A Wide-Dynamic-Range Time-Based CMOS Imager

by

Micah G. O'Halloran

Bachelor of Science in Electrical and Computer Engineering
University of Florida, May 2000

Master of Science in Electrical Engineering and Computer Science
Massachusetts Institute of Technology, September 2002

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2008

Author .....
Department of Electrical Engineering and Computer Science
January 31, 2008

Certified by .....
Rahul Sarpeshkar, Ph.D.
Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Terry P. Orlando, Ph.D.
Chairman, Committee on Graduate Students
Department of Electrical Engineering and Computer Science

A Wide-Dynamic-Range Time-Based CMOS Imager

by

Micah G. O'Halloran

Submitted to the Department of Electrical Engineering and Computer Science
on January 31, 2008, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This thesis describes a novel dual-threshold time-based current sensing algorithm suitable for use in wide-dynamic-range CMOS imagers. A prototype 150 × 256 pixel imager employing this algorithm experimentally achieves 95.5 dB dynamic range, 37 dB peak signal-to-noise ratio (SNR), and a highly-linear transfer characteristic while consuming 1.79 nJ/pixel/frame, making it one of the most energy-efficient wide-dynamic-range imagers reported. The individual pixels experimentally achieve 98.8 dB dynamic range and 44 dB peak SNR. The array performance lags slightly behind that of the individual pixels due to the additional noise power contributed to the array data by pixel-to-pixel mismatch effects, attributed primarily to gain and dark-current fixed pattern noise (FPN). The dual-threshold algorithm is also shown to improve low-illumination SNR by 6.1 dB and overall array dynamic range by more than 6.0 dB compared with auto-zeroing alone. The prototype imager implements pixels and their associated 18-bit timing memories in separate on-chip arrays linked by a 200 MHz time-domain-multiplexed communication bus, enabling a pixel pitch of 12.5 μm with 42.7% fill factor in a 0.18-μm 1.8-V CMOS process.

Four innovations are contributed by this thesis over previous work, leading to the performance outlined above. First, a novel dual-threshold time-based current sensing algorithm is proposed that forces each single-slope integrating pixel to cross two threshold levels per frame – once just after reset and a second time after a near-optimal amount of photo-generated charge has been collected. This differential measurement technique eliminates offset FPN and pixel reset noise, and reduces comparator 1/f noise. Second, synchronous threshold detection is employed, yielding significant power savings compared with asynchronous approaches in this application, and the resulting time-domain quantization noise introduced by the synchronous detection is analyzed. Third, a method of optimizing the global dual-threshold waveform and associated pixel threshold-detection times is presented. The method ensures that the quantization noise introduced by the algorithm remains negligible compared to the intrinsic pixel noise floor, while simultaneously minimizing the number of threshold detections employed, and thus energy consumed. Fourth, a novel capacitively-coupled pixel topology is introduced that enables highly-linear responses to be achieved with this algorithm while minimizing the common-mode input range of the pixel comparator, simplifying its design. Together, these innovations result in energy-efficient wide-dynamic-range pixel operation. The imager is thus suitable for use in portable applications in environments that are challenging for conventional imagers, e.g., when indoor or shadowed lighting and outdoor lighting are simultaneously present in an image.

Thesis Supervisor: Rahul Sarpeshkar, Ph.D.
Title: Associate Professor of Electrical Engineering

Acknowledgments

I would first like to thank my advisor, Prof. Rahul Sarpeshkar, for his support and encouragement throughout graduate school. His enthusiasm for learning continues to impress me, and his goal for intuition in all problems inspires me to strive for the same.

Without my colleagues and friends in AVBS, as well as in other groups in the department, graduate school would not have been nearly as enjoyable or enriching. I would like to thank all of you for your support and friendship during these years, and hopefully for many to come.

I am eternally grateful to the Catalyst Foundation, who graciously funded my research with little taxation. Without their support, none of the work presented here would have been possible.

Finally, I would like to give a special thank you to my wife Amy. Her support and love helped me stay sane through this arduous process. Time for some Racer 5, stink!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter motivates the need for imagers capable of capturing wide-dynamic-range scenes, outlines the main contributions of this thesis, and presents an overview of the material presented in the following chapters.

1.1 Motivation

Conventional charge-coupled devices (CCDs) and CMOS active-pixel sensors (APSs) have each found a niche within the field of digital imaging due to their complementary strengths in the areas of power consumption, image quality, and ease of system integration. However, both technologies share a common shortfall – the inability to faithfully capture images with wide intra-scene dynamic range. While the dynamic range of these sensors typically falls between 10^3:1 (60 dB) and 10^4:1 (80 dB), real-world scenes can exhibit illuminations spanning a range of 10^5:1 (100 dB) or more [1]. When confronted with such a scene, these sensors either fail to capture significant information in the darkest areas of the image, saturate in the brightest areas, or both. In a growing number of applications, particularly within the automotive and security sectors, the resulting information loss is not tolerable, and has led the push to develop modified sensor topologies with improved dynamic range [2].

The moderate dynamic range of conventional imagers can be traced to their use of pixel-level capacitive integration to overcome the shot-noise limitations associated with measuring small photocurrents [3]. Since CCD and APS imagers encode photocurrent information in the amount of charge or voltage integrated by each pixel during a globally-defined integration window, they must balance the need for a long window in dimly-illuminated pixels to

overcome the intrinsic noise floor against the need for a short window in highly-illuminated pixels to avoid saturation. These conflicting requirements can only be met to a certain extent, leading to an overall moderate dynamic range.

Various CMOS APS imager topologies have been proposed to address this limitation. Among the potential solutions, time-based pixels, which encode photocurrent information in the amount of time it takes each pixel to integrate to a globally-defined threshold, offer several potential advantages compared with traditional CCD and APS approaches. Unfortunately, a major limitation of many past time-based approaches has been that they fail to achieve wide-dynamic-range capture in video-rate applications. In fact, when compared with conventional CCD and APS pixels, many time-based approaches capture a roughly complementary portion of the input photocurrent range. However, a few implementations have recognized and addressed this problem through modifications to the generic time-based approach. One solution employs a time-varying threshold that begins each frame at a high level to maximize the charge quanta collected by brightly-illuminated pixels, and decreases over the course of the frame to ensure dimly-illuminated pixels also reach threshold within an allotted video frame time. This technique serves as the starting point for the work presented in this thesis.

1.2 Thesis Contributions

This thesis helps improve the performance of video-rate time-based imagers through the following four contributions:

- A novel dual-threshold time-based current sensing algorithm is proposed that forces each single-slope integrating pixel to cross two threshold levels per frame – once just after reset and a second time after a near-optimal amount of photo-generated charge has been collected. This differential measurement technique eliminates offset FPN and pixel reset noise, and reduces comparator 1/f noise.

- Synchronous threshold detection is employed and is shown to yield significant power savings compared with asynchronous approaches in this application. The resulting time-domain quantization noise introduced by the synchronous threshold detections is also analyzed.

- A method of optimizing the global dual-threshold waveform and associated pixel threshold-detection times is presented. The method ensures that the quantization noise introduced by the algorithm remains negligible compared to the intrinsic pixel noise floor, while simultaneously minimizing the number of threshold detections employed, and thus energy consumed.

- A novel capacitively-coupled pixel topology is introduced that enables highly-linear responses to be achieved with the dual-threshold algorithm while minimizing the common-mode input range of the pixel comparator, simplifying its design.

These innovations are incorporated into the design of a prototype dual-threshold time-based CMOS imager. The imager implements pixels and their associated 18-bit timing memories in separate on-chip arrays linked by a 200 MHz time-domain-multiplexed communication bus, enabling a pixel pitch of 12.5 μm with 42.7% fill factor in a 0.18-μm 1.8-V CMOS process. The prototype experimentally achieves an array dynamic range of 95.5 dB and 37 dB peak SNR, while consuming 1.79 nJ/pixel/frame, making it one of the most energy-efficient wide-dynamic-range imagers reported.

1.3 Thesis Organization

This chapter motivated the need for wide-dynamic range imagers and outlined the main contributions of this thesis. Chapter 2 discusses the basics of solid-state imaging, starting with photons and working towards basic pixel structures, and briefly reviews the dynamic-range-expansion techniques that have been employed in previous work.

Chapter 3 begins by comparing generic time-based pixels with traditional integrating pixels, highlighting the pros and cons of the time-based approach. It then systematically presents the problems encountered in past time-based pixel implementations, in the process developing the synchronous dual-threshold capacitively-coupled pixel and algorithm to address these limitations. Finally, it analyzes the intrinsic and quantization noise of the dual-threshold pixel and algorithm, deriving theoretical predictions of pixel dynamic range and SNR performance.

Chapter 4 outlines the design of a prototype 150 $\times$ 256 pixel dual-threshold CMOS imager. The pixels and their associated 18-bit memory cells are discussed in detail, as

is the time-domain-multiplexed pixel-to-memory communication strategy and implementation. Particular attention is paid to communication latency and energy consumption. Finally, the support circuitry necessary for proper operation of the pixel and memory arrays is also discussed.

Chapter 5 discusses the experimental setup and techniques employed to test the prototype imager chip, and presents measured performance data from it. It also compares these measured results with theory and previously-reported implementations.

Chapter 6 summarizes the contributions made in this work, and suggests potential directions that could be explored to build upon it.

Chapter 2

Background Information

Modern image sensors consist of a one- or two-dimensional (1D or 2D) array of spatially discrete photosensitive elements, called pixels, whose primary tasks are to transduce local incident electromagnetic radiation into a quantity that can be more readily manipulated and measured electronically, and to convey either this raw transduction product or a processed version of it to the periphery of the array. Various methods of transduction have been employed in past sensors, with the chosen approach strongly depending on the particular slice of the electromagnetic spectrum the sensor is designed to detect and on the desired sensitivity of the detection. Fortuitously, within the visible region of the spectrum – which we are solely concerned with here – the most ubiquitous present-day circuit substrate, silicon, exhibits a combination of properties that together make it a sensitive transducer of photons into mobile electron-hole (e-h) pairs. This combination of electrical and optical properties has enabled the realization of highly-integrated solid-state imagers, in which photon transduction, e-h pair measurement, and downstream signal processing are integrated on a single slab of crystalline silicon.

This chapter begins with a brief discussion of the most important features of the photon-to-e-h pair transduction process. It then reviews the various methods that have been employed in past pixel designs to collect, measure, and convey the local e-h pair generation rates from the pixel to the periphery of the array. Particular attention is paid to the engineering choices that have led to the dynamic range limitations of modern CMOS active-pixel sensors (APS) and the solutions that have been proposed to extend this dynamic range.

2.1 Silicon Transduction of Photons

A photon possessing an energy greater than the silicon bandgap energy of approximately 1.12 eV (@ T = 300 K) can be absorbed by the crystal lattice, in the process generating a mobile *e-h* pair. Since the energy of visible photons (400 nm $\leqslant \lambda \leqslant$ 700 nm) spans the range from approximately 1.77 eV to 3.1 eV, all visible photons are capable of *e-h* pair generation in silicon. This absorption/generation process exhibits several dependencies that are relevant to imager design. First, absorption is not confined to the surface of the crystal, but is distributed over a range of depths below it, with the probability of a photon surviving to a depth x without being absorbed modeled by the Beer-Lambert law:

$$P(x) = e^{-\alpha x}. \tag{2.1}$$

Second, the absorption coefficient α in the exponent of Equation 2.1 is a function of both the absorbing material and the wavelength of the incident photons. Values for α as a function of photon wavelength have been empirically determined for silicon, and a representative curve which includes the visible region of the spectrum is illustrated in Figure 2-1. From this data, one can calculate that over 99% of violet photons ($\lambda \approx$ 400 nm) are absorbed within a depth of 0.7 μm into the crystal, while it takes roughly 20 μm to absorb the same percentage of red photons ($\lambda \approx$ 700 nm).

This spatial and wavelength dependence of photon transduction into *e-h* pairs directly impacts pixel performance in practice. Since each pixel infers incident photon flux by measuring the local *e-h* pair generation rate, unless the measurement technique collects *e-h* pairs at all depths up to roughly 20 μm with equal probability, the pixel will exhibit a spectrally-varying quantum efficiency. This parameter, defined by

$$\eta(\lambda) = \frac{I_{photo}}{q} \cdot \frac{h \cdot c}{\lambda} \cdot \frac{1}{I_{lux} \cdot f(\lambda) \cdot A_{pd}}, \tag{2.2}$$

quantifies the number of charge quanta collected by the photodiode per incident photon in terms of the photon wavelength λ, the photodiode current I_{photo}, the input illuminance I_{lux}, the wavelength-dependent illuminance-to-irradiance conversion factor $f(\lambda)$, and the photodiode area A_{pd}, along with Planck's constant $h = 6.626 \times 10^{-34}$ J·s, the speed of light $c = 3 \times 10^{8}$ m/s, and the unit charge $q = 1.602 \times 10^{-19}$ C. The dependence of this

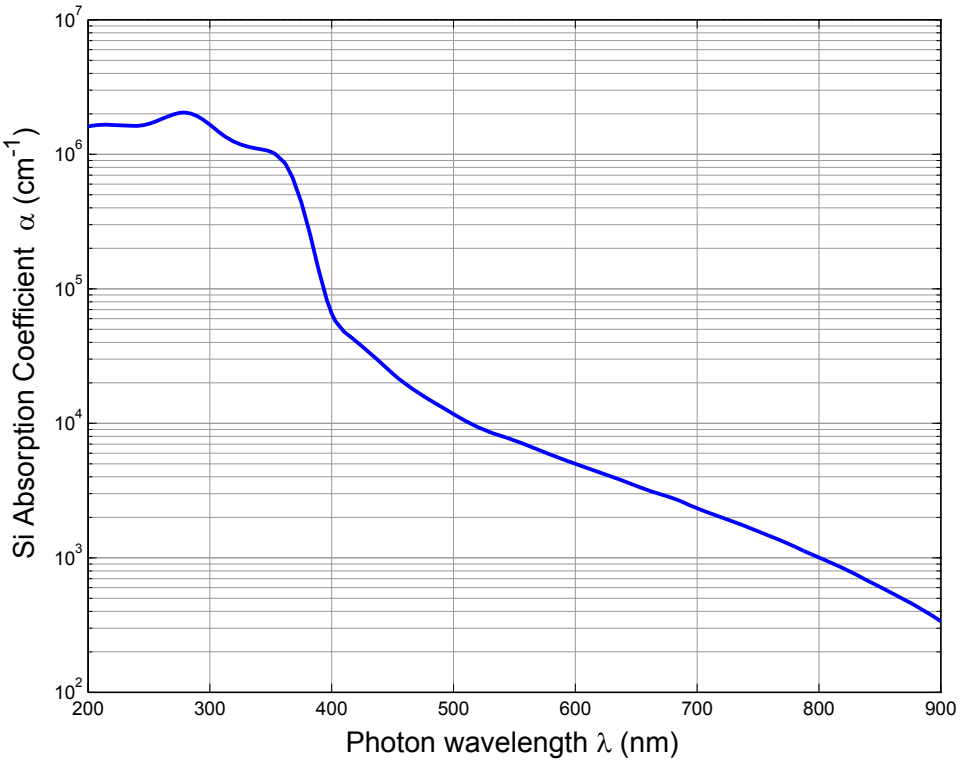

Figure 2-1: Silicon absorption coefficient versus photon wavelength (adapted from [4]).

parameter on photon wavelength can potentially result in an inseparable mixing of spectral and irradiance information, or as is the case in at least one design, can be capitalized upon to infer color information [5].

2.2 Collection of *e-h* Pairs

The magnitude of the 99% absorption depths calculated above are critical in that they are shallow enough to enable a significant percentage of the resulting *e-h* pairs to be collected by structures fabricated near the crystal's surface. The primary goal of any collection structure is to spatially segregate the electrons from the holes, both to decrease the risk of recombination and so that the net generated charge can be measured. Typically this is accomplished using either the built-in electric ($\vec{E}$) field within a *pn*-junction's space-charge region, or an $\vec{E}$ field created by applying a voltage potential across a MOS structure (Figure 2-2). Both approaches yield a photocurrent along the direction of the $\vec{E}$ field that is proportional to the local *e-h* pair generation rate.

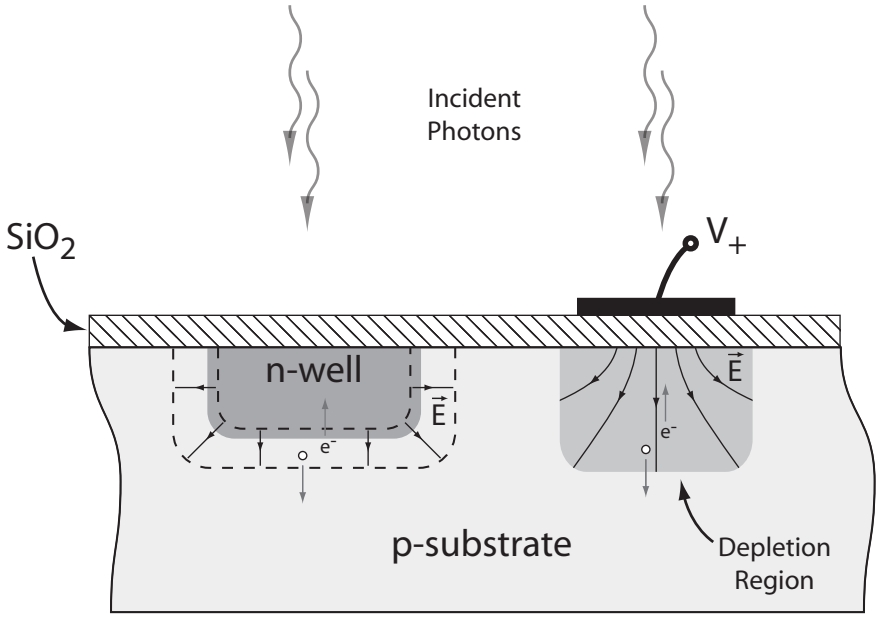

Figure 2-2: Two *e-h* pair collection structures. Electrons and holes can be easily separated using the electric field in a diode's space-charge region (*left*), or a MOS depletion region (*right*). Note that for the structures shown, the electrons are collected while the holes are discarded to the substrate.

A major limitation of both of these structures in modern CMOS processes is the relatively shallow depths over which these $\vec{E}$ fields extend. For example, in the case of the photodiode structure, the space-charge region extends over a width of approximately

$$w_{scr} = \Big[2\frac{\epsilon_{si}}{q}(\phi_{bi} + V_{pd})\Big(\frac{1}{N_A} + \frac{1}{N_D}\Big)\Big]^{\frac{1}{2}}, \tag{2.3}$$

where N_A and N_D are the p and n region doping concentrations, respectively, ϵ_{si} is the permittivity of silicon, V_{pd} is the photodiode reverse-bias voltage, and ϕ_{bi} is the junction built-in potential, given by

$$\phi_{bi} = \frac{kT}{q} ln\Big[\frac{N_A N_D}{n_i^2}\Big]. \tag{2.4}$$

In a typical 0.18-μm process, the well-to-substrate diode's metallurgical junction is located at a depth of around 2 μm, and the doping levels are in the neighborhood of $N_A = 5 \times 10^{14}$ cm^{-3} in the p-substrate and $N_D = 5 \times 10^{17}$ cm^{-3} in the n-well. At a reverse bias of $V_{pd} = 1.8$ V and temperature T = 300 K, these relations yield a space-charge region width of approximately $w_{scr} = 2.6$ μm, which extends almost solely on the substrate side of the

junction due to the drastic difference in doping concentrations between the two regions. Thus, for this process, *e-h* pairs generated between a depth of 2 μm and 4.6 μm will be immediately separated and collected with high efficiency[1], while those generated outside of this region will be captured with lower efficiency since they must first diffuse into the space-charge region before they can be collected. Based on the frequency-dependent absorption depth dependencies presented in Section 2.1, we would expect this photodiode's quantum efficiency to be significantly less than 100% at all frequencies, and to peak somewhere in the middle of the visible spectrum.

The quantum efficiency of the MOS structure typically exhibits similar peaking in the middle of the visible spectrum, but for a different reason [6]. Unlike the photodiode, the MOS depletion region's $\vec{E}$ field extends all the way to the Si-SiO_2 interface over the entire sensor area, and thus it is reasonable to expect the quantum efficiency of this structure to be superior for high-frequency photons. However, a significant fraction of these photons are absorbed by the polysilicon gate before reaching the substrate. Like the photodiode, the quantum-efficiency of the MOS structure at the low-frequency end of the spectrum is limited by the depth to which the depletion region reaches, which is governed by the relation

$$w_{dep} = \left[\frac{2\epsilon_{si}\psi_s}{qN_A}\right]^{\frac{1}{2}}, \tag{2.5}$$

where ψ_s is the substrate potential at the Si-SiO_2 interface, called the surface potential, and N_A is the doping concentration of the silicon below the MOS gate. Although not shown in Figure 2-2, MOS channel regions in standard CMOS processes are typically doped at much higher levels than the underlying substrate, on the order of the n-well doping levels mentioned above, thus the MOS depletion depth will usually be much smaller than the space-charge region width calculated for the photodiode above. However, in technologies specifically tailored to imager design, i.e., CCD processes, the substrate doping can be reduced significantly, allowing for depletion depths on the order of hundreds of microns at reasonable voltage bias levels [7].

[1]This statement ignores the space-charge region along the perimeter of the junction, which will improve the photodiode response to high-frequency photons.

2.3 Shot-Noise Limited Photocurrent Measurement

It was mentioned in Chapter 1 that modern pixels employ capacitive integration to overcome the shot-noise limitations associated with measuring small photocurrents. To understand this statement, consider modeling a photocurrent with mean $\overline{I_{photo}}$ as a Poisson arrival process. Under this model, the average number of charge quanta arriving per unit time is given by $\lambda = \frac{\overline{I_{photo}}}{q}$, while the probability of actually observing n charge quanta arriving during a window of length Δt is given by

$$P(n, \lambda, \Delta t) = \frac{(\lambda \cdot \Delta t)^n}{n!} e^{-\lambda \cdot \Delta t}. \tag{2.6}$$

An interesting property of Poisson arrival processes is that both their mean and variance are given by

$$\overline{n} = \sigma_n^2 = \lambda \cdot \Delta t. \tag{2.7}$$

Thus, by integrating the number of charge quanta arriving over a window of length Δt, the maximum SNR to which $\overline{I_{photo}}$ can be estimated is given by

$$\mathrm{SNR}_{\mathrm{max,Poisson}} = 10 \cdot \log_{10} \left(\frac{\overline{n}}{\sigma_n} \right)^2 = 10 \cdot \log_{10}(\lambda \cdot \Delta t) \quad \mathrm{dB}. \tag{2.8}$$

This result demonstrates that the measurement SNR is fundamentally limited by the number of charge quanta collected $\lambda \cdot \Delta t$, and explains why modern image sensors attempt to maximize Δt, as will be seen in the following sections.

2.4 Standard Pixel Topologies

The vast majority of solid-state imagers designed to operate in the visible region rely on the mechanisms outlined in the previous sections to generate local photocurrents that are (ideally) proportional to the incident photon flux at each pixel. The primary way in which the resulting imager implementations differ is in how they measure these photocurrents. The various approaches that have been employed in previous designs are discussed in this section.

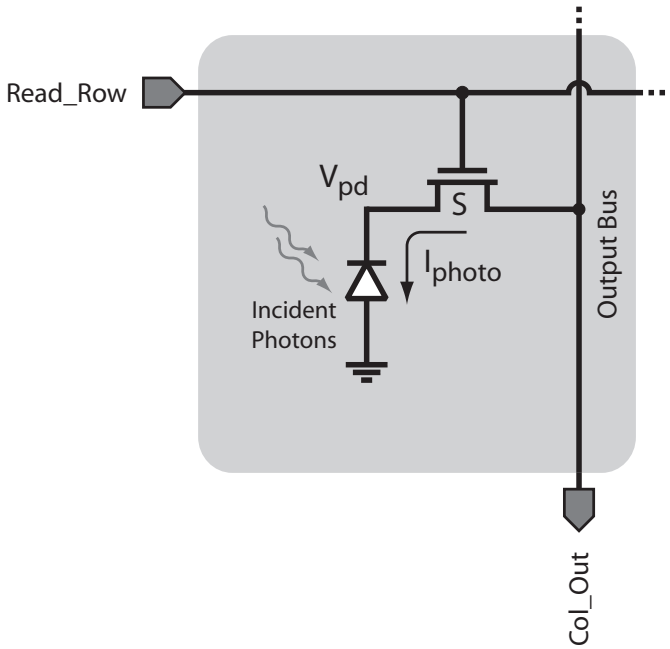

Figure 2-3: A generic passive pixel sensor.

2.4.1 Passive Pixel Sensors

The defining characteristic of a passive pixel sensor (PPS) is that it does not contain active pixel-level signal amplification. A generic passive pixel, shown in Figure 2-3, consists of a photodetector, in this case a photodiode, and a switching element S, which is used to periodically connect the photodiode to an output bus that is shared among many pixels, typically all pixels in the same column of the array. The earliest photodiode-based PPS designs were operated in non-integrating mode [8, 9], which would be implemented in this generic pixel by periodically closing switch S and sensing the value of I_{photo} for a length of time t_{read} using circuitry connected to the output bus at the periphery of the array. Since many pixels share the same output bus in this approach, $t_{read} \ll t_{frame}$, where t_{frame} is the total time allotted to capture a single image frame from the entire pixel array.

One major disadvantage of the non-integrating passive pixel approach is that each pixel's photocurrent is measured only during the short time interval t_{read}. Soon after non-integrating PPS sensors were introduced, it was demonstrated that the same pixel structure could be operated in integrating-mode, significantly improving its sensitivity and SNR [3]. In this mode, depicted in Figure 2-4, switch S is closed at the start of a new frame for a

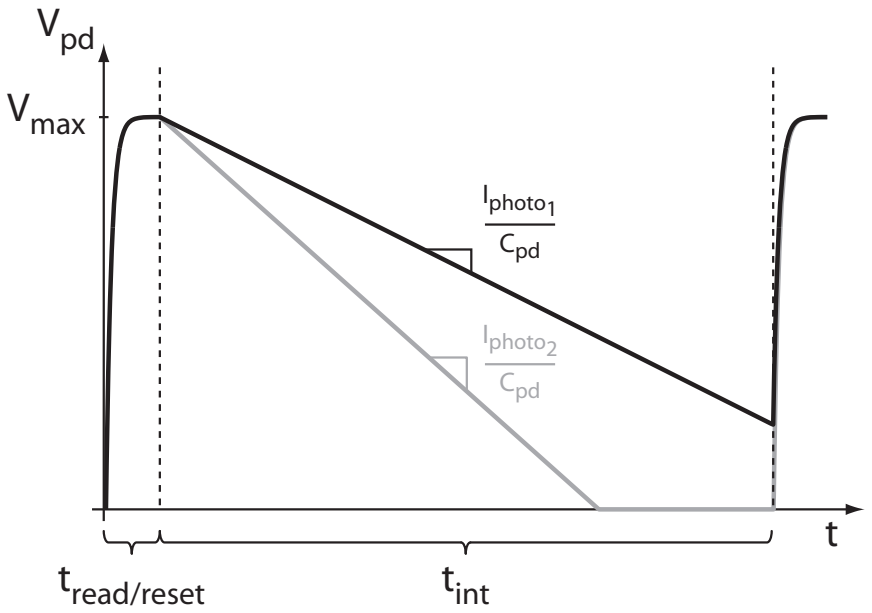

Figure 2-4: This figure depicts two potential waveforms for a passive-pixel operating in integrating mode. I_{photo_1} represents a typical non-saturating input, while I_{photo_2} represents a typical saturating input. Note that for simplicity the nonlinearity of C_{pd}, the reverse-biased photodiode depletion capacitance, is ignored and the sensor saturation is arbitrarily assumed to occur at 0V.

time t_{read}, allowing circuitry at the periphery of the array to simultaneously measure the charge integrated by the pixel during the previous frame and reset the reverse-bias photodiode voltage V_{pd} to an initial voltage V_{max} in preparation for the current frame's integration period. Next, switch S is opened and I_{photo} is allowed to discharge the reverse-biased photodiode depletion capacitance C_{pd} for the remainder of the frame time, t_{int}, at which point the pixel is again read/reset. By inferring I_{photo} based on the total charge integrated over the full frame interval $t_{frame} = t_{read} + t_{int}$ rather than over the much shorter read interval t_{read}, the theoretical maximum SNR of the measurement is improved by

$$\mathrm{SNR_{new}} - \mathrm{SNR_{old}} = 10 \cdot \log_{10}\left[\frac{t_{frame}}{t_{read}}\right] \quad \mathrm{dB}. \tag{2.9}$$

However, this increase in sensitivity and SNR does come at a price. As illustrated by the waveform associated with I_{photo_2} in Figure 2-4, under high illumination the photodiode parasitic capacitance can discharge so deeply that it forward biases, an event known as saturation. Because all saturating photocurrents yield virtually the same photodiode voltage

at the end of t_{int}, the present algorithm will measure nearly the same charge from all of them[2]. This many-to-one mapping makes these saturating input photocurrent levels indistinguishable.

It is interesting to note that in his original paper on integrating-mode operation of the *pn*-junction, Weckler recognized that the technique imposes limitations on sensor dynamic range [3]. However, because of its inherent sensitivity and SNR advantages, nearly all modern solid-state sensors employ integration. Thus, as will be seen later in this chapter and in the next, the vast majority of dynamic range expansion techniques that have been proposed attempt to address the saturation phenomena described above.

Despite their advantages of simplicity and high fill factor, passive pixel sensors have lost popularity due to their susceptibility to column bus parasitic effects, which introduce noise terms that tend to increase with array size [6]. The most detrimental of these effects are due to pixel-level leakage currents and parasitic capacitance [10]. The leakage currents arise due to subthreshold leakage in the pixel's MOS switch and optically-generated carriers in the substrate that diffuse to and are collected by the MOS drain terminal (assumed to be the terminal connected to the output bus in Figure 2-3). These leakage currents are contributed to the bus by every pixel, independent of whether or not the pixel is being read, and cannot be distinguished from the integrated photocurrent of the pixel itself. Since all pixels contribute to the leakage, it grows in proportion to the number of rows in the array, decreasing overall sensor SNR as rows are added.

The parasitic bus capacitance C_{bus} also degrades the read noise of PPS arrays, but in a more subtle way. To understand the mechanisms involved, refer to Figure 2-5, which depicts a simple charge integrator topology that might be used as the readout circuit at the end of the column bus. Assuming the amplifier gain is large, when a pixel is connected to the column bus its photodiode voltage $V_{pd}d$ is forced to V_{ref} and the charge that was collected during the previous integration period on the photodiode's parasitic capacitance C_{pd} is transferred to the feedback capacitor C_f. Since the ratio $\frac{C_{pd}}{C_f}$ defines the voltage gain of this transfer, C_f must be comparable in size to C_{pd} to avoid a significant loss in sensitivity [11]. It is also desirable for the amplifier feedback factor $f = \frac{C_f}{C_f+C_{bus}}$ to be as close to unity as possible, since for a fixed power consumption the integrator's settling time and output noise power grow as $\frac{1}{f}$ [6]. Unfortunately, typically C_{bus} is much larger than C_f

[2]The saturation voltage is not exactly 0V, rather it is governed by the diode exponential characteristic.

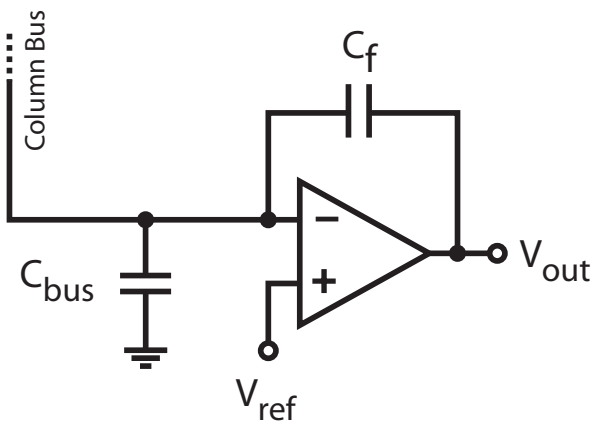

Figure 2-5: Charge integrator topology which might be used to read out PPS charge packets.

and, like the leakage currents, is proportional to the number of rows in the array.

2.4.2 Charge-Coupled Devices

Charge-coupled devices (CCDs) consist of two or more closely-spaced MOS capacitors capable of storing packets of minority carriers and transferring these packets from one capacitor to another with very high efficiency [12]. As was explained in Section 2.2, applying a voltage of the proper sign to the gate of a MOS capacitor generates a depletion region (potential well) in the semiconductor that can separate *e-h* pairs and trap minority carriers near the Si-SiO_2 interface. As illustrated in Figure 2-6, by properly timing the creation and removal of potential wells at adjacent MOS capacitors, minority carriers can be forced to transfer from one well to another. These collection and charge-transfer properties allow CCD devices to be used to create imaging arrays. In a simplistic approach, photo-generated carriers are accumulated in CCD pixels over the entire array during the integration time t_{int}, then the charge packets are sequentially shifted to the periphery of the array and measured by a charge integrator, much like the one shown in Figure 2-5. Note that due to their integration-based operation, CCD pixels are subject to the same saturation and dynamic range limitations as integrating PPS designs.

Although the operation of a CCD imaging array described above is nearly identical to that of an integrating PPS array, there is a significant difference between the two approaches that contributes to the superior noise performance of CCD imagers. Rather than transfer-

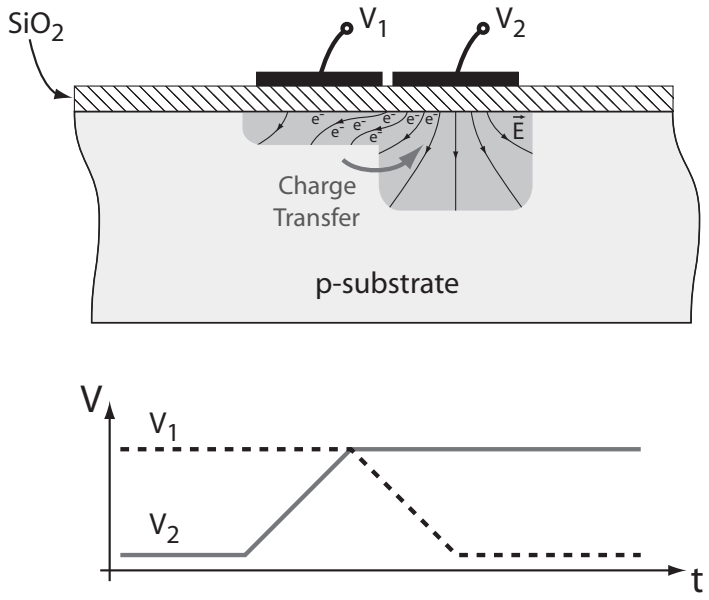

Figure 2-6: Charge transfer between adjacent MOS capacitors. In this illustration, the MOS capacitor on the left initially contains a pool of minority carriers, possibly collected from photo-generation (see Figure 2-2). By properly timing the creation and removal of potential wells at adjacent MOS capacitors, a lateral $\vec{E}$-field can be generated that, along with diffusion, transfers minority carriers between the wells with high efficiency.

ring each pixel's integrated charge to the periphery of the array via a shared bus, CCD imagers transfer the charge packets out through a large number of neighbor-to-neighbor exchanges. The advantage of this approach is that the parasitic capacitance C_{bus} is completely eliminated from the virtual ground of the charge-integrating readout amplifier (see Figure 2-5), dramatically reducing its read noise. However, since it can take many successive transfers, and thus significant time, for a charge packet to reach the periphery of the array, and since MOS capacitors are effective collectors of stray minority carriers, CCDs are still subject to errors due to stray charge collection. Dedicated light-shielded CCD transfer lines are one method of reducing this error term, but in general it is combated by employing rapid transfer rates, which requires both high voltages and high operating frequencies. As a result, CCD imagers tend to consume significantly more power than their CMOS counterparts. Despite this drawback, many of the lowest read and fixed pattern noise imagers available are CCD-based [6, 13].

2.4.3 Active Pixel Sensors

As opposed to CCDs, which overcome bus parasitics by essentially eliminating the bus, an active pixel sensor (APS) limits the effect of bus parasitics by employing active pixel-level amplification to buffer pixel readout [11]. A standard three-transistor photodiode-based APS pixel is shown in Figure 2-7. Compared to a passive-pixel, this circuit has two extra transistors, M_r and M_f, and readout now occurs on the column bus in voltage mode rather than charge mode. However, the principles of operation remain very similar to an integrating

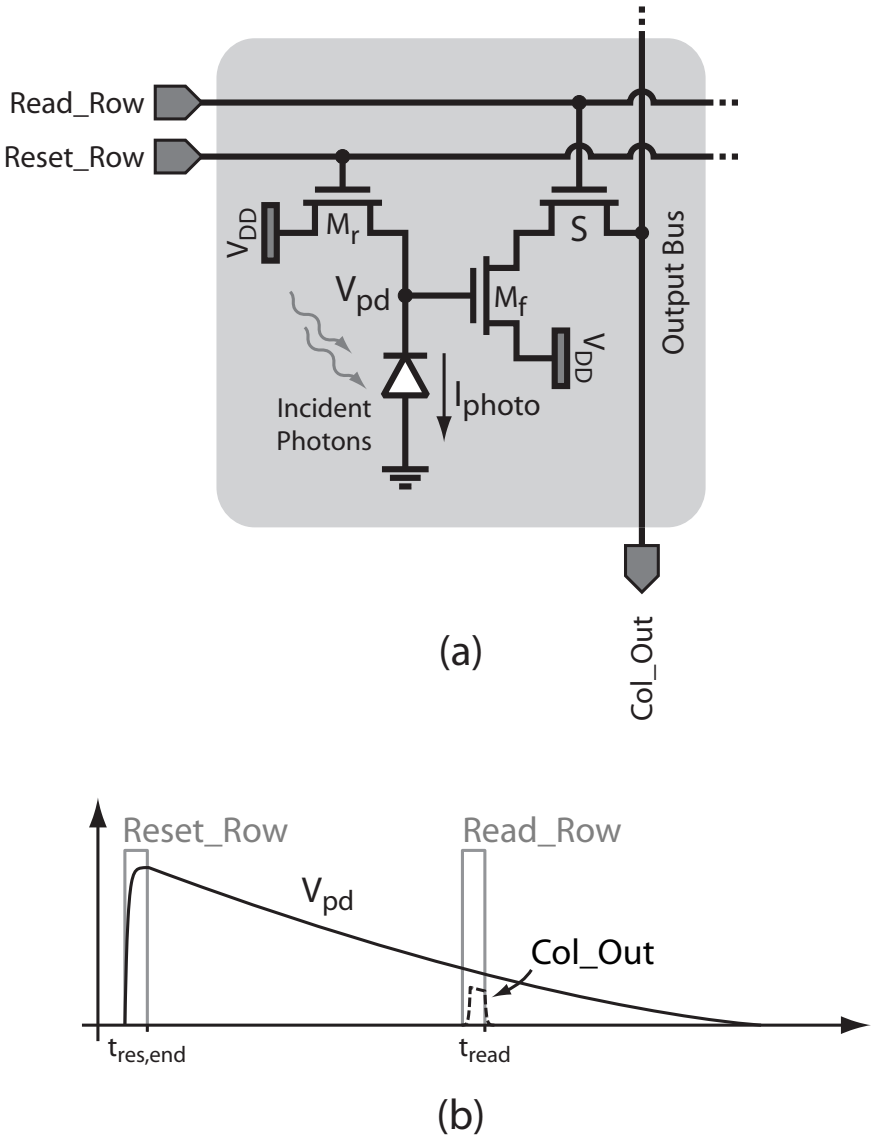

Figure 2-7: A three-transistor active pixel sensor (a) and typical operating curves (b).

passive pixel. As shown in part (b) of the figure, at the start of a new frame `Reset_Row` pulses `HIGH`, turning M_r `ON` and resetting the photodiode voltage V_{pd} to roughly a threshold voltage below V_{DD}. The reset phase ends when `Read_Row` transitions `LOW` at $t_{res,end}$, allowing the pixel photocurrent I_{photo} to discharge the parasitic capacitance of node V_{pd}, which includes the nonlinear photodiode capacitance C_{pd} as well as other parasitics due to the source of M_r and the gate of M_f, for an integration period of length $t_{int} = t_{read} - t_{res,end}$. Just before the end of the integration period, switch S is activated by `Read_Row` going `HIGH` and a fixed current source (not shown) at the bottom of the column bus biases M_f in source-follower configuration. The voltage V_{pd} can be inferred from the resulting column bus voltage at t_{read} and knowledge of the gate-to-source voltage of M_f. Additionally, the amount of collected charge can be determined if the properties of the parasitic capacitance at V_{pd} are known.

The source-follower readout topology employed in active pixel arrays eliminates both major sources of bus error present in passive pixel designs. First, to achieve high readout speeds, each column's source-follower current source is biased at a level much higher than the typical column leakage current, making the voltage readout error due to this effect negligible. Second, under the constraint of constant output bandwidth, the thermal noise power at the output of a source-follower topology will scale as $\frac{1}{C_{bus}}$, decreasing with increasing bus capacitance[3]. This decreased sensitivity to bus parasitics comes at the the expense of increased FPN, primarily due to pixel-to-pixel variations in the threshold voltage of transistor M_f, and increased $\frac{1}{f}$ noise, due to the same transistor [13]. However, correlated double sampling (CDS) techniques have been successfully employed to reduce these error terms, leaving reset noise at V_{pd} as the dominant error source in these pixels [6].

2.5 Wide-Dynamic-Range Pixels

This section briefly outlines techniques that have been employed in previous designs to provide wide-dynamic-range pixel operation. The logarithmic pixel, which is discussed first, differs slightly from the integrating pixels seen so far in that it operates in continuous time, while the remaining techniques address the saturation problem of integrating pixels mentioned above.

[3]However, to achieve constant bandwidth the power consumption must increase with C_{bus}.

2.5.1 Logarithmic Pixels

Logarithmic pixels employ a device with an exponential V-to-I relationship, typically a sub-threshold MOS transistor, to logarithmically-compress input photocurrents into the voltage domain [14]. A generic three-transistor logarithmic pixel is shown in Figure 2-8, with M_f and S forming an addressable source-follower voltage readout chain and M_{log} performing the I_{photo}-to-V_{pd} compression described by

$$V_{pd} = V_{DD} - V_{TS} - \frac{\phi_t}{\kappa} \ln \frac{I_{photo}}{I_s}. \tag{2.10}$$

In this equation, V_{TS} and I_s are the threshold voltage and current scale factor of M_{log}, respectively, ϕ_t is the thermal voltage, which is roughly 25.9 mV @ T = 300 K, and κ accounts for the limited gate control of the transistor surface potential and is typically around 0.7.

In practice, the generic logarithmic pixel exhibits several limitations that have been the focus of ongoing research [15–21]. First, the pixel output voltage swing is typically much smaller than the available range, leaving room for sensitivity improvements. For example,

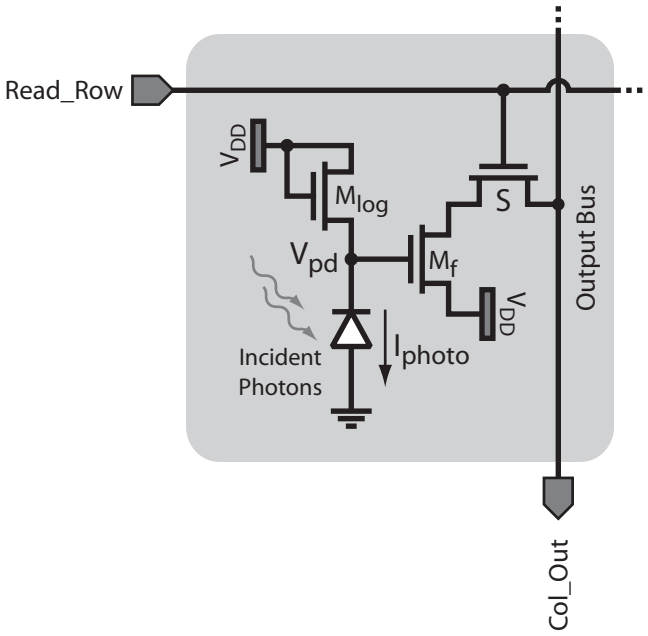

Figure 2-8: A three-transistor logarithmic pixel.

with $\kappa \approx 0.7$, the pixel compresses a five-decade change in I_{photo} into a V_{pd} swing of roughly 425 mV. Second, pixel-to-pixel mismatch in V_{TS} and I_s introduces relatively high levels of FPN, which reduces the overall array SNR. Third, the generic pixel response time is inversely proportional to input illumination, and under dim illumination can be slower than the desired frame rate.

2.5.2 Multiple-Saturation Pixels

Rather than allowing highly-illuminated pixels to quickly discharge over the maximum available swing, multiple-saturation pixels introduce one or more intermediate saturation levels that vary over the course of the available frame time [22–24]. To understand how this can improve pixel dynamic range, refer to the implementation of this technique illustrated in Figure 2-9. The basic pixel structure is identical to that of the generic three-transistor APS pixel from Figure 2-7. However, in this technique, rather than driving the gate of M_r to GND at $t_{res,end}$, V_{sat} drops to an intermediate voltage level, establishing an intermediate pixel saturation level. This modification does not affect the pixel response to low photocurrents, as illustrated by $V_{pd,2}$, but has a significant impact on the pixel's high-photocurrent response. The intermediate saturation level causes $V_{pd,1}$ to saturate at roughly a threshold voltage below V_{sat} at time $t_{sat,MS}$, and remain at this level until time t_f. At this point in time, V_{sat} undergoes a second transition, this time falling to GND, and $V_{pd,1}$ resumes discharging. When the pixel is read at t_{read}, $V_{pd,1}$ provides non-saturated voltage information about the pixel photocurrent. This represents a significant improvement compared with the response exhibited by the standard APS pixel to the same photocurrent, which as illustrated by the dashed light-grey line, saturates at time $t_{sat,APS}$. A potential drawback of this technique is that all prior information is lost at intermediate saturations, resulting in reduced charge collection in brightly-illuminated pixels and a reduction in the maximum pixel SNR [25].

2.5.3 Multiple-Sampling Pixels

Multiple-sampling designs address the problem of brightly-illuminated pixel saturation by enabling each pixel to select from a range of possible integration windows, depending on its local photocurrent level [25–39]. This technique has been implemented in several ways, but the basic principle can be best understood using the implementation illustrated in Figure 2-10. The pixel topology is again the same as the generic three-transistor APS pixel

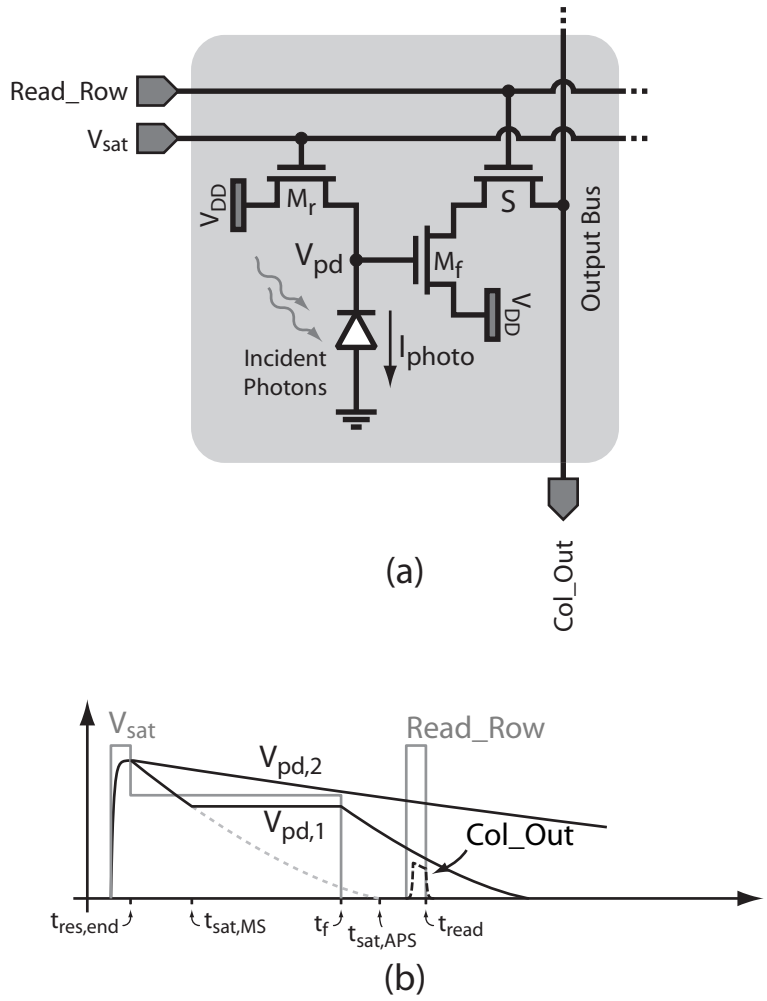

Figure 2-9: A three-transistor multiple-saturation pixel.

from Figure 2-7. Pixel operation is also identical to the standard APS pixel, except each pixel is read more than once during a frame, in this case twice, once at $t_{read,1}$ and once at $t_{read,2}$. This allows highly-illuminated pixels to provide output information at early read times before they saturate, as illustrated by $V_{pd,1}$, and dimly-illuminated pixels to provide information at later read times, as illustrated by $V_{pd,2}$. Since there is no way to know a priori which pixels will be subject to a particular illumination, all pixels are read at the pre-defined times, and the information provided during the last read before pixel saturation, or the end of the frame, is recorded. A potential drawback of this technique is illustrated by the response $V_{pd,1}$, where the last read before saturation occurs at $t_{read,1}$. Since the pixel has not collected the maximum possible charge by this point in time, the maximum SNR

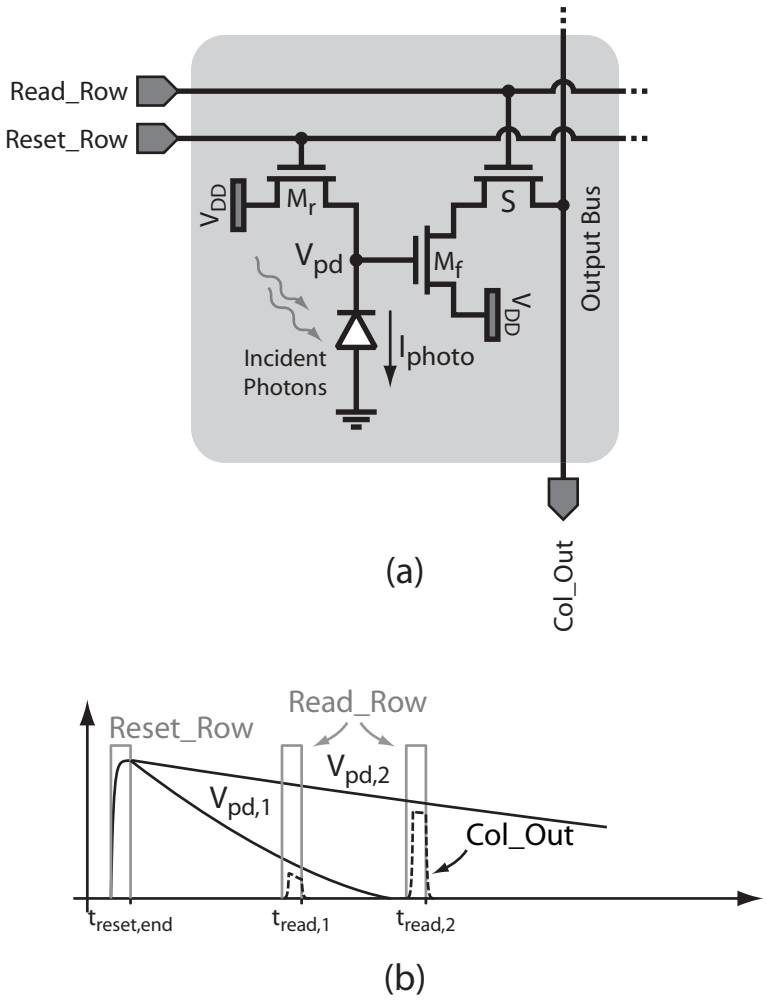

Figure 2-10: A three-transistor multiple-sampling pixel.

of its photocurrent estimate, as defined by Equation 2.8, will be less than optimal.

2.5.4 Time-Based Pixels

Another class of wide-dynamic-range imagers employs time-based pixels to encode photocurrent information. Since this approach forms the basis for the work presented in the next chapter, a detailed discussion of past time-based implementations is delayed until then.

2.6 Summary

This chapter introduced the basic features of the photon-to-e-h pair transduction process and discussed the methods that have been employed in past pixel designs to collect, measure, and convey the local e-h pair generation rates from the pixel to the periphery of the array. The limited dynamic range of CCD and APS pixels was shown to stem from their use of integration to overcome shot-noise limitations associated with measuring small photocurrents. Finally, several approaches that have been proposed to extend the dynamic range of standard CMOS APS pixels were discussed.

Chapter 3

Dual-Threshold Wide-Dynamic-Range Imaging Algorithm

Time-based pixels possess several intrinsic performance advantages over traditional integrating pixels in the realm of wide-dynamic-range image capture. However, to realize these advantages in practice, several potential drawbacks of the time-based approach must be addressed. This chapter begins by examining the operation of a generic time-based pixel and compares it with that of a traditional integrating pixel, in the process highlighting the pros and cons of the time-based approach. It then systematically addresses the problems encountered in past time-based pixel designs, in the process developing a novel dual-threshold synchronous capture algorithm that addresses many of these limitations. Finally, the theoretical dynamic range and SNR performance of the dual-threshold algorithm is analyzed in terms of the fundamental noise limitations faced by all solid-state imagers. This work sets the stage for the prototype imager implementation discussed in Chapter 4.

