PIXEL ADC DESIGN FOR HYBRID CMOS IMAGE SENSORS

Cheng Ma (1542575)

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

Submission to

The Faculty of Electrical Engineering

Mathematics and Computer Science

In Partial Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

In Electrical Engineering

Delft University of Technology

August 2010

COMMITTEE MEMBERS:

prof. dr.ir. Albert J.P. Theuwissen

dr. Ryoichi Ishihara

dr. ir. Michiel Pertijs

dr. ir. David San Segundo Bello

ACKNOWLEDGMENTS:

I would like to acknowledge a lot of people that make my study in TU Delft and IMEC a cherished memory.

First of all, I want to express my sincere gratitude to prof. dr. ir. Albert J.P. Theuwissen, my supervisor in TU Delft for his support of my thesis and the warm encouragement during the most difficult time of the thesis work.

I also want to thank prof. Chris van Hoof for giving me the opportunity for this interesting research work, and providing me the funding and great research environment.

David San Segundo Bello, my daily supervisor in IMEC, gave me continuous support and help during my research work and the writing of the thesis. Thanks very much for his encouragement when I was depressed, suggestions and guidelines when I got stuck. He is always willing to share his knowledge and experience. He is a great supervisor and friend.

Besides my supervisors, I would very much like to give my thanks to Padmakumar Ramachandra Rao, Koen De Munck and Kiki Minoglou for sharing their knowledge on the sensors and also the 3D stacking concept with me. They gave me a lot of inspirations on the working of my design.

I also would like to thank Nick van Helleputte, Munir Abdalla Mohamed, Junaid Aslam, Francesco Cannillo, Lino Delizia, Adi Xhakoni and all the other colleagues in the group for sharing their knowledge on the designing tools and their experience on the circuits. It was a great pleasure to work and study with them.

In addition, I want to thank all my friends in IMEC: Lu Zhang, Zhichao Lv, Guangyu Sun, Yan Li, and Kangnian Zhao for their support on my thesis and the happiness life staying with them.

And I also would like to thanks all my friends in Delft: Lin Qi, Yinan Wang, Shuoxi Chen, Yunlong Hou, Xiaobo Deng, Ronghui Zhang, Jie Shen. Thanks for the happiness with you guys.

Finally, I would like to express my deepest gratitude to my beloved family, my parents. Their support, understanding and love give me the confidence to overcome all the difficulties that I have faced.

June 2010

ABSTRACT

This thesis presents the design of a pixel level analog-to-digital converter (ADC) circuit for hybrid CMOS image sensors. There are several methods to increase the dynamic range of the sensor through the readout algorithm. The multiple sampling method can increase the dynamic range (DR) without a loss of signal-to-noise ratio (SNR). The idea is to acquire several images during one frame. Based on the multiple sampling method, the method described in this thesis uses the special feature of the pinned photodiode to select the charge transfer time locally to change the total integration time in one frame based on the input light intensity. The comparator is reused during the different conversion phases to save power and area. Only one comparison is needed after each charge transfer, so compared with traditional multiple sampling method for DR enhancement this method can save a lot of power. In-pixel memory is used to store the converted data, and these data are read by column wise current sense amplifiers. The simulation results show that the charge transfer of the pinned photo diode is controlled locally and for a 7-bit dynamic range enhancement the additional power is only 2.2% compared with no DR enhancement.

Keywords: Pixel ADC, pinned photodiode, smart pixel, slope ADC, dynamic range enhancement, multiple sampling.

Contents

List of Tables

List of Figures

1 Introduction

Image sensors find applications in security, industrial, consumer, medical and scientific fields due to their ability to convert an optical image to an electric signal. Over the last few years, imager technology has developed rapidly thanks to applications in cell phones as well as consumer digital cameras. There is no doubt that image sensor technologies will continue to develop in the future.

The two dominant types of semiconductor-based image sensors are charge-coupled devices (CCDs) and complementary metal-oxide semiconductor (CMOS) image sensors (CIS). CCDs were the dominant solid-sate image sensor technology since the mid-eighties due to being relatively simple devices and their ability to produce high quality, low noise imagers. At that time, CMOS performance was limited by the available lithography technologies. Since the 1990s, the developments in CMOS process technology has made CMOS image sensors serious competitors to CCDs. CMOS image sensors have the advantage of easily integrating analog and digital signal processing blocks in the same chip as the sensor. This benefit has helped to scale down the physical dimensions and lowered the cost and power consumption of the sensor. Other arguments favoring CMOS include operation with a single power supply, and the ability to do region-of-interest (or windowed) read-out [1].

This thesis deals with the design of Analog-to-Digital converters (ADCs) for CMOS image sensors.

1.1 The imaging system

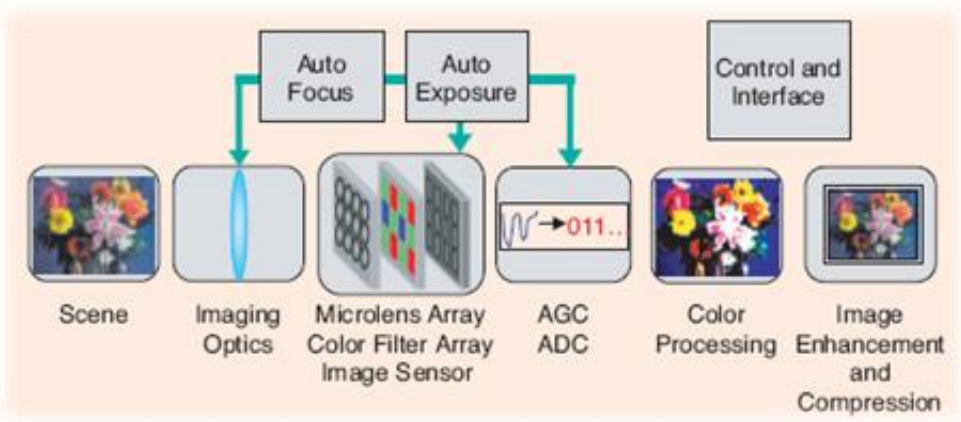


Figure 1-1 Imaging system pipeline[1]

Figure 1-1 shows a simple imaging system pipeline. First, the image scene is focused on the image sensor with the help of imaging optics. If the application requires it, a color filter array is placed on top of the image sensor arrays for color sensing. Because of the color filter, each

pixel behind the filter array produces an electrical signal corresponding to one single color (red, green or blue). An analog-to-digital converter (ADC) is used to convert the generated signal into the digital domain for digital post processing such as color processing, image enhancement, data compression for storage, etc. The digital image data after the ADC block can also be fed back for auto exposure and auto focus as shown in the figure.

1.2 Fundamentals of image sensors

The generation of electrical signal as a result of incoming light on the sensor is based on the photo-electric effect, discovered by Albert Einstein in 1905. Most semiconductor-based image sensors use photodiodes as shown in Figure 1-2 .

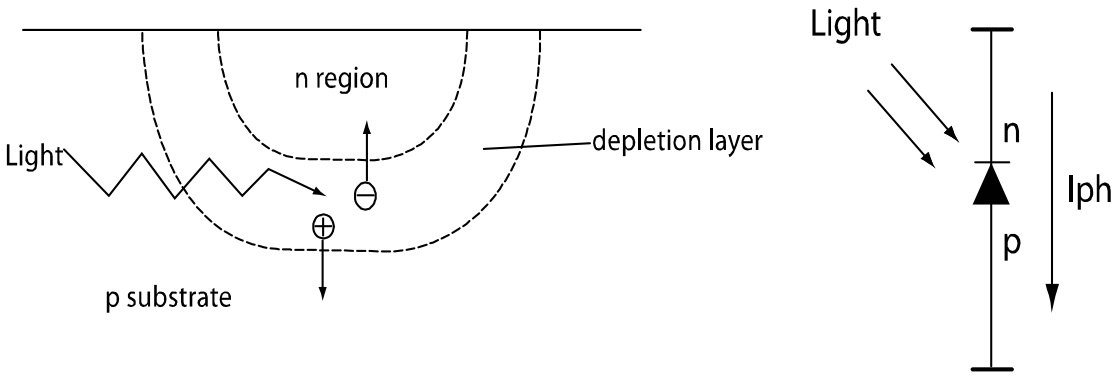


Figure 1-2 Photodiode under the illumination of light[2]

When the photodiode is exposed to light, the energy of the incident photons can be transferred to the individual silicon atoms. If this transferred energy is larger than the band gap of the material (in our case Silicon), the electrons in the valence band of the material can jump to the conduction band and generate an electron-hole pair. In the p-n junction, an electrical field is generated in the depletion region, so when the electron-hole pairs are generated in the depletion region, the electrons will drift towards the n-doped region and the holes will drift towards the p-doped region, generating a photocurrent between the terminals of the photodiode. The magnitude of the photocurrent is proportional to the incoming photons per unit time, which corresponds to the intensity of the light. It is not easy to measure the photon current directly as this current is typically very small in magnitude, so most modern image sensors get the light intensity information by integrating the photocurrent on a capacitor and measuring the resulting voltage. The easiest way to perform this integration is to use the junction capacitance of the photodiode as it is shown in Figure 1-3. After the photo-diode is reset, the generated charges accumulate onto the junction capacitance and the voltage across the diode starts to decrease. The difference between the reset voltage and the final voltage carries the information of the light intensity.

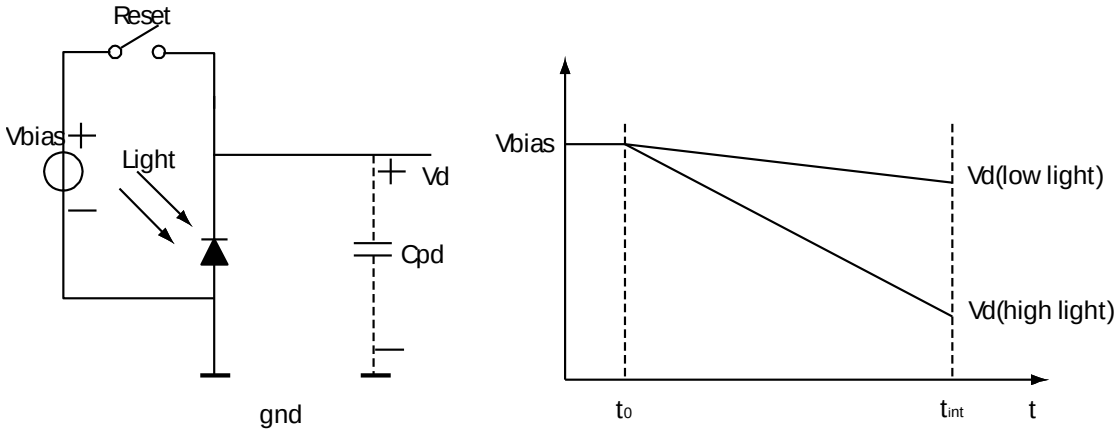


Figure 1-3 Integrating photodiode and plotting voltage vs. time [2]

1.3 Noise in image sensors

Noise sources in image sensors limit their performance, particularly for low light intensities. There are two types of noise sources in image sensors: temporal noise sources and fixed pattern noise sources.

1.3.1 Temporal Noise

Temporal noise is temporally random and not constant from frame to frame. It is described by statistic distributions and can be reduced by averaging successive frames.

There are several different kinds of temporal noise sources in image sensors. Here we will only describe the photon shot noise and the reset noise (kTC noise) which are especially important in image sensors. More information about the temporal noise sources can be found in [2].

Figure 1-4 illustrates the main temporal noise sources in image sensors.

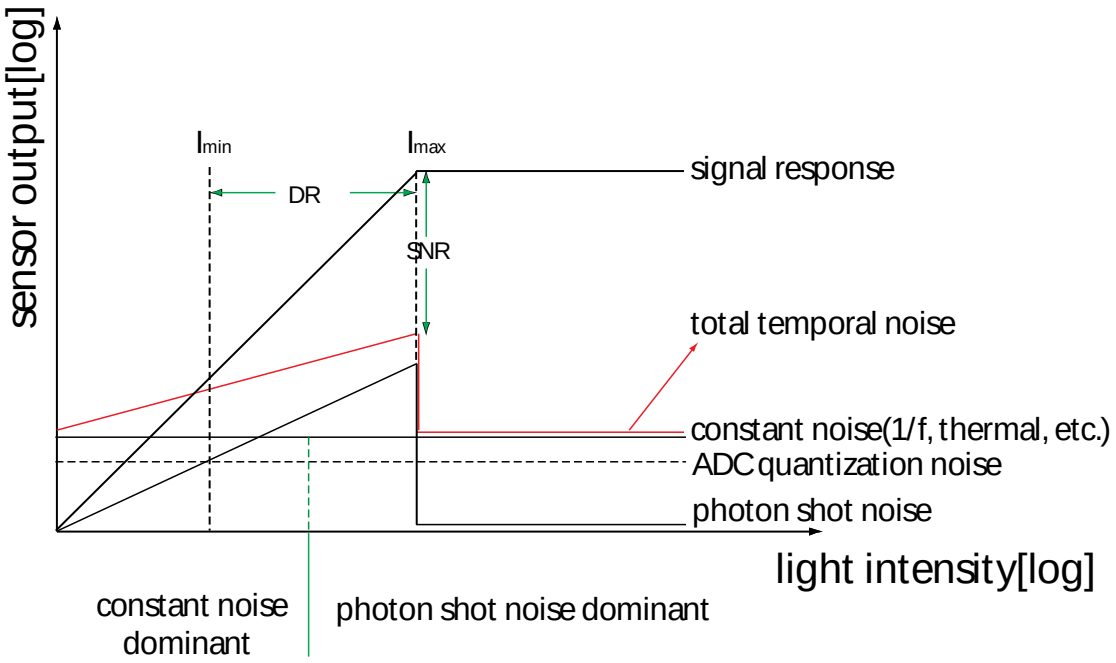


Figure 1-4 Sensor output response to light and noise sources

1.3.1.1 Photon Shot Noise

Photon shot noise describes the fundamental statistical uncertainty on the amount of photoelectrons that are generated by light falling on a photo detector. It relates to fundamental physical laws, rather than to IC technology or circuit design. The magnitude of the photon shot noise is equal to the square root of the mean number of electrons generated in the photodiode described as below:

$$\overline{N_{psn}} = \sqrt{N_{pe}} \qquad \text{Equation 1-1}$$

With $\overline{N_{psn}}$ being the magnitude of shot noise (in the units of number of electrons) and N_{pe} the number of photo-generated electrons. This relationship is also illustrated in Figure 1-4. When the sensor is in saturation (sensor output does not increase with the increase of light intensity), the photon shot noise goes down as the well capacity of the sensor is full of charges and the variation of the charges in the well is very small.

In most image sensors, with high illumination level the photon shot noise will be the dominant noise source. This can be used to simplify the ADC design, as will be shown in the next chapter.

1.3.1.2 kTC Noise(Reset Noise)

As described in section 1.2, the photo-generated electrons start to accumulate on the junction capacitor after the reset operation. As with other kinds of switched-capacitor circuits, this reset operation generates sampling noise equal to:

$$V_{n\_kTC} = \sqrt{\frac{kT}{C}} \qquad \text{Equation 1-2}$$

The magnitude of the reset noise can be large compared to other temporal noise sources such as thermal noise, 1/f noise, etc.

1.3.2 Fixed Pattern Noise

Fixed pattern noise is a kind of spatial noise that shows the variation of the output from pixel to pixel or column to column in the sensor array. It does not change from frame to frame. As human eyes are more sensitive to spikes in the image, pattern noise is very important in image sensor design. Fixed Pattern noise in image sensors is mainly caused by the mismatch in the readout circuits and in the sensor.

If column level or pixel level ADCs are used (see chapter 2), the ADC offset and gain errors will be the main noise source of FPN. Additionally the mismatch of the in-pixel circuits (e.g. source follower or other analog circuits) before the ADC will also contribute to the FPN. If a chip level ADC architecture is chosen, these circuit mismatches will not contribute to the FPN as they are shared by all the pixels.

FPN caused by the mismatch of the sensor is called Photon Response Non Uniformity (PRNU). It shows the sensor output non uniformity between different pixels in the Quantum Efficiency (QE). QE is the parameter which describes the relationship between the number of photon generated electrons and incoming photons.

1.4 Specifications of image sensors

Like all other integrated circuit applications, there are a number of parameters that determine the performance of the image sensor. In this section we will only list the most relevant parameters that will be used for the comparison among the different image sensors designs explained in the rest of this thesis.

1.4.1 Dynamic Range

The dynamic range (DR) of an image sensor shows its ability to capture scenes with different illumination. Figure 1-4 illustrates the image sensor output response to light and noise sources. DR is defined as the ratio of the largest and smallest signal levels that can be measured. The largest illumination level I_{max} is typically determined by the full well capacitance of the photodiode and the integration time. The lowest illumination level I_{min} that can be detected is determined by the noise floor. The DR of the sensor shown in Figure 1-4 is:

$$DR = \frac{I_{max}}{I_{min}} \qquad \text{Equation 1-3}$$

1.4.2 Signal-to-Noise Ratio

The signal to noise ratio (SNR) defines the image quality for the illumination that can be detected. It is determined by the ratio of signal power to noise power of the image sensor. The SNR in CIS for high light intensity applications is measured just before the sensor is saturated.

Unlike other types of sensors, in image sensors the SNR is typically smaller than the DR. The reason for this is that the photon shot noise is signal dependent, as shown earlier in section 1.3.

1.4.3 Fill Factor

Fill factor (FF) is the ratio of the photo sensitive area to the total pixel area. When more transistors are placed into the pixel, the fill factor goes down because the total pixel area is limited. As the pixel size gets smaller, the FF becomes even worse. For monolithic image sensors the metal layers on top of the photo-detector will limit the amount of light that can be collected in the pixel.

1.4.4 Sensitivity

Sensitivity describes the sensor output response of the image sensor with respect to the input light intensity. With a certain pixel area, the sensitivity depends on the FF and Quantum Efficiency (QE). It shows the ratio of collected charges to the number of incident photons.

1.4.5 Conversion Gain

The conversion gain is defined as the output voltage for one photo-generated electron. It depends only on the capacitor of the conversion node. The conversion gain is given as:

$$CG = \frac{q}{C_{cn}} \qquad \text{Equation 1-4}$$

With C_{cn} the capacitance of the conversion node and q the charge of one electron.

1.4.6 Resolution

Resolution is the parameter that shows the number of pixels that are integrated in the sensor chip. In many applications especially in consumer electronics, most people think that when the number of pixels is doubled, the performance is also doubled. This is clearly not the case, as when the resolution becomes larger for the same size of the chip, the pixel size decreases. As a consequence, some of the charge carriers generated in the silicon due to incident light may diffuse into the nearby pixels. The sensitivity of the sensor also goes down because the FF becomes smaller.

1.4.7 Power consumption

Power efficiency is always the target for integrated circuits. In conventional image sensors, the power is mostly consumed by the analog readout circuits. If the sensor includes analog-to-digital conversion, the ADCs will also contribute, and in particular if column level or pixel level ADC architectures are used.

1.5 CMOS pixel architectures

CMOS pixel circuits include the basic sensor structure and some in-pixel signal processing circuit to output the incoming light intensity information. The architecture of CMOS pixel can be divided into three groups: passive pixel sensor (PPS), active pixel sensor (APS) and digital pixel sensor (DPS). The output signal can be a voltage (PPS and APS) or a digital number if we incorporate the analog-to-digital converter into the pixel (DPS). Schematics of these kinds of pixel circuits are shown in Figure 1-5.

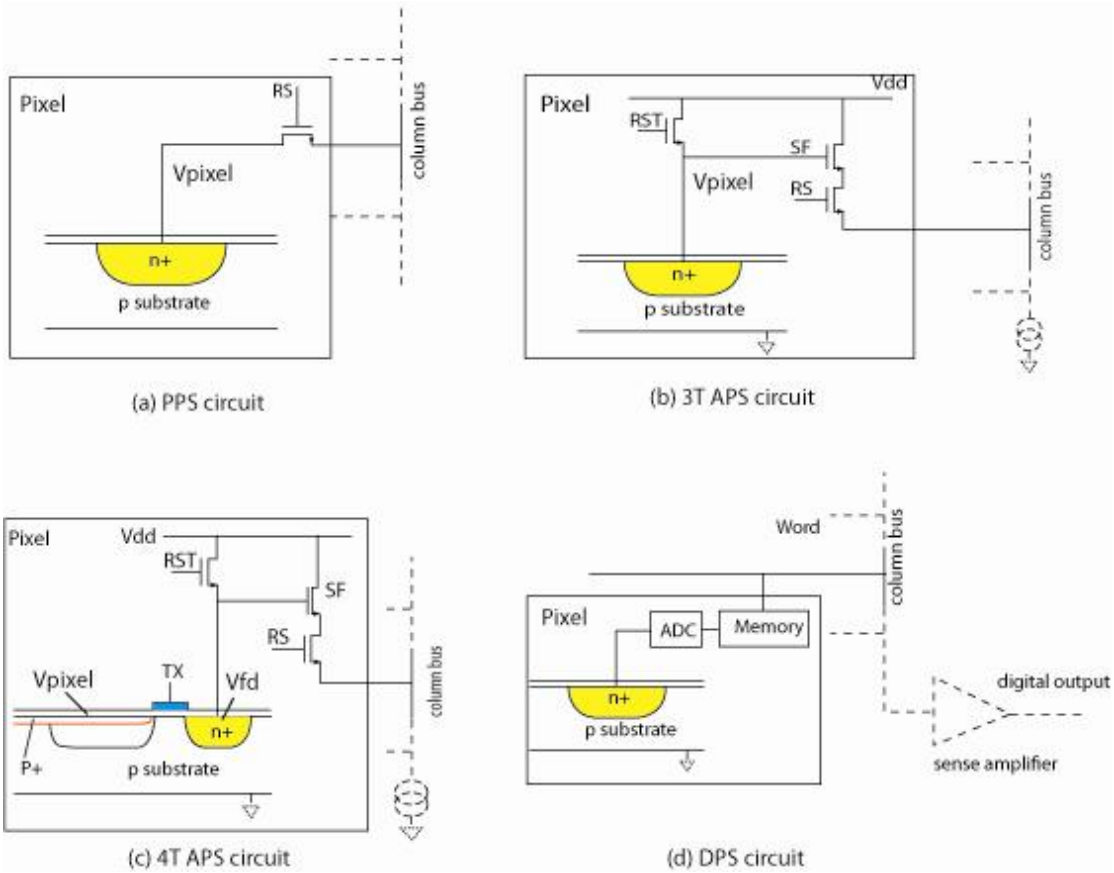


Figure 1-5 PPS, APS and DPS circuit schematic

1.5.1 Passive pixel sensors (PPS)

PPS is the first generation of CMOS image sensor architecture. As shown in Figure 1-5(a)[3], there are no amplifiers in the pixel. Only one row select transistor is used as a switch to read the photo-generated charge in the photo diode. Because of the mismatch between the small

pixel capacitance and the large vertical bus capacitance, the sensor suffers a high noise level and a slow readout. The main advantage of PPS is its high FF because of the simple pixel circuit.

1.5.2 Active pixel sensors (APS)

In the late 1960s, image sensors with in-pixel amplifiers, as illustrated in Figure 1-5(b) [3], were introduced. This new type of pixel consists of a photodiode, a reset transistor, a source follower transistor (in-pixel amplifier) and a row select transistor. Because three transistors are used in the pixel, this type of pixel with a photodiode is often called a 3T pixel.

The operation of a 3T pixel (illustrated in Figure 1-6) is as follows: In the beginning, the photodiode is reset to V_{DD}, after that the photodiode starts to accumulate the photo generated electrons in the integration capacitance and the voltage starts to decrease. At the end of the integration, the signal voltage (V_{signal}) is read out. The photodiode is then reset again for the next frame and the reset voltage (V_{reset}) is read. The difference between the two voltages (V_{reset} and V_{signal}) is proportional to the light intensity. The subtraction of the two signals can also suppress the thermal noise, 1/f noise of the readout transistor, but the reset noise introduced by resetting the photodiode will not be suppressed. This noise will be increased by a factor of $\sqrt{2}$ because the voltages are not obtained in the same frame. Because this operation does not remove the kTC noise, we call this Delta Double Sampling (DDS). The reason why we cannot read the two voltages in the same frame is that the pixel array is readout row-by-row and stored in the column structure, the double sampling operation needs to be operated in a short time period, the time interval between the two samples in one frame is the total integration time, which is quite long in low frame rate applications. More detailed information about 3T pixels can be found in [2].

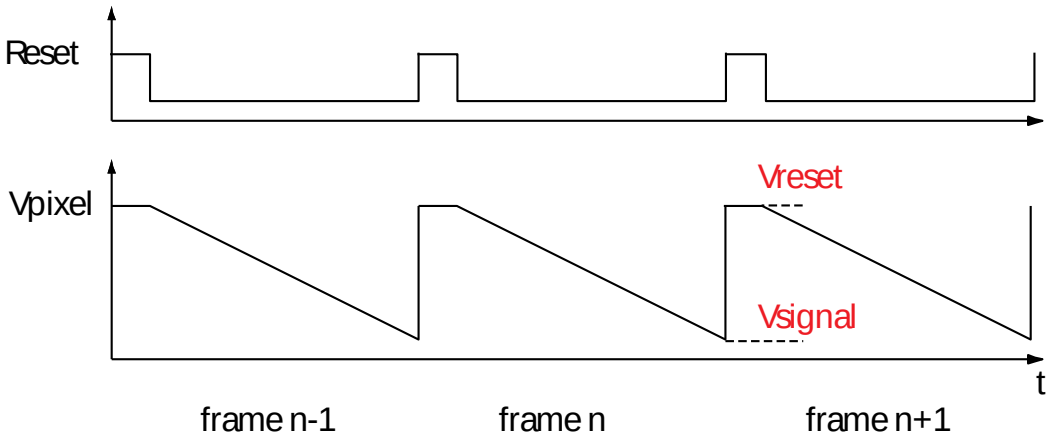


Figure 1-6 Timing diagram of 3T pixel operation

In order to incorporate Correlated Double Sampling (CDS) which can suppress the reset noise in the pixel, the pinned photodiode pixel was introduced as illustrated in Figure 1-5(c) [3]. This so-called 4T pixel has nearly the same structure as the 3T pixel. The main difference is that it employs a pinned diode, which adds a transfer gate (TX) and a floating diffusion (FD) node to the basic 3T APS pixel. When TX is switched on, all the charge integrated in the photodiode flows into the FD which acts as a sampling capacitor.

The readout operation of a 4T pixel is shown in Figure 1-7 and is as follows. First, the FD is reset by the reset transistor and the voltage V_{reset} is sampled by the readout circuit. After the charge integration in the photodiode, the transfer gate is enabled, allowing all the accumulated charge to flow completely from the photodiode to the FD. This charge transfer operation is very fast. After the charge transfer, the voltage V_{signal} at FD is then sampled and the two samples are subtracted from each other to get CDS.

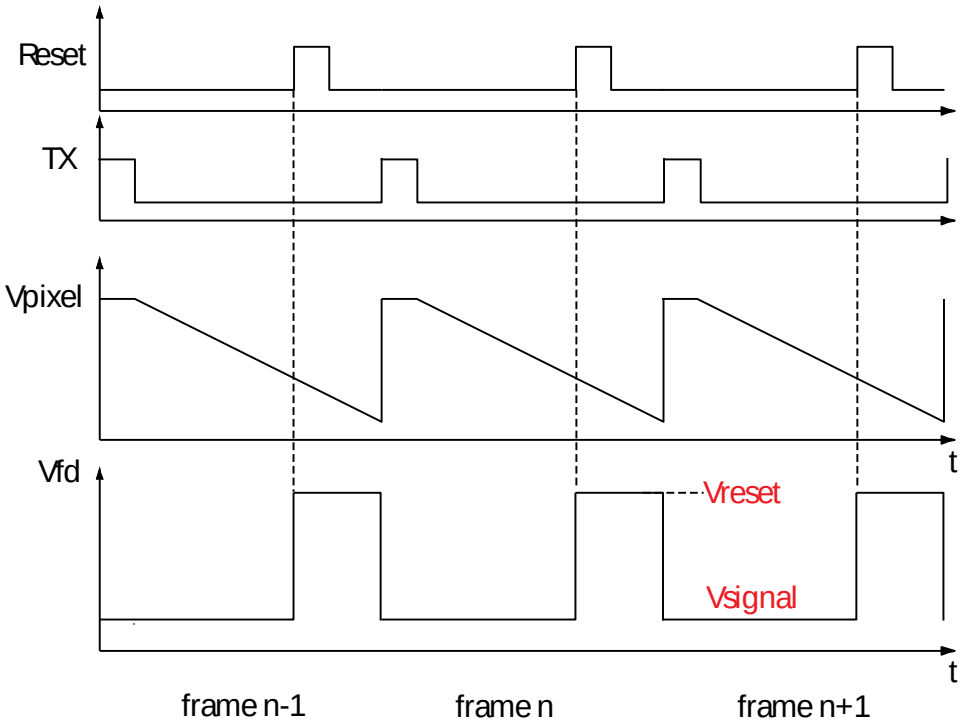


Figure 1-7 Time diagram of 4T pixel operation

1.5.3 Digital pixel sensor (DPS)

DPS is the most recently developed CMOS image sensor architecture. The schematic of such a pixel is shown in Figure 1-5(d) [1]. An ADC or even basic digital processing circuit can be incorporated into each pixel and the photon intensity information is read digitally. DPS offers several advantages over analog image sensors, including better scaling with CMOS technology due to reduced analog circuit performance demands and the elimination of read related column FPN and column readout noise. Besides, because the readout is performed fully parallel, high speed readout or low noise, low power design can be realized. This

advantage can give us the opportunity to increase the performance of the image sensor such as its DR as will be described later. The main problem with DPS is the increase in pixel size and/or low fill factor.

