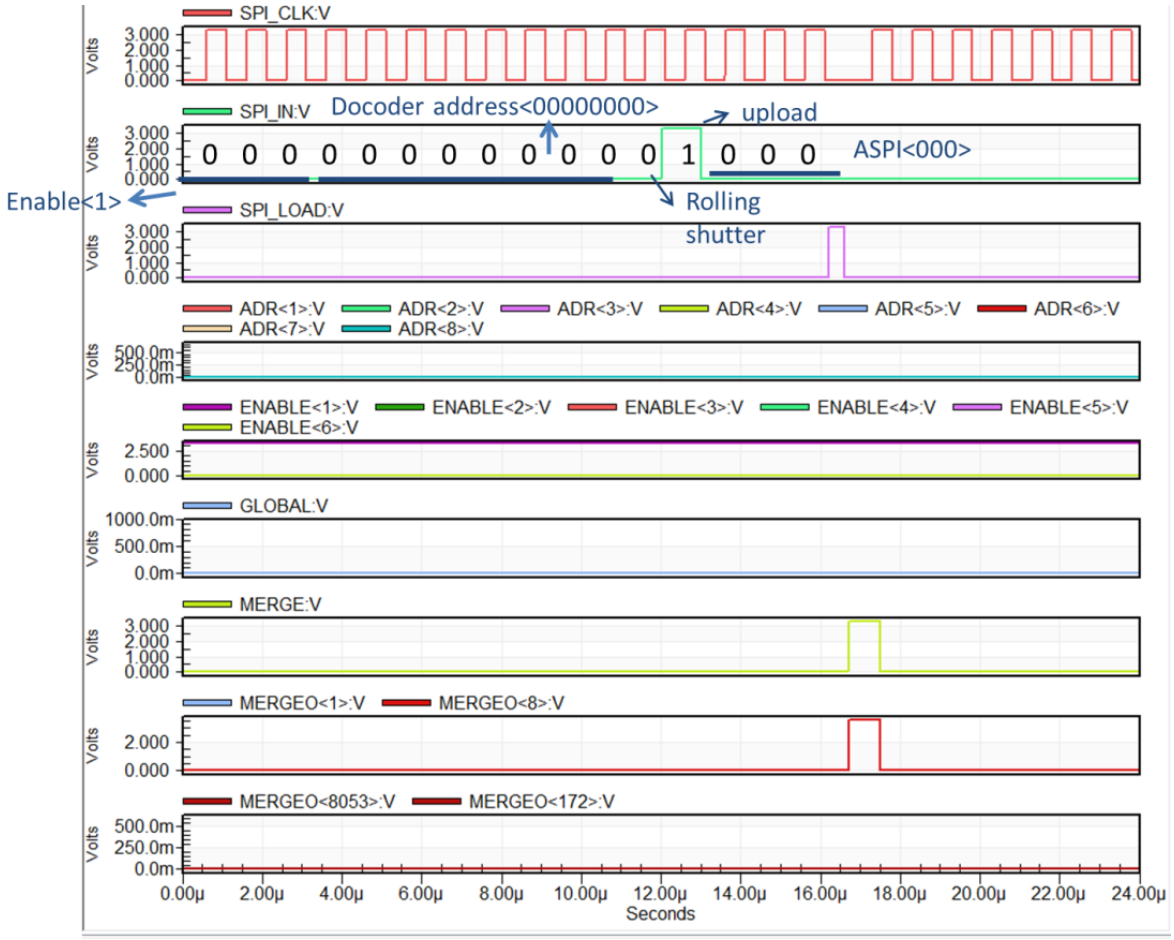


Figure 4.5: Simulation result of SPI selecting one kernel row of pixel

Figure 4.5 shows the simulation of SPI selecting one kernel row of pixel.

- The first three waveforms at the top are the SPI clock signal (SPI_CLK), SPI input data (SPI_IN) and SPI data load signal (SPI_LOAD).
- The traces in the next three waveforms are the address (ADR<1:8>), enable (ENABLE<1:6>) and global signal of the pixel row drivers.
- The traces in the last three waveforms are the input MERGE signal of the pixel row driver and the corresponding outputs of the MERGE signals.

From 0 µs to 16.7 µs, SPI uploads an address, enable and global signals to select one specified row. According to the SPI_IN curve, we choose to select the first row in the first pixel driver block, whose address is <0000000>.From the ENABLE<1:6> curves, we can see that only ENABLE <1> is high, other ENABLE <2:6> is low, meaning only one pixel row driver block is selected.

Moreover, after the SPI_LOAD pulse, we give a pulse of input MERGE signal, and take the output MERGEO<1> of row <1> and MERGEO<8> of row <8>, to compare with the output of row <172> and row<8053>. Note that 8 rows belong to one kernel row. We can see only the kernel row with the correct address (e.g. here is row<1:8>) can get the pixel operation signals (e.g. here is MERGE signal). The simulation proves the validity of the system row selection, which means there is no conflict in addressing one row.