3.1 Generic Time-Based Pixel

A generic time-based pixel, along with a set of typical operating waveforms, is shown in Figure 3-1. The pixel is composed of a photodiode, comparator, reset transistor M_r, and memory element, in this case a K-bit register. As illustrated by the set of typical operating

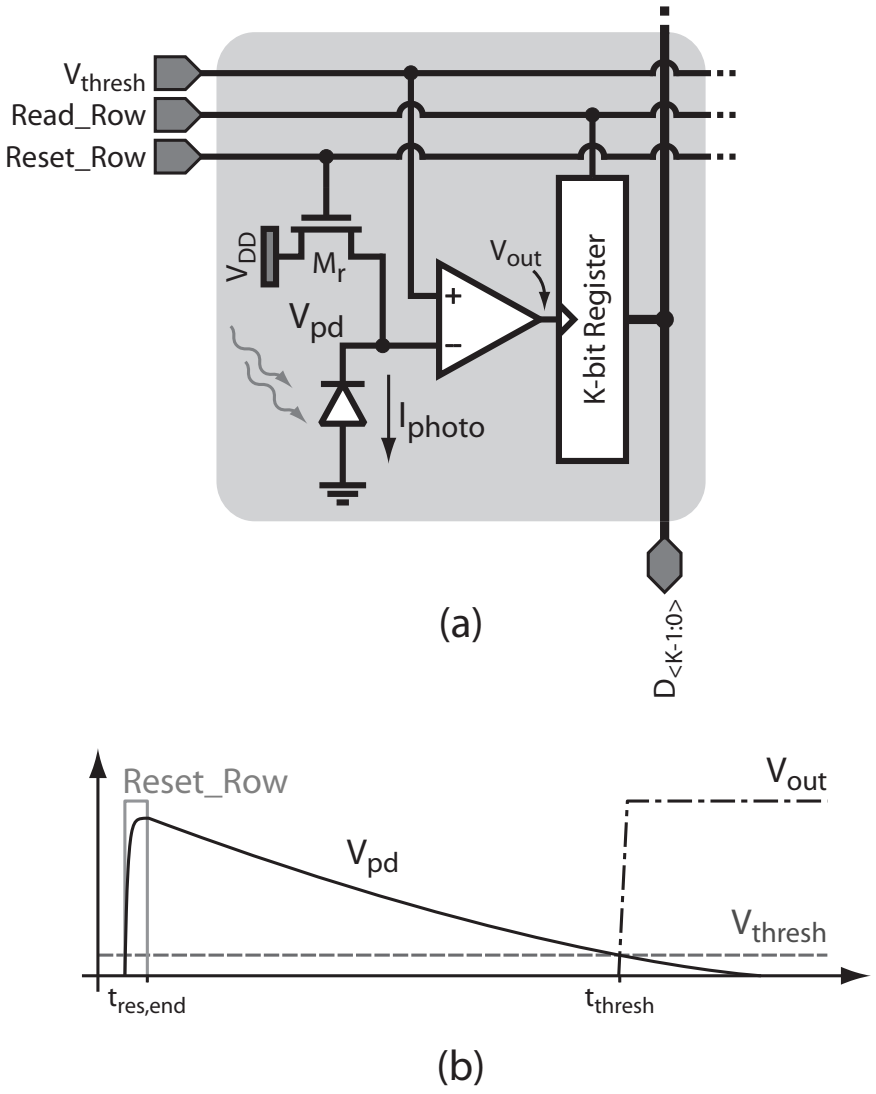

Figure 3-1: Generic time-based pixel (a) and operation (b).

waveforms shown in part (b) of the figure, each frame begins with a reset phase, during which `Reset_Row` pulses `HIGH`, turning M_r `ON` and resetting V_{pd} to roughly $V_{DD} - V_{th}$, a threshold voltage below V_{DD}. The reset phase ends when `Reset_Row` transitions `LOW` at $t_{res,end}$, allowing V_{pd} to begin discharging towards GND at a rate proportional to the photo-generated current I_{photo}. During this de-integration phase, the comparator monitors its inputs for the instant when V_{pd} crosses the constant reference voltage level V_{thresh}, at which point it outputs a `LOW`-to-`HIGH` transition on V_{out}. This transition causes the K-bit register to latch the current state of the data bus $\texttt{D}_{\texttt{<K-1:0>}}$, which is externally driven with

a non-repeating time-dependent digital sequence. Once the de-integration phase has ended, the external drive of $\texttt{D}_{\texttt{<K-1:0>}}$ is disabled and the pixel's latched K-bit code can be read out on the bus by pulsing `Read_Row HIGH` (not shown). The latched digital code can be used to infer I_{photo} with a precision that is proportional to the rate of change in the time-dependent digital sequence. Note that due to the non-linearity of the photodiode depletion capacitance, even assuming I_{photo} is constant over the entire de-integration interval, the decay in V_{pd} will be non-linear. However, since the pixel voltage swing is fixed between V_{DD} and V_{thresh} independent of I_{photo}, the pixel always reaches threshold when a constant charge Q_{const} has been de-integrated from the capacitance. Thus the mapping from I_{photo} to $\frac{1}{(t_{thresh}-t_{res,end})}$ remains perfectly linear in this scheme.

3.2 Advantages of Time-Based Pixels

To illustrate the differences between the generic time-based pixel and the generic active pixel sensor described in Section 2.4.3 and shown in Figure 2-7, consider the response of these two topologies to a range of photocurrents, as shown in Figure 3-2. For clarity, the column read voltages and comparator output voltages are only shown for the mid-range photocurrent $I_{photo,2}$, which both pixels sense correctly. Considering input levels above $I_{photo,2}$, clearly a range of photocurrents exists which, as illustrated by $I_{photo,1}$, the standard active pixel cannot sense due to V_{pd} saturating at GND before t_{read}, but which the time-based pixel correctly detects via earlier threshold crossings. Conversely, both pixels correctly detect inputs below $I_{photo,2}$ correctly, but the time-based pixel takes a significant amount of time to respond in this region, as illustrated by the fact that $V_{pd,3}$ does not cross V_{thresh} within the time limits of the figure. Focusing for the moment on the advantages offered by the time-based pixel, this example clearly illustrates two important ones:

- **Direct I → t Conversion Yields Wide Dynamic Range:** Time-based pixels use the limited-dynamic-range voltage domain only as an intermediate variable to transform information between the wide-dynamic-range current and time domains. As a result, they possess an intrinsic dynamic range that is typically several orders of magnitude larger than that of a standard active pixel. Furthermore, since technology scaling trends tend to simultaneously yield higher timing and lower voltage dynamic ranges, the time-based approach should scale well with process.

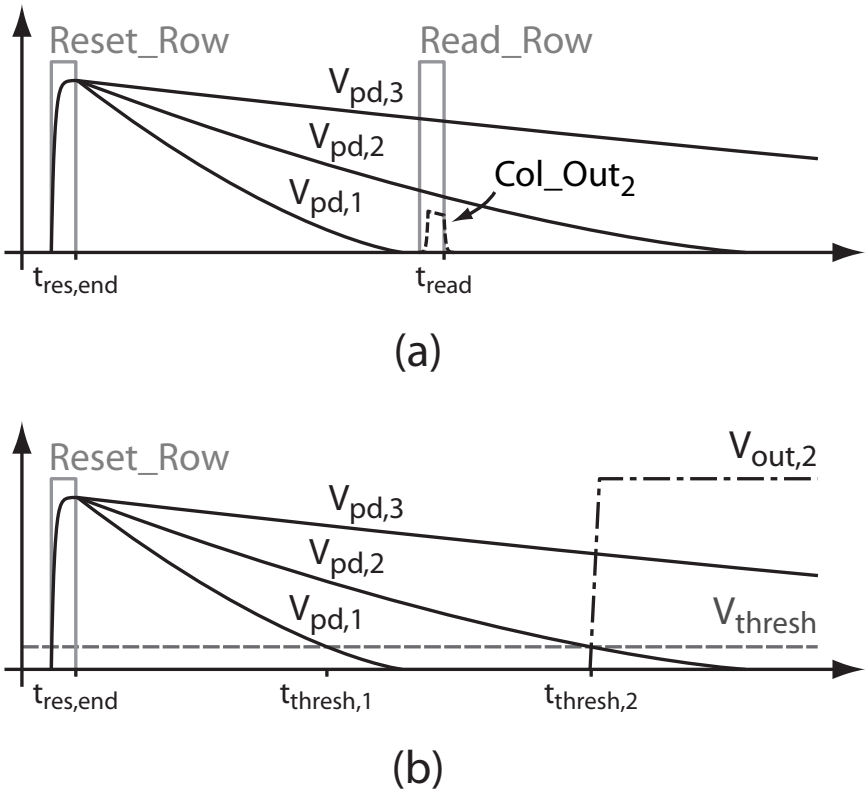

Figure 3-2: Comparison of generic active (a) and time-based (b) pixel responses to three photocurrent levels.

- **Maximum Photo-Generated Charge Collection:** Pixel SNR can be improved by either decreasing sensor noise, increasing photo-generated charge collection, or both. Since the time-based approach requires every pixel to de-integrate the same maximal number of charge quanta from V_{pd} in order to reach threshold, it guarantees the photo-generated signal term is maximized, independent of the photocurrent level.

3.3 Limitations of Past Time-Based Imagers

While the dynamic range and SNR benefits illustrated in the previous section make time-based pixels promising for wide-dynamic-range image capture, past hardware implementations based on this technique have revealed a variety of problems with this approach. The following subsections discuss these limitations along with possible solutions.

3.3.1 Poorly-Defined Frame Rate

As illustrated above in Figure 3-2, the generic active pixel provides updated information after a pre-defined de-integration interval of length ($t_{read} - t_{res,end}$), regardless of I_{photo}, while the generic time-based pixel's de-integration interval length ($t_{thresh} - t_{res,end}$) is inversely proportional to I_{photo}, yielding slow update rates for dimly-illuminated pixels. This poses a potential problem when employing the time-based approach in video-rate applications, which expect updated pixel information to be available at least every t_{frame} seconds, where $f_{frame} = \frac{1}{t_{frame}}$ is know as the frame rate and is typically around 30 Hz. Several classes of time-based pixels have been presented in the literature that exhibit limitations due to this effect, while others have recognized and circumvented this problem. Each of these approaches is discussed in the following sections.

Single-Slope Pixels

Except for variations in the specific comparator and memory topologies employed, the first class of pixels, which we will refer to as single-slope topologies, are implemented and operate in a manner similar to the generic time-based pixel of Figure 3-1 [40–46]. As should be expected based on the above discussion, the performance of these pixels suffers when video rate outputs are required from them. Specifically, based on the data provided in these references, requiring a 30 Hz frame rate from these designs drastically reduces their achieved dynamic range, in one case from 145 dB to 71 dB [40], due to the loss of low illumination information. It is interesting to note that among these designs, 71 dB is also the maximum dynamic range achieved with $f_{frame} = 30$ Hz, which is not significantly better than a standard active pixel sensor would achieve under the same conditions. To understand this intuitively from Figure 3-2, notice that under the fixed frame rate limitation the active pixel and generic time-based pixel capture mutually exclusive portions of the photocurrent dynamic range, with the active pixel saturating for input levels that cause the time-based pixel to reach threshold, and the time-based pixel not reaching threshold for input levels that the active pixel senses correctly. It turns out that for typical values of pixel frame rate, voltage swing, and timing precision, the dynamic range of these two photocurrent regions is comparable, implying that the dynamic range of the two pixels will be as well [47].

Frequency-Based Pixels

A second class of pixels, which we will refer to as frequency-based topologies, operate in a manner similar to the generic time-based pixel except that, upon reaching threshold, these pixels automatically self-reset and immediately begin a new de-integration phase [42, 48–52]. A generic implementation of such a pixel is shown in Figure 3-3, along with a set of typical operating waveforms illustrating that this topology functions as a photocurrent-controlled free-running oscillator. Assuming $f_{osc} = \frac{1}{t_{osc}} \propto I_{photo}$, which is an excellent approximation until t_{osc} approaches the same order of magnitude as the comparator delay,

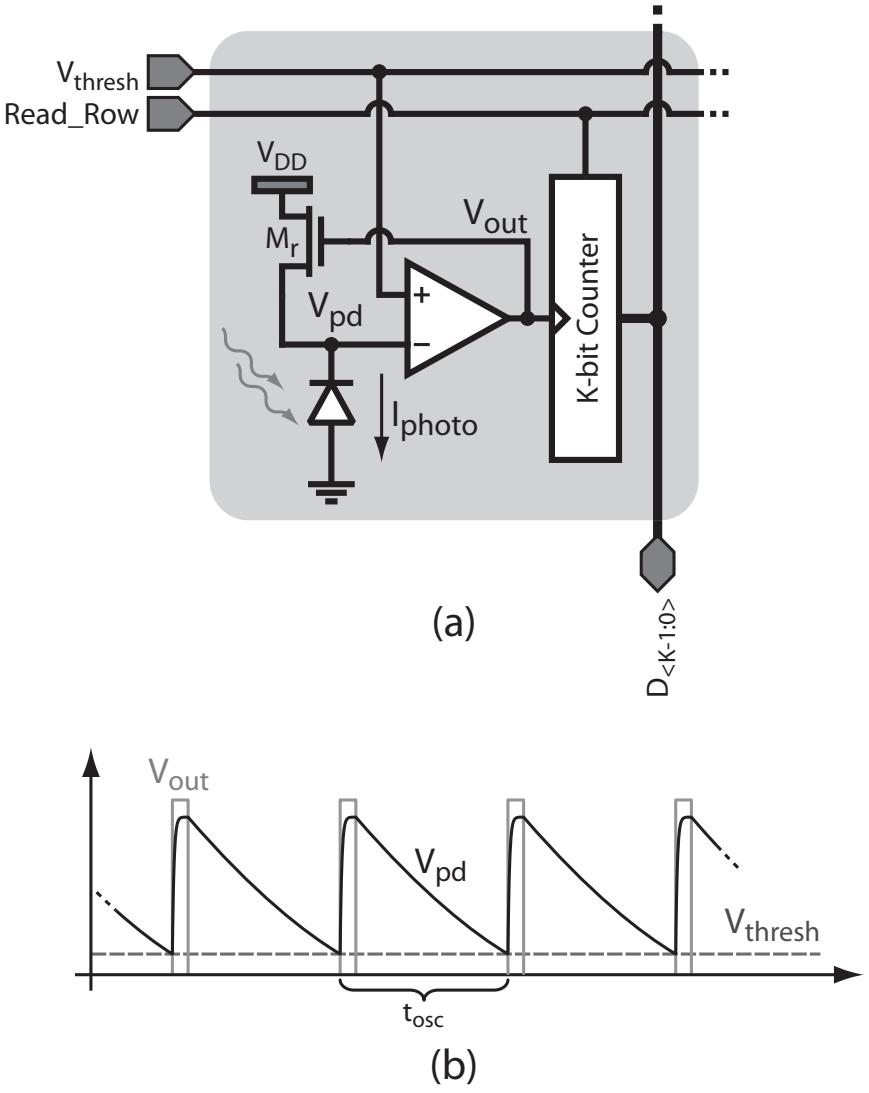

Figure 3-3: Generic frequency-based pixel (a) and operation (b).

the pixel photocurrent can be inferred by measuring the pulse frequency at V_{out}. Several implementations have chosen to estimate f_{osc} using a pixel-level K-bit counter to record the number of pulses generated within a fixed window of time [42, 49, 51], while other designs have chosen to transmit each pulse to external hardware, which either performs a similar windowed pulse count or measures the inter-spike interval times directly [48, 50, 52]. Eliminating the pixel-level counter improves fill factor, but requires low-latency timing information to be communicated between each pixel and the external hardware, a task has been most commonly performed using some form of address-event representation (AER), an asynchronous communication scheme described in [53].

Considering the use of frequency-based pixels in video-rate applications, notice that as a direct result of their ideal transfer characteristic $f_{osc} \propto I_{photo}$, frequency-based pixels exhibit slow update rates in response to low illumination inputs. However, unlike single-slope pixels, which are synchronously reset at the start of each new frame and thus never reach threshold under dim illumination, frequency-based pixels' self-resetting scheme allows them to always reach threshold, even under low illumination, and thus achieve better dynamic range than their single-slope counterparts. Unfortunately, since these same dimly-illuminated pixels exhibit $f_{osc} < f_{frame}$, their superior dynamic range is achieved at the expense of temporal resolution, making them equally unattractive for video applications [52].

Σ-Δ Pixels

A third class of closely-related pixels employ Σ-Δ modulation at either the pixel [54–57] or column [58, 59] level to quantize pixel photocurrent. The pixels implemented in [54–56] are similar to the frequency-based pixel of Figure 3-3, except the comparator is clocked and the reset transistor M_r is replaced with a charge-mode DAC. Thus, rather than asynchronously detecting threshold crossings and resetting V_{pd} to a known voltage, these pixels detect threshold crossings synchronously and trigger the DAC to inject a fixed amount of charge onto V_{pd}. A second approach retains the basic asynchronous oscillator topology shown in Figure 3-3, but oversamples the comparator output V_{out} to generate a 1^{st}-order Σ-Δ sequence [57]. Due to the similarity between this class of pixels and the frequency-based designs discussed above, it is not surprising that Σ-Δ pixels also exhibit poor temporal resolution under low illumination [57].

Single-Slope Varying-Threshold Pixels

As was recognized in [60], the low-illumination update rate of the generic time-based pixel is fundamentally limited by the fact that, independent of I_{photo}, the same fixed amount of charge must be de-integrated from V_{pd} in order for the pixel to reach threshold and update its stored information. Since the single-slope, frequency-based, and Σ-Δ pixels described above are all derivatives of the generic time-based pixel, their low-illumination limitations can all be traced to this same root cause. A potential solution to this problem is to allow V_{thresh} to vary over the course of the frame interval, thus modulating the amount of charge that must be de-integrated from V_{pd} in order to reach threshold during different portions of the frame [60]. For example, as illustrated in Figure 3-4, ramping V_{thresh} upward at a constant rate during the de-integration phase guarantees that the generic time-based pixel reaches threshold at or before a globally-defined time $t_{frame,end}$, regardless of the value of I_{photo} [60]. This technique demarcates a fourth class of time-based pixels, which

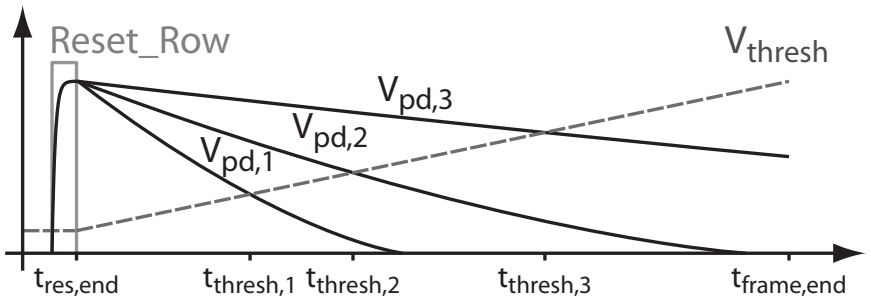

Figure 3-4: Response of generic time-based pixel with time-varying V_{thresh}.

we will refer to as single-slope varying-threshold topologies, that can provide expanded dynamic range without sacrificing temporal resolution, making them promising for video-rate imaging [61–67].

Since this single-slope varying-threshold technique will serve as the basis for the novel dual-threshold algorithm employed in this work, it is important to recognize the tradeoffs that are made in exchange for this improved temporal response. First, increasing V_{thresh} above its minimum value enables dimly-illuminated pixels to reach threshold faster by requiring them to collect less photo-generated charge than their brightly-illuminated counterparts. While this is the desired effect, the resulting reduction in photo-generated charge

collection reduces these dimly-illuminated pixels' SNR when compared with the generic time-based approach. Second, the simple mapping $I_{photo} \propto \frac{1}{(t_{thresh} - t_{res,end})}$ no longer holds when V_{thresh} is not fixed over the frame interval, an effect that must be accounted for when interpreting the resulting pixel threshold timing information.

Other Time-Based Pixels

In addition to the single-slope varying-threshold pixel introduced above, at least two other time-based pixel topologies capable of capturing wide-dynamic-range photocurrent data at a fixed frame rate have been presented in the literature. The first employs a hybrid of the single-slope and active-pixel techniques, using the former to capture timing information from highly-illuminated pixels and the latter to record voltage information from dimly-illuminated pixels at the end a fixed de-integration interval [68,69]. Compared to the single-slope varying-threshold pixel, the primary drawback of this hybrid pixel is the overhead associated with performing both time- and voltage-domain recording and readout.

The second approach employs an adaptive dual-slope algorithm to quantize pixel photocurrents [70], which begins with each pixel individually selecting a photocurrent integration time t_1 from a set of pre-defined windows of geometrically-increasing length. The pixel chooses t_1 as large as possible to maximize the charge it collects at V_{pd}, but short enough that V_{pd} does not saturate. The second phase of the dual-slope algorithm measures the time t_2 it takes for a known reference current I_{ref} to bring V_{pd} back to its original voltage level. Since the net change in V_{pd} across this dual-slope integration is zero, the charge integrated at this node during the two phases must be equal in magnitude, allowing I_{photo} to be expressed in terms of I_{ref} by plugging t_1 and t_2 into the relationship $|I_{photo}| \cdot t_1 = |I_{ref}| \cdot t_2$. The primary disadvantage of this approach is that it requires each pixel possess a precise copy of the global reference current I_{ref} in order to implement the second phase of the dual-slope algorithm, which is difficult to achieve in tight-pitch pixel arrays where area is at a premium.

3.3.2 Fixed-Pattern-Noise

Ideally, an imaging array exposed to a perfectly uniform source of illumination should generate identical data from every pixel. In reality, natural variations in parameters such as photodiode capacitance, dark current, and illuminance-to-charge conversion as well as

comparator offset voltage lead to fixed pixel-to-pixel response mismatch across the array. This mismatch is commonly referred to as fixed pattern noise (FPN) to distinguish it from temporal noise, which introduces frame-to-frame variance rather than fixed mismatch in the pixel response [13]. While many FPN effects can be reduced by increasing the unit area of each element that is to be matched, due to the limited area available in each pixel this can only help to a certain extent. The residual random mismatch, which is present in all imaging arrays, must be either accepted or counteracted with circuit, algorithmic, or external post-correction techniques.

Fortunately, one of the most significant sources of FPN in past time-based imagers has been due to comparator input-referred offset voltage [52, 57, 60], which can be significantly reduced by employing auto-zeroing during pixel reset [63, 66–68, 71]. To illustrate the advantages of this technique, the topology and response of a fixed voltage reset pixel and an auto-zeroing pixel are compared in Figure 3-5. In the pixel shown in part (a), V_{pd} is initialized to a known voltage, in this case V_{DD}, when `Reset` is `HIGH`, and I_{photo} begins discharging

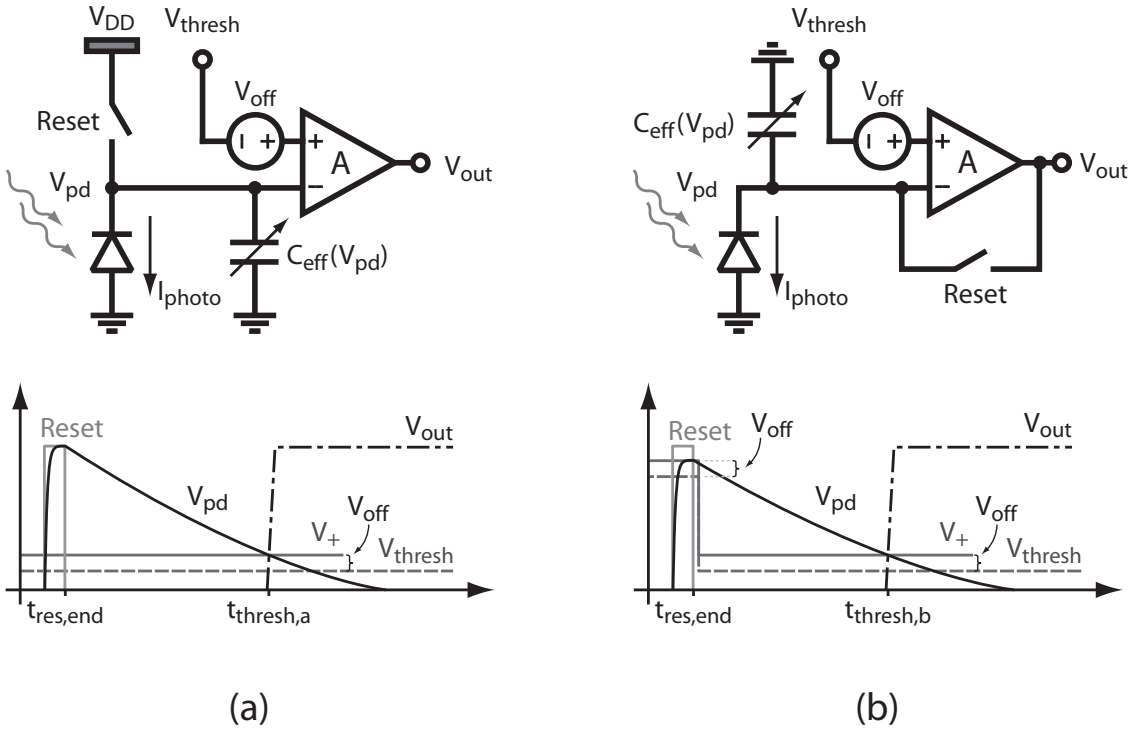

Figure 3-5: Comparison of fixed voltage reset (a) and auto-zeroed (b) pixel responses.

V_{pd} towards GND when the reset phase ends at $t_{res,end}$. During this discharge phase, the comparator monitors its inputs and drives V_{out} `HIGH` when $V_{pd} = V_+ = V_{thresh} + V_{off}$, where the static mismatch in the comparator's internal components is modeled by the input-referred offset voltage V_{off}. The photocurrent I_{photo} can then be estimated using

$$I_{photo} = -\frac{1}{t_{thresh} - t_{res,end}} \left[\int_{V_{pd}(t_{res,end})}^{V_{pd}(t_{thresh})} C_{eff}(V_{pd})\, dV_{pd} \right], \tag{3.1}$$

where $C_{eff}(V_{pd})$ is the net effective voltage-dependent parasitic capacitance between V_{pd} and GND, including the nonlinear photodiode capacitance $C_{pd}(V_{pd})$, and for now is assumed to be well-known. To an external observer, who has knowledge of V_{thresh}, $t_{res,end}$, and $t_{thresh,a}$, but not of V_{off}, the best estimate for I_{photo} is obtained by substituting $V_{pd}(t_{res,end}) = V_{DD}$, $V_{pd}(t_{thresh}) = V_{thresh}(t_{thresh,a}) + V_{off}$, and $V_{off} = 0$ V into this equation. This clearly results in an error in the upper limit of the integral by an unknown amount V_{off}.

Next, compare this result with that of the auto-zeroed pixel shown in part (b) of the figure. In this approach, when `Reset` is `HIGH`, the comparator is connected in unity-negative feedback, establishing the initial value $V_{pd} \approx V_{thresh} + V_{off}\left[\frac{A}{1+A}\right]$. Once the reset phase ends at $t_{res,end}$, I_{photo} again begins to discharge V_{pd} while the comparator monitors its inputs and drives V_{out} `HIGH` when $V_{pd} = V_+ = V_{thresh} + V_{off}$. Again armed only with knowledge of V_{thresh}, $t_{res,end}$, and $t_{thresh,b}$, the best estimate for I_{photo} is obtained by substituting $V_{pd}(t_{res,end}) = V_{thresh}(t_{res,end}) + V_{off}\left[\frac{A}{1+A}\right]$, $V_{pd}(t_{thresh}) = V_{thresh}(t_{thresh,b}) + V_{off}$, and $V_{off} = 0$ V into Equation 3.1. In this case, the upper and lower limits will be in error by V_{off} and $V_{off}\left[\frac{A}{1+A}\right]$, respectively, which are essentially equivalent as long as the amplifier gain A is large.

To see why this latter approach typically provides a more accurate estimate of I_{photo} than the former, suppose for a moment that $C_{eff}(V_{pd})$ was actually a constant capacitance C_{const}, independent of V_{pd}. Substituting $C_{eff}(V_{pd}) = C_{const}$, $V_{pd}(t_{res,end}) = V_{thresh}(t_{res,end}) + V_{off}\left[\frac{A}{1+A}\right]$ and $V_{pd}(t_{thresh}) = V_{thresh}(t_{thresh,b}) + V_{off}$ into Equation 3.1 yields

$$I_{photo} = -\frac{C_{const}}{t_{thresh,b} - t_{res,end}} \left[V_{thresh}(t_{thresh,b}) - V_{thresh}(t_{res,end}) + V_{off}\left(\frac{1}{1+A}\right) \right], \tag{3.2}$$

illustrating that when A is reasonably large the estimate for I_{photo} doesn't depend on V_{off}. In reality $C_{pd}(V_{pd})$ is not a constant, but its dependence on V_{pd} is usually sufficiently weak

that measuring I_{photo} using the auto-zeroed approach provides much of the benefit illustrated by Equation 3.2.

While auto-zeroing can significantly reduce the effect of static comparator offsets, residual FPN due to dynamic effects such as reset transistor charge injection mismatch and comparator latching mismatch cannot be removed using this technique [66]. These residual FPN terms can be modeled by a second offset voltage $V_{off,\overline{az}}$ in series with the first, which cannot be measured by the auto-zeroing operation but will add an offset to the measured threshold crossings. To counteract this residual FPN error, we propose employing a novel dual-threshold algorithm to capture pixel data. Similar to the single-slope varying-threshold pixel operation described earlier, the dual-threshold approach employs a single-slope de-integration interval to measure pixel photocurrent. However, unlike the monotonically ramping threshold voltage shown previously in Figure 3-4, the dual-threshold algorithm employs a V_{thresh} waveform with two separate threshold detection regions, as shown in Figure 3-6. The additional threshold detection region, implemented between $t_{res,end}$ and

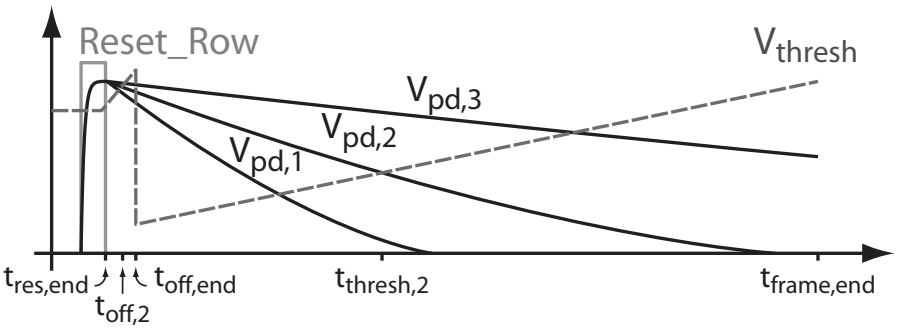

Figure 3-6: Response of generic time-based pixel to dual-threshold V_{thresh} waveform.

$t_{off,end}$, provides an initial pixel threshold crossing at time t_{off}, which varies with the pixel photocurrent I_{photo} and initial value $V_{pd}(t^{+}_{res,end})$. After the initial measurement phase ends at $t_{off,end}$, V_{thresh} returns to the original path it traced previously in Figure 3-4. As long as I_{photo} is not so large that it causes $V_{pd}(t_{off,end}) < V_{thresh}(t_{off,end})$, the pixel will reach threshold for a second time at t_{thresh}, just like it did in the single-threshold case. The average pixel photocurrent I_{photo} can then be estimated using the relation

$$I_{photo} = -\frac{1}{t_{thresh} - t_{off}} \left[\int_{V_{pd}(t_{off})}^{V_{pd}(t_{thresh})} C_{eff}(V_{pd})\, dV_{pd} \right], \tag{3.3}$$

along with $V_{pd}(t_{off}) = V_{thresh}(t_{off}) + V_{off}$ and $V_{pd}(t_{thresh}) = V_{thresh}(t_{thresh}) + V_{off}$. Since both the upper and lower integral limits were measured using threshold crossings, both $V_{thresh}(t_{off})$ and $V_{thresh}(t_{thresh})$ will include the unknown offset term $V_{off,\overline{az}}$. Therefore, based on the arguments presented above centered around Equation 3.2, a nonzero value of $V_{off,\overline{az}}$ won't introduce significant errors in the I_{photo} estimate. On the other hand, in the standard auto-zero approach, the value of the lower limit is assumed while the upper limit is measured. In this case, the lower limit will not include the offset $V_{off,\overline{az}}$ but the upper limit will, and nonzero values of this offset will degrade the estimate of I_{photo}.

3.3.3 Nonlinear Charge-to-Voltage Conversion

While most $\Sigma-\Delta$ converters [54–56,58,59] as well as dual-slope converters [70] are naturally immune to capacitive nonlinearity due to inherent properties of these algorithms, the same is not true of single-slope converters. This fact was observed in the previous section, where the voltage dependent capacitance $C_{eff}(V_{pd})$ was shown to lead to an undesirable nonlinearity in the voltage-to-charge inversion employed in Equations 3.1 and 3.3, complicating the calculation of I_{photo} and limiting the effectiveness of the auto-zeroing and dual-threshold FPN-reduction techniques. One solution to this problem that has been proposed in the literature is to add a linear charge integrator circuit between the pixel photodiode and comparator, but this technique requires the addition of an operational amplifier and linear capacitor to each pixel, at a significant cost in area [40]. The novel solution proposed here requires the addition of a single linear capacitor to each pixel to linearize the pixel charge-to-voltage conversion.

To illustrate how this additional capacitor can be used to linearize the pixel response, consider the two approaches to threshold detection illustrated in Figure 3-7. The approach shown in part (a) is the basic auto-zeroing technique, which was discussed in detail in the previous section. The approach shown in part (b), which includes the linear capacitor C_{coup} between nodes V_{pd} and V_{thresh}, also begins with an auto-zero phase when `Reset` is `HIGH`, but in this case the unity-negative feedback establishes an initial value $V_{pd} \approx V_{ref} + V_{off}\left[\frac{A}{1+A}\right]$ since the comparator reference level is now fixed at V_{ref}. Once the reset phase ends at $t_{res,end}$, I_{photo} begins to discharge V_{pd} while the comparator monitors its inputs for a threshold condition. Due to the addition of C_{coup}, movements in V_{thresh}

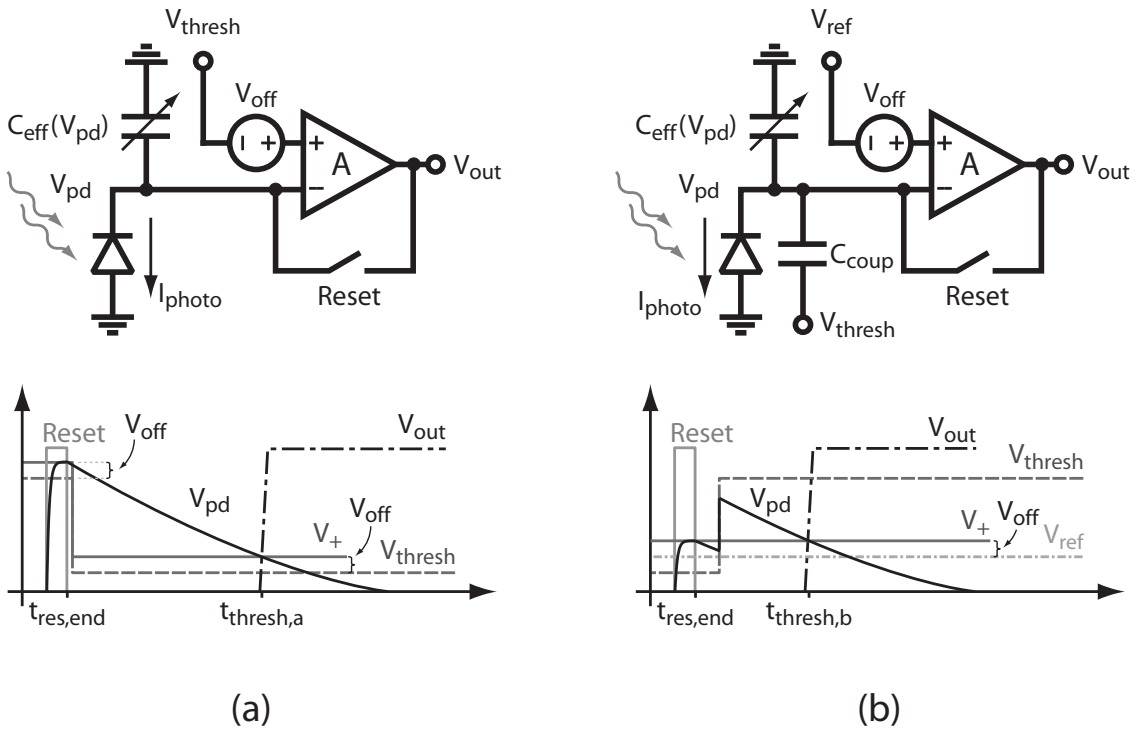

Figure 3-7: Two approaches to comparator-based threshold detection. (a) Adjustable threshold level is applied directly to the comparator. (b) Adjustable threshold level is capacitively coupled to the integrating node. Note that the comparators are assumed to be ideal for simplicity.

are now capacitively coupled into V_{pd}[1], modulating the amount of charge that I_{photo} must de-integrate from V_{pd} before the comparator finally detects a threshold condition at time $t_{thresh,b}$, where again $V_{pd} = V_+ = V_{ref} + V_{off}$.

While these two techniques accomplish similar goals, the latter one offers several important advantages, all stemming from the fact that, independent of the size of the movement in V_{thresh}, the comparator threshold level remains fixed at $V_{ref} + V_{off}$. This leads to several simplifications in the pixel comparator design, as discussed in Section 4.1.1, but the primary advantage we are concerned with here has to do with the effective voltage-to-charge conversion of this approach. Since the technique forces $V_{pd}(t_{res,end}) = V_{pd}(t_{thresh,b})$, considering only these two voltage boundary conditions, it is clear that across the time interval from $t_{res,end}$ to $t_{thresh,b}$ no net charge is sunk from or sourced to V_{pd} by $C_{eff}(V_{pd})$. Furthermore, again only considering these boundary conditions, it is clear that across this same interval

[1]Note the opposite polarity of movements in V_{thresh} in the two cases.

the net charge coupled into V_{pd} by C_{coup} is governed by

$$\Delta Q_{coup} = C_{coup} \cdot [V_{thresh}(t_{thresh,b}) - V_{thresh}(t_{res,end})]. \tag{3.4}$$

Finally, since the net voltage change at V_{pd} across this time interval is zero, the amount of charge sunk by I_{photo} must be equivalent to the charge sourced by C_{coup}. Equating these two terms yields

$$I_{photo} = \frac{C_{coup}}{t_{thresh,b} - t_{res,end}}[V_{thresh}(t_{thresh,b}) - V_{thresh}(t_{res,end})], \tag{3.5}$$

which is the linear mapping we desire. If C_{coup} is implemented using a linear, well-matched capacitor such as a metal-insulator-metal (MIM) or poly-to-poly (P2P) structure, a single global voltage waveform can be used to precisely control the amount of charge each pixel must de-integrate to reach threshold. The primary cost paid for this linearization is that the coupling capacitor increases the total parasitic capacitance at V_{pd}, which as will be seen in Section 3.4.2, can negatively impact pixel noise performance. As a final note, the dual-threshold technique can be applied equally well to this capacitively-coupled pixel to reduce residual FPN terms, and yields the following generalized relationship

$$I_{photo} = \frac{C_{coup}}{t_{thresh} - t_{off}}[V_{thresh}(t_{thresh}) - V_{thresh}(t_{off})], \tag{3.6}$$

which will serve as the basis for photocurrent calculations in the remainder of this work.

3.3.4 Time-Domain Noise

Since time-based pixels encode photocurrent information in the time-domain, variability in the latency between the occurrence of a threshold crossing and the recording of timing data will degrade the overall noise performance of the imager. One group of past time-based pixel implementations have minimized this effect by incorporating either digital or analog memory within the pixel to record timing information, similar to the generic time-based pixel in Figure 3-1 [40–46,49,51,60,62,68–70]. The major drawback of this approach is that it sacrifices pixel fill factor, which does not exceed 23% in any of these references, even at a pixel pitch of 45μ m×45 μm [42]. A second group of implementations have chosen to not record any timing information on the imager chip, and instead employ some variation of

the address-event representation (AER) communication scheme [50,52,61,63,66,67]. In this approach, when a pixel detects a threshold crossing event it asynchronously informs a global arbiter, whose job is to coordinate the transmission of these events, along with the addresses of the pixels that generated them, to an off-chip entity via a shared communication bus. Only when an event is received by the off-chip entity is timing information recorded for the pixel. The major drawback of this approach is its potential to exhibit large, variable communication latency when exposed to certain degenerate scenes. In particular, many AER-based imagers perform poorly when exposed to scenes containing large regions of uniform illumination, which result in many pixels reaching threshold and attempting to transmit events on the shared communication bus at the same time. One analysis predicts more than 44% average photocurrent measurement errors across a 720×480 AER-based array exposed to a largely uniformly-illuminated scene [67]. It is worth noting that a rolling-shutter solution proposed in this same reference is able to reduce these errors to an acceptable level for this uniformly-illuminated scene. However, this simply shifts the latency problem onto a different degenerate scene, due to the fundamental inability of the AER approach to communicate many simultaneous events with low latency.

The approach taken in this work represents a compromise between these two extremes. The digital memory used to record pixel timing information is integrated on the same die as the pixels, but is removed from the pixel layout and placed in a separate, parallel memory array, as shown in the simplified representation in Figure 3-8(a). This improves pixel fill factor significantly, while leveraging off the high on-chip communication speeds of modern processes to keep the pixel-to-memory latency low. Additionally, rather than asynchronously transmit threshold crossing events between the pixels and memory, which as discussed above introduces scene-dependent latencies, a synchronous time-domain multiplexed communication strategy is employed. In this approach, each row of pixels is allotted small time slices during which it has guaranteed access to a shared column bus and can transmit pixel threshold information in parallel to the memory array. These time slices occur on a pre-defined schedule, which is designed to guarantee the resulting pixel-to-memory latency variance remains below the inherent noise floor of the pixel timing information. From the pixel and memory cell's perspective, it is as if there is a virtual switch between them, which can close only during brief, pre-determined instants in time to transmit the current state of the pixel V_{out} signal to the memory cell. A discussion of how the locations

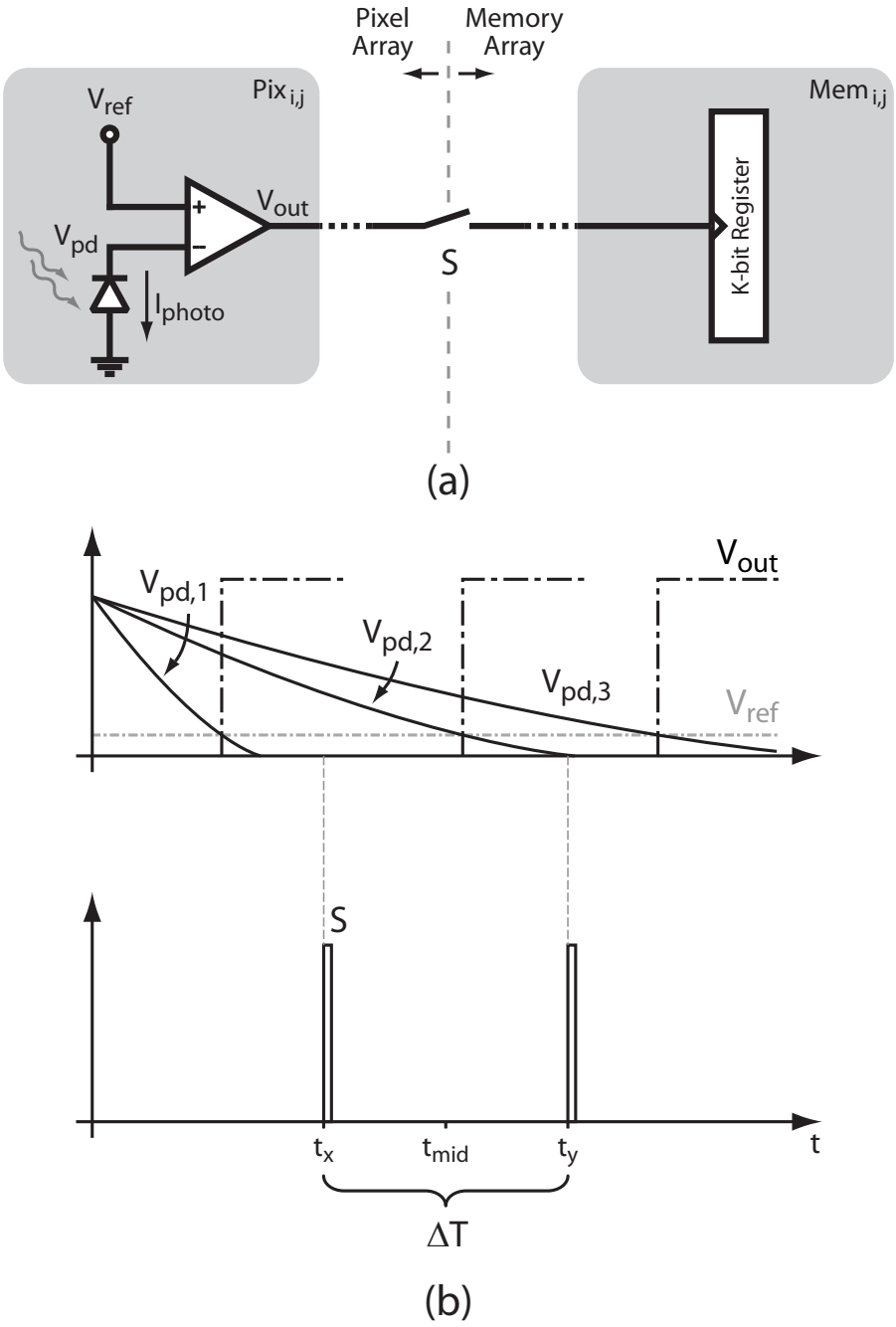

Figure 3-8: Pixels and memory are implemented in two separate arrays (a) to improve fill factor, and communication occurs in synchrony with S, introducing bounded timing errors (b).

of these time slices are chosen can be found in Section 3.4.4, while details of the hardware implementation of this time-domain multiplexing strategy can be found in Chapter 4.

An important consequence of this synchronous communication scheme is that it guaran-

Case	$V_{out}(t_x)$	$V_{out}(t_y)$	Comment
1	1	1	Threshold crossing before t_x
2	0	1	Threshold crossing between t_x & t_y
3	0	0	No threshold crossing before t_y
-	1	0	Assumed to be impossible

Table 3.1: Truth table for pixel-to-memory communication example.

tees the pixel-to-memory communication latency is bounded, with a pre-determined maximum timing variance. To make this statement clearer, refer to Figure 3-8(b), which depicts the locations of two pre-defined pixel-to-memory communication time slices at times t_x and t_y, along with the response of the simplified pixel to three different photocurrent levels. Since the pixel comparator is assumed ideal, V_{out} will be either `LOW` or `HIGH` at t_x and t_y, leading to the four possible combinations of state shown in Table 3.1. Each of the three allowed cases is illustrated by an associated V_{pd} waveform in the figure, and is interpreted in the comment column.

For the pair of communication times t_x and t_y, the most important case is number two, which corresponds to a threshold crossing between these two times. Labeling $I_{photo,x}$ and $I_{photo,y}$ as the levels that would cause the pixel to cross threshold exactly at t_x and t_y, respectively, any I_{photo} within the range $I_{photo,y} < I_{photo} < I_{photo,x}$ will generate a threshold crossing within the window $t_x < t < t_y$. Since this range of photocurrents cannot be further distinguished based on the information available to the memory element, it simply records a digital code corresponding to the expected value of the threshold crossing time t_{mid}, resulting in a quantization of the pixel information. Under the standard assumption that the true threshold crossing could have occurred at any time between t_x and t_y with uniform probability, the resulting timing variance introduced by this quantization step is given by

$$\sigma_t^2 = \sigma_{(t-t_{mid})}^2 = \frac{1}{\Delta T}\left[\int_{-\Delta T/2}^{\Delta T/2} t^2 dt\right] = \frac{1}{\Delta T}\left[\frac{t^3}{3}\Big|_{-\Delta T/2}^{\Delta T/2}\right] = \frac{\Delta T^2}{12}, \tag{3.7}$$

where $\Delta T = t_y - t_x$, as shown in Figure 3-8. Note that this result is independent of the value t_{mid}, and depends only on the spacing ΔT between successive transmission times in the synchronous pixel-to-memory communication scheme. Thus, as long as ΔT is chosen small enough to ensure that σ_t^2 is significantly less than the timing uncertainty introduced by the other noise sources present in the pixel, the noise introduced by this timing quantization

will not significantly reduce the SNR of the photocurrent measurement.

The basic time-domain quantization noise analysis presented above will be expanded in Section 3.4.3, where the entire video frame will be subdivided into a discrete set of time intervals. The analysis presented there is slightly more general, in that it simultaneously accounts for both timing and voltage quantization noise terms, allowing movements in the dual-threshold waveform V_{thresh} to be included. Together, the movements in V_{thresh} along with the discrete set of pixel-to-memory communication times will define the quantization noise of the imager over its entire photocurrent detection range.

3.3.5 Asynchronous vs. Synchronous Threshold Detection

Among the various time-based pixel implementations discussed in Section 3.3.1, both asynchronous [40, 41, 43–46, 48–52, 57, 60–63, 66–69] and synchronous [42, 54–56, 58, 59, 64, 65, 70] threshold detection schemes have been employed. Unfortunately, often little or no justification has been given as to why one approach was chosen over the other. As demonstrated by the implementation in [57], the use of a synchronous pixel-to-periphery communication strategy does not automatically imply that the threshold detection must also occur synchronously. However, as this section demonstrates, detecting threshold crossings synchronously should result in lower pixel power consumption in this particular application.

Begin by considering Figure 3-9, which contains a simplified pixel schematic along with two sets of waveforms illustrating pixel operation during both asynchronous (a) and synchronous (b) threshold detection. In the asynchronous approach, the comparator is biased with a constant current $I_{bias,a}$ and exhibits a delay of width t_{delay} between the time its inputs cross threshold and the time its output V_{out} crosses V_{IH} – the minimum voltage where it is guaranteed to be recognized as a digital `HIGH` level. Such a delay is present in all real comparators, but its magnitude can be decreased by increasing the comparator bandwidth. In employing the synchronous pixel-to-memory communication strategy discussed in the previous section to sample the output of this asynchronous pixel, this delay can be effectively counteracted by delaying the sampling of V_{out} by a similar amount, for example shifting the ideal sample time of t_x to $t_x + t_{delay}$. In the synchronous approach to threshold detection shown in part (b) of the figure, rather than biasing the comparator at a constant current level $I_{bias,a}$, it is synchronously turned on at a time t_{setup} before each ideal sampling time, where $t_{setup} < t_{delay}$ is an additional interval that allows startup transients to die

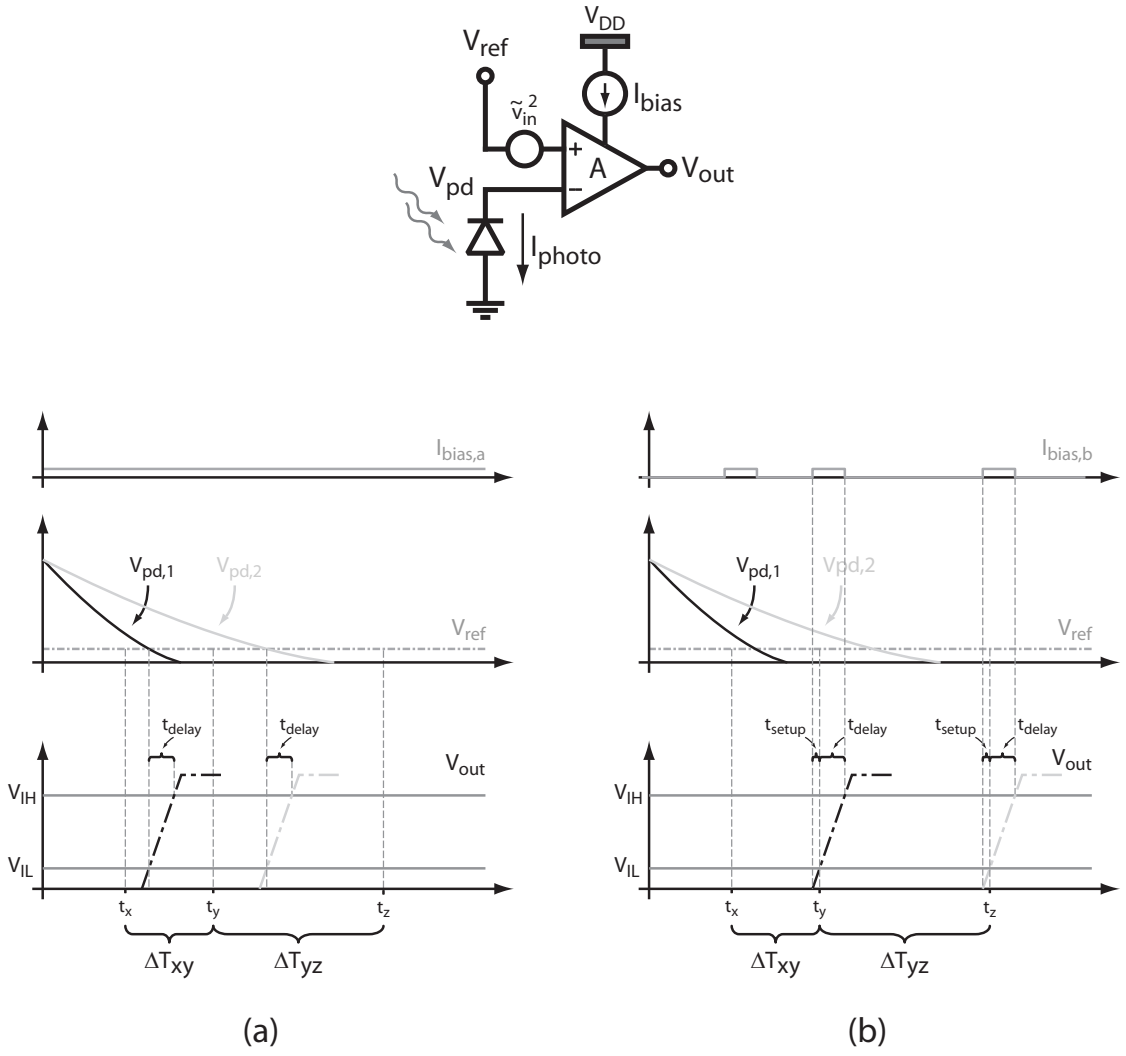

Figure 3-9: Simplified asynchronous pixel schematic (a) and operating waveforms illustrating comparator delay (b).

away. The comparator is left on for a time $t_{delay} + t_{setup}$, at which point its output status V_{out} is sampled and transmitted to the memory array, and it is then turned off.

Now that the differences in operation are clear, compare the energy expended by these two schemes over the two intervals ΔT_{xy} and ΔT_{yz} shown in the figure. In the asynchronous case, while t_{delay} does smear the energy consumption across the ideal threshold detection boundaries, this smearing is roughly equivalent at both the start and end of each interval, allowing the energy consumed in an interval ΔT to be given by $E_{asynch} = V_{DD} \cdot I_{bias,a} \cdot \Delta T$. In the synchronous case, each comparison simultaneously provides information for the end of one interval and the start of the next, allowing the energy consumed to be divided

between the two. This factor of two is counteracted by the fact that comparisons occur at both the start and end of each interval, giving an average energy consumption $E_{synch} = V_{DD} \cdot I_{bias,a} \cdot (t_{delay} + t_{setup})$ of one comparison per interval. Note that implicit in this equation is the assumption that $(t_{delay} + t_{setup}) < \Delta T$, otherwise the energy consumption for the synchronous and asynchronous approaches will be the same.