1.6 Research Motivation

In this work, the ADC is designed to be used in hybrid imagers. A concept diagraph of the hybrid image sensor is shown in Figure 1-8.

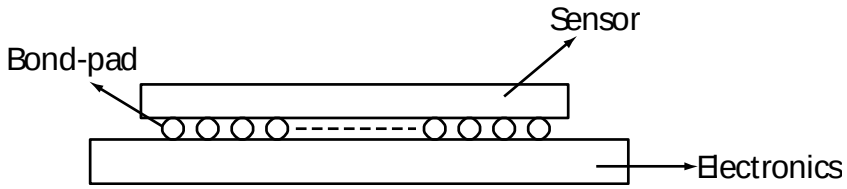


Figure 1-8 Concept diagraph of 3D-stacked image sensor

The upper layer contains the sensor part, which is back-side illuminated (BSI). BSI involves turning the image sensor upside down and applying the color filter and micro lens array to the backside of the pixels so that the sensor collects the light through the backside. The lower layer contains the electronics such as analog readout, digital processing and other periphery circuits. The connections between the two different layers are bond-pads.

The performance of the ADC has a very large influence on the performance of the sensor. It would be very helpful to find the state-of-the-art of the ADC architectures and designs that is suitable for hybrid structure by reviewing the papers on the image sensors that have been reported. The design can also be used in the 3D-stacked images, which several different layers are connected with Through Silicon Via (TSV).

2 ADC architectures used in image sensors

Most modern sensors incorporate analog to digital converters for back end digital signal processing. A CMOS imager has two main properties that differentiate it from other sensors. Firstly, it consists of a large array of light-sensitive pixels, which allows for a parallelized analog-to-digital conversion. Secondly, due to this large sensor array, the total data rate is much higher than most other sensors[3]. As a result, the performance of the ADC has an important impact on the whole performance of the sensor.

There are mainly three categories of ADCs incorporated with image sensors: chip level, column level and pixel level.

The block diagram of a CMOS image sensor with a chip level ADC is shown in Figure 2-1[2]. The analog outputs corresponding to one row are read by the column circuits. After the CDS circuit, the signal is fed into the ADC.

The advantage of chip level ADC is that because the CDS amplifier and ADC are implemented in chip level and shared for all the pixels, uniformity is ensured. There are two main drawbacks to chip level ADCs. The first one is that, because of the large number of pixels the CDS amplifier and ADC need to operate at very high speeds. The second problem is that this approach has a longer analog signal chain compared to the column-level or pixel-level A/D conversion approach. Since the gain in each of the analog circuits is typically limited to one, each sub-circuit will significantly contribute to the overall noise of the analog signal path.

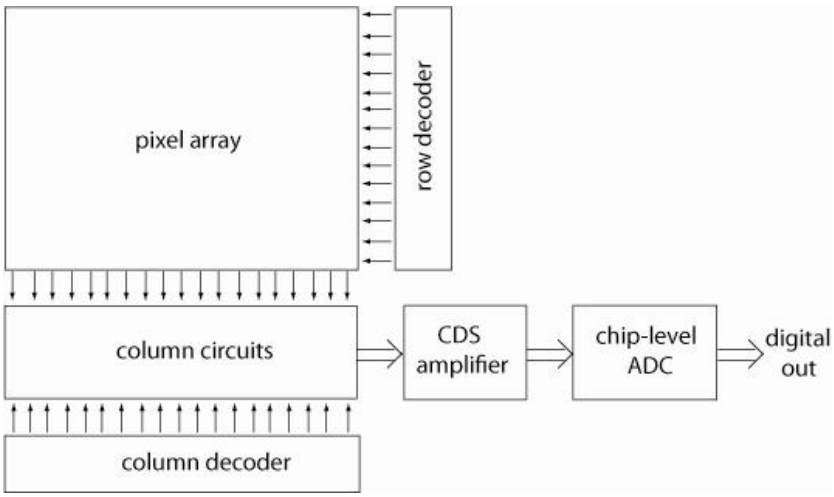


Figure 2-1 Chip level ADC schematic

To lower the operation speed of the ADC and shorten the analog path, column level ADCs are used. This approach is, at present, the most widely used because of its good compromise between frame rate, fill factor, power consumption and speed. The block diagram of an

image sensor with column level ADCs is illustrated in Figure 2-2[2]. In this architecture, both the CDS amplifier and ADC are located at the column level. The outputs corresponding to one row are sampled onto the column level capacitance. After CDS operation, they are digitized and then stored in the digital column memory.

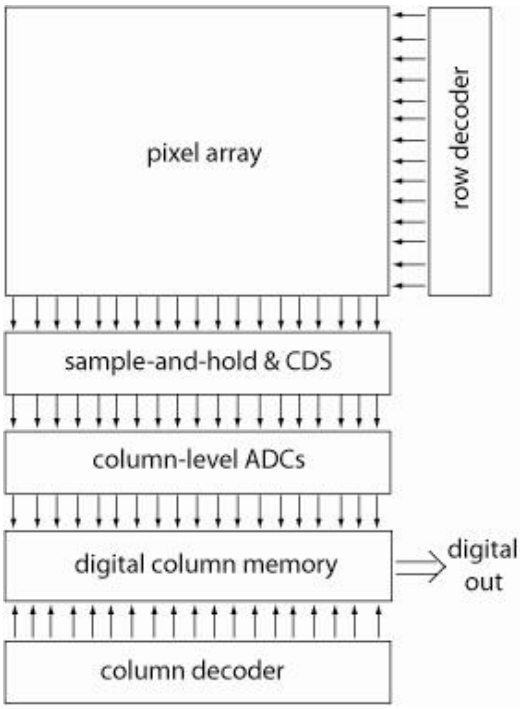


Figure 2-2 Column level ADC schematic

Image sensors incorporating column level ADCs will have hundreds to thousands of ADCs that operate in parallel in each column. As a result, column FPN generated by the offset and gain mismatches of the column circuits will create artifacts in the image that are highly visible by the human eye.

We can bring the analog signal processing and ADC down to the pixel level, creating the DPS as we have mentioned before. This architecture is illustrated in Figure 2-3[2]. Because of this fully parallel A/D conversion, very high readout speeds can be achieved, and the pixel FPN is less visible compared to column FPN [2]. Another advantage is that it easily scales to large resolutions because of fully parallel digital processing. The main problem is that because the ADCs are place in the pixel, the pixel pitch will be relatively large.

In the following we will focus our attention on column level and pixel level ADCs.

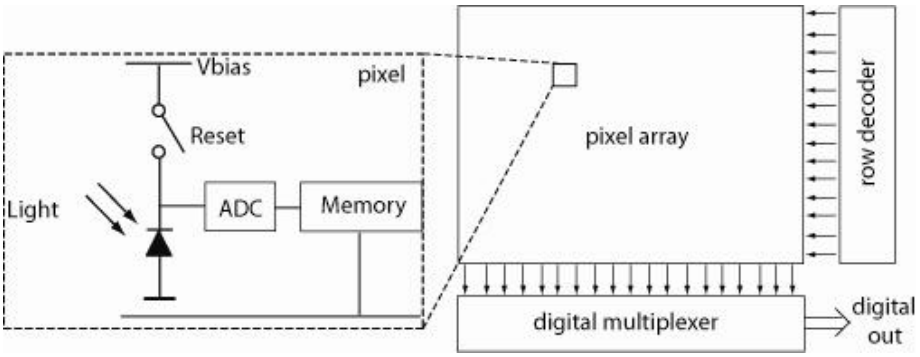


Figure 2-3 Pixel level ADC schematic

2.1 Column level ADC

As it has been explained in the previous chapter, the column ADC is used to convert the signals corresponding to the pixels in one column within one frame. The sampling rate of the column ADC should at least be:

$$F_s = fps \times N_{rows} \quad \text{Equation 2-1}$$

With F_s the sampling rate of the ADC, fps the frame rate of the sensor and N_{rows} the vertical resolution of the sensor.

The difference with ADCs used for other type of sensor is that, because human eyes are very sensitive to spikes in the image, the integral nonlinearity (INL) performance is not so critical, while the differential nonlinearity (DNL) is very important[2].

As the ADCs are placed in the column of the image arrays, the area is very critical of the ADC, especially the width of the ADCs. We need to fit the ADCs in the image array pitch, so the width should be the same as the pixel pitch if we place column ADC arrays on one side of the imaging array. To overcome this limitation, we can place the ADCs both above and below the pixel array, so that their width can be twice as much as the pixel pitch. We can also choose to place one ADC for the conversion of two or three columns, as a compromise between column and chip level ADCs.

There are mainly three types of architectures used in column ADC: ramp, cyclic and successive approximation ADCs. Other architectures such as pulse width modulation (PWM) [4]and Divide-by-Two [5] are not so common and will not be explained in the following.

2.1.1 Ramp ADC

Ramp ADCs use a ramp as a reference to compare with the signal voltage from the pixel. There are many variations on this architecture, which we will summarize in the following.

Single slope (SS) [2, 6-7] is the most popular architecture for column ADCs because of its simplicity. The timing diagram of a SS ADC is shown in Figure 2-4. Only one comparator and a memory is needed in each column. The ramp generator and the counter are common for all ADCs. As a result, it needs much less chip area compared to other types of ADCs. When the ramp signal crosses the column pixel signal, the output of the counter is latched into the column memory.

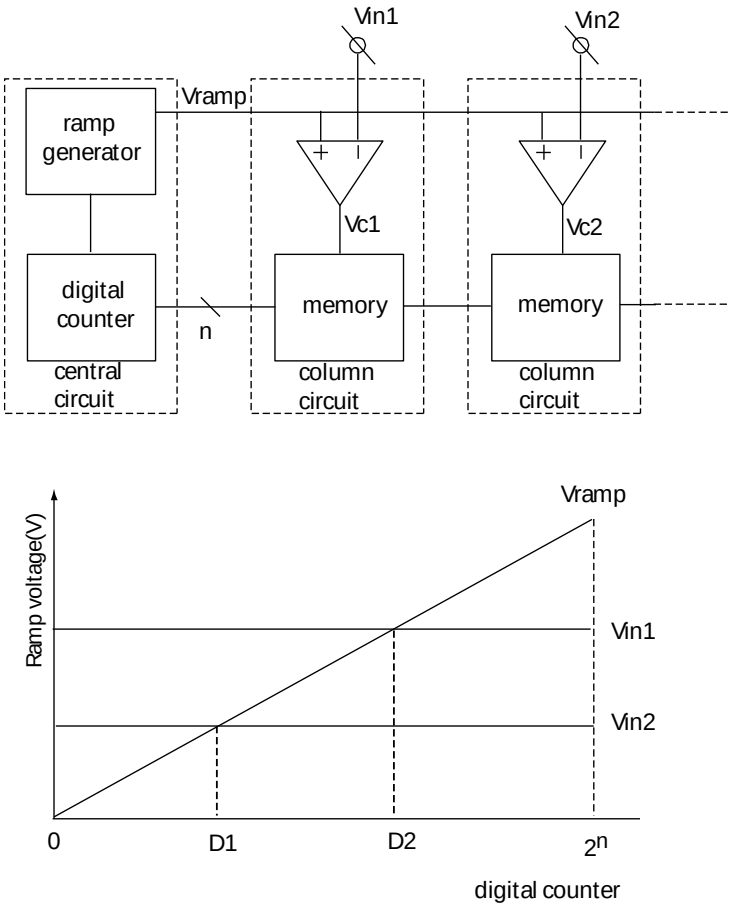


Figure 2-4 Timing diagram of SS ADC

Its simple circuitry makes this type of ADC relatively easy to fit into the pixel pitch, and column non-uniformity is easily reduced (less FPN).

In [7] the counter is placed in each column to implement digital CDS by first down counting for the reset signal and then up counting after the integration. The result is then latched and no subtraction is needed and only one memory is needed. Of course the penalty is more area to place one counter in each column.

The main problem with SS ADC is the slow conversion speed. For a resolution of M bits, $2^M - 1$ cycles are needed for each row. As a result, SS ADC limits the readout circuit speed and is thus not suitable for high speed applications.

In [2], the Multiple Ramp Single Slope (MRSS) architecture is introduced to try to solve the speed problem associated with SS ADC while keeping its simplicity. The timing diagram of a MRSS ADC is shown in Figure 2-5. In principle it is a two step ADC. One coarse ramp and several fine ramps with the same slope but different offsets are used. In the first phase, the coarse ramp is input to all column comparators, deciding the coarse bit of the signal and choosing which fine ramp will be used for the fine conversion. Then the results are fed back for controlling and the fine A/D conversion is performed outputting all the fine bits concurrently among all the columns. The MRSS architecture suffers from matching problems of the sub-ramps, limitation in the number of coarse conversion bits due to the circuit complexity, and higher power consumption due to the need of more sub-ramp signals and switches in the column.

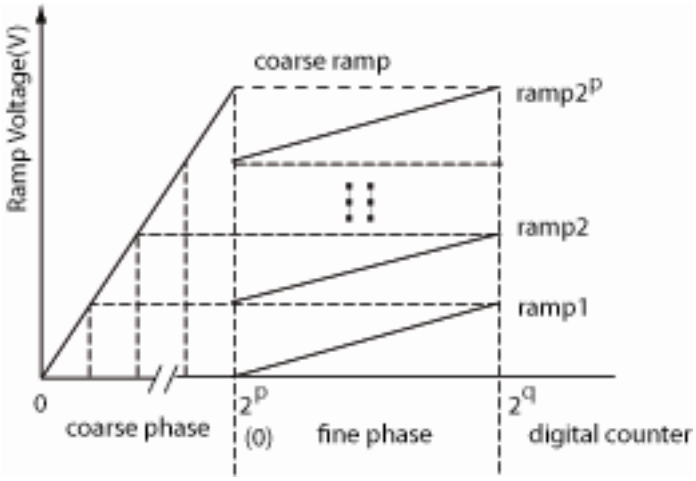


Figure 2-5 Timing diagram of MRSS ADC

The problem of slope matching and limitation of the number of bits for the coarse conversion in MRSS is solved in [8] by placing a hold capacitor in the ADC to store the coarse voltage levels after the coarse conversion , this voltage is then added with the fine ramp to compare with the analog signals to determine the fine conversion bits. Only one fine ramp is used in this case, so the increase of the coarse bits will not increase the number of sub-ramps and the switch lines complexity in each pixel. And of course the matching problems are gone because only one fine ramp is used.

When designing an ADC for image sensors, the quantization noise is kept slightly lower than the front end constant noise floor, as shown in Figure 2-6.

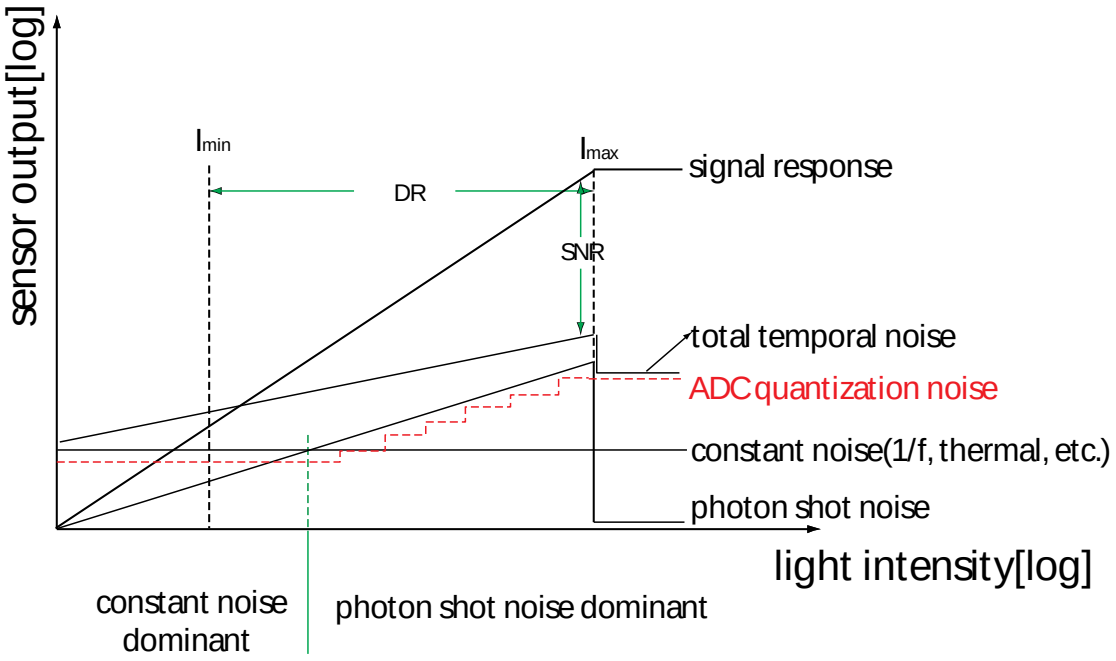


Figure 2-6 Noise diagram of the MRMS ADC

For higher illumination levels, the photon shot noise increases and will be the dominant noise source. This property can be used to increase the quantization step for high illumination levels without influence in the sensor performance. This can be used to increase the conversion speed without any hardware adjustment in what is called the Multiple Ramp Multiple Slope ADC (MRMS) [2]. The timing diagram of this algorithm is shown in Figure 2-7.

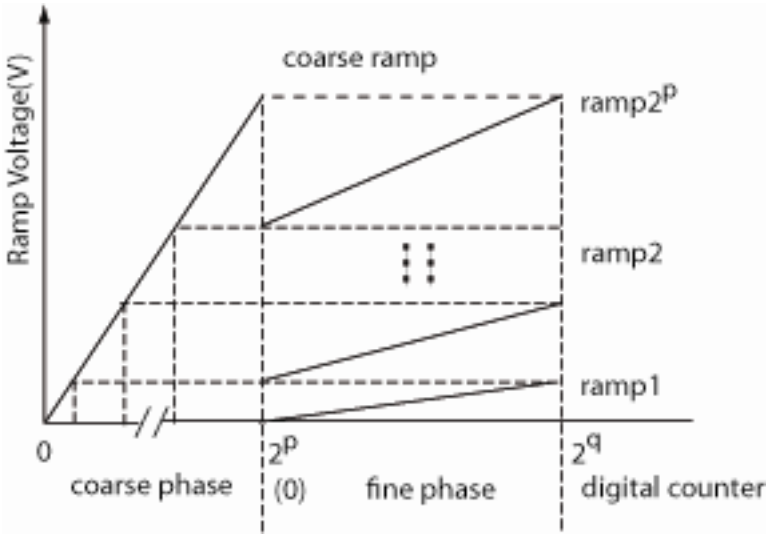


Figure 2-7 Timing diagram of MRSS ADC

The slope of the fine ramp is higher when the illumination level, determined in the coarse conversion, is higher. Either lower power consumption or higher conversion speed can be

achieved with this method. This algorithm can also be applied to the SS ADC, resulting in the piecewise SS ADC.

Another variation on the ramp ADC architecture is the Single-Slope Look-Ahead Ramp (SSLAR) ADC described in [9]. This ADC uses the statistics on the sampled row signals to achieve higher ADC speed and lower power consumption with a slight degradation of the image quality.

2.1.2 SAR ADC

Although many methods have been proposed to increase the speed of the ramp ADC, many conversion cycles are still needed, especially for high resolution. SAR ADCs need only M conversion cycles for a resolution of M bits. The main disadvantages of SAR ADCs are the relatively large DAC area and the complex control circuit needed in each ADC which makes this type of ADC difficult to fit into the narrow pixel pitch.

A regular SAR sequence includes a trial of a new binary value, a comparison and latching the result. Pipelining the third stage (latching) with the first stage (new binary value) can be used to speed up the conversion as is done in [10]. The readout of the digital data of one row can be pipelined with the conversion of the next row by adding a row of SRAM cells in the column level. Because of its high speed, the ADC can also be shared by two or more columns. Offset and mismatch among the capacitors which will cause fixed pattern noise can be compensated by adding smaller calibration capacitor banks in parallel with the capacitor bank for conversion [10-11]. For high resolution SAR ADCs (beyond 10 bits), the capacitor bank has to be very carefully designed due to parameter mismatch. This means that the area of the DAC will be very large if the resolution goes up. One possibility to solve this problem is to scale the largest or smallest capacitance and at the same time change the reference voltage [11-12].

The first CMOS APS with on-chip column-parallel SAR ADC was proposed in 1997 [13]. The DAC is designed with a capacitor bank based on the charge distribution algorithm. The comparator only consumes power during conversion and is switched off for voltage sampling and digital readout.

A schematic of a SAR ADC for image sensors is shown in Figure 2-8. The variable pixel signal is applied to the capacitor bank in order to be compensated during the convergence process.

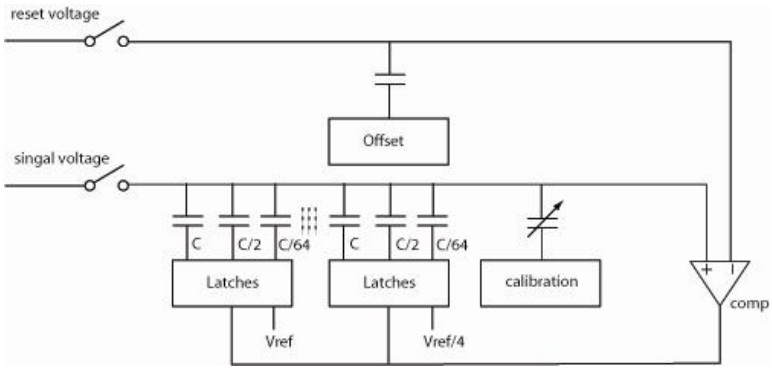


Figure 2-8 Schematic of SAR ADC for image sensors with mismatch compensation and accuracy enhancement[11]

2.1.3 Cyclic ADC

Cyclic ADCs have the same number of conversion cycles as SAR ADC. The difference is that instead of a DAC in the ADC, a cyclic ADC amplifies by two the difference signal between the input and the reference. Most of the designs incorporate the Redundant Signed Digit Coding method, which is also called the 1.5-bit/cycle algorithm, because it needs less precision in the comparator, leading to a great reduction of the power dissipation. Detailed explanation of Cyclic Redundant Signed Digit A/D converter can be found in[14].

The main problem with cyclic ADC is that the amplifier needs a very accurate gain matching. This means accurate capacitor matching and amplifier setting, which result in an increased power consumption.

In order to design high performance cyclic ADCs for image sensors (low area, high sampling rate, low power consumption), reuse of components and pipeline of timing slots should be incorporated, which makes the circuit and switch control logic very complicated. These methods will not be shown here and readers can find them in [15-17].

2.1.4 Comparison of column level ADCs

The extracted data from reviewed papers describing column level ADCs are shown in Table 2-1. The ADC figure of merit (FOM) is an important parameter to compare the performance of ADCs. It is calculated with the following equation:

$$FOM = \frac{Power}{2^{ENOB_{SNR}} \times F_s} \quad \text{Equation 2-2}$$

Where Power is the total power consumption of the ADC, $ENOB_{SNR}$ is the effective number of bits calculated from the SNR, and F_s is the sampling frequency of the ADC.

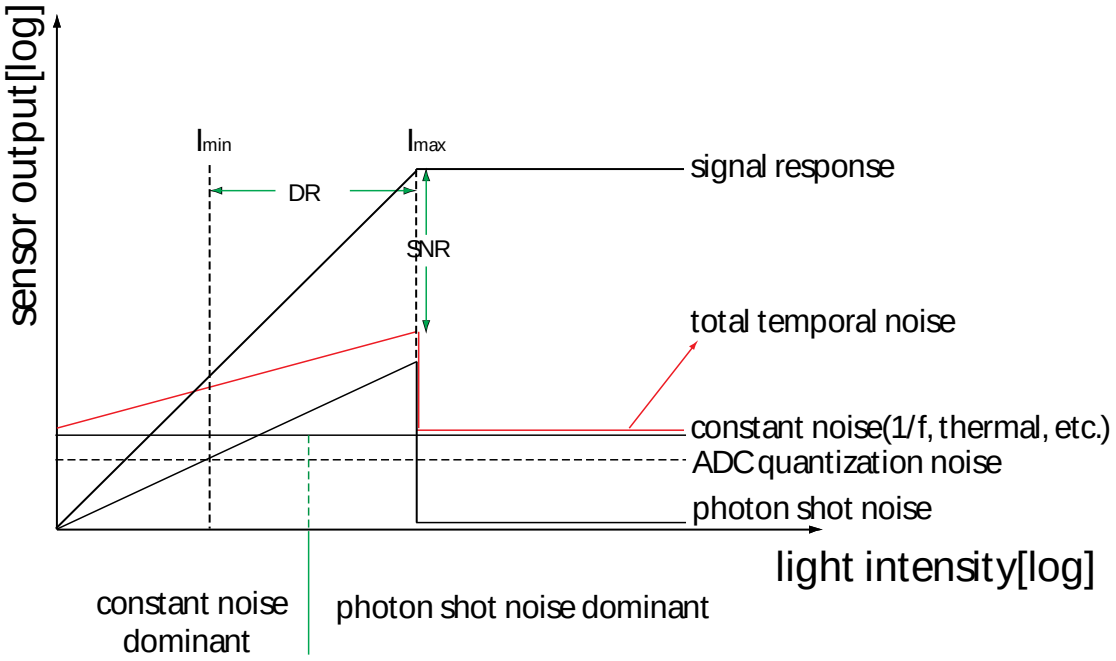


Figure 2-9 Sensor output response to light and noise sources

In the FOM calculation for image sensors in Table 2-1, we calculate the ENOB from the reported dynamic range. For a clear explanation of this decision, we recall the Figure 1-4, shown again in Figure 2-9. In the ADC design, the quantization noise is kept below the constant noise source level. In the high light intensity region, the noise is dominated by photon shot noise. So the ENOB or the noise performance of the ADC only affects the lower limit of the incoming light that can be detected. As the DR describes the ratio of the largest signal to the smallest signal that the sensor can handle, it is more appropriate to use the DR than the SNR when comparing the performance of the image sensor. Besides, in most of the papers reviewed the SNR is not given, and typically only the number of output bits or the DR is reported. In addition, in some designs, power is consumed for additional circuitry used to increase the DR. Taking all this into consideration we chose to use the $ENOB_{DR}$ (ENOB calculated from the reported DR) to calculate the FOM for ADCs in image sensors, which is:

$$FOM = \frac{Power}{2^{ENOB_{DR}} \times F_s} \quad \text{Equation 2-3}$$

Table 2-1 shows the data extracted from the papers reviewed. In the table, we use the reported $ENOB_{SNR}$ when the DR is not reported.

Table 2-1 Performance of the column level ADCs.

Ref.	ADC type	Sampling	Power/ADC	Area/ADC	ENOB	DR	FOM

		Rate(kHz)	(μW)	(μm^2)	(bits)	(bits)	(pJ/conversion)
[12]	SAR	588[3]	41	11088	12	n.a.	0.0170
[15]	Cyclic	345[3]	149	n.a.	12[4]	19.8	0.1054
[17]	Cyclic	435[3]	300	n.a.	13[4]	11.5[2]	0.2380
[16]	Cyclic	2000	430	88000	12[4]	9.67[2]	0.2631
[2]	Single Slope	19.6[3]	5.7	n.a.	10[4]	n.a	0.2840
[8]	Two Step Single Slope	250[3]	112.5[5]	n.a.	10[4]	10.5[2]	0.3169
[7]	Single Slope	259.2[1]	302.08[5]	n.a.	12[4]	11[2]	0.5678
[5]	Divide-by-two	2000	350.00	n.a.	8.2	n.a	0.5951
[11]	SAR	414.72[1]	297.62[5]	n.a.	10[4]	9.5[2]	0.9854
[2]	MRMS	81.3[3]	95	n.a.	10[4]	n.a	1.1411
[2]	MRSS	64.6[3]	95	n.a.	10[4]	n.a	1.4361
[18]	Single Slope	30.72[1]	58.6[5]	n.a.	11[4]	10.2[2]	1.6524
[19]	SAR	200	234.375[5]	n.a.	9	n.a	2.2888
[20]	SAR	259.2[1]	386.18[5]	n.a.	10[4]	8.84[2]	3.2426
[2]	Single Slope	19.3[3]	77.5	n.a.	10[4]	n.a	3.9214
[13]	SAR	27.03[3]	50	50400	8[4]	n.a	7.2266

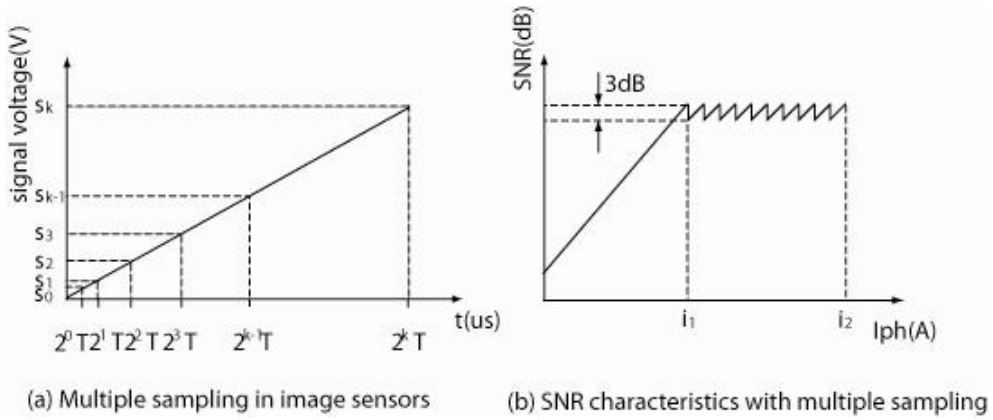
1. Data calculated from the frame rate and number of rows in the image array
2. Data calculated from the reported value in units of dB
3. Data calculated from the conversion time
4. Data obtained from the reported output digital bits
5. Data calculated from the total power divided by the number of ADCs in the sensor.