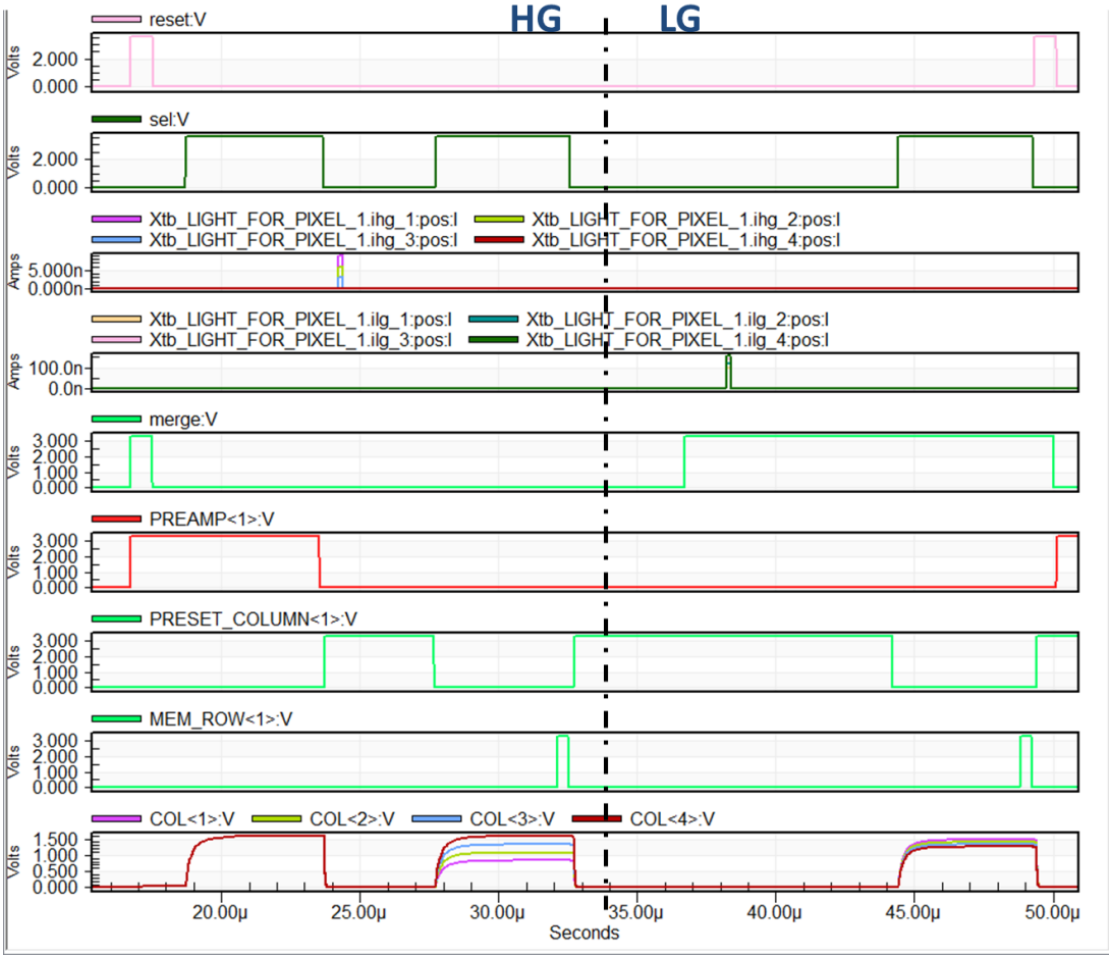


Figure 4.6: Simulation result of pixel readout

Figure 4.6 shows the pixel readout part.

- The first trace is the pixel reset signal.
- The second trace is the select signal of pixel readout.
- The traces in the third and fourth waveforms are the current pulses used to model the photocurrent transferred by transfer gate in the pixel. We take 4 pixels as examples in Figure 4.6. In HG mode, we give 9k e-, 6k e-, 3k e- and 0 e- respectively into pixel<1:4>. In LG mode, we give 100k e-, 120k e-, 140k e- and 160k e- respectively into pixel<1:4>.
- The next waveform is the MERGE signal.
- The sixth waveform is PREAMP signal. The pulse of the PREAMP signal is for CDS, to offer a reference voltage V_{ref_column}.
- The seventh waveform is PRESET_COLUMN signal. The pulse of the PRESET_COLUMN signal is to preset the column input to VSSA.
- The eighth waveform is MEM_ROW signal. The pulse MEM_ROW is to store the column signals on the accumulation capacitors in each column, and then they will be multiplexing readout on 8 lines.
- The outputs of 4 columns COL<1:4> are shown in the last four traces at the bottom of Figure 4.6.