Though this analysis demonstrates that $E_{synch} \leq E_{asynch}$, it does not prove that $E_{synch} < E_{asynch}$. To arrive at this latter inequality, begin by noting that because the sample times $t_x + t_{delay}$, $t_y + t_{delay}$, etc., are defined globally for the entire pixel array, unless t_{delay} is a constant across every pixel, the method employed for counteracting delay will exhibit some residual timing uncertainty. More explicitly, modeling the delay as a random variable with average $\overline{t_{delay}}$ and variance $\sigma^2_{t_{delay}}$, the technique employed will negate the average delay, but will not counteract the effect of the delay variance. Since this mismatch is the result of parametric differences between comparator parasitics, its standard deviation $\sigma_{t_{delay}}$ should scale roughly in proportion to t_{delay}, i.e., $\sigma_{t_{delay}} \approx \gamma \cdot t_{delay}$ where γ is a constant representing the relative mismatch across the array. At this point, recall that the timing variance introduced by an inter-sample spacing of ΔT is given by $\frac{(\Delta T)^2}{12}$. Assuming the maximum additional timing variance allowed due to $\sigma^2_{t_{delay}}$ can be no greater than this value[2], we arrive at the relation

$$(\gamma \cdot t_{delay})^2 \leq \frac{\Delta T^2}{12}. \tag{3.8}$$

Taking as an example an array characterized by a 10% delay mismatch ($\gamma = 0.1$), a fairly generous estimate, the maximum allowable delay that will keep the variance below the desired bound is given by $t_{delay} \leq 2.89 \cdot \Delta T$. Returning to the two threshold detection strategies, since this bound must hold for all intervals, it will be limited by the smallest interval ΔT_{xy}, leading to the requirement that $t_{delay} \leq 2.89 \cdot \Delta T_{xy}$. If the maximum possible delay $t_{delay} = 2.89 \cdot \Delta T_{xy}$ is used as the design goal for the asynchronous comparator, the relationship $E_{asynch} \approx E_{synch}$ will approximately hold as long as the longest interval $\Delta T_{yz} \leq 2.89 \cdot \Delta T_{xy}$. However, as will be seen in Section 3.4.4 the ratio $\frac{\Delta T_{yz}}{\Delta T_{xy}}$ can exceed 230× using the proposed algorithm, leading to $E_{synch} < E_{asynch}$ over these intervals and for the array as a whole.

[2]Usually it will be required to be significantly less.

3.3.6 Power Consumption and Fill Factor

Many of the suggested algorithmic and hardware modifications presented in the previous sections have cited improvements in pixel fill factor and/or power consumption as potential benefits. Still, in order to implement the resulting synchronous dual-threshold algorithm, each pixel must contain a local comparator, linear capacitor, and threshold communication circuitry, in addition to the obligatory photodiode. Together, these first three components consume a large fraction of the pixel's area and power budgets, making their design a critical step in optimizing the performance of the overall imaging array. A discussion of their implementations, along with other hardware details, is presented in Chapter 4.

3.4 Synchronous Dual-Threshold Algorithm

The previous section outlined the major limitations encountered in past time-based imager implementations, and began to explore how these limitations can be addressed through symbiotic changes at the algorithmic and hardware level. In this section, the synchronous dual-threshold algorithm that evolved during this discussion is formalized, and its theoretical performance limits are analyzed.

3.4.1 Pixel Topology and Algorithm Overview

Compiling the algorithmic and hardware modifications introduced in Section 3.3, a high-level view of the proposed synchronous dual-threshold pixel and its associated operating waveforms can be generated, as shown in Figure 3-10. The pixel topology shown in part (a) is composed of a synchronous auto-zeroing comparator, linear coupling capacitor, photodiode, and pixel-to-memory communication block, and includes a parasitic voltage-dependent capacitance between V_{pd} and GND. Part (b) of the figure illustrates a typical set of pixel waveforms over a single frame capture. The frame begins with an auto-zeroing reset phase, which establishes the initial condition $V_{pd}(t_{res,end}) = V_{ref}$ on the photodiode cathode and a low voltage level $V_{thresh}(t_{res,end})$ on the opposite terminal of C_{coup}. The first threshold crossing phase begins at $t_{res,end}$, during which V_{thresh} makes an initial upward step and then sweeps downward through a range of voltages over the interval from $t_{res,end}$ to $t_{off,end}$. The size of the initial step and the voltage sweep are chosen to ensure that all pixels will detect a threshold crossing during this phase, while the rate at which the comparator and pixel-

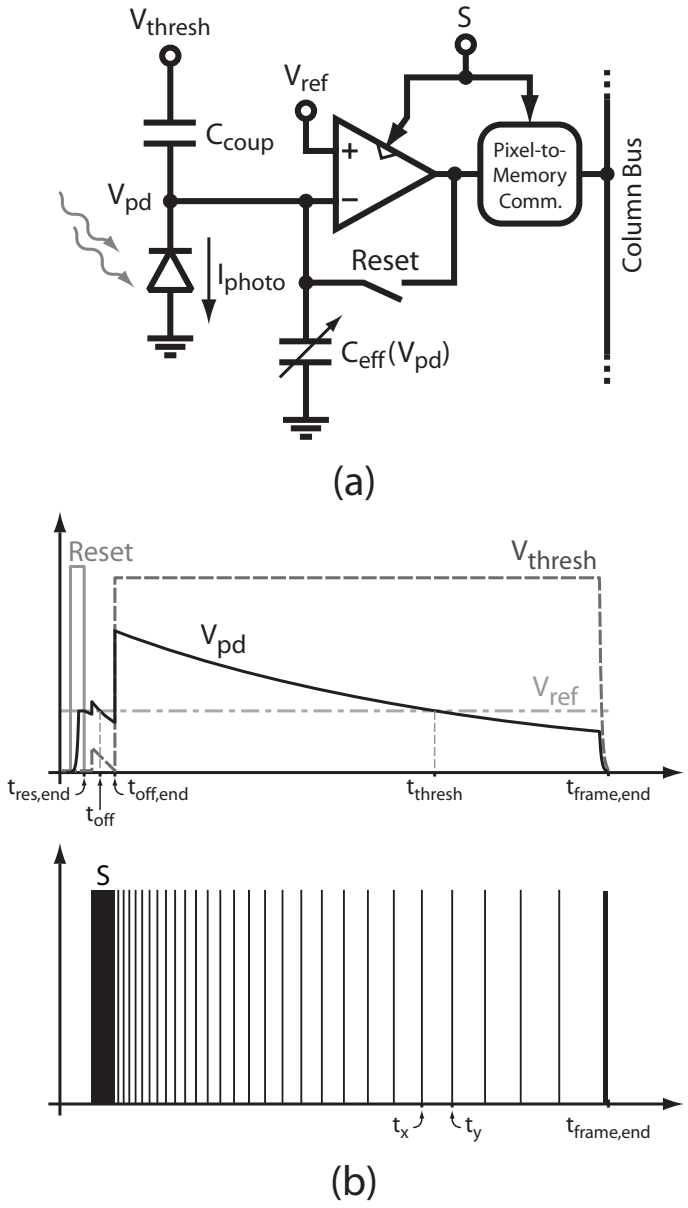

Figure 3-10: Dual-threshold pixel topology (a) and representative operating waveforms (b). Note the length of the reset and offset phases have been exaggerated to illustrate details within this time interval.

to-memory communication block are triggered by `S` is as high as possible. This allows the voltage sweep to occur in the minimum possible time while simultaneously minimizing the quantization noise incurred in measuring t_{off}, the first threshold crossing time. The second threshold crossing phase begins at $t_{off,end}$, where V_{thresh} steps up to its maximum level, coupling the maximum possible amount of charge into V_{pd}. For much of the remaining frame

time V_{thresh} remains fixed at this peak voltage while the comparator and pixel-to-memory communication block are repeatedly triggered by `S` at a rate that exhibits a slowing trend over the course of the frame. In this particular example, the photocurrent is large enough to cause V_{pd} to cross V_{ref} during this flat-top region of V_{thresh}. Based on the polling interval dictated by `S` in this region, the pixel-to-memory communication block will inform the pixel's associated memory element, via the column bus, that the second threshold crossing t_{thresh} occurred at some time between t_x and t_y, leading to a second quantization error. Finally, in the case where the photocurrent is not large enough to force the second threshold crossing to occur during the flat-top region, V_{thresh} follows a steep downward descent while the comparator and communication circuits are again rapidly triggered by `S`, forcing the second threshold crossing to occur at some time during the descent.

Since this algorithm ensures that $V_{pd}(t_{off}) = V_{pd}(t_{thresh}) = V_{ref}$, the net charge sourced or sunk by the parasitic voltage-dependent capacitance $C_{eff}(V_{pd})$ between t_{off} and t_{thresh} is guaranteed to be zero. Additionally, since the net change in V_{pd} over this time interval is zero, the net charge capacitively coupled into this node by C_{coup} due to movements in V_{thresh} must be equal and opposite the charge sunk from the node by I_{photo}, yielding the relation

$$I_{photo} = \frac{C_{coup}}{t_{thresh} - t_{off}}[V_{thresh}(t_{thresh}) - V_{thresh}(t_{off})]. \tag{3.9}$$

This is the same relation expressed in Equation 3.6, repeated here for convenience. Finally, as discussed previously in Section 3.3.2 this algorithm is relatively immune to static comparator offsets due to its use of auto-zeroing, as well as some dynamic offsets such as reset transistor charge injection and comparator latching mismatch, due to its dual-threshold recording strategy.

The remainder of this section explores the various attributes of the synchronous dual-threshold pixel and algorithm outlined above in more detail. Namely, the impact of intrinsic noise sources on pixel performance is analyzed, a method of choosing the pixel-to-memory communication times in waveform `S` is developed based on quantization noise considerations, and the rationale for the general shape of the V_{thresh} waveform is presented based on both quantization noise as well as SNR considerations.

3.4.2 Intrinsic Noise Analysis

This section analyzes the dominant intrinsic noise sources present in the synchronous dual-threshold pixel, and shows how each of them impacts the pixel SNR and dynamic range. Since these noise terms arise due to many different effects across the pixel, it is important to choose a common node, as well as signal format, to refer them to. The most common choice, which is adopted here, is to express each in terms of electrons at node V_{pd}, allowing the results to be directly compared with the integrated photocurrent charge in its native format.

Photocurrent Signal and Shot Noise

Deriving the number of photo-generated electrons collected by the pixel due to an input I_{photo} is complicated by the fact that the answer depends on the exact shape of the V_{thresh} waveform. This is demonstrated by the relationship presented in Equation 3.9, which states that the amount of charge collected during the integration is a function of the values of V_{thresh} at times t_{off} and t_{thresh}. Since at this point we have not explicitly defined this waveform, it may seem pointless to proceed with the current task. Fortunately, several approximations can be made, based solely on the general shape of V_{thresh} shown in Figure 3-10, that yield results within a few percent accuracy. This is deemed reasonable in trade for the insight gained from the simplified form of the resulting equations, and can always be corrected for in numerical simulations once V_{thresh} has been established.

To visualize a method of solving this problem, consider the graphical representation of Equation 3.9 shown in Figure 3-11. This figure interprets the equation as the result of defining an effective charge threshold waveform $Q_{thresh} = C_{coup} \cdot V_{thresh}$ and looking for the points in time where the total integrated charge due to I_{photo} intersects it. The benefit of this viewpoint is that it offers an equivalent but conceptually simpler interpretation of the dual-threshold algorithm, since it avoids the nonlinearities present in the V_{pd} waveform shown in Figure 3-10.

Based on this new interpretation of the dual-threshold algorithm, the total photo-generated charge is now given by $Q_{photo} = Q_{thresh}(t_{thresh}) - Q_{thresh}(t_{off})$, which still requires V_{thresh} to be explicitly defined in order to yield a solution. Instead, the approach taken here will be to approximate the Q_{thresh} waveform using the ideal waveform $Q_{thresh,ideal}$ shown in

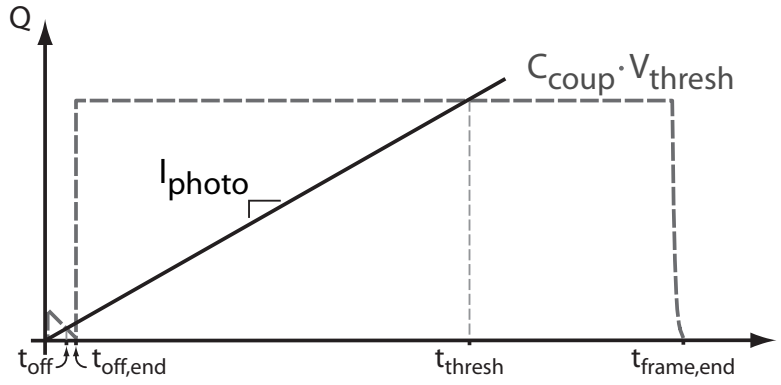

Figure 3-11: Graphical interpretation of dual-threshold algorithm as charge threshold operation.

Figure 3-12, and use this waveform to solve for Q_{photo} as a function of I_{photo}. The removal of the initial threshold crossing phase as well as the brick-wall response at time $t_{frame,end}$ results in a slight overestimate of the total charge collected by an amount $\Delta Q_1 + \Delta Q_2$, as shown in the figure. However, the error introduced by this approximation is usually on the order of a few percent, primarily due to the fact that the in designing the V_{thresh} waveform, the goal is to lie as close to $V_{thresh,ideal}$ as possible at every point in time (see Section 3.4.4). The utility of this approximation is that we can immediately see that the

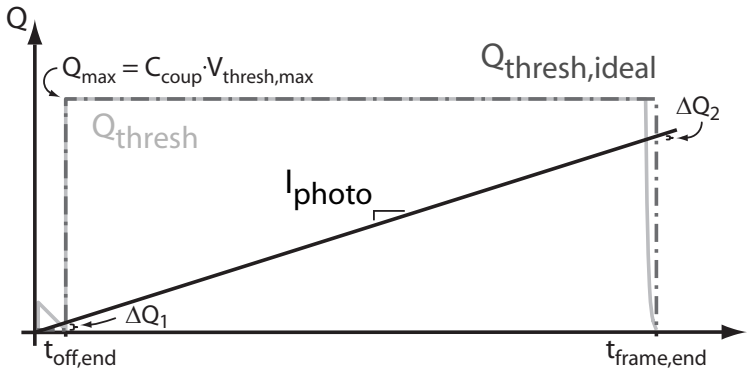

Figure 3-12: Ideal charge threshold waveform used for noise analysis.

number of electrons collected for a given input photocurrent is defined by

$$e^-_{photo,sig}(I_{photo}) = \begin{cases} \frac{C_{coup} \cdot V_{thresh,max}}{q} & \text{if } I_{photo} \geq \frac{C_{coup} \cdot V_{thresh,max}}{t_{frame,end}}, \\ \frac{I_{photo} \cdot t_{frame,end}}{q} & \text{if } I_{photo} < \frac{C_{coup} \cdot V_{thresh,max}}{t_{frame,end}}. \end{cases} \tag{3.10}$$

We can also determine that the maximum non-saturating photocurrent level is defined by

$$I_{photo,max} = \frac{C_{coup} \cdot V_{thresh,max}}{t_{off,end}}, \tag{3.11}$$

which demonstrates why it is important for the offset-measurement phase of the algorithm be as short as possible. Finally, since the photo-generated shot noise electron variance is equivalent to the number of charges collected (see Section 2.3), we can define

$$\sigma^2_{e^-_{photo,shot}}(I_{photo}) = e^-_{photo,sig}(I_{photo}). \tag{3.12}$$

Dark Current Shot Noise and Fixed-Pattern Noise

Even under zero incident illumination all real photodiodes exhibit some level of average reverse-bias leakage current I_{dark} due to thermal carrier generation. Since the value of I_{photo} modulates the effective pixel integration time $t_{int} = t_{thresh} - t_{off}$, the amount of charge integrated at V_{pd} due to I_{dark} will be input-dependent. Using the same ideal charge threshold waveform $Q_{thresh,ideal}$ as in the previous section, this integration time can be approximated to be

$$t_{int}(I_{photo}) = \begin{cases} \frac{C_{coup} \cdot V_{thresh,max}}{I_{photo}} & \text{if } I_{photo} \geq \frac{C_{coup} \cdot V_{thresh,max}}{t_{frame,end}}, \\ t_{frame,end} & \text{if } I_{photo} < \frac{C_{coup} \cdot V_{thresh,max}}{t_{frame,end}}. \end{cases} \tag{3.13}$$

The number of electrons integrated at a given pixel due to I_{dark} can then be represented as a function of both I_{dark} and I_{photo} using

$$e^-_{dark,avg}(I_{dark}, I_{photo}) = I_{dark} \cdot t_{int}(I_{photo}) \tag{3.14}$$

Employing a set of reference dark pixels which are shielded from input illumination, the average value of I_{dark} can be estimated for the pixel array, allowing the average number of

electrons $e^-_{dark,avg}(I_{dark}, I_{photo})$ collected by each pixel in the array to be calculated and subtracted. However, similar to the photocurrent, I_{dark} exhibits a shot noise electron variance given by

$$\sigma^2_{e^-_{dark,shot}}(I_{dark}, I_{photo}) = e^-_{dark,avg}(I_{dark}, I_{photo}), \tag{3.15}$$

which cannot be counteracted due to its random nature. Additionally, the average value of I_{dark} varies across the array, and is best approximated as a random variable with a variance $\sigma^2_{I_{dark}}$, leading to a dark-current fixed-pattern noise with an electron variance of

$$\sigma^2_{e^-_{dark,FPN}}(I_{dark}, I_{photo}) = \sigma^2_{I_{dark}} \cdot \left[\frac{t_{int}(I_{photo})}{q}\right]^2. \tag{3.16}$$

The only way to eliminate this FPN term would be to periodically measure and store the average dark current of each pixel in the array, and individually subtract these measured values from each image frame. This requires the use of a mechanical shutter to periodically zero the array illumination, and is not implemented in this work.

It is instructive to view the behavior of these dark current shot and fixed-pattern noise terms, along with the photocurrent signal and shot noise term, graphically as a function of I_{photo}. Since the ultimate goal is to gain insight into the pixel signal-to-noise ratio and dynamic range, it makes the most sense to plot the power of each of these terms, in $electrons^2$, as is done in Figure 3-13. The figure caption lists the representative set of pixel parameters used to generate this plot. The main features to notice in this plot are the following:

- At the lower end of the photocurrent range, the photo-generated signal power grows at 20 dB/dec, while the photo-generated shot noise grows at only 10 dB/dec. This is due to the fact that the total collected photo-generated charge grows in proportion to I_{photo} in the lower region of the Q_{thresh} characteristic (see Figure 3-12), resulting in proportional growth in the shot noise variance and squared growth in the signal power. Since the effective integration time remains fixed at its maximum level $t_{frame,end}$ over this range of inputs, the dark current shot and fixed-pattern noise terms remain constant.

- The knee observed in all of the signals just above 0.3 pA is due to the total integrated photo-charge intersecting the flat-top region of Q_{thresh} for the first time (see Figure 3-

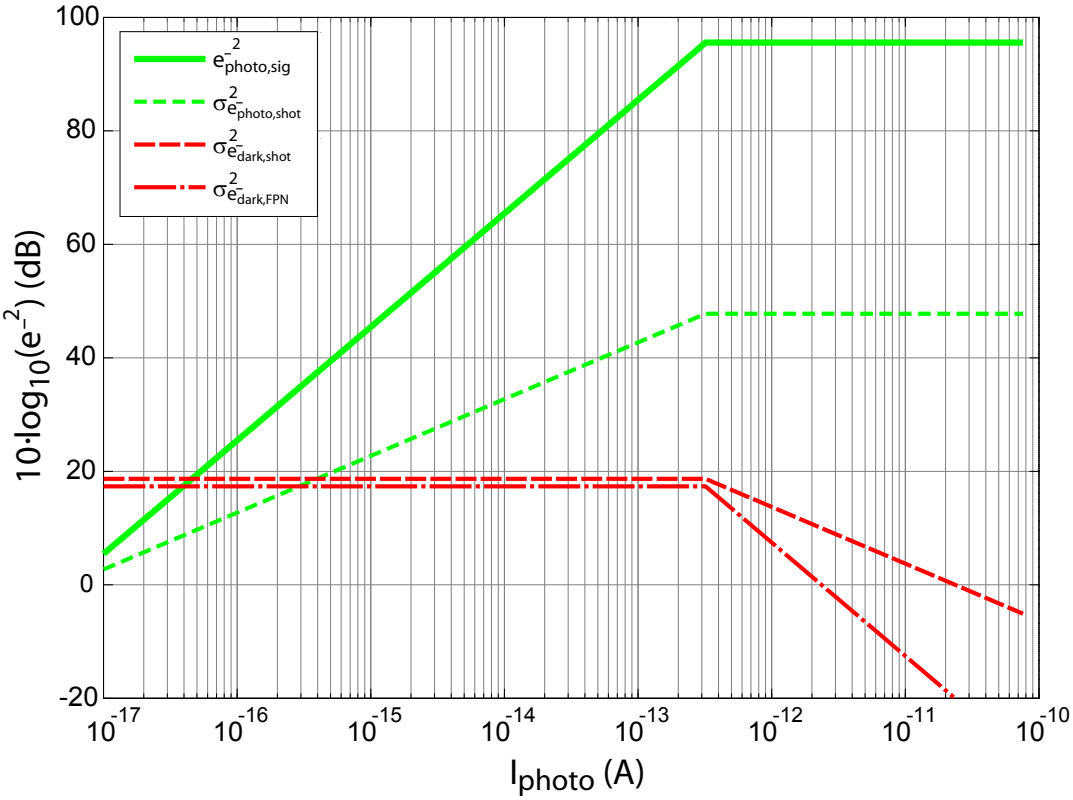

Figure 3-13: Plot of photo-generated signal and shot noise power, and dark current shot noise and FPN power as a function of photocurrent. Equations 3.10 – 3.16 were used to generate these plots, along with the following parameters: $t_{off,end} = 128$ μs, $t_{frame,end} = 30$ ms, $C_{coup} = 8$ fF, $V_{thresh,max} = 1.2$ V, $I_{dark} = 0.4$ fA, and $\sigma^2_{I_{dark}} = [(0.1) \cdot I_{dark}]^2$.

12). *This knee also marks the location where the standard active pixel sensor with the same parameters would saturate.* Past this input level, the amount of charge collected before the pixel reaches t_{thresh} remains fixed, thus so does the power and variance of this charge. Since $t_{int} \propto \frac{1}{I_{photo}}$ in this flat-top region, the dark current shot noise and fixed-pattern noise decrease at -10 dB/dec and -20 dB/dec in this region, respectively.

- All of the waveforms stop abruptly at 75 pA due to the fact that this is the maximum input level that can be sensed before the pixel saturates, as can be calculated by substituting the given parameters into Equation 3.11.

Offset Fixed-Pattern Noise

Offset fixed-pattern noise appears as non-uniformities in the response of an imaging array exposed to uniform illumination due to pixel- and/or column-level circuit offsets. The most significant source of offset in past time-based imagers has been attributed to comparator input-referred offset voltage [52, 57, 60], which, as discussed in detail in Section 3.3.2, can be significantly reduced by employing auto-zeroing during pixel reset [63, 66–68, 71]. Additional sources of offset FPN due to finite comparator gain, reset transistor charge injection mismatch, and comparator latching mismatch, were also mentioned in that section. These residual terms, which cannot be removed using auto-zeroing, prompted the introduction of the dual-threshold algorithm, which effectively counteracts them. As a result, errors due to offset FPN should be negligible in the dual-threshold pixel.

Reset Noise

During the pixel auto-zero phase, the unity-negative feedback loop around the comparator establishes an initial value at the photodiode cathode of $V_{pd} \approx V_{ref} + V_{off}\left[\frac{A}{1+A}\right]$, where V_{off} is the static input-referred comparator offset voltage, and A is the comparator open-loop gain. In addition to being driven to this DC level, the node is also subject to random voltage fluctuations due to both thermal and 1/f noise in the amplifier, reset transistor, V_{thresh} generator, and photodiode parasitic resistance. When the auto-zero phase ends at time $t_{res,end}$, both the DC level and the instantaneous value of the noise fluctuation are sampled onto V_{pd}, resulting in a random offset voltage that is fixed for the remainder of the current frame, but that varies from frame-to-frame. Typically, due to the large bandwidths involved during the auto-zero phase, the reset noise is dominated by the thermal components, which generate an electron variance at V_{pd} that is governed by

$$\sigma^2_{e^-_{thermal,reset}} = \frac{N \cdot kT \cdot [C_{coup} + C_{eff}(V_{ref})]}{q^2}, \tag{3.17}$$

where N is a scaling factor that accounts for the effective number of noise generators present in the circuit, and the voltage dependent capacitance $C_{eff}(V_{pd})$ is evaluated at the average DC level V_{ref}. Fortunately, though the magnitude of this noise can be significant, since it manifests as a sampled offset that is fixed between the times $t_{res,end}$ and $t_{frame,end}$ of a given frame, it will be counteracted by the dual-threshold algorithm. Thus, errors due to

this effect should be negligible in the dual-threshold pixel.

Read Noise

The thermal and 1/f noise sources mentioned above also affect the pixel during the threshold crossing phases of operation by temporally perturbing the effective comparator threshold level around its desired average value. The comparator, V_{thresh} generator, and photodiode parasitic resistance are the primary contributors to this noise term. To account for these terms, their individual power-spectral densities are integrated over the pixel's effective bandwidth, which is usually limited by the comparator, and are then combined and referred to the comparator's input terminals to provide an effective voltage variance of $\sigma^2_{v,read}$. This in turn can be cast as an electron variance at V_{pd} using the relation

$$\sigma^2_{e^-_{read}} \approx 2 \cdot \sigma^2_{v,read} \cdot \left[\frac{C_{coup} + C_{eff}(V_{ref})}{q}\right]^2, \tag{3.18}$$

where the factor of two accounts for the fact that the dual-threshold algorithm employs two threshold crossings, between which the read noise should be uncorrelated. This equation ends up slightly overestimating the read noise due to the fact that the low-frequency portions of the 1/f noise are in fact correlated between the two threshold crossings, and are partially canceled by the algorithm for the same reason that reset noise and offset FPN noise are.

Gain Fixed-Pattern Noise

The final major intrinsic noise contribution in the time-based pixel is due to gain fixed-pattern noise. Similar to offset FPN, this term also results in non-uniformities in the response of the array when it is exposed to uniform illumination, but is due to gain rather than offset mismatch between the pixels. To understand the roots of this FPN effect, consider the manner in which the pixel response given in Equation 3.9 depends on the value of C_{coup}. Random variations in this capacitance from pixel-to-pixel introduces a random scaling, or gain, factor in the response of each pixel. A similar gain mismatch can also arise due to pixel-to-pixel variation in the illuminance-to-photocurrent conversion described by Equation 2.2. The net gain error can be modeled by a single random variable $\alpha_{gain,FPN}$, which scales the response of each pixel, and exhibits a mean and variance across the pixel array of $\overline{\alpha_{gain,FPN}} = 1$ and $\sigma^2_{\alpha_{gain,FPN}}$, respectively. Since the number of photo-generated

electrons collected by each pixel is interpreted using the expected value of $\overline{\alpha_{gain,FPN}} = 1$, the measured versus true number of electrons at V_{pd} will in error by the pixel's particular value of the scaling random variable $\alpha_{gain,FPN}$. This pixel-to-pixel mismatch can be modeled as a variance in the number of electrons at node V_{pd} defined by

$$\sigma^2_{e^-_{gain,FPN}} = \sigma^2_{\alpha_{gain,FPN}} \cdot [e^-_{photo,sig}(I_{photo})]^2. \tag{3.19}$$

Similar to offset FPN, gain FPN can in theory also be counteracted by measuring and storing the gain error of each pixel, and multiplying each image frame by this reference frame. This is usually not implemented, and is not employed in this work.

Pixel Signal-to-Noise Ratio and Dynamic Range

By adding the read noise and gain fixed-pattern noise terms to the plot shown in Figure 3-13, we arrive at the overall pixel signal and noise power dependencies shown in Figure 3-14. As predicted by Equations 3.18 and 3.19, the pixel read noise is constant, independent of I_{photo}, while the gain FPN grows as a fixed fraction of the signal power. Additionally, the total noise power, which is calculated by assuming all of the displayed noise powers are uncorrelated and therefore can be summed directly, is shown in this plot. The utility of this plot becomes evident if we consider the equations for the pixel SNR and dynamic range, which are given by

$$\text{Signal-to-Noise Ratio} = 10 \cdot \log_{10}\left[\frac{\text{Signal power}}{\text{Total noise power}}\right], \tag{3.20}$$

and

$$\text{Dynamic Range} = 20 \cdot \log_{10}\left[\frac{\text{Maximum non-saturating input}}{\text{Minimum detectable signal}}\right], \tag{3.21}$$

Since $log[\frac{a}{b}] = log(a) - log(b)$, the pixel SNR can be interpreted graphically as the vertical distance between the signal and total noise power curves shown in the figure. Further, we can immediately see that the signal and total noise powers are equal for an input level of roughly $I_{photo} = 0.12$ fA, and that the maximum non-saturating input is roughly $I_{photo} = 75$ pA, putting the dynamic range of this pixel at around 116dB. The plots reveal other useful information as well, such as the fact that attempting to reduce the photodiode dark current fixed-pattern noise will not significantly improve our dynamic range, as it is primarily limited

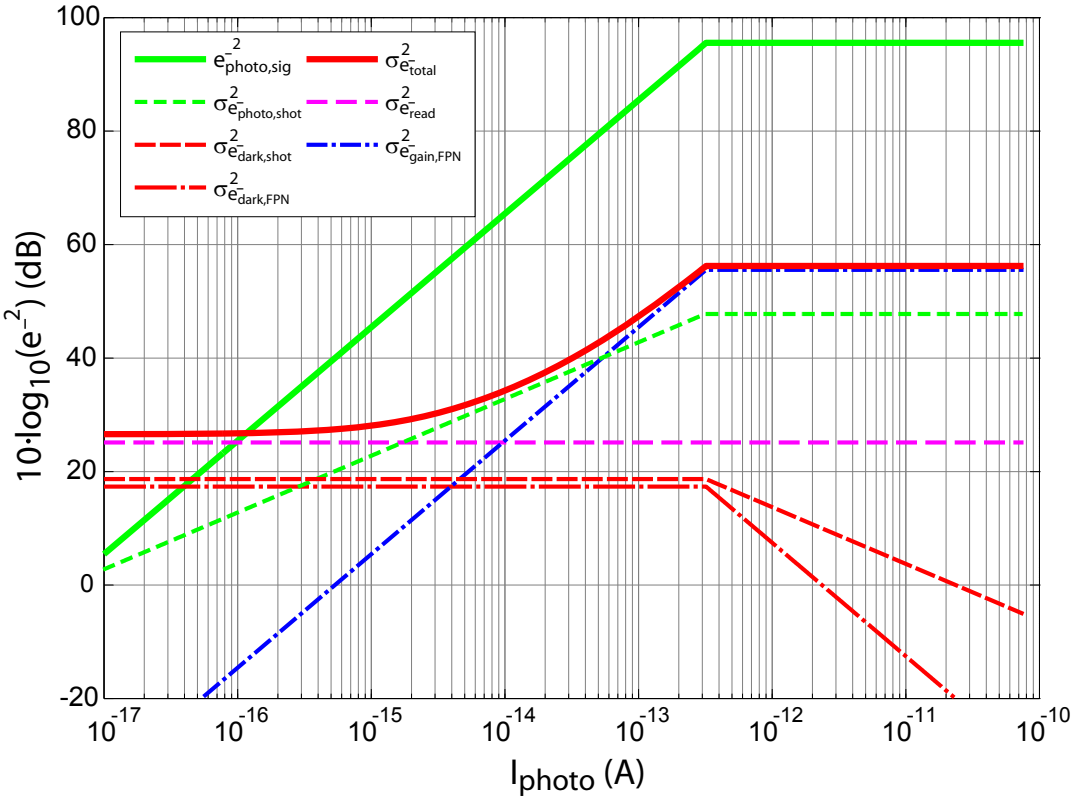

Figure 3-14: Plot of all dominant pixel signal and noise terms. Equations 3.10 – 3.19 were used to generate these plots, along with the following parameters: $t_{off,end} = 128$ μs, $t_{frame,end} = 30$ ms, $C_{coup} = 8$ fF, $V_{thresh,max} = 1.2$ V, $I_{dark} = 0.4$ fA, $\sigma^2_{I_{dark}} = [(0.1) \cdot I_{dark}]^2$, $V_{ref} = 0.85$ V, $C_{eff}(V_{ref}) = 8.45$ fF, $\sigma^2_{v,read} = 15 \times 10^{-9}$ V^2, and $\sigma^2_{\alpha_{gain,FPN}} = (0.01)^2$.

by the read noise at low photocurrents. Also notable is the fact that, since the gain FPN grows as a fixed fraction of the signal power, the maximum SNR we can hope to achieve out of the pixel is limited to

$$\text{SNR}_{\max} = 10 \cdot log_{10}\left[\frac{[e^-_{photo,sig}(I_{photo})]^2}{\sigma^2_{e^-_{gain,FPN}}}\right] = 10 \cdot log_{10}\left[\frac{1}{\sigma^2_{\alpha_{gain,FPN}}}\right]. \tag{3.22}$$

This shows that with 1% gain mismatch variance, the SNR can be no greater than 40 dB, which agrees well with the maximum SNR observed in Figure 3-14.

3.4.3 Quantization Noise Analysis

The goal of this section is to develop a set of bounds that must be met by V_{thresh} and the pixel-to-memory communication timing waveform `S`, in order to guarantee that the resulting quantization noise remains below the intrinsic pixel noise floor derived above. To accomplish this goal, the charge threshold waveform Q_{thresh} that was previously discussed in Section 3.4.2 will be analyzed over its three different regions of operation.

Descending Edge of Q_{thresh}

The charge threshold waveform Q_{thresh} is shown in Figure 3-15 along with a line of integrating charge resulting from a low-level photocurrent input I_{photo}. Focusing on the zoomed

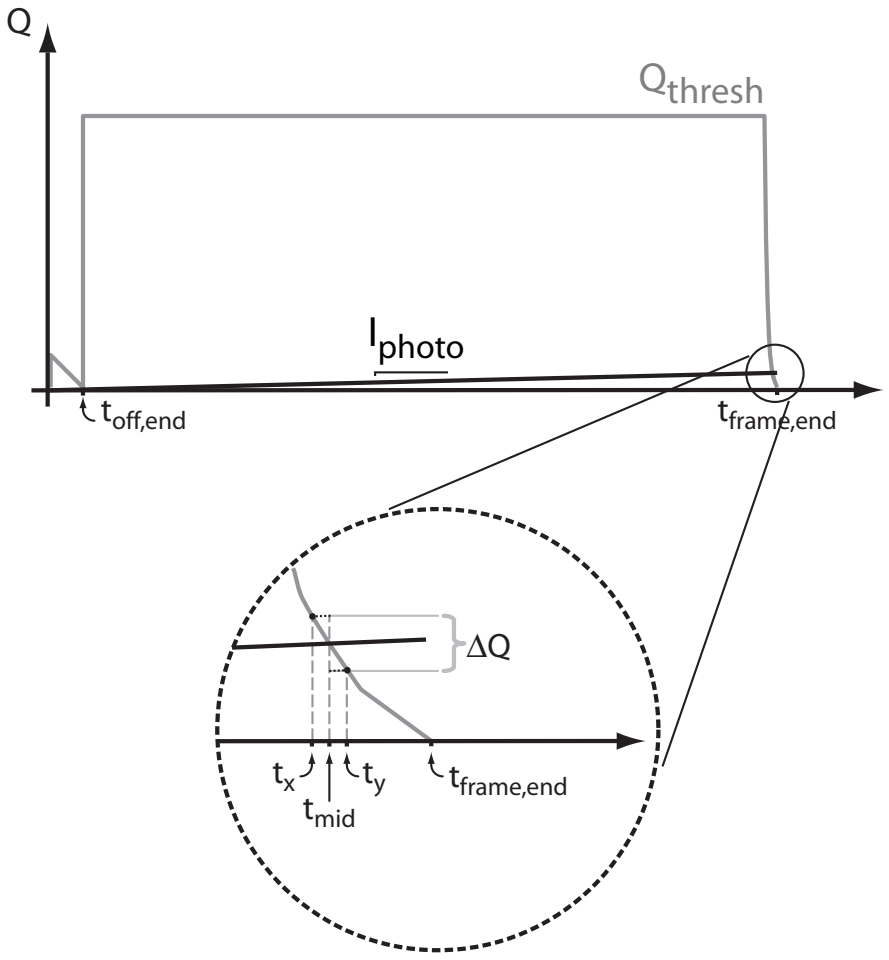

Figure 3-15: Derivation of quantization noise for low I_{photo} levels using Q_{thresh}.

view, suppose t_x and t_y are two successive times at which the pixel's threshold status is transmitted to its memory element. If the pixel's integrated charge crosses Q_{thresh} during this interval, this choice of transmission times will result in a quantized threshold crossing time of $t_{mid} = \frac{t_x+t_y}{2}$ being recorded. As drawn, the integrating charge crosses Q_{thresh} at exactly t_{mid}, resulting in no error between the quantized and true threshold crossing times. However, if the initial conditions of this charging ramp happened to be perturbed, for example due to charge uncertainty resulting from reset noise, the integrating charge could cross Q_{thresh} as early as t_x^+ or as late as t_y^-, resulting in a time-domain quantization error between the true versus recorded threshold crossing times. Note that this error arises due only to an uncertainty in the exact threshold crossing time – if the exact time were known, the amount of charge present at V_{pd} at that moment could be inferred through Q_{thresh}, and is therefore completely deterministic.

Instead of leaving this error in the time domain, we would like to refer it to an equivalent uncertainty in the the charge domain, since this is the format in which all of the intrinsic noise sources have been represented. This can be accomplished by taking t_{mid} to be the true threshold crossing time, and calculating the range over which the charge at this instant in time must be allowed to vary to account for all possible ways the integrating charge can cross t_{mid}. As depicted in the zoomed view, the resulting charge uncertainty ΔQ can be expressed as

$$\Delta Q = \Delta Q_{thresh} + I_{photo} \cdot \Delta T, \tag{3.23}$$

where $\Delta Q_{thresh} = [Q_{thresh}(t_x) - Q_{thresh}(t_y)]$ and $\Delta T = (t_y - t_x)$. Since we don't know the true manner in which the integrating charge traversed this interval, apply the standard assumption that all possible traversals occur with equal probability. This allows the result of Equation 3.7 to be employed, and yields an effective electron variance at V_{pd} of

$$\sigma^2_{e^-_{quant,low}} = \frac{1}{12} \cdot \left[\frac{\Delta Q}{q}\right]^2 = \frac{1}{12} \cdot \left[\frac{\Delta Q_{thresh} + I_{photo} \cdot \Delta T}{q}\right]^2. \tag{3.24}$$

Typically, ΔT is chosen to be the smallest available interval ΔT_{min} to minimize the difference between Q_{thresh} and $Q_{thresh,ideal}$ over the descending regions of these waveforms. As a result, once ΔT_{min} is known, only ΔQ_{thresh} needs to be chosen.

To yield further insight into this result, note that applying a standard analog-to-digital conversion at a fixed time t_{mid} with step size ΔQ_{thresh} would yield the same quantization

noise as given by this equation with $\Delta T = 0$. The additional charge uncertainty introduced by the $I_{photo} \cdot \Delta T$ term accounts for the fact that the input can continue to vary while the A/D conversion is being performed.

Flat-Top Region of Q_{thresh}

Moving now to the flat-top region of Q_{thresh}, as shown in Figure 3-16, suppose t_x and t_y are chosen as a new pair of times for the pixel-to-memory communication to occur. As in the previous case, due to potential variations in the exact threshold crossing time, a time-domain quantization error will occur which we would like to represent as if it were due to a charge uncertainty at time t_{mid}. The required charge uncertainty ΔQ at time t_{mid} is again

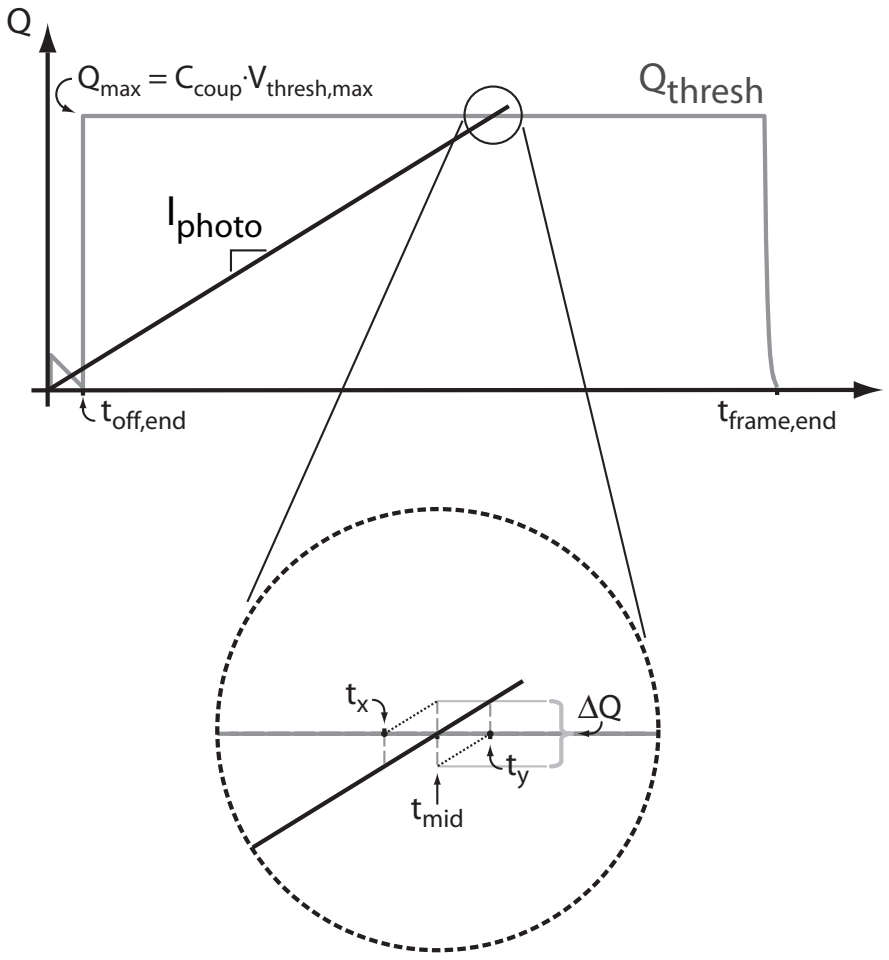

Figure 3-16: Derivation of quantization noise for high I_{photo} levels using Q_{thresh}.

illustrated in the zoomed view, and is defined by the relation given previously in Equation 3.23. However, in this case, $\Delta Q_{thresh} = 0$ due to the fact that Q_{thresh} is flat in this region of operation. Thus, we obtain the simplified form for the electron variance at V_{pd} of

$$\sigma^2_{e^-_{quant,high}} = \frac{1}{12} \cdot \left[\frac{\Delta Q}{q}\right]^2 = \frac{1}{12} \cdot \left[\frac{I_{photo} \cdot \Delta T}{q}\right]^2. \tag{3.25}$$

Further insight can be gained into this result by making the approximation that $I_{photo} \approx \frac{Q_{max}}{t_{mid}}$, which is fairly accurate due to the fact that at these high photocurrent levels the initial conditions are too small to significantly perturb this relation. Substituting this approximation for I_{photo} into Equation 3.25 yields

$$\sigma^2_{e^-_{quant,high}} = \frac{1}{12} \cdot \left[\frac{Q_{max}}{q}\right]^2 \cdot \left[\frac{\Delta T}{t_{mid}}\right]^2, \tag{3.26}$$

which demonstrates that it is the relative uncertainty in the threshold crossing time $\frac{\Delta T}{t_{mid}}$ that controls the quantization noise over this region of the characteristic.

Offset-Measurement Region of Q_{thresh}

The final source of pixel quantization noise arises during the initial threshold crossing of the offset-measurement phase, which is illustrated in Figure 3-17. Notice that there is no conceptual difference between this phase of operation and the descending region of Q_{thresh} discussed above. Therefore, the charge quantization error will be the same as in that case, and is given by

$$\sigma^2_{e^-_{quant,off}} = \frac{1}{12} \cdot \left[\frac{\Delta Q}{q}\right]^2 = \frac{1}{12} \cdot \left[\frac{\Delta Q_{thresh} + I_{photo} \cdot \Delta T}{q}\right]^2, \tag{3.27}$$

where again $\Delta Q_{thresh} = [Q_{thresh}(t_x) - Q_{thresh}(t_y)]$ and $\Delta T = (t_y - t_x)$. As before, ΔT is chosen to be the smallest available interval ΔT_{min}, which in this case minimizes the length of the offset phase and increases the maximum non-saturating photocurrent, defined by Equation 3.11. Thus, once ΔT_{min} is known, only ΔQ_{thresh} needs to be designed. Finally, one difference that does arise between the descending region at the end of Q_{thresh} and this offset measurement region is that in the former case it is guaranteed that $I_{photo} \leq \frac{Q_{max}}{t_{frame,end}}$, while in the latter I_{photo} can span the entire range of potential photocurrents.

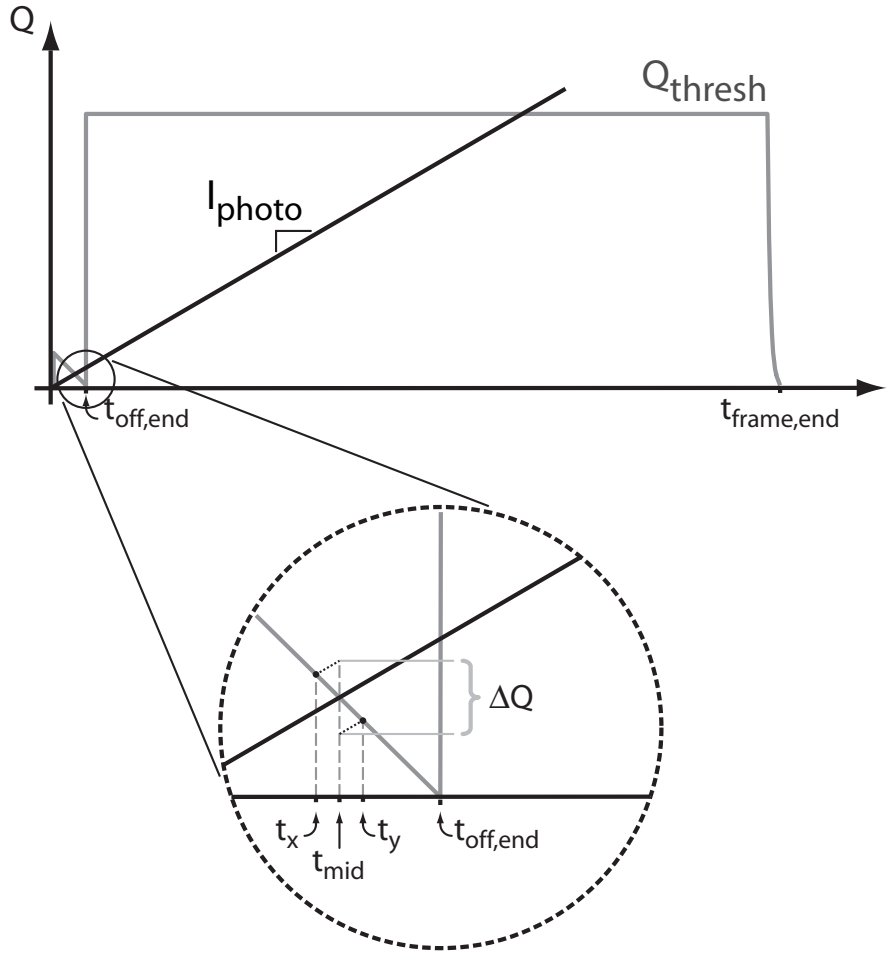

Figure 3-17: Derivation of offset-measurement region quantization noise.

3.4.4 Defining V_{thresh} and S

Armed with the noise analysis presented above, we are now in a position to define the threshold waveform V_{thresh} and pixel-to-memory communication timing waveform S. Based on this prior analysis, it is clear that $Q_{thresh} = V_{thresh} \cdot C_{coup}$ should be designed to lie as close as possible to the ideal waveform $Q_{thresh,iedal}$ shown in Figure 3-12, and that the length of the offset measurement phase should be kept as short as possible. Besides ensuring the accuracy of the assumptions that have been previously made, working to meet these goals also enables each pixel to collect as close to the theoretical maximum number of electrons possible for a given photocurrent level, which helps maximize its overall SNR. The remaining goal is to ensure that the waveforms' resulting quantization noise remains

negligible in comparison to the intrinsic pixel noise floor. These requirements actually leave significant flexibility in the exact definitions of V_{thresh} and S, particularly the quantization noise requirement, which can be met by various combinations of sample timings and voltage movements. Therefore, only the general strategy that was used to design the V_{thresh} and S waveforms will be outlined here.

Step 1: Check Quantization Noise Limit Due to ΔT_{min}

Determine the expected value for ΔT_{min}, which will be limited by the amount of time required by the hardware imager to perform two successive comparisons and pixel-to-memory threshold transfers across the entire pixel array. Use the resulting value to add the electron variance noise term

$$\sigma^2_{e^-_{quant,min}} = 2 \cdot \frac{1}{12} \cdot \left[\frac{I_{photo} \cdot \Delta T_{min}}{q} \right]^2, \tag{3.28}$$

to the noise plot of Figure 3-14. Based on Equations 3.24, 3.25, and 3.27, this term represents a lower bound on the achievable quantization noise, which is fundamentally limited by ΔT_{min}. The factor of two accounts for the two quantization errors accrued using the dual-threshold algorithm, one during the initial threshold crossing and one during the second threshold crossing. Look for locations where this quantization noise approaches the intrinsic noise floor of the pixel, signaling a potential decrease in pixel SNR that can only be counteracted by working to reduce ΔT_{min}. A representative plot has been generated for the example pixel using $\Delta T_{min} = 1.5$ μs, and is shown in Figure 3-18. At the highest photocurrent levels, the quantization noise bound lies less than 10 dB below the intrinsic noise floor, and will slightly reduce this pixel's SNR in this region.

Step 2: Generate Descending Edge of V_{thresh} and S

Begin generating the V_{thresh} and S waveforms starting at time $t_{frame,end}$ with V_{thresh} initialized to a low voltage level and working backwards in time, as shown in Figure 3-19. This waveform terminus serves as the final comparison and pixel-to-memory threshold transmission time on the waveform S. To generate the next-to-last transmission time, take a step of width ΔT_{min} backward along the time axis, and increment V_{thresh} upward by an amount ΔV_{low}. The value of this voltage increment is chosen such that $\sigma^2_{e^-_{quant,low}}$, given in Equation 3.24, remains a factor of $f_{quant,power}$ below the low-photocurrent noise floor shown in

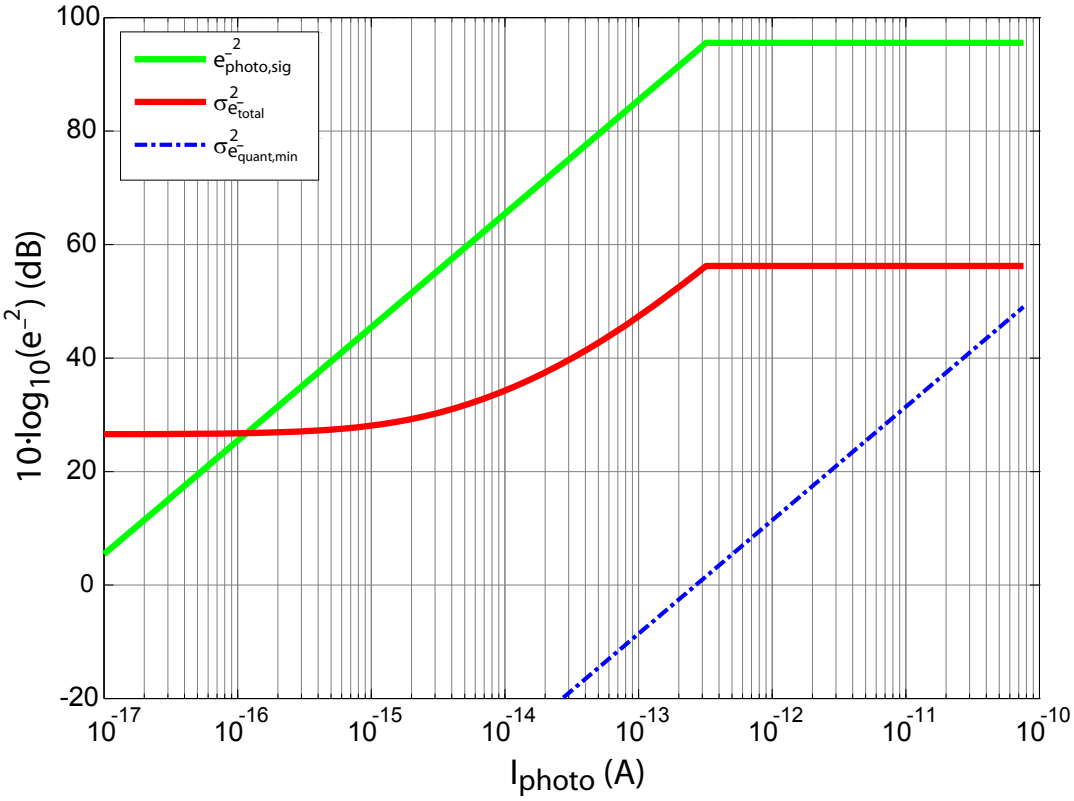

Figure 3-18: Plot of pixel signal, noise floor, and quantization noise lower bound for dual-threshold crossing algorithm with $\Delta T_{min} = 1.5$ μs.

Figure 3-18. Continue generating preceding transmission times and V_{thresh} voltage steps by repeating this procedure of stepping backwards along the time axis by an amount ΔT_{min}, and incrementing the V_{thresh} waveform upwards. Note that at each location on the V_{thresh} waveform, we can use the approximate relationship $I_{photo} \approx \frac{C_{coup} \cdot V_{thresh}(t_{thresh})}{t_{thresh}}$ to estimate the photocurrent level that can be expected to cause a threshold crossing to occur in that region of the curve[3]. This photocurrent estimate, along with noise plot in Figure 3-18, can then be used to determine the intrinsic pixel noise floor, enabling the step size of V_{thresh} to increase as the noise floor rises. This allows V_{thresh} to reach its flat-top region at $t_{flat\text{-}top,end}$ using the least amount of time and number of comparisons. The final voltage step between times $(t_{flat\text{-}top,end} + \Delta T_{min})$ and $t_{flat\text{-}top,end}$ may be smaller than the preceding step size ΔV_{high}, but this simply results in reduced quantization noise and is of no consequence.

[3]This is approximate because it assumes the initial condition of zero charge on V_{pd} at time zero.