This table can give us an insight about the best architectures for the needed specifications. For example, for high frame rate or high conversion speed SAR, Cyclic or Divide-by-Two converters are the best choices, as they only need M times comparisons for M bits output. On the other hand they need larger area and have higher FPN. For low frame rates and small area, ramp ADCs are the best choice due to their low power consumption and area occupation.

2.2 DR enhancement by multiple sampling method

Multiple sampling can be used to increase the DR of the imager as is illustrated in Figure 2-10[21]. If the photon integrated voltage is always available, multiple sampling can be done at time $2^0\mathrm{T}, 2^1\mathrm{T}, 2^2\mathrm{T}, \dots, 2^k\mathrm{T}$. The sample at $2^0\mathrm{T}$ needs to be converted into M bits, while for the samples at $2^1\mathrm{T}, 2^2\mathrm{T}, \dots, 2^k\mathrm{T}$ we only need to convert the LSBs. Take Figure 2-10(a) for example. The voltage at 2T is two times the voltage at T for constant illumination, so the digital out bits from MSB to LSB+1 at 2T sample are equal to the digital output bits from MSB-1 to LSB at sample T. The total output for each pixel is combined into $\mathrm{M}+\mathrm{K}$ bits. And in the design the total number of comparisons needed for one pixel in each frame is

$(M + K) \times (2^M - 1)$ times. Figure 2-10(b) shows the SNR characteristics with multiple sampling method. Whenever the illumination is higher than the maximum illumination level that the sensor can handle with a certain integration time, the integration time is halved. And The SNR dips for 3dB at the illumination level that the integration time is halved as is shown in the figure. That is because the signal is halved while the photon shot noise is only $\sqrt{2}$ times lower (assuming photon shot noise is the dominant noise source). Beyond that illumination level, the SNR goes up with the increase of the illumination, and then it dips for 3dB again when the integration time is halved again.



(a) Multiple sampling in image sensors (b) SNR characteristics with multiple sampling

Figure 2-10 Schematic of multiple sampling to increase the DR of image sensors

2.3 Pixel Level ADC

Pixel-level ADCs are placed in the pixel very close to the generated image signals. The signal processing chain is short compared with column level or chip level ADCs. As a result less noise will be introduced by the analog signal processing circuits. In addition, because there is one ADC for each pixel, the illumination level information is available during the exposure and can be used to control the operation of each pixel dynamically (smart pixel image sensor). The main problem with pixel level ADC is the need for larger pixel area because we need to place one ADC in each pixel.

Different from traditional conversion algorithms to convert a voltage into digital value, some pixel level conversion algorithms embed the normal sensor operation in the conversion algorithm to increase the performance of the sensor. In the coming sections, we will briefly describe the following conversion algorithms: delta-sigma converters, multichannel bit serial (MCBS), single slope (SS), two-step ADC, charge counting ADC and dual-slope ADC.

2.3.1 Delta-sigma Converters

Delta-sigma Converters show higher dynamic range and better performance scaling into deep sub-micron CMOS technologies due to the fact that the over-sampling approach relies

on transistor speed rather than precision. This allows the data processing circuit to be simple and insensitive to process variations [22]. Imagers incorporating pixel level delta sigma converters are described in [22-25]. The schematic of a delta-sigma converter and an implementation for imagers is shown in Figure 2-11. The one bit DAC is designed with an analog shift register, using the three PMOS transistors shown in the figure similar to a 3-phase CCD transfer structure.

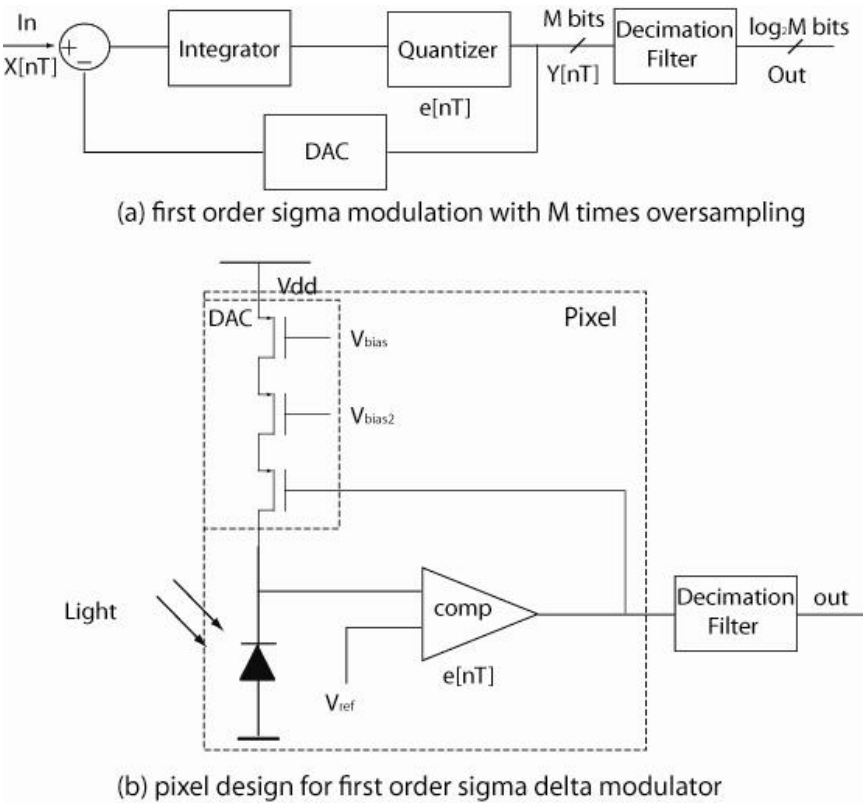


Figure 2-11 Schematic for delta-sigma converters in image sensors and circuit implementation

The input is the generated photon current and the integrator is the photodiode junction capacitance. When the comparator input is larger than the reference voltage (V_{ref}), a fixed amount of charge is fed back to the photodiode capacitance. The feedback charge can be controlled by V_{bias1} and V_{bias2} to increase the dynamic range of the imager.

The main problem with the delta sigma converter described above is that since the decimation filter is outside of the pixel array, too much data needs to be read from the pixel. The design in [26] placed a decimation filter in each pixel to reduce the output bit rates. But of course the penalty is larger area.

2.3.2 Multichannel bit-serial (MCBS)

A MCBS ADC implemented in the pixel level uses successive comparisons to output one bit at a time simultaneously from all pixels, so only one bit memory is needed in each pixel [27]. Another advantage of MCBS is that it can readily implement variable step-size quantization.

Table 2-2 shows an example using a 3-bit Gray code. Suppose S is the signal that we want to convert. According to the table, the least significant bit (LSB) can be generated by judging whether $S \in (1/8, 3/8] \cup (5/8, 7/8]$. If this is the case, LSB=1. The other bits are obtained similarly.

Table 2-2 Gray-code quantization table for the m = 3 example

ADC Input Range	Codeword
0- 1/8	0 0 0
1/8-2/8	0 0 1
2/8-3/8	0 1 1
3/8-4/8	0 1 0
4/8-5/8	1 1 0
5/8-6/8	1 1 1
6/8-7/8	1 0 1
7/8 – 1	1 0 0

The schematic diagram of MCBS is shown in Figure 2-12. When determining the LSB value, the RAMP signal begins at zero and monotonically steps through the boundary points $(1/8, 3/8, 5/8, 7/8)$. At the same time, the signal BITX starts at zero and changes value right before RAMP changes. When the output of the comparator changes, the BITX value is latched into the one bit buffer.

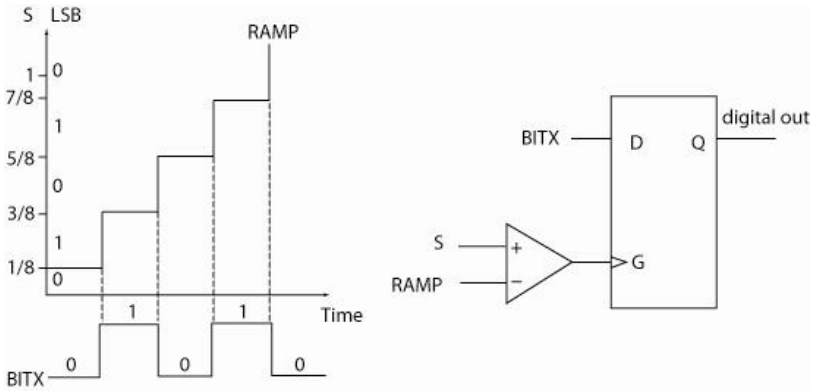


Figure 2-12 Schematic of MCBS ADC

Ideally in order to quantize S into M digital bits, at least a total of $2^M - 1$ comparisons are needed. But different RAMP signals are needed for each bit, which increases the design

complexity. In this design, the RAMP signal is the same simple single slope signal for all the output bits, so the needed number of comparisons is $M \times (2^M - 1)$.

2.3.3 Single Slope ADC

Single Slope ADCs can also be used in the pixel level because of its simple and robust circuit. The problem with this architecture in the pixel level is that, for M-bit resolution conversion, an M-bit memory should be placed at the pixel level and this will inevitably increase the pixel area. The first design with SS ADC in pixel level is reported in[28]. A very high frame rate of 10 000 fps was achieved by pipelining the reset, integration and readout., If this chip were used in low frame rate applications, the power consumption can be reduced by dynamically switching off the comparators when no conversion is needed.

2.3.4 Two-step SS ADC

While most of the methods for DR enhancement reduce the SNR of the sensor, [29] reported a pixel level ADC that can increase the SNR and DR at the same time by resetting the photodiode when the well capacitor is saturated in a fixed integration time. Timing diagram of the ADC is shown in Figure 2-13. By counting the number of reset times, a coarse conversion is achieved. The fine conversion can be performed by comparing the remaining voltage at the end with a single slope ramp.

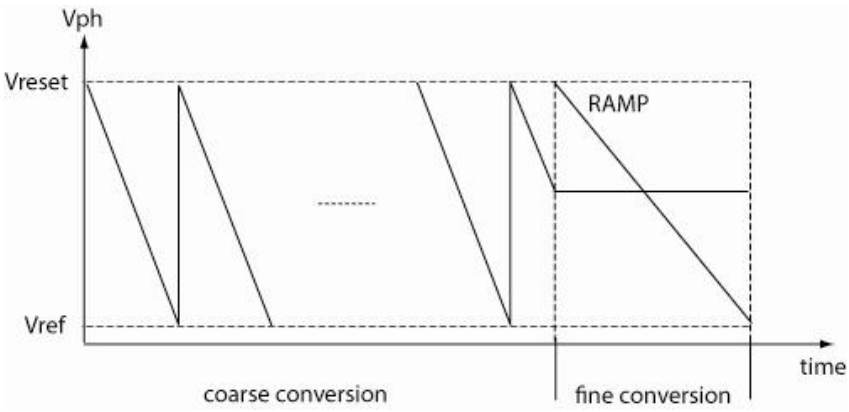


Figure 2-13 Schematic of the two-step ADC to increase the DR of image sensors

2.3.5 ADCs using charge counting technique

The charge packets counting technique[30] has the same operating principle as the two-step ADC described above. The first step is done by counting the reset times to get the digital bits. Two methods are described in the paper. One is called voltage resetting and the other one charge resetting technique. Voltage resetting is done by directly connecting the upper capacitor plate to the reset voltage while charge resetting is done by feeding the same amount of charge into the capacitor when resetting. The voltage reset version has a more efficient layout implementation and the charge reset version shows higher linearity.

2.3.6 Dual-slope ADC

The dual-slope A/D converter reported in [31] uses the same principle for DR enhancement as the multiple sampling described in the section 2.2, but the implementation is different. Timing diagram of this design is shown in Figure 2-14. The electronic shutter is closed when the photon integrating voltage is larger than V_{mid}. The time for integration is converted into the MSBs. Then the integrated voltage is discharged with a constant current source and the charging time is digitized. The problem with this design is that two different counters and comparators are used: a clocked comparator with SR Latch in the first phase for controlling the shutter and a continuous time comparator in the discharging phase. In the second discharging phase, the discharging current source should be matched very well across pixels and a cascode current source is used in each pixel, increasing the pixel circuit complexity.

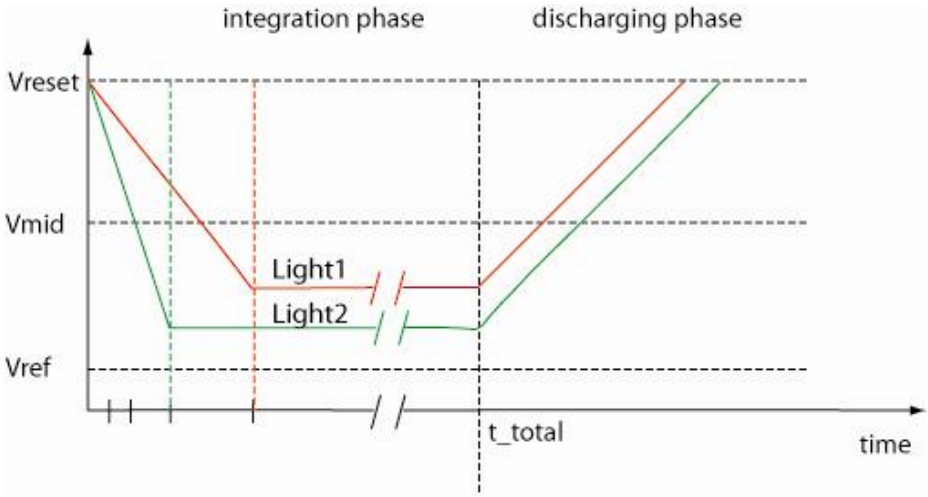


Figure 2-14 Schematic of the dual-slop ADC increasing the DR of image sensors

2.3.7 Comparison of pixel level ADCs

The extracted data from reviewed papers about the pixel level ADC performance is shown in Table 2-3.

Table 2-3. ADC performance of pixel level ADCs

Ref.	ADC Type	Sampling Rate(kHz)	Power/ADC (µW)	Area/ADC (μm^2)	ENOB (bits)	DR (bits)	FOM (pJ/conversion)
[29]	Two Step SS	1	0.05	n.a	12^2	19.6^2	0.0001
[22]	Delta Sigma	0.03	0.00088	n.a	9.6^2	16.3^2	0.0004
[31]	Dual Slope	3	7	4000	8^4	19	0.0045
[30]	voltage reset	0.5^3	0.25	48	n.a	14.7	0.0188
[30]	charge reset	0.5^3	0.25	264	n.a	14.7	0.0188

[28]	Single Slope	40[3]	1.2[5]	n.a	8[4]	n.a	0.0482
[30]	voltage reset	0.5	1.8	48	n.a	14.7	0.1353
[30]	charge reset	0.5	1.8	264	n.a	14.7	0.1353
[32]	PWM	0.2[3]	0.0744	n.a	8[4]	n.a	1.4531
[26]	Delta Sigma	0.05[1]	0.68	992	13[2]	n.a	1.6602
[21]	MCBS	0.57[3]	0.2441[5]	n.a	8[4]	n.a	1.6830
[27]	MCBS	0.48[3]	0.2441[5]	n.a	8[4]	n.a	1.9836
[24]	Delta Sigma	0.12[1]	0.0436[5]	n.a	6.9[4]	13.5[4]	3.0411

1. Data calculated from the frame rate and number of rows in the image array
2. Data calculated from the reported value in units of dB
3. Data calculated from the conversion time
4. Data obtained from the reported output digital bits
5. Data calculated from the total power divided by the number of ADCs in the sensor.

For area efficiency MCBS and, delta-sigma would be good choice, as they output one bit a time, so no memory is needed to be placed in the pixel. MCBS has a higher conversion speed than delta-sigma converters because it is a Nyquist converter. For high DR requirement or smart pixel, two step SS, charge counting and dual slope converters are probably the best choices, as they perform the conversion during the integration. Single slope ADC is a good choice if a simple pixel structure is required.

3 Design of a Pixel Level ADC

In the previous chapter, different architectures for column and pixel level ADC have been presented. In this chapter, a pixel level ADC with DR enhancement for hybrid image sensors will be presented.

3.1 Design specifications

The ADC is designed to be used in hybrid imagers. The sensor containing the photodiode and the transfer gate is placed in a first layer. The electronics such as the source follower, ADC and other periphery circuits are placed in a different layer. The pixels in the two layers are connected by two bumps: one for controlling the transfer gate and the connecting the floating diffusion to the input of the source follower as is shown in Figure 3-1.

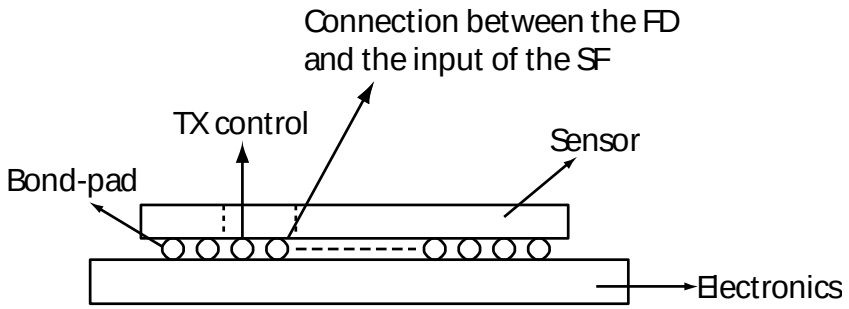


Figure 3-1 Diagram shown the connections for the hybrid sensor

The main target of the design presented in this chapter is to increase the higher limit of the DR in the sensor without much additional power and no SNR dip. The basic idea is to increase the DR in a similar way to the multiple sampling method which has been shown in section 2.3.2.

In Figure 3-2, a block diagram of the ADC array is depicted, along with the additional circuitry that is required to operate the sensor. All the control signals are globally distributed among all the pixels. The sense amplifier array is in the lower part of the chip, and the row decoder is in the right part of the chip, selecting the in-pixel memories and read the latched gray-code out.

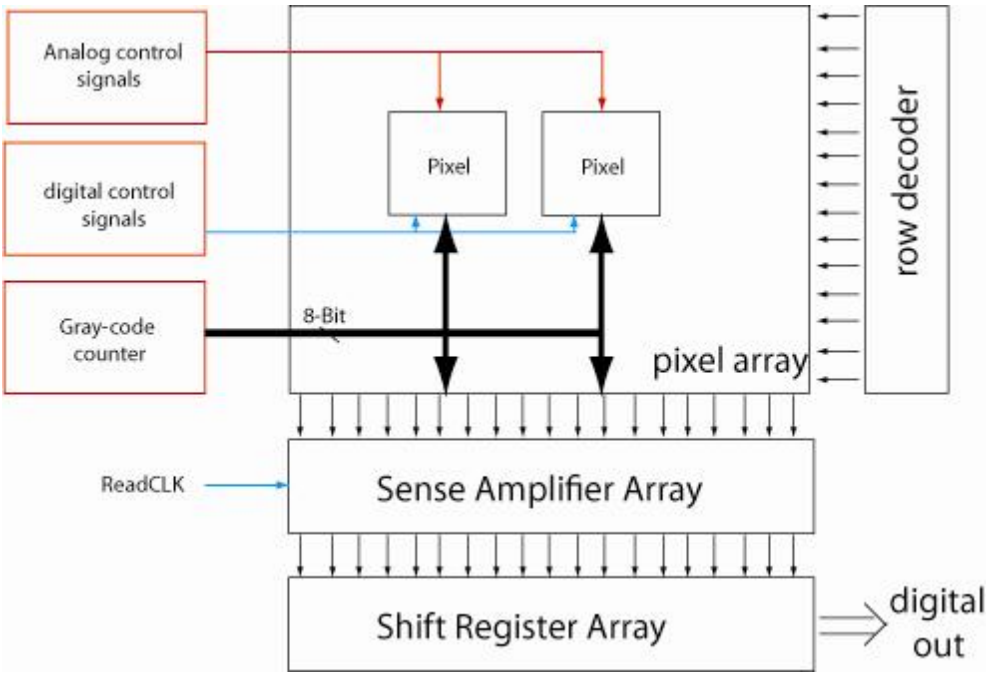


Figure 3-2 Block diagram of the sensor array with board-level circuitry

The bias for the source follower and the comparator will be disabled when it is not needed to decrease the power consumption. The comparator is reused for different conversion phases to reduce the area. Digital CDS is performed to remove the reset noise, 1/f noise and the offset of the comparator. Two different memory cells are used in the pixel to store the coarse and fine bits.

Table 3-1 shows the specifications of the image sensor. It is a high speed application with pinned photodiode. The FOM is also calculated with DR as have been shown in section 2.1.4.

Table 3-1 Specifications required and technology used for the image sensor

Pixel Size	< 20um x 20um
Pixel Type	4T Pinned Photodiode
Dynamic Range	>= 12 bits
Frame rate	>= 1000fps
FOM	<0.5pJ/conversion
Technology	TSMC 0.18µm

3.2 Pixel operation

The operation of a 4T pinned photodiode has already been explained in section 1.5. The photo-generated charges are integrated in the photodiode. After the transfer gate is activated, all these charges flow into the floating diffusion.

Conversion methods using a non-pinned photodiode such as delta-sigma, PWM, charge counting, or two-step SS continuously compare the photon integrated voltage with a

reference voltage. In pinned photodiode sensor, the integrated voltage is only available at discrete times after the charge transfer, so these methods cannot be used.

The proposed pixel level ADC uses the multiple sampling method to increase the DR of the image sensor. The main difference to the method shown in section 2.3.2 is that the method described in this thesis only does a comparison after a charge transfer, and not continuously. If the comparator and the SF are powered on only during the comparison time, a lot of power can be saved.

A schematic of the pixel is shown in Figure 3-3. The output of the source follower is connected to the input of the comparator, and the comparator output is used to control the transfer gate of the pinned photodiode as well as the memory to latch the digital code. The gray code counter and the ramp that is connected to the positive input of the comparator are globally distributed and common to all pixels.

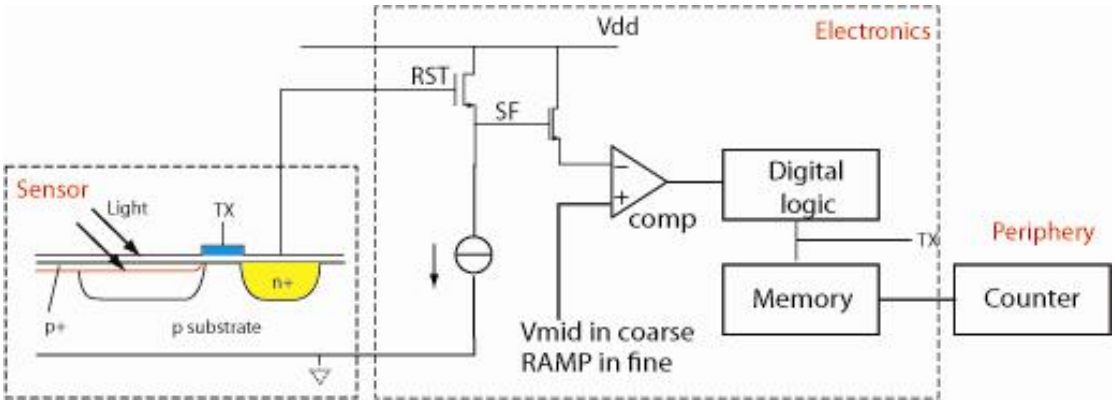


Figure 3-3 Circuit schematic for the proposed pixel level ADC

There are two conversion phases: coarse and fine conversion phase. Coarse conversion is to convert the coarse bits, which indicate the total integration time, fine conversion is to convert the voltage value at the floating diffusion. During the coarse conversion phase, the photo-generated charge is accumulated on the photodiode. The charge transfer gate TX is activated depending on the light intensity level. The shortest integration time is T and the longest integration time is 2^KT. The time slots for the operation of the pixel circuit are shown in Figure 3-4.

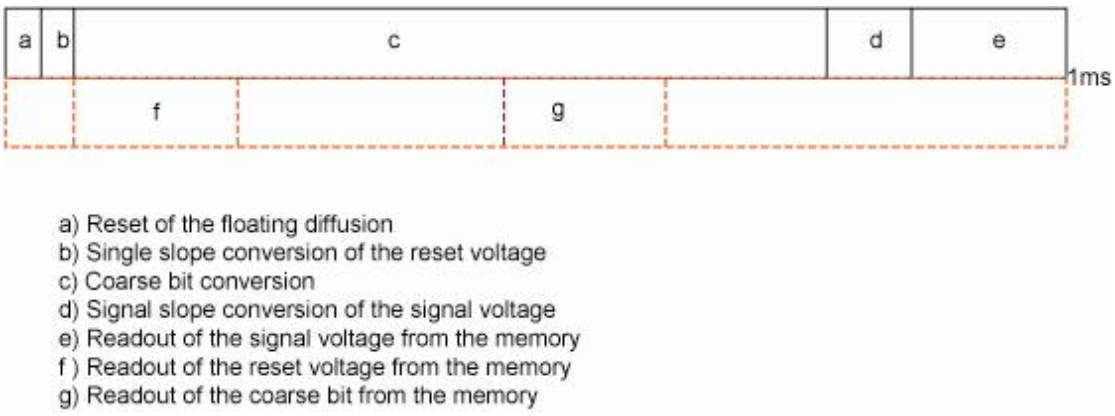


Figure 3-4 Time slots for the operation of the pixel circuit

Detailed operation algorithm is shown below:

1. Reset the floating diffusion (FD)
2. Single slope conversion of the reset signal voltage V_{reset}
3. Read the digital data corresponding to the reset voltage
4. Charge transfer at $2^{m}T$, with $m < k$.
5. Comparison of the SF output voltage with reference voltage
6. If voltage of SF beyond reference voltage, then charge transfer at $2^{(m+1)}T$, and go to step '5', otherwise no charge transfer anymore and hold the voltage level to the fine conversion phase.
7. Read the digital data corresponding to the integration time T_{int}
8. Single slope conversion of the signal voltage level V_{signal}
9. Read the digital data corresponding to the signal voltage

Figure 3-5 shows how the pixel functions for low light. Figure 3-6 shows how it works with a relatively high light intensity.

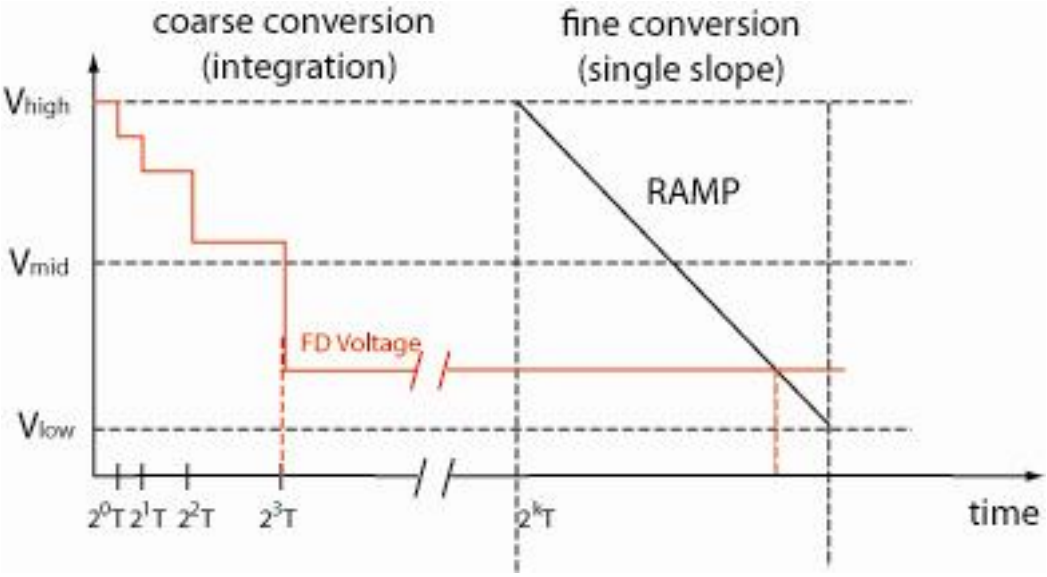


Figure 3-5 Waveform with low light intensity for the proposed pixel level ADC

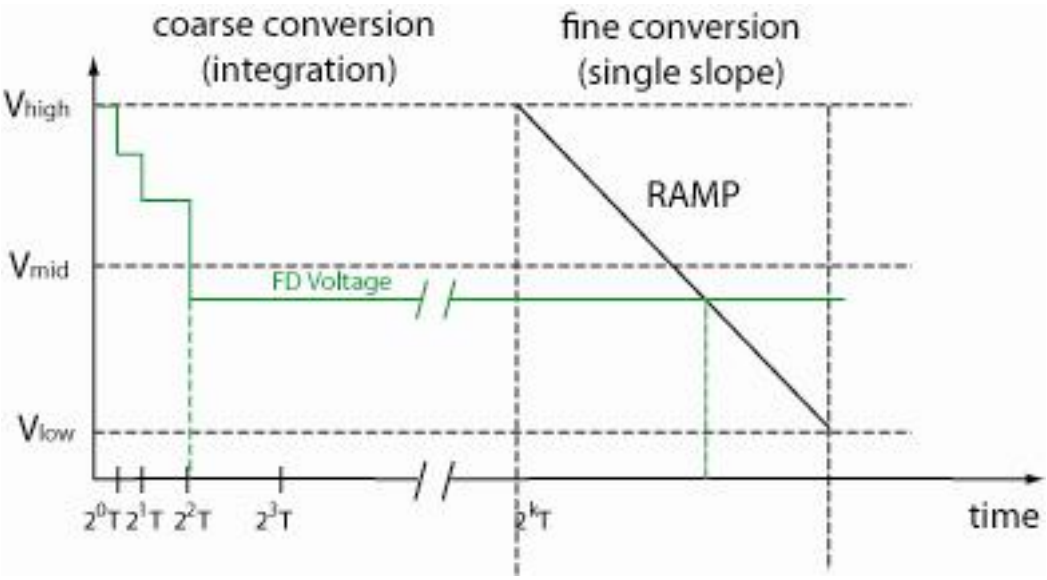


Figure 3-6 Waveform with high light intensity for the proposed pixel level ADC

For the low light intensity, after the charge transfer at 2^2T, the FD voltage is above the reference voltage V_{mid}, so another charge transfer is executed at 2^3T. After this the FD voltage is below V_{mid}, and no charge transfer is executed any more. The charges are kept in the photodiode for the second (fine) conversion phase.