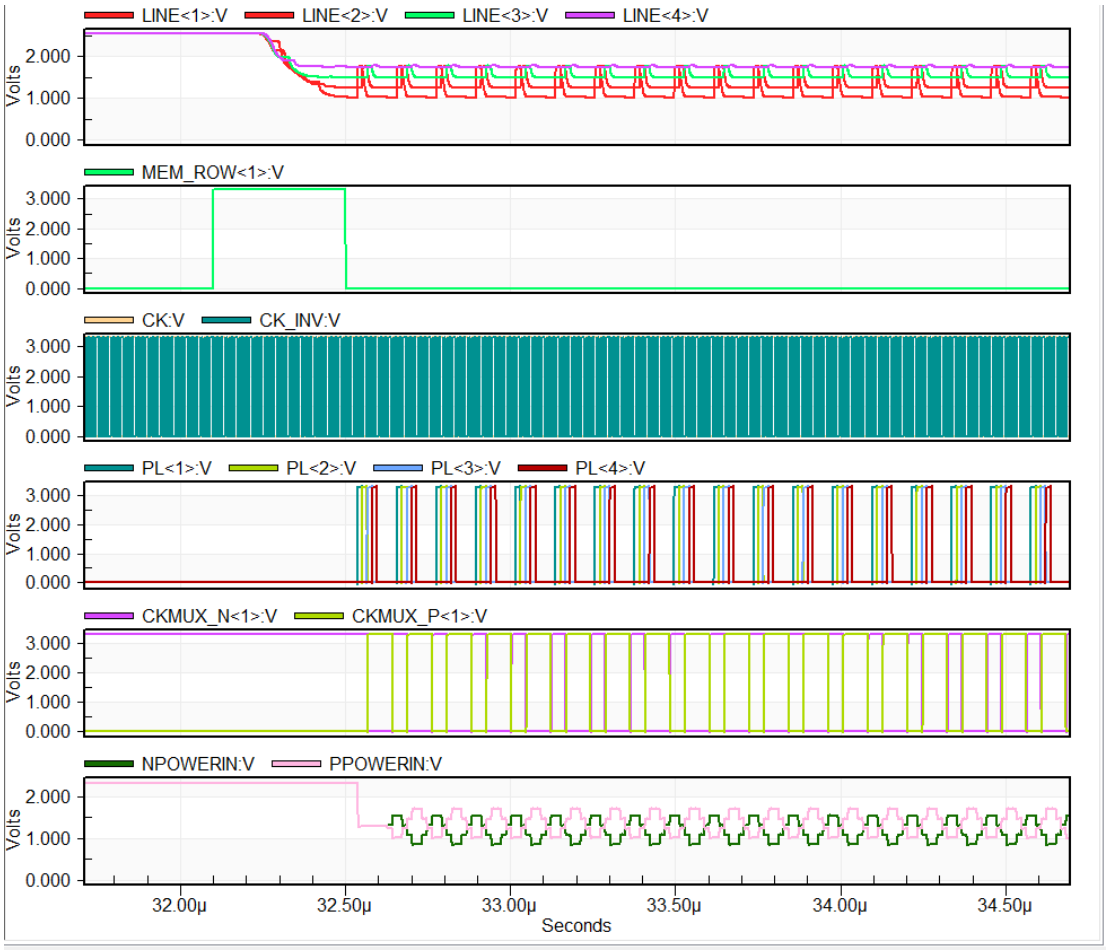


Figure 4.7: simulation result of channel readout HG signals

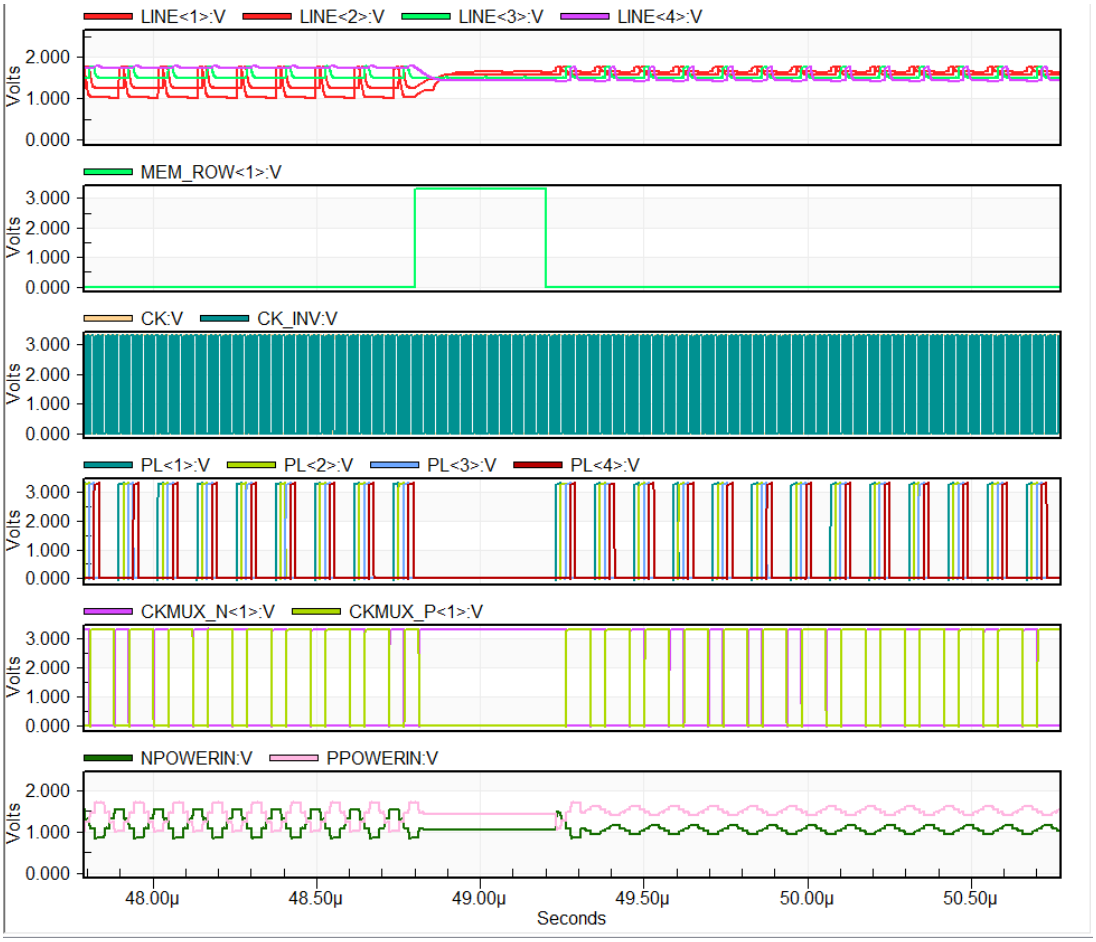


Figure 4.8: simulation result of channel readout LG signals

Figure 4.7 and Figure 4.8 show the channel readout for HG signals and LG signals. We take line<1:4> as examples.

- The four traces in the first waveform are the signal on 4 readout lines.
- The second waveform MEM_ROW is the same signal as the system SYNC to synchronized reset the DFFs in state machine and multiplexing.
- The third waveform shows the clock of the system.
- The traces in the fourth waveform are PL<1:4> signals. After signals are sent to the 8 readout lines, next stage is the single-ended to differential signal converter. The PL signal is the line preset, which preset the line to a dark signal voltage V_{dark}.
- The traces in the fifth waveform are CKMUX_P and CKMUX_N signal to realize the sample and hold in S2D stage, which are the ϕ_1 and ϕ_2 in single-ended to differential converter as explain in Chapter 3.
- The last two traces at bottom are the output differential signals called PPOWERIN and NPOWERIN.

In Figure 4.8 we can see that, before the pulse of MEM_ROW, is the readout of the HG signals and after is the readout of the LG signals.

4.1.3.2. Readout noise simulation