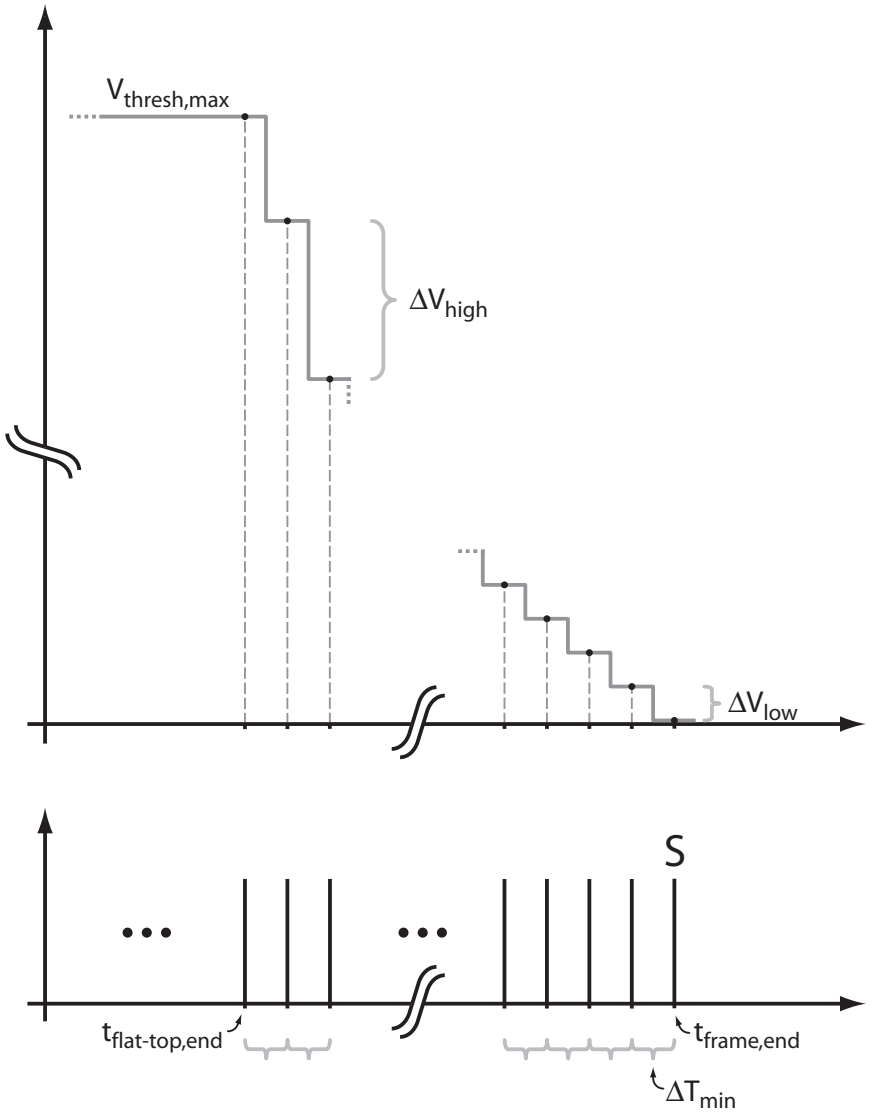

Figure 3-19: General strategy for generating descending edge of V_{thresh} and S.

Step 3: Generate Flat-Top Region of V_{thresh} and S

The design of the flat-top region begins at $t_{flat\text{-}top,end}$, the final transmission time generated by the previous step, and proceeds backwards in time as illustrated in Figure 3-20. Since the value of V_{thresh} is fixed at $V_{thresh,max}$ over this entire region, only a set of appropriate transmission times needs to be defined. The particular times must be chosen such that $\frac{\Delta T}{t_{mid}} \leq \beta$ for each interval, where β is the value of this ratio that ensures the quantization noise power $\sigma^2_{e^-_{quant,high}}$, given by Equation 3.26, lies a factor of $f_{quant,power}$ below the pixel's

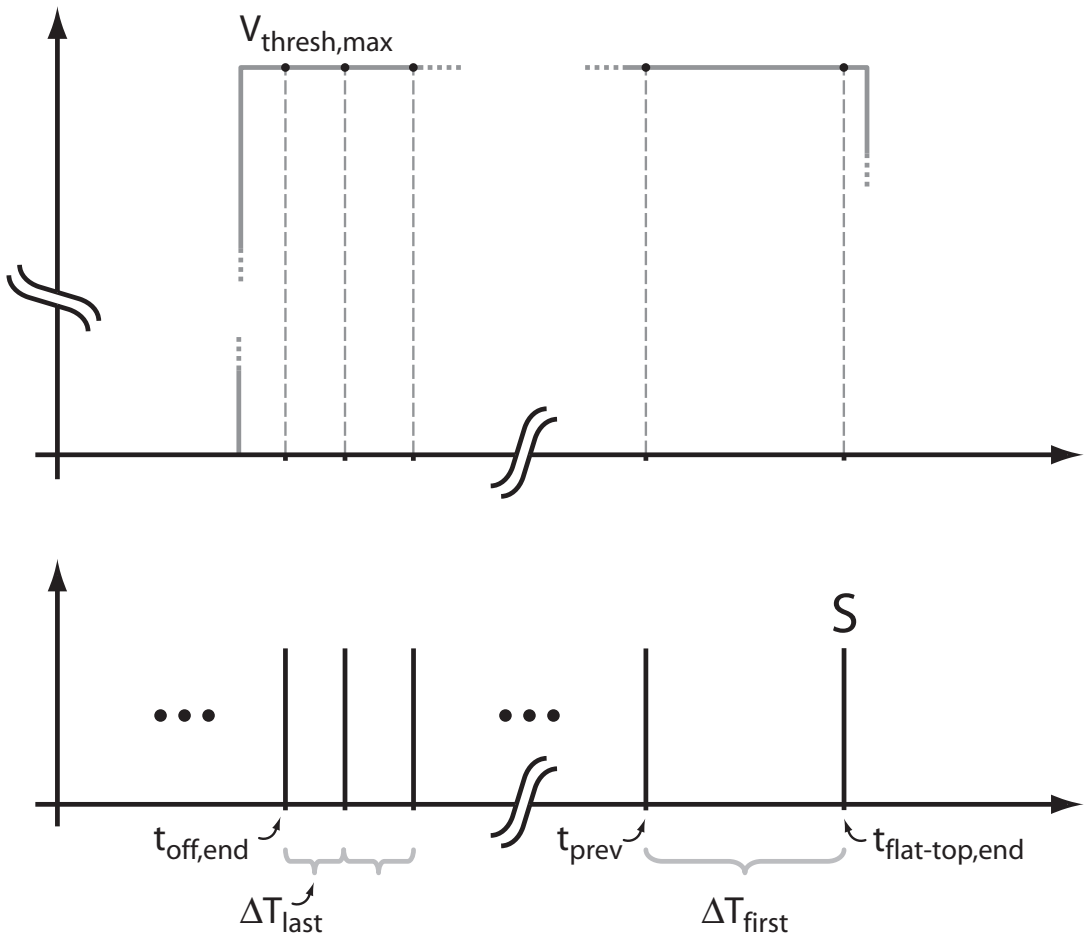

Figure 3-20: General strategy for generating flat-top region of V_{thresh} and S.

intrinsic noise floor. Once β has been calculated, which only needs to be done once since the intrinsic noise floor is fixed in this region, the remaining task is to choose transmission times that satisfy $\frac{\Delta T}{t_{mid}} \leq \beta$. Starting with the known transmission time $t_{flat\text{-}top,end}$, we can implicitly define the location of a slightly earlier transmission time t_{prev} by requiring that it meet the constraint

$$\frac{\Delta T_{first}}{t_{mid}} = \frac{[t_{flat\text{-}top,end} - t_{prev}]}{\left[\frac{t_{flat\text{-}top,end}+t_{prev}}{2}\right]} = \beta. \tag{3.29}$$

Solving this equation for t_{prev} gives

$$t_{prev} = t_{flat\text{-}top,end} \cdot \left[\frac{2-\beta}{2+\beta}\right], \tag{3.30}$$

which demonstrates that this earlier transmission time is located at a fixed fraction of the known transmission time. This relationship can be recursively applied to yield additional transmission times, all of which meet the required constraint, until either $\Delta T_{last} \cdot \frac{2-\beta}{2+\beta} < \Delta T_{min}$ or time $t_{off,end}$ is reached. Regardless of which condition terminates the recursion, the first sampling time of the flat-top region is considered to be $t_{off,end}$.

Step 4: Generate Offset Region of V_{thresh} and S

The design of the offset region begins at time zero[4] and at an offset voltage of $V_{off,max}$ and proceeds forward in time, as shown in Figure 3-21. The value $V_{off,max}$ is chosen high enough that all pixels in the array, regardless of their exact initial conditions and photocurrent levels, are guaranteed to generate an initial threshold crossing event during the offset phase. The first comparison and transmission time occurs at time ΔT_{min} and voltage $V_{off,max}$. The next comparison and transmission occurs at time $2 \cdot \Delta T_{min}$, and voltage $V_{off,max} - \Delta V_{high,off}$, where the voltage step size is chosen small enough to keep $\sigma^2_{e^-_{quant,off}}$, defined in Equation 3.27, at least a factor of $f_{quant,power}$ below the pixel's high-photocurrent intrinsic noise floor. We can continue generating new transmission times and V_{thresh} offset steps by repeating this procedure of stepping forwards along the time axis by an amount ΔT_{min}, and decrementing the V_{thresh} waveform downwards. Similar to the descending portion of the waveform discussed earlier, different photocurrent levels should intersect this offset waveform at different times, with some perturbations added on top due to variations in initial conditions. We can again use the approximate relationship $I_{photo} \approx \frac{C_{coup} \cdot V_{thresh}(t_{off})}{t_{off}}$ to estimate the photocurrent level that can be expected to cause a threshold crossing in a particular region of the offset waveform. This photocurrent estimate, along with noise plot in Figure 3-18, can then be used to determine the intrinsic pixel noise floor in that region, and the step size can be adjusted accordingly. This is particularly important as we move downward in voltage during the offset phase because the intrinsic noise floor decreases for lower photocurrents, requiring the step size to be reduced.

Example

A MATLAB script was written combining the approaches outlined above to automatically generate V_{thresh} and S given a set of input pixel parameters [72]. An additional constraint

[4]The pixel reset phase is impelemented at the very end of the previous frame to allow for this.

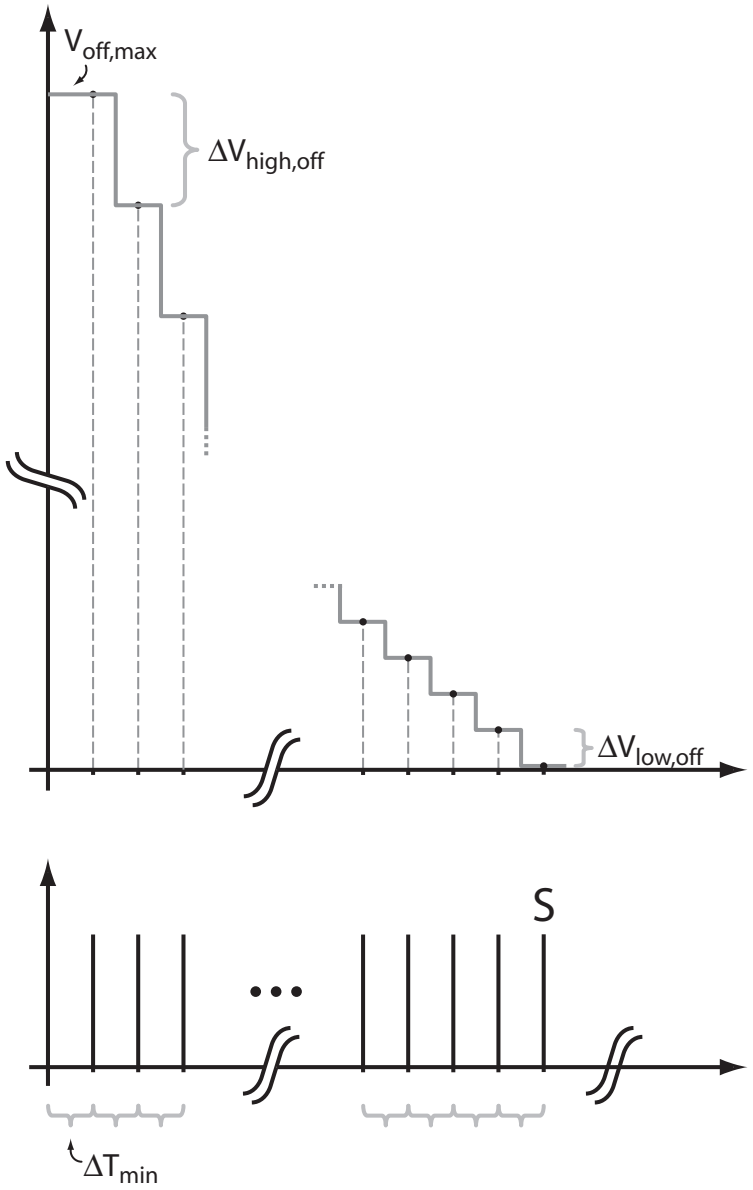

Figure 3-21: General strategy for generating offset phase of V_{thresh} and `S`.

was added to the script requiring V_{thresh}'s voltage steps be discrete multiples of a unit step size ΔV_{LSB}, enabling the waveform to be generated exactly using a DAC with this minimum step size. A representative V_{thresh} waveform created by this script is shown in Figure 3-22. It is plotted using discrete points to illustrate both its voltage level and the transmission timing of `S`. A close-up view of the offset region of the waveform is shown in Figure 3-23. A knee was implemented in the offset region to take advantage of the intrinsic

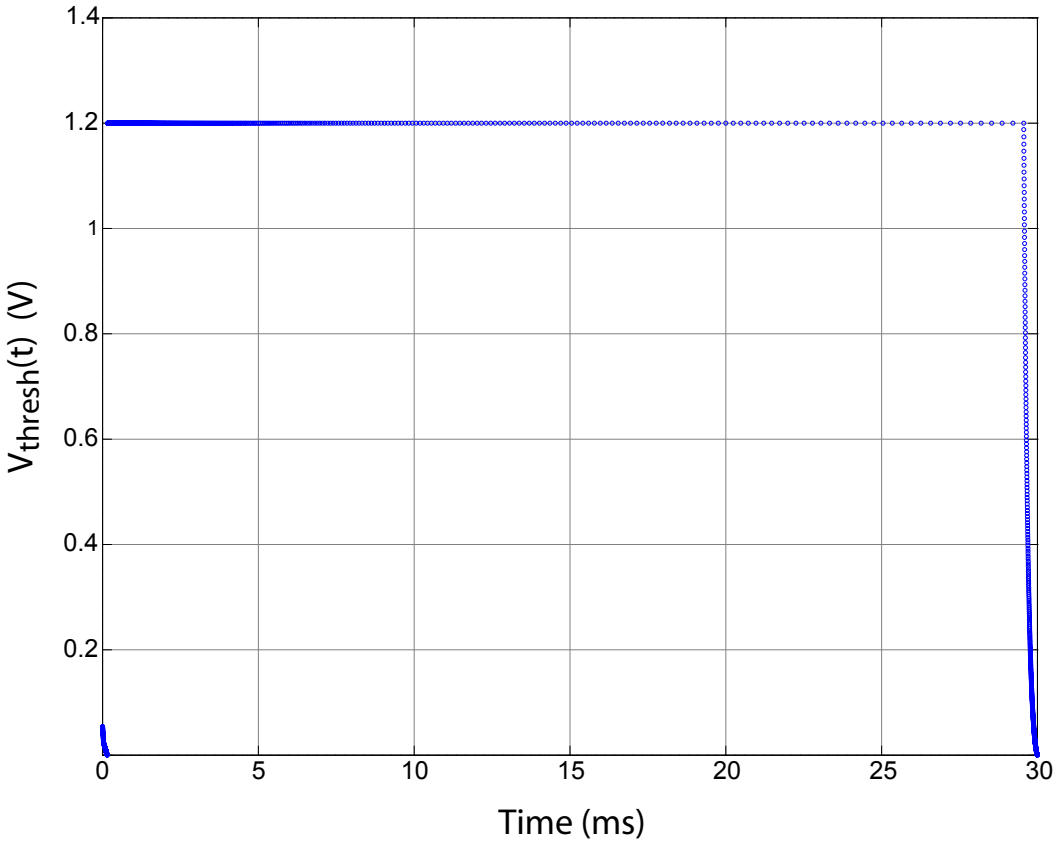

Figure 3-22: A representative V_{thresh} waveform designed to contribute a quantization noise power a factor of $f_{quant,power} = 10$ below the intrinsic pixel noise floor shown in Figure 3-18. The waveform contains a total of 835 sample times, a maximum inter-sample interval ratio of more than 230 $\times$, and was designed using the following parameters: $\Delta T_{min} = 1.5$ μs, $t_{frame,end} = 30$ ms, $C_{coup} = 8$ fF, $V_{thresh,max} = 1.2$ V, and a DAC minimum step size of $\Delta V_{LSB} = \frac{V_{thresh,max}}{2^{12}-1}$.

noise floor varying with photocurrent level. Even subject to uncertain initial conditions, low-level photocurrents should cross threshold below the knee voltage, where small voltage steps are employed to minimize quantization noise. Only higher-level photocurrents should reach threshold above the knee voltage, where larger steps can be used to help minimize the overall length of the offset interval. More complicated approaches employing multiple knees can be imagined, but were not explored. A close-up view of the descending edge of the waveform is shown in Figure 3-24. Unlike the sharp knee-based step size adaptation employed in the offset region, the descending edge step size is smoothly adapted to allow the quantization noise to track the changing intrinsic noise floor, allowing both the number

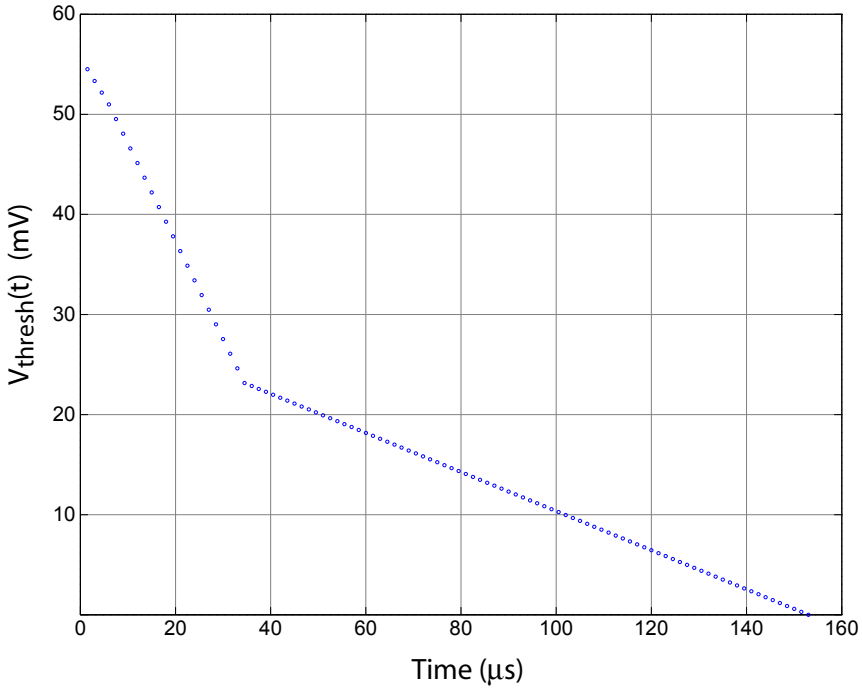

Figure 3-23: Close-up view of the V_{thresh} waveform offset region, illustrating the knee adaptation in step size.

of comparisons and the width of this interval to be minimized.

An updated noise plot that includes the quantization noise power associated with this particular V_{thresh} waveform is shown in Figure 3-25. The discrete jumps visible in the quantization noise power above 1 fA are due to the discrete step size limitation imposed by the DAC. Notice that at very high photocurrent levels the quantization noise power begins to rise due to the intrinsic limit set by ΔT_{min}, as discussed earlier, but then disappears just below 60 pA. This location marks the $I_{photo,max}$ level associated with this particular V_{thresh} waveform, and is slightly lower than the upper limit in the remaining curves because they were generated assuming $t_{off,end} = 128$ μs, while for this particular V_{thresh} waveform $t_{off,end} \approx 162$ μs.

3.5 Summary

This chapter discussed the intrinsic advantages of time-based imagers, as well as the main disadvantages that have been observed in past implementations based on this approach. It then introduced a novel dual-threshold algorithm and capacitively-coupled pixel architecture

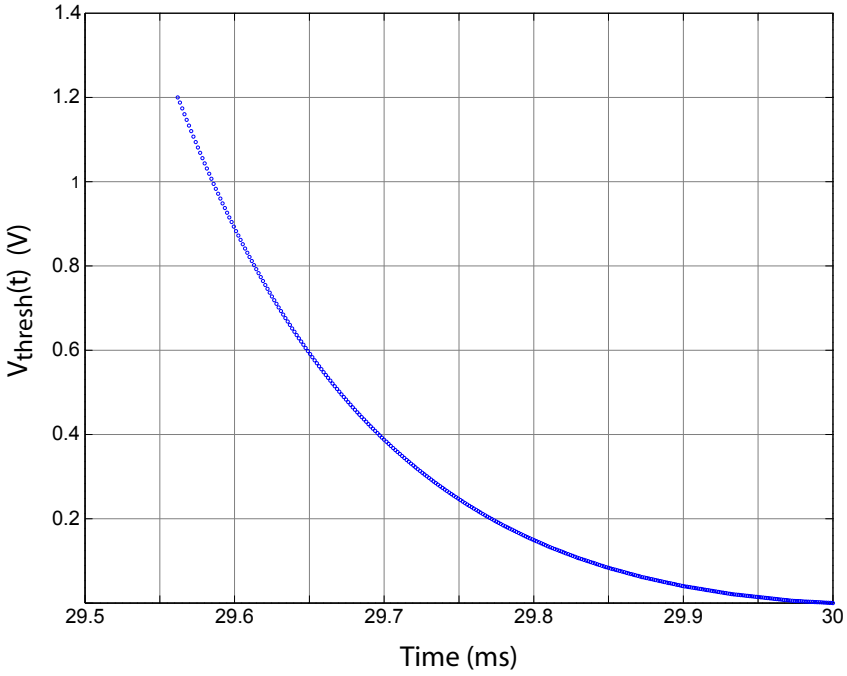

Figure 3-24: Close-up view of the V_{thresh} waveform descending edge, illustrating the smooth adaptation in step size.

that together addressed many of these past limitations. Some of the highlights of this approach include:

- The dual-threshold algorithm achieves wide-dynamic range, high SNR, and is inherently insensitive to offset fixed-pattern noise.
- The capacitively-coupled pixel exhibits an inherently linear response, with only a minimal increase in pixel hardware.
- Time-domain quantization errors associated with the synchronous pixel-to-memory communication can be engineered to lie well-below the intrinsic pixel noise floor, preserving the inherent pixel SNR. This allows the pixel memory to be transferred to a separate on-chip array, improving fill factor.

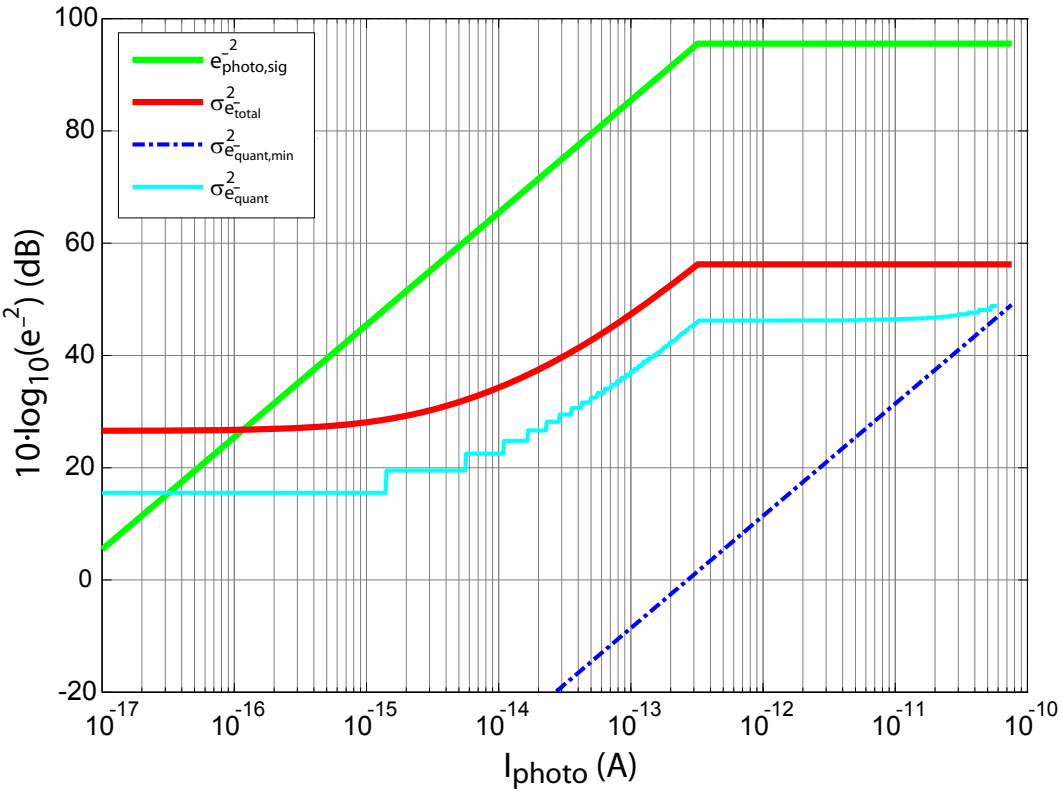

Figure 3-25: Plot of pixel signal, noise floor, and quantization noise contributed by the example V_{thresh} waveform shown in Figure 3-22.

Chapter 4

Prototype Imager Design

The previous chapter introduced a novel time-based dual-threshold wide-dynamic-range imaging algorithm and analyzed its operation and performance, primarily from a systems perspective. The present chapter expands upon this work by discussing the design of a prototype imager implementing this wide-dynamic-range algorithm in a standard $0.18\mu m$ CMOS process. The prototype imager is composed of several major interacting blocks, as illustrated in Figure 4-1. As the details of each block are discussed in the following sections, the reader may find it helpful to refer to this figure as a reminder of how they collectively form the imager as a whole.

4.1 150 × 280 Spiking Pixel Array

An array of 150 × 280 spiking pixels forms the core of the prototype time-based imager. The right-most 24 columns of the array are covered with a metal shield to block incoming photons, allowing them to serve as a dark current reference for the array, while the remaining 150 × 256 segment of the pixel array is available for image capture. Other than the metal photon shield, all pixels in the array are identical in design and layout, and are constructed as described below.

4.1.1 Spiking Pixel Design

As discussed in Chapter 3, each time-based pixel is built from four key components: a photodiode, a comparator, a linear capacitor, and some form of memory. While integrating the memory within the pixel enables low-latency communication between the memory and

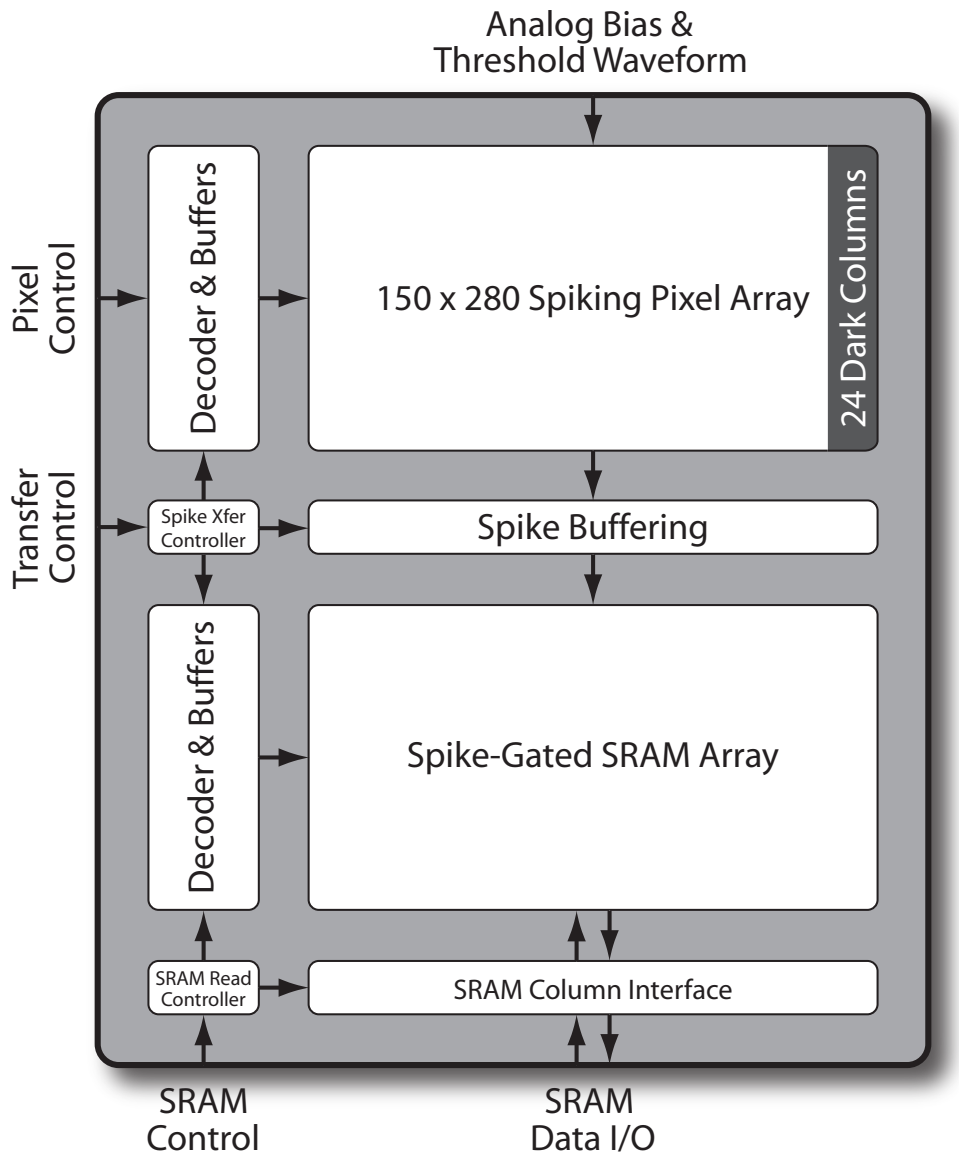

Figure 4-1: Block diagram of prototype imager chip.

comparator – a critical feature for recording time-encoded information – it can seriously degrade pixel fill factor [40–46, 49, 51, 60, 62, 68–70]. In an effort to maximize fill factor, the prototype imager implements the pixels and memory as two separate arrays, as shown in Figure 4-1. Ensuring low-latency communication between these two arrays is critical (and is discussed in Section 4.2), however for the moment we will only consider the resulting modification to the local pixel topology: the memory is replaced with a spike-generation circuit. A block-level view of the prototype spiking pixel with the aforementioned modification is

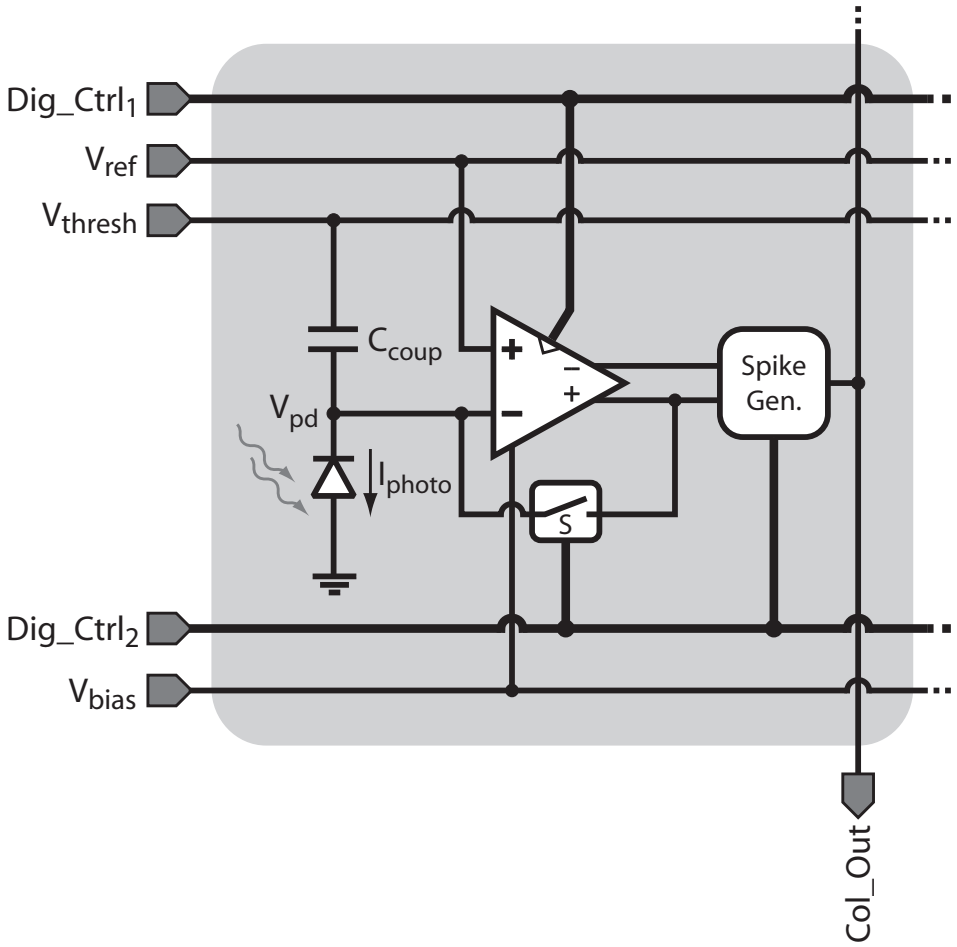

Figure 4-2: Block level spiking pixel schematic.

shown in Figure 4-2. The design of each pixel component is discussed below.

Photodiode

In standard CMOS processes, several p-n junctions are available for implementing the pixel photodiode, each with their own unique properties. In order to select the optimal junction, it is helpful to review the various factors that affect photodiode and pixel performance:

- From the perspective of photon-to-photocurrent conversion, a junction with a large space-charge region width is desirable, as this increases the electron collection volume and thus quantum efficiency. Based on Equation 2.3, it is clear that a junction's space-charge region width increases as the doping of the lightly-doped side is decreased.

- The junction depths in modern processes lie in the range of 0.2 μm to 2.0 μm below

the wafer surface, while the range of depths over which visible photons are absorbed is on the order of 0.7 μm to 20 μm (see Section 2.1). Therefore, we would expect the deeper junctions to exhibit superior quantum efficiency in the green-to-red region of the visible spectrum.

- As shown in Equation 3.18, the pixel's input-referred comparator noise yields an effective r.m.s electron noise at the integrating node that is proportional to the parasitic capacitance at this node. The reverse-biased photodiode's contribution to this parasitic is dominated by its depletion capacitance, which is commonly modeled by

$$C_{dep} = C_{jo} \cdot Area \cdot \left[1 - \frac{V_d}{\phi_o}\right]^{-m_j} + C_{jswo} \cdot Perimeter \cdot \left[1 - \frac{V_d}{\phi_{swo}}\right]^{-m_{jsw}}, \quad (4.1)$$

  where C_{jo} (C_{jswo}), ϕ_o (ϕ_{swo}), and m_j (m_{jsw}) represent the zero-bias junction capacitance per unit area (perimeter), bottom-face (sidewall) junction built-in potential, and bottom-face (sidewall) grading coefficient, respectively, and V_d is the forward bias across the diode. Based on this equation, for a photodiode with a given area and perimeter, C_{dep} can be minimized by selecting a junction with low C_{jo} and C_{jswo} and increasing the reverse bias across it.

- It is advantageous to employ a photodiode with a small reverse-bias leakage current, or dark current, to minimize extraneous noise in dim illumination conditions. Two separate mechanisms can contribute to the overall reverse-bias leakage of a junction. First, the saturation current given by

$$I_s = qAn_i^2\left(\frac{1}{N_A}\frac{D_e}{L_e} + \frac{1}{N_D}\frac{D_h}{L_h}\right) \quad (4.2)$$

  flows through any reverse-biased diode. Second, thermal carrier generation within the junction space-charge region contributes an additional leakage term that can be modeled by

$$I_{gen} = \frac{qAn_i w_{SCR}}{2\sqrt{\tau_{eo}\tau_{ho}}}. \quad (4.3)$$

  Without delving into the details of these equations, two important dependencies can be immediately seen that have direct impact on our junction choice. The first equation shows that the saturation current leakage decreases with increased low-side doping

(either N_A and N_D, depending on the junction), while the second equation demonstrates that the leakage due to thermal generation increases with the space-charge region width w_{SCR}.

Since decreasing the junction low-side doping both increases its depletion width and reduces its depletion capacitance, the first and third requirements can be satisfied simultaneously. However, based on Equations 4.2 and 4.3, decreased low-side doping and increased depletion region width also lead to higher dark current. Striking the proper balance between these conflicting dependencies requires detailed reverse-bias junction leakage characterization. In particular, values for the strongly process-dependent electron and hole generation time constants τ_{eo} and τ_{ho} are needed, but not available. Therefore, the decision was made to optimize the photodiode with respect to the first three metrics only, which all suggest the use of the n-well/p-substrate junction. As will be shown later, the rectangular portion of the pixel area allotted to this photodiode measures 6.55 μm×10.34 μm. Substituting these dimensions, along with the available process parameters, into Equation 4.1, gives a mid-rail ($V_{DD}/2 = 0.9$ V) estimate for the photodiode depletion capacitance of

$$C_{dep}(V_{DD}/2) \approx 5.73 \text{ fF}. \tag{4.4}$$

Based on the limited process leakage data available, it is also possible to estimate a reverse-bias leakage current in the photodiode of $I_{reveak} \approx 0.4$ fA.

Linear Charge-Coupling Capacitor

As discussed previously in Section 3.3.3, the use of a linear capacitor to couple movements in V_{thresh} into to the pixel integrating node V_{pd} allows all critical comparisons to occur near a fixed common-mode input voltage V_{ref}, enabling the linearized pixel response described in Equation 3.6. Additional benefits of this technique include the fact that it significantly relaxes the input range over which the comparator must operate in a high-gain mode, simplifying its design and potentially allowing it to scale well to low rail voltages, and that it eliminates potential dependencies of the comparator input offset voltage on common-mode. These advantages will become more evident when the comparator design is considered in the next section.

A direct result of the relation developed in Equation 3.6 is that the value of C_{coup} affects

both the peak SNR and responsivity of the pixel. Based on the noise analysis presented in Section 3.4.2, in order to achieve at least 40 dB peak SNR for high-illuminance inputs the pixel must collect $N_e \geqslant 10^{\left(\frac{40\ \text{dB}+7\ \text{dB}}{10\ \text{dB}}\right)} \approx 5 \times 10^4$ electrons without saturating, where the extra 7 dB is included to ensure that gain FPN dominates over photo-generated shot noise. As discussed previously in Section 3.4.4, the maximum voltage swing on V_{thresh} is limited by the DAC implementation to roughly $V_{thresh,max} = 1.2$ V. Thus, based on Equation 3.6, the coupling capacitance should be chosen such that

$$C_{coup} \geqslant \frac{N_e \cdot q}{V_{thresh,max}} \approx 6.7 \text{ fF}. \tag{4.5}$$

Due to variation in capacitance per unit area over process, the nominal value of C_{coup} was chosen to be

$$C_{coup} = 8 \text{ fF}, \tag{4.6}$$

which corresponds to 6.7 fF along the minimum process edge. Finally, the chosen value of C_{coup} directly affects pixel responsivity, defined as the slope of the pixel's voltage integrating response evaluated around a given illuminance level. Based on the earlier observation that $V_{pd} = V_{ref}$ at both the beginning and end of de-integration, the proper interpretation of responsivity in this pixel is the $\frac{dV_{thresh}}{dt}(I_{lux})$ corresponding to a given illumination I_{lux}, which can be calculated as

$$\text{Responsivity} = \frac{I_{photo}/I_{lux}}{C_{coup}}. \tag{4.7}$$

The important trend to note from this relation is that responsivity improves with decreasing C_{coup}.

Synchronous Comparator

Since it consumes a significant fraction of the total pixel area and power budget, the comparator design is one of the most crucial aspects of the entire imager. Making its design even more difficult, aside from the scant area and power budgets, the comparator should also be capable of performing low-noise, low-offset comparisons in a noisy mixed-signal environment bombarded with stray photon-generated substrate carriers. Several algorithmic choices were made in Chapter 3 specifically to help achieve these design requirements. For example, employing a clocked architecture (see Section 3.3.5) helps reduce power con-

sumption and synchronizes mixed-signal crosstalk, while the two-pronged offset correction strategy consisting of auto-zeroing followed by dual-threshold frame capture (Section 3.3.2) mitigates mismatch errors and reset noise. Additionally, the introduction of the linear charge-steering capacitor (see Section 3.3.3) enables all critical comparisons to occur at a fixed voltage level, relaxing the comparator's input common-mode requirements. Together, these techniques enable the use of the compact, yet powerful, comparator topology depicted in Figure 4-3.

Using various configurations of the three digital control lines `Latch`, `Res_A`, and `Res_B`, this circuit can be reconfigured into several different topologies, each of which is employed during a different phase of pixel operation. Since the function of several of the circuit elements varies across phases, the logical approach to understanding the comparator as a whole is to consider each phase in turn. Is is helpful to refer to both Figures 4-2 and 4-3 during the following discussion.

- **Auto-Zero Phase:** This configuration is used once at the start of each new frame to auto-zero the comparator input offset and sample its value onto the capacitance at the pixel node V_{pd}. The phase begins with switch S closing to connect the comparator in unity-feedback while `Latch`, `Res_A`, and `Res_B` are driven `LOW`, `LOW`, and `HIGH`, respectively. This turns M_{11} and M_8 `ON` and turns M_7 and M_4 `OFF`, which diode-connects transistor M_9 and creates a simple current mirror from M_9 and M_{10}. The gate of transistor M_1 is biased by a fixed global reference voltage V_{bias} to generate a tail current for the differential pair formed by M_2 and M_3. The fixed comparator reference voltage level V_{ref} biases the gate of M_2 high enough above ground to allow this transistor to support the full tail current without bringing M_1 out of saturation. Feedback from V_{o+} through switch S biases the gate of transistor M_3 near V_{ref} as well, but with the comparator input-referred offset voltage V_{off} superimposed on top of it. Finally, the diode-connection of M_9 also provides the gate drive for transistors M_5 and M_6, and is sufficiently high to bias them as cascodes for the differential pair transistors M_2 and M_3, boosting their output impedance. The overall topology is that of a low-swing operational transconductance amplifier (OTA) biased in unity-feedback.

- **Common-Mode Feedback Bias Phase:** This phase establishes the dynamic bias point for the capacitive common-mode feedback (CMFB) network within the com-

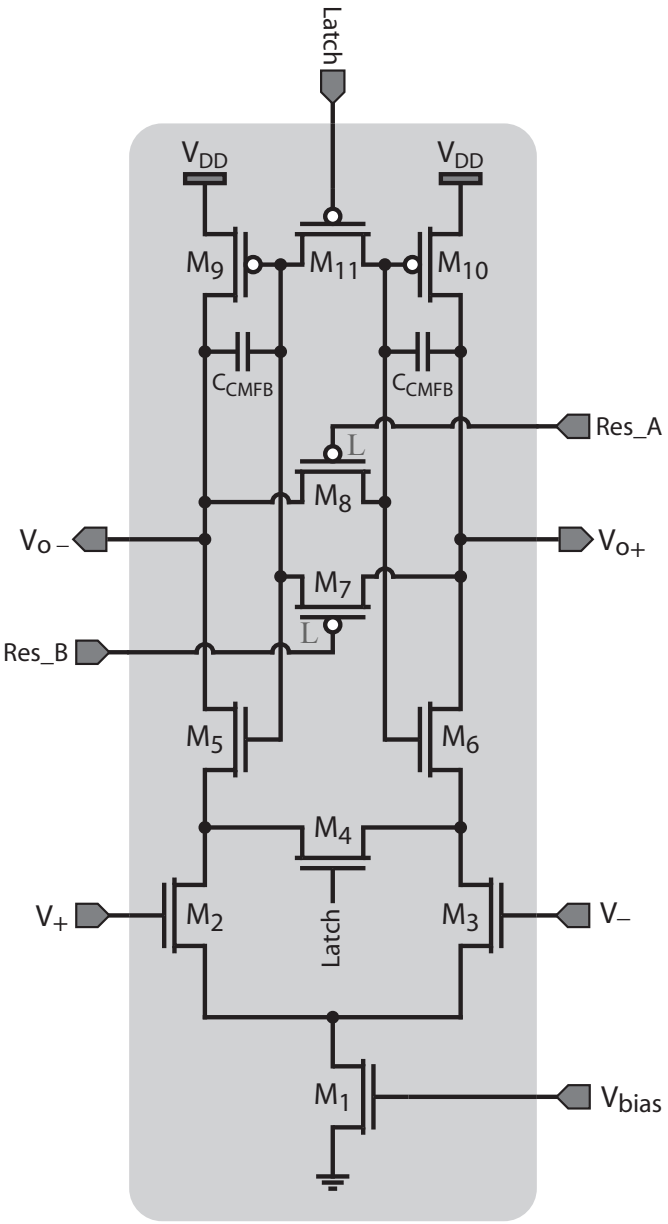

Figure 4-3: Synchronous fully-differential pixel comparator.

parator, and is employed at the start of each comparison. Switch S remains open during the entire phase, while `Latch`, `Res_A`, and `Res_B` are all driven `LOW`. This biases the comparator in a configuration similar to the auto-zero phase, except M_7 is now `ON`, diode-connecting M_{10}. Independent of the exact current steering of the input differential pair, all of the tail current reaches the parallel diode-connected transistors M_9 and M_{10}. These two transistors self-bias $V_{gs,M_{9,10}}$ at the precise level necessary

to support the tail current generated by M_1, and apply this voltage across the two shorted common-mode feedback (CMFB) capacitors, C_{CMFB}. This bias voltage is sampled on the parasitic capacitance at the shared inner node of the two CMFB capacitors when `Res_A` and `Res_B` simultaneously transition `HIGH`, opening M_7 and M_8. The positive charge injection ΔQ_{inj} due to M_7 and M_8 opening decreases the output common-mode level by roughly $\Delta V_{CM,out} \approx -\Delta Q_{inj}/(2 \cdot C_{CMFB})$, which can be significant for small values of C_{CMFB}.

- **Differential Low-Noise Amplification Phase:** After the CMFB bias is established, this phase is employed to amplify the input differential signal (V_{ref} - V_{pd}), reducing the effect of latch transistor mismatch and mixed-signal noise on the comparator in the following two phases. It begins where the CMFB bias step ends – at the instant `Res_A` and `Res_B` simultaneously transition `HIGH` to sample the bias point. The circuit topology is now that of a fully-differential amplifier, with M_5 and M_6 serving to cascode the input differential pair M_2 and M_3, increasing the gain to the output nodes V_{o+} and V_{o-}. The common-mode feedback capacitors form an ideal, large-signal-linear CMFB network, regulating the gate bias of M_9 and M_{10} to keep the amplifier in its high-gain region. Small differential inputs are amplified to generate a large differential output, which is naturally applied across the CMFB capacitors. During this low-noise amplification phase, all signals within the imager are held fixed, minimizing the effect of mixed-signal crosstalk on the amplified output. Added crosstalk immunity is provided by the fully-differential topology.

- **Sample-and-Hold Phase:** This phase is used to sample and temporarily store the output from the low-noise amplification so that it can be used to drive the final comparator latching step. The phase begins with `Latch` switching from `LOW` to `HIGH`, simultaneously turning transistor M_{11} `OFF` and transistor M_4 `ON`. Turning M_{11} `OFF` forces the amplified differential output voltage developed during the previous phase to be instantly sampled-and-held across the two CMFB capacitors, each of which now forms a local capacitive integrator structure[1] with transistors M_9 and M_{11}. Since ideally no net charge can be added to or subtracted from the integrator virtual ground nodes (i.e., the gates of M_9 and M_{10}) once M_{11} opens, the sampled voltage across each

[1]Each integrator structure is analogous to an operational amplifier with capacitive negative feedback.

capacitor should remain fixed past this point. Charge injection from M_{11} turning `OFF` does enter these two virtual ground nodes, however as long as it is not too large, it will be primarily a common-mode disturbance that won't significantly alter the stored differential voltage.

Ideally, the voltages sampled across these two capacitors will solely determine the eventual latching direction of the comparator, as these values were generated during the low-noise amplification phase. In reality, the latching direction can still be significantly influenced by the post-sampling values of V_+ and V_-, which both experience significant transient disturbances following any switching signal applied to the imager array. Transistor M_4 solves this problem by connecting the drains of M_2 and M_3 together at the same instant that `Latch` triggers the sample-and-hold. This relegates the former input differential pair to serve as two parallel cascodes for transistor M_1, making the voltage movements at V_+ and V_- unimportant past this instant. Although V_{bias} experiences similar mixed-signal crosstalk, it manifests as a common-mode disturbance in the comparator bias current, which the differential structure rejects. Closing M_4 also configures M_5 and M_6 as a differential pair in preparation for the latching phase of operation.

- **Latch and Power-Down Phase:** The final phase of comparator operation uses positive feedback to amplify the output voltage differential sampled in the previous phase into a digital comparison decision. This phase begins where the sample-and-hold phase ends, and is initiated by driving both `Res_A` and `Res_B` to a `LOW` level. This turns M_7 and M_8 `ON`, cross-coupling the two latch structures formed by the transistor pairs M_5/M_9 and M_6/M_{10}. The latch outputs have been preset by the result of the differential amplification and sample-and-hold operations, pre-disposing the resulting positive-feedback loop to latch the circuit in a similar direction. Eventually, one output will saturate at V_{DD} while the other saturates at GND, resulting in static power dissipation in the comparator due only to subthreshold leakage. Implementing M_7 and M_8 with low-threshold devices (signified by the 'L's in Figure 4-3) allows for faster settling to the rails at the expense of higher static subthreshold leakage. Finally, while voltage offset due to transistor mismatch between the two latch structures will influence the latching direction, this error term is reduced by the gain of the differential

amplification phase.

In all phases of comparator operation discussed above, it has been implicity assumed that the inputs V_+ and V_- lie within a few millivolts of each other. When this is not the case, the transistor operating regions will no longer be as stated. However, it is only over this small differential input range that the comparator must operate as described – for larger differentials it becomes heavily pre-disposed to latching in a particular direction, making its task simple.

At this point, let's consider the biasing strategy employed to ensure the comparator operates as described for small differential inputs. In traveling from the upper rail $V_{DD} = 1.8$ V to ground, four voltage drops are traversed that require careful engineering: $V_{gs,M_{9,10}}$, $V_{gs,M_{5,6}}$, $V_{ds,M_{2,3}}$, and V_{ds,M_1}. Starting with the latter two terms, assuming all transistors will eventually be biased near moderate-inversion allows us to estimate a rough value of $V_{dsat} \approx 150$ mV, therefore biasing $V_{ds,M_{1\text{-}3}} > V_{dsat} + 100$ mV ≈ 250 mV should be sufficient to keep transistors $M_{1\text{-}3}$ saturated. An additional drop of $V_{dsat} + 100$ mV ≈ 250 mV is needed across $V_{ds,M_{5,6}}$ to ensure these cascode transistors remain saturated, implying a minimum operating voltage at V_{o+} and V_{o-} of roughly $3 \cdot (V_{dsat} + 100 \text{ mV}) \approx 750$ mV. Since $V_{o+} \approx V_{ref}$ during auto-zeroing, V_{ref} was biased at 850 mV to ensure high loop gain during this phase, and $M_{2,3}$ were sized such that $V_{gs,M_{2,3}} \approx 600$ mV, biasing $V_{ds,M_1} \approx 250$ mV. Applying the assumption of $V_{dsat} \approx 150$ mV to the PMOS transistors $M_{9,10}$, the maximum operating voltage at V_{o+} and V_{o-} should be limited to $V_{DD} - V_{dsat} - 100$ mV ≈ 1.55 V to keep these transistors saturated. To maximize the output swing during the low-noise differential amplification phase, the output common-mode voltage at the end of the common-mode bias phase should lie near $\frac{1.55 \text{ V} + 0.75 \text{ V}}{2} = 1.15$ V. This is accomplished by sizing $M_{9,10}$ such that $V_{gs,M_{9,10}} \approx -650$ mV and ensuring that the jump in output common-mode due to the positive charge injection from M_7 and M_8 at the end of the CMFB bias sampling phase is less than 100 mV. Finally, $M_{5,6}$ are sized such that $V_{gs,M5,6} \approx 650$ mV to achieve the desired $V_{ds,M2,3} \approx 250$ mV.

Finally, it is worth mentioning a few remaining implementation-specific issues. To minimize the input-referred $\frac{1}{f}$ noise of the comparator, extra area was allotted to $M_{2,3}$, as these two transistors are the primary contributors to this term. To help the comparator tail currents match better across the pixel array, M_1 was also allotted additional area. The remaining transistors were implemented near minimum size. The nominal comparator

Parameter	Value
I_{bias}	1 μA
C_{CMFB}	2.4 fF
V_{ref}	850 mV
Differential gain	44 dB
Diff. gain settling time (1%)	200 ns
Total input-referred noise	60×10^{-9} V^2

Table 4.1: Nominal comparator parameters for prototype imager.

parameters as implemented in the prototype imager are listed in Table 4.1.

Reset Switch

As described in the previous section, switch S (see Figure 4-2) temporarily closes during pixel reset, an event that occurs once at the start of each new frame. During this brief interval, the switch auto-zeros the comparator and samples the result on the capacitance at node V_{pd}. This simultaneously replenishes any charge de-integrated from V_{pd} during the previous frame, while eliminating much of the comparator offset from the current one. During the remainder of the frame the switch is nominally `OFF`, but must accomplish another important task: minimize its own parasitic leakage to V_{pd}. This is crucial, since in modern processes transistors that are digitally `OFF` typically conduct subthreshold currents orders of magnitude larger than many of the photocurrent levels the pixel is designed to sense. Fortunately, through careful design of the switch topology, this detrimental effect can be virtually eliminated [73]. One such low-leakage switch topology, which was used to implement S, is shown in Figure 4-4. The switch is controlled by two digital inputs, `Samp_Off` and `Samp_Off2`. When both signals are `HIGH`, M_1 and M_2 turn `ON` and M_3 turns `OFF`, connecting V_{pd} to V_{o+} through the internal node V_{mid}. To open the switch, `Samp_Off2` is first driven `LOW`, disconnecting V_{pd} from V_{mid}. A short time later `Samp_Off` is driven `LOW`, toggling the analog multiplexer formed by M_2 and M_3, which forces V_{mid} to V_{DD}.