The light intensity depends on the integration time (coarse bits) and the voltage difference between the reset voltage V_{reset} and the signal voltage V_{signal}, described with the following equation:

$$I_{ph} = \frac{V_{reset} - V_{signal}}{T_{int}}$$

Without multiple charge transfer, the maximum photon current that can be measured is:

$$I_{ph1\_max} = \frac{V_{reset} - V_{low}}{2^k T}$$

With multiple charge transfer method, the maximum photon current that can be measured is:

$$I_{ph2\_max} = \frac{V_{reset} - V_{low}}{T}$$

So the DR is enhanced by k bits as:

$$DR_{Enhancement} = \frac{I_{ph2\_max}}{I_{ph1\_max}}$$

3.2.1 Negative impact on the noise performance

Because the operation of the pinned photodiode in our conversion algorithm is different from the normal operation (see section 1.5.2), there are some negative impacts on the noise performance, listed below. More information on this can be found in [33].

a. The time between the two samples (reset & signal) levels needed for CDS is relatively long (the largest integration time), especially for low frame rate. So only part of the 1/f noise of the source follower will be cancelled. For each application, the area and bandwidth of the source follower should be well designed to keep this noise within the specifications.
b. The charge will be held in the FD for a long time (nearly the largest integration time), and as a result the dark current collected in the floating diffusion will be large.
c. Each charge transfer will add additional noise in the floating diffusion. The higher the number of charge transfers, the more noise will be added.

As the leakage at the FD node is through the contact metallization by the carrier trap in the interface between silicon and contact, one method to reduce the leakage at the FD node is to readout the voltage at FD using a floating gate (FG) that is placed on top of the FD[34]. Because the target of this thesis is to increase the high limit of the dynamic range, and even if the noise level is high because of this operation method, we can also design the quantization noise of the ADC to be relatively large for low power or higher speed. So this conversion method is also worth for investigation.

3.2.2 Correlated Double Sampling

In order to compensate the kT/C noise, 1/f noise, the offset of the comparator and the mismatch of the source follower, CDS is used. There are two ways to perform CDS, in the analog domain or in the digital domain.

A block diagram of the analog CDS is shown in Figure 3-7.

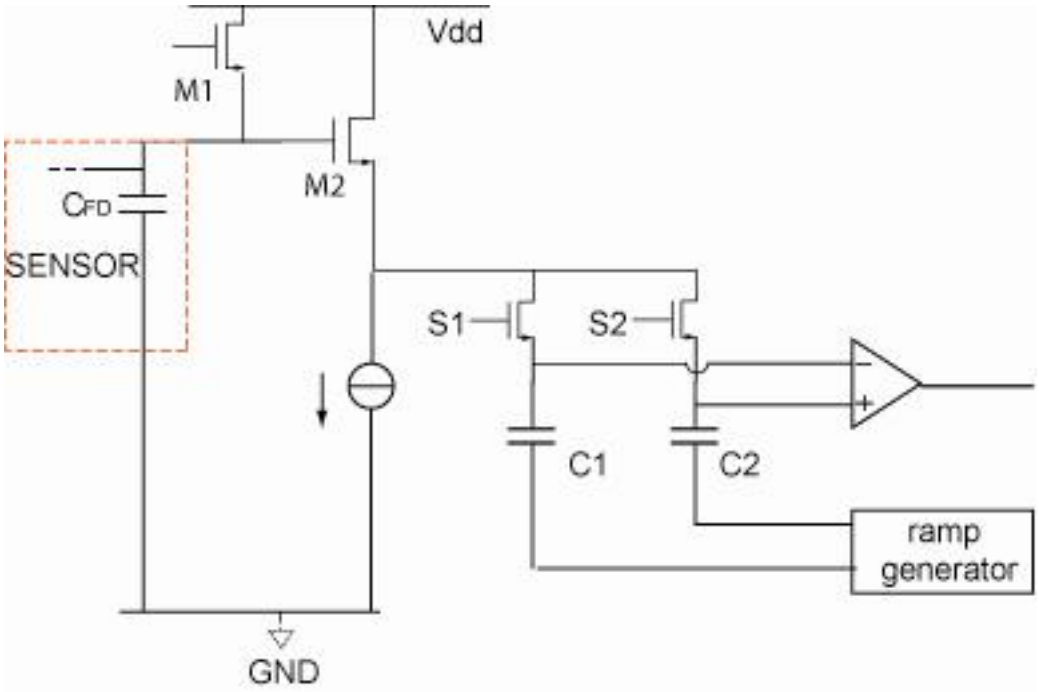


Figure 3-7 Block diagram for Analog CDS operation

After the reset of the floating diffusion, the reset voltage is sampled with switch S_1 on C_1. After the charge transfer, the signal voltage is sampled on the capacitor C_2 with switch S_2. In order to minimize the mismatch of the charge injection of the sampling switches, the size of the switch should be minimized, and the sampling capacitor should be relatively large. The bottom plates of the sampling capacitors are connected to the ramp generator. During the conversion, the difference between the two voltages sampled on the capacitors is converted into the digital domain. In order to convert this differential voltage into the digital domain without any subtraction circuit, a differential ramp is connected to the bottom plates of the capacitors. More detailed explanation for this conversion can be found in [2].

A block diagram of the digital CDS is shown in Figure 3-8.

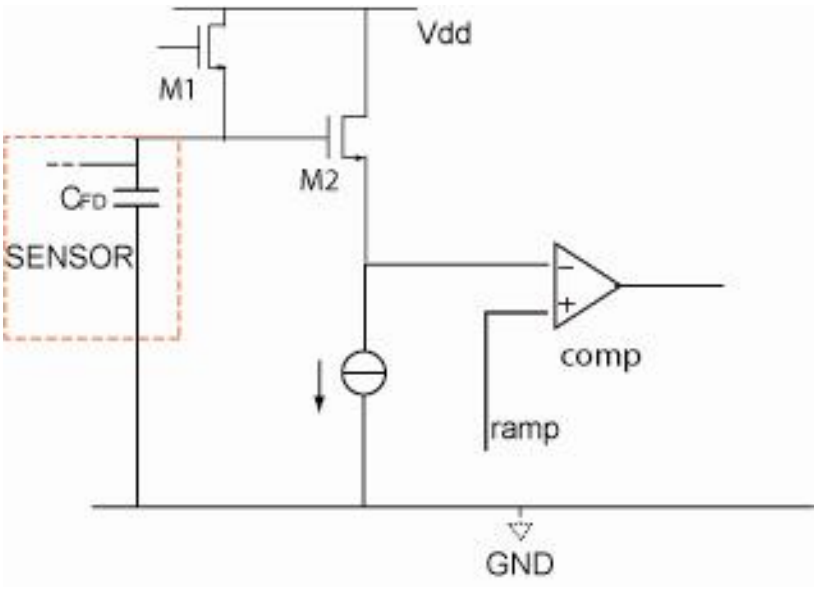


Figure 3-8 Block diagram for Digital CDS operation

After the reset of the floating diffusion, the comparator is powered up and a ramp is input to the positive input of the comparator to convert the reset voltage into digital domain. This digital code is stored into the in-pixel memory and is read out afterwards. After the charge transfer, the signal voltage is converted into the digital domain, stored into the memory and read. These two digital codes are subtracted to get the information of the light intensity.

The advantage of digital CDS is that the circuit is quite simple and, unlike analog CDS, no additional circuit is needed in the pixel. But there are some disadvantages:

1. Two ADC conversions have to be performed: one for the reset voltage and one for the signal voltage. This can increase both the power consumption and the total conversion time. On the other hand, for the conversion of the reset signal, the ramp does not need to cover the whole range from V_{low} to V_{high}, but rather a small range, which can help to keep the additional power and conversion time under control.
2. The power consumption for the digital memory part is mainly consumed by writing the data into the memory and read the data out. In our case, the power consumption for the writing of the data into the memory will only increase a little as the conversion for reset signal does not need take much time, the power consumption for the readout of the memory will be doubled.
3. The total quantization noise power is doubled. The reason is that the quantization noise for the reset voltage and signal voltage are not correlated.

Compared with digital CDS, the advantage of analog CDS is that only one conversion is needed, and thus the quantization noise power will not be doubled. But there are also some problems:

1. Large capacitance needs large area, which increases the total pixel area.
2. In order to achieve a fast settling time, the current that drives the source follower should be large. As this current source is placed in each pixel, this gives large power consumption and we need this in-pixel current to be very small and bias the source follower in the weak inversion. Digital CDS does not have this problem because the input capacitance of the comparator is very small.
3. The offset of the comparator cannot be compensated easily. In [2], the offset is compensated by adding two compensating capacitors at the output of the comparator and do auto-zero before the conversion. This of course complicates our design and is not suitable for a pixel circuit.

Taking into account all these considerations, a digital CDS operation is chosen for our design.

3.3 System level modeling

The system level modeling is done with an ideal sensor (no noise, no charge left in the pinned photo diode after charge transfer) and an ideal comparator (no noise and no offset). Figure 3-9 shows the voltage of the FD after the coarse conversion phase. As can be seen

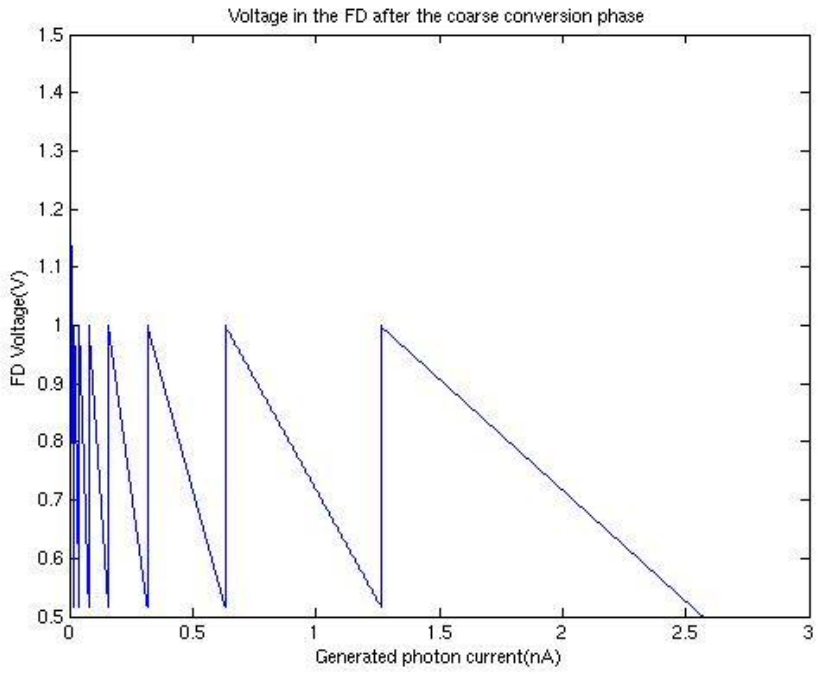


Figure 3-9 Hold voltage in the floating diffusion after the coarse conversion phase vs. the photon generated current

from the graph, with the increase of the generated photon current, the voltage in the FD first decreases to a certain level, then goes to 1 Volt (V_{mid} in Figure 3-3b) and decreases again.

Figure 3-10 shows the number of charge transfers with respect to the generated photon current. The higher the photon generated current, the less transfers are needed.

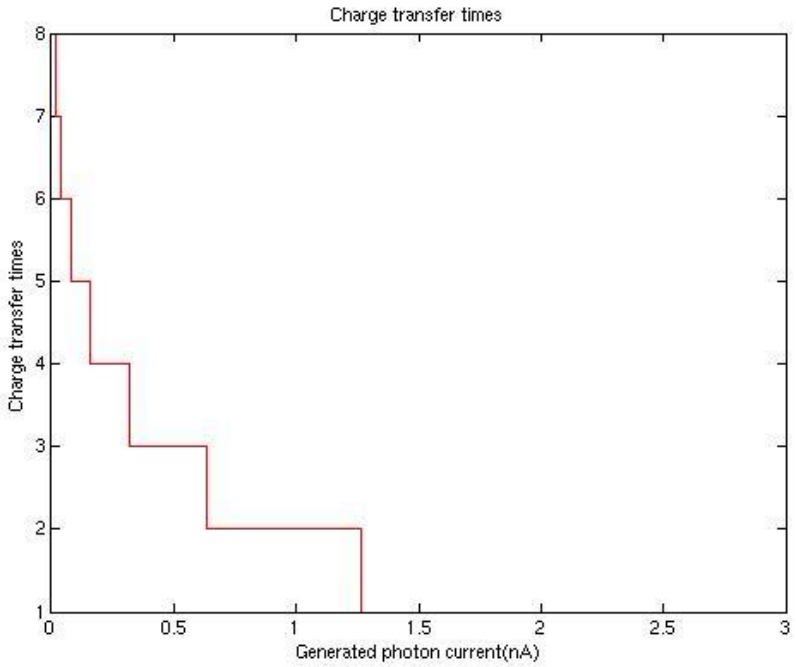


Figure 3-10 The charge transfer times vs. the generated photon current

Figure 3-11 shows the converted digital output with respect to the generated photon current. It can be seen that the total characteristic curve is very linear. The quantization step doubles every time when the integration time is halved.

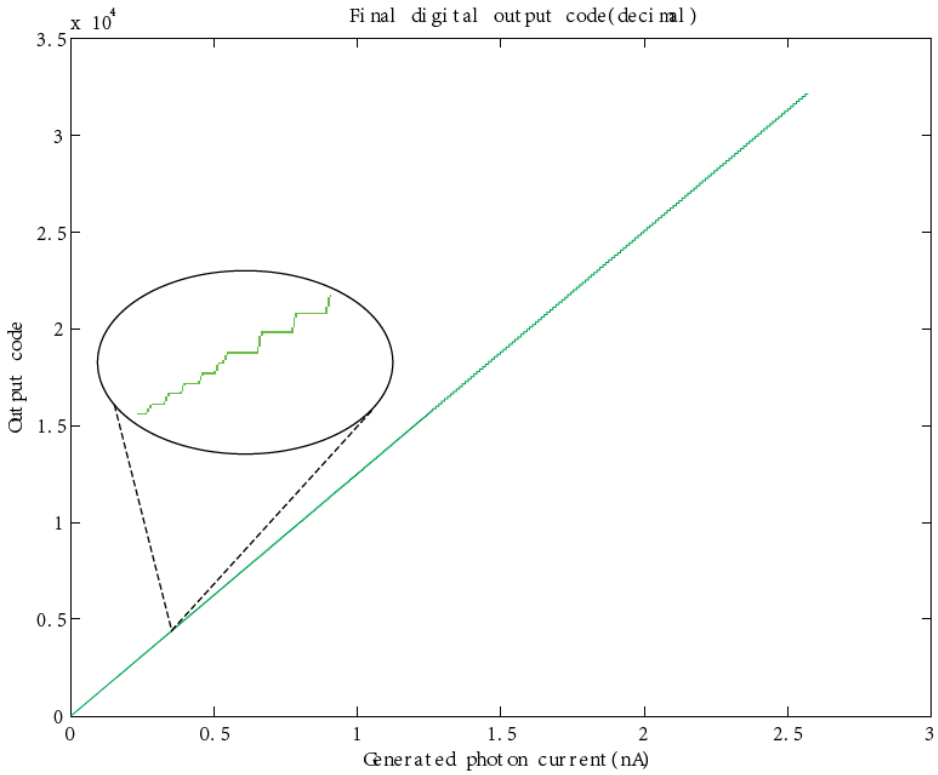


Figure 3-11 The output digital code vs. generated photon current

This can be explained as follows. Suppose that for 8 charge transfers, we can convert photocurrent in the range from 0 to A into the digital codes 0 to 255. When the generated photon current is larger than A, the number of charge transfers will be 7 and now we can quantize the current in the range from A to 2A into the digital codes 128 to 255. So for current ranging from A to 2A, the quantization step doubles. With this method, the highest photon generated current that can be measured increases from around 20pA with the longest integration time (800µs) to more than nearly 2.6nA with the shortest integration time (6.25µs).

In our design, the target frame rate is 1000fps, which means that the time for each frame is 1ms. The maximum integration time in our case is designed to be 800µs, and the remaining 200µs are used for the fine phase single slope conversion and the readout of the in-pixel memory.

Suppose that all the charges that are kept in the photo diode can be transferred in 3µs. For a DR enhancement of 7 bit, the minimum integration time is:

$$T_{min} = \frac{T_{max}}{2^k}$$

With a k value of 7, the calculated result is that the minimum integration time is 6.25µs.

3.4 Design of the pixel level ADC circuit

In this design, we choose the fine conversion phase to be 8 bit, and the DR enhancement is 7 bit. The total DR of the sensor is thus equivalent to 15 bits. Because each two samples give a one bit DR enhancement, for 7-bit DR enhancement we need to do 8 charge transfers ('k' value in Figure 3-5 & Figure 3-6 should be 7). In order to store the value of charge-transfer times, a 3-bit memory should be used, so plus the memory needed to store the value of the fine conversion the total in-pixel memory should be 11 bits.

The pixel block is shown in Figure 3-12. The output of the comparator is not connected to the memory blocks. Instead, a switch and two latches are placed in between. Also the output of the comparator is not used as the input to the digital AND gate directly. The reasons for this will be explained more detailed in the coming sections.

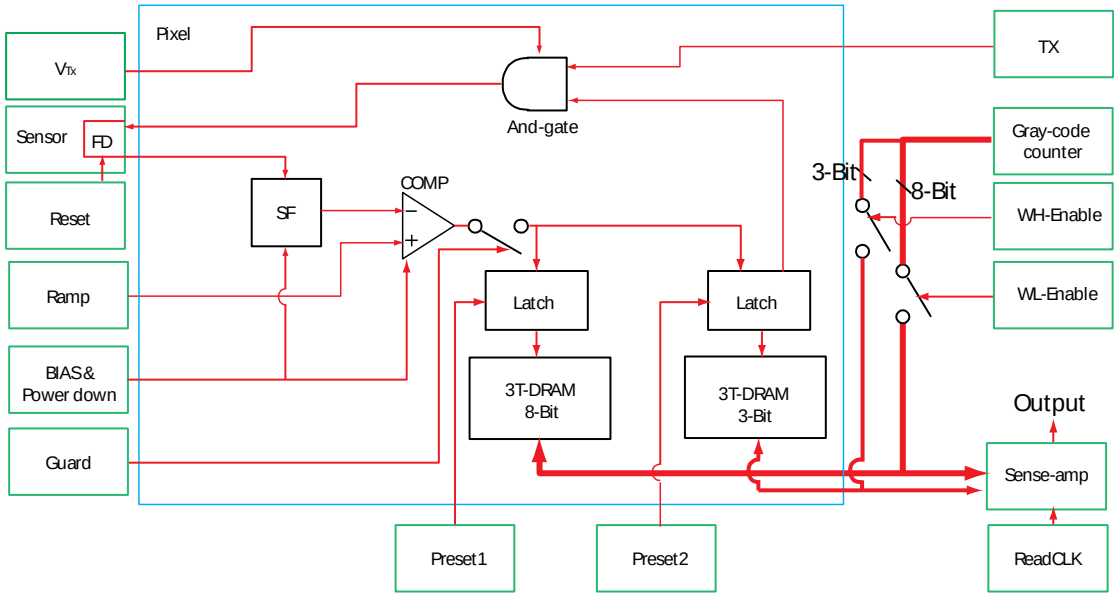


Figure 3-12 Block diagram of the pixel

Table 3-2 lists the main control signals.

Table 3-2 Explanation of main control signals for the pixel

Signal	Description
RESET	Input reset signal to reset the floating diffusion
PRESET_L	Initialize the latch for the coarse bit memory
PRESET_H	Initialize the latch for the fine bit memory
TX_IN	Global distributed transfer gate control signal
GUARD	Control signal to avoid the influence of the power-on of the comparator
BIAS_SF	Bias signal for the source follower
BIAS_COMP	Bias signal for the comparator

RAMP	Global distributed ramp signal
ReadCLK	Read signal for the memory

Figure 3-13 shows a timing diagram illustrating the operation of the pixel. After the reset of the floating diffusion, a ramp signal is input to the comparator for converting the reset signal. The result of this conversion is read with the 'ReadCLK'. After each charge transfer, the source follower and comparator are powered on for coarse bit conversion. After the seventh charge transfer, the coarse bits are read. In the fine phase, the signal voltage is converted and read out.

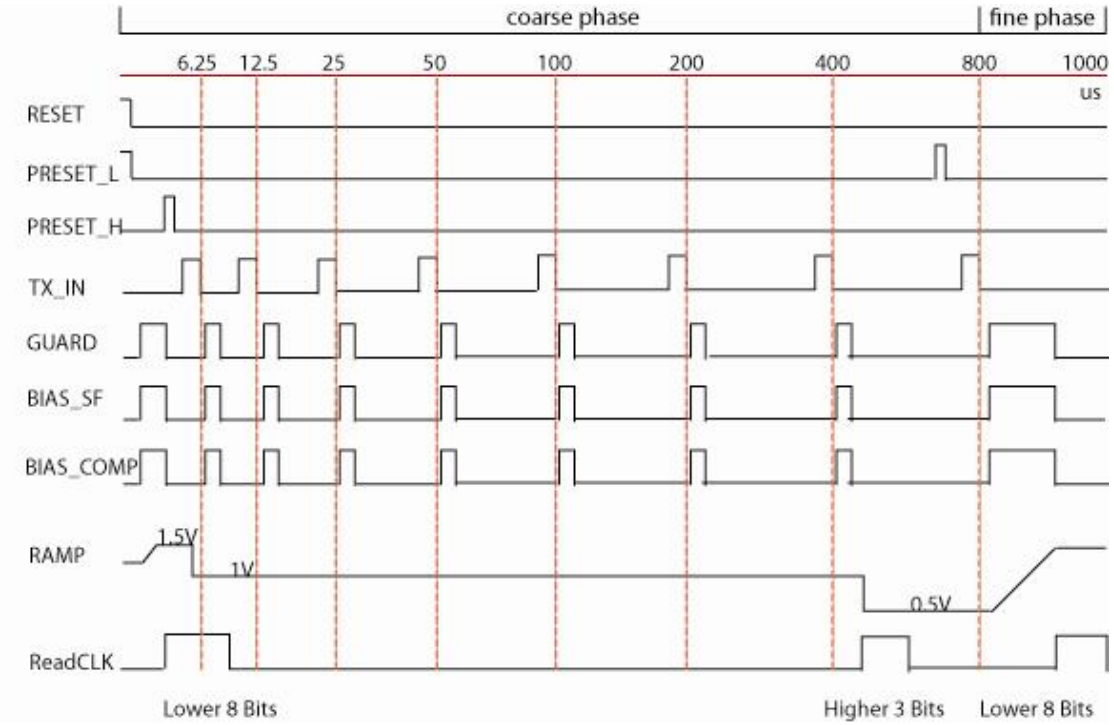


Figure 3-13 Main control signals in the pixel circuit

3.4.1 Comparator

As the comparator is placed in each pixel, the area of the comparator should be as small as possible. The schematic of the comparator that is used in the design is shown in Figure 3-14.

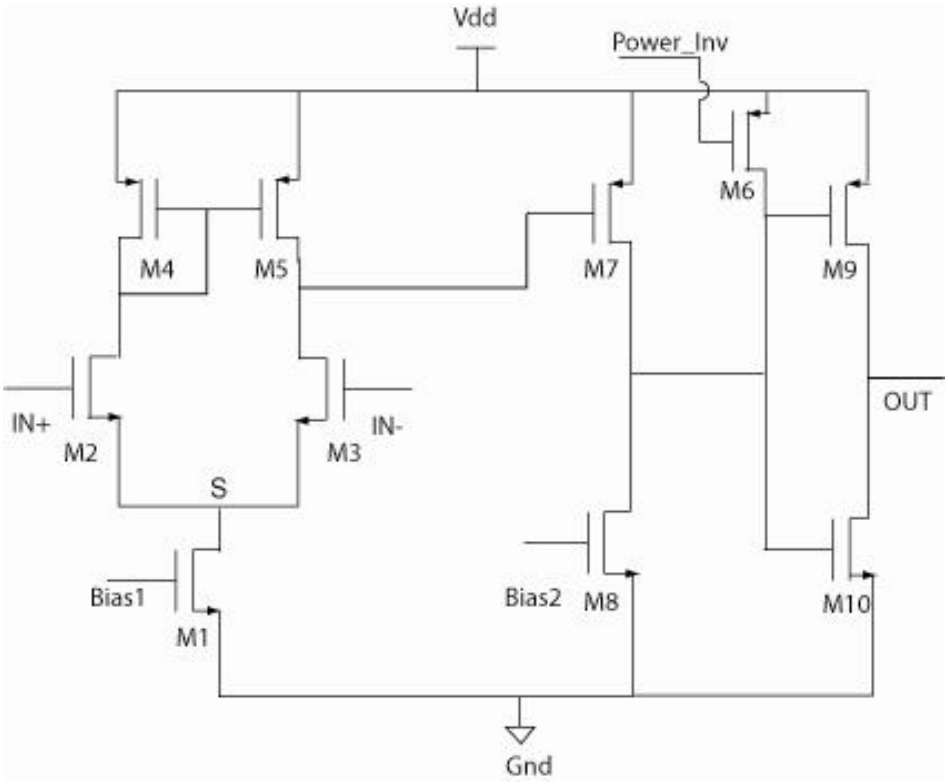


Figure 3-14 Schematic of the comparator

It consists of a differential gain stage and a single-ended gain stage, followed by a CMOS inverter. Because after each charge transfer, we only need to do one comparison, the bias voltage of the two current sources can be pulled down to ground when we don't need to do the comparison. During the power down, the output node of the second stage (drain of transistor M_7) is floating, and it is unpredictable. Because this voltage is used as the input of the third stage (inverter), if this voltage is in the region where both transistors M_9 and M_{10} are 'on', then there will be a direct connection from power supply to ground and a large current will flow through the last stage inverter. The easiest way to avoid this problem is to pull this floating node either to V_{dd} or ground when the comparator is powered down. Normally a NMOS transistor should be used because a PMOS transistor needs an additional NWELL. In our case, transistor M_6 is added to pull up the output of the second stage when the comparator is powered down. The reason for this is that the output of the comparator is connected to a latch which is triggered when its input is high, so we want the output of the comparator to be low when it is powered off.

Because the comparator is used in the pixel level, the speed of the comparator does not need to be very fast, but the power consumption of the comparator should be very low. In order to achieve this low power consumption, we decided to bias the input transistor of the

comparator in weak inversion, while the current sources are biased in strong inversion and saturation to achieve good matching.

The design for the comparator should make sure that for the whole common mode input range (in our case is 0.5-1.5V), all the transistors work in correct operation region. For a certain common mode input, the current mirror of M_4 & M_5 should be kept in saturation for good matching. For a transistor (M_2 and M_3)) to work correctly in weak inversion, the drain source voltage of that transistor should be larger than 4-6V_T [35]. This gives the following equation:

$$V_{ci} - V_{gs2} + 6V_T \leq V_{DD} - (V_{th4} + V_{sat4})$$

Additionally, for the lowest common mode input the tail current source M_1 should be also kept in saturation. This gives the following equation:

$$V_{ci} - V_{gs2} \geq V_{th1} + V_{sat1}$$

These two formulas give the limit for sizing of the transistors. The simulation result of the comparison time versus the input voltage difference is shown in Figure 3-15.