The main noise contributors can be categorized as Table 4-4 shows. They are roughly separated by three different locations in the system. The detail descriptions of each noise are listed as follows.

Table 4-4: Noise sources in the system

Location	Noise type	Description
In the pixel (on-chip)	Pixel reset kTC noise	Largely cancelled by CDS [4.1]
	Pixel SF 1/f noise	Partly cancelled by CDS [4.1]
	Dark current shot noise (DCSN)	Depending on integration time. Cooling system helps to reduce. DSNU can be partly reduced by software calibration
	Dark signal non-uniformity (DSNU)	
	Photo shot noise (PSN)	Depending on signal level, PRNU can be partly reduced by software calibration
	Photo response non-uniformity (PRNU)	
	Pixel fixed pattern noise (FPN)	Offset mismatch in pixel can be reduced by CDS [4.2]. FPN can be partly reduced by software calibration
In the periphery (on-chip)	Column fixed pattern noise (FPN)	
	Readout stage kTC, thermal and 1/f noise	Bandwidth reduction helps, can be predicted by noise transient simulation
In the system (off-chip)	ADC noise	Based on the ADC type chosen
	Power supply noise	Depending on the PCB design
	Substrate noise	Depending on the PCB design

Thanks to CDS and software calibration, the main on-chip noise sources we need focus on are the 1/f noise of SF and kTC noises in the readout path.