To understand how this switch achieves low `OFF` state leakage, consider the terminal conditions of transistor M_1 whenever S is `OFF`. The gate and bulk are both biased at GND, while V_{mid} is biased at V_{DD}, ensuring this node serves as M_1's drain terminal. This leaves V_{pd} to serve as the source terminal, and ultimately dictate the overall leakage of S via the

subthreshold NMOS drain-to-source current relation

$$I_{DS} = I_{os}\frac{W}{L}e^{\kappa(V_{GS}-V_{TS})/\phi_t}\Big(1 - e^{-V_{DS}/\phi_t}\Big). \tag{4.8}$$

In this equation, I_{os} is a process-dependent current-scaling constant, W/L is the transistor width over length ratio, V_{TS} is the transistor threshold voltage, κ is a weak function of the applied source-to-bulk voltage and is typically around 0.7, and ϕ_t is the thermal voltage (roughly 25.9 mV @ T = 300 K) [74]. The threshold voltage V_{TS} is defined by the body-effect relation

$$V_{TS} = V_{TO} + \gamma\Big(\sqrt{\phi_o + V_{SB}} - \sqrt{\phi_o}\Big), \tag{4.9}$$

and is a function of the nominal threshold voltage V_{TO}, the applied source-to-bulk voltage V_{SB}, and γ and ϕ_o, which are process-dependent parameters [74]. It has already been mentioned that in the case where V_{pd} is biased at GND, the subthreshold current I_{DS} is substantial. However, since $V_{SB} = V_{pd}$, simply raising the voltage of this terminal will simultaneously increase V_{TS} while decreasing V_{GS}, both of which lead to exponential decreases in I_{DS}. Based on simulations of the process used to implement the prototype imager, ensuring $V_{pd} > 400$ mV is enough to completely eliminate subthreshold conduction in M_1 at the fast process corner. The discussion that follows explains how this is achieved in the prototype pixel.

Combining the capacitive-coupling technique of Figure 3-7 with a specific V_{thresh} waveform, for example the waveform in Figure 3-22, the representative set of pixel waveforms illustrated in Figure 4-5 can be generated. Notice that the leakage of M_1 must only be

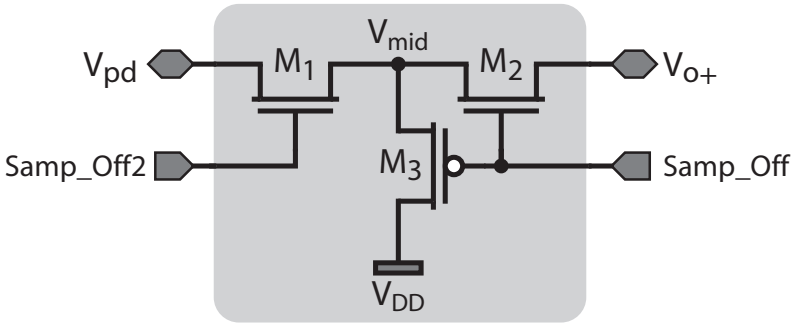

Figure 4-4: Low-leakage pixel reset switch topology.

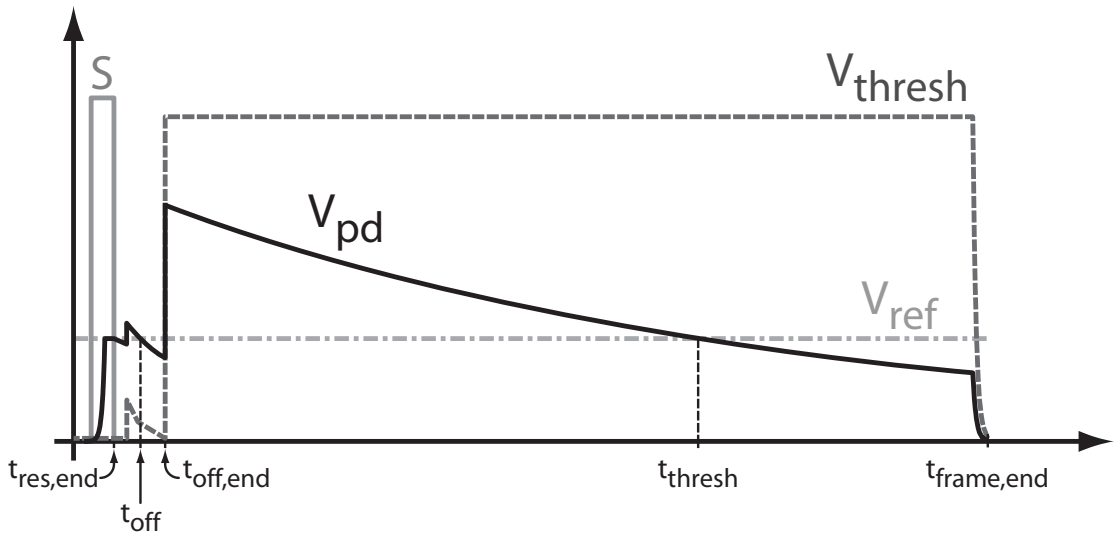

Figure 4-5: A representative set of pixel waveforms over a single frame interval. Note the length of the offset phase has been exaggerated to illustrate details within this time interval.

minimized between the end of the reset phase at $t_{res,end}$ and the second threshold of the dual-threshold algorithm[2] at t_{thresh}. Including the effect of the comparator offset, it is clear from Figure 4-5 that $V_{pd} \geqslant V_{ref} + V_{off}$ over the sub-interval starting at the end of the offset phase $t_{off,end}$ and ending at t_{thresh}. While V_{off} can be positive or negative, its magnitude is typically on the order of tens of millivolts while $V_{ref} = 850$ mV, thus $V_{pd} \gg 400$ mV over this entire sub-interval.

On the other hand, during the complementary sub-interval starting at $t_{res,end}$ and ending at $t_{off,end}$, there are clearly regions where $V_{pd} < V_{ref}$. While this alone is not detrimental to pixel performance, problems could arise if V_{pd} de-integrates below the established 400 mV lower boundary during this period. To determine if this is possible, recall that the maximum non-saturating photocurrent input (see Equation 3.11) is given by

$$I_{photo,max} = \frac{C_{coup} \cdot V_{thresh,max}}{t_{off,end}}, \tag{4.10}$$

where as discussed in Section 3.4.4 the pixel reset phase occurs at the end of previous frame to allow the offset phase to begin at $t_{res,end} = 0$. For any photocurrent larger than this value, the step in V_{thresh} at time $t_{off,end}$ will be of insufficient magnitude to pull V_{pd} back above

[2]No pixel data is recorded past the second threshold, so leakage past this point doesn't matter.

its auto-zeroed level, thus the second threshold will never occur and the pixel saturates. Applying $I_{photo,max}$ to the pixel, let's now calculate $\Delta V_{pd,max}$ over the interval from $t_{res,end}$ to $t_{off,end}$. Notice that V_{thresh} undergoes no net change across this interval, making its contribution to $\Delta V_{pd,max}$ zero, and that $I_{photo,max}$ must discharge a parasitic capacitance at V_{pd} consisting of C_{coup} plus the effective voltage-dependent parasitic capacitance between V_{pd} and GND, $C_{eff}(V_{pd})$. To simplify the resulting expression for $\Delta V_{pd,max}$, temporarily replace the nonlinear $C_{eff}(V_{pd})$ with an equivalent linear capacitance C_{eq}, whose value is chosen such that the same charge is extracted from both capacitors over a voltage swing from V_{high} to V_{low} [75]. This substitution enables the derivation of the implicit relation

$$\begin{aligned} \Delta V_{pd,max} &= -\frac{I_{photo,max} \cdot (t_{off,end} - t_{res,end})}{C_{coup} + C_{eq}} \\ &= -\frac{V_{thresh,max} \cdot C_{coup}}{C_{coup} + C_{eq}}, \end{aligned} \tag{4.11}$$

where the solution is constrained by the requirement that $\Delta V_{pd,max} = V_{low} - V_{high}$. This result simply states that the maximum change in V_{pd} during the offset phase is governed by the total charge consumed by $I_{photo,max}$ during this interval, divided by the total capacitance this charge is de-integrated from. Setting $C_{coup} = 8$ fF, $V_{high} \approx V_{ref} = 850$ mV, and $V_{thresh,max} = 1.2$ V, and substituting the necessary process parameters into the standard equations for C_{eq}, MATLAB can be used to find $\Delta V_{pd,max} = -561$ mV. This implies that $V_{pd}(t_{off,end}^{-}) = 289$ mV, signaling a potential problem. Returning to simulation, $V_{pd} = 289$ mV yields roughly 6× more leakage at the fast process corner than the minimum leakage achieved when $V_{pd} > 400$ mV. Fortunately, due to the inverse dependence of t_{thresh} on illumination, this leakage increase is more than compensated for by the several orders of magnitude decrease in the effective integration time when compared to low illumination inputs, thus the noise contribution due to this term remains completely negligible.

Finally, while subthreshold conduction must be avoided during pixel data capture, past t_{thresh} it is a welcomed effect, as it can be used to prevent blooming. Blooming can potentially occur when more than one photodiode shares a common terminal, typically the substrate. If one of the photodiodes becomes forward biased, the minority carrier concentration increases in the vicinity of this junction, establishing a net outward diffusion of carriers from the photodiode into the common substrate. Inevitably, a fraction of these carriers will diffuse into and be collected by the depletion regions of neighboring photodiodes,

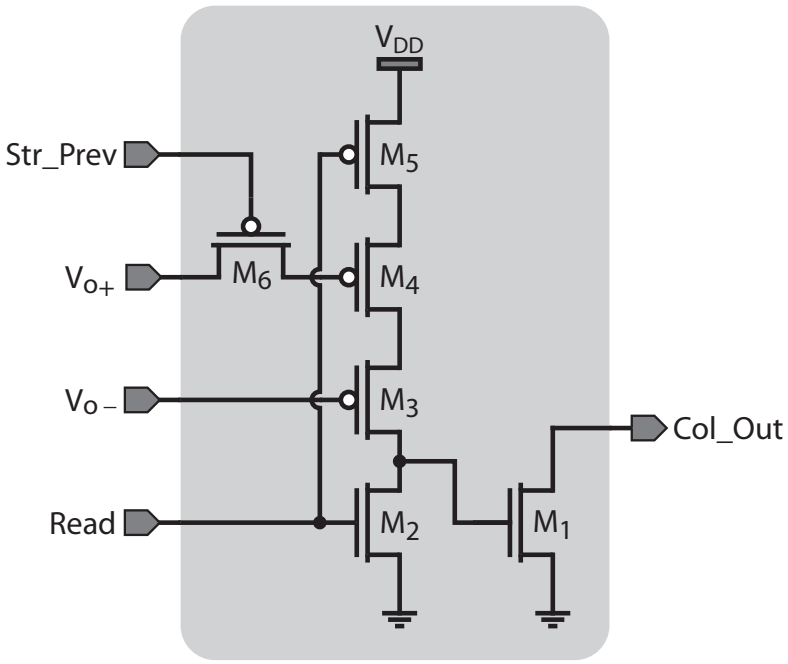

Figure 4-6: Circuit used to implement pixel-level spike generation.

corrupting these pixels' data. Transistor M_1 can prevent blooming by establishing a lower bound on V_{pd} that ensures all photodiodes remain reverse-biased. This lower bound varies with I_{photo}, and can be calculated by solving Equations 4.8 and 4.9 for the $V_{SB} = V_{pd}$ that sets $I_{DS} = I_{photo}$. Alternatively, simulations can be used to ensure that over the range of expected I_{photo} levels, V_{pd} always saturates before the photodiode forward-biases.

Spike Generator

The spike generation circuit is the final component of the prototype pixel design. The purpose of this block is to generate a single synchronous digital pulse on the shared column bus `Col_Out` if the pixel comparator detected a threshold crossing during the most recent comparison. This function is implemented using the circuit shown in Figure 4-6. To understand how the spike generator operates, refer to the test circuit and timing waveforms illustrated in Figure 4-7. This test circuit contains an ideal comparator, the spike generator block of Figure 4-6, and a shared column-level load resistor, which together model the operation of the spike generator in the actual imager array. As the timing waveforms demonstrate, a single cycle of spike generator operation begins with a `LOW` pulse on `Str_Prev` which,

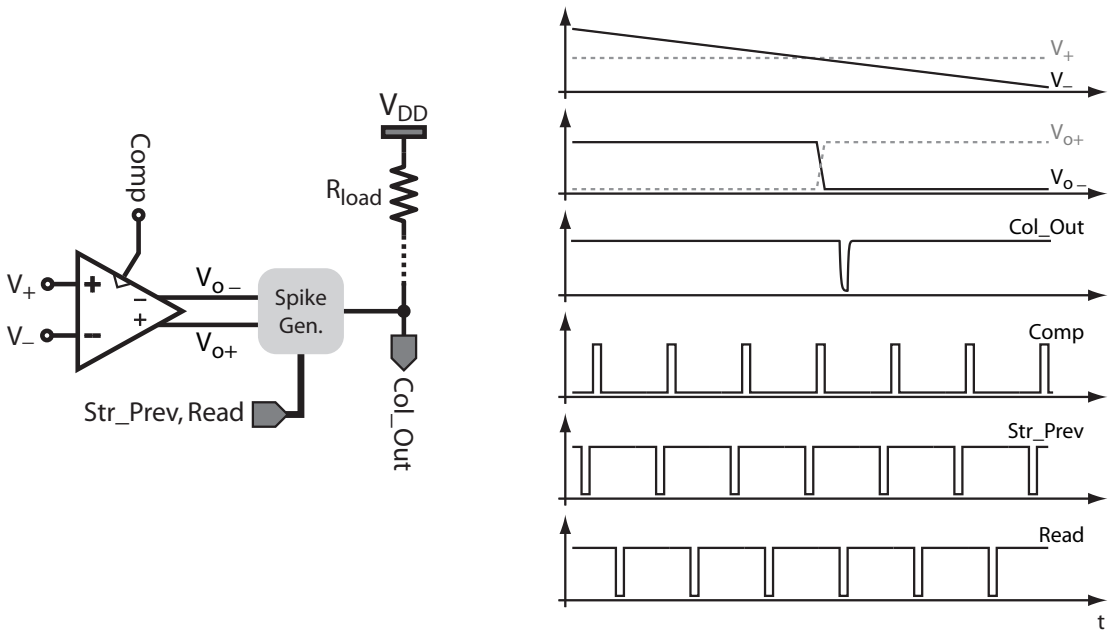

Figure 4-7: Illustration of spike generation circuit operation.

referring to Figure 4-6, briefly turns M_6 `ON` and samples the latched V_{o+} from the previous comparison (at t_{N-1}) onto the parasitic capacitance at the drain of M_6. Following this sampling operation, `Comp` pulses `HIGH` (at t_N) and the ideal comparator generates a new comparison result and latches it at its output – a simplified but conceptually accurate model of the actual pixel comparator. Finally, `Read` is pulsed `LOW`, which turns M_2 `OFF` and M_5 `ON`. During this brief pulse interval, if the stored result from the previous comparison $V_{o+}(t_{N-1})$ was `LOW` and the result of the current comparison $V_{o-}(t_N)$ is `LOW`, transistors $M_{3\text{-}5}$ will all be `ON`, driving the gate of M_1 `HIGH`. This forces M_1 to sink current from `Col_Out`, pulling the wired-OR column bus `LOW` and communicating to the periphery of the pixel array that the currently selected pixel crossed threshold between times t_{N-1} and t_N. Note that this particular event, the comparator having just crossed threshold, is the only one that is sufficient to generate the spike on `Col_Out`. Thus, under normal operation each pixel generates only two spikes per frame, minimizing the energy each must expend in driving the large parasitic capacitance on the column bus.

Layout

The 12.5 μm×12.5 μm spiking pixel layout has been sliced into the six cross-sections shown in Figure 4-8 to allow a few of its more salient features to be visible. Starting at the substrate, the first slice depicts transistor-level details along with the location of the 6.55 μm×10.34 μm n-well/p-substrate photodiode. Although a small portion of this photodiode area is occluded by the n^+ contacts in each of its upper corners, a net 42.7% pixel fill factor is still achieved. The next four slices show the low-to-mid level metal routing in the pixel. *Metal 1* is used primarily for transistor-level connections, but also vertically routes the analog bias lines V_{ref} and V_{bias} through the pixel, allowing them to easily connect to the necessary transistor gates. The ground and power planes run in *Metal 2* and *Metal 4*, respectively, and sandwich the digital control lines that run horizontally in *Metal 3*. This sandwich helps minimize parasitic capacitive coupling from the digital control lines to the sensitive analog portions of the pixel. Moving to the upper metal layers, the two common-mode feedback capacitors C_{CMFB} and the linear coupling capacitor C_{coup} are fabricated using thin-oxide MIM structures formed wherever *Metal Cap* and *Metal 5* overlap[3], with all three piggy-backed on top of the pixel's circuit region to maximize photodiode area. Special attention was paid to the routing of the four connections between the CMFB capacitor plates and the comparator. A pair of shielded conduits were constructed between *Metal 5* and *Metal 1* in which to route these signals, as is most evident in the fifth slice where two pairs of signals are passing through the *Metal 4* layer surrounded on all sides by V_{DD}. On the lower metals, particularly *Metal 3*, there is no room for a true conduit. However, through careful engineering of the transistor level layout, it was possible to route the digital lines `Samp_Off2` and `Str_Prev` on either side of the conduit signals. Since the time intervals during which these two digital lines transition and the CMFB capacitors are needed are mutually exclusive, these lines effectively shield the signals as they pass through the digital layer. Finally, V_{thresh} along with two parallel `Col_Out` lines are routed vertically along the pixel edges in *Metal 5*. A careful examination of the layout slices in Figure 4-8 would reveal that the pixel spike generator is not connected to either of the `Col_Out` lines. This is because these connections vary with pixel location in the array as a whole, as discussed in the next section.

[3]The process requires *Metal Cap* plates to be contacted from above, explaining the use of *Via 5* and *Metal 6* within these regions.

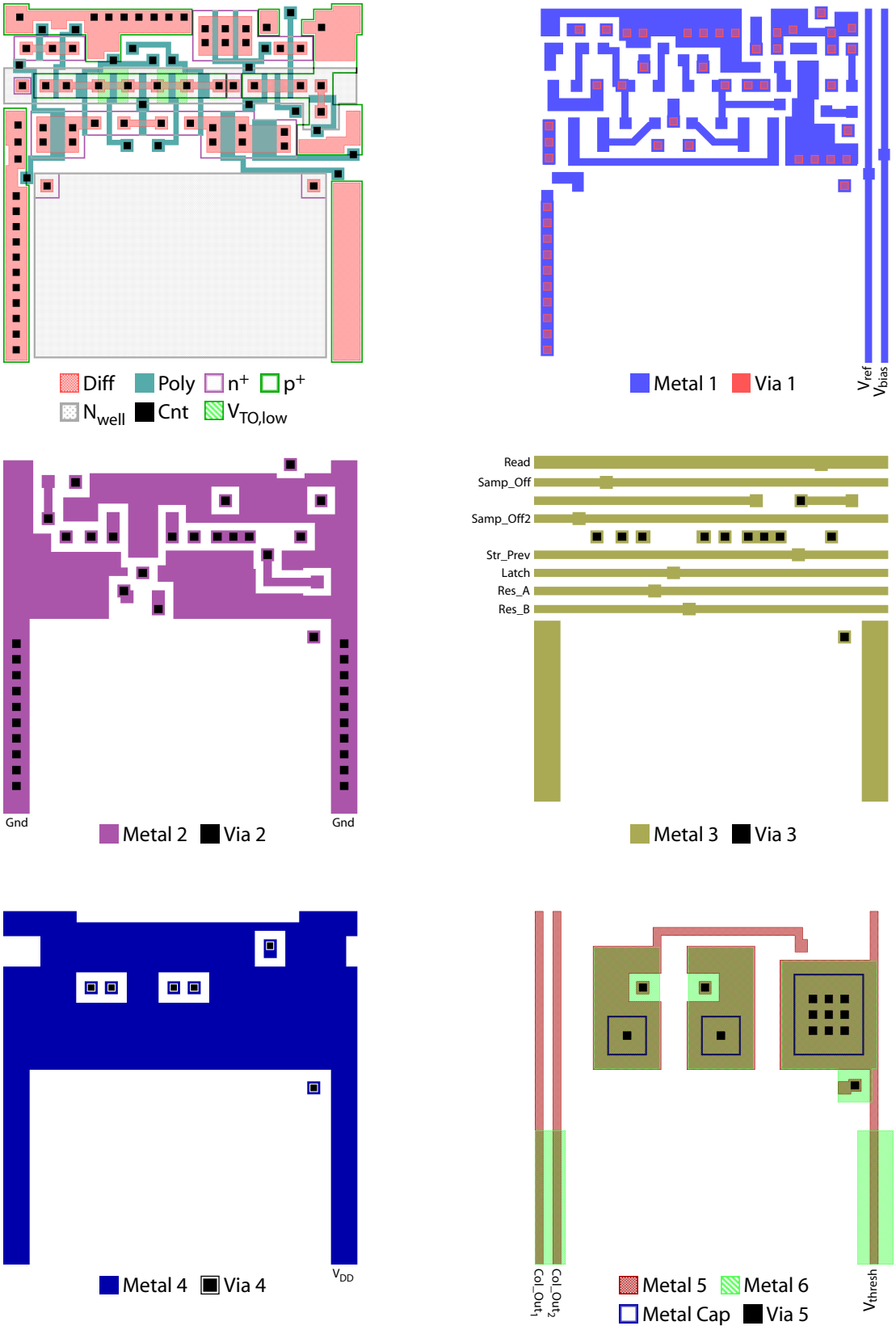

Figure 4-8: Six cross-sectional views of the 12.5 μm×12.5 μm spiking pixel layout.

4.1.2 Pixel Array Design

The design of the pixel array is actually almost complete at this point, as most of its structure is dictated by the unit pixel described in the previous section. However, this should not be taken to imply that an arbitrary pixel design can be arrayed to create a successful imager. On the contrary, though they were not discussed in detail, array-level design considerations such as supply bypassing and parasitic resistance, capacitance, delay, and coupling in the pixel layout, were all taken into account. It was this co-design of the pixel and array that is enabling the seamless transition between the two.

The structure of a full $N \times M$ pixel array is illustrated in Figure 4-9 and includes two final details that were alluded to earlier but have not been formally discussed. First, as mentioned in the previous section, two parallel `Col_Out` lines are routed vertically through each pixel. These two lines, $\texttt{Col\_Out}_{\texttt{x,1}}$ and $\texttt{Col\_Out}_{\texttt{x,2}}$, connect to the pixel spike generator outputs on odd and even rows, respectively, allowing spikes from pairs of rows to be output simultaneously to the periphery of the array. The second detail, which was previously shown in Figure 4-7, is that each column is terminated in a load resistor R_{load} to V_{DD}. This forces each `Col_Out` to nominally reside at V_{DD}, but to exhibit a negative voltage swing in response to a pixel spike generator sinking current from it. Employing load resistors rather than a pre-charging scheme simplifies the array readout, as sequential pairs of rows can be read without having to reset the columns in between. However, unlike pre-charge transistors, the location of the load resistors with respect to the array is important. Comparing the two load resistor placement options – the top versus the bottom of the array – and including the effects of the `Col_Out` lines' parasitic resistance, the former option is preferred because it maximizes the voltage swing at the bottom of the array, where the voltage-mode spike detection circuitry is located (see Figure 4-1). The remaining task at this point is to choose a value for R_{load}, which requires a better understanding of the pixel-to-memory communication strategy. Therefore, this choice is delayed until the next section.

4.2 Pixel-to-Memory Communication

As discussed briefly in Section 4.1.1, the pixels and memory were implemented in two separate arrays in the prototype imager. While this does yield a significant increase in pixel fill factor, the physical separation between the pixel comparators and their associated memory

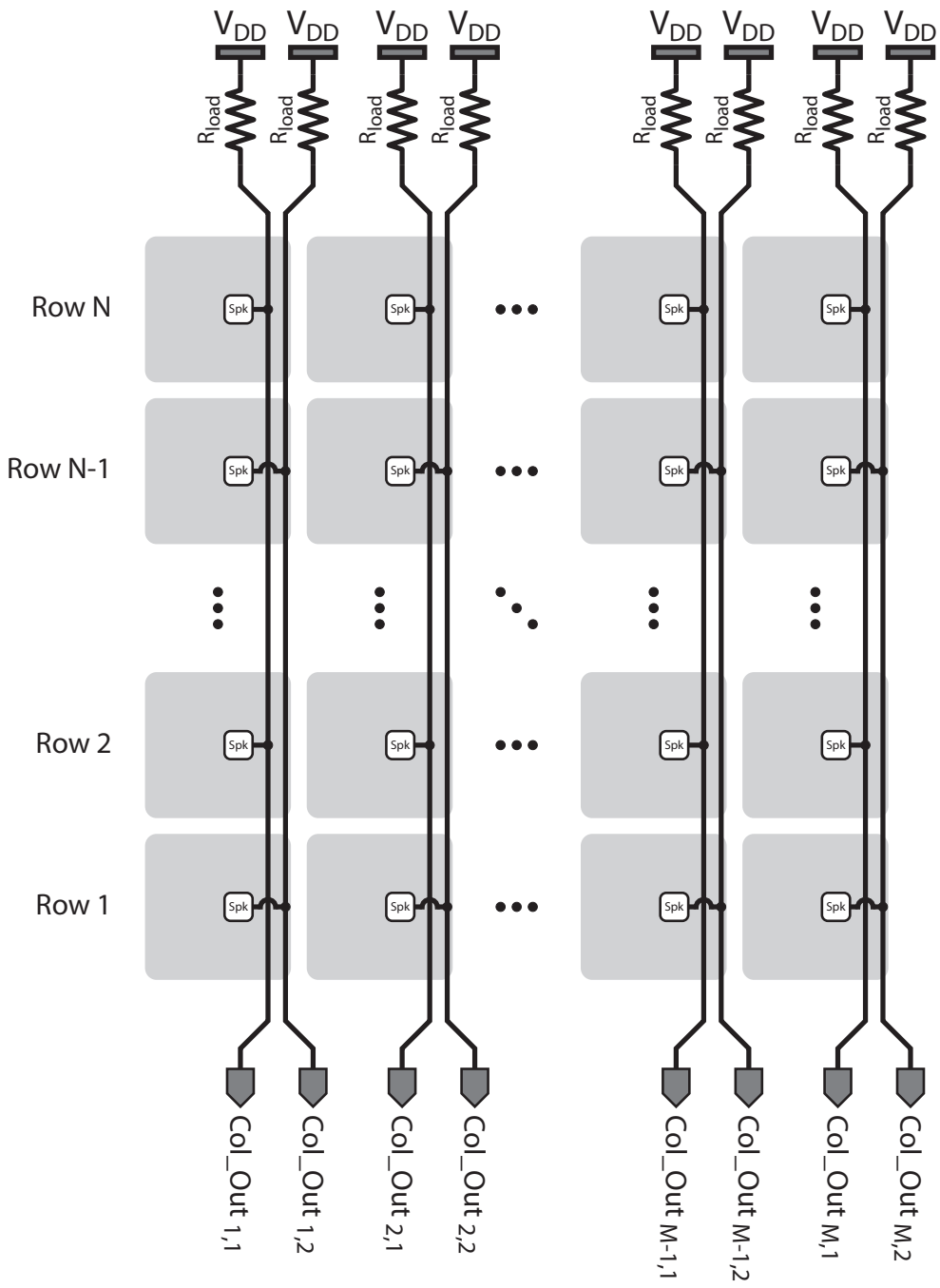

Figure 4-9: Pixel array structure illustrating column load resistors and dual `Col_Out` lines.

cells introduces a transmission latency between them – a potentially serious problem for a time-based imager. To facilitate a discussion of this latency issue, as well as gain a better understanding of the desired interaction between the pixel array and the remaining imager blocks, it is helpful at this point to formalize the prototype imager's pixel-to-memory

communication strategy.

Begin by taking a moment to reconsider the generic time-based pixel introduced in Figure 3-1. It is clear that the comparator transmits a single bit of information to the memory cell after each comparison, encoding whether the photodiode voltage was above or below threshold at the decision time. In the prototype imager, this direct-connection comparator-to-memory link is replaced with a more complicated time-division multiplexing strategy which, in order to avoid information loss, must be capable of ensuring that every pixel/memory cell pair can still exchange this bit after each comparison. This is where the choice to employ synchronous comparisons, and to explicitly account for the resulting temporal quantization noise, becomes extremely valuable. In the synchronous approach, the comparison times and the data to be recorded if the threshold is crossed during a comparison form a one-to-one mapping. Thus, as long as the pixel-to-memory bit transmission is completed before the start of the next comparison, any latency added by the transmission is guaranteed to not impact imager performance. The bound on the allowable pixel row transfer time t_{row_xfer} that guarantees successful transmission for the array as a whole can be expressed as

$$t_{row\_xfer} = \frac{\Delta T_{min} - t_{comp}}{N/2}, \tag{4.12}$$

where ΔT_{min} is the minimum time between two successive comparisons over the entire frame capture (see Section 3.4.3), t_{comp} is the time required to perform a full comparison, and $N/2$ is half the number of rows in the pixel array, since the rows are transmitted in pairs using dual `Col_Out` lines.

The overall time-division multiplexed transfer strategy is then to step through each pair of pixel rows in succession, allowing them t_{row_xfer} time to transmit their bits to the memory array. For design purposes, it is helpful to establish a target value for this row transfer time. Rather than explicitly solving Equation 4.12 based on parameter estimates, t_{row_xfer} should be designed to be as small as possible, as this will allow for a smaller ΔT_{min} and thus an increased pixel saturation limit. Based on the extracted parasitic resistance and capacitance of the `Col_Out` lines, a spike time constant of $\tau_{spike} = 1.5$ ns can be achieved with

$$R_{load} = 1.8 \text{ k}\Omega. \tag{4.13}$$

This value of R_{load} represents a compromise between the desire for fast response time and

large voltage swings on the columns, under the constraint that the spike generators are capable of sinking a little over 1 mA. Simulations of the resulting pixel-to-memory transfer path, with estimates of spike buffer and memory array delay included, show that

$$t_{row\_xfer} = 5 \text{ ns} \tag{4.14}$$

achieves robust data transfer, thus this value was chosen for the prototype design. Plugging this value into Equation 4.12, along with $t_{comp} \approx 300$ ns and $N = 150$, we can estimate the prototype will operate down to $\Delta T_{min} = 675$ ns.

A final caveat of the pixel-to-memory communication strategy is that rather than transmit the raw comparator result to the memory cell as is done in the generic dual-threshold pixel, a spike generator is included in each pixel to further encode the comparator information before it is sent. This block detects and generates a spike only at the transitions in the comparator output, based on the fact that the memory cell actually latches based only on these transition times. This step is equivalent to applying a discrete differentiator to the V_{o-} comparator output (see Figure 4-7), and takes advantage of the fact that, since the `Col_Out` lines are resistively tied to V_{DD}, it takes very little energy for the pixel to transmit a `HIGH` level, but considerable energy to transmit a `LOW` level. Because a typical frame capture sequence employs many hundreds of comparisons, during which the average pixel's comparator output will exercise both logic levels equally, this two-spike encoding scheme saves several orders of magnitude in communication energy compared to transmitting the raw comparator output.

4.3 Spike Buffering

The purpose of the spike buffering block is to terminate each of the pixel array's `Col_Out` lines in a high-impedance, low-capacitance load while providing a low-latency buffered version of the pixel spike data to the spike-gated SRAM array. As discussed in the previous section, minimizing the parasitic capacitance on the `Col_Out` lines allows the desired transient response time τ_{spike} to be achieved with the largest possible R_{load}, which in turn reduces the required current drive, and thus area, of each pixel spike generator. As illustrated in Figure 4-10, this block is implemented using $2M$ parallel digital buffers, which are globally enabled by the control signal `En_Spk_Buf`. When enabled, each buffer receives data

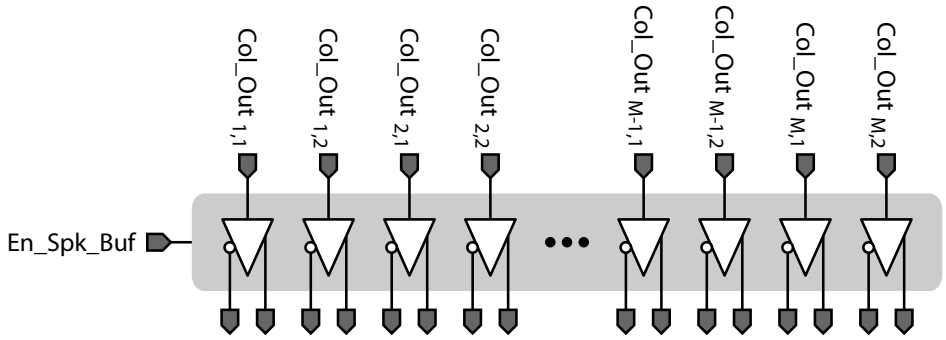

Figure 4-10: The spike buffering block consists of $2M$ parallel gated buffers.

En_Spk_Buf	Col_Out	Spk	$\overline{\texttt{Spk}}$
0	X	0	1
1	0	0	1
1	1	1	0

Table 4.2: Truth table for spike detection buffer operation.

from a single `Col_Out` line and outputs both a non-inverted `Spk` and inverted $\overline{\texttt{Spk}}$ version of it, as required by the spike-gated SRAM array (see Section 4.4). When disabled, the buffer outputs default to `Spk` = `LOW` and $\overline{\texttt{Spk}}$ = `HIGH`, regardless of the state of `Col_Out` (again, see Section 4.4). This behavior is summarized in Table 4.2.

The internal structure of a single gated buffer is shown in Figure 4-11. Transistors $M_{1\text{-}4}$ form a tri-state inverter that actively drives node $\texttt{S}_{\texttt{int}}$ when `En_Spk_Buf` is `HIGH`, while M_5 forces $\texttt{S}_{\texttt{int}}$ to be `HIGH` when `En_Spk_Buf` is `LOW`. The three inverters employ minimum-size NMOS devices and 3× PMOS devices to achieve balanced rise and fall times. The $1 \rightarrow 25$ inverting output buffer consists of a three-inverter chain, progressively sized $2\times$ -$7\times$ -25× relative to a minimum inverter. The size of the final stage is chosen relative to the parasitic capacitance the buffer must drive in the spike-gated SRAM array, while the geometric interstage growth of roughly 3.5× provides near-minimum delay through it [76]. The non-inverting version of the output buffer includes a fourth, minimum-size inverter at the front of this three-inverter chain. Finally, the layout of the overall spike-detection buffer was pitch-matched to $\frac{1}{2}$ the pixel pitch, or 6.25 μm, since two of them are required per pixel column.

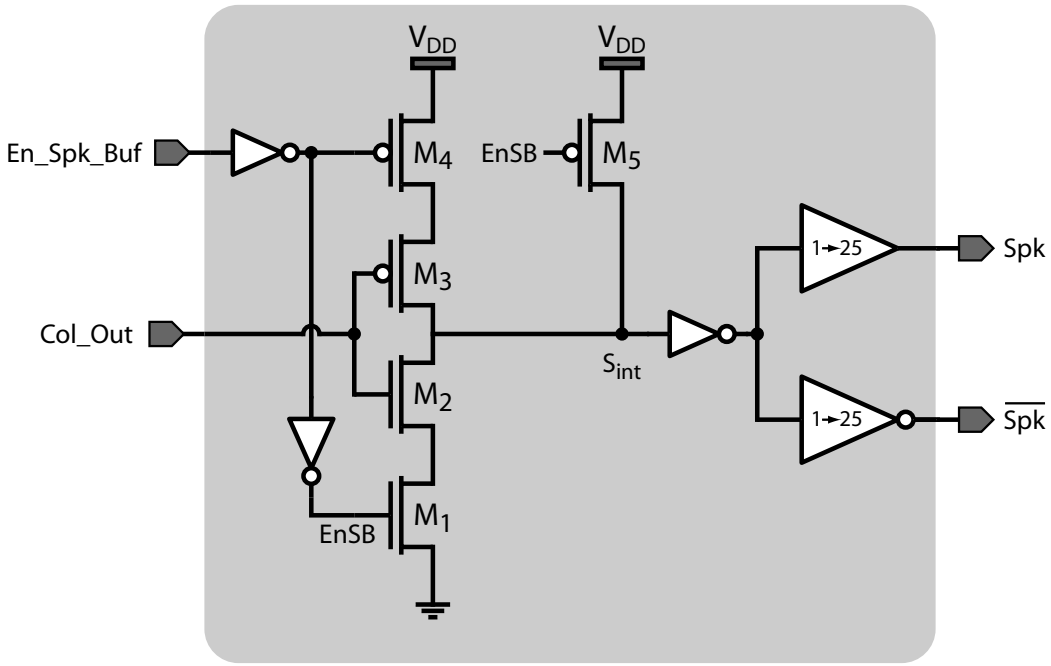

Figure 4-11: Schematic of a single spike detection buffer.

4.4 Spike-Gated SRAM Array

As explained in Section 4.2, after each comparison, any pixel that has detected a threshold crossing generates a spike that is synchronously transmitted to the pixel's dedicated memory cell in the spike-gated SRAM array. Upon detecting the arrival of one of these spikes, the job of the memory cell is to record the current state of a global, time-varying digital code. Since non-saturated pixels will generate two spikes per image frame, the memory cells must be able to record both an offset and irradiance value for their pixel, and they must also be capable of driving these recorded values onto a shared bus, transferring them to the periphery of the array to be read and transmitted off chip. This section discusses the circuits that were used to implement this functionality.

4.4.1 Six-Transistor SRAM Cell

Both subthreshold leakage as well as junction leakage due to optically-generated carriers pose potential problems for any form of dynamic memory element in this application. Thus, although employing a three-transistor (3T) DRAM cell would yield a more compact array,

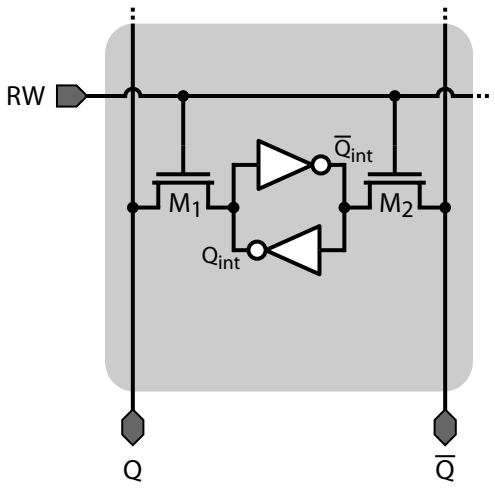

Figure 4-12: Schematic of a six-transistor (6T) SRAM cell.

the six-transistor (6T) SRAM cell is the prudent choice for the unit memory element due to its inherent leakage current rejection. The basic 6T SRAM structure, shown in Figure 4-12, consists of an internal bistable latch formed from two cross-coupled inverters and two access transistors M_1 and M_2. Cell operation can be divided into write and read phases, as explained below.

- **Write Phase:** To write a `0` (`1`) to the cell, `Q` is externally driven `LOW` (`HIGH`) while $\overline{\texttt{Q}}$ is driven `HIGH` (`LOW`). Next, `RW` is pulsed `HIGH`, turning M_1 and M_2 `ON` and pulling the internal latch nodes $\texttt{Q}_{\texttt{int}}$ and $\overline{\texttt{Q}}_{\texttt{int}}$ towards the state established on the bit lines. In order for the latch to successfully transition to the desired state, it is critical that the two access transistors are strong enough to overpower the inverter PMOS transistors and pull the latch's internally `HIGH` node below the inverters' threshold level.

- **Read Phase:** To read the cell state, both bit lines `Q` and $\overline{\texttt{Q}}$ are pre-charged `HIGH` and then enter a tri-state mode. Next, `RW` is pulsed `HIGH` for a pre-determined discharge period, connecting the bit lines to $\texttt{Q}_{\texttt{int}}$ and $\overline{\texttt{Q}}_{\texttt{int}}$ through M_1 and M_2. Since one internal latch node is `LOW` while the other is `HIGH`, one bit line will discharge significantly during this interval while the other will not. To avoid corrupting the cell's stored state, it is critical that the access transistors are not so strong that they pull the latch's internally `LOW` node above the inverters' threshold level. Once `RW` returns `LOW`, ending the discharge interval, an external cross-coupled latch (see Section 4.8.4) is used to

Q	$\overline{\text{Q}}$	$\text{Q}_{\text{int,n}}$	$\overline{\text{Q}}_{\text{int,n}}$	$\text{Q}_{\text{int,n+1}}$	$\overline{\text{Q}}_{\text{int,n+1}}$
0	0	X	X	**?**	**?**
0	1	X	X	0	1
1	0	X	X	1	0
1	1	X	X	$\text{Q}_{\text{int,n}}$	$\overline{\text{Q}}_{\text{int,n}}$

Table 4.3: Write-phase state evolution of six-transistor SRAM cell.

amplify the resulting differential bit line voltage into a full-scale digital result, which is accessible at the array periphery.

An additional interesting property of the 6T SRAM cell's write behavior can be inferred from the read phase operation described above. Since the cell is designed to ensure that its stored state is not inadvertently altered by a read operation, during which both Q and $\overline{\text{Q}}$ are pre-charged HIGH, driving both of these bit lines HIGH during a write operation should also leave the cell state unchanged. Conversely, since M_1 and M_2 can both overpower the internal inverters' PMOS transistors, driving both bit lines LOW during a write operation yields an unpredictable result. Thus, a full description of the 6T SRAM cell's write-phase state evolution as a function of the bit lines Q and $\overline{\text{Q}}$ is as described in Table 4.3. This ability of the 6T SRAM cell to retain its initial state across a write operation will prove useful in the design of the spike-gated memory cell in the next section.

A final important aspect of the SRAM cell is the amount of area it consumes. Using minimum-size NMOS and PMOS devices in the inverter, it is possible to size M_1 and M_2 very close to minimum-size and achieve the operation described above over all process corners. The resulting SRAM cell measures 2.2 μm wide by 3.26 μm tall.

4.4.2 Spike-Gated Memory Cell

Due to delays introduced by the Col_Out lines and the spike buffers, a several nanosecond delay $t_{spk\_dly,on}$ exists between the generation of a spike at a pixel and the arrival of the spike at the pixel's corresponding memory cell. A similar delay $t_{spk\_dly,off}$ occurs between spike termination and the departure of the spike from the memory cell. To robustly reject this delay, the spike-gated SRAM cell shown in Figure 4-13 was employed. The cell consists of three main blocks, with 80% of its area dedicated to the 18-bit SRAM register that is used to store the pixel offset and illuminance data. The RW input to this data register is

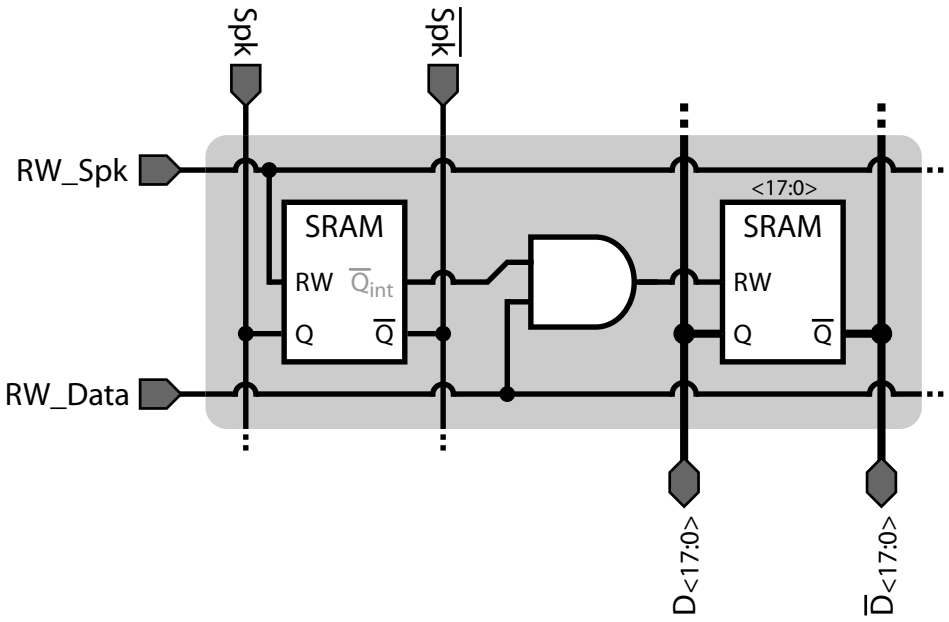

Figure 4-13: Schematic of the spike-gated memory cell.

derived by AND-ing the row-level control signal `RW_Data` with the $\overline{\mathtt{Q}}_{\mathtt{int}}$ node of the final memory cell component, a single SRAM bit. The state of this SRAM bit is sampled from `Spk` and $\overline{\mathtt{Spk}}$ when the row-level `RW_Spk` signal is pulsed `HIGH`.

To understand how this memory cell operates in concert with the overall pixel-to-memory communication strategy, refer to the test circuit and timing waveforms shown in Figure 4-14. The test circuit contains a single pixel column, with its associated load resistor and spike buffer, connected to a single spike-gated memory cell, implemented as shown in Figure 4-13. Together, these components form a bare-bones representation of the pixel-to-memory communication chain. During the following discussion, assume the pixel shown is the only one in the column prepared to transmit a spike and that the column spike buffer is enabled, though the gating signal `En_Spk_Buf` is not explicitly shown. Spike transmission commences when the row-level control signal $\mathtt{Read}_{\mathtt{i}}$ goes `LOW`, enabling the pixel's spike generator to sink current through R_{load}. The RC time-constant dictated by R_{load} and the parasitic resistance and capacitance of `Col_out` yields an approximately exponential decay on this line and, along with the spike buffer, introduces the delay $t_{spk\_dly,on}$. Since no other pixels in the column are prepared to transmit a spike, when $\mathtt{Read}_{\mathtt{i}}$ goes `HIGH` and $\mathtt{Read}_{\mathtt{i+1}}$ goes `LOW` after a transfer window of width t_{row_xfer}, `Col_Out` will exhibit a comple-

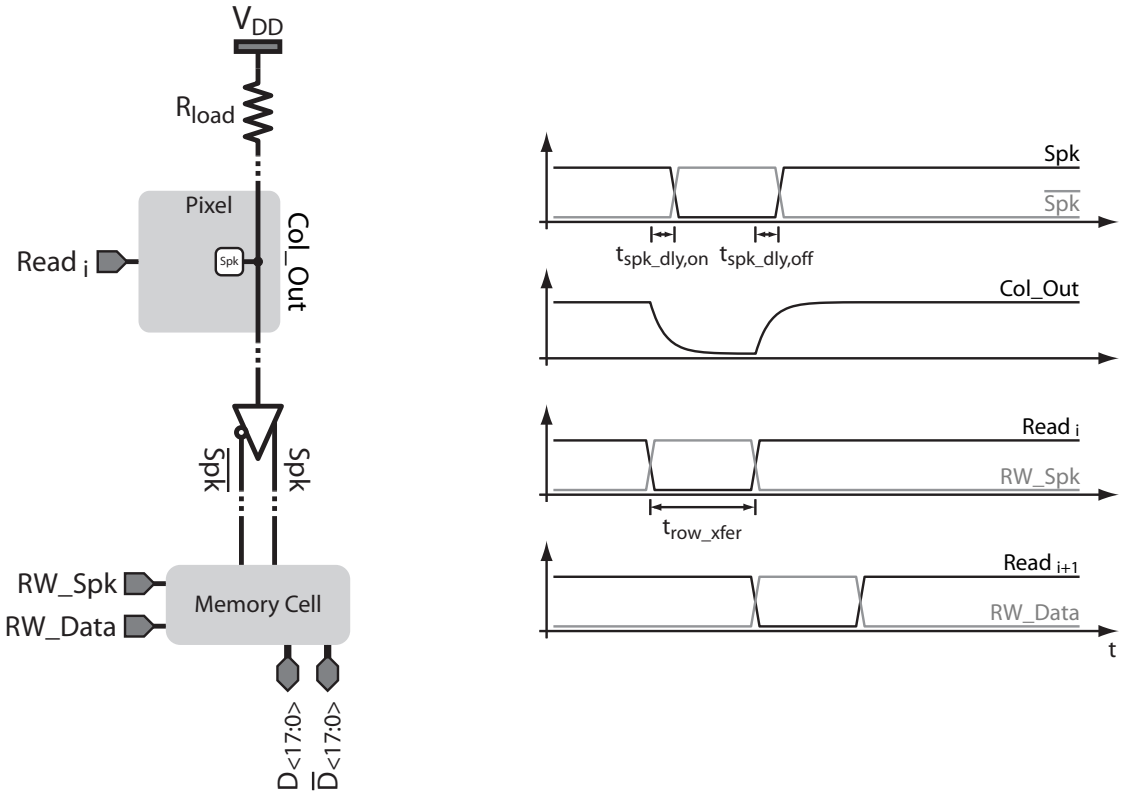

Figure 4-14: Illustration of the spike-gated memory cell operation.

mentary exponentially decaying rise back to V_{DD} and a similar delay $t_{spk\_dly,off}$ will occur. Typically, $t_{spk\_dly,off}$ and $t_{spk\_dly,on}$ are not equal, thus the memory cell must wait until after the longer of these two parasitic delays for the values of `Spk` and $\overline{\texttt{Spk}}$ to become valid. This is accomplished using the memory cell's spike-sampling SRAM bit, which if clocked with $\texttt{RW\_Spk} = \overline{\texttt{Read}_\texttt{i}}$, will sample the values of `Spk` and $\overline{\texttt{Spk}}$ at the end of the spike transfer window. Notice that some overlap between the edges of $\texttt{Read}_\texttt{i}$, $\texttt{Read}_{\texttt{i+1}}$, and `RW_Spk` can be tolerated in this sampling scheme because the parasitic delay $t_{spk\_dly,off}$ ensures that `Spk` and $\overline{\texttt{Spk}}$ remain valid, simplifying the design of the time-domain multiplexing controller discussed in Section 4.5.

Once the pixel spike status has been stored in the memory cell's spike-sampling SRAM bit, its value is used to gate the SRAM register's `RW_Data` signal. In the case where a spike was detected, $\overline{\texttt{Q}_\texttt{int}}$ will be `HIGH` and the output of the AND gate will track `RW_Data` as it pulses `HIGH`, latching the current state of the $\texttt{D}_{<17:0>}/\overline{\texttt{D}}_{<17:0>}$ bus in the 18-bit SRAM register. On the other hand, if no spike was detected, the output of the AND gate will

remain LOW and no data will be sampled. In the prototype imager it was convenient to choose $\texttt{RW\_Data} = \overline{\texttt{Read}}_{\texttt{i+1}}$, thus memory cells that detect a spike from a pixel in row pair i will sample the data bus at the same time that pixels in row pair $i+1$ are transmitting spike information.

A very useful feature of this memory cell topology is that it allows multiple spikes to trigger data storage to the SRAM register without corrupting the data stored by previous spikes. This feature is based upon the ability of the 6T SRAM cell to either update or hold its internal state during a write operation, as shown in Table 4.3. Thus, by properly encoding the data on the bus lines $\texttt{D}_{<17:0>}$ and $\overline{\texttt{D}}_{<17:0>}$, arbitrary bits within the memory cell's SRAM register can be either updated or held constant in response to a spike arrival.

Once an entire frame of image data has been recorded, the remaining job of each memory cell is to transmit its stored pixel data to the periphery of the memory array. A read of the SRAM register proceeds exactly as described in Section 4.4.1, and is controlled by the array-level circuitry described in Section 4.8.4. The only added complication to this procedure is the fact that the SRAM register's RW signal is not directly accessible from outside the memory cell. However, as mentioned in Section 4.3, the default output state the spike buffers when En_Spk_Buf is LOW, which is always the case except during spike transmission, is Spk = LOW and $\overline{\texttt{Spk}}$ = HIGH. This set of bit line states allows RW_Spk and RW_Data to indirectly control each SRAM register's RW signal.

The width of the memory cell layout was pitch-matched to 4× the pixel pitch, or 50 μm. This was necessary in order to allow the 18 SRAM register bit cells to be arrayed horizontally, which in turn enabled the register's $18 \cdot 2 = 36$ bit lines to be routed vertically through the cell on a single metal layer. The remaining width of the cell was used for the spike-sampling SRAM bit, the AND gate, and vertical routing of power, ground, and spike lines. The overall memory cell measures 50 μm wide by 3.26 μm tall, making it a little over 4% larger than a pixel.

4.4.3 Memory Cell Array Design

Similar to how the unit pixel layout dictated the overall pixel array structure, the form of the memory cell array is pre-determined by the unit memory cell layout. Therefore, as in the case of the unit pixel, array-level considerations influenced many of the decisions made during the design of the unit memory cell. One such decision, the use of a 4 : 1 memory cell

to pixel width ratio, simplified the memory cell bus layout, but complicates the physical mapping between these cells and their associated pixels. To understand how this issue was resolved, refer to the strategy depicted in Figure 4-15, which focuses on a 2×4 block of pixels, an 8×1 block of memory cells, and a 1×8 block of spike buffers. The connections between the pixels and spike buffers are familiar from earlier, but the introduction of the 8-spike bus along the memory cell column is new. Though it wasn't mentioned in the previous section, this spike bus is actually already routed vertically through the unit memory cell layout, allowing spike information from all eight spike buffers to reach any memory cell in the entire column. The manner in which the individual memory cells within a single 8×1 block tap off their `Spk` and $\overline{\texttt{Spk}}$ inputs from this bus, and therefore map to each pixel in the 2×4 pixel block, is represented by the internal numbering system shown in the figure. A similar mapping can be imagined for each pair of memory cell and pixel blocks in this super-column[4], where all pixel blocks share the same set of eight `Col_Out` lines and all memory cell blocks share the same 8-spike bus. The overall imager can then be viewed as being comprised of multiple copies of this super-column arrayed horizontally.

Finally, it is possible to develop a mathematical representation of the connection strategy presented above. Assuming pixel row and column numbering begins in the bottom left corner of the pixel array and that memory cell row and column numbering begins in the top left corner of the memory array, the correspondence between each pixel/memory cell pair can be represented through the following set of relations

$$\begin{gathered} Pixel_{i,j} \leftrightarrow MemoryCell_{u,v} \\ u = 4\big[i - 2[(i+1) \bmod 2]\big] + \big[[(j-1) \bmod 4] + 1\big], \quad v = \lceil \tfrac{j}{4} \rceil \\ i = (\lceil \tfrac{u}{4} \rceil + 1) - 2[(\lceil \tfrac{u}{4} \rceil + 1) \bmod 2], \quad j = 4(v-1) + [(u-1) \bmod 4] + 1 \end{gathered} \tag{4.15}$$

where $i \in \{1, 2, \ldots, N\}$, $j \in \{1, 2, \ldots, M\}$, $u \in \{1, 2, \ldots, 4N\}$, and $v \in \{1, 2, \ldots, \lceil \frac{M}{4} \rceil\}$.

4.5 Spike Transfer Controller

It has been stated previously that the pixel-to-memory spike transfer occurs in a time-domain multiplexed manner, with each pair of pixel rows allotted a window of length t_{row_xfer} within which to transmit spike information to their corresponding spike-gated

[4]The term super-column is used since multiple pixel columns are contained within it.