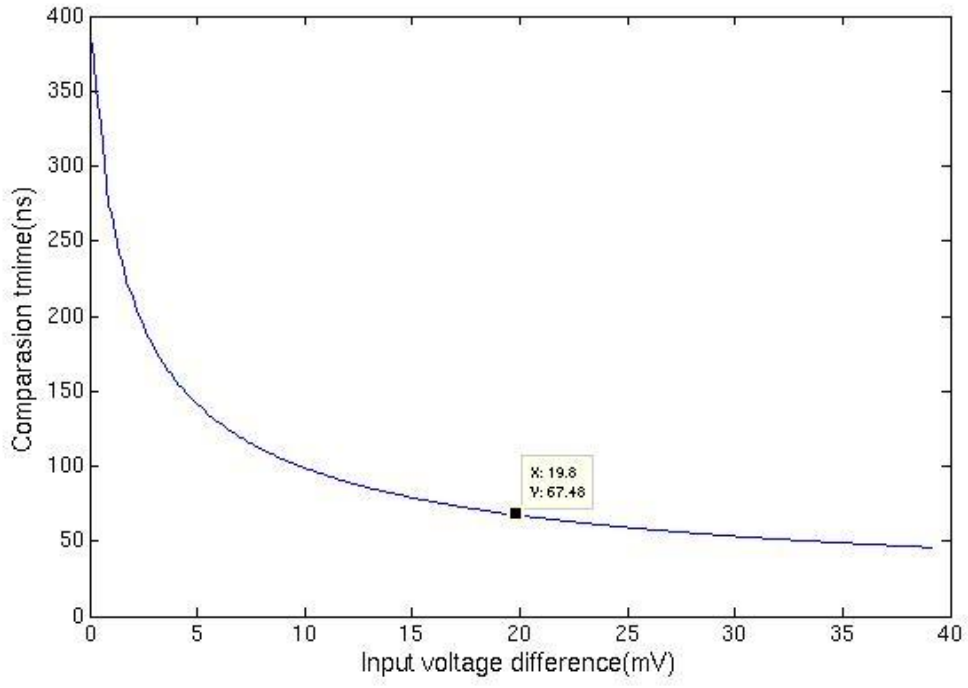


Figure 3-15 Comparison time with respect to the input voltage difference

In our case, the common mode input of the comparator is 1V, and for an '8-bit' conversion, the LSB of the ADC is around 3.9mV. As can been seen in the figure, for an input voltage difference of half an LSB, the worst case comparison time is around 70ns.

Taking into account the possible coupling between the output of the comparator and the metal wires used in the pixel and the mismatch between the different comparators, we decide to make the total conversion time for an 8-bit conversion with 1V input to be 25.6μs, which is well within the specifications.

The noise analysis of the comparator is shown in Table 3-3. As can be seen from the table, the main contribution is due to thermal and flicker noise of the input transistors. The thermal noise can be reduced by increasing the biasing current, but this will increase the power consumption. The flicker noise can be reduced by increasing the size of the input transistor, but this will increase the size of the transistor. The total in-band (noise) input referred noise of the comparator is 140μV, which is quite low compared with the quantization noise (rms value of 1.128mV) of the 8-bit ADC.

Table 3-3 Noise contribution of each transistor for the comparator

Name	Type	Noise contribution(V_{rms})	Percentage (%)
M2	thermal	0.0184	23.35
M3	thermal	0.0184	23.34
M2	flicker	0.0166	18.83
M3	flicker	0.0166	18.83
M4	thermal	0.0105	7.51
M5	thermal	0.0095	6.16

3.4.2 Source follower and bias current source

The floating diffusion of the sensor could be connected directly to one input of the comparator as shown in Figure 3-16. But this does not work correctly: the voltage at the floating diffusion node changes with the ramp signal. The reason for this is that when the ramp is applied to the positive input of the comparator, the voltage of the source node of the input transistor changes as can be seen in Figure 3-17. Because of the gate-source capacitance of the input transistor, there will be charge redistribution between the floating diffusion and the capacitance C_{gs} shown in Figure 3-18. This causes the changes in the floating diffusion voltage.

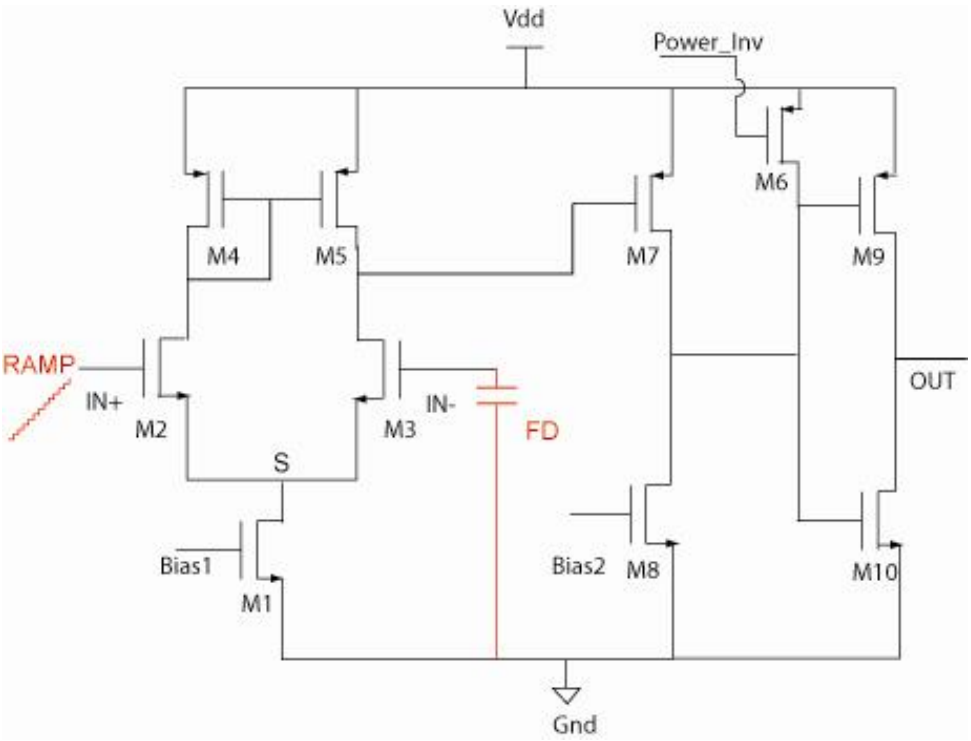


Figure 3-16 Floating diffusion connected directly to the input of the comparator

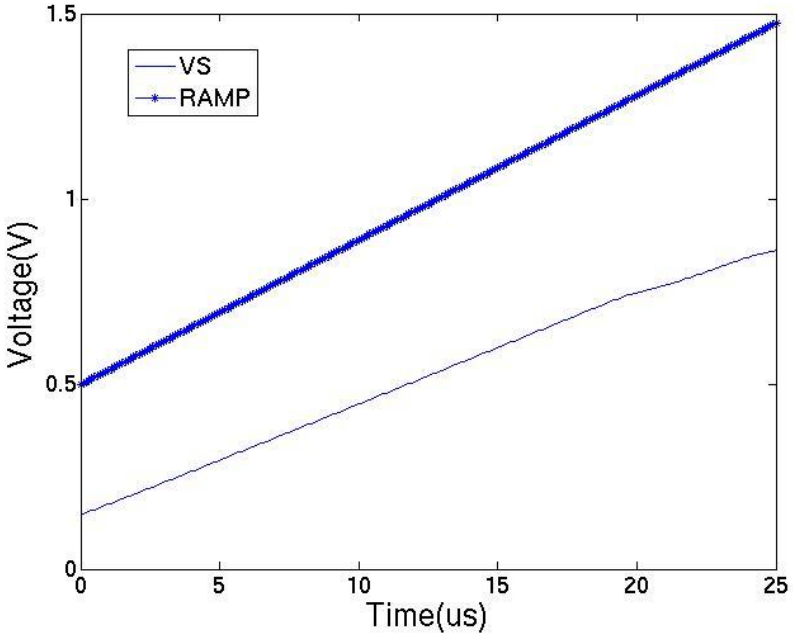


Figure 3-17 Output voltage at node S with respect of the ramp during a conversion

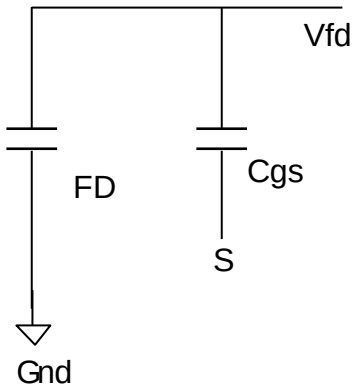


Figure 3-18 Model for the charge redistribution between the FD and Cgs

The solution is to add a source follower between the floating diffusion and the negative input of the comparator. The load of the source follower is the input capacitance of the comparator. This capacitance is very small (in our case it is 5fF) and so the current that is needed to drive it does not need to be very large. We can bias the source follower in the weak inversion and save power. The current source still needs to be operated in the saturation region for good matching. Because the current is so small, in order to keep the current source operating in saturation, the W/L ratio of the transistor should be very small.

Because of the small current the source follower has a small g_m, and high thermal noise. To reduce this noise problem, the best way is to increase the load capacitance and reduce the bandwidth of the source follower (limited by the settling time). The total output referred noise is 896μV, much smaller than half an LSB. In the design, the main purpose of the digital CDS is to remove the mismatch of the source follower threshold voltage and the offset of the comparator.

The output with respect to the input of the source follower is shown in Figure 3.19. For the comparator input range from 0.5V to 1.5V, the input of the SF (FD Voltage) should range from 1V to 2.3V, so for the reset transistor of the floating diffusion we use a thick oxide transistor and 3.3V power supply.

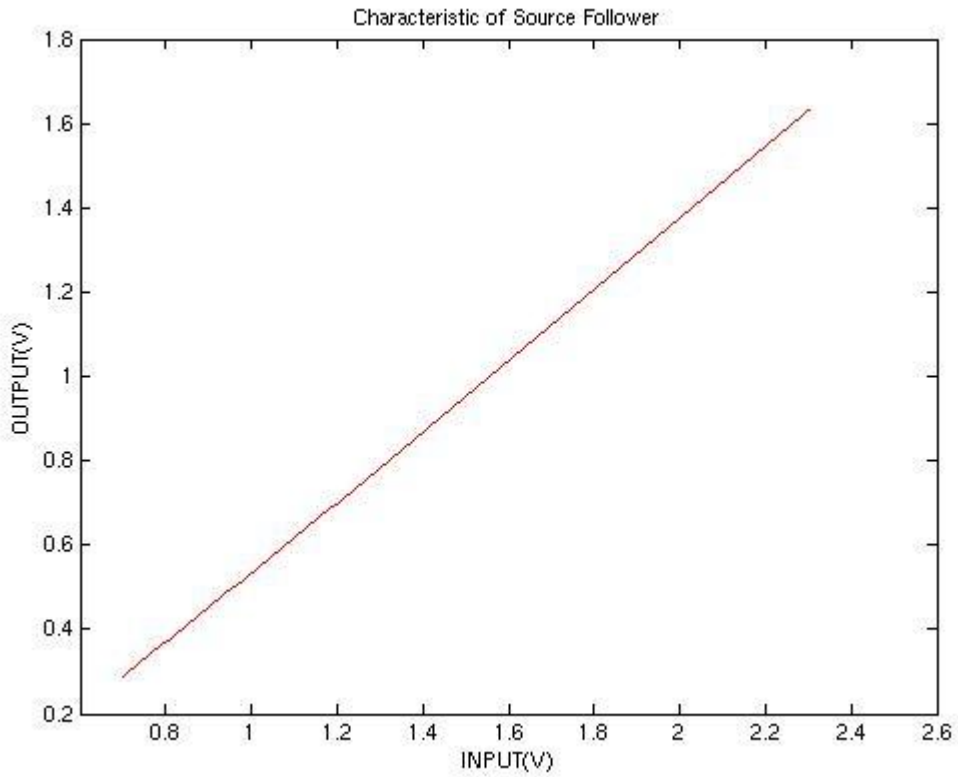


Figure 3-19 Simulation results of the in-pixel source follower

3.4.3 Bias and power down circuit

Figure 3-20 shows the schematic for the bias and power down circuit.

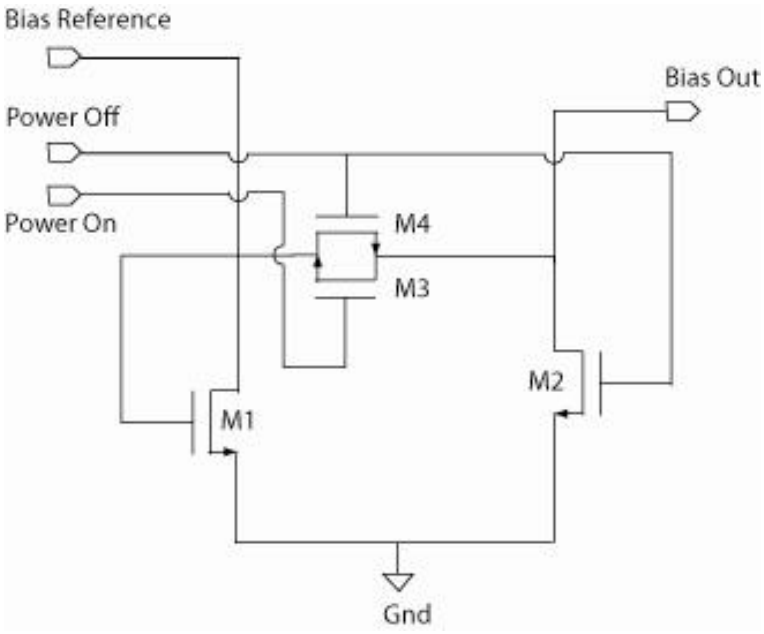


Figure 3-20 Schematic of the bias and power down circuit

This circuit shown in Figure 3-20 is used for both the comparator and the source follower. The comparator should be powered up after the correct settling of its negative input. This includes both the settling of the source follower bias voltage and the settling of the source follower output. Because after the power-on of the comparator, its bias voltage must also settle before it can work correctly, during the settling time the comparator output does not give a correct result. This may trigger the latch that is connected to the output. So we decide to use a guard signal as shown in Figure 3-12 to avoid this.

The timing slot for the bias of the source follower, the comparator and the guard signal is shown in Figure 3-21.

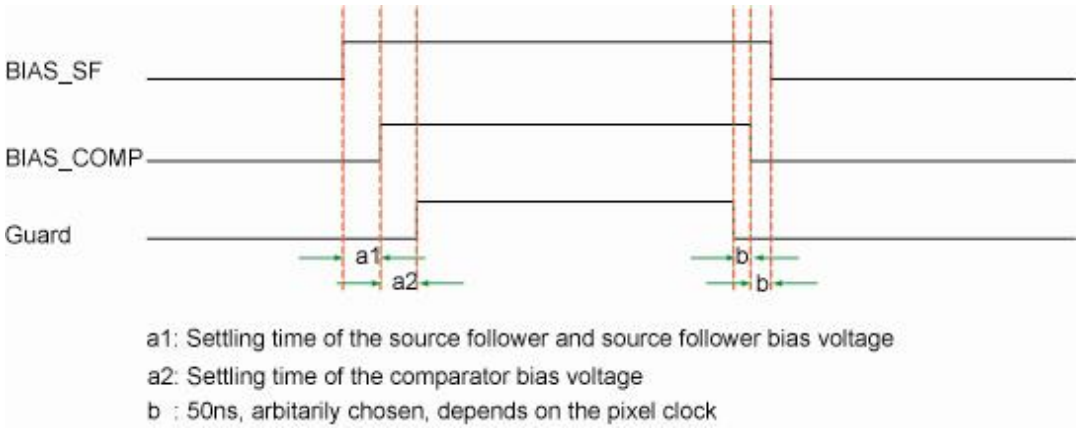


Figure 3-21 Detailed timing for bias of the source follower, comparator and the guard signal

3.4.4 In-pixel memory

As we use 8-bit ADC to convert the reset and signal voltages into the digital domain, an 8-bit in-pixel memory should be used to store the values of the reset and signal voltages. For the coarse bits we need also a 3-bit memory. A good way to save area is to reuse the memory for the two different conversion phases. But this is not suitable for our design. As can be seen from the sequence in Figure 3-22, the coarse conversion happens in parallel with the readout of reset signal. In the readout of the reset signal, the value that is stored in the memory should be kept unchanged during the whole readout phase, while the coarse conversion needs to write the gray-code into the memory. This means that in our design, the memory cannot be reused. In total, an 11-bit in-pixel memory is needed into each pixel, with 3-bit to store the value of the coarse bit and 8-bit memory to store the value of the reset voltage and signal voltage.

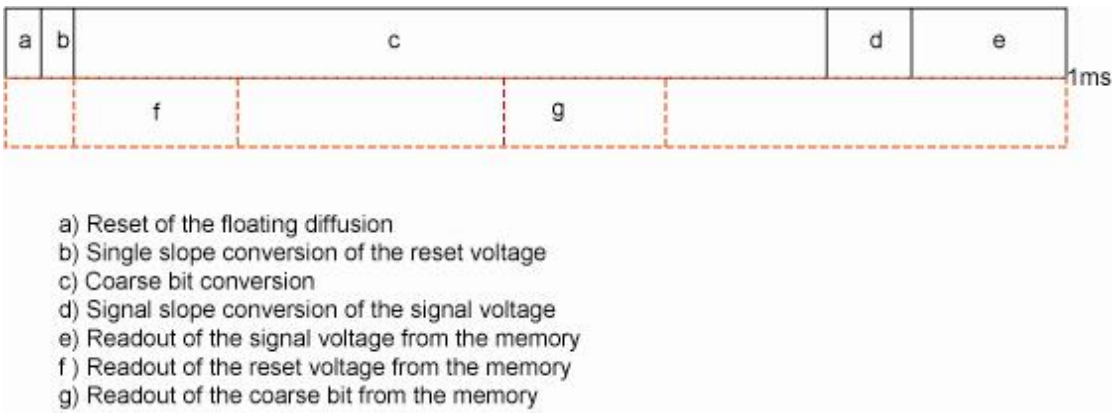


Figure 3-22 Time slots for the operation of the pixel circuit

In our design, 3T DRAM cells are chosen as is shown in Figure 3-23.

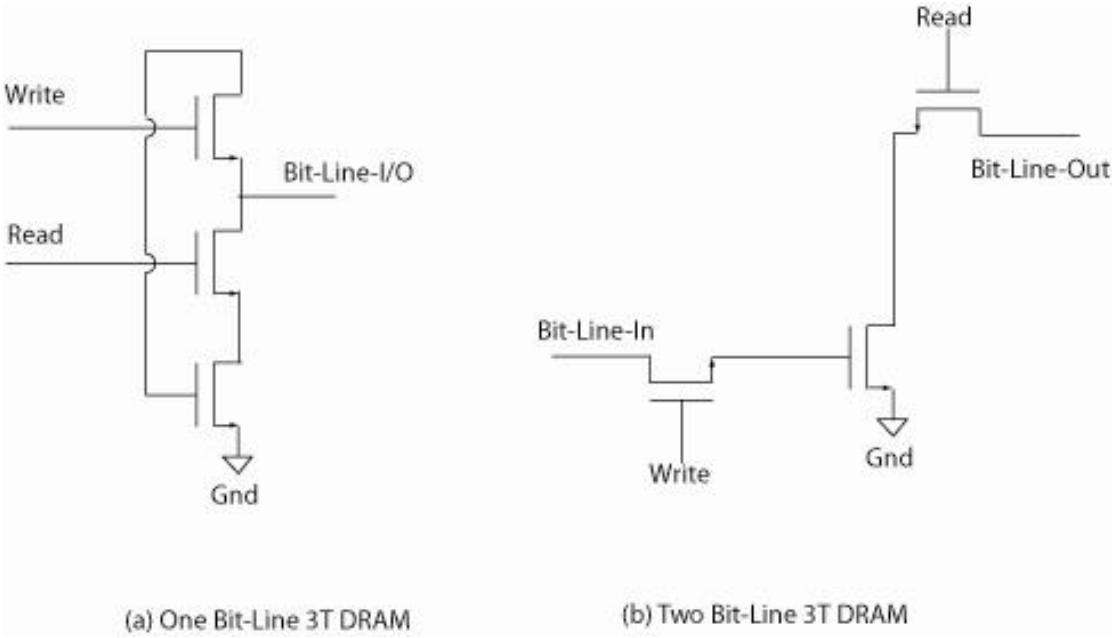


Figure 3-23 Memory circuit that is used in the pixel

Figure 3-23(a) shows the one bit line 3T DRAM that is chosen for the 8-bit memory. When the "write" signal goes low, the gray-code that appears in the bit-line I/O is latched and the data is stored into the gate source capacitance of transistor M_3. Before reading the memory cell, the write select transistor should be switched off, and then the bit line is pre-charged to a high voltage (V_{dd} or $V_{dd}/2$). When the "read" select signal goes high, the current flowing through transistors M_2 and M_3 will depend on the data that is stored on the gate source capacitance of M_3, and the bit line will either be discharged or not.

Figure 3-23 (b) shows the memory cell that is used for the coarse bits. The only difference compared with the structure shown in Figure 3-23(a) is that two different bit lines are used instead of one. The reason for this is explained in the following.

Take Figure 3-23(a) for example, when a digital '1' is stored on the gate source capacitance of M_1, because of the non ideality of the MOS transistor, the off resistance of transistor M_1 will not be infinite. If the voltage on the bit-line is at ground, then the gate capacitance will be discharged through transistor M1. The simulation result for this is shown in Figure 3-24. The same situation will occur with a '0' stored on the gate capacitance and the bit-line at V_{dd}. In order to decrease the charging or discharging speed, the length of the access transistor M_1 should be increased to increase the off resistance of the transistor. Another way is to increase the gate source capacitance of the transistor M_3, which will increase the charge that is stored. Both of these two methods will result in an increase of the area of the memory cell.

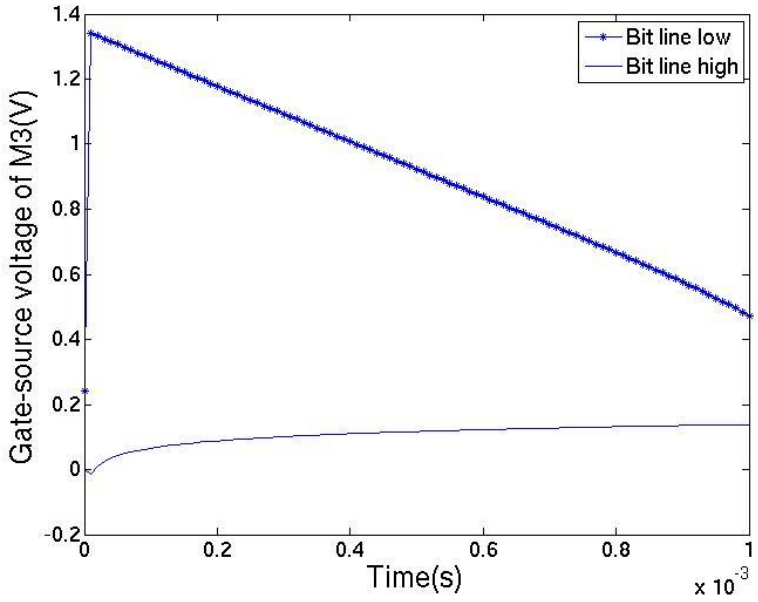


Figure 3-24 The gate voltage of M3 with respect to time

Using the memory cell in Figure 3-23(a), we can only read the data that has been latched into the memory (the write transistor is open). So in our application with memory cell in Figure 3-21(a) we need to read the coarse bits after the coarse conversion which lasts 800us. Or we can latch a '0' (stands for 8 times charge transfer in the coding) after the seventh time charge transfer if the comparator output does not flip (means there will be 8 times charge transfer). To achieve this we can input a '1.5V' in the positive input of the comparator to let the comparator output flip and latch the '0', but this operation will also give a wrong result of the control signal for the transfer gate(the eighth charge transfer pulse does not appear).

The benefit of using the memory cell of Figure 3-23(b) is that we can 'read' while 'writing'. In our application, we can read the coarse bits after nearly half of the coarse conversion time. This is because, after the charge transfer at 400us, if the comparator output does not flip (total 8 times charge transfer). So what we can do is to input a gray-code '0', and readout this '0' out (comparator output does not need to flip).

The name and description for the control signals for this memory are shown in Table 3-4.

Table 3-4 Name and description for the control signals of the in-pixel memory array

Name	Description
Write_H	Write signal for coarse bits
Write_L	Write signal for fine bits
RS0	Read select signal for D<0:3>
RS1	Read select signal for D<4:7>
RS2	Read select signal for D<8:9>
RS3	Read select signal for D<10>
BL<0:11>	BL<0:7> is the bit-line of the memory for fine bits (shown in Figure 3-23(a)). BL<8:10> is the Bit-Line-Out signal for coarse bits (shown in Figure 3-23(b)). BL<11> is a dummy bit-line used for the sense-amplifier.
DI<8:10>	The Bit-Line-In signal for coarse bits(shown in Figure 3-23(b))

3.4.5 Latch

The reason for placing a latch in front of the memory is that for the coarse conversion, we want to latch the digital code at the first time that comparator output flips. If there is no latch, then during the coarse bit conversion, every time the comparator flips, the gray-code that appears in the bit-line will refresh the memory and overwrite the result that had been latched previously.

The schematic of the latch circuit that is used in our design is shown in Figure 3-26. It contains five minimum sized transistors.

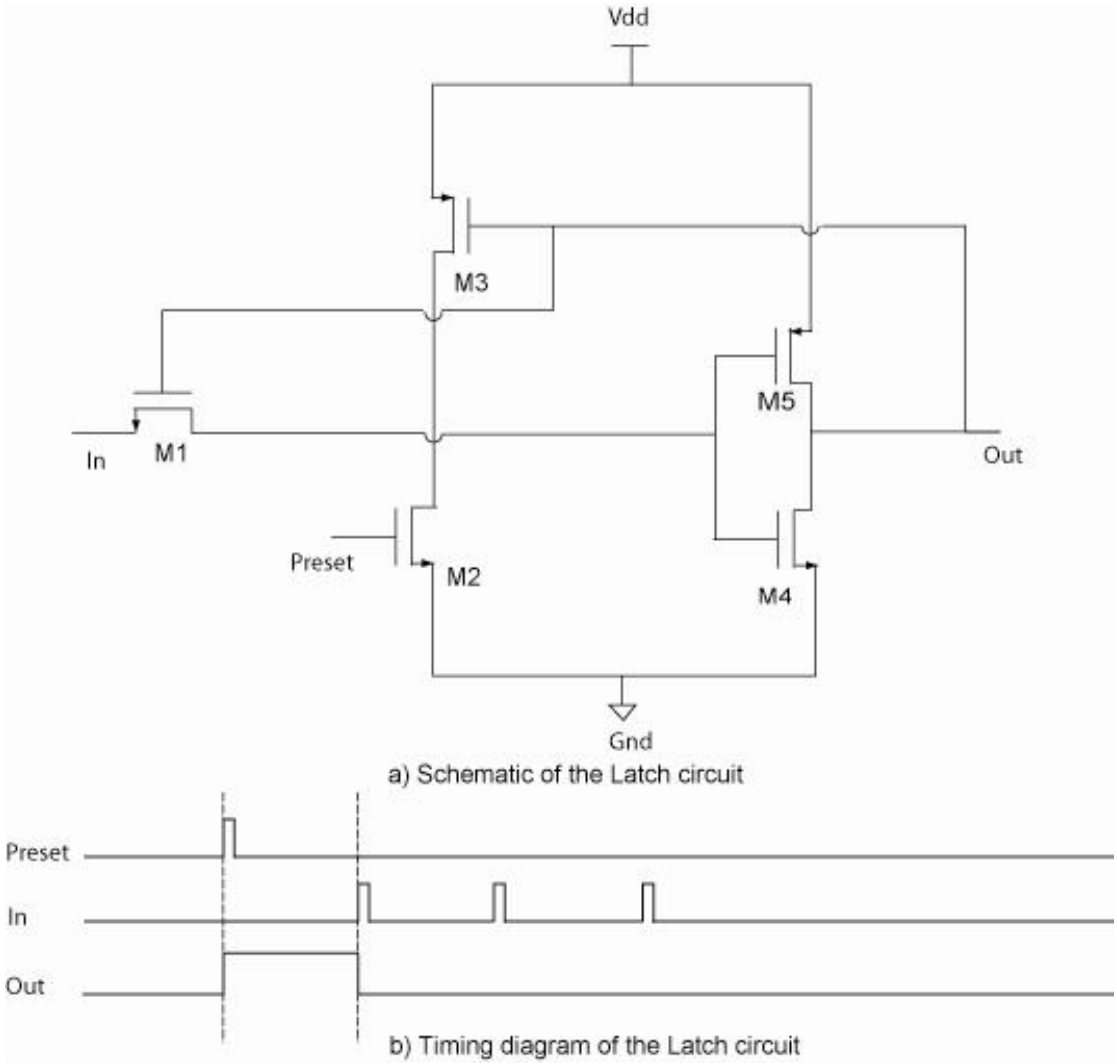


Figure 3-25 Schematic and timing diagram of the latch circuit

In the beginning a pulse is applied to the 'Preset' signal to initialize the latch. After the initialization of the latch, the drain of transistor M1 is set to ground, the output of the latch is set to V_{dd}, and transistor M_1 is closed. When the input signal 'In' is high, the output flips and the transistor M_1 is open. Whatever happens at the input, the output does not change.

3.4.6 Transfer gate control

The local transfer gate control signal is determined by the global control signal (TX_IN) and the local comparator output result, as is shown in Figure 3-26.