For the 1/f noise of SF, it can be estimated by

$$V_{\frac{1}{f},RMS} = \sqrt{\frac{KF \times ln\frac{4f_{max}}{f_{min}}}{W \times L}} \tag{4-4}$$

Where KF is a factor dominated by the fabrication process, f_{max} and f_{min} is the frequency range and W and L are the dimension of the SF MOSFET.

For the noise in the readout path, the kTC noise of the sampling capacitors contributes the most [4.3]. The kTC noise can be calculated as

$$V_{noise,RMS} = \sqrt{\frac{kT}{C}} \tag{4-5}$$

Where k = 1.38 × 10^{-23} J/K is Boltzmann's constant, T is the temperature and C is the sampling capacitance [4.1].

There are three capacitors generating noise in our readout path. One is the CDS capacitor, second is the sampling capacitor in each column, and the last one is the sampling capacitor in single-ended to differential converter stage.

To predict the temporal noise, the noise transient simulation is based on one single signal readout path, which is briefly illustrated in Figure 4.9. Only one pixel, one readout column and one set of S2D converters and output buffers are involved in the simulation.

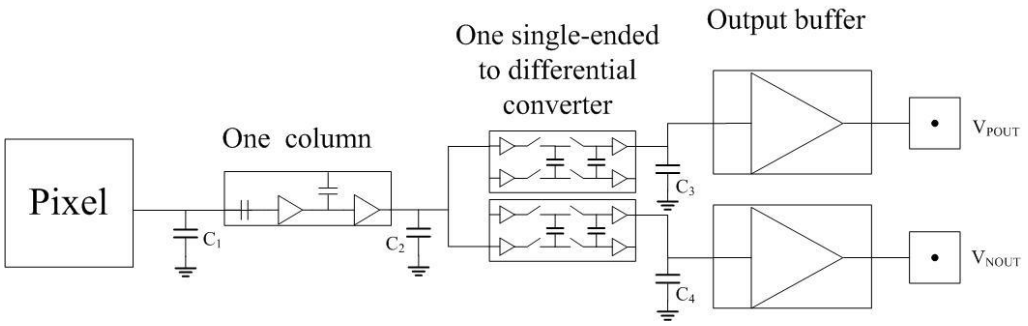


Figure 4.9: Diagram of schematic used in noise simulation

To take the worst case into consideration, we add parasitic capacitors C_1-C_4 which are generated by the metal connections. They are roughly calculated by 1 pF/cm and the length is measured from the longest connection existed in the layout. C_1 comes from the column load wire, C_2 is the line wire and C_3 and C_4 are the wires connect the output signals from single-ended to differential converters to the inputs of the buffers in IO.

The simulation result is list in Table 4-5 below. As discussed in Chapter 3, users can switch the gain in the S2D converter stage between 1 and 0.55, which corresponds to a gain of 1.75 and 1 to the signal range of the system.

Table 4-5: Result of noise transient simulation

Mode	$Noise_{RMS}$ [μV]
Gain=1.75	585
Gain=1	423

Our 12-bit ADC has a maximum 2 V peak to peak range. From the ADC datasheet, we know the ADC SNR is 71.5 dB and input referred RMS noise is 0.4 LSB at 65 MHz. According to our pixel whose range is 1 V, and the gain of range is 1.75, so the output signal range of our sensor is 1.75 V.

The total RMS noise of the system is referred to

$$Noise_{RMS}\ System = \sqrt{(Noise_{RMS}\ ADC^2 + Noise_{RMS}\ sensor^2)} \qquad (4\text{-}6)$$

Table 4-6 shows the system noise specification based on the selected ADC. Based on equation (4-6), we can use the sensor RMS noise and ADC RMS noise to calculate the total system RMS noise at the input of the ADC.

Table 4-6: System noise specification

ADC LSB [V]	4.883E-4
ADC RMS noise [V]	1.953E-4
Sensor RMS noise [V]	5.85E-4
Total RMS noise [V]	6.167E-4
Total RMS noise in LSB	1.263

After we get the total system RMS noise at the input of the ADC, we can calculate the system SNR.

The SNR of ADC is competed by [4.4]

$$SNR = 20\,log\left(\frac{rms\ Signal}{rms\ Noise}\right) = 6.02\text{ENOB} + 1.761\ \text{dB} \quad (4\text{-}7)$$

Where ENOB is the effective number of bits.

Since the peak to peak range of the ADC is 2 V, but we only use 1.75 V, so the SNR becomes

$$SNR = 20\,log\left\{\left(\frac{rms\ Signal_{p-p}}{rms\ Noise}\right) \times \left(\frac{rms\ Signal}{rms\ Signal_{p-p}}\right)\right\} \quad (4\text{-}8)$$

$$= 6.02ENOB + 1.76 + 20\,log\left(\frac{1.75}{2}\right) \quad (4\text{-}9)$$

From Table 4-6, we know the total input referred RMS noise is 1.263 LSB, so to get the ENOB here is using the 12 bits to deduct the input referred RMS noise, which gives ENOB= 10.74 bits.