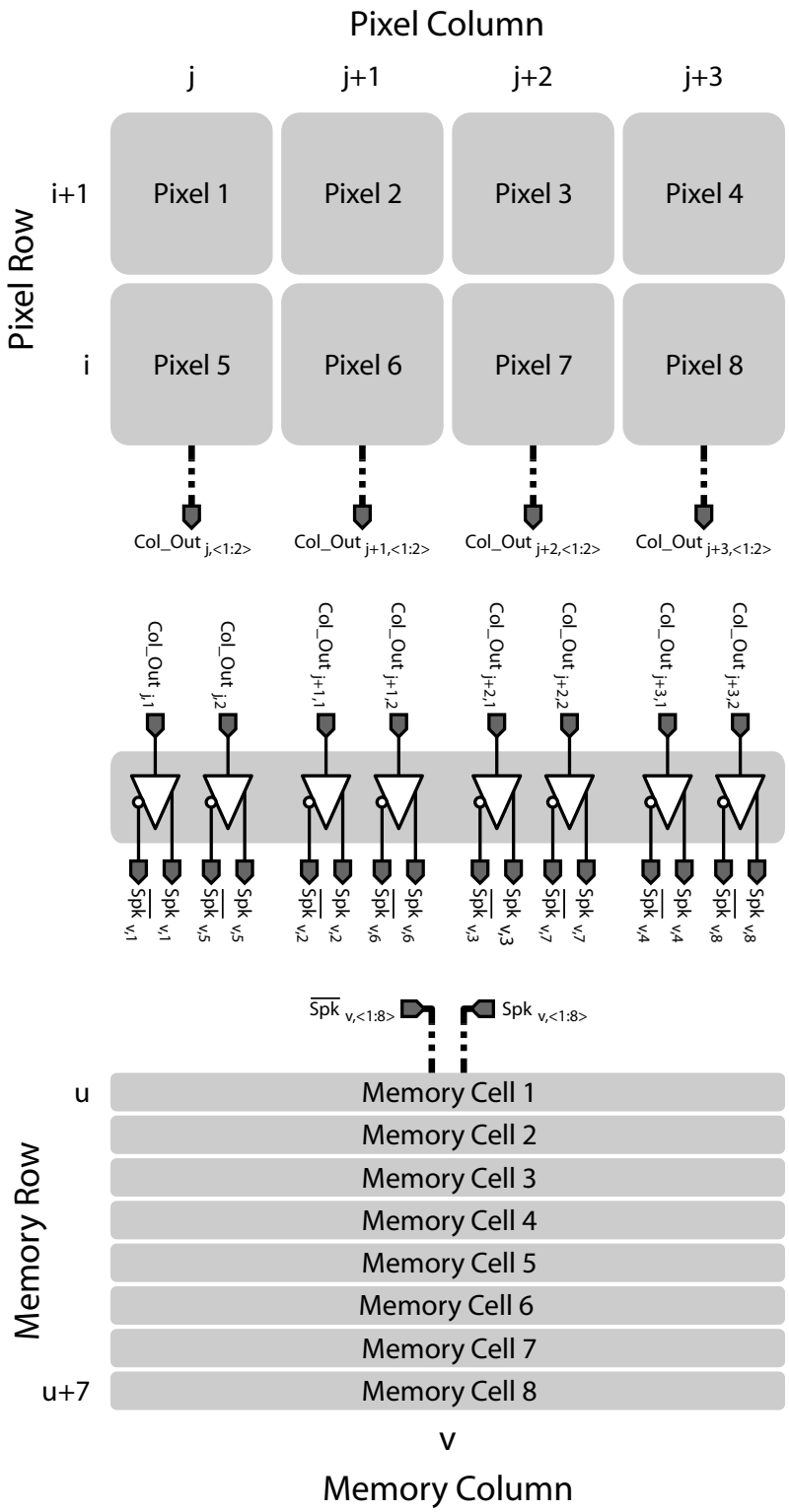

Figure 4-15: Mapping between pixels and their associated memory cells.

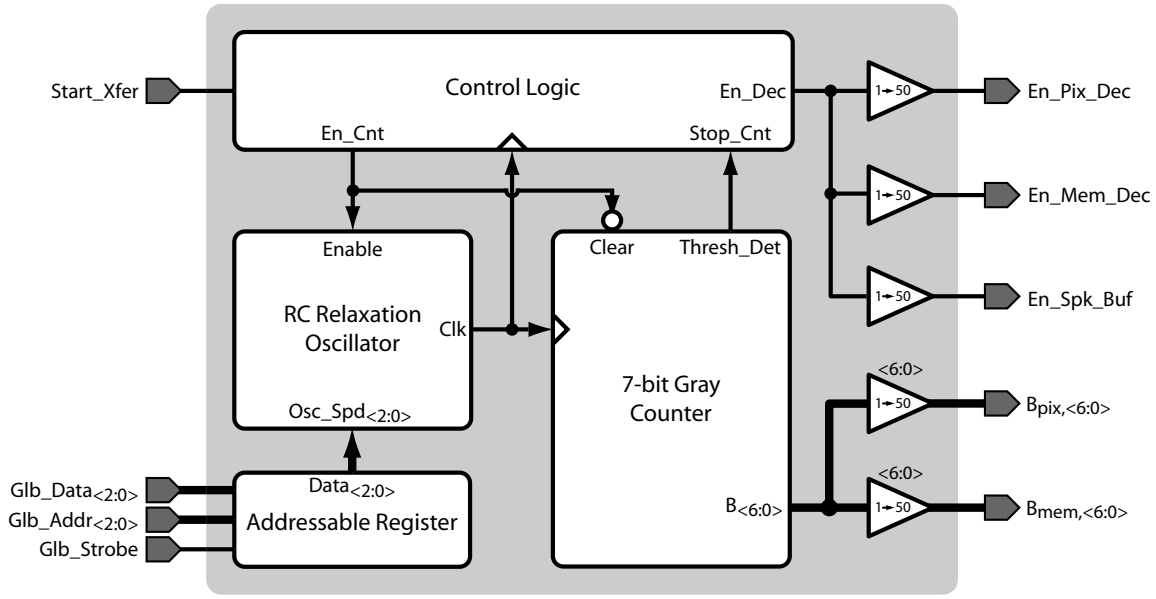

Figure 4-16: Block diagram of spike transfer controller.

memory cells. The spike transfer controller, working in concert with the row decoders and buffers shown in Figure 4-1, coordinates this communication by generating an incrementing 7-bit Gray code, starting at `0000000` and incrementing upwards to a maximum of `1101011`[5], with a transition period of t_{row_xfer}. This Gray code is transmitted in parallel to the row decoders, which use it to synchronize spike transmission and reception between corresponding sets of pixel and memory cell rows. Since Gray code transitions involve only a single bit flip, using this coding scheme in the transfer controller, along with simple combinational logic in the row decoders, guarantees glitch-free control of the spike communication. The details of the spike transfer controller are discussed in this section, while the row decoder details are addressed in the next sections.

A block diagram of the spike transfer controller is shown in Figure 4-16. The controller comprises a tunable relaxation oscillator, a 3-bit register to adjust the oscillator period, a 7-bit Gray counter, some control logic, and several output buffers. Together, these blocks implement the behavior illustrated by the timing waveforms in Figure 4-17. In examining these waveforms, assume the controller begins in its rest state prior to the arrival of the pulse on the off-chip input signal `Strt_Xfer`, which is equivalent to the signal `S` shown in Figure 3-10. This rest state is characterized by `LOW` values for both `En_Dec` and `En_Cnt`, the

[5]This maximum value is a function of the number of pixel rows N, as explained in Section 4.7.

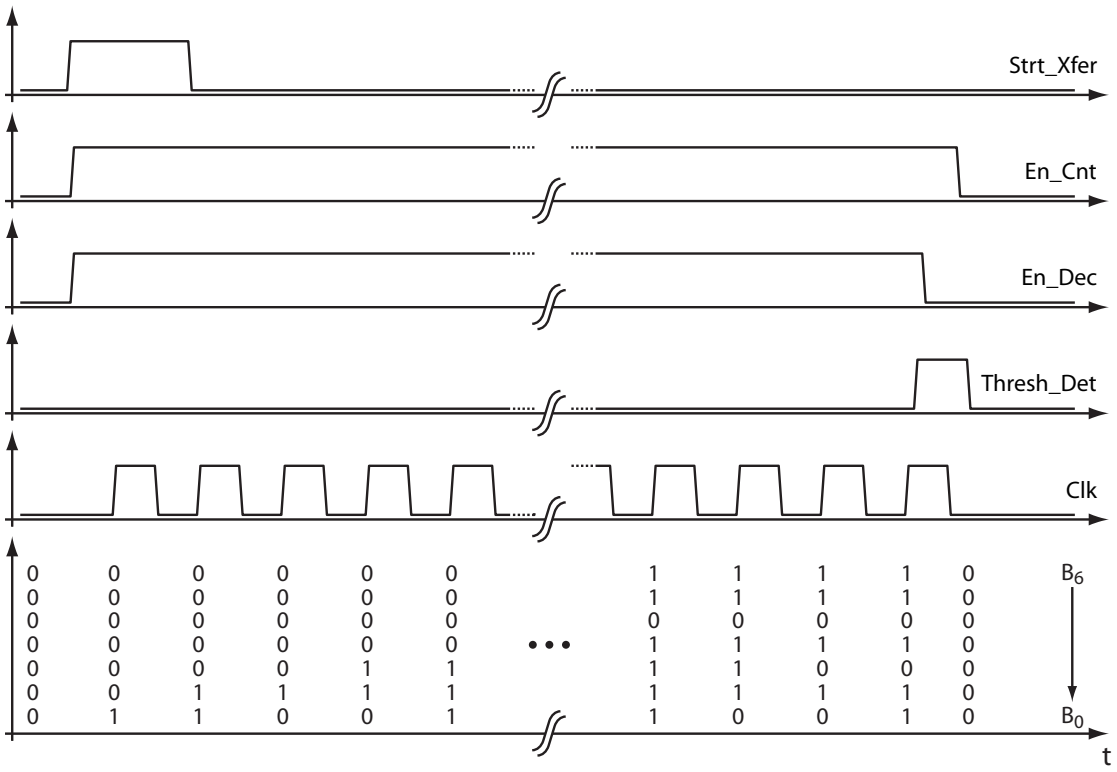

Figure 4-17: Waveforms illustrating spike transfer controller operation.

latter of which disables the oscillator, forcing `Clk` to sit at a `LOW` level, and activates the low-true `Clear` input to the Gray counter, forcing the counter state to `0000000`. This rest state prevails until a pulse arrives on `Strt_Xfer`, causing both `En_Cnt` and `En_Dec` to go `HIGH`, enabling the oscillator and disabling the counter's low-true `Clear` input. After a short delay, the first rising edge of `Clk` arrives, incrementing the Gray counter output from `0000000` to `0000001`, with succeeding positive edges occurring at a nominal period of t_{row_xfer} and incrementing the output up through `1101011`. Upon reaching this maximum level, the counter raises `Thres_Det`, signalling to the control logic that the terminal code has been reached. Within the control logic block, a `HIGH` level on `Stop_Cnt` immediately resets `En_Dec`, and at the following *falling* edge of `Clk` also resets `En_Cnt`, returning the controller to its initial rest state until the next pulse on `Strt_Xfer`. The purpose of resetting `En_Dec` before `En_Cnt` is to disable the pixel and memory row decoders before the Gray counter undergoes the non-Gray transition from `1101011` to `0000000`, thus avoiding potential glitches in the decoder outputs at this transition. The internal structure of the various blocks composing this controller are discussed in the following subsections.

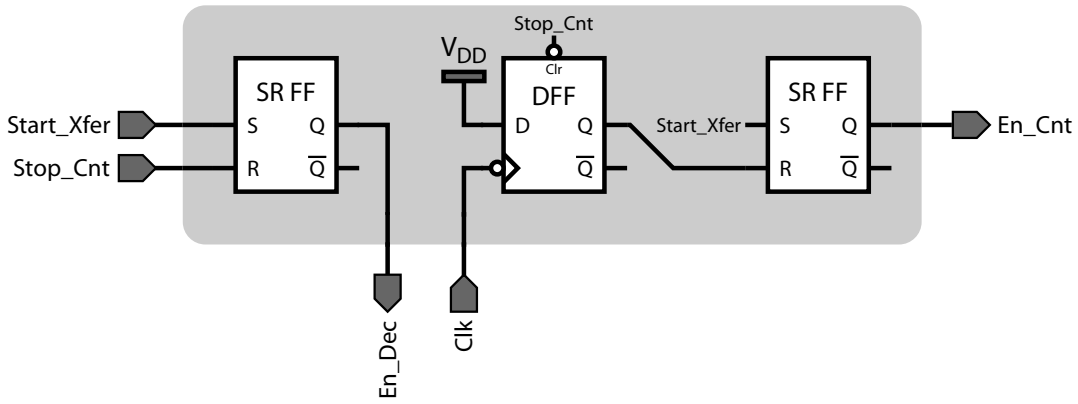

Figure 4-18: Spike transfer control logic block.

4.5.1 Control Logic and Output Buffers

The control logic and output buffers are fairly self-explanatory, and therefore their details will be outlined first. The control logic block is built using two SR flip-flops (SRFFs) and one D flip-flop (DFF), which are connected as shown in Figure 4-18. It can be easily verified that this circuit implements the functionality described above for the control logic block. Each of the $1 \rightarrow 50$ output buffers is constructed with a scaled inverter chain sized $1\times$ -$4\times$ -$14\times$ -$50\times$ relative to a minimum inverter. The buffer outputs drive bus lines within the pixel and memory row decoders, as well as the spike buffers.

4.5.2 Addressable Register

Several components within the imager need the ability to exchange digital information with an off-chip entity. In some cases, for example the SRAM array, data must be able to flow both into and out (I/O) of the chip, while in others data is only sent in, typically to fine-tune some analog parameter that can vary across process. This is the case with the spike transfer controller, which incorporates a local 3-bit register to tune the relaxation oscillator period to the nominal value of $t_{row\_xfer} = 5ns$. To allow every such block that requires it access to the external world while keeping the imager pin count manageable, a single 18-bit I/O bus was provided. Internally, this bus is divided into separate uni-directional input and output busses, $\texttt{Glb\_Data}_{<17:0>}$ and $\texttt{Glb\_Out}_{<17:0>}$, respectively. While $\texttt{Glb\_Out}_{<17:0>}$ only connects to the SRAM (see Section 4.8), $\texttt{Glb\_Data}_{<17:0>}$ is routed to six blocks across the imager. To manage access to this input bus, a copy of the generic K-bit addressable register shown

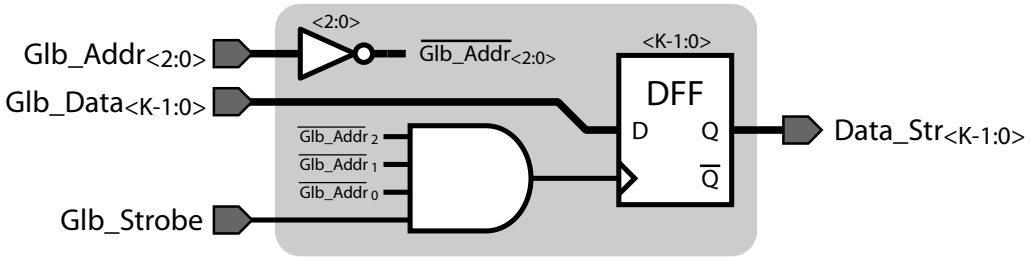

Figure 4-19: Generic K-bit addressable register, wired for global address `000`.

in Figure 4-19 was incorporated into each of these blocks. Each register connects to the lower K bits of the global data bus, as well as a 3-bit address bus `Glb_Addr`$_{<2:0>}$ and strobe signal `Glb_Strobe`. Three inputs of a 4-input AND gate are hard-wired to the selected register address, with inverted copies of the address bits generated locally as needed. The fourth input is connected to `Glb_Strobe`, which is used to clock the register's K parallel D flip-flops once the address and data busses are valid. In the case of the spike transfer controller, $K = 3$ and the register address was chosen to be `000`.

4.5.3 RC Relaxation Oscillator

The gated RC relaxation oscillator topology shown in Figure 4-20 consists of an inverting string of digital gates coupled with a tunable RC delay in a negative feedback loop. The use of a NAND gate in the loop allows the external `Enable` signal to gate the feedback, and thus the oscillator, `ON` and `OFF`. A set of nominal operating waveforms illustrating this gating capability are shown in Figure 4-21. When `Enable` is `LOW`, the loop nodes V_{res}, V_{RC}, and V_b all settle to `HIGH` levels, thus `Clk` is `LOW` and the oscillator is disabled. Since V_b rests at a `HIGH` level, when `Enable` does eventually go `HIGH`, V_{res} quickly transitions `LOW` and causes V_{RC} to begin an exponential decay towards GND. As V_{RC} reaches the input threshold voltage of the $1 \rightarrow 3$ digital buffer, represented by the light-grey line on the V_{RC} waveform, V_b will begin to decrease. Due to the high gain of this buffer around threshold, even small movements in V_b are quickly amplified via the positive feedback loop formed by the buffer and the C_1/C_2 capacitive divider, which together implement a dynamic Schmitt trigger. A fraction $f = \frac{C_1}{C_1+C_2}$ of the resulting rail-to-rail transition in V_b couples back into V_{RC}, pulling this node past and well below the buffer threshold level. However, since

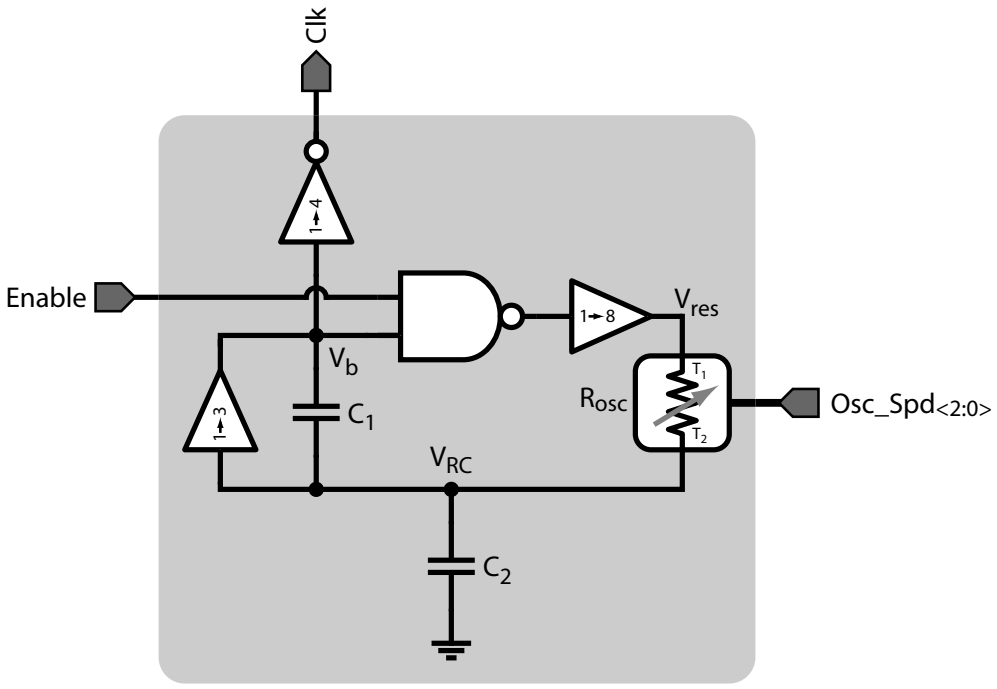

Figure 4-20: Gated relaxation oscillator with tunable period.

this transition also leads to a change in the polarity of V_{res}, V_{RC} begins an exponentially decaying rise back towards V_{DD}, producing a similar cycle of opposite polarity.

The oscillations that result from the behavior described above exhibit an approximate period of

$$T_{osc} = \tau \ln \left[\frac{(\alpha + f)(\alpha - (1 + f))}{\alpha(\alpha - 1)} \right] + t_{delay} \tag{4.16}$$

where $\tau = R_{osc}(C_1 + C_2)$, αV_{DD} is the input threshold voltage of the $1 \rightarrow 3$ buffer, and t_{delay} is the net delay through the loop's digital gates. The $1 \rightarrow 4$ inverting and $1 \rightarrow 8$ non-inverting buffers were implemented with scaled inverter chains sized $1\times$ -$1\times$ -$4\times$ and $1\times$ -$1\times$ -$4\times$ -$8\times$ relative to a minimum inverter, respectively. Since V_{RC} is more analog in nature, the $1 \rightarrow 3$ buffer's input inverter was implemented with increased-length devices to limit its shoot-through current and its P-to-N sizing ratio was chosen to achieve $\alpha = \frac{1}{2}$, while the output stage was a simple $3\times$ minimum inverter. Based on simulations with these components, setting $C_1 = C_2 = 20$ fF and $R_{osc} = 65$ kΩ yields a nominal oscillation period very close to the desired $t_{row\_xfer} = 5$ ns.

Finally, to counteract potential perturbations in this nominal oscillation period due to process variation, R_{osc} was implemented using the 3-bit resistor DAC shown in Figure 4-22.

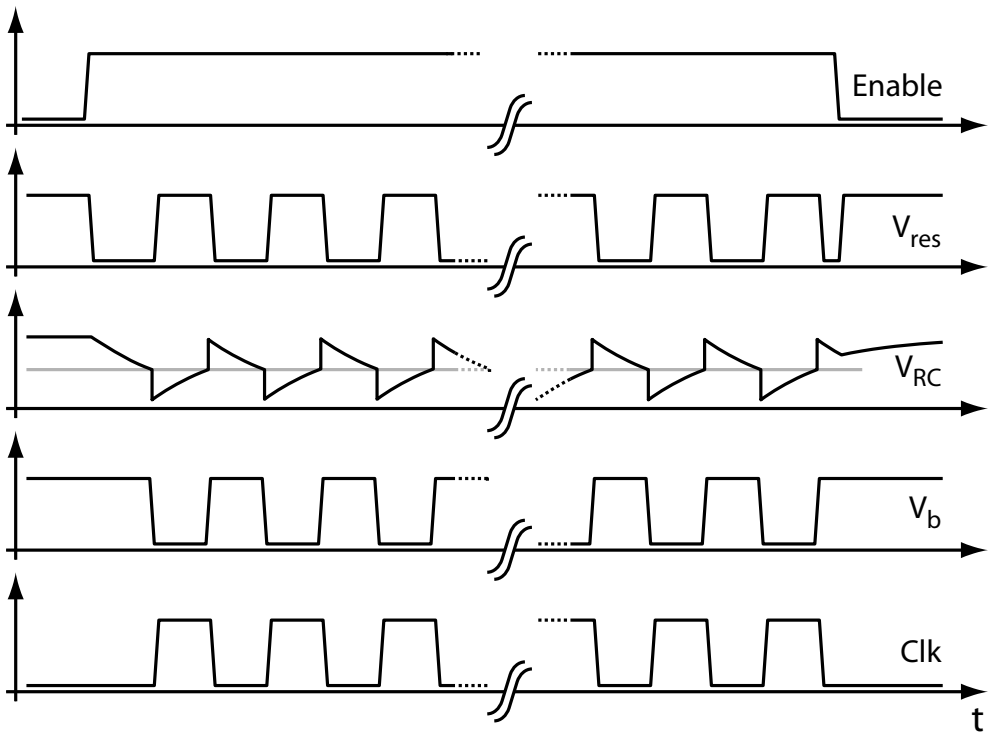

Figure 4-21: Waveforms illustrating gated relaxation oscillator operation.

The resistance between T_1 and T_2 is a function of the digital input $B_{<2:0>}$, and can be varied from approximately 5 k$\Omega + 10 \cdot R_{unit}$ when the input is `000` to 5 k$\Omega + 3 \cdot R_{unit}$ when the input is `111`. Choosing $R_{unit} = 10$ kΩ achieves the desired nominal $R_{osc} = 65$ kΩ with a mid-range input of `100` and gives a tuning range of roughly $\pm 50\%$, which is large enough to allow worst-case resistance, capacitance, and gate delay variations over process to be nulled.

4.5.4 7-bit Gray Counter

An N-bit Gray code sequence traverses each of the 2^N possible N-bit codes while requiring only a single bit flip to transition between neighboring codes. It is possible to construct such a sequence recursively, as will be illustrated with the help of Table 4.4. Focusing on only columns $\mathtt{B}_{<3:0>}$ for the moment, notice that toggling a single bit naturally creates a 1-bit Gray code matrix $\mathbf{G_1} = [\,0\;1\,]^T$, which forms the 2×1 block above the horizontal line spanning column $\mathtt{B}_0$. Next, to generate $\mathbf{G_2}$, mirror a copy of $\mathbf{G_1}$ about this horizontal line, and to the left of the resulting 4×1 block construct a new column with the first 2^1 rows

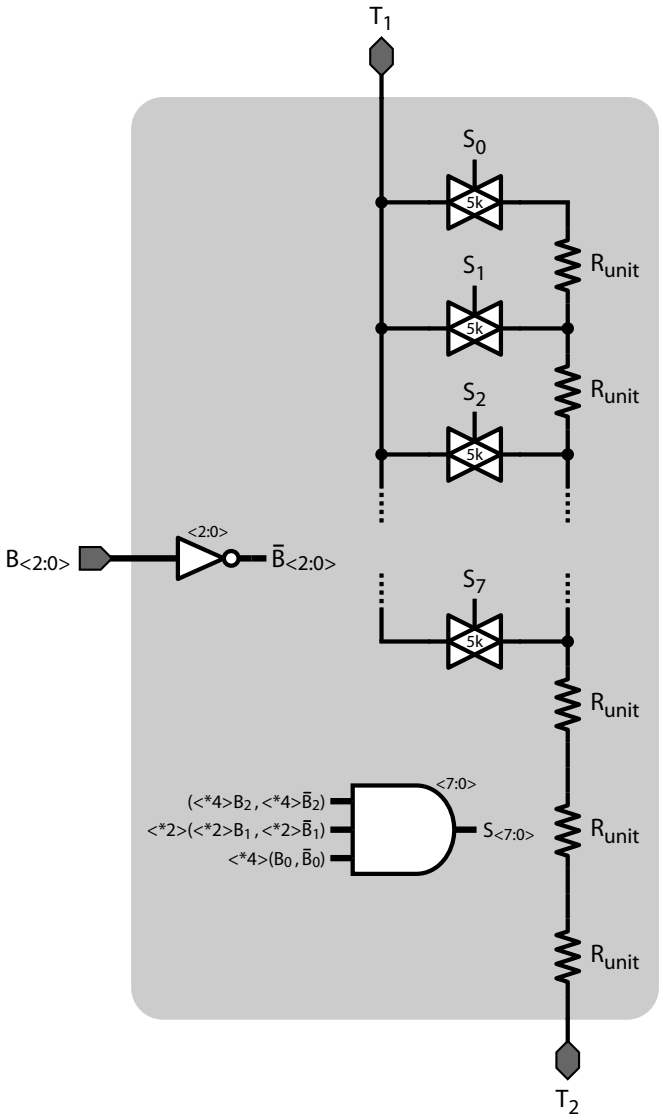

Figure 4-22: 3-bit resistor DAC used in relaxation oscillator.

set to `0` and the second 2^1 rows set to `1`. The resulting 4×2 block, which lies above the horizontal line spanning columns $\mathtt{B}_{<1:0>}$, is $\mathbf{G_2} = \left[\begin{smallmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{smallmatrix}\right]^T$. Following a similar procedure, the N-bit Gray code matrix $\mathbf{G_N}$ can be generated by mirroring $\mathbf{G_{N-1}}$ about a horizontal line of symmetry and adding a column to the left of the resulting $2^N \times (N-1)$ block in which the first 2^{N-1} rows are set to `0` and the latter 2^{N-1} rows are set to `1`.

To gain insight into how a hardware Gray counter might be implemented, notice that the recursive algorithm described above ensures that for $K > 0$, every transition in $\mathtt{B}_K$ occurs during the same state of $\mathtt{B}_{<K-1:0>}$, regardless of bits $\mathtt{B}_{K+1}$ and above. However, since

B_3	B_2	B_1	B_0	B_{-1}
0	0	0	0	1
0	0	0	1	0
0	0	1	1	1
0	0	1	0	0
0	1	1	0	1
0	1	1	1	0
0	1	0	1	1
0	1	0	0	0
1	1	0	0	1
1	1	0	1	0
1	1	1	1	1
1	1	1	0	0
1	0	1	0	1
1	0	1	1	0
1	0	0	1	1
1	0	0	0	0

Table 4.4: Columns $B_{<3:0>}$ form the 4-bit Gray code matrix $\mathbf{G_4}$ with horizontal lines marking the recursive mirroring steps. Column B_{-1} is an additional bit used in the hardware counter implementation.

$B_{<K-1:0>}$ remains fixed across transitions in B_K, an ambiguity exists in trying to determine transitions in B_K based on $B_{<K-1:0>}$ alone. An additional complication arises due to the fact that no bits lie below B_0 to define its transitions. Fortunately, both issues can be resolved with the addition of a single bit B_{-1}, which toggles on every code transition. The initial state of B_{-1} is not important as long as it is accounted for in the overall design, and it was arbitrarily chosen to start at 1, as shown in Table 4.4. With the addition of this bit, and extrapolating from the 4-bit to the 7-bit case, a set of signals can be defined which govern

the transitions of each bit in the Gray counter as follows

$$
\begin{aligned}
T_{B_6} &= B_5 \cdot \overline{B}_4 \cdot \overline{B}_3 \cdot \overline{B}_2 \cdot \overline{B}_1 \cdot \overline{B}_0 \cdot \overline{B}_{-1} \\
T_{B_5} &= B_4 \cdot \overline{B}_3 \cdot \overline{B}_2 \cdot \overline{B}_1 \cdot \overline{B}_0 \cdot \overline{B}_{-1} \\
T_{B_4} &= B_3 \cdot \overline{B}_2 \cdot \overline{B}_1 \cdot \overline{B}_0 \cdot \overline{B}_{-1} \\
T_{B_3} &= B_2 \cdot \overline{B}_1 \cdot \overline{B}_0 \cdot \overline{B}_{-1} \\
T_{B_2} &= B_1 \cdot \overline{B}_0 \cdot \overline{B}_{-1} \\
T_{B_1} &= B_0 \cdot \overline{B}_{-1} \\
T_{B_0} &= B_{-1}
\end{aligned}
\tag{4.17}
$$

where B_K transitions only if its corresponding T_{B_K} is a 1. Combining these transition signals with a set of eight toggle flip-flops (TFFs) yields the 7-bit Gray counter implementation shown in Figure 4-23. Note that the TFF associated with B_{-1} is preset by the Clear signal and is clocked on negative edges to avoid any potential race conditions, and that the code 1101011 generates the Thresh_Det signal.

4.6 Pixel Row Decoder and Buffers

Both the pixel and memory row decoders receive buffered copies of the En_Dec and $B_{<6:0>}$ control signals from the spike transfer controller and, using identical arrays of combinational logic, decipher these inputs to coordinate the pixel-to-memory spike communication. As shown in the lower portion of Figure 4-24, the pixel row decoder is constructed using 75 parallel 8-input AND gates, each of which is hard-wired to output a HIGH level only when En_Pix_Dec is HIGH and a particular 7-bit Gray code is present on $B_{pix,<6:0>}$. As discussed in Section 4.5, the enable signal En_Pix_Dec is included to eliminate potential decoder output glitches due to the non-Gray transition in $B_{pix,<6:0>}$ from 1101011 to 0000000 at the end of each full spike transfer period. In hard-wiring the AND gates' Gray code values, which were arbitrarily chosen to follow the incremental mapping shown in the figure, the required bit inversions, represented by a bubble on the corresponding input leg, were generated using local inverters. The output of each AND gate feeds a pair of $1 \rightarrow 100$ inverting buffers – implemented with scaled inverter chains sized 1× -2× -7× -25× -100× relative to a minimum inverter – which drive the Read lines of a pair of pixel rows LOW during the appropriate spike

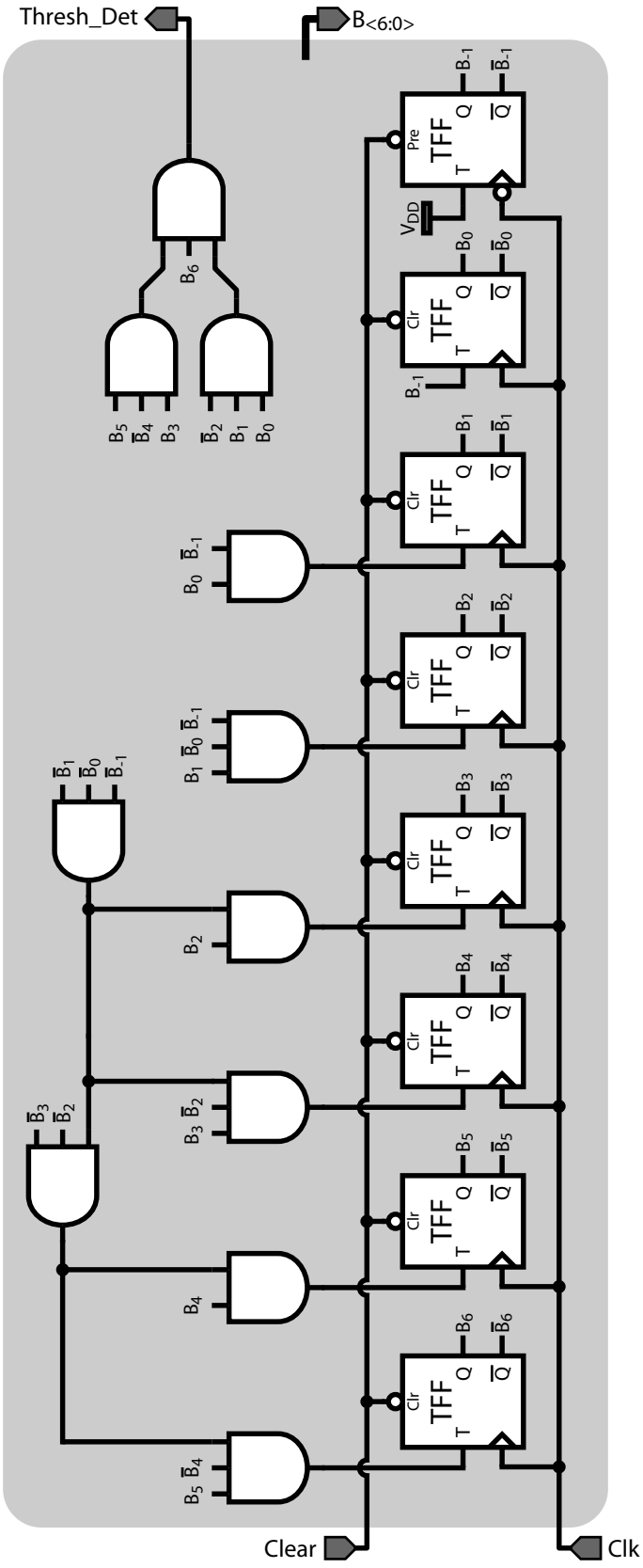

Figure 4-23: 7-bit Gray counter implementation.

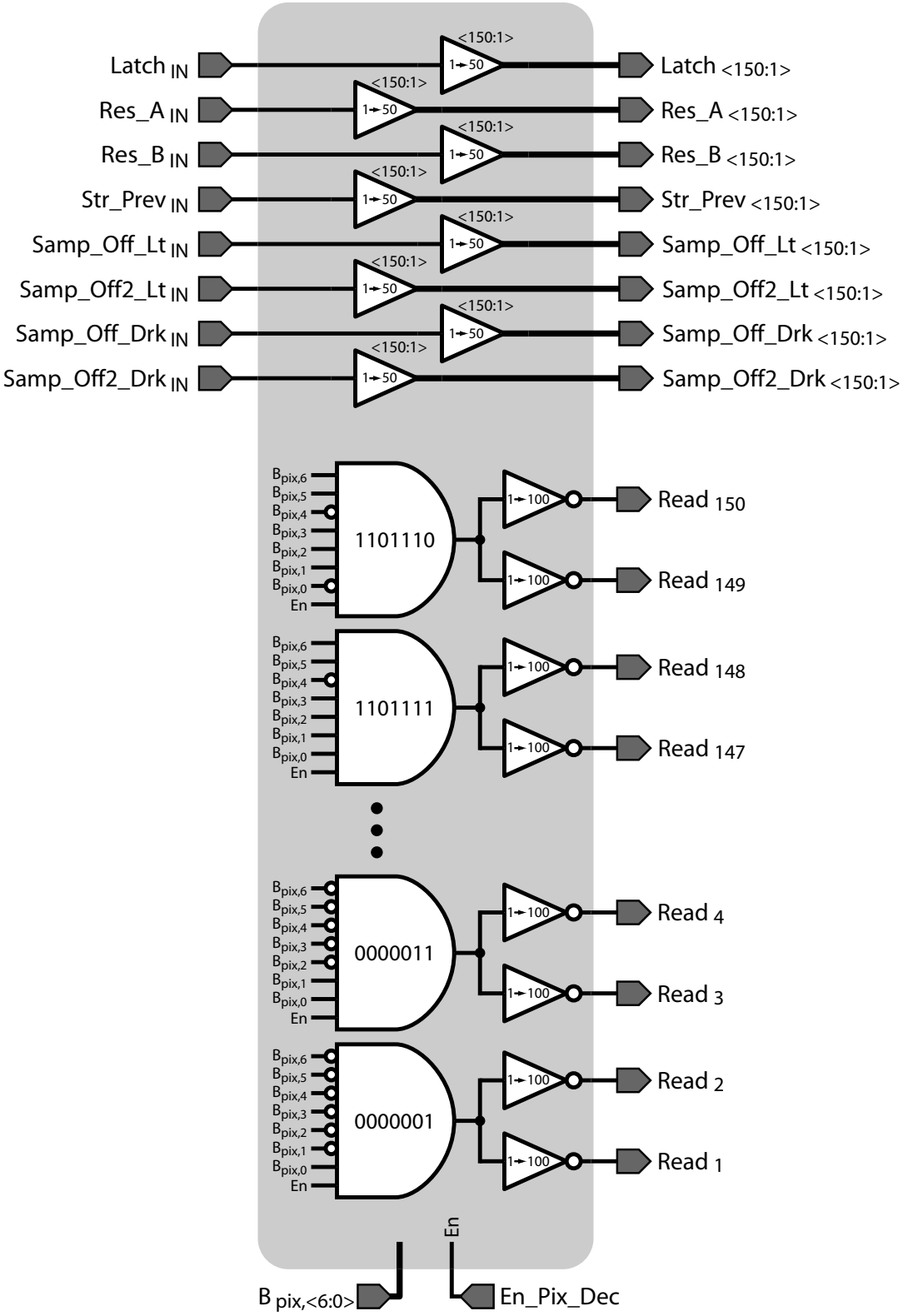

Figure 4-24: Pixel row decoder and control signal buffers for 150 row array.

transfer window.

In addition to the decoder logic and buffers, this block also contains an array of $1 \rightarrow 50$ non-inverting buffers which interface several global pixel control signals, provided from off-chip for maximum testing flexibility, with the corresponding row-level control lines in the pixel array. Although the buffers are shown grouped in the figure for clarity, in the layout they are distributed along the edge of the pixel array in a row-by-row manner, similar to the decoder's AND gates and inverting buffers.

4.7 SRAM Row Decoder and Buffers

In addition to helping the spike transfer controller coordinate spike communication, the memory row decoder also works in conjunction with the SRAM read controller to select a target memory row during SRAM read operations (see Section 4.9). As a result of this added functionality, the structure of this block, shown in Figure 4-25, is concomitantly more complex than that of the pixel row decoder presented in the previous section. To understand the operation of this decoder as a whole, begin by considering a pair of segments, for example Segments 1 & 2, each of which controls a block of memory cells eight rows tall (see Section 4.4.3). During the spike transfer mode of decoder operation, where `En_Mem_Dec` is `HIGH` and `Rd_Row` is `LOW`, the upper 8-input AND gates in both segments are enabled while the lower ones are disabled, resulting in three conditions under which any of the segments' outputs can be driven `HIGH`. First, when the Gray code $\texttt{B}_{\texttt{mem,<6:0>}}$ is `0000001`, the upper AND gate in Segment 1 forces $\texttt{RW\_Spk}_{\texttt{<1:8>}}$ to go `HIGH`, enabling the memory cells in rows $<1:8>$ to receive spike information from their corresponding pair of pixel rows. Second, when $\texttt{B}_{\texttt{mem,<6:0>}}$ transitions to `0000011`, the upper AND gate in Segment 2 forces both $\texttt{RW\_Spk}_{\texttt{<9:16>}}$ and $\texttt{RW\_Data}_{\texttt{<1:8>}}$ to go `HIGH`, enabling the memory cells in rows $<9:16>$ to receive spike information from their respective pixels while simultaneously sampling SRAM bus data into any memory cells in rows $< 1:8 >$ that received a spike in the previous interval. Following a similar line of reasoning, when $\texttt{B}_{\texttt{mem,<6:0>}}$ transitions to `0000010`, both $\texttt{RW\_Spk}_{\texttt{<17:24>}}$ and $\texttt{RW\_Data}_{\texttt{<9:16>}}$ go `HIGH`, thus these staggered spike transfer and data storage intervals propagate in lockstep through each eight-row block in the memory cell array. Due to the boundary condition at Segment 75, an extra 8-input AND gate is required to detect when $\texttt{B}_{\texttt{mem,<6:0>}}$ transitions to `1101010` and force $\texttt{RW\_Data}_{\texttt{<593:600>}}$ to go

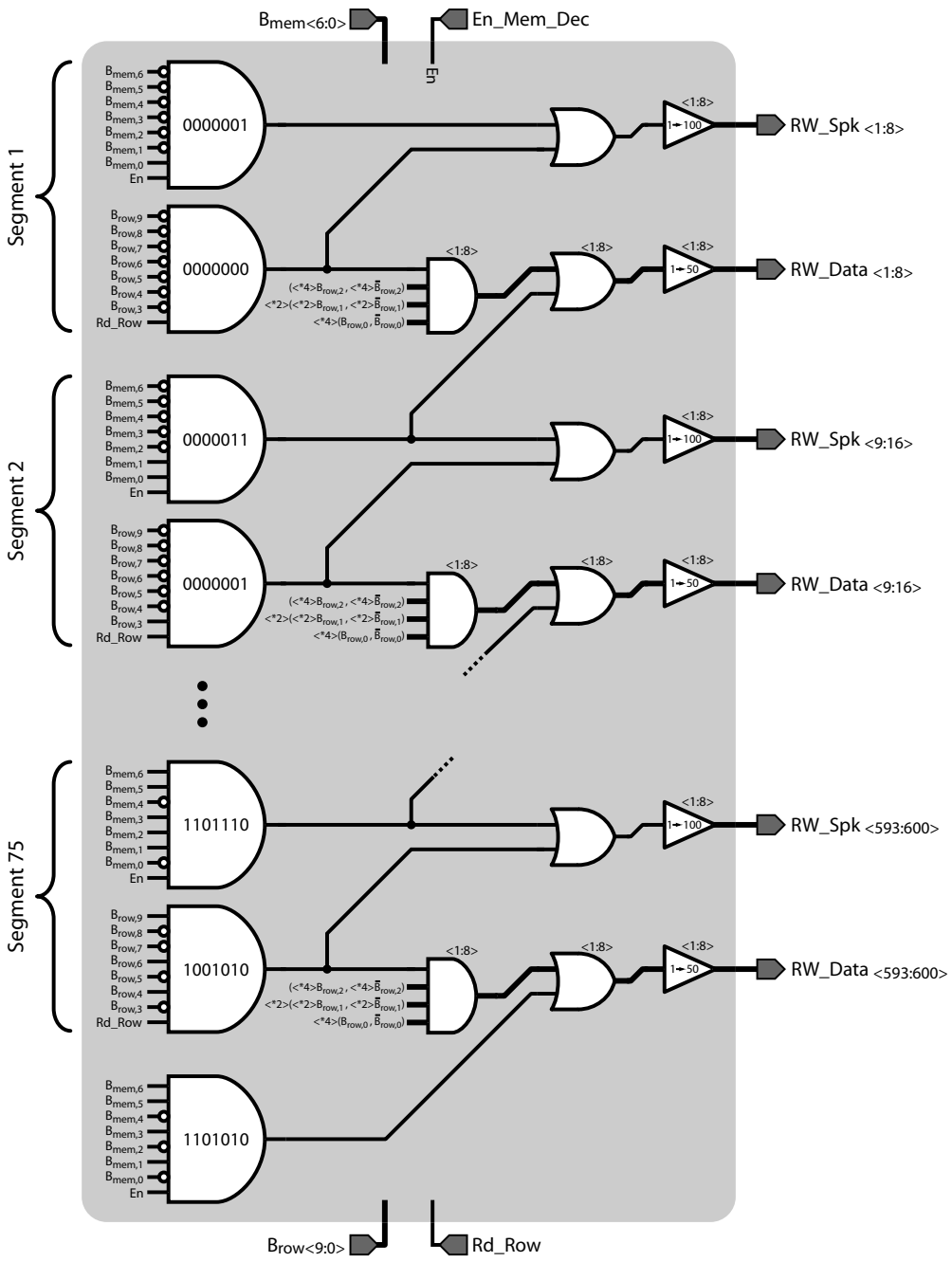

Figure 4-25: Memory row decoder and buffers for 150 row pixel array.

`HIGH`, explaining why the spike controller was designed to reset on the next Gray code in the sequence, `1101011`, as mentioned in Section 4.5.

During the other mode of decoder operation, SRAM read mode, `En_Mem_Dec` is `LOW` and `Rd_Row` is `HIGH`, which disables the upper 8-input AND gate in each section and enables the

lower one. These lower 8-input AND gates are hard-wired to decode the upper seven bits of the row address provided by the SRAM read controller, $\texttt{B}_{\texttt{row,<9:3>}}$. Taking Segment 1 as an example, when $\texttt{B}_{\texttt{row,<9:3>}}$ is 0000000, the segment's lower AND gate outputs a HIGH level, which forces $\texttt{RW\_Spk}_{\texttt{<1:8>}}$ to go HIGH and enables the segment's eight 4-input AND gates to decode the lower three bits of the row address $\texttt{B}_{\texttt{row,<2:0>}}$. Based on the value of $\texttt{B}_{\texttt{row,<2:0>}}$, a single output in the set $\texttt{RW\_Spk}_{\texttt{<1:8>}}$ will be driven HIGH, causing the memory cells in this row to output their data on the SRAM data bus.

4.8 SRAM Column Interface

As discussed in Section 4.4.2, two 18-bit data busses $\texttt{D}_{\texttt{<17:0>}}$ and $\overline{\texttt{D}}_{\texttt{<17:0>}}$ are routed vertically through each column in the memory cell array. The SRAM column interface, in conjunction with the SRAM read controller, is in charge of managing data I/O along these column busses. As shown in Figure 4-26, this is accomplished using 70 dedicated column-level I/O blocks, a tri-state encoder and buffer block, and two 18-bit addressable registers, along with a set of input control signals. Each column-level I/O block contains a pair of 18-bit tri-state input drivers, an 18-bit latch and tri-state output buffer, and an AND gate hard-wired to respond to a unique 7-bit column address. When the global control signal R_W is LOW data is to be written to the memory cell array, and the tri-state encoder generates an appropriate set of control signals for the columns' tri-state input drivers based on the two 18-bit codes stored in the addressable registers A and B. When R_W is HIGH data is to be read from a row in the memory cell array, and the tri-state encoder commands the column drivers to first pre-charge both sets of data busses when Pre_Tri is HIGH, and then enter a tri-state mode when Pre_Tri goes LOW. Following this initialization of the column busses, the SRAM read controller and memory row decoder enable the appropriate memory array row and allow each pair of bit lines to develop a differential read voltage (see Section 4.4.1), which the latch blocks amplify into a full-scale digital result when Latch_Col goes HIGH. Each column's 18-bit tri-state output buffer connects to the global output bus $\texttt{Glb\_Out}_{\texttt{<17:0>}}$, but it only actively drives the bus when both En_Col_Buf is HIGH and the state of $\texttt{B}_{\texttt{col,<6:0>}}$ causes the column's AND gate output to go HIGH. The decoder structure ensures that only a single column buffer is enabled at a time, while the remaining buffers reside in tri-state. Each of the blocks that constitute this column interface block are discussed below.

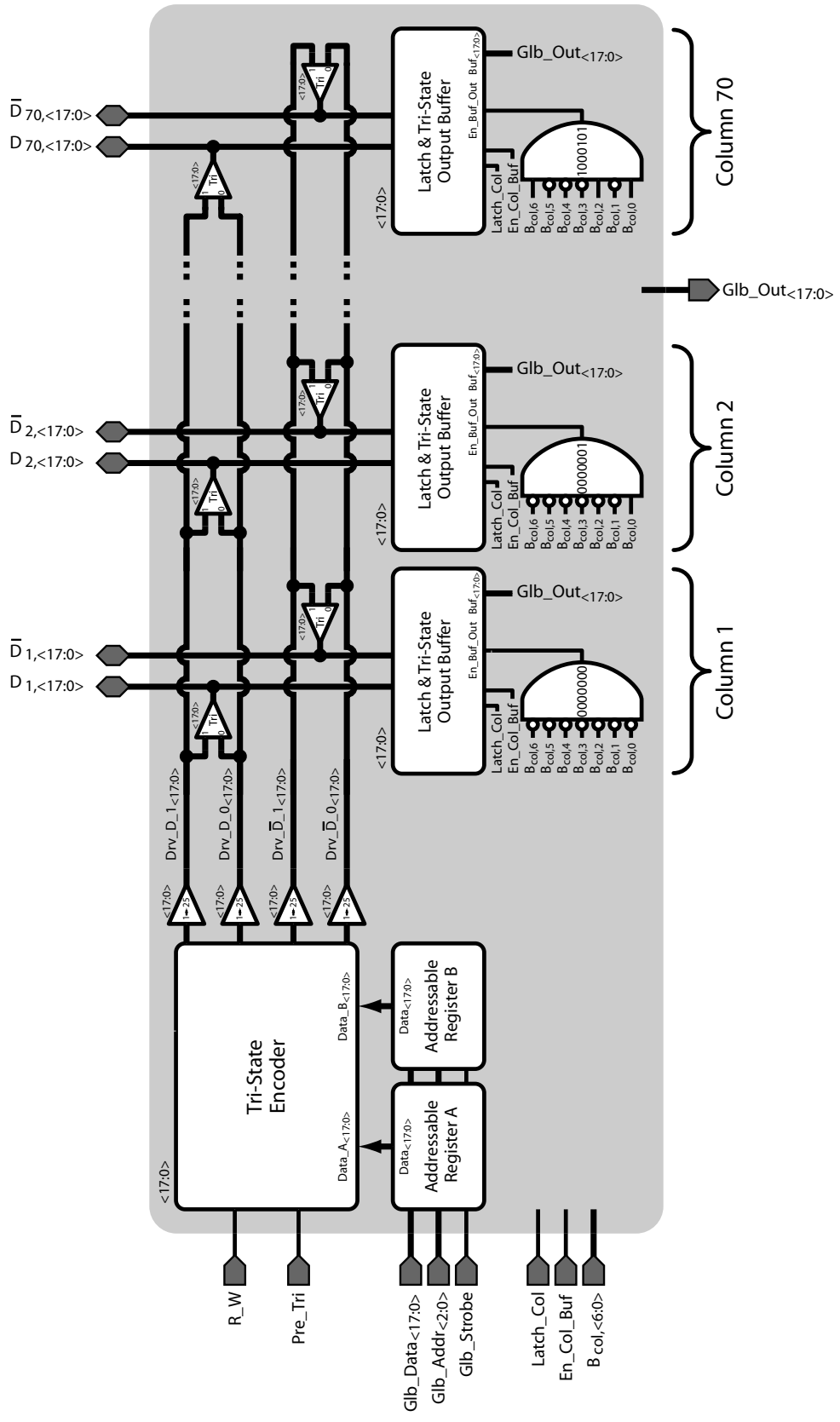

Figure 4-26: Block diagram of SRAM column interface.

R/W	Pre_Tri	Data_A$_x$	Data_B$_x$	Drv_D_1$_x$	Drv_D_0$_x$	Drv_$\overline{D}$_1$_x$	Drv_$\overline{D}$_0$_x$	Comment
0	X	0	0	1	0	1	0	Hold
0	X	0	1	0	1	1	0	Write 0
0	X	1	0	1	0	0	1	Write 1
0	X	1	1	1	0	1	0	Hold
1	0	X	X	0	0	0	0	Tri-state
1	1	X	X	1	0	1	0	Pre-charge

Table 4.5: Tri-state encoder operation.

4.8.1 Addressable Registers A & B

As discussed in Section 4.4.1 and shown in Table 4.3, by selecting from three of the four possible states that can be driven on an SRAM cell's I/O lines Q and $\overline{Q}$, the cell can be instructed to either update its internal state to a 1 or a 0 or to hold its previous state during a write operation. In order to independently select from these three options for each bit of the column bus, the column interface requires two bits of control information per bus bit. These two bits are provided on the global input bus Glb_Data$_{<17:0>}$ in two successive phases, and are locally stored in the 18-bit addressable registers A and B, which can be accessed at the global addresses 010 and 011, respectively (see Section 4.5.2). The outputs of these two registers are provided in parallel to the tri-state encoder block, which use each pair of bits to control the column-level tri-state input drivers during write operations.

4.8.2 Tri-State Encoder

The tri-state encoder generates the necessary signals to control the column-level tri-state input drivers during both read and write operations. This block is composed of 18 identical encoder cells, each of which independently controls the two column bus lines associated with a single SRAM bit based on the logic shown in Table 4.5. When the the global control signal R_W is 0, which commands data be driven on the column busses, the encoder outputs the necessary control signals to instruct the tri-state input drivers to either hold the SRAM bit state constant, or update it to a 0 or 1, depending on which of the four possible input states is present on Data_A$_x$ and Data_B$_x$. When R_W is 1 , the encoder ignores the bits stored in the addressable registers and instead responds only to the state of the Pre_Tri signal. This input results in the encoder commanding the pre-tri input drivers to either pre-charge both column bit lines when it is 1, or place both bit lines into a tri-state mode

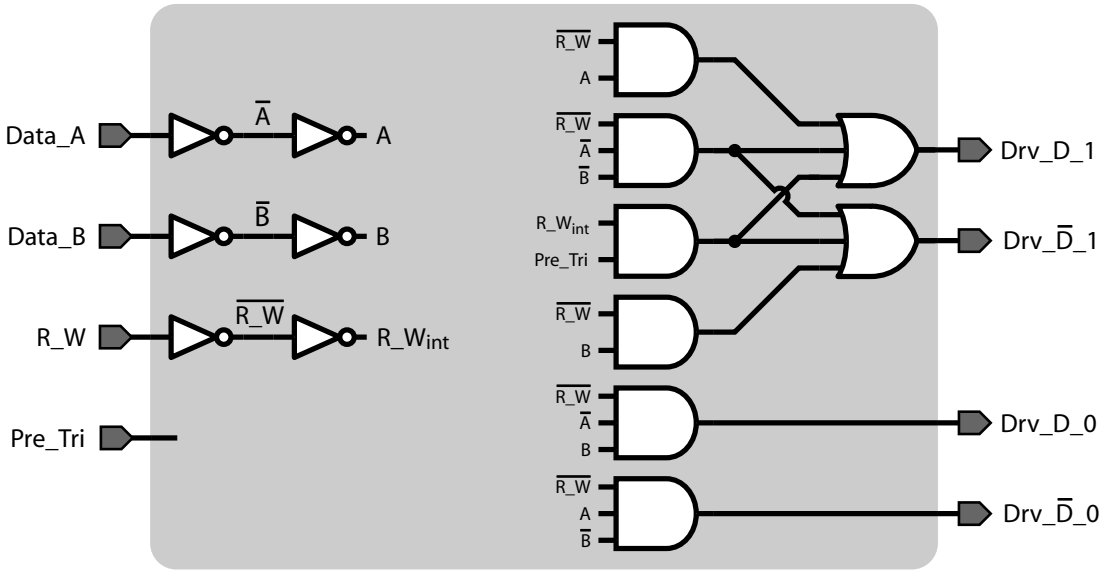

Figure 4-27: The tri-state encoder logic associated with a single column bus bit.