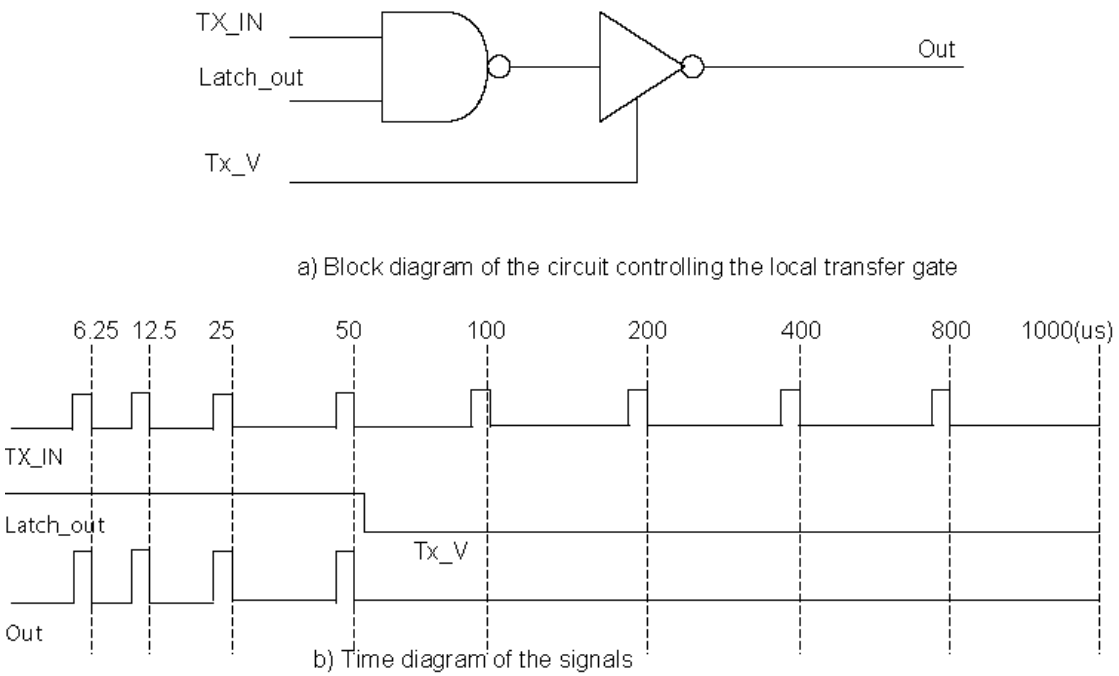


Figure 3-26 Blocks generating local transfer gate control signal

In order to increase the flexibility for controlling the transfer gate of local sensor, a DC voltage signal TX_V is used to supply the inverter as shown in Figure 3-26(a).

3.5 Layout considerations

The final pixel layout is shown in Figure 3-27, the total area is 18*18um^2.

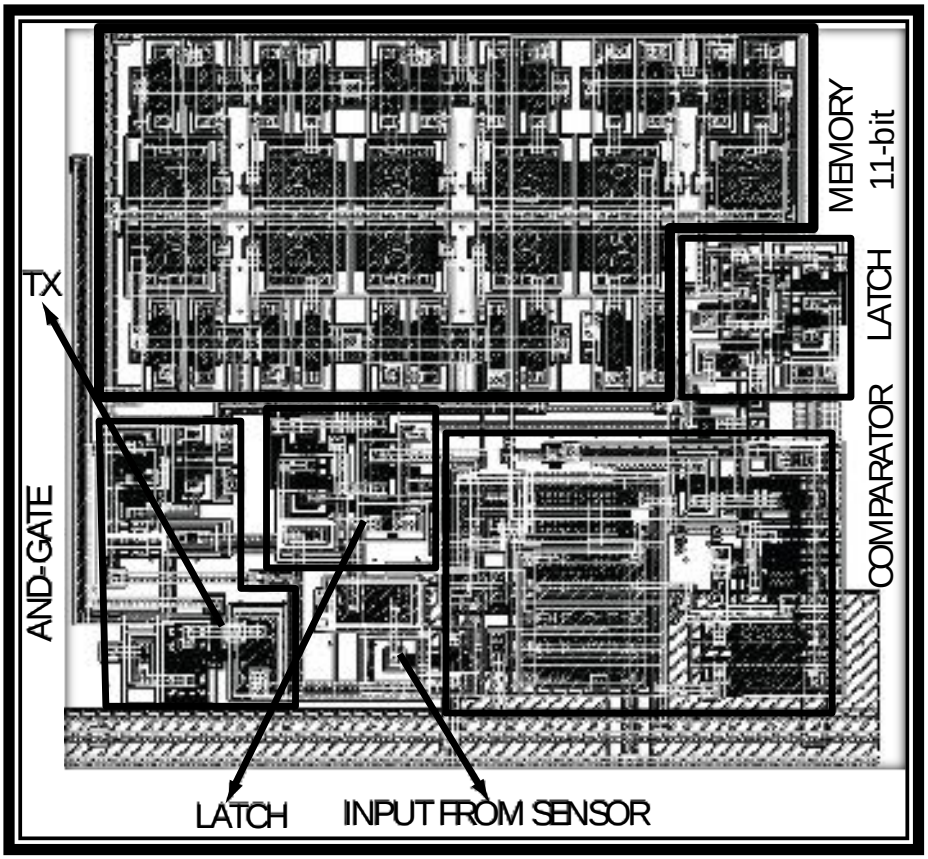


Figure 3-27 Layout of the total pixel

The floating diffusion of the sensor is connected to the input of the source follower via a bump-bond as have been shown earlier in Figure 3-1. Because there are digital bit-lines crossing the pixel, the cross-talk between the digital and analog signals will be important. We need to protect the floating diffusion connection very well so that the digital signals do not influence the floating node. Also, the bias of the source follower should be well protected. Any variation on the bias signal will directly influence the output of the source follower which is connected to the input of the comparator. In the layout, all these signals are either covered or surrounded with power line, ground or other AC ground signals. Metal 3 is used for the control signals and metal 4 is used for digital bit-lines. Metal 5 is used for ground and power line. This also shields the sensor from the digital bit-lines because metal 6 is used for the bond to the upper sensor layer.

Table 3-5 summarizes the signals that are used in each pixel.

Table 3-5 Description of the signals that is used in each pixel

Name	Type	I/O	Description
Vdd!	Power supply	IN	Power supply of the system

Gnd!	ground	IN	Ground of the system
TX_V	Power supply	IN	Power supply for the output of the transfer gate
BL<0:7>	Digital	INOUT	Bit line for fine memory bits
BL<8:10>	Digital	OUT	Bit-Line-Out signal for coarse memory bits
BL<11>	Digital	OUT	Dummy Bit-Line-Out signal for coarse memory
DI<8:10>	Digital	IN	Bit-Line-In signal for coarse memory bits
RS0	Digital	IN	Read select signal for D<0:3>
RS1	Digital	IN	Read select signal for D<4:7>
RS2	Digital	IN	Read select signal for D<8:9>
RS3	Digital	IN	Read select signal for D<10>
GUARD	Digital	IN	Guard signal for the comparator
POWER_INV	Digital	IN	Control signal for the third stage of the comparator
PRESET_H	Digital	IN	Latch control signal(coarse bits)
PRESET_L	Digital	IN	Latch control signal(fine bits)
TX_IN	Digital	IN	Globally distributed transfer gate control signal
BIAS_COMP	Analog	IN	Bias voltage for the comparator placed in each pixel
BIAS_SF	Analog	IN	Bias voltage for the in-pixel source follower
RAMP	Analog	IN	Globally distributed ramp signal
SENSOR_IN	Analog	IN	Input voltage signal for the sensor
TX_OUT	Analog	OUT	Output control signal to the transfer gate of the sensor

3.6 Simulation results

Figure 3-28 shows the simulated post-layout INL curve of the ADC. The result shows very good intrinsic linearity of the 8-bit ADC, which is below the target specification of 0.5LSB. Figure 3-29 shows the DNL curve of the ADC (target specification: 0.5LSB), the curvature of the DNL curve is due to that the comparator speed becomes slightly faster when the common mode input voltage goes up. Figure 3-30 shows the number of charge transfers with respect to the generated photo-current. The dotted line shows the simulation result. The slight difference to the system modeling result is because the conversion-gain that is

used for the system modeling and simulation is not exactly the same (due to rounding of the number). But this is not really a problem, as shown in Figure 3-31. At the time 2^2T, the signal voltage $V_{signal1}$ is (slightly) below the reference voltage V_{mid}, ideally, no charge transfer will be activated at 2^3T, but due to noise or offset problems, the output of the comparator is not correct, and an additional charge transfer is activated at 2^3T. Because the light intensity corresponds to the two voltage difference (V_{signal} & V_{reset}) divided by the total integration time, the final calculated value of the light intensity will not have much influence. The only thing that we need to take care is that the signal voltage $V_{signal2}$ will not be below the level of V_{low}. This can be guaranteed by resetting the FD at a certain level that V_{reset} is slightly smaller than V_{high}(Also for the purpose of convert the V_{reset} in digital domain).

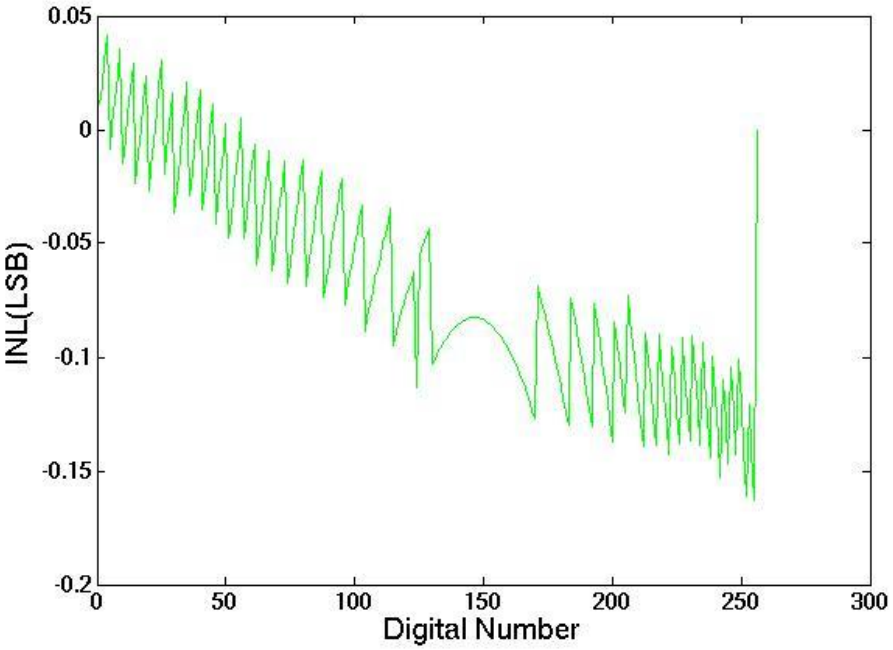


Figure 3-28 INL characteristic of the 8-Bit ADC

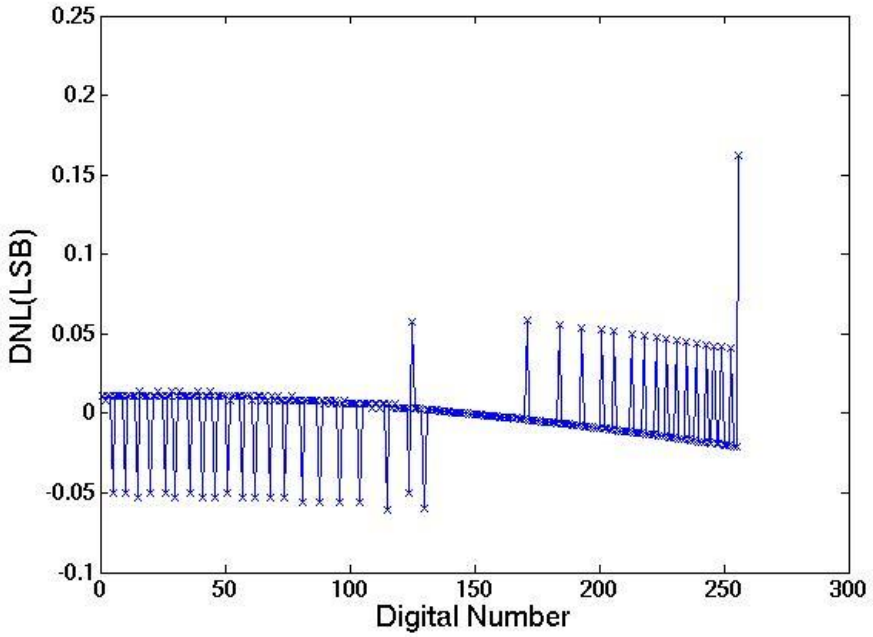


Figure 3-29 DNL characteristic of the ADC

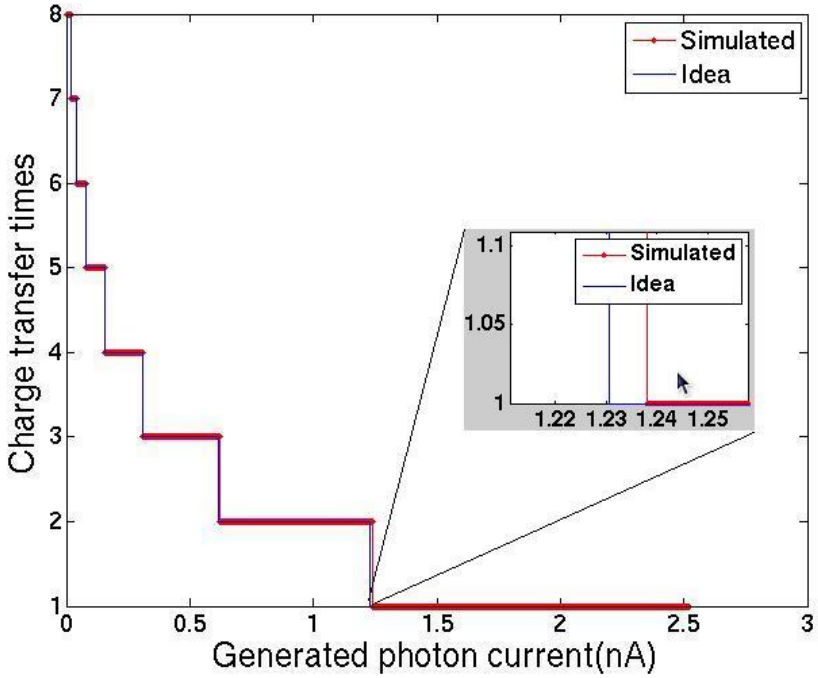


Figure 3-30 Charge transfer times (coarse bit) with respect to the generated photon current

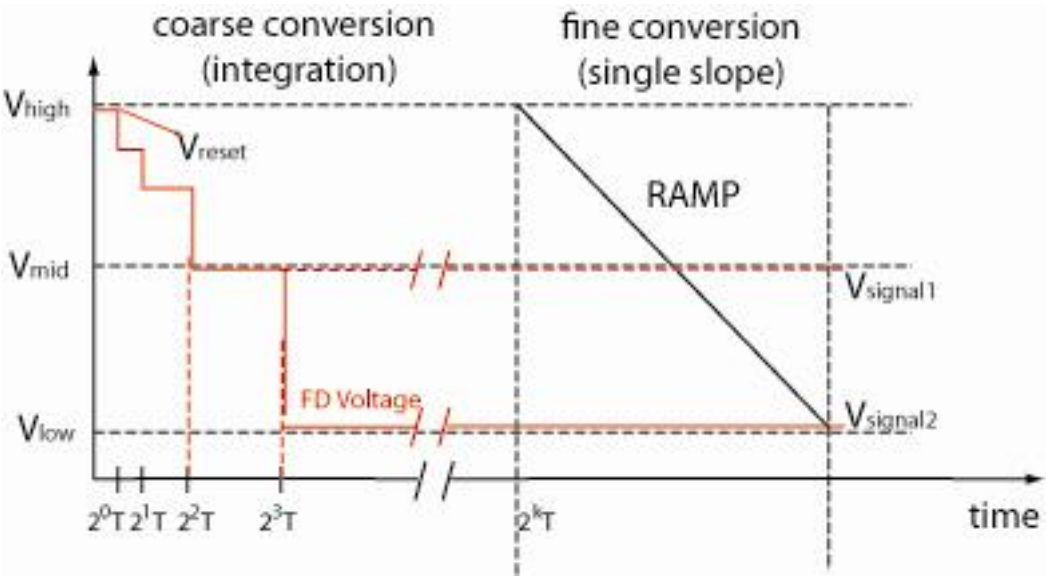


Figure 3-31 Wave form when the coarse bit conversion is not correct due to noise, offset, etc.

The SNR curve of the sensor is shown in Figure 3-32, the noise include photon shot noise, doubled quantization noise because of digital CDS, 1/f noise and thermal noise of the SF and the comparator. Maximum SNR is 38.91dB, and the DR of the sensor is 85.29dB (13.88bits) with 42.24dB (6.72bits) enhanced by the method described in this thesis.

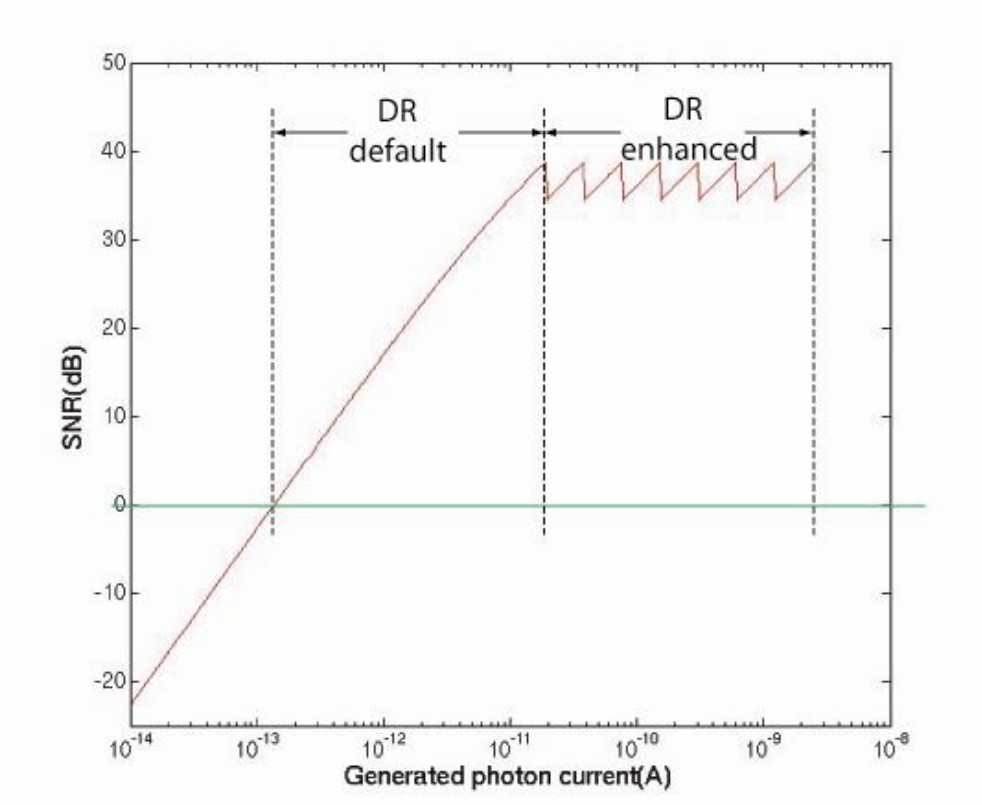


Figure 3-32 SNR curve of the image sensor

The bias current used for the source follower is 200nA, and for the comparator it is 1μA. Because of the DR enhancement, there will be 7 times additional "power on" of the source follower and comparator, with the source follower being "on" for 400ns and the comparator for 200ns. The single slope conversion for the reset voltage takes 3μs, for the signal voltage takes 25.6μs.

During the single slope ADC operation, the average current is 1.152μA. So the power consumption for the single slope ADC is:

Power_ADC = 1.8V*1.152μA = 2.074μW.

Compared with the quantization noise, the input referred noise of the comparator is negligible, so the $ENOB_{SNR}$ of the ADC is 8 bits. We calculated the FOM of the single slope ADC is:

$$\mathrm{FOM_{SNR}} = \frac{\mathrm{Power}}{2^{\mathrm{ENOB_{SNR}}} \times \mathrm{F_s}} = \frac{2.074 \times 10^{-6}}{2^8 \times 3.9 \times 10^4} = 0.208pJ/conversion$$

The total analog power consumption for each pixel is:

Power_pixel = (P_{single_slope}+ $P_{DR\_enhancement}$)

= ((25.6μs+3μs)*1.2μA*1.8V+ (0.4μs*0.2μA*7+0.2μs*1μA)*1.8V)/1ms=63nW

The additional power because of the 7-bit DR enhancement is:

Power_addtional = $\frac{P_{DR\_enhancement}}{P_{single\_slope}} = \frac{1.368}{61.776} = 2.2\%$

FOM for the total pixel is:

$$\mathrm{FOM_{DR}} = \frac{\mathrm{Power}}{2^{\mathrm{ENOB_{DR}}} \times \mathrm{F_s}} = \frac{63 \times 10^{-9}}{2^{13.88} \times 1 \times 10^3} = 0.004pJ/conversion$$

The specifications for the final designed pixel are shown in Table 3-6.

Table 3-6 Specifications for the designed pixel

Specification	Designed	Target
Pixel Size	18μm x 18μm	20μm x 20μm
Pixel Type	Pinned Photodiode	Pinned Photodiode
SNR(single slope)	8 bits	8 bits
DR(sensor)	13.88 bits	12 bits
Frame rate	1000 fps	1000 fps
FOM_{SNR}(single slope)	0.208pJ/conversion	0.5pJ/conversion
FOM_{DR}(sensor)	0.004pJ/conversion	0.5pJ/conversion

INL(LSB)	0.23(worst case)	0.5
DNL(LSB)	0.23(worst case)	0.5

3.7 Pixel data readout

A block diagram of the read out architecture of the pixel memory cells is shown in Figure 3-33. A number of n memory cells are connected to the bit line BL1 and the other n memory cells are connected to the bit line BL2. Two dummy memory cells are connected to the bit-line. Because the 3T-DRAM cell has only one output, in order to read the data that is stored in the memory, a reference cell (dummy cell) should be connected to the other bit line, as is shown in the figure.

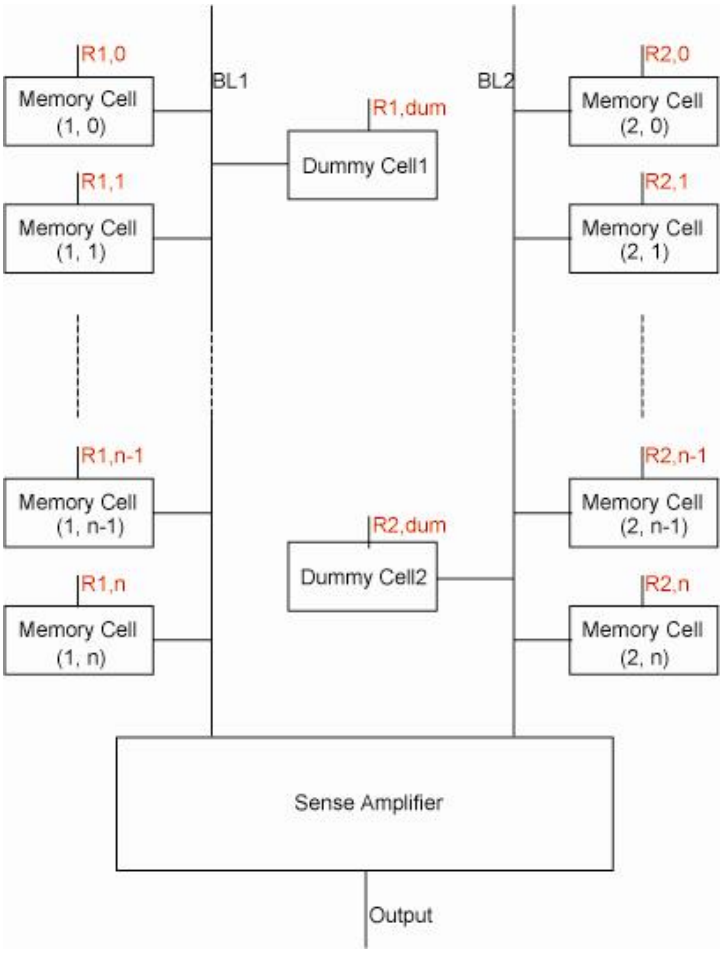


Figure 3-33 Block diagram for memory cell readout

Figure 3-34 shows the timing diagram of the control signals in the block. Every time when one cell is connected for readout, the dummy cell that is connected to the other bit-line is also connected.

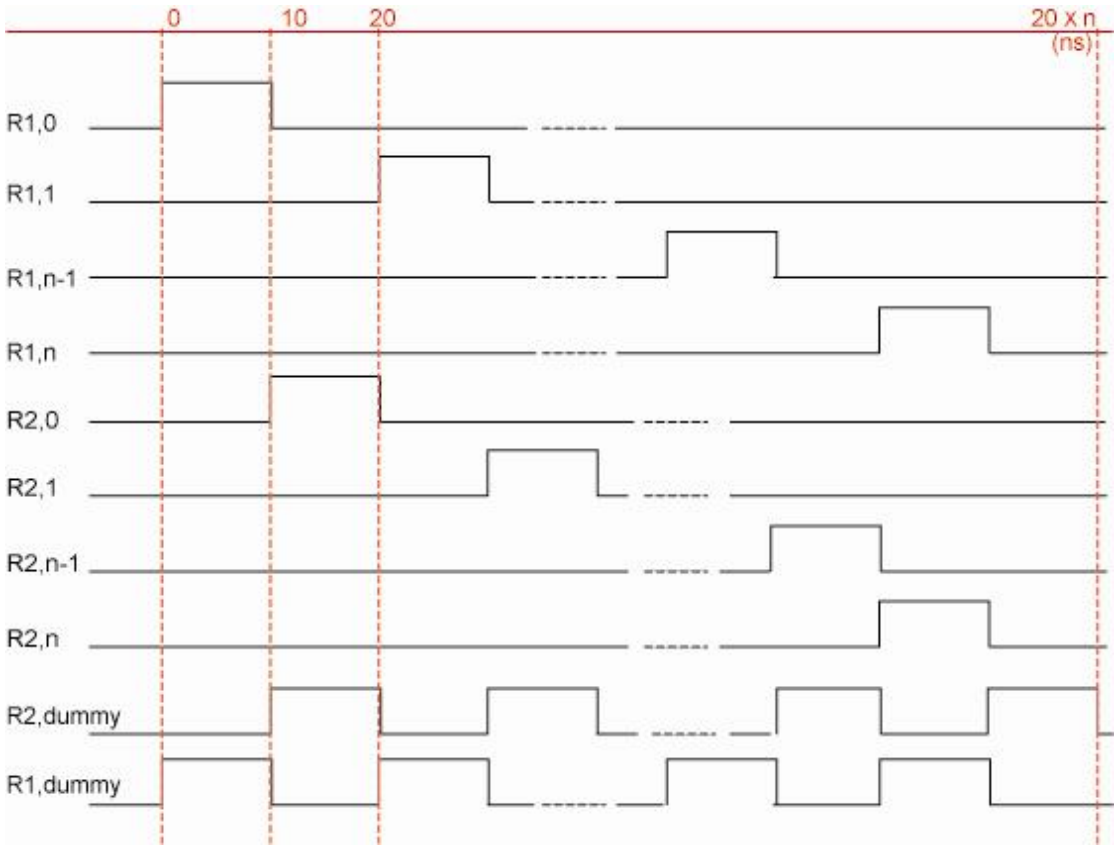


Figure 3-34 Time diagram for the control signal of the readout block

3.7.1 Sense amplifier

The sense amplifiers used in the pixel memory readout play a major role in the functionality of the system. They have functions of amplification, delay reduction, power reduction and signal restoration. More detailed information for the sense amplifiers can be found in [36].

In our design, we use a so called clamped bit-line current mode sense amplifier[37]. The benefit with this architecture is that instead of charging and discharging the bit-lines, the load of this sense amplifier is the drain capacitance of the transistor, which is much smaller. So the speed of this sense amplifier is very fast and also the power consumption is low. Besides, the array size of the pixel can be easily scaled without redesign of the sense amplifier. This is because normal voltage sense-amplifiers do a readout operation by compare the speed of charging or discharging the bit-line capacitance which is also changing with the scaling of the array size. The schematic of this sense amplifier is shown in Figure 3-35.