Hence, we can get an estimated SNR of the system which is about 65 dB.

4.1.4. Design for test

Except the Mixed Boundary Scan (MBS) access to some critical node in the data path, which has been discuss in chapter 3, there are some other test structures.

Quantum efficiency test

The quantum efficiency (QE) test structure is for the measurement of QE. It is located in the empty corner in pixel row driver block. The pixel of QE test is almost the same as the real functional pixel we use, only difference is the transfer gate (TG) in QE pixel are tied to VPIX to make it permanent on. The test plan of QE is shown in Figure 4.10.

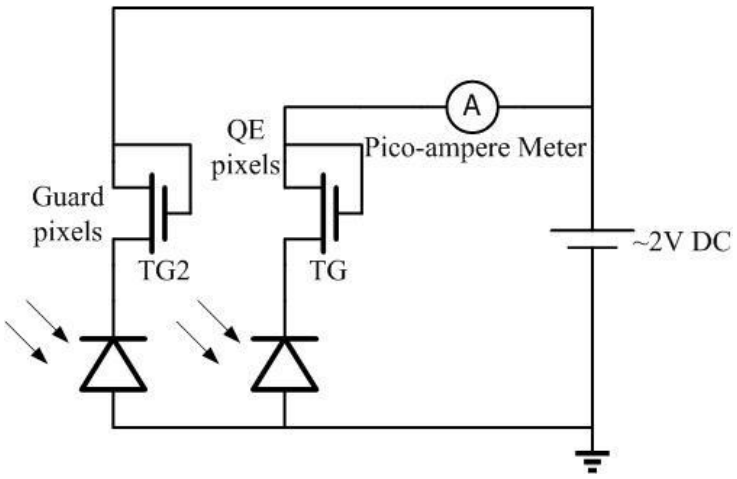


Figure 4.10: Diagram of QE test structure

According to Figure 4.10, we will measure the photocurrent or particle-generated current by a pico-ampere meter. The calculation of QE is given by [4.5]

$$QE = \frac{R_\lambda}{\lambda} \times \frac{hc}{e} \quad (4\text{-}10)$$

Where λ is the photo wavelength, h is the Planck constant, c is the speed of light in a vaccum and e is the elementary charge. R_λ is the responvity [A/W], which can be calculated by

$$R_\lambda = \frac{\frac{I_{photo}}{Area}}{P_\lambda} \quad (4\text{-}11)$$

Where I_{photo} is the measured photocurrent, "Area" is the effective area of the QE structure and P_λ is the intensity of incident light.

Stand-alone test

Three kinds of stand-alone test are involved in this sensor.

- A single stand-alone pixel.
- A break out pixel is used to let us access every node inside pixel to check the transistors, FD, and barrier.
- A sheet resistance test of the pinned diode structures are used to measure the "depletion voltage". It includes pinned diodes, with larger or smaller width, some with a transfer gate on top and various combinations of implants involved.

The layout of those test structure is shown in Figure 4.11.

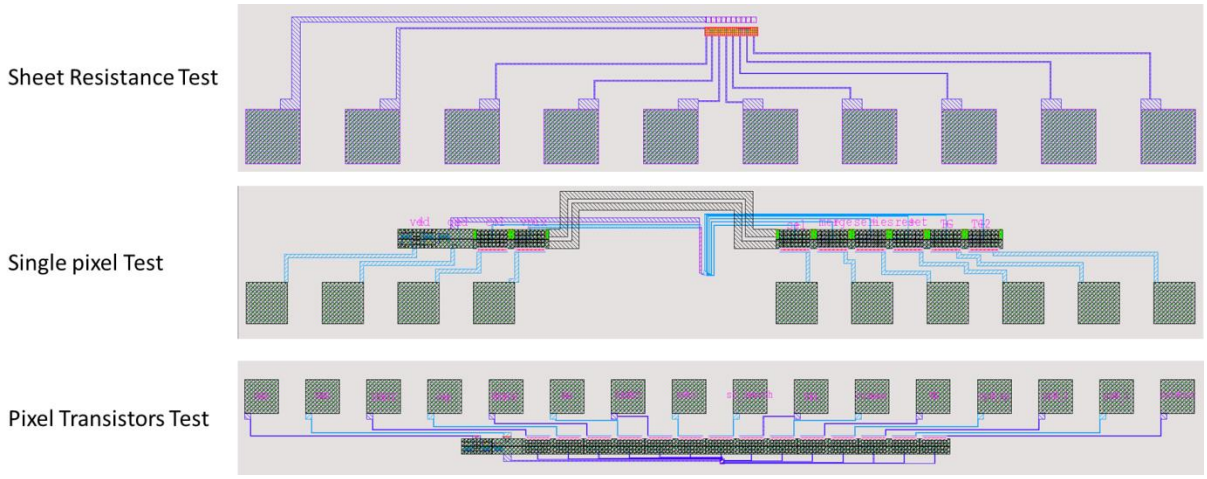


Figure 4.11: Layout of the stand-alone test structure

Linearity test

The linearity test structure is shown in Figure 4.12. It offers the possibility to measure the linearity of the analog readout circuitry independent of the pixel. It is located at the beginning of column readout.