In_1	In_0	Output
0	0	Tri-state
0	1	0
1	0	1
1	1	Not Allowed

Table 4.6: Tri-state input driver operation.

when it is 0. The logic with which this behavior was implemented for each bit is shown in Figure 4-27, while the $1 \rightarrow 25$ output buffers were constructed using scaled inverter chains sized $1\times$ -$2\times$ -$7\times$ -$25\times$ relative to a minimum inverter.

4.8.3 Tri-State Input Drivers

Each column-level I/O block contains a pair of 18-bit tri-state input drivers, which can either actively drive the column data busses or enter a tri-state output mode, based on the control signals provided by the tri-state encoder. The drivers are each constructed from 18 identical unit cells, which independently operate based on the logic shown in Table 4.6. This driver response elucidates the control signal output behavior of the tri-state encoder described above in Table 4.5. Besides implementing this logical functionality, each driver must also

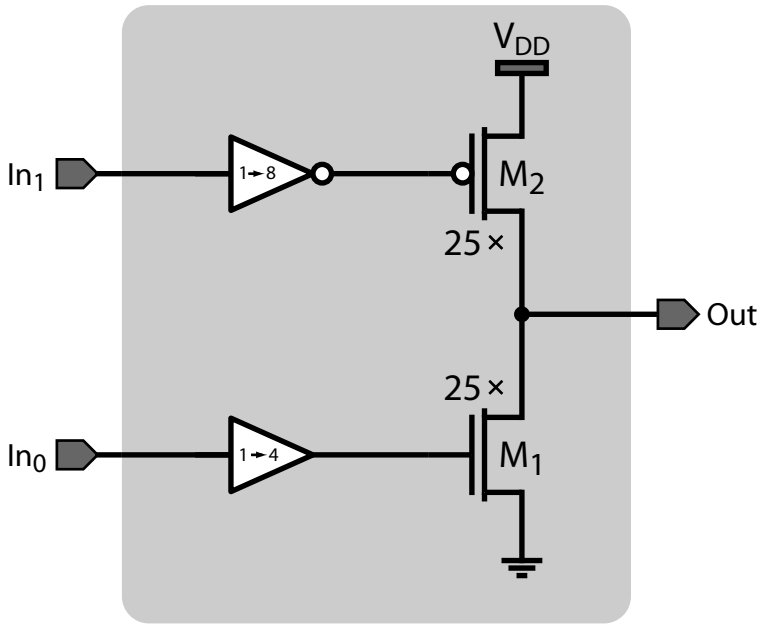

Figure 4-28: Tri-state input driver used to drive each bit line.

be capable of driving a significant capacitive load as well, which is accomplished using the unit driver cell topology shown in Figure 4-28. This driver consists of a 25× minimum size output inverter in which the PMOS and NMOS devices are independently driven by a $1 \rightarrow 8$ inverting and $1 \rightarrow 4$ non-inverting buffer, respectively. This split control allows both output devices to be simultaneously turned `OFF`, implementing the high impedance tri-state.

4.8.4 Latch, Tri-State Output Buffer, and Column Decoder

The 18-bit latch and tri-state output buffer block amplifies the differential voltages developed on $\texttt{D}_{<17:0>}$ and $\overline{\texttt{D}}_{<17:0>}$ during an SRAM row read into a full-scale digital result and, when selected via the column decoder, drives this data onto the global output bus $\texttt{Glb\_Out}_{<17:0>}$. Each block is composed of 18 copies of the single-bit latch and tri-state buffer shown in Figure 4-29. Transistors $M_{1\text{-}6}$ form a gated cross-coupled latch which, when `Latch_Col` goes `HIGH`, turns `ON` and amplifies the differential read voltage across `D` and $\overline{\texttt{D}}$ into full-scale digital levels. The tri-state output buffer, formed by transistors $M_{7\text{-}8}$ along with the $1 \rightarrow 8$ inverting and $1 \rightarrow 4$ non-inverting buffers, is identical to the tri-state input driver topology of Figure 4-28. However, the three additional AND gates force the buffer into tri-state mode unless `En_Col_Buf` and `En_Buf_Out` are both `HIGH`, as depicted by the

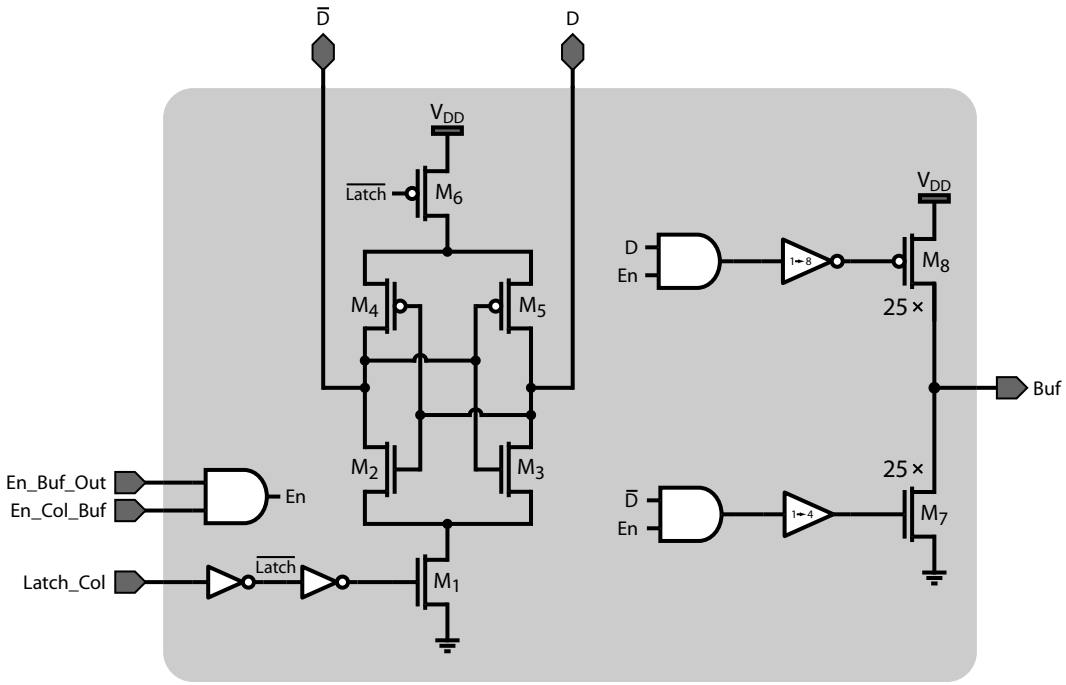

Figure 4-29: Single-bit latch and tri-state output buffer.

En_Col_Buf	En_Buf_Out	D	$\overline{D}$	Buf
0	0	X	X	Tri-state
0	1	X	X	Tri-state
1	0	X	X	Tri-state
1	1	0	0	Not Allowed
1	1	0	1	0
1	1	1	0	1
1	1	1	1	Not Allowed

Table 4.7: Tri-state output buffer operation.

logic in Table 4.7. Thus, as shown in Figure 4-26, the hard-wired 7-input AND gate decoder and column address bus $\mathtt{B}_{\mathtt{col},<6:0>}$ select a single column output buffer to drive data on the global data bus $\mathtt{Glb\_Out}_{<17:0>}$, while the remaining column buffers are guaranteed to be tri-stated.

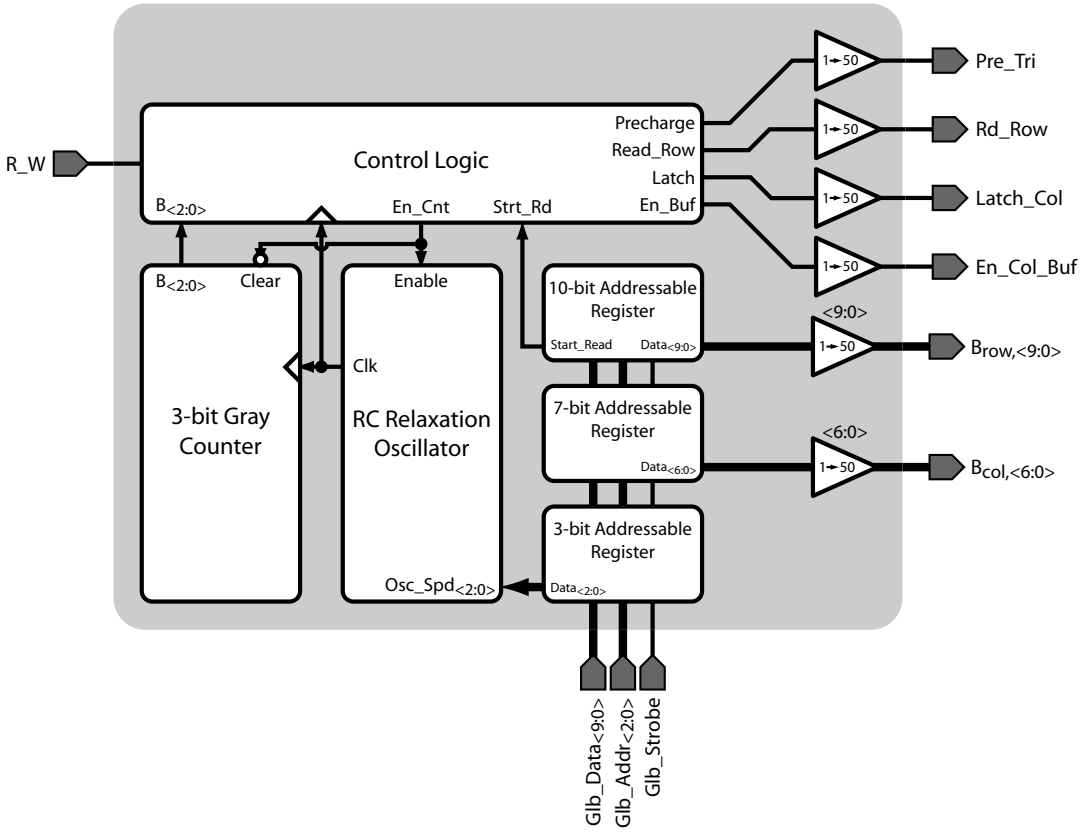

Figure 4-30: Block diagram of SRAM read controller.

4.9 SRAM Read Controller

Much like the spike transfer controller coordinates pixel-to-memory spike communication, the SRAM read controller works in conjunction with the SRAM row decoder and column interface to perform data reads from the SRAM array. For design simplicity, the internal structure of this block, shown in Figure 4-30, makes maximum reuse of the components used to construct the spike transfer controller (see Figure 4-16). The controller contains three addressable registers: a 10-bit row address register accessible at global address `101`, a 7-bit column address register accessible at global address `100`, and a 3-bit tuning register accessible at global address `001`, which is used to adjust the nominal oscillator period to $10ns$. Similar to the spike transfer controller, the SRAM read controller also contains an RC relaxation oscillator, a Gray counter, and a control logic block, which collectively implement the state-machine behavior illustrated by the timing waveforms shown in Figure 4-31. As

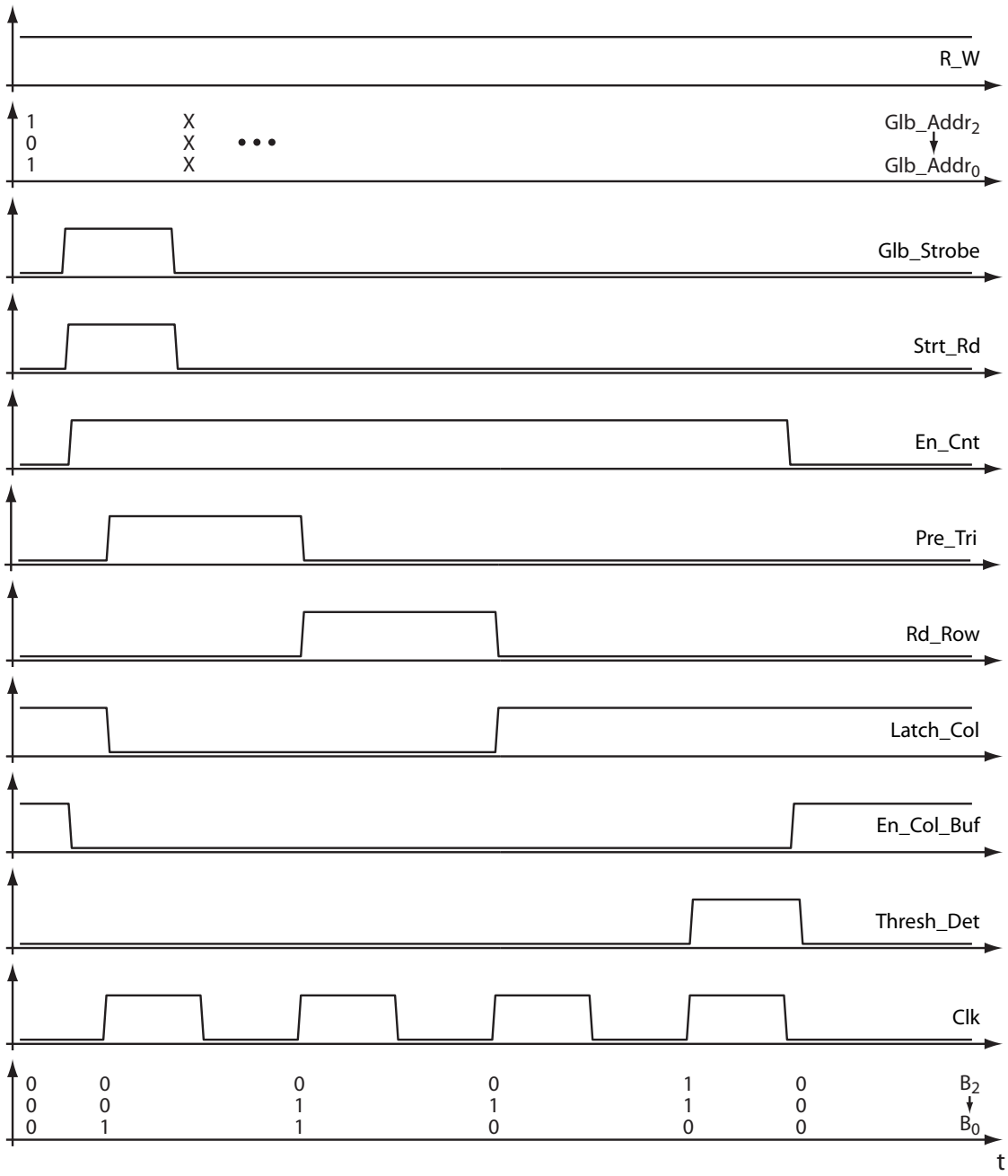

Figure 4-31: Waveforms illustrating SRAM read controller operation.

depicted in these waveforms, the SRAM read controller nominally rests in an inactive state except during brief intervals after the arrival of a pulsed trigger signal, which in this case is the `Strt_Rd` signal generated by the 10-bit row address register. Referring to the generic addressable register in Figure 4-19, the `Strt_Rd` signal is equivalent to the output of the 4-input AND gate, thus it pulses `HIGH` in slightly delayed synchrony with `Glb_Strobe` when

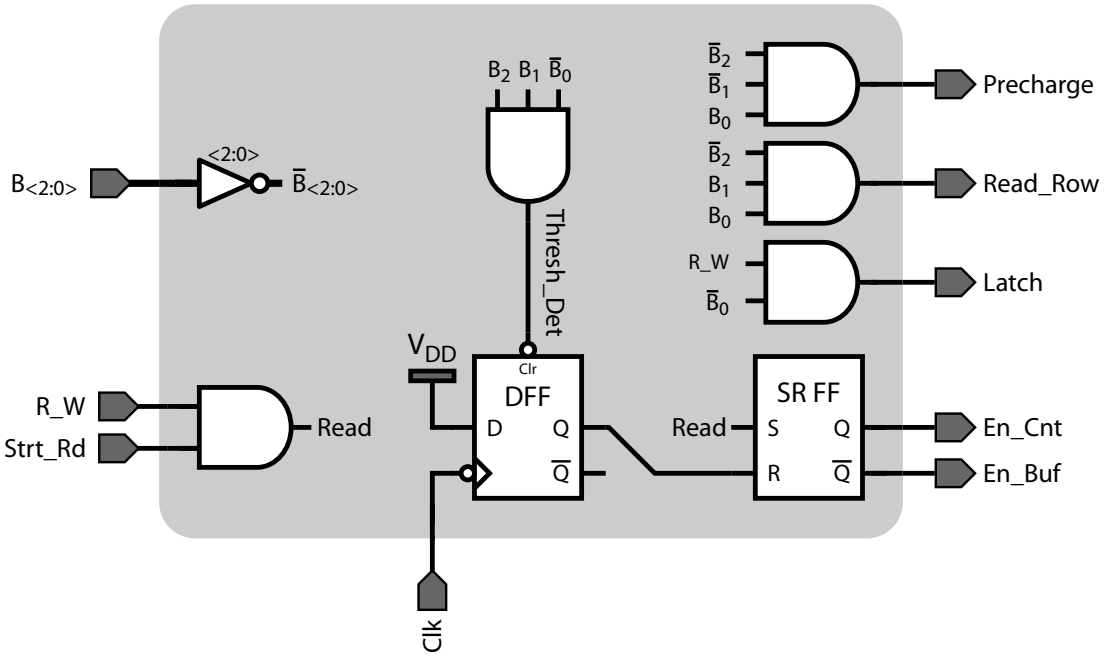

Figure 4-32: SRAM read control logic block.

`Glb_Addr`$_{<2:0>}$ is `101`. If the global signal `R_W` is `HIGH` when this pulse reaches the control logic block, it forces `En_Cnt` to go `HIGH`, removing the `Clear` condition from the 3-bit Gray counter and enabling the RC relaxation oscillator. As in the spike transfer controller, this causes the Gray counter to begin incrementing, in this case from a rest state of `000` up through aterminal value of `110`. However, instead of buffering this Gray code out of the controller to be decoded externally, it is fed to the control logic block where it is locally decoded to generate the `Pre_Tri`, `Rd_Row`, `Latch_Col`, and `En_Col_Buf` output signals with the timing depicted in Figure 4-31. The control logic block also internally generates the `Thresh_Det` signal shown in the figure which, on the first falling edge of `Clk`, causes `En_Cnt` to reset, disabling the oscillator and clearing the output of the Gray counter to its rest state of `000`. Since the only significant difference between this controller and the spike transfer controller is within the control logic block, the internal structure of this block alone will be shown here (see Figure 4-32). The structure of the remaining blocks can be easily inferred based on the descriptions and figures in Section 4.5.

4.10 Summary

This chapter discussed a prototype CMOS imager design that implements the time-based dual-threshold wide-dynamic-range imaging algorithm introduced in Chapter 3. The major imager blocks include:

- A 150×280 spike-based pixel array. Each 12.5 μm×12.5 μm pixel employs an n-well/p-substrate photodiode, capacitively-coupled charge reference, novel fully-differential comparator, and spike-generator, which together encode pixel threshold information in a spike-based timing format. The pixel layout achieves 42.7% fill factor.

- A parallel array of spike-gated SRAM memeory cells, which form a one-to-one mapping with the pixels in the imaging array. Based on spike-encoded threshold information from its corresponding pixel, each memory cell records up to 18 bits of timing information from a global data bus. Multiple writes to arbitrary subsets of the 18 bits are possible in a single frame, allowing various pieces of pixel information to be recorded.

- A spike transfer controller, which in conjunction with a pair of pixel and memory row decoders and a set of parallel spike buffers, coordinates spike transmission between each pixel and its associated memory cell. The entire array's pixel-to-memory spike transmission is guaranteed to complete within a nominal 375 ns transfer window.

- An SRAM read controller, row decoder, and column interface, which enable data to be written to and read from the SRAM array.

Chapter 5

Experimental Results

The previous chapter presented the design of a prototype time-based CMOS imager capable of implementing the proposed dual-threshold imaging algorithm. This chapter discusses the experimental setup and techniques that were employed to test this chip and presents measured performance data from it. The experimental results are also compared with the theory developed in Chapter 3, as well as with past designs from the literature.

5.1 Imager Die

The prototype dual-threshold imager was fabricated using a 0.18-μm CMOS 1.8-V 6-metal process, and took advantage of the linear MIM capacitor and poly resistor options. Although 3.3-V transistors were also available, only the 1.8-V devices were employed in the design. The die photo shown in Figure 5-1 illustrates the major functional blocks of the imager, and is similar to the block-diagram presented previously in Figure 4-1. The overall die measures 5000 μm $\times$5000 μm including pads.

5.2 Test Board

Since achieving high-speed pixel-to-memory communication is critical in minimizing the time ΔT_{min} between successive pixel threshold samples, this portion of the digital control was implemented on-chip (see Section 4.5). Similarly, since the SRAM read speed limits the imager output data rate, its digital control was also integrated on the die (see Section 4.9). For maximum testing flexibility, all remaining control signals are generated off-chip using

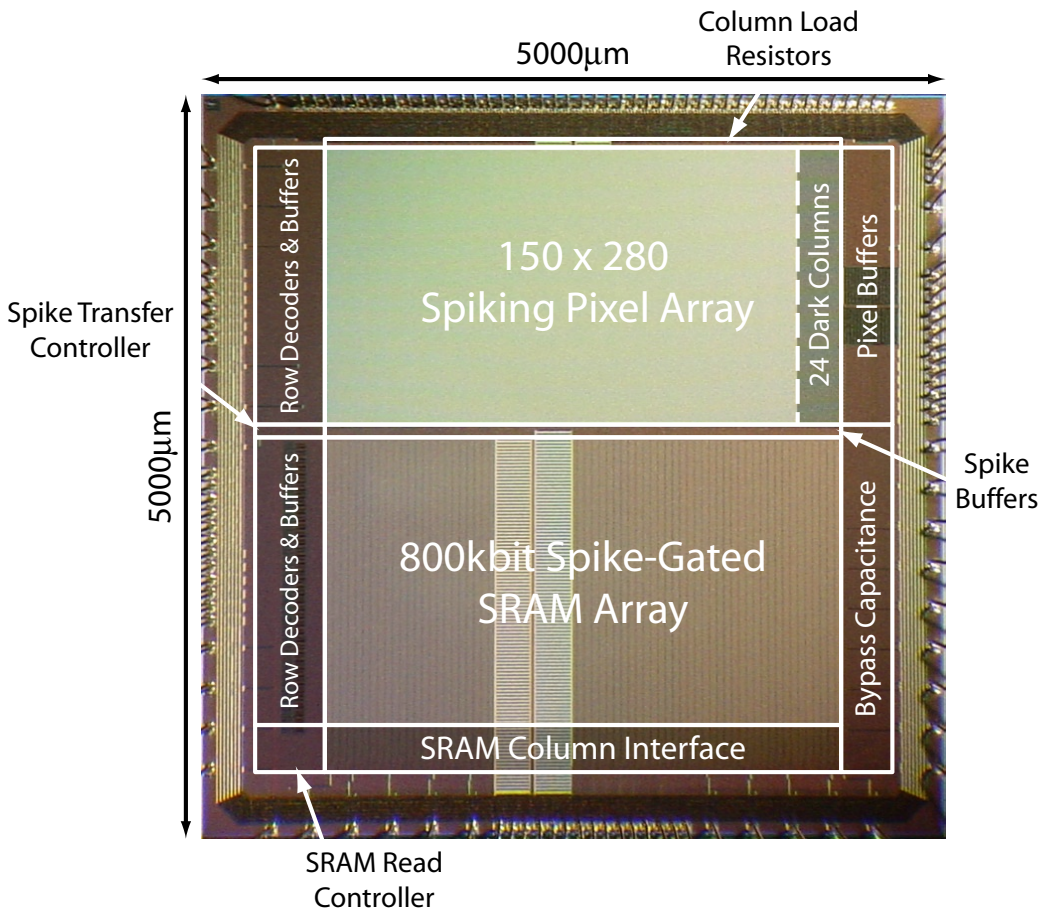

Figure 5-1: Prototype dual-threshold imager die photo.

various components included on the test board shown in Figure 5-2. Digital control vectors are generated using an AT91SAM7SE512 32-bit ARM microcontroller [77], and are used to control the imager as well as a custom 12-bit DAC. The microcontroller also contains a built-in universal serial bus (USB) transceiver, which is used to transmit image data to a host computer. Custom code was written to enable the microcontroller to comply with the USB video class (UVC) specification v1.1 [78], allowing the board to communicate directly with the built-in Windows XP driver *usbvideo.sys* [79]. The advantage of complying with this specification and driver is that MATLAB's image acquisition toolbox supports data capture from devices within this class [72]. Thus, the test board enables imager data to be generated, transferred from the imager to the microcontroller, and then streamed over USB directly into MATLAB for post-processing and recording. Two final board-level components of great importance are the 4.3-mm focal length miniature glass lens (Model V-4304.3-2.0)

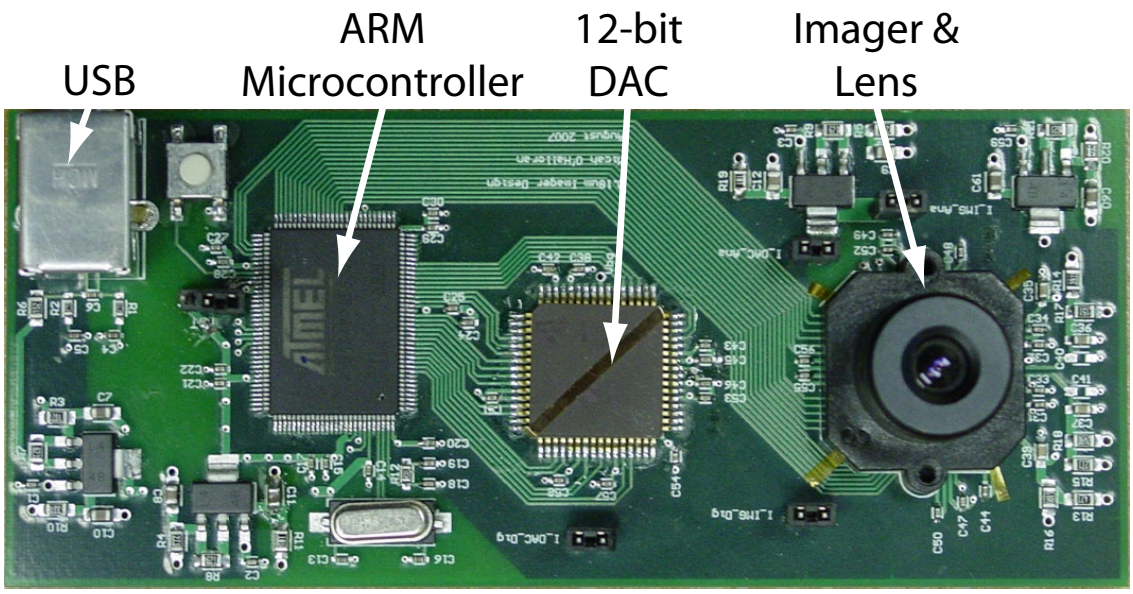

Figure 5-2: Test board used to characterize prototype imager.

and lens mount (Model V-LH4), which were purchased from [80].

Though the high-level microcontroller code, such as the USB code mentioned above, was written in C, the time-critical imager and DAC control code was written in assembly. Even with this assembly-level optimization, the minimum time between successive pixel threshold detections ΔT_{min} could only be reduced to 60 clock cycles. With the maximum microcontroller clock frequency limited to ≈ 48 MHz due to the use of the USB transceiver, the value $\Delta T_{min} \approx 1.252$ μs is nearly twice as large as the prototype is capable of achieving (see Section 4.2), and will result in a reduced experimental dynamic range.

5.3 12-bit Segmented DAC

The custom 12-bit DAC was implemented on a separate die, offering greater testing flexibility. A segmented topology consisting of a 4-bit tapped resistive ladder feeding an 8-bit switched-capacitor DAC was employed in this design, as shown in Figure 5-3. The upper 4-bits of the input code control which of the 16 equal-valued ladder resistors the two taps span, providing V_{low} and V_{high}. The binary-weighted switched-capacitor DAC then interpolates between these two input voltages based on the lower 8-bits of the input code. The switched-capacitor DAC amplifier was designed to directly drive loads exceeding 300 pF at

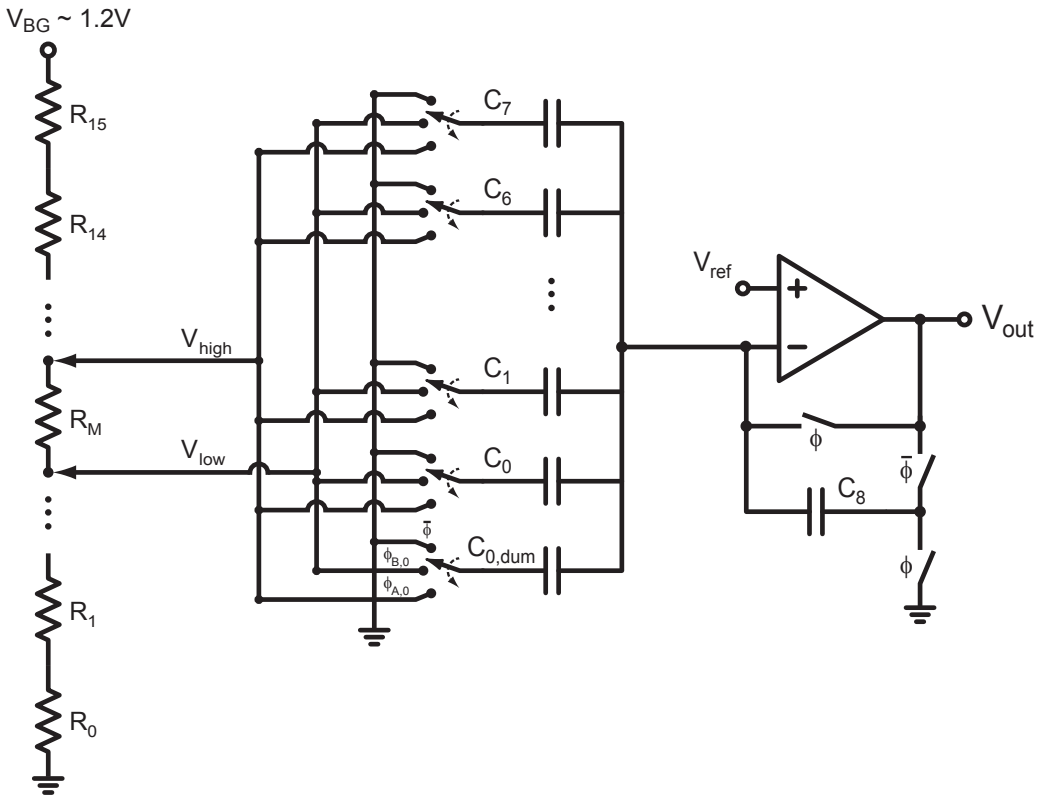

Figure 5-3: 12-bit segmented DAC topology.

greater than 5 MHz bandwidth, enabling the DAC output V_{out} to drive the large parasitic capacitance on the pixel V_{thresh} line. When powered down, V_{out} defaults to V_{DD}, which based on the discussion of reset switch leakage in Section 4.1.1 should introduce no potential leakage problems in the pixels. This allows the DAC to be powered down between pixel comparisons, leading to significant energy savings.

A representative pair of DNL and INL characteristics measured from the implemented DAC are shown in Figure 5-4. One oversight in the design that is evident in the DNL plot is that the charge transfer between the binary weighted capacitors and the integrating capacitor C_8 leads to increased output voltage variance as V_{low} and V_{high} move up the resistor ladder. This occurs due to intrinsic variance in the binary-weighted capacitors being scaled by the magnitude of V_{low} and V_{high}. One solution is to eliminate C_8 and wrap the binary-weighted capacitors around the amplifier when the DAC enters hold mode. Even with this flaw, the second plot illustrates the implementation achieves nearly 11-bits of INL precision. While the resulting $\approx \pm 1$-bit peak offset does increase the effective quantization

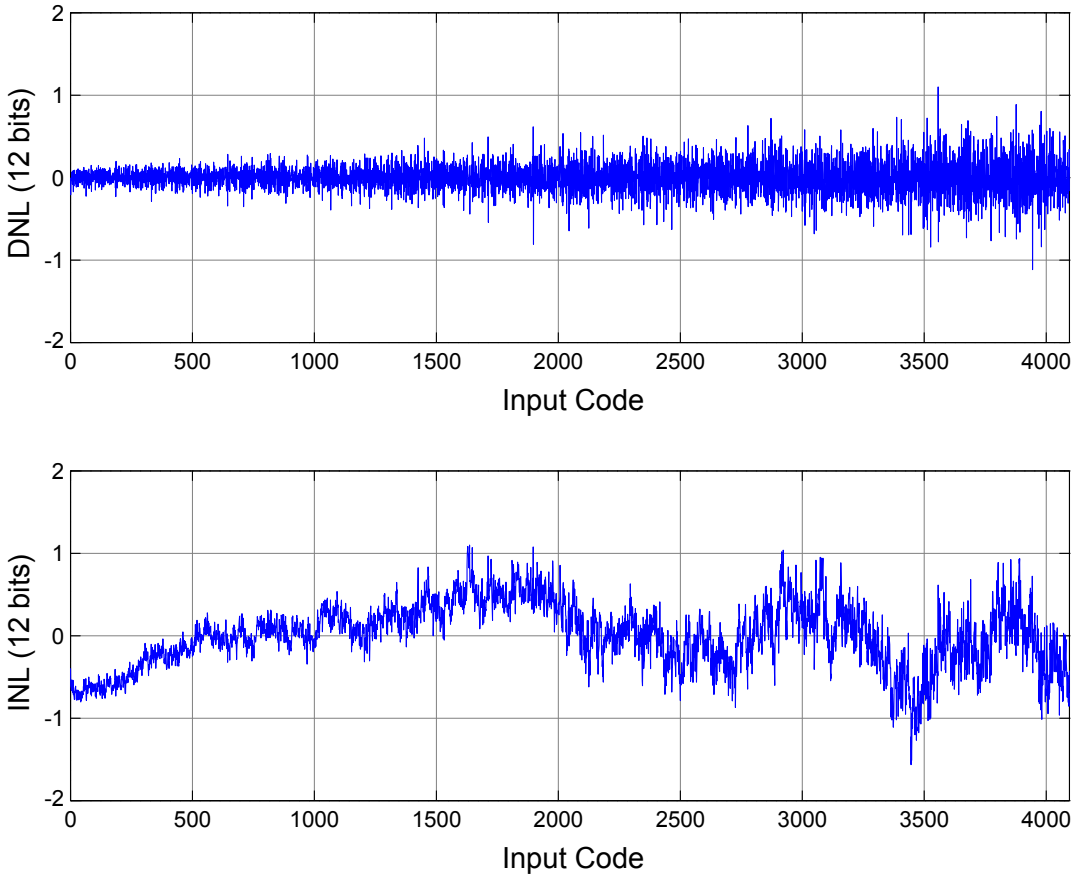

Figure 5-4: Measured DNL and INL from 12-bit DAC.

noise floor, it turns out to not be a limiting factor in the overall imager performance.

5.4 C_{coup} Characterization

Based on the ideal dual-threshold pixel response defined by Equation 3.9, determining the true average value of C_{coup} across the imager array is critical in enabling absolute photocurrent and dark current levels to be inferred from output pixel data. However, this capacitor is tricky to measure accurately due the large parasitic capacitances (relative to the expected value of C_{coup} itself) connected to both of its plates. Fortunately, using the experimental setup shown in Figure 5-5, an accurate measurement of C_{coup} can be obtained despite the presence of these parasitics. The procedure employed in conjunction with this test setup begins by biasing all pixels in their auto-zero reset mode (see Sections 3.3.2 and 4.1.1). This places each pixel comparator in unity-negative feedback, as shown, and forces

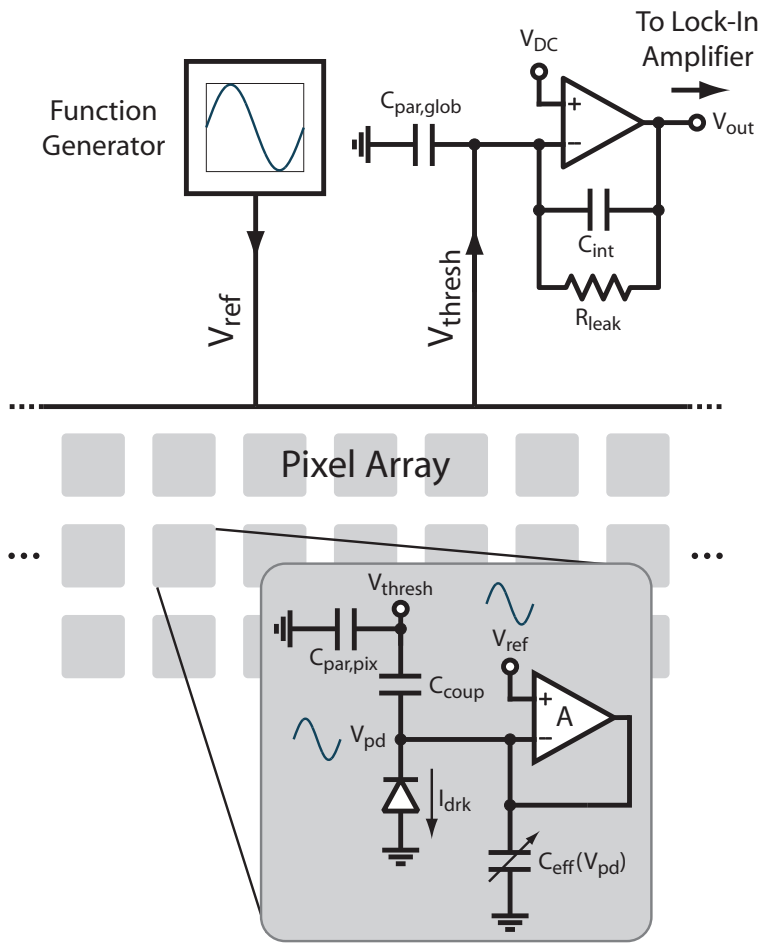

Figure 5-5: Experimental setup used to extract C_{coup}.

low-frequency movements in V_{ref} and V_{pd} to be related by

$$\Delta V_{pd} = \Delta V_{ref} \cdot \left[\frac{A}{1+A}\right], \tag{5.1}$$

where A is the open-loop gain of the comparator. Based on the comparator simulation results shown in Table 4.1, $A > 150$ V/V, thus movements in V_{ref} and V_{pd} should match within a fraction of a percent. Next, apply the composite input signal $V_{ref}(t) = [850\text{ mV} + A_{sine,in} \cdot sin(2\pi f_{osc} t)]$ with $f_{osc} = 50$ kHz. With the function generator connected to the chip, the sine wave amplitude at V_{ref} was measured to be $A_{sine,in} = 35.86$ mV. As long as f_{osc} is at least several orders of magnitude below the comparator unity-gain bandwidth, which is the case here, Equation 5.1 predicts that each pixel's V_{pd} node will oscillate at the

same frequency and with an amplitude within a fraction of a percent of $A_{sine,in}$.

Turning our attention now to the bottom plate[1] of C_{coup}, the presence of $C_{par,pix}$ and $C_{par,glob}$, which are due to pixel-level and chip/board-level parasitics associated with routing V_{thresh}, respectively, are what prohibit direct measurement of C_{coup} using a simple capacitance meter at V_{thresh}. The approach shown in the figure removes the effect of these parasitics by connecting V_{thresh} to the virtual ground terminal of an op-amp-based leaky integrator, which forces $V_{thresh} = V_{DC}$. As long as the op-amp gain and bandwidth are large enough to ensure near-ideal behavior, which is the case in this experiment, its output voltage will be given by

$$V_{out} = -V_{pd} \cdot \frac{Z_{fb}(s)}{Z_{in}(s)} = -V_{pd} \cdot \frac{\left[\frac{R_{leak}}{1+sC_{int}R_{leak}}\right]}{\left[\frac{1}{N \cdot M \cdot sC_{coup}}\right]}, \tag{5.2}$$

where $N = 150$ and $M = 280$ are the number of rows and columns in the array, respectively. Note this result is independent of the parasitic capacitances connected to V_{thresh}.

The final phase of the experiment requires accurate knowledge of C_{int}, thus before the integrator was constructed, the value of C_{int} was characterized at 50 kHz using a high-accuracy capacitance meter and found to have a value $C_{int} = 896.5$ pF. Additionally, by choosing $R_{leak} = 4.7$ MΩ, the lowpass cutoff of this parallel impedance occurs at $f_{cutoff} = \frac{1}{2\pi R_{leak} C_{int}} \approx 38$ Hz, more than three orders of magnitude below f_{osc}. Thus, to very good approximation $Z_{fb}(s) \approx \frac{1}{sC_{int}}$, which allows Equation 5.2 to be simplified to the form

$$V_{out} = -V_{pd} \cdot \frac{N \cdot M \cdot C_{coup}}{C_{int}}. \tag{5.3}$$

Using a lock-in amplifier to measure the amplitude of the output sine wave at V_{out} gave $A_{sine,out} = 13.436$ mV, which along with $A_{sine,in}$, C_{int}, and Equation 5.3 yields

$$C_{coup,extracted} = \frac{|A_{sine,out}|}{|A_{sine,in}|} \cdot \frac{C_{int}}{N \cdot M} = 7.998 \text{ fF}. \tag{5.4}$$

This result is in excellent agreement with the design value of 8 fF.

[1]The bottom plate in the physical layout, not the figure.

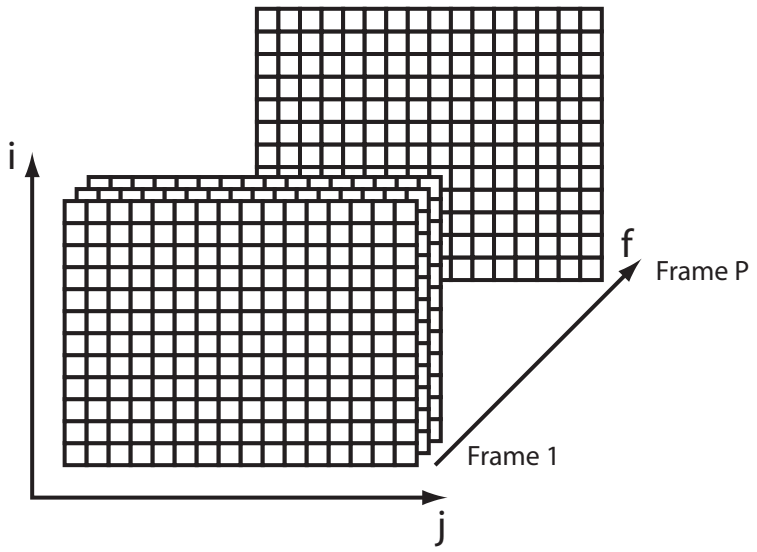

Figure 5-6: Format of experimental imager data.

5.5 I_{dark} Characterization

Armed with an average value for C_{coup}, the next experiment employed the dual-threshold algorithm to measure the photodiode dark current statistics across the pixel array. During this experiment, the chip was shielded from external illumination and the frame time was increased to $t_{frame,end} = 150$ ms to allow ample charge to be collected at each pixel before t_{thresh}. This required a modified V_{thresh} waveform with a correspondingly longer frame time, which was generated using the MATLAB script mentioned in Section 3.4.4. A series of 100 frames was captured from the test board using MATLAB, and the resulting dual-threshold pixel data was processed using Equation 3.9 to yield a dark-current data set with the form shown in Figure 5-6. This figure illustrates that the processed image data $D(i, j, f, I_{lux})$ is a function of the pixel row i and column j, the frame number f, and the input illuminance I_{lux}, which in this case is zero. The data array was analyzed to determine the average dark current of each pixel using the relation

$$\overline{I_{dark}(i,j)} = \frac{1}{P}\sum_{f=1}^{P} D(i,j,f,0), \tag{5.5}$$

where for this experiment $P = 100$. This equation averages out temporal variations, which appear as frame-to-frame differences in a particular pixel's response, to generate an estimate

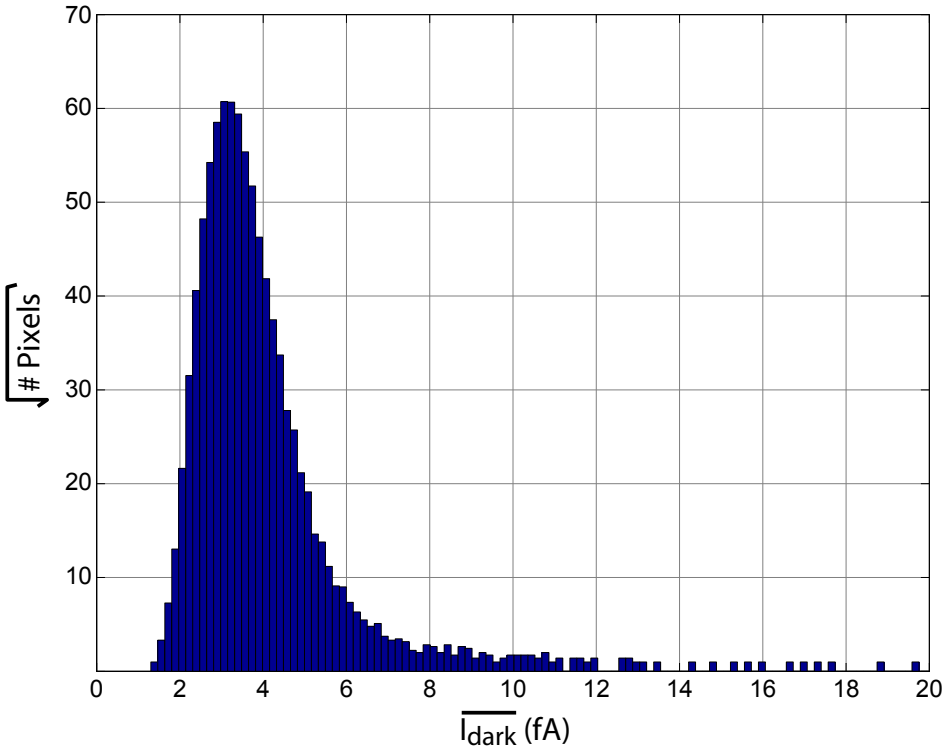

Figure 5-7: Histogram of square-root the number of pixels exhibiting a particular $\overline{I_{dark}}$. An additional 19 pixels are not represented in this histogram, as they exhibit $\overline{I_{dark}} > 20$ fA.

of $\overline{I_{dark}(i,j)}$ for each pixel. It is interesting to plot a histogram of the results, which is shown in Figure 5-7. Note that the bars in this plot represent the square-root of the number of pixels exhibiting a particular $\overline{I_{dark}}$. This compressed axis allows the outliers to the right of the central distribution to be easily observed. Furthermore, as mentioned in the caption, 19 pixels exhibit $\overline{I_{dark}} > 20$ fA, with the largest at 85 fA. The pixels to the right of the central distribution manifest as bright spots in the image, and are one of the primary reasons for the push towards pinned-photodiode technologies [81].

Returning to the task of characterizing the statistics of the imager dark current, the mean dark current over the array can be calculated by substituting the result from Equation 5.5 into the following relation

$$\overline{I_{dark,array}} = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} \overline{I_{dark}(i,j)}, \tag{5.6}$$

where $N = 150$ and $M = 256$ since we are characterizing the dark current over the photo-

sensitive portion of the array. The resulting average is

$$\overline{I_{dark,array}} = 3.4 \text{ fA}. \tag{5.7}$$

Note that this level is nearly nine times larger than the predicted value calculated in Section 4.1.1 based on Equation 4.2. As mentioned in that discussion, this can mostly be attributed to process-dependent generation mechanisms, for which no data was available a priori.

Before calculating the dark current variance, note that the goal is to characterize the average pixel-to-pixel mismatch, which will be quantified as a fixed-pattern noise term. Since variance calculations are susceptible to corruption by a few large outliers, it makes sense to limit this effect by clipping the dark current at some maximum level $\kappa \cdot \overline{I_{dark,array}}$ as follows

$$\overline{I_{dark,clipped}(i,j)} = \begin{cases} \overline{I_{dark}(i,j)} & \text{if } \overline{I_{dark}(i,j)} \leq \kappa \cdot \overline{I_{dark,array}}, \\ \kappa \cdot \overline{I_{dark,array}} & \text{if } \overline{I_{dark}(i,j)} > \kappa \cdot \overline{I_{dark,array}}. \end{cases} \tag{5.8}$$

This clipped dark current array can then be used to estimate the variance of the central distribution using the relation

$$\sigma^2_{I_{dark,array}} = \frac{1}{(N \cdot M) - 1} \sum_{i=1}^{N} \sum_{j=1}^{M} [\, \overline{I_{dark,clipped}(i,j)} - \overline{I_{dark,clipped}} \,]^2, \tag{5.9}$$

where

$$\overline{I_{dark,clipped}} = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{j=1}^{M} \overline{I_{dark,clipped}(i,j)}. \tag{5.10}$$

Evaluating these relations with $N = 150$, $M = 256$, and $\kappa = 5$ yields a variance of

$$\sigma^2_{I_{dark,array}} = (0.904 \text{ fA})^2, \tag{5.11}$$

which corresponds to roughly 25% relative pixel-to-pixel dark current mismatch.

5.6 Pixel Quantum Efficiency

As discussed in Chapter 2, due primarily to the spatial dependencies of the photon-to-electron conversion process, the number of electrons collected by a photodiode as a result of

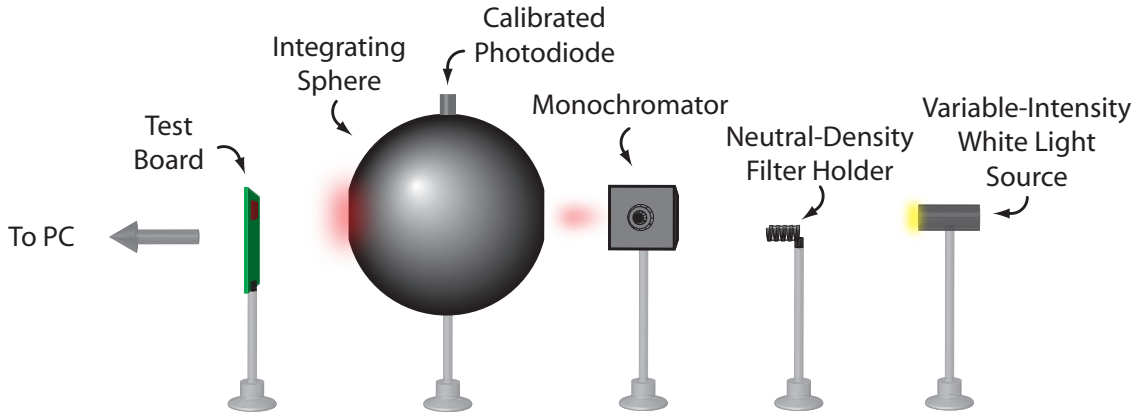

Figure 5-8: Optical test bench used to characterize imager.

each arriving photon will be less than unity. The parameter used to characterize this effect was first introduced in Equation 2.2, and is repeated here for ease

$$\eta(\lambda) = \frac{I_{photo}}{q} \cdot \frac{h \cdot c}{\lambda} \cdot \frac{1}{I_{lux} \cdot f(\lambda) \cdot A_{pd}}. \tag{5.12}$$

This parameter $\eta(\lambda)$ quantifies the number of charge quanta collected by the photodiode per incident photon in terms of the photon wavelength λ, the photodiode current I_{photo}, the input illuminance I_{lux}, the wavelength-dependent illuminance-to-irradiance conversion factor $f(\lambda)$, and the photodiode area A_{pd}, along with Planck's constant $h = 6.626 \times 10^{-34}$ J·s, the speed of light $c = 3 \times 10^8$ m/s, and the unit charge $q = 1.602 \times 10^{-19}$ C. Characterizing the photodiode quantum efficiency is a necessary step in bridging the gap between the noise analysis presented in Chapter 3, which was formulated in terms of I_{photo}, and the experimental data of this chapter, in which the independent variable is I_{lux}.

This experiment marks the first of several that will make use of various components from the optical test bench shown in Figure 5-8. In this experiment, the output of a DC-regulated white light source (Dolan-Jenner Model DC950) is fed to a digital monochromator (Mini-Chrom Model DMC1-03) which allows a narrow but tunable bandwidth ($\approx$ 2 nm) of optical frequencies to pass through to its output [82, 83]. The monochromator feeds an integrating sphere (Edmund Optics Model NT58-585) which collects and randomizes the light profile through numerous diffuse internal reflections [84]. An internal baffle also ensures that no direct optical coupling occurs between the input and output ports of the

sphere. Due to its randomizing internal reflections, the directed beam at the sphere's input is transformed into a spatially uniform illuminance at its output, which is used to expose every pixel in the lens-less prototype imager to nearly the same illuminance[2]. The absolute value of this illuminance can be simultaneously measured at a second sphere output port using a NIST-traceable calibrated photodiode (Thorlabs Model FDS100-CAL) [85].

Using this test bench, the response of the imager was characterized at 10 nm intervals over the optical spectrum from 400 nm to 700 nm. At each wavelength 100 frames of image data were captured while the calibrated photodiode was used to simultaneously record a calibration photocurrent $I_{photo,cal}$. Due to the uniform input illumination, the average pixel response can be calculated at each wavelength using

$$\overline{I_{photo}(\lambda)} = \frac{1}{N \cdot M \cdot P} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{f=1}^{P} [\, D(i,j,f,I_{lux},\lambda) - \overline{I_{dark,array}} \,], \tag{5.13}$$

with $N = 150$, $M = 256$, and $P = 100$. Notice that to help improve the accuracy of this measurement, the mean dark current of the array, calculated in Equation 5.6, is subtracted from each frame. Using the calibration curve provided with the FDS100-CAL photodiode to calculate the pixel irradiance $I_{lux} \cdot f(\lambda)$, along with the value $A_{pd} = 6.55\ \mu\text{m} \times 10.34\ \mu\text{m}$, the quantum efficiency at each wavelength was calculated using Equation 5.12 and is shown in Figure 5-9. At its peak of $\lambda = 500$ nm, the photodiode collects an average 0.495 electrons per incident photon.

5.7 Pixel Transfer Characteristic

To measure the pixel transfer characteristic, several modifications were made to the optical test bench of Figure 5-8. First, since the calibrated photodiode's spectral response is not lux-weighted, it can only be used to measure the irradiance of sources with a known spectral composition. Therefore, the white light source and monochromator were replaced with an array of 520 nm light-emitting diodes (LEDs), allowing for a wide range of input intensities with known spectral composition. By assuming the bulk of the radiant flux lies near the LED peak wavelength, each irradiance measurement taken with the calibrated photodiode can be converted to a corresponding illuminance. The second modification to the setup was the use

[2]The lens and lens mount were absent during all of the characterization experiments, except during the sample frame captures in Section 5.12.