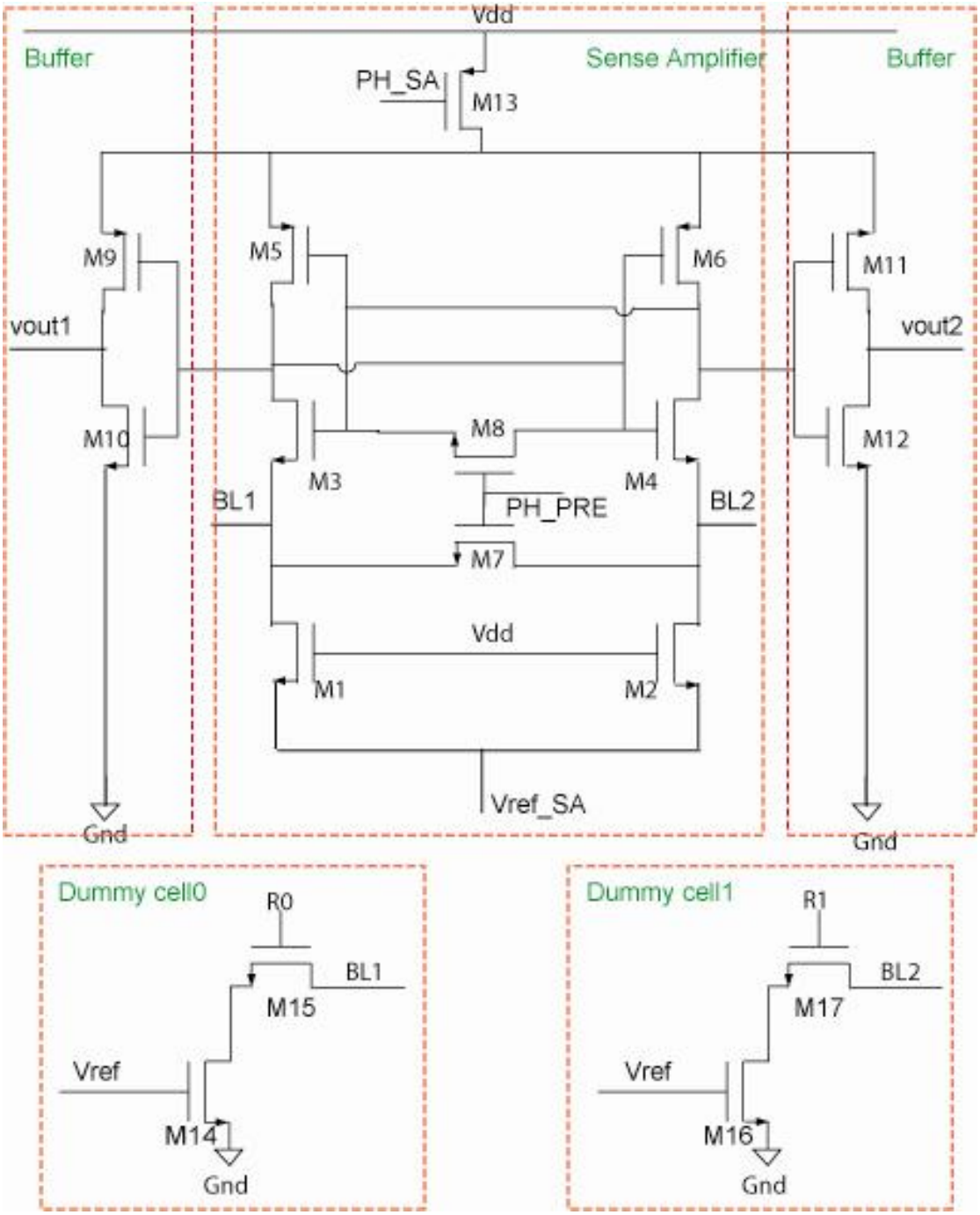


Figure 3-35 Sense amplifier and dummy cell used in each column

The timing diagram for the control signals is shown in Figure 3-36.

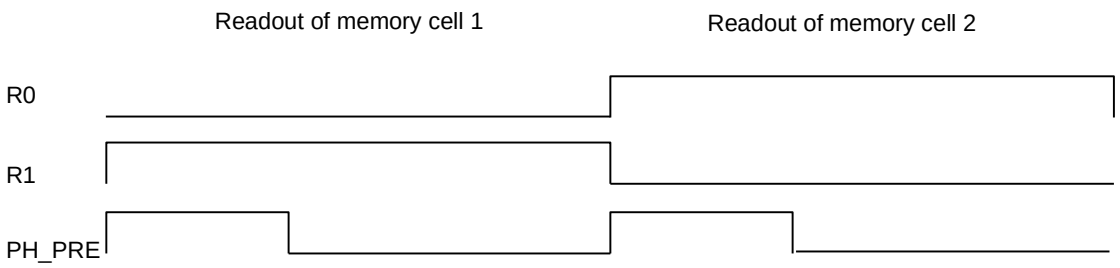


Figure 3-36 Timing diagram of the control signals in the sense amplifier

The working principle of the sense amplifier is explained in more detail in [37], but we will briefly explain its operation in the following. When PH_SA is set low, the sense amplifier starts to work. At the beginning of the readout phase, PH_PRE is pulled high. The sense amplifier is reset and the two bit-line voltages and the output of the cross coupled inverter (M_3, M_4, M_5 and M_6) are set at the same voltage level. Transistors M_1 and M_2 work in the linear region. The bit-line voltages are set by the current flowing into the two branches. When the row select signal is set high, because of the current difference that flows into the bit-line through the selected memory and dummy cells, the current flowing through M_3 and M_4 is different. This will charge the gate drain capacitance of these two transistors at a different speed. After the signal PH_PRE goes down, the positive feedback of the four cross coupled transistors will pull one side to V_{dd} and the other to ground. The two buffers connected to the outputs of the sense amplifier reduce the influence of the output load influence on the sense amplifier.

The bit-line of the memory cell shown in Figure 3-23(a) cannot be connected to the sense amplifier directly. The reason for this is that when the gray code input to the memory cell is '1', because the gates of transistors M_1 and M_2 in Figure 3-35 are connected to V_{dd}, there is a large current flowing through these two transistors. To avoid this problem, a switch is placed between this type of memory cell and the sense amplifier as shown later in section 3.7.2.

Figure 3-37 shows a simulation of the sense amplifier (10ns readout time for one bit). The row select signal is the same as shown in Figure 3-36. In this simulation the data stored in the memory that needs to be read is '1'. When R1 is high, both the dummy cell 1 and one of the memory cells connected to BL1 are selected, and V_{out1} goes high in 3ns. When R2 is high, dummy cell 0 and one of the memory cells connected to BL2 are selected, and V_{out2} goes high.

Figure 3-38 shows the current consumption for the sense amplifier when reading out two bits. The current is mainly consumed during the equalization (when PH_PRE is high) and sensing phases. The average current in this case is 4uA. Because the output is fully

differential, we only need one of the outputs to judge whether a '1' or a '0' is stored in the memory.

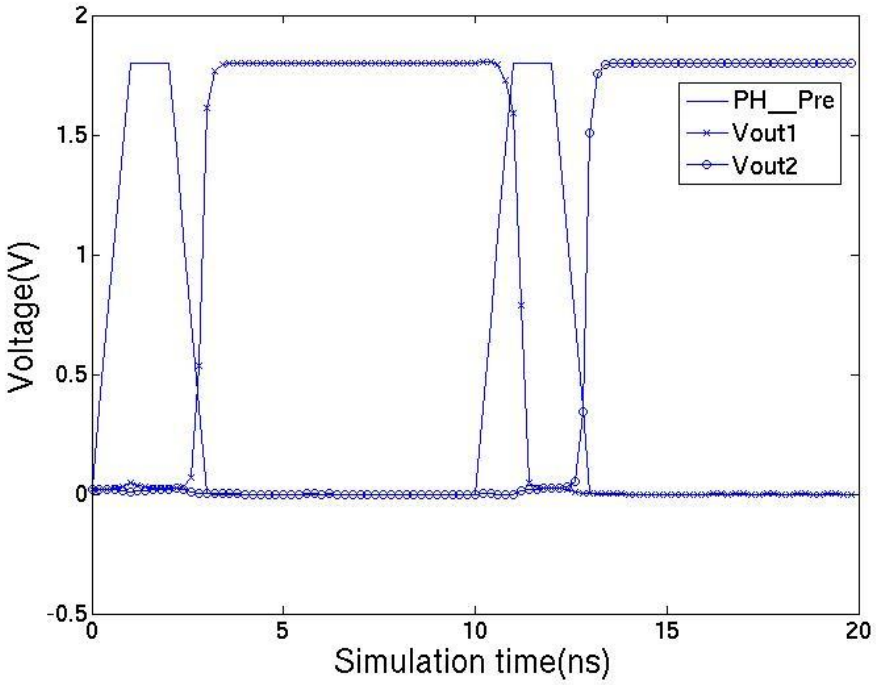


Figure 3-37 Simulation result of the current sense amplifier

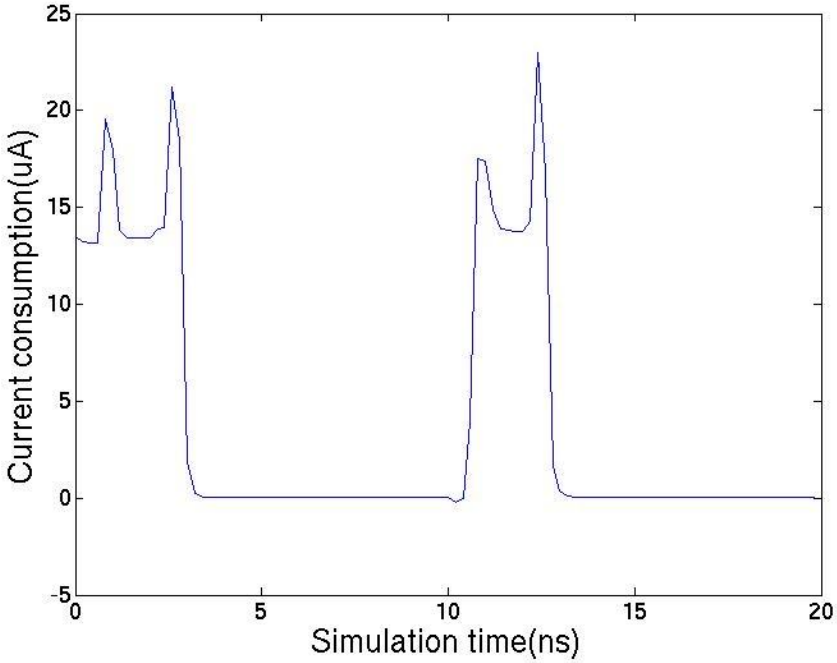


Figure 3-38 Current consumption for the sense amplifier

Transistors M_3, M_4, M_5 and M_6 in Figure 3-35 form a cross coupled inverter. The output of the sense amplifier depends on the charging or discharging speed of the drain capacitances of M_3,M_5 and M_4,M_6. In the layout of this sense amplifier, the circuit should be symmetrical, because these two outputs are coupled to other control signals such as PH_PRE via the metal interconnects. If there is a mismatch between these coupling capacitances, the output may not depend on the charging of discharging current difference, but on the coupling of the signal lines. We decide to fit 3 sense amplifiers into each pixel. This means that the width of the amplifier should be limited to 6um.

Table 3-7 summarizes the signals that are used for the sense amplifier.

Table 3-7 Signals used for the sense amplifier

Name	Type	IO	Description
PH_SA	Digital	IN	Power on signal for the sense amplifier
PH_PRE	Digital	IN	Pre-setting signal for the sense amplifier
R0	Digital	IN	Dummy cell0 select signal
R1	Digital	IN	Dummy cell1 select signal
Vref_SA	Analog	IN	Reference signal for the sense amplifier, setting the drain-source voltage of the memory cell
Vref	Analog	IN	Reference voltage of the dummy cell
Vout<1:2>	Digital	OUT	Output voltage of the sense amplifier

3.7.2 Connection of memory related blocks

Figure 3-39 shows the connections between the gray-code counter, in-pixel memory array, and the sense amplifiers.

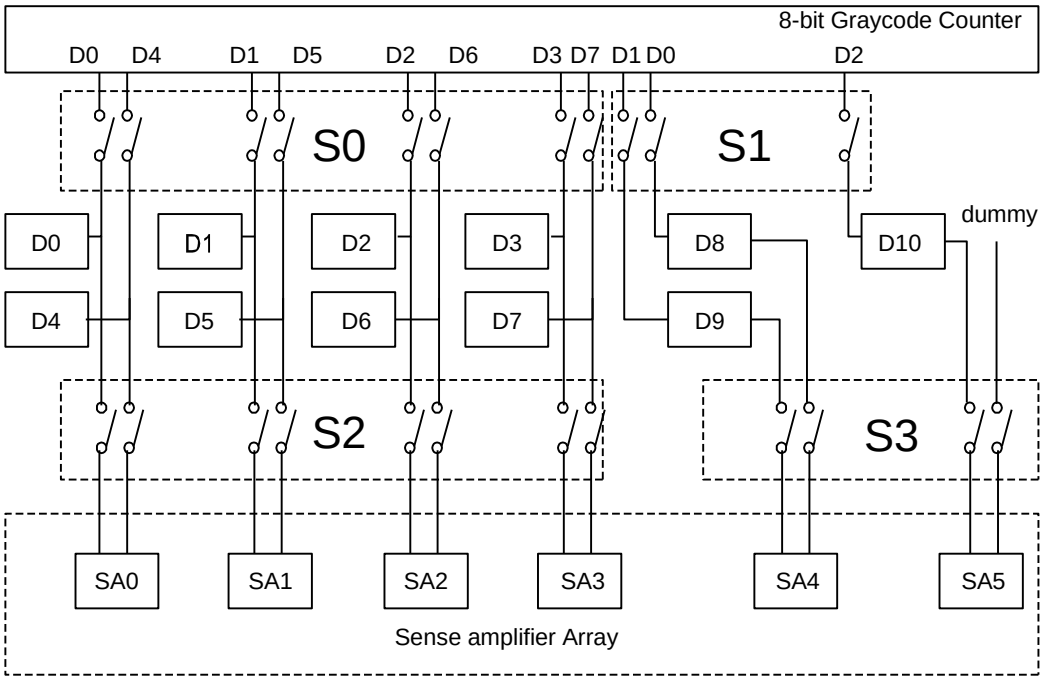


Figure 3-39 Connections for the gray code counter, memory array and sense amplifiers for one pixel

The memory cells D0-D7 use the cell shown in Figure 3-23(a), and the memory cells D8-D9 use the cell shown in Figure 3-23(b). All the memory cells are located inside the pixel circuit. The switch arrays S0, S1 connect the gray code counter to the in-pixel memory. They are switched on when the gray code needs to be input to the memory cell during the coarse and fine conversions. S2 is the switch array between the in-pixel memory and sense amplifiers. It is switched on when the data that is stored need to be readout. S3 is a dummy switch array connecting the coarse bit memory to the sense amplifiers SA4 and SA5.

3.7.3 Shift register

In order to minimize the number of pads that are used to read the data from the sense amplifier array, a shift register is connected to the output signal of the sense amplifier [37].

In our case we use a Parallel-in, Serial-out (PISO) type of shift register, which stores the parallel input data and then shifts the data sequentially. The architecture of a 4 bit shift register is shown in Figure 3-40. When the control signal Write/Shift goes high, the register starts to shift the data out. The shifting occurs at the falling edge of the clock, so the control signal should occur at the rising edge of the clock.

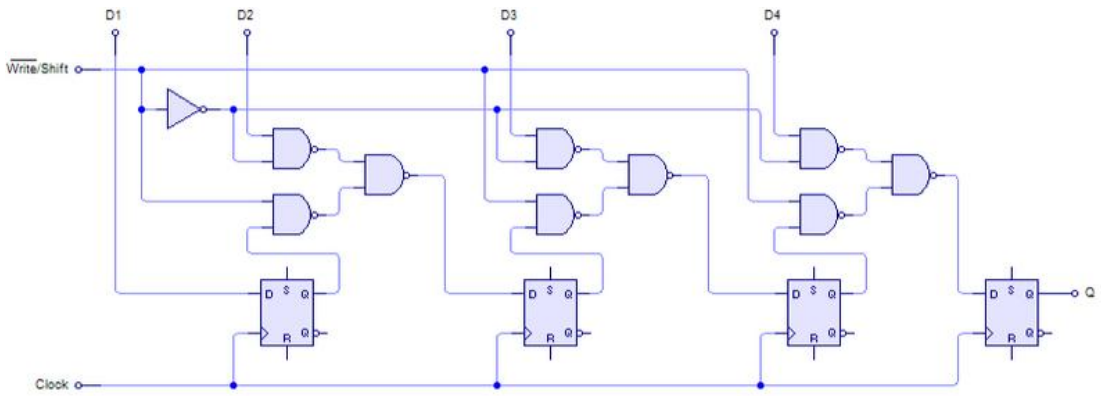


Figure 3-40 Architecture of a 4-bit PISO shift register[38]

3.8 Chip overview

We have designed a test chip to test the validity of the conversion algorithm, as well as the operation of the comparator, the memory and the readout related circuits.

The pixel array consists of 64 rows of 64 pixels. As we have shown earlier, there are 11 in-pixel memories placed inside the pixel. As each sense amplifier reads two bits, each pixel column needs six sense amplifiers to read the pixel memories. The total number of sense amplifiers is thus 6 x 64=384. We use 6 PISO shift registers to send the output of the sense amplifiers off-chip.

The clock frequency selected for the shift registers is 100MHz. Thus, reading out one row of data needs 640ns. We need two sense amplifier cycles to read different memory cells as has been explained in section 3.7.1. So the total data readout time for all 64 rows is 2 x 640ns x 64 = 81.92 µs. Most power consumption happens during the phase when PH_PRE is high for equalization and the sensing phase explained in section 3.7.1. During the readout of the memory, the power consumption of the sense amplifier is much lower. Simulation results show that the average current flowing in each sense amplifier is 106 nA.

3.8.1 Test circuits

We also need to test the different blocks of the circuit separately (e.g. offset variation of comparators, operation of the memory related circuits). To do this we add additional circuits shown in Figure 3-41 for test. It is just a controllable IO gate.

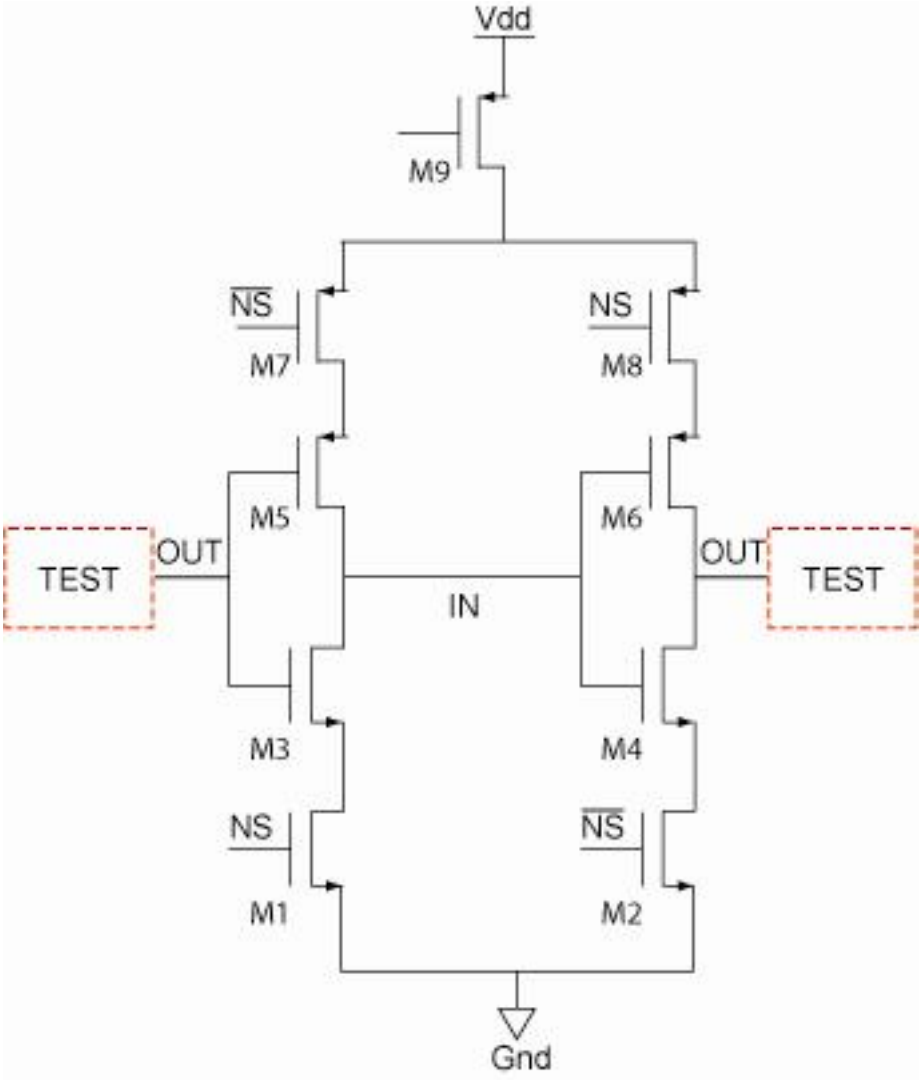


Figure 3-41 Circuit schematic for test that is connected to pixels

The signals used in the circuit are listed in Table 3-8.

Table 3-8 Description for the signals of the additional test circuit

NAME	TYPE	IO	DESCRIPTION
TS	DIGITAL	INPUT	Row select signal to choose which pixel for test
NS	DIGITAL	INPUT	Choose whether the PORT “IN” and “OUT” is input or output, when NS is high, port “IN” is input and port ”OUT” is output.
IN	DIGITAL	INOUT	Connected to pad for test
OUT	DIGITAL	INOUT	Connected to pixel for test

We use one such circuit per pixel in the leftmost column. The signal “OUT” in Figure 3-41 is connected to the node “TEST” shown in Figure 3-42. The only difference of Figure 3-42 and

Figure 3-12 is that the node 'TEST' shown in Figure 3-42 is connecting to a test circuit. To test the operation of the comparator the signals "GUARD" and "NS" are set low. We can now control the input signal to latch the gray-code into the in-pixel memory and read it out to check whether the memory readout circuits work correctly.

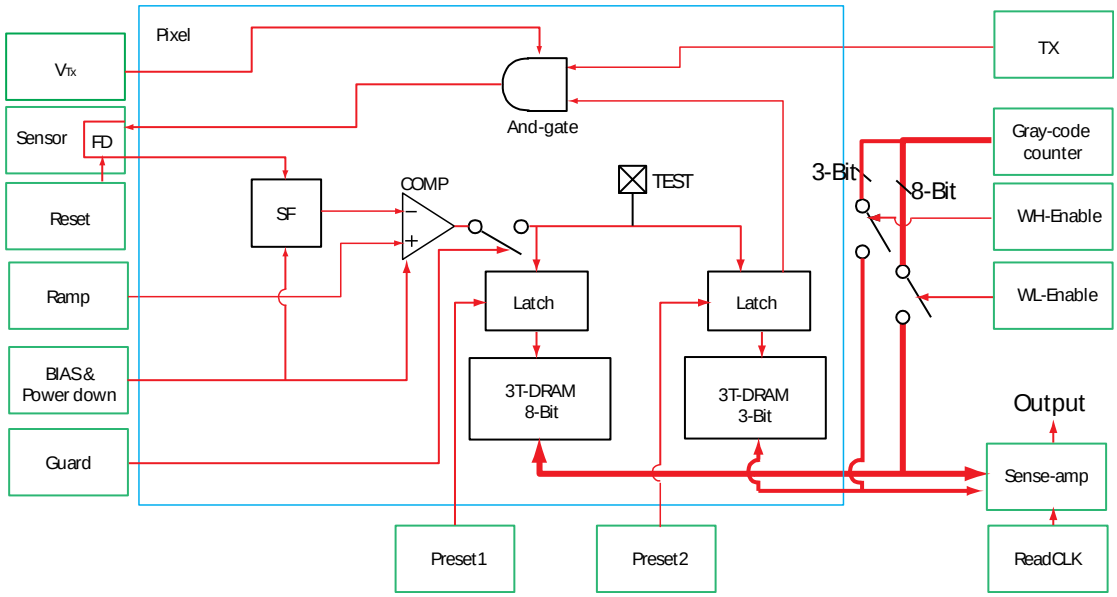


Figure 3-42 block diagram of the pixel circuit

We can also set the "GUARD" signal high, and change the PORT connecting to "TEST" into an output port. The negative input of the comparator is fixed at a certain value (activating the reset switch). The ramp is swept from V_{low} to V_{high}. We can then check at the port "IN" when the output of the comparator changes value. If we do this for all the pixels then we can measure the offset variation of the comparators. For the whole column pixel testing, we need 64 frames. In each frame time, the signal "TS" chooses which pixel is going to be tested.

The final test chip layout is shown in Figure 3-43. The output buffers and pads are not shown. The digital part which generates the digital control signals is not included either.

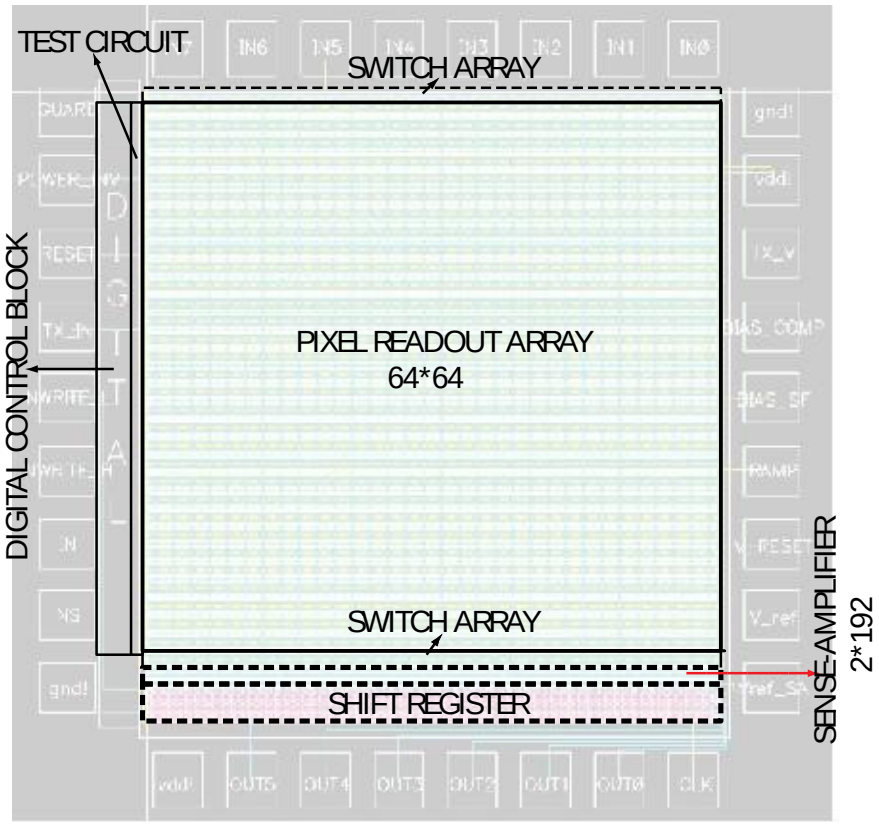


Figure 3-43 Final chip layout

Table 3-9 lists the chip specifications.

Table 3-9 Readout Circuit Overview

Technology	TSMC 0.18um
Chip size (excluding the pads and digital block)	1.2mm*1.3mm
Supply voltage	1.8V
Pixel type	Pinned Photodiode
Pixel Pitch	18um
Pixel Array	64*64
Clock Frequency	100MHz
DR(sensor)	13.88 bits
SNR(single slope ADC)	8 bits
Frame rate	1000 fps
FOM_{SNR}(single slope ADC)	0.208pJ/conversion
FOM_{DR}(DR)	0.004pJ/conversion
INL(LSB)	0.23(worst case)
DNL(LSB)	0.23(worst case)

In order to test the chip during a frame we can input a different reset voltage values while the reset switch is switched on, to control when we want to latch the coarse bits, and then read them out to check whether the output is correct.

The pins of the chip and their descriptions are listed in Table 3-10.

Table 3-10 IO Pins of the designed readout circuit

Pin name	Type	IO	Description
IN7-IN0	Digital	IN	Gray code input
VDDA & GNDA	Analog Power supply	IN	Power supply and ground for the pixel readout circuit and the sense amplifier
TX_V	Analog	IN	Input voltage to tune the transfer gate voltage
BIAS_COMP	Analog	IN	Input voltage for biasing of the comparator
BIAS_SF	Analog	IN	Input voltage for biasing of the in-pixel source follower
RAMP	Analog	IN	Ramp signal input to the comparator for comparison
V_RESET	Analog	IN	RESET voltage
V_ref	Analog	IN	Reference voltage for the sense amplifier
Vref_SA	Analog	IN	Reference voltage for the sense amplifier
CLK	Digital	IN	Shift register and digital block input CLK
OUT5-OUT0	Digital	OUT	Shift register output data
VDDD & GNDD	Digital	IN	Power supply for shift register and digital block
NS	Digital	IN	Choose whether the test port is input or output
IN	Digital	INOUT	INOUT port for test
N_WRITEH & N_WRITEL	Digital	IN	Control for writing and shifting of the data for coarse and fine bits of the shift registers
TX_IN	Digital	IN	Globally distributed transfer gate control voltage
RESET, POWER_INV & GUARD	Digital	IN	Pixel digital control signals

The control signals that need to be generated digitally are listed in Table 3-11.

Table 3-11 Digital generated control signals

Name	IO	Description
CLK	IN	Clock input of the digital block
Reset	IN	Starting of one frame
S0-S3	OUT	Control signal for switch array that connecting the gray code input, pixel array and sense amplifier array
PRESET_L PRESET_H	OUT	Control signal for the latch placed inside the pixel
RS0<0:63> RS1<0:63> RS2<0:63> RS3<0:63>	OUT	Row select signal for the readout of the in-pixel memory
R0 R1	OUT	Control signals for the dummy cell of the sense amplifier array
PH_SAL PH_SAH	OUT	Power Control signal of the sense amplifier array for coarse bits and fine bits
PH_PRE	OUT	Control signal for equalization of the sense amplifier array
TS<0:63>	OUT	Control signal to choose which pixel is being tested

3.9 Conclusions

A pixel level ADC for dynamic range enhancement with slightly additional power consumption and its related circuits has be designed. Because of the digital CDS, the digital readout of the in-pixel memory cannot be done during the integration (in our case, there is 200µs left for the fine conversion and digital readout of the memory cell). This will be a limiting factor on the array size of the pixel for high speed applications, because large array size needs longer digital readout time and will limit the longest integration time. But in a 3D structure, and if we can place the digital processing blocks for each pixel or a group of pixels, then this problem will not be a limitation anymore because all the data can be (simply) processed (e.g.: digital subtraction of two converted numbers) locally and we can then readout the final data during the integration of the charges.

The coarse conversion time equals the longest integration time, so for low frame rates the 3T DRAM that is used in our design will not work properly anymore, because the leakage of the memory will ruin the data that has been stored. In that case it would be better to use a 6T-SRAM structure or adding refresh logic in the whole circuit.