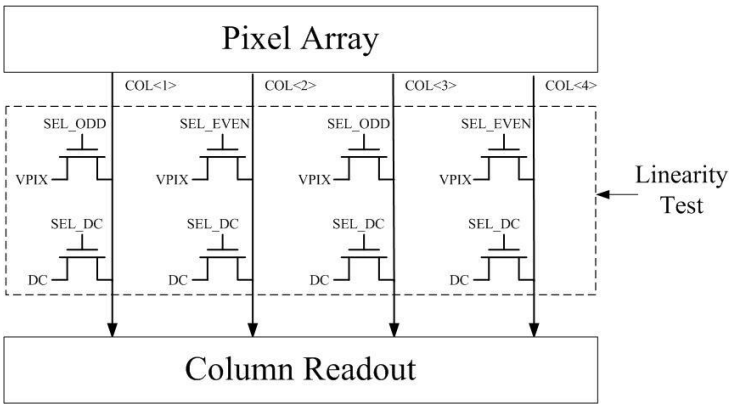


Figure 4.12: Diagram of linearity test structure

To realize the linearity test, we need at least two different DC signals. We take one from VPIX, another one is a DC signal from bond pad. Since a short between neighboring column wires is the most potential failure mode, either odd or even column wires are possible to be switched to VPIX.

4.1.5. Half frame rate mode

To achieve the possibility of lower power consumption, our sensor offers a half frame rate mode. To avoid asymmetry, the half frame rate mode is implement in the way that all odd channels and all even channels can be disable by two signals from ASPI respectively. As Figure 4.13 shows, they are called Enable odd and Enable even. Thus we can keep all the channels identical which easier our life in a stitched design. And the output of two neighboring channels are shorted on the PCB board and connected to the same ADC input.

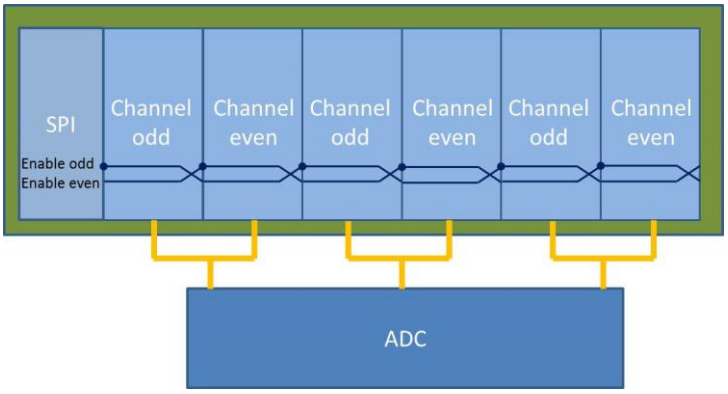


Figure 4.13: Diagram of half frame rate implementation

4.2.Top layout

The size of the sensor is shown in Table 4-7.

Table 4-7: Chip size in different configuration

sensor	Width[mm]	Height[mm]
4K	30.922	31.666
8K	57.547	58.29
12K	84.17	84.914

The layout of the whole sensor in 4K configuration is shown in Figure 4.14.