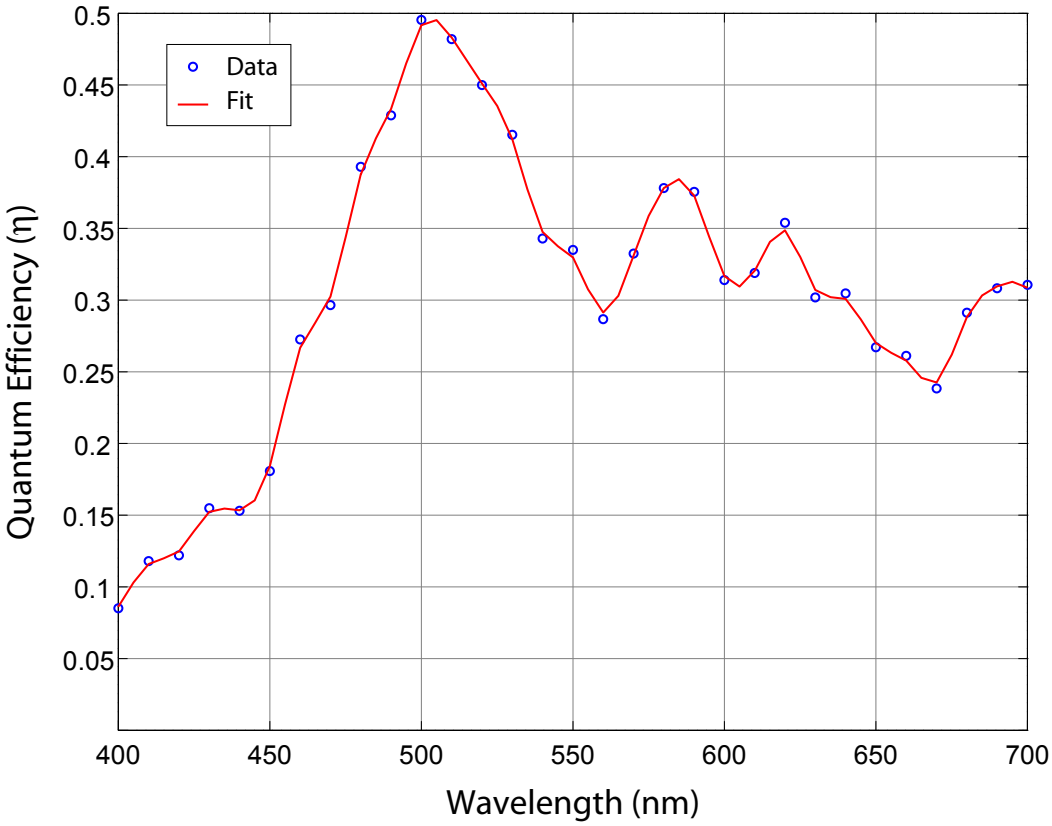

Figure 5-9: Measured quantum efficiency $\eta(\lambda)$ of imager photodiode.

of the neutral-density filter holder shown in the figure, which was placed between the LED array and the integrating sphere. When necessary, either a 10% transmission (Thorlabs Model NE10A) or 1% transmission (Thorlabs Model NE20A) absorptive neutral-density filter could be added to the holder to attenuate the radiant flux reaching the integrating sphere input port from the LEDs [86].

Using this experimental setup, the imager response was characterized at five log-spaced points per decade over the range 10^{-2} lux $\leq I_{lux} \leq 10^{4}$ lux, with a total of 100 frames recorded at each illuminance. For $I_{lux} \geq 1$ lux, no neutral density filters were needed, as the incident illuminance could be accurately measured using the calibrated photodiode. For $I_{lux} < 1$ lux, the incident illuminance was first accurately set a factor of f_{ND} above the desired level using the calibrated photodiode, and a neutral density filter with a transmission of $\frac{100}{f_{ND}}$ percent was placed between the LED array and the integrating sphere during frame capture. To obtain maximum accuracy with this method, the responses of both the 10%

and 1% neutral density filters were separately characterized, and were found to transmit $\frac{100}{f_{ND,10\%}} = 8.398\%$ and $\frac{100}{f_{ND,1\%}} = 0.761\%$ of the LED radiant flux, respectively. After processing the resulting dual-threshold digital pixel data using Equation 3.9 to obtain the data array $D(i,j,f,I_{lux})$, the mean response of each pixel was calculated using

$$\overline{I_{photo}(i,j,I_{lux})} = \frac{1}{P}\sum_{f=1}^{P}[\, D(i,j,f,I_{lux}) - \overline{I_{dark,array}}\,], \tag{5.14}$$

where $P = 100$ and again the mean array dark current calculated in Equation 5.6 is subtracted to help improve measurement accuracy. The mean array response was then calculated using

$$\overline{I_{photo}(I_{lux})} = \frac{1}{N \cdot M}\sum_{i=1}^{N}\sum_{j=1}^{M}\overline{I_{photo}(i,j,I_{lux})}, \tag{5.15}$$

with $N = 150$ and $M = 256$. The measured transfer characteristic is shown in Figure 5-10, along with a plot of the ideal characteristic

$$I_{photo} = I_{lux} \cdot 26.34 \times 10^{-15}\ \text{A/lux}, \tag{5.16}$$

which was generated using Equation 5.12 and the values $A_{pd} = 6.55\ \mu\text{m} \times 10.34\ \mu\text{m}$, $\lambda = 520$ nm, $f(\lambda) \approx \frac{1}{(0.71)\cdot 683}\ \frac{\text{W}}{\text{lux}\cdot\text{m}^2}$, and $\eta(\lambda) = 0.450$ e$^-$/photon. The two characteristics, which are in good agreement, demonstrate the inherent linearity of the capacitively-coupled dual-threshold pixel response.

Finally, notice that the pixels saturate at $I_{lux} = 2274$ lux, corresponding to a measured value of $I_{photo,max} = 68.15$ pA. The V_{thresh} waveform employed in this experiment was generated using $\Delta T_{min} = 1.252\ \mu$s and $t_{frame,end} = 30$ ms, and achieved a value of $t_{off,end} = 130.1\ \mu$s. With $C_{coup} = 8$ fF and $V_{thresh,max} \approx 1.15$ V due to roughly 50 mV of the swing being consumed by the offset phase at high photocurrents, the theoretical maximum photocurrent can be estimated using Equation 3.11 to be 70.7 pA, in good agreement with the measured level.

5.8 Pixel Responsivity

The pixel responsivity was previously defined in Section 4.1.1 as the time-rate-of-change in the threshold waveform as a function of the input illuminance $\frac{dV_{thresh}}{dt}(I_{lux})$ which was

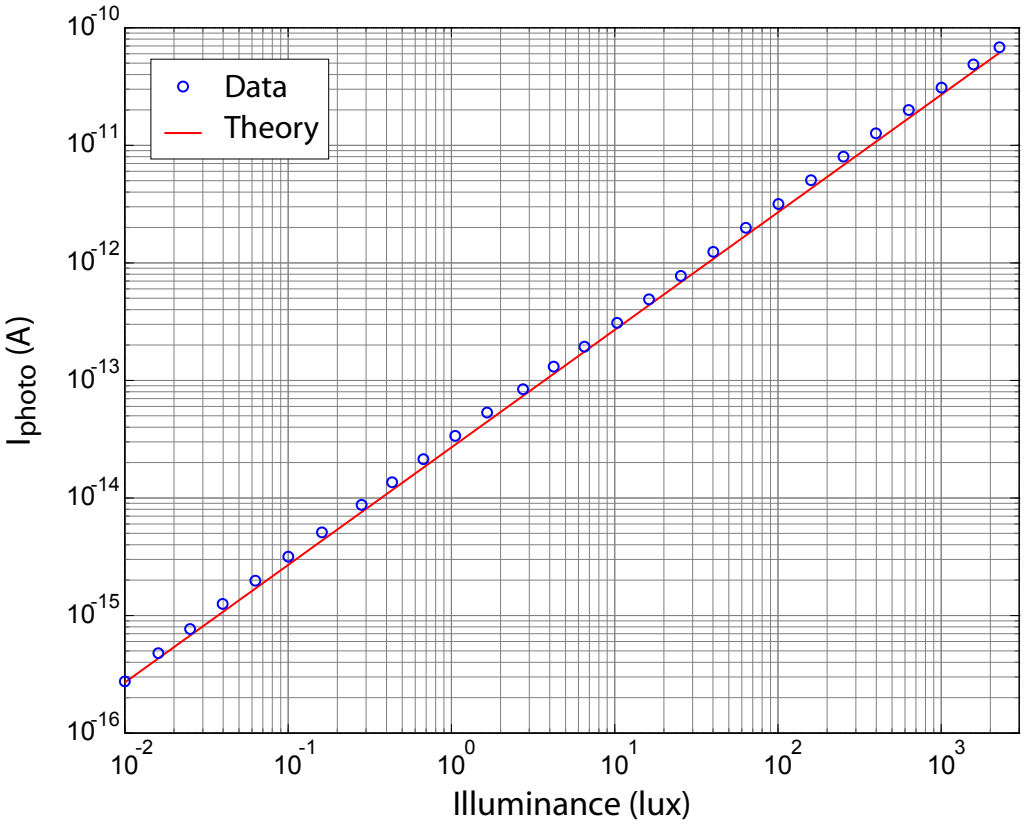

Figure 5-10: The measured and theoretical pixel transfer characteristics with $t_{frame,end} =$ 30 ms demonstrate the inherent linearity of capacitively-coupled dual-threshold pixel response.

re-interpreted in Equation 4.7 in the following form

$$\text{Responsivity} = \frac{I_{photo}/I_{lux}}{C_{coup}}. \tag{5.17}$$

Based on the level of matching between the measured and ideal transfer characteristics depicted in Figure 5-10, the ideal relationship given in Equation 5.16 can be substituted directly into Equation 5.17 to find

$$\text{Responsivity} = \frac{26.34 \times 10^{-15}\ \text{A/lux}}{8 \times 10^{-15}\ \text{F}} = 3.292\ \frac{\text{V}}{\text{lux} \cdot \text{s}}. \tag{5.18}$$

As discussed previously, the primary way to improve this metric is by reducing the value of C_{coup} employed in the pixel design.

5.9 Pixel Signal-to-Noise Ratio and Dynamic Range

In addition to enabling the transfer characteristic of Figure 5-10 to be plotted, the data recorded in Section 5.7 can also be used to determine the pixel SNR and dynamic range as a function of I_{lux}. The pixel SNR is based on a measure of pixel noise power that excludes pixel-to-pixel FPN effects, which can be generated by calculating the mean intra-pixel variance at each level of illumination using the relation

$$\overline{\sigma^2_{I_{photo,pixel}}(I_{lux})} = \frac{1}{N \cdot M \cdot (P-1)} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{f=1}^{P} [\, D(i,j,f,I_{lux}) - \overline{I_{photo}(i,j,I_{lux})}\,]^2. \quad (5.19)$$

Substituting this result, along with the mean response calculated in Equation 5.15, into Equation 3.20 gives

$$\mathrm{SNR}_{\mathrm{pixel}}(I_{lux}) = 10 \cdot \log_{10} \left[\frac{\overline{I_{photo}(I_{lux})}^2}{\overline{\sigma^2_{I_{photo,pixel}}(I_{lux})}} \right]. \quad (5.20)$$

A plot of the measured pixel SNR versus illuminance, along with the response predicted by the noise theory of Chapter 3, is shown in Figure 5-11. The plot demonstrates that the pixels experimentally achieve 98.8 dB dynamic range and 44 dB peak SNR. Initially the agreement between measurement and theory was poor due to a higher-than-expected pixel noise floor. Though reset noise does not affect the dual-threshold pixel noise floor, by measuring the frame-to-frame covariance in $V_{thresh}(t_{off})$ and $V_{thresh}(t_{thresh})$ for individual pixels, this noise term was also observed to be significantly higher than expected. Both effects were ultimately traced to the photodiode voltage-dependent parasitic capacitance, which was experimentally shown to be a factor of 4.46 times larger than the value predicted using process data in Equation 4.4. As shown in Equations 3.17 and 3.18, this results in an increase in both reset and read noise charge uncertainty at V_{pd}. While this effect brings the expected and measured reset noise into agreement, an additional factor of two increase in the theoretical comparator input-referred noise is necessary to achieve good agreement between the predicted and measured total pixel noise floors.

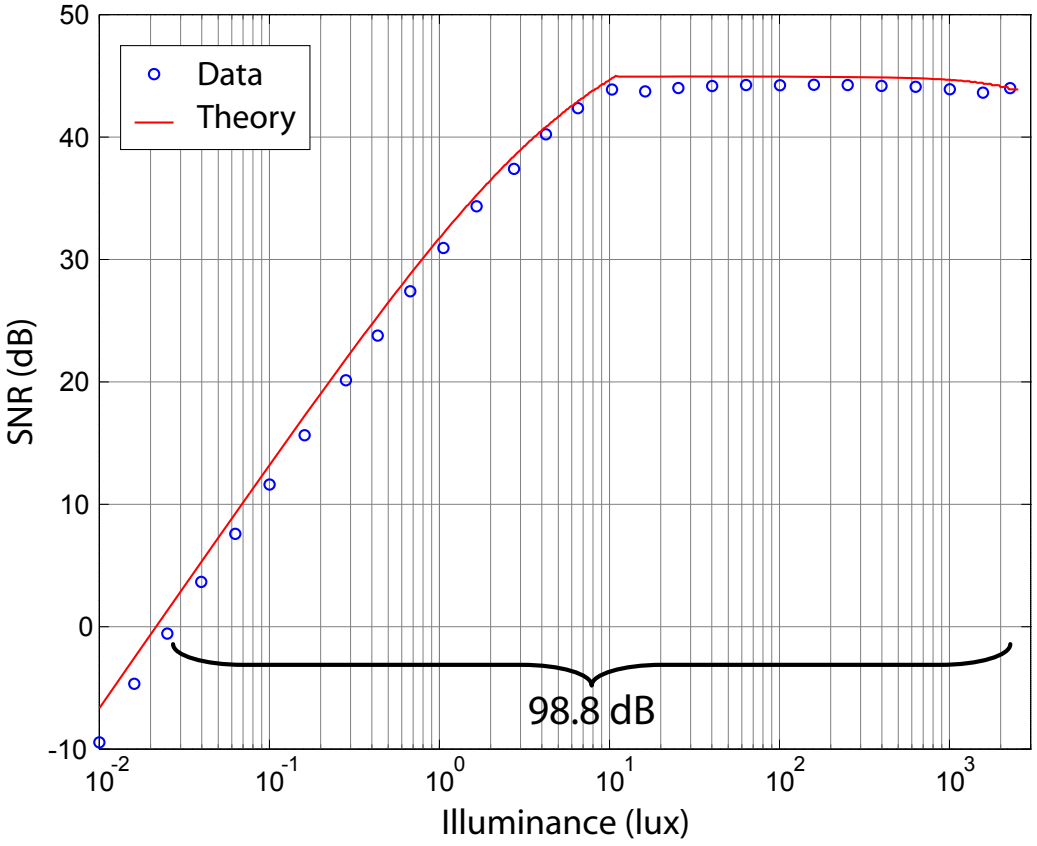

Figure 5-11: The measured and theoretical pixel SNR characteristics with $t_{frame,end} = 30$ ms. The pixels experimentally achieve 98.8 dB dynamic range and 44 dB peak SNR.

5.10 Array Signal-to-Noise Ratio and Dynamic Range

The array SNR is calculated using a measure of pixel noise power that includes the effects of pixel-to-pixel mismatch, which in the dual-threshold pixel can be attributed primarily to gain and dark-current FPN. As with the dark-current characterization presented earlier, to avoid allowing a few outlier pixels with exceptionally large dark current to corrupt the low-illumination variance, the data array $D(i,j,f,I_{lux})$ was post-processed using the following relation with $\kappa = 5$

$$D_{pp}(i,j,f,I_{lux}) = \begin{cases} [\, D(i,j,f,I_{lux}) - \overline{I_{dark,array}} \,] & \text{if } \overline{I_{dark}(i,j)} \leq \kappa \cdot \overline{I_{dark,array}}, \\ [\, D(i,j,f,I_{lux}) - \overline{I_{dark}(i,j)} + \kappa \cdot \overline{I_{dark,array}} \,] & \text{if } \overline{I_{dark}(i,j)} > \kappa \cdot \overline{I_{dark,array}}. \end{cases} \tag{5.21}$$

The mean and variance of this post-processed data as a function of illuminance were then calculated using

$$\overline{I_{photo,pp}(I_{lux})} = \frac{1}{N \cdot M \cdot P} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{f=1}^{P} D_{pp}(i,j,f,I_{lux}), \tag{5.22}$$

and

$$\sigma^2_{I_{photo,array}}(I_{lux}) = \frac{1}{(N \cdot M \cdot P) - 1} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{f=1}^{P} [\, D_{pp}(i,j,f,I_{lux}) - \overline{I_{photo,pp}(I_{lux})}\,]^2, \tag{5.23}$$

where in both equations $N = 150$, $M = 256$, and $P = 100$. Substituting these two results into Equation 3.20 gives

$$\mathrm{SNR}_{\mathrm{array}}(I_{lux}) = 10 \cdot \log_{10} \left[\frac{\overline{I_{photo,pp}(I_{lux})}^2}{\sigma^2_{I_{photo,array}}(I_{lux})} \right]. \tag{5.24}$$

A plot of the measured array SNR versus illuminance, along with the response predicted by the noise theory of Chapter 3, is shown in Figure 5-12. The plot demonstrates that the array experimentally achieves 95.5 dB dynamic range with 37 dB peak SNR, and the level of agreement between measurement and theory is excellent. As expected, the peak array SNR is lower than the peak pixel SNR due to the noise power added by gain FPN, which based on the measurements is around 1.5%. The array dynamic range is also slightly below that of the individual pixels due to the noise power added by dark-current FPN.

To quantify the SNR and dynamic range improvement provided by the dual-threshold algorithm, the raw dual-threshold pixel data was re-processed using Equation 3.9. However, instead of subtracting the measured offset voltage $V_{thresh}(t_{off})$ and time t_{off} on a pixel-by-pixel and frame-by-frame basis, only the mean levels $\overline{V_{thresh}(t_{off})}$ and $\overline{t_{off}}$, averaged over all pixels and all frames at a given illuminance level, were used to calculate each pixel's photocurrent. This step removes fixed average offsets that are common to all pixels, but unlike the true dual-threshold approach, does not remove offset FPN from the resulting data array $D_{off,FPN}(i,j,f,I_{lux})$. Analyzing this data using Equations 5.21-5.24, the array SNR plot shown in Figure 5-13 was generated, which illustrates that the dual-threshold algorithm improves the array dynamic range by more than 6.0 dB compared to auto-zeroing alone. In this case the improvement can be primarily attributed to the algorithm's elimination of

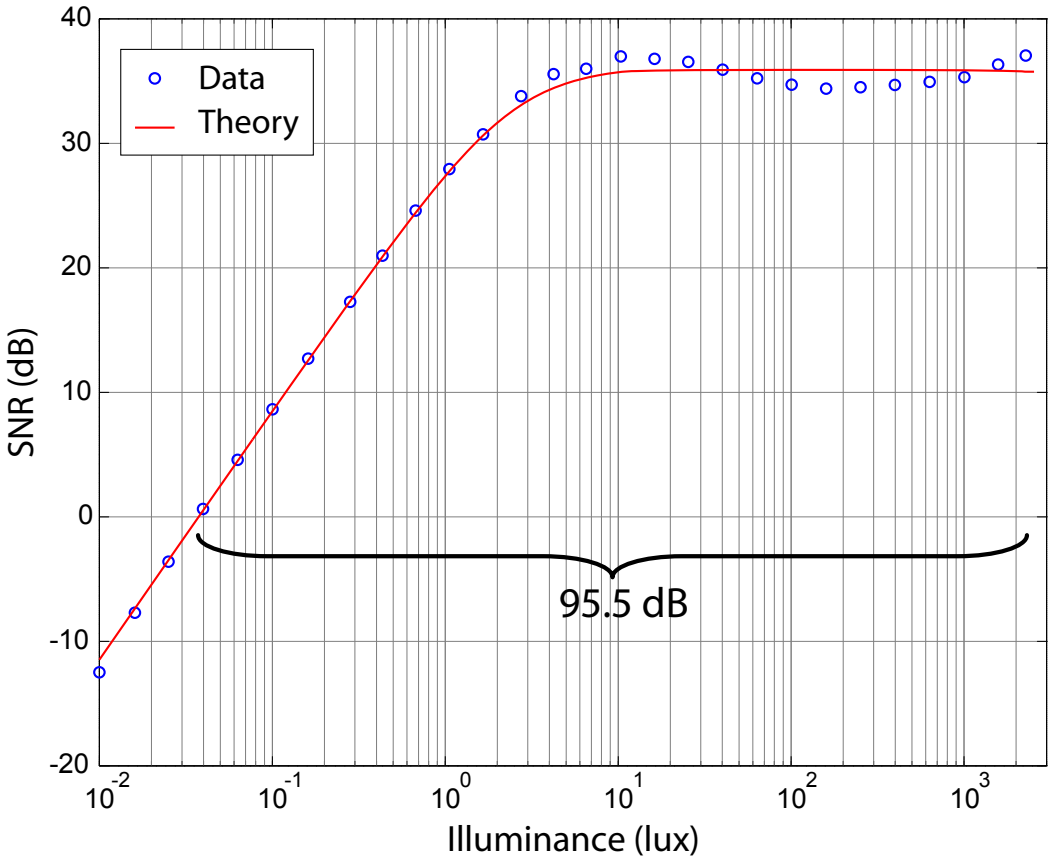

Figure 5-12: The measured and theoretical array SNR characteristics with $t_{frame,end}$ = 30 ms. The array experimentally achieves 95.5 dB dynamic range and 37 dB peak SNR.

pixel reset noise.

5.11 Array Energy Efficiency

Besides dynamic range and SNR, the power consumed by an imaging array during frame capture is another important measure of its performance. To allow for direct comparisons of power consumption across various designs, the following metric was defined

$$\text{Energy Efficiency} = \frac{\text{Total power consumed}}{\#\text{ pixels} \cdot \text{Frame rate}}, \tag{5.25}$$

which normalizes each array's power by the number of pixels it contains and its frame rate. Based on experimental measurements, the prototype dual-threshold imager achieves

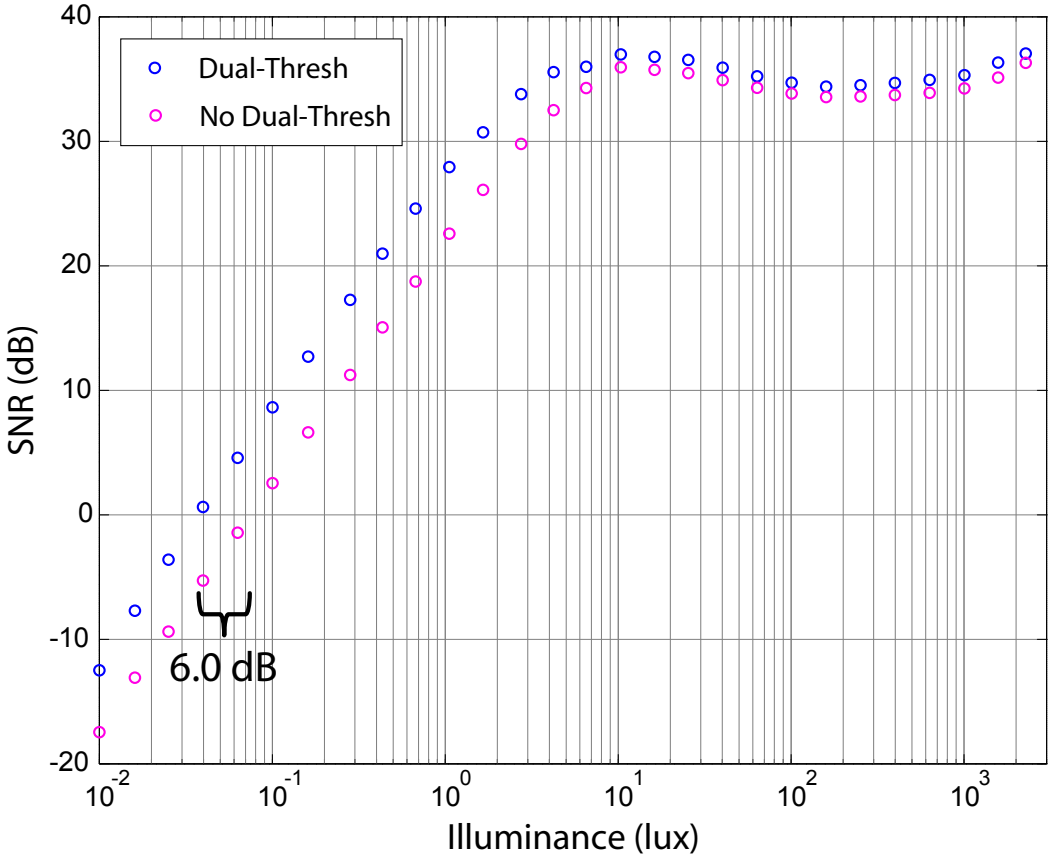

Figure 5-13: Array dynamic range is improved by more than 6.0 dB using the dual-threshold algorithm compared with auto-zeroing alone. In this case the improvement is primarily due to the algorithm's elimination of pixel reset noise.

an energy efficiency of

$$\text{Energy Efficiency} = 1.79 \text{ nJ/pixel/frame.} \tag{5.26}$$

This level is compared with other reported designs in the scatter plot shown in Figure 5-14, which maps each design's achieved energy efficiency versus its reported array dynamic range, with more efficient implementations occurring towards the bottom-right corner of the figure. The plot demonstrates that the prototype dual-threshold imager is one of the most energy-efficient wide-dynamic-range imagers reported.

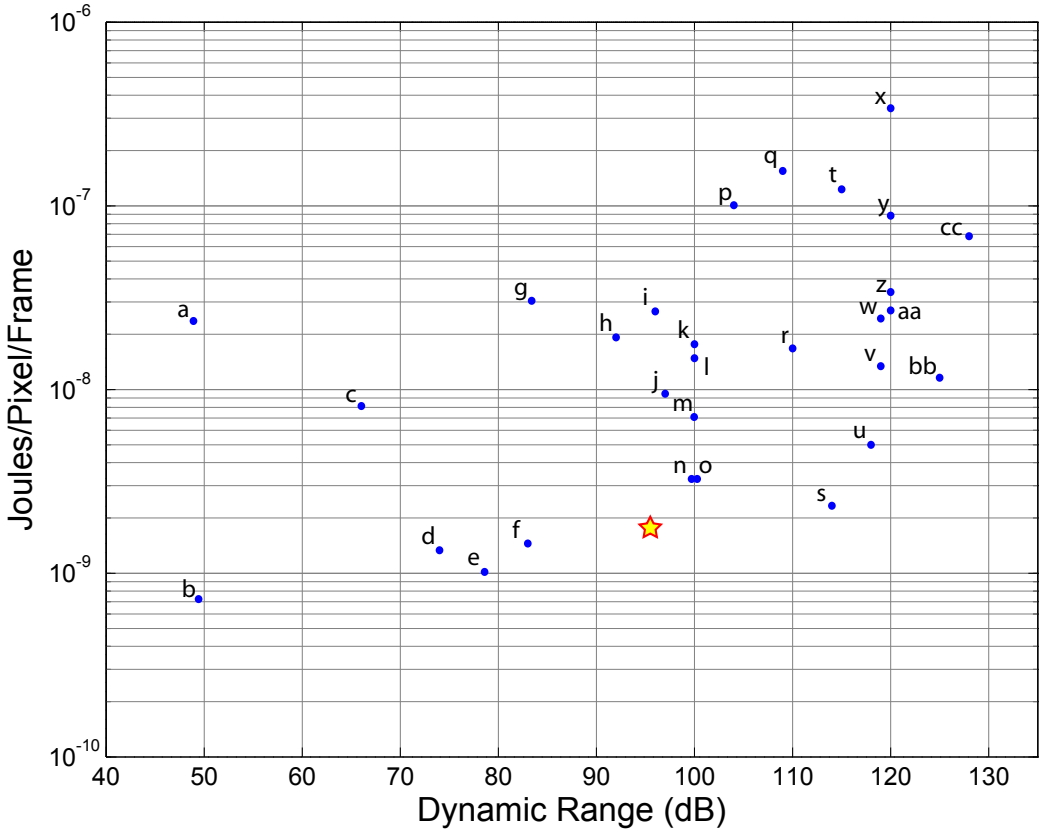

Figure 5-14: Scatter plot of energy efficiency versus array dynamic range for previously reported imager designs. The star represents the performance achieved in the prototype dual-threshold imager. The lettered dots correspond to the following references: a - [52], b - [87], c - [25], d - [57], e - [88], f - [56], g - [37], h - [34], i - [22], j - [30], k - [89], l - [90], m - [24], n - [48], o - [27], p - [67], q - [35], r - [91], s - [70], t - [65], u - [29], v - [31], w - [33], x - [69], y - [39], z - [21], aa - [18], bb - [28], cc - [68].

5.12 Sample Image

To illustrate the intrinsic advantage of wide-dynamic-range operation, the sample 80-dB dynamic range scene shown in Figure 5-15(a) was captured using the prototype 150×256 dual-threshold imager. Notice that both the brightly-illuminated car and the dimly-illuminated white circle under the table are simultaneously recorded by the imager. Compare this with the frame shown in Figure 5-15(b), which illustrates how the same scene would appear if captured with a 62-dB dynamic range CMOS APS imaging array – a typical number for this type of sensor. With the available dynamic range focused on the dimly-illuminated region under the table, all of the information outside the window is lost to saturation. Note that

95.5 dB dynamic range sensor
(a)

62 dB dynamic range sensor
(b)

Figure 5-15: A sample 80-dB dynamic range image captured using the prototype 95.5-dB dynamic range imager (a) and a 62-dB dynamic range imager (b), illustrating a loss of information due to pixel saturation in the latter case. In both cases log compression has been applied to the linear data to allow the image's full dynamic range to be displayed in a single frame, and $t_{frame,end} = 30$ ms.

both images have been log-compressed to enable their full dynamic range to be visible both on-screen and in print, with the minimum and maximum log levels in each image stretched to cover the full grayscale gamut. This gives the false impression that the image in part (b) has higher contrast, due to its reduced dynamic range, when in reality the recorded contrast of both images is the same.

Parameter	Value
Technology	0.18-μm 1.8-V CMOS 1P-6M, N-well
Power supply	1.8 V
Die area	5000 μm $\times$5000 μm
Number of pixels	150(V) $\times$ 280(H) (24 dark columns)
Pixel type	Photodiode
Pixel pitch	12.5 μm$\times$12.5 μm
Fill factor	42.7%
Threshold waveform DAC	12-bit, 1.2 V swing
Coupling capacitance	7.998 fF (measured)
Photodiode capacitance	25.56 fF (measured @ $V_{pd} = 850$ mV)
Other parasitic caps at photonode	2.8 fF (calculated @ $V_{pd} = 850$ mV)
Pixel responsivity	3.292 V/(lux$\cdot$s)
Dark current	5.02 nA/cm^2 (measured @ T = 300K)
Gain FPN	1.5%
Array performance	95.5 dB dynamic range, 37 dB peak SNR (measured)
Pixel performance	98.8 dB dynamic range, 44 dB peak SNR (measured)
Power consumption	375 μW (measured @ 5fps)

Table 5.1: Prototype dual-threshold imager characteristics and performance.

5.13 Summary

This chapter presented the experimental setup and techniques that were employed to test the prototype 150×256 pixel dual-threshold CMOS imager, and presented measured performance data from it. The experimental results were also compared with the theory developed in Chapter 3, as well as with past designs from the literature. A summary of the relevant prototype characteristics is given in Table 5.1.

Chapter 6

Conclusions

6.1 Summary

This thesis focused on improving the performance of video-rate time-based imagers, focusing specifically on the class of time-varying threshold designs that have been previously reported. The following contributions were made to this area:

- A novel dual-threshold time-based current sensing algorithm was proposed that forces each single-slope integrating pixel to cross two threshold levels per frame – once just after reset and a second time after a near-optimal amount of photo-generated charge has been collected. This differential measurement technique eliminates offset FPN and pixel reset noise, and reduces comparator 1/f noise.

- Synchronous threshold detection was employed and was shown to yield significant power savings compared with asynchronous approaches in this application. The resulting time-domain quantization noise introduced by the synchronous threshold detections was also analyzed.

- A method of optimizing the global dual-threshold waveform and associated pixel threshold-detection times was presented. The method ensures that the quantization noise introduced by the algorithm remains negligible compared to the intrinsic pixel noise floor, while simultaneously minimizing the number of threshold detections employed, and thus energy consumed.

- A novel capacitively-coupled pixel topology was introduced that enables highly-linear

responses to be achieved with the dual-threshold algorithm while minimizing the common-mode input range of the pixel comparator, simplifying its design.

These innovations were incorporated into the design of a prototype dual-threshold time-based CMOS imager. The imager implements pixels and their associated 18-bit timing memories in separate on-chip arrays linked by a 200 MHz time-domain-multiplexed communication bus, enabling a pixel pitch of 12.5 μm with 42.7% fill factor in a 0.18-μm 1.8-V CMOS process. The prototype experimentally achieved an array dynamic range of 95.5 dB and 37 dB peak SNR, while consuming 1.79 nJ/pixel/frame, making it one of the most energy-efficient wide-dynamic-range imagers reported.

6.2 Future Work

Several paths are worth exploring in further developing the ideas presented in this thesis. These include:

- The prototype imager performance was limited by poor photodiode parasitic models. Namely, both the dark current and parasitic depletion capacitance of the n-well/p-substrate junction were significantly larger than predicted by the available models. Both effects led to an increased pixel noise floor and reduced dynamic range. Further work in optimizing this photodiode should yield significant improvements in array dynamic range. One avenue worth exploring is to look at adapting the techniques that have been presented for use with pinned-photodiodes.

- The performance of the prototype was further limited by the experimental setup, in which the minimum inter-pixel sampling time ΔT_{min} was almost doubled compared with what the prototype is theoretically capable of achieving. Thus, it is expected that the present design should be capable of achieving an additional 6 dB of dynamic range in the high-illumination region of the response.

- Though the number of columns can scale indefinitely, the number of array rows is limited by the minimum inter-pixel sampling time ΔT_{min}, the pixel-to-memory communication window t_{row_xfer}, and the comparator response time. Reducing t_{row_xfer} by decreasing the noise margins of the pixel-to-memory communication should allow

the present approach to easily scale to arrays with several thousand rows that are capable of achieving comparable performance to that presented in this work.

Bibliography

[1] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *Proc. Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*, Los Angeles, CA, USA, Aug. 1997, pp. 369–378.

[2] B. Hosticka, B. Hosticka, W. Brockherde, A. Bußmann, T. Heimann, R. Jeremias, A. Kemna, C. Nitta, and O. Schrey, "CMOS imaging for automotive applications," *IEEE Trans. Electron Devices*, vol. 50, no. 1, pp. 173–183, 2003.

[3] G. P. Weckler, "Operation of *p-n* junction photodetectors in a photon flux integrating mode," *IEEE J. Solid-State Circuits*, vol. SC-2, no. 3, pp. 65–73, Sep. 1967.

[4] R. S. Muller and T. I. Kamins, *Device Electronics for Integrated Circuits*, 2nd ed. New York: Wiley, 1986.

[5] W. B. Merrill, "Color separation in an active pixel cell imaging array using a triple-well structure," U.S. Patent 5 965 875, Oct. 12, 1999.

[6] E. R. Fossum, "CMOS image sensors: Electronic camera-on-a-chip," *IEEE Trans. Electron Devices*, vol. 44, no. 10, pp. 1689–1698, Oct. 1997.

[7] S. Holland, D. Groom, N. Palaio, R. Stover, and M. Wei, "Fully depleted, back-illuminated charge-coupled devices fabricated on high-resistivity silicon," *IEEE Trans. Electron Devices*, vol. 50, no. 1, pp. 225–238, Jan. 2003.

[8] S. R. Morrison, "A new type of photosensitive junction device," *Solid-State Electron.*, vol. 5, pp. 485–494, 1963.

[9] J. W. Horton, R. V. Mazza, and H. Dym, "The scanistor–A solid-state image scanner," *Proc. IEEE*, vol. 52, no. 12, pp. 1513–1528, Dec. 1964.

[10] I. L. Fujimori, C.-C. Wang, and C. G. Sodini, "A 256×256 CMOS differential passive pixel imager with FPN reduction techniques," *IEEE J. Solid-State Circuits*, vol. 35, no. 12, pp. 2031–2037, Dec. 2000.

[11] P. J. W. Noble, "Self-scanned silicon image detector arrays," *IEEE Trans. Electron Devices*, vol. ED-15, no. 4, pp. 202–209, Apr. 1968.

[12] W. S. Boyle and G. E. Smith, "Charge coupled semiconductor devices," *Bell Syst. Tech. J.*, vol. 49, pp. 587–593, Apr. 1970.

[13] P. W. Fry, P. J. W. Noble, and R. J. Rycroft, "Fixed-pattern noise in photomatrices," *IEEE J. Solid-State Circuits*, vol. SC-5, no. 5, pp. 250–254, Oct. 1970.

[14] S. Chamberlain and J. Lee, "A novel wide dynamic range silicon photodetector and linear imaging array," *IEEE J. Solid-State Circuits*, vol. 19, no. 1, pp. 41–48, Feb. 1984.

[15] T. Delbrück and C. Mead, "Adaptive photoreceptor with wide dynamic range," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS '94)*, vol. 4, London, UK, 1994, pp. 339–342.

[16] S. Kavadias, B. Dierickx, D. Scheffer, A. Alaerts, D. Uwaerts, and J. Bogaerts, "A logarithmic response CMOS image sensor with on-chip calibration," *IEEE J. Solid-State Circuits*, vol. 35, no. 8, pp. 1146–1152, Aug. 2000.

[17] E. C. Fox, J. Hynecek, and D. R. Dykaar, "Wide-dynamic-range pixel with combined linear and logarithmic response and increased signal swing," in *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications*, vol. 3965, no. 1. SPIE, May 2000, pp. 4–10.

[18] M. Loose, K. Meier, and J. Schemmel, "A self-calibrating single-chip CMOS camera with logarithmic response," *IEEE J. Solid-State Circuits*, vol. 36, no. 4, pp. 586–596, Apr. 2001.

[19] D. Joseph and S. Collins, "Modeling, calibration, and correction of nonlinear illumination-dependent fixed pattern noise in logarithmic CMOS image sensors," *IEEE Trans. Instrum. Meas.*, vol. 51, no. 5, pp. 996–1001, Oct. 2002.

[20] G. Storm, J. Hurwitz, D. Renshaw, K. Findlater, R. Henderson, and M. Purcell, "Combined linear-logarithmic CMOS image sensor," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'04)*, San Francisco, CA, USA, Feb. 2004, pp. 116–517.

[21] G. Storm, R. Henderson, J. Hurwitz, D. Renshaw, K. Findlater, and M. Purcell, "Extended dynamic range from a combined linear-logarithmic CMOS image sensor," *IEEE J. Solid-State Circuits*, vol. 41, no. 9, pp. 2095–2106, Sep. 2006.

[22] S. Decker, D. McGrath, K. Brehmer, and C. Sodini, "A 256 × 256 CMOS imaging array with wide dynamic range pixels and column-parallel digital output," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 2081–2091, Dec. 1998.

[23] R. Oi and K. Aizawa, "Wide dynamic range imaging by sensitivity adjustable CMOS image sensor," in *Proc. International Conference on Image Processing (ICIP'03)*, vol. 2, Barcelona, Spain, Sep. 2003, pp. 583–586.

[24] N. Akahane, S. Sugawa, S. Adachi, K. Mori, T. Ishiuchi, and K. Mizobuchi, "A sensitivity and linearity improvement of a 100-dB dynamic range CMOS image sensor using a lateral overflow integration capacitor," *IEEE J. Solid-State Circuits*, vol. 41, no. 4, pp. 851–858, Apr. 2006.

[25] D. Yang, A. Gamal, B. Fowler, and H. Tian, "A 640 × 512 CMOS image sensor with ultrawide dynamic range floating-point pixel-level ADC," *IEEE J. Solid-State Circuits*, vol. 34, no. 12, pp. 1821–1834, Dec. 1999.

[26] P. Acosta-Serafini, I. Masaki, and C. Sodini, "A 1/3" VGA linear wide dynamic range CMOS image sensor implementing a predictive multiple sampling algorithm with overlapping integration intervals," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1487–1496, Sep. 2004.

[27] W. Bidermann, A. E. Gamal, S. Ewedemi, J. Reyneri, H. Tian, D. Wile, and D. Yang, "A 0.18 μm high dynamic range NTSC/PAL imaging system-on-chip with embedded DRAM frame buffer," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'03)*, San Francisco, CA, USA, Feb. 2003, pp. 212–488.

[28] A. Breidenassel, K. Meier, and J. Schemmel, "A flexible scheme for adaptive integration time control CMOS image sensor," in *Proc. IEEE Sensors*, Oct. 2004, pp. 280–283.

[29] W. Brockherde, A. Bußmann, C. Nitta, B. Hosticka, and R. Wertheimer, "High-sensitivity, high-dynamic range 768 × 576 pixel CMOS image sensor," in *Proc. European Solid-State Circuits Conference (ESSCIRC'04)*, Leuven, Belgium, Sep. 2004, pp. 411–414.

[30] S.-W. Han, S.-J. Kim, J.-H. Choi, C.-K. Kim, and E. Yoon, "A high dynamic range CMOS image sensor with in-pixel floating-node analog memory for pixel level integration time control," in *Proc. Symposium on VLSI Circuits*, Honolulu, HI, USA, Jun. 2006, pp. 25–26.

[31] M. Mase, S. Kawahito, M. Sasaki, Y. Wakamori, and M. Furuta, "A wide dynamic range CMOS image sensor with multiple exposure-time signal outputs and 12-bit column-parallel cyclic A/D converters," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2787–2795, Dec. 2005.

[32] T. Ogi, T. Yasuda, T. Hamamoto, and K. Aizawa, "Smart image sensor for wide dynamic range by variable integration time," in *Proc. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI'03)*, Tokyo, Japan, Jul. 2003, pp. 179–184.

[33] M. Schanz, C. Nitta, A. Bußmann, B. Hosticka, and R. Wertheimer, "A high-dynamic-range CMOS image sensor for automotive applications," *IEEE J. Solid-State Circuits*, vol. 35, no. 7, pp. 932–938, Jul. 2000.

[34] O. Schrey, J. Huppertz, G. Filimonovic, A. Bußmann, W. Brockherde, and B. Hosticka, "A 1K×1K high dynamic range CMOS image sensor with on-chip programmable region-of-interest readout," *IEEE J. Solid-State Circuits*, vol. 37, no. 7, pp. 911–915, Jul. 2002.

[35] O. Yadid-Pecht and E. Fossum, "Wide intrascene dynamic range CMOS APS using dual sampling," *IEEE Trans. Electron Devices*, vol. 44, no. 10, pp. 1721–1723, Oct. 1997.

[36] O. Yadid-Pecht and A. Belenky, "Autoscaling CMOS APS with customized increase of dynamic range," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'01)*, San Francisco, CA, USA, Feb. 2001, pp. 100–101.

[37] ——, "In-pixel autoexposure CMOS APS," *IEEE J. Solid-State Circuits*, vol. 38, no. 8, pp. 1425–1428, Aug. 2003.

[38] S. Benthien, T. Lule, B. Schneider, M. Wagner, M. Verhoeven, and M. Bohm, "Vertically integrated sensors for advanced imaging applications," *IEEE J. Solid-State Circuits*, vol. 35, no. 7, pp. 939–945, Jul. 2000.

[39] T. Lule, M. Wagner, M. Verhoeven, H. Keller, and M. Bohm, "100000-pixel, 120-dB imager in TFA technology," *IEEE J. Solid-State Circuits*, vol. 35, no. 5, pp. 732–739, May 2000.

[40] J. M. Raynor and P. Seitz, "A linear array of photodetectors with wide dynamic range and near photon quantum-noise limit," *Sensors and Actuators A*, vol. 61, pp. 327–330, Jun. 1997.

[41] V. Brajovic and T. Kanade, "A VLSI sorting image sensor: Global massively parallel intensity-to-time processing for low-latency adaptive vision," *IEEE Trans. Robot. Autom.*, vol. 15, no. 1, pp. 67–75, Feb. 1999.

[42] A. Bermak, "A CMOS imager with PFM/PWM based analog-to-digital converter," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'02)*, vol. 4, Scottsdale, AZ, USA, May 2002, pp. 53–56.

[43] A. Kitchen, A. Bermak, and A. Bouzerdoum, "PWM digital pixel sensor based on asynchronous self-resetting scheme," *IEEE Electron Device Lett.*, vol. 25, no. 7, pp. 471–473, Jul. 2004.

[44] ——, "A digital pixel sensor array with programmable dynamic range," *IEEE Trans. Electron Devices*, vol. 52, no. 12, pp. 2591–2601, Dec. 2005.

[45] A. Bermak and Y.-F. Yung, "A DPS array with programmable resolution and reconfigurable conversion time," *IEEE Trans. VLSI Syst.*, vol. 14, no. 1, pp. 15–22, Jan. 2006.

[46] A. Bermak and A. Kitchen, "A novel adaptive logarithmic digital pixel sensor," *IEEE Photon. Technol. Lett.*, vol. 18, no. 20, pp. 2147–2149, Oct. 2006.

[47] J. Harris, "The changing roles of analog and digital signal processing in CMOS image sensors," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, vol. 4, Orlando, FL, USA, May 2002, pp. 3976–3979.

[48] W. Yang, "A wide-dynamic-range, low-power photosensor array," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'94)*, San Francisco, CA, USA, Feb. 1994, pp. 230–231.

[49] F. Andoh, H. Shimamoto, and Y. Fujita, "A digital pixel image sensor for real-time readout," *IEEE Trans. Electron Devices*, vol. 47, no. 11, pp. 2123–2127, Nov. 2000.

[50] E. Culurciello, R. Etienne-Cummings, and K. Boahen, "High dynamic range, arbitrated address event representation digital imager," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'01)*, vol. 3, Sydney, Australia, May 2001, pp. 505–508.

[51] J. Doge, G. Schonfelder, G. Streil, and A. Konig, "An HDR CMOS image sensor with spiking pixels, pixel-level ADC, and linear characteristics," *IEEE Trans. Circuits Syst. II*, vol. 49, no. 2, pp. 155–158, Feb. 2002.

[52] E. Culurciello, R. Etienne-Cummings, and K. Boahen, "A biomorphic digital image sensor," *IEEE J. Solid-State Circuits*, vol. 38, no. 2, pp. 281–294, Feb. 2003.

[53] K. Boahen, "A throughput-on-demand address-event transmitter for neuromorphic chips," in *Proc. 20th Anniversary Conference on Advanced Research in VLSI*, Atlanta, GA, USA, Mar. 1999, pp. 72–86.

[54] B. Fowler, A. E. Gamal, and D. Yang, "A CMOS area image sensor with pixel-level A/D conversion," in *Proc. IEEE International Solid-State Circuits Conference (ISSCC'94)*, San Francisco, CA, USA, Feb. 1994, pp. 226–227.

[55] B. Fowler, "CMOS area image sensors with pixel level A/D conversion," Ph.D. dissertation, Stanford University, Oct. 1995.

[56] D. Yang, B. Fowler, and A. El Gamal, "A 128×128 pixel CMOS area image sensor with multiplexed pixel level A/D conversion," in *Proc. IEEE Custom Integrated Circuits Conference (CICC'96)*, San Diego, CA, USA, May 1996, pp. 303–306.

[57] L. McIlrath, "A low-power low-noise ultrawide-dynamic-range CMOS imager with pixel-parallel A/D conversion," *IEEE J. Solid-State Circuits*, vol. 36, no. 5, pp. 846–853, May 2001.

[58] J. Fortier, N. Tarr, A. Swaminathan, and C. Plett, "1.2V 0.18 μm CMOS imager with column-level oversampling," in *Proc. European Solid-State Circuits Conference (ESSCIRC'01)*, Villach, Austria, Sep. 2001, pp. 105–108.

[59] Z. Ignjatovic 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," in *Proc. Symposium on VLSI Circuits*, Honolulu, HI, USA, Jun. 2006, pp. 23–24.

[60] Y. Ni, F. Devos, M. Boujrad, and J. H. Guan, "Histogram-equalization-based adaptive image sensor for real-time vision," *IEEE J. Solid-State Circuits*, vol. 32, no. 7, pp. 1027–1036, Jul. 1997.

[61] X. Guo, M. Erwin, and J. Harris, "Ultra-wide dynamic range CMOS imager using pixel-threshold firing," in *Proc. World Multiconference on Systemics, Cybernetics, and Informatics (WMSCI'01)*, vol. 15, Orlando, FL, USA, Jul. 2001, pp. 485–489.

[62] S. Kleinfelder, S. Lim, X. Liu, and A. E. Gamal, "A 10000 frames/s CMOS digital pixel sensor," *IEEE J. Solid-State Circuits*, vol. 36, no. 12, pp. 2049–2059, Dec. 2001.

[63] X. Guo, "A time-based asynchronous readout CMOS image sensor," Ph.D. dissertation, University of Florida, Gainesville, FL, Dec. 2002.

[64] Q. Luo, "A time-based synchronous readout CMOS imager," Ph.D. dissertation, University of Florida, Gainesville, FL, Dec. 2002.

[65] Q. Luo, Q. Luo, and J. Harris, "A time-based CMOS image sensor," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'04)*, vol. 4, Vancouver, BC, May 2004, pp. 840–843.

[66] X. Qi, X. Guo, and J. Harris, "A time-to-first spike CMOS imager," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'04)*, vol. 4, Vancouver, BC, May 2004, pp. 824–827.

[67] X. Guo, X. Qi, and J. Harris, "A time-to-first-spike CMOS image sensor," *IEEE Sensors J.*, vol. 7, no. 8, pp. 1165–1175, Aug. 2007.

[68] D. Stoppa, A. Simoni, L. Gonzo, M. Gottardi, and G.-F. D. Betta, "Novel CMOS image sensor with a 132-dB dynamic range," *IEEE J. Solid-State Circuits*, vol. 37, no. 12, pp. 1846–1852, Dec. 2002.

[69] D. Stoppa, M. Vatteroni, D. Covi, A. Baschirotto, A. Sartori, and A. Simoni, "A 120-dB dynamic range CMOS image sensor with programmable power responsivity," *IEEE J. Solid-State Circuits*, vol. 42, no. 7, pp. 1555–1563, Jul. 2007.

[70] S.-M. Lee, H. Park, and B. Wooley, "Per-pixel floating-point ADCs with electronic shutters for a high dynamic range, high frame rate infrared focal plane array," in *Proc. IEEE Custom Integrated Circuits Conference (CICC'06)*, San Jose, CA, USA, Sep. 2006, pp. 647–650.

[71] C.-H. Lai, Y.-C. King, and S.-Y. Huang, "A 1.2-V 0.25-μm clock output pixel architecture with wide dynamic range and self-offset cancellation," *IEEE Sensors J.*, vol. 6, no. 2, pp. 398–405, Apr. 2006.

[72] "MATLAB," The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760, USA. [Online]. Available: www.mathworks.com

[73] M. O'Halloran and R. Sarpeshkar, "A 10-nW 12-bit accurate analog storage cell with 10-aA leakage," *IEEE J. Solid-State Circuits*, vol. 39, no. 11, pp. 1985–1996, Nov. 2004.

[74] Y. P. Tsividis, *Operation and Modeling of the MOS Transistor*, 2nd ed. New York: McGraw-Hill, 1999.

[75] D. A. Johns and K. Martin, *Analog Integrated Circuit Design*, 1st ed. New York: Wiley, 1997.

[76] J. M. Rabaey, *Digital Integrated Circuits: A Design Perspective*, 1st ed. New Jersey: Prentice Hall, 1996.

[77] "AT91SAM 32-bit ARM microcontrollers," Atmel Corporation, 2325 Orchard Parkway, San Jose, CA 95131, USA. [Online]. Available: www.atmel.com

[78] "Universal serial bus device class definition for video devices v1.1," Jun. 2005. [Online]. Available: www.usb.org

[79] "Windows XP," Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA. [Online]. Available: www.microsoft.com

[80] "4300 series miniature glass lenses," Marshall Electronics, Inc., 1910 E. Maple Ave., El Segundo, CA 90245, USA. [Online]. Available: www.mars-cam.com

[81] R. Guidash, T.-H. Lee, P. Lee, D. Sackett, C. Drowley, M. Swenson, L. Arbaugh, R. Hollstein, F. Shapiro, and S. Domer, "A 0.6 μm CMOS pinned photodiode color imager technology," in *Proc. IEEE International Electron Devices Meeting (IEDM'97)*, Washington, D.C., USA, Dec. 1997, pp. 927–929.

[82] "DC950 fiber-optic illuminator," Dolan-Jenner Industries, 159 Swanson Road, Boxborough, MA 01719, USA. [Online]. Available: www.dolan-jenner.com

[83] "Mini-chrom DMC1-03 digital monochromator," Optometrics, 8 Nemco Way, Stony Brook Industrial Park, Ayer, MA 01432, USA. [Online]. Available: www.optometrics.com

[84] "General purpose integrating sphere NT58-585," Edmund Optics, Inc., 101 East Gloucester Pike, Barrington, NJ 08007, USA. [Online]. Available: www.edmundoptics.com

[85] "FDS100-CAL NIST-traceable Si photodiode," Thorlabs, 435 Route 206 North, Newton, NJ 07860, USA. [Online]. Available: www.thorlabs.com

[86] "NE10A and NE20A absorptive neutral density filters," Thorlabs, 435 Route 206 North, Newton, NJ 07860, USA. [Online]. Available: www.thorlabs.com

[87] K.-B. Cho, A. Krymski, and E. Fossum, "A 1.5-V 550-μW 176 $\times$ 144 autonomous CMOS active pixel image sensor," *IEEE Trans. Electron Devices*, vol. 50, no. 1, pp. 96–105, Jan. 2003.

[88] B. Pain, G. Yang, B. Olson, T. Shaw, M. Ortiz, J. Heynssens, C. Wrigley, and C. Ho, "A low-power digital camera-on-a-chip implemented in CMOS active pixel approach," in *Proc. International Conference On VLSI Design*, Goa, India, Jan. 1999, pp. 26–31.

[89] "OV10620 high dynamic range imager," OmniVision, 1341 Orleans Drive, Sunnyvale, CA 94089, USA. [Online]. Available: www.ovt.com

[90] "MT9V032 high dynamic range imager," Micron Technology, Inc., 8000 S. Federal Way, P.O. Box 6, Boise, ID 83707, USA. [Online]. Available: www.ovt.com

[91] "KAC-9618 high dynamic range imager," Eastman Kodak Company, Image Sensor Solutions, 1999 Lake Avenue, Rochester, NY 14650, USA. [Online]. Available: www.kodak.com