4 Conclusions and future work

In this work we describe a new method to increase the dynamic range of an image sensor by using the special operation of the pinned photodiode. By generating a local, pixel-level charge transfer control signal, smart pixels can be made which control the integration time locally depending on the light intensity. Only one comparison is needed after each charge transfer, so a lot of power can be saved by powering down the comparator when it is not used. Digital control signals (GUARD, POWER_INV in our design) are used to guarantee that the correct data is latched in the memory when the comparator is powered down.

The floating diffusion (FD) of the sensor cannot be connected to the input of the comparator directly due to its low capacitance, which could lead to charge redistribution between the FD and the input capacitance of the comparator. For that reason, a source follower is needed as buffer between the FD and the input of the comparator.

Digital CDS is chosen in the design for simplicity of the in-pixel circuitry. The longest integration time will be limited by the frame rate and the digital readout time of the code that has been latched into the memory. This will also have constraints on the sizing and choice of the in-pixel memory circuit due to the leakage of the memory cell. The shortest integration time will be limited by the time needed to convert the reset voltage. The timing of the digital control signals should be well defined to guarantee that the correct transfer gate control signal is generated locally based on the incoming light intensity.

A two-bit-line 3T DRAM structure is chosen for the coarse bit memory in order to reduce the time that the data need to be stored in the memory. For low frame rate applications it might be better to change the coarse bit memory structure into SRAM or investigate the possibility of adding refresh logic circuit in the chip as have been explained in section 3.9. A clamped bit line sense amplifier structure is chosen for the readout of the memory because of its high speed and low power and also no need to redesign if the array size is scaled. In order to reduce the number of output pads, six shift registers are used to readout the data from the sense amplifier array.

In order to test the working of the different blocks in the chip, additional circuitry has been added. The operation of the digital control signals for test has also been explained in detail. Because of the slightly complicated digital control, the digital signals should be generated from a digital macro on chip. To conclude the design of the test chip this digital control block needs to be designed. Also the possibility of using other types of memory cells needs to be investigated. For example, 1T DRAM for fine bits and SRAM for coarse bits, as this part consumes around half of the area of the total pixel readout circuit. The noise limit of this design needs to be investigated on silicon, as this will have a strong influence on the low limit of the DR. For an actual application, the timing of the digital control signals needs to be

recalculated based on frame rate, integration times, etc…(i.e. the time needed to convert the reset voltage, the longest integration time, digital readout time, etc.), also the single slope ADC and the memory part should be redesigned (LSB value, number of bits) based on the noise level measurement of the sensor with the operation shown in this thesis.

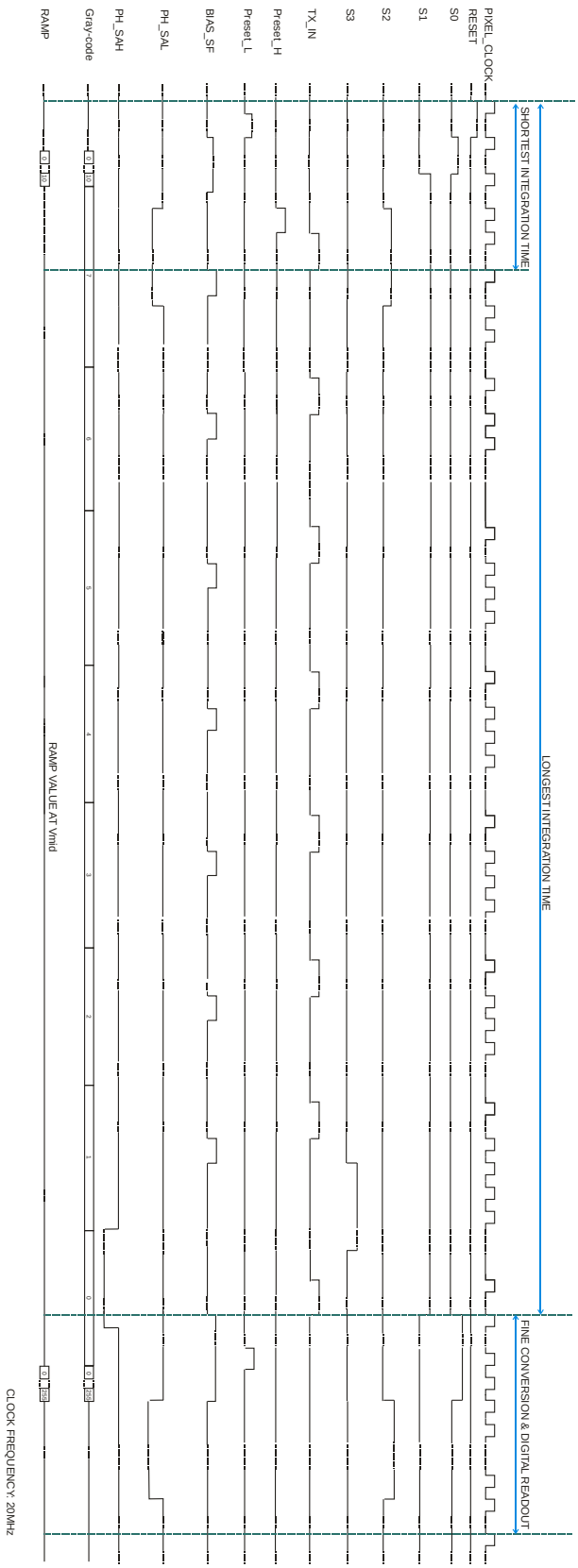
References

1. El Gamal, A. and H. Eltoukhy, *CMOS image sensors.* IEEE Circuits and Devices Magazine, 2005. **21**(3): p. 6-20.
2. Snoeij, M., *Analog Signal Processing for CMOS Image Sensors.* PHD thesis, Delft University of Technology, 2007.
3. Theuwissen, A., *CMOS image sensors: State-of-the-art.* Solid State Electronics, 2008. **52**(9): p. 1401-1406.
4. Shouho, M., et al. *A Low-Voltage Pulse-Width-Modulation Image Sensor*. in *IEEE Workshop on Charge-Coupled Devices & Advanced Image Sensors*. 2005.
5. Lee, J. and G. Han, *A/D Converter using Iterative Divide-by-Two Reference for CMOS Image Sensor*, in *SoC Design Conference, 2008. ISOCC '08. International* 2008. p. 35-36.
6. Sasaki, M., et al., *A wide-dynamic-range CMOS image sensor based on multiple short exposure-time readout with multiple-resolution column-parallel ADC.* IEEE Sensors Journal, 2007. **7**(1): p. 151-158.
7. Nitta, Y., et al., *High-speed digital double sampling with analog CDS on column parallel ADC architecture for low-noise active pixel sensor.* IEEE ISSCC Dig. Tech. Papers, 2006: p. 500-501.
8. Lim, S., et al., *A High-Speed CMOS Image Sensor With Column-Parallel Two-Step Single-Slope ADCs.* IEEE Transactions on Electron Devices, 2009. **56**(3): p. 393-398.
9. Nelson, F., M. Alam, and S. Ay, *A Single-Slope Look-Ahead Ramp (SSLAR) ADC for Column Parallel CMOS Image Sensors*, in *Microelectronics and Electron Devices, IEEE Workshop on*. 2009. p. 1-4.
10. Krymski, A., et al., *A high speed, 500 frames/s, 1024x1024 CMOS active pixel sensor.* Proc. IEEE Symp. VLSI Circuits, 1999: p. 137-138.
11. Krymski, A., et al., *A high-speed, 240-frames/s, 4. 1-Mpixel CMOS sensor.* IEEE Transactions on Electron Devices, 2003. **50**(1): p. 130-135.
12. Matsuo, S., et al., *8.9-Megapixel Video Image Sensor With 14-b Column-Parallel SA-ADC.* IEEE Transactions on Electron Devices, 2009. **56**: p. 2380-2389.
13. Zhou, Z., B. Pain, and E. Fossum, *CMOS active pixel sensor with on-chip successive approximation analog-to-digital converter.* IEEE Transactions on Electron Devices, 1997. **44**(10): p. 1759-1763.
14. Ginetti, B., P. Jespers, and A. Vandemeulebroecke, *A CMOS 13-b cyclic RSD A/D converter.* IEEE journal of solid-state circuits, 1992. **27**(7): p. 957-965.
15. Mase, M., et al., *A wide dynamic range CMOS image sensor with multiple exposure-time signal outputs and 12-bit column-parallel cyclic A/D converters.* IEEE journal of solid-state circuits, 2005. **40**(12): p. 2787-2795.

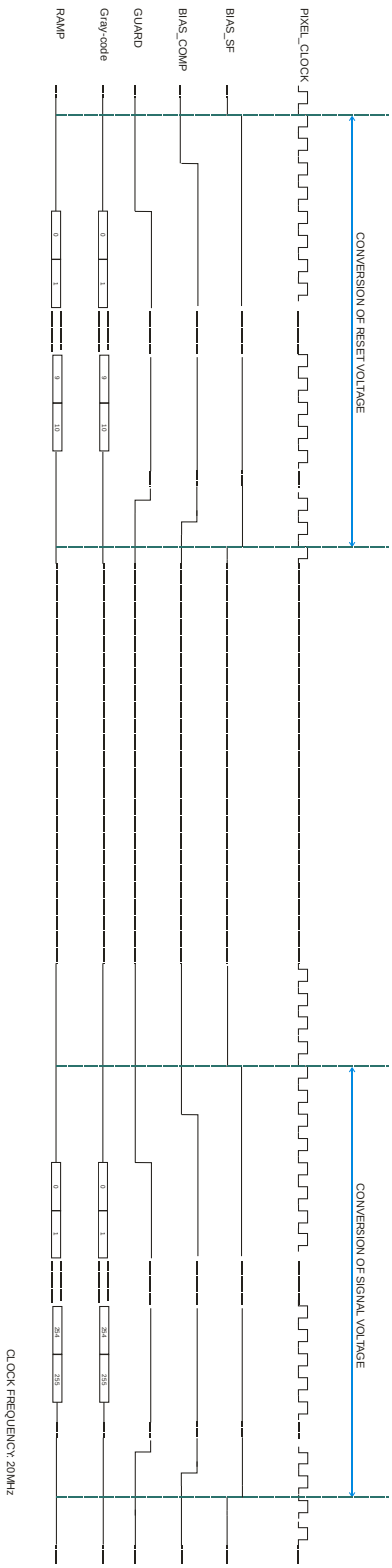
16. Furuta, M., et al., *A High-Speed, High-Sensitivity Digital CMOS Image Sensor With a Global Shutter and 12-bit Column-Parallel Cyclic A/D Converters.* IEEE journal of solid-state circuits, 2007. **42**(4): p. 766-774.
17. Park, J., et al., *A High-Speed Low-Noise CMOS Image Sensor With 13-b Column-Parallel Single-Ended Cyclic ADCs.* IEEE Transactions on Electron Devices, 2009. **56**: p. 2414-2422.
18. Findlater, K., et al., *SXGA pinned photodiode CMOS image sensor in 0.35 µm technology.* IEEE ISSCC Dig. Tech. Papers, 2003. **46**: p. 218-219.
19. Mansoorian, B., et al., *A 250 mW, 60 frames/s 1280*720 pixel 9 b CMOS digital imagesensor.* IEEE ISSCC, Digest of technical papers, 1999: p. 312-313.
20. Takayanagi, I., et al., *A 1.25 inch 8.3 M pixel digital output CMOS APS for UDTV application.* Proc. ISSCC 2003, 2003. **1**: p. 216-217.
21. Yang, D., et al., *A 640*512 CMOS image sensor with ultrawide dynamic range floating-point pixel-level ADC.* IEEE journal of solid-state circuits, 1999. **34**(12): p. 1821-1834.
22. Ignjatovic, Z. and M. Bocko, *A 0.88 nW/pixel, 99.6 dB Linear-Dynamic-Range Fully-Digital Image Sensor Employing a Pixel-Level Sigma-Delta ADC.* Symposium on VLSI Circuits Digest of Technical Papers, 2006: p. 23-24.
23. Fowler, B. and A. El Gamal, *CMOS image sensor with pixel level A/D conversion*. 1995, Google Patents, Patent number: 5,461,425.
24. Yang, D., B. Fowler, and A. Gamal, *A 128 x 128 Pixel CMOS Area Image Sensor with Multiplexed Pixel Level A/D Conversion.* in CICC96, 1996: p. 303-306.
25. Maricic, D., Z. Ignjatovic, and M. Bocko, *An Oversampling Digital Pixel Sensor with a Charge Transfer DAC Employing Parasitic Capacitances.* 2009 52nd IEEE International Midwest Symposium on Circuits and Systems, 2009: p. pp.415-418.
26. Mahmoodi, A. and D. Joseph, *Pixel-level delta-sigma ADC with optimized area and power for vertically-integrated image sensors.* in Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on, 2008: p. 41-44.
27. Yang, D., B. Fowler, and A. El Gamal, *A Nyquist-rate pixel-level ADC for CMOS image sensors.* IEEE journal of solid-state circuits, 1999. **34**(3): p. 348-356.
28. Kleinfelder, S., et al., *A 10000 frames/s CMOS digital pixel sensor.* IEEE journal of solid-state circuits, 2001. **36**(12): p. 2049-2059.
29. Rhee, J. and Y. Joo, *Wide dynamic range CMOS image sensor with pixel level ADC.* Electronics Letters, 2003. **39**(4): p. 360-361.
30. Peizerat, A., et al., *Pixel-level A/D conversion: comparison of two charge packets counting techniques*, in *IISW*. 2007. p. 200-203.

31. Wooley, S., *Per-pixel floating-point ADCs with electronic shutters for a high dynamic range, high frame rate infrared focal plane array.* IEEE Custom Intergrated Circuits Conference (CICC), 2006: p. 647-650.
32. Wu, X. and A. Bermak, *A Low Power Digital Pixel Sensor with a dynamically biased ADC*, in *International SoC Design Conference*. 2008. p. p.105 -108.
33. Wang, X., *Noise in Sub-Micron CMOS Image Sensors.* PHD thesis, Delft University of Technology, 2008.
34. Kim, D., et al., *A Dual-Capture Wide Dynamic Range CMOS Image Sensor Using Floating-Diffusion Capacitor.* IEEE Transactions on Electron Devices, 2008. **55**(10): p. pp. 2590 - 2594
35. Vittoz, E. *Micropower techniques*. 1994: Prentice-Hall, Inc.
36. Rabaey, J., A. Chandrakasan, and B. Nikolic, *Digital integrated circuits*. 2002: Prentice Hall Englewood Cliffs, New Jersey.
37. Blalock, T. and R. Jaeger, *A high-speed clamped bit-line current-mode sense amplifier.* IEEE journal of solid-state circuits, 1991. **26**(4): p. 542-548.
38. http://en.wikipedia.org/wiki/Shift_register.

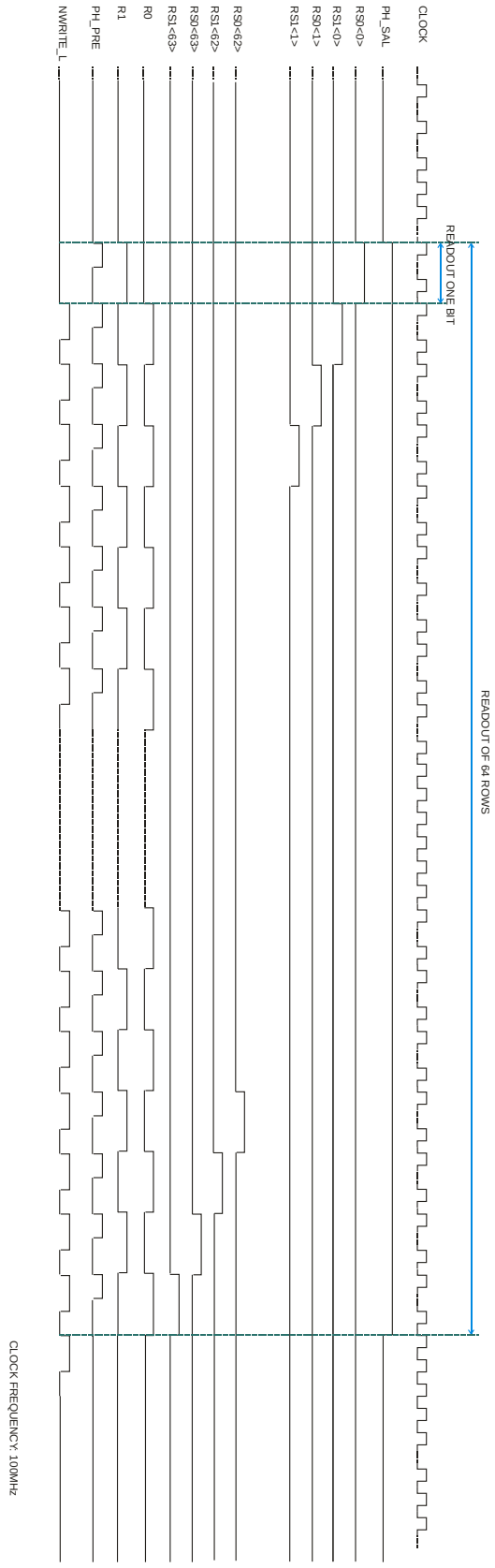
Appendix: Timing for Main Digital Control Signals



Appendix: Timing for Digital Signals Related to Single Slope Conversion



Appendix: Timing for Digital Signals Related to Digital Readout Circuits(Fine Bits)



Appendix: Description of digital signals

```
================================================================================
File name:        t_digital.vhd

Authors:          CHENG MA

Locations:        Leuven, Belgium

Description:      Signal description for digital block. Also contains the signal for input. In this
file, the signal 'RESET' is used as a reference signal for the starting of one frame. For testing
of the chip, the 'RESET' is connecting to a DC voltage. The total test include need 66 frames,
frame 1 is used to test the working of the chip, frame 2 is used to test the working of the
memory related circuits in case comparator does not work. Frame 3-66 are used for testing
the working of the comparator and its mismatch between different pixels.

================================================================================

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity Digital is
   port   (CLOCK_IN                : in  std_logic;
          --CLOCK for digital readout
            CLOCK_IN_PIXEL         : in  std_logic;
          --Clock for the in-pixel control signals
           RESET_REF               : in  std_logic;
          --System resetting
            RESET_SYSTEM           : in  std_logic;

            TX_IN                  : out std_logic;
            PRESET_L               : out std_logic;
            PRESET_H               : out std_logic;

            NWRITE_L               : out std_logic;
            NWRITE_H               : out std_logic;

            GUARD                  : out std_logic;
            NS                     : out std_logic;
```

```
        --Switch array control signals
         S0                  : out std_logic;
         S1                  : out std_logic;
         S2                  : out std_logic;
         S3                  : out std_logic;


         RS0                 : out std_logic_vector(63 downto 0);
         RS1                 : out std_logic_vector(63 downto 0);
         RS2                 : out std_logic_vector(63 downto 0);
         RS3                 : out std_logic_vector(63 downto 0);

         R0                  : out std_logic;
         R1                  : out std_logic;

         PH_SAL              : out std_logic;
         PH_SAH              : out std_logic;
         PH_PRE              : out std_logic;
        TS                   : out std_logic_vector(63 downto 0)
        );

end entity;

architecture structure of Digital is

         signal CLOCK_PIXEL          : std_logic;

        --Counter for one frame, generating the control signals
         signal COUNTER_FRAME_DRV    : std_logic_vector(14 downto 0);
         signal COUNTER_FRAME_REG    : std_logic_vector(14 downto 0);
         signal COUNTER_FRAME_UGN    : unsigned(14 downto 0);

          --Registers for switch control signals, for generating row select signals
         signal S2_REG               : std_logic;
         signal S3_REG               : std_logic;
```

```
        signal S2_DRV                   : std_logic;
        signal S3_DRV                   : std_logic;

        --Counter helping generating the row select signals
        signal Counter_local_Low_DRV    : std_logic_vector(14 downto 0);
        signal Counter_local_Low_REG    : std_logic_vector(14 downto 0);
        signal Counter_local_Low_UGN    : unsigned(14 downto 0);

        signal Counter_local_High_DRV   : std_logic_vector(14 downto 0);
        signal Counter_local_High_REG   : std_logic_vector(14 downto 0);
        signal Counter_local_High_UGN   : unsigned(14 downto 0);

       --Counter counting the number of frames
        signal COUNTER_NFRAME_DRV       : std_logic_vector(14 downto 0);
        signal COUNTER_NFRAME_REG       : std_logic_vector(14 downto 0);
       signal COUNTER_NFRAME_UGN        : unsigned(14 downto 0):= (others=>'0');



  --=====================================================

begin

    --generating row select signals for fine bits
    Sig_Gen_Low:  for i in 0 to 63 generate
            begin
             RS0(i) <= '1' when (COUNTER_LOCAL_Low_UGN > 128*i and
COUNTER_LOCAL_Low_UGN <=128 * i+64) else '0';
             RS1(i) <= '1' when (COUNTER_LOCAL_Low_UGN > 128*i+64 and
COUNTER_LOCAL_Low_UGN <=128 * (i+1)) else '0';
            end generate Sig_Gen_Low;

    --generating row select signals for high bits
    Sig_Gen_High: for j in 0 to 63 generate
            begin
             RS2(j) <= '1'
                when (COUNTER_LOCAL_High_UGN > 128*j
                and     COUNTER_LOCAL_High_UGN <=128 * j+64) else '0';
```

```
            RS3(j) <= '1'
                when (COUNTER_LOCAL_High_UGN > 128*j+64
                and       COUNTER_LOCAL_High_UGN <=128 * (j+1)) else '0';
          end generate Sig_Gen_High;

--generating TS signal
Sig_Gen_TS:  for i in 0 to 63 generate
          begin
           TS(i) <= '1' when (COUNTER_NFRAME_UGN = i+3) else '0';
          end generate Sig_Gen_TS;


--generating select signals for dummy memory inside the sense amplifiers.
R1 <= '1'
     When (
     ((COUNTER_LOCAL_High_UGN rem 128) <= 64
     and(COUNTER_LOCAL_High_UGN rem 128) > 0)
     or ((COUNTER_LOCAL_Low_UGN rem 128) <= 64
     and(COUNTER_LOCAL_Low_UGN rem 128) > 0)) else '0';
R0 <= '1'
     when (
      (((COUNTER_LOCAL_High_UGN rem 128) <= 127
      and(COUNTER_LOCAL_High_UGN rem 128) > 64 )
     or ((COUNTER_LOCAL_High_UGN rem 128) = 0
     and (s3_REG = '1'))
      )
      or
      (((COUNTER_LOCAL_Low_UGN rem 128) <= 127
     and(COUNTER_LOCAL_Low_UGN rem 128) > 64)
     or ((COUNTER_LOCAL_Low_UGN rem 128) = 0
     and (s2_REG = '1'))
     )
      )  else '0';

      COUNTER_FRAME_UGN <= unsigned(COUNTER_FRAME_REG);
      COUNTER_LOCAL_Low_UGN <= unsigned(COUNTER_LOCAL_Low_REG);
      COUNTER_LOCAL_High_UGN <= unsigned(COUNTER_LOCAL_High_REG);
```

```
Sequential: process(CLOCK_IN_PIXEL, RESET_REF)
begin
   if (clock_in_pixel'event and clock_in_pixel = '1') then
     if(RESET_REF = '1') then
       COUNTER_FRAME_REG <= (OTHERS=>'0');
     else
       COUNTER_FRAME_REG <= COUNTER_FRAME_DRV;
       S2_REG          <= S2_DRV;
       S3_REG          <= S3_DRV;
     end if;
   end if;
end process Sequential;

Sequential_Row_select: process(CLOCK_IN)
begin
  if (clock_in'event and clock_in = '1') then
       if(S2_REG = '0') then
         COUNTER_LOCAL_Low_REG <= (OTHERS=>'0');
       else
         COUNTER_LOCAL_Low_REG <= COUNTER_LOCAL_Low_DRV;
       end if;

       if(S3_REG = '0') then
         COUNTER_LOCAL_High_REG <= (OTHERS=>'0');
       else
         COUNTER_LOCAL_High_REG <= COUNTER_LOCAL_High_DRV;
       end if;
     end if;
end process Sequential_Row_select;

Combinational: process(COUNTER_FRAME_UGN)
begin
   COUNTER_FRAME_DRV <= std_logic_vector(COUNTER_FRAME_UGN + 1);
end process Combinational;
```

```
Combinational_local_Low: process(Counter_Local_Low_UGN)
begin
  COUNTER_LOCAL_Low_DRV <= std_logic_vector(COUNTER_LOCAL_Low_UGN + 1);
end process Combinational_local_Low;

Combinational_local_High: process(Counter_Local_High_UGN)
begin
  COUNTER_LOCAL_High_DRV <= std_logic_vector(COUNTER_LOCAL_High_UGN + 1);
end process Combinational_local_High;

   S0     <= '1'
         when ((COUNTER_FRAME_UGN > 0 and COUNTER_FRAME_UGN <= 40)
         or (COUNTER_FRAME_UGN > 15980 and COUNTER_FRAME_UGN <= 16500))
         else '0';
   S1     <= '1'
         when (COUNTER_FRAME_UGN > 44 and COUNTER_FRAME_UGN <= 15980)
         else '0';
   S2_DRV <= '1'
         when ((COUNTER_FRAME_UGN > 44 and COUNTER_FRAME_UGN <= 1684)
         or (COUNTER_FRAME_UGN > 16502 and COUNTER_FRAME_UGN <= 18142))
          else '0';
   S3_DRV <= '1'
         when (COUNTER_FRAME_UGN > 8180 and COUNTER_FRAME_UGN <= 9820)
          else '0';

   S2 <= S2_REG;
   S3 <= S3_REG;

   PH_SAL <= NOT(S2_REG);
   PH_SAH <= NOT(S3_REG);
   PH_PRE <= '1'
         when (((COUNTER_LOCAL_High_UGN rem 64) = 1)
         or ((COUNTER_LOCAL_Low_UGN rem 64) = 1))
         else '0';

--The signals for outside of the chip for test
TX_IN <= '1'when ((COUNTER_FRAME_UGN > 45 and COUNTER_FRAME_UGN <= 105)
```

```
        or (COUNTER_FRAME_UGN > 170 and COUNTER_FRAME_UGN <= 230)
        or (COUNTER_FRAME_UGN > 420 and COUNTER_FRAME_UGN <= 480)
        or (COUNTER_FRAME_UGN > 920 and COUNTER_FRAME_UGN <= 980)
        or (COUNTER_FRAME_UGN > 1920 and COUNTER_FRAME_UGN <= 1980)
        or (COUNTER_FRAME_UGN > 3920 and COUNTER_FRAME_UGN <= 3980)
        or (COUNTER_FRAME_UGN > 7920 and COUNTER_FRAME_UGN <= 7980)
        or (COUNTER_FRAME_UGN > 15920 and COUNTER_FRAME_UGN <= 15980)
        ) else '0';

PRESET_L <= '1'
                when ((COUNTER_FRAME_UGN = 1)
                or (COUNTER_FRAME_UGN = 15999))
                 else '0';
PRESET_H <= '1'
                when (COUNTER_FRAME_UGN = 63) else '0';

NWRITE_L <= '1'
                when (((COUNTER_LOCAL_Low_UGN rem 64) /= 0)
                and (COUNTER_LOCAL_Low_UGN > 64))
                 else '0';
NWRITE_H <= '1'
                when (((COUNTER_LOCAL_High_UGN rem 64) /= 0)
                and (COUNTER_LOCAL_High_UGN > 64))
                 else '0';

--The second frame GUARD signal is switched off
GUARD <= '1'when (
           ((COUNTER_FRAME_UGN > 2 and COUNTER_FRAME_UGN <= 39)
        or (COUNTER_FRAME_UGN > 109 and COUNTER_FRAME_UGN <= 111)
        or (COUNTER_FRAME_UGN > 234 and COUNTER_FRAME_UGN <= 236)
        or (COUNTER_FRAME_UGN > 484 and COUNTER_FRAME_UGN <= 486)
        or (COUNTER_FRAME_UGN > 984 and COUNTER_FRAME_UGN <= 986)
        or (COUNTER_FRAME_UGN > 1984 and COUNTER_FRAME_UGN <= 1986)
        or (COUNTER_FRAME_UGN > 3984 and COUNTER_FRAME_UGN <= 3986)
        or (COUNTER_FRAME_UGN > 7984 and COUNTER_FRAME_UGN <= 7986)
        or (COUNTER_FRAME_UGN > 15984 and COUNTER_FRAME_UGN <= 16499)
        ) and (COUNTER_NFRAME_UGN /= 2)
        )
```

```
            else '0';

  --Controlling the inout port for Testing
  NS <= '0' when (COUNTER_NFRAME_UGN = 2) else '1';

  --Counting the number of frames
  Sequential_FRAME: process(RESET_REF, RESET_SYSTEM)
  begin
   if(RESET_SYSTEM = '1') then
      COUNTER_NFRAME_REG <= (OTHERS=>'0');
   end if;

   if(RESET_REF'event and RESET_REF = '1') then
    COUNTER_NFRAME_REG <= COUNTER_NFRAME_DRV;
   end if;
  end process Sequential_Frame;

  FRAME: process(COUNTER_NFRAME_UGN)
  begin
   if (COUNTER_NFRAME_UGN = 66) then
    COUNTER_NFRAME_DRV <= (others=>'0');
   else
    COUNTER_NFRAME_DRV <= std_logic_vector(COUNTER_NFRAME_UGN + 1);
   end if;
  end process Frame;

  COUNTER_NFRAME_UGN <= unsigned(COUNTER_NFRAME_REG);

end architecture structure;
```