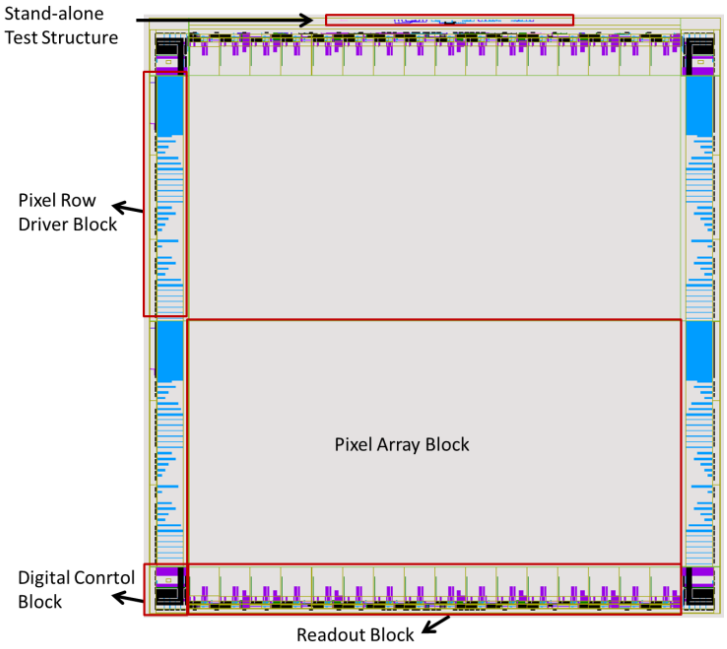


Figure 4.14: Top layout in 4K configuration

4.3.Chapter conclusion

In this chapter, the top overview of the sensor is explained. The system timing diagrams in both global shutter and rolling shutter mode are shown. The system simulation, noise analysis and top layout are given. Some test structures are illustrated.

In the next chapter, there are the conclusion and future work of this thesis.

4.4.References

[4.1] Theuwissen, A. J. P. and Charbon, E., ET4390 Imaging sensors Lectures, Delft University of Technology, 2013.

[4.2] El Gamal, A., 392b Lectures and Classnotes, Stanford University, 2001.

[4.3] Theuwissen, A. J. P., "Solid-State Imaging with Charge-Coupled Devices." Norwell, MA: Kluwer, 1995.

[4.4] “Quantization (signal processing)”, available at http://en.wikipedia.org/wiki/Quantization_(signal_processing), May. 2014.

[4.5] “Quantum efficiency”, available at http://en.wikipedia.org/wiki/Quantum_efficiency, June. 2014.

Chapter 5
Conclusion

In this thesis work, I, as a part of the Caeleste team, contributed to the design of a stitched HDR particle sensor. The features of this sensor are:

- A 92 dB high dynamic range is achieved by an extra MIM capacitor in pixel.
- High resolution is realized with stitching technology. The sensor can be made in variations of 4K*4K, 8K*8K or 12K*12K pixels.
- The column CDS reduces the noise.
- High speed readout at a frequency up to 65 MHz, over 64 or more parallel outputs.

5.1.My contributions to this work

My contributions were, amongst others:

- The major part of schematic design in the pixel row driver block and digital control block. Some part of the sub-circuits schematic design in the readout block.
- Part of the functional verification of sensor schematic design.
- The major part of the layout design and the verification of the whole layout using LVS and DRC.
- Help with the realization of stitched architecture and risk analysis during the design.
- Making datasheet and help with the test plan.

5.2.Future work

The tape-out of the chip was approved by the foundry in May, 2014. Wafers are expected to be back in three month. The PCB test system is ready, and the test of this chip will be executed as soon as it comes back. There are still some possibilities for future improvements.

Firstly, in a mature design, there should be no strict power-on sequence or timing sequence. However, in this actual design, a specified start-up sequence of power supplies is required. Amongst the reasons are that the current limiter must be programmed before the VPIX can be powered on. Thus, VDDD needs to be first powered on, and we send data over SPI to make current limiter work. An improved implementation of the current limiter or a built-in test to get the detailed value of power on peak current to improve the layout might help us to get rid of the complicated power up sequence.

Secondly, after we do the test on the real silicon, we can better understand how much the extra MIM capacitor increases the dynamic range. Meanwhile, the noise in the present design is about 1 LSB for the selected ADC. Based on better understanding of the noise mechanisms, one can think about methods to further reduce it to be less than 1 LSB.

Finally, the present design doesn't have a way to test if the sensor is alive or not in the dark. In other Caeleste image sensors, grey scale pixels are placed at the edges of the pixel array. It is not possible in the present sensor as its pixel matrix is part of a stitched block. To figure out a smart way that does

not break the symmetry in stitching while guarantee the validity of sensor function in the dark deserves some thinking.

List of Acronyms

APS	Active Pixel Sensor
ASPI	Addressable Serial Parallel Interface
CCD	Charge-coupled Device
CDS	Correlated Double Sampling
CMOS	Complementary Metal-Oxide-Semiconductor
DCSN	Dark Current Shot Noise
DR	Dynamic Range
DSNU	Dark Signal Non-Uniformity
FD	Floating Diffusion
FOT	Frame Overhead Time
FPN	Fixed Pattern Noise
HDR	High Dynamic Range
HG	High Gain
LBT	Line Blanking Time
LDR	Low Dynamic Range
LG	Low Gain
MBS	Mixed Boundary Scan
PPD	Pinned Photodiode
PRNU	Photo Response Non-Uniformity
PSN	Photo Shot Noise
QE	Quantum Efficiency
Q_{FW}	Full-Well Charge
S2D	Single-ended to Differential Signal Converter
SD	Sense Diffusion
SNR	Signal to Noise Ratio